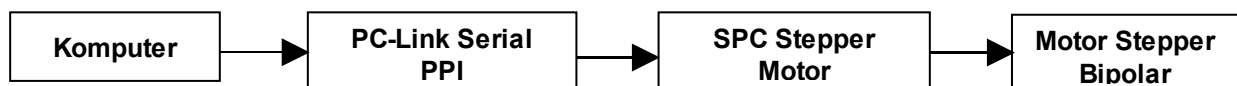


Application note sebelumnya yaitu AN147, telah dibahas cara penggunaan PC Link Serial PPI untuk mengatur gerakan motor DC dengan bantuan modul SPC DC Motor. Pada aplikasi kali ini akan membahas cara mengatur gerakan motor stepper dengan bantuan modul SPC Stepper Motor. Modul SPC Stepper Motor dihubungkan ke 7 pin I/O PC-Link Serial PPI dan akan menggunakan 2 macam komunikasi ke PC-Link Serial PPI yaitu komunikasi secara paralel dan komunikasi secara I²C.

Aplikasi ini memerlukan modul sebagai berikut :

- 1 PC-Link Serial PPI
- 1 SPC Stepper Motor
- Motor Stepper Bipolar

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 1
Blok Diagram AN150

Hubungan antara modul-modul tersebut adalah sebagai berikut :

PC-Link Serial PPI	SPC Stepper Motor
VCC(J1)	VCC (J7)
GND(J1)	GND(J7)
Port 1.1* (J3)	SCL(J1)
Port 1.0* (J3)	SDA(J1)
Port 1.2* (J3)	S1 (J4)
Port 1.3* (J3)	S2 (J4)
Port 1.4* (J3)	S3 (J4)
Port 1.5* (J3)	S4 (J4)
Port 1.6* (J3)	RST (J4)

*) Pin ini tidak mutlak dan dapat diganti pin lain dengan cara mengubah program

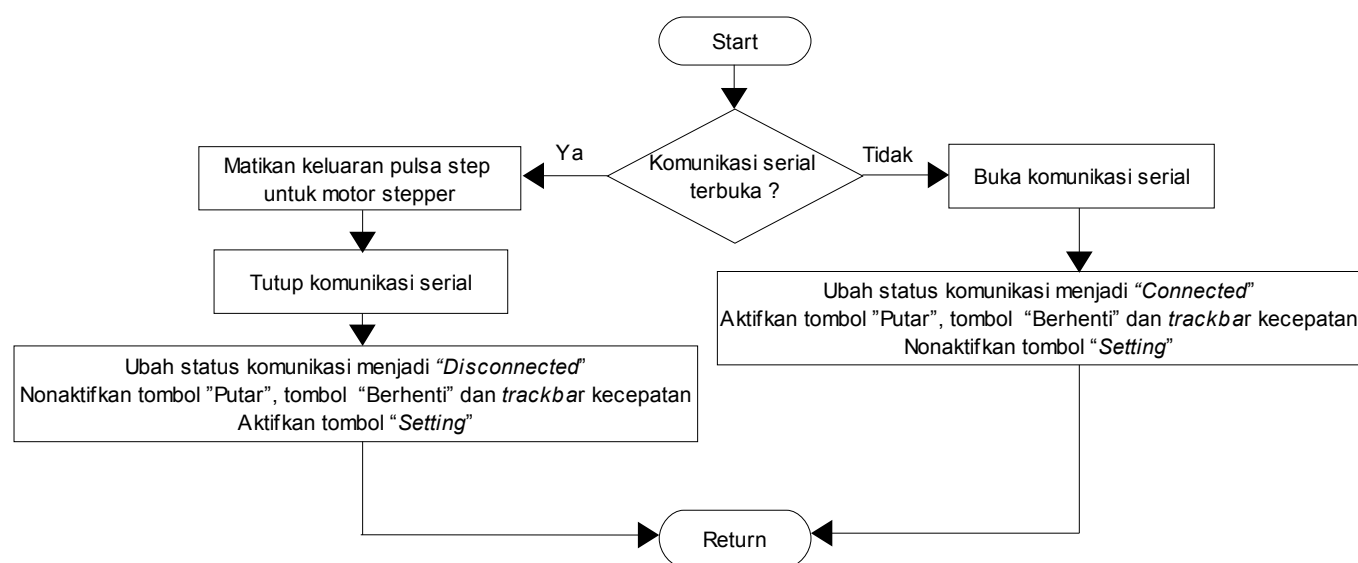
Tabel 1
Hubungan PC-Link Serial PPI dengan SPC Stepper Motor

SPC Stepper Motor	Motor Stepper
Aout	A (pin step ke-1)
Bout	B (pin step ke-2)
Cout	C (pin step ke-3)
Dout	D (pin step ke-4)

Tabel 2
Hubungan SPC Stepper Motor dengan Motor Stepper Bipolar

Alamat I²C yang dipakai SPC Stepper Motor pada aplikasi ini adalah 000b. *Jumper J2* dipasang untuk memberi resistor *pull up* pada jalur SDA dan SCL pada komunikasi I²C. Setelah semua rangkaian dan sumber tegangan terhubung dengan benar, bukalah program **stepper.exe**

