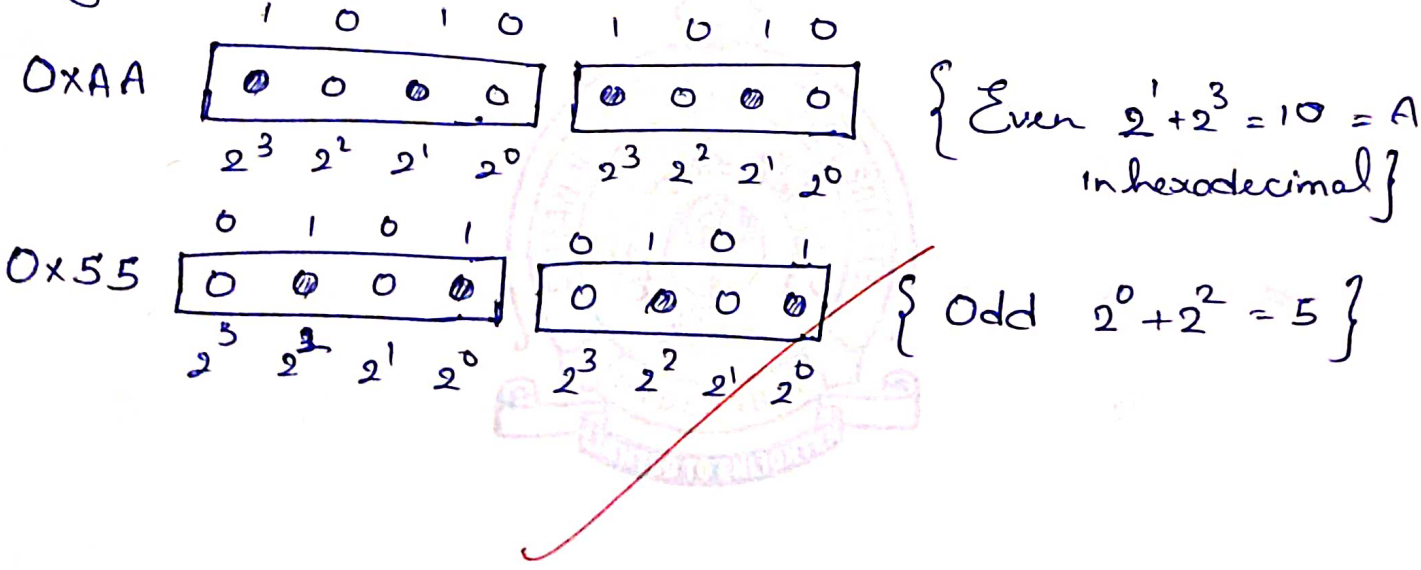
Expt. No. 01 Page No. ..... Date : .....

AIM:- An Embedded C program to demonstrate the Interfacing of 8051 with LED by glowing alternate LED's on microcontroller kit.

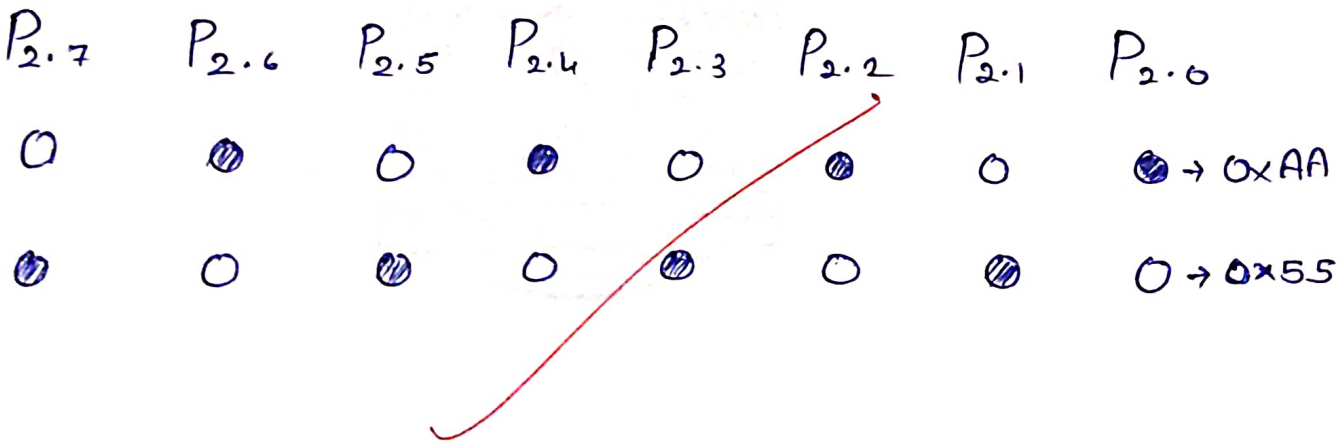
DESCRIPTION:- In this program, our objective is to make LEDs glow alternatively. To achieve this, we will initialize the LEDs with 0x55 and 0xAA and then glow the LEDS in alternating fashion. To ensure that the pattern is Clearly visible, we will add a delay function between each LED Change.



