Zagazig University
Faculty of Engineering
Department of Electronics and Communication Engineering

Microcontroller Based Design of Digital Transmitters for Temperature Measurements in Reactors

By

Eng. Moteaa Abd El Hameed Mohammad Nassar

Supervised by

Prof. Dr. Mohammad Shaker Mohammad Ismail

Dr. Adel Zaghloul Mahmoud

2011

Acknowledgment

First of all, I am indebted to **ALLAH** who guided me to the right path and directed me to what is beneficial.

I would like to express my deepest thanks and appreciation to my supervisors:

Prof. **Mohammad Shaker,** Dr. **Adel Zaghloul** for their great supervision, outstanding, guidance, active support and cooperation to bring this work to success. I can not deny that without them this work might be different

I wish to express my deep thanks to engineer. **Refaat Mohammad Fikrey**, engineer. **Samir Adly** for their assistance and for suppling needed data during this research.

My special thanks should be extended to my **husband**, **my family** for their encouragement and support during the research.

Abstract

Temperature transmitter is one of the most important transmitters in the nuclear reactor it is used for RTD (Resistance Temperature Detector) signal conditioning. It has built-in current excitation, instrumentation amplifier, linearization and current output circuitry which amplifies the RTD signal and gives linearization to it. It is a part of a system to get temperature and monitoring it. This system is very cost and complicated. In this work a digital system is implemented by using microcontroller techniques that replaces the existing system, one chip (PIC16f877) is used to build a digital system, which is more accurate and give more performance and low costs.RTD is the sensing element of temperature, its resistance increases with temperature**.** There are many types of transmitters in the reactor such as temperature, pressure, level and flow but temperature one is chosen because of temperature is one of the most important parameters in process control. Accurate measurements of the temperature are not easy and to obtain accuracies better than 0.5 ^{o}C great care is needed. PIC (Programmable Interface Controller) microcontroller is used for interfacing the transmitter for its major advantages, the built-in A/D converter and EEPROM of PIC16f 877 is used in our research, the PIC is programmed by using C language, Mikro-C compiler and PROTEUS are used for designing the circuit and simulating it. The temperature is read in digital form and displayed on LCD, stored the upper limit, lower limit in the EEPROM of the PIC.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABRIVIATIONS

A/D:	Analog / Digital
ADC:	Analog to Digital Converter
ALU:	Arithmetic-Logic Unit
ARPCS:	Automatic Reactor Power Control System
CCP:	Capture and Compare Module
CDU:	Conditioned Unit
CISC:	Complex Instruction Set Computer
CPU:	Central Processing Unit
CWIS:	Chimney Water Injection System
DAC:	Digital to Analog Converter
DSP:	Digital Signal Processing
EEPROM:	Electrically Erasable Programmable Read Only Memory
EVA:	Evacuation Alarm
GPR:	General Purpose Registers
GPRS:	General Packet Radio Service
IC:	Integrated Circuit
I& C:	Instrumentation and Control
I/O:	Input/output
LCD:	Liquid Crystal Display
LED:	Light Emitting Diode
MCU:	Microcontroller Unit

MPR: Multiple Purpose Reactor

MSSP: Master Synchronous Serial Port

OP-AMP: Operational Amplifier

PCB: Printed Circuit Board

PDL: Program Description Language

PIC: Programmable Interface Controller

RAM: Random Access Memory

RISC: Reduced Instruction Set Computer

ROM: Read Only Memory

RPS: Reactor Protection System

RTD: Resistance Temperature Detector

SBC: Single Board Computer

SCS: Supervision & Control System

SEEPROM: Serial Electrically Erasable Programmable Read Only Memory

SFR: Special Function Registers

SSIU: Safety Setting Input Unit

SSS : Safety System Setting

TU: Trip Unit

USART: Universal Synchronous/Asynchronous Receiver And Transmitter

VPLU: Voting & Protective Logic Unit

CHAPTER 1
INTRODUCTION

In this study, a digital system is implemented by using microcontroller techniques; temperature is taken as an example. Digital systems have more advantage than the analog one; they are more accurate and give more performance and low costs. There are many types of transmitters in the reactor but temperature one is chosen in this work because temperature measurements and control are vital in many industrial processes and accurate control of the temperature is not easy, resistance temperature detector (RTD) sensors are used for sensing temperature because they are considered to be the most accurate and stable sensors.

C Language is used in programming microcontroller; mikroC [1] compiler (mikro -C Version (8pt2) software) is used for compiling the program, mikroC is a powerful, feature rich development tool for programmable interface controller (PIC) micros. It is designed to provide the programmer with the easiest possible solution for developing applications for embedded systems, without compromising performance or control.

PROTEUS is used for designing and simulating the system (version 7pt4 sp3). The strength of its architecture has allowed us to integrate first conventional graph based simulation and now - with PROTEUS VSM - interactive circuit simulation into the design environment. For the first time ever it is possible to draw a complete circuit for a micro-controller based system and then test it interactively, all from within the same piece of software. Meanwhile, ISIS retains a host of features aimed at the PCB design, so that the same design can be exported for production with ARES or other PCB layout software.

Programmable Interface Controllers (PICs) [2] are a family of microcontrollers by microchip technology. PIC microcontrollers have attractive features and they are suitable for a wide range of applications. The PIC was the first widely available device to use flash memory, which makes it ideal for experimental work. Flash memory allows the program to be replaced quickly and easily with a new version. It is now commonplace, not least in our USB memory sticks, but also in a wide range of electronic systems where user data need to be retained during power down. Cheap flash memory microcontrollers have transformed the teaching of microelectronics – they are re-usable and the internal architecture is fixed, making them easier to explain. It is one of the most important developments in electronics since the invention of the microprocessor itself. It can be used in many applications, used to implement GPRS [3] based positioning system. The system is implemented using the microchip PIC 16F877A microcontroller. The small instruction set of the PIC is also a major advantage – only 35 instructions to learn. Compare that with a complex processor such as the Pentium, which is quite terrifying compared with the PIC. The quality of the PIC technical documentation is also a major factor. Microcontrollers contain all the components required for a processor system in one chip: a CPU, memory and I/O. A complete system can therefore be built using one MCU chip and a few I/O devices such as a keypad, display and other interfacing circuits.

For these reasons, the PIC was used into our search. PIC 16F877 is used. This is now used widely as a more advanced teaching device, because it has a full complement of interfaces: analog input, serial ports, slave port and so on, plus a good range of hardware timers.

1.1 Practical Temperature Measurements

Temperature measurements and control are vital in many industrial processes. Accurate control of the temperature is essential in nearly all chemical processes. In some applications, an accuracy of around 5-10 ^{o}C may be acceptable. There are also some industrial applications which require accuracy temperature 4-1 ^{o}C accuracy. Temperature sensors come in many different forms and a number of techniques have evolved for the measurement of temperature. There are new forms of sensors which require no contact with the medium whose temperature is to be sensed. The majority of sensors still require touching the solid, liquid, or the gas whose temperature is to be measured.

1.1.1 Types of Temperature Sensors

There are many types of sensors to measure the temperature [2]. Some sensors such as the thermocouples, RTDs, and thermistors are the older classical sensors and they are used extensively due to their big advantages. The new generation of sensors such as the integrated circuit sensors and radiation thermometry devices are popular only for limited applications. The choice of a sensor depends on the accuracy, the temperature range, speed of response, thermal coupling, the environment (chemical, electrical, or physical), and the cost.

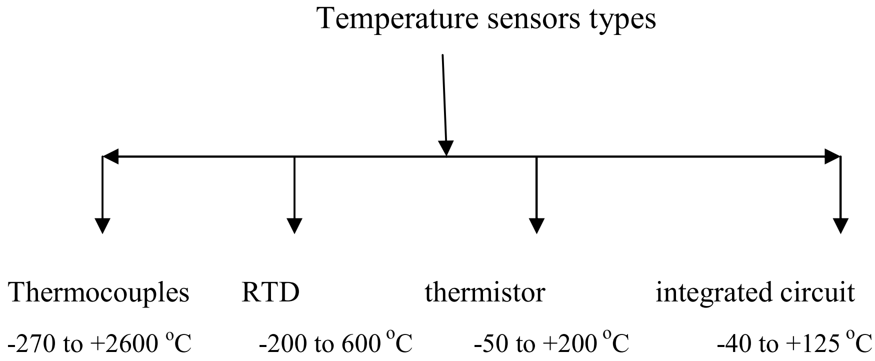
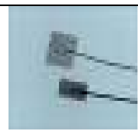


Fig. (1.1) Temperature sensors types

As shown in Figure (1.1), thermocouples are best suited to very low and very high temperature measurements. The typical measuring range is -270 °C to +2600 °C Thermocouples are low cost and very robust. They can be used in most chemical and physical environments. External power is not required to operate them and the typical accuracy is ±1°C. RTDs are used in medium range temperatures, ranging from -200 °C to +600 °C they offer high accuracy, typically ±0.2°C, RTDs can usually be used in most chemical and physical environments, but they are not as robust as the thermocouples. The operation of RTDs [4] require external power. Thermistors are used in low to medium temperature applications, ranging from -50°C to+200°C they are not as robust as the thermocouples or the RTDs and they can't easily be used in chemical environments. Thermistors are low cost and their accuracy is around

±0.2^{o}C. Semiconductor sensors are used in low temperature applications, ranging from -40 ^{o}C to about +125 ^{o}C Their thermal coupling with the environment is not very good and the accuracy is around +1^{o}C Semiconductors are low cost and some models offer digital outputs, enabling them to be directly connected to computer equipment without the need of A/D converters. Radiation thermometry devices measure the radiation emitted by hot objects, based upon the emissivity of the object. But the emissivity is usually not known accurately, and additionally it may vary with time, making accurate conversion of radiation to temperature difficult. Also, radiation from outside the field of view may enter the measuring device, resulting in errors in the conversion. Radiation thermometry devices have the advantages that they can be used to measure temperatures in a wide range (450 ^{o}C to 2000 ^{o}C with accuracy better than 0.5%. Radiation thermometry requires special signal processing hardware and software and is not covered in this work. The advantages and disadvantages of various types of temperature sensors are given in Table 1.1.

Table 1.1 Comparison of temperature sensors

sensor	advantages	disadvantages	Sensor shape
Thermocouple	Wide operating Temperature range Low cost Rugged	Nonlinear Low sensitivity Reference junction compensation required Subject to electrical noise	
RTD	Linear Wide operating Temperature range High stability	Slow response time Expensive Current source required Sensitive to shock	
Thermistor	Fast response time Low cost Small size Large change in resistance vs temperature	Nonlinear Current source required Limiting operating temperature range Not easily interchangeable without recalibration	
Integrated circuit	Highly linear Low cost Digital output sensors Can be directly connected to a microprocessor without an A/D converter	Limited operating temperature range Voltage or current source required Self heating errors Not good thermal coupling with the environment	

1.1.2 Measurement errors

There could be several sources of errors during the measurement of temperature. Some important errors are described in this section.

1.1.2.1 Sensor self heating

RTDs, thermistors, and semiconductor sensors require an external power supply so that a reading can be taken. This external power can cause the sensor to heat, causing an error in the reading. The effect of self heating depends on the size of the sensor and the amount of power dissipated by the sensor. Self heating can be avoided by using the lowest possible external power, or by calibrating the self heating into the measurement.

1.1.2.2 Electrical noise

Electrical noise can introduce errors into the measurement. Thermocouples produce extremely low voltages and as a result of this, noise can easily enter into the measurement. This noise can be minimized by using low-pass filters, avoiding ground loops, and keeping the sensors and the lead wires away from electrical machinery.

1.1.2.3 Thermal coupling

It is important that the sensor used makes a good thermal contact with the measuring surface. If the surface has a thermal gradient (e.g. as a result of poor thermal conductivity) then the placement of the sensor should be chosen with care. If the sensor is used in a liquid, the liquid should be stirred to cause a uniform heat distribution. Semiconductor sensors usually suffer from good thermal contact since they are not easily mountable to the surface whose temperature is to be measured.

1.1.2.4 Sensor time constant

This can be another source of error. Every type of sensor has a time constant such that it takes time for a sensor to respond to a change in the external temperature. The time constant is defined as the time it takes for the output to reach 63% of its final steady-state value. Errors due to the sensor time constant can be minimized by improving the thermal coupling, or by using a sensor with a small time constant.

1.1.2.5 Sensor leads

Sensor leads are usually copper and therefore they are excellent heat conductors. These wires can lead to errors in measurements if placed in an environment with a temperature different to the measured surface temperature. These errors can be minimized by using thin wires, or by taking care in placing the lead wires.

1.1.3 Selecting a temperature sensor

Selecting the appropriate sensor is not always easy. This depends on factors such as the temperature range, required accuracy, environment, speed of response, ease of use, cost, interchangeability and so on. Traditionally, thermocouples are used in high temperature chemical industries such as glass and plastic processes. Environmental applications, electronics hobby market, and automotive industries generally use thermistors or integrated circuit sensors. RTDs are commonly used in lower temperature, higher precision chemical industries.

1.2 Introduction to Microcontrollers

Today, we see many industrial and domestic products like remote controllers, telephone bill printing machines, automatic power regulators, microwave ovens, engines, indicating and measuring instruments and similar products. Automation is needed to facilitate the process or mechanism for its operation and control. Data storage and processing is an integral part of any automatic control system. The need is to have a device, so called 'microcontroller', [5] which allows controlling the timing and sequencing of these machines and processes. Further, with the help of microcontroller, it is possible to carry out simple arithmetic and logical operations. Any system that has a remote controller, almost certainly contains a microcontroller.

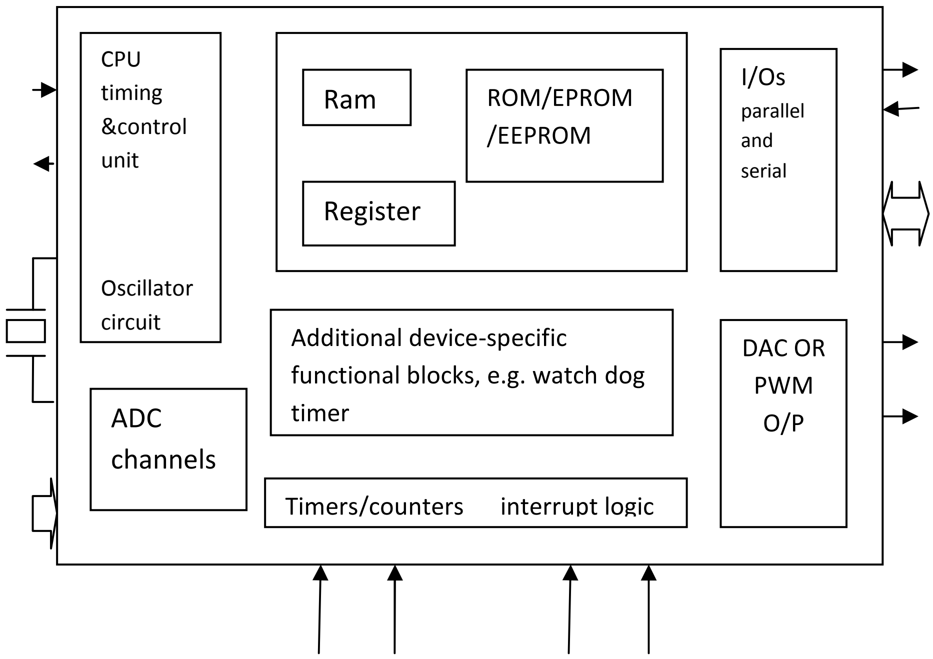


Fig. (1.2) General Block Diagram of a Microcontroller.

All these functional blocks on a single integrated circuit (IC) results into a reduced size of control board, low power consumption, more reliability and ease of integration within an application design.

Microcontrollers are single-chip microcomputers, more suited for control and automation of machines and processes. Microcontrollers have central processing unit (CPU), memory, input/ output ports (I/O), timers and counters, analog- to-digital converter (ADC), digital-to-analog converter (DAC), serial ports, interrupt logic, oscillator circuitry and many more functional blocks on chip.

Figure (1.2) shows a general block diagram of a microcontroller. Automation, and provides more flexibility. The device can be programmed to make the system intelligent. This is possible because of the data processing and memory capability of microcontrollers .some of the commonly used microcontrollers are Intel MCS-51,MCS-96,Motrola68HC12 family, microchips peripheral interface controller(PIC) family of microcontrollers 16CXX,17CXX,etc.

1.2.1 Embedded versus external memory devices

Embedded devices are becoming very popular now-a-days. Such a device [6] has all functional blocks on chip, including the program and data memory. There is no external data / address bus provided. For example, ATMEL 89C2051 is one example of embedded controller, which has timers/ counters, on-chip RAM, EEPROM,I/Os, a precision comparator along with CPU and timing and control unit. The code is executed from the internal program memory only however; we see normally that devices like 8031 from MCS-51 family need external program memory interface. These devices are external memory devices.

1.2.2 8-BIT and 16-BIT Microcontrollers

Several features define the word length of a processor. One can define an 8-bit microcontroller in a much broader sense, as a device which has most of its registers 8-bit wide. There is almost no direct dependency of this definition on the width of the data address bus. For example, 8088 has8-bit microcontroller family. MCS-96 is a 16 bit microcontroller family. From the application point of view, it is very important to decide if an 8-bit or a 16-bit microcontroller is to be used in a typical design. 8-bit microcontrollers have dominated over the 16-bit microcontrollers. The reason that many designers are familiar with 8-bit microcontrollers and 16-bit operations can always be performed on 8-bit controllers, by writing suitable programs.

1.2.3 CISC and RISC processors

Complex instruction set computers (CISC) and reduced instruction set computers (RISC) are common terminologies used while talking about microcontrollers or microprocessors. CISC processors have large number of instructions. A larger instruction set helps assembly language programmers by providing flexibility to write effective and short programs. The objective of CISC architecture is to write a program in as few lines of assembly language code as possible. This is made possible by developing processor hardware that can understand and execute a number of operations. Programmers want to have fewer, simpler and faster instructions, than the large, complex and slower CISC instructions. This is however, at the cost of writing more instructions to accomplish a task. One advantage of RISC chips require lesser hardware implementation, which makes them simpler to design and hence lesser cost of production. And it is easier to write optimized compilers, because of a small number of instructions. One example of RISC microcontrollers is the popular PIC family of microcontrollers by microchip.

1.2.4 Harvard and Von Neumann architectures

There are two major classes of computer architectures, Harvard architecture and von Neumann architecture. Many special designs of microcontrollers and DSP use Harvard architecture. Harvard architecture uses separate memories for program and data with their independent address and data buses. Because of two different streams of data and address, there is no need to have any time division multiplexing of address and data buses. Not only the architecture supports parallel buses for address and data, but also it allows a different internal organization such that instruction can be pre-fetched and decoded while multiple data are being fetched and operated on. Further, the data bus may have different size than the address bus. This allows the optimal bus widths of the data and address buses for fast execution of the instruction. PIC microcontrollers by microchip use Harvard archicteure. In von Neumann architecture, programs and data share the same memory space. Von Neumann architecture allows storing or modifying the programs easily. However, the code storage may not be optimal and requires multiple fetches to form the instruction. Program and data fetches are done using time division multiplexing which affect the performance.

This thesis consists of five chapters,

Chapter 2: introduce a multiple purpose reactor system, its configuration, its main systems, instrumentation, and the existing circuit of temperature transmitter,

Chapter 3: introduce RTD sensor, its principles, its theory of operation and its practical circuit,

Also, about PIC microcontrollers architecture, their registers, their memories and their instructions,

And also, about analog to digital and real-time clocks, A/D converters in general, A/D modules in microcontroller ,its registers and how to implement it in a program ,

Chapter 4: introduce the simulation circuit, system design, results analysis and experimental work.

Chapter 5: gives the conclusion and references.

CHAPTER 2

MULTIPLE PURPOSE REACTOR SYSTEM

2.1 INSTRUMENTATION and CONTROL

From the (instrumentation and control) I&C point of view, the (Multiple Purpose Reactor) MPR, includes two main systems:

a) Supervision & Control System (SCS)

The SCS includes all the components required for reactor [7] process control during normal operation and plant incidents.

b) Reactor Protection System (RPS)

The RPS encloses all electrical and mechanical devices and circuits involved in generating those initial signals associated with protective functions [8] that are carry out by the safety systems actuation. Figure (2.1) shows a context diagram. A defense in depth strategy is followed. As related to the control of the reactor power the following level of actions [9] are implemented by the SCS;

- Automatic regulation to compensate deviation from a preset set point is carried out by the Automatic Reactor Power Control System (ARPCS)
- Alarms are triggered to call for operator attention
- If certain conditions are not satisfied, two limitation actions are executed;
  - Power reduction
  - Automatic insertion of the four safety rods

Independent of the SCS, the RPS will shutdown the reactor when any of the safety system settings of the first or the second shutdown systems is reached.

