# DESIGN AND CONSTRUCTION OF A MICROCONTROLLER BASED TEMPERATURE MONITORING DEVICE WITH AUTOMATIC CUT-OUT

BY

IMEVBORE A C ABIODUN

MATRIC. NO 2003/15382EE

Federal University of Technology, Minna

ELECTRICAL/COMPUTER ENGINEERING DEPARTMENT,

FEDERAL UNIVERSITY OF TECHNOLOGY MINNA

NOVEMBER, 2008

# DESIGN AND CONSTRUCTION OF A MICROCONTROLLER BASED TEMPERATURE MONITORING DEVICE WITH AUTOMATIC CUT-OUT

BY

IMEVBORE A C ABIODUN

MATRIC. NO 2003/15382EE

A Thesis submitted to the department of Electrical and Computer Engineering, Federal University of Technology, Minna

ELECTRICAL/COMPUTER ENGINEERING DEPARTMENT,

FEDERAL UNIVERSITY OF TECHNOLOGY MINNA

NOVEMBER, 2008

# DECLARATION

I Imevbore Christopher Aikhojie Abiodun, declare that this project was done by me and has never been presented elsewhere for the award of a degree. I also hereby relinquish this copyright to the FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA.

Imevbore A. Abiodun

.........................................

(Name of student)

...................17/11/08

(Signature and date)

Engr. J.A. Abolarinwa

.........................................

(Name of supervisor)

.......................17.11.2008

(Signature and date)

Engr. Dr. Y.A. Adediran

.........................................

(Name of HOD)

.........................................

(Signature and date)

.........................................

(Name of External Examiner

.........................................

(Signature and date)

# DEDICATION

This project work is dedicated to GOD almighty the author and finisher of our faith. To him be all the glory, honour, power, praise and majesty forever, Amen.

# ABSTRACT

The aim of this project is to design and construct a microcontroller based temperature monitoring and automatic cut-out device. This was achieved using an LM35 precision temperature sensor as the input unit of the design. The sensor was interfaced with an ADC that converts the output voltage signals from the sensor to binary. These binary codes are transferred through the ADC data bus interfaced with the AT80C59 Microcontroller input Port1. The Microcontroller is coded with a user's programme, which it uses to make logical decisions and manipulations. If these conditions are met, the microcontroller sends a signal that is driven by a transistor switch to activate a 12V 400Ω power relay. This relay initiates a switching "OFF" of a 5V power LED signifying a tripping-off of any electrical device that is connected to it. The triggering of the Relay is as a relationship with the temperature variations around the LM35 temperature sensor. All this while, as the ADC is converting the LM35's analogue output to digital signal, a number of IC's (seven segment) are used to provide for a discrete digital temperature read-out of the sensed temperature measured in °C unit.

# ACKNOWLEDGEMENT

Thanks to almighty GOD for his grace, mercies and endless support, without whom it couldn't have been possible.

Special thanks to my parents, Mr and Mrs S.E Imevbore and my siblings for their moral, financial and unending prayers during the course of this project work.

I also acknowledge the guidance of the Doherty Ishienmojie's family for every necessary support given me during the write-up of this work.

My final thanks to my supervisor Engr. J. A. Abolarinwa for providing me with materials, counsels, assistance, and discipline, without whom I wouldn't have learnt practically. Also to my colleagues, who rendered their precious time teaching me and providing me with both hard and soft copies of materials that were needful for the realisation of this work and to my friends, thanks for being there.

# TABLE OF CONTENT

CONTENTS PAGES

LIST OF FIGURES

# CHAPTER ONE

# INTRODUCTION

The microcontroller since the 1970's has made tremendous advances in the micro-electronics industries. With fewer configured and programmable chip, a lot of small and sizable electronic circuitries can be designed to achieve various task.

The microcontroller is a single computer chip that executes a user program to perform specific tasks on systems that are directly interfaced to it. Microcontroller based systems are normally smaller, cheaper, flexible, and friendly to operate and are also ideal for this cause of project which is aimed at measuring and controlling temperature of systems sensitive to temperature (Heat).

The microcontroller is an 8-bit device that is interfaced with other system components for the purpose of control. This device provides for a highly flexible and cost effective solution to many embedded control appliances. The choice for a microcontroller in this system is to overcome large circuitry size, a nd to provide for an automatic functionality in control devices as it incorporates various components in a single chip.

The microcontroller device is aimed at controlling temperature sensitive systems using an LM35 precision temperature sensor as an input device. The sensor outputs a voltage of 10mV for every degree rise in temperature and has a relationship with the power relay switch to trigger "OFF" a system under control. This design also provides for a digital temperature read-out of the sensed temperature, using a seven segment visual display and a light emitting diode to be automatically powered off if the ambient temperature exceeds the input preset temperature.

## 1.1 AIM AND OBJECTIVE

The aim of this project is to design, construct and test a microcontroller based temperature monitoring and automatic cut-out device. That will achieve the following;

i. Increases switching functionality
ii. Control automatically switching operations to a reasonable degree of accuracy
iii. Reduce the direct effort of humans to switching operations
iv. Produce a discrete temperature readout of a system under control
v. Protection of expensive mechanisms susceptible to temperature from damages in industries, power motors,, laboratories, hospitals etc.
vi. To gainfully expose the student to practical skills gained through theoretical knowledge received from the classroom.

## 1.2 METHODOLOGY

The microcontroller based device comprises of six modules namely:

1 The power unit
2 The temperature sensor (LM35)
3 The ADC circuit
4 The 89C51 microcontroller unit
5 The electromagnetic relay switch unit
6 The seven segment display unit.

### 1.2.1The power unit

This circuit provides the system with DC voltages for operations. This is the steady voltage used to drive the various components without damaging any of the devices.

### 1.2.2 The LM35 temperature sensor circuit

This circuit is the input unit of the system, converting temperature to voltage signals. It produces a 10mV signal for every degree rise in temperature.

### 1.2.3 The ADC circuit

This circuit converts the output voltage signal from the LM35 temperature sensor to digital values through its 1byte data bus to the microcontroller.

### 1.2.4 The Microcontroller circuit

The microcontroller used for this design is the AT89C55. It is a low power, high performance CMOS 8-bit microcomputer. This device uses a user program to make logical decisions on data sent to it from the ADC to control devices that are interfaced to it.

### 1.2.5 The electromechanical relay switch

This circuit turns "OFF" any electronic device connected to it when an input signal is energized through it from the microcontroller.

### 1.2.6 The visual seven segment display

This unit provides a visual digital readable display for the monitored temperature of the system.

