Embedded Network System Design and Implementation of Embedded Network For Sensors With Monitoring And Storage

Dilipsai V[1], Asmitha Prabhu B[2], Shreyas D[3], Doddakula Meghana[4], G. Divya Muralidhar[5]
[1, 2, 3, 4] Dept of Electronics And Communication Engineering
[5] Asst Professor, Dept of Electronics And Communication Engineering
[1,2,3,4,5] Nitte Meenakshi Institute Of Technology, Bengaluru

Abstract- *In recent decade, industries have turned into smart industries by the advent of recently developed technologies around the world. The growth in technologies like networking and other provides a brighter future and open scenario to interconnect different equipments in the industries, which gives us a new field of research which is called embedded network system. In this paper we present a prototype of embedded system for acquisition, storage and retrieval of data from multiple sensors via RS485 bus. In terms of hardware, we have designed and implemented our system with a modular approach to facilitate development and debugging. The development of real-time, proactive and smart embedded monitoring system by use of the technology promises to solve monitoring problems in the industries with large number of machines. For effective affordable solutions in managing sensor data, Embedded Network System provides these immediate solutions. Properly implemented, this system not only helps in sensor data acquisition but also helps in storing and monitoring the real time data of the sensors to maintain the health of the machines.*

Keywords- Cloud controller, I2C protocol, RS485, Sensor Client, Storage server

I. INTRODUCTION

The growth of MSME's (Micro Small and Medium size Enterprises) has revolutionized the economic stability of many developing countries. The machines used by them are very bulky with lot of monitoring sensors which is difficult to monitor. Many a times monitoring may fail and causes huge loss of resources to the owners. The sensors used include monitoring of proximity, alignment, temperature, speed control, over heat detection, time management and many more.

With the growth of embedded technology and computer networks the vision of embedded network systems is coming close to reality, where number of embedded devices can be connected using networking. We propose a design here to monitor all these devices using a single network from one location. The machines must be integrated with advanced monitoring and alerting systems to receive and respond to the information from the sensors. We might need a wired network such as RS485 bus which works as a half-duplex data transfer. One of the fundamental challenge faced during a design of sensor networks is loss and collision of data. So to avoid this we have designed a packet protocol called "HR^2" abbreviated as Hear record respond. In addition, a database is maintained for the supervisor to review the data. For this we interface a keypad and LCD (liquid crystal display) display with menu system for the user to choose the sensor from the options provided. All the data here is stored in an EEPROM chip. . It is like a centralized storage system, which not only stores data but user can also access the data of desired time. Each sensor will have a unique id and will be registered before it starts communicating with the embedded cloud. Supervisor (users) can access the stored sensor data along with date and time.

This project "Embedded Network System" will be useful for many industries which involves continuous monitoring of machines. This project comes up with an idea of monitoring the different sensors on the machines as well as give alerts for precautions to be taken to avoid any problems which helps in saving the time and resources. So by using various equipments this project tries to implement the basic sensor network design for industrial machines considering temperature and humidity as measuring factors in low cost and high reliability.

II. PROJECT OVERVIEW

The system mainly achieves the real-time monitoring in fields by finding temperature of machines, light, humidity, gas detection and saves the collected data in the database of storage server along with time stamp. To achieve good human services, we have designed system solutions in this paper and gave the monitoring site according to the actual situation to visually monitor the current status, retrieve the previous data with the aid of keypad and LCD and facilitate the timely monitoring of the health of each machine. The Cloud Controller here is capable of collecting data from different sensors and send them to the Storage Server and retrieve it back to the Master Client.

The overall project circuit is divided into 4 main parts:

1) Cloud Controller: It is the main unit of the system. It acts as a router, which helps in routing the data within the system. All the communication takes place through this unit. It receives the packets from master client or sensor data acquisition system and provides write or read permission to the network.
2) Storage server: This unit is integrated with real time clock (RTC) and electronically erasable programmable read only memory (EEPROM). Storage server will store the sensor data along with date and time
3) Sensor Clients: It collects data from the sensors and sends them to the cloud controller for storage.
4) Master Client: It is used for retrieving the current and previously stored values. This unit is provided with 4x4 keypad and LCD. 4x4 keypad is used to browse the menu with the help of 16x2 LCD. Sensor data is accessed using its unique ID. An option to choose time of storage is also provided, which will allow the user to browse current data as well as previously stored data.

A. *Objectives*

- To store data of the sensors.
- To overcome the problems faced in the current factory sensor system.
- To access the stored sensor data along with date and time.
- To develop communication packet formats for communication through RS485 bus.
- Establishing a communication between all the peripheral devices.

