	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

SUBJECT :-HARDWARE PROGRAMME & INTERFACEING LAB

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

LAB DO'S AND DONT'S

DO's

1. Know the location of all safety and emergency equipment used in the lab.

2. Know fire drill procedures and lcoations of all exits.

3. Know the location of the closest telephone.

4. Familiarize yourself with all lab procedures before doing the lab exercise.

5. Report **ALL** accidents, hazards or chemical spills to the instructor (no matter how small).

6. Keep you work area clean and clutter free.

7. Tie back all long hair and remove dangling jewelry during lab.

8. Always be sure that electrical equipment is turned in the "off" position before plugging it into a socket.

9. Handle all animals with care
.

DON'T'S

1. NEVER experiment on your own.

2. Do not eat or drink in the lab room at any time (other than permitted by instructor).

3. Do not chew gum or eat candy during lab exercises.

4. NEVER add water to an acid.

5. Do not wear contacts in the lab without proper eye protection.

6. NEVER smell, taste or touch chemicals.

7. NEVER work in the lab alone.

8. NEVER use electrical equipment around water.

9. NEVER mix chemicals before asking the instructor.

10. NEVER return unused chemicals to the original container.

11. Absolutely NO HORSEPLAY is allowed in the lab area!

10. Use extreme care when handling sharp objects.

11. Dispose of all chemicals, broken glass and other lab materials into the proper containers as directed by the instructor.

12. When heating liquids in a test tube, **always** point the test tube away from other students.

12. NEVER leave the lab are without washing your hands.

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

Internal Marks: 30 L T P
External Marks: 20 0 0 2

Total Marks: 50

LIST OF EXPERIMENTS

Note: Any Eight Experiments each from Part A and Part-B

Part-A: List of Experiments using 8085/8086:

1. Study of 8085 and 8086 Microprocessor Kits.
2. Write a program to add two 8-bit number using 8085.
3. Write a program to add two 16-bit number using 8085.
4. Write a program to subtract two 8-bit number using 8085.
5. Write a program to subtract two 16-bit number using 8085.
6. Write a program to multiply two 8 bit numbers by repetitive addition methodusing 8085.
7. Write a program to sort series using bubble sort algorithm using 8085.
8. Write a program to copy 12 bytes of data from source to destination using 8086.
9. Write a program to find maximum and minimum from series using 8086.
10. Write a program to control the operation of stepper motor using 8085/8086 microprocessors and 8255 PPI.
11. Write a program to control speed of DC motor using 8085/8086 microprocessors and 8255 PPI.

Part-B: List of Experiments using 8051:

1. Study of 8051/8031 Micro controller kits.
2. Write a program to add two numbers lying at two memory locations and display
the result.
3. Write a program for multiplication of two numbers lying at memory location and display the result.
4. Write a Program to arrange 10 numbers stored in memory location in Ascending and Descending order.
5. Write a program to show the use of INT0 and INT1.
6. Write a program of Flashing LED connected to port 1 of the Micro Controller

7. Write a program to generate a Ramp waveform using DAC with micro controller.
8. Write a program to interface the ADC.
9. Write a program to control a stepper motor in direction, speed and number of steps.
10. Write a program to control the speed of DC motor.
11. Interfacing of high power devices to Micro-controller port-lines, LED, relays and
LCD display.

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

AIM: To study 8086 microprocessor kit

.**THEORY:**CPU: DYNA-86 is based on intel 8086 high performance CPU operating at 8 MHz speed.

Memory: Monitor Firmware in two 27256 EPROMz is placed in the highest 64 KB bank (F0000H to FFFFFH). 64KB static RAM with powerful battery backup is provided in the address range 00000 to 0FFFFH.

Hexpad/Display interface

8279 keyboard display controller is used for Hexpad keys & displays (8nos. of 7 segment displays)

Serial Interface: Serial interface is available through a RS-232 compatible port. 8251 USART along with 1488, 1489 driver chips provides necessary signals for this interface. The signals are brought out on the 9 pin D-type male connector (J5). Baud rates from 300 to 9600 can be selected through software.

Timer: Three channels of 16 bit Timer/Counter are provided using 8254. CHANNEL 0 is used for Baud rate generation. CH1 and CH2 signals are brought out on a 7 pin Relimate connector and can be used by the user.

Interrupt Controller:
The 8259 interrupt controller provides 8 prioritized interrupt levels. IRQ5 to IRQ7 are brought out on 50 pin FRC connector and can be used by the user. IRQ3 is connected to "INT" key of Hex keypad. 8259 is programmed for edge trigger. Except IRQ3 all other interrupts are masked.

Parallel I/O Interface:
Two 8255's are present onboard, out of which 1 is used for DYNA-PIO cards and 1 for printer interface. All the 48 lines of 8255's are available to the user and are brought out on the two numbers of 26 pin FRC male

connectors. The PIO cards from Dynalog supported on DYNA-86 are listed in Appendix B, and can be interfaced on connector J2. The connector J1 is used for printer interface in serial mode.

System Software

The DYNA-86 Microprocessor kit has vast software features. It supports two different modes of operation:

1. HEX KEYPAD mode
2. SERIAL mode

POWER SUPPLY

The kit is normally used with the Dynalog's SMPS 04 Model Power Supply. The 6 pin female connector can be plugged in 6 pin Male Connector soldered on board

The power requirement of DYNA-86 board is:

+5 V	3 Amps
+12V	250 mA
-12V	250mA

CONNECTORS ON BOARD

The pin details of all the connectors are given in Appendix A.

SERIAL CONNECTOR

All the signals for the RS 232C compatible serial interface are brought out on the 9-pin D type Male (DTM) connector onboard. The serial cables can be directly connected to this connector.

Relimate Connector for TIMER

A 7-pin Relimate connector is provided, which has timer interface lines terminated on it. It can be used for user applications.

FRC for 8255 I/O interface

Two 26 pin FRC male connectors are provided onboard for 8255's I/O interface. The 3 ports, 8 bit each, 24 lines of each, 8255 are provided on this connector. Connector J2 is used for interfacing DYNA-PIO cards whereas J1 is used to connect printer in serial mode.

FRC for Buffered Bus

A 50 pin FRC male connector provided is for Bus expansion purpose. All the address, Data and control lines along with the DRQ & interrupts are terminated on this connector (J7). The same connector is used to interface DYNA-86 with DYNA-series study cards.

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

EXPERIMENT NO.: -2

AIM:- Write a program to add two 8 bit numbers using 8085.

APPRATUS:- 8085 mp kit

Procedure: -

Prepare the Flow Chart for Program

Step 1 : To enter the program
1. Switch on the 8085 mp kit.
2. Press 'Reset'.
3. Press 'Exmem'.
4. Enter the memory address where program is to be stored i.e. 2000.
5. Press 'Next'.
6. Now enter the Opcode for Ist instruction i.e. 21.
7. Now Press 'Next' and enter the next Opcode.
8. Now subsequently enter the all Opcode.
9. Now press 'Fill'.

Step 2 : To enter the data
1. Press 'Reset'.
2. Press 'Exmem'.
3. Enter the memory address where first data is to be stored i.e. 2501.
4. Press 'Next'.
5. Enter the first data.
6. Press 'Next'.
7. Enter the second data.
8. Now press 'Fill'.

Step 3 : Execution
1. Press 'Reset'.
2. Press 'Go'.

3. Enter the starting address of program.
4. Press 'Next'
5. If display 'E' it mean execution is done.
6. If display 'Err' it mean there is an error in the program.

