DESIGN AND CONSTRUCTION OF MICROCONTROLLER-BASED AUTOMATIC COLLEGE BELL

BY

UMAR MUSA
MATRICULATION NUMBER:
2005/22055EE

A PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING, SCHOOL OF ENGINEERING AND ENGINEERING TECHNOLOGY, FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA, IN PARTIAL FULFILMENT OF THE REQUIREMENTS OF THE AWARD OF THE BACHELOR OF ENGINEERING (B. ENG.) DEGREE IN ELECTRICAL AND COMPUTER ENGINEERING

NOVEMBER, 2010

DEDICATION

This project work is solely dedicated to the Holy prophet Muhammad (S.A.W.), my parents, and the entire members of my family and all those that have contributed for the success of this project.

No Certification

DECLARATION

I Umar Musa declare that this work was done by me and hereby relinquish the copyright to the Federal University of Technology, Minna.

UMAR MUSA	MR. P.O.ABRAHAM-ATTAH
(Name of student)	*(Name of Supervisor)*
08/10/2010	10/11/2010
(Signature and date)	*(Signature and date)*
ENGR.A.G RAJI	
(Name of H.O.D)	*(Name of External Examiner)*
(Jan. 27, 2011)	
(Signature and date)	*(Signature and date)*

ABSTRACT

This project presents the design and construction of an Automatic College Bell is used mainly both in Primary and Secondary Schools and places where time becomes an important parameter of day activities. The design works on the principle of timing programmed in microcontroller chip. It provides ten (10) different schedules, an emergency ringing is included and it is active for every schedule selected. The features included are: the time-select switches, microcontroller, LED indicators, switching transistor, relay and the electric bell.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER ONE

GENERAL INTRODUCTION

1.1 Introduction

Nowadays science and technology is improving almost every day all over the world; this then gives rise in embracing a microcontroller base automatic college bell. This is use in various schools (both primary and secondary), colleges, industries and various places where time becomes an important issue to consider and in comforting our living conditions in various places named above. The design and construction of automatic college bell is important because, most of the present days work activities are not normally operated again. They are automatically carried out by the use of electronic components.

This project, which is the Design and Construction of an Automatic College Bell, is another electronic project that steps further in improvement of our living condition in both Primary and Secondary Schools where a student, known as Time-Keeper is assigned for the ringing of the bell.

A bell is signal in a School, either a real bell, a distributed ringer or a sound heard over the intercom that tells the students when it is time to go to class in the morning and when it is time to change classes during the day. Typically, the first bell tells the students that it is time to report class [1].

The bell is an important instrument in both Primary and Secondary Schools and even in the industries where the bell timer plays a critical role in running the day [2]. Bells are also associated with clocks indicating the hour by ringing. Clock towers or bell towers can be heard over long distances which was especially important in the time when clocks were too expensive for wide spread use. A bell is a simple sound-making device. The bell is a percussion instrument and an idiophone. Its form is usually an open ended hollow drum which resonates upon being struck. The striking implement can be a tongue suspended within the bell known as a clapper, a small free sphere enclosed within the body of the bell, or a separate mallet [3].

Bell are usually made of cast metal, but small bells can also be made from ceramic or glass. Bell can be of all sizes: from tiny dress accessories to Church bells weighing tons. The ringing of bells known as bell ringing, and such a bell produces a very loud, clear tone. If the bell is cast, it is called a "maiden bell" while "tuned bells" are worked after casting to produce a precise note. The traditional metal for these bells is a bronze of about 20% tin. Known as bell metal, this alloy is also the traditional alloy for the finest Turkish and Chinese cymbals. Other materials sometimes used for large bells include brass and iron. The process of casting bells is called bell making or bell founding [3].

1.2 Project Aim and Objectives

The main aim and objective of this project is to design and construct an automatic college bell for the use in primary and secondary schools, colleges, industries and some other interesting areas of applications where timing is needed. This is also built to be adaptive or easily interface with other systems e.g. security and fault detection systems; and also design of a reliable and effective automatic college bell system that facilitates easy operation of the bell automatically.

1.3 Features of the Project

The following are the features of the designed project

(i.) The bell is played at preset times.

(ii.) There is no need of assigning person for ringing the bell every time.

(iii.) Accuracy in timing is achieved.

(iv.) It eliminates manual operation of the bell.

(vi.) Provides ten (10) different possible event schedules each with emergency.

1.4 Limitation

It is important to know that everything in this world has a guided rules facts or conditions that limit its operations and nothing is absolutely 100% efficient. Therefore, this project design has limitation, that is, the range at which it operates. It has ten (10) different time selections period. Each selection has an emergency ringing and does not operate outside this limit.

CHAPTER TWO

LITERATURE REVIEW

2.1 History of the Bell

Bells were known in China before 2000 BC and in Egypt, India, Greece, Rome, and other ancient cultures. From earliest times they were used as signaling devices, as ritual objects, and as magical, often protective, amulets (often hung in doorways or around the necks of animals). The use of bells in churches spread through Europe in the 6th to 11th centuries and were first used in Eastern Christian churches in the 9th century [4].

The earliest metal bells were apparently hammered into a cup shape from a flat piece of metal. When the process of casting metal was discovered, many bells were cast of bronze. The casting of bells declined in late antiquity, and hand bells of the cowbell type came into use; these were made of thin metal plates bent into rectangles and fastened with rivets. About AD 800 the casting process resurfaced, making possible the manufacture of large tuned bells [4].

By about 1400 the characteristic campaniform shape of Western bells had evolved: square-shouldered, with straight or slightly concave sides (the waist), flaring out and thickening near the rim (the sound bow). This stronger shape also improved the tone, and in the 15th to 18th centuries bell-makers in the Low Countries specialized in producing bells so well tuned that they could be played in harmony. In England, sets of somewhat differently tuned bells were rung in complicated permutations of a standard sequence, a process known as change ringing [4].

A type of iron hand-bell was developed in sub-Saharan Africa and remains an integral part of many African musical traditions. Because the hand-bells typically have no clapper, they are struck with a beater to produce sounds. The sharp, penetrating sound of the iron hand-bell can also be heard in the African-influenced music of Latin America [4].

2.1.1 Hand Bell

A hand bell is a bell designed to be rung by hand. To ring a hand bell, a ringer grasps the bell by its slightly flexible handle-traditionally made of leather, but often now made of plastic, and move the wrist to make the hinged clapper inside the bell strike [5]. The first tuned hand bells were developed by brothers Robert and William Cor in Aldbourne, Wiltshire, England between 1696 and 1724. The Cor brothers originally made latten bells for hame boxes, but for reasons unknown, they began tuning their bells more finely to have an accurate fundamental tone, and fitted them with hinged clappers that only move in one plane [5].

Hand bells were first brought to the United States from England by Margaret Shurcliff in 1902. She was presented with a set of ten (10) hand bells in London by Arthur Hughes, the general manager of the White-Chapel Bell Founding after completing two separate $2\frac{1}{2}$ hours change ringing peals in one day [5].

