PROJECT REPORT

ON

PROCESS CONTROL SYSTEM USING SMART CARD

Submitted by

P-1152

A.RAJA
D.SARAVANAN
K.SHANMUGASUNDARAM
J.TAMILINIAN

Guided by

Mr. V. KANDASAMY, M.Tech

Lecturer, Dept. of EEE

In partial fulfillment of the requirements for the Award of the degree of Bachelor of Engineering in Electrical and Electronics Engineering Branch of Bharathiar University, Coimbatore

Estd-1984

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641 006

ISO 9001:2000 Certified

2003- 2004

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641 006

Estd-1984

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ISO 9001:2000 Certified

CERTIFICATE

This is to certify that the project report entitled

PROCESS CONTROL SYSTEM USING SMART CARD

Is the bonafide work done by

A.RAJA	**(2KEEE32)**
D.SARAVANAN	**(2KEEE64)**
K.SHANMUGASUNDARAM	**(2KEEE66)**
J.TAMILINIAN	**(2KEEE65)**

In partial fulfillment of the requirements for the Award of the degree of Bachelor of Engineering in Electrical and Electronics Engineering Branch of Bharathiar University, Coimbatore

Guide
Date:

Head of the Department

Certified that the candidate _____ with University Register No._____ was examined in Project work Viva-Voce Examination on _____

Internal Examiner

External Examiner

CONTENTS

ACKNOWLEDGEMENT

We express our sincere thanks to **Dr.K.K.Padmanabhan B.Sc (Engg)., M.Tech., Ph.D.,MISTE,FIE,** Principal, Kumaraguru College Of Technology for having provided us the opportunity to undertake this project.

Our heartful thanks to **Dr.T.M.Kameswaran B.E., M.Sc (Engg).,Ph.D., MISTE,Sr.MIEEE,FIE**, Head, Department Of Electrical and Electronics Engineering, for his great encouragement and inspiration throughout the curriculum.

We express our gratitude to our guide, **Mr.V.Kandasamy M.Tech.**, Lecturer Department of Electrical and Electronics, Kumaraguru College of Technology, who guided us throughout the project and encouraged us to successfully complete the project.

Finally we thank our parents and friends who have enriched our life
With their love and support. We are grateful to the Divine without whom nothing would have been possible.

SYNOPSIS

Temperature control is one of the most essential processes in many industries. Temperature has to be regulated in industrial sectors to avoid power loss. In most of the industries, temperature control is done using conventional devices like thermistor and thermostat. Accurate and remote control is impossible through these devices. This Poor temperature control causes increased losses and also penalizes the manufacturer.

This describes the significance of our project in industrial environment; the reliability of the existing system is very low that it cannot adopt itself to the ever-growing demand. So, we utilize most advanced technique to control the temperature. The other parameters like voltage and number of products of packing machine is monitored in this system.

Accurate control of temperature leads to efficient usage of power and improves the industry's economy also automates the entire plant, thus avoiding human errors.

This system has PIC micro controller and smart card with other control devices. PIC micro controller acts as the master of this system. The entire operation depends on the software coding in PIC. This ensures unauthorized person cannot change the set temperature. The datas are send to the smart card for a specified time interval. The data present in the smart card can be viewed through the personal computer using I^2C and USART protocols.

INTRODUCTION

1. INTRODUCTION

Nowadays the technology has been developing in all aspects, especially in industrial field. In these fields the industries have been modernizing to achieve quality and quantity end product.

The temperature control is one of the main process taking places in the industry. The increment of temperature beyond the reference value causes the problems like high power wastage and it also reduce the life of the components. So, it's very significant to maintain the temperature within the safe limit.

There are many techniques available for controlling the temperature like thermostat, thermo couple, thermistor and microprocessor etc, rather than this; PIC controller is the latest one. The advantage of PIC over the others is Précised temperature control.

The aim of our project is data acquisition and control using embedded technology. We have chosen to implement the control of a machine with the help of PIC micro controller. The data acquisition is achieved by **Smart Card.**

Smart card is a portable memory device, which is capable of storing data. Basically these types of cards are used for prepaid purposes. Nowadays, in industries smart cards are used for effective data storage and also these are very much suitable for remote monitoring.

Data acquisition involves Control of temperature, monitoring of voltage and output involved in packing machine.

PROBLEM STATEMENT

2. PROBLEM STATEMENT

The objective of our project is to control the temperature within the predefined value and also monitor the voltage fluctuation as well as the output of machine involved in the packing process.

This system consists of PIC micro controller and smart card with other control devices. PIC controls the entire system by acting as master. According to the software coding in the PIC, temperature will be maintained. Also the parameters like temperature, voltage and number of outcome of the machine are stored in the smart card for every predefined duration.

In traditional data acquisition system, micro controller kit is connected with PC to retrieve data. In this approach PC is the most essential device and data's are to be retrieved through the PC .So the PC has to run for the whole day. This causes power wastage and dedicated PC is required for each application, which in turn increases the cost of the system.

To avoid the above discrepancies, **Smart Card** is used to storing data. The process of writing and reading data has been done in the Smart Card using I^2C (Inter Integrated Communication) protocol and USART (Universal Synchronous Asynchronous Receiver Transmitter) protocol is used for PC to PIC communication. Smart Card data can be viewed using card reader, which in turn is connected to PC. Hence the usage of PC is reduced to a great extent.

MAIN BLOCK DIAGRAM

3. MAIN BLOCK DIAGRAM

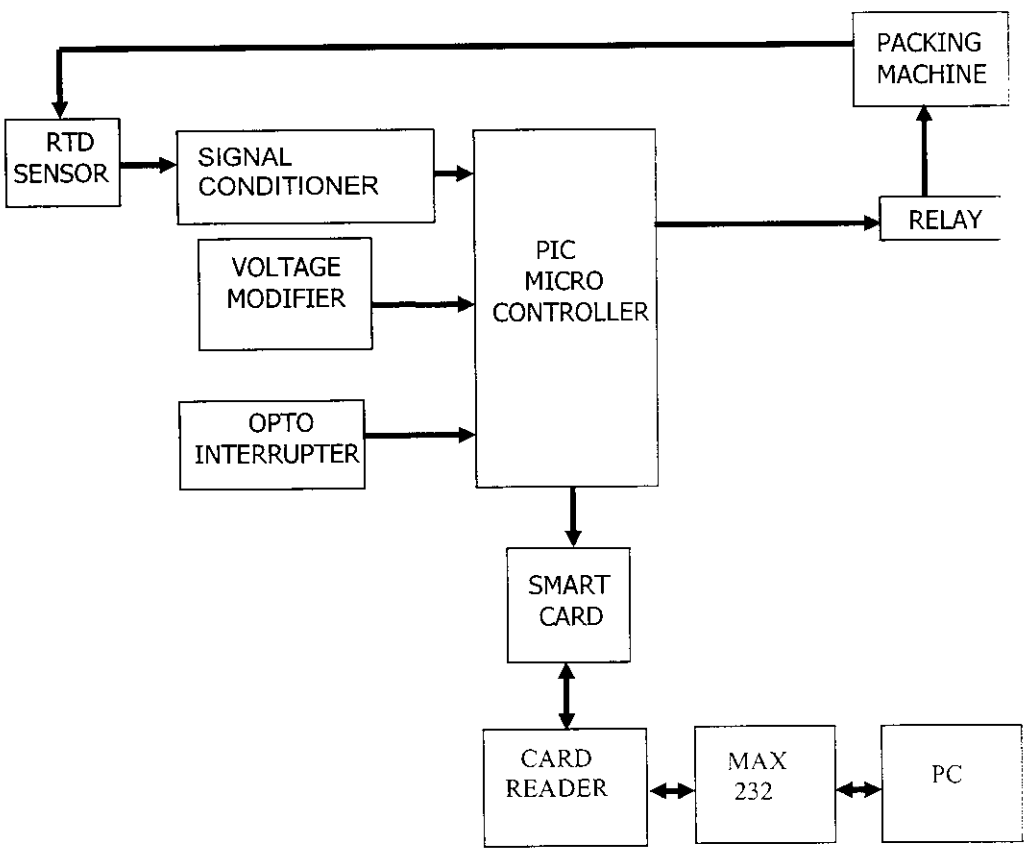


Fig.1. Block Diagram Of Process Control System

3.1 Process Control Systems

Process Control Systems is used for maintaining the temperature of heating element within the permissible limit and also monitoring the voltage and number of out coming products of the packing machine. This system has two major units.

- Feeder unit
- Reader unit

Feeder unit is fixed in packing machine. PIC is used as the heart of the system and relay, RTD sensor, Opto interrupter are the supporting components.

The RTD sensor monitors the temperature of the heating element continuously. This is a transducer, which converts temperature into an equivalent resistance form. This resistance is changed into analog signal by means of signal conditioning device and it's given to PIC.

PIC has an inbuilt analog to digital converter. It converts analog voltage to digital value and compares the digital value with the reference value.

If the temperature exceeds the safer limit it enables the pin in which sugar cube relay is connected. Relay disconnects the supply of heating element. it again connects the supply line of heating element when the temperature attains the safer limit. Fig .1 shows the block diagram of process control system .

Voltage Monitoring

The applied input voltage of the machine is monitored in order to determine the maximum fluctuation within the specified time duration. Potential divider circuit is used for this purpose. Potential divider circuit produces half of the input voltage as output. The applied voltage is reduced using step down transformer and this voltage is converted into DC by means of diode bridge rectifier. This DC voltage is applied to potential divider. The output of potential divider is in analog form and is fed to PIC. This is converted to digital value and stored in general purpose register. This value

is then sending to the Smart Card. The change in digital value with respect to analog value depends on potential divider ratio.

Counting Process

Counters are used to count the number of products. Opto interrupter circuit carries this task. Opto interrupter produces a pulse whenever the out coming product blocks the light emitted by LED. It provides pulse to the PIC and it counts the pulses given to it and stores it in register. This count value is sent to the smart card.

PIC writes temperature, voltage and number of products into Smart Card using I^2C protocol. (Inter integrated chip communication). PC retrieves the stored data in the Smart Card through USART (Universal Synchronous Asynchronous Receiver Transmitter) protocol. GUI tools convert the values into an actual form. Thus the three parameters can be monitored by the system. This data is used to find the performance of the machine and these data's are very useful while implementing the TPM (Total Production Management) in the industry. This helps to improve the productivity of the machine and the quality of the product.

