DESIGN AND CONSTRUCTION OF ELECTRONIC OVER/ UNDER VOLTAGE MONITORING, CUT-OUT UNITS.

BY

NTUKIDEM EFFIONG JAMES

2003/15415EE

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NOVEMBER, 2008

DESIGN AND CONSTRUCTION OF ELECTRONIC OVER/UNDER VOLTAGE MONITORING, CUT-OUT.

NTUKIDEM EFFIONG JAMES.

A THESIS SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING, IN PARTIAL FULFILLMENT OF REQUIREMENTS FOR THE AWARD OF A B.ENG DEGREE
FEDERAL UNIVERSITY OF TECHNOLOGY
MINNA, NIGER STATE, NIGERIA

NOVEMBER, 2008

DEDICATION

The project is dedicated to the glory of ALMIGHTY GOD for his mercies and grace in the course of my study. Also to my parents Engr. And Mrs. .E. Ntukidem for their Unflinching support and prayers.

DECLARATION

I Ntukidem Effiong James, declare that this work was done by me and has never been presented elsewhere for the award of a degree. I also here by relinquish the copyright to the Federal University of Technology, Minna.

NTUKIDEM .E. JAMES.

(Name of Student)

11/11/2008

(Signature/Date)

MR .N. SALAWU.

(Name of Supervisor)

11/11/08

(Signature/Date)

ENGR. Dr.Y. A. ADEDIRAN

(Name of H.O.D)

(Signature/Date)

(Name of External Supervisor)

(Signature/Date)

ACKNOWLEDGEMENT

Firstly, I give all glory, praise, honour and adoration to ALMIGHTY GOD who has been very faithful and good to me all the days of life in school. Secondly my joy appreciation goes to my parents who were always there for me.

My profound appreciation gratitude goes to my willing, listening, patience and accommodating project supervisor in person of Mr. Nathaniel Salawu, whose enormous contribution cannot be overlooked, and my Head of Department in person of Engr. Dr Y. A. Adediran

I also want to use this medium to say thank you to my sitters and brothers (Glory, Imeh, Alice, ini, and sister Affiong), may the good lord continue to guide you and bless you. I cannot forget Mr. and Mrs. Inyang and Mrs. Bose Okeneye for their financial support and prayers support. My appreciation also goes to my aunties and uncles. I cannot forget to say a bigger thank to my friends that have seen me through to this point of my life. My thanks goes to Anderson, Peace, Joy, Chinex, Niyi, Abdulbasit Subair , Eddy, Eno, Tayo, Sikiru, Samuel, James Emenoge, Beauty, seun. Ebenezer, Mrs Martha Dauda, Sa'adatu etc.

Lastly my thanks also goes to the distinguished lecturers of the department of electrical and computer engineering whose works have been tremendously used as an authoritative signpost.

ABSTRACT

Frequent fluctuation of power line and power failure has shown that electrical components have had their life span shortened. Practically, no power system can maintain voltage at rated valued at the consumer terminal at all times. Over voltage or under voltage may arise. Causes of this abnormalities or problem include the ability to have a balance network, switching surges, thunderstorm, physical contact between conductors of an electrical circuit, etc. This malfunction due to these condition may either reversible or irreversible. In the second case, the properties will have change sufficiently operation to be impossible.

This may range from component unwanted change to total failure of the device. It is in this light of the aforementioned problems that efforts are being made to ensure the development and design of protective device with sensitivity and efficient operating ability.

TABLE OF CONTENT

Title..... i

Dedication..... ii

Declaration..... iii

Acknowledgment..... iv

Abstract..... v

List of figures..... viii

Lists of table..... ix

Chapter one: General introduction

1.0 Introduction.....1

1.1 Project aims and objectives.....3

1.2 project methodology.....3

1.3 project source of materials used and constraint.....3

Chapter two: literature review

2.0 Literature review.....4

2.1Historical background.....4

2.2 Previous work of others.....5

2.3 Theoretical background.....6-7

Chapter three: Design and construction

3.0 Circuit diagram and calculation.....8

3.1Power supply..... 8-9.

3.2 8bits- Analogue-digital converter.....10-13

3.3 8bits system controller.....14-16

3.4 Load switch.....17

3.5 4 digit visual display.....18-20

3.6 256 bytes non volatile memory.....21-22

3.7 AC line voltage computation.....23

3.8 Software subsystem.....24-25

3.9 Complete circuit.....26

Chapter four: test, result and discussion

4.1 Test.....27

4.2 Result.....27

4.3 Discussion.....27

4.4 Summary.....28

Chapter five: conclusion

5.1 Conclusion.....29

5.2 Recommendation.....29

Reference.....30

Appendix A..... 31

Appendix B.....32-47

LIST OF FIGURES

Fig. 3.1 Power supply.....9

Fig. 3.2(a) Pin diagram of ADC 0804.....10

Fig. 3.2(b) ADC system connection.....13

Fig. 3.2(c) 50kΩ illustration.....14

Fig. 3.3 Port diagram of 89C51.....15

Fig. 3.4 load switch.....18

Fig. 3.5(a) 4-digit multiplexed display.....20

Fig. 3.6 24C02 EEPROM.....22

Fig. Complete circuit.....26

LIST TABLES

Table 3.2 Pin label function 11

Table 3.3 port label function of 89C51 15

Table 4.4 showing input voltage against output state 28

CHAPTER ONE

1.0 INTRODUCTION

Incessant and rising inconsistencies power failure in Nigeria status shows that Power supply transmission and distribution are steady on the decrease and voltage fluctuation (over/under voltage) are steadly power supply has become a cliché and the importance of voltage protection for electrical appliances in case of voltage cannot be over emphasized. This has damage our domestic and industrial appliance and has also led to the decrease in demand for electronics which is an integral part of our society which is on the increase. Due to development, improvement in circuit protection devices to protect this system cannot be avoided as technology advances.

There are various common power problems associated with distribution and transmission of electric power (with typical examples of damages that it causes). Over-voltage- increase voltage above the required volt for a long period of time causes light bulb to fail. Over-under is caused by malfunction of alternation, voltage regulation, poorly adjusted 'fast charge' controllers, battery charges and solar panels. Causes of this are static e.g. wind, sand and dust can become highly charged and impart relatively high voltage to expose overhead electronic conductors. (b) switching surges- due to inductive nature of system load an abrupt switching off a circuit causes an over voltage in the system, it is common with motor / auto-transformer and lines with poor power factor connection.[4].

Under-voltage- the decrease in voltage lower than stipulated voltage over a long period time causes damages in appliances e.g. blowing of devices embedded in the appliance. Voltage- spike- transient over voltage i.e. spike or peak, causes wear or

acute damage to electronic equipments. Power failure- total loss of utility power causes electrical appliances to stop working. This has actually led to the research and discovery of so many under /over-voltage. Regulator to solve this problem, but most of this regulators are analogue in nature and do not have devices like non-volatile storage I.C, micro controller, voltage to frequency converters I.C, not user friendly i.e. having a 4-digital-7-segment display which makes the analogue regulation not 95% reliable.[1]

This project tag: DESIGN AND CONSTRUCTION OF OVER / UNDERVOLTAGE MONITORING UNIT. Comprises of:

i. A line – ADC system,
ii. A central processing unit
iii. A non- volatile memory storage unit
iv. An A.C electromechanical relay
v. 4 – digital -7 –segment display unit
vi. User interface in the form of switch

The local A.C main line voltage is rectified to produce a 12V DC. The DC input voltage is fed into a ADC that convert the analogue signal to digital signal that is fed into a timer on the micro-controller where it is converted to its absolute value when gated for a desired window period. The computed value of the line voltage is compared with the settings determine by the User. The line voltage is compare with both the lower and upper user presets. If the live Voltage falls inside of the window, the protective feature is activated, i.e. the load is disconnection until the line voltage fall back within the window. A high power 10A electromechanical relay to provide A.C switching.

The User setting is keyed into the system via push buttons and stored in non- volatile memory to provide a semi permanent storage. These settings are retained on the event of power failure. The voltage level is displayed on the 7-segment display for visual interaction.

1.1 Project's aims and objectives

In order to maintain durability of electronic and electrical devices, by avoiding Fluctuations in voltage either over or under condition, this project as the name implies, monitors voltage fluctuation be it high or low and also display using the 7 segment display by means of micro controller chip.

1.2 Project's methodology

The design and construction of this project requires the employments of a noise-filters, micro-controller, non-volatile memory storage unit, 4-digital-7-segment display unit, line-monitoring conversion subsystem, user interface in form of switch, relay etc.

1.3 Project's sources of materials used and constraints

With the assistance of previous set of students inside and outside the department, most of the materials used for this project were purchased from the market and a few from past projects It was not easy getting the whole project's components at a time, as they were not readily available at the first attempt of purchase.

CHAPTER TWO

2.0 Literature review

The problem of voltage fluctuation due to improper maintainers of the balancing of the Network of either the star or delta became more pronounced when the transmission and a distribution stations were so far from the consumers. However, as a solution to these problems, one of the earliest attempts of obtaining regulated and stable voltages employed diffident process e.g. different design using transistor, comparators to regulate voltages etc. some of this are costly, bulky, not 100% reliable and the mechanical parts easily wear out, resulting to improper contact between taps of the transformers.

