# “Designing a Microcontroller Based Temperature Data Logger”

*A thesis*

*Submitted towards the partial fulfillment of*

*requirements for the award of the degree of*

## Masters of Engineering

## (Electronic Instrumentation & Control)

*Submitted by*
**Deepika**

**Roll No. 8044209**

*Under the esteemed guidance of*

**Mr. Nirbhowjap Singh**
**Lecturer**
**EIED**

**Mr. Mandeep Singh**
**Lecturer**
**EIED**

**Department of Electrical and Instrumentation Engineering**
**Thapar Institute of Engineering & Technology**
**Deemed University, Patiala – 147004**
**June 2006**

# Declaration

I hereby certify that the work which is being presented in the thesis entitled, **"Designing a Microcontroller Based Temperature Data logger"** in partial fulfillment of the requirements for the award of degree of Master of Engineering in Electronics (Instrumentation and Control) Engineering at Electronics and Instrumentation Department of Thapar Institute of Engineering and Technology (Deemed University), Patiala, is an authentic record of my own work carried out under the supervision of Nirbhowjap Singh (Lecturer) and M.D Singh (Lecturer).

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other university.

**Deepika**

**Date: ---------------------** **Roll. No. 8044209**

It is certified that the above statement made by the student is correct to the best of my knowledge and belief.

(------------------------------)
**Mr. Nirbhowjap Singh**
**Lecturer, EIED**
**TIET, Patiala**

(------------------------------)
**Mr. Mandeep Singh**
**Lecturer, EIED**
**TIET, Patiala**

**(------------------------------)**
**Mrs. Manbir Kaur**
**Head EIED**
**TIET, Patiala**

**(------------------------------)**
**Dr. T.P. Singh**
**Dean of Academic Affairs**
**TIET, Patiala**

# Acknowledgement

The real spirit of achieving a goal is through the way of excellence and austerous discipline. I would have never succeeded in completing my task without the cooperation, encouragement and help provided to me by various personalities.

First of all, I render my gratitude to the ALMIGHTY who bestowed self-confidence, ability and strength in me to complete this work. Without his grace this would never come to be today’s reality.

With deep sense of gratitude I express my sincere thanks to my esteemed and worthy Supervisor Mr. Nirbhowjap Singh (Lecturer) and Co-Supervisor Mr. Mandeep Singh (Lecturer) Department of Electrical & Instrumentation Engineering, TIET (Deemed University) Patiala, for his valuable guidance in carrying out this work under their effective supervision, encouragement, enlightenment and cooperation. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. Their feedback and editorial comments were also invaluable for writing of this thesis.

I shall be failing in my duties if I do not express my deep sense of gratitude towards Mrs. Manbir Kaur, A.P. & Head of the Department of Electrical & Instrumentation Engineering, TIET (Deemed University) Patiala, who has been a constant source of inspiration for me throughout this work.

I am also thankful to all the staff members of the Department for their full cooperation and help.

I would also like to convey my gratitude to Mr. Rawel Singh (Lab Superintendent) without whose expert guidance and support, this job would have been compelling.

My greatest thanks are to all who wished me success especially my parents, my brother and sister whose support and care makes me stay on earth. I would also like to thank Abhishek, who helped me a lot in completing this work. And I also thank all those who helped me directly and indirectly in completion of my thesis.

# Table of contents

---

**Title** **Page No.**

# List of abbreviations

---

| | |
|---|---|
| ADC | Analog to Digital Converter |
| ALU | Arithmetic and logical unit |
| AT | Atmel |
| CLK | Clock |
| CPU | Central Processing Unit |
| EPROM | Erasable and programmable read only memory |
| EEPROM | Electrically erasable programmable read only Memory |
| GND | Ground |
| I/O | Input/Output |
| IC | Integrated circuit |
| LCD | Liquid Crystal Display |
| MCU | Microcontroller unit |
| MHZ | Megahertz |
| PCB | Printed Circuit Board |
| RAM | Random access memory |
| ROM | Read only memory |
| RTC | Real time clock |
| RTD | Resistive temperature device |

| | |
|---|---|
| XTAL | Crystal |

# List of Figures

# List of tables

---

# Abstract

With the advancement of technology, the processes are becoming more and more complex. Due to this increase in complexity, for efficient analysis of process the number of parameters required for data acquisition also increases. Data Acquisition is simply the gathering of information about a system or process. It is the process of collecting data in an automated fashion from analog and digital measurement sources such as sensors and devices under test. Before the computer age, most data was recorded manually or on strip-chart recorders. Many new generation data acquisition products have been developed due to emergence of microcontroller that enables real-time gathering, analysis, logging and viewing of data. To meet these requirements the demand of an improved, efficient and up-to-date data logger is increasing.

A data logger is an electronic instrument that can record digital or analog measurements over a period of time. It consists of a sensor, microcontroller, and a data storage device. Data loggers have an on-board memory that is large enough to hold data that is recorded over a longer period of time. The memory in these data loggers may be flash memory, EEPROM or Static Random Access Memory that is battery backed. Data loggers are provided with real time clocks to record the date and time of acquisition.

In this thesis, a data logger for specific application has been designed. The system works around the famous 8051 family. The system is designed and developed to measure the temperature with the help of temperature sensors and the result is stored in memory such as EEPROM for post process analysis. During the testing, it is verified that there is continuous and correct acquisition of data. It is also verified that the data is sequentially stored in memory. This verification is done by using an LCD display. The focus of design is on portability and low power consumption for battery operated applications. The designed system is tested under different conditions: at room temperatures, at low temperatures, at different temperatures, readings taken over long time. For all these conditions, the system performed accurately.

# Chapter 1: Introduction

The processes to collect, analyze and store the data for later use is called logging. It is a process to record events during a test or measurement with the use of a system or a product. The human brain and its memory, the nature's creation, no doubt is the best data logging mechanism. Where there is the need to collect information faster than a human, data loggers can possibly collect the information and in cases where accuracy is essential. A data logger is a device that can be used to store and retrieve the data [1]. Data logging also implies the control of how sensor collects and analyzes the data. It is commonly used in scientific experiments and in monitoring systems. Data loggers automatically make a record of the readings of the instruments located at different parts of plant. The type of information recorded is determined by the user. Their advantage is that they can operate independently of a computer and they are available in various shapes and sizes. The range includes simple economical single channel fixed function loggers to more powerful programmable devices capable of handling hundreds of inputs [2].

Temperature is the ever-changing parameter because of exposition to huge array of stimuli from their environment. It can be measured via a diverse array of sensors. All of them infer temperature by sensing some change in a physical characteristic. One must be careful when measuring temperature to ensure that the measuring instrument (thermometer, thermocouple, etc) is really the same temperature as the material that is being measured. Under some conditions heat from the measuring instrument can cause a temperature gradient, so the measured temperature is different from the actual temperature of the system. In such a case the measured temperature will vary not only with the temperature of the system, but also with the heat transfer properties of the system [3].

The objective of this work is to use data logging for temperature measurement. In order to meet the above requirements, a low cost, versatile, portable data logger is designed. A microcontroller based temperature data logger has been developed for measuring temperature at different input channels of ADC. The device is designed to receive data from temperature sensors and to store the results on external non-volatile electrically erasable programmable read only memory (EEPROM) for post process analysis. An integrated Liquid crystal display (LCD) is also used for real time display of data acquired from various sensors.

In the present work, temperatures indicated by three channels are compared with standard temperature obtained from clinical thermometer after the intervals of time. Based on these results, the accuracy of the channels is noted and by comparing them, the most accurate one among these is found.

# Chapter 2: Literature Review

This chapter describes the introduction to data loggers and literature survey.

## 2.1 Introduction to data loggers

The data logger is an invaluable tool to collect and analyze experimental data, having the ability to clearly present real time analysis with sensors and probes able to respond to parameters that are beyond the normal range available from the most traditional equipment [4]. The differences between various data loggers are based on the way that data is recorded and stored.

### 2.1.1 Definition of Data Loggers

Data logger is an electronic device that automatically records, scans and retrieves the data with high speed and greater efficiency during a test or measurement, at any part of the plant with time [4]. The type of information recorded is determined by the user i.e. whether temperature, relative humidity, light intensity, voltage, pressure or shock is to be recorded, therefore it can automatically measures electrical output from any type of transducer and log the value. A data logger works with sensors to convert physical phenomena and stimuli into electronic signals such as voltage or current. These electronic signals are then converted into binary data. The binary data is then easily analyzed by software and stored on memory for post process analysis.

### 2.1.2 Characteristics of Data Loggers:

Data loggers possess the following characteristics [5]:

1.) **Modularity:** Data loggers can be expanded simply and efficiently whenever required, without any interruption to the working system.
2.) **Reliability and Ruggedness:** They are designed to operate continuously without interruption even in the worst industrial environments.
3.) **Accuracy:** The specified accuracy is maintained throughout the period of use.
4.) **Management Tool:** They provide simple data acquisition, and present the results in handy form.

5.) **Easy to use:** These communicate with operators in a logical manner, are simple in concept, and therefore easy to understand, operate and expand.

### 2.1.3 Operation of data logger:

The ability to take sensor measurements and store the data for future use is, by definition, a characteristic of a data logger. However, a data-logging application rarely requires only data acquisition and storage. Inevitably, the ability to analyze and present the data to determine results and make decisions based on the logged data is needed. A complete data-logging application typically requires most of the elements illustrated below[5].

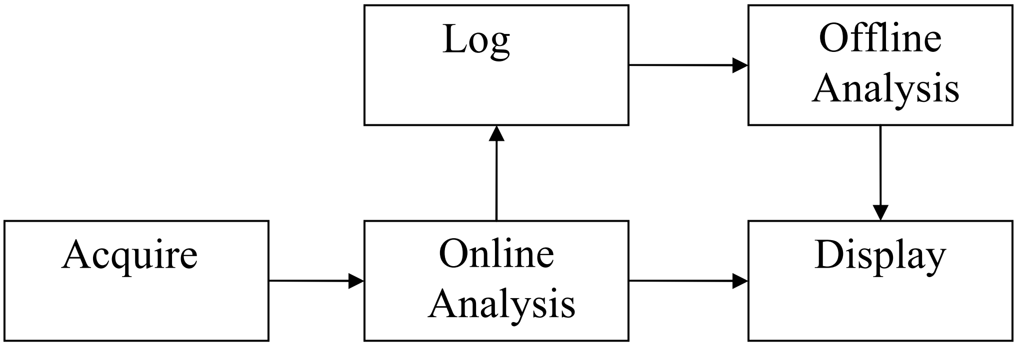


**Figure 2.1 Block diagram of data logger**

1.) **Acquire** – This step includes your sensors and data logger hardware as well as conversion of physical phenomena into digital signals.

2.) **Online analysis** – This step includes any analysis that is likely to be done before storing the data. A common example of this is converting the voltage measurement to meaningful scientific units, such as degree celcius. These complex calculations and data compression are completed before logging the data. Every data logging software application should complete this conversion from binary value to voltage and the conversion from voltage to scientific units.

3.) **Log** – This step refers to the storage of analyzed data including any formatting required for the data files.

4.) **Offline Analysis** - This step includes any analysis that is to be done after storing the data. A common example is looking for trends in historical data or data reduction.

5.) **Displaying, reporting** - This step includes the creation of any reports that are needed to make to present data and displaying the data. However, this can also present data straight from online analysis. This represents the ability to monitor and view the data as acquired and analyzed in addition to simply viewing historical data. i.e. it should have the following components:

- Hardware to digitize what is to be logged including sensors, signal conditioning, and analog-to-digital conversion hardware.
- Long-term data storage.
- Data-logging software for data acquisition, analysis, and presentation.

### 2.1.4 Advantages of Data Loggers:

1.) Data Loggers don't interfere with the users in performing their tasks [6].

2.) They can operate independently of a computer and they are available in various shapes and sizes.

3.) The range of data loggers varies from simple channel inputs to multichannel devices.

### 2.1.5 Applications of Data Loggers:

They can be used in the following applications such as:

1.) In unattended recording at weather stations to record parameters like temperature, wind speed / direction, solar radiation and relative humidity.

2.) For hydrographic recording of water flow, water pH, water conductivity, water level and water depth.

3.) In the recording of soil moisture levels.

4.) To record gas pressure and to monitor tank levels.

5.) In transportation monitoring, troubleshooting, educational science, quality studies, field studies and general research.

6.) Remote collection of recorded data and alarming or unusual parameters are possible with the help of data loggers where these are connected to modems and cellular phones.

## 2.2 Literature survey

**Dr. Saul Greenburg [1]** has described the concept of logging and how logging is done is in detail. Logging is a process to record events with the use of data loggers during a test or field use of a system or a product. Logging is one of the usability methods that can and should be used to gather more supplementary information as an integral part of the iterative design of the usability engineering cycle. Logging has the major advantage compared with other usability methods of not interfering with the users in their performing their tasks. Users can basically ignore the log and use the system in exactly the way they would anyway.

**AndrewJThompson, JohnLBahr and NeilRThomson [2]** have described low cost, very low power consumption, self-contained; digital data logger capable of independent operation for long periods of time has been developed. Its primary purpose was for temperature measurement (0.1o resolution), but any voltage in the range 0-5V can be measured. The logger can record up to 8 channels of 10-bit data simultaneously and stores data in 512Kbyte of on-board memory.

**H S kalsi [5]** has detailed the concept of data loggers and its basic operation is described. A data logger is a comprehensive and highly advanced data acquisition sytem. It is made versatile and flexible, to render it suitable for widely varying applications, specific requirements being met simply by setting up a suitable program. It can measure electrical output from any type of transducer and log the value automatically.

**Peter Roberson [6**] has reviewed the use of data loggers. The development of the fundamental skills involved with setting up of experimental apparatus, presenting data, producing an interpreting graphs, means that, at this stage, the traditional approach to performance of experiments is still an integral component. The use of the computer and data logger can be seen as an added bonus to enhance the opportunities for the new ways to explore traditional themes or to perform experiments that were previously very difficult, time consuming, or dangerous. The data logger can be of enormous benefit to allow for improvements in time efficiency, clear presentation of data to allow easier analysis and interpretation, difficult data rapidly displayed to allow clear visual interpretation of relationship between variables

**Muhammad Ali Mazidi and Janice Gillispe Mazidi [7]** discussed the overview of 8051 microcontroller. Microcontrollers and microprocessors are widely used in embedded system products. An embedded product uses a microcontroller to do one task and one task only. In addition to the description of criteria for choosing a microcontroller, the interfacing with the real world devices such as LCDs, ADCs, sensors and keyboard is described in detail. Finally; they discussed the issue of interfacing external memories, both RAM and ROM.

**Matthew Chapman [10]** discussed about the Intel 8051 microcontroller and its large family of descendants. The 8051 family of micro controllers is based on an architecture which is highly optimized for embedded control systems. One 8051 processor cycle consists of twelve oscillator periods. Each of the twelve oscillator periods is used for a special function by the 8051 core such as op code fetches and samples of the interrupt daisy chain for pending interrupts. An overview of 8051, its memory organization, addressing modes etc. are described in detail.

**Craig Steiner [12]** discussed about the 8051 family of microcontrollers. In addition to the types of memory, special function registers, basic registers, basic registers, addressing modes discussed in this tutorial additional features including introduction to 8052 and timers are also described.

**Craig Steiner [14]** gave a tutorial on LCD programming**.** This tutorial has presented the underlying concepts of programming an LCD display. A detailed description of Control and data signals of LCD is provided. The 44780 LCD offers many other functions which are accessed using other commands. Subroutines for initializing, for giving command and data to the LCD is discussed. Thus, it provides information from initializing to displaying the data.

**Craig Steiner [16]** gave a tutorial on real time clock. It provides overview of real time clock. Real-Time-Clock (RTC) is, as the name suggests, a clock which keeps track of time in a "real mode." While there are a number of 8051-compatible microcontrollers that have built-in, accurate real-time clocks (especially from Dallas Semiconductor), some simple applications may benefit from a software RTC solution that uses the built-in capabilities of an 8051 microcontroller. The drawbacks of using it and its solution is also discussed in this tutorial.

***Judy Ritchie [18]*** *discussed a comparison of data loggers. A guide to data logging is provided. A data logger is an electronic instrument that records measurements (temperature, relative humidity, light intensity, on/off, open/closed, voltage, pressure and events) over time. Typically, data loggers are small, battery-powered devices that are equipped with a microprocessor, data storage and sensor. How does a data logger work, where are they used, who uses them, what to look in for a data logger, why to choose data logger over a data acquisition system and their applications are discussed in this article.*

**S.J.Perez, M.A.Calva, R.Castañeda [19]** described a microcontroller-based data logging system to record temperature and relative humidity for acoustic measurement applications. The system is simple to use, requires no additional hardware and allows the selection of amount of data and the time intervals between them. The collected data can easily be exported to a PC computer via a serial port.

