

Ministerul Educației și Tineretului al Republicii Moldova

Universitatea Tehnică a Moldovei

Departament „Informatica aplicată”

RAPORT

Lucrarea de laborator nr.4

A efectuat:

St. gr. FAF 141

N. Botnari

A verificat:

Lect . sup

A. Bragarenco

Chișinău 2016

Topic: Actuators. Generate PWN signal.

Tasks: Develop an application that will read data from a button press and rotate the motor either clockwise or counterclockwise.

Theory

L293 Driver

The L293 and L293D are quadruple high-current half- H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

DC motors

The DC motor uses a combination of schematic and programmatic modelling techniques. The schematic model is shown below, and demonstrates rather nicely how electrical circuits may be used to simulate mechanical phenomena.

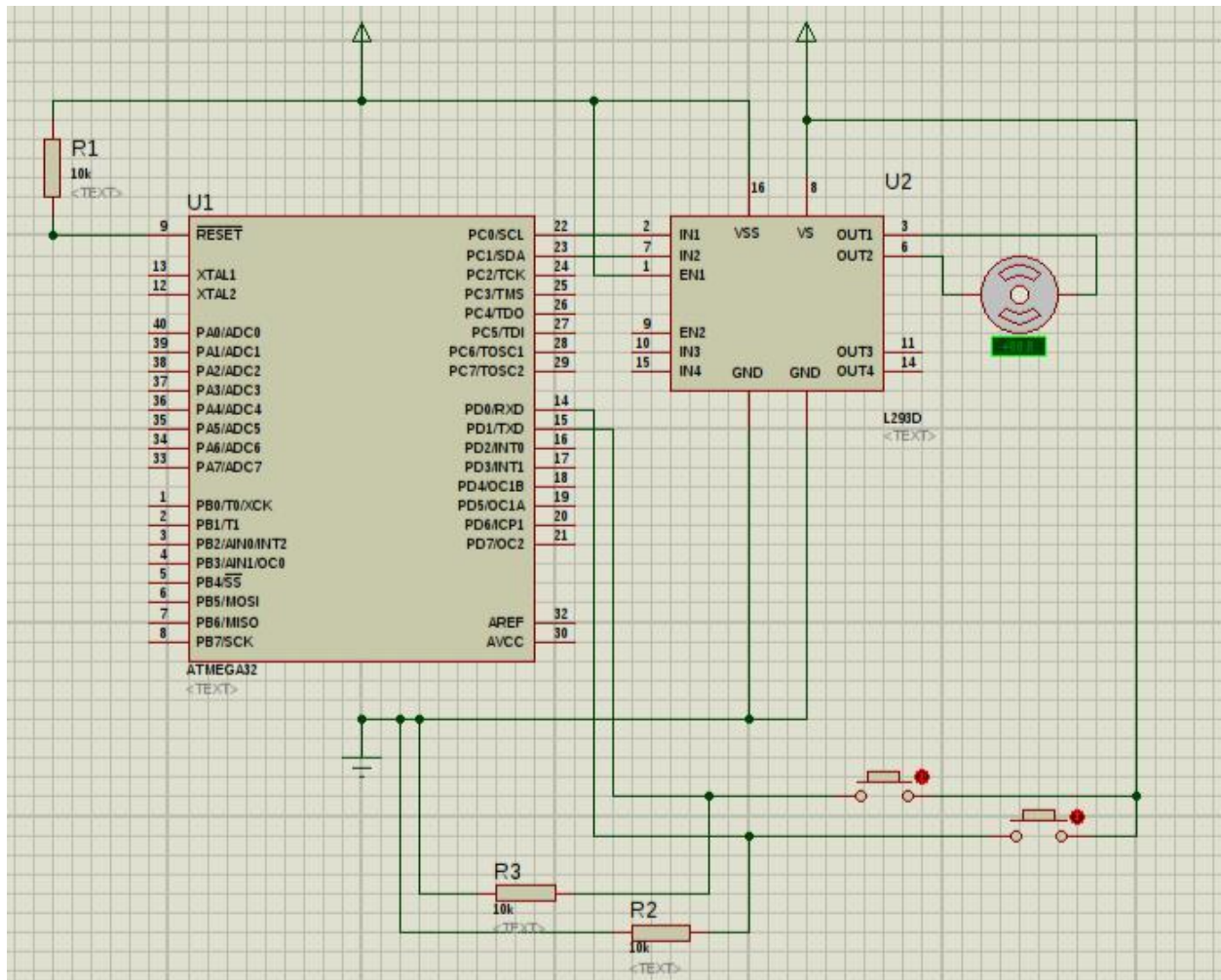
Solution

L293 driver which initializes the Motor

```
void L293_init() {  
    DDRC = 0xFF; //PORTB as Output  
}  
  
void L293_clockwise()  
{  
    PORTC = 0x02; //00000010  
}  
  
void L293_antiClockwise()  
{  
    //Rotates Motor in Anticlockwise PORTC =  
    0x01; //00000001  
}  
  
void L293_stop()  
{  
    PORTC = 0x00; //00000000  
}
```

Processed signal goes to L293 driver, which can make switch polarity and make motor move.

In1	In2	meaning
0	0	Free
0	1	Clockwise
1	0	Anticlockwise
1	~	Stop



Conclusion:

I learned how to control a motor using PWN signal, how to make it move in 2 directions using L293 driver.

Appendix

Button.h

```
#ifndef BUTTON_H_
#define BUTTON_H_
#include <avr/io.h>

void initButtonOne();
void initButtonTwo();
int isButtonOnePressed();
int isButtonTwoPressed();

#endif /* BUTTON_H_ */
```

Button.c

```
#include "button.h"

void initButtonOne() {
    DDRD &= ~(1 << PORTD0) ;
}

void initButtonTwo() {
    DDRD &= ~(1 << PORTD1) ;
}

int isButtonOnePressed() {
    return PIND & (1<<PORTD0);
}

int isButtonTwoPressed() {
    return PIND & (1<<PORTD1);
}
```

L293.h

```
#ifndef LAB4_SRC_L293_H_
#define LAB4_SRC_L293_H_

void L293_init();
void L293_antiClockwise();
void L293_clockwise();
void L293_stop();
```

```
void L293_free();
```

```
#endif /* LAB4_SRC_L293_H_ */
```

L293.c

```
#include <avr/io.h>
```

```
#include "L293.h"
```

```
void L293_init() {  
    DDRC = 0xFF; //PORTB as Output  
}
```

```
void L293_clockwise()  
{  
    PORTC = 0x02; //00000010  
}
```

```
void L293_antiClockwise()  
{  
    //Rotates Motor in Antilockwise  
    PORTC = 0x01; //00000001  
}
```

```
void L293_stop()  
{  
    PORTC = 0x00; //00000000  
}
```

main.c

```
#define F_CPU 8000000ul  
#include <avr/io.h>  
#include <util/delay.h>  
#include <avr/interrupt.h>
```

```
#include "L293.h"  
#include "button.h"
```

```
int main(void)  
{  
    L293_init();  
    initButtonOne();  
    initButtonTwo();  
  
    while(1) {
```

```
        if(isButtonOnePressed()) {  
            L293_clockwise();  
        } else if(isButtonTwoPressed())  
{  
            L293_antiClockwise();  
        } else {  
            L293_stop();  
        }  
    }  
}
```