MKSSS's

CUMMINS COLLEGE OF ENGINEERING FOR WOMEN

DEPARTMENT OF COMPUTER ENGINEERING

Internet of Things Laboratory

Assignment 6

Shreya Pawaskar
Cno - C22018881961
Roll no - 4947
Division - C

Problem Statement: Write an application using a Beagle board to control the operation of hardware by implementing Stepper motor rotation clockwise and anticlockwise using Full step and a half step.

Theory:

Our setup

- Our stepper motor type is Permanent Magnet
  Stepper motor: PM42S-100 from Minebea
- Steps per Rotation = 100
- Step angle is 3.6 degree
- No. of phases 4

Types of Stepper Motor: The other 3 types of stepper motor available are

– Hybrid synchronous stepper
– Variable reluctance stepper
– Lavet type stepping motor

Types of Stepper Motors

• Permanent magnet motors use a permanent magnet (PM) in the rotor and operate on the attraction or repulsion between the rotor PM and the stator electromagnets

• Variable reluctance (VR) motors have a plain iron rotor and operate based on the principle that minimum reluctance occurs with minimum gap, hence the rotor points are attracted toward the stator magnet poles

• Hybrid stepper motors are named because they use a combination of PM and VR techniques to achieve maximum power in a small package size

Step Angle Calculation

$$\text{Step Angle}(\Theta_S) = \frac{360°}{S}$$

$$S = mN_r$$

$$m = \textit{number of phases}$$

$$N_r = \textit{number of rotor teeth}$$

- 25 rotor teeth
- 4 phases
- Step angle is 3.6 degrees

Full Stepping

• This is the usual method for full-step driving the motor

• Two phases are always on so the motor will provide its maximum rated torque

STEP	L1	L2	L3	L4
1	H	H	L	L
2	L	H	H	L
3	L	L	H	H
4	H	L	L	H

Half Stepping

• When half-stepping, the drive alternates between two phases on and a single phase on

• This increases the angular resolution

• Its angle per step is half of the full step, thus for our stepper motor, there will be 25*8 = 200 steps per full rotation and each step will be 360/200 = 1.8°

Half Stepping

STEP	L1	L2	L3	L4
1	H	L	L	L
2	H	H	L	L
3	L	H	L	L
4	L	H	H	L
5	L	L	H	L
6	L	L	H	H
7	L	L	L	H
8	H	L	L	H

1) What is a Stepper Motor?

- ➔ A stepper motor, also known as a step motor or stepping motor, is a brushless DC electric motor that divides a full rotation into a number of equal steps.
- ➔ They are DC motors that move in discrete steps.
- ➔ They have multiple coils that are organized in groups called "phases".
- ➔ By energizing each phase in sequence, the motor will rotate, one step at a time.
- ➔ With computer-controlled stepping, you can achieve very precise positioning and/or speed control.

2) Explain different types of Stepper motors.

There are three main types of stepper motors:

- Permanent Magnet Stepper.

  PM steppers have rotors that are constructed with permanent magnets, which interact with the electromagnets of the stator to create rotation and torque. PM steppers usually have comparatively low power requirements and can produce more torque per unit of input power. This solution guarantees a good torque and also a detent torque. This means the motor will resist, even if not very strongly, to a change of position regardless of whether a coil is energized. The drawback of this solution is that it has a lower speed and a lower resolution compared to the other types.
- Variable Reluctance Stepper.

  VR stepper rotors are not built with permanent magnets. Rather, they are constructed with plain iron and resemble a gear, with protrusions or “teeth” around the circumference of the rotor. The teeth lead to VR steppers that have a very high degree of angular resolution; however, this accuracy usually comes at the expense of torque.
- Hybrid Synchronous Stepper.

HS stepper rotors use the best features of both PM and VR steppers. The rotor in an HS motor has a permanent magnet core, while the circumference is built from plain iron and has teeth. A hybrid synchronous motor, therefore, has both high angular resolution and high torque.

