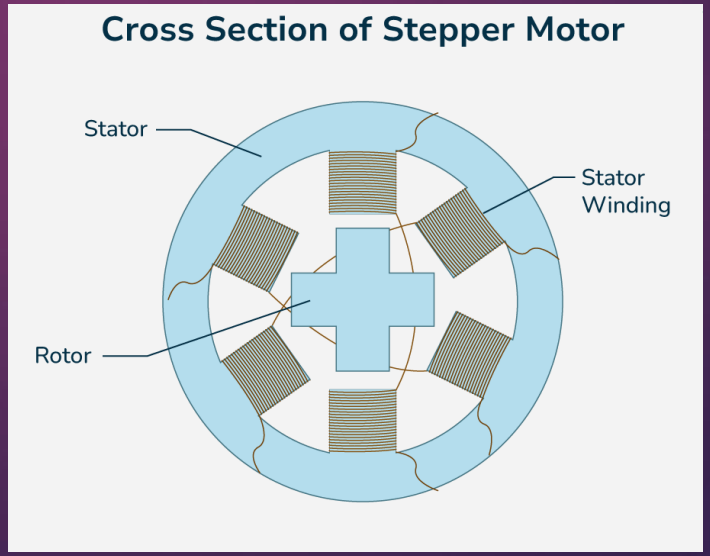
# Microprocessor Interfacing & Programming

LECTURE 29

# Stepper Motors

- A stepper motor is a widely used device that translates electrical pulses into mechanical movement.
- In disk drives, dot matrix printers and robotics, the stepper motor is used for position control.
- Stepper motor commonly have a permanent magnet rotor (shaft) surrounded by a stator.



# Four-Step Sequence

- As the sequence of power is applied to each stator winding, the rotor will rotate.
- There are several widely used sequences, each of which has a different degree of precision.
- A winding is a long copper wire wrapped many times around an iron core to create an electromagnetic field.
- When electricity flows through the winding:

1. It creates a magnetic field
2. That magnetic field interacts with parts of the motor
3. Causing rotation or movement

Note that although we can start with any of the sequences in Table 17-3, once we start we must continue in the proper order. For example, if we start with step 3 (0110), we must continue in the sequence of steps 4, 1, 2, etc.

**Table 17-3: Normal Four-Step Sequence**

Clockwise 

| Step # | Winding A | Winding B | Winding C | Winding D |
|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 |
| 2 | 1 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 0 | 0 | 1 | 1 |

