

Done at Home

Программирование микроконтроллеров avr

Подключение шагового двигателя (28BYJ_48) к avr

admin | 26.09.2014

0 Comment

Подключение шагового двигателя к avr

Шаговый двигатель:

Шаговый двигатель— это синхронный бесщёточный электродвигатель с несколькими обмотками, в котором ток, подаваемый в одну из обмоток статора, вызывает фиксацию ротора. Последовательная активация обмоток двигателя вызывает дискретные угловые перемещения (шаги) ротора.

Преимущества: Главное преимущество шаговых приводов – точность. При подаче потенциалов на обмотки шаговый двигатель повернется строго на определенный угол. К приятным моментам можно отнести стоимость шаговых приводов, в среднем в 1,5-2 раза дешевле сервоприводов. Шаговый привод, как недорогая альтернатива сервоприводу, наилучшим образом подходит для автоматизации отдельных узлов и систем, где не требуется высокая динамика.

Недостатки: Возможность «проскальзывания» ротора — наиболее известная проблема этих двигателей. Это может произойти при превышении нагрузки на валу, при неверной настройке управляющей программы (например, ускорение старта или торможения не адекватно перемещаемой массе), при приближении скорости вращения к резонансной. Электрически это никак не может быть замечено, поэтому контроллер все последующие движения будет производить с ошибкой. Для ответственных применений устанавливают датчики обратной связи (контроль вращения или перемещения), но такие датчики достаточно дороги. Наличие датчика позволяет обнаружить проблему, но автоматически скомпенсировать её без остановки производственной программы возможно только в очень редких случаях. Чтобы избежать проскальзывания ротора, как один из способов, можно увеличить мощность двигателя.

Описание шагового двигателя 28BYJ_48 :



Номинальное напряжение : 5В

Редуктор : 1/64

Шаг угол : 5.625° /64

Номинальная частота : 100 Гц

DC сопротивление обмоток : 50 Ом±7%(25°C)

Крутящий момент : >34.3 мН.м(120Hz)

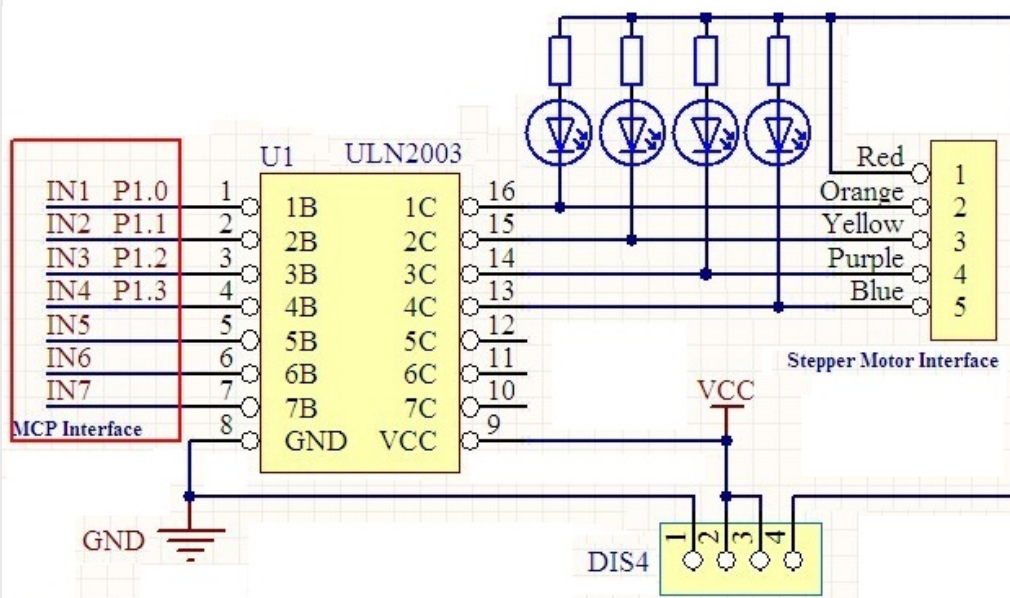
Момент трения : 600-1200 г/см

Номинальная тяга : 300 г/см

Класс изоляции : А

Вес : 30 г

Модуль драйвера шагового двигателя на базе ULN2003:



Микросхема ULN2003 по сути своей является набором мощных составных ключей для применения в цепях индуктивных нагрузок. Может быть применена для управления нагрузкой значительной мощности, включая электромагнитные реле двигателей постоянного тока, электромагнитные клапаны, в схемах управления различными шаговыми двигателями и другие.

