EE 128
Lab 6:Stepper Motor Control

Section 021
TA: Mohsen Karimi

Tanish Arora
Leya Zeng

1. Abstract

This Lab's main objectives are to getting familiar with :

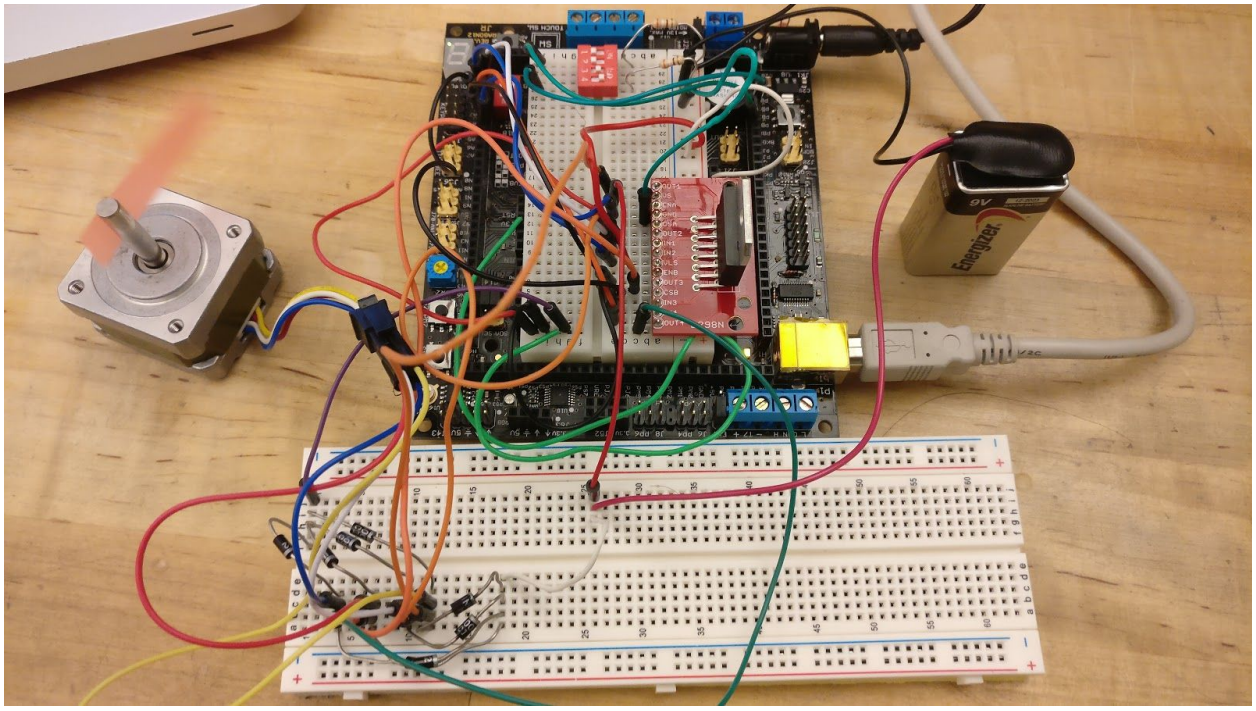
- To get familiar with control of stepper motors
- Clock ticks and control motor speed: degrees/second

