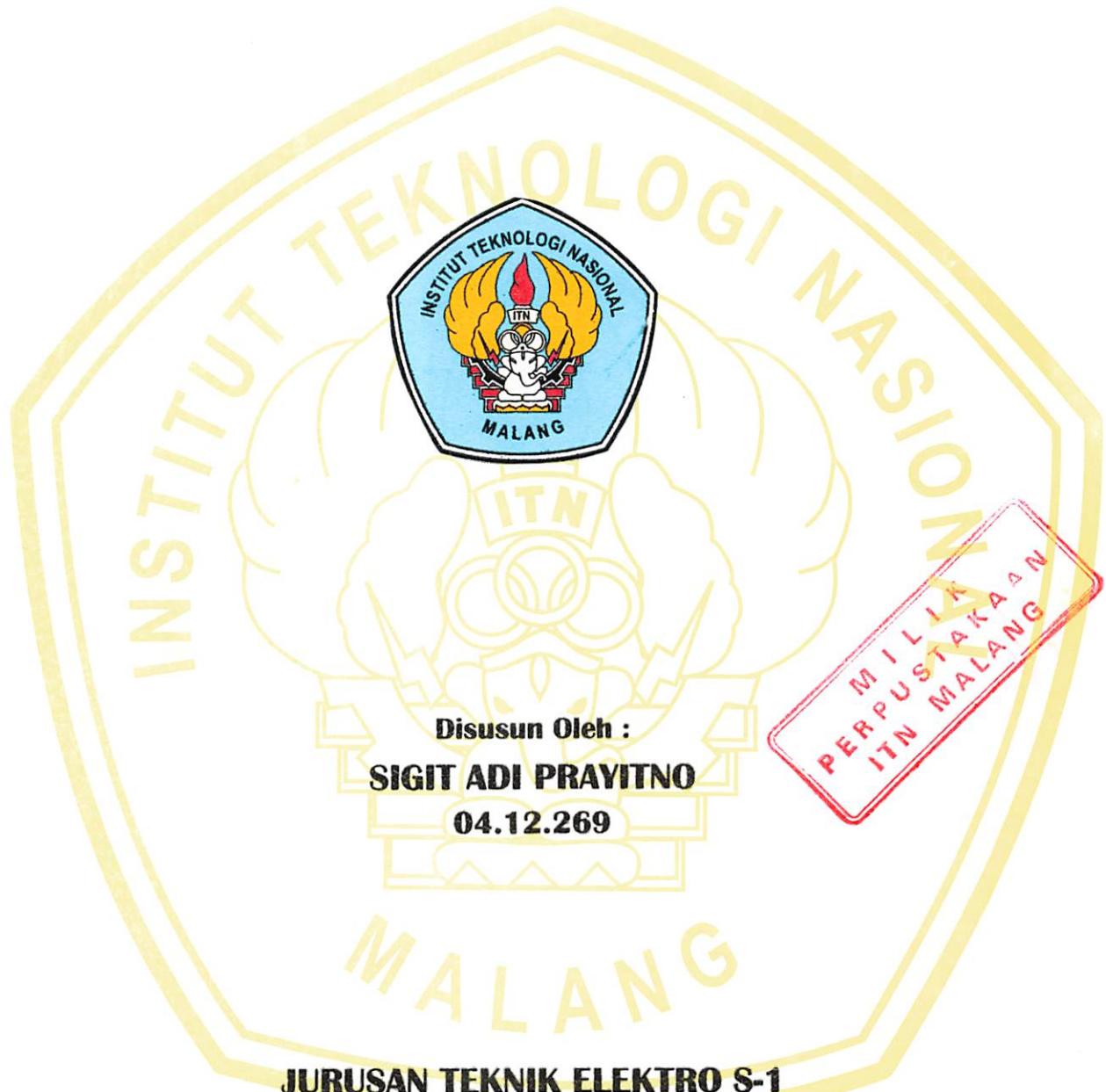


# **SKRIPSI**

**PERANCANGAN DAN PEMBUATAN ALAT PENGEBORAN  
PCB BERBASIS *COMPUTER NUMERIC CONTROL (CNC)***



**JURUSAN TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK ELEKTRONIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2009**

8000

GENERAL INFORMATION CONCERNING THE  
EXECUTIVE INFORMATION SYSTEM  
MANAGEMENT INFORMATION SYSTEMS  
TELEINFORMATION AND COMPUTER SYSTEMS

801000

GENERAL INFORMATION  
MANAGEMENT INFORMATION

TELEINFORMATION

THE COMPUTER SYSTEMS WHICH ARE USED  
FOR INFORMATION FROM INFORMATION SYSTEMS

802000

**LEMBAR PERSETUJUAN**  
**PERANCANGAN DAN PEMBUATAN ALAT PENGEBORAN**  
**PCB BERBASIS COMPUTER NUMERIC CONTROL (CNC)**

**SKRIPSI**

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh  
Gelar Sarjana Teknik Elektronika Strata Satu (S-1)*

Disusun Oleh :

SIGIT ADI PRAYITNO

NIM : 04.12.269

Diperiksa dan Disetujui

Dosen Pembimbing I

Irmalia S. Faradisa, ST, MT  
NIP. P. 1030000365

Dosen Pembimbing II

I Komang Sumawirata, ST,MT  
NIP.P.103010361

Mengetahui,

Ketua Jurusan Teknik Elektro S-1



Ir. F. Yudi Limpraptono, MT  
NIP. Y. 1039500274

**JURUSAN TEKNIK ELEKTRO S-1**  
**KONSENTRASI TEKNIK ELEKTRONIKA**  
**FAKULTAS TEKNOLOGI INDUSTRI**  
**INSTITUT TEKNOLOGI NASIONAL MALANG**  
**2009**



INSTITUT TEKNOLOGI NASIONAL  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK ELEKTRONIKA

**BERITA ACARA UJIAN SKRIPSI**  
**FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Sigit Adi Prayitno  
NIM : 04.12.269  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika  
Judul Skripsi : PERANCANGAN DAN PEMBUATAN ALAT  
PENGEBORAN PCB BERBASIS *COMPUTER NUMERIC  
CONTROL ( CNC )*

Dipertahankan di hadapan majelis penguji Skripsi jenjang Strata satu ( S-1 ) pada :

Hari : Selasa  
Tanggal : 06 Oktober 2009  
Dengan Nilai : 83,7 (A) *Suy*



( Ir. H. Sidik Noertjahjono, MT )  
NIP.Y.1028700163

Penguji I

( Joseph Dedy Irawan, ST, MT )  
NIP. 132315178



( Ir. F. Yudi Limpraptono, MT )  
NIP.Y.1039500274

Penguji II

( Sotyoahadi, ST, MSc )  
NIP.Y.1039700309

## ABSTRAK

**Sigit Adi Prayitno**

Jurusan Teknik Elektro

Institut Teknologi Nasional Malang

[ada\\_shigit@ymail.com](mailto:ada_shigit@ymail.com)

Sangat banyak aplikasi dari teknik pengeboran yang dipakai dalam dunia industri, seperti industri elektronika, industri konveksi, industri pengepakan dan lain-lainnya. Dalam dunia industri elektronika, biasanya bor digunakan untuk mengebor gambar pada *Printed Circuit Board* (PCB). Permasalahan yang sering dihadapi oleh pengusaha PCB adalah kecepatan dari produksi PCB dan ketepatan posisi gambar pada PCB. Skripsi ini merancang dan membuat sebuah alat bor pada PCB secara otomatis yang dilengkapi dengan mekanisme yang dirancang khusus untuk menjaga PCB agar tetap pada jalur yang ditentukan.

Sistem kontrol alat ini menggunakan Mikrokontroller ATMega 8535 yang bertugas mengontrol semua peralatan pendukungnya seperti *motor Stepper* dan lainnya. PCB dideteksi menggunakan sensor *limit switch* yang selanjutnya akan mengirimkan sinyal pada mikrokontroler untuk memulai proses kerja. Pada proses pengeboran, PCB akan kembali dideteksi oleh sensor *limit switch* yang berfungsi untuk menghentikan *motor stepper* dan mengintruksikan untuk melakukan proses pengeboran. Setelah proses ini selesai mikrokontroler akan memerintahkan *motor stepper* untuk berjalan kembali. Sehingga dihasilkanlah PCB yang sudah terlubangi, proses itu akan berulang-ulang terus secara berkesinambungan.

**Kata kunci :** Bor PCB Otomatis, Mikrokontroler

## KATA PENGANTAR

Dengan ucapan syukur kepada Tuhan Yang Maha Esa atas segala Kasih dan Anugerah-Nya, telah memberikan kekuatan, ketekunan, kesabaran, bimbingan dan perlindungan sehingga penulis dapat menyelesaikan laporan skripsi dengan judul :

### **“ PERANCANGAN DAN PEMBUATAN ALAT PENGEBORAN PCB BERBASIS COMPUTER NUMERIC CONTROL (CNC) “**

Pembuatan skripsi ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata 1 di Institut Teknologi Nasional Malang. Dalam penyusunan skripsi ini penulis banyak mendapat bantuan baik moril maupun materiil, saran dan dorongan semangat dari berbagai pihak, untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE selaku rektor ITN Malang
2. Bapak Ir. Sidik Noertjahono, MT selaku Dekan Fakultas Teknologi Industri.
3. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S-1 ITN Malang.
4. Ibu Irmalia S. Faradisa, ST. MT selaku Dosen Pembimbing I.
5. Bapak I Komang Sumawirata, ST. MT selaku Dosem Pembimbing II.
6. Orangtua serta saudara-saudara kami yang telah memberikan doa restu, motivasi, semangat dan biaya.
7. Semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

## **Kata Pengantar**

---

Dengan segala keterbatasan kemampuan dan pengetahuan maka penyusun menyadari bahwa Laporan Skripsi ini masih banyak terdapat kekurangan, oleh karena itu segala saran dan kritik yang konstruktif dari semua pihak sangat kami harapkan guna penyempurnaan Laporan Skripsi.

Akhirnya dengan segala kerendahan hati penyusun mengharap semoga laporan ini dapat bermanfaat bagi pihak yang memerlukan.

Malang, Oktober 2009

**Penulis**

## DAFTAR ISI

<b>LEMBAR PERSETUJUAN.....</b>	i
<b>BERITA ACARA.....</b>	ii
<b>ABSTRAK.....</b>	iii
<b>KATA PENGANTAR.....</b>	iv
<b>DAFTAR ISI.....</b>	vi
<b>DAFTAR GAMBAR.....</b>	ix
<b>DAFTAR ISI.....</b>	xii
<b>BAB I PENDAHULUAN.....</b>	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Metodologi .....	4
1.6 Sistematika Pembahasan.....	5
<b>BAB II TEORI PENUNJANG.....</b>	6
2.1 Mikrokontroler AVR ATmega 8535.....	6
2.1.1 Arsitektur AVR ATmega 8535.....	7
2.1.2 Pin – pin Atmega 8535 .....	10
2.1.3 Deskripsi Pin – pin ATMega 8535 .....	11
2.1.4 Port Sebagai Input/Output Digital .....	13
2.1.5 Organisasi Memori AVR ATmega8535 .....	14
2.1.5.1 Program Memori.....	15

2.1.5.2 Data Memori.....	15
2.2 Serial Port RS-232 .....	17
2.3 Optocoupler IC 4N25 .....	20
2.4 Driver Motor Stepper.....	21
2.4.1 The Metal Oxide Semiconductor FET (MOSFET) .....	21
2.4.2 Simbol Rangkaian MOSFET .....	23
2.4.3 Karakteristik MOSFET .....	23
2.5 Motor Stepper .....	25
2.5.1 Deskripsi .....	25
2.5.2 Jenis Motor Stepper .....	28
2.5.2.1 Stepper Motor Unipolar .....	28
2.5.2.2 Stepper Motor Bipolar .....	30
2.6 Motor DC.....	32
2.7 <i>Limit Switch</i> .....	34
2.8 Sistem Mekanik.....	35
2.9 Borland Delphi.....	36
2.9.1 <i>IDE (Integrated Development Enviroment)</i> .....	37
2.9.2 <i>Menu Borland Delphi</i> .....	39
<b>BAB III PERANCANGAN DAN PEMBUATAN ALAT .....</b>	<b>41</b>
3.1 Perancangan Sistem .....	41
3.2 Prinsip Kerja .....	42
3.3 Perancangan Hardware .....	43
3.3.1 RS-232 Serial Interface .....	43
3.3.2 <i>Limit Switch</i> .....	44

---

***Daftar Isi***

---

3.3.3 Mikrokontroler.....	45
3.3.4 Driver Motor Stepper.....	46
3.3.5 Driver Motor DC .....	48
3.3.6 Mekanik Mesin .....	50
3.4 Perancangan Perangkat Lunak ( Software ) .....	52
3.4.1 Program Aplikasi Mikrokontroler .....	52
3.4.2 Program Aplikasi <i>Borland Delphi7</i> .....	55
<b>BAB IV PENGUJIAN ALAT .....</b>	<b>57</b>
4.1 Pengujian Rangkaian Driver Relay.....	57
4.1.1 Alat dan Bahan .....	57
4.1.2 Langkah Pengujian.....	57
4.1.3 Hasil dan Analisa Pengujian <i>Driver Relay</i> .....	58
4.2 Pengujian Rangkaian Driver Motor Stepper.....	60
4.2.1 Alat dan Bahan .....	60
4.2.2 Langkah Pengujian.....	60
4.2.3 Hasil dan Analisa Pengujian <i>Driver Motor Stepper X</i> .....	61
4.2.4 Hasil dan Analisa Pengujian <i>Driver Motor Stepper Y</i> .....	62
4.2.5 Hasil dan Analisa Pengujian <i>Driver Motor Stepper Z</i> .....	64
4.3 Hasil Pengujian Keseluruhan.....	65
<b>BAB V PENUTUP.....</b>	<b>67</b>
5.1 Kesimpulan.....	67
5.2 Saran .....	67

**DAFTAR PUSTAKA****LAMPIRAN**

## **DAFTAR GAMBAR**

<b>BAB II TEORI PENUNJANG .....</b>	<b>6</b>
Gambar 2.1 Pin – pin Atmega 8535 kemasan 40-pin .....	10
Gambar 2.2 Arsitektur ATmega 8535 .....	11
Gambar 2.3 Pemetaan program memori .....	15
Gambar 2.4 Pemetaan Data Memori .....	16
Gambar 2.5. IC MAX 232.....	18
Gambar 2.6. <i>1 byte of Async Data</i> .....	18
Gambar 2.7. Konektor DB-9 .....	19
Gambar 2.8. Skematik <i>optocoupler 4N25</i> .....	20
Gambar 2.9. Jenis MOSFET .....	22
Gambar 2.10. Simbol <i>MOSFET channel N</i> .....	23
Gambar 2.11 . MOSFET jenis pengisian.....	24
Gambar 2.12. <i>Karakteristik keluaran MOSFET tipe enhancement</i> ....	25
Gambar 2.13. Motor Stepper.....	26
Gambar 2.14. <i>Simbol Stepper Motor Unipolar</i> .....	28
Gambar 2.15. Pembalikan Arus Pada <i>Stepper Motor Unipolar</i> .....	29
Gambar 2.16. Sekuen <i>Putaran Stepper Motor Unipolar</i> .....	29
Gambar 2.17 <i>Simbol Motor Stepper Bipolar</i> .....	30
Gambar 2.18 Sekuen Putaran <i>Motor Stepper Bipolar</i> .....	31
Gambar 2.19 Konsep <i>H-Bridge</i> pada <i>Motor Stepper Bipolar</i> .....	31
Gambar 2.20 Garis – Garis Medan Magnet .....	33
Gambar 2.21 Gaya Yang Dihasilkan Motor DC .....	33

Gambar 2.22 Torsi Yang Ditimbulkan Motor DC .....	34
Gambar 2.23. Fisik <i>Limit Switch</i> .....	35
Gambar 2.24. Kontrusi Dasar Mekanik .....	36
Gambar 2.25 Kontruksi Ulir .....	36
Gambar 2.26. IDE ( <i>Integrated Development Enviroment</i> ) .....	37
<b>BAB III PERANCANGAN DAN PEMBUATAN ALAT .....</b>	<b>41</b>
Gambar 3.1 Diagram Sistem Secara Keseluruhan. ....	41
Gambar 3.2 Rangkaian RS-232 <i>Serial Interface</i> . ....	44
Gambar 3.3 Rangkaian <i>Limit Switch</i> . ....	45
Gambar 3.4 Rangkaian Minimum System ATMega8535 dan Alokasi Pin. ....	46
Gambar 3.5 Rangkaian <i>Driver Motor Stepper</i> .....	47
Gambar 3.6. Rangkaian <i>Driver Motor</i> .....	48
Gambar 3.7 Struktur Mekanik.....	50
Gambar 3.8 Desain Mekanik.....	51
Gambar 3.9 <i>Flow Chart Software</i> Pada Mikrokontroler .....	54
Gambar 3.10 <i>Flow Chart Software</i> Pada <i>Borland Delphi7</i> .....	56
<b>BAB IV PENGUJIAN ALAT .....</b>	<b>57</b>
Gambar 4.1 Diagram Blok Pengukuran <i>Driver Relay</i> .....	57
Gambar 4.2 Pengukuran Rangkaian <i>Driver Relay</i> .....	58
Gambar 4.3 Hasil Pengukuran Arus Ib Trasistor BC557 .....	58
Gambar 4.4 Hasil Pengukuran Arus Ie Trasistor BC557.....	59

---

***Daftar Gambar***

---

Gambar 4.5 Blok Diagram <i>Driver Motor Stepper</i> .....	60
Gambar 4.6 Pengukuran Arus IG Transistor IRFZ44N.....	61
Gambar 4.7 Pengukuran Rangkaian <i>Driver Motor Stepper X</i> . .....	61
Gambar 4.8 Pengukuran Rangkaian <i>Driver Motor Stepper Y</i> .....	62
Gambar 4.9 Pengukuran Rangkaian <i>Driver Motor Stepper Z</i> .....	64
Gambar 4.10 Hasil Pengujian Pada <i>Program Delphi7</i> .....	66
Gambar 4.11 Hasil Pengujian Keseluruhan. .....	66

## DAFTAR TABEL

<b>BAB II TEORI PENUNJANG .....</b>	<b>6</b>
Tabel 2.1 Fungsi Pin RS-232 dalam DB-9 .....	19
Tabel 2.2 Tahapan Putaran <i>Motor Stepper</i> .....	32
<b>BAB III PERANCANGAN DAN PEMBUATAN ALAT .....</b>	<b>41</b>
Tabel 3.1. Rumus penghitungan nilai UBRR untuk berbagai <i>mode</i> operasi.....	52
<b>BAB IV PENGUJIAN ALAT .....</b>	<b>57</b>
Tabel 4.1 Perbandingan Hasil Perhitungan Dan Hasil Pengukuran Arus Ib Pada Transistor BC55759	
Tabel 4.2 Perbandingan Hasil Perhitungan Dan Hasil Pengukuran Arus Ie Pada Transistor BC557 .....	59
Tabel 4.3 Perbandingan Hasil Perhitungan Dan Hasil Pengukuran Arus Id Pada Transistor IRFZ44N untuk <i>Driver Motor StepperX</i> .62	
Tabel 4.4 Perbandingan Hasil Perhitungan Dan Hasil Pengukuran Arus Id Pada Transistor IRFZ44N untuk <i>Driver Motor StepperY</i> 63	
Tabel 4.5 Perbandingan Hasil Perhitungan Dan Hasil Pengukuran Arus Id Pada Transistor IRFZ44N untuk <i>Driver Motor StepperZ</i> 65	
Tabel 4.6 Hasil Pengujian Keseluruhan.....	66

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Perkembangan teknologi yang semakin kompleks dalam semua bidang telah memunculkan suatu gagasan untuk menciptakan alat baru yang bisa membantu penghematan waktu dan proses kerja dari manusia. Salah satunya dibidang Elektronika Industri yaitu mesin pengeboran otomatis pada PCB. Skripsi ini merupakan pengembangan dari skripsi yang sudah ada yaitu “ *Perancangan Dan Pembuatan Alat Bor PCB Berbasis Mikrokontroler AT89S51* ” oleh Fransiskus Tiala A.K, Tugas Akhir, Program D-III Teknik ITN, Malang 2006, sedangkan alat ini adalah mesin pengeboran PCB otomatis berbasis *Computer Numeric Control (CNC)*.

Alat bor ini berbeda dengan alat bor pada umumnya, disamping itu alat ini sudah mengadopsi sistem otomatisasi, juga dapat melubangi PCB dengan tepat dan mudah, sehingga alat ini cukup efektif dan efisien serta mudah dalam pengoperasiannya. Penerapan teknologi yang tepat guna dalam bidang pengeboran sekarang masih jarang dilakukan. Diharapkan dengan menggunakan mesin pengeboran otomatis ini kita dapat banyak menghemat waktu dan biaya karena semua pengontrolan pada alat ini sudah diatur pada komputer dan mikrokontroler.

Proses pengeboran pada mesin ini dilakukan secara otomatis dengan memasukkan data koordinat PCB atau *drill file* ke *software* utama pada PC untuk diproses. Hasil proses PC yang berupa parameter-parameter tertentu akan dikirim secara *serial* ke mikrokontroler. Selanjutnya mikrokontroler akan memberikan aktivasi bit ke tiap driver motor stepper untuk menggerakkan motor sesuai dengan

parameter yang kehendaki. Disamping itu mikrokontroler akan mengaktifkan motor bor selama proses pengeboran berlangsung. Prinsip kerja alat ini hampir sama dengan alat bor otomatis pada umumnya hanya saja pada alat ini terdapat suatu mekanisme untuk menjaga PCB agar tetap pada jalur yang ditentukan, sehingga tingkat kegagalannya dapat dikurangi.

## **1.2 Rumusan Masalah**

Permasalahan utama yang akan dibahas dalam penulisan skripsi ini yaitu :

### **1. Pembuatan Perangkat Mekanik.**

Bagaimana membuat sebuah mekanik yang tepat sehingga mesin dapat berjalan stabil dan sesuai dengan koordinat yang diharapkan.

### **2. Pembuatan Perangkat Keras**

Bagaimana mikrokontroler AVR Atmega8535 mengatur atau sebagai pengolah data input sehingga menjadikan data output untuk sistem pergerakan mesin.

### **3. Komunikasi Data *Serial***

Bagaimana membuat komunikasi data *serial* sehingga dapat menyalurkan informasi yang telah diolah oleh *software* dari komputer ke mikrokontroller.

---

## **1.3 Batasan Masalah**

Pada skripsi ini merupakan *prototype* dari alat pengeboran pada PCB masal yang biasanya digunakan di industri – industri elektronika. Ukuran alat disini tidak sama dengan ukuran alat yang ada pada industri pengeboran PCB.

Dan PCB hasil pengeboran tidaklah sebanyak PCB yang dihasilkan pada industri – industri pengeboran PCB. Adapun batasan masalah pada skripsi ini adalah sebagai berikut :

1. Hanya menggunakan 1 ukuran PCB, menggunakan PCB berukuran kurang lebih panjang 25 cm dan lebar 15 cm.
2. Jarak per satu putaran *motor stepper* kurang lebih 50 mil (*Imperial*) atau 1.27 mm.
3. Posisi yang diijinkan horizontal atau vertical.
4. Tidak membahas catu daya yang digunakan.
5. Tidak membahas mengenai torsi *motor Stepper* serta motor bor.

#### **1.4 Tujuan**

Tujuan dari penulisan skripsi ini diantara lain sebagai berikut :

1. Mengaplikasikan komputer dan mikrokontroler sebagai pengontrol sistem pengeboran PCB.
2. Membuat alat semi otomatis yang dapat mengurangi kesalahan dalam proses pengeboran PCB.
3. Memperingan perkerjaan operator.

**1.5 Metodologi**

Pengerjaan skripsi ini dilakukan dengan langkah – langkah sebagai berikut:

**1. Tahap Studi Literatur**

Pada tahap ini yang dilakukan adalah merancang desain mekanik dan mempelajari bahasa pemrogramannya.

**2. Tahap Pembuatan mekanik dan elektronik (*hardware*)**

Pada tahap ini yang dilakukan adalah membuat perancangan perangkat keras dengan membuat desain mekanik dan mengatur letak motor yang mampu bergerak dengan baik, membuat hardware dan membuat program untuk mengontrol pergerakan motor dan membuat blok diagramnya.

**3. Tahap Pembuatan Program (*software*)**

Pembuatan perangkat lunak atau *software* disini dilakukan berdasarkan pada perancangan sistem *software* yang digunakan. Pembuatan perangkat lunak ini harus disesuaikan dengan metode yang digunakan dalam perancangan *software* tersebut. Pembuatan perangkat lunak ini dilakukan secara bertahap berdasarkan pada perancangan sistem *software* melalui *flow chart* sistem.

**4. Tahap Analisis Dan Interpretasi Data**

Tahap analisa dilakukan untuk mengetahui kerja dari sistem dan pengambilan data dilakukan per sumbu terutama sumbu X dan sumbu Y yang meliputi nilai input koordinat dan implementasi nilai koordinat yang dicapai oleh alat.

## **1.6 Sistematika Pembahasan**

Sistematika pembahasan dalam penyusunan penulisan skripsi ini adalah sebagai berikut :

### **BAB I Pendahuluan**

Bab ini berisi tentang penguraian secara singkat latar belakang, tujuan, perumusan masalah, batasan masalah, metodologi penulisan, dan sistematika penulisan.

### **BAB II Teori Penunjang**

Pada bab ini akan dijelaskan tentang teori penunjang yang digunakan dalam pembuatan skripsi ini. Teori tersebut antara lain mengenai Mikrokontroler ATmega 8535, RS-232, *Driver motor Stepper*, *Motor Stepper*, Motor DC, Sistem Mekanik dan *Borland Delphi*.

### **BAB III Perancangan Dan Pembuatan Alat**

Dalam bab ini akan diuraikan tentang tahap perencanaan serta proses pembuatan perangkat keras (*hardware*) dan pembuatan perangkat lunak (*software*).

### **BAB IV Pengujian Perangkat Keras Dan Perangkat Lunak**

Bab ini membahas tentang pengujian rangkaian pada masing-masing blok dan secara menyeluruh sehingga kekurangan dari alat tersebut dapat diketahui. Serta berisi data dan analisa dari hasil pengujian yang telah dilakukan.

### **BAB V Penutup**

Bab ini berisi tentang kesimpulan dari pembahasan permasalahan dan saran-saran untuk perbaikan dan penyempurnaan pembuatan alat.

## **BAB II**

### **TEORI PENUNJANG**

Pada bab ini akan dibahas mengenai teori penunjang dari peralatan yang direncanakan. Pokok pembahasan pada bab ini adalah :

1. Mikrokontroler AVR ATmega 8535.
2. *Serial Port RS-232*
3. *Optocoupler IC 4N25*
4. *Driver Motor Stepper*
5. *Motor Stepper*
6. Motor DC
7. *Limit Switch*
8. Sistem Mekanik
9. *Borland Delphi*

#### **2.1 Mikrokontroler AVR ATmega 8535.**

Mikrokontroler ATmega 8535 merupakan salah satu keluarga AVR yang mempunyai 4 kelompok yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega dan AT86RFxx. ATmega 8535 ini mempunyai sistem mikroprosesor 8 bit dengan kecepatan maksimal 16 MHz, kapabilitas *memory flash* 8 KB, SRAM sebesar 512 byte dan EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 512 byte. Atmega 8535 juga mempunyai ADC *internal* dengan fidelitas 10 bit sebanyak 8 *channel*. IC mikrokontroler ini mempunyai kemampuan untuk berkomunikasi *Serial Interface* ( USART ) dengan kecepatan maksimal 2,5 Mbps.

### 2.1.1 Arsitektur AVR ATmega 8535

AVR merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Hampir semua instruksi dieksekusi dalam satu *siklus clock*. AVR mempunyai 32 register general-purpose, timer / counter fleksibel dengan mode compare, interrupt internal dan eksternal, serial UART, programmable Watchdog Timer, dan mode power saving. Beberapa diantaranya mempunyai ADC dan PWM internal. AVR juga mempunyai *In-System Programmable Flash on-chip* yang mengijinkan memory program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI. Chip AVR yang digunakan untuk Skripsi ini adalah Atmega8535.

Atmega 8535 adalah mikrokontroler CMOS 8-bit daya-rendah berbasis arsitektur RISC yang ditingkatkan. Kebanyakan instruksi dikerjakan pada satu siklus *clock*, Atmega 8535 mempunyai throughput mendekati 1 MIPS per MHz membuat disainer sistem untuk mengoptimasi komsumsi daya *versus* kecepatan proses. Mikrokontroler Atmega 8535 memiliki sejumlah keistimewaan sebagai berikut :

#### 1. *Advanced RISC Architecture*

- 130 Powerful Instructions – Most Single Clock Cycle Execution.
- 32 x 8 General Purpose Working Registers.
- Fully Static Operation.
- Up to 16 MIPS Throughput at 16 MHz.
- On-chip 2-cycle Multiplier

**2. Nonvolatile Program and Data Memories**

- *8K Bytes of In-System Self-Programmable Flash*
  - *Endurance: 10,000 Write/Erase Cycles*
- *Optional Boot Code Section with Independent Lock Bits*
  - *In-System Programming by On-chip Boot Program*
  - *True Read-While-Write Operation*
- *512 Bytes EEPROM*
  - *Endurance: 100,000 Write/Erase Cycles*
- *512 Bytes Internal SRAM*
- *Programming Lock for Software Security*

**3. Peripheral Features**

- *Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes.*
- *One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode.*
- *Real Time Counter with Separate Oscillator.*
- *Four PWM Channels.*
- *8-channel, 10-bit ADC.*
  - *8 Single-ended Channels*
  - *7 Differential Channels for TQFP Package Only*
  - *2 Differential Channels with Programmable Gain at 1x, 10x, or 200x for TQFP Package Only*
- *Byte-oriented Two-wire Serial Interface.*
- *Programmable Serial USART.*

- *Master/Slave SPI Serial Interface.*
- *Programmable Watchdog Timer with Separate On chip Oscillator.*
- *On-chip Analog Comparator*

**4. Special Microcontroller Features**

- *Power-on Reset and Programmable Brown-out Detection.*
- *Internal Calibrated RC Oscillator.*
- *External and Internal Interrupt Sources.*
- *Six Sleep Modes: Idle, ADC Noise Reduction, Power save, Power-down,Standby and Extended Standby*

**5. I/O and Packages.**

- *32 Programmable I/O Lines.*
- *40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44 pad MLF*

**6. Operating Voltages**

- *2.7 - 5.5V for ATmega8535L.*
- *4.5 - 5.5V for ATmega8535*

**7. Speed Grades**

- *0 - 8 MHz for ATmega8535L.*
- *0 - 16 MHz for ATmega8535*

### 2.1.2 Pin – pin Atmega 8535

Berikut merupakan gambar 40 pin mikrokontroler Atmega 8535 yang ditunjukkan pada gambar 2.1. dibawah ini :

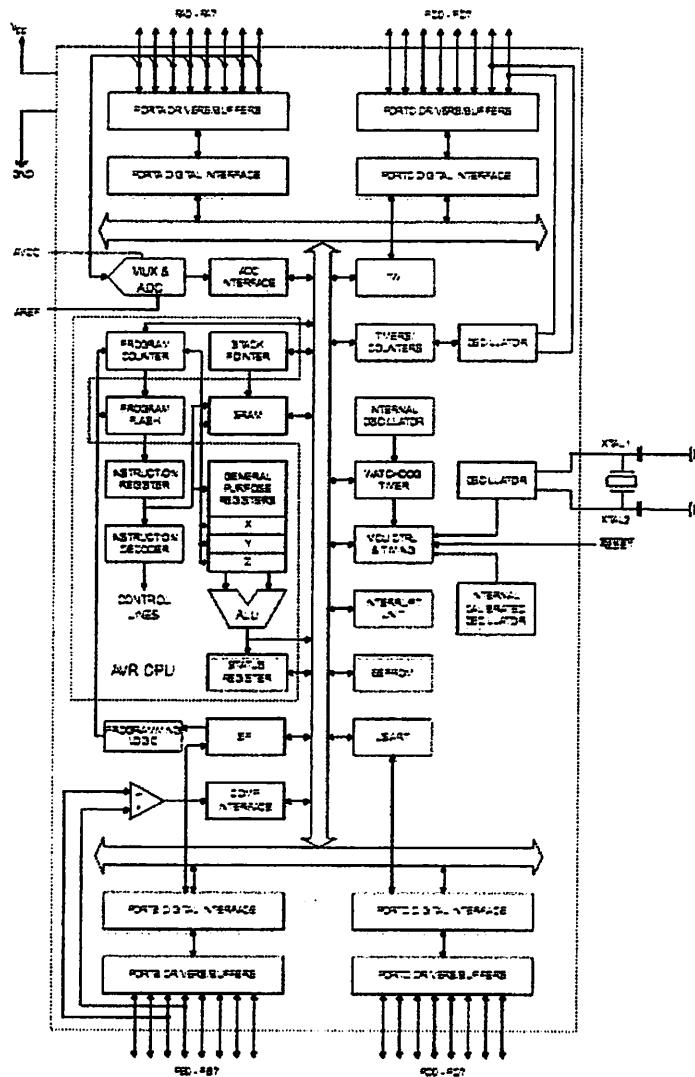
PDIP			
(XCK/T0) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
(SS) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5
(TXD) PD1	15	26	PC4
(INT0) PD2	16	25	PC3
(INT1) PD3	17	24	PC2
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP1) PD6	20	21	PD7 (OC2)

Gambar 2.1 Pin – pin Atmega 8535 kemasan 40-pin

Sumber : [www.datasheetatmega8535.com](http://www.datasheetatmega8535.com)

Pin-pin pada Atmega 8535 dengan kemasan 40-pin DIP (*dual in-line package*) ditunjukkan oleh gambar 2.1. Guna memaksimalkan performa dan *paralelisme*, AVR menggunakan arsitektur *Harvard* (dengan *memory* dan *bus* terpisah untuk program dan data). Arsitektur CPU dari AVR ditunjukkan oleh gambar 2.2 Instruksi pada memori program dieksekusi dengan *pipelining single level*. Selagi sebuah instruksi sedang dikerjakan, instruksi berikutnya diambil dari memori program.

Berikut merupakan gambar Arsitektur mikrokontroler Atmega 8535 yang ditunjukkan pada gambar 2.2. dibawah ini :



Gambar 2.2 Arsitektur ATmega 8535  
Sumber : [www.datasheetatmega8535.com](http://www.datasheetatmega8535.com)

### 2.1.3 Deskripsi Pin – pin ATMega 8535

- **VCC (power supply)**
- **GND (ground)**
- **Port A (PA7..PA0)**

Port A berfungsi sebagai input analog pada A/D Converter.

Port A juga berfungsi sebagai suatu Port I/O 8-bit dua arah, jika A /

D *Converter* tidak digunakan. Pin - pin Port dapat menyediakan resistor *internal pull-up* (yang dipilih untuk masing-masing bit). Port A output *buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Ketika pin PA0 ke PA7 digunakan sebagai input dan secara *eksternal* ditarik rendah, pin – pin akan memungkinkan arus sumber jika resistor internal *pull-up* diaktifkan. Pin Port A adalah *tri-stated* manakala suatu kondisi *reset* menjadi aktif, sekalipun waktu habis.

➤ **Port B (PB7..PB0)**

Port B adalah suatu Port I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Port B output *buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Sebagai input, pin port B yang secara *eksternal* ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. Pin Port B adalah *tri-stated* manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

➤ **Port C (PC7..PC0)**

Port C adalah suatu Port I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Port C output *buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Sebagai input, pin port C yang secara *eksternal* ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. Pin Port C adalah *tri-stated* manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

### ➤ Port D (PD7..PD0)

Port D adalah suatu Port I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Port D output *buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Sebagai input, pin port D yang secara eksternal ditarik rendah akan arus.

#### 2.1.4 Port Sebagai *Input/Output* Digital

Atmega 8535 mempunyai empat buah port yang bernama PortA, PortB, PortC, dan PortD. Keempat port tersebut merupakan jalur *bi-directional* dengan pilihan *internal pull-up*.

Tiap port mempunyai tiga buah register bit, yaitu DDxn, PORTxn, dan PINxn. Huruf ‘x’ mewakili nama huruf dari port sedangkan huruf ‘n’ mewakili nomor bit.

Bit DDxn terdapat pada I/O address DDRx, bit PORTxn terdapat pada I/O address PORTx, dan bit PINxn terdapat pada I/O address PINx. Bit DDxn dalam *register DDRx (Data Direction Register)* menentukan arah pin.

Bila DDxn diset 1 maka Px berfungsi sebagai pin output. Bila DDxn diset 0 maka Px berfungsi sebagai pin input. Bila PORTxn diset 1 pada saat pin terkonfigurasi sebagai pin input, maka resistor *pull-up* akan diaktifkan. Untuk mematikan resistor *pull-up*, PORTxn harus diset 0 atau pin dikonfigurasi sebagai pin output. Pin port adalah *tri-state* setelah kondisi reset. Bila PORTxn diset 1 saat pin terkonfigurasi sebagai pin output maka pin port akan berlogika 1. Dan bila PORTxn diset

0 pada saat pin terkonfigurasi sebagai pin output maka pin port akan berlogika 0.

Saat mengubah kondisi port dari kondisi *tri-state* ( $DDxn=0$ ,  $PORTxn=0$ ) ke kondisi output high ( $DDxn=1$ ,  $PORTxn=1$ ) maka harus ada kondisi peralihan apakah itu kondisi *pull-up* enabled ( $DDxn=0$ ,  $PORTxn=1$ ) atau kondisi output low ( $DDxn=1$ ,  $PORTxn=0$ ). Biasanya, kondisi *pull-up* enabled dapat diterima sepenuhnya, selama lingkungan impedansi tinggi tidak memperhatikan perbedaan antara sebuah *strong high driver* dengan sebuah *pull-up*. Jika ini bukan suatu masalah, maka bit PUD pada *register* SFIOR dapat diset 1 untuk mematikan semua *pull-up* dalam semua port.

Peralihan dari kondisi input dengan *pull-up* ke kondisi output *low* juga menimbulkan masalah yang sama. Kita harus menggunakan kondisi *tri-state* ( $DDxn=0$ ,  $PORTxn=0$ ) atau kondisi output *high* ( $DDxn=1$ ,  $PORTxn=0$ ) sebagai kondisi transisi.

