

MICROCONTROLLER TRAINER

Model - LGS-51

OPERATIONAL MANUAL

LOGSUN SYSTEMS

**B2 RAUNAK, 134/4+B, MAYUR COLONY,
KOYHRUD, PUNE-411038**

PHONE .- 020 -24356456 , 32937840

**Email - sales@logsun.com/support@logsun.com
www.logsun.com**



Dear Users,

First we are congratulating you for buying a LGS51 series trainer kit. If you are new to 8051 core the LGS51 series Trainer is designed to be both useful and instructive in illustrating certain concepts commonly encountered in 8051/52-based development. Although many kits already exist, this kit is designed specifically to form the basis of thorough technical discussions and tutorials consistent with Experts in the field of Microcontroller development & Experience Faculties in Microcontroller Subject. Nothing was included in this Trainer kit that does not serve a specific educational purpose and implementations that would confuse or unnecessarily complicate the design were intentionally omitted in favor of easy-to-understand approaches so that the important topics could be covered without being lost in the details.

This Trainer is also designed with a Indian market in mind. Recognizing that not all parts are easily available in all parts of the India, this kit intentionally uses common parts that should not be difficult to acquire.

The Kit is was developed with the 8031/51 in mind but can be used with any 40-pin 8052 pin-compatible derivative including the traditional true-blue 8052, 8051, 8032, 8031, 89C55WD, 89VRD2 etc.

LGS51-2 has 16 x 2 LCD display (20x4 optionally available), other facilities available like Memory 32K BB RAM, 8 +16K additional RAM, 32K Monitor EPROM. Memory expandable up to 128K, 48 I/O Lines from 8255 x 2nos. 2 nos. of serial ports from 8051 & 8251, on board 8253. You can access 14 I/O lines from 8051 core CPU which is very useful for students to interface other devices directly to microcontroller like DAC, stepper motor, DC motor etc. The Kit has built in Line assembler & disassembler & with the use of PS2 Type keyboard user can directly write/enter the program in assembly language and also checked the entered code in mnemonics using disassembler facility. So without help of PC any type of development can be completed. PC interfacing is available as a built in feature.

To develop the skills in applications using 8051 we have wide range of interfacing modules to select like ADC, DAC, stepper motor etc.

The Books we referred are 1) The 8051 Microcontroller Architecture, programming & applications – Kenneth Ayala 2) The 8051 Microcontroller & Embedded Systems - M A Mazidi , Data books of Philips, ATmel.

Further if you need any assistance, support you may contact us at Logsun Systems, 4A, Shree Sadguruniwas Society, Hingane (KH), Sinhgad Road, Pune- 411051 or call us at 020 24356456/32937840 or you can email us to support@logsun.com . Assuring our best of services.

From

Team Logsun Systems



INDEX

S. NO.	CONTENT	PAGE NO.
01	HARDWARE DETAILS	02
02	MONITOR COMMANDS	04
03	SERIAL LINK COMMANDS	09
04	PROCEDURE / ENTER & EXECUTE A PROGRAM	11
05	SERIAL COMMUNICATION USING HYPERTERMINAL	16
06	USEFUL SUBROUTINES	18
07	SAMPLE PROGRAMS	19
08	HARDWARE EXERCISES	30
09	CONNECTOR DETAILS	33
10	LINE ASSEMBLER AND DISASSEMBLER	35
11	SCHEMATIC DIAGRAM	37
12	COMPONENT LAYOUT	39
13	8051 INSTRUCTION SETS	40

MODEL-LGS-51 Table of Command Keys

Sr.no.	Keys	Description
1	S	Substitute Data into RAM/ Read Data from
	Memory	
2	C	Compare Block/Data in Memory
3	P	To Check Data of previous location in Substitute Command
4	M	Move Block of memory within memory.
5	F	Fill Data into memory Block
6	G / E	To, Execute the program in full swing or single step.
7	R	To check/Modify register Bank & SFR
8	W	Word Search of string of 4
9	I	Serial Data in from PC or any terminal
10	O	Serial Data out to PC or any Terminal
11	T	Terminal mode to use trainer from any PC
12	Esc	Return to Command mode
13	Enter	To proceed to next step in any above command.
14		All other keys are available to user

Other Imp. Points- In LGS-51 kit you can use **8031 with external EPROM, 8751/52/ 89c51/52 with internal EPROM** as per customer's requirement.

User RAM	- Battery backup-from-8000 to FFFFH- 32K
User RAM	- Scratch Pad From 2000 to 3FFFH- 8K
Display	- 16*1, 16*2, 16*4
Keyboard	- Any standard IBM compatible.

HARDWARE DETAILS

- 1) 0000H To 1FFFH-8K Monitor EPROM (Program Memory)
- 2) 2000H To 3FFFH-8K Scratch Pad RAM Available to user, after **27FFH**. (Data Memory)
- 3) 8000H To FFFFH-32K User RAM Battery backup. (Data Memory)
- 4) 0003H Data memory CW for 8255
 - 0000H Port A
 - 0001H Port B
 - 0002H Port C
- 5) 0203H Data CW for 8253
 - 0200H Channel 0
 - 0201H Channel 1
 - 0202H Channel 2
- 6) 0400H LCD Display
- 7) Interrupt vector table for LGS 51 kit

Interrupt	RAM Location
INT0	FF03
TFO	FF0B
INT1	FF13
TF1	FF1B
RI and TI	FF23
- 8) Internal Memory Mappings,
 - ; 00h - 0fh bank 0/1 user access
 - ; 10h - 27h monitor scratch / monitor bank 2 (r0 - r7)
 - ; 28h - 3fh user scratch
 - ; 40h - 5fh monitor data
 - ; 60h - 7fh monitor stack

(For 8052/89C52 user can use additional Internal RAM from 80H to FFH)

ckc	equ	41h
pkc	equ	42h
lgk	equ	43h
roff	equ	44h
adrbf	equ	45h
dtbf	equ	46h
sec	equ	49h
dgctr	equ	4bh
btctr	equ	4ch
loadr	equ	4dh
hiadr	equ	4eh
txctr	equ	4fh
xctrl	equ	50h
xctrh	equ	51h
xptrl	equ	52h
xptrh	equ	53h

- 9) Out of four Register Banks provided by 51 families, User can use two Banks i.e. Bank0 & Bank1. User has access to these R Banks & SFR's like A, B, DPTR, SP etc through "R" Command key. You can check, make changes & modify Registers using this command.
- 10) All address data & control signals are brought on 50 pin STD Bus.
- 11) All T/C, CLK of 8253 are brought on 14 pin polarised connector.
- 12) Interrupts of CPU are available on 50 pin Bus.
- 13) All 24 I/O pins from 8255 are brought on 26 pin Polarised FRC connector along with GND. & VCC.
- 14) 9 pin D type male connector is provided for RS232C serial communication.
- 15) 5 pin keyboard connector is provided for IBM PC type keyboard.
- 16) 2 pin Power Supply connector is provided for 5V DC @ 1A.
- 17) 16*1 or 16*2 LCD display is provided as per customer's requirement

Monitor Commands

SUBSTITUTE COMMAND (S)

The Substitute command is for to Enter the Data/Op-code in to the memory, to read the data from memory, to modify the specific memory location. This Substitute Command is operate as follows

Press S SUBSTITUTE? Invoke Substitute Command
Press Enter key & authorise this command
Press any other key except Enter & ESC

EXTERNAL RAM? INTERNAL RAM?

These options are explained in details as follows-

a) **Substitute External RAM** : It allows you to enter the data in external memory. It displays the chosen external memory address & data at that location & allows it to be dynamically changed. Pressing Enter or P key at address & data on display only enters that data in memory location. Pressing Esc brings you at command mode without entering the changed data in memory.

To invoke this command press S & Enter then

EXTERNAL RAM ? – Press Enter here to use external memory.

