Internet of Things Laboratory

Assignment 4

Prajakta Deokule

C22019221332

4329

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

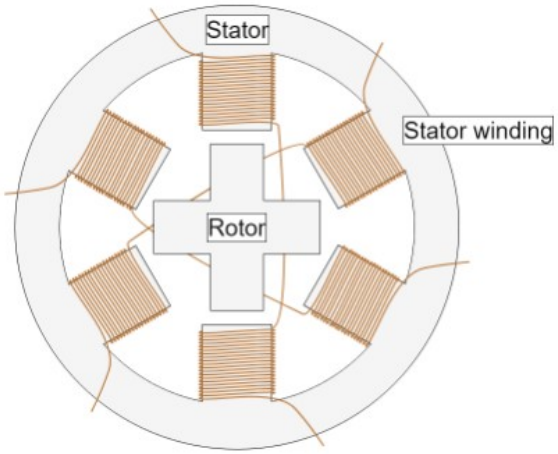
1. **What is a Stepper motor?**

   Stepper motors 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.

2. **Explain different types of Stepper motors.**

   - **Permanent magnet motors-** The rotor is a permanent magnet that aligns with the magnetic field generated by the stator circuit
   - **Variable reluctance (VR) motors-** It is the *basic type of stepper motor* that has been in existence for a long time and it ensures easiest way to understand principle of operation from a structural point of view. As the name suggests, the angular position of the rotor depends on the reluctance of the magnetic circuit formed between the stator poles (teeth) and rotor teeth.
   - **Hybrid stepper motors-** are named because they use a combination of PM and VR techniques to achieve maximum power in a small package size.

3. **Explain different components of Stepper motor - stator, rotor, windings, torque with a diagram.**



Components of a stepper motor

Stator- The stator is the part of the motor responsible for creating the magnetic field with which the rotor is going to align. The main characteristics of the stator circuit include its number of phases and pole pairs, as well as the wire configuration. 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- The rotor is made up of three components: rotor 1, rotor 2 and a permanent magnet. The rotor is magnetized in the axial direction so that, for example, if rotor 1 is polarized north, rotor 2 will be polarized south.

Windings- Windings made of conductive material around stators facilitate current flow. The stepper motor rotates through the sequential switching of current flowing through the windings.

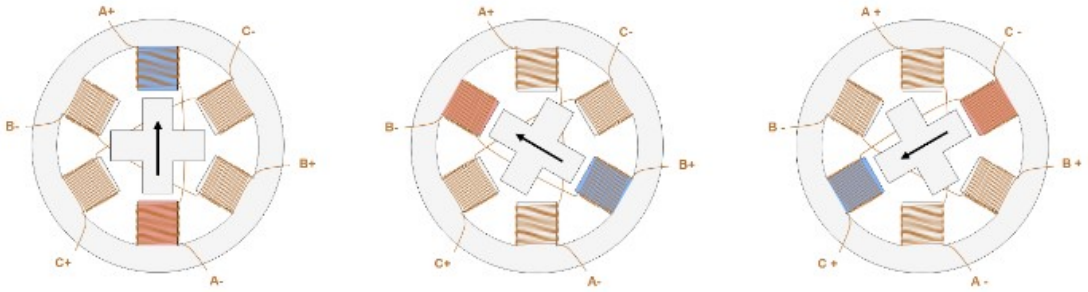
Torque- There are 2 types of torque: Holding torque and Detent torque. Holding Torque- stepper motor's ability to hold a load in place when the motor coils are energized, due to the attraction between the rotor and stator.

Detent torque occurs when the motor is not energized — that is, no current is flowing through the motor windings. It occurs because of the attractive forces between the magnets in the motor's rotor and the stator. Both permanent magnet and hybrid stepper motors experience detent torque, but variable reluctance designs do not.

4. Explain working of Stepper motor.

As all with electric motors, stepper motors have a stationary part (the stator) and a moving part (the rotor). On the stator, there are teeth on which coils are wired, while the rotor is either a permanent magnet or a variable reluctance iron core.