## 1.3 PROJECT LAYOUT

This write up is divided into five chapters;

Chapter one focuses on the introduction giving a brief overview of the project topic, states the aim, objectives, scope of project, sources of materials used and the problem definitions.

Chapter two focuses on the literature review bringing to light historical developments, advancements, limitations if any on previous works.

Chapter three focuses on the circuit design and analysis, specifying the circuit design and analysis, their requirements, approaches to their achievements and defending their choice over contemporaries.

Chapter four focuses on the system construction and testing, performance of the system, evaluation, test and actual result of expected theoretical results.

Chapter five focuses on the summary of the project work specifying the project area of applications, problems encountered, limitations, references to sourced materials as well as suggestions for future improvements.

# CHAPTER TWO

# LITERATURE REVIEW

Motion control in electronics means to accurately control the movements of an object based on speed, load, inertia or the combination of all three [1]. Temperature is by far the most measured parameter impacting both the Physical, biological and chemical world in numerous ways yet, the application of the complexities of temperature and its measurements has been relatively slow to develop through the years. Some of the technological processes require a degree of control over temperature but to control temperature, one must be able to know what you are controlling [2].

Earliest attempts to measure and control temperature brought about the discoveries of devices and equipments to meet this task. Before these inventions, history has it that the Romans discovered the hypocaust a technology of heating the room. During summer it aids to keep the rooms warm. Its technology is such that heat is pressured through a furnace having a basement made of brick or a stone, the hot air passes through the flues in the walls of the room above the furnace and hence causing heating of the room. Measuring temperature by the first century involves carefully observing the temperature if it melts wax, sulphur, boil water or how red- hot metals glow depending on the applied pressure of heat. This of course, were unreliable because it could neither be read or clearly defined though their relevance of comfort to man [3].

Galileo in 1952 invented the first temperature document thermometer but it had limitations as it was affected by atmospheric pressures. The 18th century experienced a productive era in the development of temperature monitoring devices. In 1821, two scientists Sir Seeback and Sir Humphrey discovered the thermocouples; that current could be produced by unequally heating two dissimilar metals called the sensing junction the voltage developed across the two junctions is proportional to the difference between them. A measuring meter was connected across the circuit to obtain a voltage reading. It was also discovered that platinum could be used as an excellent temperature deter (RTD) because of its positive coefficient of resistance compared to that of other metals. This period marked the discoveries of different electrical sensors and measuring precisions of temperature unit [4, 5].

As technology advanced in the 19th Century, it was discovered that different metals expand at different rates, one bounded to another while heated will bend to one direction and when cooled would bend the opposite direction. The (BIMETS) were discovered to react this way and the bimetallic sensors were discovered to work with the principle of unequal expansion between two metals. The bimetallic switching device (Temperature sensitive switch) comprises of a fixed contact part held on the cover and a movable contact borne on a spring snap washer which is moved by a bimetallic snap switch. A first heating resistor is connected to the switching device (Bimetallic strip) which it locks in a self holding manner when switching device is activated in response to excess temperature. A second heating resistor is also connected with the switching device and then produces heat in response to current flow. When excess current flows through the second resistor, the switching device is instantaneously activated. Thus the device is triggered at both excess temperature and excess current [6].

In the 20th century, semiconductors devices like the IC's sensors, non-contact IC's, fiber-optics temperature sensors were discovered and used to experiment on various tests in temperature monitoring and control, using different inputs and output device. In implementing a temperature sensor for monitoring and control, the device is connected to a poly block in compare mode, when the temperature exceeds its upper preset limit a switch is triggered to activate the lightening of an electric bulb. The idea is that the temperature sensor produces an output voltage that is proportional to the temperature in degree Fahrenheit. The voltage is just the temperature divided by 100, so when its temperature output is 0V, the sensor outputs a voltage of 0.70v, a temperature to voltage relationship of (1/0.7). The idea is to use the IC sensor to compare the incoming voltage with a reference voltage of its own. The reference voltage is set by using the trim-pot dial on the sensor. As its comparing the temperature voltage relationship, if the ambient exceeds the preset temperature, activation is triggered to initiate the lightening of the bulb [7].

Basically, there are other temperature monitoring device that uses almost the same type of principle but different designs based on choice, this project forms the basis of incorporating a microcontroller to realize the desired result. With a few numbers of integrated circuits, requiring the smallest amount of physical space, requiring the least energy, flexible to use and costing as little as possible.

Although there are a number of limitations, the significance of this project is sincerely defined.

# CHAPTER THREE

## DESIGN AND IMPLEMENTATION

The microcontroller based temperature monitoring and automatic control unit is designed to sense temperature (HEAT) above a preset input value and automatically cut-off a device connected to it, displaying the temperature of the surrounding and system under surveillance. The device is made up of five units consisting of different components performing certain specified tasks.

This chapter shall focus on some of the characteristic features of this component as concise as possible and includes;

i. The power unit
ii. The temperature sensing device(LM35)
iii. The ADC
iv. The control unit(Microcontroller)
v. The display unit

The block diagram of the entire circuit system is shown below.

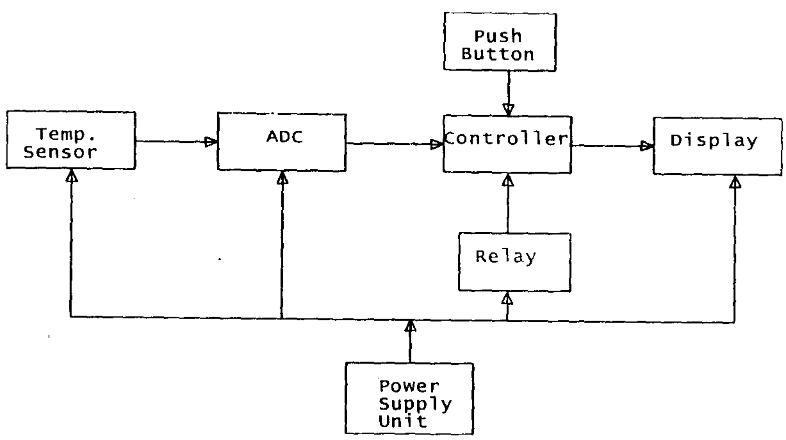



Fig 3.The system block diagram

## 3.1 THE POWER UNIT

The circuit requires a 12v and 5v dc for its operation. Using dry cell will require frequent replacements because of its electrolytic nature, offsetting these obvious disadvantages, the power supply is designed to be dependent on ac converted to dc at least for this project. This unit comprises of the transformer, the bridge rectifier, voltage regulations and filtering capacitors.

