Microcontroller Base Automatic Temperature control System

This project report is submitted to the Department of Electrical & Electronic Engineering (EEE), Daffodil International University, Bangladesh, in Partial fulfillment of the requirements for the Degree of "Bachelor of Science in Electrical & Electronic Engineering".

Prepared By

Md.Monjur Hossain	ID: 121-33-837
Mahmudul Hasan murad	ID: 121-33-851

Supervised By

Md. MahmudurRahman

Senior Lecturer Department of Electrical & Electronic Engineering (EEE)
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY DHAKA, BANGLADESH, MARCH 2015

DECLARATION

We hereby declare that, this project has been done by us under the supervision of Md. MahmudurRahman, Senior Lecturer, Department of EEE, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Md. Mahmudur Rahman

Senior Lecturer

Department of EEE

Daffodil International University

Submitted by:

Md.Monjur Hossain	**Mahmudul Hasan Murad**
ID: 121-33-837	ID: 121-33-851
Department of EEE	Department of EEE
Daffodil International University	Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty Allah for his divine blessing makes us possible to complete this project successfully.

We fell grateful to and wish our profound indebtedness to our Supervisor Md. Mahmudur Rahman, Senior Lecturer, Department of EEE Daffodil International University, Dhaka. Deep knowledge & keen interest of our supervisor has worked as an influencer to carry out project. His endless patience , scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to Prof. Dr. Md. Samsul Alam, Dean, Faculty of Eng. And Dr. Fayzur Rahman, Head, Department of EEE, for his kind help to finish our project and also to other faculty member and the staff of EEE department of Daffodil International University.

We would like to thanks our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

__ _ ___ __ _ __ _ __***Authors***

ABSTRACT

First of all according to the collected information about Temperature control System then total circuit design has been completed using Proteus software.

Infrared sensor used in this project. Which detect Temperature? It is used as a measure of Temperature of a mechanical component. Produce analog voltage sensing the present Temperature at its input.

AVR ATMEGA32 PU Microcontroller was programmed with programming language C in code vision AVR compiler. Microcontroller calculating this value transmitted it to a LCD to display the RPM.

JHD 162D Liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs).They are usually more compact, lightweight, portable, less expensive, more reliable, and easier on the eyes. This is a basic 16 character by 2 line display with a snazzy green background with white characters.

12 V Transformer is used as its main power supply. In this transformer primary voltage is 220 V and in secondary coil voltage is 12 V AC. LM7805 & LM7812 voltage converter used to convert voltage (12 V and 5 V). A switch is used to start the system.

TABLE OF CONTENT

Contents

List of Figures

Chapter-1

Introduction

1.1 Background:

NTELLIGENT TEMPERATURE MONITORING AND CONTROL SYSTEM USING AVR MICROCONTROLLER

Controlling temperature has been a prime objective in various applications including refrigerators, air conditioners, air coolers, heaters, industrial temperature conditioning and so on. Temperature controllers vary in their complexities and algorithms. Some of these use simple control techniques like simple on-off control while others use complex Proportional Integral Derivative (PID) or fuzzy logic algorithms. In this project I'm going to discuss about a simple control algorithm and utilize it intelligently unlike analogue controllers. Here are the features of our controller:

1: Audio-visual setup for setting temperature limits.
2: Fault detection and evasive action.
3: Temperature monitoring and display.
4: Audio-visual warning.
5: System status.
6: Settable time frame.
7: Data retention with internal EEPROM memory

For this design I used an ATMega32L AVR microcontroller with an internal clock frequency of 8MHz, a 2×16 LCD for presenting data visually, a 4-button input interface, a tiny piezo sounder for audible indications and some LEDs for showing faults and simulating real-life devices like heater and coolers. I used Mikro C for AVR compiler from Mikroelektronika to develop the firmware for this controller.

Hardware:

The hardware consists of a four-button interface, four LEDs, a piezo tweeter or sounder, a 2×16 LCD, a LM35 temperature sensor, an AVR ATMega32L microcontroller and some other passive parts. Two LEDs connected to PORTD0 and PORTD1 pins simulate on and off operation of relay switches that are actually present in actual applications to control a heater and an air conditioner. The hardware is powered by a 5V source preferably with a 5V regulator like 7805. If relays are used then a 12/24V source will also be needed to power the relays. The AVR micro's AVREF and AVCC pins should be both connected to the 5V source. The distance between the LM35 sensor and the AVRs ADC pin must not be greater than 10-12cm for proper temperature reading. Though I did this project in a prototyping board made with a strip board, a PCB version is more preferred. Two additional LEDs connected to PD2 and PD3 indicates the failure or malfunctioning status of the air conditioner and the heater.

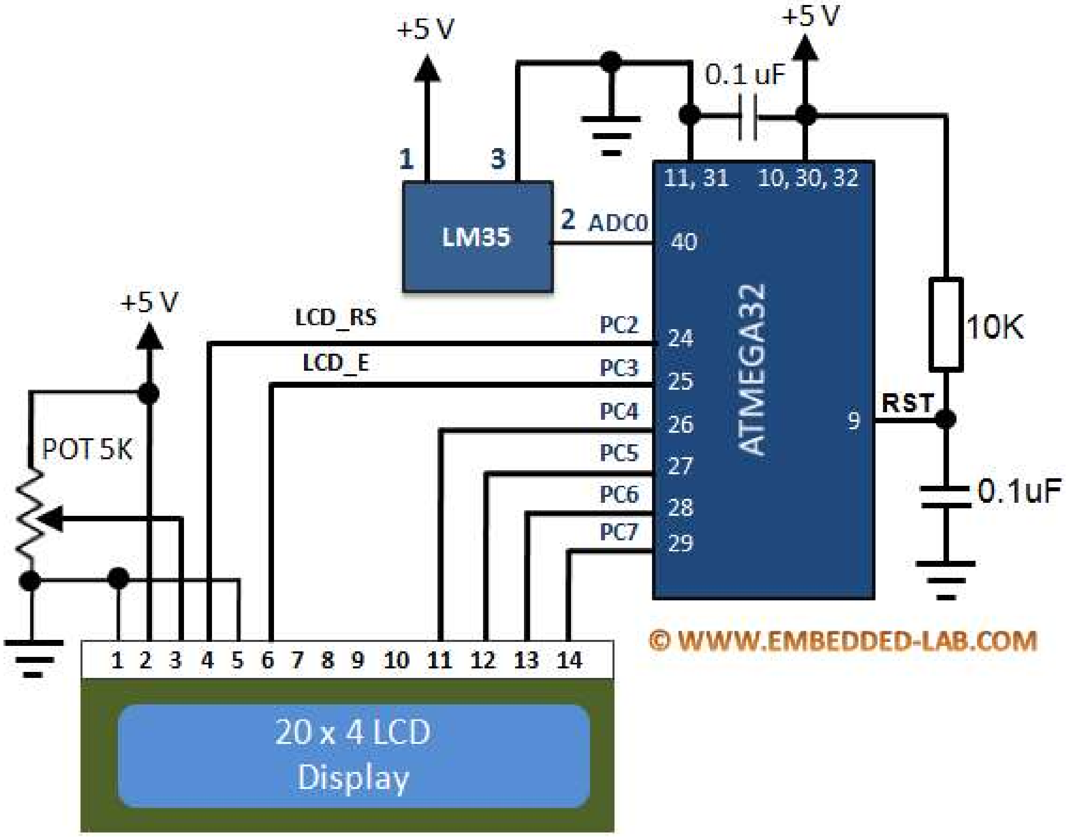


FIG1.1: Microcontroller and display circuit

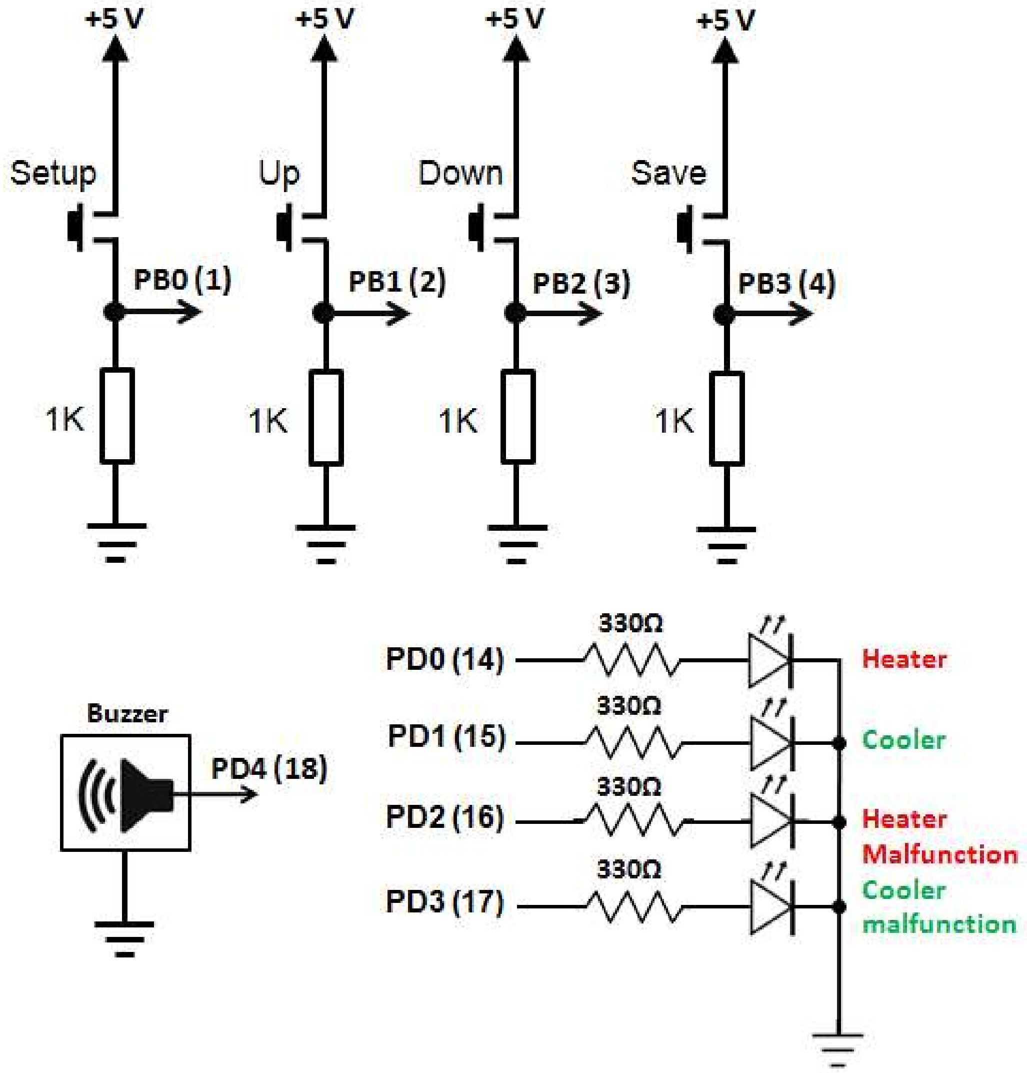


FIG1.2:Input/Output circuit

Firmware:

The entire firmware relies on logical decisions at every step. I wrote the code in a way that the complete code is divided into understandable small functional blocks or functions. Each of these functions is doing specific tasks.

Starting from the top of the code, I declared some definitions of port pins, LCD pins and EEPROM locations. Next I declared global variables and function prototypes.

I'm now going to explain the tasks each functions do. The first function that's called in the main() function is the *setup_mcu()* function. It configures the I/O ports for general tasks, LCD and audio output pin. It also sets flags *"c"* and *"h"* zero. These flags are set whenever the system faces an alarm. *"h"* means heater fault flag and *"c"* means cooler fault flag. Next to the *setup_mcu()* function is the *scan_keys()* function which simple reads the buttons and a generate specific tone for the key pressed and return a specific value for that particular key. Thus key press, key debouncing delays and tone generation accompanied with key press is made universal throughout the entire code. The *adc_avg()* function takes 64 samples of channel 0 of the AVR's built-in ADC and makes the average of these samples to reduce noise and ensure accuracy. This function returns an unsigned long value since 10-bit ADC resolution is used. Following the *adc_avg()* function is the temperature display function *temp_display()*. All it does is simply show the current value of the temperature on the LCD display. The *settings()* function is the most important part of the program. When entered this function has two menus. The first one according to the program is the reset fault service request and the second allows the user to setup the high and low temperature values along with the number of passes that will be allowed for the temperature to reach the nominal value. Whenever these values are exceeded an alarm goes high and a particular temperature control device (heater or cooler) is set. Thus setting the right values ensure proper system operation and avoid false alarms. The *inc_dec* function is used for the menu operations whenever when we need to change a value. It is called to setup temperature values and pass values. The *display_common* function is used for common display purposes as needed in the *inc_dec* function. Next the *compare_temp()* routine compares temperature and shows a message in the LCD if the temperature is within limit or beyond limit. Another important part of the program is the function called *controller_state*. It is here the controller sets on or off the temperature control devices and also generates alarm in the event of any error. Associated with this routine is the *check_fault* routine when the conditions for fault are checked. The *read_memory* routine reads the stored data for temperature limits and others in the AVR internal EEPROM. Finally the last three functions of the code *are fault_messages()* which shows the fault type, light the specific LED the specific fault and generate a warning tone for all faults; *settings_demanded()* calls for settings menu and *all_tasks()* is the collection of all the tasks done in the main function.

