

PENGATURAN PUTARAN DUA *MOTOR STEPPER* MELALUI LPT1

Riyanto¹, Yuneli Wikundhari²

^{1,2}Program Studi Diploma III Teknik Telekomunikasi, Purwokerto
¹riyanto@akatelsp.ac.id, ²yunelly_maniezzz@yahoo.co.id

Abstract

Existence of complication various industrial electronic system can cause to go down the result of industry. Hereinafter, intention of this final project is visual language application of basic for the application at a band conveyor system industry or factory so that water down in control of produced goods. Here visual language of basic used as control instruct rotation from stepper motor [pass/through] a PC. Here used PC only one, this matter because of to be more performance efficiency from the system easier so that in the case of observation and also control.

Keywords: Motor Stepper, Visual Basic 6.0, Parallel Port

1. Pendahuluan

Salah satu keinginan setiap orang adalah merasakan kemudahan dalam kegiatan sehari – hari, sehingga orang berfikir untuk membuat suatu alat yang dapat meringankan kegiatannya tanpa menguras tenaga, dan dapat memantau kinerja alat tersebut dengan perangkat lunak pada PC.

PC atau *Personal Computer*. Secara definisi komputer diterjemahkan sebagai sekumpulan alat elektronik yang saling bekerjasama, dapat menerima data (*input*), mengolah data (proses) dan memberikan informasi (*output*) serta terkoordinasi

dibawah kontrol program yang tersimpan di memorinya. PC merupakan jenis komputer yang dipakai untuk perorangan, bentuk dari PC relatif lebih kecil dibandingkan dengan komputer generasi sebelumnya yaitu *main frame*. Besar *main frame* adalah satu ruangan penuh.

Pada PC terdapat dua buah *port* yaitu *port serial* dan *port parallel*. *Port serial* ialah pengiriman data secara *serial* (data dikirim satu per satu secara berurutan), sehingga komunikasi *serial* jauh lebih lambat daripada komunikasi *paralel*. *Port serial* lebih sulit ditangani karena peralatan yang dihubungkan ke *port serial* harus berkomunikasi dengan menggunakan

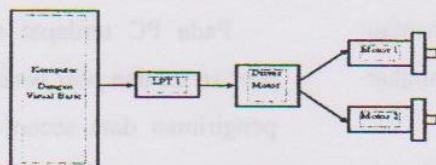
transmisi *serial*, sedang data di komputer diolah secara *parallel*. Oleh karena itu data dari dan ke *serial port* harus dikonversikan ke dan dari bentuk *parallel* untuk bisa digunakan. *Port parallel* ialah *port* data di komputer untuk mentransmisi 8 bit data dalam sekali detak.

Pada pengaturan putaran *motor stepper* melalui LPT1. Perangkat lunak yang digunakan adalah bahasa pemrograman *microsoft visual basic 6.0*.

2. Metode Penelitian

A. Hardware

Sistem pengendalian *motor stepper* terdiri dari beberapa perangkat keras yang akan mendukung kinerja program yang sudah disusun. Adapun secara garis besar gambaran keseluruhan sistem kendali *motor stepper* menggunakan *Visual Basic 6.0* adalah sebagai berikut:



Gambar 1. Diagram blok pengendali *motor stepper*

1. Komputer yang Dilengkapi

Program *Visual Basic 6.0*

Komputer merupakan pusat pengendalian yang akan mengendalikan rangkaian

berdasarkan perintah software *motor stepper* yang terdapat pada program *Visual Basic 6.0*. Komputer tidak boleh padam bila pengendalian harus dilakukan dalam waktu yang lama. Bila komputer padam maka secara otomatis program pengendalian yang sedang berjalan akan terhenti.

2. LPT (*Line Printer*)

Untuk menghubungkan antara komputer dengan perangkat kendali digunakanlah *port parallel*. Komunikasi yang terjadi hanya satu arah saja dimana komputer akan mengirimkan data ke perangkat kendali dan tidak bisa membaca data dari perangkat kendali.



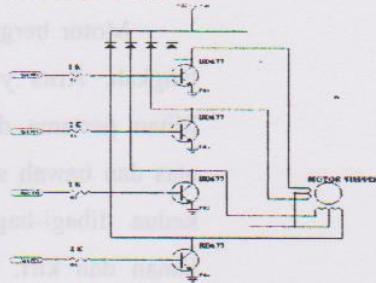
Gambar 2. Pin Konektor DB-25

Alamat *port* yang digunakan adalah 888 atau alamat LPT 1, dan jalur yang digunakan adalah jalur data dimana ada delapan jalur data yang digunakan yaitu D0-D7.

