Experiment No.: 1

Interfacing LED bank to 8051 microcontroller using timer with interrupt.

Roll No.: _____ Batch: _____
Date of Performance: _____
Date of Assessment: _____

Particulars	Marks
Regularity (05)	
Practical Conduction (10)	
Program Output (05)	
Understanding (Viva) (05)	
Total (25)	
Signature of Course Teacher	

Experiment: 1

AIM: Interfacing LED bank to 8051 microcontroller using timer with interrupt.

PREREQUISITE:

Before starting this lab, students should be familiar with:

- Basics of 8051 microcontroller architecture and pin configuration.
- LED interfacing techniques with microcontrollers.
- Timer and interrupt concepts in embedded systems.
- Keil µVision and Nuvoton ISP-ICP Utility tools

OBJECTIVE:

- By the end of this experiment, students will be able to:
- Understand the working of 8051 timers and interrupts.
- Configure timer interrupts for generating periodic signals.
- Interface LED bank to 8051 and control it using timer interrupts.
- Develop and simulate programs in Keil software.

HARDWARE REQUIREMENTS:

- 8051 microcontroller Development Board (Arise51 V1.1)
- USB Cable (To program and power the board)
- Power Adapter (optional)
- FRC Cables for connecting peripherals
- Pin to Pin Connectors for connecting peripherals
- Personal Computer (PC) or Laptop

SOFTWARE REQUIREMENTS:

- Keil µVision Software
- Nuvoton ISP-ICP Utility Application
- CP2102 Drivers for USB to Serial Converter

THEORY:

1) Introduction:

An LED bank is a set of multiple LEDs connected to the output port of a microcontroller. By programming the microcontroller, LEDs can be turned ON or OFF in specific patterns or sequences.

The 8051 microcontroller includes two timers (Timer 0 and Timer 1) that can be used to generate accurate time delays or periodic interrupts. Interrupts are signals that temporarily stop the main program to execute a specific subroutine (ISR - Interrupt Service Routine), and then return to the main program.

2) Timer Operation in 8051:

Timers in 8051 can operate in different modes (Mode 0, 1, 2, and 3). In Mode 1, the timer works as a 16-bit timer. The timer counts from 0000H to FFFFH, and when it overflows, it sets the TF flag, which can trigger an interrupt if enabled.

3) Interrupt System:

8051 has five interrupt sources:

- External Interrupt 0 (INT0)
- Timer 0 Overflow Interrupt
- External Interrupt 1 (INT1)
- Timer 1 Overflow Interrupt
- Serial Communication Interrupt

When an interrupt occurs, the processor executes the corresponding ISR.

4) LED Interfacing:

LEDs are connected to one of the ports (commonly P1 or P2) of 8051 via current-limiting resistors. A logic HIGH or LOW signal turns each LED ON or OFF depending on the configuration.

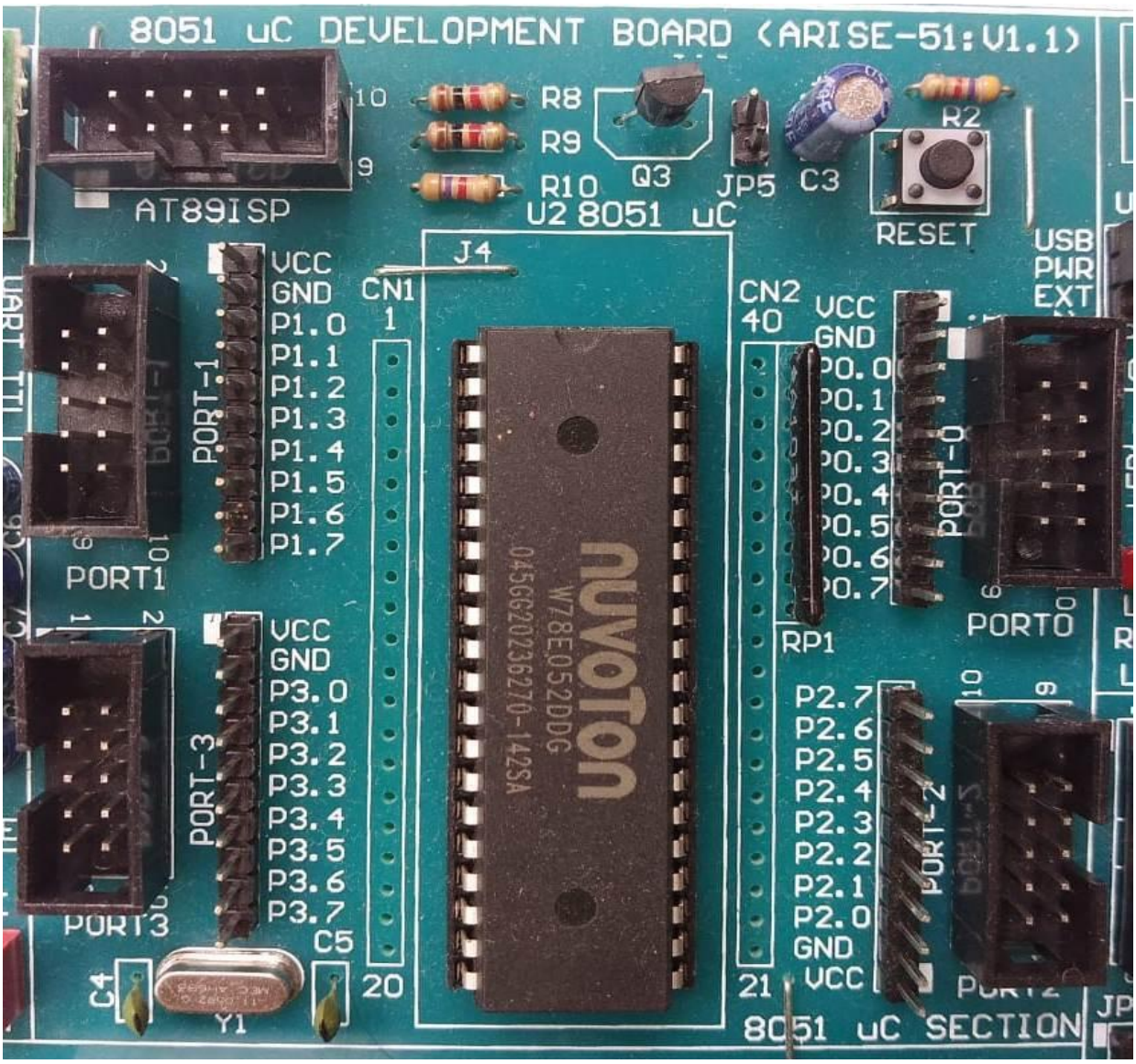
5) Working Principle:

In this experiment, Timer 0 is configured in Mode 1 to generate interrupts at regular intervals. Each time an interrupt occurs, the ISR toggles the output pattern on the LED bank, resulting in blinking or shifting patterns.

