INTERNATIONAL JOURNAL OF ADVANCE RESEARCH AND DEVELOPMENT
IJARnD
(Volume3, Issue3)
Available online at: www.ijarnd.com

Microcontroller based Data Acquisition System using RTU Protocol

V. Sai Sumanth Srivatsa

Student, Sreenidhi Institute of Science and Technology, Hyderabad, Telangana

ABSTRACT

The objective of this paper is to design and implement a data acquisition system (DAQ) by using serial RS-232 and SPI communication protocols PC platform. The developed DAQ system should be able to acquire both analog signals as well as digital signals. The system converts the analog signals into digital data and sends the data into the computer using RS-232 serial communication for further processing. The development of the system has been divided into two phases. The first phase is to the design of SPI interface, while the second phase is to the design of the RS-232 interface. The SPI interface deals with the ADC and amplifier communication with FPGA and the RS-232 interface deals with the communication between PC and FPGA. The SPI and RS-232 communication protocol have been developed using C programming language

Data acquisition system project is a fine combination of analog and digital electronics. This project is designed so as to fulfill the requirements of the industry applications, home applications. The project consists of parameters monitoring, parameter storage which is also known as Data Logger. PC interface is one of the main features of the project in which various data like the value of parameters, date and time are sent to the PC. The microcontroller as a main component of the project. Now a Microcontroller has become the main component of many of the electronic circuits. Also, Liquid Crystal Display (LCD) and EEPROM are used on the major basis for the display and storage purpose. I2C bus protocol is used while interfacing 8051 with EEPROM. This project which will consist of two basis modules. First is "Data monitoring" & other is "Data Storage".

