Microcontroller based Modbus Protocol Converter using PIC 17C756

Sangeetha.V, Vinothini.V, Suganya.R, Rajaprabha.R and Bhagavathi Sivakumar.C

Abstract: Our paper is aimed at developing an industrial oriented Embedded Serial Data Interpreter for protocol conversion. At present, many industrial establishments face the problem of protocol compatibility with the external monitoring devices. This problem has been solved by my paper. We have successfully deployed it for monitoring ESP controllers manufactured by BHEL Ranipet. The Electrostatic Precipitator (ESP) controllers manufactured by BHEL are using dedicated protocol for communicating with each other. Hence they cannot be directly linked with the Distributed Control System (DCS) panel, which uses universal Modbus protocol for communication with all the power system components to monitor its operation. Hence, We have interfaced the ESP controllers with DCS via Serial Data Interpreter.In order to test the functionality of the paper, a simulation has been performed in BHEL simulation lab. The simulation results agreed with the predicted results confirming the successful operation of the Interpreter.

I. INTRODUCTION

This technical note will describe about the setup and configuration of MICRO CONTROLLER BASED MODBUS PROTOCOL CONVERTER, using PIC17C756 for the conversion of BHEL's dedicated protocol into universal Modbus protocol for interfacing ESP controllers with DCS via Serial Data Interpreter.Serial Data Interpreter transfers the data to virtually any communication enabled devices, using RS-232C and RS-485 serial communication interfaces. i.e. RS-232C interface from ESP controllers to Serial Data Interpreter and by RS-485 interface from Serial Data Interpreter to DCS. This report describes how the PIC handles the interrupts of controllers efficiently and transfers the data to DCS with the help of universal Modbus protocol. The development of firmware for PIC, its deployment using PIC tools and its simulation is also described in this document.

The system has been designed using VLSI technique and entire integration of components is done in BHEL, Ranipet. In order to test the functionality of the system, a simulation has been performed in the laboratory. The results are quite satisfactory agreeing with predicted results.

Sangeetha.V, Vinothini.V, Suganya.R, Rajaprabha.R and BhagavathiSivakumar.C are with Sri Shakthi Institute of Engineering and Technology, Coimbatore. sangeethasscet@gmail.com, vinobharathivlsi@gmail.com

1.1 PROBLEM BACKGROUND

In BHEL there are two controllers which are used to monitor and control the working of Electrostatic precipitator. These controllers work on their dedicated protocol.At present the status of these controllers cannot be viewed at the DCS. Any problem in these controllers can be identified and corrected only at the workplace.

1.2 NEED FOR PROTOCOL CONVERTER

Thus in BHEL they are in need of a protocol converter for two main reasons.

- To migrate from an old system to a new one.
- To monitor these controllers remotely by converting the dedicated protocol to Modbus protocol used at the DCS.

We have selected Microchip product for easy adaptability in various applications.

1.3 BHEL CONTROLLERS

BHEL's advanced precipitator controller and Rapper controller are the two controllers used to control the ESP. These two microcontrollers control the rapper drives, temperature level, oil level etc. of the ESP.

2. SERIAL INTERFACE

2.1 INTRODUCTION

Serial interfaces are cables used for serial data communication between DCE (Data Communications Equipment) and DTE (Data Terminal Equipment). Data Communications Equipment is devices such as modem, TA adapter, plotter etc. while Data Terminal Equipment is Computer. Serial interface coordinates the flow of data, control signals and timing information between the DTE and DCE. Electronic Industries Association (EIA), in an effort to standardize interface equipment between DTE and DCE, agreed on a set of standards called the RS-232 specifications. RS-232C, the third revision was introduced in1969 and was compatible with the later version RS-232D introduced in1987.

RS-232 specifications identify the mechanical, electrical, functional and procedural description for the interface between DTE and DCE. RS-232 has a maximum speed of 20 kbps for a cable length of 12 meters. **RS232** is the only interface capable of full duplex communication. RS-485 is most recent version introduced by EIA has a data rate of 35 Mbps for a cable length of 12 meters. **RS485** is the only interface capable of internetworking multiple transmitters and receivers in the same network. **RS485** is now popular with computers, PLCs, micro controllers and intelligent sensors in scientific and technical applications. RS-485 cables are designed with a combination of twisting and shielding as

STP, *shielded twisted pair* and **FTP**, *foiled twisted pair* networking cables for high noise immunity. With **RS485** communication distances of 1200 m are possible.