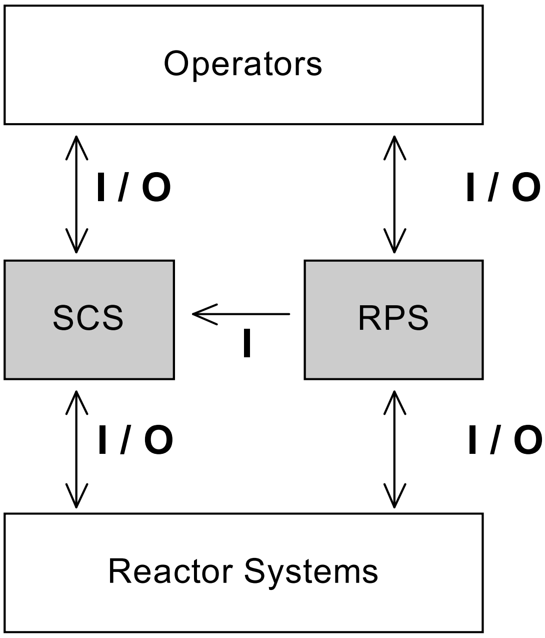


Fig. (2.1) I&C Context Diagram
SCS (Supervision & Control System), RPS (Reactor Protection System)

2.2 Reactor protection system(RPS)

The RPS design follows two basic requirements:

- It provides an independent and reliable system that monitors the safety variables and initiates appropriate protective actions if any of such variables reaches the Safety Setting.
- The Reactor Protection System brings the reactor to a safe condition in the event of anticipated operational occurrence (incidents) and accident conditions.

Architecture

The Reactor Protection System is configured with three independent redundancies; each is composed of three major devices as can be seen in Figure (2.2).

- Sensors and transducers
- Conditioning modules
- Trip Unit

Each redundancy performs the monitoring of the safety variables. There is a measurement channel [10] for each safety variable. Different components (e.g. transducers), measurement principles and methods are employed according to the process variable to be monitored. The measurement channels may be grouped as follows:

- Nuclear channels: Start-Up and Power channel.
- Process channels to monitor the thermal-hydraulic variables.
- Radioprotection channels to monitor area radiation.
- Special purpose channels: seismic, test loop facilities monitoring and miscellaneous.

In addition the system includes two redundant and independent gating logic units per protective action to form the actuation signals. These units are called Voting and Protective Logics and are also shown in Figure (2.2).

The output signals of the logic units, depending on their logical status, triggers the protective actions to be carry out by the safety actuation systems.

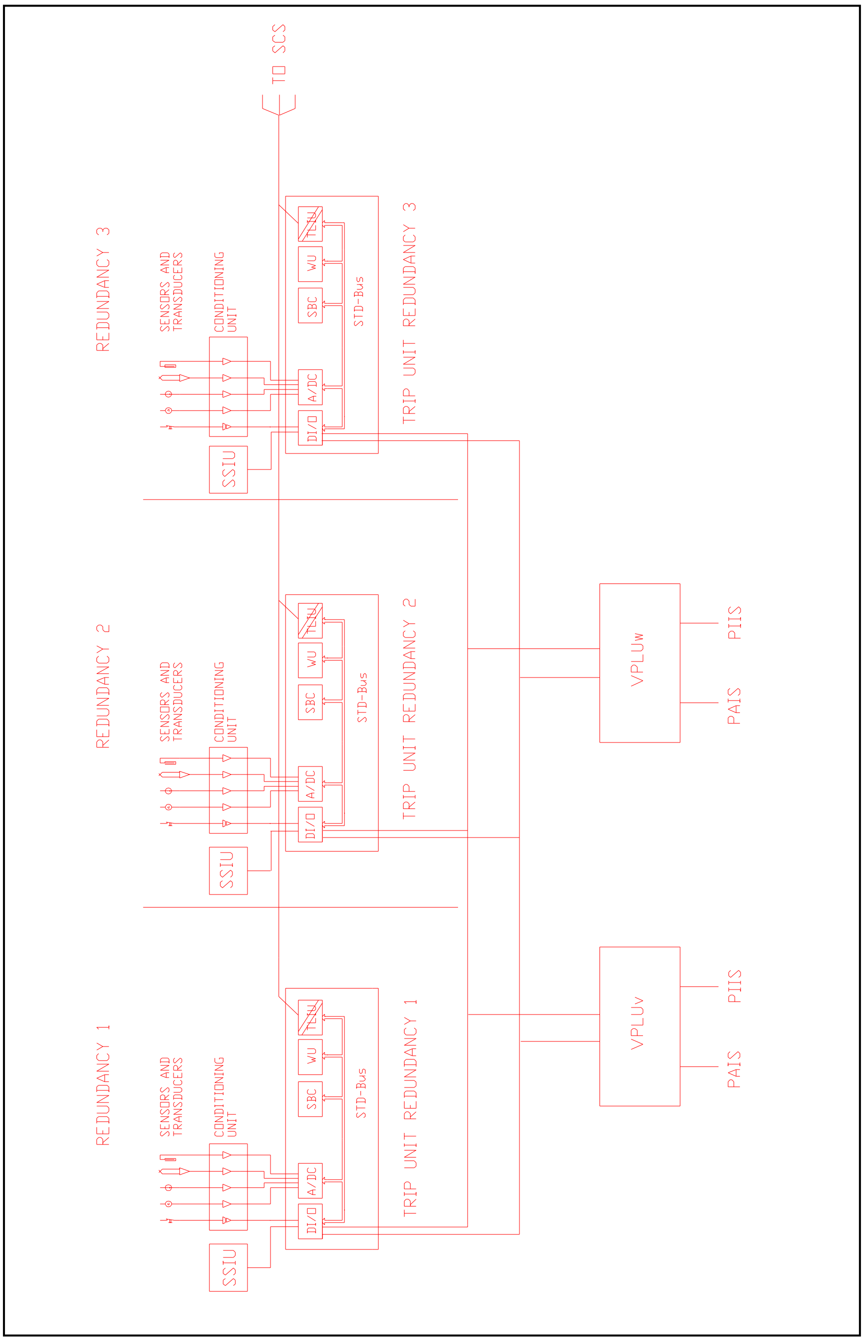


Fig. (2.2) Reactor Protection System.

The RPS also includes the protection interlocks to supervise the operational configuration changes of the safety systems.

The power requirements of the RPS are satisfied by means of three uninterrupted power supplies (one per redundancy).

Sensors and transducers

The sensors and transducers acquire the safety process [11, 12] variables from the plant and convert them to electrical signals. They are used as inputs for the initiation of the protective actions or protection interlocks.

Conditioning Modules

The Conditioning Modules perform the interface between the sensors and transducers and the digital/analog inputs of the Field Units. Those secondary variables that are derived from the signals of the transducers are obtained by different types of conditioning modules.

The Trip Unit (TU) is a microprocessor based data acquisition [13] and comparison device that acquires the safety variables and compares them with the safety system settings. Based on the result of the comparison, individual trip outputs, for each signal of a safety variable, are fed by the TU into the Voting and Protective Logic Unit (VPLU) in order to initiate a protective action accordingly. The TU is configured around a Single Board Computer (SBC). The communication link between the SBC and the Input / Output modules is the STD-Bus (IEEE 961 Standard Microcomputer Bus).

The Trip Units Figure (2.3) has the following modules interconnected via the STD-Bus:

- SBC based on the 80286 microprocessor
- Digital Input / Output module
- Analog to digital converter
- Field Bus communication channel interface

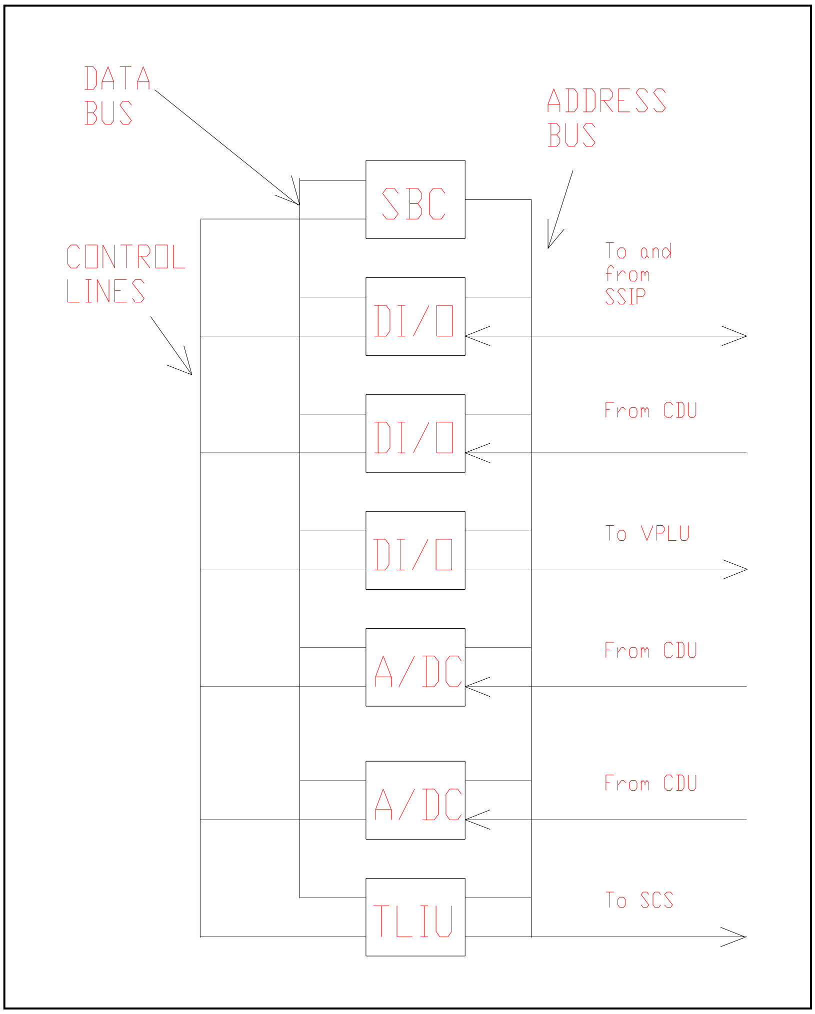


Fig. (2.3) Trip Unit Architecture.

The TU is configured as an embedded system [13] where the signals processing sequence is stored in a standalone program. The physical support is a non-volatile type memory (ROM) to preclude modifications, even by the CPU itself, and to assure very fast access time and high reliability of the storage media. The intermediate variables are stored in read/write type memory (RAM). Error detection codes are used to supervise the integrity of the program and data stored in the ROM / RAM memory.

The main functions of TU consist of:

- Signals' acquisition from the conditioning modules
- Processing the analog signals to convert them to digital format
- Carrying out the comparisons; first with the admissible electrical levels and after that, with the Safety System Settings.
- Transmission, via individual outputs, the result of the comparisons to the voting and protective logic.

The SBC includes a watchdog timer that must be strobed within certain time period. An external watchdog is also provided to lead the whole system to a failsafe condition in case of a critical failure; in this case an SCRAM is requested. The safety system settings are set in the Safety Settings Input Panel (SSIU) and are transferred to a predefined RAM memory bank when the TU is powered on. The transfer is done via a special purpose parallel port. The SSIU is a peripheral device connected to the TU and it is used to set the safety system settings to their trigger thresholds. It contains a front panel with numeric inputs (in engineering units) for each safety variable.

When the TU reads the Safety System Settings (SSS) it verifies that they are within the admissible ranges predefined. The storage of the SSS is an off-line operation and the TU disables the modifications of the settings while the reactor is in operation mode.

The transfer of information between the TU and the SCS is done through a dual port memory. The transference of information is only one way (TU → SCS). All the address, control and data lines that link the TU with the dual port memory are connected through optical isolators. This means there is no electric connection between the TU and the Dual Port Memory (SCS).

Voting and Protective Logic Unit (VPLU)

The VPLU carries out the processing of the logic signals, coming from the field unit, to generate the trigger signals for the safety actuation systems.

2.2.1 Conventional instrumentation

Conventional instrumentation is the instrumentation [14] used to measure variables other than seismic and radiation. This instrumentation includes sensors to measure:

- Pressure
- Differential pressure
- Water level
- Flow
- Temperature
- Differential temperature
- Valve position

The measurements for the safety system are done by three redundant sensors. The signals from sensors belonging to the safety chains are sent to the interconnection terminal panel of the trip unit by three different paths in order to avoid the occurrence of common mode failures. The signals from redundant sensors will be handled by coincidence logic with a two out three success criterion. This coincidence logic is implemented in the VPLU .In those cases where measurements are sent both to the safety logics and to the supervision and control system, sensors and transmitters common to both systems are used, a galvanic isolation is done from the signal going to the control system. The signals originated in the transmitters are wired up to analogical input modules belonging to the corresponding trip unit. The signals from field contacts are sent directly to digital conditioning module. The electric paths and process measurements belonging to each of the redundant channels of a same measurement, are different, in order to avoid common cause failures.

Field instruments

The instruments used in the reactor, except otherwise stated, are of the general purpose industrial type. The field transmitters [15] are of the "two wires" type, with an output current of 4 to 20 mA. The electric connection of the transmitters up to the field connectioning box is done using instrumentation standard materials. General purpose instrumentation is used except in those applications when higher specifications are required. When this occurs, special specifications are recorded on the Instrument Data sheet.

For under water measurements made in the Reactor Pool, electrical signals are wired over mounting plates on the wall until they reach the instrumentation channel Stainless steel sheathed with alumina isolation is used within the Reactor Pool under the pool water level Once inside the instrumentation channel, each redundant measurement is laid out by a different path until it reach the junction box at the open end of pool level. All accessories used for signal's lay out into the reactor pool are made of stainless steel.

Parameters measured

The following variables are measured to feed into the supervision and control system and the reactor protection system field units:

* Core inlet coolant temperature
* Core outlet coolant temperature
* Differential temperature through the core
* Differential pressure through the core
* Core cooling circuit flap valves position
* Reactor tank water level
* First shutdown system air pressure
* Second shutdown system Gadolinium tanks level
* Second shutdown system Gadolinium tanks pressure
* Chimney water injection water tanks level

Figure (2.4) presents a P&I diagram of the conventional instrumentation and signals related to the core and core cooling system. This Figure shows the importance of temperature in the reactor system.

- **Temperature measurement materials**

  Detectors to be used are stainless steel sheathed, Alumina isolated RTD.

- **Core inlet coolant temperature**

  Core inlet coolant temperature is measured at the inlet coolant collector.

Detectors used are PT-100 thermo Three redundant measuring channels are provided. Each redundant measuring channel includes a PT-100 resistance temperature detector specified in accordance to DIN 43760 Class A.

The measurement range is	0 °C / 100 °C.
Permitted deviation	better than ± 0.8 °C
Response time (typical)	5 sec

PT-100 temperature detectors are connected to RTD to current converters at the Conditioning Unit. These converters provide a 4 to 20 mA output signal.

The current signals provided by the RTD detectors are wired to the Trip Unit and processed by Analog Input modules. The TU each incoming signal is processed by the ADC. The result of the comparison against the Safety System Setting is now sent to the Voting and Protective Logic Unit (VPLU).

- **Core outlet coolant temperature**

  Core outlet coolant temperature is measured at the control rod guide box level.

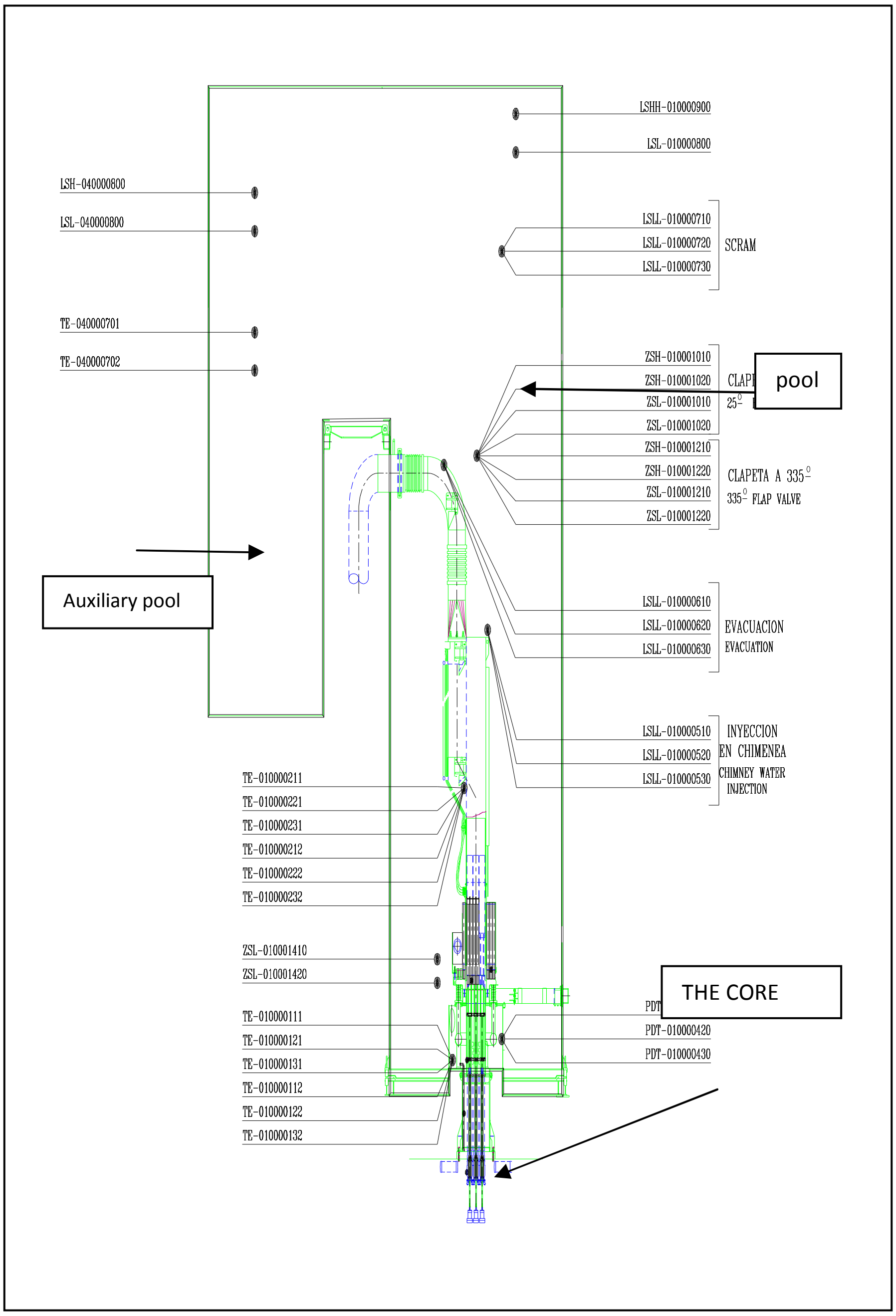


Fig. (2.4) Core Related Conventional Instrumentation.

Detectors used are of the PT-100 RTD type, specified in accordance to DIN 43760 Class A. Three redundant channels are used.

The measurement range is	0 °C / 100 °C.
Permitted deviation	better than ± 0.8 °C
Response time (typical)	5 sec

Signals delivered by RTD are processed by RTD to current converters at the CDU. Out coming signals are wired to AI Module at the TU, then processed by the ADC and after signal isolation sent by a different path to the VPLU.

- **Core coolant differential temperature (out-in)**

  Core inlet coolant temperature is also measured into the inlet coolant collector for core differential temperature measurement purposes. Core outlet coolant temperature for differential temperature purposes, is measured in the same position that those used for outlet coolant temperature.

  Detectors used are PT-100 thermo resistances, specified in accordance to DIN 43760 Class A. Three redundant measuring channels are provided.

The measurement range is	0 °C / 40 °C
Accuracy	better than 1°C
Response time (typical)	5 sec

Core outlet coolant temperature for differential temperature is measured at the control rod guide box level. Three redundant measuring channels are used for differential temperature measurement.RTD detectors output signals are processed by differential temperature to current converters at the CDU.

Converter's output signals are wired to the TU where are processed by the ADC, then to the VPLU.

2.3 Transmitter circuit in MPR

In MPR the temperature transmitter consists of the RTD sensor and the transmitter which mainly consists of XTR103 (a special IC) which is used for RTD signal conditioning. It has built-in current excitation, instrumentation amplifier linearization and current output circuitry on a single chip.

(XTR103)

Fig. (2.5) Transmitter Circuit for RTD Signal conditioning.

Figure (2.5) shows the basic circuit of the transmitter of the MPR system, as seen the transmitter has special integrated circuit (XTR103) and some resistors adjusted to achieve the output current from 4mA to 20mA.

- **General block diagram of MPR Conventional Instrumentation System:**

As seen in Figure (2.6) there are many sensors in the reactor; temperature, pressure, level and flow. Signals from sensors goes to transmitters , the outputs of them are 4 to 20 mA ,this current is taken across resistor to give voltage after that this voltage is taken to a analog input module(EL-085) to get 0 to 5 voltage range to be suitable for the analog to digital converter then to interface to see values on displays. This is a general brief descripsion of the reactor system.