2.1 Historical background

Electricity was discovered by Michael Faraday (1791-1867)[2]. After its discovery, Voltage drop and resistance losses on the voltage D.C distribution circuit were the first set of problem encountered in its utilization. For electronic equipment to perform well, they need a certain stability of the power supply. As a solution to these problems, however, a remarkable was made by George Westinghouse by introducing the A.C transformer. From This idea of the transformer, came out many approaches of voltage regulations [2].

2.2 Previous works of others

The present technological dispensation has changed voltage stabilization and protection technique greatly. It came up with an approach known as regulated D.C inversion [1]. This uses the principle of switch-mode power supply. The regulated D.C output from the power supply is inverted, using pus-pull inversion and stepped up to the required constant A.C output voltage using a transformer. It uses zener diode and potentiometers for voltage protection. The output of the system is a square wave A.C voltage which could be filtered to obtain a pure sinusoidal. This method produces good regulation and protection. The system is usually not bulky but very expensive to construct than other system [1]

Another is the construction of an improved over voltage and under voltage cut-out device consists of the following; power system units, Schmitt trigger, load and visual indication of over voltage, under voltage and normal line voltage. The input stage consist of a 240/15V transformer which keeps tag of the min voltage and supplies a corresponding stepped down value to the rectifier and filter stage.[5]

The power supply units consist of rectifier and smoothing capacitor, full wave rectifier which convert the A.C voltage i.e. input to a D.C value that is used in operating the entire circuit. The Schmitt triggers consist of comparator and resistor which are basically used to sense different variation of the voltage. The load switch stag consist of transistors, relay and load. The transistor is use for both sensing voltage condition and a driving relay, the relay and load is the o/p of this project and it is here that the 'cutting out' action is carried out. This kind of system is not user friendly and not reliable. [5]

Another is construction of a high/low voltage cut-ff time delay, which say when a power is switched on the appliance gets connected only after the time delay of a fixed time, set in the protection device. In event of over voltage, when supply voltage is higher than upper trip-point voltage, pre-settable by a variable resistor, the sensing circuit sends appropriate signal to disconnect the load through its relay. Over voltage could result from phase-to-phase short circuit due to snapping and breaking of conductor lines etc. in event of under voltage, when sensing circuit senses an under voltage below the lower trip-point voltage, it sends the appropriate signal to its switching circuit to disconnect the load. Lower trip point voltage is also pre-settable by a variable resistor

. Under voltage may result from overloading of distribution transformers. Too expensive to Construct.[3]

2.3 Theoretical background

This project over voltage/ under voltage monitoring unit therefore seeks, to present a simple, reliable and effective form of protection against system abnormal condition of over voltage and under voltage to prolong the of the electrical and electronic equipment connected to it. This comprises of the following units

1. A LINE –MONITORING CONVERSION SUBSYSTEM
2. A CENTRAL PROCESSING UNIT
3. A NON VOLATILE MEMORY STORAGE UNIT
4. AN, AC ELECTROMECHNICAL SWTICH.
5. 4-7-SEGMENT DISPLAY UNIT

6. USER INTERFACE IN THE FORM OF SWITCHES

The local A.C main line voltage is rectified to a12V DC. the DC input voltage is fed into a ADC chip that convert the input voltage into a digital signal that is fed into a timer on the micro controller where it is converted to its absolute value when gated for a desired window period. The computed value of the line voltage is compared with the settings determined by the user. The line voltage is compare with both the lower and upper user presets. If the line voltage falls outside of the window, the protective feature is activated, i.e. the load is disconnected until the line input voltage falls back to within the window. A high power 10A electromechanical relay to provide A.C switching. The user setting is keyed into the system via push buttons and stored in non-volatile memory to provide a semi permanent storage. These settings we retained in the event of a power outage. The voltage level is displayed on the 7-segment display for visual interaction.

CHAPTER THREE

DESIGN AND CONSTRUCTION

3.0 Circuit Design and calculations

The operation of this project depends on the circuit and hence the design of the Circuit is in fact the focus of this project. The knowledge of how under/ over voltage protection with a cut-off is very necessary.

The electronic over voltage/under voltage cut-off unit was design around the following subunits

i- 12v 0.5A step-down transformer power supply

ii- 8 bits analog-digital converter

iii- 8 bits microcontroller

iv- 4-digital -7-segment display

v- User input keys.

We can now investigate the important of each stage: the components used and reason they are preferred. A good choice of components will yield good result.

3.1 POWER SUPPLY

A regulated +5V_{DC} supply voltage was requiring for system operation. This voltage was derived From a step-down transformer of 12 V_{RMS} secondary voltage of 0.5A rated current. The transformer Performed a secondary function asides the above mentioned. It also provided the ADC with the Required input voltage needed to compute the total supply AC mains voltage. The system power

Supply was derived from the transformer and a four full wave bridge rectifier as Shown below:

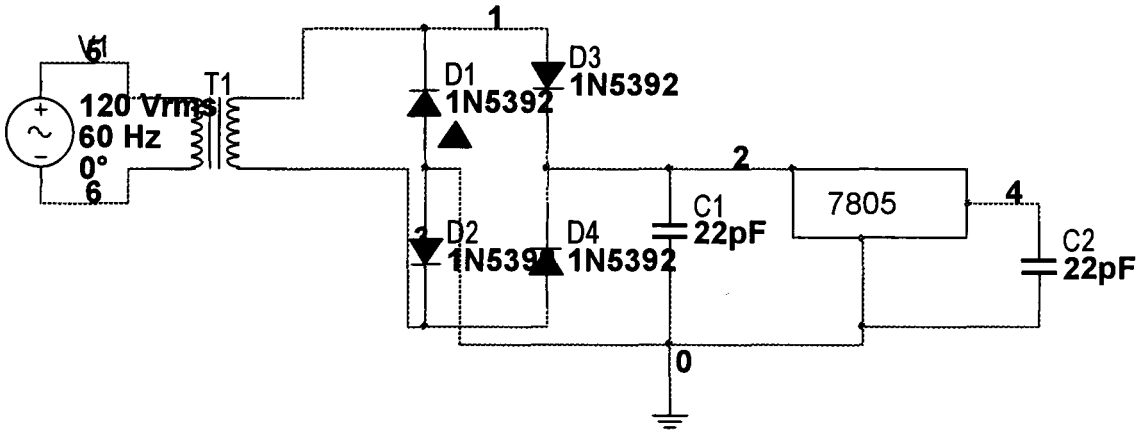


Figure 3.1: Circuit Diagram of the Power Supply

The 12V RMS secondary voltage was converted into a pulsating dc voltage of amplitude:

$$V_{DC\ PEAK} = V_{RMS}\sqrt{2} - 1.4 \quad (1)$$

$$V_{DC\ PEAK} = 12\sqrt{2} - 1.4 \approx 15.5V \quad (2)$$

This voltage was smoothened by a capacitor of value deduced from,

$$C = It/V = \text{maximum current}$$

$$t = 1/2F = 1/2 \times 50 = 25F \quad (3)$$

$$V = \text{maximum AC ripple voltage}$$

The load current was computed from the summation of the total current drawn by each sub-system:

- ADC 0804-5MA
- MICROCONTROLLER-15MA
- SEVEN-SEGMENT DISPLAY-300MA PEAK

TOTAL CURRENT = 320MA

The maximum ac ripple voltage was fixed at 1V peak

$$C = It / dv$$

(3a)

$$C = 0.32 \times 1/(2 \times 50), \quad 1V = O.OO32F = 3200UF$$

A value of 2200uF was used. The resulting smoothened dc voltage was fed into a 7805 regulator

To provide a regulated +5V dc voltage require by the logic sub-system.

