SCADA BASED AUTOMATION SYSTEM

Using MODBUS protocol

Submitted in partial fulfillment of the requirements

Of the degree of

(Electronics & Telecommunication Engineering)

By

RON GULAM HUSSAIN	**(13ET73)**
DANDOTI SAIFUDDIN	**(13ET67)**
LAKHA AVESH AMIN	**(13ET69)**
KHAN MOHD UZAIR Z	**(12ET33)**

Supervisor:

(Asst. Prof.Rahul B.Khadase)

(Electronics & Telecommunication Engineering)

ANJUMAN-I-ISLAM'S
KALSEKAR TECHNICAL CAMPUS
PANVEL

(2015-2016)

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

1) -------------------------------------	----------------------
2) -------------------------------------	----------------------
3) -------------------------------------	----------------------
4) -------------------------------------	----------------------
(Name of students & Roll No)	(Signature)

Date:

Place:

CERTIFICATE

This is to certify that the project entitled "**SCADA BASED AUTOMATION SYSTEM using MODBUS protocol**" is the bonafide work carried out by final year BE students of EXTC, KalsekarTechnical Campus, Panvel, during the year 2015-2016, in partial fulfillment of the requirements for the award of the Degree of B.E in Electronics & Telecommunication Branch.

----------------------------	---------------------------------
(Prof. Mujib Tamboli)	(Asst. Prof.Rahul B. Khadase)

(External Examiner)

Date:

Place:

ACKNOWLEDGEMENT

It is indeed a matter of great pleasure and proud privilege to be able to present this project on “**SCADA BASED AUTOMATION SYSTEM using MODBUS protocol.**

The completion of the project work is a milestone in student life and its execution is inevitable in the hands of guide. We are highly indebted the project guide **Asst. Prof RAHUL KHADASE** for his invaluable guidance and appreciation for giving form and substance to this report. It is due to his enduring efforts; patience and enthusiasm, which has given a sense of direction and purposefulness to this project and ultimately made it a success.

We would like to tender our sincere thanks of staff members for their co-operation.

We would wish to thank the non-teaching staff and our senior friends who have helped us all the time in one way or the other .Really it is highly impossible to repay the debt of all the people who have directly or indirectly helped us for performing the project.

PREFACE

We take an opportunity to present report on "**SCADA BASED AUTOMATION SYSTEM using MODBUS protocol**" and put before readers some useful information regarding our project.

We have made sincere attempts and taken every care to present this matter in precise and compact form, the language being as simple as possible.

We are sure that the information contained in this volume would certainly prove useful for better insight in the scope and dimension of this project in its true perspective.The task of completion of the project though being difficult was made quite simple, interesting and successive due to deep involvement and complete dedication of our group members.

TABLE OF CONTENT

Topics **Page no**

LIST OF FIGURES & TABLES Page No

ABSTRACT

This project focuses on conversion of manually operated temperature & Load (Bulb) Controller system towards fully automated system by using Microcontroller and peripherals like RTD, ADC &Relay Circuit and monitor whole process on computer screen using SCADA software.

Our prime concern is to demonstrate industrial automation of temperature monitoring & controlling and load controlling system using little hardware and more refined software details.

CHAPTER-1

1.1. BASIC DEFINATIONS

1.1.1. **Communication**: The imparting or exchanging of information by speaking, writing, or using some other medium. Means of sending or receiving information, such as telephone lines or computers. Communication are of two types either wired or wireless, in wireless communication the transmission and reception of data carried out by using air as a medium and in wired communication by physical wires such as coaxial or fiber optic cables.

1.1.2. **Automation**: Automation is the technique, method or system of operating or controlling a process by highly automatic means, as by electronic devices, reducing human intervention to a minimum.

1.1.3. **Protocol:** A protocol is a set of rules that end points in a telecommunication connection when they communicate. It determines the way of communication from transmitter to receiver and vice versa.

1.2. INTRODUCTION

- In this project we are designing a system in which we are controlling & monitoring the temperature of boiler setup and also controlling the Load (Bulb) by switching it ON/OFF and monitoring the whole process on pc screen using SCADA software.

- Our project consist of three major parts,
    - SCADA as monitoring software.
    - Microcontroller and Peripherals such as (ADC, RTD, Relay Circuit, Load (Bulb)) as a controlling interface.
    - Boiler setup as a slave 1 and Load (Bulb) as a slave 2.

- They all are connected serially in round robin manner and communicate with each other using MODBUS protocol.

CHAPTER-2

2.1. BLOCK DIAGRAM

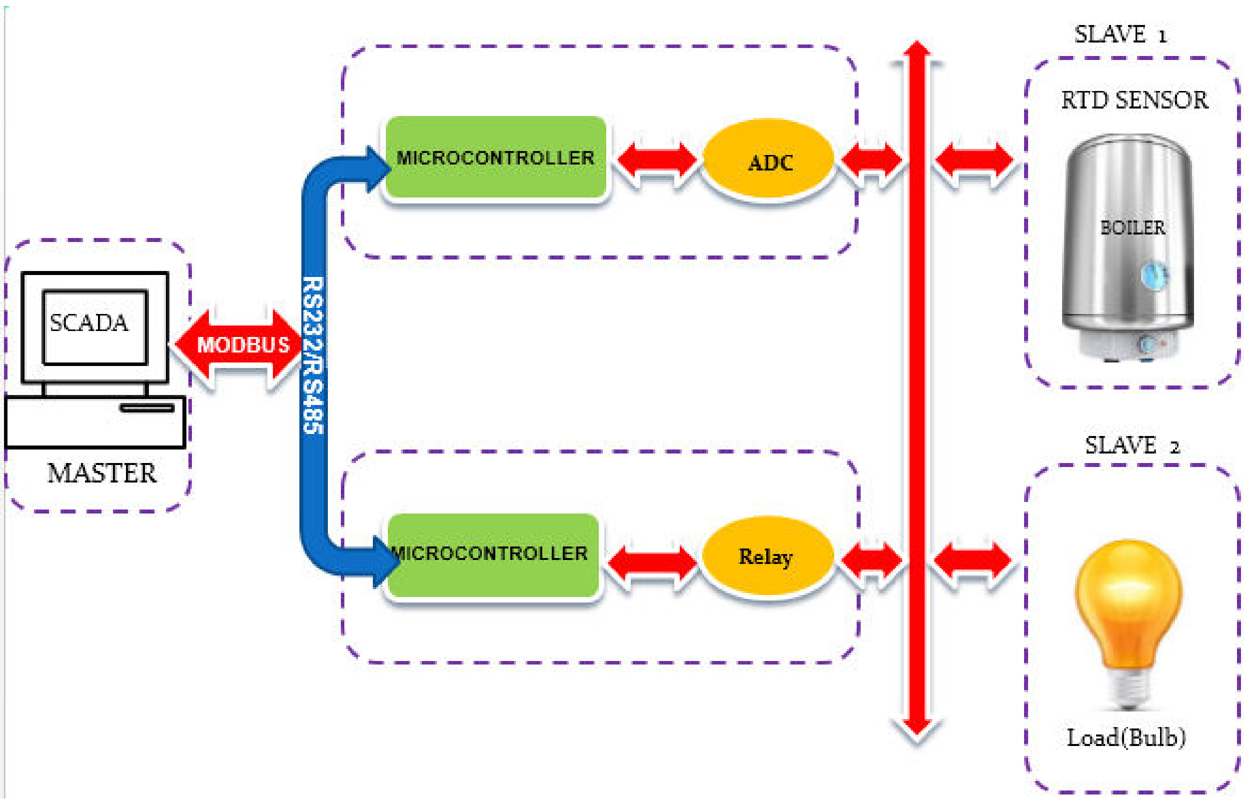


Fig 2.1.Block Diagram of Project

2.2. PROJECT MODULES

2.2.1MICROCOTROLLER:

A **microcontroller** (sometimes abbreviated **µC**, **uC** or **MCU**) is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.

Types of microcontroller

Microcontrollers can be classified on the basis of internal bus width, architecture, memory and instruction set.

THE 8, 16 AND 32-BITS MICROCONTROLLERS

THE 8-BIT MICROCONTROLLER

When the ALU performs arithmetic and logical operations on a byte (8-bits) at an Instruction, the microcontroller is an 8-bit microcontroller. The internal bus width of8-bit microcontroller is of 8-bit. Examples of 8-bit microcontrollers are Intel 8051 family And Motorola MC68HC11 family.

THE 16-BIT MICROCONTROLLER

When the ALU performs arithmetic and logical operations on a word (16-bits) at an Instruction, the microcontroller is a 16-bit microcontroller. The internal bus width of16-bit microcontroller is of 16-bit. Examples of 16-bit microcontrollers are Intel 8096Family and Motorola MC68HC12 and MC68332 families. The performance and computing capability of 16 bit microcontrollers are enhanced with greater precision asCompared to the 8-bit microcontrollers.

THE 32-BIT MICROCONTROLLER

When the ALU performs arithmetic and logical operations on a double word (32-Bits) at an instruction, the microcontroller is a 32-bit microcontroller. The internal bus Width of 32-bit microcontroller is of 32-bit. Examples of 32-bit microcontrollers are Intel80960 family and Motorola M683xx and Intel/Atmel 251 family. The performance and computing capability of 32 bit microcontrollers are enhanced with greater precision as Compared to the 16-bit microcontrollers.

MICROCONTROLLER MEMORY TYPES

There are number of different types of control store (Program memory) that are available in different versions and different manufacturers of 8051.There is a fairly simple convention that is used to identify what type of control store a device has.

"X" value	Control system Type
0	None
3	Mask ROM
7	EPROM
9	EPROM/FLASH

Table 2.1.The following is the list of convention used for 8x51

2.2.1.1MICROCONTROLLER PIN DIAGRAM

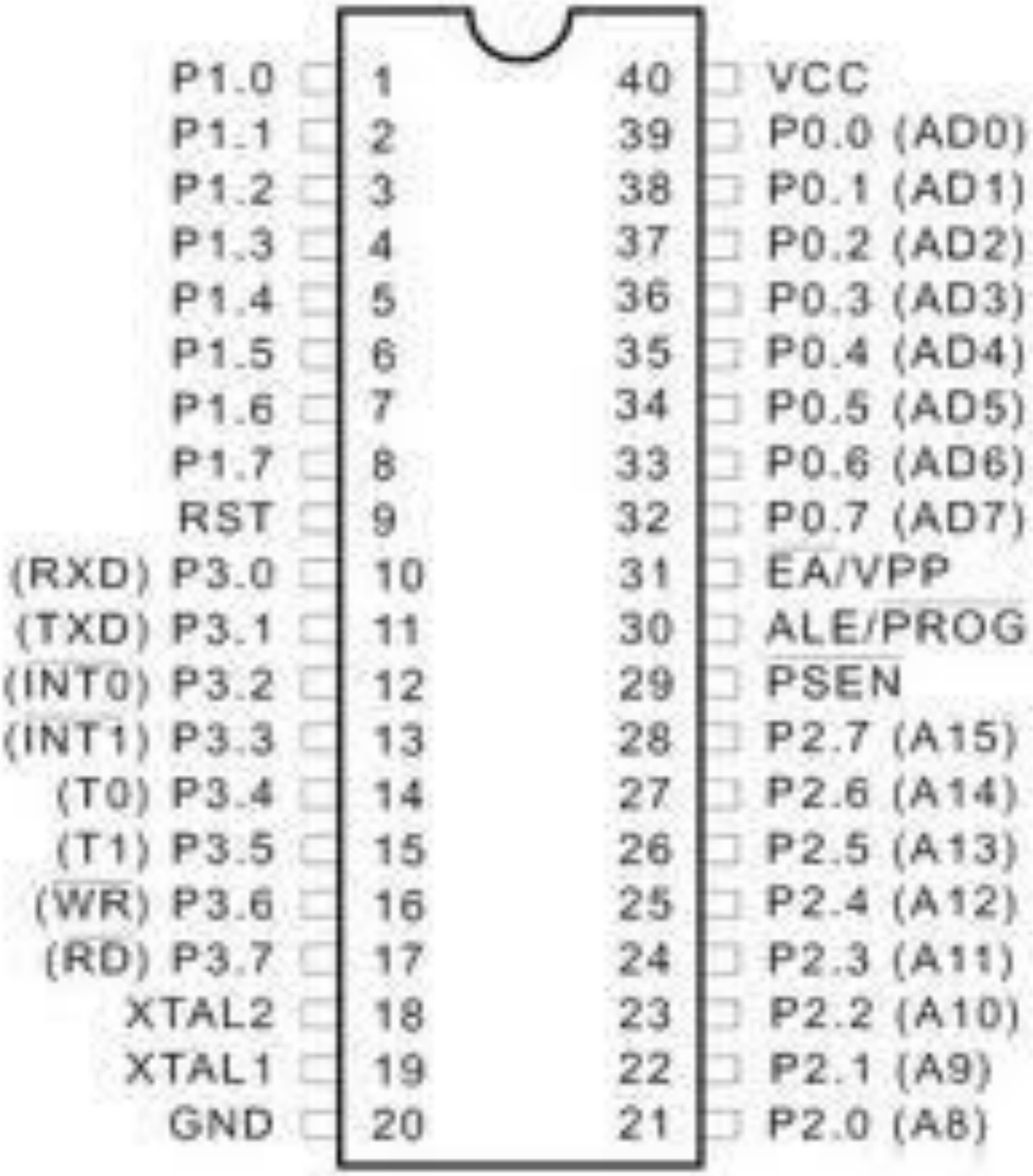


Fig 2.2.Pin diagram of microcontroller

PIN DESCRIPTION

VCC

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs. Port 0 may also be configured to be the multiplexed low order address/data bus during accesses to external program and data memory. In this mode P0 has internal pull-ups. Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pull-ups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pull ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. Port 1 also receives the low-order address bytes during Flash programming and program verification.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register. Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL) because of the pull-ups. Port 3 also serves the functions of various special features of the microcontroller as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (extenal interrupt 0)
P3.3	$\overline{\text{INT1}}$ (extenal interrupt 1)
P3.4	T0 (timer 0 extenal input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (extenal data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

Table 2.2.Port 3 functions of microcontroller

Port 3 also receives some control signals for Flash programming and Programming verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/PROG

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory.