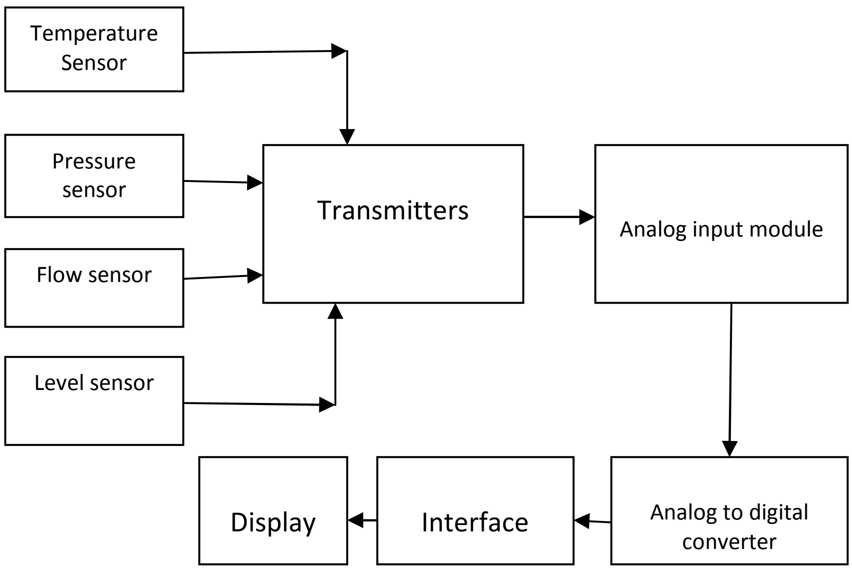


Fig. (2.6) General Block Diagram of MPR Conventional Instrumentation System.

2.4 MPR temperature system

Figure(2.7) shows the block diagram of MPR temperature system, the output of the transmitter is 4 to 20 mA , The current signals provided by the RTD detectors are wired to the Trip Unit and processed by Analog Input modules (EL-085) to get 0 to 5 voltage range to be suitable for the analog to digital converter which is(VL1295) in this system.This converted voltage goes to a microprocessor which has a software to convert it to a temperature degrees seeing it on displays. This is a breif description of the existing MPR temperaure system.

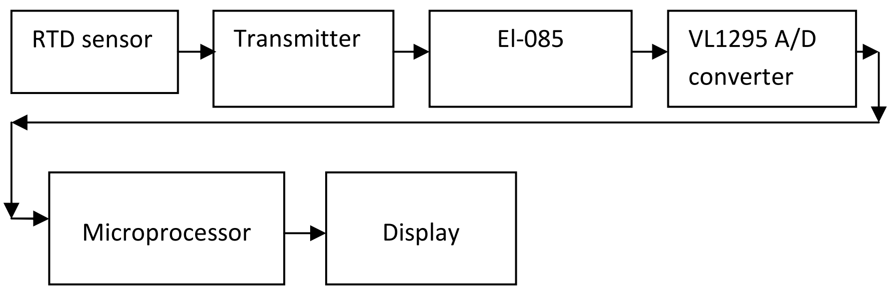


Fig. (2.7) Block Diagram of Multiple Purpose Reactor Temperature System.

CHAPTER 3

SYSTEM ELEMENTS DESCRIPTION

3.1 RTD Temperature Sensors

3.1.1 RTD principles

Resistance Temperature Detector (RTD) is a temperature sensing device whose resistance increases with temperature. It operates on the principle that the electrical resistance of metals change with temperature. Figure (3.1) shows the sensor of the RTD

RTD sensor

Fig. (3.1) RTD Temperature Sensor

Planar resistance temperature detector (RTD) can be manufactured with microelectronics processing techniques [16]. However, the manufactured planar resistor requires an extra step for adjustment of the 0degC reference resistance R_0.

In practice metals with high melting points which can withstand the effects of corrosion, and those with high resistivity are used for temperature sensing. The resistivity of some commonly used RTD metals are indicated in Table 3.1[17].

Table 3.1 Resistivity of Some RTD Metals at (T= 0)

metal	silver	copper	gold	tungsten	nickel	platinum
Resistivity ($10^{-8}\Omega$.m)	1.467	1.54	2.05	4.82	6.16	9.6

From this table we can be noticed that;

- Gold and silver have low resistivities and as a result their resistances are relatively low, making the measurement difficult.
- Copper has low resistivity but is sometimes used because of its low cost.
- The most commonly used RTDs are made of nickel, platinum, or nickel alloys.

- Nickel sensors are used in cost sensitive applications such as consumer goods and they have a limited temperature range.
- Nickel alloys, such as nickel-iron is lower in cost than the pure nickel and in addition it has a higher operating temperature.
- Platinum is by far the most common RTD material, mainly because of its high resistivity and long term stability in air.

RTDs have excellent accuracies over a wide temperature range and some RTDs have accuracies better than 0.001 ^{o}C with drift less than 0.1^{o}C/year

RTDs are difficult to measure [18] because of their low resistances and only slight changes with temperature, usually in the order of 0.40 $\Omega/^{o}$c.

RTDs are resistive devices and a current must pass through the device so that the voltage across the device can be measured. This current can cause the RTD to self-heat and consequently it can introduce errors into the measurement. This self-heat can be minimized by using the smallest possible excitation current. The amount of self-heat also depends on where and how the sensor is used. An RTD can self-heat much quicker in still air than in a moving liquid.

To accurately measure such small changes in resistance, special circuit configurations are usually needed. Because, long leads could cause errors as it introduces extra resistance to the circuit.

In order to achieve high stability and accuracy, RTD sensors must be contamination free. Below about 250̊c the contamination is not much of a problem, but above this temperature, special manufacturing techniques are used to minimize the contamination of the RTD element.

The RTD sensors are usually manufactured in two forms:

* Wire wound RTDs are made by winding a very fine strand of platinum wire into a coil shape around a non-conducting material (e.g. ceramic or glass) until the required resistance is obtained.

The assembly is then treated to protect short-circuit and to provide vibration resistance.

Although the wire wound RTDs are very stable, the thermal contact between the platinum and the measured point is not very good and results in slow thermal response.

* Thin film RTDs are made by depositing a layer of platinum in a resistance pattern on a ceramic substrate. The film is treated to have the required resistance and then coated with glass or epoxy for moisture resistance and to provide vibration resistance.

Thin film RTDs have the advantages that they provide a fast thermal response, are less sensitive to vibration, and they cost less than their wire wound counterparts. Thin film RTDs can also provide a higher resistance for a given size

Thin film RTDs are less stable than the wire wound ones but they are becoming very popular as a result of their considerably lower costs.

3.1.2 RTD temperature resistance relationship

The resistance of metal is defined as;

$$R_T = R_0 \; [1 + \alpha\,(T - T_o)] \qquad (3.2.a)$$

Where

R_T is the resistance of the metal at temperature T ^oC

R_0 is the resistance of the metal at temperature 0 ^oC

α is the temperature coefficient of resistance

T is the temperature (^oC)

T_o is a constant (^oC)

Setting T_0 to 0^oc, equation 3.2.a becomes;

$$R_T = R_0\ [1 + \alpha T] \tag{3.2.b}$$

This equation is a simplified model of the RTD temperature-resistance relationship.

From equation (3.2.b), effect of temperature on length and cross sectional area of the metal can be neglected. By using curve fitting technique, the relationship between resistivity of the platinum and the temperature in the range studied is approximately linear with good agreement and this relation is: [19]

$$\rho = (0.04021340806 * t + 9.671677086)*10^{-8} \tag{3.2.c}$$

In practice, temperature-resistance relationship of the RTDs is approximated by an equation known as the Callendar-Van Dusen equation [20] which gives very accurate results.

$$R_T = R_o\ [1 + a\,T + b\,T^2 + c\,(T - 100)^3] \tag{3.3.a}$$

Where, a, b, and c are constants which depend upon the material.
Above 0^oc, the constant c is equal to zero therefore, Eq.3.3.a can be approximated to;

$$R_T = R_0\ [1 + a\,T + b\,T^2\,] \tag{3.3.b}$$

From this equation the relation between temperature and relative resistance (R_t/R_0) can be drawn. Figure (3.2) shows this relation, from the Figure it can be noticed that, from T=0 to 100 ^{o}c the relationship between R_t/R_0 and T can be considered linear.

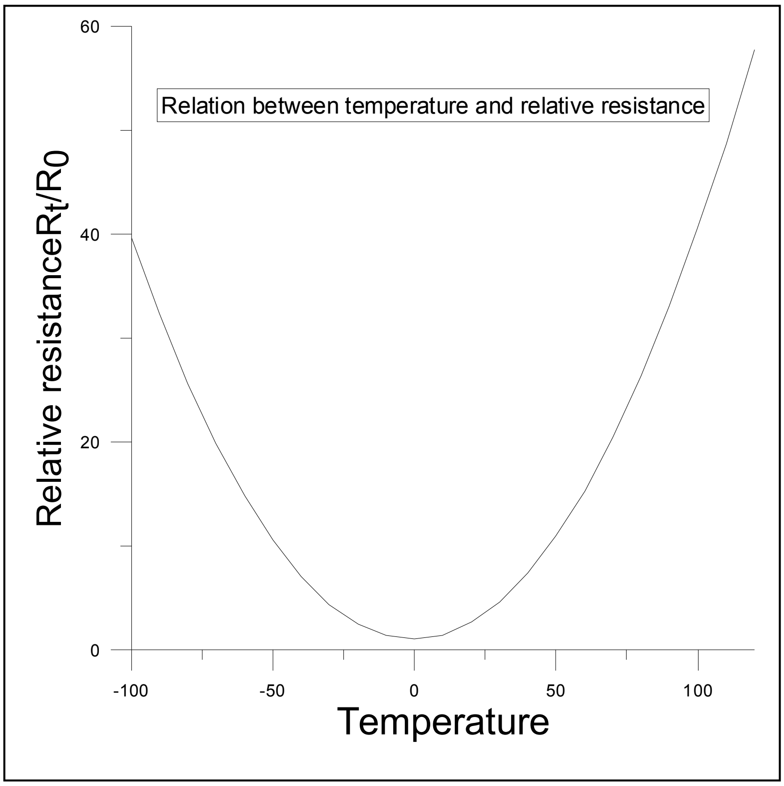


Fig. (3.2) Relation between Temperature and Relative Resistance

In practice it is required to calculate the positive temperature from knowledge of the RTD resistance. From Eq.3.3.b, the value of T is;

$$T = \{-a + [\, a^2 + 4\, b\, (R_T / R_0\ - 1)]^{0.5}\} / 2b \qquad (3.4)$$

3.1.3 RTD Standards

Platinum RTDs conform to either two standard types; [21]

i- The IEC/DIN standard defines pure platinum which is contaminated with other platinum group metals.

ii- The reference-grade platinum is made from 99.99% pure platinum.

The main difference is in the purity of the platinum used.

The most commonly used international RTD standard is the IEC 751 which is based on platinum RTDs with a resistance of 100 Ω at $0^{O}C$ and α parameter of 0.00385.

3.1.4 Practical RTD Circuits

Platinum RTDs are very low resistance devices and they produce very little resistance changes for large temperature changes. (A $1^{o}c$ temperature change will cause a 0.385 Ω change in resistance) so, even a small error in measurement of the resistance can cause a large error in the measurement of the temperature.

A high excitation current however should be avoided since it could give rise to self-heating of the sensor. The resistance of the wires leading to the sensor could give rise to errors when long wires are used. RTDs are usually used in bridge circuits for precision temperature measurement applications. Various practical RTD circuits are given in this section.

Simple current source circuit

Figure (3.3) shows a simple RTD current source circuit where a constant current source I is used to pass a current through the RTD. The voltage across the RTD is measured and then the resistance of the

RTD is calculated. The temperature can then be found by using Eq.3.4.

This circuit has the disadvantage that the resistance of the wires could add to the measured resistance and hence it could cause errors in the measurement.

Care should also be taken not to pass a large current through the RTD since this can cause self heating of the RTD and hence change of its resistance.

If the sensor element is unable to dissipate this heat, it will cause the resistance of the element to increase and hence an artificially high temperature will be reported.

Simple voltage source circuit

Figure (3.4) shows how an excitation current can be passed through an RTD by using a constant voltage source. This circuit suffers from the same problems as the one in Figure (3.3) the voltage across the RTD element is:

$$V_T = V_S \; R_T / (R_S + R_T) \tag{3.5.a}$$

And the resistance of the RTD element can be calculated as:

$$R_T = R_S \; V_T / (V_S - V_T) \tag{3.5.b}$$

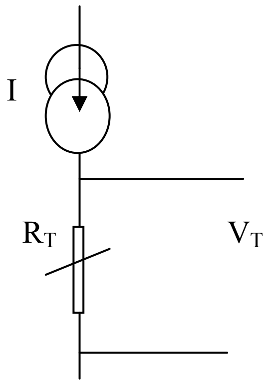


Fig. (3.3) Simple Current Source RTD circuit

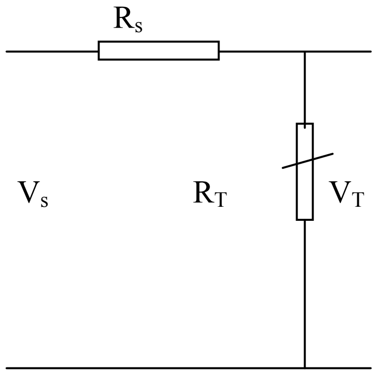


Fig. (3.4) Simple Voltage Source RTD circuit

Four-wire RTD measurement

It is used to compensate for the resistance of the lead wires (when lead wires greater than about 5 m long),

As shown in Figure (3.5), in the four-wire measurement method, one pair of wires carry the current through the RTD, the other pair senses the voltage across the RTD. Resistances R_{L1} and R_{L4} are the lead wires carrying the current and resistances R_{L2} and R_{L3} are the lead wires for measuring the voltage across the RTD. Since only negligible current flows through the sensing wires (voltage measurement device having a very high internal resistance), the lead resistances R_{L2} and R_{L3} can be neglected. Four-wire RTD measurement gives very accurate results and is the preferred method for accurate, precision RTD temperature measurement applications.

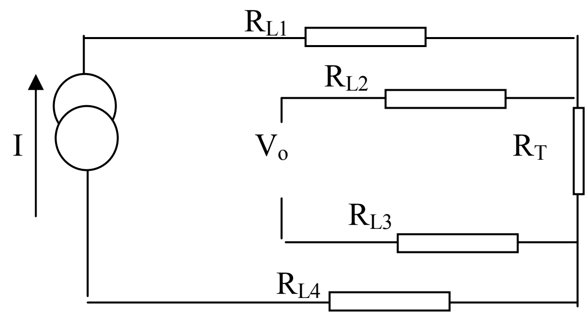


Fig. (3.5) Four-wire RTD Circuit

Simple RTD bridge circuit

As shown in Figure (3.6), a simple Wheatstone bridge circuit can be used with the RTD at one of the legs of the bridge.

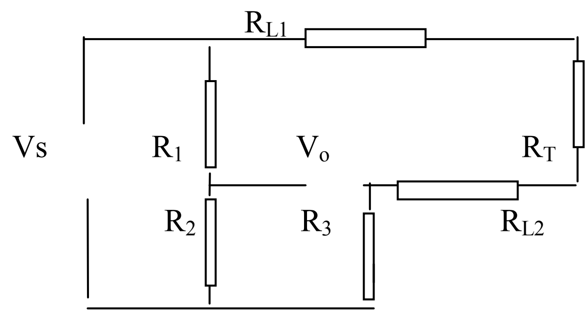
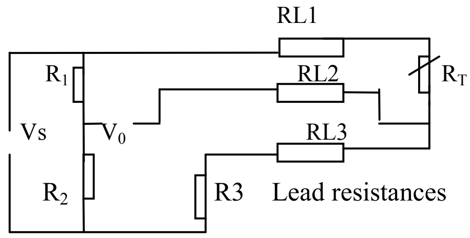


Fig. (3.6) Simple RTD Bridge Circuit

As the temperature changes so does the resistance of the RTD and the output voltage of the bridge. This circuit has the disadvantage that the lead resistances R_{L1} and R_{L2} add to the resistance of the RTD, giving an error in the measurement.

Three-wire RTD bridge circuit

As shown in Figure (3.7), R_{L1} and R_{L3} carry the bridge current. When the bridge is balanced, no current flows through R_{L2} and thus the lead resistance R_{L2} does not introduce any errors into the measurement. The effects of R_{L1} and R_{L3} at different legs of the bridge cancel out since they have the same lengths and are made up of the same material.



Circuit equations:

$R_1+RL1+R_T+RL2= R_2+R3+RL3+RL2$, $R_1= R_2$

$R_T= R3$

Fig. (3.7) Three-Wire RTD Bridge Circuit.

3.2 Mid-Range PIC Architecture

The mid-range PICs [21] that have achieved greater success and popularity. In addition, as the PIC architecture increases in complexity and power, so does the size, intricacy, and cost of the devices.PIC microcontrollers are used in many applications such as:[22] for online monitoring and fiber fault identification in optical fiber communication. For this propose the PIC microcontroller is used control any optical switch connected to it

For many purposes an 80-pin PIC with 64Kbytes of program memory, 1K ERPOM, 70 I/O ports, 16 A/D channels, is more complex than necessary.

In fact, some high-end PICs appear to be closer to micro-processors than to microcontrollers. Furthermore, the programming complexity of these high-end PICs is also much greater than their mid-range counterparts because their instruction set has double the number of instructions and the assembly language itself is more difficult to learn and follow.

Finally, the circuits in which we typically find the high-end devices are more advanced and elaborate and their design requires greater engineering skills.

For these reasons, and for the natural space limitations of a single volume, we do not discuss the high-performance family or 8-bit PICs or any of the 16-bit products.

It can be argued that the baseline PICs do find extensive use and are quite practical for many applications. Although this is true, the baseline PICs are quite similar in architecture and programming to their mid-range relatives.

In most cases the difference between a baseline and mid-range device is that the low-end one lacks some features or has less program space or storage.

So someone familiar with the mid-range devices can easily port their knowledge to any of the simpler baseline products.

This conclusion has been to limit the coverage to the mid-range family of PICs. Within this family it has been concentrated on the two most used, documented, and popular PICs: the 16F84 (also 16F84A) and the 16F877. The F84 sets the lower limit of complexity and sophistications and the F877 the higher limit.

Figure (3.8) shows the internal structure of PIC 16F877, as seen in the figure, it has central processing unit (CPU), memory, input/ output ports (I/O), timers and counters, analog- to-digital converter (ADC), digital-to-analog converter (DAC), serial ports, interrupt logic, oscillator circuitry and many more functional blocks on chip.

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes

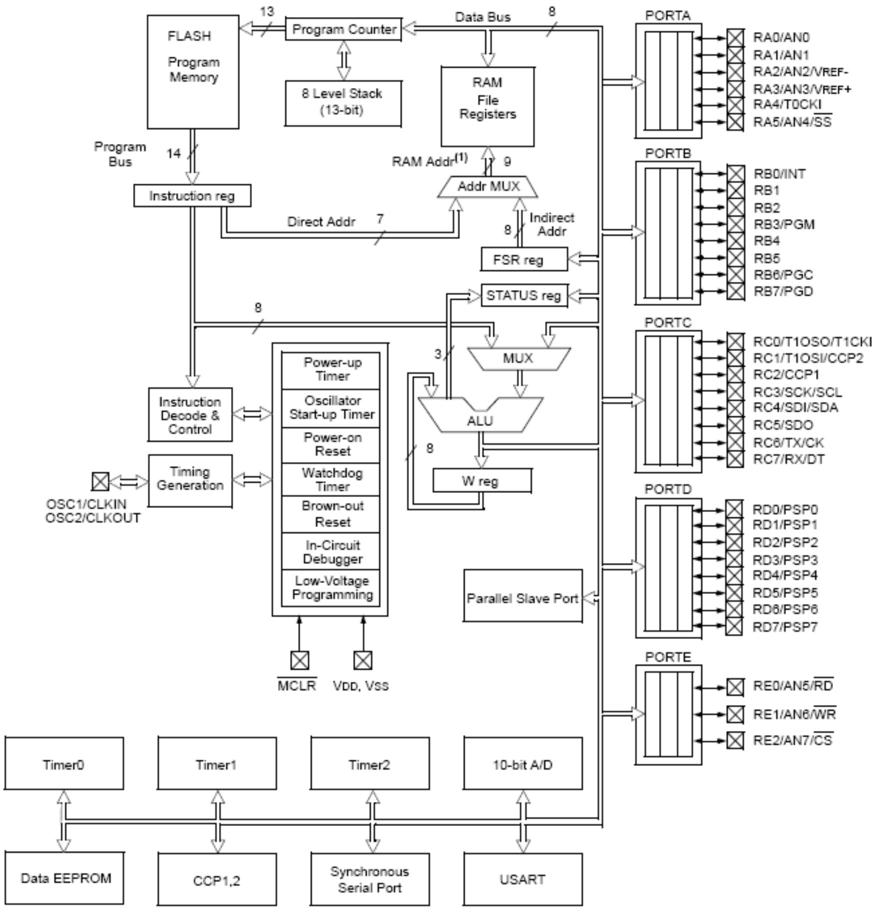


Fig. (3.8) PIC16F874 and PIC16F877 Block Diagram

3.2.1 Processor architecture and design

PIC [23] microcontrollers are unique in many ways. We start by mentioning several general characteristics of the PIC: Harvard architecture, RISC processor design, single-word instructions, machine, data memory configuration, and characteristic instruction formats.

Harvard Architecture

The PIC microcontrollers do not use the conventional von Neumann architecture but a different hardware design often referred to as Harvard architecture.

Originally, Harvard architecture referred to a computer design in which data and instruction used different signal paths and storage areas. In other words, data and instructions are not located in the same memory area but in separate ones.

One consequence of the traditional von Neumann architecture is that the processor can either read or write instructions or data but cannot do both at the same time, since both instructions and data use the same signal lines.

In a machine with Harvard architecture, on the other hand, the processor can read and write instructions and data to and from memory at the same time.

This results in a faster, albeit more complex, machine. Figure (3.9) shows the program and data memory space in a mid-range PIC.

