**1. INTRODUCTION**

**I**n today’s life, everyone gives importance to time. Time does not wait for anybody. Everything should be performed in time & accurately. Now a day’s school/college bells are manually operated. Hence there is a big question of accuracy. Also there is necessity of manpower and money. Hence here we should use automatic control system, which saves our manpower and money & also highest accuracy. Hence we have selected the project.

**What is our System?**

In market there many digital clocks available with bells but rings only at specific time. For e.g. Alarm Clock and some bells that ring after some time intervals and that cannot stop after specific time. For e.g. Musical Clock But all these limitation have been removed by our project. It rings only according to our college time table.

**O**ur Project takes over the task of Ringing of the Bell in Colleges. It replaces the Manual Switching of the Bell in the College. It has an Inbuilt Real Time Clock (DS1307 /DS 12c887) which tracks over the Real Time. When this time equals to the Bell Ringing time, then the Relay for the Bell is switched on. The Bell Ringing time can be edited at any Time, so that it can be used at Normal Class Timings as well as Exam Times. The Real Time Clock is displayed on LCD display. The Microcontroller AT89S52 is used to control all the Functions, it get the time through the keypad and store it in its Memory. And when the Real time and Bell time get equal then the Bell is switched on for a predetermined time.

** **

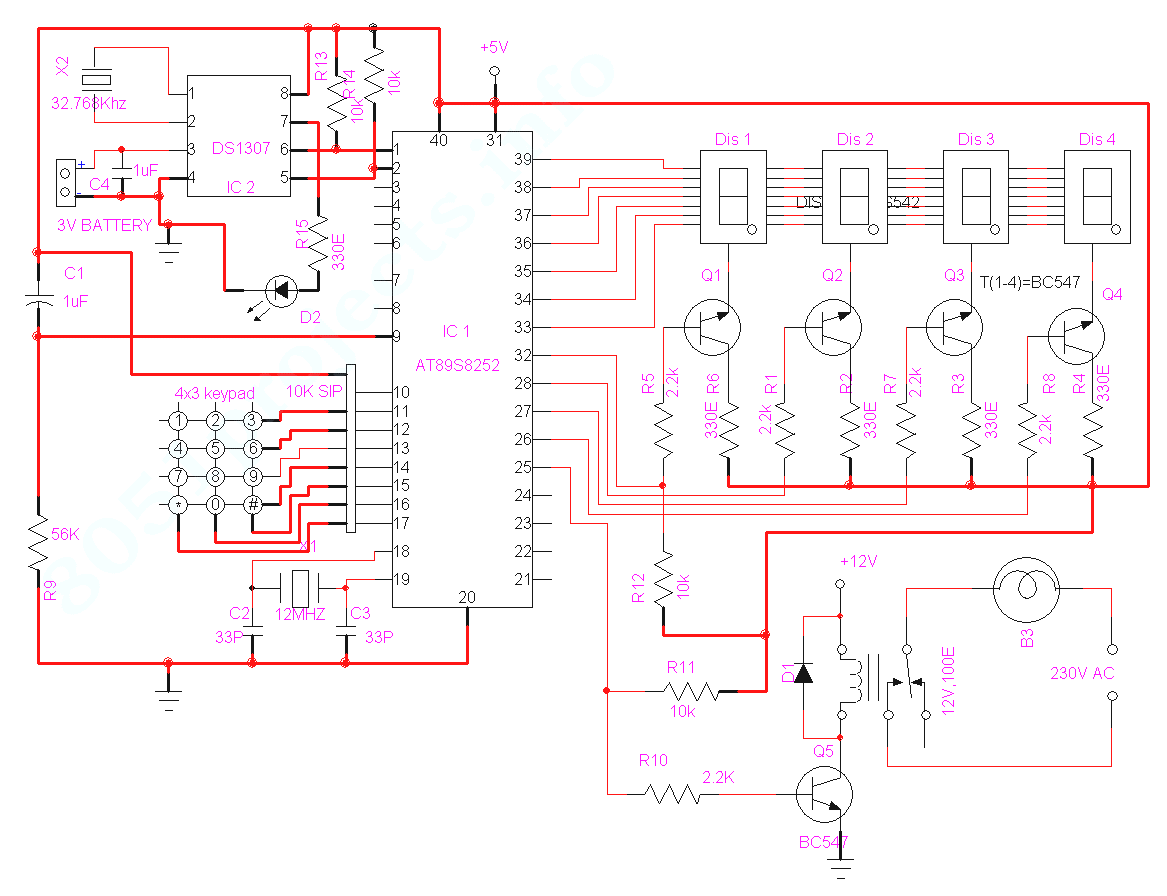
**Figure** 1.1*Conventional Bell* **Figure** 1.2*Manually operated College Bell*



**Figure** 1.3 *Automatic College Bell*

**2. CIRCUIT DESCRIPTION**

**2.1. Circuit DIAGRAM:-**

****

**Figure** 2.1.1 *Circuit Diagram of Automatic College Bell*

**2.2. FUNCTION of Circuit:-**

In the circuit shown above, we provide 220V A.C. power supply to the “Step-Down Transformer” which converts 220V A.C. into 12V A.C. (i.e. stepped down the power supply). Now this 12V A.C. is converted into 12V D.C. with the help of “Full Wave Rectifier” which consists of 2 Diodes & 2 Condensers [a filter capacitor (1000µF)].

Two different voltage levels are required for our circuit – One is 12V D.C. to operate relay switch. Second is 5V D.C. supply to operate microcontroller “AT89S8252”. For this purpose we will use voltage regulator “LM7805” which can take 8V -25V as I/P & provide 5V constant voltage.

Here we have used “Atmel AT89S8252” microcontroller to control various timing of the ringing. Here we also use a “12MHz Crystal” which will provide the microcontroller a reference time.

We have used “Assembly Language” to program this microcontroller and we have also used a microcontroller programmer. We have used different types of capacitors and resistors in this circuit. We have used two 33pF capacitor which are acting as a High Pass Filter [H.P.F.]. The 10KΩ resistor is used for RESET circuit to provide negative potential to RESET pin of microcontroller.

We have used IC DS 1307 which is a low-power clock/calendar with 56 bytes of

Battery-backed SRAM. It uses an external 32.768 kHz crystal. The oscillator circuit does not require any external resistors or capacitors to operate. The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed.

We have used four seven segment display for the displaying the real time. Here BC 547 is used for the amplification process.

The microcontroller can operate on 5V and 10mA current maximum but we have to operate 12V relay switch which consume more than 100A current. So, we have to amplify this current and voltage. For this purpose we are using transistor.

**2.3. OPERATION:-**

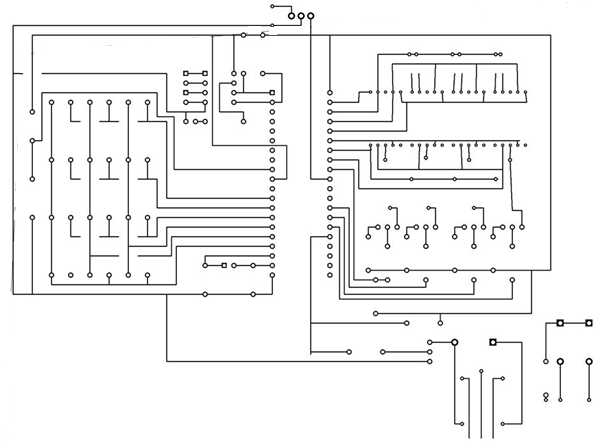
* Switch ON the power
* In Display the real time will display.
* It has an Inbuilt Real Time Clock (DS1307 /DS 12c887) which tracks over the Real Time. When this time equals to the Bell Ringing time, then the Relay for the Bell is switched on.
* If one want to change the belling time.

Input the desire time from the keypad provided.

At the set time the buzzer will ring.

* For changing the input time press \* followed by # on the keypad and set the belling time.
* One can set many ringing time at a time.
* The input time must be set with respect of RTC.