This pin is also the program pulse input (PROG) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data

PSEN

Program Store Enable is the read strobe to external program memory. When the AT89C51 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

EA/VPP

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset. EA should be strapped to VCC for internal program executions. This pin also receives the 12-volt programming enable Voltage (VPP) during Flash programming, for parts that require 12-volt VPP.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier

2.2.1.2. BOLCKDIAGRAM OF MICROCONTROLLER

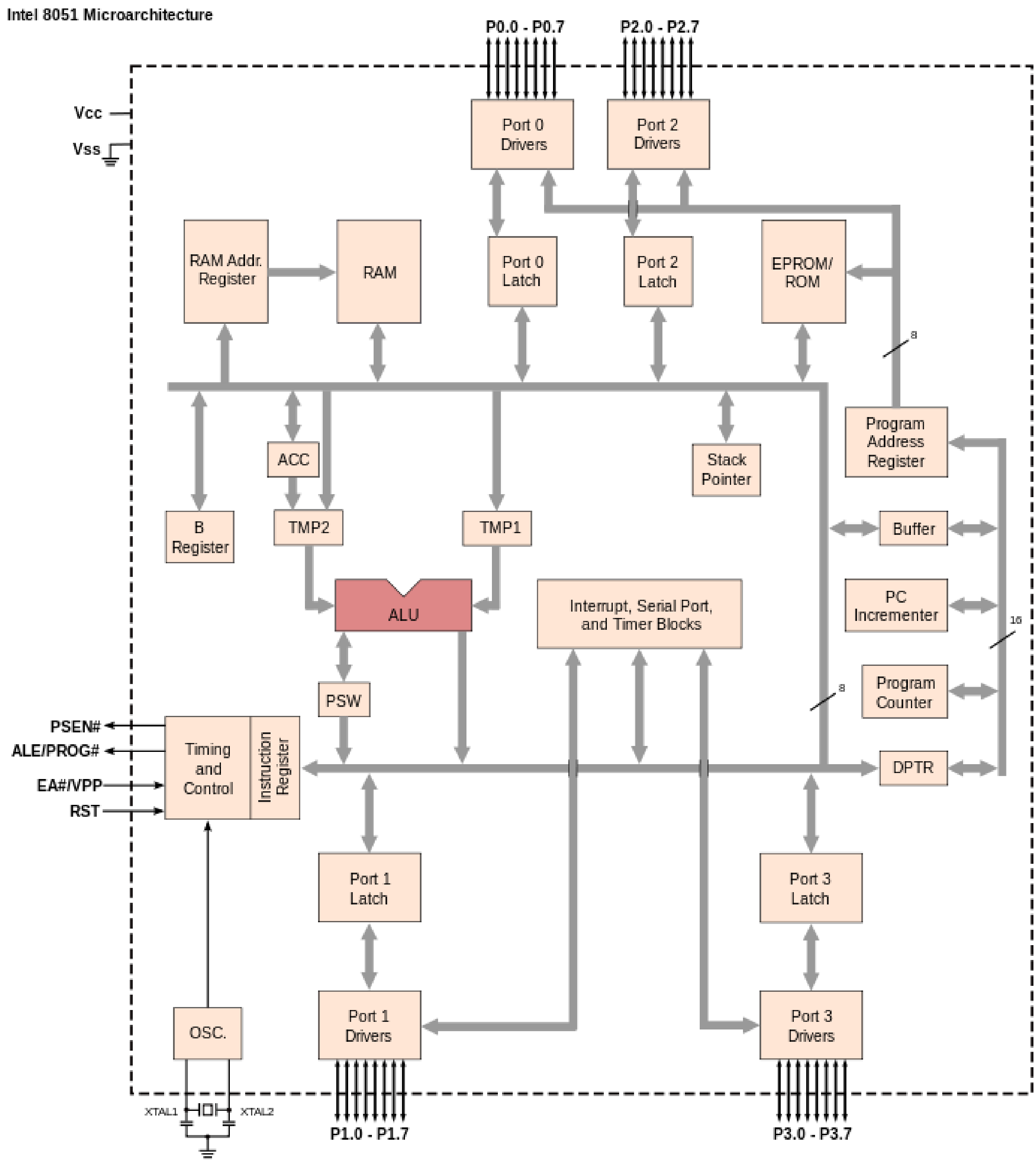


Fig 2.3.Internal architecture of microcontroller

2.2.1.3. MAX 232 IC (DUAL DRIVER/RECEIVER)

GENERAL DESCRIPTION

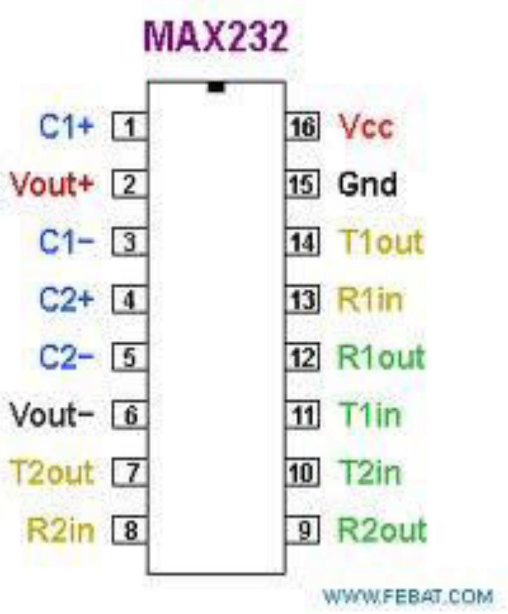


Fig 2.4.Pin diagram of MAX 232 IC

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels.

These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC library.

The MAX232 from Maxim was the first IC which in one package contains the necessary drivers (two) and receivers (also two), to adapt the RS-232 signal voltage levels to TTL logic. It became popular, because it just needs one voltage (+5V) and generates the necessary RS-232 voltage levels (approx. -10V and +10V) internally. This greatly simplified the design of circuitry. Circuitry designers no longer need to design and build a power supply with three voltages (e.g. -12V, +5V, and +12V), but could just provide one +5V power supply, e.g. with the help of a simple 78x05 voltage regulator.

The MAX232 has a successor, the MAX232**A**. The ICs are almost identical, however, the MAX232A is much more often used (and easier to get) than the original MAX232, and the MAX232A only needs external capacitors 1/10th the capacity of what the original MAX232 needs.It should be noted that the MAX232 (A) is just a driver/receiver. It does not generate the necessary RS-232 sequence of marks and spaces with the right timing, it does not decode the RS-232 signal, and it does not provide a serial/parallel conversion. **All it does is to convert signal voltage levels**. Generating serial data with the right timing and decoding serial data has to be done by additional circuitry, e.g. by a 16550 UART or one of these small micro controllers (e.g. Atmel AVR, Microchip PIC) getting more and more popular.

The MAX232 and MAX232A were once rather expensive ICs, but today they are cheap. It has also helped that many companies now produce clones (i.e. Sipex). These clones sometimes need different external circuitry, e.g. the capacities of the external capacitors vary. It is recommended to check the data sheet of the particular manufacturer of an IC instead of relying on Maxim's original data sheet.

The original manufacturer (and now some clone manufacturers, too) offers a large series of similar ICs, with different numbers of receivers and drivers, voltages, built-in or external capacitors, etc. E.g. The MAX232 and MAX232A need external capacitors for the internal voltage pump, while the MAX233 has these capacitors built-in. The MAX233 is also between three and ten times more expensive in electronic shops than the MAX232A because of its internal capacitors. It is also more difficult to get the MAX233 than the garden variety MAX232A.

A similar IC, the MAX3232 is nowadays available for low-power 3V logic.

PIN DESCRIPTION

Pin No	Function	Name
1	Capacitor connection pins	Capacitor 1 +
2		Capacitor 3 +
3		Capacitor 1 -
4		Capacitor 2 +
5		Capacitor 2 -
6		Capacitor 4 -
7	Output pin; outputs the serially transmitted data at RS232 logic level; connected to receiver pin of PC serial port	T_2 Out
8	Input pin; receives serially transmitted data at RS 232 logic level; connected to transmitter pin of PC serial port	R_2 In
9	Output pin; outputs the serially transmitted data at TTL logic level; connected to receiver pin of controller.	R_2 Out
10	Input pins; receive the serial data at TTL logic level; connected to serial transmitter pin of controller.	T_2 In
11		T_1 In
12	Output pin; outputs the serially transmitted data at TTL logic level; connected to receiver pin of controller.	R_1 Out
13	Input pin; receives serially transmitted data at RS 232 logic level; connected to transmitter pin of PC serial port	R_1 In
14	Output pin; outputs the serially transmitted data at RS232 logic level; connected to receiver pin of PC serial port	T_1 Out
15	Ground (0V)	Ground
16	Supply voltage; 5V (4.5V – 5.5V)	Vcc

Table 2.3.Pin description of Max 232 IC

FEATURES

1. In line with all the technical standard RS-232C
2.Only requires a single 5V power supply
3.Chip charge pump with a boost, voltage, polarity reversal ability to generate 10V and-10V voltage V
4.Low power consumption, the typical supply current of 5mA. The internal integration of two RS-232C driver
5. The internal integration of the two RS-232C receivers

PIN OUT FOR MAX232

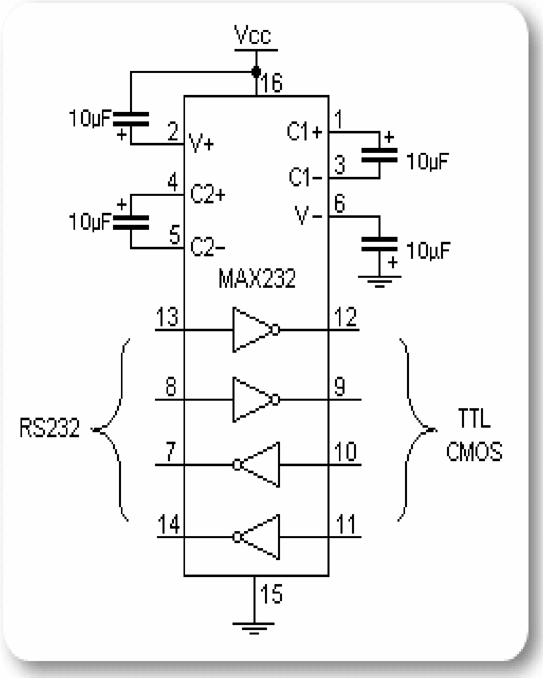


Fig 2.4.Pinout of Max 232 IC

The MAX232 IC is used to convert the TTL/CMOS logic levels to RS232 logic levels during serial communication of microcontrollers with PC. The controller operates at TTL logic level (0-5V) whereas the serial communication in PC works on RS232 standards (-25 V to + 25V). This makes it difficult to establish a direct link between them to communicate with each other.

The intermediate link is provided through MAX232. It is a dual driver/receiver that includes a capacitive voltage generator to supply RS232 voltage levels from a single 5V supply. Each receiver converts RS232 inputs to 5V TTL/CMOS levels. These receivers (R_1 & R_2) can accept ±30V inputs. The drivers (T_1 & T_2), also called transmitters, convert the TTL/CMOS input level into RS232 level.

The transmitters take input from controller's serial transmission pin and send the output to RS232's receiver. The receivers, on the other hand, take input from transmission pin of RS232 serial port and give serial output to microcontroller's receiver pin. MAX232 needs four external capacitors whose value ranges from 1µF to 22µF.