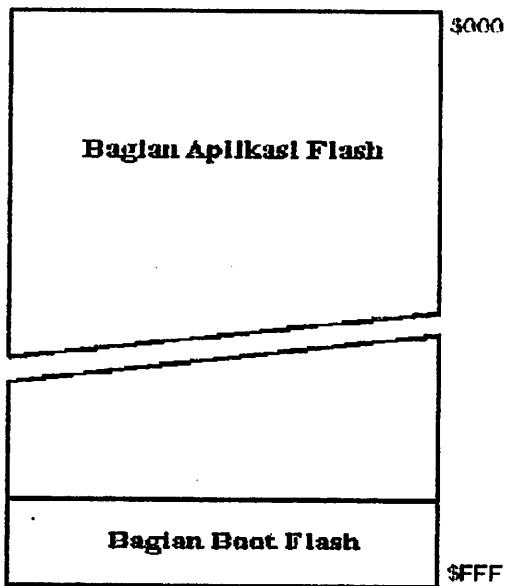
Bit 2 – PUD : *Pull-up Disable*. Bila bit diset bernilai 1 maka *pull-up* pada port I/O akan dimatikan walaupun *register* DDxn dan PORTxn dikonfigurasikan untuk menyalakan *pull-up* ( $DDxn=0$ ,  $PORTxn=1$ ).

### 2.1.5 Organisasi Memori AVR ATmega8535

AVR arsitektur mempunyai dua ruang memori utama, Ruang Data Memori dan Ruang Program Memori. Sebagai tambahan, ATmega8535 memiliki fitur suatu EEPROM Memori untuk penyimpanan data. Semua tiga ruang memori adalah *reguler* dan *linier*.

### 2.1.5.1 Program Memori

ATmega 8535 berisi 8K bytes *On-Chip* di dalam sistem Memori *flash Reprogrammable* untuk penyimpanan program. Karena semua AVR instruksi adalah 16 atau 32 bits lebar, *Flash* adalah berbentuk 4K x 16. Untuk keamanan perangkat lunak, *Flash* Ruang program memori adalah dibagi menjadi dua bagian, bagian *boot* program dan bagian aplikasi program. *Flash Memory* mempunyai suatu daya tahan sedikitnya 10,000 *write/erase Cycles*. ATmega 8535 *Program Counter* (PC) adalah 12 bit lebar, alamat ini 4K lokasi program memori. Pemetaan program memori pada Atmega 8535 seperti pada gambar 2.3 dibawah ini :



Gambar 2.3 Pemetaan program memori  
Sumber : [www.datasheetatmega8535.com](http://www.datasheetatmega8535.com)

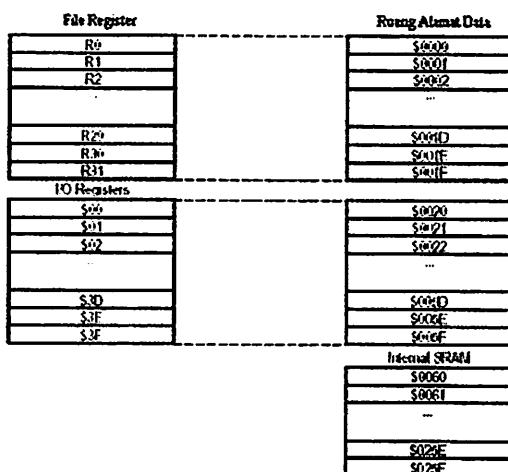
### 2.1.5.2 Data Memori

608 lokasi alamat data memori menunjuk *file register*, I/O memori, dan internal data SRAM. Yang pertama 96 lokasi alamat

*file register* dan I/O memori penempatan menunjuk Memori I/O dan yang berikutnya 512 lokasi alamat internal data SRAM.

Lima perbedaan mode pengalamatan data memori cover: Langsung, Tidak langsung dengan jarak, Tidak langsung dengan *Pre-Decrement*, dan Tidak langsung dengan *Post-Increment*. Di dalam *file register*, *register R26 ke R31* memiliki fitur penunjukan pengalamatan *register* tidak langsung.

Jangkauan pengalamatan langsung adalah keseluruhan ruang data. Mode Tidak langsung dengan jarak jangkauan 63 lokasi alamat dari alamat dasar yang diberi oleh Y-*Register* atau Z-*Register*. Manakala penggunaan *register* mode tidak langsung dengan *pre-decrement* otomatis dan *postincrement*, alamat *register* X, Y, dan Z adalah *decremented* atau *incremented*. 32 tujuan umum kerja *register*, 64 I/O *register*, dan 512 bytes data *internal SRAM* di dalam ATmega8535 adalah semua dapat diakses melalui semua mode pengalamatan.



Gambar 2.4 Pemetaan Data Memori  
Sumber : [www.datasheet.atmega8535.com](http://www.datasheet.atmega8535.com)

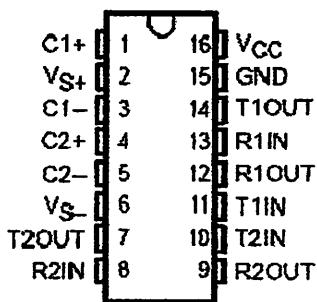
## 2.2 Serial Port RS-232

Max 232 merupakan pengubah TTL ke level tegangan RS-232. Max 232 memiliki dua *driver* mengonversikan RS-232 ke level TTL, dan dua penerima yang merubah TTL ke RS-232. Max 232 memiliki 16 pin dan dioperasikan dengan empat buah kapasitor yang memiliki nilai  $1\mu F$ .

RS-232 merupakan salah satu jenis antar muka (*interface*) dalam proses transfer data antar perangkat dalam bentuk serial transfer. RS-232 merupakan singkatan dari *Recommbed Standard number 232*. Alat ini dibuat oleh *Electronic Industry Assosiation*, untuk *Interface* antara peralatan *terminal data* dan peralatan komunikasi data, dengan menggunakan *data binner serial* sebagai data yang ditransmisikan IC MAX 232 ini mempunyai empat buah bagian konverter yaitu dua buah *driver receiver* dan dua buah *driver transmitter*.

Saluran kabel data menggunakan *standard RS-232*, dimana *logic 0 (low)* dinyatakan sebagai tegangan antara +3 volt sampai +10 volt, dan *logic 1 (high)* dinyatakan sebagai tegangan antara -3 volt sampai -10 volt. *Level* tegangan ini tidak sesuai dengan tegangan *level* tegangan yang dipakai pada port port *seri* ATMega8535 yang menggunakan *Standard TTL (Transistor Transistor Logic)*, yaitu *level* tegangan baku dalam rangkaian – rangkaian *digital*.

Dalam *standard TTL*, *logic 0 (low)*, dinyatakan sebagai tegangan antara 0 volt sampai 0.8 volt, dan *logic 1 (high)* dinyatakan sebagai tegangan antara 3.5 volt sampai 5 volt. Untuk dua MCU yang dihubungkan secara *serial* pada jarak tertentu maka dibutuhkan IC MAX 232 karena *level* tegangan TTL terlalu kecil untuk ditransfer.

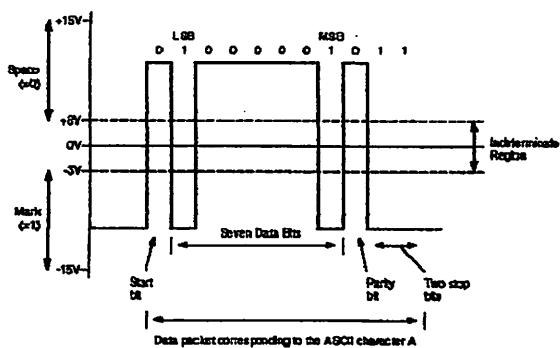


Gambar 2.5. IC MAX 232

Sumber : [www.datasheet\\_max232.com](http://www.datasheet_max232.com)

RS-232 merupakan seperangkat alat yang berfungsi sebagai *interface* dalam proses *transfer data* secara *serial*. Metode pengiriman secara *serial RS-232* adalah *asinkron*. Pengiriman *asinkron* berarti tidak membutuhkan waktu sebagai *sinkronisasi*. Dalam pengiriman secara *serial asinkron*, *clock* dapat dikirimkan, tetapi dikondisikan oleh *timing start bit* yang merupakan isyarat dari sumber ke tujuan untuk mendekodekan adanya pengiriman karakter sudah selesai dikirim.

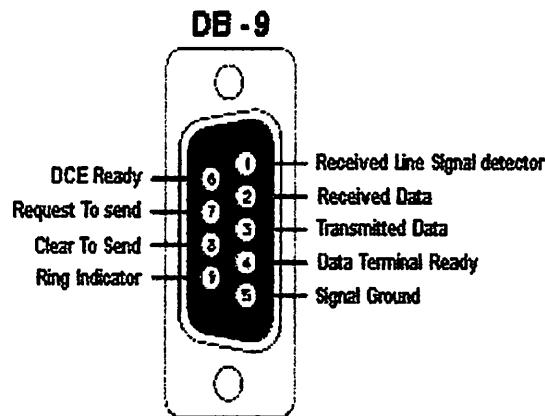
Karakteristik *electris* dan sistem RS-232 adalah mempunyai tegangan keluaran antara -15 volt sampai dengan +15 volt. Tegangan +3 sampai +15 volt untuk logika '0' / *spacing* dan tegangan -3 sampai -15 volt untuk logika '1' / *marking*. Hal tersebut dinyatakan dalam gambar 2.6. dibawah ini :



Gambar 2.6. 1 byte of Async Data

Sumber : [www.datasheet\\_max232.com](http://www.datasheet_max232.com)

ada pin out menggunakan koneksi kabel data *serial* yang menggunakan *Standard RS-232*. Fasilitas ini menggunakan konektor DB-9. Gambar konektor DB-9 seperti terdapat dalam gambar 2.7.



Gambar 2.7. Konektor DB-9

Sumber : Interfacing Komputer dan mikrokontroler.

Fungsi masing – masing pin pada DB-9 seperti dalam tabel 2.1 dibawah ini:

Tabel 2.1 Fungsi Pin RS-232 dalam DB-9

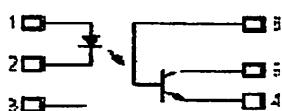
Pin	Nama	Fungsi
1	DCD ( <i>Data Carrier Detect</i> )	Mendeteksi Sinyal Carrier dari modem lain
2	RD ( <i>Received Data Line</i> ) / (RXD)	Pengiriman data serial dari DCE ke DTE
3	TD ( <i>Transmitted Data Line</i> ) / (TXD)	Pengiriman data serial dari DTE ke DCE
4	DTR ( <i>Data Terminal Ready</i> )	Memberitahukan DCE bahwa DTE telah aktif dan siap untuk bekerja
5	Ground	Referensi semua tegangan antar muka
6	DSR ( <i>Data Set Ready</i> )	Memberitahukan DTE bahwa DCE telah aktif dan siap untuk bekerja
7	RTS ( <i>Request To Send</i> )	Memberitahukan DCE bahwa DTE akan mengirim data
8	CTS ( <i>Clear To Send</i> )	Memberitahukan DTE bahwa DCE siap menerima data
9	RI ( <i>Ring Indikator</i> )	Aktif jika <i>modem</i> menerima sinyal ring jalur telepon

Sumber : Interfacing Komputer dan mikrokontroler.

Jalur data (TXD dan RXD) untuk *transport* data, TXD adalah jalur output serial handphone, data dikirim dari pin ini. Sedangkan RXD adalah penerima untuk *handphone*, data yang datang akan diterima oleh pin ini. Pin ke empat adalah *output* (RTS) dimana sebuah sinyal akan diberikan pada alat yang dihubungkan dengan maksud meminta kiriman data. CTS adalah sinyal masukan yang menunggu sinyal dari alat yang terhubung ketika alat tersebut menerima sinyal RTS dan bisa menerima data maka ia akan mengirimkan sinyal balik yang merupakan CTS. DTR adalah sinyal keluaran yang memberi tanda bahwa ada alat yang terhubung dan akan mengirimkan data. DSR merupakan sinyal input yang mana jika alat yang terhubung menerima sinyal DTR maka akan memberi sinyal balik kemudian diterima sebagai sinyal DSR.

### 2.3 Optocoupler IC 4N25

*Optocoupler* merupakan sebuah perangkat elektronik yang terdiri dari dua buah komponen yaitu LED (*Light Emitting Diode*) sebagai sumber cahaya *Infrared* (IR) dan photo transistor sebagai penangkap cahaya *Infrared*. Kekuatan cahaya yang dihasilkan LED tergantung dari arus yang mengalir pada LED tersebut. Sedangkan arus kolektor pada transistor juga ditentukan oleh kekuatan intensitas cahaya IR dari LED tersebut, berikut skematik dari *optocoupler* :



- PIN 1: LED ANODE
- 2: LED CATHODE
- 3: N.C.
- 4: Emitter
- 5: Collector
- 6: Base

Gambar 2.8. Skematik *optocoupler* 4N25

Sumber : [www.jameco.com](http://www.jameco.com)

Pada gambar 2.8. dapat dilihat rangkaian elektrik dalam kemasan *optocoupler* 4N25. Penempatan antara LED dan *phototransistor* diberi jarak tertentu yang digunakan untuk melewaskan berkas cahaya dari LED ke *phototransistor*. Celah kosong ini berfungsi sebagai jarak agar jalur elektrik LED tidak terhubung dengan *phototransistor*. Komponen ini biasa digunakan sebagai pengendali terhadap beban yang bertegangan lebih tinggi dari tegangan sistem kontrol.

Adapun cara kerja dari *optocoupler* yaitu bila cahaya dari LED mati maka *phototransistor* tidak dapat bekerja atau tersumbat (*Cutoff*), sehingga arus kolektor tidak mengalir. Sebaliknya jika cahaya dari LED menyala maka *phototransistor* akan bekerja (*saturation*), sehingga arus kolektor akan mengalir. Disini *optocoupler* digunakan untuk mengisolasi mikrokontroler ATMega8535 dari adanya arus balik yang terjadi pada rangkaian *driver* motor dan dengan adanya tambahan *diode* diharapkan dapat mencegah tegangan balik yang dihasilkan kumparan *relay* sesaat setelah dimatikan.

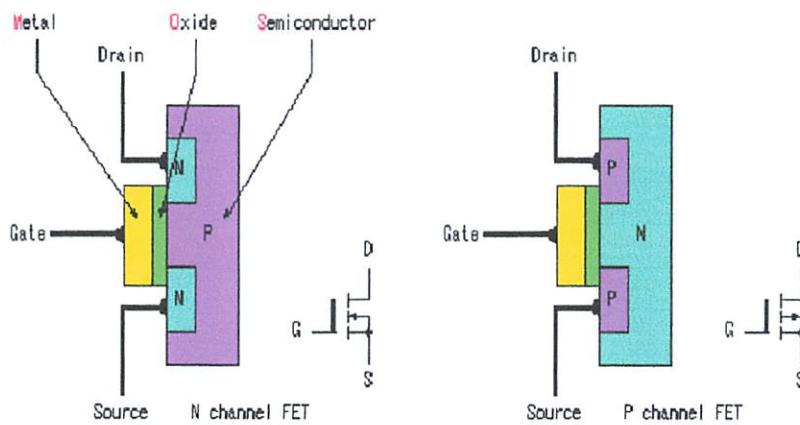
## 2.4 *Driver Motor Stepper*

*Driver motor stepper* yang digunakan adalah *driver motor stepper* berbasis transistor MOSFET IRFZ44 seperti yang dijelaskan dibawah ini :

### 2.4.1 *The Metal Oxide Semiconductor FET (MOSFET)*

Dalam JFET, besar keefektifan pada *channel* dikontrol oleh medan listrik yang diberikan ke *channel* melalui P-N *junction*. Bentuk lain dari piranti pengaruh medan dicapai dengan penggunaan bahan *elektroda gate* yang dipisahkan oleh lapisan *oxide* dari *channel*.

semikonduktor. Pengaturan metal oxide semikonduktor (*MOS*) mengijinkan karakteristik *channel* dikontrol oleh medan listrik dengan memberikan tegangan diantara *gate* dan *body semikonduktor* dan pemindahan melalui lapisan *oxide*. Seperti halnya piranti yang disebut dengan *MOSFET* atau *MOS Transistor*. Hal ini penting digaris bawahi dengan kenyataan bahwa *IC* lebih banyak dibuat dengan piranti *MOS* dari pada jenis piranti semikonduktor lain. Jenis *MOSFET* tipe *N channel* dan *P channel* ditunjukkan pada gambar 2.9. dibawah ini :



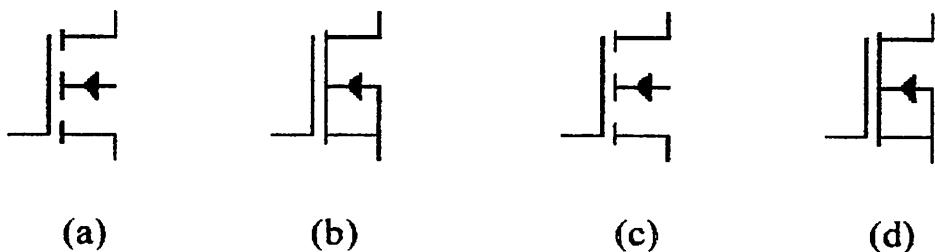
Gambar 2.9. Jenis MOSFET  
Sumber : [www.datasheet.irfz44n.com](http://www.datasheet.irfz44n.com)

Ada dua tipe *MOSFET*. *Deplesi MOSFET* mempunyai tingkah laku yang sama dengan *JFET* pada saat tegangan *gate* nol dan tegangan *drain* tetap, arus akan maksimum dan kemudian menurun dengan diberikan potensial *gate* dengan *polaritas* yang benar (piranti *normally on*). Jenis yang lain dari piranti ini disebut dengan *Enhancement MOSFET* yang menunjukkan tidak ada arus pada saat tegangan *gate* nol dan besar arus keluaran besar dengan bertambah besar potensial *gate*

(*normally off*). Kedua tipe dapat berada dalam salah satu jenis *channel P* atau *N*.

#### 2.4.2 Simbol Rangkaian MOSFET

Terdapat 4 simbol yang digunakan untuk *MOSFET channel N* yang ditunjukkan pada Gambar 2.10. Simbol-simbol dalam Gambar 2.10. (a) dan (b) dapat digunakan untuk *enhancement* dan *depletion device*. Simbol rangkaian dalam Gambar 2.10. (c) hanya digunakan pada piranti mode enhancement. Sedangkan pada Gambar 2.10. (d) juga digunakan pada piranti deplesi *MOSFET*. *Mosfet* tipe *N channel* ditunjukkan pada gambar 2.10 dibawah ini :

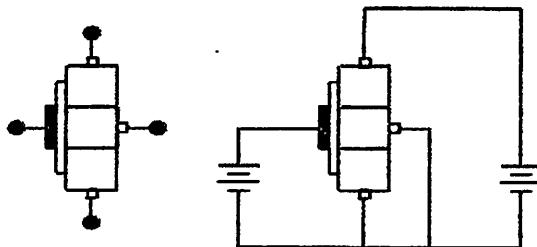


Gambar 2.10. Simbol *MOSFET channel N*.

Sumber : [www.datasheet irfz44n.com](http://www.datasheet irfz44n.com)

#### 2.4.3 Karakteristik MOSFET

*MOSFET* adalah semikonduktor *FET oksida logam* yang mempunyai sumber, gerbang dan penguras. Akan tetapi berbeda dengan *JFET*, gebang *MOSFET* diisolasi dengan saluran. Maka arus gerbang sangat kecil, untuk gerbang positif atau negatif. *MOSFET* yang sangat penting dalam rangkaian-rangkaian digital dikenal dengan *MOSFET* jenis pengisian pada Gambar 2.11.



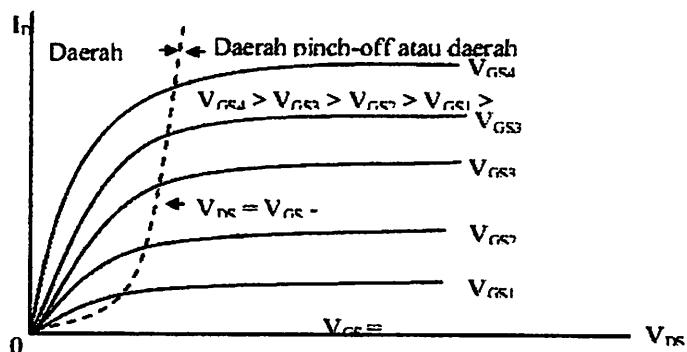
Gambar 2.11 . MOSFET jenis pengisian.  
Sumber : [www.datasheet irfz44n.com](http://www.datasheet irfz44n.com)

Untuk memperoleh arus penguras, kita harus menerapkan tegangan yang cukup positif pada gerbang. Gerbang akan bekerja sebagai sebuah pelat kapasitor, *dioksida silikon* bekerja sebagai bahan *dielektrik* dan *substrat-p* sebagai pelat kapasitor yang lain.

Lapisan elektron bebas yang terbentuk berdampingan dengan dioksida silikon. Lapisan ini tidak bekerja sebagai suatu semikonduktor tipe-p melainkan nampak sebagai konduktor tipe-n disebabkan oleh elektron-elektron bebas yang diimbangi. Maka lapisan bahan p yang bersinggungan dengan dioksida silikon disebut sebuah lapisan inversi tipe- n.

Tegangan gerbang sumber minimum yang menghasilkan lapisan inversi tipe-n disebut tegangan ambang (*threshold voltage*) dinyatakan dengan  $V_{gs}$ . Apabila tegangan gerbang kurang dari tegangan ambang, tidak ada arus yang mengalir dari sumber ke penguras. Tetapi apabila tegangan gerbang lebih besar dari tegangan ambang, lapisan inversi tipe-n menghubungkan sumber ke penguras dan diperoleh arus. Tegangan ambang tergantung pada tipe khusus dari *MOSFET*. Untuk *MOSFET IRFZ44* tegangan ambangnya 2v - 6v.

Gambar 2.12. memperlihatkan suatu kumpulan dari kurva-kurva penguras *MOSFET* jenis pengisian. Lengkungan yang paling bawah adalah kurva  $V_{gs}$ . Apabila  $V_{gs}$  kurang dari  $V_{gs0}$  arus penguras secara ideal sama dengan nol *dan MOSFET berada pada keadaan tertentu*. Apabila  $V_{gs}$  lebih besar dari  $V_{gs0}$  arus penguras muncul.



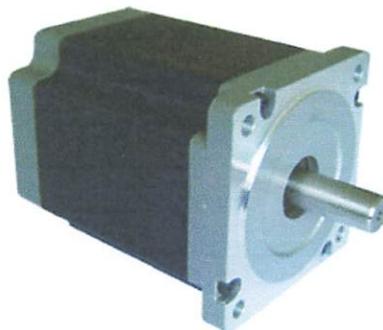
Gambar 2.12. Karakteristik keluaran *MOSFET* tipe enhancement.  
Sumber : [www.datasheet irfz44n.com](http://www.datasheet irfz44n.com)

## 2.5 Motor Stepper

### 2.5.1 Deskripsi

Motor Stepper merupakan peralatan elektromekanis yang dapat mengubah energy listrik menjadi suatu putaran per step. Motor ini harus dihubungkan sebuah driver yang memiliki kemampuan *switching* tinggi. Driver tersebut akan dihubungkan dengan mikrokontroler atau pengontrol digital yang akan memberikan pulsa untuk menggerakkan motor. Tiap-tiap pulsa yang diberikan oleh mikrokontroler akan mengerakkan penyangga motor sesuai dengan nilai pergeseran sudut yang ditentukan. Nilai pergeseran tersebut disebut dengan *step angel* yang nilainya berbeda untuk tiap-tiap motor stepper. Pengeturan posisi dan kecepatan pada motor stepper dapat dilakukan dengan mengatur *delay* pulsa dari frekuensi melalui mikro.

Motor stepper memiliki konfigurasi dan jenis yang beragam sesuai kebutuhan sang pengguna. Terdapat Motor stepper yang memiliki *torsi* tinggi dengan kecepatan rendah dan *step angel* kecil, adapula motor stepper dengan kecepatan tinggi dan *torsi* rendah. Berikut ini adalah bentuk dasar dari motor stepper yang ditemui dipasaran :



Gambar 2.13. Motor Stepper

Sumber : <http://homepages.which.net/~pul.hills/Motor/Motor.html>

Stepper motor memiliki *step angel* yang bervariasi. Mulai dari yang bernilai  $0.9^\circ$  per step hingga  $90^\circ$  per step. Pada umumnya nilai *step angel* motor stepper yang ada dipasaran sebesar  $1.8^\circ$  per step dengan jumlah per rotasi 2 sebesar 200. Sebenarnya nilai pergeseran step juga dapat digunakan untuk menentukan banyaknya step dalam satu putaran. Misalnya motor stepper dengan *step angel*  $1.8^\circ$  akan berputar sebanyak 200 kali dalam satu putaran ( $1.8^\circ \times 200 = 360^\circ$ ) sedangkan motor stepper dengan *step angel*  $0.9^\circ$  akan berputar sebanyak 400 kali dalam satu putaran ( $0.9^\circ \times 400 = 360^\circ$ ).

Sebenarnya semakin kecil *step angel* dari motor stepper akan menambah akurasi dari motor stepper. Namun, motor stepper juga memiliki batas nilai pulsa tiap detik yang dapat diberikan padanya supaya bergerak secara optimal. Pada umumnya motor stepper berukuran besar

dengan torsi kuat memiliki rata-rata input pulsa maksimal sebesar 200 hingga 300 pulsa per detik dengan kecepatan rata-rata putaran 60 rpm untuk 1 putaran dan 180 rpm untuk 3 putaran. Beberapa motor stepper dengan ukuran lebih kecil mampu menerima hingga ribuan pulsa tiap detiknya, namun *torsi* yang dimiliki motor tersebut *relative* lebih kecil hingga tidak dapat digunakan putaran yang dihubungkan dengan beban.

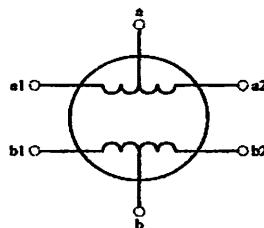
Motor stepper juga memiliki beberapa *coil* yang berperan sebagai induktor ketika mengenai suatu tegangan. *Coil* ini berfungsi untuk menghasilkan medan elektromagnetik yang akan berpengaruh besar pada putaran motor stepper dan *torsi* yang dihasilkan. Besar kecilnya medan elektromagnetik *coil* ini tergantung pada besar kecilnya arus yang diberikan. Sehingga apabila diinginkan torsi maksimal motor stepper maka dapat dilakukan dengan memberikan arus maksimal pada *coil*.

Motor stepper juga dapat digambarkan sebagai sebuah motor tanpa komulator. Seluruh kawat pada motor stepper merupakan bagian dari stator sedangkan rotornya merupakan magnet permanent. Sehingga fungsi komutator sepenuhnya dihandle secara eksternal oleh motor *controller*. Perlu dicatat bahwa motor stepper tidak dapat dikondisikan untuk berputar pada kecepatan maksimal secara langsung pada saat posisi tidak berputar. Sehingga penambahan kecepatan pada motor stepper harus dilakukan secara bertahap. Kecepatan rendah hendaknya diberikan pada 1/3 beberapa *millisecond* pertama, 2/3 pada 50 hingga 75 dan selanjutnya dapat diberi kecepatan maksimal.

## 2.5.2 Jenis Motor Stepper

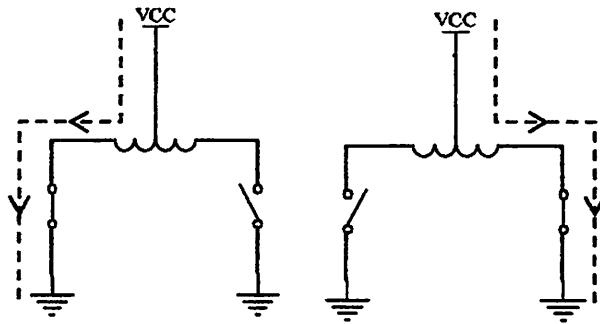
### 2.1.5.1 Stepper Motor Unipolar

*Stepper Motor Unipolar* merupakan sejenis motor stepper yang memiliki 6 katub. Katub-katub tersebut terdiri dari 4 katub *phasa* dan 2 *common (center tapped winding)*. Ciri khas dari motor stepper jenis ini adalah adanya kutub *common* atau dapat disebut sebagai *center tapped*. Kutub *common* berfungsi untuk menghimpun seluruh *center tapped* untuk dihubungkan pada tegangan motor (*Supply Motor*) kemudian katub-katub yang lain akan dihubungkan pada *ground* untuk memudahkan transfer tegangan dari tegangan motor ke tiap-tiap kutub. System pengkabelan pada unipolar motor dapat dilihat pada gambar dibawah ini :



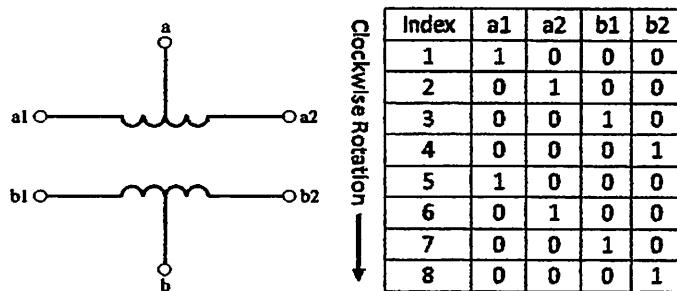
Gambar 2.14. Simbol Stepper Motor Unipolar  
Sumber : <http://homepages.which.net/~pul.hills/Motor/Motor.html>

Stepper Motor Unipolar baik permanent magnet maupun *Hybrid* baik yang memiliki 5 maupun 6 kabel pada umumnya terhubung seperti gambar diatas, yaitu dengan adanya sebuah *center tapped* akan dihubungkan ke katub positif dari catu daya, kemudian katub-katub yang lain akan dihubungkan ke *gound* untuk melakukan pergerakan motor stepper searah maupun berlawanan arah jarum jam seperti gambar 2.17 dibawah ini :



Gambar 2.15. Pembalikan Arus Pada Stepper Motor Unipolar  
Sumber : <http://homepages.which.net/~pul.hills/Motor/Motor.html>

Pada Stepper Motor Unipolar jumlah *phasa* adalah 2 kali jumlah *coil* sejak masing-masing *coil* dibagi menjadi 2. Gambar dibawah ini menunjukan hubungan antara *phasa* dari Stepper Motor Unipolar.



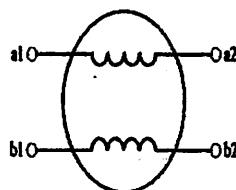
Gambar 2.16. Sekuen Putaran Stepper Motor Unipolar  
Sumber : <http://homepages.which.net/~pul.hills/Motor/Motor.html>

Sebagai tambahan, pada perpindahan step standard juga dimungkinkan adanya perpindahan dengan torsi tinggi dan secara *Halfstep*. Pada perpindahan dengan torsi tinggi 2 buah kawat akan aktif secara bersamaan pada waktu yang sama pada setiap step. Kombinasi 2 kawat ini akan menyebabkan motor bergerak secara *Fullstep* maupun *Halfstep*. Perpindahan secara *Halfstep* terjadi apabila terdapat *aktivasi* bergantian antar tiap kawat. Misalnya

pertama-tama kawat pertama ON sementara kawat yang lain OFF, kemudian kawat kedua ON sementara yang lainnya OFF, dst. Penggantian ON-OFF antar kawat ini akan menyebabkan motor stepper bergerak setengah *step angel* atau *Halfstep*.

### 2.1.5.2 Motor Stepper Bipolar

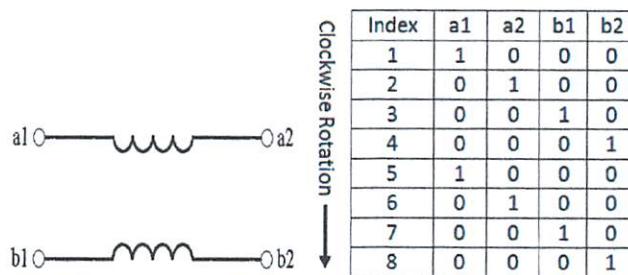
Pada dasarnya kontruksi dari Motor Stepper Bipolar hampir sama dengan Motor Stepper Unipolar. Namun, Motor Stepper Bipolar memiliki system pengkabelan yang lebih sederhana yaitu tanpa menggunakan *center tapped*, sehingga hanya terdapat 4 buah kabel yang terhubung secara silang. Yaitu 1A-1B-2A-2B, 1A terhubung dengan 2A sedangkan 1B terhubung dengan 2B. pada dasarnya motor stepper jenis *Unipolar* dapat diubah menjadi *Bipolar* hanya dengan mengabaikan *common* pin atau menghubungkan *common* pin dengan *gound*. Motor Stepper Bipolar dapat dilihat pada gambar 2.10. dibawah ini :



Gambar 2.17 Simbol Motor Stepper Bipolar  
Sumber : <http://homepages.which.net/~pul.hills/Motor/Motor.html>

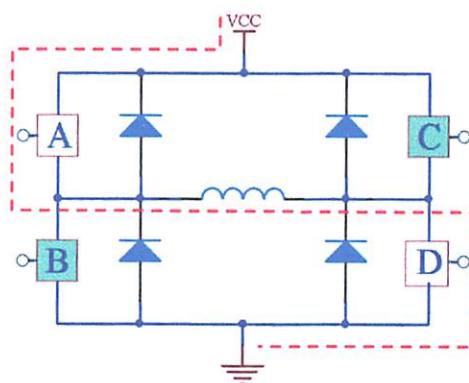
Motor Stepper Bipolar telah dikenal sebagai motor stepper yang memiliki torsi tinggi dan *relative* lebih tinggi dari *motor stepper* manapun namun untuk menggerakkan motor stepper ini diperlukan *driver* yang lebih kompleks daripada rangkaian *driver*

*Motor Stepper Unipolar.* Motor Stepper Bipolar memiliki *coil* terpisah yang dapat digerakkan dengan memberikan tegangan aktif high maupun aktif low pada tiap ujung kawatnya. Kemudian untuk mengubah arah pergerakan motor dapat dilakukan dengan mengubah polaritas tegangan pada ujung kawat. Gambar dibawah ini menunjukkan 4 *phasa coil* dari Motor Stepper Bipolar beserta drive sequence.



Gambar 2.18 Sekuen Putaran Motor Stepper Bipolar  
Sumber : <http://homepages.which.net/~pul.hills/Motor/Motor.html>

Mikrokontroler sebagai pengontrol utama pergerakan motor harus mampu membalik polaritas tegangan pada sepanjang *coil*, sehingga arus dapat mengalir melalui 2 arah. Mekanisme pembalikan tegangan sepanjang 2 *coil* ini dinamakan *H-Bridge*, seperti yang tergambar dibawah ini:



Gambar 2.19 Konsep *H-Bridge* pada Motor Stepper Bipolar  
Sumber : <http://homepages.which.net/~pul.hills/Motor/Motor.html>

### 2.5.3 Tahap Pergerakan Stepper Motor

Table 2.2 dibawah ini menggambarkan 3 macam pergerakan motor stepper polaritas tiap-tiap kutub digambarkan dengan +/- . Pergerakan akan terus berulang pada setiap akhir step.

Tabel 2.2 Tahapan Putaran Motor Stepper

Nama	Langkah	Keterangan
Satu Fasa	0001 0010 0100 1000	Mengkonsumsi sedikit daya. Hanya berfasa tunggal diberi tenaga pada waktu yang sama.
Dua Fasa	0011 0110 1100 1001	Urutan ini memberi tenaga dua fasa bersebelahan. Untuk meningkatkan kecepatan putar dan meningkatkan torsi putar.
Half-Step	0001 0011 0010 0110 0100 1100 1000 1001	Secara efektif menggandakan resolusi langkah motor, tapi torsi tidak seragam untuk setiap langkah. urutan ini mengurangi resonansi motor yang suatu waktu dapat menyebabkan motor macet di frekuensi resonansi tertentu.

Sumber : <http://homepages.which.net/~pul.hills/Motor/Motor.html>

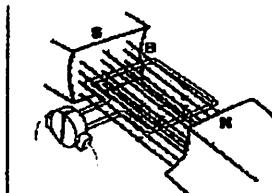
## 2.6 Motor DC

Motor DC pada saat ini digunakan pada industri yang memerlukan gerakan dengan kepresision yang sangat tinggi untuk pengaturan kecepatan pada torsi yang konstan. Motor DC berfungsi mengubah tenaga listrik menjadi tenaga mekanis dimana gerak tersebut berupa putaran dari motor. Prinsip dasar dari motor arus searah adalah kalau sebuah kawat berarus diletakkan antara kutub magnet (U-S), maka pada kawat itu akan bekerja suatu gaya yang menggerakkan kawat itu. Arah gerakkan kawat dapat ditentukan dengan kaidah tangan kiri, yang

berbunyi sebagai berikut: "Apabila tangan kiri terbuka diletakkan diantara kutub U dan S, sehingga garis-garis gaya yang keluar dari kutub utara menembus telapak tangan kiri dan arus didalam kawat mengalir searah dengan arah keempat jari, maka kawat itu akan mendapat gaya yang arahnya sesuai dengan arah ibu jari".

Adapun prinsip kerjanya sebagai berikut:

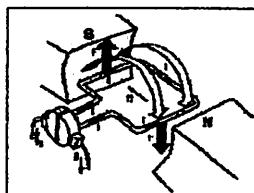
1. Adanya garis – garis gaya medan magnet (*fluks*), antara kutub yang ada di *stator*. Garis – garis gaya medan magnet tersebut mempunyai arah dari kutub magnet positif ke kutub *magnet negative*.



