I

Name ____________________

Roll No. ____________ Year 20 ______ 20 ______

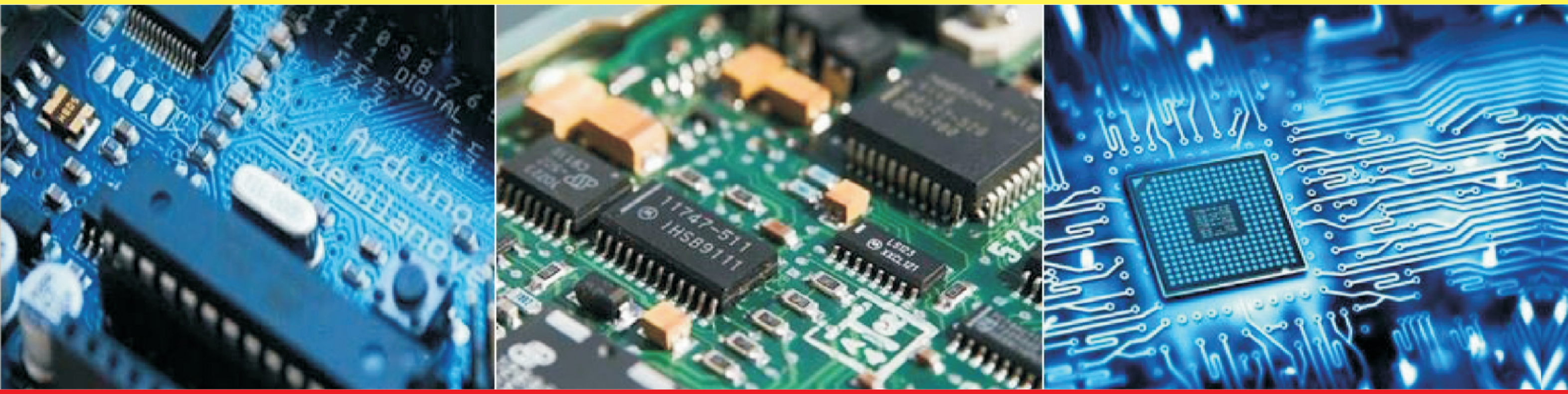
Exam Seat No. ____________________

ELECTRONICS GROUP | SEMESTER - IV | DIPLOMA IN ENGINEERING AND TECHNOLOGY

A LABORATORY MANUAL

FOR

MICROCONTROLLER AND APPLICATIONS (22426)



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION, MUMBAI

(Autonomous) (ISO 9001 : 2015) (ISO / IEC 27001 : 2013)

VISION

To ensure that the Diploma level Technical Education constantly matches the latest requirements of technology and industry and includes the all-round personal development of students including social concerns and to become globally competitive, technology led organization.

MISSION

To provide high quality technical and managerial manpower, information and consultancy services to the industry and community to enable the industry and community to face the changing technological and environmental challenges.

QUALITY POLICY

We, at MSBTE are committed to offer the best in class academic services to the students and institutes to enhance the delight of industry and society. This will be achieved through continual improvement in management practices adopted in the process of curriculum design, development, implementation, evaluation and monitoring system along with adequate faculty development programmes.

CORE VALUES

MSBTE believes in the followings:

- Education industry produces live products.
- Market requirements do not wait for curriculum changes.
- Question paper is the reflector of academic standards of educational organization.
- Well designed curriculum needs effective implementation too.
- Competency based curriculum is the backbone of need based program.
- Technical skills do need support of life skills.
- Best teachers are the national assets.
- Effective teaching learning process is impossible without learning resources.

A Laboratory Manual
for

Microcontroller and Applications

(22426)

Semester-IV

(EJ, DE, ET, EN, EX, EQ, IE, IS, IC)

Maharashtra State
Board of Technical Education, Mumbai
(Autonomous) (ISO 9001:2015) (ISO/IEC 27001:2013)

Maharashtra State Board of Technical Education,
(Autonomous) (ISO 9001 : 2015) (ISO/IEC 27001 : 2013)
4th Floor, Government Polytechnic Building, 49, Kherwadi,
Bandra (East), Mumbai - 400051.
(Printed on November 2018)

Maharashtra State
Board of Technical Education
Certificate

This is to certify that Mr. / Ms ……………………………………….
Roll No……………………….of ……….. Semester of Diploma
in ……………………………………………………. of Institute
……………………………………….   (Code………………..)
has attained pre-defined practical outcomes(PROs) satisfactorily in course **Microcontroller and Applications (22426)** for the academic year 20…….to 20…..... as prescribed in the curriculum.

Place ……………….   Enrollment No……………………
Date:…....................   Exam Seat No. …………………......

Course Teacher   **Head of the Department**   **Principal**

Seal of the
Institute

Preface

The primary focus of any engineering laboratory/ field work in the technical education system is to develop the much needed industry relevant competencies and skills. With this in view, MSBTE embarked on this innovative 'I' Scheme curricula for engineering diploma programmes with outcome-base education as the focus and accordingly, relatively large amount of time is allotted for the practical work. This displays the great importance of laboratory work making each teacher; instructor and student to realize that every minute of the laboratory time need to be effectively utilized to develop these outcomes, rather than doing other mundane activities. Therefore, for the successful implementation of this outcome-based curriculum, every practical has been designed to serve as a ***'vehicle'*** to develop this industry identified competency in every student. The practical skills are difficult to develop through 'chalk and duster' activity in the classroom situation. Accordingly, the 'I' scheme laboratory manual development team designed the practicals to ***focus*** on the ***outcomes***, rather than the traditional age old practice of conducting practicals to 'verify the theory' (which may become a byproduct along the way).

This laboratory manual is designed to help all stakeholders, especially the students, teachers and instructors to develop in the student the pre-determined outcomes. It is expected from each student that at least a day in advance, they have to thoroughly read through the concerned practical procedure that they will do the next day and understand the minimum theoretical background associated with the practical. Every practical in this manual begins by identifying the competency, industry relevant skills, course outcomes and practical outcomes which serve as a key focal point for doing the practical. The students will then become aware about the skills they will achieve through procedure shown there and necessary precautions to be taken, which will help them to apply in solving real-world problems in their professional life.

This manual also provides guidelines to teachers and instructors to effectively facilitate student-centered lab activities through each practical exercise by arranging and managing necessary resources in order that the students follow the procedures and precautions systematically ensuring the achievement of outcomes in the students.

Microcontroller is used in almost all the domestic, industrial, consumer goods and other high end products. Automation is used in every field of engineering and microcontroller is inbuilt element of these systems and devices. Diploma engineers have to deal with various microcontroller based systems and maintain them. This course is intended to develop the skills to maintain and solve the application problems related to microcontrollers.

Although all care has been taken to check for mistakes in this laboratory manual, yet it is impossible to claim perfection especially as this is the first edition. Any such errors and suggestions for improvement can be brought to our notice and are highly welcome.

Programme Outcomes (POs) to be achieved through Practical of this Course

Following programme outcomes are expected to be achieved through the practical of the course

PO1. **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Electronics and Telecommunication engineering problems.

PO2. **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.

PO3. **Experiments and practice:** Experiments and practice: Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems.

PO4. **Engineering tools:** Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations

PO5. **The engineer and society:** Assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to practice in field of Electronics and Telecommunication engineering.

PO6. **Environment and sustainability:** Apply Electronics and Telecommunication engineering solutions also for sustainable development practices in societal and environmental contexts.

PO7. **Ethics:** Apply ethical principles for commitment to professional ethics, responsibilities and norms of the practice also in the field of Electronics and Telecommunication engineering.

PO8. **Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.

PO9. **Communication:** Communicate effectively in oral and written form.

PO10. **Life-long learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics and Telecommunication engineering and allied industry.

Program Specific Outcomes (PSO):-.

PSO1. Electronics and Telecommunication Systems: Maintain various types of Electronics and Telecommunication systems.

PSO2. EDA Tools Usage: Use EDA tools to develop simple Electronics and Telecommunication engineering related circuits.

List of Industry Relevant Skills

- The following industry relevant skills of the competency '**Maintain microcontroller based systems**' are expected to be developed in students by undertaking the practicals of this laboratory manual.

1. Identify the relevant microcontroller.
2. Interface various I/O devices with microcontroller.
3. Interpret the program.
4. Maintain microcontroller based systems.
5. Use features available with given microcontroller.
6. Test the circuit for the given application.
7. Find faults and trouble shoot the given system.

Practical- Course Outcome matrix

Course Outcomes (COs)

a. Analyze architecture of microcontroller ICs.
b. Interpret the program for 8051in assembly language for the given operations.
c. Interpret the program by using timer, interrupt and serial ports /parallel ports.
d. Interface the memory and I/O devices to 8051 microcontroller.
e. Maintain microcontroller used in different application

Pro. No.	Practical Outcomes (PrO)	CO a.	CO b.	CO c.	CO d.	CO e.
1.	Identify various blocks of 8051 microcontroller development board.	√	-	-	-	-
2.	Write sample assembly language program using various addressing modes and assembler directives.	-	√	-	-	-
3.	Write an assembly language program (ALP) to perform arithmetic operations : addition, subtraction, multiplication and division.	-	√	-	-	-
4.	Write an ALP to transfer data from source to destination location of internal/external data memory.	-	√	-	-	-
5.	Write an ALP to find smallest/largest number from the given data bytes stored in internal/external data memory locations.	-	√	-	-	-
6.	Write an ALP for arranging numbers in ascending /descending order stored in external memory locations.	-	√	-	-	-
7.	Write an ALP to generate delay using register.	-	-	√	-	-
8.	Write an ALP to transfer 8 bit data serially on serial port.	-	-	√	-	-
9.	Interface LED with microcontroller and turn it ON with microcontroller interrupt.	-	-	√	-	-
10.	Develop an ALP to generate pulse and square wave by using Timer delay.	-	-	√	-	-
11.	Interface 4 X 4 LED matrix with 8051 to display various pattern.	-	-	-	√	-
12.	Interface 7-segment display to display the decimal number from 0 to 9.	-	-	-	√	-
13.	Interface relay with microcontroller and turn it ON and OFF.	-	-	-	√	-
14.	Interface LCD with 8051 microcontroller to display the characters and decimal numbers.	-	-	-	√	-
15.	Interface the given keyboard with 8051 and display the key pressed.	-	-	--	√	-
16.	Interface ADC with 8051 microcontroller and verify input/output.	-	-	-	-	√

Pro. No.	Practical Outcomes (PrO)	CO a.	CO b.	CO c.	CO d.	CO e.
17.	Interface DAC with 8051 microcontroller and observe following waveforms: square wave, triangular wave, saw-tooth wave .	-	-	-	-	√
18.	Interface stepper motor to microcontroller and rotate in clockwise and anti-clockwise direction at the given angles.	-	-	-	-	√

Guidelines to Teachers

1. Teacher is expected to refer complete curriculum document and follow guidelines for implementation
2. Teacher should provide the guideline with demonstration of practical to the students with all features.
3. Teacher shall explain prior concepts to the students before starting of each practical
4. Involve students in performance of each practical.
5. Teacher should ensure that the respective skills and competencies are developed in the students after the completion of the practical exercise.
6. Teachers should give opportunity to students for hands on experience after the demonstration.
7. Teacher is expected to share the skills and competencies to be developed in the students.
8. Teacher may provide additional knowledge and skills to the students even though not covered in the manual but are expected the students by the industry.
9. Give practical assignment and assess the performance of students based on task assigned to check whether it is as per the instructions.
10. Assess the skill achievement of the students and COs of each unit.
11. At the beginning Teacher should make the students acquainted with any of the simulation software environment as few experiments are based on simulation.
12. It is desirable to paste the photo of actual experimental setup or draw block diagram of experimental setup.
13. Practical No.1 should not be consider for Practical (ESE-End Semester Exam).

Instructions for Students

1. Listen carefully the lecture given by teacher about course, curriculum, learning structure, skills to be developed.
2. Before performing the practical student shall read lab manual of related practical to be conducted.
3. For incidental writing on the day of each practical session every student should maintain a ***dated log book*** for the whole semester, apart from this laboratory manual which s/he has to ***submit for assessment to the teacher***.
4. Organize the work in the group and make record of all observations.
5. Students shall develop maintenance skill as expected by industries.
6. Student shall attempt to develop related hand-on skills and gain confidence.
7. Student shall develop the habits of evolving more ideas, innovations, skills etc. those included in scope of manual
8. Student shall refer technical magazines, IS codes and data books.
9. Student should develop habit to submit the practical on date and time.
10. Student should well prepare while submitting write-up of exercise.

Content Page

List of Practicals and Progressive Assessment Sheet

Sr No	Title of the practical	Page No.	Date of performance	Date of submission	Assessmen marks(25)	Dated sign. of teacher	Remarks (if any)
1*	Identify various blocks of 8051 microcontroller development board.						
2	Write sample assembly language program using various addressing modes and assembler directives.						
3	Write an assembly language program (ALP) to perform arithmetic operations : addition, subtraction, multiplication and division.						
4*	Write an ALP to transfer data from source to destination location of internal/external data memory.						
5	Write an ALP to find smallest/largest number from the given data bytes stored in internal/external data memory locations.						
6	Write an ALP for arranging numbers in ascending /descending order stored in external memory locations.						
7*	Write an ALP to generate delay using register.						
8	Write an ALP to transfer 8 bit data serially on serial port.						
9	Interface LED with microcontroller and turn it ON with microcontroller interrupt.						
10*	Develop an ALP to generate pulse and square wave by using Timer delay.						
11*	Interface 4 X 4 LED matrix with 8051 to display various pattern.						
12	Interface 7-segment display to display the decimal number from 0 to 9.						
13*	Interface relay with microcontroller and turn it ON and OFF.						

14*	Interface LCD with 8051 microcontroller to display the alphabets and decimal numbers.						
15	Interface the given keyboard with 8051and display the key pressed.						
16*	Interface ADC with 8051 microcontroller and verify input/output.						
17*	Interface DAC with 8051 microcontroller and observe following waveforms: square wave, triangular wave, saw-tooth wave.						
18*	Interface stepper motor to microcontroller and rotate in clockwise and anti-clockwise direction at the given angles.						
Total Marks							

- **The practical marked as '*' are compulsory,**
- **Column 6th marks to be transferred to Performa of CIAAN-2017**.

Practical No.1: Identify various blocks of 8051 microcontroller development board.

I Practical Significance

Microcontroller has wide application in electronic system needing real time processing/control, starting from domestic application such as washing machine, TV and air conditioners. Microcontrollers are also used in automobiles, process control industries, cell phones, robotics and in space application. This practical will help the students to develop skills to Identify various blocks of 8051 development board.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Engineering tools:** Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations.
- **Individual and team work:** Function effectively as leader and team member in diverse/multidisciplinary teams.
- **Lifelong learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics and Telecommunication engineering and allied industry.

III Competency and Practical skills

This practical is expected to develop the following skills for the industry identified competency '**Maintain microcontroller-based systems':**

- Identify the components on microcontroller development board and understand their functions.
- Use integrate development environment (IDE).

IV Relevant Course Outcome(s)

- Analyze architecture of microcontroller ICs

V Practical Outcome

- Identify various blocks of 8051 microcontroller development board.

VI Relevant Affective domain related Outcome(s)

- Demonstrate working as a leader/a team member.
- Maintain tools and equipment.
- Follow ethical practices.

VII Minimum Theoretical Background

Microcontroller is a single chip microcomputer made through VLSI fabrication. 8051 is the microcontroller of the MCS-51 family introduced by Intel Corporation. It has inbuilt components such as CPU, internal RAM and ROM, timers/counters, serial ports, interrupts and I/O ports.

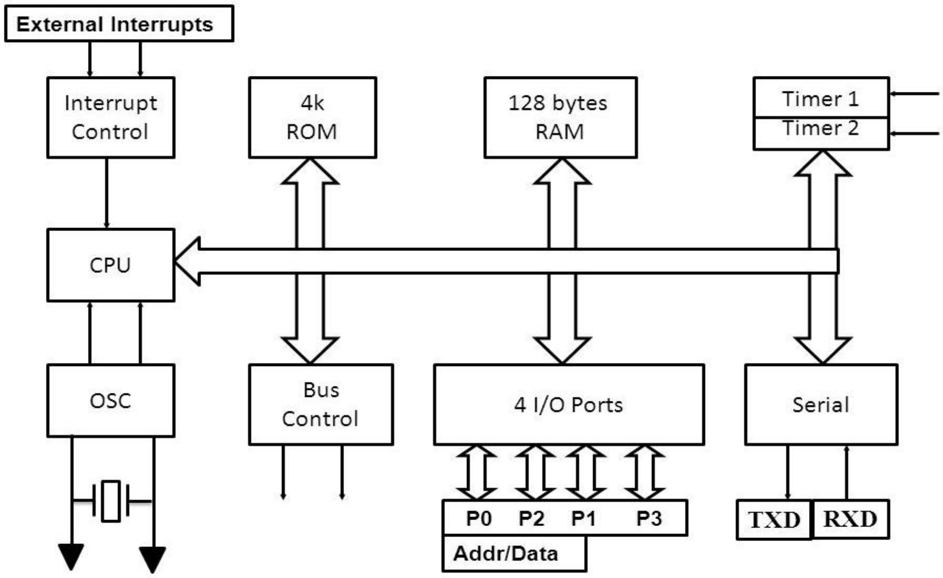


Fig 1.1 Block diagram of 8051

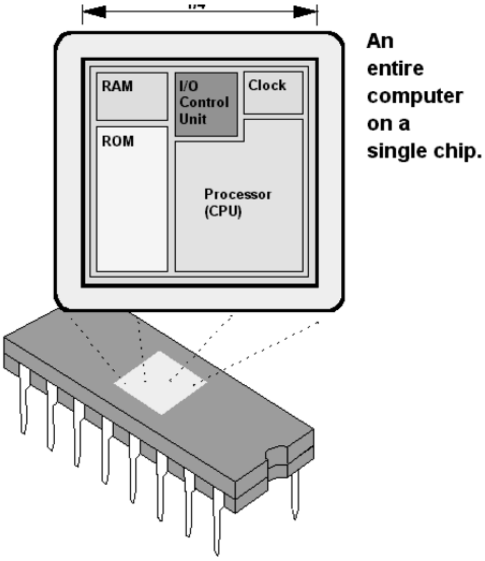


Fig 1.2 System On Chip - 8051

Specification of microcontroller 8051

- It is an 8-bit microcontroller.
- It has 128 bytes of Internal RAM.
- It has 4kilo bytes of Internal ROM.
- It has two 16 bit internal timers/counters.
- It has 4 eight-bit parallel ports.
- Programmable full duplex serial channel.
- It has three internal and two external interrupts.

Pin diagram of 8051: 8051 is a 40 pin IC and operates on +5 volts DC supply.

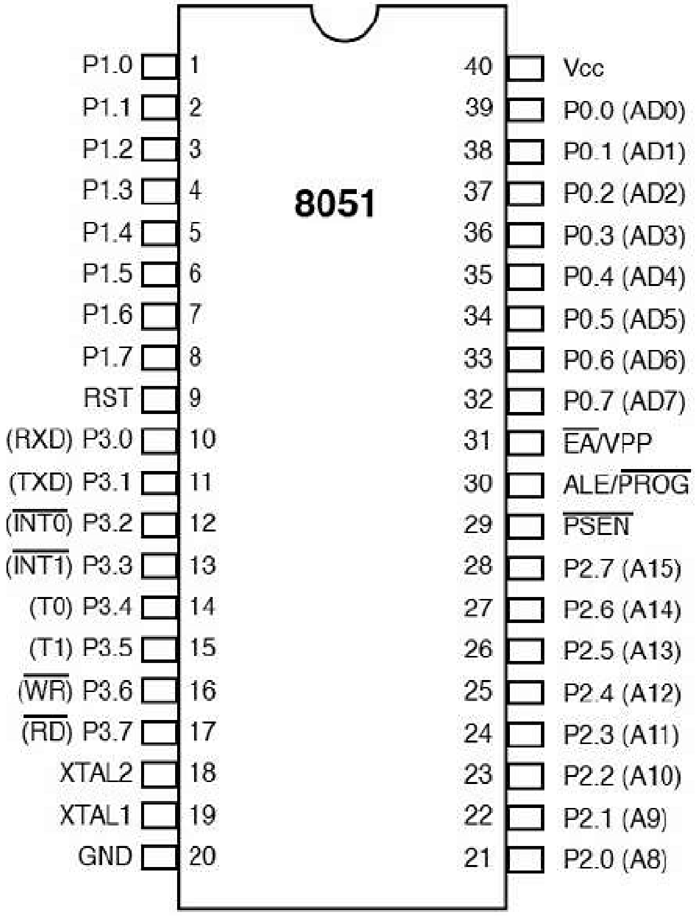


Fig. 1.3 Pin diagram of 8051

Keil IDE:

Keil is 8051 development tool which includes a text Editor, Assembler Debugger, linker, Simulator, C-complier, hex converter, locator and some in-built features like logic analyzer to observe various waveforms. It also includes terminal emulator. Keil supports all 8051 derivatives and valuable tool for embedded software development.

The Development board:

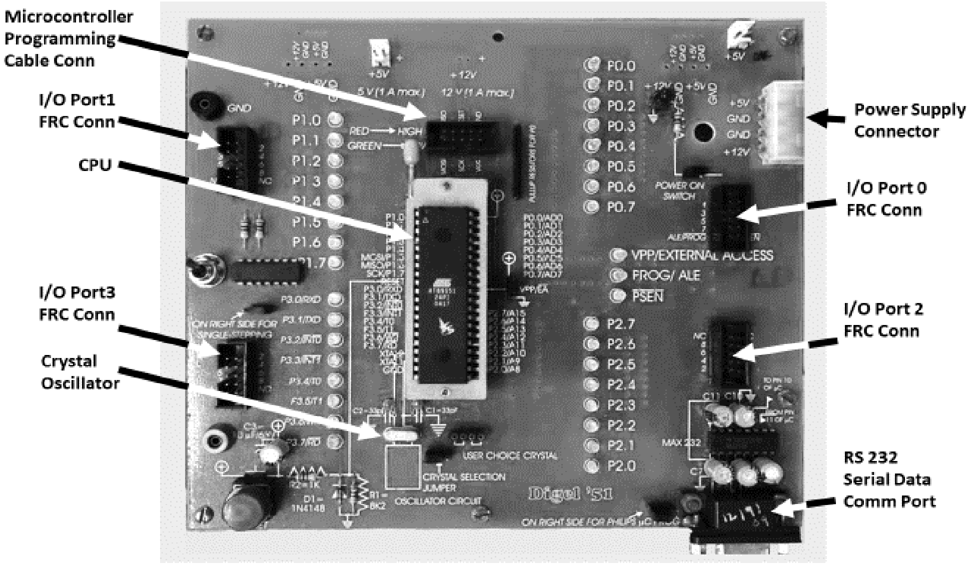


Fig. 1.4 8051 Development Board

The development board has the 89S51 microcontroller along with some necessary component like MAX 232, resistor network etc. It is a devise used to develop and design a prototype embedded system. Port pins are taken out for interfacing various peripherals. It has a provision to download the hex file of user program which is generated by Keil or any other IDE. FLASH MAGIC software is used to download the hex file into the code memory of microcontroller.

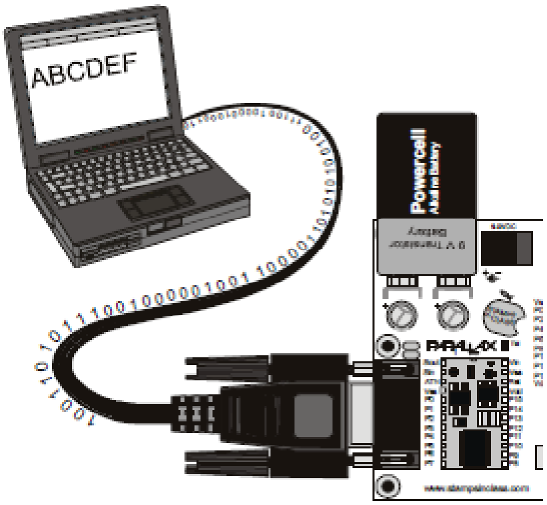


Fig. 1.5 8051 Programming through serial cable

VIII Resources required

Sr. No.	Instrument /Components	Specification	Quantity
1.	Microcontroller kit	Single board system with 8K RAM,ROM memory with battery backup,16X4,16X2LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler,RS-232,USB, interfacing facility with built in power supply.	1 No.
2.	Desktop PC	Loaded with open source IDE, simulation and program downloading software.	1 No.

IX Precautions to be Followed

Do not power up development board when identifying blocks.

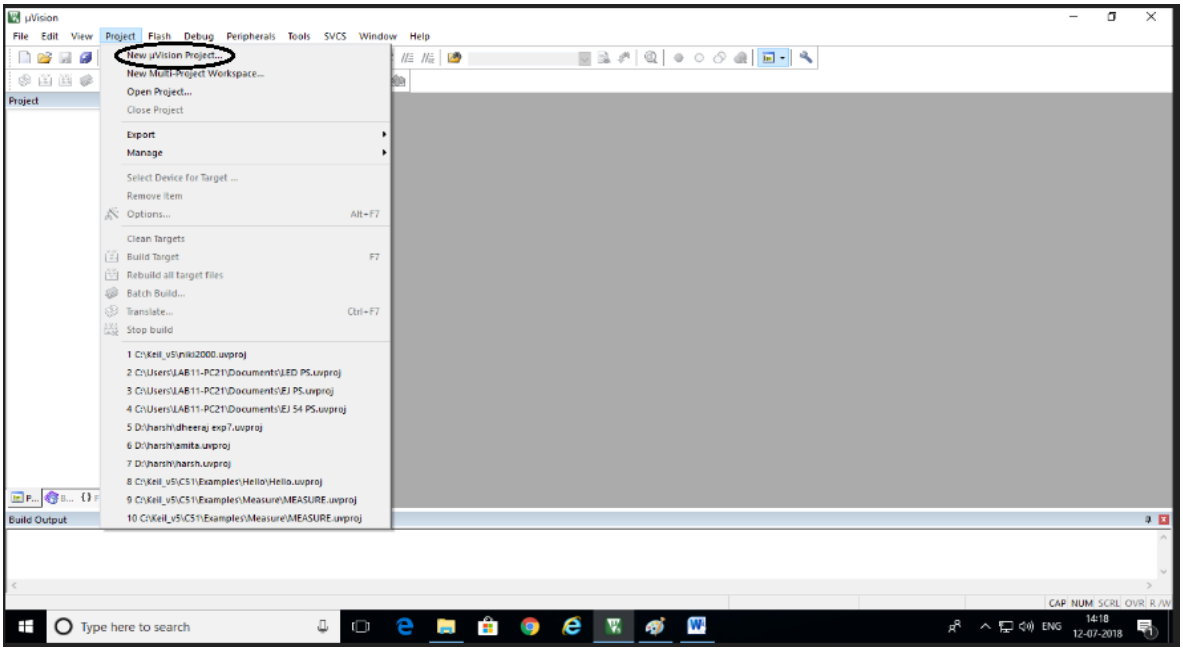
X Procedure

Steps for creating a project using Keil software:

1. Start Keil by double clicking on Keil icon. (Keil automatically opens the last project which was opened previously, when Keil was closed).

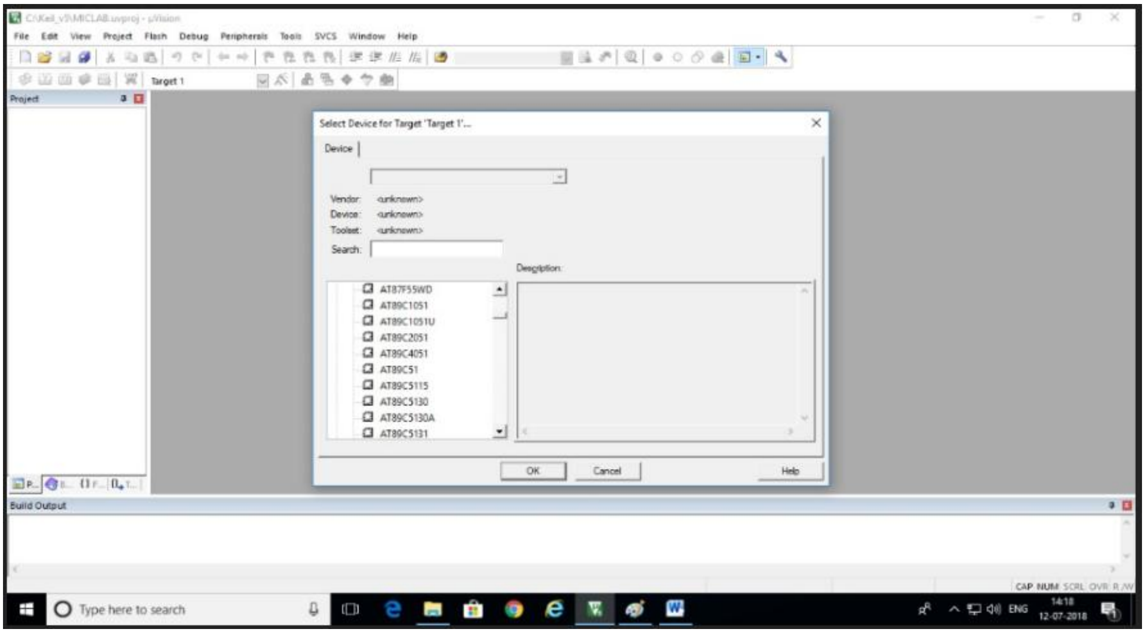
To create new project, Click on Project and select new project.

Select appropriate location for new project and type project name, click on save button.


µVision
File
Edit
View
Project
Flash
Debug
Peripherals
Tools
SVCS
Window
Help
New µVision Project...
New Multi-Project Workspace...
Open Project...
Close Project
Export
Manage
Select Device for Target ...
Remove Item
Options... Alt+F7
Clean Targets
Build Target F7
Rebuild all target files
Batch Build...
Translate... Ctrl+F7
Stop build
1 C:\Keil_v5\niki2000.uvproj
2 C:\Users\LAB11-PC21\Documents\LED PS.uvproj
3 C:\Users\LAB11-PC21\Documents\EJ PS.uvproj
4 C:\Users\LAB11-PC21\Documents\EJ 54 PS.uvproj
5 D:\harsh\dheeraj exp7.uvproj
6 D:\harsh\amita.uvproj
7 D:\harsh\harsh.uvproj
8 C:\Keil_v5\C51\Examples\Hello\Hello.uvproj
9 C:\Keil_v5\C51\Examples\Measure\MEASURE.uvproj
10 C:\Keil_v5\C51\Examples\Measure\MEASURE.uvproj
Project
Build Output
Type here to search
CAP NUM SCRL OVR R/W
ENG
14:18
12-07-2018

"Select device for Target Target-1" window will open. It displays a list of manufacturers of microcontrollers.

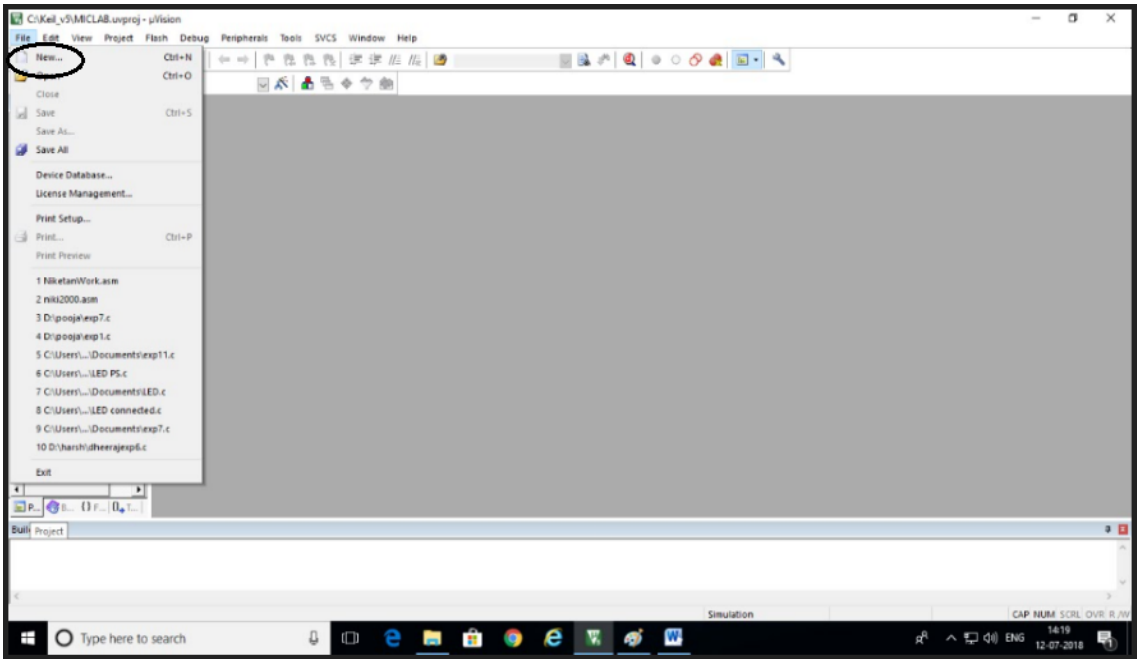
Double click on ATMEL or INTEL, list of supported microcontrollers gets displayed. Select 80C51AH from INTEL or AT89C51 (or as per the target board) for ATMEL then click ok.



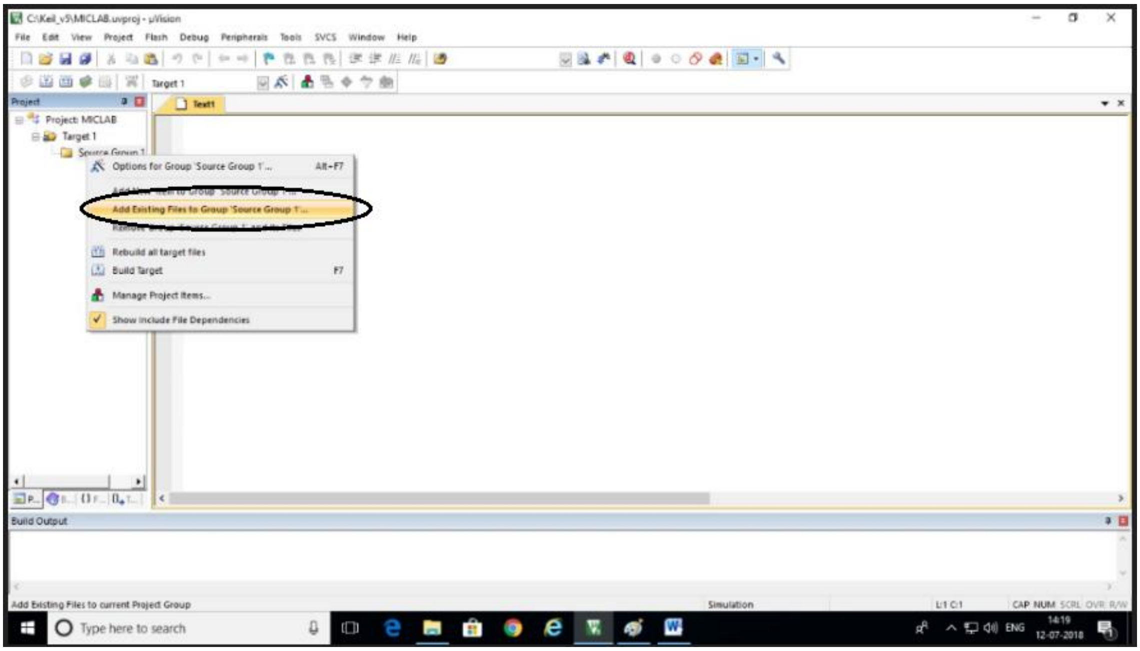
Select Device for Target 'Target 1'...

Vendor: <unknown>
Device: <unknown>
Toolset: <unknown>
Search:
Description:

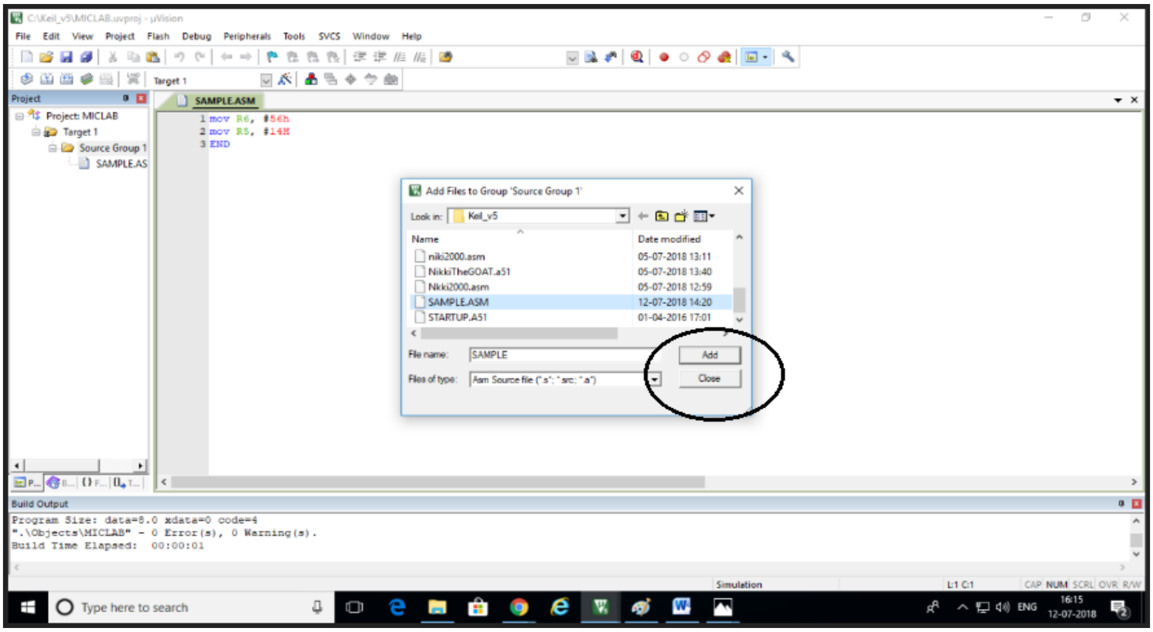
Click file pull down menu. Select new, a text editor window will open. Save this file in a same folder where project was stored. Give extension as .ASM or. A51.



On left hand project work space window will display Target1 and Source group1.
Right click on source group; Add files to source group 1.


C:\Keil_v5\MICLAB.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Text1
Project: MICLAB
Target 1
Source Group 1
Options for Group 'Source Group 1'... Alt+F7
Add New Item to Group 'Source Group 1'...
Add Existing Files to Group 'Source Group 1'...
Remove Group 'Source Group 1' and its Files
Rebuild all target files
Build Target F7
Manage Project Items...
Show Include File Dependencies
Build Output
Add Existing Files to current Project Group
Simulation
L:1 C:1
CAP NUM SCRL OVR R/W
Type here to search
ENG
14:19
12-07-2018

Select file type as asm source file. Now all .asm file Name will be displayed. Select appropriate file, click ADD and close.



```
mov R6, #56h
mov R5, #14H
END
```

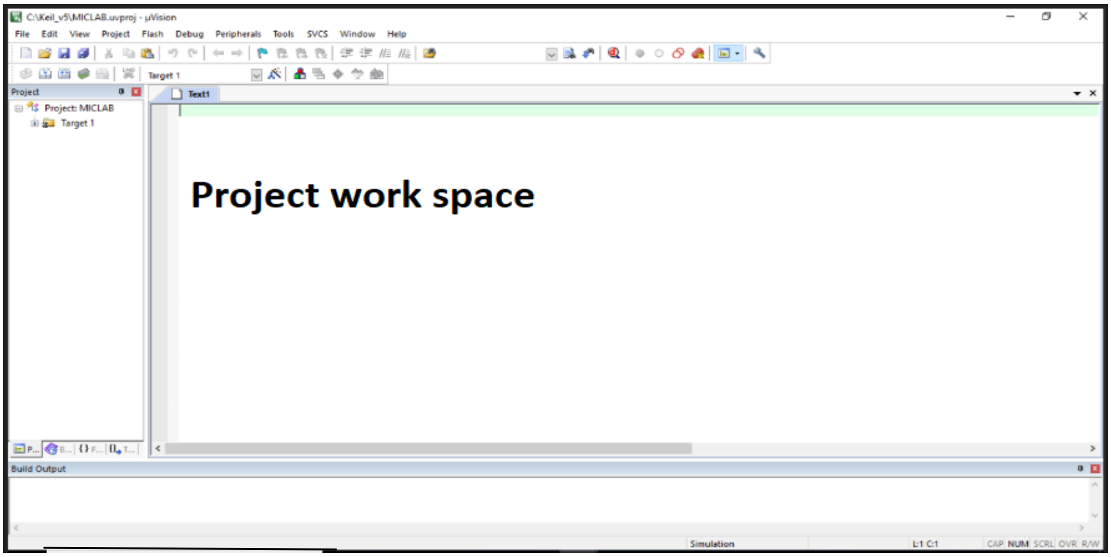
Name	Date modified
niki2000.asm	05-07-2018 13:11
NikkiTheGOAT.a51	05-07-2018 13:40
Nkki2000.asm	05-07-2018 12:59
SAMPLE.ASM	12-07-2018 14:20
STARTUP.A51	01-04-2016 17:01

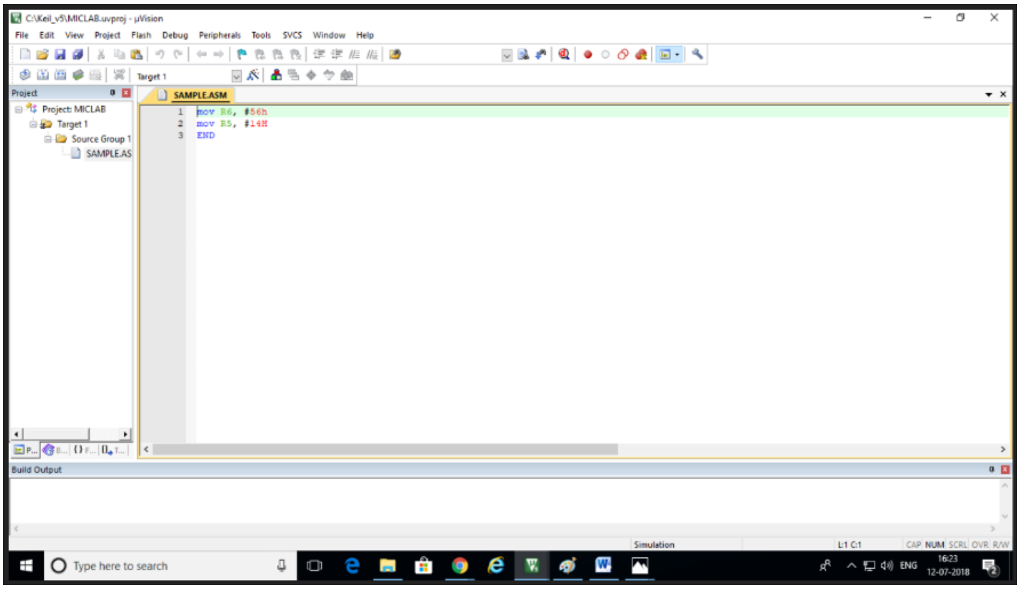
File name: SAMPLE

Files of type: Asm Source file (*.s*; *.src; *.a*)

```
Program Size: data=8.0 xdata=0 code=4
".\Objects\MICLAB" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:01
```

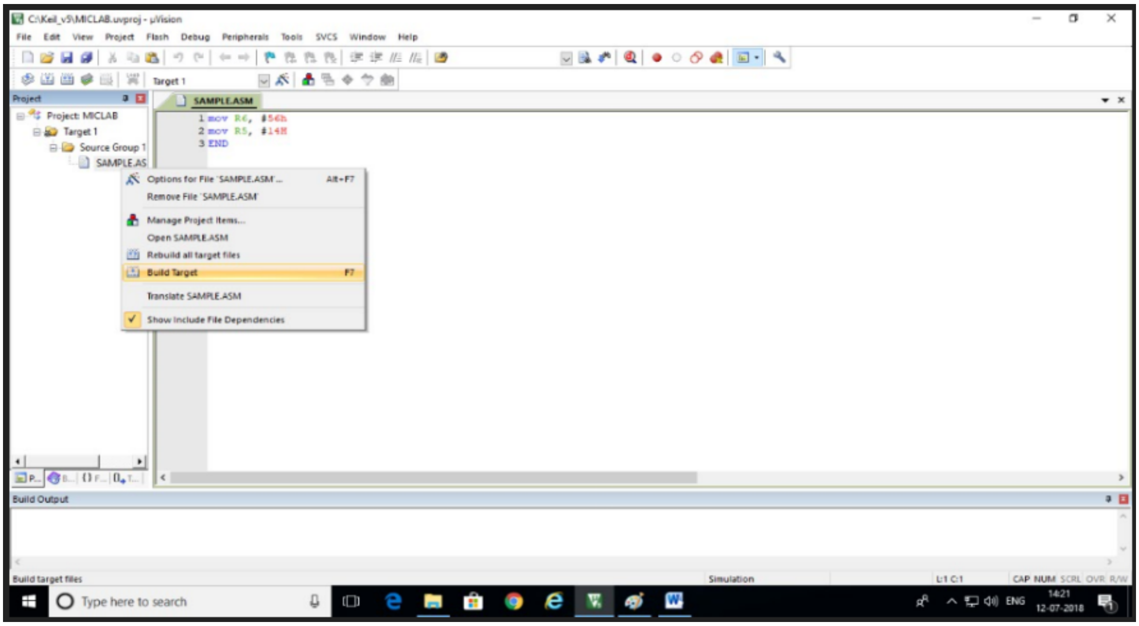
Project work space window will display ‘Target 1’ and ‘Source group 1’ with added file name.