Counter-clockwise 

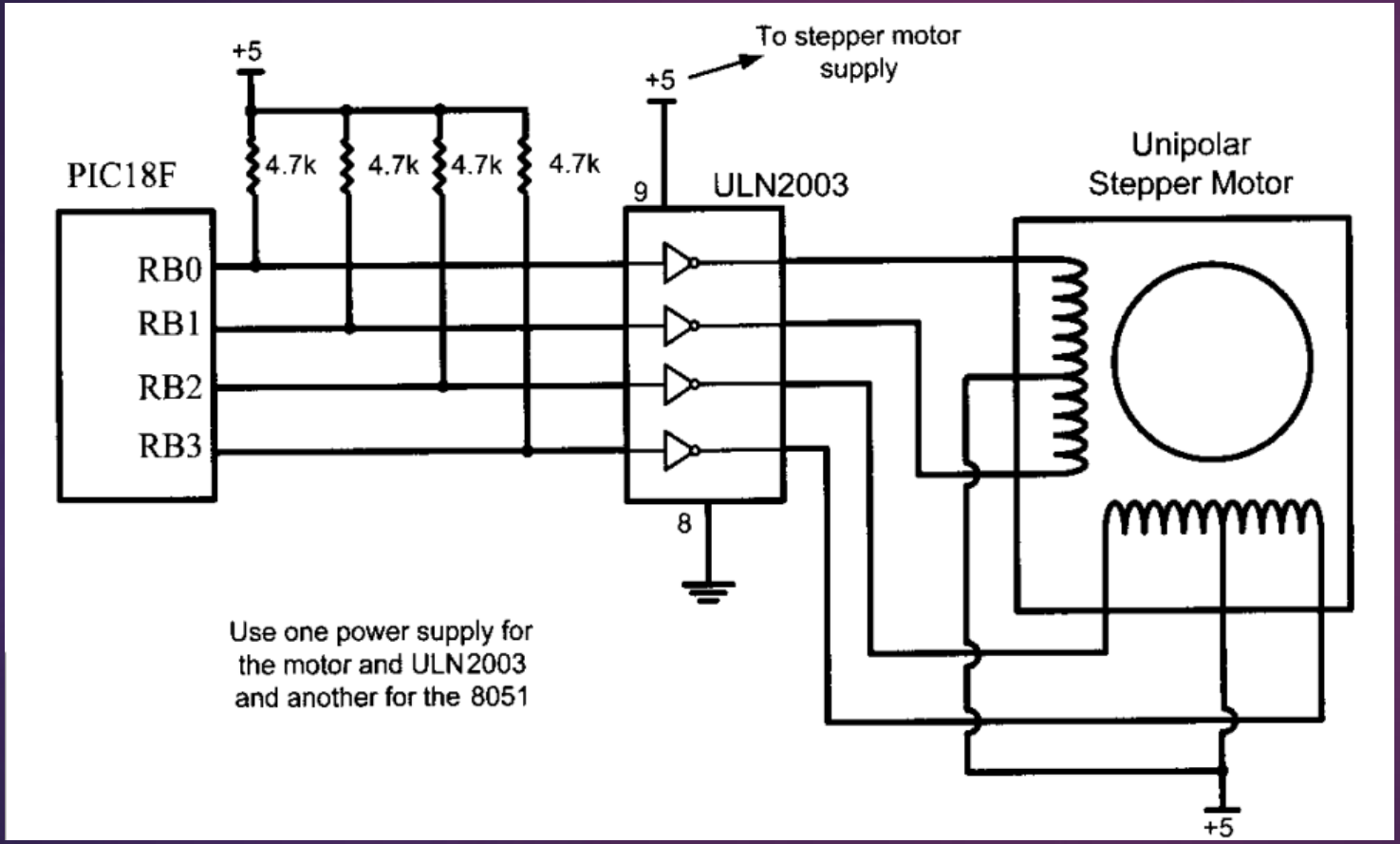
# Step Angle

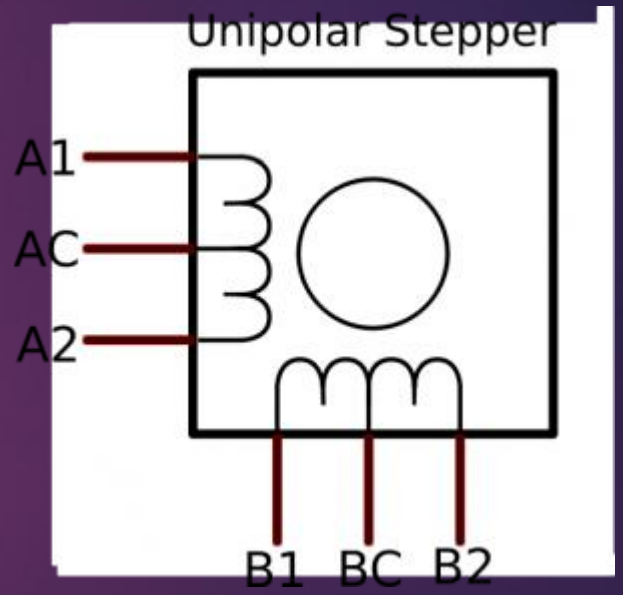
- How much movement is associated with a single step?
- The step angle is the minimum degree of rotation associated with a single step.
- Steps per Revolution: The total number of steps needed to rotate one complete rotation or 360 degrees.
- Step angle = 360 degrees / Number of steps per revolution

**Table 17-4: Stepper Motor Step Angles**

| Step Angle | Steps per Revolution |
|---|---|
| 0.72 | 500 |
| 1.8 | 200 |
| 2.0 | 180 |
| 2.5 | 144 |
| 5.0 | 72 |
| 7.5 | 48 |
| 15 | 24 |

# Example


+5
To stepper motor supply
+5
PIC18F
4.7k
4.7k
4.7k
4.7k
9
ULN2003
Unipolar Stepper Motor
RB0
RB1
RB2
RB3
8
Use one power supply for the motor and ULN2003 and another for the 8051
+5


Unipolar Stepper
A1
AC
A2
B1
BC
B2

# 2 phase stepper motor interfacing

- A 2-phase unipolar stepper motor has two coils (Phase A and Phase B), but each coil has two halves, giving 4 control wires:

1. Coil A1
2. Coil A2
3. Coil B1
4. Coil B2

- Using the 4 wires, the controller energizes the coils in a 4-step sequence to make the rotor move.
- 2 phase means two phases energized at a time to create stronger torque and smoother rotation.
- The PIC cannot supply enough current to drive the motor coils directly, so the ULN2003 transistor array acts as the driver.

Describe the PIC18 connection to the stepper motor of Figure 17-9 and code a program to rotate it continuously.

**Solution:**

The following steps show the PIC18 connection to the stepper motor and its programming:

1. Use an ohmmeter to measure the resistance of the leads. This should identify which COM leads are connected to which winding leads.
2. The common wire(s) are connected to the positive side of the motor's power supply. In many motors, +5 V is sufficient.
3. The four leads of the stator winding are controlled by four bits of the PIC18 port (RB0–RB3). Because the PIC18 lacks sufficient current to drive the stepper motor windings, we must use a driver such as the ULN2003 to energize the stator. Instead of the ULN2003, we could have used transistors as drivers, as shown in Figure 17-11. However, notice that if transistors are used as drivers, we must also use diodes to take care of inductive current generated when the coil is turned off. One reason that using the ULN2003 is preferable to the use of transistors as drivers is that the ULN2003 has an internal diode to take care of back EMF.

```
MyReg           SET    0x30                ;loc 30H for MyReg
R2              SET    0x20                ;loc 20H for R2 Reg
                CLRF   TRISB               ;Port B as output
                MOVLW  0x66                ;load step sequence
                MOVWF  MyReg
BACK            MOVFF  MyReg,PORTB         ;issue sequence to motor
                RRNCF  MyReg,F             ;rotate right clockwise
                CALL DELAY                 ;wait
                BRA    BACK                ;keep going
DELAY
                MOVLW  0xFF
                MOVWF  R2
D1              NOP
                DECF   R2,F
                BNZ    D1
                RETURN
                END
```

Change the value of DELAY to set the speed of rotation.
We can use the single-bit instructions BSF and BCF instead of RRNCF to create the sequences.