Keywords: *Modbus, RTU, Data Acquisition System.*

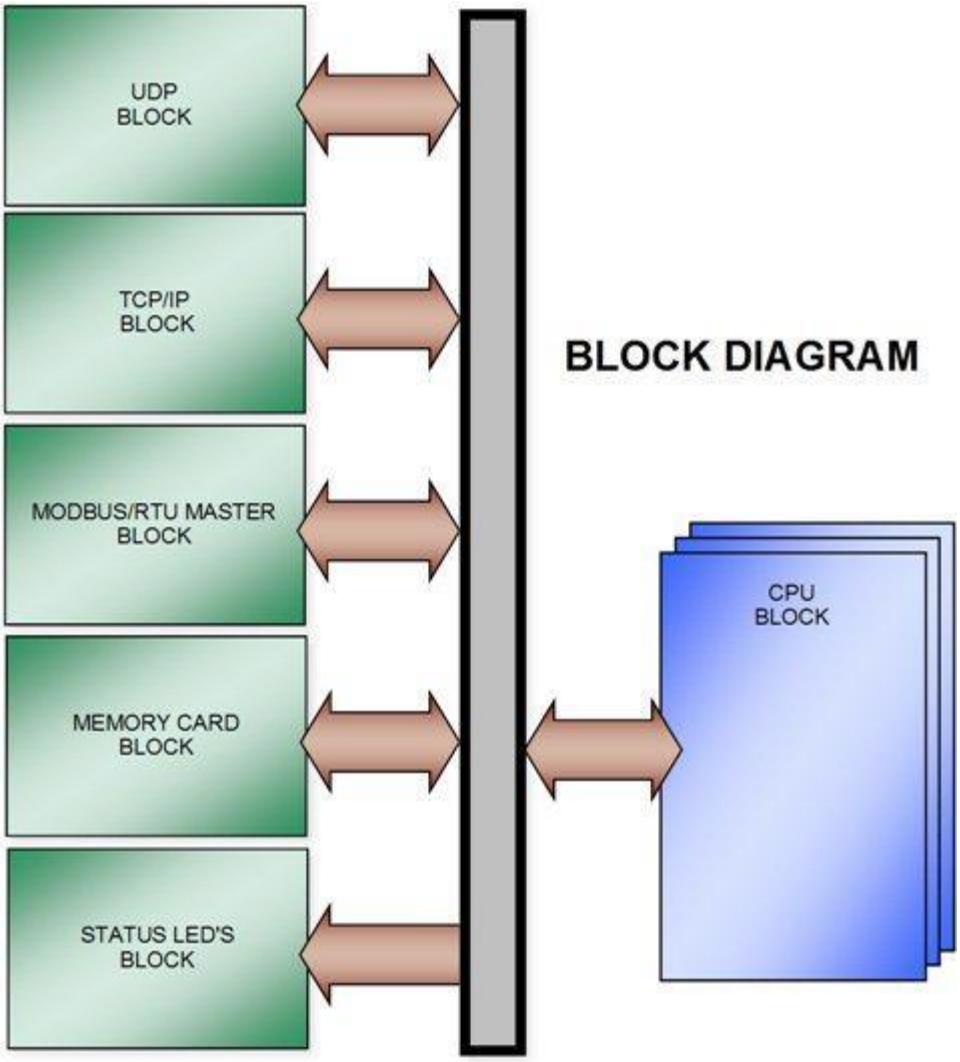
1. INTRODUCTION

In industrial environments, there is always need for analyzing the behavior of a process through its variables, and a data acquisition system can be the solution. Also, it is very common that the devices distributed over the plant to act, measure and control be interconnected by a communication network (Modbus, Profibus, Fieldbus, etc.). Data acquisition is the process that measures an electrical or physical phenomenon such as voltage, current, temperature, frequency etc. Every data acquisition system shares a common goal in order to acquire, analyze and present the information. Data acquisition systems incorporate signals, sensors, actuators, signal conditioners, data acquisition devices and application software.

In industrial automation, an industrial network is defined as the communication protocols used to supervise and control a given process, such as a quick and precise exchange of data between sensors, actuators, computers, programmable logic controllers (CLP) among others.

In anticipation of this market need, a new system was proposed for the acquisition of data applied to industrial networks whose protocol is the Modbus and the physical means the serial 485.e o meio físico serial 485. The acquisitor features a Modbus/RTU master on the Modbus/RTU network.

The proposed data acquisition system can be divided into three functional blocks. The first one is a TCP/IP block responsible for the configuration of rules of acquisition, monitoring and transference of data via dedicated software. The second is a Modbus/RTU slave block on the Modbus/RTU network. And the third a script block in the FAT32 format responsible for the storage of data in a SD micro memory card. This article will cover, in addition to an introduction to the data acquisition system, a study of the topology used in industrial networks, the protocol flexibility, and reliability when delivering data to the client and case studies in Brazil´s industrial networks.



2. RS-232 PROTOCOL HARDWARE IMPLEMENTATION

A. Serial communication

Serial communication enables different types of equipment to communicate with outside world, where information to be transformed is represented in serial data bits and will be sent in a serial way over a single line. The personal computer has a serial data bits and will be sent in a serial way over a single line. The personal computer has a serial port known as a communication port or COM port in PC [6]. Serial ports are controlled by a special chip called UART (Universal asynchronous receiver transmitter). Different application uses different pins on the serial port and this basically depends on the functions required. Devices that use serial cables for their communications are split into two categories DCE (Data communication equipment) and DTE (Data terminal equipment) device.

B. Interfacing DAQ with RS-232

In order to establish the communication link between the PC and the data acquisition module, it is necessary to use either serial or parallel communication channel. The information is transferred between data processing equipment and peripherals is in the form of digital data which is transmitted either in a serial or parallel mode; Parallel communications are used mainly for connections between test instruments or computers and printers, while serial is often used between computers and other peripherals. Serial transmission involves the sending of data one bit at a time, over a single communication line [7]. In contrast, parallel communications require at least as many lines as there are bits in a word being transmitted. In asynchronous serial communication, the date is transmitted by a bit in order, in which two transition lines are needed to realize bidirectional communication. Serial communication is utilized in the DAQ system in order to simplify the hardware and save the cost, format and save your graphic images using a suitable graphics processing program that will allow you to create the images as PostScript (PS), Encapsulated PostScript (EPS), or Tagged Image File Format (TIFF), sizes them, and adjusts the resolution settings. If you created your source files in one of the following you will be able to submit the graphics without converting to a PS, EPS, or TIFF file: Microsoft Word, Microsoft PowerPoint, Microsoft Excel, or Portable Document Format (PDF).

C. RS-232 Serial port

RS-232 is a telecommunication standard for binary serial communications between devices. It provides the roadmap for

The way devices communicate each other using serial ports. The devices are commonly referred to as a DTE (Data terminal equipment) and DCE (Data communication equipment). Updated designations for the RS-232 protocols have included EIA-232 and the more current EIA/TIA-232. (Telecommunication industry association).

