Yashwantrao Chavan College of Science, Karad

Department of Electronics

Departmental Facilities

Manuals

INDEX

Yashwantrao Chavan College of Science ★ Karad ★
ELECTRONICS

HOD

HEAD
Department of Electronics
Yashwantrao Chavan College of Science,
Karad

Principal

Principal
Yashwantrao Chavan College of Science, Karad

YASHWANTRAO CHAVAN COLLEGE OF SCIENCE, KARAD

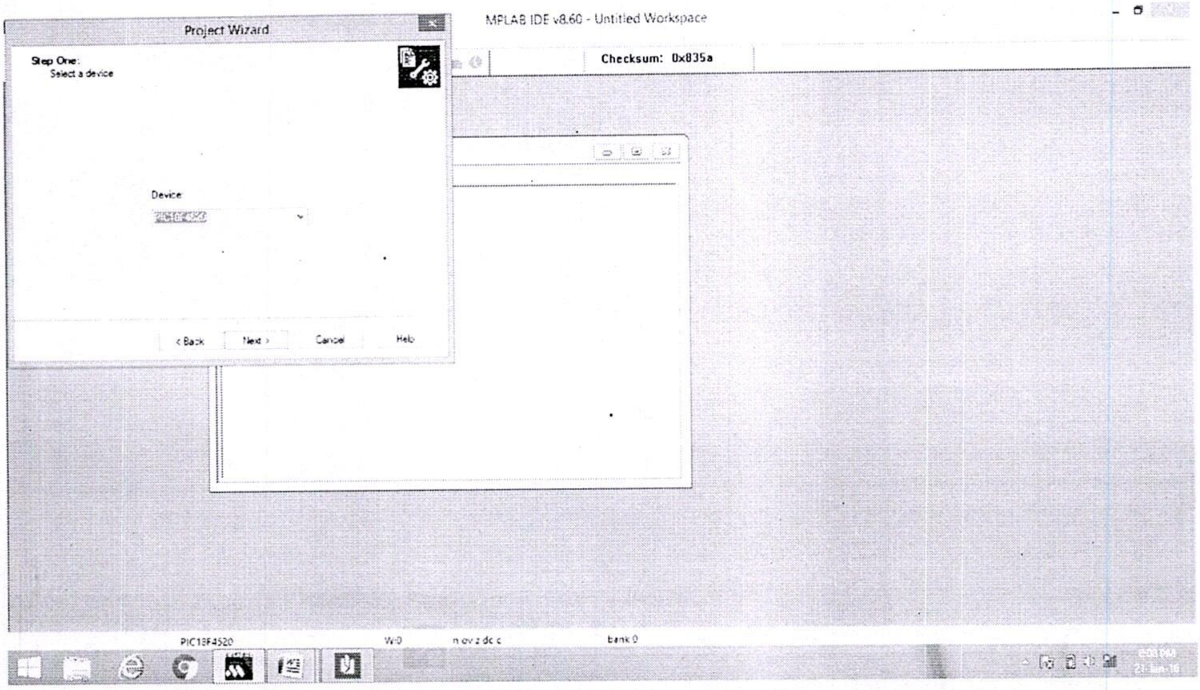
Department of Electronics

MPLAB- Software Manual

Developed By : A. A. Mulla

HEAD
Department of Electronics
Yashwantrao Chavan College of Science,
Karad

Principal
Yashwantrao Chavan College of Science, Karad



MPLAB IDE v8.60 - Untitled Workspace

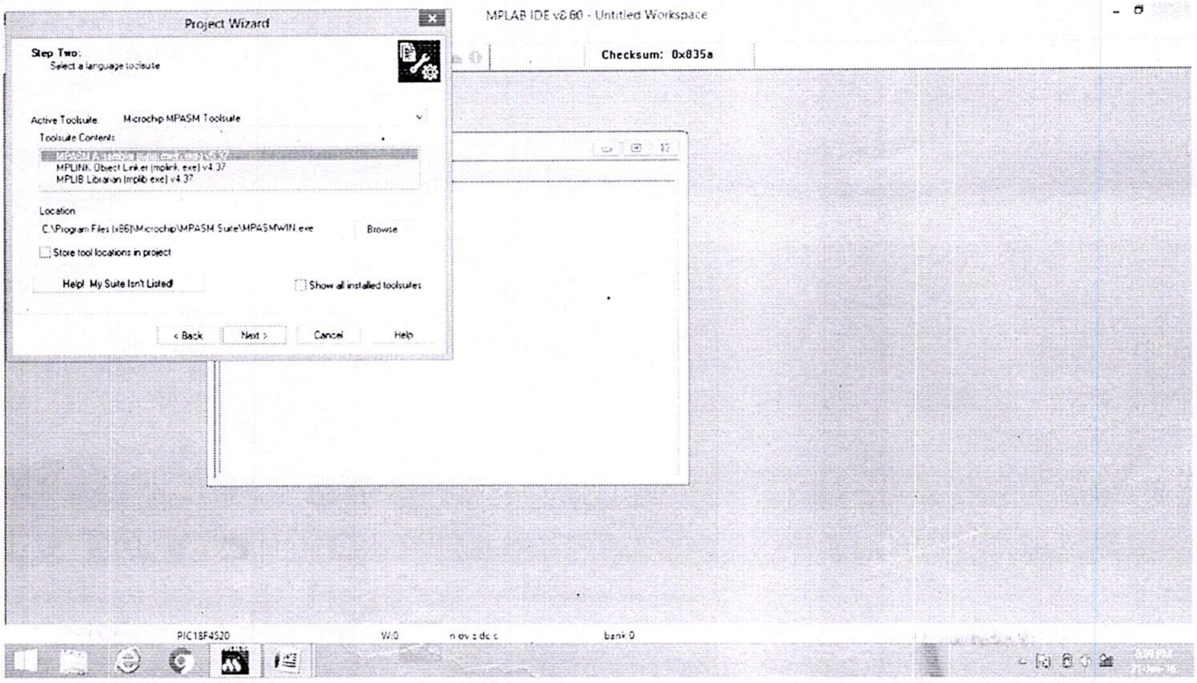
Checksum: 0x835a

Project Wizard

Step One:
Select a device

Device: PIC18F4520

PIC18F4520   W:0   n ov z dc c   bank 0



MPLAB IDE v8.60 - Untitled Workspace

Checksum: 0x835a

Project Wizard

Step Two:
Select a language toolsuite

Active Toolsuite: Microchip MPASM Toolsuite

Toolsuite Contents

MPASM Assembler (mpasmwin.exe) v5.37
MPLINK Object Linker (mplink.exe) v4.37
MPLIB Librarian (mplib.exe) v4.37

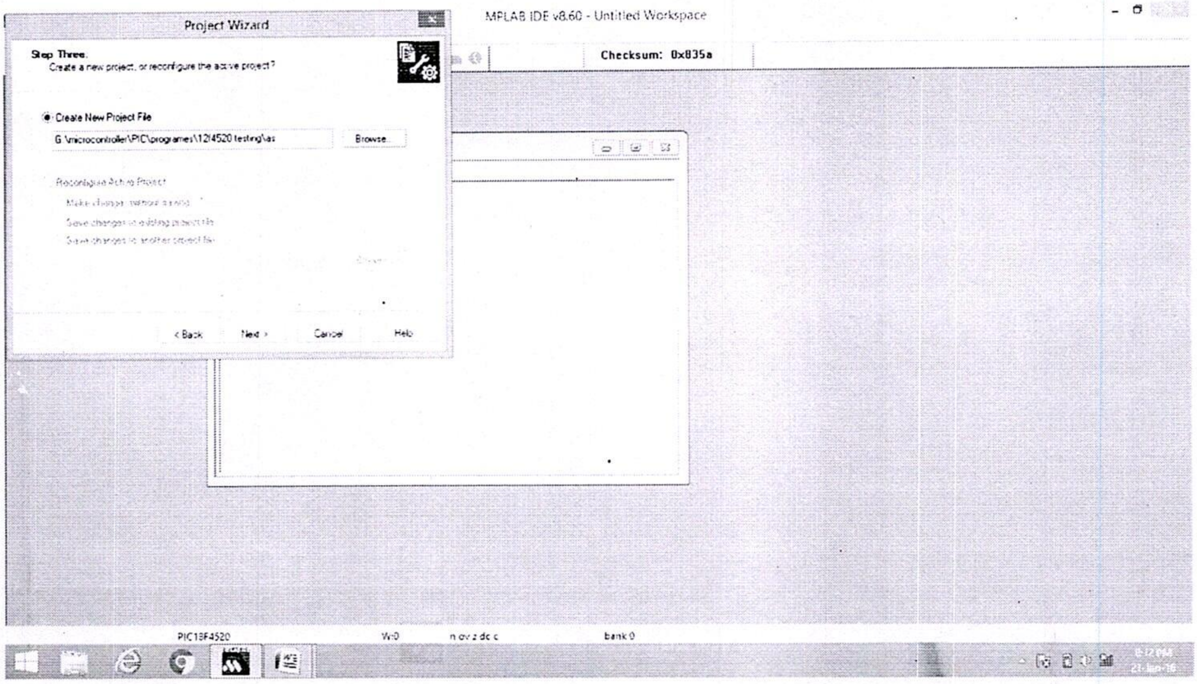
Location

C:\Program Files (x86)\Microchip\MPASM Suite\MPASMWIN.exe

□ Store tool locations in project

□ Show all installed toolsuites

PIC18F4520   W:0   n ov z dc c   bank 0



MPLAB IDE v8.60 - Untitled Workspace

Checksum: 0x835a

Project Wizard

Step Three:
Create a new project, or reconfigure the active project?