CODE:

```
//led.c
#include<REGX51.h>
#define LED P2
void delay(unsigned int d);
int main(void)
{
      while(1)
      {
      LED=0x55;
      delay(1000);
      LED=0xAA;
      delay(1000);
      }
}
void delay(unsigned int d)
{
      unsigned int i,j;
      for(i=0;i<d;i++)
      for(j=0;j<100;j++);
}
```

OUTPUT:



Expt. No. 02 Page No. ..... Date : .....

AIM: An Embedded C-program to demonstrate interfacing of 8051 with LED to shift the bits from left to right and right to left.

DESCRIPTION:. The port 2 consists of 8 LEDs. The program initializes the LED. Next step is to assign the values 0x01. In the loop, it sequentially shifts the bits using shift operator & sends data to LED. After each shift, it provides a small delay to make shifting visible. Once the bits are shifted, it reset data to 0x80 & repeat the process.

Expt. No. ..... Page No. ..... Date : .....

OUTPUT:

$P_{2.7}$	$P_{2.6}$	$P_{2.5}$	$P_{2.4}$	$P_{2.3}$	$P_{2.2}$	$P_{2.1}$	$P_{2.0}$
0	0	0	0	0	0	0	●
0	●	0	0	0	0	●	0
0	0	0	0	0	●	0	0
0	0	0	0	●	0	0	0
0	0	0	●	0	0	0	0
0	0	●	0	0	0	0	0
0	●	0	0	0	0	0	0
●	0	0	0	0	0	0	0
0	●	0	0	0	0	0	0
0	0	●	0	0	0	0	0
0	0	0	●	0	0	0	0
0	0	0	0	●	0	0	0
0	0	0	0	0	●	0	0
0	0	0	0	0	0	●	0
0	0	0	0	0	0	0	●

Roll No. : ..... Lecturer's Signature : .....

Expt. No. 03 Page No. ..... Date : .....

AIM:- An Embedded C program to demonstrate interfacing of 8051 with LED by implementing a binary counter.

DESCRIPTION: Here binary counter is implemented to count the value. It sets an initial value, typically 0. In the loop, it increments the counter & updates the LED based on counter value. After each counter it provides a small delay to control speed of counting. If counter reaches it maximum value i.e 15 it reset back to initial value & repeat the counting process.

