UNIVERSITAS INDONESIA

PAPAN INFORMASI ELEKTRONIK DENGAN PS2 *KEYBOARD*

TUGAS AKHIR

EVY CHRISTANTO SRI NUGROHO
07 06 19 9294

FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI, 2010

UNIVERSITAS INDONESIA

PAPAN INFORMASI ELEKTRONIK DENGAN PS2 *KEYBOARD*

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

EVY CHRISTANTO SRI NUGROHO
07 06 19 9294

FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI, 2010

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya saya sendiri,
Dan semua sumber baik yang dikutip maupun dirujuk
Telah saya nyatakan dengan benar.

Nama	**: Evy Christanto Sri Nugroho**
NPM	**: 0706199294**
Tanda Tangan	**:**
Tanggal	**: 23 Juni 2010**

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh:

Nama	: Evy Christanto Sri Nugroho
NPM	: 0706199294
Program Studi	: Strata 1 Ekstensi
Judul Tugas Akhir	: Papan Informasi Elektronik dengan PS2 *Keyboard*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian pernyataan untuk memperoleh gelar Sarjana Teknik pada program studi strata 1 ekstensi, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing	: Dr. Ir. Retno Wigajatri P, MS	(.....)
Penguji	: Dr. Abdul Muis ST, M. Eng	(.....)
Penguji	: Dr. Ir. Dodi Sudiana M. Eng	(.....)
Ditetapkan di	: Universitas Indonesia, Depok	
Tanggal	: 23 Juni 2010	

iii Universitas Indonesia

KATA PENGANTAR / UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan tugas akhir ini. Penulisan tugas akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Elektro pada Fakultas Teknik Universitas Indonesia.. Oleh karena itu, saya mengucapkan terima kasih kepada:

(1) Dr. Ir. Retno Wigajatri P.MS selaku dosen pembimbing yang telah menyediakan waktu, tenaga, ide dan pikiran untuk mengarahkan saya dalam penyusunan tugas akhir ini.

(2) Kedua orangtua tercinta yang telah memberi dukungan dengan kasih sayang, Citra Huseni, Amd yang dengan sabar selalu mendoakan dan memberi semangat, serta Junior Ompusunggu, ST atas bimbingannya selama mengerjakan tugas akhir ini.

(3) Bapak panggung dan Bapak erisman, S.Si yang memberikan keleluasan waktu selama ini di PT Sanken Indonesia, serta Analisis dan repair group yang telah membantu tugas akhir ini sampai akhir.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga Tugas Akhir ini membawa manfaat bagi pengembangan ilmu.

Depok, 23 Juni 2010

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

==========

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini :

Nama	: Evy Christanto Sri Nugroho
NPM	: 0706199294
Program Studi	: S1 – Ekstensi
Departemen	: Teknik Elektro
Fakultas	: Teknik
Jenis karya	: Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty - Free Right*)** atas karya ilmiah saya yang berjudul :

Papan Informasi Elektronik dengan PS2 *Keyboard*

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalih media / formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis / pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 23 Juni 2010
Yang menyatakan

(Evy Christanto Sri Nugroho)

ABSTRAK

Nama : Evy Christanto Sri Nugroho
Program Studi : S1 - Ekstensi
Judul : Papan Informasi Elektronik dengan PS2 *Keyboard*

Informasi tentang model yang sedang berada di *line* produksi PT Sanken Indonesia sangat dibutuhkan untuk mengoptimalkan kinerja personil bagian kontrol kualitas. Pada tugas akhir ini dilakukan rancang bangun papan informasi elektronik yang sesuai dengan kebutuhan dan memenuhi keterbatasan PT Sanken Indonesia. Papan informasi elektronik terdiri dari ATMega8535 sebagai pengolah dan peyimpan data, LCD untuk tampilan informasi tulisan bagi operator, power supply, *dot matrix display* serta melibatkan PS2 *keyboard* untuk memasukkan data. D*ot matrix* sebagai perangkat *display* dapat menampilkan tulisan bergeser dengan kriteria kecepatan looping 3.06s dan setiap karakter dapat dimunculkan dengan selang waktu sekitar 95ms. Perangkat ini juga mampu menyimpan data kendati terjadi hubungan putus listrik.

