Aquarium Management

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

In

Electronics and Communication Engineering

Under the Supervision of

Miss Pragya Gupta

By

SNEH SRIJAN (081045)

DIKSHIT CHHABRA (081055)

NISHANT SHARMA (081059)

to

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

CERTIFICATE

This is to certify that project report entitled "Aquarium Management", submitted by Sneh Srijan (081045), Dikshit Chhabra (081055) and Nishant Sharma (081059) in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: 01/06/12

Supervisor's Name: Pragya Gupta

Designation: Sr. Lecturer (E.C.E Dept.)

……………………………

Acknowledgement

This project is an outcome of our serious effort to implement "Aquarium Management" as part of our Bachelor's Degree Program.

To automate living things with the non-living has been a curious question in our day-to-day life and so we made an effort to present something which could help us in this endeavor using the Embedded Systems and under the guidance of our mentor and Guide Miss Pragya Gupta, not only were we able to obtain a vision but also stood motivated at every step.

The prospect of working in a group with a high level of accountability fostered a spirit of teamwork and created a feeling of oneness which thus motivated us to perform to the best our ability and create a report of the highest quality.

This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Date: 01|06|12

Sneh Srijan (081045)

Dikshit Chhabra (081055)

Nishant Sharma (081059)

TABLE OF CONTENTS

List of Figures

Abstract

In today's world managing aquariums have become increasingly hectic. People have aquarium in their homes and shops, but they don't really find time to manage the environment for fishes inside the aquarium. As a result fishes die inside the aquarium.

This project, "AQUARIUM MANAGEMENT" is a step towards the automatic sensing and controlling of various aspects related to the aquarium environment, necessary for the healthy growth of fishes.

Through this project we will be focusing on sensing the:-

1. Temperature of aquarium
2. pH of water inside the aquarium
3. Lighting conditions for the aquarium

In the future we will be focusing on "Microcontroller AT89C51" to control these aspects. Whenever the temperature of aquarium reaches out of bound, the buzzer starts the alarm and the fan connected to microcontroller through relay gets ON. Similarly the pH and lighting conditions will also be controlled through this project.

Chapter 1: INTRODUCTION

An aquarium (plural *aquariums* or *aquaria*) is a vivarium consisting of at least one transparent side in which water-dwelling plants or animals are kept. The term combines the Latin root '*aqua*', meaning water, with the suffix –'*arium*', meaning "a place for relating to".

An aquarist owns fish or maintains an aquarium, typically constructed of glass or high strength acrylic plastic. Cuboid aquaria are also known as fish tanks or simply tanks, while bowl-shaped aquaria are also known as fish bowls.

A number of environmental factors including light and temperature affect fish culture. The temperature of water has profound effect because fish cannot breed above or below the critical temperature limits. Temperature between 24°C and 33°C is found to be the best to induce spawning in fishes. This particular temperature range is also necessary for the healthy growth of nursery fish fries (young fishes). Rise of water temperature due to sunlight may adversely affect the fish rearing process.

pH requirement for the healthy growth of fishes is between 6.5 – 7.5 . The fish are accustomed to a certain H+ concentration. If suddenly, they face new water coming in their tank, which only contains one hundredth of the normal H+. This is a shock by itself, a very serious shock. While in stress, the fish tries to adapt to the new situation. No organism can adapt to this sort of fluctuations!! Hence, this needs to be taken care of.

Proper light conditions also affect the fish culture and hence a provision for this should also be there.

1.1 Temperature Control

Fish are cold blooded creatures, they cannot create their own body temperature, and therefore the water around them will give their body heat to maintain a correct metabolism. Fish originate from all corners of the globe, in different parts of the world the fish will live in different temperature scales. They can tolerate a slight variance from the natural water temperature, but beyond that the fish will start to suffer which in turn will induce stress in the fish; this can then lead to fatalities in the aquarium as the fishes immune system will start to break down and bacterial infections can start to attack the fish.

Often, one of the more overlooked elements of an aquarium is its temperature. Aquarists are so concerned with water quality and the biological state of the aquarium that they do not always pay much attention to the temperature.

As most aquarists are aware, stress reduces the natural ability of an aquarium inhabitant to fight off disease and infection. It comes as no surprise that much of an aquarist's job centers around reducing sources of stress in the aquarium. Among other things, temperature of the aquarium can be a source of stress.

There are two major temperature situations that aquarists must try and avoid. The first is continual improper water temperature. Generally speaking, most aquarium inhabitants prefer water temperatures around 25°C. Typically, if water temperature increases above 30°C, or below 24°C, disaster is imminent. The second temperature situation to avoid is rapid temperature changes. This is particularly deceiving since it doesn't have to be temperature changes much greater that a few degrees Celsius. This is one of the reasons that new fish are usually introduced into the aquarium by slowly equilibrating their water temperature with that of the aquarium (i.e. you float the bag they come home in at the surface of the aquarium for a while).

1.2 pH Control

1.2.1 What is pH?

A pH meter is an electronic instrument used for measuring the pH (acidity or alkalinity) of a liquid (though special probes are sometimes used to measure the pH of semi-solid substances). A typical pH meter consists of a special measuring probe (a glass electrode) connected to an electronic meter that measures and displays the pH reading.

1.2.2 The pH loop

A pH measurement loop is made up of three components, the pH sensor, which includes a measuring electrode, a reference electrode, and a temperature sensor; a preamplifier; and an analyser or transmitter. A pH measurement loop is essentially a battery where the positive terminal is the measuring electrode and the negative terminal is the reference electrode. The measuring electrode, which is sensitive to the hydrogen ion, develops a potential (voltage) directly related to the hydrogen ion concentration of the solution. The reference electrode provides a stable potential against which the measuring electrode can be compared.

1.2.2.1 The probe

The pH measurement is the activity of hydrogen ions surrounding a thin-walled glass bulb at the tip of the probe. A pH sensing probe produces a small electrical voltage, usually about 0.06 volt per pH unit.

This voltage is measured and then displayed as pH units on your pH meter. Most pH probes consist of a body, an electrode, and a cable that connects it to the controller as you can see from the photo below (Fig. 1.1).

1.2.2.2 How does a pH probe work?

The readings of a pH probe can be affected by one of two possible ways. Your aquarium water's pH and the output signal from the probe's electrode, will both be changed with temperature differences. These two effects together or separately, can cause errors in the control of your pH measurements and in your calibrations. The pH probe has a chemical sensing element that has a thin membrane which allows H+ ions to pass through. The electrode of the probe is also filled with a neutral solution or an equal amount of H+ or OH- ions.

Whenever the pH probe is immersed in an acidic solution, H+ ions pass through the membrane. This causes a positive potential on the chemical sensing electrode to be developed.

This potential difference between the sensing electrode and the reference point is measured with a pH meter.

It is then displayed on your automatic pool controller as a pH read out. Now if your pH chemical probes are immersed in an alkaline solution, then a higher concentration of H+ ions will exists within the probe.

The concentration of ions inside the pH probe is more than that found outside the probe itself. This reading causes the H+ ions within the probe to pass through its membrane; thus, leaving an excess of OH- ions within the probe. This causes a negative potential to be sensed by your pH meter. Again this will be displayed on pH meter as a pH read out.

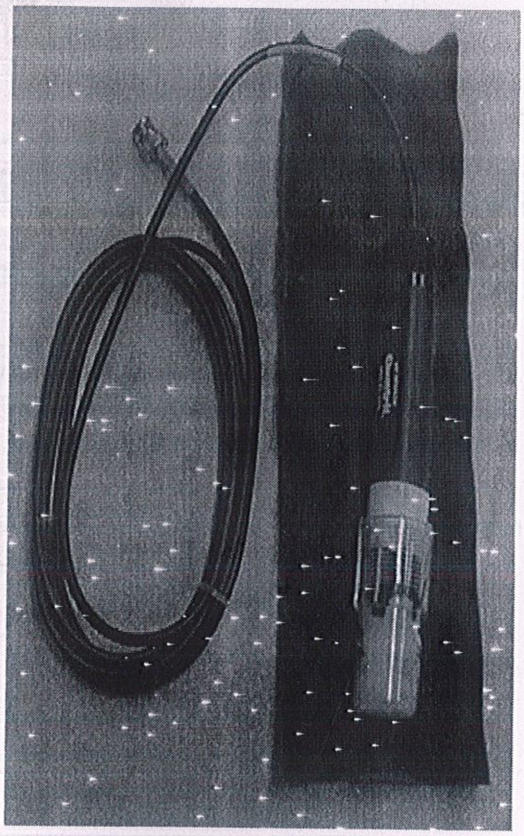


Fig. 1.1: pH PROBE

1.2.2.3 The Meter

The meter circuit is no more than a voltmeter that displays measurements in pH units instead of volts. The input impedance of the meter must be very high because of the high resistance—approximately 20 to 1000 MΩ — of the glass electrode probes typically used with pH meters. The circuit of a simple pH meter usually consists of operational amplifiers in an inverting configuration, with a total voltage gain of about −17. The inverting amplifier converts the small voltage produced by the probe (+0.059 volt/pH) into pH units, which are then offset by seven volts to give a reading on the pH scale. For example:

1 At neutral pH (pH 7) the voltage at the probe's output is 0 volts.

$$0 * 17 + 7 = 7$$

2 At basic pH, the voltage at the probe's output ranges from 0 to +0.41 volts

$$7 * 0.059 = 0.41V$$

So for a sample of pH 10 (3 pH units above neutral),

$$3 * 0.059 = 0.18 V$$

The output of the meter's amplifier is

$$0.18 * 17 + 7 = 10$$

3 At acid pH, the voltage at the probe's output ranges from –0.41 volts to 0. So for a sample of pH 4 (3 pH units below neutral),

$$-3 * 0.059 = -0.18 V$$

The output of the meter's amplifier is

$$-0.18 * 17 + 7 = 4$$

1.2.3 Nernst equation for a pH electrode

Nernst equation is a mathematical description of an ideal pH electrode behavior. Hermann Walther Nernst (1864 – 1941) was a German chemist that introduced in 1889 a well known equation, correlating chemical energy and the electric potential of a galvanic cell or battery.

1.2.3.1 The pH electrode potential

As explained on the pH definition page of this site, the pH value is defined as the negative logarithm of the H+ activity in a given solution ($pH = \log_a H+$).

1.2.3.2 Equation of a straight line

The equation $E = E_0 - kT*pH$ is the potential or voltage (millivolt; mV) relation of a pH electrode. It is the equation of a straight line. The slope factor is the term "kT" and it provides the amount of change in total potential (mV) for every change in pH unit.

1.2.3.3 Temperature

The equation $E = E_0 - kT*pH$ may be stated for any temperature. However, the slope or Nernst factor (kT) will change when temperature changes (T is not constant).