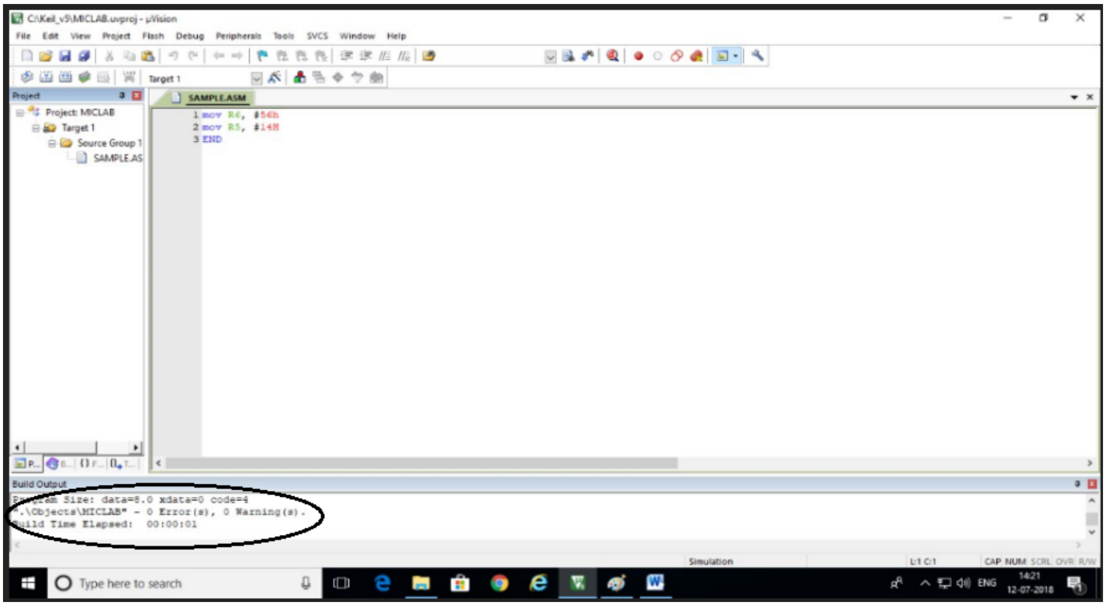
```
mov R6, #56h
mov R5, #14H
END
```

Type assembly language program. End with END directive. Save the file periodically. Right click on source group, click on Build target or press F7.



```
mov R6, #56h
mov R5, #14H
END
```

Output window will display the errors if any. If there are some errors, then remove the errors and repeat from step number 12 until no errors.



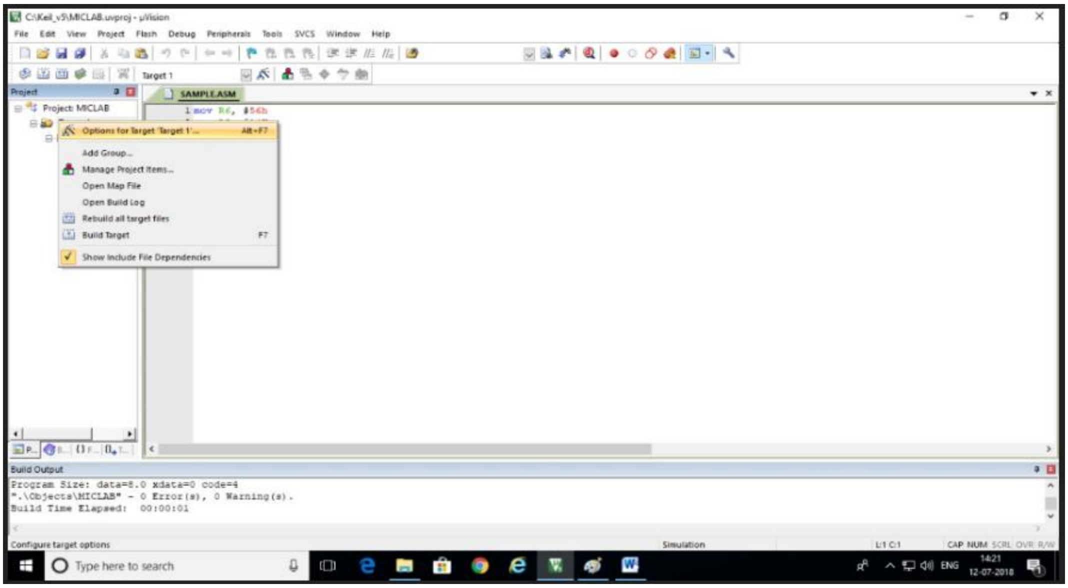
```
mov R6, #56h
mov R5, #14H
END
```

```
Program Size: data=8.0 xdata=0 code=4
".\Objects\MICLAB" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:01
```

To create a hex file, follow this procedure

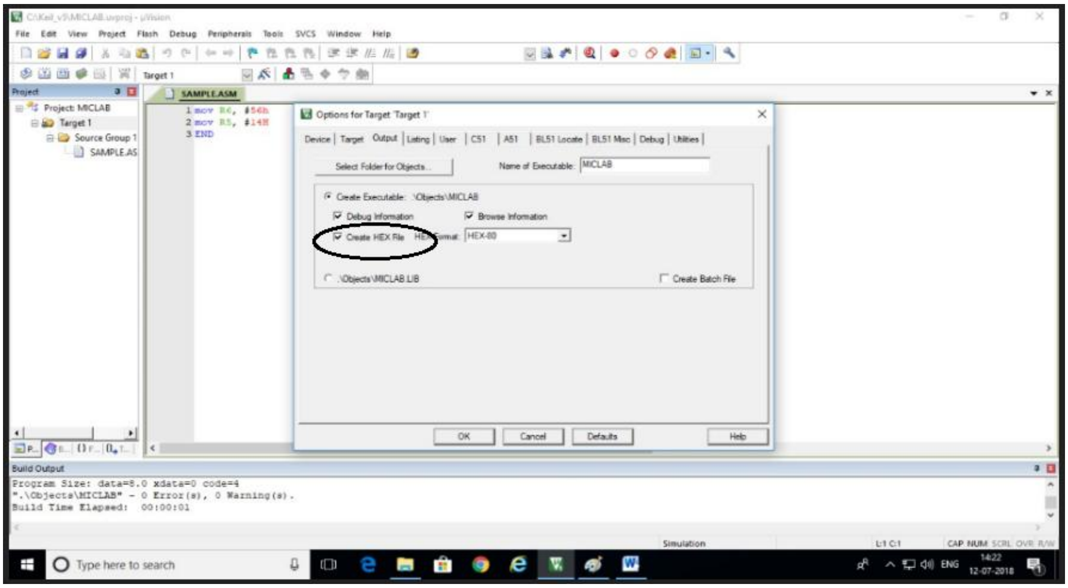
a. Right click on target in project window.
b. Click on options for target ‘target 1’.
c. Set target frequency 11.0592 MHz
d. Click on output tab and checkmark the option “create hex file”.
e. Click ok
f. Repeat step 12 again.
   Observe output window. Hex file is created.

This step is optional for the experiments which need only simulation method to observe the results.



```
mov R6, #56h
```

```
Program Size: data=8.0 xdata=0 code=4
".\Objects\MICLAB" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:01
```


C:\Keil_v5\MICLAB.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project: MICLAB
Target 1
Source Group 1
SAMPLE.ASM
Options for Target 'Target 1'
Device Target Output Listing User C51 A51 BL51 Locate BL51 Misc Debug Utilities
Select Folder for Objects...
OK
Cancel
Defaults
Help
Build Output
Simulation
Type here to search

```
mov R6, #56h
mov R5, #14H
END
```

Name of Executable: MICLAB

● Create Executable: .\Objects\MICLAB

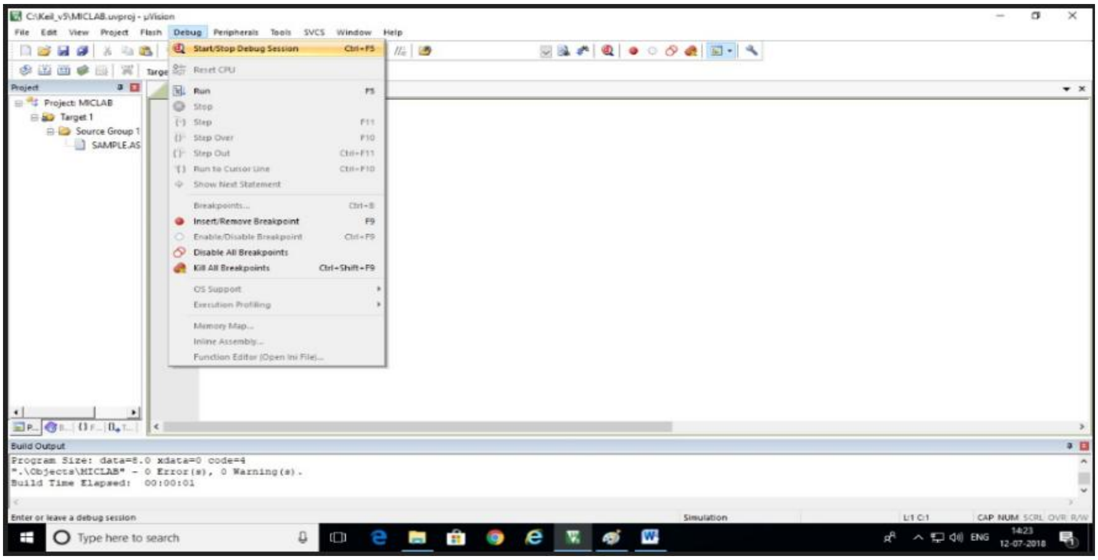
☑ Debug Information ☑ Browse Information

☑ Create HEX File HEX Format: HEX-80

○ .\Objects\MICLAB.LIB □ Create Batch File

```
Program Size: data=8.0 xdata=0 code=4
".\Objects\MICLAB" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:01
```

To start the simulation. Click on Debug pull down. Then select start/Stop debug session

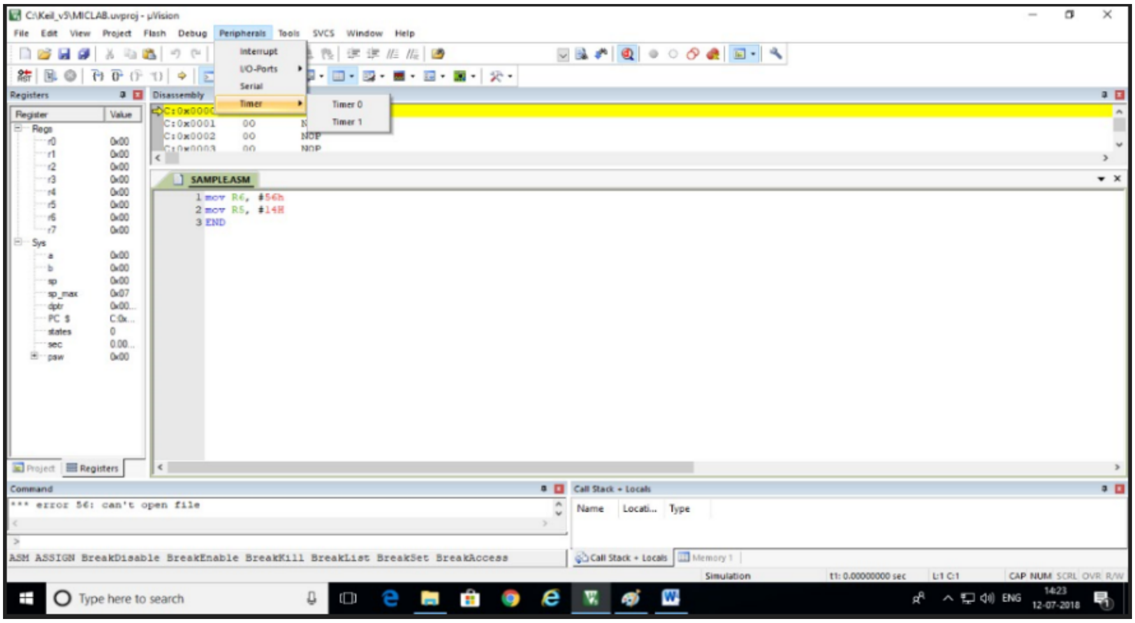

C:\Keil_v5\MICLAB.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Start/Stop Debug Session Ctrl+F5
Reset CPU
Run F5
Stop
Step F11
Step Over F10
Step Out Ctrl+F11
Run to Cursor Line Ctrl+F10
Show Next Statement
Breakpoints... Ctrl+B
Insert/Remove Breakpoint F9
Enable/Disable Breakpoint Ctrl+F9
Disable All Breakpoints
Kill All Breakpoints Ctrl+Shift+F9
OS Support
Execution Profiling
Memory Map...
Inline Assembly...
Function Editor (Open Ini File)...
Project
Project: MICLAB
Target 1
Source Group 1
SAMPLE.AS
Build Output
Enter or leave a debug session
Simulation
L:1 C:1
CAP NUM SCRL OVR R/W
Type here to search
ENG
14:23
12-07-2018

```
Program Size: data=8.0 xdata=0 code=4
".\Objects\MICLAB" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:01
```

On start of debug session, project window will display all internal registers of 8051 and their contents. To execute the program step by step, go no clicking on “step over” button.

Observe the logic levels of port pins, timers, interrupt etc, by clicking on Peripherals and select appropriate.

Execute the program step by step and observe the logic levels on port pins.



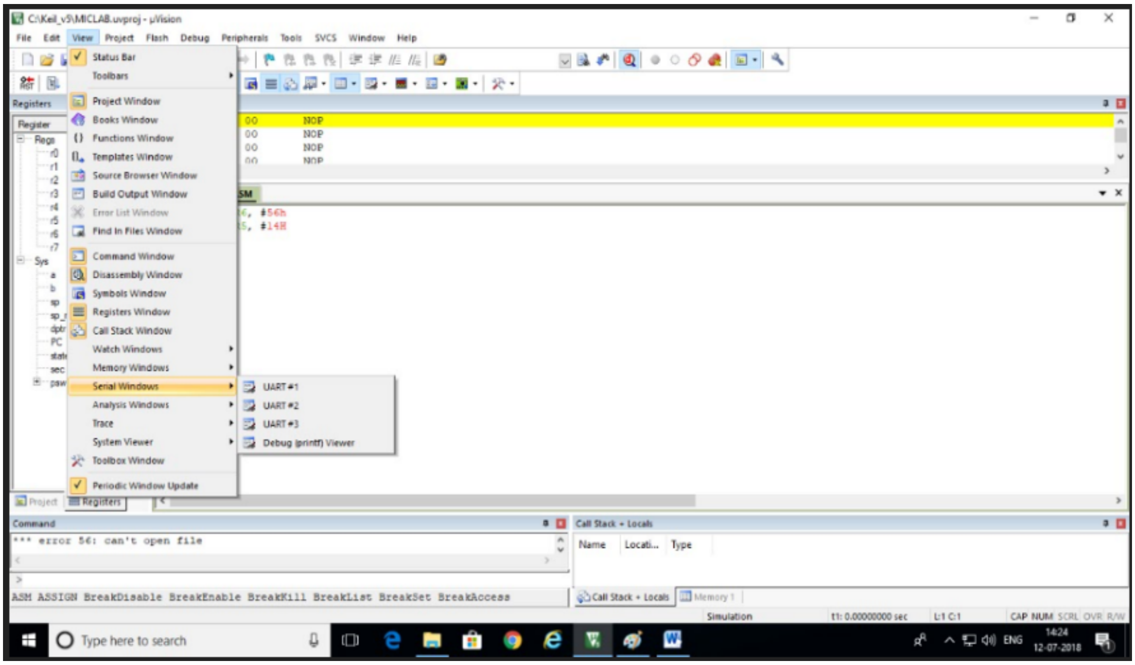
Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x00
sp_max	0x07
dptr	0x00...
PC $	C:0x...
states	0
sec	0.00...
psw	0x00

```
C:0x000
C:0x0001    00    N
C:0x0002    00    NOP
C:0x0003    00    NOP
```

```
mov R6, #56h
mov R5, #14H
END
```

```
*** error 56: can't open file
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Observe the serial communication by clicking VIEW pull down and select serial window-1 Option.



```
00        NOP
00        NOP
00        NOP
00        NOP
```

```
*** error 56: can't open file
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

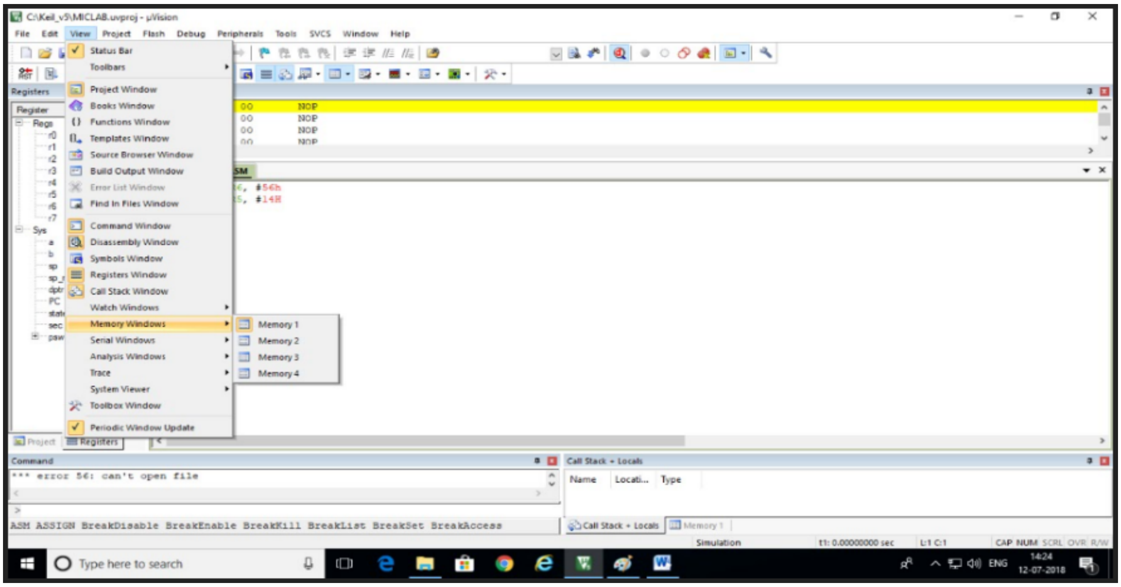
Observe the contents of internal, external and code memory contents.

a. Click on memory window button

b. Memory window will get displayed near output window with address bar.

c. Type “i: address 8-bit H” for internal memory.” X: address16 bit H” for external memory for code memory type “C: address”.

d. To modify the contents of memory, right click on contents of any memory location and enter new valued to be loaded in that memory location



```
00        NOP
00        NOP
00        NOP
00        NOP
```

```
*** error 56: can't open file
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Please note that the features of Sr. no.16 to 18 are available only in debug mode.

XI Resources used

Sr. No.	Instrument /Components	Specification	Quantity
1.			
2.			

XII Actual procedure followed (use blank sheet provided if space is not sufficient)

....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....

XIII Precautions followed (use blank sheet provided if space not sufficient)

....
....
....
....

XIV Observations (use blank sheet provided if space not sufficient)

Observe development board and list various components and write their functions.

Sr. No.	Component	Function

XV Result (Identification of various block of 8051 development board)

....

....

....

....

XVI Interpretation of Results (Hardware and software features of 8051 development board and Keil)

....

....

....

....

XVII Conclusions and Recommendations

....

....

....

....

XVIII Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. List internal and external interrupts available to 8051 microcontroller?
2. List any four features of Keil IDE.
3. List MCS51 family members and write their specifications.
4. Explain the selection criteria for selecting a microcontroller device.
5. List I/O ports available in 8051 microcontroller.

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

....

....

....

....

....

....

....

....

XIX References / Suggestions for further Reading

1. http://www.keil.com
2. https://en.wikipedia.org/wiki/Intel_MCS-51
3. http://www.circuitstoday.com/getting-started-with-keil-uvision
4. https://www.ee.iitb.ac.in/uma/~wel/wel12/.../ATMEL%2089C51.pdf

XX Assessment Scheme

Performance Indicators		Weightage
Process related: 15 Marks		**60%**
1	Use of IDE tools for programming	20%
2	Identifying components on developer kit	30%
3	Follow ethical practices.	10%
Product related: 10 Marks		**40%**
4	Observations and recording	20%
5	Answer to sample questions.	15%
6	Timely Submission of report.	05%
Total (25 Marks)		**100 %**

Name of Team Members

1. …………………………………..
2. …………………………………..
3. …………………………………..
4. …………………………………..

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No. 2: Write sample assembly language program using various addressing modes and assembler directives

I Practical Significance

The addressing modes specifies the way data can be moved or copied from source to destination location thus providing various options and flexibility for data transfer. This practical will help the students to develop skills to allow the programmer to write structured program which is essential to code maintainability.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems.
- **Lifelong learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics and Telecommunication engineering and allied industry.

III Competency and Practical Skills

This practical is expected to develop the following skills for the industry identified competency ‘**Maintain microcontroller-based systems’:**

- Use of addressing modes and how it provides flexibility in software development.
- Use of directives to instruct the assembler.
- Develop assembly programs for microcontroller.

IV Relevant Course Outcome(s)

- Interpret the program for 8051 in assembly language for the given operations.

V Practical Outcome

- Write sample assembly language program using various addressing modes and assembler directives.

VI Relevant Affective domain related Outcome(s)

1. Follow ethical practices.

VII Minimum Theoretical Background

Addressing Modes: A microcontroller provides various methods for accessing data needed in the execution of an instruction. The various methods of accessing the data are called addressing modes.

1. **Immediate addressing mode:** The data is provided in instruction itself. The data is identified by #before the numerical value
   Ex: MOV A, #05H (The immediate date 05h provided in instruction is moved into A register).

2. **Register addressing mode:** The registers hold the data. The permitted registers are A, R7-R0 of each register bank.
   Ex: MOV A, R0 content of R0 register is copied into Accumulator.

3. **Direct addressing mode:** The data is in the RAM memory location and this address is given as part of instruction.
   Ex: MOV A, 30H Content of RAM address 30H is copied into Accumulator.

4. **Register Indirect addressing mode:** Here the address of memory location is indirectly provided by a register.
   The '@' sign indicates that the register holds the address of memory location
   Ex: MOV A, @R0 Copy the content of memory location whose address is given in R0 register to register A.

5. **Register specific mode:** The operand is specified by certain specific registers such as accumulator or DPTR
   Ex: RRA Rotate the contents of accumulator to the right

6. **Indexed Addressing mode:** This addressing mode is basically used for accessing data from look up table. Here the address of memory is indexed.
   Ex: MOVC A, @A+DPTR. The content of A register is added with content of DPTR and the resultant is the address of memory location from where the data is copied to A register.

Assembler Directives:

The assembler directives are instructions to the assembler to carry out certain activity during the assembly process. The common assembler directives are:

ORG	indicates the beginning of the address
DB	used to define 8-bit data in decimal, binary, hexadecimal, ASCII formats
EQU	used to define a constant without occupying a memory location
END	indicates end of the source file

VIII Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1.	Desktop PC	Loaded with open source IDE, simulation and program downloading software	1 No.

IX Precautions to be Followed

1. Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select 80c51AH 0r AT89C51.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output on the project window. It will display all internal registers of 8051 and their contents.
11. Note the contents of the registers in observation table.

SAMPLE PROGRAM 1: Write a program to move data from accumulator to memory location 30h using immediate and indirect addressing mode.

Step 1-Algorithm

1. Start
2. Move the immediate data 23H into accumulator
3. Initialize R1 with memory location 30H
4. Move the content of the accumulator to the memory location pointer by R1.
5. Stop

Step 2-Flow Chart

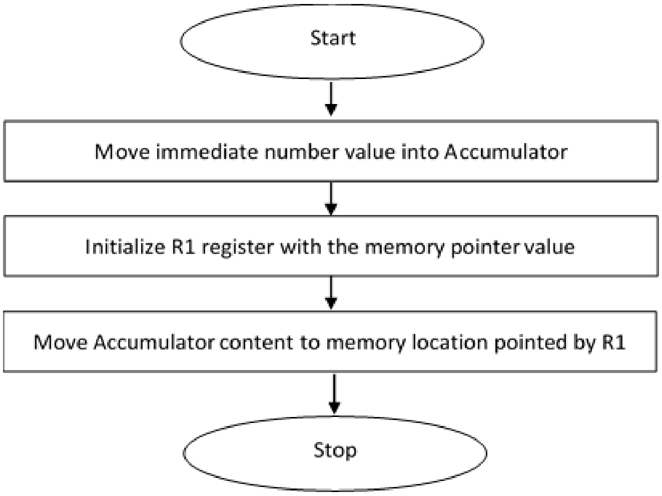
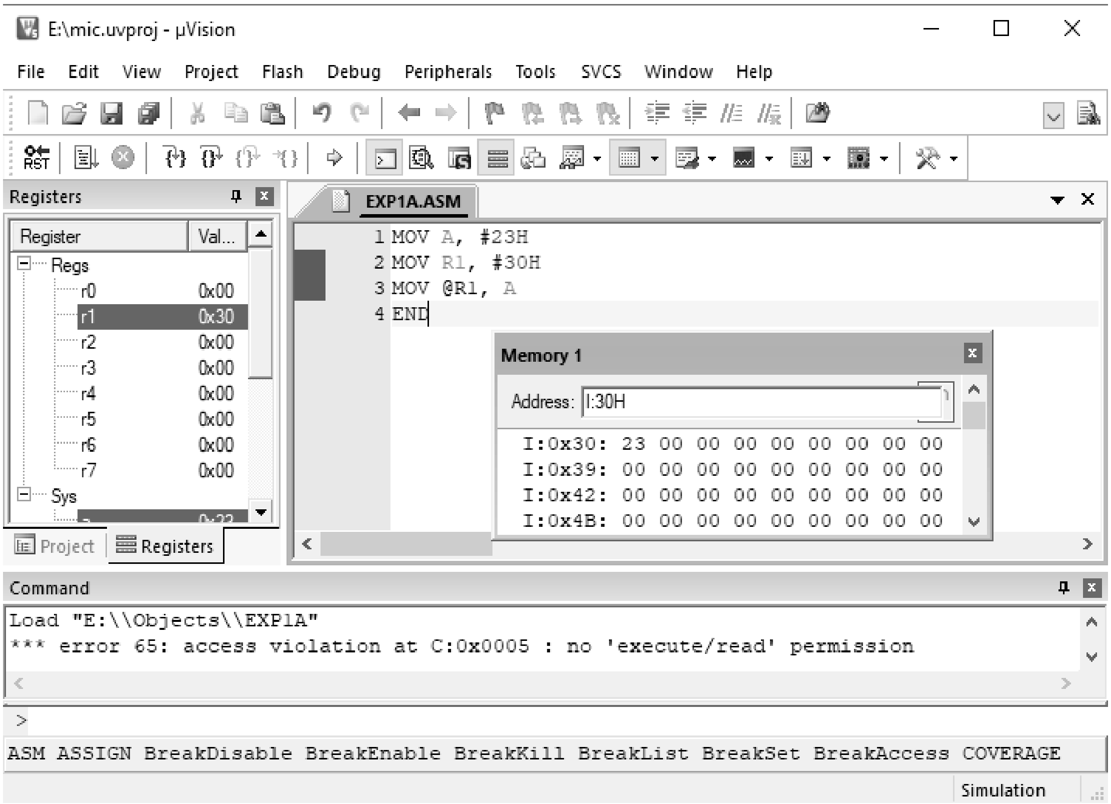

Start
Move immediate number value into Accumulator
Initialize R1 register with the memory pointer value
Move Accumulator content to memory location pointed by R1
Stop

Fig 2.1 Flowchart to move data from accumulator to memory location 30h

Step 3- Assembly Language Sample Program

Memory Address	Hex Code	Label	Mnemonics	Comments
			ORG 0000H	
C:0x0000	7423		MOV A, #23H	;Move the data 23h into accumulator
C:0x0002	7930		MOV R1, #30H	;Move the address 30 h into register R1
C:0x0004	F7		MOV @R1, A	;Move the contents of accumulator in 30h memory location
			END	;Stop



Register	Val...
r0	0x00
r1	0x30
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00

```
MOV A, #23H
MOV R1, #30H
MOV @R1, A
END
```

Address: I:30H

```
I:0x30: 23 00 00 00 00 00 00 00 00
I:0x39: 00 00 00 00 00 00 00 00 00
I:0x42: 00 00 00 00 00 00 00 00 00
I:0x4B: 00 00 00 00 00 00 00 00 00
```

```
Load "E:\\Objects\\EXP1A"
*** error 65: access violation at C:0x0005 : no 'execute/read' permission
```

```
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE
```

Fig 2.2 : Output window

SAMPLE PROGRAM 2: Write a program using ORG, DB and END directives

Step 1-Algorithm

- Start.
- Use ORG directive to set memory location at 0000H.
- Use data directive to write data 27h at memory location 0000H.
- Initialize data pointer to memory location 0000H.
- Clear contents of Accumulator.
- Move into accumulator a byte of data located at address pointed by DPTR.
- Initialize Data pointer to new memory location 2000H.
- Move the contents of Accumulator to address pointed by DPTR.
- Stop

Step 2-Flowchart

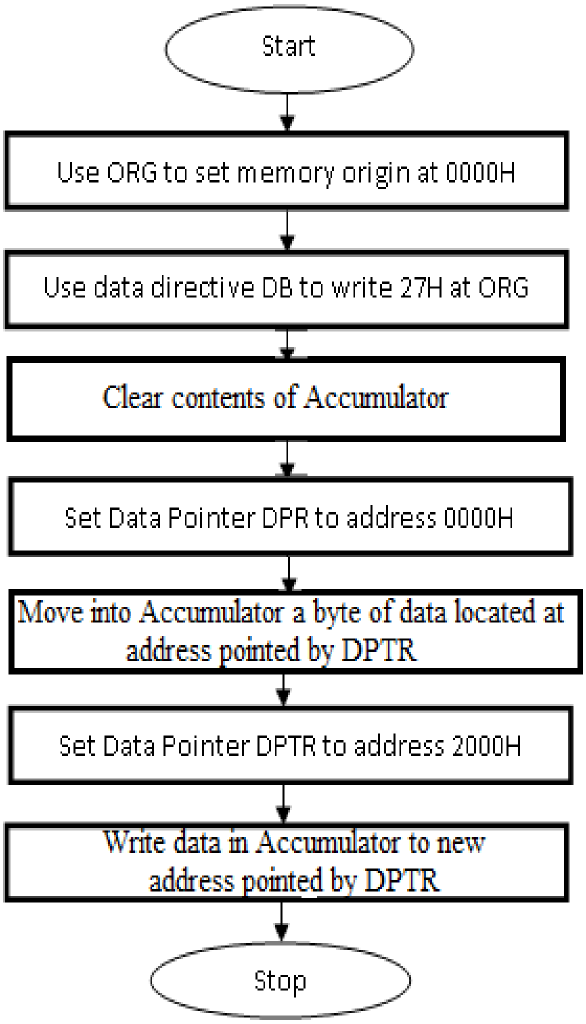
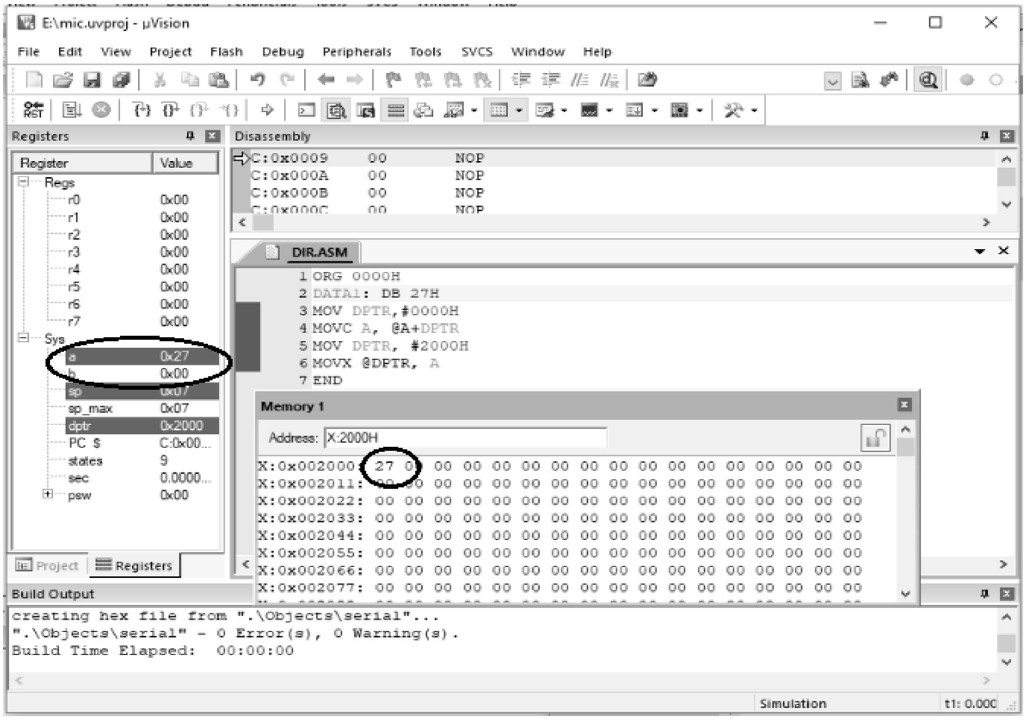

Start
Use ORG to set memory origin at 0000H
Use data directive DB to write 27H at ORG
Clear contents of Accumulator
Set Data Pointer DPR to address 0000H
Move into Accumulator a byte of data located at address pointed by DPTR
Set Data Pointer DPTR to address 2000H
Write data in Accumulator to new address pointed by DPTR
Stop

Fig 2.3 Flowchart for ORG, DB and END directives program

Step 3: Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments
			ORG 0000H	;Use ORG directive to set memory location at 0000H
C:0x0000	27	DATA1:	DB 27H	;8 bit data is Hexadecimal value
C:0x0001	900000		MOV DPTR, #0000H	;Initialize data pointer to 0000H
C:0x0004	E4		CLR A	; Clear Accumulator
C:0x0005	93		MOVC A, @A+DPTR	;Move code byte at ACC+DPTR to ACC
C:0x0006	902000		MOV DPTR, #2000H	;Initialize data pointer to new memory location 2000H
C:0x0009	F0		MOVX @DPTR, A	;Move the contents of accumulator to address pointed by DPTR
			END	;Stop



Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x27
b	0x00
sp	0x07
sp_max	0x07
dptr	0x2000
PC $	C:0x00...
states	9
sec	0.0000...
psw	0x00

```
C:0x0009     00          NOP
C:0x000A     00          NOP
C:0x000B     00          NOP
C:0x000C     00          NOP
```

```
ORG 0000H
DATA1: DB 27H
MOV DPTR,#0000H
MOVC A, @A+DPTR
MOV DPTR, #2000H
MOVX @DPTR, A
END
```

Address: X:2000H

```
X:0x002000: 27 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x002011: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x002022: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x002033: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x002044: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x002055: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x002066: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x002077: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
creating hex file from ".\Objects\serial"...
".\Objects\serial" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:00
```

Fig 2.4 Output window

Problem statement #1 for student: Write a program to transfer the data 25h into register R2 of Bank 0 and register R2 of Bank 1 using immediate and register addressing mode.

Step 1-Algorithm	**Step 2-Flowchart**

Step 3- Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments

Problem statement #2 for student: Write a program using EQU directive.

Step 1-Algorithm	Step 2-Flowchart

Step 3- Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments

XI Resources Used

S. No.	Instrument /Components	Specification	Quantity
1			

XII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....

....

....

....

....

....

....

....

....

....

XIII Precautions Followed (use blank sheet provided if space not sufficient)

....

....

....

....

XIV Observations

After execution of sample program 1

Accumulator	
R1	
30H memory location	

After execution of sample program 2

Accumulator	
DPTR	
2000H memory location	

XV Results (Output of the Program)

....

....

....

....

XVI Interpretation of Results (Give meaning of the above obtained results)

....

....

....

....

XVII Conclusions and Recommendation

....

....

....

XVIII Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. State significance of the symbols @ used in addressing mode.
2. Develop a program to copy the value 55H into RAM memory locations 40H to 41H using (a) direct addressing mode, (b) register indirect addressing mode without a loop, and (c) with a loop.
3. State the contents of each ROM location for the following data

```
ORG 2000H
        DATA1:  DB “Earth”
        DATA2: DB “987-65”
DATA3: DB “GABEH 98”
```

4. Copy the byte in Accumulator to register R2 using at least three different addressing modes.

[Space for Answers]

....

....

....

....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....

....

....

....

....

....

XIX References / Suggestions for further reading

1. The 8051 Microcontroller and Embedded system Using Assembly and C- Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. Mckinlay- Pearson /Prentice Hall, , 2nd edition, Delhi,2008, ISBN 978-8177589030
2. https://nptel.ac.in/courses/Webcourse-contents/IISc-BANG/Microprocessors%20and%20Microcontrollers/pdf/Teacher_Slides/mod2/M2L2.pdf
3. https://nptel.ac.in/courses/Webcourse-contents/IIT-KANPUR/microcontrollers/chap2.pdf

XX Assessment Scheme

Performance Indicators		Weightage
Process related: 15 Marks		**60%**
1	Use of IDE tools for programming	20%
2	Coding and Debugging ability	30%
3	Follow ethical practices.	10%
Product related: 10 Marks		**40%**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition	15%
6	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Name of Team Members

1 …………………………......

2 …………………………......

3 …………………………..

4 …………………………..

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No. 3: Write an assembly language program (ALP) to perform arithmetic operations: addition, subtraction, multiplication and division.

I Practical Significance

8051 microcontroller has single instruction arithmetic operations. Applications such as BCD and ASCII conversions and checksum byte testing require arithmetic operations. This practical will help the students to develop skills to write assembly program for arithmetic operations.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems.
- **Engineering tools**: Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations.

III Competency and Practical Skills

This practical is expected to develop the following skills for the industry identified competency '**Maintain microcontroller based systems':**

- Develop the code for the given problem.
- Debug the code and rectify the errors.
- Run the code.

IV Relevant Course Outcome(s)

- Interpret the program for 8051 in assembly language for the given operations.

V Practical Outcome

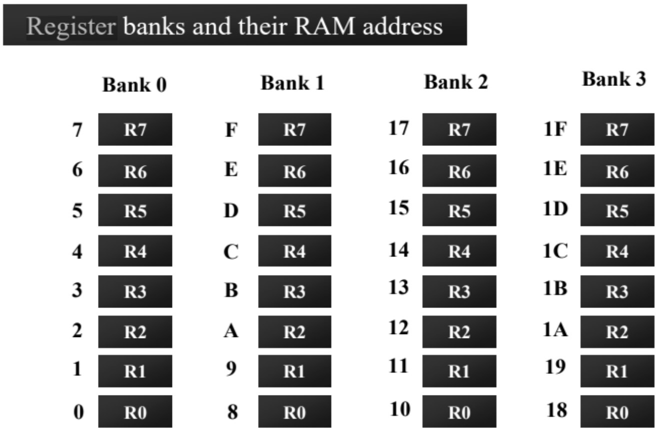
- Write an assembly language program (ALP) to perform arithmetic operations: addition, subtraction, multiplication and division.

VI Relevant Affective domain related Outcome(s)

- Follow ethical practices.

VII Minimum Theoretical Background

Accumulator is an 8 bit register and is used for all arithmetic and logic operations. To perform arithmetic operations, it is necessary that one of the two operands should be in accumulator. Register B is used for multiplication and division purpose. There are four register banks available in 8051; each register bank consists of 8 registers (R0 to R7).

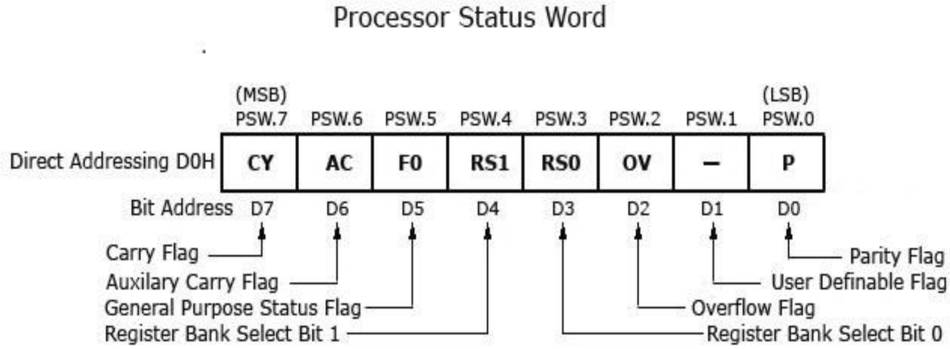


Register banks and their RAM address

Bank 0		Bank 1		Bank 2		Bank 3	
7	R7	F	R7	17	R7	1F	R7
6	R6	E	R6	16	R6	1E	R6
5	R5	D	R5	15	R5	1D	R5
4	R4	C	R4	14	R4	1C	R4
3	R3	B	R3	13	R3	1B	R3
2	R2	A	R2	12	R2	1A	R2
1	R1	9	R1	11	R1	19	R1
0	R0	8	R0	10	R0	18	R0

Fig 3.1 Register Banks

8051 uses DPTR, a 16 bit register to access the 16-bit data from external memory. It is used in MOVX, MOVC command
Arithmetic operations affect flags in PSW register of 8051



Processor Status Word

	(MSB) PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	(LSB) PSW.0
Direct Addressing D0H	CY	AC	F0	RS1	RS0	OV	–	P
Bit Address	D7	D6	D5	D4	D3	D2	D1	D0

Carry Flag
Auxilary Carry Flag
General Purpose Status Flag
Register Bank Select Bit 1
Parity Flag
User Definable Flag
Overflow Flag
Register Bank Select Bit 0

Fig 3.2 Program Status Word Register

Basic arithmetic operations are –

For addition	ADD/ADDC Destination, Source
For subtraction	SUBB Destination, Source
For multiplication	MUL AB
For division	DIV AB

VIII Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1	Desktop PC	Loaded with open source IDE, simulation and program downloading software	1 No.

IX Precautions to be Followed

1. Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL and select AT89C51.
5. Type the program in text editor and save as filename.asm extension.

Compile the Program

6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output on the project window.
11. Note the values of the result of various operations in the observation table.

SAMPLE PROGRAM 1: To perform following: arithmetic operations: addition, subtraction, multiplication and division.

Step 1-Algorithm

1. Move value 02 H in accumulator.
2. Add 03 H to the number stored in accumulator (02 H).
3. Move contents of accumulator to R0 register.
4. Clear accumulator.
5. Move number 05 H in accumulator.
6. Subtract value stored in accumulator (05 H) with 02 H.
7. Move contents of accumulator to register R1.
8. Clear accumulator and carry flag.
9. Move 03H in accumulator, 02H in register B.
10. Multiply the two numbers.
11. Store product in R2 and R3.
12. Clear accumulator.
13. Move 95 H and 10 H in registers A and B.
14. Divide the two numbers.
15. Move result in R4 and R5
16. Stop

Step 2-Flow Chart

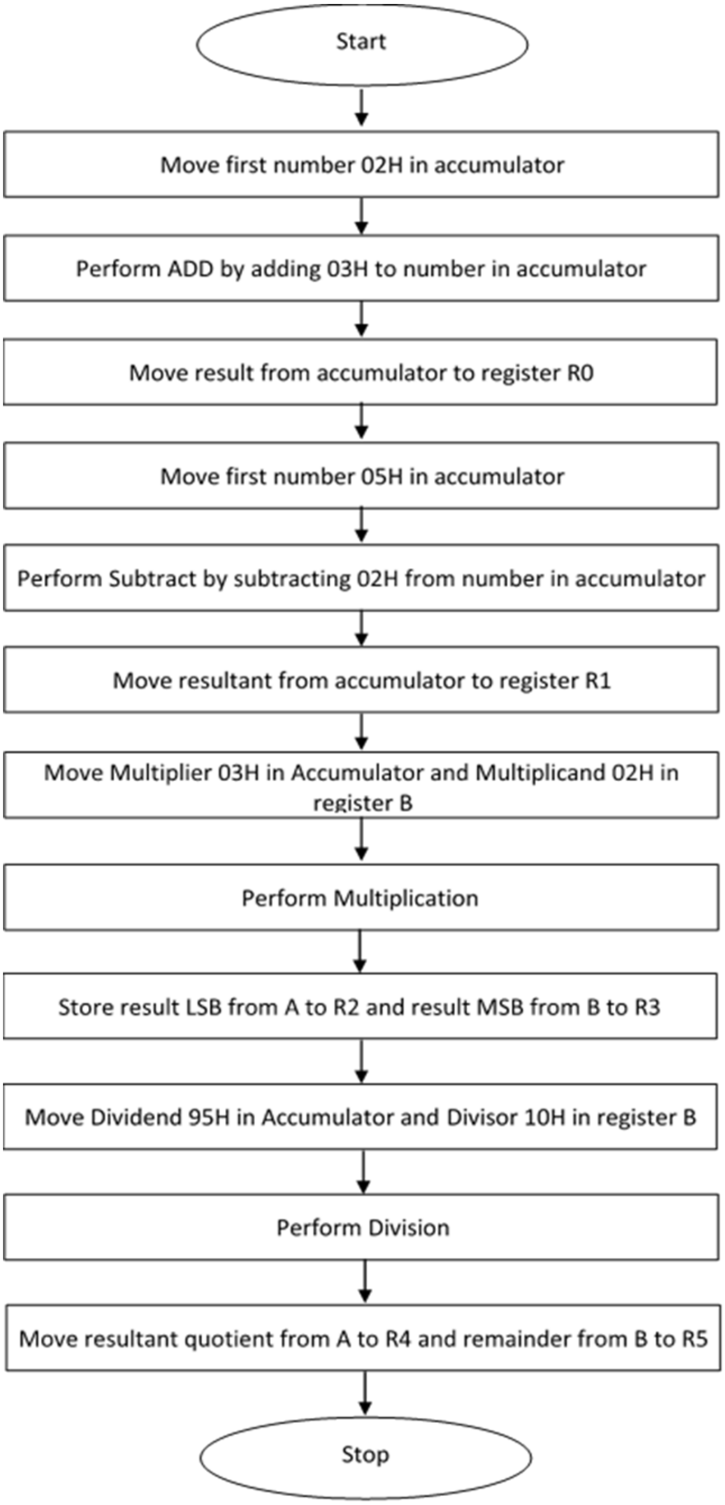
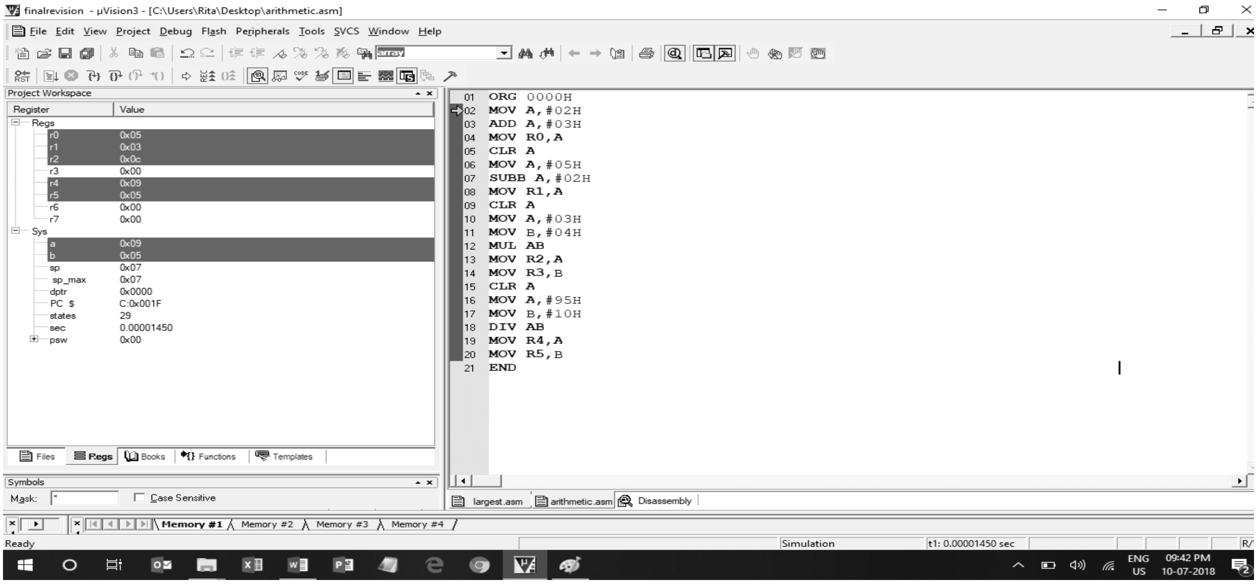

Start
Move first number 02H in accumulator
Perform ADD by adding 03H to number in accumulator
Move result from accumulator to register R0
Move first number 05H in accumulator
Perform Subtract by subtracting 02H from number in accumulator
Move resultant from accumulator to register R1
Move Multiplier 03H in Accumulator and Multiplicand 02H in register B
Perform Multiplication
Store result LSB from A to R2 and result MSB from B to R3
Move Dividend 95H in Accumulator and Divisor 10H in register B
Perform Division
Move resultant quotient from A to R4 and remainder from B to R5
Stop

Fig 3.3 Flowchart for arithmetic operations

Step 3- Assembly Language Sample Program

Memory Address	Hex Code	Label	Mnemonics	Comments
			ORG 0000H	
C:0x0000	7402		MOV A, #02H	;Move the data 02h in accumulator
C:0x0002	2403		ADD A, #03H	;Add the contents of accumulator with 03h
C:0x0004	F8		MOV R0, A	;Store the result in R0 register
C:0x0005	E4		CLR A	;Clear the contents of accumulator
C:0x0006	C3		CLR C	;Clear the carry flag
C:0x0007	7405		MOV A, #05H	;Move the data 05h in accumulator
C:0x0009	9402		SUBB A, #02H	;Subtract the contents of accumulator and 02h
C:0x000B	F9		MOV R1, A	;Store the result in R1 register
C:0x000C	E4		CLR A	;Clear the contents of accumulator
C:0x000D	7403		MOV A, #03H	;Move the data 03h in accumulator
C:0x000F	75F0		MOV B, #04H	;Move the data 04h in accumulator
C:0x0011	A4		MUL AB	;Multiply the contents of A and B
C:0x0012	FA		MOV R2, A	;Store the result (LSB) in R0
C:0x0013	ABF0		MOV R3, B	;Store the result (MSB) in R1
C:0x0015	E4		CLR A	;Clear the contents of accumulator
C:0x0016	7495		MOV A, #95H	;Move the data 95h in accumulator
C:0x0018	75F010		MOV B, #10H	;Move the data 10h in accumulator
C:0x001B	84		DIV AB	;Divide the contents of A and B
C:0x001C	FC		MOV R4, A	;Store the quotient in R4 register
C:0x001D	ADF0		MOV R5, B	;Store the Remainder in R5 register
			END	;Stop


finalrevision - µVision3 - [C:\Users\Rita\Desktop\arithmetic.asm]
File Edit View Project Debug Flash Peripherals Tools SVCS Window Help
array
Project Workspace
Files Regs Books Functions Templates
Symbols
Mask: *
□ Case Sensitive
largest.asm arithmetic.asm Disassembly
Memory #1 Memory #2 Memory #3 Memory #4
ENG US 09:42 PM 10-07-2018

Register	Value
Regs	
r0	0x05
r1	0x03
r2	0x0c
r3	0x00
r4	0x09
r5	0x05
r6	0x00
r7	0x00
Sys	
a	0x09
b	0x05
sp	0x07
sp_max	0x07
dptr	0x0000
PC $	C:0x001F
states	29
sec	0.00001450
psw	0x00

