DESIGN AND CONSTRUCTION OF A DIGITAL CLOCK AND CALENDER WITH PRESET ALARM

BY

SALAWU LAKIBU
2003/15474EE

DEPARTMENT OF ELECRICAL AND COMPUTER ENGINEERING
SCHOOL OF ENGINEERING AND ENGINEERING TECHNOLOGY

FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA

NOVEMBER, 2008

DESIGN AND CONSTRUCTION OF A DIGITAL CLOCK AND CALENDER WITH PRESET ALARM

BY

SALAWU LAKIBU
2003/15474EE

A THESIS SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING, FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA.
IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF THE BACHELOR OF ENGINEERING (B. ENG) DEGREE IN ELECTRICAL AND COMPUTER ENGINEERING.

NOVEMBER, 2008

DEDICATION

I dedicate this thesis to Almighty GOD, who has granted me the knowledge to meet this academic challenge and who by his grace and mercy I have enjoyed enormous divine protection, provision and guidance.

I also dedicate it to my sweet mother, Hajia Aminatu and my beloved guardian, Hajia Yusuf.

DECLARATION

I, Salawu Lakibu, declare that this work was done by me and has never been presented elsewhere for the award of a decree.
I also hereby relinquish the copyright to the federal university of Technology, Minna.

Salawu Lakibu……
(Name of student)
……………………… 14/11/08
(Signature and date)

Mr. Usman Nuhu Galadima
(Name of supervisor)
……………………… 04/11/08
(Signature and date)

Prof. Y.A. Adediran
(Name of H.O.D)
………………………
(Signature and date)

………………………………
(Name of External Examiner)
………………………
(Signature and date)

ACKNOWLEDGEMENT

I gratefully acknowledge my deep indebtedness to Almighty God for his protection, provision and preservation from the beginning to this time. I greatly appreciate my able supervisor Mr.U.N. Galadimah for his honest supervision, courageous advice and constructive critism. I also appreciate my H.O.D., Prof.Y.A. Adediran and all others Electrical/computer departmental lecturers, who has seriously supported and encouraged me during my stay in the department.

In addition, I wish to express profound gratitude to my beloved parents, Mr. and Mrs. Aminatu Salawu, my in-laws, Dr. A.O. Jibril and Mal. A. Abdulhalim and my siblings, Fatimah, Halimat, and Suleiman Salawu. I also choose not to forget my level adviser, Mr. Suleiman zubir, for his relevant advice and all my class mates for there support.

ABSTRACT

Time keeping is an ancient phenomenon. Over the ages, time has been kept using different methods and instruments like Obliisks, water clock and so on. The design and construction of microcontroller-based digital clock and calendar as described in this project report is intended to produce a simple means keeping and displaying time and date in a better and more accurate way. The project is divided into four main different modules. Namely: Control unit, Display unit , Power unit and Keypad unit. The control unit has microcontroller that was programmed in assemble language to serve as the brain of the work. Displayed unit has six 7-segment combined to display the date and time in digital form. Power unit consist of dual power sources. The last segment, keypad, was built up with seven push buttons.

TABLE OF CONTENTS

LIST OF FIGURE

CHAPTER ONE

1.0 GENERAL INTRODUCTION

Time telling is an essential aspect of human life. Human beings generally live by time because all his activities are based on time passage. With brief and relevant explanation, this project report titled *"design and construction of a microcontroller-based digital clock, calendar and presetting alarming system"* has served as important tool to show the essence of time passage. The report gives an account of the work carried out during the various stages of the project [1]

The idea of the time passage that was calculated and recorded in the past was based on rotational characteristics of some useful astronomical or celestial bodies [2]. Celestial bodies – the Sun, Moon, Planets, and Stars – have provided us a reference for measuring the passage of time throughout our existence. In the past, measurement of time was based entirely on the motion of these celestial bodies through the sky [a.a]. Ancient civilization relied upon the apparent motion of these bodies through the sky to determine seasons [2]. The timekeeping in the ancient civilization of the pioneers as well as this modern civilization results into an entity called calendar. Calendar has its traces down to the measurements of seconds in the time-measurement [3].

According to the oxford advanced Learner's dictionary, calendar is a system by which time is divided into fixed periods, showing the beginning and end of a year. A calendar is a system of measuring time for the needs of civil life, by dividing time into days, weeks, months and years.

Calendar division was completely based on the movement of the Sun and the moon in the earth surface. In that respect, a day is the average time required for one complete rotation of the earth about its axis. In a similar vein, a month is the average time measurement of the Moon to revolve completely round the Sun in one circle. A year measurement was also not left out. One year measurement was based on one complete revolution of earth in circle around the Sun. such an account which results to a complete one year seasonal count is called tropical, solar or seasonal year.

To achieve this, the earliest human being used the position of the Sun in the sky to approximate the time of the day. If the Sun was on the horizon, it meant that either day was starting or ending. If the sun was high in the sky, it was about midday. However, this system of measuring the time had obvious flaws listed below:

(a) One could not tell the exact time due to the indistinct position of the Sun in the sky.

(b) It was difficult to see the Sun at night.

(c) The sun could be hard to see if there is heavy cloud coverage [3].

Around five (5) thousand years ago, Sumerian in the Tigris-Euphrates valley in today's Iraq had a calendar that helpfully divided the year into thirty (30) days to form month each. The day was divided into twelve (12) period (each hour corresponding to two (2) hours of our time), and divided these periods into 30 parts (each like 4 of our minutes) [4].

As I earlier said, the concept of a year was based on the earth's motion around the sun. The time from one fixed point, Such as a solstice or equinox, to the next is called a tropical year. Its length is currently 365.242190 days, but it varies. Around 1900 its length was 365.242196days and around 2100 it will be 365.242184 days [3]. Therefore, one complete solid solar year contains 365 days, 5 hours, 48 minutes and 46 seconds.

In the same respect, the concept of a month was based on the moon's motion around the earth. The length of the months is approximately one twelfth of a year (28 to 31 days) and is justified to fit the 12 months into a solar year [4]. In modern calendars, however, the number of days in a month is not based on the phase of the moon. To avoid any form of difficulties, our present civilization has adopted a 365 days solar calendar with a leap year occurring every fourth year.

Alongside with the calendar is a clock that keeps track of time in its own form. Clock on the other hand makes us to keep track of time in the form of hours, minutes and seconds, with a complete cycle of 24 hours making one (1) day.

These seconds, minutes, and hours are all meaningful units of calendar. Before any substantial changes is recorded in calendar, the changes of these units of calendar, which results in the calendar changes, must be kept track of.

Once it is a new day, then we have a complete event of a new calendar day [4].

The calendar and clock work together and serve the purpose of making us to be aware of time in the form of seconds, minutes, hours, days, weeks, months, and years as they pass by. The calendars and the clocks also serve as reminders. As the year past by, we have calendars and clock being used. The clocks that are in use up to day are of different format or classes. By the nature in which clocks are made of, they are classified broadly into analogue and digital clocks. The digital type which is adopted in this project was further classified, by the mode of their counting, into 12-hour clocking count and 24-hour clocking count.

The 24-hour counting clock or the 24-hour based timing system is the model used in this project. A complete circle of 24 hours counted makes one (1) day. The clock counts from 0 o' clock (the beginning of a new day) through 23 o'clock (which signify the ending of the day). Once it is a new day, we have a complete event of a new calendar day [5].

Most of clocks today make use of batteries and due to the advancement in the electronics technology, integrated circuit ranging from a simple counter IC up to as high as microcontrollers are now being embedded in the electronics design of digital calendar and clocks.

Due the fast growth in the electronics industry, a microcontroller with the part number AT89C51 was used in this project work.

In a way of enhancement and more reliability, two power sources were used to supply it energy. Power from 9V battery and that of power holding company of Nigeria (PHCN) were used.

Microcontroller–based digital clock with calendar and presetting alarming system is an electronic device or equipment that uses main component as microcontroller and quartz crystal for determine the frequency of the microcontroller [1].

The controller was used because of its compatibility with MCS-51th Series products which make it easy to be handled. The controller also has some useful characteristics as follow:

1. It has 2KB of reprogrammable flash memory.
2. It has fully static operation of 0Hz -to- 24MHz and
3. It is made with an endurance of 1000 write/erase cycle.

The usefulness of the third characteristics is that, if the software program burn onto the controller is not yielding the expected result, the program can be erase off the controller and reprogrammed it as many time as 1000 times [1].

The construction of the named project, consist of five (5) different electronic sub circuits unit. These five module or units of the main circuit are explained in detail in chapter three. The names of these sub circuits are listed bellow:

- Alarming circuit – unit
- Power supply circuit unit
- Display unit
- Keypad unit and
- Control circuit unit

Once the program is written and programmed onto the microcontroller to keep track of time, months, years, leap years and alarm output and also carry out all specified instructions, the device or the equipment would be able to inform and provide an individual with present date and time. The digital clock and calendar has the features of dates, real time clock chip, Months, hours, seconds, leap years, and alarm output. The alarm module consists of a sound device, using speaker or any other sound device. Also in inclusion is an external circuit that consist high frequency switching transistors used in controlling the inputs signal and interrupt to the controller. Multiplexed output signal is being taking through port zero (P0) of the controller into the display unit. The device would be able to serve the purpose of any automated calendar/clock since it is digital, software based, and using microcontroller. It properly combines the functionalities of the microcontroller and the efficiency and effectiveness of seven-segment display (7–segment display) to provide a flexible and effective electronic device.

1.1 AIMS AND OBJECTIVES

- The design and construction of the microcontroller-based digital clock calendar with presetting alarm is aimed at solving the problems of cost, complexity, portability and accuracy because the earlier clock/calendars were very large in size, complex in structure and expensive to build.
- The project is also aimed at providing a longer life span digital calendar/clock that has several years guaranty.
- The most interesting part of the objective is that the project is armed to realize a digital clock with calendar using microcontroller and acquaints one with the way chips are programmed and handled.\

1.2 METHODOLOGY

The project work titled "digital clock and calendar with presetting alarm" is physically divided into two. Namely:

Stand alone power pack and the main project work.

The procedures of achieving the whole package could be described as followed.

Right away in the standalone power pack, a main source of 230V AC from Power Holding Company of Nigeria (PHCN) was fed across 12Vdc, 300mA step-down transformer. The output of the transformer that was converted to pulsating dc voltage by a bridge rectifier of four diodes incorporated together was thereby smoothing by 2200µf electrolytic capacitors. An LM7809 voltage regulator regulated its input voltage to a steady 9Vdc output voltage. While in the main project work, the 9V dc regulated output voltages. each from PHCN and dc battery, were applied through an OR-gate form by combination of two diodes. The resultant output voltage was further smoothened by another set of 2200µf capacitors and regulated further to a 5V steady dc voltage by a voltage regulated IC called LM7805. This 5V final dc voltage was use to serve the main circuit and its sub circuits. Frequency was also considered as well as voltage. 12MHz quartz crystal was used to set the frequency of microcontroller. The 12MHz was internally divided by a factor of 12 to give 1MHz which mean that the microcontroller counts up to 1,000,000 times in one second. Using relevant registers within the controller, the 1000000 machine count was divided and stored in three registers to keep an account of second's passage. That is in every 1000000 count of machine cycles, the measurement in second is incremented by unit of one (1).