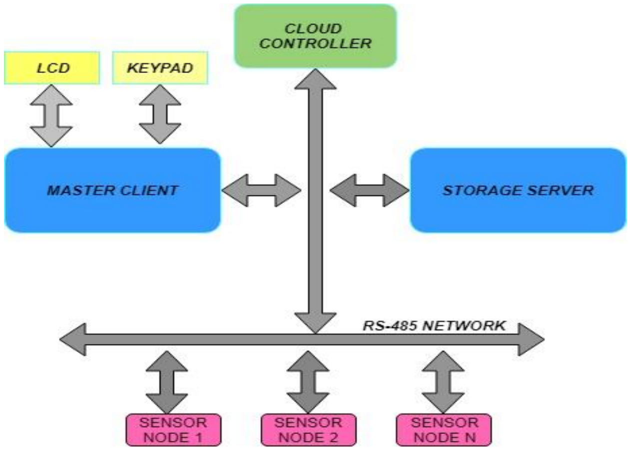

Figure 1: Basic Block Diagram

III. COMPONENTS DETAILS

A. *Microcontroller SST89E516RD:*

The SST89E516RD is an 80C51 microcontroller with 64 kB flash memory and 1024 bytes of data RAM. The key feature is that it consists of an X2 mode operation. The application can be made to run with 80C51 clock rate or the X2 mode to achieve twice the throughput at same clock frequency. Another method to get good performance is by reducing the clock frequency by half, thereby reducing the EMI. The flash memory programming supports parallel and serial programming. Parallel programming offers gang-programming at high speeds, while serial programming (ISP) allows the device to be reprogrammed. This microcontroller is also In-Application Programmable (IAP) device.

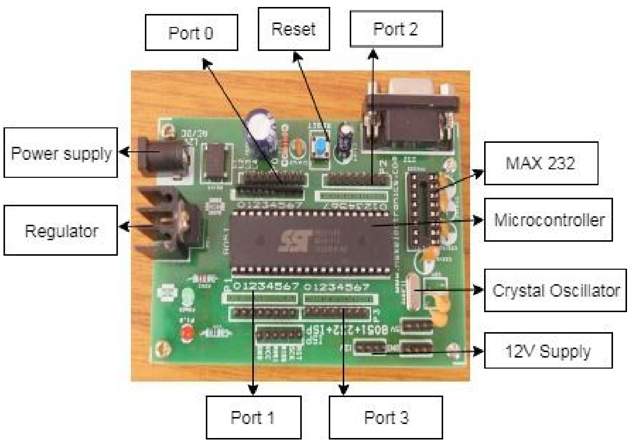

Figure 2: Microcontroller SST89E516RD

B. *I2C Protocol*

I2C stands for a standard Inter-integrated circuit. I2C is appropriate for interfacing to devices on a single board. The I2C-bus is for bidirectional, two-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Data transfer may be initiated only when the bus is not busy. Storage controller will communicate between EEPROM and RTC using I2C protocol.

C. *I2C Based Serial EEPROM (AT24C64)*

An EEPROM is a kind of nonvolatile memory, which means it is used for storing digital data permanently without any power supply. To save the pin count and board space, serial EEPROM will be used in the project. AT24C64 has a memory capacity of 8x8x1024 Bits = 8192 bytes = 8 KB of data storage, 32bytes per page and 3 programmable address. So, eight of such devices can be connected to a common bus. The EEPROM is a part of storage server module and will be used to store the sensor data along with sensor id, day no, HH:MM

In EEPROM Sensor data is stored as:

Sensor id	Data1	Data2	Day No	Hour	Min

Figure 3: EEPROM sensor data format

D. *I2C based serial RTC DS1307*

A real-time clock (RTC) is an integrated circuit that keeps track of the current time. It automatically updates time in seconds, minutes, hours, days, months, years without any intervention, it even take cares of leap year corrections. It is used in the storage server of the project, to keep update of time. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 operates as a slave device on the I2C bus. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply.

E. *Other Components Of The Project*

1) Hardware components:

- MCUP89V51RD2 NXP
- MAX485 (RS485 - TTL Convertor and vice versa)
- AT24C64 I^2C Serial EEPROM From Atmel
- DS1307 Real Time Clock
- 4x4 keypad matrix
- 16X2 LCD
- ADC0804 SAR based ADC
- Temperature sensor (LM35)
- Light Depend Resistor (LDR)

2) Software components:

- Keil µVision3 to write, compile, debug and simulate embedded code.
- HyperTerminal to interface to serial port and test various parts of project
- HERCULES version 3.2.3 software from HW for serial packet interface

IV. RS485 STANDARD INTERFACE

The Electronics Industry Association (EIA) has produced standards for RS485, RS422, RS232, and RS423 that deal with data communications. EIA standards where previously marked with the prefix "RS" to indicate Recommended Standard. EIA-485 or RS-485 is a standard defining the electrical characteristics of drivers and receivers for use in balanced digital multipoint systems. These characteristics make such networks useful in industrial environments and similar applications. RS485 meets the requirements for a truly multi-point communications network, and the standard specifies up to 32 drivers and 32 receivers on a single (2-wire) bus. Due its twisted pair characteristics, noise will be reduced.