HARDWARE DESCRIPTION

4. HARDWARE DESCRIPTION

4.1 PIC Micro controller

Micro controller is the tiny chip. It has inbuilt memory, timer, ports and other additional features. There are several companies, manufacturing the micro controllers like Intel, Motorola and Microchip. PIC is the product of microchip. The following are the special characteristics of PIC,

Features of PIC

- Long Word Instructions
- Single Word Instructions
- Instruction Pipeline
- Single Cycle Instruction
- Reduced Instruction Set
- Register File Architecture
- Orthogonal (Symmetric) Instructions
- Instruction Flow/Pipelining

PIC Registers

Arithmetic Logical Unit (ALU)

PIC Micro controllers contain an 8-bit ALU and an 8-bit working register. The ALU is a general-purpose arithmetic and logical unit. It performs arithmetic and Boolean functions between the data in the working register and any register file.

The ALU is 8-bits wide and is capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register). The other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register. The W register is an 8-bit working Register used for ALU operations. It is not an addressable register. Fig .2 shows the block diagram of ALU.

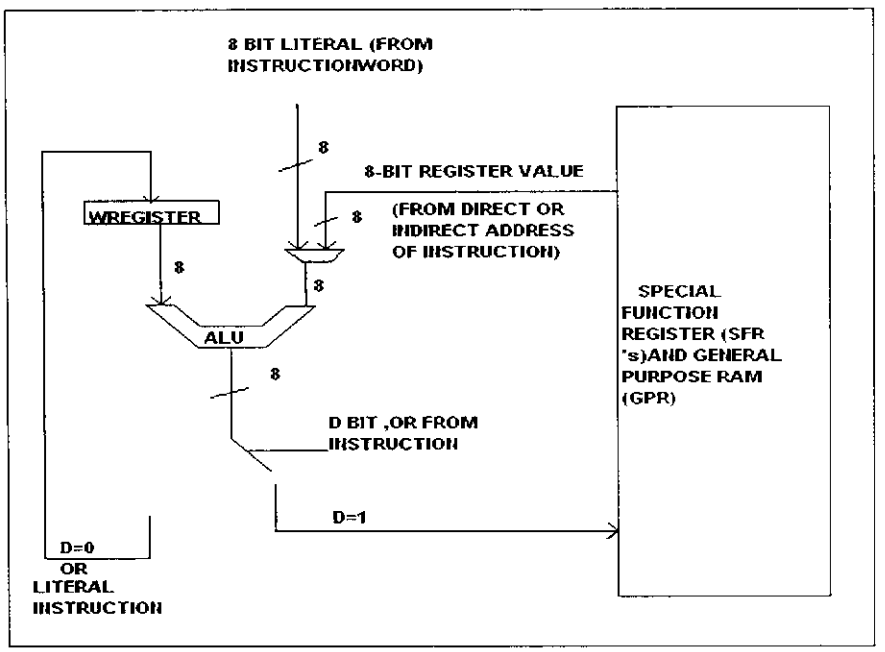


Fig. 2. Block diagram of ALU

Status Register

The STATUS register contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory. Since this register controls the selection of the Data Memory banks, it is required to be present in every bank. Also, this register is in the same relative position (offset) in each bank.

The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the TO and PD bits are not writable. Therefore, the result of an instruction with the STATUS registers, as destination may be different than intended. For example, CLRF STATUS will clear the upper-three bits and set the Z bit. This leaves the STATUS register as 000u u1uu (where u=unchanged).

It is recommended therefore, that only BCF, BSF, SWAPF and MOVWF instructions be used to alter the STATUS register because these instructions do not affect the Z, C or DC bits from the STATUS register. For other instructions not affecting any status bits.

IRP	RP1	RP0	T0	PD	Z	DC	C
Bit 7							Bit 0

bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)

1 = Bank 2, 3 (100h - 1FFh)

0 = Bank 0, 1 (00h - FFh)

For devices with only Bank0 and Bank1 the IRP bit is reserved, always maintain this bit clear.

bit 6:5 **RP1:RP0**: Register Bank Select bits (used for direct addressing)

11 = Bank 3 (180h - 1FFh)

10 = Bank 2 (100h - 17Fh)

01 = Bank 1 (80h - FFh)

00 = Bank 0 (00h - 7Fh)

Each bank is 128 bytes. For devices with only Bank0 and Bank1 the IRP bit is reserved, always maintain this bit clear.

bit 4 **TO**: Time-out bit

1 = After power-up, CLRWDT instruction, or SLEEP instruction

0 = A WDT time-out occurred

bit 3 **PD**: Power-down bit

1 = After power-up or by the CLRWDT instruction

0 = By execution of the SLEEP instruction

bit2 **Z**: Zero bit

1 = The result of an arithmetic or logic operation is zero

0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit carry/borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)
(for borrow the polarity is reversed)

1 = A carry-out from the 4th low order bit of the result occurred

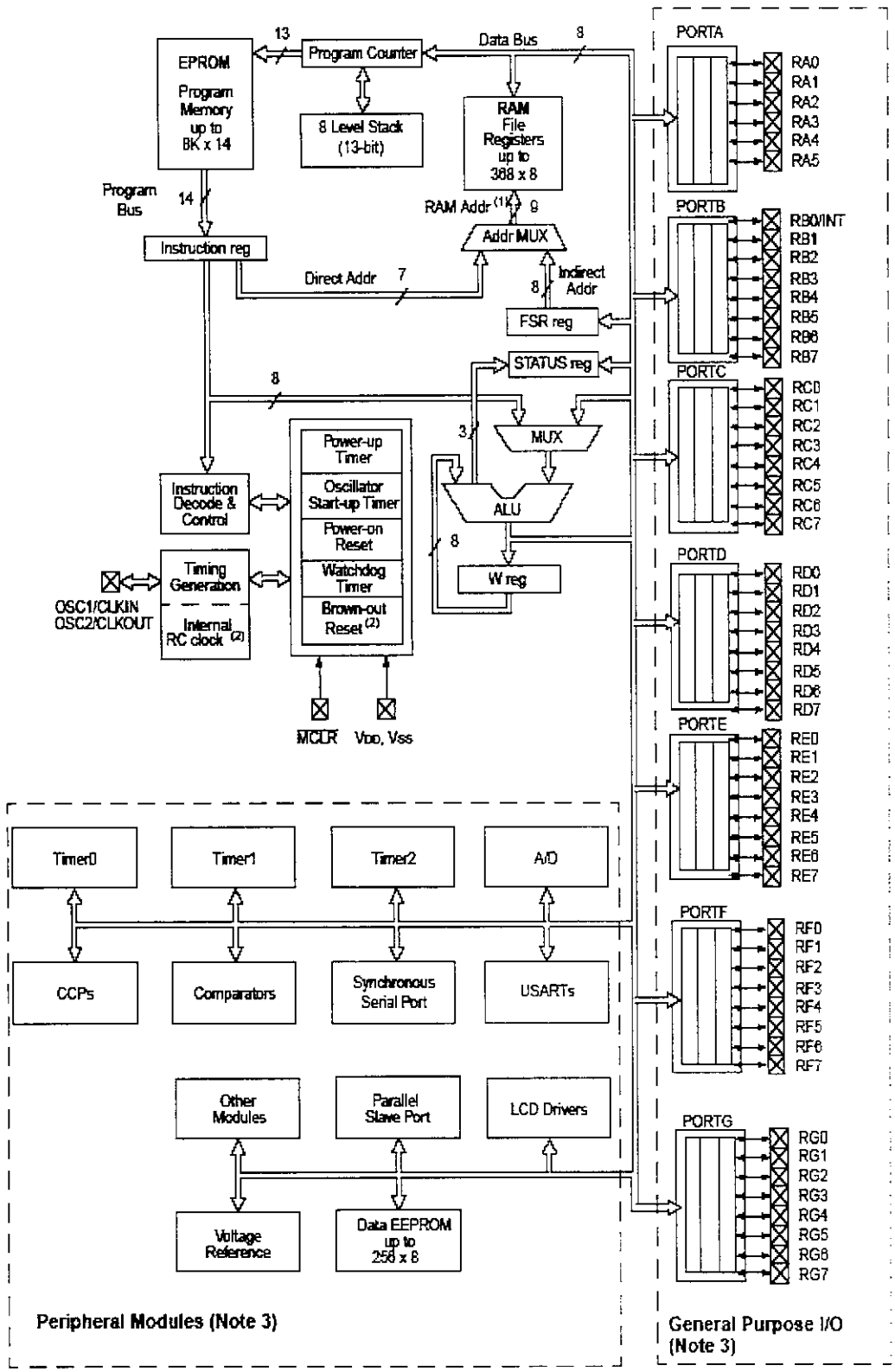
0 = No carry-out from the 4th low order bit of the result

bit 0 **C**: Carry/borrow bit (ADDWF, ADDLW,SUBLW,SUBWF instructions)

1 = A carry-out from the most significant bit of the result occurred

0 = No carry-out from the most significant bit of the result occurred

PIC ARCHITECTURE



Memory Organization

There are two-memory blocks

1.Program memory

2.Data memory.

Each block has its own bus, so that access to each block can occur during the same oscillator cycle. The data memory can further be broken down into

General Purpose RAM,

Special Function Register (SFRs).

Program Memory Organization

Mid-Range MCU devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. The width of the program memory bus (instruction word) is 14-bits. Since all instructions are a single word, a device with an 8K x 14 program memory has space for 8K of instructions. This makes it much easier to determine if a device has sufficient program memory for a desired application.

Program Counter (Pc)

The program counter (PC) specifies the address of the instruction to fetch for execution. The PC is 13-bits wide. The low byte is called the PCL register. This register is readable and writable. The high byte is called the PCH register. This register contains the PC<12:8> bits and is not directly readable or writable. All updates to the PCH register go through the PCLATH register.