3.2 8-BIT ANALOG TO DIGITAL CONVERTER

Since the ac live voltage to be deduced is an analogue quantity, a means of converting it into a digital value is required. A device capable of translating the analogue input voltage successive approximation register analogue to digital converter, the ADC0804 has the pin diagram and specification stated below:

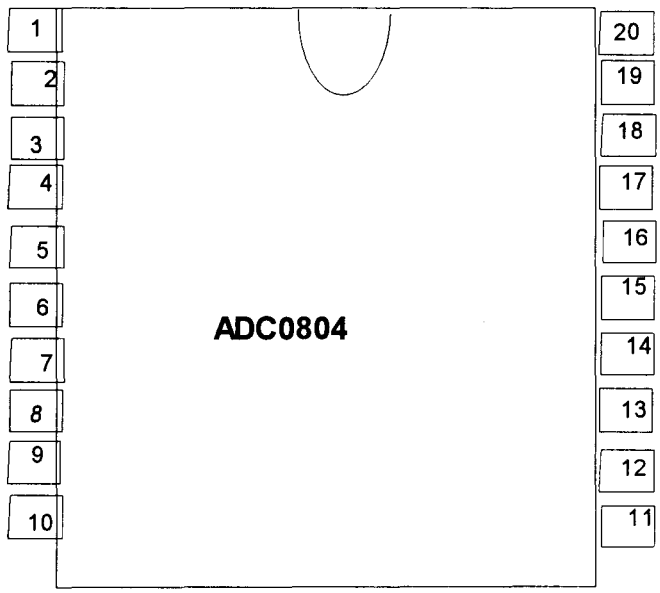


Fig. 3.2: (a) PIN diagram of ADC0804

Table 3.2: (a) Pin labels function

PIN NO	SYMBOL	INPUT/OUTPUT OR POWER	DESCRITION
1	$\overline{CS}$	INPUT	Chip select line from microcontroller control
2	RD	INPUT	Read line from microcontroller control
3	WR	INPUT	Write line from microcontroller control
4	CLK IN	INPUT	Clock
5	INTR	OUTPUT	Interrupt line goes to microcontroller interrupt input
6	$V_{in}(+)$	INPUT	Analog voltage (+ve input)
7	$V_{in}(-)$	INPUT	Analog voltage (-ve input)
8	AGND	POWER	Analog ground
10	DGND	POWER	Digital ground
11	DB_7	OUTPUT	USB data output
12	DB_6	OUTPUT	Data output
13	DB_5	OUTPUT	Data output
9	$V_{ref/2}$	INPUT	Alternate reference voltage
14	DB_4	OUTPUT	Data output
15	DB_3	OUTPUT	Data output

Table 3.2: (a) pin label functions (continuation)

16	DB_2	OUTPUT	Data output
17	DB_1	OUTPUT	Data output
18	DB_0	OUTPUT	LSB data output
19	CLKR	INPUT	Connect external resistor for cl
20	V_{cc} (or rep)	POWER	+ of 5v power supply and Primary reference voltage

The device was interfaced with the 89C51 controller over P.1,with P3.0 providing the ADC input.

The device was setup for a full scale input voltage of 2.56volts. The device was run off a cl source of value computed from:

$$F = (1/1.1RC)$$

(4)

R = resistance between pin 19 and 4,

Typically 10KΩ

C = Capacitance from pin 4 to ground.

The device has a maximum operating frequency of 1.1MHZ and a minimum of 100KHZ of about 500HZ was selected since the speed of the system is not terribly important. the dc input voltage has a zero bandwidth

The value of the capacitance require for 500KHZ operation was computed from the expression

$$500000 = \frac{1}{1.1 \text{ X } 10^4 \text{ X } C}$$

C = 1.8 PF

The ADC 0804 was fed from a divided by 10 attenuator on its input pin 6 as depicted below:

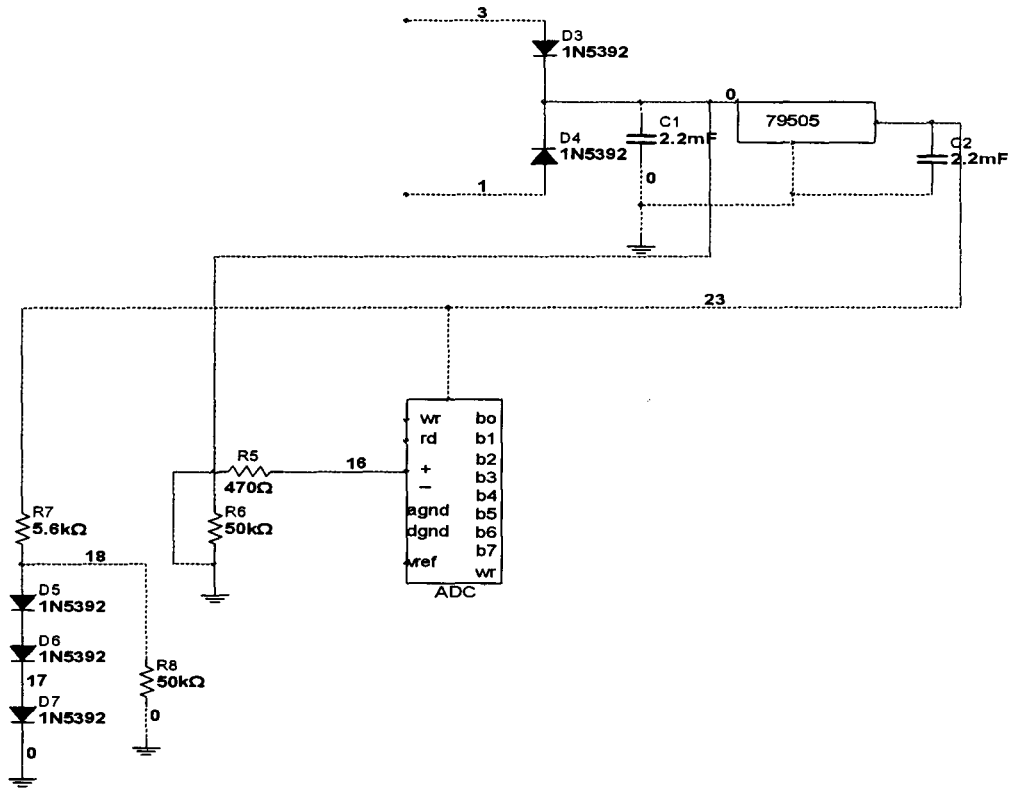


Figure 3.2: (b) ADC System Connection

The input divided by 10 attenuator was made from a 50KΩ resistance as illustrated below:

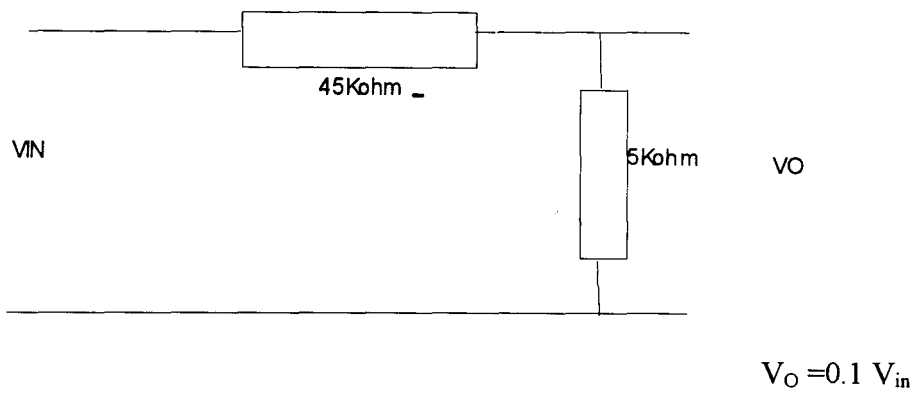


Figure 3.2(c): a 50KΩ Illustration.

The alternator was used to reduce the dc rectified voltage to within the input voltage ranges of the 0804

Conversion initiated by the controller via P3.0 corrected to the ADC WR input.

A conversion process is initiated by WR pulsing low, then high, conversion data is now ready typically 100us after the rising edge of WR.

3.3 8-BIT SYSTEM CONTROLLER:

An AT89C51 microcontroller was used for system control purposes. The device is a low power device with the specification shown below:

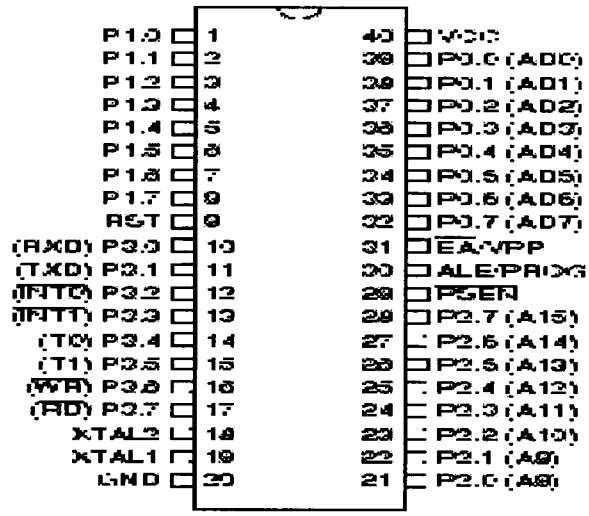


Figure 3.3: Ports Diagram of AT89C51 Microcontroller

Table 3.3: Port label function of 89C51 microcontroller

PORT NO	SYMBOL	INPUT/OUTPUT	DESCRITION
1.0-1.7	PORT 1.0-1.7	INPUT/OUTPUT	Port 1.0-1.7 is an 8 bits bi-directional I/O port with register pullups, it also receive the low-order address bytes during flash programming and verification
0.0-0.7	ADO-AD7	INPUT/OUTPUT	Port 0.0-0.7 is an 8 bits bi-directional I/O port with register pullups, it also receive the low-order address bytes during flash programming and verification

Table 3.3: Port label function of 89C51 microcontroller (cont'd).