EX RAM ADDR 0000 – It asks for 4 digit hex address of external memory, at which you examine/ modify the data & press Enter to get display of address & data at that memory location e.g. 8000H.

EXT ADDR 8000 00 – It shows you that address & data, it can be changed by directly entering data. Otherwise pressing Enter you can see next address or pressing P you can see previous address. Pressing Esc you come to command mode.

b) **Substitute Internal Memory**:-Here user is allowed to use (examine/modify) internal memory directly. Care should be taken that not to disturb

any crucial registers, which may lead to malfunctioning of kit & your program, & user has to reset the system. Hence user is advised he should only change the contents of internal memory locations only available for user.

Press S & Enter to enter inside substitute command press any other key except ENTER & Esc till display is-

INTERNAL RAM ? – Press enter here for internal memory use.

INT RAM ADDR 00 – It asks you internal memory address (00 to FFH), if you press Enter only it starts from default address 00H, press Enter

INT ADDR 00 61 – It shows you contents of internal memory address of 00H. You can change the data here with Enter to be followed or just press Enter. It shows you next internal memory address 01H. You can see the previous address by pressing P or press Esc for command mode.

MOVE COMMAND. (M)

This command allows user to move a Memory Block data to another memory area. User can move a specified memory area by start & end address to another memory area specified by destination address i.e. the specified block is copied from destination address onwards up to the capacity given with end address. The original block remains undestroyed provided destination address does not fall within start & end address.

Press M & Enter

MOVE BLOCK? – Press Enter to use move block type

START ADDR 0000 – It asks you start address of the block, which is to be moved & press Enter e.g. 8000H.

END ADDR 0000 – It asks you end address or memory block marking which is to be moved & press Enter e.g. 80FFH.

DEST ADDR 0000 – It asks you the destination address at which the marked block is being moved to destination address & press Enter e.g. 8200H

O.K. – After block has been moved to specified destination address. Here you will find block 8000H–80FFH is copied as it is at 8200H destination address. In other word the memory contents of 8000H-80FFH & 8200H-82FFH are exactly same.

FILL COMMAND (F)

This command allows user to fill a specified memory block by start and end address with constant Data byte. This is useful for clearing the memory contents with FFH data or putting 00H (NOPs) in testing the software. Press F & Enter then you will get a following display-

FILL DATA? – Press Enter here.

DATA 00 – It asks you the constant data byte that is to be filled in specified memory block. e.g. give here 55H & Enter.

START ADDR 0000 – It asks start address of memory block where the constant data is filled. e.g. 8300H & Enter.

END ADDR 0000 – It asks you End address of block marked for filling with constant data byte e.g. 83FFH & Enter.

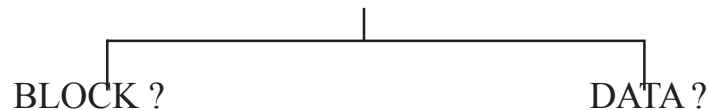
O.K.. – At the end of constant data byte in specified memory area.

Here you can find that Byte= 55H is copied in specified block of 8300H-83FFH. Press Esc to come to command mode.

COMPARE COMMAND (C)

This command allows the user to compare the two memory blocks or a constant data for specified memory block. The two options are displayed in round robin fashion.

Press C Compare? Invoke compare command
Press Enter to authorise the command.
Press any other key except Enter & Esc.



These options are explained in details as follows-

- a) **Compare Block** – This option asks the first memory block specified with start & End address & another memory block with destination address. It does not destroy any of the blocks being compared. If block contents are matching it will display O.K. but if mismatching is

found at a particular address then that error address & corresponding data is displayed on display, here if press Enter then it continues the remaining comparison. If press Esc it comes to command mode.

BLOCK ? – Press Enter to compare memory block.

START ADDR 0000 – It asks you start address of first memory block. Give 4 digit hex address & press Enter. e.g. 8000H

END ADDR 0000 – It asks you End address marking of first block. Give 4 digit hex address & press Enter. e.g. 80FFH.

DEST ADDR 0000 – It asks you the destination address of another memory block to be compared with first one. Give 4 digit hex address & press Enter. e.g. 8200H.

ERR ADDR 0000 00 – During comparison if mismatch is found that error address with data is displayed otherwise

O.K.. – The block is compared & data is matched. Pressing Esc to come to command mode.

- b) **Compare Data** – This option allows user to compare a constant data byte with data bytes in specific memory block from start & End address. Here also if any mismatch is found then that error address & mismatched data will be displayed. If data is matched then O.K. is displayed.

DATA? – Press Enter to compare data

DATA 00 – It asks for a data byte to be compared with data bytes from specified memory block. Give 2 digit hex data & Enter. e.g. 55H

START ADDR 0000 – It asks you the start address of the memory block to be compared. Give 4 digit hex address & press Enter. e.g. 8300H

END ADDR 0000 – It asks you end address of marked block. Give 4 digit hex address & press Enter. e.g. 83ffH

ERR ADDR 0000 00 – This is the mismatched address & data. Here if you press Enter to continue next comparison.

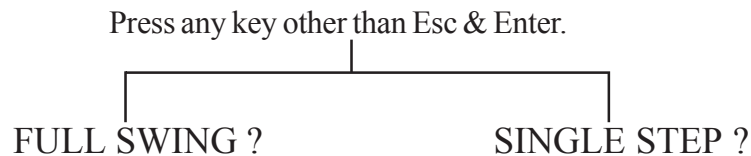
O.K. – At the end of successful comparison.

GO TO COMMAND/EXECUTE (G)/(E)

Now we will see how to execute your programs using GO TO command. This command has two options as listed below-

User can execute his program in FULL SWING mode in which the complete program is executed in single shot. In SINGLE STEP mode, in which only one instruction is executed at a time. The two options are executed in round robin fashion.

Press G/E GO TO?EXECUTE? Invoke GOTO/EXECUTE command
Press Enter to authorise this command.



These options are explained in details as follows-

- a) **FULL SWING** – This allows user to execute their program in single shot. To invoke this command press Enter after Go to.
FULL SWING? – Press Enter to authorise the command.
START ADDR 0000 – It asks you the address of user program from which the execution starts in full swing. Give here 4 digit hex address & press Enter.
BLANK DISPLAY – If user is not used display console in his program the blank display will occur. If user is using display for displaying any message then those message will appear on display.
- b) **SINGLE STEP MODE** – This mode allows user to execute each instruction in their program, this displays each instruction address & first byte on instruction. At each instruction user can verify & check the contents of external memory & can see the effect on execution after each instruction. It is highly useful in program debugging. Since INT 0 is used for single stepping, INT 0 is not available to user, but user can single step EPROM program also.

SINGLE STEP ? – Press Enter to go to single stepping.

START ADDR 0000 – It asks you the address from which user wants to execute program in single step mode. Give 4 digit hex address & then press Enter. It shows you the address of instruction & first byte at that location. To continue execution of program go on pressing Enter key.

SERIAL LINK COMMANDS

There are three commands using serial link with PC or any terminal as serial communication device.

Commands	keyboard	Description.
SERIAL IN	I	Receive on serial to kit.
SERIAL OUT	O	Send on serial from kit.
TERMINAL	T	Terminal mode enable.

These options are explained in details as follows-

- a) **SERIAL IN** – This is used to receive the data from serial link in Binary format & saves that data in specified memory block. Press I then you will observed following display-

SERIAL DATA IN? – Press Enter to invoke this command.

START ADDR 0000 – It asks you the Start address of memory location from where you want to store the data. Give 4 digit hex address & press Enter.

END ADDR 0000 – It asks you the End address of memory location. Give 4 digit hex address & press Enter.

WAIT – Now it receives data from serial link & displays wait until end of file is received or end of file is exceeded. After reception of file the display is

O.K.. – End of reception of file.