● Create New Project File

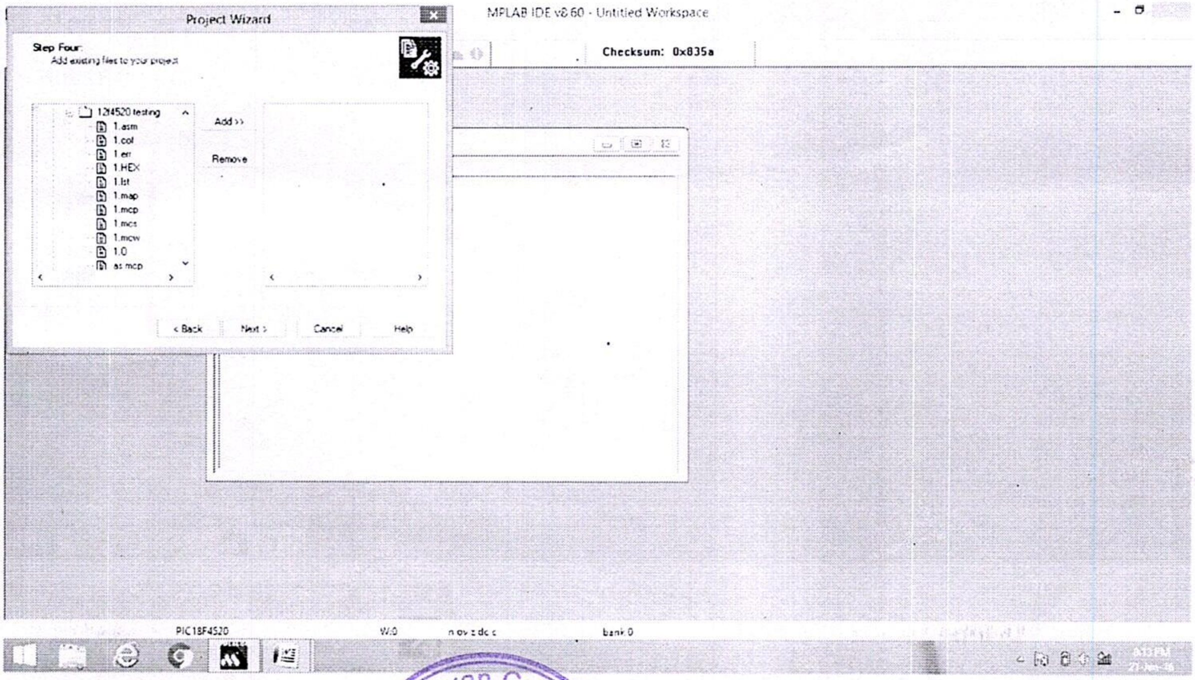
G:\microcontroller\PIC\programes\12f4520 testing\as

○ Reconfigure Active Project

Make changes without saving

Save changes to existing project file

Save changes to another project file



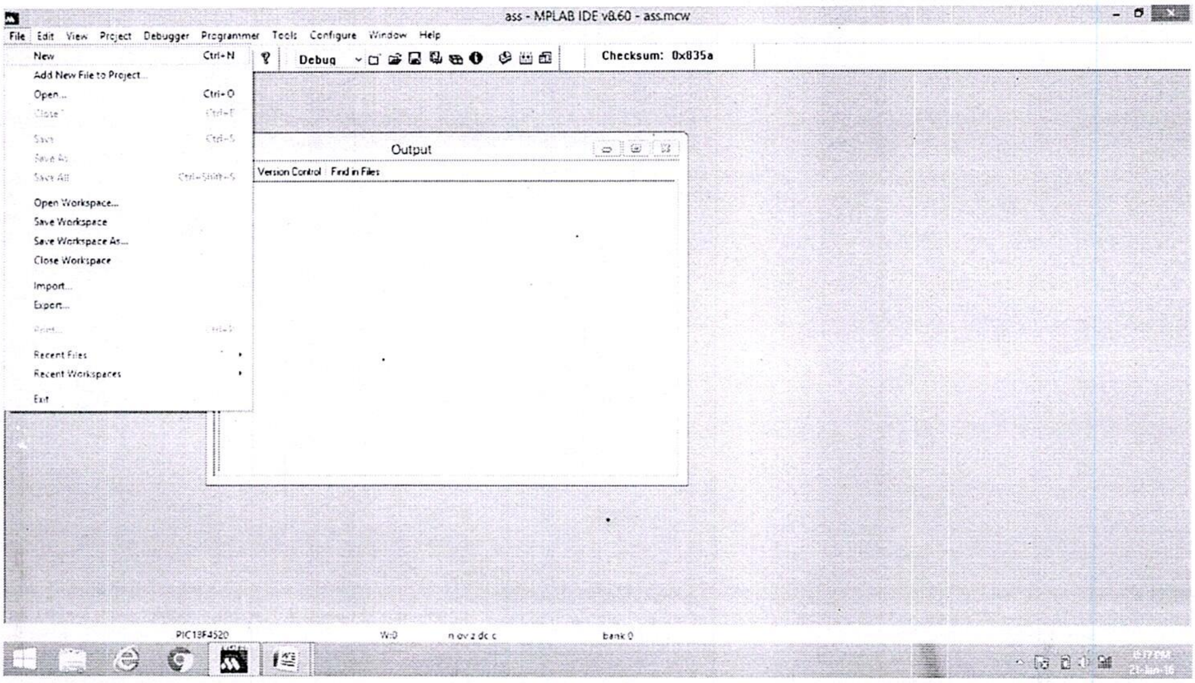
MPLAB IDE v8.60 - Untitled Workspace

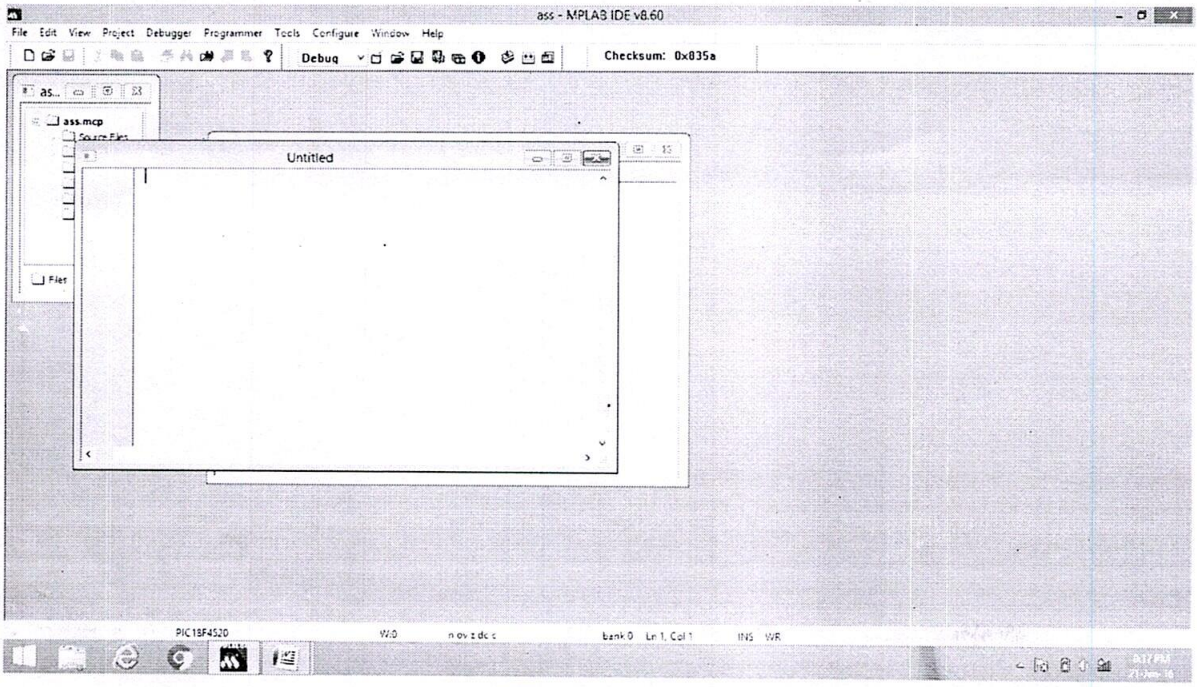
Checksum: 0x835a

Project Wizard

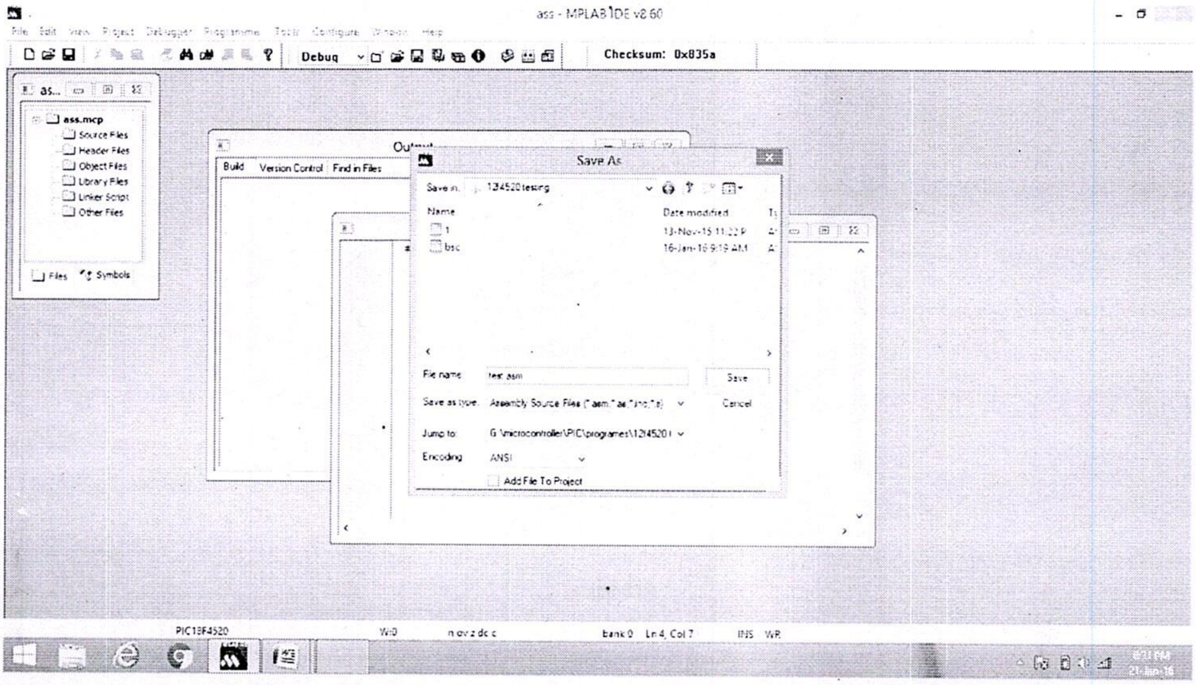
Step Four:
Add existing files to your project

- 12f4520 testing
  - 1.asm
  - 1.cof
  - 1.err
  - 1.HEX
  - 1.lst
  - 1.map
  - 1.mcp
  - 1.mcs
  - 1.mcw
  - 1.0
  - as.mcp