2.0-2.7	A9-A15	INPUT/OUTPUT	Port 2.0-2.7 is an 8 bits bi-directional I/O port with register pull-ups, it also receive the low-order address bytes during flash programming and verification
3.0	RXD	INPUT/OUTPUT	(Serial Input Port)
3.1	TDX	INPUT/OUTPUT	(Serial Input Port)
3.2	INTRO	INPUT/OUTPUT	(External Interrupt 0)
3-.3	INTR 1	INPUT/OUTPUT	(External Interrupt 1)
3.4	T0	INPUT/OUTPUT	(Time 0 External Input)
3.5	T1	INPUT/OUTPUT	(Time 1 External Input)
3.6	WR	INPUT/OUTPUT	(External Data Memory Write Strobe)
3.7	RD	INPUT/OUTPUT	(External Data Memory read Strobe)
18	XTAL1	INPUT	Input to the inverting oscillator amplifier and input to the internal clock operating circuit
19	XTALI 2	OUTPUT	O/P from the inverting oscillator amplifier
29	PSEN		Program store enable is the read store to external program memory
3O	ALE/PROG	OUTPUT	Address latch enable output pulse for latching pulse for latching the low byte of the address during excess to external memory
31	EA/VPP		External access enable, EA must be strapped to GND in order to enable the device to fetch code from external program memory location starting at 0000H up FFFFH
20	GND		GROUND
9	RST	INPUT	A high on this pin for two machine cycle while the oscillation is running resets the device
40	V_{CC}		Supply voltage

The controller was used in the design to:

i Control the conversion process

ii Refresh the display

iii Write user preset value to and read from the connected non-volatile memory

iv Compute the value of the corresponding line voltage ac voltage from the digitized output of ADC

The microcontroller runs of an external crystal of 12MHz, yielding an instruction execution time of 1ns.was interfaced with other system device as shown in fig. 3.4

The control also provided load switching in the event of the ac live voltage violating the upper or lower voltage setting via a 6V, 10A relay. The microcontroller was used to read the status of three Input keys, providing a user interface for the system via which the user preset can be inputted.

3.4 LOAD SWITCH:

A dc relay was incorporated for load disconnection at over voltage or under voltage. The relay was controlled via a PNP transistor as diagramed below:

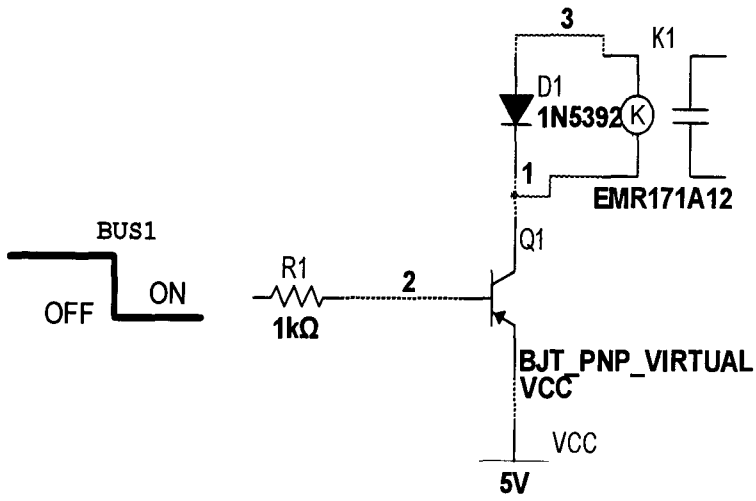


Figure 3.4: Load Switch

The relay had a coil resistance of 40KΩ at 6V, corresponding to a relay coil current of:

$6/400 = 15\text{mA}$

Thus, the collector current of the transistor is 15mA.

The PNP device used has a gain of typically 200

$$I_B = I_C/H_{FE} = 0.05/(2\times10^2) = 0.75\times10^{-4} = 75\text{mA}$$

$$R = (V_{CC}-V_{BE}-V_{ON}(89C510))/I_B = (5-0.7-0.02)/7.5\times[10]^{4} = (4.3/7.5)\times10^5$$

$\approx 56K\Omega$

To allow for overdrive and switching at all possible instances, a 1Kohm base resistance was used.

3.5 4-DIGIT VISUAL DISPLAY

A 4-digit 7-segment display was used incorporated for visual feedback. The display was multiplexed into Port 0, with individual digit drivers interfaced with P2. The display was

multiplexed to reduce the amount of wiring needed to covert to the controller. A multiplexed display always has all the common segments on the separated digits converted together and brought out a Single lead. To drive a multiplexed display, a means of controlling each display position was required.

A software routine was used to periodically refresh the display.

A typical display refresh loop is shown below. To display '1234' on a 4-digit multiplexed display;

i- Turn off all the drivers

ii- Place '1' on the common day= ta path

iii- Turn on the digit driver for digit 1

iv- Delay for persistence(some millisecond)

v- Turn off the digit driver for digit 1

vi- Repeat steps i-iv ,changing the data to zero display, and the related digit drive.

The display was interfaced with the controller as shown below,

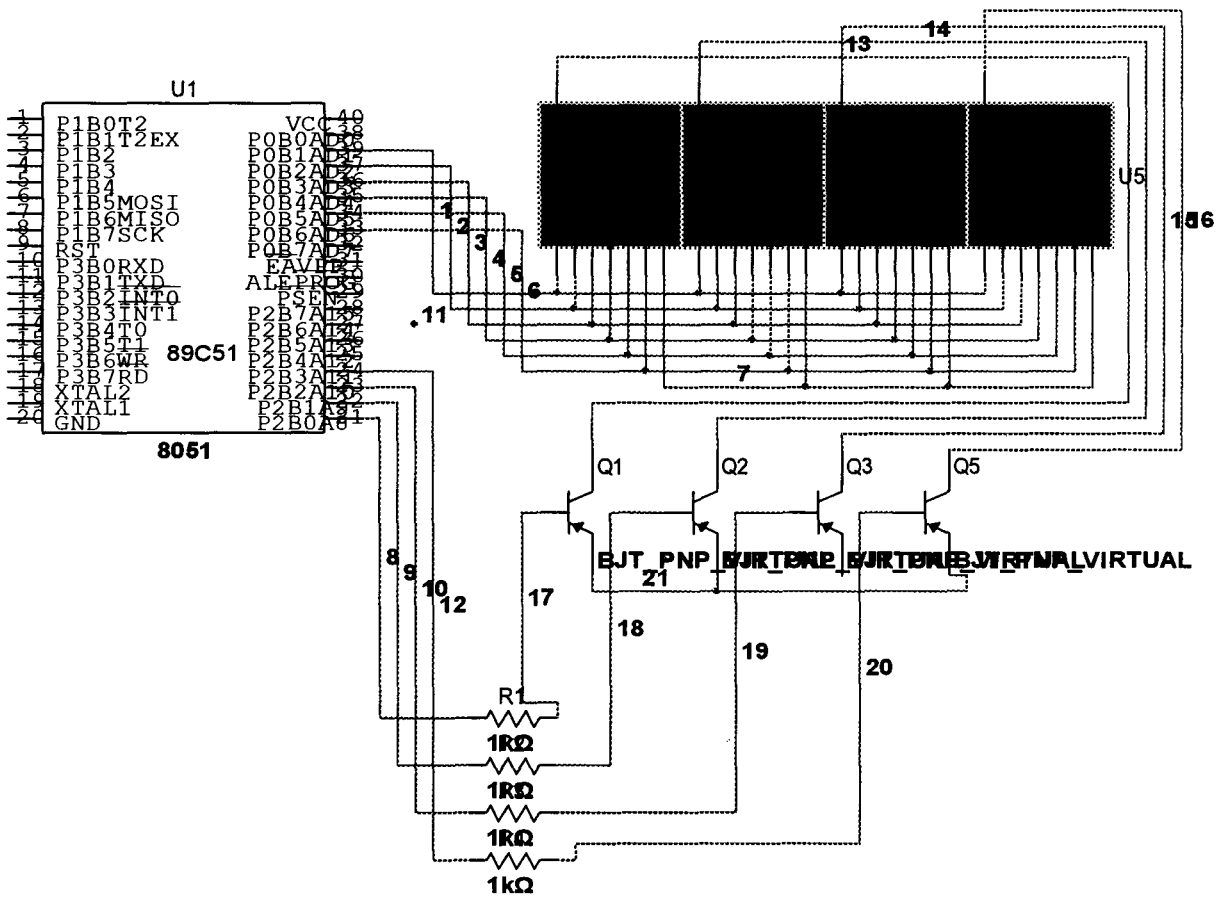


Figure 3.5: (a) 4- Digit Multiplexed Display

P0 was the common data path for current for segments a_0-a_5, P2.0-P2.3 provided the digit drivers Control lines.

Associated with each digit is a PNP driver (a_2-a_5) for switching ON/OFF.

In an n-digit multiplexed display, the pulse current required by each segment in I_F, where

n= no of digits

I_F=continuous LED forward current

typically 5mA-20mA. A value of 15mA was chosen as the continuous LED current yielding

a Peak segment current of

4×15mA =60mA

Each digit then drew 7×60mA =420mA peak current.