- b) **SERIAL OUT** – This is used to transmit a memory block data over serial link in Binary format. Press O then you will observe following display-

- 5) Once inside a command pressing any other than Enter/Esc on round robin basis till you press Enter key can see various options within.
- 6) Now we will see how to enter a very simple program, which adds two 8-bit numbers.

Addition of two 8-bit numbers -

Memory Add.	Op code	Label Instruction	Comment
8000	74 24	Start: Mov A, #24	Load the accumulator with 24H
8002	75 F0 42	MOVFO, #42H	Load register B with 42H
8005	25 F0	ADD A, F0	Add contents of register B to register A
8007	02 00 55	LJMP CMNDMP	Return to command mode.

Sequences of keys to be pressed along with response of trainer.

Sr. No.	Press key	Display	Description
1		LOGSUN SYSTEMS	After Power ON. Now it is a KEYCODE command mode.
2	S	SUBSTITUTE?	Use substitute command to enter Program.
3	Enter	EXTERNAL RAM?	Use substitute External memory mode.
4	Enter	EX RAM ADDR 0000	Asks memory address. Enter here
5	8		4 digit address 8000H entering 8
6	0		then 0 then 0 then 0. If you make
7	0		any mistake while entering, don't
8	0		worry, enter 4 digits again in sequence starting from the first leftmost position.
9	Enter	EXT ADDR 8000 00	The address & data are displayed here
10	7	EXT ADDR 8000 07	Enter a new data byte 74 at address 8000. Note that data

11	4	EXT ADDR 8000.74	entered in memory when you press Enter. While entering 2 digit data press both the digits in correct sequence before pressing Enter key.
12	Enter	EXT ADDR 8001 00	Address is incremented by 1 displaying corresponding data. At this step if press P address is decremented by 1, displaying corresponding data.
13	P	EXT ADDR 8000 74	You can check / modify data previously entered.
14	Enter	EXT ADDR 8001 00	Now enter remaining data bytes each followed by a Enter key until you arrive at address 8007H. You can check data again by decrementing address one by one.

Execution of the program :

Sr. No.	Press key	Display	Description
1	Esc	COMMAND :	Come to command mode.
2	G/ E	GO TO ?/EXECUTE?	Use GO TO/EXECUTE command to execute program.
3	Enter	FULL SWING?	Use go to full swing mode.
4	Enter	START ADDR 0000	Asks program address.
5	8	8	Execute program at 8000H & comes to command mode
6	0	80	
7	0	800	
8	0	8000	
9	Enter		

Checking contents of registers-

Sr. No.	Press key	Display	Description
1	R	REGISTER?	Use to check Register contents.
2	Enter	BANK 0?	Bank 0 registers are selected.
3	Any key other	SFR'S?	SFR'S are selected. Now press Enter than Esc/Enter key to see register contents.
4	Enter	A 66	Now it will shows the contents of all registers in round robin fashion if press Enter key.

Serial Communication using Hyper Terminal of Windows-95-98-2000 (Uploading & Downloading)

- 1) Use 9 to 25 or 9 to 9 pin 3 wire serial link to interface trainer with computer
- 2) At PC make hyper terminal active using following steps
 - a) In Programs select Accessories & in Accessories select Communications then in Communications select **Hyper terminal** & Click
 - b) Next select HYPERTRM & double click.
 - c) You will see Window in that enter any name you wish. Ignore the Icon Window.
- 3) Now from file menu select properties; you will get a window on your screen.
- 4) Select configure from that window; port setting window appear on screen. Please ensure that following settings are there-
 - a) Bits per second –1200/2400/4800 (you have select one from these)
 - b) Data bits-8
 - c) Parity- None
 - d) Stop bits-2
 - e) Flow control-None
- 5) If all settings are correct then close port setting window.
- 6) Now select setting properties & select ASCII set up from that window.
- 7) Be sure that following two terms are selected in ASCII sending & ASCII receiving window respectively
 - a) Send line ends with line feeds.
 - b) Wrap lines that exceed terminal width.

A) For sending data from PC to kit

- 1) Press 'I' from keyboard of kit for serial data in command.
- 2) Give four digit start address from where you want to store downloaded data & press Enter.
- 3) Give four digit End address & press Enter.
- 4) Wait is displayed on your kit. Now your kit is ready for receive data.
- 5) Now select Hyper terminal mode from PC.
- 6) Select Transfer & go to send text file.
- 7) Give file name which you want to download & click on open.
- 8) Your data is downloaded in kit.

B) For sending data from kit to PC

- 1) Go to Hyper terminal mode & select Capture text from transfer menu .
- 2) A capture text window will appear on screen. Give file name where you want to store data.
- 3) Click on start; Now your PC is ready to receive data from kit.
- 4) Now press 'O' from keyboard of kit & press enter.
- 5) Give four digit start address from where you want to send data & press Enter.
- 6) Give four digit end address & press enter.
- 7) Thus your data is stored to your file in PC.

Useful Subroutines

- 1) 0050H-**Command Mode**-Program goes back to Command mode without disturbing the register content.
- 2) 0055H- **Command Mode**-Program goes back to Command mode without disturbing the register content and showing command on display.
- 3) 005AH- **Msg-Out**- R1 Points 0 to 16 points of LCD cursor. Look up table address is loaded in DPTR. Message is displayed till the end of Text ETX (03H) Or consecutive 16 bytes are displayed from Look up Table.
- 4) 0060H-**Number Out**- contents of Accumulator i.e. numeric contents are displayed at cursor position defined by register- R1
- 5) 0065H- **Read Key**- If any key is presses, its ASCII value will stored in Accumulator & its carry is reset. If no key is pressed carry is set & Accumulator contents are random.
- 6) 0070H- Delay- This call generates 10ms delay directly. (e. g. if user wants 1sec delay, load R4 with 100 counts means 64H & call delay routine.)

e.g. MOV R4, #64h
 LOOP: LCALL DELAY
 DJNZ R4, LOOP
 RET

- 7) 0090H-Clear Display- Clears LCD Display and cursor goes back to initial position.

SAMPLE PROGRAMS

SOFTWARE EXERCISE

1) PROGRAM TO MULTIPLY TWO 8 BIT NOS.THE RESULT IS STORED

```
9000          ORG 9000H
9000 909300    MOV DPTR,#9300H    ;INITIALISE POINTER.
9003 E0        MOVX A,@DPTR      ;TAKE FIRST NO.IN A.
9004 A3        INC DPTR          ;MOVE TO NEXT LOCATION.
9005 F5F0      MOV B,A           ;STORE 1ST NO.IN B.
9007 E0        MOVX A,@DPTR      ;TAKE 2ND NO.IN A.
9008 A4        MUL AB            ;MULTIPLY BOTH NOS.
9009 120055    LCALL CMDMOD      ;JMP TO COMMAND MODE.
```

;IN THE ABOVE PROGRAM YOU HAVE TO ENTER THE NOS. ;IN 9300H AND 9301H LOCATION.AFTER THE EXECUTION ;OF THE PROGRAM THE RESULT WILL BE STORED IN ACC. ;TAKE CARE THAT RESULT SHOULD BE WITHIN 'FFH'.

2) PROGRAM TO DIVIDE TWO 8 BIT NOS.STORED AT 9310H AND 9311H.THE RESULT IS STORED IN ACC. AND REMAINDERIS STORED IN B.

```
9020          ORG 9020H
9020 909310    MOV DPTR,#9310H    ;INITIALISE POINTER AT 9310H.
9023 E0        MOVX A,@DPTR      ;TAKE FIRST NO. IN ACC.
9024 F5F0      MOV B,A           ;STORE ACC. IN B.
9026 A3        INC DPTR          ;MOVE TO NEXT NUMBER.
9027 E0        MOVX A,@DPTR      ;TAKE SECOND NO IN ACC.
9028 84        DIV AB            ;MAKE DIVISION.
9029 120055    LCALL CMDMOD      ;JUMP TO COMMAND MODE.
```

IN THIS PROGRAM YOU HAVE TO ENTER TWO NOS. AT 9310H AND 9311H. AFTER EXECUTION OF THE PROGRAM THE DIVISION WILL BE IN ACC. AND THE REMAINDER (IF ANY) WILL BE STORED IN B REGISTER.