Kata kunci:
Papan Informasi Elektronik, *Keyboard*, *dot matrix*, ATMega8535, LCD

ABSTRACT

Name : Evy Christanto Sri Nugroho
Study Program: S1 - Ekstensi
Title : Electronic Information Boards with PS2 *Keyboard*

Information about models which are running on the production line of PT. Sanken Indonesia is needed to optimize the performance of quality control personnel. In this project an electronic information board design for special purpose that suits the needs and meet the limitations of PT. Sanken Indonesia is conducted. Electronic information board is consist of ATMega8535 as data processing and storage, LCD to display text information for operators, power supply, dot matrix display and involves PS2 keyboard to enter data. Dot matrix display device can display the article shifts with speed loops 3,06s and each character can be raised at intervals of about 95ms. This device is also capable of storing data occurred despite broken electrical connection.

Keywords:
Electronic Information board, Keyboard, dot matrix, ATMega8535, LCD

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

PT. Sanken Indonesia adalah perusahaan *manufacture* yang bergerak di bidang pembuatan *power supply switching* atau sering disebut dengan smps (*switching mode power supply*) diantaranya, *low power* (adaptor untuk laptop, *UPS*), consumer produk atau biasa disebut dengan *CP* (televisi, radio, dvd, walkman, dll), dan *office automation* (untuk mesin *photocopy*, *printer*, komputer, dll).

Semua jenis *power supply* tersebut dibuat secara massal dan dikerjakan oleh operator yang berada dalam suatu *line* produksi. Di PT Sanken Indonesia, terdapat 25 *line* produksi yang masing-masing membuat *power supply* lebih dari satu model. Rata-rata, dalam satu *line* dapat dibuat 6-15 model *power supply* selama 8 jam kerja secara kontinyu. Akibatnya, proses pergantian model jadi sangat cepat.

Namun, dengan banyaknya model yang dibuat dan target produksi yang cukup tinggi, PT Sanken Indonesia juga dituntut untuk kualitas hasil produksi. Oleh karena itu, untuk setiap model senantiasa dilakukan pengecekan oleh staf atau operator bagian kontrol kualitas untuk periode tertentu baik di area produksi, final test elektrik, dan ekspor.

Pada prakteknya, proses pergantian model yang cepat ini menimbulkan masalah yaitu staf ataupun operator bagian kontrol kualitas kesulitan mendeteksi pergantian model yang cepat. Akibatnya produk yang harusnya dilakukan proses selanjutnya terpaksa ditunda beberapa saat untuk keperluan pengecekan kualitas produk. Hal lebih lanjut, akan menghambat jadwal dari bisnis plan yang sudah dibuat.

Walaupun pada masing-masing *line* produksi ada seorang *leader* produksi, namun mereka sangat sibuk mengatur pergantian material dan pengontrolan terhadap operator produksi, sehingga tidak mungkin untuk melaporkan pergantian model ini. Oleh karena itu dibutuhkan akses informasi yang cepat tentang pergantian model yang sedang diproduksi. Gambar 1.1 menunjukkan tidak adanya papan informasi didepan *line* produksi .

Gambar 1.1 *Line* produksi di PT.Sanken Indonesia

Papan informasi elektronik sebenarnya sudah banyak sekali dijual dipasaran, tetapi harga yang dimuat sangat mahal sektar 5-7 juta[11]. Selain itu, penggunaan papan informasi yang ada saat ini masih tergolong rumit dalam pengoperasiannya. Untuk mengubah isi pesan diperlukan sebuah komputer. Cara seperti ini membutuhkan waktu yang lama, kurang efektif dan membutuhkan area yang cukup besar. Area yang tersedia adalah 60cmx45cm. Bila harus menggunakan PC (*personal computer*), artinya harus menyediakan tempat untuk PC, sedangkan area yang ada tidak terlalu besar seperti yang dijelaskan diatas tadi. Disamping itu, daya listrik yang dibutuhkan relatif besar.

Termotivasi oleh masalah tersebut pada skripsi ini dilakukan rancang bangun papan informasi elektronik yang isi pesannya dapat diinput dengan ps2 *keyboard* dan didesain khusus untuk area yang terbatas sehingga menjadi lebih ekonomis. Alat ini diharapkan mampu menampilkan pesan pada dotmatrik LED dan LCD sesuai dengan pesan yang diinput dari *keyboard* serta mampu menampilkan pesan berjalan sehingga dapat terlihat menarik bagi orang yang membacanya.

1.2 Tujuan Penulisan

Untuk mengatasi masalah yang telah disebutkan pada sub bab 1.1 maka pada tugas akhir ini dilakukan rancang bangun alat papan informasi elektronik yang disambungkan dengan PS/2 *keyboard.*

1.3 Pembatasan Masalah

Pada tugas akhir ini, rancang bangun papan informasi elektronik dengan PS/2 *keyboard* dibatasi pada :

1. Penggunaan display yang disesuaikan dengan area produksi.
2. Penggunaan PS/2 *keyboard* sebagai input untuk karakter.
3. Penggunaan *software* dengan bahasa *C* .