If T = 25 °C the equation will be: $E = E_0 - 0.0592*pH$

Let's draw the graph of this equation:

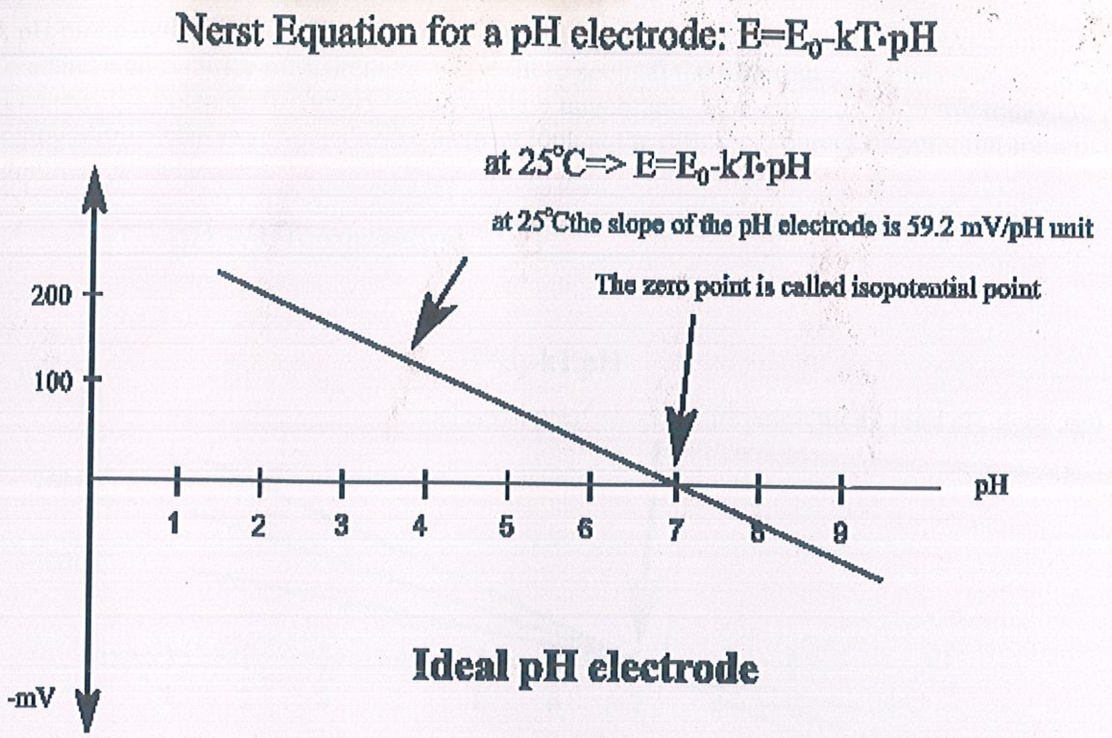


Fig. 1.2: Graph for Nernst Equation

1.2.3.4 The pH versus temperature characteristic of a pH sensor

The pH versus temperature relation of a pH sensor is described, according to Nernst equation, with the following equation:

$$E = E_0 - kT*pH$$

The slope factor of the equation, "kT", will change when temperature changes. And that mean the millivolt (mV) output of the pH electrode will change with temperature.

The figure above shows the effect on the electrode signal at various temperatures. The isothermal point of a pH electrode is defined as the intersection point of different temperature lines, see the figure. The isopotential point, or the zero point, is then point where electrode output is 0.0 mV.

For an ideal pH electrode, this would represent an isothermal intersection at the zero point (0 mV at pH 7) for different temperatures.

1.2.3.5 The different potentials

A pH probe delivers a sensitive electrical potential (voltage) with a number of sources of error. To achieve an accurate measurement you want to keep all the different potentials stable, except, of course, the potential of the pH sensitive glass membrane. Most pH sensors in use today are pH combination electrodes. So, let's take a closer look at the different sources of potential for a pH combination electrode.

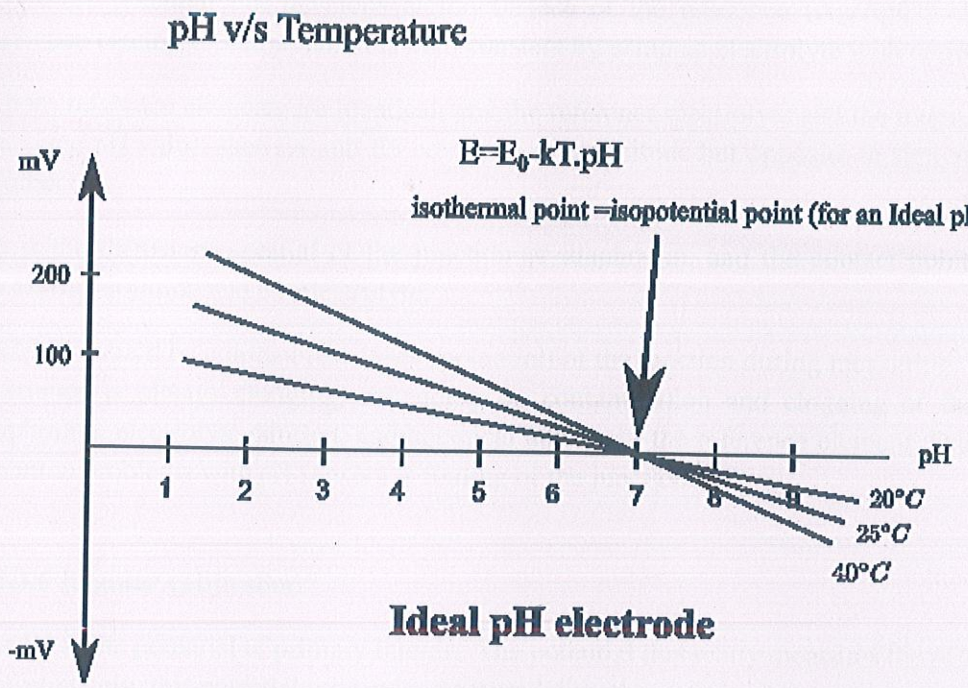


Fig. 1.3: pH v/s Temperature

Six different potentials form the total output voltage (mV) of a pH electrode:

E1 is the potential of the hydrogen sensitive glass - the real pH effect.

E2 is the asymmetry potential. When a pH electrode is in contact with a solution with the same pH as its internal buffer, there should be no potential difference across the glass membrane. However, in reality, there is a potential difference of a few millivolts, and it is known as the asymmetry potential.

Things that limit the ability of the ion exchange mechanism in the glass membrane will cause an asymmetry potential. For example, different gel layer thickness and on the thickness of the glass membrane itself.

E3 is the potential on the inside of the membrane. As the pH of the internal buffer solution does not change in value, this potential should be constant at all times.

E4 is the potential that develops at the surface of the reference (Ag/AgCl) element of the measuring electrode. The potential is kept constant by using an internal buffer with a constant pH value.

E5 is the potential that develops at the surface of the reference (Ag/AgCl) element of the reference electrode. The potential is kept constant by using an electrolyte with constant pH.

If both reference elements are identical, and the reference electrolyte and the internal buffer have the same pH value, then E4 and E5 are equal in magnitude but opposite in sign and cancel one another.

E6 is the diffusion potential of the junction or diaphragm, and the contact point between the measuring solution and the electrolyte.

It ideally would be constant resistance (potential) of the junction during measuring. However, the resistance is always changing, depending on contamination and clogging of the junction or diaphragm, electrolyte dilution, and chemical attacks of the reference element. One of the most common problems with pH sensors is fouling of the junction.

1.2.3.6 Regular calibration

The E1 is the potential of primary interest. The potential that really measures the pH value. Unfortunately, this potential cannot be measured directly.

What the pH meter (voltmeter) measures is a sum of all potentials:

$$U = E1 + E2 + E3 + E4 + E5 + E6$$

During ideal conditions, all the potentials E2 - E6 would be constant and E1 would be the only potential that vary during a pH measurement. However, in practice that's not the case, and the way we can compensate for these undesirable electrode potential fluctuations is regular calibration.

1.3 Light Control

A photoresistor is a light-dependent resistor (LDR) that covers the spectral sensitivity similar to that of the human eye. The active elements of these photoresistors are made of Cadmium Sulfide (CdS). Light enters into the semiconductor layer applied to a ceramic substrate and produces free charge carriers. A defined electrical resistance is produced that is inversely proportional to the illumination intensity. In other words, darkness produces high resistance, and high illumination produces very small amounts of resistance.

A photoresistor is made of a high resistance semiconductor. If light falling on the device is of high enough frequency, photons absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electron (and its hole partner) conduct electricity, thereby lowering resistance.

A photoelectric device can be either intrinsic or extrinsic. An intrinsic semiconductor has its own charge carriers and is not an efficient semiconductor, e.g. silicon. In intrinsic devices the only available electrons are in the valence band, and hence the photon must have enough energy to excite the electron across the entire bandgap. Extrinsic devices have impurities, also called dopants, added whose ground state energy is closer to the conduction band; since the electrons do not have as far to jump, lower energy photons (i.e., longer wavelengths and lower frequencies) are sufficient to trigger the device. If a sample of silicon has some of its atoms replaced by phosphorus atoms (impurities), there will be extra electrons available for conduction. This is an example of an extrinsic semiconductor. Photoresistors are basically photocells.

1.3.1 Uses for Light Dependent Resistors

Light dependent resistors are a vital component in any electric circuit which is to be turned on and off automatically according to the level of ambient light - for example, solar powered garden lights, and night security lighting.

An LDR can even be used in a simple remote control circuit using the backlight of a mobile phone to turn on a device - call the mobile from anywhere in the world, it lights up the LDR, and lighting (or a garden sprinkler) can be turned on remotely!

1.3.2 Using an LDR in the Real World

In a real world circuit, the LED (and resistor) between the positive voltage input (V_{in}) and the collector (C) of the transistor would be replaced with the device to be powered.

Typically a relay is used - particularly when the low voltage light detecting circuit is used to switch on (or off) a 240V mains powered device. When darkness falls (if the LDR circuit is configured that way around) the relay is triggered and the 240V device, for example a security light, switches on.

Chapter 2: Hardware Design

2.1 Temperature Control Design (BLOCK DIAGRAM)

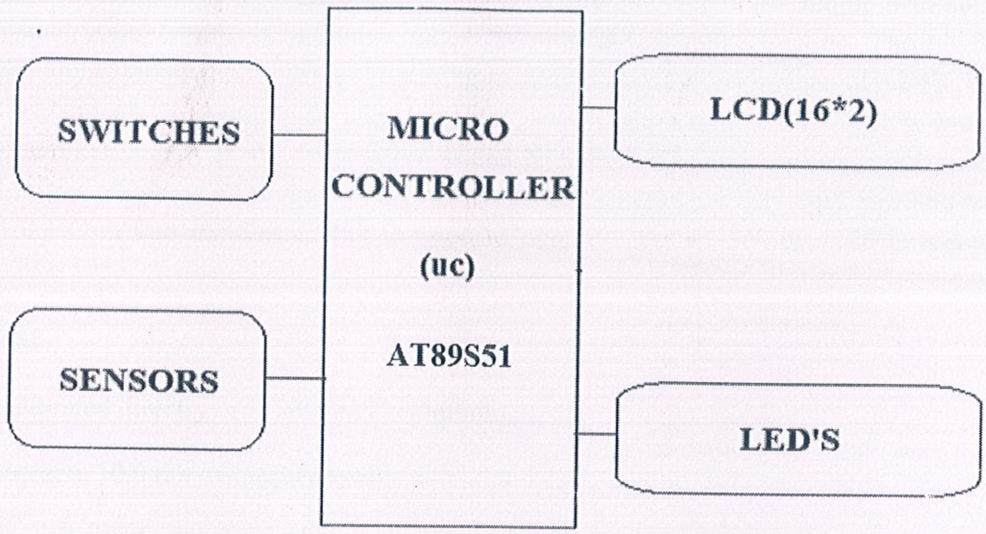


Fig. 2.1: Block Diagram

2.1.1 COMPONENTS USED

Temperature Sensor LM35

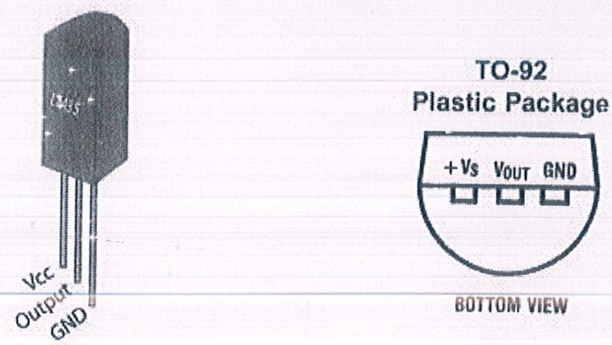


Fig. 2.2: LM-35

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 ±1⁄4°C at room temperature and ±3⁄4°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. 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.