3) PROGRAM TO FIND THE LARGEST NO FROM THE GIVEN ARRAY AND DISPLAY THE LARGEST NO AT LCD DISPLAY.

```

9040          ORG 9040H
9040 758190    MOV SP,#90H
9043 909320    MOV DPTR,#9320H    ;INITIALISE THE POINTER.
9046 780A      MOV R0,#0AH        ;MAKE R0 AS A COUNTER.
9048 752900    MOV 29H,#00H       ;MAKE INT.MEM.ADD.FOR
                                RESULT.

904B E0        NEXT: MOVX A,
               @DPTR              ;TAKE FIRST NO.
904C B52902    CJNE A,29H,NX       ;COMPARE WITH
                                MIN.NO.i.e.00H
904F 0155      AJMP SKIP           ;JUMP TO COMPARE THE
                                NOS.
9051 4002      NX:JC SKIP          ;JUMP IF CARRY IS
                                GENERATED.
9053 F529      MOV 29H,A          ;OTHERWISE STORE ACC. IN
                                INT.MEM.
9055 A3        SKIP:INC DPTR       ;MOVE TO NEXT MEM.
                                LOCATION.
9056 D8F3      DJNZ R0,NEXT        ;DECREMENT COUNTER AND
                                JUMP IF NOT=0
9058 E529      MOV A,29H          ;MOVE INT.MEM.ADD.TO
                                ACCUMULATOR.
905A 7901      MOV R1,#01H        ;SET LOCATION FOR
                                MAX.NO. DISPLAY.
905C 120060    LCALL NUMOUT        ;DISPLAY THE NO.
905F 120050    LCALL CMDMO1       ;JUMP TO COMMAND MODE.

```

IN THIS PROGRAM YOU HAVE TO ENTER TEN NUMBERS RANDOMLY ; AT 9320AH FOR NEXT 10 LOCATIONS. AFTER THE EXECUTION OF ; THE PROGRAM IN THE INTERNAL MEMORY ADD.(29H) THE LARGEST NO. ; WILL BE STORED. AND ALSO IT WILL DISPLAY ON LCD DISPLAY.

4) PROGRAM TO FIND THE SMALLEST NO. FROM THE GIVEN ARRAY AND DIPLAY THE NO.ON LCD DISPLAY.

```

9070          ORG 9070H
9070 758190    MOV SP,#90H
9073 909330    MOV DPTR,#9330H    ;INITIALISE MEM.POINTER.
9076 780A      MOV R0,#0AH        ;MAKE COUNTER FOR
                                TEN NOS.
9078 7CFF      MOV R4,#0FFH       ;LOAD LARGEST NO.
                                IN INT.MEM.
907A E0        GO: MOVX A,@DPTR   ;MOVE THE FIRST NO IN A.
907B B50402    CJNE A,04,SWP       ;COMPARE WITH
                                LARGEST NO.
907E 0183      AJMP YES           ;JUMP TO GET SMALL NO.
9080 5001      SWP:JNC YES         ;JUMP IF NO CARRY IS
                                GENERATED.
9082 FC        MOV R4,A           ;MOVE THE A TO INT.
                                MEM. ADD.
9083 A3        YES:INC DPTR       ;ICREAMENT THE POINTER.
9084 D8F4      DJNZ R0,GO          ;DECREMENT R0 AND JUMP
                                IF NOT=0.
9086 EC        MOV A,R4           ;MOVE INT. MEM.ADD.
                                TO ACCU.
9087 7901      MOV R1,#01H        ;SET POSITION FOR
                                MIN.NO.DIPLAY.
9089 120060    LCALL NUMOUT        ;DISPLAY THE SMALLEST NO.
908C 120050    LCALL CMDMO1       ;JUMP TO COMMAND MODE.

```

IN THIS PROGRAM YOU HAVE TO ENTER THE ARRAY OF TEN DIFFERENT NUMBERS AT 9330H MEMORY LOCATION FOR NEXT 10 LOCATIONS(i.e.UPTO 9339H). AFTER THE EXECUTION OF THE PROGRAM THE SMALLEST NUMBER WILL BE STORED AT ACC.AND IT IS ALSO DISPLAYED ON THE LCD DISPLAY.

5) PROGRAM TO TRANSFER THE BLOCK FROM ONE MEMORY LOCATION TO ANOTHER MEMORY LOCATION.

```

9090      ORG 9090H
9090 758190  MOV SP,#90H      ;INITIALISE STACK POINTER.
9093 909340  MOV DPTR,#9340H  ;INITIALISE SOURCE POINTER
9096 7B50    MOV R3,#50H      ;STORE LOWER ADD. OF
                        DESTI.ADD.
9098 7C93    MOV R4,#93H      ;STORE HIGH.ADD.
                        OF DESTI.ADD.
909A 7A0A    MOV R2,#0AH      ;SET COUNTER FOR TEN NOS.
909C E0      LOOP:MOVX A,
@DPTR      ;MOVE FIRST NO. IN ACC.
909D C083    PUSH DPH         ;PUSH HIGHER BYTE
                        IN STACK .
909F C082    PUSH DPL         ;PUSH LOWER BYTE IN
STACK 90A1 8B82  MOV DPL,R3   ; L O A D
LOW.ADD.      OF DESTI.ADD.IN DPTR.
90A3 8C83    MOV DPH,R4      ;LOAD HIGH.ADD.OF
                        DESTI.ADD.IN DPTR.
90A5 F0      MOVX @DPTR,A    ;MOVE ACC.TO POINTER. .
90A6 D082    POP DPL         ;POP LOWER ADD. FROM
                        STACK.
90A8 D083    POP DPH         ;POP HIGHER ADD. FROM
                        STACK .
90AA A3      INC DPTR        ;INCREMENT POINTER.

```

```

90AB 0B      INC R3          ;INCREMENT R3.
90AC BB0001  CJNE R3,#00,SKIP1 ;JUMP IF NOT=00.
90AF 0C      INC R4          ;INCRENT R4.
90B0 DAEA    SKIP1:DJNZ R2,LOOP ;DECREMENT R2 AND JUMP
                        IF NOT=00.
90B2 120055  LCALL CMDMOD    ;JUMP TO COMMAND MODE.

```

;IN THE ABOVE PROGRAM YOU HAVE TO EDIT TEN NOS.IN THE SOURCE POINTER(i.e.AT 9340H) AND AFTER THE EXECUTION OF THE PROGRAM THE ARRAY OF TEN NOS. WILL BE COPIED ON THE DESTI. ADDR. i.e.AT 9350H).