1.4 Metode Penulisan

Metodologi penyelesaian masalah dalam pembuatan alat dan penyusunan laporan tugas akhir adalah:

1. Perancangan dan realisasi perangkat keras
    1. Penentuan jumlah karakter yang dibutuhkan.
    2. Menentukan ukuran alat yang disesuaikan dengan area produksi.
    3. Membuat program dengan menggunakan *AVR Wizard Codevision*.
2. Realisasi alat

    Setelah mendapat skema yang lebih rinci, mulai untuk menginventaris kebutuhan komponen sebagai perangkat kerasnya (*hardware*) dengan membuat daftar komponen dan kemudian mengadakannya. Selanjutnya, membuat layout PCB dan mencetaknya. Setelah pembuatan *hardware* selesai maka dilanjutkan dengan pembuatan perangkat lunak (*software*) untuk interfacing antar komponen-komponen perangkat kerasnya.
3. Uji coba

    Setelah pembuatan *hardware* & *software* dilakukan uji coba untuk mengetahui apakah alat dapat bekerja sesuai dengan yang diharapkan.

1.5 Sistematika Penulisan

Pendahuluan berisikan tentang latar belakang masalah, perumusan masalah, tujuan, ruang lingkup proyek akhir, metodologi perancangan dan sistematika pembahasan laporan proyek akhir yang menerangkan sekilas tentang isi yang dikandung pada setiap bab dalam buku laporan ini.

BAB I Pendahuluan berisikan tentang latar belakang masalah, perumusan masalah, tujuan, batasan tugas akhir, metodologi perancangan dan sistematika pembahasan tugas akhir yang menerangkan dengan singkat tentang isi yang dikandung pada setiap bab buku ini.

BAB II Landasan Teori berisikan tentang konsep-konsep dasar teori yang berhubungan dengan rancang bangun alat yang dibahas secara ringkas dan jelas, sehingga dapat digunakan sebagai penunjang pembuatan alat tersebut.

BAB III Perancangan dan Realisasi membahas mengenai tahap-tahap perancangan dan realisasi sistem yang dibuat, dan metode yang dipakai dalam alat yang mencakup perancangan *hardware* dan *software*.

BAB IV Pengujian dan Analisa menguraikan tentang pengujian sistem, yang berpengaruh terhadap kinerja sistem. Selain itu dicantumkan pula hasil pengukuran dan analisanya.

BAB V Penutup berisikan kesimpulan dari perancangan alat yang telah direalisasikan berikut saran dalam pengembangan sistem dengan kelebihan dan kekurangannya.

BAB II

LANDASAN TEORI

Sebelum melakukan rancang bangun, dibutuhkan tentang teori yang melandasi dari komponen-komponen rancang bangun seperti yang dijelaskan berikut ini

2.1 Mikrokontroler ATMega8535[1]

Arsitektur mikrokontroler jenis AVR pertama kali dikembangkan pada tahun 1996 oleh dua orang mahasiswa Norwegian Institute of Technology yaitu Alf-Egil Bogen dan Vegard Wollan.

Mikrokontroler AVR kemudian dikembangkan lebih lanjut oleh Atmel. Seri pertama AVR yang dikeluarkan adalah mikrokontroler 8 *bit* AT90S8515, dengan konfigurasi pin yang sama dengan mikrokontroler 8051, termasuk *address* dan data bus yang termultipleksi.

Mikrokontroler AVR menggunakan teknologi RISC dimana set instruksinya dikurangi dari segi ukurannya dan kompleksitas mode pengalamatannya. Pada awal era industri komputer, bahasa pemrograman masih menggunakan kode mesin dan bahasa *assembly*. Untuk mempermudah dalam pemrograman para desainer komputer kemudian mengembangkan bahasa pemrograman tingkat tinggi yang mudah dipahami manusia. Namun akibatnya, instruksi yang ada menjadi semakin komplek dan membutuhkan lebih banyak memori. Akibatnya siklus eksekusi instruksi ini menjadi semakin lama. Dalam AVR dengan arsitektur RISC 8 bit, semua instruksi berukuran 16 *bit* dan sebagian besar dieksekusi dalam 1 siklus *clock*. Berbeda dengan mikrokontroler MCS-51 yang instruksinya bervariasi antara 8 *bit* sampai 32 *bit* dan dieksekusi selama 1 sampai 4 siklus mesin, dimana 1 siklus mesin membutuhkan 12 periode *clock*.