Gambar 2.20 Garis – Garis Medan Magnet  
Sumber : Prinsip-Prinsip Elektronika

2. Pengantar yang dialiri arus ditempatkan pada jangkar yang berada dalam medan magnet tadi, sehingga akan menghasilkan gaya ( F ) dan dapat dirumuskan

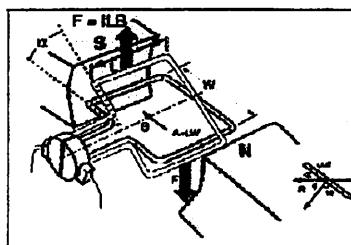
$$F = B * I * L \dots\dots\dots(1)$$



Gambar 2.21 Gaya Yang Dihasilkan Motor DC  
Sumber : Prinsip-Prinsip Elektronika

3. Pada pengantar timbul gaya yang menghasilkan torsi.

$$T = IBA \cdot \sin \theta \dots\dots\dots(2)$$



Gambar 2.22 Torsi Yang Ditimbulkan Motor DC  
Sumber : Prinsip-Prinsip Elektronika

## 2.7 Limit Switch

*Limit switch* adalah sensor yang digunakan pada pendekripsi *limit* dari perubahan mekanik. *Limit* dari sebuah perpindahan dapat dideteksi dengan menggunakan kontak mekanik yang sederhana untuk menutup pulsa sirkuit atau *trigger*. Oleh sebab itu maka informasi yang diberikan oleh *Limit switch* hanyalah dua jenis (*on/off*, ada/tidak) dan dapat dinyatakan dalam satu bit. Dalam hal ini, *Limit switch* terdapat dalam *transduser digital*. penambahan *logic* diperlukan apabila *direction contact* diperlukan. *Limit switch* tersedia baik untuk perpindahan *rectilinear* dan *angular*. *Mikroswitch* adalah *switch solid state* yang dapat digunakan sebagai *Limit switch*. *Mikroswitch* biasanya digunakan pada operasi perhitungan, sebagai contoh untuk perhitungan dari kelengkapan produk pada sebuah rumah usaha. Meskipun *mechanical device* terdiri dari persambungan, roda gigi, *ratchet wheels* dan *pawls*, dan *so forth*, dapat digunakan sebagai sebuah *limit switch*, *electrical* dan *solid state switches* biasanya diutamakan untuk ketelitian, daya tahan, syarat gaya rendah (*practical zero*), harga rendah dan ukuran kecil. Beberapa sensor setara dapat digunakan sebagai elemen dari *Limit switch*. sinyal sensor setara digunakan pada *required manner*, sebagai contoh untuk mengaktifkan *counter*, *mechanical switch*, *relay circuit*, atau sebagai input pada sebuah kontrol komputer.



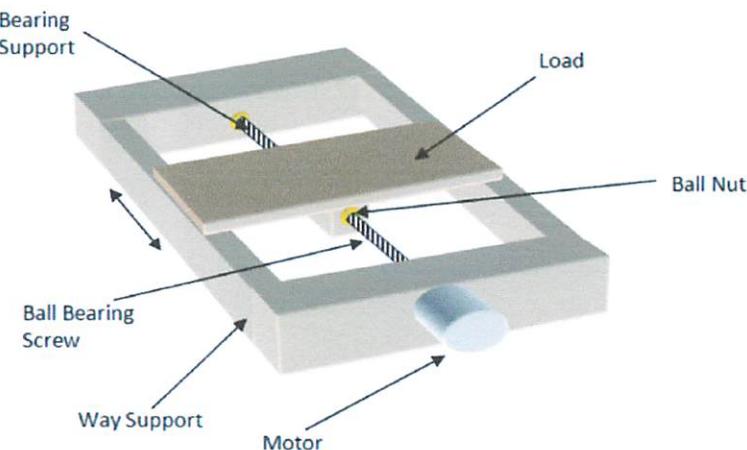
Gambar 2.23. Fisik *Limit Switch*

Sumber : [www.st\\_electronics.com](http://www.st_electronics.com)

## 2.8 Sistem Mekanik

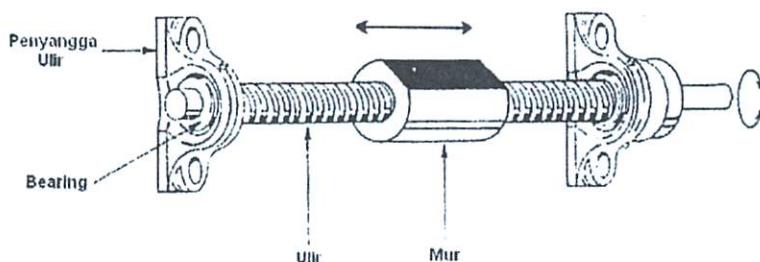
Sistem mekanik pada mesin ini menggunakan sistem kontrol pengeseran ulir yaitu system pengontrolan yang didasarkan pada torsi, kecepatan maupun banyaknya step pada sebuah motor. Sistem penggeseran ulir banyak diimplementasikan pada beberapa jenis motor diantaranya motor DC, Servo maupun Stepper.

Pada sistem digunakan motor stepper dengan pertimbangan akurasi yang tinggi, torsi yang kuat dan pengontrolan yang *relative* mudah yaitu menggunakan *system control loop* terbuka yaitu system kontrol tanpa adanya Umpan Balik. *Motor stepper* dihubungkan pada *lead screw* atau yang dikenal dengan mata ulir sehingga perubahan step pada motor akan mengakibatkan pergeseran beban pada ulir sesuai dengan jumlah step yang dikenakan pada motor. Hubungan antara motor dengan ulir dapat dilihat pada gambar 2.24 dibawah ini.:



Gambar 2.24. Kontruksi Dasar Mekanik  
Sumber : CNC Printed Circuit Board Drilling Machine

Komponen utama untuk mendukung terjadinya pergeseran beban adalah ulir, seperti yang terlihat pada gambar 2.25 dibawah ini :



Gambar 2.25 Kontruksi Ulir  
Sumber : CNC Printed Circuit Board Drilling Machine

## 2.9 Borland Delphi

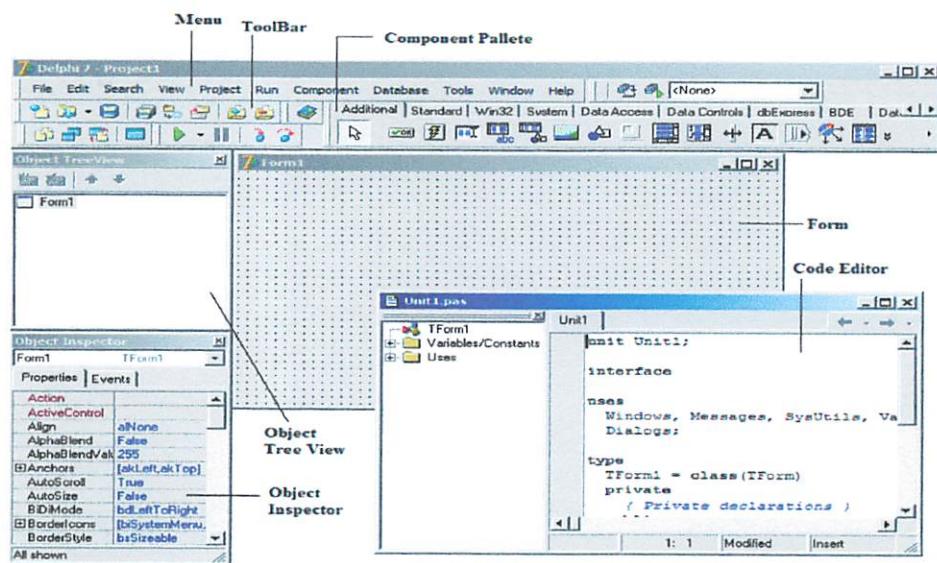
Secara umum, *Borland Delphi* adalah sebuah program untuk membuat aplikasi – aplikasi berbasis *Windows*. Bahasa pengembangan yang digunakan oleh Delphi adalah bahasa *Pascal*. *Turbo Pascal* dikenal dengan kelebihan dalam kecepatan eksekusi dan kompilasi, dibandingkan dengan bahasa pemrograman lain yang berkembang saat ini. *Integrated Development Environment* (IDE) yang diperkenalkan dan diterapkan oleh *Turbo Pascal* sangat memudahkan para *programmer* merealisasikan program aplikasi mereka. Dengan IDE seorang

programer dapat dengan cepat dan mudah menulis kode program, melakukan kompilasi, melihat kesalahan (*error*) program, serta langsung menuju letak kesalahan dan memperbaiki kesalahan tersebut. Kemudian *Turbo Pascal* dirubah menjadi yang berorientasi *obyek* (*Object Oriented Programming*) berbasis tampilan *visual* yang menarik, dan dilengkapi kemampuan akses ke basis data. Inilah yang kemudian dikenal sebagai Delphi.

Delphi dapat digolongkan ke dalam bahasa tingkat tinggi (*High Type Language*) karena segala kemudahan ditawarkan untuk perancangan sebuah aplikasi.

### 2.9.1 IDE (*Integrated Development Enviroment*)

*IDE* adalah sebuah lingkungan yang berisi *tool – tool* yang diperlukan untuk desain, menjalankan dan mengetes sebuah aplikasi, disajikan dan terhubung dengan baik sehingga memudahkan pengembangan program. Di Delphi, *IDE* terdiri dari :



Gambar 2.26. IDE (*Integrated Development Enviroment*)  
Sumber : Pemrograman Borland Delphi7

**a. Main Window**

*Main Window* adalah bagian utama dari *IDE*. *Main Window* mempunyai semua fungsi utama dari program – program *Windows* lainnya.

**b. Menu Utama**

Menu utama dipakai untuk membuka atau menyimpan *file*, memanggil *wizard*, menampilkan jendela lain, mengubah option dan lain sebagainya.

**c. Toolbar**

Dengan menu *toolbar* dapat melakukan beberapa operasi pada menu utama yang setiap tombol berisi informasi mengenai fungsi dari tombol tersebut.

**d. Form Designer**

Jendela kosong yang digunakan untuk merancang aplikasi *Windows*.

**e. Code Editor**

Merupakan bagian yang terpenting di lingkungan Delphi. Jendela ini dipakai untuk menuliskan program Delphi.

**f. Code Explorer**

*Code explorer* digunakan untuk memudahkan *navigasi* didalam *file* unit.

**g. Object Treeview**

Merupakan daftar dari komponen-komponen apa saja yang telah kita pergunakan dan juga merupakan peta dari program yang kita buat.

### **2.9.2 *Menu Borland Delphi***

#### **1. *Menu File***

Berisi fasilitas untuk membuat *Project* baru, menyimpan *Project*, membuka *Project*, dan keluar dari IDE Delphi.

#### **2. *Menu Edit***

Berisi fasilitas untuk melakukan *editing* atau perubahan pada kode program, juga pengaturan *form* dan unit ( ukuran, penempatan, kontrol, dsb ).

#### **3. *Menu Search***

Berisi Fasilitas untuk melakukan pencarian atau penggantian kata dalam tubuh kode program ( *unit* ) dan juga mencari letak kesalahan program.

#### **4. *Menu View***

Berisi fasilitas untuk mengatur tampilan IDE Delphi. Misalnya *Object Inspector*, daftar komponen, pengaturan *Toolbar*, *Form*, dan *Unit*.

#### **5. *Menu Project***

Berisi fasilitas yang berkaitan dengan properti dari *Project*, misalnya menambahkan atau memisahkan *Form* dan *Unit* dari sebuah *Project*.

#### **6. *Menu Run***

Berisi fasilitas untuk Kompiler Delphi, yang terpenting adalah *Run* dan *Reset*

**7. *Menu Component***

Berisi fasilitas untuk mengatur properti *Component Pallet* dan instalasi komponen baru.

**8. *Menu Database***

Berisi fasilitas yang berkaitan dengan pembuatan aplikasi data.

**9. *Menu Tools***

Berisi fasilitas untuk melakukan pengaturan *direktori, library, path* penyimpanan *file-file* penting dalam Delphi, dan *tools* yang bekerjasama dengan Delphi.

**10. *Menu Window***

Berisi fasilitas untuk berpindah dari satu jendela kerja ke jendela kerja yang lain dalam IDE Delphi.

**11. *Menu Help***

Berisi fasilitas menerima bantuan atau keterangan tentang Delphi atau keterangan tentang Delphi.

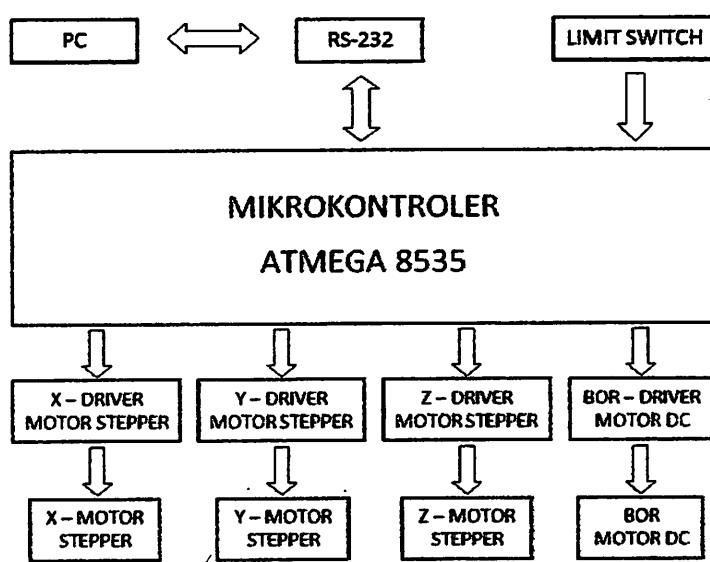
## BAB III

### PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini akan dibahas secara *detail* mengenai perancangan dan pembuatan alat secara keseluruhan. Termasuk setiap blok diagram yang menyusun sistem, meliputi perancangan sistem, prinsip kerja, *hardware* dan *software*.

#### 3.1 Perancangan Sistem

Secara garis besar sistem yang dirancang dan dibuat ditunjukkan pada gambar 3.1 dibawah ini:



Gambar 3.1 Diagram Sistem Secara Keseluruhan.

Fungsi masing – masing diagram blok :

#### 1. PC

Berfungsi sebagai informasi data masukan *drill file PCB* dan mengirim informasi status ke mikrokontroler.

#### 2. RS-232

Berfungsi sebagai *interface* antara PC dengan mikrokontroler.

### **3. Limit Switch**

Berfungsi untuk membatasi pergeseran ulir supaya tidak keluar dari batas koordinat yang telah ditentukan.

### **4. Mikrokontroler**

Berfungsi untuk pemroses data masukan dari PC dan untuk mengendalikan *Driver motor stepper* dan *driver relay*.

### **5. Driver Motor**

Berfungsi untuk memberikan activeasi *bit* ke *motor stepper*.

## **3.2 Prinsip Kerja**

Proses pengeboran pada mesin ini dilakukan secara otomatis dengan memasukkan data koordinat PCB pada PC kemudian pengeboran akan dilakukan secara otomatis oleh mesin.

Konfigurasi mesin bor PCB otomatis ini dapat dilihat pada gambar diatas. Secara garis besar mesin ini terdiri dari Mikrokontroler ATMega8535 sebagai prosessor utama. Tiga buah *driver motor stepper* berbasis Transistor IRFZ44 serta sebuah driver motor DC. Kontrol utama pengaturan motor DC menggunakan *Optocoupler* yang aktivasinya dilakukan oleh Mikrokontroler ATMega8535 secara otomatis.

Pada blok diatas *drill file* akan diinputkan ke *software* utama pada PC untuk diproses. Hasil proses PC yang berupa parameter – parameter tertentu akan dikirim secara *serial* ke mikrokontroler. Selanjutnya mikrokontroler akan memberikan *aktiviasi bit* ke *driver motor stepper* untuk menggerakkan motor

sesuai dengan parameter yang dikehendaki. Disamping itu mikrokontroler akan mengaktifkan motor selama proses pengeboran berlangsung.

Sistem tersebut dilengkapi dengan *limit switch* yang berfungsi untuk membatasi pergeseran ulir supaya tidak keluar dari batas koordinat yang telah ditentukan. *Limit switch* ini merupakan sebuah intrupsi global yang apabila diaktifkan akan menghentikan sistem sesaat dan akan mengembalikan pada posisi yang aman. Mekanisme kerja dari *limit switch* ini yaitu pada mesin berkerja dan pergeseran ulir mengenai *limit switch* maka *limit switch* akan memberikan sinyal positif ke mikrokontroler akan menghentikan proses sesaat dan memutar balik atau dengan kata lain aktifnya *limit switch* akan menyebabkan pin instrupsi pada mikrokontroler akan berkerja dan mengnonaktifkan driver motor sesaat kemudian mikrokontroler akan memberikan pulsa dan bit kepada driver untuk memutar balik motor secara otomatis. Berikut ini akan dijelaskan perancangan *hardware* dan *software* pada mikrokontroler maupun pada PC.

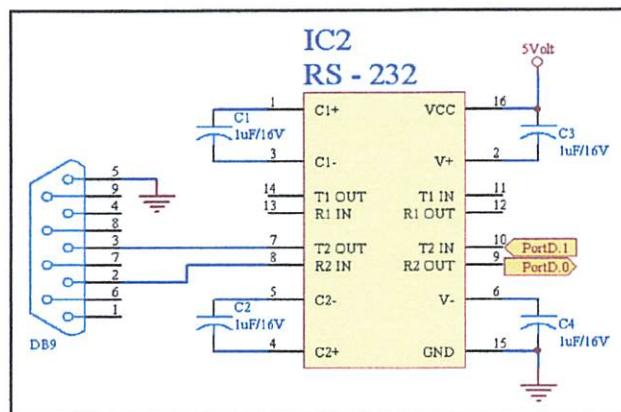
### **3.3 Perancangan *Hardware***

Perancangan *hardware* terbagi menjadi beberapa bagian yaitu Mikrokontroler, RS-232 Serial *Interface*, *Driver Motor Stepper*, *Driver Motor DC* dan Mekanik Mesin. Secara lengkap dijelaskan pada sub bab berikut :

#### **3.3.1 RS-232 Serial *Interface***

*Serial interface* pada mikrokontroler hanya menggunakan level tegangan 5 volt untuk merepresentasikan logika ‘1’ dan level tegangan 0 volt. Sedangkan untuk komunikasi yang lebih jauh dengan CPU maka diperlukan konversi level tegangan. Maka digunakan standart RS-232

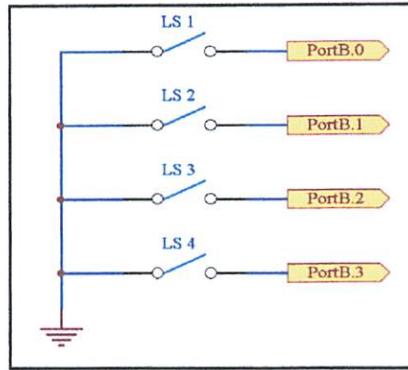
serial *interface* yang merupakan protokol komunikasi serial yang sangat umum digunakan. RS-232 menggunakan level tegangan -12 volt untuk merepresentasikan logika ‘1’ dan menggunakan +12 volt untuk merepresentasikan logika ‘0’. Pada alat ini digunakan konverter RS-232 buatan MAXIM yaitu MAX232.



Gambar 3.2 Rangkaian RS-232 *Serial Interface*.

### 3.3.2 Limit Switch

Rangkaian *limit switch* dalam perencanaan ini menggunakan *limit switch normally close*. Apabila pergerakan motor mencapai batas minimum atau maximum dari panjang ulir yang telah ditentukan dan menekan *limit switch* pada bagian tertentu dari mekanik maka *limit switch* akan memberikan aktivasi *bit* ke mikrokontroler. Jadi *limit switch* digunakan sebagai inputan untuk mikrokontroler seperti pada gambar 3.3 di bawah ini :

Gambar 3.3 Rangkaian *Limit Switch*.

### 3.3.3 Mikrokontroler

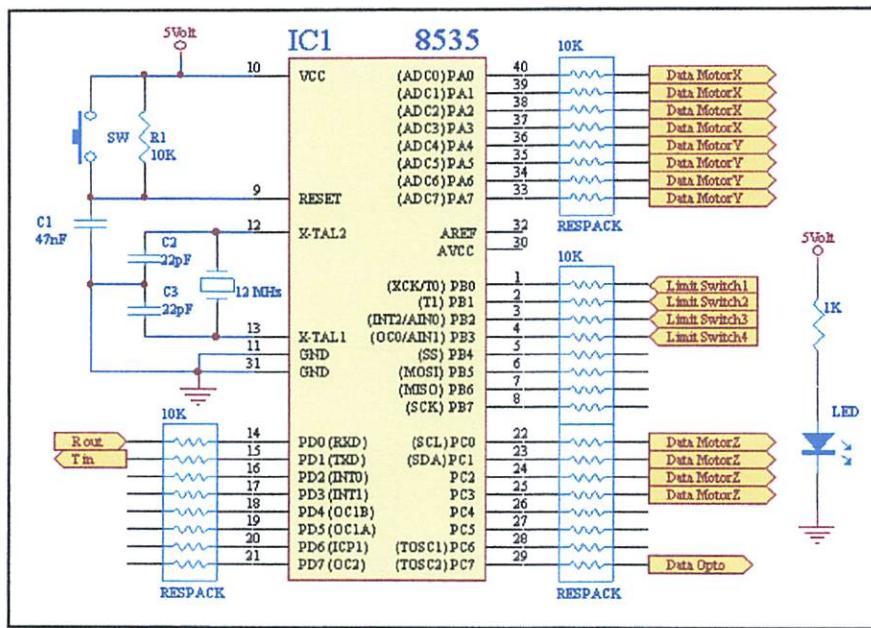
Mikrokontroler yang digunakan adalah jenis ATMega8535 yang merupakan keluarga mikrokontroler AVR buatan Atmel. Mikrokontroler jenis ini merupakan mikrokontroler yang memiliki arsitektur RISC 8 bit, dimana semua interupsi dikemas dalam kode 16 bit dan sebagian besar dieksekusi dalam 1 *siklus clock*. Proses eksekusi ini *relative* lebih cepat dibandingkan dengan mikrokontroler jenis MCS-51 yang memerlukan waktu eksekusi sebesar 12 *siklus clock*. ATMega memiliki kecepatan proses sebesar hingga 16 MHz dan penerimaan data *serial* hingga 2.5 Mbp sehingga sangat cocok digunakan pada sistem dengan konfigurasi pengiriman data *serial* yang kompleks dan menuntut proses eksekusi yang cepat. Frekuensi *clock* menggunakan *crystal* 11.0592 MHz. sehingga mendukung untuk komunikasi *serial* dengan *baudrate* 115200 bps. Resistor dan kapasitor yang dipasang pada reset bernilai 10K  $\Omega$  dan 22  $\mu$ F sehingga lama waktu *reset* adalah :

$$T = RC$$

$$T = 10^4 \cdot 10 \cdot 10^{-6} \text{ s}$$

$$T = 0.1 \text{ s}$$

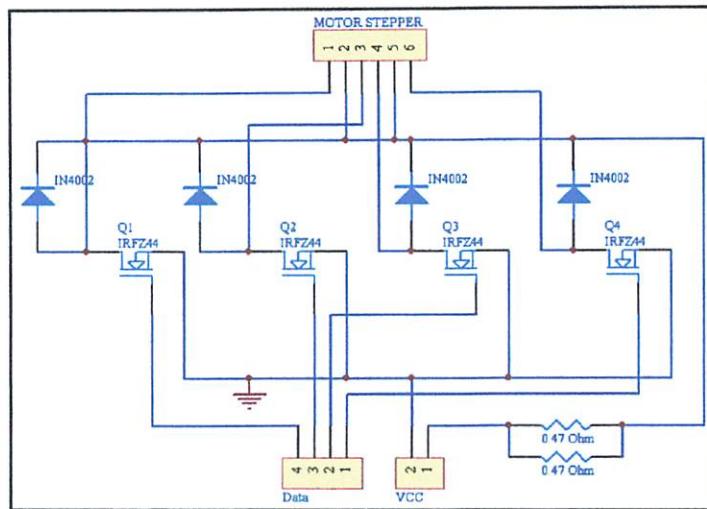
Dalam waktu 0.1 detik tersebut diharapkan cukup untuk melakukan proses *reset* yaitu mengosongkan semua *register* dan *internal RAM*. Berikut merupakan rangkaian skematik *hardware* mikrokontroler yang ditunjukan oleh gambar 3.4 dibawah ini :



Gambar 3.4 Rangkaian Minimum System ATMega8535 dan Alokasi Pin.

### 3.3.4 Driver Motor Stepper

*Driver motor stepper* pada rangkaian ini menggunakan transistor IRFZ44N yang merupakan rangkaian *switching*. Sebelum masuk ke motor kita beri transistor sebagai saklar otomatis. Pada inputan motor kita beri kaki *drain* transistor dan dari output mikrokontroler kita beri kaki *gate* transistor yang telah diprogram untuk menyala secara bergantian dan berurutan.



Gambar 3.5 Rangkaian Driver Motor Stepper

Prinsip kerja dari transistor IRFZ44 adalah bila kaki pada *gate* diberi tegangan bias maka transistor akan ON. Rangkaian *switching* ini, kaki *drain* transistor mendapat *supply* tegangan dari *motor stepper* yang mana *center tapped* pada kaki motor mendapat tegangan positif dari *power supply*. Begitu ada arus dari keluaran mikrokontroler maka transistor akan saturasi dan *motor stepper* akan bergerak sesuai dengan program. *Motor stepper* ini memiliki besar hambatan sebesar 1.8 Ohm dan besar tegangan maximal 3.3 volt. Maka arus yang mengalir pada *motor stepper* adalah :

$$\frac{1}{R_p} = \frac{1}{R_1} + \frac{1}{R_2} \quad \dots \dots \dots \quad (3.1)$$

$$\frac{1}{R_p} = \frac{1}{0.47} + \frac{1}{0.47}$$

$$R_p = \frac{0.47}{2} \Omega$$

$$R_p = 0.235 \Omega$$

$$I_{Motor} = \frac{V_{CC}}{R_{Motor} + R_p} \quad \dots \dots \dots \quad (3.2)$$

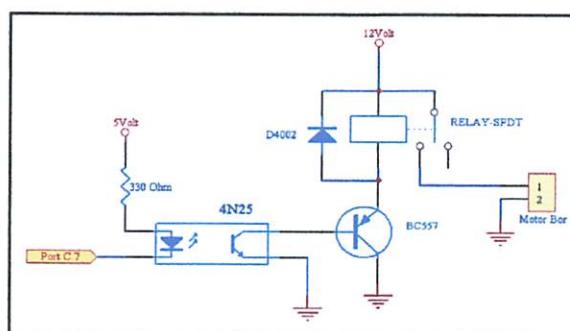
$$I_{Motor} = \frac{3}{1.8 + 0.235}$$

$$I_{Motor} = \frac{3}{2.035} \text{ A}$$

$$I_{Motor} = 1.47 \text{ A}$$

### 3.3.5 Driver Motor DC

Motor DC digunakan untuk memutar mata bor sehingga dapat digunakan untuk melakukan pengeboran. Untuk menggerakkan motor diperlukan driver motor DC yang terhubung terhadap salah satu pin pada mikrokontroler. Pemasangan *optocoupler* berfungsi sebagai penahan *bouncing* yang muncul akibat kelebihan arus yang ditimbulkan oleh beban, sehingga dapat menyebabkan kerusakan fisik maupun nonfisik (*program*) pada mikrokontroler. Mikrokontroler berfungsi untuk memberikan sinyal aktivasi pada *driver* motor DC untuk menggerakkan maupun menghentikan pergerakan motor DC. Seperti hanya motor DC komponen umumnya *driver* yang digunakan tersusun atas beberapa komponen yaitu *OptoCoupler 4N25*, Transistor BC557, Resistor, Diode dan Relay yang berfungsi sebagai *switching*. Berikut ini adalah rangkaian *driver* motor DC yang digunakan:



Gambar 3.6. Rangkaian *Driver Motor*

Cara kerja dari rangkaian diatas adalah untuk mengaktifkan *relay*, mikrokontroler akan mengeluarkan logika “0” pada PortB.7, dengan begitu akan ada arus yang mengaliri LED yang menyebabkan LED mengeluarkan cahaya dengan intensitas tertentu. Cahaya itu akan membuat transistor NPN pada *optocoupler* menjadi saturasi. Dengan rangkaian seperti diatas, kolektor dari transistor pada *optocoupler* terhubung pada basis transistor BC557 dan emitornya pada *ground*, maka ketika transistor pada *optocoupler* saturasi akan membuat BC557 juga mengalami saturasi.

$$I_E = \frac{V_{CC} - V_{CE}}{R_{Relay}} \dots \dots \dots \quad (3.3)$$

$$= \frac{12V - 0.3V}{345\Omega}$$

$$= 30.7 \text{ mA}$$

$$I_C = I_E - I_B \quad \dots \quad (3.4)$$

$$I_C = I_B \cdot h_{fe} \dots \quad (3.5)$$

Dari persamaan 3.4 dan 3.5 didapatkan :

$$I_B = \frac{I_E}{hfe + 1} \dots \quad (3.6)$$

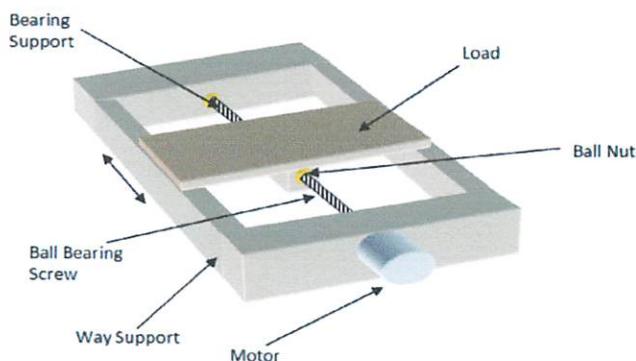
$$= \frac{0.0307}{240 + 1}$$

$$= 0.127 \text{ mA}$$

Dari *datasheet* dapat diketahui arus kolektor maksimum yang dapat mengalir pada transistor pada *optocoupler* 4N25 adalah 150 mA,

### 3.3.6 Mekanik Mesin

Desain mekanik menggunakan ulir yang pada ujungnya diberi *motor stepper*. Sehingga ulir akan berputar sesuai dengan banyaknya putaran pada *motor stepper*. Pada ulir ini juga dipasang beban yang dihubungkan pada baut, sehingga putaran ulir dapat bergeser ke kanan maupun ke kiri sesuai arah putaran *motor stepper*. Gambaran dasar dari mekanik mesin dapat dilihat pada gambar 3.7 dibawah ini :



Gambar 3.7 Struktur Mekanik

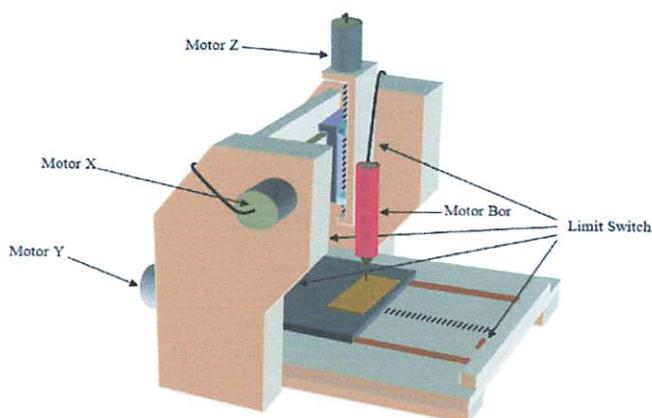
Pada gambar diatas beban akan bergeser ke kiri maupun ke kanan sesuai arah putaran motor. Begitu pula kecepatan pergeseran beban juga bergantung pada kecepatan putaran motor. Semakin cepat putaran motor maka pergeseran beban akan semakin cepat. Oleh karena itu untuk mendukung proses diatas perlu digunakan motor dengan torsi yang cukup tinggi. Pada aplikasi pengeboran PCB ini atau secara lebih sederhana dapat dikatakan sebagai aplikasi pendekripsi koordinat, model mekanik diatas terasa sangat tepat karena memiliki ketelitian yang *relative* tinggi meski dengan kecepatan yang *relative* lambat.

Setiap ulir dengan berdiameter tertentu memiliki *standart* panjang pergeseran berbeda beda karena persatu putaran baut. Misalnya ulir berdiameter 8 mm memiliki pergeseran 1.27 mm persatu putaran baut, Jadi jika *motor stepper* berputar satu putaran penuh maka alat pergeseran 1.27 mm. sedangkan ulir berdiameter 20 mm memiliki pergeseran 1.5 mm per satu putaran baut. Oleh karena itu untuk membuat ulir bergeser dengan panjang tertentu, cukup dengan memberikan *step* dengan jumlah tertentu pada *motor stepper* tanpa adanya *feed back*. Berikut ini spesifikasi bentuk mekanik dari mesin CNC ini :

Diameter Ulir 8 mm = pergeseran alat 1.27mm/satu putaran *motor stepper*

Panjang ulir sumbu – X = 40 cm      Panjang ulir sumbu – Z = 20 cm.

Panjang ulir sumbu – Y = 50 cm



Gambar 3.8 Desain Mekanik

Pada tiap akhir sumbu mekanik diatas, dipasang sebuah *limit switch* sebagai proteksi supaya pergerakan motor tidak keluar dari *track* yang telah ditentukan. Apabila *limit switch* ini ditekan oleh *body* motor maka *limit switch* akan mengirim sinyal positif ke mikrokontroler untuk mengembalikan posisi motor supaya tidak keluar dari *track* yang telah ditentukan.

### 3.4 Perancangan Perangkat Lunak (*Software*)

Perangkat lunak adalah sebuah jembatan yang menghubungkan keseluruhan komponen yang ada pada sebuah komputer. Pada perancangan ini ada dua macam, yaitu perancangan perangkat lunak untuk mikrokontroler menggunakan *Basic Compiler* Mikrokontroler AVR ATMega8535 dan perancangan perangkat lunak untuk komputer menggunakan *Delphi7*.

#### 3.4.1 Program Aplikasi Mikrokontroler

Program mikrokontroler bertujuan untuk mengontrol masukan dan keluaran. Pembuatan clock pada mikrokontroler memakai Crystal 11.0592 Mhz dan kapasitor 30pF. Hal-hal yang dikontrol mikrokontroler adalah proses pergerak *motor stepper*. Motor stepper yang pakai dalam pembuatan alat ini memiliki sudut  $1.8^\circ$  jadi untuk memutar motor stepper satu putaran adalah sebagai berikut :

$$N = \frac{\text{sudut satu putaran}}{\text{sudut motor stepper}}$$

$$N = \frac{360^\circ}{1.8^\circ}$$

$$N = 200 \text{ step}$$

$N$  = banyak step yang dibutuhkan *motor stepper* untuk satu putaran.

Berikut adalah program untuk menjalankan satu putaran *motor stepper* pada mikrokontroler.

*For A = 1 To 50*

*Porta = &B00001010*

*Waitms 3*

*Porta = &B00001100*

*Waitms 3*

*Porta = &B00000101*

*Waitms 3*

*Porta = &B00000011*

*Waitms 3*

*Next*

$$A = \frac{360^\circ}{1.8^\circ \times \text{data\_step}}$$

$$A = \frac{360^\circ}{1.8^\circ \times 4}$$

$$A = \frac{360^\circ}{7.2^\circ}$$

$$A = 50$$

*A* = banyak pengulangan pada program untuk satu putaran *motor stepper*.

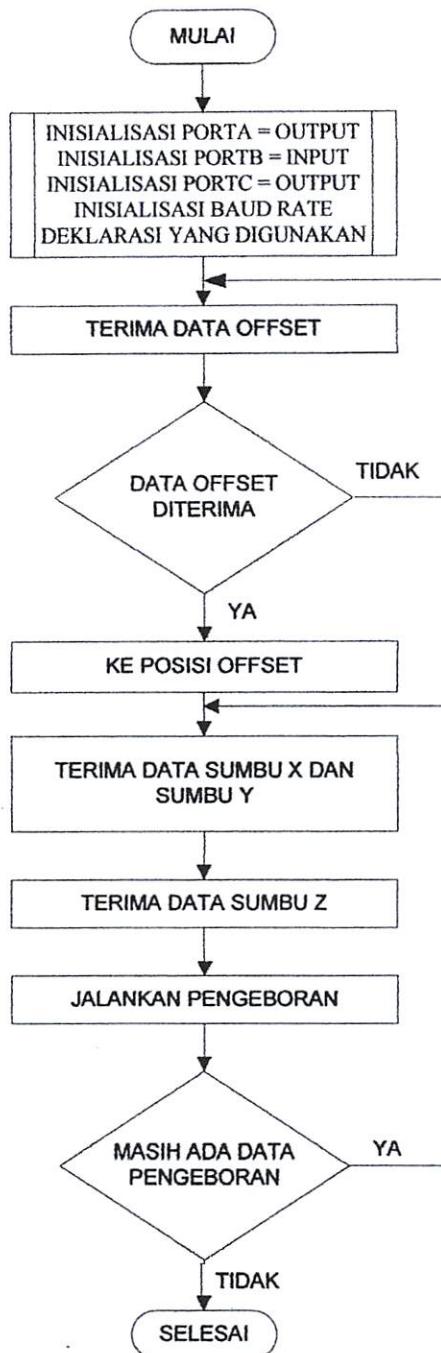
$$t = 3 \times \text{banyak\_step}$$

$$t = 3 \times 200$$

$$t = 600\text{ms}$$

*t* = waktu yang dibutuhkan untuk menjalankan satu putaran *motor stepper*.

Perancangan perangkat lunak (*software*) yang digunakan dalam perencanaan dan pembuatan alat akan dipaparkan dalam *flowchart* sistem secara keseluruhan. Pembuatan *software* hanya dilakukan pada mikrokontroler menggunakan bahasa pemrograman *Basic Compiler* dengan bantuan *compiler BASCOM AVR*. Diagram alir atau *Flowchart* program secara keseluruhan adalah sebagai berikut :

Gambar 3.9 *Flow Chart Software* Pada Mikrokontroler

Pada program diatas mikrokontroler akan menunggu data status dari PC. Setelah data tersebut diterima, mikrokontroler akan mengirim status ke PC yang menandakan status telah dikirim dan mikrokontroler siap menerima perintah berikutnya.

### 3.4.2 Program Aplikasi *Borland Delphi7*

Pada Program *Borland Delphi7* bertujuan untuk memproses data *drill file PCB* dan mengirim informasi status ke mikrokontroler. Langkah – langkah komunikasi *serial* antara PC dengan mikrokontroler dalam penulisan kode programnya ditunjukkan sebagai berikut :

1. Program ini akan mengaktifkan *port*, *serial* ketika tombol *start* ditekan. Untuk penulisan kode program klik tombol start, kemudian pilih even dan klik ganda *OnClick*, maka akan muncul *editor* program.

