I

Name ______________________________

Roll No. ______________ Year 20 ________ 20 ______

Exam Seat No. ______________________________

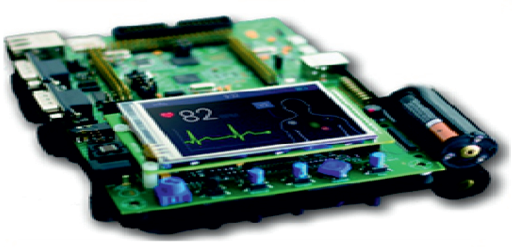
ELECTRONICS GROUP | SEMESTER - V | DIPLOMA IN ENGINEERING AND TECHNOLOGY

A LABORATORY MANUAL

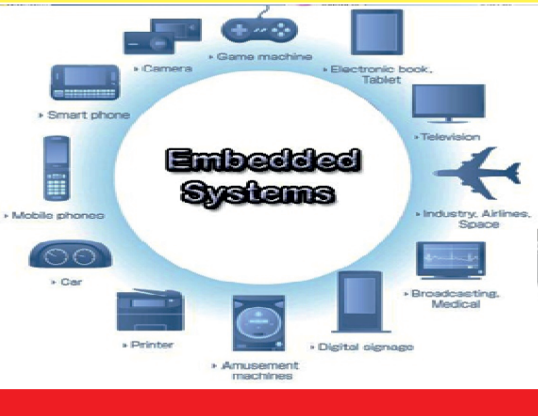
FOR

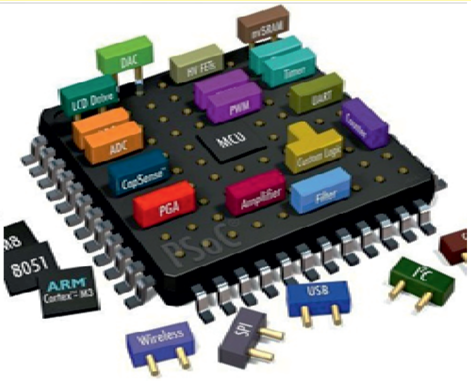
EMBEDDED SYSTEMS

(22532)



User Interface
Input Variables
Embedded Computer
Software
Hardware
Output Variables
Link to Other Systems





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

Embedded Systems

(22532)

Semester-V

(DE/IE/IS/EJ/ET/EN/EX/EQ/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 May,2019)

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 **Embedded Systems(22532)** 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.

In the rapidly growing digital world, role of embedded systems is increasingly vital in various domains such as industrial and home automation, entertainment systems, medical equipments and many more. The core of all such system is powered by electronic hardware and associated software. It is therefore evident to impart the knowledge of the related technology and hands on skills to develop and maintain electronics hardware based embedded systems.

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 Embedded Systems.’ are expected to be developed in students by undertaking the practicals of this laboratory manual.**

1. Identify the electronic component of the system.
2. Connect the Development board with computer.
3. Use of IDE tools such as Editor, Compiler, Debugger and Simulator.
4. Select the appropriate IC with help of data sheet.
5. Interface Various devices to given microcontroller.
6. Execution and Interpret the result.
7. Simulate the program execution.
8. Test and Trouble shoot the given Interface circuit.

Practical- Course Outcome matrix

Course Outcomes (COs)

a. Select the relevant microcontrollers for various industrial applications.
b. Use ‘Embedded C’ programming language to maintain embedded systems.
c. Interpret the communication standards of embedded systems.
d. Develop basic applications using embedded systems.
e. Interpret features of Real Time Operating System.

Pro. No.	Practical Outcomes (PrO)	CO a.	CO b.	CO c.	CO d.	CO e.
1.	Identify the pins of 8051 and AVR microcontrollers.	√	-	-	-	-
2.	Identify the pins and features of PIC microcontrollers.	√	-	-	-	-
3.	Identify the features of ARM microcontroller on the basis of IC number.	√	-	-	-	-
4.	Use Integrated development environment tool for developing embedded ‘C’ programs (Using MicroProC/Keil).	-	√	-	-	-
5.	Execute the ‘C’ program to perform following arithmetic operations on 8-bit data: addition, subtraction, multiplication and division.	-	√	-	-	-
6.	Develop and Test the ‘C’ program to perform following arithmetic operations on 16-bit data: addition, subtraction.	-	√	-	-	-
7.	Develop and Test the ‘C’ program to perform data transfer from source to destination (Use internal data memory locations).	-	√	-	-	-
8.	Interface RS232 connector to PC using MAX232 IC.	-	-	√	-	-
9.	Develop and test the ‘C’ program to turn on LED (S) with key (S) press.	-	-	-	√	-
10.	Interface 89C51/AVR microcontroller and write the ‘C’ program to display numbers from 0 to 9 on 7-segment display with specified delay.	-	-	-	√	-
11.	Interface 89C51/AVR microcontroller and write C program to display string on given 16 x 2 LCD.	-	-	-	√	-
12.	Interface 89C51/AVR microcontroller and write ‘C’ language program to read key code from 4 x 4 matrix keyboard and LCD display.	-	-	-	√	-

Course Outcomes (COs)

a. Select the relevant microcontrollers for various industrial applications.
b. Use ‘Embedded C’ programming language to maintain embedded systems.
c. Interpret the communication standards of embedded systems.
d. Develop basic applications using embedded systems.
e. Interpret features of Real Time Operating System.

Pro. No.	Practical Outcomes (PrO)	CO a.	CO b.	CO c.	CO d.	CO e.
13.	Interface 89C51/AVR microcontroller and write C program to convert analog signal into digital form using given 8 bit ADC and store the converted digital data in memory.	-	-	-	√	-
14.	Interface 89C51 and write C program to generate square and saw tooth waveforms using given 8 bit DAC.	-	-	-	√	-
15.	Interface 89C51 /AVR microcontroller and write C program to rotate stepper motor with different speeds in clockwise and counter clockwise direction.	-	-	--	√	-
16.	Interface 89C51 and write C program to observe the real time status of control signals for the given waveform generated using DAC (Use IDE tool MicroProC/Keil).	-	-	-	-	√

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. **Development Board for practical 1, 2, 3 should be made available for illustration purpose.**
14. Practical No. 1, 2, 3 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

11 *	Interface 89C51/AVR microcontroller and write C program to display string on given 16 x 2 LCD.	91					
12 *	Interface 89C51/AVR microcontroller and write 'C' language program to read key code from 4 x 4 matrix keyboard and LCD display.	105					
13 *	Interface 89C51/AVR microcontroller and write C program to convert analog signal into digital form using given 8 bit ADC and store the converted digital data in memory.	117					
14 *	Interface 89C51 and write C program to generate square and sawtooth waveforms using given 8 bit DAC.	129					
15 *	Interface 89C51 /AVR microcontroller and write C program to rotate stepper motor with different speeds in clockwise and counter clockwise direction.	142					
16	Interface 89C51 and write C program to observe the real time status of control signals for the given waveform generated using DAC (Use IDE tool MicroProC/Keil).	155					
Total Marks							

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

Practical No.1: Identify the various pins of 8051 and AVR Microcontrollers

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 pins of 8051 and AVR Microcontroller.

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.
  Function effectively as a 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 Embedded Systems':**

- Identify the functions of various pins of 8051 and AVR Microcontroller.
- Understand the significance of the pins and their use in device.

IV Relevant Course Outcome

- Select the relevant microcontrollers for various industrial applications.

V Practical Outcome

- Identify the pins of 8051 and AVR Microcontrollers.

VI Relevant Affective domain related Outcome(s)

- Demonstrate working as a leader/a team member.
- Practice energy conservation
- Follow ethical practices.

VII Minimum Theoretical Background

Microcontroller is a single chip microcomputer made through VLSI fabrication. **8051 Microcontroller:** 8051 microcontroller of the MCS-51 family is introduced by Intel

Corporation. It has inbuilt components such as CPU, internal RAM and ROM, timers/counters, serial ports, interrupts and I/O ports.

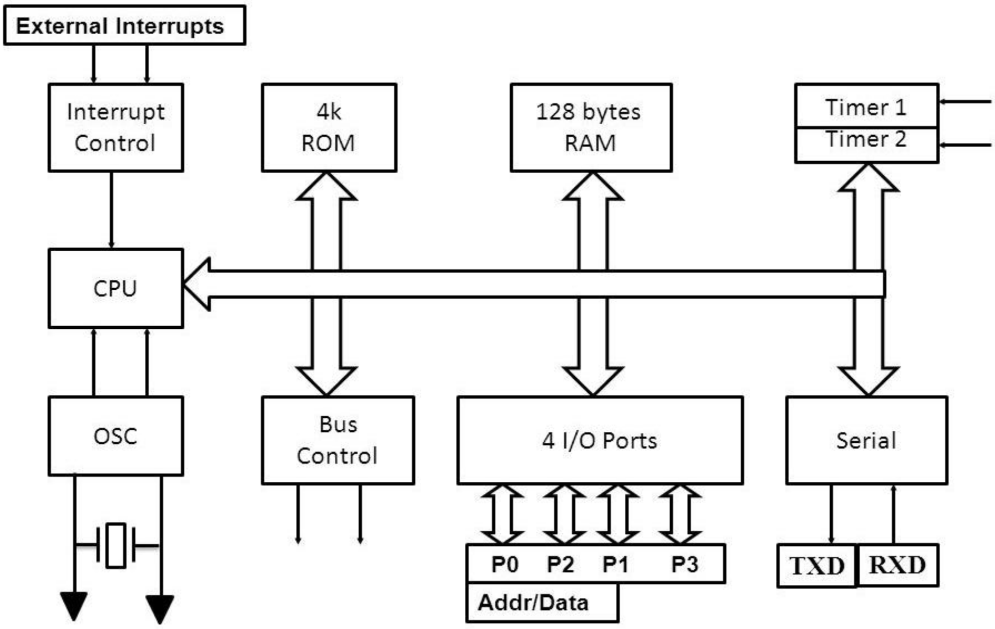


Figure1.1: Block diagram of 8051

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

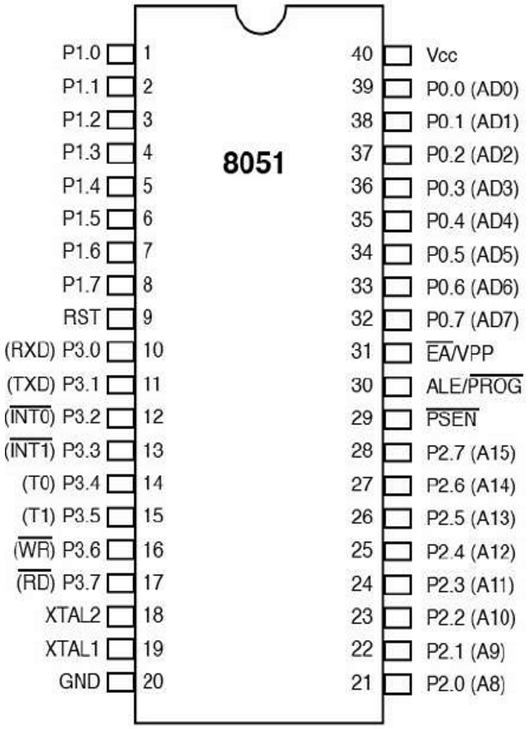


Figure 1.2: Pin diagram of 8051

Table No 1.1: Pin Functions of 8051

Pin number	Pin Name	Pin function
1 to 8	Port 1 pins	Serves as I/O port
9	RESET pin	Used to reset the microcontroller to its initial values
10 to 17	Port 3 Pins	Serves some functions like interrupts, timer input, control signals, serial communication signals RxD and TxD,
18 & 19	X1, X2	Used for interfacing an external crystal to get the system clock.
20	Gnd	Ground Pin
21 to 28	Port 2 pins	Serves as I/O port. Higher order address bus signals are also multiplexed using this port.
29	$\overline{PSEN}$	Program Store Enable used to read a signal from the external program memory.
30	$\overline{EA}$	External Access input used to enable/disable the external memory interfacing.
31	ALE	Address Latch Enable used to demultiplex the address-data signal of port.
32 to 39	Port 0 pins	Serves as I/O port. Lower order address and data bus signals are multiplexed using this port.
40	Vcc	Used to provide power supply to the circuit

Microcontroller Development board:

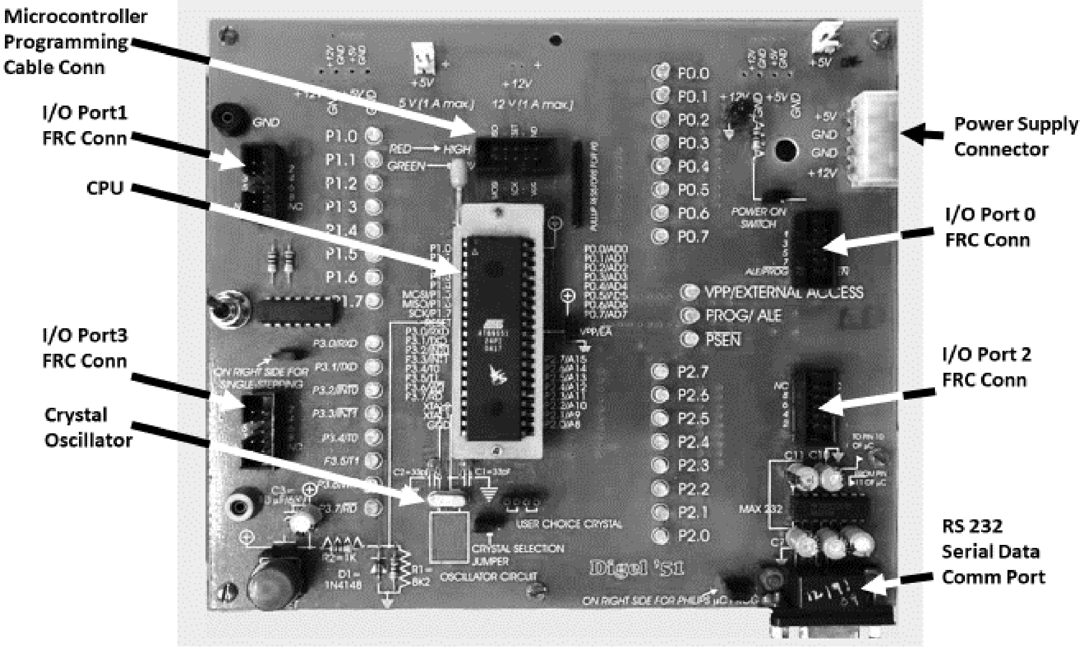


Figure 1.3: 8051 Development Board

AVR Microcontroller:

AVR was developed in the year 1996 by Atmel Corporation. AVR derives its name from its developers and stands for Alf-EgilBogen VegardWollan RISC microcontroller, also known as Advanced Virtual RISC.

The AVR microcontrollers are based on the advanced RISC architecture and consist of 32 x 8-bit general purpose working registers. Within one single clock cycle, AVR can take inputs from two general purpose registers and put them to ALU for carrying out the requested operation, and transfer back the result to an arbitrary register. The ALU can perform arithmetic as well as logical operationsover the inputs from the register or between the register and a constant. Single register operations like taking a complement can also be executed in ALU.

AVR follows Harvard Architecture format in which the processor is equipped with separate memories and buses for Program and the Data information. Here while an instruction is being executed, the next instruction is pre-fetched from the programmemory.

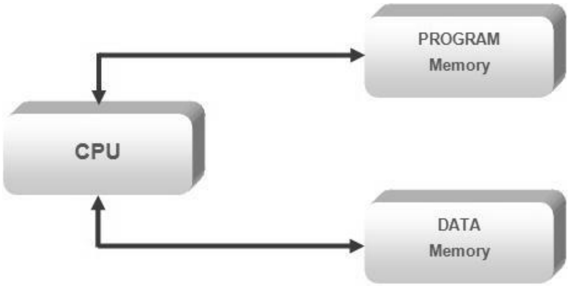


Figure 1.4: AVR Harvard Model

Pin Diagram of AVR Microcontroller:

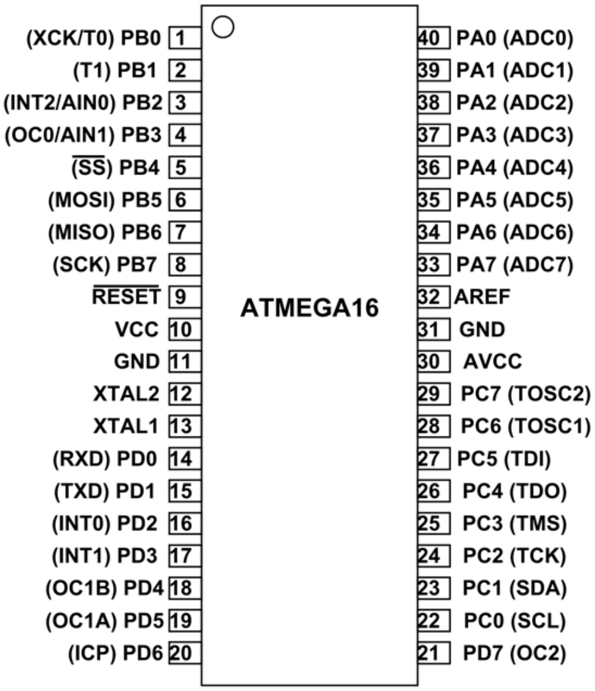


Figure 1.5: Pin Diagram of AVR Microcontroller

Table No 1.2: Pin functions of AVR Microcontroller

Pin number	Pin Name	Pin function
1-8	Port B: PB7…PB0	8-bit bi-directional I/O port
9	$\overline{RESET}$	Generate a reset to the processor
10	Vcc	Digital supply voltage
11,31	GND	Ground Pin
12	XTAL1:	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
13	XTAL2:	Output from inverting oscillator amplifier
14-21	Port D PD7…PD0	• 8-bit bi-directional I/O port
22-29	Port C PC7…PC0	• 8-bit bi-directional I/O port • If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs
30	AVCC	Supply voltage pin for Port A and the A/D Converter.
32	Aref:	Aref is the analog reference pin for the A/D Converter
33-40	Port A: PA7…PA0	• Serves as the analog inputs to the A/D Converter • Serves as an 8-bit bi-directional I/O port

AVR Microcontroller Development Board:

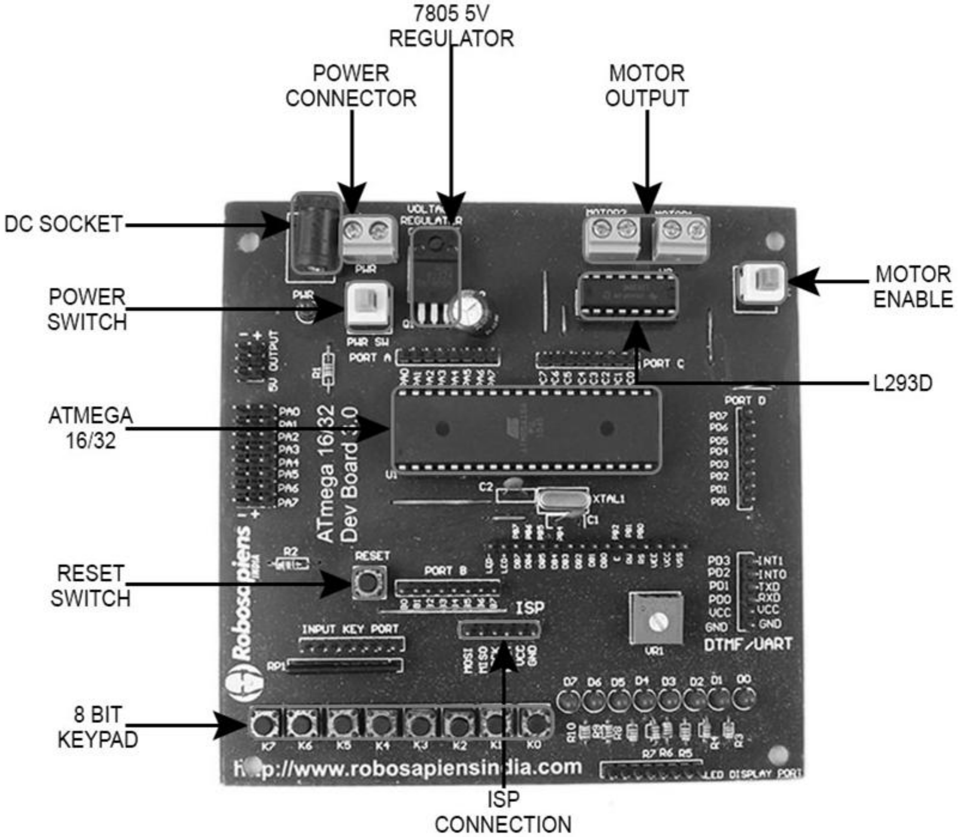


Figure 1.6: AVR Development Board

VIII Resources required

Sr. No.	Instrument /Components	Specification	Quantity
1.	8051 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.	AVR Microcontroller kit	Single Board system with 1KB SRAM, 512 Bytes EPROM, SPI Interface, Onboard 8 Keypad, Onboard ISP, Onboard L293D Motor driver, on board DC power supply.	1 No.

IX Precautions to be followed

1. Do not power up development board when identifying pins.
2. Refer the Data Sheets for the given microcontroller.

X Resources used

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

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

....

....

....

....

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

Observe Pin out Diagram for AVR Microcontroller and list various pins and write their functions.

Sr. No.	Pins	Function
1.	AVcc	
2.	Aref	
3.	XTAL1 and XTAL2	
4.	RxD	
5.	TxD	

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

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

XIV 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. Compare the development board features of 8051 and AVR:
   a) Size of RAM
   b) Size of ROM
   c) SPI Interface
   d) ISP availibity
2. List functions of following pins of 8051: ALE, $\overline{EA}$, $\overline{PSEN}$
3. List the I/O ports available in AVR micro controller.

[Space for Answers]

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

....

....

....

....

....

XV References / Suggestions for further Reading

1. http://www.keil.com
2. https://en.wikipedia.org/wiki/Intel_MCS-51
3. http://www.avr-tutorials.com/
4. https://en.wikipedia.org/wiki/AVR_microcontrollers
5. https://www.keil.com/dd/docs/datashts/atmel/at89s51_ds.pdf

XVI Assessment Scheme

Performance Indicators		Weightage
Process related: 15 Marks		**60%**
1	Identifying the development board	20%
2	Identifying components on developer kit	30%
3	Follow ethical practices.	10%
Product related: 10 Marks		**40%**
4	Correct pin functions	20%
5	Answer to sample questions.	15%
6	Timely Submission of report.	05%
Total (25 Marks)		**100 %**

Names of student Team Member

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

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

Practical No. 2: Identify the pins and features of PIC microcontrollers

I Practical Significance

There are various categories of microcontroller available as per applications. This practical will help the students to develop skills to categorize the given microcontroller on the basis of IC number and to determine the pin functions

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 Embedded Systems':**

- Identify the pin functions
- Determine the category of given microcontroller

IV Relevant Course Outcome

- Select the relevant microcontroller for various industrial applications.

V Practical Outcome

- Identify the pins and features of PIC microcontrollers

VI Relevant Affective domain related Outcome(s)

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

VII Minimum Theoretical Background

PIC microcontroller was developed in the year 1993 by microchip technology. The term PIC stands for Peripheral Interface Controller. Initially this was developed for supporting PDP computers to control its peripheral devices, and therefore, named as a peripheral interface device. These microcontrollers are very fast and easy to execute a program compared with other microcontrollers. PIC Microcontroller architecture is based on Harvard architecture. PIC microcontrollers are very popular due to their ease of programming, wide availability, easy to interfacing with other peripherals, low cost, large user base and serial programming capability (reprogramming with flash memory), etc.

PIC Microcontroller Nomenclature:
Example: PIC 16F877A

- PIC: Personal/Peripheral Interface Controller.
- 16: Series of PIC microcontrollers
- F: Flash Memory
- 877: Series No.
- A: Advanced version of PIC-F16877

Features of PIC Microcontroller:

1. It has RISC (reduced instruction set computer) architecture.
2. It has ROM (read only memory) of size 2M (megabytes).
3. It has On chip program (code) ROM (read only memory) in the form of flash memory.
4. It has RAM (random access memory) between 256 bytes to 4096 bytes.
5. It has Data EEPROM (Electrical Erasable Programmable Read Only Memory).
6. It include Timers.
7. It has ADC (analog to digital converter).
8. It include USART PROTOCOL for PC communication.
9. It contains I/O port between 16 to 72 pins.
10. All I/O port register are bit accessible and port accessible both.
11. It include SPI PROTOCOL and I2C PROTOCOL for memory communication.
12. There is 2-stage pipeline in PIC18F458.

Pin diagram of PIC Microcontroller:

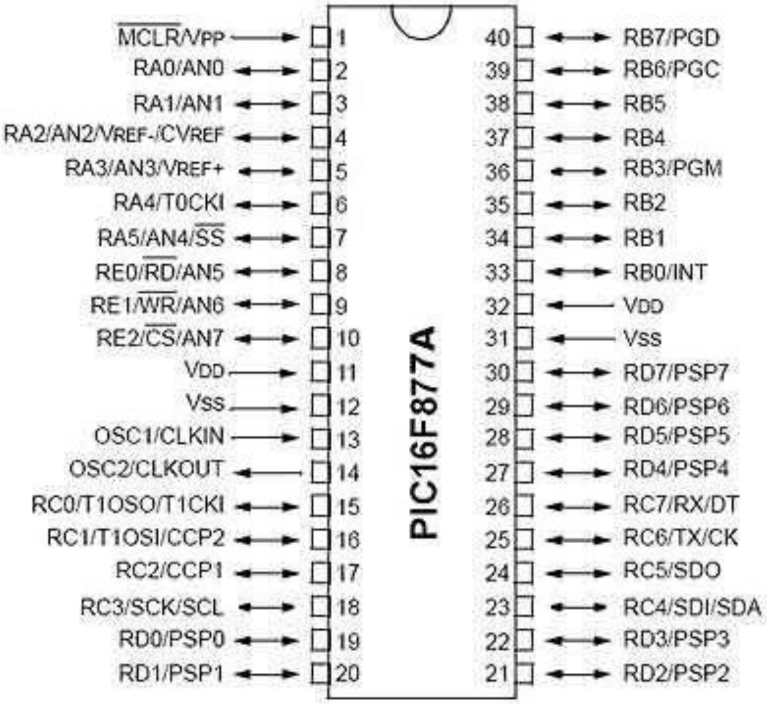


Figure 2.1: Pin Diagram of PIC Microcontroller

Table No 2.1 Pin Functions of PIC Microcontroller

Pin no.	Pin Name	Pin function
1	MCLR	Master clear pin of this IC. It resets the microcontroller
2	RA0/AN0:	Used as an analog pin AN0
3	RA1/AN1:	Analog input 1
4	RA2/AN2/Vref-	Analog input2. Or negative analog reference voltage can be given to it
5	RA3/AN3/Vref+	Analog input 3. Or can act as the analog positive reference voltage
6	RA0/T0CKI	To timer0, act as the clock input pin, the type of output is open drain.
7	RA5/SS/AN4	Analog input 4 and can be used as the slave select for that port.
8	RE0/RD/AN5	PORT E bidirectional input output port or can be the analog input 5 or for parallel slave port it can act as a ‘read control’ pin which will be active low.
9	RE1/WR/AN6	Analog input 6. And for the parallel slave port it can act as the ‘write control’ which will be active low.
10	RE2/CS’/AN7	Analog input 7, or for the parallel slave port it can act as the ‘control select’ which will also be active low just like read and write control pins.
11,32	VDD	These two pins are the positive supply for the input/output and logic pins. Both of them should be connected to 5V.
12,31	VSS	Ground reference for input/output and logic pins.
13	OSC1/CLKIN	Oscillator input or the external clock input pin
14	OSC2/CLKOUT	Oscillator output pin. A crystal resonator is connected between pin 13 and 14 to provide external clock to the microcontroller.
15	RC0/T1OCO/T1CKI	PORTC bidirectional input output port., clock input of timer 1 or the oscillator output of timer 2.
16	RC1/T1OSI/CCP2	Oscillator input of timer 1 or the capture input/compare 2 output/ PWM 2 output.
17	RC2/CCP1	Capture 1 input/ compare 1 output/ PWM 1 output.
18	RC3/SCK/SCL	Output for SPI or I2C modes and can be the input/output for synchronous serial clock.
23	RC4/SDI/SDA	SPI data in pin. Or in I2C mode it can be data input/output pin
24	RC5/SDO	Data out of SPI in the SPI mode
25	RC6/TX/CK	Synchronous clock or USART Asynchronous transmit pin.
26	RC7/RX/DT	Synchronous data pin or the USART receive pin
19-30	Port D	Bidirectional input and output port.
33-40	PORT B:	PORTB pins. Out of which RB0 can be used as the external interrupt pin and RB6 and RB7 can be used as in-circuit debugger pins

PIC Development Board [PIC 16Fxx Board]:

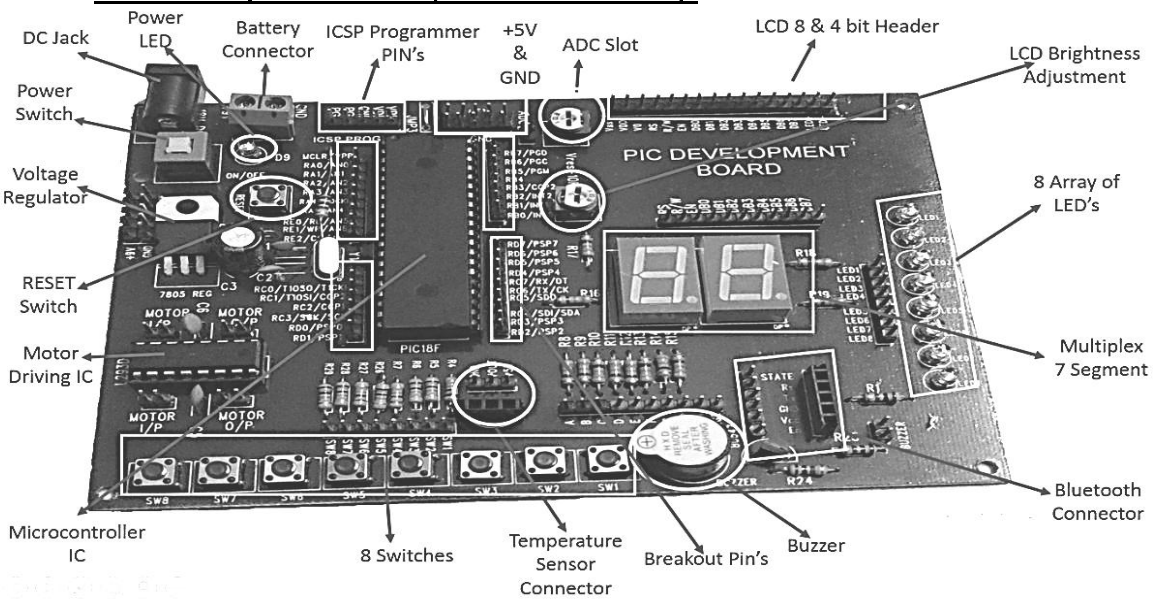


Figure 2.2: PIC Development Board

(Courtesy www.microchip.com)

VIII Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1.	PIC Microcontroller kit	Single board system with 4X4 matrix keyboard ,16X2LCD display, two analog inputs, seven segment display, RS-232 cable,USB, interfacing facility with built in power supply, LM 35 temperature sensor, USART, SPI EEPROM.	1 No.