Features:

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- Rated for full −55° to +150°C range
- Operates from 4 to 30 volts

8-Bit A/D Converter ADC 0804

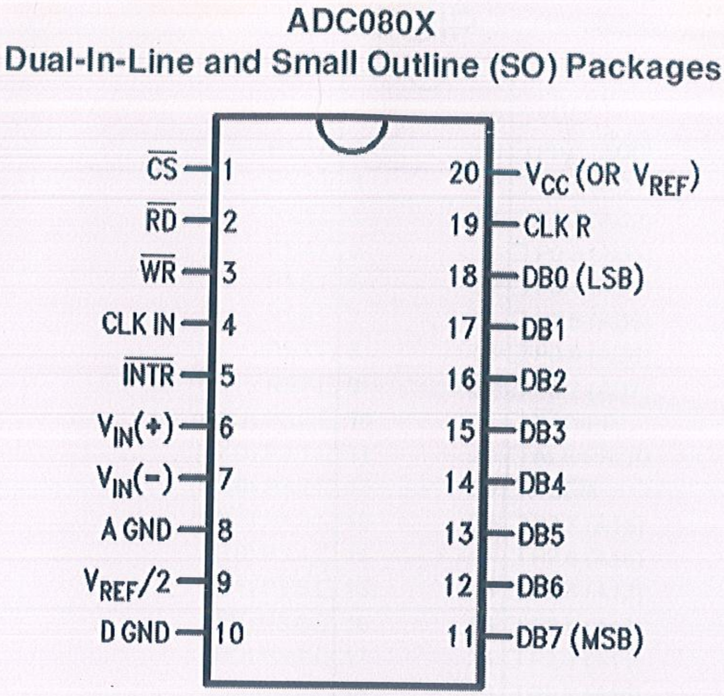


Fig. 2.3: ADC 0804

DESCRIPTION

The ADC080X family is CMOS 8-Bit, successive approximation A/D converters which use a modified potentiometric ladder and are designed to operate via three-state outputs. These converters appear to the processor as memory locations or I/O ports, and hence no interfacing logic is required. The differential analog voltage input has good common mode-rejection and permits offsetting the analog zero-input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

FEATURES

- Easy interface to all microprocessors
- 0V to 5V input range with single 5V power supply
- Outputs meet TTL voltage level specifications

Micro-Controller AT89C51

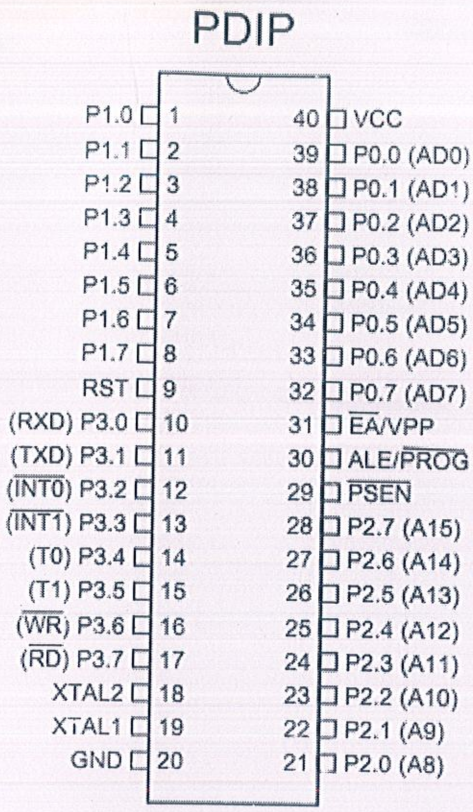


Fig. 2.4: AT89C51

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.

FEATURES:

- 4K Bytes of In-System Reprogrammable Flash Memory
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters

LCD

Fig. 2.5: 16x2 LCD Display

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

Pin No.	Symbol	Level	Description
1	V_{SS}	0V	Ground
2	V_{DD}	5.0V	Supply Voltage for logic
3	※V_{BT}	(Variable)	Operating voltage for PLED Brightness adjhstment
4	RS	H/L	H: DATA, L: Instruction code
5	R/W	H/L	H: Read(MPU→ Module) L: Write(MPU→ Module)
6	E	H,H→ L	Chip enable signal
7	DB0	H/L	Data bit 0
8	DB1	H/L	Data bit 1
9	DB2	H/L	Data bit 2
10	DB3	H/L	Data bit 3
11	DB4	H/L	Data bit 4
12	DB5	H/L	Data bit 5
13	DB6	H/L	Data bit 6
14	DB7	H/L	Data bit 7
15	NC	-	
16	NC	-	

Fig. 2.6: LCD pinout

Transformer

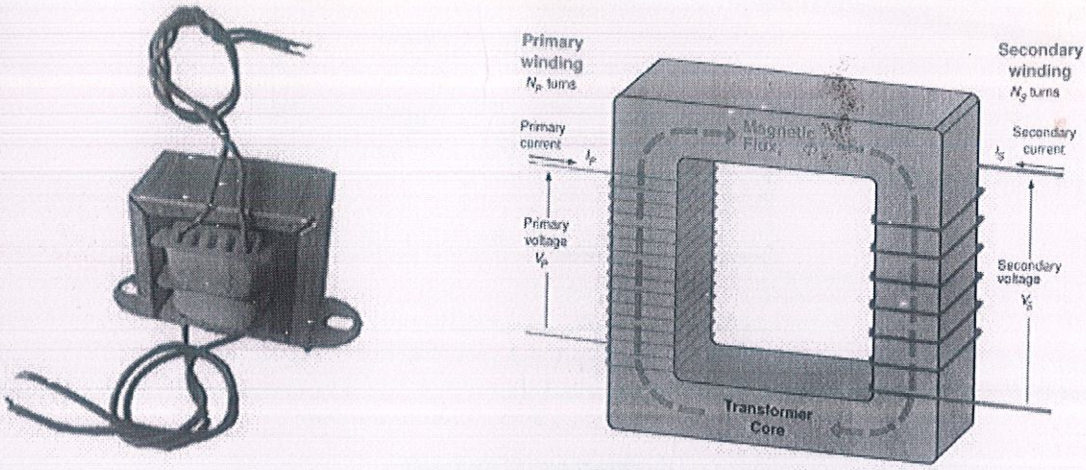


Fig 2.7: Transformer

A "transformer" changes one voltage to another. This attribute is useful in many ways. The number of times the wires are wrapped around the core ("turns") is very important and determines how the transformer changes the voltage.

- If the primary has fewer turns than the secondary, you have a step-up transformer that increases the voltage.
- If the primary has more turns than the secondary, you have a step-down transformer that reduces the voltage.

A transformer doesn't change power levels. If you put 100 Watts into a transformer, 100 Watts come out the other end. [Actually, there are minor losses in the transformer. But transformers come pretty close; perhaps 95% efficient.]

A transformer is made from two coils of wire close to each other (sometimes wrapped around an iron or ferrite "core"). Power is fed into one coil (the "primary"), which creates a magnetic field. The magnetic field causes current to flow in the other coil (the "secondary"). Note that this doesn't work for direct current (DC): the incoming voltage needs to change over time - alternating current (AC) or pulsed DC.

Rectifier

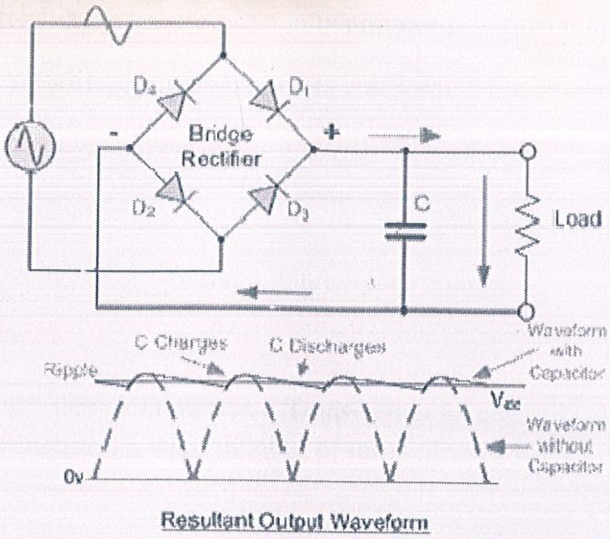


Fig. 2.8: Bridge Rectifier

A rectifier changes alternating current into direct current. This process is called rectification. The three main types of rectifier are the half-wave, full-wave, and bridge. A rectifier is the opposite of an inverter, which changes direct current into alternating current.

The bridge rectifier, also called a diode bridge, consists of four diodes connected together in a square. Two diodes are connected at their anodes, and the other two are connected at their cathodes. These form the rectified output terminals. The remaining ends are joined to form two input terminals. It is usually packaged as one component with four terminals. The bridge rectifier allows for full-wave rectification without the need for an earthed center tap on the transformer.

Even the bridge rectifier has some variation in its output voltage, so a filter is required to smooth out this ripple. A capacitor connected across the output terminals acts as a basic filter by storing energy during the peak voltage, and releasing it when the voltage falls. This removes most of the ripple but does not result in a steady voltage. A choke and second capacitor are usually added to further smooth the ripple.

Power Supply

To provide a useable low voltage the Power supply unit needs to do a number of things:-

- Reduce the Mains AC (Alternating current) voltage to a lower level.
- Convert this lower voltage from AC to DC (Direct current)
- Regulate the DC output to compensate for varying load (current demand)
- Provide protection against excessive input/output voltages.

Reduction of AC Mains

This is achieved by using a device known as a Transformer an electromagnetic device consisting of an ferrous iron core which has a large number of turns of wire wound around it, known as the Primary Winding

The ends of the turns of wire are being connected to the input voltage (in this case Mains AC).

A second number of turns of wire are wound around the Primary Winding, this set being known as the Secondary Winding.

The difference between the number of turns provides us with a way of reducing (in our case) a high AC voltage to a lower one.

Conversion of AC to DC

To convert our now low AC voltage to DC we use a Rectifier Diode connected to the Secondary Winding.

As our low AC voltage will be working at a frequency of 50Hz (Mains AC frequency) it is desirable to reduce the inherent hum on this to a lower level.

This is achieved by a technique known as Smoothing ("Ironing" out the bumps in the AC).

A simple way to reduce the hum is to use Full Wave Rectification.

Today this is usually done by four diodes in a bridge configuration known as a Bridge Rectifier. (This can be four individual diodes or a dedicated self contained package)

Regulation of Output Voltage

The Electrolytic Capacitor is a device capable of storing energy the amount of energy and the time it remains stored depending on the value.

In a simple POWER SUPPLY UNIT the easiest way to provide regulation to compensate for varying load conditions is to use a pair of relatively high value Electrolytic Capacitors.

Their values in this case being in the region of 470uF to 2000uF depend on the application and the amount of current required from the output of the unit.

One of these capacitors is connected across the DC output of the rectifier diode(s) or bridge, this capacitor also providing an extra degree of smoothing the output waveform.

The second capacitor is connected via a low value, medium to high wattage resistor, which assists in limiting the current demand.

Protection against excessive voltages

In a simple POWER SUPPLY UNIT the easiest way to do this is by providing fuses at the input to the transformer, generally in the live side of the mains supply, also at the DC outputs.

In the event of an excessive input voltage, or excessive current being drawn from the output, one of these fuses should normally blow protecting the POWER SUPPLY UNIT and the equipment connected to it.

The transformer may also be fitted with an internal or external thermal fuse, which will open if the transformer becomes hot due to the aforementioned conditions.

2.2 pH Control Design

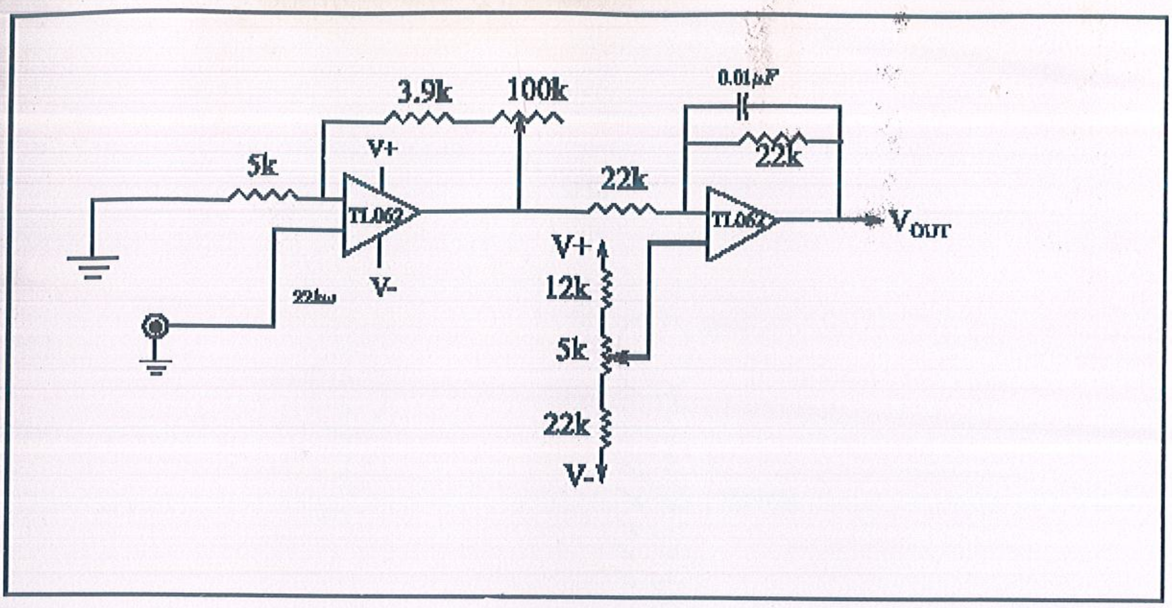


Fig. 2.9: pH Control

A pH probe works something like a little battery.