3. SPI PROTOCOL HARDWARE IMPLEMENTATION

A. SPI Bus principle

The serial peripheral interface bus is a synchronous serial data link standard named by Motorola that operates in full duples mode, sometimes SPI is called a "four wire" serial bus. There are one master and one or more slave devices in the communication. The general Master-Slave configuration is as shown in Fig 2.

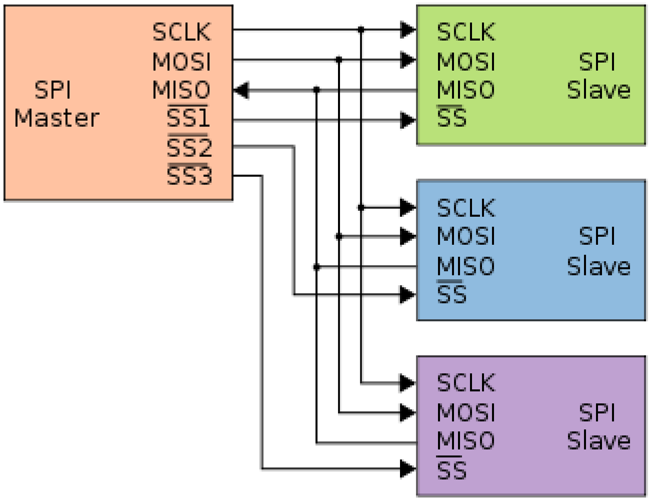


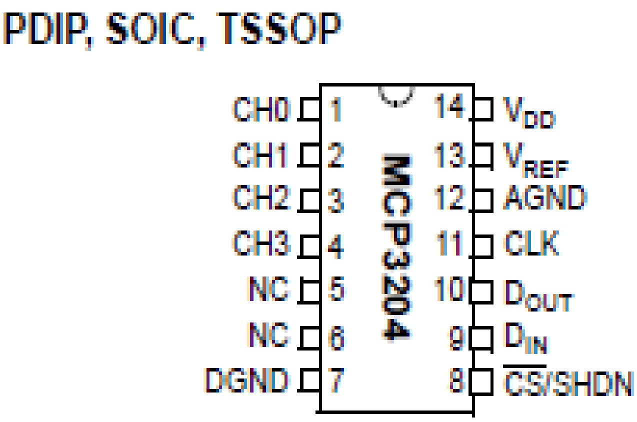
Fig 2. General SPI bus system with 1 Master device and with 3 slaves

4. ADC (MCP3204)

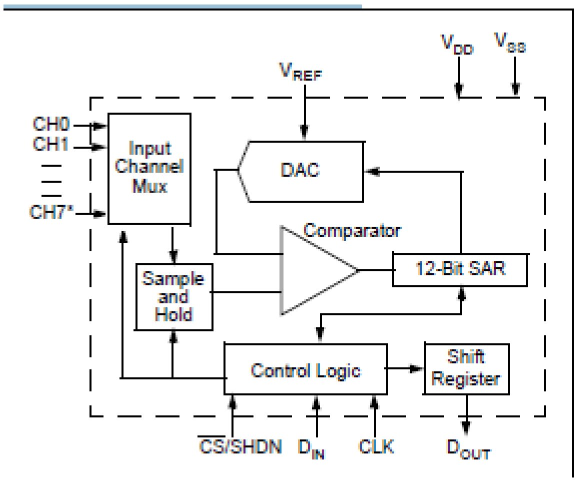
The Microchip Technology Inc. MCP3204/3208 devices are successive approximation 12-bit Analog-to-Digital (A/D) Converters with on-board sample and hold circuitry. The MCP3204 is programmable to pro-vide two pseudo-differential input pairs or four single-ended inputs. The MCP3208 is programmable to pro-vide four pseudo-differential input pairs or eight single-ended inputs. Differential Nonlinearity (DNL) is speci-fied at ±1 LSB, while Integral Nonlinearity (INL) is offered in ±1 LSB (MCP3204/3208-B) and ±2 LSB (MCP3204/3208-C) versions.

Communication with the devices is accomplished using a simple serial interface compatible with the SPI proto-col. The devices are capable of conversion rates of up to 100 ksps. The MCP3204/3208 devices operate over a broad voltage range (2.7V - 5.5V). Low current design permits operation with typical standby and active currents of only 500 nA and 320 µA, respec-tively. The MCP3204 is offered in 14-pin PDIP, 150 mil SOIC and TSSOP packages. The MCP3208 is offered in 16-pin PDIP and SOIC packages.