6) PROGRAM TO ARRANGE THE NOS.AT THE GIVEN MEMORY LOCATION IN ASSENDING ORDER.

```

90C0      ORG 90C0H
90C0 7C00    MOV R4,#00H      ;INITIALISE STACK AT 90H.
90C2 758150  MOV SP,#50H      ;LOAD 00H TO R4 REGISTER.
90C5 7E94    MOV R6,#94H      ;LOAD DOWN CUNTR.IN R5.
90C7 7A00    MOV R2,#00H      ;LOAD HIGHER BYTE
                        OF MEM.POINTER.
90C9 909400  START1:MOV DPTR,
#9400H      ;LOAD FIRST NO.IN DPTR.
90CC E0      WAIT:MOVX A,@DPTR ;STORE FIRST NO. IN ACC.
90CD FD      MOV R5,A         ;MAKE BACKUP FOR
                        FIRST NO.
90CE A3      INC DPTR         ;MOVE TO NEXT NUMBER.
90CF E0      MOVX A,@DPTR     ;STORE NEXT NO.IN ACC.
90D0 FF      MOV R7,A         ;MAKE BACKUP FOR
                        NEXT NO.
90D1 9D      SUBB A,R5        ;MAKE SUBTRACTION OF
                        TWO NOS.
90D2 400E    JC XCHNG         ;IF CARRY SETS, JUMP TO
                        EXCHANGE

```

90D4 0A	INC R2	;INCREMENT R2.
90D5 EA	MOV A,R2	
90D6 BA0AF3	NEX:CJNE R2, #0AH, WAIT	;BE IN LOOP UNTILL R2=69H.
90D9 7A00	MOV R2,#00H	;INCENENT R4.
90DB 0C	INC R4	
90DC BC0AEA	CJNE R4,#0AH,START1	;BE IN LOOP UNTILL R4=0AH.
90DF 120055	LCALL CMDMOD	;IF R4=0AH JUMP TO COMMAND MODE.
90E2 EF	XCHNG:MOV A,R7	;LOAD SECOND NO. IN ACC.
90E3 C083	PUSH DPH	;PUSH LOW.BYTE OF DPTR IN STACK.
90E5 C082	PUSH DPL	;PUSH HIGH.BYTE OF DPTR IN STACK.
90E7 8A82	MOV DPL,R2	;LOAD FIRST LOCATION IN DPTR.
90E9 8E83	MOV DPH,R6	;LOAD FIRST LOCATION IN DPTR.
90EB F0	MOVX @DPTR,A	;LOAD SECOND NO. IN FIRST LOCATON
90EC D082	POP DPL	;RESTORE SECOND LOCATION IN DPTR.
90EE D083	POP DPH	;RESTORE SECOND LOCATION IN DPTR.
90F0 ED	MOV A,R5	;LOAD FIRST NO. IN ACC.
90F1 F0	MOVX @DPTR,A	;STORE FIRST NO.IN SECOND LOCATION.
90F2 0A	INC R2	;INCREMENT R2.
90F3 1290D6	LCALL NEX	;JUMP TO GET NEXT NOS FROM POINTER.

IN THE ABOVE PROGRAM YOU HAVE TO FILL TEN NOS.AT 9400H LOCATION. AFTER EXECUTION OF THE PROGRAM THE TEN NOS

WOULD BE ARRANGED IN ASSENDING ORDER AT THE SAME LOCATON.

7) PROGRAM TO ARRANGE NOS IN DESENDING ORDER

9100	ORG 9100H	
9100 7C00	MOV R4,#00H	;INITIALISE STACK AT 90H.
9102 758150	MOV SP,#50H	;LOAD 00H TO R4 REGISTER.
9105 7E95	MOV R6,#95H	;LOAD DOWN CUNTR.IN R5.
9107 7A00	MOV R2,#00H	;LOAD HIGHER BYTE OF MEM.POINTER.
9109 909500	START:MOV DPTR ,#9500H	;LOAD FIRST NO.IN DPTR.
910C E0	WAIT1:MOVXA ,@DPTR	;STORE FIRST NO. IN ACC.
910D FD	MOV R5,A	;MAKE BACKUP FOR FIRST NO.
910E A3	INC DPTR	;MOVE TO NEXT NUMBER.
910F E0	MOVXA,@DPTR	;STORE NEXT NO.IN ACC.
9110 FF	MOV R7,A	;MAKE BACKUP FOR NEXT NO.
9111 9D	SUBB A,R5	;MAKE SUBTRACTION OF TWO NOS.
9112 500D	JNC XCNG	;IF CARRY SETS, JUMP TO EXCHANGE
9114 0A	HEX:INC R2	;INCREMENT R2.
9115 BA0AF4	CJNE R2,#0AH, WAIT1	;BE IN LOOP UNTILL R2=0AH.
9118 7A00	MOV R2,#00H	;RESET R2 TO 00H.
911A 0C	INC R4	;INCENENT R4.
911B BC0AEB	CJNE R4,#0AH,START	;BE IN LOOP UNTILL R4=0AH.
911E 120055	LCALL CMDMOD	;IF R4=0AH JUMP TO COMMAND MODE.
9121 EF	XCNG:MOV A,R7	;LOAD SECOND NO. IN ACC.
9122 C083	PUSH DPH	;PUSH LOW.BYTE OF DPTR IN STACK.

9124 C082	PUSH DPL	;PUSH HIGH.BYTE OF DPTR IN STACK.
9126 8A82	MOV DPL,R2	;LOAD FIRST LOCATION IN DPTR.
9128 8E83	MOV DPH,R6	;LOAD FIRST LOCATION IN DPTR.
912A F0	MOVX @DPTR,A	;LOAD SECOND NO. IN FIRST LOCATON.
912B D082	POP DPL	;RESTORE SECOND LOCATION IN DPTR.
912D D083	POP DPH	;RESTORE SECOND LOCATION IN DPTR.
912F ED	MOV A,R5	;LOAD FIRST NO. IN ACC.
9130 F0	MOVX @DPTR,A	;STORE FIRST NO.IN SECOND LOCATION.
9131 129114	LCALL HEX	;JUMP TO GET NEXTNOS FROM POINTER.

IN THE FOLLOWING PROGRAM YOU HAVE TO FILL THE TEN NOS.AT 9500H.AFTER EXECUTION OF THE PROGRAM THE NOS WOULD BE ARRANGED IN DESENDING ORDER AT THE SAME LOCATION.

8) PROGRAM FOR ADDITION OF 2 DIGITS NUMBERS.

9140	ORG 9140H	;PROGRAM SHOULD BE OPERATED AT 9140H.
9140 758190	MOV SP,#90H	;INITIALISE STACK POINTER.
9143 7A22	MOV R2,#22H	;1ST BYTE TO BE ADDED.
9145 7B26	MOV R3,#26H	;2ND BYTE TO BE ADDED.
9147 7901	MOV R1,#01H	;SET CURSOR POSITION.
9149 EA	MOV A,R2	;NUMBER TO BE DISPLAYED.
914A 120060	CALL NUMOUT	;DISPLAY ON LCD DISPLAY.
914D 7904	MOV R1,#04H	;SET CURSOR POSITION.
914F EB	MOV A,R3	;2ND DIGIT TO DISPLAYED.
9150 120060	LCALL NUMOUT	;DISPLAY ON LCD DISPLAY.
9153 EA	MOV A,R2	;READ FIRST BYTE TO

		BE ADDED.
9154 2B	ADD A,R3	;ADD 2ND BYTE.
9155 FA	MOV R2,A	;R2+R3=ADD.
9156 7400	MOV A,#00H	;MAKE ACC.ZERO.
9158 3400	ADDC A,#00H	;IF CARRY AFTER ADDITION INCREMENT R3.
915A FB	MOV R3,A	;MOVE ACC.INTO R3.
915B 7907	MOV R1,#07H	;CURSOR POSITION OF RESULT.
915D 120060	LCALL NUMOUT	;DISPLAY THE NUMBER.
9160 EA	MOV A,R2	;STORE R2 IN ACC.
9161 120060	LCALL NUMOUT	;DISPLAY THE NUMBER.
9164 120055	LCALL CMDMOD	;WAIT FOR ANY KEY PRESS.SAVE ALL REGISTERS & GO TO COMMAND PROMPT.

EXECUTION OF THIS PROGRAM ILLUSTRATES THE ADDDDTION OF TWO DIGIT DECIMAL NUMBER.THE PROGRAM IS WRITTEN AND EXECUTED AT 9140H. THE RESULT WILL BE DISPLAY ON LCD DISPLAY IN THE SAEQUENCE OF NUM1, NUM2 & ADDITION.

9) PROGRAM FOR SUBTRACTION OF 2 DIGITS NUMBERS.