All the modules in the project: master cloud controller, storage servers, sensor clients, master client will communicate via RS-485 network only. Features of RS-485 are listed in table 1.

The RS-485 differential line consists of two pins:

- A - non-inverting pin
- B - inverting pin

An optional, third pin may be present which is SC or G - reference pin.

The RS-485 standard states:

For a MARK (logic 1), the driver's A terminal is negative relative to the B terminal.

For a SPACE (logic 0), the driver's A terminal is positive relative to the B terminal.

Table 1: Features of RS485

Standard	EIA RS-485
Physical Media	Twisted pair
Network Topology	Point-to-point, Multi-dropped, Multi-point
Maximum Devices	32 drivers or receivers
Maximum Distance	1200 meters (4000 feet)
Mode of Operation	Half duplex
Maximum Baud Rate	100 kbit/s - 10 Mbit/s
Voltage Levels	-7 V to +12 V

MAX485 transceiver is used for applications where many devices communicate with each other using RS485 physical layer. Drivers are short-circuit current limited and are protected against excessive power dissipation by thermal shutdown circuitry that places the driver outputs into a high-impedance state. The receiver input has a fail-safe feature that guarantees a logic-high output if the input is open circuit. It can transmit and receive at data rates up to 2.5Mbps Driver Enable (DE) and Receiver Enable (RE) pins are included when disabled, the driver and receiver outputs are high impedance. Have standard 12kΩ Receiver Input impedance. The MZX485 interface is shown in the figure.

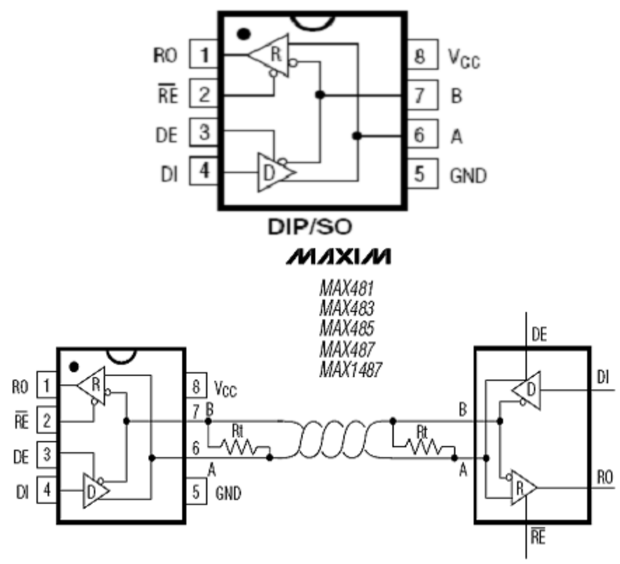

Figure 4: Pins Diagram of MAX485

Figure 4 shows the pin diagram of MAX485. Pin description is given below:

- RO Receiver Output: If A > B by 200mV, RO will be high; If A < B by 200mV, RO will be low.
- RE Receiver Output Enable. RO is enabled when RE is low; RO is high impedance when RE is high.
- DE Driver Output Enable. The driver outputs, A and B, are enabled by bringing DE high. They are high impedance when DE is low. If the driver outputs are enabled, the parts function as line drivers. While they are high impedance, they function as line receivers if RE is low.
- DI Driver Input. A low on DI forces output A low and output B high. Similarly, a high on DI forces output A high and output B low.

RS485 adopts differential signal negative logic. +0.2 V - +6 V is indicated as '0' and -6 V to -0.2 is indicated as '1'. There are two-wire and four-wire connection modes. In CAMABIO system, two-wire is used. This connection is of bus topology structure, in which a maximum 256 nodes can be connected to the same bus. In most circumstances, when connecting RS485 communication links, only a twisted pair is used to connect the 'A' and 'B' terminals of each interface. In theory, the maximum communication distance of RS485 can extend up to 1,200 meters. Logic '0' and '1' are indicated by the voltage differences between the two wires +0.2-6 V and -0.2-6 V respectively. The interface signal level is lower than that of RS-232-C, so the interface circuit chip is not easily damaged. Also the level is compatible with TTL level, which facilitates the connection with a TTL circuit. Maximum data transmission speed is 10 Mbps. The RS485 interface adopts the role of balancing driver and differential receiver, so that the common mode disturbance immunity is improved, i.e. it has good noise immunity. Maximum 256 transceivers are allowed to connect to the bus. Normally only two wires, a shielded twisted pair (STP), is needed in the half duplex network formed by the RS485 interface.