**2.4. PCB LAYOUT:-**

****

**Figure** 2.4.1 *PCB Layout of Automatic College Bell*

**Figure** 2.4.2 *PCB Rear Side* **Figure** 2.4.3 *PCB Front Side*

**3. SOFTWARE PROGRAMMING**

ASSEMBLER AUTOMATIC COLLEGE BELL

RB0 EQU 000H ; Select Register Bank 0

RB1 EQU 008H ; Select Register Bank 1 ...poke to PSW to use

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

; PORT DECLERATION

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

SDA EQU P1.1 ;SDA=PIN5

SCL EQU P1.0 ;SCL=PIN6

DS1307W EQU 0D0H ; SLAVE ADDRESS 1101 000 + 0 TO WRITE

DS1307R EQU 0D1H ; SLAVE ADDRESS 1101 000 + 1 TO READ

KEYS EQU P3

ROW1 EQU P3.1

ROW2 EQU P3.2

ROW3 EQU P3.3

ROW4 EQU P3.4

COL1 EQU P3.5

COL2 EQU P3.6

COL3 EQU P3.7

DIS\_A EQU P0.2

DIS\_B EQU P0.3

DIS\_C EQU P0.4

DIS\_D EQU P0.6

DIS\_E EQU P0.5

DIS\_F EQU P0.1

DIS\_G EQU P0.0

DIS1 EQU P0.7

DIS2 EQU P2.7

DIS3 EQU P2.6

DIS4 EQU P2.5

RELAY EQU P2.4

WMCON DATA 96h ; watchdog and memory control register

EEMEN EQU 00001000b ; EEPROM access enable bit

EEMWE EQU 00010000b ; EEPROM write enable bit

WDTRST EQU 00000010b ; EEPROM RDY/BSY bit

DPS EQU 00000100b ; data pointer select bit

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

DSEG ; This is internal data memory

ORG 20H ; Bit adressable memory

FLAGS DATA 20H

LASTREAD BIT FLAGS.0

SQW BIT FLAGS.4

ACK BIT FLAGS.5

BUS\_FLT BIT FLAGS.6

\_2W\_BUSY BIT FLAGS.7

CANCEL BIT FLAGS.1

CANCEL1 BIT FLAGS.2

ALARM BIT FLAGS.3

BITCNT DATA 21H

BYTECNT DATA 22H

SECS DATA 24H ; ' SECONDS STORAGE RAM

MINS DATA 25H ; ' MINUTES ' '

HRS DATA 26H ; ' HOURS ' '

DAY DATA 27H ; ' DAY ' '

DATE1 DATA 28H ; ' DATE ' '

MONTH DATA 29H ; ' MONTH ' '

YEAR DATA 2AH ; ' YEAR ' '

CONTROL DATA 2BH ; FOR STORAGE OF CONTROL REGISTER WHEN READ.

ALM\_HOUR DATA 2CH ; INTERNAL (ALARM HOURS) STORAGE.

ALM\_MIN DATA 2DH ; INTERNAL (ALARM MINUTES) STORAGE.

ALM\_CNTRL DATA 2EH ; INTERNAL STORAGE FOR ALARM (ON) TIME.

COUNT DATA 2FH

SPEED DATA 30H

VALUE\_1 DATA 31H

VALUE\_2 DATA 32H

VALUE\_3 DATA 33H

VALUE\_4 DATA 34H

NUMBER1 DATA 35H ;temp to store dialled number

KBELL DATA 36H

NUMB1 DATA 37H ;Temp Reg to store pressed Keys

NUMB2 DATA 38H ;Temp Reg to store pressed Keys

NUMB3 DATA 39H ;Temp Reg to store pressed Keys

NUMB4 DATA 3AH ;Temp Reg to store pressed Keys

KEY DATA 3BH

TIM DATA 3CH

STACK DATA 3FH

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

; \*\*\*MACRO'S\*\*\*

SCL\_HIGH MACRO

SETB SCL ; SET SCL HIGH

JNB SCL,$ ; LOOP UNTIL STRONG 1 ON SCL

ENDM

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

CSEG AT 0 ; RESET VECTOR

;---------==========----------==========---------=========---------

; PROCESSOR INTERRUPT AND RESET VECTORS

;---------==========----------==========---------=========---------

ORG 00H ; Reset

JMP MAIN

ORG 000BH ;Timer Interrupt0

JMP REFRESH

ORG 001BH ;Timer Interrupt1

JMP RELAY\_TIMER

;---------==========----------==========---------=========---------

; Main routine. Program execution starts here.

;---------==========----------==========---------=========---------

MAIN:

MOV PSW,#RB0 ; Select register bank 0

MOV SP,STACK

CLR RELAY ;Switch OFF relay

MOV SPEED,#00H

MOV COUNT,#00H

MOV KBELL,#00H

CLR ALARM

MOV VALUE\_1,#15H

MOV VALUE\_2,#15H

MOV VALUE\_3,#15H

MOV VALUE\_4,#15H

CLR DIS1

CLR DIS2

CLR DIS3

CLR DIS4

MOV TMOD,#01H ;enable timer0 for scanning

MOV TL0,#00H

MOV TH0,#0FDH

SETB ET0

SETB EA

SETB TR0 ;Start the Timer

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; INITILIZE RTC

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SETB SDA ; ENSURE SDA HIGH

SCL\_HIGH ; ENSURE SCL HIGH

CLR ACK ; CLEAR STATUS FLAGS

CLR BUS\_FLT

CLR \_2W\_BUSY

CLR SQW

CALL OSC\_CONTROL ;Initilize the RTC

ACALL SQW\_CONTROL\_1HZ

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; CHECK FOR ENTER THE TIME

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LCALL SEND\_START ; SEND 2WIRE START CONDITION

MOV A,#DS1307W ; SEND DS1307 WRITE COMMAND

LCALL SEND\_BYTE

MOV A,#08H ; SET POINTER TO REG 08H ON DS1307

LCALL SEND\_BYTE

LCALL SEND\_STOP ; SEND STOP CONDITION

LCALL SEND\_START ; SEND START CONDITION

MOV A,#DS1307R ; SEND DS1307 READ COMMAND

LCALL SEND\_BYTE

LCALL READ\_BYTE ; READ A BYTE OF DATA

MOV R1,A

LCALL SEND\_STOP ; SEND 2WIRE STOP CONDITION

MOV NUMBER1,#01H

CJNE A,#0AAH,KEYBOARD1

AJMP START\_PROGRAM

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; KEYBOARD ROUTINE

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

KEYBOARD1:

MOV KBELL,#0FFH

KEYBOARD:

MOV KEY,#00H

SETB COL1

SETB COL2

SETB COL3

K11: CLR ROW1

CLR ROW2

CLR ROW3

CLR ROW4

MOV A,KEYS

ANL A,#11100000B

CJNE A,#11100000B,K11 ;check till all keys released

K2: ACALL DEALAY ;call 20 msec delay

MOV A,KEYS ;see if any key is pressed

ANL A,#11100000B ;mask unused bits

CJNE A,#11100000B,OVER ;key pressed, await closure

SJMP K2

OVER: ACALL DEALAY

MOV A,KEYS

ANL A,#11100000B

CJNE A,#11100000B,OVER1

SJMP K2

OVER1: MOV A,KEYS

ORL A,#11111110B

MOV KEYS,A

CLR ROW1

MOV A,KEYS

ANL A,#11100000B

CJNE A,#11100000B,ROW\_1

MOV A,KEYS

ORL A,#11111110B

MOV KEYS,A

CLR ROW2

MOV A,KEYS

ANL A,#11100000B

CJNE A,#11100000B,ROW\_2

MOV A,KEYS

ORL A,#11111110B

MOV KEYS,A

CLR ROW3

MOV A,KEYS

ANL A,#11100000B

CJNE A,#11100000B,ROW\_3

MOV A,KEYS

ORL A,#11111110B

MOV KEYS,A

CLR ROW4

MOV A,KEYS

ANL A,#11100000B

CJNE A,#11100000B,ROW\_4

LJMP K2

ROW\_1: RLC A

JC MAT1

MOV KEY,#01H

AJMP K1

MAT1: RLC A

JC MAT2

MOV KEY,#02H

AJMP K1

MAT2: RLC A

JC K1

MOV KEY,#03H

AJMP K1

ROW\_2: RLC A

JC MAT3

MOV KEY,#04H

AJMP K1

MAT3: RLC A

JC MAT4

MOV KEY,#05H

AJMP K1

MAT4: RLC A

JC K1

MOV KEY,#06H

AJMP K1

ROW\_3: RLC A

JC MAT5

MOV KEY,#07H

AJMP K1

MAT5: RLC A

JC MAT6

MOV KEY,#08H

AJMP K1

MAT6: RLC A

JC K1

MOV KEY,#09H

AJMP K1

ROW\_4: RLC A

JC MAT7

MOV KEY,#10H ;for \*

AJMP K1

MAT7: RLC A

JC MAT8

MOV KEY,#00H ;for 0

AJMP K1

MAT8: RLC A

JC K1

MOV KEY,#12H ;for =

K1:

MOV A,KBELL

CJNE A,#0FFH,KB\_RET1

MOV A,KEY

CJNE A,#10H,CXCX0 ;Key to Erase last dislled NUMBER1

MOV KEY,#00H

MOV NUMBER1,#01H

MOV VALUE\_1,#15H

MOV VALUE\_2,#15H

MOV VALUE\_3,#15H

MOV VALUE\_4,#15H

AJMP KEYBOARD

KB\_RET1: JMP KB\_RET

CXCX0: MOV A,NUMBER1

CJNE A,#01H,CXCX1

MOV A,KEY

CLR C

SUBB A,#03H ; Chk Key Pressed 0,1

JNC CXCX5

MOV A,KEY

INC NUMBER1

MOV NUMB1,KEY

MOV VALUE\_1,KEY

AJMP KEYBOARD

CXCX1: CJNE A,#02H,CXCX2

MOV A,NUMB1

CJNE A,#02,JKJL

MOV A,KEY

CLR C

SUBB A,#04H ; Chk Key Pressed 0,1,2,3

JNC CXCX5

JKJL: MOV A,KEY

CLR C

SUBB A,#10H ; Chk Key Pressed 0,1...8,9

JNC CXCX5

INC NUMBER1

MOV NUMB2,KEY

MOV VALUE\_2,KEY

AJMP KEYBOARD

CXCX2: CJNE A,#03H,CXCX3

MOV A,KEY

CLR C

SUBB A,#06H ; Chk Key Pressed 0,1...,5

JNC CXCX5

INC NUMBER1

MOV NUMB3,KEY

MOV VALUE\_3,KEY

AJMP KEYBOARD

CXCX3: CJNE A,#04H,CXCX4

MOV A,KEY

CLR C

SUBB A,#10H ; Chk Key Pressed 0,1,....,8,9

JNC CXCX5

INC NUMBER1

MOV NUMB4,KEY

MOV VALUE\_4,KEY

CXCX5: AJMP KEYBOARD

CXCX4: CJNE A,#05H,CXCX5

MOV A,KEY

CJNE A,#12H,CXCX5 ;Key to OK TIME

CALL FLASHING

MOV KBELL,#00H

MOV A,NUMB1

SWAP A

ORL A,NUMB2

MOV NUMB2,A

MOV A,NUMB3

SWAP A

ORL A,NUMB4

MOV NUMB4,A

;(((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

; STORE THE TIME TO RTC CHIP

;(((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

LCALL SEND\_START ; SEND 2WIRE START CONDITION

MOV A,#DS1307W ; LOAD DS1307 WRITE COMMAND

LCALL SEND\_BYTE ; SEND WRITE COMMAND

MOV A,#08H ; SET DS1307 DATA POINTER TO BEGINNING

LCALL SEND\_BYTE ; OF USER RAM 08H

MOV A,#0AAH ; WRITE BYTE TO ENTIRE RAM SPACE

LCALL SEND\_BYTE

LCALL SEND\_STOP ; SEND 2WIRE STOP CONTION

LCALL SEND\_START ; SEND 2WIRE START CONDITION

MOV A,#DS1307W ; LOAD DS1307 WRITE COMMAND

LCALL SEND\_BYTE ; SEND WRITE COMMAND

MOV A,#01H ; SET DS1307 DATA POINTER TO BEGINNING

LCALL SEND\_BYTE ; OF 00H

MOV A,NUMB4 ; SET DS1307 DATA POINTER TO BEGINNING

LCALL SEND\_BYTE ; OF 00H

MOV A,NUMB2

CLR ACC.6

LCALL SEND\_BYTE

LCALL SEND\_STOP ; SEND 2WIRE STOP CONTION

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; MAIN PROGRAM

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

START\_PROGRAM:

CALL READ\_CLOCK

MOV R1,#25H ;GET MIN AND DISPLAY

MOV A,@R1

ANL A,#0FH

MOV VALUE\_4,A

MOV R1,#25H

MOV A,@R1

ANL A,#0F0H

SWAP A

MOV VALUE\_3,A

MOV R1,#26H ;GET HOUR AND DISPLAY

MOV A,@R1

CLR C

SUBB A,#12H

JNC CCX

MOV A,@R1

CCX:

CJNE A,#00H,HHGH

MOV A,#12H

HHGH:

ANL A,#0FH

MOV VALUE\_2,A

MOV R1,#26H

MOV A,@R1

CLR C

SUBB A,#12H

JNC CCX1

MOV A,@R1

CCX1:

CJNE A,#00H,HHGH1

MOV A,#12H

HHGH1:

ANL A,#0F0H

SWAP A

MOV VALUE\_1,A

CALL LOAD\_ALRM

CLR ROW4

SETB COL2

JB COL2,NEXT1

;(((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

; EMERGENCY BELL

;(((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

SETB RELAY

JNB COL2,$

CLR RELAY

AJMP START\_PROGRAM

;(((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

;(((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

;

;(((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

NEXT1: CLR ROW4

SETB COL3

JB COL3,START\_PROGRAM

CALL SQW\_CONTROL\_32KHZ

MOV NUMBER1,#01H

SETB CANCEL

SETB CANCEL1

MOV DPTR,#0001H

START\_PROG:

ORL WMCON, #EEMEN ; enable EEPROM accesses

MOVX A,@DPTR

CJNE A,#0FFH,TFT1

MOV VALUE\_1,#16H

MOV VALUE\_2,#16H

AJMP TFT3

TFT1: MOV R1,A ;GET MIN AND DISPLAY

ANL A,#0FH

MOV VALUE\_2,A

MOV A,R1

ANL A,#0F0H

SWAP A

MOV VALUE\_1,A

TFT3: INC DPTR

MOVX A,@DPTR

CJNE A,#0FFH,TFT2

XRL WMCON, #EEMEN ; disable EEPROM accesses

MOV VALUE\_3,#16H

MOV VALUE\_4,#16H

JMP KEYBOARD

TFT2:

MOV R1,A

ANL A,#0FH

MOV VALUE\_4,A

MOV A,R1

ANL A,#0F0H

SWAP A

MOV VALUE\_3,A

JMP KEYBOARD

START\_PM:

CLR ROW4

SETB COL3

JNB COL3,$

CALL DEC\_DPTR ;store the count of timings

CALL DEC\_DPTR

MOV A,DPL

MOV DPTR,#0100H

MOV WMCON,#18H

MOVX @DPTR,A

CZTHD:

MOV A,WMCON ;Check for eeprom finished or not

JNB ACC.1,CZTHD

MOV WMCON,#08H

CALL SQW\_CONTROL\_1HZ

AJMP START\_PROGRAM

;(((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

; CHECK FOR TIME IS EQUAL

;(((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

LOAD\_ALRM:

MOV DPTR,#0100H

ORL WMCON, #EEMEN ; enable EEPROM accesses

MOVX A,@DPTR

MOV B,#02H

DIV AB

MOV R5,A

MOV DPTR,#0001H

REPEAT:

MOVX A,@DPTR

MOV ALM\_HOUR,A

INC DPTR

MOVX A,@DPTR

MOV ALM\_MIN,A

INC DPTR

MOV A,HRS

CJNE A,ALM\_HOUR,CHKK

MOV A,MINS

CJNE A,ALM\_MIN,CHKK

MOV A,SECS

ANL A,#01111111B

MOV SECS,A

MOV A,#00H

CJNE A,SECS,CHKK

;Time Is Equal

JB ALARM,CHKK

ORL TMOD,#10H ;ENABLE TIMER 0

MOV TL1,#08H

MOV TH1,#01H

SETB ET1

MOV TIM,#100

SETB TR1

SETB RELAY

SETB ALARM

CHKK: DJNZ R5,REPEAT

XRL WMCON, #EEMEN ; disable EEPROM accesses

RET

;(((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

KB\_RET:

MOV A,KEY

CJNE A,#10H,CAXCX0 ;Key to Erase last dislled NUMBER1

JB CANCEL,START\_PM1

MOV KEY,#00H

MOV NUMBER1,#01H

MOV VALUE\_1,#15H

MOV VALUE\_2,#15H

MOV VALUE\_3,#15H

MOV VALUE\_4,#15H

SETB CANCEL

CLR CANCEL1

AJMP KEYBOARD

START\_PM1:

AJMP START\_PM

CAXCX0:

CJNE A,#12H,CAXX5

CLR CANCEL

AJMP CAXCX5

CAXX5:

MOV A,NUMBER1

CJNE A,#01H,CAXCX1

MOV A,KEY

CLR C

SUBB A,#03H ; Chk Key Pressed 0,1,2

JNC CAXCX5

MOV A,KEY

INC NUMBER1

MOV NUMB1,KEY

MOV VALUE\_1,KEY

CLR CANCEL

AJMP KEYBOARD

CAXCX1: CJNE A,#02H,CAXCX2

MOV A,NUMB1

CJNE A,#02,JAKJL

MOV A,KEY

CLR C

SUBB A,#04H ; Chk Key Pressed 0,1,2,3

JNC CAXCX5

JAKJL: MOV A,KEY

CLR C

SUBB A,#10H ; Chk Key Pressed 0,1...8,9

JNC CAXCX5

INC NUMBER1

MOV NUMB2,KEY

MOV VALUE\_2,KEY

CLR CANCEL

AJMP KEYBOARD

CAXCX2: CJNE A,#03H,CAXCX3

MOV A,KEY

CLR C

SUBB A,#06H ; Chk Key Pressed 0,1...,5

JNC CAXCX5

INC NUMBER1

MOV NUMB3,KEY

MOV VALUE\_3,KEY

CLR CANCEL

AJMP KEYBOARD

CAXCX3: CJNE A,#04H,CAXCX4

MOV A,KEY

CLR C

SUBB A,#10H ; Chk Key Pressed 0,1,....,8,9

JNC CAXCX5

INC NUMBER1

MOV NUMB4,KEY

MOV VALUE\_4,KEY

CLR CANCEL

SETB CANCEL1

CAXCX4:

AJMP KEYBOARD

CAXCX5:

JNB CANCEL1,CAXCX4

CALL DEC\_DPTR

MOV A,VALUE\_1

SWAP A

ORL A,VALUE\_2

MOV NUMB2,A

MOV A,VALUE\_3

SWAP A

ORL A,VALUE\_4

MOV NUMB4,A

MOV WMCON,#18H

MOV A,NUMB2

MOVX @DPTR,A

CTHD: MOV A,WMCON ;Check for eeprom finished or not

JNB ACC.1,CTHD

INC DPTR

MOV A,NUMB4

MOVX @DPTR,A

CTTHD: MOV A,WMCON ;Check for eeprom finished or not

JNB ACC.1,CTTHD

INC DPTR

MOV WMCON,#08H ; DISable EEPROM WRITE

AJMP START\_PROG

;(((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

DEALAY:

PUSH ACC

MOV R1,#20

REPP2: MOV A,#0A6H

MD\_OLP:

INC A

NOP

NOP

NOP

NOP

NOP

NOP

NOP

NOP

JNZ MD\_OLP

NOP

DJNZ R1,REPP2

POP ACC

RET

;((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

DEC\_DPTR:

XCH A,DPL ;Exchange A for DPL

DEC A ;Decrement A (which is DPL)

CJNE A,#0FFh,\_dec\_dptr2 ;If A (DPL) is not #0FFh, continue normally

DEC DPH ;If A=FFh, we need to decrement DPH

\_dec\_dptr2:

XCH A,DPL ;Exchange A for DPL (thus saving DPL and restoring A)

RET

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; DELAY TIMER FOR BELL

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

RELAY\_TIMER:

DJNZ TIM,GAHJ

CLR TR1

CLR RELAY

CLR ALARM

RETI

GAHJ: MOV TL1,#08H

MOV TH1,#01H

SETB TR1

RETI

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; SUB SETS THE DS1307 OSCILLATOR

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

OSC\_CONTROL:

ACALL SEND\_START ; GENERATE START CONDITION

MOV A,#DS1307W ; 1101 0000 ADDRESS + WRITE-BIT

ACALL SEND\_BYTE ; SEND BYTE TO 1307

MOV A,#00H ; ADDRESS BYTE TO REGISTER 00H

ACALL SEND\_BYTE ; SECONDS REGISTER, ALWAYS LEAVE

SETB LASTREAD ; REG 00H-BIT #7 = 0 (LOW)

ACALL SEND\_STOP ; IF REG 00H-BIT #7 = 1 CLOCK

ACALL SEND\_START ; OSCILLATOR IS OFF.

MOV A,#DS1307R ; 1101 0001 ADDRESS + READ-BIT

ACALL SEND\_BYTE ;

ACALL READ\_BYTE ; READ A BYTE FROM THE 1307

CLR ACC.7 ; CLEAR REG 00H-BIT #7 TO ENABLE

OSC\_SET: ; OSCILLATOR.

PUSH ACC ; SAVE ON STACK

ACALL SEND\_STOP ;

ACALL SEND\_START ;

MOV A,#DS1307W ; SETUP TO WRITE

ACALL SEND\_BYTE ;

MOV A,#00H ; REGISTER 00H ADDRESS

ACALL SEND\_BYTE ;

POP ACC ; GET DATA TO START OSCILLATOR

ACALL SEND\_BYTE ; SEND IT

ACALL SEND\_STOP

RET

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; THIS SUB CONTROLS THE SQW OUTPUT 1HZ

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SQW\_CONTROL\_1HZ:

LCALL SEND\_START ; SEND START CONDITION

MOV A,#DS1307W ; SET POINTER TO REG 07H ON

; DS1307

LCALL SEND\_BYTE

MOV A,#07H

LCALL SEND\_BYTE

MOV A,#90H ; SQW/OUT ON AT 1HZ

JNB SQW,SQW\_SET ; JUMP IF SQW BIT IS ACTIVE

MOV A,#80H ; TURN SQW/OUT OFF – OFF HIGH

SQW\_SET:

LCALL SEND\_BYTE

LCALL SEND\_STOP

RET

;–––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––

;

; THIS SUB CONTROLS THE SQW OUTPUT 32KHZ

;

;–––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––

SQW\_CONTROL\_32KHZ:

LCALL SEND\_START ; SEND START CONDITION

MOV A,#DS1307W ; SET POINTER TO REG 07H ON DS1307

LCALL SEND\_BYTE

MOV A,#07H

LCALL SEND\_BYTE

MOV A,#93H ; SQW/OUT ON AT 1HZ

JNB SQW,SQW\_SET3 ; JUMP IF SQW BIT IS ACTIVE

MOV A,#80H ; TURN SQW/OUT OFF – OFF HIGH

SQW\_SET3:

LCALL SEND\_BYTE

LCALL SEND\_STOP

RET

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; THIS SUB READS ONE BYTE OF DATA FROM THE DS1307

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

READ\_BYTE:

MOV BITCNT,#08H; SET COUNTER FOR 8-BITS DATA

MOV A,#00H

SETB SDA ; SET SDA HIGH TO ENSURE LINE

; FREE

READ\_BITS:

SCL\_HIGH ; TRANSITION SCL LOW-TO-HIGH

MOV C,SDA ; MOVE DATA BIT INTO CARRY

RLC A ; ROTATE CARRY-BIT INTO ACC.0

CLR SCL ; TRANSITION SCL HIGH-TO-LOW

DJNZ BITCNT,READ\_BITS

; LOOP FOR 8-BITS

JB LASTREAD,ACKN

; CHECK TO SEE IF THIS IS

; THE LAST READ

CLR SDA ; IF NOT LAST READ SEND ACK-BIT

ACKN:

SCL\_HIGH ; PULSE SCL TO TRANSMIT ACKNOWLEDGE

CLR SCL ; OR NOT ACKNOWLEDGE BIT

RET

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; SUB SENDS START CONDITION

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SEND\_START:

SETB \_2W\_BUSY ; INDICATE THAT 2-WIRE

CLR ACK ; OPERATION IS IN PROGRESS

CLR BUS\_FLT ; CLEAR STATUS FLAGS

JNB SCL,FAULT

JNB SDA,FAULT

SETB SDA ; BEGIN START CODITION

SCL\_HIGH

CLR SDA

ACALL DEELAY

CLR SCL

RET

FAULT:

SETB BUS\_FLT

RET

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; SUB SENDS STOP CONDITION

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SEND\_STOP:

CLR SDA

SCL\_HIGH

SETB SDA

CLR \_2W\_BUSY

RET

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; SUB DELAYS THE BUS

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DEELAY:

NOP ; DELAY FOR BUS TIMING

RET

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; THIS SUB SENDS 1 BYTE OF DATA TO THE DS1307

; CALL THIS FOR EACH REGISTER SECONDS TO YEAR

; ACC MUST CONTAIN DATA TO BE SENT TO CLOCK

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SEND\_BYTE:

MOV BITCNT,#08H; SET COUNTER FOR 8-BITS

SB\_LOOP:

JNB ACC.7,NOTONE; CHECK TO SEE IF BIT-7 OF

SETB SDA ; ACC IS A 1, AND SET SDA HIGH

JMP ONE

NOTONE:

CLR SDA ; CLR SDA LOW

ONE:

SCL\_HIGH ; TRANSITION SCL LOW-TO-HIGH

RL A ; ROTATE ACC LEFT 1-BIT

CLR SCL ; TRANSITION SCL LOW-TO-HIGH

DJNZ BITCNT,SB\_LOOP; LOOP FOR 8-BITS

SETB SDA ; SET SDA HIGH TO LOOK FOR

SCL\_HIGH ; ACKNOWLEDGE PULSE

CLR ACK

JNB SDA,SB\_EX ; CHECK FOR ACK OR NOT ACK

SETB ACK ; SET ACKNOWLEDGE FLAG FOR

; NOT ACK

SB\_EX:

ACALL DEELAY ; DELAY FOR AN OPERATION

CLR SCL ; TRANSITION SCL HIGH-TO-LOW

ACALL DEELAY ; DELAY FOR AN OPERATION

RET

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; SUB READS THE CLOCK AND WRITES IT TO THE SCRATCHPAD MEMORY

; ON RETURN FROM HERE DATE & TIME DATA WILL BE STORED IN THE

; DATE & TIME REGISTERS FROM 24H (SECS) TO 2AH (YEAR)

; ALARM SETTINGS IN REGISTERS 2CH(HRS) AND 2DH(MINUTES).