1.2 Counting temperature

Operation:

The system works as described here. Upon start-up the system sets up the required internal hardware of the AVR micro and then read the EEPROM memory. If the EEPROM locations contain garbage values then they are set with default values otherwise the previously stored values are read. After reading the EEPROM, the system starts to monitor temperature and waits for actions if any temperature limit is exceeded. The system at this point shows current temperature and system status. If the user wishes to set parameters then he/she has to press the setup button and enter the settings menu. In the settings menu there are three settings and these are high and low temperature limits and the number of passes the system will make prior to issuing a fault message. If, for example, the high temperature limit is set to 40?C, the low temperature limit is set to 20?C and the number of system passes is set to 45, and the current temperature gradually rises to 41?C from 30?C, the system will trip high temperature alarm and start the cooler. The LCD display will show high temperature alarm. Now if the temperature starts to decline and reach a value in between (40?C – (40?C -20?C)/2) =30?C [i.e t_delta] and 20?C within the 45 system passes then the cooler is turned off and no fault message is generated. The system resumes to normal state. If the temperature didn't decline to the range mentioned as above then it is assumed that the cooler is either faulty or some other thing is causing too much heat generation which is exceeding the cooler's capacity. Thus a fault warning is issued for the cooler and it is shut down until the fault has been cleared. In this way both the hardware and the cooler is protected from damage. The same scenario happens during the low temperature alarm. If both the cooler and the heater fail then the system goes in a complete halt state until reset or given attention.

Controlling temperature has been a prime objective in various applications including refrigerators, air conditioners, air coolers, heaters, industrial temperature conditioning and so on. Temperature controllers vary in their complexities and algorithms. Some of these use simple control techniques like simple on-off control while others use complex Proportional Integral Derivative (PID) or fuzzy logic algorithms. In this project I'm going to discuss about a simple control algorithm and utilize it intelligently unlike analogue controllers. Here are the features of our controller: Audio-visual setup for setting temperature limits. Fault detection and evasive action. Temperature monitoring and display. Audio-visual warning. System status. Settable time frame. Data retention with internal EEPROM memory

1.3 Experimental Setup

The proposed block diagram of the hardware developed in the present work. It requires the following: Inductive proximity sensors operate under the electrical principle of inductance. Inductance is the phenomenon where a fluctuating current, which by definition has a magnetic component, induces an Electro Motive Force (EMF) in the temperature.

A temperature sensor has four components, the coil, Oscillator, detection circuit and output circuit. The oscillator generates a fluctuating magnetic field the shape of the doughnut around the winding of the coil that locates in the

Device's sensing face [6]. The Photograph of temperature sensor used in this present work is shown in Sensor ATMEGA32 Microcontroller LCD Module.

The detection distance of an inductive proximity sensor depends on the target materials capacity for conducting electricity. Also, the target's thickness will have an influence of its detection. Generally thin materials are easier for an temperature sensor to detect the temperature.

The microcontroller used in the present work is a low-cost and popular. This microcontroller is selected for the following reasons:

1. It is a 40-pin IC having the architecture and instruction set compatibility with ATMGA32 microcontroller series.

2. It has on-chip precision analog comparator

3. Interrupt control, timer and serial transmission facilities

4. Only disadvantage with this IC is that there are only two ports available (one 8-bit port and the other 5-bit port).Cost-effectiveness and low-pin count have attracted us to use this microcontroller. Selection of 110592 MHz crystal

1.4 Objective:

1. To design a low cost reliable RPM meter

1.5 Methodology :

1. Collection of information from books and internet.

2. Collection of components from local market.

1.6 Major Components of the project:

- Voltage regulator (7805)(7812)
- Capacitor (1000uF,22pF)
- Resistor (220Ω,1kΩ,10kΩ)
- LED (red)
- Temperature sensor(LM35)
- Relay(12v)
- Micro-controller (ATMEGA32)
- LCD (alphanumeric, 16x2)
- Nut-screw
- PCB Board
- Wire as needed

Chapter-2

Design Criteria Temperature control System

2.1 Introduction

Design is an important part of all circuits and devices. Here, design criteria describe the circuits components use to assemble the microcontroller based RPM Meter system with their working principle and operation in this circuit

2.2 Block Diagram:

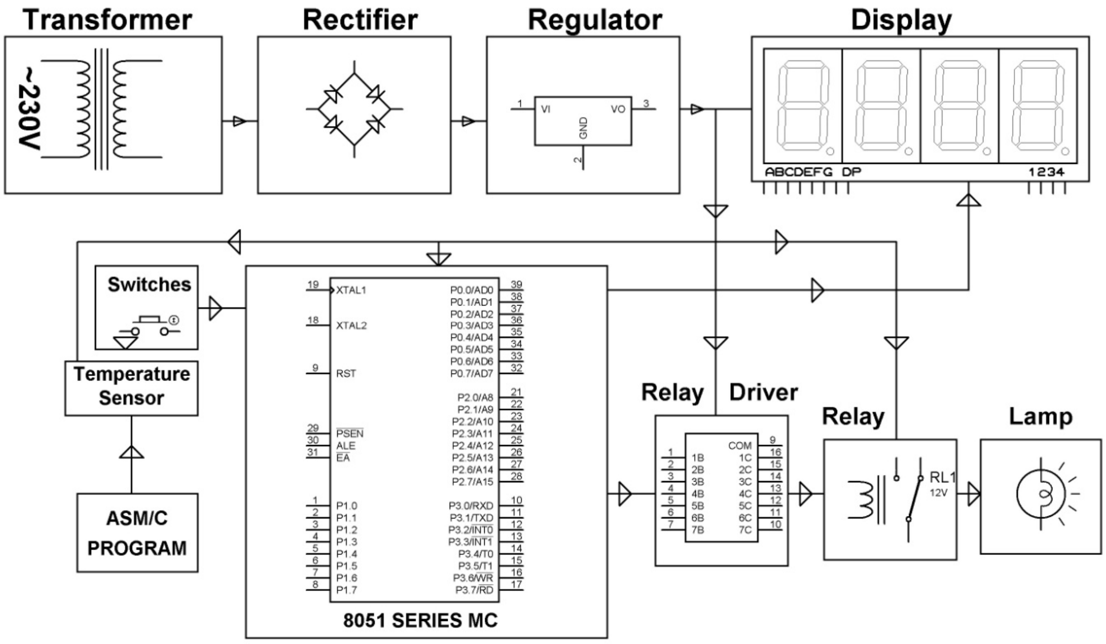
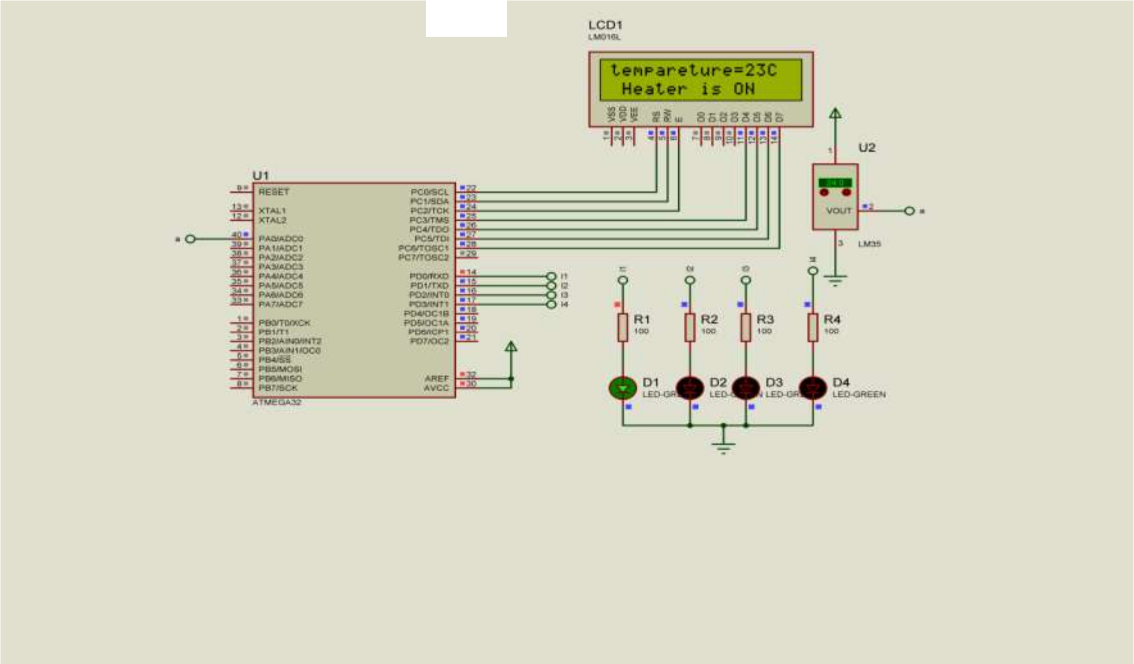


Fig: 2.1 Block Diagram of the Temperature Control System

2.3 Circuit Diagram



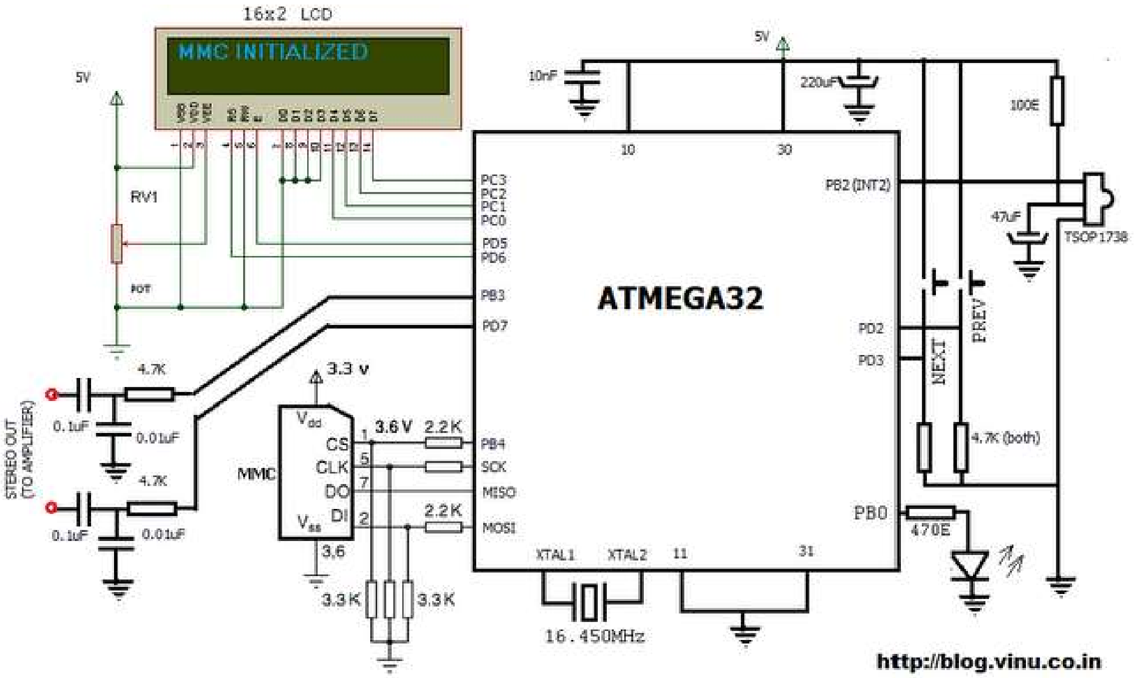


Fig: 2.2 Approximated diagram

2.4 Working Principle of the Circuit:

To describe the working principle the system can be divided into three unite:

1. Power supply unit.

2.Sensor unit.

3. Controlling process and display unit.

4.Relay Unit

Power supply designed into two parts in this circuit, One part is used 220 V AC within a G2R 14 AC 120 relay for the purpose of boiling water .and the other part builds with 12V step down transformer, Capacitor, Rectifier diode and a Voltage regulator. The power input is 220 V AC into the transformer's primary coil. So the output is 12 V AC from the secondary coil. This 12 V AC converted to 12V DC by rectifier and capacitor. We need 5 V DC for microcontroller and sensor .So used a voltage regulator (7805) to convert this 1 V to 5V.