**Rajesh Luharuka, Robert X. Gao, Sundar Krishnamurty [20**] have discussed a microcontroller-based portable GSR data logger for physiological sensing. The device is configured to receive skin conductance data from a commercial GSR instrument, store them on its on-board memory, and relay them to a computer via the RS-232 serial port. The focus of the design is on portability and low power consumption for battery-driven ambulatory applications. A PIC microcontroller was used as the central control unit for the data flow coordination. The data logger prototype implemented using conventional ICs is small enough for physiological sensing in the field.

For the design and development of the system, the methodology used involves the software and hardware implementation. The actual implementation of the system involves the following steps:

1.) **System Definition**: Broad definition of system hardware including microcontroller and its interface with display, ADC, memory, keypad etc.

2.) **Circuit Design**: Selection of 8051 microcontroller and other interfacing devices, as per system definition. Design of hardware circuit and its testing on laboratory kits with some simple microcontroller software routines.

3.) **PCB Design and Fabrication**: Generation of schematic diagrams and the production of circuit board layout data for the procurement of the circuit board.

4.) **Hardware Modifications**: Making any hardware changes found necessary after the initial hardware tests, to produce a revised circuit board schematic diagram and layout.

5.) **Software Design**: Developing algorithm for the system, allocating memory blocks as per functionality, coding and testing.

6.) **Integration and Final Testing:** Integrating the entire hardware and software modules and its final testing for data logging operation.

Thus the complete design is divided into two parts:

1.) Hardware Implementation.
2.) Software Implementation.

## 3.1 Hardware Implementation

It involves the details of the set of design specifications. The hardware design consists of, the selection of system components as per the requirement, the details of sub-systems that are required for the complete implementation of the system and full hardware schematics for the PCB layout. Design of the circuit and its testing has been carried out. It involves the component selection, component description and hardware details of the system designed.

1.) Component selection and description.
2.) Hardware details of the system designed.

## 3.1.1 Component selection and description

Temperature measurement using microcontroller based data logger includes the following components:

1.) Temperature Sensor (LM35)
2.) Analog to Digital Converter(ADC 0808)
3.) Microcontroller(AT89C51)
4.) Keypad Matrix(4×4 matrix)
5.) Liquid Crystal Display(HD44780)
6.) External Memory(AT28C64)

### 1.) Temperature Sensor

- **Selection of Suitable Transducer:**

For measuring the temperature, the choice of sensor is of utmost importance [7]. The sensors are used in many fields includes Thermocouples, Resistive temperature devices (RTDs and thermistors) and bimetallic devices. The factors for the selection of sensor that we take into account includes the inherent accuracy for durability, range of operation, susceptibility to external noise influences, ease of maintenance and installation, handling during installation (delicacy), ease of calibration, and type of environment it will be used in.

The temperature sensor used for this purpose is LM35 because of the following features:

- **Features of LM35:**

1.) Calibrated directly in ° Celsius (Centigrade).
2.) Linear + 10.0 mV/°C scale factor.

3.) 0.5°C accuracy guaranteeable (at +25°C).

4.) Rated for full −55° to +150°C range.

5.) Suitable for remote applications.

6.) Low cost due to wafer-level trimming.

7.) Operates from 4 to 30 volts.

8.) Less than 60 µA current drain.

9.) Low self-heating, 0.08°C in still air.

10.) Nonlinearity only ±1⁄4°C typical.

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature [8]. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C over a full −55 to +150°C temperature range. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies.

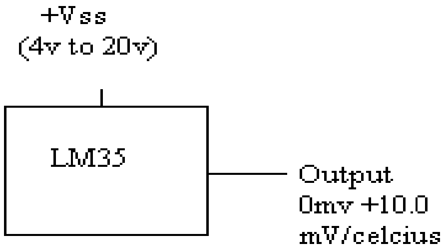


**Figure 2.2 LM35 Temperature Sensor**

## 2.) Analog to Digital Converter (ADC)

In physical world parameters such as temperature, pressure, humidity, and velocity are analog signals. A physical quantity is converted into electrical signals. We need an analog to digital converter to translate the analog signals to digital numbers so that the microcontroller can read them. Thus, an analog-to-digital converter (ADC) is an electronic circuit that converts continuous signals to discrete digital numbers. Analog to digital converters are the most widely used devices for data acquisition [9].

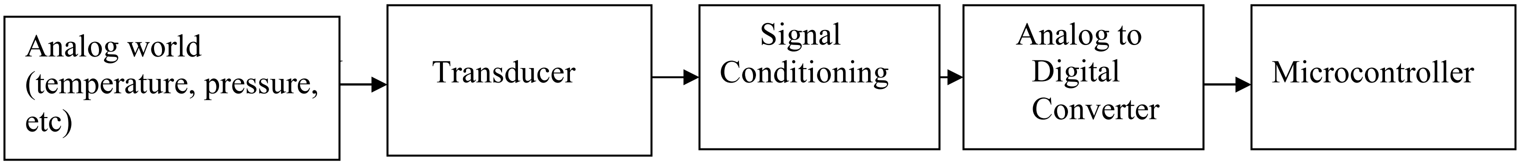


**Figure 2.3 Getting data from the analog world**

- **How to select an Analog Channel**

How to select the channel using three address pins A, B, C is shown in Table 2.1.

| **Select Analog Channel** | **C** | **B** | **A** |
|---|---|---|---|
| IN0 | 0 | 0 | 0 |
| IN1 | 0 | 0 | 1 |
| IN2 | 0 | 1 | 0 |
| IN3 | 0 | 1 | 1 |
| IN4 | 1 | 0 | 0 |
| IN6 | 1 | 1 | 0 |
| IN7 | 1 | 1 | 1 |

**Table 2.1 How to select an Analog Channel**

The ADC 0804 is most widely used chip. For this work, the analog to digital converter chosen is ADC 0809. Because ADC 0804 has only one analog input, this chip has 8 of the analog inputs. This chip allows us to monitor up to 8 different transducers using only a single chip. The 8 analog input channels are multiplexed and selected according to the requirement. In ADC 0808/0809, $V_{ref\ (+)}$ and $V_{ref\ (-)}$ set the reference voltage. If $V_{ref(-)}$ =Gnd and $V_{ref(+)}$ =5V, the step size is 5V/256=19.53 .Therefore to get a 10 mV step size we need to set $V_{ref(+)}$ =2.56 V and $V_{ref(-)}$ =Gnd.

- **Pin diagram of ADC 0808/0809**

We use A, B, C addresses to select IN0-IN7 and activate Address latch enable (ALE) to latch in the address. SC is for Start Conversion. EOC is for End of Conversion and OE is for Output Enable. The output pins D0-D7 provides the digital output from the chip.$V_{ref(-)}$ and $V_{ref(+)}$ are the reference voltages. Since there is no self clocking in this chip so the clock must be provided from an external source to the Clock (CLK) pin.

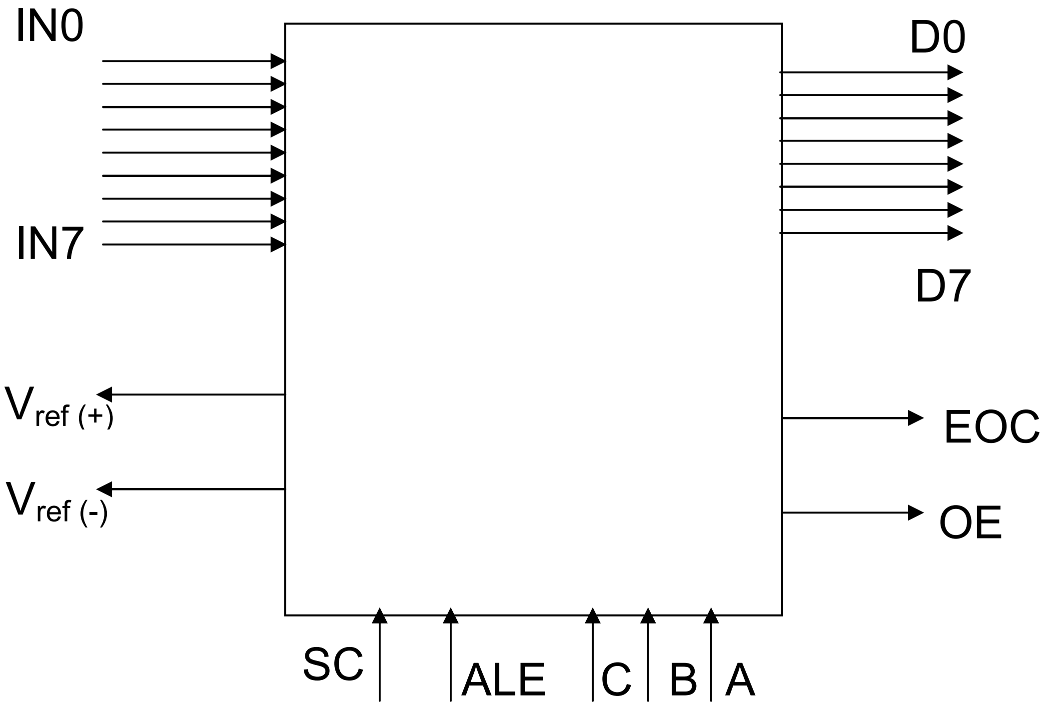


**Figure 2.4 Pin Diagram of ADC 0808/0809**

- **Steps to program the ADC 0808/0809**

The following are the steps to get data from analog input of ADC 0808/0809 into the microcontroller [7]:

1.) Select an analog channel by providing bits to A, B, C addresses.

2.) Activate the ALE (address latch enable) pin. It needs an L-to-H pulse to latch in the address.

3.) Activate SC (start conversion) by an H-TO-L pulse to initiate conversion.

4.) Monitor EOC (end of conversion) to see whether conversion is finished H-to-L output indicates that the data is converted and is ready to be picked up.

5.) Activate OE (output enable) to read data out of the ADC chip. An H-to-L pulse to the OE pin will bring digital data out of the chip.

In ADC 0808/0809 there is no self clocking and the clock must be provided from an external source to the CLK pin.

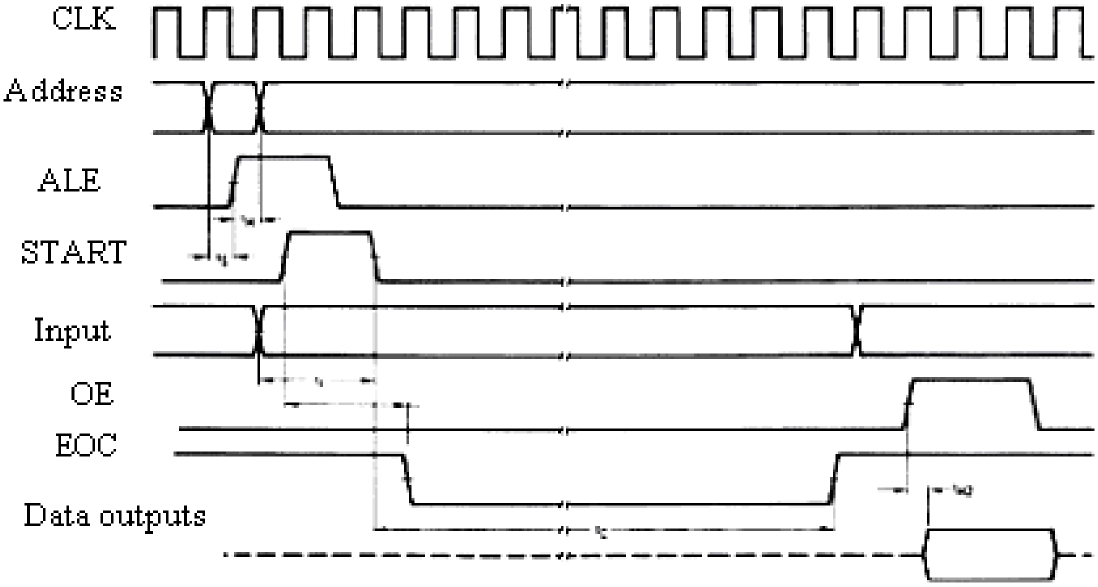


**Figure 2.5 Timing diagram of ADC 0808/0809**

There are 8 clock periods per approximation. Even though there is no conversion in progress the ADC0808/ADC0809 is still internally cycling through these 8 clock periods [9]. A start pulse can occur any time during this cycle but the conversion will not actually begin until the converter internally cycles to the beginning of the next 8 clock period sequence. As long as the start pin is held high no conversion begins, but when the start pin is taken low the conversion will start within 8 clock periods. The EOC output is triggered on the rising edge of the start pulse. It, too, is controlled by the 8 clock period cycle, so it will go low within 8 clock periods of the rising edge of the start pulse. One can see that it is entirely possible for EOC to go low before the conversion starts internally, but this is not important, since the positive transition of EOC, which occurs at the end of a conversion, is what the control logic is looking for. Once EOC does go high this signals the interface logic that the data resulting from the conversion is ready to be read. The output enable (OE) is then raised high.

## 3.) Microcontroller chip

### ➢ Criteria for choosing a microcontroller

1.) The first and foremost criterion for choosing a microcontroller is that it must meet the task at hand efficiently and cost effectively [7]. In analyzing the needs of a microcontroller-based project, it is seen whether an 8-bit, 16-bit or 32-bit microcontroller can best handle the computing needs of the task most effectively. Among the other considerations in this category are:

(a) Speed – What is the highest speed that the microcontroller supports?
(b) Packaging – Does it come in 40-pin DIP (dual inline package) or a QFP (quad flat package), or some other packaging format? This is important in terms of space, assembling, and prototyping the end product.
(c) Power consumption – This is especially critical for battery-powered products.
(d) The number of I/O pins and the timer on the chip.
(f) How easy it is to upgrade to higher –performance or lower consumption versions.
(g) Cost per unit – this is important in terms of the final cost of the product in which a microcontroller is used.

2.) The second criterion in choosing a microcontroller is how easy it is to develop products around it. Key considerations include the availability of an assembler, debugger, a code –efficient compiler, technical support.

3.) The third criterion in choosing a microcontroller is its ready availability in needed quantities both now and in the future. Currently of the leading 8-bit microcontrollers, the 8051 family has the largest number of diversified suppliers. By supplier is meant a producer besides the originator of the microcontroller. In the case of the 8051, this has originated by Intel several companies also currently producing the 8051.

Thus the microcontroller AT89C51 satisfying this entire criterion is chosen for this work.

- **AT89C51 Microcontroller**

The 8051 family of microcontrollers is based on an architecture which is highly optimized for embedded control systems. It is used in a wide variety of applications from military equipment to automobiles to the keyboard. Second only to the Motorola 68HC11 in eight bit processors sales, the 8051 family of microcontrollers is available in a wide array of variations from manufacturers such as Intel, Philips, and Siemens. These manufacturers have added numerous features and peripherals to the 8051 such as I2C interfaces, analog to digital converters, watchdog timers, and pulse width modulated outputs. Variations of the 8051 with clock speeds up to 40MHz and voltage requirements down to 1.5 volts are available. This wide range of parts based on one core makes the 8051 family an excellent choice as the base architecture for a company's entire line of products since it can perform many functions and developers will only have to learn this one platform [10].

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer, which provides a highly-flexible and cost-effective solution to many embedded control applications [11].
The basic architecture of AT89C51 consists of the following features:

1.) An eight bit ALU
2.) 32 discrete I/O pins (4 groups of 8) which can be individually accessed
3.) Two 16 bit timer/counters
4.) Full duplex UART
5.) 6 interrupt sources with 2 priority levels
6.) 128 bytes of on board RAM
7.) Separate 64K byte address spaces for DATA and CODE memory

One 8051 processor cycle consists of twelve oscillator periods. Each of the twelve oscillator periods is used for a special function by the 8051 core. The time required for any 8051 instruction can be computed by dividing the clock frequency by 12, inverting that result and multiplying it by the number of processor cycles required by the instruction in question [12]. Therefore, if you have a system which is using an 11.059MHz clock, you can compute the number of instructions per second by dividing this value by 12. This gives an instruction frequency of 921583 instructions per second. Inverting this will provide the amount of time taken by each instruction cycle (1.085 microseconds).

## 4.) Keypad Matrix

At the lowest levels, keyboards are organized in a matrix of rows and columns. The CPU accesses both the rows and columns through ports, therefore, with two 8-bit ports; an 8×8 matrix of keys can be connected to the microcontroller. When a key is pressed, a row and a column make a contact, otherwise, there is no connection between rows and columns [7].The detection of the key varies from scanning of the key to grounding the rows and columns.