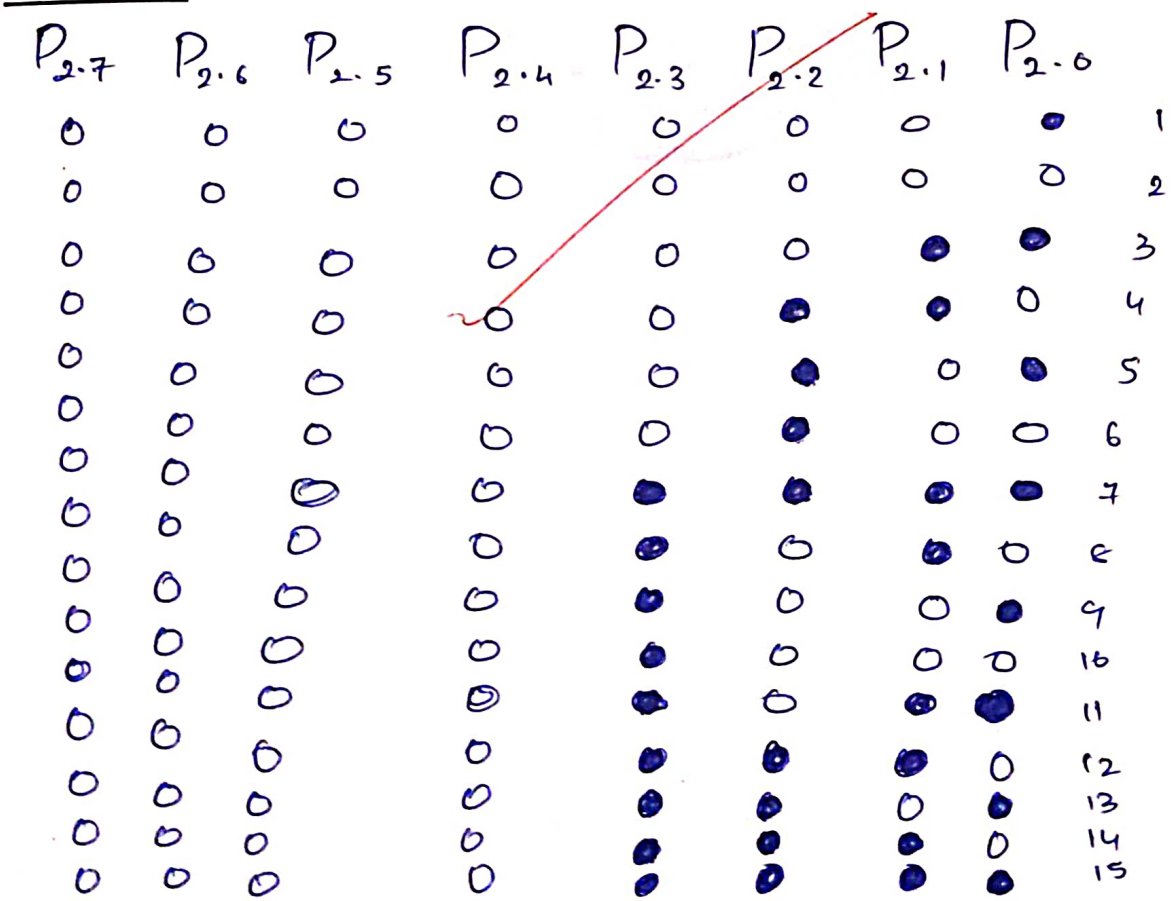
CODE:

```
//binarycnt
#include<REGx51.h>
#define LED P2
void delay(unsigned int);
int main()
{
     unsigned int i;
     while(1)
     {
          LED=0x00;
          for(i=0;i<=15;i++)
          {
               LED++;
               delay(1000);
          }
     }
}

void delay(unsigned int d)
{
     unsigned int i,j;
     for(i=0;i<d;i++)
     for(j=0;j<100;j++);

}
```

OUTPUT:



Roll No. : .....

Lecturer's Signature : .....

Expt. No. 04 Page No. ..... Date : .....

AIM: An Embedded c program to demonstrate interfacing of 8051 with relays & buzzers.

DESCRIPTION: A Relay is a device that responds to small current (or) voltage change by activating switches. On the other hand, Buzzers are devices which makes sounds & mostly used for signaling.

The program set initial state for LED, relays & buzzers (all turned off). Here we create 4 relays 1, 2 & 3 & buzzers for 4, 5, 6, 7th LED's In loop, it check input conditions to determine when to active relays & buzzer. When input is detected, it activates relay & LED as well. It provides small delay to determine duration of relay & buzzers.

CODE:

```
//relbuz.c
#include<REGX51.H>
#define Relay1 P2_4
#define Relay2 P2_5
#define Relay3 P2_6
#define Buzzer P2_7
void  delay(unsigned int);

int main()
{
      P2=0x00;
      while(1)
      {
            Relay1=1;
            delay(1000);
            Relay1=0;
            delay(1000);
            Relay2=1;
            delay(1000);
            Relay2=0;
            delay(1000);
            Relay3=1;
            delay(1000);
            Relay3=0;
            delay(1000);
            Buzzer=1;
            delay(1000);
            Buzzer=0;
            delay(1000);
      }
}
void  delay(unsigned int d)
{
      unsigned int i,j;
      for(i=0;i<d;i++)
      for(j=0;j<100;j++);
}
```

OUTPUT: Buzzer Relay3 Relay2 Relay1

$P_{2.7}$	$P_{2.6}$	$P_{2.5}$	$P_{2.4}$	$P_{2.3}$	$P_{2.2}$	$P_{2.1}$	$P_{2.0}$
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0

Expt. No. 05 Page No. ..... Date : .....