6)On-board peripherals and Connection Details

1. Microcontroller Section

This section consists on minimum circuit of 8051 microcontroller i.e clock circuit, reset circuit, ISP Connector and Port Pins. All Ports pin available on 10 FRC and Berg strip connector as shown in figure below. FRC Connector AT89ISP is used for programming the AT89 series microcontroller from Atmel. Serial Port ISP supported by 8051 microcontrollers from NXP and Nuvoton. Therefore we are using USB to Serial converter for downloading the Hex File into microcontroller.
Reset button is used to reset the microcontroller. Clock circuit consists of 11.0592MHz crystal. Therefore while calculation of baud rate, timer delay the FSOC is taken as 11.0592MHz only.



2. Power Section

The board can powered using 9-12V DC adapter or from USB using proper jumper setting. The JP1 connector is to select the proper power source. Make sure short link in correct position.

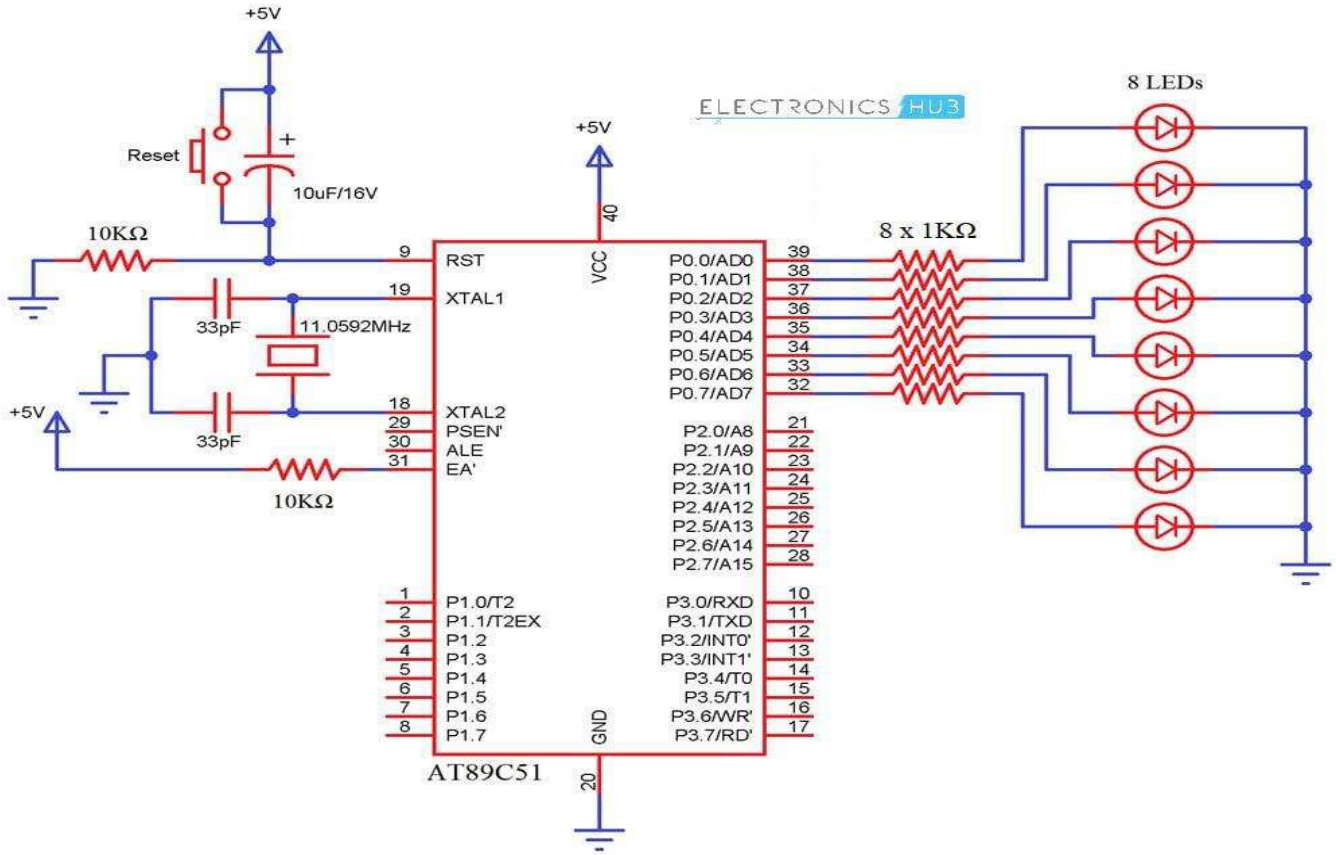
3. LED Section

This section consists of 8 LED's for digital output. LED's are connected in Common Cathode Mode. The LED's can be interface to any pin/ pins of microcontroller using pin to pin connector or 10-FRC Connector.

4. USB to UART Section

This section consists of USB to UART converter. By default this connected to on-chip UART of microcontroller. The connecter 'UART_TTL' is used to access the TTL signals. There are two options for USB To UART either CP2102 or CH340. Accordingly user need to install the drivers.

LED INTERFACING DIAGRAM:



PROCEDURE:

- Open Keil µVision and create a new project for 8051.
- Add a new C source file and write the LED interfacing code.
- Compile the code and check for errors.
- Generate the HEX file.
- Open Nuvoton ISP-ICP Utility and load the HEX file.
- Flash the program to the 8051 board.
- Observe the LEDs flashing at a fixed rate.