```
ORG 0000H
MOV A,#02H
ADD A,#03H
MOV R0,A
CLR A
MOV A,#05H
SUBB A,#02H
MOV R1,A
CLR A
MOV A,#03H
MOV B,#04H
MUL AB
MOV R2,A
MOV R3,B
CLR A
MOV A,#95H
MOV B,#10H
DIV AB
MOV R4,A
MOV R5,B
END
```

Ready

Simulation

t1: 0.00001450 sec

Fig 3.4 Output window

Problem statement 1 for student: Write a program to perform series addition of ten bytes and store the lower byte of result in 30h and higher byte of result in 31h

Step 1-Algorithm	**Step 2-Flowchart**

Step 3- Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments

XI Resources Used

S. No.	Instrument /Components	Specification	Quantity
1.			

XII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....
....
....
....
....
....
....
....
....
....

XIII Precautions Followed (use blank sheet provided if space not sufficient)

....
....
....
....

XIV Observations for sample program (use blank sheet provided if space not sufficient)

Sr. No.	Register used in the code	Operation	Result after execution
1	R0	Addition	
2	R1	Subtraction	
3	R2	Multiplication (LSB)	
4	R3	Multiplication (MSB)	
5	R4	Division (Quotient)	
6	R5	Division (Remainder)	

XIX Results (Output of the Program)

....
....
....
....

XX Interpretation of Results (Give meaning of the above obtained results)

....
....
....
....

XXI Conclusions and Recommendation

....

....

....

XXII Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. If CY=1, A=95H, and B=4FH prior to the execution of “SUBB A, B”, what will be the contents of A after the subtraction?
2. Is this a valid 8051 instruction? “DIV A, R1”. Explain your answer.
3. Write the status of the CY, AC and P flag after the addition of 9CH and 64H in the following instructions.
   MOV A, #9CH
   ADD A, #64H

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XIX References / Suggestions for further reading

1. http://what-when-how.com/8051-microcontroller/arithmetic-instructions/
2. https://www.youtube.com/watch?v=nLsN7_6FsPs
3. https://www.pantechsolutions.net/...tutorials/subtraction-of-two-numbers-using-8051
4. https://www.youtube.com/watch?v=nLsN7_6FsPs

XX Assessment Scheme

	Performance Indicators	Weightage
	Process related: 15 Marks	**60%**
1	Use of IDE tools for programming	20%
2	Coding and Debugging ability	30%
3	Follow ethical practices.	10%
	Product related: 10 Marks	**40%**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition	15%
6	Timely Submission of report, Answer to sample questions	05%
	Total (25 Marks)	**100 %**

Name of Team Members

1 …………………………......

2 ………………………….......

3 …………………………..

4 …………………………..

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No. 4: Write an ALP to transfer data from source to destination location of internal/external data memory

I Practical Significance

8051 microcontroller is used in embedded system. Understanding its memory organization helps in making optimal use of internal RAM and ROM and for applications which require additional memory, the external memory can be accessed. This practical will help the students to develop skills to transfer data from source to destination location.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems
- **Engineering tools:** Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations.

III Competency and Practical Skills

This practical is expected to develop the following skills for the industry identified competency '**Maintain microcontroller based systems':**

- To use different instructions to transfer, exchange contents of blocks.

IV Relevant Course Outcome(s)

- Interpret the program for 8051 in assembly language for the given operations.

V Practical Outcome

- Write an ALP to transfer data from source to destination location of internal/external data memory.

VI Relevant Affective domain related Outcome(s)

- Follow ethical practices

VII Minimum Theoretical Background

Block Transfer: In block transfer the content of source memory location are transferred to destination memory location. Block transfer is done from internal RAM memory location internal RAM memory location or from internal RAM memory location to external RAM memory location.

External Memory: 8051 microcontroller has 128 bytes of internal RAM memory (data memory) and 4 Kbytes of internal ROM memory (program memory).In addition to internal memory 64 Kbytes of data memory and 64 Kbytes of program can be interfaced with 8051 microcontroller

To interface external memory to 8051 EA' pin should be connected to ground.

.

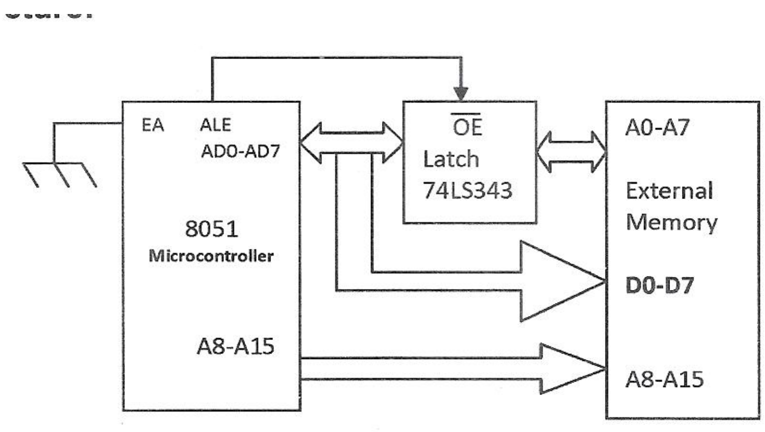


Fig. 4.1 Interfacing of 8051 with external memory

VIII Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1.	Desktop PC	Loaded with open source IDE, simulation and program downloading software	1 No.

IX Precautions to be Followed

1. Check rules/syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select 80c51AH 0r AT89C51.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output on the project window. It will display all internal registers of 8051 and their contents.

SAMPLE PROGRAM 1: Write a program to transfer a block of data from internal memory location 20H to internal memory location 40H.

Step 1: Algorithm

1. Set program starting address.
2. Initialize memory pointer as source.
3. Initialize memory pointer as destination

4. Initialize iteration counter .
5. Move content of first location into accumulator.
6. Move the content of accumulator to first destination location.
7. Increment source location.
8. Increment destination location.
9. Decrement iteration count, and jump to step 5, if not zero.
10. Stop

Step 2: Flowchart

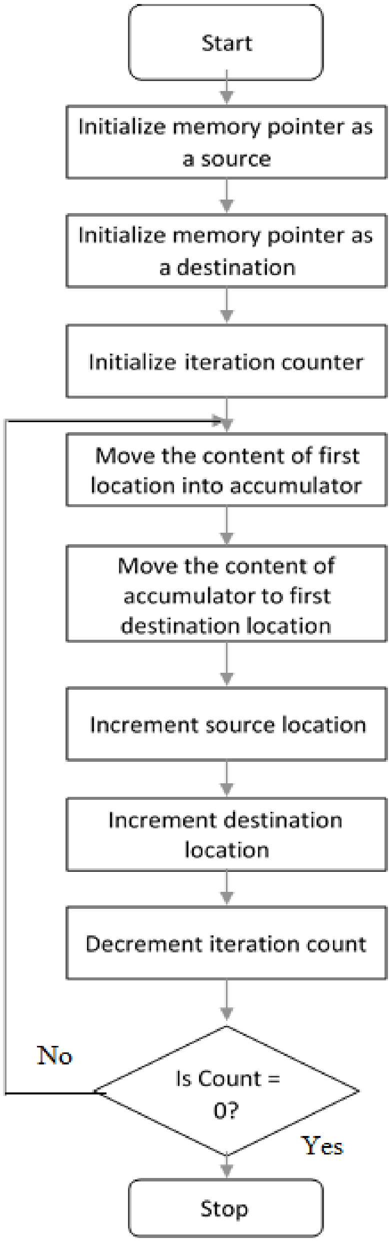
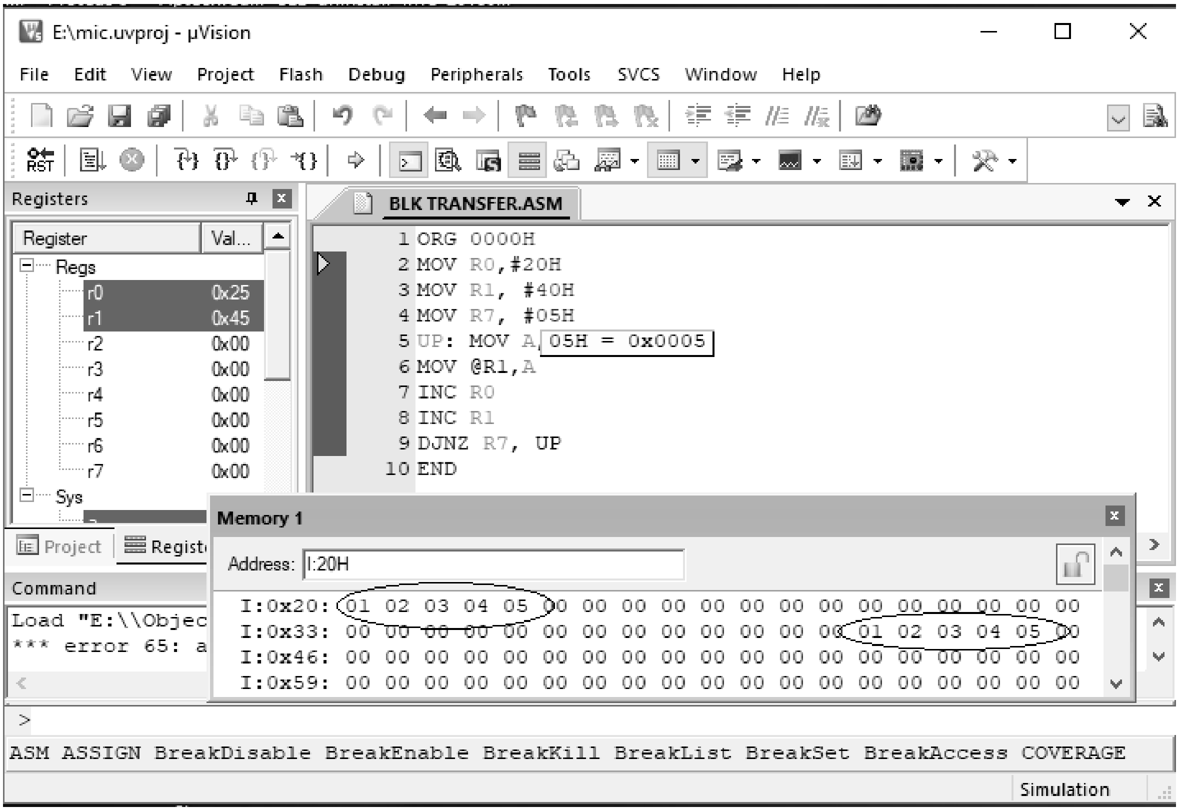

Start
Initialize memory pointer as a source
Initialize memory pointer as a destination
Initialize iteration counter
Move the content of first location into accumulator
Move the content of accumulator to first destination location
Increment source location
Increment destination location
Decrement iteration count
Is Count = 0?
No
Yes
Stop

Fig. 4.2 Flowchart to transfer a block of data

Step 3: Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments
C:0x0000			ORG 0000H	
C:0x0000	7820		MOV R0,#20H	;Initialize source pointer R0 to 20H
C:0x0002	7940		MOV R1, #40H	;Initialize destination pointer R1 to 40H
C:0x0004	7F05		MOV R7, #05H	;Initialize byte counter
C:0x0006	E6	UP:	MOV A, @R0	;Move the contents of first source location to Accumulator
C:0x0007	F7		MOV @R1,A	;Move the contents of Accumulator to first destination location
C:0x0008	08		INC R0	;Increment the contents of R0
C:0x0009	09		INC R1	;Increment the contents of R1
C:0x000A	DFFA		DJNZ R7, UP	; Decrement counter by one Is it zero? No ,jump to UP
			END	



Register	Val...
Regs	
r0	0x25
r1	0x45
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	

```
ORG 0000H
MOV R0,#20H
MOV R1, #40H
MOV R7, #05H
UP: MOV A,
MOV @R1,A
INC R0
INC R1
DJNZ R7, UP
END
```

```
05H = 0x0005
```

Address: I:20H

```
I:0x20: 01 02 03 04 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x33: 00 00 00 00 00 00 00 00 00 00 00 00 00 01 02 03 04 05 00
I:0x46: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x59: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
Load "E:\\Objec
*** error 65: a
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE
```

Fig 4.3 Output window

Problem statement 1 for student: Write a program to transfer a block of five bytes from internal memory location 40H to external memory location 2000H

Step 1: Algorithm	**Step 2-Flowchart**

Step 3- Assembly Language Program Code

Memory Address	Hex Code	Label	Mnemonics	Comments

XI Resources Used

Sr. No.	Name of Resource.	Specification	Quantity
1			

XII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....

....

....

....

....

....

....

....

....

....

....

....

....

XIII Precautions Followed (use blank sheet provided if space not sufficient)

....

....

....

....

XIV Observations for problem statement 1 (use blank sheet provided if space not sufficient)

Before execution		After execution	
Memory location	Data	Memory location	Data
40H		2000H	
41H		2001H	
42H		2002H	
43H		2003H	
44H		2004H	

XV Results (Output of the Program)

....

....

....

....

XVI Interpretation of Results (Give meaning of the above obtained results)

....

....

....

....

XVII Conclusions and Recommendation

....

....

....

....

XVIII Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Give syntax of two instructions used in program to exchange data.
2. Write a program to exchange higher nibble and lower nibble of data 48H.
3. Write the contents of register A after execution of the following instructions

```
MOV A, #28H.
 RRA
 RRA
 END
```

[Space for Answers]

....

....

....

....

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

....

....

....

....

XIX References / Suggestions for further reading

1. www.circuitstoday.com/8051-programming-tutorial-chapter-1.
2. https://thinkelectronics.org › Electronics Projects › 8051 Programs.
3. http://www.polyengineeringtutor.com/Introduction%20to%208051.pdf

XX Assessment Scheme

Performance Indicators		Weightage
Process related: 15 Marks		**60%**
1	Use of IDE tools for programming	20%
2	Coding and Debugging ability	30%
3	Follow ethical practices.	10%
Product related: 10 Marks		**40%**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition	15%
6	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Name of Team Members

1 …………………………......

2 ………………………….......

3 …………………………..

4 …………………………..

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No. 5: Write ALP to find smallest / largest number from the given data bytes stored in internal/external data memory locations

I Practical Significance

CJNE and DJNZ instructions of 8051 microcontroller perform two tasks in a single instruction. The application of these instructions can be demonstrated in finding the smallest and largest numbers from a block of numbers. This practical will help the students to develop skills to use the compare and loop instructions.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems.
- **Engineering tools:** Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations.

III Competency and Practical Skills

This practical is expected to develop the following skills for the industry identified competency: ‘**Maintain microcontroller based systems’:**

- Develop the code for the given problem.
- Debug the code and rectify the errors.
- Run the code
- Use of application of compare and loop instructions.

IV Relevant Course Outcome(s)

- Interpret the program for 8051 in assembly language for the given operations.

V Practical Outcome

- Write ALP to find smallest/largest number from the given data bytes stored in internal/external data memory locations

VI Relevant Affective domain related Outcome(s)

- Follow ethical practices

VII Minimum Theoretical Background

Largest number or smallest number can be found from a block of data using the compare and loop instructions. 8051 microcontroller is an example of CISC machine which has few complex instructions like CJNE and DJNZ.

CJNE destination, source, relative address.

In 8051, the actions of comparing and jumping are combined into a single instruction called CJNE. It compares two operands and jump to relative address if they are not equal. In addition, it changes the CY flag to indicate if the destination operand is larger or smaller. This instruction is used to find the largest/smallest number.

DJNZ Rn, relative

These instruction decrements the contents of register and jump to the relative address if not zero.

Both of these instructions are used to find the largest/smallest number.

VIII Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1.	Desktop PC	Loaded with open source IDE, simulation and program downloading software	1 No.

IX Precautions to be Followed

1. Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select 80c51AH 0r AT89C51.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output on the project window. It will display all internal registers of 8051 and their contents.
11. Note down the readings in observation table

SAMPLE PROGRAM 1: Write ALP to find smallest number from the given five bytes stored in internal memory locations 40H onwards and store the result in location 50H.

Step 1: Algorithm

1. Initialize source pointer R0 to 40H.
2. Initialize byte counter.
3. Load Accumulator with largest one byte number.
4. Move the contents of source location to B register.
5. Compare the two numbers.
6. If number is less then next number then go to step 8.
7. Replace number with next number which is largest.
8. Increment memory pointer to read next number in the array.
9. Decrement byte counter by 1.
10. If byte counter is not zero go to step 4.
11. Store the result.
12. Stop

Step 2: Flowchart

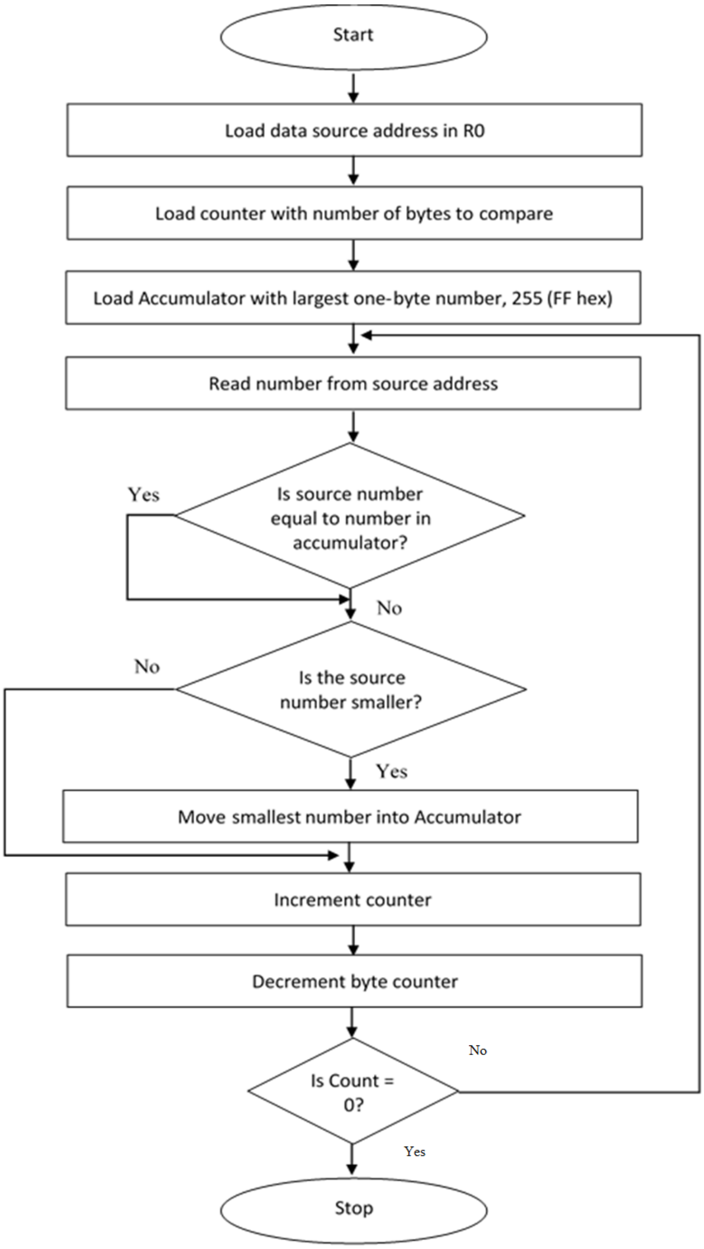
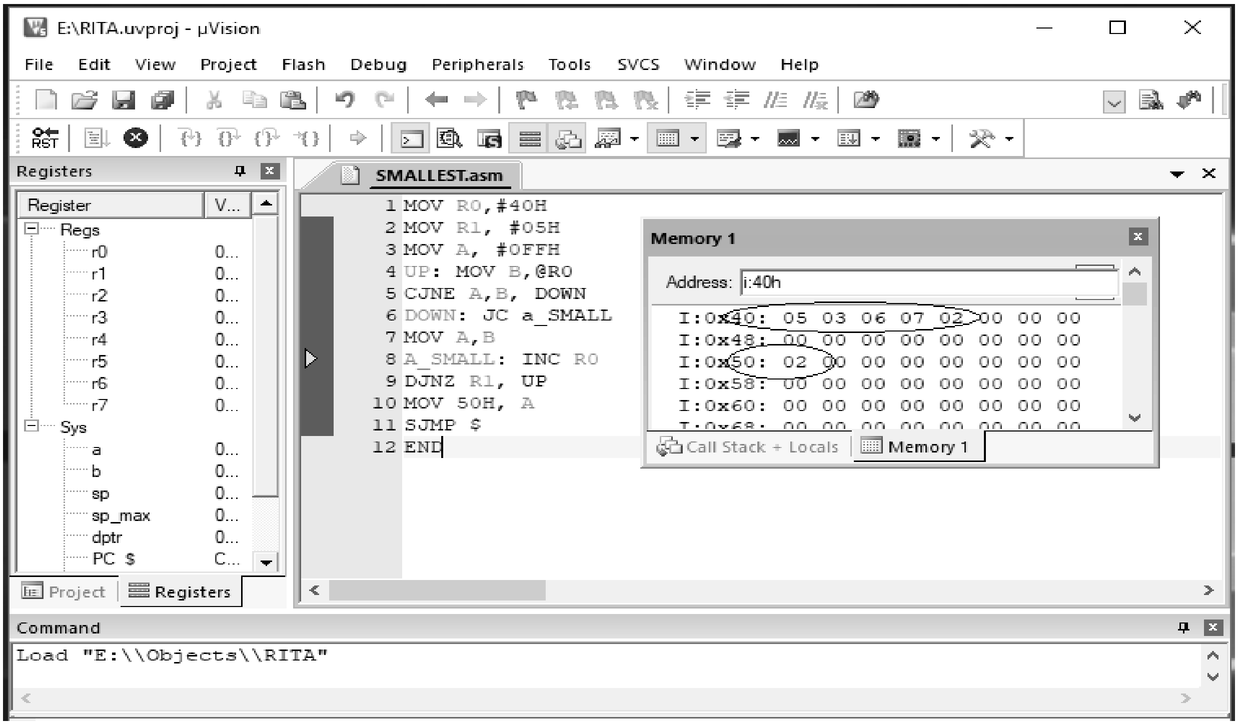

Start
Load data source address in R0
Load counter with number of bytes to compare
Load Accumulator with largest one-byte number, 255 (FF hex)
Read number from source address
Yes
Is source number equal to number in accumulator?
No
No
Is the source number smaller?
Yes
Move smallest number into Accumulator
Increment counter
Decrement byte counter
No
Is Count = 0?
Yes
Stop

Fig 5.1 Flowchart to find smallest number

Step 3: Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments
			ORG 0000H	
C:0x0000	7840		MOV R0,#40H	;Initialize source pointer R0 to 40H
C:0x0002	7905		MOV R1, #05H	;Initialize byte counter
C:0x0004	74FF		MOV A, #0FFH	
C:0x0006	86F0	UP:	MOV B,@R0	;read first byte to B register
C:0x0008	B5F000		CJNE A,B, DOWN	;compare first byte to max value
C:0x000B	4002	DOWN:	JC A_SMALL	;check carry
C:0x000D	E5F0		MOV A,B	;move small value to A
C:0x000F	08	A_SMALL:	INC R0	;Increment the contents of R0
C:0x0010	D9F4		DJNZ R1, UP	; Decrement counter by one Is it zero? No ,jump to UP
C:0x0012	F550		MOV 50H, A	;store smallest number to 50H
C:0x0014	80FE		SJMP $	;Halt after the smallest value is available
			END	



```
MOV R0,#40H
MOV R1, #05H
MOV A, #0FFH
UP: MOV B,@R0
CJNE A,B, DOWN
DOWN: JC a_SMALL
MOV A,B
A_SMALL: INC R0
DJNZ R1, UP
MOV 50H, A
SJMP $
END
```

Address: i:40h

```
I:0x40: 05 03 06 07 02 00 00 00
I:0x48: 00 00 00 00 00 00 00 00
I:0x50: 02 00 00 00 00 00 00 00
I:0x58: 00 00 00 00 00 00 00 00
I:0x60: 00 00 00 00 00 00 00 00
I:0x68: 00 00 00 00 00 00 00 00
```

```
Load "E:\\Objects\\RITA"
```

Fig 5.2 Output window

Problem statement for student: Write ALP to find largest number from the given five bytes stored in memory locations 20H onwards and store the result in location 60H

Step 1: Algorithm	**Step 2-Flowchart**

Step 3- Assembly Language Program Code

Memory Address	Hex Code	Label	Mnemonics	Comments

XI Resources Used

Sr. No.	Name of Resource.	Specification	Quantity
1			

XII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....

....

....

....

XIII Precautions Followed (use blank sheet provided if space not sufficient)

....

....

....

....

XIV Observations for sample program (use blank sheet provided if space not sufficient)

Before execution		After execution	
Memory location	Data	Memory location	Data
40H		50H	
41H			
42H			
43H			
44H			

XV Results (Output of the Program)

....

....

....

....

XVI Interpretation of Results (Give meaning of the above obtained results)

....

....

....

....

XVII Conclusions and Recommendation

....

....

....

....

XVIII Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. State the need of counter in finding the largest or smallest number.
2. List any four instructions to exchange data and give their syntax.
3. Write and execute an ALP to count zero's in a data 64H
4. Write and execute an ALP to find the average of five numbers

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XIX References / Suggestions for further reading

1. http://nptel.ac.in/courses/Webcourse-contents/IIT-KANPUR/microcontrollers/micro/ui/Course_home2_6.htm
2. https://www.elprocus.com/8051-assembly-language-programming/
3. https://thinkelectronics.org/8051-program-to-find-smallest-number-in-blockhttp://what-when-how.com/8051-microcontroller/logic-and-compare-instructions
4. https://www.pearson.com/us/higher-education/product/Mazidi-8051-Microcontroller-and-Embedded-Systems-The/9780138610227.html

XX Assessment Scheme

Performance Indicators		Weightage
Process related: 15 Marks		**60%**
1	Use of IDE tools for programming	20%
2	Coding and Debugging ability	30%
3	Follow ethical practices.	10%
Product related: 10 Marks		**40%**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition	15%
6	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Name of Team Members

1 …………………………......

2 …………………………......

3 …………………………..

4 …………………………..

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No. 6: Write ALP for arranging numbers in ascending / descending order stored in external memory locations.

I Practical Significance

Sorting is any process of arranging information systematically in ascending or descending order. This allows us to write better programs like indexing to fetch the information faster, allows faster search techniques, removes duplicate information and has many uses in statistical applications. This practical will help the students to develop skills to understand how to access data from external memory and use of branch instructions.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems.
- **Engineering tools**: Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations.

III Competency and Practical Skills

This practical is expected to develop the following skills for the industry identified competency '**Maintain microcontroller based systems':**

- Develop the code for the given problem.
- Debug the code and rectify the errors.
- Run the code
- Use of application of instructions related to external memory.

IV Relevant Course Outcome(s)

- Interpret the program for 8051 in assembly language for the given operations.

V Practical Outcome

- Write ALP for arranging numbers in ascending /descending order stored in external memory locations

VI Relevant Affective domain related Outcome(s)

- Follow ethical practices.

VII Minimum Theoretical Background

Ascending order/Descending order

The block of data consists of numbers in random order, to arrange these numbers in ascending or descending order bubble sort method is used.

If the given block of data has to be sorted in ascending order, then bubble sort will start by comparing the first element of the block with the second element, if the first element is greater than the second element, it will swap both the elements, and then move on to compare the second and the third element, and so on.

Byte counter –to access data from block of data

Pass counter –to repeat this comparison are required to arrange the numbers in ascending or descending order.

Branch instructions like JNC—jump if no carry and CJNE—compare and jump if not equal to are used to sort the numbers.

External data memory is read/write. Since external data memory is indirectly accessed through a data pointer register (which must be loaded with an address), it is slower than access to internal data memory.

DPTR, as the name suggests, is used to point to data. It is used by a number of commands which allow the 8051 to access external memory. When the 8051 accesses external memory it will access external memory at the address indicated by DPTR.

MOVX instruction

MOVX is a widely used instruction allowing access to external data memory space. To bring externally stored data into the CPU, we use the instruction “MOVX A, @DPTR”. This instruction will read the byte of data pointed to by register DPTR and store it in the accumulator

VIII Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1	Desktop PC	Loaded with open source IDE, simulation and program downloading software	1 No.

IX Precautions to be Followed

1. Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select 80c51AH 0r AT89C51.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output on the project window. It will display all internal registers of 8051 and their contents.
11. Note down the values of external memory location 3000H in the observation table.

SAMPLE PROGRAM 1: Write ALP for arranging five numbers in ascending order stored in external memory location 3000H.

Step 1-Algorithm

1. Initialize comparison or pass counter
2. Initialize memory pointer to read number from array
3. Initialize byte counter
4. Read numbers from the array
5. Compare two numbers.
6. If number <=next number, then go to step 9.
7. Interchange or swap numbers.
8. Increment memory pointer to read next number from array.
9. Decrement byte counter by one.
10. If word counter is not equal to zero, then go to step 2.
11. Stop

Step 2-Flow Chart

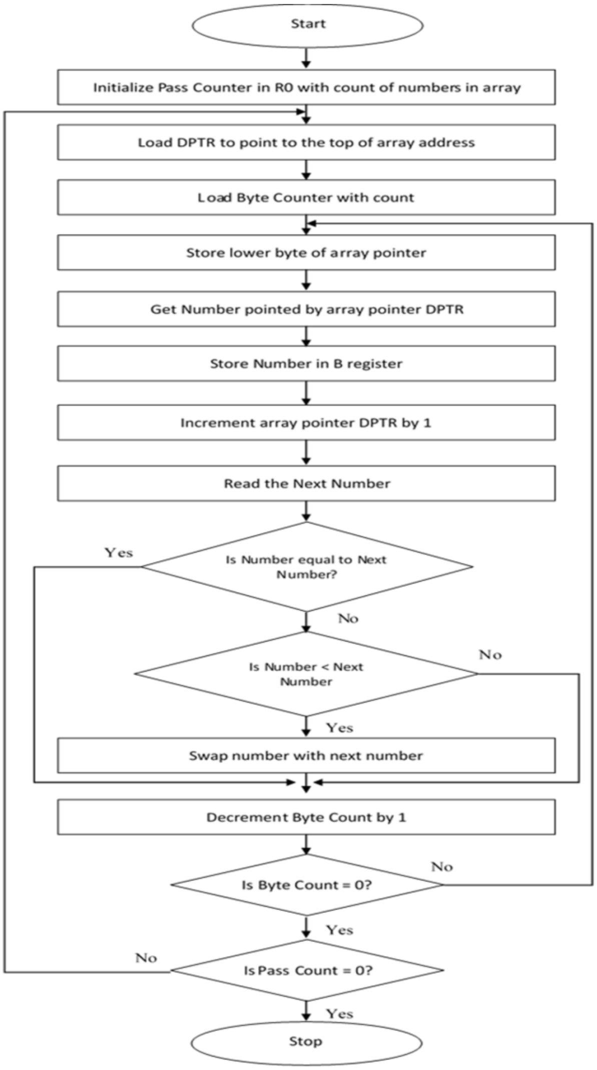
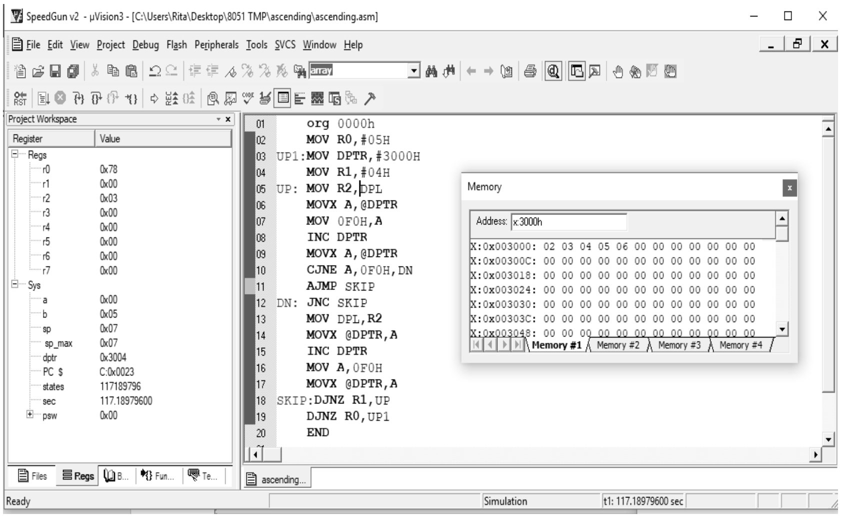

Start
Initialize Pass Counter in R0 with count of numbers in array
Load DPTR to point to the top of array address
Load Byte Counter with count
Store lower byte of array pointer
Get Number pointed by array pointer DPTR
Store Number in B register
Increment array pointer DPTR by 1
Read the Next Number
Yes
Is Number equal to Next Number?
No
Is Number < Next Number
No
Yes
Swap number with next number
Decrement Byte Count by 1
No
Is Byte Count = 0?
Yes
No
Is Pass Count = 0?
Yes
Stop

Fig 6.2 Flowchart to arrange numbers in ascending order

Step 3- Assembly Language Sample Program

Memory Address	Hex Code	Label	Mnemonics	Comments
			ORG 0000h	
C:0x0000	7805	UP1:	MOV R0, #05H	;Initialize pass counter
C:0x0002	903000		MOV DPTR, #3000H	;Initialize memory pointer
C:0x0005	7904	UP:	MOV R1, #04H	;Initialize byte counter
C:0x0007	AA82		MOV R2, DPL	;Save the lower byte address
C:0x0009	E0		MOVX A, @DPTR	;Read number from array
C:0x000A	F5F0		MOV 0F0H, A	;Transfer the number to B register
C:0x000C	A3		INC DPTR	;Increment memory pointer
C:0x000D	E0		MOVX A, @DPTR	;Read next number from array
C:0x000E	B5F002		CJNE A, 0F0H, DN	;Compare number with next number
C:0x0011	011C		AJMP SKIP	
C:0x0013	5007	DN:	JNC SKIP	;If number>next number then go to SKIP
C:0x0015	8A82		MOV DPL, R2	;Else exchange the number with next number
C:0x0017	F0		MOVX @DPTR, A	
C:0x0018	A3		INC DPTR	
C:0x0019	E5F0		MOV A, 0F0H	
C:0x001B	F0		MOVX @DPTR, A	
C:0x001C	D9E7	SKIP:	DJNZ R1, UP	;Decrement byte and if count byte is not zero go to Up
C:0x001E	D8E0		DJNZ R0, UP1	;Decrement pass counter and if not zero go to UP1
			END	


SpeedGun v2 - µVision3 - [C:\Users\Rita\Desktop\8051 TMP\ascending\ascending.asm]
File Edit View Project Debug Flash Peripherals Tools SVCS Window Help
array
Project Workspace
Files Regs B... Fun... Te...
ascending...
Memory
Memory #1 Memory #2 Memory #3 Memory #4
Ready
Simulation
t1: 117.18979600 sec

Register	Value
Regs	
r0	0x78
r1	0x00
r2	0x03
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x05
sp	0x07
sp_max	0x07
dptr	0x3004
PC $	C:0x0023
states	117189796
sec	117.18979600
psw	0x00

```
     org 0000h
     MOV R0,#05H
UP1:MOV DPTR,#3000H
     MOV R1,#04H
UP:  MOV R2,DPL
     MOVX A,@DPTR
     MOV 0F0H,A
     INC DPTR
     MOVX A,@DPTR
     CJNE A,0F0H,DN
     AJMP SKIP
DN:  JNC SKIP
     MOV DPL,R2
     MOVX @DPTR,A
     INC DPTR
     MOV A,0F0H
     MOVX @DPTR,A
SKIP:DJNZ R1,UP
     DJNZ R0,UP1
     END
```

Address: x:3000h

```
X:0x003000: 02 03 04 05 06 00 00 00 00 00 00 00
X:0x00300C: 00 00 00 00 00 00 00 00 00 00 00 00
X:0x003018: 00 00 00 00 00 00 00 00 00 00 00 00
X:0x003024: 00 00 00 00 00 00 00 00 00 00 00 00
X:0x003030: 00 00 00 00 00 00 00 00 00 00 00 00
X:0x00303C: 00 00 00 00 00 00 00 00 00 00 00 00
X:0x003048: 00 00 00 00 00 00 00 00 00 00 00 00
```

Fig 6.2 Output window

1) **Problem statement for student:** Write ALP for arranging five numbers in descending order stored in external memory location 2000H.

.

Step 1-Algorithm	**Step 2-Flowchart**

Step 3- Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments

XI Resources Used

S. No.	Instrument /Components	Specification	Quantity
1.			

XII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....
....
....
....
....
....
....
....
....
....

XIII Precautions Followed (use blank sheet provided if space not sufficient)

....
....
....
....

XIV Observations for sample program (use blank sheet provided if space not sufficient)

Before execution		After execution	
Memory location	Data	Memory location	Data
3000H		3000H	
3001H		3001H	
3002H		3002H	
3003H		3003H	
3004H		3004H	
3005H		3005H	

XV Results (Output of the Program)

....
....
....
....

XVI Interpretation of Results (Give meaning of the above obtained results)

....

....

....

....

XVII Conclusions and Recommendation (Actions/decisions to be taken based on the interpretation of results).

....

....

....

....

XVIII Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. Write a program to (a) load the accumulator with the value 55H, and (b) complement the ACC 7 times.
2. Discuss the program ROM space allocation for each of the following cases. (a) EA’ = 0 for the 8751 (89C51) chip. (b) EA’ = Vcc with both on-chip and off-chip ROM for the 8751. (c) EA’ = Vcc with both on-chip and off-chip ROM for the 8752.

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

....

....

....

....

XIV References / Suggestions for further reading

1. http://nptel.ac.in/courses/Webcourse-contents/IIT-KANPUR/microcontrollers/micro/ui/Course_home2_6.htm
2. https://www.youtube.com/watch?v=FOx8X8U4AXE
3. https://thinkelectronics.org/8051-program-arrange-numbers-ascending-order/
4. https://www.pearson.com/us/higher-education/product/Mazidi-8051-Microcontroller-and-Embedded-Systems-The/9780138610227.html

XX Assessment Scheme

Performance Indicators		Weightage
Process related: 15 Marks		**60%**
1	Use of IDE tools for programming	20%
2	Coding and Debugging ability	30%
3	Follow ethical practices.	10%
Product related: 10 Marks		**40%**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition	15%
6	Timely Submission of report, Answer to sample questions	05%
Total (25 Marks)		**100 %**

Name of Team Members

1 …………………………......

2 ………………………….......

3 …………………………..

4 …………………………..

Marks Obtained			Dated signature of Teacher
Process Related (15)	Product Related (10)	Total (25)	

Practical No. 7: Write an ALP to generate delay using register.

I Practical Significance

Creating a delay is most important and widely used function in almost all microcontroller-based programming. Its use varies from creating a pause to allow microcontroller to interact with real world, like blinking of LED, monitoring a switch or input at regular interval. In signal processing, a “delay” is used to monitor signal at a predetermined sampling rate. This practical will help the students to develop skills to create the time delay by using instruction machine cycles and microcontroller crystal frequency.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems..
- **Engineering tools:** Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations.

III Competency and Practical Skills

This practical is expected to develop the following skills for the industry identified competency ‘**Maintain microcontroller based systems’:**

- Develop the code for given problem.
- Debug the code and rectify the errors.
- Run the code
- To generate delays **by appropriate delay calculations.**

IV Relevant Course Outcome(s)

- Interpret the program by using timer, interrupt and serial ports/parallel ports.

V Practical Outcome

- Write an ALP to generate delay using register.

VI Relevant Affective domain related Outcome(s)

- Follow ethical practices.

VII Minimum Theoretical Background

The delay length in 8051 microcontroller depends on two factors:

1. The crystal frequency.
2. The number of clock per machine.

Machine cycle for the 8051

The CPU takes a certain number of clock cycles to execute an instruction. In the 8051 family, these clock cycles are referred to as *machine cycles.* In the 8051 family, the length of the machine cycle depends on the frequency of the crystal oscillator connected to the 8051 systems. The crystal oscillator, along with on-chip circuitry, provides the clock source for the 8051 CPU. The frequency of the crystal connected to the 8051 family can vary from 4 MHz to 30 MHz, depending on the chip rating and manufacturer.

Very often the 11.0592 MHz crystal oscillator is used to make the 8051 -based system compatible with the serial port of the IBM PC. In the original 8051, one machine cycle lasts 12 oscillator periods. Therefore, to calculate the machine cycle for the 8051, we take 1/12 of the crystal frequency, then take its inverse, as shown below:
For 11.0592MHz/12 = 921.6KHz; machine cycle is 1/921.6KHz=1.085microsecond
Various 8051 versions and its clocks per machine cycle are as shown below:

Chip / manufacturer	Clocks per machine cycle
AT89C51 Atmel	12
P89C54X2 Philips	6
DS85000Dalias Semi	4
DS 89C420/30/40/50 Dalias Semi	1

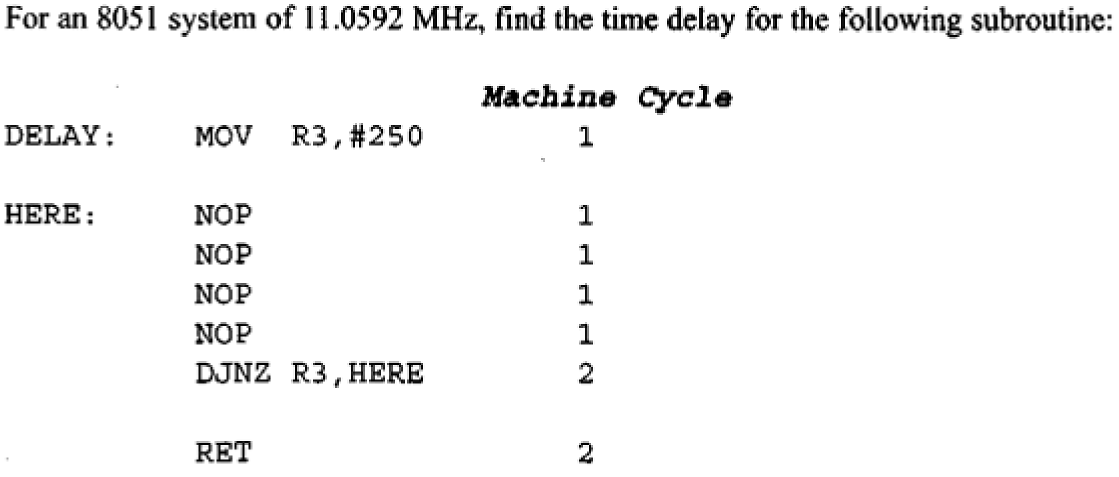
There are two methods for delay calculations which are:
1. LOOP Technique.
2. Using Timer

1. **LOOP TECHNIQUE**: This is very simple software-based delay technique. In this method we load a number in a RAM location and decrement it till it becomes Zero**.**

LOOP WITHIN A LOOP: For longer delays we use this technique. In previous case only a single RAM location was used, here the number depends on the number of LOOPS used

2. USING TIMER: Many times, we require precise internal time delays between two actions this can be accomplished using software techniques like Loop Technique, but these delays keep the processor occupied because of which other important functions cannot be done. To relieve the processor of this burden we can use TIMERS provided by the controller. 8051 has two internal timers T0 and T1 (8052 has 3 timers T0, T1 and T2) these timers can be controlled individually.

The following example explains how to calculate time delay for Loop method:



For an 8051 system of 11.0592 MHz, find the time delay for the following subroutine:

```
                                 Machine Cycle
DELAY:     MOV   R3,#250             1

HERE:      NOP                       1
           NOP                       1
           NOP                       1
           NOP                       1
           DJNZ  R3,HERE             2

           RET                       2
```

Solution:

The time delay inside the HERE loop is [250(1 + 1 + 1 + 1 + 2)] x 1.085 µsec=1500 x 1.085 µsec = 1627.5 µsec. Adding the two instructions outside the loop we have 1627.5 µsec + 3 x l.085 µsec = 1630.755 µsec

VIII Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1	Desktop PC	Loaded with open source IDE, simulation and program downloading software	1 No.

IX Precautions to be Followed

1. Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select 80c51AH 0r AT89C51.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output on the project window. It will display all internal registers of 8051 and their contents.
11. Observe the waveform on logic analyzer and sketch the same in observations

SAMPLE PROGRAM 1: Write a program to toggle all the bits of port 1 every 200 ms. Crystal frequency is 11.0592 MHz

Step- Algorithm

1. Move the contents 55H into accumulator.
2. Send the contents of Accumulator to port P1.
3. Call delay routine
4. Complement the contents of accumulator.
5. Jump back to repeat the steps.
6. Write a delay routine for 200 msec.

Step 2-Flow Chart

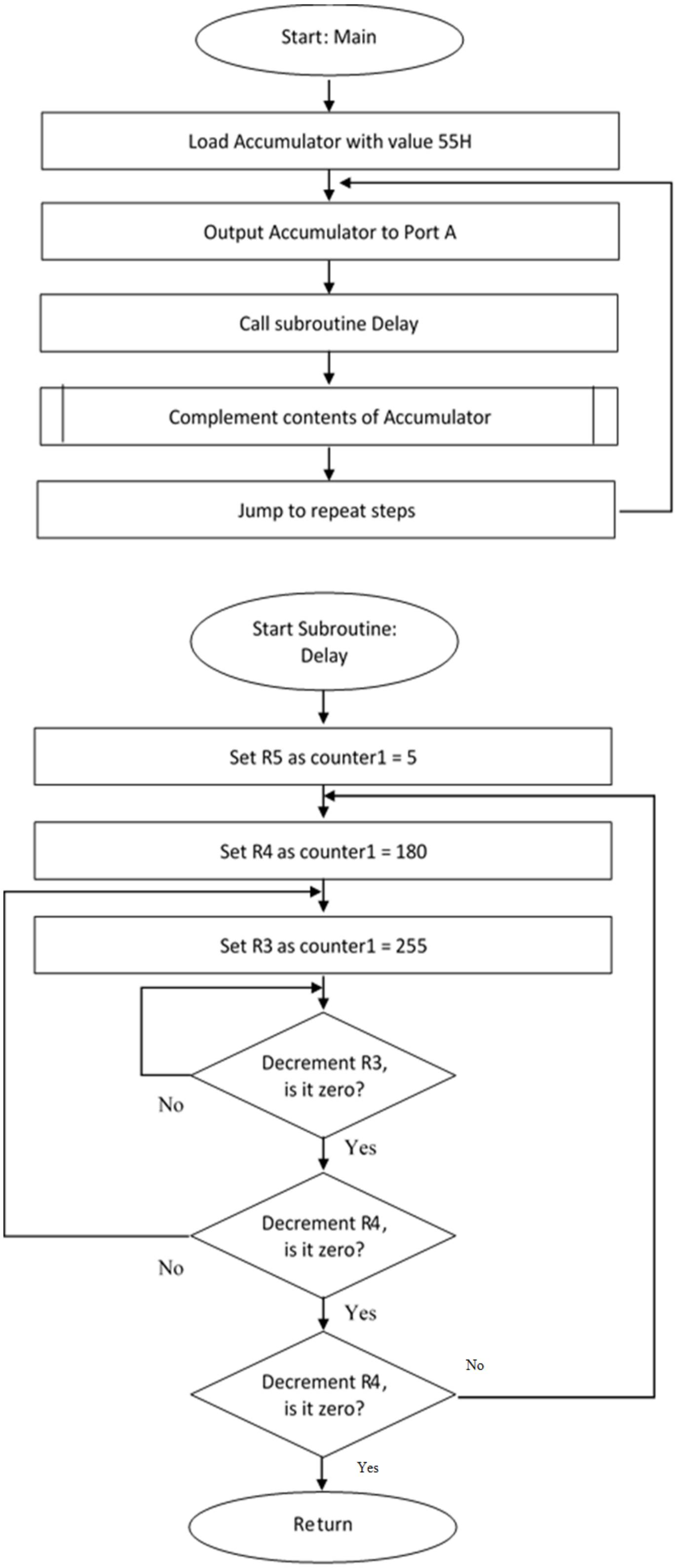

Start: Main
Load Accumulator with value 55H
Output Accumulator to Port A
Call subroutine Delay
Complement contents of Accumulator
Jump to repeat steps
Start Subroutine: Delay
Set R5 as counter1 = 5
Set R4 as counter1 = 180
Set R3 as counter1 = 255
Decrement R3, is it zero?
No
Yes
Decrement R4, is it zero?
No
Yes
Decrement R4, is it zero?
No
Yes
Return

Fig7.1 Flowchart to toggle all the bits of port 1

Step 3-Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments
			ORG 0000H	
C:0x0000	800D		SJMP START	
C:0x0002	7D02	DELAY:	MOV R5, #2	;DELAY routine for 200 msec
C:0x0004	7CB4	HERE1:	MOV R4, #180	
C:0X0006	7BFF	HERE2:	MOV R3, #255	
C:0X0008	DBFE	HERE3:	DJNZ R3, HERE3	
C:0X000A	DCFA		DJNZ R4, HERE2	
C:0x000C	DDF6		DJNZ R5, HERE1	
C:0x000E	22		RET	
C:0x000F	7455	START:	MOV A, #55H	;Load the data 55h into accumulator
C:0X0011	F590	AGAIN:	MOV P1, A	;Send the contents of A to P1
C:0x0013	1102		ACALL DELAY	;Call Delay routine
C:0x0015	F4		CPL A	;Complement the contents of A
C:0x0016	80F9		SJMP AGAIN	;Branch to Again
			END	;Stop.

Sample Calculation of delay:

DELAY:	MOV R5, #2	2 machine cycles
HERE1:	MOV R4, #180	2 machine cycles
HERE2:	MOV R3, #255	2 machine cycles
HERE3:	DJNZ R3, HERE3	2 machine cycles
	DJNZ R4, HERE2	2 machine cycles
	DJNZ R5, HERE1	2 machine cycles
	RET	1 machine cycle

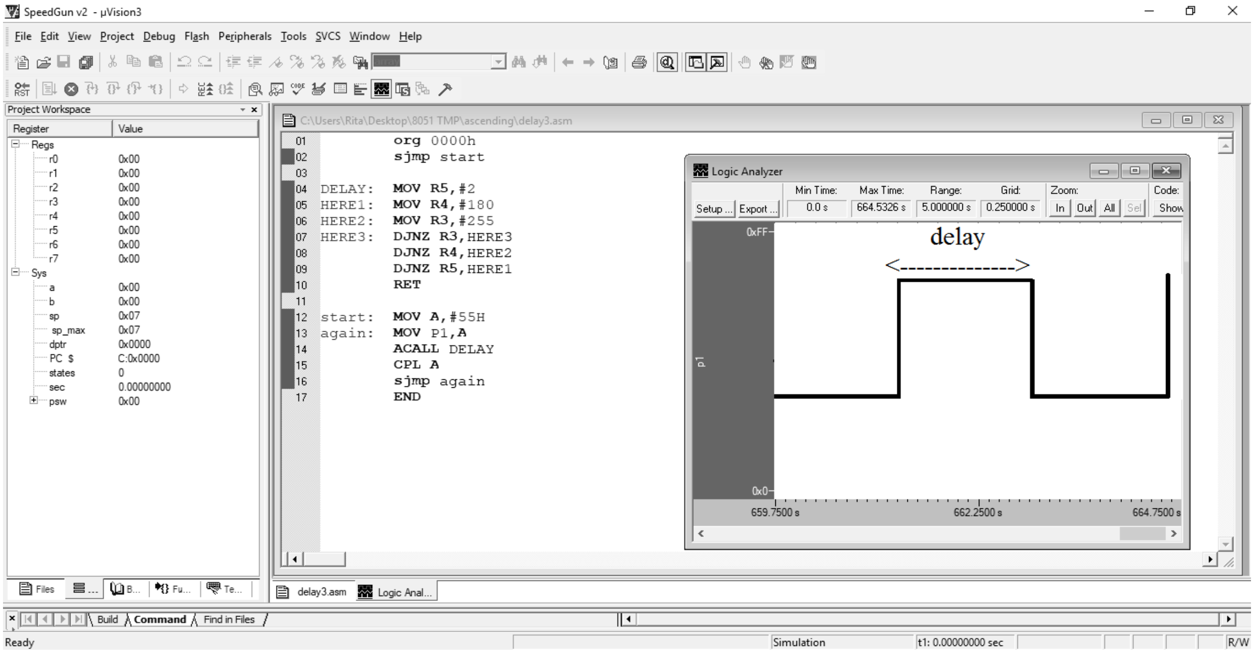
Machine cycles in HERE3 loop = 255 x 2 = 510 machine cycles

Machine cycles in HERE2 loop = 4 + 180 x (machine cycles for HERE3) = 4 + 180 x 510 = 91,804 machine cycles

Machine cycles in HERE1 loop = 4 + 2 x (machine cycles for HERE2) = 4 + 2 x 91804 = 183,608 machine cycles

Machine cycle including RET = 1 + 183,608 = 183,609

Therefore, the total delay by subroutine DELAY = 183,608 x 1.0852 msec = 199.2158 msec = approx. 200 msec



Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC $	C:0x0000
states	0
sec	0.00000000
psw	0x00

```
        org 0000h
        sjmp start

DELAY:  MOV R5,#2
HERE1:  MOV R4,#180
HERE2:  MOV R3,#255
HERE3:  DJNZ R3,HERE3
        DJNZ R4,HERE2
        DJNZ R5,HERE1
        RET

start:  MOV A,#55H
again:  MOV P1,A
        ACALL DELAY
        CPL A
        sjmp again
        END
```

Min Time:	Max Time:	Range:	Grid:
0.0 s	664.5326 s	5.000000 s	0.250000 s

t1: 0.00000000 sec

Fig7.2 Output Window

1. **Problem statement for student**: Write a subroutine to generate a delay of 10ms

Step 1-Algorithm	Step 2-Flowchart

Step 3- Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments

XI Resources Used

S. No.	Instrument /Components	Specification	Quantity
1.			

XII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....

....

....

....

....

....

....

....

....

....

XIII Precautions Followed (use blank sheet provided if space not sufficient)

....

....

....

....

XIV Observations for sample program (use blank sheet provided if space not sufficient)

Draw the waveform and mention the delay

XV Results (Output of the Program)

....

....

....

....

XVI Interpretation of Results (Give meaning of the above obtained results)

....

....

....

....

XVII Conclusions and Recommendation

....

....

....

....

XVIII Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. For an 8051 microcontroller how long will it take to execute the following instruction. The crystal frequency is 11.0592 MHz i) MOV R3,55H ii) DEC R2.
2. On what factors does delay length depend upon?
3. For three different 8051 versions, what will be the machine cycle for the following crystal frequencies:
   i)11.0592MHz ii) 16MHZ iii) 20MHz .Find period of machine cycle in each case

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XIX References / Suggestions for further reading

1. The 8051 Microcontroller and Embedded system Using Assembly and C- Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. Mckinlay- Pearson /Prentice Hall, , 2nd edition, Delhi,2008, ISBN 978-8177589030
2. https://www.youtube.com/watch?v=TWMai3oirnM

XX Assessment Scheme

Performance Indicators		Weightage
Process related: 15 Marks		**60%**
1	Use of IDE tools for programming	20%
2	Coding and Debugging ability	30%
3	Follow ethical practices.	10%
Product related: 10 Marks		**40%**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition	15%
6	Timely Submission of report, Answer to sample questions	05%
Total (Marks)		**100 %**

Name of Team Members

1 ………………………......

2 ……………………….......

3 …………………………..

4 …………………………..

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No. 8: Write an ALP to transfer 8-bit data serially on serial port.

I Practical Significance

Many applications require microcontrollers to either accept the data in serial form or output the data in serial form. Serial communication is commonly used in applications such as industrial automation systems, scientific analysis and certain consumer products. This practical will help the students to develop skills to understand the concepts of serial port and how they are interfaced with microcontroller.

II Relevant Program Outcomes (POs)

- **Discipline knowledge**: Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice**: Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems.
- **Engineering tools**: Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations.

III Competency and Practical skills

This practical is expected to develop the following skills for the industry identified competency '**Maintain microcontroller based systems':**

- Develop the code for the given problem
- Debug the code and rectify the errors.
- Run the code
- Select appropriate value of TH1 register needed for the corresponding baud rate

IV Relevant Course Outcome(s)

- Interpret the program by using timer, interrupt and serial ports /parallel ports

V Practical Outcome(PROs)

- Write an ALP to transfer 8 bit data serially on serial port

VI Relevant Affective domain related Outcome(s)

- Follow ethical practices.

VII Minimum Theoretical Background

In serial communication two formats of data transfer are used 1) Asynchronous 2) Synchronous.

One of the major differences is that in Synchronous Transmission, the sender and receiver should have synchronized clocks before data transmission. Whereas Asynchronous Transmission does not require a clock, but it adds a parity bit to the data before transmission.

1. Asynchronous data transfer.

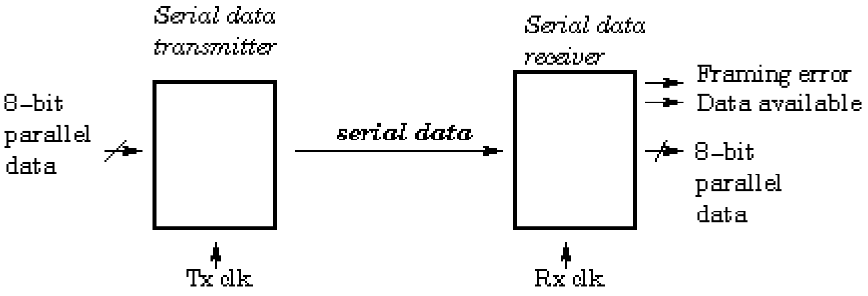


Fig. 8.1 Asynchronous data transfer

2. Synchronous data transfer.

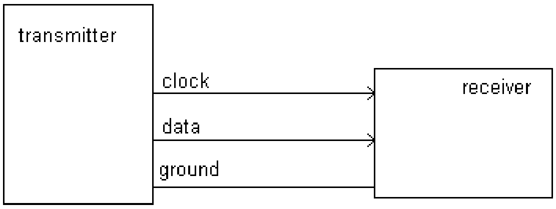


Fig. 8.2 Synchronous data transfer

For serial communication SCON SFR is used. It controls the serial bits.

SCON REGISTER:

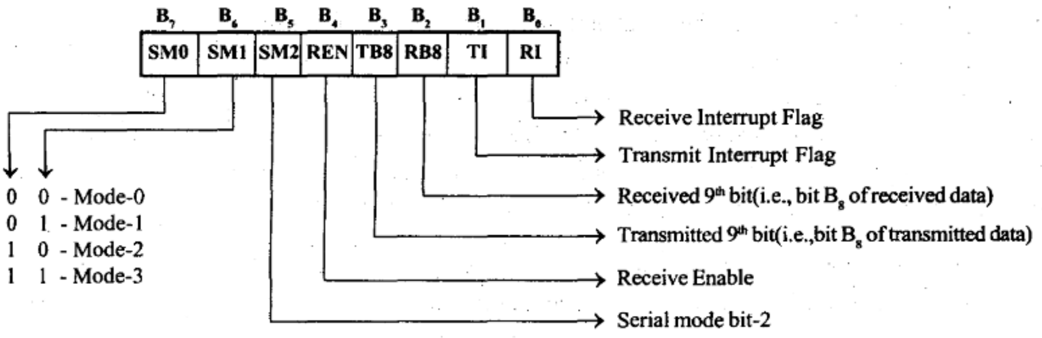


Fig. 8.3 SCON register format

Baud Rate

In serial communication the rate at which data bits are transmitted generates a term baud rate, the baud rate is defined as bits/seconds or the changes in voltage levels/second.

Table No: 8.1 Values of Timer 1 Register for various Baud Rates

Baud Rate	TH1(Decimal)	TH1(Hex)
9600	-3	FD
4800	-6	FA
2400	-12	F4
1200	-24	E8

Note: XTAL = 11.0592 MHz.

SBUF Register

Serial buffer SFR is used for selection of baud rate in serial communication.

SBUF is a 8 bit register used in serial communication of 8051.Serial data is send by writing to the register SBUF while data is received by reading the same register. The SBUF is as shown below:

Bit 7							Bit 0

SBUF has physically two registers, one write only and other is read only. Both registers use one address 99H

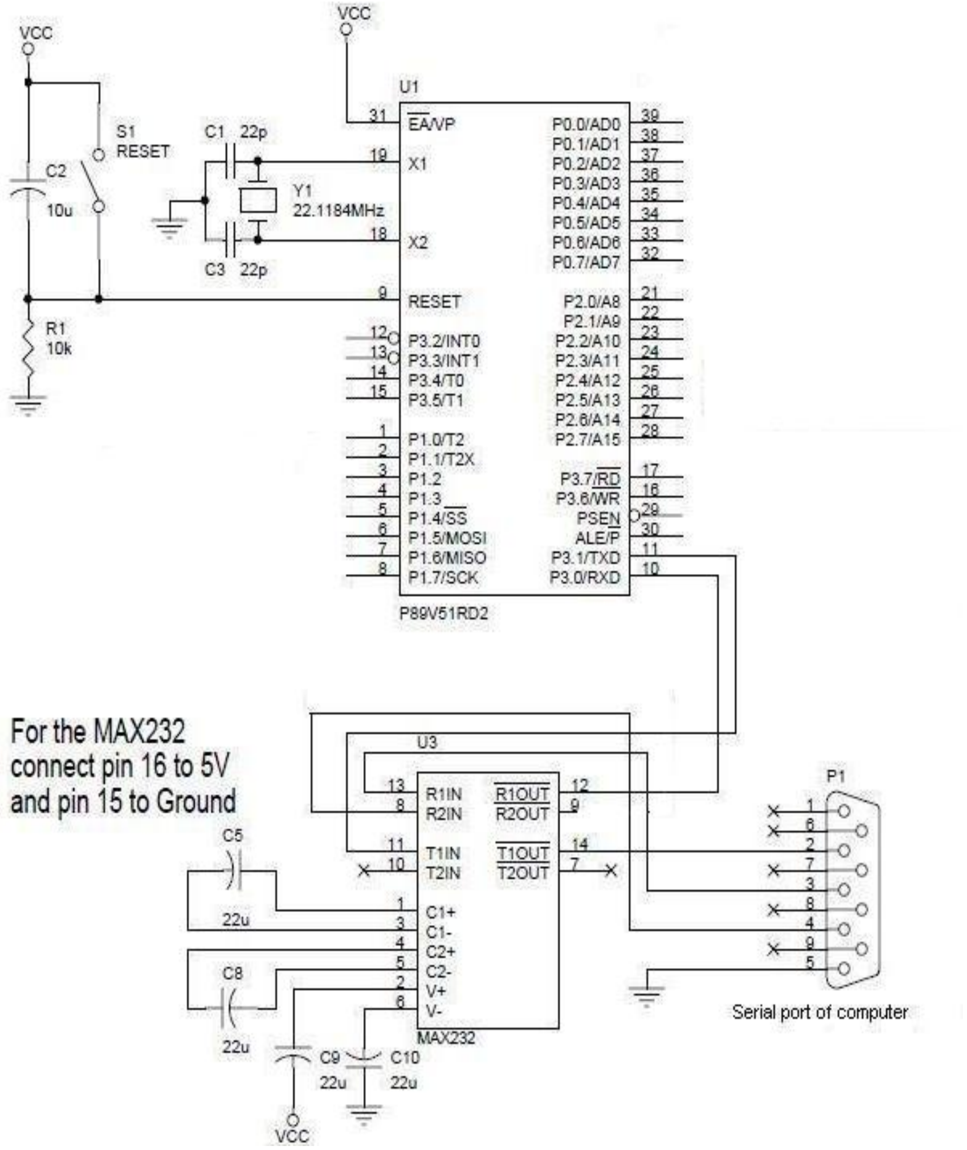


Fig 8.4 Interfacing of 8051 microcontroller to PC with serial communication

A line driver such as the MAX232 chip is required to convert RS232 voltage levels to TTL levels, and vice versa. 8051 has two pins that are used specifically for transferring and receiving data serially. These two pins are called TXD and RXD and are part of the port 3 group (P3.0 and P3.1) these pins are TTL compatible; therefore, they require a line driver to make them RS232 compatible.

VIII Resources required

Sr. No.	Instrument/ Components	Specification	Quantity
1	Microcontroller kit	Single board systems with 8K RAM, ROM memory with battery back up,16X4,16 X2, LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler,RS-232,USB, interfacing facility with built in power supply.	1 No.
2	Desktop PC	Loaded with open source IDE, simulation and program downloading software.	1 No.

IX Precaution

- Check rules / syntax of assembly programming.

X Procedure

Write Program

1. Start Keil by double clicking on Keil icon.
2. Create a new project.
3. Select device for Target.
4. Double click on ATMEL or INTEL and select 80c51AH 0r AT89C51.
5. Type the program in text editor and save as .asm or .a51.

Compile the Program

6. Right click on source group and build the target.
7. Check for any errors in the output window and remove if any.

Run, Debug the Program

8. Click on Debug and start simulation and start/stop debug session.
9. Run the program step by step.
10. Observe the output on the project window. It will display all internal registers of 8051 and their contents. The output can be observed in UART1 window.
11. Hyper Terminal, a Windows XP application, can be used to receive or transmit serial data through RS232. To open Hyper Terminal, go to Start Menu, select all programs, go to Accessories, click on Communications and select Hyper Terminal.
12. To start a new connection, go to File menu and click on new connection. The connection window opens up. Give a name to your connection and select 1st icon and click on OK. Connection property window opens here. Select Bit rate as 9600bps, Data bits 8, Parity as none, stop bit 1, Flow control none and click OK. Now the serial data can be read on hyper terminal.
13. In program, Timer1 is used with auto reload setting. The baud rate is fixed to 9600bps by loading TH1 to 0xFD. The value 0x50 is loaded in the SCON register. This will initialize the serial port in Mode1. The program continuously receives a character (say ‘a’) from the serial port of the computer and transmits it back

SAMPLE PROGRAM 1: Write an ALP to transfer 8 bit data serially on serial port.

Step 1: Algorithm

1) Initialize timer 1 and set it in mode 2.
2) Set the baud rate to 4800.
3) Move 50H in SCON register.
4) Start timer 1.
5) Move letter 'A' to SBUF register.
6) Wait till the last bit is transmitted.
7) Clear TI flag for the next character.
8) Keep sending 'A'.
9) Stop

Step 2: Flowchart

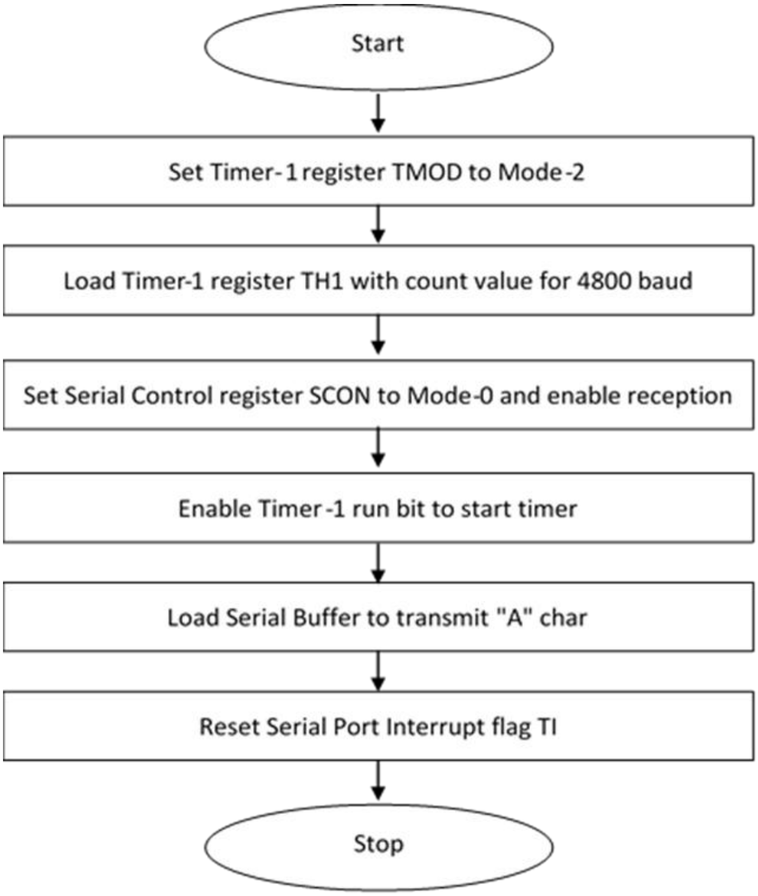
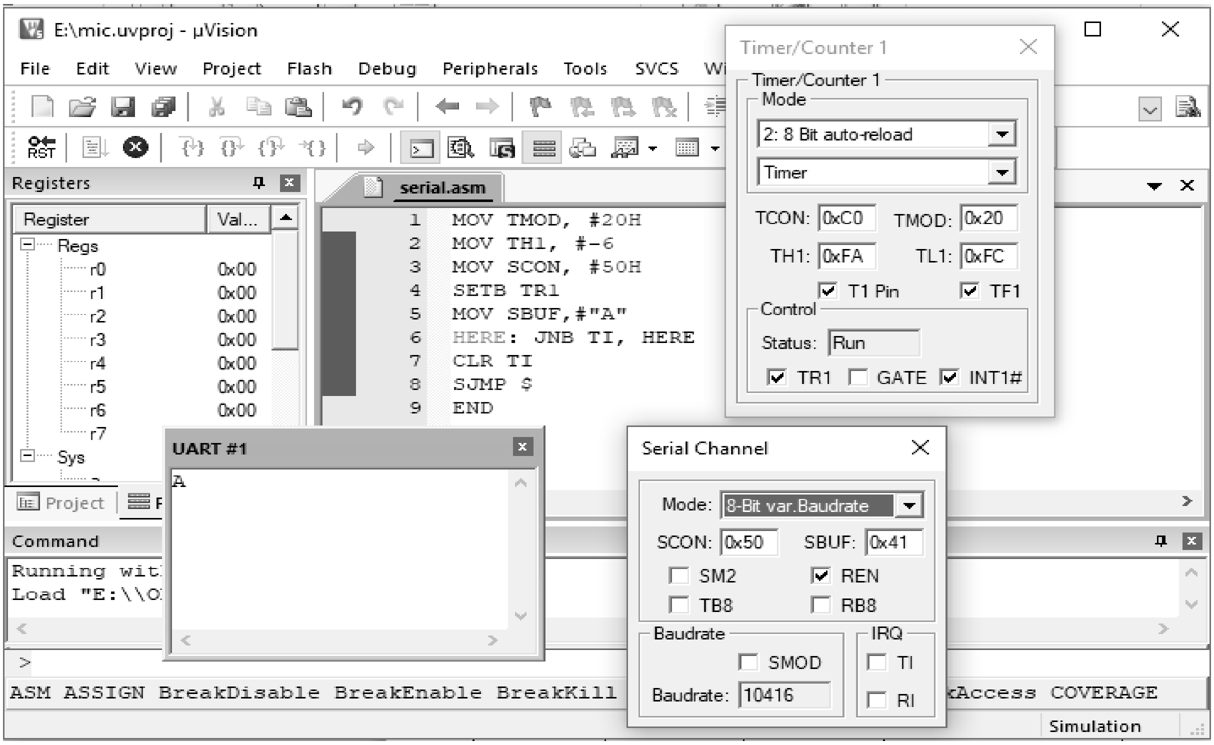

Start
Set Timer- 1 register TMOD to Mode -2
Load Timer-1 register TH1 with count value for 4800 baud
Set Serial Control register SCON to Mode-0 and enable reception
Enable Timer -1 run bit to start timer
Load Serial Buffer to transmit "A" char
Reset Serial Port Interrupt flag TI
Stop

Fig. 8.5 Flowchart to transfer 8 bit data serially on serial port

Step 3- Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments
C:0x0000	758920		MOV TMOD, #20H	;Timer 1,mode 2
C:0x0003	758DFA		MOV TH1, #-6	;4800 baud rate
C:0x0006	759850		MOV SCON, #50H	;8 bit,1 stop, REN enabled
C:0x0009	D28E		SETB TR1	;Start timer 1
C:0x000B	759941		MOV SBUF,#"A"	;Letter “A” to be transferred
C:0x000E	3099FD	HERE:	JNB TI, HERE	;Wait for the last bit
C:0x0011	C299		CLR TI	;Clear T1 for next character
C:0x0013	80FE		SJMP $	;Wait
			END	


E:\mic.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Wi
Registers
Register Val...
Regs
Sys
Project
Command
serial.asm
Simulation
ASM ASSIGN BreakDisable BreakEnable BreakKill
kAccess COVERAGE

```
MOV TMOD, #20H
MOV TH1, #-6
MOV SCON, #50H
SETB TR1
MOV SBUF,#"A"
HERE: JNB TI, HERE
CLR TI
SJMP $
END
```

Registers: r0 0x00, r1 0x00, r2 0x00, r3 0x00, r4 0x00, r5 0x00, r6 0x00, r7