Package Types



Functional Block Diagram



5. PIN DESCRIPTIONS

TABLE 3-1: PIN FUNCTION TABLE

Name	Function
VDD	+2.7V to 5.5V Power Supply
DGND	Digital Ground
AGND	Analog Ground
CH0-CH7	Analog Inputs
CLK	Serial Clock
DIN	Serial Data In
DOUT	Serial Data Out
$\overline{CS}$/SHDN	Chip Select/Shutdown Input
VREF	Reference Voltage Input

5.1 DGND

Digital ground connection to the internal digital circuitry.

5.2 AGND

Analog ground connection to the internal analog circuitry.

5.3 CH0 - CH7

Analog inputs for channels 0 - 7 for the multiplexed inputs. Each pair of channels can be programmed to be used as two independent channels in single-ended mode or as a single pseudo-differential input, where one channel is IN+ and one channel is IN. See Section 4.1, “Analog Inputs”, and Section 5.0, “Serial Communications”, for information on programming the channel configuration.

5.4 Serial Clock (CLK)

The SPI clock pin is used to initiate a conversion and clock out each bit of the conversion as it takes place. See Section 6.2, “Maintaining Minimum Clock Speed”, for constraints on clock speed.

5.5 Serial Data Input (D_{IN})

The SPI port serial data input pin is used to load channel configuration data into the device.

5.6 Serial Data Output (D_{OUT})

The SPI serial data output pin is used to shift out the results of the A/D conversion. Data will always change on the falling edge of each clock as the conversion takes place.

5.7 Chip Select/Shutdown (CS/SHDN)

The CS/SHDN pin is used to initiate communication with the device when pulled low and will end a conver-sion and put the device in low power standby when pulled high. The CS/SHDN pin must be pulled high between conversions.

6. DEVICE OPERATION

The MCP3204/3208 A/D converters employ a conventional SAR architecture. With this architecture, a sample is acquired on an internal sample/hold capacitor for 1.5 clock cycles starting on the fourth rising edge of the serial clock after the start bit has been received. Fol-lowing this sample time, the device uses the collected charge on the internal sample/hold capacitor to produce a serial 12-bit digital output code. Conversion rates of 100 ksps are possible on the MCP3204/3208. See Section 6.2, "Maintaining Minimum Clock Speed", for information on minimum clock rates. Communication with the device is accomplished using a 4-wire SPI-compatible interface.

6.1 Analog Inputs

The MCP3204/3208 devices offer the choice of using the analog input channels configured as single-ended inputs or pseudo-differential pairs. The MCP3204 can be configured to provide two pseudo-differential input pairs or four single-ended inputs, while the MCP3208 can be configured to provide four pseudo-differential input pairs or eight single-ended inputs. Configuration is done as part of the serial command before each con-version begins. When used in the pseudo-differential mode, each channel pair (i.e., CH0 and CH1, CH2 and CH3 etc.) is programmed to be the IN+ and IN- inputs as part of the command string transmitted to the device. The IN+ input can range from IN- to (V_{REF} + IN-). The IN- input is limited to ±100 mV from the V_{SS} rail.

The IN- input can be used to cancel small signal com-mon-mode noise which is present on both the IN+ and IN- inputs.

When operating in the pseudo-differential mode, if the voltage level of IN+ is equal to or less than IN-, the resultant code will be 000h. If the voltage at IN+ is equal to or greater than {[V_{REF} + (IN-)] - 1 LSB}, then the output code will be FFFh. If the voltage level at IN- is more than 1 LSB below V_{SS}, the voltage level at the IN+ input will have to go below V_{SS} to see the 000h output code. Conversely, if IN- is more than 1 LSB above V_{SS}, then the FFFh code will not be seen unless the IN+ input level goes above V_{REF} level.

For the A/D converter to meet specification, the charge holding capacitor (C_{SAMPLE}) must be given enough time to acquire a 12-bit accurate voltage level during the 1.5 clock cycle sampling period.

7. SERIAL COMMUNICATIONS

Communication with the MCP3204/3208 devices is accomplished using a standard SPI-compatible serial interface. Initiating communication with either device is done by bringing the CS line low (see Figure 5-1). If the device was powered up with the CS pin low, it must be brought high and back low to initiate communication. The first clock received with CS low and D_{IN} high will constitute a start bit. The SGL/DIFF bit follows the start bit and will determine if the conversion will be done using single-ended or differential input mode. The next three bits (D0, D1, and D2) are used to select the input channel configuration. Table 5-1 and Table 5-2 show the configuration bits for the MCP3204 and MCP3208, respectively. The device will begin to sample the ana-log input on the fourth rising edge of the clock after the start bit has been received. The sample period will end on the falling edge of the fifth clock following the start bit.