This Project is a reliable circuit that does the task of monitoring frequency of a rotation very accurately.

Power supply unit:

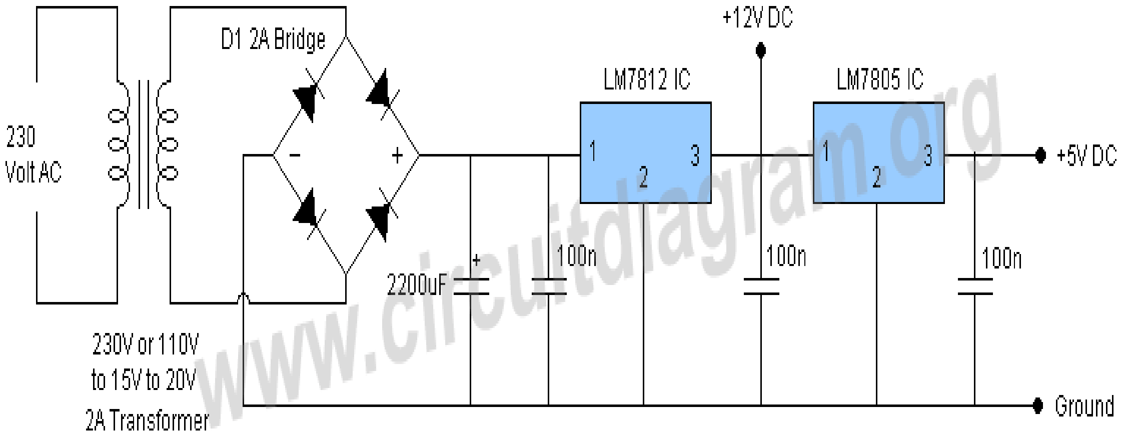


Fig 2.3: Power supply Unit

Sensor unit:

As the infrared sensor provide analog voltage with detecting rotation of object and connected to the microcontroller interrupt pin, Following the data of infrared sensor output carve and with calculating make a program with those respective voltage. Install this program in microcontroller and connected an Alphanumeric 16x2 Display with it for show the RPM Unite.

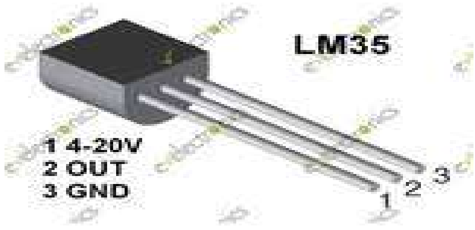


Fig 2.4:Pin Diagram

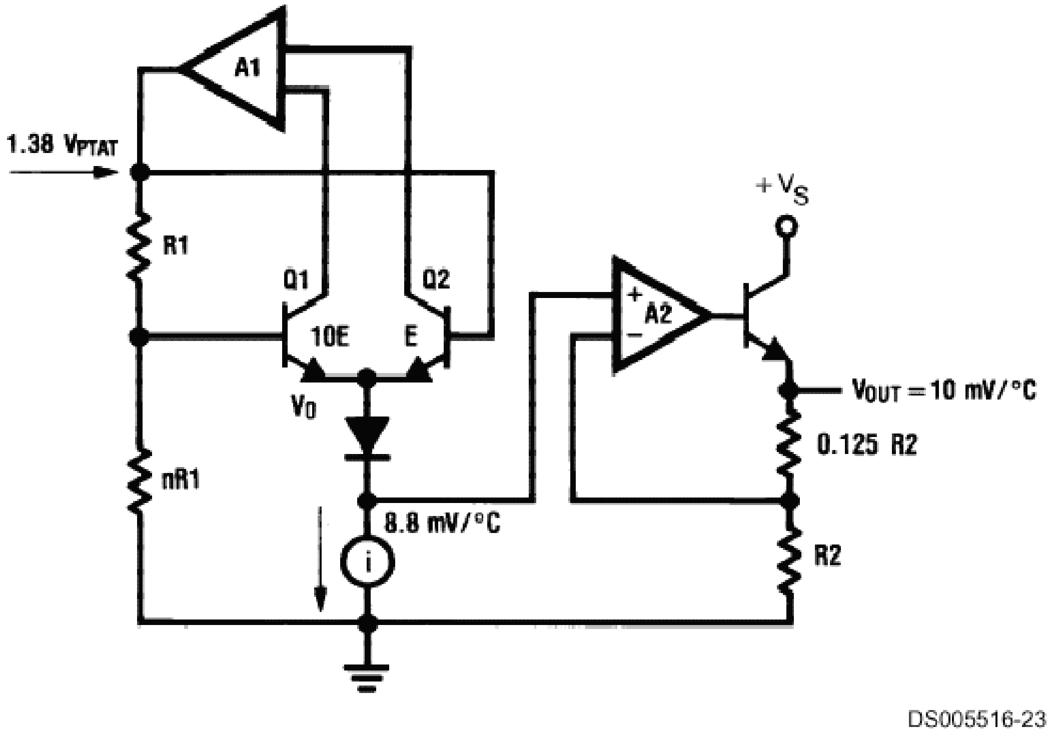


Fig 2.5: Circuit diagram

Controlling process and display unit:

According to those voltage sensing of lm35 making a program in PIC18F452 Microcontroller. When sensor output voltage is 250mv then the sensing temperature is +25°C and when the output voltage is 1500mv then the temperature is +150°C.Those temperature of the oven shows on a 16x2 characteristic Alphanumeric display. Generally at +100°C temperature water has boiled. The program has made with thinking that when the oven temperature reaches +150°C the oven should be off and when the temperature reached +25°C the oven should be on. Microcontroller does this process with proving signal to the G2R 14 AC 120 relay.

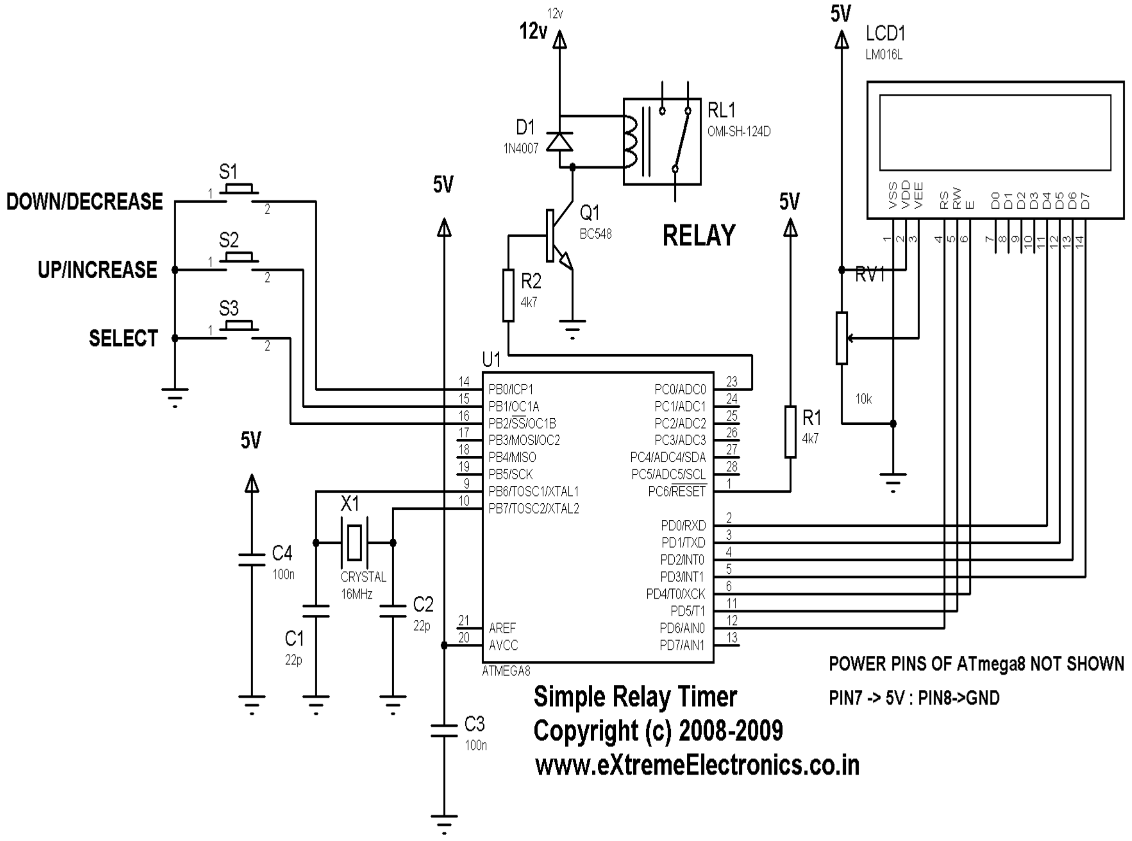


Fig 2.6: Controlling process and display unit.

RELAY UNIT:

A **relay** is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such assolid-state relays.

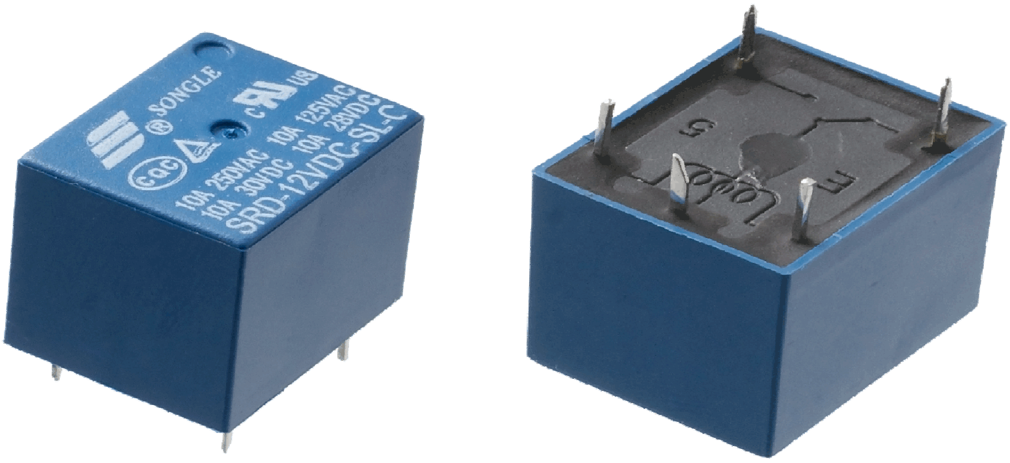
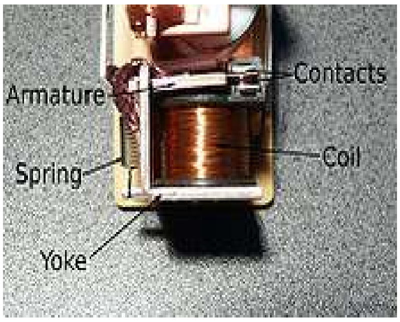


FIG2.7:Relay

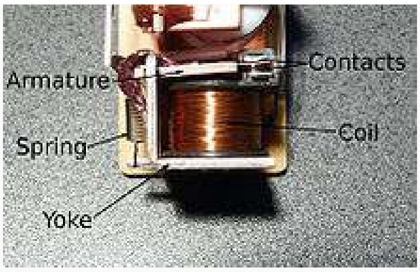
Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits as amplifiers: they repeated the signal coming in from one



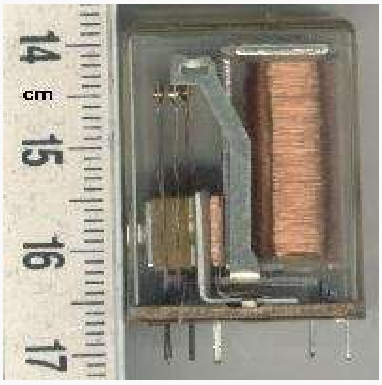
circuit and re-transmitted it on another circuit. Relayswere used extensively in telephone exchanges and early computers to perform logical operations.

A type of relay that can handle the high power required to directly control an electric motor or other loads is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protective relays".

Basic design and operation:



Simple electromechanical relay.



Small "cradle" relay often used in electronics. The "cradle" term refers to the shape of the relay's armature.