Tulis kode program sebagai berikut :

```
Procedure TForm1.Button1_Click (Sender:Tobject);  
Begin  
    ComPort.Open ;
```

```
End ;
```

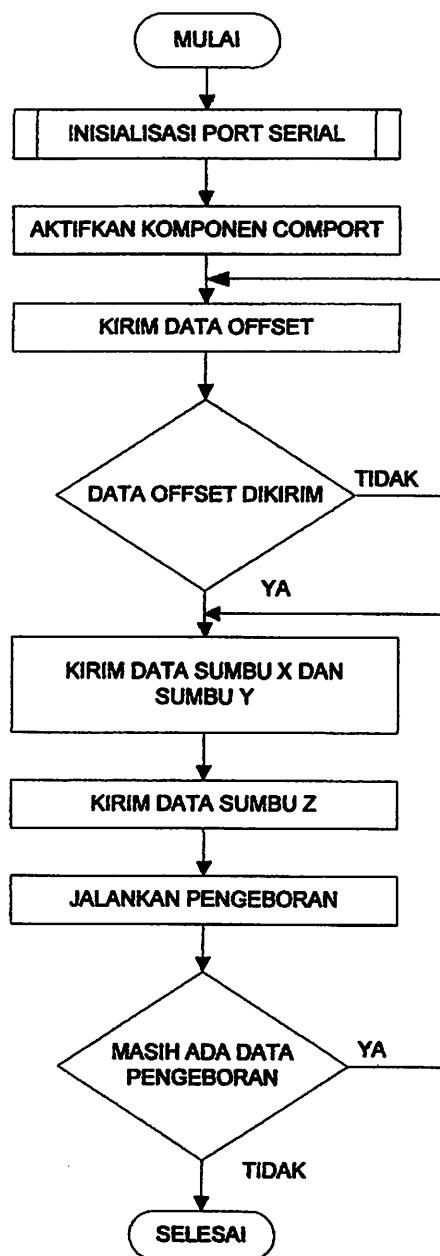
2. Dan progam ini akan mengeluarkan data digital melalui *port serial* dan penulisan kode programnya sebagai berikut :

```
Var
```

```
Str: string;  
Begin  
    Str :=Label.Caption ;  
    Str := str+ #13#10 ;  
    ComPort1.WriteString(Str) ;  
    Z:=StrToFloat(Label11.Caption)  
End;
```

Setelah menerima data, mikrokontroler akan menggerakkan mata bor menuju titik yang dikehendaki kemudian pada setiap satu kali pengeboran mikrokontroler akan mengirim status dan menunggu 5 byte data berikutnya dari PC. Proses ini berlangsung berulang-ulang hingga pengeboran selesai. Apabila pengeboran selesai, maka mikrokontroler akan mengirim status.

Diagram alir atau *Flowchart* program secara keseluruhan pada program *Borland Delphi7* adalah sebagai berikut :



Gambar 3.10 *Flow Chart Software Pada Borland Delphi7*

## BAB IV

# PENGUJIAN ALAT

Pengujian dilakukan untuk mengetahui sejauh mana perlatan dapat bekerja sesuai dengan perencanaan. Langkah pengujian dilakukan melalui 2 tahap, yakni pengujian pada setiap blok dan pengujian pada sistem keseluruhan. Tahap pertama dimaksudkan untuk mengetahui sejauh mana blok-blok rangkaian dapat berjalan, sedangkan tahap kedua dilakukan setelah diperoleh kepastian bahwa tiap blok rangkaian telah berjalan sesuai rencana.

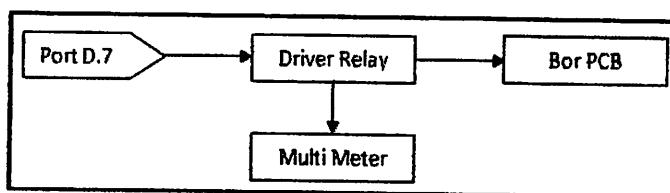
### 4.1 Pengujian Rangkaian *Driver Relay*

#### 4.1.1 Alat dan Bahan

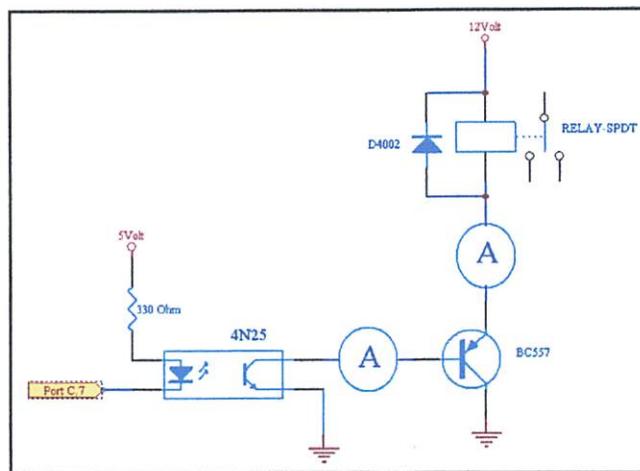
1. Rangkaian *Driver Relay*
2. Multimeter Digital SUNWA DT-930T
3. Rangkaian Mikrokontroler ATMega8535
4. Catu daya 12 Volt
5. Catu daya 5 Volt

#### 4.1.2 Langkah Pengujian

1. Menghubungkan alat dan bahan seperti gambar 4.1
2. Berikan masukan aktif *low* pada *input* rangkaian *driver* melalui port mikrokontroler.
3. Amati hasil yang ditunjukkan amperemeter



Gambar 4.1 Diagram Blok Pengukuran *Driver Relay*

Gambar 4.2 Pengukuran Rangkaian *Driver Relay*

#### 4.1.3 Hasil dan Analisa Pengujian *Driver Relay*



Gambar 4.3 Hasil Pengukuran Arus Ib Transistor BC557

Dari hasil perhitungan didapatkan arus Ib pada transistor BC557 sebesar 0.127 mA, Maka dari data tersebut dapat dicari persentase *error* hasil pengujian terhadap perhitungan adalah:

$$I_1 = \text{Hasil\_perhitungan}$$

$$I_2 = \text{Hasil\_pengujian}$$

$$\% \text{Error} = \frac{|I_1 - I_2|}{I_1} \times 100\%$$

$$\% \text{Error} = \frac{|0.127 - 0.13|}{0.0127} \times 100\% = 2\%$$

$$\% \text{Error} = \frac{|0.127 - 0.12|}{0.0127} \times 100\% = 5.5\%$$

Tabel 4.1 Perbandingan Hasil Perhitungan Dan Hasil Pengukuran Arus Ib  
Pada Transistor BC557

No	Arus Ib		Error (%)
	Hasil Perhitungan (mA)	Hasil Pengukuran (mA)	
1	0.127	0.13	2
2	0.127	0.12	5.5



Gambar 4.4 Hasil Pengukuran Arus Ie Trasistor BC557

Dari hasil perhitungan didapatkan arus Ib pada transistor BC557 sebesar 39 mA, Maka akan dicari persentase *error* hasil pengujian terhadap perhitungan adalah:

$$I_1 = \text{Hasil\_perhitungan}$$

$$I_2 = \text{Hasil\_pengujian}$$

$$\% \text{Error} = \frac{|I_1 - I_2|}{I_1} \times 100\%$$

$$\% \text{Error} = \frac{|30.7 - 30|}{30.7} \times 100\% = 2.33\%$$

Tabel 4.2 Perbandingan Hasil Perhitungan Dan Hasil Pengukuran Arus Ie  
Pada Transistor BC557

No	Arus Ie		Error (%)
	Hasil Perhitungan (mA)	Hasil Pengukuran (mA)	
1	30.7	30	2.33
2	30.7	29	5.8

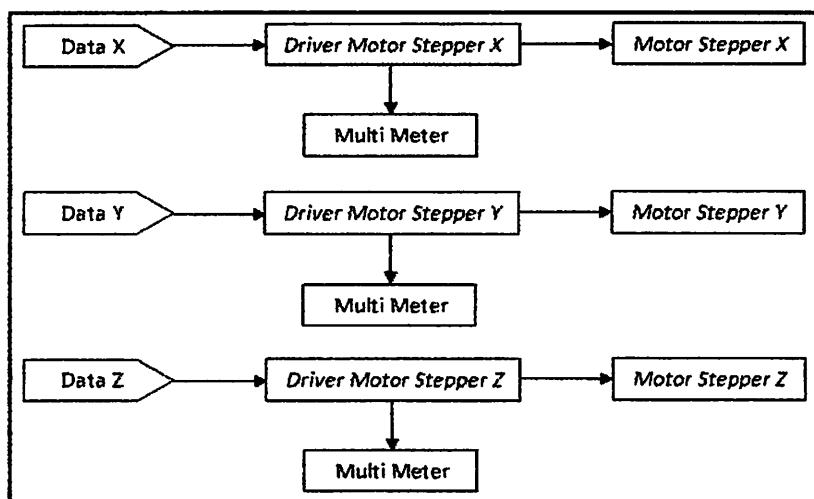
## 4.2 Pengujian Rangkaian Driver Motor Stepper

### 4.2.1 Alat dan Bahan

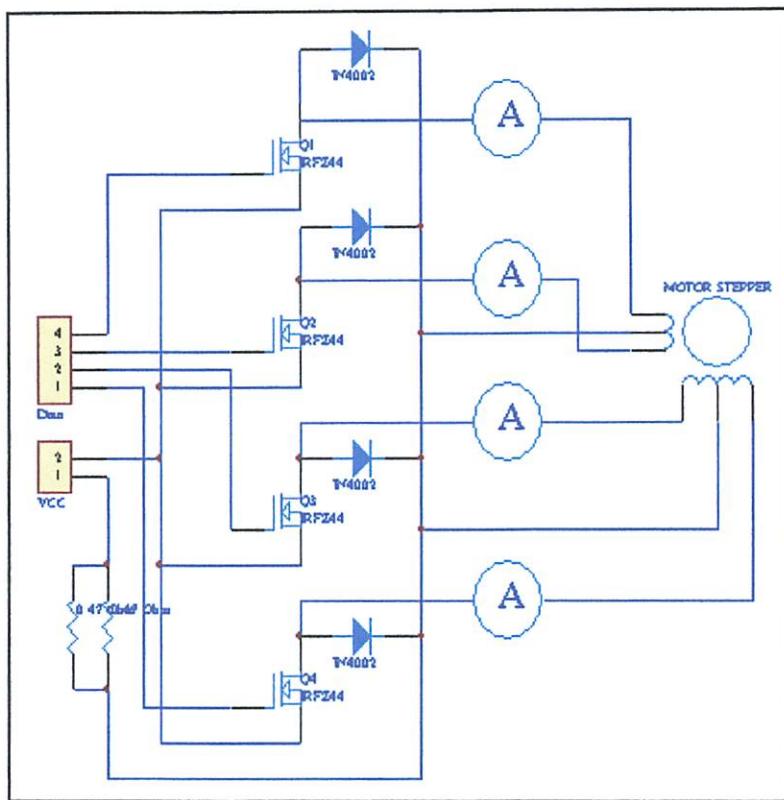
1. Rangkaian Driver Motor Stepper.
2. Multimeter Digital SUNWA DT-930T
3. Catu daya 3 Volt

### 4.2.2 Langkah Pengukuran

1. Menghubungkan alat dan bahan seperti gambar 4.4
2. Berikan masukan aktif *high* pada *input* rangkaian *driver* melalui port mikrokontroler.
3. Amati hasil yang ditunjukkan amperemeter.

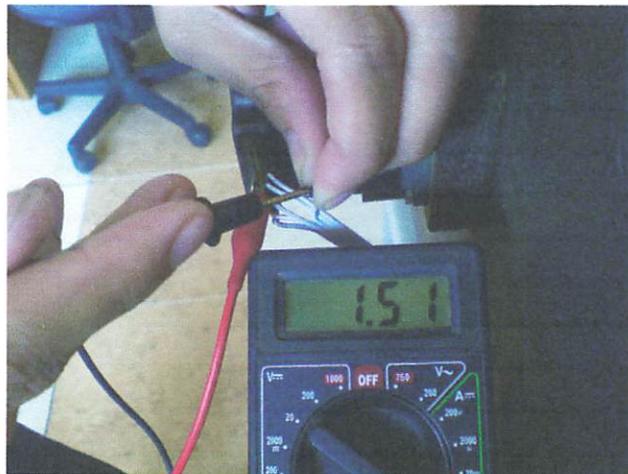


Gambar 4.5 Blok Diagram *Driver Motor Stepper*.



Gambar 4.6 Pengukuran Arus Ig Transistor IRFZ44N.

#### 4.2.3 Hasil dan Analisa Pengujian *Driver Motor Stepper X.*



Gambar 4.7 Pengukuran Rangkaian *Driver Motor Stepper X.*

Dari hasil perhitungan didapatkan arus  $I_D$  pada transistor IRFZ44N sebesar 1.47A, Maka dari data tersebut dapat dicari persentase *error* hasil pengujian terhadap perhitungan adalah:

$$I_1 = \text{Hasil\_perhitungan}$$

$$I_2 = \text{Hasil\_pengujian}$$

$$\%Error = \frac{|I_1 - I_2|}{I_1} \times 100\%$$

$$\%Error = \frac{|1.47 - 1.51|}{1.47} \times 100\% = 2.72\%$$

$$\%Error = \frac{|1.47 - 1.43|}{1.47} \times 100\% = 2.72\%$$

$$\%Error = \frac{|1.47 - 1.48|}{1.47} \times 100\% = 0.68\%$$

$$\%Error = \frac{|1.47 - 1.46|}{1.47} \times 100\% = 0.68\%$$

Tabel 4.3 Perbandingan Hasil Perhitungan Dan Hasil Pengukuran Arus Id

Pada Transistor IRFZ44N

No	Arus Id		Error (%)
	Hasil Perhitungan (mA)	Hasil Pengukuran (mA)	
1	1.47	1.51	2.72
2	1.47	1.43	2.72
3	1.47	1.48	0.68
4	1.47	1.37	0.68

#### 4.2.4 Hasil dan Analisa Pengujian *Driver Motor Stepper Y.*



Gambar 4.8 Pengukuran Rangkaian *Driver Motor Stepper Y.*

Dari hasil perhitungan didapatkan arus Id pada transistor IRFZ44N sebesar 1.47A, Maka dari data tersebut dapat dicari persentase *error* hasil pengujian terhadap perhitungan adalah:

$$I_1 = \text{Hasil\_perhitungan}$$

$$I_2 = \text{Hasil\_pengujian}$$

$$\% \text{Error} = \frac{|I_1 - I_2|}{I_1} \times 100\%$$

$$\% \text{Error} = \frac{|1.47 - 1.45|}{1.47} \times 100\% = 1.36\%$$

$$\% \text{Error} = \frac{|1.47 - 1.47|}{1.47} \times 100\% = 0\%$$

$$\% \text{Error} = \frac{|1.47 - 1.52|}{1.47} \times 100\% = 3.40\%$$

$$\% \text{Error} = \frac{|1.47 - 1.46|}{1.47} \times 100\% = 0.68\%$$

Tabel 4.4 Perbandingan Hasil Perhitungan Dan Hasil Pengukuran Arus Id

Pada Transistor IRFZ44N

No	Arus Id		Error (%)
	Hasil Perhitungan (mA)	Hasil Pengukuran (mA)	
1	1.47	1.45	1.36
2	1.47	1.47	0
3	1.47	1.52	3.40
4	1.47	1.46	0.68

#### 4.2.5 Hasil dan Analisa Pengujian Driver Motor Stepper Z.



Gambar 4.9 Pengukuran Rangkaian Driver Motor Stepper Z.

Dari hasil perhitungan didapatkan arus  $I_D$  pada transistor IRFZ44N sebesar 1.47A, Maka dari data tersebut dapat dicari persentase *error* hasil pengujian terhadap perhitungan adalah:

$$I_1 = \text{Hasil\_perhitungan}$$

$$I_2 = \text{Hasil\_pengujian}$$

$$\%Error = \frac{|I_1 - I_2|}{I_1} \times 100\%$$

$$\%Error = \frac{|1.47 - 1.49|}{1.47} \times 100\% = 1.36\%$$

$$\%Error = \frac{|1.47 - 1.45|}{1.47} \times 100\% = 1.36\%$$

$$\%Error = \frac{|1.47 - 1.53|}{1.47} \times 100\% = 4.08\%$$

$$\%Error = \frac{|1.47 - 1.52|}{1.47} \times 100\% = 3.40\%$$

Tabel 4.5 Perbandingan Hasil Perhitungan Dan Hasil Pengukuran Arus Id  
Pada Transistor IRFZ44N

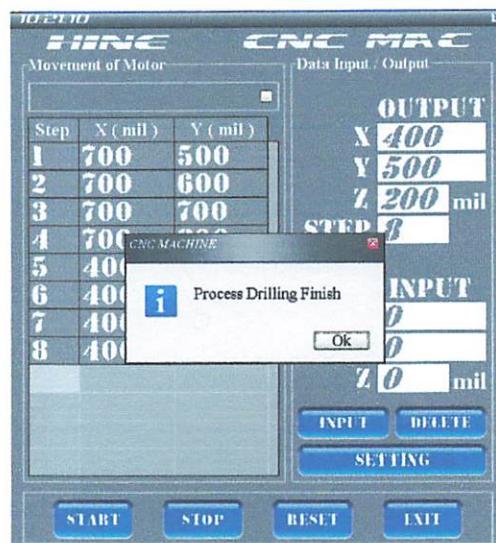
No	Arus Id		Error (%)
	Hasil Perhitungan (mA)	Hasil Pengukuran (mA)	
1	1.47	1.49	1.36
2	1.47	1.45	1.36
3	1.47	1.53	4.08
4	1.47	1.52	3.40

Pengujian alat ini dilakukan untuk mengetahui kerja dari sistem. Pada sistem ini pengambilan data dilakukan per sumbu terutama sumbu X dan sumbu Y yang meliputi nilai input dari koordinat dan implementasi nilai koordinat yang dicapai oleh alat. Begitu pula pengambilan data waktu pengeboran yang dilakukan oleh keseluruhan sistem. Selanjutnya akan dikalkulasi nilai *error* yang ditimbulkan oleh sistem untuk kemudian diberikan saran dan masukan untuk pengembangan alat ke depan.

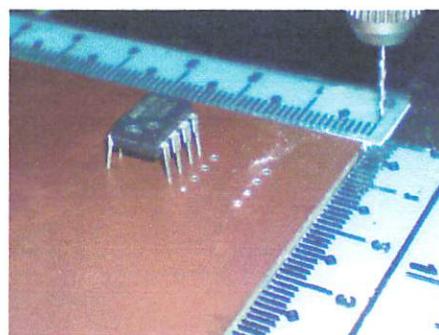
Pengeboran dilakukan secara otomatis dengan memasukan *drill file* dari *PCBWizard 3.5 Professional Edition* sebagai input koordinat. Data dari *file* ini memiliki satuan *mil* yang harus dikonversi terlebih dahulu kedalam bentuk mm maupun cm kemudian dikonversi lagi dalam bentuk nilai step yang dibutuhkan oleh motor.

#### 4.3 Hasil Pengujian Keseluruhan

Berikut adalah proses dan hasil pengujian alat juga diberikan data berdasarkan input dan output yang dieksekusi oleh mesin. Pada pengujian ini ada 8 *step* dengan waktu penyelesaian selama 1.24 menit :



Gambar 4.10 Hasil Pengujian Pada Program Delphi7.



Gambar 4.11 Hasil Pengujian Keseluruhan.

Tabel 4.6 Hasil Pengujian Keseluruhan.

Step	Input				Output		Error	
	X (mil)	Y (mil)	X (mm)	Y (mm)	X (mm)	Y (mm)	X (mm)	Y (mm)
1	700	500	17.78	12.50	17.74	12.48	0.04	0.02
2	700	600	17.78	15.24	17.74	15.22	0.04	0.02
3	700	700	17.78	17.78	17.76	17.76	0.02	0.02
4	700	800	17.78	20.32	17.76	20.28	0.02	0.04
5	400	800	10.16	20.32	10.11	20.31	0.05	0.01
6	400	700	10.16	17.78	10.13	17.74	0.03	0.04
7	400	600	10.16	15.24	10.15	15.20	0.01	0.04
8	400	500	10.16	12.50	10.13	12.48	0.03	0.02
Error Rata – Rata							0.030	0.0262

$$\% \text{ Error X} = 0.03\text{mm} \times 100\% = 3\%$$

$$\% \text{ Error Y} = 0.0262\text{mm} \times 100\% = 2.62\%$$

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Dari hasil pengujian dan pengukuran seluruh sistem, maka dapat diambil beberapa kesimpulan dan saran untuk keperluan pengembangan selanjutnya adalah berikut :

1. Sistem pengeboran yang telah didesain dapat berjalan dengan baik dan memiliki akurasi pengeboran yang cukup tinggi dengan *error* rata – rata sebesar 0.03 mm untuk sumbu X dan 0.0262Y.
2. *Error* disebabkan oleh *bearing* pada bagian mekanik bor yang berukuran 4,5mm sedangkan *As* yang dipakai pada mekanik bor tersebut berukuran 4mm jadi pada saat proses pengeboran berlangsung mekanik bergetar.
3. Dari hasil pengujian keseluruhan yang telah dijalankan dapat diperoleh besar *error* pada sumbu X sebesar 3% dan sumbu Y sebesar 2,62%.
4. Motor stepper jenis Unipolar memiliki torsi yang cukup tinggi untuk menggerakkan mekanik.

#### **5.2 Saran**

Saran penulis untuk pengembangan alat menuju hasil yang lebih baik adalah sebagai berikut :

1. Mekanik sebaiknya didesain menggunakan struktur logam yang ringan supaya pergerakan motor berjalan lebih cepat.
2. Perlu ditambahkan As lebih besar pada mekanik untuk menyangga pergerakan mekanik.

## **DAFTAR PUSTAKA**

- [1] Yildirim, Alper. "CNC Printed Circuit Board Drilling Machine". Spiring 2003.haceteppe University
- [2] Hordeski, Michael. "CAD/CAM Techiques". 1996 Prentice-Hall
- [3] Wardhana, Lingga. "Belajar Sendiri Microkontroler ATMega 8535", PT. Elek Media Komputindo, Jakarta
- [4] Barnett, Cox, & O'Cull. "Embedded C Programming and the Atmel AVR". 2003, Thomson-Dermal Learning
- [5] Tomonari Furukawa, "Robot Design", School of Mechanical and Manufacturing Engineering Univercity of New South Wales.
- [6] Malvino, "Prinsip-Prinsip Elektronika Edisi Kedua", 1979.
- [7] Tiala, Fransiskus A.K., "Perancangan Dan Pembuatan Alat Bor PCB Berbasis Mikrokontroler AT89S51", Tugas Akhir, Program D-III Tehnik ITN, Malang 2006
- [8] Iswanto."Antarmuka Port Pararel Dan Port Serial",2008
- [9] Interfacing the motor stepper  
<http://www.ams2000.com:80/pdf/step101.pdf>
- [10] Interfacing the serial / rs232 port,  
<http://wwwbeyondlogic.org/serial/serial.pdf>
- [11] [www.st\\_electronics.com](http://www.st_electronics.com)
- [12] <http://homepages.which.net/~pul.hills/Motor/Motor.html>
- [13] <http://homepages.which.net/~pul.hills/SpeedControl/SpeedControl-lers.html>

# **LAMPIRAN**



## FORMULIR BIMBINGAN SKRIPSI

Nama : SIGIT ADI PRAYITNO  
Nim : 04.12.269  
Masa Bimbingan : 19 JULI 2009 s/d 19 JANUARI 2010  
Judul Skripsi : PERANCANGAN DAN PEMBUATAN ALAT PENGEBORAN PCB BERBASIS COMPUTER NUMERIC CONTROL (CNC)

NO	Tanggal	Uraian	Paraf Pembimbing
1	13/8 09	Revisi BAB I, II, III	
2	20/8 09	Acc BAB I	
3		Revisi BAB III Acc. BAB II	
4	2/9 09	Acc maklum suatu	
5	2/10 09	Acc BAB IV	
6		" " V	
7		Mjlan kompre	
8			
9			
10			

Malang, 2 - 10 - 2009  
Dosen Pembimbing I

Irmalia S. Faradisa, ST, MT  
NIP. P. 1030100365



## FORMULIR BIMBINGAN SKRIPSI

Nama : SIGIT ADI PRAYITNO  
Nim : 04.12.269  
Masa Bimbingan : 19 JULI 2009 s/d 19 JANUARI 2010  
Judul Skripsi : PERANCANGAN DAN PEMBUATAN ALAT PENGEBORAN PCB BERBASIS COMPUTER NUMERIC CONTROL (CNC)

NO	Tanggal	Uraian	Paraf Pembimbing
1	12-08-2009	Konsultasi <u>Pus I</u> & <u>Pus II</u> Baboth Revisi 8+	JG
2	29-08-2009	How Chas di Mat: (proses ujoran)	JG
3	31/08	Ditambahkan perujinan alat Cetakan dan Gerobak.	JG
4	07-09	malesah Cemaraan berasil	JG
5	09-09	Rec Layanan Skripsi	JG
6		Rec Kompre	JG
7			
8			
9			
10			

Malang,  
Dosen Pembimbing II

I Komang Sumawirata, ST, MT  
NIP.P. 1030100361

INSTITUT TEKNOLOGI NASIONAL MALANG  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO

### Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Sigit Adi  
NIM : 0412269  
Perbaikan meliputi

- .) perhitungan delay untuk PC & Micro  
→ dapatkan  $360^\circ$  dist.
- .) perhit. Pengesaran 1,27 mm dan hub.  
( $\omega$  atau).
- .) ~~Umpama~~ → ~~Supwana~~

Malang, 06 - 10 - 2009

( 2m )  
V



INSTITUT TEKNOLOGI NASIONAL  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK ELEKTRONIKA

## FORMULIR PERBAIKAN SKRIPSI

Nam : Sigit Adi Prayitno  
NIM : 04.22.269  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika  
Judul : PERANCANGAN DAN PEMBUATAN ALAT  
PENGEBORAN PCB BERBASIS COMPUTER  
*NUMERIC CONTROL ( CNC )*

Hari / Tanggal Ujian Skripsi : Selasa / 06 Oktober 2009

No	Tanggal	Uraian	Paraf
1	06/10/2009	- Perhitungan untuk delay mikro didapatkan 360° - Pergerakan 1.27 mm dan hubungan ke alat - Lampiran	

Disetujui,

Penguji I

( Joseph Dedy Irawan, ST, MT )  
NIP. 132315178

Mengetahui,

Penguji II

( Sotirohadi, ST, MSc )  
NIP.Y.1039700309

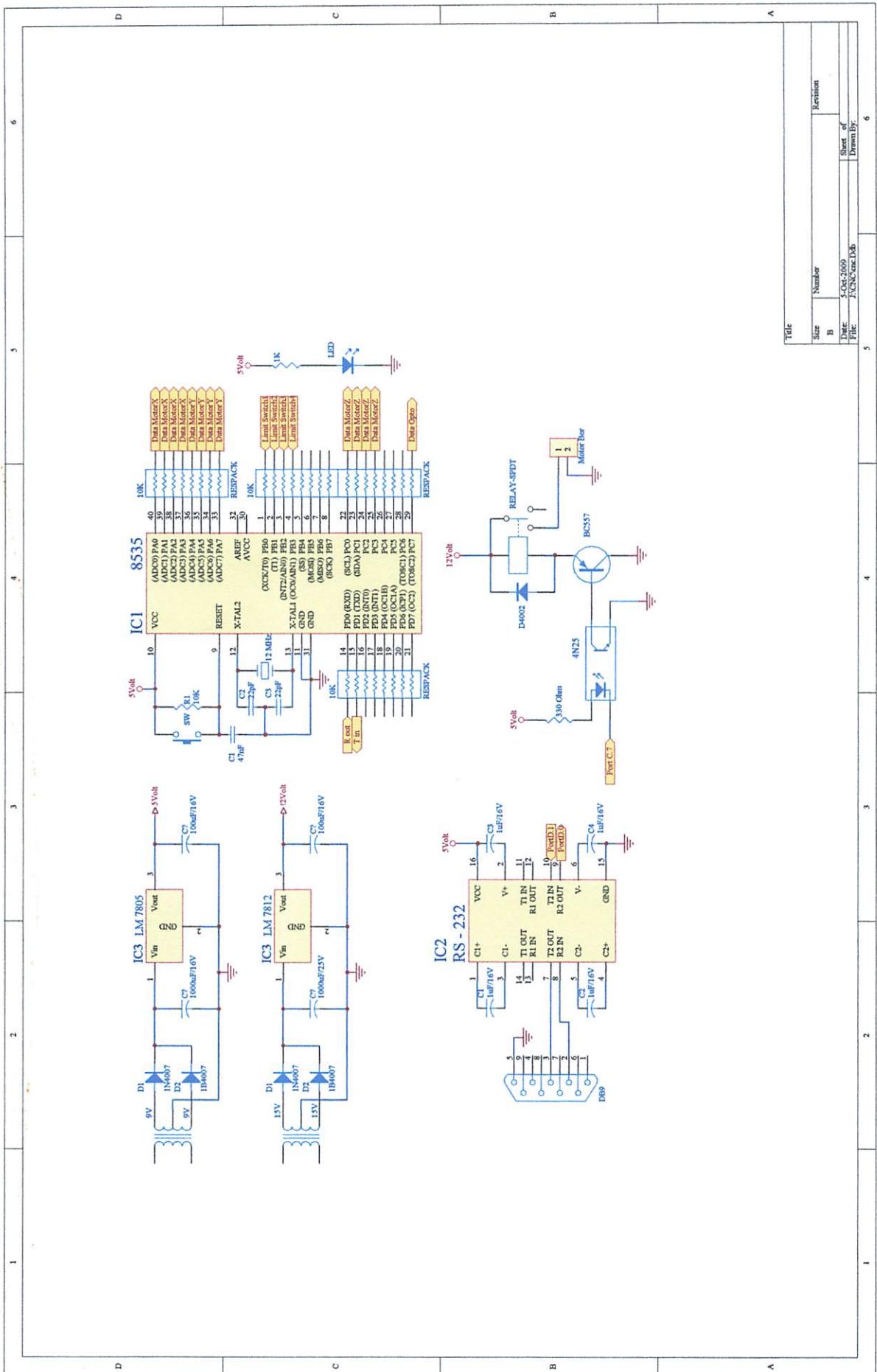
Mengetahui,

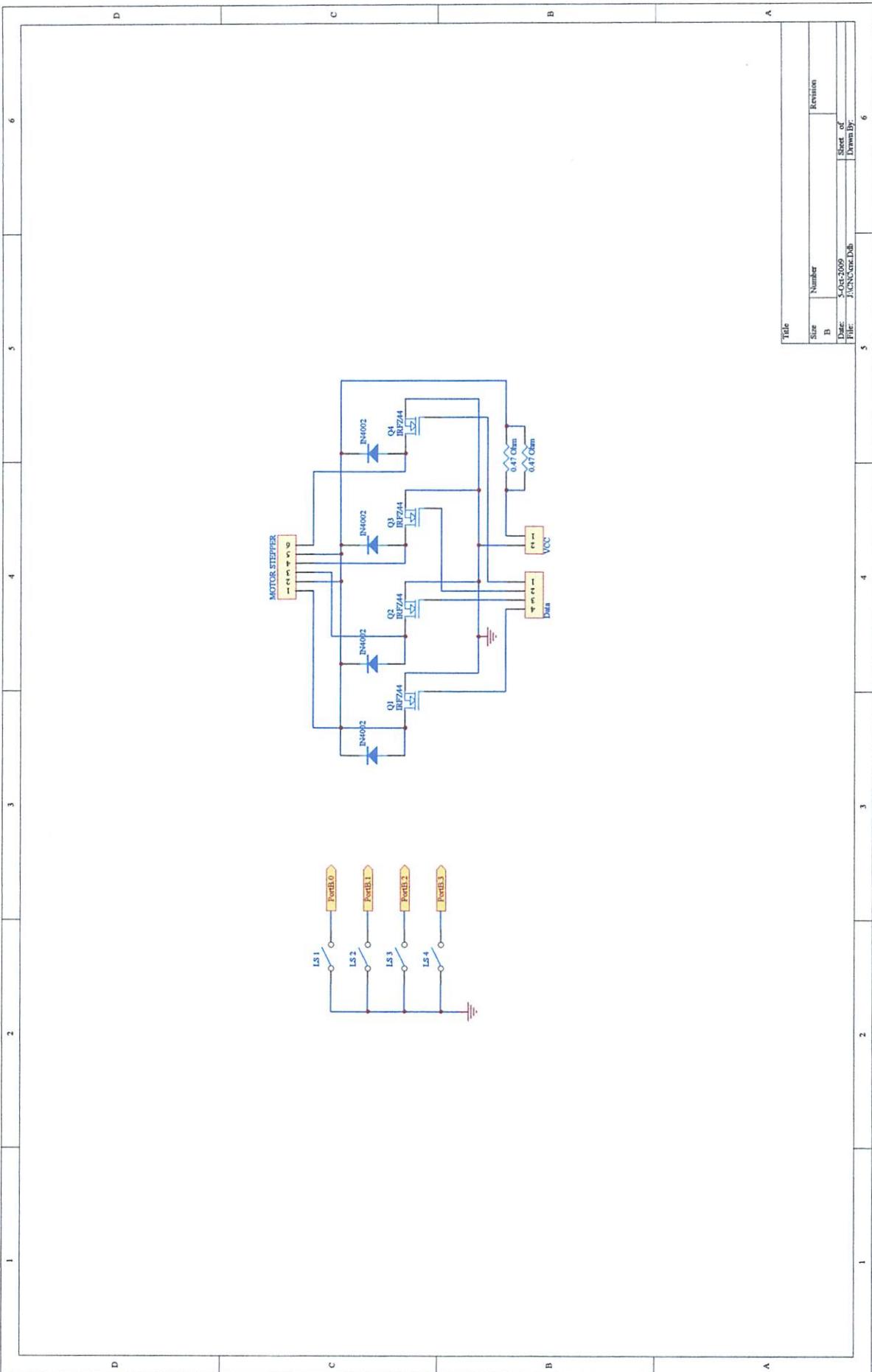
Dosen Pembimbing I

( Irmalia S. Faradisa, ST. MT )  
NIP.P. 1030000365

Dosen Pembimbing II

( I Komang Sumawirata, ST. MT )  
NIP.P. 1030100361





## Source code Microcontroller

```
$regfile "m8535.dat"
$crystal = 11059200
$baud = 9600

Dim W As String * 1
Dim A As Word , Q As Word

Config Porta = Output , Portc = Output
Config Portb = Input .

Declare Sub Move_to_right()
Declare Sub Move_to_left()
Declare Sub Move_to_front()
Declare Sub Move_to_behind()
Declare Sub Start_drilling()
Declare Sub End_drilling()
Declare Sub Reset_x()
Declare Sub Reset_y()

Portc = &B11110000
'-----
'Sub Rutin

Do

'-----
'Motor X
    W = Waitkey()
    If W = "1" Then
        Portc = &B00000110
        Call Move_to_right()
    End If

    If W = "2" Then
        Portc = &B00000110
        Call Move_to_left()
    End If

'-----
'Motor Y

    If W = "3" Then
        Portc = &B00000110
        Call Move_to_front()
    End If
```

```

If W = "4" Then
    Portc = &B00000110
Call Move_to_behind()
End If

'-----
'Motor Z

If W = "5" Then
    Call Start_drilling()
End If

If W = "6" Then
    Call End_drilling()
End If

'-----
'Stop Motor
If W = "7" Then
    Porta = &B00000000
    Portc = &B11110000
End If

'-----
'Reset Motor
If W = "8" Then
    Call Reset_x()
End If

Loop
End

Sub Move_to_right()
If Pinb.0 = 0 Then
    Print "S"
Else
    For A = 1 To 50
        Porta = &B00001010
        Waitms 3
        Porta = &B00000110
        Waitms 3
        Porta = &B00000101
        Waitms 3
        Porta = &B00001001
        Waitms 3
    Next
End If
End Sub

Sub Move_to_left()
If Pinb.1 = 0 Then
    Print "S"
Else
    For A = 1 To 50
        Porta = &B00001010

```

```

Waitms 3
Porta = &B00001001
Waitms 3
Porta = &B00000101
Waitms 3
Porta = &B00000110
Waitms 3
Next
End If
End Sub

Sub Move_to_front()
If Pinb.3 = 0 Then
    Print "S"
Else
    For A = 1 To 50
    Porta = &B10100000
    Waitms 3
    Porta = &B10010000
    Waitms 3
    Porta = &B01010000
    Waitms 3
    Porta = &B01100000
    Waitms 3
    Next
End If
End Sub

Sub Move_to_behind()
If Pinb.2 = 0 Then
    Print "S"
Else
    For A = 1 To 50
    Porta = &B10100000
    Waitms 3
    Porta = &B01100000
    Waitms 3
    Porta = &B01010000
    Waitms 3
    Porta = &B10010000
    Waitms 3
    Next
End If
End Sub

Sub Start_drilling()
For A = 1 To 50
Portc = &B00001010
Waitms 3
Portc = &B00000110
Waitms 3
Portc = &B00000101
Waitms 3
Portc = &B00001001
Waitms 3
Next

```

```

End Sub

Sub End_drilling()
    For A = 1 To 50
        Portc = &B00001010
        Waitms 3
        Portc = &B00001001
        Waitms 3
        Portc = &B00000101
        Waitms 3
        Portc = &B00000110
        Waitms 3
    Next
End Sub

Sub Reset_x()
If Pinb.1 = 0 Then
    Call Reset_y()
Else
    For A = 1 To 50
        Porta = &B00001010
        Waitms 3
        Porta = &B00001001
        Waitms 3
        Porta = &B00000101
        Waitms 3
        Porta = &B00000110
        Waitms 3
    Next
End If
End Sub

Sub Reset_y()
If Pinb.2 = 0 Then
    Porta = &B00000000
Else
    For A = 1 To 50
        Porta = &B10100000
        Waitms 3
        Porta = &B01100000
        Waitms 3
        Porta = &B01010000
        Waitms 3
        Porta = &B10010000
        Waitms 3
    Next
End If
End Sub

