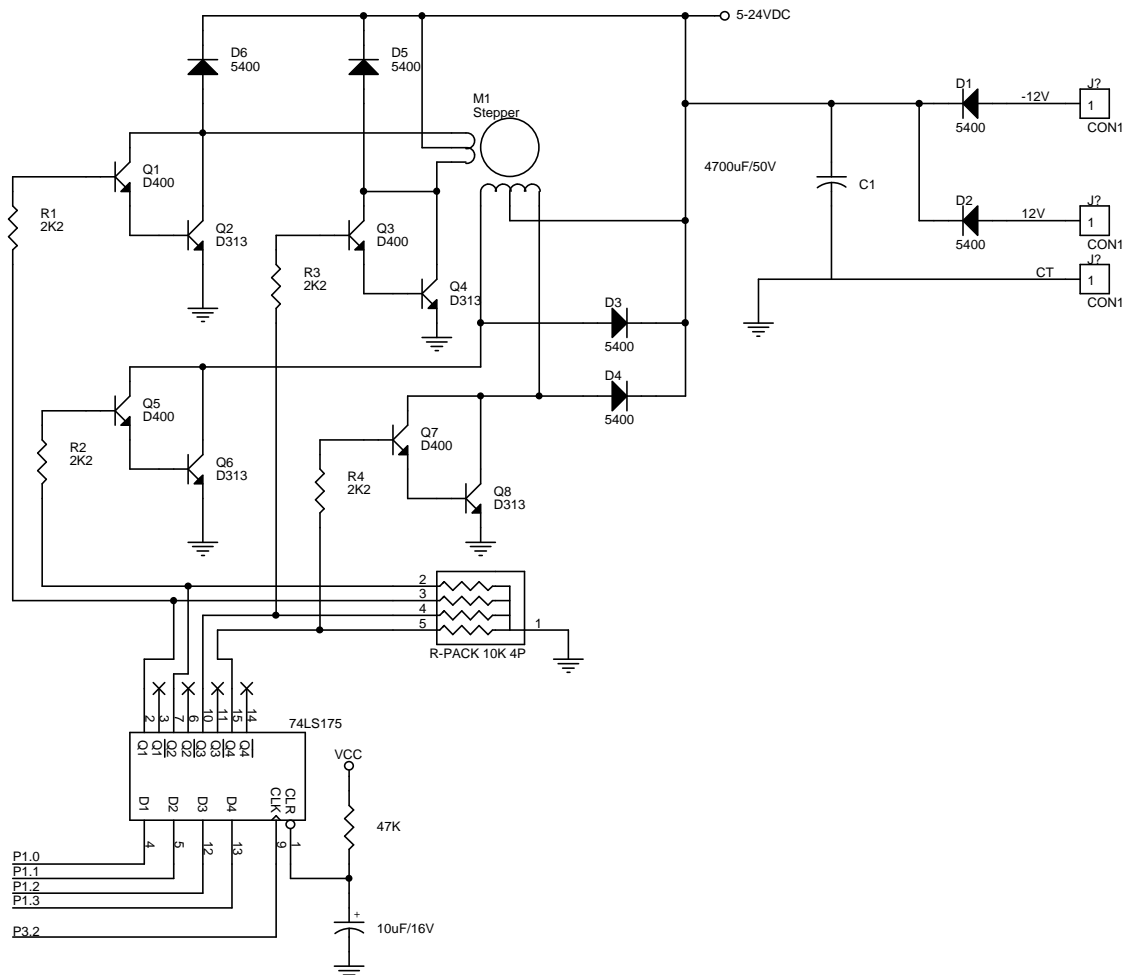


Kendali Motor Stepper dengan Modul DST-52 dan STP-06

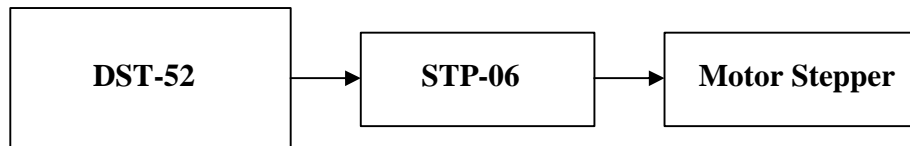
Aplikasi kali ini membahas pengaturan pergerakan motor stepper. Ada tiga parameter pergerakan yang bisa diatur, yaitu: arah, jumlah step dan kecepatan.



Gambar 1
Rangkaian Pengendali Stepper

Gambar 1 menunjukkan rangkaian pengendali stepper yang dirancang dengan menggunakan susunan transistor dengan arus kolektor maksimum 5A sehingga modul ini mampu mengendalikan motor stepper dengan arus 5A pula.