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

READ\_CLOCK:

MOV R1,#24H ; SECONDS STORAGE LOCATION

MOV BYTECNT,#00H

CLR LASTREAD

ACALL SEND\_START

MOV A,#DS1307W

ACALL SEND\_BYTE

MOV A,#00H

ACALL SEND\_BYTE

ACALL SEND\_STOP

ACALL SEND\_START

MOV A,#DS1307R

ACALL SEND\_BYTE

READ\_LOOP:

MOV A,BYTECNT

CJNE A,#09H,NOT\_LAST

SETB LASTREAD

NOT\_LAST:

ACALL READ\_BYTE

MOV @R1,A

MOV A,BYTECNT

CJNE A,#00H,NOT\_FIRST

MOV A,@R1

CLR ACC.7 ; ENSURE OSC BIT=0 (ENABLED)

MOV @R1,A

NOT\_FIRST:

INC R1

INC BYTECNT

MOV A,BYTECNT

CJNE A,#0AH,READ\_LOOP

ACALL SEND\_STOP

RET

;&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

; 7 SEGMENT DISPLAY ROUTINE

;&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

DISP:

MOV R2,SPEED

CJNE R2,#00H,AAS1

CLR DIS\_A

CLR DIS\_B

CLR DIS\_C

CLR DIS\_D

CLR DIS\_E

CLR DIS\_F

SETB DIS\_G

RET

AAS1: CJNE R2,#01H,AS2

CLR DIS\_B

CLR DIS\_C

SETB DIS\_A

SETB DIS\_D

SETB DIS\_E

SETB DIS\_F

SETB DIS\_G

RET

AS2: CJNE R2,#02H,AS3

CLR DIS\_A

CLR DIS\_B

CLR DIS\_D

CLR DIS\_E

CLR DIS\_G

SETB DIS\_C

SETB DIS\_F

RET

AS3: CJNE R2,#03H,AS4

CLR DIS\_A

CLR DIS\_B

CLR DIS\_C

CLR DIS\_D

CLR DIS\_G

SETB DIS\_E

SETB DIS\_F

RET

AS4: CJNE R2,#04H,AS5

CLR DIS\_B

CLR DIS\_C

CLR DIS\_F

CLR DIS\_G

SETB DIS\_A

SETB DIS\_D

SETB DIS\_E

RET

AS5: CJNE R2,#05H,AS6

CLR DIS\_A

CLR DIS\_C

CLR DIS\_D

CLR DIS\_F

CLR DIS\_G

SETB DIS\_B

SETB DIS\_E

RET

AS6: CJNE R2,#06H,AS7

CLR DIS\_A

CLR DIS\_C

CLR DIS\_D

CLR DIS\_E

CLR DIS\_F

CLR DIS\_G

SETB DIS\_B

RET

AS7: CJNE R2,#07H,AS8

CLR DIS\_A

CLR DIS\_B

CLR DIS\_C

SETB DIS\_D

SETB DIS\_E

SETB DIS\_F

SETB DIS\_G

RET

AS8: CJNE R2,#08H,AS9

CLR DIS\_A

CLR DIS\_B

CLR DIS\_C

CLR DIS\_D

CLR DIS\_E

CLR DIS\_F

CLR DIS\_G

RET

AS9: CJNE R2,#09H,AS10

CLR DIS\_A

CLR DIS\_B

CLR DIS\_C

CLR DIS\_D

CLR DIS\_F

CLR DIS\_G

SETB DIS\_E

RET

AS10: CJNE R2,#15H,AS11 ;symbol for -

SETB DIS\_A

SETB DIS\_B

SETB DIS\_C

SETB DIS\_D

SETB DIS\_E

SETB DIS\_F

CLR DIS\_G

RET

AS11: CJNE R2,#16H,AS12 ;switch off all disp

SETB DIS\_A

SETB DIS\_B

SETB DIS\_C

SETB DIS\_D

SETB DIS\_E

SETB DIS\_F

SETB DIS\_G

RET

AS12: MOV SPEED,#00H

AJMP DISP

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; INTRRUPT ROUTINE TO REFRESH THE DISPLAY

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

REFRESH:

PUSH PSW ; save current registerset

MOV PSW,#RB1

PUSH ACC

INC COUNT

MOV R4,COUNT

QA1: CJNE R4,#01H,QA2

MOV SPEED,VALUE\_1

SETB DIS1

CLR DIS2

CLR DIS3

CLR DIS4

CALL DISP

AJMP DOWN

QA2: CJNE R4,#02H,QA3

MOV SPEED,VALUE\_2

CLR DIS1

SETB DIS2

CLR DIS3

CLR DIS4

CALL DISP

AJMP DOWN

QA3: CJNE R4,#03H,QA4

MOV SPEED,VALUE\_3

CLR DIS1

CLR DIS2

SETB DIS3

CLR DIS4

CALL DISP

AJMP DOWN

QA4: CJNE R4,#04H,QA5

MOV SPEED,VALUE\_4

CLR DIS1

CLR DIS2

CLR DIS3

SETB DIS4

CALL DISP

AJMP DOWN

QA5: MOV COUNT,#01H

MOV R4,COUNT

AJMP QA1

DOWN: MOV TL0,#0FFH

MOV TH0,#0F0H

POP ACC

POP PSW

RETI

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FLASHING:

CALL DELAY ;Display on/off for 2 times

CALL DELAY

MOV VALUE\_1,#16H

MOV VALUE\_2,#16H

MOV VALUE\_3,#16H

MOV VALUE\_4,#16H

CALL DELAY

CALL DELAY

MOV VALUE\_1,NUMB1

MOV VALUE\_2,NUMB2

MOV VALUE\_3,NUMB3

MOV VALUE\_4,NUMB4

CALL DELAY ;Display on-off for 2 times

CALL DELAY

MOV VALUE\_1,#16H

MOV VALUE\_2,#16H

MOV VALUE\_3,#16H

MOV VALUE\_4,#16H

CALL DELAY

CALL DELAY

MOV VALUE\_1,NUMB1

MOV VALUE\_2,NUMB2

MOV VALUE\_3,NUMB3

MOV VALUE\_4,NUMB4

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DELAY:

MOV R1,#0CCH

REP2: MOV R2,#0FFH

REP1: NOP

DJNZ R2,REP1

DJNZ R1,REP2

RET

END

**4. COMPONENT REQUIREMENT**

**4.1. COMPONENT LIST:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. no.** | **Name of Components** | **Type** | **Quantity** |
| 1. | IC 89S8252 | Microcontroller | 1 |
| 2. | IC DS 1307 | Real Time Clock | 1 |
| 3. | IC 7805 Voltage Regulator | 5V | 1 |
| 4. | Transformer | Step-Down | 1 |
| 5. | Crystal | 12 MHz, 32.768KHz | 1,1 |
| 6. | Diode | 1N4700 | 3 |
| 7. | Relay Switch | 12V Magnetic Relay | 1 |
| 8. | Resistor | (2.2,10,56) KΩ,330E | 4,5,1 & 5 |
| 9. | Transistor (BC 547) | NPN | 5 |
| 10. | Storage Capacitor | 1 µF 25 V | 2 |
| 11. | Ceramic Capacitor | 33 pF | 2 |
| 12. | LED | General | 1 |
| 13. | Display | Seven Segment | 4 |
| 14. | Keypad | 4\*3 | 1 |
| 15. | Buzzer | 6-12 V operated | 1 |
| 16. | I.C. Base | 8 Pin & 40 Pin | 1,1 |

### 4.2. COMPONENT DESCRIPTION:-

**4.2.1. AT89S8252 (MICROCONTROLLER):-**

**PIN CONFIGURATION:-**

**Figure** 4.2.1.1*Pin Configuration* **Figure** 4.2.1.2 *IC AT89S8252*

The Atmel AT89S8252 provides the following standard features: 8K bytes of downloadable Flash, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next external interrupt or hardware reset.

Features of Atmel AT89C2051 are as follows:-

* Compatible with MCS-51 Products
* 8K Bytes of In-System Reprogrammable Downloadable Flash Memory
* 2K Bytes EEPROM
* 4V to 6V Operating Range
* Fully Static Operation: 0 Hz to 24 MHz
* Three-level Program Memory Lock
* 256 x 8-bit Internal RAM
* 32 Programmable I/O Lines
* Three 16-bit Timer/Counters
* Nine Interrupt Sources
* Programmable UART Serial Channel
* SPI Serial Interface
* Low-power Idle and Power-down Modes
* Interrupt Recovery from Power-down
* Programmable Watchdog Timer
* Dual Data Pointer
* Power-off Flag

The AT89S8252 IC’s pin description is given as follows:-

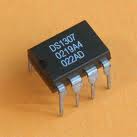
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pin Number** | **Description** |  | **Pin Number** | **Description** |
| 1 | Port 1.0 - T2 |  | 21 | Port 2.0 – A8 |
| 2 | Port 1.1 – T2 EX |  | 22 | Port 2.1 – A9 |
| 3 | Port 1.2 – Port1 |  | 23 | Port 2.2 – A10 |
| 4 | Port 1.3 – Port1 |  | 24 | Port 2.3 –A11 |
| 5 | Port 1.4 - SS |  | 25 | Port 2.4 – A12 |
| 6 | Port 1.5 – MOSI |  | 26 | Port 2.5 – A13 |
| 7 | Port 1.6 – MISO |  | 27 | Port 2.6 – A14 |
| 8 | Port 1.7 – SCK |  | 28 | Port 2.7 – A 15 |
| 9 | RST |  | 29 | PSEN |
| 10 | Port 3.0 – RXD |  | 30 | ALE/PROG |
| 11 | Port 3.1 – TXD |  | 31 | EA/VPP |
| 12 | Port 3.2 – INT0 |  | 32 | Port 0.7 – AD7 |
| 13 | Port 3.3 – INT1 |  | 33 | Port 0.6 –AD6 |
| 14 | Port 3.4 – T0 |  | 34 | Port 0.5 – AD5 |
| 15 | Port 3.5 – T1 |  | 35 | Port 0.6 - AD4 |
| 16 | Port 3.6 – WR |  | 36 | Port 0.3 – AD3 |
| 17 | Port 3.7 – RD |  | 37 | Port 0.2 – AD2 |
| 18 | XTAL2 – Crystal |  | 38 | Port 0.1 – AD1 |
| 19 | XTAL1 - Crystal |  | 39 | Port0.0 – AD0 |
| 20 | GND |  | 40 | VCC |