2.1.2 Some other Bells in Existence

i. The Tsar Kolokol bell is the largest bell still in existence. It weighs 160 tones. It was casted in 1733, in Moscow, Russia [3, 4].

ii. The world peace bell was the largest functioning swinging bell until 2006. It is located in Newport, Kentucky, United States, cast by Paccard of France. The bell itself weighs 66,000Ib while with clapper and supports the total weight which swings when the bell is tolled is 89,390Ib [3].

iii. The Liberty bell is an American bell of great historic significance, located in Philadelphia, Pennsylvania. It previously hung in independence hall and was rung on July4, 1776 to mark American independence [3].

iv. Little John, named after the character from the legends of Robin Hood, is the bell within the Clock Tower of Nottingham Council House. It is the deepest bell in the United Kingdom and its chimes are said to be heard over the greatest distance of in the UK [3].

v. Big Ben is the hour bell of the great clock in St. Stephen's Tower at the Palace of West minister, the home of Houses of Parliament in the United Kingdom [3].

2.2 Background Information

A college bell is the bell that is mostly use in both Primary and Secondary schools which serve as the sound or alert signal to both students and staff. A college bell rang out for the first time since before the Second World War at Stanton Peak Primary School, England. Many of the bells listed above are ringing manually; therefore there is need for these bells to be rung automatically without human intervention. Hence, that is the aim and objective of this project. The design and construction of this project is achieved by the use of electronic components which serve as the mechanism behind the operation of the bell. The major component in this project is microcontroller which coordinates and controls the overall activities of the other components. This can better be understood by referring to the fig.2.1. This is an effective and useful project for educational institutions [7]. In most Schools and Colleges, the Timekeeper rings the bell after every period (usually of 40-min duration). The timekeeper has to depend on his wrist watch or clock and some time will forget to ring the bell in time [7].

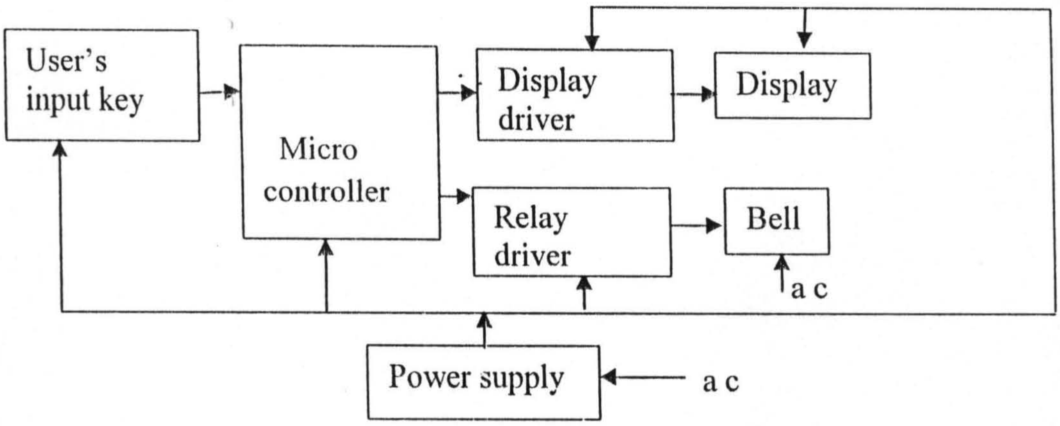

Fig 2.1 Block diagram of automatic College Bell

(A.) Power Supply

Power supplies are an essential part of all electronic systems from the simplest to the complex [8, 10]. A basic power supply consists of a transformer, a rectifier, a filter and a regulator [8, 9, 10, 11, 12, 14, 18]. A power supply filter greatly reduces the fluctuation in the output voltage of a half-wave or full-wave rectifier and produces a nearly constant-level dc voltage. Filtering is necessary because electronic circuits required a constant source of dc voltage and current to provide power and biasing for proper operation. Filtering is accomplished using capacitors [8, 9, 10, 14], as you will see in the next chapter. Voltage regulation is usually accomplished with integrated circuit voltage regulators. A voltage regulator prevents changes in the filtered dc voltage due to variation in line voltage or load [8, 9, 10, 11, 12, 14, 18].

(B.) Time-select Switching Unit

This project is design to provide maximum number of ten different schedules. This unit allow for the selection of the schedule suitable for every School. The unit consists of normally opened switches. Each of the switches represents a single schedule which can be chosen by individual. When any of these switches is activated, it directs the microcontroller to carry out some specific task as programmed internally.

(C.) Microcontroller

A microcontroller (also microcontroller unit, MCU or µC) is a small computer on a single integrated circuit consisting of a relatively simple CPU combined with support functions such as a crystal oscillator, timers, watchdog, serial and analog I/O etc. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a, typically small, read/write memory [15].

Microcontrollers are designed for small applications. Thus, in contrast to the microprocessors used in personal computers and other high-performance applications, simplicity is emphasized. Some microcontrollers may operate at clock frequencies as low as 32kHz, as this is adequate for many typical applications, enabling low power consumption (milliwatts or microwatts). They will generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nanowatts, making many of them well suited for long lasting battery applications. Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, remote controls, office machines, appliances, power tools, and toys. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes [15].

The first single chip microprocessor was the 4 bit Intel 4004 released in 1971, with other more capable processors available over the next several years. These however all required external chip(s) to implement a working system, raising total system cost, and making it impossible to economically computerize appliances [15].

The first computer system on a chip optimized for control applications - microcontroller was the Intel 8048 released in 1975, with both RAM and ROM on the same chip. This chip went on to be found in over a billion PC keyboards, and numerous applications [15, 21]. The whole story has its beginnings in the far 80s when Intel launched the first series of microcontrollers called the MCS 051. Even though these microcontrollers had quite modest features in comparison to the new ones, they conquered the world very soon and became a standard for what nowadays is called the microcontroller [21]. Most microcontrollers at this time had two variants. One had an erasable EEPROM program memory, which was significantly more expensive than the PROM variant which was only programmable once [15, 21].

In 1993, the introduction of EEPROM memory allowed microcontrollers (beginning with the Microchip PIC16x84) to be electrically erased quickly without an expensive package as required for EPROM, allowing both rapid prototyping, and In System Programming [15].

The same year, Atmel introduced the first microcontroller using Flash memory. Other companies rapidly followed suit, with both memory types. Cost has plummeted over time, with the cheapest microcontrollers being available for well under $0.25 in 2009 and 32 bit microcontrollers under $5. Nowadays microcontrollers are low cost and readily available for hobbyists, with large online communities around certain processors [15]. Therefore, the microcontroller, after the appropriate switch has been use to activate the time, process internally and give out the outputs to both relay driver and the seven segment display which indicate the input time.

(D.) LED Indicators

Light-emitting diodes (LEDs) are good devices to visually display a HIGH (1) or LOW (0) digital state [13]. Thus they are used here to indicate that there is supply from the source.

(E.) Relay Driver

This consists of transistor and relay. A transistor is used to establish the current necessary to energize the relay in the collector circuit [14]. The bipolar transistor is a very commonly used switch in digital electronic circuits. It is three-terminal semiconductor component that allows an input signal at one of its terminals to cause the other two terminals to become a short or an open circuit [13]. Also in [14] the application of transistor is not limited solely to the amplification of signals. Through proper design transistors can be used as switches for computer and control applications. Transistor can be used to: switch currents, voltage and power; perform digital logic functions; and amplify time-varying signals [9, 12]. The output from the microcontroller is connected to the base of the transistor. With no input at the base of the transistor, the base current, collector current and coil current are essentially 0A, and the relay sits in the unenergized state (normally open, NO). However, when a negative pulse (PNP type) is applied to the base, the transistor turns on, establishing sufficient current through the coil of the electromagnet to close the relay [14], thereby triggering the output (Bell). Ideally, at turn-off the current through the coil and the transistor will quickly drop to zero, the arm of the relay will be released, and the relay will simply remain dormant until the next 'ON' signal [14].