IX Precautions to be Followed

1. Do not power up development board when identifying pins.
2. Refer the Data Sheets for the given microcontroller.

X Resources Used:

Sr. No.	Instrument /Components	Specification	Quantity
1			
2			

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

....

....

....

....

....

XII Observations

Observe Pin out Diagram and list various pins and write their functions

Sr No	Pins	Functions
1.	RA0/AN0	
2.	RC7/RX/DT	
3.	OSC1/CLKIN	
4.	RE2/CS’/AN7	

XIII Conclusions and Recommendation

....

....

....

....

XIV 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 significance of PIC microcontroller over 8051 microcontroller.
2. List out the different available PIC micro-controller ICs and also their comparative features.

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XV References / Suggestions for further reading

1. http://www.keil.com
2. http://www.nskelectronics.com/pic_development_board.html
3. http://microcontrollerslab.com/pic-microcontroller-development-boards-features/
4. https://ww1.microchip.com/downloads/en/devicedoc/35007b.pdf

XVI Assessment Scheme

Performance Indicators		Weightage
Process related: 15 Marks		**60%**
1	Identifying Development Board	20%
2	Identifying components on development board.	30%
3	Follow ethical practices.	10%
Product related: 10 Marks		**40%**
4	Correct pin functions	20%
5	Answer to sample questions	15%
6	Timely Submission of report.	05%
Total (25 Marks)		**100 %**

Names of student Team Member

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

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

Practical No. 3: Identify the features of ARM microcontrollers on the basis of IC number

I Practical Significance

There are various categories of microcontroller available as per applications. The experiment aims to categorize the given microcontroller on the basis of IC number and to understand functional fundamental of ARM 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.
- **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 Embedded Systems':**

1. Identify the pin functions
2. Determine the category of given microcontroller

IV Relevant Course Outcome

- Select the relevant microcontroller for various industrial applications.

V Practical Outcome

- Identify the features of ARM microcontrollers on the basis of IC number.

VI Relevant Affective domain related Outcome(s)

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

VII Minimum Theoretical Background

The ARM stands for Advanced RISC machine and it is a 32-bit reduced instructions set computer (RISC) microcontroller. It was first introduced by the Acron computers' organization in 1987. The ARM is a family of the microcontroller developed by the different manufacturers such as ST microelectronics, Motorola and so on.

RISC processors are designed to perform a smaller number of types of computer instructions so that they can operate at a higher speed, performing more millions of instructions per second (MIPS).

ARM processors are extensively used in consumer electronic devices such as smartphones, tablets, multimedia players and other mobile devices, such as wearables. Because of their reduced instruction set, they require fewer transistors, which enables a smaller die size for the integrated circuitry (IC). The ARM processor's smaller size, reduced complexity and lower power consumption makes them suitable for increasingly miniaturized devices.

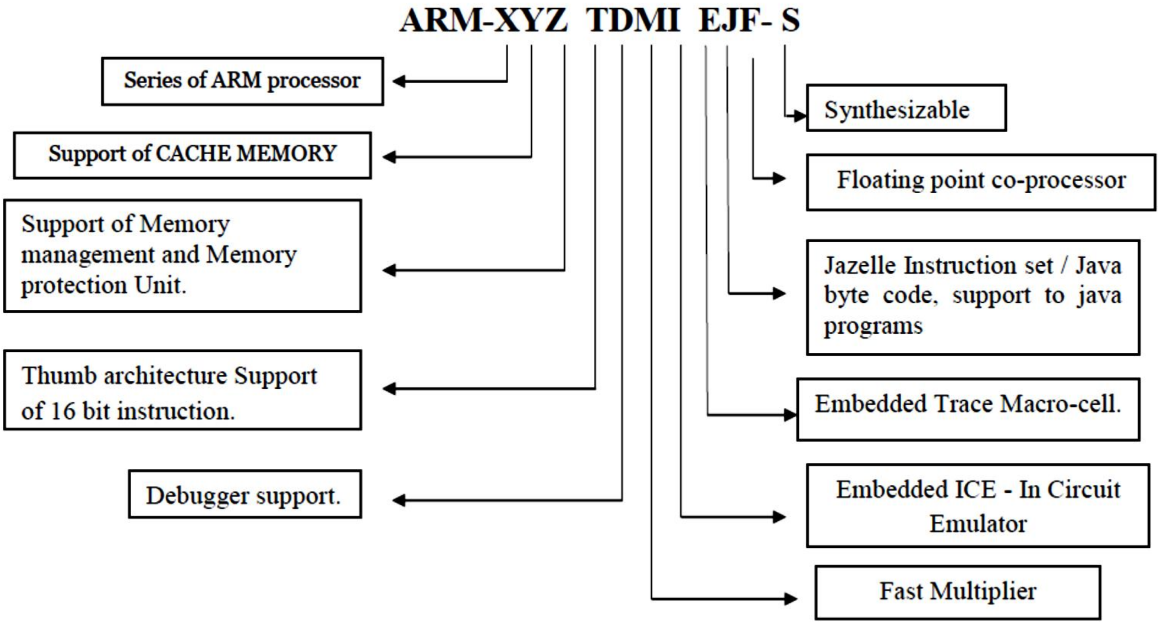

ARM-XYZ TDMI EJF- S
Series of ARM processor
Support of CACHE MEMORY
Support of Memory management and Memory protection Unit.
Thumb architecture Support of 16 bit instruction.
Debugger support.
Synthesizable
Floating point co-processor
Jazelle Instruction set / Java byte code, support to java programs
Embedded Trace Macro-cell.
Embedded ICE - In Circuit Emulator
Fast Multiplier

Figure 3.1: ARM Processor Nomenclature

Example: ARM7 TDMI

Features of ARM Microcontroller:

1. It is a RISC Controller
   - 32-bit high performance CPU
   - 3-stage pipeline and compact one
2. It has THUMB-2 technology
   - Optimal merges of 16/32 bit instructions
   - High performance
3. It supports tools and RTOS and It has core Sight debug and trace
   - JTAG or 2-pin serial wire debug connection
   - Support for multiple processors
4. Low power Modes
   - It supports sleep modes
   - Control the software
   - Multiple power domains
5. Nested vectored interrupt controller (NVIC)
   - Low latency, low jitter interrupts response
   - No need for assembly programming

ARM Microcontroller Development Board:

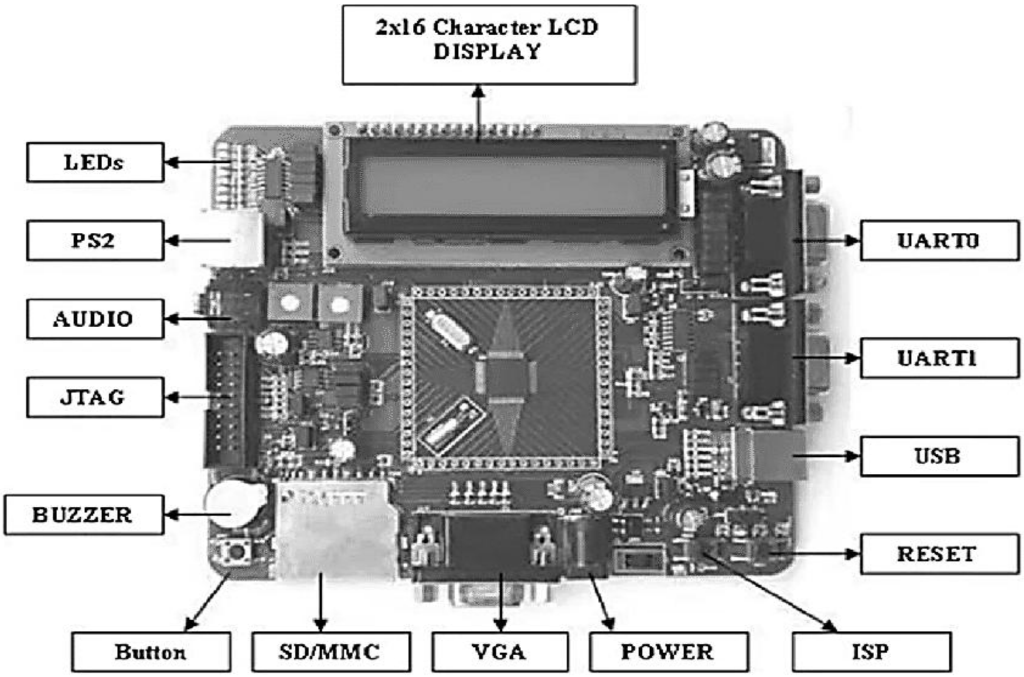


Figure 3.2: ARM Processor Development Board

(Courtesy www.microchip.com)

VIII Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1.	ARM Microcontroller kit	NXP's ARM7TDMI LPC2148 MCU operating at 60 MHz, UART with RS232, USB automatic programming, 5v DC Supply, JTAG programming allocation, 2 X 20 pin FRC connector for expansion interfaces, 4 X 10 pin FRC connector for expansion interfaces, 512K Flash – Program Memory, 32K+8K RAM – Data Memory.	1 No.

IX Precautions to be Followed

1. Do not power up development board when identifying pins.
2. Refer the Data Sheets for the given microcontroller.

X Resources Used:

Sr. No.	Instrument /Components	Specification	Quantity
1			
2			

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

....

....

....

....

XII Observations

For the given IC number write the nomenclature of the IC:

Sr. No.	IC number	Nomenclature
1.	ARM926EJ	
2.	ARM968ES	

XIII Conclusions and Recommendation

....

....

....

....

XIV 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 significance of ARM microcontroller over 8051 microcontroller.
2. List out the different available ARM micro-controller ICs and their cost.
3. Specify the different types of ARM processors for the following:
   a) High performance
   b) High efficiency
   c) Ultra high efficiency.
4. What is meant by Flash memory?
5. Determine the cost of ARM Development Board.

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

XV References / Suggestions for further reading

1. https://community.arm.com/processors/b/blog/posts/arm-fundamentals-introduction-to-understanding-arm-processors
2. https://en.wikipedia.org/wiki/ARM_architecture
3. https://www.arm.com/products/silicon-ip-cpu
4. https://www.engineersgarage.com/articles/arm-advanced-risc-machines-processors
5. http://lars.nocrew.org/computers/processors/ARM/ARM7/manual.pdf

XVI Assessment Scheme

Performance Indicators		Weightage
Process related: 15 Marks		**60%**
1	Identifying Development Board	20%
2	Identifying components on Development Board	30%
3	Follow ethical practices.	10%
Product related: 10 Marks		**40%**
4	Identifying correct ARM nomenclature	20%
5	Answer to sample questions	15%
6	Timely Submission of report,	05%
Total (25 Marks)		**100 %**

Names of student Team Member

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

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

Practical No.4: Use Integrated development environment tool for developing embedded 'C' programs (Using MicroProc /Keil)

I Practical Significance

Keil development tools for the 8051 microcontroller family support every level of developer from the professional applications engineer to the student just learning about embedded software development. The industry-standard Keil C Compilers, Macro Assemblers, Debuggers, Real-time Kernels, and Single-board Computers support all 8051-compatible derivatives. This practical will help the students to develop coding skills using Keil IDE for a particular defined problem.

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 Embedded Systems.
- Use integrate development environment (IDE) to develop embedded C program.

IV Relevant Course Outcome(s)

Use 'Embedded C' programming language to maintain embedded systems.

V Practical Outcome

Use integrated development environment tool for developing embedded 'C' programs. (Using MicroProC/Keil)

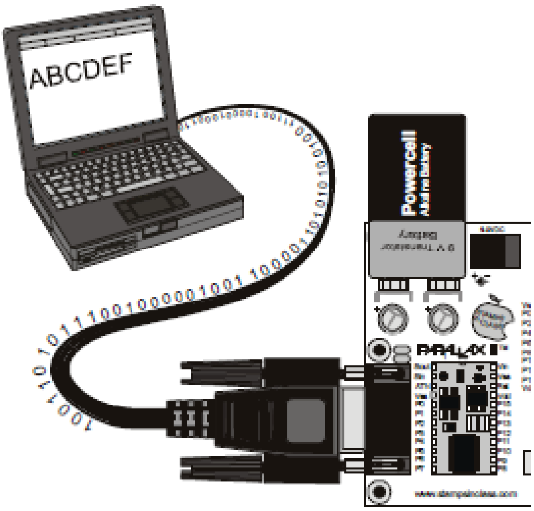
VI Relevant Affective domain related Outcome(s)

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

VII Minimum Theoretical Background

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 has the 89C51 microcontroller along with some necessary component like MAX 232, resistor network etc. It is a device 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.



Figurer 4.1: 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

Follow steps in sequence to execute programs properly

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).
2. To create new project, Click on Project and select new project. (refer Figure 4.2)
3. Select appropriate location for new project and type project name, click on save button.

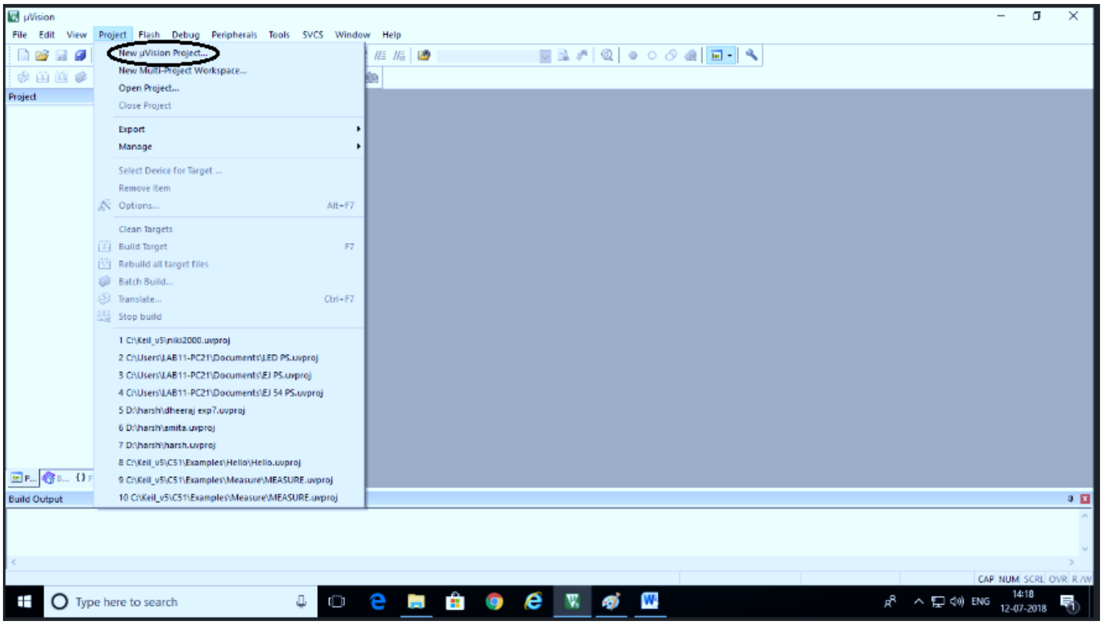
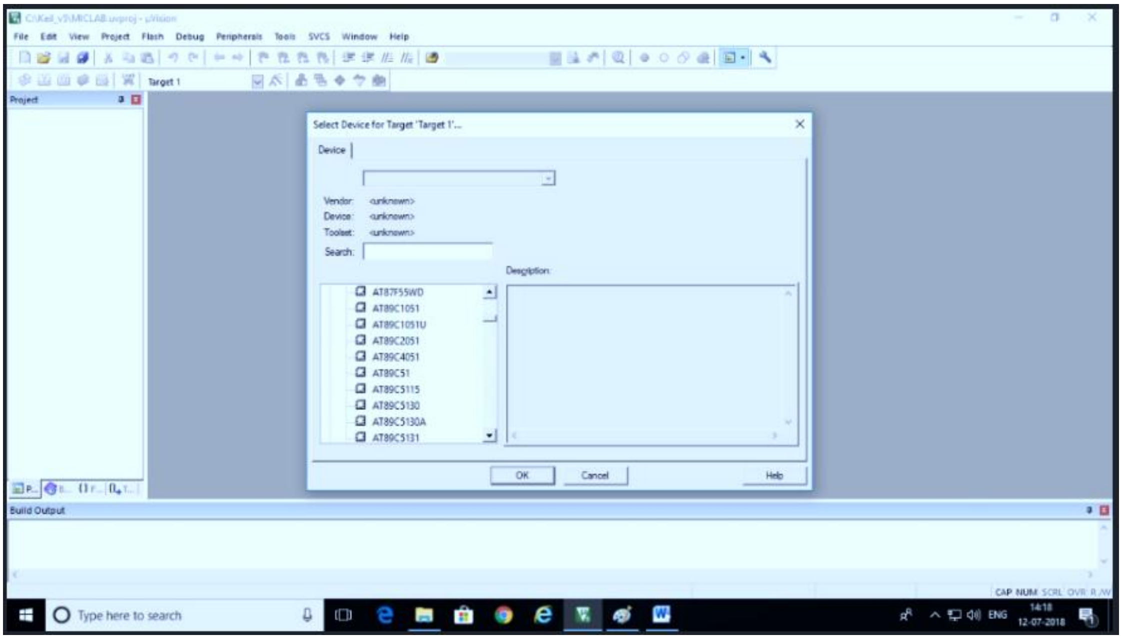

µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project
New µVision Project...
New Multi-Project Workspace...
Open Project...
Close Project
Export
Manage
Select Device for Target ...
Remove Item
Options... Alt+F7
Clean Targets
Build Target F7
Rebuild all target files
Batch 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
Build Output
CAP NUM SCRL OVR R/W
Type here to search
ENG
14:18
12-07-2018

Figure 4.2: Creating new project Window

4. "Select device for Target Target-1" window will open. It displays a list of manufacturers of microcontrollers. (refer Figure 4.3)
5. Double click on ATMEL or INTEL, list of supported microcontrollers gets displayed. Select the microcontroller as per the target board then click ok.



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

Figure 4.3: Selecting Target Device Window

6. 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 .C. (refer Figure 4.4)

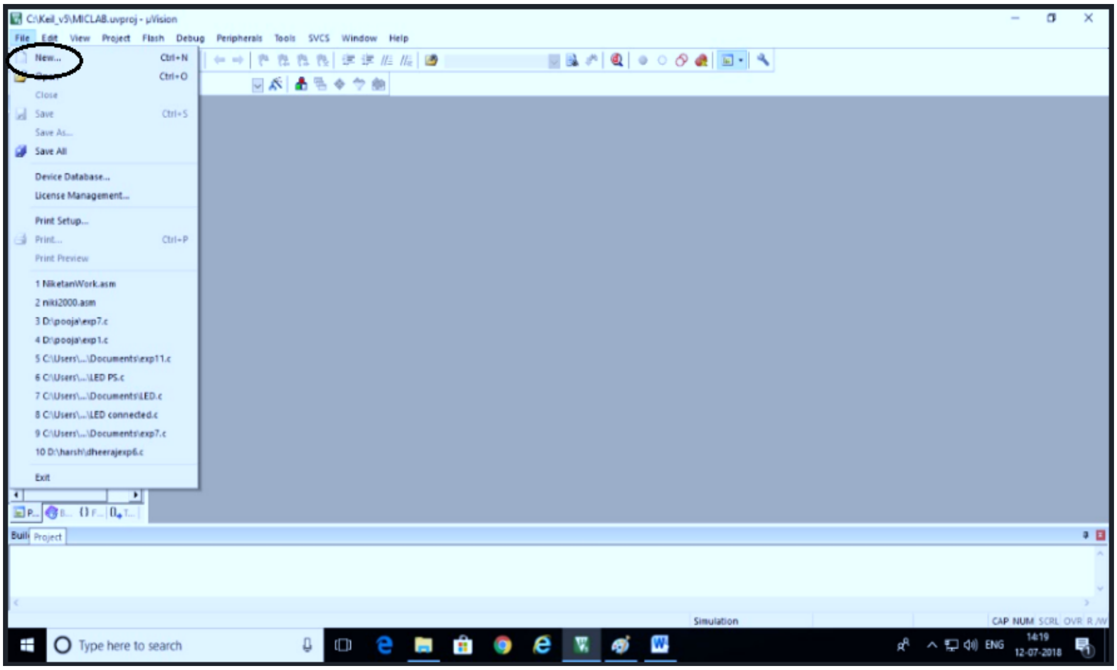


Figure 4.4: Creating new file Window

7. On left hand project work space window will display Target1 and Source group1.
8. Right click on source group; Add files to source group 1. (refer Figure 4.5)

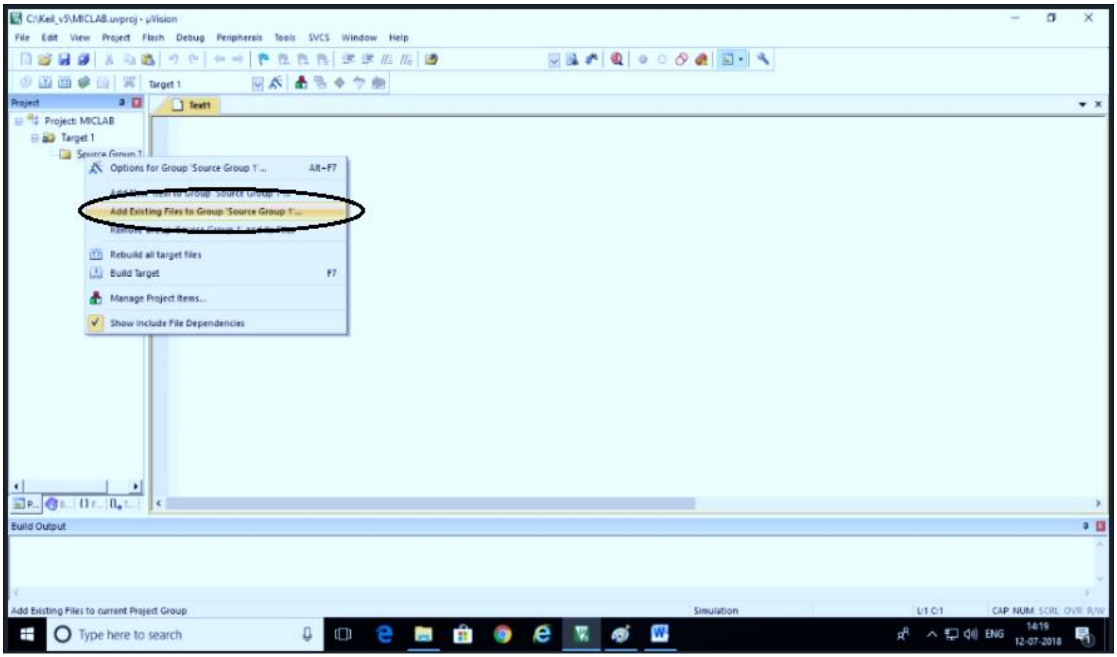
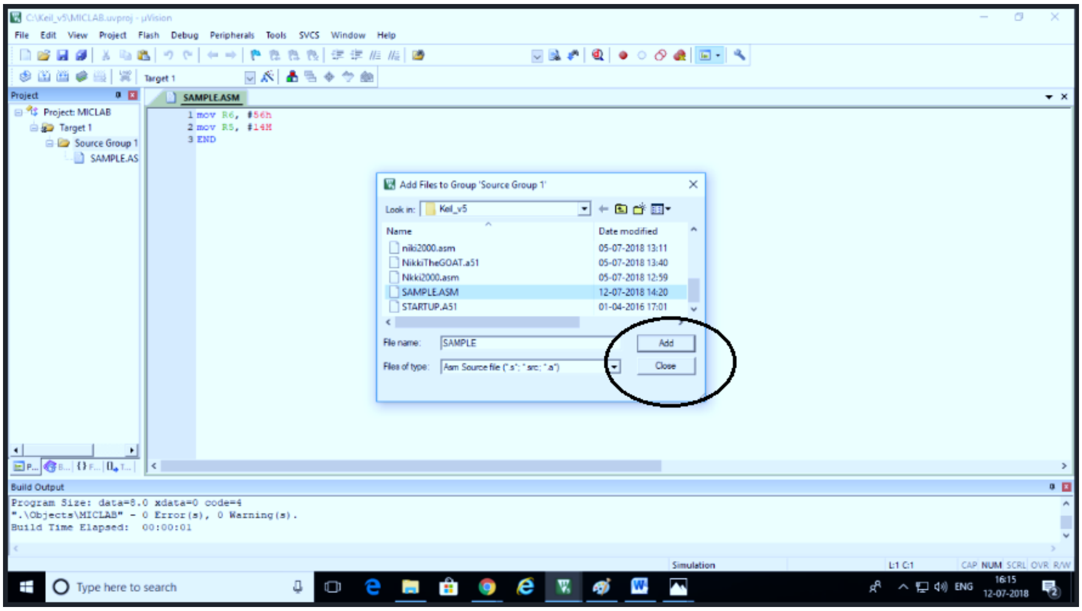

C:\Keil_v5\MICLAB.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Project: MICLAB
Target 1
Source Group 1
Text1
Options for Group 'Source Group 1'... Alt+F7
Add New Item to Group 'Source Group 1'...
Add Existing Files to Group 'Source Group 1'...
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