Yashwantrao Chavan College of Science
ELECTRONICS
★ Karad ★



Save in: 12f4520 testing

Name	Date modified
1	13-Nov-15 11:22 P
bsc	16-Jan-16 9:19 AM

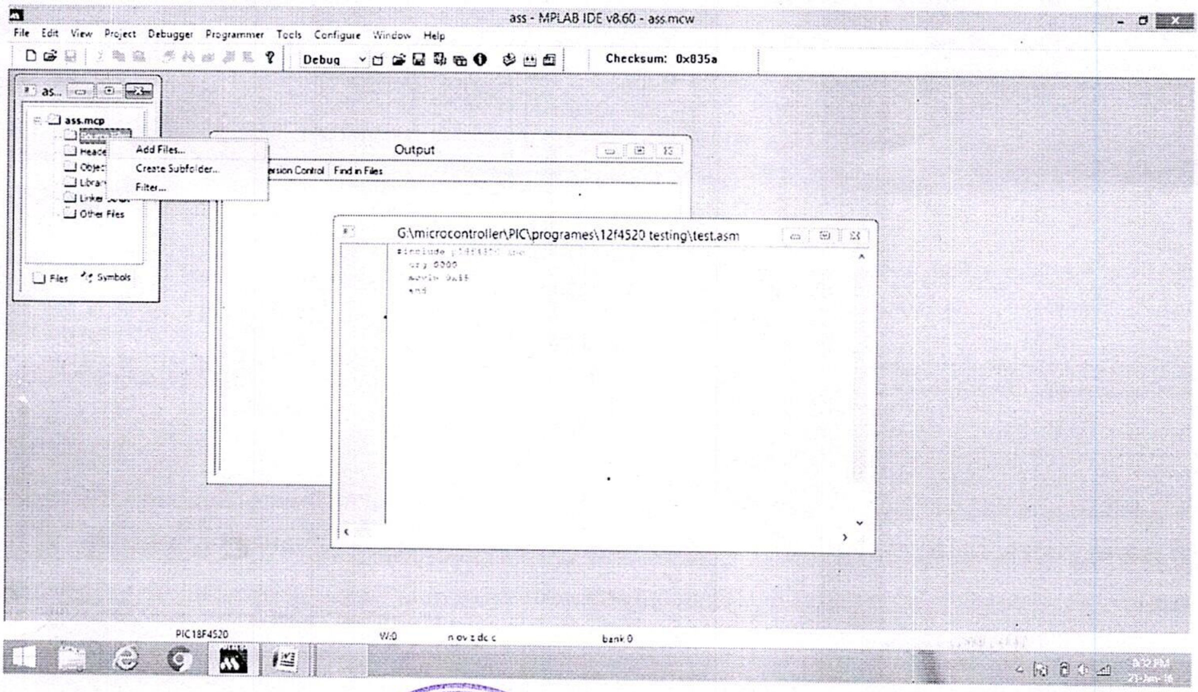
File name: test.asm

Save as type: Assembly Source Files (*.asm;*.as;*.inc;*.s)

Jump to: G:\microcontroller\PIC\programes\12f4520 t

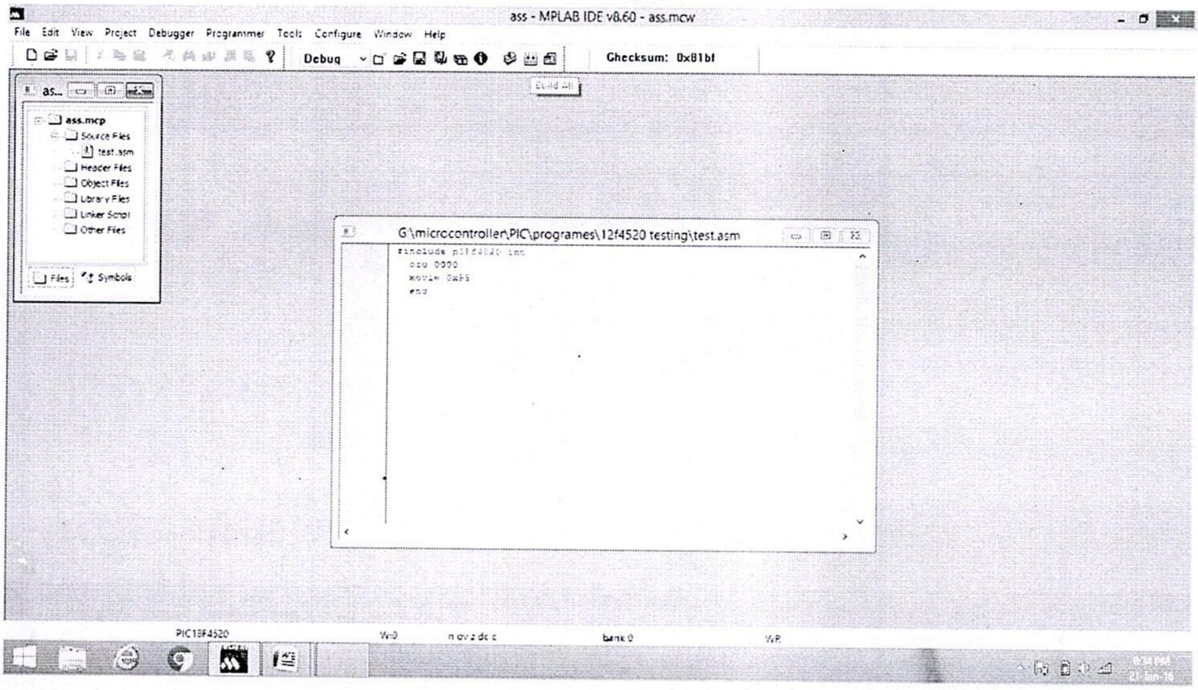
Encoding: ANSI

□ Add File To Project



G:\microcontroller\PIC\programes\12f4520 testing\test.asm

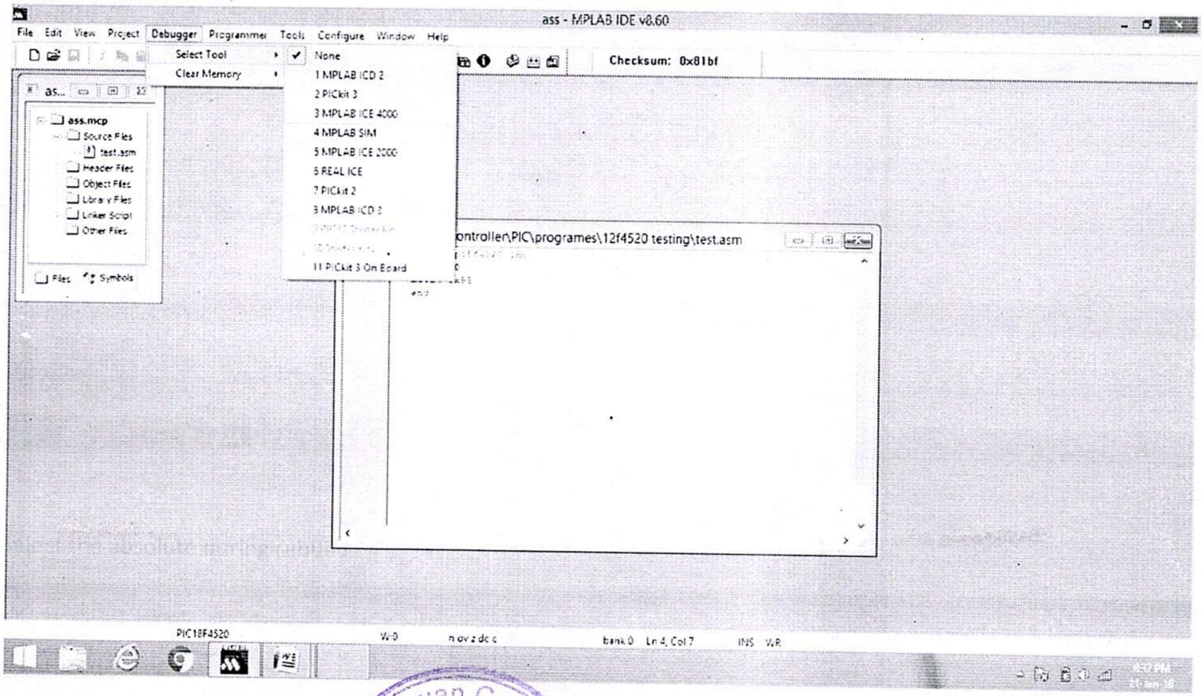
```
#include p18f4520.inc
  org 0000
  movlw 0x55
  end
```



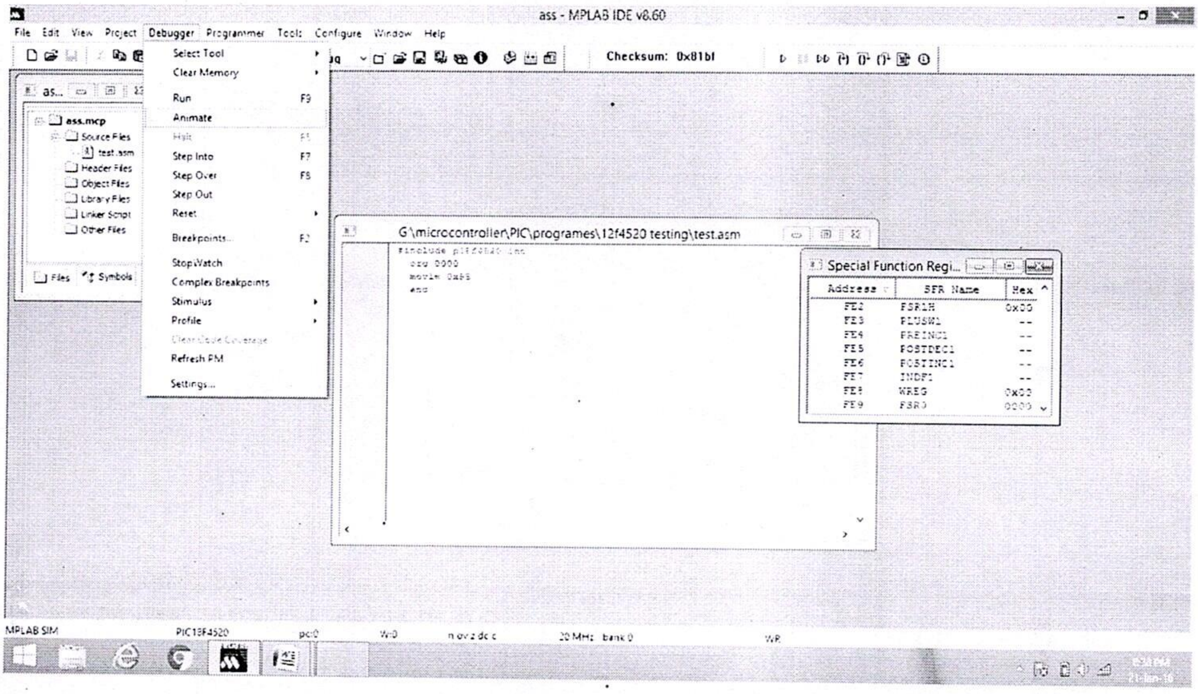
G:\microcontroller\PIC\programes\12f4520 testing\test.asm

```
#include p18f4520.inc
  org 0000
  movlw 0x55
  end
```

