International Journal of Industrial Electronics and Control.
ISSN 0974-2220 Volume 6, Number 1 (2014), pp. 7-13

http://www.irphouse.com

Use of Microprocessor in LED Dial Clock

Dr. O. P. Garg[*] and Dr. D. K. Kaushik[**]

[*]Head of the Department of Electronics,
R. K. S. D. College, Kaithal (Haryana)
[**]Principal, M. M. College, Fatehabad (Haryana)

ABSTRACT:

The authors in the present paper used a popular Microprocessor 8085 for the design of LED dial clock. The LED dial clock works as real time clock but its time is seen on the circular board having 60 LEDs for minutes and 12 LEDs are used for hrs. However, one LED flickers indicating seconds are continuously being counted by the clock. The program for the Microprocessor is written in Assembly Language of Microprocessor 8085. The circuit and software both were found to work satisfactory.

Working:

Here the design of microprocessor based LED dial clock has been discussed which runs using microprocessor 8085. It works purely on the basis of the software; and some hardware has also been used. The software is prepared in the assembly language of 8085A microprocessor and it was tested on Vinytics 8085 μp kit and found to work very fine.

In this dial clock 73 LEDs are used which are interfaced with the different ports of two PPI 8255 ICs available on μp kit. Out of 73 LEDs, 12 LEDs of Amber color are used to show hours, 60 LEDs of red color are used to show minutes and one green LED alternately glows for half second showing that the seconds are continuously being counted by the clock. All these LEDs are connected in a circle in the form of a dial of a wall clock. The beauty of this project is the software part. It is worthwhile to mention that no electronic circuit was used, except the LEDs are connected to two PPI-8255 ICs available with the microprocessor kit.

Figures 1 and 2 show the circuit diagrams in which LEDs are interfaced with the μp-Kit. The cathodes of all the 12 Amber colored Hours LEDs are connected to the ground. The Anodes of these LEDs are connected to the Port A and Port B of 8255–II, the anodes of LEDs showing Hrs 1 to 8 are connected respectively to D0–D7 Bits of Port A of 8255-II and the anodes of LEDs showing Hours 9 to 12 are connected respectively to D0–D3 bits of Port B of 8255–II. Look–up table 1 provides Logic 1 to

Hours LEDs 1-8. However, logic 1 to Hours LEDs 9–12 are provided using the software. The bit D0 of Port C of 8255–II is connected to the anode of Green LED (showing second) and cathode of this LED is also grounded.

The minutes red colored 60 LEDs are divided into 6 Groups (Group 0 to Group 5) and each group contains 10 LEDs. The cathodes of each group are connected together. The D0–D5 bits of Port B of 8255–I are connected to the cathodes (Common Points) of Group 0 to Group 5 LEDs respectively. The ground signal is provided to the cathode pin of each Group through the software using Look up table 2.

The anodes of 10 LEDs of each of 6 Groups are connected in parallel as shown in fig.2. In this way 10 different Anode leads (1 to 10) are finally available to be connected to different pins of Port A and Port C of 8255–I. D0–D7 bits of Port A of 8255–I are connected to 1–8 Anodes leads and 9–10 Anode leads are connected to D0–D1 bits of Port C of 8255–I respectively. Look up table 3 provides Logic 1 to 1–8 Anode Leads; and to the 9–10 Anode leads the Logic 1, is provided by the software.