```
Running wit
Load "E:\\O
```

UART #1

```
A
```

Timer/Counter 1

Mode: 2: 8 Bit auto-reload; Timer

TCON: 0xC0 TMOD: 0x20

TH1: 0xFA TL1: 0xFC

☑ T1 Pin ☑ TF1

Control — Status: Run

☑ TR1 □ GATE ☑ INT1#

Serial Channel

Mode: 8-Bit var.Baudrate

SCON: 0x50 SBUF: 0x41

□ SM2 ☑ REN

□ TB8 □ RB8

Baudrate — □ SMOD — Baudrate: 10416

IRQ — □ TI □ RI

Fig 8.6 Output window

Problem statement 1 for student: Write a program to transfer to send letter “YES” at baud rate 9600 continuously**.**

Step 1-Algorithm	**Step 2-Flowchart**

Step 3- Assembly Language Program Code

Memory Address	Hex Code	Label	Mnemonics	Comments

XI Resources Used

Sr. No.	Instrument /Components	Specification	Quantity
1.			
2.			

XII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XIII Precautions Followed (use blank sheet provided if space not sufficient**)**

....

....

....

....

XIV Observations for sample program (use blank sheet provided if space not sufficient) (UART window)

XV Result (Output of the Program)

....

....

....

....

XVI Interpretation of Results (Give meaning of the above obtained results)

....

....

....

....

XVII Conclusions and Recommendation

....

....

....

....

XVIII Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. State the SFR used to transfer character by serial communication.
2. With XTAL = 11.0592 MHz, find the TH1 value needed to have the following baud rates. (a) 9600 (b) 2400 (c) 1200
3. Name the timer of 8051 used to set the baud rate.

[Space for Answers]

....

....

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

....

....

....

....

....

XIX References / Suggestions for further Reading

1 http://nptel.ac.in/courses/Webcourse-contents/IIT-KANPUR/microcontrollers/micro/ui/Course_home2_6.htm

2 https://techdifferences.com/difference-between-synchronous-and-asynchronous-transmission.html

3 https://www.engineersgarage.com/microcontroller/8051projects/interface-serialport-R

XX Assessment Scheme

Performance indicators		Weightage
Process related: 15 Marks		**60%(15)**
1	Use of IDE tools for programming	20%
2	Coding and Debugging ability	30%
3	Follow ethical practices.	10%
Product related: 10 Marks		**40%(10)**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition	15%
6	Timely Submission of report, Answer to sample questions	05%
	Total	**100 %(25)**

Name of Team Members

1 …………………………......

2 ………………………….......

3 …………………………..

4 …………………………..

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No. 9: Interface LED with microcontroller and turn it ON with microcontroller interrupt.

I Practical Significance

LED is most common semiconductor device used in many electronic system as visual indicator or signal transmission / power indication purposes. The LEDs are also used for design message display boards and traffic control signal lights etc. Interrupts are most important feature of Microcontroller. This practical will help the students to develop skills to understand the fundamental interfacing concept for 8051 microcontrollers and significance of Interrupts.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems.
- **Engineering tools:** Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations.
- **Lifelong learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics and Telecommunication engineering and allied industry

III Competency and Practical Skills

This practical is expected to develop the following skills for the industry identified competency '**Maintain microcontroller based systems':**

- Interface LED and switch to microcontroller.
- Use of key to provide external interrupt on pin P3.2.

IV Relevant Course Outcome(s)

- Interpret the program by using timer, interrupt and serial ports /parallel ports.

V Practical Outcome

- Interface LED with microcontroller and turn it ON with microcontroller interrupt.

VI Relevant Affective domain related Outcome(s)

- Follow safe practices.
- Maintain tools and equipment.
- Follow ethical practices.

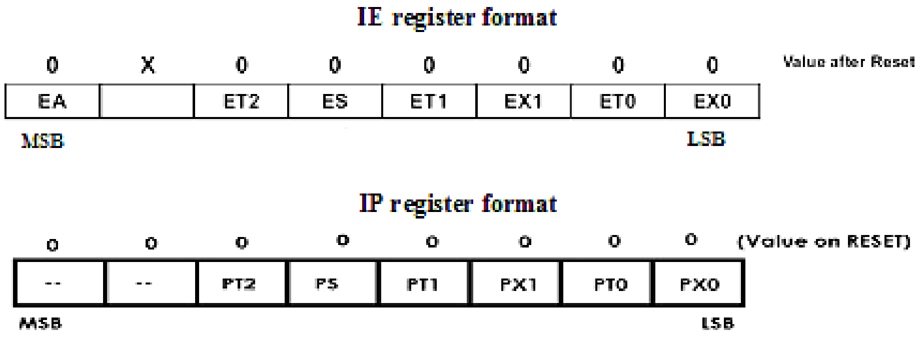
VII Minimum Theoretical Background

As its name suggests, **Interrupt** is a subroutine call that interrupts microcontrollers main operations or work and causes it to execute any other program, which is more important at the time of operation. The feature of Interrupt is very useful as it helps in case of emergency operations. An Interrupts gives us a mechanism to put on hold the ongoing operations, execute a subroutine and then again resume its original task.

Generally five interrupt sources are there in 8051 Microcontroller. Out of these, $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ are external interrupts that could be negative edge triggered or low level triggered. They are located on pins P3.2 and P3.3of port 3 respectively. They are enabled or disabled using the IE register when the external interrupt flag is edge triggered, the CPU clears interrupt flag in response to the interrupt call. When it is level triggered then the interrupt flag is cleared at high level of the interrupt signal.

The table shows the vector addresses for the interrupts:

Interrupt	Flag	Vector
System reset	RST	0000H
External interrupt 0	IE0	0003H
Timer 0	TF0	000BH
External interrupt 1	IE1	0013H
Timer 1	TF1	001BH
Serial port	RI or TI	0023H



IE register format

0	X	0	0	0	0	0	0	Value after Reset
EA		ET2	ES	ET1	EX1	ET0	EX0	
MSB							LSB	

IP register format

0	0	0	0	0	0	0	0	(Value on RESET)
--	--	PT2	PS	PT1	PX1	PT0	PX0	
MSB							LSB	

Fig 9.1 IE and IP register format

Light emitting diodes: LEDs are the most commonly used components in many applications. It has two terminals positive and negative as shown in the figure:

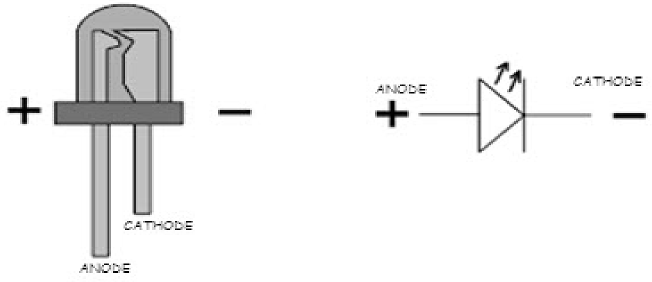


Fig 9.2 LED diagram

Commonly used LEDs will have voltage drop of 1.9v to 2.1v and current of 15mA (Typically) or 20mA(high brightness) to glow at full intensity. This is applied through the output pin of the microcontroller

VIII Practical Circuit diagram:

a) Sample Circuit diagram

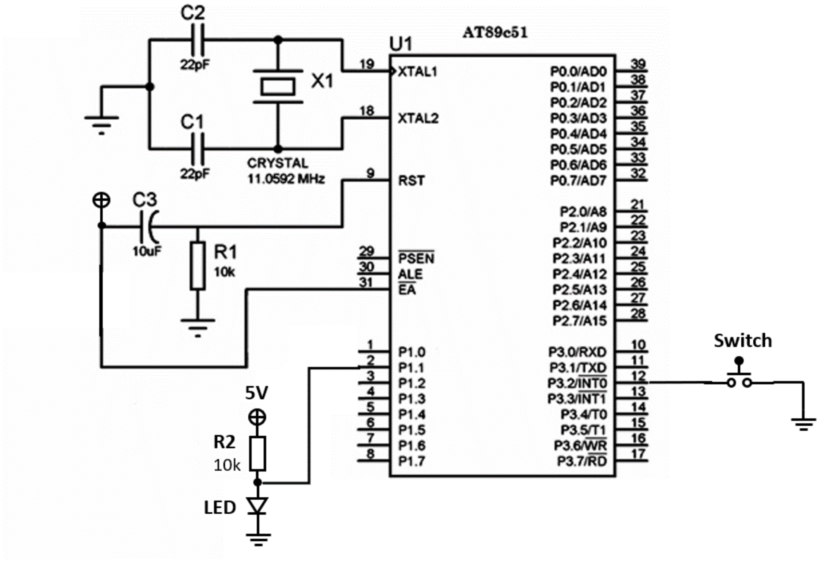


Fig 9.3 8051 connection to LED and switch

b) Practical setup

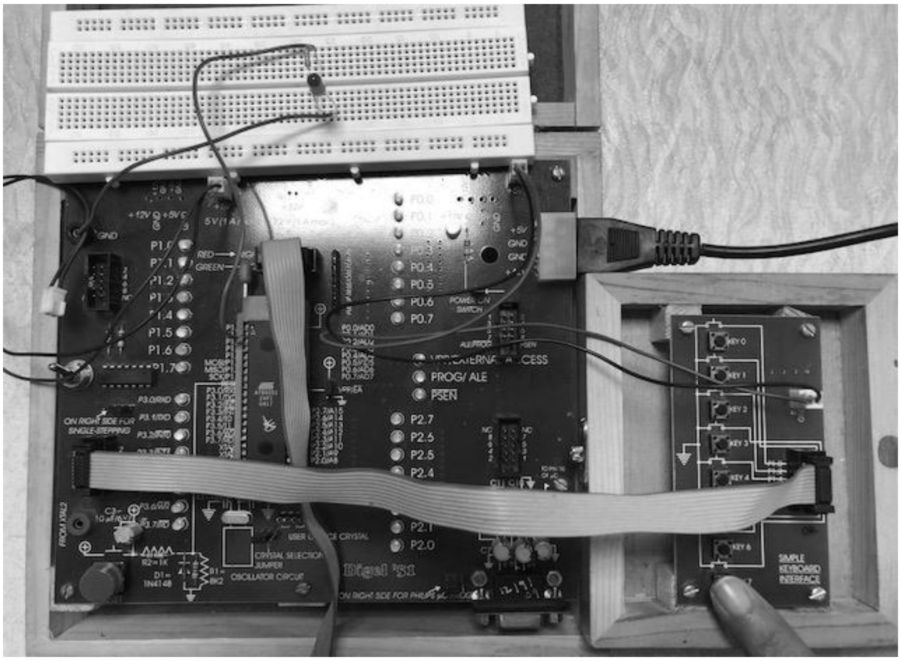


Fig 9.4
Practical Setup

c) Simulation diagram

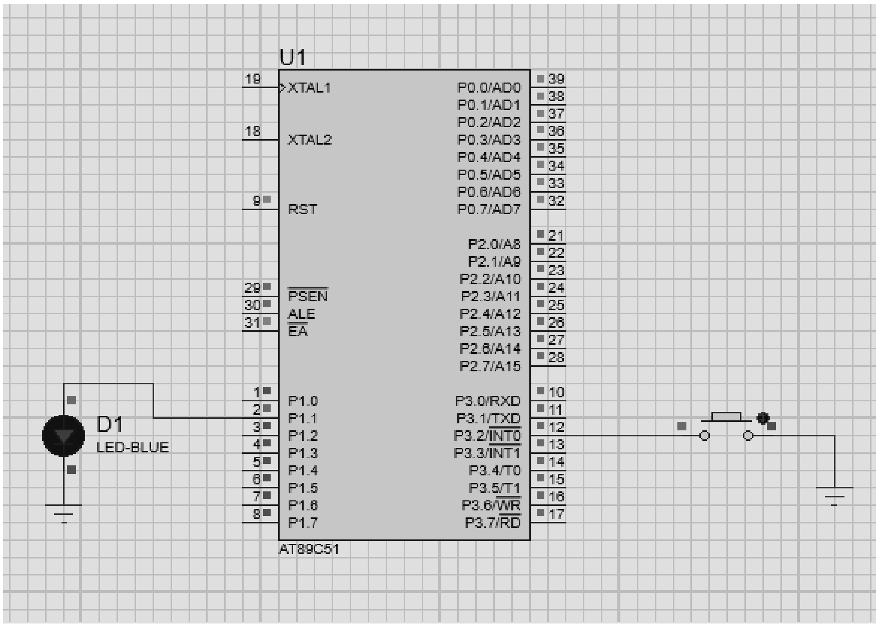


Fig 9.5 Simulation diagram

d) Actual circuit used in Laboratory

e) Actual Experimental set up used in laboratory

IX Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1.	Microcontroller kit	Single board system with 8K RAM,ROM memory with battery backup,16X4,16X2LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler,RS-232,USB, interfacing facility with built in power supply.	1 No.
2.	Desktop PC	Loaded with open source IDE, simulation and program downloading software.	1 No.

X Precautions to be followed

1. Use always current limiting resistor before LED connected to microcontroller

XI Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop assembly program using Integrated Development Environment (Keil IDE) or any other relevant software tool.
4. Debug program on IDE.
5. Execute program on IDE.
6. Create hex file for the program.
7. Download hex code in EPROM/Flash memory of microcontroller.
8. Interface LED to microcontroller as per circuit diagram shown in fig.
9. Observe the LED to glow when external interrupt occurs on P3.2.

SAMPLE PROGRAM 1: **:** Interface LED with microcontroller and turn it ON with microcontroller interrupt.

Step 1: Algorithm

Main Program

1. Initialize port P1 as output.
2. Initialize port P3 as input.
3. Clear pin P1.1
4. Enable interrupt INT0.
5. Wait for Interrupt.

INT0 ISR

1. Set pin P1.1
2. Return from ISR

Step 2: Flowchart

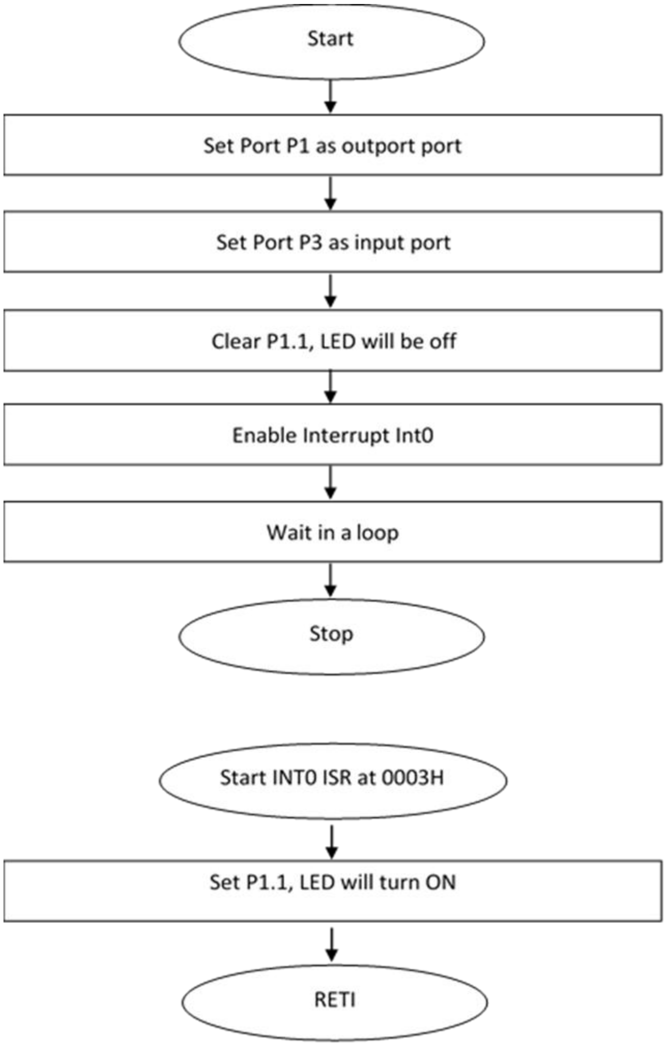

Start
Set Port P1 as outport port
Set Port P3 as input port
Clear P1.1, LED will be off
Enable Interrupt Int0
Wait in a loop
Stop
Start INT0 ISR at 0003H
Set P1.1, LED will turn ON
RETI

Fig 9.6 Flowchart to turn LED ON with microcontroller interrupt.

Step 3: Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments
			ORG 0000H	
C:0x0000	020006		LJMP START	
			ORG 0003H	;Interrupt service routine for INT0
C:0x0003	D291		SETB P1.1	
C:0x0005	32		RETI	
C:0x0006	759000	START:	MOV P1,#00H	;Main program for initialization
C:0x0009	75B0FF		MOV P3,#0FFH	
C:0x000C	75A881		MOV IE,#81H	;Enable hardware interrupt INT0
C:0x000F	80FE	HERE:	SJMP HERE	
			END	

Problem statement for student Interface two LEDs with microcontroller and turn them ON with microcontroller interrupts.

Step 1: Algorithm	**Step 2: Flowchart**

Step 3- Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments

XII Resources Used

S. No.	Instrument /Components	Specification	Quantity
1.			
2.			

XIII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....
....
....
....
....
....
....
....
....
....
....

XIV Precautions Followed (use blank sheet provided if space not sufficient)

....
....
....
....

XV Observations (use blank sheet provided if space not sufficient)

LED will become ________ (ON/OFF) after occurrence of_________ (INT0/INT1) Interrupt

XVI Result (Output of the Program)

....
....
....
....

XVII Interpretation of Results (Give meaning of the above obtained results)

....
....
....
....

XVIII Conclusions and Recommendation

....
....
....
....

XIX Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. With a single instruction, show how to disable all the interrupts.
2. List the interrupts of 8051 with vector address, port pin, flag affecting, priority level.

[Space for Answers]

....

....

....

....

....

....

XIX References / Suggestions for further reading

1. https://www.electronicshub.org › Free Project Circuits › Mini Projects
2. https://www.elprocus.com/led-interfacing-with-8051-microcontroller/
3. https://www.engineersgarage.com/microcontroller/.../interface-leds-AT89C51-circuit

XX Assessment Scheme

Performance indicators		Weightage
Process related(15 Marks)		**60% (15)**
1	Coding and Debugging ability	30%
2	Making connections of hardware	20%
3	Follow ethical practices.	10%
Product related (10 Marks)		**40%(10)**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition.	15%
6	Timely Submission of report, Answer to sample questions.	05%
	TOTAL	**100% (25)**

Name of Team Members

1 ………………………….....

2 ………………………......

3 …………………………..

4 …………………………..

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No. 10: Develop an ALP to generate pulse and square wave by using Timer delay.

I Practical Significance

In 8051 timers are used to generate delays or as counters to count events happening outside the microcontroller. In time required applications two available 16 bit timers are operated in different modes to generate specific delay. This practical will help the students to develop skills to program timers and generate delays.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems.
- **Engineering tools:** Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations.

III Competency and Practical Skills

This practical is expected to develop the following skills for the industry identified competency '**Maintain microcontroller based systems':**

- Select appropriate mode of timer for application.
- Program two 16 bit timers/counters independently.
- Generate required time delay.

IV Relevant Course Outcome(s)

- Interpret the program by using timer, interrupt and serial ports /parallel ports.

V Practical Outcome

- Develop an ALP to generate pulse and square wave by using Timer delay.

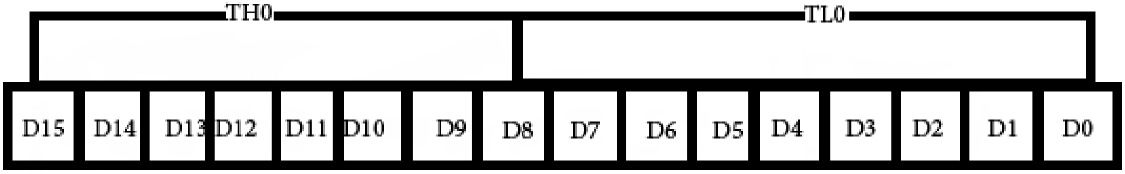
VI Relevant Affective domain related Outcome(s)

- Practice good housekeeping
- Demonstrate working as a leader/a team member.
- Follow ethical practices.

VII Minimum Theoretical Background

Four special function registers are connected with TIMER/COUNTER operation.
TMOD - Timer Mode Register
TCON - Timer Control Register
TH0, TL0 - Timer/Counter – 0
TH1, TL1 - Timer/Counter – 1

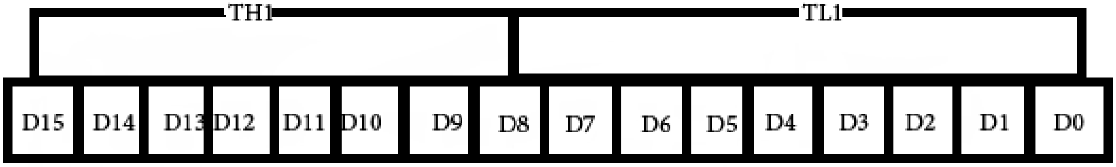
Timer0: 16 bit



TH0								TL0							
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Fig 10.1 Timer 0

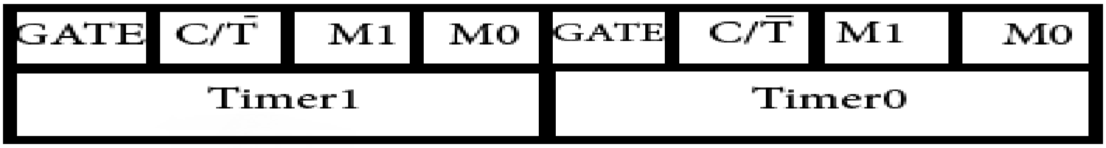
Timer1: 16 bit



TH1								TL1							
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Fig 10.2 Timer 1

TMOD (timer mode) Register



GATE	$C/\bar{T}$	M1	M0	GATE	$C/\bar{T}$	M1	M0
Timer1				Timer0			

Fig 10.3 TMOD Register

Gate: If bit is set Timer/counter is enable only while the INTx pin is high and the TRx control pin is set when cleared the timer is enabled whenever the TRx control bit is set
C/T': Timer or counter selected. Cleared for timer operation(input from internal system clock) Set for counter operation(input from Tx input pin)

M1 Mo:
0 0 – Mode 0 -13 bit timer mode
0 1 – Mode 1 -16 bit timer mode
1 0 – Mode 2 -8 bit auto reload
11 – Mode 3 –Split timer mode

Clock source for timer

Timer needs a clock pulse to tick. For the 8051 timers if C/T' = 0, the crystal frequency is attached to the 8051 is the source of the clock for timer. Timer's use 1/12 of XTAL frequency, regardless of machine cycle time

Find the timer's clock frequency and its period for 8051; If XTAL frequency = 11.0592 MHz
1/12 x 11.0592 MHz = 921.6 kHz;
T = 1/921.6 KHz = 1.085μs

TCON (timer control) Register: The timers are started by using instructions to set timer Start bits TR0 and TR1, which are called timer run control bits. When timer counts to its maximum value, it sets a flag TF0 or TF1

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Fig 10.4 TCON Register

Timer Mode 1:

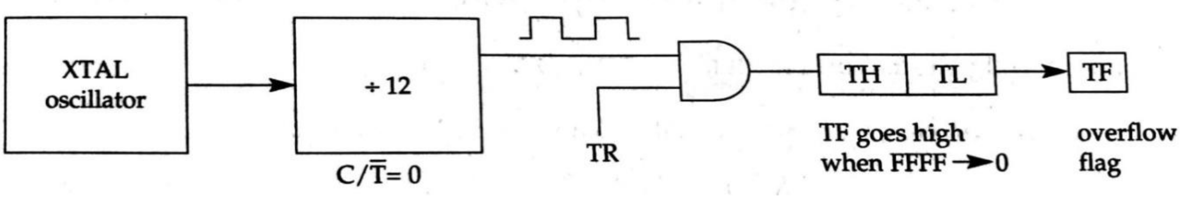


Fig 10.5 Timer mode 1 diagram

Timer delay calculation for mode 1 assuming XTAL = 11.0592 MHz

(FFFF – YYXX +1) x 1.085µs

Where YYXX are TH, TL initial values respectively. Notice that values YYXX are in hex.

Example: The number of counts for the rollover is FFFFH-FFF2H = 0DH (13 decimal). However, we add one to 13 because of the extra clock needed when it rolls over from FFFF to 0 and raises the TF flag. this gives 14x1.085µs = 30.38 µs gives us the time delay generated by the timer

VIII Practical Circuit diagram:

a) Sample Circuit diagram

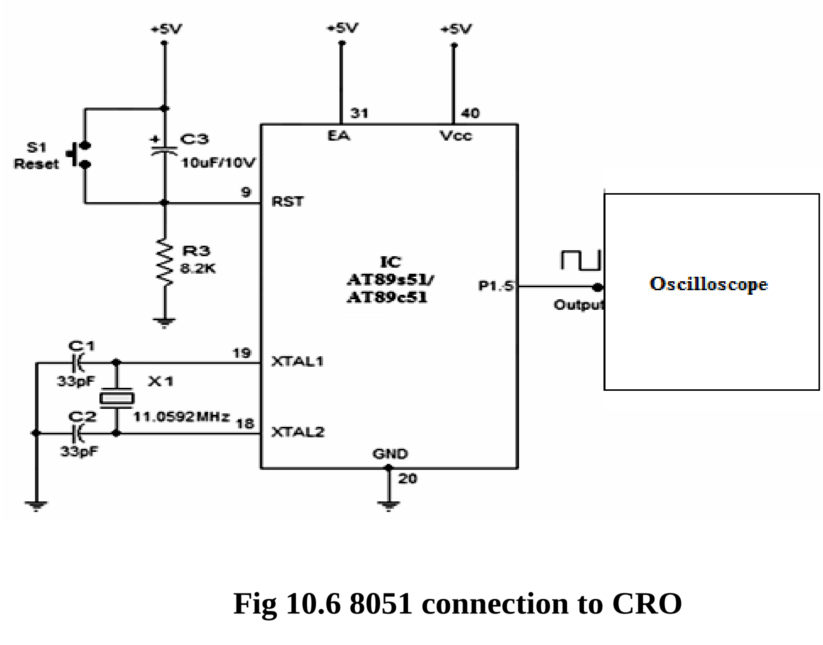


Fig 10.6 8051 connection to CRO

b) Simulation diagram

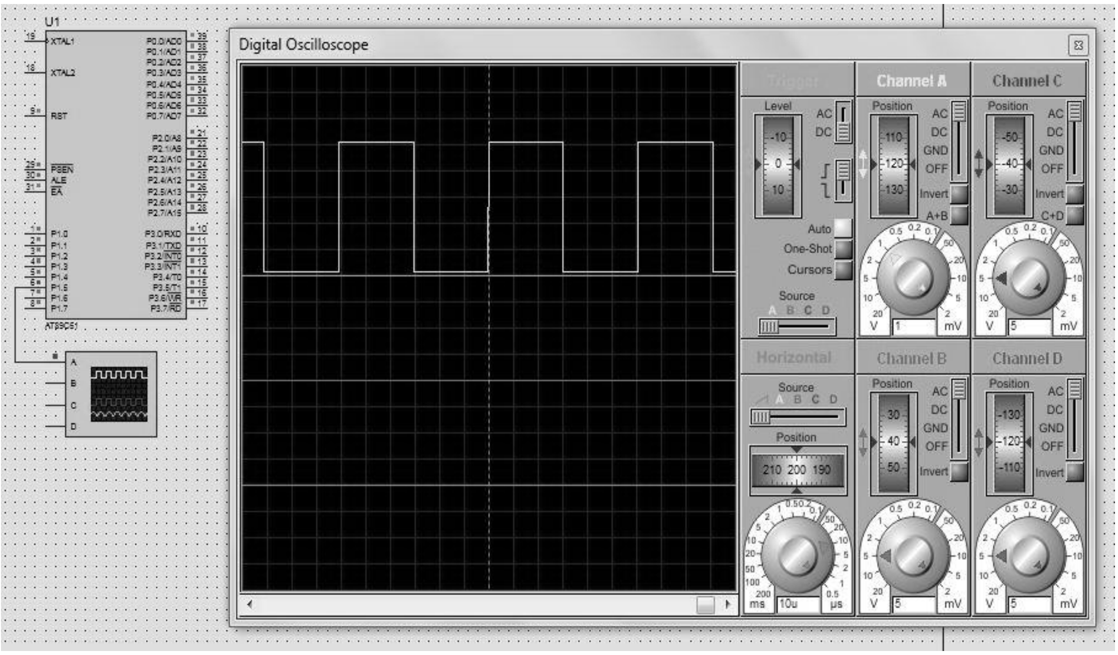


Fig 10.7 Simulation diagram

c) Actual circuit used in laboratory

IX Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1.	Microcontroller kit	Single board system with 8K RAM,ROM memory with battery backup,16X4,16X2LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler,RS-232,USB, interfacing facility with built in power supply.	1 No.
2.	Desktop PC	Loaded with open source IDE, simulation and program downloading software.	1 No.
3.	CRO	Bandwidth AC 10Hz ~ 20MHz (-3dB). DC ~ 20MHz (-3dB), X10 Probe	1 No.

X Precautions to be followed

1. Check rules/syntax of assembly programming.

XI Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop assembly program using Integrated Development Environment (Keil IDE) or any other relevant software tool.
4. Debug program on IDE.
5. Execute program on IDE.
6. Create hex file for the program.
7. Download hex code in EPROM/Flash memory of microcontroller.
8. Connect CRO probe to port pin and observe waveform.
9. Measure ON time and OFF time on CRO and draw the same in observation Table.

SAMPLE PROGRAM 1: To generate square wave by using Timer delay for count value FFF2H.

Step 1: Algorithm

1. Load the TMOD value register indicating which timer (Timer 0 or Timer 1) is to be used and which timer mode(0 or 1) is selected.
2. Load registers TL and TH with initial count values.
3. Start the timer.
4. Keep monitoring the timer flag (TF) with the “JNB TFx, target” instruction to see if it is raised. Get out of the loop when TF becomes high.
5. Stop the timer.
6. Clear the TF flag for the next round.
7. Go back to Step 2 to load TH and TL again.

Step 2: Flowchart

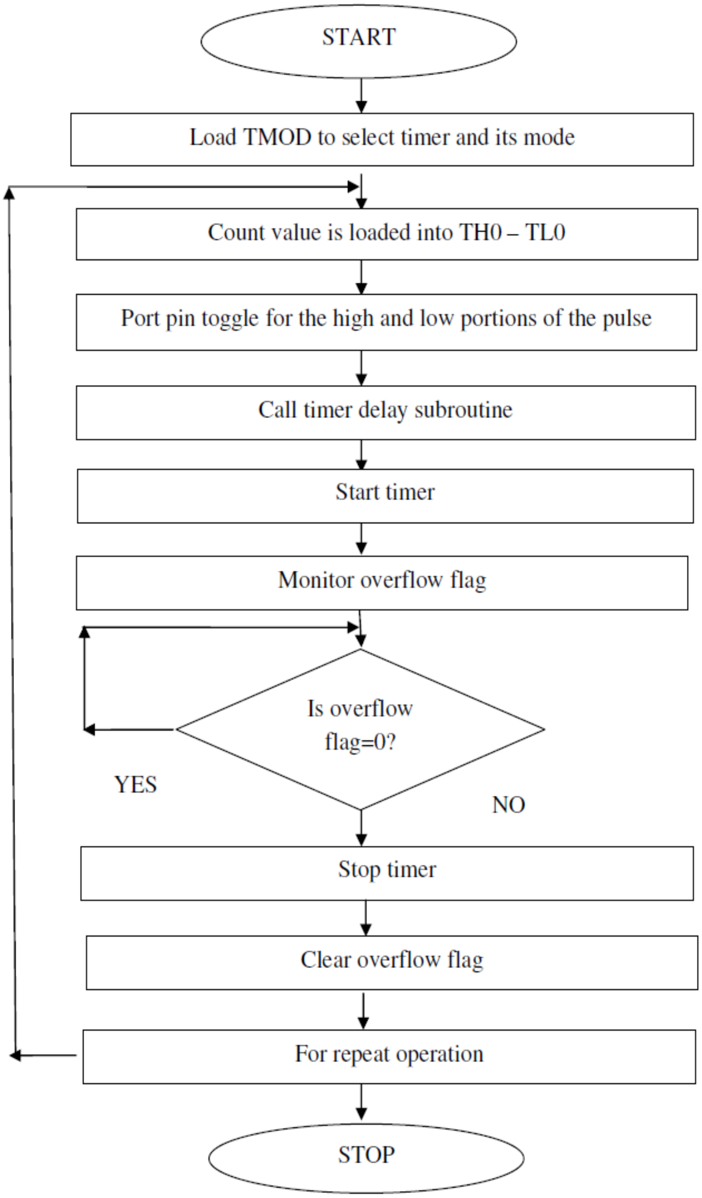

START
Load TMOD to select timer and its mode
Count value is loaded into TH0 – TL0
Port pin toggle for the high and low portions of the pulse
Call timer delay subroutine
Start timer
Monitor overflow flag
Is overflow flag=0?
YES
NO
Stop timer
Clear overflow flag
For repeat operation
STOP

Fig 10.8 Flowchart for square wave

Step 3: Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments
			ORG 0000h	
C:0x0000	758901		MOV TMOD ,#01H	;Timer 0, mode 1
C:0x0003	758AF2	HERE:	MOV TL0,#0F2H	;TL0=F2H
C:0x0006	758CFF		MOV TH0,#0FFH	;TH0=FFH
C:0x0009	B295		CPL P1.5	;toggle P1.5
C:0x000B	110F		ACALL DELAY	
C:0x000D	80F4		SJMP HERE	;load TH, TL again
C:0x000F	D28C	DELAY:	SETB TR0	;start Timer 0
C:0x0011	308DFD	AGAIN:	JNB TF0,AGAIN	;monitor Timer 0 overflow flag until it rolls over
C:0x0014	C28C		CLR TR0	;stop Timer 0
C:0x0016	C28D		CLR TF0	;clear Timer 0 flag
C:0x0018	22		RET	
			END	

Problem statement for student: Develop assembly program to generate pulse width of 5ms on P2.3 using timer 0 mode1. Assume XTAL = 11.0592MHz

Step 1: Algorithm	Step 2: Flowchart

Step 3- Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments

XII Resources Used

Sr. No.	Instrument /Components	Specification	Quantity
1.			
2.			
3.			

XIII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XIV Precautions Followed (use blank sheet provided if space not sufficient)

....

....

....

....

XV Observations for sample program (use blank sheet provided if space not sufficient)

Draw Square wave and show TON and TOFF time

XVI Result (Output of the Program)

....

....

....

....

XVII Interpretation of Results (Give meaning of the above obtained results)

....

....

....

....

XVIII Conclusions and Recommendation

....

....

....

....

XIX Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. Calculate the number to be loaded into TH, TL to generate square wave of ON time = 3ms and OFF time = 10ms use mode 1. Assume XTAL of 22MHz.
2. To get a 2 ms delay, what number should be loaded into TH, TL using mode 1. Assume XTAL= 11.0592MHz.

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XIX References / Suggestions for further reading

1. http://nptel.ac.in/courses/Webcourse-contents/IIT-KANPUR/microcontrollers/micro/ui/Course_home2_8.htm
2. http://www.circuitstoday.com/8051-timers-counters.
3. The 8051 Microcontroller and Embedded system Using Assembly and C- Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. Mckinlay- Pearson /Prentice Hall, , 2nd edition, Delhi,2008, ISBN 978-8177589030.

XX Assessment Scheme

Performance Indicators		Weightage
Process related(15 Marks)		**60%**
1	Coding and Debugging ability	30%
2	Making connections of hardware	20%
3	Follow ethical practices.	10%
Product related (10 Marks)		**40%**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition.	15%
6	Timely Submission of report, Answer to sample questions.	05%
Total (25 Marks)		**100%**

Name of Team Members

1 …………………………......

2 ………………………….......

3 …………………………..

4 …………………………..

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No. 11: Interface 4 X 4 LED matrix with 8051 to display various pattern.

I Practical Significance

LED matrix displays are used as stadium displays, decorative displays and as visual signals to human eye, to convey a message or meaning .LED matrix displays are interfaced with microcontroller I/O port to display characters and different patterns. This practical will help the students to develop skills to interface 4 X 4 LED matrix display to microcontroller and display various pattern.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems
- **Engineering tools:** Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations.

III Competency and Practical Skills

This practical is expected to develop the following skills for the industry identified competency '**Maintain microcontroller based systems':**

- Select appropriate type of LED grid for given application.
- Interface 4x4 LED matrix display to the microcontroller.
- Generate different patterns on LED matrix.

IV Relevant Course Outcome(s).

- Interface the memory and I/O devices to 8051 microcontroller.

V Practical Outcome

- Interface 4 X 4 LED matrix with 8051 to display various pattern.

VI Relevant Affective domain related Outcome(s)

- Practice good housekeeping.
- Demonstrate working as a leader/a team member.
- Follow ethical practices.

VII Minimum Theoretical Background

In LED dot matrix display the LEDs are connected at the column and row intersections of the matrix. LEDs in the same row are connected together and LEDs in the column are connected together. Transistors are act as switches and used to control LEDs in the matrix.

LED (Light Emitting Diode)

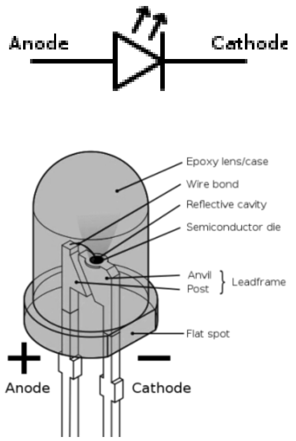


Fig 11.1 LED symbol and construction

- **Specifications: LED**
  1. Current: 20 mA
  2. Voltage drop: 1.9 to 2.1 V
  3. Power dissipation: 40 mW
  4. Color: RED

- **Operation:** Voltage +5V to anode with respect to cathode ground LED will turn ON.

Accessing individual LEDs: You turn on an individual LED by setting its row and column pins to the proper logic 1. For example, referring to figure 11.2 LED matrix the switch in column 2 is closed which ties the anodes of all of the LEDs in that column to positive voltage and on the left the switch in row 1 is closed causing a ground level to be applied to the cathode of all of the LEDs in that row. The LED at the intersection of column 2 and row 1 is forward biased and turns on.

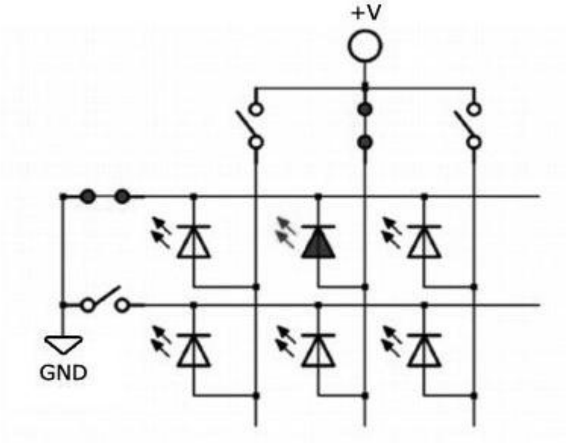


Fig 11.2 Addressing individual LED: 2nd column and 1st row

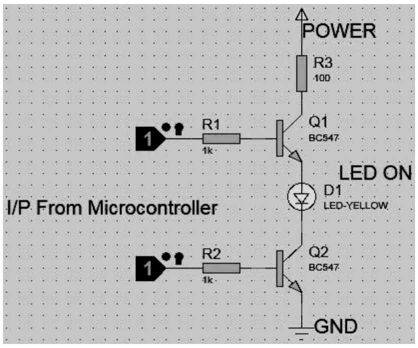


Fig 11.3 LED ON logic through transistor

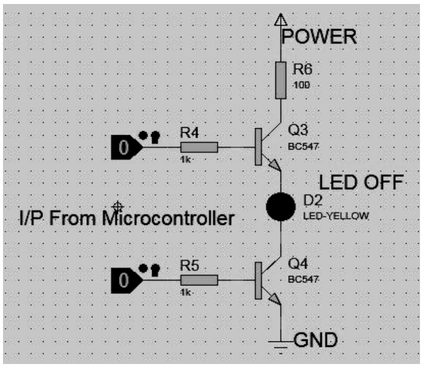


Fig 11.4 LED OFF logic through transistor

VIII Practical Circuit diagram:

a) Sample circuit diagram

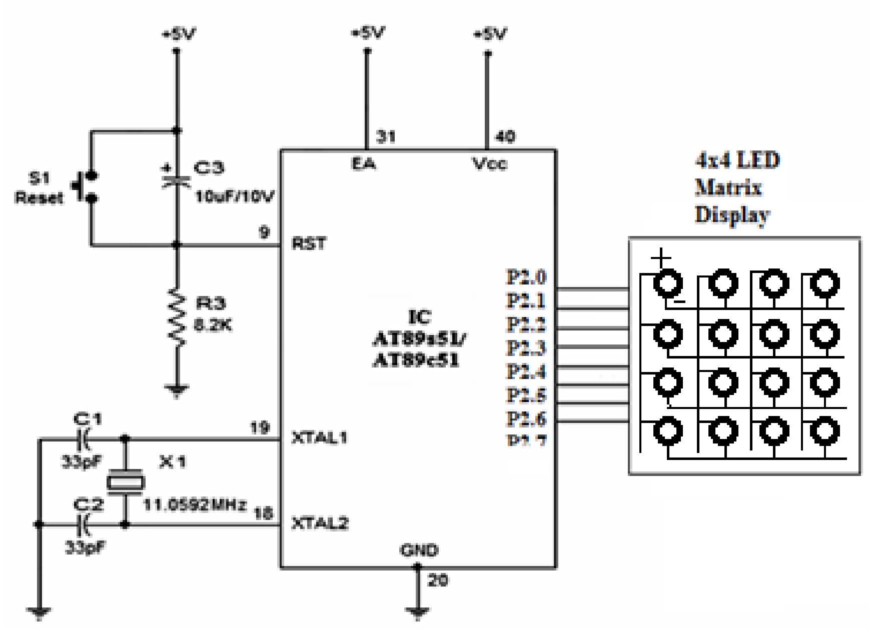


Fig 11.5 8051 connection to 4x4 LED matrix display

b) Simulation diagram

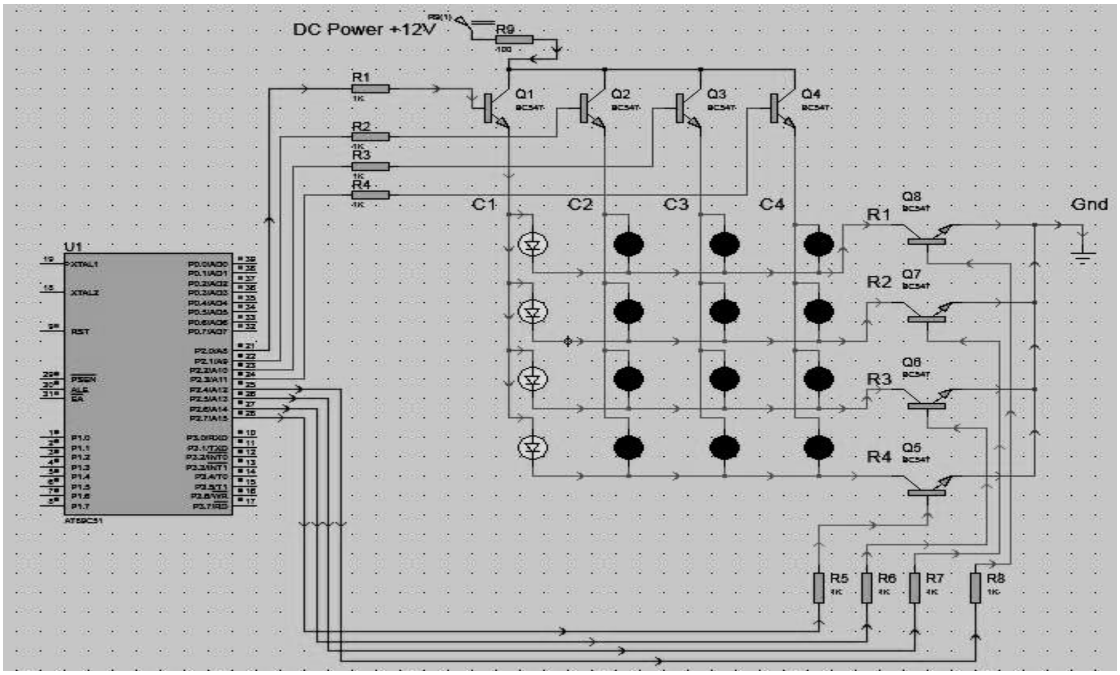


Fig 11.6 Simulation diagram

c) Actual circuit used in laboratory

IX Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1	Microcontroller kit	Single board system with 8K RAM,ROM memory with battery backup,16X4,16X2LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler,RS-232,USB, interfacing facility with built in power supply.	1 No.
2	Desktop PC	Loaded with open source IDE, simulation and program downloading software.	1 No.
3	4 X4 LED matrix	Suitable to interface with 8051 trainer kit	1 No

X Precautions to be Followed

1. Always use driver circuit to interface LED matrix to microcontroller.

XI Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop assembly program using Integrated Development Environment (Keil IDE) or any other relevant software tool.
4. Debug program on IDE.
5. Simulate program on IDE.
6. Create hex file for the program.
7. Download hex code in EPROM/Flash memory of the microcontroller.
8. Interface4 X 4 LED matrix to 8051.
9. Observe different patterns on 4x4 LED matrix

SAMPLE PROGRAM 1: Write program to display various patterns on 4x4 LED matrix.

Step 1: Algorithm

1. Make port P2 used to interface 4x4 LED matrix as an output port.
2. Send F1H value to the port to turn ON first column all LED.
3. Add delay.
4. Send 1FH value to the port to display first row all LED.
5. Add delay.
6. Send 11H value to the port to display first column first row LED.
7. Add delay.
8. Send 88H value to the port to display fourth column fourth row LED.
9. Add delay.
10. For repeat operation go to step 2.

Step 2: Flowchart

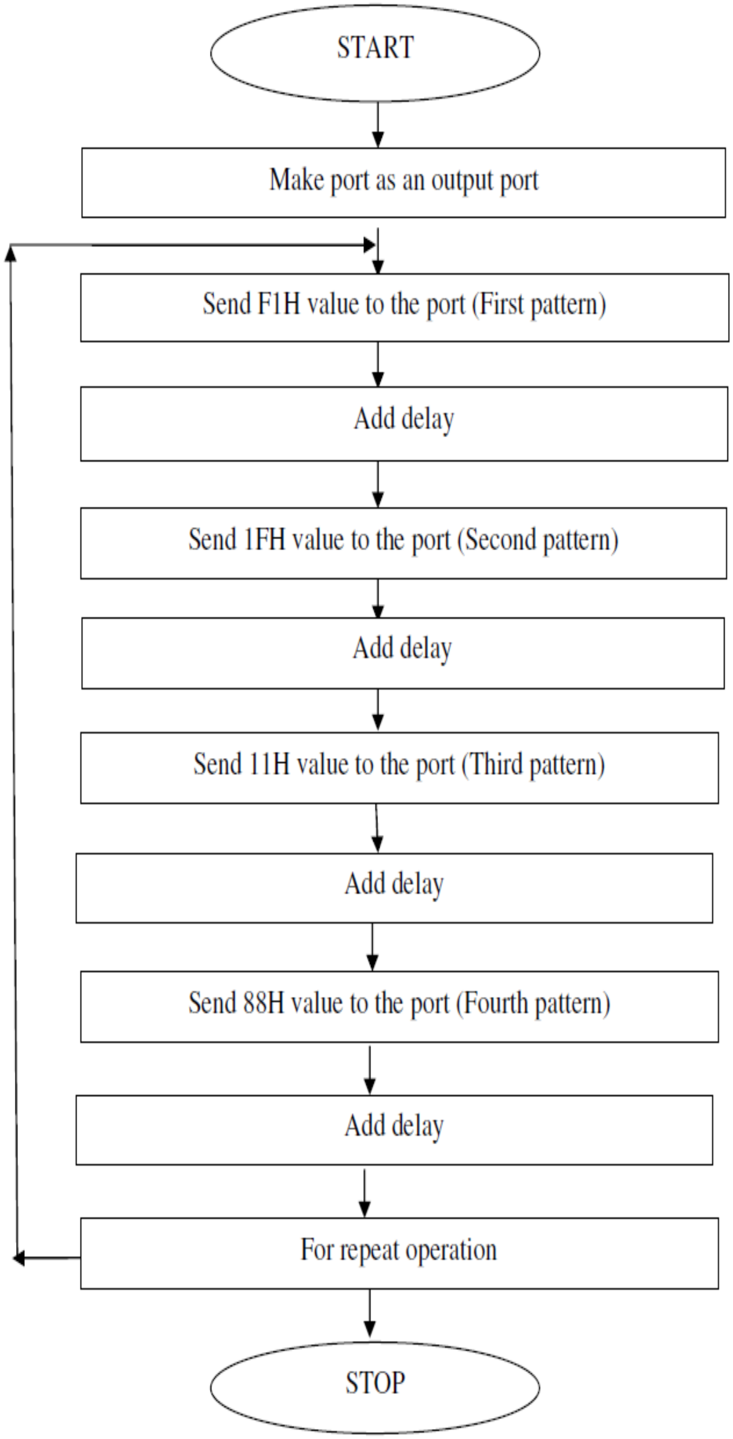

START
Make port as an output port
Send F1H value to the port (First pattern)
Add delay
Send 1FH value to the port (Second pattern)
Add delay
Send 11H value to the port (Third pattern)
Add delay
Send 88H value to the port (Fourth pattern)
Add delay
For repeat operation
STOP

Fig . 11.2 Flowchart for 4x4 LED matrix

Step 3: Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments
			ORG 0000H	
C:0x0000	75A000		MOV P2,#00H	;make port as output
C:0x0003	75A0F1		MOV P2,#0F1H	;send value to LED matrix
C:0x0006	1119	RPT:	ACALL DELAY	;add delay
C:0x0008	75A01F		MOV P2,#1FH	;send value to LED matrix
C:0x000B	1119		ACALL DELAY	;add delay
C:0x000D	75A011		MOV P2,#11H	;send value to LED matrix
C:0x0010	1119		ACALL DELAY	;add delay
C:0x0012	75A088		MOV P2,#88H	;send value to LED matrix
C:0x0015	1119		ACALL DELAY	;add delay
