International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 8
August 2017

Academic Science

MODBUS Protocol Implementation for Microcontroller (cortex M3)

Devanshi N. Patel
MITCOE, Pune
Prof. Sunil B. Somani
MITCOE, Pune

ABSTRACT: -

With the evolution from the distributed Communicationto centralized communication and now need for the busfield communication has come into existence. So the system communicating with the various devices at one time for the data acquisition is needed. This prove to be reliable and have a fast communication system. The paper presents the design of the system that can implement a master-slave communication. The communication is not restricted to the particular network devices only. This system can have Bi-directional communication. Master asks for the status or results from slave and in return slave responds to the request and sends its result's which can control the further process of communication. The system got Cortex controller (Cortex M3), 32-bits ARM processor that provides speed to the system. This system implements MODBUS protocol, an application layer protocol for the communication between master and slave through RS485 for communication interface between devices. Here RTU mode of communication is used in MODBUS. In this paper RTU mode is implemented as it is faster than the ASCII mode.

Keywords: MODBUS Communication protocol, Cortex M3, RS485, Data acquisition.

1 INTRODUCTION

In the design of the reliable system for the data acquisition, the first thing needs to be done is to select the communication protocol and the mode of communication that is to be used to communicate amongst the devices and then to build a architecture that can support such system. The problem of distant, dispersed communication with the devices has been removed in this system. Busfield communication that is communicating through address is used to control and access the data for the data acquisition system.

The working of system starts with two modes: 1.Normal 2.MODBUS. In normal mode communication through RS485 with peripherals is done by LPC1768. It waits for the request from master, if there is any request it enters MODBUS mode or stays in normal mode. Master on requirement demands the slave to update the status or carry out the operation it is supposed to do. For communication with the slave, address is required and can have 247 slaves at a time. So by giving the slave address it can either communicate in unicast or broadcast mode. If in the broadcast mode, every slave receives the command from the master but the one with the matching address responds to it. But again there lies the check that needs to be performed. Master sends the frame consisting of the fields like Address, Function code, Address to write, Data to write and CRC. When slave receives the frame, first it checks for the address then if it matches it goes to the other field and checks for the data. Function codes are for read and write coils and registers.

On the slave side, the CRC check needs to be performed. CRC check is performed to confirm that the frame received is correct and the fields of the frame are correct. In the procedure of the CRC check, the CRC has to be separated which is sent by the master and then apply the procedure of the CRC and check if the CRC matches with the received CRC. If it does then further functionality is carried out and the results are forwarded to the master in the same format as received from the master.

The system discussed above is implemented in Fig.1 which shows that one workstation there can be many masters and masters can communicate with slave of other devices too. Slave can either be the other whole

system or the PC. This system is designed for the generalized application, devices can communicate with any device connected by just setting the configuration, setting flags, information is already stored, one has to read it and set configuration to communicate with the device.

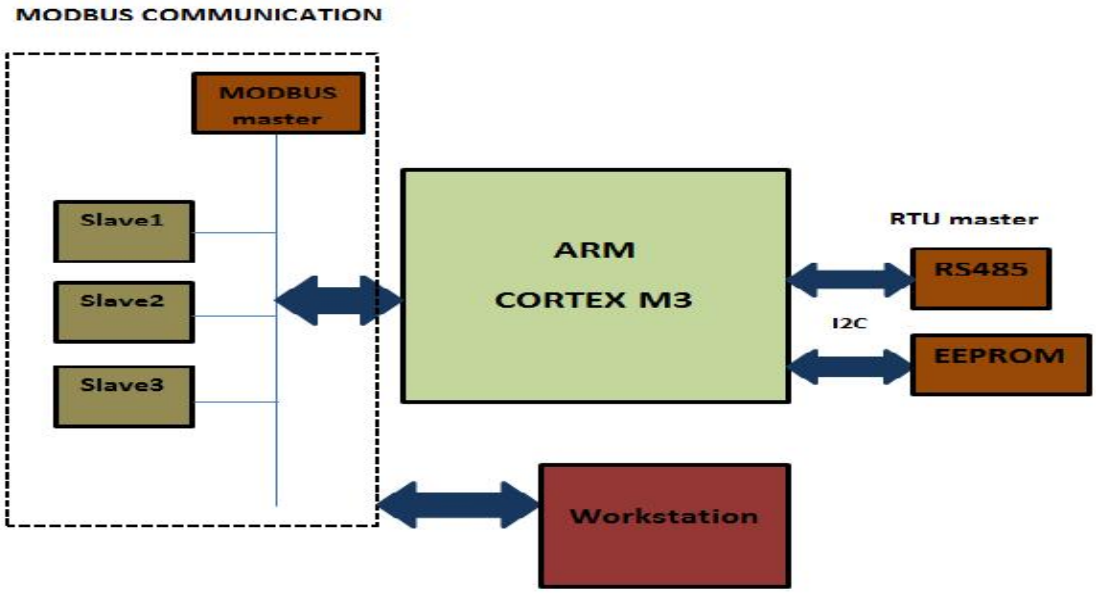

Fig 1: Block diagram of the overall system

After all the devices are available communication then an application layer messaging protocol needs to be implemented. So at application layer MODBUS is implemented and at the Physical layer RS485 is there.

