AIRCRAFT CABIN TEMPERATURE AND PRESSURE MANAGEMENT SYSTEM

A MAJOR PROJECT REPORT

Submitted by:

ARJUN PRATAP SINGH [R640209009]
RAJAT SAXENA [R640209031]
SAGAR VERMA [R640209035]

of

BACHELOR OF TECHNOLOGY

In

AVIONICS ENGINEERING

K

Under the Supervision of Prof. A.J ARUN JEYA PRAKASH

UNIVERSITY OF PETROLEUM & ENERGY STUDIES
विद्याया शक्ति वशम्
Harnessing Energy through Knowledge

DEPARTMENT OF AEROSPACE ENGINEERING
COLLEGE OF ENGINEERING STUDIES (CoES)
UNIVERSITY OF PETROLEUM & ENERGY STUDIES
DEHRADUN- 248007, INDIA

APRIL 2013

Dedicated to Our Guide

Prof. A.J Arun Jeya Prakash

&

Our Parents.

THESIS CERTIFICATE

We hereby certify that the work which is being presented in the project report entitled "**Aircraft Cabin Temperature and Pressure Management System**"in partial fulfillment of the requirements for the satisfactory performance for B.Tech Avionics Engineering, Major Project submitted in the Department of Aerospace Engineering, University of Petroleum and Energy Studies, Dehradun is an authentic record of our own work carried out during a period from July 2012 to April 2013.

SUBMITTED BY:

Arjun Pratap Singh [R640209009]
Rajat Saxena [R640209031]
Sagar Verma [R640209035]

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Prof. A.J Arun Jeya Prakash

Date: 16 April 2013

ACKNOWLEDGEMENTS

It is with a sense of great satisfaction and pride that we are sitting down to pen out our major project thesis report. First and foremost we sincerely salute our esteemed Institution **University of Petroleum & Energy Studies, Dehradun** for giving this golden opportunity for fulfilling our warm dreams of becoming a bachelor.

The first and foremost person we would like to express our deep sense of gratitude and profound thanks to our guide **Prof. A.J ARUN JEYA PRAKASH,** Department of Aerospace Engineering, UPES Dehradun for his valuable advice, suggestions, insurmountable guidance which played a vital role in carrying out our project work successfully. Our thanks to **Prof. (Dr.) Om Prakash,** Head, Department of Aerospace Engineering, UPES, Dehradun for his valuable guidance towards our project.

We would like to thank **D.R.F Ms. Vindhya Devalla,** for her help & encouragement during the work. We also thank all faculty, staff, and students of Department of Aerospace Engineering and for rendering help during various stages of the project work. We would like to thanks our parents for their support and encouragement for carrying out our project work in a well sincere manner.

Arjun Pratap Singh [R640209009]
Rajat Saxena [R640209031]
Sagar Verma [R640209052]

ABSTRACT

This project is about developing a control system to maintain the cabin pressure and temperature of an aircraft for the passengers and crew comfort. An Aircraft cabin model is made in such a way that it is completely sealed and air locked. Pressure sensors and temperature sensors are placed inside the sealed cabin and the values are digitally converted using an A/D converter and fed as input to the micro controller system. The control law is programmed in such a way that it accepts the current value of the pressure and temperature with respect to time and compares it with the pre defined threshold, which is normal room temperature and pressure which makes the passengers feel comfort.

Cabin Temperature and Pressure are the very important components of the aircraft. The pressurization system is very essential, when the aircraft is going to fly over 3000m above the sea level to protect the passengers and crew. Temperature is also a very important factor as at that altitude increases the outside temperature becomes very less and has to be balanced with the normal environmental conditions for the passengers and crew to survive. The general problems faced are, hypoxia, altitude sickness, decompression sickness, barotrauma etc. to avoid such problems a well-established technique is being developed in order to manage the pressure and temperature inside the cabin. In this project, compressed air is pumped into the cabin to provide high pressure and temperature is also varied by employing a filament lamp setup. Depending upon the pressure and Temperature variation the control law which is programmed in the embedded system takes the necessary action to maintain the desired pressure and temperature inside the cabin.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction to Aircraft Cabin Temperature and Pressure management System

Millions of people travel in airplanes all around the world everyday and most of them probably take for granted the advanced system that is being used to maintain their comfort during the flight. If it weren't for the airplane cabin temperature management system during flight then it would not be possible for people to survive flying at the high altitudes that they do in modern aircraft.

The first flights that were ever attempted were carried out at very low altitudes, sometimes hardly even clearing the ground so cabin temperature was not an issue for the pilot (and co-pilots). As people learnt more about the physics of flight and technology advanced, planes were able to travel further and reach greater heights between take-off and landing. The planes also developed in size, ascending from the one and two seat models that were purely for transporting the pilot and co-pilot up to larger planes that could hold a number of people. In the early years following the Second World War, the experiments and trials of the 1920s and 30s came to fruition in the form of pressurized cabins in regular airline service. On July 8 1940 the first flight of the Boeing Stratoliner took place. This was the first airliner with a temperature maintained cabin and was able to fly at an altitude of 20,000 feet, avoiding the turbulence that is associated with flying at lower altitudes. At the time, this development allowed piston-engine aircraft to fly at altitudes above most of the weather. With the advent of the turbojet engine, the fuel consumption of which precludes economic flight at lower levels, of transport aircraft became universal. Eventually technology advanced to the stage it's at today where we have large commercial airlines that are capable of carrying hundreds of people at once over long distances, traveling at high speeds and high altitudes.

When we consider a large commercial plane it is necessary to understand the changing environment that the plane experiences when it travels from its current position to its destination. It begins on a runway at sea level, or thereabouts, and ascends up to a cruising height for majority of the flight. Prior to landing the airplane must slowly descend back to sea level and then make a safe landing at its destination. When the plane is cruising it usually travels at an altitude of about 40,000 feet, sometimes higher, sometimes lower, depending on the atmospheric conditions. It is not possible for a human to survive at these altitudes due to the air pressure and the lack of breathable oxygen. For this reason it is necessary for a plane to have a system which can control the pressure within the aircraft and create a comfortable environment for passengers and crew.

The ultimate system would be able to maintain the pressure inside the cabin similar to that at sea level as this is where the human body is most comfortable. As this is not possible due to the physics of the system, a best case scenario must be developed and it must be monitored and controlled so as to coincide with the changing conditions outside of the airplane. It is also important to note that the maximum altitude that can be safely and comfortably sustained without respiratory support is 10,000 feet. As the plane ascends and descends during its flight the pressure outside of the plane varies dramatically so the pressure control system must be able to constantly adjust to these changing conditions and try to maintain the cabin environment as constant as possible.

Along with the pressurization of the cabin it is also necessary to consider the quality of the air that the passengers are experiencing. While the air inside the cabin must be maintained at pressure that is acceptable for a human it must also be kept at a comfortable temperature. Just like the air conditioning system in your house or car, an airplane has an air conditioning system which governs the temperature of the air inside the cabin. The air must be maintained within a certain temperature range to ensure the comfort of the passengers and crew, especially the crew as they cannot perform at their best in an environment that is either too hot or too cold.

These are most of the aspects that the environmental control system of large commercial airplane should cover; if there are any more then they will be discussed in the report if and when they arise. Even though this report will be looking into the pressurization and cabin air control of a large commercial airplane; the physics of the system, and most likely many of the components that are used, can be related back to smaller planes which all experience the differing air pressure associated with flying at altitude.

1.1.1 Cabin Temperature Management System

It is the refrigeration unit of the Environmental Control System (ECS) used in pressurized gas turbine-powered aircraft. Normally an aircraft has two or three of Air Cycle Machines. Each ACM and its components are often referred as an air conditioning pack. No condensation or evaporation of a refrigerant is involved, and the cooled air output from the process is used directly for cabin ventilation or for cooling electronic equipment.

History of Cabin Temperature Management System

Air cycle machines were first developed in the 19th century for providing chilling on ships. The technique is a reverse Brayton cycle.