In a neutral solution pH probe would generate 0V.

In acidic solution pH probe would generate a positive voltage proportional to the ionic strength.

In basic solution pH probe would generate a negative voltage proportional to the ionic strength.

This circuit uses op-amp U1a to amplify and buffer the high impedance signal from the probe and adjust the slope, which is done by adjusting the gain loop. The output of this stage if adjusted properly would be -7V to +7V with -7V being pH of 14 and +7V being pH of 0.

What is needed for this voltage to be inverted and offset, and both these tasks are accomplished using op-amp U1b. The output at this stage should be 0V to 14V indicating the pH of the sample being tested.

2.3 Light Control Design

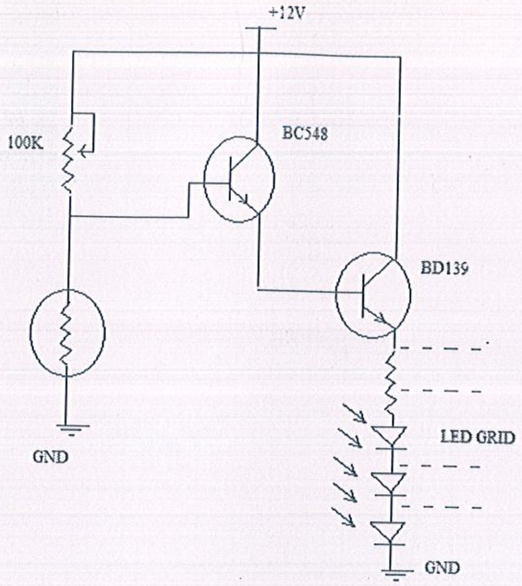


Fig. 2.10: Light Control

- The resistance of the Light Dependent Resistor (LDR) varies according to the amount of light that falls on it.
- $R_L = (500 / Lux)$ Kohm
- With the LDR connected to 5V through a 3.3K resistor, the output voltage of the LDR is
- $Vo = 5*R_L / (R_L+3.3)$

2.4 CONSTRUCTION/WORKING OF THE CIRCUIT

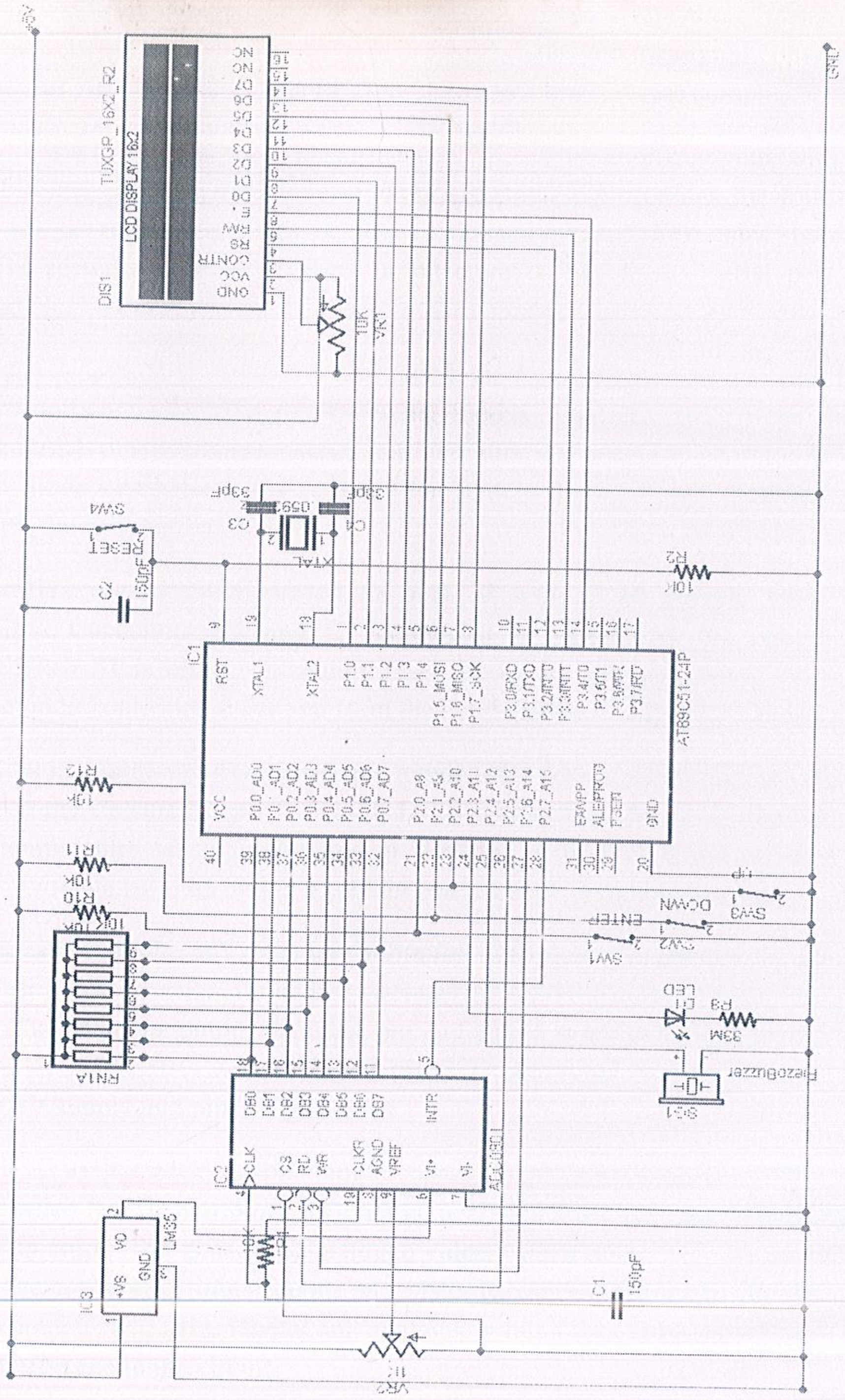


Fig. 2.11: Temperature Control (Full)

DESCRIPTION:

Fig. 2.11 shows the circuit of the microcontroller- based temperature meter. It comprises microcontroller AT89C51, temperature sensor LM35, analogue to- digital converter ADC0804, voltage regulator 7805 (Fig. 2.12), an LCD module and a few discrete components.

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4 kB of Flash programmable and erasable read only memory (PEROM). It provides the following standard features: 4 kB of flash, 128 bytes of RAM, 32 input/output (I/O) lines, two 16-bit timer/ counters, five-vector two-level interrupt architecture, a full-duplex serial port, and 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, timers/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.

An 11.0592MH2 crystal is connected to pins 18 and 19 to provide basic clock to the microcontroller. Capacitors C4 and C5 connected in parallel to the crystal maintain the resonance. Switch S1 is used to manually reset the microcontroller, while the power-on reset signal for the microcontroller is derived from the combination of capacitor C3 and resistor R2.

IC LM35 (IC3) is a three-terminal, precision temperature sensor whose output voltage is linearly proportional to the Celsius temperature with 110.0 mV/"C scale factor. It 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. It is rated for full -55oC to 150.C range and operates off 4V-30V input. It gives 0V output for 0°C temperature. The analogue output (Vout) at pin 2 of LM35 is fed to Vin (+) pin 6 of analogue-to-digital converter ADC0804, whose V,, (-) plr:r7 is connected to ground. Pin i. of LM35 is connected to 5V supply and pin 3 is grounded.

ADC0804 (IC2) is a CMOS, 8-bit, single-channel analogue-to-digital converter. It features conversion time of less than 100 ms, differential analogue input voltage, TTL-compatible inputs and outputs, on-chip clock generator, analogue voltage input range from 0V to 5V, and no zero adjustment. The conversion time depends on resistor R3 and capacitor C6. The conversion rate in free-running mode is 640 kHz. Digital and analogue ground should be separated in ADC0804 to avoid any interference in the circuit.

The resolution of 8-bit ADC0804 is 19.53 mV, which doesn't match with the scale factor of LM3S and therefore can cause error. To avoid this error, the full-scale range of ADC0804 is made 0-256V by adjusting the voltage at pin 9 ($V_{ref}/2$) to 1.28V through

1-kilo-ohm preset VR2. In ADC0804, the input analogue voltage is divided by its step size to give digital output. For each 10mV rise and fall of the analogue input at Vin (+), digital outputs at DBO throughDB7 increase and decrease respectively. The maximum input voltage that can be converted by the ADC is 2.55V (10mV x 2SS) giving full-scale output of FF hex value in this system.

The 8-bit digital output of ADC0804 (DB0 through DB7) is connected to 8-bit port p0 of the microcontroller. Signals RD, WR and INTR of the ADC are connected to P27, P2.6 and P2.5 of the microcontroller. These signals of the ADC act as handshaking signals with microcontroller IC1.RD and WR are the input pins of the ADC, while INTR is the output pin. Through INTR signal, the microcontroller gets to know when the conversion from analogue into digital is completed by the ADC.

The microcontroller makes WR Pin 'low' and RD pin 'high' to start the conversion. Pin INTR goes high for the end of conversion. A transition from high to low on INTR indicates end of conversion. Then the microcontroller makes RD 'low' and WR 'high' to read the 8-bit data at DBO through DB7 through microcontroller port P0. Through its firmware, the microcontroller multiplies the digital input at port 0 with the step size value of ADC0804 and then divides with the temperature/volt scale factor of LM35 to give the measured and calibrated °C temperature.

The measured temperature is instantaneously displayed on the LCD. Port Pl, of the microcontroller is connected to data port pins 7 through 14 of the LCD module. The handshake signals of the LCD (RS, R/W and Enable) are connected to P3.2, P3.3 and.P3.4 of the microcontroller, respectively. All the data is sent to the LCD in ASCII form to display. Only the commands are sent in hex form to the LCD. RS signal is used to distinguish between data (RS=1) and command (RS=O). Use Preset VR1 to control the contrast of the LCD.

Switches S2, S3 and S4 connected to pins P2.2, P2.1 and P2.0 are treated as 'Up', 'down' and enter, buttons, respectively by the temperature setting of the microcontroller. Measuring interval and threshold temperature values are to be entered by the user using switches S2, S3 and S4 whenever the microcontroller starts. The range of measuring interval is 1 second to 99 seconds.

According to this measuring interval, the microcontroller measures the temperature value in its flash memory. Whenever the measured temperature rises above the threshold value, the piezoelectric buzzer at pin P2.4 sounds until the temperature becomes lower than these threshold values.

Time between measurements of two temperature samples is treated as waiting time measuring interval. So during waiting time, LED2 connected at pin P2.3 of the microcontroller glows. It is turned off during measurement. Resistor R4 acts as the current limiter for LED2.

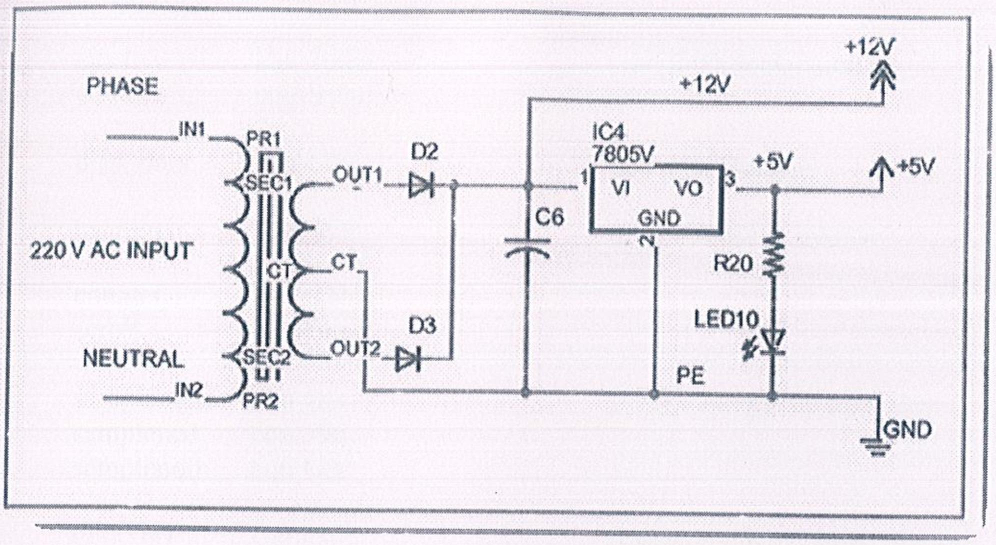


Fig. 2.12: Power Supply Circuit

Fig. 2 shows the power supply circuit. The 230V, 50 Hz AC mains is stepped down by transformer X1 to deliver secondary output of 9V 500mA.