2.2 RS-232C INTERFACE

RS-232C is a single ended communication cable with 25 pin (DB25-DB25) or (DB25-DB9) compatible connector. The terminal load capacitance of the cable is specified as 2500 pF, which includes cable capacitance. The impedance of the terminating end must be between 3000 Ω and 7000 Ω and the output impedance is specified greater than 300 Ω.

This means circuits powered by 5 VDC are capable of driving RS232C circuits directly. The RS-232 interface presupposes a common ground between the DTE and DCE.

The output signal level usually swings between +12V and - 12V. The "dead area" between +3v and -3v is designed to absorb line noise. Many receivers designed for RS-232 are sensitive to differentials of 1v or less. This can cause problems when using pin powered widgets - line drivers, converters, modems etc. These type of units need enough voltage & current to power them self's up.Typical UART (the RS-232 I/O chip) allows up to 50ma per output pin - so if the device needs 70ma to run we would need to use at least 2 pins for power. Some devices are very efficient and only require one pin (Transmit or DTR pin) to be high - in the "SPACE" state while idle. An RS-232 port can supply only limited power to another device. The number of output lines, the type of interface driver IC, and the state of the output lines are important considerations.

Table 1 : Pin Details

RS232-C	Description	Circuit EIA
1	Shield Ground	AA
7	Signal Ground	AB
2	Transmitted Data	BA
3	Received Data	BB
4	Request To Send	CA
5	Clear To Send	CB
6	DCE Ready	CC
20	DTE Ready	CD
22	Ring Indicator	CE
8	Received Line Signal Detector	CF
23	Data Signal Rate Select (DTE/DCE Source>	CH/CI
24	Transmit Signal Element Timing (DTE Source)	DA
15	Transmitter Signal Element Timing (DCE Source)	DB
17	Receiver Signal Element Timing (DCE Source)	DD
18	Local Loopback / Quality Detector	LL
21	Remote Loopback	RL/CG
14	Secondary Transmitted Data	SBA
16	Secondary Received Data	SBB
19	Secondary Request To Send	SCA
13	Secondary Clear To Send	SCB
12	Secondary Received Line Signal Detector/ Data signal Rate Select (DCE Source)	SCF/CI
25	Test Mode	TM
9	Reserved for Testing	
10	Reserved for Testing	
11	Unassigned	

Data is transmitted and received on pins 2 and 3 respectively. Data Set Ready (DSR) is an indication from the Data Set (i.e., modem) that it is on. Similarly, DTR indicates to the Data Set that the DTE is on. Data Carrier Detect (DCD) indicates that a good carrier is being received from the remote modem. Pins 4 RTS (Request To Send - from the transmitting computer) and 5 CTS (Clear To Send - from the Data set) are used to control. In most Asynchronous situations, RTS and CTS are constantly on throughout the communication session. However where the DTE is connected to a multipoint line, RTS is used to turn carrier on the modem on and off. On a multipoint line, it's imperative that only one station is transmitting at a time. When a station wants to transmit, it raises RTS. The modem or UART turns on carrier, typically waits a few milliseconds for carrier to stabilize, and then raises CTS. The DTE transmits when it sees CTS up. When the station has finished its transmission, it drops RTS and the modem drops CTS and carrier together. Clock signals (pins 15, 17, & 24) are only used for synchronous communications. The modem or UART extracts the clock from the data stream and provides a steady clock signal to the DTE. Note that the transmit and receive clock signals do not have to be the same, or even at the same baud rate. All signals are measured in reference to a common ground, which is called the signal ground (AB). A positive voltage between 3 and 15 Vdc represents a logical 0 and a negative voltage between 3 and 15 Vdc represents a logical 1.This switching between positive and negative is called bipolar. The zero state is not defined in RS232 and is considered a fault condition (this happens when a device is turned off).

3. INTRODUCTION TO MODBUS PROTOCOL CONVERTER

Modicon programmable controllers can communicate with each other and with other devices over a variety of networks. Supported networks include the Modicon Modbus and Modbus Plus industrial networks. Networks are accessed by built–in ports in the controllers or by network adapters, option modules, and gateways that are available from Modicon. The common language used by all Modicon controllers is the Modbus protocol. This protocol defines a message structure that controllers will recognize and use, regardless of the type of networks over which they communicate. It describes the process a controller uses to request access to another device, how it will respond to requests from the other devices, and how errors will be detected and reported. It establishes a common format for the layout and contents of message fields. The Modbus protocol provides the internal standard that the Modicon controllers use for parsing messages. During communications on a Modbus network, the protocol determines how each controller will know its device address, recognize a message addressed to it, determine the kind of action to be taken, and extract any data or other information contained in the message. If a reply is required, the controller will construct the reply message and send it using Modbus protocol.