A simple electromagnetic relay consists of a coil of wire wrapped around a soft iron core, an iron yoke which provides a low reluctance path for magnetic flux, a movable iron armature, and one or more sets of contacts (there are two in the relay pictured). The armature is hinged to the yoke and mechanically linked to one or more sets of moving contacts. It is held in place by a spring so that when the relay is de-energized there is an air gap in the magnetic circuit. In this condition, one of the two sets of contacts in the relay pictured is closed, and the other set is open. Other relays may have more or fewer sets of contacts depending on their function. The relay in the picture also has a wire connecting the armature to the yoke. This ensures continuity of the circuit between the moving contacts on the armature, and the circuit track on the printed circuit board (PCB) via the yoke, which is soldered to the PCB.

When an electric current is passed through the coil it generates a magnetic field that activates the armature, and the consequent movement of the movable contact(s) either makes or breaks (depending upon construction) a connection with a fixed contact. If the set of contacts was closed when the relay was de-energized, then the movement opens the contacts and breaks the connection, and vice versa if the contacts were open. When the current to the coil is switched off, the armature is returned by a force, approximately half as strong as the magnetic force, to its relaxed position. Usually this force is provided by a spring, but gravity is also used commonly in industrial motor starters. Most relays are manufactured to operate quickly. In a low-voltage application this reduces noise; in a high voltage or current application it reduce sarcing.

When the coil is energized with direct current, a diode is often placed across the coil to dissipate the energy from the collapsing magnetic field at deactivation, which would otherwise generate a voltage spike dangerous to semiconductor circuit components. Some automotive relays include a diode inside the relay case. Alternatively, a contact protection network consisting of a capacitor and resistor in series (snub ber circuit) may absorb the surge. If the coil is designed to be energized with alternating current (AC), a small copper "shading ring" can be crimped to the end of the solenoid, creating a small out-of-phase current which increases the minimum pull on the armature during the AC cycle

Chapter-3

Temperature Sensor

3.1 Introduction:

A passive infrared sensor (PIR sensor) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors.

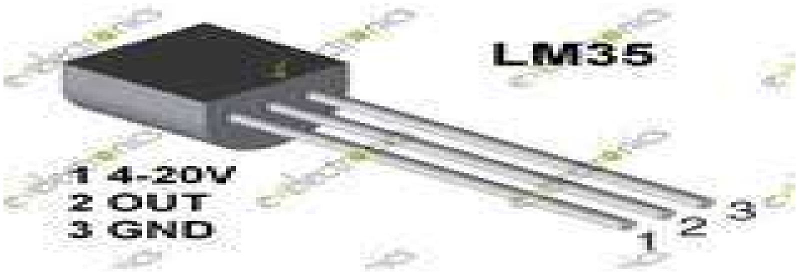


Fig 3.1: Temperature Sensor

All objects with a temperature above absolute zero emit heat energy in the form of radiation. Usually this radiation is invisible to the human eye because it radiates at infrared wavelengths, but it can be detected by electronic devices designed for such a purpose.

The term passive in this instance refers to the fact that PIR devices do not generate or radiate any energy for detection purposes. They work entirely by detecting the energy given off by other objects. It is important to note that PIR sensors don't detect or measure "heat" per se; instead they detect the Infrared radiation emitted from an object which is different from but often associated/correlated with the object's temperature (e.g., a detector of X-rays or gamma rays would not be considered a heat detector, though high temperatures may cause the emission of X or gamma radiation).

3.2 Working Process

Infrared radiation enters through the front of the sensor, known as the 'sensor face'. At the core of a PIR sensor is a solid state sensor or set of sensors, made from piezoelectric materials—materials which generate energy when exposed to heat. Typically, the sensors are approximately 1/4 inch square (40 mm^2), and take the form of a thin film. Materials commonly used in PIR sensors include gallium nitride (GaN), cesium nitrate ($CsNO_3$), polyvinyl fluorides, derivatives of phenyl pyridine, and cobalt phthalocyanine. The sensor is often manufactured as part of an integrated circuit.

3.3 Circuit Diagram

The PIR sensor is typically mounted on a printed circuit board containing the necessary electronics required to interpret the signals from the sensor itself. The complete assembly is usually contained within housing, mounted in a location where the sensor can cover area to be monitored.

The housing will usually have a plastic "window" through which the infrared energy can enter. Despite often being only translucent to visible light, infrared energy is able to reach the sensor through the window because the plastic used is transparent to infrared radiation. The plastic window reduces the chance of foreign objects (dust, insects, etc.) from obscuring the sensor's field of view, damaging the mechanism, and/or causing false alarms. The window may be used as a filter, to limit the wavelengths to 8-14 micrometers, which is closest to the infrared radiation emitted by humans. It may also serve as a focusing mechanism; see below.

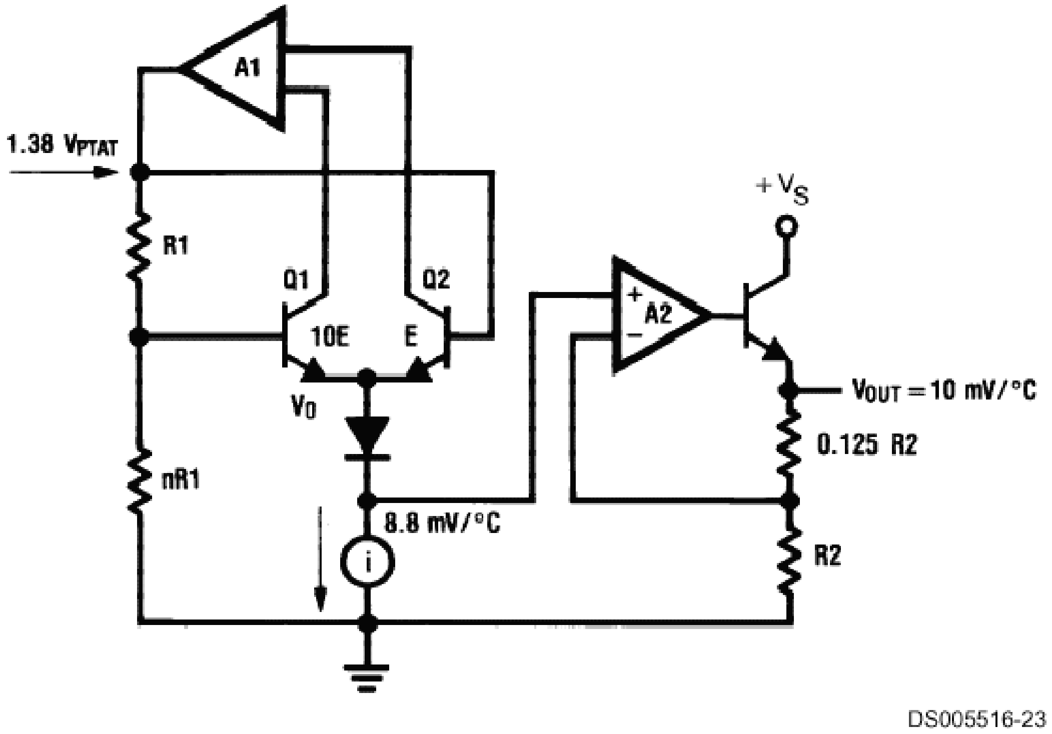
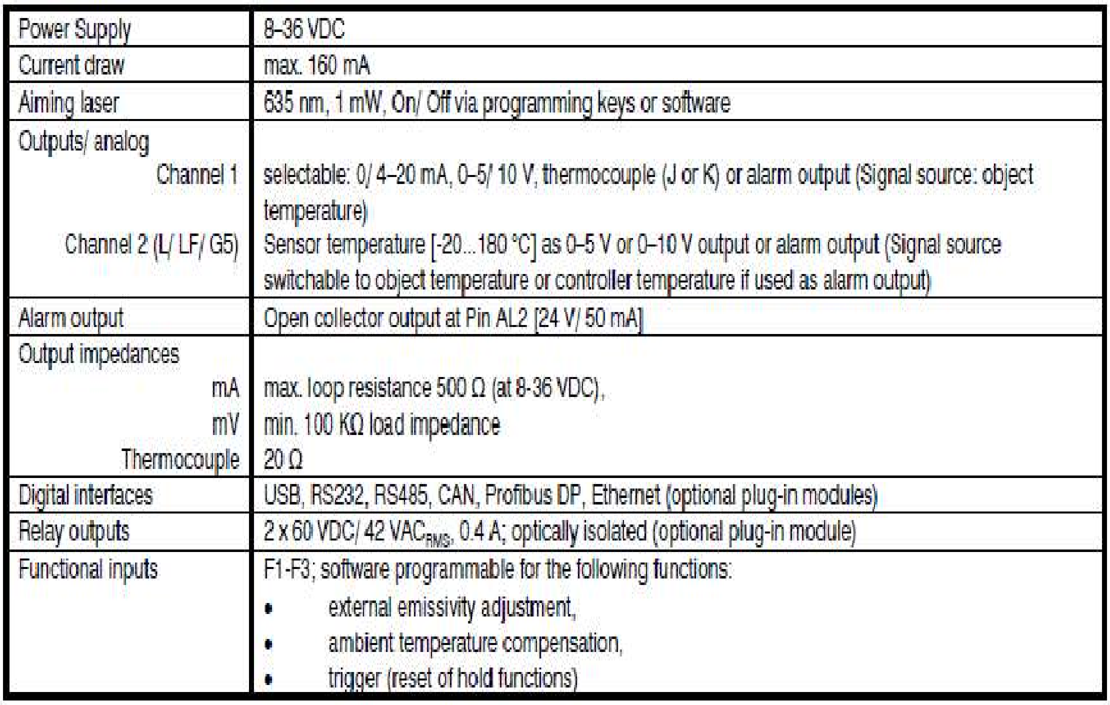


Fig: 3.2 Circuit Diagram of Sensor

3.4 Applications

- Industrial sector
- Different scientific analysis
- Institutional laboratory works

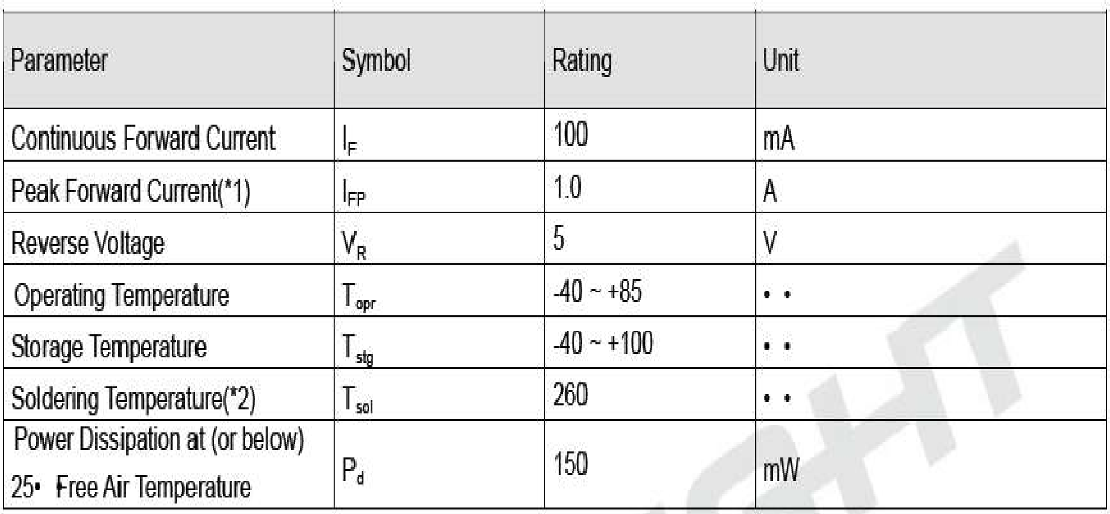
3.5 Specifications



Power Supply	8–36 VDC
Current draw	max. 160 mA
Aiming laser	635 nm, 1 mW, On/ Off via programming keys or software
Outputs/ analog	
Channel 1	selectable: 0/ 4–20 mA, 0–5/ 10 V, thermocouple (J or K) or alarm output (Signal source: object temperature)
Channel 2 (L/ LF/ G5)	Sensor temperature [-20...180 °C] as 0–5 V or 0–10 V output or alarm output (Signal source switchable to object temperature or controller temperature if used as alarm output)
Alarm output	Open collector output at Pin AL2 [24 V/ 50 mA]
Output impedances	
mA	max. loop resistance 500 Ω (at 8-36 VDC),
mV	min. 100 KΩ load impedance
Thermocouple	20 Ω
Digital interfaces	USB, RS232, RS485, CAN, Profibus DP, Ethernet (optional plug-in modules)
Relay outputs	2 x 60 VDC/ 42 VAC_{RMS}, 0.4 A; optically isolated (optional plug-in module)
Functional inputs	F1-F3; software programmable for the following functions: • external emissivity adjustment, • ambient temperature compensation, • trigger (reset of hold functions)

