

ME133 Lab 6 Hints

Exercise 1 Part 2

Measure the voltage drop of motor, using `analogRead()` to get the voltage value at both ends of the motor. Show the voltage drop value on Serial Plotter when adjusting the pwm signal.

Exercise 2 Part 1

In the main loop, should add a function to periodically update the stepper state every “per” milliseconds, where “per” is time between step changes that maintains the current rpm (microseconds seconds per state). The function structure is:

```
if (micros() > t0 + per) {
    t0 = micros();
    .....
    set_step(wave_drive_states[i]);
}
```

To calculate period, using following equation:

```
per = (60.0 * npr) / (rpm * npc);
```

where npr is the number of steps per revolution (2038), npc is number of steps per cycle (for example, npc = 4 in wave drive state), tr is the refresh period, in milliseconds. By default, the buttons are only polled every 100ms (tr) and Serial updates are only written every 500ms (tu). This allows the Arduino to maintain the timing of the motor states with infrequent interruptions. Using `(micros() > tr0 + tr*1000)` command to set periodically read state of buttons and pot every tr period, tr*1000 transfers millisecond to microsecond (depends on tr unit you set).

To set step of motor, use the following example code. The code sets the pins driving the motor based on the bit values in state. For example, in wave drive, `set_step(wave_drive_states[1])` will take signal to pins as 0001 (decimal 1), then it should take `wave_drive_states[2]` and output 0010 (decimal 2), until the end of state.

```
const int wave_drive_states[] = {4, 1, 2, 4, 8};

void loop() {
    set_step(wave_drive_states[i]) // “i” from 2 to the end
}

void set_step(int state) {
    for (int i = 3; i >= 0; i--) {
        digitalWrite(PIN_M[3 - i], state % 2 ? HIGH : LOW);
        state /= 2;
    }
}
```

Define state flow for drive modes. The format is {n, s0,s1,...,s(n-1)}, where n is the number of states and s0 to s(n-1) are the states as implemented as binary, the following tables show the sequences of states for each drive mode. For example,

```
const int wave_drive_states[ ] = {4, 1, 2, 4, 8};
```

Wave Drive						
State	A	B	C	D	Binary	Decimal
0	0	0	0	1	0001	1
1	0	0	1	0	0010	2
2	0	1	0	0	0100	4
3	1	0	0	0	1000	8

Half Step						
State	A	B	C	D	Binary	Decimal
0	0	0	0	1	0001	1
1	0	0	1	1	0011	3
2	0	0	1	0	0010	2
3	0	1	1	0	0110	6
4	0	1	0	0	0100	4
5	1	1	0	0	1100	12
6	1	0	0	0	1000	8
7	1	0	0	1	1001	9

Full Step						
State	A	B	C	D	Binary	Decimal
0	1	0	0	1	1001	9
1	0	0	1	1	0011	3
2	0	1	1	0	0110	6
3	1	1	0	0	1100	12

In the wave drive mode, the rotor pins directly engage the poles of the stator. In full step mode, the rotor pins are pulled between adjacent poles of the stator. Finally, in the half step mode, the pins of the rotor alternate between being directly engaged by the stator poles and being engaged between adjacent poles. From these tables it is clear that the half step pattern results from the interleaving of the wave drive and full step patterns.

Exercise 2 Part 2

In the loop function, we use the `setSpeed()` function to specify the speed at which the stepper motor should move and the `step()` function to specify how many steps to take. The positive and negative values of the step indicate different directions of rotation.

When using `setSpeed()`, make sure you have set the maximum speed (`rpm_max`) as 20, to avoid motor stall. Using following code to set speed (`A0_value` comes from potentiometer):

```
rpm = A0_value * ((rpm_max) / 1023.0)
```

To calculate step, using following equation:

```
step = ((rpm/60.0)*npr*(tr/1000.0))
```

Then set speed and step:

```
myStepper.setSpeed(rpm);
```

```
myStepper.step(step);
```