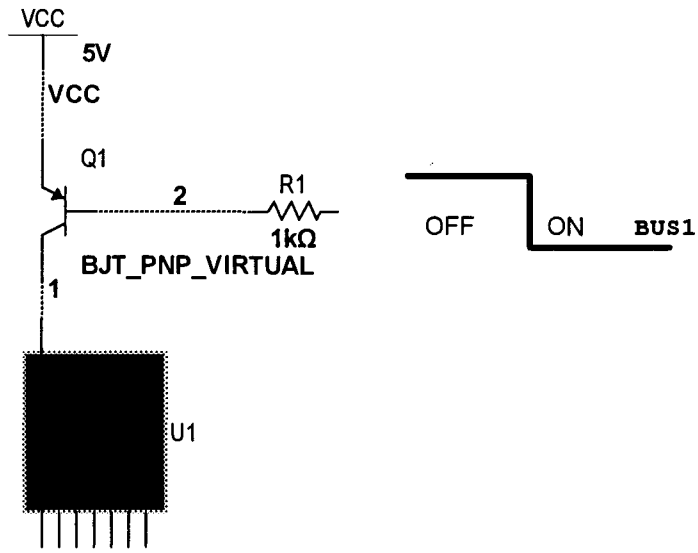


Figure 3.5: (b) The 7-Segment Display Switching

I_C = 420mA,

I_B =(420mA) /2×102 = 2.1×10^{-3}

$R_B = (V_{CC}- V_B V_{0L}$ (89C51))/I_B = (5-0.7-0.02)/0.0021 = 4-3/$2.1\times10^{-3} \approx 2K\Omega$

The resistance value was halved to 1KΩ for better display lighting in bright ambient light.

3.6 256-BYTE NON-VOLATILE MEMORY

A 256-byte I^2C E^2ROM device was incorporated in the unit to provide a non-volatile storage of the User upper/lower limits presets.The device is an 8-pin integrated circuit with the pin out shown below.

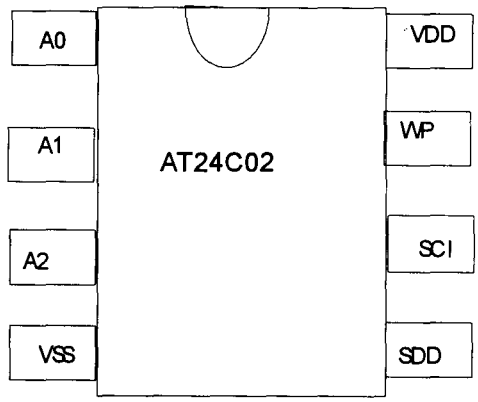


Figure 3.6: 24C02 EPPROM

The device was design on the I^2C bus architecture patented by Philips semiconductor.

The 24C02 memory has a communication interface that is not supported in hard wave on the 89C51.

Therefore software emulation was used to create the necessary control state.

Two memory locations were used to store the user preset voltage limits.

At power The memory was interfaced with the controller over P3.3 and P3.4 (SC1 and SDD respectively).

This was design on the I^2C bus architecture patented by Philips semi-conductor. The 24C02 memory has a communication interface that is not supported hard wave on the generic 89C51; therefore software emulation was used to create the necessary control states.

Two memory locations were used to store the up, the preset are loaded into two ROM locations where they can be manipulated as desired.

The manipulated value (when altered by the user charging the preset levels) are re-written to the Memory and retained even in the event of power outage.

This feature provides a "set and forget" convenience in that the setting only has to be made once and except the user decides to adjust the limits/remain that way.

3.7 AC LINE VOLTAGE COMPUTATION:

The controller used a looked-up table to compute the ac live level based on the expression given below.

A 240:12v transformer was used,

Vp: Vs =240:12v = 20:1

The RMS to peak scaling factor is root 2 rectified voltages is thus

$$(V_{AC} / 20) \sqrt{2} - 1.4 \qquad (5)$$

1.4 = The voltage drop in the bridge rectifier.

To reduce the input voltage into the ADC0804, +10 attenuator was used.

A reference voltage of 2,56v was used on the ADC, providing LSB output charge for every 10MV on the input pin.

The ADC output value is then related the ac main voltage by the expression.

$$V_{ADC} = [[(V_{AC} / 20) \sqrt{2}]] / 10$$

Since all other variables are know,

$$V_{ADC} = K V_{AC.}$$

Due to the lack of floating point mathematical capability in the controller, a program was used to Generate the AC live voltage for an ADC output of 0-255.These values were loaded into the controller's flash memory as a lookup table, each entry containing three bytes, corresponding to hundreds, tens, units data respectively.

The ADC output was used to index the table and load the appropriate values based on the pre-computed results. The three bytes are then displayed as the live voltage level.

Using this approach, a 98% array was observed between displayed voltage readings and the measured ac live voltage (using meters).The above method also eliminated the software overhead required for computing the ac live voltage from the expression $V_{ADC} = [[(V_{AC} / 20) \sqrt{2}]] / 10.$

A LED was used to provide an error indication if the ac live voltage violates either the upper or lower preset levels.

3.8 SOFTWARE SUBSYSTEM

A software control routine was loaded into the controller to define the system functionality.

The routine

Consists of modules that:

1. Initialized the system at power up.
2. A-D conversion.
3. Computed the voltage using the lookup table.
4. Compared the live voltage with the user presets'
5. Display the voltage level on the display.
6. Decodes the input keys interfaced for user inputs.
7. The software listing is included in the appendix for reference.

3.9 THE COMPLETE CIRCUILT

The complete circuit diagram of the operating principles of the project work is shown below.

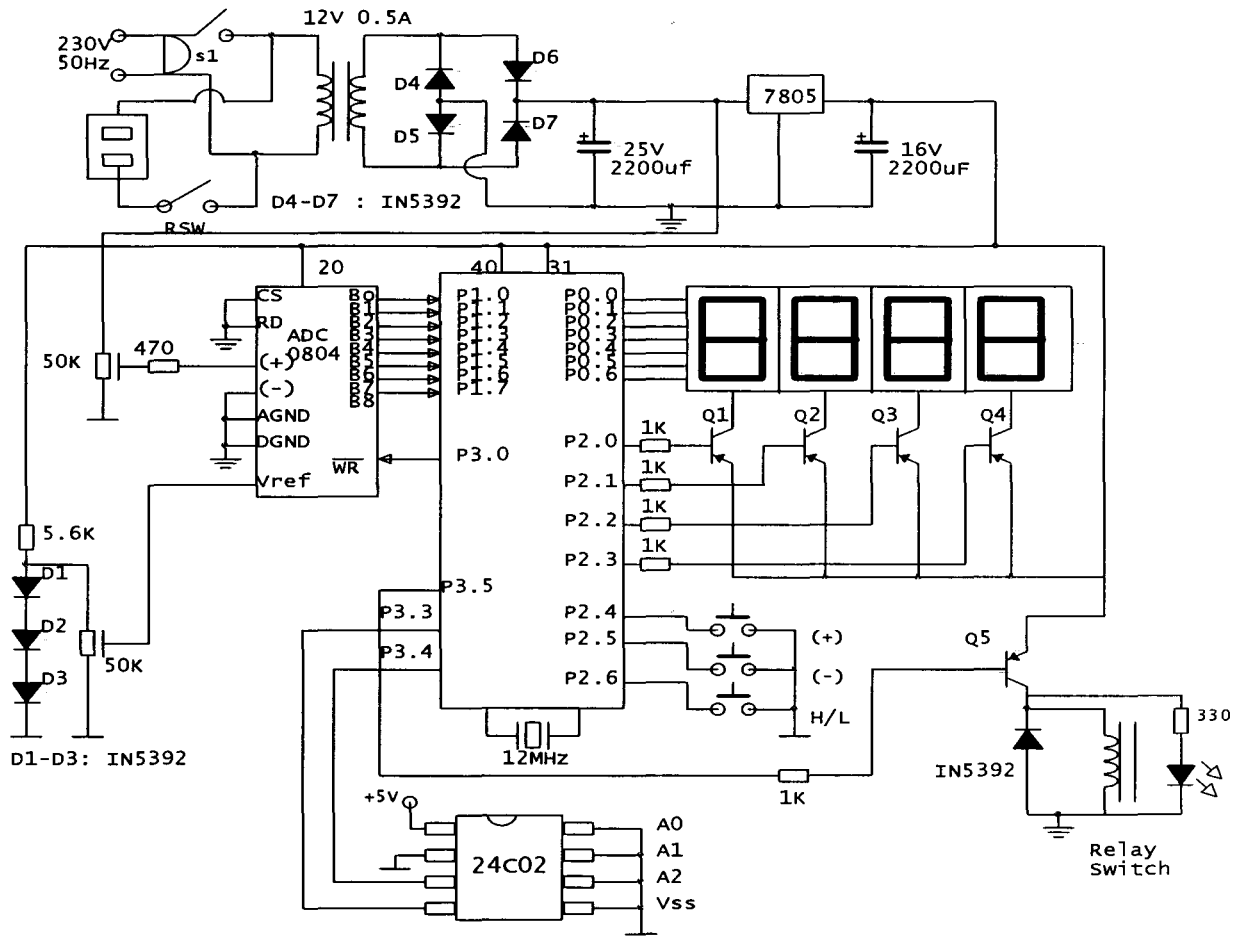


Fig. 3.9 The Complete Circuitry

CHAPTER FOUR

TEST, RESULT AND DISCUSSION

4.1 Test

Before the connection of all the components, they were tested using a multi-meter. i.e. test of continuity, rated voltage, current etc.