C:0x0017	80EA		SJMP RPT	
C:0x0019	7A0A		MOV R2,#10	;delay subroutine
C:0x001B	7B64	DELAY:	MOV R3,#100	
C:0x001D	7CC8	HERE2:	MOV R4,#200	
C:0x001F	DCFE	HERE1:	DJNZ R4,HERE	
C:0x0021	BDFA	HERE:	DJNZ R3,HERE1	
C:0x0023	DAF6		DJNZ R2,HERE2	
C:0x0025	22		RET	
			END	

Problem statement for student: Develop assembly program to turn ON and OFF all LEDs connected to port 1 with some delay.

Step 1: Algorithm	Step 2-Flowchart

Step 3- Assembly Language Program Code

Memory Address	Hex Code	Label	Mnemonics	Comments

XII Resources Used

Sr. No.	Name of Resource.	Specification	Quantity
1.			
2.			
3.			

XIII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....

....

....

....

....

....

....

....

....

....

....

XIV Precautions Followed (use blank sheet provided if space not sufficient)

....

....

....

....

....

XV Observations for sample program (use blank sheet provided if space not sufficient)

HEX Value	**LED status (ON/OFF)**	
	Column	**Row**
F1H		
1FH		
11H		
88H		

XVI Results (Output of the Program)

....

....

....

....

XVII Interpretation of Results (Give meaning of the above obtained results)

....

....

....

....

XVIII Conclusions and Recommendation (Actions/decisions to be taken based on the interpretation of results).

....

....

....

....

XIX Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Give syntax of two instructions used in program to exchange data.
2. Write and execute a program to glow the second line display on 4x4 LED matrix display continuously.

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XX References / Suggestions for further reading

1. https://www.youtube.com/watch?v=-1UPPHjR0vk
2. https://www.youtube.com/watch?v=MmCGDJ90Qt4
3. https://www.youtube.com/watch?v=lZyc6ulpkyM

XXI Assessment Scheme

Performance indicators		Weightage
Process related(15 Marks)		**60% (15)**
1	Coding and Debugging ability	30%
2	Making connections of hardware	20%
3	Follow ethical practices.	10%
Product related (10 Marks)		**40%(10)**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition.	15%
6	Timely Submission of report, Answer to sample questions.	05%
	TOTAL	**100% (25)**

Name of Team Members

1 …………………………......

2 ………………………….......

3 …………………………..

4 …………………………..

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No. 12: Interface 7-segment display to display the decimal number from 0 to 9.

I Practical Significance

In electronic displays like pricing menu at petrol pump, in metros, digital clocks and in many electronics appliances most commonly used display device is a 7-segment display (SSD). Seven segment displays are used to display decimal numbers from 0 to 9 and in some cases, basic characters also. This practical will help the students to develop skills to interface 7-segment display to microcontroller and display number from 0 to 9.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems.
- **Engineering tools:** Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations.

III Competency and Practical Skills

This practical is expected to develop the following skills for the industry identified competency '**Maintain microcontroller based systems':**

- Select appropriate type of 7- segment display for given application.
- Interface 7-segment display to the microcontroller in static or multiplexed mode.
- Display numbers and characters on 7 segment display.

IV Relevant Course Outcome(s)

- Interface the memory and I/O devices to 8051 microcontroller.

V Practical Outcome

- Interface 7-segment display to display the decimal number from 0 to 9.

VI Relevant Affective domain related Outcome(s)

- Follow safe practices
- Maintain tools and equipment.
- Follow ethical practices

VII Minimum Theoretical Background

7-segment display may use a light-emitting diode (LED) for each segment, or other light-generating or controlling techniques such as cold cathode gas discharge, vacuum fluorescent, incandescent filaments, and other. Types of LED based 7-segment display are i) Common Cathode Display ii) Common Anode. Display

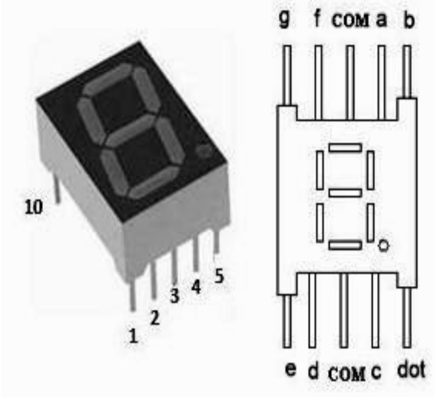


Fig 12.1 Seven segment display and pin configuration

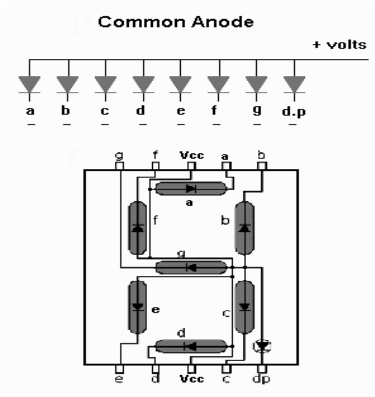


Fig 12.2 CA 7-segment display constructional diagram

Specifications: 7-segment display

- Part No: LDS-A516RI
- Digit/Alpha: Size:0.56" RED
- Voltage drop across per LED:2.2V
- Max current per LED: 10mA~20mA
- Peak wavelength :660nm

LED segment is ON for logic ‘0’ (Ground) and OFF for logic ‘1’ (+5V).

Table 12.1 CA type SSD Operation

	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	
Digit	**DP**	**G**	**F**	**E**	**D**	**C**	**B**	**A**	**Hex code**
0	1	1	0	0	0	0	0	0	0C0 H
1	1	1	1	1	1	0	0	1	0F9 H
2	1	0	1	0	0	1	0	0	0A4 H
3	1	0	1	1	0	0	0	0	0B0 H
4	1	0	0	1	1	0	0	1	99 H
5	1	0	0	1	0	0	1	0	92 H
6	1	0	0	0	0	0	1	0	82 H
7	1	1	1	1	1	0	0	0	0F8 H
8	1	0	0	0	0	0	0	0	80 H
9	1	0	0	1	0	0	0	0	90 H

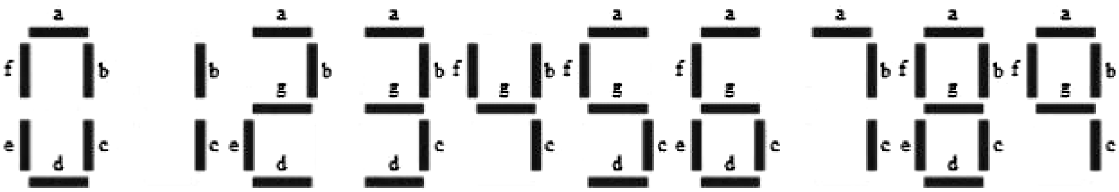


Fig 12.3 Seven segment display number patterns

VIII Practical Circuit diagram:

a) Sample Circuit diagram

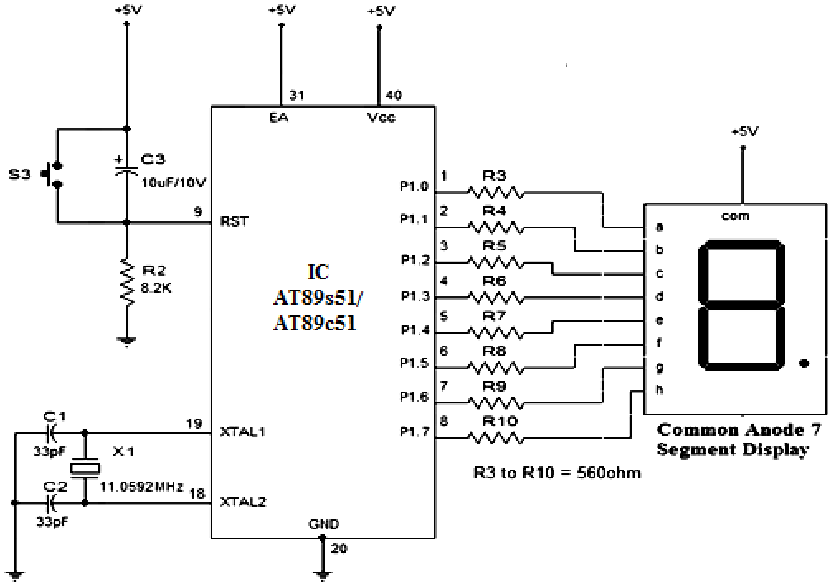


Fig 12.4 8051 connection to CA seven segment display

b) Practical Setup

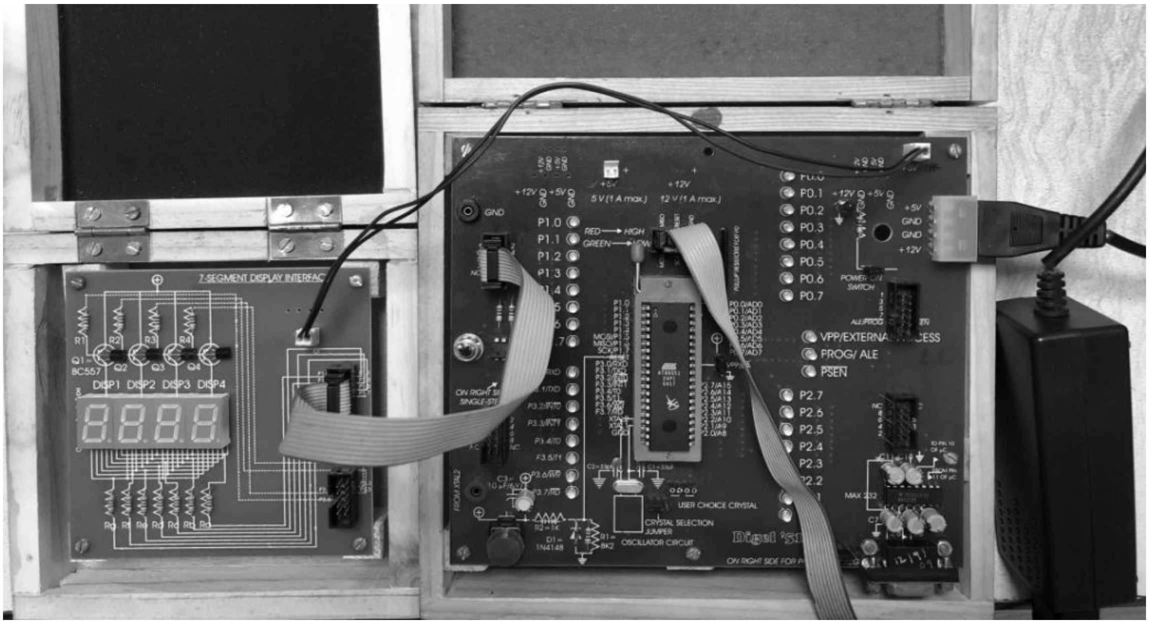


Fig 12.5 Practical setup

c) **Simulation diagram**

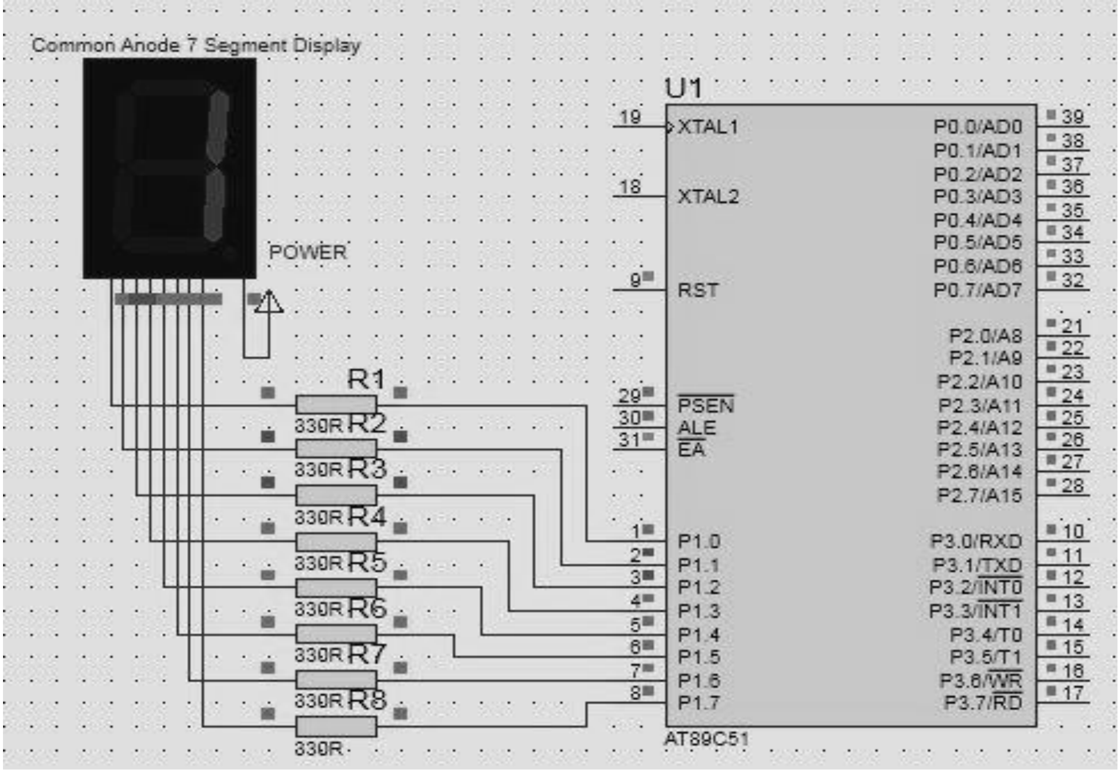


Fig 12.6 Simulation diagram

d) **Actual circuit used in Laboratory**

e) Actual Experimental set up used in laboratory

IX Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1.	Microcontroller kit	Single board system with 8K RAM,ROM memory with battery backup,16X4,16X2LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler,RS-232,USB, interfacing facility with built in power supply.	1 No.
2.	Desktop PC	Loaded with open source IDE, simulation and program downloading software.	1 No.
3.	7-segment LED Display	0.56 in 1-digit Red, common anode/common cathode display.	1No.

X Precautions to be followed

1. Use always current limiting resistor before interfacing 7-segment display to microcontroller.
2. For safe operation use seven segment displays at 25° temperature.

XI Procedure

1. Write algorithm for given problem.
2. Draw flowchart.
3. Develop assembly program using Integrated Development Environment (Keil IDE) or any other relevant software tool.
4. Debug program on IDE.
5. Execute program on IDE.
6. Create hex file.
7. Download hex code in EPROM/Flash memory of microcontroller.
8. Interface CA type 7 segment display to microcontroller as per circuit diagram shown in fig 12.4.
9. Observe and draw the display of numbers on 7-seven segment display.
10. Record the hex value in observation table.

SAMPLE PROGRAM 1: Write a program to display decimal no 0 to 9.

Step 1: Algorithm

1. Make the Port P1 used to Interface 7-segment display as an output port.
2. Set counter register R2 =10 for 0 to 9 digits.
3. Load DPTR with memory address where table is stored.
4. Clear Accumulator.
5. Read stored hex code of decimal digit from memory into Accumulator.
6. Send code to output port where 7-segment display is connected.
7. Increment memory pointer i.e. DPTR.
8. Decrement the counter register R2 and compare with 0 is counter =0? NO- go to step 4 to send next digit code.
9. For repeat operation go to step 2.
10. Stop

Step 2: Flowchart

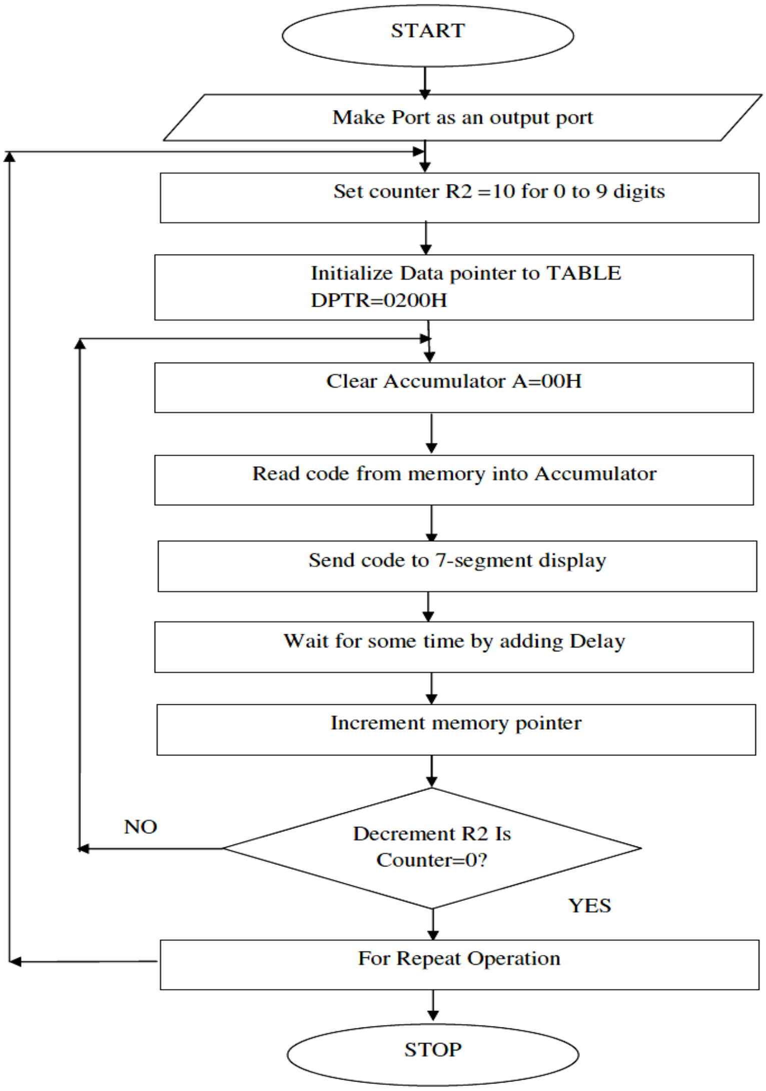

START
Make Port as an output port
Set counter R2 =10 for 0 to 9 digits
Initialize Data pointer to TABLE
DPTR=0200H
Clear Accumulator A=00H
Read code from memory into Accumulator
Send code to 7-segment display
Wait for some time by adding Delay
Increment memory pointer
NO
Decrement R2 Is
Counter=0?
YES
For Repeat Operation
STOP

Fig 12.7 Flowchart to display decimal no 0 to 9

Step 3: Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments
			ORG 0000H	
C:0x0000	759000		MOV P1,#00H	;Make port as output port
C:0x0003	7A0A	REPEAT:	MOV R2,#10	;Set register as counter of 10 bytes
C:0x0005	900200		MOV DPTR,#TABLE	;Load address of memory into Data pointer
C:0x0008	E4	UP:	CLR A	;Clear accumulator
C:0x0009	93		MOVC A,@A+DPTR	;Read hex code from memory into accumulator
C:0x000A	F590		MOV P1,A	;Send hex code to port

C:0x000C	1113		ACALL DELAY	
C:0x000E	A3		INC DPTR	;Increment memory pointer to read next digit hex code
C:0x000F	DAF7		DJNZ R2,UP	; Decrement counter & jump if not equal to zero to label UP.
C:0x0011	80F0		SJMP REPEAT	;Repeat loop
C:0x0013	7B19	DELAY:	MOV R3,#25	;Delay Subroutine
C:0x0015	7C64	L3:	MOV R4,#100	
C:0x0017	7D64	L2:	MOV R5,#100	
C:0x0019	DDFE	L1:	DJNZ R5,L1	
C:0x001B	DCFA		DJNZ R4,L2	
C:0x001D	DBF6		DJNZ R3,L3	
C:0x001F	22		RET	
			ORG 0200H	
		TABLE:	DB 0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0F8H, 80H, 90H	;Decimal 0 to 9 hex code stored at code memory starting at location 0200H onward
			END	

Problem statement for student: Develop assembly program to display Alphabets A, B, C, D, E and F on Common Anode 7-segment display

Step 1: Algorithm	**Step 2: Flowchart**

Step 3- Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments

XII Resources Used

S. No.	Instrument /Components	Specification	Quantity
1.			
2.			
3.			

XIII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XIV Precautions Followed (use blank sheet provided if space not sufficient)

....

....

....

....

XV Observations for sample program (use blank sheet provided if space not sufficient)

Sr. NO.	Memory Location	Hex Value
1	C:0x0200	
2	C:0x0201	
3	C:0x0202	
4	C:0x0203	
5	C:0x0204	
6	C:0x0205	
7	C:0x0206	
8	C:0x0207	
9	C:0x0208	
10	C:0x0209	

XVI Result (Output of the Program)

....
....
....

XVII Interpretation of Results (Give meaning of the above obtained results)

....
....
....
....

XVIII Conclusions and Recommendation

....
....
....
....

XIX Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. List the numbers for CC type display in which segment ‘a’ is activated.
2. Write Steps for testing seven segment displays.
3. Write and execute a program to display continuously ‘0 to 9’ if switch is pressed else ‘9 to 0’ on 7 segment display

[Space for Answers]

....
....
....
....
....
....
....
....
....
....
....
....
....
....

……

……

……

XX References / Suggestions for further reading

1. https://en.wikipedia.org/wiki/Seven-segment_display
2. https://pdf1.alldatasheet.com/datasheet-pdf/view/162375/ETC1/LDS-A516RI.html.
3. https://www.youtube.com/watch?v=1elRGva9bNg.

XXI Assessment Scheme

Performance Indicators		Weightage
Process related(15 Marks)		**60%**
1	Coding and Debugging ability	30%
2	Making connections of hardware	20%
3	Follow ethical practices.	10%
Product related (10 Marks)		**40%**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition.	15%
6	Timely Submission of report, Answer to sample questions.	05%
Total (25 Marks)		**100%**

Name of Team Members

1 …………………………......

2 ………………………….......

3 …………………………..

4 …………………………..

.

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No. 13: Interface relay with microcontroller and turn it ON and OFF

I Practical Significance

In Industrial applications low power devices microcontrollers drive relays are use to control electrical loads beyond their direct drive capability. Relays are used wherever it is necessary to control a high power or high voltage circuit with a low power circuit, especially when galvanic isolation is desirable. Electromechanical protective relays are used to detect overload and other faults on electrical lines by opening and closing circuit breakers. This practical will help the students to develop skills to interface relay to microcontroller and turn it ON and OFF.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems.
- **Engineering tools:** Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations

III Competency and Practical Skills

This practical is expected to develop the following skills for the industry identified competency '**Maintain microcontroller based systems':**

- Select appropriate type of relay for given application.
- Interface relay to the microcontroller.
- Use relay to switch on and off electrical loads.

IV Relevant Course Outcome(s)

- Interface the memory and I/O devices to 8051 microcontroller.

V Practical Outcome

- Interface relay with microcontroller and turn it ON and OFF.

VI Relevant Affective domain related Outcome(s)

- Follow safe practices.
- Demonstrate working as a leader/a team member
- Maintain tools and equipment..

VII Minimum Theoretical Background

When an electric current is passed through the coil it generates a magnetic field that activates the armature and the consequent movement of the movable contact either makes or breaks (depending upon construction) a connection with a fixed contact.

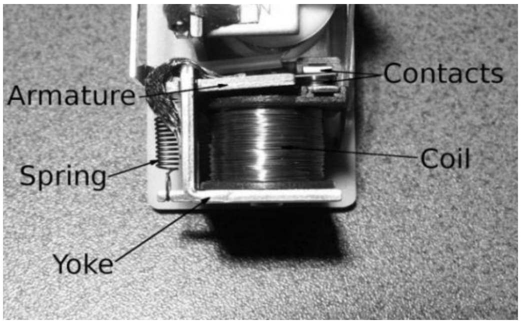


Fig 13.1 Simple electromechanical relay

Specifications: SPDT Relay

- Voltage: 12V to 15V
- Coil resistance: 400 ohm
- Current: 30mA

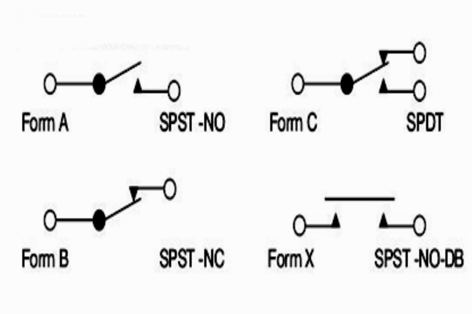


Fig 13.2 Contact forms

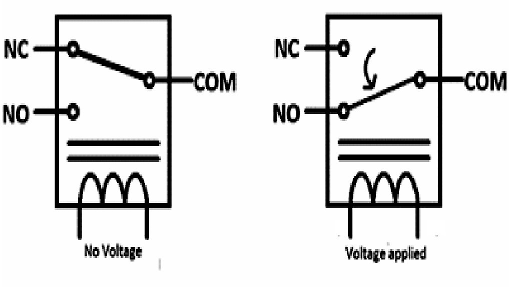


Fig 13.3 Form C SPDT Relay

VIII Practical Circuit diagram:

a) Sample circuit diagram

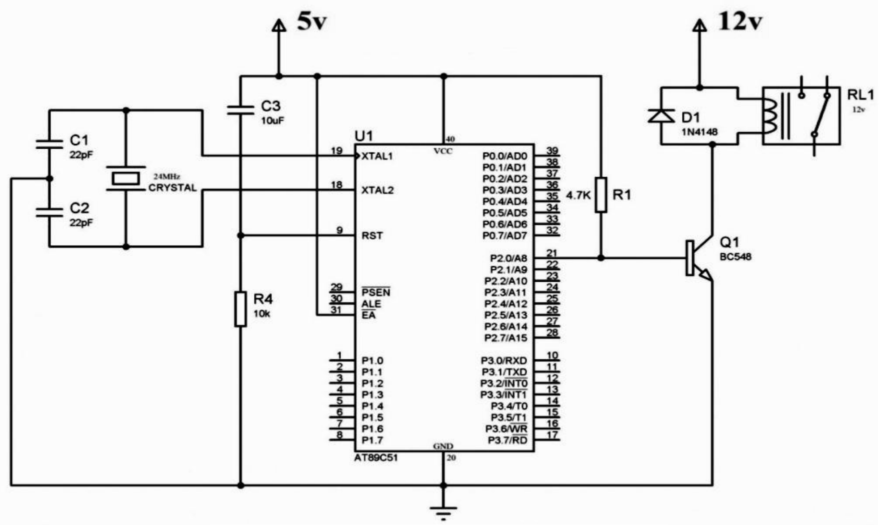


Fig 13.4 8051 connection to Relay

b) Practical setup

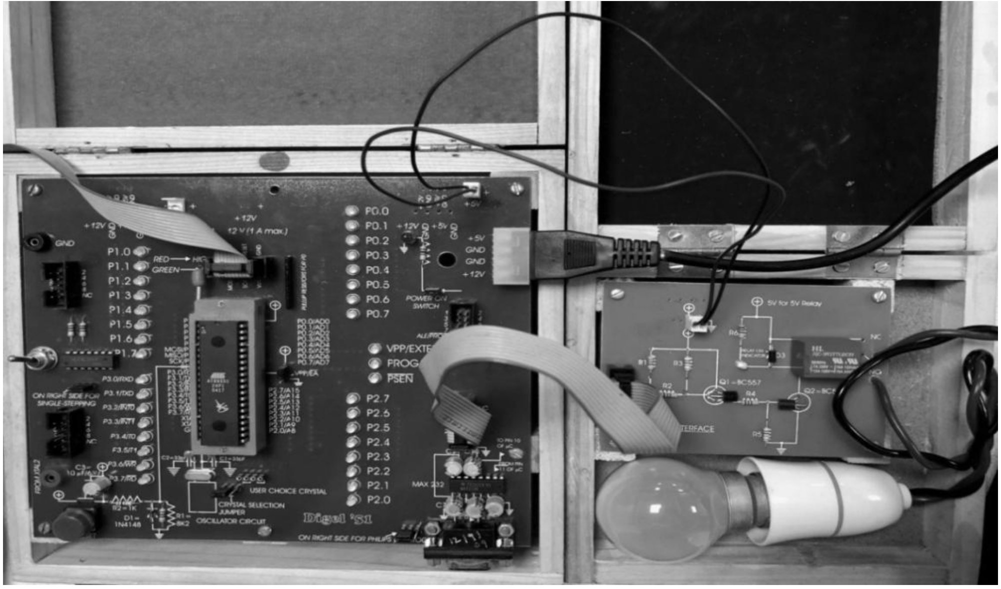


Fig 13.5 Practical Setup

c) Simulation diagram

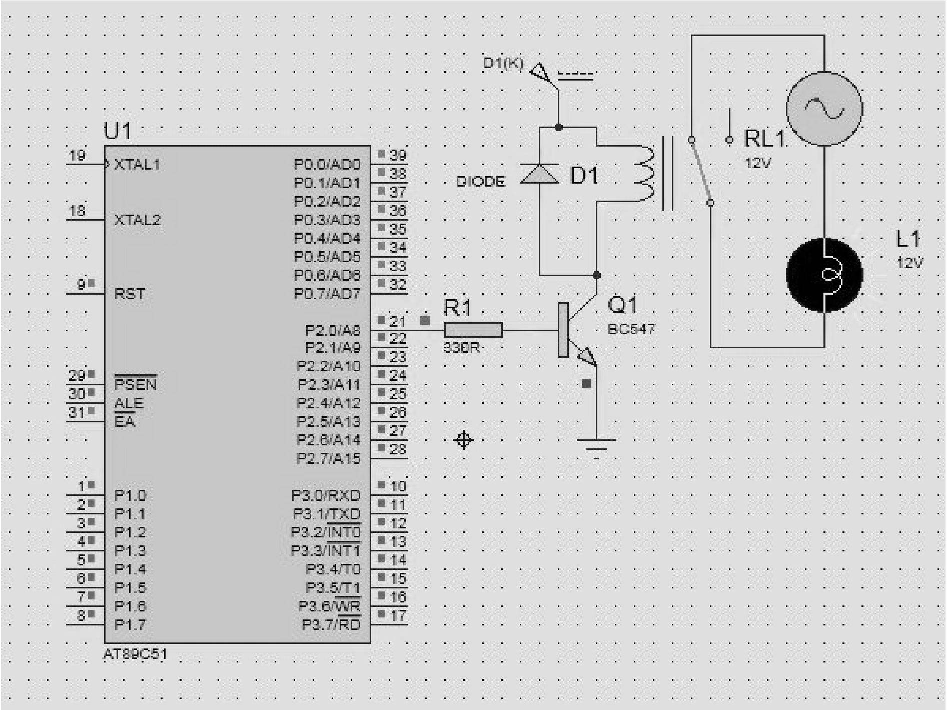


Fig 13.6 Simulation diagram

d) Actual circuit used in Laboratory

e) Actual Experimental set up used in laboratory

XI Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1.	Microcontroller kit	Single board systems with 8K RAM,ROM memory with battery backup,16X4,16X2LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler,RS-232,USB, interfacing facility with built in power supply.	1 No.
2.	Desktop PC	Loaded with open source IDE, simulation & program downloading software	1 No.
3.	Relay trainer board	Suitable to interface with 8051 trainer kit	1 No.

X Precautions to be Followed

1. Use always driver circuit before interfacing relay to the microcontroller.
2. Use fly back diode to avoid voltage spikes.

XI Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop assembly program using Integrated Development Environment (IDE) or any other relevant software tool.
4. Debug program on IDE.
5. Execute program on IDE.
6. Create hex file for the above program.
7. Download hex code in EPROM/Flash memory of microcontroller.
8. Interface relay to the microcontroller as per circuit diagram shown in Fig 13.4.
9. Observe the relay operation by connecting electrical load and note it in observation Table 13.3

SAMPLE PROGRAM 1: Write a program to turn ON/OFF relay.

Step 1-Algorithm

1. Make the Port pin P2.0 used to Interface relay as an output pin.
2. Turn on relay by setting port bit.
3. Add delay
4. Turn off relay by clearing bit
5. Add delay
6. For repeat operation go to step 2
7. Stop

Step 2-Flow Chart

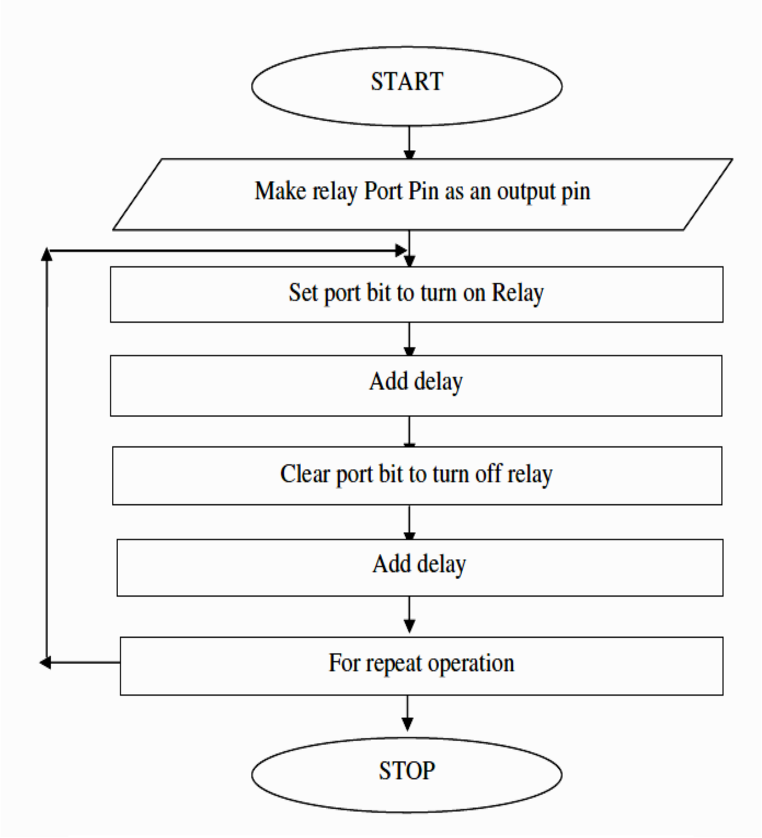

START
Make relay Port Pin as an output pin
Set port bit to turn on Relay
Add delay
Clear port bit to turn off relay
Add delay
For repeat operation
STOP

Fig 13.7 Flowchart to turn ON and OFF relay

Step 3- Assembly Language Sample Program

Memory Address	Hex Code	Label	Mnemonics	Comments
			ORG 0000H	
C:0x0000	D2A0	MAIN:	SETB P2.0	;relay on
C:0x0002	110A		ACALL DELAY	
C:0x0004	C2A0		CLR P2.0	;relay off
C:0x0006	110A		ACALL DELAY	
C:0x0008	80F6		SJMP MAIN	
C:0x000A	7B64	DELAY:	MOV R3,#100	;Delay Subroutine
C:0x000C	7CFF	L3:	MOV R4,#255	
C:0x000E	7DFF	L2:	MOV R5,#255	
C:0x0010	DDFE	L1:	DJNZ R5,L1	
C:0x0012	DCFA		DJNZ R4,L2	
C:0x0014	DBF6		DJNZ R3,L3	
C:0x0016	22		RET	
			END	

Problem statement for student: Develop assembly program to turn ON and OFF relay for 10 times.

Step 1-Algorithm	Step 2-Flowchart

Step 3- Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments

XII Resources Used

S. No.	Instrument /Components	Specification	Quantity
1.			
2.			
3.			

XIII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....

....

....

....

....

....

....

....

....

....

....

....

XIV Precautions Followed (use blank sheet provided if space not sufficient)

....

....

....

....

XV Observations for sample program (use blank sheet provided if space not sufficient)

Sr. NO.	Step	Port Pin Status	Logic 1 (+5V) / Logic 0 (0V)	Relay Status ON/OFF
1	Step 1	P2.0		
2	Step 2	P2.0		

XVI Results (Output of the Program)

....

....

....

....

XVII Interpretation of Results (Give meaning of the above obtained results)

....

....

....

....

XVIII Conclusions and Recommendation

....

....

....

....

XIX Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. Explain what will happen if driver circuit is not connected between microcontrollers and relay.
2. Draw the Interfacing diagram of microcontroller with relay using ULN 2003A IC.
3. Write Steps for testing a relay.

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

………

XIX References / Suggestions for further reading

1. https://en.wikipedia.org/wiki/Relay
2. http://www.circuitstoday.com/how-to-test-a-relay
3. The 8051 Microcontroller and Embedded system Using Assembly and C- Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. Mckinlay- Pearson /Prentice Hall, , 2nd edition, Delhi,2008, ISBN 978-8177589030

XX Assessment Scheme

Performance Indicators		Weightage
Process related(15 Marks)		**60%**
1	Coding and Debugging ability	30%
2	Making connections of hardware	20%
3	Follow ethical practices.	10%
Product related (10 Marks)		**40%**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition.	15%
6	Timely Submission of report, Answer to sample questions.	05%
Total (25 Marks)		**100%**

Name of Team Members

1 …………………………......

2 ………………………….......

3 …………………………..

4 …………………………..

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No. 14: Interface LCD with 8051 microcontroller to display the characters and decimal numbers

I Practical Significance

LCDs are used in a wide range of applications including Computers, televisions, instrument panels, calculator’s displays. They are common in consumer devices and have replaced cathode ray tube (CRT) displays in most applications. LCDs are interfaced with microcontrollers to display the given integer and character.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems.
- **Lifelong learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics and Telecommunication engineering and allied industry.

III Competency and Practical Skills

This practical is expected to develop the following skills for the industry identified competency ‘**Maintain microcontroller based systems’:**

- Select appropriate type of LCD for given application.
- Interface LCD display to microcontroller in 4 bit and 8 bit mode.
- Display given integer and character on LCD.

IV Relevant Course Outcome(s)

- Interface the memory and I/O devices to 8051 microcontroller.

V Practical Outcome

- Interface LCD with 8051 microcontroller to display the characters and decimal numbers.

VI Relevant Affective domain related Outcome(s)

- Follow good housekeeping.
- Maintain tools and equipment.
- Follow ethical practices.

VII Minimum Theoretical Background

A liquid-crystal display (LCD) is a flat panel display that uses the light modulating properties of liquid crystals. Liquid crystals do not emit light directly. An LCD is of two types depending upon how they made with either a passive matrix or an active matrix display grid.

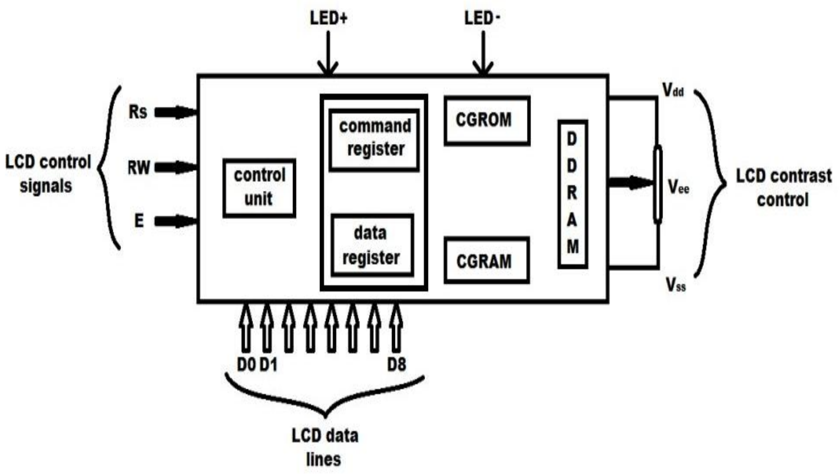


Fig. 14.1 Block Diagram of LCD display

Specifications:16 x2 LCD display

- Part No: JHD162A
- Char Dots：5 x 8
- Number of data line: 8 bit parallel
- Driving Mode：1/16 duty,1/5 bias
- Available Types: TN, STN (Yellow, Green, Grey, B/W)

Table 14.1 Operating parameters – Pin description of LCD module

Pin No	Function	Name	Pin No	Function	Name
1	Ground (0V)	Ground	**9**	8-bit data pins	DB2
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc	**10**	8-bit data pins	DB3
3	Contrast adjustment; through POT	V_{EE}	**11**	8-bit data pins	DB4
4	Selects command register	Register Select(RS)	**12**	8-bit data pins	DB5
5	Selects data register	Read/write(RW)	**13**	8-bit data pins	DB6
6	Sends data to data pins when a high to low pulse is given	Enable(E)	**14**	8-bit data pins	DB7
7	8-bit data pins	DB0	**15**	Backlight V_{CC} (5V)	LED+
8	8-bit data pins	DB1	**16**	Backlight Ground (0V)	LED-(GND)

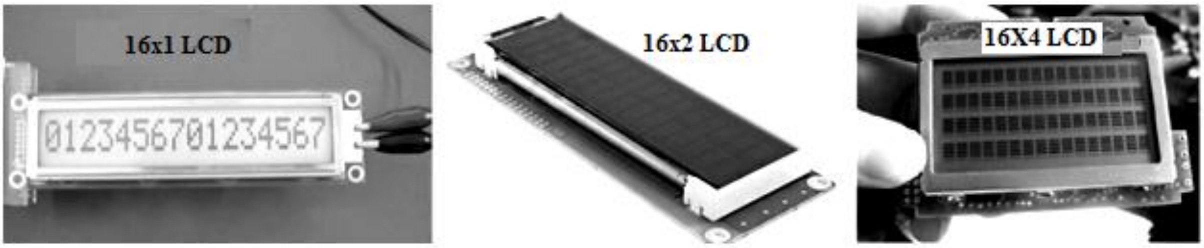


Fig. 14.2 Types of LCD

Basic unit of display is pixel and PIXEL= picture + element.

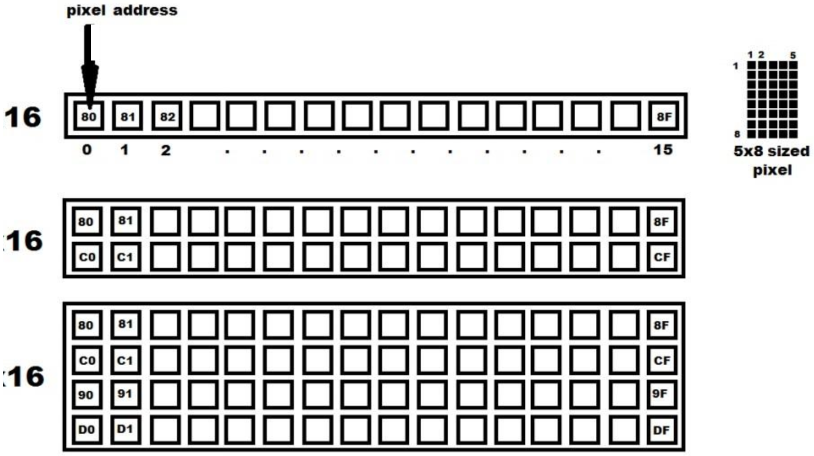


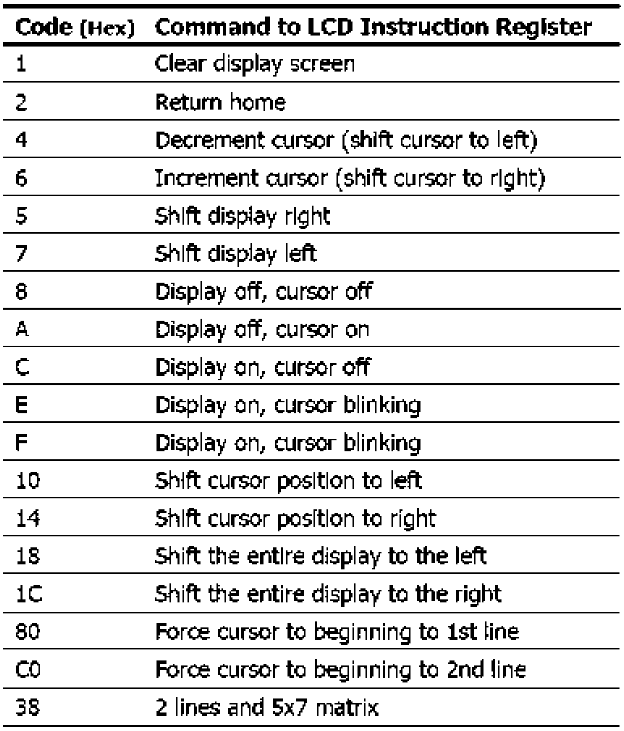
Fig 14.3 shows that each LCD have its own pixel rows and columns like 1×16 has single row and sixteen columns i.e. 16 pixel and each pixel size is of 5×8 as shown but we can see only 5×7 cause last raw is used by cursor. Each character is display according to these pixels size

Table 14.2 LCD control lines and function

Rs -> Register select	RW -> Read/Write	EN -> Enable

Rs=1 -data register
Rs=0 -command
	RW=1 -reading from LCD.
	RW=0 -writing to LCD.
		EN= high to low (Logic ‘1’ delay Logic ‘0’) for use LCD module.

Table 14.3 LCD commands



Code (Hex)	Command to LCD Instruction Register
1	Clear display screen
2	Return home
4	Decrement cursor (shift cursor to left)
6	Increment cursor (shift cursor to right)
5	Shift display right
7	Shift display left
8	Display off, cursor off
A	Display off, cursor on
C	Display on, cursor off
E	Display on, cursor blinking
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning to 1st line
C0	Force cursor to beginning to 2nd line
38	2 lines and 5x7 matrix

a) **Practical Circuit diagram :** Sample circuit diagram

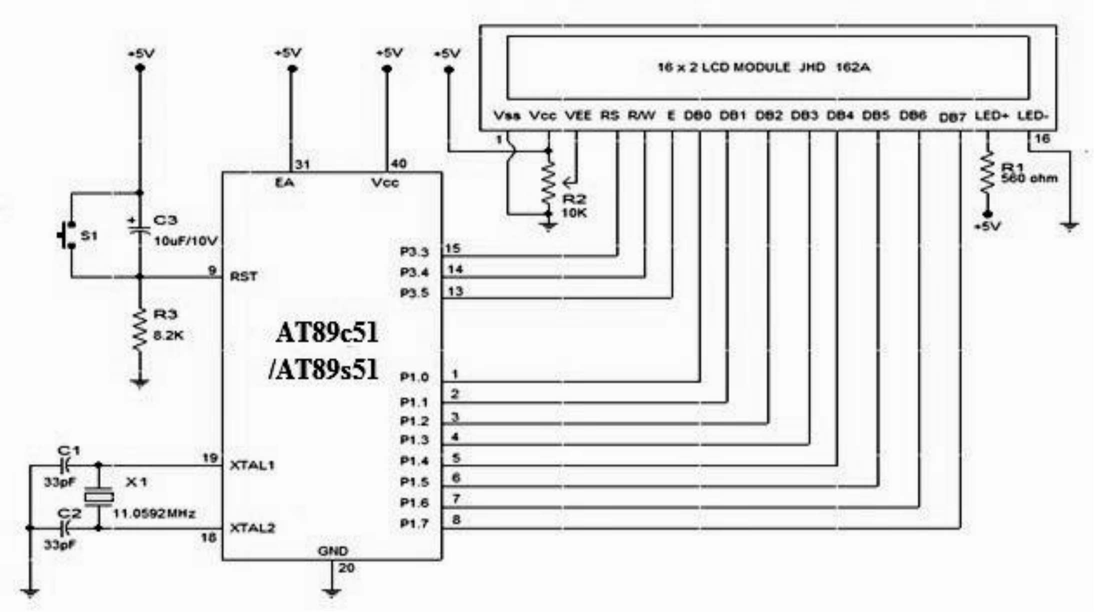


Fig 14.4 8051 connection to 16x2 LCD display

a) Practical setup

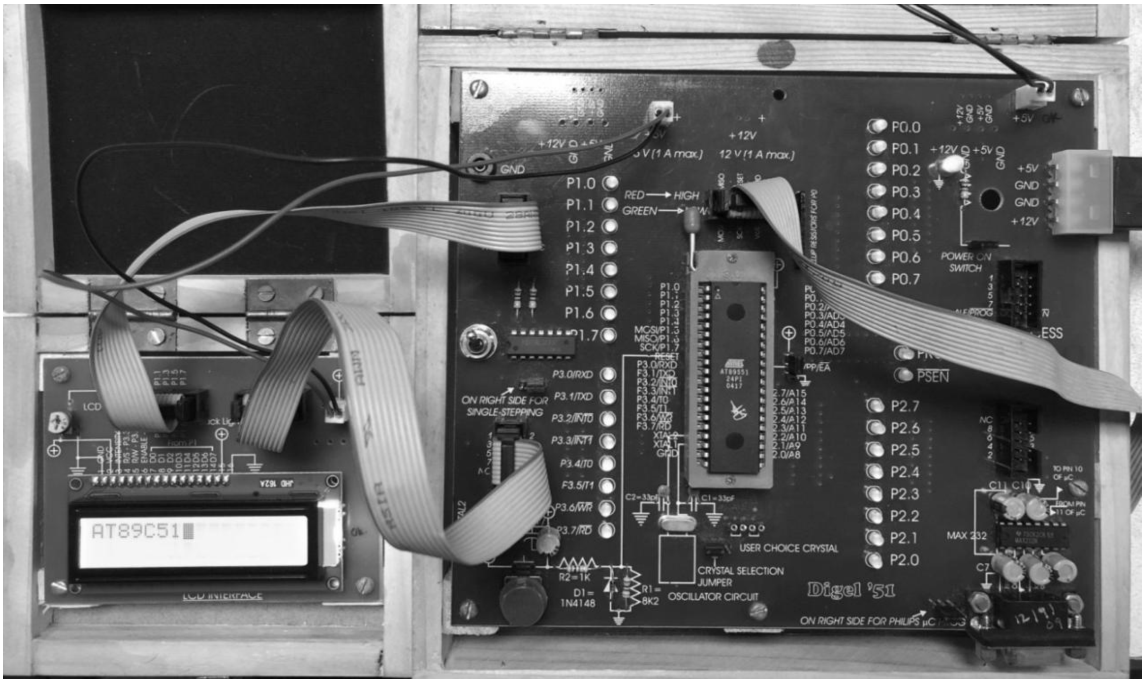


Fig 14.5 Practical Setup

b) Simulation diagram

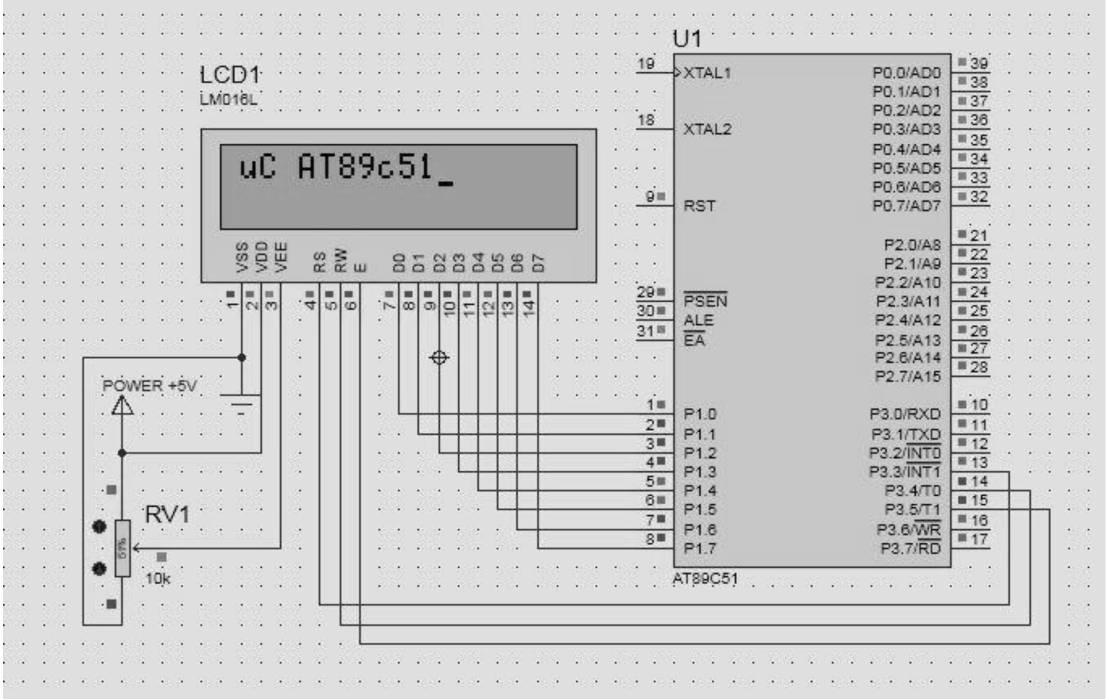


Fig 14.6 Simulation diagram

c) Actual circuit used in Laboratory

d) Actual Experimental set up used in laboratory

XI Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1.	Microcontroller kit	Single board systems with 8K RAM,ROM memory with battery backup,16X4,16X2LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler,RS-232,USB, interfacing facility with built in power supply.	1 No.
2.	Desktop PC	Loaded with open source IDE, simulation & program downloading software	1 No.
3.	LCD Trainer board	Suitable to interface with 8051 trainer kit	1

X Precautions to be Followed

1. LCD panel is made up of glass avoid applying strong pressure on to the surface of display area.
2. Ensure proper connection then only give electric supply to circuit.

XI Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop assembly program using Integrated Development Environment (IDE) or any other relevant software tool.
4. Debug program on IDE.
5. Execute program on IDE.
6. Create hex file for the above program.
7. Interface LCD display to microcontroller as per circuit diagram shown in Fig 14.4
8. Download hex code in EPROM/Flash memory of microcontroller
9. Observe output on LCD display and note it in observation Table 14.3

SAMPLE PROGRAM 1: Write a program to display “uC AT89c51” on LCD.

Step 1-Algorithm

1. Define control signals RS, RW and EN for LCD
2. Make LCD connected port as an output port.
3. Initialize LCD by sending commands.
4. Load DPTR with program memory address.
5. Set register as counter R2 =10 (decimal) for to display “uC AT89c51"
6. Clear Accumulator
7. Read ASCII from code memory into Accumulator.
8. Send code to output port where LCD is connected.
9. Increment memory pointer.
10. Decrement R2 counter. Is count =0? NO- go to step 6.
11. For repeat operation go to step 4.
12. Stop

Step 2-Flow Chart

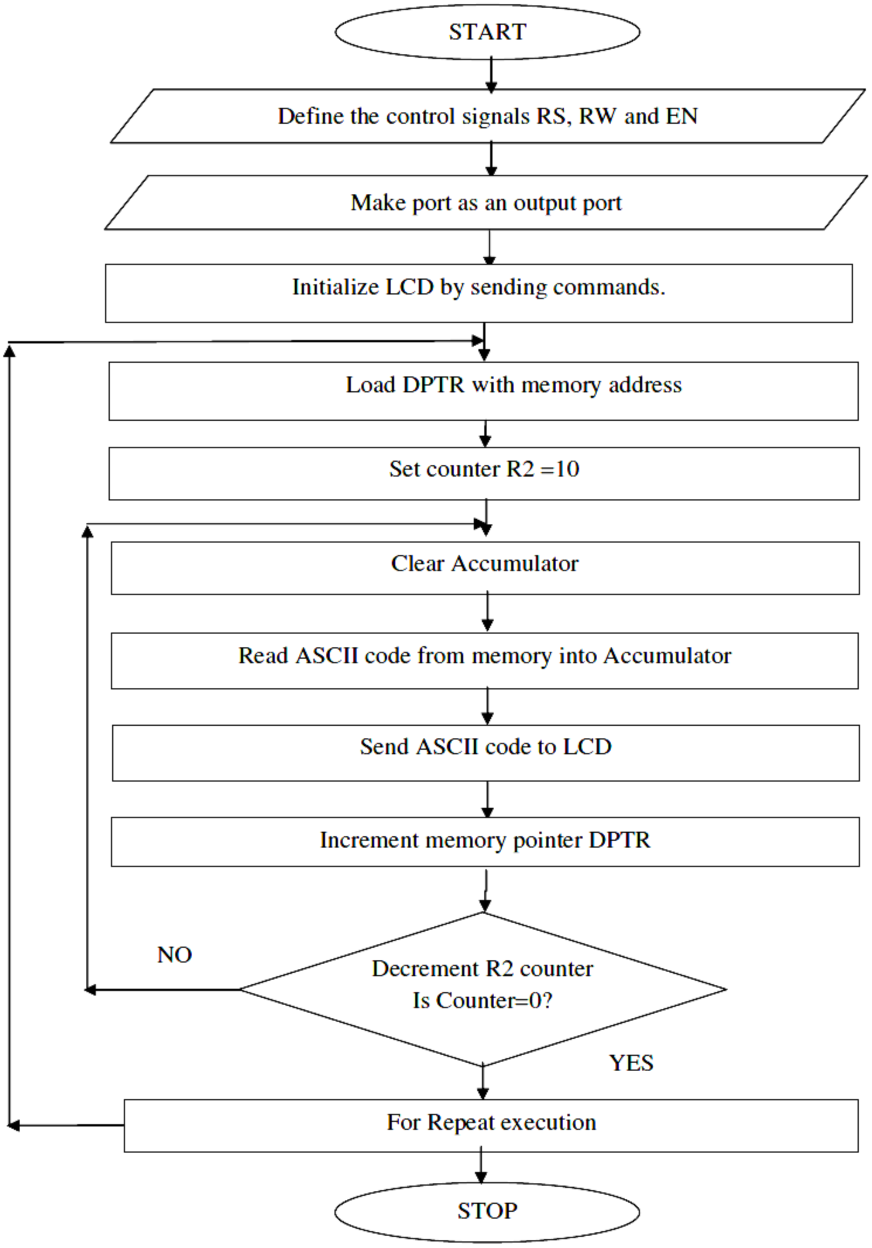

START
Define the control signals RS, RW and EN
Make port as an output port
Initialize LCD by sending commands.
Load DPTR with memory address
Set counter R2 =10
Clear Accumulator
Read ASCII code from memory into Accumulator
Send ASCII code to LCD
Increment memory pointer DPTR
NO
Decrement R2 counter
Is Counter=0?
YES
For Repeat execution
STOP

Fig 14.7 Flowchart to display alphabets and decimal numbers on LCD

Step 3- Assembly Language Sample Program

Memory Address	Hex Code	Label	Mnemonics	Comments
			RS BIT P3.3	;Replace a bit address by a symbol
			RW BIT P3.4	
			EN BIT P3.5	
			ORG 0000H	