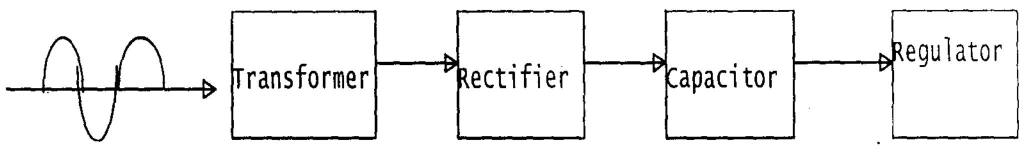



Fig3.2 Block diagram for the power unit

## 3.2 THE TRANSFORMER

The transformer used for this project is a 240v/12v 500mA 50Hz step- down transformer having lesser windings on the secondary side. The output of the transformer is connected as inputs to the bridge rectifier except for the heating element that takes ac supply of 220v from the mains.

For an ideal transformer,

Input VA = Output VA

$V_1 I_1 = V_2 I_2$

$V_1/V_2 = 1/K$

In this design, a transformer with the following characteristics was used;

1. Primary voltage $Vp = 240v$
2. $Secondary\ voltage\ Vs = 24v$

3. *Secondary current* $Is = 500mA$
4. *Primary current* $Ip =?$

*Using the transformer formular*

$Vp/Vs = Is/Ip$

$Ip = VsIs/Vp$

$(24 \cdot 500)/240$

$Ip = 50mA$

*This indicates that the transformer used for this project is a step – down transformer with ratings of* $240v, 50mA/24v, 500mA$.

## 3.3 THE RECTIFIER

A rectifier is a circuit that changes ac voltage to dc using switching elements or diodes. The process is called rectification and could be called a half wave or full wave though the full wave is more efficient because it converts both polarities of the input to dc. The bridge rectifier is a full wave containing either four element or six element connected as the arm of bridge circuit.

*The 24v rms ac* was rectified into a dc voltage value of

$Vrms\sqrt{2} - 1.4$

$\sqrt{2}$ = *Rms – to – peak connection factor*

$1.4 = 2 \times Vf$

$VF$ = *forward voltage drop of the two adjacent diode pair*

$Vdc = 24\sqrt{2} = 1.4 = 32.5411v$

## 3.4 SMOOTHENING

The dc voltage was smoothened using a minimum capacitance value deduced from the formula

$C = It/\Delta v$

I = load current and its given as 500mA

t= period for full wave rectification and its given $as\ 1/2f$

f = frequency

therefore, $t = ½ \times 50 = 0.01$

$\Delta V$
$= ripple\ voltage\ obtained\ as\ fullwave\ i.e\ the\ voltage\ delivrerd\ to\ the\ regulator\ and\ its\ given$

$\Delta V = Vdc - (Vout + 2)$

$Vdc = 32.54$

$Vout$ =forward voltage

2=minimum output – input differential voltage for the 7812

$\Delta v = 32.54 - (12 + 2)$

= 18.54 as minimum output voltage

$C = it/\Delta v$

$=500 \times 1000 \times 0.00053494 = 0.000269678f\ C1 = 269\mu f$

For better smoothening with manufacturer specification, a three fold /four fold of a capacitance value of 2200uf is used to eradicate ac ripples from the rectified ac voltage. A further capacitor of 330uf was also used for smoothening to improve system

Specification at low ac line voltages. The smoothed dc voltage was regulated further to 5v dc voltage using a 7805 regulator device as shown below.

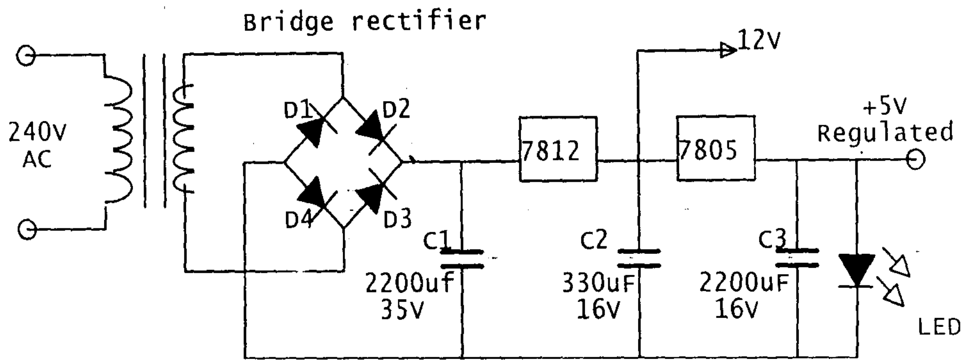



Fig 3.2 The AC to DC rectification circuit.

The regulated 12v dc was used to power the relay for ("ON/OF") switching operations and the 5v dc was also used to power the LM35, ADC, Microcontroller, the LEDS and the switching transistors in the circuit.

### 3.2. CAPACITOR

Capacitors are electrolytic material that accumulates charges when currents pass through them. A filtering capacitor is used to filter ac ripples from the rectifier to obtain a smoother ac voltage.

### 3.2 THE REGULATOR

A regulator is a controller designed to maintain the state of the controlled variable despite fluctuation of the load. It is also used to remove the last of ripples in the output from the rectifier. The 7812 and 7805 keep the terminal voltage of the dc supply constant at 12v and 15v respectively irrespective of variations in input ac voltage.

### 3.3 THE SENSING UNIT

This is the input unit of the design and it comprises of The LM35 is an integrated circuit that converts temperature into voltage signals (a transducer).its characteristic sensing range is between $0^0$c-$100^0$c. In this design, it is interfaced directly to an 8-bit

ADC that translates the 8-bit temperature to voltage signal into logical values that is readable by the Microcontroller.

The sensor's output is prevented and stabilized by a 4.7k resistance in series with 100uf capacitor which acts against incessant fluctuations from the displayed digital readout.