Fundamental Elements of the System

Bleed air from the engines, an auxiliary power unit, or a ground source, which can be in excess of 150 °C is directed into a primary heat exchanger. Outside air at ambient temperature and pressure is used as the coolant in this air-to-air heat exchanger. Once the hot air has been cooled, it is then compressed by the centrifugal compressor. This compression heats the air (the maximum air temperature at this point is about 250 °C) and it is sent to the secondary heat exchanger, which again uses outside air as the coolant. The pre-cooling through the first heat exchanger increases the efficiency of the ACM because it lowers the temperature of the air entering the compressor, so that less work is required to compress a given air mass (the energy required to compress a gas by a given ratio rises as the temperature of the incoming gas rises).At this point, the temperature of the compressed cooled air is somewhat greater than the ambient temperature of the outside air. The compressed, cooled air then travels through the expansion turbine which extracts work from the air as it expands, cooling it to below ambient temperature (down to −20 °C or −30 °C). After the air has been cooled down, water vapor in the air condenses, forming fog or high humidity. To get rid of this, the air exiting the expansion turbine is passed through a water separator that absorbs the moisture. The air can now be combined in a mixing chamber with a small amount of uncooled engine bleed air. This warms the air to a desired temperature, and then the air is vented into the cabin or to electronic equipment.

1.1.2 Cabin Pressure Management System

Cabin pressurization is used to create a safe and comfortable environment for aircraft passengers and crew flying at high altitude by pumping conditioned air into the cabin. This air is usually bled off from the engines at the compressor stage. The air is then cooled, humidified, mixed with recirculated air if necessary and distributed to the cabin by one or more environmental control systems. The cabin pressure is regulated by the outflow valve.

Need for Pressurization

Pressurization becomes necessary at altitudes beyond 12,500 feet (3,800 m) to 14,000 feet (4,300 m) above sea level to protect crew and passengers from the risk of a number of physiological problems caused by the low outside air pressure above that altitude; it also serves to generally increase passenger comfort. The principal physiological problems are as follows:

HYPOXIA: The low partial pressure of oxygen at an altitude reduces the alveolar oxygen tension in the lungs and subsequently in the brain, leading to sluggish thinking, dimmed vision, loss of consciousness, and ultimately death.

BARO TRAUMA: As the aircraft climbs or descends, passengers may experience discomfort or acute pain as gases trapped within their bodies expand or contract. ...

The Mechanics of Pressurization

Pressurization is achieved by the design of an airtight fuselage engineered to be pressurized with a source of compressed air and controlled by an environmental control system (ECS). The most common source of compressed air for pressurization is bleed air extracted from the compressor stage of a gas turbine engine. The part of the bleed air that is directed to the ECS, is then expanded and cooled to a suitable temperature by passing it through a heat exchanger and air cycle machine . Hot trim air can be added downstream of air conditioned air coming if it is needed to warm a section of the cabin that is colder than others.

Advantages of Aircraft Environmental System

Advantages of air cycle systems are the refrigerant (air) is free, the compressor is already part of the engine and APU though there is an SFC impact, air is directly used for cooling or heating therefore no evaporator required, efficient heat transfer, minor leakage is not a problem and mechanically simple system. Therefore, lighter system weight, safer and more reliable.

Chapter 2

LITERATURE REVIEW

2.1 Cabin pressurization system:

Cabin altitude is the distance above sea level at which the atmosphere exerts the same pressure as that experienced in the cabin. The maximum cabin altitude under normal flight operation is set at 8000ft. This is equivalent to a cabin pressure of 75kPa (10.85psi) as compared to the 101kPa (14.65psi) experienced at ground level. Once the cabin altitude climbs above 10000ft an alarm will sound in the cockpit, should cabin altitude continue to climb above 14000ft the planes' oxygen masks will deploy as breathing will become difficult.

The cabin altitude is one of the most important aspects of the environmental control system of a plane regarding the pressurization of the cabin. Figure 1 displays the relationship between the cabin altitude and the actual altitude of the aircraft.

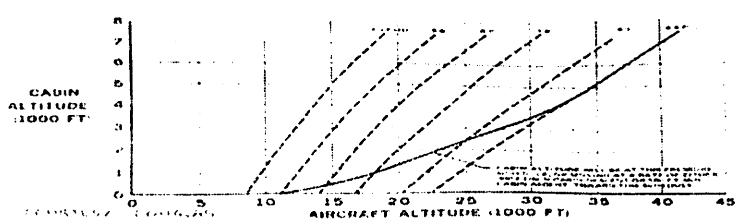


Figure 2.1 Airplane Cabin Altitude Schedule

As expected the pressure inside the cabin of the airplane at sea level is equal to the normal air pressure experienced at sea level. It is not necessary for the cabin of an airplane to be pressurized until it travels higher than about 10,000 feet. Below this altitude it is possible for humans to withstand the changes in the air, but once the plane starts to gain more altitude it is necessary for the plane to be pressurized. As the altitude of the airplane increases so does the cabin altitude, but at a much slower rate due to the operation of the pressurization system. The rate of increase is almost linear, although once the airplane reaches an altitude of approximately 32,000 feet the cabin altitude begins to increase at a greater rate.

STANDARD ATMOSPHERIC PRESSURE			
Altitude (ft)	Pressure (p.s.i.)	Altitude (ft)	Pressure (p.s.i.)
Sea Level	14.7	16,000	8.0
2,000	13.7	18,000	7.3
4,000	12.7	20,000	6.8
6,000	11.8	22,000	6.2
8,000	10.9	24,000	5.7
10,000	10.1	26,000	5.2
12,000	9.4	28,000	4.8
14,000	8.6	30,000	4.4

The altitude where the standard air pressure is equal to 10.9 p.s.i can be found at 8,000 feet.

At an altitude of 28,000 feet, standard atmospheric pressure is 4.8 p.s.i. By adding this pressure to the cabin pressure differential of 6.1 p.s.i.d., a total air pressure of 10.9 p.s.i. is obtained.

Table 2.1 Variation of Pressure with Altitude

For the standard cruising height of about 40,000 feet the altitude inside the aircraft is around 8,000 feet providing the pressurization system is operating properly. If there was a problem with the system then the pressure would not behave according to the above schedule and may even be quite unpredictable depending on the nature of the problem. For the comfort of passengers cabin pressure must not be allowed change at too rapid a rate. Rapid changes in pressure cause pockets of gas in the body to expand and contract. As a result passengers can experience pain in the ears, teeth and head. To prevent this, the rate of climb of cabin pressure must be restricted to 5m/s (1000ft/min) and the rate of descent restricted to 2.3m/s (450ft/min). One very important point to note in relation to the pressurization of the cabin is the need to keep the pressure difference at a minimum. While the health and safety of passengers is one reason why the airplane cabin must be pressurized, it is also necessary to maintain the structure of the plane. If the pressure inside the cabin of the plane was very low compared to pressure outside of the airplane then a large pressure difference would be present. If this scenario were to occur then the airplane structure would collapse so there are pressure devices in place to prevent this taking place.

2.2 Cabin Temperature:

It is possible to see the importance of controlling the temperature within an aircraft cabin when considering the requirements of the passengers and crew. The most important requirements shown on the plot are oxygen and drinking water, food is present but it doesn't present as being dependent on temperature. Oxygen and drinking water, however, are dependent on the temperature and are very necessary for the survival of people on the airplane.

As would be expected the amount of drinking water consumed increases as the temperature increases. The interesting point to note is that the amount consumed doesn't begin to increase until a certain temperature, but once it does it increases quite dramatically. The temperature, at which this occurs, approximately 60°F (15.5oC), is an important value for the environmental control system. Once the temperature inside the cabin of the aircraft moves above this value the demand for drinking water increases. If the temperature inside the cabin continues to climb then the demand for drinking water, when considering a capacity filled airplane, is probably going to far outweigh supply. The result of this scenario could be catastrophic so a competent air conditioning system is a necessary aspect of an environmental control system. The demand for oxygen depending on temperature is slightly different to that of drinking water. While the consumption of oxygen increases as the temperature increases above 80°F, it also increases as the temperature decreases below 80°F. In fact, the maximum oxygen demand occurs around 35°F leading to the fact that the optimum temperature is somewhere between 60° to 70°F. Considering this it is necessary for the environmental control system to be able to effectively regulate the temperature of the air inside the cabin so the demand for water and oxygen doesn't overcome the available supplies. Continuing to look at the temperature associated with the cabin during flight, Figure 2 shows us the temperature of the air that is supplied to the bleed system during flight.