2.2.2. ADC (ANALOG TO DIGITAL CONVERTOR):

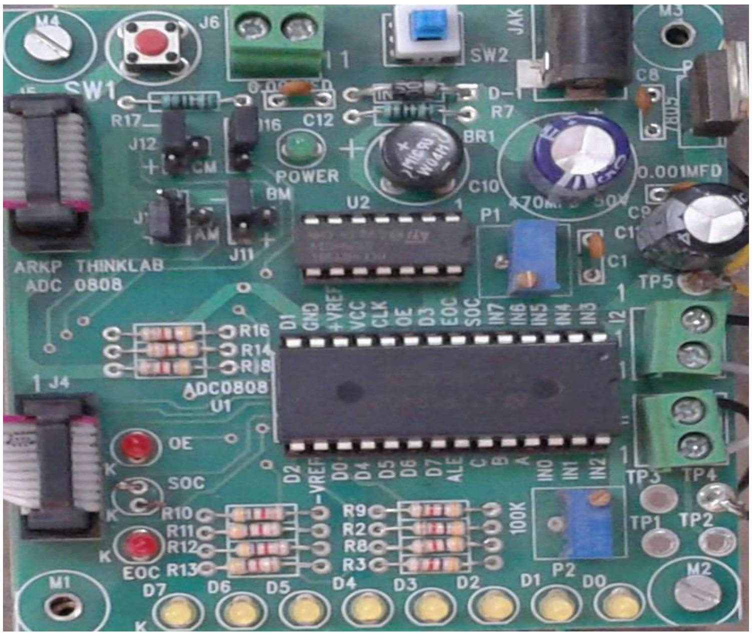


Fig.2.5.ADC 0808 PCB Board

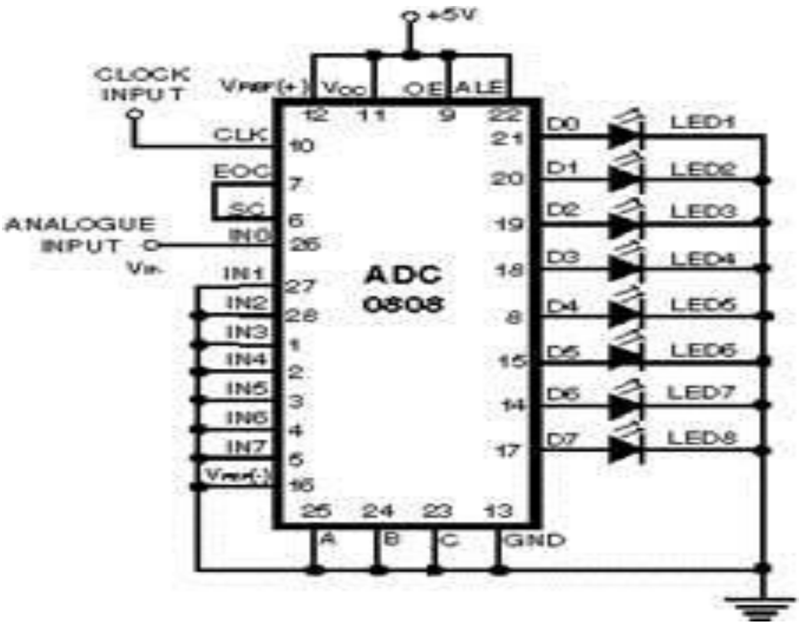


Fig.2.6.ADC0808 Pin Connections

ADC0808

The ADC0808 is an 8-bit analog to digital converter with a 8 channel analog multiplexer. The 8-bit A/D converter uses the successive approximation method for analog to digital conversion.

The ADC0808 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals. The device eliminates the need for external zero and full-scale. Adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and Latched TTL TRI-STATE outputs. The design of the ADC0808 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port)

The ADC0808 can be controlled with the help of 4 control signals:

- ALE (Address Latch Enable): On a Low to High transition at this pin the ADC latches the channel address on its multiplexed address lines.
- START: A High to Low transition at this pin will start the analog to digital conversion
- OUTPUT ENABLE: A High signal at this pin will latch the conversion output onto the output lines, which can be read by the Microcontroller
- EOC (End of Conversion): The EOC pin goes low after a conversion has completed

2.2.2.1. PIN DIAGRAM OF ADC0808

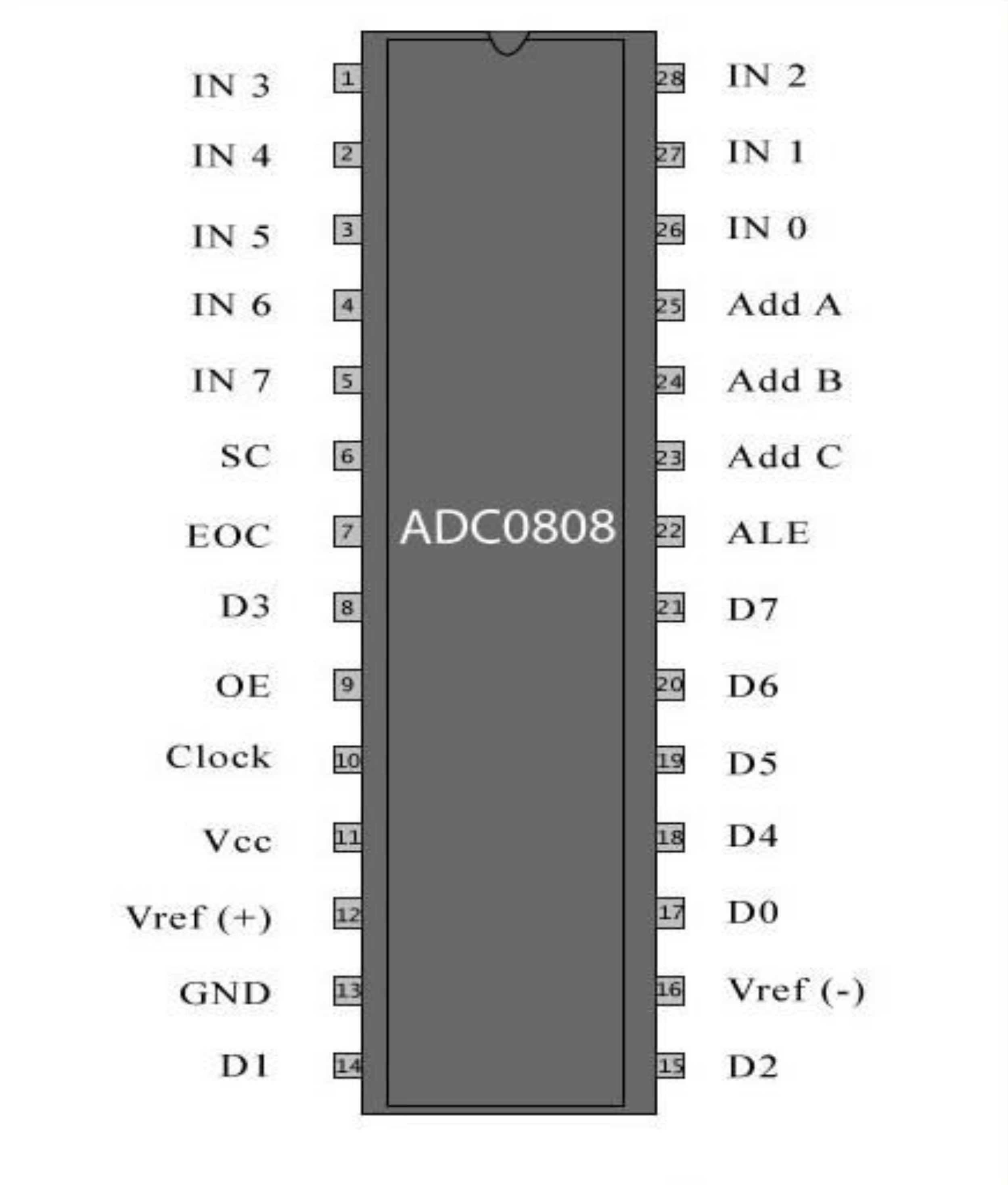


Fig 2.7.Pin diagram of ADC0808

FEATURES

- Easy interface to all microprocessors
- Operates ratio metrically or with 5 VDC or analog span
- Adjusted voltage reference
- No zero or full-scale adjust required
- 8-channel multiplexer with address logic
- 0V to VCC input range
- Outputs meet TTL voltage level specifications
- ADC0808 equivalent to MM74C949

KEY SPECIFICATIONS

- Resolution 8 Bits
- Total Unadjusted Error ±½ LSB and ±1 LSB
- Single Supply 5 VDC
- Low Power 15 mW
- Conversion Time 100 µs

FUNCTIONAL DESCRIPTION

MULTIPLEXER

The device contains an 8-channel single-ended analog signal multiplexer. A particular input channel is selected by using the address decoder. *Table 1* shows the input states for the address lines to select any channel. The address is latched into the decoder on the low-to-high transition of the address latch
Enable signal.

SELECTED ANALOG	**ADDRESS LINES**
CHANNELS	**C B A**
IN0	L L L
IN1	L L H
IN2	L H L
IN3	L H H
IN4	H L L
IN5	H L H
IN6	H H L
IN7	H H H

Table 2.4 Analog channel selection

ADC INITIALIZATION AND CONVERSION SEQUENCE:

An ADC conversion routine should be written by strictly following this sequence:

1. Select the ADC channel number and write it to the P0 which is connected to the address lines of the ADC.

2. Low to High transition at the ALE pin to latch the address into the ADC.

3. High to Low transition at the START pin to start ADC conversion.

4. Give a finite delay before next Instruction to allow the ADC enough time for conversion.

5. Set OUTPUT ENABLE pin of ADC 0808 to latch conversion result onto output lines.

6. Read ADC output into a variable

7. Reset OUTPUT ENABLE pin

ADC RESOLUTION:

The ADC resolution is defined as the lowest analog value that can be represented by a corresponding unique digital value.

For an n-bit ADC resolution is given by:

$$\textbf{Resolution} = \frac{\textbf{Vref}}{\textbf{2\^{}n-1}}$$

Where **Vref:** Reference voltage used in successive approximation technique

N: number of bits used by ADC to store digital data.

2.2.3.RTD(RESISTANCE TEMPRATURE DETECTOR):

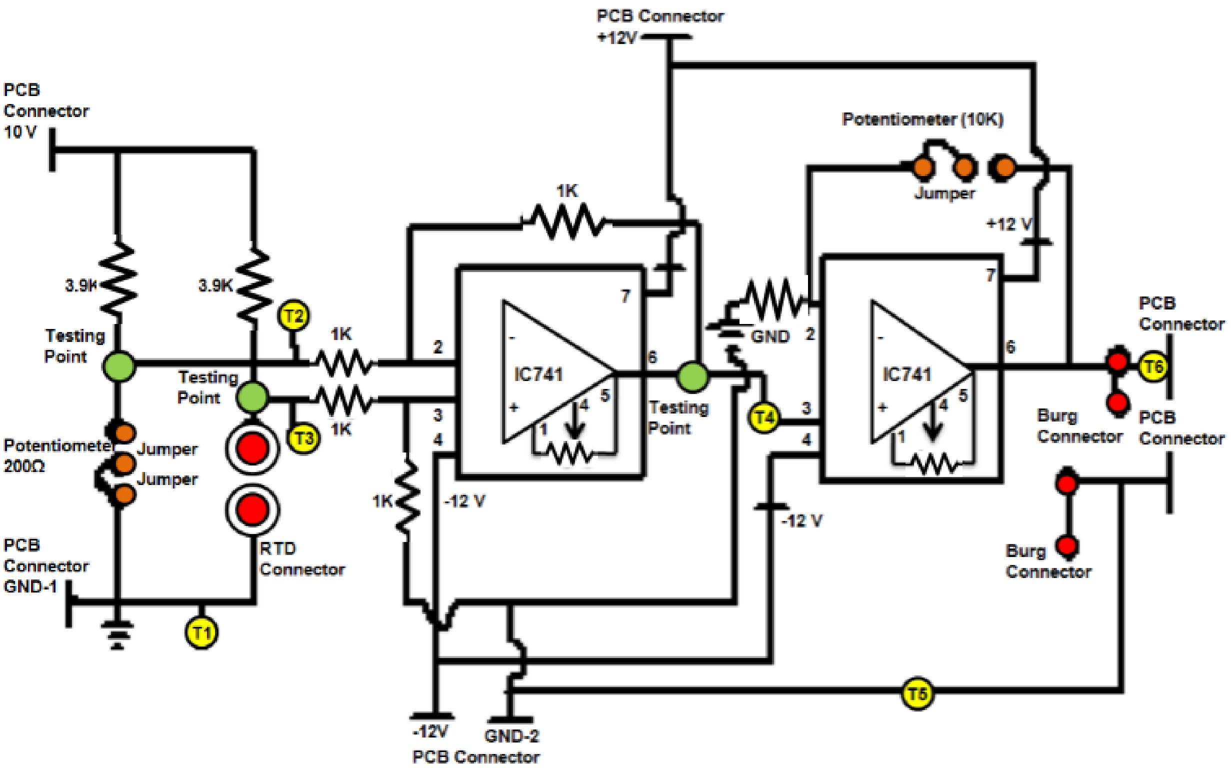


Fig.2.8.Circuit diagram of RTD sensor

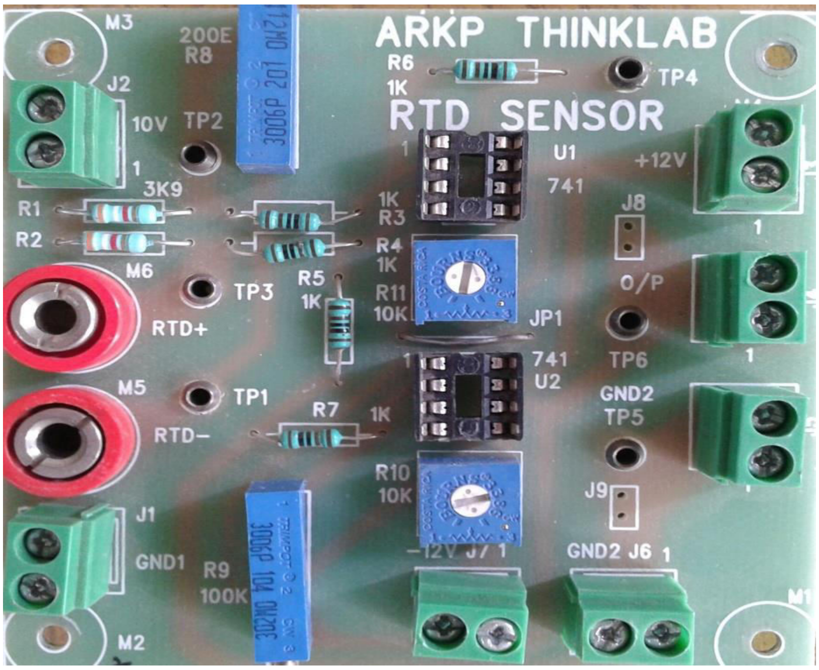


Fig.2.9.RTD Sensor PCB

RTD

This is the circuit arrangement of RTD, which is used to sense the temperature. Basically RTD are sensors used for measuring the temperature.RTD is made from pure material typically platinum nickel or copper. They are replacing thermocouples due to higher accuracy and repeatability.