```

## Source code Borland Delphi7

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, Grids, StdCtrls, ExtCtrls, DBCtrls, DBGrids,
  SUIButton, SUIEdit,
  SUIGrid, SUIMgr, SUIDlg, SUIImagePanel, CPort, SUIForm,
  SUIProgressBar;

type
  TForm1 = class(TForm)
    Timer1: TTimer;
    Timer2: TTimer;
    Timer3: TTimer;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Timer4: TTimer;
    Timer5: TTimer;
    Timer6: TTimer;
    suifileTheme1: TsuiFileTheme;
    suifileTheme2: TsuiFileTheme;
    StringGrid1: TsuiStringGrid;
    suiButton1: TsuiButton;
    suiButton2: TsuiButton;
    suiButton3: TsuiButton;
    suiButton4: TsuiButton;
    suiButton5: TsuiButton;
    suiButton6: TsuiButton;
    Label8: TLabel;
    suiEdit2: TsuiEdit;
    suiEdit3: TsuiEdit;
    suiEdit1: TsuiEdit;
    suiEdit4: TsuiEdit;
    suiEdit5: TsuiEdit;
    suiEdit6: TsuiEdit;
    suiMessageDialog1: TsuiMessageDialog;
    Timer7: TTimer;
    Label9: TLabel;
    suiPanel1: TsuiPanel;
    suiMessageDialog2: TsuiMessageDialog;
    suiImagePanel1: TsuiImagePanel;
```

```

ComPort1: TComPort;
GroupBox1: TGroupBox;
suiStringGrid2: TsuiStringGrid;
GroupBox2: TGroupBox;
GroupBox3: TGroupBox;
Label11: TLabel;
Label12: TLabel;
Timer8: TTimer;
GroupBox4: TGroupBox;
suiButton7: TsuiButton;
suiEdit7: TsuiEdit;
Label10: TLabel;
Label13: TLabel;
suiButton8: TsuiButton;
suiImagePanel2: TsuiImagePanel;
Label14: TLabel;

procedure FormCreate(Sender: TObject);
procedure suiButton2Click(Sender: TObject);
procedure suiButton1Click(Sender: TObject);
procedure suiButton3Click(Sender: TObject);
procedure suiButton4Click(Sender: TObject);
procedure suiButton5Click(Sender: TObject);
procedure suiButton6Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure Timer3Timer(Sender: TObject);
procedure Timer4Timer(Sender: TObject);
procedure Timer5Timer(Sender: TObject);
procedure Timer6Timer(Sender: TObject);
procedure Timer7Timer(Sender: TObject);
procedure Timer8Timer(Sender: TObject);
procedure suiButton7Click(Sender: TObject);

private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  i,j,k,l,m,n,o,r,s,gerak : Integer;
  a,b,c,x,y,z : real;
  Str: String;

implementation

uses Math;

{$R *.dfm}

procedure TForm1.FormCreate(Sender: TObject);
begin
  c:=0;

```

```

i:=0;
j:=12;
k:=0;
l:=0;
n:=0;
o:=0;
x:=0;
y:=0;
z:=0;
ComPort1.Open;
suiStringGrid2.Cells[0,0]:=' Step';
suiStringGrid2.Cells[1,0]:='    X ( mil )';
suiStringGrid2.Cells[2,0]:='    Y ( mil )';

end;

procedure TForm1.suiButton1Click(Sender: TObject);
begin

suiEdit6.Text:='0';
if i = 0 then
begin
  MessageDlg('Insert Data X and Y',mtInformation,[mbOK],0)
end;
if (z > 0) and (i = 0) then
begin
MessageDlg('Please Check Again',mtInformation,[mbOK],0)
end;
if z = 0  then
begin
  MessageDlg('Insert Data Z',mtInformation,[mbOK],0)
end;
if (z > 0) and (i > 0) then
begin

suiButton1.Enabled:=false;
suiButton3.Enabled:=false;
suiButton4.Enabled:=false;
suiButton5.Enabled:=false;
suiButton6.Enabled:=false;
suiButton7.Enabled:=false;
if k > l then
  Timer2.Enabled:=true
else if l > n then
  Timer3.Enabled:=true
else if n > o then
  Timer4.Enabled:=true
else
  Timer1.Enabled:=true
end;

end;
procedure TForm1.suiButton2Click(Sender: TObject);
begin

```

```

Timer1.Enabled:=false;
Timer2.Enabled:=false;
Timer3.Enabled:=false;
Timer4.Enabled:=false;
Label11.Caption:='7';
Label12.Caption:='. Stop';
suiButton1.Enabled:=true;
suiButton3.Enabled:=true;
suiButton4.Enabled:=true;
suiButton5.Enabled:=true;
suiButton6.Enabled:=true;
suiButton7.Enabled:=true;
label5.Font.Color:=clWhite;
label6.Font.Color:=clWhite;
label10.Font.Color:=clWhite;

end;

procedure TForm1.suiButton3Click(Sender: TObject);
begin
  ComPort1.Open;
  Label11.Caption:='8';
  Label12.Caption:='. Reset';
  Timer1.Enabled:=false;
  Timer2.Enabled:=false;
  Timer3.Enabled:=false;
  Timer4.Enabled:=false;
  Timer8.Enabled:=true;

  a:=0;
  b:=0;
  c:=0;
  k:=0;
  l:=0;
  m:=0;
  n:=0;
  o:=0;
  x:=0;
  y:=0;
  z:=0;

  suiEdit1.Text:='0';
  suiEdit2.Text:='0';
  suiEdit3.Text:='0';
  suiEdit4.Text:='0';
  suiEdit5.Text:='0';
  suiEdit6.Text:='0';
  label10.Font.Color:=clWindow;
  StringGrid1.FixedRows:= 0;
end;

procedure TForm1.suiButton4Click(Sender: TObject);
begin

  StringGrid1.Cells[0,i]:=IntToStr(i+1);
  StringGrid1.Cells[1,i]:=suiEdit4.Text;

```

```

StringGrid1.Cells[2,i]:=suiEdit5.Text;
suiEdit7.Text:=suiEdit6.Text;
z:=StrToFloat(suiEdit7.Text);

i:=i+1;
suiEdit4.Text:='0';
suiEdit5.Text:='0';

if j = i then
begin
j:=i;
StringGrid1.RowCount:=j ;
j:=j+1;
end;
end;

procedure TForm1.suiButton5Click(Sender: TObject);
begin
if i = 0 then
begin
MessageDlg('Insert Data Input',mtInformation,[mbOK],0)
end
else
begin
i:=i-1;
StringGrid1.Cells[0,i]:='';
StringGrid1.Cells[1,i]:='';
StringGrid1.Cells[2,i]:='';

if j > 14 then
begin
j:=j-1;
StringGrid1.RowCount:=j-1 ;
end;
end;
end;
end;

procedure TForm1.suiButton6Click(Sender: TObject);
begin
Timer8.Enabled:=false;
ComPort1.Close;
if suiMessageDialog2.ShowModal =
suiMessageDialog2.Button2ModalResult then
  Timer7.Interval:= 0
else
  Timer7.Interval:= 100;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
x:=StrToFloat(suiEdit1.Text);
a:=StrToFloat(StringGrid1.Cells[1,k]);
y:=StrToFloat(suiEdit2.Text);
b:=StrToFloat(StringGrid1.Cells[1,1]);
suiEdit3.Text:=IntToStr(m+1);
StringGrid1.FixedRows:= m+1;

```

```

Timer8.Enabled:=true;

if x = a then
begin
  Timer2.Enabled:=true;
  k:=k+1;
  m:=m+1;
  Timer1.Enabled:=false;
end;

if x < a then
begin
x:= x + 1;
suiEdit1.Text:=FloatToStr(x);
label10.Font.Color:=clWhite;
label5.Font.Color:=clAqua;
Label11.Caption:='1';
Label12.Caption:=' X - Move To Right';
  if x = a then
    begin
      Timer2.Enabled:=true;
      k:=k+1;
      m:=m+1;
      Timer1.Enabled:=false;
    end;
end;

if x > a then
begin
x:= x - 1;
suiEdit1.Text:=FloatToStr(x);
label10.Font.Color:=clWhite;
label5.Font.Color:=clAqua;
Label11.Caption:='2';
Label12.Caption:=' X - Move To Left';
  if x = a then
    begin
      Timer2.Enabled:=true;
      k:=k+1;
      m:=m+1;
      Timer1.Enabled:=false;
    end;
end;
end;

procedure TForm1.Timer2Timer(Sender: TObject);
begin
y:=StrtoFloat(suiEdit2.Text);
b:=StrtoFloat(StringGrid1.Cells[2,1]);
if y = b then
begin
  Timer3.Enabled:=true;
  l:=l+1;
  Timer2.Enabled:=false;
end;

```

```

if y < b then
begin
y:= y + 1;
suiEdit2.Text:=FloatToStr(y);
label5.Font.Color:=clWhite;
label6.Font.Color:=clAqua;
Label11.Caption:='3';
Label12.Caption:='. Y - Move To Front';
if y = b then
begin
  label5.Font.Color:=clWhite;
  Timer3.Enabled:=true;
  l:=l+1;
  Timer2.Enabled:=false;
end;
end;

if y > b then
begin
y:= y - 1;
suiEdit2.Text:=FloatToStr(y);
label5.Font.Color:=clWhite;
label6.Font.Color:=clAqua;
Label11.Caption:='4';
Label12.Caption:='. Y - Move To Behind';
if y = b then
begin
  Timer3.Enabled:=true;
  l:=l+1;
  Timer2.Enabled:=false;
end;
end;
end;

procedure TForm1.Timer3Timer(Sender: TObject);
begin
z:=StrToInt(suiEdit7.Text);
label6.Font.Color:=clWhite;
label10.Font.Color:=clAqua;
Label11.Caption:='5';
Label12.Caption:='. Z - Start Drilling';

c:=c+1;
if c = z then
begin
  n:=n+1;
  Timer4.Enabled:=true;
  Timer3.Enabled:=false;
end;
end;

procedure TForm1.Timer4Timer(Sender: TObject);
begin
c:=c-1;
label6.Font.Color:=clWhite;
label10.Font.Color:=clSilver;

```

```

Label11.Caption:='6';
Label12.Caption:=' Z - End Drilling';
if c = 0 then
begin
  if m = i then
  begin
    Label11.Caption:='7';
    k:=0;
    l:=0;
    x:=0;
    y:=0;
    label10.Font.Color:=clWindow;
    Label11.Caption:='';
    Label12.Caption:='';
    suiButton1.Enabled:=true;
    suiButton3.Enabled:=true;
    suiButton4.Enabled:=true;
    suiButton5.Enabled:=true;
    suiButton6.Enabled:=true;
    suiButton7.Enabled:=true;
    Timer1.Enabled:=false;
    Timer2.Enabled:=false;
    Timer3.Enabled:=false;
    Timer4.Enabled:=false;
    Timer8.Enabled:=false;
    suiMessageDialog1.ShowModal;
  end
  else
  begin
    o:=o+1;
    suiEdit6.Color:=clWindow;
    Timer1.Enabled:=true;
    Timer4.Enabled:=false;
  end;
end;
end;

procedure TForm1.Timer5Timer(Sender: TObject);
begin
  case gerak of
  0: Label9.Caption:= ' CNC MACHINE ';
  1: Label9.Caption:= ' CNC MACHINE ';
  2: Label9.Caption:= 'CNC MACHINE ';
  3: Label9.Caption:= 'NC MACHINE C';
  4: Label9.Caption:= 'C MACHINE CN';
  5: Label9.Caption:= ' MACHINE CNC';
  6: Label9.Caption:= 'MACHINE CNC ';
  7: Label9.Caption:= 'ACHINE CNC H';
  8: Label9.Caption:= 'CHINE CNC MA';
  9: Label9.Caption:= 'HINE CNC MAC';
  10: Label9.Caption:= 'INE CNC MACH';
  11: Label9.Caption:= 'NE CNC MACHI';
  12: Label9.Caption:= 'E CNC MACHIN';
  13: Label9.Caption:= ' CNC MACHINE';
  14: Label9.Caption:= ' CNC MACHINE ';

```

```

15: Label9.Caption:= ' CNC MACHINE ';
16: Label9.Caption:= ' CNC MACHINE ';
17: Label9.Caption:= 'CNC MACHINE ';
18: Label9.Caption:= 'NC MACHINE C';
19: Label9.Caption:= 'C MACHINE CN';
20: Label9.Caption:= ' MACHINE CNC';
21: Label9.Caption:= ' MACHINE CNC ';
22: Label9.Caption:= 'ACHINE CNC H';
23: Label9.Caption:= 'CHINE CNC MA';
24: Label9.Caption:= 'HINE CNC MAC';
25: Label9.Caption:= 'INE CNC MACH';
26: Label9.Caption:= 'NE CNC MACHI';
27: Label9.Caption:= 'E CNC MACHIN';
28: Label9.Caption:= ' CNC MACHINE';
29: Label9.Caption:= ' CNC MACHINE ';
30: Label9.Caption:= ' CNC MACHINE ';
31: Label9.Caption:= ' ';
32: Label9.Caption:= ' CNC MACHINE ';
33: Label9.Caption:= ' ';
34: Label9.Caption:= ' CNC MACHINE ';
35: Label9.Caption:= ' ';
36: Label9.Caption:= ' CNC MACHINE ';

end;
inc(gerak);
if gerak = 37 then
begin

    Label9.Caption:= '';
    gerak:= 0;
end;
end;

procedure TForm1.Timer6Timer(Sender: TObject);
begin
    suiPanel1.Caption:=FormatDateTime ('hh:mm:ss',time);
end;

procedure TForm1.Timer7Timer(Sender: TObject);
begin
    Timer5.Enabled:=false;
    form1.AlphaBlendValue:= form1.AlphaBlendValue - 25 ;
    if form1.AlphaBlendValue <= 25 then
        begin

            Timer7.Interval:= 0;
            form1.AlphaBlendValue:= 0;
            Form1.Close;
        end;
end;

procedure TForm1.Timer8Timer(Sender: TObject);
begin
    Str := Label11.Caption;
    str := Str + #13#10;
    ComPort1.WriteString(Str);
    z:=StrToFloat(Label11.Caption);

```

```
end;

procedure TForm1.suiButton7Click(Sender: TObject);
begin
  ComPort1.ShowSetupDialog;
end;

end.
```

## Features

High-performance, Low-power AVR<sup>®</sup> 8-bit Microcontroller

### Advanced RISC Architecture

- 130 Powerful Instructions – Most Single Clock Cycle Execution
- 32 x 8 General Purpose Working Registers
- Fully Static Operation
- Up to 16 MIPS Throughput at 16 MHz
- On-chip 2-cycle Multiplier

### Nonvolatile Program and Data Memories

- 8K Bytes of In-System Self-Programmable Flash  
Endurance: 10,000 Write/Erase Cycles
- Optional Boot Code Section with Independent Lock Bits  
In-System Programming by On-chip Boot Program  
True Read-While-Write Operation
- 512 Bytes EEPROM  
Endurance: 100,000 Write/Erase Cycles
- 512 Bytes Internal SRAM
- Programming Lock for Software Security

### Peripheral Features

- Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
- Real Time Counter with Separate Oscillator
- Four PWM Channels
- 8-channel, 10-bit ADC
  - 8 Single-ended Channels
  - 7 Differential Channels for TQFP Package Only
  - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x for TQFP Package Only
- Byte-oriented Two-wire Serial Interface
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator

### Special Microcontroller Features

- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated RC Oscillator
- External and Internal Interrupt Sources
- Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby

### I/O and Packages

- 32 Programmable I/O Lines
- 40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad MLF

### Operating Voltages

- 2.7 - 5.5V for ATmega8535L
- 4.5 - 5.5V for ATmega8535

### Speed Grades

- 0 - 8 MHz for ATmega8535L
- 0 - 16 MHz for ATmega8535



# 8-bit AVR<sup>®</sup> Microcontroller with 8K Bytes In-System Programmable Flash

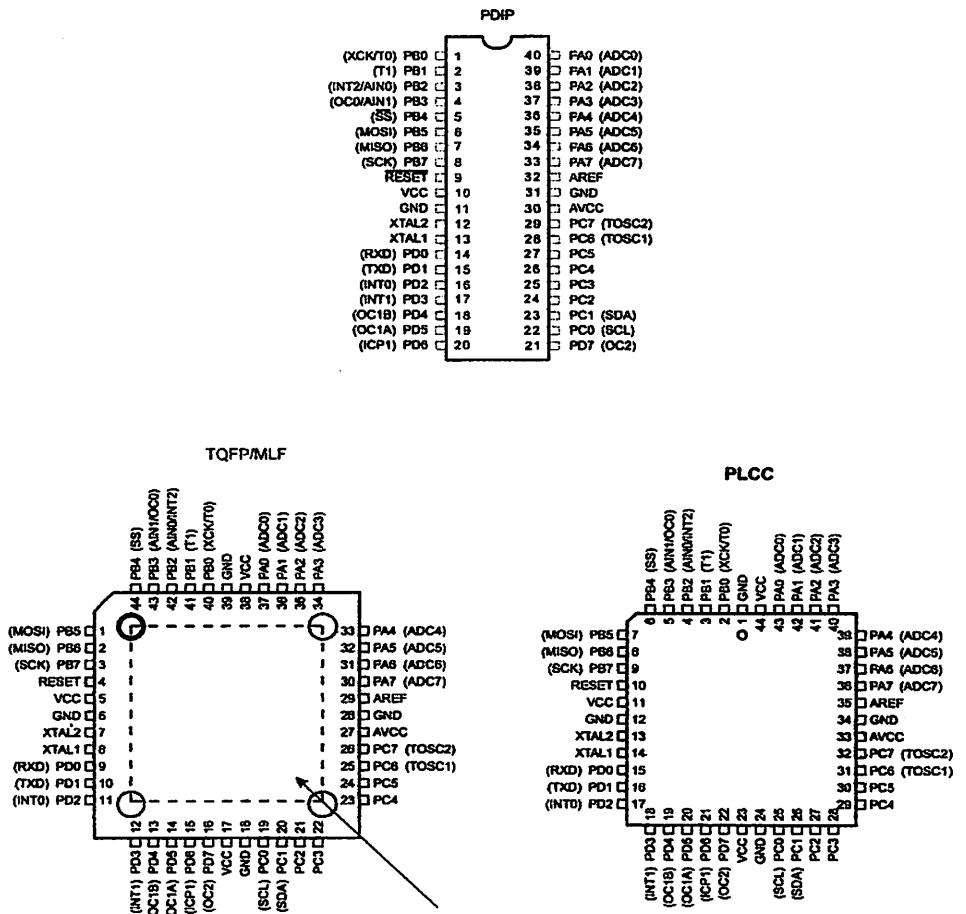
## ATmega8535 ATmega8535L

## Preliminary



# Pin Configurations

Figure 1. Pinout ATmega8535



NOTE: MLF Bottom pad should be soldered to ground.

## Disclaimer

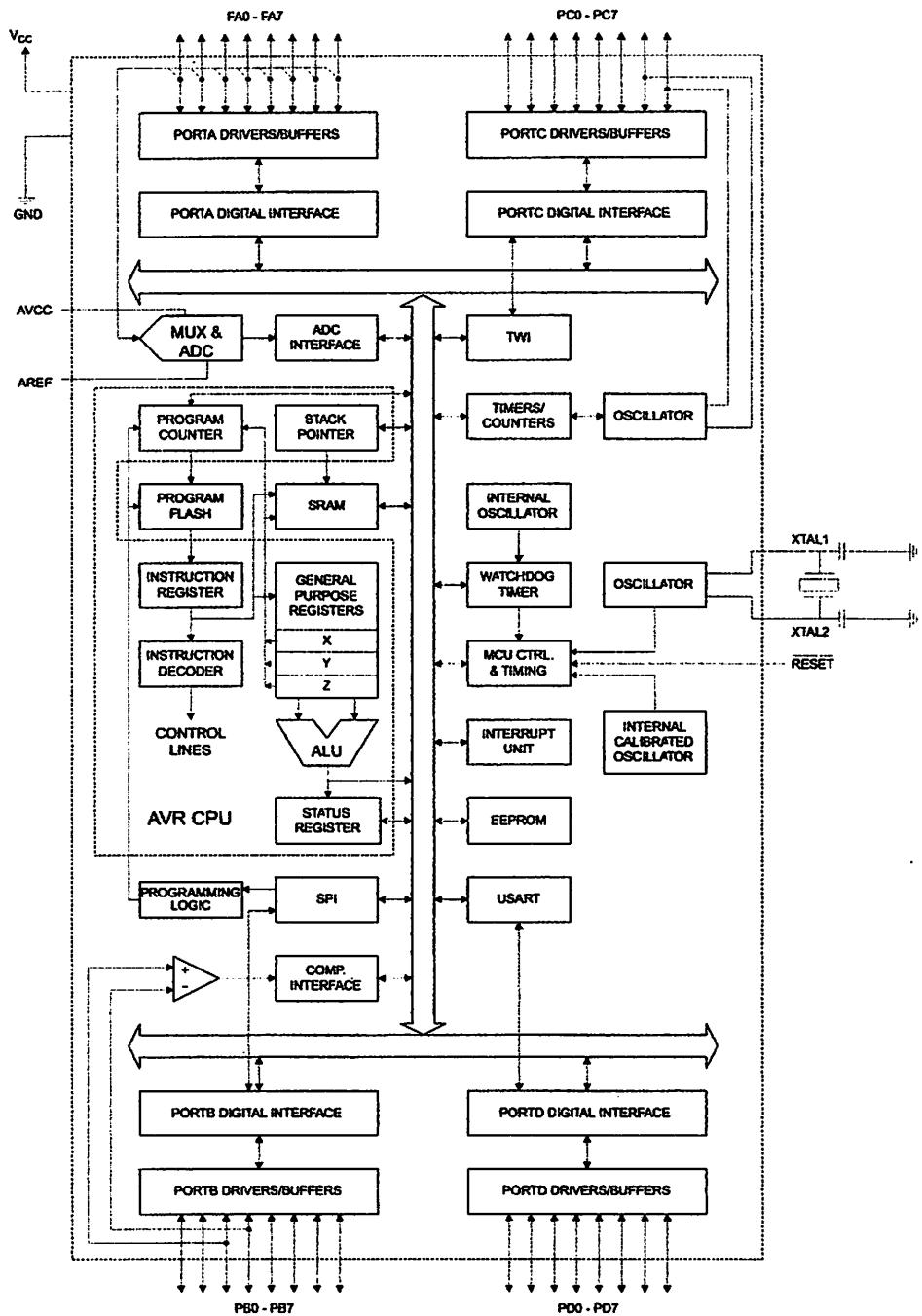
Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

## ATmega8535(L)

## Overview

The ATmega8535 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing instructions in a single clock cycle, the ATmega8535 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

## Block Diagram





The AVR core combines a rich instruction set with 32 general purpose working registers. All 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega8535 provides the following features: 8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes EEPROM, 512 bytes SRAM, 32 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain in TQFP package, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the asynchronous timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega8535 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega8535 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, In-Circuit Emulators, and evaluation kits.

## AT90S8535 Compatibility

The ATmega8535 provides all the features of the AT90S8535. In addition, several new features are added. The ATmega8535 is backward compatible with AT90S8535 in most cases. However, some incompatibilities between the two microcontrollers exist. To solve this problem, an AT90S8535 compatibility mode can be selected by programming the S8535C fuse. ATmega8535 is pin compatible with AT90S8535, and can replace the AT90S8535 on current Printed Circuit Boards. However, the location of fuse bits and the electrical characteristics differs between the two devices.

## AT90S8535 Compatibility Mode

Programming the S8535C fuse will change the following functionality:

- The timed sequence for changing the Watchdog Time-out period is disabled. See "Timed Sequences for Changing the Configuration of the Watchdog Timer" on page 43 for details.
- The double buffering of the USART Receive Register is disabled. See "AVR USART vs. AVR UART – Compatibility" on page 143 for details.

# ATmega8535(L)

## Pin Descriptions

V<sub>CC</sub>

Digital supply voltage.

GND

Ground.

Port A (PA7..PA0)

Port A serves as the analog inputs to the A/D Converter. Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATmega8535 as listed on page 58.

Port C (PC7..PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D (PD7..PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega8535 as listed on page 62.

RESET

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 35. Shorter pulses are not guaranteed to generate a reset.

XTAL1

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting Oscillator amplifier.

V<sub>VCC</sub>

AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to V<sub>CC</sub>, even if the ADC is not used. If the ADC is used, it should be connected to V<sub>CC</sub> through a low-pass filter.

AREF

AREF is the analog reference pin for the A/D Converter.



## About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C Compiler documentation for more details.

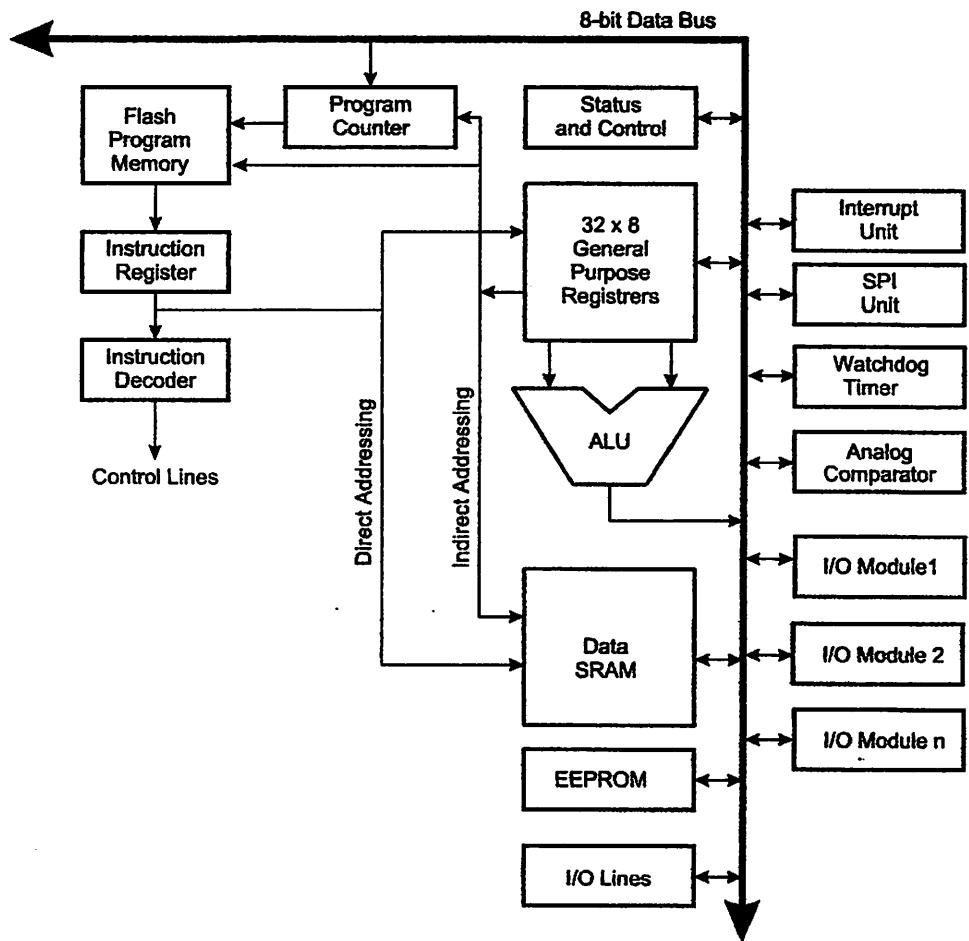
## AVR CPU Core

### Introduction

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

### Architectural Overview

**Figure 3. Block Diagram of the AVR MCU Architecture**



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept

## ATmega8535(L)

enables instructions to be executed in every clock cycle. The program memory is In-System Re-Programmable Flash memory.

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of these address pointers can also be used as an address pointer for look up tables in Flash program memory. These added function registers are the 16-bit X-, Y-, and Z-registers, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

Program Flash memory space is divided in two sections, the Boot Program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The Stack Pointer SP is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional Global Interrupt Enable bit in the Status Register. All interrupts have a separate Interrupt Vector in the Interrupt Vector table. The interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, 0x20 - 0x5F.

## ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the "Instruction Set" section for a detailed description.





## Status Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will, in many cases, remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	SREG
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I: Global Interrupt Enable**

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- **Bit 6 – T: Bit Copy Storage**

The Bit Copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source or destination for the operated bit. A bit from a register in the Register file can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half carry is useful in BCD arithmetic. See the "Instruction Set Description" for detailed information.

- **Bit 4 – S: Sign Bit,  $S = N \oplus V$**

The S-bit is always an exclusive or between the Negative Flag N and the Two's Complement Overflow Flag V. See the "Instruction Set Description" for detailed information.

- **Bit 3 – V: Two's Complement Overflow Flag**

The Two's Complement Overflow Flag V supports two's complement arithmetics. See the "Instruction Set Description" for detailed information.

- **Bit 2 – N: Negative Flag**

The Negative Flag N indicates a negative result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

- **Bit 1 – Z: Zero Flag**

The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

- **Bit 0 – C: Carry Flag**

The Carry Flag C indicates a carry in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

## General Purpose Register File

The Register File is optimized for the AVR Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

Figure 4 shows the structure of the 32 general purpose working registers in the CPU.

**Figure 4. AVR CPU General Purpose Working Registers**

General Purpose Working Registers	7      0	Addr.
R0		0x00
R1		0x01
R2		0x02
...		
R13		0x0D
R14		0x0E
R15		0x0F
R16		0x10
R17		0x11
...		
R26		0x1A
R27		0x1B
R28		0x1C
R29		0x1D
R30		0x1E
R31		0x1F
		X-register Low Byte
		X-register High Byte
		Y-register Low Byte
		Y-register High Byte
		Z-register Low Byte
		Z-register High Byte

Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions.

As shown in Figure 4, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y-, and Z-pointer Registers can be set to index any register in the file.





## The X-register, Y-register, and Z-register

The registers R26..R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the Data Space. The three indirect address registers X, Y, and Z are defined as described in Figure 5.

**Figure 5. The X-, Y-, and Z-registers**

X-register	15 7	XH 0   7	XL 0	0
		R27 (0x1B)	R26 (0x1A)	
Y-register	15 7	YH 0   7	YL 0	0
		R29 (0x1D)	R28 (0x1C)	
Z-register	15 7	ZH 0   7	ZL 0	0
		R31 (0x1F)	R30 (0x1E)	

In the different addressing modes, these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the instruction set reference for details).

## Stack Pointer

The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. The Stack Pointer Register always points to the top of the Stack. Note that the Stack is implemented as growing from higher memory locations to lower memory locations. This implies that a Stack PUSH command decreases the Stack Pointer.

The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer must be set to point above 0x60. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when the return address is pushed onto the Stack with subroutine call or interrupt. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when data is popped from the Stack with return from subroutine RET or return from interrupt RETI.

The AVR Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present.

Bit	15	14	13	12	11	10	9	8	SPH
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPL
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
ReadWrite	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

## Instruction Execution Timing

This section describes the general access timing concepts for instruction execution. The AVR CPU is driven by the CPU clock  $\text{clk}_{\text{CPU}}$ , directly generated from the selected clock source for the chip. No internal clock division is used.

Figure 6 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access Register File concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

**Figure 6. The Parallel Instruction Fetches and Instruction Executions**

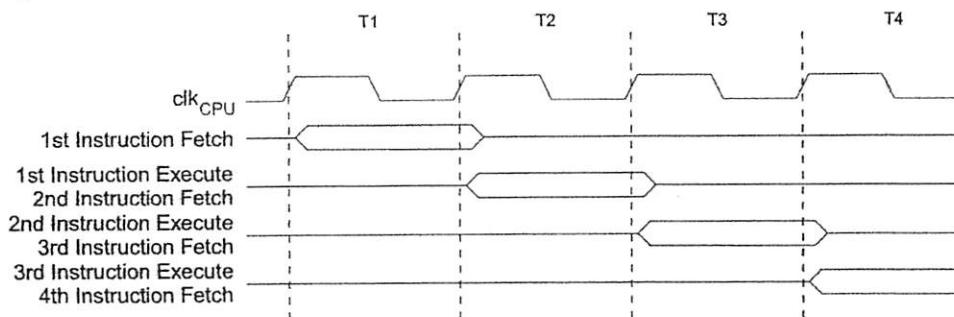
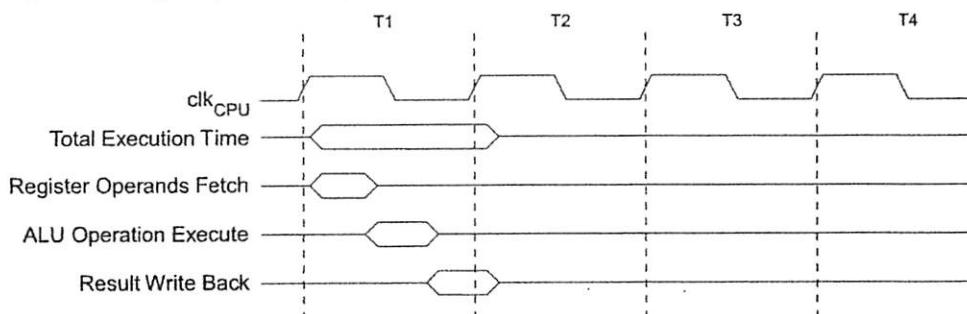


Figure 7 shows the internal timing concept for the Register file. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

**Figure 7. Single Cycle ALU Operation**



## Reset and Interrupt Handling

The AVR provides several different interrupt sources. These interrupts and the separate Reset Vector each have a separate Program Vector in the program memory space. All interrupts are assigned individual enable bits which must be written logic one together with the Global Interrupt Enable bit in the Status Register in order to enable the interrupt. Depending on the Program Counter value, interrupts may be automatically disabled when Boot Lock bits BLB02 or BLB12 are programmed. This feature improves software security. See the section "Memory Programming" on page 234 for details.

The lowest addresses in the program memory space are, by default, defined as the Reset and Interrupt Vectors. The complete list of vectors is shown in "Interrupts" on page 44. The list also determines the priority levels of the different interrupts. The lower the address, the higher the priority level is. RESET has the highest priority, and next is INT0 – the External Interrupt Request 0. The Interrupt Vectors can be moved to the start of the Boot Flash section by setting the IVSEL bit in the General Interrupt Control Register (GICR). Refer to "Interrupts" on page 44 for more information. The Reset Vector can





also be moved to the start of the Boot Flash section by programming the BOOTRST Fuse, see "Boot Loader Support – Read-While-Write Self-Programming" on page 221.

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared and all interrupts are disabled. The user software can write logic one to the I-bit to enable nested interrupts. All enabled interrupts can then interrupt the current interrupt routine. The I-bit is automatically set when a Return from Interrupt instruction – RETI – is executed.

There are basically two types of interrupts. The first type is triggered by an event that sets the interrupt flag. For these interrupts, the Program Counter is vectored to the actual Interrupt Vector in order to execute the interrupt handling routine, and hardware clears the corresponding interrupt flag. Interrupt flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared. If an interrupt condition occurs while the corresponding interrupt enable bit is cleared, the interrupt flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software. Similarly, if one or more interrupt conditions occur while the Global Interrupt Enable bit is cleared, the corresponding interrupt flag(s) will be set and remembered until the Global Interrupt Enable bit is set, and will then be executed by order of priority.

The second type of interrupts will trigger as long as the interrupt condition is present. These interrupts do not necessarily have interrupt flags. If the interrupt condition disappears before the interrupt is enabled, the interrupt will not be triggered.

When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

Note that the Status Register is not automatically stored when entering an interrupt routine, nor restored when returning from an interrupt routine. This must be handled by software.

When using the CLI instruction to disable interrupts, the interrupts will be immediately disabled. No interrupt will be executed after the CLI instruction, even if it occurs simultaneously with the CLI instruction. The following example shows how this can be used to avoid interrupts during the timed EEPROM write sequence.

#### Assembly Code Example

```
in r16, SREG      ; store SREG value
cli               ; disable interrupts during timed sequence
sbi EECR, EEMWE ; start EEPROM write
sbi EECR, EEWE
out SREG, r16    ; restore SREG value (I-bit)
```

#### C Code Example

```
char cSREG;
cSREG = SREG; /* store SREG value */
/* disable interrupts during timed sequence */
_CLI();
EECR |= (1<<EEMWE); /* start EEPROM write */
EECR |= (1<<EEWE);
SREG = cSREG; /* restore SREG value (I-bit) */
```

When using the SEI instruction to enable interrupts, the instruction following SEI will be executed before any pending interrupts, as shown in this example.

### Assembly Code Example

```
sei    ; set global interrupt enable  
sleep ; enter sleep, waiting for interrupt  
; note: will enter sleep before any pending  
; interrupt(s)
```

### C Code Example