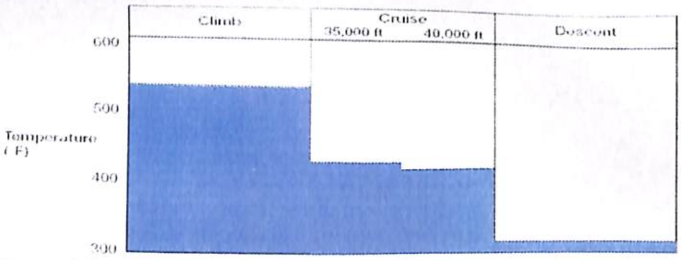


Figure 2.2 Average air temperature supplied to the Bleed System during flight

The temperature is at its greatest when the airplane is climbing because this is when the engines are doing the most amount of work. More fuel is being burnt and a greater amount of air is passing through the engines compared to any other stage of flight. The extremely high temperatures that are being produced in the engine at this stage of flight create very hot air that is supplied to the bleed system. As a result the air supplied to bleed system would have to be cooled further by the planes air conditioning system to provide passenger comfort. Understandably the temperature is slightly lower during the cruise stage of the flight as the engines are still working to keep the plane in the air but they are not required to help the plane gain altitude. As a result we get air supplied to the bleed system with a slightly lower temperature. Thirdly, during the descent stage of the flight the air is at its lowest temperature namely because the engines are doing the least amount of work in this stage of flight. During the descent stage the pressure outside the plane increases, and as such the rate of increase of pressure inside the cabin must be controlled.

CHAPTER 3

OBJECTIVE AND SCOPE

This project is about developing a control system to maintain the cabin pressure and temperature of an aircraft for the passengers and crew comfort. An Aircraft cabin model is made in such a way that it is completely sealed and air locked. Pressure sensors and temperature sensors are placed inside the sealed cabin and the values are digitally converted using an A/D converter and fed as input to the microcontroller system. The control law is programmed in such a way that it accepts the current value of the pressure and temperature with respect to time and compares it with the pre defined threshold, which is normal room temperature and pressure which makes the passengers feel comfort

CHAPTER 4

Methodology for Temperature Control

4.1 INTRODUCTION

A temperature control system relies upon a controller, which is connected to a temperature sensor. It compares the actual temperature to the desired control temperature, or set point, and provides an output to a control element. Mostly the control element is a heater or a cooler/fan.

ADC is used to convert the output voltage of the temperature sensor into digital value which is connected to a microcontroller. The microcontroller then compares the current temperature with the set temperature & if the current temperature is lesser than set temperature the relay is turned ON, which can be used to turn ON a Heater. The Set point can be changed with the help of two switches.

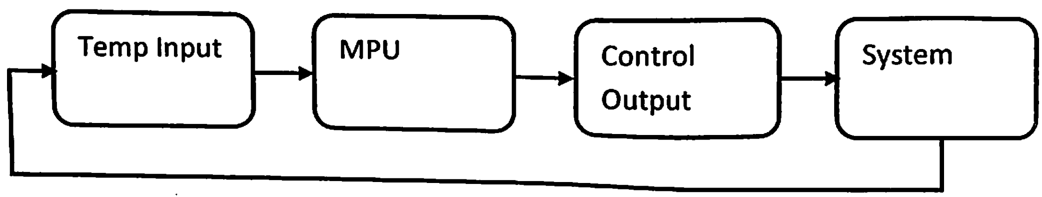


Fig 4.1 Temperature Controller Schematic Diagram

4.2 TYPES OF CONTROLLERS

There are three basic types of controllers:

1. On-off
2. Proportional
3. PID.

Depending upon the system to be controlled, the operator will be able to use one type or another to control the process.

4.2.1 ON/OFF CONTROL:

An on-off controller is the simplest form of temperature control device. The output from the device is either on or off, with no middle state. An on-off controller will switch the output only when the temperature crosses the set point.

For heating control, the output is on when the temperature is below the set point, and off above set point.

Since the temperature crosses the set point to change the output state, the process temperature will be cycling continually, going from below set point to above, and back below. In cases where this cycling occurs rapidly, and to prevent damage to contactors and valves, an on-off differential, or "hysteresis," is added to the controller operations. This differential requires that the temperature exceed set point by a certain amount before the output will turn off or on again. On-off differential prevents the output from "chattering" or making fast, continual switches if the cycling above and below the set point occurs very rapidly.

On-off control is usually used where a precise control is not necessary, in systems which cannot handle having the energy turned on and off frequently, where the mass of the system is so great that temperatures change extremely slowly, or for a temperature alarm. One special type of on-off control used for alarm is a limit controller.

4.2.2 PROPORTIONAL CONTROL:

Proportional controls are designed to eliminate the cycling associated with on-off control. A proportional controller decreases the average power supplied to the heater as the temperature approaches set point. This has the effect of slowing down the heater so that it will not overshoot the set point, but will approach the set point and maintain a stable temperature.

This proportioning action can be accomplished by turning the output on and off for short time intervals. This "time proportioning" varies the ratio of "on" time to "off" time to control the temperature. The proportioning action occurs within a "proportional band" around the set point temperature. Outside this band, the controller functions as an on-off unit, with the output either fully on (below the band) or fully off (above the band). However, within the band, the output is turned on and off in the ratio of the measurement difference from the set point. At the set point (the midpoint of the proportional band), the output on:off ratio is 1:1; that is, the on-time and off-time are equal. if the temperature is further from the set point, the on- and off-times vary in proportion to the temperature difference. If the temperature is below set point, the output will be on longer; if the temperature is too high, the output will be off longer.

4.2.3 PID CONTROL:

The third controller type provides proportional with integral and derivative control, or PID. This controller combines proportional control with two additional adjustments, which helps the unit automatically compensate for changes in the system. These adjustments, integral and derivative, are expressed in time-based units; they are also referred to by their reciprocals, RESET and RATE, respectively.

The proportional, integral and derivative terms must be individually adjusted or "tuned" to a particular system using trial and error. It provides the most accurate and stable control of the three controller types, and is best used in systems which have a relatively small mass, those which react quickly to changes in the energy added to the process.

It is recommended in systems where the load changes often and the controller is expected to compensate automatically due to frequent changes in set point, the amount of energy available, or the mass to be controlled.. These are also known as auto tune controllers.