As we know the temperature can be negative as well as positive. So we have to set the reference point that will determine the temperature is positive or negative. So first arm is to determine whether the temperature is positive or negative. The second arm is used to sense temperature from remote location. The difference of these 2 arms is given to the differentiating amplifier, where we get difference of 2 voltages.
Now the amplification stage will amplify the difference voltage in order to match the resolution. As resolution of ADC is 20mv and output generated by diff amplifier will be 1 or 2mv that does not gets detected by ADC. So to overcome this we use amplification.

Actual Temp	**Resistance(Ohm)**	**V2**	**V2 (mili)**	**Diffrence**	**Diffrence*21.04**	**Practical Temp**
0	100	0.25	250	0	0	0
1	100.39	0.25095053	250.9505	0.000951	0.0199992	1.02
2	100.78	0.25190088	251.9009	0.001901	0.039994501	2.04
3	101.17	0.25285104	252.851	0.002851	0.059985904	3.06
4	101.56	0.25380102	253.801	0.003801	0.07997341	4.08
5	101.95	0.25475081	254.7508	0.004751	0.099957021	5.10
6	102.34	0.25570042	255.7004	0.0057	0.119936737	6.12
7	102.73	0.25664984	256.6498	0.00665	0.13991256	7.14
8	103.12	0.25759907	257.5991	0.007599	0.15988449	8.16
9	103.51	0.25854812	258.5481	0.008548	0.179852529	9.18
10	103.9	0.25949699	259.497	0.009497	0.199816679	10.19
11	104.29	0.26044567	260.4457	0.010446	0.219776939	11.21
12	104.68	0.26139417	261.3942	0.011394	0.239733312	12.23
13	105.07	0.26234248	262.3425	0.012342	0.259685798	13.25
14	105.46	0.26329061	263.2906	0.013291	0.279634399	14.27
15	105.85	0.26423855	264.2386	0.014239	0.299579116	15.28
16	106.24	0.26518631	265.1863	0.015186	0.319519949	16.30
17	106.63	0.26613388	266.1339	0.016134	0.3394569	17.32
18	107.02	0.26708127	267.0813	0.017081	0.359389971	18.34
19	107.4	0.26800419	268.0042	0.018004	0.378808205	19.33
20	107.79	0.26895122	268.9512	0.018951	0.398733616	20.34
21	108.18	0.26989806	269.8981	0.019898	0.41865515	21.36
22	108.57	0.27084472	270.8447	0.020845	0.438572808	22.38
23	108.96	0.27179119	271.7912	0.021791	0.45848659	23.39
24	109.35	0.27273748	272.7375	0.022737	0.478396498	24.41
25	109.73	0.27365932	273.6593	0.023659	0.497792171	25.40
26	110.12	0.27460525	274.6052	0.024605	0.517694433	26.41
27	110.51	0.27555099	275.551	0.025551	0.537592825	27.43
28	110.9	0.27649655	276.4965	0.026497	0.557487347	28.44
29	111.28	0.27741768	277.4177	0.027418	0.576868032	29.43

30	111.67	0.27836288	278.3629	0.028363	0.596754918	30.45
31	112.06	0.27930789	279.3079	0.029308	0.616637937	31.46
32	112.45	0.28025271	280.2527	0.030253	0.636517091	32.48
33	112.83	0.28117314	281.1731	0.031173	0.655882806	33.46
34	113.22	0.2821176	282.1176	0.032118	0.675754332	34.48
35	113.61	0.28306188	283.0619	0.033062	0.695621996	35.49
36	113.99	0.28398177	283.9818	0.033982	0.71497652	36.48
37	114.38	0.28492569	284.9257	0.034926	0.734836563	37.49
38	114.77	0.28586943	285.8694	0.035869	0.754692747	38.50
39	115.15	0.28678879	286.7888	0.036789	0.774036088	39.49
40	115.54	0.28773216	287.7322	0.037732	0.793884658	40.50
41	115.93	0.28867535	288.6754	0.038675	0.813729373	41.52
42	116.31	0.28959418	289.5942	0.039594	0.833061542	42.50
43	116.7	0.29053701	290.537	0.040537	0.852898648	43.52
44	117.08	0.29145549	291.4555	0.041455	0.872223406	44.50
45	117.47	0.29239795	292.398	0.042398	0.892052909	45.51
46	117.85	0.29331608	293.3161	0.043316	0.91137026	46.50
47	118.24	0.29425818	294.2582	0.044258	0.931192164	47.51
48	118.62	0.29517596	295.176	0.045176	0.950502113	48.50
49	119.01	0.2961177	296.1177	0.046118	0.970316421	49.51
50	119.4	0.29705926	297.0593	0.047059	0.990126885	50.52
51	119.78	0.29797651	297.9765	0.047977	1.00942569	51.50
52	120.16	0.29889358	298.8936	0.048894	1.028720847	52.49
53	120.55	0.2998346	299.8346	0.049835	1.048519979	53.50
54	120.93	0.30075132	300.7513	0.050751	1.067807746	54.48
55	121.32	0.30169198	301.692	0.051692	1.087599296	55.49
56	121.7	0.30260835	302.6083	0.052608	1.106879678	56.47
57	122.09	0.30354865	303.5487	0.053549	1.12666365	57.48
58	122.47	0.30446467	304.4647	0.054465	1.145936651	58.47
59	122.86	0.30540461	305.4046	0.055405	1.16571305	59.48
60	123.24	0.30632028	306.3203	0.05632	1.184978674	60.46
61	123.62	0.30723577	307.2358	0.057236	1.204240659	61.44
62	124.01	0.30817518	308.1752	0.058175	1.224005755	62.45
63	124.39	0.30909032	309.0903	0.05909	1.24326037	63.43
64	124.77	0.31000529	310.0053	0.060005	1.262511348	64.41
65	125.16	0.31094416	310.9442	0.060944	1.282265152	65.42
66	125.54	0.31185878	311.8588	0.061859	1.301508767	66.40
67	125.92	0.31277323	312.7732	0.062773	1.320748748	67.39
68	126.31	0.31371156	313.7116	0.063712	1.340491269	68.39
69	126.69	0.31462566	314.6257	0.064626	1.359723892	69.37
70	127.07	0.31553959	315.5396	0.06554	1.378952886	70.35

Table 2.5.RTD PT100 Observation Table

CHAPTER-3

SOFTWARES

3.1.1. SCADA:

SCADA is not a specific technology but a type of application, any application that gets data about a system in order to control that system is a SCADA application.

What is SCADA?

- Supervisory Control and Data Acquisition.
- One Supervisory Controller (Master).
- Multiple Sub-Controllers (Slaves).
- It works like a supervisor who supervises the entire plant area.

Fig 3.1 SCADA control room

Why SCADA?

Previously without SCADA software, an industrial process was entirely controlled by PLC, CNC, PID & micro controllers having programmed in certain languages or codes.

These codes were either written in assembly language or relay logic without any true animation that would explain the process running.

It is always easy to understand the status of the process if it is shown with some animations rather than written codes**.**

3.1.2. FUNCTIONS OF SCADA

1. Data acquisition

2. Data communication

3. Data presentation

4. Control

3.1.2.1. DATA ACQUISITION:

SCADA system needs to monitor hundreds or thousands of sensors. For most analogue factors, there is a normal range defined by a bottom and top level e.g. temperature in a server room between 15 and 25 degrees Centigrade. If the temperature goes outside this range, it will trigger a threshold alarm. In more advanced systems, there are four threshold alarms for analogue sensors, defining Major Under, Minor Under, Minor Over and Major Over alarms.

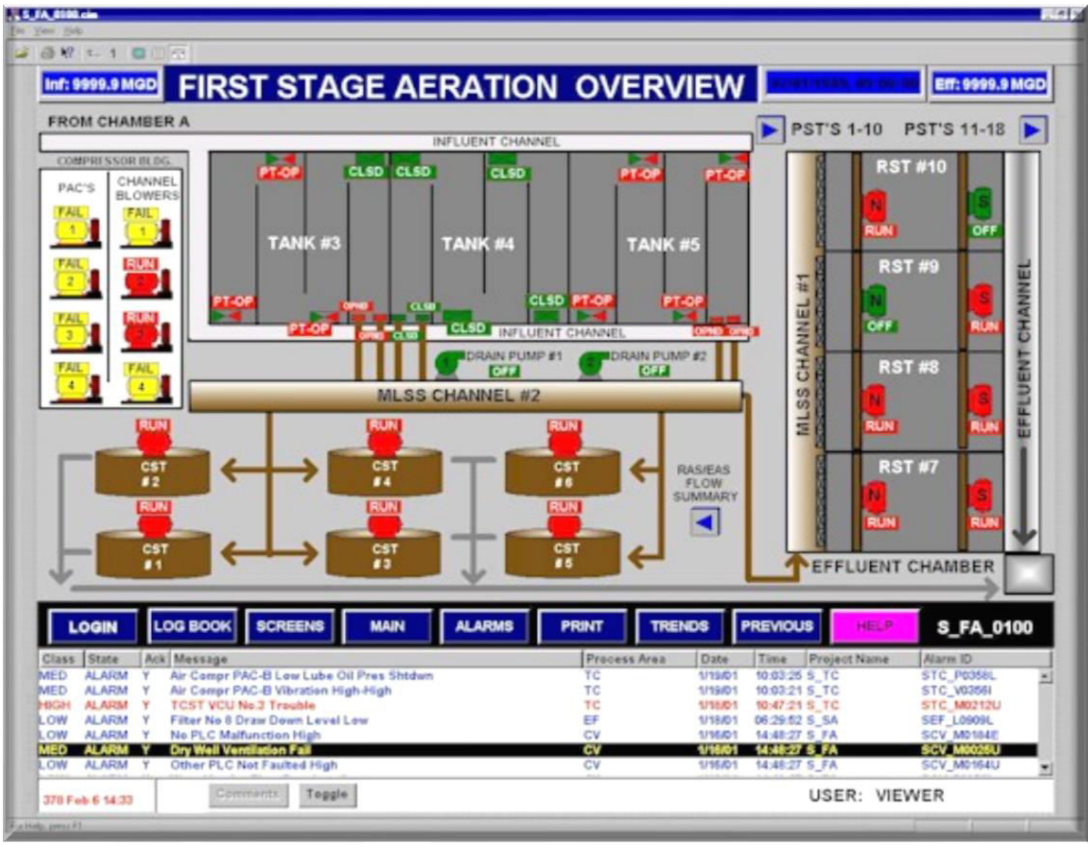


Fig 3.2.SCADA Data acquisition

3.1.2.2. DATA COMMUNICATION

A communications network is required to monitor multiple systems from a central location. Sensors and control relays can’t generate or interpret protocol communication - a remote telemetry unit (RTU) is needed to provide an interface between the sensors and the SCADA network.