C:0x0000	759000		MOV P1, #00H	;Set P1 as o/p port where LCD is connected
C:0x0003	120015		LCALL LCD_INIT	;Call LCD initialize subroutine
C:0x0006	900050		MOV DPTR, #MSG	; Load program memory address into DPTR
C:0x0009	7A0A		MOV R2, #10	;Set counter of 10
C:0x000B	E4	DISP_MSG:	CLR A	
C:0x000C	93		MOVC A,@A+DPTR	;Read data from memory into A register
C:0x000D	120037		LCALL LCDDATA	
C:0x0010	A3		INC DPTR	;Increment pointer to next location
C:0x0011	DFF8		DJNZ R7, DISP_MSG	;Repeat loop for 10 times
C:0x0013	80FE		SJMP $	
C:0x0015	7438	LCD_INIT:	MOV A, #38h	; 2 lines and 5×7 matrix (8-bit mode)
C:0x0017	112A		ACALL LCDCMD	
C:0x0019	740E		MOV A, #0eh	; Display on, cursor on
C:0x001B	112A		ACALL LCDCMD	
C:0x001D	7406		MOV A, #06h	; Increment cursor (shift cursor to right)
C:0x001F	112A		ACALL LCDCMD	
C:0x0021	7401		MOV A, #01h	; Clear display screen
C:0x0023	112A		ACALL LCDCMD	
C:0x0025	7480		MOV A, #80H	; Force cursor to beginning to 1st line
C:0x0027	112A		ACALL LCDCMD	
C:0x0029	22		RET	
C:0x002A	F590	LCDCMD:	MOV P1,A	;Send command to lcd
C:0x002C	C2B3		CLR RS	;Select command register
C:0x002E	C2 B4		CLR RW	;Select write operation
C:0x0030	D2B5		SETB EN	
C:0x0032	C2B5		CLR EN	;Latch command to lcd

C:0x0034	1144		ACALL DELAY	;Wait for sometime
C:0x0036	22		RET	
C:0x0037	F590	LCDDATA:	MOV P1,A	;Send data to lcd
C:0x0039	D2 B3		SETB RS	;Select data register
C:0x003B	C2 B4		CLR RW	;Select write operation
C:0x003D	D2B5		SETB EN	
C:0x003F	C2B5		CLR EN	;Latch data to lcd
C:0x0041	1144		ACALL DELAY	;Wait for sometime
C:0x0043	22		RET	
C:0x0044	7B32	DELAY:	MOV R3,#50	;Delay subroutine
C:0x0046	7CFF	LOOP2:	MOV R4,#255	
C:0x0048	DCFE	LOOP1:	DJNZ R4,LOOP1	
C:0x004A	DBFA		DJNZ R3,LOOP2	
C:0x004C	22		RET	
			ORG 0050H	
		MSG:	DB “uC AT89c51"	;Define data byte to code memory
			END	

Problem statement for student: Develop assembly program to display data “Wel” on 1st line and “Come” on 2nd line of LCD

Step 1-Algorithm	Step 2-Flowchart

Step 3- Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments

XII Resources Used

S. No.	Instrument /Components	Specification	Quantity
1.			
2.			
3.			

XIII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....

....

....

....

....

....

....

....

....

....

....

XIV Precautions Followed (use blank sheet provided if space not sufficient)

....

....

....

....

....

....

XV Observations for sample program (use blank sheet provided if space not sufficient)

Sr. No.	LCD Memory Location	Observed Alphabet and decimal number
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

XVI Results (Output of the Program)

....

....

....

....

....

XVII Interpretation of Results (Give meaning of the above obtained results)

....

....

....

....

....

XVIII Conclusions and Recommendation

....

....

....

XIX Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. Write down cost of LCD module for at least 3 different Manufacturer with specifications.
2. State possible reasons of 16x2 LCD display not working.
3. Write assembly program for checking busy flag in LCD display.
4. Write appropriate command to shift entire data to right.

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XIX References / Suggestions for further reading

1. https://en.wikipedia.org/wiki/Liquid-crystal_display.
2. https://pdf1.alldatasheet.com/datasheet-pdf/view/127934/ETC1/JHD162A.html
3. The 8051 Microcontroller and Embedded system Using Assembly and C- Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. Mckinlay- Pearson /Prentice Hall, , 2nd edition, Delhi,2008, ISBN 978-8177589030

XX Assessment Scheme

Performance indicators		Weightage
Process related(15 Marks)		**60%**
1	Coding and Debugging ability	30%
2	Making connections of hardware	20%
3	Follow ethical practices.	10%
Product related (10 Marks)		**40%**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition.	15%
6	Timely Submission of report, Answer to sample questions.	05%
Total (25 Marks)		**100%**

Name of Team Members

1 ………………………......

2 ……………………….......

3 …………………………..

4 …………………………..

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No. 15: Interface the given keyboard with 8051 and display the key pressed.

I Practical Significance

The predominant interface between humans and hardware is the keyboard in almost in all electronic applications. The matrix keypad allows a designer to implement a large number of inputs with a small number of microcontroller port pins. These keypads were evolved originally for telephonic applications involving touch tone signaling. This practical will help the students to develop skills to interface given keyboard to the microcontroller and display key pressed.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems.
- **Engineering tools:** Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations.
- **Lifelong learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics and Telecommunication engineering and allied industry

III Competency and Practical Skills

This practical is expected to develop the following skills for the industry identified competency '**Maintain microcontroller based systems':**

- Select appropriate type of matrix keypad for application.
- Interface matrix keypad to the microcontroller.
- Provide input data through keypad to the system.

IV Relevant Course Outcome(s)

- Interface the memory and I/O devices to 8051 microcontroller.

V Practical Outcome

- Interface the given keyboard with 8051 and display the key pressed.

VI Relevant Affective domain related Outcome(s)

- Follow safe practices.
- Follow ethical practices.
- Demonstrate working as a leader/a team member

VII Minimum Theoretical Background

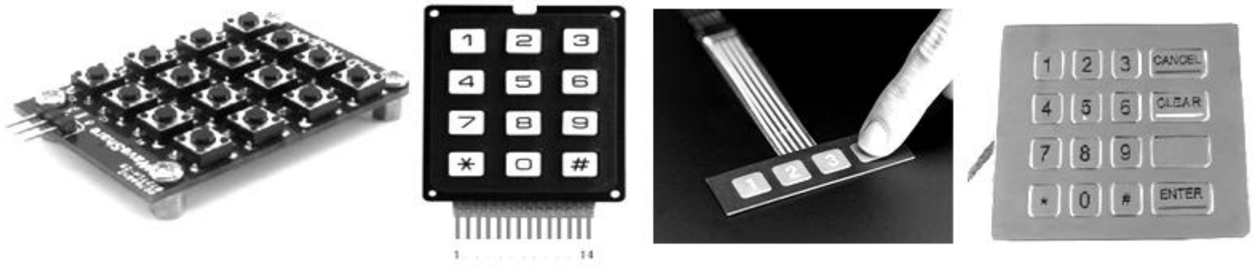


Fig 15.1 Keypad Types

4 x 4 matrix keypad connected to a single port of microcontroller. The keypad columns and rows are connected to the port pins. The keypad can be decoded to find out which key was pressed. When a key is pressed on the keypad, a row and column make a contact; otherwise, there is no connection

Specifications: Keypad

- Maximum voltage across each key: 24V
- Maximum Current through each key: 30mA
- Maximum operating temperature: 0°C to + 50°C
- Easy interface
- Long life

Table 15.1 4x4 Keypad: - Rows (R1, R2, R3, R4) Columns (C1, C2, C3, C4) (Refer Fig.15.2)

Step1:- Make R1- 0 Checks C1, C2, C3,C4 If C1=0 – ‘0’ is pressed If C2=0 – ‘1’ is pressed If C3=0 – ‘2’ is pressed If C3=0 – ‘3’ is pressed	Step3:- Make R3- 0 Checks C1, C2, C3,C4 If C1=0 – ‘8’ is pressed If C2=0 – ‘9’ is pressed If C3=0 – ‘A’ is pressed If C3=0 – ‘B’ is pressed
Step2:- Make R2- 0 Checks C1, C2, C3,C4 If C1=0 – ‘4’ is pressed If C2=0 – ‘5’ is pressed If C3=0 – ‘6’ is pressed If C3=0 – ‘7’ is pressed	Step4:- Make R4- 0 Checks C1, C2, C3,C4 If C1=0 – ‘C’ is pressed If C2=0 – ‘D’ is pressed If C3=0 – ‘E’ is pressed If C3=0 – ‘F’ is pressed

.

.

VIII Practical Circuit diagram :

a) Sample circuit diagram

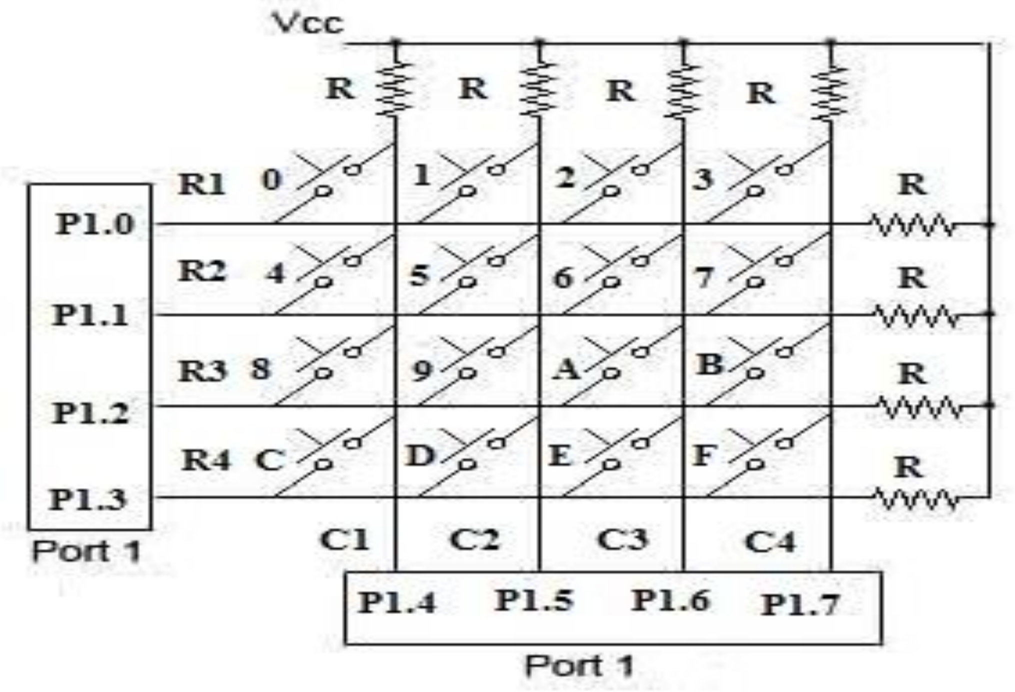


Fig 15.2 8051 connection to 4x4 matrix keypad

b) Practical setup

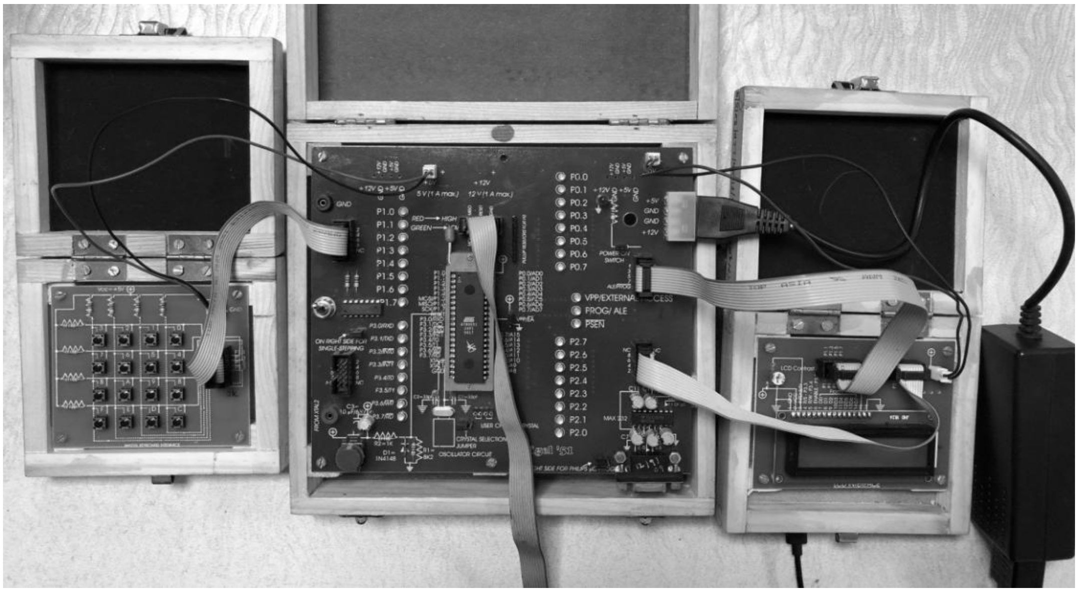


Fig 15.3 Practical Setup

c) Simulation diagram

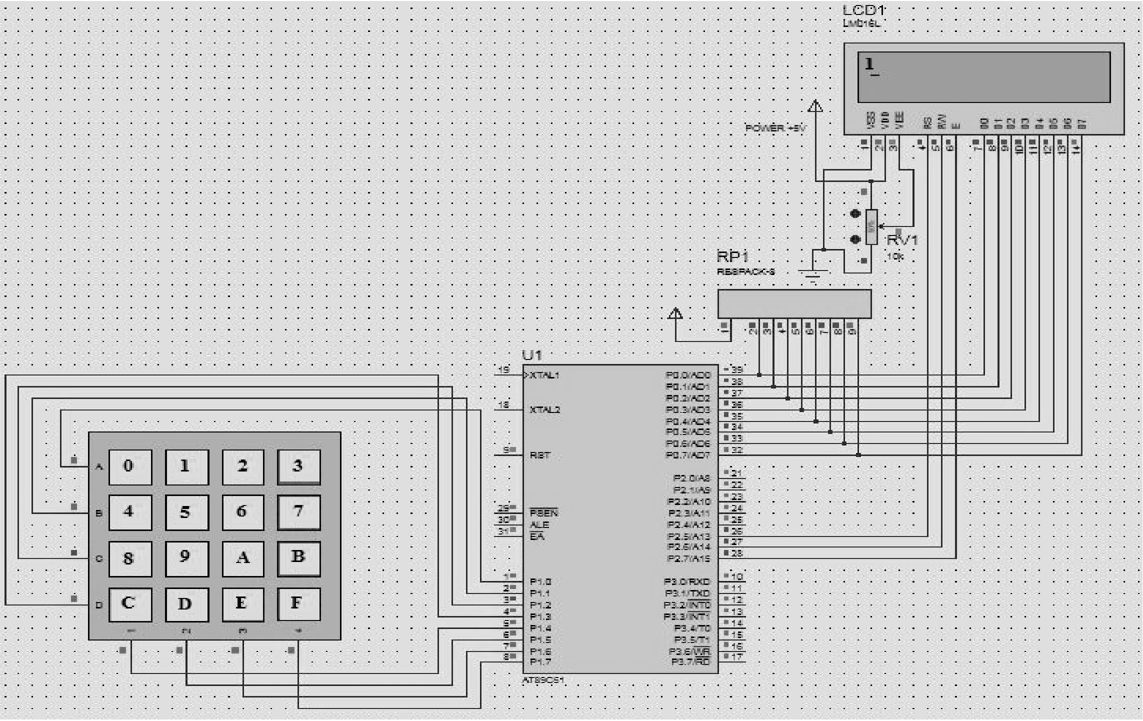


Fig 15.4 Simulation diagram

d) Actual circuit used in laboratory

e) Actual Experimental set up used in laboratory

IX Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1.	Microcontroller kit	Single board systems with 8K RAM,ROM memory with battery backup,16X4,16X2LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler,RS-232,USB, interfacing facility with built in power supply.	1 No.
2.	Desktop PC	Loaded with open source IDE, simulation and program downloading software	1 No.
3.	Keyboard 4x4 trainer board	Suitable to interface with 8051 trainer kit	1No.

X Precautions to be Followed

1. LCD panel is made up of glass avoid applying strong pressure on to the surface of display area.
2. Ensure proper connection then only give electric supply to circuit.

XI Procedure

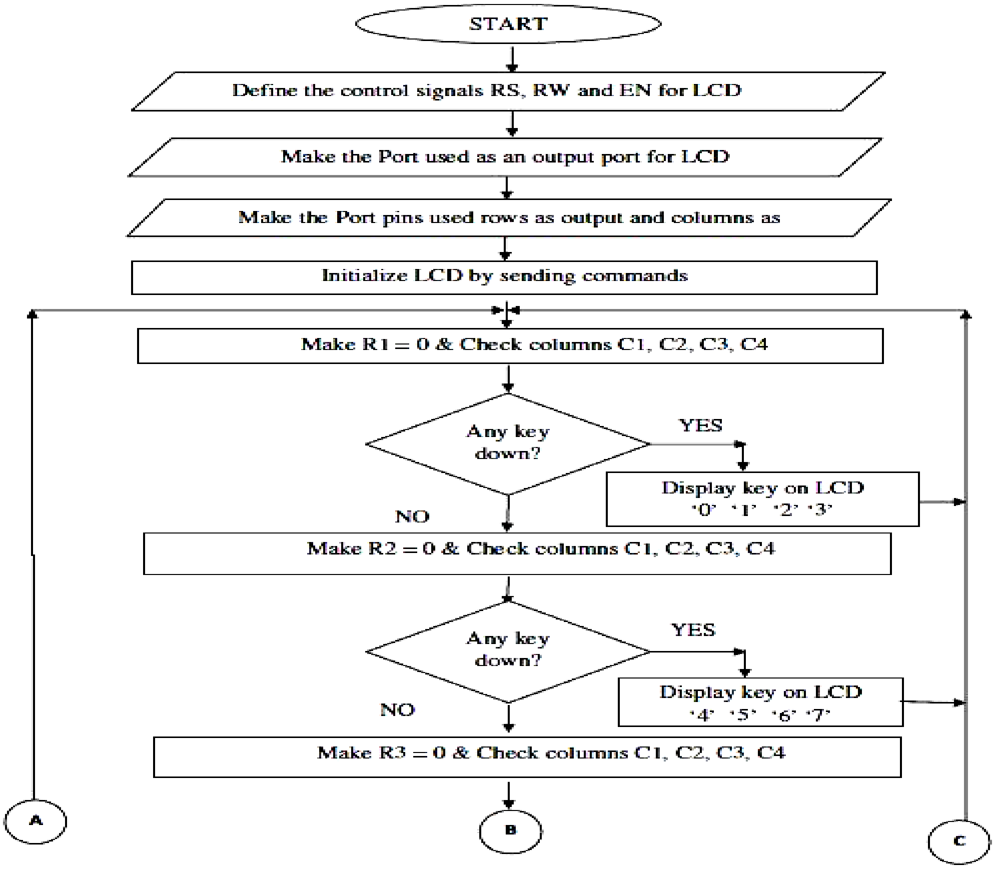
1. Write algorithm for given problem.
2. Draw flowchart.
3. Develop assembly program using Integrated Development Environment (IDE).
4. Debug and simulate program on IDE.
5. Create hex file.
6. Download hex code in EPROM/Flash memory of microcontroller.
7. Interface 4x4 keypad and 16x2 LCD to microcontroller.
8. Observe output status of key and note it in observation Table

SAMPLE PROGRAM 1: Write a program to display key pressed on LCD.

Step1- Algorithm

1. Define LCD control pins RS, RW, EN.
2. Make LCD Port P0 as an output port.
3. Make the keypad Port P1 pins as an input port.
4. Initialize LCD by sending commands.
5. Make R1 low& read columns C1, C2, C3 and C4
   If C1=0 display ‘0’, If C2=0 display ‘1’, If C3=0 display ‘2’, If C4=0 display ‘3’
6. Make R2 low& read columns C1, C2, C3 and C4.
   If C1=0 display ‘4’, If C2=0 display ‘5’, If C3=0 display ‘6’, If C4=0 display ‘7’.
7. Make R3 low& read columns C1, C2, C3 and C4.
   If C1=0 display ‘8’, If C2=0 display ‘9’, If C3=0 display ‘A’, If C4=0 display ‘B’
8. Make R4 low& read columns C1, C2, C3 and C4.
   If C1=0 display ‘C’, If C2=0 display ‘D’, If C3=0 display ‘E’, If C4=0 display ‘F’
9. Go to step 5 to scan keypad.
10. Stop

Step 2-Flow Chart


START
Define the control signals RS, RW and EN for LCD
Make the Port used as an output port for LCD
Make the Port pins used rows as output and columns as
Initialize LCD by sending commands
Make R1 = 0 & Check columns C1, C2, C3, C4
Any key down?
YES
Display key on LCD '0' '1' '2' '3'
NO
Make R2 = 0 & Check columns C1, C2, C3, C4
Any key down?
YES
Display key on LCD '4' '5' '6' '7'
NO
Make R3 = 0 & Check columns C1, C2, C3, C4
A
B
C

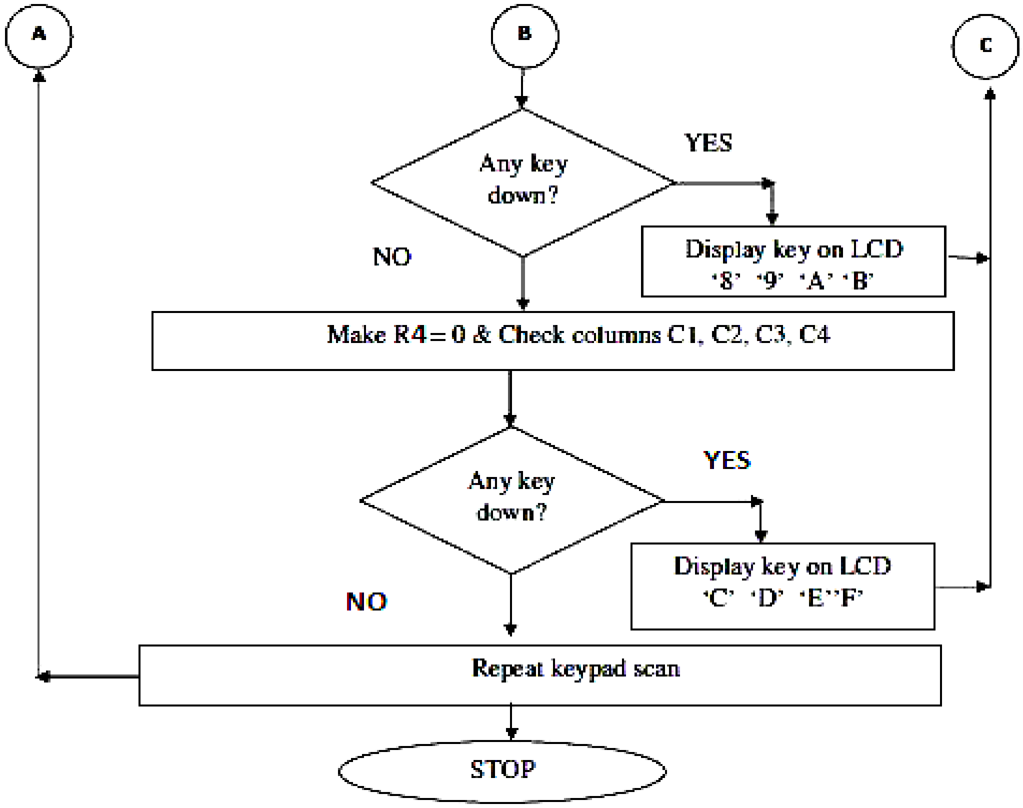


Fig 15.5 Flowchart to display the key pressed on LCD

Step 3-Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments
			ROW1 BIT P1.0	;keypad row pins
			ROW2 BIT P1.1	
			ROW3 BIT P1.2	
			ROW4 BIT P1.3	
			COL1 BIT P1.4	;keypad column pins
			COL2 BIT P1.5	
			COL3 BIT P1.6	
			COL4 BIT P1.7	
			RS BIT P2.5	;LCD control pins
			RW BIT P2.6	
			EN BIT P2.7	
			ORG 0000H	
C:0x0000	758000		MOV P0,#00H	;make LCD port as output
C:0x0003	11AB		ACALL LCDINIT	
C:0x0005	7590FF	KEYSCAN:	MOV P1,#0FFH	;step 1 make port as input
C:0x0008	C290		CLR ROW1	;make row1=0

C:0x000A	30943E		JNB COL1,K0	;check col1
C:0x000D	309441		JNB COL2,K1	;check col2
C:0x0010	309444		JNB COL3,K2	;check col3
C:0x0013	309447		JNB COL4,K3	;check col4
C:0x0016	7590FF		MOV P1,#0FFH	;step 2
C:0x0019	C291		CLR ROW2	;make row2=0
C:0x001B	309445		JNB COL1,K4	
C:0x001E	309548		JNB COL2,K5	
C:0x0021	30964B		JNB COL3,K6	
C:0x0024	30974E		JNB COL4,K7	
C:0x0027	7590FF		MOV P1,#0FFH	;step 3
C:0x002A	C292		CLR ROW3	;make row3=0
C:0x002C	30954C		JNB COL1,K8	
C:0x002F	30954F		JNB COL2,K9	
C:0x0032	309652		JNB COL3,KA	
C:0x0035	309755		JNB COL4,KB	
C:0x0038	7590FF		MOV P1,#0FFH	;step 4
C:0x003B	C293		CLR ROW4	;make row4=0
C:0x003D	309453		JNB COL1,KC	
C:0x0040	309556		JNB COL2,KD	
C:0x0043	309659		JNB COL3,KE	
C:0x0046	30975C		JNB COL4,KF	
C:0x0049	80BA		SJMP KEYSCAN	;repeat scanning
C:0x004B	7430	K0:	MOV A,#”0”	;key 0 detected
C:0x004D	11CD		ACALL DISPLAY	
C:0x004F	0105		AJMP KEYSCAN	
C:0x0051	7431	K1:	MOV A,#”1”	;key 1 detected
C:0x0053	11CD		ACALL DISPLAY	
C:0x0055	0105		AJMP KEYSCAN	
C:0x0057	7432	K2:	MOV A,#”2”	;key 2 detected
C:0x0059	11CD		ACALL DISPLAY	
C:0x005B	0105		AJMP KEYSCAN	
C:0x005D	7433	K3:	MOV A,#”3”	;key 3 detected
C:0x005F	11CD		ACALL DISPLAY	
C:0x0061	0105		AJMP KEYSCAN	
C:0x0063	7434	K4:	MOV A,#”4”	;key 4 detected
C:0x0065	11CD		ACALL DISPLAY	

C:0x0067	0105		AJMP KEYSCAN	
C:0x0069	7435	K5:	MOV A,#”5”	;key 5 detected
C:0x006B	11CD		ACALL DISPLAY	
C:0x006D	0105		AJMP KEYSCAN	
C:0x006F	7436	K6:	MOV A,#”6”	;key 6 detected
C:0x0071	11CD		ACALL DISPLAY	
C:0x0073	0105		AJMP KEYSCAN	
C:0x0075	7437	K7:	MOV A,#”7”	;key 7 detected
C:0x0077	11CD		ACALL DISPLAY	
C:0x0079	0105		AJMP KEYSCAN	
C:0x007B	7438	K8:	MOV A,#”8”	;key 8 detected
C:0x007D	11CD		ACALL DISPLAY	
C:0x007F	0105		AJMP KEYSCAN	
C:0x0081	7439	K9:	MOV A,#”9”	;key 9 detected
C:0x0083	11CD		ACALL DISPLAY	
C:0x0085	0105		AJMP KEYSCAN	
C:0x0087	7441	KA:	MOV A,#”A”	;key A detected
C:0x0089	11CD		ACALL DISPLAY	
C:0x008B	0105		AJMP KEYSCAN	
C:0x008D	7442	KB:	MOV A,#”B”	;key B detected
C:0x008F	11CD		ACALL DISPLAY	
C:0x0091	0105		AJMP KEYSCAN	
C:0x0093	7443	KC:	MOV A,#”C”	;key C detected
C:0x0095	11CD		ACALL DISPLAY	
C:0x0097	0105		AJMP KEYSCAN	
C:0x0099	7444	KD:	MOV A,#”D”	;key D detected
C:0x009B	11CD		ACALL DISPLAY	
C:0x009D	0105		AJMP KEYSCAN	
C:0x009F	7445	KE:	MOV A,#”E”	;key E detected
C:0x00A1	11CD		ACALL DISPLAY	
C:0x00A3	0105		AJMP KEYSCAN	
C:0x00A5	7446	KF:	MOV A,#”F”	;key F detected
C:0x00A7	11CD		ACALL DISPLAY	
C:0x00A9	0105		AJMP KEYSCAN	
C:0x00AB	7438	LCDINIT:	MOV A,#38H	;init LCD 2 lines, 5x7 matrix
C:0x00AD	11C0		ACALL COMMAND	

C:0x00AF	740E		MOV A,#0EH	;LCD on cursor on
C:0x00B1	11C0		ACALL COMMAND	
C:0x00B3	7406		MOV A,#06H	;clear LCD command
C:0x00B5	11C0		ACALL COMMAND	
C:0x00B7	7401	CLEAR:	MOV A,#01H	;shift cursor right
C:0x00B9	11C0		ACALL COMMAND	
C:0x00BB	7480		MOV A,#80H	;Force cursor to beginning of 1st line
C:0x00BD	11C0		ACALL COMMAND	
C:0x00BF	22		RET	
C:0x00C0	F580	COMMAND:	MOV P0,A	;issue command code
C:0x00C2	C2A5		CLR RS	
C:0x00C4	C2A6		CLR RW	
C:0x00C6	D2A7		SETB EN	
C:0x00C8	11DE		ACALL DELAY	
C:0x00CA	C2A7		CLR EN	
C:0x00CC	22		RET	
C:0x00CD	F580	DISPLAY:	MOV P0,A	;issue data
C:0x00CF	D2A5		SETB RS	
C:0x00D1	C2A6		CLR RW	
C:0x00C3	D2A7		SETB EN	
C:0x00D5	11DE		ACALL DELAY	
C:0x00D7	C2A7		CLR EN	
C:0x00D9	11DE		ACALL DELAY	;add delay
C:0x00DB	11B7		ACALL CLEAR	;LCD clear
C:0x00DD	22		RET	
C:0x00DE	7B32		MOV R3,#50	
C:0x00E0	7CFF		MOV R4,#255	
C:0x00E2	DCFE	DELAY:	DJNZ R4,LOOP1	
C:0x00E4	DBFA	LOOP2:	DJNZ R3,LOOP2	
C:0x00E6	22	LOOP1:	RET	
			END	

XII Resources Used

Sr. No.	Instrument /Components	Specification	Quantity
1.			
2.			
3.			
4.			
5.			

XIII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....

....

....

....

....

....

....

....

....

....

....

XIV Precautions Followed (use blank sheet provided if space not sufficient)

....

....

....

....

XV Observations for sample program (use blank sheet provided if space not sufficient)

Sr. No.	Key Pressed	Output on LCD
1	2	
2	5	
3	9	
4	B	
5	E	

XVI Results (Output of the program)

....

....

....

....

....

XVII Interpretation of Results (Give meaning of the above obtained results)

....

....

....

....

....

....

XVIII Conclusions and Recommendation

....

....

....

XIX Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. If the switch 3 is pressed which column & row show logic 0 at respective port pin?.
2. Write specifications of 4x4 keypad in terms of.
   a) Maximum voltage across each key.
   b) Maximum Current through each key.
   c) Maximum operating temperature.
3. Explain debounce effect.
4. Write down following requirement of port pins and number of switches.

Matrix Keypad	No of switches	Rows Port Pins	Columns Port Pins
1x4 Matrix Keypad			
3x4 Matrix Keypad			
4x4 Matrix Keypad			
8x8 Matrix Keypad			

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XIX References / Suggestions for further reading

1. https://en.wikipedia.org/wiki/Keypad.
2. https://components101.com/misc/4x4-keypad-module-pinout-configuration-features-datasheet.
3. The 8051 Microcontroller and Embedded system Using Assembly and C- Muhammad
   Ali Mazidi, Janice Gillispie Mazidi, Rolin D. Mckinlay- Pearson /Prentice Hall, , 2nd edition, Delhi,2008, ISBN 978-8177589030.

XX Assessment Scheme

Performance Indicators		Weightage
Process related(15 Marks)		**60% (15)**
1	Coding and Debugging ability	30%
2	Making connections of hardware	20%
3	Follow ethical practices.	10%
Product related (10 Marks)		**40%(10)**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition.	15%
6	Timely Submission of report, Answer to sample questions.	05%
Total (25 Marks)		**100% (25)**

Name of Team Members

1 …………………………......

2 ………………………….......

3 …………………………..

4 …………………………..

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No. 16: Interface ADC with 8051 microcontroller and verify input / output.

I Practical Significance

In the real world most of the signals sensed and processed by humans are analog signals. Analog-to-digital conversion is the primary means by which analog signals are converted into digital data that can be processed by computers for various purposes. This practical will help the students to develop skills to interface LCD to the microcontroller and verify input/output.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems.
- **Engineering tools:** Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations.
- **Lifelong learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics and Telecommunication engineering and allied industry

III Competency and Practical Skills

This practical is expected to develop the following skills for the industry identified competency '**Maintain microcontroller based systems':**

- Select appropriate type of ADC for given application.
- Interface ADC to the microcontroller.
- Verify input and output.

IV Relevant Course Outcome(s)

- Maintain microcontroller used in different application.

V Practical Outcome

- Interface ADC with 8051 microcontroller and verify input/output.

VI Relevant Affective domain related Outcome(s)

- Follow safe practices.
- Practice good housekeeping.
- Practice energy conservation
- Maintain tools and equipment

VII Minimum Theoretical Background

An analog-to-digital converter, or simply ADC, is a semiconductor device that is used to convert an analog signal into a digital code. An analog signal is a signal that may assume any value within a continuous range. Examples of analog signals commonly encountered every day are sound, light, temperature, and pressure, all of which may be represented electrically by an analog voltage or current.

Types of ADC:

Serial ADC: Serial ADC's consisting of just one output pin that delivers the output code one bit at a time.

Parallel ADC: Parallel ADC's consisting of several output pins that deliver all the bits of the output code at the same time.

Specifications ADC0808 Chip

ADC0808 IC - analog to digital
Resolution -8 Bits
Input Channels- 8
Single Supply- 5VDC
Low Power- 15mW
Conversion Time -100μs
Output's meets TTL voltage level.
The clock frequency range of ADC is 10 KHz to 1280 KHz.
Typically 680 kHz used.
Low power consumption

ADC0808 Analog signal selection:

Analog Channel	C	B	A
IN0	0	0	0
IN1	0	0	1
IN2	0	1	0
IN3	0	1	1
IN4	1	0	0
IN5	1	1	1
IN6	1	1	0
IN7	1	1	1

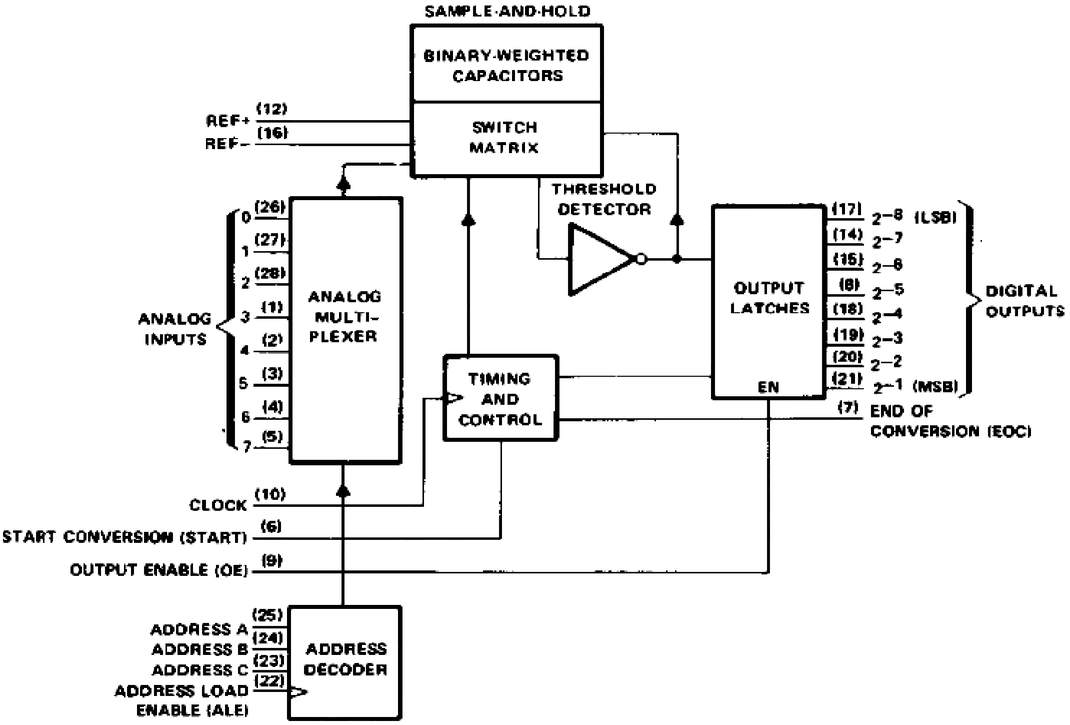


Fig 16.1 Functional block diagram of ADC 0808 chip

VIII Practical Circuit diagram:

a) Sample circuit diagram

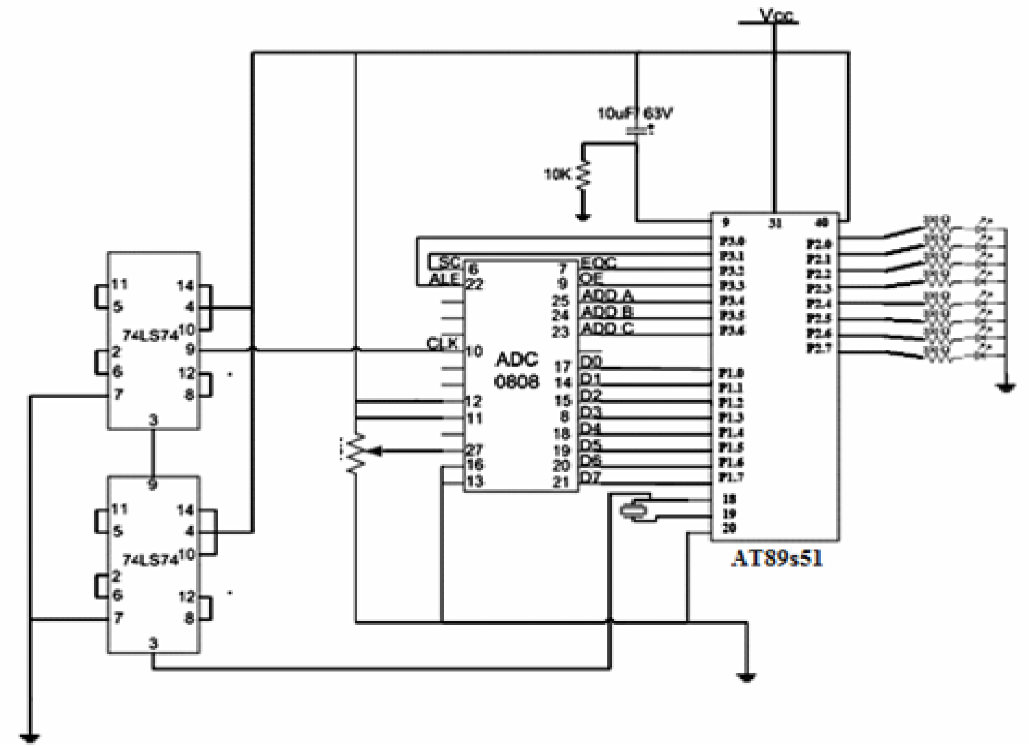


Fig 16.2 8051 connection to ADC 0808

b) Practical Setup

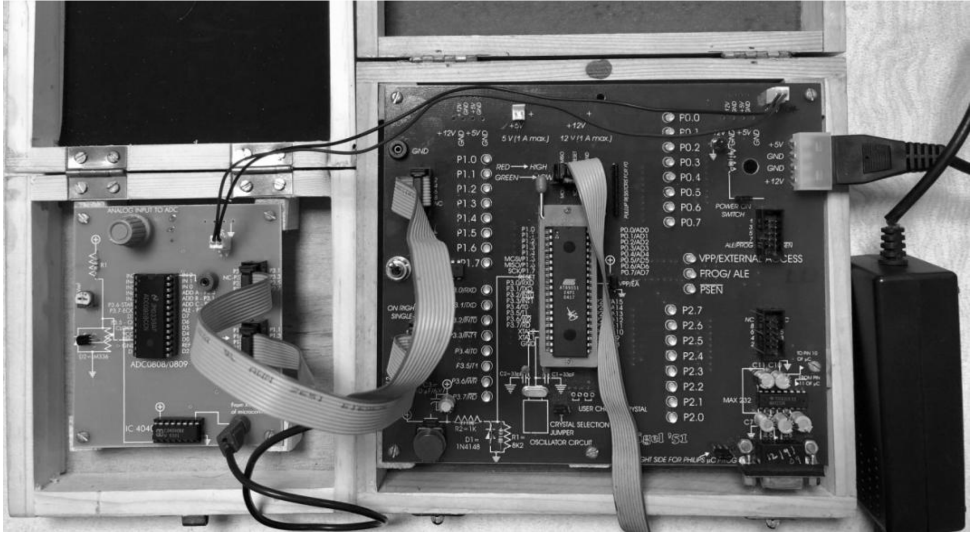


Fig 16.3 Practical Setup

c) Actual circuit used in laboratory

d) Actual Experimental set up used in laboratory

IX Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1.	Microcontroller kit	Single board systems with 8K RAM,ROM memory with battery backup,16X4,16X2LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler,RS-232,USB, interfacing facility with built in power supply.	1 No.
2.	Desktop PC	Loaded with open source IDE, simulation and program downloading software	1 No.
3.	ADC (0808) trainer board	Suitable to interface 8051 board.	1 No.

X Precautions to be Followed

1. Refer datasheet for to provide clock frequency to ADC 0808 chip.
2. Care must be taken while taking observations during power up.
3. Use current limiting resistors for LED’s.

XI Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop assembly program using Integrated Development Environment (IDE) or any other relevant software tool.
4. Debug program on IDE.
5. Execute program on IDE.
6. Create hex file for the above program.
7. Download hex code in EPROM/Flash memory of the microcontroller.
8. Interface ADC 0808 IC to the microcontroller as per circuit diagram shown in fig 16.2
9. To make changes in input refer observation Table and note hex output observed on LEDs.

SAMPLE PROGRAM 1: Write a program to read the data from ADC and display on 8 LEDs connected to port 2.

STEP1- Algorithm

1. Select an analog channel by providing bits to A, B, and C addresses according to the analog signal selection table.
2. Activate the ALE (address latch enable) pin.
3. Activate SC (start conversion) to initiate conversion.
4. Monitor EOC (end of conversion) to see whether conversion is finished. H-to L output indicates that the data is converted and is ready to be picked up. If we do not use EOC, we can read the converted digital data after a brief time delay. The delay size depends on the speed of the external clock we connect to the CLK pin.
5. Activate OE (output enable) to read data out of the ADC chip.

Note: In ADC0808 that there is no self-clocking and the clock must be provided from an external source to the CLK pin. Although the speed of conversion depends on the frequency of the clock connected to the CLK pin, it cannot be faster than 100 microseconds.

Step 2-Flow Chart

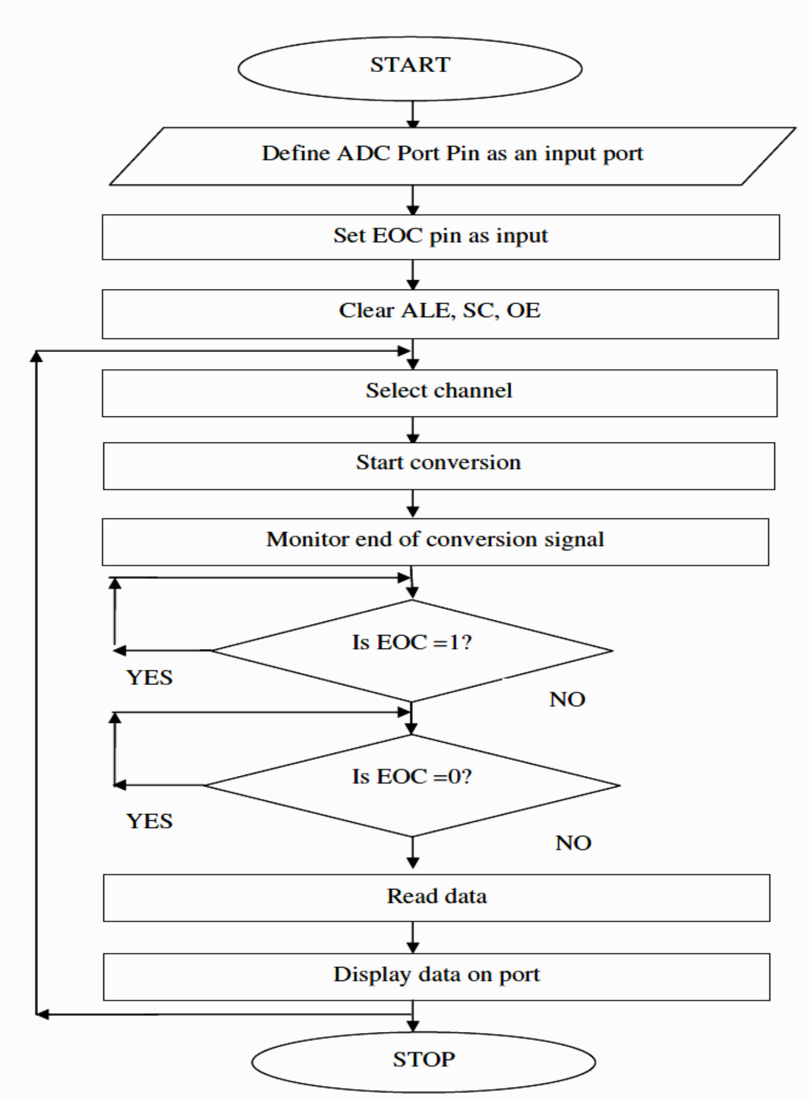


Fig 16.4 Flowchart for ADC

Step 3-Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments
			ALE BIT P3.0	
			SC BIT P3.1	
			EOC BIT P3.2	
			OE BIT P3.3	
			ADDR_A BIT P3.4	
			ADDR_B BIT P3.5	
			ADDR_C BIT P3.6	
			MY_DATA EQU P1	
			ORG 0000H	
C:0x0000	7590FF		MOV MY_DATA,#0FFH	;make P1 as input
C:0x0003	D2B2		SETB EOC	;make EOC an input
C:0x0005	C2B0		CLR ALE	;clear ALE
C:0x0007	C2B1		CLR SC	;clear WR
C:0x0009	C2B3		CLR OE	;clear RD
C:0x000B	C2B6	BACK:	CLR ADDR_C	;C=0
C:0x000D	C2B5		CLR ADDR_B	;B=0
C:0x000F	D2B4		SETB ADDR_A	;A=1(select channel 1)
C:0x0011	1131		ACALL DELAY	
C:0x0013	D2B0		SETB ALE	;latch address
C:0x0015	1131		ACALL DELAY	
C:0x0017	D2B1		SETB SC	;start conversion
C:0x0019	1131		ACALL DELAY	
C:0x001B	C2B0		CLR ALE	
C:0x001D	C2B1		CLR SC	

C:0x001F	20B2FD	HERE:	JB EOC,HERE	;wait
C:0x0022	30B3FD	HERE1:	JNB EOC,HERE1	
C:0x0025	D2B3		SETB OE	
C:0x0027	1131		ACALL DELAY	
C:0x0029	E590		MOV A,MY_DATA	
C:0x002B	F590		MOV P2,A	
C:0x002D	C2B3		CLR OE	
C:0x002F	80DA		SJMP BACK	
C:0x0031	7B19	DELAY:	MOV R3,#25	;Delay Subroutine
C:0x0033	7C64	L3:	MOV R4,#100	
C:0x0035	7D64	L2:	MOV R5,#100	
C:0x0037	DDFE	L1:	DJNZ R5,L1	
C:0x0039	DCFA		DJNZ R4,L2	
C:0x003B	DBF6		DJNZ R3,L3	
C:0x003D	22		RET	
			END	

XII Resources Used

S. No.	Instrument /Components	Specification	Quantity
1.			
2.			
3.			

XIII Actual Procedure Followed (use blank sheet if space not sufficient)

....

....

....

....

....

....

....

....

XIV Precautions Followed (use blank sheet if space not sufficient)

....

....

....

....

....

....

XV Observations (use blank sheet if space not sufficient)

Input Voltage	Output HEX Value observed on LED's
1V	
2V	
3V	
4V	
5V	

XVI Results (Output of the Program)

....

....

....

....

XVII Interpretation of Results (Give meaning of the above obtained results)

....

....

....

....

XVIII Conclusions and Recommendation (Actions/decisions to be taken based on the interpretation of results).

....

....

....

....

XIX Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. If Vref pin is connected to 2.56V then what will be the step size of ADC0808?
2. List two applications where serial ADC and parallel ADC chips are used.
3. Draw timing diagram for selecting a channel and read data for ADC 0808.
4. Distinguish ADC 0804, ADC 0808 and ADC 0848.

[Space for Answers]

....

....

....

....

....

XIX References / Suggestions for further reading

1. https://en.wikipedia.org/wiki/Analog-to-digital converter.
2. https://pdf1.alldatasheet.com/datasheet-pdf/view/155397/TI/ADC0808.html
3. The 8051 Microcontroller and Embedded system Using Assembly and C- Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. Mckinlay- Pearson /Prentice Hall, , 2nd edition, Delhi,2008, ISBN 978-8177589030.

XX Assessment Scheme

Performance Indicators		Weightage
Process related(15 Marks)		**60%**
1	Coding and Debugging ability	30%
2	Making connections of hardware	20%
3	Follow ethical practices.	10%
Product related (10 Marks)		**40%**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition.	15%