Table: 3.1 Specification of Temperature Sensor

3.6 Rating



Parameter	Symbol	Rating	Unit
Continuous Forward Current	I_F	100	mA
Peak Forward Current(*1)	I_{FP}	1.0	A
Reverse Voltage	V_R	5	V
Operating Temperature	T_{opr}	-40 ~ +85	• •
Storage Temperature	T_{stg}	-40 ~ +100	• •
Soldering Temperature(*2)	T_{sol}	260	• •
Power Dissipation at (or below) 25• Free Air Temperature	P_d	150	mW

Table: 3.2 Rating of the Parameter

Chapter-4

ATMEGA32- Microcontroller

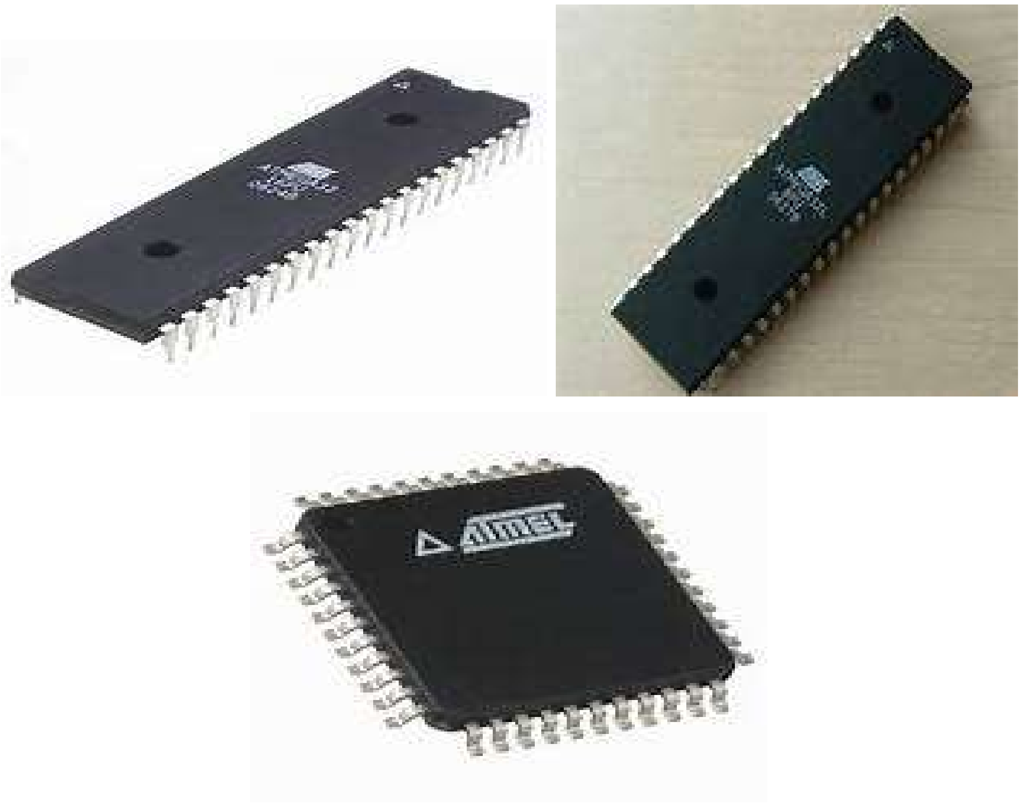


Fig 4.1 ATMEGA32 Microcontroller

4.1 Microcontroller Core Features

Features:

• High-performance, Low-power Atmel ®AVR® 8-bit Microcontroller.

• Advanced RISC Architecture

–131 Powerful Instructions – Most Single-clock Cycle Execution

–32 × 8 General Purpose Working Registers

–Fully Static Operation

–Up to 16 MIPS Throughput at 16MHz

–On-chip 2-cycle Multiplier

• High Endurance Non-volatile Memory segments

–32Kbytes of In-System Self-programmable Flash program memory

–1024Bytes EEPROM

–2Kbytes Internal SRAM

–Write/Erase Cycles: 10,000 Flash/100,000 EEPROM

–Data retention: 20 years at 85°C/100 years at 25°C(1)

–Optional Boot Code Section with Independent Lock Bits

In-System Programming by On-chip Boot Program True Read-While-Write Operation

–Programming Lock for Software Security

• JTAG (IEEE std. 1149.1 Compliant) Interface

–Boundary-scan Capabilities According to the JTAG Standard

–Extensive On-chip Debug Support

–Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface

• Peripheral Features

–Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes

–One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode

–Real Time Counter with Separate Oscillator

–Four PWM Channels

–8-channel, 10-bit ADC 8 Single-ended Channels 7 Differential Channels in TQFP Package Only 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
–Byte-oriented Two-wire Serial Interface
–Programmable Serial USART
–Master/Slave SPI Serial Interface
–Programmable Watchdog Timer with Separate On-chip Oscillator
–On-chip Analog Comparator

• Special Microcontroller Features
–Power-on Reset and Programmable Brown-out Detection
–Internal Calibrated RC Oscillator
–External and Internal Interrupt Sources
–Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby

• I/O and Packages
–32 Programmable I/O Lines
–40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
• Operating Voltages

–2.7V - 5.5V for ATmega32L
–4.5V - 5.5V for ATmega32

• Speed Grades
–0 - 8MHz for ATmega32L
–0 - 16MHz for ATmega32

• Power Consumption at 1MHz, 3V, 25°C
–Active: 1.1mA
–Idle Mode: 0.35mA
–Power-down Mode: < 1µA

4.2: Peripheral Features

• Timer0: 8-bit timer/counter with 8-bit presale

• Timer1: 16-bit timer/counter with presale, can be incremented during SLEEP via

External crystal/clock

• Timer2: 8-bit timer/counter with 8-bit period register, presales and postscaler

• Two Capture, Compare, PWM modules

- Capture is 16-bit, max .resolution is 12.5 ns

- Compare is 16-bit, max .resolution is 200 ns

- PWM max .resolution is 10-bit

• 10-bit multi-channel Analog-to-Digital converter

• Synchronous Serial Port (SSP) with SPI (Master mode) and I2C(Master/Slave)

• Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection

‘• Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)

• Brown-out detection circuitry for Brown-out Reset (BOR)

4.3 Pin configur:

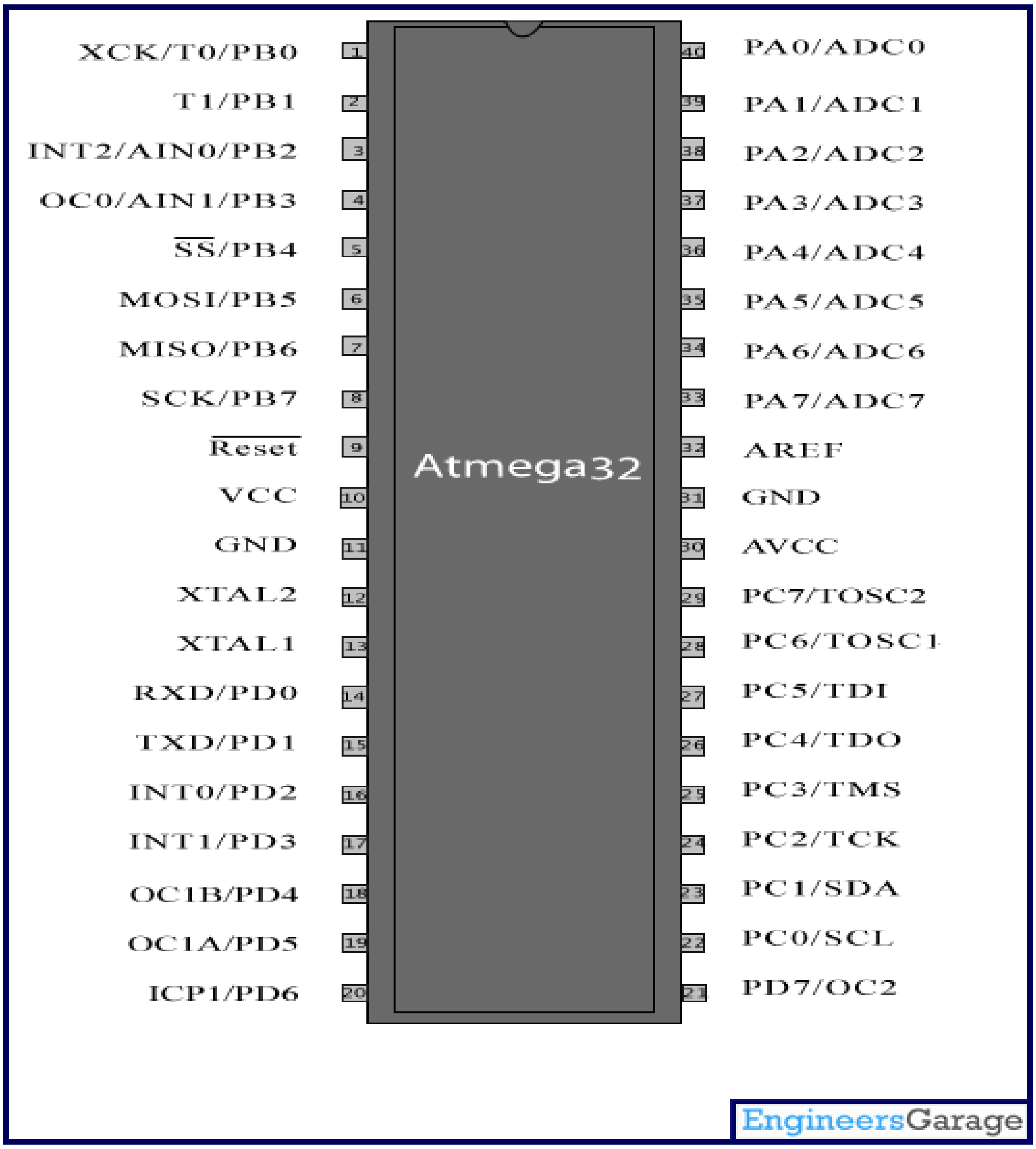


Fig 4.2: Pin Configaration

4.4:AVR ATMEGA32- 40 pin Microcontroller Block Diagram:

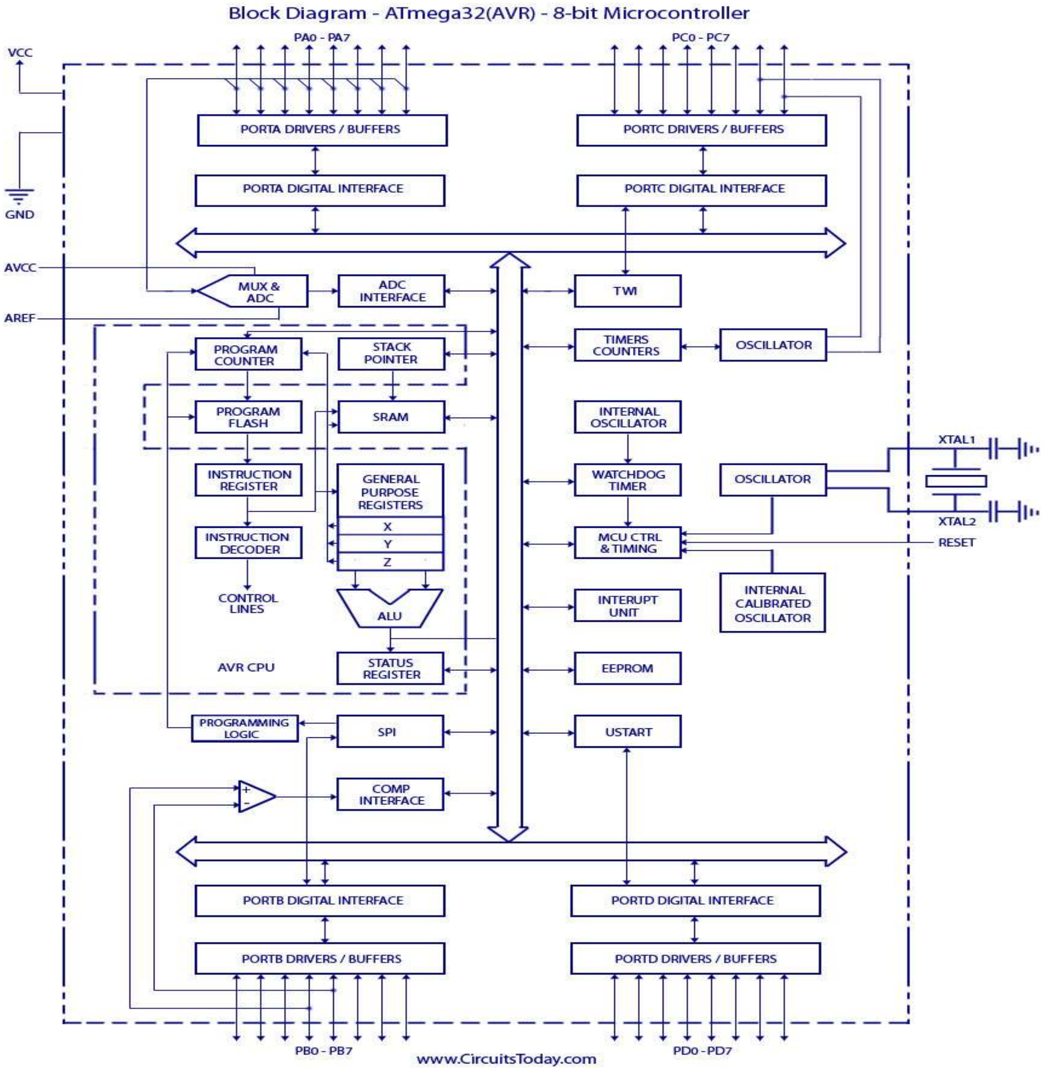


Fig: 4.3: Block Diagram of Microcontroller

4.5 Pin Description

VCC Digital supply voltage.

