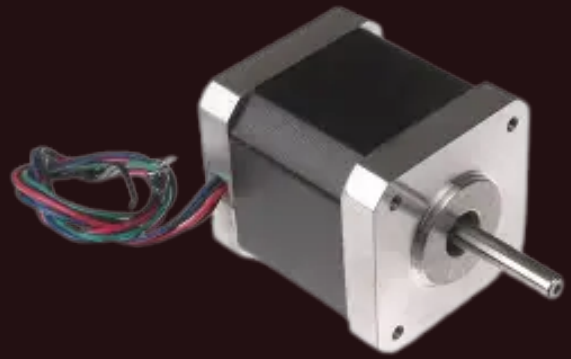
Stepper Motor Interfacing with PIC18F4550 Microcontroller

Hamza Safwan
2022198

Overview of Key Concepts Concepts

- Stepper Motor Basics
- PIC18F4550 Role
- ULN2003A Driver IC
- Serial Communication
- Simulation & Flowchart



Components Used

PIC18F4550 Microcontroller

Handles UART communication and and logic control, sending signals signals to the driver IC.

ULN2003A Driver IC

Amplifies control signals and interfaces with the stepper motor coils.

4-Phase Stepper Motor

Converts electrical pulses into mechanical steps for precise moti on.

What is a Stepper Motor?

Electromechanical Device

Stepper motors convert electrical pulses into discrete mechanical movements, movements, enabling precise control in systems like printers and CNC machines. machines.

Unipolar Configuration

The most common type used here is a 4-phase unipolar motor, with four windings windings and a common terminal, driven in specific step sequences.

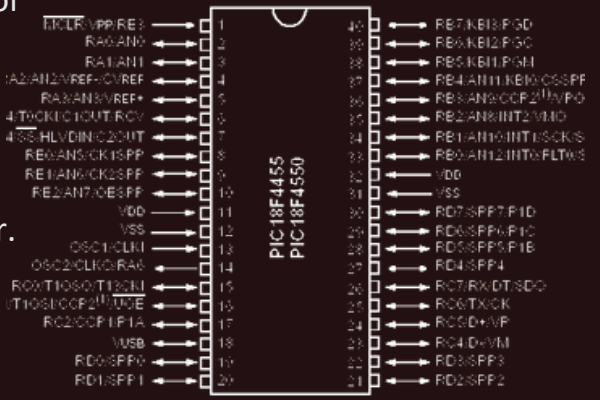
Step Angle Control

Each control pulse rotates the motor a fixed step angle, allowing accurate control of accurate control of position and speed.

PIC18F4550 Microcontroller

Controller and Logic Center

The PIC18F4550 is responsible for sending control signals to the stepper motor stepper motor and receiving commands through UART.



Port Configuration

PORTD pins (RD0–RD3) output 8-step half-step sequences to the motor driver. motor driver. RC7 receives input via UART.

Programming Environment

It is programmed using XC8 compiler in MPLAB X IDE, supporting serial serial communication and real-time motor control.

Half-Step Sequence Logic
L ogic

- 8 steps for fine resolution
- Alternating single/dual-phase
- Precise angular control
- Clockwise and counterclockwise
- Sequence governs smoothness

Clockwise ↓

Step #	Winding A	Winding B	Winding C	Winding D
1	1	0	0	1
2	1	0	0	0
3	1	1	0	0
4	0	1	0	0
5	0	1	1	0
6	0	0	1	0
7	0	0	1	1
8	0	0	0	1

Counter-clockwis ↑

8-Step Half-Step Sequence

Precision Control

Half-step drive energizes coils in alternating single and dual-phase modes, modes, allowing finer angular resolution and smoother rotation.

Directional Control

Steps 1 to 8 produce clockwise motion; the reverse sequence controls counter-clockwise rotation.

Winding Logic

Winding states are controlled to step the rotor in precise increments, as shown in the table for A, B, C, and D windings.

Serial Communication Configuration

UART Setup

The PIC18F4550 is configured for asynchronous serial communication at 9600 communication at 9600 bps, using the built-in UART module.