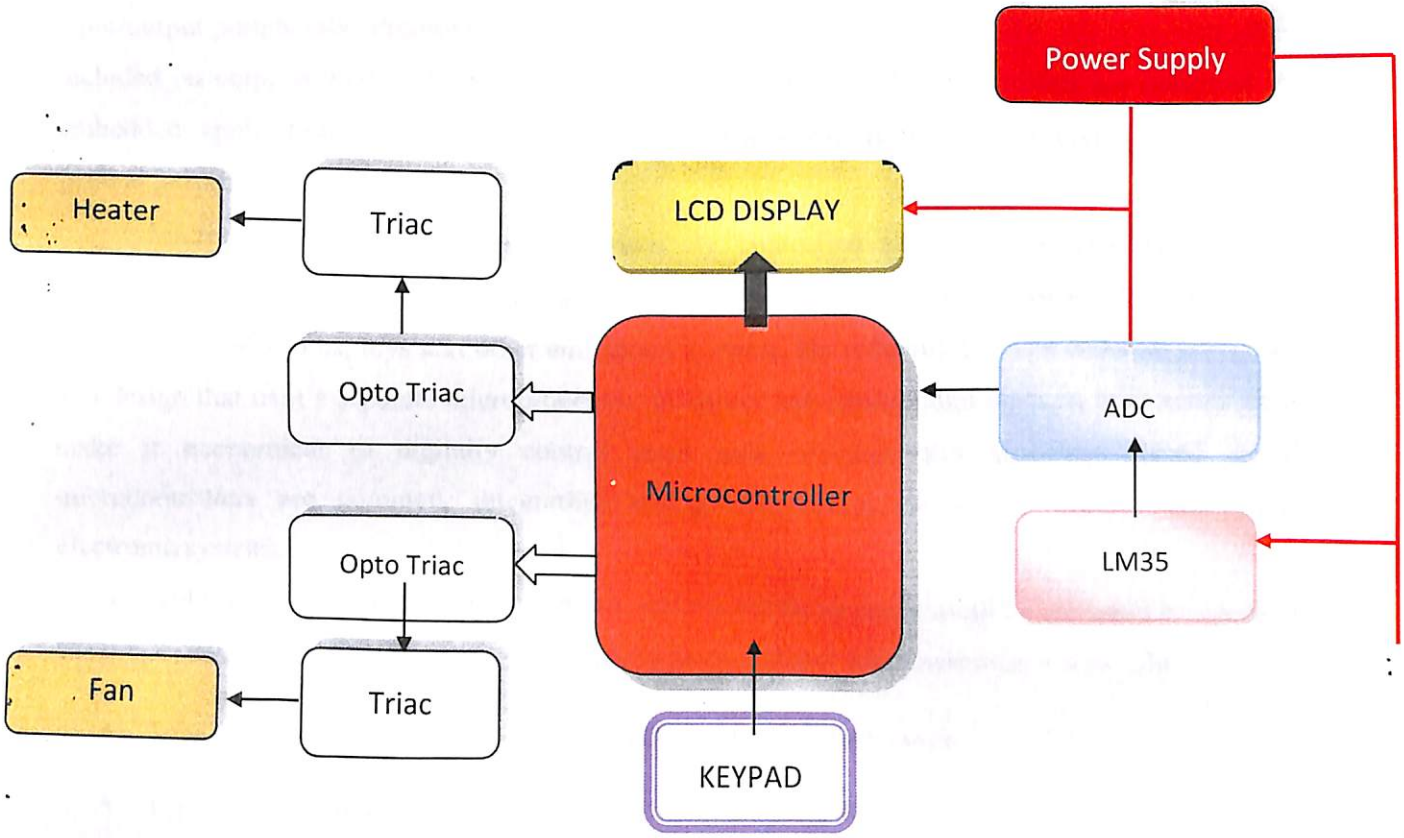


Fig 4.2 Block Diagram

CHAPTER 5
COMPONENTS USED

5.1 MICROCONTROLLER:

A microcontroller (sometimes abbreviated μC, uC or MCU) is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.

The controller is the heart of entire system, and the whole system should be analyzed in selecting the proper controller. The following items should be considered when selecting a controller:

- Type of input sensor (thermocouple, RTD) and temperature range
- Type of output required (electromechanical relay, SSR, analog output)
- Control algorithm needed (on/off, proportional, PID)
- Number and type of outputs (heat, cool, alarm, limit)

5.1.1 Description:

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 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 in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 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 con-tents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

5.1.2 Pin Configuration:

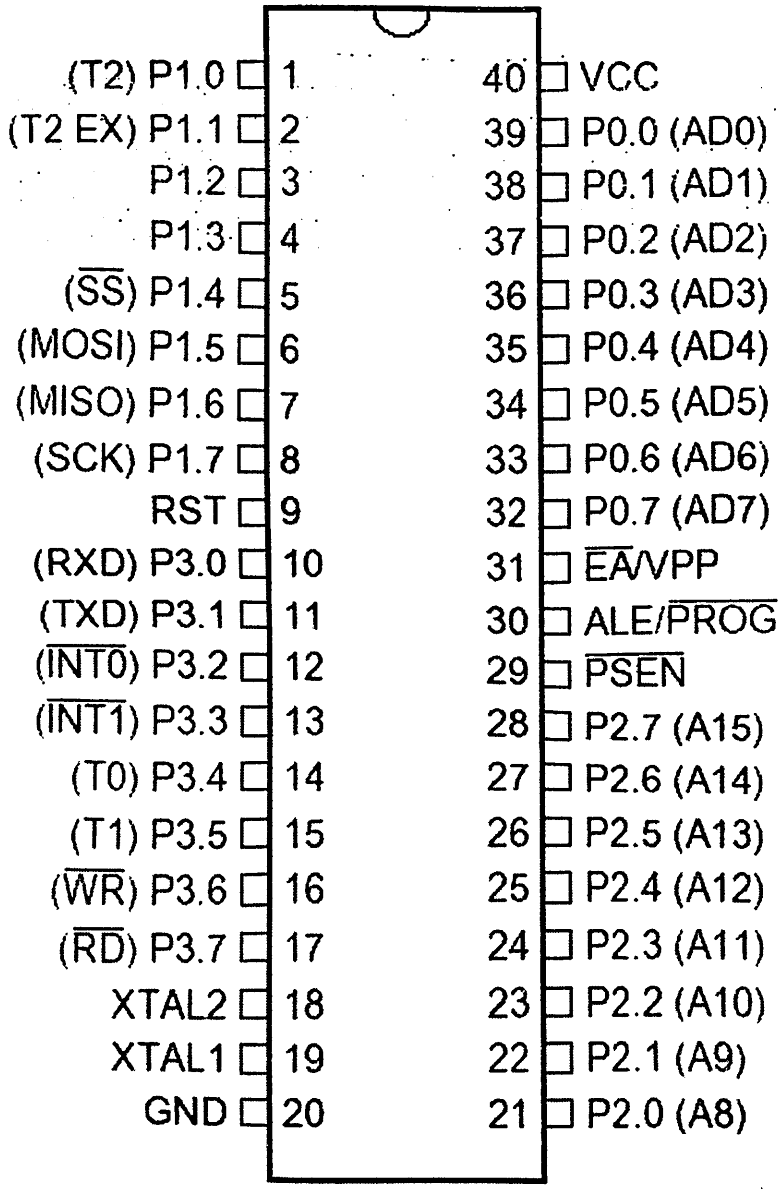


Fig. 5.1.2 Pin Configuration of AT89S52

5.1.3 Pin Description:

1) **VCC:** Supply voltage.

2) **GND:** Ground.

3) **Port 0:** Port 0 is an 8-bit open drain bidirectional 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 can 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 pull-ups. Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pull-ups are required during program verification.

4) **Port 1:** Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. 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 pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table. Port 1 also receives the low-order address bytes during Flash programming and verification.

Port Pin	**Alternate Functions**
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

Table no. 5.1 Port 1 Pin Functions

5) **Port 2:** Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. 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 pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. 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, Port 2 uses strong internal pull-ups when emitting 1s. During accesses to external data memory that uses 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.

6) **Port 3:** Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. 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 inter-nal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL) because of the pull-ups. Port 3 receives some control signals for Flash programming and verification. Port 3 also serves the functions of various special features of the AT89S52, as shown in the following table.

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

Table no. 5.2 Port 3 Pin Functions

7) RST: Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives high for 98 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

8) ALE/PROG: Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (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.

9) PSEN: Program Store Enable (PSEN) is the read strobe to external program memory. When the AT89S52 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 exter-nal data memory.

10) 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 VCC for internal program executions. This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming.

11) XTAL1: Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

12) XTAL2: Output from the inverting oscillator amplifier.

5.2 KEYPAD:

The keypad consists of 2 keys i.e. Increment & Decrement. These TACTILE SWITCHES are used to Increase & Decrease the temperature set point correspondingly.

5.3 ADC:

The output of the temperature sensor is an analog signal & cannot be directly given to the microcontroller. We need to convert this analog signal into digital so we will be using MCP3201 which is a successive approximation 12-bit ADC.

5.4 Temperature Sensor:

We are using LM 35 as temperature sensor. LM 35 is a precision temperature sensor whose output is linearly proportional to Celsius Temperature. The LM35 is rated to operate from -55° Centigrade to 150° Centigrade.

5.5 LCD:

The 16 X 2 LCD Display provides a user interface. A liquid crystal display (commonly abbreviated LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. It is prized by engineers because it uses very small amounts of electric power, and is therefore suitable for use in battery-powered electronic devices.