```
_SEI(); /* set global interrupt enable */  
_SLEEP(); /* enter sleep, waiting for interrupt */  
/* note: will enter sleep before any pending interrupt(s) */
```

### Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. After four clock cycles, the Program Vector address for the actual interrupt handling routine is executed. During this four clock cycle period, the Program Counter is pushed onto the Stack. The Vector is normally a jump to the interrupt routine, and this jump takes three clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served. If an interrupt occurs when the MCU is in sleep mode, the interrupt execution response time is increased by four clock cycles. This increase comes in addition to the start-up time from the selected sleep mode.

A return from an interrupt handling routine takes four clock cycles. During these four clock cycles, the Program Counter (two bytes) is popped back from the Stack, the Stack Pointer is incremented by two, and the I-bit in SREG is set.





## AVR ATmega8535 Memories

### In-System Reprogrammable Flash Program Memory

This section describes the different memories in the ATmega8535. The AVR architecture has two main memory spaces, the Data Memory and the Program Memory space. In addition, the ATmega8535 features an EEPROM Memory for data storage. All three memory spaces are linear and regular.

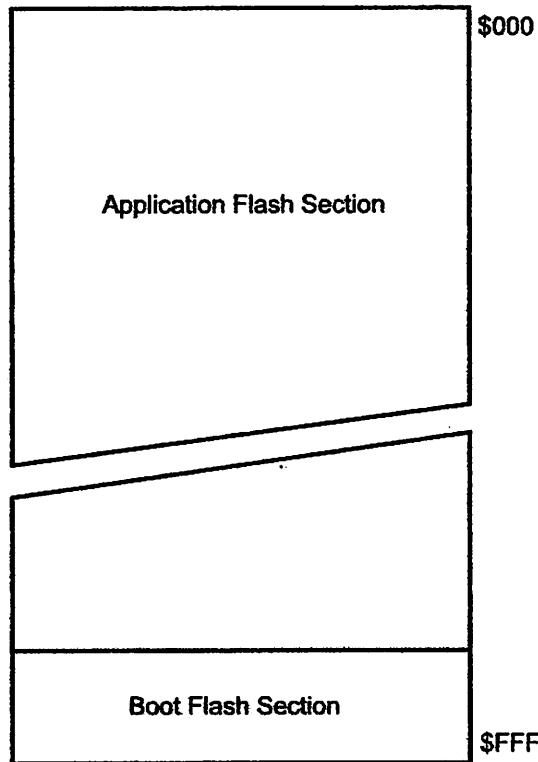
The ATmega8535 contains 8K bytes On-chip In-System Reprogrammable Flash memory for program storage. Since all AVR instructions are 16 or 32 bits wide, the Flash is organized as 4K x 16. For software security, the Flash Program memory space is divided into two sections, Boot Program section and Application Program section.

The Flash memory has an endurance of at least 10,000 write/erase cycles. The ATmega8535 Program Counter (PC) is 12 bits wide, thus addressing the 4K program memory locations. The operation of Boot Program section and associated Boot Lock bits for software protection are described in detail in "Boot Loader Support – Read-While-Write Self-Programming" on page 221. "Memory Programming" on page 234 contains a detailed description on Flash Programming in SPI or Parallel Programming mode.

Constant tables can be allocated within the entire program memory address space (see the LPM – Load Program Memory instruction description).

Timing diagrams for instruction fetch and execution are presented in "Instruction Execution Timing" on page 11.

**Figure 8. Program Memory Map**



## RAM Data Memory

Figure 9 shows how the ATmega8535 SRAM Memory is organized.

The 608 Data Memory locations address the Register File, the I/O Memory, and the internal data SRAM. The first 96 locations address the Register File and I/O Memory, and the next 512 locations address the internal data SRAM.

The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement, and Indirect with Post-increment. In the Register File, registers R26 to R31 feature the indirect addressing pointer registers.

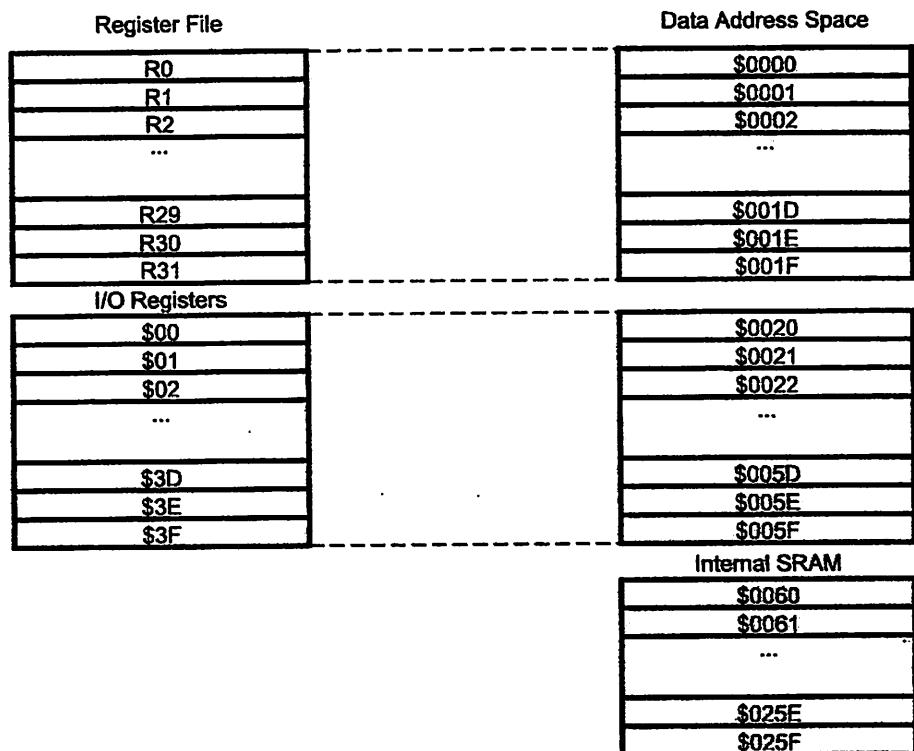
The direct addressing reaches the entire data space.

The Indirect with Displacement mode reaches 63 address locations from the base address given by the Y- or Z-register.

When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y, and Z are decremented or incremented.

The 32 general purpose working registers, 64 I/O Registers, and the 512 bytes of internal data SRAM in the ATmega8535 are all accessible through all these addressing modes. The Register File is described in "General Purpose Register File" on page 9.

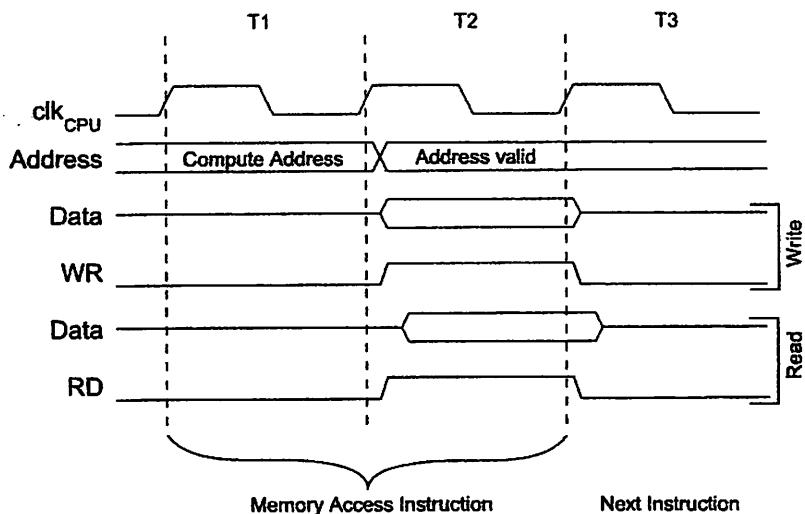
**Figure 9. Data Memory Map**



## Data Memory Access Times

This section describes the general access timing concepts for internal memory access. The internal data SRAM access is performed in two  $\text{clk}_{\text{CPU}}$  cycles as described in Figure 10.

**Figure 10. On-chip Data SRAM Access Cycles**



## EEPROM Data Memory

The ATmega8535 contains 512 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described in the following, specifying the EEPROM Address Registers, the EEPROM Data Register, and the EEPROM Control Register.

"Memory Programming" on page 234 contains a detailed description on EEPROM Programming in SPI or Parallel Programming mode.

## EEPROM Read/Write Access

The EEPROM Access Registers are accessible in the I/O space.

The write access time for the EEPROM is given in Table 1. A self-timing function, however, lets the user software detect when the next byte can be written. If the user code contains instructions that write the EEPROM, some precautions must be taken. In heavily filtered power supplies,  $V_{CC}$  is likely to rise or fall slowly on Power-up/down. This causes the device, for some period of time, to run at a voltage lower than specified as minimum for the clock frequency used, see "Preventing EEPROM Corruption" on page 20 for details on how to avoid problems in these situations.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. Refer to the description of the EEPROM Control Register for details on this.

When the EEPROM is read, the CPU is halted for four clock cycles before the next instruction is executed. When the EEPROM is written, the CPU is halted for two clock cycles before the next instruction is executed.

**The EEPROM Address Register – EEARH and EEARL**

Bit	15	14	13	12	11	10	9	8	EEARH
	—	—	—	—	—	—	—	EEAR8	EEARL
	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R/W	
	R/W								
Initial Value	0	0	0	0	0	0	0	X	
	X	X	X	X	X	X	X	X	

- Bits 15..9 – Res: Reserved Bits

These bits are reserved bits in the ATmega8535 and will always read as zero.

- Bits 8..0 – EEAR8..0: EEPROM Address

The EEPROM Address Registers – EEARH and EEARL specify the EEPROM address in the 512 bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 511. The initial value of EEAR is undefined. A proper value must be written before the EEPROM may be accessed.

**The EEPROM Data Register – EEDR**

Bit	7	6	5	4	3	2	1	0	EEDR
	MSB							LSB	
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

- Bits 7..0 – EEDR7..0: EEPROM Data

For the EEPROM write operation, the EEDR Register contains the data to be written to the EEPROM in the address given by the EEAR Register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

**The EEPROM Control Register – EECR**

Bit	7	6	5	4	3	2	1	0	EECR
	—	—	—	—	EERIE	EEMWE	EEWE	EERE	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	X	0	

- Bits 7..4 – Res: Reserved Bits

These bits are reserved bits in the ATmega8535 and will always read as zero.

- Bit 3 – EERIE: EEPROM Ready Interrupt Enable

Writing EERIE to one enables the EEPROM Ready Interrupt if the I-bit in SREG is set. Writing EERIE to zero disables the interrupt. The EEPROM Ready interrupt generates a constant interrupt when EEWE is cleared.

- Bit 2 – EEMWE: EEPROM Master Write Enable

The EEMWE bit determines whether setting EEWE to one causes the EEPROM to be written. When EEMWE is set, setting EEWE within four clock cycles will write data to the EEPROM at the selected address if EEMWE is zero, setting EEWE will have no effect. When EEMWE has been written to one by software, hardware clears the bit to zero after four clock cycles. See the description of the EEWE bit for an EEPROM write procedure.

- Bit 1 – EEWE: EEPROM Write Enable

The EEPROM Write Enable Signal EEWE is the write strobe to the EEPROM. When address and data are correctly set up, the EEWE bit must be written to one to write the





value into the EEPROM. The EEMWE bit must be written to one before a logical one is written to EEWE, otherwise no EEPROM write takes place. The following procedure should be followed when writing the EEPROM (the order of steps 3 and 4 is not essential):

1. Wait until EEWE becomes zero.
2. Wait until SPMEN in SPMCR becomes zero.
3. Write new EEPROM address to EEAR (optional).
4. Write new EEPROM data to EEDR (optional).
5. Write a logical one to the EEMWE bit while writing a zero to EEWE in EECR.
6. Within four clock cycles after setting EEMWE, write a logical one to EEWE.

The EEPROM can not be programmed during a CPU write to the Flash memory. The software must check that the Flash programming is completed before initiating a new EEPROM write. Step 2 is only relevant if the software contains a Boot Loader allowing the CPU to program the Flash. If the Flash is never updated by the CPU, step 2 can be omitted. See "Boot Loader Support – Read-While-Write Self-Programming" on page 221 for details about Boot programming.

**Caution:** An interrupt between step 5 and step 6 will make the write cycle fail, since the EEPROM Master Write Enable will time-out. If an interrupt routine accessing the EEPROM is interrupting another EEPROM access, the EEAR or EEDR Register will be modified, causing the interrupted EEPROM access to fail. It is recommended to have the Global interrupt Flag cleared during all the steps to avoid these problems.

When the write access time has elapsed, the EEWE bit is cleared by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEWE has been set, the CPU is halted for two cycles before the next instruction is executed.

- **Bit 0 – EERE: EEPROM Read Enable**

The EEPROM Read Enable Signal EERE is the read strobe to the EEPROM. When the correct address is set up in the EEAR Register, the EERE bit must be written to a logic one to trigger the EEPROM read. The EEPROM read access takes one instruction, and the requested data is available immediately. When the EEPROM is read, the CPU is halted for four cycles before the next instruction is executed.

The user should poll the EEWE bit before starting the read operation. If a write operation is in progress, it is neither possible to read the EEPROM, nor to change the EEAR Register.

The calibrated Oscillator is used to time the EEPROM accesses. Table 1 lists the typical programming time for EEPROM access from the CPU.

**Table 1. EEPROM Programming Time**

Symbol	Number of Calibrated RC Oscillator Cycles <sup>(1)</sup>	Typ Programming Time
EEPROM Write (from CPU)	8448	8.4 ms

Note: 1. Uses 1 MHz clock, independent of CKSEL Fuse settings.

The following code examples show one assembly and one C function for writing to the EEPROM. The examples assume that interrupts are controlled (e.g., by disabling interrupts globally) so that no interrupts will occur during execution of these functions. The examples also assume that no Flash Boot Loader is present in the software. If such code is present, the EEPROM write function must also wait for any ongoing SPM command to finish.

## Assembly Code Example

```
EEPROM_write:  
    ; Wait for completion of previous write  
    sbic EECR,EEWE  
    rjmp EEPROM_write  
    ; Set up address (r18:r17) in address register  
    out EEARH,r18  
    out EEARL,r17  
    ; Write data (r16) to Data Register  
    out EEDR,r16  
    ; Write logical one to EEMWE  
    sbi EECR,EEMWE  
    ; Start eeprom write by setting EEWE  
    sbi EECR,EEWE  
    ret
```

## C Code Example