Stack

The stack allows a combination of up to 8 program calls and interrupts to occur. The stack contains the return address from this branch in program execution. Mid-Range MCU devices have an 8-level deep x 13-bit wide hardware stack. The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSH onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POP in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not modified when the stack is PUSH or POP.

Data Memory Organization

Data memory is made up of the Special Function Registers (SFR) area, and the General Purpose Registers (GPR) area. The SFRs control the operation of the device, while GPRs are the general area for data storage and scratch pad operations. The data memory is banked for both the GPR and SFR areas. The GPR area is banked to allow greater than 96 bytes of general purpose RAM to be addressed. SFRs are for the registers that control the peripheral and core functions.

Banking requires the use of control bits for bank selection. These control bits are located in the STATUS Register (STATUS<7:5>). To move values from one register to another register, the value must pass through the W register. This means that for all register-to-register moves, two instruction cycles are required. The entire data memory can be accessed either directly or indirectly. Direct addressing may require the use of the RP1:RP0 bits. Indirect addressing requires the use of the File Select Register (FSR). Indirect addressing uses the Indirect Register Pointer (IRP) bit of the STATUS register for accesses into the Bank0 / Bank1 or the Bank2 / Bank3 areas of data memory.

Special Function Registers (SFR)

The CPU and Peripheral Modules use the SFRs for controlling the desired operation of the device. These registers are implemented as static RAM. The SFRs can be classified into two sets, those associated with the "core" function and those related to the peripheral functions. Switching between these banks requires the RP0 and RP1 bits in the STATUS register to be configured for the desired bank. A Power-on Reset and other resets initialize some SFRs, while other SFRs are unaffected.

Port Control registers

Port A and Tris A Register

The RA4 pin is a Schmitt Trigger input and an open drain output. All other RA port pins have TTL input levels and full CMOS output drivers. All pins have data direction bits (TRIS registers), which can configure these pins as output or input.

Setting a TRISA register bit puts the corresponding output driver in a hi-impedance mode. Clearing a bit in the TRISA register puts the contents of the output latch on the selected pin(s).

Port b and Tris b Register

PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. Setting a bit in the TRISB register puts the corresponding output driver in a high-impedance input mode. Clearing a bit in the TRISB register puts the contents of the output latch on the selected pin(s).

Port c and Tris c Register

PORTC is an 8-bit bi-directional port. Each pin is individually configurable as an input or output through the TRISC register. PORTC pins have Schmitt Trigger input buffers. When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override.

4.2 SMART CARD

Smart Cards are plastic cards with integrated circuit chips. These integrated circuits chips are highly specialized in that; they contain memory with in a single integrated circuit chip. This configuration enhances security of the chip as it shields the relatively low level electrical signal between elements from view by the outside world. The physical environments in which Smart Cards are used require that the chips be physically small. This physical limitation greatly restricts the amount of memory that can be incorporated on the chips.

Smart Cards are quickly becoming the preferred platform for conducting transaction and verifying identities around the world. These small devices look like a credit card but are capable of acting like a computer, thus the term Smart Card.

The portable convenience and security of the Smart Card has lead to increasing demand. It can be used to verify a users identity to log on to a computer network; allow a physician to scan a patient medical record and it will soon be the modern replacement of many forms. ID, travel, credit and access cards.

Advantages

1. It Can store a hundred times more information than a magnetic strip card
2. More durable and much more secure than a magnetic strip card
3. Can perform multiple functions in a wide range of industries
4. Compactable with portable electronic devices such as phones, personal digital assistance (PDA), PC
5. Constantly evolving to accommodate technological advancements.

There are a variety of memory Smart Card in sizes that range from 1 kilobit to 256 kilobits. The memory size of the smart used in our system is128 kilobits.

Physical Characteristics

Smart Cards present a variety of faces, depending primarily on the type of integrated circuit chip embedded in the plastic card and the physical form of the connection mechanism between the card and reader. They can be very inexpensive token for financial transaction such as credit card and telephone calling tokens, or loyalty tokens from a variety of businesses. They can be access tokens for getting through locked doors, riding on a train or driving an automobile on toll road. They can function as identity tokens for logging into a computer system or accessing a World Wide Web server with an authenticated identity. Three such variants are of particular interest:

1.cards with surface contact leading to a memory-only integrated circuit chip

2.cards with an electromagnetic connection to a microprocessor- integrated circuit chip

3.cards with surface contact leading to a microprocessor- integrated circuit chip

The very earliest Smart Cards are memory cards containing an integrated circuit chip comprised of only non-volatile memory and the necessary circuitry to read and write that memory. Today, such cards still constitute the largest number of Smart Cards in use. These cards are relatively inexpensive and provide modest security for a variety of applications.

A memory card, as its name implies, is a card that contains an embedded integrated circuit chip providing nonvolatile memory for storing information in a permanent semi permanent fashion. The circuitry of the Smart Card exposes, through a standard electrical connector, the control lines for addressing selected memory locations as well as for reading and writing those memory locations through electrical connectors on the face of the card.

There is no on board processor to support high-level communications protocol between the reader and the card. Rather, memory card use a synchronous communication mechanism between the reader and the card.

Essentially, the communication channel is always under the direct control of the reader side. The card circuitry response in a direct (synchronous) way to the very low level commands issued by the reader for addressing memory locations and for reading from or writing to selected locations.

In some recent memory cards, security enhancements have been incorporated through the provision of memory addressing circuitry within the chip that requires a shared secret between the terminal (which is writing to the card chip) and chip itself. These are often called logic cards. Auxiliary elements (such as power line frequency, voltage filters and memory mapping registers) included in the chip for security or functionality reasons.

Size

The small size needed for chip features requires leading edge technology. However, in order for chips to be inexpensive and reliable, we often need to turn to older, more mature technologies.

Width

The width of the internal bus structure indicates the number of memory address lines running between components within a chip; that is, width is generally indicative of the number of bits in individually addressable sections of memory. Minimizing chip size generally tends to call for a selection of fewer address lines; therefore, most Smart Card chips are currently based on 8-bit microprocessor. This microprocessor also tends to be the older and more major technologies.

Memory

The type of memory used in Smart Card chips is electrically erasable and programmable read-only memory (EEPROM). The contents of this type of memory in a Smart Card chip can actually be modified during normal use of the card. Hence, programs or data can be stored in EEPROM during normal operation of the card and then read back by applications that are using the card. The electrical characteristics of EEPROM memory are such that it can only be erased and then reprogrammed a finite (but reasonably large) number of times, generally around 100,000 times. While somewhat limited, techniques have evolved which make this type of memory quite useful for typical Smart Card uses. The net result is that Smart Card chips tend to make use of varying amounts of each memory type, depending on the specific application for which the Smart Card is to be used. The most powerful chips used in Smart Cards today have RAM sizes in the 256-byte to 32-KB range, ROM sizes in the 16-KB to 128-KB range, and EEPROM sizes in the 1-Kb to 256-KB range.

Interface Devices (Readers)

A Smart Card contains no independent power sources or clock signal to drive its embedded processor. Consequently, it must be plugged in to a device that can provide it with both power and clock. This device formally referred to us an interface device and is often called a reader or a terminal. The term reader is generally used when the reader is providing a connection between a Smart Card and another computer system such as a PC.

4.3 OPTO INTERRUPTER

This is a subminiature photo reflector whose GaAs infrared emitting diode and silicon transistor are assembled in the same package allowing for easy installation and handling. This has an excellent S/N ratio (more than 40dB) and contains a built–in filter for cutting visible light

It's an optical sensor. Opto interrupter has two parts,

Transmitter,

Receiver.

LED acts as a transmitter and phototransistor works as a receiver. If the light ray emitted by a LED is interrupted, it will produce one pulse, which is given to PIC. This Opto interrupter circuit is utilized for counting the number of out coming products. Opto interrupter and its internal diagram are shown in Fig. 4 and 4.1.