CHAPTER THREE

DESIGN AND CONSTRUCTION

As understood from the chapter two, there is need to analyze each block of the project for proper design and implementation. Therefore, this design can be done as follows:

3.1 Design of Power Supply Unit

The dc power supply convert the standard 230V, 50Hz ac available to all wall outlets in to a constant dc voltage. A typical dc power supply consists of five stages: the transformer, the rectifier, the filter, the voltage regulator and the load. Thus the ICs and the other components used run on a power supply of 5V, hence the supply must be regulated to prevent fluctuation in voltage level. The block diagram is as shown in fig 3.1 bellow.

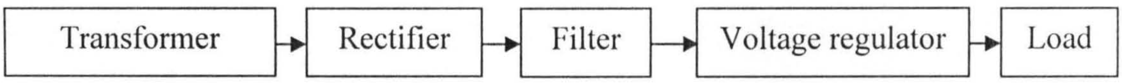

Fig 3.1 A Typical dc Power Supply Block Diagram

The various sections of the power supply block diagram are briefly explained as follows:

(i.) **Transformer:** its job is either to step up or (mostly) step down the ac supply voltage to suit the requirement of the solid state electronic devices and circuits fed by the dc power supply [9].

(ii.) **Rectifier:** it is a circuit which employs one or more diodes to convert ac voltage into pulsating dc voltage [8, 9, 10, 12, and 13]. A rectifier can be either a half-wave rectifier or full-wave rectifier [8]. In this design a full-wave rectifier was employed.

(iii.) **Filter:** the function of filter of this circuit element is to remove the fluctuations or pulsations (called ripples) present in the output voltage supplied by the rectifier [9]. This is done by connecting a capacitor filter to the rectifier.

(iv.) **Voltage Regulator:** its main function is to keep the terminal voltage of the dc supply constant even when the ac input line voltage to the transformer, or the load varies [8, 9, and 18].

(v.) **Load:** the load block is usually a circuit for which the power supply is producing the dc voltage and load current [8].

The complete circuit diagram of the power supply is as shown in the figure bellow

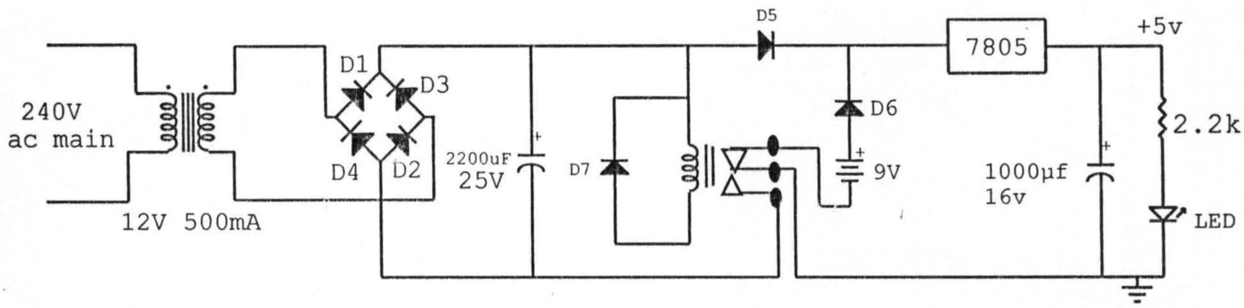

Fig 3.2 Circuit Diagram of Power Supply Unit

The main voltage of 220V is stepped down by a 220/12V, 500mA transformer. It is then rectified by full wave bridge diode rectifier. The waveform at this stage has no negative component, but a lot of ripples. Smoothening capacitors are needed to reduce the ripple to an acceptable level. The resulting ripple voltage (dc) can be calculated.

The load causes the capacitor to discharge between half cycles. If the load current stays constant, as it for small ripple, then

$$I = C\frac{dv}{dt} \quad \text{...} 3.0$$

The frequency of the full wave signal is double the input frequency. The full wave rectifier inverts each negative half cycle, so that we get double the number of positive half cycles. The effect is to double the frequency.

Therefore,

$f_{out}=2f_{in}$ (I.e. twice the value of the input) 3.1

$$dt = \frac{1}{2f_{in}} \quad \text{..........3.2}$$

$$= \frac{1}{2\times50}$$

The maximum current that can be drawn by the main circuit is determined by the voltage regulator following the filtering capacitor, the 7805.

The standard 7800 series can produce output current in excess of 1A when used with adequate heat sink. Therefore, it can supply a maximum of 1A. This current will be drawn from the supply. Thus I_{load} =1A (maximum). The value of capacitance C can be calculated from

$$C = I\frac{dt}{dv} \quad \text{..........3.3}$$

But generally, dv which is the ripple voltage is chosen to be 30% of V_P where V_P is the peak voltage.

Therefore,

$V_P = V_{rms}\sqrt{2}$ 3.4

where V_{rms} = 12V, since the transformer of 220/12V was used

$V_P = 12\sqrt{2} = 16.94V$

For bridge rectifier, $V_{P\,(out)} = 12\sqrt{2} - 1.4$

$=16.97 - 1.4 = 15.57V$

$$dv = \frac{30}{100} \times 15.57 = 4.67V$$

$$\text{Therefore, } C = \frac{1 \times 0.01}{4.67} = 2.141 \times 10^{-3}F$$

$$= 2{,}141\mu F$$

So, the commercial value of 2,200μF, 25V was used in order to reduce the ripple to the nearest minimum. Then the expected ripple voltage using this value of capacitor is

$$dv = \frac{1 \times 0.01}{2200} = 4.55V$$

This means that the output wave form goes from peak value of 15.57V to (15.57 - 4.55 = 11.02V). It may be noted that the input voltage to the IC regulator must be at least 2V above the output voltage. This is required in order to maintain regulation.

Therefore, the peak value of 15.57V to 11.02V is acceptable since the output voltage is 5V.

The ripple is neglected by the 7805 to a negligible value.

The average voltage going to 7805 is calculated by

$V_P - 0.5dv = 15.57 - 0.5 \times 4.55$

$= 13.30V$

The 1000μf capacitor connected across the output act to store charges so that the relay switches in millisecond when power failure occurs. The capacitor discharges the stored charge to hold the microcontroller millisecond's operative current before the backup (battery) takes over. This process is known as cold redundancy. The capacitor also acts as line filter to improve the transient response.

For power indication, a light emitting diode (LED 1) is connected from the positive supply line immediately after the capacitor to ground through a resistor. The resistor value is determined by the current carrying capacitor of the diode. A typical red LED will drop 1.7V cathode to anode when forward biased and will illuminate with 10 to 20mA flowing through it. Since the red LED is used as an indicator, then the required limiting resistor can be calculated as:

$13.30 = V_d - I_d R = 1.7 + 10mA \times R$

Where Id=10mA, V_d=1.7V

$$R = \frac{13.3-1.7}{10\times10^{-3}} = \frac{11.6}{10\times10^{-3}}$$

$$= 1.16K\Omega$$

The commercial value of 2.2KΩ was used in the design.

The diode and resistor served as a path to ground which the smoothing capacitor can discharge after the supply has been turned off. This prevents high voltages that might damage other part of the circuit.