3.1 COMMUNICATION ON MODBUS NETWORKS

Standard Modbus ports on Modicon controllers use an RS–232C compatible serial interface that defines connector pin outs, cabling, signal levels, transmission baud rates, and parity checking. Controllers can be networked directly or via modems.

Controllers communicate using a master–slave technique, in which only one device (the master) can initiate transactions (called ‘queries’). The other devices (the slaves) respond by supplying the requested data to the master, or by taking the action requested in the query. Typical master devices include host processors and programming panels. Typical slaves include programmable controllers. The master can address individual slaves, or can initiate a broadcast message to all slaves. Slaves return a message (called a ‘response’) to queries that are addressed to them individually. Responses are not returned to broadcast queries from the master.

The Modbus protocol establishes the format for the master’s query by placing in to it the device (or broadcast) address, a function code defining the requested action, any data to be sent, and an error–checking field. The slave’s response message is also constructed using Modbus protocol. It contains fields confirming the action taken, any data to be returned, and an error–checking field. If an error occurred in receipt of the message, or if the slave is unable to perform the requested action, the slave will construct an error message and send it as its response.

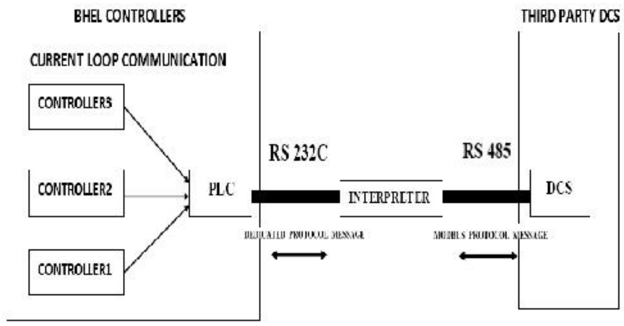


Figure 1: Block Diagram of MODBUS Networks

The speed of the differential interface **RS485** is far superior to the single ended versions. This has been done to avoid reflections of signals. The slew rate of RS-485 is indefinite, which means we have maximum communication speed on the line. To avoid reflections on longer cables it is necessary to use appropriate termination resistors.

RS485 is used in situations with a severe ground level shift of several volts, where at the same time high bit rates are possible because the transition between logical **0** and logical **1** is only a few hundred millivolts. **RS485** receivers with an input resistance of 12 kΩ it is possible to connect 32 devices to the network. Currently available high-resistance **RS485** inputs allow this number to be expanded to 256. **RS485** repeaters are also available which make it possible to increase the number of nodes to several thousands, spanning multiple kilometers. For higher speeds and longer lines, the termination resistances are necessary on both ends of the line to eliminate reflections we can use 100 Ω resistors on both ends.How does **RS485** function in practice? Default, all the senders on the **RS485** bus are in tri-state with high impedance. In higher level protocols, one of the nodes is defined as a master which sends queries or commands over the **RS485** bus. All other nodes receive these data. Depending of the information in the sent data, zero or more nodes on the line respond to the master. In this situation, bandwidth can be used for almost 100%. There are other implementations of **RS485** networks where every node can start a data session on its own. Because there is a chance of data collision with this implementation, theory tells us that in this case only 37% of the bandwidth will be effectively used. With such an implementation of a **RS485** network it is necessary that there is error detection implemented in the higher level protocol to detect the data corruption and resend the information at a later time.

There is no need for the senders to explicitly turn the RS485 driver on or off. RS485 drivers automatically return to their high impedance tri-state within a few microseconds after the data has been sent. Therefore it is not needed to have delays between the data packets on the RS485 bus. **RS485** is used as the electrical layer for many well known interface standards, including Profibus and Modbus.

4. CURRENT LOOP

Current loop interface offer the most cost effective approach to long distance noise immune data transmission. Current loop is a signaling technique that has been in existence for many years - actually going back to the earliest teletype printers. The 20mA current loop is suitable for distance to 2000 feet at data rates up to 19.2k baud with careful attention to interfacedesign.Itcan be used at longer distances when data rates are as low as 300 baud. The fundamental elements of a 20mA current loop are a current source, a current switch and a current detector. The transmitter is the current switch and the receiver is the current detector. The interface that contains the current source is called the active unit and all other units are referred to as passive units. In a20mA loop