GND Ground.

Port A (PA7..PA0) Port A serves as the analog inputs to the A/D Converter.
Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used.
Port pins can provide internal pull-up resistors (selected for each bit).
The Port A output buffers have sym metrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.5

Port B (PB7..PB0) Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The
Port B output buffers have symmetrical drive characteristics with both high sink and source
capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up
resistors are activated. The Port B pins are tri-stated when a reset condition becomes active,
even if the clock is not running.
Port B also serves the functions of various special features of the ATmega32

Port C (PC7..PC0) Port C is an 8-bit bi-directional I/O port with internal pull-up resistors
ThePort C output buffers have symmetrical drive characteristics with both high sink and source

capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.

The TD0 pin is tri-stated unless TAP states that shift out data are entered.

Port C also serves the functions of the JTAG interface and other special features of theATmega32

Port D (PD7..PD0) Port D is an 8-bit bi-directional I/O port with internal pull-up resistors ThePort D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega32

RESET Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 37. Shorter pulses are not guaranteed to generate a reset.

XTAL1 Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2 Output from the inverting Oscillator amplifier.

AVCC AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally con-nected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.

AREF AREF is the analog reference pin for the A/D Converter

4.6 Overview The Atmel

®AVR®ATmega32 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega32 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

This document contains device specific information. Additional information may be found in the PIC micro™ Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip website. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

The user must ensure that the page select bits are programmed so that the desired program memory page is addressed. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is popped off the stack. Therefore, manipulation of the PCLATH<4:3> bits is not required for the return instructions (which POPs the address from the stack).

Chapter-5

Alphanumeric Display

5.1 Introduction

Alphanumeric displays are used in a wide range of applications, including palmtop computers, word processors, photocopiers, point of sale terminals, medical instruments, cellular phones, etc. the 16x2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. A full list of the characters and symbols is printed on pages 7/8 (note these symbols can vary between brand of LCD used).This booklet provides all the technical specifications for connecting the unit, which requires a single power supply (+5 V).

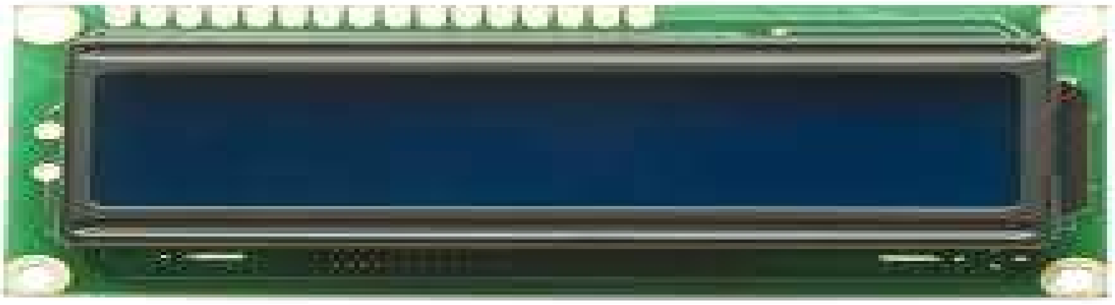


Fig: 5.1Alphanumeric Display

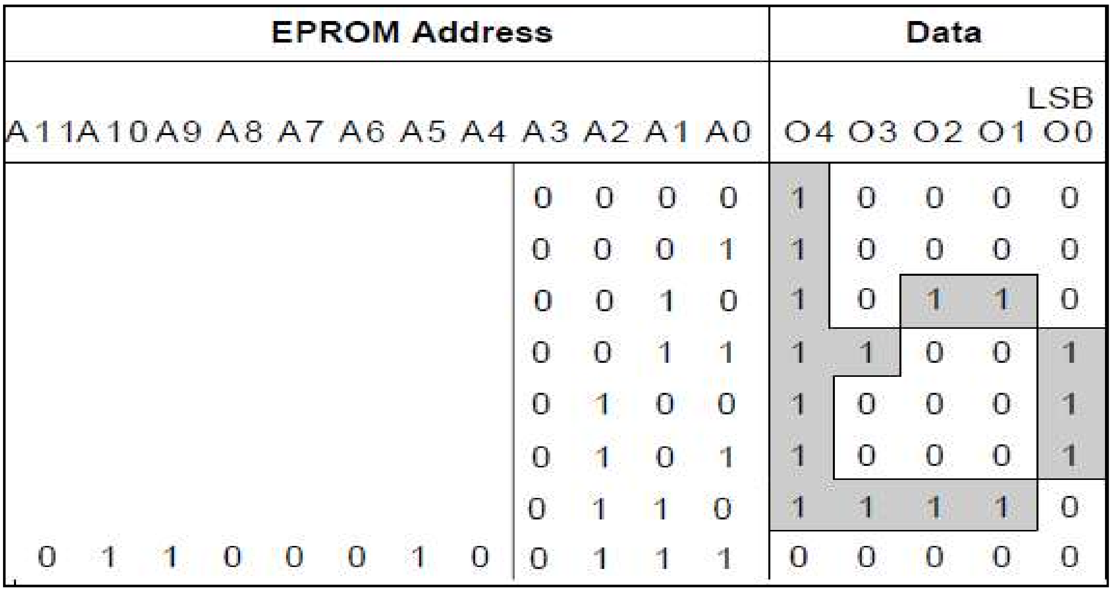
5.2 Further information

An LCD (Liquid Crystal Display) basically works on the concept of Light Polarization of a ‘Liquid Crystal’ under the influence of an Electric Field. Every LCD contains a Back-Light behind the Liquid Crystal array, which acts as a light source. When an Electric Field is applied across certain fluids, it changes the way they allow light to pass through them, that is, it changes the orientation of the liquid crystal molecules as a result they do not allow light to pass through them. Hence, by applying suitable potential difference, we can control if light passes or doesn't pass through the LCD pixels.

The LCD is then divided into small pixels, the smallest partition of the screen, which can be controlled and polarized. A pixel is just a small dot on the screen. By combing a set of nearby pixels (dots), blocks are made, example, in a 16x2 LCD one block has 5x8 dots (or 5x10 dots in another configuration). The advantage in these kinds of LCDs is that the controller already has stored patterns of dots, forming legible alphabets or numbers, in its EEPROM. By sending suitable commands to the LCD controller, we can call the stored patterns to be displayed on the screen, hence not requiring us to explicitly state the patterns in our code.

This also means, we can't make our own set of patterns, and have to use the exiting stored ones only

5.3 Code Variation



EPROM Address												Data				
																LSB
A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	O4	O3	O2	O1	O0
								0	0	0	0	1	0	0	0	0
								0	0	0	1	1	0	0	0	0
								0	0	1	0	1	0	1	1	0
								0	0	1	1	1	1	0	0	1
								0	1	0	0	1	0	0	0	1
								0	1	0	1	1	0	0	0	1
								0	1	1	0	1	1	1	1	0
0	1	1	0	0	0	1	0	0	1	1	1	0	0	0	0	0

Table: 5.1 Code Variation of LCD Display

5.4 Pin Description:

Pin 1 – Vss (Common Source of Electrons i.e. Ground) 0 V

Pin 2 - Vdd (Common Drain for Electrons i.e. High Voltage) 5 V

Pin 3 - Vee (Common Emitter – used to control Contrast) Ground via resistor

Pin 4 – RS (Register Select) Used to tell LCD controller weather data on Data Bus is a Command

Or Text Data .High Signal for Data and Low Signal for Command.

Pin 5 – R/W (Read Write) Used to tell LCD controller, if we wish to Read or Write to the LCD.

High for Read, low for Write.

Pin 6 – E (Enable) Used for Enabling and Disabling the LCD

Pin 7 to 14 – LCD Data Bus

Pin 15 – Positive Terminal for Backlight (5V)

Pin 16 – Negative Terminal for Backlight (0V)

Provide Ground to Pin.1 (Vss), +5V to Pin.2 (Vdd) and a 2.5k Resistor (or a POT) to the Pin.3 (Vee). Connect the Pin.4 (RS) to PD.5, Pin.5 (R/W) to PD.6 and Pin.6 (E) to PD.7

Since we will be only writing to the LCD, you can also just ground the R/W pin. Give +5V and Ground to the LCD back-light supply (Pin.15 and Pin.16).

5.5 Outline Dimension and Block Diagram:

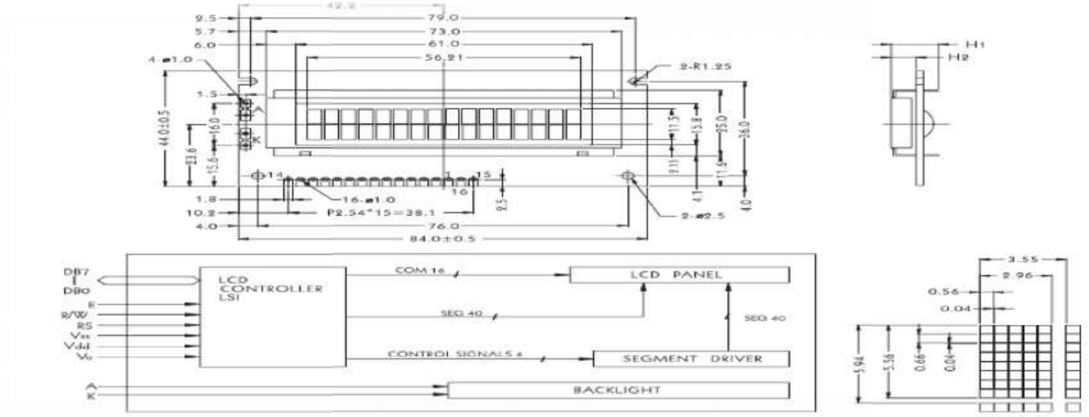


Fig: 5.2 Outline Dimension and Block Diagram

5.6 Electrical Characteristics:

Item	Symbol	Condition	Standard Value			Unit	Applicable
Power Voltage	VDD		4.5	5.00	5.5	V	Vdd
Input H. Level Voltage	VIH		2.2	-	Vdd	V	RS,R/W.E DBO□- DB7
Input L. Level Voltage	VIL		-0.3		0.6	V	
Output H-Level Voltage	VOH	_IOH=0.205 mA	2.4		-	V	DBO□- DB7
Output L-Level Voltage	VOL	IoL=1.2mA	-		0.4	V	RS,R/W,E DBO-DB7
I/O Leakage Current	IIL	Vin＝0 –Vdd	-1		1.0	µ A	
Supply Current	IDD	Vdd＝5V	2		-	m A	Vdd

Table: 5.2 Electrical Characteristic of Numeric Display

5.7 Timing Characteristic:

Item		Symbol	Min	Max	Unit
Enable cycle time		TCYCE	500	-	Ns
Enable pulse width	High level	PWEH	220	-	Ns
Enable rise/fall time		TER.TE F	-	25	Ns
Setup time	RS.R/W	TAS	40	-	ns
Address hold time		TAH	10	-	Ns
Data set up time		TDSH	60	-	Ns
Data delay time		TDDR	60	120	Ns
Data hold time(writing)		TH	10	-	Ns
Data hold time(reading)		TDHR	20	-	Ns
Clock oscillating frequency		TOSC	270(TYP)		KHz

Table: 5.3 Timing Characteristic of Alphanumeric Display

5.8 Interfacing with MPU:

Interfacing with 4-bit MPU can be made through I/0 port of 4-bit Maui there are enough I/0 Ports, at Can be transferred by 8-bit , however ,if there are not data transfer can be done by 4-bi in twice (select Interface is 4-bit long),and sequence will be complicated in this case. Please take into account that 2 cycles of BF check is necessary, While 2 cycles of data transfer are also necessary.

