**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САУ**

**Лабораторная работа №3**

**по дисциплине «МПУ»**

**Вариант 7**

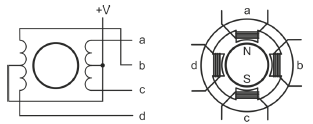
|  |  |  |
| --- | --- | --- |
| Студенты гр. 6 |  | Раскин В.Ю. |
| Преподаватель |  |  |

Санкт-Петербург

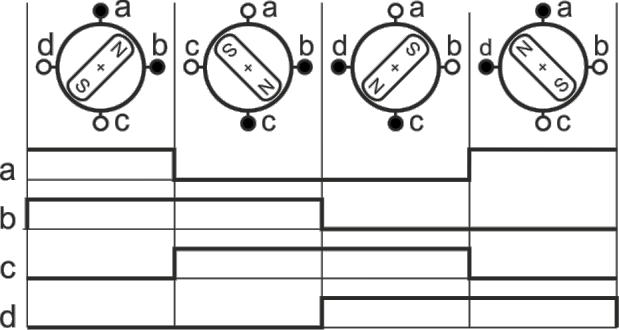
2018

**Цель работы**. Изучение принципов программного управления шаговым двигателем с использованием параллельного порта МК.

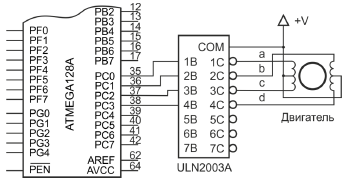
**Задание.** Составьте программу управления шаговым двигателем для реализации заданного режима управления и графика изменения скорости вращения.



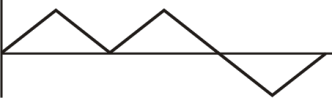
*Рис.1 Конструкция и схема включения обмоток ШД*



*Рис.2 Схема коммутации обмоток для двухфазного полушагового режима ШД*

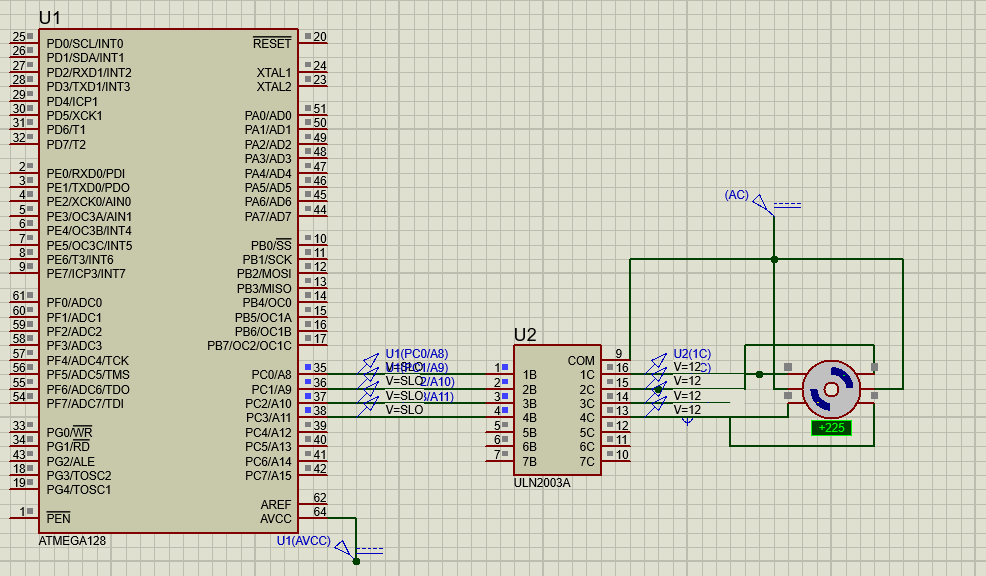
**

*Рис.3 Схема подключения ШД*



*Рис.4 График движения ШД*

*Proteus*

**

*Блок схема:*

**

**Код программы.**

#include <mega128a.h>

#include <delay.h>

#define Time\_1 100 // максимальное время задержки в мсек

#define Time\_step 500 //время через которое изменяется задержка в мсек

#define STEP\_ENGINE 5 //изменение времени задержки в мсек

#define time\_pause 3000 // время между повторением алгоритма (мс)

#define Time\_2 50 // минимальное время задержки в мсек

void main(void)

{

int i=0;

int time ; //время задержки

int step\_engine[4] = { //0b0000DCBA

0b00000011,

0b00000110,

0b00001100,

0b00001001

};

int counter = 0;

DDRC=0x0F;

while (1)

{

time = Time\_1;

//------прямое включение направления--------------

while(time >= Time\_2 ) //разгон двигателя

{

for(i=0;i <= 3; i++)//

{

PORTC |= step\_engine[i]; // сформировать фронт импульса

delay\_ms(time); // задать длительность импульса

PORTC &= ~step\_engine[i];

}

if(counter >= Time\_step/(4\*time) )

{

time -= STEP\_ENGINE;

counter = 0;

}

counter ++;

}

time = Time\_2 ;

while(time <= Time\_1 ) //замедление двигателя

{

for(i=0;i <= 3; i++)//

{

PORTC |= step\_engine[i]; // сформировать фронт импульса

delay\_ms(time); // задать длительность импульса

PORTC &= ~step\_engine[i];

}

if(counter >= Time\_step/(4\*time))

{

time += STEP\_ENGINE;

counter = 0;

}

counter ++;

}

//------------------------------------------------------

time = Time\_1;

//------прямое включение направления--------------

while(time >= Time\_2 ) //разгон двигателя

{

for(i=0;i <= 3; i++)//

{

PORTC |= step\_engine[i]; // сформировать фронт импульса

delay\_ms(time); // задать длительность импульса

PORTC &= ~step\_engine[i];

}

if(counter >= Time\_step/(4\*time) )

{

time -= STEP\_ENGINE;

counter = 0;

}

counter ++;

}

time = Time\_2 ;

while(time <= Time\_1 ) //замедление двигателя

{

for(i=0;i <= 3; i++)//

{

PORTC |= step\_engine[i]; // сформировать фронт импульса

delay\_ms(time); // задать длительность импульса

PORTC &= ~step\_engine[i];

}

if(counter >= Time\_step/(4\*time))

{

time += STEP\_ENGINE;

counter = 0;

}

counter ++;

}

//------------------------------------------------------

time = Time\_1;

//----------обратное включение направления----------------------

while(time >= Time\_2 ) //разгон двигателя

{

for(i = 3 ;i >= 0; i--)//

{

PORTC |= step\_engine[i]; // сформировать фронт импульса

delay\_ms(time); // задать длительность импульса

PORTC &= ~step\_engine[i];

}

if(counter >= Time\_step/(4\*time) )

{

time -= STEP\_ENGINE;

counter = 0;

}

counter ++;

}

time = Time\_2 ;

while(time <= Time\_1 ) //замедление двигателя

{

for(i = 3 ;i >= 0; i--)//

{

PORTC |= step\_engine[i]; // сформировать фронт импульса

delay\_ms(time); // задать длительность импульса

PORTC &= ~step\_engine[i];

}

if(counter >= Time\_step/(4\*time) )

{

time += STEP\_ENGINE;

counter = 0;

}

counter ++;

}

//---------------------------------------------------------------------

delay\_ms(time\_pause);// время между переключением направления двигателя

}

}

**Вывод**: Изучили принципы программного управления шаговым двигателем с использованием параллельного порта МК.