2. Procedure

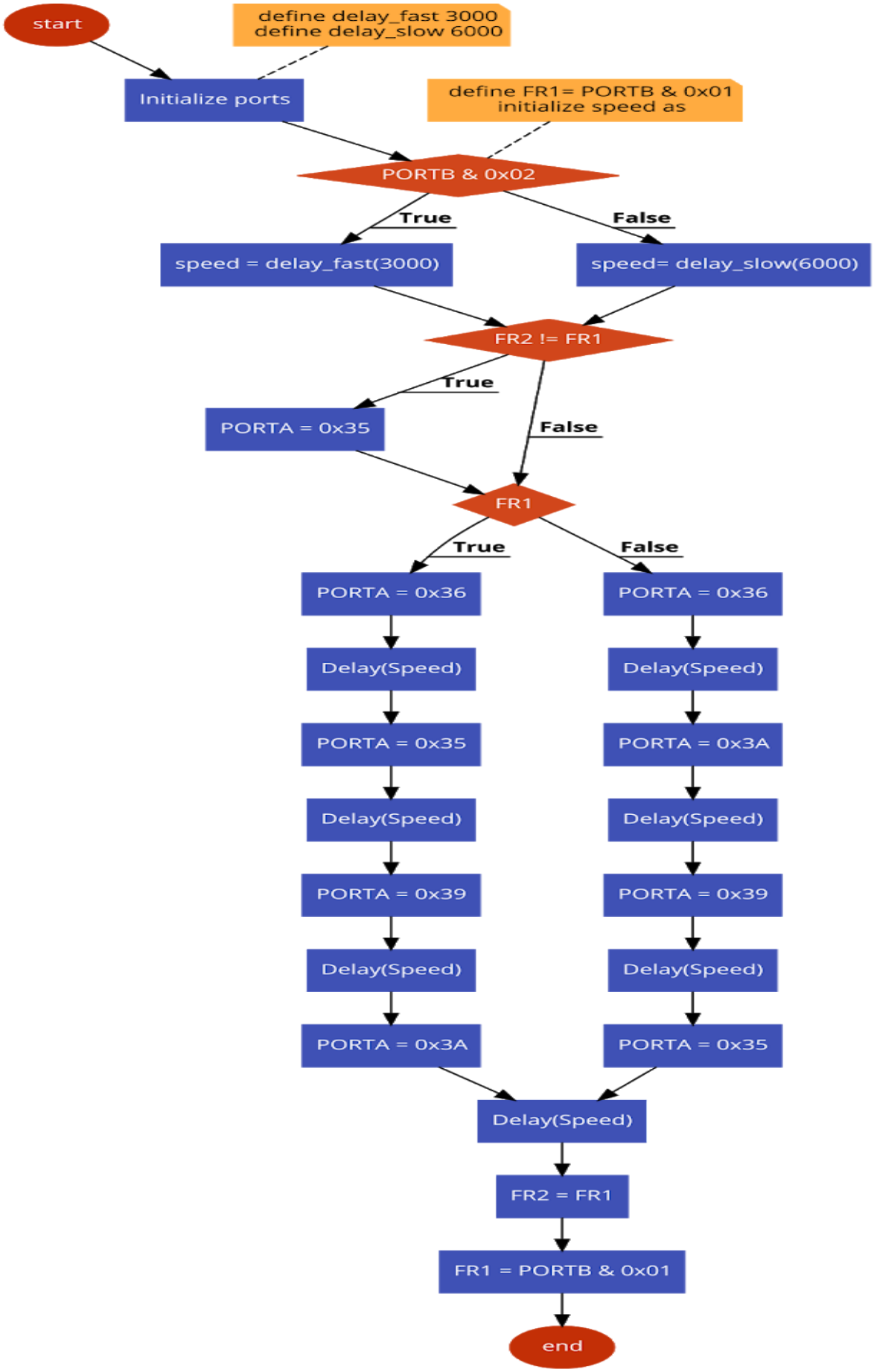
1) Design a schematic before going to the lab
2) Connect all the hardware according to the block diagram and make sure all the connections are right
3) Run the code on the lecture slides to make sure your connections are right and you're not missing any wire : use a multimeter if necessary to make those changes.
4) After checking all the wires and making sure everything works, write your initial version of code to implement when ROT_DIR on dip switch is 0, it rotates Clockwise, while if its 1, then it rotates counter clockwise. If ROT_SPD on dip switch is 0 then it rotates at 22.5 degrees , if it's 1 then it rotates at 180 degrees.