1.3 Scope of the Work

The project work titled "microcontroller-based digital clock and calendar" is physically divided into two. Namely:

Stand alone power pack and the main project work.

The prototype allows physically visualization of date and time in their digital form.

Right away in the standalone power pack, a main source of 230V AC from Power Holding Company of Nigeria (PHCN) powers the work. A 9V dc battery can also be used to serve the same purpose. While in the main project work, a 9V dc regulated output voltage from either PHCN or battery was applied to the microcontroller, the chief brain of the work. The signal generated by the controller, base on the programming source code written onto it. causes effect of physical visualization of date and time on the display unit.

1.4 Sources of Material

First and foremost, world wide wed is a large set of interlinked document, images and resources [1]. It is quite easy using word wide web to source for relevant information. So, internet was the source called upon. The second source of information for this prototype, are the school and departmental libraries. Move also, other relevant textbooks and useful application software consulted were listed in the page of references.

1.5 Project Layout

This thesis comprises five chapters. It is of charter one down to charter five. Chapter one contains General introduction of various field of study needed, project objectives, methodology, scope of the work and project outline. Chapter two dealt with literature review and theoretical background of every important component. Chapter three contains design and implementation of the thesis. Chapter four contains testing and result. Chapter five contains the conclusion and recommendation.

CHAPTER TWO
LITERATURE REVIEW/THEORETICAL BACKGROUND
2.1 Historical Background

Since the beginning of time, humans have been looking for better ways to answer the age-old question: "What time is it?" Throughout the last 4,500 years or so, methods used to tell time have evolved greatly, and are still improving today. From the most primitive sundials to the advanced [cesium fountain] (microcontroller based) clock of today, scientists have always worked to make clocks that are more accurate than the last generation of timekeeping devices [7].

The earliest humans used the position of the sun in the sky to approximate the time of day. If the sun was on the horizon, it meant that either the day was starting or ending. If the sun was high in the sky, it was about midday. However, this system had obvious flaws: you couldn't tell the exact time, you couldn't see the sun at night, and the sun could be hard to see if there was heavy cloud cover.

Then, in around 3500 B.C., the Egyptians began to construct huge pillars which served as primitive sundials. These huge pillars told time by casting shadows on the ground, which changed position depending on the time of day. This was a large improvement over simply looking at the sun, because it was much easier to tell the time by looking at a shadow on the ground than by hurting your eyes looking at the sun. By 1500 B.C., smaller, more refined sundials begin to appear, although, like previous sundials, they had limitations: they did not work at night or on cloudy days, and were not very accurate.

The next great advance in timekeeping occurred in about 3450 B.C. when a primitive hourglass was invented. This device was basically a bowl with a hole on the bottom.

Water dripped through the hole slowly as the day wore on, and grooves cut into the side of the bowl measured the passage of time. They had many advantages over sundials, because they would work on cloudy days and at night. However, this device required careful calibration, because the water poured out faster when the bowl was full since the water pressure was greater. Also, these devices would not work in freezing weather.

A huge advance occurred in the 1300's when mechanical clocks, which used weights or springs. began to appear. At first, they had no faces, and no hour or minute hands; rather, they struck a bell every hour. Later, clocks with hour, and then minute hands began to appear. These early mechanical clocks worked by using an escapement, a lever that pivoted and meshed with a toothed wheel at certain intervals. These controlled the movement, or "escape" of either the weights or the springs that were powering the clock, in order to regulate the speed at which the gears and wheels which measured the time turned.

In the 1400's, another important discovery in timekeeping was made: it was learned that coiled springs, which used small coiled springs unwinding at a speed controlled by an escapement, were able to move the hands on a clock as well as weights or springs of previous, larger clocks. This discovery made smaller clocks, and later watches, possible.

Then, in 1656, Christian Huygens invented the pendulum clock, which used weights and a swinging pendulum. These clocks were much more accurate than previous clocks, off by less than a minute a day, compared to the 15 minutes a day of earlier clocks. The bigger the pendulum, the more accurate the clock was.

In 1714, the British Parliament offered a cash reward to anyone who could invent a clock accurate enough for use in navigation at sea. Thousands of sailors died because they were unable to find their exact position, because the exact time was needed to find longitude, and pendulum clocks would not work at sea. For every minute lost by a clock, it meant that there would be a navigational error of 15 miles, and sailors died because they were lost or smashed against rocks as they were unable to figure out their exact position. Then, in 1761, after 4 attempts, John Harrison finally succeeded at inventing a small clock accurate enough to use for navigation at sea. This tiny pocket watch lost only 5 seconds in 6 and ½ weeks.

During the mid to late 1800's, many countries saw the need to create standard time zones so that everyone could agree on the time and nations could work more efficiently. In 1852, Great Britain implemented a telegraph network that transmitted "Greenwich Mean Time" (GMT), so the whole country would be running on exactly the same time. In 1884, delegates from many countries met and agreed on worldwide time zones.

At the dawn of the 20th century, only women wore wristwatches. No self-respecting "real man" would wear one. However, in the First World War, soldiers wore wristwatches because taking out a pocket watch to check the time was difficult or impossible in battle. After the war was over, it was considered "socially acceptable" to wear wrist watches and they became popular. The next great advancement in timekeeping was in 1967, when the atomic clock, which used the oscillations of cesium-133 atoms to tell time, was invented. This clock had an error ratio of 1 second for every 1.4 million years. Recently, in 1999, scientists developed the cesium fountain atomic clock, which is off by only one second every 20 million years. This clock is the most accurate in the world.

Throughout the last 5,000 years, many advances have been made in the field of timekeeping. Timekeeping techniques have continuously evolved, and will keep evolving in the future. Perhaps one day, humans will invent a clock that accomplishes that long sought after, seemingly unreachable goal: a timekeeping device which is 100% accurate and never gains or loses a second, not even in billions of years. Also, as time progresses, everyday clock will gain more features, such as the ability to automatically adjust for daylight savings time, and the ability to synchronize with atomic clocks through radio waves. Watches will gain new features as well, such as integrated radios and displays that show altitude, temperature, and heart rate. Aside from that, who knows what the future will bring?

2.1.1 Calendar

The common theme of calendar making is the desire to organize units of time to satisfy the needs and preoccupations of society. In addition to serving practical purposes, the process of organization provides a sense of understanding and controlling time itself [9].

These calendars serve as a link between mankind and the universe. It is little wonder that calendars have held a sacred status and have served as a source of social order and cultural identity. Calendars have provided the basis for planning agricultural, hunting and migration cycles for maintaining cycles of religious and civil events.

According to a recent estimate by Fraser in 1987 that there are about forty calendars used in the world today [9]. This chapter would only discuss three of them. The emphasis will be placed on their function rather than culture.

2.1.1.0 The Gregorian calendar

The Gregorian calendar today serves as an international standard for civil use. As part of its usefulness, it regulates the ceremonial cycle of the Roman Catholic and protestant Churches. In fact, its original purpose was ecclesiastical. Although a variety of other calendars are in use today, they are restricted to a particular religion or culture.

The Gregorian calendar resulted from a perceived need to reform the method of calculating dates of Easter. Under the Julian calendar the dating of Easter had become standardized, using March 21 as the date of the equinox and the metonic cycle as the basis for calculating lunar phases [10].

By the thirteenth (13^{th}) century, it was realized that the true equinox had regressed from March 21 (its supposed date at the time of the council of Nicea) to a date earlier in the month [10]. As a result, Easter was drifting away from its springtime position and was losing its relation with the Jewish Passover.

Over the next four centuries, scholars debated the "correct" time for celebrating Easter and the means of regulating this time calendrically. The Church made intermittent attempts to solve the Easter question, without reaching a consensus [10].

By the sixteenth (16^{th}) century the equinox had shifted by ten days, and astronomical New moons were occurring four days before ecclesiastical New Moons. At the request of council of Trent, Pope Pius V introduced a new Breviary in 1568 and Missal in 1570, both of which included adjustments to the lunar table and the leap-yea system. Pope Gregory xiii, who succeeded pope Pius in 1572, soon convened a commission to consider reform of the calendar, since he considered his predecessor's measures inadequate.

The recommendations of Pope Gregory's calendar commission were instituted by the papal bull "inter Gravissimus" signed on 1582 February 24. Ten days were deleted from the calendar, so that 1582 October 4 was followed by 1582 October 15, thereby causing the vernal equinox of 1583 and subsequent years to occur about March 21. And a new table of New Moons and full Moons was introduced for determining the date of Easter [10].

The finally refined modern table adopted for Gregorian calendar in today life to aid in taking note of date and time passage is shown in table 2.1

Table 2.1: Days per each month in Gregorian calendar

Months	**Number of day**
JANUARY	31
FEBUARY	28 or 29
MATCH	31
APRIL	30
MAY	31
JUNE	30
JULY	31
AUGUST	31
SEPTEMBER	30
OCTOBER	31
NOVEMBER	30
DECEMBER	31

2.1.1.1 The Islamic Calendar

The Islamic calendar is a purely lunar calendar in which months correspond to the lunar phase cycle. As a result, the cycle of twelve lunar months regresses through the seasons over a period of about 33 years. For religious purposes, Muslims begin the months with the first visibility of the lunar crescent after conjunction. For civil purposes, a tabulated calendar that approximates the lunar phase cycle is often used.

The seven-day week is observed with each day beginning at sunset. Number, with day 1 (one) beginning at sunset on Saturday and ending as sunset on Sunday, specifies weekdays. Day 5, which is called Juma'a is the day for congregational prayer.

Unlike the Sabbath days of the Christians and Jews, however, Juma'a is not a day of rest. Juma'a begins at sunset on Thursday and ends at sunset on Friday [11].

The form of the Islamic calendar, as a lunar calendar, was laid down by the Prophet Muhammad in the Qur'an (Sur ix, verse 36 – 37) and in his Sermon at the farewell pilgrimage. This was a departure from the lunisolar calendar commonly used in the Arab world, in which months were based on first sighting of the lunar crescent, but an intercalary month was added as deemed necessary. The Islamic calendar starts with the first month called "Muharram" and ends with the last month named "Zhul Haj" [11].

2.1.1.2 The Chinese calendar

This is one of the special calendars with some interesting features in the world. The Chinese calendar is a lunisolar calendar based on calculations of the position of the sun and moon. Months of 29 or 30 days begin on days at astronomical New Moons, with an intercalary month being added every two or three years.

Since the calendar is based on the true position of the sun and moon, the accuracy of the calendar depends on the accuracy of the astronomical theories and calculations [12].

Although the Gregorian calendar is used in the People's Republic of China for administrative purposes, the traditional Chinese calendar is used for setting traditional festivals and for timing agricultural activities in the countryside.

In China, the calendar was a sacred document, sponsored and propagated by the reigning monarch. For more than two millennia, a Bureau of Astronomy made astronomical observations, calculated astronomical events such as eclipses, prepared astrological predictions, and maintained the calendar (Needham in 1959). Analysis of surviving astronomical records inscribed on oracle bones reveals a Chinese lunisolar calendar, with intercalation of lunar months, dating back to the Shang dynasty of the fourteenth century B.C. various intercalation schemes were developed for the early calendars, including the nineteen-year and 76-year lunar phase cycles that came to be known in the west as the Metonic cycle and callipic cycle [12].