A very popular standard exists which allows us to communicate with the vast majority of LCDs regardless of their manufacturer. The standard is referred to as HD44780U, which refers to the controller chip which receives data from an external source (in this case, the 8051) and communicates directly with the LCD.

5.6 OPTO TRIAC:

Triacs work on 230V and our controller works on 5v. Moreover the controller cannot provide the necessary gate current and voltage to trigger a triac hence an Opto triac is used. So we can say that opto trac provides the necessary gate current and voltage required for the Triac and also provides Isolation between the Triac and the controller.

5.7 TRIAC:

We are using Triac as a switch and it provides the necessary gate current and voltage required for the Heater/Fan. TRIACs are bidirectional and so current can flow through them in either direction.

5.8 POWER SUPPLY:

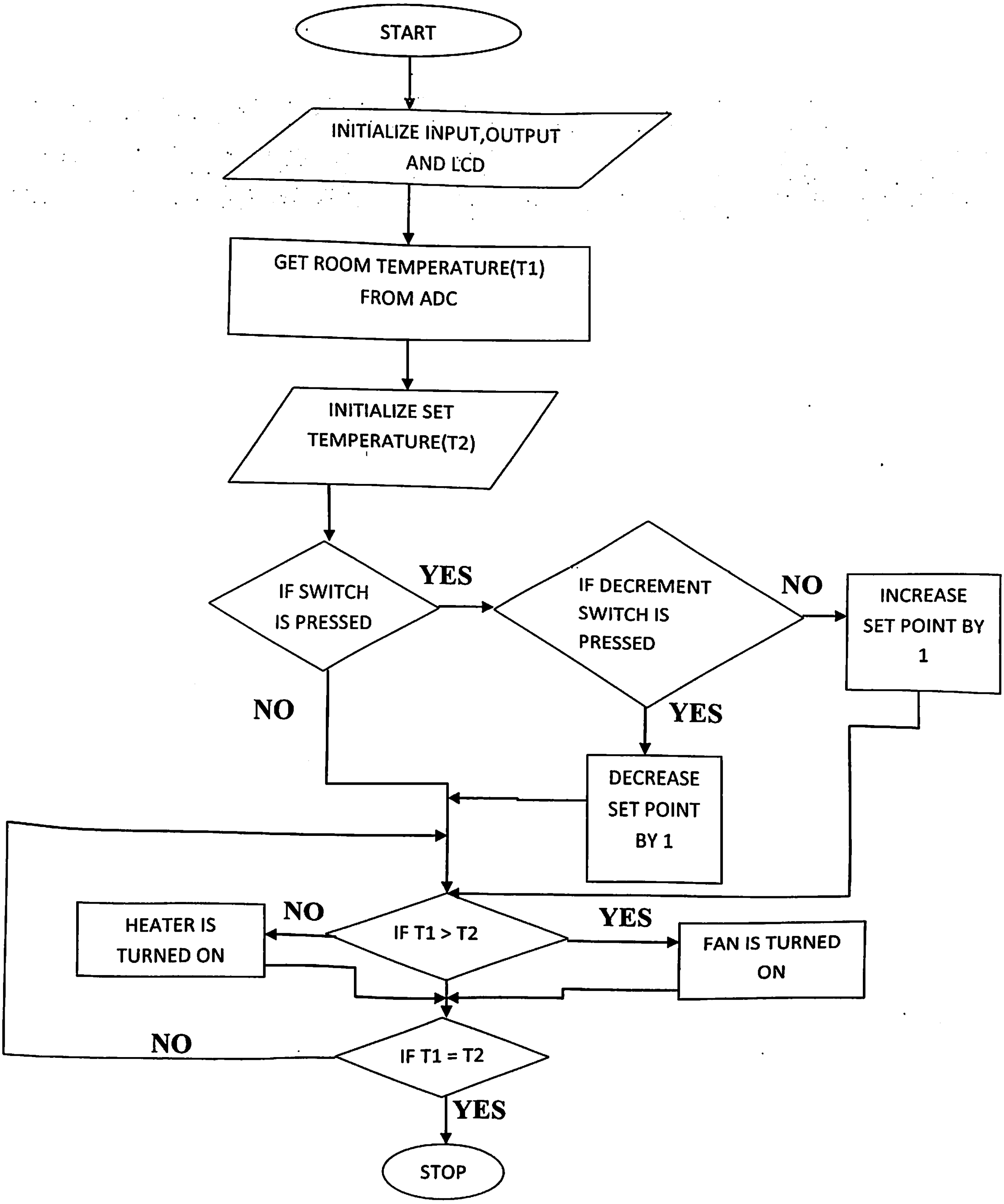
We require regulated 5 volts for microcontroller, ADC, Temperature Sensor and LCD. These voltages are generated from 230v line voltage.

3.9 HEATER/FAN:

Any resistive/inductive element like bulb, coil and furnace can be used as a heating unit to increase the room temperature. Any cooling device like fan, cooler and AC can be used as a cooling device.

CHAPTER 6

FLOW CHART



CHAPTER 7

ALGORITHM USED

1) start

2) Initialize all Input output Pins.

3) Initialize LCD

4) check if any switch is pressed if yes go to 12.

5) check if data is received from IR sensor if yes go to 16.

6) get temperature reading from ADC.

7) Display reading on lcd.

8) compare temperature with the setpoint.

9) if current temperature > Set point then turn on the fan and turn off the heater.

10) if current temperature < Set point then turn off the fan and turn on the heater.

11) if current temperature = Set point then turn off the fan and the heater.

12) go to 4

13) if switch pressed is increment switch then increment setpoint by 1

14) if switch pressed is decrement switch then decrement setpoint by 1

15) go to 4

16) if inc switch of remote is pressed then increment set point by 1

17) if dec switch of remote is pressed the decrement set point by 1

18) go to 4

19) stop

CHAPTER 8

POWER SUPPLY AND MICROCONTROLLER PIN DESCRIPTION

8.1 POWER SUPPLY

1) The circuit uses standard power supply comprising of a step-down transformer from 230Vto 12V and 4 diodes forming a bridge rectifier that delivers pulsating dc which is then filtered by an electrolytic capacitor of about 4700µF two numbers.
2) The filtered dc being unregulated, IC LM7805 is used to get 5V DC constant at its pin no 3 irrespective of input DC varying from 7V to 15V.
3) The input dc shall be varying in the event of input ac at 230volts section varies from 160V to 270V in the ratio of the transformer primary voltage V1 to secondary voltage V2 governed by the formula V1/V2=N1/N2. As N1/N2 i.e. no. of turns in the primary to the no. of turns in the secondary remains unchanged V2 is directly proportional to V1.
4) Thus if the transformer delivers 12V at 220V input it will give 8.72V at 160V.Similarly at 270V it will give 14.72V.
5) Thus the dc voltage at the input of the regulator changes from about 8V to 15V because of A.C voltage variation from 160V to 270V the regulator output will remain constant at 5V.
6) The regulated 5V DC is further filtered by a small electrolytic capacitor of 10µF for any noise so generated by the circuit.
7) One LED is connected of this 5V point in series with a current limiting resistor of 330Ω to the ground i.e., negative voltage to indicate 5V power supply availability.
8) The unregulated 12V point is used for other applications as and when required.

8.2 STANDARD CONNECTIONS TO 8051 SERIES MICROCONTROLLER

1) ATMEL series of 8051 family of micro controllers need certain standard connections.
2) The actual number of the Microcontroller could be "89C51" , "89C52", "89S51", "89S52", and as regards to 20 pin configuration a number of "89C2051".
3) The 4 set of I/O ports are used based on the project requirement.

4) Every microcontroller requires a timing reference for its internal program execution therefore an oscillator needs to be functional with a desired frequency to obtain the timing reference as t =1/f.
5) A crystal ranging from 2 to 20 MHz is required to be used at its pin number 18 and 19 for the internal oscillator.
6) It may be noted here the crystal is not to be understood as crystal oscillator It is just a crystal, while connected to the appropriate pin of the microcontroller it results in oscillator function inside the microcontroller. Typically 11.0592 MHz crystal is used in general for most of the circuits using 8051 series microcontroller.
7) Two small value ceramic capacitors of 33pF each is used as a standard connection for the crystal as shown in the circuit diagram.