2 DESIGN OF THE SYSTEM

To set up application based on the MODBUS communication,it is divided into mainly two parts.

Master and Slave.

1. System is started and all the parameters that are needed to be initialized are initialized. All the peripherals that are dependent or the values that needed to be evaluated all are set either as input or output.That means UART,Timer,Counter and other peripherals must be initialized.
2. Now the system works in two modes 1.Normal,2.MODBUS.It checks for the interrupt in any case.If no interrupt is their it works in normal mode and when the interrupt is received it enters into MODBUS mode.As RS-485 is used,32 loads at one time can be addressed. So the peripherals that interrupted the program need to be served. Here the master slave communication takes place.
3. If no interrupt is there it will wait till any interrupt is there.If they got the interrupt it willcheck for the frame fields like Address of the slave and the CRC check and serve theinterrupt.It checks if these field matches with the slave address and the CRC generatedby the slave.If it does it will process further fields.
4. It will read the information in interrupt sent by the master.That is the frame consistsof the fields like slaveaddress,function code,address,bytes will be received one by oneand after matching the field of slave address first and then the CRC.
5. Then the channel will be selected for serving particular interrupt.That is to select theUART port and to set it as transmitter or receiver.
6. Then the mode is set whether to read or write the data from the slave is to decided.
7. Data is collected after performing the operations on slave.
8. The operated data is stored for the further use by the processor or the other slave.
9. After serving to that slave again program waits for the interrupt.

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 8
August 2017

Academic Science

I HARDWARE FOR ACCQUISITION SYSTEM

Communication system adopts RS485 bus as transmission lines which transfers data collected by data acquisition to PC and implements corresponding monitoring in accordance with Modbus communication protocol. Cortex M3 is connected to PC by using RS485 link which based on MAX3485. The RXD and TXD of UART in the cortex M3 are respectively connected to MAX3485's RO and DI to send and receive data; PI of cortex M3 in the I/O mode connecting to MAX3485's receiving enable pin RE and sending enable pin DE, when PT is low, cortex M3 receives data from the bus while PI is high, controller will send data to the bus. RS485 bus is a differential signal transmission, but for the reliability, three transmission lines is used which are 485_A, 485_B and GND.[1]Data which is received or obtained can be displayed on the LCD, to make it convenient for further process and modify it and also can be transferred to the center workstation through RS485.

II COMPOSITION OF MODBUS PROTOCOL

MODBUS is an application layer protocol which is responsible for the transfer of the packet between master and slave. Each device got device address which is unique on the same network, at a time only 1 master device and multiple slaves are allowed. Master device can communicate with slave device using unipoint communication, which also can communicate in broadcast with all slave devices. Communication will only start when the masters requests for the data, else slave can't start the communication. Modbus protocol got two modes of transmission: ASCII and RTU mode. In this the RTU mode is used. What is necessary is to form the PDU which consists of address field, data field, CRC check and function code. Address can go upto 247 devices as it can communicate to 247 devices at one time. Upto 125 can be used and other are reserved. Function code is used for the function to be performed by the slave of write or read from the master.

Table 1. MODBUS Registers

CODE	FUNCTION
01(01H)	Read coil
02(02H)	Read Input coil
03(03H)	Read Multiple holding Registers
04(04H)	Read Input Registers
05(05H)	Write Single Coil (Output)
06(06H)	Write binary to Single Register
15(0FH)	Force Multiple Coils (Outputs)
16(10H)	Preset Multiple Registers

Data field is the data that wants to transmit it can be n*bits. CRC check is used to check the valid slave address. The stored CRC is checked with the CRC sent from the master.

3 MODBUS IMPLEMENTATION

To start the transmission process request must be sent from master. After that when the interrupt is generated it goes to slave side. The units receive data from the host by serial port interrupting mode byte by byte, and go in a serial port interrupt byte by byte. The system hosts are constantly detecting the network bus. When the first data is received (address field), each slave device checks if it’s their address. If it matches, the slave devices continue to receive other data, if not, they exit the interrupt. After receiving the message, the CRC check is to be performed. If the CRC result matches with the CRC of sent from the master, the communication is normal and the function code calls the data acquisition subroutine, then returns the response frame.