The transformer output is rectified by a full-wave rectifier comprising diodes D1 through D4, filtered by capacitor C1 and regulated by IC LM7805 to provide +5V DC output. Capacitor C2 provides further filtering. LED1 indicates DC power and R1 acts as the current limiter.

Chapter 3: Software Design

```
;***************************Temperature measurement system**********************

;********A51 Assmbler used to convert assembly code to machine code*******************

        $MOD51

        aux1        equ  0a2h            ;variables declaration
        count       equ  10h

        disp_data   equ 11h              ;lcd data variable

        lcdcom      equ  12h             ;lcd command variable

        counter     equ  13h
        scroll      equ 14h

        sampleno2   equ 18h              ;3 digit sample no variables
        sampleno1   equ 19h
        sampleno0   equ 1ah

        interval1   equ 21h              ;2 digit recording interval
        interval0   equ 22h

        threshold1  equ 23h              ;2 digit threshold value in degree Celsius
        threshold0  equ 24h

        tval1       equ 25h              ;2 digit temperature value in degree Celsius
        tval0       equ 26h
        temp        equ 29h

        t3          equ 38h              ;temporary variables
        t2          equ 39h
        t11         equ 3ah
        t00         equ 3bh
        hex1        equ 3ch
        hex0        equ 3dh
        m641        equ 3eh
        m640        equ 3fh
        out7        equ 40h
        out6        equ 41h
        out5        equ 42h
        out4        equ 43h
        out3        equ 44h
        out2        equ 45h
```

```
out1        equ 46h
out0        equ 47h
TMP_3               equ 48h
TMP_2               equ 49h
TMP_1               equ 4ah
TMP_0               equ 4bh
rem1        equ 4ch
rem0        equ 4dh

Ekey        equ 0a0h                    ;input keys address location
dkey        equ 0a1h
ukey        equ 0a2h
led1        equ 0a3h

buzzer      equ 0a4h                    ;buzzer location

intr        equ 0a5h                    ;ADC handle shake signals address location
wrr         equ 0a6h
rdd         equ 0a7h

adc_data    equ 80h                     ;ADC data 1 byte bus address

org 0000h

start_program:

        mov sp,#50h

        clr led1                        ;reset leds and buzzer
        clr buzzer

        mov count,#00h

        mov counter,#00h

        lcall lcd_initialize            ;initialize lcd

        lcall delay14m

        lcall welcome_msg               ;welcome strings display

        lcall clear_lcd

        mov dptr,#0730h                 ;string "Enter parameters" display on LCD
        lcall display_text

        lcall delay1sec
```

```
        lcall cursoron                  ;display cursor on LCD

        clr ri

        mov interval1,#30h

        mov interval0,#31h              ;Initialize recording interval to 01 sec

        lcall clear_lcd                 ;clear the LCD

        mov dptr,#0740h                 ;string "measuring interval" display on lcd"
        lcall display_text

        lcall delay14m

        mov lcdcom,#0c5h                ;set cursor position

        lcall command                   ;routine to command the LCD

        mov disp_data,interval1

        lcall data_routine              ;routine to display a ASCII character on the
current LCD cursor position

        mov disp_data,interval0
        lcall data_routine
        mov disp_data,#' '
        lcall data_routine
        mov disp_data,#'s'
        lcall data_routine
        mov disp_data,#'e'
        lcall data_routine
        mov disp_data,#'c'
        lcall data_routine

again_keyi:

        mov lcdcom,#0c6h

        lcall command

        setb ukey                       ;Scanning the input keys
        setb dkey
        setb Ekey

upi:
```

```
        jb ukey,downi                   ;UP key

        mov a,#01h

        lcall intfound                  ;routine to keep measuring interval in range
from 01 to 99 secs

        sjmp Enteri

    downi:

        jb dkey,Enteri                  ;DOWN key
        mov a,#02h
        lcall intfound

    Enteri:

        jb Ekey,again_keyi              ;Enter key

        lcall delay140m                         ;debouncing delay

        lcall delay140m


        clr ri

        mov threshold1,#32h             ;initialize threshold value of temperature to
20 degree celcius

        mov threshold0,#30h

        lcall clear_lcd

        mov dptr,#0750h                 ;string "threshold value" display on LCD
        lcall display_text
        mov lcdcom,#0c5h
        lcall command
        mov disp_data,threshold1
        lcall data_routine
        mov disp_data,threshold0
        lcall data_routine
        mov disp_data,#' '
        lcall data_routine
        mov disp_data,#'C'
        lcall data_routine
        nop
```

```
again_keyt:

        mov lcdcom,#0c6h
        lcall command
        setb ukey
        setb dkey
        setb Ekey

upt:

        jb ukey,downt
        mov a,#01h

        lcall thfound                   ;routine to  keep threshold value from 20 to
49 degree celcius

        sjmp Entert

downt:

        jb dkey,Entert
        mov a,#02h
        lcall thfound

Entert:

        jb Ekey,again_keyt
        lcall delay140m
        lcall delay140m

        lcall clear_lcd

        mov dptr,#0760h                 ;string "Parameter saved" display on LCD

        lcall display_text

        lcall delay1sec

start_sample:

        mov sampleno0,#30h              ;initializing no of samples to zero
        mov sampleno1,#30h
        mov sampleno2,#30h

        lcall clear_lcd

        mov dptr,#0770h                 ;string "temperature" display on LCD
```

```
        lcall display_text
        mov lcdcom,#0c0h
        lcall command                 ;string "samples no." dispaly on LCD
        mov dptr,#0780h
        lcall display_text
        lcall cursoroff               ;make cursor off
    sample_again:
        nop
    no_Ser:
        lcall start_adc               ;start the adc to measure temp
        lcall read_adc                ;read adc byte
        lcall get_temp                ;convert adc byte into temperature and
record into the flash
        lcall update_sample           ;increase no of samples by one
        lcall show_temp               ;display temp on lcd
        lcall check_thsold            ;compare the temperature with the threshold
value
        lcall wait_time               ;waiting time according to recording interval
        ljmp sample_again


;***********************************SUBROUTINES***************************
    start_adc:                        ;start the conversion of ADC..conversion
frequency is around 640KHz

        clr led1
        clr wr
        lcall delay14m
        setb wr
```

```
        setb intr
        ret
;******************************************************************************
read_adc:                               ;read the digital output from ADC

        mov adc_data,#0ffh

adc_not:

        ;jb intr,adc_not
        lcall delay14m
        setb rdd
        clr rdd
        mov a,adc_data
        setb intr
        setb rdd
        mov temp,a
        ret
;******************************************************************************
get_temp:                               ;convert adc digital output into temperature
and store its 2 digit value into flash

        lcall calculations

        mov tval1,out2                          ;current temp value in tval1 and tval0
        mov tval0,out1

gett_over:

        ret
;******************************************************************************
update_sample:                          ;routine increase number of samples by one

        anl sampleno0,#0fh
        anl sampleno1,#0fh
        anl sampleno2,#0fh

        mov t3,#00h
        mov t2,sampleno2
        mov t11,sampleno1
        mov t00,sampleno0
```

```
        lcall dec_to_hex                    ;convert decimal to hexadecimal value

        clr c

        mov a,hex0                          ;add one to hex value of no of samples
        add a,#01h
        mov hex0,a
        mov a,hex1
        addc a,#00h
        mov hex1,a

        mov t3,#00h
        mov t2,#00h
        mov t11,hex1
        mov t00,hex0

        lcall hex_to_dec                    ;convert hex value to decimal number

        mov sampleno2,out2
        mov sampleno1,out1
        mov sampleno0,out0
        ret

;***************************************************************************

    show_temp:                              ;routine to display 2 digit temperature to
LCD

        mov lcdcom,#8bh
        lcall command
        mov disp_data,#':'
        lcall data_routine
        mov disp_data,tval1
        lcall data_routine
        mov disp_data,tval0
        lcall data_routine
        mov disp_data,#' '
        lcall data_routine
        mov disp_data,#'C'
        lcall data_routine
        mov lcdcom,#0cbh
        lcall command
        mov disp_data,#':'
        lcall data_routine
        mov disp_data,sampleno2
        lcall data_routine
```

```
mov disp_data,sampleno1
lcall data_routine
mov disp_data,sampleno0
lcall data_routine
ret

;********************************************************************************

check_thsold:                    ;routine compare temp value with threshold
values

mov a,tval1
anl a,#0fh
mov b,#0ah
mul ab
mov r0,a
mov a,tval0
anl a,#0fh
add a,r0

mov r1,a                         ;hex value of current temperature
mov a,threshold1
anl a,#0fh
mov b,#0ah
mul ab
mov r0,a
mov a,threshold0
anl a,#0fh

add a,r0                         ;hex value of threshold temperature
clr c
subb a,r1

jnc temp_notmore

temp_more:

setb buzzer                      ;Buzzer on if current temperature more than
threshold value

sjmp check_over

temp_notmore:

clr buzzer                       ;Buzzer off if current temperature more than
threshold value
```

```
check_over:

        ret

;*****************************************************************************

wait_time:                          ;This routine is used to provide measuring interval delay

        clr wrr
        clr ri
        setb led1
        mov a,interval1
        anl a,#0fh
        cjne a,#00h,not_zero
        sjmp lsb_digit

not_zero:

        mov b,#0ah
        mul ab
        mov r0,a
        setb Ekey

hererO:

        mov r3,#04h

here1_i1:

        mov r4,#0E0h

here2_i2:

        mov r5,#0ffh

here3_i3:

        nop
        nop
        nop
        djnz r5,here3_i3
        djnz r4,here2_i2
        djnz r3,here1_i1
        djnz r0,hererO

lsb_digit:
```

```
        mov a,interval0
        anl a,#0fh
        cjne a,#00h,not_zero1
        sjmp wait_over

    not_zero1:

        mov r0,a

    herer01:

        mov r3,#04h

    here1_ii1:

        mov r4,#0e0h

    here2_ii2:

        mov r5,#0ffh

    here3_ii3:

        nop
        djnz r5,here3_ii3
        djnz r4,here2_ii2
        djnz r3,here1_ii1
        djnz r0,herer01

    wait_over:

        clr led1
        lcall delay140m
        lcall delay140m
        setb wrr
        ret

;*****************************************************************************

    welcome_msg:                    ;This routine displays the welcome message at
power on

        nop

    scrolling:                              ;startting cursor position from where
scrolling starts
```

```
        mov scroll,#8fh

    scrollagain:

        lcall clear_lcd
        lcall cursoroff
        mov lcdcom,scroll
        lcall command


        mov dptr,#700h                          ;string "Major project"          display
on LCD
        lcall display_text
        mov a,scroll
        dec a
        mov scroll,a
        lcall delay140m
        cjne a,#7fh,scrollagain
        lcall delay1sec
        lcall delay1sec
        lcall clear_lcd
        mov r2,#00h

    flash_again:

        mov dptr,#710h                          ;string "Data acquisition" dispaly on
LCD
        lcall display_text
        lcall delay14m
        mov lcdcom,#0c0h
        lcall command

        mov dptr,#0720h                    ;string "System" display on second line of
LCD
        lcall display_text
        lcall delay1sec
        lcall delay140m
        lcall clear_lcd
        lcall delay140m
        lcall delay140m
        lcall delay140m
        inc r2
        mov a,r2

        cjne a,#02h,flash_again                 ;message on LCD blinks fro 2 times
```

```
ret
;*****************************************************************************

calculations:                         ;routine converts the adc 8 bit output to 2-
digit temperature value in ASCII

        mov a,temp
        mov b,#0ah
        div ab
        mov out2,a
        mov out1,b
        orl out2,#30h
        orl out1,#30h