AIM: An Embedded C program to demonstrate interfacing of 8051 with switch

DESCRIPTION: In Embedded System, a switch refers to an electronic component that acts as a digital input device. When the switch is operated, it changes its state & embedded system can detect this state change as a signal.

Expt. No. ..... Page No. ..... Date : .....

CODE:

```
//switches.c
#include<REGX51.h>
#define LED P2
#define SWITCH P1
int main()
{
      LED=~SWITCH;
}
```

OUTPUT:

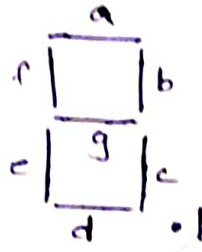
	$P_{2.7}$	$P_{2.6}$	$P_{2.5}$	$P_{2.4}$	$P_{2.3}$	$P_{2.2}$	$P_{2.1}$	$P_{2.0}$
Switches	⊡	⊡	⊡	⊡	⊡	⊡	⊡	⊡
LEDs	$P_{2.7}$	$P_{2.6}$	$P_{2.5}$	$P_{2.4}$	$P_{2.3}$	$P_{2.2}$	$P_{2.1}$	$P_{2.0}$
	0	0	0	0	0	●	0	0

When switch $P_{2.4}$ is pressed, its corresponding LED is glowed i.e $P_{2.0}$. This can be performed for all the other switches.

Expt. No. 06 Page No. Date :

AIM: An Embedded c program to demonstrate interfacing of 8051 with seven segment display

DESCRIPTION: A 7 segment is a form of electric device for displaying numerals i.e an alternative to more complex dot matrix displays. It is composed 7 segments that are individually ON/OF



The 7 segments are represented using the alphabets a b c d e f g & is connected to port 0

The anode pins of 7 segment are connected to port 3.3, 3.4, 3.5 respectively

	h	g	f	e	d	c	b	a	
0	1	1	0	0	0	0	0	0	- 0xC0
1	1	1	1	1	1	0	0	1	- 0xFA
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

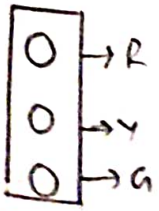
Roll No. : Lecturer's Signature :

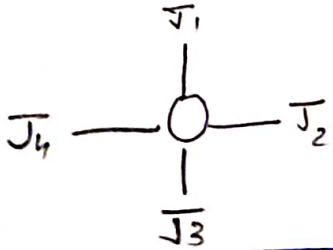
Expt. No. 07 Page No. ..... Date : .....

AIM:- An Embedded C program to demonstrate interfacing of 8051 with LEDs by developing a traffic application

DESCRIPTION:- We demonstrate traffic control signal i.e all possible way where in flow of opposite direction is allowed & also writing hex code for traffic control combination we send data to LED. Example LED(0x99) then delay for some time & repeat the process

J_4		J_3		J_2		J_1		
G	R	G	R	G	R	G	R	
0	1	0	1	0	1	0	0	0x54
0	1	0	1	0	1	1	0	0x56
0	1	0	1	0	0	0	1	0x51
0	1	0	1	1	0	0	1	0x59
0	1	0	0	0	1	0	1	0x45
0	1	1	0	0	1	0	1	0x65
0	0	0	1	0	1	0	1	0x15
1	0	0	1	0	1	0	1	0x95



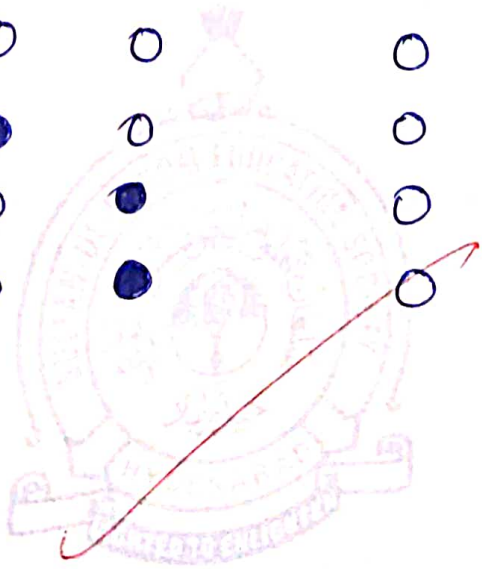


Roll No. : ..... Lecturer's Signature : .....