Figure 4.5: Adding files to Source Group Window

9. Select file type as C source file. Now all .C file Name will be displayed. Select appropriate file, click ADD and close. (refer Figure 4.6)



```
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

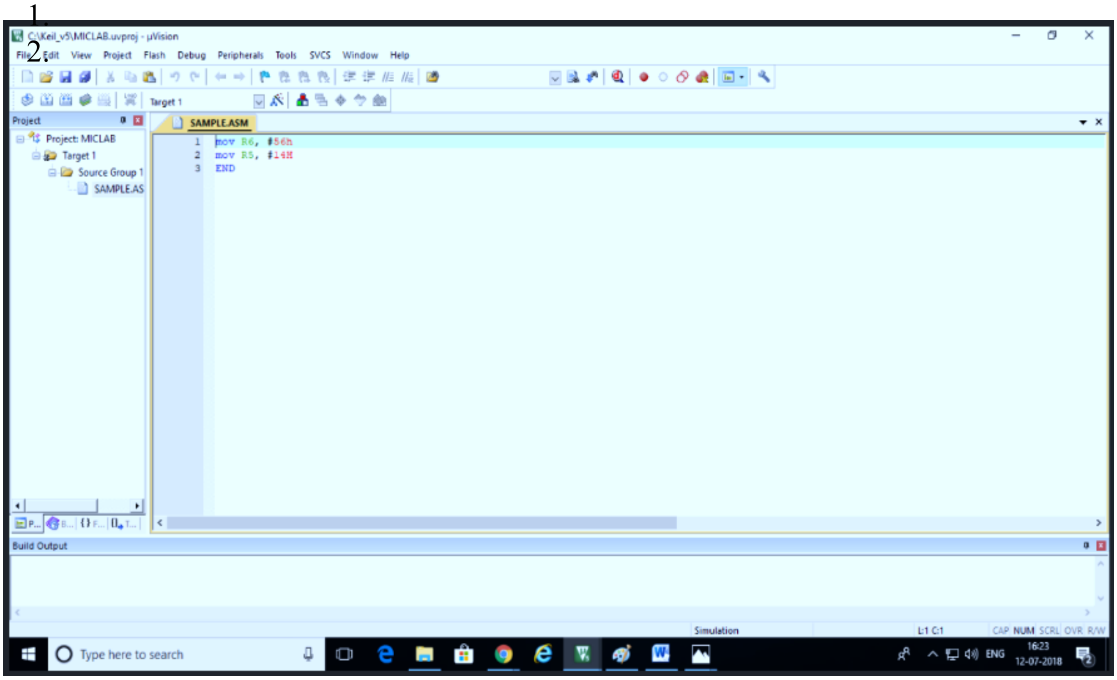
Files of type: Asm Source file (*.s*; *.src; *.a*)

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

Figure 4.6: Selecting .C file Window

10. Project work space window will display ‘Target 1’ and ‘Source group 1’ with added file name. (refer Figure 4.7)
11. Type embedded C program. Save the file periodically.

1.

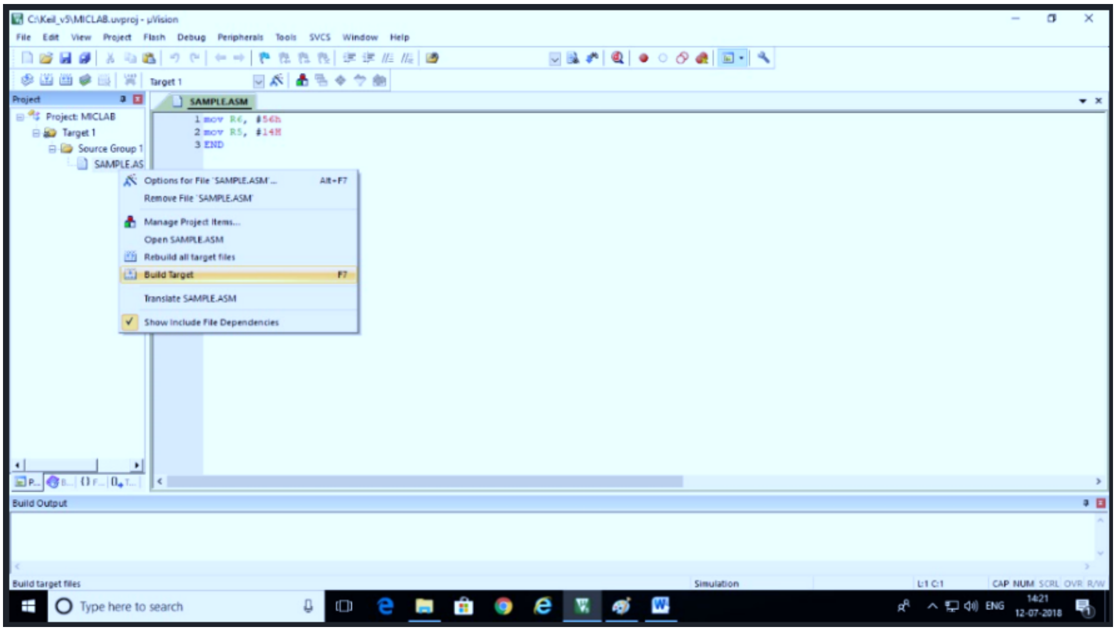


2.

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

Figure 4.7: Project Work space Window

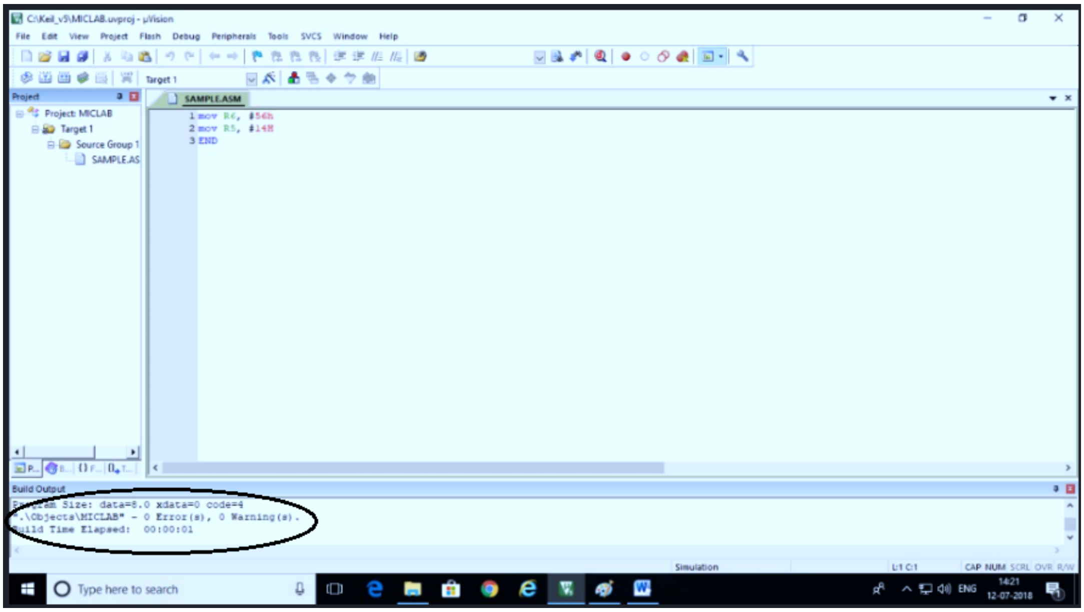
12. Right click on source group, click on Build target or press F7. (refer Figure 4.8)



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

Figure 4.8: Building Target Window

13. 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. (refer Figure 4.9)



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

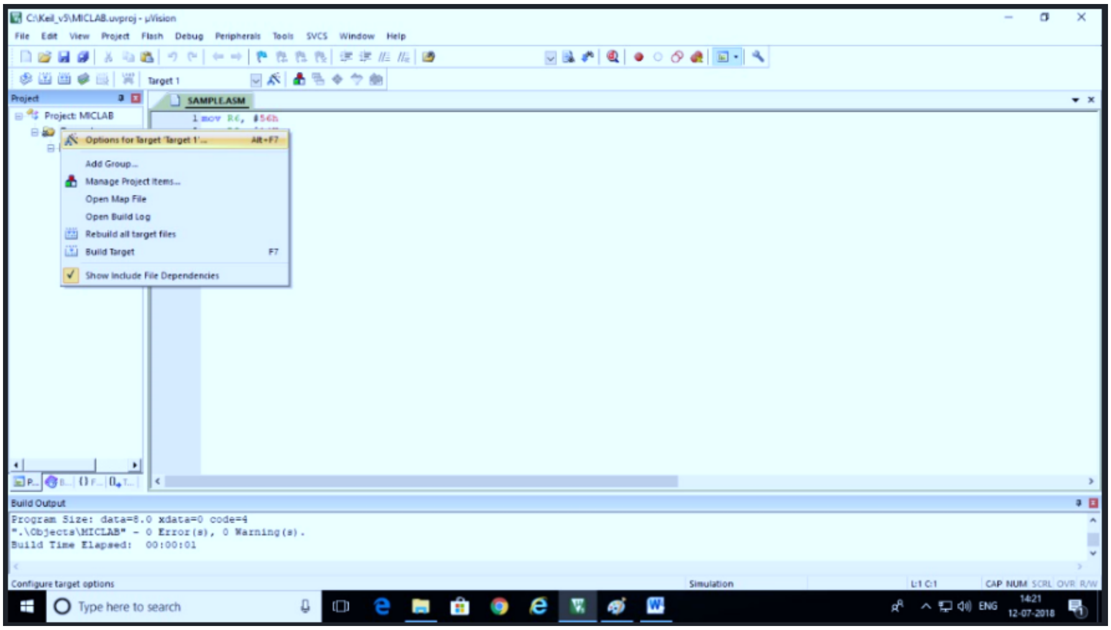
```
Program Size: data=8.0 xdata=0 code=4
".\Objects\MICLAB" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:01
```

Figure 4.9: Debugging Window

14. To create a hex file, follow this procedure (refer Figure 4.10 & 4.11)
    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.

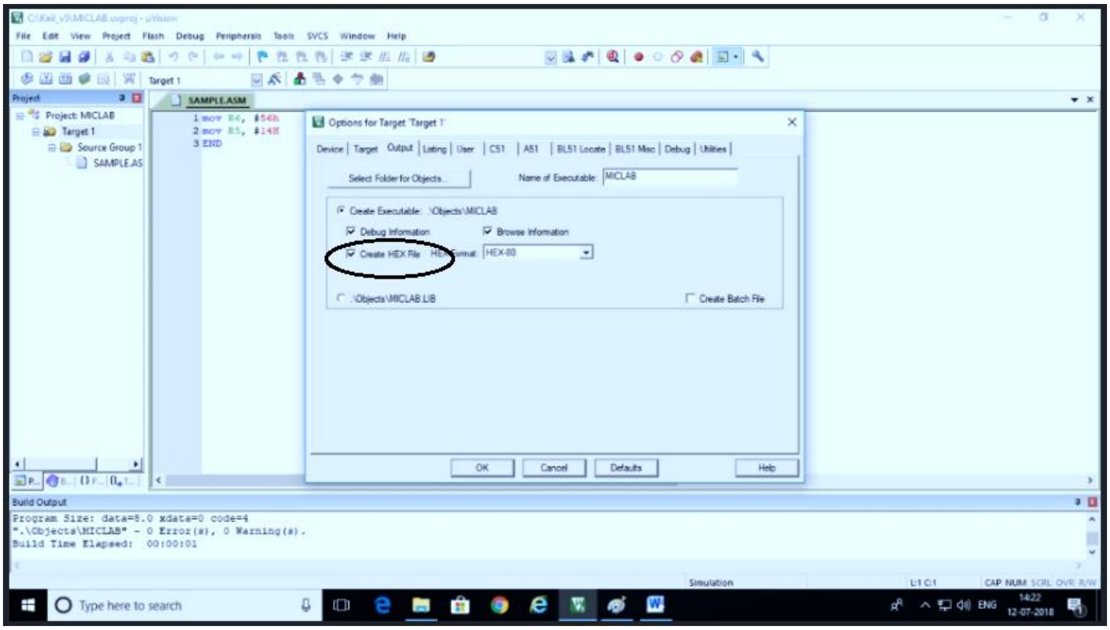
15. This step is optional for the experiments which need only simulation method to observe the results.



```
1 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
```

Figure 4.10: Options for Target 1 Window



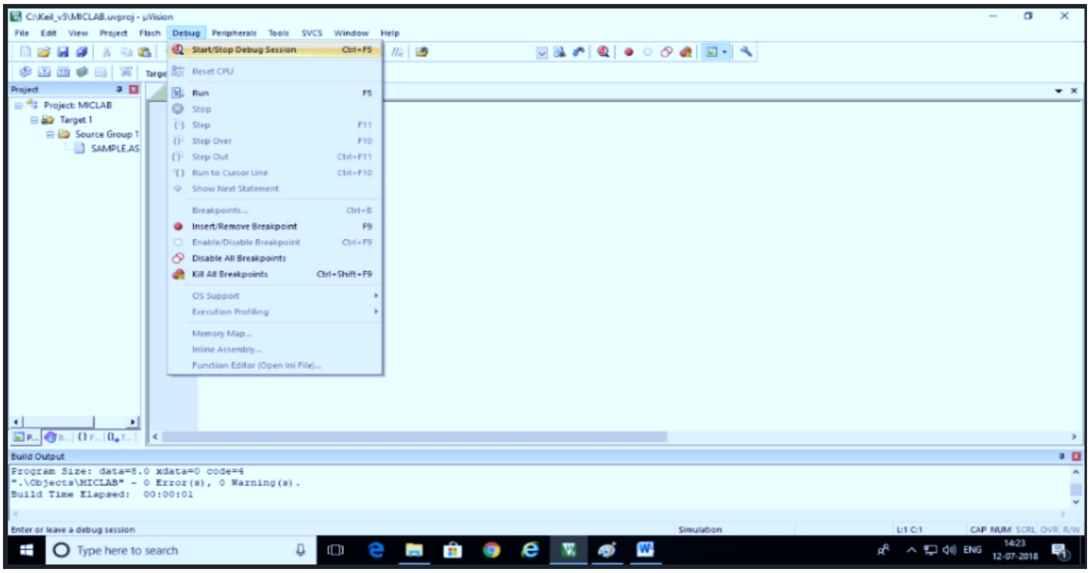
```
1 mov R6, #56h
2 mov R5, #14H
3 END
```

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
```

Figure 4.11: Create Hex File Window

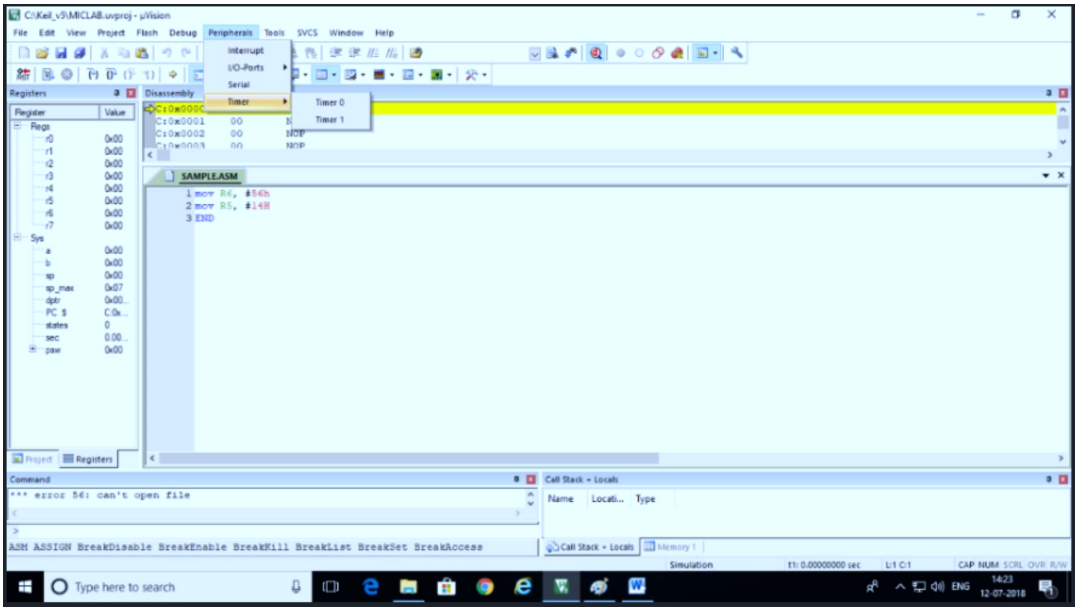
16. To start the simulation. Click on Debug pull down. Then select start/Stop debug session. (refer Figure 4.12)


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
Type here to search

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

Figure 4.12: Start/Stop Debug Window

17. 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.
18. Observe the logic levels of port pins, timers, interrupt etc, by clicking on Peripherals and select appropriate. (refer Figure 4.13)
19. 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:0x0001    00    N
C:0x0002    00    NOP
C:0x0003    00    NOP
```

```
1 mov R6, #56h
2 mov R5, #14H
3 END
```

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

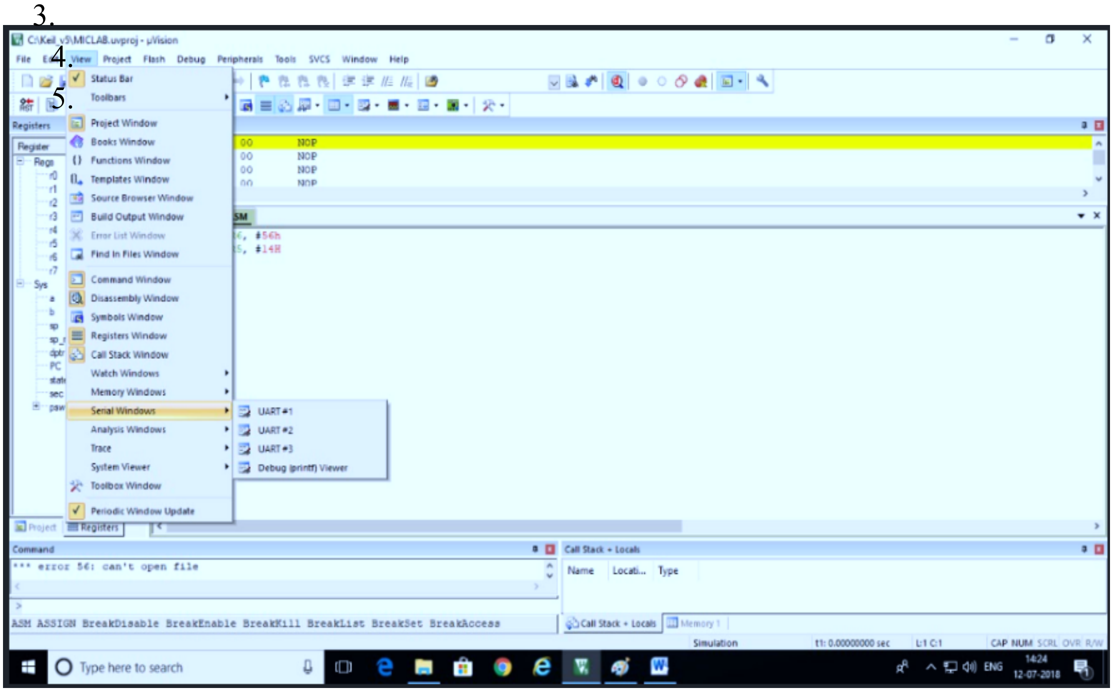
Figure 4.13: Output Window

20. Observe the serial communication by clicking VIEW pull down and select serial window-1 Option. (refer Figure 4.14)

3.

4.

5.



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

```
*** error 56: can't open file
```

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

Figure 4.14: UART Window

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 Result (Identification of various block of 8051 development board)

....

....

....

....

XV Interpretation of Results

....

....

....

....

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

....

....

....

....

XVII 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 any four features of Keil IDE.
2. Give the functions of: Editor, Complier, Linker, Debugger.

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XVIII References / Suggestions for further reading

1. http://www.keil.com

XIX Assessment Scheme

Performance Indicators		Weightage
Process related: 15 Marks		**60%**
1	Identifying given IDE	20%
2	Identifying features of IDE	30%
3	Follow ethical practices.	10%
Product related: 10 Marks		**40%**
4	Correct procedure	20%
5	Answer to sample questions	15%
6	Timely Submission of report.	05%
Total (25 Marks)		**100 %**

Names of student Team Member

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

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

Practical No. 5: Execute the 'C' program to perform following arithmetic operations on 8-bit data: addition, subtraction, multiplication and division.

I Practical Significance

This practical will help the students to develop skills to write embedded C 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

- Use 'Embedded C' programming language to maintain embedded system.

V Practical Outcome

- Execute the 'C' program to perform following arithmetic operations on 8-bit data: addition, subtraction, multiplication and division.

VI Relevant Affective domain related Outcome(s)

- Follow ethical practices.

VII Minimum Theoretical Background:

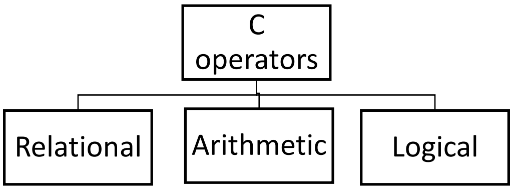


Figure 5.1: C language operators

These are following C arithmetic operators

Arithmetic operations	C language operator	Example
Addition	+	z=x+y;
Subtraction	-	p=q-r;
Multiplication	*	c=a*b;
Division	/	(Quotient) u=s/t
Modulo Division	%	(Remainder) k = i % j

Note: In above example variable c is unsigned integer type rest variables are unsigned char type.

VIII Resources Required

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

IX Precautions to be Followed

1. Check rules / syntax of C 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. .C extension.
6. Right click on source group and Add file to source group
7. Compile the Program
8. Right click on source group and build the target.
9. Check for any errors in the output window and remove if any.
   Run, Debug the Program
10. Click on Debug and start simulation and start/stop debug session.
11. Run the program step by step.
12. Observe the output on the project window.
13. Note the values of the result of various operations in the observation table.

SAMPLE PROGRAM 1: To perform following: arithmetic operations: addition and multiplication of two 8 bit numbers.

Step 1-Algorithm

1. Declare 3 unsigned char variables and 1 unsigned int variable
2. Make P1 and P2 as Output
3. Assign data to two variables
4. Do the addition

5. Display the result on port 1
6. Multiply two numbers
7. Display result on port 2
8. Stop

Step 2-Flow Chart

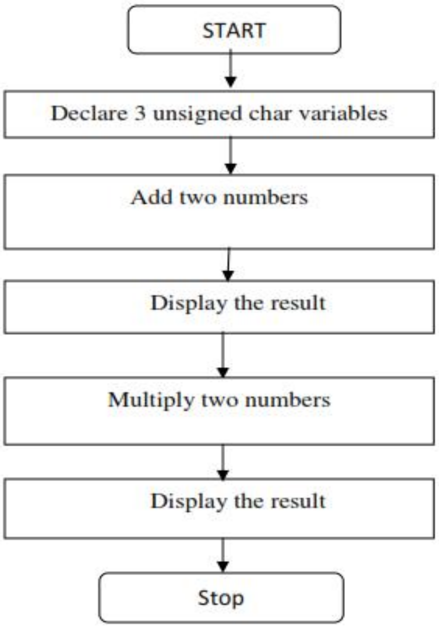


Figure 5.2: Flowchart for arithmetic operations

Step 3-Sample Program

C language program	**Comments**
#include<reg51.h>	/* Special function register declarations */
void main (void)	/* for the intended 8051 derivative */
{	
unsigned char p,q,r;	//Declare three 8 bit characters
unsigned int s;	//Declare one 16 bit integer
P1=0x00;	// P1 as outpour port
P2=0x00;	// P2 as outpour port
p=0x04;	//Assign values to character
q=0x02;	
r= p+q;	//Perform the addition
P1=r;	//Display addition result on P1
s=p*q;	//Perform the multiplication
P2=s;	// Display multiplication result on P2
while(1);	
}	

Problem statement for student: Develop and execute embedded ‘C’ language program to perform Subtraction and Division on 8 bit data.

Step 1:-Algorithm

Step 2:-Flowchart

Step 3:- Program

C language program	Comment

XI Resources Used

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

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 (use blank sheet provided if space not sufficient)
Observation for Problem statement given to student

Sr. No.	Input data byte 1	Operation	Input data byte 2	Result
1		Subtraction		
2		Division		

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. Find the value of z after execution of the following program

```
# include <reg51.h>
 void main (void)
{
unsigned char x ,y, z;
x=250;
y=168;
P1= 0x00;
z=x-y;
P1=z;
 while(1);
}
```

2. Write C program to find two’s complement of a no. 0xAA
3. Give the significance of Overflow flag during multiplication and division.

[Space for Answers]

....

....

....

....

....

XIX References / Suggestions for further reading

1. https://fresh2refresh.com/c-programming/c-operators-expressions/c-arithmetic-operators/

XX Assessment Scheme

Performance Indicators		Weightage
Process related (15 Marks)		**60%**
1	Coding ability	30%
2	Debugging ability	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%**

Names of student Team Member

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

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

Practical No. 6: Develop and test the 'C' program to perform following arithmetic operations on 16 bit data: addition, subtraction

I Practical Significance

This practical will help the students to develop skills to write embedded C program for arithmetic operations on 16 bit data.

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

- Use 'Embedded C' programming language to maintain embedded system.

V Practical Outcome

Develop and test the 'C' program to perform following arithmetic operations on 16 bit data: addition, subtraction

VI Relevant Affective domain related Outcome(s)

- Follow ethical practices.

VII Minimum Theoretical Background

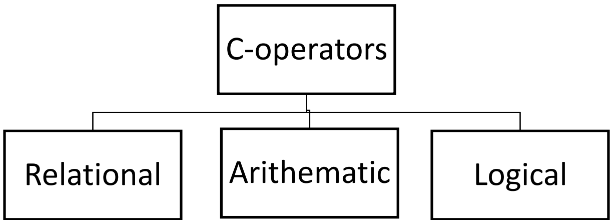


Figure 6.1: C language operators

These are following C arithmetic operators

Arithmetic operations	C language operator	Example
Addition	+	z=x+y;
Subtraction	-	p=q-r;
Multiplication	*	c=a*b;
Division	/	(Quotient) u=s/t
Modulo Division	%	(Remainder) k = i % j

Note: In above example variable c is unsigned integer type rest variables are unsigned char type.

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 C 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 .c 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 subtraction of two 16 bit number.

Step 1-Algorithm

1. Declare 3 unsigned integers.
2. Assign data to two integers.
3. Do the subtraction of two 16 bit numbers.
4. Check the result.
5. Stop

Step 2-Flow Chart

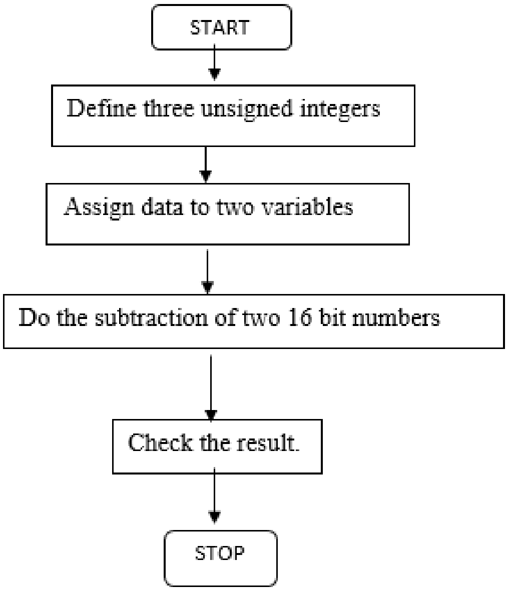
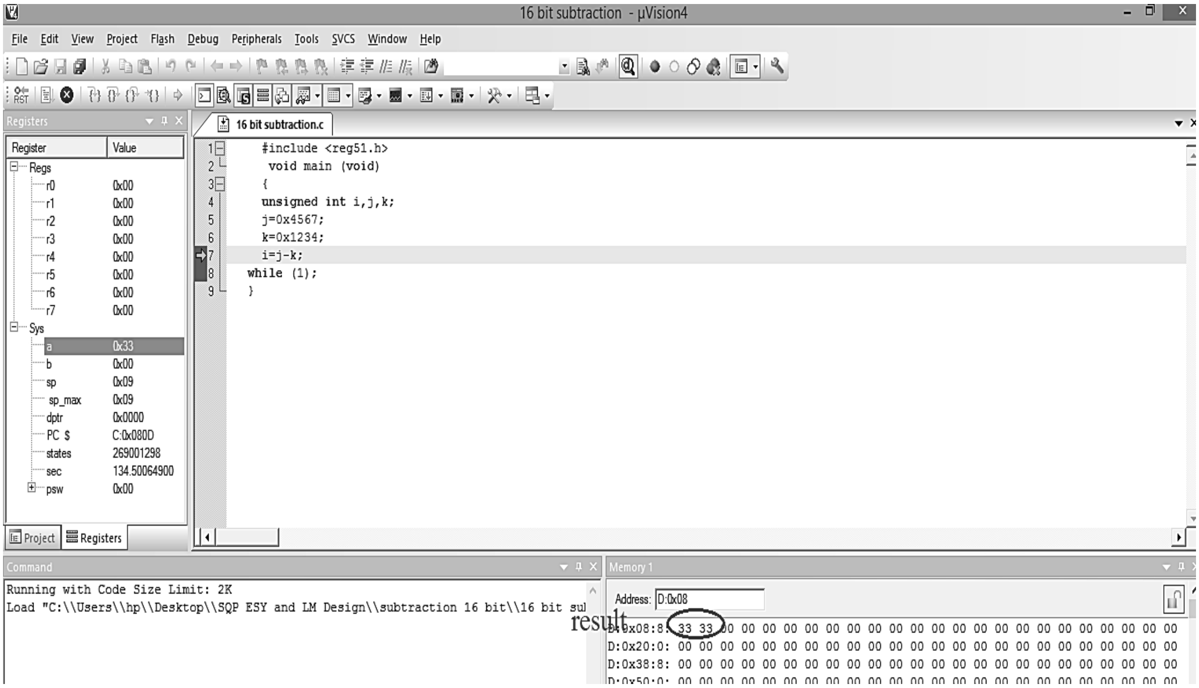

START
Define three unsigned integers
Assign data to two variables
Do the subtraction of two 16 bit numbers
Check the result.
STOP

Figure 6.2: Flowchart for arithmetic operations

Step 3-Sample Program

C language program	Comments
#include<reg51.h>	/* Special function register declarations */
void main (void)	/* for the intended 8051 derivative */
{	
unsigned int i,j,k;	// declare 3 unsigned integers
j=0x4567;	// define values to the integers
k=0x1234;	
i=j-k;	//perform the subtraction.
while (1);	//stop.
}	


16 bit subtraction - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
16 bit subtraction.c
Project
Registers
Command
Memory 1
Address: D:0x08
result

```c
  #include <reg51.h>
   void main (void)
  {
  unsigned int i,j,k;
  j=0x4567;
  k=0x1234;
  i=j-k;
while (1);
 }
```

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x33
b	0x00
sp	0x09
sp_max	0x09
dptr	0x0000
PC $	C:0x080D
states	269001298
sec	134.50064900
psw	0x00

```
Running with Code Size Limit: 2K
Load "C:\\Users\\hp\\Desktop\\SQP ESY and LM Design\\subtraction 16 bit\\16 bit sul
```

```
D:0x08:8: 33 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x20:0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x38:8: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x50:0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Figure 6.3: Output Window

Problem statement for student:Develop and executean embedded ‘C’ language program to perform 16 bit addition.

Step 1-Algorithm

Step 2-Flowchart

Step 3- Program

C language program	Comment

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 (use blank sheet provided if space not sufficient)
Observation for Problem statement for student

Sr. No.	Memory location	Result
1		
2		
3		

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. Illustrate with example how subtraction can be performed without borrow.
2. Draw the format of PSW and give the status of flags after execution of sample program for student.
3. Write a embedded ‘C’ program to perform subtraction using 2’s complement.

[Space for Answers]

....

....

....

....

....