From the earliest records the beginning of the year occurred at a New Moon near the winter solstice. The choice of month for beginning the civil year varied with time and place, however, in the late second century B.C., a calendar reform established the practice, which continues today, of requiring the winter solstice to occur in month 11. This reform also introduced the intercalation system in which dates of New Moons are compared with the 24 solar terms. However, calculations were based on the mean motions resulting from the cyclic relationship. Inequalities in the Moon's motions were incorporated as early as the seventh century A.D. (Sivin, 1969), but the sun's mean longitude was used for calculating the solar terms until 1644 (Liu and Stephenson, in press).

2.2 THEORETICAL BACKGROUND

2.2.1 Power Supply

In order to avoid the frequency power interruption of PHCN, dual power sources wee made available for this project work. The project was powered by a 9V alkaline battery and 220 / 240Va.c of the PHCN main source.

The source comprises

(i) Step – down transformer

(ii) Bridge rectifier

(iii) Electrolytic and Non – electrolytic capacitor

(iv) Regulators

(v) IN5392 diode and

(vi) 9V alkaline battery.

2.2.1.0 Step – Down Transformer

Transformers are electric device that are used to step-up or step-down AC current and voltages Transformer is shown in fig 2.1. This device uses two electromagnetic coils to transform or change AC – voltage levels. It's primary and secondary coils are separately wound around an iron core. An alternating input current at the primary coil produces a magnetic field that continually switches ON and OFF.

The core transfers this field to the secondary coil where it induces an output current. The degree of change in voltage depends on the rating of turn in the primary and secondary coil [ad].

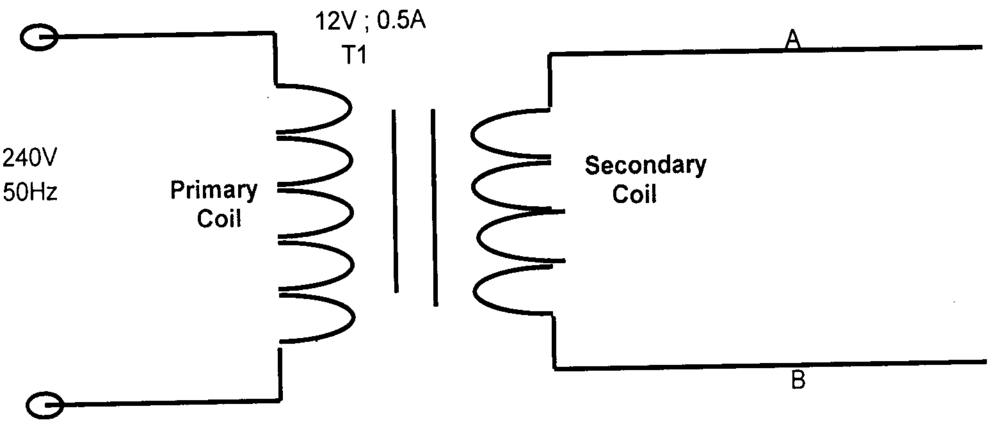

Fig 2.1 Step – Down Transformer

Voltage transformation ration (K) is $E_2/E_1 = N_2/N_1 = K$ (1)

Also $I_2/I_1 = V_1/V_2 = 1/K$ (2)

2.2.1.1 Bride Rectifier

One typical full – wave rectifier is the bride rectifier circuit. It can be used in transformer – type power supplies as well as line – operated power supplies.

The circuit uses four diodes to obtain full – wave rectification. The schematic for a full – wav, bridge rectifier circuit and its wave form is shown in figure 2.2

During the positive half – cycle, the anodes D_2 and D_3 are forward biased while D_1 and D_4 are reversed biased. Electron flow though D_2 and D_3 only. During the negative half – cycle, the polarity of the AC line voltage are reversed [13].

As a result, diodes D_1 and D_4 become forward biased and conduct electrons. Current flows through the load in the same direction, in any of the half – cycle, producing a full – wave pulsating DC as in the figure 2.2 waveform.

The DC voltage is given by $V_{DC} = V_{rms}\sqrt{2} - 1.4$

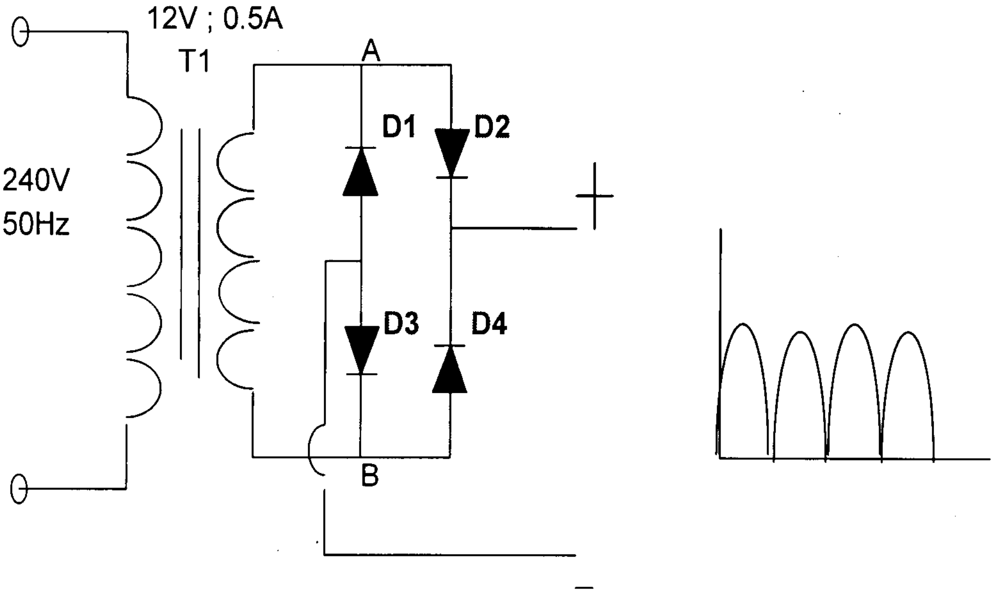

Fig 2.2 Bridge Rectifier Connected To Transformer And Its Waveform

2.2.1.2 Capacitor

Capacitance (C) is the ability of an electric circuit or component to store electric energy by means of an electrostatic field. The capacitor is an electrical electronic component specially designed for this purpose. Basically, a capacitor consists of two metal plates (conductors) placed near each other and separated by an insulating material called the dielectric. The dielectric can be air or any non conducting material such as paper, mica, or ceramic. The number of electrons that a capacitor can store for a given applied voltage is a measured of its capacitance. It is measured in farad (f). It could also be in microfarad, nanofarad,[13] etc.

The voltage rating of a capacitor indicates the maximum voltage that can be safely applied to its plates.

$$Q = CV$$

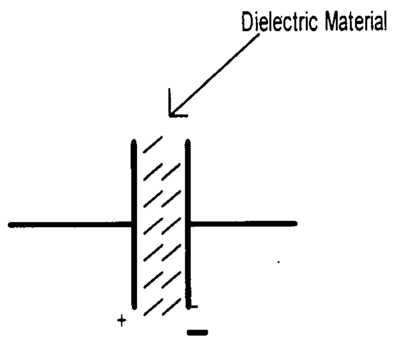

Fig 2.3 Symbolic Diagram of Capacitor

2.2.1.3 Regulator

An output of a power supply is usually decreased when a load is connected to it. That is the power supply is unregulated. Unregulated DC power – supply circuits do not deliver pure DC output. Also, the DC output voltage changes with input line – voltage changes and changing load conditions. [13]

A circuit that essentially eliminates variation in output power – supply voltage under conditions of changing input or changing load is called a voltage regulator. When a voltage regulator is connected to a rectifier circuit, the result is a regulated power supply. Figure 2.4 shows voltage regulator

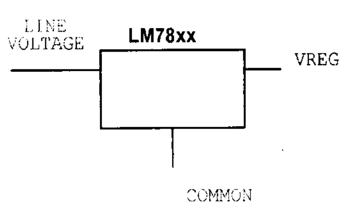

Fig 2.4 Symbolic Diagram of LM78xx Series Voltage regulator

2.2.1.4 Diode

Diode is a semiconductor material or device that ideally has Zero resistance when it is reversed biased.

Popular cylinder – shaped diodes use a dark band for lead identification. The lead end nearest to the band is the cathode, and the lead on the opposite end is the anode. Diode normally conducts fully in one direction. That is current flow from anode terminal to cathode terminal. The schematic symbol of diode is shown the figure 2.5.

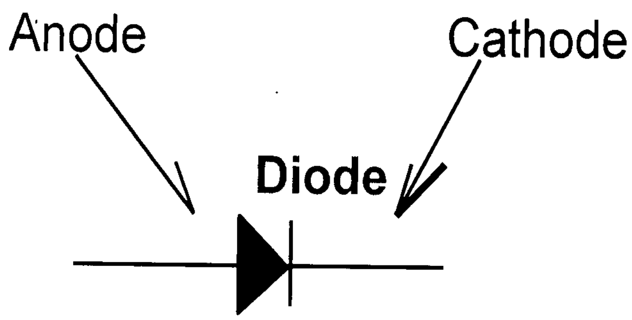

Fig 2.5 Symbolic Diagram of Diode

2.2.1.5 Battery

A dc source used in this project work is a 9V alkaline batter which serves as a back – up power supply for an unexpected power interruption of PHCN. The battery will immediately come into operation as soon as there is power failure.

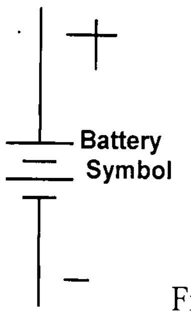

Fig 2.6 Symbolic diagram of battery

CHAPTER THREE

3.0 DESIGN AND IMPLEMENTATION

The design consists of AT89C51 controller programmed in low level language for the control of design. Quartz crystal was used to generate clock pulse for the control of the microcontroller to execute its instruction, while the output is displayed through 6 (six) of 7 –segment display (LED). The whole project is powered by a 9 volt alkaline battery which was regulated to 5V by a 5 volt voltage regulator integrated circuit, normally known as (LM 7805). Keypad of seven (7) buttons is used to control and update the time and date settings.

The design block diagram is shown in fig 3.1.