3) Explain different components of Stepper Motor - stator, rotor, Windings, Torque

★ Stator:

- The stationary part of the motor has a number of coil windings.
- A magnet on a shaft spins within the stator, creating alternating current (AC).
- The number of phases is the number of independent coils, while the number of pole pairs indicates how main pairs of teeth are occupied by each phase.

★ Rotor:

- Rotors are the moving part in an Alternator that have permanent magnets that move around the Stator's iron plates to generate an Alternating Current (AC).
- Rotors require existing motion to function, so only once the engine or turbine is already running will a Rotor work with a Stator to provide a charge.
- The rotor is a permanent magnet that aligns with the magnetic field generated by the stator circuit.

★ Windings:

- It is material (such as wire) wound or coiled about an object like a stator.

★ Torque:

- Torque is the measure of the force that can cause an object to rotate about an axis.
- A stepper motor divides a full rotation into a number of equal steps, important for many industrial motor and motion control applications.

4) Explain the working of the Stepper Motor.

The stepper motor rotor is a permanent magnet, when the current flows through the stator winding, the stator winding produces a vector magnetic field. The magnetic field drives the rotor to rotate by an angle so that the pair of magnetic fields of the rotor and the magnetic field direction of the stator are consistent. When the stator's vector magnetic field is rotated by an angle, the rotor also rotates with the magnetic field at an angle. Each time an electrical pulse is an input, the motor rotates one degree further. The angular displacement it outputs is proportional to the number of pulses input and the speed is proportional to the pulse frequency. Change the order of winding power, the motor will reverse. Therefore, it can control the rotation of the stepping motor by controlling the number of pulses, the frequency, and the electrical sequence of each phase winding of the motor.

5) How to calculate step angle?

- The step angle of the stepper motor is defined as the angle traversed by the motor in one step.
- To calculate step angle, we simply divide 360 by the number of steps a motor takes to complete one revolution.

$$\text{Step Angle}(\Theta_s) = \frac{360^\circ}{S}$$

$$S = mN_r$$

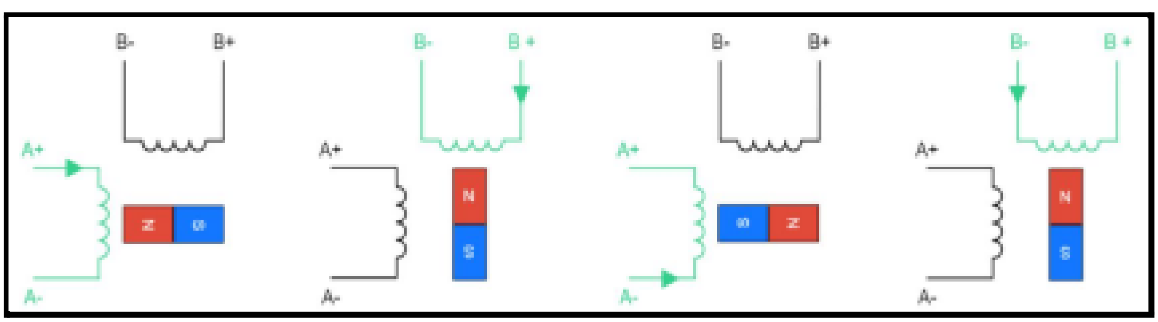
$$m = \text{number of phases}$$

$$N_r = \text{number of rotor teeth}$$

6) Explain different types of step sequencing with their step values - Wave step, Half step, Full step.