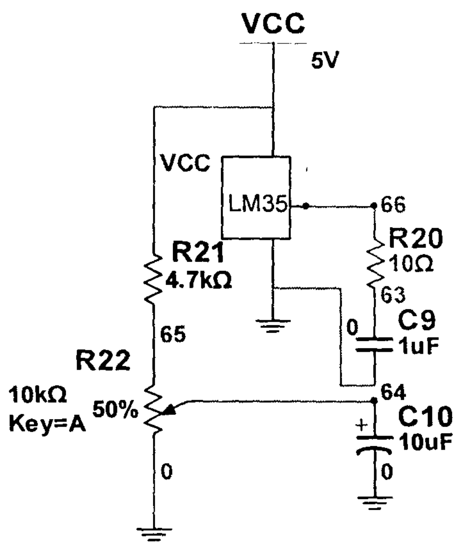



Fig 3.3 The temperature sensor (LM35)

### 3.4 THE CONVERSION UNIT

This unit comprises of the ADC (Analogue to Digital Converter)

The analogue to digital conversion unit performs the function of converting analogue signal to digital signals. It has the following features;

I. Easy interface to all microprocessors, or operates "stand alone"

II. Differential analog voltage inputs

III. Logic inputs and outputs meet both MOS and TTL

IV. Works with 2.5V (LM336) voltage reference

V. On-chip clock generator

VI. 0V to 5V analog input voltage range with single 5V supply.

VII. No zero adjust required

VIII. 0.3" standard width 20-pin DIP package

The input of the IC is pin 6 and 7 indicated as $V_{in}$+ and $V_{in}$-. The $V_{IN}$ was connected to the LM35 output and the Vin- to ground. The IC has an internal clock generator circuit that produces a frequency of F=1 where R and C are values of components connected to the CLKMV and CLK out PIN respectively. The clock frequency used is 606 KHz obtained gives a conversion time of approximately 100NS

The pin 8 and pin 1 is the digital ground terminal of the IC. The two pins are both connected to the ground of the design. Pin 20 is the terminal compacted to 5V power supply form the power unit for powering the IC. The pins P (2 – 9), that is D7 to D0 gives the 8bit binary output of the IC. The IC was design to be interfaced with the microcontroller, in order to execute some instructions. The functional circuit diagram is given below.

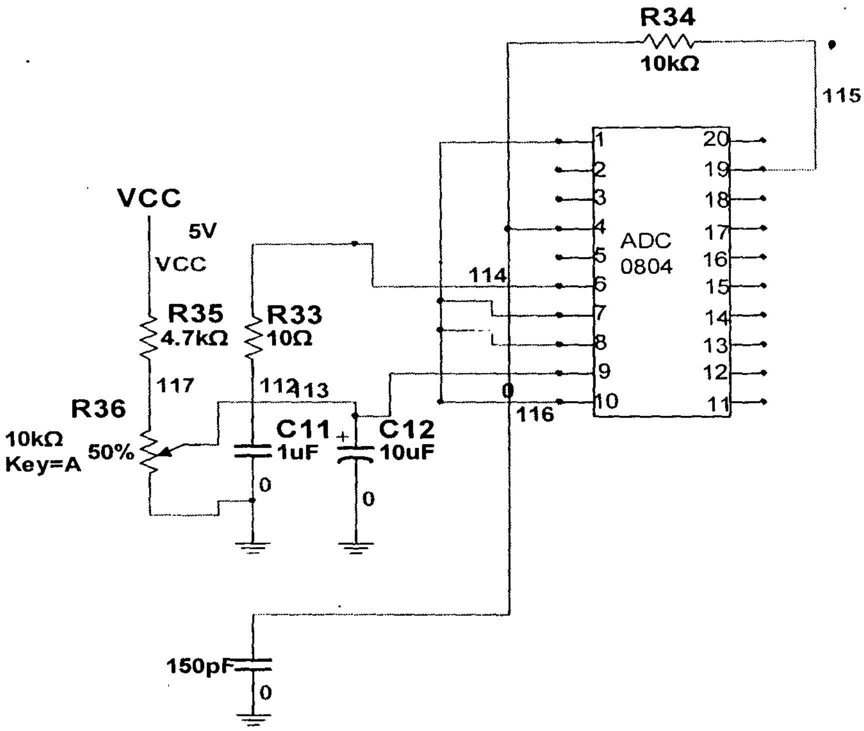



Fig 3.4 The ADC circuit configuration

The pin 9 that is $V_{reff}/2$ terminal is used to achieve the desired calibration since it adjusts the sensitivity of the device. The reference voltage is usually half the expected full analogue input. In this design, the sensitivity of the sensor is10MV/°C. Therefore for an accurate calibration, the reference voltage must in-line with the input sensors sensitivity. When the reference voltage is at 2.5V, the analogue input coverage is 5V. The sensitivity of the adjust is shown below:

$$2Vreff/255 = step\ size$$

$$Vreff = 2.5mV$$

$$Step\ size = 5/255 = 0.02V$$

$$Step\ size = 20mV$$

$$Step\ size = 10mV = 2(Vreff)/255$$

$$Vreff = 255 \cdot 10mV/2$$

$$Vreff = 1.275mV$$

### 3.5 THE CONTROLL UNIT

This is achieved by using the Microcontroller

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K-b of Flash Programmable and Erasable Read Only Memory (PEROM). The On-chip Flash allows the program memory to be reprogrammed in-system or by a conventional Nonvolatile memory programmer.

By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications. It provides the following standard features which are:

I. Compatible with MCS-51™ Products

II. 4K Bytes of In-System Reprogrammable Flash Memory

III. Endurance: 1,000 Write/Erase Cycles

VI. Fully Static Operation: 0 Hz to 24 MHz

V. Three-Level Program Memory Lock

VI. 128 x 8-Bit Internal RAM

VII. 32 Programmable I/O Lines

VIII. Two 16-Bit Timer/Counters

IX. Six Interrupt Sources

X. Programmable Serial Channel

XI. Low Power Idle and Power Down Modes

The microcontroller has four ports which are Port 0, Port 1, Port 2 and Port 3 which are explained below

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

### 3.5.2 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. Port1 also receives the low-order address bytes during Flash programming and verification.

### 3.5.3 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 pullups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that uses 16-bit addresses (MOVX @ DPTR). In this application it 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.

### 3.5.4 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 internal 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 also serves the functions of various special features of the AT89C51 as listed below: Port 3 also receives some control signals for Flash programming and verification.