The back up power source was added to the power supply in case of main power supply failure. This is achieved by connection of the relay which switches to battery. The anti-parallel diode is connected across the relay to ensure unidirectional flow of current.

3.2 Design of User's Input Keys

This unit made provision for the users to set the time of the day, input time of the bell ringing and also to clear the input time of the bell ringing if not wanted or if mistake is made by input time. These are achieved by pressing the appropriate keys of the keyboard.

The unit consists of six (6) push-to-on switches S1 through S6. Each switch is connected from the pin of microcontroller has ability to sink or source current. The type of microcontroller used in this design is AT89S51.The circuit diagram of the unit is as shown in the next page.

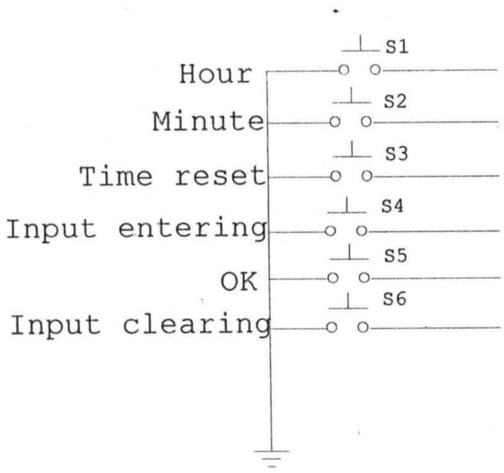

Fig3.3 Circuit Diagram of User's Input Keys

In order to set the time of the day, time reset key (S3) has to be pressed first, follow by hour and minute keys to set the hour and minute digits respectively, and finally press the OK key (S5) to store and continue with the set time.

Also, for entering the time of the bell ringing, press the input entering key (S4) first, follow by the hour and minute keys as in the case of time reset, and finally OK key too to store the inputted time.

Input clearing key (S6) clears all the input entered if pressed. This key can be used, if inputs entered are wrong or not wanted.

3.3 Design of Microcontroller Unit

The major component in this project design is AT89S51 microcontroller which controls, coordinates and directs all the activities and behaviors of this design. Most control application required extensive input/output and need to work with individual bits. The AT89S51 addresses both of these needs by having 32 input/output, bit manipulation and bit checking.

The microcontroller contains the system software clock as programmed within the chip. The input from user's input keys goes to the microcontroller and the microcontroller executes the specific functions by either display the result on the seven-segment display or ring the bell. The circuit diagram of this unit is shown in fig3.4

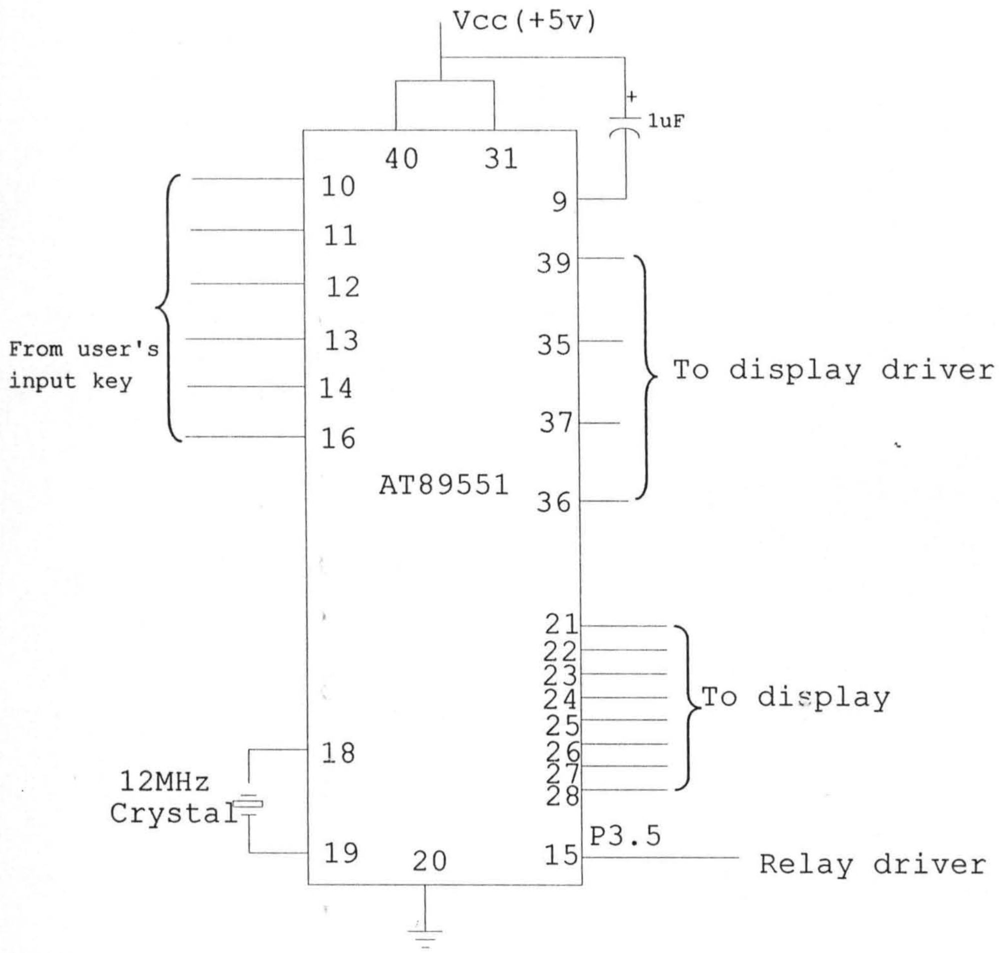

Fig 3.4 Circuit Diagram of Microcontroller Unit

3.3.1 Features of AT89S51

- Compatible with MCS-51 products
- 4K bytes of In-system reprogrammable flash memory
- Endurance: 1,000 write/erase cycle
- Fully static operation: 0Hz to 24Hz
- Three –level program memory lock
- 128×8bit internal RAM

⇨ 32 programmable I/O lines

⇨ Two 16-bit timer/counters

⇨ Six interrupt sources

⇨ Programmable serial channel

⇨ Low-power idle and power-down mode.

3.3.2 Oscillator Characteristics

XTAL1 and XTAL2 are the input respectively, of an inverting amplifier which can be configured for used as on-chip oscillator, as shown in fig 3.4 Either A quartz crystal or ceramic resonator may be used. The oscillatory circuit is the 'heartbeat' of the system and it is crucial to correct operation for example if the oscillator fails the system will not function at all; if the oscillator run irregularly, any timing calculation performed by the system will be inaccurate. In this design, a quartz crystal was used.

3.4 Design of 4-digit 7-segment Display

For visual interactively with the user, a display provided a visual presentation of the system time to the user and also enables presetting.