### ➢ Scanning and identifying the key

A 4×4 matrix connected to two ports. The rows are connected to an output port and the columns are connected to an input port. If no key has been pressed, reading the input port will yield 1s for all the columns since they are connected to high($V_{CC}$).if all the rows are grounded and a key is pressed, one of all the columns will have 0 since the key pressed provides the path to ground. It is the function of the microcontroller to scan the keyboard continuously to detect and identify the key pressed.

### ➢ Grounding the rows and columns

To detect a pressed key, the microcontroller grounds all rows by providing 0 to the output latch, and then read its columns. If the data read from the columns is D3-D0 =1111, no key is pressed and the process continues until a key is press is detected.

However, if one of the column bits has a zero, this means that a key in D1 column has been pressed. After a key pressed is detected, the microcontroller will go through the process of identifying the key. Starting with the top row; the microcontroller grounds it by providing a low to row D0 only; then it reads the columns. If the data read is all 1s, no key in that row is activated and the process is moved to the next row. It grounds the next row, reads the columns, and checks for any zero. This process continues until the row is identified. After identification of the row in which the key is pressed, the next task is to find out which column the pressed key belongs to.

- **Stages for the detection and identification of key activation**

1.) To make sure that the preceding key has been released, 0s are output to all rows at once, and the columns are read and checked repeatedly until all the columns are high. When all columns are found to be high, the program waits for a short amount of time before it goes to the next stage of waiting for a key to pressed [7].

2.) To see if any key is pressed, the columns are scanned over and over in an infinite loop until one of them has a 0 on it. After the key press detection, it waits 20 ms for the bounce and then scans the columns again. This serves two functions:

(i) It ensures that the first key press detection was not an erroneous one due to a spike noise, and

(ii) The 20- ms delay prevents the same key press from being interpreted as a multiple key press. If after the 20-ms delay the key is still unpressed, it goes to the next signal to detect which row it belongs to, otherwise it goes back into the loop to detect a real key press.

3.) To detect which row the key press belongs to, it grounds one row at a time, reading the columns each time. If it finds that all columns are high, this means that the key press cannot belong to that row, therefore, it grounds the next row and continues until it finds the row the key press belongs to. Upon finding the row that the keypress belongs to, it sets up the starting address for the look-up table holding the scan codes for that row and goes to the next stage to identify the key.

4.) To identify the key press, it rotates the column bits, one bit at a time, into the carry flag and checks to see if it is low. Upon finding the zero, it pulls out the ASCII code for that key from the look-up table, otherwise it increments the pointer to point to the next element of the look-up table.

## 5.) Liquid Crystal Display

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. It uses very small amounts of electric power, and is therefore suitable for use in battery-powered electronic devices. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other [13].

More microcontroller devices are using 'smart LCD' displays to output visual information. LCD displays designed around Hitachi's LCD HD44780 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 8x80 pixels of the display. Hitachi LCD displays have a standard ASCII set of characters plus Japanese, Greek and mathematical symbols.

For an 8-bit data bus, the display requires a +5V supply plus 11 I/O lines. For a 4-bit data bus it only requires the supply lines plus seven extra lines. When the LCD display is not enabled, data lines are tri-state which means they are in a state of high impendence (as though they are disconnected) and this means they do not interfere with the operation of the microcontroller when the display is not being addressed.

In LCD we can put data at any location. For 16×2 LCD, the address locations are:

**Table 2.2 Cursor addresses for LCD**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| First line | 80 | 81 | 82 | 83 | 84 | 85 | 86 | through | 8F |
| Second line | C0 | C1 | C2 | C3 | C4 | C6 | C7 | through | CF |

- **Signals to the LCD**

The LCD also requires 3 control lines from the microcontroller:

**Enable (E)**

This line allows access to the display through R/W and RS lines. When this line is low, the LCD is disabled and ignores signals from R/W and RS. When (E) line is high, the LCD checks the state of the two control lines and responds accordingly [14].

**Read/Write (R/W)**

This line determines the direction of data between the LCD and microcontroller. When it is low, data is written to the LCD. When it is high, data is read from the LCD.

**Register select (RS)**

With the help of this line, the LCD interprets the type of data on data lines. When it is low, an instruction is being written to the LCD. When it is high, a character is being written to the LCD.

Logic status on control lines:

**E** - 0 Access to LCD disabled
-1 Access to LCD enabled
**R/W** -0 Writing data to LCD
-1 Reading data from LCD
**RS** -0 Instruction
-1 Character

- **Writing and reading the data from the LCD**

Writing data to the LCD is done in several steps:

1.) Set R/W bit to low
2.) Set RS bit to logic 0 or 1 (instruction or character)
3.) Set data to data lines (if it is writing)
4.) Set E line to high
5.) Set E line to low

Read data from data lines (if it is reading)

1.) Set R/W bit to high
2.) Set RS bit to logic 0 or 1 (instruction or character)
3.) Set data to data lines (if it is writing)
4.) Set E line to high
5.) Set E line to low

## 6.) External Memory

The microcontroller AT89C51 that has been used in designing a data logger has internal memory of 8k. In the design of data logger, the need is to write in the memory that is why using the external memory is significant as it is not possible to write data in internal memory of microcontroller. For this application, the contents of memory are needed to be written and erased a number of times and for this reason electrically–erasable–and programmable memory (EEPROM) is to be used. Internally EEPROM is similar to erasable programmable read only memory (EPROM).AT28C64 has been used since it has large memory i.e.64k. The AT28C64 is accessed like a Static RAM for the read or write cycles without the need for external components [15]. During a byte write, the address and data are latched internally, freeing the microprocessor address and data bus for other operations. Following the initiation of a write cycle, the device will go to busy state and automatically clear and write the latched data using an internal control timer.

- **Features of AT28C64:**

1.) Fast Read Access Time – 120 ns
2.) Fast Byte Write – 200 µs or 1 ms
3.) Self-timed Byte Write Cycle
   – Internal Address and Data Latches
   – Internal Control Timer
   – Automatic Clear before Write
4.) Direct Microprocessor Control
   – READY/BUSY Open Drain Output
   – DATA Polling

5.) Low Power

– 30 mA Active Current

– 100 µA CMOS Standby Current

6.) High Reliability

– Endurance: 104 or 105 Cycles

– Data Retention: 10 Years

7.) 5V ± 10% Supply

➢ **Control Operations**

**Read:** The AT28C64 is accessed like a Static RAM. When CE and OE are low and WE is high, the data stored at the memory location determined by the address pins is asserted on the outputs. The outputs are put in a high impedance state whenever CE or OE is high.

**Byte Write:** Writing data into the AT28C64 is similar to writing into a Static RAM. A low pulse on the WE or CE input with OE high and CE or WE low (respectively) initiates a byte write. The address location is latched on the falling edge of WE (or CE); the new data is latched on the rising edge. Internally, the device performs a self-clear before write. Once a byte write has been started, it will automatically time itself to completion.

**Ready/busy:** Pin 1 is an open drain RDY/BUSY output that can be used to detect the end of a write cycle. RDY/BUSY is actively pulled low during the write cycle and is released at the completion of the write. The open drain connection allows for OR-tying of several devices to the same RDY/BUSY line. The RDY/BUSY pin is not connected for the AT28C64X.

**Data polling:** The AT28C64 provides DATA polling to signal the completion of a write cycle. During a write cycle, an attempted read of the data being written results in the complement of that data for I/O7 (the other outputs are indeterminate). When the write cycle is finished, true data appears on all outputs.

**Write Protection**: Inadvertent writes to the device are protected against in the following ways:

1.) $V_{CC}$ sense –If $V_{CC}$ is below 3.8V (typical) the write function is inhibited.
2.) $V_{CC}$ power on delay – Once $V_{CC}$ has reached 3.8V the device will automatically time out 5 ms (typical) before allowing a byte write.
3.) Write inhibit – Holding any one of OE low, CE high or WE high inhibits byte write cycles.

**Chip Clear**: The contents of the entire memory of the AT28C64 may be set to the high state by the CHIP CLEAR operation. By setting CE low and OE to 12 volts, the chip is cleared when a 10 msec low pulse is applied to WE.

**Device Identification**: An extra 32 bytes of EEPROM memory are available to the user for device identification. By raising A9 to 12 ± 0.5V and using address locations 1FE0H to 1FFFH the additional bytes may be written to or read from in the same manner as the regular memory array.

## 3.1.2 Hardware details of system designed

Figure 3.1 shows the schematic for the implementation of the hardware for this purpose.

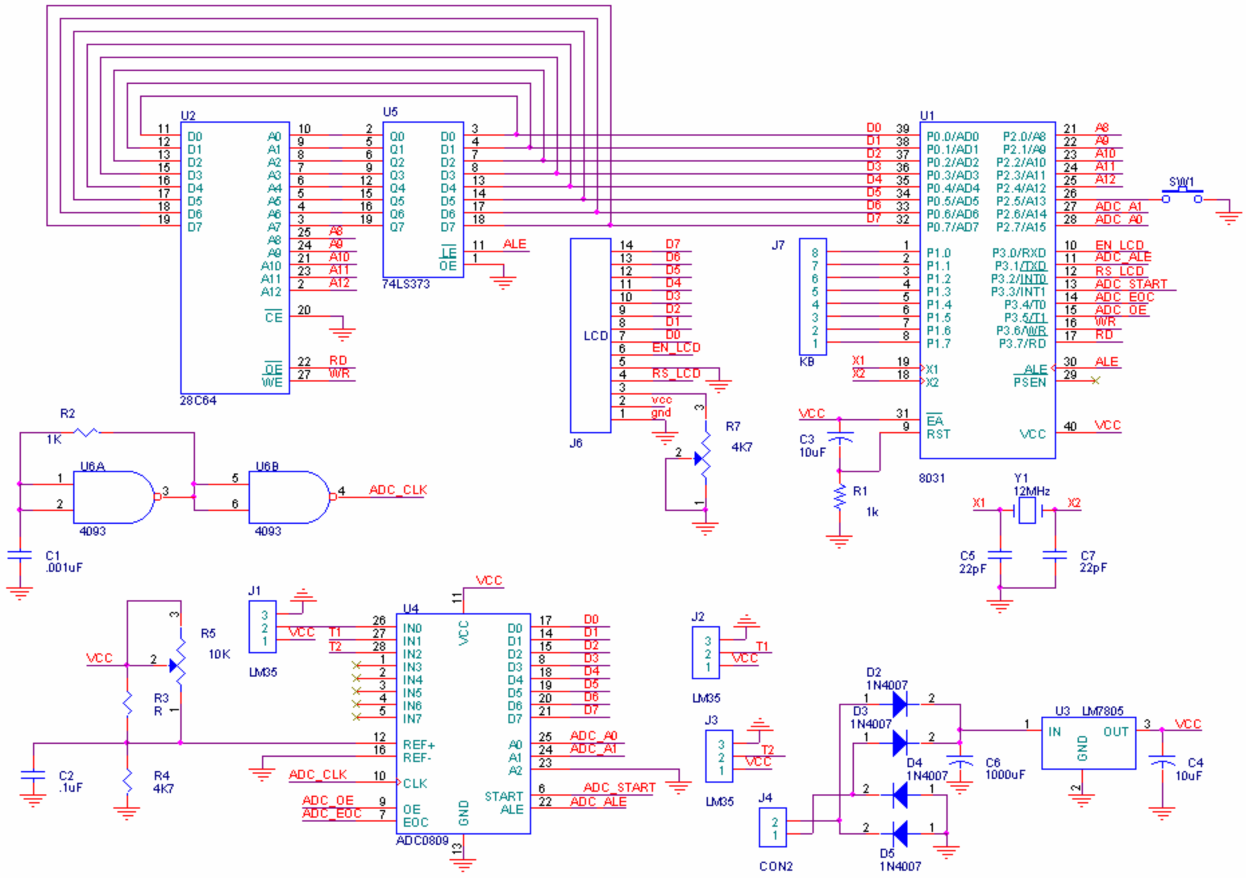


Figure 3.1 Schematic diagram

Figure 3.2 shows the block diagram of the system hardware

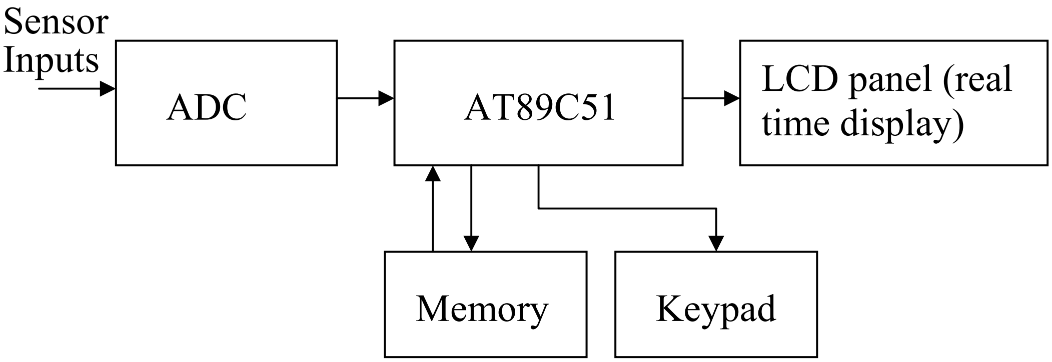


Figure 3.2 Block diagram of system hardware

The details of the circuit component connections are as given below:

- **Microcontroller AT89C51**

The four I/O ports of the microcontroller are used for interfacing the external peripherals. 8 bits of port 0 are interfaced to external memory AT28C64. The memory chip AT28C64 is interfaced to the microcontroller through the latch 74LS373. 8 bits of port 1 are connected to the keyboard. First 5 bits of port 2 is connected to address lines of memory. A pushbutton is connected to P2.5. P2.6 and P2.7 are connected to the address lines of ADC. P3.0 and P3.2 are connected to pin 6 and pin 4 of LCD. P3.3, P3.4and P3.5 are connected to pin 22, 7 and 9 of ADC.P3.6 and P3.7 are connected to 22 and 27 pin of memory. Pin 30 is connected to pin 11 of latch. Pin 18 and 19 are connected to a crystal. Pin 31 is connected to $V_{CC}$ and the Pin 9 is connected to ground via resistance of 1-kilo ohm. Pin 40 is connected to $V_{CC}$. Pin 30 is connected to pin 11 of Latch.

- **Memory AT28C64**

8 data lines of memory are connected to port 0 of the microcontroller chip. 8 address lines are connected to latch 74LS373. The remaining address lines are connected to P2.0-P2.4. Chip enable pin is connected to ground. Output enable pin is connected to P3.7 (read pin) and Write enable pin is connected to P3.6 (write pin) of the controller chip.

- **Latch 74LS 373**

It is used for the purpose of multiplexing the data. The 8 input pins are connected to address lines of memory. Data pins are connected port 0 of the controller chip. Latch enable pin of latch is connected to ALE pin of the microcontroller and the pin Output enable is connected to ground.

- **Hitachi HD44780 LCD**

The 8 data pins of LCD are connected to the 8-bits of the port 0 to send data to the LCD. The control signals of the LCD module RS and EN are connected to the pins P3.0 and P3.2.RW pin is connected to ground. Pin 1 is connected to Gnd and pin 2 to $V_{CC}$. Pin 3 is connected to ground via variable resistance that is used to adjust the contrast.

- **Keypad**

Keypad is connected to port 1 of the microcontroller chip.

- **LM35 temperature sensor**

First pin is connected to Vcc, Second pin to the input channel and the third pin is connected to ground. Output of first LM35 is connected to IN0, Output of second LM35 is connected to IN1 and Output of third LM35 is connected to IN2.

- **ADC 0808**

Pin 6 is connected to P3.3, Pin 9 to P3.5, Pin 7 to P3.4, and Pin 22 to P3.1 of the microcontroller chip. Schmitt trigger NAND gate is used to provide clock to the chip at pin 10.Pins 24, 25 are connected to P2.6 and P2.7 of microcontroller. The 8 data pins are connected to data pins of LCD. INO, IN1 and IN2 are connected to the outputs of the three LM35. IN3-IN7 are not connected. Pin 11 is connected to $V_{CC}$. Pin 13, Pin 16, Pin 23 are connected to ground. Pin 12 is connected to a variable voltage.