The most recent arguments in favor of the Harvard architecture are based on the access speed to main memory. Making a CPU faster while memory accesses remain at the same speed represents little total gain, especially if many memory accesses are required.

This situation is often referred to as the von Neumann bottleneck and machines that suffer from it are said to be memory bound.

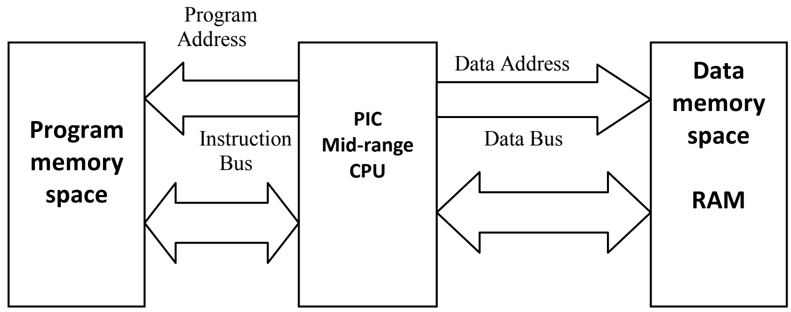


Fig. (3.9) Mid-range PIC Memory (Harvard Architecture).

Several generations of microcontrollers, including the Microchip PICs, have been based on the Harvard architecture. These processors have separate storage for program and data and a reduced instruction set. The midranges PICs, in particular, have 8-bit data words but either 12-, 14-, or 16-bit program instructions. Since the instruction size is much wider than the data size, an instruction can contain a full-size data constant.

Single-word Instructions

One of the consequences of the PIC's Harvard architecture is that the instructions can be wider than the 8-bit data size. Since the device has separate buses for instructions and data, it is possible for instructions to be sized differently than data items.

Being able to vary the number of bits in each instruction opcode makes possible the optimization of program memory and the use of single-word instructions that can be fetched in one bus cycle.

In the mid-range PICs each instruction is 14-bits wide and every fetch operation brings into the execution unit one complete operation code. Since each instruction takes up one 14-bit word, the number of words of program memory in a device exactly equals the number of program instructions that can be stored. In a von Neumann machine, instruction storage and fetching becomes a much more complicated issue. Since von Neumann instructions can span multiple bytes, there is no assurance that each program memory location contains the first opcode of a multi-byte instruction. As in conventional processors, the PIC architecture has a two-stage instruction pipeline; however, since the fetch of the current instruction and the execution of the previous one can overlap in time, one complete instruction is fetched and executed at every machine cycle. This is known as instruction pipelining.

Since each instruction is 14-bits wide and the program memory bus is also 14-bits wide, each instruction contains all the necessary information, so it can be executed without any additional fetching.

The one exception is when an instruction modifies the contents of the program counter. In this case, a new instruction must be fetched, requiring an additional machine cycle. The PIC clocking system is designed so that an instruction is fetched, decoded, and executed every four clock cycles. In this manner, a PIC equipped with a 4MHz oscillator clock beats at a rate of 0.25µs. Since each instruction executes at every four clock cycles, each instruction takes 1µs.

Instruction Format

All members of the mid-range family of PICs have 14-bit instructions [24] and a set of 35 instructions.

The format for the instructions follows three different patterns: byte-oriented, bit-oriented, and literal and control instructions. The opcode field has variable number of bits in the PIC instruction set.

This scheme allows implementing 35 different instructions while using a minimum of the 14 available opcode bits.

Also note that instructions that reference a file register do so in a 7-bit field. The numerical range of seven bits is 128 values. For this reason, the mid-range PICs that address more than 128 data memory locations must resort to banking techniques.

In this case, a bit or bit field in the STATUS register serves to select the bank currently addressed. A similar situation arises when addressing program memory with an 11-bit field. Eleven bits allow 2048 addresses, so if a PIC is to have more than 2K program memory it is necessary to adopt a paging scheme in which a special function register is used to select the memory page where the instruction is located. Paging is required only in devices that exceed the 2K program space limit that can be encoded in 11 bits.

Mid-range device versions

The device names used by Microchip use different encodings to represent different versions of the various devices. For example, the first letter following the family affiliation designator represents the memory type of the device, as follows:

1. The letter C, as in PIC16Cxxx, refers to devices with EPROM type memory.
2. The letters CR, as in PIC16CRxxx, refer to devices with ROM type memory.
3. The letter F, as in PIC16Fxxx, refers to devices with flash memory.

The letter L immediately following the affiliation designator refers to devices with an extended voltage range. For example, the PIC16LFxxx designation corresponds to devices with extended voltage range.

3.2.2 Mid-Range CPU and instruction set

In a digital system, the central processing unit (CPU) is the component that executes the program instructions and processes data. It provides the fundamental functionality of a digital system and is responsible for its programmability. In the PIC architecture, the CPU is the part of the device [25] which fetches and executes the instructions contained in a program.

The arithmetic-logic unit (ALU) is the CPU element that performs arithmetic, bitwise, and logical operations. It also controls the bits in the STATUS register as they are changed by the execution of the various program instructions. For example, if the result of executing an instruction is zero, the ALU sets the zero bit in the STATUS register.

Mid-Range Instruction Set

The mid-range PIC instruction set consists of 35 instructions [26], divided into three general groups:

1. Byte-oriented and byte-wise file register operations
2. Bit-oriented and bit-wise file register operations
3. Literal and control instructions

STATUS and OPTION Registers

The STATUS register is one of the SFRs in the mid-range PICs. The bits in this register reflect the arithmetic status of the ALU, the RESET status, and the bits that select which memory bank is currently being accessed. Because the bank selection bits are in the STATUS register it must be present and at the same relative position in every bank. Table 3.2 is a bitmap of the STATUS register.

Table 3.2 Status Register

Bits	7	6	5	4	3	2	1	0
	IRP	RP-1	RP-0	T0	PD	Z	DC	C

Bit 7 IRP:

Register Bank Select bit (used for indirect addressing)

1 = Bank 2, 3 (0x100 - 0x1ff)

0 = Bank 0, 1 (0x000 - 0xff)

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

Bit 6:5 RP-1, RP-0:

Register Bank Select bits (used for direct addressing)

11 = Bank 3 (0x180 - 0x1ff)

10 = Bank 2 (0x100 - 0xx17f)

01 = Bank 1 (0x80 - 0xff)

00 = Bank 0 (0x00 - 0x7f)

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

Bit 2 Z:

Zero bit

1 = The result of an operation is zero

0 = The result of an operation is not zero

Bit 1 DC:

Digit carry/borrow bit for ADDWF, ADDLW, SUBLW, and SUBWF instructions. For borrow the polarity is reversed.

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

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

Bit 0 C:

Carry/borrow bit for ADDWF, ADDLW, SUBLW, and SUBWF instructions

1 = A carry-out from the most significant bit

0 = No carry-out from the most significant bit

The STATUS register can be the destination for any instruction. If it is, and the Z, DC, or C bits are affected, then the write operation to these bits is disabled. In addition, the TO and PD bits are not writable. Some instructions may have an unexpected action on the STATUS register bits, for example, the instruction Clrf STATUS clears the upper 3 bits, sets the Z bit, and leaves all other bits unchanged.

For this reason, it is recommended that only instructions that do not change the Z, C, and DC bits be used to alter the STATUS register. The only ones that qualify are BCF, BSF, SWAPF, and MOVWF.

The OPTION register:

The OPTION register is actually named the OPTION_REG to avoid name clash with the option instruction. The OPTION_REG register contains several bits related to interrupts, the internal timers, and the watchdog timer. Table 3.3 is a bitmap of the OPTION_REG register.

Table 3.3 Option Register

Bits	7	6	5	4	3	2	1	0
	RPBU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0

Bit 7 RPBU:

PORTB Pull-up Enable bit

1 = PORTB pull-ups are disabled

0 = PORTB 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 TOCS:

TMR0 Clock Source Select bit

1 = Transition on T0CKI pin

0 = Internal instruction cycle clock (CLKOUT)

Bit 4 TOSE:

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

Bit 2-0 PS2:PS0:

Prescaler Rate Select bits

3.2.3 EEPROM Data Storage

EEPROM (pronounced double-e PROM or e-squared PROM) stands for electrically- erasable programmable read-only memory. EEPROM is used in computers and digital devices as non-volatile storage. EEPROM is not RAM, since RAM is volatile and EEPROM retains its data after power is removed. EEPROM is found in USB flash drives and in the non-volatile storage of several microcontrollers, including many PICs.

One advantage of EEPROM is that it can be erased and written electrically [27], without removing the chip. The predecessor technology, named EPROM, required that the chip be removed from the circuit and placed under ultraviolet light. EEPROM simplifies the erasing and re-writing process. EEPROM data memory refers to both on-board EEPROM memory and to EEPROM memory ICs as separate circuit components. In general, EEPROM elements are classified according to their electrical interfaces into serial and parallel.

Most EEPROM memories used in PICs are serial EEPROMs, also called SEEPROMs. The typical use of serial EEPROM on-board memory and EEPROM on ICs is in the storage of passwords, codes, configuration settings, and other information to be remembered after the system is turned off. For example, a PIC-based security system can use EEPROM memory to store the system password. Since EEPROM can be written, the user can change this password and the new one will also be remembered.

EEPROM in Mid-Range PICs

The mid-range PICs are equipped with EEPROM memory in three possible sizes: 64 bytes, 128 bytes, and 256 bytes. EEPROM memory allows read and write operations. This memory is not mapped into the processor's data or program area, but in a separate block that is addressed through some SFRs. The registers related to EEPROM operations are:

1. EECON1
2. EECON2 (not a physically implemented register)
3. EEDATA
4. EEADR

EECON1 contains the control bits, and EECON2 is used to initiate the EEPROM read and write operations. The 8-bit data item to be written must first be stored in the EEDATA register, while the address of the location in EEPROM memory is stored in the EEADR register. The EEPROM address space always starts at 0x00 and extends linearly to maximum in the device. When a write operation is performed, the contents of the EEPROM location are automatically erased. The EEPROM memory used in PICs is rated for high erase/write cycles.

3.2.4 Data memory organization

The structure and organization of data memory in the PIC hardware [28] also has some unique and interesting features. The programmer accustomed to the flat, addressable memory space of the von Neumann computer with its multiple machine registers may require some time in order to gain familiarity with the PIC's data formats.

The w register

PICs have only one addressable register called the work register or the w register. The CISC programmer who is used to having multiple general purpose registers into which data can be moved and later retrieved has to become used to a single machine register that takes part in practically every instruction. Add to this the lack of an addressable stack into which data can be pushed and popped, and you see that PIC programming is a different paradigm.

The data registers

PIC's data memory consists of registers, also called file registers. These behave more like conventional variables, and can be addressed directly and indirectly. All data registers are 8-bits. Data registers come in two types: general purpose registers (GPRs) and special function registers (SFRs).

The SFRs: The special function registers [29] are defined by the device architecture and have reserved names. For example, the TMR0 register is part of the system timer, the STATUS register holds several processor flags, and the INTCON register is used in controlling interrupts. Some SFRs can be written and read and others are read-only. Some reserved and not-implemented SFR bits always read as zero. Two SFR registers, which are used in indirect addressing, have special characteristics: one of them (the indirect address register) is not a physical register, and the other one (the FSR register) is used to initialize the indirect pointer. The SFR are allocated starting at the lowest RAM address (address 0). The general purpose registers do not start at the same address offset in each bank. However, there is a common area that extends from 0x70 to 0x7f that is accessible no matter which bank is selected. In applications that require frequent bank switching, this 16-byte area is very valuable real-estate since user variables created in it are accessible no matter which bank is currently selected. Bank is selected. GPRs created outside this common area are only accessible when the corresponding bank is selected.

Figure (3.10) is a map of the register file in the 16F87x family. Note in this Figure that the general purpose registers do not start at the same address offset in each bank. The registers in boldface in this Figure are accessible from any bank. These registers, such as STATUS and the indirect addressing registers FSR and INDF are bank-independent. Also, some registers are mirrored in more than one bank. For example, the PORTB register is accessible in bank 0 and in bank 2, and the TRISB register in bank 1 and bank 3. The mirrored registers are designed to simplify data access and minimize bank changes in applications.

Bank 0		Bank 1		Bank 2		Bank 3	
INDF	0x00	**INDF**	0x80	**INDF**	0x100	**INDF**	0x180
TMR0	0x01	OPTION*	0x81	TMR0	0x101	OPTION*	0x181
PCL	0x02	**PCL**	0x82	**PCL**	0x102	**PCL**	0x182
STATUS	0x03	**STATUS**	0x83	**STATUS**	0x103	**STATUS**	0x183
FSR	0x04	**FSR**	0x84	**FSR**	0x104	**FSR**	0x184
PORTA	0x05	TRISA	0x85		0x105		0x185
PORTB	0x06	TRISB	0x86	PORTB	0x106	TRISB	0x186
PORTC	0x07	TRISC	0x87		0x107		0x187
PORTD	0x08	TRISD	0x88		0x108		0x188
PORTE	0x09	TRISE	0x89		0x109		0x189
PCLATH	0x0a	**PCLATH**	0x8a	**PCLATH**	0x10a	**PCLATH**	0x18a
INTCON	0x0b	**INTCON**	0x8b	**INTCON**	0X10b	**INTCON**	0x18b
PIR1	0x0c	PIE1	0x8c	EEDATA	0x10c	EECON1	0x18c
PIR2	0x0d	PIE2	0x8d	EEADR	0x10d	EECON2	0x18d
TMR1L	0x0e	PCON	0x8e	EEDATH	0x10e	Reserved	0x18e
TMR1H	0x0f		0x8f	EEADRH	0x10f	Reserved	0x18f
T1CON	0x10		0x90	General Purpose Registers	0x110	General Purpose Registers	0x190
TMR2	0x11	SSPCON2	0x91				
T2CON	0x12	PR2	0x92				
SSPBUF	0x13	SSPADD	0x93				
SSPCON	0x14	SSPTAT	0x94				
CCPR1L	0x15		0x95				
CCPR1H	0x16		0x96				
CCP1CON	0x17		0x97				
RCSTA	0x18	TXSTA	0x98				
TXREG	0x19	SPBRG	0x99				
RCREG	0x1a		0x9a				
CCPR2L	0x1b		0x9b				
CCPR2H	0x1c		0x9c				
CCP2CON	0x1d		0x9d				
ADRESH	0x1e	ADRESL	0x9e				
ADCON0	0x1f	ADCON1	0x9f				
General Purpose Registers	0x20	General Purpose Registers	0xA0				
			0xef		0x16f		0x1ef
Common area 0x70-0x7f		Common area 0x70-0x7f	0xf0	Common area 0x70-0x7f	0x170	Common area 0x70-0x7f	0x1f0
	0x7f		0xff		0x17f		0x1ff

Fig. (3.10) 16F87x File Register Map

The GPRs: General purpose registers are created and named by the programmer and must be allocated in the reserved memory space. In the 16F84A all GPRs are mapped to the same memory area, no matter in which bank they are defined. The GPR memory space actually extends from 0x0c to 0x4f (68 bytes). A different situation exists in the 16F87x PICs, in which only 16 bytes of GPR space is mirrored in all three banks. This is the memory referred to as the common area. In the 16F87x the total available GPR space is as follows:

BANK 0 (96 bytes), BANK 1 (80 bytes), BANK2 (96 bytes) and BANK3 (96 bytes) so, Total = 368 bytes

3.2.5 Mid-range I/O and Peripheral Modules

Mid-range devices contain special modules to implement peripheral and I/O functions [30]. The more complex the device the more peripheral modules are likely to be present. For example, a simple mid-range PIC like the 16F84A contains few peripheral modules specifically, EEPROM data memory, I/O ports, and a timer module. The 16F87x PICs, on the other hand, in addition to I/O ports, EEPROM, and three individual timers, have a parallel slave port, a WPM (capture and compare) module, an MSSP (master synchronous serial port) module, a USART (universal asynchronous/synchronous receiver and transmitter) module, and an A/D (analog-to-digital converter) module. Other members of the mid-range family have additional or different peripheral and I/O modules. In the following sections, we briefly describe the architecture of the most common peripheral modules.

I/O Ports

Ports provide PICs access to the outside world and are mapped to physical pins on the device. In some mid-range PICs some port pins for I/O ports are multiplexed with alternate functions of peripheral modules. When a peripheral module is enabled, that pin ceases to be a general purpose I/O.

Port pins can be configured either as input or output, that is, general ports are bidirectional. Each port has a corresponding TRIS register which determines if a port is designated as input or output. A value of 1 in the port's TRIS register makes the port an input and a value of 0 makes the mapped port an output. Typically, input ports are used in communicating with input devices, such as switches, keypads, and input data lines from hardware devices. Output ports are used in communicating with output devices, such as LEDs, seven-segment displays, LCDs (liquid-crystal displays), and data output lines to hardware devices. Although port pins are bitmapped, they are read and written as a unit. For example, the PORTA register holds the status of the eight pins possibly mapped to Port-A, while writing to PORTA writes to the port latches. Write operations to ports are actually read-modify-write operations. In other words, the port pins are first read, then the value is modified, and then written to the port's data latch. Some of the port pins are multiplexed; for example, pin RA4 is multiplexed with the Timer0 module clock input; therefore, it is labeled RA4/T0CKI pin. Other PORTA pins are multiplexed with analog inputs and with other peripheral functions. The device data sheets contain information about the functions assigned to each device pin.

3.3 Analog to Digital and Real-time Clocks

Digits are a human invention; nature does not count or measure using numbers. Natural forces and phenomena are measured by using digital representations [31], but the forces and phenomena themselves are continuous. Time, pressure, voltage, current, temperature, humidity, gravitational attraction, all exist as continuous entities which we measure volts, pounds, hours, amperes, or degrees, so as to better understand them and to be able to perform numerical calculations. In this sense, natural phenomena occur in analog quantities. Sometimes they are digitized so as to facilitate measurements and manipulations.

For example, a potentiometer in an electrical circuit allows reducing the voltage level from the circuit maximum to ground, or zero level. In order to measure and control the action of the potentiometer, we need to quantify its action by producing a digital value within the physical range of the circuit; that is, we need to convert an analog quantity that varies continuously between 0 and 5 volts, to a discrete digital value range. If, in this case, the voltage range of the potentiometer is from 5 to 0 volts, we can digitize its action into a numeric range of 0 to 500 units, or measure the angle or rotation of the potentiometer disk in degrees from 0 to 180.

The device that performs either conversion is called an A/D or analog- to-digital converter. The reverse process, digital-to-analog, is also necessary, although not as often as A/D. In this chapter we explore A/D conversions in PIC software and hardware.

3.3.1 A/D Converters

In electronics, the typical A/D or ADC converter is a device that takes a voltage input and returns a binary digital number. Figure (3.11) is a block diagram of an A/D converter. The electronic A/D converter requires an input in the form of an electrical voltage [32]. Non-electric quantities must be changed into a voltage level before the conversion can be performed. The device that performs this conversion is called a transducer.

For example, a digital barometer must be equipped with a transducer that converts the measurement into voltage levels. The voltage levels can then be fed into an A/D converter and the result output in digital form.

Converter Resolution

An ideal A/D converter outputs into an infinite number of discrete steps that exactly represent the analog quantity. Needless to say, such a device cannot exist, and a real A/D. converter must be limited to a numeric range .For example, the device in Figure (3.11) outputs a voltage range of 0 to +5 volts in four binary digits that represent values between 0 and 15. Another A/D converter may produce output in eight binary digits, and another in sixteen binary digits. The number of discrete values in the conversion is called the resolution. The converter's resolution is usually expressed in bits. Figure (3.12) represents an A/D converter with a voltage range of 0 to

+5 volts and a resolution of three bits, the number of intervals is odd, so the quantizer is mid-tread.

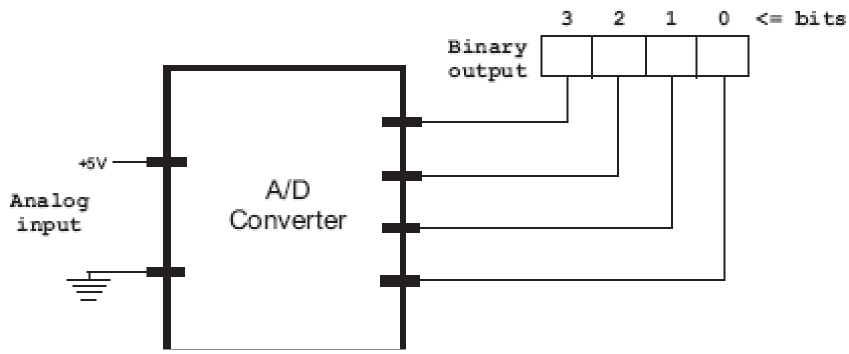


Fig. (3.11) A/D Converter Block Diagram

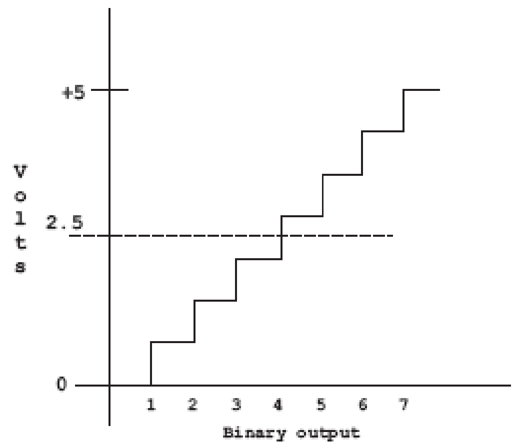


Fig. (3.12) Converter Quantization Error.