3. Experiment System Specification :

***Lab hardware photo*:**

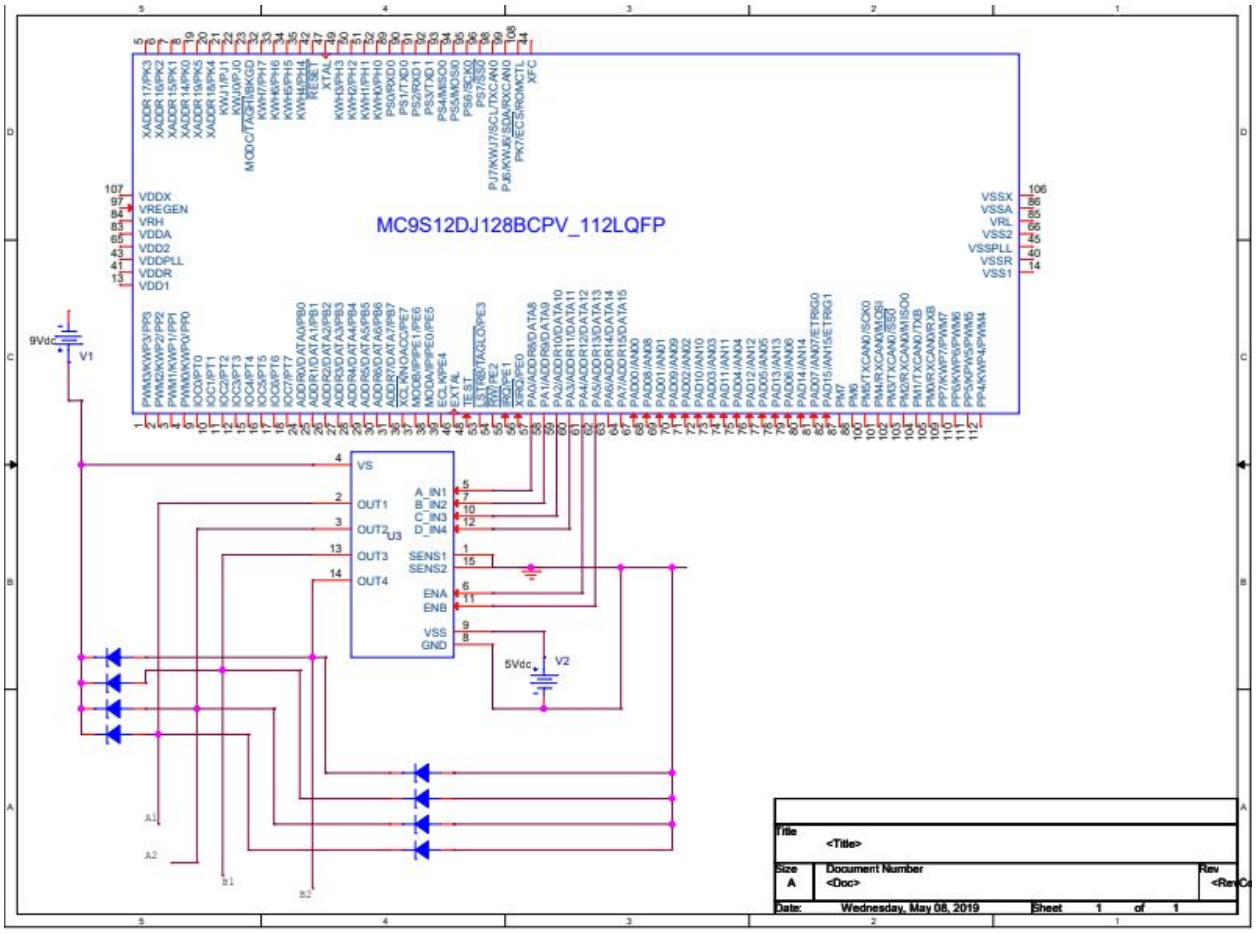


Flowchart; 9s12 Microcontroller



4. Hardware Design :

9S12 Controller;