Suatu *port* berisi satu set sinyal dari CPU yang mengirimkan atau menerima data dengan komponen lain. Didalam pemberian sinyal, sinyal *start* adalah

"1" dan sinyal *stop* adalah "0" seperti sistem biner.

3. Driver Motor



Gambar 3. Rangkaian pengatur putaran motor stepper

Elemen penyusun rangkaian kendali:

a. 8 Buah Resistor 2K

Gambar 4. Resistor 2K

Maka nilai resistansinya: Gelang merah nilainya =2 dan gelang hitam nilainya = 0. Jadi satuannya adalah 20. Gelang ketiga adalah faktor pengali, dan jika warna gelangya merah berarti faktor pengalinya adalah 100. Sehingga nilai resistansi resistor tersebut adalah nilai satuan x faktor pengali atau $20 \times 100 = 2K$ Ohm dan toleransinya adalah 5%.

Rangkaian Kendali menggunakan 8 buah Resistor 2K Ohm yang terhubung dengan pin 2 sampai pin 9 *port parallel*. Resistor berfungsi sebagai memperkecil arus

yang akan masuk ke basis transistor BD139 karena arus yang diperlukan oleh basis agar Transistor sangat kecil.

b. 8 Buah Transistor BD139 (NPN)

8 buah Transistor BD139 yang berfungsi sebagai saklar. Kerja motor stepper akan tergantung dari ada tidaknya arus yang masuk ke basis Transistor BD139. Bila tidak ada arus yang masuk ke basis, maka motor stepper tidak akan bekerja. Sedangkan bila ada arus yang masuk ke basis maka motor akan bekerja.

Transistor BD 139 merupakan transistor *power*, khusus untuk transistor *power* sangat perlu untuk mengetahui spesifikasi P_{Dmax} . Spesifikasi ini menunjukkan temperatur kerja maksimum yang diperbolehkan agar transistor masih bekerja normal. Sebab jika transistor bekerja melebihi kapasitas daya P_{Dmax} , maka transistor dapat rusak atau terbakar.

c. 8 Buah Dioda 1000 A

Dioda ini bertugas untuk mengendalikan tegangan balik dari motor.

d. VCC (Power Supply)

Motor stepper membutuhkan tegangan yang stabil agar dapat tetap bekerja. Saat ini keperluan untuk membuat regulator, telah tersedia IC khusus seperti 7812 untuk regulator tegangan positif 12 VDC.

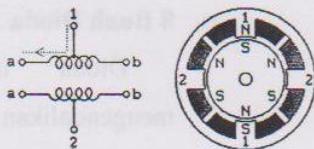
e. *Printed Circuit Board (PCB)*

Jenis PCB yang digunakan adalah PCB gambar, PCB ini dipilih dengan alasan kerapian rangkaian.

4. *Motor Stepper*

a. *Jenis Unipolar Motor*

Stepper motor terdiri dari berbagai ukuran dan jenis, dimulai dari disket motor kecil samapai permesinan yang sangat besar. *Motor stepper* yang digunakan untuk Tugas Akhir ini berjumlah dua buah dan kedua-duanya berjenis *unipolar motor*. Hal ini dikarenakan langkah *unipolar motor* yang mudah untuk diamati, dan kedua motor memiliki satu pusat lilitan sebagai penghubung ke *power supply*.



Gambar 5. *Motor Stepper* jenis *unipolar*

Kutup *unipolar stepper motor*, kedua-duanya Maknit

tetap. Dan menggunakan 5 atau 6 kawat. Gambar 3.9, dengan satu pusat pada tiap dua lilitan.

Motor bergerak 30 derajat tiap langkah. Arus yang mengalir pada lilitan pertama dibagi-bagikan pada atas dan bawah stator, sedang lilitan kedua dibagi-bagikan antara motor kanan dan kiri. Rotor adalah suatu maknit tetap dengan 6 kutub, 3 selatan dan 3 utara, yang diatur di sekitar *circumference* nya.