- **Power supply section**

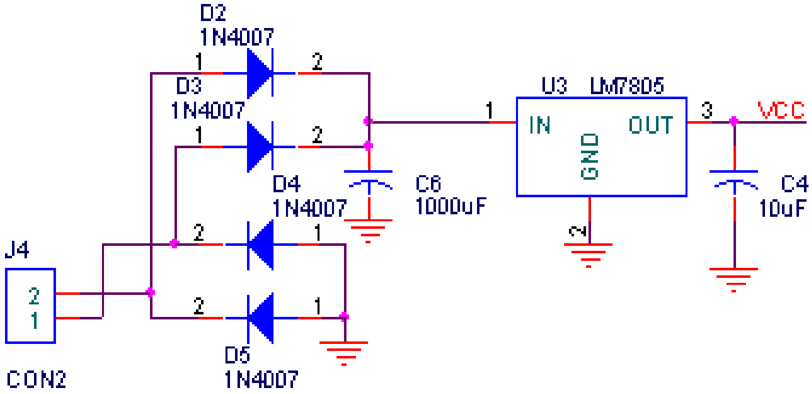


Figure 3.3 Power supply section

A power supply section is the regulated DC power supply of +5 Volts. +5 Volts is generated using LM7805 fixed voltage regulator. Rectification of the AC supply is carried out using 4 IN4007 diodes connected to pin 1 of LM7805.Pin 2 is connected to ground and Pin 3 is connected to $V_{CC}$. The output of this section is free from ripples and distortions.

## 3.2 Software design and development:

Software design includes developing algorithm for the system, allocating memory blocks as per functionality, writing the separate routines for different interfacing devices and testing them on the designed hardware. Interfacing of microcontroller with ADC, LCD, MEMORY, etc. has been carried out using various software modules. The control program is written in assembly language. The software is able to show the real time values from the analog channels for immediate analysis. For designing the software for this work; the flow of software between the hardware components is to be understood first.

### 3.2.1 Algorithm for the stepwise designing of the software for the system:

For the designing of algorithm of the system two steps are to be taken into account:

1.) Scanning the data.
2.) Offline analysis.

**1.) Scanning of data:** it involves the analysis that is done before storing the data. Algorithm for scanning of data is as:

1.) In this algorithm, first step is to initialize the LCD panel.
2.) The keyboard is checked for the value of hours, minutes and seconds and its value is displayed on the LCD.
3.) The time displayed will then be checked for time corresponding to the channel.
4.) The address of that channel is sent to the analog to digital converter.
5.) The value obtained is then converted to appropriate form for display.
6.) This value is then stored in memory. The loop will repeat itself until all the values are stored in the memory.

Flowchart for scanning of data is shown in figure 3.4

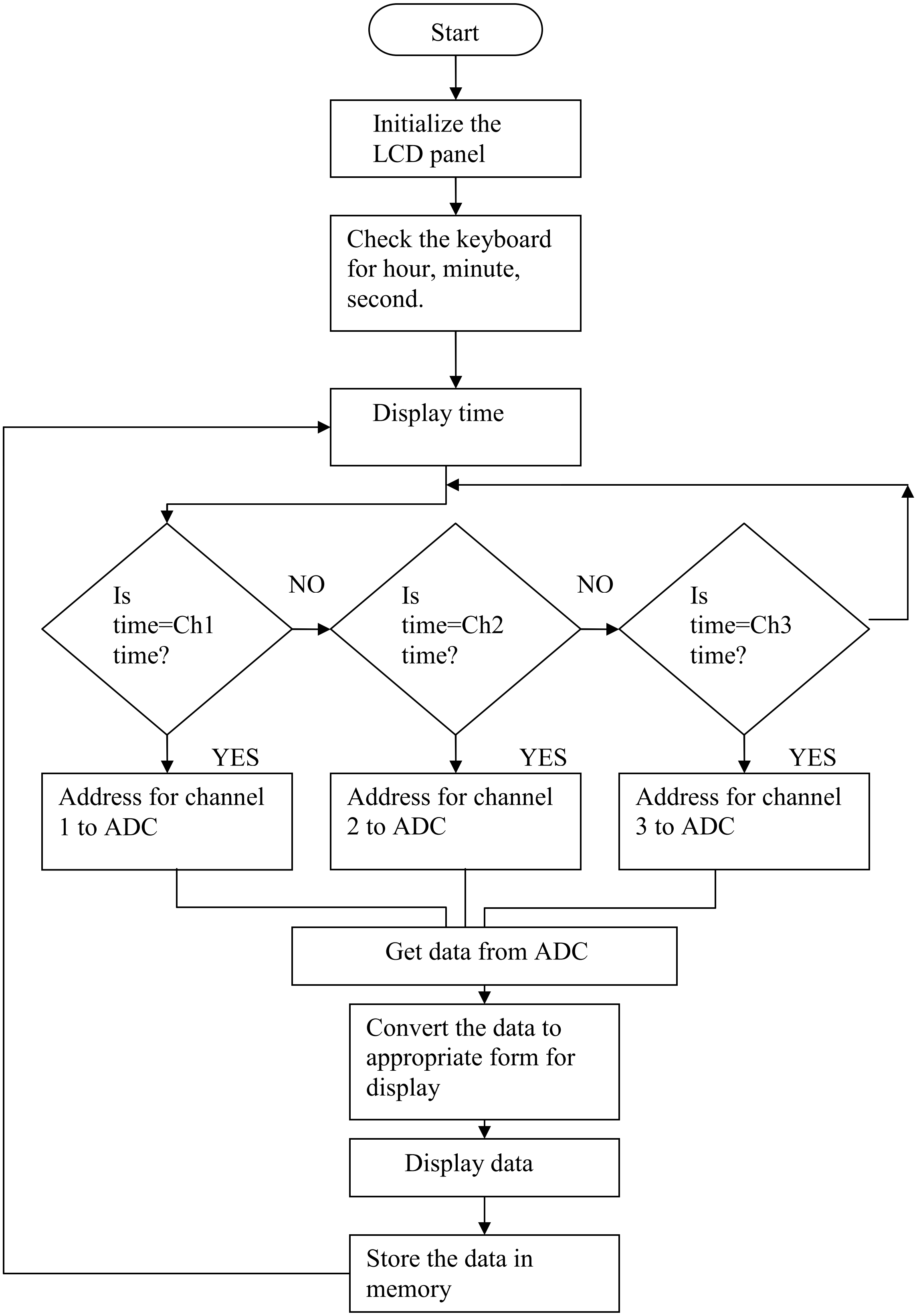


**Figure 3.4 Flowchart showing Scan mode of the system**

**2.) Offline analysis:** It involves the analysis that is to be done after storing the data that is to view the values of data stored in the memory.

Flowchart for offline analysis of data is shown in figure 3.5

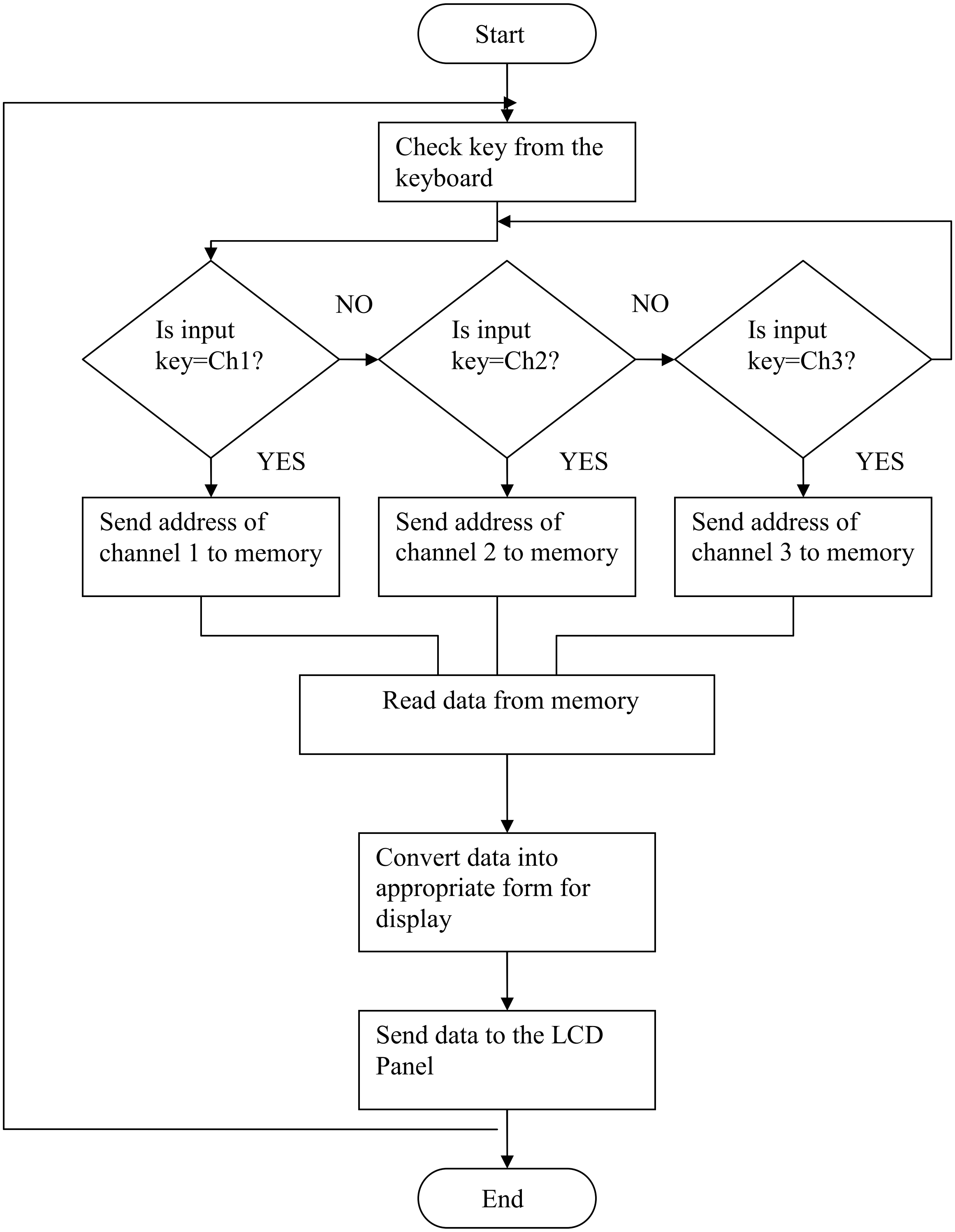


**Figure3.5 Flowchart for offline analysis of the system**

Algorithm for offline analysis of data is as:

1.) In this algorithm first step is to check the input key from the keyboard.

2.) It will check whether the input key is of channel1 (Ch1), channel2 (Ch2) or channel3 (Ch3).

3.) The address corresponding to that channel is sent to the memory.

4.) The data stored at that memory location is read and is converted to appropriate form for display.

5.) The value is then displayed on the LCD panel.

6.) The loop will repeat itself until all the values are read from the memory.

Software algorithms for controlling the various components of the hardware are:

### 3.2.2 Interfacing ADC0808 with the microcontroller

Algorithm for interfacing ADC 0808 with microcontroller is as:

1.) Firstly, the address lines ‘adc_adr0’ and ‘adc_adr1’ are selected.

2.) The start of conversion line ‘adc_start’ is set and then cleared.

3.) The end of conversion line ‘adc_eoc’ is then checked. If it satisfies the condition ‘adc_eoc’=0, the output enable line ‘adc_oe’ is set; otherwise it will execute the loop until the condition is satisfied.

4.) In the next step, the contents of port 0 are moved to the acquired channel

5.) Finally the output enable line ‘adc_oe’ is cleared. And the loop is executed till all the channels are selected.

Flowchart for interfacing ADC0808 with the microcontroller is shown in figure 3.6:

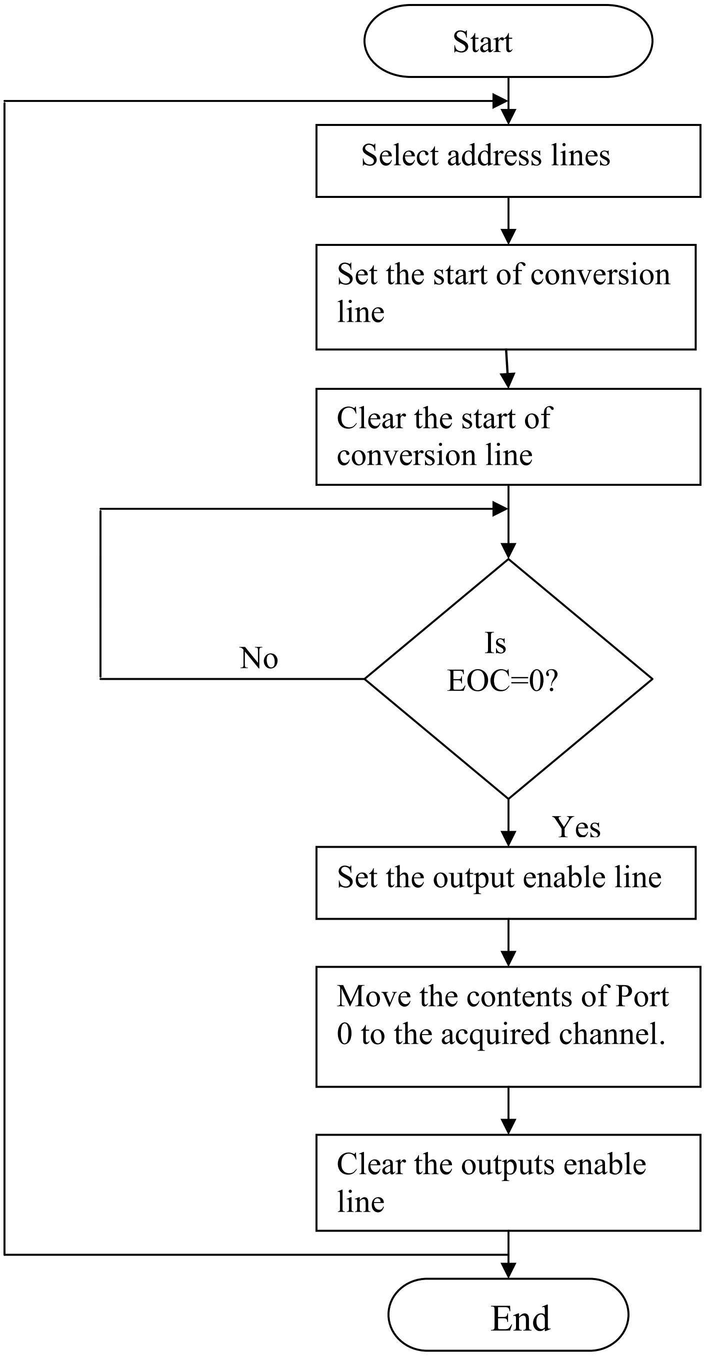


**Figure 3.6 Flowchart for interfacing ADC with microcontroller**

### 3.2.3 Interfacing keypad to the microcontroller

Flowchart for the keypad interfacing is shown in figure 3.7

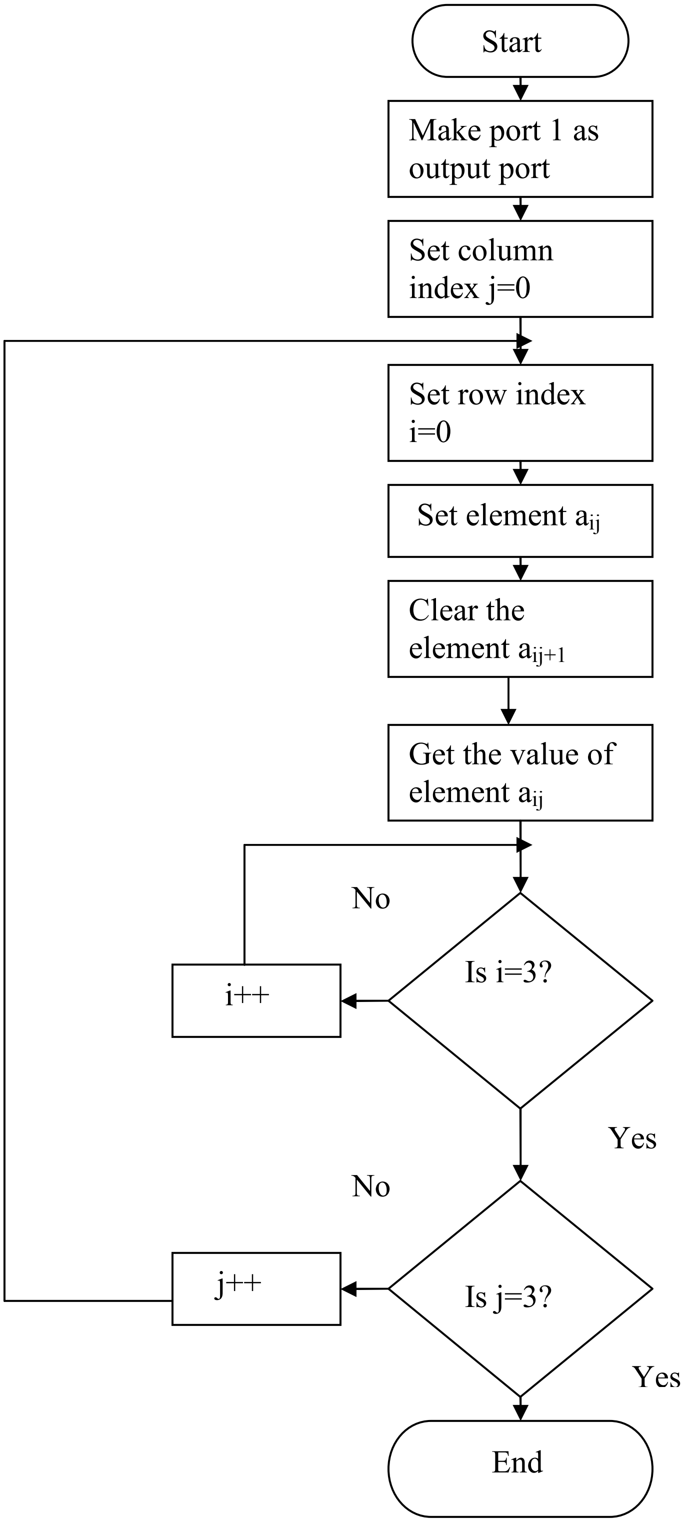


Figure 3.7 Flowchart for interfacing keypad to the microcontroller

Algorithm for interfacing keypad with the microcontroller is as:

1.) In this i is considered to be index of rows and j to be index of columns.
2.) Firstly, port 1 is made the output port.
3.) Set the column index row index equal to 0.
4.) Set the row index row index equal to 0.
5.) Set the element $a_{j.}$
6.) Clear the element $a_{j+1.}$
7.) Get the value of element $a_{ij.}$
8.) It will then check the condition for rows if the index is equal to maximum value otherwise it will increment its value and repeat the step again.
9.) If it is equal to maximum value it will check the column index if column index is equal to maximum value, it will exit from the loop. Otherwise it will increment the column index and go back to third step again.