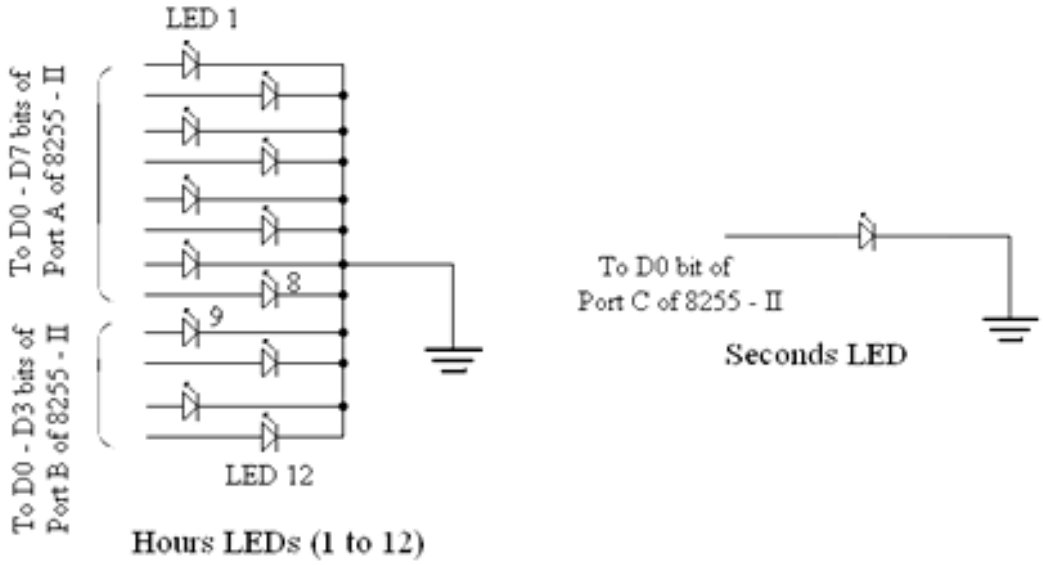

Fig. 1

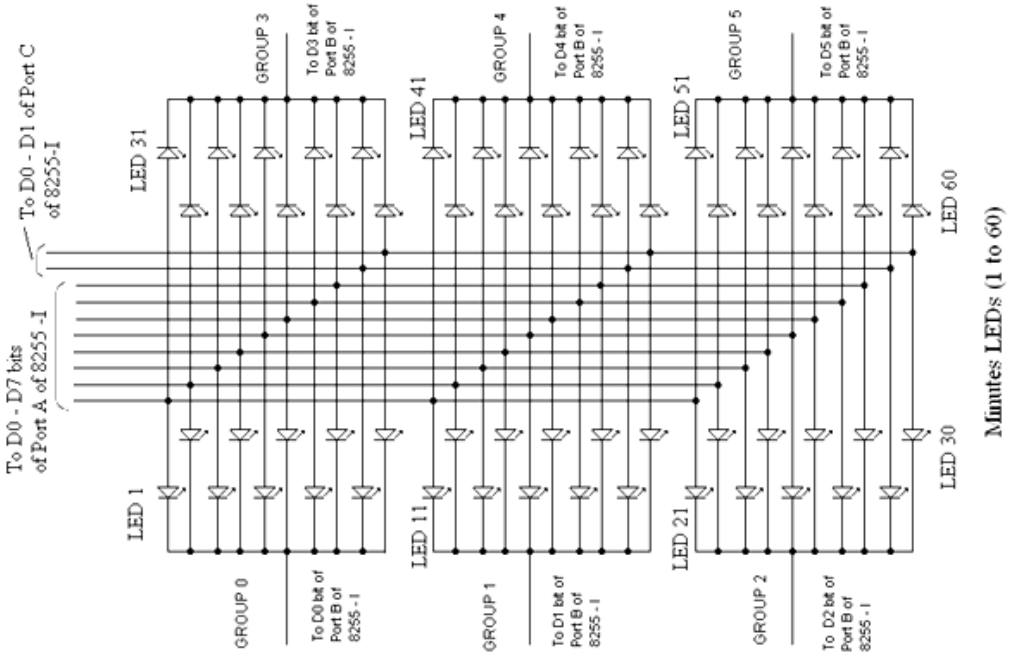

Fig. 2

Table–1

Memory Location	Data Stores	Binary Equivalent	Comments (Port A of 8255-II)
2101 H	01H	0000 0001	Provides Logic 1 to D0 bit (Hrs.1)
2102 H	02H	0000 0010	Provides Logic 1 to D1 bit (Hrs.2)
2103 H	04H	0000 0100	Provides Logic 1 to D2 bit (Hrs.3)
2104 H	08H	0000 1000	Provides Logic 1 to D3 bit (Hrs.4)
2105 H	10H	0001 0000	Provides Logic 1 to D4 bit (Hrs.5)
2106 H	20H	0010 0000	Provides Logic 1 to D5 bit (Hrs.6)
2107 H	40H	0100 0000	Provides Logic 1 to D6 bit (Hrs.7)
2108 H	80H	1000 0000	Provides Logic 1 to D7 bit (Hrs.8)

Table–2

Memory Location	Data Stores	Binary Equivalent	Comments (Port B of 8255-I)
2200 H	FEH	1111 1110	Provides Logic 0 to D0 bit (Group 0)
2201 H	FDH	1111 1101	Provides Logic 0 to D1 bit (Group 1)
2202 H	FBH	1111 1011	Provides Logic 0 to D2 bit (Group 2)
2203 H	F7H	1111 0111	Provides Logic 0 to D3 bit (Group 3)
2204 H	EFH	1110 1111	Provides Logic 0 to D4 bit (Group 4)
2205 H	DFH	1101 1111	Provides Logic 0 to D5 bit (Group 5)