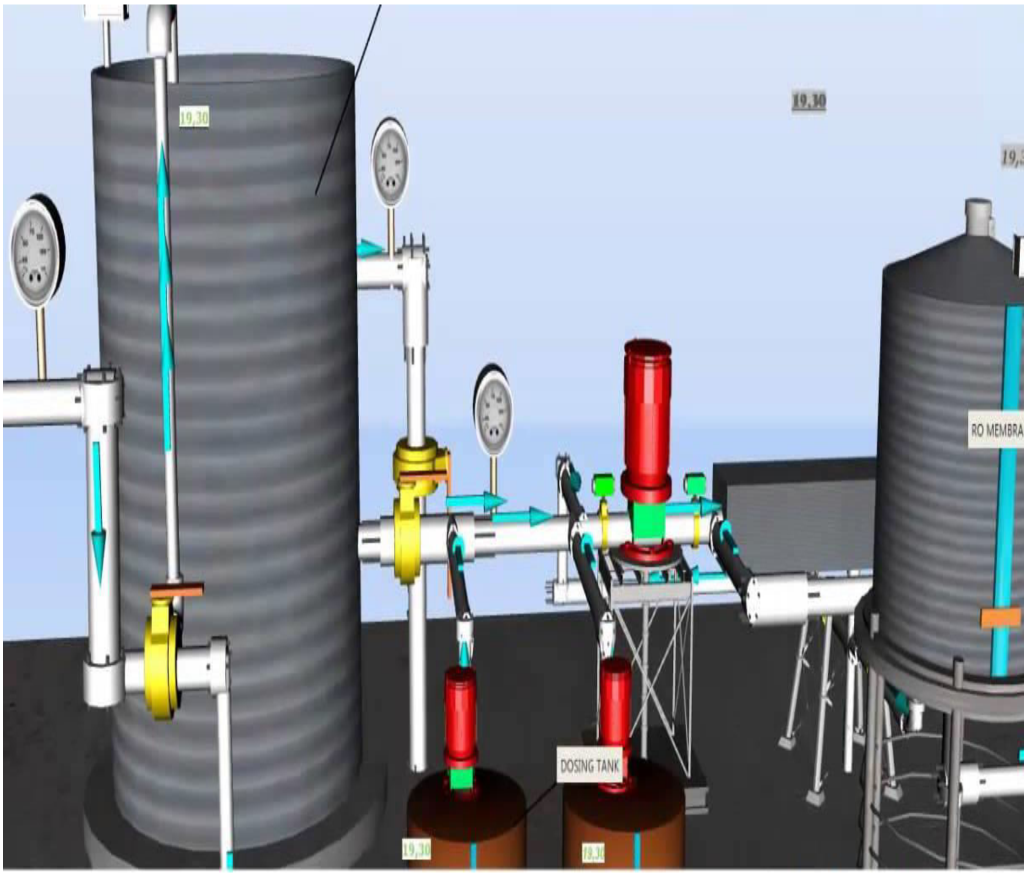


Fig 3.3 SCADA Data communication

3.1.2.3. DATA PRESENTATION

RTU encodes sensor inputs into protocol format and forwards them to the SCADA master. SCADA systems report to human operators over a master station, HMI (Human-Machine Interface) or HCI (Human-Computer Interface).

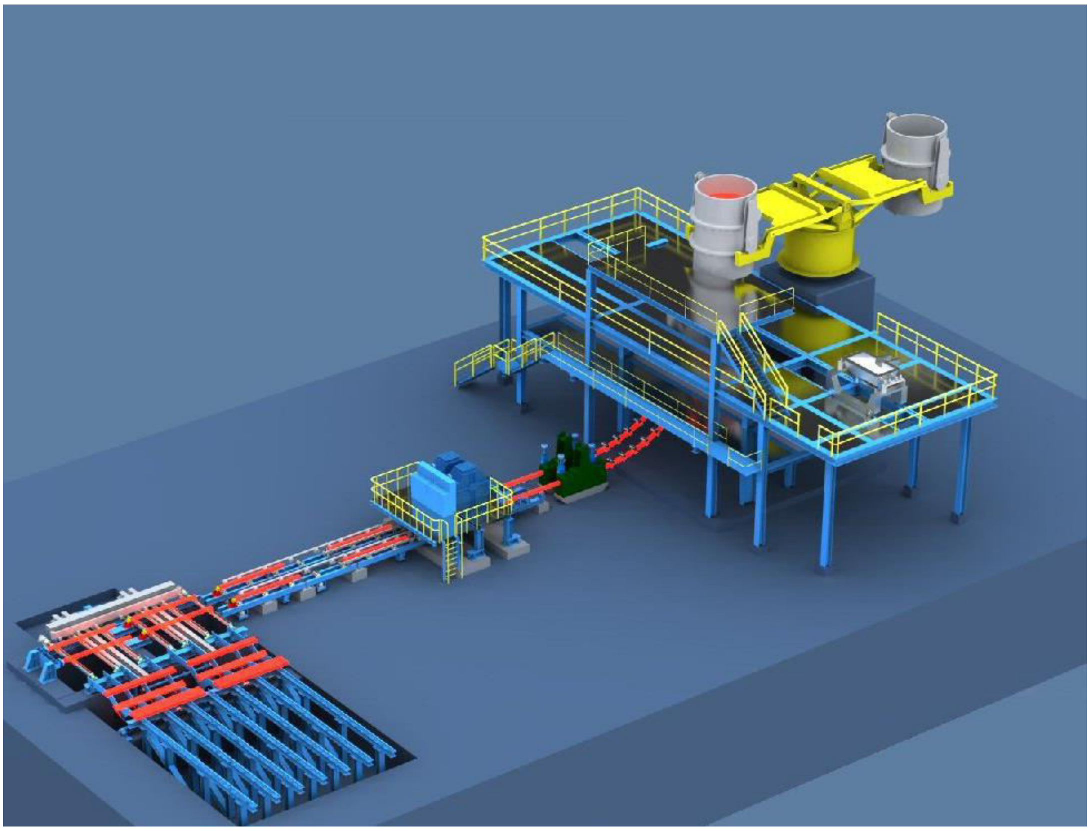
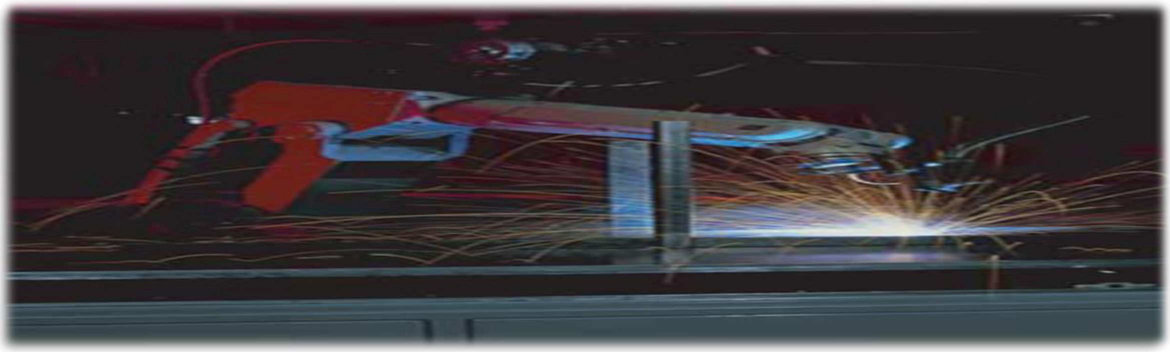


Fig 3.4 SCADA Data presentation

3.1.2.4. SCADA CONTROL

Manufacturing:



Buildings, facilities and environments:

Water and sewage:

3.1.2. MODSCAN32

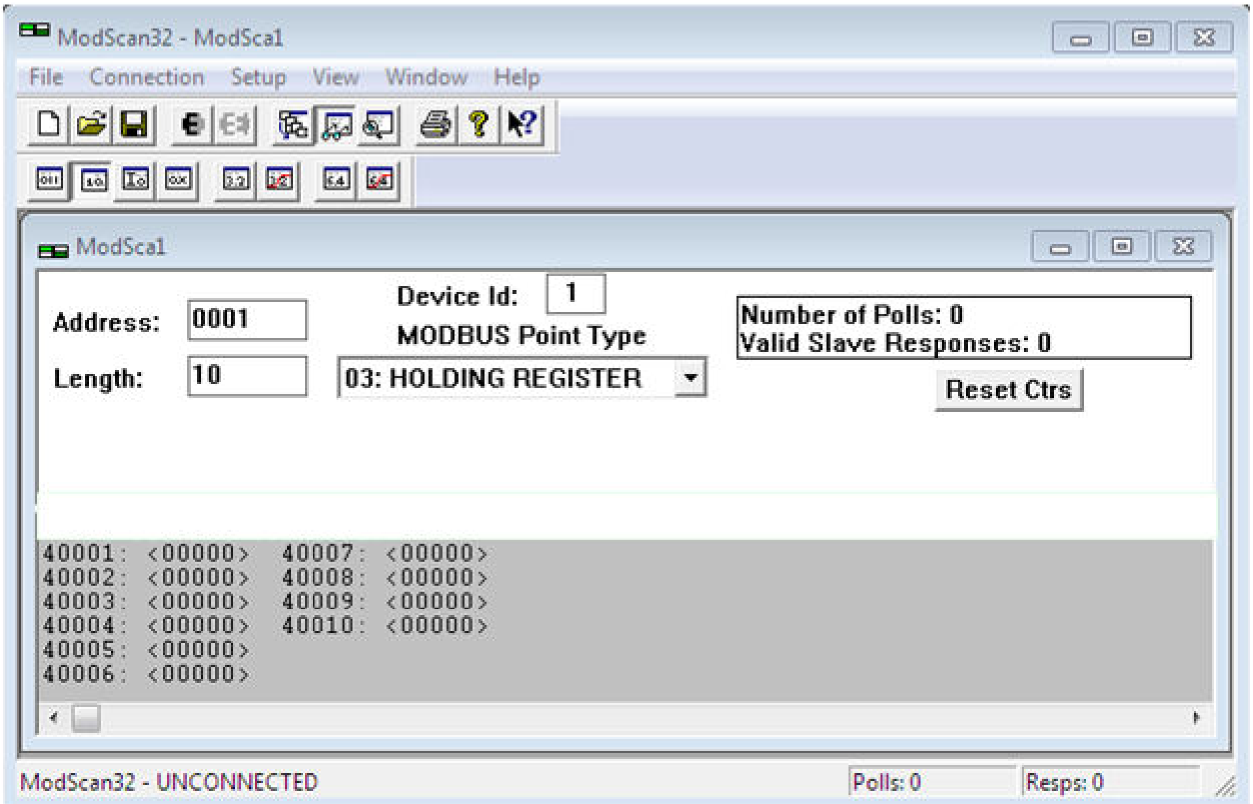


Fig.3.5. front panel of modscan32 software

MODSCAN DOCUMENT/VIEW ARCHITECTURE

ModScan utilizes the standard Windows Multiple-Document-Interface, (MDI), and architecture for displaying Modbus data to the user. Each document represents a series, (array), of Modbus data points identified by the following parameters:

Slave Device Address	Represents the physical device attached to the Modbus network
Data Type	Internal data representation, (i.e. input, coil, register)
Data Address	Point address within the device
Length	Number of points to scan/display

DATA DEFINITION SPLITTER VIEW

The upper half of each Document's View represents the data selected for display, (and possible capture to a historical data file). In most testing applications, the ModScan will only be connected to a single Modbus slave device, however, in a multidrop Modbus network; there may be several devices accessible from a single connection. The "Device Id" edit control allows you to specify the slave address for the source of the data. Likewise, edit controls are available to select the point type, data address, and number of data points to access.

DATA DISPLAY SPLITTER VIEW

As data is received from the slave device, it is displayed to the lower splitter view of the associated document. Any errors incurred during the exchange of information will be displayed on the first line. The font and colors used to display the data is configurable via the **View, Config** menu options.

Modbus register data may be displayed in any of the following formats:

Binary	Data displayed as 16 discrete values.
Decimal	Ranges from -32767 to 32768
Hexadecimal	0000-ffff
Floating Point	IEEE Standard Floating Point Notation (Requires two registers per value)
Word-Swapped Floating Point	Inverted Floating Point used by some processors
Double-Precision Floating Point	64-bit Floating Point Notation
Word Swapped Dbl Precision	Inverted 64-bit Float Values

CONNECTING TO A MODBUS NETWORK

ModScan may be used to obtain data from Modbus slave device connected to the PC in one of three basic physical arrangements. The most common connection is via any one of the four available PC serial COM ports. ModScan uses the standard Win32 software drivers for communication with the COM ports, thereby providing support for any hardware serial boards which may be installed in the Windows operating system, (including RS-232, RS-485, etc.). You have complete control over the operating characteristics of the serial connection by selecting the appropriate baud rate, parity, and control line, (handshaking), properties to match the slave device(s).

CHAPTER-4

PROTOCOL

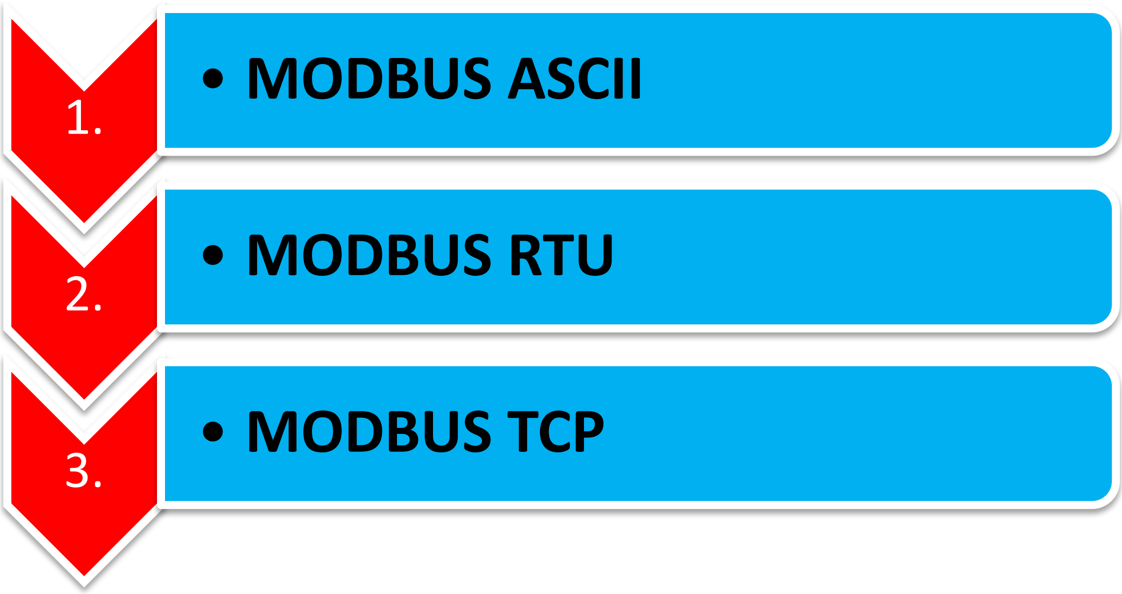
4.1. MODBUS

4.1.1. NTRODUCTION

- MODBUS is a master-slaves communication protocol where a master can communicate with one or more slaves.
- The master controls the complete transmission and the connected devices are slaves which send data on master request.
- When the master wants some information from slaves, it sends a query (message) that contains slave ID, function codes, data and CRC16.