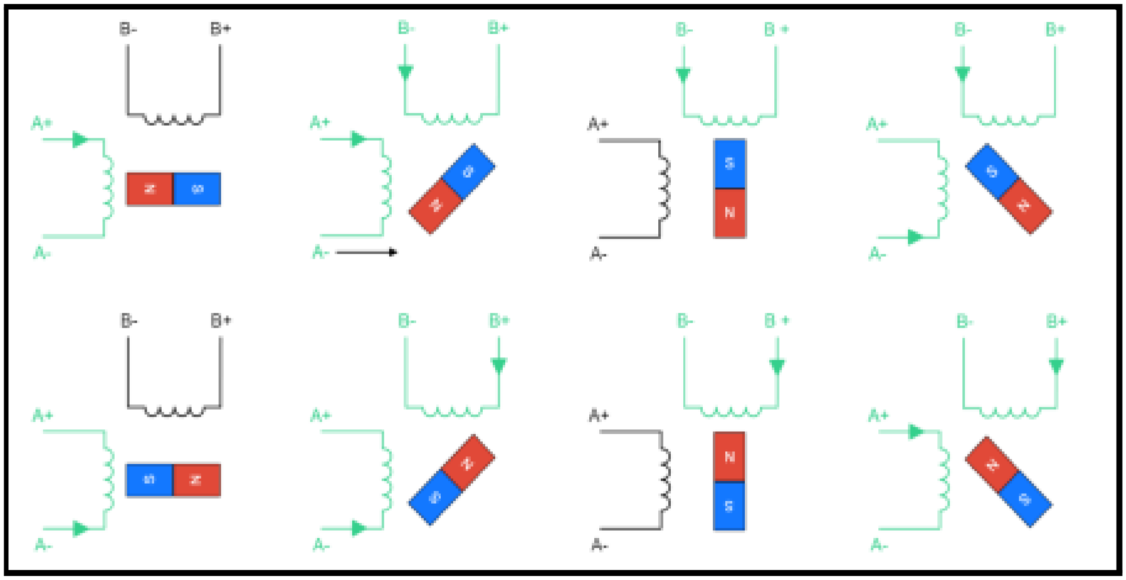
- **Wave step:**

Only one phase at a time is energized. Starting from the left, the current is flowing only in phase A in the positive direction and the rotor, represented by a magnet, is aligned with the magnetic field generated by it. In the next step, it flows only in phase B in the positive direction, and the rotor spins 90° clockwise to align with the magnetic field generated by phase B. Later, phase A has energized again, but the current flows in the negative direction, and the rotor spins again by 90°. In the last step, the current flows negatively in phase B and the rotor spins again by 90°.



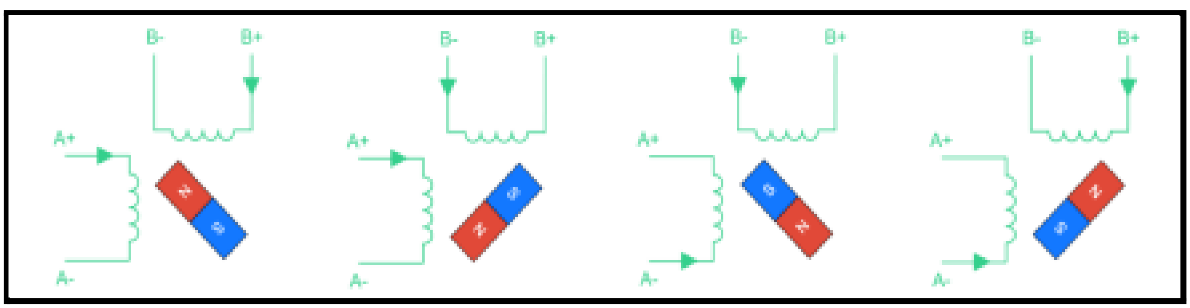
- **Half step:**

Half-step mode is a combination of wave and full-step modes. Using this combination allows for the step size to be reduced by half (in this case, 45° instead of 90°). The only drawback is that the torque produced by the motor is not constant, since it is higher when both phases are energized, and weaker when only one phase is energized.



- **Full step:**

Two phases are always energized at the same time. The steps are similar to the wave mode ones, the most significant difference being that with this mode, the motor is able to produce a higher torque since more current is flowing in the motor and a stronger magnetic field is generated.



7) List specifications of Stepper Motor used in the laboratory.