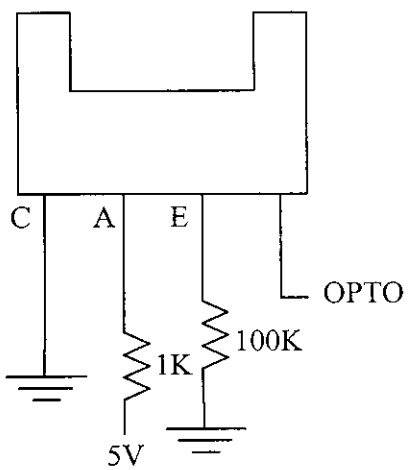


Fig: 4 Opto interrupter

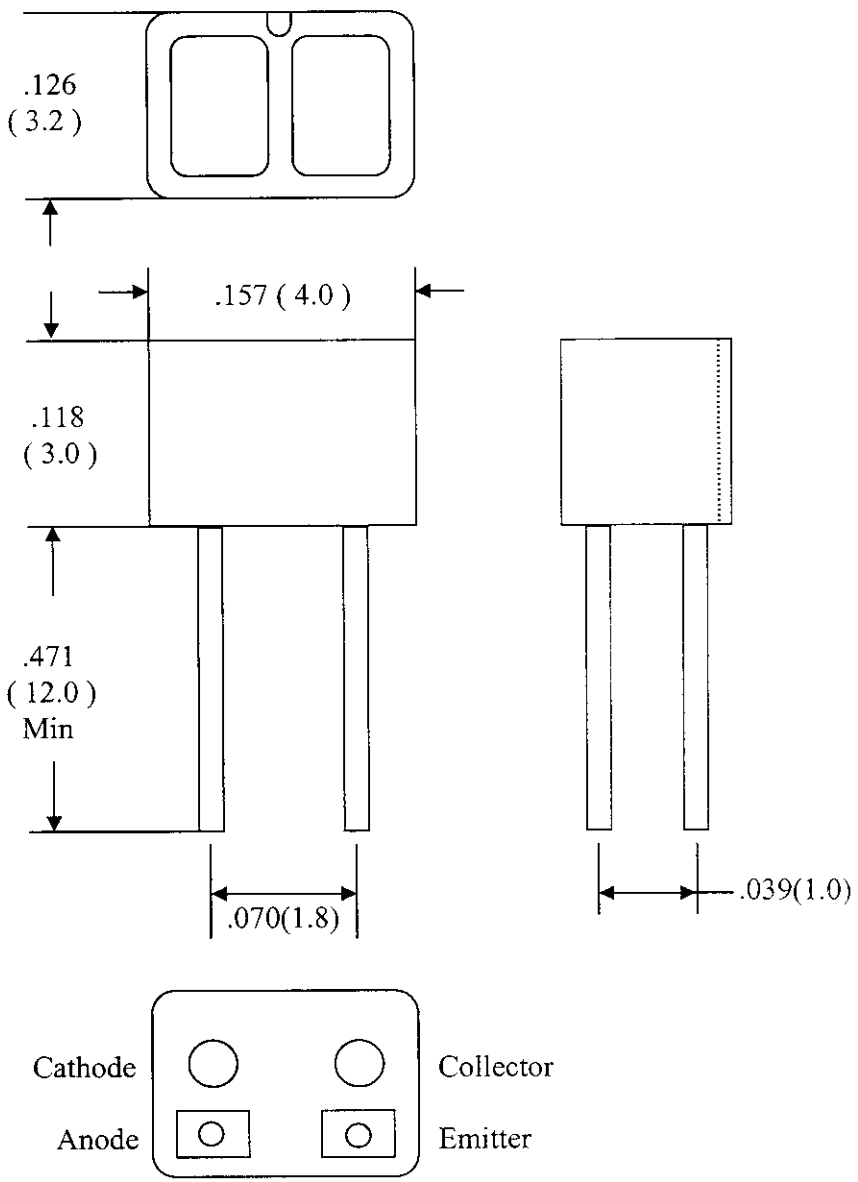


fig :4.1 Internal diagram of opto interrupter

4.4 RTD SENSOR

RTD stands for Resistance Temperature Detector. It's a transducer (it is a device which converts one form of energy into another form), which converts temperature into resistance. RTD consists of three pins.

Temperature is given as input to RTD and corresponding electrical value will be produced (Resistance) at the output. RTDs are designed to measure temperature in a variety of process and laboratory applications. These RTDs are specifically designed for use in two different process temperature ranges and they will provide accurate and repeatable temperature measurement through a broad range of -328 to 1112°F (-200 to 600°C) RTD sensors are constructed using silver plated copper internal leads, Teflon, and other suitable wire insulations with potting compounds to resist moisture penetration. Fig .5 indicates the diagram of RTD.

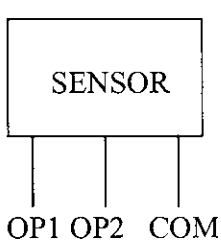


Fig: 5 RTD sensor

The resistance of a conductor changes when its temperature is changed. This property is utilized for measurement of temperature. The variation of resistance (R) with temperature (T) can be represented by the below relationship.

R=R0 (1+a1T+a2T2+..... + anTn)

R0 =Resistance at temperature T= 0 and a1, a2, an are constants. RTD sensor uses the change in electrical resistance of conductor to determine the temperature.

Requirements Of A Conductor Material Used In RTD

- The change in material per unit change in temperature should as large as possible.
- The material should have a high value of resistivity so that minimum volume of material is used for the construction of RTD.
- The resistance materials should have a continuous and stable relationship with temperature.

Platinum is mostly used in RTD sensor because of its good thermal characteristics.

Gold and silver are rarely used to make RTD on account of low resistivities. Tungsten has relatively high resistivity, but it's reserved for high temperature applications. Copper is used occasionally as an RTD element.

Its low linearity and low cost make it in an economical alternative.

- Upper limit temperature --- 120 c

The most common RTDs are made of platinum, nickel, or nickel alloys. The economical nickel wires are used over a limited range. They are quite non –linear and tend to drift. For the RTD sensors platinum is the best choice. The common values of a resistance for a platinum RTD range from 10 ohm to thousands of ohms.

4.5 SUGAR CUBE RELAY

This is a simple switching device that is positive -on and positive- off. When the coil is pulsed it takes a lever that mechanically changes the state of the contact. The current stays on with only a pulse. To turn it, a pulse is sent through its coil whereby the relay "ratchets" off. This eliminates the need to reverse the polarity as in a latching relay. There is also no constant drain in power by using this type of relay. Fig .6 represents the diagram of sugar cube relay. Generally relays are meant for disconnecting main circuit from the faulty condition. In our system sugar relay is used for protection it disconnects the supply line to the heating element, whenever the temperature exceeds the permissible limit and it connects the line again, if the temperature reaches within the safer limit.

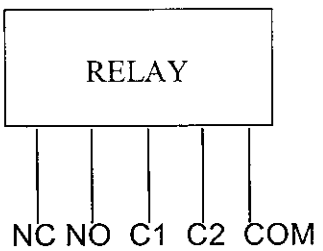


Fig : 6 Sugar Cube Relay

4.6 SIGNAL CONDITIONING CIRCUIT

The output of the RTD sensor is resistance. This resistance value is converted into voltage (Analog signal). This task is achieved by signal conditioning circuit .Fig 7 shows the signal conditioning circuit.

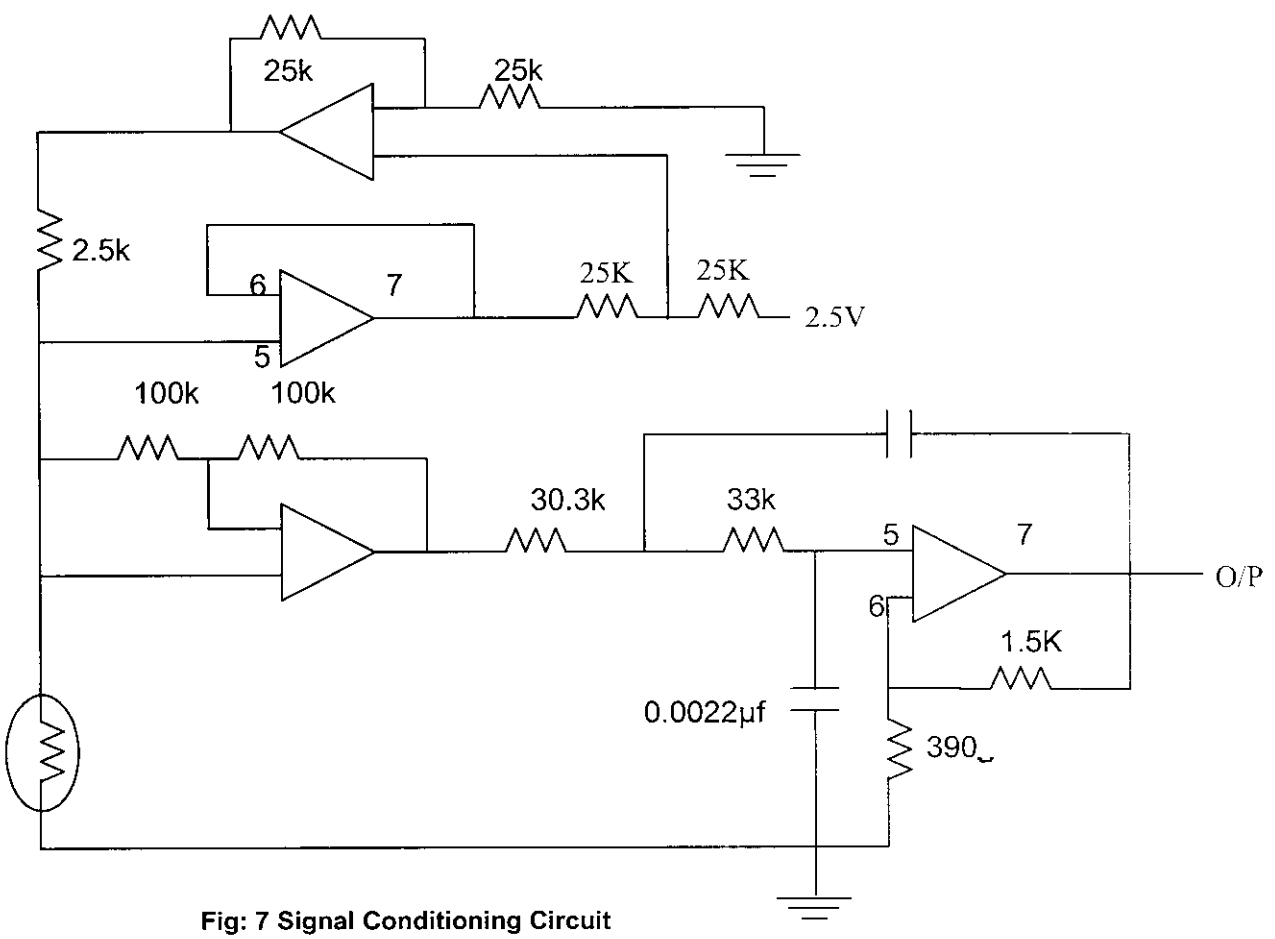


Fig: 7 Signal Conditioning Circuit

4.7 POTENTIAL DIVIDER CIRCUIT

Potential divider is a circuit, which divides the given input voltage. Fig. 8 shows the circuit configuration of potential divider.

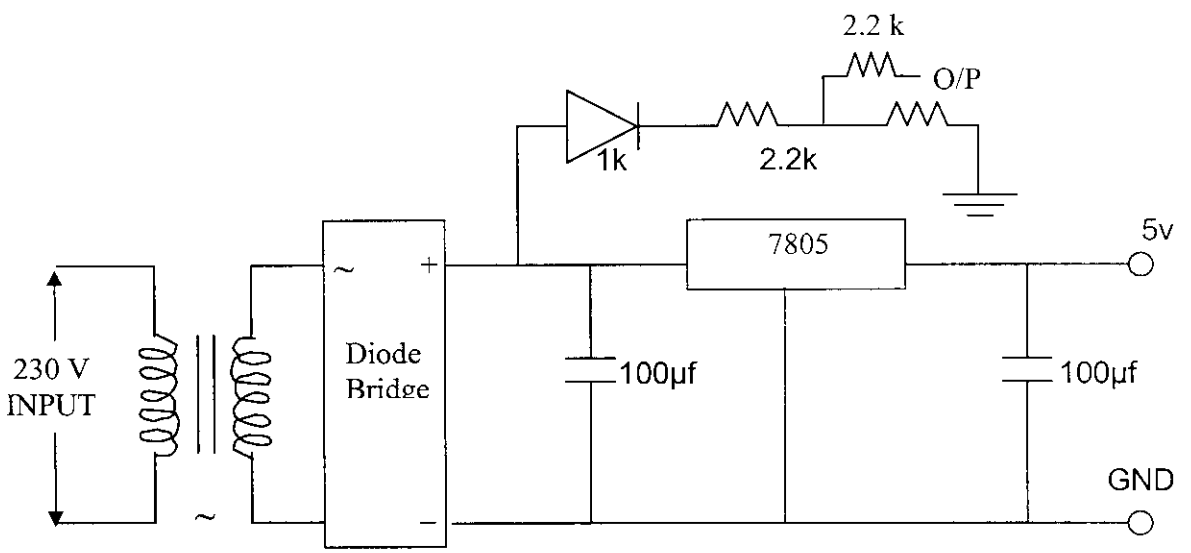


Fig :8 Potential Divider Circuit

The input (230V) is applied to the potential divider circuit .it will produce the half of the input as the output. The output of the potential divider is given to the PIC.