Expt. No. ..... Page No. ..... Date : .....

OUTPUT:-

Junction 4		Junction 3		Junction 2		Junction 1	
G	R	G	R	G	R	G	R
$P_{2.7}$	$P_{2.6}$	$P_{2.5}$	$P_{2.4}$	$P_{2.3}$	$P_{2.2}$	$P_{2.1}$	$P_{2.0}$
O	●	O	●	O	●	O	O
O	●	O	●	O	●	●	O
O	●	O	●	O	O	O	O
O	●	O	●	●	O	O	●
O	●	O	O	O	●	O	●
O	●	●	O	O	●	O	●
O	O	O	●	O	●	O	●
●	O	O	●	O	●	O	O



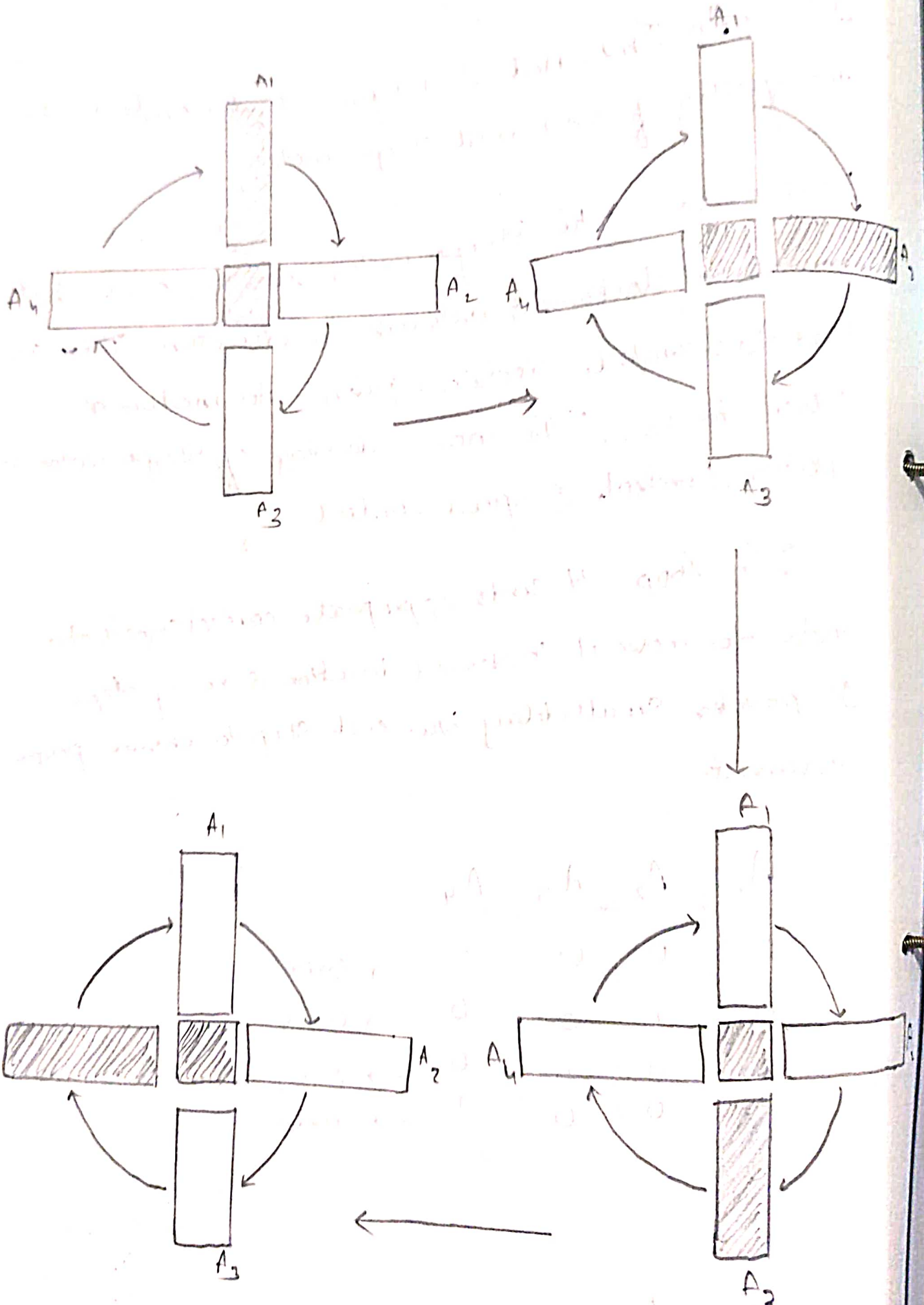
Expt. No. 08 Page No. Date :