### 3.2.4 Interfacing memory to the microcontroller

For interfacing memory to the microcontroller, the algorithm for writing to and reading from the memory are made separately:

Algorithm for writing data to the memory is as:

1.) Firstly the contents of address buffer high 'addr_buf0_hi' are moved to data pointer high dph.
2.) The contents of address buffer low 'addr_buf0_lo' are moved to data pointer low dpl.
3.) The value of selected channel 'val_adc_ch-'is moved to the accumulator a.
4.) Then the contents of accumulator are moved to memory address pointed by the data pointer.
5.) Then a delay of 20 ms is called.

Flowchart for writing data to the memory are shown in figure 3.8

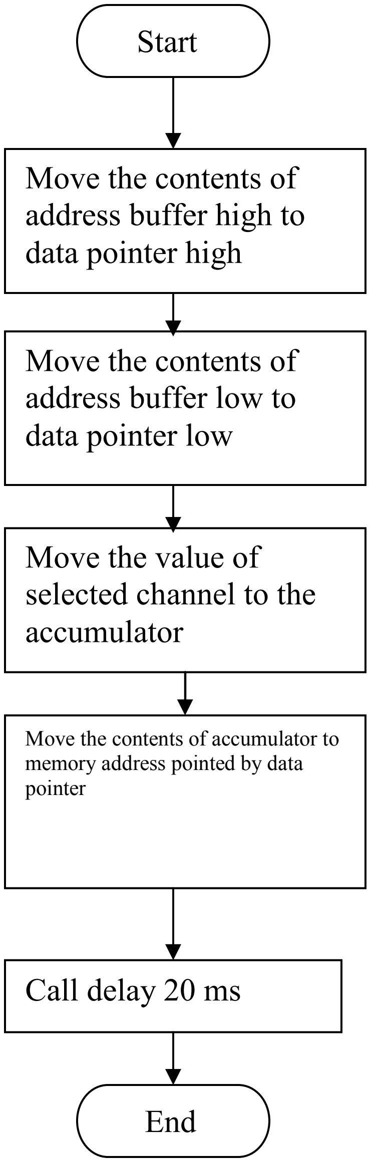


**Figure 3.8 Flowchart for writing data to the memory**

Algorithm for reading the data from the memory is as:

1.) The contents of address buffer high 'addr_buf0_hi' are moved to data pointer high dph.

2.) The contents of address buffer low 'addr_buf0_lo' are moved to data pointer low dpl.

3.) In next step, the value of memory address pointed by the data pointer is moved to the accumulator.

Flowchart for reading data from the memory is shown in figure 3.9

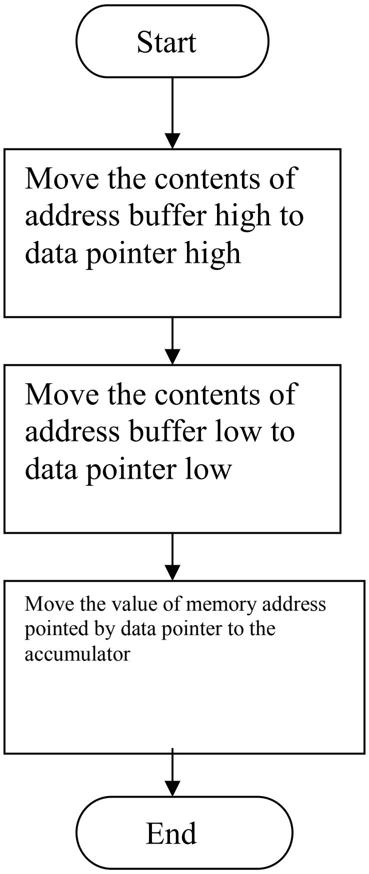


**Figure 3.9 Flowchart for reading data from memory**

### 3.2.5 Interfacing LCD to the microcontroller

For interfacing a LCD to the microcontroller it has to be first initialized then command and data are sent to it.

Algorithm for initializing the LCD is as:

1.) Firstly, the interface length is set.

2.) A high to low pulse is applied to the pin ‘en_lcd’.

3.) A delay of 20 ms is then called.

4.) The display is turned on and a high to low pulse is again applied to the pin ‘en_lcd’.

5.) The cursor move direction is set in next step and shift of display is specified.

6.) A delay subroutine of 50 ms is called and a high to low pulse is applied to pin ‘en_lcd’.

Flowchart for LCD initialization is shown in figure 3.10

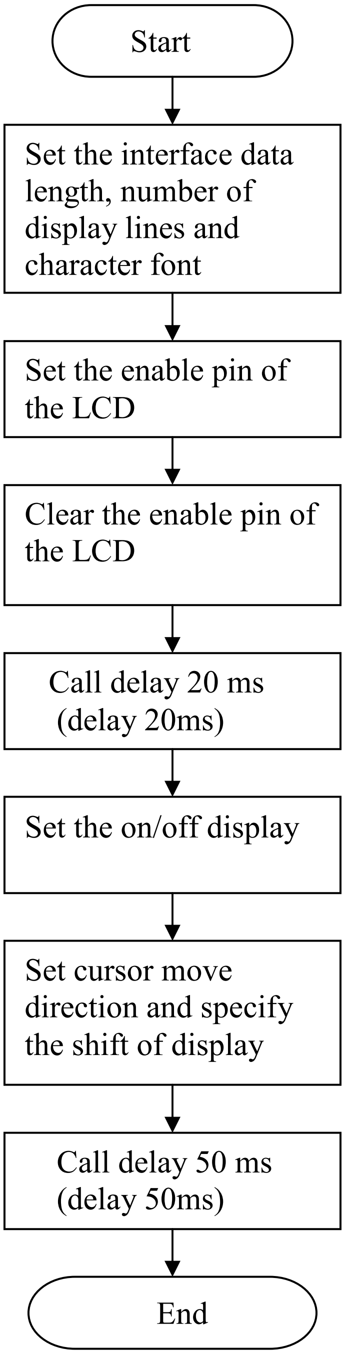


**Figure 3.10 Flowchart for LCD initialization**

Algorithm for writing data to the LCD is as:

1.) In writing data to the LCD, register select pin (rs) is set to low.
2.) Then the cursor on/off and blink of cursor position character is set.
3.) A high to low pulse is sent to the enable pin ‘en_lcd’. The register select pin (rs) is then set.
4.) The contents of the accumulator are moved to port 0. A high to low pulse, in order for the LCD to latch in the data present at the data pins is applied to the enable pin ‘en_lcd’.

For writing the data to the LCD panel flowchart is shown as:

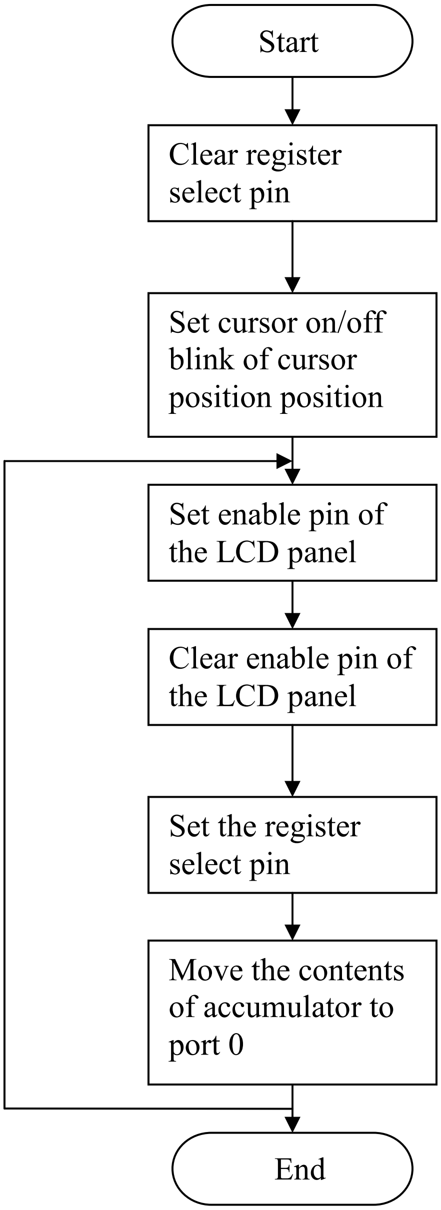


**Figure 3.11 Flowchart for writing data to the LCD**

# Chapter 4: Results and Discussions

## 4.1 Results

. In this system, Temperature measurements from the three channels are taken. The performance of the three channels is distinguished on the basis of their accuracy. All the sensors are specified with accuracy. The accuracy indicates how closely the sensor can measure the actual or real world parameter value. The more accurate a sensor is, better it will perform.

The readings are taken under different conditions for some time interval. Also the readings are taken at different temperatures in a time interval. Comparing the readings obtained from the three channels under the different conditions the most accurate channel among them is found.

Table 4.1, 4.2, 4.3, 4.4 shows the readings of measurement of temperature obtained from the three channels at different conditions.

### 4.1.1 Experimental results taken under normal conditions

Table 4.1 shows the readings of channels at room temperature after time intervals of 5 minutes. The readings of the channels are compared with the readings of temperatures obtained from the standard temperature indications obtained from the thermometer.

| Time (in minutes) | Standard temperature (°C) | Ch1 (°C) | Ch2 (°C) | Ch3 (°C) |
|---|---|---|---|---|
| 8:05 | 37.9 | 36 | 00 | 00 |
| 8:10 | 37.9 | 36 | 00 | 00 |
| 8:15 | 37.9 | 36 | 38 | 00 |
| 8:20 | 37.9 | 36 | 38 | 00 |
| 8:25 | 37.9 | 36 | 38 | 00 |
| 8:30 | 37.9 | 36 | 38 | 37 |
| 8:35 | 37.9 | 37 | 38 | 37 |
| 8:40 | 37.9 | 37 | 38 | 37 |
| 8:45 | 37.9 | 37 | 39 | 37 |

**Table 4.1 Results of measurement at room temperature**

### 4.1.2 Experimental results taken at low temperature conditions

Table 4.2 shows the readings obtained from the three channels at interval of 5 minutes at air conditioned temperatures. The readings of the channels are compared with the readings of temperatures obtained from the standard temperature indications obtained from the thermometer.

| Time (in minutes) | Standard temperature (°C) | Ch1 (°C) | Ch2 (°C) | Ch3 (°C) |
|---|---|---|---|---|
| 10:05 | 37 | 34 | 00 | 00 |
| 10:10 | 37 | 34 | 00 | 00 |
| 10:15 | 37 | 34 | 33 | 00 |
| 10:20 | 37 | 34 | 33 | 00 |
| 10:25 | 37 | 34 | 33 | 00 |
| 10:30 | 37 | 34 | 33 | 32 |
| 10:35 | 37 | 31 | 33 | 32 |
| 10:40 | 37 | 31 | 33 | 32 |
| 10:45 | 37 | 31 | 32 | 32 |
| 10:50 | 37 | 31 | 32 | 32 |
| 10:55 | 37 | 31 | 32 | 32 |
| 11:00 | 37 | 31 | 32 | 35 |

**Table 4.2 Results of measurement under low temperature conditions**

### 4.1.3 Experimental results taken at different temperatures

Table 4.3 shows the readings of the channels at different temperatures in a time of 30 minutes. The readings of the channels are compared with the readings of temperatures obtained from the standard temperature indications obtained from the thermometer.

| **Time (in minutes)** | **Standard temperature (°C)** | **Ch1 (°C)** | **Ch2 (°C)** | **Ch3 (°C)** |
|---|---|---|---|---|
| 30 | 37.2 | 34 | 35 | 36 |
| 30 | 37.5 | 32 | 34 | 33 |
| 30 | 38.1 | 35 | 36 | 35 |
| 30 | 38.3 | 34 | 36 | 35 |
| 30 | 38.8 | 35 | 36 | 36 |
| 30 | 39.6 | 36 | 38 | 38 |
| 30 | 41.1 | 40 | 42 | 43 |

**Table 4.3 Results of measurements taken at different temperatures**

### 4.1.4 Experimental results taken over a long time (12 hrs.)

Table 4.4 shows the readings of channels for a long time of 12 hrs. The readings of the channels are compared with the readings of temperatures obtained from the standard temperature indications obtained from the thermometer.

| **Time (in minutes)** | **Standard temperature (°C)** | **Ch1 (°C)** | **Ch2 (°C)** | **Ch3 (°C)** |
|---|---|---|---|---|
| 8:00 | 37 | 00 | 00 | 00 |
| 8:30 | 37 | 34 | 35 | 35 |
| 9:00 | 37 | 35 | 36 | 36 |
| 9:30 | 37 | 34 | 35 | 35 |
| 10:00 | 37 | 34 | 35 | 36 |
| 10:30 | 37 | 35 | 36 | 36 |
| 11:00 | 37 | 36 | 37 | 36 |
| 11.30 | 37 | 36 | 37 | 36 |
| 12:00 | 37 | 36 | 37 | 36 |
| 12:30 | 37 | 36 | 37 | 36 |
| 1:00 | 37 | 36 | 37 | 37 |
| 1:30 | 37 | 36 | 37 | 37 |
| 2:00 | 37 | 36 | 37 | 37 |
| 2:30 | 38 | 36 | 37 | 37 |
| 3:00 | 38 | 36 | 37 | 37 |
| 3:30 | 38 | 36 | 37 | 37 |

| | | | | |
|---|---|---|---|---|
| 4:00 | 38 | 36 | 37 | 37 |
| 4:30 | 38 | 36 | 37 | 37 |
| 5:00 | 38 | 36 | 38 | 37 |
| 5:30 | 38 | 36 | 38 | 37 |
| 6:00 | 38 | 36 | 38 | 37 |
| 6:30 | 38 | 36 | 38 | 37 |
| 7:00 | 38 | 36 | 38 | 37 |
| 7:30 | 38 | 36 | 38 | 37 |
| 8:00 | 38 | 37 | 38 | 38 |

**Table 4.4 Results of measurement taken over a long time**

## 4.2 Discussions

From the above tables of readings obtained by comparing standard temperature with the temperature of channels, the accuracy of the channels is to be discussed. Accuracy is the degree of conformity of a measured/calculated quantity to its actual (true) value that is the quality of nearness to the truth or the true value.

From table 4.1, it is clear that Channel 2 is more close to the standard reading. The error found is .1% .the error of channel 1 is found to be 1.9% and of channel 3 is .9%. So channel 2 is more accurate one among the three because lesser is the error more will be the accuracy.

**From table 4.2, it is clear that channel 2 is more accurate because of less deviation among the values and close to standard one.**

**From table 4.3, it is clear that the temperature of channel 2 is more constant than other channels and also close to standard temperature .So channel 2 is more accurate at different temperatures.**

**From table 4.4, the temperature of channel 2 is more close to standard temperature for a long time, as compared to other channels.**

From the above discussion, it is concluded that among the three channels, the channel 2 is most accurate at different temperatures and under different conditions.