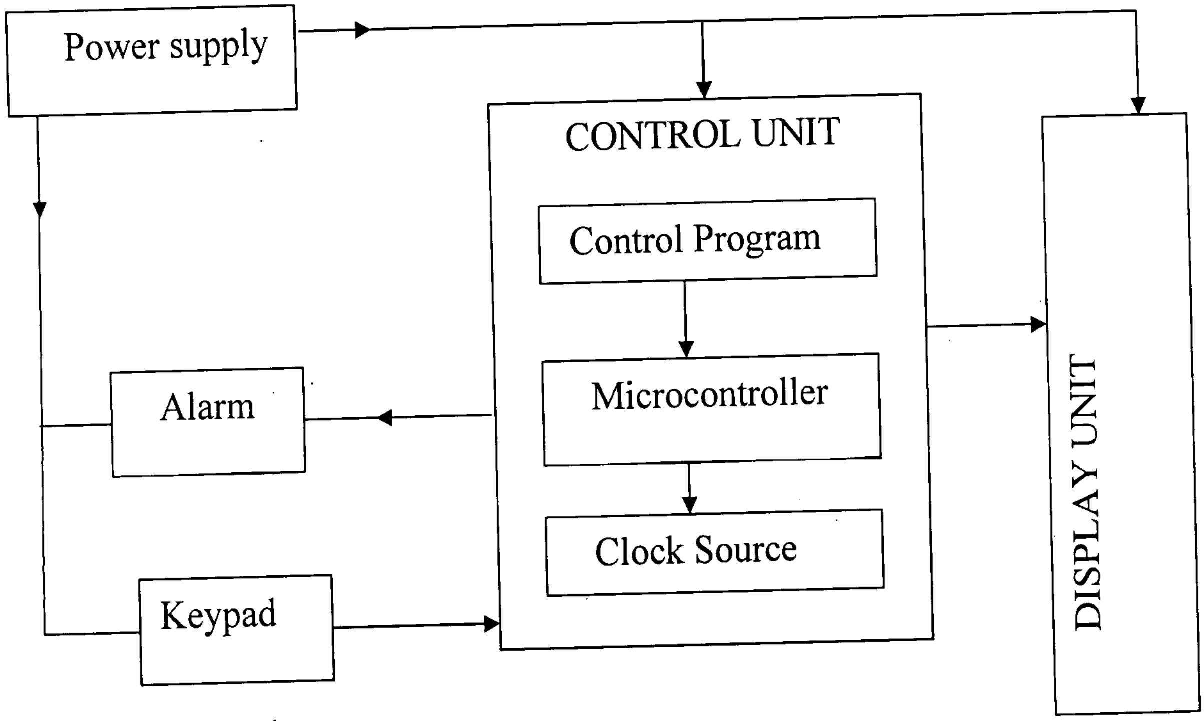

Fig 3.1 Block diagram of Microcontroller- based Digital clock and calendar

3.1. Control unit

This is the heart of the whole design, which controls all other part of the project. It consists mainly of:

(i) Microcontroller

(ii) Control program and

(iii) Clock source.

3.1.1 MICROCONTROLLER

3.1.1.1 Description of Microcontroller

Microcontroller or programmable Logic controller is a single computer chip (integrated circuit) that executes a user program. Its functions are determined by the program stored within it. The task of a microcontroller involves interlocking, counting, sequencing, timing and monitoring [1].

It was used here to monitor the operations of digital clock and calendar. The AT89C51 microcontroller is a low-power, high- performance CMOS, 8- bit- microcomputer with 8KB of Flash erasable and programmable read only memory (EPROM). The device is manufactured using Atmel's high density non-volatile memory technology and is compatible with the industry standard MCS-51^{tm} instruction set and pin out. The on-chip Flash allows the program memory to be reprogrammed which could also be achieved by the use of conventional nonvolatile memory programmer. It has Pin-out configuration as shown bellow in the figure 3.2.

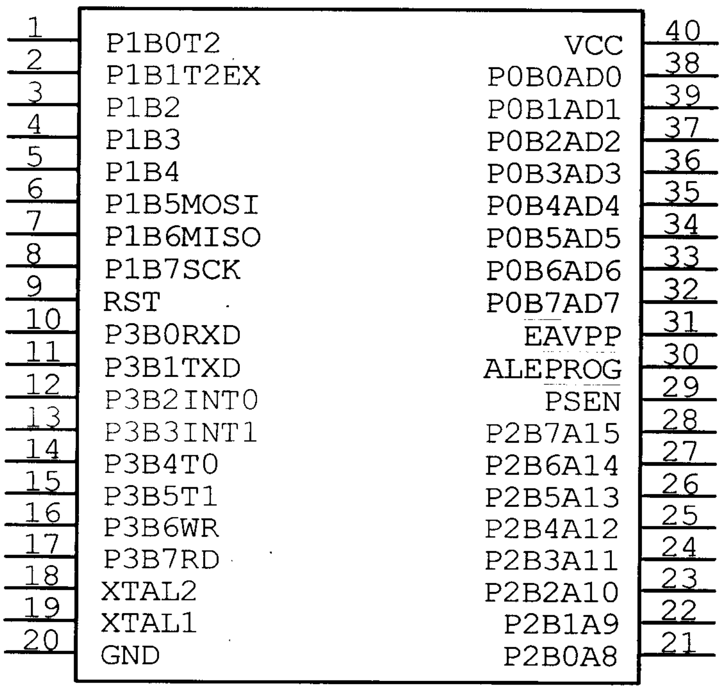

FIG 3.2 AT89C51 Pin-Out

The device is manufactured using Atmel's high-density non-volatile memory technology and is compatible with the industry –standard 8051 and 80152 instructions set and pin out configuration. Figure 3.3 shows the internal block diagram of the controller.

The Atmel AT89C51 is a powerful microcomputer which provides a highly- flexible and cost- effective solution to many control application. It has the following features:

(i) Compatible with MCS – 51™ products

(ii) 8k Bytes of in-system Reprogrammable Flash Memory

(iii) Endurance: 1,000 Write/Erase cycles

(iv) Fully static operation: 0Hz to 24MHz

(v) 32 programmable input and output lines

(vi) Three 16-bit Timer/counters

(vii) Eight interrupt sources

(viii) Programmable serial channel

(ix) Low – power idle and power – down Modes

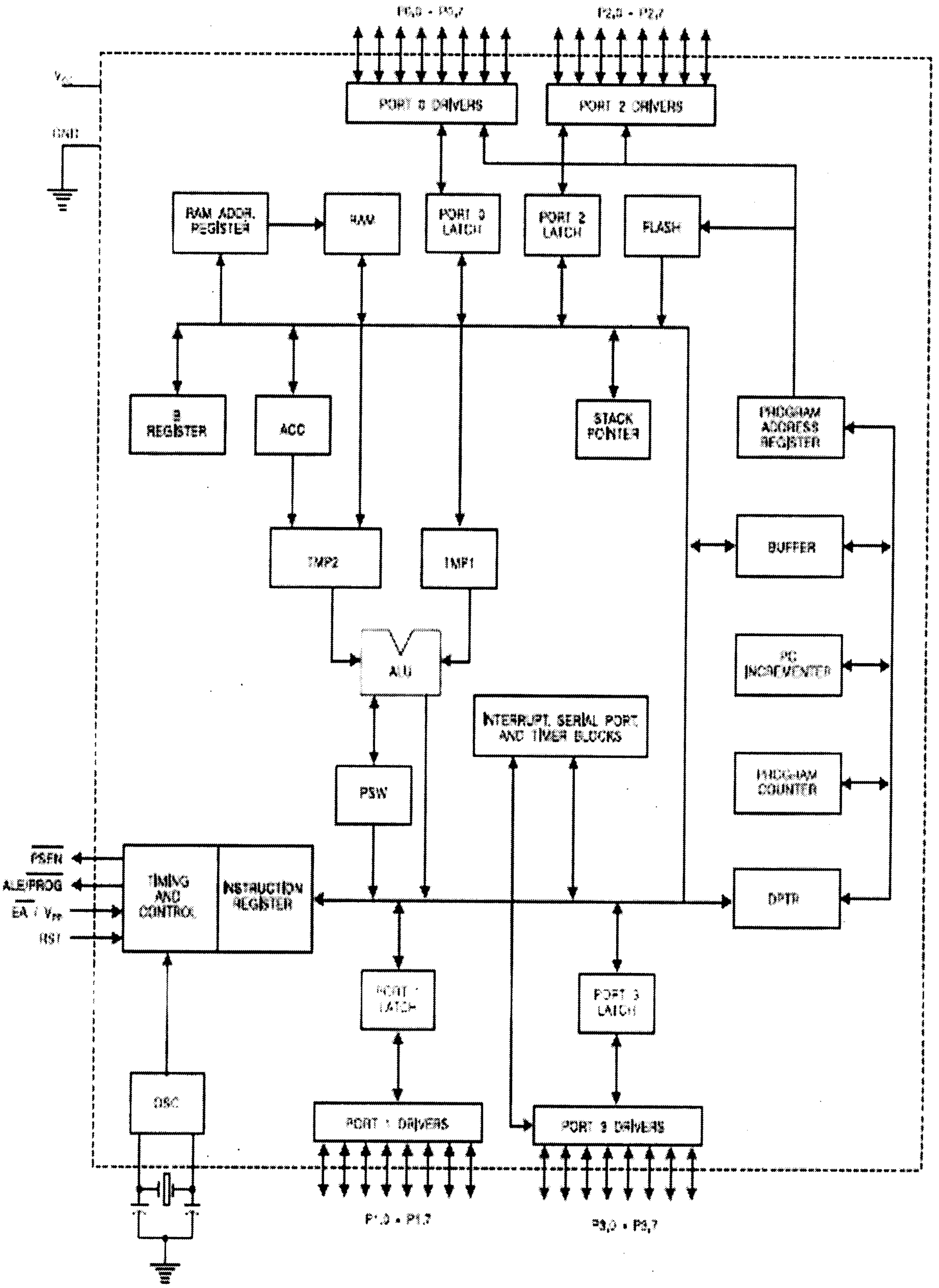

Fig 3.3 Internal Block diagram of AT89C51 Microcontroller

3.1.2 Control program

The control program is the software that is controlling the prototype. Program is a written source code that is embedded in an electronics device for the purpose of controlling the device. The term program here is being viewed through electrical and electronics perspective.

In other words, control program is an embedded programming language or firmware written and burnt onto a particular electronics device to serve as a functioning tool that controls the device. This programming language is classified into Low – level Language and High – level language. In this project, Low- level language is employed.

Low – level Languages are symbolic in which alphabets or combination alpha-numeric is used to represent computer operations and memory addresses. One good example of it is Assembly language which is the language use here in this project. Assembly languages share certain features with machine languages. For instance, it is possible to manipulate specific bit in both assembly and machine language. Programmers use assembly languages when it is important to minimize the time it takes to run a program, because the translation from assemble language to machine language is relatively simple. The source code of the program that controls this project is shown at the appendix 1.

3.1.2.1 Software overview

Since the smallest meaningful unit of time is second, the systems' software was written to keep track of elapsed seconds and adjust the time keeping counter accordingly.

The program defined various registers to hold the different parameters needed for maintaining accurate time and calendar operation as exemplified below:

An internal timer was loaded with a value that caused an interrupt every 250μS. Using a 12MHz crystal, the machine cycle of 1μS was obtained. Thus, for one second time passage, 1,000,000 m/Cs would have elapse. In other word, period (T) is an inverse of frequency (F)

That is, period = 1/frequency

But one machine cycle is equal to twelve instruction cycles.

Therefore machine cycles =crystal frequency/instruction cycle

Machine cycle= 12000000 = 1000000Hz

Period =1/1000000 = 0.000001 =1μS.