Микросхемы ULN2003 это транзисторная сборка Дарлингтона с выходными ключами повышенной

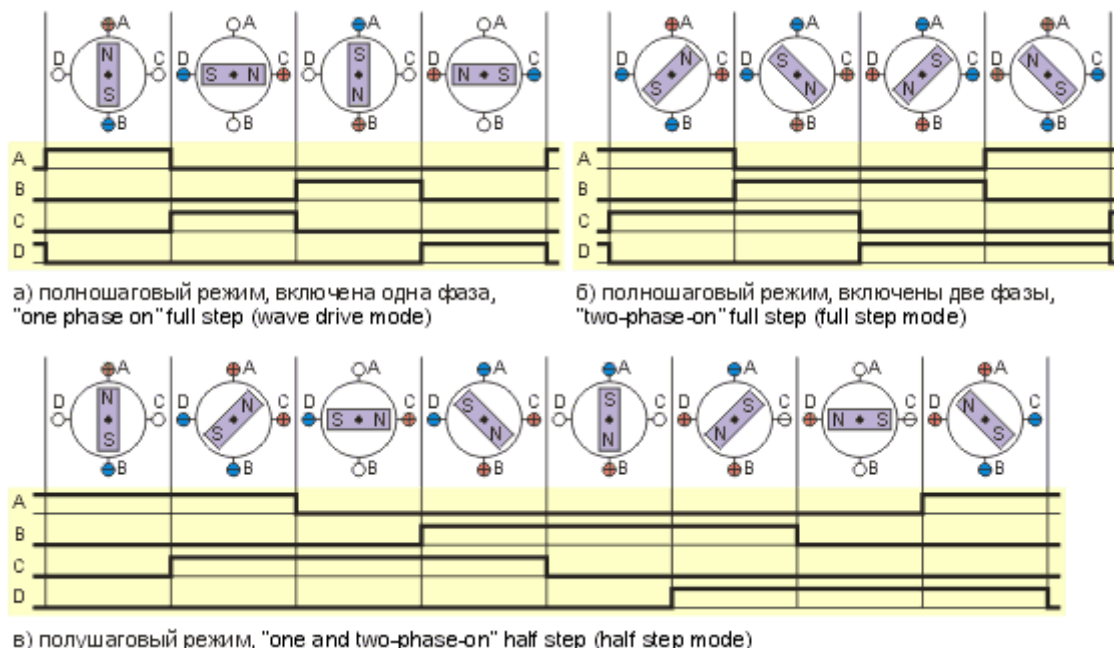
мощности, имеющая на выходах защитные диоды, которые предназначены для защиты управляющих электрических цепей от обратного выброса напряжения от индуктивной нагрузки.

Схема подключения:

Любые 4 ножки микроконтроллера)).

Алгоритм управления ([источник](#)):

Существует несколько способов управления фазами шагового двигателя.



Первый способ обеспечивается попеременной коммутации фаз, при этом они не перекрываются, в один момент времени включена только одна фаза (рис 8а). Этот способ называют "one phase on" full step или wave drive mode. Точки равновесия ротора для каждого шага совпадают с «естественными» точками равновесия ротора у незапитанного двигателя. Недостатком этого способа управления является то, что для биполярного двигателя в один и тот же момент времени используется 50% обмоток, а для униполярного – только 25%. Это означает, что в таком режиме не может быть получен полный момент.

Второй способ – управление фазами с перекрытием: две фазы включены в одно и то же время. Его называют "two-phase-on" full step или просто full step mode. При этом способе управления ротор фиксируется в промежуточных позициях между полюсами статора (рис. 8б) и обеспечивается примерно на 40% больший момент, чем в случае одной включенной фазы. Этот способ управления обеспечивает такой же угол шага, как и первый способ, но положение точек равновесия ротора смещено на пол-шага.

Третий способ является комбинацией первых двух и называется полушаговым режимом, "one and two-phase-on" half step или просто half step mode, когда двигатель делает шаг в половину основного. Этот метод управления достаточно распространен, так как двигатель с меньшим шагом стоит дороже и очень заманчиво получить от 100-шагового двигателя 200 шагов на оборот. Каждый второй шаг запитана лишь одна фаза, а в остальных случаях запитаны две (рис. 8в). В результате угловое перемещение ротора составляет половину угла шага для первых двух способов управления. Кроме уменьшения размера шага этот способ управления позволяет частично избавиться от явления резонанса. Полушаговый режим обычно не позволяет получить полный момент, хотя наиболее совершенные драйверы реализуют модифицированный полушаговый режим, в котором двигатель

обеспечивает практически полный момент, при этом рассеиваемая мощность не превышает номинальной.