XIX References / Suggestions for further reading

1. https://fresh2refresh.com/c-programming/c-operators-expressions/c-arithmetic-operators/
2. https://www.pantechsolutions.net/...tutorials/subtraction-of-numbers-using-8051

XX Assessment Scheme

Performance indicators		Weightage
Process related (15 Marks)		**60%**
1	Coding ability	30%
2	Debugging ability	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%**

Names of student Team Member

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

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

Practical No. 7: Develop and test the 'C' program to perform data transfer from source to destination. (Use Internal Memory Locations)

I Practical Significance

This practical will help the students to develop skills to write embedded C program to perform data transfer from source to destination using internal memory locations.

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

- Use 'Embedded C' programming language to maintain embedded system.

V Practical Outcome

Develop and test the 'C' program to perform data transfer from source to destination. (Use Internal Memory Locations)

VI Relevant Affective domain related Outcome(s)

- Follow ethical practices.

VII Minimum Theoretical Background

The block transfer is the process of transferring contents of group of memory locations from source address to destination address. After block transfer the contents of destination address are replaced by new contents.

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 .C 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 data transfer from source locations to destination location.

Step 1-Algorithm

1. Initialize two arrays one for source locations and other destination locations.
2. Define unsigned integer as a counter.
3. Initialize the counter equal to number to be transferred i.e. count value.
4. Transfer the data byte by byte from source to destination.
5. Repeat the process for all ten bytes.
6. Stop the process.

Step 2-Flow Chart:

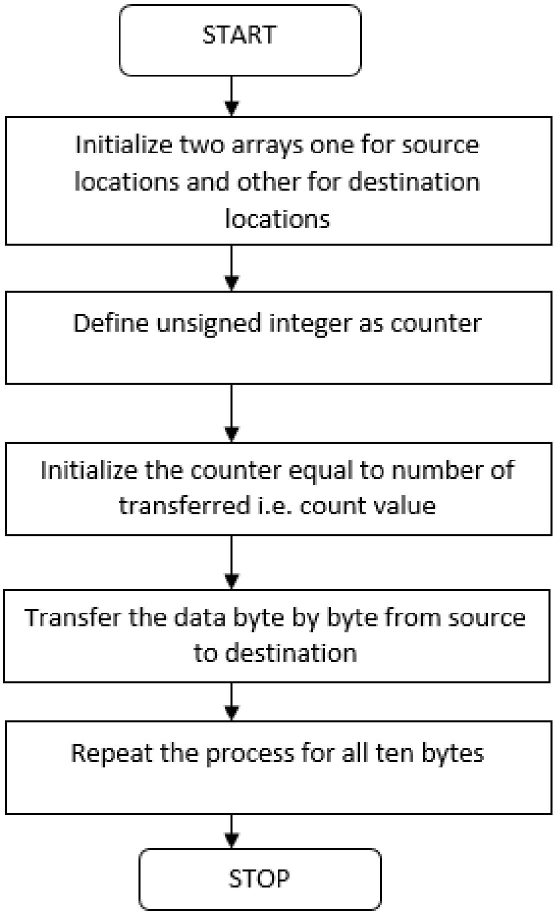
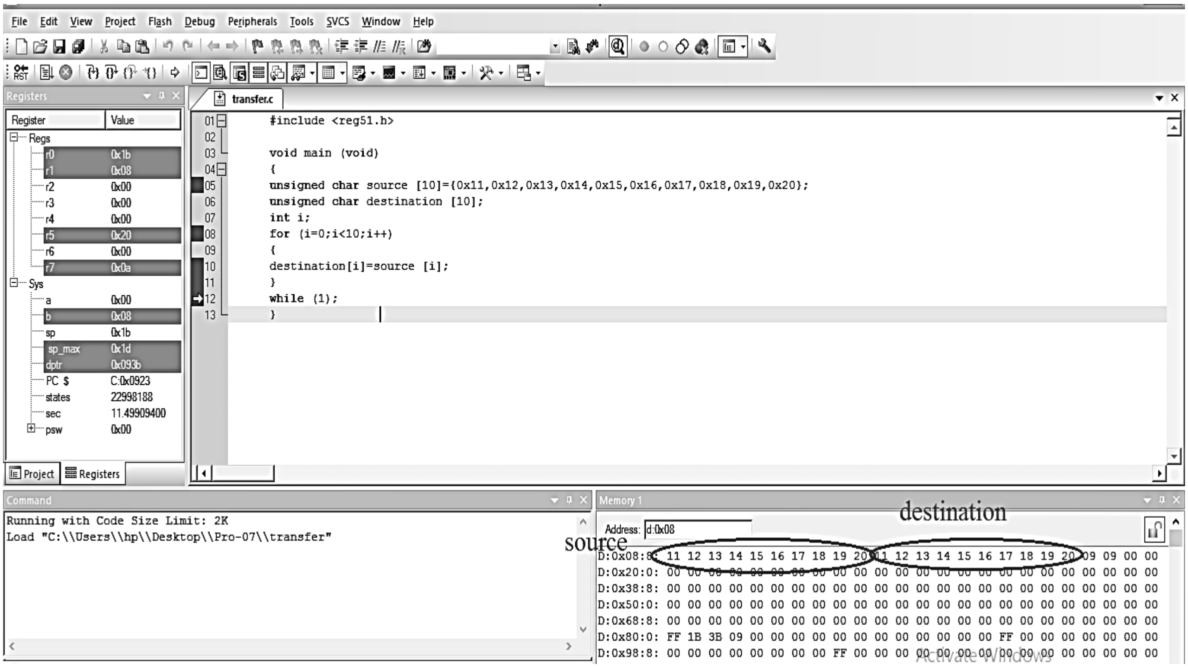

START
Initialize two arrays one for source locations and other for destination locations
Define unsigned integer as counter
Initialize the counter equal to number of transferred i.e. count value
Transfer the data byte by byte from source to destination
Repeat the process for all ten bytes
STOP

Figure7.1: Flowchart for arithmetic operations

Step 3-Sample Program:

C language program	**Comments**

#include <reg51.h>
void main (void)
{
unsigned char source [10]=
{0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18,0x19,0x20};
unsigned char destination [10];
unsigned char i;
for (i=0;i<10; i++)
{
destination [i]= source [i];
}
while (1);
}
	/*Special function register declarations*/
	/* for the intended 8051 derivative */
	
	//define source array.
	
	//define destination array
	//unsigned character i for counter
	//For loop
	
	//Transfer byte from source to destination.
	//do only once
	//stop



Register	Value
Regs	
r0	0x1b
r1	0x08
r2	0x00
r3	0x00
r4	0x00
r5	0x20
r6	0x00
r7	0x0a
Sys	
a	0x00
b	0x08
sp	0x1b
sp_max	0x1d
dptr	0x093b
PC $	C:0x0923
states	22998188
sec	11.49909400
psw	0x00

```
#include <reg51.h>

void main (void)
{
unsigned char source [10]={0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18,0x19,0x20};
unsigned char destination [10];
int i;
for (i=0;i<10;i++)
{
destination[i]=source [i];
}
while (1);
}
```

```
Running with Code Size Limit: 2K
Load "C:\\Users\\hp\\Desktop\\Pro-07\\transfer"
```