6	Timely Submission of report, Answer to sample questions.	05%
Total (25 Marks)		**100%**

Name of Team Members

1 ………………………….....

2 …………………………......

3 …………………………..

4 …………………………..

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No. 17: Interface DAC with 8051 microcontroller and observe following waveforms: square wave, triangular wave, sawtooth wave

I Practical Significance

The digital to analog converter (DAC) is a device widely used to convert digital pulses to analog signals. This practical will help the students to develop skills to interface DAC with 8051 and generate different analog waveforms.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems.
- **Engineering tools:** Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations.
- **Lifelong learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics and Telecommunication engineering and allied industry

III Competency and Practical Skills

This practical is expected to develop the following skills for the industry identified competency '**Maintain microcontroller based systems':**

- Select appropriate type DAC for given application.
- Interface DAC with microcontroller.
- Generate different analog signals.

IV Relevant Course Outcome(s)

- Maintain microcontroller used in different application.

V Practical Outcome

- Interface DAC with 8051 microcontroller and observe following waveforms: square wave, triangular wave, sawtooth wave.

VI Relevant Affective domain related Outcome(s)

- Follow safe practices.
- Practice good housekeeping.
- Practice energy conservation
- Maintain tools and equipment

VII Minimum Theoretical Background

In DAC the number of data bit input decides the resolution since the number of analog output levels is equal to 2_n, where n is the number of data bit inputs. An 8 input DAC provides 256 discrete voltage (or current). The most commonly used, 8 bit, R/2R method followed DAC is DAC 0808

Two methods adopted for designing a DAC

1. Binary weighted **2.** R/2R Ladder

Specifications DAC 0808 Chip

- DAC 0808 - 8-bit monolithic digital-to-analog converter (DAC)
- Full scale output current settling time of 150 ns
- Power dissipation 33 mW with ±5V supplies.
- Relative accuracy: ±0.19% error maximum
- No inverting digital inputs are TTL and CMOS compatible
- High speed multiplying input slew rate: 8 mA/µs
- Power supply voltage range: ±4.5V to ±18V

Equation:

$$I_{out} = I_{ref}\left(\frac{D7}{2}+\frac{D6}{4}+\frac{D5}{8}+\frac{D4}{16}+\frac{D3}{32}+\frac{D2}{64}+\frac{D1}{128}+\frac{D0}{256}\right)$$

Where D0 is the LSB, D7 is the MSB for the inputs, and I_{ref} is the input current. The I_{ref} current is generally set to 2.0mA.

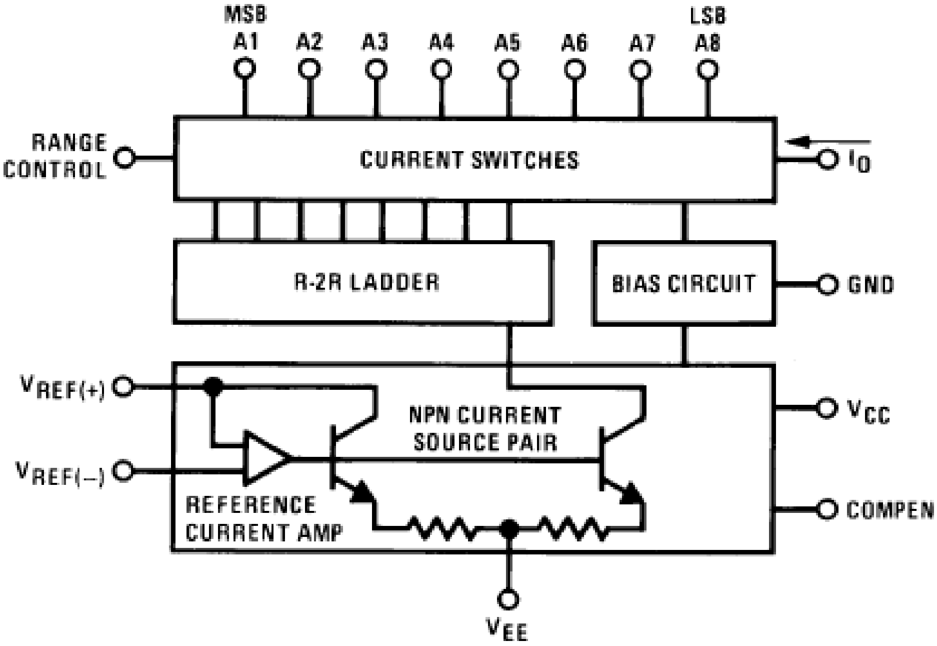


Fig 17.1 Functional block diagram of DAC 0808 chip

VIII Practical Circuit diagram:

a) Sample circuit diagram

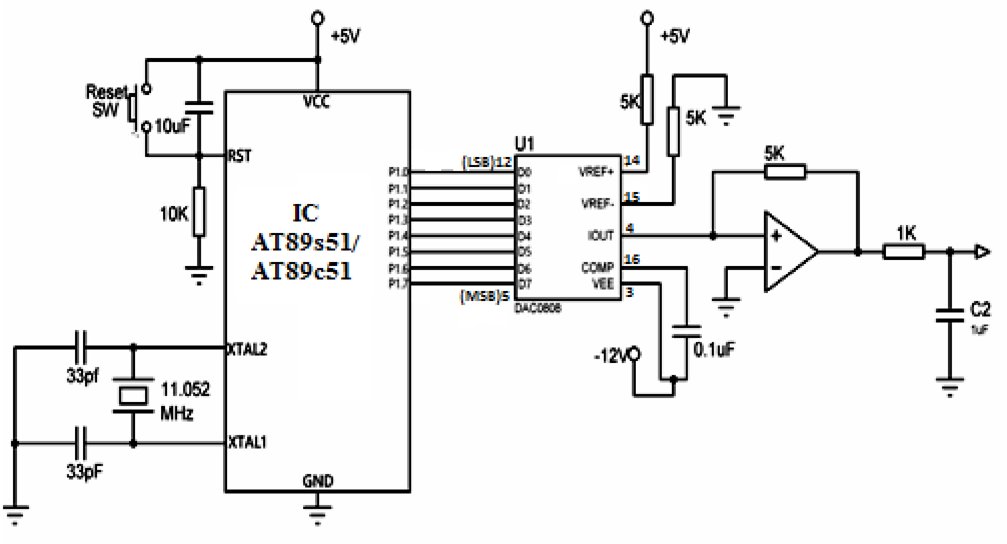


Fig 17.2 8051 connection to DAC 0808

b) Practical setup

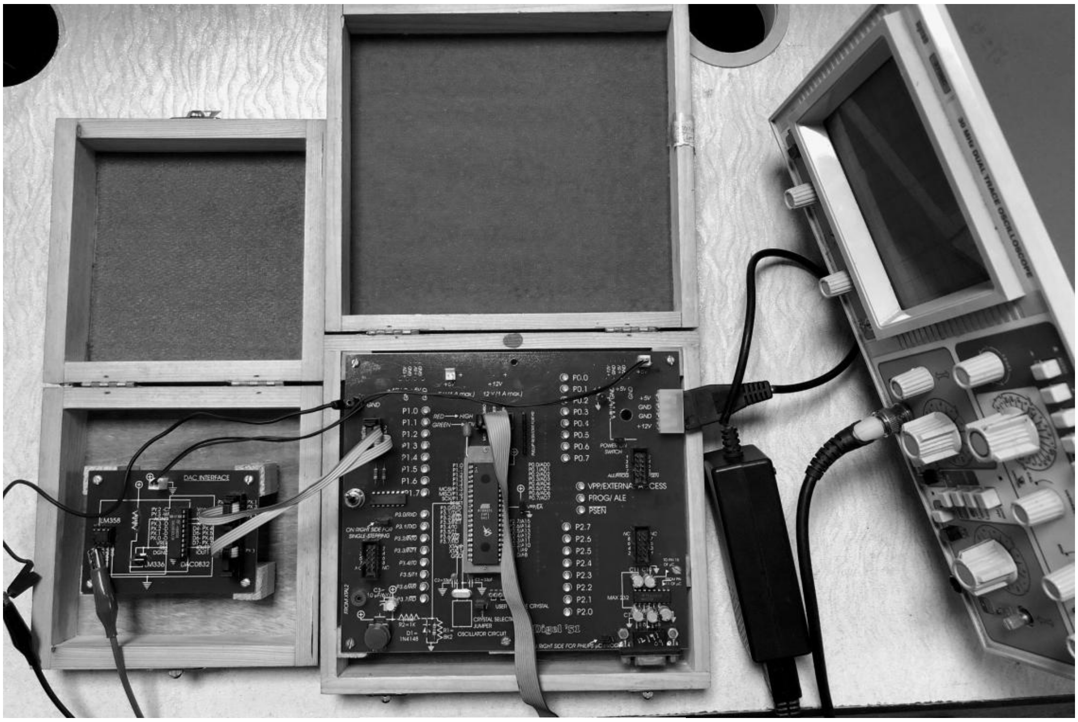


Fig 17.3 Practical Setup

c) Actual circuit used in laboratory

d) Actual Experimental set up used in laboratory

IX Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1	Microcontroller kit	Single board systems with 8K RAM,ROM memory with battery backup,16X4,16X2LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler,RS-232,USB, interfacing facility with built in power supply.	1 No.
2	Desktop PC	Loaded with open source IDE, simulation and program downloading software	1 No.
3	DAC (0808) trainer board	Suitable to interface 8051 board.	1 No

X Precautions to be Followed

1. Operate DAC chip as per specifications given in the datasheet otherwise damage may occur to the device.

XI Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop assembly program using Integrated Development Environment (IDE) or any other relevant software tool.
4. Debug program on IDE.
5. Execute program on IDE.
6. Create hex file for the above program.
7. Download hex code in EPROM/Flash memory of microcontroller.
8. Interface DAC 0808 IC to the microcontroller as per circuit diagram shown in fig 17.2
9. Observe Triangular waveform on CRO and draw the same pattern in observation mentioning Ton and Toff time.

SAMPLE PROGRAM 1: Write a program to generate triangular waveform using DAC.

STEP1- Algorithm

1. Make the Port used to Interface DAC as an output port.
2. Clear Accumulator.
3. Send 00H value to DAC
4. Increment value.
5. Compare current value with highest value 0FFh and send it to DAC till it reaches.
6. Decrement value.
7. Compare current value with lowest value 00H and send it to DAC till it reaches.
8. For repeat operation go to step3.
9. Stop.

Step 2-Flow Chart

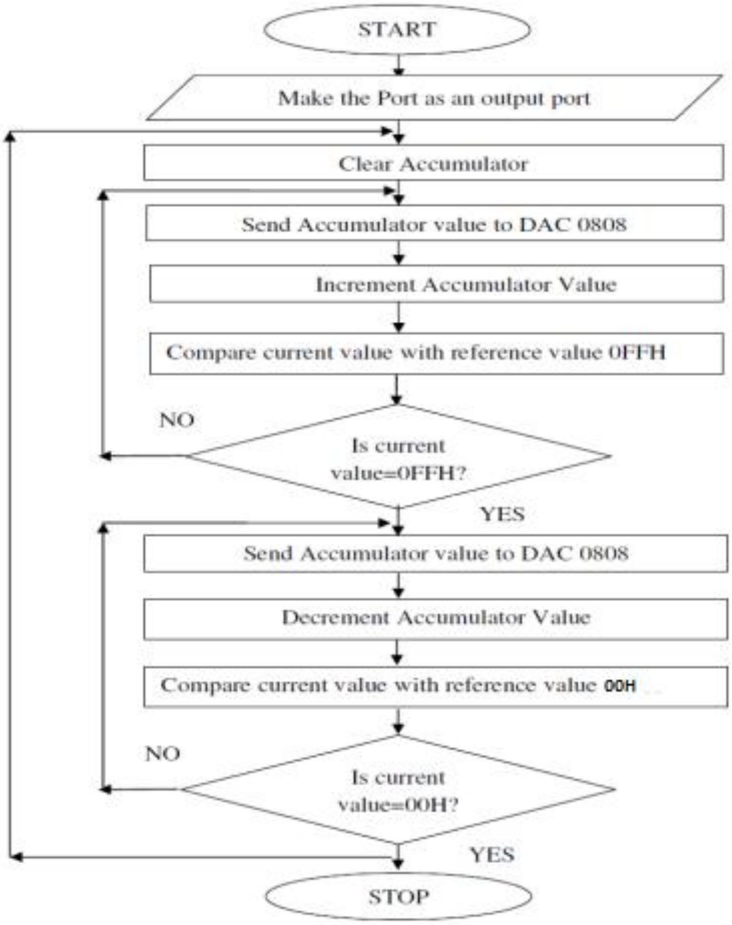

START
Make the Port as an output port
Clear Accumulator
Send Accumulator value to DAC 0808
Increment Accumulator Value
Compare current value with reference value 0FFH
NO
Is current value=0FFH?
YES
Send Accumulator value to DAC 0808
Decrement Accumulator Value
Compare current value with reference value 00H
NO
Is current value=00H?
YES
STOP

Fig 17.4 Flowchart to generate triangular waveform

Step 3-Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments
			ORG 0000h	
C:0x0000	7400	REPEAT:	MOV A,#00h	;clear A
C:0x0002	F590	INCR:	MOV P1,A	;send value to P1
C:0x0004	04		INC A	;increment value
C:0x0005	B4FFFA		CJNE A,#0FFh,INCR	;compare with highest value
C:0x0008	F590	DECR:	MOV P1,A	
C:0x000A	14		DEC A	;decrement value
C:0x000B	B400FA		CJNE A,#00h,DECR	;compare with lowest value
C:0x000E	80F0		SJMP REPEAT	;repeat
			END	

Problem statement for student: Develop assembly program to generate sawtooth waveform using DAC 0808

Step 1-Algorithm	**Step 2-Flowchart**

Step 3- Assembly Language Program:-

Memory Address	Hex Code	Label	Mnemonics	Comments

XII Resources Used

S. No.	Instrument /Components	Specification	Quantity
1.			
2.			
3.			
4.			
5.			

XIII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....

....

....

....

....

....

....

....

....

XIV Precautions Followed (use blank sheet provided if space not sufficient)

....

....

....

....

XX Observations for sample program (use blank sheet provided if space not sufficient)

XXI Results (Output of the Program)

....

....

....

....

....

....

XXII Interpretation of Results (Give meaning of the above obtained results)

....

....

....

....

XXIII Conclusions and Recommendation

....

....

....

....

XXIV Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. If Iref = 2mA and all the inputs to the DAC are high then find maximum current of DAC 0808 IC.
2. To generate a sine wave using DAC 0808 find decimal values representing magnitude of the sine of angles between 0 and 360 degrees.
   Refer Vout = 5V+ (5xsinθ).
3. If a switch SW is connected to port pin P0.0. Write a program to do the following.
   a) When SW=0 the DAC output gives a triangular waveform.
   b) When SW=1 the DAC output gives a staircase waveform.

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XIX References / Suggestions for further reading

1 https://en.wikipedia.org/wiki/Digital-to-analog_converter.
2 https://pdf1.alldatasheet.com/datasheet-pdf/view/512341/TI1/DAC0808.html
3 The 8051 Microcontroller and Embedded system Using Assembly and C- Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. Mckinlay- Pearson / Prentice Hall, , 2nd edition, Delhi,2008, ISBN 978-8177589030.

XX Assessment Scheme

	Performance indicators	Weightage
	Process related(15 Marks)	**60%**
1	Coding and Debugging ability	30%
2	Making connections of hardware	20%
3	Follow ethical practices.	10%
	Product related (10 Marks)	**40%**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition.	15%
6	Timely Submission of report, Answer to sample questions.	05%
	Total (25 Marks)	**100%**

Name of Team Members

1 …………………………......

2 ………………………….......

3 …………………………..

4 …………………………..

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No. 18: Interface stepper motor to microcontroller and rotate in clockwise and anti-clockwise direction at the given angles.

I Practical Significance

Different field applications require precise positioning, repeatability of movement in clockwise and anticlockwise direction with good accuracy. Stepper motors are controlled by microcontrollers in such areas like in computer peripherals, Business machines, process control and for making robots. This practical will help the students to develop skills to interface stepper motor to 8051 and rotate in clockwise and anticlockwise direction.

II Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Electronics and Telecommunication engineering knowledge to solve broad-based Electronics and Telecommunications engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics and Telecommunication engineering problems.
- **Engineering tools:** Apply relevant Electronics and Telecommunications technologies and tools with an understanding of the limitations.

III Competency and Practical Skills

This practical is expected to develop the following skills for the industry identified competency '**Maintain microcontroller based systems':**

- Select appropriate type of stepper motor for given application.
- Interface stepper motor with microcontroller.
- Apply full step or half step sequence as per requirement of movement.

IV Relevant Course Outcome(s)

- Maintain microcontroller used in different application.

V Practical Outcome

- Interface stepper motor to microcontroller and rotate in clockwise and anti-clockwise direction at the given angles.

VI Relevant Affective domain related Outcome(s)

- Follow safe practices.
- Maintain tools and equipment.
- Follow ethical practices.

VII Minimum Theoretical Background

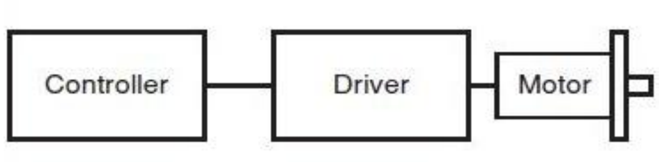


Fig 18.1 Basic Stepper motor system

Stepper motors convert electrical energy into precise mechanical motion. These motors rotate a specific incremental distance per each step. The number of steps executed controls the degree of rotation of the motor’s shaft

Table 18.1 Stepper Motor Step Angles

Step Angle	Steps per revolution
0.72	500
1.8	200
2.0	180
2.5	144
5	72

Stepping Modes:

Wave Drive (1 phase on)
Full Step Drive (2 phases on)
Half Step Drive (1 & 2 phases on)
Micro stepping (Continuously varying motor currents)

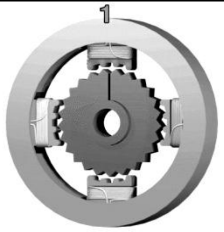
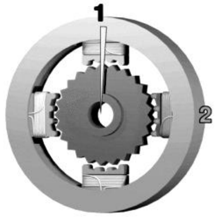
Table 18.2 Two Coil Excitation full Step Sequence

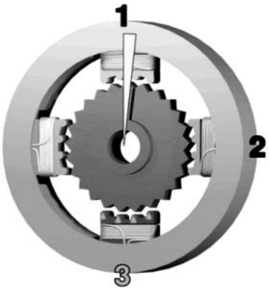
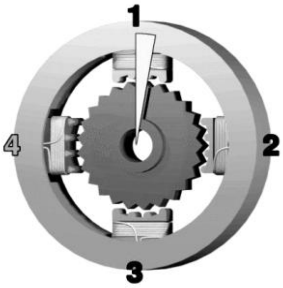
P2.3	P2.2	P2.1	P2.0	Port Pin	
Stator Windings of Stepper Motor					
A	B	C	D	Hex code	Direction
1	0	0	1	09 H	CW
1	1	0	0	0C H	
0	1	1	0	06 H	
0	0	1	1	03 H	CCW

Specifications: Permanent Magnet Stepper Motor

Part No-42BYG228-R
Drive system- Unipolar
Step angle-1.8° ±5%
Voltage & Current-12V DC at 400 mA
No of steps per revolution-200
Holding Torque-2000 g-cm
Ambient temperature--10°C to +55°C

Table 18.3 Stepper motor working

	
Frame 1: The top electromagnet (1) is turned on,	**Frame 2:** The top electromagnet (1) is turned

attracting the nearest teeth of the gear-shaped iron rotor. With the teeth aligned to electromagnet 1, they will be slightly offset from right electromagnet (2).	off, and the right electromagnet (2) is energized, pulling the teeth into alignment with it. This results in a rotation of 1.8° in this example.
	
Frame 3: The bottom electromagnet (3) is energized; another 1.8° rotation occurs.	**Frame 4:** The left electromagnet (4) is energized, rotating again by 1.8°. When the top electromagnet (1) is again enabled, the rotor will have rotated by one tooth position; since there are 50 teeth, it will take 200 steps to make a full rotation in this example.

VIII Practical Circuit diagram :

a) Sample circuit diagram

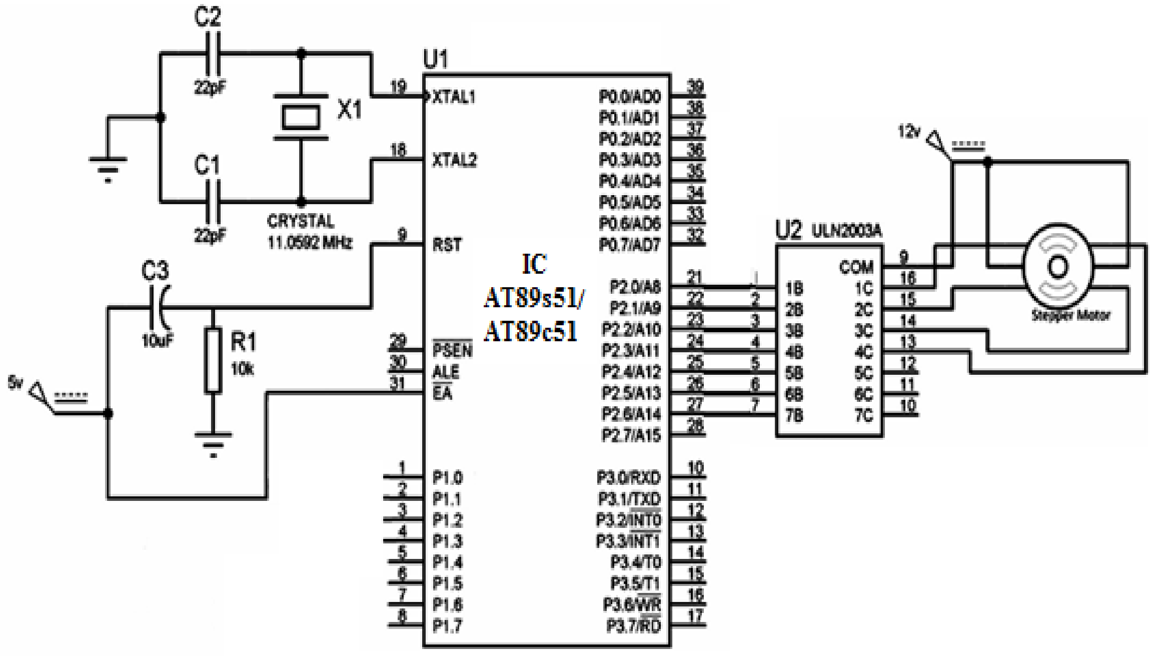


Fig 18.2 8051 connection to stepper motor

b) Practical setup

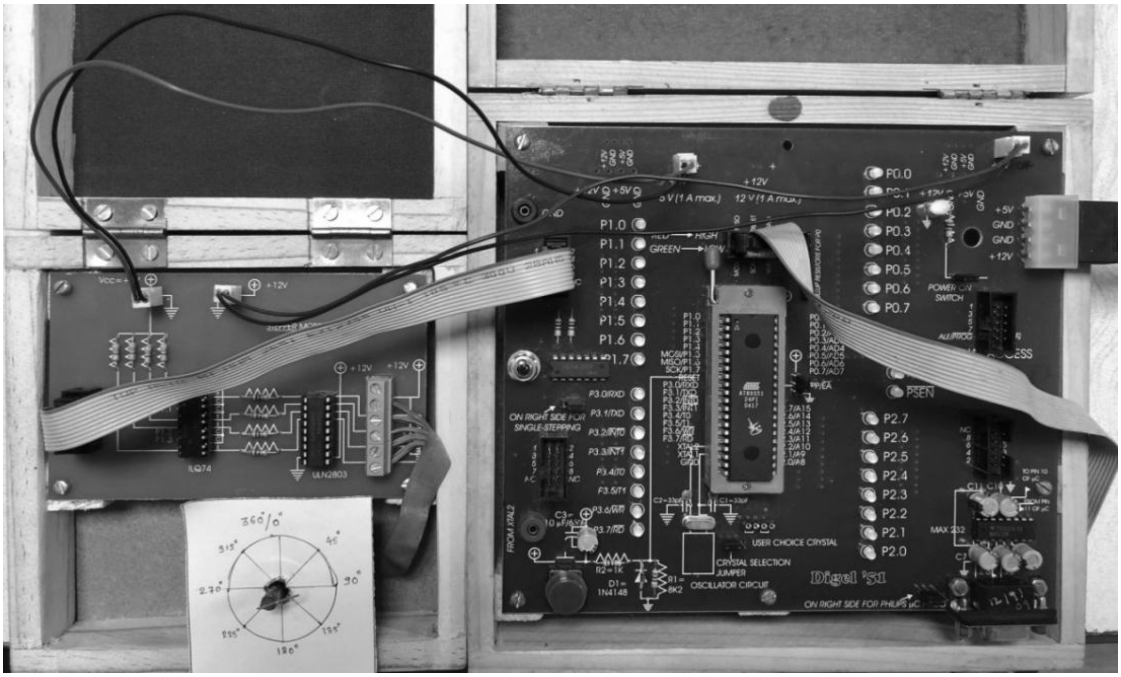


Fig 18.3 Practical Setup

c) Simulation diagram

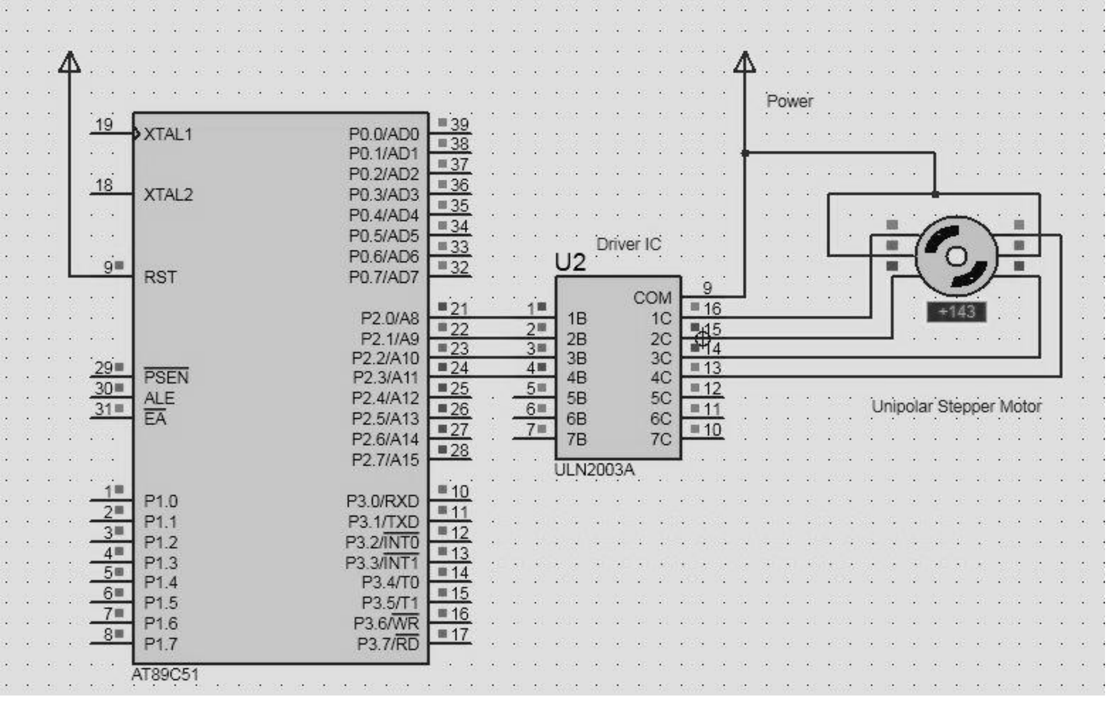


Fig 18.4 Simulation diagram

d) Actual circuit used in laboratory

e) Actual Experimental set up used in laboratory

IX Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1.	Microcontroller kit	Single board systems with 8K RAM,ROM memory with battery backup,16X4,16X2LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler,RS-232,USB, interfacing facility with built in power supply.	1 No.
2.	Desktop PC	Loaded with open source IDE, simulation and program downloading software	1 No.
3.	Stepper Motor Trainer	1.8° Step angle, 50/100 RPM Stepper motor with ULN 2003 Driver.	1 No.

X Precautions to be Followed

1. Use always driver circuit while interfacing stepper motor to microcontroller.

XI Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop assembly program using Integrated Development Environment (IDE) or any other relevant software tool.
4. Debug program on IDE.
5. Execute program on IDE.
6. Create hex file for the above program.
7. Download hex code in EPROM/Flash memory of the microcontroller.
8. Interface stepper motor to microcontroller as per circuit diagram shown in fig 18.5
9. Observe rotation of stepper motor and record in observation Table.

SAMPLE PROGRAM 1: Write a program to rotate stepper motor in clockwise direction by $180^{0.}$

Step 1- Algorithm

1. Make the Port used to Interface stepper motor as an output port.
2. Set register as counter R2 =25 for 100 steps i.e.180° rotation.
3. Set register as counter R3= 4 for full step code
4. Initialize pointer to table which is in code memory i.e. DPTR.
5. Clear accumulator.
6. Read data from code memory.
7. Send code to stepper motor.
8. Increment DPTR to access next memory location code.
9. Decrement R3 and check for zero. Is counter R3=0? NO- go to step 5 else go to next.
10. Decrement R2 and check for zero. Is counter R2=0? NO- go to step 3 else go to next.
11. Stop.

Step 2-Flow Chart

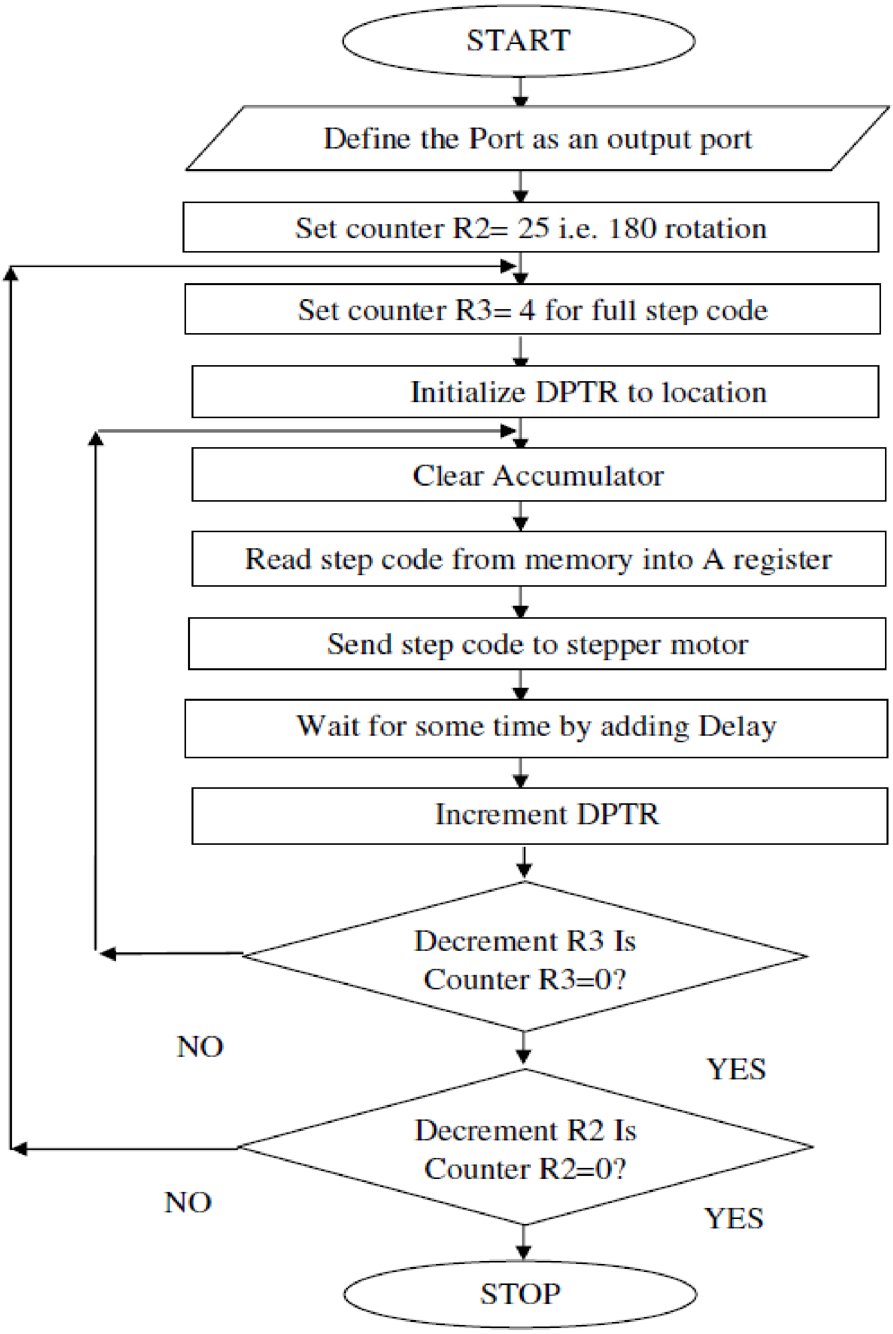

START
Define the Port as an output port
Set counter R2= 25 i.e. 180 rotation
Set counter R3= 4 for full step code
Initialize DPTR to location
Clear Accumulator
Read step code from memory into A register
Send step code to stepper motor
Wait for some time by adding Delay
Increment DPTR
Decrement R3 Is Counter R3=0?
NO
YES
Decrement R2 Is Counter R2=0?
NO
YES
STOP

Fig 18.5 Flowchart for stepper motor to rotate in clockwise 100 steps

Step 3-Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments
			ORG 0000H	
C:0x0000	759000		MOV P1,#00H	;Define port as output port
C:0x0003	7A19		MOV R2,#25	;Set register as counter of 25 for 180 rotation
C:0x0005	7B04	UP1:	MOV R3,#4	;set counter of 4 for full step code sequence
C:0x0007	900200		MOV DPTR,#TABLE	;load address of program memory into Data pointer
C:0x000A	E4	UP:	CLR A	;clear accumulator
C:0x000B	93		MOVC A,@A+DPTR	;read step code from memory into accumulator
C:0x000C	F590		MOV P1,A	;send step code to port
C:0x000E	1117		ACALL DELAY	;add delay
C:0x0010	A3		INC DPTR	;increment memory pointer to read next step code
C:0x0011	DBF7		DJNZ R3,UP	; decrement counter & jump to memory location labeled as UP if not equal to zero.
C:0x0013	DAF0		DJNZ R2,UP1	; decrement counter & jump to memory location labeled as UP1 if not equal to zero.
C:0x0015	80FE		SJMP $	;wait
C:0x0017	7C19	DELAY:	MOV R4,#25	;delay Subroutine
C:0x0019	7D64	L3:	MOV R5,#100	
C:0x001B	7E64	L2:	MOV R6,#100	
C:0x001D	DEFE	L1:	DJNZ R6,L1	
C:0x001F	DDFA		DJNZ R5,L2	
C:0x0021	DCF6		DJNZ R4,L3	
C:0x0023	22		RET	
			ORG 0050H	
C:0x0050		TABLE:	DB 09H, 0CH, 06H, 03H	; Step code stored at code memory starting at location 0050H onward.
			END	

Problem statement for student: Develop assembly program for stepper motor to rotate 90° in anticlockwise direction using rotate instruction. Assume step angle 1.8°

Step 1-Algorithm	**Step 2-Flowchart**

Step 3- Assembly Language Program

Memory Address	Hex Code	Label	Mnemonics	Comments

XII Resources Used

S. No.	Instrument /Components	Specification	Quantity
1.			
2.			
3.			
4.			
5.			

XIII Actual Procedure Followed (use blank sheet provided if space not sufficient)

....

....

....

....

....

....

....

....

....

....

XIV Precautions Followed (use blank sheet provided if space not sufficient)

....

....

....

....

XV Observations for sample program (use blank sheet provided if space not sufficient)

Step Code	Port Pin	Status - logic 1 (+5V) / logic 0 (0V)
9H	**P2.4 - P2.0**	
CH	**P2.4 - P2.0**	
6H	**P2.4 - P2.0**	
3H	**P2.4 - P2.0**	

XVI Results (Output of the Program)

....

....

....

....

XVII Interpretation of Results (Give meaning of the above obtained results)

....

....

....

....

XVIII Conclusions and Recommendation

....

....

....

....

XIX Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. Write specifications of 4 different stepper motors available in the market.
2. Calculate no of steps to move stepper motor 64 degree in clockwise direction if step angle is 2 degree, steps per revolution 180 & no of rotor teeth 45.
3. If a motor takes 90 steps to make one complete revolution, what is the step angle for this motor?.

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XX References / Suggestions for further reading

1. https://en.wikipedia.org/wiki/Stepper_motor.
2. https://www.jameco.com/Jameco/Products/ProdDS/155433.pdf
3. The 8051 Microcontroller and Embedded system Using Assembly and C- Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. Mckinlay- Pearson /Prentice Hall, , 2nd edition, Delhi,2008, ISBN 978-8177589030
4. https://www.youtube.com/watch?v=TWMai3oirnM

XXI Assessment Scheme

Performance Indicators		Weightage
Process related(15 Marks)		**60%**
1	Coding and Debugging ability	30%
2	Making connections of hardware	20%
3	Follow ethical practices.	10%
Product related (10 Marks)		**40%**
4	Correctness of algorithm/ Flow chart	20%
5	Relevance of output of the problem definition.	15%
6	Timely Submission of report, Answer to sample questions.	05%
Total (25 Marks)		**100%**

Name of Team Members

1 …………………………......

2 ………………………….......

3 …………………………..

4 ………………………….

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

List Of Laboratory Manuals Developed by MSBTE

First Semester:

1	Fundamentals of ICT	22001
2	English	22101
3	English Work Book	22101W
4	Basic Science (Chemistry)	22102
5	Basic Science (Physics)	22102

Second Semester:

1	Bussiness Communication Using Computers	22009
2	Computer Peripherals & Hardware Maintenace	22013
3	Web Page Design with HTML	22014
4	Applied Science (Chemistry)	22202
5	Applied Science (Physics)	22202
6	Applied Machines	22203
7	Basic Surveying	22205
8	Applied Science (Chemistry)	22211
9	Applied Science (Physics)	22211
10	Fundamental of Electrical Engineering	22212
11	Elements of Electronics Engineering	22213
12	Elements of Electrical Engineering	22215
13	Basic Electronics	22216
14	C Language programming	22218
15	Basic Electronics	22225
16	Programming in C	22226
17	Fundamental of Chemical Engineering	22231

Third Semester:

1	Applied Multimedia Techniques	22024
2	Advanced Serveying	22301
3	Highway Engineering	22302
4	Mechanics of Structures	22303
5	Building Construction	22304
6	Concrete Technology	22305
7	Strength Of Materials	22306
8	Automobile Engines	22308
9	Automobile Transmission System	22309
10	Mechanical Operations	22313
11	Technology Of Inorganic Chemicals	22314
12	Object Oriented Programming Using C++	22316
13	Data Structure Using ‘C’	22317
14	Computer Graphics	22318
15	Database Management System	22319
16	Digital Techniques	22320
17	Principles Of Database	22321
18	Digital Techniques & Microprocessor	22323
19	Electrical Circuits	22324
20	Electrical & Electronic Measurment	22325
21	Fundamental Of Power Electronics	22326
22	Electrical Materials & Wiring Practice	22328
23	Applied Electronics	22329
24	Electrical Circuits & Networks	22330
25	Electronic Measurments & Instrumentation	22333
26	Principles Of Electronics Communication	22334
27	Thermal Engineering	22337
28	Engineering Matrology	22342
29	Mechanical Engineering Materials	22343
30	Theory Of Machines	22344

Fourth Semester:

1	Hydraulics	22401
2	Geo Technical Engineering	22404
3	Chemical Process Instrumentation & Control	22407
4	Fluid Flow Operation	22409
5	Technology Of Organic Chemical	22410
6	Java Programming	22412
7	GUI Application Development Using VB.net	22034
8	Microprocessor	22415
9	Database Managment	22416
10	Electric Motors And Transformers	22418
11	Industrial Measurement	22420
12	Digital Electronic And Microcontroller Application	22421
13	Linear Integrated Circuits	22423
14	Microcontroller & Applications	22426
15	Basic Power Electronics	22427
16	Digital Communication Systems	22428
17	Mechanical Engineering Measurments	22443
18	Fluid Mechanics and Machinery	22445
19	Fundamentals Of Mechatronics	22048
20	Micro Project & Industrial Training Assessment Manual	22049

Fifth Semester:

1	Network Management & Administration	17061
2	Solid Modeling	17063
3	CNC Machines	17064
4	Behavioral Science(Hand Book)	17075
5	Behavioral Science (Assignment Book)	17075
6	Windows Programming using VC++	17076
7	Estimation and Costing	17501
8	Public Health Engineering	17503
9	Concrete Technology	17504
10	Design of Steel Structures	17505
11	Switchgear and Protection	17508
12	Microprocessor & Application	17509
13	A.C. Machines	17511
14	Operating System	17512
15	Java Programming	17515
16	System Programming	17517
17	Communication Technology	17519
18	Hydraulic & Pneumatics	17522
19	Advanced Automobile Engines	17523
20	Basic Electrical & Electronics	17524
21	Measurement and Control	17528
22	Power Engineering	17529
23	Metrology & Quality Control	17530
24	Computer Hardware & Networking	17533
25	Microcontroller	17534
26	Digital Communication	17535
27	Control System & PLC	17536
28	Audio Video Engineering	17537
29	Control System	17538
30	Industrial Electronics and applications	17541
31	Heat Transfer Operations	17560
32	Chemical Process Instrumentation & control	17561

Sixth Semester:

1	Solid Modeling	17063
2	Highway Engineering	17602
3	Contracts & Accounts	17603
4	Design of R.C.C. Structures	17604
5	Industrial Fluid Power	17608
6	Design of Machine Elements	17610
7	Automotive Electrical and Electronic Systems	17617
8	Vehicle Systems Maintenance	17618
9	Software Testing	17624
10	Advanced Java Programming	17625
11	Mobile Computing	17632
12	System Programing	17634
13	Testing & Maintenance of Electrical Equipments	17637
14	Power Electronics	17638
15	Illumination Engineering	17639
16	Power System Operation & Control	17643
17	Environmental Technology	17646
18	Mass Transfer Operation	17648
19	Advanced Communication System	17656
20	Mobile Communication	17657
21	Embedded System	17658
22	Process Control System	17663
23	Industrial Automation	17664
24	Industrial Drives	17667
25	Video Engineering	17668
26	Optical Fiber & Mobile Communication	17669
27	Therapeutic Equipment	17671
28	Intensive Care Equipment	17672
29	Medical Imaging Equipment	17673

Pharmacy Lab Manual

First Year:

1	Pharmaceutics - I	0805
2	Pharmaceutical Chemistry - I	0806
3	Pharmacognosy	0807
4	Biochemistry and Clinical Pathology	0808
5	Human Anatomy and Physiology	0809

Second Year:

1	Pharmaceutics - II	0811
2	Pharmaceutical Chemistry - II	0812
3	Pharmacology & Toxicology	0813
4	Hospital and Clinical Pharmacy	0816

HEAD OFFICE

Secretary,
Maharashtra State Board of Technical Education
49, Kherwadi, Bandra (East), Mumbai - 400 051
Maharashtra (INDIA)
Tel: (022)26471255 (5 -lines)
Fax: 022 - 26473980
Email: -secretary@msbte.com
Web -www.msbte.org.in

REGIONAL OFFICES:

MUMBAI

Deputy Secretary (T),
Mumbai Sub-region,
2nd Floor, Govt. Polytechnic Building,
49, Kherwadi, Bandra (East)
Mumbai - 400 051
Phone: 022-26473253 / 54
Fax: 022-26478795
Email: rbtemumbai@msbte.com

PUNE

Deputy Secretary (T),
M.S. Board of Technical Education,
Regional Office,
412-E, Bahirat Patil Chowk,
Shivaji Nagar, Pune
Phone: 020-25656994 / 25660319
Fax: 020-25656994
Email: rbtepn@msbte.com

NAGPUR

Deputy Secretary (T),
M.S. Board of Technical Education
Regional Office,
Mangalwari Bazar, Sadar, Nagpur - 440 001
Phone: 0712-2564836 / 2562223
Fax: 0712-2560350
Email: rbteng@msbte.com

AURANGABAD

Deputy Secretary (T),
M.S. Board of Technical Education,
Regional Office,
Osmanpura, Aurangabad -431 001.
Phone: 0240-2334025 / 2331273
Fax: 0240-2349669
Email: rbteau@msbte.com