Step 4 : Verification

1. Press 'Reset'.
2. Press 'Exmem'.
3. Enter the memory address where result is stored.

ADDRESS	OPCODE	LABEL	MNEMONIC	OPERAND	COMMENTS
2000	21,01,25		LXI	H, 2501 H	Get address of 1st no. in H-L pair
2003	7E		MOV	A, M	1st no. in accumulator
2004	23		INX	H	Increment content of H-L pair
2005	86		ADD	M	1st no. + 2nd no., result in accumulator
2006	23		INX	H	Increment content of H-L pair
2007	77		MOV	M, A	Store result in 2503 H
2008	76		HLT		Halt

4. Press 'Next'
5. See the contents of that address that is result.

Program:

Result: - Check the result on the memory location defined in the program

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

EXPERIMENT NO.: -3

AIM: - Write a program to add two 16-bit number using 8085.

Apparatus: - 8085 microprocessor kit.

Program:

ADDRESS	OPCODE	LABEL	MNEMONIC	OPERAND	COMMENTS
2006	2A,02,20,EB	START	LHLD	2002 H	load HL pair with l(2002);h(2003)
2009	EB		XCHG		Exchange HL pair with DE pair
200A	2A,00,20		LHLD	2000 H	load HL pair with L-(2000);H(2001)
200D	7D		MOV	A,L	A←L
200E	93		ADD	E	A←A+E
200F	6F		MOV	L,A	Restore the result and LSB in Reg L.
2010	7C		MOV	A,H	A←H
2011	9A		ADDC	D	A←A+D with carry
2012	67		MOV	H,A	Restore The Result And MSB In Reg H
2013	22		SHLD	2004	Store 16 bit result 2004←(L); 2005←(H)
2016	76		HLT		Stop

Result: -Check the result on the memory location defined in the program.

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

EXPERIMENT NO.: -4

AIM: - Write a program to subtract the two 8-bit numbers using 8085.

Apparatus: - 8085 microprocessor kit.

Procedure: -Prepare the Flow Chart for Program

PROGRAM:- Write a program for the subtract two 8-bit numbers using

ADDRESS	OPCODE	LABEL	MNEMONIC	OPERAND	COMMENTS
2000	21,01,25		LXI	H, 2501 H	Get address of 1st no. in H-L pair
2003	7E		MOV	A, M	1st no. in accumulator
2004	23		INX	H	Increment content of H-L pair
2005	96		SUB	M	1st no. - 2nd no., result in accumulator
2006	23		INX	H	Increment content of H-L pair
2007	77		MOV	M, A	Store result in 2503 H
2008	76		HLT		Halt

8085.

2501 DATA 1st no. to be subtracted
2502 DATA 2nd no. to be subtracted
2503 RESULT difference

Result: - Check the result on the memory location defined in the program.

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

EXPERIMENT NO.: -5

AIM: - Write a program to subtract two 16-bit number using 8085.

ADDRESS	OPCODE	LABEL	MNEMONIC	OPERAND	COMMENTS
2004	2A,00,20		LHLD	2000 H	Load HL pair with multiplicand
2007	5E		MOV	E,M	Reg E is loaded with multiplicand
2008	16,00		MVI	D,00	Reg D is loaded with 00
200A	23		INX	H	Increment in HL,pointer to multiplier location
200B	7E		MOV	A,M	Load Acc. With multipier
200C	0E,08		MVI	C,08	Set Reg C counter = 8 bits.
200E	21,00,00		LXI	H,0000	Intialize HL reg pair to store the result/partial product.
20011	17	NEXT	RAL		Rotate Acc through carry to check multiplier bit
20012	D2,16,20		JNC	ADDI	If multiplier bit=0 i.e if not carry skip at 2015.
2015	19		DAD	D	Add the partial product and the multiplicand HL←HL+DE
2016	0D	ADDI	DCR	C	Decrement counter to check the bits of multiplier processed
2017	CA,1E,20		JZ STORE		If all bits process If zero flag is set then skip step 201A and store the result

Apparatus: - 8085 microprocessor kit.

Result: -Check the result on the memory location defined in the program

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

EXPERIMENT NO.: -6

AIM: - Write a program to multiply two 8-bit numbers by repetitive addition method using 8085.

Apparatus: - 8085 microprocessor kit.

Procedure: -

1. Switch ON the microprocessor training kit with built in power supply.
2. When the kit is ON then check, "UP 85" display on the display board.
3. Now press the REL/EXMEM key on the key board and check that "." (Dot) is displayed.
4. Load machine code at the specific addresses in the user defined memory from 2000 to 3FFF.
5. Fill the machine code on next address location by pressing the NEXT key on keyboard and check the previous code by pressing PRE key.
6. After entering the whole program, verify it by execution.

ADDRESS	OPCODE	LABEL	MNEMONIC	OPERAND	COMMENTS
2000	21,00,25		LXI	H, 2500 H	Address for count in H-L pair
2003	4E		MOV	C, M	Count in register C
2004	3E,00		MVI	A, 00	Initial value of sum = 00
2006	23	LOOP	INX	H	Address of next data in H-L pair
2007	86		ADD	M	Previous sum + next no.
2008	0D		DCR	C	Decrement count.

2009	C2,06,20		JNZ	LOOP	If count is not zero, then jump to LOOP
200C	32,50,24		STA	2450 H	Store sum in 2450 H
200F	76		HLT		Halt

PROGRAM:-

Result: - Check the result on the memory location defined in the program.

EXPERIMENT NO.: -7

AIM: - Write a program to sort series using bubble sort algorithm using 8085.

Apparatus: - 8085 microprocessor kit

Procedure: -

1. Switch ON the microprocessor training kit with built in power supply.
2. When the kit is ON then check, "UP 85" display on the display board.
3. Now press the REL/EXMEM key on the key board and check that "." (Dot) is displayed.
4. Load machine code at the specific addresses in the user defined memory from 2000 to 3FFF.

5. Fill the machine code on next address location by pressing the NEXT key on keyboard and check the previous code by pressing PRE key.
6. After entering the whole program, verify it by execution.

PROGRAM:

ADDRESS	OPCODE	LABEL	MNEMONIC	OPERAND	COMMENTS
2000	2A,01,25		LHLD	2501 H	Get data in H-L pair.
2003	29		DAD	H	Shift it left by one bit
2004	32,03,25		STA	2503 H	Store result in 2503 and 2504 H
2007	76		HLT		Halt

Result: - Check the result on the memory location defined in the program.

EXPERIMENT NO.: -8

AIM: - Write a program to copy 12 bytes of data from source to destination using 8086.

Apparatus: - Dynalog DYNA-86 microprocessor training kit and Switch Mode Power Supply (SMPS).

Procedure: -
Switch ON the microprocessor training kit and SMPS.
When the kits is ON then check, "FRIEND" displays on the display.
Now press the SET key on the key board and check that "." (Dot) is displayed.

MEMORY ADDRESS	MACHINE CODE	LABEL	MNEMONIC	OPERAND	COMMENTS

0000	B8,00,00		MOV	AX@DATA	INTIALIZE DS REGISTER
0003	8E,D88		MOV	DS,AX	
0005	FC		CLD		DIRECTION=UP
0006	B9,00,0C		MOV	CX,0CH	SET COUNTER TO 12
0009	BE,00,00 CR		MOV	SI,OFFSET SOURCE	SOURCE
000C	BF,00,0CR		MOV	DI,OFFSET DEST	DESTINATION
000F	B9,00,04		MOV	CX,4	SET UP COUNTER
0012	F3>A4		REP	MOVSB	COMPARE SOURCE WITH DESTINATION
0014	B8,4C,00		MOV	AX,AC00H	RETURN TO DOS
0017	CD,21		INT	21H	
0019			MAIN	ENDP	
0019			DATA		

Load machine code at the specific addresses in the user defined memory from C000 to FFFF.
Fill the machine code on next address location by pressing the INR key on keyboard and check the previous code by pressing DCR key.
After entering the whole program, verify it by execution.