        ret

;*****************************************************************************

division_16:                          ;32 bit by 16 bit division

    MOV    R7,#0                      ;msb remiander

    MOV    R6,#0                          ;lsb remiander ;zero out partial
remainder

    MOV    TMP_0,#0
    MOV    TMP_1,#0
    MOV    TMP_2,#0
    MOV    TMP_3,#0
    MOV    R1,hex1                    ;load divisor
    MOV    R0,hex0

    MOV    R5,#32                     ;loop count

Div_loop:

    LCALL   Shift_D                   ;shift the dividend and return MSB in C

    MOV    A,R6                       ;shift carry into LSB of partial remainder
    RLC    A
    MOV    R6,A
    MOV    A,R7
    RLC    A
    MOV    R7,A
    CLR    C

    MOV    A,R7                       ;subtract R1 from R7 to see if R1 < R7
```

```
        SUBB    A,R1                         ; A = R7 - R1, carry set if R7 < R1
        JC      Cant_sub
        JNZ     Can_sub                      ;jump if R7>R1
          CLR     C
        MOV     A,R6
        SUBB    A,R0                         ; A = R6 - R0, carry set if R6 < R0
        JC      Cant_sub


Can_sub:

        CLR     C
        MOV     A,R6
        SUBB    A,R0                         ; A = R6 - R0
        MOV     R6,A
        MOV     A,R7
        SUBB    A,R1                         ; A = R7 - R1 - Borrow
        MOV     R7,A
        SETB    C                            ; shift a 1 into the quotient
        ljmp    Quot


Cant_sub:

          CLR     C

Quot:

          LCALL    Shift_Q          ; Test for completion
        DJNZ    R5,Div_loop

                              ; Now we are all done, move the TMP values back
into OP

        MOV     t00,TMP_0
        MOV     t11,TMP_1
        MOV     t2,TMP_2
        MOV     t3,TMP_3
        RET

Shift_D:

        CLR     C
        MOV     A,t00
```

```
        RLC     A
        MOV     t00,A
        MOV     A,t11
        RLC     A
        MOV     t11,A
        MOV     A,t2
        RLC     A
        MOV     t2,A
        MOV     A,t3
        RLC     A
        MOV     t3,A
        RET

Shift_Q:

        MOV     A,TMP_0
        RLC     A
        MOV     TMP_0,A
        MOV     A,TMP_1
        RLC     A
        MOV     TMP_1,A
        MOV     A,TMP_2
        RLC     A
        MOV     TMP_2,A
        MOV     A,TMP_3
        RLC     A
        MOV     TMP_3,A
        RET

;*****************************************************************************

        hex_to_dec:                     ;Routine convert 4 byte hex into 8 digit decimal
ascii

                mov out7,#00h           ;Input in t3,t2,t1,t0 and output in
out7,out6,out5,out4,out3,out2,out1,out0 variables

                mov out6,#00h
                mov out5,#00h
                mov out4,#00h
                mov out3,#00h
                mov out2,#00h
                mov out1,#00h
                mov out0,#00h

                mov hex1,#00h
```

```
        mov hex0,#0ah

        mov a,t2
        cjne a,#00h,xbn56
        mov a,t11
        cjne a,#00h,xbn57
        lcall division_16
        mov out0,r6
        lcall division_16
        mov out1,r6
        mov out2,t00
        sjmp over56

xbn57:

        lcall division_16
        mov out0,r6
        lcall division_16
        mov out1,r6
        lcall division_16
        mov out2,r6
        lcall division_16
        mov out3,r6
        mov out4,t00
        sjmp over56

xbn56:

        lcall division_16
        mov out0,r6
        lcall division_16
        mov out1,r6
        lcall division_16
        mov out2,r6
        lcall division_16
        mov out3,r6
        lcall division_16
        mov out4,r6
        lcall division_16
        mov out5,r6
        lcall division_16
        mov out6,r6
        mov out7,t00

over56:
```

```
        orl out7,#30h           ;converting all decimal values to ASCII
        orl out6,#30h
        orl out5,#30h
        orl out4,#30h
        orl out3,#30h
        orl out2,#30h
        orl out1,#30h
        orl out0,#30h
        ret

;******************************************************************************

    dec_to_hex:                 ;Routine convert the 4 digit decimal into 2
byte hex

        mov r0,#00h             ;Input in t3 t2 t1 t0 and output in hex1 hex0
        mov r1,#00h
        mov r2,#00h
        mov r3,#00h
        mov r4,#00h
        mov r5,#00h
        mov r6,#00h
        mov r7,#00h
        mov hex1,#00h
        mov hex0,#00h
        mov r0,t3
        cjne r0,#00h,load0
        ljmp load1

    load0:

        mov a,r0
        mov b,#0ah
        mul ab
        mov r0,a
        mov r7,a
        mov m641,#64h

    load1:

        mov r2,t2
        cjne r2,#00h,load2
        ljmp load3

    load2:
```

```
        mov m640,#64h

load3:

        mov r4,t11
        cjne r4,#00h,load5
        ljmp load6

load5:

        mov a,r4
        mov b,#0ah
        mul ab
        mov r4,a
        mov r6,a

load6:

        mov r5,t00
        cjne r5,#00h,starting
        nop

starting:

        mov r5,t00
        cjne r5,#00h,here5
        ljmp now2

here5:

        inc hex0
        djnz r5,here5

now2:

        mov a,r6
        mov r4,a
        cjne r4,#00h,here4
        ljmp now3

here4:

        inc hex0
        djnz r4,here4

now3:
```

```
        mov r2,t2
        cjne r2,#00h,here3
        ljmp now4

here3:

        mov r3,m640

here2:

        clr c
        mov a,hex0
        add a,#01h
        mov hex0,a
        mov a,hex1
        addc a,#00h
        mov hex1,a

cont:

        djnz r3,here2
        djnz r2,here3

now4:

        mov a,r7
        mov r0,a
        cjne r0,#00h,here1
        ljmp home

here1:

        mov r1,m641

here0:

        clr c
        mov a,hex0
        add a,#01h
        mov hex0,a
        mov a,hex1
        addc a,#00h
        mov hex1,a

cont1:

        djnz r1,here0
```

```
        djnz r0,here1
home:
        ret
;*****************************************************************************
thfound:                                ;routine provide threshold value range from
20 to 49 degree celcius for user
        mov r6,a
        cjne r6,#01h,dec_th
        inc threshold0
        mov a,threshold0
        cjne a,#3ah,ter_return
        mov a,threshold1
        cjne a,#34h,notninerth
        mov threshold1,#32h
        mov threshold0,#30h
        sjmp ter_return
notninerth:
        inc threshold1
        mov threshold0,#30h
        sjmp ter_return
dec_th:
        cjne r6,#02h,ter_return
        dec threshold0
        mov a,threshold0
        cjne a,#2fh,ter_return
        mov a,threshold1
        cjne a,#32h,notzerorth
        mov threshold1,#34h
        mov threshold0,#39h
        sjmp ter_return
notzerorth:
        dec threshold1
        mov threshold0,#39h
ter_return:
```

```
        mov lcdcom,#0c5h
        lcall command
        mov disp_data,threshold1
        lcall data_routine
        mov disp_data,threshold0
        lcall data_routine

        lcall delay140m                         ;debouncing delay
        lcall delay140m
        lcall delay140m
        ret

;*****************************************************************************

    intfound:                                   ;routine provide recording interval range
from 01 to 99 secs for user

        mov r6,a
        cjne r6,#01h,dec_int
        inc interval0
        mov a,interval0
        cjne a,#3ah,int_return
        mov a,interval1
        cjne a,#39h,notniner
        mov interval0,#31h
        mov interval1,#30h
        sjmp int_return

    notniner:

        inc interval1
        mov interval0,#30h
        sjmp int_return

    dec_int:

        cjne r6,#02h,int_return
        dec interval0
        mov a,interval0
        cjne a,#30h,int_return
        mov a,interval1
        cjne a,#30h,notzeror
        mov interval0,#39h
        mov interval1,#39h
        sjmp int_return
```

```
notzeror:

        dec interval1
        mov interval0,#39h

int_return:

        mov lcdcom,#0c5h
        lcall command
        mov disp_data,interval1
        lcall data_routine
        mov disp_data,interval0
        lcall data_routine

        lcall delay140m                             ;debouncing delay
        lcall delay140m
        lcall delay140m
        ret

;**************************************************************************

display_text:                           ;display a 16 byte in ASCII string on lcd

        mov count,#0fh

nextchar:

        mov a,#00h
        movc a,@a+dptr
        mov disp_data,a
        lcall data_routine
        inc dptr
        mov a,count
        cjne a,#00h,next1
        sjmp here11

next1:

        dec count
        sjmp nextchar

here11:

        ret

;**************************************************************************
```

```
lcd_initialize:                         ;initialize lcd

        mov lcdcom,#38h                 ;5*7 matrix 16*2 char lcd
        lcall command
        lcall delay40u

        mov lcdcom,#01h                 ;clear lcd
        lcall command
        lcall delay14m

        mov lcdcom,#0Eh                 ;blink off, cursor ON, display ON
        lcall command
        lcall delay40u

        mov lcdcom,#06h                 ;shift cursor right
        lcall command
        lcall delay40u
        ret

;***********************************************************************************

clear_lcd:

        mov lcdcom,#01h                 ;clear LCD screen
        lcall command
        lcall delay14m
        mov lcdcom,#80h
        lcall command
        ret

;***********************************************************************************
cursoroff:                              ;lcd cursor off

        mov lcdcom,#0ch
        lcall command
        ret

;***********************************************************************************

cursoron:                               ;lcd cursor on

        mov lcdcom,#0eh
        lcall command
        ret

;***********************************************************************************
```

```
command:                                        ;routine to send the command to LCD

        clr p3.2        ;rs=0
        clr p3.3        ;rw=0
        lcall delay10u  ;delay of 10usec
        setb p3.4       ;E=1;
        lcall delay10u  ;delay of 10usec
        mov p1,lcdcom          ;mov data
        lcall delay10u  ;delay of 10usec
        clr p3.4        ;E=0
        lcall delay10u
        setb p3.3       ;rw=1
        lcall delay40u
        ret

;**********************************************************************************

data_routine:                                   ;routine to send ASCII data to LCD

        setb p3.2       ;rs=1
        clr p3.3        ;rw=0
        lcall delay10u
        setb p3.4       ;E=1;
        lcall delay10u
        mov p1,disp_data
        clr p3.4        ;E=0
        lcall delay40u
        ret

;**********************************************************************************

;******************Delay of 10usec*********************************************

delay10u:                                       ;delay of 10 usec

        mov r4,#04h

here10u:

        djnz r4,here10u

        ret

;**************Delay of 40usec*************************************************

delay40u:                                       ;delay of 40 usec
```

```
        mov r3,#09h
    here1_40u:
        mov r4,#01h
    here2_40u:
        djnz r4,here2_40u
        djnz r3,here1_40u
        ret
;********************delay of 14m*********************************************
    delay14m:                               ;delay of 14 msec
        mov r3,#0ffh
    here1_14m:
        mov r4,#18h
    here2_14m:
        djnz r4,here2_14m
        djnz r3,here1_14m
        ret
;****************************************************************************
    delay140m:                              ;delay of 140 msec
        mov r3,#0ffh
    here1_140m:
        mov r4,#0ffh


    here2_140m:
        djnz r4,here2_140m
        djnz r3,here1_140m
        ret
;********************delay of 1s**********************************************
```

```
delay1sec:                                    ;delay of 1 sec

        mov r3,#08h

    here1_1s:

        mov r4,#0ffh

    here2_1s:

        mov r5,#0ffh

    here3_1s:

        djnz r5,here3_1s
        djnz r4,here2_1s
        djnz r3,here1_1s
        ret

org 0700h
db '4th YEAR PROJECT'
org 0710h
db 'AQUARIUM MNGMNT'
org 0720h
db 'SYSTEM'
org 0730h
db 'Enter Parameters'
org 0740h
db 'Measure Interval'
org 0750h
db 'Threshold value '
org 0760h
db 'Parameters Saved'
org 0770h
db 'Temperature '
org 0780h
db 'Sample no. '

;********************************PROGRAM ENDS HERE************************

end
```

The software is well commented and easy to understand. Written in Assembly language and assembled using A51 assembler.