9170	ORG 9170H	;PROGRAM IS TO BE ENTERED AT 9170H.
9170 758190	MOV SP,#90H	;INITIALISE STACK POINTER.
9173 7A47	MOV R2,#47H	;1ST BYTE.
9175 7B34	MOV R3,#34H	;2ND BYTE.
9177 7901	MOV R1,#01H	;SET CURSOR POSITION.
9179 EA	MOV A,R2	;1ST BYTE TO BE DISPLAYED.
917A 120060	LCALL NUMOUT	;DISPLAY ON LCD DISPLAY.
917D 7904	MOV R1,#04H	;SET CURSOR POSITION.
917F EB	MOV A,R3	;2ND BYTE TO BE DISPLAYED.
9180 120060	LCALL NUMOUT	;DISPLAY ON LCD DISPLAY.
9183 C3	CLR C	;CLEAR C.

9184 EA	MOV A,R2	;R2-R3=RESULT.
9185 9B	SUBB A,R3	;SUBTRACT R3 FROM ACC.
9186 FA	MOV R2,A	;MOVE SUBTRACTION TO R2.
9187 7400	MOV A,#00H	;CLEAR ACC.
9189 9400	SUBB A,#00H	;SUBTRACT 00 FROM ACC.
918B FB	MOV R3,A	;MOVE ACC. TO R3.
918C 7907	MOV R1,#07H	;STORE 07 IN R1.
918E 120060	LCALL NUMOUT	;DISPLAY THE NUMBER.
9191 7909	MOV R1,#09H	;MOVE 09 IN R1.
9193 EA	MOV A,R2	;MOVE R2 TO ACC.
9194 120060	LCALL NUMOUT	;DISPLAY THE NUMBER.
9197 120055	LCALL CMDMOD	;IF ANY KEY PRESS, SAVE ALL ;REGISTERS & GO TO COM MAND PROMPT.

EXECUTION OF THIS PROGRAM ILLUSTRATES THE SUBTRACTION OF TWO DIGIT DECIMAL NUMBERS. THE PROGRAM IS WRITTEN AND EXECUTED AT 9170H. THE RESULT WILL BE DISPLAY ON LCD DISPLAY IN THE SEQUENCE OF NUM1 NUM2 & SUBTRACTION.

10) PROGRAM FOR ROLLING DISPLAY THIS PROGRAM DISPLAYS A ROLLING MESSAGE "LOGSUN SYSTEMS PUNE--" ON LCD DISPLAY TILL ANY KEY IS PRESSED. THIS PROGRAM ILLUSTRATES USE OF SOME OF THE MONITOR UTILITIES LIKE MSGOUT, DELAY ETC. USER CAN TRY FOR ANOTHER MESSAGE ALSO. THE PROGRAM WRITTEN AND EXECUTED AT 91A0H.

91A0	ORG 91A0H	;PROGRAM IS TO BE ENTERED AT 91A0H.
91A0 758190	MOV SP,#90H	;INITIALISE THE STACK.
91A3 9091D4	RLDISP:MOV DPTR, #LK_LGSYS	;POINTERS TO MESSAGE.
91A6 AA82	MOV R2,DPL	;STORE DPTR TEMPORARY
91A8 AB83	MOV R3,DPH	
91AA 7900	LOP:MOV R1,#00H	;START POSITION OF DISPLAY.

91AC 12005A	LCALL MSGOUT	;DISPLAY MESSAGE.
91AF 1291CC	LCALL DE_1SEC	;DELAY 1SEC.
91B2 120090	LCALL CLRDISP	;CLEAR THE DISPLAY.
91B5 8B83	MOV DPH,R3	;INCREAMENT MESSAGE STACK POINTER.
91B7 8A82	MOV DPL,R2	;TO GET ROLLING EFFECT.
91B9 A3	INC DPTR	;INCREAMENT DPTR
91BAAA82	MOV R2,DPL	;SHIFT LOW BYTE IN R2.
91BC AB83	MOV R3,DPH	;SHIFT HIGH BYTE IN R3.
91BE E0	MOVX A,@DPTR	;STORE DPTR IN ACC.
91BF B40302	CJNE A,#03H,CNT_RL	;COMPARE WITH 03H.
91C2 80DF	SJMP RLDISP	;DISP. CHAR. STRING TILL END OF TEXT.
91C4 120065	CNT_RL:LCALL POLKEY	;CALL SUBROUTINE.
91C7 40E1	JC LOP	;IF CARRY, REPEAT.
91C9 120055	LCALL CMDMOD	;GO TO COMMAND PROMPT.
91CC 7C64	DE_1SEC:MOV R4, #64H	;LOAD CONUT FOR 100 NOS.
91CE 120070	LP_DL_1SEC:LCALL DELAY_10MS;CALL DELAY_10MSEC.	
91D1 DCFB	DJNZ R4,LP_DL_1SEC	;REPEAT TILL ZERO.
91D3 22	RET	;RETURN.
91D4 20204C4F	FLK_LGSYS: DB ' LOGSUN SYSTEMS PUNE ----',03H	
91D8 4753554E		
91DC 20535953		
91E0 54454D53		
91E4 2050554E		
91E8 45202D2D		
91EC 2D2D03		

THE EXECUTION OF THE PROGRAM WILL ROLL THE MESSAGE "LOGSUN SYSTEMS PUNE ----" CONTINEOUSLY UNTILL "Esc" BUTTON FROM KEY BOARD IS PRESSED. END

HARDWARE EXERCISES

1) PROGRAM TO CHECK THE PORT 1 OF 8051. IN THE FOLLOWING PROGRAM THE EXECUTION WILL GIVE THE SQUARE WAVE AT THE PORT1.

```

8000          ORG 8000H
8000 758190    MOV SP,#90H          ;INITIALISE THE STACK.
8003 759055    START5:MOV P1,#55H;LOAD 55 TO PORT 1.
8006 128011    LCALL DELAY          ;CALL DELAY SUBROUTINE.
8009 7590AA    MOV P1,#0AAH         ;LOAD AA ON PORT 1.
800C 128011    LCALL DELAY          ;CALL DELAY SUBROUTINE.
800F 80F2      SJMP START5          ;CONTINUE THE PROCESS.
8011 78FF      DELAY:MOV R0,#0FFh;SET RO AT FFH.
8013 79FF      MOV R1,#0FFH         ;SET R1 AT FFH
8015 D9FE      lp2:DJNZ R1,lp2       ;DECREMENT R1 AND JMP.
8017 D8FE      lp1:DJNZ R0,lp1       ;DECREMENT RO AND JMP.
8019 22        RET

```

2) PROGRAM TO OBTAIN THE SQUARE WAVE AT PORT A OF 8255, HAVING TIME PERIOD 1 Sec.

```

8100          ORG 8100H
8100 7480      MOV A,#80H           ;INIT.ALL PORT AS O/P.
8102 900003    MOV DPTR, #0003H    ;LD CWR ADD.IN DPTR.
8105 F0        MOVX @DPTR,A         ;LD ACC. AT DPTR.
8106 7400      CK:MOV A,#00H        ;LD 00HI ACC.
8108 900000    MOV DPTR,#0000H      ;LD PORT A ADD.
810B F0        MOVX @DPTR,A         ;LD ACC. IN DPTR.
810C 120070    LCALL DELAY_10ms     ;CALL DELAY OF 10ms
810F 74FF      MOV A,#0FFH          ;LD FFHIN ACC.
8111 900000    MOV DPTR,#0000H      ;LD PORT A ADD.
8114 F0        MOVX @DPTR,A         ;LD ACC. IN DPTR.
8115 120070    LCALL DELAY_10ms     ;CALL DELAY OF 10ms.
8118 80EC      SJMP CK              ;JUMP TO CONTINUE