**4.2.2.****DS 1307 (REAL TIME CLOCK):-**

**PIN CONFIGURATIONS:-**

****

**Figure** 4.2.2.1 *Pin Diagram*

****

**Figure** 4.2.2.2 *IC DS 1307*

The DS1307 serial real-time clock (RTC) is a lowpower, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I2C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

Feature of IC DS1307 are as follows:

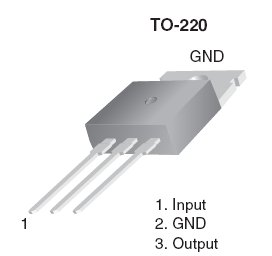
* Real-Time Clock (RTC) Counts Seconds
* Minutes, Hours, Date of the Month, Month, Day of the week, and Year with Leap-Year
* Compensation Valid Up to 2100
* 56-Byte, Battery-Backed, General-Purpose RAM with Unlimited Writes
* Programmable Square-Wave Output Signal
* Automatic Power-Fail Detect and Switch Circuitry
* Consumes Less than 500nA in Battery-Backup
* Mode with Oscillator Running
* Optional Industrial Temperature Range: - 40°C to +85°C

**PIN DISCRIPTION**:-

|  |  |
| --- | --- |
| **PIN Number** | **Description** |
| 1 | X1 – Crystal |
| 2 | X2 - Crystal |
| 3 | VBAT |
| 4 | GND |
| 5 | SDA |
| 6 | SCL |
| 7 | SQW/OUT |
| 8 | VCC |

**4.2.3.****LM7805 VOLTAGE REGULATOR:-**

The **78xx** (also sometimes known as **LM78xx**) series of devices is a family of self-contained fixed linear voltage regulator integrated circuits. The 78xx family is a very popular choice for many electronic circuits which require a regulated power supply, due to their ease of use and relative cheapness. When specifying individual ICs within this family, the *xx* is replaced with a two-digit number, which indicates the output voltage the particular device is designed to provide (for example, the **7805** has a 5 volt output, while the 7812 produces 12 volts). The 78xx line are positive voltage regulators, meaning that they are designed to produce a voltage that is positive relative to a common ground. There is a related line of **79xx** devices which are complementary negative voltage regulators. 78xx and 79xx ICs can be used in combination to provide both positive and negative supply voltages in the same circuit, if necessary.



**Figure** 4.2.3.1 *IC7805*

78xx ICs have three terminals and are most commonly found in the TO220 form factor, although smaller surface-mount and larger TO3 packages are also available from some manufacturers. These devices typically support an input voltage which can be anywhere from a couple of volts over the intended output voltage, up to a maximum of 35 or 40 volts, and can typically provide up to around 1 or 1.5 amps of current (though smaller or larger packages may have a lower or higher current rating).

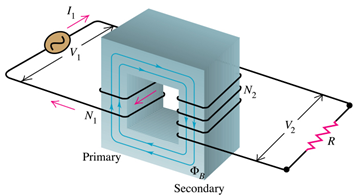
##### 4.2.4. TRANSFORMER:-

###### Definition: -

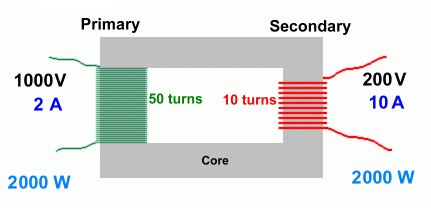
###### The transformer is a static electro-magnetic device that transforms one alternating voltage (current) into another voltage (current). However, power remains the some during the transformation. Transformers play a major role in the transmission and distribution of ac power.

**Principle:** -

Transformer works on the principle of mutual induction. A transformer consists of laminated magnetic core forming the magnetic frame. Primary and secondary coils are wound upon the two cores of the magnetic frame, linked by the common magnetic flux. When an alternating voltage is applied across the primary coil, a current flows in the primary coil producing magnetic flux in the transformer core. This flux induces voltage in secondary coil.



**Figure** 4.2.4.1 *Step-Up Transformer*



**Figure** 4.2.4.2 *Step-Down Transformer*

Transformers are classified as: -

1. *Based on position of the windings with respect to core i.e.*
2. Core type transformer
3. Shell type transformer
4. *Transformation ratio:*
5. Step up transformer
6. Step down transformer
7. *Core & shell types:* Transformer is simplest electrical machine, which consists of windings on the laminated magnetic core. There are two possibilities of putting up the windings on the core.
8. Winding encircle the core in the case of core type transformer
9. Cores encircle the windings on shell type transformer.
10. *Step up and Step down:* In these voltage transformation takes place according to whether the primary is high voltage coil or a low voltage coil.
11. Lower to higher-> Step up
12. Higher to lower-> Step down

##### 4.2.5. CRYSTAL:-

A piezoelectric **crystal** is an electronic circuit that uses the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a very precise frequency. This frequency is commonly used to keep track of time (as in quartz wristwatches), to provide a stable clock signal for digital integrated circuits, and to stabilize frequencies for radio transmitters and receivers. The most common type of piezoelectric resonator used is the quartz crystal, so oscillator circuits designed around them were called "crystal oscillators".

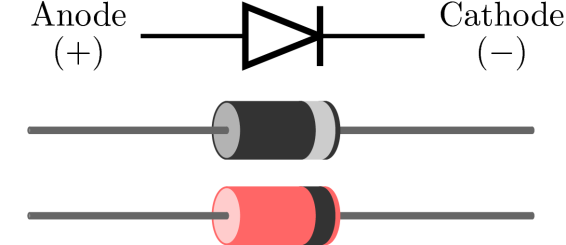
****http://t0.gstatic.com/images?q=tbn:Kiuo5IHDtji0NM:http://upload.wikimedia.org/wikipedia/commons/thumb/c/cb/Crystal-oscillator-IEC-Symbol.svg/120px-Crystal-oscillator-IEC-Symbol.svg.png

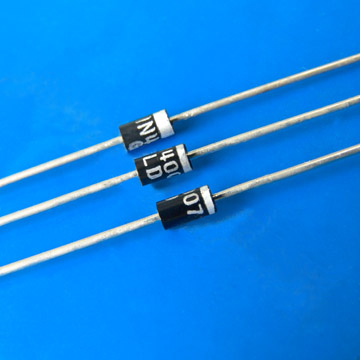
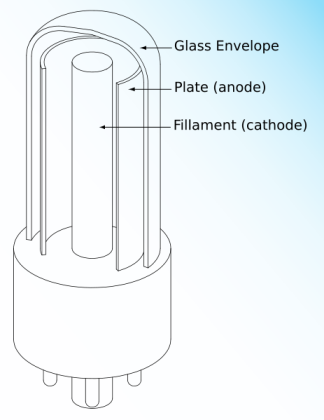
**Figure** 4.2.5.1 *Crystal*

The crystal oscillator circuit sustains oscillation by taking a voltage signal from the quartz resonator, amplifying it, and feeding it back to the resonator. The rate of expansion and contraction of the quartz is the resonant frequency, and is determined by the cut and size of the crystal. When the energy of the generated output frequencies matches the losses in the circuit, an oscillation can be sustained.

A regular timing crystal contains two electrically conductive plates, with a slice or tuning fork of quartz crystal sandwiched between them. During startup, the circuit around the crystal applies a random noise AC signal to it, and purely by chance, a tiny fraction of the noise will be at the resonant frequency of the crystal. The crystal will therefore start oscillating in synchrony with that signal. As the oscillator amplifies the signals coming out of the crystal, the signals in the crystal's frequency band will become stronger, eventually dominating the output of the oscillator. Natural resistance in the circuit and in the quartz crystal filter out all the unwanted frequencies.

##### 4.2.6. DIODE:-

****



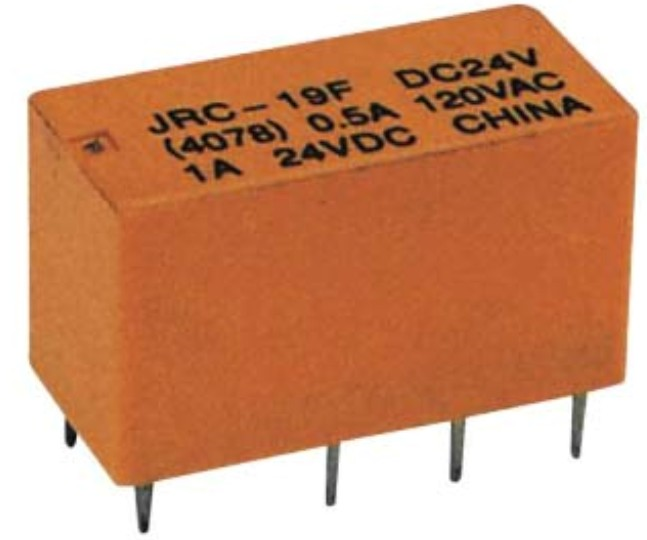
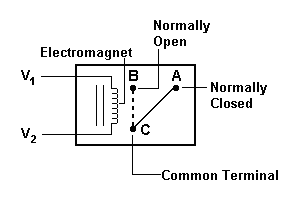
**Figure** 4.2.6.1 *Diode*

A diode is a two-terminal device. Diodes have two active electrodes between which the signal of interest may flow, and most are used for their unidirectional electric current property.