4.8 RS 232

It's a protocol totally relevant to the PC.This protocol enables the PC to read the data from smart card through PIC.

RS 232 Connector Pin Assignment

RS 232 connector was originally developed to use 25 pins. In this, pin-out provisions were made for a secondary RS232 communication channel. Now a day, the smaller 9-pin version is most commonly used, its shown in the Fig .9.

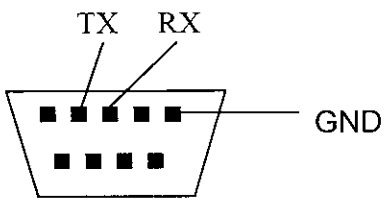
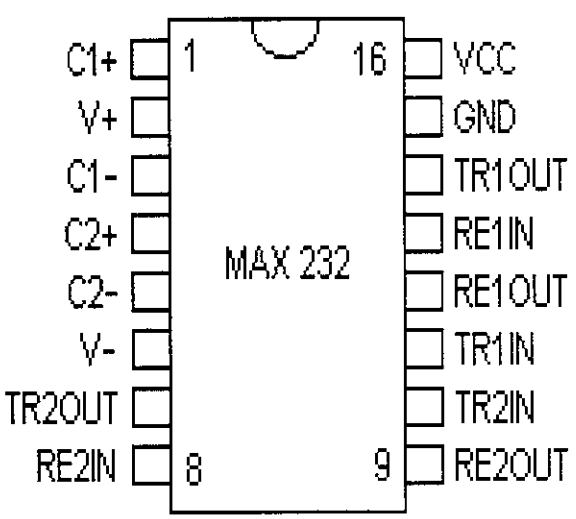


Fig: 9 RS 232

4.9 MAX 232:

This IC is popularly known as a voltage shifter. It's a 16- pin IC used for the PC to PIC communication. This IC is connected between card reader and PC.

PIN DIAGRAM:



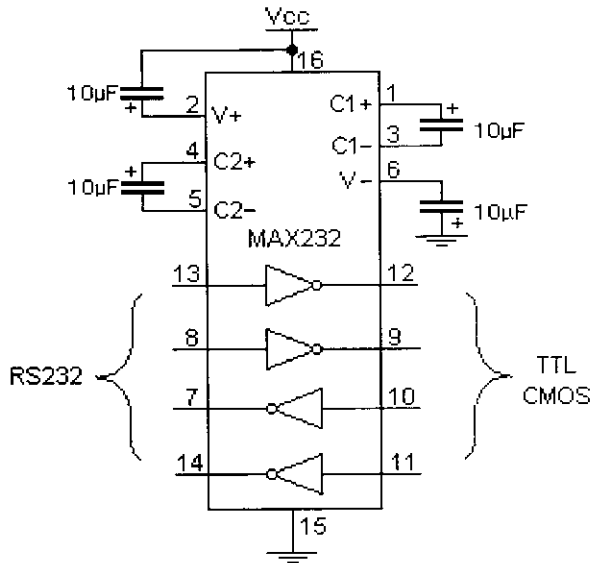


Fig: 10 MAX 232

Voltage level of PC

HIGH -- +12V

LOW -- -12V

Voltage level of PIC

HIGH- +5V

LOW- 0V

The voltage level of PIC is 0 TO 5 Volts. In order to make communication between PC and PIC voltage conversion is essential. MAX 232 achieves this voltage conversion. This internally consists of diodes, logic gates, and capacitors. MAX 232 modifies the voltage level with respect to PC and vice versa. Fig. 10 shows the pin diagram of MAX 232 IC.

RESULTS AND IMPLEMENTATION

5.RESULTS AND IMPLEMENTATION

The various steps involved in implementation are as follows:

5.1 Analog To Digital Conversion

The analog-to-digital (A/D) converter module has up to eight analog inputs. The A/D allows conversion of an analog input signal to a corresponding 8-bit digital number. The output of the sample and hold is the input into the converter, which generates the result via successive approximation. The analog reference voltage is software selectable to either the device's positive supply voltage (VDD) or the voltage level on the VREF pin. The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode. The A/D module has three registers. These registers are:

- A/D Result Register (ADRESH)
- A/D Control Register0 (ADCON0)
- A/D Control Register1 (ADCON1)

The ADCON0 register controls the operation of the A/D module. The ADCON1 register configures the functions of the port pins. The I/O pins can be configured as analog inputs (one I/O can also be a voltage reference) or as digital I/O.Fig.11 shows the ADC diagram.

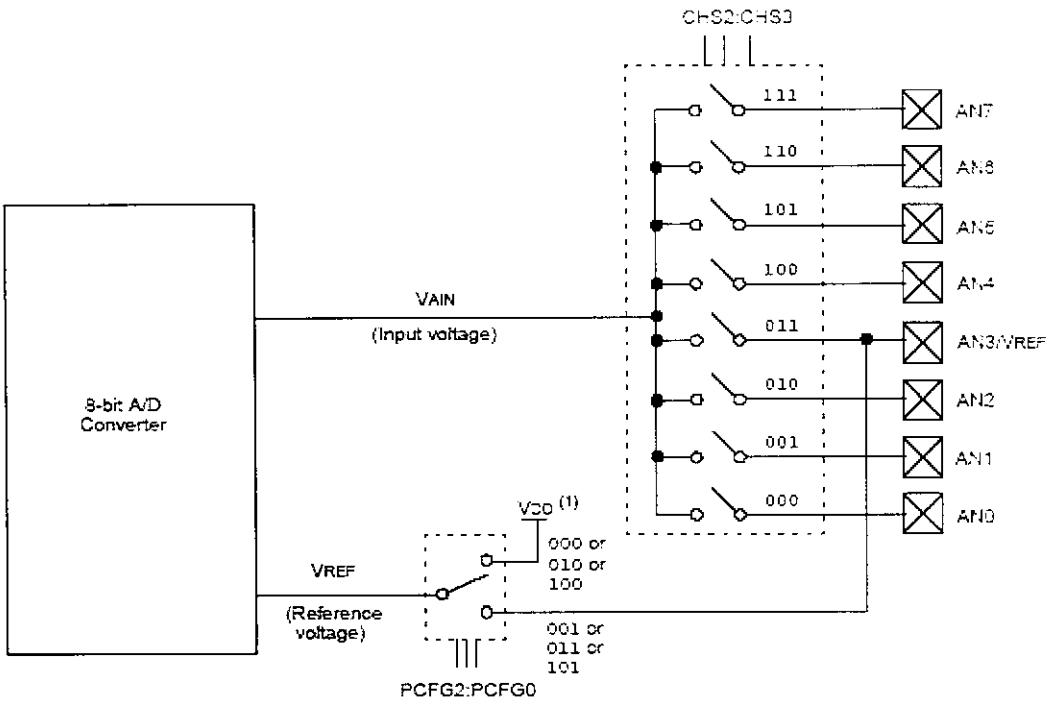


Fig: 11 ADC Diagram

Control Registers

ADCON0 Register

ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	Resv	ADON

Bit 7 Bit 0

bit 7:6 **ADCS1:ADCS0**: A/D Conversion Clock Select bits

00= FOSC/2

01= FOSC/8

10= FOSC/32

11= FRC (clock derived from the internal A/D RC oscillator)

bit 5:3 **CHS2:CHS0**: Analog Channel Select bits

000= channel 0, (AN0)

001= channel 1, (AN1)

010= channel 2, (AN2)

011= channel 3, (AN3)

100= channel 4, (AN4)

101= channel 5, (AN5)

110= channel 6, (AN6)

111= channel 7, (AN7)

Note:

For devices that do not implement the full 8 A/D channels, the unimplemented selections are reserved. Do not select any unimplemented

Channels.

bit 2 **GO/DONE:**A/D Conversion Status bit

When ADON = 1

1 = A/D conversion in progress

(Setting this bit starts the A/D conversion. This bit is automatically cleared by hardware when the A/D conversion is complete)

0 = A/D conversion not in progress

bit 1 **Reserved:** Always maintain this bit cleared.

bit 0 **ADON**: A/D On bit

1 = A/D converter module is operating

0 = A/D converter module is shutoff and consumes no operating current

ADCON1 Register

--	--	--	--	--	PCFG2	PCFG1	PCFG0

Bit 7 Bit 0

bit 7:3 **Unimplemented**: Read as '0'

bit 2:0 **PCFG2:PCFG0**: A/D Port Configuration Control bits

PCFG2:PCFG0	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
000	A	A	A	A	A	A	A	A
001	A	A	A	A	VREF	A	A	A
010	D	D	D	A	A	A	A	A
011	D	D	A	A	VREF	A	A	A
100	D	D	D	D	A	D	A	A
101	D	D	D	D	VREF	D	A	A
11x	D	D	D	D	D	D	D	D

A = Analog input D = Digital I/O

Procedural Approach

- RA0 is set to one to make it as input.
- Sending the data 0x80 to this register configures ADCON1; in this register the bit ADFM is set.
- Sending the data 0x81 to this register configures ADCON 0, in this register ADON bit is set to enable conversion and also ADCS1 is set to select the frequency of conversion.
- Delay subroutine is called (10micro sec).
- ADGO bit is set to one.
- ADGO bit is checked to know either the A TO D conversion is completed or not.
    - ADGO is zero --- Conversion has done
    - ADGO is one --- Conversion hasn't done.