Karena resolusi sudut motor lebih tinggi, rotor harus mempunyai perbandingan lebih sedikit yaitu 30 derajat.

b. *Sudut Putar Motor Stepper*

Motor stepper merupakan satu-satunya jenis motor yang bekerja secara digital (sesuai masukan pulsa logika). Tiap masukan pulsa logika dapat menghasilkan satu langkah pada *motor stepper*. Sehingga dalam 1 revolusi tiap langkah *motor stepper* dapat dihitung sudutnya:

$$\text{Sudut} = 360^\circ / n \text{ rotor}$$

Besar sudut putar *motor stepper* tergantung dari jumlah rotor yang menyusunnya. Karena jenis motor stepper berbeda-beda, maka jumlah rotornya juga berbeda-beda.

Motor stepper hanya berhenti setelah jumlah pulsa sudah tercapai, sehingga *motor stepper* tidak perlu rangkaian pengereman.

c. Kecepatan Motor Stepper

Kecepatan dalam *motor stepper* mempunyai range yang sangat lebar, sesuai dengan jumlah rotor yang menyusunnya. *Motor stepper* dirancang mampu berjalan mencapai panas diantara (50°-90°C). Bagaimanapun, terlalu banyak putaran dan semakin cepat putaran *motor stepper* menyebabkan pemanasan berlebihan dan kerusakan pada isolasi motor dan lilitan.

Kecepatan dari putaran batang *motor stepper* secara langsung dihubungkan dengan frekwensi dari pulsa masukan dalam satu putaran dan panjang perputaran sesuai dengan banyaknya pulsa masukan. Urutan pulsa diterapkan secara langsung dengan dihubungkan ke arah putaran batang motor.

B. Software Motor Stepper

Software merupakan bagian komputer yang menangani pengolahan data gerakan *motor stepper*. Untuk dapat melakukan pengendalian, maka diperlukan

software yang sesuai dengan keperluan tersebut. Hubungan antara *input*, *output* dan *software* dapat dilihat pada gambar di bawah ini:

1. Form Utama

Untuk membangun *software* pengendali, digunakan bahasa pemrograman *Visual Basic 6.0*. Adapun komponen-komponen dari bahasa pemrograman yang dipakai adalah sebagai berikut:

a. Form1

Form merupakan tempat untuk meletakkan komponen program yang digunakan sesuai dengan kebutuhan dari pembuat program.

b. Tombol Perintah (*Command Button*)

Komponen ini berfungsi untuk mengeksekusi perintah apabila tombol perintah diklik. Ada 3 tombol perintah yang digunakan dalam program ini yakni **KANAN**, **KIRI** dan **BERHENTI**. Adapun fungsi dari masing-masing tombol perintah dapat dilihat sebagai berikut:

1) KANAN

Perintah ini digunakan untuk mengendalikan kedua motor ke arah kanan dengan waktu tunda

dan jumlah langkah yang telah ditentukan.

2) **KIRI**

Perintah ini digunakan untuk mengendalikan kedua motor ke arah kiri dengan waktu tunda dan jumlah langkah yang telah ditentukan.

3) **BERHENTI**

Tombol ini digunakan apabila pengguna program ingin menghentikan gerak kedua motor secara tiba-tiba, dengan catatan *option* manual telah dipilih.

c. Label

Dalam program ini terdapat 3 label yang digunakan untuk menampilkan informasi yang berasal dari program itu sendiri. Adapun fungsi dari masing-masing label tersebut:

- 1) **Label1** berfungsi untuk menginformasikan label **Waktu Tunda**
- 2) **Label2** berfungsi untuk menginformasikan label **Jumlah Langkah**
- 3) **Label3** berfungsi untuk menginformasikan label **Arah Gerak**

d. Kotak Teks (Text Box)

Komponen ini berfungsi untuk menampilkan isi teks pada saat

program tersebut sedang digunakan. Perbedaan antara kotak teks dengan label adalah label menampilkan teks yang terdapat dalam properti *caption*, sedangkan kotak teks menampilkan teks yang terdapat dalam properti teks. Ada 2 kotak teks yang digunakan dalam program ini yaitu:

- 1) **Txt1** berfungsi untuk menampilkan waktu tunda putaran motor.
- 2) **Txt2** berfungsi untuk menampilkan jumlah langkah motor.

e. Option Button

Dalam program ini *optionbutton* yang digunakan untuk menjalankan pengendalian rangkaian *motor stepper*. Ada 2 *optionbutton* yang digunakan:

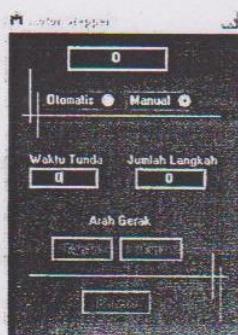
- 1) **Otomatis** yang berfungsi untuk menggerakkan motor secara otomatis sesuai dengan jumlah langkah motor dan kecepatan motor yang telah ditentukan. Motor tidak akan berhenti sampai nilai jumlah langkah motor yang ditentukan.
- 2) **Manual** yang berfungsi untuk menggerakkan motor secara manual, sesuai dengan jumlah langkah motor dan kecepatan motor yang telah ditentukan. Motor akan berhenti jika tombol berhenti ditekan.

f. Inpout32.dll

Komponen ini berfungsi sebagai antarmuka antara rangkaian kendali

dengan komputer yang terhubung menggunakan *port parallel*. Komponen ini dapat di-*download* dari internet. Setelah mendapatkan *file* ini maka harus disalin ke *folder C:/windows/system* agar dapat digunakan.

Setelah komponen-komponen tersebut dikumpulkan pada *form* program maka tampilan program adalah sebagai berikut:



Gambar 6. *Form* Utama

3. Hasil Dan Pembahasan

A. Perangkat Keras

Setelah selesai merancang *hardware* dan *software*, maka selanjutnya kedua komponen ini akan diuji apakah dapat bekerja seperti yang diinginkan atau tidak. Setelah pengujian menunjukkan hasil yang baik, maka didapat suatu alat yang siap dijalankan untuk mengendalikan *motor stepper* dari jarak dekat yang menggunakan simulasi kendali

dengan menggunakan *software motor stepper* yang telah dibuat.

Hasil dari perancangan alat pengendali *motor stepper* melalui LPT1 dapat dilihat pada gambar di bawah ini:



Gambar 7. Bagian Sisi Pengendalian Perangkat

Pada gambar di atas menunjukkan bahwa peralatan kendali terdiri dari 2 bagian yaitu bagian pertama adalah pusat kendali yakni komputer berfungsi untuk mengirim data perintah pengaturan pengendalian menggunakan simulasi dengan mengetikkan jumlah langkah dan waktu *delay* putaran *motor stepper* pada *software motor stepper* yang telah dibuat, yang kedua adalah rangkaian pengendali yakni bagian yang menerima *output* pengendali dari pusat pengendalian berupa kombinasi bit biner hasil terjemahan data dari komputer yang mengendalikan arah putar, waktu *delay*, dan langkah *motor stepper*.

B. Bagian Pusat Pengendali (Komputer)

Pada gambar di atas terlihat bahwa komputer digunakan sebagai pusat

- 2) Tampilan program seperti ditunjukkan oleh gambar 6 akan muncul setelah *icon* di klik.
- 3) Memilih mode yang digunakan “MANUAL” atau “OTOMATIS” sesuai kebutuhan.

Jika mode otomatis dipilih maka jumlah langkah *motor stepper* yang terakhir atau maksimal akan sesuai dengan jumlah yang ada pada kotak “Jumlah Langkah” dan dalam hal ini tombol berhenti tidak berfungsi.

Dalam mode manual, kotak “Jumlah Langkah” tidak terlihat. Dikarenakan pada mode manual “Jumlah Langkah” tidak diperlukan, motor akan melangkah dengan jumlah yang tidak tentu, motor hanya akan berhenti setelah tombol “Berhenti” ditekan.

- 4) Untuk keluar dari program, adalah dengan cara menekan tombol close pada ujung kanan tampilan halaman program, maka secara otomatis program akan ditutup.

E. Hasil Percobaan

1. Hasil percobaan pengendalian

Dari hasil percobaan yang telah dilakukan, pada *motor*

stepper baik menggunakan *mode* otomatis maupun *mode* manual memiliki hasil yang sama. Pada percobaan ini digunakan “Waktu tunda” 0.8 dan “Jumlah langkah” manual 10, sedangkan untuk *mode* otomatis saat motor mencapai langkah ke 8 tombol berhenti ditekan sehingga motor secara otomatis berhenti pada langkah ke 9.

a. Ke arah kanan

Tabel 4.1 Tabel hasil pengamatan gerak motor ke arah kanan

KONDI SI	BINER	DESIM AL
0	0000.00	0
	00	
1	0001.00	17
	01	
2	0010.00	34
	10	
3	0100.01	68
	00	
4	1000.10	136
	00	
5	0001.00	17
	01	
6	0010.00	34
	10	
7	0100.01	68
	00	
8	1000.10	136

00

KONDISI	BINER	DESIMAL
0	0000.0000	0
1	1000.1000	136
2	0100.0100	68
3	0010.0010	34
4	0001.0001	17
5	1000.1000	136
6	0100.0100	68
7	0010.0010	34
8	0001.0001	17
9	1000.1000	136
10	0100.0100	68
9	0001.00	17
	01	
10	0010.00	34
	10	

Karena pengendalian ini menggunakan kedua motor maka deklarasi logika untuk kedua motor binernya digabungkan.