8.3 RESET

1) Pin no 9 is provided with an reset arrangement by a combination of an electrolytic capacitor and a register forming RC time constant.
2) At the time of switch on, the capacitor gets charged, and it behaves as a full short circuit from the positive to the pin number 9.
3) After the capacitor gets fully charged the current stops flowing and pin number 9 goes low which is pulled down by a 10k resistor to the ground.
4) This arrangement of reset at pin 9 going high initially and then to logic 0 i.e., low helps the program execution to start from the beginning.
5) In absence of this the program execution could have taken place arbitrarily anywhere from the program cycle.
6) A pushbutton switch is connected across the capacitor so that at any given time as desired it can be pressed such that it discharges the capacitor and while released the capacitor starts charging again and then pin number 9 goes to high and then back to low, to enable the program execution from the beginning.
7) This operation of high to low of the reset pin takes place in fraction of a second as decided by the time constant R and C.

For example: A 10µF capacitor and a 10kΩ resistor would render a 100ms time to pin number 9 from logic high to low, there after the pin number 9 remains low.

8.4 EXTERNAL ACCESS (EA):

1) Pin no 31 of 40 pin 8051 microcontroller termed as EA‾ is required to be connected to 5V for accessing the program form the on-chip program memory.
2) If it is connected to ground then the controller accesses the program from external memory. However as we are using the internal memory it is always connected to +5V.

CHAPTER 9
PIN DIAGRAM AND COMPONENTS

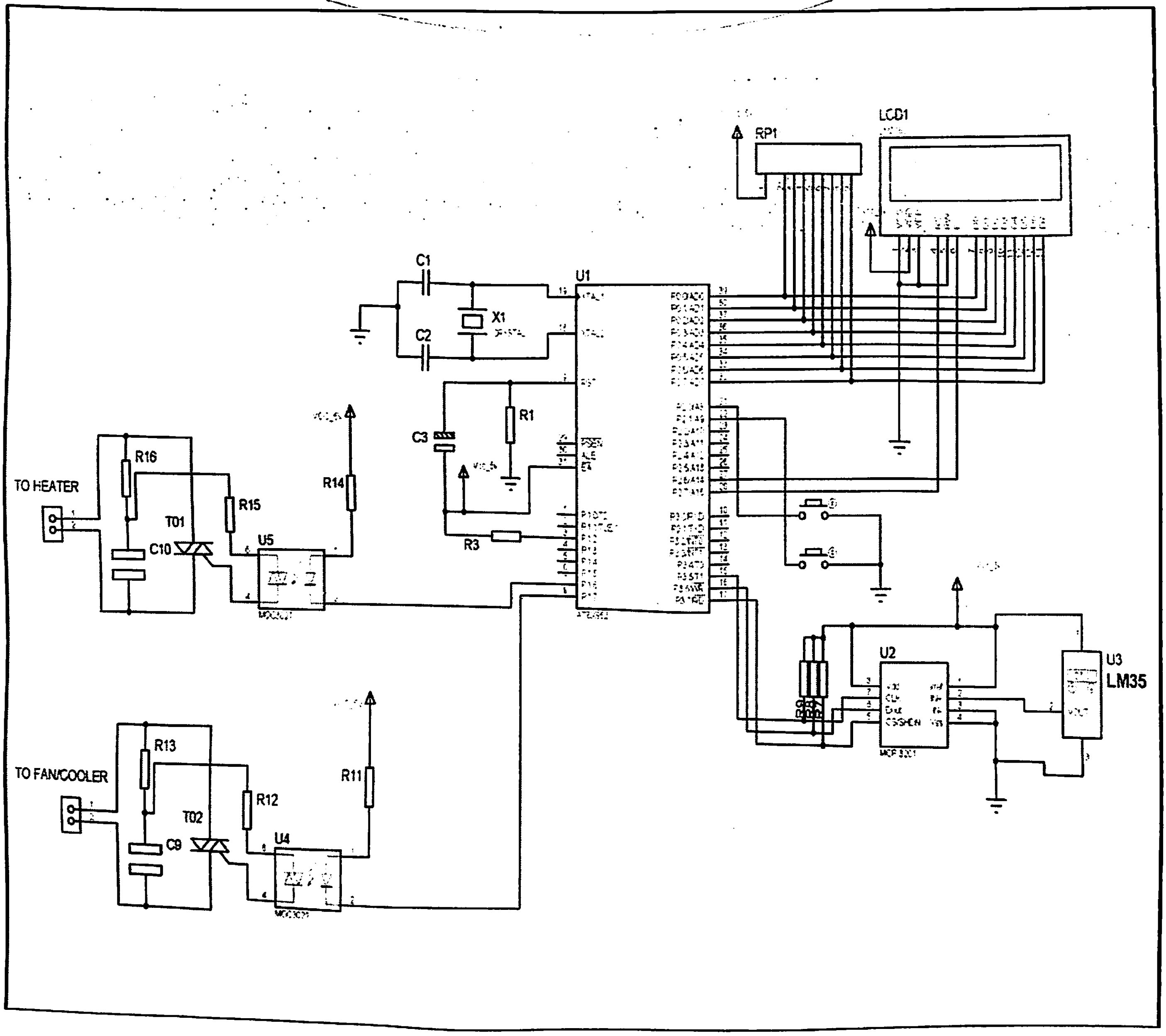


Fig 9.1 Pin Diagram

Components

Resistors,"R1": 100k

Resistors,"R12" : 1K

Resistors,"R13" :330ohm

Resistors,"R15" : 1k

Resistors,"R16" : 330ohm

Resistors,"R7" : 2.2k

Resistors,"R8" : 2.2k

Resistors,"R9" : 2.2k

Resistors,"R11" : 1k

Resistors,"R14" : 1k

Capacitors,"C1" : 33pf

Capacitors,"C2" : 33pf

Capacitors,"C3" : 10uf

Capacitors,"C9", "C10" : 0.1uf/230v

Integrated Circuits,"U1",AT89S52,

Integrated Circuits,"U2",MCP 3201: ADC

Integrated Circuits,"U3",LM35,

Integrated Circuits,"U4",MOC3021: Opto Triac

Integrated Circuits,"U5",MOC3021: Opto Triac

Miscellaneous,"LCD1",LM016L,: 16x2 Alphanumeric display

Miscellaneous,"RP1",,: 10k resistor bank

Miscellaneous,"T01",, : BTA12: Triac

Miscellaneous,"T02",,: BTA12: Triac

Miscellaneous,"TO FAN/COOLER",,

Miscellaneous,"TO HEATER",,

Miscellaneous,"X1",CRYSTAL,: 11.0592Mhz

CHAPTER 10

RESULTS AND DISCUSSIONS

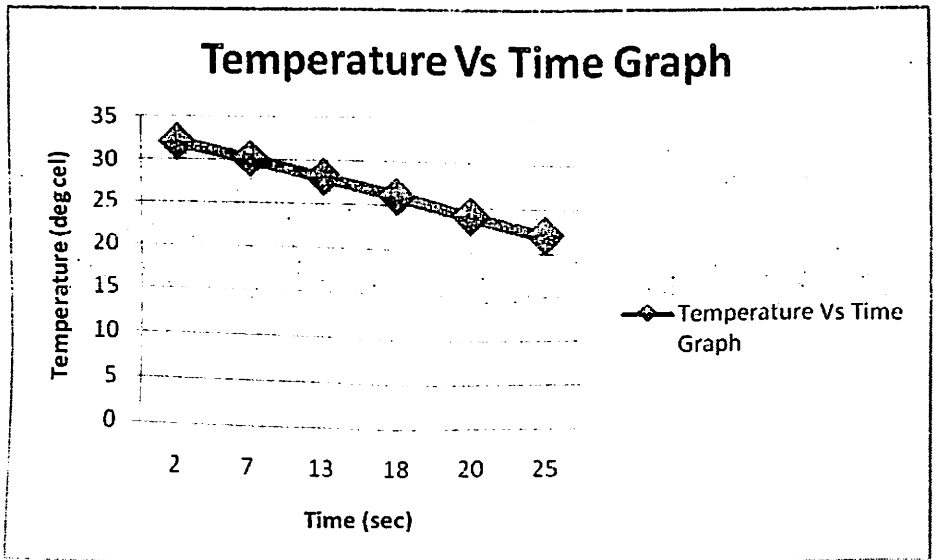


Fig 10.1 Temperature Vs Time Graph

When Set temperature = 33^0 C

Room Temperature = 22^0 C

On the X axis is temperature the Temperature Sensor sensed while it came down from set temperature to Room Temperature.