Suppose that a value of 2.5 volts were input into the A/D converter in Figure (3.12). Since the output has a resolution in the range 0 to 7, the converter's output would be either 4 or 5. The non-linear characteristic of the output determines a quantization error that increases as the converter resolution decreases. Converters used in PIC circuits have a resolution of either 8, 10, or 12 bits.

In each case the output range, or quantization level, is 0 to 255, 0 to 1023, or 0 to 4095. The voltage resolution of the converter is its maximum voltage range divided by the number of quantization levels. In our case the device with a voltage range of 5 volts and a range of 1023 levels has a voltage resolution of

Voltage resolution = 5/1023 = 4.88 mV.

3.3.2 ADC Configuration

The analog-to-digital converter performs accurately only if the input voltage is within the converter's valid range. This range is usually selected by setting high and low voltage references on converter pins [33]. For example, if +4 volts is input into the converter's positive reference pin and +2 volts into the negative reference pin, then the converter's voltage range lies between these values.

In many PIC applications the converter range is selected as the system's supply voltage and ground, that is, +5 and 0 volts. When a different range is externally referenced, there is a general restriction that the range cannot exceed the system's positive and negative limits (V_{dd} and V_{ss}). Also, a minimum difference is required between the high and low voltage references.

The output of the ADC is a digital representation of the original analog signal. In this context, the term quantization refers to subdividing a range into small but measurable increments. The quantization process can introduce a quantization error, which is similar to a rounding error.

The time required for the holding capacitor on the ADC to charge is called the acquisition time. The holding capacitor on the ADC must be given sufficient time to settle to the analog input voltage level before the actual conversion is initiated. Otherwise, the conversion is not accurate.

The acquisition time is determined by the impedance of the internal multiplexer and that of the analog source. The exact acquisition time can be determined from the device's data sheet, although 10K ohms are the maximum recommended source impedance for 8- and 10-bit converters and 2.5K ohms for 12-bit converters. Most analog-to-digital converters in PIC applications, either internal or external, are of the successive approximation type.

The successive approximation algorithm performs a conversion on one bit at a time, beginning with the most significant bit and ending with the least significant bit. To determine each bit in the range, the value of the input signal is tested to see if it is in the upper or lower portion of this range. If in the upper portion, the conversion bit is a 1, otherwise it is a 0. The next most significant bit is then tested in the lower half of the remaining range.

The process is continued until the least-significant bit has been determined.

3.3.3 PIC On-Board A/D Hardware

A few years ago, A/D conversions always required the use of devices such as ADC0831, the LTC1298, and the MAX 190 and MAX 191. Nowadays, many PIC microcontrollers come with onboard A/D hardware. One of the advantages of using onboard A/D converters is saving interface lines. In the PIC world, where I/O lines are often in short supply, this advantage is not insignificant. At the time we are writing, PICs equipped with A/D converters have either 8- or 10-bit resolution and can receive analog input in 2 to 16 different channels. The 16F877 with eight analog input channels at a 10-bit resolution is discussed. Nowadays, these PICs are easy to obtain. On the other hand, if the resolution required exceeds 10-bits then the designer has to resort to an independent A/D IC, such as the LTC1298, which has a 12-bit resolution, or to others with even higher numbers of output bits.

A/D Module on the 16F87x

The PICs of the 16F87x family are equipped with an analog-to-digital converter module. The number of lines depends on the specific version of the device: 28-pin devices have five A/D lines and all others have eight lines. The converters use a sample and hold capacitor to store the analog charge and perform a successive approximation algorithm to produce the digital result.

The converter resolution is 10 bits, which are stored in two 8-bit registers. One of the registers has only four significant bits.

The A/D module has high- and low-voltage reference inputs that are selected by software. The module can operate while the processor is in SLEEP mode, but only if the A/D clock pulse is derived from its internal RC oscillator.

The module contains four registers accessible to the application:

• ADRESH - Result High Register

• ADRESL - Result Low Register

• ADCON0 - Control Register 0

• ADCON1 - Control Register 1

Of these, it is the ADCON0 register that controls most of the operations of the A/D module. Port-A pins RA0 to RA5 and Port-E pins RE0 to RE2 are multiplexed as analog input pins into the A/C module. In the 28-pin versions of the 16F87x, port pins RA0 to RA5 provide the five input channels. In all other implementations of the 16F87X, Port-E pins RE0 to RE2 provide the three additional channels.

Figure (3.13) shows the registers associated with A/D module operations.

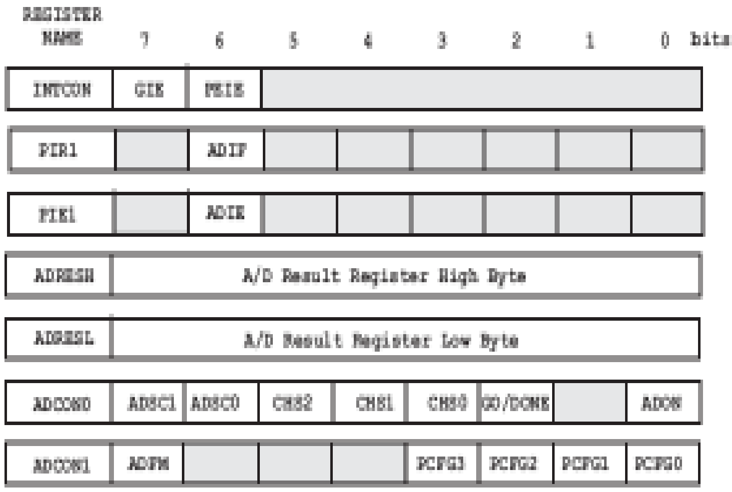


Fig. (3.13) Registers Related to A/D Module Operations

The ADCON0 Register

The ADCON0 register is located in bank 0, at address 0x1f. Seven of the eight bits are meaningful in A/D control and status operations. Figure (3.14) is a bitmap of the ADCON0 register. In this figure, bits 7 and 6, labeled ADSC1 and ADSC0, are the selection bits for the A/D conversion clock. The conversion time per bit is defined as TAD in PIC documentation. A/D conversion requires a minimum of 12 TAD in a 10-bit ADC. The source of the A/D conversion clock is software selected. The four possible options for TAD are:

1. Fosc/2
2. Fosc/8
3. Fosc/32
4. Internal A/D module RC oscillator (varies between 2 and 6 µs)

bits:	7	6	5	4	3	2	1	0
	ADSC1	ADSC0	CHS2	CHS1	CHS0	GO/DONE		ADON

Fig. (3.14) ADCON0 Register Bitmap

Bit 7-6 **ADCS1:ADCS0:** A/D Conversion Clock Select bits
- 00 = FOSC/2
- 01 = FOSC/8
- 10 = FOSC/32
- 11 = FRC (internal A/D module RC oscillator)

Bit 5-3 **CHS2:CHS0:** Analog Channel Select bits
- 000 = channel 0, (RA0=AN0)
- 001 = channel 1, (RA1=AN1)
- 010 = channel 2, (RA2=AN2)
- 011 = channel 3, (RA3=AN3)
- 100 = channel 4, (RA5=AN4)
- 101 = channel 5, (RE0=AN5) | not active
- 110 = channel 6, (RE1=AN6) | in 28-pin
- 111 = channel 7, (RE2=AN7) | 16F87x PICS

Bit 2 **GO/DONE:**
- A/D On bit
- 1 = A/D converter module is operating
- 0 = A/D converter module is shut-off and consumes no power

Bit 1 Unimplemented: Read as '0'

Bit 0 ADON**:** A/D Conversion Status bit If ADON = 1:
- 1 = A/D conversion in progress (setting this bit starts the A/D conversion)
- 0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)

The conversion time is the analog-to-digital clock period multiplied by the number of bits of resolution in the converter, plus the two to three additional clock periods for settling time, as specified in the data sheet of the specific device. The various sources for the analog-to-digital converter clock represent the main oscillator frequency divided by 2, 8, or 32. The third choice is the use of a dedicated internal RC clock that has a typical period of 2 to 6 μs. Since the conversion time is determined by the system clock, a faster clock results in a faster conversion time.

The A/D conversion clock must be selected to ensure a minimum Tad time of 1.6 μs. The formula for converting processor speed (in MHz) into Tad microseconds is as follows:

Tad=1/ Tosc/Tdiv

Where

Tad is A/D conversion time,

Tosc is the oscillator clock frequency in MHz,

Tdiv is the divisor determined by bits ADSC1 and ADSC0 of the ADCON0 register.

For example, in a PIC running at 5MHz if we select the Tosc/8 option (divisor equal 8) the A/D conversion time per bit is calculated as follows:

Tad= 1/5/8=1.6 μs

In this case, the minimum recommended conversion speed of 1.6 μs is achieved. However, in a PIC with an oscillator speed of 10MHz, this option produces a conversion speed of 0.8 μs, less than the recommended minimum. In this case we would have to select the divisor 32 option, giving a conversion speed of 3.2 μs.

In Table 3.4, converter speeds of less than 1.6 μs or higher than 10 μs are not recommended. Recall that the Tad speed of the converter is calculated per bit, so the total conversion time in a 10-bit device (such as the 16F87x) is approximately the Tad speed multiplied by 10 bits, plus 3 additional cycles. Therefore, a device operating at a Tad speed of 1.6 μs requires 1.6 μs * 13, or 20.8 μs for the entire conversion.

Table 3.4 A/D converter tad at various oscillator speeds

A/C Converter Tad at Various Oscillator Speeds

		TAD IN MICROSECONDS			
OPERATION	ADCS1:ADCS0	20MHZ	10MHZ	5MHZ	1.25MHZ
Fosc/2	00	0.1	0.2	0.4	**1.6**
Fosc/8	01	0.4	0.8	**1.6**	**6.4**
Fosc/32	10	**1.6**	**3.2**	**6.4**	25.6
RC	11	**2-6**	**2-6**	**2-6**	**2-6**

Note: values in bold are within the recommended limits

Bits CHS2 to CHS0 in the ADCON0 register (see Figure 3.14) determine which of the analog channels is selected. This is required, since there are several channels for analog input but only one A/2 converter circuitry. So the setting of this bit field determines which of six or eight possible channels is currently read by the A/D converter.

An application can change the setting of these bits in order to read several analog inputs in succession.

Bit 2 of the ADCON0 register, labeled GO/DONE, is both a control and a status bit. Setting the GO/DONE bit starts A/D conversion.

Once conversion has started, the bit indicates if it is still in progress. Code can test the status of the GO/DONE bit in order to determine if conversion has concluded.

Bit 0 of the ADCON0 register turns the A/D module on and off. The initialization routine of an A/D-enabled application turns on this bit. Programs that do not use the A/D conversion module leave the bit off to conserve power.

The ADCON1 Register

The ADCON1 register also plays an important role in programming the A/D module.

Bit 7 of the ADCON1 register is used to determine the bit justification of the digital result. This is possible because the 10-bit result is returned in two 8-bit registers; therefore, the six unused bits can be placed either on the left- or the right-hand side of the 16-bit result.

If ADCON1 bit 7 is set then the result is right-justified; otherwise it is left-justified. Figure (3.15) shows the location of the significant bits.

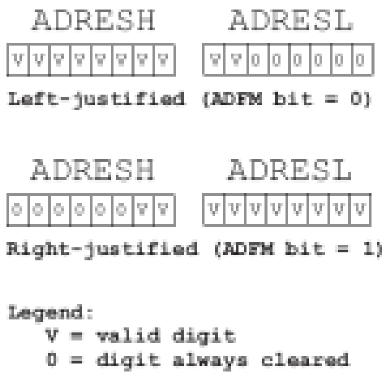


Fig. (3.15) Location of the Significant Bits

One common use of right justification is to reduce the number of significant bits in the conversion result. For example, an application on the 16F877 that uses the A/D conversion module requires only 8-bit accuracy in the result. In this case, code can left-justify the conversion result, read the ADRESH register, and ignore the low-order bits in the ADRESL register. By ignoring the two low-order bits, the 10-bit accuracy of the A/D hardware is reduced to eight bits and the converter performs as an 8-bit accuracy unit.

The bit field labeled PCFG3 to PCFG0 in the ADCON1 register determines port configuration as analog or digital and the mapping of the positive and negative voltage reference pins. The number of possible combinations is limited by the four bits allocated to this field, so the programmer and circuit designer must select the option that is most suited to the application when the ideal one is not available. Table 3.5 shows the port configuration options. For example, there is a circuit that calls for two analog inputs, wired to ports RA0 and RA1, with no reference voltages. In Table 3.6, we can find two options that select ports RA0 and RA1 and are analog inputs: these are the ones selected with PCFG bits 0100 and 0101.

The first option also selects port RA3 as analog input, even though not required in this case. The second one also selects port RA3 as a positive voltage reference, also not required. Either option works in this case; however, any pin configured for analog input produces incorrect results if used as a digital source. Therefore, a channel configured for analog input cannot be used for non-analog purposes. On the other hand, a channel configured for digital input should not be used for analog data since extra current is consumed by the hardware. Finally, channels to be used for analog-to-digital conversion must be configured for input in the corresponding TRIS register.

SLEEP Mode Operation

The A/D module can be made to operate in SLEEP mode. As mentioned previously, SLEEP mode operation requires that the A/D clock source be set to RC by setting both ADCS bits in the ADCON0 register. When the RC clock source is selected, the A/D module waits one instruction cycle before starting the conversion. During this period, the SLEEP instruction is executed, thus eliminating all digital switching noise from the conversion. The completion of the conversion is detected by testing the GO/DONE bit. If a different clock source is selected, then a SLEEP instruction causes the conversion-in-progress to be aborted and the A/D module to be turned off.

Table 3.5 A/D converter port configuration options

A/D Converter Port Configuration Options

PCFG3:	An7	An6	An5	An4	An3	An2	An1	An0			CHAN/
PCFG0	Re2	Re1	Re0	Ra5	Ra3	Ra2	Ra1	Ra0	Vref+	Vref-	Refs
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	Vre+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	Vre+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	Vre+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	Vre+	Vre-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	Vre+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	Vre+	Vre-	A	A	RA3	RA2	4/2
1100	D	D	D	A	Vre+	Vre-	A	A	RA3	RA2	3/2
1101	D	D	D	D	Vre+	Vre-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	Vre+	Vre-	D	A	RA3	RA2	1/2

Legend:

D = digital input

A = analog input

CHAN/Refs = analog channels/voltage reference inputs

Finally Programming the A/D module consists of the following steps:

1. Configure the PIC I/O lines to be used in the conversion. All analog lines are initialized as input in the corresponding TRIS registers.
2. Select the ports to be used in the conversion by setting the PCFGx bits in the ADCON1 register. Selects right- or left-justification.
3. Select the analog channels, select the A/D conversion clock, and enable the A/D module.
4. Wait the acquisition time.
5. Initiate the conversion by setting the GO/DONE bit in the ADCON0 register.
6. Wait for the conversion to complete.
7. Read and store the digital result.

CHAPTER 4

SYSTEM DESIGN AND RESULTS

The description of the circuit is presented in this chapter. PROTEUS software is used (version7pt4sp3) for the simulation the strength of its architecture has allowed us to integrate first conventional graph based simulation and now - with PROTEUS VSM - interactive circuit simulation into the design environment. It is possible to draw a complete circuit for a microcontroller based system and then test it interactively, all from within the same piece of software. Meanwhile, ISIS retains a host of features aimed at the PCB designer, so that the same design can be exported for production with ARES or other PCB layout software.

C language is used [34-37] for writing the program, mikro-Ccompiler Software (version 8pt2) for its feature. MikroC allows you to quickly develop and deploy complex applications:

- Write C source code using the built-in Code Editor (Code and Parameter Assistants, Syntax Highlighting, Auto Correct, Code Templates, and more…)
- Use the included mikroC libraries to dramatically speed up the development: data acquisition, memory, displays, conversions, communications… Practically all P12, P16, and P18 chips are supported.

- Monitor your program structure, variables, and functions in the Code Explorer.
- Generate commented, human-readable assembly, and standard HEX compatible with all programmers.
- Inspect program flow and debug executable logic with the integrated Debugger.
- Get detailed reports and graphs: RAM and ROM map, code statistics, assembly listing, calling tree, and more…
- We have provided plenty of examples for you to expand, develop, and use as building bricks in your projects. Copy them entirely if you deem fit – that's why we included them with the compiler.

4.1 block diagram of the system

Figure (4.1) shows the block diagram of the system, it is considered an A/D measurement system which contained signal conditioning block (operational amplifier), sample-and-hold block, A/D converter block and system controller block (these blocks contained in the microcontroller block) then the digital output displayed on LCD.

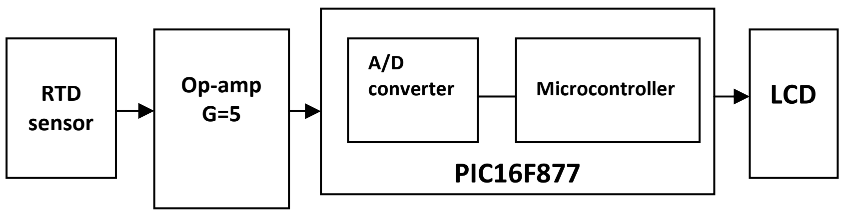


Fig. (4.1) Block Diagram of the System.

4.1.1 RTD sensor

Resistance temperature detectors measure temperature more directly.RTD is selected when sensitivity and flexibility are important. PT-100 RTD has a temperature coefficient of resistance of 0.0038Ω/Ω/^{o}C and 100 Ω resistance at 0 ^{o}C. It can be used to measure the temperature in the range -200 to 660 ^{o}C. Normally a constant current of around 1 mA is passed through the RTD and the voltage across it is measured using a high-impedance voltmeter. This type of measurement is called as four- wire transmitter.

$$T = \frac{-R_o A + [R^2_o A^2 - 4 R_o B (R_o - R_T)]^{0.5}}{2 R_o B} \qquad (4\text{-}1)$$

Where

T is the temperature,
R_0 is the resistance at 0^{o}C,
R_T is the resistance at T ^{o}C and,
A, B are constants

4.1.2 Operational amplifier

This is 741 op-amp [38-39] which used to amplify the RTD voltage; it is adjusted to give gain (5) its output goes to the microcontroller.

4.1.3 PIC16F877

This is the main block of the system .The PIC16F877 [40-41] contains a 8192 • 14 Flash EEPROM, 368 bytes of data RAM, 256 bytes of EEPROM, 33 I/O port pins, 8 channels of A/D converters,

I^2C and SPI bus compatible pins, PWM output, capture and compare registers, parallel slave port pins, power-on reset, watchdog timer, power-saving sleep mode, brown-out detection circuitry, in-circuit programming support, USART, timers, and 14 sources of interrupts, including an external interrupt and various internal interrupt sources. The amount of program memory provided by the PIC16F877 should be sufficient for many temperature monitoring and control applications. These applications also need large data memories since most of the operations use non-integer, floating point arithmetic, requiting several bytes to store a single variable in the data memory. Its built-in A/D converter is used to digitize the RTD voltage then calculate T and display it on the LCD.

4.1.4 LCD

Liquid crystal display is the digital display which used to monitor the temperature; it is LM016L (16*2) display [42].

4.2 Circuit Diagram of the System

4.2.1 Circuit Description

Figure (4.2) shows the schematic diagram of the circuit, In this circuit, a (5) volt constant power supply and resistor 1K are used to give exciting current to RTD which was replaced by resistor has the same values. The voltage of RTD is goes to the noninverting leg in 741-OP-AMP which adjusted to give gain (5), the output of it goes to ANI0 of PIC16F877 controller which its A/D converter was configured, LCD display was configured and it is connected to port B of the microcontroller, to monitor the temperature and used a led to give Alarm if the temperature is greater than or less than valid limit.

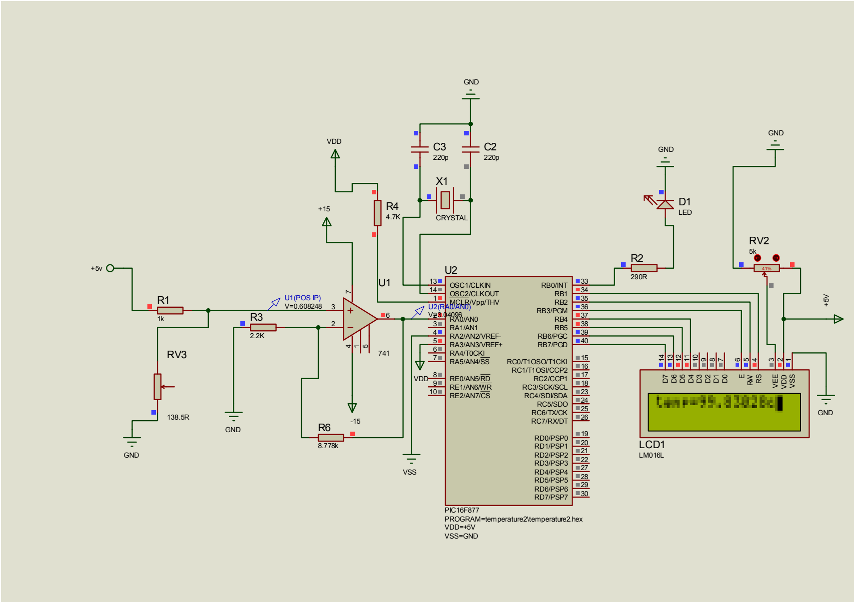


Fig. (4.2) Circuit Diagram of the System.