Motor 1 untuk kondisi pertama = 0000.0001 atau 1

Motor 2 untuk kondisi pertama = 0001.0000 atau 16

Maka agar dapat menggerakkan keduanya secara bersamaan maka jumlahkan keduanya sehingga didapat =

0001.0001 atau 17. Sedangkan untuk kondisi selanjutnya tambahkan nilai 17 pada kondisi sebelumnya, untuk seterusnya akan didapat perputaran kedua motor secara bersama-sama ke arah kanan.

b. Ke arah kiri

Tabel 4.2 Tabel hasil pengamatan gerak motor ke arah kiri

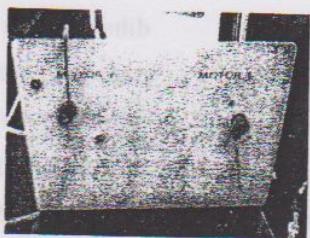
Karena pengendalian ini menggunakan kedua motor maka deklarasi logika untuk kedua motor binernya digabungkan.

Motor 1 untuk kondisi pertama = 0000.1000 atau 8

Motor 2 untuk kondisi pertama = 1000.0000 atau 128

Maka agar dapat menggerakkan keduanya secara bersamaan maka jumlahkan keduanya sehingga didapat = 1000.1000 atau 136.

Sedangkan untuk kondisi selanjutnya kurangi nilai 17 pada kondisi sebelumnya, untuk seterusnya akan didapat perputaran kedua motor secara bersama-sama ke arah kiri.



Gambar 9. Saat perangkat dalam kondisi 10, putaran ketiga

Dari hasil pengamatan di atas, maka dapat disimpulkan bahwa untuk mencapai satu putaran penuh kedua motor memiliki jumlah langkah yang berbeda. Untuk motor 1 hanya memerlukan 20 langkah dalam satu putaran dengan tiap langkah ditempuh 18° , sedangkan untuk motor 2 memerlukan 50 langkah untuk satu kali putaran, dengan tiap langkah ditempuh 7.2° .

Pengendalian *motor stepper* ini dirancang khusus untuk berjalan pada kecepatan rendah atau membatasi kecepatan maksimum. Hal ini dikarenakan kecepatan motor yang rendah mampu meminimalisasi selip.

Selain itu karakteristik motor yang mampu memegang tenaga putaran (*holding torque*) akan selalu memegang posisi motor.

Maka saat motor mencapai posisi terakhir motor tidak dapat digerakan hal ini dikarenakan pulsa terakhir yang masuk pada motor tetap dijaga.

Untuk memperkecil permasalahan dalam motor, kecepatan diatur konstan agar panas yang dihasilkan setelah melakukan putaran tidak terlalu tinggi. Semakin cepat perputaran motor, semakin tinggi suhu motor. *Motor stepper* dirancang berjalan pada suhu ($50^\circ-90^\circ$ C). Bagaimanapun, terlalu banyak putaran menyebabkan pemanasan berlebihan dan kerusakan pada isolasi motor dan lilitan.

2. Perhitungan Sudut Putar Motor

Motor stepper dalam 1 kali putar mencapai sudut 360° .

Dari hasil pengamatan diperoleh 1 revolusi motor 1 hanya bergerak sebanyak 20 langkah.

$$\text{Sudut} = 360^\circ / n \text{ rotor} \\ 360^\circ / 20 = 18^\circ$$

Dari hasil pengamatan diperoleh 1 revolusi motor 2 hanya bergerak sebanyak 50 langkah.

$$\text{Sudut} = 360^\circ / n \text{ rotor} \\ 360^\circ / 50 = 7.2^\circ$$

Dari hasil perhitungan didapat bahwa motor 1 dan 2 memiliki jenis yang sama yaitu unipolar namun jumlah rotor yang tersusun di dalamnya berbeda, sehingga untuk mencapai 1 putaran penuh juga ditempuh berbeda sesuai dengan tiap sudut langkah yang mampu dicapainya.