Select the absolute during rebuild



Yashwantrao Chavan College of Science
ELECTRONICS

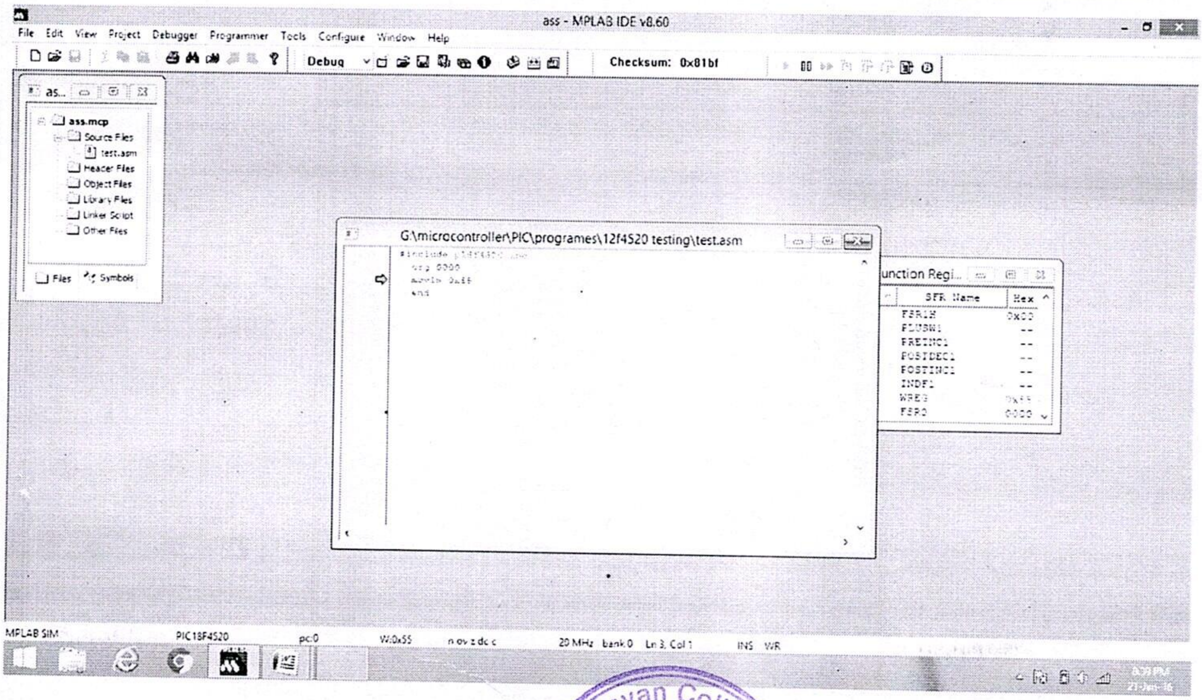


G:\microcontroller\PIC\programes\12f4520 testing\test.asm

```
#include p18f4520.inc
  org 0000
  movlw 0x55
  end
```

Special Function Regi...

Address	SFR Name	Hex
FE2	FSR1H	0x00
FE3	PLUSW1	--
FE4	PREINC1	--
FE5	POSTDEC1	--
FE6	POSTINC1	--
FE7	INDF1	--
FE8	WREG	0x00
FE9	FSR0	0000



G:\microcontroller\PIC\programes\12f4520 testing\test.asm

```
#include p18f4520.inc
  org 0000
  movlw 0x55
  end
```

unction Regi...

SFR Name	Hex
FSR1H	0x00
PLUSW1	--
PREINC1	--
POSTDEC1	--
POSTINC1	--
INDF1	--
WREG	0x55
FSR0	0000

Yashwantrao Chavan College of Science
ELECTRONICS
★ Karad ★

HEAD
Depa... ...lectronics
Yashwan... ...avan College of Science,
Karad

USER'S MANUAL

FOR

89V51RD2 DEVELOPMENT BOARD

Manufactured By

LOGSUN SYSTEMS

B-2, RAUNAK, 134/4+3B, NEAR MAYUR COLONY,

KOTHRUD, PUNE-411038

Ph.-020 24356456/25410129/32937840

Email:-logsun@eth.net

www.logsun.com

1. INTRODUCTION

Logsun's PR51-V4 development have 8-bit 89V51RD2. A Microcontroller based evaluation module. PR51-V4 is a general-purpose development board for RD2 microcontroller. PR51-V4can be used extensively to test and validate Programs. At the heart of the development board is 89V51RD2P, this provides advance features like ISP, I2C and IAP. The microcontroller has 64KB internal flash memory and 1KB on-chip RAM. The development board comes with RS-232 interface to allow user to program the micro controller directly from PC. .PR51-V4 board and related software routines help the system designers to rapidly design and prototype their Designs based on RD2 Core. It provides a complete development platform with Different Modules interface that accelerates the task of designers to run application software on target RD2 Controller hardware, thus providing a platform to benchmark their system, save time & expense of building their own application test board and enabling them to get their designs to market quickly.PR51-V4 is a unique hardware and software combination providing designers, the tools to develop most advanced .PR51-V4 series Microcontroller applications. The PR51-V4 hardware reference and software application programs also simplify RD2 based hardware and software development.

5. PERIPHERALS

- 16 X2 Lines Character LCD Display.
- Two RS232 Serial Interfaces
- Real Time Clock Interface
- Serial EPROM Interface

All the peripherals of LGS-RD2 are implemented as independent module. Only the required ports of the Micro controller are brought out on the connector for the ease of developer so that any port can be connected to any module with a very small footprint.

5.1 Liquid crystal display: In LGS-RD2 LCD is given in the form of plug and play. LCD can be connected to the Microcontroller through the port 0. LCD is Connected in the 4-bit mode or 8-bit mode. And the standard subroutine is given with the LGS- RD2 so that the application can be easily demonstrated and also for further implementation the subroutine can be easily embedded for which one has to do very few changes. Wide range of instruction functions: Clear displays, cursor home, display ON/OFF, cursor ON/OFF, cursor shift, display shift.

NOTE : SW2 is used to select a 4-bit or 8-bit mode of LCD.

Keep all pins of SW2 in ON position for 8-bit mode and install the links J2, J4, J5 in the 1 & 2 position.
For 4 bit mode install the links J2, J4, J5 in the 2 & 3 position.

6. GETTING STARTED

PR51-V4 Includes:

- LGS- PR51-V4 Development Board.
- NULL MODEM Serial Cable.
- Power Adaptor.
- System CD-ROM.
- Power Supply Requirements PR51-V4 Development Board is provided with +5Vdc . The board has a 5V regulators on-board, which provides supply to the entire devices on the board.

Configuring the system: Connect the Serial cable provided to the RS232 connector on PR51-V4 board & COM1/COM2 of the Computer. Plug in Power adaptor in 230VAC mains socket & connect the +5Vpin of power adaptor to the PR51-V4 board, the board is now ready to be switched on.

SR.NO.	HEX addr.	Port Name	Assigned Hardware of project 51 card.
1.	90H to 94H	P1.0 to P1.4	Not used
2.	95H to 97H	P1.5 to P1.7	Used for ISP
3.	80H to 87H	P0.0 to P0.7	For 8-bit LCD
4.	A1H	P2.1	SDA (Serial Data) For I2C Communication
5.	A0H	P2.0	SCL (Serial Clock) For I2C Communication
6.	A3H	P2.3	RS For 8-bit LCD
7.	A4H	P2.4	R/W For 8-bit LCD
8.	A5H	P2.5	EN For 8-bit LCD
9.	A2H	NC	Not used
10.	A6H to A7H	NC	Not used
11.	B0H	P3.0	RXD
12.	B1H	P3.1	TXD
13.	B2H	P3.2	Not used
14.	B3H	P3.3	Not used
15.	B4H to B7H	P3.4 to P3.7	Not used
16.	84H to 87H	P0.4 to P0.7	For 4-bit LCD
17.	80H	P0.0	RS For 4-bit LCD
18.	81H	P0.1	R/W For 4-bit LCD
19.	82H	P0.2	EN For 4-bit LCD

8. HARDWARE DESCRIPTIONS

***Note*: Above connection details are given for reference, if you are using ON-BOARD facilities viz. LCD, IO Expansion. If the user is not using above-mentioned facilities, all the ports are available free to use, as per design of user. (Refer block diagram for more details).**

LIST OF I2C ADDRESS OF DEVICES CONNECTED TO I2C BUS

RTC PCF 8583 Write	**0A2H**
RTC PCF 8583 Read	**0A3H**
Serial EPROM AT24C04 Read	**0A1H**
Serial EPROM AT24C04 Write	**0A0H**

►**For RTC 8583**: To change the time and date of the RTC 8583.Make changes in the program wherever the provision for changing the time and date is given. Then assemble the program again. Now, again transfer the new ". HEX " to the kit. At the time of changing, hold **P1.0** connector to ground and apply reset simultaneously.

7.1 FRC CONNECTORS

FH4

Pin no	Detail
1	P3.0
2	P3.1
3	P3.2
4	P3.3
5	P3.4
6	P3.5
7	P3.6
8	P3.7
9	VCC
10	GND

FH2

Pin no	Detail
1	P1.0
2	P1.1
3	P1.2
4	P1.3
5	P1.4
6	P1.5 ISP
7	P1.6 ISP
8	P1.7 ISP
9	VCC
10	GND

FH1

Pin no	Detail
1	P0.0
2	P0.1
3	P0.2
4	P0.3
5	P0.4
6	P0.5
7	P0.6
8	P0.7
9	VCC
10	GND