# Chapter 5: Conclusion and Future Scope

## 5.1 Conclusion

The data logger is an invaluable tool to collect and analyse experimental data, having the ability to clearly present real time results, with sensors and probes able to respond to parameters that are beyond the normal range available from most traditional equipment. Data loggers used for measuring the temperature might have certain limitations in terms of speed, memory and cost. Also data loggers with increased number of channels are complex.

In this work, an attempt has been done to design a data logger, which is of less cost, portable, very low power consumption, self contained. It is an efficient data logger, which works in real time mode. The reduced number of channels also makes the system simple. The logger can use up to 8 channels of analog to digital converter in performing its task but that will result in increased number of channels.

A step-by-step approach in designing a Microcontroller based system for temperature measurement has been followed. According to the study and analysis of various parts of the system, a design has been carried out. The results obtained from the measurement have shown that the system perform well under all the conditions.

From this work, it is concluded that in this system among the three channels, Channel 2 more accurately measures the temperature in the given span of time.

## 5.2 Future Scope

1.) The performance of microcontroller based temperature data logger has been found on the expected lines. However, there exists a scope for further improvement in its speed, number of channels, power consumption, and PC interface software for post data analysis.

2.) The system can be modified with the use of graphical LCD panel so that the analysis is done by the system itself. The number of analog channels can be increased to monitor more sensor outputs.

3.) The low power requirement of this data logger makes it easy to use. The device can be made to perform better by providing the power supply with the help of battery source which can be rechargeable or non-rechargeable, to reduce the requirement of main AC power.

4.) This system can be connected to communication devices such as modems, cellular phones, or satellite terminal to enable the remote collection of recorded data or alarming of certain parameters. The new system will email information based upon a regular schedule of based upon alarms.

5.) Moreover, system can be made user friendly by interfacing it with user friendly software and thus can support the post process analysis. There lies the scope to make the system application specific.

6.) The system can also be modified to change the scan time of the channels.

7.) The performance of the system can also be improved by choosing microcontroller chip, which performs better than 8051 chip. It differs in memory capacity, operating speed, instruction cycle period etc .A variety of microcontroller chips are available with better features. Thus, the system can be made more efficient and of better performance by the use of chip having better features than 8051.

# References

[1] http/www.loggingtutorial/dr.saulgreenburg/htm.

[2] Andrew J Thompson, John L Bahr and Neil R Thomson, “Low power data logger”, proceedings of conference department of physics, university of otago, Dunedin.

[3] http://en.wikipedia.org/wiki/temperature-measurement

[4] www.ni.com/dataloggers.

[5] H S kalsi, “Electronic instrumentation”, Tata McGraw-Hill Ltd., New Delhi, 1999.

[6] Peter Roberson, “Using data loggers”, science teachers’ workshop, 2004, north Sydney.

[7] Muhammad Ali Mazidi and Janice Gillispe Mazidi, “The 8051 microcontroller and embedded systems”, Pearson education ltd., India, 2004.

[8] National Semiconductor Corporation, LM35 datasheet, precision centigrade temperature sensors, Atmel data book, November 2000 update.

[9] National semiconductor corporation, ADC0808/ADC 0809 data sheet, 8-bit Microprocessor compatible A/D converters with 8-channel multiplexer, national Semiconductor data book, October 2002 update.

[10] http://ing.pub.ro/engleza/materials/4/8051.pdf.

[11] Atmel corporation, AT89C51 data sheet, 8-bit microcontroller with 4k bytes flash, Atmel Data book, 2000 update.

[12] 8051 tutorial, Copyright 1997 - 2005 by vault information services LLC. (http://8052.com/tut8051.phtml).

[13] http://en.wikipedia.org/wiki/Liquid_crystal_display.

[14] Introduction to LCD programming tutorial by Craig Steiner Copyright 1997 - 2005 by Vault information services LLC (http://8052.com/tutlcd.phtml).

[15] Atmel corporation, 64k (8kx8) parallel EEPROMS, AT28C64 and AT28C64X data sheet Atmel data book, 1999 update.

[16] Software based real time clock tutorial by Craig Steiner Copyright 1997 - 2005 by Vault information services LLC (http://8052.com/tutrtc.phtml).

[17] A guide to data logging tutorial by Judy Ritchie, Copyright 1996-2003 by Onset Computer Corporation.

[18] S. J. Perez, M. A. Calva, R. Castaneda, "A microcontroller based data logging System" Instrumentation and Development Vol. 3 Nr. 8, copyright 1997, Journal of the Mexican Society of Instrumentation.

[19] www.academic.hws.edu/geo/logger/paper/loggerpaper.pdf.

[20] Rajesh Luharuka, Robert X. Gao, Sundar Krishnamurty, "A microcontroller-based portable GSR data logger for physiological sensing" , proceedings of conference department of Mechanical and Industrial Engineering ,University of Massachusetts, Amherst.

# ***Appendix-1***

Datasheet of AT89C51

## Features

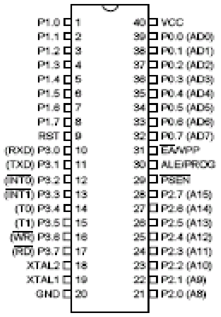
- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

## Description

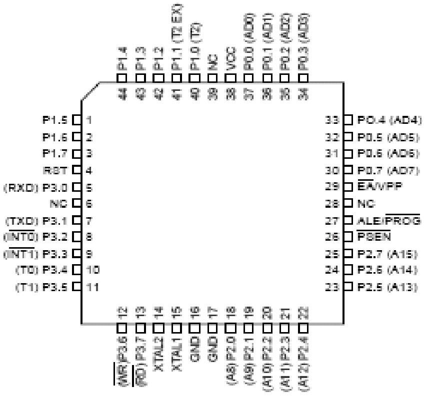
The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

## Pin Configurations

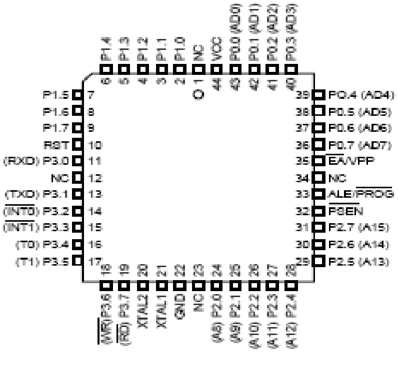
PDIP



PQFP/TQFP



PLCC



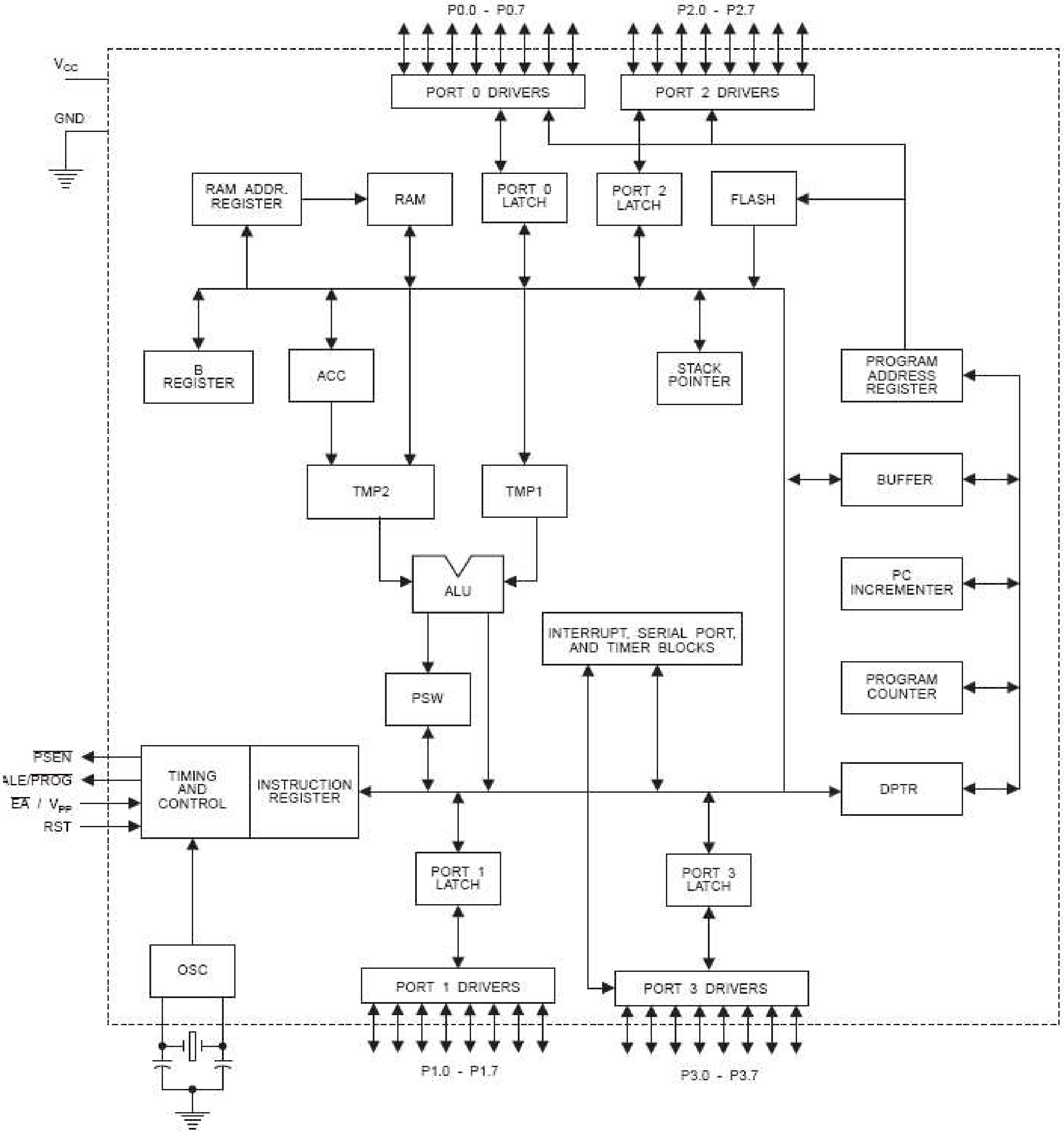
**8-bit Microcontroller with 4K Bytes Flash**

**AT89C51**

**Not Recommended for New Designs. Use AT89S51.**

Rev. 0265G–02/00

## Block Diagram



The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power-down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

## Pin Description

### VCC

Supply voltage.

### GND

Ground.

### Port 0

Port 0 is an 8-bit open-drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

### Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current ($I_{IL}$) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and verification.

### Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current ($I_{IL}$) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

### Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current ($I_{IL}$) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

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

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

### RST

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

### ALE/$\overline{\text{PROG}}$

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{\text{PROG}}$) during Flash programming.

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

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

**$\overline{\text{PSEN}}$**

Program Store Enable is the read strobe to external program memory.

When the AT89C51 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.

**$\overline{\text{EA}}$/VPP**

External Access Enable. $\overline{\text{EA}}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{\text{EA}}$ will be internally latched on reset.

$\overline{\text{EA}}$ should be strapped to $V_{CC}$ for internal program executions.

This pin also receives the 12-volt programming enable voltage ($V_{PP}$) during Flash programming, for parts that require 12-volt $V_{PP}$.

**XTAL1**

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

**XTAL2**

Output from the inverting oscillator amplifier.

## Oscillator Characteristics

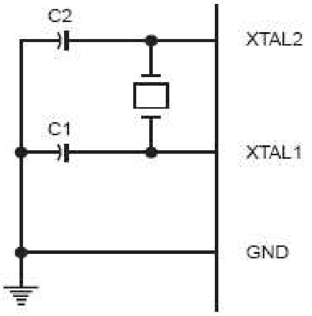
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

## Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

It should be noted that when idle is terminated by a hard ware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

**Figure 1.** Oscillator Connections

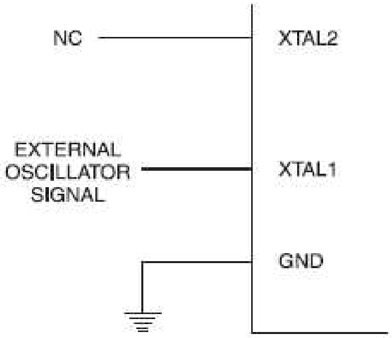


Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

## Status of External Pins During Idle and Power-down Modes

| Mode | Program Memory | ALE | $\overline{\text{PSEN}}$ | PORT0 | PORT1 | PORT2 | PORT3 |
|---|---|---|---|---|---|---|---|
| Idle | Internal | 1 | 1 | Data | Data | Data | Data |
| Idle | External | 1 | 1 | Float | Data | Address | Data |
| Power-down | Internal | 0 | 0 | Data | Data | Data | Data |
| Power-down | External | 0 | 0 | Float | Data | Data | Data |

**Figure 2.** External Clock Drive Configuration



## Power-down Mode

In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before $V_{CC}$ is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

## Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below.

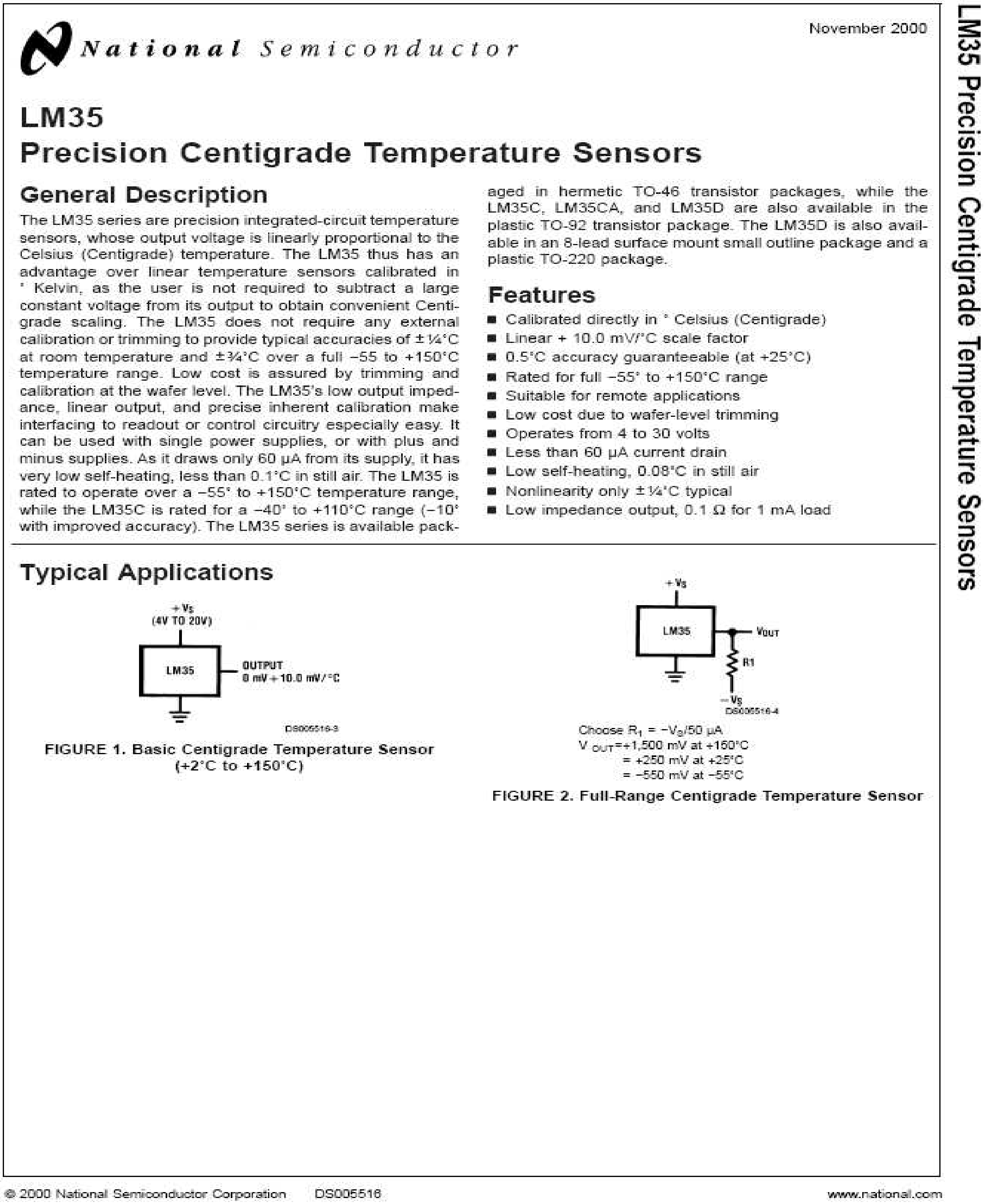
When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of $\overline{EA}$ be in agreement with the current logic level at that pin in order for the device to function properly.