Assembler directives and comments are used for proper understanding of the software. The hex code generated by the assembler is burnt into the microcontroller using a suitable programmer.

The ranges for measuring time interval (01 to 99 seconds) and threshold temperature value (20°C to 49°C) are set by the software. The values of the measured temperature and the number of samples taken until power-on, are displayed on the LCD screen as shown in Fig. 6. The number of samples is updated according to the measuring interval.

Each port of the microcontroller is made input through software by putting high on the respective pin or port. By default, all the ports act as outputs. Instead of using timer, nested loops are used to provide delays at various locations of the software. The values for the loops are calculated according to the crystal frequency and the machine cycles taken by the used instructions. Functioning of all the keys (up, down and enter) is also handled by the software. Pooling, identification of keys, and limits for up and down are provided by the software.

Chapter 4: Testing and Results

Results of Temperature Sensor:

Temperature Range (°C)	Sensor output (V_{OUT})
20-25	324mV
25-30	364mV
30-35	424mV
35-40	478mV
40-45	526mV
45-50	563mV
50-55	627mV

ADC Readings:

Bit Position	ADC Input Voltage					
	0V	1V	2V	3V	4V	5V
D0	0	1	1	1	0	1
D1	0	0	0	0	0	1
D2	0	1	1	1	0	1
D3	0	0	0	0	1	1
D4	0	1	0	1	0	1
D5	0	1	1	0	0	1
D6	0	0	1	0	1	1
D7	0	0	0	1	1	1

Chapter 5: Future Work

For many years now applications have been available which allow you to control various elements of the aquarium whilst you are away from it. The can be simple things such as electrical timers to the more advanced controllers.

There are also some applications which allow you to monitor the aquarium over the internet so that you can check various parameters – again these are normally quite expensive.

Watching the aquarium whilst you are away is very simple as you can just hook up a webcam and broadcast it over the internet.

There is a relatively new one now available and it looks to be very good at first sight.

It is called TankedCAM.

What TankedCAM allows you to do is basically monitor your aquarium whilst you are away from home. This could be when you are at work, travelling, on vacation etc.

TankedCAM gives you the ability to be able to view your aquarium using a live video feed. It also allows you to pan the camera, check water temperature, pH levels, feed the fish and operate various equipment such as lighting, moonlights etc.

You can watch, monitor and control your aquarium directly from your I-Phone!

DATASHEETS

AT89C51 DATASHEET

Features

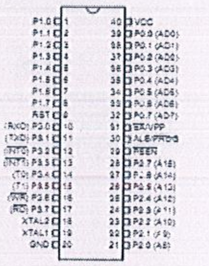
- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt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

Description

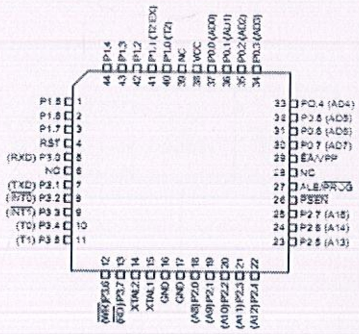
The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and 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

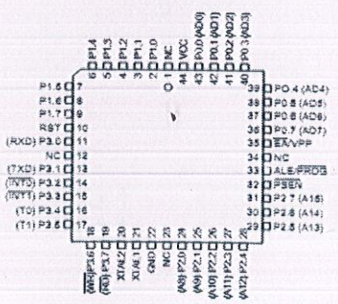
PDIP



PQFP/TQFP



PLCC



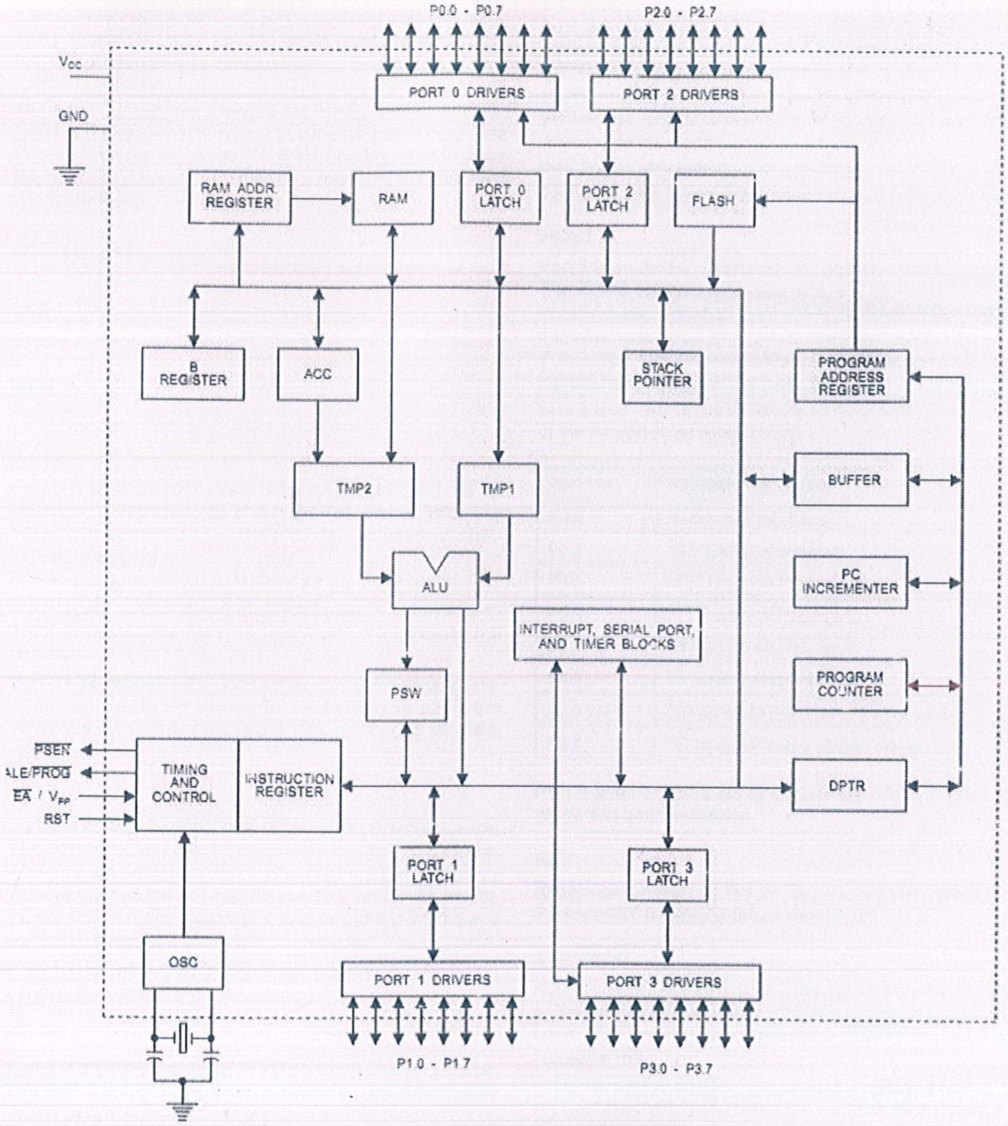
8-bit Microcontroller with 4K Bytes Flash

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.

PSEN

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

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

EA/VPP

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

EA should be strapped to 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

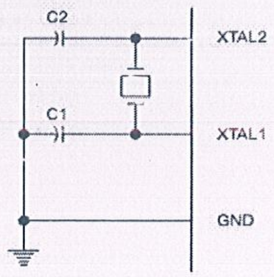
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected 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

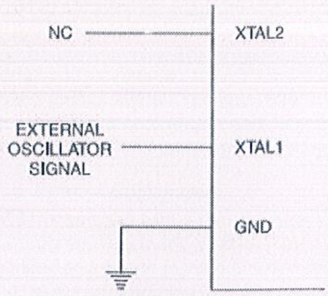


Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	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

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low-voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third-party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V_{PP} = 12V	V_{PP} = 5V
Top-side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H) = 1EH (031H) = 51H (032H) =F FH	(030H) = 1EH (031H) = 51H (032H) = 05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figure 3 and Figure 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}/V_{PP}$ to 12V for the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{Data}$ Polling: The AT89C51 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/$\overline{PROG}$ low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 51H indicates 89C51
(032H) = FFH indicates 12V programming
(032H) = 05H indicates 5V programming

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

Mode		RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}/V_{PP}$	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L		H/12V	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L		H/12V	H	H	H	H
	Bit - 2	H	L		H/12V	H	H	L	L
	Bit - 3	H	L		H/12V	H	L	H	L
Chip Erase		H	L	(1)	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10 ms $\overline{PROG}$ pulse.

Figure 3. Programming the Flash

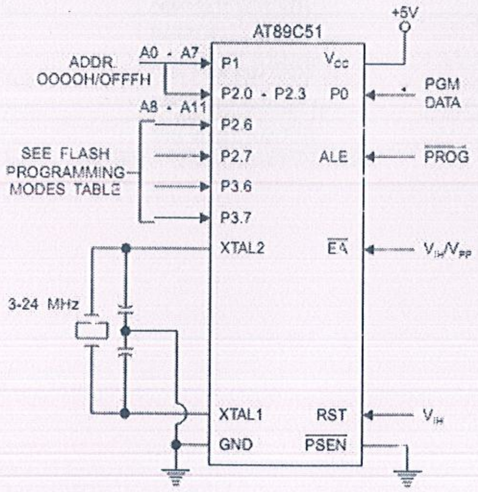
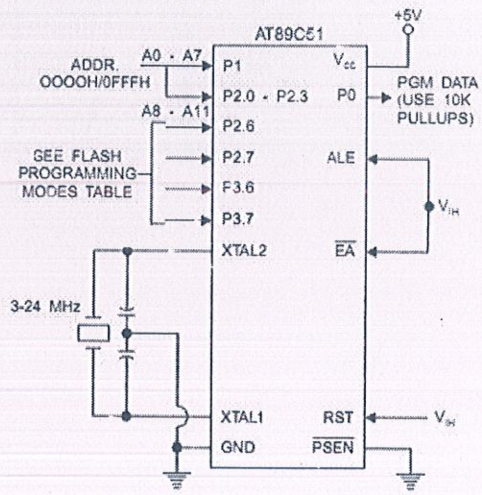


Figure 4. Verifying the Flash



Flash Programming and Verification Characteristics

T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V_{PP} [(1)]	Programming Enable Voltage	11.5	12.5	V
I_{PP} [(1)]	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold after $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold after $\overline{PROG}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		μs
t_{GHSL} [(1)]	V_{PP} Hold after $\overline{PROG}$	10		μs
t_{GLGH}	$\overline{PROG}$ Width	1	110	μs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float after $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	μs
t_{WC}	Byte Write Cycle Time		2.0	ms

Note: 1. Only used in 12-volt programming mode.

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	15.0 mA

*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.

DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 5.0V ±20% (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC} - 0.1	V
V_{IL1}	Input Low-voltage ($\overline{EA}$)		-0.5	0.2 V_{CC} - 0.3	V
V_{IH}	Input High-voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
V_{IH1}	Input High-voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
V_{OL}	Output Low-voltage(1) (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low-voltage(1) (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High-voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ±10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High-voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ±10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, VCC = 5V ±10%		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pull-down Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power-down Mode(2)	V_{CC} = 6V		100	µA
		V_{CC} = 3V		40	µA

Notes:
1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port: Port 0: 26 mA
   Ports 1, 2, 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.