# Why Delay required?

- If we change the coil pattern too fast, the rotor cannot physically follow because of inertia and electromagnetic limitations.
- This causes the motor to miss steps, vibrate or stall.
- Therefore, we add delay between each step to give the rotor time to move to the new position.
- If the delay is to short, the motor vibrates or skips steps.
- If the delay is too long, the motor rotates too slowly.

# Number of teeth on rotor

- In 4 step switching sequence, the same two windings will be ON.
- How much movement is associated with these 4 steps?
- After completing every 4 steps, the rotor moves only one tooth pitch.
- Therefore, in a stepper motor with 200 steps per revolution, the rotor has 50 teeth because 4x50 = 200 steps that are neede to complete one revolution.
- The smallet the step angle, the more teeth the rotor passes.

**Teeth of rotor:**

In machines (motors, generators, turbines), the rotor sometimes has teeth, small projections around its edge that interact with the stator or help shape the magnetic field.

## Steps per second and rpm relation

The relation between rpm (revolutions per minute), steps per revolution, and steps per second is as follows.

$$Steps\ per\ second = \frac{rpm \times Steps\ per\ revolution}{60}$$

Give the number of times the four-step sequence in Table 17-3 must be applied to a stepper motor to make an 80-degree move if the motor has a 2-degree step angle.

**Solution:**

A motor with a 2-degree step angle has the following characteristics:

| | | | |
|---|---|---|---|
| Step angle: | 2 degrees | Steps per revolution: | 180 |
| Number of rotor teeth: | 45 | Movement per 4-step sequence: | 8 degrees |

To move the rotor 80 degrees, we need to send 10 consecutive 4-step sequences, because $10 \times 4$ steps $\times$ 2 degrees = 80 degrees.

# Example

A switch is connected to pin RD7 (PORTD.7). Write a program to monitor the status of SW and perform the following:
(a) If SW = 0, the stepper motor moves clockwise.
(b) If SW = 1, the stepper motor moves counterclockwise.

**Solution:**

```
MyReg     SET   0x30              ;loc 30H for MyReg
          BSF   TRISD,RD7         ;RD7 as input pin
          CLRF  TRISB             ;Port B as output
          MOVLW 0x66              ;load step sequence
          MOVWF MyReg
BACK      BTFSS PORTD,RD7         ;check the SW
          BRA   OVER        ;It is high. Make it clockwise
          MOVFF MyReg,PORTB       ;issue sequence to motor
          RRNCF MyReg,F           ;rotate right clockwise
          CALL  DELAY             ;wait
          BRA   BACK              ;keep going
OVER      MOVFF MyReg,PORTB       ;issue sequence to motor
          RLNCF MyReg,F           ;rotate left clockwise
          CALL  DELAY             ;wait
          BRA   BACK              ;keep going
```

# Stepper motor control with PIC18 C

```
#include <p18f458.h>
void main()
   {
      TRISB=0x0;          //PORTB as output
      while(1)
         {
            PORTB = 0x66;
            MSDelay(100);
            PORTB = 0xCC;
            MSDelay(100);
            PORTB = 0x99;
            MSDelay(100);
            PORTB = 0x33;
            MSDelay(100);
         }
   }
```

A switch is connected to pin RD7. Write a C program to monitor the status of SW and perform the following:
(a) If SW = 0, the stepper motor moves clockwise.
(b) If SW = 1, the stepper motor moves counterclockwise.

**Solution:**

```
#include <p18f458.h>
#define SW PORTDbits.RD7
void MSDelay(int ms);
void main()
  {
    TRISD=0x80;        //RD7 as input pin
    TRISB=0x0;         //PORTB as output
    while(1)
      {
        if(SW == 0)
          {
          PORTB = 0x66;
          MSDelay(100);
          PORTB = 0xCC;
          MSDelay(100);
          PORTB = 0x99;
          MSDelay(100);
          PORTB = 0x33;
          MSDelay(100);
          }
        else
          {
          PORTB = 0x66;
          MSDelay(100);
          PORTB = 0x33;
          MSDelay(100);
          PORTB = 0x99;
          MSDelay(100);
          PORTB = 0xCC;
          MSDelay(100);
          }
      }
  }

void MSDelay(unsigned int value)
  {
    unsigned int x, y;
    for(x=0;x<1275;x++)
      for(y=0;y<value;y++);
  }
```