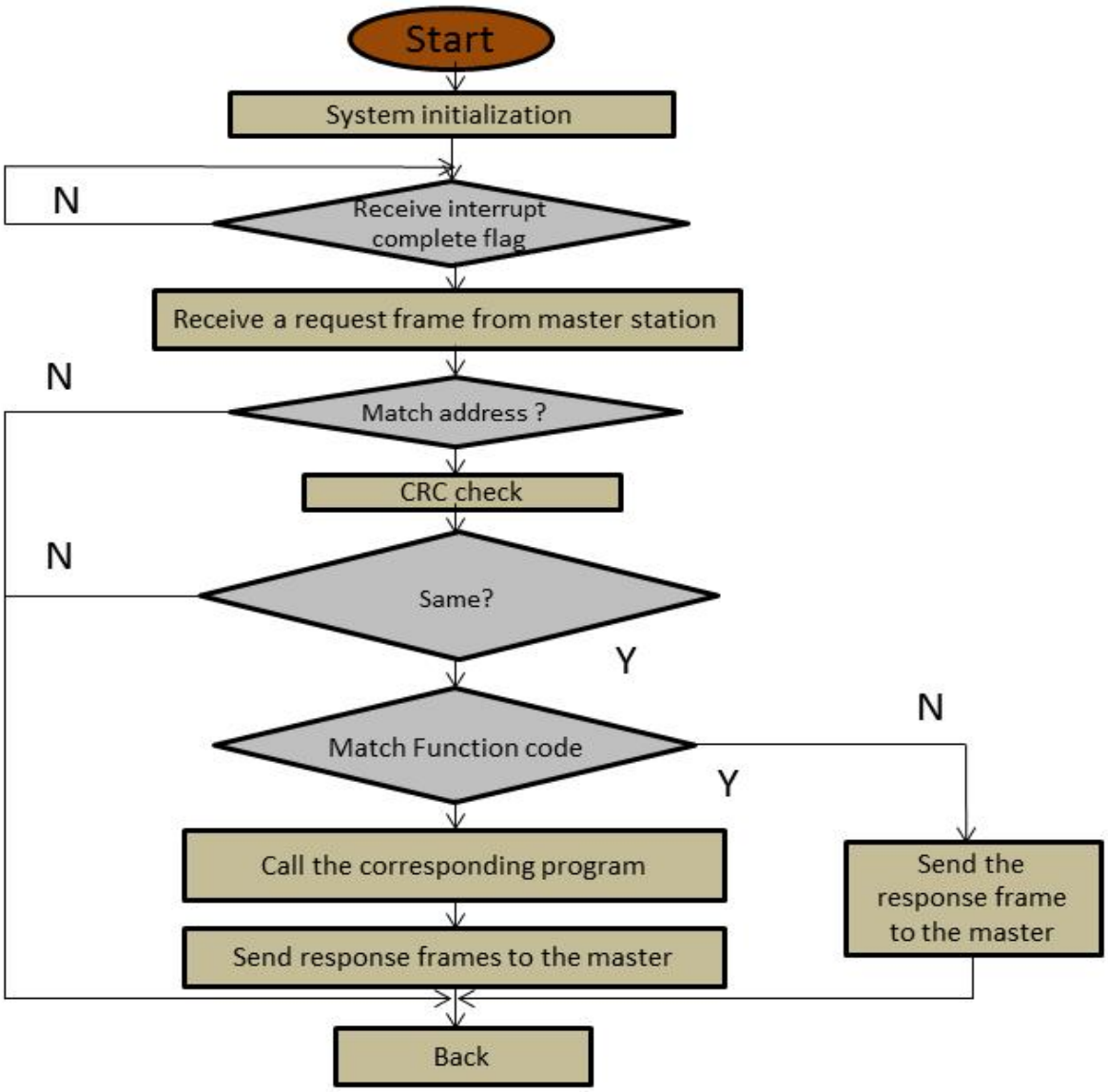

Fig 2: MODBUS implementation steps

4 EXPERIMENTAL RESULTS

The MODBUS master sends the frame that consists of the following field to the slave to request for communication and in response slave confirms the communication if the address field matches.

So, in Table 2 the request frame sent from the master for the slave with slave address 01 is shown. After that function code is 06 which write the single holding register from 60004. As the address is given 04 so it will write at 60004. Next field is for the value to be written at that given address so it is stated as 0001. Then comes the CRC field which generates the CRC for the above 06 bytes and appends to the frame and forms the request frame for sending to slave.

In Table 3 the response frame sent from the slave to master is shown. It consists of the first field of slaveaddress which says that the data sent is from the slave with slave address 01. The next filed is function code which says for this function the process was done. The next field is for the CRC generated for the response generated and appended to the response frame.