For final presentation and testing, the lamb dimmer was used to control the AC line voltage to test the cutout of the system in monitoring the over/under voltage.

4.2 Result

Under normal operating condition of the line voltage, the system is working, i.e. setting the upper and the lower limit of the voltage from the microcontroller pre-set buttons to the voltage from the line voltage and the voltage needed by appliance.

But if the line voltage is outside the pre-set ranges of voltage value, the electro-mechanical relay trips the system off through the PORT 2 of the microcontroller.

4.3 Discussion

This circuit was constructed using readily available semi-conductor devices such as a standard bi-polar power NPN transistor, a 89C51 microcontroller, 24C02 ROM, seven segment display, and ADC. These components were tested to ensure they were in good working condition. A potentiometer (light dimmer) was used to adjust the AC voltage. The output of the potentiometer was set to 180AC and then connected to the primary of the

transformer, and then the pre- set was set within and out of this AC to energize or de-energize the relay, a bulb as an indicator to check if the system is ON/OFF.

4.4 Summary

The table shows the relationship between the light dimmer and pre-set values of the circuit, i.e. if the pre-set value of the lower limit is 40V and higher value is 180V. The light dimmer was use to vary the line voltage.

Table 4.4 Showing input voltage against output state

DIMMER VOLTAGE O\P	CIRCUIT MODE
LESS THAN 40V	OFF
HIGHER THAN 180V	OFF
BETWEEN THE PRESETED VALUE	ON

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

When the AC is within the ranges of the pre-set voltage i.e. the switching device is switched ON i.e. the relay is energized and outsides this range the relay is de- energized

5.2 Recommendations

To increase reliability and survivability of appliances e.g. television, handset charger etc, this device should be manufactured along side home appliances as a guard.

REFERENCES

[1] Tasimu Ishaya Dodo, Design and construction of an Automatic voltage regulator for over/under voltage supply F.U.T Nigeria 2005, pp4. Unpublished

[2] Rowland .A, Okon, Design and construction of an under/over voltage control switch F.U.T, Minna, Nigeria. Unpublished

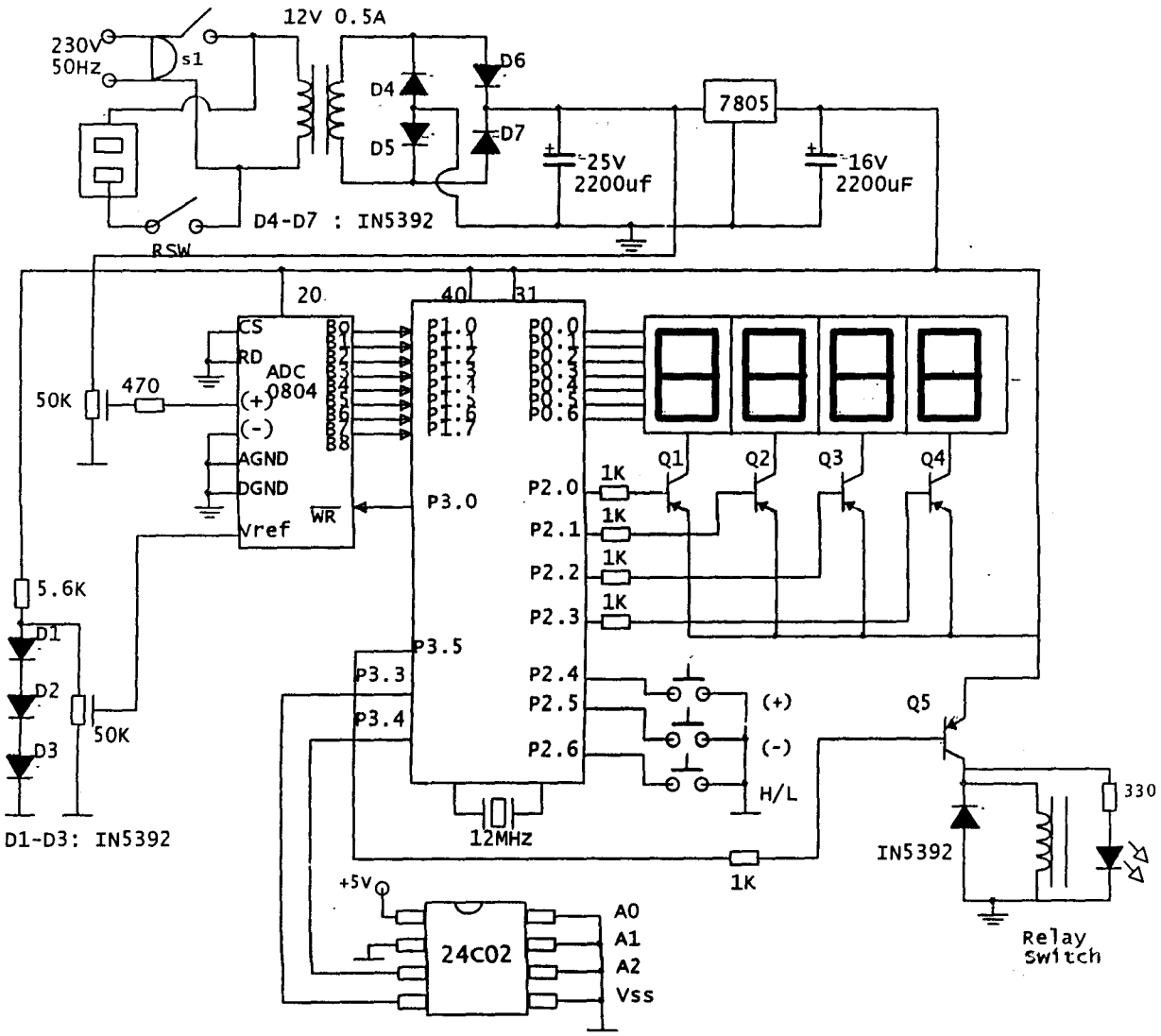
[3] Danjuma Undong Favour, Design and construction of a High/ Low voltage cut-off time delay pp4. Unpublished

[4] Electrical high voltage engineering By Engineer (Dr) Oria Usifo (1st Edition)

[5] Ounlola Moses Oluwatayo, Design and construction of an Improved over/under voltage cut-off device F.U.T Minna, Nigeria, 2005 Unpublished