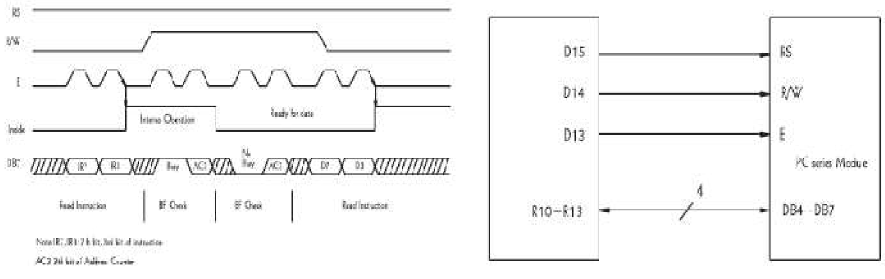


Fig: 5.3 Interfacing Diagram with MPU

Features of Interfacing

1. Interface with 8-bit MPU is available.

2.192 Kind of alphabets, numerals, symbols and special characters can be displayed by built-in character generator (RAM).

3. Other preferred characters can be displayed by character generator.

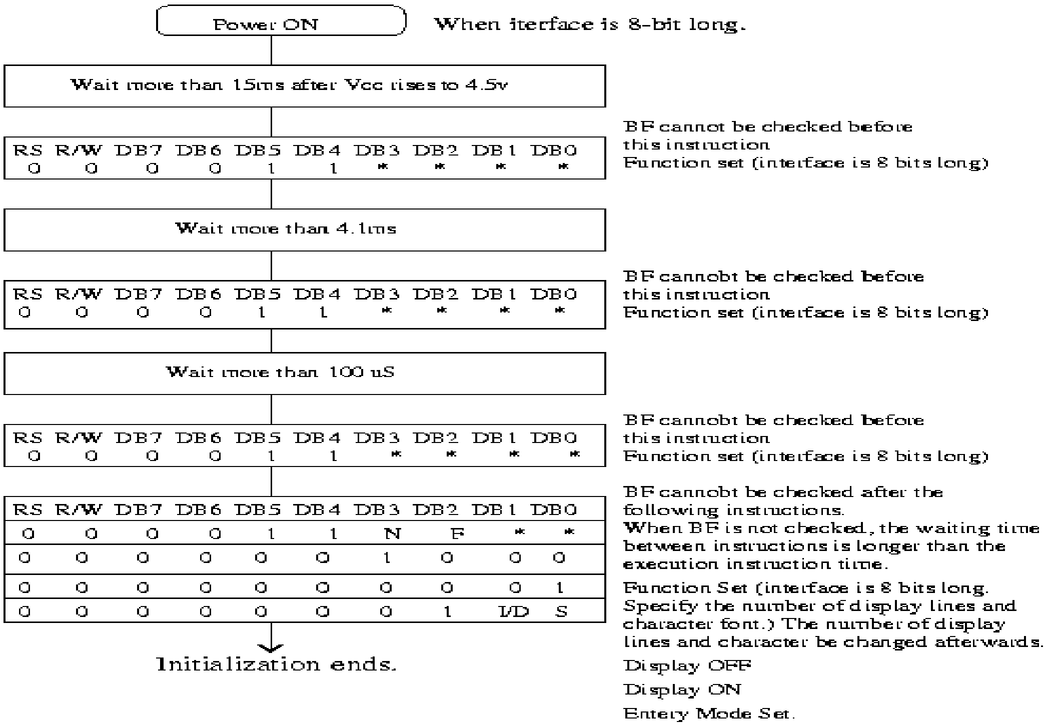
4. Various functions of instruction are available by programming such as clear display, cursor at home, On/of cursor, Blink character, shift display shift cursor read/write display data…etc.

5. Single power supply +5 V drive (except for extended type).

6Compact and light weight design which can be easily assembled in devices.

5.9 Initialization

If power supply conditions are not satisfied, which for proper operation of internal rest Circuit, It is required to make initialization along with instruction. Make following procedures.


Power ON
When iterface is 8-bit long.
Wait more than 15ms after Vcc rises to 4.5v
BF cannot be checked before this instruction
Function set (interface is 8 bits long)
Wait more than 4.1ms
BF cannobt be checked before this instruction
Function set (interface is 8 bits long)
Wait more than 100 uS
BF cannobt be checked before this instruction
Function set (interface is 8 bits long)
BF cannobt be checked after the following instructions.
When BF is not checked, the waiting time between instructions is longer than the execution instruction time.
Function Set (interface is 8 bits long. Specify the number of display lines and character font.) The number of display lines and character be changed afterwards.
Display OFF
Display ON
Entery Mode Set.
Initialization ends.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	*	*	*	*

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	*	*	*	*

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	*	*	*	*

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	N	F	*	*
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	I/D	S

Fig: 5.4 Instruction of Installation Process Numeric Display

Chapter-6

Voltage Regulator

6.1 Description :

The LM7085 is a series of low dropout positive voltage regulators with a maximum dropout of 1.5V at 3A of load current. It has the same pin-out as TI's industry standard LM317. The LM1085 is available in an adjustable version, which can set the output voltage with only two External resistors,

It is also available in three fixed voltages: 3.3V, 5.0V and 12.0V. The fixed versions integrate the adjust resistors. The LM1085 circuit includes a zener trimmed band gap reference, current limiting and thermal shutdown. The LM1085 series is available in TO-220 and DDPAK/TO-263 packages. Refer to the LM1084 for the 5A version, and the LM1086 for the 1.5A version

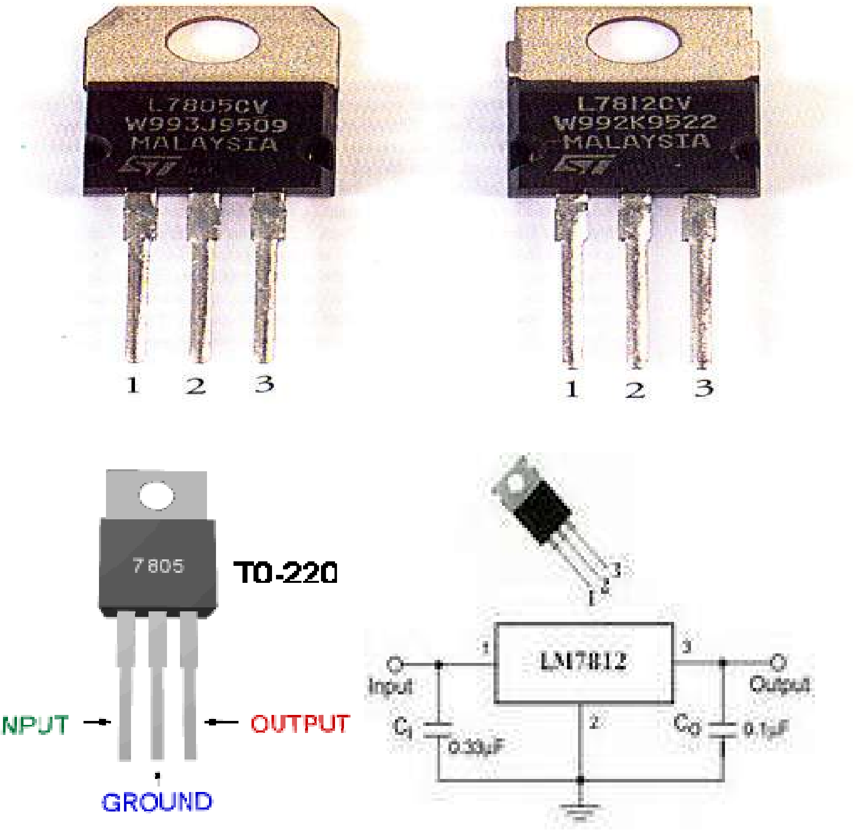


Fig: 6.1 LM7805 And LM7812 Voltage Regulators

6.2 Applications

1. Industrial sector
2. Different scientific analysis.
3. Institutional laboratory works.

6.3Connection Diagram

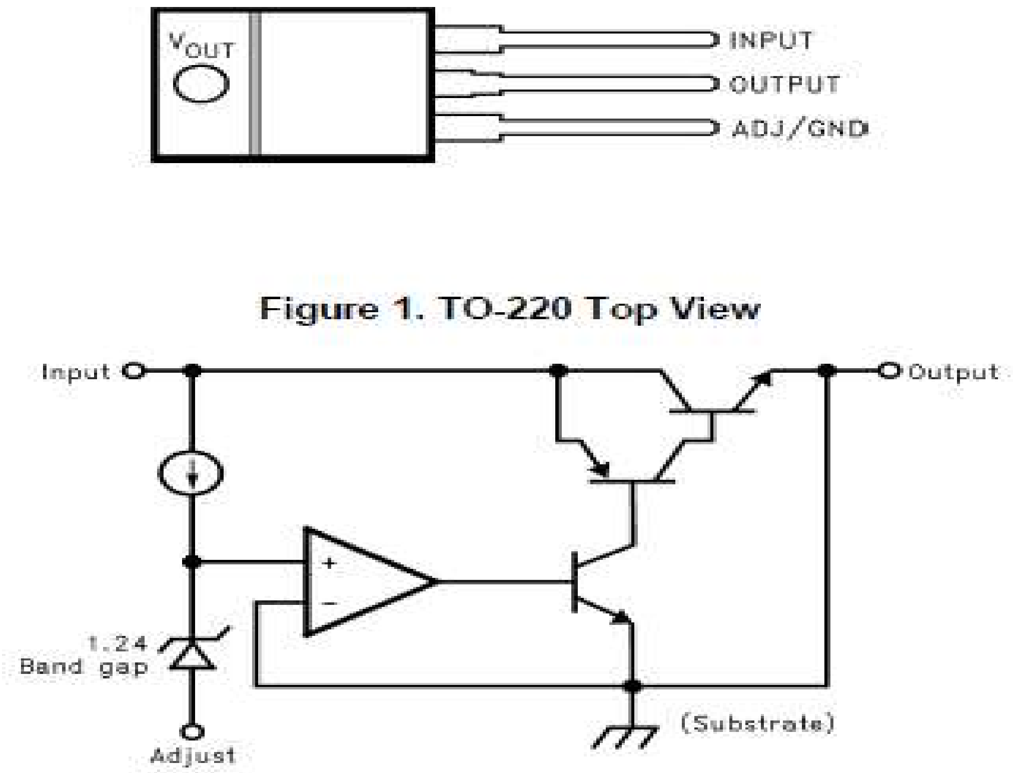


Figure 1. TO-220 Top View

Fig: 6.2 Connection Diagram of LM7805

6.4 Specifications:

Absolute Maximum Ratings(1) (2)	Maximum Input to Output Voltage Differential
LM1085-ADJ	29V
LM1085-12	18V
LM1085-3.3	17V
LM1085-5.0	25V
Power Dissipation (3)	Internally Limited
unction Temperature (TJ)(4)	150°C
Storage Temperature Range -	65°C to 150°C
Lead Temperature	260°C, to 10 sec
ESD Tolerance (5)	2000V

Table: 6.1 Specification of LM 7805 Voltage Regulator

Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is intended to be functional, but specific performance is not ensured. For ensured specifications and the test conditions, see the Electrical Characteristics.

If Military/Aerospace specified devices are required, please contact the Texas Instruments Sales Office/ Distributors for availability and Power dissipation is kept in a safe range by current limiting circuitry. Refer to Overload Recovery in Application

The maximum power dissipation is a function of TJ (max),èJA, and TA. The maximum allowable power dissipation at any ambient temperature is PD = (TJ (max)–T A)/èJA. All numbers apply for packages soldered directly into a PC board. Refer to Thermal Considerations in the Application Notes.For testing purposes, ESD was applied using human body model, 1.5kÙ in series with 100uF.

6.5 Electrical Characteristics: Limits in standard type are for TJ = 25°C only; limits in boldface type apply over the operating junction temperature (TJ) range of -40°C to +125°C.