Table–3

Memory Location	Data Stores	Binary Equivalent	Comments (Port A of 8255–I)
2300 H	01H	00000001	Provides Logic 1 to D0 bit
2301 H	02H	00000010	Provides Logic 1 to D1 bit
2302 H	04H	00000100	Provides Logic 1 to D2 bit
2303 H	08H	00001000	Provides Logic 1 to D3 bit
2304 H	10H	00010000	Provides Logic 1 to D4 bit
2305 H	20H	00100000	Provides Logic 1 to D5 bit
2306 H	40H	01000000	Provides Logic 1 to D6 bit
2307 H	80H	10000000	Provides Logic 1 to D7 bit

The software provides a signal continuously to Green LED so that it remains ON and OFF alternately for half a second. The minutes 60 LEDs glow one by one after every one minute. Port B of 8255–I provides a low voltage (0 volt) to the cathodes of one block for 10 minutes and to the anode of these LEDs (1 to 10) are provided positive voltage (logic 1) alternately for one minute through D0–D7 bits of Port A and D0–D2 bits of Port B of 8255–I. However, Hours' 12 Amber colored LEDs glow one by one after every an Hour time, as positive voltage (logic 1) is provided to the anodes of these LEDs after every an Hour. As discussed earlier positive voltage to Hours LEDs (1 to 8) are provided by D0–D7 bits of Port A of 8255–II and to Hours LEDs (9 to 12) are provided by D0–D3 bits of Port B of 8255–II after every an hour time. All the LEDs glow in a sequence as the minutes and hours hands move in an analog clock.

The current time is stored in the beginning in some memory locations. Hours, minutes and seconds of the current time are stored in the memory locations say 2000 H to 2002 H respectively. The software in the assembly language of 8085A for providing appropriate signal to different LEDs is given below:

Software: Main Program

MEMORY LOCATIONS WHERE CURRENT TIME IS STORED

2000H - CURRENT HRS
2001H - CURRENT MINUTES
2002H - CURRENT SECONDS

Label	Mnemonics	Operand	Comments
	LXI SP,	XXXX H	; Initialize Stack Pointer.
	MVI A,	80H	; Control word for 8255-I.
	OUT	03H	; Works all ports of 8255-I as o/p port.
	OUT	0B H	; Works all ports of 8255-II as o/p port.

```
START   LXI H,      2000 H   ; Loads the H-L register pair with the address of
                             Hrs. of current time.
        MOV A,      M        ; Moves the Hrs. of current time to acc.
        CPI         09 H     ; It is compared with 09 H.
        JC          PT0      ; If ACC<9, then jump to PT0.
        JZ          PT1      ; If ACC=9, then jump to PT1.
        CPI         10 H     ; ACC contents are compared with 10 H.
        JZ          PT2      ; If ACC=10 then jump to PT2.
        CPI         11 H     ; If ACC=11 then jump to PT3.
        MVI A,      00 H
        OUT         08 H     ; Logic 0 to all bits of Port A 8255-II.
        MVI A,      08 H
        OUT         09 H     ; Logic 1 to D3 bit of port B of 8255-II.
        JMP         PT5      ; Jumps to PT5.
PT1     MVI A,      00 H
        OUT         08 H     ; Logic 0 to all the bits of port A of 8255-II.
        MVI A,      01 H
        OUT         09 H     ; Logic 1 to D0 bit of port B of 825-II.
        JUMP        PT5      ; Jumps to PT5.
PT2     MVI A,      00 H
        OUT         08 H     ; Logic 0 all the bits of port A of 8255-II.
        MVI A,      02 H
        OUT         09 H     ; Logic 1 to D1 bit of Port B of 8255-II.
        JMP         PT5      ; Jumps to PT5.
PT3     MVI A,      00 H
        OUT         08 H     ; Logic 0 all the bits of port A of 8255-II.
        MVI A,      04 H
        OUT         09 H     ; Logic 1 to D2 bit of port B of 8255-II.
        JMP         PT5      ; Jumps to PT5.
PT0     MOV E,      M        ; Hrs. of current time is < 9 loaded to E reg.
        MVI A,      00 H
        OUT         09 H     ; Logic 0 to all the bits of port B of 8255-II.
        MVI D,      21 H
        LDAX        D        ; Loads Acc data (table 15.1) of the current time,
                             which is less than 9.
        OUT         08 H     ; Logic 1to D1 port A of 8255-II (table 15.1).
PT5     INX H                ; H-L reg. pair shows min. of the current ime.
        MOV A,      M        ; Moves current min to ACC.
        ANI         F0 H     ; Make lower Nibble of ACC to zero.
        MVI B,      04 H
LOOP    RLC                  ; Rotate ACC 4 times to shift
        DCR         B
        JNZ         LOOP
        MOV E,      A        ; ACC conts are loaded to E register.
        INR         D
        LDAX D
        OUT         01 H     ; Logic 0 to the cathodes of one group from
                             group 0 to group 5 (table 15.2).
        MOV A,      M
```

```
        ANI     0F H    ; Logic 0 to higher Nibble of ACC.
        CPI     09 H    ; Compared with 09.
        JZ      AK      ; If ACC = 09 then jump to AK.
        CPI     08 H    ; Compared with 08.
        JZ      AK1     ; If ACC=08 then jump to AK1.
        MVI A.  00 H
        OUT     02 H    ; Logic 0 to all the bits of Port C of 8255-I.
        MOV E,  A
        INR     D
        LDAX D          ; Logic 1 to one of the bits of D0-D7 (table
                          15.3).
        OUT     00 H
        JMP     AK2     ; Jump to AK2.
AK1     MVI A,  00 H
        OUT     00 H    ; Logic 0 to all the bits of Port A    of 8255-I.
        MVI A,  01 H
        OUT     02 H    ; Logic 1 to D0 bit of port C of 8255-I.
        JMP     AK2     ; Jump to AK2.
AK      MVI A,  00 H
        OUT     00 H    ; Logic 0 to all the bits of Port A    of 8255-I.
        MVI A,  02 H
        OUT     02 H    ; Logic 1 to D1 bit of port C of 8255-I.
AK2     INX H
PT7     MOV A,  M       ; Moves current seconds to ACC.
        ADI     01 H    ; ACC = ACC+1
        DAA             ; Decimal adjust the Accumulator.
        CPI     60 H    ; Is ACC=60?
        JZ      PT6     ; If yes jump to PT6.
        MOV M,  A       ; Else current secs to its corresponding locations.
        CALL    DELAY   ; Calls one second delay program .
        JMP     PT7     ; Jumps to PT7.
PT6     MVI A,  00 H
        MOV M,  A       ; Stores 00 to locations of current seconds.
        DCX H
        MOV A,  M       ; Mins. of the current time is loaded to ACC.
        ADI     01 H    ; ACC = ACC + 1
        DAA             ; Decimal adjust the Accumulator.
        CPI     60 H    ; Is ACC=60?
        JZ      PT8     ; If yes jump to PT8.
        MOV M,  A       ; Stores minutes to the Minutes locations.
        JMP     START   ; Jump to START to glow mins and Hrs.
PT8     MVI A,  00 H
        MOV M,  A       ; Stores 00 to locations of current minutes.
        DCX H
        MOV A,  M       ; Hrs. of the current time is loaded to ACC.
        ADI     01 H    ; ACC = ACC+1
        DAA             ; Decimal adjust the Accumulator.
        CPI     13 H    ; Is ACC=13?
        JZ      PT9     ; If yes jump to PT9.
```

```
        MOV M,    A       ; Stores Hrs. to the Hrs. locations.
        JMP       START   ; Jump to START to glow mins and Hrs.
PT9     MVI A,    01 H
        MOV M,    A       ; Stores Hrs.=01 in Hrs.
        JMP       START   ; Jump to START to glow mins and Hrs.
```

Delay Subroutine Program

Label	Mnemonics	Operand	Comments
DELAY	MVI A,	01 H	
	OUT	0A H	; Seconds LED glows for half second.
	LXI D,	FA00 H	
NXT1	DCX	D	; Decrement D-E register pair.
	MOV A,	D	; Moves the contents stored in $M_{D\text{-}E}$ to Acc.
	ORA	E	; Contents of A and E regs ORed bit by bit.
	JNZ	NXT1	; If result is not zero then jump to NXT1.
	MVI A,	00 H	
	OUT	0A H	; Seconds LED remains off for half second.
	LXI D,	FA00 H	
NXT2	DCX	D	; Decrement D-E register pair.
	MOV A,	D	; Moves the contents stored in $M_{D\text{-}E}$ to Acc.
	ORA	E	; Contents of A and E regs ORed bit by bit.
	JNZ	NXT2	; If the result is not zero then jump to NXT2.
	RET		; Return to main program.

The above program written in the Assembly language is fed to the Microprocessor board and as it is executed after connected LEDs to the PPIs as discussed above shows the LED dial clock. It was found to work very fine.