4.2.2 Operation of the Circuit

From the circuit diagram:

$R_5 = V_5.R_1 / 5\text{-}V_5$ (4-2)

Where

R_5 is RTD resistance,

V_5 is the voltage across the RTD, and

$R_1 = 1\text{k}\Omega$.

Thus:

$R_5 = 1000.V_5 / (5\text{-}V_5) \quad \Omega$

From equations (4.1) and (4.2) with

$R_0 = 100\ \Omega$, $A=3.9083*10^{-3}$ and $B= -5.775*10^{-7}$

Thus,

$$T = \frac{-0.39083 + [0.15274 - 2310*10^{-7} (R_T - 100)]^{0.5}}{-1155*10^{-7}} \qquad (4\text{-}3)$$

Thus, if R_T is known, we can calculate the temperature from Eq.4-3.

4.2.3 Circuit PDL

Start

Configure portA as input

Configure portB as output

Configure LCD

Configure A/D converter

DO forever

Read analog data of RTD (voltage) from channel 0

Calculate RTD resistance using equation (4-2)

Calculate the temperature t using equation (4-3)

Format the temperature for the display

Display the temperature on LCD

Wait for 1 second

END DO

Store upper limit & lower limit in EEPROM

If temperature greater than upper limit, LED ON

If temperature less than lower limit, LED OFF

If temperature between upper limit & lower limit, LED TOGLLE

END

4.3 EXPERMINTAL work

4.3.1 Hardware Requirements are:-

- microcontroller programmer;
- microcontroller development board or breadboard with the required components;
- microcontroller chips;
- personal computer;

A microcontroller programmer is connected to a PC and is used to download the program to the target microcontroller program memory. Its model is (BK PRECISION) and data is transferred to microcontroller by parrel cable, hexadecimal code is loaded to the EEPROM of it.

The system is initially built and tested on a test board. The system is to be used in industrial applications, so a printed circuit board design of the system is created.

A PC is required mainly for two purposes during the development of a microcontroller based project: the program is developed and compiled on the PC(C language and MikroC compiler was used for developing the project , (from the compiler it was found that : used ROM : 2770(33%), free ROM : 5421(67%),used RAM : 116(31%) and free RAM : 252(69%)) , and the PC is used to transfer the code to the device programmer so that the program memory of the target microcontroller can be loaded with the program.

4.3.2 Specifications

The system is required to have the following specifications:

Temperature range:	10ºc to 100 ºc accurate to a few ºc
Sensor:	Platinum RTD (replaced by variable Resistor)
Display type:	LCD
Display format:	6 characters, displayed as "nn.m C"
Display update:	1 second
System:	Microcontroller based

4.3.3 Experimental circuit:

As seen in Figure (4.2) variable resistor of 200Ω is used as input (instead of RTD sensor: 100Ω $\longrightarrow$ 0°c, 138.5 Ω $\longrightarrow$ 100 °c), PIC16F877 microcontroller is used to digitize the signal and display temperature on LCD. The circuit analysis was described in sections (4.2.1, 4.2.2). Figures (4.3, 4.4) show the system on the test board as seen in Figure (4.3), the LED was lighting because the temperature was above the limited range (100°c), in Figure (4.4) the LED was toggling because the temperature was between (10°c, 100°c) so the operation of the circuit is verified on the test board and it works good. Figure (4.5) shows the printed circuit board of the system

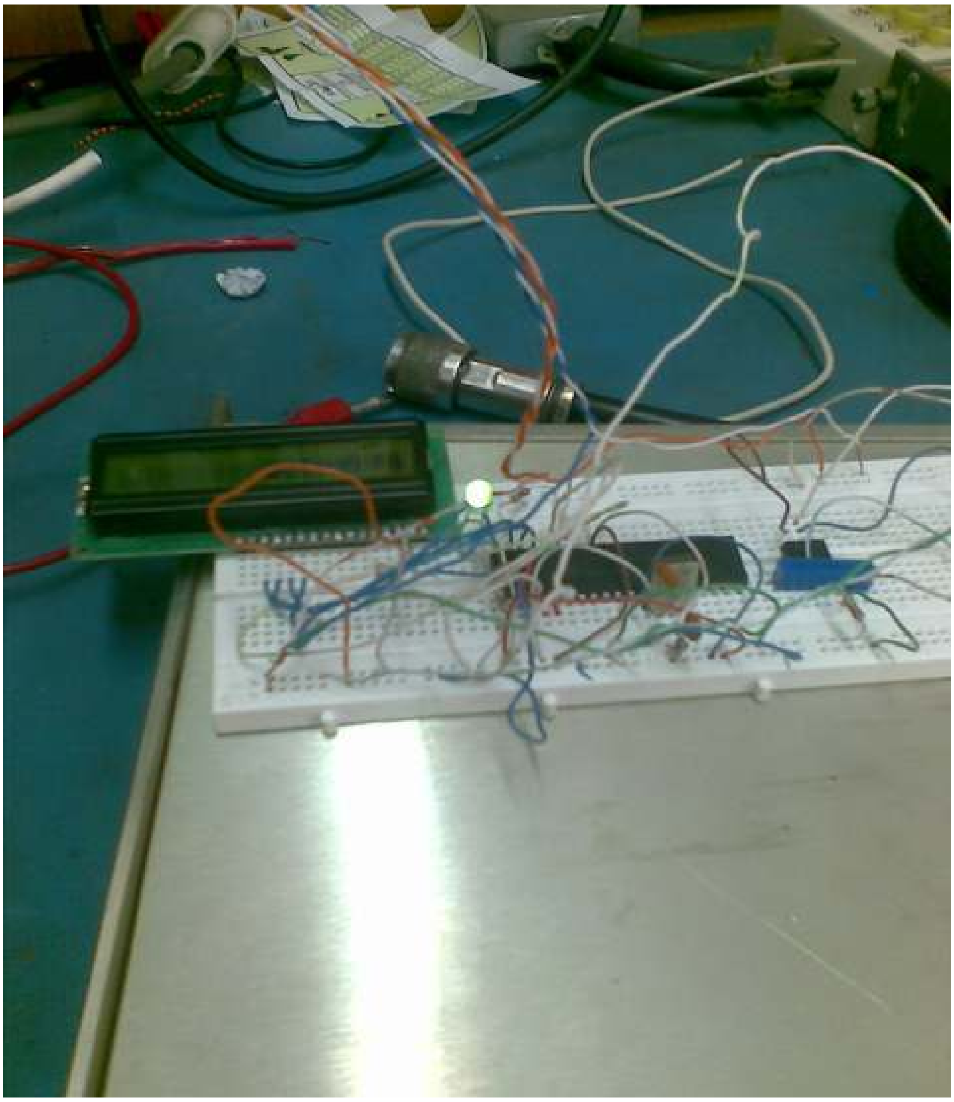


Fig. (4.3) the System on the Test Board

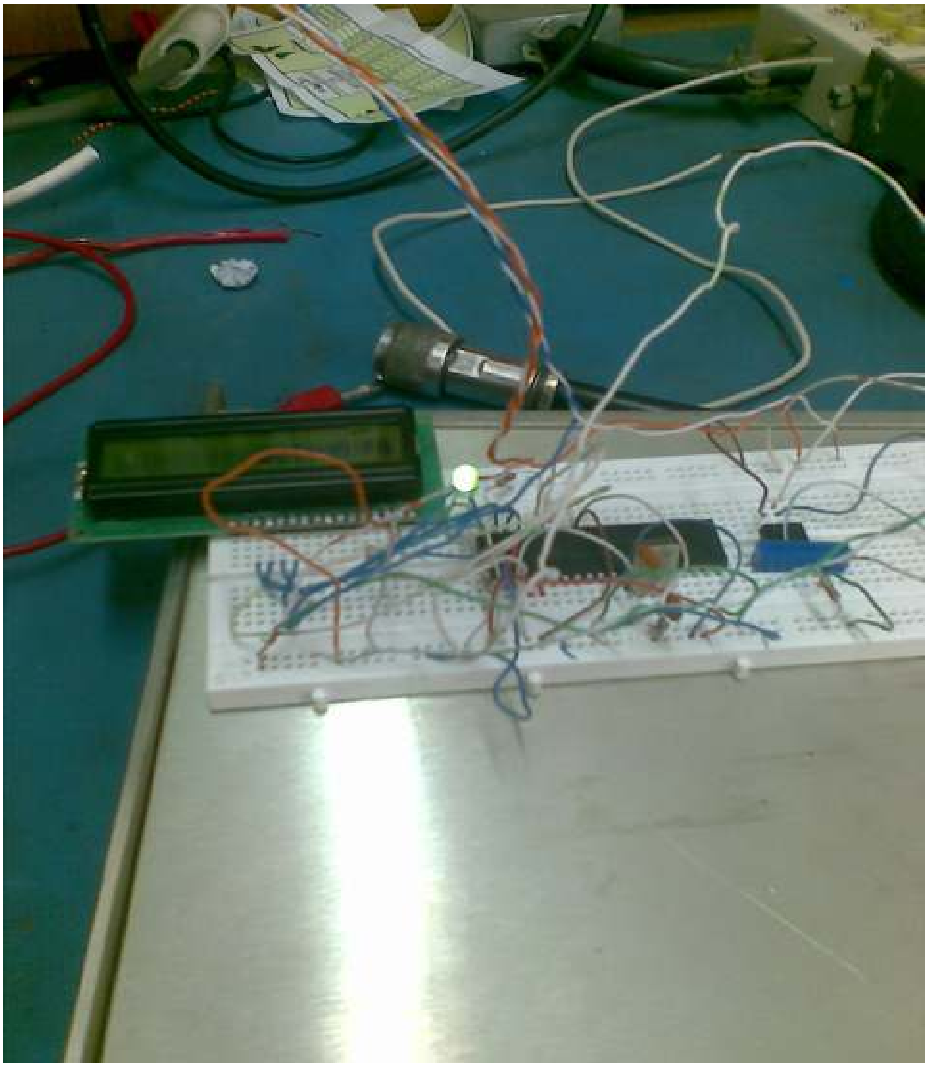


Fig. (4.4) the System on the Test Board with Led Toggling

Fig. (4.5) Printed Circuit Board of The System.

4.4 Results and Discussion

First, the hexadecimal code of the program is loaded to the microcontroller. Then run it and registered various ranges of temperature transmitters in the reactor ranges (10-100)˚C, (10-70)˚C and (0-50)˚C.These ranges found in various systems in the reactor. Table(4-1) shows the comparison between simulated, experimental, field and reference values of temperature range(10-100)˚C, in this table reading of system 0200-TT-034 was taken, the percentage error was calculated (with respect to span and reference values of temperature) for experimental, field and simulated values of temperature, when simulated and experimental values of temperature are compared, the percentage of error for simulation and experimental is better than the field one(i.e. percentage of error for experimental and simulation is less than the field one(in seven values of temperature). Figures(4.6,4.7,4.8) show plot of field, simulation, experimental and reference values of temperature, experimental, field and simulation error%(with respect to reference values of temperature and span) between input resistance and output temperature. As seen in Figure (4.6) the relation between input resistance and output temperature for T_r, T_f, T_s and T_p is considered linear, we can distinguish between them if colors are used. It starts from 100Ω because the RTD is (pt100) i.e. has 100Ω at 0˚c.

.

Table (4-1) A Values of Resistance & Temperature in Field, Practical & Simulation for range (10-100) ^{o}c.

R	T_{ref} (1)	T_f (2)	T_s (3)	T_p (4)
103.9	10	10.26	10.37	10.299
107.4	19	19.49	18.972	18.891
110.9	28	28.28	28.266	28.275
114.38	37	37.55	37.018	37.009
117.85	46	46.04	45.847	45.915
121.32	55	55.57	55.394	55.371
124.77	64	64.1	64.386	64.297
128.22	73	73.58	72.807	72.891
131.66	82	81.85	81.957	81.976
135.08	91	91.83	91.185	91.213
138.50	100	99.96	99.839	99.861

Table (4-1) B

R	T_P error%	T_F error %	T_S error%
103.9	2.99	2.6	3.7
107.4	-0.57368	2.57895	-0.1474
110.9	0.98214	1	0.95
114.38	0.02432	1.48649	0.04865
117.85	-0.18478	0.08696	-0.3326
121.32	0.67454	1.03636	0.71636
124.77	0.46406	0.15625	0.60312
128.22	-0.149315	0.79452	-0.2644
131.66	-0.02926	-0.1829	-0.061
135.08	0.234066	0.91209	0.2033
138.50	-0.139	-0.04	-0.17

Table (4-1) C

R	T_P error(s)%	T_F error(s)%	T_S error(s)%
103.9	0.33222	0.28889	0.4111
107.4	-0.12111	0.54444	-0.031
110.9	0.30555	0.31111	0.2956
114.38	0.01	0.61111	0.02
117.85	-0.09444	0.04444	-0.17
121.32	0.41222	0.63333	0.4378
124.77	0.33	0.11111	0.4289
128.22	-0.12111	0.64444	-0.214
131.66	-0.02667	-0.1667	-0.056
135.08	0.23667	0.92222	0.2056
138.50	-0.15444	-0.0444	-0.189

(1)reference temperature from tables, (2) field temperature from MPR, (3) simulated temperature from program, (4) Practical temperature

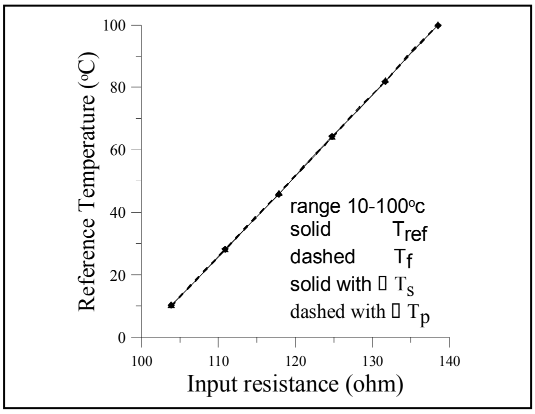


Fig. (4.6) Relation between Input Resistance & Output Temperature Range (10-100) ºC.

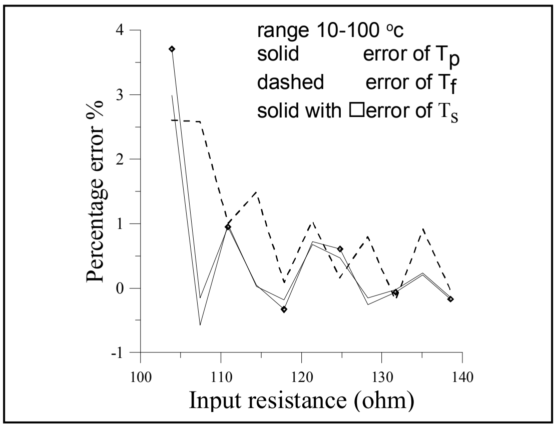


Fig. (4.7) Relation between Practical, Field and Simulated Error% Referenced to Reference Temperature range (10-100) ºC.

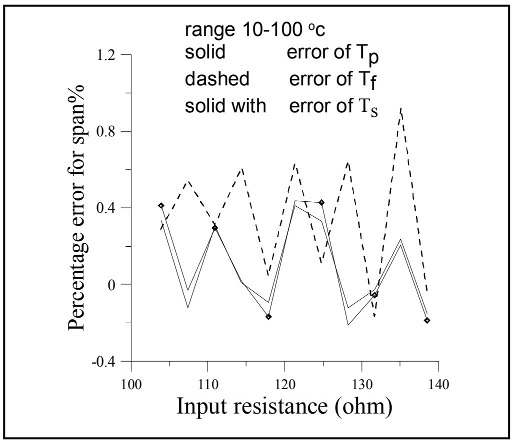


Fig. (4.8) Relation between Practical, Field and Simulated Error% Referenced to Span Temperature Range (10-100) ºC.

Table (4-2) shows the comparison between simulated, experimental, field and reference values of temperature range(10-70)℃, in this table reading of system 0200-TT-007 was taken, the percentage error was calculated (with respect to span and reference values of temperature) for experimental, field and simulated values of temperature, when simulated and experimental values of temperature are compared, the percentage of error for simulation and experimental is better than the field one(i.e. percentage of error for experimental and simulation is less than the field one (in eight values of temperature)).Figures(4.9,4.10,4.11) show plot of field, simulation, experimental and reference values of temperature, experimental, field and simulation error %(with respect to reference values of temperature and span) between input resistance and output temperature. As seen in Figure (4.9) the relation between input resistance and output temperature for T_r , T_f, T_s and T_p is considered linear, we can distinguish between them if colors are used. It starts from 100Ω because the RTD is (pt100) i.e. has 100Ω at 0̊c.

Table (4-2) A Values of Resistance & Temperature in Field, Practical & Simulation for range (10-70) °c.

R	T_{ref} (1)	T_f (2)	T_s (3)	T_p (4)
103.90	10	10.26	10.370	10.299
106.16	16	16.27	15.893	15.913
108.47	22	21.19	22.059	22.072
110.90	28	27.14	28.266	28.275
113.09	34	34.4	33.884	33.872
115.54	40	39.09	40.162	40.153
117.71	46	46.01	45.848	45.915
120.16	52	51.01	52.202	52.192
122.33	58	57.43	57.955	57.975
124.77	64	63.17	64.386	64.297
127.07	70	69.97	70.209	70.197

Table (4-2) B

R	T_P error%	T_F error %	T_S error%
103.90	2.99	2.6	3.74
106.16	-0.54375	1.6875	-0.66875
108.47	0.32727	-3.68182	0.272272
110.90	0.98214	-3.07143	0.95
113.09	-0.37647	1.17647	-0.341176
115.54	0.3825	-2.275	0.405
117.71	-0.18478	0.02174	-0.330434
120.16	0.36923	-1.90385	0.388461
122.33	-0.04310	-0.98276	-0.077586
124.77	0.46406	-1.29688	0.603125
127.07	0.28143	-0.04286	0.298571

Table (4-2) C

R	T_P error(s)%	T_F error(s)%	T_S error(s)%
103.9	0.49833	0.43333333	0.62333
107.4	-0.145	0.45	-0.1783
110.9	0.12	-1.35	0.09983
114.38	0.45833	-1.43333333	0.44333
117.85	-0.21333	0.66666666	-0.1933
121.32	0.255	-1.51666666	0.27
124.77	-0.14166	0.01666666	-0.2533
128.22	0.32	-1.65	0.33666
131.66	-0.04166	-0.95	-0.075
135.08	0.495	-1.38333333	0.64333
138.50	0.32833	-0.05	0.34833

(1) Reference temperature from tables, (2) field temperature from MPR, (3) simulated temperature from program, (4) Practical temperature

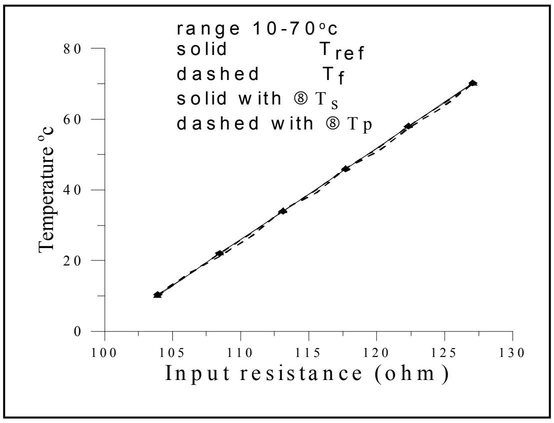


Fig. (4.9) Relation between Input Resistance & Output Temperature range (10-70)°c.

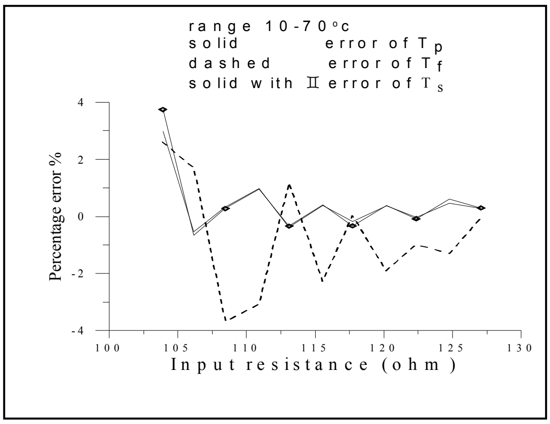


Fig. (4.10) Relation between Practical, Field and Simulated error% Referenced to Reference Temperature range (10-70) °C.

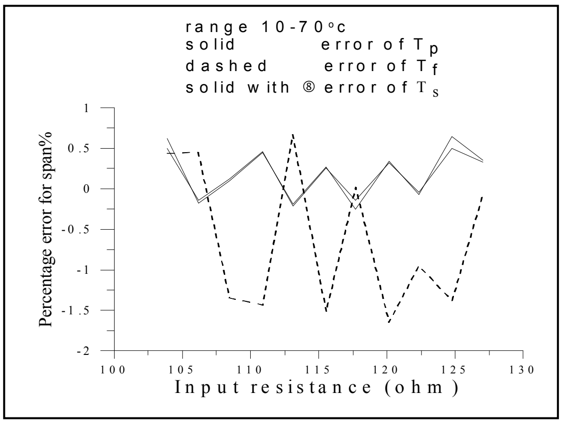


Fig. (4.11) Relation between Practical, Field and Simulated error% Referenced to span temperature range (10-70) °C.