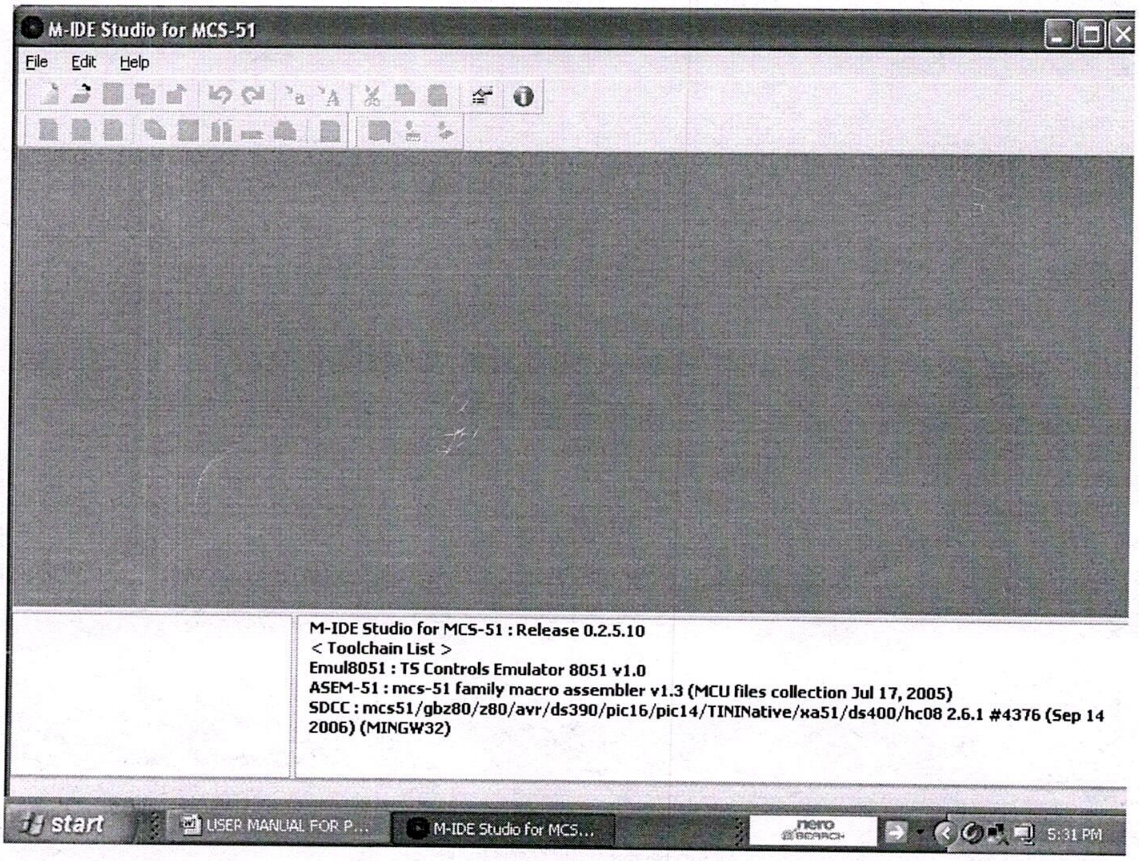
FH3

Pin no	Detail
1	P2.0
2	P2.1
3	P2.2
4	P2.3
5	P2.4
6	P2.5
7	P2.6
8	P2.7
9	VCC
10	GND

J9

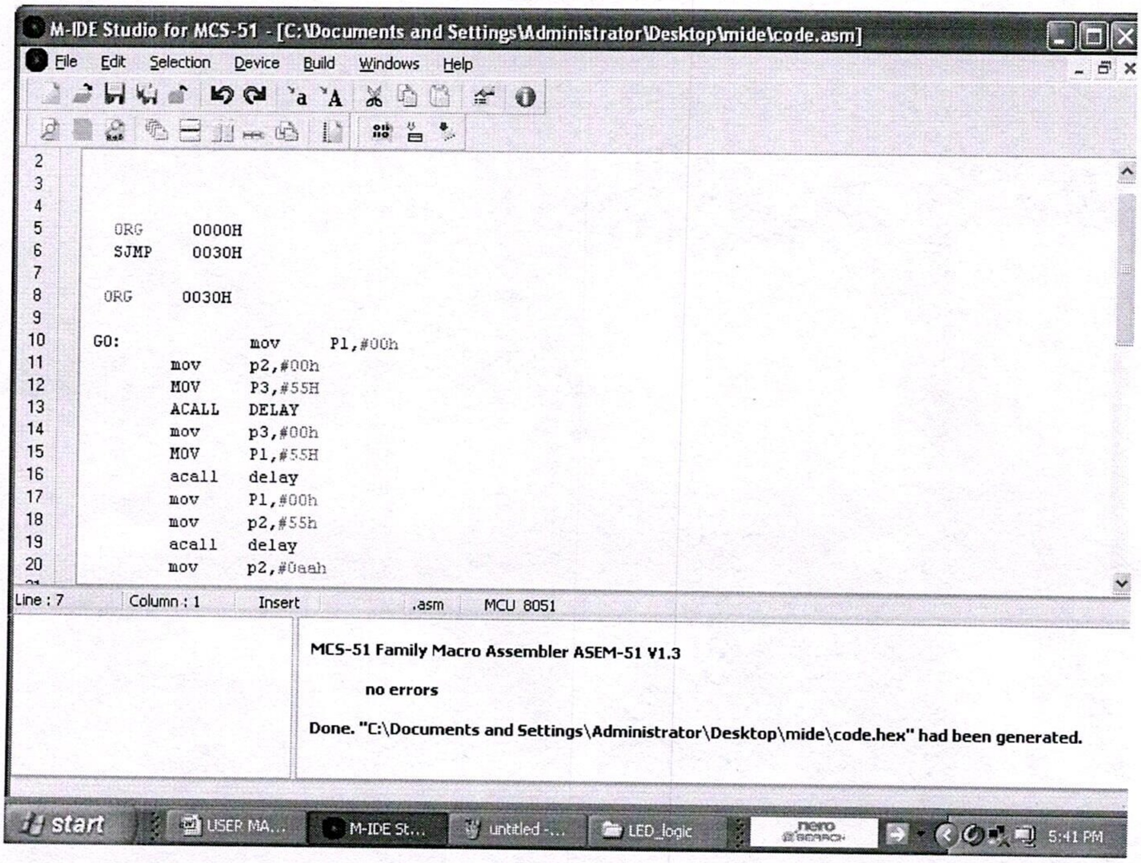
Pin No	Detail
1	GND
2	RST
3	PSEN
4	ALE

1-ASSEMBLER AND SIMULATOR MIDE(MICROCONTROLLER INTEGRATED DEVELOPMENT ENVIRONMENT)
Step 1- Open MIDE Compiler



```
M-IDE Studio for MCS-51 : Release 0.2.5.10
< Toolchain List >
Emul8051 : TS Controls Emulator 8051 v1.0
ASEM-51 : mcs-51 family macro assembler v1.3 (MCU files collection Jul 17, 2005)
SDCC : mcs51/gbz80/z80/avr/ds390/pic16/pic14/TININative/xa51/ds400/hc08 2.6.1 #4376 (Sep 14
2006) (MINGW32)
```

Step3- Go to file and select save as option. We can save our project any location with extension main.asm format. Then go to build option and select build . We can see our output on our respective location in hex format.



```
  ORG      0000H
  SJMP     0030H

 ORG      0030H

GO:              mov      P1,#00h
         mov      p2,#00h
         MOV      P3,#55H
         ACALL    DELAY
         mov      p3,#00h
         MOV      P1,#55H
         acall    delay
         mov      P1,#00h
         mov      p2,#55h
         acall    delay
         mov      p2,#0aah
```

```
MCS-51 Family Macro Assembler ASEM-51 V1.3

        no errors

Done. "C:\Documents and Settings\Administrator\Desktop\mide\code.hex" had been generated.
```

2.Now browse * .Hex File that is to be down load.

3. Keep **SW3** Switch in up position for serial download and down position for USB downloading .

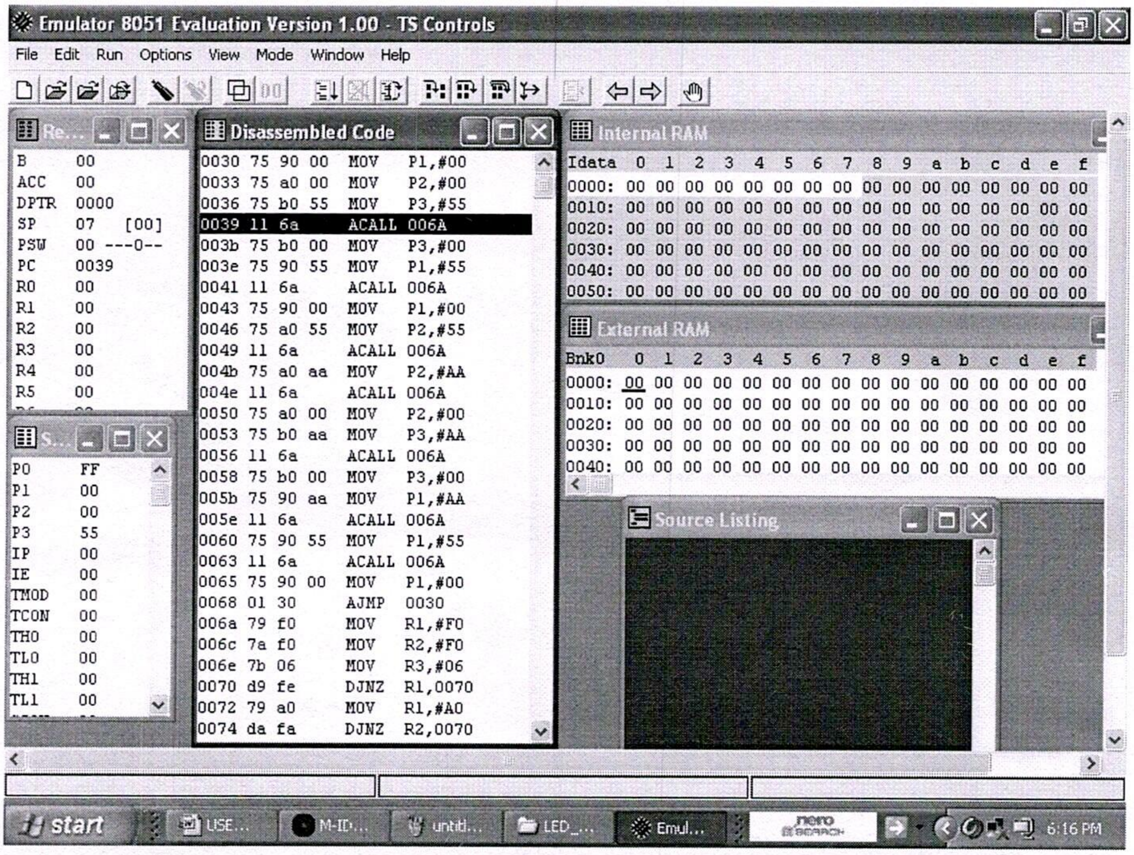
4. Click on 'start' for down loading the program .

5. Switch ON the power supply through switch.

wait until window shows the 'Finish' message.