On the Y axis is the time it took to cool down to the temeprature on the X axis to the set temperature.

The setup was put in a room with normal temeprature conditions and was left untouched and the reading were taken using a stopwatch.

Temperture (^{0} C)	Time (Sec)
32	2
30	7
28	13
26	18
24	20
22	25

Table no. 10.1 Temperature and Time Table

CONCLUSION

The project "**CABIN TEMPERATURE AND PRESSURE MANAGEMENT SYSTEM**" has been successfully designed and tested. Integrating features of all the hardware components used have developed it Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit Secondly, using highly advanced IC's and with the growing technology the project has been successfully implemented.

REFERENCES

- Edwards, M and E, 1990, The Aircraft Cabin: Managing the Human Factors, Gower Technical, England
- Hunt, Space, 1995, The Airplane Cabin Environment: Issues Pertaining to Flight Attendant Comfort, Boeing Company
- Marini RA. Breathing easy: are proposed new standards for air quality in passenger cabins too low? Frequent Flyer. March 2000:20-25
- Air quality, ventilation, temperature, and humidity in aircraft. O'Donnell A, Donnini G, Nguyen VH Am Soc Heating Refrigerating Air-conditioning, Eng J.1991;33:42-46
- Google Images.
- Higgins M, Keates N, Costello D., How safe is airplane air? Wall Street Journal. June 9, 2000:W.
- The Microcontroller handbook, Edward Romero, 1994; 46:74-96.
- AT89S52 Data Sheet, ATMEL.
- Wikipedia.org

