EE128: Data Acquisition, Instrumentation and Process Control
University of California, Riverside

Lab 6: Stepper Motor Control

Spring 2019

Objective

To get familiar with the control of stepper motors.

References

- Dragon12-JR Trainer User's Manual
- Freescale MC9S12DG256 Device User Guide
- Freescale HCS12 Core User Guide
- Reference Guide for D–Bug12 Version 4.x.x
- L298 Dual Full-bridge Driver Specification

Equipment

- PC running MS Windows
- Digital Multi-Meter (DMM)
- Power supply (+9V)
- Dragon12-JR 9S12DG256 EVB
- **Breadboard** (optional, you need to bring one)

Parts

- 1 each 2-phase, 12V DC bipolar stepper motor
- 1 each L298N motor driver
- 1 each 4-bit DIP switch (piano type)
- 2 each 470Ω, 10-pin bussed resistor network (SIP)
- 8 each 2n4001 diodes

Software

- Freescale CodeWarrior for HC12 v5.1
- RealTerm

Background

Stepper Motors

Stepper motors are electric motors that rotate from one position to next in discrete intervals (steps). A step refers to the fraction of a full 360 of the motor shaft rotation. Typical rotation angles are 0.9, 1.8, 3.6, 4, 7.5 and 15 degrees. With reduction gears, a smaller step angle can be achieved with an amplified torque.

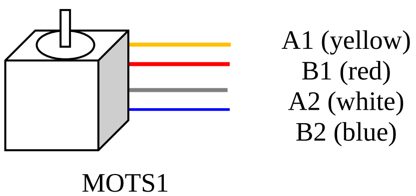
The primary stepper motor application is position control. Some disk drives use stepper motors to position the read/write head; image scanners and printers use stepper motors to rotate the paper feed carriage as well as to position the print head horizontally relative to the paper; more recently, 3D printers use stepper motors to control 3 axes of motion.

Compared to servo-based motion control systems, a stepper motor controller is typically run open-loop. Thus, it is important to make sure the motor is powerful enough to overcome maximum possible load and inertia while not losing any step counts.

The stepper motor to be used in this experiment has the following electrical characteristics:

- 2-phase, bipolar operation
- 5.625degrees per steps (64 steps per revolution)
- 12V operation, typical current 32mA
- phase resistance 300ohm; phase inductance 31 mH
- detention torque 310g-cm; holding torque 360g-cm
- 0.35 pounds

The stepper motor wire colors and driving sequences are given below:



Step	A1	A2	B1	B2
0	-	+	+	-
1	+	-	+	-
2	+	-	-	+
3	-	+	-	+

Figure 1. Stepper motor wire colors and control sequences (two-phase on, full-step)

Stepper Driving Circuit

The L298N is an integrated monolithic circuit in a 15-lead Multiwatt package. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