Digital Output

(ADRESH * 256)+ ADRESL

5.2 COUNTING PROCESS

Control Register

The OPTION_REG register is a readable and writable register, which contains various control bits to configure the TMR0/WDT prescaler, the External INT Interrupt, TMR0, and the weak pull-ups on PORTB.

OPTION_REG Register

To achieve a 1:1 prescaler assignment for the TMR0 register, assign the prescaler to the Watchdog Timer.

bit 7 **RBPU(1):**Weak Pull-up Enable bit

1 = Weak pull-ups are disabled

0 = Weak pull-ups are enabled by individual port latch values

bit 6 **INTEDG**: Interrupt Edge Select bit

1 = Interrupt on rising edge of INT pin

0 = Interrupt on falling edge of INT pin

bit 5 **T0CS**: TMR0 Clock Source Select bit

1 = Transition on T0CKI pin

0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE**: TMR0 Source Edge Select bit

1 = Increment on high-to-low transition on T0CKI pin

0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA**: Prescaler Assignment bit

1 = Prescaler is assigned to the WDT

0 = Prescaler is assigned to the Timer0 module

bit 2:0 **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMRO Rate	WDT Rate
0 0 0	1 : 2	1 : 1
0 0 1	1 : 4	1 : 2
0 1 0	1 : 8	1 : 2
0 1 1	1 : 16	1 : 8
1 0 0	1 : 32	1 : 16
1 0 1	1 : 64	1 : 32
1 1 0	1 : 128	1 : 64
1 1 1	1 : 258	1 : 128

PIR Register

Depending on the number of peripheral interrupt sources, there may be multiple Peripheral Interrupt Flag registers (PIR1, PIR2). These registers contain the individual flag bits for the peripheral interrupts. These registers will be generically referred to as PIR. Although, the PIR bits have a general bit location within each register, future devices may not be able to be consistent with that. It is recommended that you use the supplied Microchip Include files for the symbolic use of these bits. This will allow the Assembler/Compiler to automatically take care of the placement of these bits within the specified register. Interrupt flag bits get set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit,

TMR1IF: TMR1 Overflow Interrupt Flag bit

1 = TMR1 register overflowed (must be cleared in software)

0 = TMR1 register did not overflow bit

TMR2IF: TMR2 to PR2 Match Interrupt Flag bit

1 = TMR2 to PR2 match occurred (must be cleared in software)

0 = No TMR2 to PR2 match occurred bit.

CCP1IF: CCP1 Interrupt Flag bit

Capture Mode

1 = A TMR1 register capture occurred (must be cleared in software)

0 = No TMR1 register capture occurred

Compare Mode

1 = A TMR1 register compare match occurred (must be cleared in software)

0 = No TMR1 register compare match occurred

PWM Mode

Unused in this mode bit

CCP2IF: **CCP2** Interrupt Flag bit

Capture Mode

1 = A TMR1 register capture occurred (must be cleared in software)

0 = No TMR1 register capture occurred

Compare Mode

1 = A TMR1 register compare match occurred (must be cleared in software)

0 = No TMR1 register compare match occurred

PWM Mode

Unused in this mode bit

SSPIF: Synchronous Serial Port Interrupt Flag bit

1 = the transmission/reception is complete

0 = Waiting to transmit/receive bit

RCIF USART Receive Interrupt Flag bit

1 = The USART receive buffer, RCREG, is full (cleared when RCREG is read)

0 = The USART receive buffer is empty bit

TXIF: USART Transmit Interrupt Flag bit

1 = The USART transmit buffer, TXREG, is empty (cleared when TXREG is written)

0 = The USART transmit buffer is full bit

ADIF: A/D Converter Interrupt Flag bit

1 = An A/D conversion completed (must be cleared in software)

0 = The A/D conversion is not complete

INTCON REGISTER

The INTCON Register is a readable and writable register, which contains various enable and flag bits.

GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF

Bit 7 Bit 0

bit 7 **GIE:** Global Interrupt Enable bit

1 = Enables all un-masked interrupts

0 = Disables all interrupts

bit 6 **PEIE:** Peripheral Interrupt Enable bit

1 = Enables all un-masked peripheral interrupts

0 = Disables all peripheral interrupts

bit 5 **T0IE:** TMR0 Overflow Interrupt Enable bit

1 = Enables the TMR0 overflow interrupt

0 = Disables the TMR0 overflow interrupt

bit 4 **INTE:**INT External Interrupt Enable bit

1 = Enables the INT external interrupt

0 = Disables the INT external interrupt

bit 3 **RBIE (1):** RB Port Change Interrupt Enable bit

1 = Enables the RB port change interrupt

0 = Disables the RB port change interrupt

bit 2 **T0IF:** TMR0 Overflow Interrupt Flag bit

1 = TMR0 register has overflowed (must be cleared in software)

0 = TMR0 register did not overflow

bit 1 **INTF:**INT External Interrupt Flag bit

1 = The INT external interrupt occurred (must be cleared in software)

0 = The INT external interrupt did not occur

bit 0 **RBIF (1):** RB Port Change Interrupt Flag bit

1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)

0 = none of the RB7:RB4 pins have changed state

5.3 I^2C COMMUNICATION

Inter Integrated chip is an expansion for I^2C. This protocol is used for IC-TO-IC communication. In our system, I^2C protocol is used for interfacing the smart card and PIC.

I^2C REGISTERS

Totally, Five registers are used for I^2C communication. There are,

SSPSTAT: Status Register

SSPADD: Clock Register

SSPCON: Control Register 1

SSPCON 2: Control Register 2

SSPBUF: Data Register

Synchronous Serial Port

The Synchronous Serial Port (SSP) module is a serial interface useful for communicating with other peripherals or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The SSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI™)
- Inter-Integrated Circuit (I^2C™)
  - Slave mode
  - I/O slope control, and Start and Stop bit detection to ease software implementation of Master and Multi-master modes

Control Registers

SSPSTAT: Synchronous Serial Port Status Register

SMP	CKE	D/A	P	S	R/W	UA	BF

bit 7 bit 0

bit 7 **SMP:** SPI data input sample phase

SPI Master Mode

1 = Input data sampled at end of data output time

0 = Input data sampled at middle of data output time

SPI Slave Mode

SMP must be cleared when SPI is used in slave mode

bit 6 **CKE**: SPI Clock Edge Select

CKP = 0 (SSPCON<4>)

1 = Data transmitted on rising edge of SCK

0 = Data transmitted on falling edge of SCK

CKP = 1 (SSPCON<4>)

1 = Data transmitted on falling edge of SCK

0 = Data transmitted on rising edge of SCK

bit 5 **D/A**: Data/Address bit (I^2C mode only)

1 = Indicates that the last byte received or transmitted was data

0 = Indicates that the last byte received or transmitted was address

bit 4 **P**: Stop bit

(I2C mode only. This bit is cleared when the SSP module is disabled)

1 = Indicates that a stop bit has been detected last (this bit is '0' on RESET)

0 = Stop bit was not detected last

bit 3 **S**: Start bit

(I2C mode only. This bit is cleared when the SSP module is disabled)

1 = Indicates that a start bit has been detected last (this bit is '0' on RESET)

0 = Start bit was not detected last

bit 2 **R/W**: Read/Write bit information (I^2C mode only)

This bit holds the R/W bit information following the last address match. This bit is only valid from the address match to the next start bit, stop bit, or not ACK bit.

1 = Read

0 = Write

bit 1 **UA**: Update Address (10-bit I^2C mode only)

1 = Indicates that the user needs to update the address in the SSPADD register

0 = Address does not need to be updated

bit 0 **BF**: Buffer Full Status bit Receive (SPI and I^2C modes)

1 = Receive complete, SSPBUF is full

0 = Receive not complete, SSPBUF is empty

Transmit (I^2C mode only)

1 = Transmit in progress, SSPBUF is full

0 = Transmit complete, SSPBUF is empty

SSPCON: Synchronous Serial Port Control Register

WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSP
bit 7							bit 0

bit 7 **WCOL**: Write Collision Detect bit

1 = The SSPBUF register is written while it is still transmitting the previous word(must be cleared in software)

0 = No collision

bit 6 **SSPOV**: Receive Overflow Indicator bit

In SPI mode:

1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost and the SSPBUF is no longer updated. Overflow can only occur in slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In master mode the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

0 = No overflow

In I^2C mode:

1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in transmit mode. SSPOV must be cleared in software in either mode.

0 = No overflow

bit 5 **SSPEN**: Synchronous Serial Port Enable bit

In both modes, when enabled, these pins must be properly configured as input or output.

In SPI mode:

1 = Enables serial port and configures SCK, SDO, SDI, and SS as the source of the serial port pins

0 = Disables serial port and configures these pins as I/O port pins

In I^2C mode:

1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins

0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP**: Clock Polarity Select bit

In SPI mode:

1 = Idle state for clock is a high level

0 = Idle state for clock is a low level

In I^2C mode:

SCK release control

1 = Enable clock

0 = Holds clock low (clock stretch) (Used to ensure data setup time)

bit 3:0 **SSPM3:SSPM0**: Synchronous Serial Port Mode Select bits

0000= SPI master mode, clock = FOSC/4

0001= SPI master mode, clock = FOSC/16

0010= SPI master mode, clock = FOSC/64

0011= SPI master mode, clock = TMR2 output/2

0100= SPI slave mode, clock = SCK pin. SS pin control enabled.