5. Software Design:

```
#include <hidef.h> /* common defines and macros */
#include <mc9s12dg256.h>
#include <stdio.h>

#define delay_fast 3000
#define delay_slow 6000
unsigned long i;

void init(){
  DDRA = 0xff;
  PORTA = 0x30;
  DDRB = 0x00;
  PORTB = 0xff;
}

void Delay(unsigned long Speed){
  for(i = 0; i < Speed ;i++);
  return;
```

```
}

void main(void) {
  unsigned char FR1;
  unsigned char FR2;
  unsigned long Speed;
  init();
  while (1) {
    FR1 = PORTB & 0x01;
    Speed = (PORTB & 0x02)?delay_fast:delay_slow;
    if (FR2 != FR1) {
      PORTA = 0x35;
    }
    if (FR1) {
      PORTA = 0x36;
      Delay(Speed);
      PORTA = 0x35;
      Delay(Speed);
      PORTA = 0x39;
      Delay(Speed);
      PORTA = 0x3A;
      Delay(Speed);
      FR2 = FR1;
      FR1 = PORTB & 0x01;
    }else{
      PORTA = 0x36;
      Delay(Speed);
      PORTA = 0x3A;
      Delay(Speed);
      PORTA = 0x39;
      Delay(Speed);
      PORTA = 0x35;
      Delay(Speed);
      FR2 = FR1;
      FR1 = PORTB & 0x01;
    }
  }
}
```

- ***High Level Description***
    1) Set PORTA to be output for stepper motor and PORTB for Dip switch input.
    2) Check the inputs from the Dip switch, and check if the ROT_SPD is on or off, if its on, then the delay time would be less (in order to rotate it 180 degrees) if its off, then it would be more (in order to rotate it at a slower speed at 22.5 degrees).
    3) Check ROT_Dir in order to determine the rotation speed. If its 1, then its Counter Clockwise, but if it's 0 then it would be clockwise direction.
    4) Set values for PORTA for both clockwise and counterclockwise.

6. Problems Encountered:

There were no major problems with this lab. The lab went pretty smooth, The minor problems like a wrong direction of the diode that caused the motor to 'spin-in-place' and warm battery.

8.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.**

   Ans) No. In human eyes the motor starts and turns instantaneously, but it is not, transient response always takes time.

   The magnetic field required to spin the motor generated by the electricity in the motor is instantaneous, but due to the nature of all mechanical parts, it takes time for energy to transfer for the motor to start to spin.

   Same with switch direction, magnetic field generated by changing the current flow is instantaneous, but it takes time for the mechanical parts to 'react' to the force.

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.

Step	A1	A2	B1	B2	C1	C2	D1	D2
1	+	-	-	-	-	-	-	-
2	-	-	-	-	-	-	+	-
3	-	-	+	-	-	-	-	-
4	-	-	-	-	+	-	-	-
5	-	+	-	-	-	-	-	-
6	-	-	-	-	-	-	-	+
7	-	-	-	+	-	-	-	-
8	-	-	-	-	+	-	-	-

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

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.

Step	A1	A2	B1	B2	C1	C2	D1	D2
1	+	-	-	-	-	-	-	-
2	+	-	-	-	-	-	+	-
3	-	-	-	-	-	-	+	-
4	-	-	+	-	-	-	+	-
5	-	-	+	-	-	-	-	-
6	-	-	+	-	+	-	-	-
7	-	-	-	-	+	-	-	-
8	-	+	-	-	+	-	-	-
9	-	+	-	-	-	-	-	-
10	-	+	-	-	-	-	-	+
11	-	-	-	-	-	-	-	+
12	-	-	-	+	-	-	-	+
13	-	-	-	+	-	-	-	-
14	-	-	-	+	-	+	-	-
15	-	-	-	-	-	+	-	-
16	+	-	-	-	-	+	-	-

9. Conclusion:

This lab was overall really interesting , as we got to work with the 9S12 microcontroller and the stepper motor and their integration of it. It was not only limited to using the controller itself , we also got to

experience the L298N board which helps in transferring of data and powering the stepper motor itself. In this lab, both Leya and Tanish worked on both hardware and software designs.