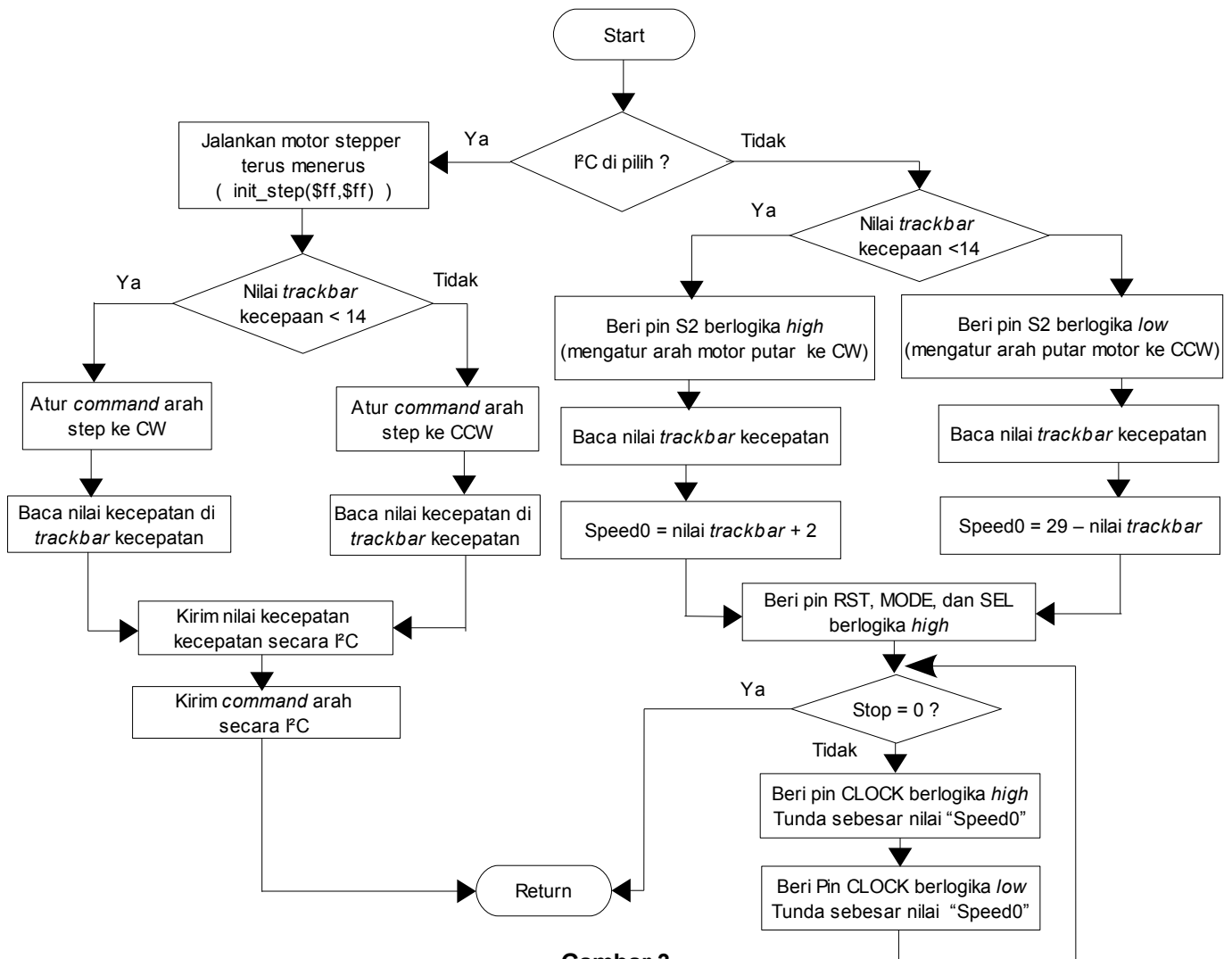
Flowchart program yang digunakan adalah sebagai berikut:



Gambar 2
Flowchart Tombol "Connect/Disconnect"

Cara kerja dari program berdasarkan *flowchart* Tombol "**Connect/Disconnect**" adalah sebagai berikut :

1. Pada keadaan awal, program akan menunggu tombol "Connect/Disconnect" ditekan. Jika tombol ditekan dan komunikasi serial belum terbuka, maka program akan membuka komunikasi serial dan mengaktifkan tombol "Putar", tombol "Berhenti", mengubah status komunikasi menjadi "Connected" , serta menonaktifkan tombol "Setting Port".
2. Jika keadaan komunikasi serial sebelumnya sudah terbuka, maka program akan menghentikan pulsa step untuk motor stepper dengan mengirim perintah `init_step (0,0)`. Selain itu program akan menonaktifkan tombol "Putar", tombol "Berhenti", trackbar kecepatan dan mengaktifkan tombol "Setting Port".