Address: d:0x08

```
D:0x08:8: 11 12 13 14 15 16 17 18 19 20 11 12 13 14 15 16 17 18 19 20 09 09 00 00
D:0x20:0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x38:8: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x50:0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x68:8: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x80:0: FF 1B 3B 09 00 00 00 00 00 00 00 00 00 00 00 FF 00 00 00 00 00 00 00 00
D:0x98:8: 00 00 00 00 00 00 00 00 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Figure 7.2: Output Window

Problem statement for student: Write an embedded ‘C’ program to perform transfer 5 bytes from source to destination locations.

Step 1-Algorithm

Step 2-Flowchart

Step 3- Program

C language program	Comment

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 (use blank sheet provided if space not sufficient)

Observation for Problem statement for student

Source Locations	Data Values	Destination Locations	Data values

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 an embedded C language program to exchange a upper nibble with lower nibble.
2. Explain the significance of $\overline{EA}$ pin.
3. Write and execute an embedded C language program to store ASCII character “ABCDEF” in internal RAM memory locations and examine the contents of 128 Byte RAM space to allocate the ASCII values.

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XIX References / Suggestions for further reading

1. https://fresh2refresh.com/c-programming.
2. https://www.pantechsolutions.net/...tutorials

XX Assessment Scheme

Performance Indicators		Weightage
Process related (15 Marks)		**60%**
1	Coding ability	30%
2	Debugging ability	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%**

Names of student Team Member

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

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

Practical No. 8: Interface RS 232 connector to PC using MAX 232 IC.

I Practical Significance

Several devices collect data from sensors and need to send it to computer, for further processing. Data transfer/communication is generally done in two ways: parallel and serial. In the parallel mode, data transfer is fast and uses more number of lines. This mode is good for short range data transfer.

Serial communication uses only one or two data lines to transfer data and is generally used for long distance communication. In serial communication the data is sent as one bit at a time.

This practical will help the students to develop the skills to interface 8051 microcontroller (AT89C51) to a computer via serial port, RS232. Serial communication is commonly used in applications such as industrial automation systems, scientific analysis and certain consumer products

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

- Interpret the communication standards of embedded systems.

V Practical Outcome

Interface RS232 connector to PC using MAX 232 IC.

VI Relevant Affective domain related Outcome(s)

- Follow ethical practices.

VII Minimum Theoretical Background

The microcontroller AT89C51 has an inbuilt UART for carrying out serial communication. The serial communication is done in the asynchronous mode. A serial port is a physical interface to establish data transfer between computer and an external hardware or device. This transfer, through serial port, takes place bit by bit.

IBM introduced the DB-9 RS-232 version of serial I/O standard, which is most widely used in PCs and several devices. In RS232, high and low bits are represented by following voltage ranges as given in table 8.1.

Table 8.1: Voltage ranges in TTL and RS 232

RS232 line type and logic level	RS232 voltage	TTL voltage to/from MAX232
Data transmission (Rx/Tx) logic 0	+3 V to +15 V	0 V
Data transmission (Rx/Tx) logic 1	-3 V to -15 V	5 V
Control signals (RTS/CTS/DTR/DSR) logic 0	-3 V to -15 V	5 V
Control signals (RTS/CTS/DTR/DSR) logic 1	+3 V to +15 V	0 V

RS232 voltage levels are not compatible with TTL logic. Therefore, while connecting an RS232 to microcontroller system, a voltage converter is required. This converter converts the microcontroller output level to the RS232 voltage levels, and vice versa. IC MAX232, also known as line driver, is very commonly used for this purpose.

The simplest connection between a PC and microcontroller requires a minimum of three pins, RxD (receiver, pin2), TxD (transmitter, pin3) and ground (pin5) of the serial port of computer.

TxD pin of serial port connects to RxD pin of controller via MAX232. And similarly, RxD pin of serial port connects to the TxD pin of controller through MAX232.

MAX232 has two sets of line drivers for transferring and receiving data. The line drivers used for transmission are called T1 and T2, whereas the line drivers for receiver are designated as R1 and R2.

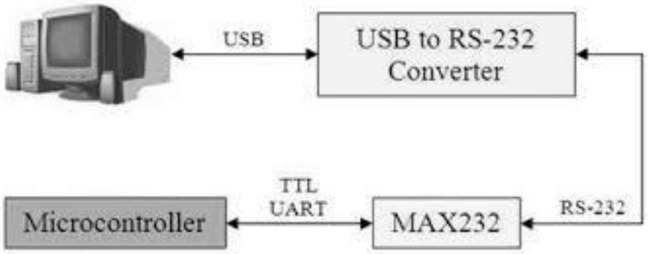


Figure 8.1: Connection of MAX232 with computer and the controller

It is basically a 16 pin IC with the transmitter pins connected to the microcontroller and the port such that the input transmitter pin get TTL input from the Microcontroller and the output transmitter pin supply output to the RS232 port. The receiver pins are connected to the RS232 port such that the input receiver pin receive RS232 standard input from the PC port and the output receiver pin supplies the TTL input to the Microcontroller. Thus the transmitter takes input from the Microcontroller and gives output to the RS232 port whereas the receiver takes input from the RS232 port and gives output to the Microcontroller.

VIII Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1	Desktop PC	Loaded with open source IDE, simulation and program downloading software	1 No.
2	8051 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.
3	RS 232C	DB 9 connector	1No.
4	MAX 232 IC	Interconnecting IC between RS 232 and Microcontroller	1No.

IX Precautions to be Followed

1. Check rules / syntax of assembly programming.
2. Check the connections between the devices.
3. Don’t power on the supply before the connections are completed.

X Procedure:

Write Program

1. Connect the 8051 Trainer to the PC’s COM port and use HyperTerminal to test the operation of the program.
2. Start Keil by double clicking on Keil icon.
3. Create a new project.
4. Select device for Target.
5. Double click on ATMEL and select AT89C51.
6. Type the program in text editor and save as filename .C extension.

Compile the Program

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

Run, Debug the Program

1. Click on Debug and start simulation and start/stop debug session.
2. Run the program step by step.
3. Observe the output.

SAMPLE PROGRAM 1: Write a embedded ‘C’ program to transfer ‘A’ serially at 4800 baud rate continuously.

Step 1-Algorithm:

1. Initialize TMOD register with respective timer and its mode.
2. Load the value in the timer register corresponding to the baud rate.
3. Load SCON with 50H value indicating mode 1, 1 stop and start bit.
4. Set the timer run bit.
5. Load the character ‘A’ in SBUF.
6. Start transmitting the data by setting the timer run bit.
7. Check TI to determine whether the transmission process if completed.
8. Repeat the process continuously.

Step 2-Flow Chart:

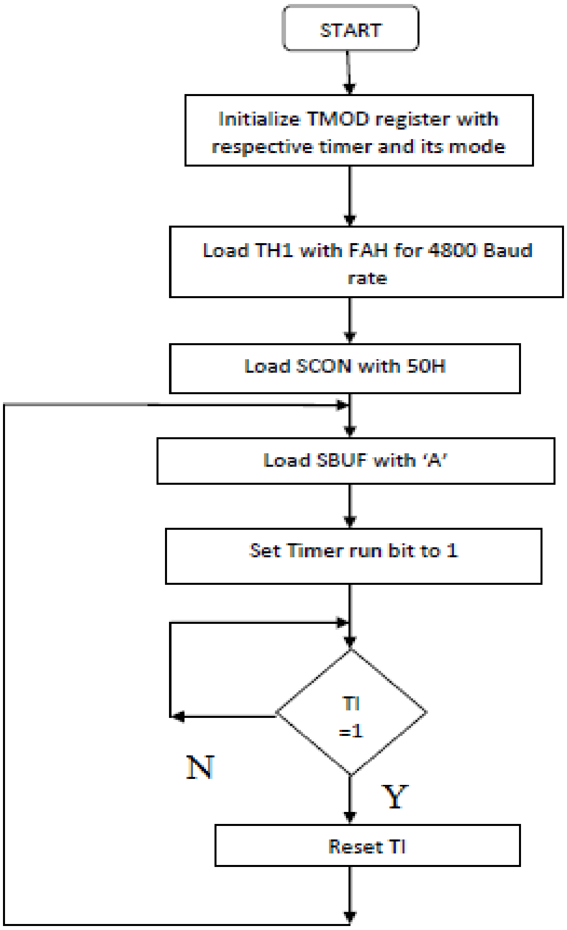

START
Initialize TMOD register with respective timer and its mode
Load TH1 with FAH for 4800 Baud rate
Load SCON with 50H
Load SBUF with 'A'
Set Timer run bit to 1
TI =1
N
Y
Reset TI

Figure8.2: Flowchart for Transferring Serial Data

Step 3-Sample Program:

C language program	Comments

#include <reg51.h>
void main (void)
{
TMOD =0X20;
TH1=0XFA;
SCON=0X50;
TR1=1;
while (1)
{
SBUF = 'A';
while (TI= = 0);
TI=0;
}
}
	/*Special function register declarations*/
	/* for the intended 8051 derivative */
	
	//Use Timer 1 , 8 bit auto reload
	//4800 baud rate
	
	//place a value in buffer

Problem statement for student: Write an embedded ‘C’ program transfer ‘YES’ serially at baud rate of 9600.

Step 1-Algorithm

Step 2-Flowchart

Step 3- Program

C language program	Comment

XI Resources Used

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

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 (use blank sheet provided if space not sufficient)

Observation for Problem statement given to student

....

....

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. Explain term Baud rate with example.
2. Draw the format of SCON.
3. Calculate the value to be loaded in timer register for transferring a character at baud rate of 9800 if the crystal frequency is 12MHz.
4. Give the specifications of MAX 232C.
5. Explain the role of pumping capacitors in MAX 232.

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XIX References / Suggestions for further reading

1. https://fresh2refresh.com/c-programming.
2. https://www.pantechsolutions.net/...tutorials

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%**

Names of student Team Member

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

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

Practical No. 9: Develop and test the ‘C’ program to turn on LED (S) with key (S) press.

I Practical Significance

Input and output devices are the important components of embedded system, it is not possible to have any embedded system application without the input and output device. Switch and LED are the basic examples of input and output device. This practical will help the students to develop skills to code and interface key and LED to microcontrollers in embedded systems.

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 Embedded Systems’.**

- Develop ‘C’ code using IDE tools.
- Use Input/output port pins of microcontroller.
- Interface LED and Switch to microcontroller.

IV Relevant Course Outcome(s)

- Develop basic applications using embedded systems.

V Practical Outcome

- Develop and test the ‘C’ program to turn on LED (S) with key (S) press.

VI Relevant Affective domain related Outcome(s)

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

VII Minimum Theoretical Background

Electrical Switch: The switch is a basic input device, use to control the operation of any output device using the microcontroller or control unit. It basically breaks the electrical
circuit and interrupts the flow of current.

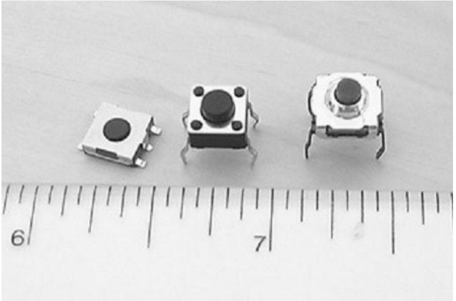


Figure 9.1: Switches

Connection of electrical Switch:

Positive Logic: In this connection, a pull-down resistor is connected to ground. When switch is pressed then logic asserts high and when released the switch logic assert low.

Negative Logic: In this connection, a pull-up resistor is connected to Vcc. When switch is pressed then logic asserts low and when released the switch logic assert high.

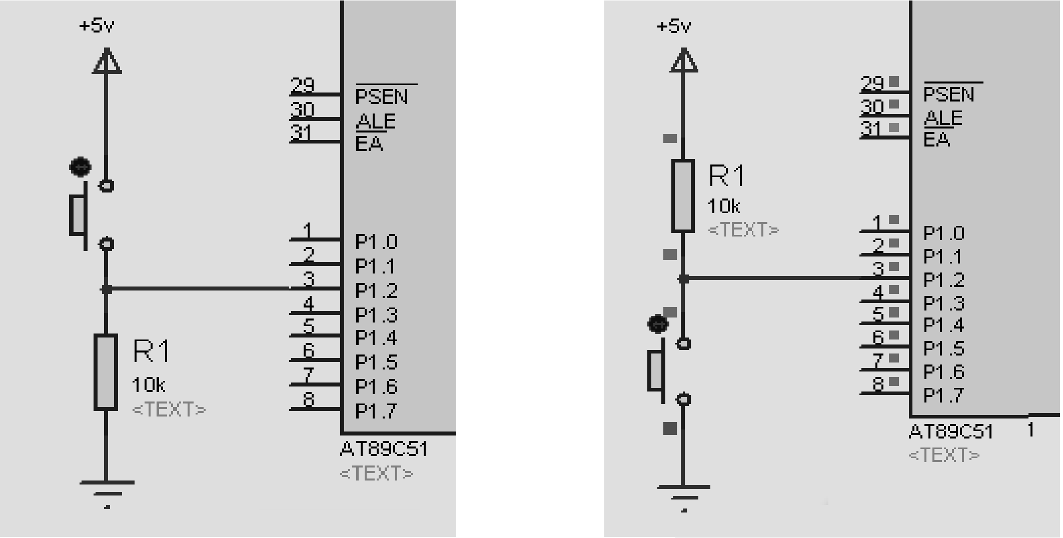


Figure 9.2: Switch logic

Note: **We may face the problem with the mechanical switch when we pressed and release the switch then it oscillates. It’s called bouncing of the switch. We can resolve the bouncing problem with the help of hardware or software. In software, if we give the delay of few milliseconds between the times, when we read the status of the switch then we resolved the switch bouncing problem.**

Light emitting diodes: LED has two leads cathode and anode. The length of cathode leads is lesser than the length of anode but sometimes they come in equal size. When the length of both leads cathode and anode are equal in the size that time identification of anode and cathode can be done by observing their filament, the cathode has broader filament than the anode as shown in the Figure 9.3:

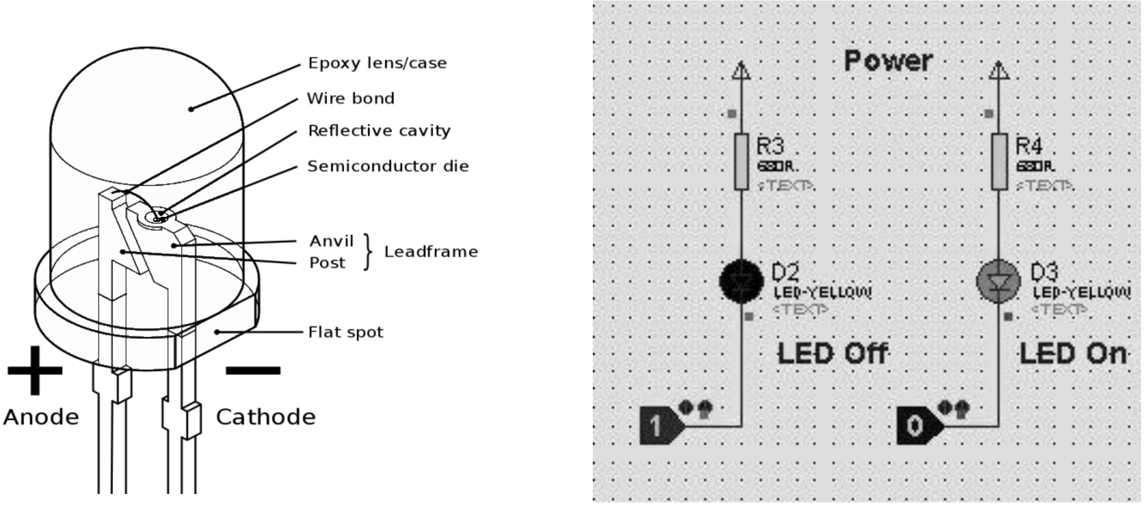


Figure 9.3 LED constructional diagram and its connections

VIII Practical Circuit diagram:

a) Sample Circuit diagram

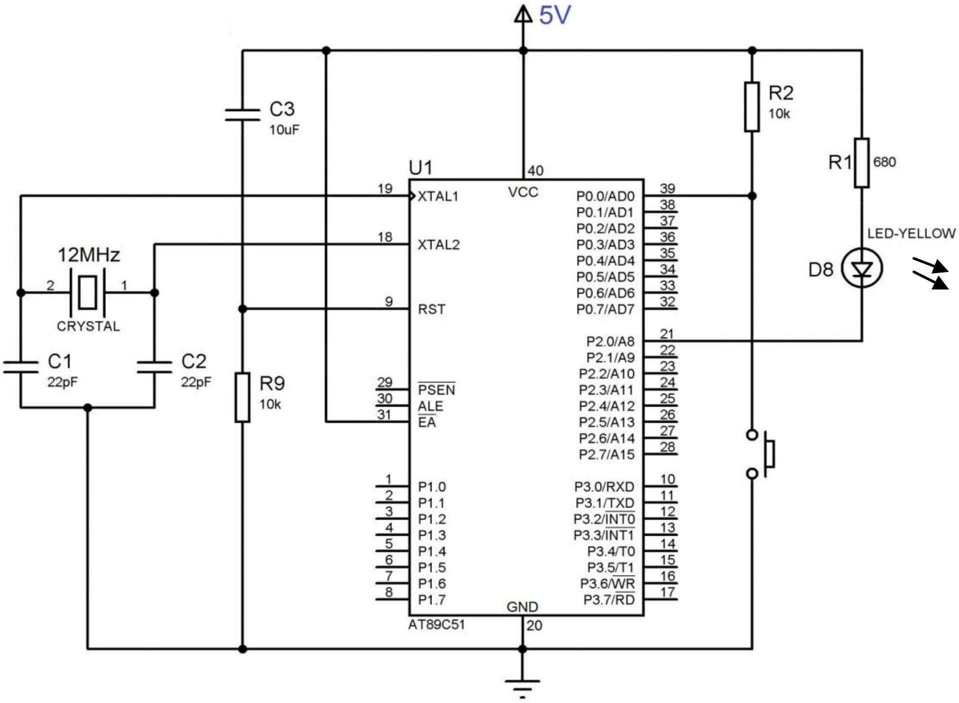


Figure 9.4: 89C51 connection to LED and switch

b) **Practical setup**

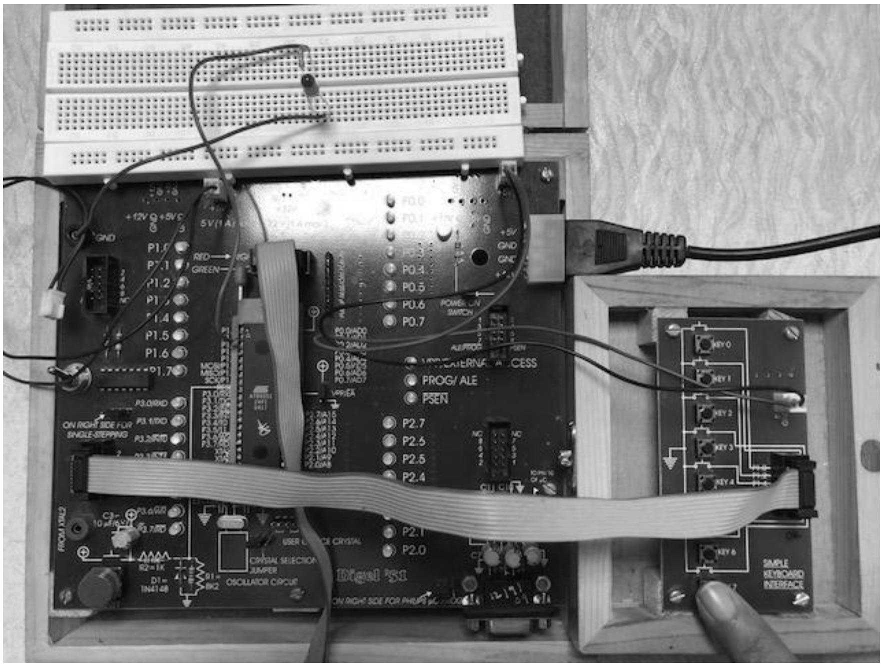
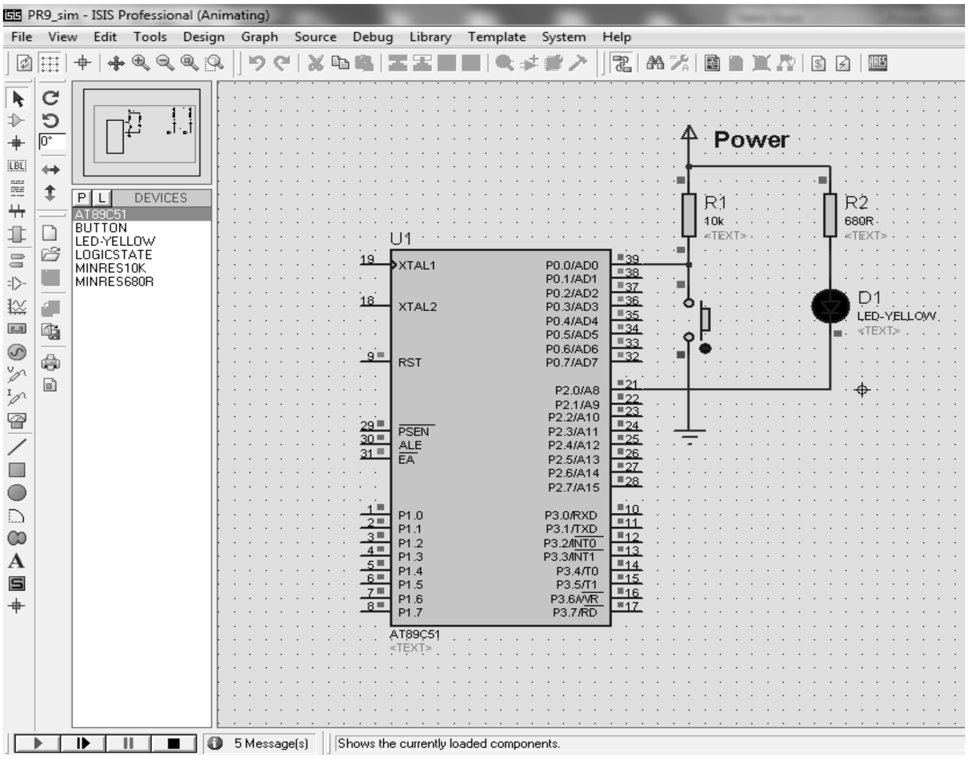


Figure 9.5: Practical Setup

c) **Simulation diagram**



AT89C51
BUTTON
LED-YELLOW
LOGICSTATE
MINRES10K
MINRES680R

5 Message(s)

Shows the currently loaded components.

Figure 9.6: 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 (89C51)	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.	Switch, LED	Push to ON, 5mm/3mm Red/Yellow color	1 No. each

X Precautions to be followed

1 Use always current limiting resistor between LED and Vcc.

XI Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop Embedded ‘C’ program using 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 Figure. 9.4
9. Press switch and observe LED On/Off status.

SAMPLE PROGRAM 1: **:** Develop and execute embedded ‘C’ program to read the status of key and turn ON a LED connected to port pin of 89C51 microcontroller and turn OFF after some delay.

Step 1: Algorithm

1. Initialize port pin P0.0 as input.
2. Initialize port pin P2.0 as output.
3. Read the status of key.
4. If key is closed then turn ON the LED.
5. Add some delay
6. Turn OFF the LED.
7. Repeat from step3.

Step 2: Flowchart

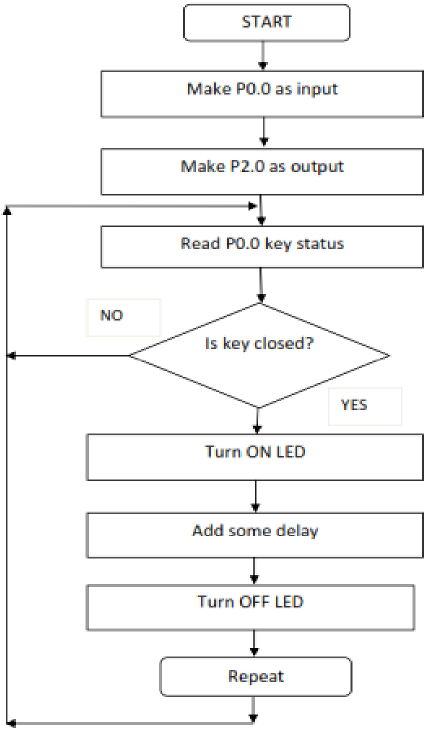


Figure 9.7: Flowchart for Key and LED

Step 3: ‘C’ Language Program	Comments
#include<reg51.h>	/* Special function register declarations */
	/* for the intended 8051 derivative */
sbit LED_pin=P2^0;	//Define LED PIN
sbit KEY_pin=P0^0;	//Define key PIN
void delay_ms(unsigned int);	//Function prototype declaration
void main (void)	
{	
KEY_pin=1;	//Make key pin input
LED_pin=0;	//Make LED pin output
LED_pin=1;	//Make LED off initially
	//Iinfinite loop
while(1)	
{	
if(KEY_pin == 0)	//If key pressed
{	
LED_pin = 0;	//LED ON
delay_ms(200);	//Delay
LED_pin = 1;	//LED OFF
}	
}	
}	

void delay_ms(unsigned int k)
{
unsigned int i, j;
for(i=0;i<k;i++)
{
for(j=0;j<1275;j++);
}
}
	//Delay function

Problem statement for student: Assume that bit P2.3 is an input and represents the condition of an oven (Oven door open or close). If door closed turn ON LED indicator which is connected to port pin P1.5.

Step 1: Algorithm

Step 2: Flowchart

Step 3- ‘C’ Language Program	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 (use blank sheet provided if space not sufficient)

LED will become _______ (ON/OFF) when oven door is closed.

XVI Result (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. Two switches are connected to pin P1.0 and P1.1 with two LED's to P2.0 and P2.1 respectively. Develop C code to turn ON LED1 when switch1 is pressed and turn ON LED2 when switch2 is pressed.
2. Using DMM measure the supply voltage given to development board:
   i. Check the voltage at supply pins.
   ii. Check voltage at pin 40 of microcontroller IC.
3. Refer Figure 9.4, change the LED connections. Connect anode of LED to P2.0 and cathode to ground. Modify the program given in Step 3 and turn ON the LED.

[Space for Answers]

....

....

....

....

....

....

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

XX References / Suggestions for further reading

1. https://aticleworld.com/interfacing-of-switch-and-led-using-the-8051/
2. https://electrosome.com/8051-keilc-push-button-switch-at89c51/
3. https://en.wikipedia.org/wiki/Light-emitting_diode
4. https://en.wikipedia.org/wiki/Switch

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%**

Names of student Team Member

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

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

Practical No. 10: Interface 89C51/AVR microcontroller and write the ‘C’ program to display numbers from 0 to 9 on 7-segment display with specified delay.

I Practical Significance

Seven segment displays are used to indicate numerical information. Seven segments display can display digits from 0 to 9 and even we can display few characters like A, B, C, D, E, F, G, H etc. These are very popular and have many more applications like pricing menu at petrol pump, in metros, digital clocks, electronic meters, Instrument panels. This practical will help the students to develop coding 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.
- **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 Embedded Systems’.**

- Develop ‘C’ code using IDE tools.
- Use Input/output port pins of microcontroller.
- Interface 7 segment displays to microcontroller in static or multiplexed mode.

IV Relevant Course Outcome(s)

- Develop basic applications using embedded systems.

V Practical Outcome

- Interface 89C51/AVR microcontroller and write the ‘C’ program to display numbers from 0 to 9 on 7-segment display with specified delay.

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

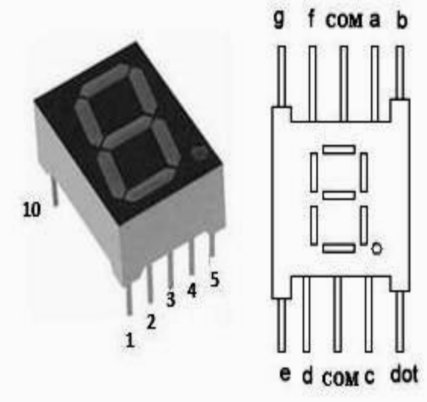


Figure 10.1: Seven segment display and pin configuration

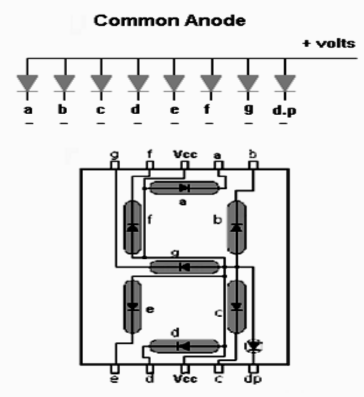


Figure 10.2: CA 7-segment display constructional diagram

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

Table 10.1: 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	0xC0
1	1	1	1	1	1	0	0	1	0xF9
2	1	0	1	0	0	1	0	0	0xA4
3	1	0	1	1	0	0	0	0	0xB0
4	1	0	0	1	1	0	0	1	0x99
5	1	0	0	1	0	0	1	0	0x92
6	1	0	0	0	0	0	1	0	0x82
7	1	1	1	1	1	0	0	0	0xF8
8	1	0	0	0	0	0	0	0	0x80
9	1	0	0	1	0	0	0	0	0x90

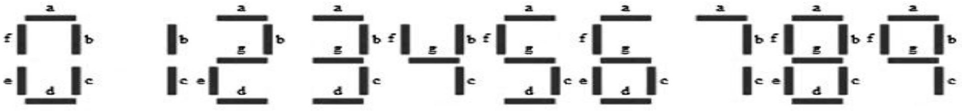


Figure 10.3: Seven segment display number patterns for common Anode

Note: For Common Anode LED segment is ON for logic ‘0’ (Ground) and OFF for logic ‘1’ (+5V).

VIII Practical Circuit diagram:

a) Sample Circuit diagram

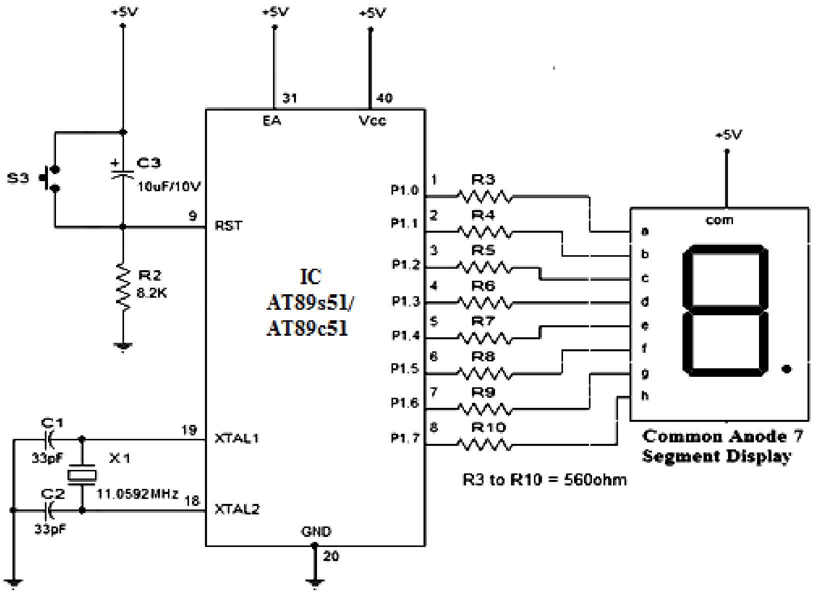


Figure 10.4: 89C51 connection to CA seven segment display

b) Practical setup

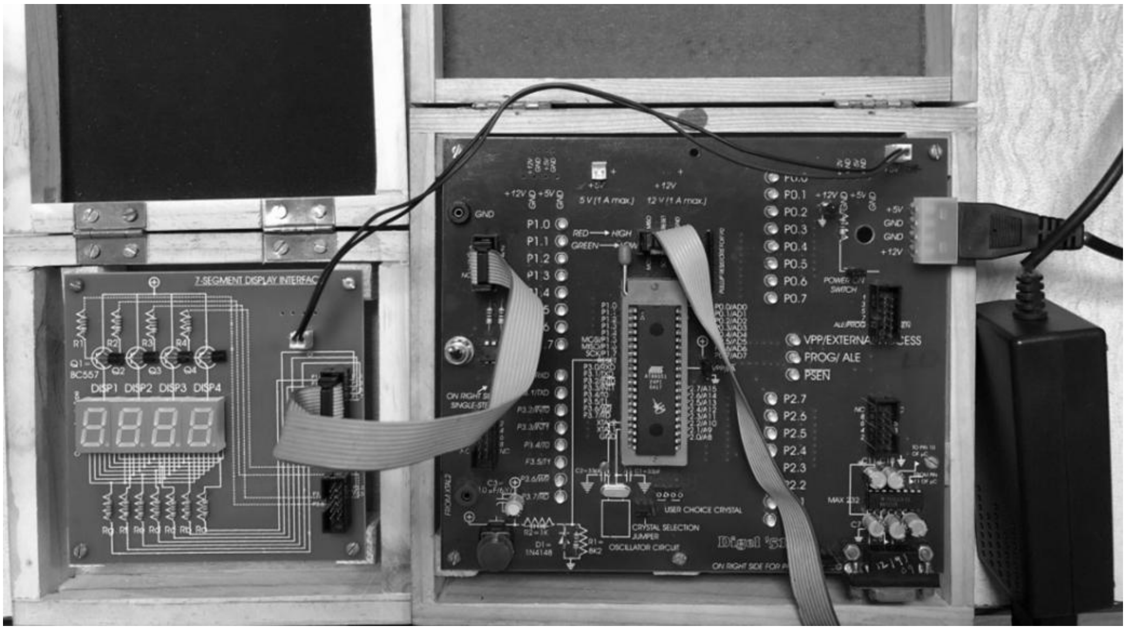
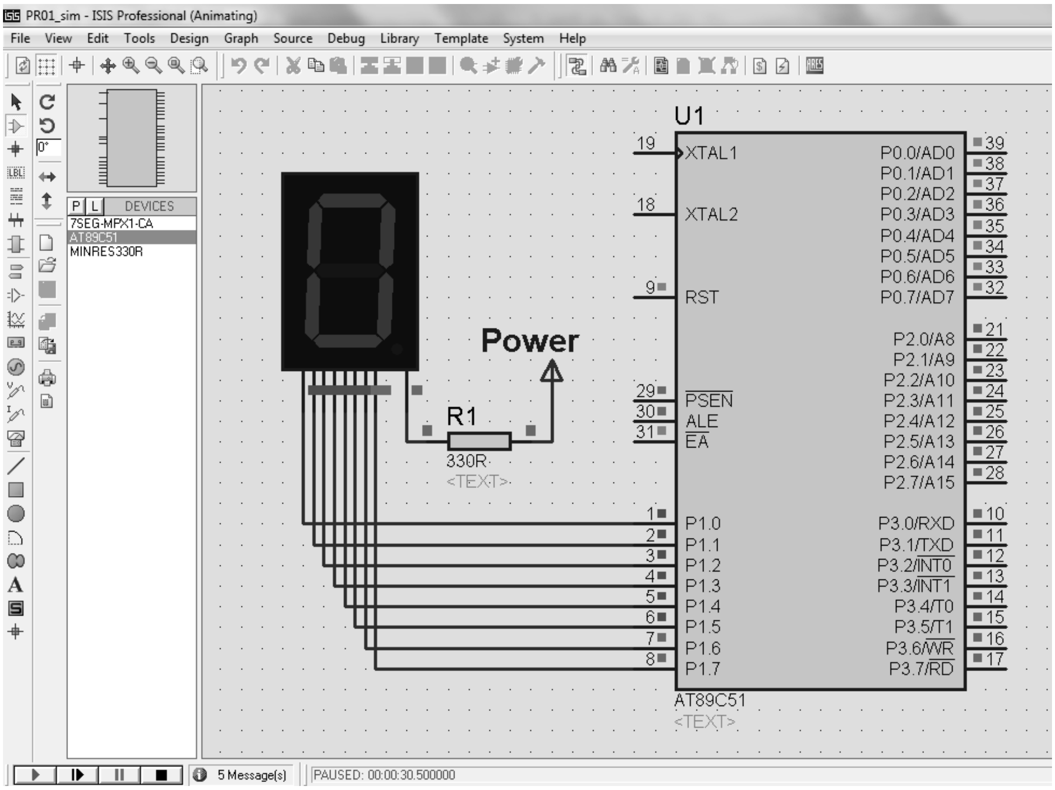


Figure 10.5: Practical Setup

c) Simulation diagram



PAUSED: 00:00:30.500000

Figure 10.6: 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 (8051)	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 for the same.
3. Develop C program using 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 7- segment display to microcontroller as per circuit diagram shown in Figure. 10.4
9. Observe numbers on 7 segment display.

SAMPLE PROGRAM : Develop and execute embedded 'C' program to display numbers from 0 to 9 on common anode 7 segment display with delay.

Step 1: Algorithm

1. Make port P1 as output.
2. Initialize all the segment hex values of the digits in an array.
   example:
   unsigned char arr[10]= {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x67};
   Take for loop and assign array values to the PORT1
3. Add time delay.
4. Repeat the process from step 2.

Step 2: Flowchart

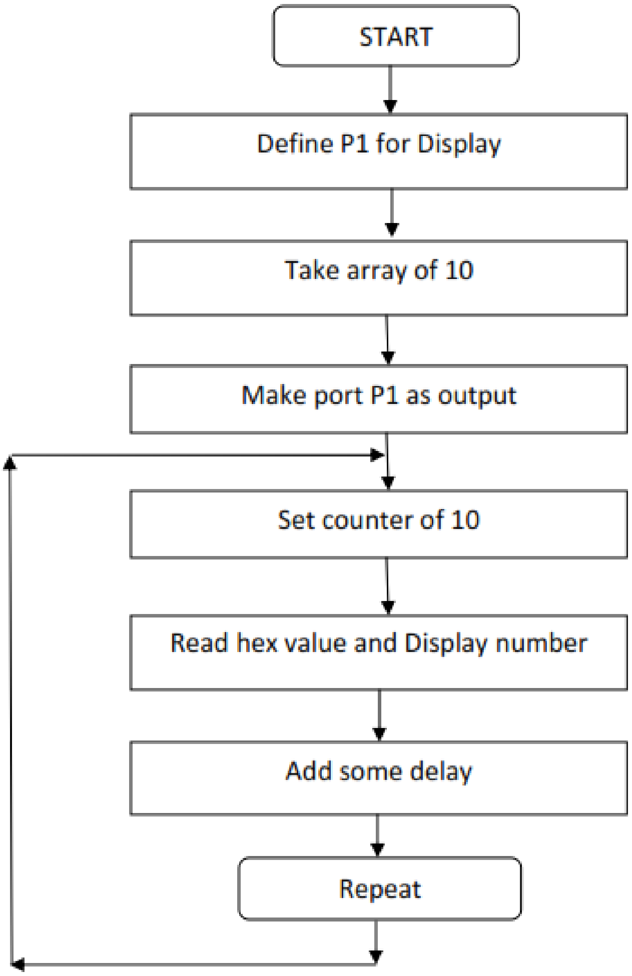


Figure 10.7: Flowchart for display number 0 to 9 on 7-segment

Step 3: 'C' Language Program	**Comments**
#include<reg51.h>	/*Special function register declarations */
	/* for the intended 8051 derivative */
#define Display_7seg P1	
void delay_ms(unsigned int);	//Function prototype declaration
void main (void)	
{	
unsigned char counter;	
unsigned char	
arr[10]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,	
0x82,0xf8,0x80,0x98};	
P1=0x00;	//Make port P1 output
while(1)	//Forever loop
{	
for (counter=0;counter<10;counter++)	
{	
Display_7seg=arr[counter];	
delay_ms(200);	
}	
}	
}	
void delay_ms(unsigned int k)	//Delay function
{	
int i,j;	
for(i=0;i<k;i++)	
{	
for(j=0;j<1275;j++);	
}	
}	

Problem statement for student: Assume bit P2.0 is an input to microcontroller which represents the condition of water tank empty and P2.7 input for water tank Full.

1. When the tank is empty then input P2.0 goes low display 'E'.
2. When the tank is full then input P2.7 goes low display 'F' on 7-segment connected to port P1.

Step 1: Algorithm

Step 2: Flowchart

Step 3- ‘C’ Language Program	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 (use blank sheet provided if space not sufficient)

Write the character/number for CA 7 segment display to following hex codes

Hex code	Character/Number
0x86	
0x8E	

XVI Result (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. Tabulate the hex values for common cathode 7 segment display.
2. Modify given sample program ‘C’ code to common cathode 7-segment display and observe the output.

[Space for Answers]

....

....

....

....

....

....

....

....

XX References / Suggestions for further reading

1. https://www.electronicshub.org/interfacing-7-segment-display-8051/
2. https://en.wikipedia.org/wiki/Seven-segment-display
3. https://pdf1.alldatasheet.com/datasheet-pdf/view/162375/ETC1/LDS-A516RI

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%**

Names of student Team Member

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

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

Practical No. 11: Interface 89C51/AVR microcontroller and write C program to display string on given 16 x 2 LCD.

I Practical Significance

LCDs are used in a wide range of applications to display message on printer, coffee machine, remote, computers, televisions, instrument panels, calculator etc. They are common in consumer devices and have replaced cathode ray tube (CRT) displays in most applications. This practical will help the students to develop skills to interface LCD to microcontrollers in embedded systems 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.
- **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 Embedded Systems'.**

- Develop 'C' code using IDE tools.
- Use Input/output port pins of microcontroller.
- Interface LCD to microcontroller.

IV Relevant Course Outcome

- Develop basic applications using embedded systems.

V Practical Outcome

- Interface 89C51/AVR microcontroller and write C program to display string on given 16 x 2 LCD.

VI Relevant Affective domain related Outcome(s)

- Follow safe practices.
- 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.

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

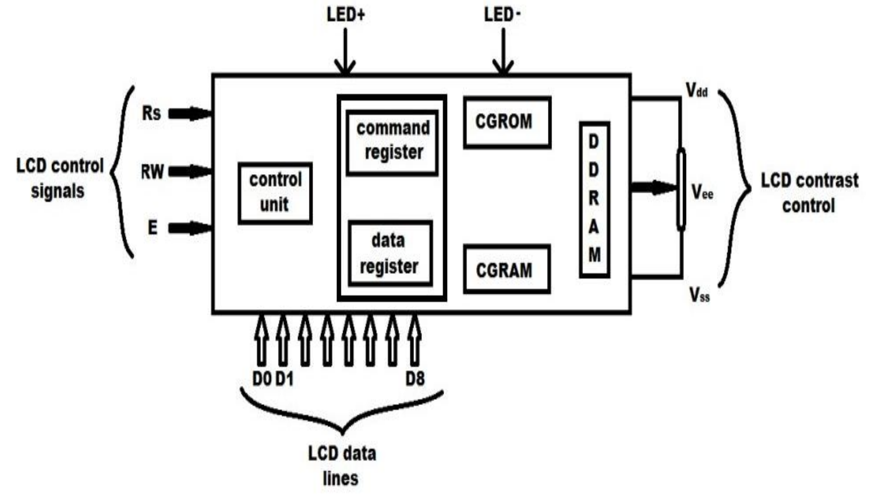


Figure. 11.1: Block Diagram of LCD display

Specifications: 16x2 LCD display

- Part No: JHD162A
- Char Dots : 5 x 8
- Number of data line: 8 bit parallel
- Available Types: (Yellow, Green, Grey, B/W)

Table 11.1: 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)

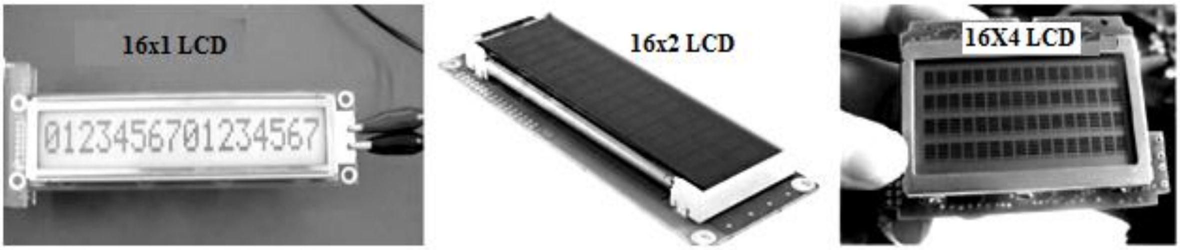


Figure 11.2: Types of LCD

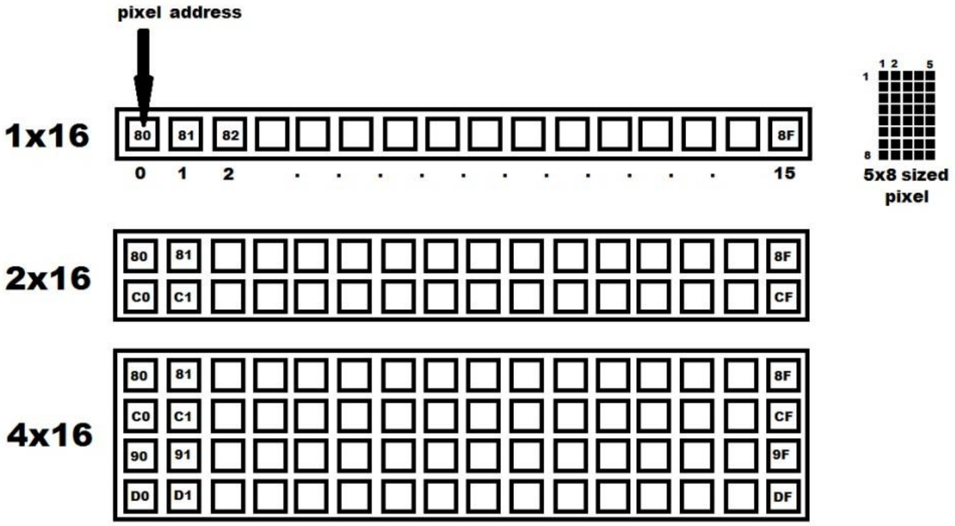


Figure 11.3: LCD pixel format

Table 11.2: LCD control lines and function

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

Rs=1 -Data register
Rs=0 -Command
	RW=1 -Read from LCD.
	RW=0 -Write to LCD.
		EN= high to low (Logic '1' delay Logic '0') for use LCD module.

Table 11.3: LCD commands

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

VIII Practical Circuit diagram:

a) Sample Circuit diagram

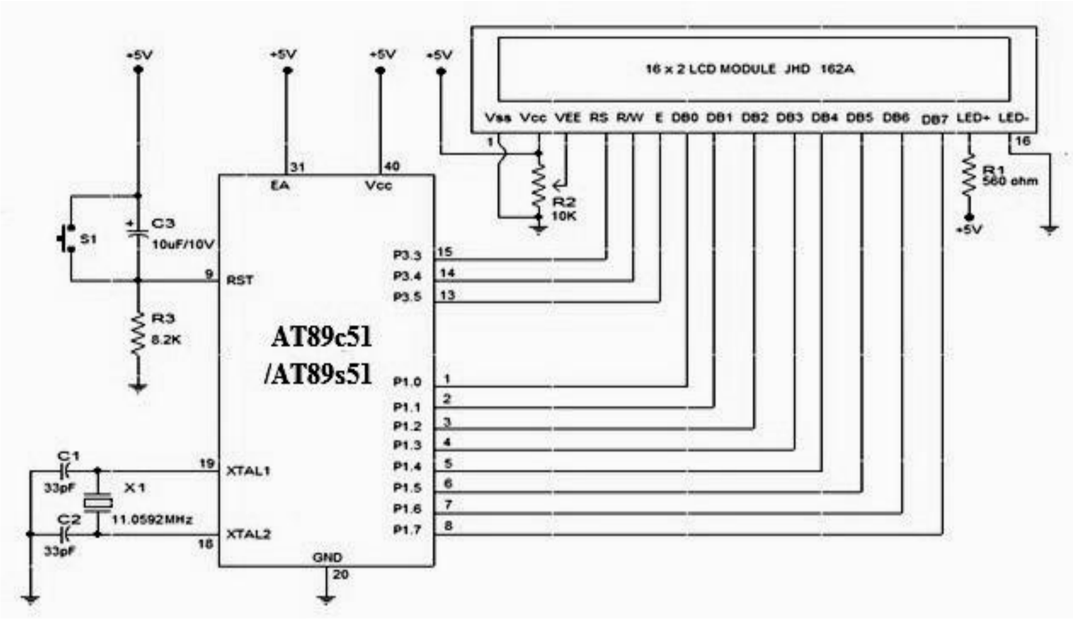


Figure 11.4: 89C51 connection to LED and switch

b) Practical setup

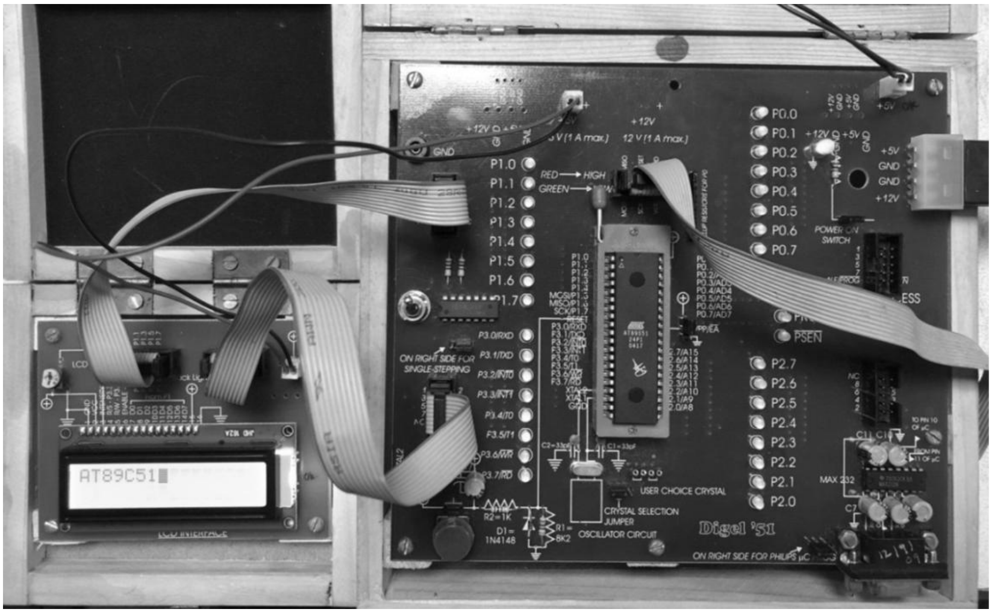


Figure 11.5: Practical Setup

c) Simulation diagram

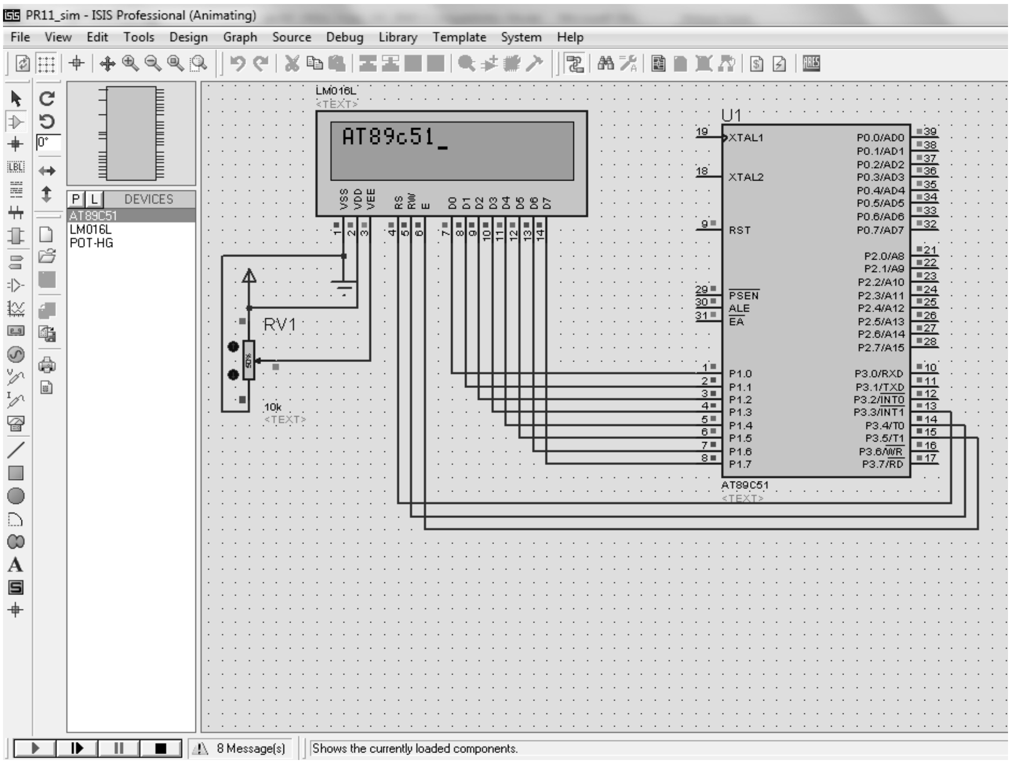


Figure 11.6: Simulation diagram

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 (89C51)	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.	LCD Trainer board	Suitable to interface with 8051 trainer kit	1 No.

X Precautions to be followed

1. LCD panel is made up of glass avoid applying strong pressure on to the surface of display area.

XI Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop embedded ‘C’ program using 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 LCD to microcontroller as per circuit diagram shown in Figure. 11.4
9. Observe characters and numbers displayed on LCD.

SAMPLE PROGRAM 1: **:** Develop and execute C language program for 16x2 LCD and display “AT89c51” on it.

Step 1: Algorithm

1. Initialize the LCD using the LCD initialization commands.
2. Set the address of DDRAM, where you want to display the character.
3. Write the character on the data bus of LCD.
4. If passing the command then make RS pin low (RS = 0) either for data make RS pin high (RS = 1).
5. Set Enable pin high (EN = 1).
6. Write the command or data on the data bus.
7. Set Enable pin low (EN = 0).

Step 2: Flowchart

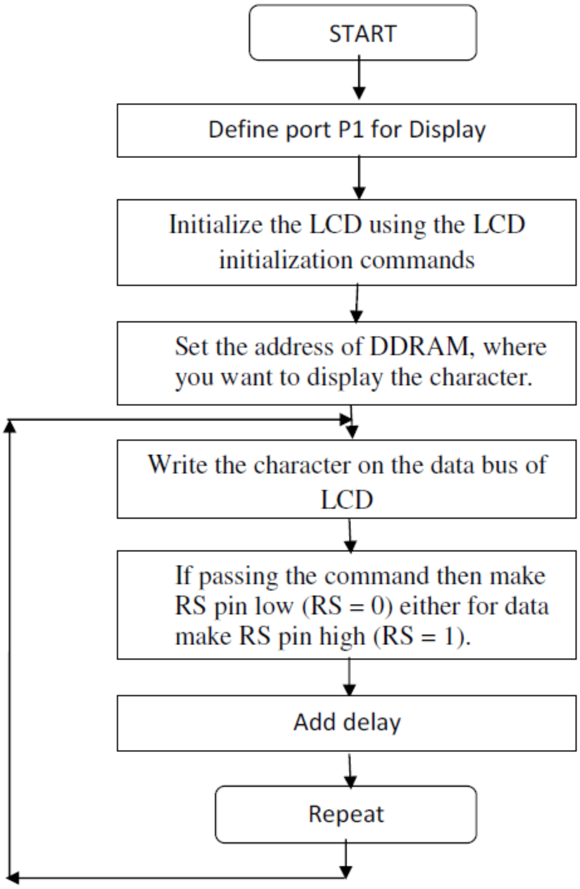

START
Define port P1 for Display
Initialize the LCD using the LCD initialization commands
Set the address of DDRAM, where you want to display the character.
Write the character on the data bus of LCD
If passing the command then make RS pin low (RS = 0) either for data make RS pin high (RS = 1).
Add delay
Repeat

Figure 11.7: Flowchart for 16x2 display

Step 3: 'C' Language Program	**Comments**
#include<reg51.h>	/* special function register declarations */
sfr ldata =0x90;	/* for the intended 8051 derivative */
sbit rs=P3^3;	//Define sfr address P1
sbit rw=P3^4;	
sbit en=P3^5;	
sbit busy=P1^7;	
void lcdcmd(unsigned char);	
void lcddata(unsigned char);	
void lcdready(void);	
void Delay(unsigned int);	
void main(void)	
{	
ldata=0x00;	//Make port as output
lcdcmd(0x38);	//Commands to initialize LCD
lcdcmd(0x0E);	
lcdcmd(0x06);	
lcdcmd(0x01);	
while(1)	
{	
lcdcmd(0x80);	// Force cursor to beginning of 1st line
lcddata('A');	//Data for Display
lcddata('T');	
lcddata('8');	
lcddata('9');	
lcddata('c');	
lcddata('5');	
lcddata('1');	
Delay(250);	//Add some delay
}	
}	
void lcdcmd(unsigned char value)	
{	
lcdready();	
ldata=value;	
rs=0;	
rw=0;	
en=1;	
Delay(1);	
en=0;	
}	