V. IMPLEMENTATION

A. *Master Cloud Controller(MCC)*

A network is designed using RS485 bus which uses 2 wire differential signal transmission for communication. Using MAX485, a multi-drop system is implemented via which many nodes were connected to RS485, but the care must be taken such that multiple nodes should not communicate simultaneously in order to avoid packet collision. To ensure avoidance of collision, a master cloud controller was designed which will control complete bus and ensure collision free communication among the system.

Operations performed by MCC

- The whole network will be controlled by MCC

- It will get the data to be stored from sensor client
- The requests from master client will be received by MCC
- It will also communicate with storage server to store the sensor data, to retrieve stored data, to get current time and to update time.

The flowchart of MCC is shown in figure 4.

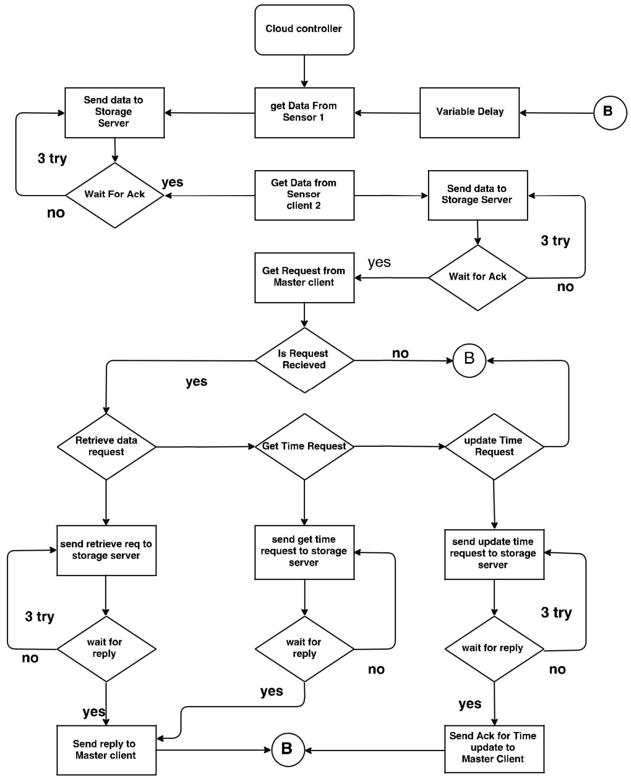

Figure 5: Flowchart of MCC

B. *Storage Server*

The storage system consists of following components:

- Storage controller (P89V51RD2)
- EEPROM (AT24C64)
- Real Time Clock (DS1307)

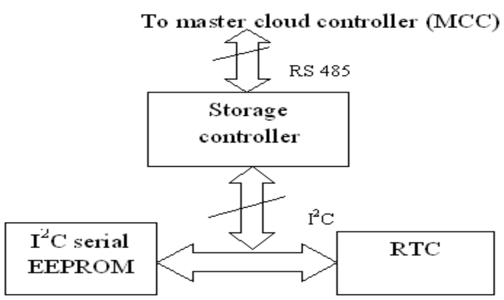

Figure 6: Block diagram of storage server

Operations performed by storage servers:

- The main operation is to stores the sensor client data at given time to the EEPROM
- It will send the stored data to the master cloud controller whenever there is any retrieve request
- User can get the current time
- User can update the time

Storage server stores the sensor data along with sensor id, Day No, HH: MM. Sensor id will be unique for each sensor.

The flowchart if storage server is shown in figure 6.