The unidirectionality most diodes exhibit is sometimes generically called the rectifying property. The most common function of a diode is to allow an electric current in one direction (called the forward biased condition) and to block the current in the opposite direction (the reverse biased condition). Thus, the diode can be thought of as an electronic version of a check valve.

Real diodes do not display such a perfect on-off directionality but have a more complex non-linear electrical characteristic, which depends on the particular type of diode technology. Diodes also have many other functions in which they are not designed to operate in this on-off manner.

##### 4.2.7. RELAY: -



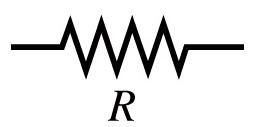
**Figure** 4.2.7.1 *Relay*

In this circuit a 12V magnetic relay is used. In magnetic relay, insulated copper wire coil is used to magnetize and attract the plunger .The plunger is normally connected to N/C terminal. A spring is connected to attract the plunger upper side. When output is received by relay, the plunger is attracted and the bulb glows.

### 4.2.8. RESISTORS:-

A Resistor is a heat-dissipating element and in the electronic circuits it is mostly used for either controlling the current in the circuit or developing a voltage drop across it, which could be utilized for many applications. There are various types of resistors, which can be classified according to a number of factors depending upon:

1. Material used for fabrication
2. Wattage and physical size
3. Intended application
4. Ambient temperature rating
5. Cost

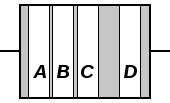
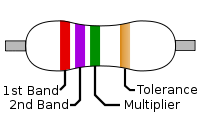
**Figure** 4.2.8.1 *Resistors*

Resistors may be classified as

1. Fixed
2. Semi variable
3. Variable resistor.

In our project carbon resistors are being used.

The **electronic color code** is used to indicate the values or ratings of electronic components, very commonly for resistors. Resistor values are always coded in ohms, capacitors in pico farads (pF), inductors in micro henries (µH), and transformers in volts.

[](http://en.wikipedia.org/wiki/File:R) [](http://en.wikipedia.org/wiki/File:4-)

**Figure** 4.2.8.2

* band **A** is first significant figure of component value
* band **B** is the second significant figure
* band **C** is the decimal multiplier
* band **D** if present, indicates tolerance of value in percent (no color means 20%)

For example, a resistor with bands of yellow, violet, red, and gold will have first digit 4 (yellow in table below), second digit 7 (violet), followed by 2 (red) zeros: 4,700 ohms. Gold signifies that the tolerance is ±5%, so the real resistance could lie anywhere between 4,465 and 4,935 ohms.

A useful mnemonic for remembering the first ten color codes matches the first letter of the color code, by order of increasing magnitude is as follows:-

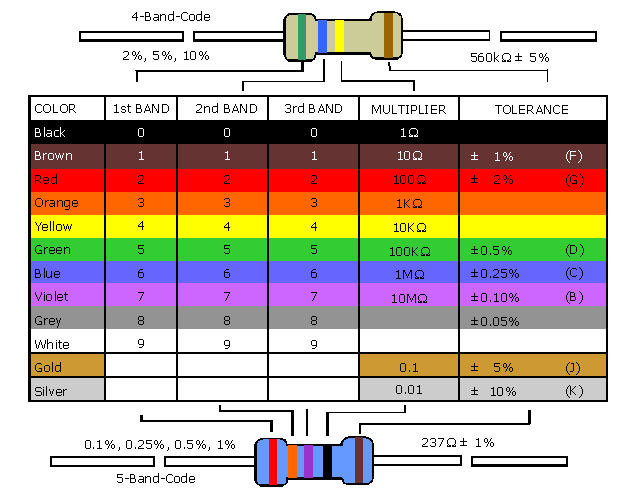
**B. B.** **R**oy of **G**reat **B**ritain has **V**ery **G**ood **W**ife.

Example:-

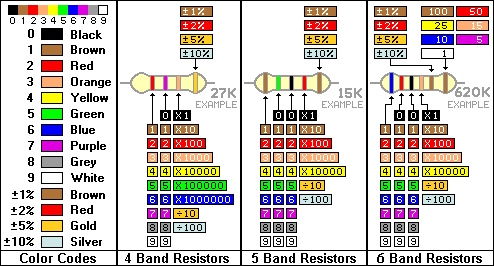
From top to bottom:

* Green-Blue-Brown-Black-Brown
  + 561 Ω ± 1%
* Red-Red-Orange-Gold
  + 22,000 Ω ± 5%
* Yellow-Violet-Brown-Gold
  + 470 Ω ± 5%
* Blue-Gray-Black-Silver
  + 68 Ω ± 10% (this wide of a tolerance is now seldom seen)
* Brown-Black-Brown
  + 10 Ω ± 20% (this wide of a tolerance is now seldom seen)
* Black
  + zero Ω

Color Coding of resistor is given in the following table:-



**Figure** 4.2.8.3

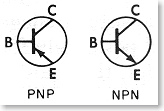


**Figure** 4.2.8.4

##### 4.2.9. TRANSISTORS: -

A transistor consists of two junctions formed by sandwiching either p-type or n-type semiconductor between a pair of opposite types. Accordingly, there are two types of transistors namely: -

(1) n-p-n transistor (2) p-n-p transistor

 C:\Users\Navs\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\images 8.jpg

**Figure** 4.2.9.1 *Transistors* **Figure** 4.2.9.2 *Transistor (BC 547)*

An n-p-n transistor is composed of two n-type semiconductors separated by a thin section of p type. However two p sections separated by a thin section of n-type form a p-n-p transistor.

A transistor raises the strength of a weak signal and thus acts as an amplifier. The weak signal is applied between emitter base junction and output is taken across the load Rc connected in the collector circuit (in common emitter configuration). In order to achieve faithful amplification, the input circuit should always remain forward biased. To do so, a dc voltage is applied in the input in addition to the signal. This dc Voltage is known as biasing voltage and its magnitude and polarity should be such that it always keeps the input circuit forward biased regardless of the polarity to the signal to be amplified.

As the input circuit has low resistance a small change in signal voltage causes an appreciable change in emitter current. This causes change in collector current (by a factor called current gain of transistor) due to transistor action. The collector current flowing through a high load resistance Rc produces a large voltage across it. Thus a weak signal applied to the input circuit appears in the amplified form in the collector circuit. This is how a transistor acts as an amplifier.

**4.2.10. CAPACITORS: -**

The fundamental relation for the capacitance between two flat plates separated by a dielectric material is given by: -

C=0.08854KA/D

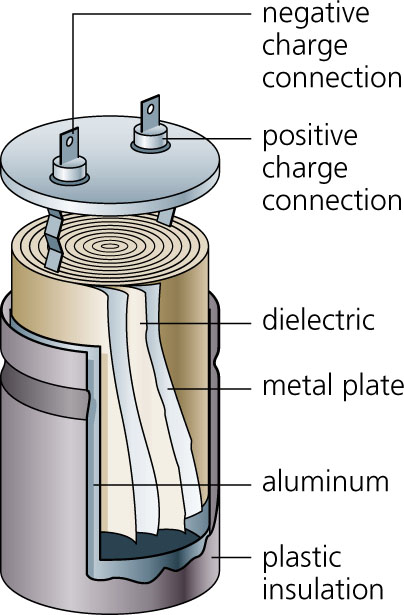
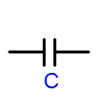
Where: -

C= capacitance in pf.

K= dielectric constant

A=Area per plate in square cm.

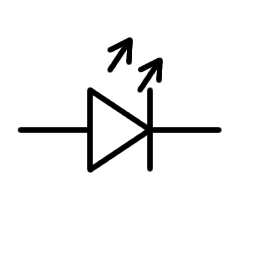
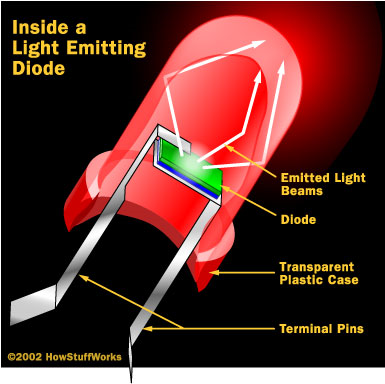
D=Distance between two plates in cm

**Figure** 4.2.10.1 *Capacitor*

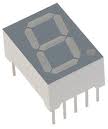
Design of capacitor depends on the proper dielectric material with particular type of application. The dielectric material used for capacitors may be grouped in various classes like Mica, Glass, air, ceramic, paper, Aluminum, electrolyte etc. The value of capacitance never remains constant. It changes with temperature, frequency and aging. The capacitance value marked on the capacitor strictly applies only at specified temperature and at low frequencies.

**4.2.11. LED (Light Emitting Diode): -**

 **Figure**4.2.11.1 *Light Emitting Diode*

As its name implies it is a diode, which emits light when forward biased. Charge carrier recombination takes place when electrons from the N-side cross the junction and recombine with the holes on the P side. Electrons are in the higher conduction band on the N side whereas holes are in the lower valence band on the P side. During recombination, some of the energy is given up in the form of heat and light. In the case of semiconductor materials like Gallium arsenide (GaAs), Gallium phosphide (GaP) and Gallium arsenide phosphide (GaAsP) a greater percentage of energy is released during recombination and is given out in the form of light. LED emits no light when junction is reverse biased.