APPENDIX A

The Complete Circuit diagram



APPENDIX B

```
INCLUDE 89c51.mc

;*********************************************************
h_bit_pattern EQU 10001001b
l_bit_pattern EQU 11000111b
mask_1 EQU 11111001b
mask_2 EQU 10100100b
e_mask EQU 10000110b
r_mask EQU 10101111b
;*************************************************
dATA_rEAD  DATA 08h
DAta_2_write DATA 09h
address DATA 0ah
slave_address EQU 0a0h
read_flag EQU 01h
write_flag EQU 00h
;*************************************
voltage_value DATA 0bh
dx_port EQU p2
data_port EQU p0
digit_1_dx BIT p2.0
digit_2_dx BIT p2.1
digit_3_dx BIT p2.2
digit_4_dx BIT p2.3
;*****************************************
relay_dx BIT p2.7
;*****************************************
adc_Select BIT p3.0
adc_write BIT p3.1
adc_intr BIT p3.2
;*************************************
dx_Ctrl DATA 0ch
data_0 DATA 0dh
data_1 DATA 0eh
data_2 DATA 0fh
data_3 DATA 10h
count DATA 11h
;*************************************

max_voltage EQU 135

;*****************************************************
clock BIT p3.3
data_out BIT p3.4
;*****************************************************
minus_key BIT p2.6
plus_key BIT p2.5
stack EQU 80
;*************************************************
```

```
adc_port EQU p1

;************************************************
upper_lower BIT p2.4
;************************************************
Over_voltage_nvram_address EQU 0
under_voltage_nvram_address EQU 1

;************************************
over_voltage DATA 14h
under_voltage DATA 15h

;*****************************************
scale EQU 2
;*****************************************
voltage_hi DATA 16h
voltage_lo DATA 17h
;*****************************************
temp_q data 18h
temp_r data 19h
;*****************************************
error_led bit p3.5
;*************************************************************************



org 0000h
start_up: CLR ea
          MOV sp,#stack
          ACALL sys_init

mainloop:  ACALL convert_voltage
           ACALL write_display
           ACALL compare_voltage
           ACALL scan_key
           SJMP mainloop
;*********************************************************888
convert_voltage: CLR adc_Write
                 NOP
                 SETB adc_Write
                 ACALL write_displaylong
                 ACALL write_displaylong
                 ACALL write_displaylong
                 ACALL write_displaylong
                 acall write_displaylong
                 acall write_displaylong
                 acall write_displaylong
                 acall write_displaylong
                 MOV voltage_Value, adc_port
                 mov dptr,#voltage_Table
                 mov a, voltage_value
                 mov b,#3
                 mul ab
```

```
                add a, dpl
                mov dpl,a
                clr a
                addc a, b
                addc a, dph
                mov dph, a
                clr a
                movc a,@a+dptr
                mov data_1,a
                clr a
                inc dptr
                movc a,@a+dptr
                mov data_2, a
                inc dptr
                clr a
                movc a,@a+dptr
                mov data_3, a
                mov a, data_1
                mov b,#100
                mul ab
                mov temp_q, b
                mov temp_r, a
                mov a, data_2
                mov b,#10
                mul ab
                add a, data_3
                add a, temp_r
                mov temp_r,a
                clr a
                addc a, temp_q
                mov temp_q, a
                mov a, temp_q
                clr c
                rrc a
                mov a, temp_r
                rrc a
                mov voltage_value, a
                mov data_0,#0ffh
                acall convert_7_seg
                RET
;*********************************************************8
process_voltage:    MOV data_0, #0ffh
                        ACALL scale_voltage
                        ACALL hex_2_bcd
                        ACALL convert_7_seg
                        RET
;*************************************************************************

scale_voltage:      MOV A, voltage_Value
                        MOV B,#scale
                        MUL ab
                        MOV voltage_hi ,B
                        MOV voltage_lo,A
                        RET
```

```
;*********************************************************************
hex_2_bcd:          MOV R1,voltage_hi
                    MOV R2,voltage_lo
                    CLR A
                    MOV R3,A
                MOV R4,A
                MOV R5,A
                MOV R6,A
                MOV R7,A
                    MOV B,#10D
                  MOV A,R2
                DIV AB
                MOV  R3,B          ;
                MOV  B,#10
                DIV  AB
                MOV  R4,B
                MOV  R5,A
                CJNE R1,#0H,HIGH_BYTE
                SJMP ENDD

HIGH_BYTE:
                    MOV  A,#6
                ADD  A,R3
                    MOV  B,#10
                    DIV  AB
                MOV  R3,B
                    ADD  A,#5
                ADD  A,R4
                    MOV  B,#10
                    DIV  AB
                MOV  R4,B
                    ADD  A,#2
                ADD  A,R5
                    MOV  B,#10
                    DIV  AB
                MOV  R5,B
                CJNE R6,#00D,ADD_IT
                SJMP CONTINUE
ADD_IT:
                  ADD A,R6
CONTINUE:
                MOV R6,A
                DJNZ R1,HIGH_BYTE
                MOV B, #10D
                MOV A,R6
                DIV AB
                MOV R6,B
                MOV R7,A
ENDD:               MOV data_1,R5
                    MOV data_2,R4
                    MOV data_3, R3
                    RET

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

```
convert_7_seg:      MOV DPTR,#xlate_Table
                    MOV A, data_1
                    MOVC A,@a+dptr
                    MOV data_1,A
                    MOV A, data_2
                    MOVC A,@a+dptr
                    MOV data_2,A
                    MOV A, data_3
                    MOVC A,@a+dptr
                    MOV data_3,A
                    RET

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

xlate_Table: DB
11000000b,11111001b,10100100b,10110000b,10011001b,10010010b,10000010b,11111000b,10000000b,1001
0000b

;*************************************************************
sys_init:           ACALL reset_Delay
                    ACALL reset_delay
                    ACALL reset_delay
                    ACALL reset_delay
                    ACALL reset_delay
                    MOV p0,#0ffh
                    MOV p1,#0ffh
                    MOV p2,#0ffh
                    MOV p3,#0ffh
                    CLR adc_select
                    MOV voltage_Value,#0
                    ACALL process_voltage
                    'ACALL write_presets
                    ACALL load_presets
                    RET

;*************************************************************
reset_Delay:        MOV R3,#0
reset_loop:         MOV R2,#0
                    DJNZ R2,$
                    DJNZ R3, reset_loop
                    RET
;;;*************************************************************
write_display:      MOV data_port, data_0
                    CLR digit_1_dx
                    ACALL delay_2_Show
                    SETB digit_1_Dx
                    MOV data_port, data_1
                    CLR digit_2_dx
                    ACALL delay_2_show
                    SETB digit_2_dx
                    MOV data_port, data_2
                    CLR digit_3_dx
                    ACALL delay_2_show
                    SETB digit_3_dx
```

```
                    MOV data_port, data_3
                    CLR digit_4_dx
                    ACALL delay_2_Show
                    SETB digit_4_dx
                    RET

;*************************************************************
load_presets:   MOV address,#over_voltage_nvram_address
                    ACALL read
                    JC show_error1
                    MOV over_voltage, data_read
                    INC address
                    ACALL read
                    JC show_error1
                    MOV under_voltage, data_read
                    RET
show_error1:
                    MOV data_0,#e_mask
                    MOV data_1, #r_mask
                    MOV data_2, #r_mask
                    MOV data_3, #mask_1
error1_loop:    ACALL write_display
                    SJMP error1_loop
;*************************************************************
write_presets:  MOV address,#over_voltage_nvram_address
                    MOV data_2_write, over_voltage
                    ACALL write
                    JC show_error2
                    INC address
                    MOV data_2_write, under_voltage
                    ACALL write
                    JC show_error2
                    RET

show_error2:    MOV data_0,#e_mask
                    MOV data_1, #r_mask
                    MOV data_2,#r_mask
                    MOV data_3,#mask_2
                    SJMP error1_loop
;*************************************************************
scan_key:           JB plus_key, key2
                    ACALL preset_temp_plus
                    JNB plus_key, scan_key
key2:               JB minus_key, key3
                    ACALL preset_temp_minus
                    JNB minus_key, key2
key3:               RET
;*************************************************************
;close relay during normal system oeroation. open at under or overvoltage



;compare_voltage:   MOV A, voltage_value;
;                   CJNE A, under_voltage, chk_again
```

```
;skip_Back:              CLR relay_dx
;
;                        SJMP chk_hi_preset
;chk_again:              JNC skip_back
;                        SETB relay_dx;
;chk_hi_preset:  MOV A, voltage_value
;                        CJNE A,over_voltage, chk_Again2
;s;kip_Back2:            CLR relay_dx
;exit_10:        RET
;
;
;chk_Again2:             JC skip_back2
;                        SETB relay_Dx
;                        RET
;*************************************************************
compare_voltage:MOV A, voltage_value;
                         CJNE A, under_voltage, chk_again
skip_Back:               CLR relay_dx
                         setb error_led
                         sjmp chk_hi_preset


chk_again:               JNC skip_back
                         SETB relay_dx;
                         clr error_led
                         ret
;*************************************************************
chk_hi_preset:  MOV A, voltage_value
                         CJNE A,over_voltage, chk_Again2
skip_Back2:              CLR relay_dx
                         setb error_led
exit_10:        RET


chk_Again2:              JC skip_back2
                         SETB relay_Dx
                         clr error_led
                         RET

;*****************************************************************************88

delay_2_Show:
show_Again:              MOV R4,#250
show_loop:               DJNZ R4, show_loop
                         RET
;*********************************************************************88
preset_temp_plus:
preset_lo_plus:  JB upper_lower,preset_hi_plus
                         PUSH voltage_value
                         MOV voltage_value, under_voltage
                         ACALL process_voltage
                         MOV data_0,#l_bit_pattern
                         ACALL write_displaylong
                         acall write_displaylong
```

```
                    MOV A, under_voltage
                    CJNE A,over_voltage, chk_Again3
skip_Back3:         DEC under_voltage
                    SJMP skip_this2
chk_Again3:         JNC skip_back3;
                    INC under_voltage
                    MOV A, under_voltage
                    XRL A, over_voltage
                    JNZ skip_this2
                    DEC under_voltage

skip_This2:         MOV voltage_value, under_voltage
                    ACALL process_voltage
re_Enter1:          MOV data_0, #l_bit_pattern
                    ACALL write_displaylong
                    acall write_displaylong
fall_out:       MOV address,#under_voltage_nvram_address
                    MOV data_2_write, under_voltage
                    ACALL write
                    POP voltage_value
                    RET

;*********************************************************************
preset_hi_plus: PUSH voltage_value
                    MOV voltage_value, over_voltage
                    ACALL process_voltage
                    MOV data_0,#h_bit_pattern
                    ACALL write_displaylong
                    acall write_displaylong
                    MOV A, over_voltage
                    XRL A, #max_voltage
                    JZ fall_out
                    INC over_voltage
re_enter2:          MOV voltage_value, over_voltage
                    ACALL process_voltage
                    MOV data_0, #h_bit_pattern
                    ACALL write_displaylong
                    acall write_displaylong
                    MOV address,#over_voltage_nvram_address
                    MOV data_2_write, over_voltage
                    ACALL write
                    POP voltage_value
                    RET