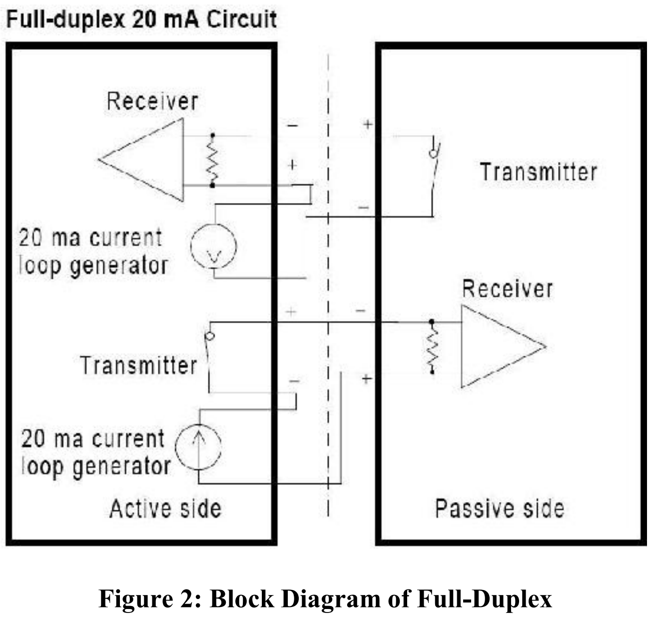


Figure 2: Block Diagram of Full-Duplex

the current flows when the loop is idle (no data being transmitted). In current loop, the data is represented by electrical current being sent in either one direction around a loop or the other direction. It has significant advantages when it comes to distance. Data sent by a current loop connection can go much further than data sent by connecting the common RS-232C interfaces. This signaling technique also provides much greater protection against electrical interference than the common RS-232C. It is particularly suited to protecting data communication being carried out in environments where the terminating equipment has high voltage sources. Here there are many deleterious effects which current loop signaling can ameliorate. Finally, many current loop systems have the benefits of optical isolation and the protection that this affords from the inaccuracies of ground loops.

4.1 SERIAL INTERFACE DRIVERS

IC drivers are electronic circuits on a chip that provide the required current and voltage needed to turn power switching elements on or off, based on the logic output signals of a digital signal processor (DSP), microcontroller or other logic device. They protect against over and under voltage as well as thermal shutdown. Typically, IC drivers are used with power switching elements such as metal oxide semiconductor field effect transistors (MOSFET) and insulated gate bipolar transistor (IGBT). There are four basic types of gate drivers: high-side, low-side, dual or half-bridge, and three-phase. A high-side gate driver is used for components connected to a positive supply. A low-side gate driver regulates components connected to a negative supply. Dual gate or half-bridge drivers have properties for both high-side and low-side gates.

IC drivers differ in terms of specifications, applications and features. Most gate drivers have 1, 2, or 4 output channels. The output voltage can be inverted or non-inverted. Gate driver performance specifications include output voltage and current, rise time and fall time, power dissipation, operating temperature and switching frequency. Rise time is defined as the time required for output voltage to reach 90% of maximum. Fall time is the time required for voltage to decline to 10% of maximum. In terms of applications, IC drivers can be used as transformers, discrete transistors, or dedicated ICs. An IGBT gate driver functions as an isolation amplifier and can provide short-circuit protection. The proper selection of a MOSFET driver depends on whether the device uses a positive or negative supply.

The RS-232C specification states that the required driver output voltage is defined as being between +5V and +15V and is positive for a logic “0” (+5V to +15V) and negative for a logic “1” (−5V to −15V). These voltage levels are defined when driver is loaded ($3000\Omega < R_L < 7000\Omega$). In applications where strict compliance to RS-232C voltage levels is not essential, a ±5V power supply to the driver may be used. The RS-232C specification further states that, during transitions, the driver output slew rate must not exceed 30V/µs. The common-mode output range for RS-485 drivers is -7V to +12V. The minimum receiver-input resistance is12kΩ for RS-485 drivers. It determines the kind of action to be taken, and extract any data or other information contained in the message. If a reply is required, the controller will construct the reply message and send it using Modbus protocol.

It determines the kind of action to be taken, and extract any data or other information contained in the message. If a reply is required, the controller will construct the reply message and send it using Modbus protocol.

5. CONCLUSION

In this paper, we developed an industrial oriented embedded serial data interpreter for protocol conversion. This system has been designed and simulated in order to obtain the predicted results. In this method, we reduced the power compared to DCS.