A common-anode arrangement was used since it allow current to be sunk through the LEDs by the microcontroller. The digits were multiplexed to reduce wiring complexity. The digits were individually driven by a PNP anode driver as shown in fig.3.5

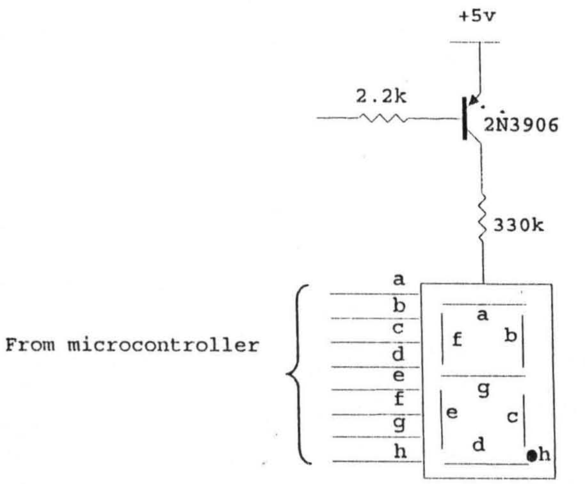

Fig 3.5 Circuit Diagram of Common Anode Display

The continuous current through each segment was fixed at 10mA. For an n-digit display, the current through each segment is computed as $I_f \times n$, where n is the number of digits in the display, I_f = forward continuous current.

For a 4-digit display and I_f of 10mA, the pulsed segment current is

Thus, $4 \times 10mA = 40mA$

The peak digit (with all LEDs ON) is thus $40mA \times 8 = 320mA$. I_C of each anode switch is thus 0.32mA. For the 2N3906 PNP transistor used, $I_C = h_{FE} = 150 - 250$. A measure h_{FE} of 220 was noted in the transistor lot.

$$I_C = h_{FE} I_B \qquad \text{.......... 3.5}$$

$$I_B = \frac{0.32}{220} = 1.45mA$$

$$R_B = \frac{V_{CC} - V_{BE}}{I_B} \qquad \text{.......... 3.6}$$

$$= \frac{5 - 0.7}{1.45} = 3K\Omega$$

The commercial value of $R_B = 2.2K\Omega$ was used in the design.

$V_{in} = V_B + V_{BE}$ 3.9

$$5 = I_B R_B + V_{BE}$$

$$R_B = \frac{5 - V_{BE}}{I_B}$$

$$R_B = \frac{5-0.7}{0.227}$$

$$\Rightarrow R_{B(max)} = \frac{4.3}{0.227} = 18.94k\Omega$$

Therefore, the actual value used should be much less than the calculated one. Thus, the base resistance R_B was chosen to be $2.2K\Omega$ to ensure saturation.

When the transistor is triggered by a negative triggering to the base, the relay is energized, thereby completing circuit for the bell and then the bell will ring which is the output of this project designed.

3.5.1 Relay as a Switch

An electromechanical relay has contacts like a manual switch, but it is controlled by external voltage instead of being operated manually. They are of two types in operation, Normally Closed (NC) and Normally Opened (NO) relays. In NC relay, the contacts are touching or closed at rest but opened by energizing the magnetic coil. While in NO relay, the contacts are not touching, they are opened, but closed by energizing the magnetic coil [8, 13, 16].

A relay provides total isolation between the triggering source applied to the terminal and the output. This total isolation is important in many digital applications, and it is a feature that certain semiconductor switches (e.g. transistors, diodes and integrated circuits) can not provides. Also, the contacts are normally rated for currents much higher than the current rating of semiconductor switches [13].

3.5.2 Protective Diode

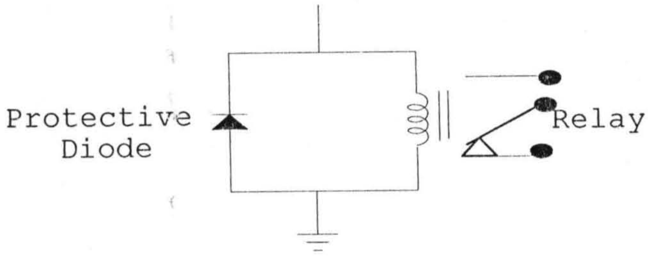

Fig 3.7 Illustration of Protective Circuit

Since the load is an ac (230-240V), relay, or solenoid (or any other device with a coil) a diode must be connected across the load to protect the transistor from the brief high voltage produced when the load is switched OFF. The diagram in fig 3.7. Shows how a protection diode is connected 'backwards' across the load, in this case a relay coil. It is known from the basic circuit courses that the current through a coil cannot change instantaneously, and in fact, the more quickly it changes, the greater the induced voltage across the coil as defined by

$$V_L=L(\frac{di_L}{dt}). \qquad 3.10$$

In this case, the rapidly changing current through the coil will develop a large voltage across the coil with the polarity shown, which will appear directly across the output of the transistor. The changes are likely that its magnitude will exceed the maximum ratings of the transistor, and the semiconductor device will be permanently damaged. The voltage across the coil will not remain at its highest switching level but will oscillate as shown until its level drops to zero as the system settles down.

This destructive action can be subdued by placing a diode across the coil as shown. During the "ON" state of the transistor, the diode is reversed-biased; it sets like an open circuit and does not affect a thing. However, when the transistor turns OFF, the voltage across the coil will reverse and forward-bias the diode, place the diode in its "ON" state. The current through the inductor established during the "ON" state of the transistor can then continue to flow through the diode eliminating the severe change in current level [14].

3.6 Complete Circuit Diagram

The fig below show the complete circuit diagram of the automatic college bell.