APPENDIX

Microcontroller Program

```
adcclk              bit     p3.5
adcop               bit     p3.6
adccs               bit     p3.7

//relay             bit     p1.7
heater              bit     p1.0
cooler              bit     p1.1
sw_inc              bit     P2.0
sw_dec              bit     P2.1

LCD_RS                      BIT             P2.7
LCD_E                       BIT             P2.6
LCD_PORT            EQU             P0
ir                          bit             p2.5

temp_bit            bit             7fh

adcdatamsb          equ     2eh
adcdatalsb          equ     2fh


temp_setpoint0      equ     30h
        //MSB
temp_setpoint1      equ     31h
        //LSB
current_temp0   equ     32h                                 //MSB
current_temp1   equ     33h                                 //LSB
temp_byte           equ     34h
sec_multiplier  equ     35h
delay0              equ     36h
delay1              equ     37h
rc5_system          equ             38h
rc5_command         equ             39h

SEC_MULTIPLIER_DEFAULT      DATA 20

VOLDEC                      data    11h
VOLINC                      data    10h
SYSTEM                      data    00h
```

```
org 0000h
        call lcd_ini                        //initialize the LCD
        call display_welcome                //display welcome on screen
        call delay


        mov temp_setpoint0,#00h
        mov temp_setpoint1,#45h
        mov TMOD,#20h
        call ini_timer0
        mov sec_multiplier,#SEC_MULTIPLIER_DEFAULT
        call update_temperature
        call display_set_temperature

loop:
        call check_switch_pressed
        call timer
        call check_ir
        ajmp loop

delay:
        mov r7,#200
l1_delay:
        mov r6,#217
l2_delay:
        djnz r6,l2_delay
        djnz r7,l1_delay
        ret

adc1:
    mov adcdatamsb,#00h
    mov adcdatalsb,#00h

    clr adccs           ;chip selected

    clr adcclk
    nop
    mov c,adcop
    mov 00h,c
    setb adcclk                  ;null bit
    clr adcclk
    nop
    mov c,adcop
    mov 73h,c

    setb adcclk
```

```
clr adcclk
nop
mov c,adcop
mov 72h,c

setb adcclk
clr adcclk
nop
mov c,adcop
mov 71h,c

setb adcclk
clr adcclk
nop
mov c,adcop
mov 70h,c

setb adcclk

clr adcclk
nop
mov c,adcop
mov 7fh,c

setb adcclk
clr adcclk
nop
mov c,adcop
mov 7eh,c

setb adcclk
clr adcclk
nop
mov c,adcop
mov 7dh,c

setb adcclk
clr adcclk
nop
mov c,adcop
mov 7ch,c

setb adcclk
clr adcclk
nop
mov c,adcop
```

```
    mov 7bh,c

    setb adcclk
     clr adcclk
    nop
    mov c,adcop
    mov 7ah,c

    setb adcclk
    clr adcclk
    nop
    mov c,adcop
    mov 79h,c

    setb adcclk
    clr adcclk
    nop
    mov c,adcop
    mov 78h,c

    setb adcclk
    setb adccs
    ret

;*****************************************************************************
*
;*****************************************************************************
*
;*****************************************************************************
*
check_switch_pressed:
        jnb sw_inc,c1_check_switch_pressed
        jnb sw_dec,c2_check_switch_pressed
        ret

c1_check_switch_pressed:
        call inc_temp_setpoint
        ret

c2_check_switch_pressed:
        call dec_temp_setpoint
        ret

inc_temp_setpoint:
        clr c
        mov a,temp_setpoint1
```

```
        add a,#01h
        da a
        mov temp_setpoint1,a
        jnc r1_inc_temp_setpoint
        clr c
        mov a,temp_setpoint0
        add a,#01h
        da a
        mov temp_setpoint0,a
r1_inc_temp_setpoint:
        call display_set_temperature
        ret

dec_temp_setpoint:
        clr c
        mov a,temp_setpoint1
        add a,#99h
        da a
        mov temp_setpoint1,a
        jc r2_dec_temp_setpoint
        clr c
        mov a,temp_setpoint0
        add a,#01h
        da a
        mov temp_setpoint0,a
r2_dec_temp_setpoint:
        call display_set_temperature
        ret


timer:
        jb tcon.5,c1_timer
        clr c
        ret
c1_timer:
        clr tcon.5
        call ini_timer0
        djnz sec_multiplier,r1_timer
        mov sec_multiplier,#SEC_MULTIPLIER_DEFAULT
        call update_temperature
r1_timer:
        ret
;*****************************************************************
;*****************************************************************
;*****************************************************************
ini_timer0:
```

```
        mov th0,#4ch            ;0.05 sec
        mov tl0,#00H
        setb TR0
        ret


/*
Sends whatever data is in the Acc to the LCD as a Command
*/
lcd_command:
        call lcd_delay              //waits till the LCD is ready
        mov LCD_PORT,a
        clr LCD_RS                        //Data we are sending is a command so RS=0
        setb LCD_E
        clr LCD_E                         //LATCH data onto LCD
        ret

/*
Sends whatever data is in the Acc to the LCD to be displayed
*/
lcd_datadisplay:
        call lcd_delay              //waits till the LCD is ready
        mov LCD_PORT,a
        setb LCD_RS                       //Data we are sending is a command so RS=0
        setb LCD_E
        clr LCD_E                         //LATCH data onto LCD
        ret

lcd_delay:
        mov delay0,#0ffh
l1_lcd_delay:
        djnz delay0,l1_lcd_delay
        ret


/*
This subroutine displays whatever data has been pointed by the data pointer till a null pointer is
detected
*/

lcd_displayline:
        clr a
        movc a,@a+dptr
        jnz cdisplayline1
        ljmp enddisplayline
```

```
cdisplayline1:
        call lcd_datadisplay
        inc dptr
        ljmp lcd_displayline
enddisplayline:
        ret

/*
Initialize the LCD
*/

lcd_ini:
        mov a,#38h
        call lcd_command
        mov a,#06h
        call lcd_command
        mov a,#01h
        call lcd_command
        mov a,#0ch
        call lcd_command
        ret


update_temperature:
        call adc1                       //get data from the adc
        call hextotemperature           //convert it into temperature
        call compare_temperature        //compare the set & current temperature
        call display_current_temp       //display the current temperature
        ret
;**********************************************************
;**********************************************************


compare_temperature:
        clr c
        mov a,current_temp0
        subb a,temp_setpoint0
        jc c2_compare_temperature
        jnz c1_compare_temperature
        clr c
        mov a,current_temp1
        subb a,temp_setpoint1
        jc c2_compare_temperature
        jnz c1_compare_temperature
//setpoint = current so turn OFF relay
```

```
        setb heater
        setb cooler
        ret


c2_compare_temperature:
//setpoint > current so turn ON relay
        setb heater
        clr cooler
        ret


c1_compare_temperature:
//setpoint < current so turn OFF relay
        clr heater
        setb cooler
        ret



/*
compare_temperature:
        clr c
        mov a,current_temp0
        subb a,temp_setpoint0
        jc c2_compare_temperature
        jnz c1_compare_temperature
        clr c
        mov a,current_temp1
        subb a,temp_setpoint1
        jc c2_compare_temperature
        jnz c1_compare_temperature
//setpoint = current so turn OFF relay
        clr relay
        ret


c2_compare_temperature:
//setpoint > current so turn ON relay
        setb relay
        ret
```

```
c1_compare_temperature:
//setpoint < current so turn OFF relay
        clr relay
        ret             */
;*************************************************************
;*************************************************************
/*
4096 steps 5/4096: 1.22mv
8.19 steps = 1 degree centigrade
82 steps = 10 degrees
82b = 52h
*/
hextotemperature:
        mov current_temp0,#00h
        mov current_temp1,#00h

l1_hextotemperature:
        clr c
        mov a,adcdatalsb
        mov temp_byte,a
        subb a,#52h
        mov adcdatalsb,a
        jnc c1_hextotemperature
        mov a,adcdatamsb
        cjne a,#00h,c2_hextotemperature
l2_hextotemperature:
        clr c
        mov a,temp_byte
        subb a,#08h
        mov temp_byte,a
        jnc c3_hextotemperature
        ret

c3_hextotemperature:
        mov a,current_temp1
        add a,#01h
        da a
        mov current_temp1,a
        jnc l2_hextotemperature
        clr c
        mov a,current_temp0
        add a,#01h
        da a
        mov current_temp0,a
        ljmp l2_hextotemperature
```

```
c2_hextotemperature:
        dec adcdatamsb
c1_hextotemperature:
        clr c
        mov a,current_temp1
        add a,#10h
        da a
        mov current_temp1,a
        jnc l1_hextotemperature
        clr c
        mov a,current_temp0
        add a,#01h
        da a
        mov current_temp0,a
        ljmp l1_hextotemperature
        ret
;*************************************************************
;*************************************************************
/*
Displays welcome on the 1st line of LCD
*/
display_welcome:
        mov a,#80h                      //LCD Location where we have to display
welcome
        call lcd_command
        mov dptr,#db_welcome            //Point the DPTR where the display string is saved
        call lcd_displayline
        mov a,#0c0h                     //Now display CODE LOCK on second Line
        call lcd_command
        mov dptr,#db_code_lock          //Point the DPTR where the display string is saved
        call lcd_displayline
        ret

db_welcome: db '   WELCOME    ',0
db_code_lock: db 'TEMP. CONTROLLER',0


display_current_temp:
        mov a,#0C0h                     //LCD Location where we have to display
welcome
        call lcd_command
        mov dptr,#db_current_temp       //Point the DPTR where the display string is saved
        call lcd_displayline
```

```
        mov a,current_temp0                     //display MSB of temperature
        anl a,#0f0h
        swap a
        orl a,#30h
        call lcd_datadisplay
        mov a,current_temp0
        anl a,#0fh
        orl a,#30h
        call lcd_datadisplay

        mov a,current_temp1                     //display LSB of temperature
        anl a,#0f0h
        swap a
        orl a,#30h
        call lcd_datadisplay
        mov a,current_temp1
        anl a,#0fh
        orl a,#30h
        call lcd_datadisplay

        mov a,#0dfh                     ;Degree Centigrade
        call lcd_datadisplay
        mov a,#'C'
        call lcd_datadisplay
        ret
db_current_temp: db 'C. TEMP=+',0


display_set_temperature:
        mov a,#80h                      //LCD Location where we have to display
welcome
        call lcd_command
        mov dptr,#db_set_temp           //Point the DPTR where the display string is saved
        call lcd_displayline

        mov a,temp_setpoint0            //display MSB of temperature
        anl a,#0f0h
        swap a
        orl a,#30h
        call lcd_datadisplay
        mov a,temp_setpoint0
        anl a,#0fh
        orl a,#30h
```

```
        call lcd_datadisplay

        mov a,temp_setpoint1                    //display LSB of temperature
        anl a,#0f0h
        swap a
        orl a,#30h
        call lcd_datadisplay
        mov a,temp_setpoint1
        anl a,#0fh
        orl a,#30h
        call lcd_datadisplay

        mov a,#0dfh                             ;Degree Centigrade
        call lcd_datadisplay
        mov a,#'C'
        call lcd_datadisplay
        ret
db_set_temp: db 'S. TEMP= ',0


check_ir:
        jb ir,r1_check_ir

        call quater_ir_delay
        jnb ir,r1_check_ir
        call quater_ir_delay
        jb ir,r1_check_ir

        call get_rc5_bit                        //toggle bit
        jnc r1_check_ir

        mov r7,#05h
        mov rc5_system,#00h                     //five bytes of system code
l1_check_ir:
        call get_rc5_bit
        jnc r1_check_ir
        mov c,temp_bit
        mov a,rc5_system
        rlc a
        mov rc5_system,a
        djnz r7,l1_check_ir

        mov r7,#06h
        mov rc5_command,#00h                    //six bytes of command
l2_check_ir:
        call get_rc5_bit
```

```
        jnc r1_check_ir
        mov c,temp_bit
        mov a,rc5_command
        rlc a
        mov rc5_command,a
        djnz r7,l2_check_ir
        mov rc5_command,a

s1_check_ir:
        setb c                                  //return with c=1 indicatind data
received

        call check_code
        call delaya
        ret

r1_check_ir:
        clr c                                   //return with c=0 indicating no data
received
        ret
;*****************************************************
;*****************************************************
;*****************************************************
get_rc5_bit:
        call quater_ir_delay
        call quater_ir_delay
        mov c,ir
        call quater_ir_delay
        call quater_ir_delay
        jnb ir,c1_get_rc5_bit
        jb temp_bit,c2_get_rc5_bit
        setb c
        ret


c1_get_rc5_bit:
        jnb temp_bit,c2_get_rc5_bit
        setb c
        ret

c2_get_rc5_bit:
        clr c
        ret
;*****************************************************************
;*****************************************************************
;*****************************************************************
```

```
quater_ir_delay:
        mov delay1,#190
l1_quater_ir_delay:
        djnz delay1,l1_quater_ir_delay
        ret
;*************************************************************************
;*************************************************************************
;*************************************************************************
;*************************************************************************
;*************************************************************************
r1_check_code:
        clr c                                   //error code didn't match
        ret


check_code:
        mov a,rc5_system
        cjne a,#SYSTEM,r1_check_code            //check if the received system code is correct

        mov a,rc5_command
c11_check_code:
        cjne a,#VOLDEC,c12_check_code
        call dec_temp_setpoint
        ret

c12_check_code:
        cjne a,#VOLINC,c13_check_code
        call inc_temp_setpoint
        ret

c13_check_code:
        clr c                                   //error code didn't match
        ret
delaya:
        mov r7,#0ffh
l1_delaya:
        mov r6,#0ffh
l2_delaya:
        mov r5,#1
        djnz r5,$
        djnz r6,l2_delaya
        djnz r7,l1_delaya
        ret
```