**`4.2.12. SEVEN SEGMENT DISPLAY:-**



**Figure** 4.2.12.1 Seven Segment Display

A seven segment display, as its name indicates, is composed of seven elements. Individually on or off, they can be combined to produce simplified representations of the Arabic numerals. Often the seven segments are arranged in an *oblique* (slanted) arrangement, which aids readability. Each of the numbers 0, 6,7and 9 may be represented by two or more different glyphs on seven-segment displays.

**4.2.13. KEYPAD (4\*3):-**



**Figure** 4.2.13.1 Keypad (4\*3)

**A** simple 4x3 keyboard that allows data entry into bus based systems. Flowcode macros for driving this E-block are available.

.

5. P.C.B. MANUFACTURING PROCESS

#### It is an important process in the fabrication of electronic equipment. The design of PCBs (Printed Circuit Boards) depends on circuit requirements like noise immunity, working frequency and voltage levels etc. High power PCBs requires a special design strategy.

The fabrication process to the printed circuit board will determine to a large extent the price and reliability of the equipment. A common target aimed is the fabrication of small series of highly reliable professional quality PCBs with low investment. The target becomes especially important for customer tailored equipments in the area of industrial electronics.

The layout of a PCB has to incorporate all the information of the board before one can go on the artwork preparation. This means that a concept which clearly defines all the details of the circuit and partly defines the final equipment, is prerequisite before the actual lay out can start. The detailed circuit diagram is very important for the layout designer but he must also be familiar with the design concept and with the philosophy behind the equipment.

**5.1. BOARD TYPES**:-

The two most popular PCB types are:

**5.1.1 Single Sided Boards**

The single sided PCBs are mostly used in entertainment electronics where manufacturing costs have to be kept at a minimum. However in industrial electronics cost factors cannot be neglected and single sided boards should be used wherever a particular circuit can be accommodated on such boards.

**5.1.2 Double Sided Boards**

Double-sided PCBs can be made with or without plated through holes. The production of boards with plated through holes is fairly expensive. Therefore plated through hole boards are only chosen where the circuit complexities and density of components does not leave any other choice.

**5.2. DESIGN SPECIFICATION:-**

***5.2.1 STEPS TAKEN WHILE PREPARING CIRCUIT***

**5.2.1.1 PCB DESIGNING:-**

The main purpose of printed circuit is in the routing of electric currents and signal through a thin copper layer that is bounded firmly to an insulating base material sometimes called the substrate. This base is manufactured with an integrally bounded layer of thin copper foil which has to be partly etched or removed to arrive at a pre-designed pattern to suit the circuit connections or other applications as required.

The term printed circuit board is derived from the original method where a printed pattern is used as the mask over wanted areas of copper. The PCB provides an ideal baseboard upon which to assemble and hold firmly most of the small components.

From the constructor’s point of view, the main attraction of using PCB is its role as the mechanical support for small components. There is less need for complicated and time consuming metal work of chassis contraception except perhaps in providing the final enclosure. Most straight forward circuit designs can be easily converted in to printed wiring layer the thought required to carry out the inversion cab footed high light an possible error that would otherwise be missed in conventional point to point wiring .The finished project is usually neater and truly a work of art.

Actual size PCB layout for the circuit shown is drawn on the copper board. The board is then immersed in FeCl3 solution for 12 hours. In this process only the exposed copper portion is etched out by the solution.

Now the petrol washes out the paint and the copper layout on PCB is rubbed with a smooth sand paper slowly and lightly such that only the oxide layers over the Cu are removed. Now the holes are drilled at the respective places according to component layout as shown in figure.

**5.2.1.2 LAYOUT DESIGN**:-

When designing the layout one should observe the minimum size (component body length and weight). Before starting to design the layout we need all the required components in hand so that an accurate assessment of space can be made. Other space considerations might also be included from case to case of mounted components over the printed circuit board or to access path of present components.

It might be necessary to turn some components around to a different angular position so that terminals are closer to the connections of the components. The scale can be checked by positioning the components on the squared paper. If any connection crosses, then one can reroute to avoid such condition.

All common or earth lines should ideally be connected to a common line routed around the perimeter of the layout. This will act as the ground plane. If possible try to route the outer supply line to the ground plane. If possible try to route the other supply lines around the opposite edge of the layout through the center. The first set is tearing the circuit to eliminate the crossover without altering the circuit detail in any way.

Plan the layout looking at the topside to this board. First this should be translated inversely, later for the etching pattern large areas are recommended to maintain good copper adhesion. It is important to bear in mind always that copper track width must be according to the recommended minimum dimensions and allowance must be made for increased width where termination holes are needed. From this aspect, it can become little tricky to negotiate the route to connect small transistors.

There are basically two ways of copper interconnection patterns under side the board. The first is the removal of only the amount of copper necessary to isolate the junctions of the components to one another. The second is to make the interconnection pattern looking more like conventional point wiring by routing uniform width of copper from component to component.

5.2.1.3 **ETCHING PROCESS**:-

Etching process requires the use of chemicals. Acid resistant dishes and running water supply. Ferric chloride is mostly used solution but other etching materials such as ammonium per sulphate can be used. Nitric acid can be used but in general it is not used due to poisonous fumes. The pattern prepared is glued to the copper surface of the board using a latex type of adhesive that can be cubed after use. The pattern is laid firmly on the copper using a very sharp knife to cut round the pattern carefully to remove the paper corresponding to the required copper pattern areas. Then apply the resistant solution, which can be a kind of ink solution for the purpose of maintaining smooth clean outlines as far as possible. While the board is drying, test all the components.

Before going to next stage, check the whole pattern and cross check with the circuit diagram. Check for any free metal on the copper. The etching bath should be in a glass or enamel disc. If using crystal of ferric- chloride these should be thoroughly dissolved in water to the proportion suggested. There should be 0.5 lt. of water for 125 gm of crystal.

To prevent particles of copper hindering further etching, agitate the solutions carefully by gently twisting or rocking the tray. The board should not be left in the bath a moment longer than is needed to remove just the right amount of copper. Inspite of there being a resistive coating there is no protection against etching away through exposed copper edges. This leads to over etching. Have running water ready so that etched board can be removed properly and rinsed. This will halt etching immediately.

Drilling is one of those operations that calls for great care. For most purposes a 0.5mm drill is used. Drill all holes with this size first those that need to be larger can be easily drilled again with the appropriate larger size.

5.2.1.4 **COMPONENT ASSEMBLY: -**

From the greatest variety of electronic components available, which runs into thousands of different types it is often a perplexing task to know which is right for a given job.

There could be damage such as hairline crack on PCB. If there are, then they can be repaired by soldering a short link of bare copper wire over the affected part.

The most popular method of holding all the items is to bring the wires far apart after they have been inserted in the appropriate holes. This will hold the component in position ready for soldering.

Some components will be considerably larger .So it is best to start mounting the smallest first and progressing through to the largest. Before starting, be certain that no further drilling is likely to be necessary because access may be impossible later.

Next will probably be the resistor, small signal diodes or other similar size components. Some capacitors are also very small but it would be best to fit these afterwards. When fitting each group of components mark off each one on the circuit as it is fitted so that if we have to leave the job we know where to recommence.

Although transistors and integrated circuits are small items there are good reasons for leaving the soldering of these until the last step. The main point is that these components are very sensitive to heat and if subjected to prolonged application of the soldering iron, they could be internally damaged.

All the components before mounting are rubbed with sand paper so that oxide layer is removed from the tips. Now they are mounted according to the component layout.

5.2.1.5 **SOLDERING: -**

This is the operation of joining the components with PCB after this operation the circuit will be ready to use to avoid any damage or fault during this operation following care must be taken.

**1.** A longer duration contact between soldering iron bit & components lead can exceed the temperature rating of device & cause partial or total damage of the device. Hence before soldering we must carefully read the maximum soldering temperature & soldering time for device.

**2.** The wattage of soldering iron should be selected as minimum as permissible for that soldering place.

**3.** To protect the devices by leakage current of iron its bit should be earthed properly.

**4.** We should select the soldering wire with proper ratio of Pb & Tn to provide the suitable melting temperature.

**5.** Proper amount of good quality flux must be applied on the soldering point to avoid dry soldering.

**6. APPLICATION & Advantages**

1. It can be used in the college, school for belling purpose.

2. It can be used in the any type of examination for belling because we can set the ringing time.

3. Automatic scheduling of college bell is possible.

4. Compact in size so takes less space.

5. Time editable facility is available.

**7. LIMITATIONS**

The all ringing time should be given at a time.

The previous ringing time will removed from the memory itself.

We have used the 24-hour mode for the input of the ringing time.

**8. FUTURWE SCOPE**

**9. REFERENCE**

* www.google.com
* www.8051projects.info
* www.en.wikipedia.org
* www.yahoo.com/search
* www.alldatasheet.com
* www.datasheetcatalog.com/datasheets\_pdf/7/8/0/5/7805.shtml
* 8051 Microcontroller and Embedded Systems by Mazidi and Mazidi
* Applied Electronics by R. S. Sedha