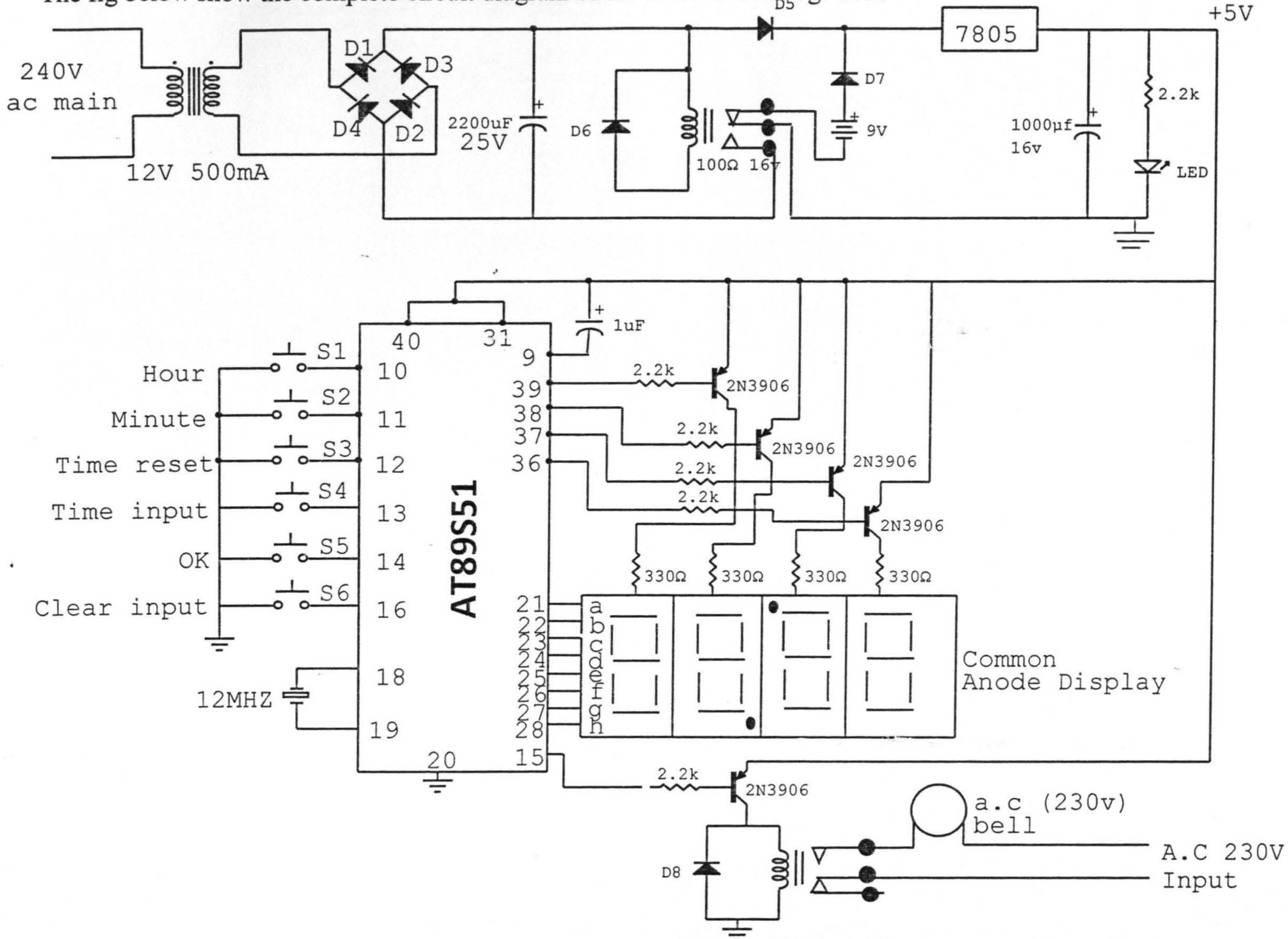

Fig 3.8 Circuit Diagram of Automatic College Bell

3.7 Construction

The complete circuit was constructed on a Vero board. The power supply, 4-digit 7-segment display circuits were constructed on a separate Vero boards and the output of each sections were taken through wires to the required points in the main board. The switches to the selection of different schedules were also constructed on a separate board and wires were drawn to the main board. This was done so that the switches could be attached separately to the casing. A plastic casing is design for this project.

The power is supplied to the main board by 7805 voltage regulator which feeds the driving circuit. The output of rectification immediately after filtration was taken to the common terminal of the relay, from it powers the transistor, whenever the relay is energized.

As a precaution during construction, IC sockets were used for the IC, so that the heat does not harm it. Also, the Vero board was checked carefully for any solder bridging.

The lists of component used are listed below.

i. Power Supply Unit

(i.) One 220V/12V, 500mA transformer

(ii.) One LED

(iii.) One 7805 regulator

(iv.) One 2,200μF, 25V capacitor

(v.) One 2.2kΩ resistor

(vi.) One 1000μf 16V capacitor

(vii.) Seven IN4007 diode

(viii.) One 6V relay

ii. Time-select Switching Unit

(i.) Six push-to-on switches (normally open)

iii. Microcontroller Unit

(i.) One AT89S51 microcontroller

(ii.) One 12MHz crystal

(iii.) One 1μF capacitor

iv. Relay Driver Unit

(i.) One 2.2kΩ resistor

(ii.) One 2N3906 NPN transistor

(iii.) One IN4007 diode

(iv.) One 6V relay

v. Common Anode Display Unit

(i.) Four 330Ω resistors

(ii.) Four seven-segment-displays

(iii.) Four 2.2kΩ resistors

(iv.) Four 2N3906 NPN transistor

vi. Output Unit

(i.) One ac (230V) electric bell

CHAPTER FOUR

TESTS, RESULTS AND DISCUSSION

4.1 Testing

An AT89S51 microcontroller is a programmable IC, which needs to be programmed to suit the design. The source code was first compiled on the notepad and then test ran on Edsim51 Simulator. Proper concentration was given to the code during compilation in order to avoid any logic errors. The hex file was then generated and transferred to the chip with the aid of a programmer.

The IC was tested on a bread board to make sure it works properly. The whole circuit was also tested on the bread board to make sure it was design correctly. During construction, each section was tested as it was built to make sure the connections were done correctly before going onto the next section. This was done by applying the correct logic signals to the ICs and observing the output.

After construction, the bell was tested for different schedules and the time was monitored with a stop-watch to know when the bell is ringing. The results obtained were tabulated in the next section

4.2 Results

After testing the bell, the following tabulated results in Table 4.1 were obtained.

Table 4.1: Results obtained from testing

S/NO.	DURATION PER PERIOD		DRIFT PER PERIOD (In seconds)
	SELECTED (In minutes)	MEASURED	
1.	50	50min 3sec	3
2.	40	40min 1sec	1
3.	30	30min 0sec	0
4.	20	20min 0sec	0

4.3 Discussion of the Results

By looking at the results obtained in section 4.2, various deductions can be made. The results are explained as follows

The drift in 50min is more than that of 40min by 2sec. While in 30min and 20min, there was no drift. This therefore, means that the higher the time schedule, the more the drift in it. This might have due to accumulation of the execution of some microsecond's instructions that were not taken into consideration during the programming. This means that the seconds' increment can be negligible since an individual timer clock varies. Thus, this is accurate enough for most applications.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The basic design of the Automatic College Bell mainly for Primary and Secondary Schools in this project remains the same though extra functions can be included. This will ring the bell at pre-scheduled times of period on each day. There are different times per period, from one School to the other. The project provides chances of inputting different time schedules for the users and after which signal is sent to the microcontroller to carry out the specific task thereby ringing the bell at a regular interval.

Therefore, from the results obtained, it can be concluded that the aim of this project work has been practically and theoretically achieved.

5.2 Possible Improvement

There are a lot of other things that can be included in the project to increase its functionality, performance and dynamism. Some of these are given below:

i. Incorporating it with an inverter to serve as the source of ac power supply.

ii. Inclusion of a rechargeable battery source for back-up in case the power fails, either from the PHCN or the incorporated inverter, so as to maintain the School periods of the day.

5.3 Recommendation

In respect to the features included in this work, it could be useful for the following recommended areas below:

i. It could be used mainly in Primary and Secondary Schools where bell play a significant role.

ii. It could be used in Industries where a specific work is to be carried out at a regular interval.

iii. It can be used in a seminar where lectures are delivered at a regular interval

iv. It can also be used for other applications where you want a sound to play at preset times.

v. Finally, it can be used in some other businesses where the bell timer plays a critical role in running the day.

REFERENCES

[1] http://www.schoolbell.com/links/themes/themes_index2.html

[2] http://www.automatictimetable.com/schooltimer.htm

[3] http://www.experiencefestival.com/bell_instrument

[4] "Bell (musical instrument)". Microsoft ® student 2008 (DVD) Redmond, WA: Microsoft Corporation, 2007

[5] http://www.en.wikipedia.org/wiki/Handbell

[6] http://www.encyclopedia.com/doc/1G1-183445763.html

[7] http://www.elecronicsforu.com/elecronicsforu/circuit/mohitcircuit.asp?p=7

[8] Thomas, L. Floyd, Electronics Fundamentals: circuits, Devices and Applications, 7th Ed. Published by Pearson Education, Upper Saddle River, New Jersey, 2007.

[9] B. L. Theraja and A. K. Theraja, A Text Book of Electrical Technology, S. Chand and Company limited, Ram Nagar, New Delhi-110055, 2003.

[10] Schuler, Charles A., Electronics, Principles and Applications, 5th Ed., Published By McGraw-Hill Companies, 1999.