## Lock Bit Protection Modes

| | Program Lock Bits | | | |
|---|---|---|---|---|
| | LB1 | LB2 | LB3 | Protection Type |
| 1 | U | U | U | No program lock features |
| 2 | P | U | U | MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash is disabled |
| 3 | P | P | U | Same as mode 2, also verify is disabled |
| 4 | P | P | P | Same as mode 3, also external execution is disabled |

# ***Appendix-2***

## Datasheet of LM35 Temperature Sensor



National Semiconductor

November 2000

LM35 Precision Centigrade Temperature Sensors

# LM35
# Precision Centigrade Temperature Sensors

## General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

## Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±¼°C typical
- Low impedance output, 0.1 Ω for 1 mA load

## Typical Applications

FIGURE 1. Basic Centigrade Temperature Sensor (+2°C to +150°C)

Choose $R_1 = -V_S/50\ \mu A$

$V_{OUT} = +1{,}500$ mV at +150°C

= +250 mV at +25°C

= −550 mV at −55°C

FIGURE 2. Full-Range Centigrade Temperature Sensor

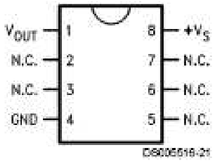
## Connection Diagrams

TO-46
Metal Can Package*

*Case is connected to negative pin (GND)

**Order Number LM35H, LM35AH, LM35CH, LM35CAH or LM35DH**
**See NS Package Number H03H**

SO-8
Small Outline Molded Package



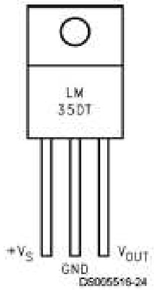
N.C. = No Connection

**Top View**
**Order Number LM35DM**
**See NS Package Number M08A**

TO-92
Plastic Package

**Order Number LM35CZ, LM35CAZ or LM35DZ**
**See NS Package Number Z03A**

TO-220
Plastic Package*



*Tab is connected to the negative pin (GND).

**Note:** The LM35DT pinout is different than the discontinued LM35DP.

**Order Number LM35DT**
**See NS Package Number TA03F**

# *Appendix-3*

Datasheet of AT28C64

## 64K (8K x 8) Parallel EEPROMs

## AT28C64
## AT28C64X

### Features

- **Fast Read Access Time – 120 ns**
- **Fast Byte Write – 200 µs or 1 ms**
- **Self-timed Byte Write Cycle**
  - **Internal Address and Data Latches**
  - **Internal Control Timer**
  - **Automatic Clear Before Write**
- **Direct Microprocessor Control**
  - **READY/$\overline{\text{BUSY}}$ Open Drain Output**
  - **$\overline{\text{DATA}}$ Polling**
- **Low Power**
  - **30 mA Active Current**
  - **100 µA CMOS Standby Current**
- **High Reliability**
  - **Endurance: $10^4$ or $10^5$ Cycles**
  - **Data Retention: 10 Years**
- **5V ± 10% Supply**
- **CMOS and TTL Compatible Inputs and Outputs**
- **JEDEC Approved Byte-wide Pinout**
- **Commercial and Industrial Temperature Ranges**

### Description

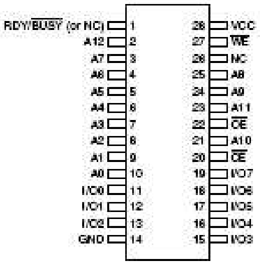
The AT28C64 is a low-power, high-performance 8,192 words by 8-bit nonvolatile electrically erasable and programmable read only memory with popular, easy-to-use features. The device is manufactured with Atmel's reliable nonvolatile technology.

*(continued)*

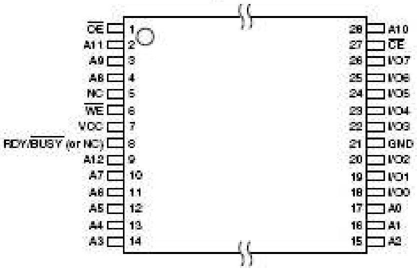
### Pin Configurations

| Pin Name | Function |
|---|---|
| A0 - A12 | Addresses |
| $\overline{\text{CE}}$ | Chip Enable |
| $\overline{\text{OE}}$ | Output Enable |
| $\overline{\text{WE}}$ | Write Enable |
| I/O0 - I/O7 | Data Inputs/Outputs |
| RDY/$\overline{\text{BUSY}}$ | Ready/$\overline{\text{Busy}}$ Output |
| NC | No Connect |
| DC | Don't Connect |

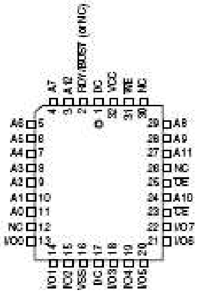
PDIP, SOIC
Top View



TSOP
Top View



LCC, PLCC
Top View



Note: PLCC package pins 1 and 17 are DON'T CONNECT.

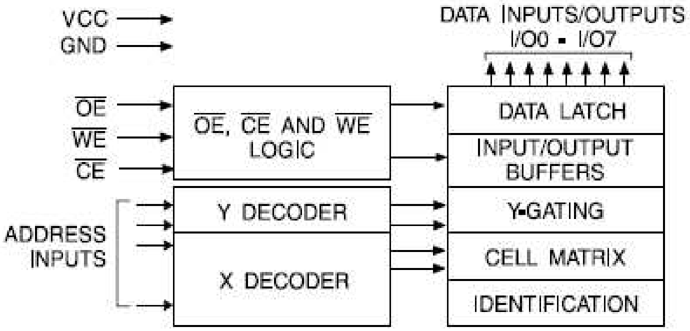
Rev. 0001H–12/99

The AT28C64 is accessed like a Static RAM for the read or write cycles without the need for external components. During a byte write, the address and data are latched internally, freeing the microprocessor address and data bus for other operations. Following the initiation of a write cycle, the device will go to a busy state and automatically clear and write the latched data using an internal control timer. The device includes two methods for detecting the end of a write cycle, level detection of RDY/$\overline{\text{BUSY}}$ (unless pin 1 is N.C.) and $\overline{\text{DATA}}$ Polling of I/$O_7$. Once the end of a write cycle has been detected, a new access for a read or write can begin.

The CMOS technology offers fast access times of 120 ns at low power dissipation. When the chip is deselected the standby current is less than 100 µA.

Atmel's AT28C64 has additional features to ensure high quality and manufacturability. The device utilizes error correction internally for extended endurance and for improved data retention characteristics. An extra 32 bytes of EEPROM are available for device identification or tracking.

## Block Diagram



## Absolute Maximum Ratings*

| | |
|---|---|
| Temperature under Bias | -55°C to +125°C |
| Storage Temperature | -65°C to +150°C |
| All Input Voltages (including NC Pins) with Respect to Ground | -0.6V to +6.25V |
| All Output Voltages with Respect to Ground | -0.6V to $V_{CC}$ + 0.6V |
| Voltage on $\overline{\text{OE}}$ and A9 with Respect to Ground | -0.6V to +13.5V |

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability

## Device Operation

**READ:** The AT28C64 is accessed like a Static RAM. When $\overline{CE}$ and $\overline{OE}$ are low and $\overline{WE}$ is high, the data stored at the memory location determined by the address pins is asserted on the outputs. The outputs are put in a high impedance state whenever $\overline{CE}$ or $\overline{OE}$ is high. This dual line control gives designers increased flexibility in preventing bus contention.

**BYTE WRITE:** Writing data into the AT28C64 is similar to writing into a Static RAM. A low pulse on the $\overline{WE}$ or $\overline{CE}$ input with $\overline{OE}$ high and $\overline{CE}$ or $\overline{WE}$ low (respectively) initiates a byte write. The address location is latched on the falling edge of $\overline{WE}$ (or $\overline{CE}$); the new data is latched on the rising edge. Internally, the device performs a self-clear before write. Once a byte write has been started, it will automatically time itself to completion. Once a programming operation has been initiated and for the duration of $t_{WC}$, a read operation will effectively be a polling operation.

**FAST BYTE WRITE:** The AT28C64E offers a byte write time of 200 µs maximum. This feature allows the entire device to be rewritten in 1.6 seconds.

**READY/$\overline{\text{BUSY}}$:** Pin 1 is an open drain RDY/$\overline{\text{BUSY}}$ output that can be used to detect the end of a write cycle. RDY/$\overline{\text{BUSY}}$ is actively pulled low during the write cycle and is released at the completion of the write. The open drain connection allows for OR-tying of several devices to the same RDY/$\overline{\text{BUSY}}$ line. The RDY/$\overline{\text{BUSY}}$ pin is not connected for the AT28C64X.

**$\overline{\text{DATA}}$ POLLING:** The AT28C64 provides $\overline{\text{DATA}}$ Polling to signal the completion of a write cycle. During a write cycle, an attempted read of the data being written results in the complement of that data for $I/O_7$ (the other outputs are indeterminate). When the write cycle is finished, true data appears on all outputs.

**WRITE PROTECTION:** Inadvertent writes to the device are protected against in the following ways: (a) $V_{CC}$ sense – if $V_{CC}$ is below 3.8V (typical) the write function is inhibited; (b) $V_{CC}$ power on delay – once $V_{CC}$ has reached 3.8V the device will automatically time out 5 ms (typical) before allowing a byte write; and (c) write inhibit – holding any one of $\overline{OE}$ low, $\overline{CE}$ high or $\overline{WE}$ high inhibits byte write cycles.

**CHIP CLEAR:** The contents of the entire memory of the AT28C64 may be set to the high state by the CHIP CLEAR operation. By setting $\overline{CE}$ low and $\overline{OE}$ to 12 volts, the chip is cleared when a 10 msec low pulse is applied to $\overline{WE}$.

**DEVICE IDENTIFICATION:** An extra 32 bytes of EEPROM memory are available to the user for device identification. By raising A9 to 12 ± 0.5V and using address locations 1FE0H to 1FFFH the additional bytes may be written to or read from in the same manner as the regular memory array.

## DC and AC Operating Range

| | | AT28C64-12 | AT28C64-15 | AT28C64-20 | AT28C64-25 |
|---|---|---|---|---|---|
| Operating Temperature (Case) | Com. | 0°C - 70°C | 0°C - 70°C | 0°C - 70°C | 0°C - 70°C |
| | Ind. | -40°C - 85°C | -40°C - 85°C | -40°C - 85°C | -40°C - 85°C |
| $V_{CC}$ Power Supply | | 5V ± 10% | 5V ± 10% | 5V ± 10% | 5V ± 10% |

## Operating Modes

| Mode | $\overline{CE}$ | $\overline{OE}$ | $\overline{WE}$ | I/O |
|---|---|---|---|---|
| Read | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | $D_{OUT}$ |
| Write[2] | $V_{IL}$ | $V_{IH}$ | $V_{IL}$ | $D_{IN}$ |
| Standby/Write Inhibit | $V_{IH}$ | X[1] | X | High Z |
| Write Inhibit | X | X | $V_{IH}$ | |
| Write Inhibit | X | $V_{IL}$ | X | |
| Output Disable | X | $V_{IH}$ | X | High Z |
| Chip Erase | $V_{IL}$ | $V_H$[3] | $V_{IL}$ | High Z |

Notes: 1. X can be $V_{IL}$ or $V_{IH}$.

2. Refer to AC programming waveforms.

3. $V_H = 12.0V \pm 0.5V$.

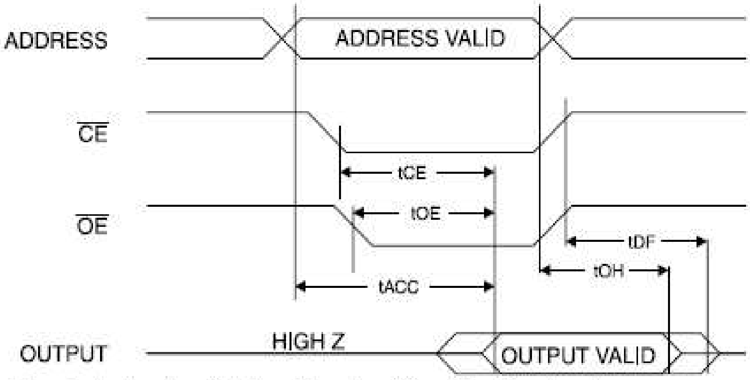
## DC Characteristics

| Symbol | Parameter | Condition | | Min | Max | Units |
|---|---|---|---|---|---|---|
| $I_{LI}$ | Input Load Current | $V_{IN}$ = 0V to $V_{CC}$ + 1V | | | 10 | µA |
| $I_{LO}$ | Output Leakage Current | $V_{I/O}$ = 0V to $V_{CC}$ | | | 10 | µA |
| $I_{SB1}$ | $V_{CC}$ Standby Current CMOS | $\overline{CE}$ = $V_{CC}$ - 0.3V to $V_{CC}$ + 1.0V | | | 100 | µA |
| $I_{SB2}$ | $V_{CC}$ Standby Current TTL | $\overline{CE}$ = 2.0V to $V_{CC}$ + 1.0V | Com. | | 2 | mA |
| | | | Ind. | | 3 | mA |
| $I_{CC}$ | $V_{CC}$ Active Current AC | f = 5 MHz; $I_{OUT}$ = 0 mA<br>$\overline{CE}$ = $V_{IL}$ | Com. | | 30 | mA |
| | | | Ind. | | 45 | mA |
| $V_{IL}$ | Input Low Voltage | | | | 0.8 | V |
| $V_{IH}$ | Input High Voltage | | | 2.0 | | V |
| $V_{OL}$ | Output Low Voltage | $I_{OL}$ = 2.1 mA<br>= 4.0 mA for RDY/$\overline{BUSY}$ | | | 0.45 | V |
| $V_{OH}$ | Output High Voltage | $I_{OH}$ = -400 µA | | 2.4 | | V |

## AC Read Characteristics

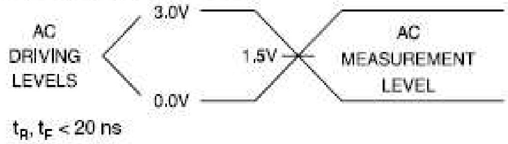
| | | AT28C64-12 | | AT28C64-15 | | AT28C64-20 | | AT28C64-25 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Symbol** | **Parameter** | **Min** | **Max** | **Min** | **Max** | **Min** | **Max** | **Min** | **Max** | **Units** |
| $t_{ACC}$ | Address to Output Delay | | 120 | | 150 | | 200 | | 250 | ns |
| $t_{CE}$(1) | $\overline{\text{CE}}$ to Output Delay | | 120 | | 150 | | 200 | | 250 | ns |
| $t_{OE}$(2) | $\overline{\text{OE}}$ to Output Delay | 10 | 60 | 10 | 70 | 10 | 80 | 10 | 100 | ns |
| $t_{DF}$(3)(4) | $\overline{\text{CE}}$ or $\overline{\text{OE}}$ High to Output Float | 0 | 45 | 0 | 50 | 0 | 55 | 0 | 60 | ns |
| $t_{OH}$ | Output Hold from $\overline{\text{OE}}$, $\overline{\text{CE}}$ or Address, whichever occurred first | 0 | | 0 | | 0 | | 0 | | ns |

## AC Read Waveforms(1)(2)(3)(4)



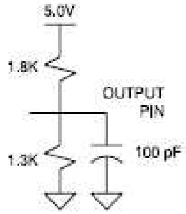
Notes:
1. $\overline{\text{CE}}$ may be delayed up to $t_{ACC}$ - $t_{CE}$ after the address transition without impact on $t_{ACC}$.
2. $\overline{\text{OE}}$ may be delayed up to $t_{CE}$ - $t_{OE}$ after the falling edge of $\overline{\text{CE}}$ without impact on $t_{CE}$ or by $t_{ACC}$ - $t_{OE}$ after an address change without impact on $t_{ACC}$.
3. $t_{DF}$ is specified from $\overline{\text{OE}}$ or $\overline{\text{CE}}$ whichever occurs first ($C_L$ = 5 pF).
4. This parameter is characterized and is not 100% tested.