```

3) PROGRAM TO CHECK THE OPEARATON OF INTERNAL TIMERS OF THE CONTROLLER, BY GENERATING SQUARE WAVE ON THE PORT1 OF 8051.

```

8200          ORG 8200H
8200 759000    STRT:MOV P1,#00H     ;SET PORT 1 TO 00H.
8203 7405      MOV A,#05H           ;LOAD DELAY MSB TO ACC.
8205 75F005    MOV B,#05H           ;LOAD DELAY LSB TO B REG.
8208 128218    LCALL HRD_TIME       ;CALL HARD WARE DELAY.
820B 7590FF    MOV P1,#0FFH        ;SET PORT 1 TO FFH
820E 7405      MOV A,#05H           ;LOAD DELAY MSB TO ACC.
8210 75F005    MOV B,#05H           ;LOAD DELAY LSB TO B REG.
8213 128218    LCALL HRD_TIME       ;CALL HARD WARE DELAY.
8216 80E8      SJMP STRT            ;CUNTINUE THE PROCESS.
8218 5388CF    HRD_TIME:ANL TCON,
                #0CFH              ;SET TIMER CONTROL.
821B 5389F0    ANL TMOD,#0F0H       ;SET TIMER MODE.
821E 438901    ORL TMOD,#01H        ;SET MODE OF TIMER .
8221 758A18    CHEK:MOV TL0,#18H    ;LOAD 18 TO TL0.
8224 758CFC    MOV TH0,#0FCH        ;LOAD FC TO TH0.
8227 438810    ORL TCON,#10H        ;LOGICAL OR WITH 10.
822A 108D02    WET:JBC TF0,OFLAG    ;JUMP ON OVERFLOW.
822D 80FB      SJMP WET             ;RETURN.
822F 5388EF    OFLAG:ANL TCON,
                #0EFH              ;AND WITH TCON.
8232 14        DECA                 ;DECREMENT ACC.
8233 B4FF02    CJNE A,
                #0FFH,NOROLL        ;IF NOT FFH, JUMP.
8236 15F0      DEC B                 ;DECREMENT B REG.
8238 B400E6    NOROLL:CJNE A,
                #00H,CHEK           ;IF NOT 00H, JUMP.
823B 22        RET

```

4) PROGRAM TO CHECK OPERATION OF INTERRUPT WITH THE MICRO CONTROLLER.THE EXECUTION OF THE PROGRAM WILL GIVE THE SQUARE WAVE AT THE PORT 1(P1.0) OF 8051 WITH TIME PERIOD OF 1mSec.

```

8300          ORG 8300H          ;
8300 758190    MOV SP,#90H        ;INIT. STACK.
8303 C28C      CLR TCON.4         ;CLEAR TCON.4.
8305 758CD0    MOV TH0,#0D0H      ;SET TH0.
8308 758AF0    MOV TL0,#0F0H      ;SET TL0.
830B 75A800    MOV IE,#00H        ;SET IE REG.
830E D28C      SETB TCON.4        ;SET TCON.4
8310 00        LOP3:NOP          ;
8311 308DFC    JNB TCON.5,LOP3    ;IF BIT RESETS,JUMP
8314 C28D      CLR TCON.5         ;CLEAR TCON.4.
8316 B290      CPL P1.0           ;COMPLEMENT PORT 1.0
8318 80F6      SJMP LOP3         ;RETURN.

                        ;END

```

Program to check the interrupt operation

- 1) Load the program
- 2) Connect the LED logic card to the port 1
- 3) See the status of port 1 LED's
- 4) Give the interrupt (connect the INT1 pin to the ground)
- 5) See the status of port 1 LED's

```

ORG 8000H
8000 75A884    MOV IE,#84H        ;Enable external interrupt INT1
8003 759055    MOV P1,#55H        ;P1=55h
80006 80FE    HERE: SJMP HERE    ;Stay here until get interrupted

```

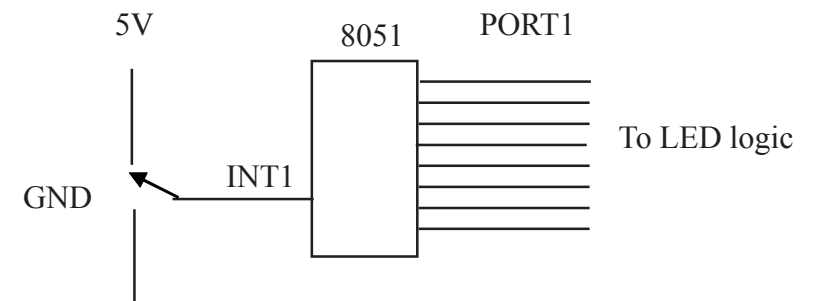
ISR for hardware interrupt INT1 to change the status of port 1

```

ORG FF13H    ;INT1 ISR
FF13 7590AA    MOV P1,#AAH        ;P1=AAh
FF16 32        RETI              ;return from ISR

```

In the above program the microcontroller is looping continuously in the here loop. Whenever the INT1 pin connects to ground the INT1 activated the controller gets out of loop & jumps to vector location FF13. The ISR for INT1 will change the status of LEDs