[11] Holt, Charles A., Electronic Circuits: Digital and Analog, Published by John Wiley & Sons, Inc.Canada, 1978.

[12] Neamen, Donald A., Electronic Circuit Analysis and Design, 2nd Ed., Published By McGraw-Hill Companies, Inc., New York, 2001.

[13] Klertz, William, Digital Electronics With VHDL, Quartus II Version, Pearson Education Inc., Upper Saddle River, New Jersey 07458, 2006.

[14] Boylestad, R. L. & Nashelsky, L., Electronic Devices and Circuit Theory, 9th Ed., published by Dorling Kindersley (India) private limited, 2007.

[15] http://en.wikipedia.org/wiki/microcontroller

[16] Paul, Horowitz, The Art of Electronics, 2nd Ed., Published By the Press Syndicate of the University of Cambridge,

[17] J. B. Gupta, Electronic Devices and Circuits, 2nd Ed., Published by Sanjeev, Kumar Kataria FOR S. K. KATARIA & SONS, 2003.

[18] Malvino, A. P. "Malvino Electronic Principles" 6th Ed., Published by Tala McGraw-Hill Publishing Co. Ltd., Patel Nagar, New Delhi 11008, 1999.

[19] Bernard Grob & Mitchel E. S., Basic Electronics, 9th Ed., Western Wisconsin Technical College, Printed in United States of America,

[20] M. L. Anand, Electronic Principles-Devices and Circuits, 1st Ed., Published by S. chand & Company Ltd, 7361, Ram Nagar, New Delhi-110055, 2000.

[21] http://www.mikroe.com/en/books/8051book/ch2/

[22] http://pdf1.alldatasheet.com/datasheet_pdf/view/77368/ATMEL/AT89S51_24AC.html.

[23] http://www.datasheetcatalog.com/datasheets_pdf/2/N/3/9/2N3906.shhtml.

APPENDIX A: USER'S MANUAL

AUTOMATIC COLLEGE BELL

A. Power Supply

Plug in the power cable into ac main. Then turn ON the toggle switch to power the device, also plug in the ac bell. The power indicator (Red LED) glows, meaning that the device is now powered.

B. Selection of the schedule

It is considered a School has a total number of eight periods with a breakfast after the fourth period. There is ten (10) different schedules. The duration per period differs from one schedule to the other, so also the duration for breakfast and so this project enable individual to input his desired time/period which the bell will ring.

To ring this Automatic College Bell, first the keeper needs to start by inputting all the time he/she want the bell to ring. Thereafter, the Bell sound every time of the period to indicates the end of consecutive periods. The Bell circuit should now be switched off manually.