CODE:

Attach separate sheet for code

CONCLUSION:

_____

_____

_____

_____

Experiment No.: 2

Interfacing Seven Segment Display to 8051 microcontroller.

Roll No.: _____ Batch: _____

Date of Performance: _____

Date of Assessment: _____

Particulars	Marks
Regularity (05)	
Practical Conduction (10)	
Program Output (05)	
Understanding (Viva) (05)	
Total (25)	
Signature of Course Teacher	

Experiment: 2

AIM: Interfacing Seven Segment Display to 8051 microcontroller.

PREREQUISITE:

Before starting this lab, students should be familiar with:

- Basics of 8051 microcontroller architecture and pin configuration.
- Concept of seven segment display (Common Anode / Common Cathode).
- Working of GPIO (Input/Output) pins in 8051.
- Keil µVision and Nuvoton ISP-ICP Utility tools

OBJECTIVE:

By the end of this experiment, students will be able to:

- Understand the working of seven segment display.
- Interface a seven segment display to 8051 microcontroller.
- Display digits (0–9) on the display through programming.
- Simulate and implement the interfacing program using Keil software.

HARDWARE REQUIREMENTS:

- 8051 microcontroller Development Board (Arise51 V1.1)
- USB Cable (To program and power the board)
- Power Adapter (optional)
- FRC Cables for connecting peripherals
- Pin to Pin Connectors for connecting peripherals
- Personal Computer (PC) or Laptop

SOFTWARE REQUIREMENTS:

- Keil µVision Software
- Nuvoton ISP-ICP Utility Application
- CP2102 Drivers for USB to Serial Converter

THEORY:

1) Introduction:

A Seven Segment Display (SSD) is an electronic display device used to display decimal numbers (0–9) and some alphabets. It consists of seven LEDs arranged in a specific pattern to form digits. Each LED is labeled as 'a' to 'g', and by turning ON/OFF specific segments, different numbers can be displayed.

2) Types of Seven Segment Displays:

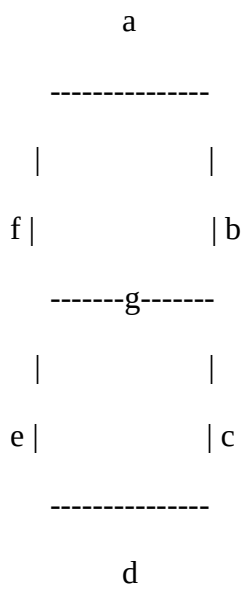
• Common Cathode (CC): All cathode terminals of LEDs are connected together to ground.
• Common Anode (CA): All anode terminals of LEDs are connected together to VCC.

In this experiment, a **Common Anode** seven segment display is used.

3) Segment Configuration:

Each segment (a–g) is connected to one port pin of the 8051 microcontroller. To display a digit, the corresponding binary code is sent to the port.

Basic Concept: 7-Segment Layout



Each digit (0–9) is formed by turning ON/OFF certain LEDs labelled **a–g** (plus an optional dot 'dp').

Each segment corresponds to one bit in an **8-bit pattern** (7 for segments + 1 for the dot).

Bit Position Assignment

For a **common anode 7-segment**, each segment lights up when its pin is driven **LOW (0)**.

Typical bit assignment (MSB → LSB):

Bit	Segment	Function
D7	a	Top segment
D6	b	Upper right
D5	c	Lower right
D4	d	Bottom
D3	e	Lower left
D2	f	Upper left
D1	g	Middle
D0	dp	Decimal point (optional)

How each code is derived (quick recipe)

1. Choose which segments (a–g, dp) must be ON for the digit.
2. For each segment that is ON, set its corresponding bit to **0** (because common anode).
3. For segments that are OFF, set the bit to **1**.
4. Write the 8-bit pattern (D7→D0) and convert to hex.

Example — digit 2: segments a, b, d, e, g are ON.

- a=0, b=0, c=1, d=0, e=0, f=1, g=0, dp=1 → 10100100 → 0xA4.

Compact visual (bit → segment)

D7	D6	D5	D4	D3	D2	D1	D0
a	b	c	d	e	f	g	dp
1	0	1	0	0	1	0	0

Digit 2: 10100100 → 0xA4.

Digit	Hex Code	Binary	Segments ON
0	0xC0	11000000	a b c d e f
1	0xF9	11111001	b c
2	0xA4	10100100	a b d e g
3	0xB0	10110000	a b c d g
4	0x99	10011001	b c f g
5	0x92	10010010	a c d f g
6	0x82	10000010	a c d e f g
7	0xF8	11111000	a b c
8	0x80	10000000	all segments
9	0x90	10010000	a b c d f g
A	0x88	1000 1000	a b c e f g
B	0x83	1000 0011	c d e f g
C	0xC6	1100 0110	a d e f
D	0xA1	1010 0001	b c d e g
E	0x86	1000 0110	a d e f g
F	0x8E	1000 1110	a e f g

4) Interfacing with 8051:

The seven segment display is interfaced to Port 2 of the 8051 microcontroller. Each pin of the port controls one segment of the display. By sending appropriate hex codes to the port, the required digit is displayed.

5) Working Principle:

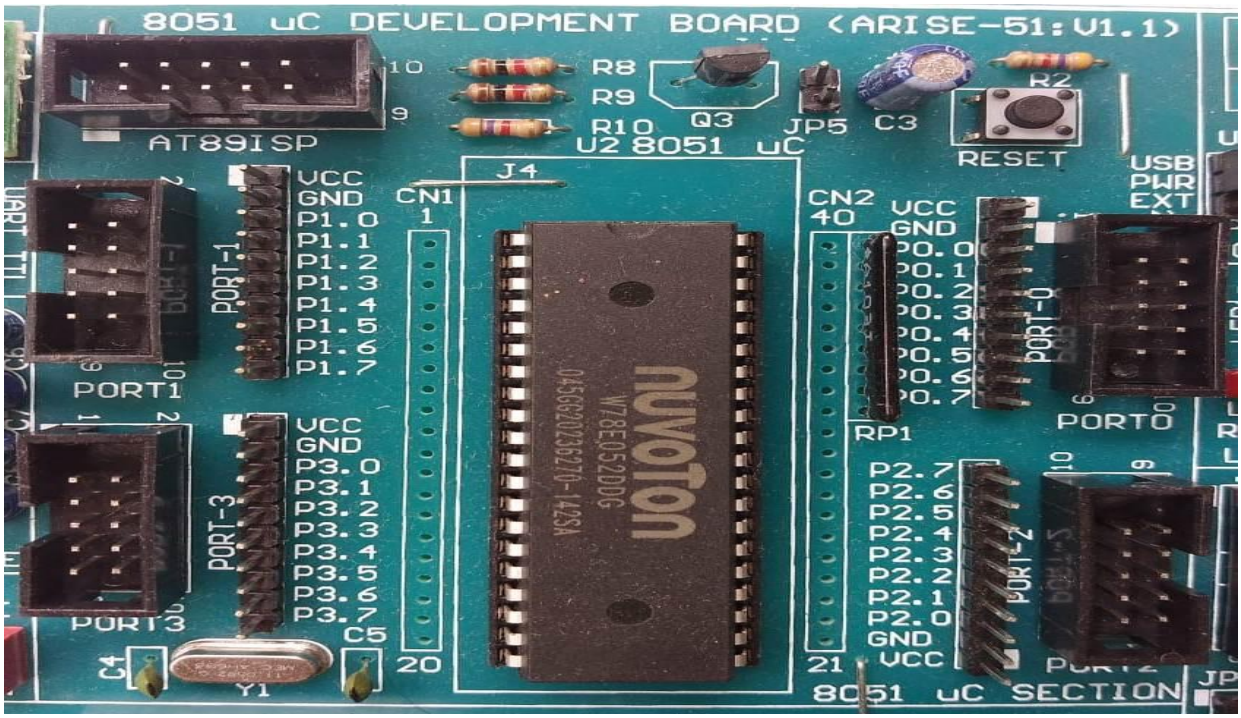
In this experiment, the microcontroller sequentially displays digits from 0 to 9 with a delay between each number. This is achieved by repeatedly writing segment codes to the output port.

6) On-board peripherals and Connection Details

1. Microcontroller Section

This section consists on minimum circuit of 8051 microcontroller i.e clock circuit, reset circuit, ISP Connector and Port Pins. All Ports pin available on 10 FRC and Berg strip connector as shown in figure below. FRC Connector AT89ISP is used for programming the AT89 series microcontroller from Atmel. Serial Port ISP supported by 8051 microcontrollers from NXP and Nuvoton. Therefore we are using USB to Serial converter for downloading the Hex File into microcontroller.

Reset button is used to reset the microcontroller. Clock circuit consists of 11.0592MHz crystal. Therefore while calculation of baud rate, timer delay the FSOC is taken as 11.0592MHz only.

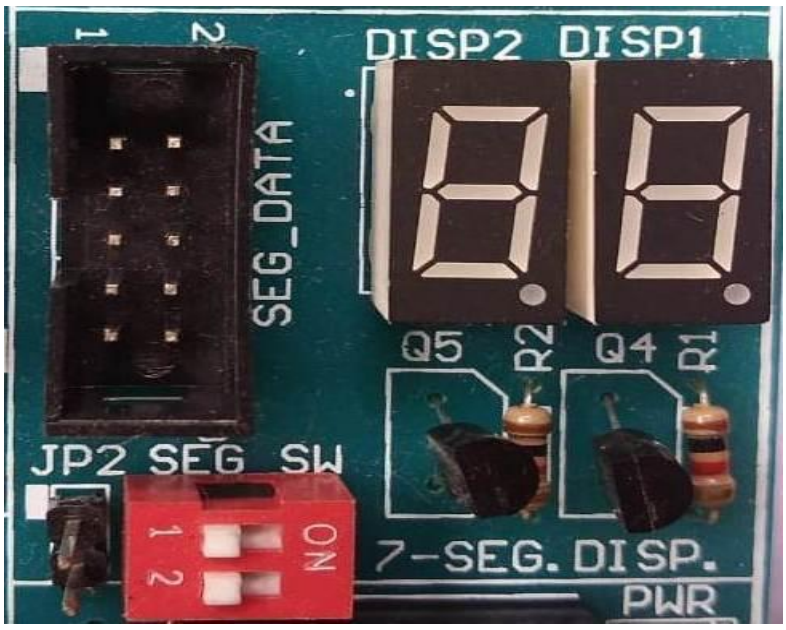


2. Power Section

The board can powered using 9-12V DC adapter or from USB using proper jumper setting. The JP1 connector is to select the proper power source. Make sure short link in correct position.