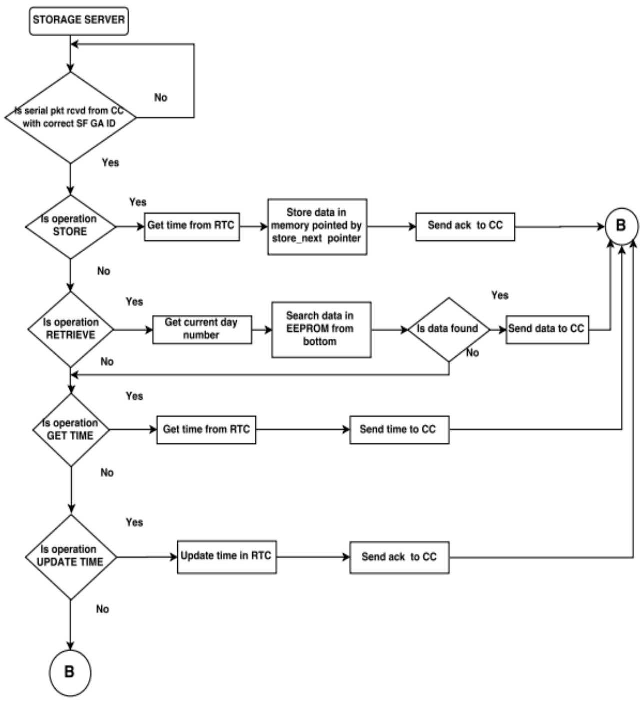

Figure 7: Flowchart of storage server

C. *Packet Protocol Used For Communication Between Peripherals:*

Master cloud controller controls all the clients in the network. All the communication in the network is done with the help of packets. Proprietary packet communication protocol have designed, which uses RS485 bus for communication. Network has distinct Start of frame named as SF and End of frame named as EF.

Packet is sent as a multiple byte grouped together. Here all important information is embedded or sandwiched between SF and EF.

Cloud controller sends packets to various systems and gets response from them. The general packet format is shown in figure 8.

SF	Group selection	Destination Address	Operation if required	Data-1	Data-n	EF

Figure 8: General packet format

For easier communication, storage servers, master client, sensor client modules are grouped separately and a hex address will be assigned to each group. This will save the time, as and when packet is received, initially group address will be checked and later destination address.

Group Selection: Each group in the system (storage server, master client and sensor clients) may have number of modules which will be assigned with unique address.

Destination Address: The address where packet has to be delivered.

Operation: It provides the information about the operation to be performed like to request data, time and to update time.

Data 1 to Data n: It carries the averaged sensor data value from sensor client system to embedded cloud. It will be in floating point number format.

VI. CONCLUSION

The system implemented is similar to cloud storage using hardware chips and embedded system. First task was to design the storage servers as centralized storage system, where sensor data will be stored safely along with date and time. Then design of master client, which decreases the number of LCD and Keypad, used in the conventional system and will ease the work of supervisor to monitor the data. For communication between the modules a network was developed using RS-485 bus. A proprietary packet communication protocol was also developed for the system

There is a chance of collision between data packets of multiple servers and client system and data may be lost. Therefore master cloud controller was designed, which will control the complete bus. Also two sensor clients (temperature and light sensors) were developed to check the working of embedded cloud.

The developed system can be used for a factory sensor system which reduces the hardware expenditure and work of the supervisor. The data stored will be safe in embedded cloud and can be easily accessed by the user.

VII. ACKNOWEDGMENT

Presenting a paper is a golden opportunity for learning and self-development. The successful completion of any project would be incomplete without the mention of people who made it possible and whose constant encouragement and guidance has been a source of inspiration. We would like to thank **Dr.N.R.Shetty**, Director, NMIT, for providing such a wonderful environment to make this project a fruitful one. We would like to express our gratitude to **Dr.H.C.Nagaraj**, principal of NMIT, for his support in bringing this project to completion. We wish to express our gratitude to the Head of the department, **Dr.Sandya S**, Dept. of ECE for providing a congenial working environment.

It gives us a great pleasure to express our gratitude to our project guide **Mrs. Divya Muralidhar**, Associate Professor, Electronics and Communication Engineering Department, Nitte Meenakshi Institute of Technology, Bengaluru for her suggestions which helped us to complete the project. We would also like to acknowledge and thank our **Parents and Friends** who have been a source of inspiration in successful completion of this project.

REFERENCES

[1] Network Sensor of Temperature & Humidity based on RS485 Interface by Lv Zhigang & Wang Peng.

[2] Design of the Multi-computer Communication System Based on RS485 by Shihong Qin & Chen Kai, IEEE.

[3] "RS-485 serial port pseudo-full-duplex communication research and application", by Su Xunwen and Wang Shaoping, Prognostics and Health Management Conference, 2010.

[4] Volume-7, Issue-2, March-April 2017International Journal of Engineering and Management Research

[5] Dual Protocol Transceivers Ease the Design of Industrial Interfaces by inersil.

[6] Design and implementation of remote terminal unit for monitoring weather using raspberry pi and RS485 standard interface by Rathod Rohan Kumar, IJSRD.

[7] http://tscm.com/rs-485.pdf