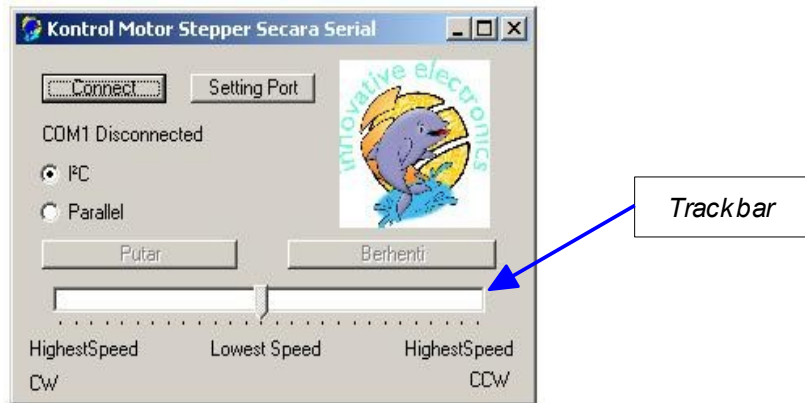
Gambar 3
Flowchart Tombol "Putar"

Cara kerja dari program berdasarkan *flowchart* Tombol "Putar" adalah sebagai berikut :

1. Pada *flowchart* tombol "Putar", program akan mengecek *radio button* I²C dan *radio button* paralel. *Radio button* yang dipilih akan menentukan jenis komunikasi antara SPC Stepper Motor dengan PC-Link Serial PPI.
3. Pada mode I²C, program memulai pengaturan motor stepper dengan perintah *init_step(\$ff,\$ff)*. Nilai *\$ff* pertama adalah *StepH* dan yang kedua adalah *StepL* dimana variabel tersebut merupakan nilai dari jumlah step yang dikirim ke SPC Stepper Motor. Inisialisasi ini akan membuat motor stepper berputar terus-menerus.
4. Program akan membaca nilai dari trackbar untuk menentukan kecepatan putar dan arah putaran motor stepper. Jika nilai dari trackbar tersebut kurang dari 14, maka nilai dari variabel **cmd** sama dengan **\$f** (Pin Run=1, pin Sel=1, pin Dir=1, dan pin Mod=1). Nilai pin Dir =1 berarti perputaran motor searah jarum jam. Jika nilai dari *trackbar* tersebut lebih besar dari 14 maka nilai dari variabel **cmd** sama dengan **\$d** (Pin Run=1, pin Sel=1, pin Dir=0, dan pin Mod=1). Nilai pin Dir =1 berarti perputaran motor berlawanan arah dengan jarum jam. Sedangkan untuk mengatur kecepatan putar motor stepper didasarkan pada nilai dari trackbar, dan nilai tersebut disimpan dalam variabel **speed0**.
5. Mengirimkan nilai dari variabel **speed0** dan nilai dari variabel **cmd** dikirim ke SPC Stepper Motor dengan rutin *init_speed* dan *init_command* secara I²C dengan menggunakan rutin *I2C_Start*, *I2C_Stop* dan *I2C_Wbyte*.
6. Pada mode paralel, program juga akan membaca nilai di *trackbar* terlebih dahulu untuk menentukan arah perputaran motor stepper dan kecepatan putar motor stepper.

7. Jika nilai *trackbar* kurang dari 14, pin S2 di SPC Stepper Motor akan diberi *high*. Jika nilai *trakcbar* lebih dari 14 maka akan diberi *low*.
8. Nilai kecepatan putar motor didapat dari nilai *trackbar* yang disimpan di variabel **speed0**.
9. Pin Reset (RST), Mode (S3) dan Sel (S1) diberi logika *high* karena menggunakan motor stepper bipolar dengan mode kerja *full mode*.
10. Motor stepper dijalankan dengan memberi pulsa pada pin *clock*. Pulsa tersebut adalah nilai logika *high* dan logika *low* secara bergantian dengan waktu tunda di variabel **speed0**. Semakin besar waktu tunda, semakin lambat pula motor stepper akan berputar.
11. Langkah di no.10 berjalan terus menerus selama variabel **stop** masih bernilai 0. Variabel **stop** tidak nol terjadi jika tombol "Berhenti" ditekan.

Tampilan program sebagai berikut:



Gambar 4
Tampilan Program

Keterangan :

- ◆ Tombol "Setting Port" : berguna untuk mengatur koneksi serial antara Komputer dengan PC-Link Serial PPI
- ◆ Tombol "Connect" : berguna untuk menutup/membuka komunikasi serial antara PC-Link Serial PPI dengan Komputer
- ◆ Tombol "Putar" : berguna untuk menjalankan motor stepper
- ◆ Tombol "Berhenti" : berguna untuk menghentikan motor stepper
- ◆ Radio button I²C dan parallel : berguna untuk menentukan jenis komunikasi yang akan digunakan antara PC-Link Serial PPI dengan SPC DC Motor
- ◆ Trackbar kecepatan : berguna untuk mengatur kecepatan dan arah putaran dari motor stepper.

Listing program terdapat di **AN150.ZIP**.

Selamat berinovasi!

Borland Delphi is copyright by Inprise Corporation.