PROGRAM:-

Result: - Check the result on the memory location defined in the program

EXPERIMENT NO.: -9

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

AIM: - Write a program to find maximum and minimum from series using 8086.

Apparatus: - Dynalog DYNA-86 microprocessor training kit and Switch Mode Power Supply (SMPS).

Procedure: -

1. Switch ON the microprocessor training kit and SMPS.
2. When the kit is ON then check, "FRIEND" display on the display board.
3. Now press the SET key on the key board and check that "." (Dot) is displayed.
4. Load machine code at the specific addresses in the user defined memory from C000 to FFFF.
5. Fill the machine code on next address location by pressing the INR key on keyboard and check the previous code by pressing DCR key.
6. After entering the whole program, verify it by execution.

PROGRAM:-

ADDRESS	OPCODE	LABEL	MNEMONIC	OPERAND	COMMENTS
			DOSSEG		
0000			.MODDEL SMALL		
0000			.STACK 100H		
0000			.CODE		
0000			MAIN PROC		
0000	B8 0000S		MOV	AX,@DATA	INTIALISE DS REG.
0003	8E D8		MOV	DS,AX	
0005	BE 0000R		MOV	SI , OFFSET LIST	INTIALISE SI REG.
0008	B0 00		MOV	A1,00 H	LARGEST POSITIVE

					NUMBER
000A	B9 000A		MOV	CX,0A H	NUMBER OF ELEMENTS
000D		BACK			
000D	3A 04		CMP A1,[SI]		IS NEXT ELEMENT > MAXIMUM
000F	73 03		JNC	AHEAD	
0011	8A 04		MOV	A1,[SI]	YES,REPLACE MAXIMUM

ADDRESS	OPCODE	LABEL	MNEMONIC	OPERAND	COMMENTS
0013	46		INC	SI	
0014		AHEAD			
0014	E2 F7			LOOP BACK	
0016	A2 000AR		MOV	RESULT,A1	
0019	B8 4C 00		MOV	AX,4C 00 H	RETURN TO DOS
001C	CD 21		INT	21 H	
001E				MAIN END P	
001E				.DATA	
0000	50 51 4E 41 2B 17 41+			LIST DB 80 ,81,78,65,43,23,65,12,10,11.	
	0C 0A 0B				
000A	??		RESULT	DB ?	
				END MAIN	

Result: - Check the result on the memory location defined in the program.

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

PART -B
EXPERIMENT NO. : -1

Aim :- Study of 8051 microcontroller kit.

Apparatus:- 8051 Microcontroller Kit.

Theory:- System Hardware:- 8bit 8051 Microcontroller at 12Mhz.

Memory:-

32KB EPROM contains firmware.

32 KB programmable/Data RAM.

LCD Display Interface:-

LCD display of 40 characters by two lines is provided.

Key Board Interface:-

An ASCII keyboard for input.

3. Serial Interface:-

Rs-232 port is provided for serial interface 8250 USART along with 1488,1489 driver chips provided necessary signals brought out on a 9-pin D type connector. Baud rates from 300to 9600 can be selected through software.

Timer (CN5):-

Three channels of 16-bit. Timer counter are provided using 8254. The signal are brought out on 10-pin FRC connector(CN5).

5. Parallel I/O Interface:-

Two 8255's are present on board. All the 48 I/O are brought out on 2 no:s of 26 Pin FRC male connector compatible with cards(CN2,CN4).

Study Card Support:

A 50 pin FRC male connector(p1) provided is for bus expansion purpose. All the address and data lines along with interrupts are terminated on this connector. Dyna series study cards can be interfaced through this connector.

Options for DYNA-51:

SMPS model SMPS-03A.
serial cable for RS-232 port
26-pin FRC cables for interfacing 8255 PIO lines.
PIO cards from DYNA log supported on 8255 interface connector.
Dyna series study cards from Dyna log.
Power Supply:
The kit is used with the dyna log SMPS 03 model power supply. The six pin female connector can be plugged in 6-pin male connector soldered on board. Power requirement of Dyna-51 board is:

+5V	3 Amps
+12V	1Amps
-12V	0.5Amps

CN6 Serial Connector:-

All the signals for the RS 232C compatible. Serial interface are brought out on the 9 pin D type male(DTM)connector.The serial cable can be directly connected to this connector.

CN5 for timer Signal10:-10 pin FRC connector is provided which has timer interface lines terminated on it.

CN2 & CN4 for 8255 I/O Interface:-

Two 26 pin FRC male connecter provided on board for 8255's I/O interface. Connector can be used for interfacing DYNA-PIO cards Centornics printer can be interfaced through this connector.

PI for System Signal Bus:-

A 50 pin FRC male connector provided is for Bus expansion. All the address data and control lines along with interrupt lines are terminated on this connector. The same connector is used to interface Dyna-51 with dyna.

Installation Procedure:-

First connect power supply(SMPS-03a)cable (6 pin female connector) to the system supply connector (CN3) with proper orientation.
For serial mode connect the serial cable to 9-pin DTM connector.
Switch on the power supply and then LCD display will show

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

DYNALOG INDIA LTD.
PRESS 'S' FOR SERIAL OR 'CR' FOR KEY MODE.

THE 8051 ARCHITECTURE
THE VARIOUS PARTS IN 8051 PROGRAMMING MODEL ARE:-
16 bit program counter and 16 bit data pointer register(DTPR).
8-bit stack pointer.
Internal ROM or EPROM(8751).
Internal RAM of 128 bytes.
4 Register banks each containing 8 registers.
16 bytes bit addressable area.
80 bytes of general purpose data memory.
32 I/O pins arranged as 4 8-bit port.
2 16-bit timers/counters,(Tl0+Th0) (Tl1+Th1).
Full duplex serial data receiver/transmitter.
Control registers TCOM,PCON,TMOD,IE,P+SCON.
Two external and 3 internal interrupt sources.

PCON-(to control power) Power control.
SBUF-Serial port data buffer.
TMOD-Timer/counter mode control.
TL0-Timer 0 low byte.
TH0-Timer 0 high byte.
TL1-Timer 1 low byte.
TH1-Timer 1 high byte.
A-Accumulator
B-Arithmetic
PC-program Counter.
DPH-Data pointer high byte.
DPL-Data pointer low byte.
PSW-Program status word.
IE-Interrupt Control.
IP-Interrupt Priority.
SCON-Serial counter control.
TCON-Timer/counter control.
System Software:-

Local Mode:

This mode supports following commands:

Command	Function	Syntax
D	Display Memory	D strt end <CR>
E	Edit Memory	E strt <CR>
C	Copy memory	C strt end destination CR>
F	Fill memory	F strt end data <CR>
I	Insert byte	I strt end data<CR>
DL	Delete byte	DL strt end <CR>
S	Search byte	S strt end data <CR>
CM	Compare block	CM strt end destination <CR>
HD	Hex to Dec conversion	HD hex value <CR>
DH	Decimal to Hex conversion	DH decimal value <CR>
A	Assemble command	A strt address <CR>
U	Dissemble Command	U strt end <CR>
I	Port input	IN address <CR>
O	Port output	O address data <CR>
GO	Run the program	GO address <CR>
T	Single step	T strt address <CR>
BR	Enter break point	BR partition
BRD	Display break point	BRD <CR>
BRE	Delete break point	BRE BR no <CR>
BRC	Clear break point	BRC <CR>
R	Register Display	R<CR>
Rno.	Edit register	Rno=data

Command	Function	Syntax
D	Display Memory	D strt end <CR>
E	Edit Memory	E strt <CR>
C	Copy memory	C strt end destination <CR>
F	Fill memory	F strt end data <CR>
I	Insert byte	I strt end data<CR>
DL	Delete byte	DL strt end <CR>
S	Search byte	S strt end data <CR>

CM	Compare block	CM strt end destination <CR>
HD	Hex to Dec conversion	HD hex value <CR>
DH	Decimal to Hex conversion	DH decimal value <CR>
A	Assemble command	A strt address <CR>
U	Dissemble Command	U strt end <CR>
I	Port input	IN address <CR>
O	Port output	O address data <CR>
GO	Run the program	GO address <CR>
T	Single step	T strt address <CR>
BR	Enter break point	BR partition
BRD	Display break point	BRD <CR>
BRE	Delete break point	BRE BR no <CR>
BRC	Clear break point	BRC <CR>
R	Register Display	R<CR>
Rno.	Edit register	Rno=data
L	Load File(Download)	L <CR>
W	Write file upload	W strt end <CR>

2.Serial Mode:
This mode supports following commands

.