## Input Test Waveforms and Measurement Level



$t_R$, $t_F$ < 20 ns

## Output Test Load



## Pin Capacitance

f = 1 MHz, T = 25°C(1)

| Symbol | Typ | Max | Units | Conditions |
|---|---|---|---|---|
| $C_{IN}$ | 4 | 6 | pF | $V_{IN}$ = 0V |
| $C_{OUT}$ | 8 | 12 | pF | $V_{OUT}$ = 0V |

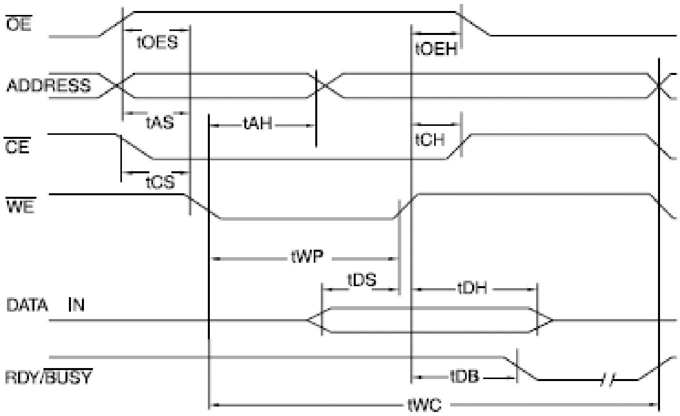
Note: 1. This parameter is characterized and is not 100% tested.

## AC Write Characteristics

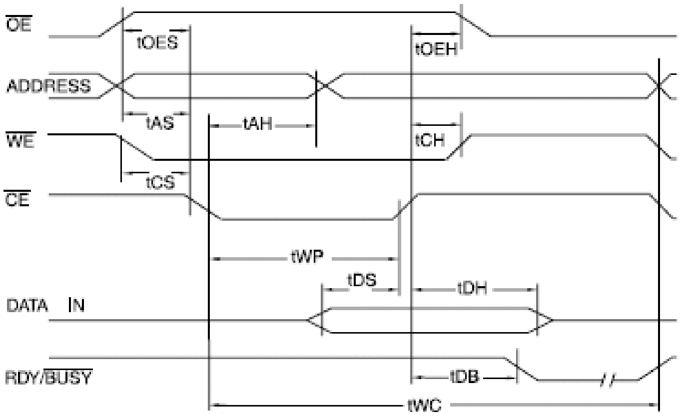
| Symbol | Parameter | | Min | Max | Units |
|---|---|---|---|---|---|
| $t_{AS}$, $t_{OES}$ | Address, $\overline{OE}$ Setup Time | | 10 | | ns |
| $t_{AH}$ | Address Hold Time | | 50 | | ns |
| $t_{WP}$ | Write Pulse Width ($\overline{WE}$ or $\overline{CE}$) | | 100 | 1000 | ns |
| $t_{DS}$ | Data Setup Time | | 50 | | ns |
| $t_{DH}$, $t_{OEH}$ | Data, $\overline{OE}$ Hold Time | | 10 | | ns |
| $t_{CS}$, $t_{CH}$ | $\overline{CE}$ to $\overline{WE}$ and $\overline{WE}$ to $\overline{CE}$ Setup and Hold Time | | 0 | | ns |
| $t_{DB}$ | Time to Device Busy | | | 50 | ns |
| $t_{WC}$ | Write Cycle Time (option available) | AT28C64 | | 1 | ms |
| | | AT28C64E | | 200 | µs |

## AC Write Waveforms

### $\overline{WE}$ Controlled



### $\overline{CE}$ Controlled



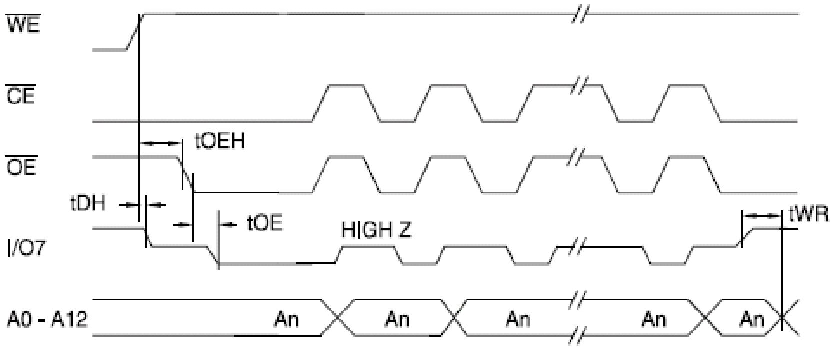
## $\overline{\text{Data}}$ Polling Characteristics[1]

| Symbol | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| $t_{DH}$ | Data Hold Time | 10 | | | ns |
| $t_{OEH}$ | $\overline{\text{OE}}$ Hold Time | 10 | | | ns |
| $t_{OE}$ | $\overline{\text{OE}}$ to Output Delay[2] | | | | ns |
| $t_{WR}$ | Write Recovery Time | 0 | | | ns |

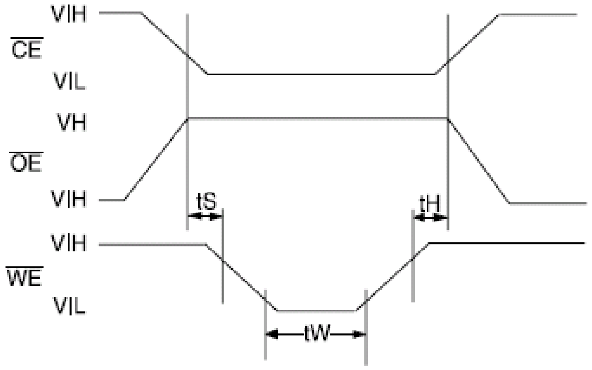
Notes: 1. These parameters are characterized and not 100% tested.

2. See "AC Read Characteristics".

## $\overline{\text{Data}}$ Polling Waveforms



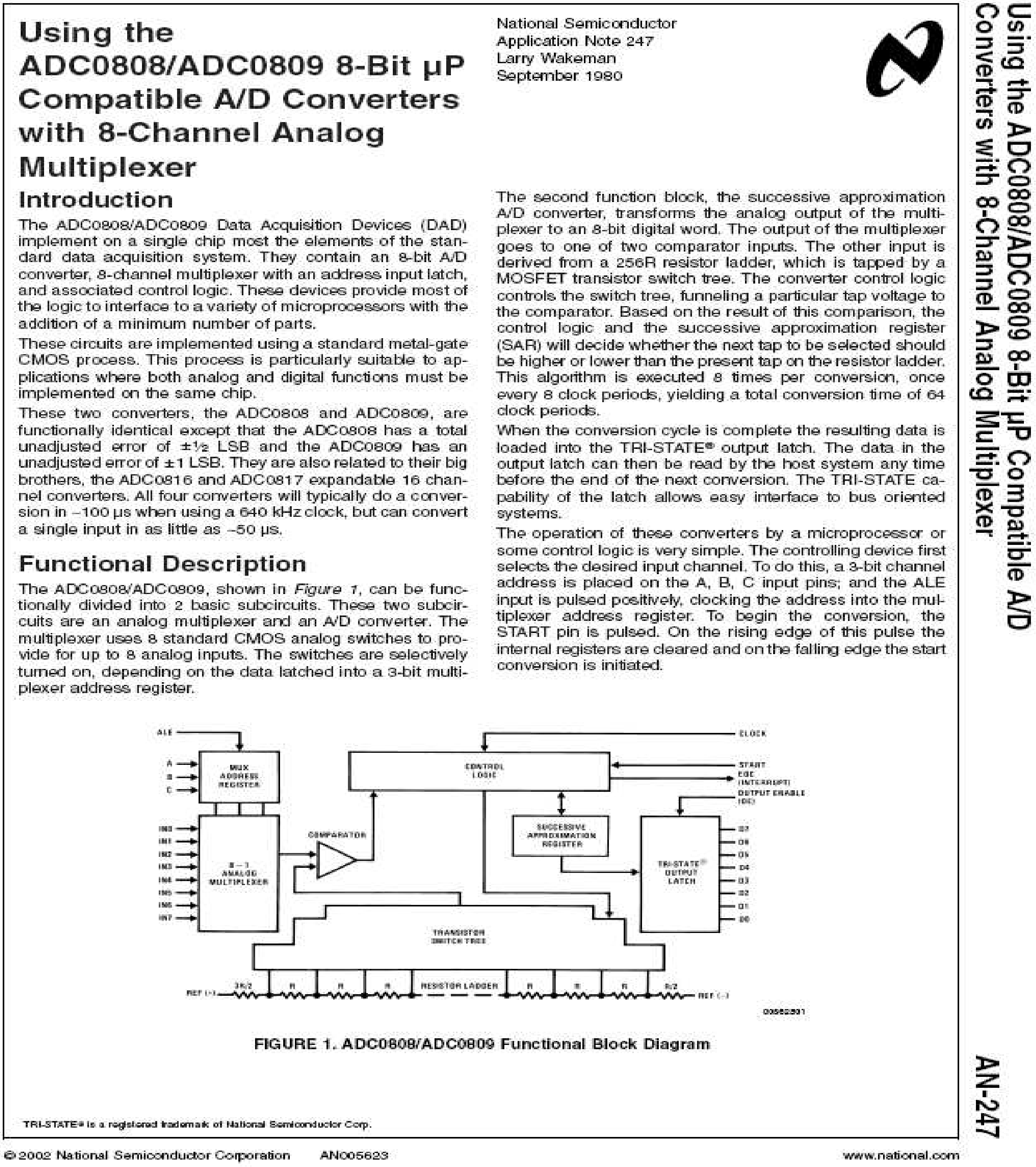
## Chip Erase Waveforms



$t_S = t_H = 1$ µsec (min.)
$t_W = 10$ msec (min.)
$V_H = 12.0 \pm 0.5V$

# <u>*Appendix-4*</u>

Datasheet of ADC 0808/0809



# Using the ADC0808/ADC0809 8-Bit µP Compatible A/D Converters with 8-Channel Analog Multiplexer

National Semiconductor
Application Note 247
Larry Wakeman
September 1980

## Introduction

The ADC0808/ADC0809 Data Acquisition Devices (DAD) implement on a single chip most the elements of the standard data acquisition system. They contain an 8-bit A/D converter, 8-channel multiplexer with an address input latch, and associated control logic. These devices provide most of the logic to interface to a variety of microprocessors with the addition of a minimum number of parts.

These circuits are implemented using a standard metal-gate CMOS process. This process is particularly suitable to applications where both analog and digital functions must be implemented on the same chip.

These two converters, the ADC0808 and ADC0809, are functionally identical except that the ADC0808 has a total unadjusted error of ±½ LSB and the ADC0809 has an unadjusted error of ±1 LSB. They are also related to their big brothers, the ADC0816 and ADC0817 expandable 16 channel converters. All four converters will typically do a conversion in ~100 µs when using a 640 kHz clock, but can convert a single input in as little as ~50 µs.

## Functional Description

The ADC0808/ADC0809, shown in *Figure 1*, can be functionally divided into 2 basic subcircuits. These two subcircuits are an analog multiplexer and an A/D converter. The multiplexer uses 8 standard CMOS analog switches to provide for up to 8 analog inputs. The switches are selectively turned on, depending on the data latched into a 3-bit multiplexer address register.

The second function block, the successive approximation A/D converter, transforms the analog output of the multiplexer to an 8-bit digital word. The output of the multiplexer goes to one of two comparator inputs. The other input is derived from a 256R resistor ladder, which is tapped by a MOSFET transistor switch tree. The converter control logic controls the switch tree, funneling a particular tap voltage to the comparator. Based on the result of this comparison, the control logic and the successive approximation register (SAR) will decide whether the next tap to be selected should be higher or lower than the present tap on the resistor ladder. This algorithm is executed 8 times per conversion, once every 8 clock periods, yielding a total conversion time of 64 clock periods.

When the conversion cycle is complete the resulting data is loaded into the TRI-STATE® output latch. The data in the output latch can then be read by the host system any time before the end of the next conversion. The TRI-STATE capability of the latch allows easy interface to bus oriented systems.

The operation of these converters by a microprocessor or some control logic is very simple. The controlling device first selects the desired input channel. To do this, a 3-bit channel address is placed on the A, B, C input pins; and the ALE input is pulsed positively, clocking the address into the multiplexer address register. To begin the conversion, the START pin is pulsed. On the rising edge of this pulse the internal registers are cleared and on the falling edge the start conversion is initiated.

FIGURE 1. ADC0808/ADC0809 Functional Block Diagram

## Functional Description (Continued)

As mentioned earlier, there are 8 clock periods per approximation. Even though there is no conversion in progress the ADC0808/ADC0809 is still internally cycling through these 8 clock periods. A start pulse can occur any time during this cycle but the conversion will not actually begin until the converter internally cycles to the beginning of the next 8 clock period sequence. As long as the start pin is held high no conversion begins, but when the start pin is taken low the conversion will start within 8 clock periods.

The EOC output is triggered on the rising edge of the start pulse. It, too, is controlled by the 8 clock period cycle, so it will go low within 8 clock periods of the rising edge of the start pulse. One can see that it is entirely possible for EOC to go low before the conversion starts internally, but this is not important, since the positive transition of EOC, which occurs at the end of a conversion, is what the control logic is looking for.

Once EOC does go high this signals the interface logic that the data resulting from the conversion is ready to be read. The output enable (OE) is then raised high. This enables the TRI-STATE outputs, allowing the data to be read. *Figure 3* shows the timing diagram.

## Analog Inputs

### RATIOMETRIC INPUTS

The arrangement of the REF(+) and REF(−) inputs is intended to enable easy design of ratiometric converter systems. The REF inputs are located at either end of the 256R resistor ladder and by proper choice of the input voltages several applications can be easily implemented.

*Figure 2* shows a typical input connection for ratiometric transducers. A ratiometric transducer is a conversion device whose output is proportional to some arbitrary full-scale value. In other words, the transducer's absolute output value is of no particular concern but the ratio of the output to the full-scale is of great importance. For example, the potentiometric displacement transducers of *Figure 2* have this feature. When the wiper is at midscale, the output voltage is $V_O = V_F \times (\text{Wiper Displacement}) = V_F \times 0.5$. This enables the use of much less accurate and less expensive references. The important consideration for this reference is noise. The reference must be "glitch free" because a voltage spike during a conversion cycle could cause conversion inaccuracies.

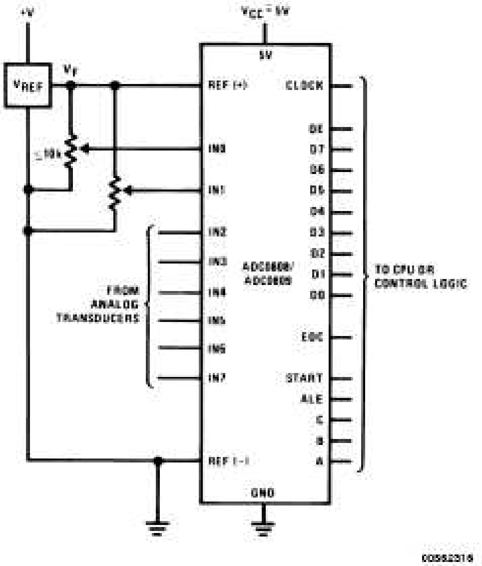


FIGURE 2. Ratiometeric Converter with Separate Reference

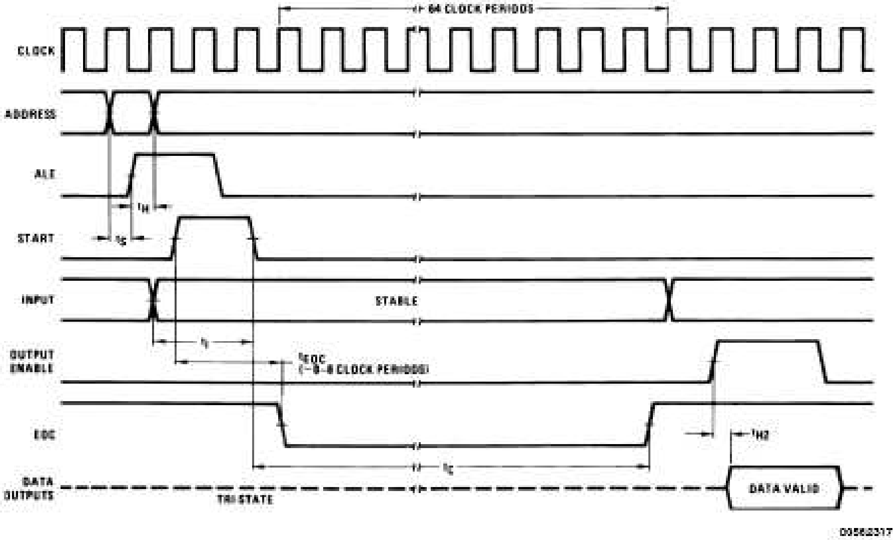


FIGURE 3. ADC0808/ADC0809 Timing Diagram