Еще один способ управления называется микрошаговым режимом или micro stepping mode. При этом способе управления ток в фазах нужно менять небольшими шагами, обеспечивая таким образом дробление половинного шага на еще меньшие микрошаги. Когда одновременно включены две фазы, но их токи не равны, то положение равновесия ротора будет лежать не в середине шага, а в другом месте, определяемом соотношением токов фаз. Меняя это соотношение, можно обеспечить некоторое количество микрошагов внутри одного шага. Кроме увеличения разрешающей способности, микрошаговый режим имеет и другие преимущества, которые будут описаны ниже. Вместе с тем, для реализации микрошагового режима требуются значительно более сложные драйверы, позволяющие задавать ток в обмотках с необходимой дискретностью. Полушаговый режим является частным случаем микрошагового режима, но он не требует формирования ступенчатого тока питания катушек, поэтому часто реализуется.

Ссылки на комплектующие:

Микроконтроллер: АТмега32 ([ссылка](#))

Макетная плата: ([ссылка](#))

Шаговый двигатель (28BYJ_48): ([ссылка](#))

Модуль драйвера шагового двигателя (ULN2003): ([ссылка](#))

Питание для макетной платы: ([ссылка](#))

Блок преобразователь 220v->12v: ([ссылка](#))

Код ([ссылка](#)):

Библиотека: Mot.h

```
1  #define Mot_DDR DDRB
2  #define Mot_PORT PORTB
3  #define Mot_Pin_1 0
4  #define Mot_Pin_2 1
5  #define Mot_Pin_3 2
6  #define Mot_Pin_4 3
7
8  // макрос уст "1" на ножке порта (x-порт, y-ножка)
9  #define CLEARBIT(x,y) x&=~(1<<y)
10 // макрос уст "0" на ножке порта (x-порт, y-ножка)
11 #define SETBIT(x,y)  x|=1<<y
12
13 void Mot_Init (void)
14 {
15     //настраиваем ножки на выход
16     Mot_DDR |= (1<<Mot_Pin_1)|(1<<Mot_Pin_2)|(1<<Mot_Pin_3)|(1<<Mot_Pin_4);
17 }
18
19 void Mot_Step (unsigned char Step)
20 {
21     Step = Step % 4;
22
23     switch (Step) {
24     case 0: // 1000
25         SETBIT(Mot_PORT, Mot_Pin_1);
26         CLEARBIT(Mot_PORT, Mot_Pin_2);
27         CLEARBIT(Mot_PORT, Mot_Pin_3);
28         CLEARBIT(Mot_PORT, Mot_Pin_4);
29         break;
30     case 1: // 0100
```

```
31     CLEARBIT(Mot_PORT,Mot_Pin_1);
32     SETBIT(Mot_PORT,Mot_Pin_2);
33     CLEARBIT(Mot_PORT,Mot_Pin_3);
34     CLEARBIT(Mot_PORT,Mot_Pin_4);
35     break;
36 case 2:    // 0010
37     CLEARBIT(Mot_PORT,Mot_Pin_1);
38     CLEARBIT(Mot_PORT,Mot_Pin_2);
39     SETBIT(Mot_PORT,Mot_Pin_3);
40     CLEARBIT(Mot_PORT,Mot_Pin_4);
41     break;
42 case 3:    // 0001
43     CLEARBIT(Mot_PORT,Mot_Pin_1);
44     CLEARBIT(Mot_PORT,Mot_Pin_2);
45     CLEARBIT(Mot_PORT,Mot_Pin_3);
46     SETBIT(Mot_PORT,Mot_Pin_4);
47     break;
48 }
49 }
```

Код: 28BYJ_48.c

```
1  #define F_CPU 1000000UL
2  #include <avr/io.h>
3  #include <util/delay.h>
4  #include "Mot.h"
5
6
7  int main(void)
8  {
9      Mot_Init();
10
11     char flag =1;
12
13     unsigned char i=0;
14     while(1)
15     {
16         Mot_Step (i);
17         _delay_ms(10);
18
19         if(flag)
20             i+=1;
21         else
22             i-=1;
23
24         if (i==0)
25             flag=1;
26         if (i==50)
27             flag=0;
28     }
29 }
```

Подключение шагового двигателя (...)



Рубрика: Все посты AVR Подключаем к AVR