### 3.5.5 THE DEVICE PORT PIN FUNCTION

Port 0 = Data Bus

Port 1 = ADC Bus

Port 2 = Control Bus

Port 3 = Push Button Bus

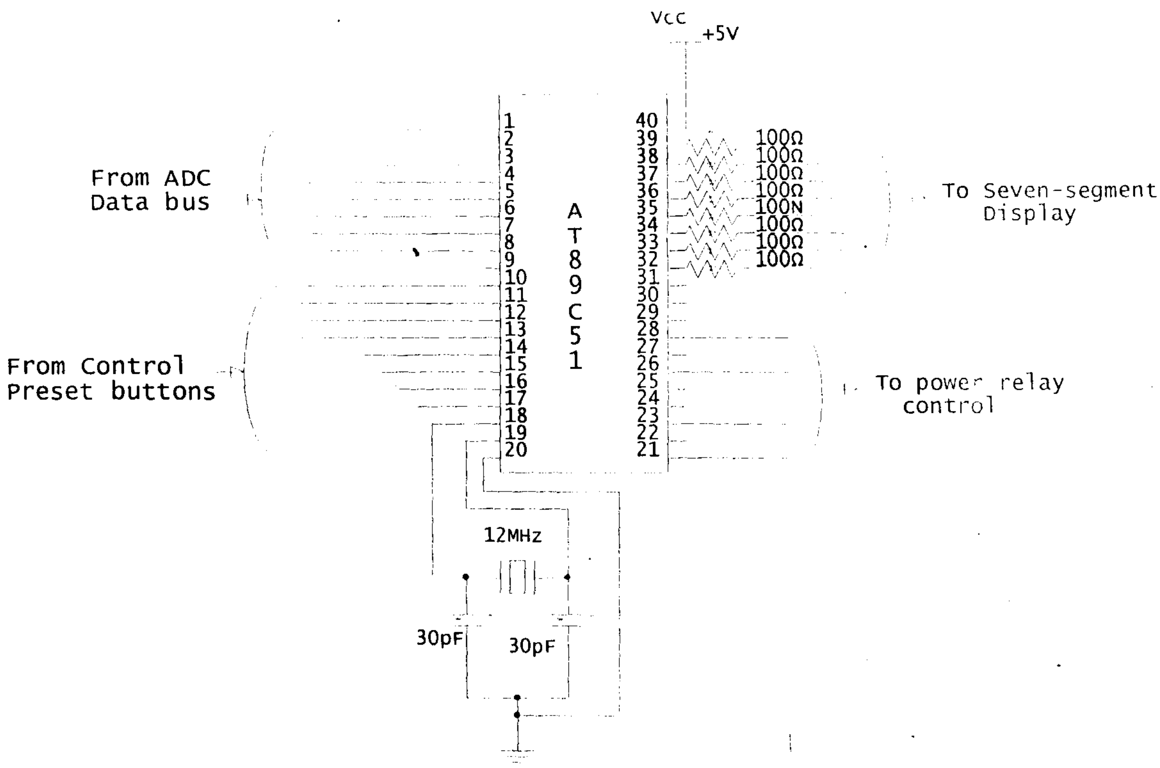
Port 0.0 – Port 0.7 = Display for the seven segment LED

Port 1.0 – Port 1.7 = Data transfer from ADC to Microcontroller

Port2.0 – Control Tx0, Port2.2 – Control Tx1, Port2.5 – Led Indicator, Port2.7 – Relay switching

Port3.0- Read Data, Port3.1 – Write Data, Port3.2 – Ext. Int.0, Port3.2 - Ext Int.1, Port3.4 - Mode, Port3.5 – Set, Port3.6 – Adjustable (+ve), Port3.7 – Adjustable(-ve)

The Microcontroller is interfaced with a number of IC's for the purpose of control. Below, in fig 3.5 is the orientation of the circuit pin configuration of the Microcontroller as used in the design of this project.




MICROCONTROLLER PIN CONFIGURATION

### 3.6 THE RELAY

A relay is an electromechanical devise for switching on and off devices with a relatively low electrical signal compared to the arising voltages. They are composed basically of two parts.

1. Operating coil

2 .Magnetic coil

When an input is introduced in the coil, a magnetic field is produced in the core of the electromagnet this action causes the switch to slide. Relays are either normally open or normally closed and are available for ac or dc excitation with a coil voltage range from 5v to 230v. The primary uses of relays are for remote switching, whereby the circuit to be switched is electrically isolated from the switching circuit. One relay was used to control the node. The relay was wired in series into the load, when an over current is detected as a result of increase in temperature variation around the sensor, the relay opens under software control by a time determined by the systems software.

The choice relay for this project work is a 12v / 400Ω dc relay for a reliable drive and efficiency.

### 3.6.1 RELAY CALCULATION

The dc relay had a coil resistance of 12v, 400Ω.

Requiring a coil current ( Ic) of

$IC - V/R$

V = relay voltage

R = relay resistance

$V = 12 \, and \, R = 400\Omega$

$IC = 12/400 = 30\mu A$

$Ib = ic / Hpe$

$Ic = 30uA$

$Hpe - 200$

$Ib = 30/200 = 15uA$

$Ib = 1.5uA$

$Rb = Vcc - Vce/Ib$

$Vcc = 5v$

$Vce = 0.7$

$Ib = 15uA$

$Rb =$

$$Rb = Vcc - \frac{Vce}{Ibc}$$

(5 − 0.7)/0.0015

0.000287Ω

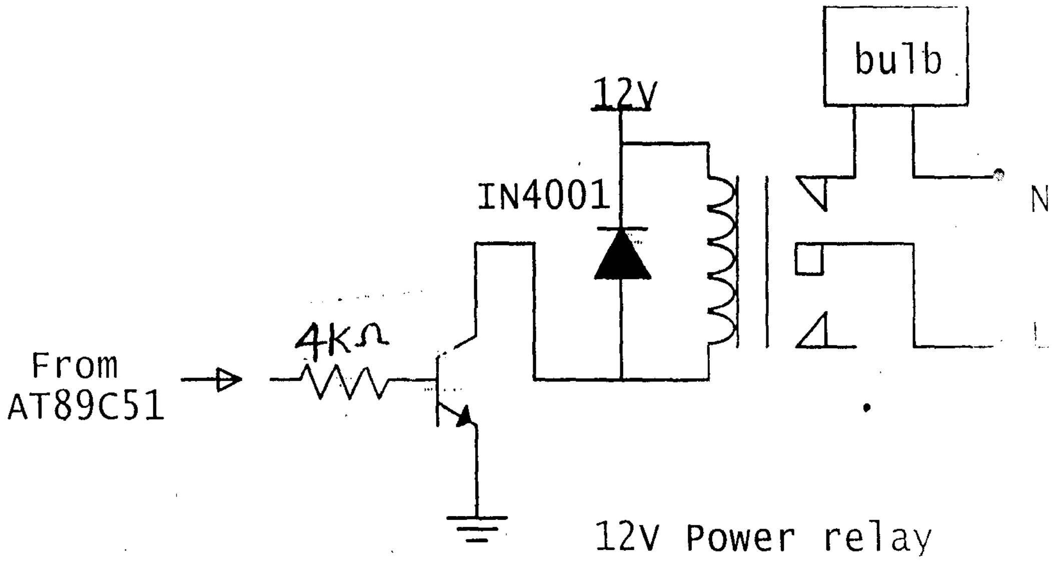



Fig 3.6. The relay circuit

To protect against inductive kick-back generated by relay turn-off, a freewheeling diode was placed in parallel with the relay to absorb inductive kickback generated by relay turn-off, thereby protecting the control device in the circuit. For guaranteed reliable smoothing and overdrive at all possible system operation states, a resistor of $A$ 4.7$k\Omega$ value was used.