CONNECTOR DETAILS

CONN. 50 PIN STD BUS

Connector 8255

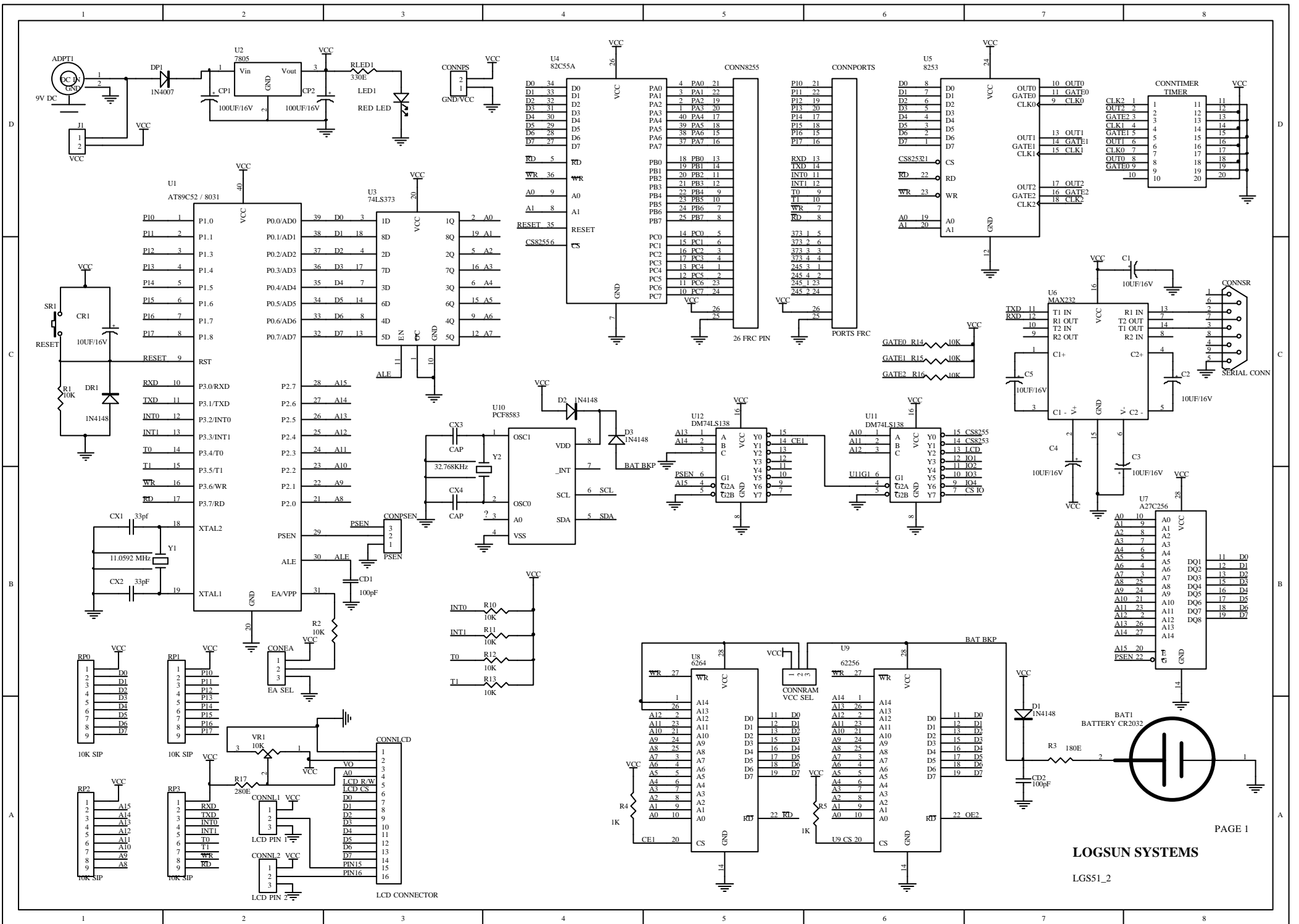
Pin No.	Pin Details	Pin No.	Pin Details	Pin No.	Pin Details	Pin No.	Pin Details
01	VCC	02	GND	01	PC4	02	PC5
03	BD1	04	BD0	03	PC2	04	PC3
05	BD3	06	BD2	05	PC0	06	PC1
07	BD5	08	BD4	07	PB6	08	PB7
09	BD7	10	BD6	09	PB4	10	PB5
11	RXD	12	TXD	11	PB2	12	PB3
13	INT 0	14	INT 1	13	PB0	14	PB1
15	T 0	16	T 1	15	PA6	16	PA7
17	INT 1	18	PSEN	17	PA4	18	PA5
19	NC	20	NC	19	PA2	20	PA3
21	RD	22	NC	21	PA0	22	PA1
23	ALE	24	WR	23	PC6	24	PC7
25	ADDR DIR	26	BA0	25	GND	26	Vcc.
27	BA 15	28	BA 1	8031/51 CONNECTOR			
29	BA 14	30	BA 2				
31	BA 13	32	BA 3				
33	BA 12	34	BA 4				
35	BA 11	36	BA 5				
37	BA 10	38	BA 7				
39	BA 9	40	BA 6				
41	BA 8	42	NC				
43	RESET	44	CS I/O				
45	DATADIR	46	IO3				
47	NC	48	IO 4	01	I/P 1	02	I/P 2
49	VCC	50	GND	03	O/P1	04	O/P2
				05	O/P 3	06	O/P4
				07	WR	08	RD
				09	T0	10	T 1
				11	INT 0	12	INT 1
				13	RXD	14	TXD
				15	P1.6	16	P1.7
				17	P1.4	18	P1.5
				19	P1.2	20	P1.3
				21	P1.0	22	P 1.1
				23	I/P 3	24	I/P 4
				25	GND	26	VCC

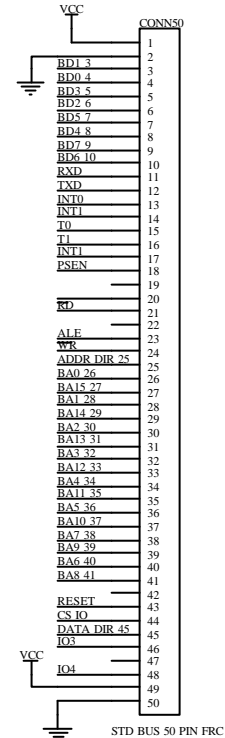
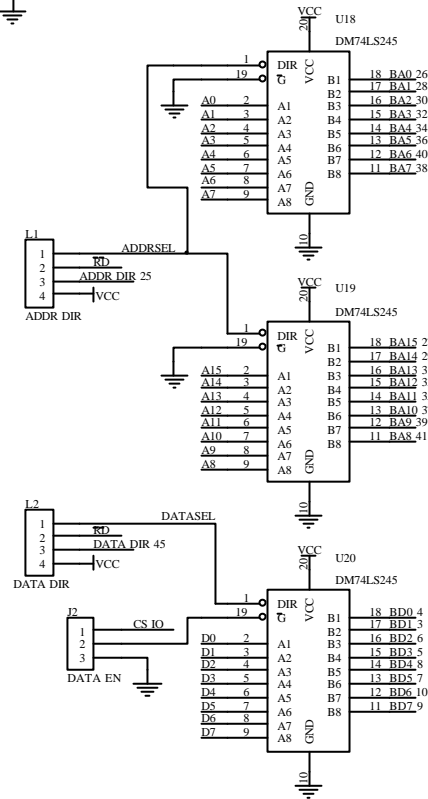
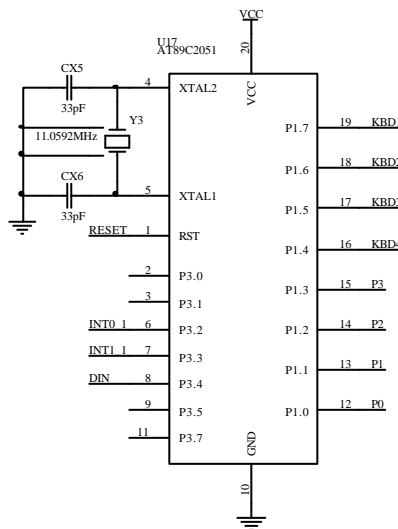
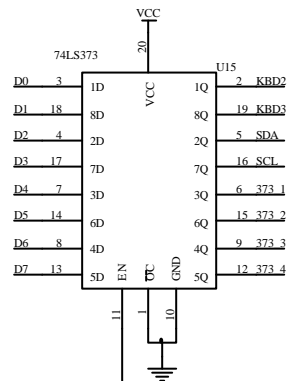
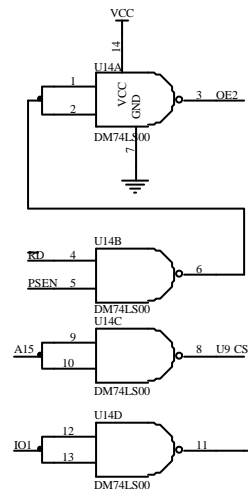
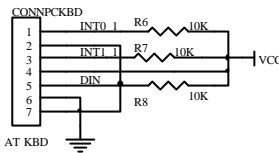
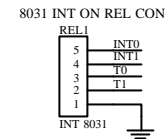
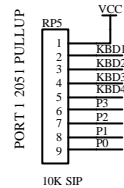
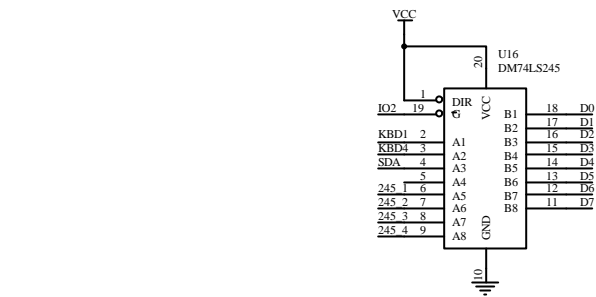
REFERENCE BOOKS-

- 1) The 8051 microcontroller and Embedded systems - Muhammad Ali Mazidi.
- 2) The 8051 Microcontroller Architecture, Programming & Applications - Kenneth J. Ayala.
- 3) The 8051 Microcontroller and Embedded Systems - Ajay Deshmukh.
- 4) Embedded/ Real-Time Systems: Concepts, Design & Programming - Dr. K.V.K.K. Prasad.

WEBSITES-

- 1) www.8051.com
- 2) www.embeddedworld.com
- 3) www.nxp.com
- 4) www.keil.com
- 5) [www.esacademy.com/software/flash magic](http://www.esacademy.com/software/flash%20magic)





[illegible]