Since a 250μs interrupt source was used, two registers were cascaded to provide a total value of <u>4,000</u> ?that was manipulated alongside with T_0 (timer Zero) to produce the required 1 – second time base. The figure 3.4 shows the graphical visualization of the system:

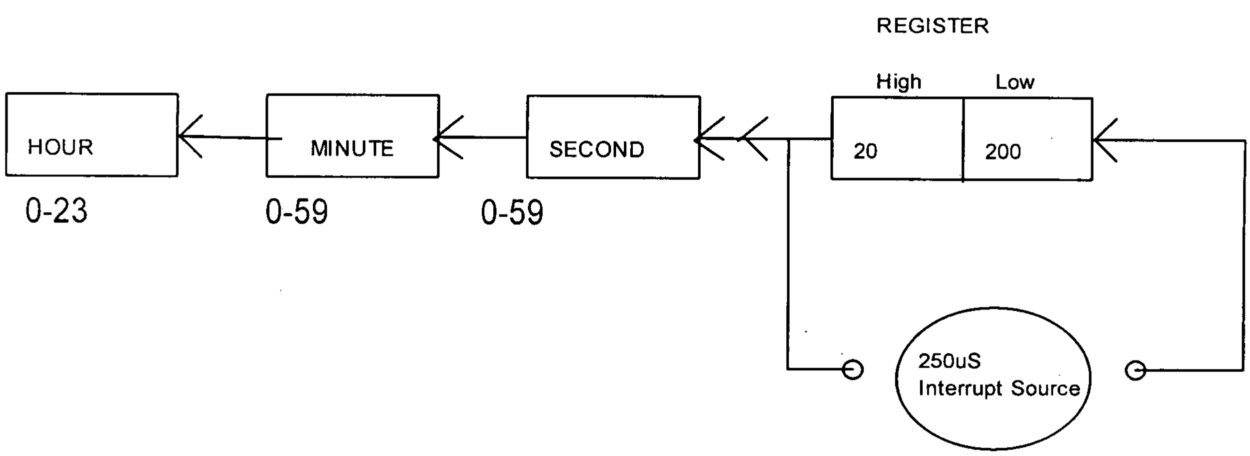

Fig 3.4: Memory Registers

Three RAM variables were designated as 'hour', 'minute' and 'second'. Second is incremented every 1 – second. If seconds is greater than 59, a minute is incremented. If minute is greater than 59, hours is incremented. If hour is greater than 23, the whole display start all over from 000000 (Zeros), that indicates a 24-hour time count.

The binary values in the registers were converted to Binary Coded Decimal (BCD) and eventually to 7-segment code using a Rom-based look-up table.

The calendar function was implemented alongside the clock functionality. RAM variables were assigned MONTH, DAY and YEAR.

Day was incremented every 24 hours. The number of days in MONTH was then compared with the maximum expected days. If DAY exceeds the number of days expected for the month, another month is loaded into MONTH. If month is greater than 12, corresponding to December, the code for January is loaded into MONTH, DAY is reset to zero and YEAR is incremented. If year is grater than 99, YEAR is reset to 00 (Zeros) and another century starts.

Leap year detection was detected by dividing YEAR by 4. If the remainder is zero, a flag, leap – year – flag, is set to 1. In written software routing, if the flag is seen as set, then number of day in February is set to 29, otherwise it is 28.

3.1.3 Clock Source

A number of materials exhibit the property known as *piezoelectricity,* a property that involves mechanical strain and electric fields. When a piezoelectric crystal is subjected to a strain (squeezed or twisted), an electric field develops in the crystal and, hence, a voltage appears between the faces of the crystal. Conversely, when a voltage is applied to the faces of the crystal, the crystal changes shape slightly. Quartz is the most commonly used piezoelectric material. Piezoelectric materials are used in pressure transducer, as well as in many other applications such as setting the frequency for microcontroller-based digital clock/calendar. AT89C51 is the microcontroller chosen in this prototype. The AT89C51 is typically driven by a crystal connected to its pin 18 (XTAL2) and pin 19 (XTALI). The frequencies of the crystal vary from type to another. The particular one to be used is being determined by the application. When a potential difference is applied across its two opposite faces, it causes the crystal to either expand or contract. If an alternating voltage is applied, the crystal is set into vibration. The frequency of the vibration is equal to the resonant frequency of the crystal as determined by its structural characteristic.

The clock source unit is made up of two basic components, which are two 30pf capacitors and one approximately 12.000MHz crystal. This unit generates the clock pulse for the microcontroller to execute its instructions. The part number of this crystal used is 12.000VHF, SPK. The interface between the clock source unit and the microcontroller is shown in figure 3.5

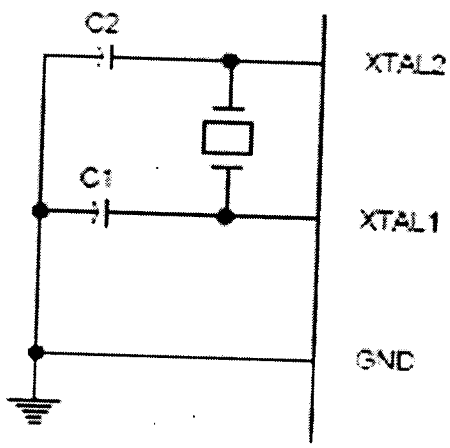

Fig 3.5, Interface between clock source and microcontroller.

This particular crystal generates 12,000,000 pulses in every one second. Normally 8051 compatible microcontroller executes one instruction in 12 clock pulses. But the microcontroller used here can handle as high clock pulses as 24000000

Clock pulses in one second.

3.2 Power supply

A 9V dc battery and an external dc source derived from PHCN were provided for the project use. The two sources are passed through an OR-gate, formed by proper combination of two diodes in the system power supply, in order to successfully pick one source at a time and avoid interruptive effect of PHCN.

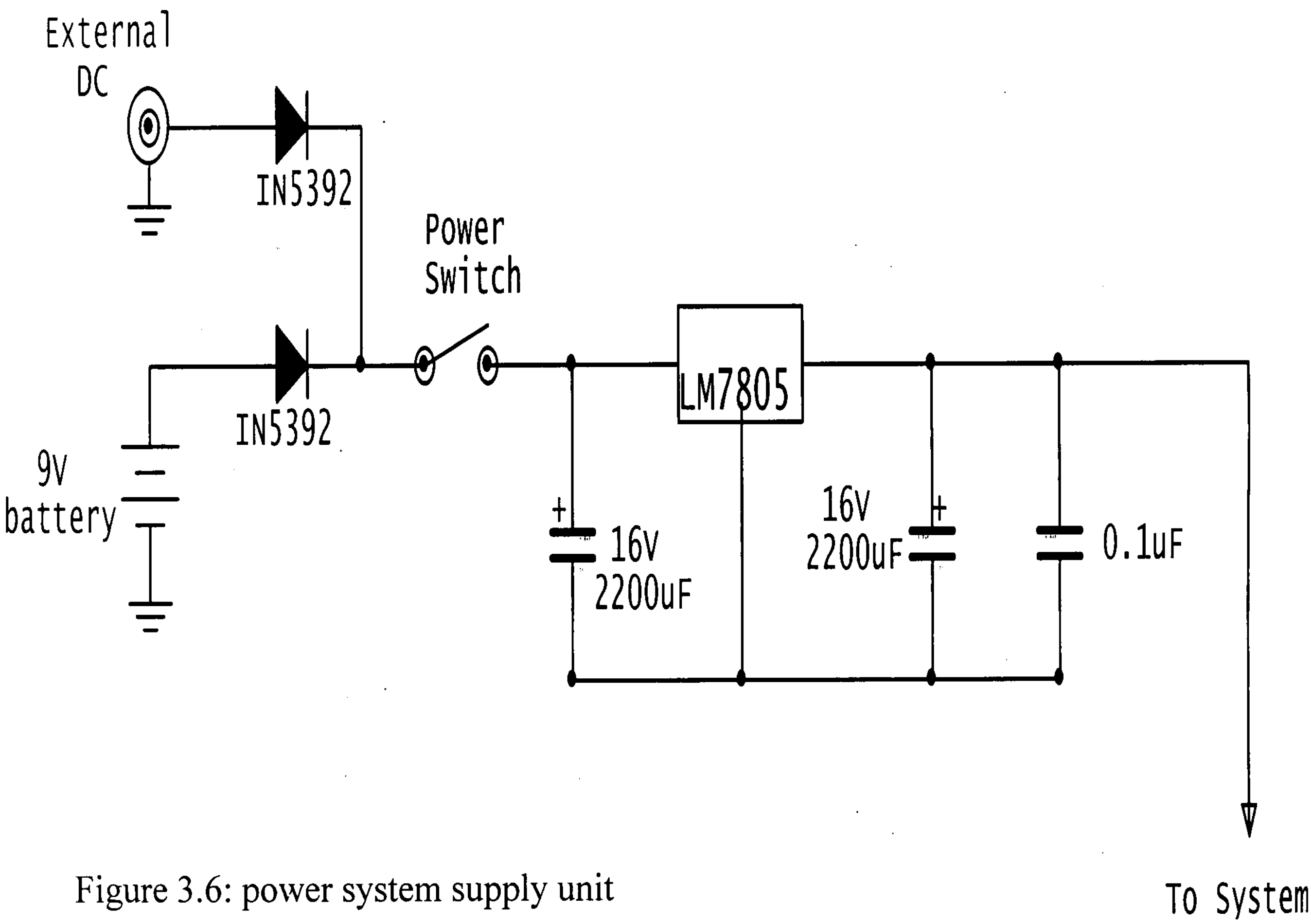

Figure 3.6: power system supply unit

D_1 and D_2 function as an uninterrupted power switching from either battery to external DC or external DC to battery. The voltage appearing at the input pin of the 7805 IC is regulated down to +5V, after stabilization by the 2200µF capacitor on the regulator. The 5-unit input is stabilized by a 2200µF capacitance.

3.2.1 Power supply unit calculation:

Figure 3.6 shows a complete design for a simple power supply that will provide approximately 430mA at 9v. A right choice of component has to be made to obtain a workable design.

From the step-down transformer rated values:

AC: 220/240V 50Hz

DC: 12V 300mA

That is the transformer for this supply has a 220/240v primary and a 12v secondary with a secondary current rating of 300mA.

$I_{rms} = I_{peak} / (2)^{1/2}$

$I_{peak} = (2)^{1/2} * I_{rms}$

$I_{peak} = 1.4142 * 0.3$

0.4243A

424.3mA

$V_{dc} = (V_{peak} (2)^{1/2} - 1.4)$

Where $V_{peak} = 12v$

$V_{dc} = 12*1.4142 - 1.4$

$V_{dc} = 16.9704 - 1.4$

$V_{dc} = 15.57v$

Bridge rectifier was used. For this configuration, the peak voltage developed across the capacitor will be 15.57v. A 7809 is chosen as the regulator. To ensure that the 7809 regulator would operate correctly, there must be at least 3v in addition to the 9v, voltage rating of the regulator. That is, at least, 12v at the input of the regulator. The filter capacitor must be chosen to ensure that this is true. The value of the capacitor must be sufficiently large to store enough charge to allow the regulator to operate when the bridge rectifier is briefly not conducting. The quantity of the capacitor charges is defined as in equations i and ii.

$Q = It$ ......(i)

$\Delta V = (1/C)Q$

$Q = \Delta VC$ ......(ii)

Equating equations i and ii

$\Delta VC = It$ ......(iii)

Where ΔV is the amount the voltage on capacitor C can decrease from its peak value and still be large enough for the regulator to operate. That is ΔV is roughly the difference between the peak voltage from the rectifier and the minimum voltage required to operate the regulator for the desired output voltage.