6. Reset the controller by pressing **SW1**. Your program will be executed now.

Step 5- For seeing step by step execution go to run option and select step otherwise we can use F11 key on keyboard.



```
B       00
ACC     00
DPTR    0000
SP      07   [00]
PSW     00 ---0--
PC      0039
R0      00
R1      00
R2      00
R3      00
R4      00
R5      00
```

```
P0      FF
P1      00
P2      00
P3      55
IP      00
IE      00
TMOD    00
TCON    00
TH0     00
TL0     00
TH1     00
TL1     00
```

```
0030 75 90 00  MOV   P1,#00
0033 75 a0 00  MOV   P2,#00
0036 75 b0 55  MOV   P3,#55
0039 11 6a     ACALL 006A
003b 75 b0 00  MOV   P3,#00
003e 75 90 55  MOV   P1,#55
0041 11 6a     ACALL 006A
0043 75 90 00  MOV   P1,#00
0046 75 a0 55  MOV   P2,#55
0049 11 6a     ACALL 006A
004b 75 a0 aa  MOV   P2,#AA
004e 11 6a     ACALL 006A
0050 75 a0 00  MOV   P2,#00
0053 75 b0 aa  MOV   P3,#AA
0056 11 6a     ACALL 006A
0058 75 b0 00  MOV   P3,#00
005b 75 90 aa  MOV   P1,#AA
005e 11 6a     ACALL 006A
0060 75 90 55  MOV   P1,#55
0063 11 6a     ACALL 006A
0065 75 90 00  MOV   P1,#00
0068 01 30     AJMP  0030
006a 79 f0     MOV   R1,#F0
006c 7a f0     MOV   R2,#F0
006e 7b 06     MOV   R3,#06
0070 d9 fe     DJNZ  R1,0070
0072 79 a0     MOV   R1,#A0
0074 da fa     DJNZ  R2,0070
```

```
Idata  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
0000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
Bnk0   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
0000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

J1

Pin No	Detail
1	GND
2	EA/VPP
3	VCC

J13

Pin No	Detail
1	VCC
2	A0 of RTc
3	1 of J10

J6

Pin No	Detail
1	3 of J11
2	Battery of RTC
3	O/P

Important : J2, J4 , J5 links setting for 4-bit and 8-bit LCD

J2

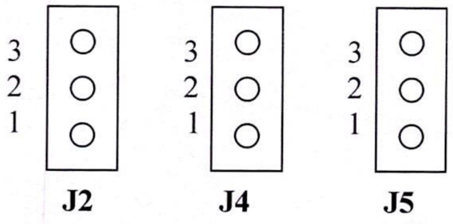
Pin No	Detail
1	P2.3
2	RS
3	LCDP0.0

J5

Pin No	Detail
1	P2.5
2	EN
3	LCDP0.2

J4

Pin No	Detail
1	P2.4
2	R/W
3	LCDP0.1



MICROPROCESSOR TRAINER

Model - LGS-85L

OPERATIONAL MANUAL

LOGSUN SYSTEMS
PUNE-411038
PHONE .- 020 - 24356456,32937840
Email - sales@logsun.com/support@logsun.com
www.logsun.com

INDEX

MODEL - LGS - 85 U
Table of Command Keys

Sr.no.	Keys	Description
1	SUB	Substitute Data into RAM/ Read Data from Memory, to check the register
2	COMP	Compare Block/Data in Memory
3	PREV	To Check Data of previous location in Substitute Command
4	MOV	Move Block/Data from memory within memory.
5	EXE	To Execute the program in full swing or single step
6	SR I/P	Serial Data in from PC or any terminal
7	SR O/P	Serial Data out to PC or any Terminal
8	ESC	Return to Command mode
9	ENTER	To proceed to next step in any above command.
10	USER	key 1 - available to user/ future expansion
11	USER	key 2- available to user/ future expansion.

HARDWARE DETAILS

1)	0000H To 3FFFH-16K	Monitor EPROM (Program Memory)
2)	4000H To 7FFFH-8K	Scratch Pad RAM Available to user, after 4100H.
3)	8000H To FFFFH-32K	User RAM Battery backup.
4}	23H	CW for 8255
	20H	Port A
	21H	Port B
	22H	Port C
5)	43H	CW for 8253
	40H	Channel 0
	41H	Channel 1
	42H	Channel 2
6)	E0H	Spare

12) All registers are available to user in SUB command.

13) All address data & control signals are brought on 50 pin FRC Connector (FRC1) Buffer STD Bus

14) All T/C, CLK of 8253 are brought on 10 pin Reliamate Connector REL2.

15) Interrupts of CPU are available on 7 pin Reliamate Connector REL1

16) All 24 I/O pins from 8255 are brought on 26 pin Polarised FRC connector (FRC2) along with GND. & VCC.

18) 9 pin D type male connector is provided for RS232C serial communication.

19) 26 pin keyboard connector (FRC3) is provided for 27 keys keyboard & 8 digit 7 segment display.

20) 2 pin/Jack type Power Supply connector is provided for 5V DC @ 1.5A.

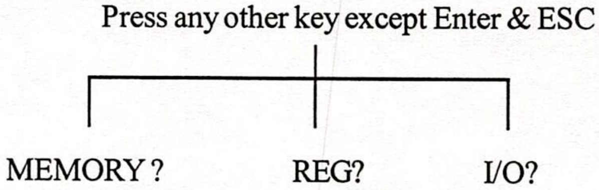
21) Reset key is provided on CPU card for CPU Reset.

Monitor Commands

SUBSTITUTE COMMAND (SUB)

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 SUB SUBSTITUTE? Invoke Substitute Command
Press Enter key & authorise this command



Press any other key except Enter & ESC

MEMORY ? REG? I/O?

These options are explained in details as follows-

a) **Substitute Meory**: 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 Prev 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 Sub & Enter then

MEMORY? – Press Enter here to use external memory.

ADDR – 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.

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 PREV you can see previous address. Pressing Esc you come to command mode.

b) **Substitute reg.**:-Here user is allowed to use (examine/modify) registers directly. When you select this option the register contains with it's name display on 7 segment display. A,B,C,D,E,H,L,PSW,PCH,PCL,SPH,SPL is the sequence in which registers are displayed. You can examine or modify the register contents when that particular register is displayed.
Press S & Enter to enter inside substitute command press any other key except ENTER & Esc till display is-

reg ? – Press enter here for register use.
A=00 – It shows you contents of registers A. You can modify this register contents also. Press enter to see next register's contents. Press ESC to come to command mode.

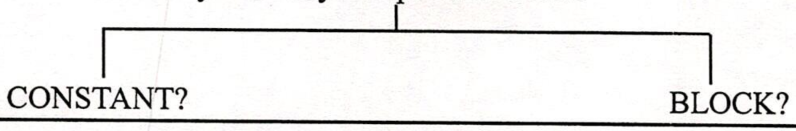
b) **Substitute IO?.**:-When you select this option you will get two options input port & output port. When monitor display output port it will asks for output address as well as data in Hex format. If you press Enter key here data will send on output port of 8255. Now when you select input port option it will asks for input address. If you press Enter key here input port no. with the corresponding data is displayed on monitor.

MOVE COMMAND. (MOV)

This command allows user to move a Memory Block data to another memory area or to move constant data to specified 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 MOV MOVE? Invoke this command.
Press enter to authorised this command

Press any other key except Enter & ESC



CONSTANT? BLOCK?

These options are explained in details as follows-

a) MOVE CONSTANT- Press MOV followed by Enter to invoke this command. This command allows you to move constant data byte to specified memory location.
   Press MOV & ENTER key to invoke this command-
   CONSTANT?- Press Enter to select data transfer.
   BYTE- It asks data which is to be transfer.
   55- Press Enter.
   START-Give 4 digit HEX address where data is to be transmitted & press Enter.
   END-Give 4 digit HEX address of memory location up to where you want to transfer & press Enter.
   DONE- It shows that your data is transfer to specified memory location. Press ESC to come to command mode.

b) MOVE BLOCK-User can transfer a memory block to another memory block.

Press MOV & Enter

CONSTANT?- Press any other key except ESC & ENTER.

BLOCK? – Press Enter to use move block type

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

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

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

DONE – 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.

COMPARE COMMAND (COMP)

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.

BLOCK?   CONSTANT?

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 done. 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 – It asks you start address of first memory block. Give 4 digit hex address & press Enter. e.g. 8000H
END – It asks you End address marking of first block. Give 4 digit hex address & press Enter. e.g. 80FFH.
DEST – It asks you the destination address of another memory block to b compared with first one. Give 4 digit hex address & press Enter. e.g. 8200H.
0000 00 – During comparison if mismatch is found that error address with da is displayed otherwise
DONE – The block is compared & data is matched. Pressing Esc to com command mode.

b) Compare constant – 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 done is displayed.

CONSTANT? – Press Enter to compare data
BYTE – 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 – It asks you the start address of the memory block to be compared. Give 4 digit hex address & press Enter. e.g. 8300H
END – It asks you end address of marked block. Give 4 digit hex address & press Enter. e.g. 83ffH
0000 00 – This is the mismatched address & data. Here if you press Enter to continue next comparison.
DONE. – At the end of successful comparison.

EXECUTE COMMAND (EXE)

Now we will see how to execute your programs using EXECUTE command. This command has three 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. In BREAK POINT mode specified part of program is executed for given no. of times. The three options are executed in round robin fashion.

Press EXE GO TO? Invoke EXECUTE command
Press Enter to authorise this command.
Press any key other than Esc & Enter.

FULL? SSTP? BREAK P?

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 EXE.
FULL ? – Press Enter to authorise the command.
ADDR – It asks you the address of user program from which the execution starts in full swing. Give here 4 digit hex address & press Enter.
WATE – Program execution is completed.
Press ESC to come to command mode.

b) **SINGLE STEP MODE** – This mode allows user to execute each instructi in their program, this displays each instruction address & first byte instruction. At each instruction user can verify & check the contents of exte memory & can see the effect on execution after each instruction. It is hi useful in program debugging. Since INT 0 is used for single stepping, I' is not available to user, but user can single step EPROM program also
SINGLE STEP? – Press Enter to go to single stepping.

ADDR – 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.

c) BREAK POINT MODE-

SERIAL LINK COMMAND

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

Commands	keyboard	Description.
SERIAL IN	S I/P	Receive on serial to kit.
SERIAL OUT	S O/P	Send on serial from kit.

These options are explained in details as follows-

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

SERIAL IN? – Press Enter to invoke this command.

INT HEX? –Press Enter if you want to transfer data in INT Hex format. Otherwise press any key other than ESC & ENTER.

BINARY?- Press Enter if you want to transfer data in binary format.

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

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

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

DONE. – 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-

SERIAL DATA OUT? – Press Enter key to invoke this command.

START ADDR 0000 - It asks you the Start address of memory location from where you want to transmit 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 send data from serial link & displays wait until end of file is received or end of file is exceeded. After sending of file the display is **O.**K.. – End of sending of file.

Useful Subroutines

1) **ACC DISPLAY:** 1041H = Display accumulator content at 'DSP_PTR' & 'DSP_PTR+1' location.
2) **BLANK DISPLAY:** 107EH = Used as a CR, LF clear display screen.
3) **WAIT DISPLAY:** 1092H = Display "WAIT" on 7-segment display.
4) **MESSAGE OUT:** 10A6H = Display message pointed by HL pair from start of display message upto the "End TXT" bit (i.e. 03H).
5) **OK DISPLAY:** 10B7H = Displays "OK" on 7-segment display.
6) **CHECK KEY:** 10CDH = Checks key & given key-code in accumulator. Executes in loop until key is pressed.
7) **DELAY:** 10DEH = For 100mS delay count in DE pair.
8) **NUMBER DISPLAY:** 10F1H = Displays number in DE pair & number of digit in B register. {Start position (right hand) in C register}.
9) **HEX TO ASCII:** 0F9FH = Nibble data in accumulator & ASCII in accumulator.
10) **ASCII TO HEX:** 02F6H =ASCII in Acc & Hex in lower nibble of Accumulator.
11) **BYTE DISPLAY:** 0FB0H = Single ASCII byte display at DSP_PTR location.

A) INT_STAT: 4000H =

* Key press = 1
* NO Key press = 0

B) DSP_PTR: 4001H = Display pointer location.
C) KEY_CODE: 4003H.

INTERRUPT ADDRESS:

H/W interrupts for 8085:

1) INT 5.5: 411EH
2) INT 6.5: 4124H
3) INT 7.5: Used for Hand shaking | Internally used.
4) TRAP: Used for SSTP. | Internally used.

S/W interrupts for 8085:

1) RST 0 = Reset. | Not for user application
2) RST 1 = Command mode with Display & reg. Save.
5) RST 2 = 4100H
3) RST 3 = Command mode without Display & reg. Save.
6) RST 4 = 4106H
7) RST 5 = 410CH
8) RST 6 = 4112H
9) RST 7 = 4118H

