Ministerul Educaţiei şi Tineretului al Republicii Moldova

Universitatea Tehnică a Moldovei

Departament „Informatica aplicată”

RAPORT

Lucrarea de laborator nr.4

LA DISCIPLINA”Programarea aplicațiilor incorporate și independente de platformă ”

A efectuat:

St. gr. FAF 141 Schidu Luca

A verificat:

Lect . supp A. Bragrarenco

Chişinău 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.

Theoretical notes

PWN signal

Pulse-width modulation (PWM), or pulse-duration modulation (PDM), is a modulation technique used to encode a message into a pulsing signal. Although this modulation technique can be used to encode information for transmission, its main use is to allow the control of the power supplied to electrical devices, especially to inertial loads such as motors. In addition, PWM is one of the two principal algorithms used in photovoltaic solar battery chargers, the other being maximum power point tracking.

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.

Working process

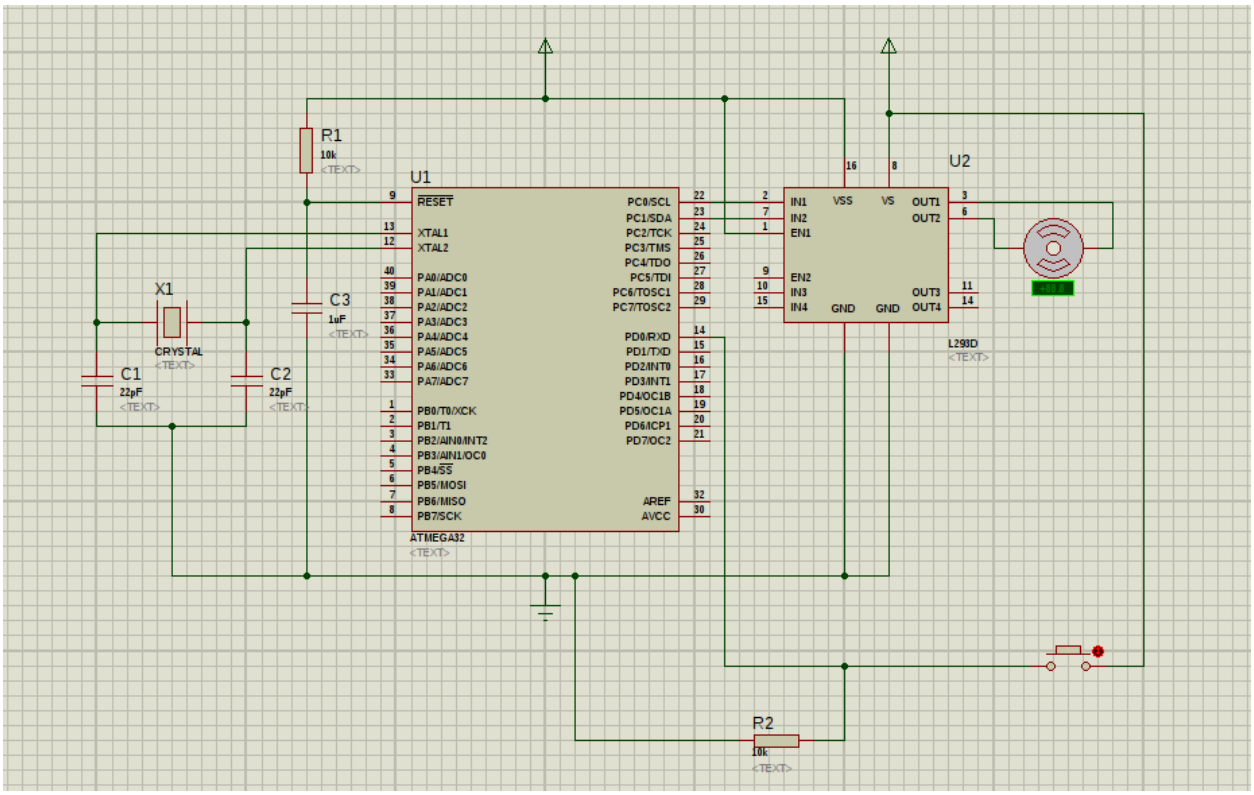
I create the driver for L293 which initializes the Motor. Also created the method that drives the motor

```
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
}
```

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	Anti-clockwise
1	1	Stop

Results:



Appendix

```
/*
 * L293.c
 *
 *  Created on: Jan 17, 2017
 *      Author: lschidu
 */

#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
}
```

```
/*
 * L293.c
 *
 *  Created on: Jan 17, 2017
 *      Author: lschidu
 */

#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
}
```

```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_antiClockwise();
		} else {
			L293_clockwise();
		}
	  }
}
```