AIM: An Embedded C-program to demonstrate the interfacing of 8051 with Stepper motor.

DESCRIPTION: The Stepper motor has 4 phases & it rotates in clockwise & anticlockwise direction. These are used to translate electrical pulses into mechanical movements. The main advantage of Stepper motor is position control & speed control.

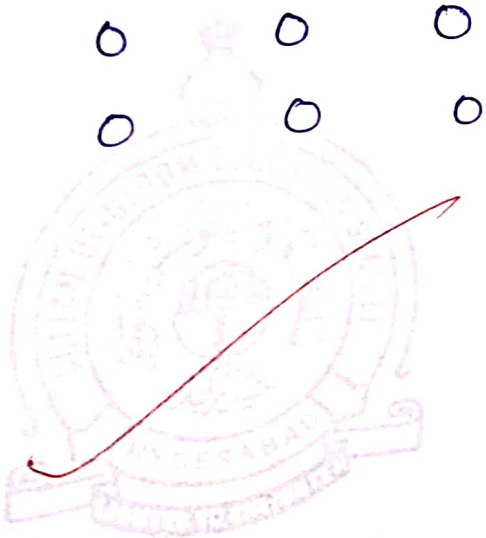
In loop, it sends appropriate control signals to motor to move it in desired direction & no. of steps. It provides small delay b/w each step to ensure proper movement.

A_1	A_2	A_3	A_4	
1	0	0	0	→ 0x0C
0	1	0	0	→ 0x04
0	0	1	0	→ 0x02
0	0	0	1	→ 0x01


A1
A2
A3
A4

OUTPUT:-

$P_{2.7}$	$P_{2.6}$	$P_{2.5}$	$P_{2.4}$	$P_{2.3}$	$P_{2.2}$	$P_{2.1}$	$P_{2.0}$
0	0	0	0	0	0	0	●
0	0	0	0	0	0	●	0
0	0	0	0	0	●	0	0
0	0	0	0	●	0	0	0
0	0	0	0	●	0	0	0
0	0	0	0	0	●	0	0
0	0	0	0	0	0	●	0
0	0	0	0	0	0	0	●



Expt. No. 09 Page No. Date :

AIM: An Embedded C program to demonstrate interfacing of LCD with 8051 microcontroller

DESCRIPTION: The program initializes necessary ports i.e. P_{3-5}, P_{3-4} & P_0 & selects the appropriate display options. It define functions to send commands & data to LCD such as initializing display, setting cursor position & writing characters.

In main loop, it calls these functions to display desired content on LCD such as text messages, numbers.

The LCD with which we are working have 14 pins, & datapins ($D_0 - D_7$), EN - Pin ($P_{3.5}$), Read/Write pin R/W = 0 - writes the data, two power supply & on R/W = 1 - reads the data ground pin.