Slaves can’t initiate the data transfer, they can only respond on master query.

4.1.2. MODBUS TYPES



4.1.2.1. MODBUS ASCII:

All MODBUS messages are sent in the same format. The only difference among the three MODBUS types is in how the messages are coded. Therefore, MODBUS ASCII is the slowest of the three protocols, but is suitable when telephone modem or radio (RF) links are used. This is because ASCII uses characters to delimit a message. Because of this delimiting of the message, any delays in the transmission medium will not cause the message to be misinterpreted by the receiving device. This can be important when dealing with slow modems, cell phones, noisy connections, or other difficult transmission mediums.

4.1.2.1. MODBUS RTU:

In MODBUS RTU, data is coded in binary, and requires only one communication byte per data byte. This is ideal for use over RS232 or multi-drop RS485 networks, at speeds from 1,200 to 115Kbaud. The most common speeds are 9,600 and 19,200 baud. MODBUS RTU is the most widely used industrial protocol, so most of this paper will focus on MODBUS RTU basics and application considerations.

4.1.2.2. MODBUS TCP:

MODBUS/TCP is simply MODBUS over Ethernet. Instead of using device addresses to communicate with slave devices, IP addresses are used. With MODBUS/TCP, the MODBUS data is simply encapsulated inside a TCP/IP packet. Hence, any Ethernet network that supports TCP/IP should immediately support MODBUS/TCP. More details regarding this version of MODBUS will be covered in a later section entitled “MODBUS over Ethernet.”

4.1.3. MODBUS NETWORK STRUCTURE

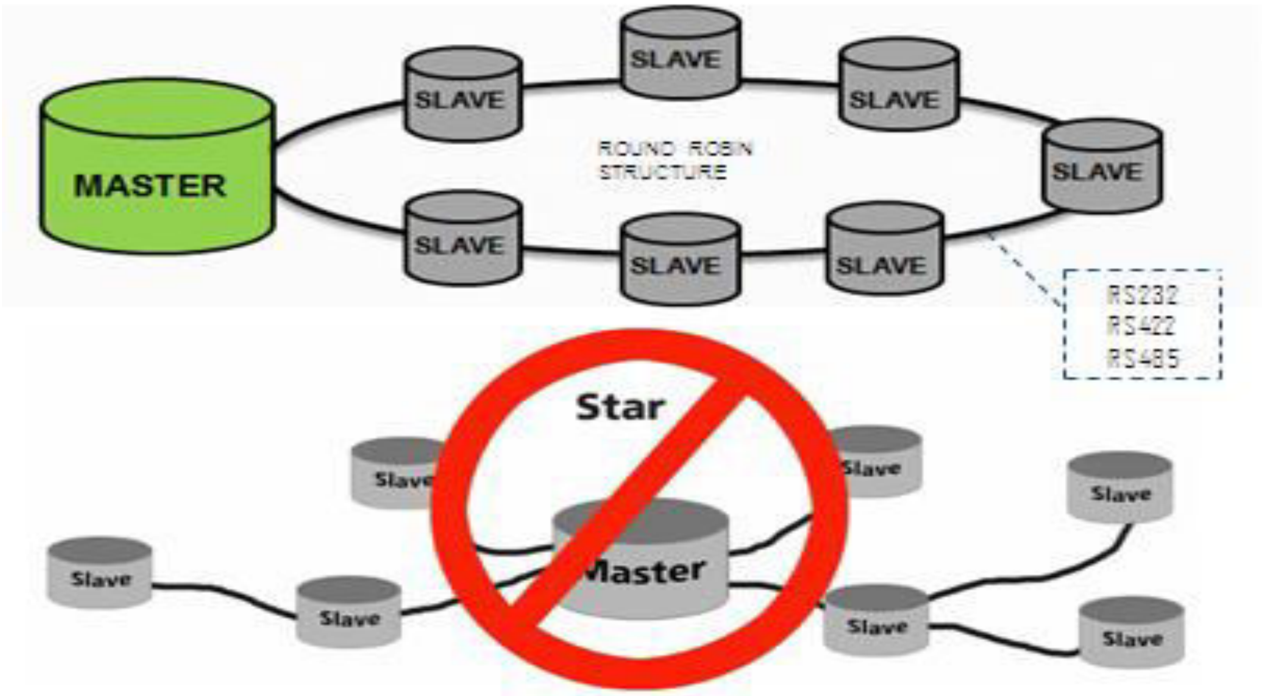
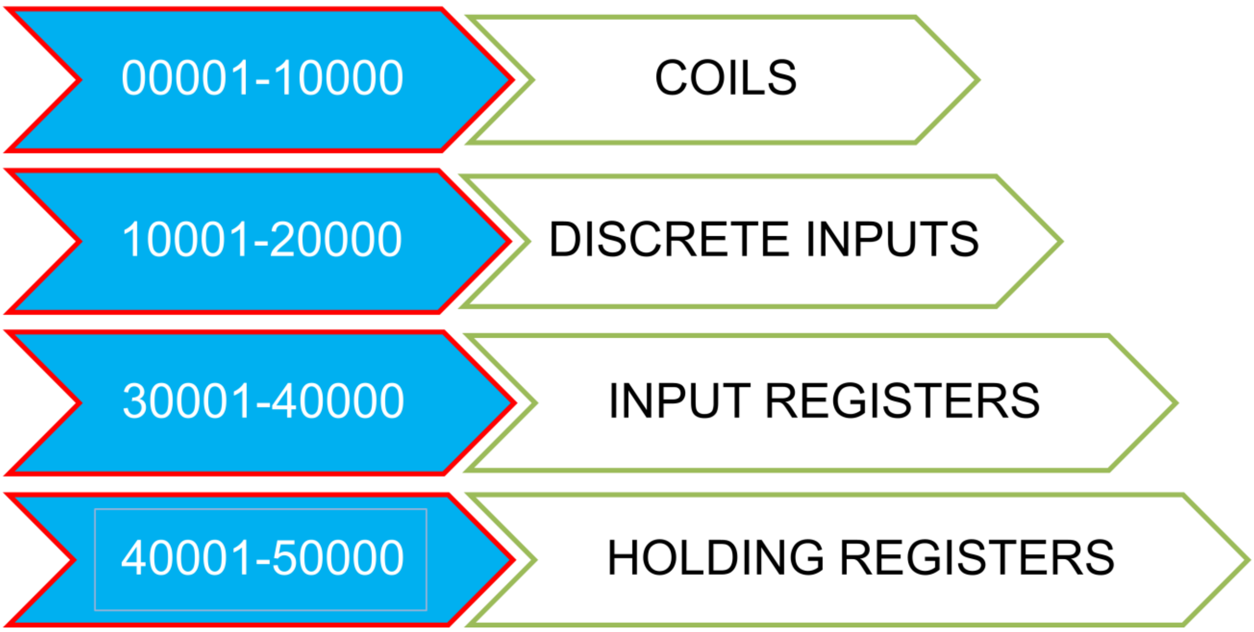


Fig.4.1.MODBUS network structure

4.1.4. MODBUS MEMORY ADDRESSING



4.1.4.1. **COILS READ/ WRITE (00001-10000):**These are read write Boolean values. They are typically used to represent outputs or internal bits which are read and written by the user.

4.1.4.2. **DISCRETE INPUT/READ ONLY (10001-20000):** These are read only Boolean values. They are typically used to represent sensor inputs and other Boolean values which are read but not written by the user.

4.1.4.3. **INPUT REGISTER READ ONLY (20001-30001):** These are read only registers and these are two byte register (byte). Master read these registers, these register represents the value of physical parameter which master wants to read from slave.

4.1.4.4. HOLDING REGISTER READ/WRITE (40001-50000): These are read/write 16 bit registers. Master sends 16 bit value to the slave to make changes in a process; e.g. tochanges the set point value.

4.1.5. MODBUS FRAME STRUCTURE

To communicate with devices (slaves), the master sends aquery (message) containing:

- Slave ID
- Function Code
- Data
- CRC Check

<table>
<tr><td rowspan="3">Slave ID (1 Byte)</td><td rowspan="3">Function Code (1 Byte)</td><td colspan="4">DATA</td><td colspan="2" rowspan="2">CRC16 (2 Byte)</td></tr>
<tr><td colspan="2">Start Address</td><td colspan="2">Number Of States</td></tr>
<tr><td>High-Byte (1 Byte)</td><td>Low-Byte (1 Byte)</td><td>High-Byte (1 Byte)</td><td>Low-Byte (1 Byte)</td><td>Low-Byte</td><td>High-Byte</td></tr>
</table>

Fig 4.2 MODBUS Frame structure

SLAVE ID:

The first byte of the frame is slave id. Slave ID is basically an 8bit address. In a MODBUS protocol Each and Every Slave have unique ID in the range between 1 to 247.No two slaves has a same ID in a network.ID “0” is used for broadcasting, all slaves will receive the master request

When Master Wants to communicate with Slave ,then the master sends the Frame to all Slaves, all the slaves will receive this frame and the one who matches this id that slave get activated and others will deactivated.

FUNCTION CODE:

Modbus commands are known as *functions*. These are simple commands to read or write, they are numbered as 01, 02, 03, 04, etc.

for example, function “01" will read one or more coils. Function "15" will write to one or more coils.

There are 255 function codes are defined in Modbus, but following are the most commonly used.

COMMANDS	FUNCTION CODES
01	Read multiple coils.
02	Read multiple discrete inputs.
03	Read multiple holding registers.
04	Read multiple input registers.
05	Write single coil.
06	Write single holding register.
15	Write multiple coils.
16	Write multiple holding registers.

Table 4.1

Function codes of MODBUS

DATA:

This section of protocol frame contains information which is to be transmitted by a master.

This information are divided into two parts that is starting address from where master wants to start reading or writing of data and second section is no of states means how many states are to be read or write by a master .

DATA			
Start Address		Number Of States	
High-Byte	Low-Byte	High-Byte	Low-Byte

Fig 4.3 MODBUS data field

CRC (CYCLIC REDUNDANCY CHECK):

It is used for error correction and detection. When a frame is transmitted, two byte value is added, that two byte value is generated using one algorithm that we called CRC16.

When receiver receives the frame, the receiver will calculate CRC16 of received frame using same algorithm then it will compare the CRC16 of master with the CRC16 of slave. If the two are match it means the frame is properly received.

CHAPTER-5

COMMUNICATION STANDARDS

5.1.1. RS232

RS-232 is a popular communication interface for connecting modems and data acquisition devices (i.e. GPS receivers, electronic balances, data loggers) to computers.

RS-232 can be plugged straight into the computer's serial port (known as COM port).

It uses serial communications where one bit is sent along a line, at a time. This is different from parallel communication which sends one or more bytes.

The time diagram of the typical signal used to transfer character 'A' (ASCII:6510 or 0x41) from device A to device B is given:

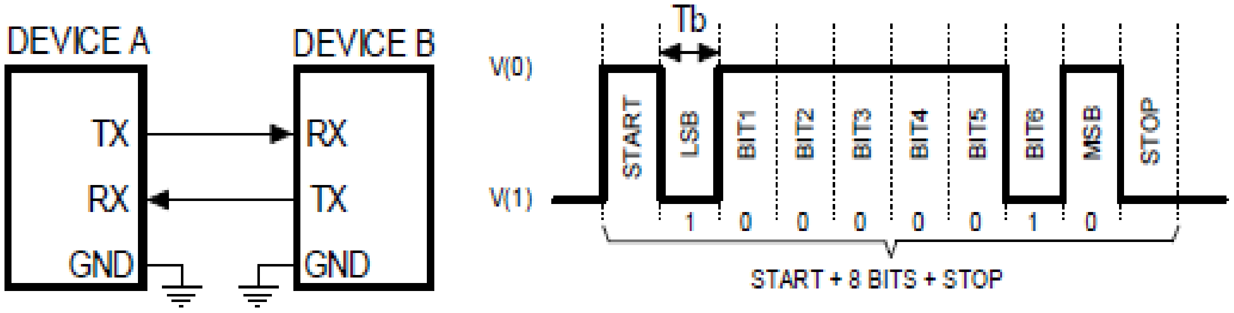


Fig 5.1 Timing diagram of RS232 signal

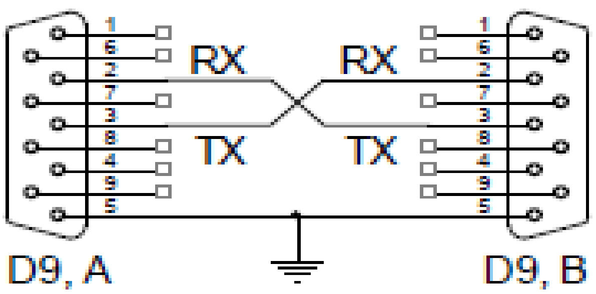


Fig 5.2 Connections of RS232

5.1.2. RS485

What is RS-485?