```
void EEPROM_write(unsigned int uiAddress, unsigned char ucData)  
{  
    /* Wait for completion of previous write */  
    while(EECR & (1<<EEWE))  
        ;  
    /* Set up Address and Data Registers */  
    EEAR = uiAddress;  
    EEDR = ucData;  
    /* Write logical one to EEMWE */  
    EECR |= (1<<EEMWE);  
    /* Start eeprom write by setting EEWE */  
    EECR |= (1<<EEWE);  
}
```

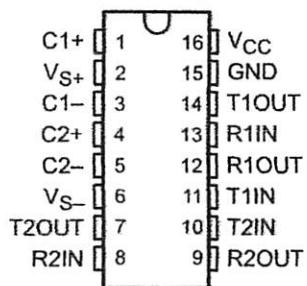


# MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047I – FEBRUARY 1989 – REVISED OCTOBER 2002

- Meet or Exceed TIA/EIA-232-F and ITU Recommendation V.28
- Operate With Single 5-V Power Supply
- Operate Up to 120 kbit/s
- Two Drivers and Two Receivers
- $\pm 30\text{-V}$  Input Levels
- Low Supply Current . . . 8 mA Typical
- Designed to be Interchangeable With Maxim MAX232
- ESD Protection Exceeds JESD 22 – 2000-V Human-Body Model (A114-A)
- Applications
  - TIA/EIA-232-F
  - Battery-Powered Systems
  - Terminals
  - Modems
  - Computers

MAX232 . . . D, DW, N, OR NS PACKAGE  
MAX232I . . . D, DW, OR N PACKAGE  
(TOP VIEW)



## description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept  $\pm 30\text{-V}$  inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

## ORDERING INFORMATION

<b>T<sub>A</sub></b>	<b>PACKAGE<sup>†</sup></b>		<b>ORDERABLE PART NUMBER</b>	<b>TOP-SIDE MARKING</b>
<i>0°C to 70°C</i>	PDIP (N)	Tube	MAX232N	MAX232N
	SOIC (D)	Tube	MAX232D	MAX232
		Tape and reel	MAX232DR	
	SOIC (DW)	Tube	MAX232DW	MAX232
		Tape and reel	MAX232DWR	
	SOP (NS)	Tape and reel	MAX232NSR	MAX232
<i>-40°C to 85°C</i>	PDIP (N)	Tube	MAX232IN	MAX232IN
	SOIC (D)	Tube	MAX232ID	MAX232I
		Tape and reel	MAX232IDR	
	SOIC (DW)	Tube	MAX232IDW	MAX232I
		Tape and reel	MAX232IDWR	

<sup>†</sup> Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at [www.ti.com/sc/package](http://www.ti.com/sc/package).



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

LinASIC is a trademark of Texas Instruments.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.


**TEXAS  
INSTRUMENTS**  
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2002, Texas Instruments Incorporated

# MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047I - FEBRUARY 1989 - REVISED OCTOBER 2002

## Function Tables

### EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

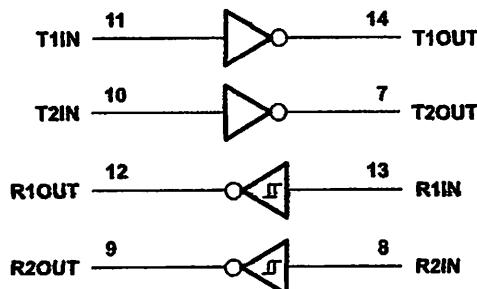
H = high level, L = low level

### EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

## logic diagram (positive logic)



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

# MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047I – FEBRUARY 1989 – REVISED OCTOBER 2002

## absolute maximum ratings over operating free-air temperature range (unless otherwise noted)<sup>†</sup>

Input supply voltage range, V <sub>CC</sub> (see Note 1)	.....	-0.3 V to 6 V
Positive output supply voltage range, V <sub>S+</sub>	.....	V <sub>CC</sub> – 0.3 V to 15 V
Negative output supply voltage range, V <sub>S-</sub>	.....	-0.3 V to -15 V
Input voltage range, V <sub>I</sub> : Driver	.....	-0.3 V to V <sub>CC</sub> + 0.3 V
Receiver	.....	±30 V
Output voltage range, V <sub>O</sub> : T1OUT, T2OUT	.....	V <sub>S-</sub> – 0.3 V to V <sub>S+</sub> + 0.3 V
R1OUT, R2OUT	.....	-0.3 V to V <sub>CC</sub> + 0.3 V
Short-circuit duration: T1OUT, T2OUT	.....	Unlimited
Package thermal impedance, θ <sub>JA</sub> (see Note 2): D package	.....	73°C/W
DW package	.....	57°C/W
N package	.....	67°C/W
NS package	.....	64°C/W
Lead temperature 1.6 mm (1/16 inch) from case for 10 seconds	.....	260°C
Storage temperature range, T <sub>stg</sub>	.....	-65°C to 150°C

<sup>†</sup> Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage values are with respect to network ground terminal.

2. The package thermal impedance is calculated in accordance with JESD 51-7.

## recommended operating conditions

		MIN	NOM	MAX	UNIT
V <sub>CC</sub>	Supply voltage	4.5	5	5.5	V
V <sub>IH</sub>	High-level input voltage (T1IN,T2IN)	2			V
V <sub>IL</sub>	Low-level input voltage (T1IN, T2IN)			0.8	V
R1IN, R2IN	Receiver input voltage			±30	V
T <sub>A</sub>	Operating free-air temperature	MAX232	0	70	°C
		MAX232I	-40	85	

## electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Note 3 and Figure 4)

PARAMETER	TEST CONDITIONS	MIN	TYP <sup>‡</sup>	MAX	UNIT
I <sub>CC</sub> Supply current	V <sub>CC</sub> = 5.5 V, All outputs open, T <sub>A</sub> = 25°C	8	10		mA

<sup>‡</sup> All typical values are at V<sub>CC</sub> = 5 V and T<sub>A</sub> = 25°C.

NOTE 3: Test conditions are C1–C4 = 1 μF at V<sub>CC</sub> = 5 V ± 0.5 V.

# MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047I – FEBRUARY 1989 – REVISED OCTOBER 2002

## DRIVER SECTION

**electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 3)**

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V <sub>OH</sub>	High-level output voltage	T <sub>1OUT</sub> , T <sub>2OUT</sub>	R <sub>L</sub> = 3 kΩ to GND	5	7	V
V <sub>OL</sub>	Low-level output voltage‡	T <sub>1OUT</sub> , T <sub>2OUT</sub>	R <sub>L</sub> = 3 kΩ to GND	-7	-5	V
r <sub>O</sub>	Output resistance	T <sub>1OUT</sub> , T <sub>2OUT</sub>	V <sub>S+</sub> = V <sub>S-</sub> = 0, V <sub>O</sub> = ±2 V	300		Ω
I <sub>OS</sub> §	Short-circuit output current	T <sub>1OUT</sub> , T <sub>2OUT</sub>	V <sub>CC</sub> = 5.5 V, V <sub>O</sub> = 0	±10		mA
I <sub>IS</sub>	Short-circuit input current	T <sub>1IN</sub> , T <sub>2IN</sub>	V <sub>I</sub> = 0	200		μA

† All typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.

‡ The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

§ Not more than one output should be shorted at a time.

NOTE 3: Test conditions are C<sub>1</sub>–C<sub>4</sub> = 1 μF at V<sub>CC</sub> = 5 V ± 0.5 V.

**switching characteristics, V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C (see Note 3)**

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
SR	Driver slew rate	R <sub>L</sub> = 3 kΩ to 7 kΩ, See Figure 2		30		V/μs
SR(t)	Driver transition region slew rate	See Figure 3		3		V/μs
	Data rate	One T <sub>OUT</sub> switching	120			kbit/s

NOTE 3: Test conditions are C<sub>1</sub>–C<sub>4</sub> = 1 μF at V<sub>CC</sub> = 5 V ± 0.5 V.

## RECEIVER SECTION

**electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 3)**

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT	
V <sub>OH</sub>	High-level output voltage	R <sub>1OUT</sub> , R <sub>2OUT</sub>	I <sub>OH</sub> = -1 mA	3.5		V	
V <sub>OL</sub>	Low-level output voltage‡	R <sub>1OUT</sub> , R <sub>2OUT</sub>	I <sub>OL</sub> = 3.2 mA		0.4	V	
V <sub>IT+</sub>	Receiver positive-going input threshold voltage	R <sub>1IN</sub> , R <sub>2IN</sub>	V <sub>CC</sub> = 5 V, T <sub>A</sub> = 25°C		1.7	2.4	V
V <sub>IT-</sub>	Receiver negative-going input threshold voltage	R <sub>1IN</sub> , R <sub>2IN</sub>	V <sub>CC</sub> = 5 V, T <sub>A</sub> = 25°C	0.8	1.2		V
V <sub>hys</sub>	Input hysteresis voltage	R <sub>1IN</sub> , R <sub>2IN</sub>	V <sub>CC</sub> = 5 V	0.2	0.5	1	V
r <sub>i</sub>	Receiver input resistance	R <sub>1IN</sub> , R <sub>2IN</sub>	V <sub>CC</sub> = 5, T <sub>A</sub> = 25°C	3	5	7	kΩ

† All typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.

‡ The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

NOTE 3: Test conditions are C<sub>1</sub>–C<sub>4</sub> = 1 μF at V<sub>CC</sub> = 5 V ± 0.5 V.

**switching characteristics, V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C (see Note 3 and Figure 1)**

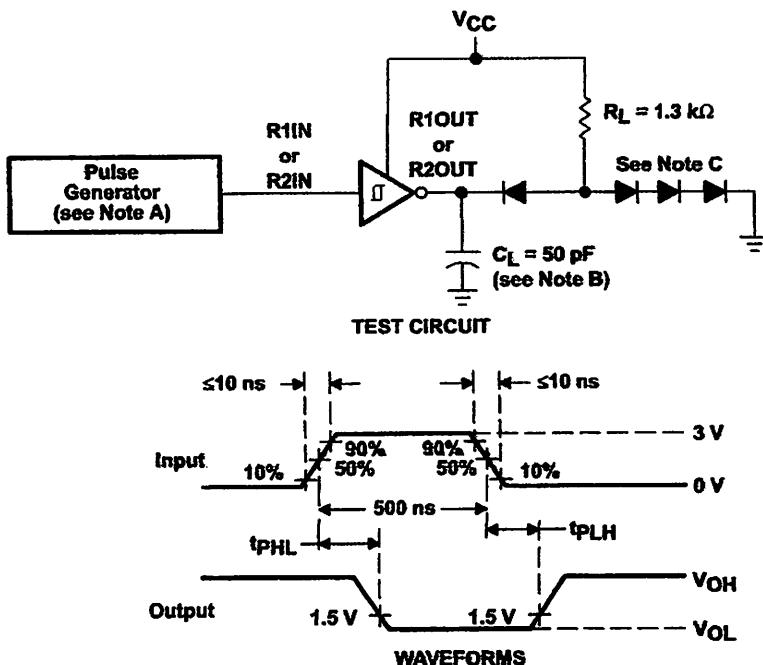
PARAMETER	TYP	UNIT
t <sub>PLH(R)</sub> Receiver propagation delay time, low- to high-level output	500	ns
t <sub>PHL(R)</sub> Receiver propagation delay time, high- to low-level output	500	ns

NOTE 3: Test conditions are C<sub>1</sub>–C<sub>4</sub> = 1 μF at V<sub>CC</sub> = 5 V ± 0.5 V.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

**PARAMETER MEASUREMENT INFORMATION**



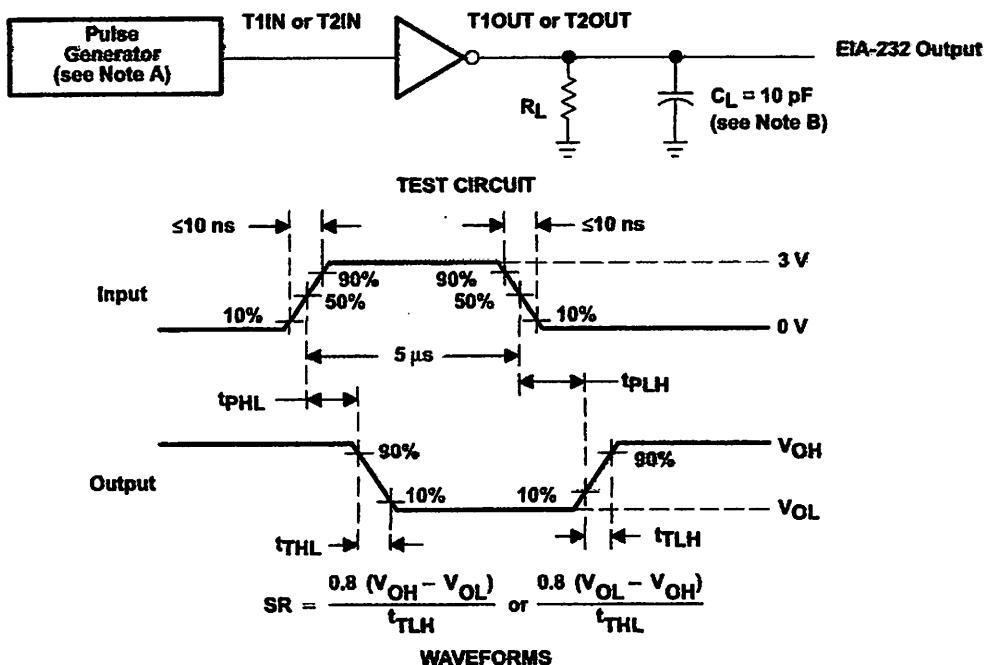
- NOTES:
- A. The pulse generator has the following characteristics:  $Z_O = 50 \Omega$ , duty cycle  $\leq 50\%$ .
  - B.  $C_L$  includes probe and jig capacitance.
  - C. All diodes are 1N3064 or equivalent.

**Figure 1. Receiver Test Circuit and Waveforms for  $t_{PHL}$  and  $t_{PLH}$  Measurements**

# MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

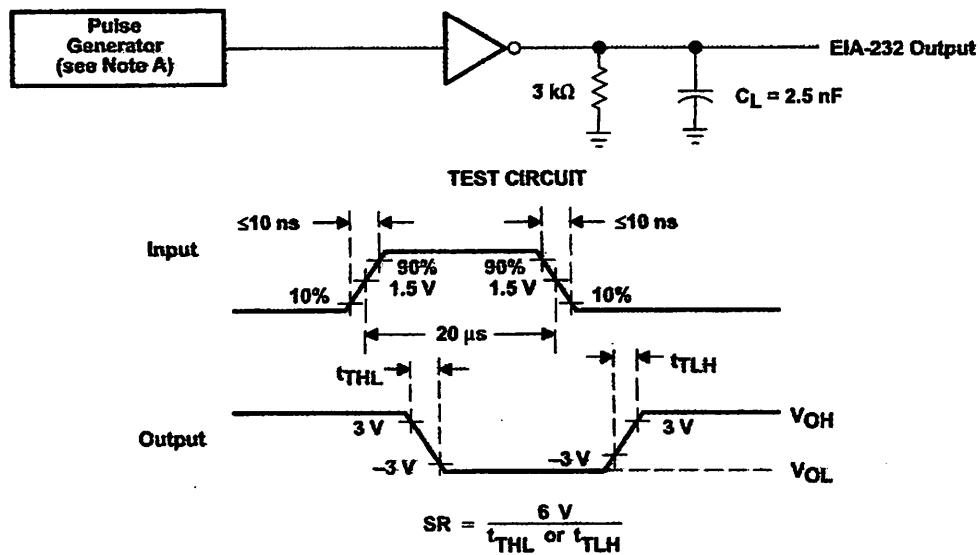
SLLS0471—FEBRUARY 1989—REVISED OCTOBER 2002

## PARAMETER MEASUREMENT INFORMATION



NOTES: A. The pulse generator has the following characteristics:  $Z_O = 50 \Omega$ , duty cycle  $\leq 50\%$ .  
 B.  $C_L$  includes probe and jig capacitance.

Figure 2. Driver Test Circuit and Waveforms for  $t_{PHL}$  and  $t_{PLH}$  Measurements (5-μs Input)



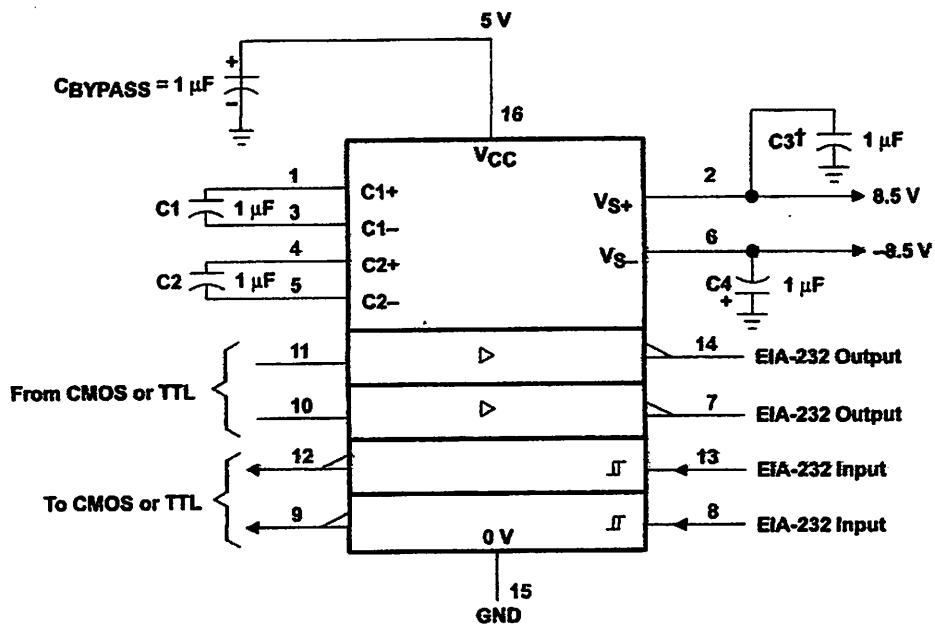
NOTE A: The pulse generator has the following characteristics:  $Z_O = 50 \Omega$ , duty cycle  $\leq 50\%$ .

Figure 3. Test Circuit and Waveforms for  $t_{THL}$  and  $t_{TLH}$  Measurements (20-μs Input)

**MAX232, MAX232I  
DUAL EIA-232 DRIVERS/RECEIVERS**

SLLS047I - FEBRUARY 1989 - REVISED OCTOBER 2002

**APPLICATION INFORMATION**



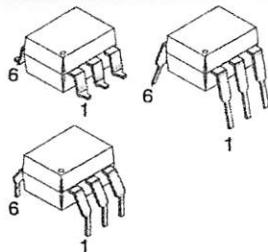
† C3 can be connected to V<sub>CC</sub> or GND.

**Figure 4. Typical Operating Circuit**

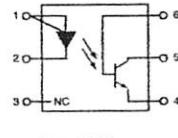
# GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPERS

4N25	4N26	4N27	4N28	4N35	4N36
4N37	H11A1	H11A2	H11A3	H11A4	H11A5

WHITE PACKAGE (-M SUFFIX)

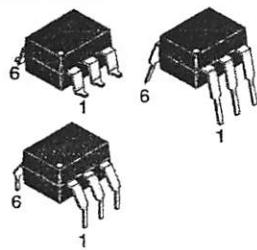


SCHEMATIC



PIN 1. ANODE  
2. CATHODE  
3. NO CONNECTION  
4. Emitter  
5. Collector  
6. Base

BLACK PACKAGE (NO -M SUFFIX)



## DESCRIPTION

The general purpose optocouplers consist of a gallium arsenide infrared emitting diode driving a silicon phototransistor in a 6-pin dual in-line package.

## FEATURES

- Also available in white package by specifying -M suffix, eg. 4N25-M
- UL recognized (File # E90700)
- VDE recognized (File # 94766)
  - Add option V for white package (e.g., 4N25V-M)
  - Add option 300 for black package (e.g., 4N25.300)

## APPLICATIONS

- Power supply regulators
- Digital logic inputs
- Microprocessor inputs



## GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPERS

**4N25  
4N37**

**4N26  
H11A1**

**4N27  
H11A2**

**4N28  
H11A3**

**4N35  
H11A4**

**4N36  
H11A5**

### ABSOLUTE MAXIMUM RATINGS ( $T_A = 25^\circ\text{C}$ unless otherwise specified)

Parameter	Symbol	Value	Units
<b>TOTAL DEVICE</b>			
Storage Temperature	$T_{STG}$	-55 to +150	°C
Operating Temperature	$T_{OPR}$	-55 to +100	°C
Wave solder temperature (see page 14 for reflow solder profiles)	$T_{SOL}$	260 for 10 sec	°C
Total Device Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	$P_D$	250 3.3 (non-M), 2.94 (-M)	mW
<b>EMITTER</b>			
DC/Average Forward Input Current	$I_F$	100 (non-M), 60 (-M)	mA
Reverse Input Voltage	$V_R$	6	V
Forward Current - Peak (300μs, 2% Duty Cycle)	$I_F(pk)$	3	A
LED Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	$P_D$	150 (non-M), 120 (-M) 2.0 (non-M), 1.41 (-M)	mW mW/°C
<b>DETECTOR</b>			
Collector-Emitter Voltage	$V_{CEO}$	30	V
Collector-Base Voltage	$V_{CBO}$	70	V
Emitter-Collector Voltage	$V_{ECO}$	7	V
Detector Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	$P_D$	150 2.0 (non-M), 1.76 (-M)	mW mW/°C

# GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPERS

4N25	4N26	4N27	4N28	4N35	4N36
4N37	H11A1	H11A2	H11A3	H11A4	H11A5

## ELECTRICAL CHARACTERISTICS ( $T_A = 25^\circ\text{C}$ unless otherwise specified)

### INDIVIDUAL COMPONENT CHARACTERISTICS

Parameter	Test Conditions	Symbol	Min	Typ*	Max	Unit
<b>EMITTER</b>						
Input Forward Voltage	( $I_F = 10 \text{ mA}$ )	$V_F$		1.18	1.50	V
Reverse Leakage Current	( $V_R = 6.0 \text{ V}$ )	$I_R$		0.001	10	$\mu\text{A}$
<b>DETECTOR</b>						
Collector-Emitter Breakdown Voltage	( $I_C = 1.0 \text{ mA}, I_F = 0$ )	$BV_{CEO}$	30	100		V
Collector-Base Breakdown Voltage	( $I_C = 100 \mu\text{A}, I_F = 0$ )	$BV_{CBO}$	70	120		V
Emitter-Collector Breakdown Voltage	( $I_E = 100 \mu\text{A}, I_F = 0$ )	$BV_{ECO}$	7	10		V
Collector-Emitter Dark Current	( $V_{CE} = 10 \text{ V}, I_F = 0$ )	$I_{CEO}$		1	50	nA
Collector-Base Dark Current	( $V_{CB} = 10 \text{ V}$ )	$I_{CBO}$			20	nA
Capacitance	( $V_{CE} = 0 \text{ V}, f = 1 \text{ MHz}$ )	$C_{CE}$		8		pF

### ISOLATION CHARACTERISTICS

Characteristic	Test Conditions	Symbol	Min	Typ*	Max	Units
Input-Output Isolation Voltage	(Non '-M', Black Package) ( $f = 60 \text{ Hz}, t = 1 \text{ min}$ )	$V_{ISO}$	5300			Vac(rms)
	('-M', White Package) ( $f = 60 \text{ Hz}, t = 1 \text{ sec}$ )		7500			Vac(pk)
Isolation Resistance	( $V_{I-O} = 500 \text{ VDC}$ )	$R_{ISO}$	$10^{11}$			$\Omega$
Isolation Capacitance	( $V_{I-O} = \&, f = 1 \text{ MHz}$ )	$C_{ISO}$		0.5		pF
	('-M' White Package)			0.2	2	pF

Note

\* Typical values at  $T_A = 25^\circ\text{C}$



# GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPERS

<b>4N25</b>	<b>4N26</b>	<b>4N27</b>	<b>4N28</b>	<b>4N35</b>	<b>4N36</b>
<b>4N37</b>	<b>H11A1</b>	<b>H11A2</b>	<b>H11A3</b>	<b>H11A4</b>	<b>H11A5</b>

## TRANSFER CHARACTERISTICS ( $T_A = 25^\circ\text{C}$ Unless otherwise specified.)

DC Characteristic	Test Conditions	Symbol	Device	Min	Typ*	Max	Unit	
Current Transfer Ratio, Collector to Emitter	$(I_F = 10 \text{ mA}, V_{CE} = 10 \text{ V})$	CTR	4N35					
			4N36	100				
			4N37					
	$(I_F = 10 \text{ mA}, V_{CE} = 10 \text{ V}, T_A = -55^\circ\text{C})$		H11A1	50				
			H11A5	30				
			4N25					
	$(I_F = 10 \text{ mA}, V_{CE} = 10 \text{ V}, T_A = +100^\circ\text{C})$		4N26	20				
			H11A2					
			H11A3					
Collector-Emitter Saturation Voltage	$(I_C = 2 \text{ mA}, I_F = 50 \text{ mA})$	V <sub>CE (SAT)</sub>	4N27					
			4N28	10				
			4N35					
			4N36	40				
	$(I_C = 0.5 \text{ mA}, I_F = 10 \text{ mA})$		4N37					
			4N25					
			4N26					
AC Characteristic	$(I_F = 10 \text{ mA}, V_{CC} = 10 \text{ V}, R_L = 100\Omega)$ (Fig.20)	T <sub>ON</sub>	4N27					
			4N28					
Non-Saturated Turn-on Time	$(I_C = 2 \text{ mA}, V_{CC} = 10 \text{ V}, R_L = 100\Omega)$ (Fig.20)	T <sub>ON</sub>	H11A1					
			H11A2					
Non Saturated Turn-on Time	$(I_C = 2 \text{ mA}, V_{CC} = 10 \text{ V}, R_L = 100\Omega)$ (Fig.20)	T <sub>ON</sub>	H11A3					
			H11A4					
			H11A5					
Non Saturated Turn-on Time	$(I_C = 2 \text{ mA}, V_{CC} = 10 \text{ V}, R_L = 100\Omega)$ (Fig.20)	T <sub>ON</sub>	4N35					
			4N36					
			4N37					

**GENERAL PURPOSE 6-PIN  
PHOTOTRANSISTOR OPTOCOUPERS**

**4N25  
4N37**

**4N26  
H11A1**

**4N27  
H11A2**

**4N28  
H11A3**

**4N35  
H11A4**

**4N36  
H11A5**

**TRANSFER CHARACTERISTICS ( $T_A = 25^\circ\text{C}$  Unless otherwise specified.) (Continued)**

AC Characteristic	Test Conditions	Symbol	Device	Min	Typ*	Max	Unit
Turn-off Time	( $I_F = 10 \text{ mA}$ , $V_{CC} = 10 \text{ V}$ , $R_L = 100\Omega$ (Fig.20))	$T_{OFF}$	4N25 4N26 4N27 4N28 H11A1 H11A2 H11A3 H11A4 H11A5		2		$\mu\text{s}$
	( $I_C = 2 \text{ mA}$ , $V_{CC} = 10 \text{ V}$ , $R_L = 100\Omega$ (Fig.20))		4N35 4N36 4N37		2	10	

\* Typical values at  $T_A = 25^\circ\text{C}$

# GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPERS

4N25  
4N37

4N26  
H11A1

4N27  
H11A2

4N28  
H11A3

4N35  
H11A4

4N36  
H11A5

## TYPICAL PERFORMANCE CURVES

Fig. 1 LED Forward Voltage vs. Forward Current  
(Black Package)

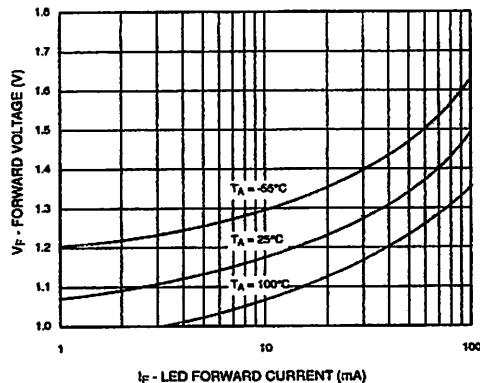


Fig. 2 LED Forward Voltage vs. Forward Current  
(White Package)

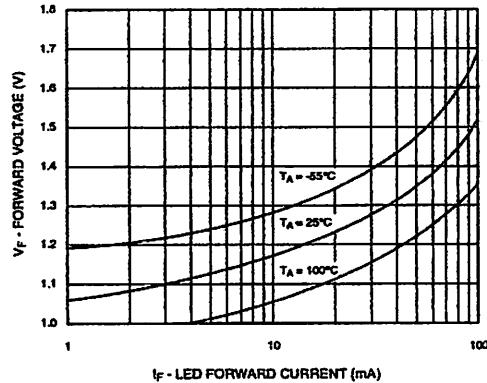


Fig. 3 Normalized CTR vs. Forward Current  
(Black Package)

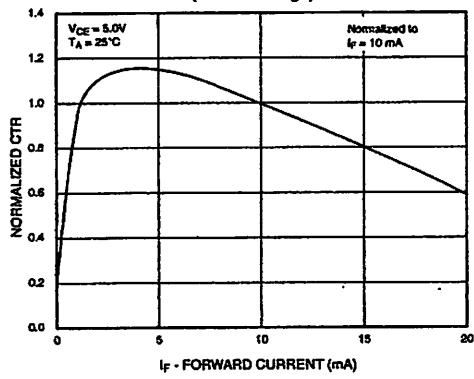


Fig. 4 Normalized CTR vs. Forward Current  
(White Package)

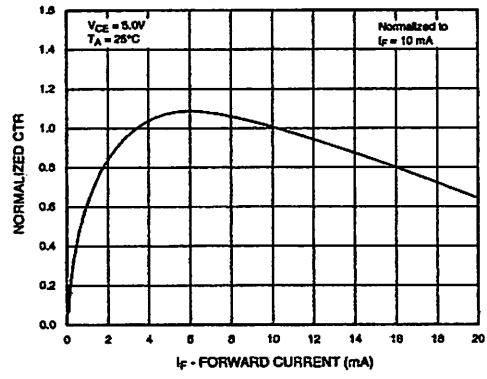


Fig. 5 Normalized CTR vs. Ambient Temperature  
(Black Package)

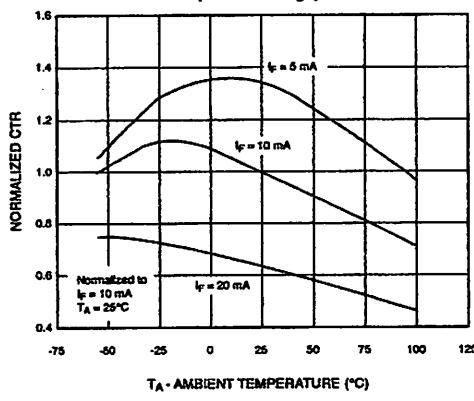
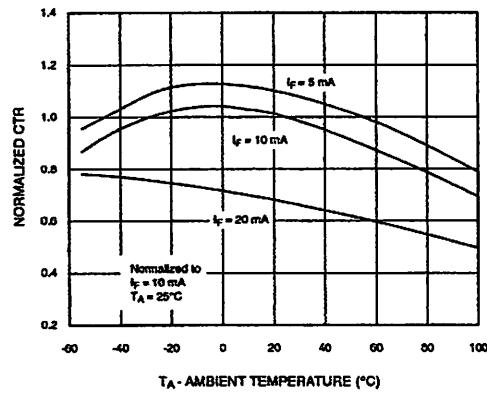
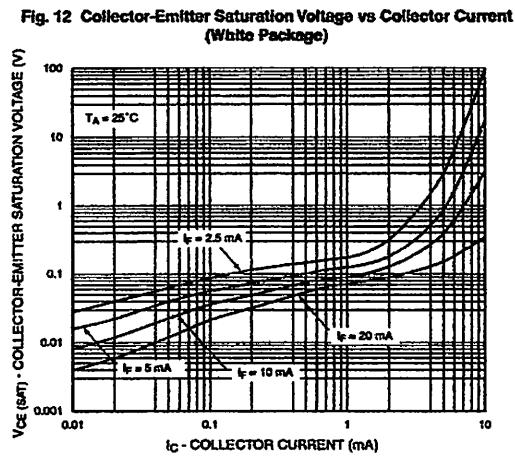
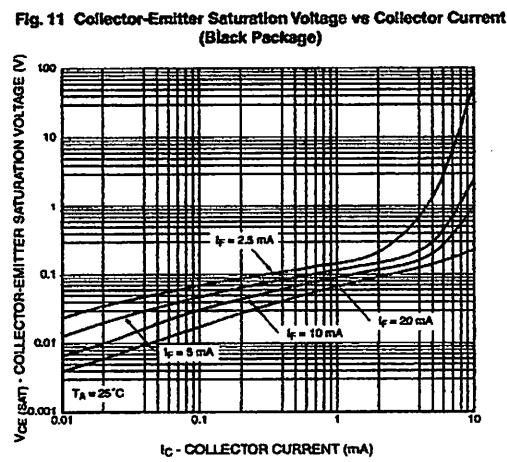
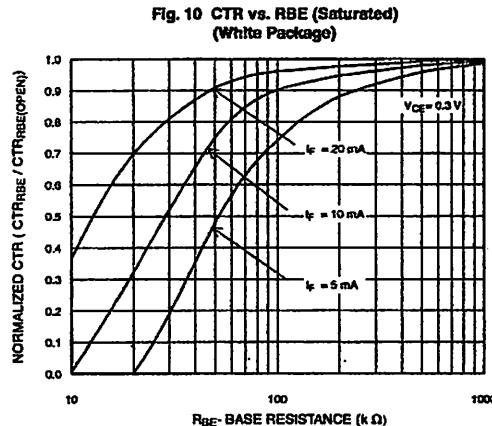
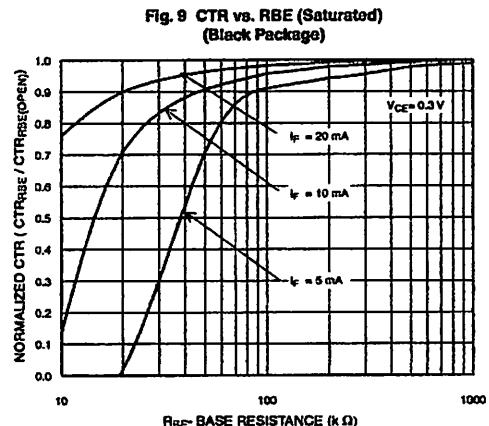
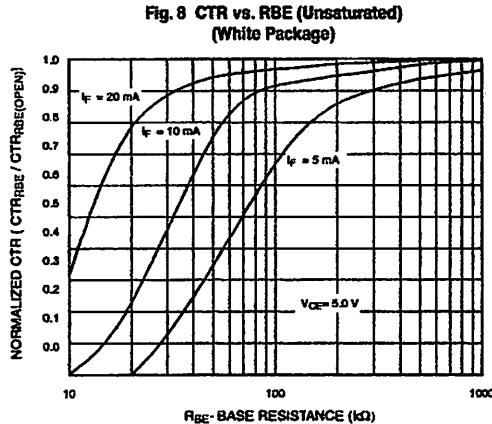
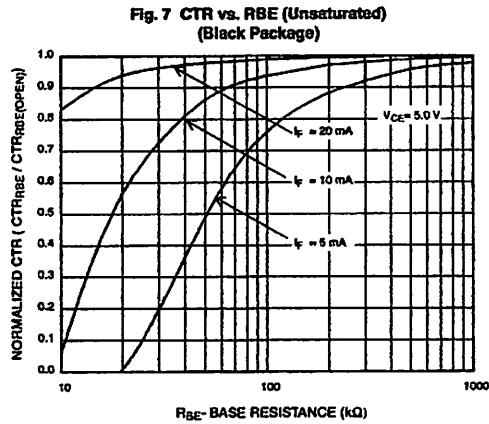


Fig. 6 Normalized CTR vs. Ambient Temperature  
(White Package)



# GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPLES

<b>4N25</b>	<b>4N26</b>	<b>4N27</b>	<b>4N28</b>	<b>4N35</b>	<b>4N36</b>
<b>4N37</b>	<b>H11A1</b>	<b>H11A2</b>	<b>H11A3</b>	<b>H11A4</b>	<b>H11A5</b>



# GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPERS

4N25  
4N37

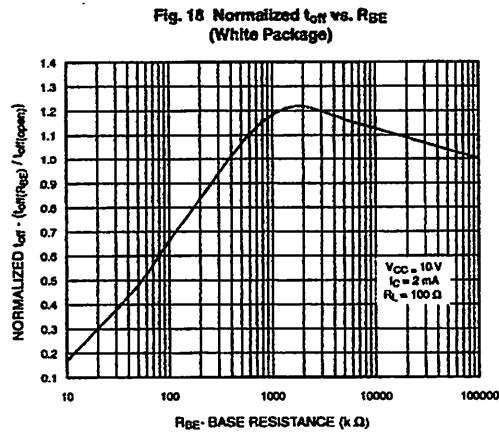
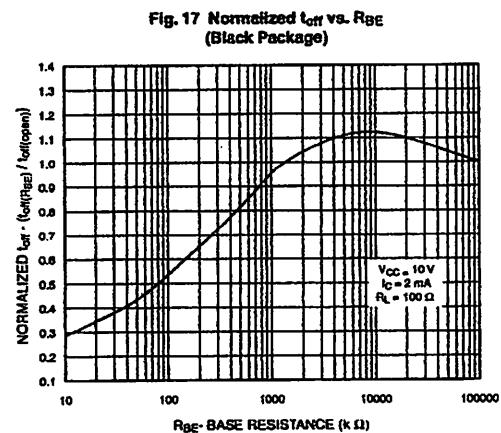
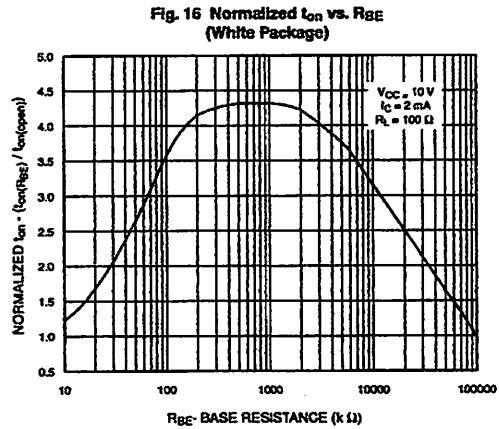
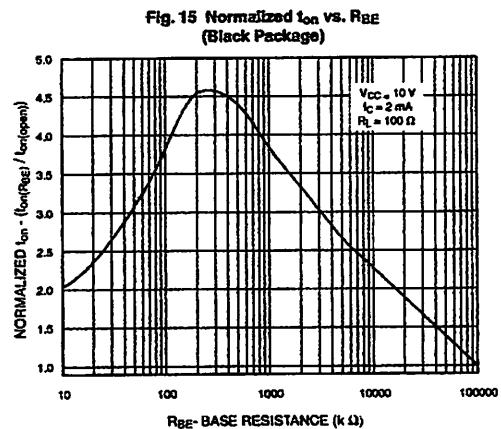
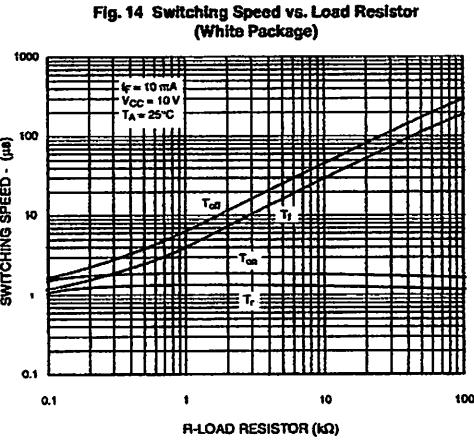
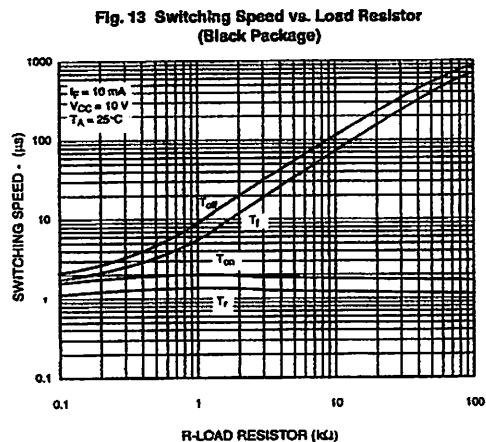
4N26  
H11A1

4N27  
H11A2

4N28  
H11A3

4N35  
H11A4

4N36  
H11A5



# GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPERS

4N25  
4N37

4N26  
H11A1

4N27  
H11A2

4N28  
H11A3

4N35  
H11A4

4N36  
H11A5

Fig. 19 Dark Current vs. Ambient Temperature

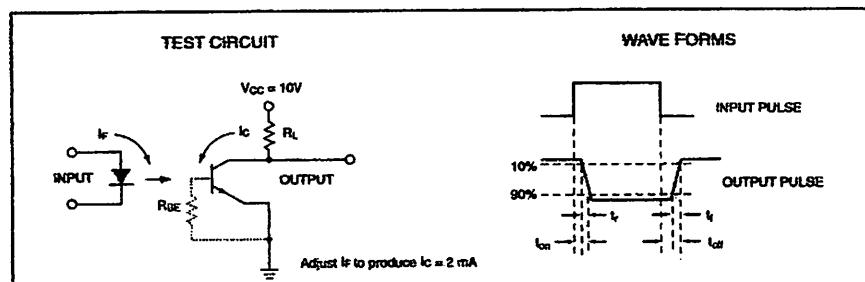
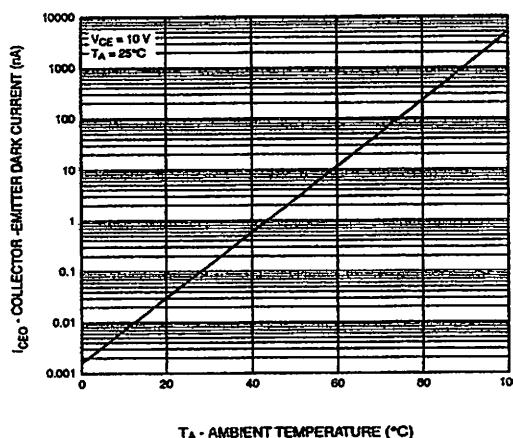


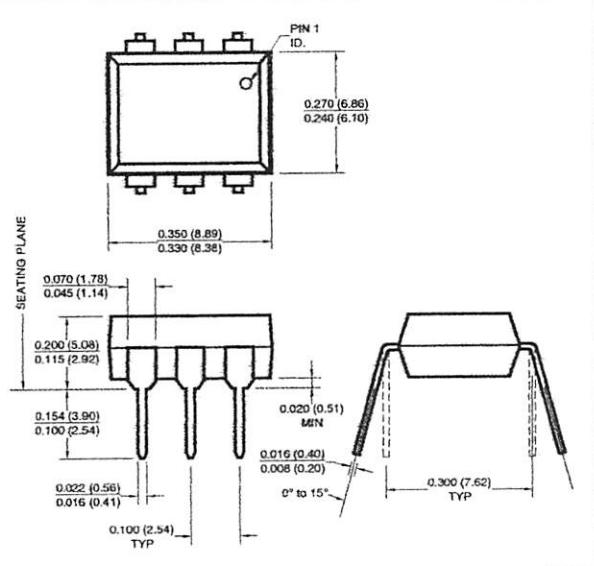
Figure 20. Switching Time Test Circuit and Waveforms

# GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPERS

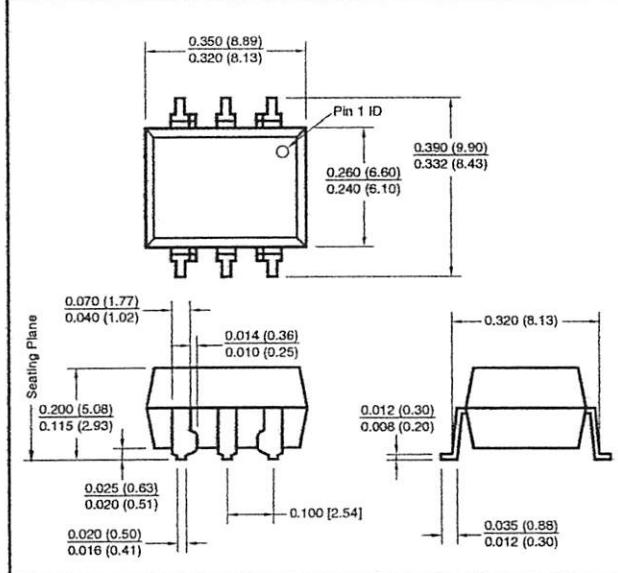
<b>4N25</b>	<b>4N26</b>	<b>4N27</b>	<b>4N28</b>	<b>4N35</b>	<b>4N36</b>
<b>4N37</b>	<b>H11A1</b>	<b>H11A2</b>	<b>H11A3</b>	<b>H11A4</b>	<b>H11A5</b>