3. Multiplexed 7-Segment Display

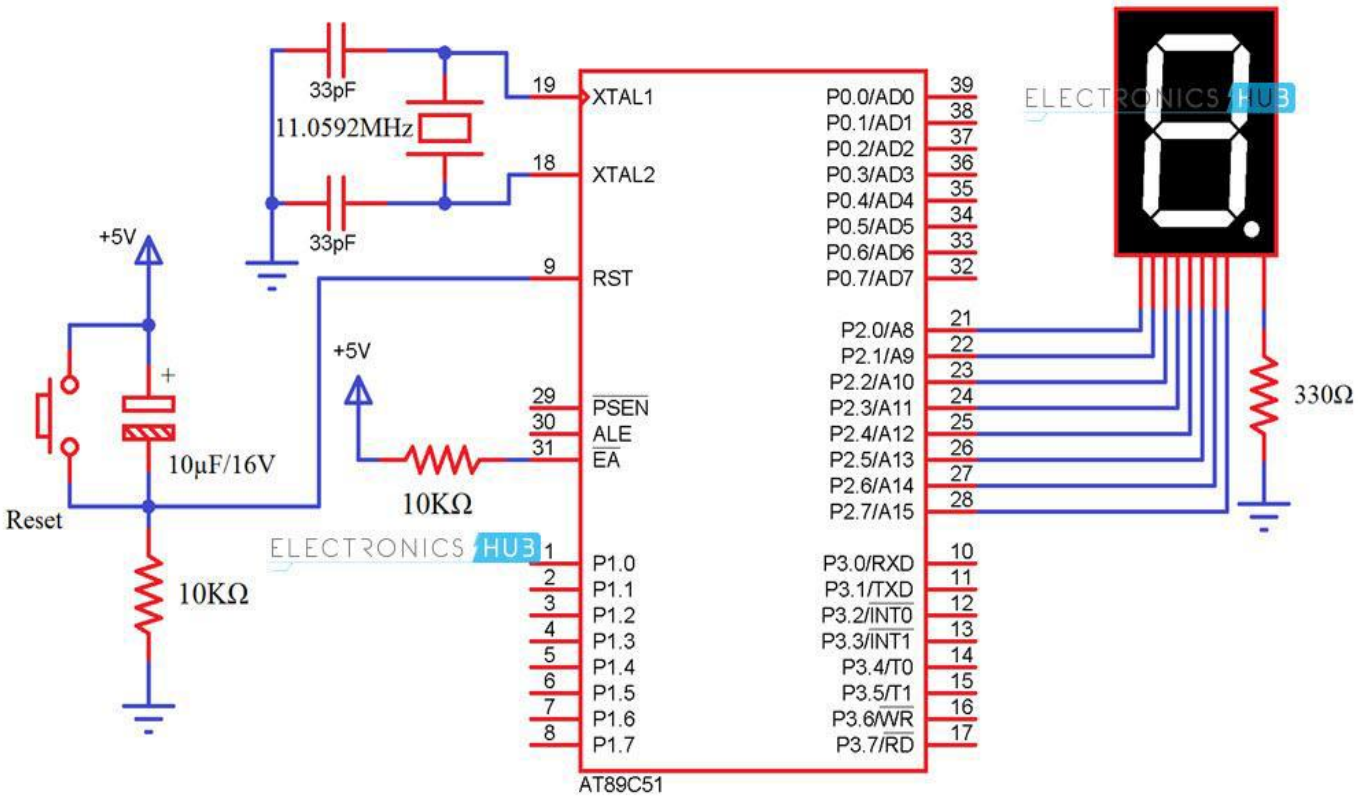
This section consists for two 7-segment display interfaced in multiplexed fashion. The 7-segment displays are in common anode fashion.



4. USB to UART Section

This section consists of USB to UART converter. By default this connected to on-chip UART of microcontroller. The connecter 'UART_TTL' is used to access the TTL signals. There are two options for USB To UART either CP2102 or CH340. Accordingly user need to install the drivers.

7) SEVEN SEGMENT INTERFACING DIAGRAM:



PROCEDURE:

- Open Keil µVision and create a new project for 8051.
- Add a new C source file and write the seven segment display interfacing code.
- Define the segment codes for digits 0–9.
- Connect the seven segment display to Port 2 of 8051.
- Compile the code and check for errors.
- Generate the HEX file.
- Open Nuvoton ISP-ICP Utility and load the HEX file.
- Flash the program to the 8051 board.
- Observe the digits 0–9 displayed sequentially on the seven segment display.

SAMPLE CODE :

Attach separate sheet for code

CONCLUSION:

Experiment No.: 3

Interfacing of LCD to 8051 microcontroller.

Roll No.: _____ Batch: _____

Date of Performance: _____

Date of Assessment: _____

Particulars	Marks
Regularity (05)	
Practical Conduction (10)	
Program Output (05)	
Understanding (Viva) (05)	
Total (25)	
Signature of Course Teacher	

Experiment: 3

AIM: Interfacing of LCD to 8051 microcontroller.

PREREQUISITE:

Before starting this lab, students should be familiar with:

- Basics of 8051 microcontroller architecture and I/O Ports.
- Working and pin configuration of 16x2 LCD module.
- Basic C programming for microcontrollers.
- Keil µVision and Nuvoton ISP-ICP Utility tools

OBJECTIVE:

By the end of this experiment, students will be able to:

- Understand the internal structure and operation of a 16x2 LCD.
- Interface LCD with 8051 microcontroller.
- Display alphanumeric characters and messages on LCD.
- Simulate and implement LCD interfacing program using Keil software.

HARDWARE REQUIREMENTS:

- 8051 microcontroller Development Board (Arise51 V1.1)
- 16x2 LCD Display Module
- 10k Potentiometer (for contrast adjustment)
- USB Cable (To program and power the board)
- Power Adapter (optional)
- FRC Cables for connecting peripherals
- Pin to Pin Connectors for connecting peripherals
- Personal Computer (PC) or Laptop

SOFTWARE REQUIREMENTS:

- Keil µVision Software
- Nuvoton ISP-ICP Utility Application
- CP2102 Drivers for USB to Serial Converter

THEORY:

1) Introduction:
A 16x2 Liquid Crystal Display (LCD) is one of the most commonly used display devices in embedded systems. It can display 16 characters per line on two lines. LCD modules are widely used because they are easy to program, inexpensive, and consume low power.

2) LCD Pin Configuration:
The standard 16x2 LCD has 16 pins:
1. VSS – Ground
2. VCC – +5V Supply
3. VEE – Contrast Adjustment (connected to potentiometer)
4. RS – Register Select (0: Command, 1: Data)
5. RW – Read/Write (0: Write, 1: Read)
6. EN – Enable Signal (High-to-Low pulse)
7–14. D0–D7 – Data Lines
15. LED+ – Backlight Anode
16. LED– – Backlight Cathode

3) Interfacing LCD with 8051:
The LCD is connected to Port 2 for data (D0–D7) and Port 3 pins (P3.0, P3.1, P3.2) for control signals (RS, RW, EN). Commands and data are sent by setting RS, RW, and pulsing EN to latch the data.