APPENDIX B: SOURCE CODE

```
RB0        EQU                  00H
RB1        EQU                  08H
display    EQU                  P2
driver     EQU                  P0
hr_key     EQU                  P3.0
min_keyEQU                      P3.1
rst_time EQU                    P3.2
input      EQU                  P3.3
ok_key     EQU                  P3.4
bell       EQU                  P3.5
clear_input  EQU                P3.6
switchesEQU                     P3
count      EQU                  10H
speed      EQU                  11H
value_1 EQU 12H
value_2 EQU                     13H
value_3 EQU                     14H
value_4 EQU                     15H
minute     EQU                  16H
hour       EQU                  17H
decimalp_1  EQU                 18H
decimalp_2  EQU                 19H
stack      EQU                  3FH
second_1     EQU                1AH
second_2     EQU                1BH
store_1  EQU                    1CH
store_2  EQU                    1DH
min_1      EQU                  1EH
min_2      EQU                  1FH
min_3      EQU                  20H
min_4      EQU                  21H
min_5      EQU                  22H
min_6      EQU                  23H
min_7      EQU                  24H
min_8      EQU                  25H
min_9      EQU                  26H
min_10   EQU                    27H
hr_1       EQU                  28H
hr_2       EQU                  29H
hr_3       EQU                  2AH
hr_4       EQU                  2BH
hr_5       EQU                  2CH
hr_6       EQU                  2DH
hr_7       EQU                  2EH
hr_8       EQU                  2FH
hr_9       EQU                  30H
hr_10      EQU                  31H
;******************************************
ORG 0000H
           JMP main
           ORG 000BH
           JMP refresh
```

```
main:
        MOV PSW, #RB0
        MOV SP, #stack
        CALL half_sec
        CALL half_sec
        CALL half_sec
        MOV driver, #0FFH
        MOV display, #0FFH
        MOV switches, #0FFH
        MOV speed, #00H
        MOV count, #00H
        MOV TMOD, #00000001B
        MOV TL0, #00H
        MOV TH0, #0FDH
        MOV IE, #82H
        CLR TF0
        SETB TR0
        MOV value_1, #01H
        MOV value_2, #02H
        MOV value_3, #00H
        MOV value_4, #00H
        MOV decimalp_1, #20H
        MOV decimalp_2, #20H
        MOV minute, #00H
        MOV hour, #0CH
        MOV second_1, #00H
        MOV second_2, #00H
        MOV R4, #00H
        CALL half_sec
;*****************************************
mainloop:
        CALL show_time
        CALL scan_key
        JMP mainloop
;*****************************************
show_time:
        CALL half_sec
        CALL blink
        CALL sec_increase
        RET
;*****************************************
half_sec:
        MOV R0, #04H
loop_1:
        MOV R1, #0F2H
        MOV R2, #00H
loop_2:
        DJNZ R2, loop_2
        DJNZ R1, loop_2
        DJNZ R0, loop_1
        RET
;*****************************************
blink:
        INC second_1
        MOV R3, second_1
        CJNE R3, #02H, change
```

```
        MOV decimalp_1, #20H
        MOV decimalp_2, #20H
        MOV second_1, #00H
        MOV R3, second_1
        RET
change:
        MOV decimalp_1, #15H
        MOV decimalp_2, #15H
        RET
;*****************************************
sec_increase:
        CJNE R3, #00H, return_1
        INC second_2
        MOV R3, second_2
        CJNE R3, #60, return_1
        MOV second_2, #00H
        CALL min_increase
        CALL hr_increase
        CALL compare
return_1:
        RET
;*****************************************
min_increase:
        INC minute
        MOV R3, minute
        CJNE R3, #60, continue_1
        MOV minute, #00H
        MOV R3, minute
continue_1:
        CALL hex_to_bcd
        MOV value_3, R5
        MOV value_4, R6
        RET
;*****************************************
hr_increase:
        CJNE R3, #00H, return_2
        INC hour
        MOV R3, hour
        CJNE R3, #24, continue_2
        MOV hour, #00H
        MOV R3, hour
continue_2:
        CALL hex_to_bcd
        MOV value_1, R5
        MOV value_2, R6
return_2:
        RET
;*****************************************
scan_key:
        JB rst_time, next_1
        JMP test_hr
next_1:
        JB input, next_3
        CJNE R4, #10, next_2
        RET
next_2:
```

```
        JMP enter_time
next_3:
        JB clear_input, next_4
        MOV R4, #00H
next_4:
        RET
;*****************************************
test_hr:
        JB hr_key, test_min
        CALL change
remain_1:
        MOV R3, #00H
        CALL hr_increase
        CALL delay
        JNB hr_key, remain_1
test_min:
        JB min_key, go_ahead
        CALL change
remain_2:
        CALL min_increase
        CALL delay
        JNB min_key, remain_2
go_ahead:
        JB ok_key, test_hr
        RET
;*****************************************
enter_time:
        INC R4
        CJNE R4, #01, proceed_1
        CALL storage
        MOV min_1, minute
        MOV hr_1, hour
pop_sub:
        MOV minute, store_1
        MOV R3, minute
        CALL continue_1
        MOV hour, store_2
        MOV R3, hour
        CALL continue_2
        RET
proceed_1:
        CJNE R4, #02, proceed_2
        CALL storage
        MOV min_2, minute
        MOV hr_2, hour
        JMP pop_sub
proceed_2:
        CJNE R4, #03, proceed_3
        CALL storage
        MOV min_3, minute
        MOV hr_3, hour
        JMP pop_sub
proceed_3:
        CJNE R4, #04, proceed_4
        CALL storage
        MOV min_4, minute
```

```
        MOV hr_4, hour
        JMP pop_sub
proceed_4:
        CJNE R4, #05, proceed_5
        CALL storage
        MOV min_5, minute
        MOV hr_5, hour
        JMP pop_sub
proceed_5:
        CJNE R4, #06, proceed_6
        CALL storage
        MOV min_6, minute
        MOV hr_6, hour
        JMP pop_sub
proceed_6:
        CJNE R4, #07, proceed_7
        CALL storage
        MOV min_7, minute
        MOV hr_7, hour
        JMP pop_sub
proceed_7:
        CJNE R4, #08, proceed_8
        CALL storage
        MOV min_8, minute
        MOV hr_8, hour
        JMP pop_sub
proceed_8:
        CJNE R4, #09, proceed_9
        CALL storage
        MOV min_9, minute
        MOV hr_9, hour
        JMP pop_sub
proceed_9:
        CJNE R4, #10, go_back
        CALL storage
        MOV min_10, minute
        MOV hr_10, hour
        JMP pop_sub
go_back:
        RET
;*******************************************
storage:
        MOV store_1, minute
        MOV store_2, hour
        CALL test_hr
        RET
;*******************************************
compare:
        CJNE R4, #00H, here_1
        JMP ret1
ere_1:
        CJNE R4, #01H, here_2
        JMP there_9
ere_2:
        CJNE R4, #02H, here_3
        JMP there_8
```

```
here_3:
        CJNE R4, #03H, here_4
        JMP there_7
here_4:
        CJNE R4, #04H, here_5
        JMP there_6
here_5:
        CJNE R4, #05H, here_6
        JMP there_5
here_6:
        CJNE R4, #06H, here_7
        JMP there_4
here_7:
        CJNE R4, #07H, here_8
        JMP there_3
here_8:
        CJNE R4, #08H, here_9
        JMP there_2
here_9:
        CJNE R4, #09H, here_10
        JMP there_1
here_10:
        MOV A, hour
        CJNE A, hr_10, there_1
        MOV A, minute
        CJNE A, min_10, there_1
        JMP ring_bell
there_1:
        MOV A, hour
        CJNE A, hr_9, there_2
        MOV A, minute
        CJNE A, min_9, there_2
        JMP ring_bell
there_2:
        MOV A, hour
        CJNE A, hr_8, there_3
        MOV A, minute
        CJNE A, min_8, there_3
        JMP ring_bell
there_3:
        MOV A, hour
        CJNE A, hr_7, there_4
        MOV A, minute
        CJNE A, min_7, there_4
        JMP ring_bell
there_4:
        MOV A, hour
        CJNE A, hr_6, there_5
        MOV A, minute
        CJNE A, min_6, there_5
        JMP ring_bell
there_5:
        MOV A, hour
        CJNE A, hr_5, there_6
        MOV A, minute
        CJNE A, min_5, there_6
```

```
        JMP ring_bell
there_6:
        MOV A, hour
        CJNE A, hr_4, there_7
        MOV A, minute
        CJNE A, min_4, there_7
        JMP ring_bell
there_7:
        MOV A, hour
        CJNE A, hr_3, there_8
        MOV A, minute
        CJNE A, min_3, there_8
        JMP ring_bell
there_8:
        MOV A, hour
        CJNE A, hr_2, there_9
        MOV A, minute
        CJNE A, min_2, there_9
        JMP ring_bell
there_9:
        MOV A, hour
        CJNE A, hr_1, ret1
        MOV A, minute
        CJNE A, min_1, ret1
ring_bell:
        CLR bell
        MOV R7, #20
loop_again:
        CALL show_time
        DJNZ R7, loop_again
        SETB bell
ret1:
        RET
;******************************************
delay:
        MOV R0, #02H
loop_3:
        MOV R1, #0F3H
        MOV R2, #00H
loop_4:
        DJNZ R2, loop_4
        DJNZ R1, loop_4
        DJNZ R0, loop_3
        RET
;******************************************
hex_to_bcd:
        MOV B, #10
        MOV A, R3
        DIV AB
        MOV R6, B
        MOV B, #10
        DIV AB
        MOV R5, B
        RET
;******************************************
refresh:
```

```
        PUSH PSW
        MOV PSW, #RB1
        PUSH ACC
        INC count
        MOV R7, count
QA_1:
        CJNE R7, #01H, QA_2
        MOV speed, value_1
        MOV driver, #0FEH
        CALL DISP
        AJMP down
QA_2:
        CJNE R7, #02H, QA_2_2
        MOV speed, value_2
        MOV driver, #0FDH
        CALL DISP
        AJMP down
QA_2_2:
        CJNE R7, #03H, QA_3
        MOV speed, decimalp_1
        CALL DISP
        AJMP down
QA_3:
        CJNE R7, #04H, QA_3_3
        MOV speed, decimalp_2
        MOV driver, #0FBH
        CALL DISP
        AJMP down
QA_3_3:
        CJNE R7, #05H, QA_4
        MOV speed, value_3
        CALL DISP
        AJMP down
QA_4:
        CJNE R7, #06H, QA_5
        MOV speed, value_4
        MOV driver, #0F7H
        CALL DISP
        AJMP down
QA_5:
        MOV count, #01H
        MOV R7, count
        AJMP QA_1
down:
        MOV TL0, #0FFH
        MOV TH0, #0F0H
        POP ACC
        POP PSW
        RETI
;*****************************************
DISP:
        MOV R6, speed
        CJNE R6, #00H, aas_1
        MOV display, #0C0H
        RET
aas_1:
```

```
        CJNE R6, #01H, aas_2
        MOV display, #0F9H
        RET
aas_2:
        CJNE R6, #02H, aas_3
        MOV display, #0A4H
        RET
aas_3:
        CJNE R6, #03H, aas_4
        MOV display, #0B0H
        RET
aas_4:
        CJNE R6, #04H, aas_5
        MOV display, #99H
        RET
aas_5:
        CJNE R6, #05H, aas_6
        MOV display, #92H
        RET
aas_6:
        CJNE R6, #06H, aas_7
        MOV display, #82H
        RET
aas_7:
        CJNE R6, #07H, aas_8
        MOV display, #0D8H
        RET
aas_8:
        CJNE R6, #08H, aas_9
        MOV display, #80H
        RET
aas_9:
        CJNE R6, #09H, aas_10
        MOV display, #90H
        RET
aas_10:
        CJNE R6, #15H, aas_11
        MOV display, #7FH
        RET
aas_11:
        CJNE R6, #20H, aas_12
        MOV display, #0FFH
        RET
aas_12:
        MOV speed, #00H
        AJMP DISP
        END
```