The basic working principle of the stepper motor is the following: By energizing one or more of the stator phases, a magnetic field is generated by the current flowing in the coil and the rotor aligns with this field. By supplying different phases in sequence, the rotor can be rotated by a specific amount to reach the desired final position. Figure given below shows a representation of the working principle. At the beginning, coil A is energized and the rotor is aligned with the magnetic field it produces. When coil B is energized, the rotor rotates clockwise by 60° to align with the new magnetic field. The same happens when coil C is energized. In the pictures, the colors of the stator teeth indicate the direction of the magnetic field generated by the stator winding.



Working of a stepper motor

5. How to calculate Step angle?

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

Step angle calculation: ø = 360/S

Where, S = number of steps required to complete 1 revolution,

i.e. S = mN, m = number of phases N = number of rotor teeth

E.g. Stepper Motor rotating in full mode takes 4 steps to complete a revolution, So step angle can be calculated as… Step Angle ø = 360° / 4 = 90° and in case of half mode step angle gets half so 45°

6. Explain different types of Step sequencing with their step value- wave step, half step, full step.

- **Wave stepping-** In wave mode, only one phase at a time is energized. It has the same number of steps as the full step drive, but the motor will have significantly less than rated torque. It is the default mode of the stepper motor, but still it is rarely used.

Wave Stepping

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

- **Full Stepping-** In full-step mode, 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. This is the usual method for driving the stepper motor.

Two phases are always on. Hence motor gives the highest torque.

Full Stepping

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**- Half-step mode is a combination of wave and full-step modes. This combination reduces step size to half. Number of phases is double that of Full stepping mode. 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.

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

7. **List specifications of Stepper motor used in laboratory practical.**

   Type - Permanent Magnet Stepper Motor (PM 425 - 100 from Minebea)

   Steps per rotation - 100

   Step Angle = 3.6 degrees

   No. of phases = 4

Code

Anticlockwise rotation using full step

```
import Adafruit_BBIO.GPIO as GPIO
import time
LED=["P9_11","P9_12","P9_13","P9_14"]
for i in range(len(LED)):
        GPIO.setup(LED[i],GPIO.OUT)
        GPIO.output(LED[i],GPIO.LOW)
time.sleep(5)

while True:
        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(1)
        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(1)
        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(1)
        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(1)
print("Anticlockwise done")
```

Clockwise rotation using full step

```
import Adafruit_BBIO.GPIO as GPIO
import time
LED=["P9_11","P9_12","P9_13","P9_14"]
for i in range(len(LED)):
        GPIO.setup(LED[i],GPIO.OUT)
        GPIO.output(LED[i],GPIO.LOW)
time.sleep(5)

while True:
        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(1)
```

```
        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(1)

       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(1)

        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(1)

print("Clockwise done")
```

Anticlockwise rotation using half step

import Adafruit_BBIO.GPIO as GPIO

```
import time
LED=["P9_11","P9_12","P9_13","P9_14"]
for i in range(len(LED)):
        GPIO.setup(LED[i],GPIO.OUT)
        GPIO.output(LED[i],GPIO.LOW)
time.sleep(5)

while True:
        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(1)

        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(1)

        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(1)

        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(1)
```

```
        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(1)


        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(1)


        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(1)


        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(1)
```

Clockwise Rotation using Half Step

```
import Adafruit_BBIO.GPIO as GPIO
import time
LED=["P9_11","P9_12","P9_13","P9_14"]
for i in range(len(LED)):
        GPIO.setup(LED[i],GPIO.OUT)
        GPIO.output(LED[i],GPIO.LOW)
time.sleep(5)

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(1)

        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(1)

        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(1)

        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(1)

        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(1)

        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(1)

        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(1)
```

Wave Stepping

```
import Adafruit_BBIO.GPIO as GPIO
import time
LED=["P9_11","P9_12","P9_13","P9_14"]
for i in range(len(LED)):
        GPIO.setup(LED[i],GPIO.OUT)
        GPIO.output(LED[i],GPIO.LOW)
time.sleep(5)

while True:
        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(1)

        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(1)

        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(1)

        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(1)
```