## Black Package (No -M Suffix)

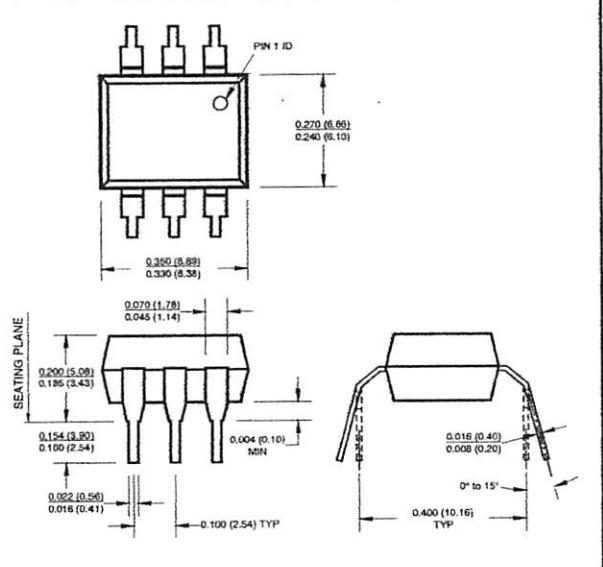
### Package Dimensions (Through Hole)



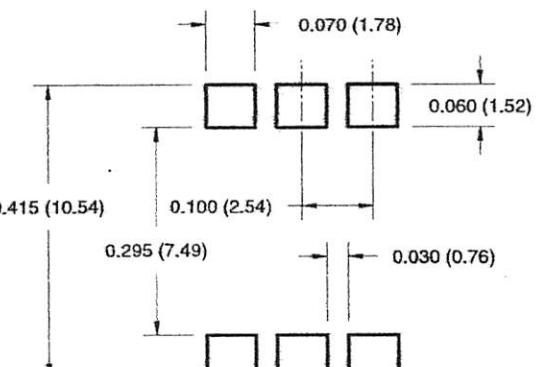
### Package Dimensions (Surface Mount)



### Package Dimensions (0.4" Lead Spacing)



### Recommended Pad Layout for Surface Mount Leadform



### NOTE

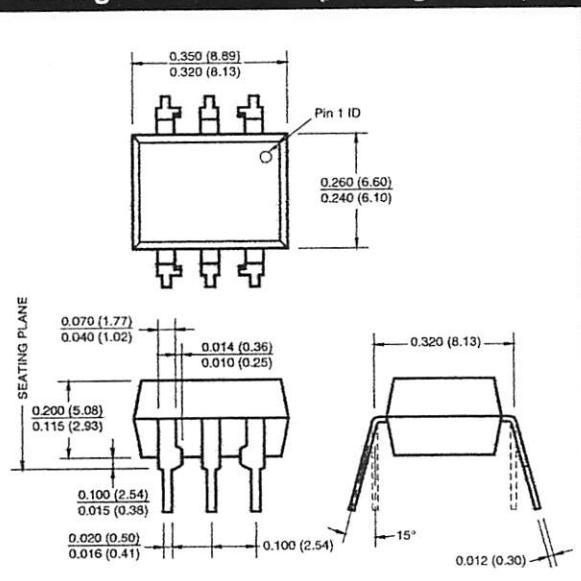
All dimensions are in inches (millimeters)

# GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPERS

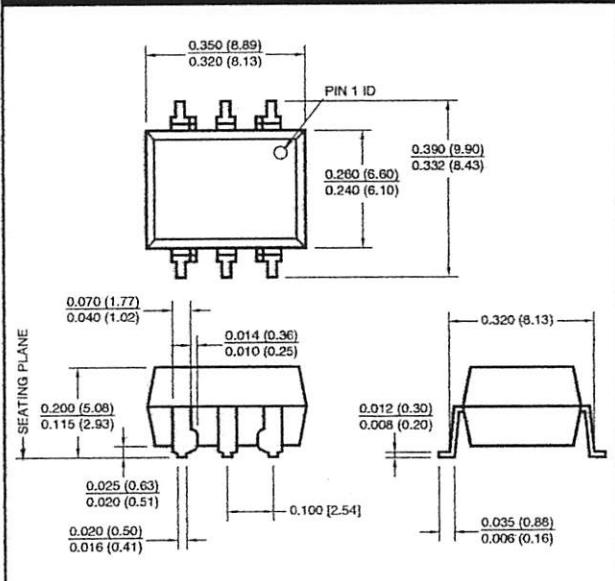
<b>4N25</b>	<b>4N26</b>	<b>4N27</b>	<b>4N28</b>	<b>4N35</b>	<b>4N36</b>
<b>4N37</b>	<b>H11A1</b>	<b>H11A2</b>	<b>H11A3</b>	<b>H11A4</b>	<b>H11A5</b>

## White Package (-M Suffix)

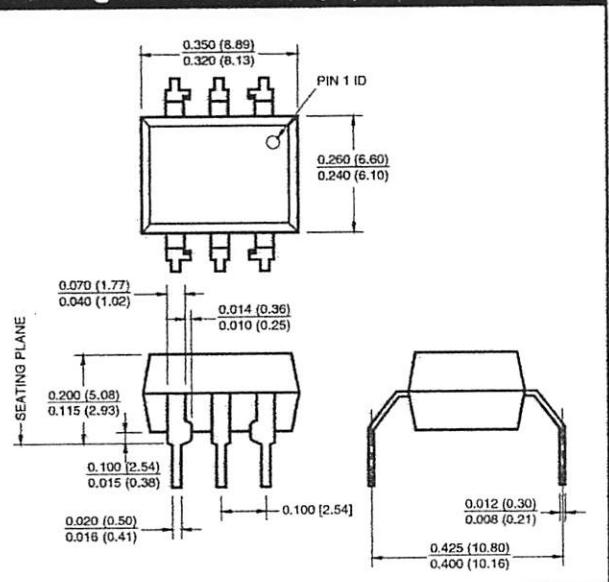
### Package Dimensions (Through Hole)



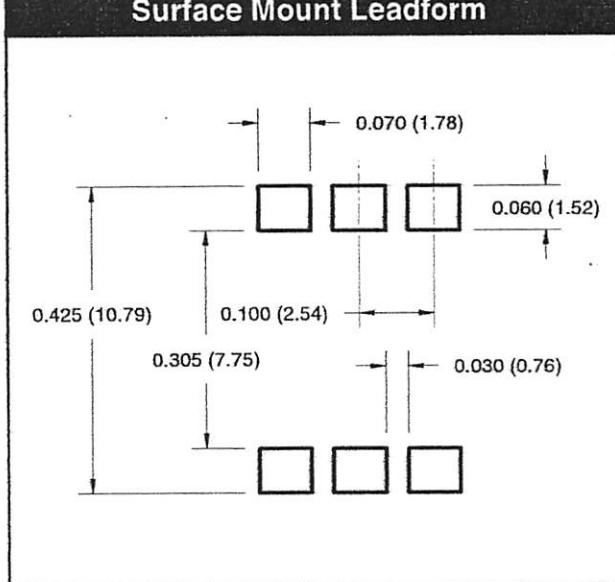
### Package Dimensions (Surface Mount)



### Package Dimensions (0.4" Lead Spacing)



### Recommended Pad Layout for Surface Mount Leadform



### NOTE

All dimensions are in inches (millimeters)

# GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPERS

4N25  
4N37

4N26  
H11A1

4N27  
H11A2

4N28  
H11A3

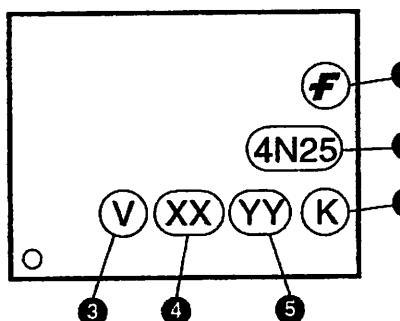
4N35  
H11A4

4N36  
H11A5

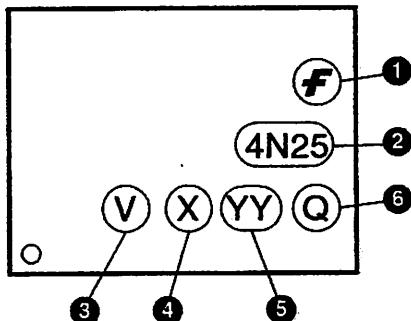
## ORDERING INFORMATION

Order Entry Identifier		
Black Package (No Suffix)	White Package (-M Suffix)	Option
.S	S	Surface Mount Lead Bend
.SD	SR2	Surface Mount; Tape and reel
.W	T	0.4" Lead Spacing
.300	V	VDE 0884
.300W	TV	VDE 0884, 0.4" Lead Spacing
.3S	SV	VDE 0884, Surface Mount
.3SD	SR2V	VDE 0884, Surface Mount, Tape & Reel

## MARKING INFORMATION



Black Package, No Suffix



White Package, -M Suffix

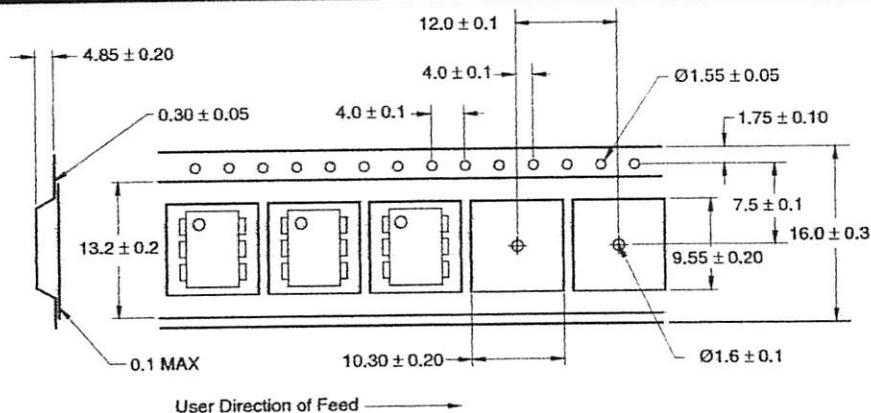
Definitions	
1	Fairchild logo
2	Device number
3	VDE mark (Note: Only appears on parts ordered with VDE option – See order entry table)
4	One or two digit year code <ul style="list-style-type: none"> <li>• Two digits for black package parts, e.g., '03'</li> <li>• One digit for white package parts, e.g., '3'</li> </ul>
5	Two digit work week ranging from '01' to '53'
6	Assembly package code

\*Note – Parts built in the white package (M suffix) that do not have the 'V' option (see definition 3 above) that are marked with date code '325' or earlier are marked in the portrait format.

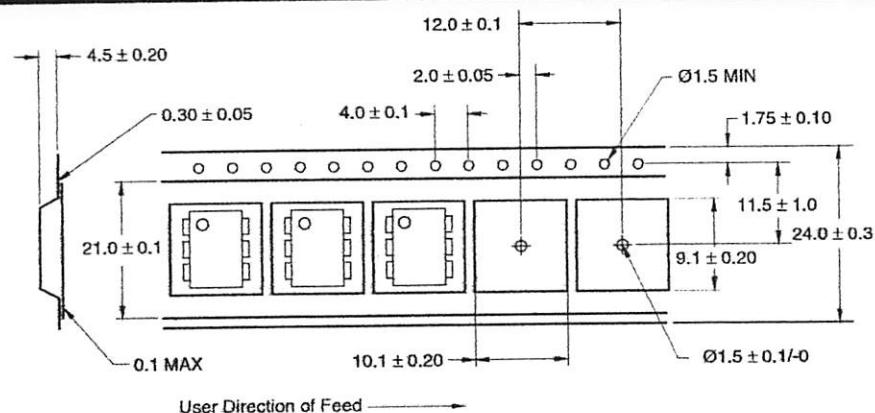
# GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPERS

<b>4N25</b>	<b>4N26</b>	<b>4N27</b>	<b>4N28</b>	<b>4N35</b>	<b>4N36</b>
<b>4N37</b>	<b>H11A1</b>	<b>H11A2</b>	<b>H11A3</b>	<b>H11A4</b>	<b>H11A5</b>

## QT Carrier Tape Specifications (Black Package, No Suffix)



## QT Carrier Tape Specifications (White Package, -M Suffix)



## GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPERS

4N25  
4N37

4N26  
H11A1

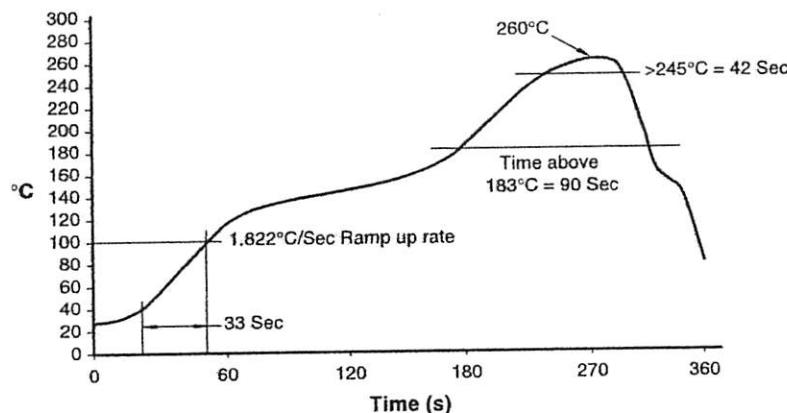
4N27  
H11A2

4N28  
H11A3

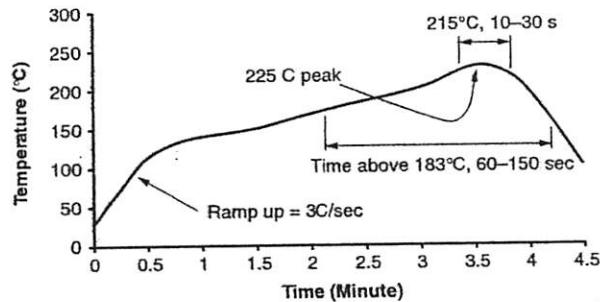
4N35  
H11A4

4N36  
H11A5

### Reflow Profile (White Package, -M Suffix)



### Reflow Profile (Black Package, No Suffix)



- Peak reflow temperature: 225°C (package surface temperature)
- Time of temperature higher than 183°C for 60–150 seconds
- One time soldering reflow is recommended

## N-channel enhancement mode TrenchMOS™ transistor

IRFZ44N

### GENERAL DESCRIPTION

N-channel enhancement mode standard level field-effect power transistor in a plastic envelope using 'trench' technology. The device features very low on-state resistance and has integral zener diodes giving ESD protection up to 2kV. It is intended for use in switched mode power supplies and general purpose switching applications.

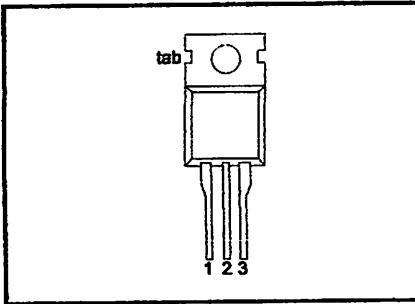
### QUICK REFERENCE DATA

SYMBOL	PARAMETER	MAX.	UNIT
$V_{DS}$	Drain-source voltage	55	V
$I_D$	Drain current (DC)	49	A
$P_{tot}$	Total power dissipation	110	W
$T_j$	Junction temperature	175	°C
$R_{DS(ON)}$	Drain-source on-state resistance $V_{GS} = 10\text{ V}$	22	$\text{m}\Omega$

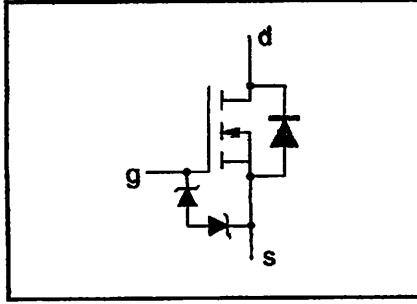
### PINNING - TO220AB

PIN	DESCRIPTION
1	gate
2	drain
3	source
tab	drain

### PIN CONFIGURATION



### SYMBOL



### LIMITING VALUES

Limiting values in accordance with the Absolute Maximum System (IEC 134)

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
$V_{DS}$	Drain-source voltage	-	-	55	V
$V_{DGR}$	Drain-gate voltage	$R_{GS} = 20\text{ k}\Omega$	-	55	V
$\pm V_{GS}$	Gate-source voltage	-	-	20	V
$I_D$	Drain current (DC)	$T_{mb} = 25\text{ }^\circ\text{C}$	-	49	A
$I_D$	Drain current (DC)	$T_{mb} = 100\text{ }^\circ\text{C}$	-	35	A
$I_{DM}$	Drain current (pulse peak value)	$T_{mb} = 25\text{ }^\circ\text{C}$	-	160	A
$P_{tot}$	Total power dissipation	$T_{mb} = 25\text{ }^\circ\text{C}$	-	110	W
$T_{sg}, T_j$	Storage & operating temperature	$T_{mb} = 25\text{ }^\circ\text{C}$	-55	175	$^\circ\text{C}$

### ESD LIMITING VALUE

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
$V_C$	Electrostatic discharge capacitor voltage, all pins	Human body model (100 pF, 1.5 kΩ)	-	2	kV

### THERMAL RESISTANCES

SYMBOL	PARAMETER	CONDITIONS	TYP.	MAX.	UNIT
$R_{th,j-mb}$	Thermal resistance junction to mounting base	-	-	1.4	K/W
$R_{th,j-a}$	Thermal resistance junction to ambient	in free air	60	-	K/W

**N-channel enhancement mode  
TrenchMOS™ transistor**
**IRFZ44N****STATIC CHARACTERISTICS** $T_J = 25^\circ\text{C}$  unless otherwise specified

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
$V_{(BR)DSS}$	Drain-source breakdown voltage	$V_{GS} = 0 \text{ V}; I_D = 0.25 \text{ mA}; T_J = -55^\circ\text{C}$	55	-	-	V
$V_{GS(TO)}$	Gate threshold voltage	$V_{DS} = V_{GS}; I_D = 1 \text{ mA}; T_J = 175^\circ\text{C}$	50 2.0 1.0	3.0	4.0	V
$I_{DSS}$	Zero gate voltage drain current	$V_{DS} = 55 \text{ V}; V_{GS} = 0 \text{ V}; T_J = -55^\circ\text{C}$	-	-	4.4	$\mu\text{A}$
$I_{GSS}$	Gate source leakage current	$V_{GS} = \pm 10 \text{ V}; V_{DS} = 0 \text{ V}; T_J = 175^\circ\text{C}$	-	0.05	10	$\mu\text{A}$
$\pm V_{(BR)GSS}$	Gate source breakdown voltage	$I_G = \pm 1 \text{ mA}; T_J = 175^\circ\text{C}$	16	-	-	V
$R_{DS(ON)}$	Drain-source on-state resistance	$V_{GS} = 10 \text{ V}; I_D = 25 \text{ A}; T_J = 175^\circ\text{C}$	-	15	22	$\text{m}\Omega$
			-	-	42	$\text{m}\Omega$

**DYNAMIC CHARACTERISTICS** $T_{mb} = 25^\circ\text{C}$  unless otherwise specified

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
$g_{fs}$	Forward transconductance	$V_{DS} = 25 \text{ V}; I_D = 25 \text{ A}$	6	-	-	S
$C_{iss}$	Input capacitance	$V_{GS} = 0 \text{ V}; V_{DS} = 25 \text{ V}; f = 1 \text{ MHz}$	-	1350	1800	pF
$C_{oss}$	Output capacitance		-	330	400	pF
$C_{rss}$	Feedback capacitance		-	155	215	pF
$Q_g$	Total gate charge	$V_{DD} = 44 \text{ V}; I_D = 50 \text{ A}; V_{GS} = 10 \text{ V}$	-	-	62	nC
$Q_{gs}$	Gate-source charge		-	-	15	nC
$Q_{gd}$	Gate-drain (Miller) charge		-	-	26	nC
$t_{d(on)}$	Turn-on delay time	$V_{DD} = 30 \text{ V}; I_D = 25 \text{ A}; V_{GS} = 10 \text{ V}; R_G = 10 \Omega$	-	18	26	ns
$t_r$	Turn-on rise time		-	50	75	ns
$t_{d(off)}$	Turn-off delay time	Resistive load	-	40	50	ns
$t_f$	Turn-off fall time		-	30	40	ns
$L_d$	Internal drain inductance	Measured from contact screw on tab to centre of die	-	3.5	-	nH
$L_d$	Internal drain inductance	Measured from drain lead 6 mm from package to centre of die	-	4.5	-	nH
$L_s$	Internal source inductance	Measured from source lead 6 mm from package to source bond pad	-	7.5	-	nH

**REVERSE DIODE LIMITING VALUES AND CHARACTERISTICS** $T_J = 25^\circ\text{C}$  unless otherwise specified

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
$I_{DR}$	Continuous reverse drain current		-	-	49	A
$I_{DRM}$	Pulsed reverse drain current		-	-	160	A
$V_{SD}$	Diode forward voltage	$I_F = 25 \text{ A}; V_{GS} = 0 \text{ V}$	-	0.95	1.2	V
		$I_F = 40 \text{ A}; V_{GS} = 0 \text{ V}$	-	1.0	-	
$t_{rr}$	Reverse recovery time	$I_F = 40 \text{ A}; -di_F/dt = 100 \text{ A}/\mu\text{s}; V_{GS} = -10 \text{ V}; V_R = 30 \text{ V}$	-	47	-	ns
$Q_{rr}$	Reverse recovery charge		-	0.15	-	$\mu\text{C}$

## N-channel enhancement mode TrenchMOS™ transistor

IRFZ44N

### AVALANCHE LIMITING VALUE

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
$W_{DSS}$	Drain-source non-repetitive unclamped inductive turn-off energy	$I_D = 45 \text{ A}$ ; $V_{DD} \leq 25 \text{ V}$ ; $V_{GS} = 10 \text{ V}$ ; $R_{GS} = 50 \Omega$ ; $T_{mb} = 25^\circ\text{C}$	-	-	110	mJ

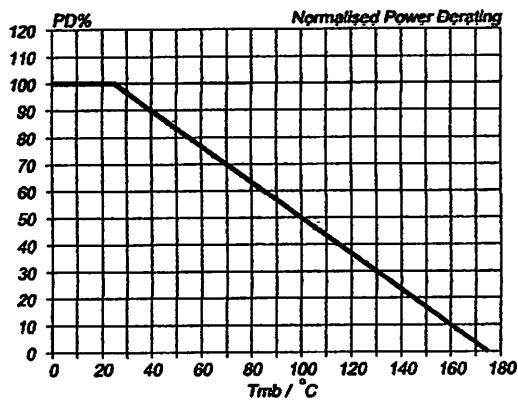


Fig. 1. Normalised power dissipation.  
 $PD\% = 100 \cdot P_D / P_{D, 25^\circ\text{C}} = f(T_{mb})$

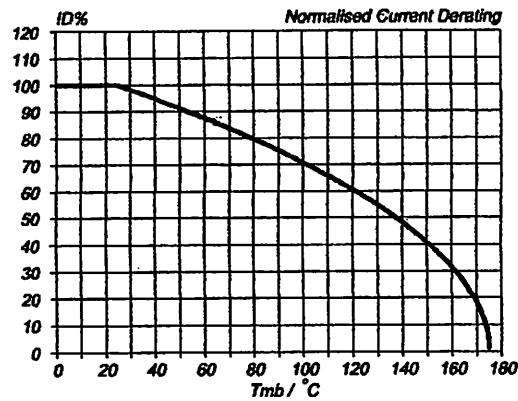
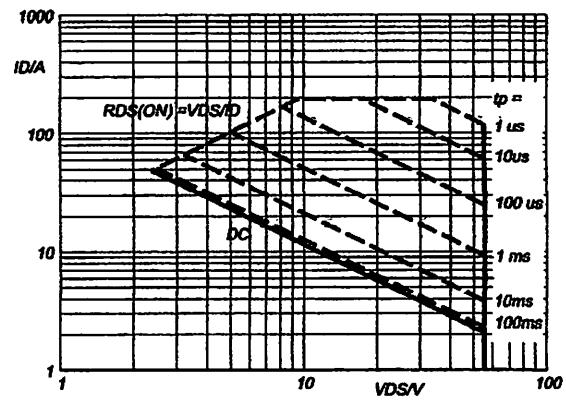


Fig. 2. Normalised continuous drain current.  
 $ID\% = 100 \cdot I_D / I_{D, 25^\circ\text{C}} = f(T_{mb})$ ; conditions:  $V_{GS} \geq 10 \text{ V}$

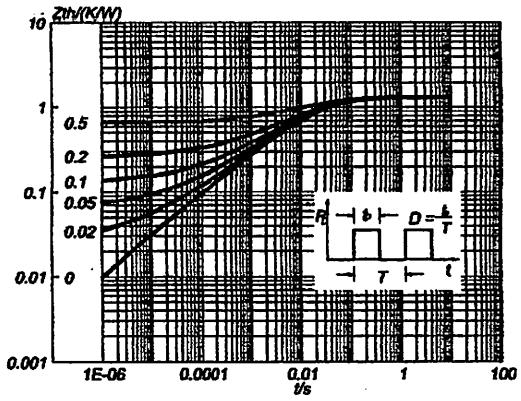


Fig. 4. Transient thermal impedance.  
 $Z_{th,mb} = f(t)$ ; parameter  $D = t_p/T$

## N-channel enhancement mode TrenchMOS™ transistor

IRFZ44N

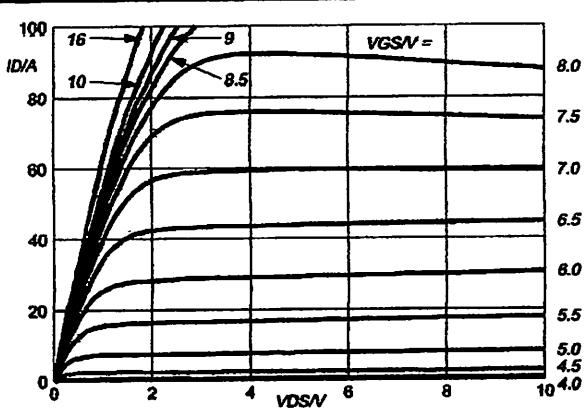


Fig.5. Typical output characteristics,  $T_J = 25^\circ\text{C}$ .  
 $I_D = f(V_{DS})$ ; parameter  $V_{GS}$

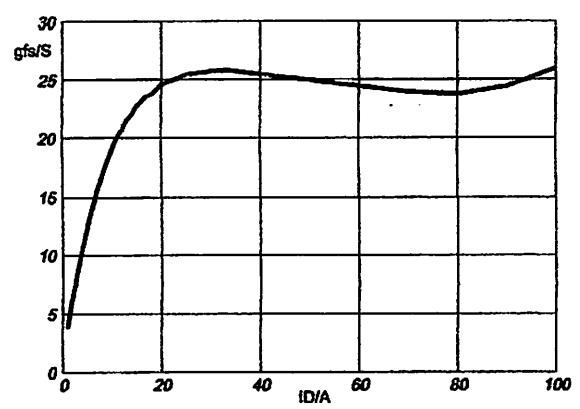


Fig.8. Typical transconductance,  $T_J = 25^\circ\text{C}$ .  
 $g_{fs} = f(I_D)$ ; conditions:  $V_{DS} = 25\text{ V}$

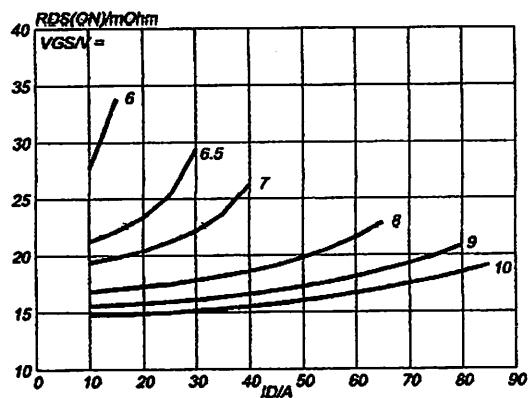


Fig.6. Typical on-state resistance,  $T_J = 25^\circ\text{C}$ .  
 $R_{DS(ON)} = f(I_D)$ ; parameter  $V_{GS}$

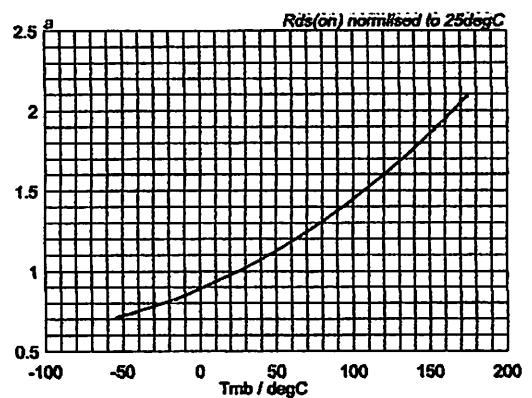


Fig.9. Normalised drain-source on-state resistance.  
 $a = R_{DS(ON)}/R_{DS(ON)25^\circ\text{C}} = f(T_J)$ ;  $I_D = 25\text{ A}$ ;  $V_{GS} = 10\text{ V}$

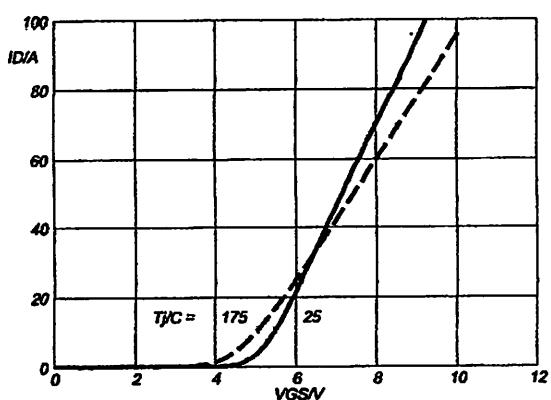


Fig.7. Typical transfer characteristics.  
 $I_D = f(V_{GS})$ ; conditions:  $V_{DS} = 25\text{ V}$ ; parameter  $T_J$

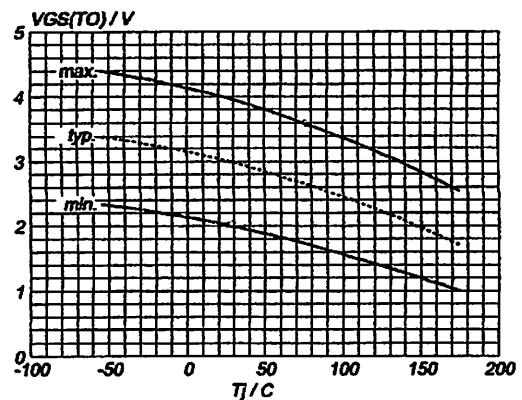


Fig.10. Gate threshold voltage.  
 $V_{GS(TO)} = f(T_J)$ ; conditions:  $I_D = 1\text{ mA}$ ;  $V_{DS} = V_{GS}$

## N-channel enhancement mode TrenchMOS™ transistor

IRFZ44N

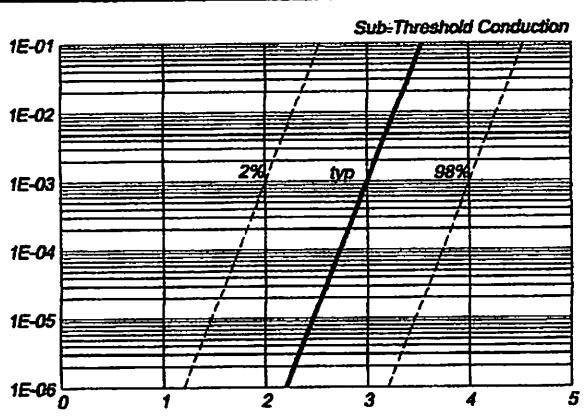


Fig. 11. Sub-threshold drain current.  
 $I_D = f(V_{GS})$ ; conditions:  $T_J = 25^\circ\text{C}$ ;  $V_{DS} = V_{GS}$

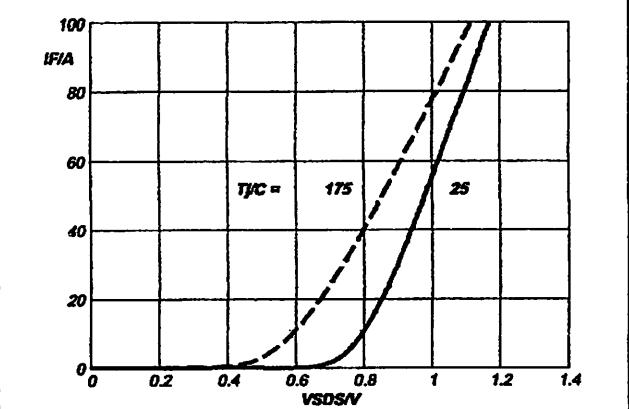


Fig. 14. Typical reverse diode current.  
 $I_F = f(V_{DS})$ ; conditions:  $V_{GS} = 0\text{ V}$ ; parameter  $T_J$

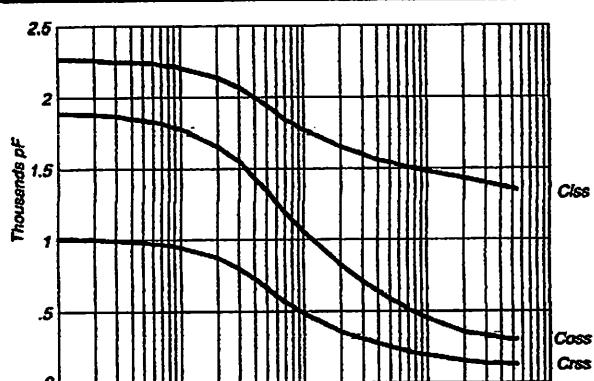


Fig. 12. Typical capacitances,  $C_{iss}$ ,  $C_{oss}$ ,  $C_{rss}$ .  
 $C = f(V_{DS})$ ; conditions:  $V_{GS} = 0\text{ V}$ ;  $f = 1\text{ MHz}$

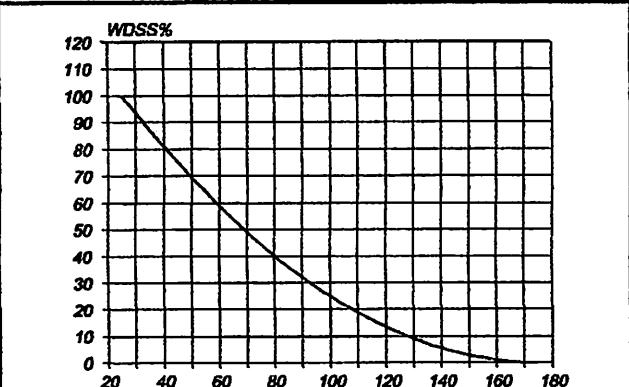


Fig. 15. Normalised avalanche energy rating.  
 $W_{DSS}\% = f(T_{mb})$ ; conditions:  $I_D = 49\text{ A}$

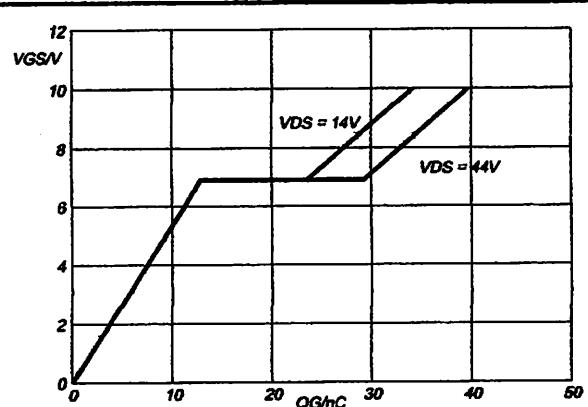


Fig. 13. Typical turn-on gate-charge characteristics.  
 $V_{GS} = f(Q_G)$ ; conditions:  $I_D = 50\text{ A}$ ; parameter  $V_{DS}$

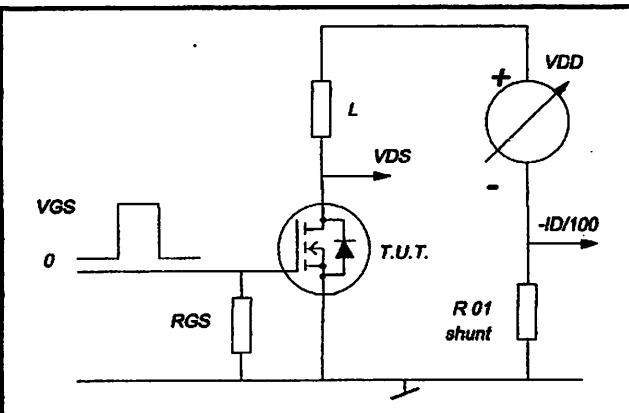


Fig. 16. Avalanche energy test circuit.  
 $W_{DSS} = 0.5 \cdot L I_D^2 \cdot BV_{DSS} / (BV_{DSS} - V_{DD})$

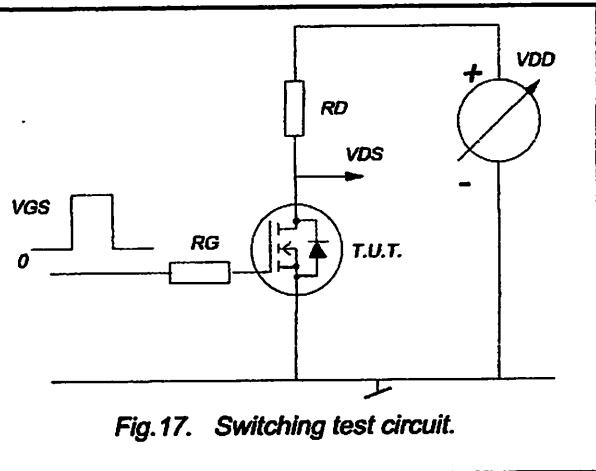
**N-channel enhancement mode  
TrenchMOS™ transistor****IRFZ44N**

Fig.17. Switching test circuit.

**N-channel enhancement mode  
TrenchMOS™ transistor**

**IRFZ44N**

**MECHANICAL DATA**

*Dimensions in mm*

Net Mass: 2 g

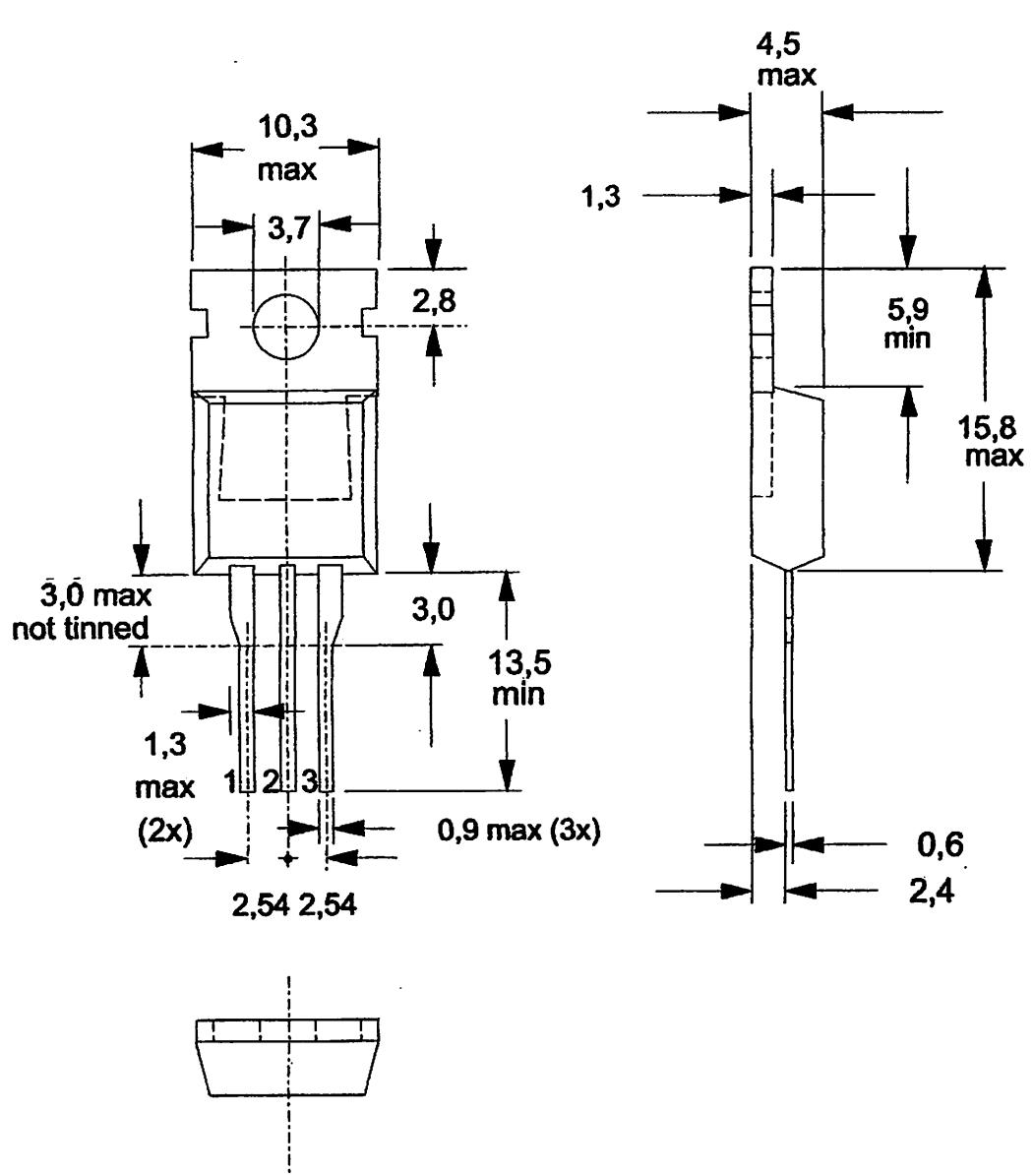


Fig.18. SOT78 (TO220AB); pin 2 connected to mounting base.

**Notes**

1. Observe the general handling precautions for electrostatic-discharge sensitive devices (ESDs) to prevent damage to MOS gate oxide.
2. Refer to mounting instructions for SOT78 (TO220) envelopes.
3. Epoxy meets UL94 V0 at 1/8".

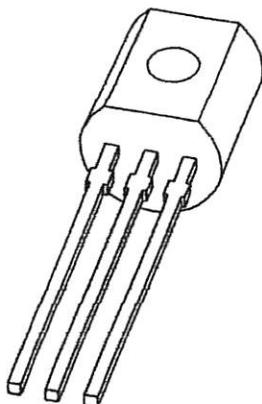
**N-channel enhancement mode  
TrenchMOS™ transistor****IRFZ44N****DEFINITIONS**

<b>Data sheet status</b>	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
<b>Limiting values</b>	
Limiting values are given in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of this specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
<b>Application information</b>	
Where application information is given, it is advisory and does not form part of the specification.	
<b>© Philips Electronics N.V. 1999</b>	
All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner.	
The information presented in this document does not form part of any quotation or contract, it is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent or other industrial or intellectual property rights.	

**LIFE SUPPORT APPLICATIONS**

These products are not designed for use in life support appliances, devices or systems where malfunction of these products can be reasonably expected to result in personal injury. Philips customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

# DATA SHEET



## **BC556; BC557** PNP general purpose transistors

Product specification  
Supersedes data of 1997 Mar 27

1999 Apr 15

**PNP general purpose transistors****BC556; BC557****FEATURES**

- Low current (max. 100 mA)
- Low voltage (max. 65 V).

**APPLICATIONS**

- General purpose switching and amplification.

**DESCRIPTION**

PNP transistor in a TO-92; SOT54 plastic package.  
NPN complements: BC546 and BC547.

**PINNING**

PIN	DESCRIPTION
1	emitter
2	base
3	collector

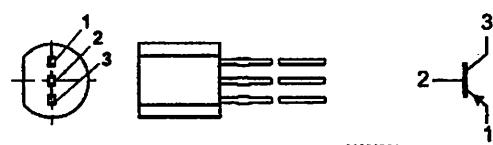


Fig.1 Simplified outline (TO-92; SOT54) and symbol.

**LIMITING VALUES**

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
$V_{CEO}$	collector-base voltage BC556 BC557	open emitter	-	-80 -50	V V
$V_{CEO}$	collector-emitter voltage BC556 BC557	open base	-	-65 -45	V V
$V_{EB0}$	emitter-base voltage	open collector	-	-5	V
$I_C$	collector current (DC)		-	-100	mA
$I_{CM}$	peak collector current		-	-200	mA
$I_{BM}$	peak base current		-	-200	mA
$P_{tot}$	total power dissipation	$T_{amb} \leq 25^\circ\text{C}$	-	500	mW
$T_{stg}$	storage temperature		-65	+150	°C
$T_j$	junction temperature		-	150	°C
$T_{amb}$	operating ambient temperature		-65	+150	°C

## PNP general purpose transistors

BC556; BC557

## THERMAL CHARACTERISTICS

SYMBOL	PARAMETER	CONDITIONS	VALUE	UNIT
$R_{th,Ja}$	thermal resistance from junction to ambient	note 1	250	K/W

## Note

1. Transistor mounted on an FR4 printed-circuit board.

## CHARACTERISTICS

 $T_j = 25^\circ\text{C}$  unless otherwise specified.

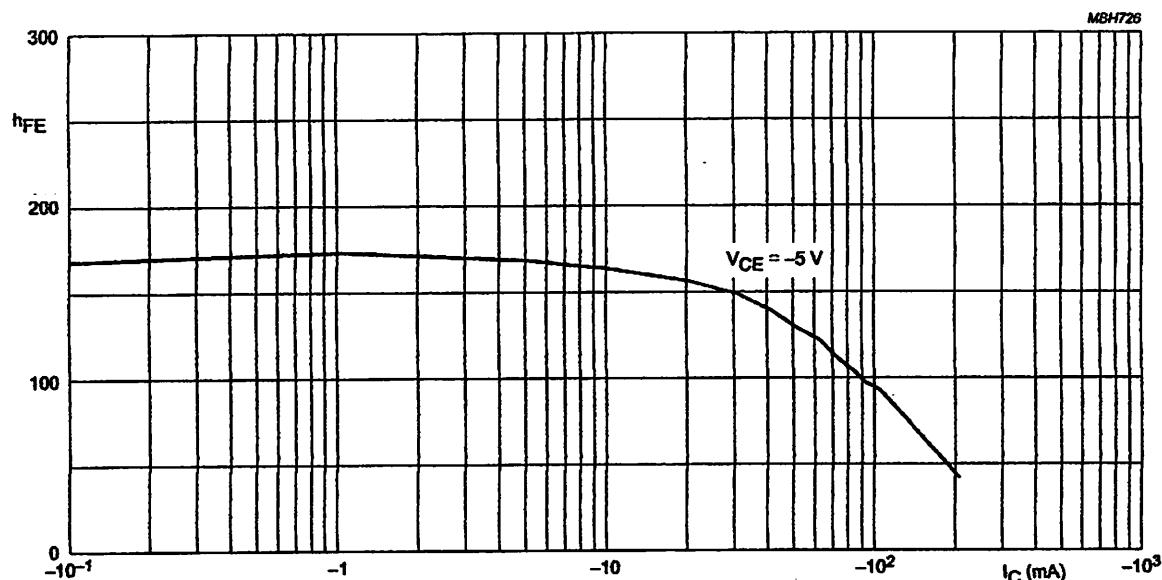
SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
$I_{CBO}$	collector cut-off current	$I_E = 0; V_{CB} = -30 \text{ V}$	-	-1	-15	nA
		$I_E = 0; V_{CB} = -30 \text{ V}; T_j = 150^\circ\text{C}$	-	-	-4	µA
$I_{EBO}$	emitter cut-off current	$I_C = 0; V_{EB} = -5 \text{ V}$	-	-	-100	nA
$h_{FE}$	DC current gain BC556 BC557 BC556A BC556B; BC557B BC557C	$I_C = -2 \text{ mA}; V_{CE} = -5 \text{ V};$ see Figs 2, 3 and 4	125	-	475	
			125	-	800	
			125	-	250	
			220	-	475	
			420	-	800	
$V_{CEsat}$	collector-emitter saturation voltage	$I_C = -10 \text{ mA}; I_B = -0.5 \text{ mA}$	-	-60	-300	mV
		$I_C = -100 \text{ mA}; I_B = -5 \text{ mA}$	-	-180	-650	mV
$V_{BEsat}$	base-emitter saturation voltage	$I_C = -10 \text{ mA}; I_B = -0.5 \text{ mA};$ note 1	-	-750	-	mV
		$I_C = -100 \text{ mA}; I_B = -5 \text{ mA};$ note 1	-	-930	-	mV
$V_{BE}$	base-emitter voltage	$I_C = -2 \text{ mA}; V_{CE} = -5 \text{ V};$ note 2	-600	-650	-750	mV
		$I_C = -10 \text{ mA}; V_{CE} = -5 \text{ V};$ note 2	-	-	-820	mV
$C_c$	collector capacitance	$I_E = i_e = 0; V_{CB} = -10 \text{ V}; f = 1 \text{ MHz}$	-	3	-	pF
$C_e$	emitter capacitance	$I_C = i_e = 0; V_{EB} = -0.5 \text{ V}; f = 1 \text{ MHz}$	-	10	-	pF
$f_T$	transition frequency	$I_C = -10 \text{ mA}; V_{CE} = -5 \text{ V}; f = 100 \text{ MHz}$	100	-	-	MHz
$F$	noise figure	$I_C = -200 \mu\text{A}; V_{CE} = -5 \text{ V}; R_S = 2 \text{ k}\Omega;$ $f = 1 \text{ kHz}; B = 200 \text{ Hz}$	-	2	10	dB

## Notes

1.  $V_{BEsat}$  decreases by about  $-1.7 \text{ mV/K}$  with increasing temperature.
2.  $V_{BE}$  decreases by about  $-2 \text{ mV/K}$  with increasing temperature.

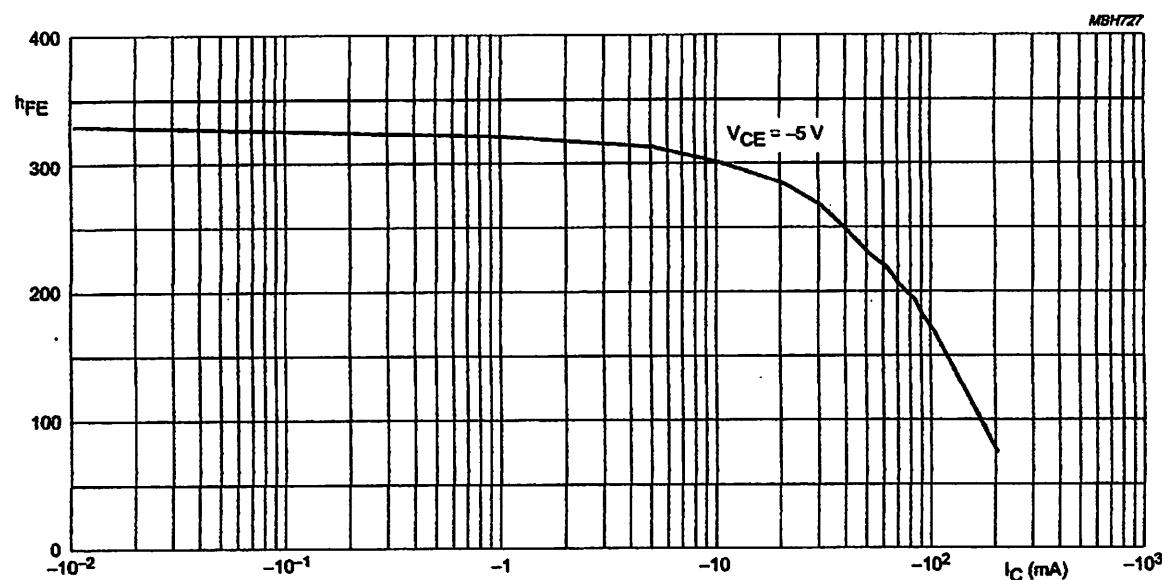
## PNP general purpose transistors

BC556; BC557



BC556A.

Fig.2 DC current gain; typical values.

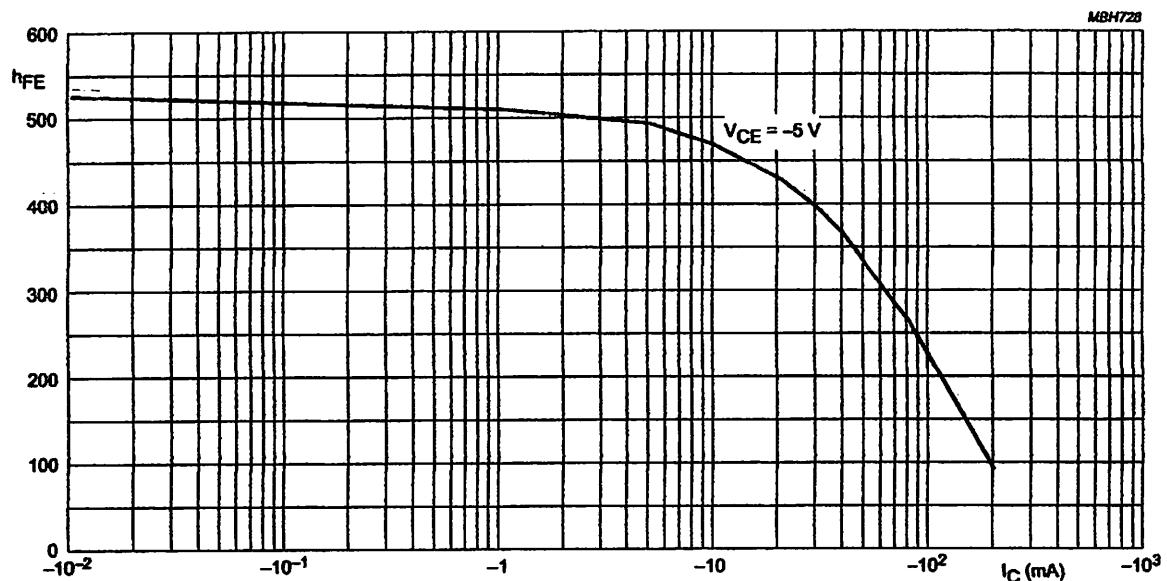


BC556B; BC557B.

Fig.3 DC current gain; typical values.

## PNP general purpose transistors

BC556; BC557



BC557C.

Fig.4 DC current gain; typical values.

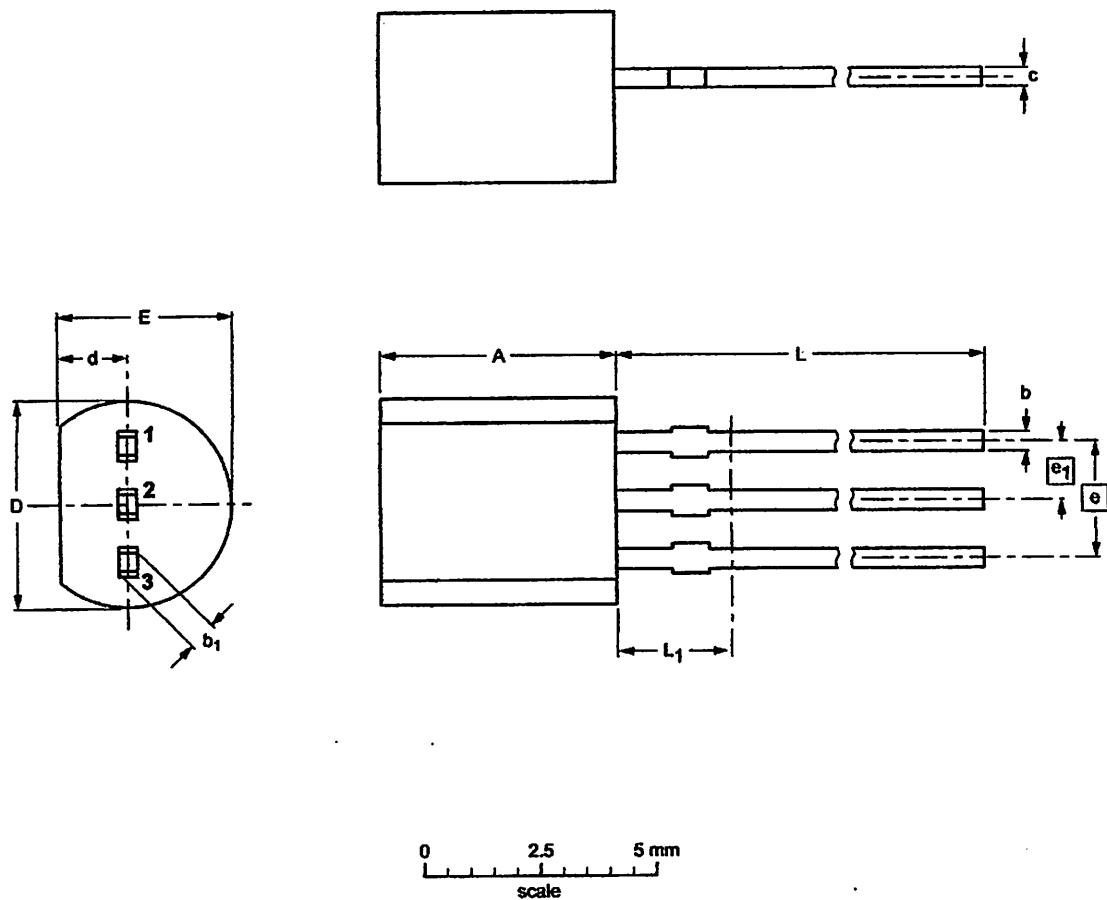
## PNP general purpose transistors

BC556; BC557

## PACKAGE OUTLINE

Plastic single-ended leaded (through hole) package; 3 leads

SOT54



## DIMENSIONS (mm are the original dimensions)

UNIT	A	b	b <sub>1</sub>	c	D	d	E	e	e <sub>1</sub>	L	L <sub>1</sub> <sup>(1)</sup>
mm	5.2	0.48	0.66	0.45	4.8	1.7	4.2	2.54	1.27	14.5	
	5.0	0.40	0.56	0.40	4.4	1.4	3.6			12.7	2.5

## Note

1. Terminal dimensions within this zone are uncontrolled to allow for flow of plastic and terminal irregularities.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT54		TO-92	SC-43			97-02-28

**PNP general purpose transistors****BC556; BC557****DEFINITIONS**

<b>Data sheet status</b>	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
<b>Limiting values</b>	
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
<b>Application information</b>	
Where application information is given, it is advisory and does not form part of the specification.	

**LIFE SUPPORT APPLICATIONS**

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.