EXPERIMENT NO. : -2

AIM:-Write a program to add two numbers lying at two memory locations and display the result in accumulator.

APPARATUS:- 8051 microcontroller kit.

PROGRAM:-

```
MOV 32H,#02H
MOV 35H,#03H
MOV A,32H
MOV R0,35H
```

ADD A,R0

THEORY:-

1. MOV 32,#02H:- This instruction is used to move the hex value '02' in the memory location 32H.

MOV 35,#03H:- This instruction is used to move the hex value '03' in the memory location 35H.

3. MOV A,32H:- This instruction is used to move the contents of memory location 32H in register A.

4. MOV R0,35H:- This instruction is used to move the contents of memory location 35H in register R0.
the register R0.

5. ADD A,R0
This instruction is used to add the contents of the accumlator and temporary register bank and stored the result in accumlator.

[A]←[A]+[R0]

6. BRC (break points to be cleared)
This instruction is used to clear all the break points which are generated before.

7. BR =Last Memory Location
This instruction is used to set the break point at the last memory location or end of Program.
8. GO 'Starting address'
This instruction is used to execute the program from first memory location
PROCEDURE:-
Turn on the 8051 microcontroller kit.
Press enter .
Write A for assemble mode & select memory location.
Now enter the program.
Finally ,the program is executed using GO instruction

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

ASSEMBLY LANGUAGE SHEET

Address	Opcode	Lable	Mnemonic / Operands	Comments
			A 9600	
9600			MOV 32H,#02H	
9603			MOV 35H,#03H	
9606			MOV A,32H	
9608			MOV R0,35H	
960A			ADD A,R0	
			BRC	
			BR=960A	
			GO 9600	
Data:	[A] ← 02H			
	[R0] ←03H			
Result:	[A] ← 05H			

CONCLUSION:- Addition of two no's. is done and result is displayed.
RESULT:- 02 + 03 = 05H

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

EXPERIMENT NO. : -3

AIM:-Write a program to multiply two numbers lying at two memory location and display the result.

APPARATUS:- 8051 microcontroller kit.

PROGRAM:-

```
MOV 32H,#02H
MOV 35H,#03H
MOV A,32H
MOV 0F0H,35H
MUL AB
```

THEORY:-

1. MOV 32H, #02H:- This instruction is used to move the hex value '02' in to memory location 32H.

2. MOV 35H, #03H:- This instruction is used to move the hex value '03' in to memory location 35H.

3. MOV A,32H:- This instruction is used to move the value of memory location 32H in to the accumulator.

4. MOV 0F0H ,35H:- This instruction is used to move the value of memory location 35H in to the accumulator.

5. MUL AB:- This instruction is used to multiply the contents of the accumlator and register B and store the result in accumlator.

[A]←[A].[B]

6. BRC (break points to be cleared)
This instruction is used to clear all the break points which are generated before.

7. BR =Last Memory Location

This instruction is used to set the break point at the last memory location or end of Program.

8. GO 'Starting address'

This instruction is used to execute the stored program at memory location 9600.

PROCEDURE:-

Turn on the 8051 microcontroller kit.
Press enter .
Write A for assemble mode & select memory location.
Now enter the program.
Finally ,the program is executed using GO instruction

ASSEMBLY LANGUAGE SHEET

Address	Opcode	Lable	Mnemonic / Operands	Comments
			A 9600	
9600			MOV 32H,#02H	
9603			MOV 35H,#03H	
9606			MOV A,32H	
9608			MOV 0F0H,35H	
960B			MUL AB	
			BRC	
			BR=960B	
			GO 9600	
Data:	[A] ← 02H			
	[B] ←03H			
Result:	[A] ← 06H			

CONCLUSION:- Multiplication of two no's. is done and result is displayed.
RESULT:- 02 X 03 = 06H

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

EXPERIMENT NO 4

AIM:-Write a program to arrange 10 nos stored in memory locations in ascending order and show them on display.

APPARATUS:- 8051 microcontroller kit

PROGRAM:-

```
LOOP1: MOV R0, #0Fh          (Counter for LOOP1)
MOV DPTR, #9000h                  (Point to beginning of array)
MOV A, R0
MOV R1, A                    (Initialize R1 - the counter for LOOP2)
LOOP2: MOVX A,@DPTR          (Copy a number of the array to the a
                              Accumulator)
MOV R2, A                    (and store it in R2)
INC DPTR                     (Move to the next number)
MOVX A, @DPTR                (and store that in the accumulator)
SUBB A, R2                   (Subtract the first from the second)
JNC Continue2                (If no carry is generated the 2nd is g
                              Greater & the no are)
MOVX A, @DPTR                (Move the second number to the accu)
XCH A, R2                   (Exchange contents of the accu. and R2.
This Makes A contain the first number and  R2 the  Second)
MOVX @DPTR,A   (Store the first number at the place where the Second one was
stored)
DEC DPL                  (Move to the previous memory location)
MOV A, R2                (Copy the second number to the accu.)
MOVX @DPTR, A        (and store it in the first number's place)
INC DPTR          (Move to the next memory location)
Conti2: DJNZ R1,LOOP2
Conti1: DJNZ R0, LOOP1        (Move on to the next pass)
Here: SJMP Here
END  (End of program)
```

PROCEDURE:-

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

Turn on the 8051 microcontroller kit.
Press enter.
Write A for assemble mode & select memory location to enter the data.
Now enter the program.
Finally ,the program is executed using GO instruction.

ASSEMBLY LANGUAGE SHEET

Address	Opcode	Lable	Mnemonic / Operands	Comments
9700		LOOP1	MOV R0, #09h	Counter for LOOP1
9702			MOV DPTR, #9600h	Point to beginning of array
9705			MOV A,R0	
9706			MOV R1,A	Initialize R1 - the counter for LOOP2
9707		LOOP2	MOVX A,@DPTR	Copy a number of the array to the a accumulator
9708			MOV R2, A	store accumulator content it in R2
9709			INC DPTR	Move to the next number
970A			MOVX A, @DPTR	store the next number in the accumulator
970B			SUBB A, R2	Subtract no. the first from the second no.
970C			JNC Continue2	If no carry is generated the 2nd is g greater
970E			MOVX A, @DPTR	Move the second number to the accumulator
970F			XCH A, R2	Exchange contents of the accumulator and R2
9710			MOVX @DPTR,A	Store the first number at the place where the Second one was stored
9711			DEC DPL	Decrement the lower byte(82H) of DPTR To move to the previous memory location
9713			MOV A, R2	Copy the second number to the

				accumulator
9714			MOVX @DPTR, A	and store it in the first number's place
9715			INC DPTR	Move to the next memory location
9716		Conti2	DJNZ R1,LOOP2	
9718		Conti1	DJNZ R0, LOOP1	Move on to the next pass
971A		Here	SJMP Here	
		END		End of program
			BRC	
			BR=971C	
			GO 9700	

DATA :-

9600 - 05
9601 - 03
9602 - 01
9603 - 08
9604 - 07
9605 - 06
9606 - 02
9607 - 09
9608 - 0A
9609 - 13

CONCLUSION:- Given 10 numbers are arranged in ascending order and are stored on the given memory location.