AC Characteristics

Under operating conditions, load capacitance for Port 0, ALE/$\overline{\text{PROG}}$, and $\overline{\text{PSEN}}$ = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	43		t_{CLCL}-13		ns
t_{LLAX}	Address Hold after ALE Low	48		t_{CLCL}-20		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}$-65	ns
t_{LLPL}	ALE Low to $\overline{\text{PSEN}}$ Low	43		t_{CLCL}-13		ns
t_{PLPH}	$\overline{\text{PSEN}}$ Pulse Width	205		$3t_{CLCL}$-20		ns
t_{PLIV}	$\overline{\text{PSEN}}$ Low to Valid Instruction In		145		$3t_{CLCL}$-45	ns
t_{PXIX}	Input Instruction Hold after $\overline{\text{PSEN}}$	0		0		ns
t_{PXIZ}	Input Instruction Float after $\overline{\text{PSEN}}$		59		t_{CLCL}-10	ns
t_{PXAV}	$\overline{\text{PSEN}}$ to Address Valid	75		t_{CLCL}-8		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}$-55	ns
t_{PLAZ}	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{\text{RD}}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{WLWH}	$\overline{\text{WR}}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{RLDV}	$\overline{\text{RD}}$ Low to Valid Data In		252		$5t_{CLCL}$-90	ns
t_{RHDX}	Data Hold after $\overline{\text{RD}}$	0		0		ns
t_{RHDZ}	Data Float after $\overline{\text{RD}}$		97		$2t_{CLCL}$-28	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}$-150	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}$-165	ns
t_{LLWL}	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	200	300	$3t_{CLCL}$-50	$3t_{CLCL}$+50	ns
t_{AVWL}	Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	203		$4t_{CLCL}$-75		ns
t_{QVWX}	Data Valid to $\overline{\text{WR}}$ Transition	23		t_{CLCL}-20		ns
t_{QVWH}	Data Valid to $\overline{\text{WR}}$ High	433		$7t_{CLCL}$-120		ns
t_{WHQX}	Data Hold after $\overline{\text{WR}}$	33		t_{CLCL}-20		ns
t_{RLAZ}	$\overline{\text{RD}}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	43	123	t_{CLCL}-20	t_{CLCL}+25	ns

ADC 0804 DATASHEET

National Semiconductor

October 1999

ADC0808/ADC0809
8-Bit µP Compatible A/D Converters with 8-Channel Multiplexer

General Description

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals.

The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE® outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.)

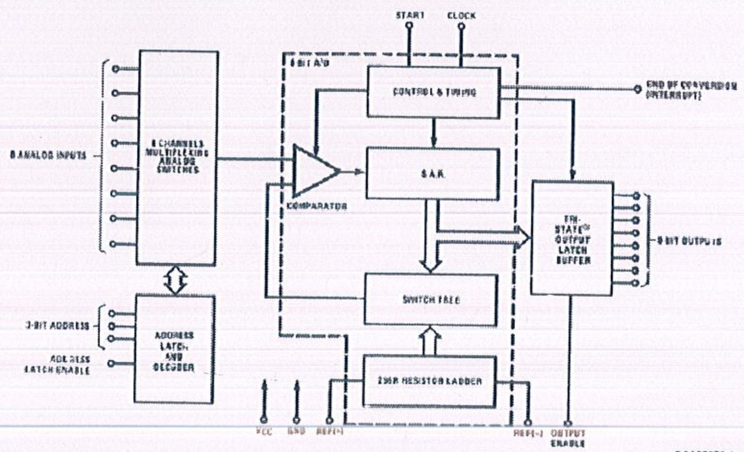
Features

- Easy interface to all microprocessors
- Operates ratiometrically or with 5 V_{DC} or analog span adjusted voltage reference
- No zero or full-scale adjust required
- 8-channel multiplexer with address logic
- 0V to 5V input range with single 5V power supply
- Outputs meet TTL voltage level specifications
- Standard hermetic or molded 28-pin DIP package
- 28-pin molded chip carrier package
- ADC0808 equivalent to MM74C949
- ADC0809 equivalent to MM74C949-1

Key Specifications

■ Resolution	8 Bits
■ Total Unadjusted Error	±½ LSB and ±1 LSB
■ Single Supply	5 V_{DC}
■ Low Power	15 mW
■ Conversion Time	100 µs

Block Diagram



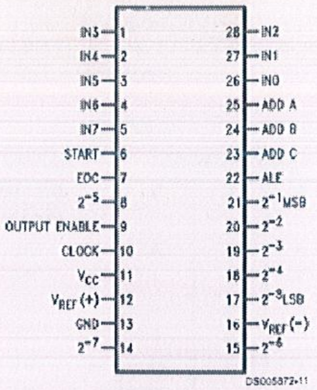
DS005672-1

See Ordering Information

TRI-STATE® is a registered trademark of National Semiconductor Corp.

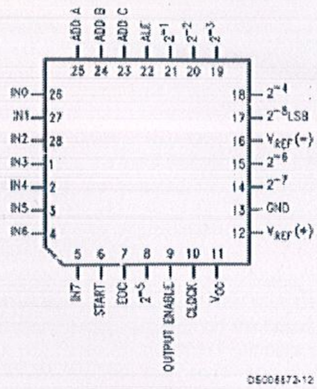
Connection Diagrams

Dual-In-Line Package



Order Number ADC0808CCN or ADC0809CCN
See NS Package J28A or N28A

Molded Chip Carrier Package



Order Number ADC0808CCV or ADC0809CCV
See NS Package V28A

Ordering Information

TEMPERATURE RANGE		−40°C to +85°C			−55°C to +125°C
Error	±½ LSB Unadjusted	ADC0808CCN	ADC0808CCV	ADC0808CCJ	ADC0808CJ
	±1 LSB Unadjusted	ADC0809CCN	ADC0809CCV		
Package Outline		N28A Molded DIP	V28A Molded Chip Carrier	J28A Ceramic DIP	J28A Ceramic DIP

Functional Description

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

TABLE 1.

SELECTED ANALOG CHANNEL	ADDRESS LINE		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

CONVERTER CHARACTERISTICS

The Converter

The heart of this single chip data acquisition system is its 8-bit analog-to-digital converter. The converter is designed to give fast, accurate, and repeatable conversions over a wide range of temperatures. The converter is partitioned into 3 major sections: the 256R ladder network, the successive approximation register, and the comparator. The converter's digital outputs are positive true.

The 256R ladder network approach (*Figure 1*) was chosen over the conventional R/2R ladder because of its inherent monotonicity, which guarantees no missing digital codes. Monotonicity is particularly important in closed loop feedback control systems. A non-monotonic relationship can cause oscillations that will be catastrophic for the system. Additionally, the 256R network does not cause load variations on the reference voltage.

The bottom resistor and the top resistor of the ladder network in *Figure 1* are not the same value as the remainder of the network. The difference in these resistors causes the output characteristic to be symmetrical with the zero and full-scale points of the transfer curve. The first output transition occurs when the analog signal has reached +½ LSB and succeeding output transitions occur every 1 LSB later up to full-scale.

The successive approximation register (SAR) performs 8 iterations to approximate the input voltage. For any SAR type converter, n-iterations are required for an n-bit converter. *Figure 2* shows a typical example of a 3-bit converter. In the ADC0808, ADC0809, the approximation technique is extended to 8 bits using the 256R network.

The A/D converter's successive approximation register (SAR) is reset on the positive edge of the start conversion (SC) pulse. The conversion is begun on the falling edge of the start conversion pulse. A conversion in process will be interrupted by receipt of a new start conversion pulse. Continuous conversion may be accomplished by tying the end-of-conversion (EOC) output to the SC input. If used in this mode, an external start conversion pulse should be applied after power up. End-of-conversion will go low between 0 and 8 clock pulses after the rising edge of start conversion.

The most important section of the A/D converter is the comparator. It is this section which is responsible for the ultimate accuracy of the entire converter. It is also the comparator drift which has the greatest influence on the repeatability of the device. A chopper-stabilized comparator provides the most effective method of satisfying all the converter requirements.

The chopper stabilized comparator converts the DC input signal into an AC signal. This signal is then fed through a high gain AC amplifier and has the DC level restored. This technique limits the drift component of the amplifier since the drift is a DC component which is not passed by the AC amplifier. This makes the entire A/D converter extremely insensitive to temperature, long term drift and input offset errors.

Figure 4 shows a typical error curve for the ADC0808 as measured using the procedures outlined in AN-179.

Functional Description (Continued)

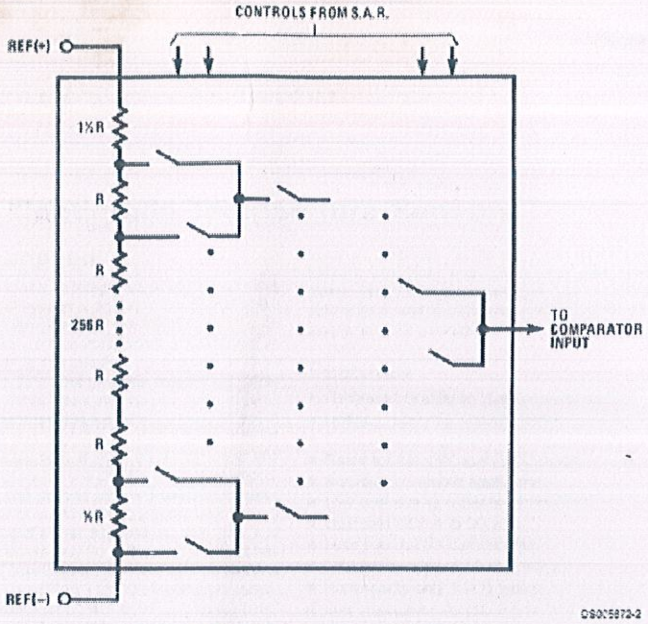


FIGURE 1. Resistor Ladder and Switch Tree

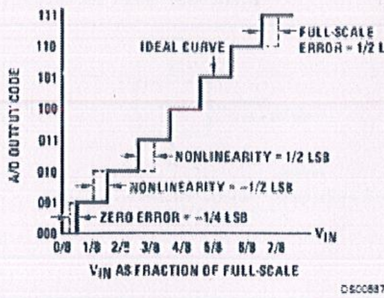


FIGURE 2. 3-Bit A/D Transfer Curve

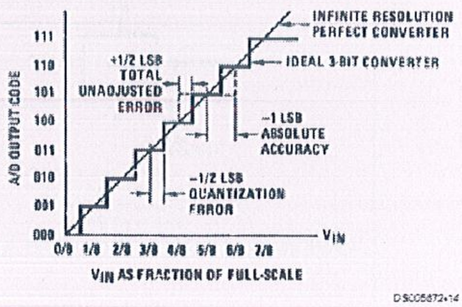


FIGURE 3. 3-Bit A/D Absolute Accuracy Curve

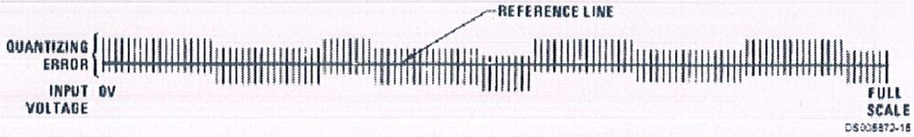


FIGURE 4. Typical Error Curve

LM35 DATASHEET

National Semiconductor

November 2000

LM35
Precision Centigrade Temperature Sensors

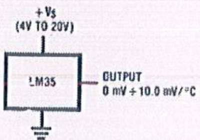
General Description

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

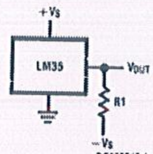
- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- 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



DS005516-3

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



DS005516-4

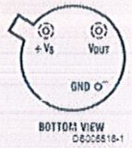
Choose $R_1 = -V_S/50\ \mu A$
V_{OUT} = +1,500 mV at +150°C
= +250 mV at +25°C
= −550 mV at −55°C

FIGURE 2. Full-Range Centigrade Temperature Sensor

 www.national.com

Connection Diagrams

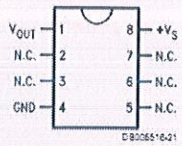
TO-46
Metal Can Package*



*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



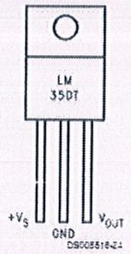
N.C. = No Connection

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

REFERENCES

- http://www.atmel.com/
- http://mchportal.com/fishkeeping-mainmenu-60/water-the-environment-mainmenu-77/65-ph-kh-gh-carbon-dioxide-injection-water-chemistry.html
- http://www.firsttankguide.net/ph.php
- http://www.aquaticcommunity.com/aquarium/growth.php
- http://www.aquahobby.com/articles/e_adjusting_pH.php
- http://blea.ch/wiki/index.php/PH_Meter
- http://www.atmel.com/products/microcontrollers/8051architecture/default.aspx
- http://www.keil.com/c51/