RS-485 is an EIA standard interface which is very common in the data acquisition world. It allows high data rates communications over long distances in real world environments and supports multiple devices communication.

Speed of RS-485

RS-485 was designed for greater distance and higher baud rates than RS-232.According to the standard, 100kbit/s is the maximum speed and distance up to 4000 feet (1200 meters) can be achieved.

RS-485 provides Half-Duplex, Multidrop communications over a single twisted pair cable. The standard specifies up to 32 drivers and 32 receivers can share a multidrop network.

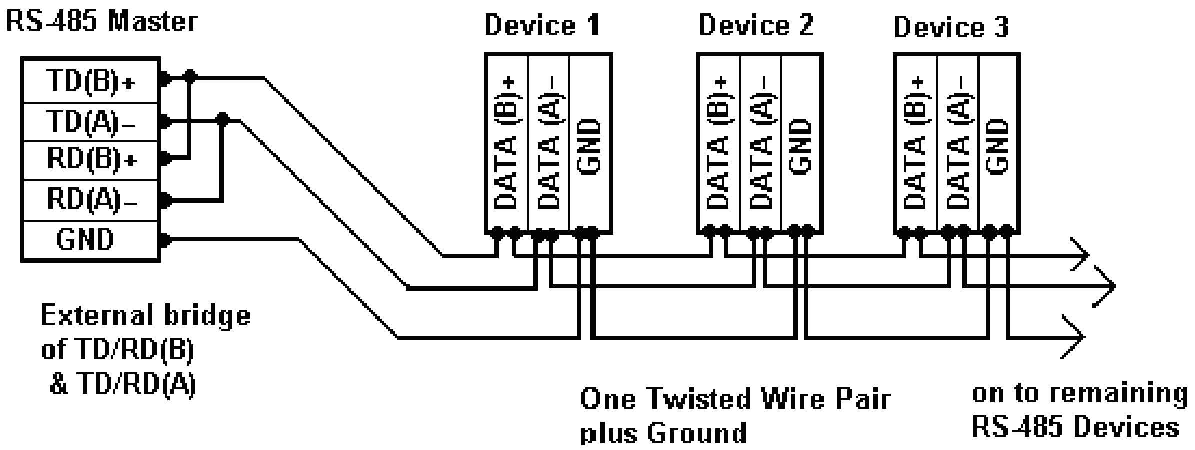


Fig 5.3 Multiple device connection of RS485

RS232 VS RS485

The architectural difference between RS-232 and RS-485 is that 232 is a bi-directional point to point link, whereas 485 is a single channel bus.

Electrically, each RS232 signal uses a single wire with symmetric voltages with common ground wire. RS485 uses two wires to carry the single signal differentially.

Only one device on a RS485 bus can transmit at a time, whereas RS232 has a limitation that it is a peer-to-peer link.

CHARACTERISTICS	RS-232	RS-485
Mode of Operation	SINGLE-ENDED	DIFFERENTIAL
Total Number of Drivers and Receivers on One Line RECEIVER	1 DRIVER 1 RECEIVER	32 DRIVER 32
Maximum Cable Length	50 FEET	4000 FEET
Maximum Data Rate @Max length	20kb/s	100kb/s
Driver Output Signal Level (Loaded Min.) Loaded	+/-5V to +/-15V	+/-1.5V
Driver Output Signal Level (Unloaded Max) Unloaded	+/-25V	+/-6V
Driver Load Impedance	3kΩ to 7kΩ	54Ω
Slew Rate (Max.)	30V/µS	N/A
Receiver Input Voltage Range	+/-15V	-7V to +12V
Receiver Input Sensitivity	+/-3V	+/-200mV
Receiver Input Resistance	3kΩ to 7kΩ	≥12kΩ

Table 5.1 Comparison between RS232 & RS485

5.1.3. RS232 TO RS485 CONVERTOR

The RS232 to RS485 converter comes in many models; sizes and shapes, here are a few examples:

Fig 5.4 Types of RS232 to RS485 Convertor

RS232 to RS485 converters are mostly used in industrial and commercial environments. The reason is that the RS485 converter can be used for multi-drop networks, meaning that you for example can connect multiple RS485 devices to one computer. Up to 32 devices can be connected in one network to communicate on a single pair of wires (plus a ground wire), and the number can even be increased by using RS485 repeaters.

This type of communication is called half-duplex communication. Most quality RS485 converters can communicate in this manner of up to 4000 feet (1200 meters). The RS485 converter is also fairly resistant to noise which is one more reason why it is favored in industrial environments.

Examples of uses for a RS232 to RS485 converter could be for connecting cameras, scales, meters, scanners, PLC's or most other industrial equipment to a computer.

5.1.3.1. RS232 to RS485 Convertor Circuit Schematic

Serving as an example, below schematic shows what a typical RS232 to RS485 converter circuitry looks like.

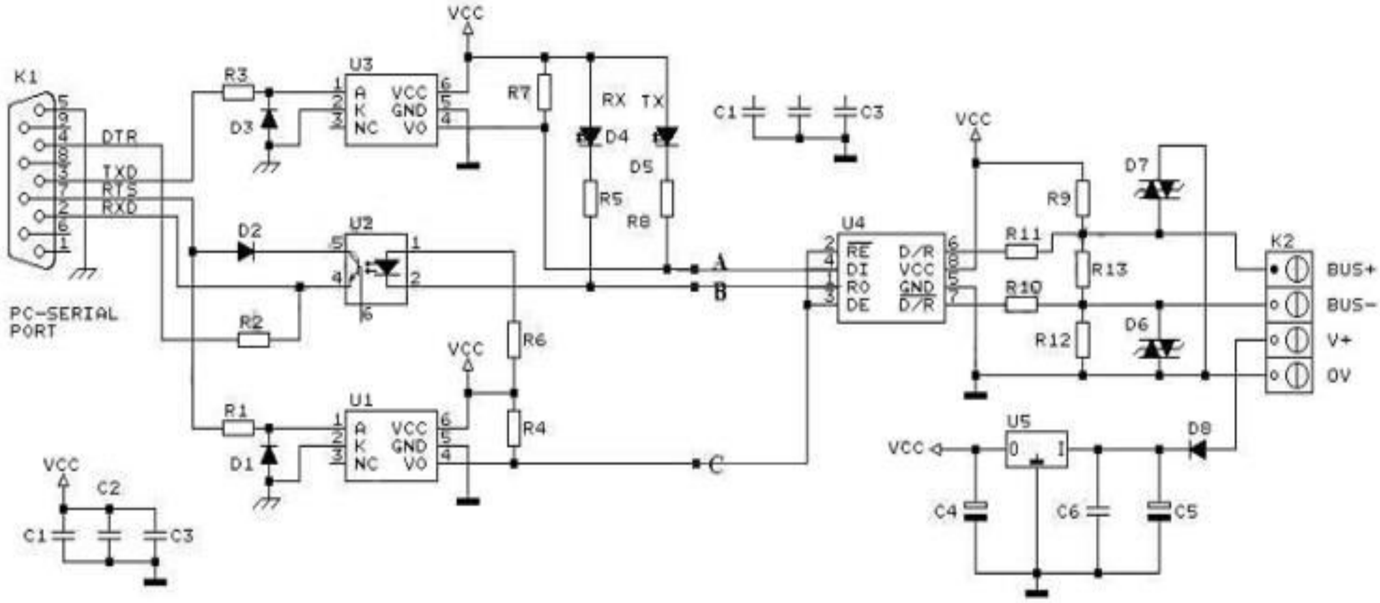


Fig 5.5 Internal circuit schematic of RS232 to RS485 convertor

CHAPTER-6

CIRCUIT IMPLEMENTATION ON PCB

6.1. What is PCB?

PCB means printed circuit board. It is designed by certain fabrication process. A conductive coating material is given on an insulating base material. This is also called as copper clad. Then by using art work the circuit can be drawn to mount the desired components as per circuit. PCB is also called as Printed Wired Board (PWB).

6.2. Why we need PCB?

Using PCB we can reduce the size of apparatus to be designed. A PCB is necessary because it holds every component firmly in its place and thus increases circuit reliability. Soldering becomes easy because of PCB.

6.3. Major types of PCB:

Single sided Board (SSB): The printed circuit board which has only one track (copper) layer is called SSB.

Double Sided Board (DSB): The printed circuit board which has two track (copper) layer is called DSB.

In our project we have used single sided PCB that is it contains only one copper track or layer.

6.4. VARIOUS STEPS INVOLVED IN FABRICATION OF PCB:

- Art Work Preparation.
- Printing.
- Etching.
- Drilling.
- Soldering.

6.4.1. ART WORK PREPARATION

Its purpose is to develop a layout for the final circuit board. It is the first and most important step as the placement of various components and conductor thickness is decided in this step. Art work is always prepared from the component side of PCB, taking the help of circuit diameter.

6.4.2. PRINTING

This involves transferring of the art work into the copper clad. These are three methods of printing.

- Direct resist method.
- Photo resists method.
- Screen printing.

Out of this we have used screen printing method. In this method the resist ink applied through a stencil or mask to the surface of the blank circuit board.

The stencil is produced and attached to a fine mesh, metal, nylon, polyester or silk screen. The resist ink is force through the opening in the stencil onto the surface of the blank board. This process produced a positive of the copper foil. When dry, the board is ready for etching.

Following equipment's are used for screen printing process:

Photosensitive film, developer, blackout solution, a rubber squeeze, the resist paint or ink.

6.4.3. ETCHING

This step involves removal of unwanted copper from copper clad. The most common etchant used is ferric chloride. The copper clad is dropped in solution. After an hour, the PCB is taken out and washed in clean water.

6.4.4. DRILLING

In this step drilling of PCB is done after removing of etch resist tape and pads. The holes are drilled according to the diameter of respective component lead.

A whole stack of board can be drilled using various jigs and bushes. The speed of drill is an important consideration with maintaining the size and location of hole with required tolerance and maintaining deformity at their edges. When holes are drilled in PCB, the laminate is uncovered in PTH.

6.4.5. SOLDERING

The last step involves soldering various components on the PCB.

Soldering is a method of joining two parts or more than two parts of metal. Fusible alloy metal alloys are used for this purpose. Prior to soldering the metal portion are to be joined and the solder must be heat. Since, the solder has a much lower melting point than the metal to be joined; it melts, while the metals remain hard. At the place where the molten solder comes in contact with hard metal, complex physiochemical process takes place.

6.5. PRECAUTIONS

- Number of jumper wires should be Minimum.
- Heat sensitive components must be kept away from heat producing ones.
- To reduce crosstalk & electro-magnetic interference all unused copper surface are connected to ground.
- Low power and high power level wires should be twisted outside the PCB to protect the circuit from electro-magnetic coupling.
- Heat sink should be connected to regulate IC, because it produce large amount of heat.
- Sufficient test point must be given and components have easy accessibility for replacement.
- Mechanical consideration has to be taken into account.
- Power shielding to be done across the pump, as it generates spikes.