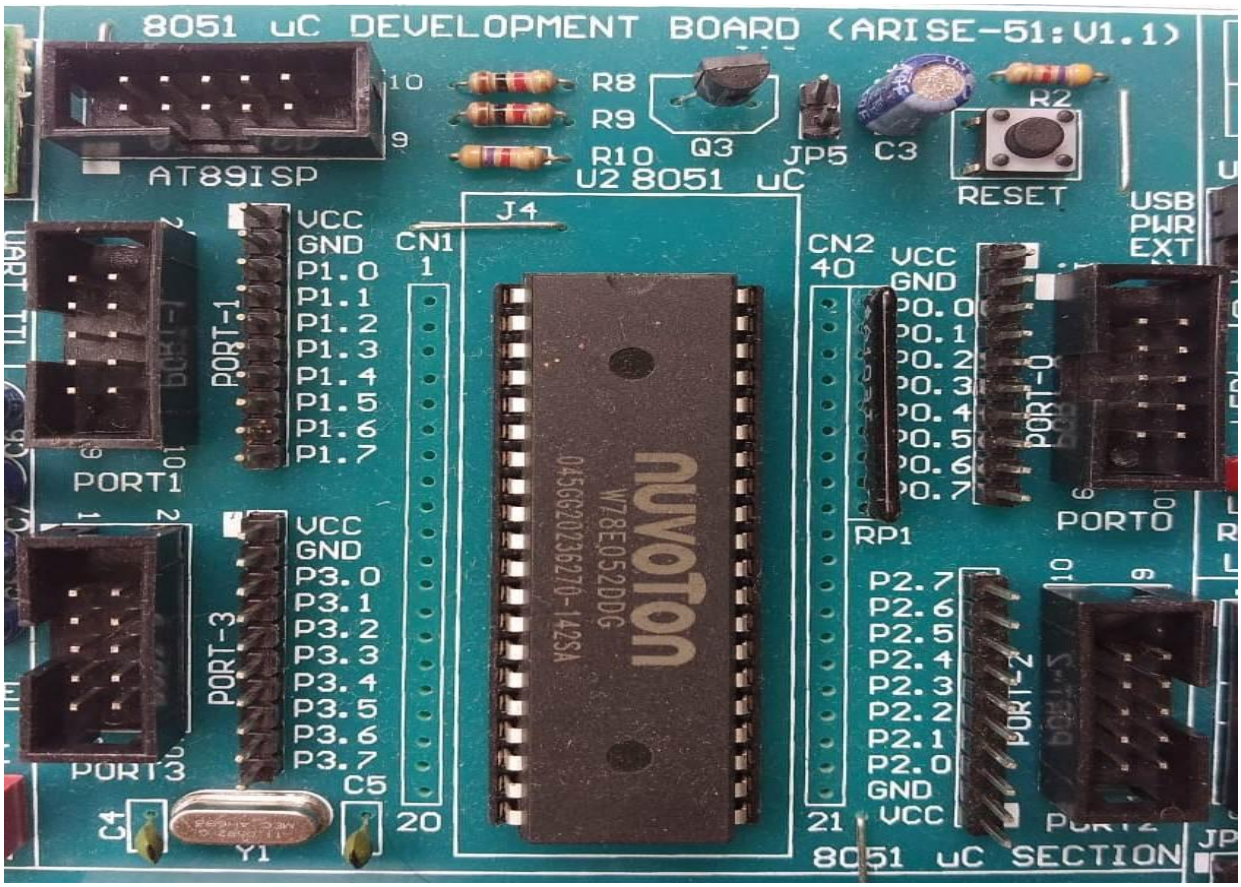
4) Common LCD Commands:
0x38 – Initialize LCD in 8-bit, 2-line, 5x7 format
0x0C – Display ON, Cursor OFF
0x01 – Clear Display
0x06 – Increment Cursor
0x80 – Move cursor to first line
0xC0 – Move cursor to second line

5) Working Principle:
When the microcontroller sends a command, the RS pin is kept low. When it sends data, the RS pin is high. RW is low for writing. After each command or data, the EN pin is toggled to latch the value.

6) On-board peripherals and Connection Details

1. Microcontroller Section

This section consists on minimum circuit of 8051 microcontroller i.e clock circuit, reset circuit, ISP Connector and Port Pins. All Ports pin available on 10 FRC and Berg strip connector as shown in figure below. FRC Connector AT89ISP is used for programming the AT89 series microcontroller from Atmel. Serial Port ISP supported by 8051 microcontrollers from NXP and Nuvoton. Therefore we are using USB to Serial converter for downloading the Hex File into microcontroller.
Reset button is used to reset the microcontroller. Clock circuit consists of 11.0592MHz crystal. Therefore while calculation of baud rate, timer delay the FSOC is taken as 11.0592MHz only.

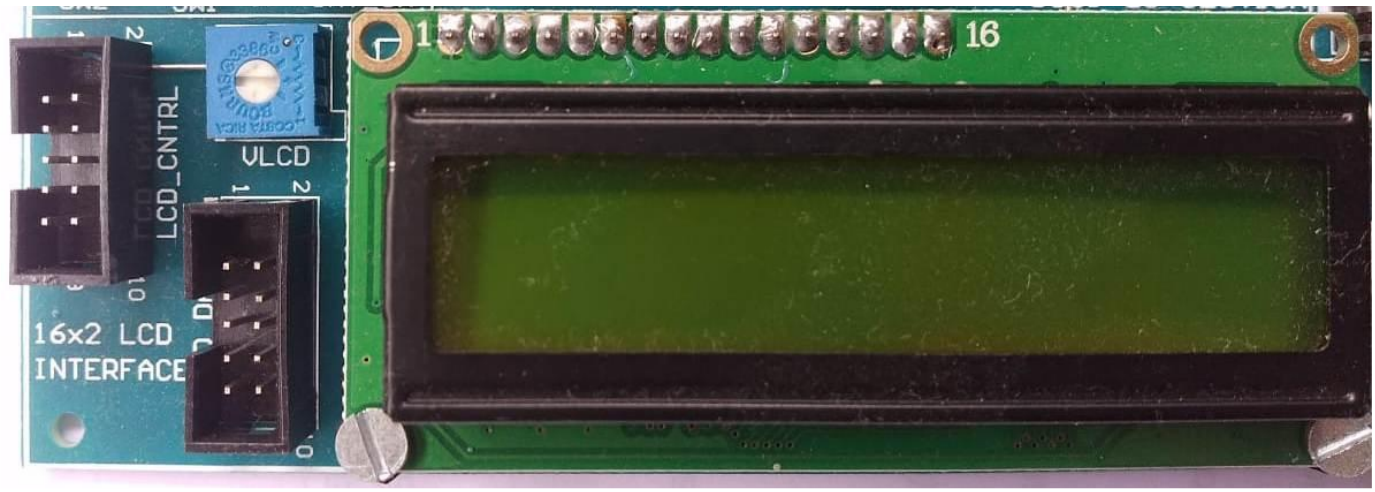


2. Power Section

The board can powered using 9-12V DC adapter or from USB using proper jumper setting. The JP1 connector is to select the proper power source. Make sure short link in correct position.