Dalam perkembangannya, AVR dibagi menjadi beberapa varian yaitu AT90Sxx, ATMega, AT86RFxx dan ATTiny.

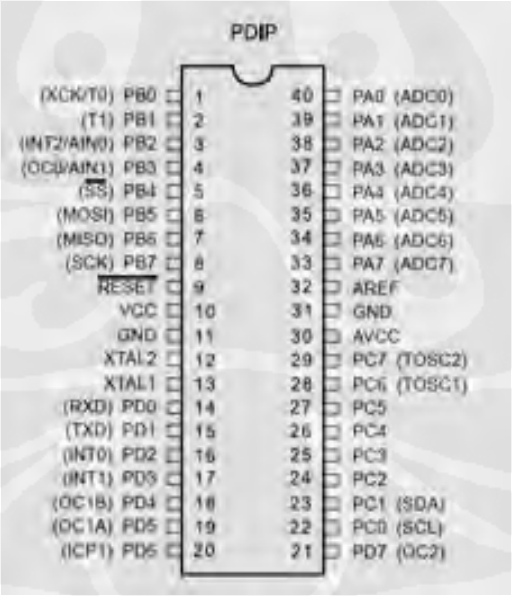
Pada dasarnya yang membedakan masing-masing varian adalah kapasitas memori dan beberapa fitur tambahan saja.

2.1.1 Karakteristik ATMega8535

Fitur yang tersedia pada ATMega8535 adalah sebagai berikut :

- Frekuensi clock maksimum 16 MHz
- Jalur I/O 32 buah, yang terbagi dalam PortA, PortB, PortC dan PortD
- Analog to Digital Converter 10 *bit* sebanyak 8 input
- Timer/Counter sebanyak 3 buah
- CPU 8 *bit* yang terdiri dari 32 *register*
- *Watchdog Timer* dengan osilator internal
- SRAM sebesar 512 byte
- Memori Flash sebesar 8 Kbyte dengan kemampuan *read while write*
- *Interrupt* internal maupun eksternal
- Port komunikasi SPI
- EEPROM sebesar 512 *byte* yang dapat diprogram saat operasi
- *Analog Comparator*
- Komunikasi serial standar USART dengan kecepatan maksimal 2,5 Mbps

Konfigurasi dari pin ATMega8535 dapat dilihat pada Gambar 2.1.



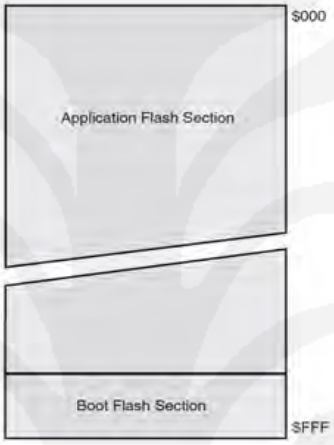
Gambar 2.1 Konfigurasi pin ATMega8535[1]

2.1.2 Peta Memori ATMega8535

ATMega8535 memiliki dua jenis memori yaitu data memory dan program memory ditambah satu fitur tambahan yaitu EEPROM memori untuk penyimpan data

- Program Memori

ATMega8535 memiliki *On-Chip In-System Reprogrammable Flash Memory* untuk menyimpan program. Untuk alasan keamanan, program memory dibagi menjadi dua bagian

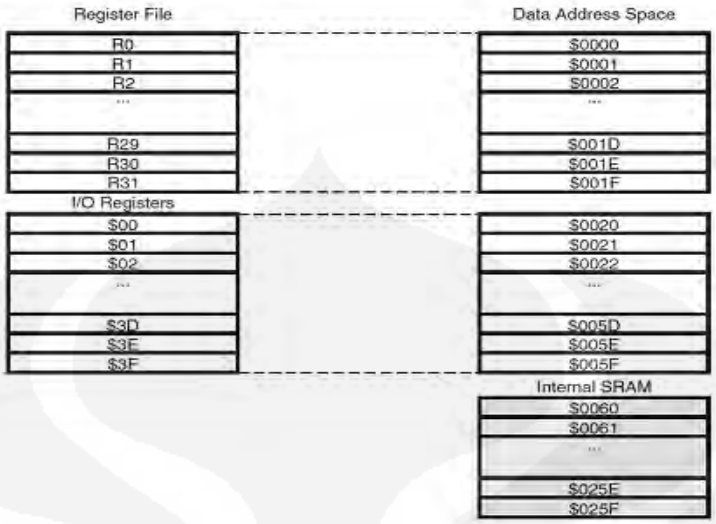


Gambar 2.2 Peta memori ATMega8535[1]