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

EXPERIMENT NO 4 (b)

AIM:-Write a program to arrange 10 nos stored in memory locations in Descending order and show them on display.

APPARATUS:- 8051 microcontroller kit

PROGRAM:-

```
LOOP1: MOV R0, #0Fh          (Counter for LOOP1)
MOV DPTR, #9000h                  (Point to beginning of array)
MOV A, R0
MOV R1, A                    (Initialize R1 - the counter for LOOP2)
LOOP2: MOVX A,@DPTR          (Copy a number of the array to the a
                              Accumulator)
MOV R2, A                    (and store it in R2)
INC DPTR                     (Move to the next number)
MOVX A, @DPTR                (and store that in the accumulator)
SUBB A, R2                   (Subtract the first from the second)

JC Continue2                 (If no carry is generated the 2nd is g
                              Greater & the no are)
MOVX A, @DPTR                (Move the second number to the accu)
XCH A, R2          (Exchange contents of the accu. and R2. This Makes A contain the
first number and  R2 the  Second)
MOVX @DPTR,A (Store the first number at the place where the Second one was
stored)
DEC DPL                    (Move to the previous memory location)
MOV A, R2                  (Copy the second number to the accu.)
MOVX @DPTR, A              (and store it in the first number's place)
INC DPTR                   (Move to the next memory location)
Conti2: DJNZ R1,LOOP2
Conti1: DJNZ R0, LOOP1          (Move on to the next pass)
Here: SJMP Here
END  (End of program)
```

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

PROCEDURE:-

Turn on the 8051 microcontroller kit.
Press enter .
Write A for assemble mode & select memory location to enter the data.
Now enter the program.
Finally ,the program is executed using GO instruction

ASSEMBLY LANGUAGE SHEET

Address	Opcode	Lable	Mnemonic / Operands	Comments
9700		LOOP1	MOV R0, #09h	Counter for LOOP1
9702			MOV DPTR, #9600h	Point to beginning of array
9705			MOV A,R0	
9706			MOV R1,A	Initialize R1 - the counter for LOOP2
9707		LOOP2	MOVX A,@DPTR	Copy a number of the array to the a accumulator
9708			MOV R2, A	store accumulator content it in R2
9709			INC DPTR	Move to the next number
970A			MOVX A, @DPTR	store the next number in the accumulator
970B			SUBB A, R2	Subtract no. the first from the second no.
970C			JC Continue2	If no carry is generated the 2nd is g greater
970E			MOVX A, @DPTR	Move the second number to the accumulator
970F			XCH A, R2	Exchange contents of the accumulator and R2
9710			MOVX @DPTR,A	Store the first number at the place where the Second one was stored
9711			DEC DPL	Decrement the lower byte(82H) of DPTR To move to the previous

				memory location
9713			MOV A, R2	Copy the second number to the accumulator
9714			MOVX @DPTR, A	and store it in the first number's place
9715			INC DPTR	Move to the next memory location
9716		Conti2	DJNZ R1,LOOP2	
9718		Conti1	DJNZ R0, LOOP1	Move on to the next pass
971A		Here	SJMP Here	
		END		End of program
			BRC	
			BR=971C	
			GO 9700	

DATA :-

9600 - 05
9601 - 03
9602 - 01
9603 - 08
9604 - 07
9605 - 06
9606 - 02
9607 - 09
9608 - 0A
9609 - 13

CONCLUSION:- Given 10 numbers are arranged in ascending order and are stored on the given memory location

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

EXPERIMENT NO.: 5

AIM: Write a program to show the use of INT0 and INT1.

APPARATUS: 8051 micro controller kit.

PROGRAM:

(i) Program to show the use of INT0

```
        ORG    0000H
        LJMP   MAIN
        ORG    000BH
        CPL    P1.2
  MOV   TL0, #00H
        MOV   TH0, #0DCH
        RETI
  ORG    30H
MAIN: MOV    TMOD, #00000001B
   MOV    TL0,  #00H
          MOV   TH0, #0DCH
          MOV    IE, #82H
          SETB    TR0
HERE:  SJMP    HERE
          END
```

(ii) Program to show the use of INT1

```
        ORG    0000H
        LJMP   MAIN

        ORG    001BH
        LJMP   ISR_T1

  ORG    0030H
MAIN: MOV   TMOD, #10H
        MOV   P0, #FFH
        MOV   TL1, #018H
```

```
        MOV   TH1, #0FCH
        MOV   IE, #88H
        SETB  TR1
BACK:   MOV   A, P0
        MOV   P1, A
        SJMP  BACK

ISR_T1: CLR   TR1
        CLR   P2.1
        MOV   R2, #4
HERE:   DJNZ  R2, HERE
        MOV   TL1, #18H
        MOV   TH1, #0FCH
        SETB  TR1
        SETB  P2.1
        RETI
        END
```

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

EXPERIMENT NO.: 6

AIM: WAP of Flashing LEDs connected at port 0 of the microcontroller.

Apparatus: 8051 micro controller kit

PROGRAM:

```
start:  mov   p0,#00h
        acall  delay
        mov   p0,#0ffh
        ljmp   start

delay:mov tmod,#10h
        mov r6,#5h
again1:mov r3,#0ffh
again:mov tl1,#08h
        mov th1,#01h
        setb tr1
back:  jnb tf1,back
        clr tr1
        clr tf1
        djnz r3,again
        djnz r6,again1
        ret
        end
```

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

EXPERIMENT NO. 7

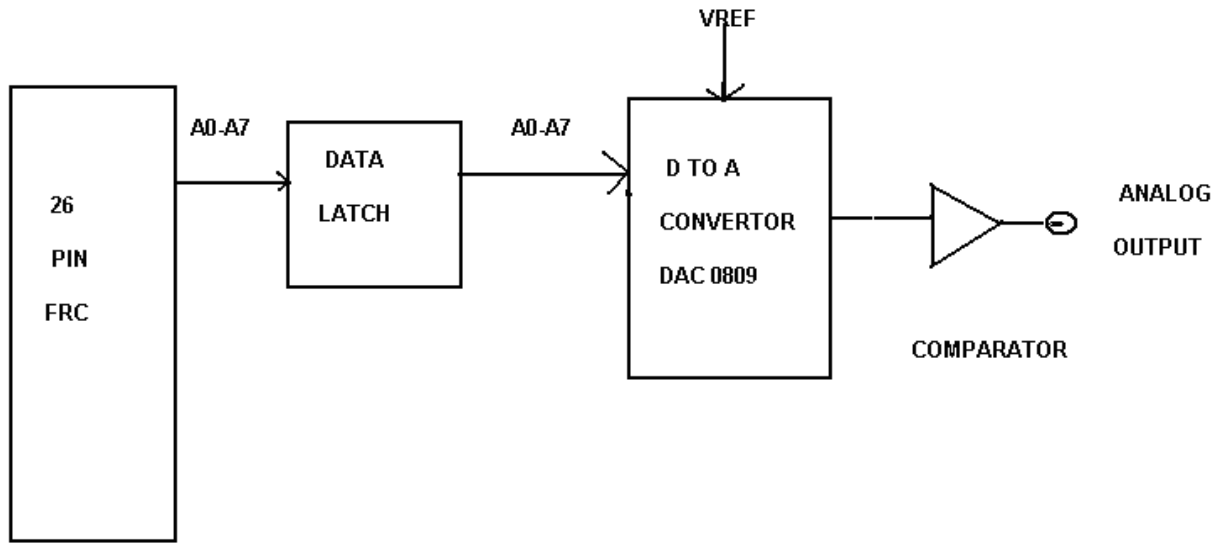
AIM:- Write a program to generate a Ramp waveform using DAC with micro controller

APPARATUS:- 8051 microcontroller kit,DAC kit ,CRO connecting leads.