Once the D0 bit is input, one more clock is required to complete the sample and hold period (D_{IN} is a "don't care" for this clock). On the falling edge of the next clock, the device will output a low null bit. The next 12 clocks will output the result of the conversion with MSB first, as shown in Figure 5-1. Data is always output from the device on the falling edge of the clock. If all 12 data bits have been transmitted and the device continues to receive clocks while the CS is held low, the device will output the conversion result LSB first, as shown in Figure 5-2. If more clocks are provided to the device while CS is still low (after the LSB first data has been transmitted), the device will clock out zeros indefinitely.

If necessary, it is possible to bring CS low and clock in leading zeros on the D_{IN} line before the start bit. This is often done when dealing with microcontroller-based SPI ports that must send 8 bits at a time. Refer to Section 6.1 for more details on using the MCP3204/ 3208 devices with hardware SPI ports.

Table 7-1: Configuration Bits for the MCP3204

Control Bit Selections				Input Configuration	Channel Selection
Single/Diff	**D2***	**D1**	**D0**		
1	X	0	0	single-ended	CH0
1	X	0	1	single-ended	CH1
1	X	1	0	single-ended	CH2
1	X	1	1	single-ended	CH3
0	X	0	0	differential	CH0 = IN+ CH1 = IN-
0	X	0	1	differential	CH0 = IN- CH1 = IN+
0	X	1	0	differential	CH2 = IN+ CH3 = IN-
0	X	1	1	differential	CH2 = IN- CH3 = IN+

*D2 is a “don’t care” for MCP3204

MODBUS PROTOCOL:

The Query: The function code in the query tells the addressed slave device what kind of action to perform. The data bytes contain any additional information that the slave will need to perform the function. For example, function code 03 will query the slave to read holding registers and respond with their contents. The data field must contain the information telling the slave which register to start at and how many registers to read. The error check field provides a method for the slave to validate the integrity of the message contents.

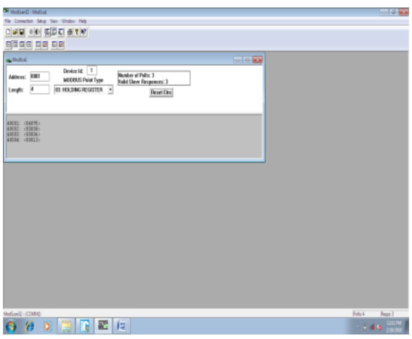
The Response: If the slave makes a normal response, the function code in the response is an echo of the function code in the query. The data bytes contain the data collected by the slave, such as register values or status. If an error occurs, the function code is modified to indicate that the response is an error response, and the data bytes contain a code that describes the error. The error check field allows the master to confirm that the message contents are valid.

RTU Mode

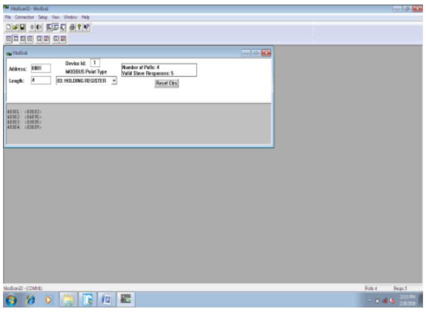
When controllers are setup to communicate on a Modbus network using RTU (Remote Terminal Unit) mode, each 8±bit byte in a message contains two 4±bit hexadecimal characters. The main advantage of this mode is that its greater character density allows better data throughput than ASCII for the same baud rate. Each message must be transmitted in a continuous stream.

8. RESULTS

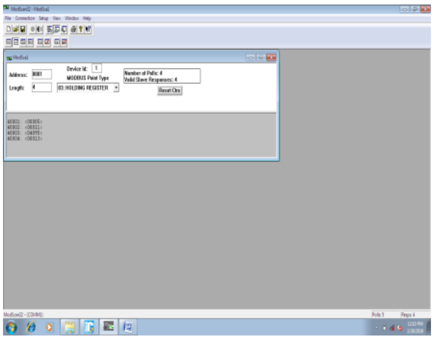
CHANNEL-0



CHANNEL-1



CHANNEL-2



CHANNEL-3