```
void lcddata(unsigned char value)
{
 lcdready();
 ldata=value;
 rs=1;
 rw=0;
 en=1;
 Delay(1);
 en=0;
}

void lcdready(void)                        //Busy flag checking
{
 busy=1;
 rs=0;
 rw=1;
 while(busy==1)
  {
   en=0;
   Delay(1);
   en=1;
  }
 }

void Delay(unsigned int k)                   //Delay function
{
 int i,j;
 for(i=0;i<k;i++)
   {
    for(j=0;j<1275;j++);
   }
 }
```

Problem statement for student: Develop and execute embedded ‘C’ program for 16x2 LCD to display your Polytechnic name “ __________ ”on LCD.

Step 1: Algorithm

Step 2: Flowchart

Step 3- ‘C’ Language Program	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 (use blank sheet provided if space not sufficient)

Write displayed character on 16x2 LCD ____________.

XVI Result (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. State necessity of busy flag checking in LCD. Give alternate solution for it.
2. Write appropriate command for following.
   a. Display on, cursor blinking
   b. Clear display
   c. Shift cursor position to left
3. Give DDRAM address for 1st,2nd, 3rd, 4th line of 16x4 LCD display.

[Space for Answers]

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

....

....

....

....

....

....

....

XX 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

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%**

Names of student Team Member

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

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

Practical No. 12: Interface 89C51/AVR microcontroller and write 'C' language program to read key code from 4 x 4 matrix keyboard and display on LCD.

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. 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 Embedded Systems'.**

- Develop 'C' code using IDE tools.
- Use Input/output port pins of microcontroller.
- Interface 4x4 matrix keypad to microcontroller.

IV Relevant Course Outcome

- Develop basic applications using embedded systems.

V Practical Outcome

- Interface 89C51/AVR microcontroller and write 'C' language program to read key code from 4 x 4 matrix keyboard and display LCD.

VI Relevant Affective domain related Outcome(s)

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

VII Minimum Theoretical Background

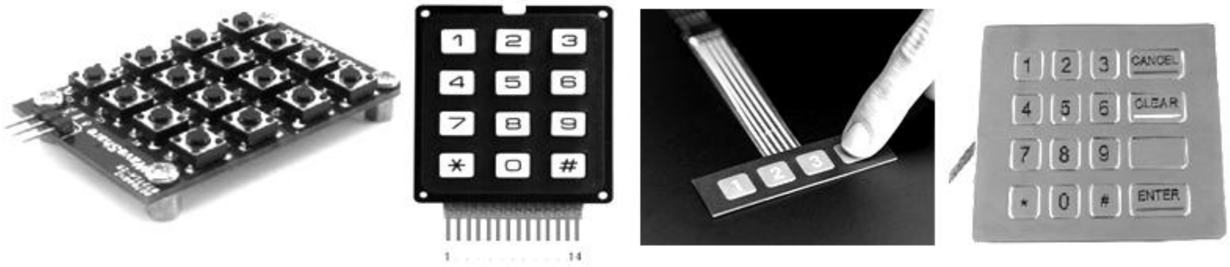


Figure 12.1: Keypad Types

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

Table 12.1: 4x4 Keypad: - Rows (R1, R2, R3, R4) Columns (C1, C2, C3, C4) (Refer Figure.12.2)

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

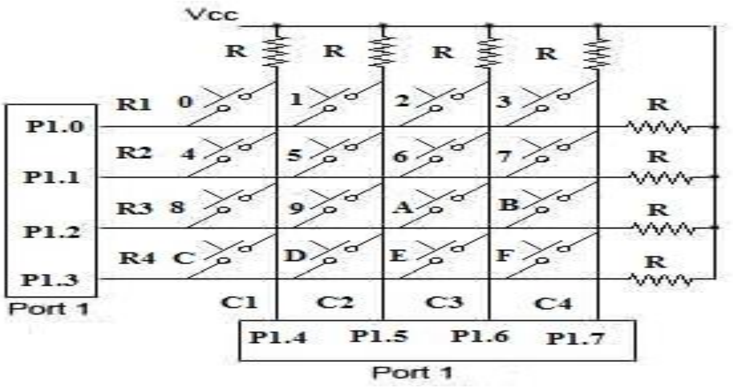


Figure 12.2: 89C51 connection to 4x4 matrix keypad

b) Practical setup

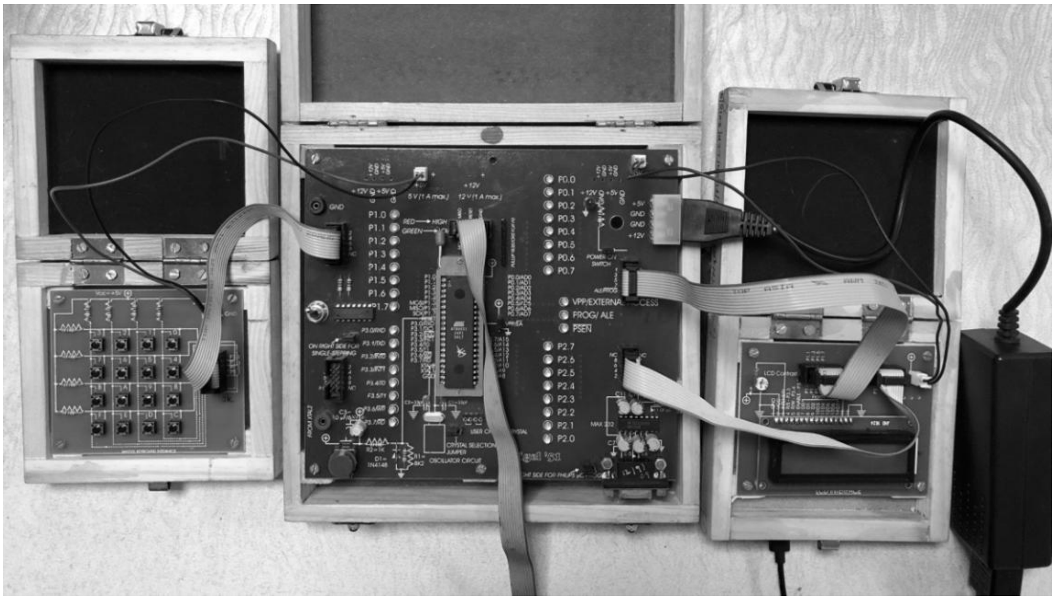


Figure 12.3: Practical Setup

c) Simulation diagram

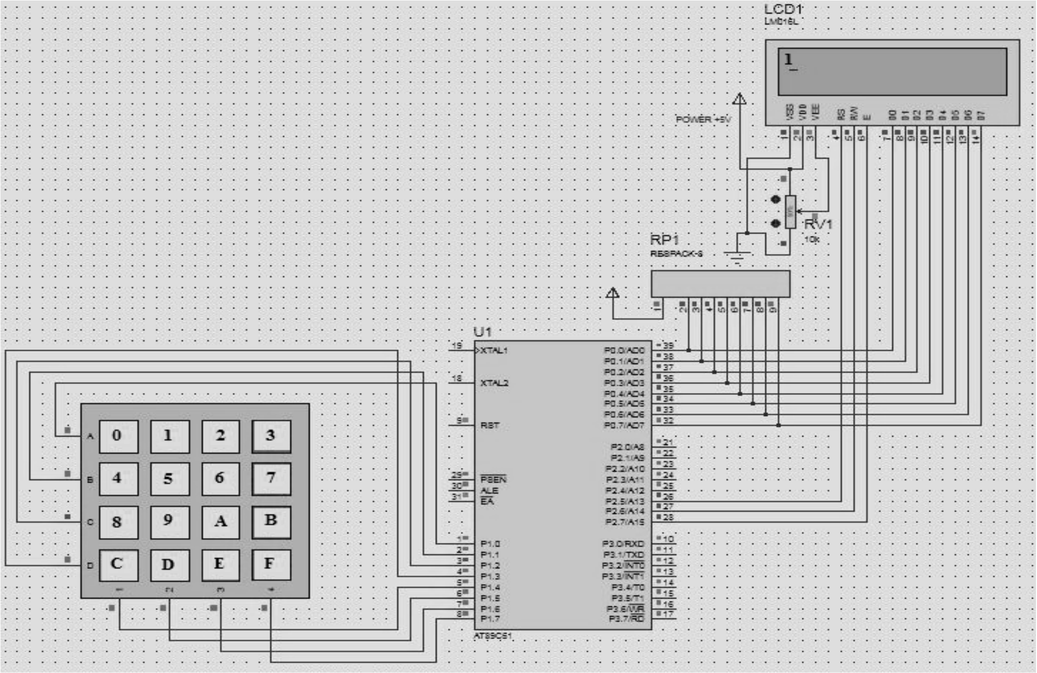


Figure 12.4: Simulation diagram

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 (89C51)	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.	Keyboard 4x4 trainer board	Suitable to interface with 8051 trainer kit	1No.
4.	16x2 LCD Display board	Suitable to interface with 8051 trainer kit	1No.

X Precautions to be followed

1. Use always current limiting resistor between Vcc and LED.

XI Procedure

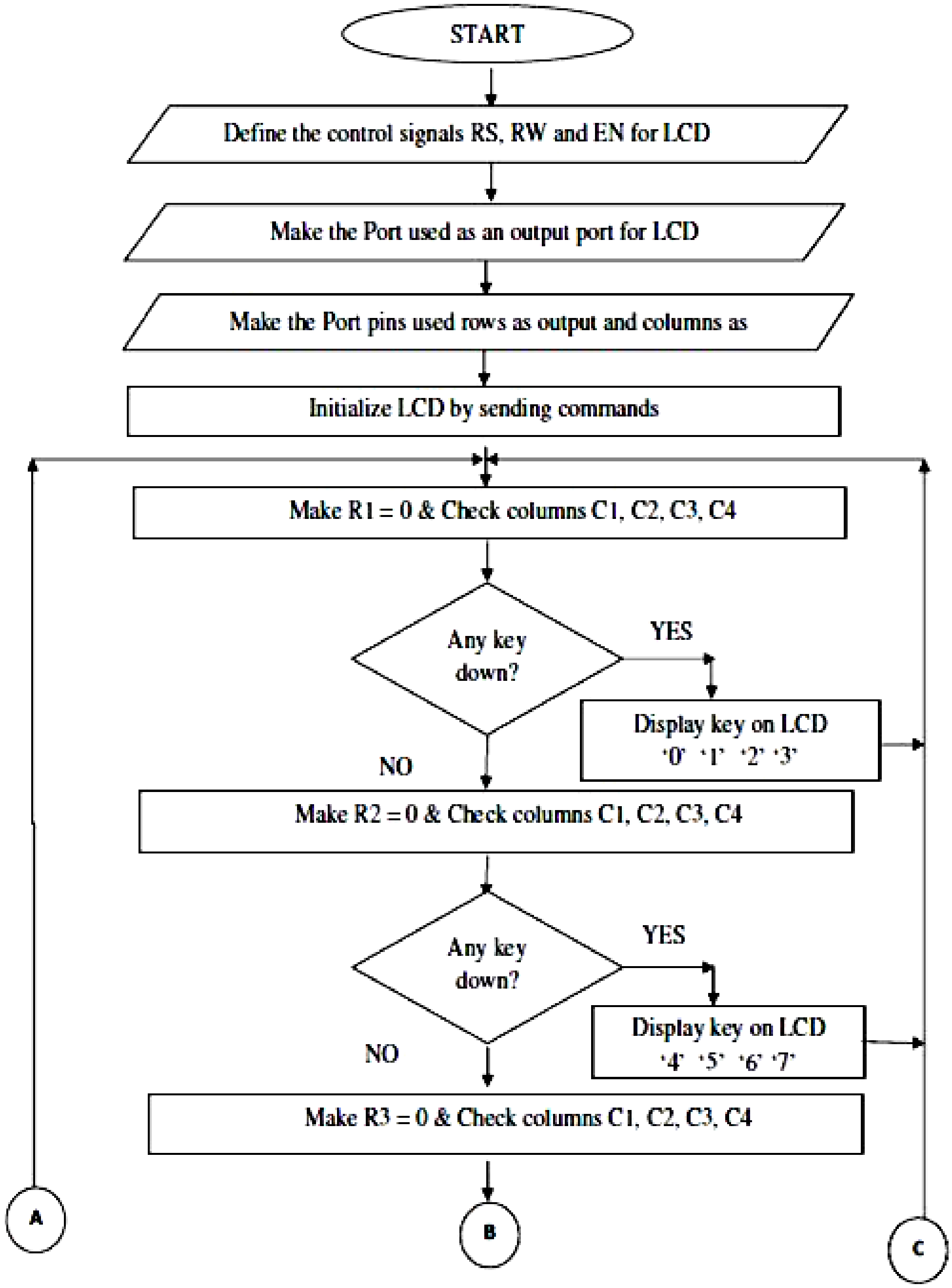
1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop embedded ‘C’ program using Keil IDE or any other relevant software tool.
4. Debug program on IDE.
5. Execute program on IDE.
6. Create hex file for the program.
7. Download hex code in EPROM/Flash memory of microcontroller.
8. Interface 4x4 keypad to microcontroller as per circuit diagram shown in Figure. 12.2
9. Press any key and observe number on 16x2 LCD.

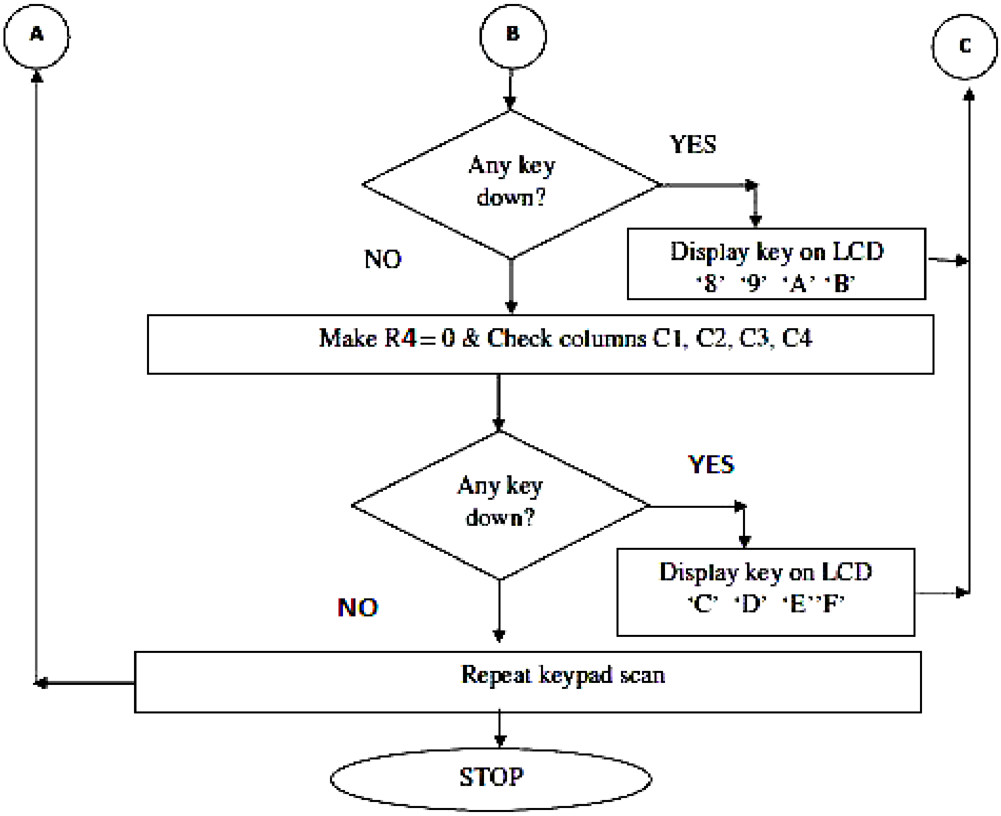
SAMPLE PROGRAM 1: **:** Develop and execute C language program to read the status of 4x4 keypad and display on 16x2 LCD.

Step 1: Algorithm

1. Initialize port pin P0 as output.
2. Initialize port pin P1 as input.
3. Initialize LCD by sending commands.
4. Set DDRAM address for to display pressed key.
5. Read the status of matrix keypad.
6. If any key down display on LCD
7. Clear LCD.
8. Repeat from step4.

Step 2: Flowchart


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


A
B
C
Any key down?
YES
NO
Display key on LCD '8' '9' 'A' 'B'
Make R4=0 & Check columns C1, C2, C3, C4
Any key down?
YES
NO
Display key on LCD 'C' 'D' 'E' 'F'
Repeat keypad scan
STOP

Step 3: 'C' Language Program	**Comments**
#include <reg51.h>	/* Special function register declarations */
	/* for the intended 8051 derivative */
#define DATA P0	//Define DATA to Port0
void lcd_init(void);	
void lcd_cmd(unsigned char);	
void lcd_display(unsigned char);	
void key_scan(void);	
void delay_ms(unsigned int);	
sbit RS = P2^5;	
sbit RW = P2^6;	//Register Select
sbit en = P2^7;	//LCD Read/Write
	//LCD Enable
unsigned char R,C,ch;	//Matrix Keypad Character Initialization
unsigned char Key[4][4] =	
{'0','1','2','3',	
'4','5','6','7',	
'8','9','A','B',	
'C','D','E','F'};	
void main()	
{	
P1=0xff;	//Make input port

```
     DATA=0x00;                             //Make output port
    lcd_init();       delay_ms(100);        //Call LCD initialization function
    lcd_cmd(0x80);
    while(1)
    {
    key_scan();       delay_ms(5);
    ch = Key[R][C];  delay_ms(5);
    lcd_display(ch);   delay_ms(100);       //Assign Key value to ch_disp

    lcd_cmd(0x01);
    }
  }
    void key_scan(void)                     //Keypad scan
    {                                       //Scanning for Row Value
     P1 = 0x0F;
     while(P1 == 0x0F);                     //Initialize Port1 to 0Fh
            if(P1 == 0x0E)                  //Checking for Row0
            R = 0;
            else if(P1 == 0x0D)             //Checking for Row1
            R = 1;
            else if(P1 == 0x0B)             //Checking for Row2
            R = 2;
            else if(P1 == 0x07)             //Checking for Row3
            R = 3;
                                            //Scanning for Column Value
            P1 = 0xF0;                      //Initialize Port1 to F0h
            while(P1 == 0xF0);
            if(P1 == 0xE0)                  //Checking for Column0
            C = 0;
            else if(P1 == 0xD0)             //Checking for Column1
            C = 1;
            else if(P1 == 0xB0)             //Checking for Column2
            C = 2;
            else if(P1 == 0x70)             //Checking for Column3
            C = 3;
            delay_ms(20);
    }
    void lcd_cmd(unsigned char cmnd)        //LCD command function
     {
            DATA = cmnd;
            RS = 0;RW = 0;en = 1;
            delay_ms(10);
            en = 0;
     }
    void lcd_display(unsigned char dat)       //LCD data function
    {
            DATA = dat;
            RS = 1;RW = 0;en = 1;
            delay_ms(10);
            en = 0;
  }
void lcd_init(void)                                  //LCD Initialization
{
lcd_cmd(0x38);          delay_ms(10);                //2 lines and 5x7 matrix
lcd_cmd(0x0e);          delay_ms(10);                //Display on, cursor blinking
lcd_cmd(0x06);          delay_ms(10);                //Shift Cursor to right
lcd_cmd(0x01);          delay_ms(10);                //Clear display screen
}

void delay_ms(unsigned int k)