3. LCD Section:

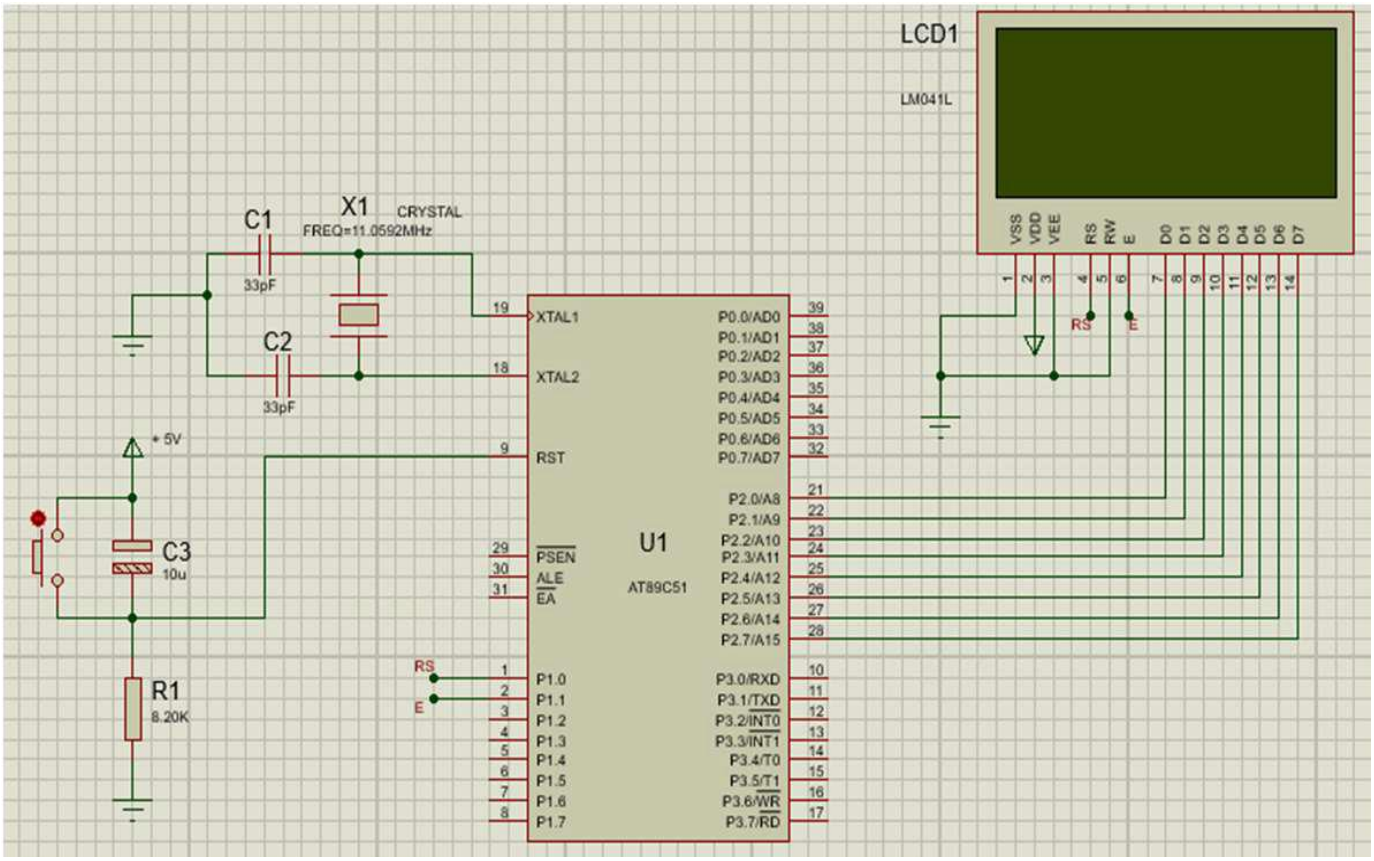
This section consists for 16x2 LCD interfacing provision. There are two 10 pin FRC Connector; one is for LCD Control Signal and other one is for LCD Data Signal.



4. USB to UART Section

This section consists of USB to UART converter. By default this connected to on-chip UART of microcontroller. The connecter 'UART_TTL' is used to access the TTL signals. There are two options for USB To UART either CP2102 or CH340. Accordingly user need to install the drivers.

7) LCD INTERFACING DIAGRAM:



PROCEDURE:

- Open Keil µVision and create a new project for 8051.
- Add a new C source file for LCD interfacing.
- Write the program to initialize and display messages on the LCD.
- Connect the LCD data and control pins as per schematic.
- Compile the code and check for errors.
- Generate the HEX file.
- Open Nuvoton ISP-ICP Utility and load the HEX file.
- Flash the program to the 8051 board.
- Power up the system and observe the message on LCD.

SAMPLE CODE :

Attach separate sheet for code

CONCLUSION:

Experiment No.: 4

Interfacing stepper motor to 8051 microcontroller

Roll No.: _____ Batch: _____

Date of Performance: _____

Date of Assessment: _____

Particulars	Marks
Regularity (05)	
Practical Conduction (10)	
Program Output (05)	
Understanding (Viva) (05)	
Total (25)	
Signature of Course Teacher	

Experiment: 4

AIM: Interfacing stepper motor to 8051 microcontroller.

PREREQUISITE:

Before starting this lab, students should be familiar with:

- Basics of 8051 microcontroller architecture and I/O Ports.
- Stepper motor working and control logic.
- Operation of motor driver ICs like ULN2003 or L293D.
- Keil µVision and Nuvoton ISP-ICP Utility tools

OBJECTIVE:

By the end of this experiment, students will be able to:

- Understand the working of stepper motors and their control sequences.
- Interface a stepper motor with the 8051 microcontroller using driver circuits
- Write code to control motor direction and speed.
- Simulate and implement stepper motor control in Keil software.