### 3.7 THE DISPLAY UNIT

This unit is made up of the seven segment display LED. This is made of seven LEDs plus an eight represented by a dot for decimal point if needed. A common technique is to use a shaped piece of translucent plastic to operate as a specialized optical fiber, to distribute the light from the LED evenly over a fixed bar shape. The seven bars are laid out as a squared-off figure "8". The result is known as a seven segment LED. Common anode display LED was used for this design because of its sharp luminance. A 3-digit seven segment display of this sort was used and it was interfaced directly with the Microcontroller for a visible display using the port 0 interface line from the Microcontroller. The value for base resistance is deduced using the formula below;

$Rb = Vcc - Vce/Ib$

$But\ Ib = Vcc - Vce/Rb$

$5 - 0.7/4.7$

$= 0.91489A$

$Rb = Vcc - Vce/Ib$

$5 - 0.7/0.91489$

$= 4.7000\Omega$

Therefore, a base resistance of 4.7kΩ chosen for this project. And for a continuous display, a 50 Hz frequency speed was used.

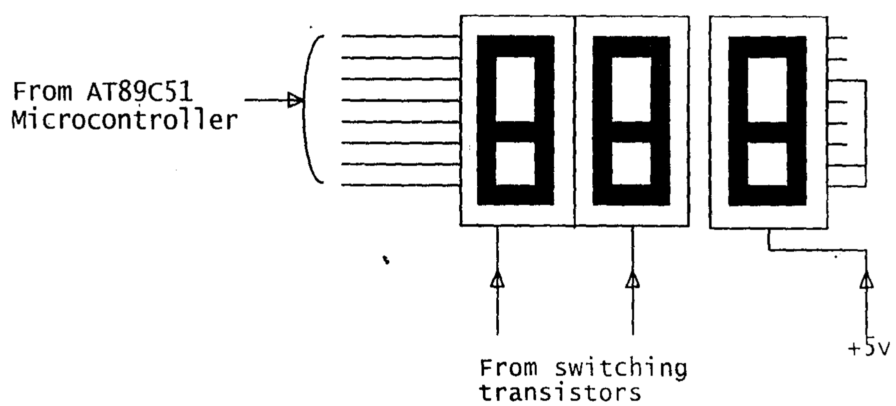



Fig 3.7 The seven segment display LED (Multiplexed)

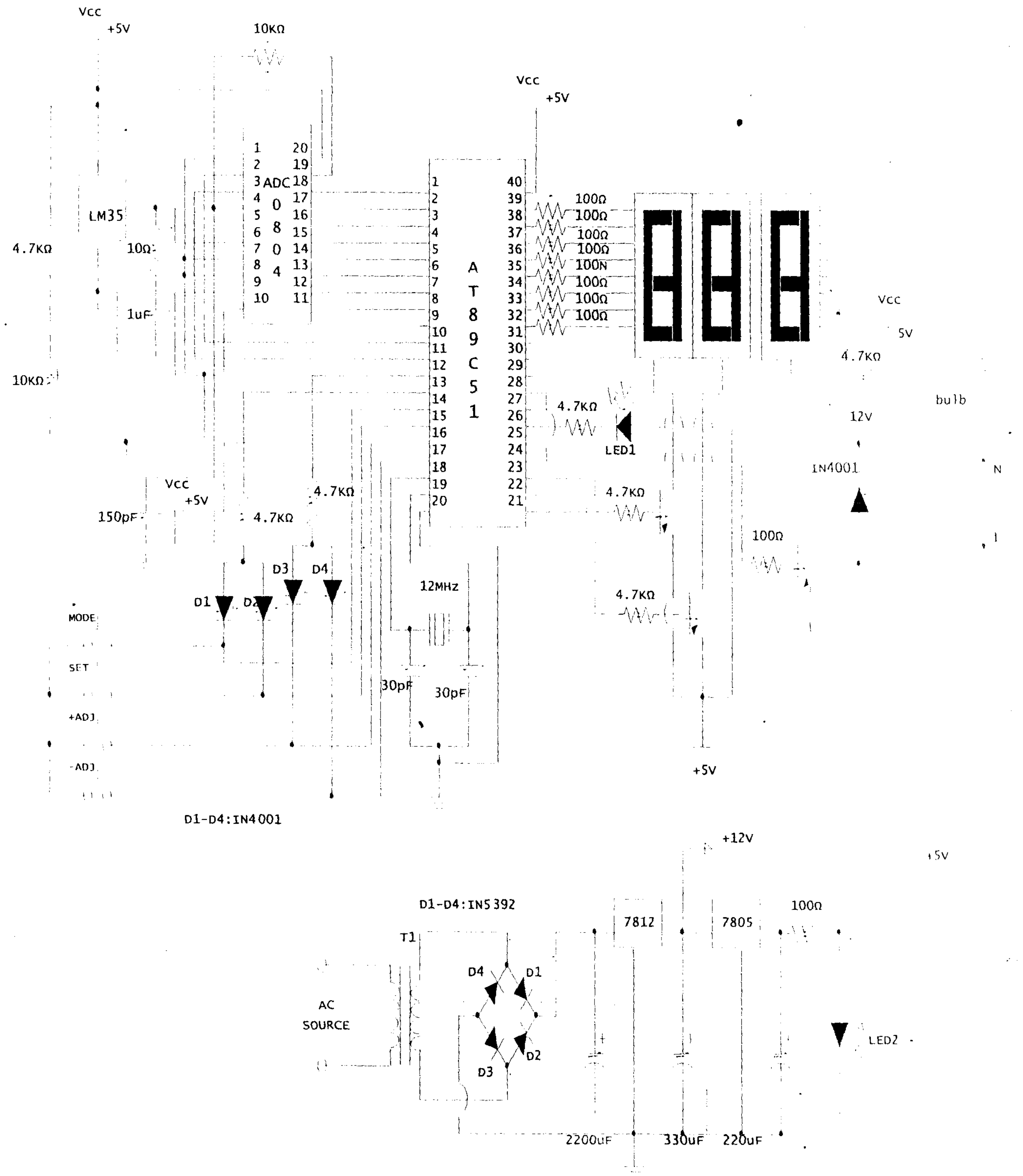



**Fig 3.8 Circuit diagram of a Microcontroller based Temperature Monitoring and automatic CUT OFF device.**

# CHAPTER FOUR

## TESTS, RESULTS AND DISCUSSIONS

All components purchased for this project were technically tested to determine their operability, stated values, continuity and durable functionalities. Testing was carried out using the digital multimeter. Most of the components had actual values less than their stated values though falling between their manufacturers operating range.

