EE128
Stepper Motor Control

Lab Report #5

Ana Cardenas Beltran
Alison Gonzales

Abstract

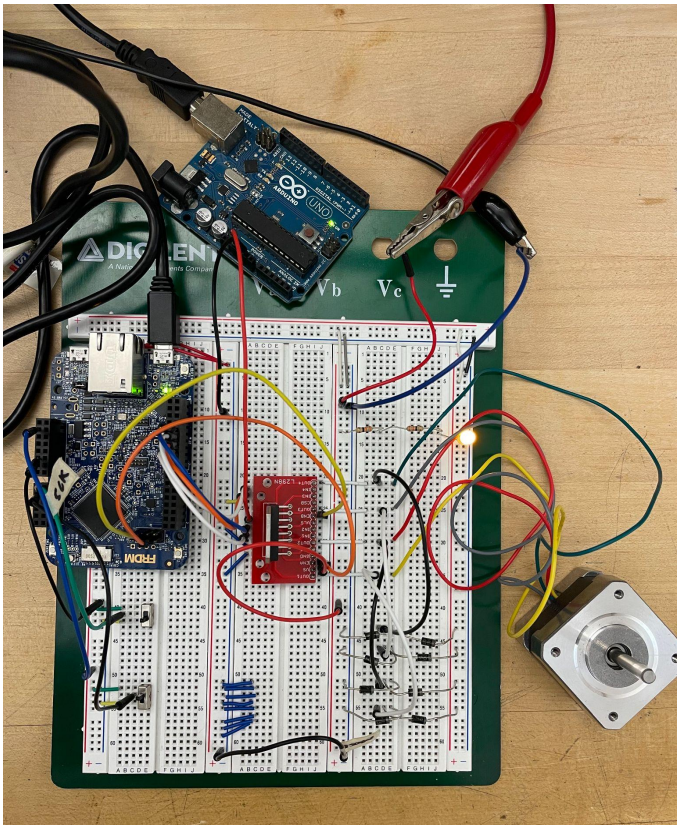
The objective of this lab is to drive a stepper motor using the Kinetis K64F microcontroller and with the L298 dual bridge driver. Additionally, multiple hardware debugging approaches were discussed and implemented while using a 2-phase, 12 V DC bipolar stepper motor.

Students accomplished the necessary skills to build a Stepper Driving Circuit through the use of the L298N. By gaining power from out sources and Arduino boards, students successfully implemented a circuit using stepper motors.