Blok diagram pengendali motor stepper bisa dilihat pada gambar 1. Untuk menggerakkan motor stepper satu step kekanan, maka P1 diberi data secara berurutan: xxxx0111, xxxx1011, xxxx1101 dan xxxx1110. Sedangkan untuk menggerakkan motor stepper satu step ke kiri, maka P1 harus diberi data dengan urutan terbalik (xxxx1110, xxxx1101, xxxx1011 dan xxxx0111). Yang perlu diperhatikan adalah antara pemberian satu data dengan data berikutnya perlu diberi delay yang disesuaikan dengan karakteristik respon motor stepper yang bersangkutan. Pada aplikasi ini dipakai motor stepper dengan karakteristik delay antar perintah sekitar 1mS, dan pergerakan motor adalah 1.8° per step.



Gambar 2
Blok Diagram Sistem

Jadi untuk menggerakkan motor satu step diperlukan waktu sekitar $4 \times 1\text{mS} = 4\text{mS}$. Perhitungan kecepatan motor bisa dilakukan sebagai berikut:

1 step setara dengan putaran 1.8° , membutuhkan waktu 4mS .

1 putaran ($360^\circ = 2\pi \text{ rad}$) setara dengan 200 step, sehingga dibutuhkan waktu $200 \times 4\text{mS} = 0.8 \text{ s}$ untuk sekali putar.

Dengan demikian kecepatan putar maksimalnya adalah: $360^\circ/0,8\text{S} = 450^\circ/\text{S}$ atau $7,85 \text{ rad/S}$.

Untuk mengecilkan putaran, perlu diberi delay. Besarnya delay yang digunakan bisa diturunkan sbb:

$$v = 360 / (0.8 + 200 \times \text{Delay})$$

$$\begin{aligned} \text{Delay} &= \{(360/v) - 0.8\} / 200 \text{ S} \\ &= 1000 \times \{(360/v) - 0.8\} / 200 \text{ mS} \\ &= (1800/v) - 4 \text{ mS} \end{aligned} \quad (1)$$

dimana v adalah kecepatan yang diinginkan dalam $^\circ/\text{S}$.

Satu misal kita menginginkan kecepatan $90^\circ/\text{S}$ maka delay yang diperlukan adalah :

$$\begin{aligned} \text{Delay} &= 1800/90 - 4 \\ &= 16 \text{ mS} \end{aligned}$$

Potongan program berikut adalah rutin untuk mengatur pergerakan motor stepper.

Potongan program 1 : Rutin PutarMotor

```

void PutarMotor(unsigned char arah,unsigned char step,unsigned int
Kecepatan){
  unsigned char i,j,x;
  x = KonversiKecepatan(Kecepatan);
  switch (arah){
  case '*':
  
```

```

        for (i=0;i<step;i++){
            for (j=0;j<4;j++){
                P1 = Tabel_Data[i];
                Delay_1mS();
            }
            Delay(x);
        }
        break;
    case '#':
        for (i=0;i<step;i++){
            for (j=4;j>0;j--){
                P1 = Tabel_Data[i];
                Delay_1mS();
            }
            Delay(x);
        }
        break;
    }
}

```

Potongan program 2 : Rutin Pengkonversian kecepatan menjadi delay

```

unsigned int KonversiKecepatan(unsigned int Kecepatan){
    return (1800/Kecepatan - 4);
}

```

Pada rutin PutarMotor, parameter yang diinputkan adalah arah, step dan kecepatan. Disini arah diwakili oleh sebuah karakter, dimana karakter '*' mewakili arah kanan dan karakter '#' mewakili arah kiri. Tentu saja karakter yang digunakan nantinya tergantung pada pemakai.

Pertama yang dilakukan dalam program diatas adalah mengkonversikan nilai kecepatan menjadi lama delay dengan menggunakan persamaan 1. Setelah itu dilihat arah pergerakan. Kemudian, sebanyak step yang diminta lakukan:

- - Ambil satu persatu tabel data yang harus dioutputkan ke port 1. Data ini dideklarasikan dalam array konstanta Tabel_Data:

```

unsigned char code Tabel_Data[] = {0x07,0x0b,0x0d,0x0e};

```

- - Antara pengoutputan satu data dengan data berikutnya diberi delay 1mS
- - Setiap selesai satu step beri delay yang sesuai untuk kecepatan yang diminta

Perbedaan arah kanan dan kiri terjadi pada pengaksesan Tabel_Data. Pada pergerakan kanan, pengaksesan Tabel_Data dimulai dari Tabel_Data[0] dan seterusnya sampai Tabel_Data[3]. Sedangkan pada pergerakan kiri, pengaksesan Tabel_Data dimulai dari Tabel_Data[3] menurun sampai Tabel_Data[0].

Aplikasi dan program lengkap dari artikel ini dapat didownload di www.delta-electronic.com bagian application note.