This part of the project involves practical exercises in making the circuit diagram on the paper into a real working hardware. The specified were carefully connected together under the guide of the circuit diagram. The entire circuits were divided into several units and each unit was tested one after the other, after which they were joined together as a single working construction. Other construction tools used for the implementation include;

1 Razor blades

2 Digital multimeter

3 Scissors

4 Precision tools

5 Soldering iron

6 Blower

The construction was quite delicate involving care and reasoning. The software and hardware implementation were also handled with care, soldering on the Vero board involves the technique of disallowing the IC's pin configuration to bridge.

After troubleshooting and modification of the device, the final circuit was tested several times and various results were obtained repeatedly.

## 4.1TEST

Tests were performed on the completed construction so as to check its response and performance with the aim of the project. The testing basically depends on the temperature regularities of the systems to the system on test.

Firstly, it involved the plugging in of the device to the ac mains power supply and powering it "ON" by the use of the set push button. This operation initiates the active/working mode of the device. The preset temperature was initialized with the increment and decrement switch push buttons to a desired temperature value which corresponds to the display on the seven segment display LED. The relationship is that of the LM35 temperature sensor which is 10°C *to* 10mV.

The digital display provides an accurate and precise temperature readout that is displayed by the seven segments LEDS. The preset buttons are well labeled for easy visual access to the temperature ranges on the rubber casings. They are also used to monitor the temperature reading and preset of the device. The response to the results of the two temperature (reading and preset) getting quite close or the same were properly monitored during the course of testing. The MC-AT89C51 replaced the op-amp, LM 339 and 555 Timer circuit combinations providing more efficient voltage comparison for overload condition and computing the timing delay ON and OFF of the relay switch contact

Numerous preset temperatures within the range of $0^0$c -$100^0$c were involved in the test.

## 4.2 DISCUSSION OF RESULT

The results obtained at the end of repeated tests were found to be consistent and matched up to expected results. Thus the aim of the project was met.

From the test result, it was observed that the operation of the device centers on switching "ON" and "OFF" at a preset temperature while displaying the ambient temperature.

## 4.3 PROJECT LIMITATIONS

1 The design functionality is programmed for temperature range of ($0^0$c-$99^0$c).

2 The design is not liquid-proof therefore it cannot be used in wet areas

# CHAPTER FIVE

## CONCLUSION AND RECOMMENDATION

### 5.1 CONCLUSION

The design and construction of a Microcontroller based temperature monitoring and control device for measuring temperature and for remote switching operations has been successfully designed. The switching operations activates at the desired preset temperatures. Also, a precise temperature display read-out of the ambient temperature was simultaneously obtained. The various result obtained where close to theoretical results

The project provided a practical knowledge to electronics components and design construction.

### 5.2 RECOMMENDATIONS

Having completed this project, the following are hereby recommended for future improvement

1. A provision for wider temperature readout could be incorporated using both a sensor of a higher temperature rating and several multiplexed digital visual display.
2. A modification in the user program to control various devices could be modified on.
3. An alarm unit for audible alert can be modified into the system.
4. For a reduction in the circuit space, a specific Microcontroller that incorporates the circuitry of an ADC can be used to realize same result

5 When it's interfaced directly with the LM35 precision temperature sensor.

# REFERENCES.

[1] Zubeiru S, Industrial electronics design, lecture handout on motion control, (6/26/2008) , federal university of technology minna.. [7/07/2008]. pp 12-14.

[2] Automatic switches (1998 copyright) [online] available

http//www.seg.co.uk/telecom/automatic.httm(2008,august.19)

[3] M.W.Anyakoha. New school physics for the experiment,.[2008,july 12] pp 44-49.

[4] William Hashleyand Emmanuel Fredman , merits students encyclopedia macmillian education company 1982, [sept,12,2008]

[5] Introduction to temperature controller, (2006 copyright) online available.

http//www.omega.com/prodinfo/temperature controllers.html [2008,sept 16]

[6] Temperature sensitive switch(2002-2004-copyright)

http//www.patentsform.us/patents/5615072fulltext.(2008,sept.12)

[7] Robert l Boylestad and Lows Nashkey,[2008], Electronic device and circuit theory eight edition pp 55-58.

[8] http;//encyclopedia.org/wiki/thermostat [24/07/2008]