0101= SPI slave mode, clock = SCK pin. SS pin control disabled. SS can be used as I/O pin

0110= I^2C slave mode, 7-bit address

0111= I^2C slave mode, 10-bit address

1000= Reserved

1001= Reserved

1010= Reserved

1011= I^2C firmware controlled master mode (slave idle)

1100= Reserved

1101= Reserved

1110= I^2C slave mode, 7-bit address with start and stop bit interrupts enabled

1111= I^2C slave mode, 10-bit address with start and stop bit interrupts enabled

SSPCON 2:

GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN

bit 7 bit 0

I^2C Transmission

- RC3 and RC4 are assigned as input for transmission.
- Sending the data 0x80 to this register configures the status register SSPSTAT.
- Clock Speed is moved to SSPADD just by sending decimal 100.
- SSPCON2 is then cleared.
- Sending the data 0x28 to this register configures SSPCON.
- Start bit is send by setting SEN pin of the SSPCON2

- Poll the SSPIF to check either the start bit is sent or not.
- SSPIF is one --- start bit is sent
- SSPIF is zero --- start bit is not sent
- Address is moved to SSPBUF
- SSPIF and ACKSTAT (it's a bit in SSPCON2 used as acknowledge indicator) are checked to know the address is sent or not.
    - ACKSTAT is one - not sent
    - ACKSTAT is zero- sent
- Data is moved to SSPBUF
- SSPIF and ACKSTAT are checked again to find the data is sent or not
- Stop bit is send just by setting PEN bit of the SSPCON2
- Finally, Poll the SSPIF to check either the stop bit is sent or not.
- The above steps are continued for all data's.

I^2C Reception

- RC3 and RC4 are assigned as input for transmission.
- Sending the data 0x80 to this register configures the status register SSPSTAT.
- Clock Speed is moved to SSPADD just by sending decimal 100.
- SSPCON2 is then cleared.
- Sending the data 0x28 to this register configures SSPCON.
- Start bit is received by setting SEN pin of the SSPCON2
- Poll the SSPIF to check either the start bit is received or not.

- SSPIF is one --- start bit is received
- SSPIF is zero --- start bit is not received

- Address is moved to SSPBUF
- SSPIF and ACKSTAT (it's a bit in SSPCON2 used as acknowledging indicator) are checked to know the address is received or not.
  - ACKSTAT is one --- not received
  - ACKSTAT is zero --- received

- Data is moved to SSPBUF
- Set RCEN as one to enable continuous reception
- SSPIF and ACKSTAT are checked again to find the data is received or not
- Stop bit is received just by setting PEN bit of the SSPCON2
- Finally, Poll the SSPIF to check either the stop bit is received or not.
- The above steps are continued for all data's.

5.4 USART

USART is the abbreviation for Universal synchronous asynchronous transmitter cum receiver. It's a protocol, which makes the communication between computer and micro controller. Fundamentally, protocol is nothing but a rules and regulations derived for the communication.

USART Registers

There are totally five registers, used in USART communication. Among that two are data registers and other two are control registers and one is baud rate register.

Data Registers

This registers holds the data during serial communication between computer and PIC.

TXREG: USED FOR TRANSMISSION

RXREG: USED FOR RECEPTION

Control Registers

These registers control the data flow and enable the USART communication. User has to set the bits of these registers before initiating the communication.

TXSTA : TRANSMISSION CONTROL REGISTER

RCSTA : RECEPTION CONTROL REGISTER

Control Registers

TXSTA: Transmit Status and Control Register

CSRC	TX9	TXEN	SYNC	-------	BRGH	TRMT	TX9D
bit 7							Bit 0

bit 7 **CSRC**:Clock Source Select bit

Asynchronous mode

Don't care

Synchronous mode

1 = Master mode (Clock generated internally from BRG)

0 = Slave mode (Clock from external source)

bit 6 **TX9**: 9-bit Transmit Enable bit

1 = Selects 9-bit transmission

0 = Selects 8-bit transmission

bit 5 **TXEN**: Transmit Enable bit

1 = Transmit enabled

0 = Transmit disabled

bit 4 **SYNC**: USART Mode Select bit

1 = Synchronous mode

0 = Asynchronous mode

bit 3 **Unimplemented:** Read as '0'

bit 2 **BRGH**: High Baud Rate Select bit

Asynchronous mode

1 = High speed

0 = Low speed

Synchronous mode

Unused in this mode

bit 1 **TRMT**: Transmit Shift Register Status bit

1 = TSR empty

0 = TSR full

bit 0 **TX9D:**

9th bit of transmit data. Can be parity bit

RCSTA: Receive Status and Control Register

SPEN	RX9	SREN	CREN	-----	FERR	OERR	RX9D

Bit 7 bit 0

bit 7 **SPEN**:Serial Port Enable bit

1 = Serial port enabled (Configures RX/DT and TX/CK pins as serial port pins)

0 = Serial port disabled

bit 6 **RX9**: 9-bit Receive Enable bit

1 = Selects 9-bit reception

0 = Selects 8-bit reception

bit 5 **SREN**: Single Receive Enable bit

Asynchronous mode

Don't care

Synchronous mode - master

1 = Enables single receive

0 = Disables single receive

This bit is cleared after reception is complete.

Synchronous mode - slave

Unused in this mode

bit 4 **CREN** : Continuous Receive Enable bit

Asynchronous mode

1 = Enables continuous receive

0 = Disables continuous receive

Synchronous mode

1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)

0 = Disables continuous receive

bit 3 **Unimplemented:**

Read as '0'

bit 2 **FERR**: Framing Error bit

1 = Framing error (Can be updated by reading RCREG register and receive next valid byte)

0 = No framing error

bit 1 **OERR**: Overrun Error bit

1 = Overrun error (Can be cleared by clearing bit CREN)

0 = No overrun error

bit 0 **RX9D**

9th bit of received data, can be parity bit.

Baud Rate Register

Baud rate (Bits / second,) This register is used to set the speed of USART communication.

Baud rate of our system: 2400

USART Transmission

- The seventh pin (RC6) of the Port C is dedicated for USART transmission. So, it is assigned as input by setting seventh pin of Tris C

- Baud rate for communication is set by SPBRG register. For that decimal value of twenty-five is move on to SPBRG.

- Bit SPEN is set to one (This is available in RCSTA) to enable serial data communication.

- Sending data 0x20 configures transmission control register TXSTA.
- In this register only TXEN pin is set to one to initiate transmission.

- Require data is moved to the TXREG.

- Finally TRMT pin is checked to know the transmission is completed or not

  TRMT IS ONE: Transmission is over

  TRMT IS ZERO: Transmission is not over

- USART transmitter is shown in Fig. 12.

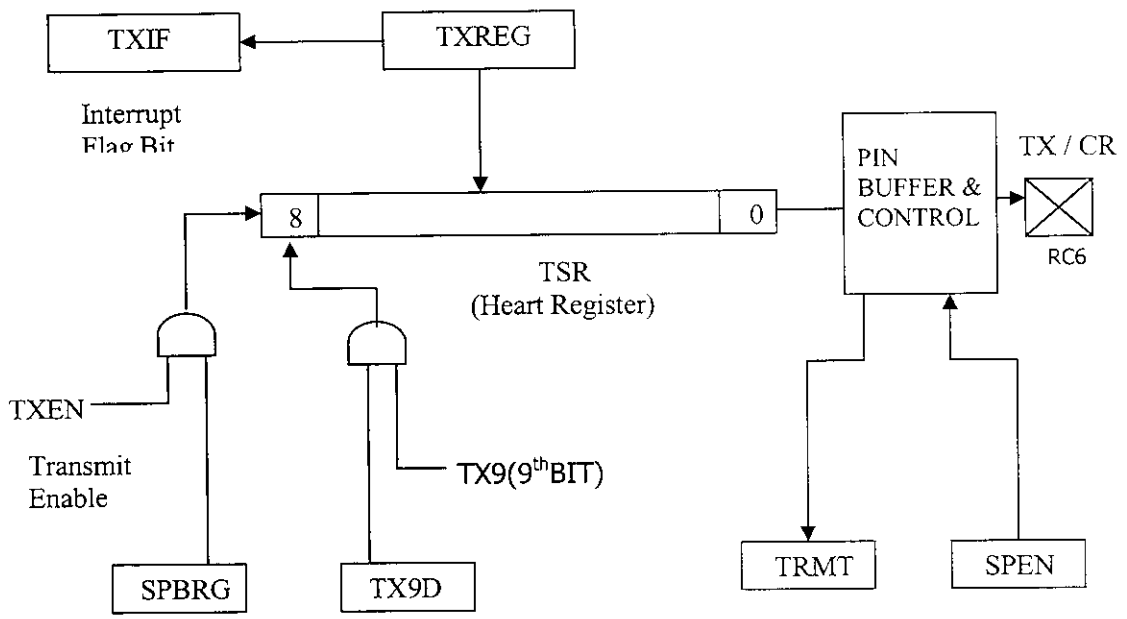


Fig: 12 USART Transmitter

USART Reception

- The eighth pin (RC7) of the Port C is dedicated for USART reception. So, it is assigned as input by setting eighth pin of Tris C.
- Baud rate for communication is set by SPBRG register. For that decimal value of twenty-five is move on to SPBRG.
- Bit SPEN is set to one (This is available in RCSTA) to enable serial data communication.
- The bit CREN (Continues receive enable bit) is set to one to enable continues data reception.

- Fig. 13 indicates the USART receiver.
- Poll the RCIF flag (flag in the PIR 1 register) to check either the reception is completed or not.

RCIF IS ONE --- Successively done

RCIF IS ZERO--- Not done.