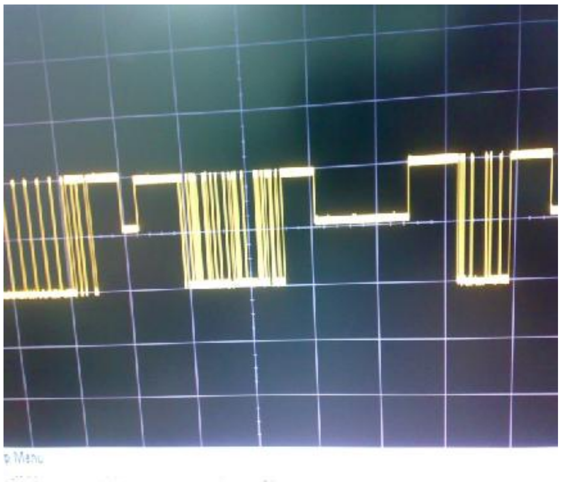
5.1 FUTURE ENHANCEMENTS

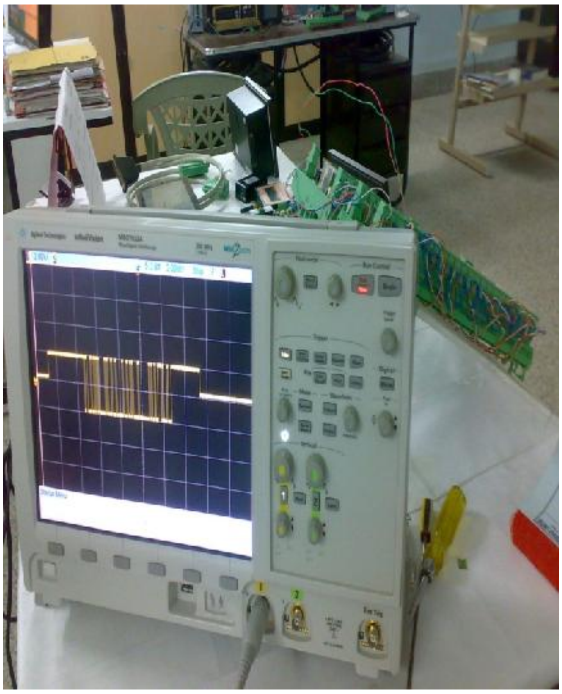
There are number interpreter into a more advanced one with the more advanced controllers and power components of opportunities for developing this protocol convertor.

Some of the future advancements that can be in near future are

- Communications can be established through optical fibres for more accurate, faster and error free connection.
- It can also be wireless i.e. a simple transmitter and receiver arrangements for remote data access.
- It can be programmed with more generalized code for accommodating any type of controller program.
- It can be programmed in such a way that a plant supervisor can carry it on the field and access any controller data (Note: All controllers given a specific code) when the code on the program matches the controller code and it sends the data to DCS and in turn DCS sends the data to the supervisor wireless on the display panel fitted on top of Interpreter.

PHOTOGRAPH





REFERENCES

1.Muhammad Ali Mazidi, Janice Gillispie Mazidi (2000) 'The 8051 Microcontroller and Embedded Systems', First Impression, New Delhi, Pearson Education.

2.Wayne Tomasi (2002) 'Advanced Electronic Communications Systems', Fifth edition, New Delhi, Prentice-Hall of India Private Limited.

3.Lakshmi Bhadrachalam, Suresh Chalasani, Senior Member, IEEE and Rajendra V. Boppana, Senior Member, IEEE.

4.Stephen A. Weis1, Sanjay E. Sarma2, Ronald L. Rivest1 and Daniel W. Engels Laboratory for Computer Science Auto-ID Center Massachusetts Institute of Technology Cambridge, MA 02139, USA.

5.Jahariah Sampe1, Khairul Parman Zakarial, Fazida Hanim, Masuri Othman University Kuala Lumpur Malaysia France Institute(UNIKL MFI), Section, Jalan Teras Jernang, 43650 Bangi, Selangor, Malaysia Faculty of Engineering Electronic and System, University Kebangsaan Malaysia, 43600 Bangi Selangor, Malaysia.

6. www.bb-elec.com
7. www.microchip.com
8. www.modicon.com

Sangeetha.v did her Bachelor of Engineering in Electronics and Communication Engineering at Sri Subramanya College of Engineering and Technology, Palani and doing Masters of Engineering in VLSI Design at Sri Shakthi Institute of Engineering and Technology, Coimbatore, India. .Her research interests include Body Area Network, VLSI Design.

V.Vinothini did her Bachelor of Engineering in Electronics and Communication Engineering at Latha mathavan Engineering College, Madurai and doing Masters of Engineering in VLSI Design at Sri Shakthi Institute of Engineering and Technology, Coimbatore, India. .Her research interests include Body Genetic Algorithm, VLSI Design.