G. Pengamatan Kerja Rangkaian Kendali Secara Elektronis

Rangkaian kendali terdiri dari komponen-komponen elektronik arus lemah dimana salah satunya adalah Transistor. Transistor mempunyai 3 kutub yakni Kolektor, Basis, dan Emitor. Aplikasi Transistor disini bukanlah sebagai penguat, melainkan sebagai saklar elektronik, sehingga bila basis transistor menerima arus (berlogika tinggi) maka seolah-olah antara kolektor dan basis akan terhubung singkat, sedangkan bila arus basis 0 (berlogika rendah), maka antara kolektor dan emitor berperan sebagai Resistor yang besarnya tak terhi

4. Kesimpulan Dan Saran

A. Kesimpulan

1. Antara komputer dengan perangkat pengendali *motor stepper*

dihubungkan menggunakan *port parallel*, yang memberikan tegangan *output* ke *driver* sebesar 5V.

2. Kedua motor memiliki jumlah rotor yang berbeda sehingga tiap langkah masing-masing motor juga berbeda. Untuk motor 1 memerlukan 20 langkah untuk menempuh 1 putaran penuh, sedangkan untuk motor 2 memerlukan 50 langkah untuk menempuh 1 putaran penuh.
3. *Motor stepper* bergerak berdasarkan urutan pulsa dengan logika bit biner dari *port printer* yang masuk pada rotor sesuai perintah *software motor stepper*.
4. Gerakan kedua motor secara bersamaan merupakan hasil dari penjumlahan kondisi kedua motor dengan logika biner. Untuk bergerak ke arah kanan jumlah biner awal yang digunakan adalah (0001.0001) atau 17 kemudian jumlahkan dengan nilai 17, begitu seterusnya untuk kondisi selanjutnya. Untuk bergerak ke arah kiri kurangkan biner awal yang digunakan adalah (1000.1000) atau 136. kemudian kurangi nilai 17, begitu seterusnya untuk kondisi selanjutnya.

B. Saran

1. Agar mengurangi ketergantungan menggunakan komputer sebagai pusat pengendali maka fungsi komputer

dapat digantikan dengan rangkaian mikrokontroler.

2. Perintah melalui komunikasi *parallel* tidak bisa dilakukan secara bersama-sama, untuk pengembangannya diharapkan bisa dilakukan secara bersama-sama sehingga bisa mempercepat proses kerja.
3. Koneksi antara perangkat kendali dengan komputer dapat diganti dengan radio agar dapat memangkas penggunaan kabel dan mampu mengontrol pada jarak yang lebih jauh.

DAFTAR PUSTAKA

Alexander, Hengky M., *Belajar Sendiri Membangun Sistem Database dengan Visual Basic 6.0 dan Access 2000*, Elex Media Komputindo, Jakarta, 2003.

Budiharto, Widodo dan Sigit Firmansyah, *Elektronika Digital dan Mikroprosesor*, Penerbit ANDI, Yogyakarta, 2005.

Endra Pitowarnao, *Mikroprosesor & Interface II*, PENS-ITS Surabaya, 1994.

Yuhefizar, *Tutorial Komputer dan Jaringan*, <http://www.ilmukomputer.com>, 2003.

Baskar Gupta, *Stepper motor control trough parallel port*, <http://www.kodeproject.com>, 17 Mei 2007.

Aswan Hamonangan, *Resistor*, <http://www.electroniclab.com>, 17 Mei 2005.

Aswan Hamonangan, *Transistor*, <http://www.electroniclab.com>, 17 Mei 2005.

Aswan Hamonangan, *Dioda*, <http://www.electroniclab.com>, 17 Mei 2005.

Jones Douglas, *Stepping Motor Types*, <http://www.cs.euoa.edu>, 2006.

Annonameous, *IN4001-IN4007*, <http://www.fairchildsemi.com/ds>, 1 September 2007.

Annonameous, *Linear Integrated Circuit*, <http://www.contek-ic.com>, 3 September 2007.

Thomson, *NPN Silicon Transistor*, <http://www.datasheetarchive.com>, 19 Mei 2007.

Annonameous, *Interfacing the Standard Parallel Port*, http://www.beyondlogic.org_spp_parallel.htm, 5 september 2007

Jan Axelsons, *Inpout32.dll for Windows*, http://www.logix4u.net_parallelport1.htm, 10Mei 2007.