- Stepper Motor Type - Permanent magnet stepper motor: PM42S-100 from Minebea
- Steps per rotation - 100
- Step angle 3.6 degree
- Number of phases 4

8) Attach Program

Anti-Clockwise: Half stepping

```python
import Adafruit_BBIO.GPIO as GPIO
import time

GPIO.setup("P9_11", GPIO.OUT)
GPIO.setup("P9_12", GPIO.OUT)
GPIO.setup("P9_13", GPIO.OUT)
GPIO.setup("P9_14", GPIO.OUT)

while True:
    GPIO.output("P9_11",GPIO.LOW)
    GPIO.output("P9_12",GPIO.LOW)
    GPIO.output("P9_13",GPIO.LOW)
    GPIO.output("P9_14",GPIO.HIGH)
    time.sleep(2)
    GPIO.output("P9_11",GPIO.LOW)
    GPIO.output("P9_12",GPIO.LOW)
    GPIO.output("P9_13",GPIO.HIGH)
    GPIO.output("P9_14",GPIO.HIGH)
    time.sleep(2)
    GPIO.output("P9_11",GPIO.LOW)
    GPIO.output("P9_12",GPIO.LOW)
    GPIO.output("P9_13",GPIO.HIGH)
    GPIO.output("P9_14",GPIO.LOW)
    time.sleep(2)
    GPIO.output("P9_11",GPIO.LOW)
    GPIO.output("P9_12",GPIO.HIGH)
    GPIO.output("P9_13",GPIO.HIGH)
    GPIO.output("P9_14",GPIO.LOW)
    time.sleep(2)
    GPIO.output("P9_11",GPIO.LOW)
    GPIO.output("P9_12",GPIO.HIGH)
    GPIO.output("P9_13",GPIO.LOW)
    GPIO.output("P9_14",GPIO.LOW)
    time.sleep(2)
    GPIO.output("P9_11",GPIO.HIGH)
    GPIO.output("P9_12",GPIO.HIGH)
    GPIO.output("P9_13",GPIO.LOW)
    GPIO.output("P9_14",GPIO.LOW)
    time.sleep(2)
    GPIO.output("P9_11",GPIO.HIGH)
    GPIO.output("P9_12",GPIO.LOW)
    GPIO.output("P9_13",GPIO.LOW)
    GPIO.output("P9_14",GPIO.LOW)
    time.sleep(2)
    GPIO.output("P9_11",GPIO.HIGH)
    GPIO.output("P9_12",GPIO.LOW)
```

```
        GPIO.output("P9_13",GPIO.LOW)
        GPIO.output("P9_14",GPIO.HIGH)
        time.sleep(2)
```

Anti Clockwise: Full Step

```
import Adafruit_BBIO.GPIO as GPIO import
time

GPIO.setup("P9_11",GPIO.OUT)
GPIO.setup("P9_12",GPIO.OUT)
GPIO.setup("P9_13",GPIO.OUT)
GPIO.setup("P9_14", GPIO.OUT)

while True:
        GPIO.output("P9_11",GPIO.LOW)
        GPIO.output("P9_12",GPIO.LOW)
        GPIO.output("P9_13",GPIO.LOW)
        GPIO.output("P9_14",GPIO.HIGH)
        time.sleep(2)
        GPIO.output("P9_11",GPIO.LOW)
        GPIO.output("P9_12",GPIO.LOW)
        GPIO.output("P9_13",GPIO.HIGH)
        GPIO.output("P9_14",GPIO.LOW)
        time.sleep(2)
        GPIO.output("P9_11",GPIO.LOW)
        GPIO.output("P9_12",GPIO.HIGH)
        GPIO.output("P9_13",GPIO.LOW)
        GPIO.output("P9_14",GPIO.LOW)
        time.sleep(2)
        GPIO.output("P9_11",GPIO.HIGH)
        GPIO.output("P9_12",GPIO.LOW)
        GPIO.output("P9_13",GPIO.LOW)
        GPIO.output("P9_14",GPIO.LOW)
        time.sleep(2)
```