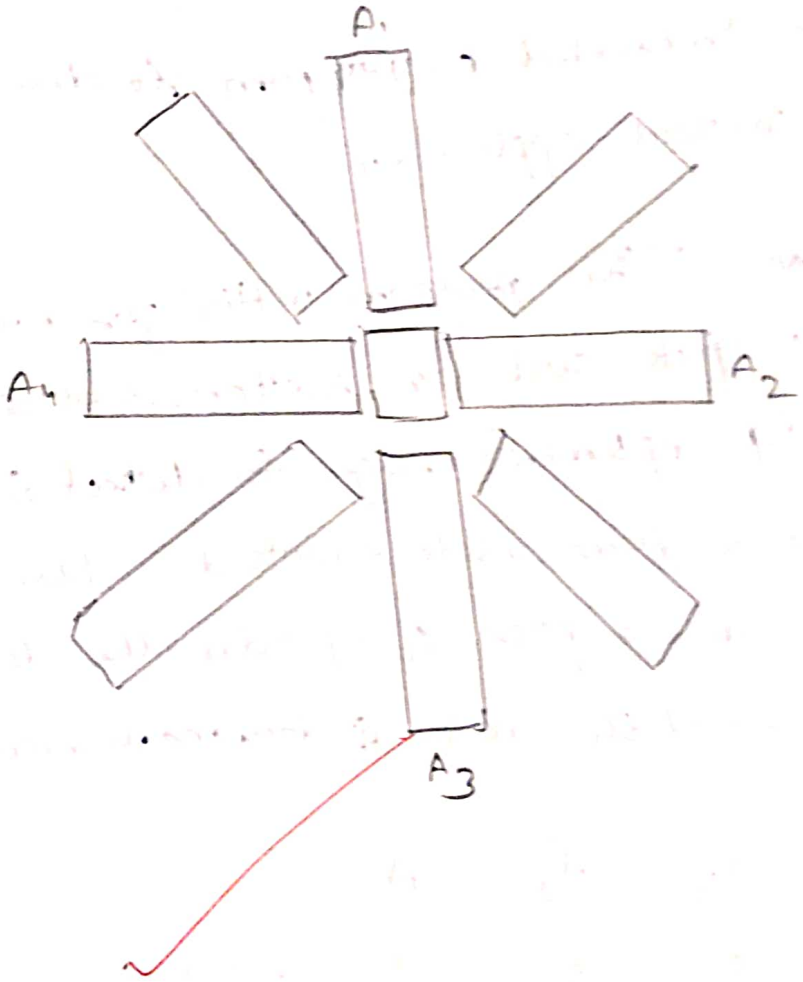
Command	description
01	clear display screen
06	increment sensors
0F	display on-cursor on
80	force cursor to begin 1st line
C0	force cursor to begin 2nd line
38	Two lines & 5x7 matrix.

Expt. No. 10 Page No. Date :

AIM:- An Embedded c-program to develop an elevator control application

DESCRIPTION:- The program initializes necessary inputs & outputs, such as buttons, sensors & motor control. It implements logic to detect & respond to button presses from inside & outside. Using sensors, it tracks current floor & updates the display. The program control the motor & door mechanisms

A_4	A_3	A_2	A_1	
0	0	0	1	- 1
0	0	1	0	- 2
0	1	0	0	- 4
1	0	0	0	- 8
0	0	1	1	- 3
0	1	1	0	- 6
1	1	0	0	- C
1	0	0	1	- 9


A1
A4
A2
A3

```
            }
        }
}
void delay(unsigned int d)
{
        int i,j;
        for(i=0;i<d;i++)
        for(j=0;j<100;j++);
}
```

OUTPUT:

An Elevator control application is demonstrated The floor numbers are displayed on 7 segment & elevator is operated using switches. The Stepper motor rotates in respected direction i.e anti-clockwise upwards & vice versa.

Expt. No. 11D Page No. ..... Date : .....

AIM: An Embedded c program to demonstrate interfacing of 8051 working for external interrupts

DESCRIPTION: An interrupt is an internal/external event that suspends the open microcontroller to inform that a device needs its service. The following mentioned registers are needed to be initialized

IE - Interrupt Enable (Asks which interrupt we want to work with)

7	6	5	4	3	2	1	0
EP	-	ET_2	ES	ET_1	EX_1	ET_0	EX_0

EA - Enable Access (has to be 1)

ET_2, ET_1, ET_0 - Enable Timer

ES - Enable Serial Communication

EX_1, EX_2 - Enable External Interrupt.