```
902B 8C           ADC H             ; Add with carry
902C 67           MOV H,A           ; Store rslt in 'H' reg.
902D 220793       SHLD 9307H        ; Store the 'HL' reg. pair
9030 CF           RST 1             ; End.
```

IN THIS PROGRAM THE TWO 16 BIT NOS.ARE FILLED AFTER THE PROGRAM,AT THE MEM.LOCATION 9303H AND 9305H. THE RESULT IS STORED AT 9307H.

3) TITLE:- "PROGAM FOR THE TRANSFER OF THE STRING OF SOME HEX NUMBERS FROM 930AH LOCATION TO 931AH LOCATION."

```
9050              ORG 9050H
9050 210A93       LXI H,930AH       ; Init.source ptr.
9053 111A93       LXI D,931AH       ; Init desti. ptr.
9056 0E05         MVI C,05H         ; conuter=05H.
9058 7E           DOWN:MOV A,M      ; Get first no.in accu.
9059 12           STAX D            ; Store first no.at desti
905A 23           INX H             ; Incr. source ptr.
905B 13           INX D             ; Incr. desti.ptr.
905C 0D           DCR C             ; Decr. counter.
905D C25890       JNZ DOWN          ; If not= 0,repeat.
9060 CF           RST 1             ; End.
```

IN THIS PROGRAM,THE STRING OF FIVE NOS.IS TO BE FILLED AT THE MEM.LOCATION [930AH],AFTER THE PROGRAM IS FILLED AND THE RESULT IS SEEN AT [931AH].

4) TITLE:- "PROGRAM TO FIND MINIMUM NUMBER FROM THE GIVEN ARRAY."

```
9070              ORG 9070H
9070 212A93       LXI H,932AH       ; Get the array.
9073 0E0A         MVI C,0AH         ; Set the counter.
9075 7E           MOV A,M           ; Take first no.
9076 23           LOOP:INX H        ; Move to the next no.
9077 BE           CMP M             ; Compare first two no.
```

```
9078 DA7C90   JC NEXT        ; If accu.<,jmp to nxt. no.
907B 7E       MOV A,M        ; Shft nxt no. in A.
907C 0D       NEXT:DCR C     ; decre. counter.
907D C27690   JNZ LOOP       ; If not zero,jump.
9080 322593   STA 9325H      ; Store result.
9083 CF       RST 1          ; End.
```

IN THIS PROGRAM THE ARRAY OF TEN NOS.IS TO BE STORED AT 932AH AND THE MINIMUM NO.IS STORED AT 9325H.

---

5) TITLE:- "PROGRAM TO FIND THE MAXIMUM NUMBER FROM THE GIVEN ARRAY."

```
9090          ORG 9090H
9090 213A93   LXI H,933AH    ; Get the array.
9093 0E0A     MVI C,0AH      ; Counter= 0AH.
9095 7E       MOV A,M        ; Get first no. in accu.
9096 23       TIME:INX H     ; Move to the nxt no..
9097 BE       CMP M          ; Compare two no.
9098 D29C90   JNC TYPE       ; If accu < no.jump.
909B 7E       MOV A,M        ; Take next no.
909C 0D       TYPE:DCR C     ; Decr. counter.
909D C29690   JNZ TIME       ; If counter not= 0, repeat.
90A0 324593   STA 9345H      ; Store result.
90A3 CF       RST 1          ; End.
```

IN THIS PROGRAM,THE ARRAY OF TEN NOS.IS TO BE STORED AT 933AH AND THE MAXIMUM NO. IS STORED AT 9345H.

---

6) TITLE:- "PROGRAM TO SUBSTRACT TWO 8 BIT NUMBERS."

```
90B0          ORG 90B0H
90B0 2A5093   LHLD 9350H     ; Load the numbers.
90B3 EB       XCHG           ; Reload the nos. in DE pair.
90B4 7A       MOV A,D        ; Shift  first no. in acc.
90B5 93       SUB E          ; Make substraction
90B6 325593   STA 9355H      ; Store result
```

```
90B9 CF           RST 1                     ; End.
```

IN THIS PROGRAM THE TWO 8 BIT NOS.ARE TO BE FILLED AT 9350H AND 9351H.RESULT OF SUBSTRACTION IS STORED AT 9355H.

7) TITLE:- "PROGRAM TO CONVERT THE BCD NUMBER, TO ITS EQUIVALANT HEX NUMBER AND RESULT IS STORED AT 9365H".

```
90C0              ORG 90C0H
90C0 3A6093       LDA 9360H                 ; Take the valid BCD no.
90C3 47           MOV B,A                   ; Store no. in B
90C4 FE10         CPI 10H                   ; Compare with 10H.
90C6 D2D090       JNC UP                    ; If no carry,jump to UP.
90C9 CAD090       JZ UP                     ; If zero,jump to UP.
90CC 78           MOV A,B                   ; Ld accu.
90CD C3DE90       JMP STOP                  ; Jump to STOP.
90D0 E6F0         UP:ANI 0F0H               ; AND with FOH
90D2 0F           RRC                       ; Shift to right.
90D3 0F           RRC                       ; Shift to right.
90D4 0F           RRC                       ; Shift to right.
90D5 0F           RRC                       ; Shift to right.
90D6 4F           MOV C,A                   ; Set counter.
90D7 78           MOV A,B                   ; Load accu.
90D8 DE06         SUB:SBI 06H               ; A= A-06H
90DA 0D           DCR C                     ; Decre.the counter.
90DB C2D890       JNZ SUB                   ; If not= zero,jump.
90DE 326593       STOP:STA 9365H            ; Store the accu.
90E1 CF           RST 1                     ; Stop.
```

IN THIS PROGRAM THE VALID BCD NO.IS STORED AT 9360H AND EQUIVALENT HEX NO. STORED AT 9365H LOCATION.

8) TITLE:- "PROGRAM TO CONVERT HEX NUMBER TO ITS EQUIVALANT BCD NUMBER.

```
90F0              ORG 90F0H
90F0 3A7A93       LDA 937AH                 ; Ld. accu.
```

```
9159 0F          RRC                          ; Rotate right.
915A DA4091      JC TO                        ; If cry.goto 'GO'.
915D CF          RST 1                        ; End execution.
```

IN THE ABOVE PROGRAM THE USER HAS TO ENTER THE DATA STRING IN RANDOM,AT 93A0H. AFTER THE EXECUTION OF THE PROGRAM,THE DATA STRING AT 93A0H WILL BE RECOVERED IN THE ASCENDING ORDER.THE ORDERED STRING IS STORED AT 93A0H.

11) TITLE:- "PROGRAM TO IMPLEMENTATION OF FIBONACI SERIES."

```
9170             ORG 9170H
9170 01B093      LXI B,93B0H                  ; Load desti. add.
9173 1600        MVI D,00H                    ; Make D zero.
9175 AF          XRA A                        ; Make acc.zero.
9176 02          STAX B                       ; Store acc.
9177 03          INX B                        ; Incr.BC reg. pair.
9178 2E01        MVI L,01H                    ; Load 01h in L.
917A 67          MOV H,A                      ; Shift acc.in H reg.
917B 85          DATE:ADD L                   ; Add L with acc.
917C 6C          MOV L,H                      ; Store H in L reg.
917D 67          MOV H,A                      ; Store acc. in H.
917E 02          STAX B                       ; Store acc.
917F 03          INX B                        ; Incr.BC reg. pair.
9180 15          DCR D                        ; Decr. counter.
9181 C27B91      JNZ DATE                     ; If not= zero,go back.
9184 CF          RST 1                        ; end execution.
```

IN THE ABOVE PROGRAM THE FIBONACI SERIES IS GENRATED.AFTER EXECUTION OF THE PROGRAM USER HAS TO CHECK THE SERIES AT DESTINATION ADDRESS. THE SERIESIS GIVEN FOR REFERANCE OF USER

```
                 ;AS FOLLOWS
                 ;FIBONACI SERIES{0,1,1,2,3,5,8,0D,15,22.......}
0000             END
```

```
10A6 =        MSG_OUT:         EQU 10A6H
107E =        BLNK_DISP:       EQU 107EH
10DE =        DELAY_10ms:      EQU 10DEH
0020 =        PA:              EQU 0020H
0021 =        PB:              EQU 0021H
0023 =        CWR:             EQU 0023H
0004 =        TLSB:            EQU 0004H
0005 =        TMSB:            EQU 0005H
0000 =        CW:              EQU 0000H
0043 =        CWR1:            EQU 0043H
0041 =        LOWBY:           EQU 0041H
10A6 =        EXT_MSG_OUT:     EQU 10A6H    ;LOAD MSG
                                              POINTER ONLY IN HL
1041 =        ACC_DSP:         EQU 1041H
4001 =        DSP_PTR:         EQU 4001H

0020 =        CS8255A:         EQU 20H      ;8255 PORT A
0021 =        CS8255B:         EQU 21H      ;8255 PORT B
0022 =        CS8255C:         EQU 22H      ;8255 PORT C
0023 =        CSW255:          EQU 23H      ;8255 ADDR
```

MONITOR - 0034H:JMP 4124H

1) TO GENERATE THE SQUARE WAVE OF 1KHZ ON THE OUT1 PIN OF 8253.