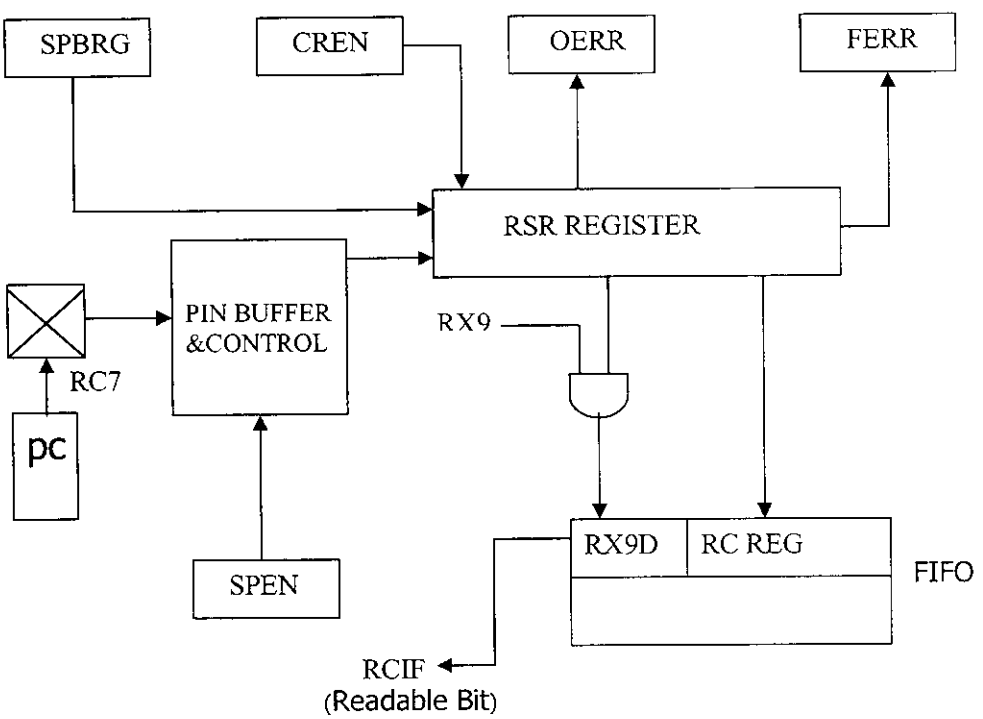


Fig: 13 USART Receiver

CONCLUSION AND FUTURE EXTENSION

6. CONCLUSION AND FUTURE EXTENSION

Thus the intelligent system measures voltage, number of products and temperature. Also the system automatically corrects the temperature by comparing it with the preset value. This is an industry-oriented system. The latest advancements are updated in this project to differentiate this from the existing system. The methodology used, is most comfortable for remote monitoring of industrial parameters. Many industries will adapt to this system, since it is economical, fast, and reliable.

The project can be extended with real time clock. This system helps to determine the time at which the temperature exceeds the set value and maximum voltage fluctuation. This will make the system more efficient. This user efficient system can be installed in packing machine, molding machine, thermal coating etc to maintain accurate temperature.

BIBLIOGRAPHY

7. BIBLIOGRAPHY

1. "Operational amplifier and linear integrated circuits", by James M.Fiore, Jaico publishing centre, 2002

2. "Electrical measurements fundamentals concepts applications", Martin U Reissland, 1989

3. "The complete guide to Visual Basic 6", by Clayton Walnum, Prentice Hall Of India Private Ltd, New Delhi, 1999.

4. "C with assembly language" Steven Holzner, BPB book centre, 1990

5. http://www.cardlogix.com/

6. www.microchip.com

APPENDIX

APPENDIX I

SOFTWARE CODING

Writing Into Smart Card

```
#include<PIC.h>
fail();
delay(char);
char i,reg=0,seg1=0,seg2=0,seg3=0,address,data,data1,data2;
main()
{
TRISB=0;
TRISC=0XD8;
PORTB=0;
PORTC=0;
SSPBUF=0;
reg=0;
SPBRG=25;
RCSTA=0;
TXSTA=0;
RCREG=0;
SPEN=1;
while(1)
{
SPBRG=25;
RCSTA=0;
TXSTA=0;
RCREG=0;
SPEN=1;
CREN=1;
```

```
while(!RCIF);
reg=RCREG;
CREN=0;
if(reg==0x31)
{
for(address=0;address<3;address++)
{
read();
transmit();
}
}
if(reg==0x32)
{
for(address=3;address<6;address++)
{
read();
transmit();
}
}
if(reg==0x33)
{
address=8;
read();
transmit();
}
if(reg==0X34)
{
CREN=1;
while(!RCIF);
CREN=0;
data=RCREG-0x30;
```

```
CREN=1;
while(!RCIF);
CREN=0;
data1=RCREG-0x30;
data=data*10+data1;
address=0x10;
write();
CREN=0;
}
if(reg==0X35)
{
address=0x10;
read();
transmit();
}
}
}
read()
{
SSPCON=0X28;
SSPSTAT=0X80;
SSPADD=100;
data=0;
SEN=1;
while(!SSPIF);
SSPIF=0;
SSPBUF=0XA0;
while(!SSPIF);
SSPIF=0;
WCOL=0;
while(ACKSTAT)
```

```
{
RSEN=1;
while(!SSPIF);
SSPIF=0;
SSPBUF=0XA0;
while(!SSPIF);
SSPIF=0;
WCOL=0;
}
SSPBUF=address;
while(!SSPIF);
SSPIF=0;
WCOL=0;
while(ACKSTAT);
RSEN=1;
while(!SSPIF);
SSPIF=0;
SSPBUF=0XA1;
while(!SSPIF);
SSPIF=0;
WCOL=0;
while(ACKSTAT);
RCEN=1;
RC1=1;
while(!SSPIF);
SSPIF=0;
data=SSPBUF;
/*ACKEN=1;
ACKDT=0;
while(!SSPIF);
SSPIF=0;
```

```
ACKEN=1;
ACKDT=1;
RC1=0;
while(!SSPIF);
SSPIF=0;*/
PEN=1;
while(!SSPIF);
SSPIF=0;
}

write()
{
SSPCON=0X28;
SSPSTAT=0X80;
SSPADD=100;
SEN=1;
while(!SSPIF);
SSPIF=0;
WCOL=0;
SSPBUF=0XA0;
while(!SSPIF);
SSPIF=0;
WCOL=0;
RB0=1;
while(ACKSTAT)
{
RSEN=1;
while(!SSPIF);
SSPIF=0;
WCOL=0;
SSPBUF=0XA0;
```

```
while(!SSPIF);
SSPIF=0;
WCOL=0;
}
SSPBUF=address;
while(!SSPIF);
SSPIF=0;
while(ACKSTAT);
SSPBUF=data;
while(!SSPIF);
SSPIF=0;
while(ACKSTAT);
PEN=1;
while(!SSPIF);
SSPIF=0;
}
transmit()
{
seg1=data/100;
data=data%100;
seg2=data/10;
seg3=data%10;
SPBRG=25;
RCSTA=0;
TXSTA=0;
SPEN=1;
TXEN=1;
TXREG=seg1+0x30;
while(!TRMT);
TXREG=seg2+0x30;
while(!TRMT);
```

```
TXREG=seg3+0x30;
while(!TRMT);
TXEN=0;
}
```

READING FROM SMARTCARD:

```
#include<PIC.h>
fail();
delay(char);
unsigned int result1,result,times=0;
char i,j,address,data,smwr,addr,count=0,settemp=0,volt;
main()
{
TRISA=0X1F;
TRISB=0;
TRISC=0X18;
ADCON1=0X80;
PORTB=0;
PORTC=0;
SSPBUF=0;
T2CON=0X25;
PR2=250;
TMR2ON=1;
addr=0;
smwr=0;
read();
INTCON=0XC0;
TMR2IE=1;
OPTION=0X28;
TMR0=0;
```

```
smwr=0;
address=2;
data=28;
write();
TMR0=0;
while(1)
{
voltage();
temperature();
if(smwr==1)
{
smwr=0;
address=addr;
data=result1;
write();
RB0=0;
address=addr+3;
data=result;
write();
address=8;
data=TMR0;
write();
RB7=1;
}
}
}

static void interrupt isr()
{
if(TMR2IF==0&&T0IF==0)
return;
```

```
if(TMR2IF==1)
{
TMR2IF=0;
times++;
if(times==250)
{
if(addr==3)
addr=0;
times=0;
addr++;
smwr=1;
}
}
if(T0IF==1)
{
count++;
T0IF=0;
}
}
temperature()
{
ADCON0=0X81;
ADCON1=0X81;
for(i=0;i<255;i++);
ADGO=1;
while(ADGO);
result=ADRESH*256+ADRESL;
result=result-140;
result=(result*10)/20;
if(result>settemp)
{
```

```
RB2=1;
/*smwr=0;
address=2;
data=result;
write();*/
}
if(result<(settemp-5))
RB2=0;
}
voltage()
{
ADCON1=0X80;
ADCON0=0X89;
result1=0;
for(j=0;j<50;j++)
{
for(i=0;i<50;i++);
ADGO=1;
while(ADGO);
result=ADRESH*256+ADRESL;
result=result/2;
result1=result+result1;
}
volt=result1/50;
}
write()
{
SSPCON=0X28;
SSPSTAT=0X80;
SSPADD=100;
SEN=1;
```

```
while(!SSPIF);
SSPIF=0;
WCOL=0;
SSPBUF=0XA0;
while(!SSPIF);
SSPIF=0;
WCOL=0;
RB0=1;
while(ACKSTAT)
{
RSEN=1;
while(!SSPIF);
SSPIF=0;
WCOL=0;
SSPBUF=0XA0;
while(!SSPIF);
SSPIF=0;
WCOL=0;
}
SSPBUF=address;
while(!SSPIF);
SSPIF=0;
while(ACKSTAT);
SSPBUF=data;
while(!SSPIF);
SSPIF=0;
while(ACKSTAT);
PEN=1;
while(!SSPIF);
SSPIF=0;
}
```

```
read()
{
SSPCON=0X28;
SSPSTAT=0X80;
SSPADD=100;
data=0;
SEN=1;
while(!SSPIF);
SSPIF=0;
SSPBUF=0XA0;
while(!SSPIF);
SSPIF=0;
WCOL=0;
while(ACKSTAT)
{
RSEN=1;
while(!SSPIF);
SSPIF=0;
SSPBUF=0XA0;
while(!SSPIF);
SSPIF=0;
WCOL=0;
}
SSPBUF=0x10;
while(!SSPIF);
SSPIF=0;
WCOL=0;
while(ACKSTAT);
RSEN=1;
while(!SSPIF);
SSPIF=0;
```

```
SSPBUF=0XA1;
while(!SSPIF);
SSPIF=0;
WCOL=0;
while(ACKSTAT);
RCEN=1;
RC1=1;
while(!SSPIF);
SSPIF=0;
settemp=SSPBUF;
/*ACKEN=1;
ACKDT=0;
while(!SSPIF);
SSPIF=0;
ACKEN=1;
ACKDT=1;
RC1=0;
while(!SSPIF);
SSPIF=0;*/
PEN=1;
while(!SSPIF);
SSPIF=0;
}
```

APPENDIX-II

PIN DIAGRAM