Symbol	Parameter	Conditions	Min(1)	Typ(2)	Typ(2)	Units
VREF	Reference Voltage(3)	LM1085-ADJ IOUT = 10mA, VIN−VOUT = 3V10mA ≤IOUT ≤ IFULL LOAD,1.5V ≤ (VIN−VOUT) ≤ 15V	1.238 1.225	1.250 1.250	1.262 1.270	V
VOUT	Output Voltage(3)	LM1085-3.3 IOUT = 0mA, VIN = 5V 0 ≤ IOUT ≤IFULL LOAD, 4.8V≤ VIN ≤15V	3.270 3.235	3.300 3.300	3.330 3.365	V
		LM1085-5.0 IOUT = 0mA, VIN = 8V 0 ≤ IOUT ≤ IFULL LOAD, 6.5V ≤ VIN ≤ 20V	4.950 4.900	5.000 5.000	5.050 5.100	V
		LM1085-12 IOUT = 0mA, VIN = 15V 0 ≤ IOUT ≤ IFULL LOAD, 13.5V ≤ VIN ≤ 25V	11.880 11.760	12.000 12.000	12.120 12.240	V

Table: 6.2 Electrical Characteristic of LM7085 Voltage Regulator

Minimum and Maximum limits are ensured through test, design, or statistical correlation. Typical values represent the most likely parametric norm at TJ = 25°C, and are provided for reference purposes only

All limits are specified by testing or statistical analysis. Typical Values represent the most likely parametric norm. If full load is defined in the current limit curves .The current limit as a function of input-to-output voltage s. Note that 30W power dissipation for the LM1085 is only achievable over a limited range of input-to-output voltage.

Chapter-7

Program

7.1 Introduction:

The purpose of this document is to analyze the selection criteria for memory chips to be used in Spacecraft computers. A general trend is to implement more autonomous functions in Spacecraft, making use of increased processor performances, but requiring larger embedded flight software sizes and larger storage on board. This note provides an introduction to the current organization of European Spacecraft On-Board Data

Handling systems and identifies the criticality of Spacecraft computer based functions. It recalls the constraints on memory planes linked to the Space environments, and gives examples of memory banks designs for typical sizes. In order to introduce the context of Space projects, the following figure represents a logic flow diagram of a typical Phase A /Phase B for a Spacecraft, where the most important trade-offs in term of Data Handling architecture and components selection have to be performed.

Both Hardware and Software activities are represented, due to a strong interaction of both designs necessary to achieve the most appropriate processing system. Design of current spacecraft on board data handling subsystem. Starting from an approved mission concept, the Spacecraft has to be designed to meet a number of operational and environmental constraints. A preliminary On-Board Data Handling architecture is defined, based on the experience gained by the industry, with a parallel allocation of hardware and software functions.

This phase A work relies on existing hardware, re-utilization of standard functions and typical software The software functions are then detailed to size more precisely the necessary resources to fulfill the Spacecraft mission, and an interactive work at subsystem level is necessary to match hardware and software functions. The Spacecraft design is then reviewed at system level to check that it will meet the mission objectives, especially taking into account Space environment.

Issues are then identified (mass, reliability, thermal aspects) and corrective actions taken as necessary for each equipment or subsystem. For that concern RAM banks, they have a particular importance at system level, since they contain most of flight software functions. Single event upsets have now been identified since early 80s and corrective hardware.

Methods have been implemented to avoid catastrophic effects. The main effect remains some degradation of the Spacecraft availability, whose impacts on the mission have to be assessed. To summarize, the steps in the selection process for memories in Space applications,

Are:

_ PRELIMINARY ARCHITECTURE

_ SIZING OF COMPUTER RESOURCES

_ ANALYSIS OF SEU EFFECTS ON MISSION

7.2 Instructions:

The C High-level Language (HLL) has become increasingly popular for programming microcontrollers.

The advantages of using C compared to assembler are numerous:

Reduced development time,

Easier maintainability and portability and

Easier reuse code.

The penalty can be larger code size and as a result of that often reduced speed. To reduce these penalties AVR architecture is tuned to efficiently decode and execute instructions that are typically generated by C compilers.

The C Compiler development was done by IAR systems before the AVR architecture and instruction set specifications were completed. The result of the co-operation between the compiler development team and the AVR development team is a microcontroller for which highly efficient, high performance code is generated. This application note describes how to utilize the advantages of the AVR architecture and the development tools to achieve more efficient C code than for any other microcontroller.

7.3 Calculation and Programming:

Since we know that the microcontroller function based on programming interface, so if we use it according to our application must follow its language, so prepared our programming language with C.

Now we use the CVR (code vision avr) software for write the program, like the fig below

Program

```
#include<avr/io.h>
#include<util/delay.h>
#include<avr/interrupt.h>

#include"lcd_lib.h"

#include<stdlib.h>

#define LOAD_A_ON PORTD|=(1<<PD0)
```

```
#define LOAD_A_OFF PORTD&=~(1<<PD0)

#define LOAD_B_ON PORTD|=(1<<PD1)

#define LOAD_B_OFF PORTD&=~(1<<PD1)

#define LOAD_C_ON PORTD|=(1<<PD2)

#define LOAD_C_OFF PORTD&=~(1<<PD2)

#define LOAD_D_ON PORTD|=(1<<PD3)

#define LOAD_D_OFF PORTD&=~(1<<PD3)

int adc_read(uint8_t ch);

void adc_init(void);

void Take_action(int tem);

int main(void)

{

        DDRD=0XFF;

          long int x,t;

          char c[20];


                    LCDinit();

                    LCDclr();

                    LCDhome();

                    LCDstring("DAFFODIL INTERNATIONAL UNIVERSITY");

                    LCDshiftRight(33);

                    LCDclr();

                    LCDhome();
                    LCDstring("BSC IN EEE         ");

                    LCDGotoXY(0,1);

                    LCDstring("BATCH NO :08       ");

                    _delay_ms(1000);

                    LCDstring("MD. MONJUR HOSSAIN  ");

                    LCDGotoXY(0,1);

                    LCDstring("ID: 121-33-837 ");

                    _delay_ms(1000);

                    LCDstring("MD. MAHAMUDUL HASAN  ");

                    LCDGotoXY(0,1);

                    LCDstring("ID: 121-33-851 ");

                    _delay_ms(1000);

                    LCDclr();

                    LCDhome();




                adc_init();

                    LCDhome();

                    while(1)

                    {

                            LCDGotoXY(0,0);

                            LCDstring("tempareture=  C");

                            x=adc_read(0);
```

```
            t=500*x/1024;
            itoa(t,c,10);
            LCDGotoXY(12,0);
            LCDstring(c);
            LCDGotoXY(15,0);
            LCDstring("  ");
            Take_action(t);
            _delay_ms(200);
                        }

            }

void Take_action(int tem)
{
    if(tem<20)

    {
        LOAD_A_ON;
        LOAD_B_OFF;
        LOAD_C_OFF;
        LOAD_D_OFF;

        LCDGotoXY(0,1);
        LCDstring(" Heater is ON ");

    }

    else if(tem>=25 && tem<30)

    {

       LOAD_A_OFF;

        LOAD_B_ON;

        LOAD_C_OFF;

        LOAD_D_OFF;



        LCDGotoXY(0,1);

        LCDstring(" Fan is on     ");



    }



    else if(tem>=30 && tem<35)

    {

        LOAD_A_OFF;

        LOAD_B_ON;

        LOAD_C_ON;

        LOAD_D_OFF;



        LCDGotoXY(0,1);

        LCDstring("Fan & AC are on     ");
```

```
          }


          else if(tem>=35 && tem<40)

          {

                    LOAD_A_OFF;

                    LOAD_B_ON;

                    LOAD_C_ON;

                    LOAD_D_ON;



                    LCDGotoXY(0,1);

                    LCDstring(" Alerm is on     ");

                              }



          }

void adc_init(void)

{

   // AREF = AVcc

   ADMUX = (1<<REFS0);

   // ADC Enable and prescaler of 128

   // 16000000/128 = 125000

   ADCSRA = (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);

}
```

```
int adc_read(uint8_t ch)

{

  // select the corresponding channel 0~7

  // ANDing with ’7? will always keep the value

  // of ‘ch’ between 0 and 7

  ch &= 0b00000111;  // AND operation with 7

  ADMUX = (ADMUX & 0xF8)|ch; // clears the bottom 3 bits before ORing

  // start single convertion

  // write ’1? to ADSC

  ADCSRA |= (1<<ADSC);

  // wait for conversion to complete

  // ADSC becomes ’0? again

  // till then, run loop continuously

  while(ADCSRA & (1<<ADSC));

  return (ADC);

}
```

7.4 Simulation

After programmed look for simulation, that our made program work properly, before simulation run this program in microcontroller with machine device software to hardware. For simulation we use proteins software, as we design the approximate diagramed with using proteins ISIS software,

In the figure shown the position of CVR

Chapter-8

Component Data Sheet

8.1 Data Sheet :

Component for the Project

S.L	Name	value	Number
01	Capacitor	1000uF,22PF	1
02	Capacitor	22pf	5
03	Regulator ic	7812 7805	1+1
04	Lcd	16*2	1
05	Microcontroll;er	ATMEGA32	1
06	Realy	12v	4
07	Diode	1N40007	10
08	Crystal	12Mhz	1
09	Femal Rell		1
10	IC Base	40 pin 16 pin	1+1
11	Green Connector		10
12	Transformer	1 Amp	1
13	PCB	N/A	1+1
14	LED(RED)	N/A	1
13	Sensor	LM35	1
14	Register	220Ω,100Ω,1MΩ	3

8.2Price List:

Components	**Code/Value**	**Quantity**	**Unit Price**	**Estimated Cost**
Power supply	220v, 12v	1Pc	120	
Voltage Regulator	LM 7805 & LM7812	1+1Pc	45+35=80	
Capacitor	(1000uF,22pf)	1+5=6pcs	5x5+15=35	
Register	(220Ω,1kΩ,10kΩ)	3pcs	3x5=15	
Micro-controller	ATMEGA32	1Pc	250x1=250	
Alphanumeric LCD	16X2	1Pc	250x1=250	
Sensor (digital)	LM35	1Pc	50x1=60	
Connecting wire		10Ft	60	
LED (red)		1Pc	5x1=05	
PVC Board			30x1=30	
PCB(Vera board)			50x2=100	
Green Connector		10pc	10x10=100	
Reiay	12V	4pc	50x4=200	
Diode	1N40007	10pc	10x5=50	
Crystal	12MHz	1pc	10x10=100	
			Total=	1545/=
In word: One thousand and twenty taka only				

Chapter-9

Advantages and Disadvantage

9.1 Advantages of Temperature Control System

1. Easy to handle.

2. Good efficiency to measure accurate temperature.

3. High accuracy.

4. Low cost

9.2 Disadvantage of Temperature Control System

1. This device can measure only room temperature.
2. It can’t shutdown hole system.
3. It can’t operate under short circuit condition.

10.1 Data Table and Result

SL	TEMPERATURE	RELAY-1 HEATER	RELAY-2 COOLING FAN	RELAY -3 AIR CONDITIONER	RELAY-4 ALARMN
1	Tem<<20	ON	OFF	OFF	OFF
2	tem>=25 && tem<30	OFF	ON	OFF	OFF
3	tem>=30 && tem<35	OFF	ON	ON	OFF
4	tem>=35 && tem<40	OFF	ON	ON	ON

10.2 Discussions

1. After powering on the system, AVR has been adjusted until the LCD start showing up some text. Controlling temperature has been a prime objective in various applications including refrigerators, air conditioners, air coolers, heaters, industrial temperature conditioning and so on. Temperature controllers vary in their complexities and algorithms.
2. There are four relays use for the switching four loads according to the room temperature .And LCD always display the room temperature and four loads status.
3. In this project, ATMEGA32 Microcontroller, Transformer, sensor, LCD, have been used.
4. Total price of this project is 1545 taka and its market price is 5000 tk.

10.3 Conclusion

This designed project is very economical for using effective Temperature monitoring system for expensive and sophisticated machineries. For its cheap rate, small size and easy operating condition, it can be widely used at homes as well as industries**.**

10.4 Recommendation

If this project is manufactured commercially then the price will be reduced.

Chapter-11

Reference

1. Balagurusamy, Programming in ANSI C, 4th Edition, 2002 by Tata McGraw-Hill Publishing Company limited. Page no-115-116,277

2. B.L. Theraja, Electrical Technology, Volume II, First Multicolour Edition, 2008 by S.Chand Group & Company publication, Page 916-1006.

3. G. Jack Lipovski, Single and multi-Chip microcontroller interfacing, 1st Edition, 2000 by Academic press publication. Page-239

4. Pradeep Kumar Srivastava, Exploring Programmable logic Controllers with Applications, 2nd Revised *Edition,* 2004 by BPB Publications, Page No. 31-54
5. V.K. Mehta, Principle of electronics, 1st Edition, 1980 by S.Chand Group & Company publication, Page 136-245.
6. www.hpinfotech.com

7. www.atmel.com/dyn/resources/prod_documents/doc2486.pdf

8. www.futurlec.com/Atmel/ATMEGA8.shtml

9. www.housestuffworks.com

10. www.electronicsforyou.com

11. www.wikipedia.org