[9] http;//early thermometer history.us/sec001.htm. [16/08/2008]

```
bDisplay0   equ P2.0
bDisplay1   equ P2.2


Indicator   equ P2.5
Relay       equ P2.7


ADCRead        equ P3.0
ADCWrite       equ P3.1
ExtInt0        equ P3.2
ExtInt1        equ P3.3
bMode              equ P3.4
bEnable            equ P3.5
bAdjustP           equ P3.6
bAdjustN           equ P3.7

;===========================================================
;Constant
Timer0H       equ 248
Timer0L       equ 204

Timer1H       equ 11
Timer1L       equ 219
;===========================================================
;===========================================================
;    VECTOR ADDRESSES
;===========================================================
;===========================================================

;===========================================================
     Org 0000h          ;RESET VECTOR ADDRESS
     ljmp Start         ;Jump to start of program
;===========================================================
     Org 0003h          ;EXTERNAL INTERRUPT0 VECTOR ADDRESS
     acall lExtInt0
         reti
;===========================================================
     Org 000Bh          ;TIMER0 INTERRUPT VECTOR ADDRESS

         acall lDisplay

         reti                        ;Not used
;===========================================================
     Org 0013h          ;EXTERNAL INTERRUPT1 VECTOR ADDRESS


         acall lExtInt1

         reti
;===========================================================
     Org 001Bh          ;TIMER1 INTERRUPT VECTOR ADDRESS
     clr TR1                ;Stop Timer0

         acall nReadADC

         mov TH1,#Timer1H       ;Timer1 reload value= 3035 (bdb)
     mov TL1,#Timer1L
         setb TR1                    ,Start Timer1

         reti                        ;Not used
;===========================================================
     Org 0023h          ;SERIAL INTERRUPT VECTOR ADDRESS
```

```
    reti                ;Not used
;===========================================================
    Org 002Bh           ;TIMER2 INTERRUPT VECTOR ADDRESS
    reti                ;Not used
;===========================================================
        Org 0038h           ;Program starts here
Start:
    mov SP,#40h         ;Stack Pionter intialized

        clr RS0             ;Bank0 selected
    clr RS1

        setb ADCRead
    setb ADCWrite
    setb ExtInt0
        setb ExtInt1
        setb bMode
        setb bEnable
        setb bAdjustP
        setb bAdjustN

        clr bDisplay1

        mov PresetTemp,#99
        mov ControlBus,#0

        mov cDigit0,#192
        mov cDigit1,#192

        mov pDigit0,#144
        mov pDigit1,#144

        clr ModeFlag
        clr EnableFlag

        clr Relay
        setb Indicator

    mov TH0,#Timer0H        ;Timer0 reload value= 63470
    mov TL0,#Timer0L
        setb ET0            ;Timer0 Interrupt enabled
        setb TR0            ;Start Timer0
    setb PT0

        mov TH1,#Timer1H        ;Timer1 reload value= 3035
    mov TL1,#Timer1H
        setb ET1            ;Timer1 Interrupt enabled
        setb TR1            ;Start Timer1


        mov ADCTimer,#2

        setb EX0            ;External Interrupt0 enabled
    clr IT0             ;Extrenal Interrupt0 on 1-0 transition
    setb EX1            ;External Interrupt1 enabled
    clr IT1             ;Extrenal Interrupt1 on 1-0 transition



        setb EA             ;Enable Global Interrupt

        ajmp $
```

```
;=========================================================
;=========================================================
;    INTERRUPT CALLS
;=========================================================
;=========================================================
lExtInt0:

        jnb bMode,lMode
        jnb bEnable,lEnable
        ajmp EndEX0

lMode:

        jnb ModeFlag,lMode1

lMode0:
        clr ModeFlag
        ajmp EndEX0
lMode1:
        setb ModeFlag
        ajmp EndEX0

lEnable:

        jnb EnableFlag,lEnable1

lEnable0:
        clr EnableFlag
        setb Indicator
        clr Relay
        ajmp EndEX0
lEnable1:
        setb EnableFlag
    clr Indicator
        setb Relay
        ajmp EndEX0

EndEX0:
        acall Debounce
        acall Debounce
        acall Debounce
        ret

;=========================================================
lExtInt1:
        push acc

    mov A,PSW
        push acc
        mov A,BCD
        push acc
        mov A,Digit0
        push acc
        mov A,Digit1
        push acc

        jnb ModeFlag,EndEX1
        jnb bAdjustP,lAdjustP
        jnb bAdjustN,lAdjustN
        ajmp EndEX1

lAdjustP:
```

```
        inc PresetTemp
        mov A,PresetTemp
        cjne A,#99, NEqualAdjP
        ajmp EndAdjP

NEqualAdjP:
        jc EndAdjP
        mov PresetTemp,#99
        ajmp EndAdjP

EndAdjP:
        mov BCD,PresetTemp
        acall Convert
        mov pDigit0,Digit0
        mov pDigit1,Digit1
        ajmp EndEX1

lAdjustN:

        dec PresetTemp
        mov A,PresetTemp
        cjne A,#255,NEAdjN
        mov PresetTemp,#0
        ajmp EndAdjN

NEAdjN:

EndAdjN:
        clr EA
        mov BCD,PresetTemp
        acall Convert
        mov pDigit0,Digit0
        mov pDigit1,Digit1
        setb EA

        ajmp EndEX1

EndEX1:
        acall Debounce
        acall Debounce


        pop acc
        mov Digit1,A
        pop acc
        mov Digit0,A
        pop acc
        mov BCD,A
        pop acc
        mov PSW,A
        pop acc
        ret
;============================================================
lDisplay:
        mov A,PSW
        push acc

        jnb bDisplay0,lDisplay1
        jnb bDisplay1,lDisplay0
        ajmp EndDisplay

lDisplay0:
```

```
        setb bDisplay1
        clr bDisplay0

        jb ModeFlag,dPreset0
        mov DataBus,cDigit0
    ajmp EndDisplay
dPreset0:
        mov DataBus,pDigit0
    ajmp EndDisplay


lDisplay1:
        setb bDisplay0
        clr bDisplay1

        jb ModeFlag,dPreset1
        mov DataBus,cDigit1
    ajmp EndDisplay
dPreset1:
        mov DataBus,pDigit1
    ajmp EndDisplay

EndDisplay:

        mov TH0,#Timer0H        ;Timer1 reload value= 3035 (bdb)
    mov TL0,#Timer0L
        setb TR0

        pop acc
        mov PSW,acc
        ret
;===========================================================

;===========================================================

nReadADC:

        push acc

    mov A,PSW
        push acc
        mov A,BCD
        push acc
        mov A,Digit0
        push acc
        mov A,Digit1
        push acc


     djnz ADCTimer,EndRADC
     mov ADCTimer,2

lRead:

         clr ADCWrite
         acall Delay0
         setb  ADCWrite

         acall Delay
         acall Delay
         acall Delay
```

```
        acall Delay
        acall Delay
        acall Delay


        clr ADCRead
        acall Delay0

        mov CurentTemp,ADCBus

        setb ADCRead

        acall Delay

        acall CheckSet

        clr EA
        mov BCD,CurentTemp
        acall Convert
        mov cDigit0,Digit0
        mov cDigit1,Digit1
        setb EA

EndRADC:
    pop acc
        mov Digit1,A
        pop acc
        mov Digit0,A
        pop acc
        mov BCD,A
        pop acc
        mov PSW,A
        pop acc
        ret

;==============================================================
;==============================================================
;     SUBROUTINE CALLS
;==============================================================
;==============================================================
Delay:

        mov Ctr1,#255
Delay1:
        mov Ctr2,#255
        djnz Ctr2,$
        djnz Ctr1,delay1
        ret

Delay0:

            mov Ctr2,#255
        djnz Ctr2,$
                ret
;==============================================================
Debounce:
        acall Delay
        ret
;==============================================================
CheckSet:

        jb EnableFlag,Compare
        ret
```

```
Compare:
        mov A,CurentTemp
        cjne A,PresetTemp,NECompare
        ajmp OffRelay

NECompare:
        jc EndComp
OffRelay:
        setb Indicator
        clr Relay
EndComp:
        ret
;===========================================================
Convert:
        mov A,BCD
        mov B,#10
        div AB
        mov Digit0,B
        mov Digit1,A

        mov Dptr,#Seg
        mov A,Digit0
        movc A,@A+Dptr
        mov Digit0,A

        mov A,Digit1
        movc A,@A+Dptr
        mov Digit1,A

        ret

;===========================================================
;===========================================================
Seg:
    db 192,249,164,176,153,146,130,248,128,144
;===========================================================
;===========================================================
         end
```