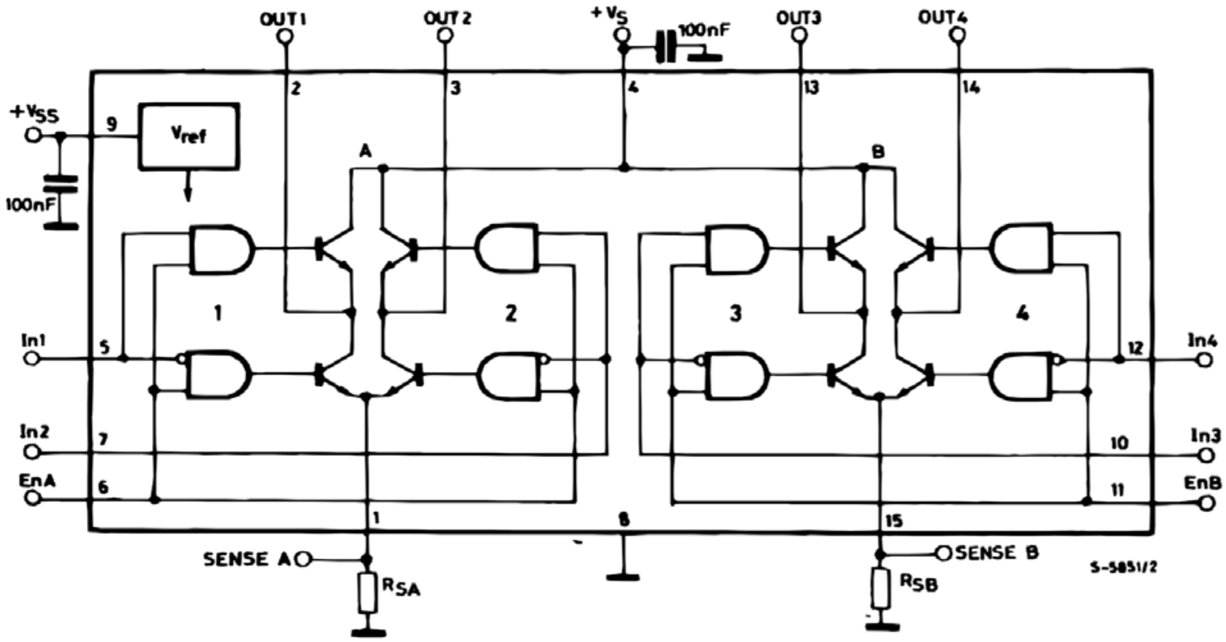


Figure 2. L298 block diagram

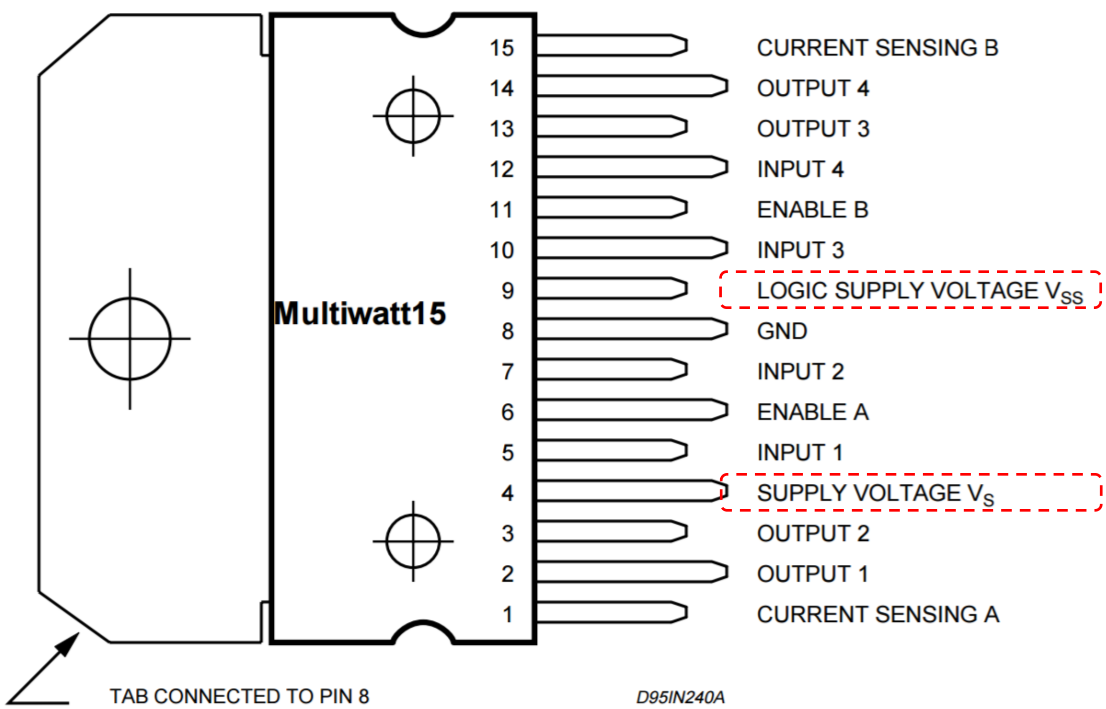


Figure 2. L298N pin connections

Specification

In this lab, we will drive the given stepper with an MC9S12DG256 microcontroller. The L298 dual bridge driver will be used as a stepper motor driver.

Your system should be able swtich among the following four operation modes:

- Rotate clockwise (CW) with an *average* speed of 22.5 degrees per second
- Rotate counterclockwise (CCW) with an *average* speed of 22.5 degrees per second
- Rotate clockwise (CW) with an *average* speed of 180 degrees per second
- Rotate counterclockwise (CCW) with an *average* speed of 180 degrees per second

The direction of the motor rotation and the angular speed are given by logic signals propoved by a 2-bit DIP switch:

- ROT_DIR: 0 for CW, 1 for CCW
- ROT_SPD: 0 for 22.5 degrees, 1 for 180 degrees

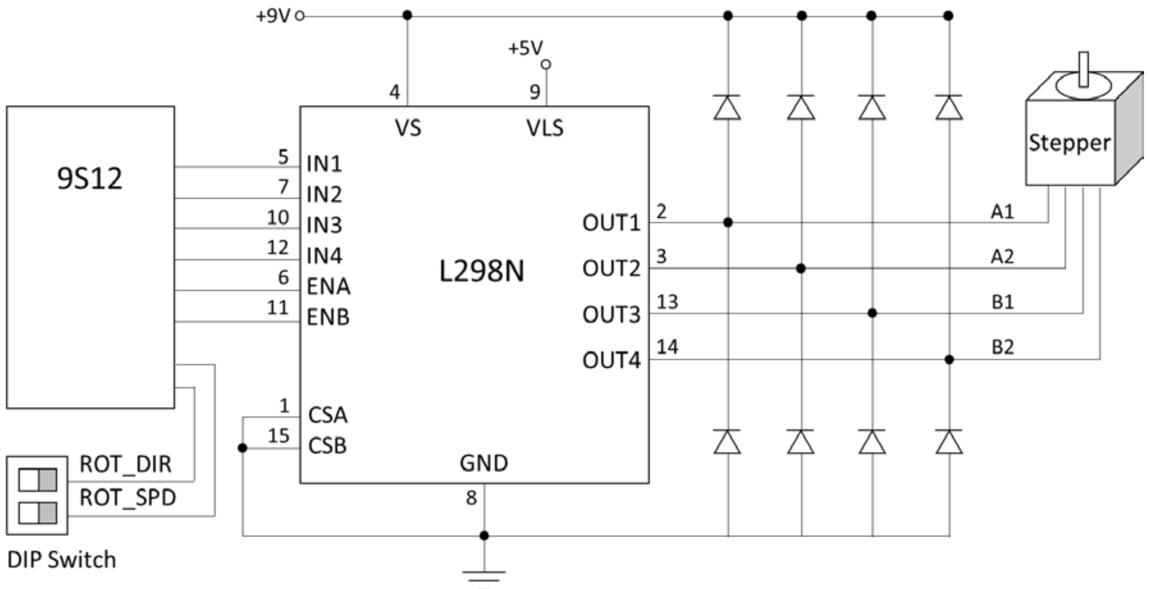


Figure 3. Overall block diagram

The overall system block diagram is shown in the above figure. It is your role to decide **which port pins to be used**. When testing, attach something (e.g., a flagging tape) to the shaft of the stepper motor so that you can easily observe the motor rotation direction and speed.

Make sure that the motor and L298 are not powerd by the 9V or 5V pins of the Dragon board. You must use an external power supply or battery pack in order to avoid overcurrent conditions.

<u>Develop and demo your system in an incremental way.</u> First, implement CW and CCW operations at 22.5 degrees/second. Second, add CW and CCW operations at 90 degrees/second and make sure that your system can switch between low and high speed and CW and CCW.

Attach a small indicator flag to the motor shaft so that the rotation is visible.

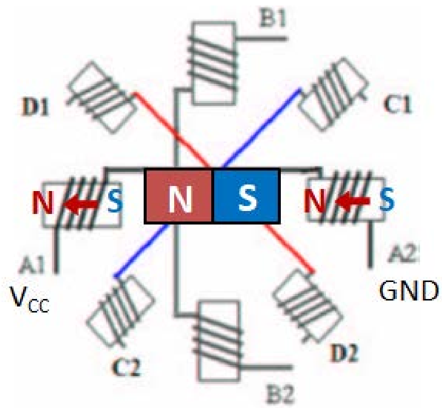
Pre-lab Questions

1. Design the required circuit for the experiment.
2. Write and compile an initial version of your source code.

Write-up Questions

1. Can a stepper motor change its speed from zero to a high value instantly? Also, can a stepper motor switch its direction while running at high speed? Answer with a brief explanation. **(3 points)**

2. Suppose that there is a 4-phase stepper motor, as shown in the right figure. The rotor magnet is assumed to have 2 poles.



a) Write a table of *clockwise* stepper control steps in "**one-phase on, full step**" mode. **(4 points)**

Step	A1	A2	B1	B2	C1	C2	D1	D2
1	+	-	-	-	-	-	-	-
…								

b) Write a table of *clockwise* stepper control steps in "**two-phase on, full step**" mode. **(4 points)**

Step	A1	A2	B1	B2	C1	C2	D1	D2
1	+	-	-	-	-	-	+	-
…								

c) Write a table of *clockwise* stepper control steps in "**half-stepping**" mode. **(4 points)**

Step	A1	A2	B1	B2	C1	C2	D1	D2
1	+	-	-	-	-	-	-	-
…								

Lab Demo (50 points)

Demonstrate your working system to the TA and get a confirmation of completion

1. Demo CW and CW operations at 22.5 degrees/second. Switching between CW and CCW should be done on the fly.

2. Demo all the four operation modes (CW and CCW at 22.5 and 90 degrees/second). The stepper motor must *not* stall when changing modes.

Lab Report (50 points)

Make sure you include the following in your report:

1. Abstract
   - Short paragraph stating the objectives and accomplishments
2. Hardware design **(10 points)**
   - Detailed schematic diagram; do not copy and paste the block diagram given the handout
   - Photos of your boards and circuits
3. Software design **(10 points)**
   - High level description of the software
   - Program code listing (including comments)
4. Discussion **(5 points)**
   - Technical challenges encountered and how they are addressed (e.g., took longer than expected due to some reasons, could not finish within the lab hours)
5. Answers to the questions **(15 points)**
6. Conclusion
   - A very short concluding remark
   - Summary of the contributions of each member