THEORY:-

DAC mean digital to analog converter which take digital signal and gives analog signal as output. Here we use DAC to generate analog signal of ramp wave by microcontroller 8051 kit which will done by using DAC . we will give the input to microcontroller by writing the program then DAC will be given the digital signal by microcontroller at its input and it will convert it in the ramp wave which can be observed o CRO screen.

BLOCK DIAGRAM:-



	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

PROCEDURE:-

Connect the DAC study card with the 8051 MC kit by 26 pin connecter as shown in figure.

Connect the CRO at the output of DAC .

Turn on the 8051 microcontroller kit.

Now enter the program.

Give the command PIO 5 the program is executed and observe waveform will display on kit screen and the waveform will be shown on CRO.

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

EXPERIMENT NO. 8

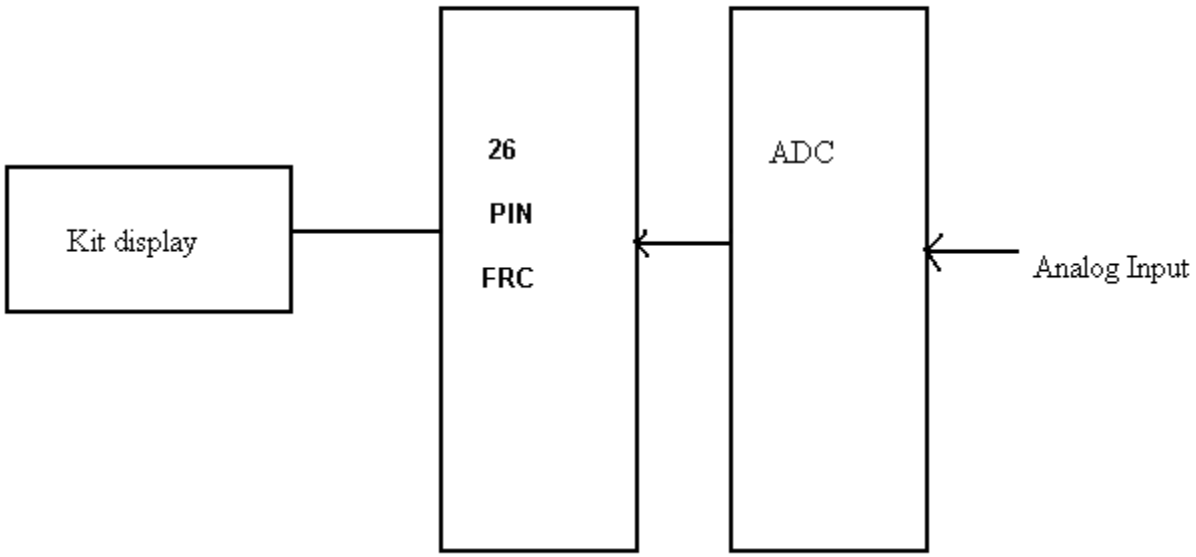
AIM:- Write a program to interface the ADC .

APPARATUS:- 8051 microcontroller kit.

THEORY:-

Analog-to-digital converters are among the most widely used devices for data acquisition. Digital computers use binary (discrete) values, but in the physical world everything is analog (continuous). ADC mean analog to digital converter which take analog signal and gives digital signal as output. we will give the input to ADC from a source of analog signal and write a program for microcontroller according to that the ADC will give the digital signal at its output.

BLOCK DIAGRAM:-



	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

PROCEDURE:-

Connect the ADC study card with the 8051 MC kit by 26 pin connecter as shown in figure.

Connect the Analog Signal source at the input of ADC .

Turn on the 8051 microcontroller kit.

Now enter the program.

Give the command PIO 1 the program is executed and the digital equivalent of signal display on kit screen.

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

Experiment N0.9

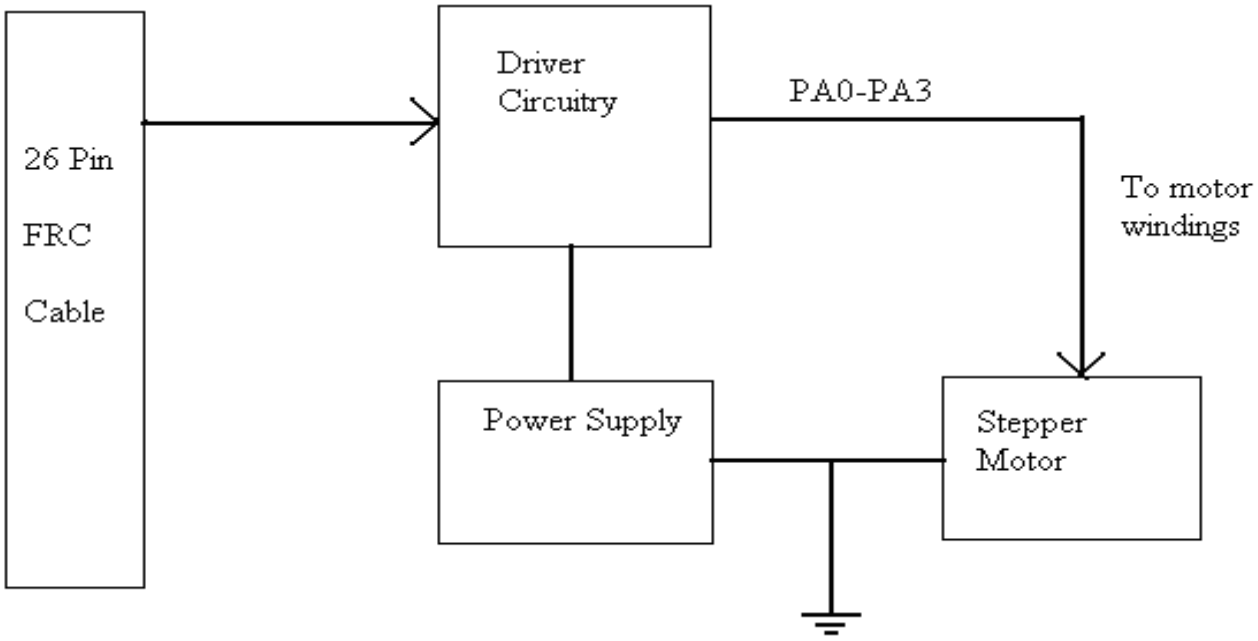
AIM:- Write a program to control a stepper motor in direction, speed and number of steps.

APPARATUS:- 8051 microcontroller kit.

THEORY:-

Stepper motor rotates in the form of steps ,and the stepping action is caused by sequential switching supply to the two phases of motor .the stepper motor is of bifilar type with 6 leads. Each of the two phases of motor has double winding with a centre tap. Switching the supply from one side to another of the windings can cause reversal of magnetic polarity without actually reversing the polarity of supply.

BLOCK DIAGRAM:-



PROCEDURE:-

Connect the components as shown in figure.
Turn on the 8051 microcontroller kit.
Give the command PIO 4 the demonstration program will be executed follow the steps which appear on screen of microcontroller kit to control the speed , direction & steps of stepper motor.