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 8
August 2017

Academic Science

Table 2: Master sends request frame to slave for function code 06.

Field	Master Write
Slave Address	01
Function code	06
Starting Address HI	00
Starting Address LO	04
Write data HI	00
Write data LO	01
CRC HI	09
CRC LO	CB

Table 3: Slave sends response frame to master for function code 06.

Field	Slave Write
Slave Address	01
Function code	06
Starting Address HI	00
Starting Address LO	04
Write data HI	00
Write data LO	01
CRC HI	E0
CRC LO	1A

Similarly for the read register function 03 the response sent from slave to master is shown in Fig.3. It reads the value written by 06 function.

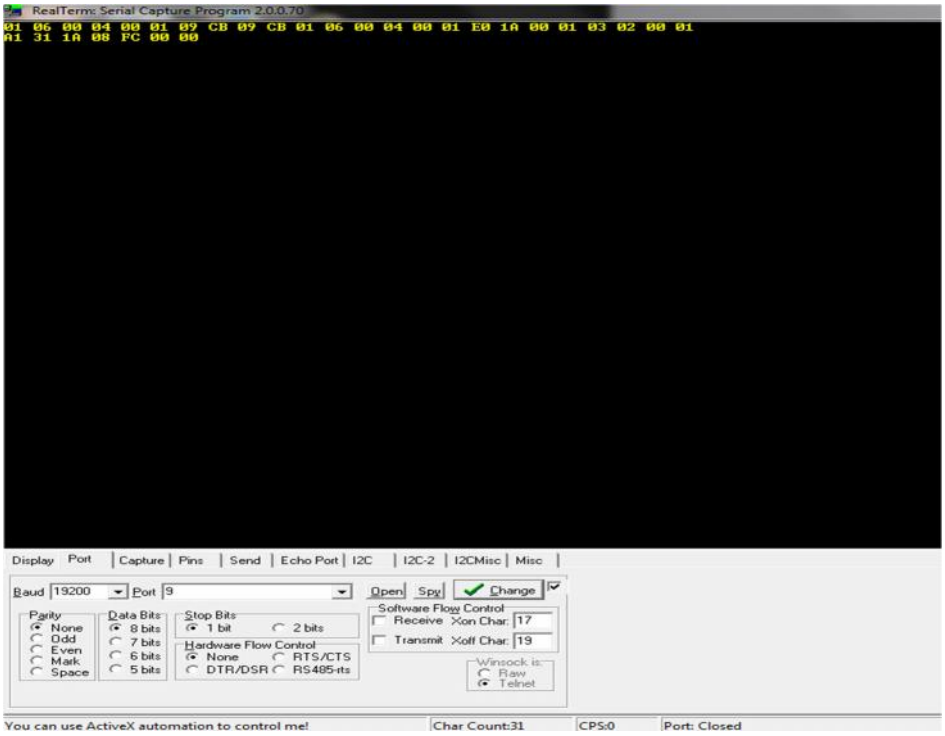

Fig 3: Hex values of the frame sent and received for the function code 06 and 03 is shown on Realterm software

CONCLUSION

For addressing the emerging need of the reliable, fast data acquisition system, Cortex M3 proves to be power efficient and has speed. So using RS485, the network was made that can communicate with various devices at one time.

So in this design master need to read the configuration set and configured the processor accordingly so that it can communicate to the particular slave for that master needs to have the slave address. So the system proved to be reliable for the communication among various devices. This system implements the Serial RTU master slave which can further be implemented at TCP/IP at industrial level which depends on the requirement of the application.

REFERENCES

[1] Kelong Wang, Daogang Peng, Lei Song, Hao Zhang- "Implementation of Modbus Communication Protocol Based on ARM Coretx-MO", IEEE International Conference on System Science and Engineering (JCSSE), pp. 69 – 73 , 2014.

[2] Dianru Jiaa , Shuangyou Wangb , Junfang Tiana, "Electric Parameter Acquisition System Based On Arm", Chemical Engineering Transactions, Vol. 46, pp. 289-294, 2015.

[3] An Energy Meter Monitoring Network using Modbus Routers, Application, Circulated by Cygnus Microsystems (P) Limited, Release 1. Date 7 July 2015. Circulated by Cygnus Microsystems (P) Limited, Hyderabad.

[4] MODBUS over Serial Line Specification Implementation guide V1.0.

[5] Sadik Tamboli, Mallikarjun Rawale, Rupesh Thoraiet, Sudhir Agashe, "Implementation of Modbus RTU and Modbus TCP communication using Siemens S7- 1200 PLC for batch process*", International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM),* pp. 258 – 263, 2015.