Table (4-3) shows the comparison between simulated, experimental, field and reference values of temperature range(0-50)°C, in this table reading of system 0620-TT-032 was taken, the percentage error was calculated (with respect to span and reference values of temperature) for experimental, field and simulated values of temperature, when simulated and experimental values of temperature are compared, the percentage of error for simulation and experimental is better than the field one (i.e. percentage of error for simulation and practical is less than the field one(in six values of temperature)).Figures(4.12, 4.13, 4.14) show plot of field, simulation, experimental and reference values of temperature, experimental, field and simulation error %(with respect to reference values of temperature and span) between input resistance and output temperature. As seen in Figure (4.12) the relation between input resistance and output temperature for T_r, T_f, T_s and T_p is considered linear, we can distinguish between them if colors are used. It starts from 100Ω because the RTD is (pt100) i.e. has 100Ω at 0°c.

Table (4-3) A Values of Resistance & Temperature in Field, Practical & Simulation for range (0-50) ^{o}c.

R	$T_{ref}^{(1)}$	$T_{f}^{(2)}$	$T_{s}^{(3)}$	$T_{p}^{(4)}$
100.000	0	0.7	0.0335	0.0314
101.925	5	4.69	4.886	4.897
103.900	10	10.16	10.370	10.298
105.775	15	14.79	15.278	15.198
107.700	20	20.12	20.205	20.216
109.625	25	24.66	25.158	25.135
111.550	30	30.2	30.135	30.098
113.610	35	35.23	35.136	35.157
115.540	40	40.6	40.162	40.201
117.470	45	45.65	45.214	45.198
119.400	50	49.98	50.291	50.218

Table (4-3) B

R	T_P error%	T_F error %	T_S error%
100.000	#DIV/0!	#DIV/0!	#DIV/0!
101.925	-2.06	-6.2	-2.28
103.900	2.98	1.6	3.7
105.775	1.32	-1.4	1.85333
107.700	1.08	0.6	1.025
109.625	0.54	-1.36	0.632
111.550	0.32667	0.6667	0.45
113.610	0.44857	0.6571	0.38857
115.540	0.5025	1.5	0.405
117.470	0.44	1.444	0.47555
119.400	0.436	-0.04	0.582

Table (4-3) C

R	T_P error(s)%	T_F error(s)%	T_S error(s)%
100.000	0.0628	1.4	0.067
101.925	-0.206	-0.62	-0.228
103.900	0.596	0.32	0.74
105.775	0.396	-0.42	0.556
107.700	0.432	0.24	0.41
109.625	0.27	-0.68	0.316
111.550	0.196	0.4	0.27
113.610	0.314	0.46	0.272
115.540	0.402	1.2	0.324
117.470	0.396	1.3	0.428
119.400	0.436	-0.04	0.582

(1) Reference temperature from tables, (2) field temperature from MPR, (3) simulated temperature from program, (4) Practical temperature

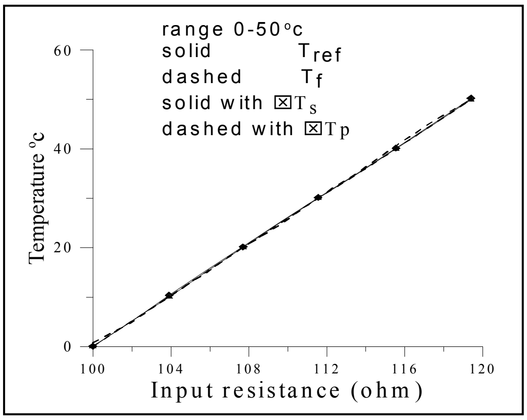


Fig. (4.12) Relation between Input Resistance & Output Temperature range (0-50) ^{o}C.

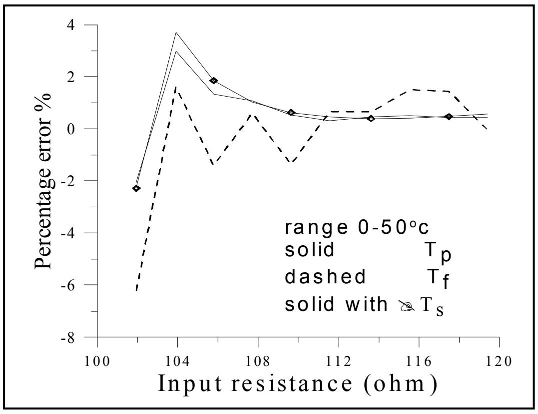


Fig. (4.13) Relation between Practical, Field and Simulated error% Referenced to Reference Temperature range (0-50) ^{o}C.

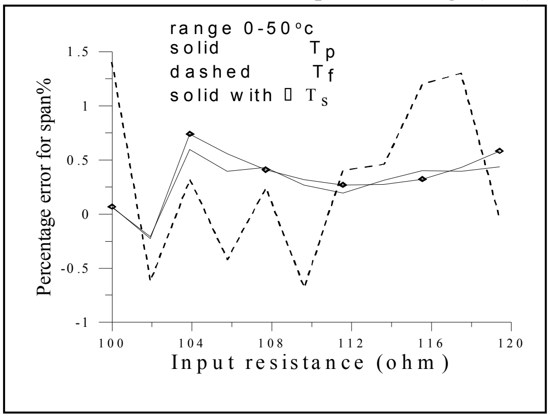


Fig. (4.14) Relation between Practical, Field and Simulated error% Referenced to Span Temperature range (0-50) ^{o}C.

Mathematical equations for relation between resistance and temperature for various ranges of temperature are given in the following equations:

Relation between T_r, T_f, T_s and T_p and R for range (10-100) ^{o}c:

$T_r = -\ 260.4399403 + 2.601117966\ R$ (4. 4.a)

$T_f = -\ 248.4942304 + 2.411201283\ R + 0.0007692216082\ R^2$ (4.4.b)

$T_s = -\ 259.5132599 + 2.594256929\ R$ (4.4.c)

$T_p = -\ 259.8235657 + 2.596783477\ R$ (4.4.d)

Relation between T_r, T_f, T_s and T_p and R for range (10-70) ^{o}c

$T_r = 2.587979153\ R - 258.8174838$ (4.5.a)

$T_f = 16032.18617 - 563.5072239\ R + 7.368340777\ R^2 - 0.0425766007\ R^3 +$

$9.214922645 * 10^{-5}\ R^4$ (4.5.b)

$T_s = -\ 259.0361997 + 2.59084497\ R$ (4.5.c)

$T_p = -\ 259.004664 + 2.590513585\ R$ (4.5.d)

Relation between T_r, T_f, T_s and T_p and R for range (0-50) ^{o}c

$T_r = -\ 257.3811507 + 2.574559081\ R$ (4.6.a)

$T_f = -234.8916699 + 2.154213571\ R + 0.001970248937\ R^2$ (4.6.b)

$T_s = -\ 258.0441521 + 2.582152577\ R$ (4.6.c)

$T_p = -\ 258.0777484 + 2.582275627\ R$ (4.6.d)

The relations for errors not occurs because their sharp changes, and so to calculate any errors we must be subtract the two equations instead of the complex equation of error.

For R=117.85Ω, from equations (4-4(a, b, c.d)), it was found:

T_r = 46.1018°c, T_f = 46.3493°c, T_s = 46.2199°c, T_p = 46.2074°c

T_F error % = 0.773, T_S error%=-0.2562, T_P error% =0.228996

T_F error % = 0.087, T_S error%= -0.3326, T_P error% = 0.1848(from table (4-1)).

For R=131.66 Ω, from equations (4-4(a, b, c.d)), it was found:

T_r = 82.0233°c, T_f = 82.2985°c, T_s = 82.0466°c, T_p = 82.0689°c

T_F error % = 0.3356, T_S error%=-0.0285, T_P error% =0.0557

T_F error % = -0.1829, T_S error%= -0.061, T_P error% =-0.0293(from table (4-1)).

If error percentages compared, it would found that the difference between them is small.

Figures (4.15, 4.16, 4.17) show fitting curves of $T_{p,}$ $T_{f,}$ $T_{s,}$ and T_r of above relations for ranges (10-100) °c, (10-70) °c, (0-50) °c.

Table (4-4) shows the comparison between proposed system and MPR system, from the obtained results in tables (4-1, 4-2, 4-3); it was found that the proposed system is more accurate than the MPR system. From Figures (2.7, 4.1) it was found that the proposed system is cheaper than the MPR system.

Table (4-4) Comparison between Existing System & Proposed System

System property	MPR system	Proposed system
Accuracy	Accurate(0.8°c)	More accurate(0.3°c)
Cost	expensive	cheap
Response Time	5sec.	5sec.
Resolution	10 bit	10 bit

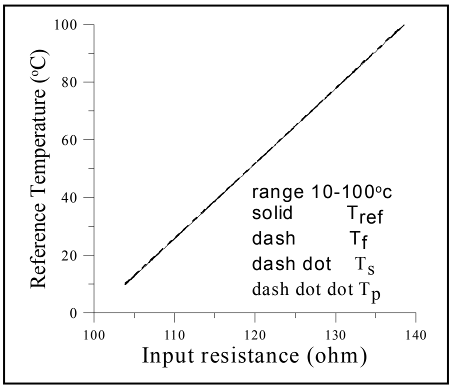


Fig. (4.15) fitting curves of T_{ref}, T_f, T_s, T_p range (10-100) ^oC

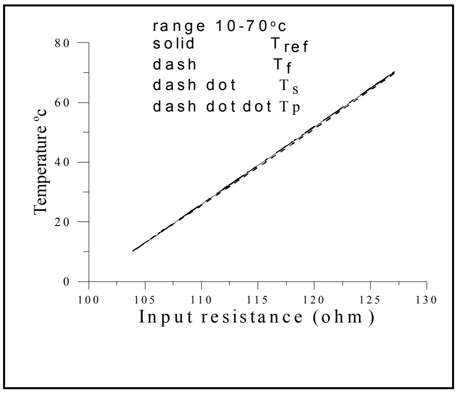


Fig. (4.16) fitting curves of T_{ref}, T_f, T_s, T_p range (10-70) ^oC

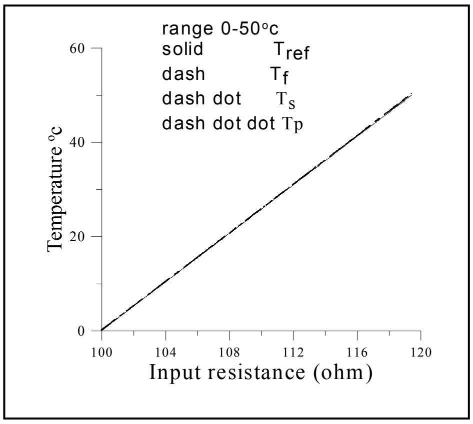


Fig. (4.17) fitting curves of T_{ref}, T_f, T_s, T_p range (0-50) ^oC

CHAPTER (5)

CONCLUSION

- Multiple purpose reactor system is a complicated and expensive system, in the work plane; this system will be re-implemented or re-designed with other techniques which give the same results easier and cheaper.
- In this work a digital system is implemented by using microcontroller techniques.
- Temperature is taken as an example.
- The complicated and much cost system is replaced by a system depends on one chip.
- RTD is the sensing element of temperature, which its resistance increases with temperature, and it's used when accuracy is needed.
- RTD operates on the principle that the electrical resistance of metals changes with temperature.
- In practice, metals with high melting points which can withstand the effects of corrosion and those with high resistivity are used for temperature sensing.

- RTDs have excellent accuracies over a wide temperature range .The most commonly used international RTD standard is the IEC 751 which is based on platinum RTDs with a resistance of 100 Ω at $0^{O}C$ and α parameter of 0.00385 and that is we used.
- PIC (Programmable Interface Controller) microcontroller is used for interfacing the transmitter, which has attractive features and it is suitable for a wide range of applications.
- The PIC was the first widely available device to use flash memory, which makes it ideal for experimental work.
- Flash memory allows the program to be replaced quickly and easily with a new version.
- PIC16F877 is used, this is now used widely as a more advanced teaching device, because it has a full complement of interfaces: analog input, serial ports, slave port, built-in A/D converter and so on, plus a good range of hardware timers.
- In this work, built-in A/D converter and EEPROM of the PIC are used.
- The PIC is programmed by using C language, mikro-C compiler and PROTEUS is used for designing the circuit and simulating it.

- The hexadecimal code of the program is loaded to the microcontroller then run it and register various ranges of temperature transmitters in the reactor (ranges (10-100)$^{\circ}$ C, (10-70)$^{\circ}$ C and (0-50)$^{\circ}$ C. for range(10-100)$^{\circ}$ C reading of system 0200-TT-007was taken, for range(10-70)$^{\circ}$C reading of system0200-TT-034was taken and for range(0-50)$^{\circ}$C reading of system 0620-TT-032 was taken .
- Error% curves were plotted for these values, when field values, simulated and practical values of temperature are compared, the percentage of error for simulation and practical is better than the field one.

REFERENCES

[1] D. Ibrahim " Advanced PIC Microcontroller Projects in C", Elsevier Ltd, Oxford PP.117-218 (2008).

[2] D. Ibrahim " Microcontroller Based Applied Digital Control", John Wiley & Sons Ltd, England. PP. 86-126 (2006).

[3] W. M. El-Medany, A. Alomary, R. Al-Hakim, S. Al-Irhayim, M. Nousif, "Implementation of GPRS-Based Positioning System Using PIC Microcontroller" Computational ,Communication Systems and Networks, cicsyn,2nd International Conference PP.365-368, 2010

[4] Bateson "Introduction to Control System Technology", 4th ed. Merrill, Columbus, OH. R.N. (1993).

[5] T. V.Sickle,"Programming Microcontroller in C", Eagle Rock, Virginia,(2001).

[6] J. Iovine "PIC Microcontroller Project Book" , McGraw-Hill, New York, (2000).

[7] H. M. Hashemian "Maintenance of Process Instrumentation in Nuclear Power Plants", Springer-Verlag Berlin Heidelberg PP 47-68 (2006).

[8]J.W.Chaplin "Instrumentation and Automation for Manufacturing" , Delmar, Albany, NY. (1992).

[9] E.O. Doeblin, "Measurement Systems Application and Design", 4th ed.McGraw-Hill, New York (1990).

[10] R.C.Dorf, and R.H. Bishop, "Modern Control Systems", 7th ed. Addison-Wesley, Reading, MA (1995).

[11] T.H. O'Dell "Circuits for Electronic Instrumentation", Cambridge Univ. Press, Cambridge, England, UK (1991).

[12] R.G. Seippel "Transducers, Sensors, and Detectors", Reston Pub. Reston VA (1983).

[13] J.Webb and K.Greshock "Industrial Control Electronics", 2nd ed. Macmillan, New York (1993).
[14] B. A. Gregory "An Introduction to Electrical Instrumentation", Macmillan, London (1973)
[15] P. Horowitz, and W.M. Hill "The Art of Electronics", 2nd ed. Cambridge University Press, New York (1989).

[16] E.J.P.Santos, I.B. Vascon" RTD based smart Temperature sensor: process development and circuit design" microelectronics department,MIEL 08. 26th International Conference, PP.333-336 July (2008).
[17] M.C. Douglas "Process Instruments and Controls Handbook", 2nd ed. Los Angeles, California (1974).
[18] D. Buchla and W. McLachlan "Applied Electronic Instrumentation and Measurement", Macmillan New York (1992).
[19] G. Mike "The RF and Microwave Handbook", Editor in Chief Mike Golio, PP.1296-1318(2001).
[20] D. Ibrahim "Microcontroller Based Temperature Monitoring and Control", Elsevier Science & Technology Books, PP.89-106, September (2002)
[21] M. Bates "Interfacing PIC Microcontrollers Embedded Design by Interactive Simulation", Elsevier (2006).
[22] A. Premadi, M.S.Ab. Rahman, N.B.Chuan,M.Saupe,K.Jumari" Application of PIC Microcontroller for online Monitoring and Fiber Fault Identification" Electrical Engineering and Informatics,ICEEI 09.International Conference, PP.463-467 September (2009).
[23] J. Sanchez and M. P. Canton" Microcontroller Programming", Taylor & Francis Group (2007).
[24] N. Gardner and P. Birnie "PIC Cook Book", Bluebird Electronics (1996).
[25] F. Stevens "Getting Started With PIC Microcontrollers", A.L. Stevens (1997)
[26] J. Lovine "PIC Robotics", McGraw-Hill companies (2004)
[27] N. Gardner "PIC Micro MCU C® an Introduction to Programming the Microchip PIC in CCS C", Bluebird Electronics (2002).

[28] S. Ball "Analog Interfacing to Embedded Microprocessor Real world design", Butterworth-Heinemann- Elsevier Group PP. 286-327(2001).
[29] T. Wilmshurst "Designing Embedded Systems with PIC Microcontrollers", Timothy Wilmshurs ,Elsevier Ltd.(2007).
[30] T. Wilmshurst "An Introduction to the Design of Small-Scale Embedded Systems", Palgrave.ISBN 0-333-92994-2 (2001).

[31] R. H. Bishop"Mechatronics Handbook", Taylor& Francis Group (2008).
[32] A.C.Fisher-Cripps"Newens interfacing companion", A.C.Fisher-Cripps (2002)
[33] M.Austin and D. Chancogne "Engineering Programming in C, Matlab and Java", Wiley, New York (1999).
[34] P. Prinz, and U.K. Prinz "C Pocket Reference", O'Reilly. ISBN 0-596-00436-2. (2003).
[35] M. J. Pont "Embedded C. Addison", Wesley, Great Britain. ISBN 0 201 79523 X. (2002).
[36] E.V. Sickle "Programming Microcontrollers in C", 2nd ed. Elsevier, USA. (2003).
[37] J. K. Roberge "Operational Amplifiers", John Wiley & Sons, New York (1975).
[38] D. F. Stout and M.Kaufman "Handbook of Operational Amplifier Circuit Design", McGraw-Hill, New York (1976).
[39] Website of Microchip Technology Inc.: www.microchip.com (2004)
[40] PIC 16F87XA Data Sheet. Microchip Technology Inc., Reference no. DS39582B; www.microchip.com (2005).
[41] PIC 16F87XA Flash Memory Programming Specification. DS39589. www.microchip.com (2003).
[42] HD44780 Data. Available from a number of websites, including http://www.electronic engineering.ch/microchip/index.html(2005)