Therefore. $\Delta V = 15.57 - 12 = 3.57v$

Frequencies relationship of bridge rectifier is;

$F_{out} = 2f_{in}$ (where $f_{in} = 50Hz$)

i.e $F_{out} = 2 * 50 = 100Hz$

but $t\ (\text{period}) = 1/f = 1/F_{out} = 1/100$

$t = 0.01sec$

Knowing that, $\Delta V = 3.57v$ and $I = 0.4243A$. Equation iii becomes

$C = I^* t/\Delta V = (0.4243*0.01)/3.57 = 1188.5\mu f$

That is, the minimum capacitor required (C_{min}) is $1188.5\mu f$

For a conservative design, a capacitor having a capacitance of $1.5*C_{min}$-to-$3*C_{min}$ (i.e $1783\mu f$ -to-$3566\mu f$) with a voltage rating of 20v-to-50v will be reasonable.

Therefore, a capacitor of capacitance $2200\mu f$ was used.

The bridge rectifier was used because it withstands the peak inverse voltage (PIV) of the circuit without any negative effect.

LM7809 integrated circuit was used to stabilize this voltage to a steady 9v.

The 424.3mA and 9v are the output rating of the external mains connected to the main project.

3.3 Display unit

The output of this design is displayed using a series arrangement of 7-segment display through a multiplexed output of microcontroller. The controller has four 8-bit ports .128 bytes of RAM, internal timers and counter e.t.c. It is programmed to control the 6-digit display. The display was multiplexed and each of its 7-segment is being switched on and off by a corresponding driver. The figure3.7 shows the display unit.

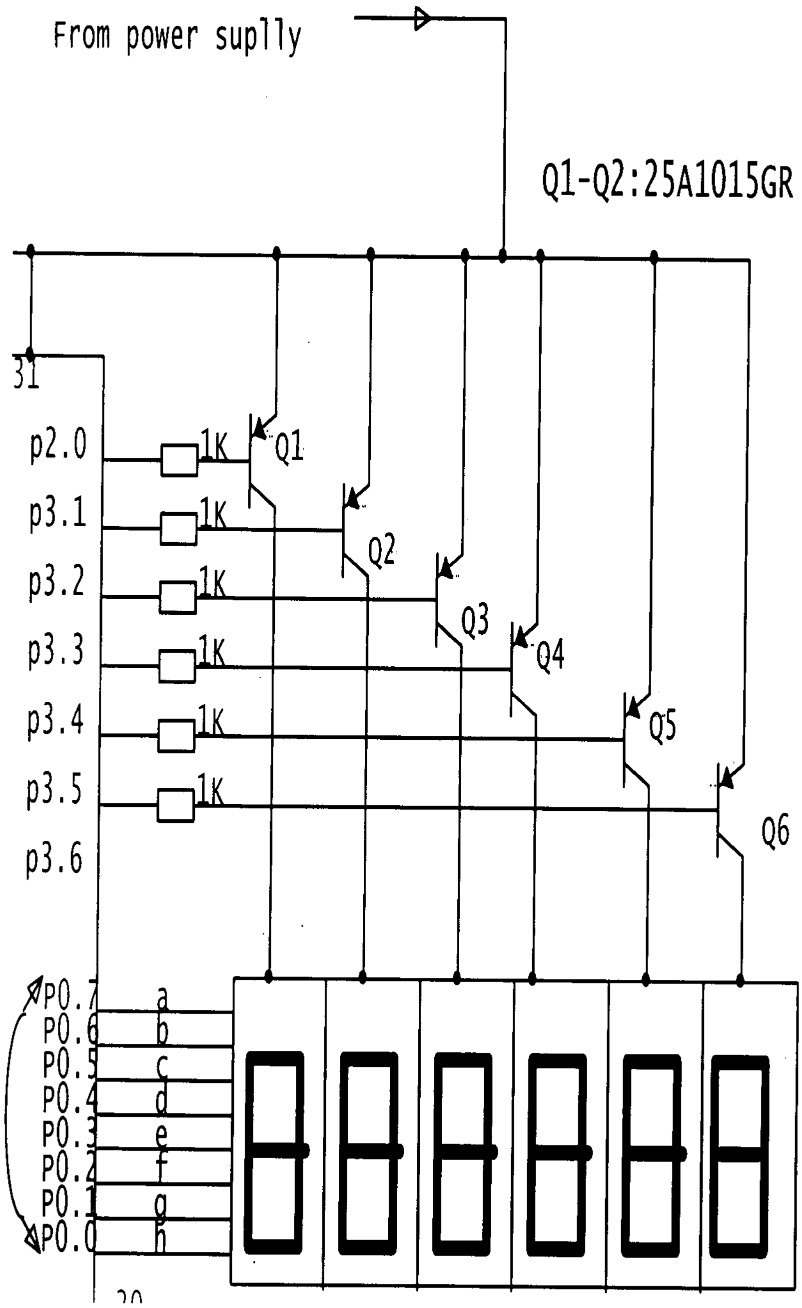

Fig 3.7 the Display Unit Configuration

3.4. Hardware Overview

The hardware of the project consists of sub circuits put together to form the main circuit diagram. The figure 3.9 shows the main circuit diagram.

The display unit was multiplexed to reduce the complexity of wiring 48 input/output pins with the controller.

In multiplexing, Frequency Division Multiplex Allocation (FDMA) was implemented. For example: to display 0,1,2,3,4,5 on the display unit, the following actions, happening in sequence at high frequency.

(i) All digital drivers (Q_1-Q_6) were turned off, that is the display was cleared or initialized to display nothing.

(ii) The binary number corresponding to zero (0) in 7-segment code equivalent was sent to the common data bus (P0)

(iii) Then Q_1 is turned on while others were switched off, for a brief period of time for persistence so that digit 0 (zero) can be displayed in the first 7-segment display.

(iv) After that, Q1 was turned off.

(v) Then, a corresponding binary code that is equivalent to digit 1 (one) in 7-segment code was sent

(vi) The Q2 was then turned on while others were switched off, for a brief period of time for persistence so that digit 1 can be displayed in the second display.

(vii) Similar procedure of the step (ii)-(iv) were repeated to display the remaining other digits in their corresponding 7-segment.

3.4.1 Alarm Unit

The unit supports functioning alarm of 24-hour format. The user first adjusts the hour/minute value. Next, the store button is activated. The adjusted hour and minute value are then stored in alarm-hour and alarm –minute. The real time is then set and alarm on-key pressed. At the end of the timing session, an audible alert sounds to indicate the end of timing. The alert can be disabled by pressing the alarm On/off key.

3.4.2 Keypad

Keypad consists of seven (7) different buttons meant for different functions were one of the sub circuits. Figure 3.8 and table 3.1 illustrate these buttons.

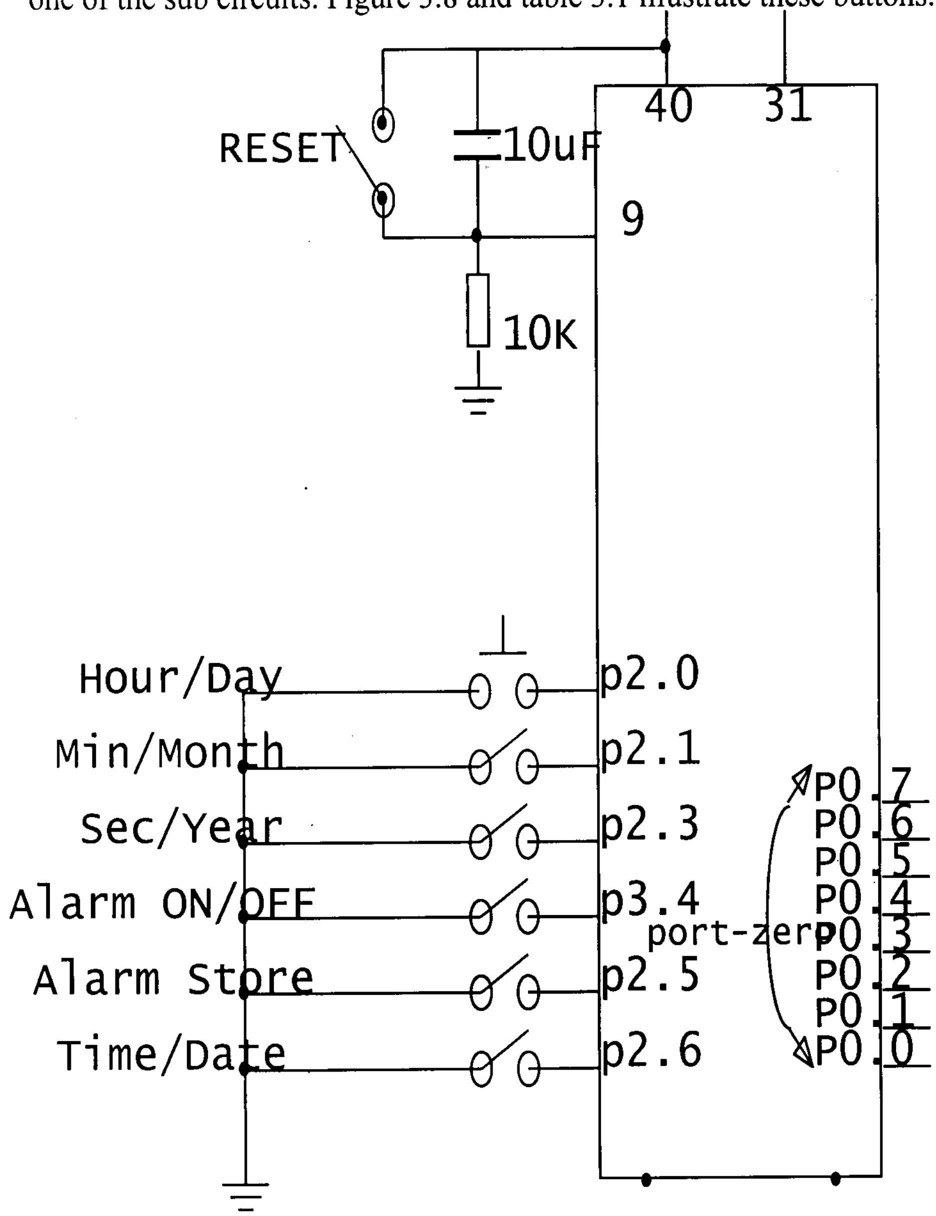

Fig 3.8 Interface of keypad with microcontroller

TABLE 3.1 DESCRIPTION OF KEY

NUMBER OF BUTTON	NAME OF BUTTON	FUNCTION
1	Hour / day	It is used to set the hour of the time display and day of the date.
2	Minute / month	It is used to set the minute of the time and the month of the date.
3	Second / year.	It is used to set the second of the time and the year of the date
4	Alarm ON/OFF	It is used to switches the alarm on and off
5	Mode	It toggle between the time and date
6	Store Alarm	It store the presetting alarm value
7	Reset	It is used to reset the diplay back to initial starting piont

3.4.3 Implementation and Design Calculation

Implementation of this project was done as follows: The microcontroller, being the central processing unit, hosts the connection of other modules. The crystal was connected between pin 18 and pin 19 of the microcontroller. The display unit was connected to port zero (P0) of chip. The reset circuit had its connection through pin 9. All other sub circuits were clearly indicated on the project circuit diagram shown in fig 3.9. The calculation of the whole design was supported or based on the following points:

(i) The transistors were selected because they could handle the peak voltage required by the displays.

(ii) For an n – digit display unit, the required current per segment = n x segment current.