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

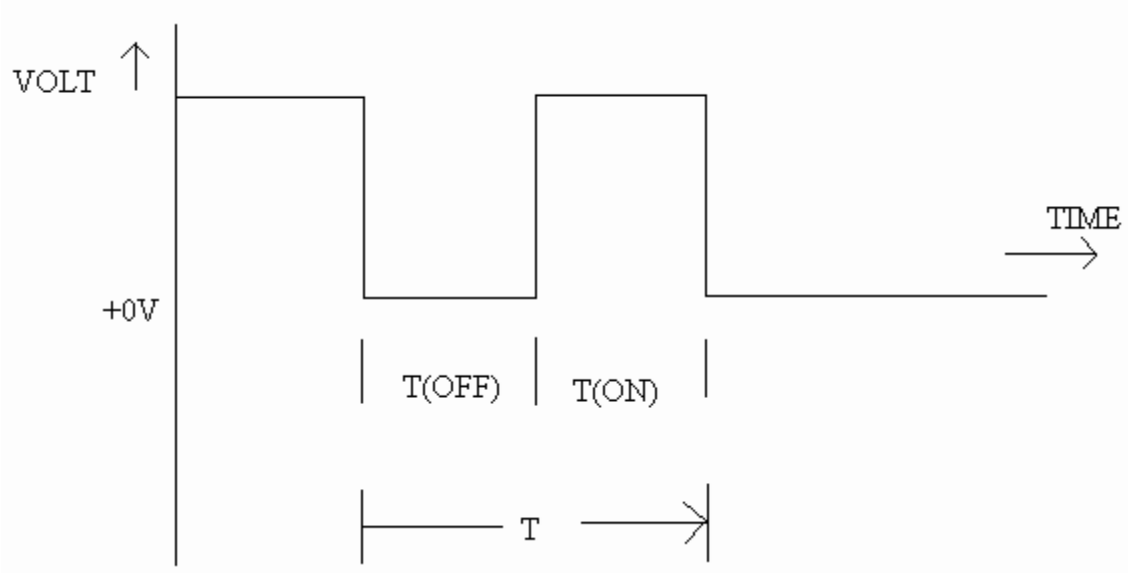
Experiment N0.10

AIM:- Write a program to control the speed of DC motor .

APPARATUS:- 8051 microcontroller kit .

THEORY:-

DC motor is used in tape recorders,VCR's. Speed of Dc motor is directly proportional to the DC voltage applied across it. By switching voltage applied the motor can be turned ON/OFF. The wave form is shown here.



The motor will rotate during the ON time and will remain ideal during OFF time. By varying the pulse width of rectangular waveform we can vary the speed of DC motor. The direction of motor can only be change by changing the polarity of applied voltage.

PROCEDURE:-

Connect the DC motor study card with the 8051 MC kit by 50 pin connecter.
Turn on the 8051 microcontroller kit.
Enter the program and execute the program by GO command.
Now enter the direction and speed number.

The data format is divided in two parts. First digit shows the direction of rotation second part f data show the speed number which is directly proportional to duty cycle of the applied voltage waveform.

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

EXPERIMENT NO.: 11

Aim:Interfacing of high power devices to Micro-controller port-lines, LED, relays and LCD display.

Apparatus: 8051 micro controller kit.

Program:

(i) Interfacing with LED

```
start:  mov p0,#0ffh
        acall delay
        mov p0,#0h
        ljmp start

delay:  mov tmod,#10h
        mov r6,#5h
again1:     mov r3,#0ffh
again:  mov tl1,#08h
        mov th1,#01h
        setb tr1
back:   jnb tf1,back
        clr tr1
        clr tf1
        djnz r3,again
        djnz r6,again1
        ret
        end
```

(ii) Interfacing with Relays

```
        ORG   0000H
        LJMP  MAIN

        ORG   0013H
        SETB  P1.3
```

```
        MOV   R3, #255
BACK: DJNZ   R3, BACK
        CLR    P1.3
        RETI

        ORG    30H
MAIN:  MOV   IE, #10000100B
HERE:  SJMP   HERE
        END
```

(iii) Interfacing with LCD

```
        mov   a,#38h
        acall   com
        acall   DELAY
        mov   a,#0eh
        acall   com
        acall   DELAY
        mov   a,#01h
        acall   com
        acall   DELAY
        mov   a,#06h
        acall   com
        acall   DELAY
        mov   a,#84h
        acall   com
        acall   DELAY
        mov   a,#'S'
        acall   dwrt
        acall   DELAY
        mov   a,#'A'
        acall   dwrt
        acall   DELAY
        mov   a,#'T'
        acall   dwrt
        acall   DELAY
```

```
        mov   a,#'W'
        acall dwrt
        acall DELAY
        mov   a,#'I'
        acall dwrt
        acall DELAY
        mov   a,#'N'
        acall dwrt
        acall DELAY
        mov   a,#'D'
        acall dwrt
        acall DELAY
again:  ljmp  again

com:    mov   p0,a
        clr   p2.0
        clr   p2.2
        setb  p2.1
        clr   p2.1
        ret
dwrt:   mov   p0,a
        setb  p2.0
        clr   p2.2
        setb  p2.1
        clr   p2.1
        ret

delay:  mov   tmod,#10h
        mov   tl1,#08h
        mov   th1,#01h
        setb  tr1
ag:     jnb   tf1,ag
        clr   tr1
        clr   tf1
ret
        end
```

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

Experiment No. 12

Aim: To transmit the message 'YES' serially at a baud rate of 9600 bps.

Apparatus: 8051 Microcontroller Kit, Serial Cable, a PC with DB-9 Port and Hyper-terminal software installed in it.

Theory:

In serial communication, the data is sent one bit at a time and uses only two wire setup in order to achieve a full duplex communication. Serial Communication is by far most used way of communication and almost all the devices practically communicate serially. 8051 has got one full duplex UART (Universal Asynchronous Receiver Transmitter). It uses TTL standard, so we need a TTL to RS232 converter. MAX232 is one such IC. It converts TTL standard signal i.e. 0V and 5V mapped to binary 0 and binary 1, respectively, to RS232 standard which generally use +12V and -12V to map binary 0 and binary 1, respectively.

In order to use 8051's built in UART; we need to know a little bit about setting it up and controlling the flow of communication.

SCON Register: SCON (Serial Control) register is use to control the UART so as to communicate serially with other devices. It is an 8 bit register and is bit addressable. The bits are as follow:

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0 and SM1 are use to setup mode of the serial communication. Only mode 1 is used these days. So, we will not consider any other mode.

SM2 is used in multiprocessor environment and is not required here, so make it 0.

REN is the receive enable bit and is used in full duplex mode. Since we are just transmitting the data, we will clear this bit.

TB8 and RB8 are not used in mode 1, so we will make them 0 too.

TI is the transmit interrupt bit and is set by hardware thus, we will not need to set it but we will be reading this bit to know when our data is sent successfully.

RI is Receive Interrupt. Since, we are not using receiving part, so we need not to consider this bit. Again, this bit is set by hardware, when a byte of data is received.

SBUF Register: SBUF register is an 8 bit register and is known as serial buffer. Any data to be transmitted serially need to be present in this register. Similarly, when a data byte is received, it is initially present in SBUF and need to be moved to somewhere else before the next byte overwrites it.

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

Setting up BAUD Rate: The BAUD rate is known as the number of bits transferred per second. Generally, the BAUD rate can be set arbitrarily, but, for the purpose of standardization between different vendors manufacturing same device, a standard value is listed. We will be using a very common value, 9600bps. In 8051, baud rate is set by timer 1. In mode 2 of timer 1, we will initialize the value 0FDH in TH1 in order to setup the BAUD rate equal to 9600. This is calculated as follow:

TH1 = 256 - ((Crystal / 384) / Baud), when PCON.7 bit is clear

TH1 = 256 - ((Crystal / 192) / Baud), when PCON.7 bit is set

Procedure:

Connect the PC with 8051 kit using a serial cable.

Setup Hyper-terminal to listen to the COM port on which the kit is connected. If PC has only one port, the possible com port is COM1.

Setup the BAUD rate to 9600 and parity bit to none in Hyper-terminal.

Now enter the program in 8051 kit by pressing "A 9600" and then the assembly code.