CHAPTER-7

7.1. PROJECT HARDWARE

Fig 7.1 Actual Project Hardware Picture

7.2. RESULTS

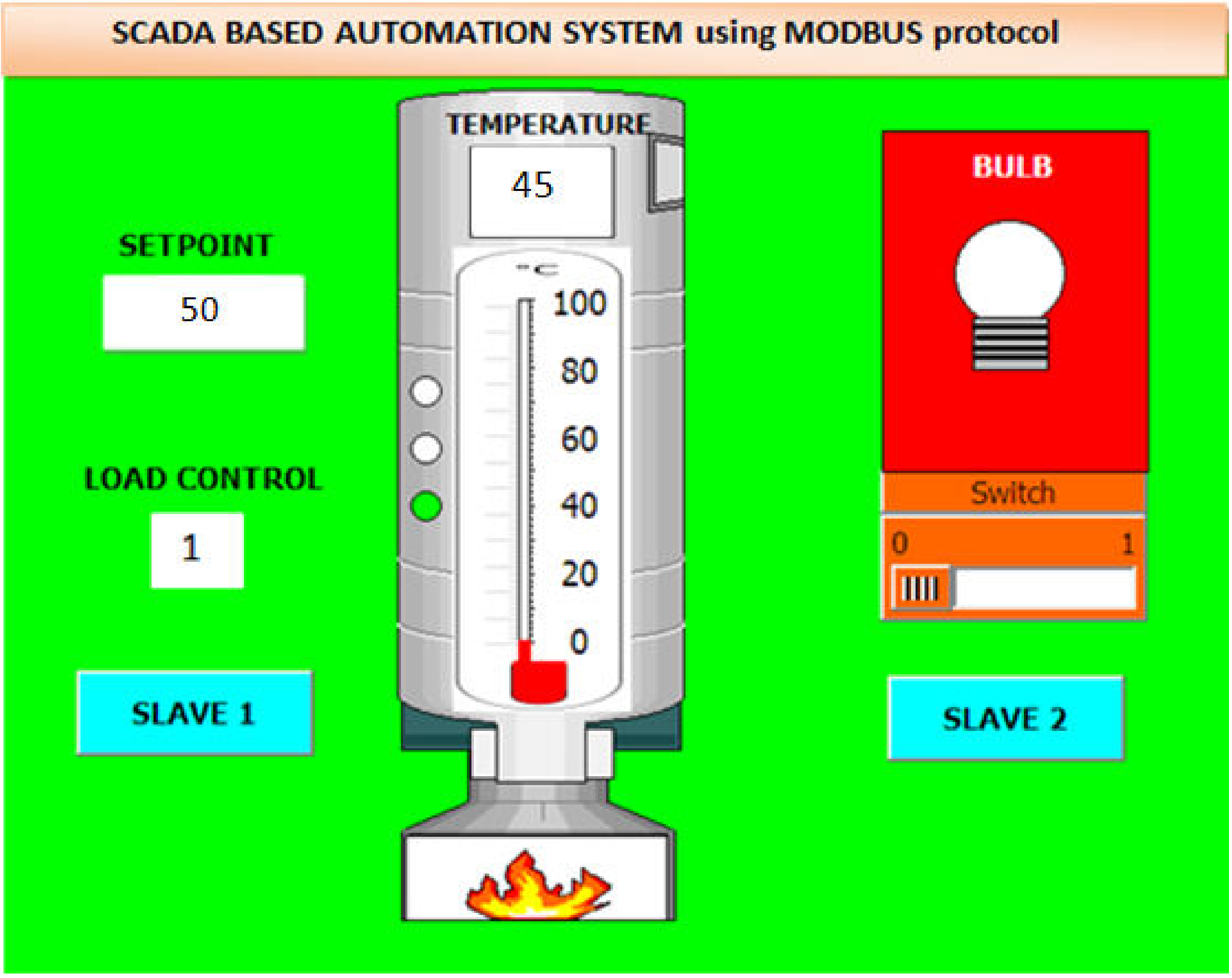


Fig 7.2 Front Panel Screen Shot of SCADA Software

CHAPTER-8

8.1. APPLICATIONS

1) In automation industries.
2) In chemical laboratories.
3) In production factories.
4) In power plants.

CONCLUSION

Group project is very important tool in brushing our knowledge and amalgamating theoretical knowledge with practical knowledge. While working on the project, we got to know latest technologies, various devices and how can implement them using various techniques, working in a group help us to know team importance, team spirit, working in a team would be useful for us in future tasks to be undertaken.

Apart from temperature controlling, the project is very useful in industrial Area for temperature controlling, we hope that our project proves Beneficial to industries automation.

REFERENCES

1. Avhad, Mayur, Vinit Divekar, Harshad Golatkar, and Sanket Joshi. "Microcontroller based automation system using industry standard SCADA." In *India Conference (INDICON), 2013 Annual IEEE*, pp. 1-6. IEEE, 2013.

2. Shankar, K. Gowri. "Control of boiler operation using PLC–SCADA." In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 2, pp. 19-21. 2008.

3. Rashid, Muhammad Harunur. *Power electronics: circuits, devices, and applications*. Pearson Education India, 2003.

4. Ahmed, Musse Mohamud, and Wai Lian Soo. "Supervisory control and data acquisition system (SCADA) based customized remote terminal unit (RTU) for distribution automation system." In *Power and Energy Conference, 2008. PECon 2008. IEEE 2nd International*, pp. 1655-1660. IEEE, 2008.

CHAPTER-9

9.1. PROJECT PAPER

Presented a paper on the topic "**SCADA BASED AUTOMATION SYSTEM USING MODBUS PROTOCOL**" at the National Conference on Computer, Electrical and Electronics Engineering (IETE cynosure, NCCEE-2016) held at Dr. BATU, Lonere in association with IETE- Navi Mumbai center during April 23rd-24th 2016

SCADA Based Automation System Using MODBUS Protocol

Avesh Lakha [1] & Saifuddin Dandoti [2] & Gulam Ron [3] & Uzair Khan [4]

A-I-Kalsekar Technical Campus, Electronics and Telecommunication Department, New Panvel.
SMDandoti95@gmail.com, owaislakha@gmail.com.

***Abstract*—Basically SCADA is the acronym stands for Supervisory Control And Data Acquisition system. It acquires the real time data from the device located at remote places and perform supervisory control on it. Now a days in different Industrial environments prefers the Automation of the Systems. SCADA is use to perform these different automated functions. This paper focuses on conversion of manually operated Temperature Controller System & Bulb towards fully automated system by using microcontroller & peripherals like RTD, ADC & Relay Circuit and monitors the whole process using SCADA.**

Software Tools Used

WinCC flexible 2008, Keil_v5, ModScan64, Flash Magic.

***Keywords*—Automation, MODBUS, SCADA.**

I. INTRODUCTION

In this project we are designing a system in which we are monitoring & controlling the temperature of Heater and also controlling the Bulb by switching it ON/OFF and monitoring that process on screen using SCADA software.

Our project consist of three major parts:

- SCADA as a Monitoring software.
- Devices such as Microcontroller, ADC, RTD, relay Circuit as a Hardware interface.
- Heater setup as a slave 1 and Bulb as a slave 2.

They all are connected serially in round robin manner and communicate with each other using MODBUS protocol. MODBUS is a "master-slave" system, where the master communicates with one or multiple slaves. The master typically is a Microcontroller.The three most common MODBUS versions used today are:

- MODBUS ASCII
- MODBUS RTU
- MODBUS/TCP

All MODBUS messages are sent in the same format. The only difference among the three MODBUS types is in how the messages are coded [1].

II. DESIGN DESCRIPTION

The Block diagram of the project is mentioned below in fig-1.

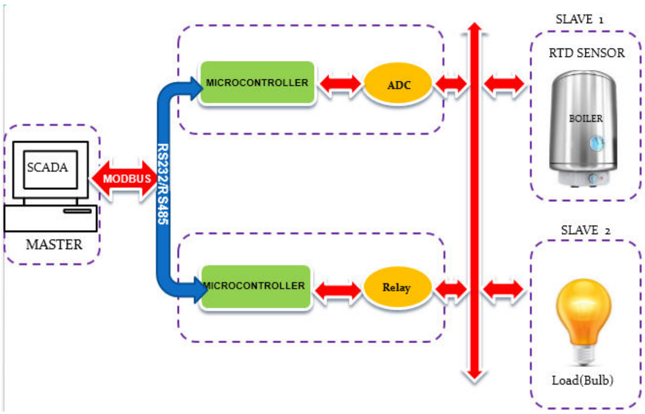


Fig. 1: Block Diagram of SCADA Based AutomationSystem

As from the Block diagram above, it is pretty much evident that the project is distributed into the following major sections,

1. Temperature Sensor from Industrial Environment RTD (PT100): For obtaining the temperature information, RTD is installed in any industrial environment, such as on head of a diesel engine or inside a refrigerator or in an incubator etc. The Range of measurements that can be made by the RTD sensor is between is -200 to 500 C.

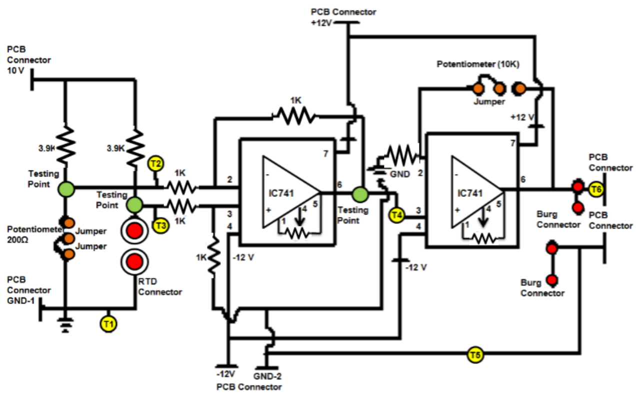


Fig. 2: Circuit Diagram of RTD for sensing the Temperature

As the temperature can be negative as well as positive. So we have to set the reference point that will determine the temperature is positive or negative. So first arm is to determine whether the temperature is positive or negative. The second arm is used to sense temperature from remote location. The difference of these 2 arms is given to the differentiating amplifier, where we get difference of 2 voltages. Now the amplification stage will amplify the difference voltage in order to level operation for ADC.

2. ADC0808: The ADC0808 is an 8-bit analog to digital converter with an 8 channel analog multiplexer. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals. The device eliminates the need for external zero and full-scale.

3. Microcontroller (AT89c51): A microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.

III. COMMUNICATIONS STANDARDS

RS-232 is a popular communication interface for connecting modems and data acquisition devices (i.e. GPS receivers, modems) to computers. It supports single device communication.

RS-485 is an EIA standard interface which is very common in the data acquisition world. It allows high data rates communications over long distances in real world environments and supports multiple devices communication.

IV. SYSTEM PROTOCOL

MODBUS is a master-slaves communication protocol where a master can communicate with one or more slaves [1]. The master controls the complete transmission and the connected devices are slaves which sends data on master request. When the master wants some information from slaves, it sends a query (message) that contains slave ID, function codes, data and CRC16.Slaves can’t initiate the data transfer, they can only respond on master query.

V. SOFTWARE DISCRIPTION

SCADA stands for Supervisory Control and Data Acquisition.

In order to automate a Power plant and minimize human intervention, there is a need to develop a SCADA system that monitors the plant and helps reduce errors caused by humans [2]. While the SCADA is used to monitor the system, Microcontroller is used for internal storage of instruction for the implementing function such as logic, sequencing, timing and Counting. The data acquisition is accomplished firstly by the RTUs scanning the field inputs connected to the RTU (it may be also called a Microcontroller). SCADA function is to control process equipment at the remote site, acquire data from the equipment, and transfer the data back to the central SCADA system.

The software simulation of this project was carried out in WinCC flexible 2008.

VI. RESULTS

Sensors collect the real time environmental parameters from the field. It will provide this information to Controller circuitry. For example set point given for the temperature is 50 degree Celsius. Temperature sensor will continuously provide the values .If temperature exceeds 50 degree Celsius, controller trigger the driver circuitry to take necessary action. In these case, the action may be switching off the heater to control the temperature.

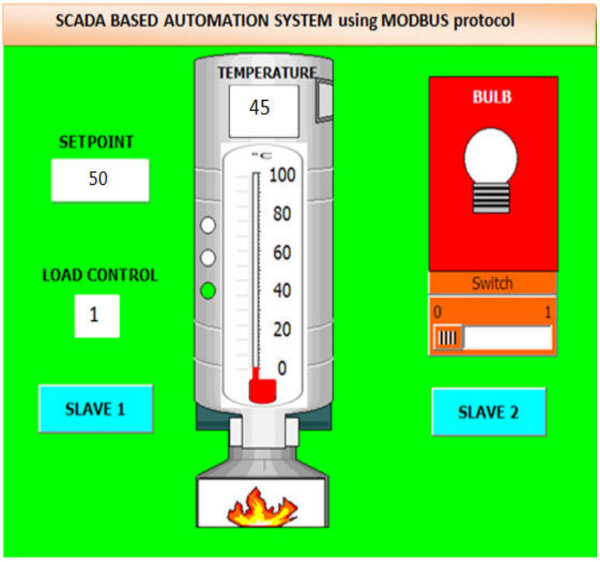


Fig. 3: SCADA Screen resuls

VII. CONCLUSIONS

Apart from temperature controlling, the project is very useful in industrial Area for temperature controlling, we hope that our project proves Beneficial to industrial automation So our prime concern is to demonstrate industrial automation of temperature monitoring & controlling and load controlling system using little hardware and more refined software details.

VIII. ACKNOWLEDGEMENT

I am grateful to Prof. Mujib Tamboli and I acknowledge with gratitude to my supervisor Prof. Mujib Tamboli, Department of Electronics & Telecommunication Engineering, and all staff members for his innovative thinking, continuous guidance, genius role and encouragement throughout the whole project period.

REFERENCES

[1] McconahaV, Jim. "Using Modbus for process control and automation."measurements 100(2011):1."

[2] Shankar, K. Gowri. "Control of boiler operation using PLC–SCADA." In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 2, pp. 19-21. 2008.

[3] Rashid, Muhammad Harunur. *Power electronics: circuits, devices, and applications*. Pearson Education India, 2003.

[4] Ahmed, Musse Mohamud, and Wai Lian Soo. "Supervisory control and data acquisition system (SCADA) based customized remote terminal unit (RTU) for distribution automation system." In *Power and Energy Conference, 2008. PECon 2008. IEEE 2nd International*, pp. 1655-1660. IEEE, 2008.

[5] Avhad, Mayur, Vinit Divekar, Harshad Golatkar, and Sanket Joshi. "Microcontroller based automation system using industry standard SCADA." In *India Conference (INDICON), 2013 Annual IEEE*, pp. 1-6. IEEE, 2013.