(iii) For a 6 – digit unit and segment current of 10mA, the forward current became:

M unit – 6 X10 = 60mA

Display current = 8X60MA = 480mA:

For each digit drive Ic = 480 mA

The transistor DC gain is 200

Therefore $I_b = I_c/H_{fe} = 0.48/2 \text{ X}10^2 = 2.4 \text{ X } 10^{-3}\text{A}$

$$R_B = V_{CC} - V_{CE}/I_b = 5\text{-}0.7/2.10^{-3} = 1.8\text{k}\Omega$$

That is, the base resistance value of the transistor is 1.8kΩ to provide the required I_C into the display. But a value of 1kΩ was used for a brighter display.

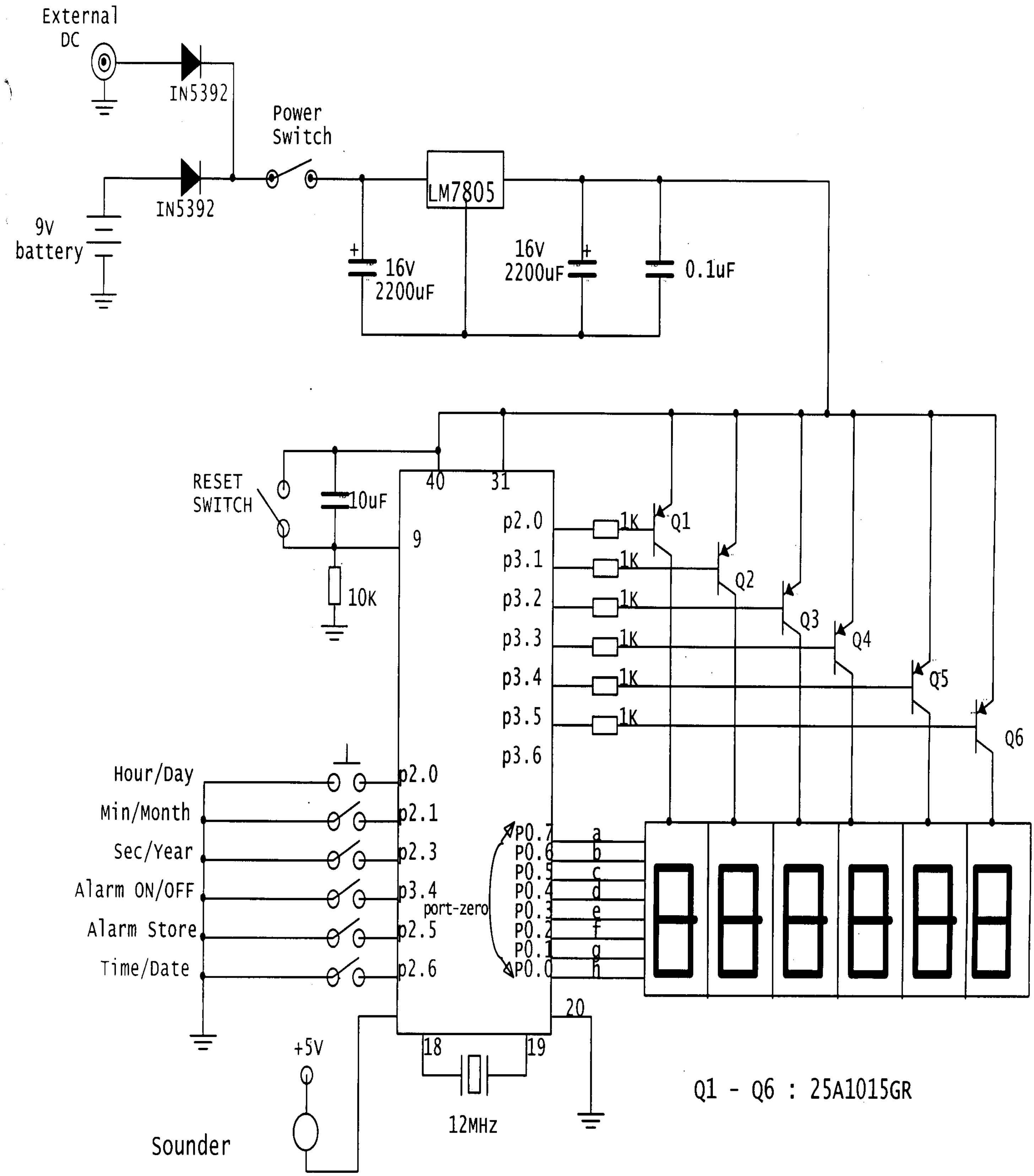

Fig 3. 9 complete diagram of the project

CHAPTER FOUR

TESTS, RESULTS AND DISCUSSION

4.1 Testing

All the components were tested for fitness using digital multimeter before being used in this project work. The resistors, diode, capacitor, cables, transistor, switches and LED were tested for short circuits and open circuits. The software program for the AT89C51 microcontroller was tested by test runing it and debugged were necessary, after which it was burnt into the microcontroller. Further testing was carried out on the project by examining the accuracy of the digital clock against a standard clock and by also checking its correctness for keeping the date of the February of the leap year.

4.2 Results

The 7-segment display unit displays the date and the time similar to the current Gregorian calendar shown in table 2.1 and real time clock.

4.3 Discussion

If there is a complete power interruption, the whole display would clear off and would be reset back to current time of the day. The time display mode is 24 counting mode. That is the 24 hour of the day, is counted from 0' o clock through 23 'o clock.

4.3 Problems and Challenges

It was a serious challenge at initial stage, to come up with precise programming source code that meets the up most target of this prototype. Handling of frequency division (the machine cycles) into the desired microcontroller register to avoid unnecessary overflow of any memory and to obtain the most accurate timepiece

Were strenuous task at start up. All these and among other challenges were achieved painstakingly.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The performances of the design and construction of this project met the design specification. It accomplishes the aims and objectives of the project, which is to design and construct a prototype that can measure the time in hour, minutes and seconds and also give the date, day and year on a display in digital form.

The project was constructed in such a way that it could be easily maintained and repair. Moreover, the presence of the quartz crystal oscillation made timing to be precise and accurate.

5.2 Recommendations

The following recommendations were made based on some areas of the design that could be improved upon, namely;

i. Hourly beep can be introduced to indicate the top of the hour

ii. The use of large LCD (liquid Crystal Display) can be used to provide more information like days, month, and year in letters.

iii. The latest sixteen-segment display can be used to increase the number of the alpha-numeric characters on the display and also enlarged the digits display.

REFERENCES

[1] R. Boylested and L. Nashelsk, [online, Available]: http://physics.nist.gov/GenInt/Time.early.html, 25,june 2008

[2] ancient calendar, [online], Available http://www.calendarzone.com/Janice Mclean1998-2004,oct 3rd 2007

[3] S. W. Amos, Digital devices [Online], Available, http://webxhibits.or/calendars/year.html

[4] Danis Russian, [Online], Available, http://www.aztecalendar.com.Denise Russian 1997-2005,) Oct 12th 2007

[5] Tom Duncan: Success in Electronics. S. Chand and Company Ltd. New Delhi (1999S)

[6] L. Jone, Early physical time. [Online] (2003), Available: http://physics.nist.gov/GenInt/Time.early.html

[7]Electronic Devices, Microsoft ® Encarta ® 2007. © 1993-2006 Microsoft Corporation

[8] Edward Hughes, Hughe Electrical and Electronic Technology, 8m edition, person education (Singapore) pet. Ltd., India, 2004,page 509.

[9]http://www.Obliquity.com/calendar/html .David Hapaper and Lynne Marie Stockman 1998, Oct 16th 2007

[10]http://www.Obliquity.com/calendar/Georgian/html .David Hapaper and Lynne Marie Stockman 1998, Oct 16th 2007

[11]http://www.Obliquity.com/calendar/Islamic/html .David Hapaper and Lynne Marie Stockman 1998, Oct 16th 2007

[12]http://www.Obliquity.com/calendar/Chinese/html .David Hapaper and Lynne Marie Stockman 1998, Oct 16, 2006