;*************************************************************************
preset_temp_minus:
preset_lo_minus: JB upper_lower,preset_hi_minus
                    PUSH voltage_value
                    MOV voltage_value, under_voltage
                    ACALL process_voltage
                    MOV data_0,#l_bit_pattern
                    CALL write_displaylong
                    acall write_displaylong
                    MOV A, under_voltage
```

```
                    JZ fall_out1
                    DEC under_voltage
                    MOV voltage_value, under_voltage
                    LJMP re_enter1
fall_out1:          JMP fall_out

;*********************************************************************
preset_hi_minus: PUSH voltage_value
                    MOV voltage_value, over_voltage
                    ACALL process_voltage
                    MOV data_0,#h_bit_pattern
          ACALL write_displaylong
          acall write_displaylong
                    MOV A, over_voltage
                    CLR C
                    SUBB A, under_voltage
                    DEC A
                    JNZ go_on_a
                    LJMP re_Enter2

go_on_a:            DEC over_voltage
                    LJMP re_Enter2



;*********************************************************************
write_displaylong:  MOV count,#60
long_loop:              ACALL write_display
                        DJNZ count, long_loop
                        RET
;*********************************************************************

write:              ACALL i2c_Start
                        MOV A, #slave_Address
                        ORL A, #write_flag
                        CALL write_byte
                        JC write_Abort
                        MOV A, address
                        CALL write_byte
                        JC write_Abort
                        MOV A, data_2_Write
                        CALL write_byte
                        JC write_Abort
                        CLR C
                        CALL i2c_Stop
                        call write_time_out
                        RET
write_abort:        CALL i2c_Stop
                        CALL write_time_out
                        RET
;*********************************************************

read:                   CALL i2c_Start
```

```
                        MOV A, #slave_Address
                        ORL A, #write_flag
                        CALL write_byte
                        JC read_Abort
                        MOV A, address
                        CALL write_byte
                        JC read_abort
                        CALL i2c_Start
                        MOV A, #slave_Address
                        ORL A, #read_flag
                        CALL write_byte
                        JC read_Abort
                        CALL read_byte
                        MOV data_Read, A
                        CALL no_Ack
                        CLR C
                        CALL I2C_STOP
                        RET
read_Abort:             CALL i2c_Stop
                        RET
;*********************************************************
i2c_start:              SETB data_out
                        SETB clock
                        CALL dly_7us
                        CLR data_out
                        LCALL dly_5us
                        CLR clock
                        CALL dly_7us
                        CLR C
                        RET
;*********************************************************
i2c_stop:               CLR data_out
                        CALL dly_5us
                        SETB clock
                        CALL dly_7us
                        SETB data_out
                        RET

;*********************************************************
write_byte:     MOV R2,#8
write_loop:     RLC A
                MOV data_out, C
                NOP
                NOP
                SETB clock
                CALL dly_7us
                CLR clock
                CALL dly_7us
                DJNZ R2,write_loop
                SETB data_out
                NOP
                NOP
                NOP
                SETB clock
```

```
                NOP
                NOP
                MOV C, data_out
                CLR clock
                RET
;*****************************************************
read_byte:      MOV R2,#8
                SETB data_out
read_loop:      NOP
                SETB clock
                CALL dly_7us
                SETB data_out
                NOP
                NOP
                MOV C, data_out
                RLC A
                NOP
                NOP
                CLR clock
                CALL dly_5us
                DJNZ R2,read_loop
                MOV data_Read, A
                RET
;*****************************************************
write_time_out: ACALL write_display
                        RET
;*****************************************************
no_Ack:         SETB data_out
                        NOP
                        NOP
                        SETB clock
                        NOP
                        NOP
                        NOP
                        CLR clock
                        RET
;*********************************************************************
dly_7us:        NOP
                        NOP
                        NOP
                        NOP
                        NOP
                        NOP
                        NOP
                        RET
;*********************************************************
dly_5us:        NOP
                        NOP
                        NOP
                        NOP
                        NOP
                        NOP
                        RET
;*********************************************************
```

```
voltage_table:
db 0,2,0
db 0,2,1
db 0,2,3
db 0,2,4
db 0,2,5
db 0,2,7
db 0,2,8
db 0,3,0
db 0,3,1
db 0,3,3
db 0,3,4
db 0,3,5
db 0,3,7
db 0,3,8
db 0,3,9
db 0,4,1
db 0,4,2
db 0,4,4
db 0,4,5
db 0,4,7
db 0,4,8
db 0,4,9
db 0,5,1
db 0,5,2
db 0,5,4
db 0,5,5
db 0,5,7
db 0,5,8
db 0,5,8
db 0,5,9
db 0,6,1
db 0,6,2
db 0,6,4
db 0,6,5
db 0,6,6
db 0,6,8
db 0,6,9
db 0,7,1
db 0,7,2
db 0,7,4
db 0,7,5
db 0,7,6
db 0,7,8
db 0,7,9
db 0,8,1
db 0,8,2
db 0,8,3
db 0,8,5
db 0,8,6
db 0,8,8
db 0,8,9
db 0,9,1
db 0,9,2
```

```
db 0,9,3
db 0,9,5
db 0,9,6
db 0,9,8
db 1,0,0
db 1,0,2
db 1,0,3
db 1,0,5
db 1,0,6
db 1,0,7
db 1,0,9
db 1,1,0
db 1,1,2
db 1,1,3
db 1,1,5
db 1,1,6
db 1,1,7
db 1,1,9
db 1,2,0
db 1,2,2
db 1,2,3
db 1,2,4
db 1,2,6
db 1,2,7
db 1,2,9
db 1,3,0
db 1,3,2
db 1,3,3
db 1,3,4
db 1,3,6
db 1,3,7
db 1,3,9
db 1,4,0
db 1,4,1
db 1,4,3
db 1,4,4
db 1,4,6
db 1,4,7
db 1,4,8
db 1,5,0
db 1,5,1
db 1,5,3
db 1,5,4
db 1,5,6
db 1,5,7
db 1,5,8
db 1,6,0
db 1,6,1
db 1,6,3
db 1,6,4
db 1,6,5
db 1,6,7
db 1,6,8
db 1,7,0
```

```
db 1,7,1
db 1,7,3
db 1,7,4
db 1,7,5
db 1,7,7
db 1,7,8
db 1,8,0
db 1,8,1
db 1,8,2
db 1,8,5
db 1,8,7
db 1,8,8
db 1,9,0
db 1,9,1
db 1,9,2
db 1,9,4
db 1,9,6
db 1,9,7
db 1,9,8
db 1,9,9
db 2,0,1
db 2,0,2
db 2,0,4
db 2,0,5
db 2,0,6
db 2,0,8
db 2,0,9
db 2,1,1
db 2,1,2
db 2,1,4
db 2,1,5
db 2,1,6
db 2,1,8
db 2,1,9
db 2,2,0
db 2,2,2
db 2,2,3
db 2,2,5
db 2,2,6
db 2,2,8
db 2,2,9
db 2,3,1
db 2,3,2
db 2,3,3
db 2,3,5
db 2,3,6
db 2,3,8
db 2,3,9
db 2,4,0
db 2,4,2
db 2,4,3
db 2,4,5
db 2,4,6
db 2,4,7
```

```
db 2,4,9
db 2,5,0
db 2,5,2
db 2,5,3
db 2,5,5
db 2,5,6
db 2,5,7
db 2,5,9
db 2,6,0
db 2,6,2
db 2,6,3
db 2,6,4
db 2,6,6
db 2,6,7
db 2,6,9
db 2,7,0
db 2,7,2
db 2,7,3
db 2,7,4
db 2,7,6
db 2,7,7
db 2,7,9
db 2,8,0
db 2,8,1
db 2,8,3
db 2,8,4
db 2,8,6
db 2,8,7
db 2,8,8
db 2,9,0
db 2,9,1
db 2,9,3
db 2,9,4
db 2,9,6
db 2,9,7
db 2,9,8
db 3,0,0
db 3,0,1
db 3,0,3
db 3,0,4
db 3,0,5
db 3,0,7
db 3,0,8
db 3,1,0
db 3,1,1
db 3,1,3
db 3,1,4
db 3,1,5
db 3,1,7
db 3,1,8
db 3,2,0
db 3,2,1
db 3,2,2
db 3,2,4
```

db 3,2,5
db 3,2,7
db 3,2,8
db 3,3,0
db 3,3,1
db 3,3,2
db 3,3,4
db 3,3,5
db 3,3,7
db 3,3,8
db 3,3,9
db 3,4,1
db 3,4,2
db 3,4,4
db 3,4,5
db 3,4,7
db 3,4,8
db 3,4,9
db 3,5,1
db 3,5,2
db 3,5,4
db 3,5,5
db 3,5,6
db 3,5,8
db 3,5,9
db 3,6,1
db 3,6,2
db 3,6,3
db 3,6,5
db 3,6,6
db 3,6,8
db 3,6,9
db 3,7,1
db 3,7,2
db 3,7,3
db 3,7,5
db 3,7,6
db 3,7,8
db 3,7,9
db 3,8,0