HARDWARE REQUIREMENTS:

- 8051 microcontroller Development Board (Arise51 V1.1)
- Stepper Motor (Unipolar)
- ULN2003 or L293D Motor Driver IC
- USB Cable (To program and power the board)
- Power Adapter (optional)
- FRC Cables for connecting peripherals
- Pin to Pin Connectors for connecting peripherals
- Personal Computer (PC) or Laptop

SOFTWARE REQUIREMENTS:

- Keil µVision Software
- Nuvoton ISP-ICP Utility Application
- CP2102 Drivers for USB to Serial Converter

THEORY:

1) Introduction:

A stepper motor is an electromechanical device that converts electrical pulses into discrete mechanical steps. The rotation angle of the motor is proportional to the number of input pulses. Stepper motors are used in precise motion control applications such as robotics, printers, and CNC machines.

2) Working Principle:

A stepper motor operates on the principle of electromagnetism. When the stator coils are energized in a specific sequence, the rotor aligns itself with the magnetic field, producing rotation in steps. By changing the sequence of excitation, the motor can rotate clockwise or counterclockwise.

3) Types of Stepper Motors:

• Unipolar Stepper Motor – Each winding has a center tap. It is easier to drive using ULN2003.

• Bipolar Stepper Motor – Requires full H-bridge drivers such as L293D.

4) Interfacing with 8051:

The 8051 microcontroller outputs control signals to the motor driver (ULN2003/L293D), which amplifies the current required for the stepper motor coils. Port 2 of the 8051 is used to send control signals to the driver inputs.

5) Step Sequence for 4-Step Operation:

• Step 1 → 0x09 (1001)

• Step 2 → 0x0C (1100)

• Step 3 → 0x06 (0110)

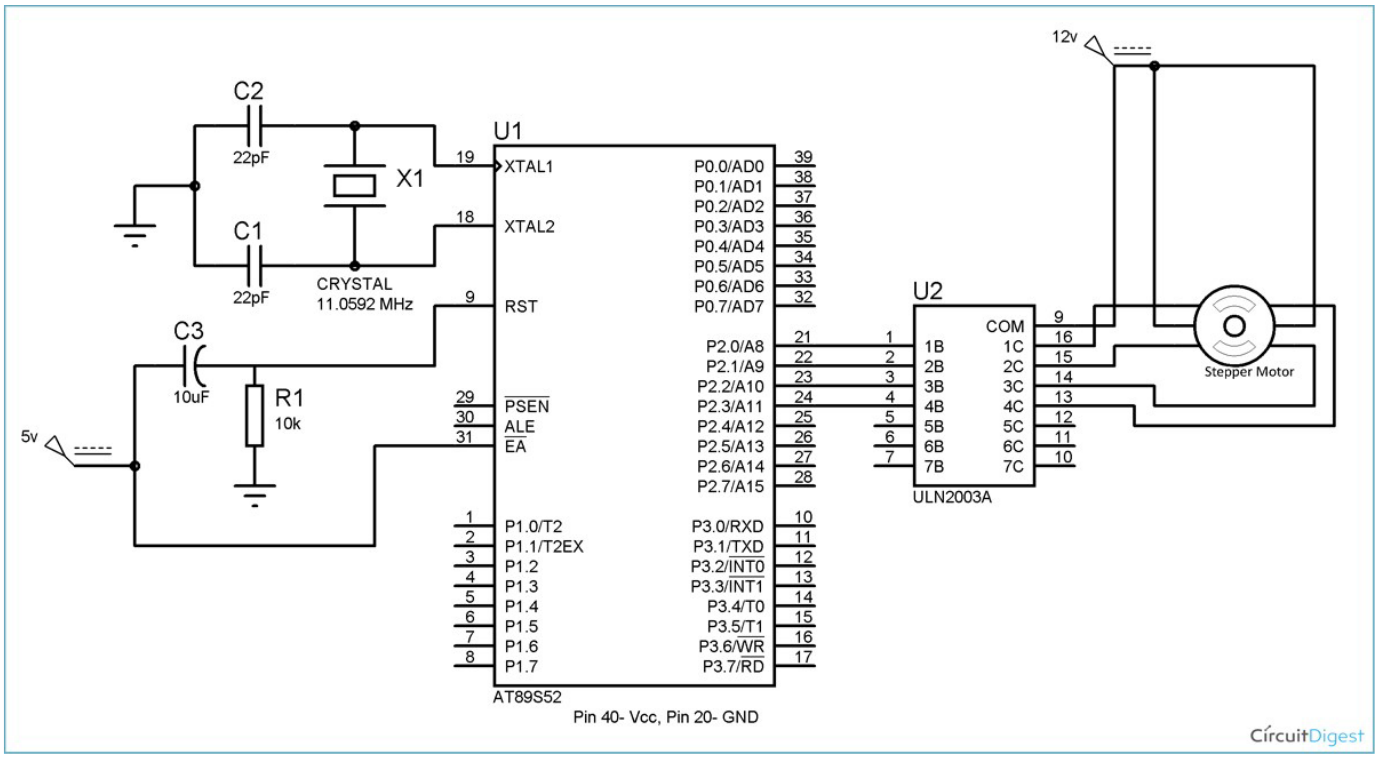
• Step 4 → 0x03 (0011)

To reverse the direction, the sequence is reversed.

6) Working:

The motor driver receives signals from the microcontroller and energizes the stepper coils accordingly. By continuously changing the energizing pattern, the rotor moves step by step, achieving controlled rotation.

7) Stepper Motor Interfacing Diagram:



PROCEDURE:

- Open Keil µVision and create a new project for 8051.
- Add a new C source file and write the stepper motor interfacing code.
- Define the step sequence for clockwise and counterclockwise rotation.
- Connect the stepper motor to Port 2 of 8051 via ULN2003/L293D driver.
- Compile the code and check for errors.
- Generate the HEX file.
- Open Nuvoton ISP-ICP Utility and load the HEX file.
- Flash the program to the 8051 board.
- Power up the system and observe the message on LCD.

SAMPLE CODE :

Attach separate sheet for code

CONCLUSION:

_____

_____

_____

_____