Insert the data to the data area 9800 and so on.

Now run the program by first pressing "BRC" and then creating break point at address of the last instruction, "BR <address>" and then pressing "GO 9600".

On successful execution, "YES" will be displayed on the terminal window of the Hyper-terminal program.

Precautions:

Make the connection securely and correctly.

Result: The UART of 8051 is studied and message "YES" is transmitted serially at a baud rate of 9600 and is displayed.

Data:

Address	Character	ASCII Value (Hex)
9800	Y	59H
9801	E	45H
9802	S	53H
9803	0	00H

	SRI SUKHMANI INSTITUTE OF ENGINEERING & TECHNOLOGY DERA BASSI		
	DEPARTMENT: ELECTRICTRONICS AND COMMUNICATION ENGINEERING		
	LABORATORY MANUAL		
	LAB: HARDWARE PROGRAMME & INTERFACEING	**SUBJECT CODE:** BTEC -508	**SEMESTER:** 5th

Program:

Address	Label	Mnemonic	Comments
9600		MOV DPTR, #9800	DPTR pointing to first data address
9603	Loop	MOVX A, @DPTR	Move the content at address pointed by DPTR to Accumulator
9604		JZ Done	Jump if we have no more letters left
9606		ACALL SerTx	Run the subroutine to send a byte of data serially
9608		INC DPTR	Get the address of next data byte
9609		SJMP Loop	Go back to the label loop to transmit next data
960B	Done	SJMP Done	Loop here endlessly
960C	SerTx	MOV TMOD, #20H	Set Timer 1 to Mode 2
960E		MOV TH1, #0FDH	Set TH1 = FDH so as to set BAUD rate of 9600bps
9610		MOV SCON, #40H	Set UART to Serial mode 1 in transmit only mode.
9612		SETB TR1	Start the Timer 1
9613		MOV SBUF, A	Put the data to be transmitted in the Serial Buffer
9614	Here	JNB TI, Here	Loop here until the Transmit Interrupt bit is raised
9616		CLR TR1	Stop the Timer 1

9617		RET	Return to calling routine

Experiment No. 13

Aim: To generate a pulse train with a 66% duty cycle on Pin P3.0.

Apparatus: 8051 kit, CRO, Connecting Cable with alligator clips.

Theory:

8051 has two hardware timers that can generate precise delays. Since these are hardware timers, these are not affected by processing speed etc.

TMOD Register: TMOD or timer mode register is used to setup the timer mode. It is an 8 bit register and is used to setup the functioning of both Timer 1 and Timer 0.

Gate	C/T	M1	M0	Gate	C/T	M1	M0
←|	Timer 1			→|←	Timer 0		→|

GATE control, when set, the timer/counter is enabled only while the INTx pin is high and TRx bit is set. When it is cleared, the timer is enabled whenever TRx control bit is set.

C/T control, when set, the timer behave as a counter and increment at each pulse at Tx Pin, where x denote 0 and 1 for T0 and T1, respectively. When cleared, it works as a timer.

Modes :	M1	M0	Mode
	0	0	13 bit Timer (Mode 0)
	0	1	16 bit Timer (Mode 1)
	1	0	8 bit auto reload Timer (Mode 2)
	1	1	Split Timer (Mode 3)

TCON Register: TCON or Timer Control register is an 8 bit, bit addressable register.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TFx bit is Timer X overflow bit, so, when timer rolls from FFFFH (16 bit) or 7FFFH (13 bit) or FFH (8 bit), to 0000H, this bit is set by hardware.

TRx bit is the Timer X run bit. It is set by program when timer needs to start. When cleared, the timer X stops.

IEx bit is used to setup external interrupt to be addressed when a high to low edge signal is applied at INTx Pin. This bit is not related to timer.

THx and TLx Registers: Since, 8051 has 16 bit timer, we need two 8 bit registers to store complete 16 bit value since 8051 itself is an 8 bit microcontroller. The THx and TLx registers are filled with some initial value and then timer is started. The timer counts from that initial value stored in THx and TLx and counts up to 0FFFFH. Therefore, higher the value in the THx-TLx register pair, lesser will be the time taken

by timer to count thus less will be the delay generated. In 8-bit auto reload mode, THx contain the initial value and each timer the timer is overflowed, it is reloaded by this initial value. Thus, THx work as a parallel load for TLx register as TLx register actually counts in this mode.

Timing and Delay Calculations: 8051 takes 12 cycles to complete one machine cycle, thus, assuming the crystal X_{TAL} = 11.0592 MHz,

Frequency of the Timer = $11.0592 \times 10^6 / 12 = 921.16$ KHz

Delay generated in 1 Cycle = $1 / 921.16 \times 10^3 = 1.085$ μs

To find the value to be loaded in THx and TLx register:

Divide the desired time delay by 1.085 μs. Let it be n.

Perform 65536 – n. Let it be m.

Now, convert m into hexadecimal equivalent as yyxxH or 0xYYXX.

Now, TLx = xxH and THx = yyH.

Our aim is to generate 66% duty cycle. Let us take 1ms of total period. Then,

$n = 1 \times 10^{-3} / 1.085 \times 10^{-6} = 921.16$ = Round off = 922

m = 65536 – 922 = 64614

M = m (in hex) = 0FC66H

Now, 66% of 1ms is 0.66 ms

$n = 0.66 \times 10^{-3} / 1.085 \times 10^{-6} = 608$ (Rounded Off)

m = 65536 – 608 = 64928

M = m (in hex) = 0FDA0H

Now, for off time, we have

n = 922 – 608 = 314

m = 65536 – 314 = 65222

M = m (in hex) = 0FEC6H

Therefore, the values to be loaded in THx and TLx are:

ON DELAY:

THx = 0FDH

TLx = 0A0H

OFF DELAY:

THX = 0FEH

TLX = 0C6H

Procedure:

Connect P3.0 Pin of the Kit with CRO.

Assemble the Program by Pressing “A 9600” and then entering the program.

Execute the program by, first clearing the break points "BRC" and then creating a breakpoint at the address of last instruction. i.e. "BR <address>" and then pressing "GO 9600".

See the square wave on the CRO display with a 66% of duty cycle.

Program:

Address	Label	Mnemonics	Comments
9600		CLR P3.0	Make Pin P3.0 ready as output pin
9602	Loop	SETB P3.0	Set the P3.0 pin, Setting up the signal's high data
9604		ACALL ON_DELAY	Call the subroutine to add precise delay of 0.66ms
9606		CLR P3.0	Clear P3.0 pin, setting up the signal's low data
9608		ACALL OFF_DELAY	Call the subroutine to add precise delay of 0.34ms
960A		SJMP Loop	Continue this forever thus generating a clock signal having duty cycle of 66%
960C	ON_DELAY	MOV TMOD, #01H	Setting up Timer 0 Mode 1
960E		MOV TH0, #0FDH	Set High Byte
9610		MOV TL0, #0A0H	Set Low Byte
9612		SETB TR0	Start Timer 0
9614	Here1	JNB TF0, Here1	Loop unless Timer Completes the counting
9617		CLR TF0	Clear the Timer Flag 0
9619		CLR TR0	Stops the Timer 0
961B		RET	Return to calling routine
961C	OFF_DELAY	MOV TMOD, #01H	Setting up Timer 0 Mode 1
961E		MOV TH0, #0FEH	Set High Byte
9620		MOV TL0, #0C6H	Set Low Byte
9622		SETB TR0	Starts the Timer 0
9624	HERE2	JNB TF0, HERE2	Loop unless Timer completes the counting

9627		CLR TF0	Clear Timer Flag 0
9629		CLR TR0	Stops the Timer 0
962B		RET	Return to calling routine

Result: The timer in 8051 is studied and a square-wave signal having 66% duty cycle is obtained and displayed on CRO.