1 0 0 0 0 1 0 1 - 0x85

IP - Interrupt priority (provides priority of interrupt which we enable)

7	6	5	4	3	2	1	0
-	-	PT_2	PS	PT_1	PX_1	PT_0	PX_0

PS - Serial port interrupt priority bit

PT_2, PT_1, PT_0 - Timer 2, Timer 1 & Timer 0 interrupt priority bit.

P_{X0}, P_{X1} - External Interrupt priority bit

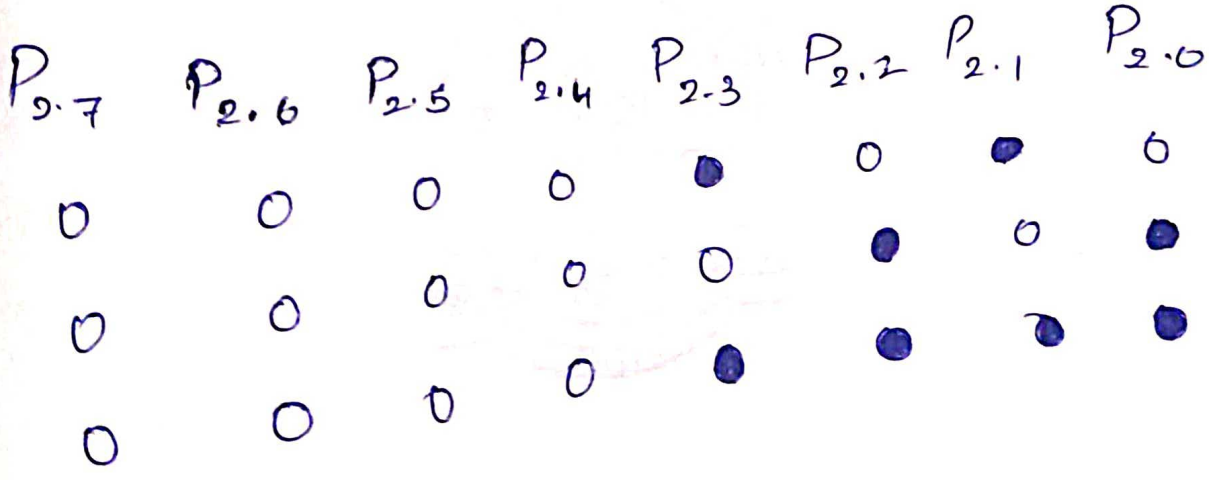
Teacher's Signature : .....

CODE:

```
//external_interrupt.c
#include<REGX51.h>
#define LED1 P2_0
#define LED2 P2_1

void isr0(void) interrupt 0
{
     LED1=~LED1;
}
void isr1(void) interrupt 2
{
     LED2 = ~LED2;
}
void main()
{
     IE=0X85;
     IP=0X01;
     P2=0X00;
     while(1);
}
```

OUTPUT:



Expt. No. 12 Page No. Date :

AIM: An Embedded c program to demonstrate interfacing of Keypad with 8051

DESCRIPTION: The program initializes the necessary ports & pins on 8051 to establish communication with keypad. It defines mapping (or) lookup table for keypad's keys, associating each key with unique code.

In the loop, it scans keypad by sequentially activating each row & checking columns for a pressed key. Upon detecting key press, it reads the corresponding code from lookup table & performs desired action (or) stores the input. By continuous repeating loop, the program enables various application such as password input & data entry.

	col1 ↓ $P_{1.4}$	col2 ↓ $P_{1.5}$	col3 ↓ $P_{1.6}$	col4 ↓ $P_{1.7}$
Row1 → $P_{1.0}$	1	2	3	↑
Row2 → $P_{1.1}$	4	5	6	↓
Row3 → $P_{1.2}$	7	8	9	Mode
Row4 → $P_{1.3}$	*	0	#	↵

Keypad is Connected to port1

row1 = row2 = row3 = row4 = 1

col1 = col2 = col3 = col4 = 0

For each row → Change column value to 1 & return corresponding key row

All the rows are connected to high voltage Vcc & all columns are connected to ground.

Steps for Execution: Create 5 files in project

- Keypad.c
- LCD.c
- LCD.H
- Delay.c
- Delay.H

.C – contains definition
.H – contains declaration

All these 5 files must be added to the source to complete the execution.