Baud Rate Calculation

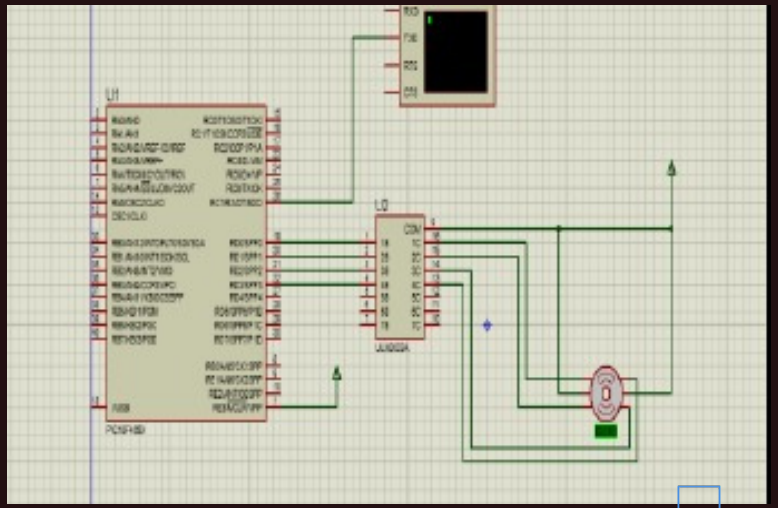
Using the formula SPBRG = (Fosc / (16 ×Baud Rate)) − 1, the SPBRG value is value is calculated as 25 for 4 MHz oscillator.

Reliable Data Transmission

This configuration allows real-time motor control via ASCII commands like 'F', commands like 'F', 'B', and 'S' from a virtual terminal.

Proteus Simulation Highlights

- PIC18F4550 connected to ULN2003A
- Stepper motor driven via PORTD
- UART terminal on RC7 for input
- Real-time command control
- Full logic circuit validation



Proteus Simulation Setup

Virtual Hardware Environment

Proteus simulation includes the PIC18F4550, ULN2003A, stepper motor, and a motor, and a virtual terminal for UART command input.

Component Interconnections

Signals from PORTD control the motor via ULN2003A, while RC7 reads serial input for real-time control.

Functional Validation

The simulation demonstrates precise rotation control and validates the working logic of the embedded system.

Project Flowchart Overview

Initialization Phase

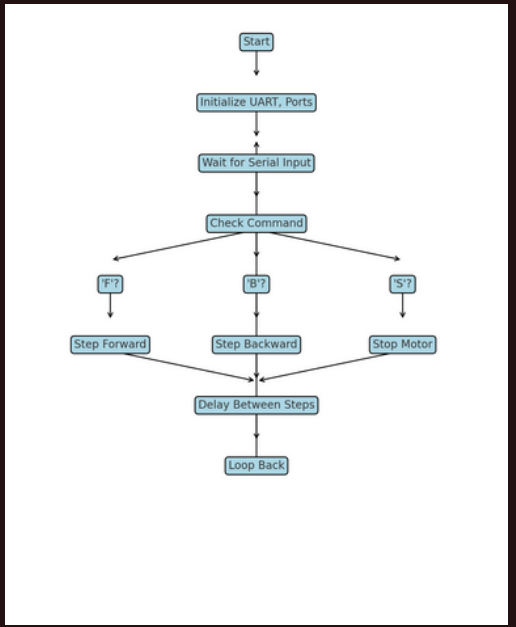
The systembegins by initializing UART and port settings, preparing the the microcontroller to receive commands.

Decision Logic

Incoming characters are evaluated to determine the action—move forward, forward, backward, or stop the motor.

Loop Execution

Basedon thecommand, the system loops through the half-step sequence with sequence with delays to animate movement.



Conclusion & Takeaways

How does serial-controlled stepper motor interfacing with PIC18F4550 enhance control sy stems?

Precision & Safety

This project demonstrates real-time control of a stepper motor using UART and embedded logic. It ensures safe operation via ULN2003A and effective simulation validation through Proteus.