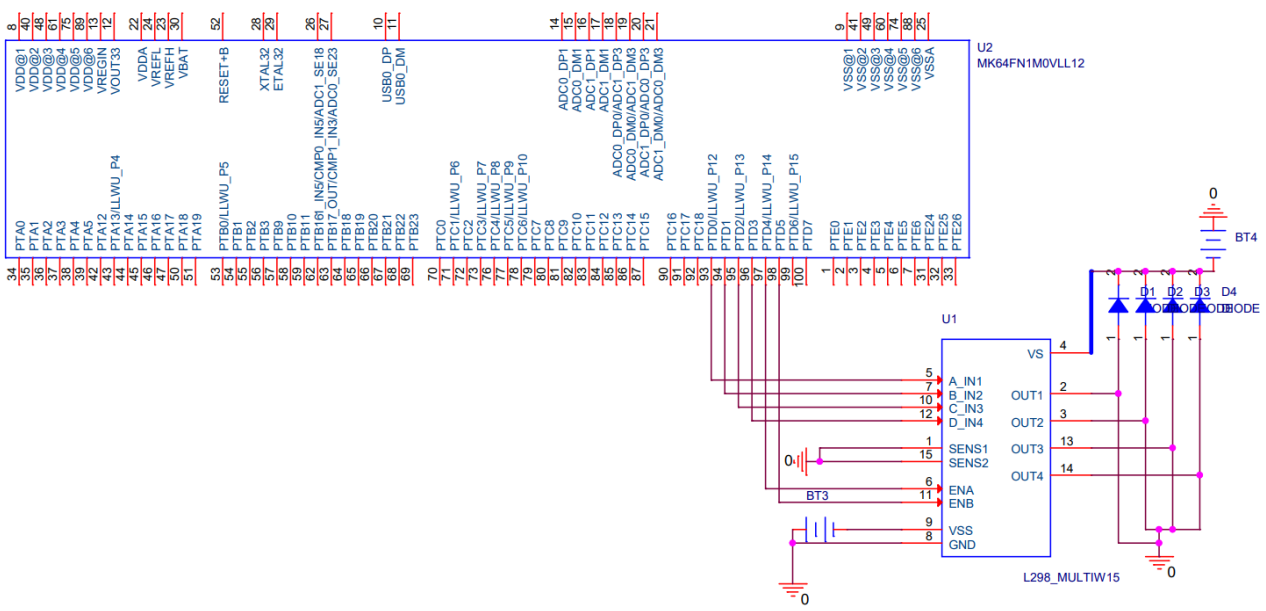
Experiment System Specification

In this lab, Freescale Kinetis Design Studio (KDS) 3.2.0 IDE was used as the main work environment for programming the Kinetis K64F development board to create a stepper driving circuit. The Arduino Uno board was used to provide additional 5V for the L298N motor driver. Multiple circuit elements: 2-phase 12 V DC bipolar stepper motor, 2 switches, 470 ohms 10-pin bussed resistor network, and 8 2n4001 diodes, students successfully implemented a stepper motor driving circuit.

Boards & Circuits



Hardware Design



Software Design

```
/* This code detects two switch inputs that allow the Stepper Motor to move in a clockwise or
counterclockwise direction and from a faster or slower rate.
 */

#include "fsl_device_registers.h"

unsigned int ROT_DIR; //0 for CW, 1 for CCW, PB2
unsigned int ROT_SPD; //0 for 22.5, 1 for 180 deg, PB3
unsigned long i;

void clockwise_fct(int);
void counterclockwise_fct(int);

int main(void)
{
	SIM_SCGC5 |= SIM_SCGC5_PORTB_MASK;    //Enable Port B Clock Gate Control
	SIM_SCGC5 |= SIM_SCGC5_PORTD_MASK;    //Enable Port D Clock Gate Control

	PORTB_GPCLR = 0x0C0100;       //PB[3:2] for GPIO
	PORTD_GPCLR = 0x3F0100;       //PD[5:0] for GPIO 0011 1111 0100

	GPIOB_PDDR = 0x0000000;       //Set PORTB for Input
	GPIOD_PDDR = 0x000003F;       //Set PORTD for Output

	while (1) {
		ROT_DIR = GPIOB_PDIR & 0x04;  //PB2
```

```
		ROT_SPD = GPIOB_PDIR & 0x08;  //PB3

		if(ROT_DIR == 0x04){
			clockwise_fct(ROT_SPD); //Call clockwise direction while reading the
appropriate speed
		}
		else {
			counterclockwise_fct(ROT_SPD); //Call counterclockwise direction while
reading the appropriate speed
		}
	}

	for (;;){}

    return 0;
}

void clockwise_fct(int speed) {

	if (speed == 0x08) {
//Full-Step
		GPIOD_PDOR = 0x36;
		for(i = 0; i < 10000; i++);
		GPIOD_PDOR = 0x35;
		for(i = 0; i < 10000; i++);
		GPIOD_PDOR = 0x39;
		for(i = 0; i < 10000; i++);
		GPIOD_PDOR = 0x3A;
		for(i = 0; i < 10000; i++);
	} else {
//Half-Step
		GPIOD_PDOR = 0x32;
		for(i = 0; i < 10000; i++);
		GPIOD_PDOR = 0x36;
		for(i = 0; i < 10000; i++);
		GPIOD_PDOR = 0x34;
		for(i = 0; i < 10000; i++);
		GPIOD_PDOR = 0x35;
		for(i = 0; i < 10000; i++);
		GPIOD_PDOR = 0x31;
		for(i = 0; i < 10000; i++);
		GPIOD_PDOR = 0x39;
		for(i = 0; i < 10000; i++);
		GPIOD_PDOR = 0x38;
		for(i = 0; i < 10000; i++);
		GPIOD_PDOR = 0x3A;
		for(i = 0; i < 10000; i++);
	}
}

void counterclockwise_fct(int speed) {

	if (speed == 0x08) {
```

```
//Full-Step
                GPIOD_PDOR = 0x3A;
                for(i = 0; i < 10000; i++);
                GPIOD_PDOR = 0x39;
                for(i = 0; i < 10000; i++);
                GPIOD_PDOR = 0x35;
                for(i = 0; i < 10000; i++);
                GPIOD_PDOR = 0x36;
                for(i = 0; i < 10000; i++);
        } else {
//Half-Step
                GPIOD_PDOR = 0x3A;
                for(i = 0; i < 10000; i++);
                GPIOD_PDOR = 0x38;
                for(i = 0; i < 10000; i++);
                GPIOD_PDOR = 0x39;
                for(i = 0; i < 10000; i++);
                GPIOD_PDOR = 0x31;
                for(i = 0; i < 10000; i++);
                GPIOD_PDOR = 0x35;
                for(i = 0; i < 10000; i++);
                GPIOD_PDOR = 0x34;
                for(i = 0; i < 10000; i++);
                GPIOD_PDOR = 0x36;
                for(i = 0; i < 10000; i++);
                GPIOD_PDOR = 0x32;
                for(i = 0; i < 10000; i++);
        }
}
```