                                                     //Delay function
{
unsigned int i,j;
for(i=0;i<=k;i++)
 {
  for(j=0;j<1275;j++)  ;
 }
}
```

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 (use blank sheet provided if space not sufficient)

Identify the row and column of the pressed key for logic

R3-R0 = 1110, C3-C0 = 1011	
R3-R0 = 1101, C3-C0 = 0111	

XVI Result (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. In Figure. 12.2 to detect the key press, which of the following is grounded?
   a. all rows
   b. one row at time
   c. both a) and b)
2. Identify the steps to detect the key press.
3. Suggest method to avoid key bounce problem.
4. Name the cable used to connect Development board to the Keypad and LCD.
5. List the number of pins required to Interface following matrix keypad:
   i) 4x1 Keypad
   ii) 4x3 Keypad

[Space for Answers]

....

....

....

....

....

....

....

....

....

XX References / Suggestions for further reading

1. https://en.wikipedia.org/wiki/Keypad
2. https://components101.com/misc/4x4-keypad-module-pinout-conFigureuration-features-datasheet

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%**

Names of student Team Member

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

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

Practical No. 13: Interface 89C51/AVR microcontroller and write C program to convert analog signal into digital form using given 8 bit ADC and store the converted digital data in memory.

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 8 bit ADC to the microcontroller and store the converted digital data in memory.

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 Embedded Systems'.**

- Develop 'C' code using IDE tools.
- Use Input/output port pins of microcontroller.
- Interface ADC to microcontroller.

IV Relevant Course Outcome(s)

- Develop basic applications using embedded systems.

V Practical Outcome

- Interface 89C51/AVR microcontroller and write C program to convert analog signal into digital form using given 8 bit ADC and store the converted digital data in memory.

VI Relevant Affective domain related Outcome(s)

- Follow safety practices.
- Practice good housekeeping.
- 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: Sound, light, temperature, pressure which may be represented electrically by an analog voltage or current.

Table 13.1: Specifications ADC0808 Chip

Parameter	Typical Values
Resolution	8 bits
Input channels	8
Single supply	5V DC
Power consumption	15 mW (Low)
Clock frequency range	10 KHz to 1280 KHz (Typically 680 KHz)

Table 13.2: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

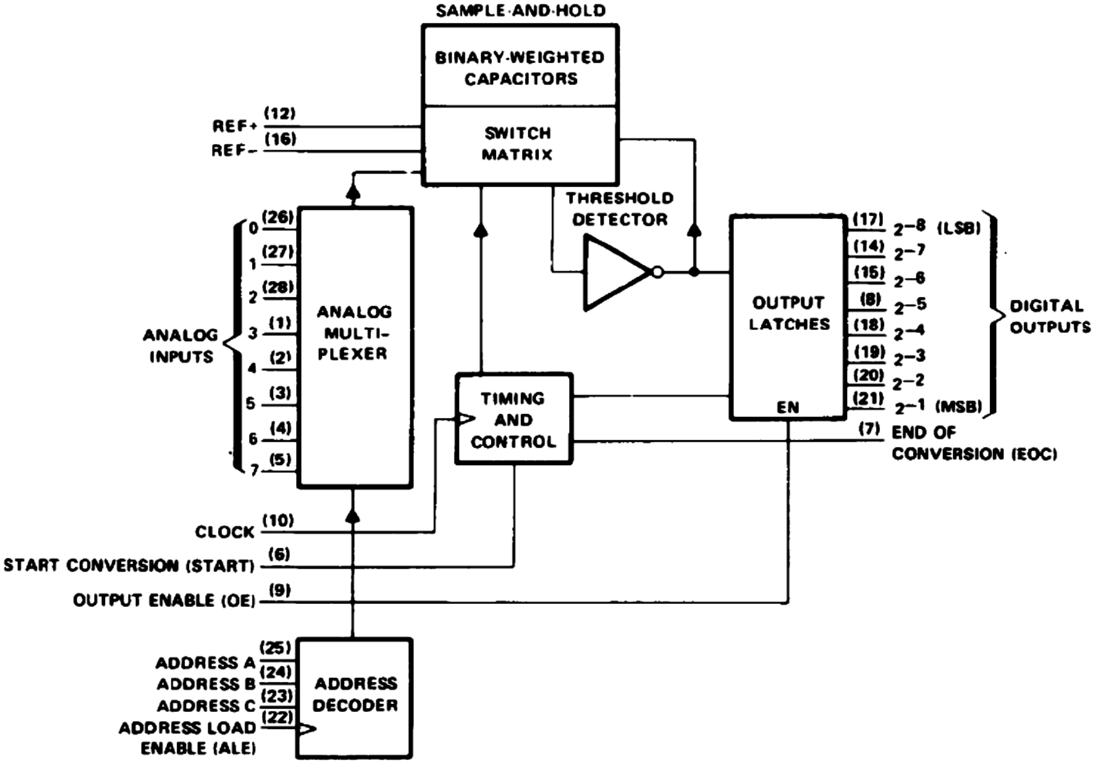


Figure 13.1: Functional block diagram of ADC 0808 chip

VIII Practical Circuit diagram:

a) Sample Circuit diagram

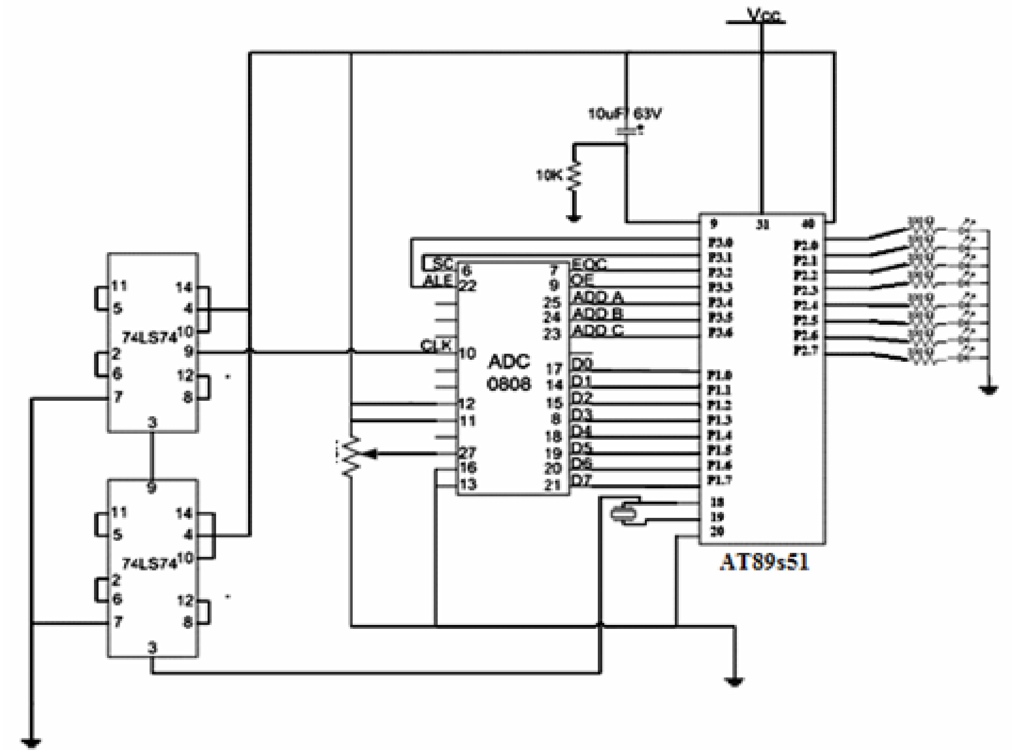


Figure 13.2: 89C51 connection to LED and switch

b) Practical setup

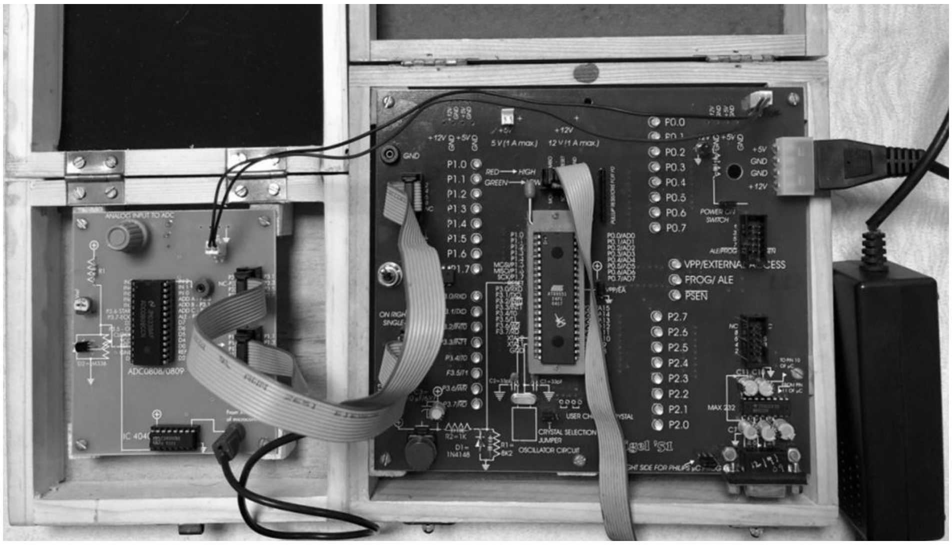


Figure 13.3: Practical Setup

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 (89C51)	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.
1.	Desktop PC	Loaded with open source IDE, simulation and program downloading software.	1 No.
2.	ADC (0808) trainer board	Suitable to interface 8051 board.	1 No.

X Precautions to be followed

1. Refer datasheet to provide clock frequency to ADC 0808 chip.
2. Care must be taken while taking observations during power up.

XI Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop embedded ‘C’ program using 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 ADC0808 to microcontroller as per circuit diagram shown in Figure.13.2
9. Observe digital data at declared variable in the sample program.

SAMPLE PROGRAM 1: **:** Develop and execute C program to read the analog data from ADC and store the converted digital data in memory.

Step 1: Algorithm

1. Select an analog channel by providing bits to A, B, and C addresses according to the analog signal selection given in table 13.2
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

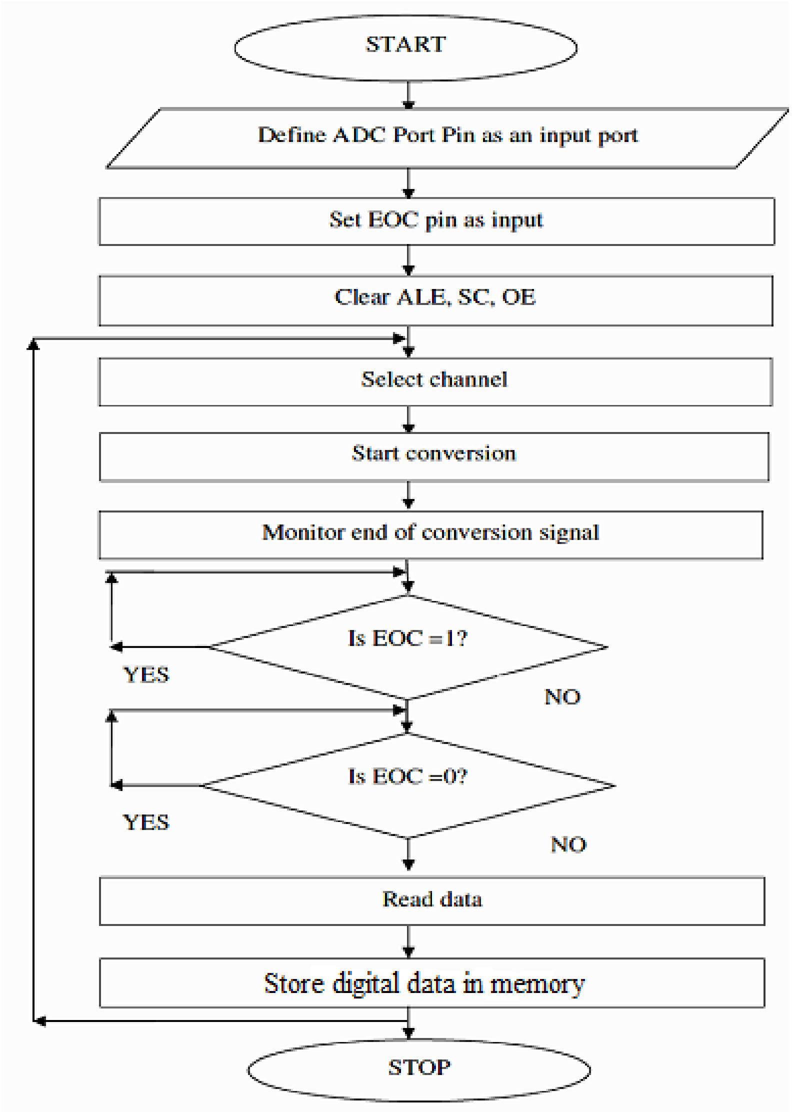


Figure 13.4: Flowchart to read data from ADC chip

Step 3: 'C' Language Program	Comments
#include <reg51.h>	/* Special function register declarations */
sfr ADC_DATA = 0x90;	/* For the intended 8051 derivative */
void delay_ms(unsigned int);	
sbit ALE=P3^0;	//Define ADC pins
sbit SC=P3^1;	
sbit EOC=P3^2;	
sbit OE=P3^3;	
sbit ADDR_A=P3^4;	
sbit ADDR_B =P3^5;	
sbit ADDR_C =P3^6;	
void main(void)	
{	

```
 unsigned char value;
ADC_DATA =0xff;                    //Make P1 as input
EOC=1;                             //Make EOC as input
ALE=0;                             //Clear ALE
OE=0;                              //Clear OE
SC=0;                              //Clear SC

While(1)
 {
 ADDR_A=1;                         //Select CH 1
 ADDR_B=0;
 ADDR_C=0;
 delay_ms(1);
 ALE=1;
 delay_ms(1);
 SC=1;                             //Start conversion
 delay_ms(1);
 ALE=0;
 SC=0;
 while(EOC==1);
 while(EOC==0);                    //Wait for data conversion
 OE=1;                             //Enable read
 delay_ms(1);
 value=ADC_DATA;                   //Store data
 OE=0;
 P2=value;
 }
}
void delay_ms(unsigned int k)      //Delay function
 {
  unsigned int i, j;
  for(i=0;i<k;i++)
  {
    for(j=0;j<1275;j++);
  }
 }
```

Problem statement for student: Assume that temperature sensor LM 35 is connected to ADC chip. Develop and execute C program to read the analog data from ADC and display the converted digital data on LED's connected to port P2.

Step 1: Algorithm

Step 2: Flowchart

Step 3- ‘C’ Language Program	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 (use blank sheet provided if space not sufficient)

Input Voltage	Output HEX Value
1V	
2V	
3V	
4V	
5V	

XVI Result (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 find the step size of ADC0808.
2. Draw timing diagram for selecting a channel and read data for ADC 0808.
3. If ADDR_A, ADDR_B, ADDR_C= 111 then name the channel which get selected.
4. Write the necessity to monitor EOC signal for ADC 0808.

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XX 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

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%**

Names of student Team Member

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

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

Practical No. 14: Interface 89C51 and write C program to generate square, saw tooth waveforms using given 8 bit DAC.

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 Embedded Systems'.**

- Develop 'C' code using IDE tools.
- Use Input/output port pins of microcontroller.
- Interface 8 bit DAC to microcontroller.

IV Relevant Course Outcome

- Develop basic applications using embedded systems.

V Practical Outcome

- Interface 89C51 and write C program to generate square, saw tooth waveforms using given 8 bit DAC.

VI Relevant Affective domain related Outcome(s)

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

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.

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 Iref current is generally set to 2.0mA.

Two methods adopted for designing a DAC

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

Table No.14.1: Specifications DAC 0808 chip

Parameter	Typical Value
DAC 0808	8-bit
Full scale output current settling time	150 ns
Power dissipation	33 mW with ±5V supplies
Relative accuracy	±0.19% error maximum
High speed multiplying input slew rate	8 mA/µs
Power supply voltage range	±4.5V to ±18V

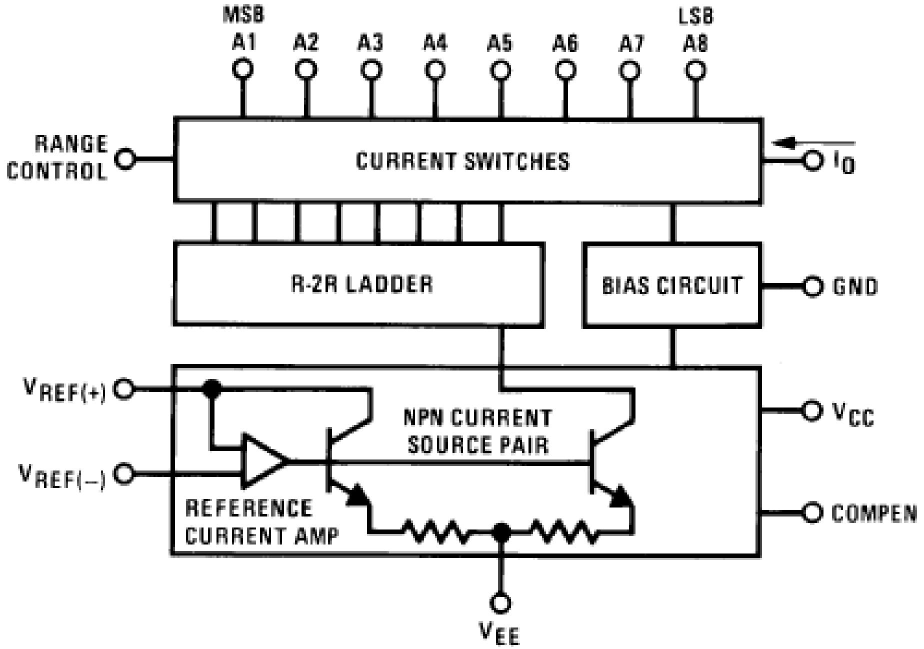


Figure 14.1: Functional block diagram of DAC 0808 chip

VIII Practical Circuit diagram:

a) Sample Circuit diagram

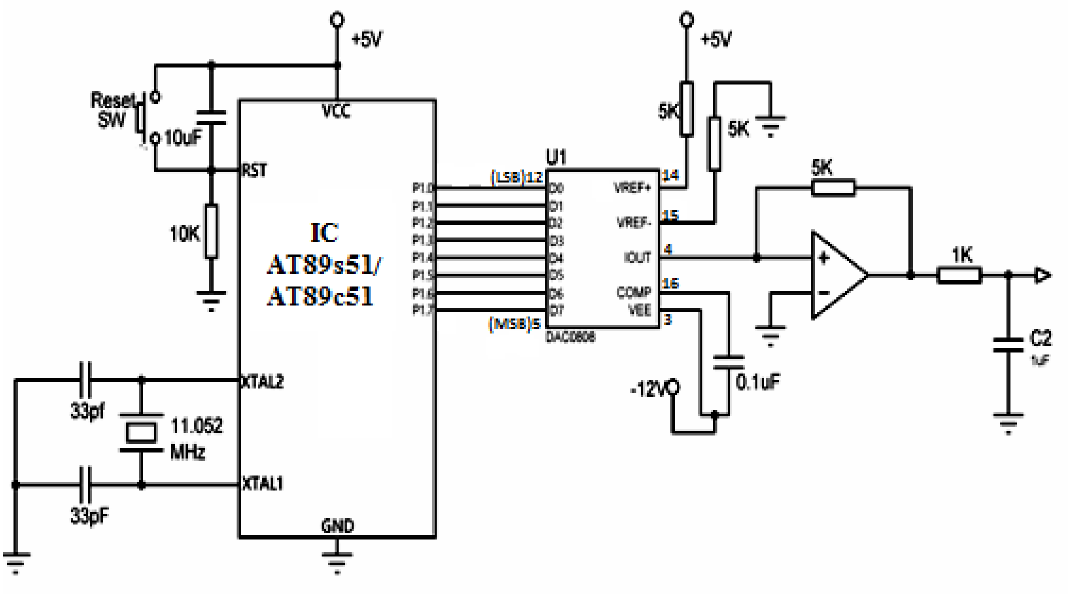


Figure 14.2: 89C51 connection to DAC0808

b) Practical setup

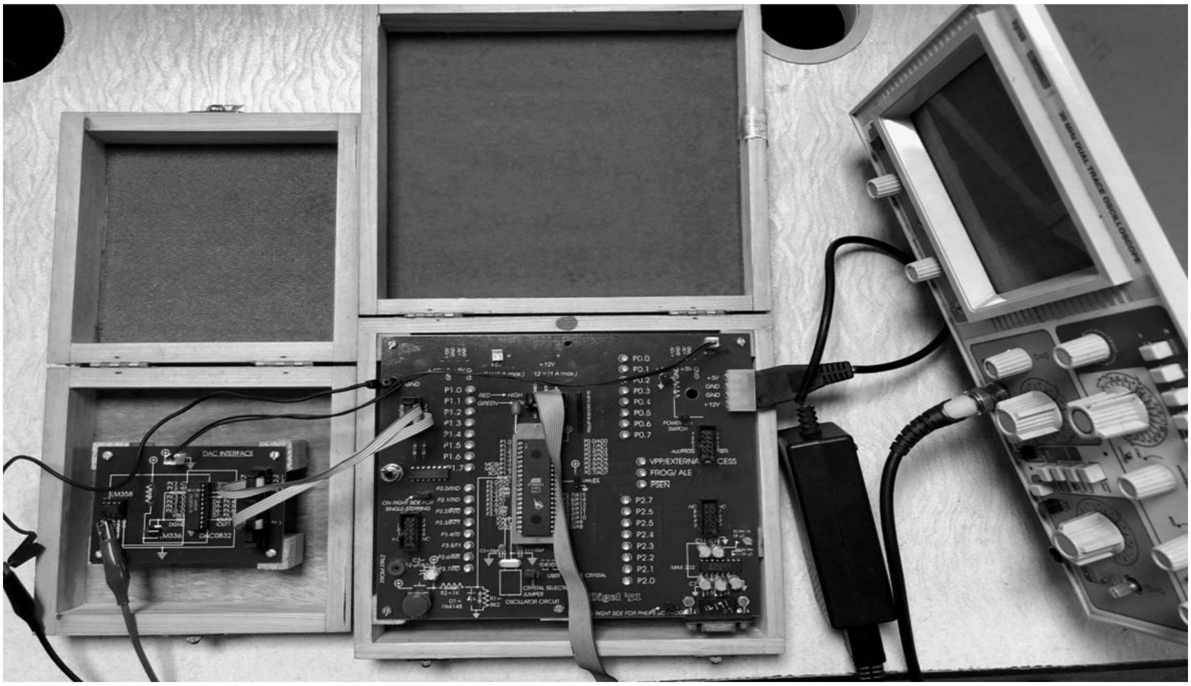
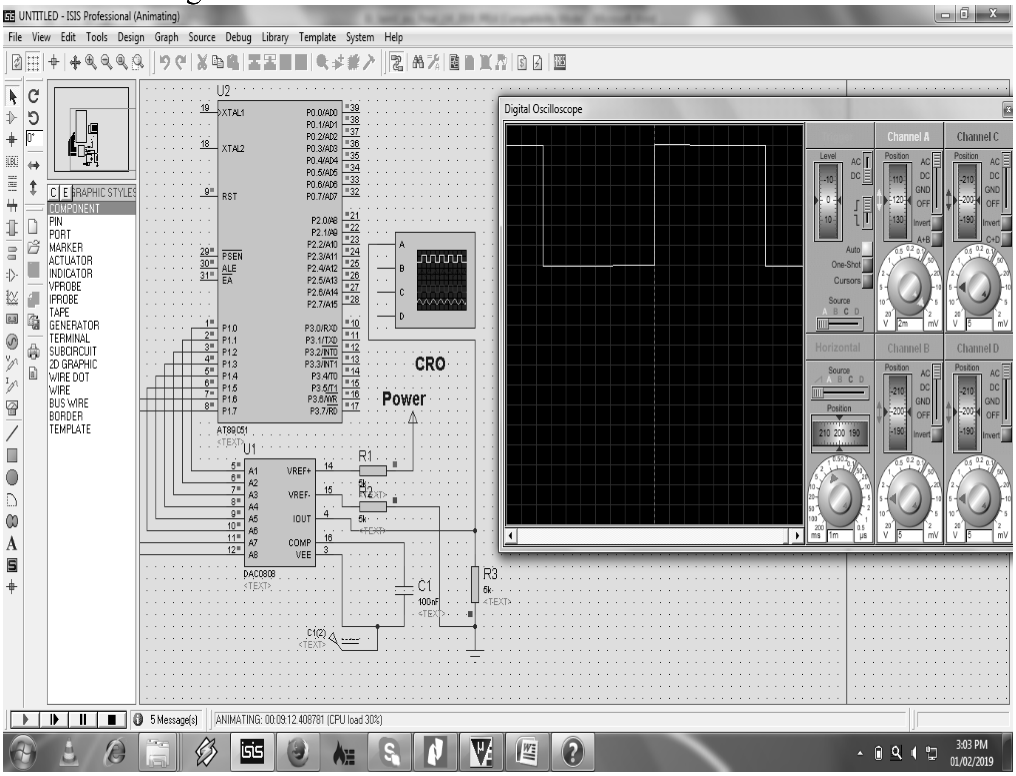


Figure 14.3: Practical Setup

c) Simulation diagram



ANIMATING: 00:09:12.408781 (CPU load 30%)

Figure 14.4: Simulation diagram for square wave.

a) Actual circuit used in Laboratory

b) Actual Experimental set up used in laboratory

IX Resources Required

Sr. No.	Instrument /Components	Specification	Quantity
1.	Microcontroller kit (89C51)	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.	DAC (0808) trainer board	Suitable to interface 8051 board.	1 No
4.	CRO	Bandwidth AC 10Hz ~ 20MHz (-3dB). DC ~ 20MHz (-3dB), X10 Probe.	1 No
5.	Digital Multimeter	3 1/2 digit display, 9999 counts digital multimeter measures: Vac, Vdc (1000V max) , Adc, Aac (10 amp max) , Resistance (0 - 100 MΩ) , Capacitance and Temperature measurement	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 embedded 'C' program using 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 DAC to microcontroller as per circuit diagram shown in Figure.14.2
9. Observe analog signal on CRO and trace it in observation table.

SAMPLE PROGRAM 1: : Develop and execute C language program to generate square waveform using given 8 bit DAC.

Step 1: Algorithm

1. Initialize port P1 as output port.
2. Send maximum value at port P1.
3. Call delay for calculated time interval to generate square wave i.e time delay.
4. Send min value to port P1.
5. Call same delay to obtain 50% duty cycle.
6. Repeat from step2.

Step 2: Flowchart

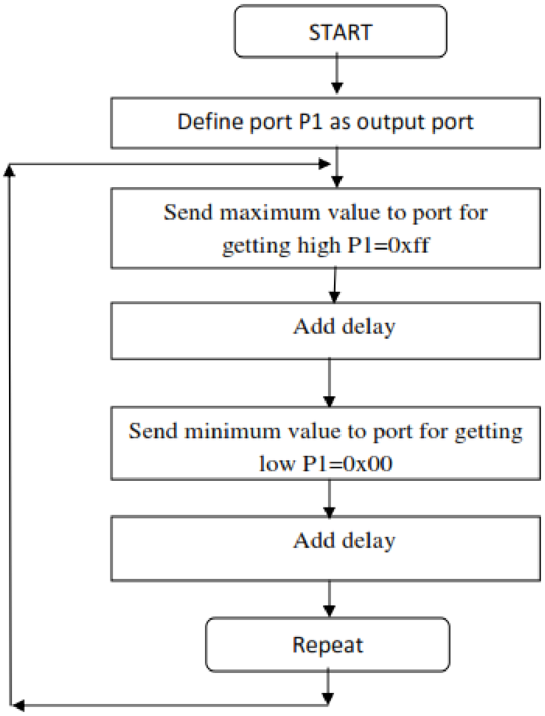

START
Define port P1 as output port
Send maximum value to port for getting high P1=0xff
Add delay
Send minimum value to port for getting low P1=0x00
Add delay
Repeat

Figure 14.5: Flowchart for square wave using DAC

Step 3: ‘C’ Language Program	**Comments**

```
#include<reg51.h>                  /* Special function register declarations */
                                   /* For the intended 8051 derivative */
void delay_ms(unsigned int);       //Function prototype declaration

void main (void)                   //Main program
 {
 P1=0x00;                          //Define port as output
 while(1)                          //Infinite loop
  {
  P1=0xff;                         //Send max value
  delay_ms(1);
  P1=0x00;                         //Send min value
  delay_ms(1);
  }
}

void delay_ms(unsigned int k)      //Delay function
{
 unsigned int i,j;

  for(i=0;i<k;i++)
  {
   for(j=0;j<1275;j++);
  }
}
```

SAMPLE PROGRAM 2 : Develop and execute C language program to generate saw tooth waveform using given 8 bit DAC.

Step 1: Algorithm

1. Initialize port P1 as output port.
2. Send maximum value at port P1.
3. Call delay for calculated time interval to generate square wave i.e time delay.
4. Send min value to port P1.
5. Call same delay to obtain 50% duty cycle.
6. Repeat from step2.

Step 2: Flowchart

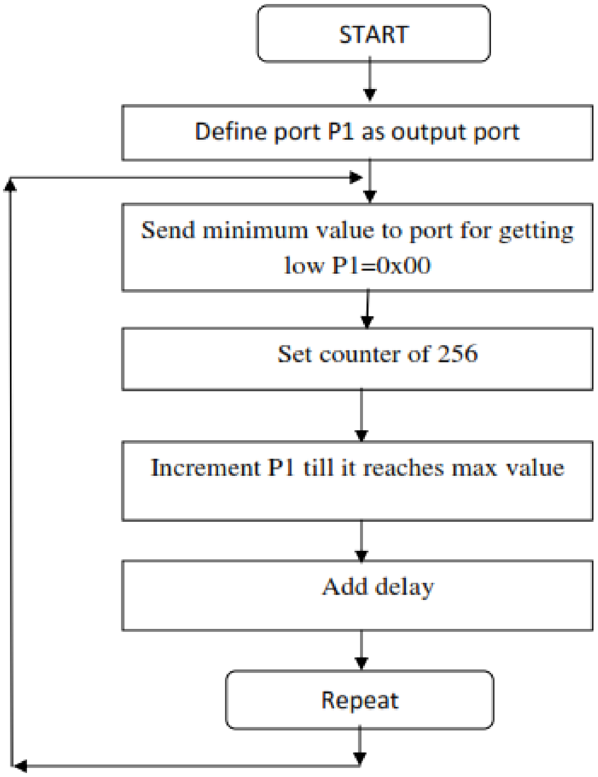

START
Define port P1 as output port
Send minimum value to port for getting low P1=0x00
Set counter of 256
Increment P1 till it reaches max value
Add delay
Repeat

Figure 14.6: Flowchart for saw tooth using DAC

Step 3: 'C' Language Program	**Comments**
#include<reg51.h>	/* Special function register declarations */
	/* For the intended 8051 derivative */
void delay_ms(int);	//Function prototype declaration
void main (void)	
{	
unsigned char k;	
P1=0x00;	//Define port as output
while(1)	//Infinite loop
{	
P1=0x00;	
for(k=0;k<255;k++)	
{	
P1++;	//Increment port value
delay_ms(1);	
}	
}	
}	
void delay_ms(unsigned int k)	//Delay function
{	
unsigned int i,j;	