i. APPENDIX A

RTD Temperature vs. Resistance Table

RTD Temperature Vs. Resistance Table

or European Curve, Alpha = .00385

1° Celsius Increments

°C	Ohm	Diff.	°C	Ohm	Diff.	°C	Ohm	Diff.	°C	Ohm	Diff.	°C	Ohm	Diff.	°C	Ohm	Diff.
−200	18.49		−140	43.87	0.42	−80	68.33	0.41	−20	92.16	0.39	± 0	100.00	0.39	+ 60	123.24	0.38
199	18.93	0.44	139	44.28	0.41	79	68.73	0.40	19	92.55	0.39	+ 1	100.39	0.39	61	123.62	0.38
198	19.36	0.43	138	44.70	0.42	78	69.13	0.40	18	92.95	0.40	2	100.78	0.39	62	124.01	0.39
197	19.79	0.43	137	45.11	0.41	77	69.53	0.40	17	93.34	0.39	3	101.17	0.39	63	124.39	0.38
196	20.22	0.43	136	45.52	0.41	76	69.93	0.40	16	93.73	0.39	4	101.56	0.39	64	124.77	0.38
195	20.65	0.43	135	45.94	0.42	75	70.33	0.40	15	94.12	0.39	5	101.95	0.39	65	125.16	0.39
194	21.08	0.43	134	46.35	0.41	74	70.73	0.40	14	94.52	0.40	6	102.34	0.39	66	125.54	0.38
193	21.51	0.43	133	46.76	0.41	73	71.13	0.40	13	94.91	0.39	7	102.73	0.39	67	125.92	0.38
192	21.94	0.43	132	47.18	0.42	72	71.53	0.40	12	95.30	0.39	8	103.12	0.39	68	126.31	0.39
191	22.37	0.43	131	47.59	0.41	71	71.93	0.40	11	95.69	0.39	9	103.51	0.39	69	126.69	0.38
190	22.80	0.43	130	48.00	0.41	70	72.33	0.40	10	96.09	0.40	10	103.90	0.39	70	127.07	0.38
189	23.23	0.43	129	48.41	0.41	69	72.73	0.40	9	96.48	0.39	11	104.29	0.39	71	127.45	0.38
188	23.66	0.43	128	48.82	0.41	68	73.13	0.40	8	96.87	0.39	12	104.68	0.39	72	127.84	0.39
187	24.09	0.43	127	49.23	0.41	67	73.53	0.40	7	97.26	0.39	13	105.07	0.39	73	128.22	0.38
186	24.52	0.43	126	49.64	0.41	66	73.93	0.40	6	97.65	0.39	14	105.46	0.39	74	128.60	0.38
185	24.94	0.42	125	50.06	0.42	65	74.33	0.40	5	98.04	0.39	15	105.85	0.39	75	128.98	0.38
184	25.37	0.43	124	50.47	0.41	64	74.73	0.40	4	98.44	0.40	16	106.24	0.39	76	129.37	0.39
183	25.80	0.43	123	50.88	0.41	63	75.13	0.40	3	98.83	0.39	17	106.63	0.39	77	129.75	0.38
182	26.23	0.43	122	51.29	0.41	62	75.53	0.40	2	99.22	0.39	18	107.02	0.39	78	130.13	0.38
181	26.65	0.42	121	51.70	0.41	61	75.93	0.40	1	99.61	0.39	19	107.40	0.38	79	130.51	0.38
180	27.08	0.43	120	52.11	0.41	60	76.33	0.40				20	107.79	0.39	80	130.89	0.38
179	27.50	0.42	119	52.52	0.41	59	76.73	0.40				21	108.18	0.39	81	131.27	0.38
178	27.93	0.43	118	52.92	0.40	58	77.13	0.40				22	108.57	0.39	82	131.66	0.39
177	28.35	0.42	117	53.33	0.41	57	77.52	0.39				23	108.96	0.39	83	132.04	0.38
176	28.78	0.43	116	53.74	0.41	56	77.92	0.40				24	109.35	0.39	84	132.42	0.38
175	29.20	0.42	115	54.15	0.41	55	78.32	0.40				25	109.73	0.38	85	132.80	0.38
174	29.63	0.43	114	54.56	0.41	54	78.72	0.40				26	110.12	0.39	86	133.18	0.38
173	30.05	0.42	113	54.97	0.41	53	79.11	0.39				27	110.51	0.39	87	133.56	0.38
172	30.47	0.42	112	55.38	0.41	52	79.51	0.40				28	110.90	0.39	88	133.94	0.38
171	30.90	0.43	111	55.78	0.40	51	79.91	0.40				29	111.28	0.38	89	134.32	0.38
170	31.32	0.42	110	56.19	0.41	50	80.31	0.40				30	111.67	0.39	90	134.70	0.38
169	31.74	0.42	109	56.60	0.41	49	80.70	0.39				31	112.06	0.39	91	135.08	0.38
168	32.16	0.42	108	57.00	0.40	48	81.10	0.40				32	112.45	0.39	92	135.46	0.38
167	32.59	0.43	107	57.41	0.41	47	81.50	0.40				33	112.83	0.38	93	135.84	0.38
166	33.01	0.42	106	57.82	0.41	46	81.89	0.39				34	113.22	0.39	94	136.22	0.38
165	33.43	0.42	105	58.22	0.40	45	82.29	0.40				35	113.61	0.39	95	136.60	0.38
164	33.85	0.42	104	58.63	0.41	44	82.69	0.40				36	113.99	0.38	96	136.98	0.38
163	34.27	0.42	103	59.04	0.41	43	83.08	0.39				37	114.38	0.39	97	137.36	0.38
162	34.69	0.42	102	59.44	0.40	42	83.48	0.40				38	114.77	0.39	98	137.74	0.38
161	35.11	0.42	101	59.85	0.41	41	83.88	0.40				39	115.15	0.38	99	138.12	0.38
160	35.53	0.42	100	60.25	0.40	40	84.27	0.39				40	115.54	0.39	100	138.50	0.38
159	35.95	0.42	99	60.66	0.41	39	84.67	0.40				41	115.93	0.39	101	138.88	0.38
158	36.37	0.42	98	61.06	0.40	38	85.06	0.39				42	116.31	0.38	102	139.26	0.38
157	36.79	0.42	97	61.47	0.41	37	85.46	0.40				43	116.70	0.39	103	139.64	0.38
156	37.21	0.42	96	61.87	0.40	36	85.85	0.39				44	117.08	0.38	104	140.02	0.38
155	37.63	0.42	95	62.28	0.41	35	86.25	0.40				45	117.47	0.39	105	140.39	0.37
154	38.04	0.41	94	62.68	0.40	34	86.64	0.39				46	117.85	0.38	106	140.77	0.38
153	38.46	0.42	93	63.09	0.41	33	87.04	0.40				47	118.24	0.39	107	141.15	0.38
152	38.88	0.42	92	63.49	0.40	32	87.43	0.39				48	118.62	0.38	108	141.53	0.38
151	39.30	0.42	91	63.90	0.41	31	87.83	0.40				49	119.01	0.39	109	141.91	0.38
150	39.71	0.41	90	64.30	0.40	30	88.22	0.39				50	119.40	0.39	110	142.29	0.38
149	40.13	0.42	89	64.70	0.40	29	88.62	0.40				51	119.78	0.38	111	142.66	0.37
148	40.55	0.42	88	65.11	0.41	28	89.01	0.39				52	120.16	0.38	112	143.04	0.38
147	40.96	0.41	87	65.51	0.40	27	89.40	0.39				53	120.55	0.39	113	143.42	0.38
146	41.38	0.42	86	65.91	0.40	26	89.80	0.40				54	120.93	0.38	114	143.80	0.38

ii. APPENDIX B
C Program of the Project

```
# define LED PORTB.F0

#define high_temp 100

#define low_temp 10

void main()

{

float rtdv,rtdr,temp,y,LSB,data1,data2;

unsigned char i,j,lcd[5],hi_flag,low_flag;

unsigned char temperature[13];

TRISA=0XFF;

TRISB=0;

PORTA=0;

PORTB=0;

Lcd_Config(&PORTB,1,3,2,7,6, 5,4);

Lcd_cmd(Lcd_clear);

Lcd_out(1,1,"temp=");

delay_ms(1000);

LSB= 5000.0/1023.0;

while(1){

rtdv=ADC_Read(0);

 ADCON1=0X80;
```

```
while(ADCON0.GO){};

rtdv=rtdv*LSB/5.001;

rtdr=rtdv*1000.0/(5000.0-rtdv);

y=0.15274-(rtdr-100.0)*2310.0e-7;

if(y>=0)y=sqrt(y);

temp=(y-0.39083)/(-0.0001155);

//temp=(y-0.3908)/(-0.000116038);

FloatTostr (temp, temperature);

j=0;

for(i=0;i<=13;i++){

if (temperature[i]!=' '){

lcd[j]=temperature[i];

j++;

}

}

Lcd_out(1,6,lcd);

Lcd_chr_cp('c');

Delay_ms(1000);

EEprom_write(0x2f,high_temp);

delay_ms(50);

EEprom_write(0x3f,low_temp);

delay_ms(50);
```

```
data1=temp;
data2=data1;
if(data2>high_temp)
{hi_flag=0xff;
 low_flag=0;}
 else
if(data2<low_temp)
{low_flag=0xff;
 hi_flag=0;}
 else
{low_flag=0;
 hi_flag=0;}
if(hi_flag==0xff)
LED=1;
 else
if(low_flag==0xff)
LED=0;
else
 LED=~LED;
}
}
```

iii. APPENDIX C

PIC16F87X DATA SHEET

MICROCHIP

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

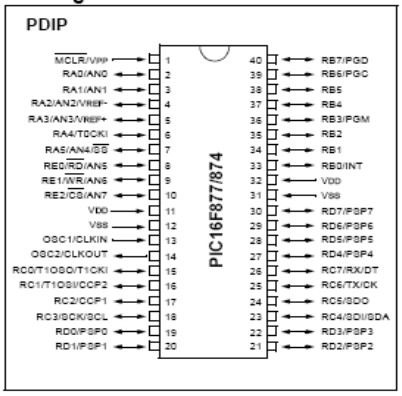
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F874
- PIC16F876
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM)
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature ranges
- Low-power consumption:
  - < 0.6 mA typical @ 3V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

Pin Diagram



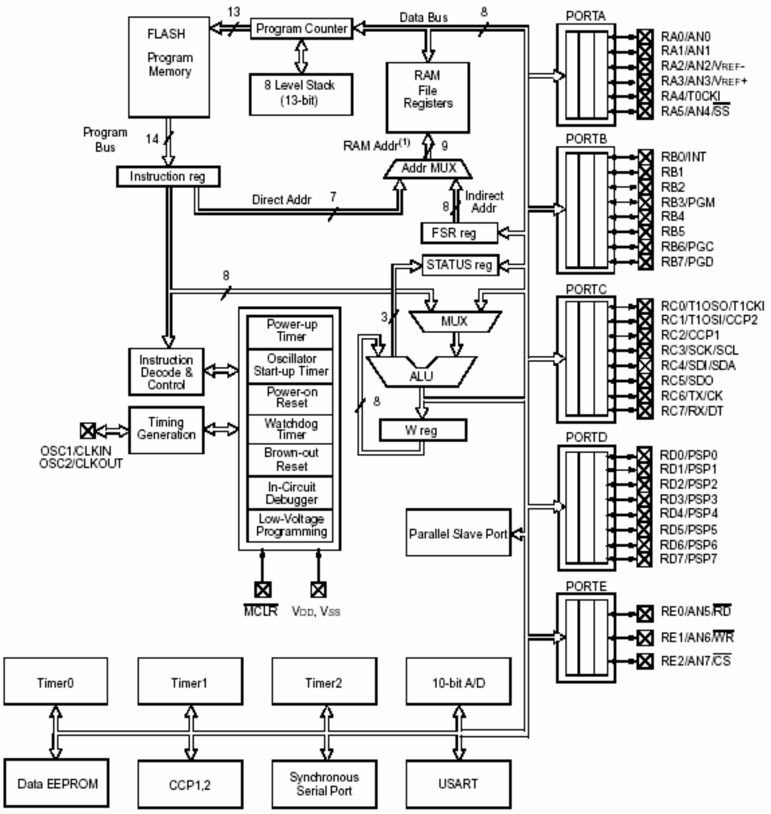
Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I^{2}C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions

FIGURE 1-2: PIC16F874 AND PIC16F877 BLOCK DIAGRAM

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS(4)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{\text{MCLR}}$/VPP	1	2	18	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
						PORTA is a bi-directional I/O port.
RA0/AN0	2	3	19	I/O	TTL	RA0 can also be analog input0.
RA1/AN1	3	4	20	I/O	TTL	RA1 can also be analog input1.
RA2/AN2/VREF-	4	5	21	I/O	TTL	RA2 can also be analog input2 or negative analog reference voltage.
RA3/AN3/VREF+	5	6	22	I/O	TTL	RA3 can also be analog input3 or positive analog reference voltage.
RA4/T0CKI	6	7	23	I/O	ST	RA4 can also be the clock input to the Timer0 timer/ counter. Output is open drain type.
RA5/$\overline{\text{SS}}$/AN4	7	8	24	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
						PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	33	36	8	I/O	TTL/ST(1)	RB0 can also be the external interrupt pin.
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3/PGM	36	39	11	I/O	TTL	RB3 can also be the low voltage programming input.
RB4	37	41	14	I/O	TTL	Interrupt-on-change pin.
RB5	38	42	15	I/O	TTL	Interrupt-on-change pin.
RB6/PGC	39	43	16	I/O	TTL/ST(2)	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.
RB7/PGD	40	44	17	I/O	TTL/ST(2)	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION (CONTINUED)

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
						PORTC is a bi-directional I/O port.
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	RC0 can also be the Timer1 oscillator output or a Timer1 clock input.
RC1/T1OSI/CCP2	16	18	35	I/O	ST	RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	17	19	36	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	18	20	37	I/O	ST	RC3 can also be the synchronous serial clock input/ output for both SPI and I^2C modes.
RC4/SDI/SDA	23	25	42	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	24	26	43	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6/TX/CK	25	27	44	I/O	ST	RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	26	29	1	I/O	ST	RC7 can also be the USART Asynchronous Receive or Synchronous Data.
						PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus.
RD0/PSP0	19	21	38	I/O	ST/TTL(3)	
RD1/PSP1	20	22	39	I/O	ST/TTL(3)	
RD2/PSP2	21	23	40	I/O	ST/TTL(3)	
RD3/PSP3	22	24	41	I/O	ST/TTL(3)	
RD4/PSP4	27	30	2	I/O	ST/TTL(3)	
RD5/PSP5	28	31	3	I/O	ST/TTL(3)	
RD6/PSP6	29	32	4	I/O	ST/TTL(3)	
RD7/PSP7	30	33	5	I/O	ST/TTL(3)	
						PORTE is a bi-directional I/O port.
RE0/$\overline{RD}$/AN5	8	9	25	I/O	ST/TTL(3)	RE0 can also be read control for the parallel slave port, or analog input5.
RE1/$\overline{WR}$/AN6	9	10	26	I/O	ST/TTL(3)	RE1 can also be write control for the parallel slave port, or analog input6.
RE2/$\overline{CS}$/AN7	10	11	27	I/O	ST/TTL(3)	RE2 can also be select control for the parallel slave port, or analog input7.
VSS	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
VDD	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28, 40	12,13, 33,34		—	These pins are not internally connected. These pins should be left unconnected.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 2-1.

The Special Function Registers can be classified into two sets: core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. Those related to the operation of the peripheral features are described in detail in the peripheral features section.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:
Bank 0											
00h[3]	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
01h	TMR0	Timer0 Module Register								xxxx xxxx	47
02h[3]	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26
03h[3]	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	18
04h[3]	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	29
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	31
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	33
08h[4]	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	35
09h[4]	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	36
0Ah[1,3]	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
0Bh[3]	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20
0Ch	PIR1	PSPIF[3]	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	22
0Dh	PIR2	—	(5)	—	EEIF	BCLIF	—	—	CCP2IF	-r-0 0--0	24
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	52
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	52
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	51
11h	TMR2	Timer2 Module Register								0000 0000	55
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	55
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	70, 73
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	67
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	57
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	57
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	58
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	96
19h	TXREG	USART Transmit Data Register								0000 0000	99
1Ah	RCREG	USART Receive Data Register								0000 0000	101
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	57
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	57
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	58
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	116
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	—	ADON	0000 00-0	111

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.

3: These registers can be addressed from any bank.

4: PORTD, PORTE, TRISD, and TRISE are not physically implemented on PIC16F873/876 devices; read as '0'.

5: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:
Bank 1											
80h(3)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
81h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	19
82h(3)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26
83h(3)	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	18
84h(3)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	29
86h	TRISB	PORTB Data Direction Register								1111 1111	31
87h	TRISC	PORTC Data Direction Register								1111 1111	33
88h(4)	TRISD	PORTD Data Direction Register								1111 1111	35
89h(4)	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	37
8Ah(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
8Bh(3)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20
8Ch	PIE1	PSPIE(2)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	21
8Dh	PIE2	—	(5)	—	EEIE	BCLIE	—	—	CCP2IE	-r-0 0--0	23
8Eh	PCON	—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	---- --qq	25
8Fh	—	Unimplemented								—	—
90h	—	Unimplemented								—	—
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	68
92h	PR2	Timer2 Period Register								1111 1111	55
93h	SSPADD	Synchronous Serial Port (I^2C mode) Address Register								0000 0000	73, 74
94h	SSPSTAT	SMP	CKE	D/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	UA	BF	0000 0000	66
95h	—	Unimplemented								—	—
96h	—	Unimplemented								—	—
97h	—	Unimplemented								—	—
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	96
99h	SPBRG	Baud Rate Generator Register								0000 0000	97
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	—	Unimplemented								—	—
9Dh	—	Unimplemented								—	—
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	116
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	0--- 0000	112

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.

3: These registers can be addressed from any bank.

4: PORTD, PORTE, TRISD, and TRISE are not physically implemented on PIC16F873/876 devices; read as '0'.

5: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:
Bank 2											
100h[3]	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
101h	TMR0	Timer0 Module Register								xxxx xxxx	47
102h[3]	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	26
103h[3]	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	18
104h[3]	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
105h	—	Unimplemented								—	—
106h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	31
107h	—	Unimplemented								—	—
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah[1,3]	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
10Bh[3]	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20
10Ch	EEDATA	EEPROM Data Register Low Byte								xxxx xxxx	41
10Dh	EEADR	EEPROM Address Register Low Byte								xxxx xxxx	41
10Eh	EEDATH	—	—	EEPROM Data Register High Byte						xxxx xxxx	41
10Fh	EEADRH	—	—	—	EEPROM Address Register High Byte					xxxx xxxx	41
Bank 3											
180h[3]	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	19
182h[3]	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26
183h[3]	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	18
184h[3]	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
185h	—	Unimplemented								—	—
186h	TRISB	PORTB Data Direction Register								1111 1111	31
187h	—	Unimplemented								—	—
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah[1,3]	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
18Bh[3]	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	41, 42
18Dh	EECON2	EEPROM Control Register2 (not a physical register)								---- ----	41
18Eh	—	Reserved maintain clear								0000 0000	—
18Fh	—	Reserved maintain clear								0000 0000	—

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.

3: These registers can be addressed from any bank.

4: PORTD, PORTE, TRISD, and TRISE are not physically implemented on PIC16F873/876 devices; read as '0'.

5: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

جامعة الزقازيق

كلية الهندسة

قسم هندسة الإتصالات و الإلكترونيات

تصميم المرسل الرقمى لقياس درجات الحرارة في المفاعلات بواسطة الميكروكنترولر

رسالة مقدمة للوفاء جزئيا بمتطلبات الحصول على

درجة الماجستير في الهندسة الكهربية

مقدمة من

م/ مطيعة عبد الحميد محمد نصار

تحت إشراف

ا.د./ محمد شاكر محمد اسماعيل

قسم الهندسة والأجهزة العلمية

مركز البحوث النووية -هيئة الطاقة الذرية

د./ عادل زغلول محمود

قسم هندسة الاتصالات والإلكترونيات

كلية الهندسة -جامعة الزقازيق

2011

ملخص البحث

مرسل الحرارة أحد أهم المرسلات فى المفاعل النووى . يستخدم فى تهيئة الإشارة لحساس الحرارة، يحتوى على مكبر لتكبير اشارته كما يعطى علاقة خطية للإشارة. هذا المرسل جزء من نظام للحصول على درجة الحرارة وعرضها رقميا ولكن هذا النظام مكلف جدا، ومعقد.

فى هذا العمل تم تصميم نظام رقمي بواسطة تقنية الميكروكنترولر واتخذت درجة الحرارة كمثال لعرضها رقميا وذلك لأهميتها .هذا النظام يكافىء النظام الموجود فى المفاعل النووى وبذلك يكون تم بناء نظام يعتمد على رقاقة واحدة و هذا النظام يعطى دقة و أداء يماثل الموجود فى المفاعل.كما أنه أقل تكلفة وأقل تعقيدا.

هناك أنواع كثيرة من المرسلات فى المفاعل النووى مثل مرسل الضغط ، مرسل التدفق ومرسل المستوى. تم اختيار مرسل الحرارة كمثال و ذلك لأن الحرارة واحدة من أهم المتغيرات فى عملية التحكم و القياسات الدقيقة للحرارة ليست عملية سهلة.

تم استخدام RTD كحساس للحرارة والتى تزداد مقاومتها مع زيادة درجة الحرارة و تستخدم عندما تكون الدقة مطلوبة وتعمل على أساس أن المقاومة الكهربية للمعادن تختلف باختلاف الحرارة.

تم استخدام (بيك ميكروكنترولر) كأساس فى هذا التصميم لمميزاته العديدة التى تناسب الكثير من التطبيقات، يحتوى على المحول من الإشارات المتماثلة إلى الرقمية والذاكرة الغير متطايرة والتى تعتبر أهم ما يميزه فهى سهلة البرمجة وسريعة، ومحتويات كثيرة فى داخله والتى أفادتنا فى هذا التصميم.

تم استخدام لغة الـ (سى) فى البرمجة،(البروتيس) فى تصميم الدائرة وعمل محاكاة لها وذلك لقوة بنائه والذى يسمح بإمكانية رسم دائرة متكاملة لتصميم نظام يعتمد على الميكروكنترولر ثم اختباره وقراءة درجة الحرارة وعرضها على شاشة عرض رقمية.

فى هذة الدائرة نقوم بعرض درجات الحرارة فى صورة رقمية على شاشة عرض رقمية، تخزين الحد الأقصى والحد الأدنى للحرارة فى الذاكرة الغير متطايرة مع وجود انذار اذا زادت أو قلت درجة الحرارة عن هذا الحد.

وبذلك يكون تم تقييم أداء نظام رقمى متكامل لعرض درجات الحرارة يعطى دقة عالية تماثل الموجودة فى المفاعل بتكلفة أقل بكثير و تقنية أسهل بكثير.

تحتوى الرسالة على خمسة فصول كالاتى :

الفصل الأول : يعرض مقدمة لموضوع الرسالة وأهميته وتسلسل فصولها.

الفصل الثاني : : يشمل وصف لنظام المفاعل المتعدد الأغراض وأنظمته الرئيسية الموجودة به و أجهزته والنظام الموجود به لقياس درجات الحرارة.

الفصل الثالث : يعرض مكونات النظام المقترح شاملا حساس الحرارة المستخدم ونظرية عمله و دوائره العملية بالإضافة إلى الميكروكنترولر المستخدم في الدائرة المقترحة بنائه و ذاكرته وسجلاته. ويشمل أيضا محول الإشارات المتماثلة إلى رقمية وكيفية ضبطه في الميكروكنترولر وبرمجته.

الفصل الرابع : يقدم تصميم النظام المقترح والنتائج العملية لهذا النظام مع وصف شامل للدائرة الإلكترونية وكيفية عملها في المحاكاة ومن ثم تنفيذها عمليا فضلا عن النتائج لمرسلات حرارة مختلفة المدى ومقارنة النتائج المأخوذة من الواقع العملي ونظيرتها من المحاكاة والدائرة العملية وتحليلها.

الفصل الخامس : يحتوى على تعليق على ما تم عرضه في الرسالة .