[13] The Art of Electronics pages176-415

APPENDIX 1

```
INCLUDE 89c51.mc


display_port equ p0
dx_port EQU p3
stack EQU 60
time_buffer1 DATA 16
time_buffer2 DATA 22
date_buffer1 DATA 28
date_Buffer2 DATA 34
max_loc DATA 40
dx_Ctrl DATA 41
alarm_flag bit 127
alarm_led BIT p2.7
alarm_hour DATA 42
alarm_minute DATA 43
alarm_Second DATA 44
hour DATA 8
minute DATA 9
second DATA 10
update_day_flag bit 126
leap_year_flag bit 124
dp_mask DATA 45
date_time bit 125
month DATA 12
day DATA 11
year DATA 13
max_day DATA 14
new_month DATA 15
day_hour_key BIT p2.0
month_minute_key BIT p2.1
year_Seconds_key BIT p2.2
alarm_key BIT p2.3
time_date_key BIT p2.4
time_2_show equ 60

time_0 data 16h
time_1 data 17h
time_2 data 18h
time_3 data 19h
time_4 data 1ah
time_5 data 1bh
date_0 data 22h
date_1 data 23h
```

```
date_2 data 24h
date_3 data 25h
date_4 data 26h
date_5 data 27h


mask equ date_0
alarm_Store_key bit p2.6
sounder_dx bit p2.5
alarm_Stored bit 123

org 0000h
AJMP sys_reset

org 000bh; tf0 isr here
AJMP tf0_isr
;***********************************************************************
*****

sys_reset:          clr ea
                    mov sp,#stack
                    acall sys_init
mainloop:           ACALL update_system
                    ACALL chk_alarm
                    ACALL scan_key
                    sjmp mainloop
;*******************************************************************


show_long2:         MOV R1,#140
loop_repeat2:  MOV R0,#time_0
                    ACALL write_display
                    DJNZ R1, loop_repeat2
                    ret


sys_init:           MOV tcon,#0
                    mov tmod,#00000010b
                    MOV th0,#06h
                    mov tl0,#06h
                    MOV month,#1
                    MOV day,#1
                    MOV year,#1
```

```
                mov max_Day,#32
                mov hour,#0
                mov minute,#0
                MOV second,#0
                CLR update_day_flag
                clr alarm_flag
                clr alarm_Stored
                SETB alarm_led
                SETB sounder_dx
                CLR leap_year_flag
                mov dx_port,#0ffh
                MOV dx_ctrl,#01111111b
                mov r7,#20
                mov r6,#200
                clr date_time
                MOV ie,#00001111b
                clr ex0
                clr ex1
                setb  ea
                setb tr0
                ret


;**************************************************************************
*****
write_display: MOV dx_port,#0ffh
                MOV dx_ctrl,#01111111b
loop:           MOV display_port, @R0
                mov dx_port,dx_ctrl
                ACALL delay_2_Show
                mov dx_port,#0ffh
                MOV A, dx_Ctrl
                setb c
                RRC A
                MOV dx_ctrl,A
                inc r0
                MOV A, R0
                CJNE a,max_loc, loop
                ret

xlate_Table:   DB
11000000b,11111001b,10100100b,10110000b,10011001b,10010010b,10000010b,11111
000b,10000000b,10010000b
;**************************************************************************
**********8
```

```
;*****************************************************************
show_date:          ACALL convert_date
                    ACALL convert_7seg_date
                    anl date_1,#01111111b
                    anl date_3,#01111111b
                    MOV R0 ,#date_buffer2
                    MOV max_loc,#date_buffer2+6
                    ACALL write_display
                    RET

show_time:          ACALL convert_time
                    ACALL convert_7seg_time
                    anl time_1 ,#01111111b
                    anl time_3,#01111111b
                    MOV R0,#time_buffer2
                    MOV max_loc,#time_buffer2+6
                    ACALL write_display
                    rET
;*****************************************************************
convert_time:       MOV R0,#hour
                    MOV R1,#time_buffer1
                    MOV max_loc,#time_buffer1+6
                    ACALL convert_2_bcd
                    RET

convert_Date:       MOV R0,#day
                    MOV R1,#date_buffer1
                    MOV max_loc,#date_buffer1+6
                    ACALL convert_2_bcd
                    RET

convert_2_bcd:              MOV A,@R0
                    MOV B,#10
                    DIV ab
                    MOV @R1,A
                    INC R1
                    MOV @R1,B
                    INC R0
                    INC R1
                    MOV A, R1
                    CJNE A,max_loc,convert_2_bcd
```

```
            RET
;*************************************************************8

convert_7seg_date:   mov R0,#date_buffer1
                     MOV R1,#date_buffer2
                     MOV max_loc,#date_buffer1+6
                     ACALL look_up
                     RET


convert_7seg_time:          MOV R0,#time_buffer1
                            MOV R1,#time_buffer2
                            MOV max_Loc,#time_buffer1+6
                            ACALL look_up
                            RET

look_up:                    mov dptr,#xlate_Table
loop_again:                 MOV A, @R0
                            MOVC A, @a+dptr
                            MOV @R1, A
                            INC R0
                            INC R1
                            MOV A, R0
                            CJNE A, max_loc, loop_again
                            RET

;****************************************************************
;*************************************************************8
scan_key:
key_1:                jb day_hour_key, key_2
                      acall adjust_day_hour
                      jnb day_hour_key, key_1
key_2:                jb month_minute_key,key_3
                      acall adjust_month_minute
                      jnb month_minute_key, key_2
key_3:                jb year_Seconds_key, key_4
                      acall adjust_year_Seconds
                      jnb year_Seconds_key, key_3
key_4:                jb alarm_key,key_5
                      acall toggle_alarm
                      jnb alarm_key, key_4
key_5:                jb time_date_key, key_6
                      acall toggle_time_date
                      jnb time_date_key, key_5
key_6:                jb alarm_store_key, exit_Scan_key
```

```
                        acall store_alarm
                        jnb alarm_Store_key, key_6
exit_scan_key:          ret

;*****************************************************************8
toggle_time_Date:       CPL date_time
                        mov r4,#40
re_monday15:            acall update_system
                        djnz r4, re_monday15
                        ret

;******************************************************************
toggle_alarm:  JNB alarm_Stored, exit_toggle_alarm
                        cpl alarm_flag
                        JNB alarm_flag, alarm_off
                        clr alarm_led
exit_toggle_alarm:setb sounder_dx
                        sjmp toggle_Delay


alarm_off:              setb alarm_led
                        setb sounder_dx
                        sjmp toggle_delay


store_alarm:            setb alarm_Stored
                        MOV alarm_hour, hour
                        MOV alarm_minute, minute
                        MOV alarm_Second, second
toggle_delay:           mov r4,#100
re_monday16:            acall update_system
                        djnz r4, re_monday16
                        ret


;********************************************************************
*88
adjust_year_Seconds:          jb date_time, adjust_year
adjust_Seconds:                     acall reinit_timer
                              INC second
                              MOV A, second
                              cjne a, #60, chk_seconds
skip_seconds:                 MOV second,#0
                              acall show_time
                              acall restart_timer
```

```
                        MOV R3,#time_2_show
re_monday1:             ACALL show_time
                        DJNZ R3, re_monday1
                        ret


chk_seconds:        JNC skip_seconds
                        acall show_time
                        acall restart_timer
                        mov r3,#time_2_Show
re_monday2:             ACALL show_time
                        djnz r3, re_monday2
                        ret


adjust_year:        INC year
                        MOV A, year
                        CJNE A, #100, chk_year
skip_year:              MOV year,#0
                        SETB leap_year_flag
                        mov r3,#time_2_show
re_monday4:             ACALL show_date
                        djnz r3, re_monday4
                        RET


chk_year:               JNC skip_year
                        CLR leap_year_flag
                        mov r3,#time_2_show
re_monday5:             ACALL show_date
                        djnz r3, re_monday5
                        RET

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

adjust_month_minute:    JB date_time, adjust_month
adjust_minute:          acall reinit_timer
                        inc minute
                        mov a, minute
                        cjne a, #60, chk_minute
skip_minute:            mov minute,#0
                        acall show_time
                        acall restart_timer
                        MOV R3,#time_2_show
            re_monday6: acall show_time
```

```
                        djnz r3, re_monday6
                        ret

chk_minute:             jnc skip_minute
                        acall show_time
                        acall restart_Timer
                        mov r3,#time_2_show
          re_monday7:   acall show_time
                        djnz r3, re_monday7
                        ret


adjust_month:       inc month
                        mov a, month
                        cjne a, #13, chk_month
skip_month:             mov month,#01h
;nuevo
                        acall restart_timer
                        mov r3,#time_2_show
          re_monday8:   acall show_date
                        djnz r3, re_monday8
                        ret


chk_month:              jnc skip_month
                        acall restart_timer
                        ;nuevo
                        mov r3,#time_2_show
     re_monday9:        acall show_date
                        djnz r3, re_monday9
                        ret
;*****************************************************************

adjust_day_hour:        jb date_time, adjust_day
adjust_hour:            acall reinit_timer
                        inc hour
                        mov a, hour
                        cjne a,#24, chk_hour
skip_hour:              mov hour,#0
                        acall show_time
                        acall restart_timer
                        ret


chk_hour:               jnc skip_hour
```

```
                acall restart_timer
                mov r3,#time_2_show
re_monday10:    acall show_time
                djnz r3, re_monday10
                ret

adjust_day:     acall chk_month
                inc day
                mov a, day
                cjne a, max_day,chk_day
                mov day,#01h
skip_day:       mov r3,#time_2_show
re_monday11:    acall show_date
                djnz r3, re_monday11
                ret

chk_day:        jnc skip_day
                mov r3,#time_2_show
re_monday12:    acall show_date
                djnz r3, re_monday12
                ret

chk_leap_year:  MOV A, year
                MOV B,#4
                DIV ab
                MOV A, B
                JZ set_leap_on
                CLR leap_year_flag
                RET

set_leap_on:    SETB leap_year_Flag
                RET


chk_month2:     MOV A,month
                RL A
                MOV DPTR,#month_Table
                JMP @a+dptr
```

```
            AJMP sys_reset
month_table:
            aJMP set_january
            AJMP set_february
            AJMP set_march
            AJMP set_april
            AJMP set_may
            AJMP set_june
            AJMP set_july
            AJMP set_august
            AJMP set_september
            AJMP set_october
            AJMP set_november
            AJMP set_December

set_january:
            MOV max_day,#32
            MOV new_month,#02h
            RET

set_february:
            ACALL chk_leap_year
            JB leap_year_Flag, go_february
            MOV max_day,#29
            MOV new_month,#03h
            RET

go_february:
            MOV max_day,#30
            MOV new_month,#03h
            RET

;fix all these l nes here
set_march:
            MOV max_day,#32
            MOV new_month,#04h
            RET

set_april:
            MOV max_day,#31
            MOV new_month,#05h
            RET

set_may:
            MOV max_day,#32
            MOV new_month,#06h
            RET

set_june:
            MOV max_day,
            MOV new_mon
            RET
```

```
set_july:           MOV max_Day,#32
                    MOV new_month,#08h
                    RET

set_August:         MOV max_Day,#32
                    MOV new_month,#09h
                    RET

set_september:      MOV max_day,#31
                    MOV new_month,#0ah
                    RET

set_october:        MOV max_day,#32
                    MOV new_month,#0bh
                    RET

set_november:       MOV max_day,#31
                    MOV new_month,#0ch
                    RET

set_December:       MOV max_day,#32
                    MOV new_month,#01h
                    RET

tf0_isr:        PUSH acc
                    PUSH psw
                    DJNZ R6, exit_isr
                    mov r6,#200
                    DJNZ R7, exit_isr
                    mov r7,#20
                    inc second
                    MOV A, second
                    cjne a, #60, go_Seconds
                            MOV second,#0
skip_seconds2:      INC minute
                    MOV A, minute
                    cjne a, #60, go_minutes
                            mov minute,#0
skip_minutes2:      INC hour
                    MOV A, hour
                    cjne a, #24, go_hours
                    mov hour,#0
skip_hours2:        SETB update_Day_flag
exit_isr:           POP psw
                    POP acc
                    RETI
```

```
go_Seconds:             jnc skip_seconds2
                        POP psw
                        POP acc
                        reti

go_minutes:             jnc skip_minutes2
                        POP psw
                        POP acc
                        reti
go_hours:               jnc skip_hours2
                        POP psw
                        POP acc
                        reti
;***********************************************************8

restart_timer:          setb tr0
                        ret

;***********************************************************
reinit_timer:           clr tr0
                        clr tf0
                        mov tl0,#06h
                        mov r7,#20
                        mov r6,#200
                        ret
;***********************************************************

update_system:                  JBC update_day_flag, update_system24
recheck:                JB date_time, show_date_now
                        ACALL show_time
                        ret

show_date_now:                  ACALL show_date
                        ret


update_system24:        ACALL chk_month2
                        INC day
                        MOV A, day
                        CJNE A, max_day, chk_Date2
go_Backdate:            MOV day,#01h
                        MOV month, new_month
                        MOV A, month
                        XRL A, #01h
```

```
                        JZ adjust_year_now
                        SJMP recheck
adjust_year_now:        mov r3,#time_2_show
re_monday13:            ACALL adjust_year
                        djnz r3, re_monday13
                        SJMP recheck

chk_date2:              JNC go_backdate
                        SJMP recheck

;*****************************************************************8
chk_alarm:              JNB alarm_flag, exit_chk_alarm
                        MOV A, alarm_hour
                        CJNE A, hour, go_2
                        MOV A ,alarm_minute
                        cjne a, minute, go_3
                        acall timer_done
exit_Chk_alarm:              ret

go_2:                   jnc exit_Chk_alarm
                        acall timer_done
                        ret

go_3:                   jnc exit_chk_alarm
                        acall timer_done
                        ret
;*****************************************************************
timer_done:             clr alarm_flag
                        clr alarm_stored
                        setb alarm_led
                        clr sounder_dx
                        RET

delay_2_show:

                mov r2,#0
                djnz r2,$
                ret

                end
```

APPENDIX 2

Operation Manual

1. Connect the standalone power pack code to the electric power outlet and fix in the av alkaline battery. Although any of these power sources can be used alone but it is advisable not to use PHCN alone.

2. Switch on the power switch if it was switched off.

3. Use button 1, button 2, and button 3 to set the hour, minute and second respectively.

4. Press button 5 once to change over to date mode.

5. Use the same buttons in number 3 to set day, month and year respectively.

6. To see the current time and date, button 5 is used to toggle between time and date.