```
   for(i=0;i<k;i++)
   {
    for(j=0;j<1275;j++);
   }
 }
```

Problem statement for student: Develop simple function generator using DAC 0808.

1. Generate square waveform when switch1 is pressed.
2. Generate saw tooth waveform when switch2 is pressed.

Step 1: Algorithm

Step 2: Flowchart

Step 3- ‘C’ Language Program	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 (use blank sheet provided if space not sufficient)

Draw waveform observed on CRO for problem statement given to student.

1) Switch1 pressed	2) Switch2 pressed

XVI Result (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. Calculate analog value for digital data (11000011) for reference current 2mA.
2. To generate a sine wave using DAC 0808 find decimal values representing magnitude of the sine of angles between 0 and 360 degrees.
   i. Refer Vout = 5V+ (5xsinθ).
3. Modify sample program no 2 to generate reverse sawtooth waveform.

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XX 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

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)**

Names of student Team Member

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

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

Practical No. 15: Interface 89C51 /AVR microcontroller and write C program to rotate stepper motor with different speeds in clockwise and counter clockwise direction.

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, making robots and process control applications. This practical will help the students to develop skills to interface stepper motor to 89c51 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.
- **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 Embedded Systems’.**

- Develop ‘C’ code using IDE tools.
- Use Input/output port pins of microcontroller.
- Interface stepper motor to microcontroller.

IV Relevant Course Outcome

- Develop basic applications using embedded systems.

V Practical Outcome

- Interface 89C51 /AVR microcontroller and write C program to rotate stepper motor with different speeds in clockwise and counter clockwise direction.

VI Relevant Affective domain related Outcome(s)

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

VII Minimum Theoretical Background

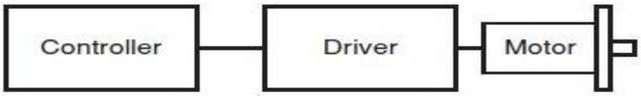


Figure 15.1: Basic Stepper motor system

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 15.1: Stepper Motor Step Angles

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

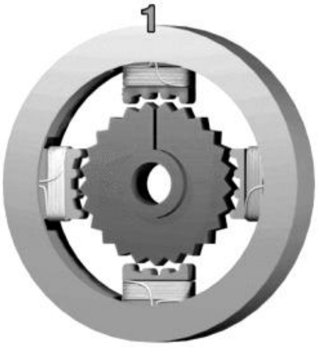
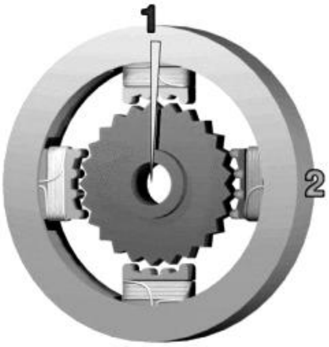
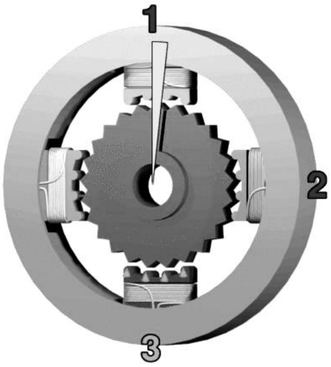
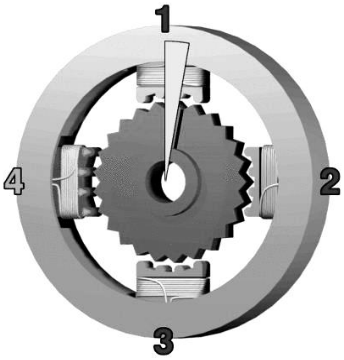
Stepping Modes:

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 15.2: 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

Table 15.3: Stepper motor working

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

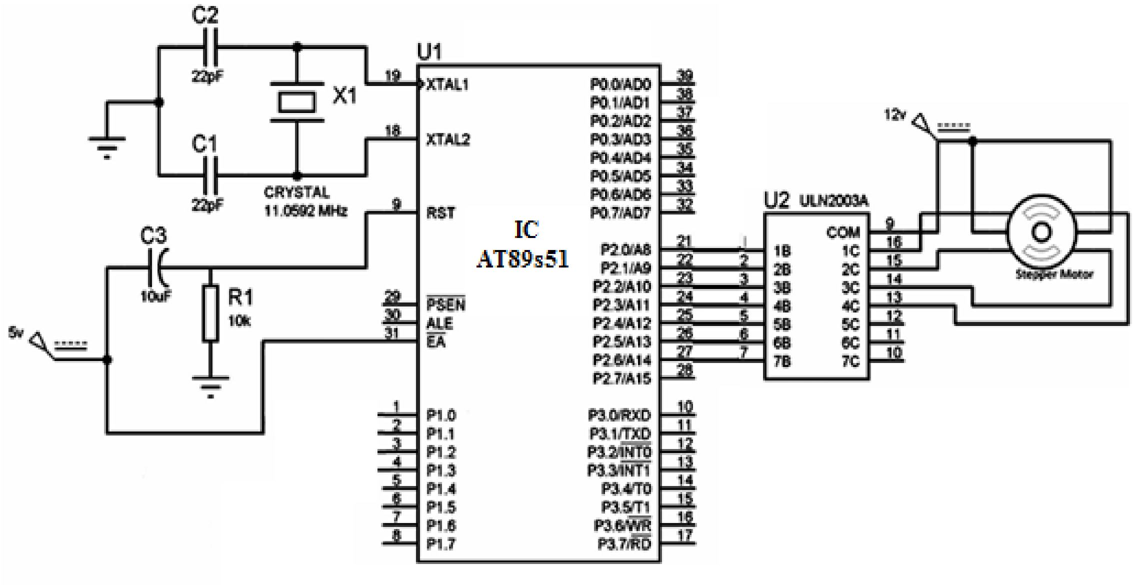


Figure 15.2: 89C51 connection to stepper motor

b) Practical setup

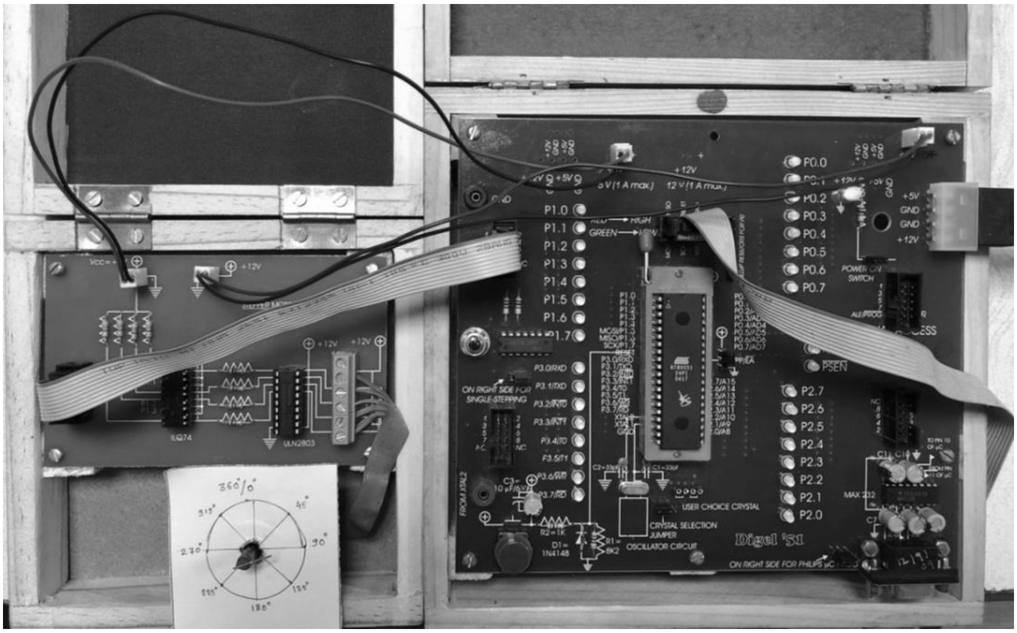


Figure 15.3: Practical Setup

c) Simulation diagram

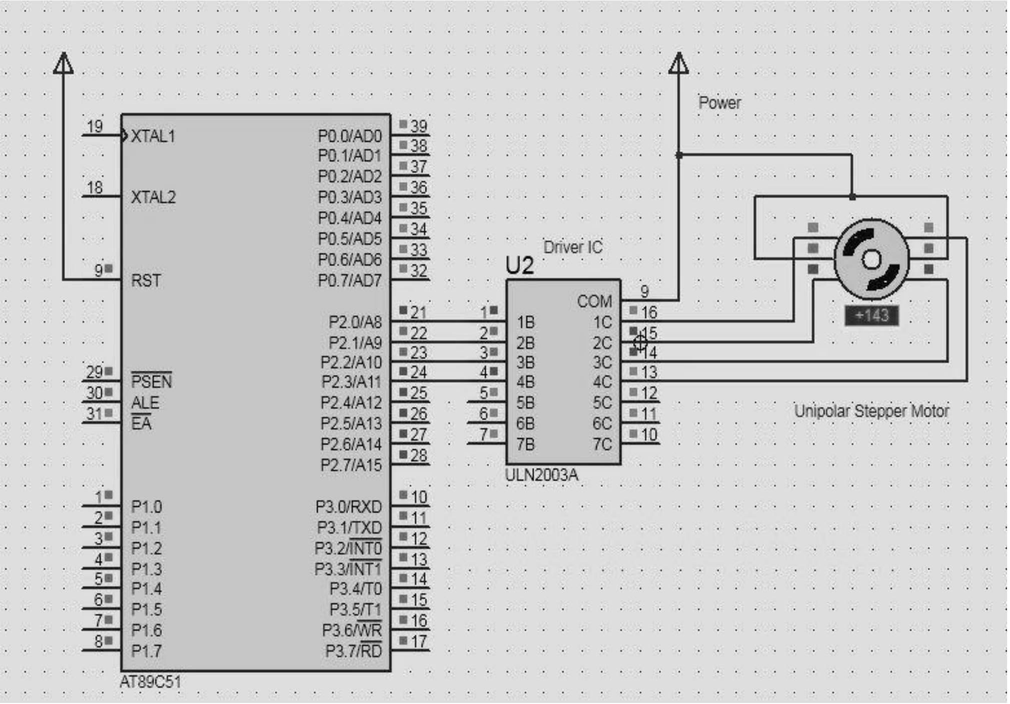


Figure 15.4: Simulation diagram

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 (89C51)	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.	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 between stepper motor and microcontroller.

XI Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop embedded ‘C’ program using 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 stepper motor to microcontroller as per circuit diagram shown in Figure.15.2

9. Observe rotation of stepper motor.

   SAMPLE PROGRAM 1: **:** Develop and execute C language program to rotate stepper motor in counter clockwise direction.

Step 1: Algorithm

1. Initialize port P2 as output.
2. Send hex value 0x33 to port 2 and add delay
3. Send hex value 0x66 to port 2 and add delay.
4. Send hex value 0xCC to port 2 and add delay.
5. Send hex value 0x99 to port 2 and add delay.
6. Repeat from step2.

Step 2: Flowchart

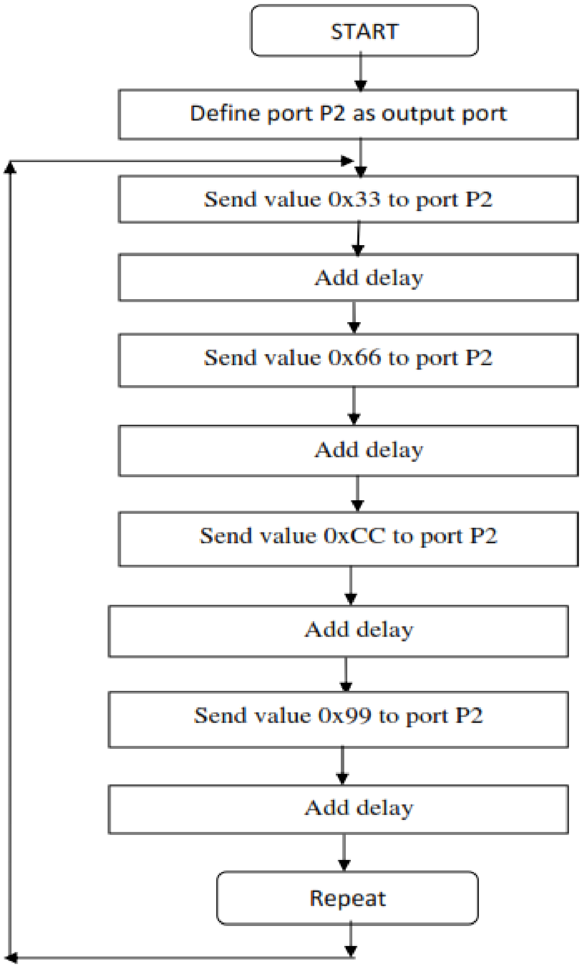

START
Define port P2 as output port
Send value 0x33 to port P2
Add delay
Send value 0x66 to port P2
Add delay
Send value 0xCC to port P2
Add delay
Send value 0x99 to port P2
Add delay
Repeat

Figure 15.5: Flowchart for stepper motor to rotate counter clockwise

Step 3: 'C' Language Program	**Comments**
#include<reg51.h>	/* Special function register declarations */ /* For the intended 8051 derivative */
void delay_ms(unsigned int);	//Function prototype declaration
void main (void) {	
P2=0x00;	//Make port as output
while(1)	//Infinite loop

{
P2=0x33;
delay_ms(100);
P2=0x66;
delay_ms(100);
P2=0xCC;
delay_ms(100);
P2=0x99;
delay_ms(100);
}
}	

void delay_ms(unsigned int k)	//Delay function

{
unsigned int i,j;
for(i=0;i<k;i++)
{
for(j=0;j<1275;j++);
}
}	

Problem statement for student: A switch is connected to pin P2.7. Write a C program to monitor the status of SW and perform the following:

1. If SW=0, the stepper motor moves clockwise.
2. If SW=1, the stepper motor moves counter clockwise.

Step 1: Algorithm

Step 2: Flowchart

Step 3- 'C' Language Program	**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 (use blank sheet provided if space not sufficient)

Write 4 step codes for clockwise and counter clockwise direction.

Clockwise:	Counter clockwise:

XVI Result (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 the 4-step sequence of a stepper motor if code start with 0110.
2. A stepper motor with a step angle of 7.5 degrees. Write requirement of steps per revolution.
3. Mention current sinking and sourcing capacity of microcontroller port pin.
4. If a motor takes 180 steps per revolution then give the step angle for this motor.
5. List the specifications of the stepper motor used in the above practical.

[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. 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%**

Names of student Team Member

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

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

Practical No. 16: Interface 89C51 and write C program to observe the real time status of control signals for the given waveform generated using DAC (Use IDE tool MicroProC/Keil).

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 89C51 and generate given analog waveform and to observe real time status of control signals.

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 Embedded Systems'.**

- Develop 'C' code using IDE tools.
- Use Input/output port pins of microcontroller.
- Interface 8 bit ADC to microcontroller
- Interface 8 bit DAC to microcontroller.

IV Relevant Course Outcome

- Interpret features of Real Time Operating System.

V Practical Outcome

- Interface 89C51 and write C program to observe the real time status of control signals for the given waveform generated using DAC (Use IDE tool MicroProC/Keil).

VI Relevant Affective domain related Outcome(s)

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

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.

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.

If an analog signal like a continuous triangular wave is applied to ADC, its equivalent digital output is obtained at which if further applied to DAC through microcontroller its output would be similar to the original analog applied at the input of ADC. This would enable to observe the real time status of the signal applied

VIII Practical Circuit diagram:

a) Sample Circuit diagram

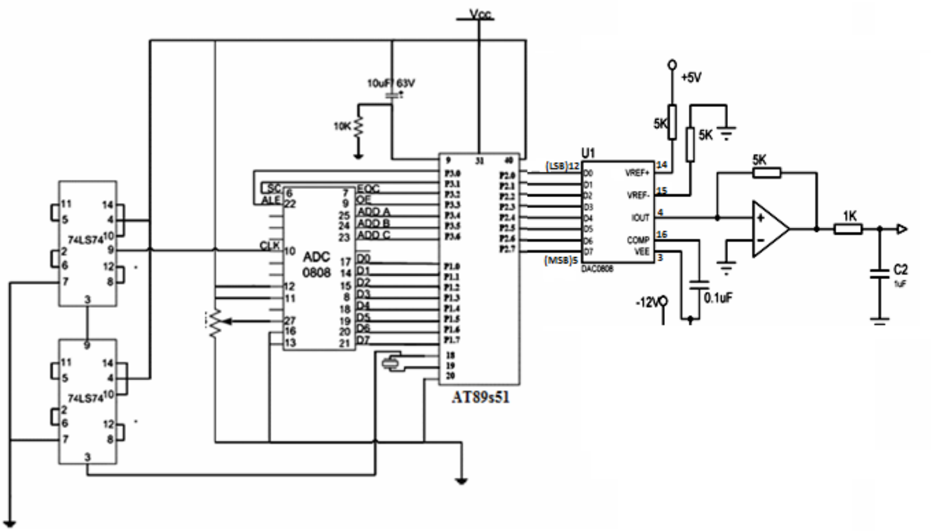


Figure 16.1: 89C51 connection to ADC and DAC 0808

b) Practical setup

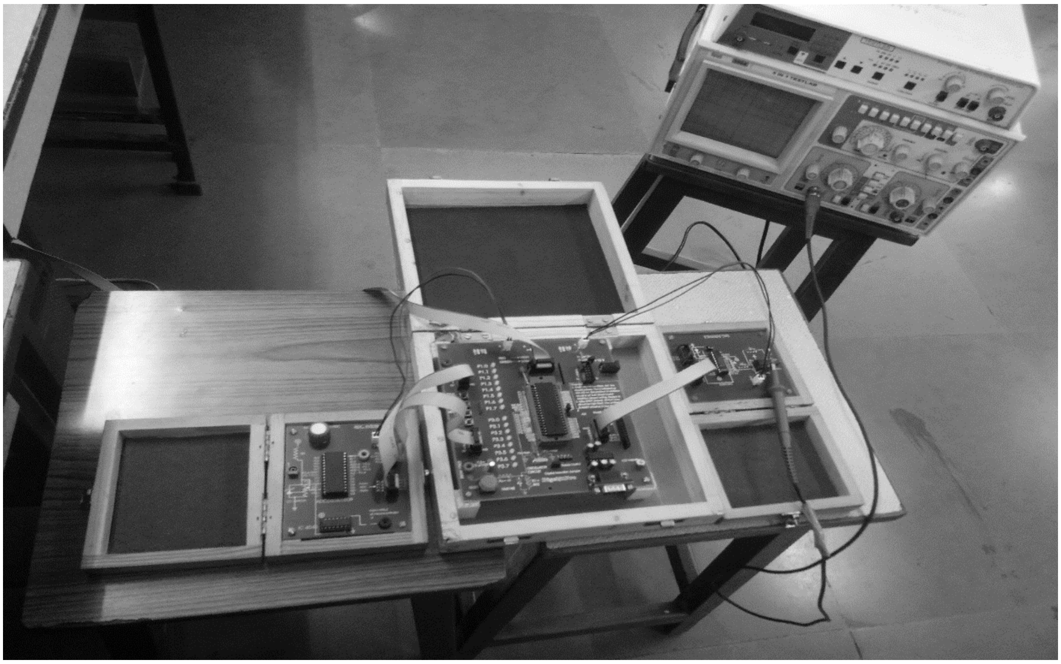


Figure 16.2: Practical Setup

c) Simulation diagram

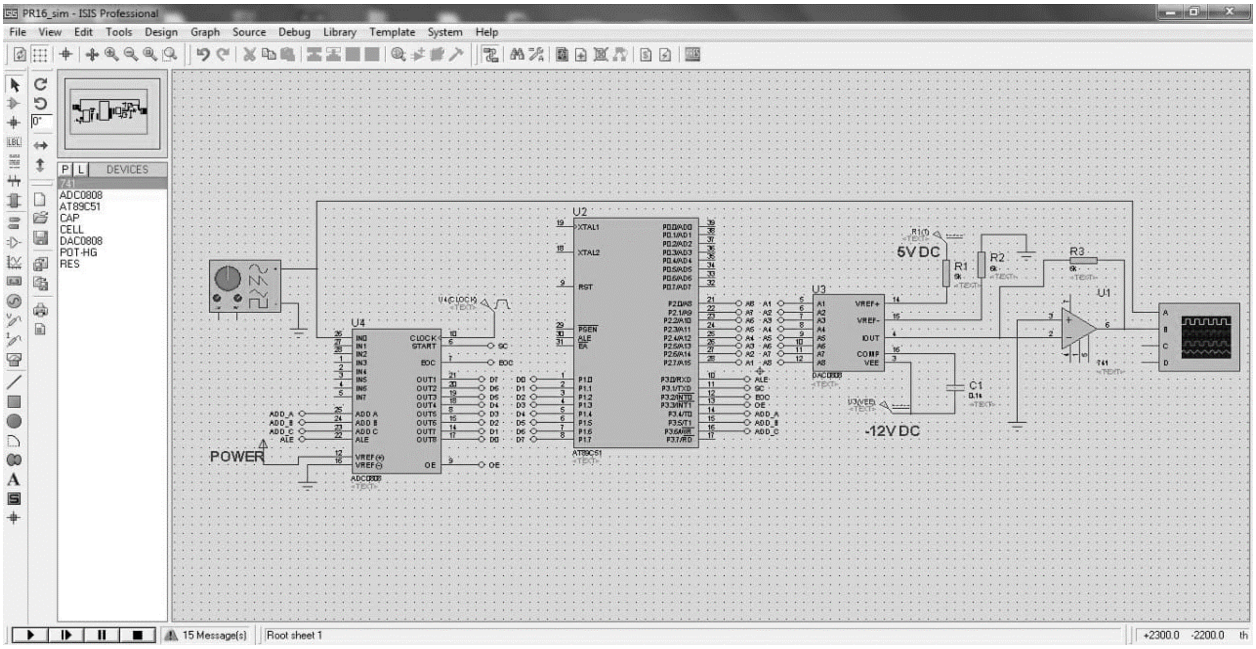


Figure 16.3(a): Simulation diagram before execution

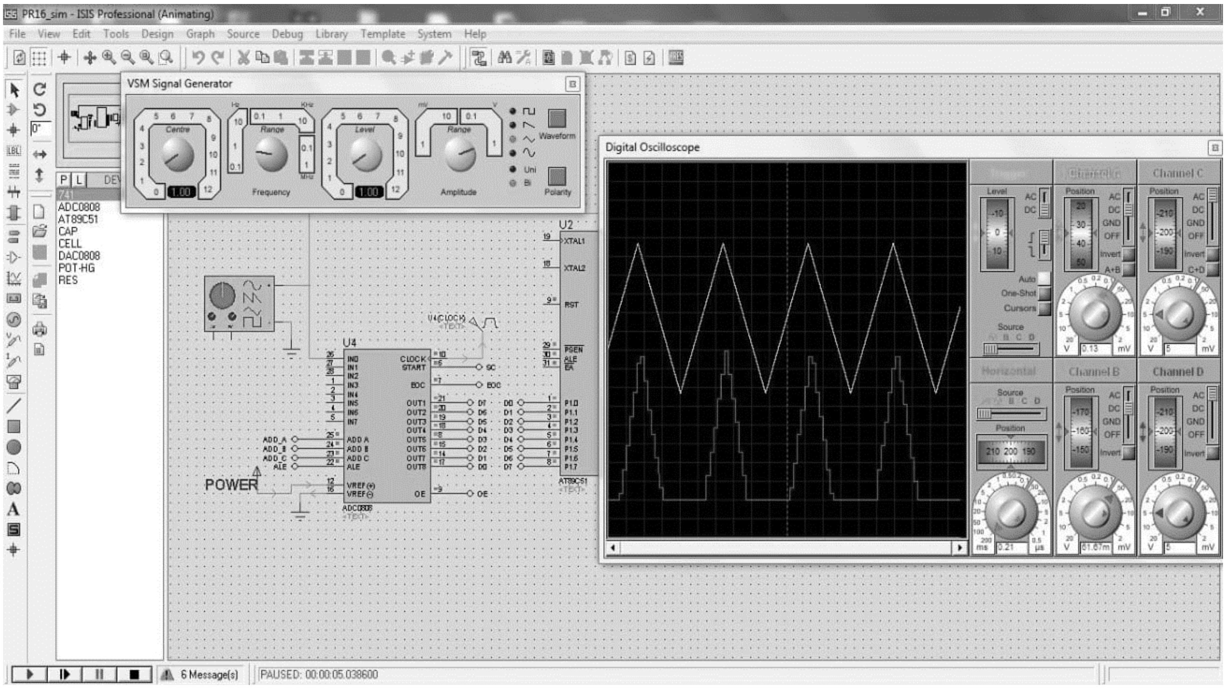


Figure 16.3(b): Simulation diagram after execution

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 (89C51)	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.	ADC and DAC (0808) trainer board	Suitable to interface 8051 board.	1 No
4.	CRO	Bandwidth AC 10Hz ~ 20MHz (-3dB). DC ~ 20MHz (-3dB), X10 Probe.	1 No
5.	Digital Multimeter	3 1/2 digit display, 9999 counts digital multimeter measures: Vac, Vdc (1000V max) , Adc, Aac (10 amp max) , Resistance (0 - 100 MΩ) , Capacitance and Temperature measurement	1 No
6.	Function Generator	0.2Hz to 2 MHz	1 No.

X Precautions to be followed

1. Operate ADC and DAC chip as per specifications given in the datasheet otherwise damage may occur to the device.
2. Use clock frequency not more than 100 µsec [10KHz]

XI Procedure

1. Write algorithm for given problem.
2. Draw flowchart for the same.
3. Develop embedded ‘C’ program using 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 ADC and DAC to microcontroller as per circuit diagram shown in Figure.16.1.
9. Connect function generator to ADC as input and provide 1Hz frequency.
10. Observe output signal on CRO and trace it in observation table.

SAMPLE PROGRAM 1: Develop and execute C language program to read data from ADC and verify the output on DAC.

Step 1: Algorithm

1. Initialize port P1 as input port.
2. Initialize port P2 as output port.
3. Select an analog channel by providing bits to A, B, and C addresses according to the analog signal selection .
4. Activate the ALE (address latch enable) pin.
5. Activate SC (start conversion) to initiate conversion.
6. 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.
7. Activate OE (output enable) to read data out of the ADC chip.
8. Read port P1 and send it to port P2.
9. Repeat from step 5.

Step 2: Flowchart

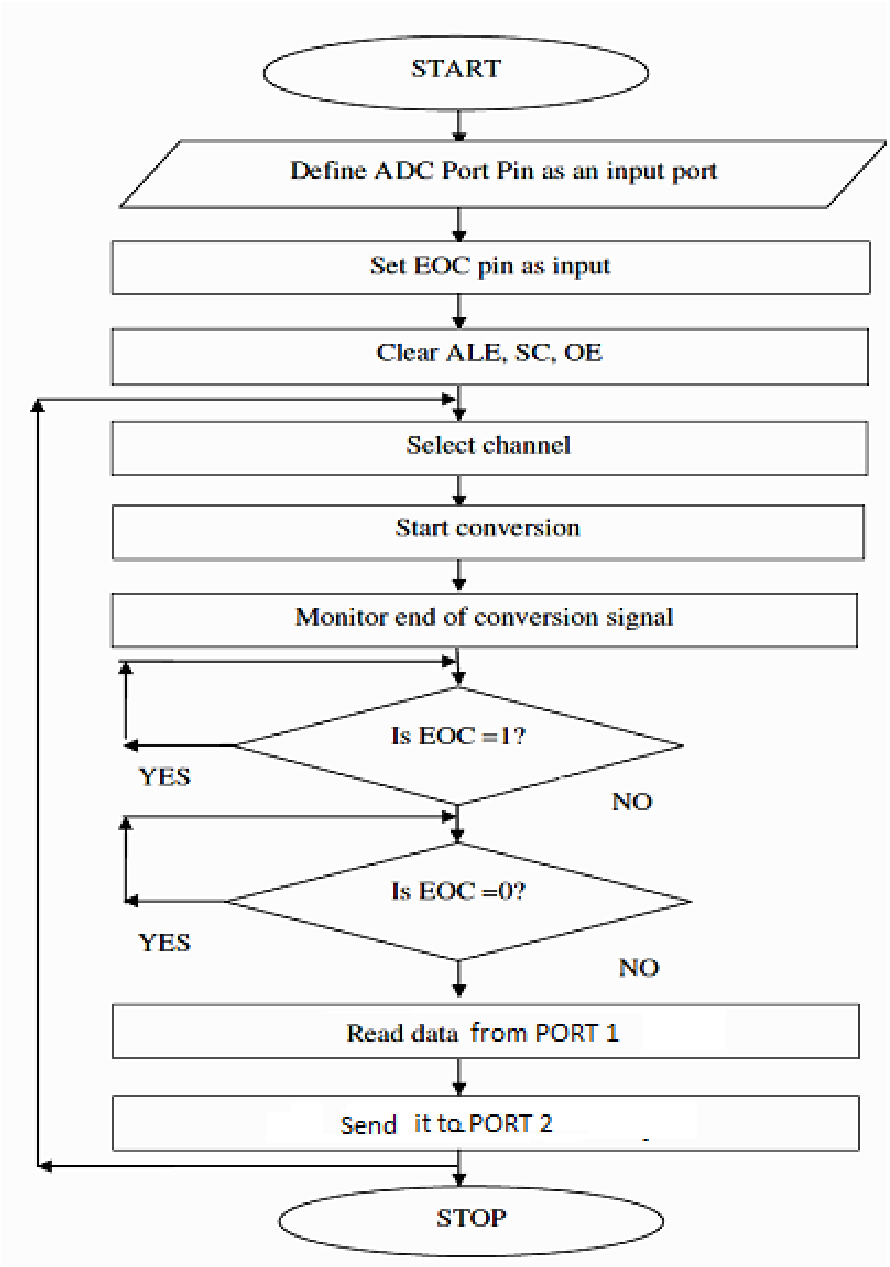


Figure 16.4 Flowchart for real time status of control signal

Step 3: 'C' Language Program	**Comments**
#include <reg51.h>	/* Special function register declarations */
sfr ADC_DATA = 0x90;	/* For the intended 8051 derivative */
sfr DAC_DATA = 0xA0;	
void delay_ms(unsigned int);	
sbit ALE=P3^0;	//Define ADC pins

```
sbit SC=P3^1;
sbit EOC=P3^2;
sbit OE=P3^3;

sbit ADDR_A=P3^4;
sbit ADDR_B =P3^5;
sbit ADDR_C =P3^6;

void main(void)
{
 unsigned char value;
 ADC_DATA=0xff;                 //Make P1 as input
 DAC_DATA  = 0x00;              // Make P2 as output
EOC=1;                          //Make EOC as input
ALE=0;                          //Clear ALE
OE=0;                           //Clear OE
SC=0;                           //Clear SC

while(1)
 {
 ADDR_A=0;                      //Select CH 0
 ADDR_B=0;
 ADDR_C=0;
 delay_ms(1);
 ALE=1;
 delay_ms(1);
 SC=1;                          //Start conversion
 delay_ms(1);
 ALE=0;
 SC=0;
 while(EOC==1);
 while(EOC==0);                 //Wait for data conversion
 OE=1;                          //Enable read
 delay_ms(1);
 value=ADC_DATA;                //Read data from P1
 OE=0;
 DAC_DATA =value;
 }                              //Send it to P2
}
void delay_ms(unsigned int k)
 {                              //Delay function
  unsigned int i, j;
  for(i=0;i<k;i++)
   {
    for(j=0;j<1275;j++);
   }
 }
```

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 (use blank sheet provided if space not sufficient)

Draw waveform observed on CRO for Sample program.

XVI Result (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. Verify the values of DAC output for following given input at ADC
   i. 1V ii. 5 V
2. State the necessity of OP AMP at the output of DAC.

[Space for Answers]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XX 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 https://en.wikipedia.org/wiki/Analog-to-digital converter.
4 https://pdf1.alldatasheet.com/datasheet-pdf/view/155397/TI/ADC0808.html

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%**

Names of student Team Member

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	22101
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	22213
12	Elements of Electrical Engineering	22215
13	Basic Electronics	22216
14	‘C’ programming Language	22218
15	Basic Electronics	22225
16	Programming in “C”	22226
17	Fundamentals 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 Chemicals	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 Measurements	22420
12	Digital Electronics And Microcontroller Applications	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

Fifth Semester:

1	Design of Steel and RCC Structures	22502
2	Public Health Engineering	22504
3	Heat Transfer Operation	22510
4	Environmental Technology	22511
5	Operating Systems	22516
6	Advanced Java Programming	22517
7	Software Testing	22518
8	Control Systems and PLC’s	22531
9	Embedded Systems	22532
10	Mobile and Wireless Communication	22533
11	Industrial Machines	22523
12	Switchgear and Protection	22524
13	Energy Conservation and Audit	22525
14	Power Engineering and Refrigeration	22562
15	Solid Modeling and Additive Manufacturing	22053
16	Guidelines & Assessment Manual for Micro Projects & Industrial Training	22057

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