Technical Problems

Some of the problems we encountered involved our hardware. Firstly, one of our motors initially did not work because the diode placement was not connected properly. The common ground was also not properly connected so the stepper motor was not receiving the voltage. Because of this, my battery power supply drained at a faster rate.Another issue was that some of our pins on the K64F were not working so our switches weren't working. We resolved this by connecting different pins to the switches.

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)

No, it cannot change it instantly because it takes multiple steps for the motor to increase to a high value. Different speeds require different steps. The motor also cannot switch its direction while running at a high speed for the same reason. It requires multiple steps to make a change.

2) Suppose that there is a 4-phase stepper motor, as shown in the right figure.
   a) Write a table of clockwise stepper control steps in “half-stepping” mode.(4 points)

Step	A	B	C	D
1	+	-	-	-
2	-	+	-	-
3	-	-	+	-
4	-	-	-	+

   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	-	-	+	-	-	-	-	-
2	-	-	-	-	+	-	-	-
3	-	+	-	-	-	-	-	-
4	-	-	-	-	-	-	-	+
5	-	-	-	+	-	-	-	-
6	-	-	-	-	-	+	-	-
7	+	-	-	-	-	-	-	-
8	-	-	-	-	-	-	+	-

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	-	-	+	-	-	-	-	-
2	-	-	+	-	+	-	-	-
3	-	-	-	-	+	-	-	-
4	-	+	-	-	+	-	-	-
5	-	+	-	-	-	-	-	-
6	-	+	-	-	-	-	-	+
7	-	-	-	-	-	-	-	+
8	-	-	-	+	-	-	-	+
7	-	-	-	+	-	-	-	-
9	-	-	-	+	-	+	-	-
10	-	-	-	-	-	+	-	-
11	+	-	-	-	-	+	-	-
12	+	-	-	-	-	-	-	-
13	+	-	-	-	-	-	+	-
14	-	-	-	-	-	-	+	-
15	-	-	-	+	-	-	+	-
16	-	-	-	+	-	-	-	-

Conclusion

In this lab, we created a stepper driving circuit while integrating the L298N motor driver. We learned how to generate different voltage sources to drive a stepper motor. We also learned how to deal with the directional position of each step and apply the following to create faster or slower speed on the motor.

For this lab, Ana worked on the code for the K64F and Arduino, debugging. Alison also worked on the code, schematics, and debugging process of both boards.