```
8100          ORG 8100H
8100 3E77     MVI A,77H         ;SET CONTROL WORD FOR 8253
8102 D343     OUT CWR1          ;OUT CW REGISTER.
8104 3E00    LOP1:MVI A,00H     ;LOAD LOWER BYTE.
8106 D341     OUT LOWBY         ;OUT TO CHANNEL ONE.
8108 3E10     MVI A,10H         ;LOAD HIGHER BYTE.
810A D341     OUT LOWBY         ;OUT TO CHANNEL ONE.
810C C30481   JMP LOP1          ;JMP TO LOOP.
```

2) PROGRAM TO GENERATE SQUARE WAVE AT THE OUTPUT OF 8255.

```
8200           ORG 8200H
8200 3E80      MVI A,80H          ;SET CONTROL WORD.
8202 D323      OUT CWR            ;STORE AT CWR.
8204 3E00      LOP3:MVI A,00H     ;LOAD 00H ACC.
8206 D320      OUT PA             ;OUT TO PORT A OF 8255.
8208 CDDE10    CALLDELAY_10ms     ;CALLDELAY SUBROUTINE.
820B 3EFF      MVI A,0FFH         ;LOAD FFH TO ACC.
820D D320      OUT PA             ;OUT TO PORT A OF 8255.
820F CDDE10    CALL DELAY_10ms    ;CALLDELAY SUBROUTINE.
8212 C30482    JMP LOP3           ;REPEAT.
```

3) PROGRAM TO VERIFY INTERRUPT (RST 6.5) USING 8085.

```
0003 =         ETX:   EQU 03H
4124           ORG 4124H          ;
4124 C31E83    JMP INT_65         ;
8300           ORG 8300H          ;
8300           START:
8300 31008F    LXI SP,8F00H       ;STACK TOP FOR PUSH/POP &
                                   CALL.
8303 FB        EI                 ;ENABLE INT.
8304 3E08      MVI A,08H          ;LOAD INT MASK PATTERN IN A.
8306 30        SIM                ;SET INTERRUPT MASK.
8307 00        NOP                ;
8308 00        NOP                ;
8309 3E0F      MVI A,0FH          ;CSW FOR 8155,ALL OUTPUT
                                   PORT.
830B D300      OUT CSW155         ;OUT TO 8155 CSW.
830D 3E80      MVIA,80H           ;LOAD CONTROL WORD FOR
                                   PORTA/B/C=OUT .
830F D323      OUT CSW255         ;OUT TO 8255 CSW.
8311 00        NOP                ;
8312 3E55      MVIA,55H           ;LOAD 55H IN ACC.
```

```
8314 D301      OUT CS8155A     ;OUT TO 8155 CONTROL
                               WORD.
8316 3E55      MVI A,55H       ;LOAD 55H IN ACC.
8318 D320      OUT CS8255A     ;OUT TO 8255 CONTROL WORD.
831A 00        LOP4:   NOP     ;
831B C31A83 JMP LOP4           ;JUMP BACK AND REPEAT.

831E           INT_65:
831E 3EAA      MVI A,0AAH      ;LOAD AAH TO ACC.
8320 D301      OUT CS8155A     ;OUT TO 8155 CONTROL WORD.
8322 3EAA      MVI A,0AAH      ;LOAD AAH TO ACC.
8324 D320      OUT CS8255A     ;OUT TO 8255 CONTROL
                               WORD.
8326 C9        RET             ;RETURN.
```

CONNECTOR DETAILS

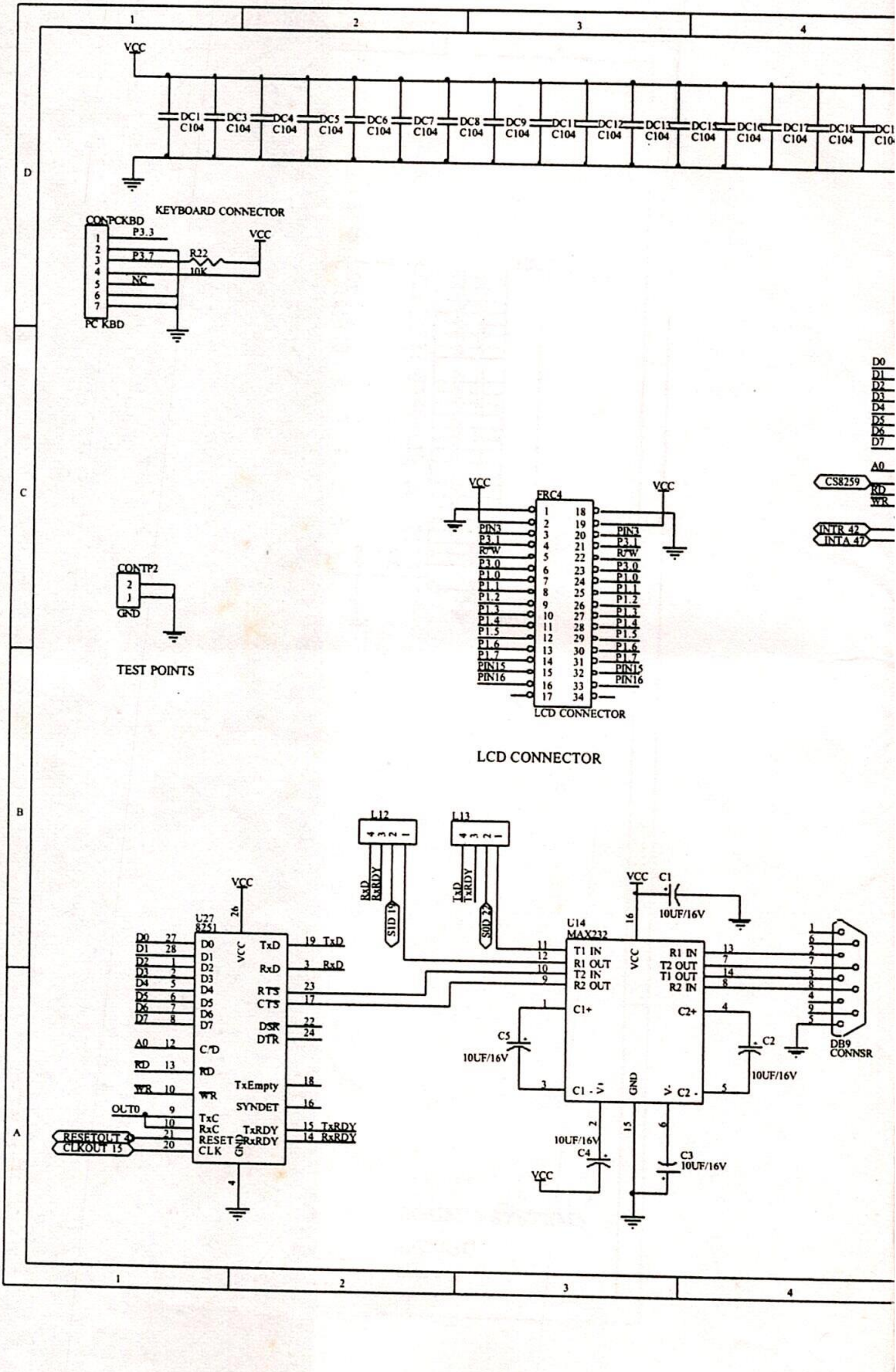
50 PIN STD BUS
FRC1

8255 Connector
FRC2

in 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	RST 7.5	12	TRAP	11	PB2	12	PB3
13	Hold	14	RST 6.5	13	PB0	14	PB1
15	CLK Out	16	HLDA	15	PA6	16	PA7
17	RDY	18	RST 5.5	17	PA4	18	PA5
19	SID	20	I/OM	19	PA2	20	PA3
21	RD	22	SOD	21	PA0	22	PA1
23	ALE	24	WR	23	PC6	24	PC7
25	Addr Dir	26	BA00	25	GND	26	Vcc.
27	BA15	28	BA01				
29	BA14	30	BA02				
31	BA13	32	BA03				
33	BA12	34	BA04				
35	BA11	36	BA05				
37	BA10	38	BA07				
39	BA09	40	BA06				
41	BA08	42	INTR				
43	RST Out	44	SP I/o CS				
45	Data Dir	46	NC				
47	INTA	48	NC				
49	VCC	50	GND				

8253 Connector REL2

Pin No.	Pin Details	Pin No.	Pin Details
01	CLK 2	07	CLK 0
02	OUT 2	08	OUT 0
03	GATE 2	09	GATE0
04	CLK 1	10	GND
05	GATE 1		
06	OUT 1		


1
2
3
4
D
C
B
A
VCC
DC1 C104
DC3 C104
DC4 C104
DC5 C104
DC6 C104
DC7 C104
DC8 C104
DC9 C104
DC1 C104
DC12 C104
DC13 C104
DC15 C104
DC16 C104
DC17 C104
DC18 C104
DC1 C10
KEYBOARD CONNECTOR
CONPCKBD
P3.3
P3.7
R22
10K
VCC
NC
PC KBD
1
2
3
4
5
6
7
CONTP2
2
1
GND
TEST POINTS
D0
D1
D2
D3
D4
D5
D6
D7
A0
CS8259
RD
WR
INTR 42
INTA 47
VCC
FRC4
VCC
PIN3
P3.1
R/W
P3.0
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
PIN15
PIN16
LCD CONNECTOR
LCD CONNECTOR
L12
L13
RxD
RxRDY
TxD
TxRDY
SID 1
SOD 2
VCC
U27
8251
26
D0 27 D0
D1 28 D1
D2 1 D2
D3 2 D3
D4 5 D4
D5 6 D5
D6 7 D6
D7 8 D7
A0 12 C/D
RD 13 RD
WR 10 WR
OUT0 9 TxC
10 RxC
RESETOUT 4 21 RESET
CLKOUT 15 20 CLK
TxD 19 TxD
RxD 3 RxD
RTS 23
CTS 17
DSR 22
DTR 24
TxEmpty 18
SYNDET 16
TxRDY 15 TxRDY
RxRDY 14 RxRDY
GND
4
VCC C1
10UF/16V
16
U14
MAX232
11 T1 IN
12 R1 OUT
10 T2 IN
9 R2 OUT
1 C1+
3 C1 -
R1 IN 13
T2 OUT 7
T1 OUT 14
R2 IN 8
C2+ 4
C2 - 5
V+
V-
GND
2
15
6
C5
10UF/16V
C2
10UF/16V
10UF/16V
C4
C3
10UF/16V
VCC
DB9
CONNSR
1
6
2
7
3
8
4
9
5