yaitu *Boot Flash Section* dan *Application Flash Section*. *Boot Flash Section* digunakan untuk menyimpan program *Boot Loader*, yaitu program yang harus dijalankan pada saat AVR *reset* atau pertama kali diaktifkan. *Application Flash Section* digunakan untuk menyimpan program aplikasi yang dibuat *user*. AVR tidak dapat menjalankan program aplikasi ini sebelum menjalankan program *Boot Loader*.

- Data Memori

Gambar 2.3 menunjukkan peta memori SRAM pada ATMega8535. Terdapat 608 lokasi *address* data memori. 96 lokasi *address* digunakan untuk *Register File* dan *I/O Memory* sementara 512 lokasi *address* lainnya digunakan untuk internal data SRAM. *Register File* terdiri dari 32 *general purpose working register*, *I/O register* terdiri dari 64 *register*.



Gambar 2.3 Data memori[1]

- EEPROM Data Memori

ATMega8535 memiliki EEPROM sebesar 512 *byte* untuk menyimpan data. Lokasinya terpisah dengan sistem *address register*, *data register* dan *control register* yang dibuat khusus untuk EEPROM.

2.1.2 Register I/O

Setiap port ATMega8535 terdiri dari 3 *register* I/O yaitu DDRx, Portx dan PINx.

- DDRx (*Data Direction Register*)

Register DDRx digunakan untuk memilih arah *pin*. Jika DDRx = 1 maka Pxn sebagai *pin* output Jika DDRx = 0 maka Pxn sebagai input.

- Portx (Port Data Register)

Register Portx digunakan untuk 2 keperluan yaitu untuk jalur output atau untuk mengaktifkan *resistor pullup*.

- PINx (Port *Input* Pin Address)

Digunakan sebagai register input.

Untuk lebih jelas,perhatikan Tabel 2.1.

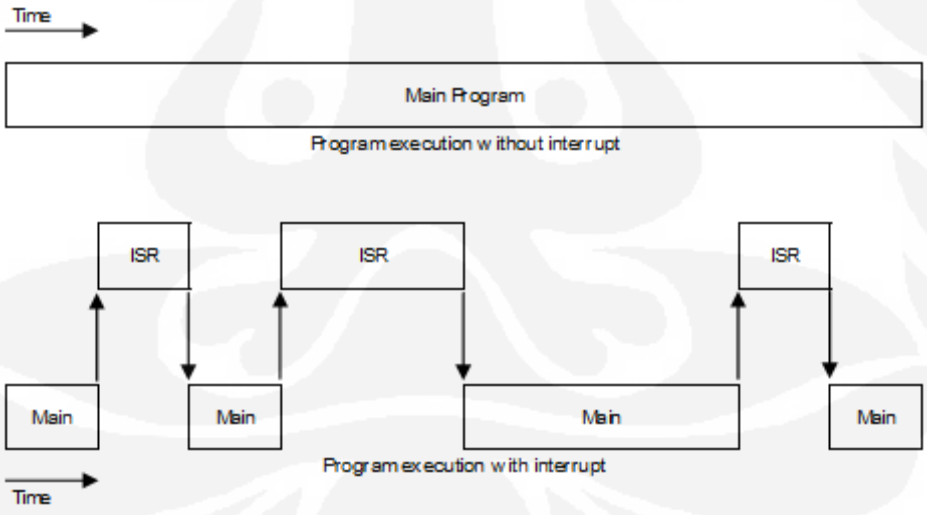
Tabel 2.1 Konfigurasi *pin* port ATMega8535[1]

DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

2.1.4 Interupsi

Interupsi adalah kondisi yang memaksa mikrokontroler menghentikan sementara eksekusi program utama untuk mengeksekusi rutin *interrupt* tertentu/*Interrupt Service Routine* (ISR).

Setelah melaksanakan ISR secara lengkap, maka mikrokontroler akan kembali melanjutkan eksekusi program utama yang tadi ditinggalkan. Gambar 2.4 menunjukkan saat program utama dikerjakan oleh mikrokontroler ATMega8535 kemudian tiba-tiba berhenti sementara waktu karena ada rutin lain yang harus ditangani oleh mikrokontroller ATMega8535, dan setelah selesai mengerjakan rutin tersebut mikrokontroler kembali mengerjakan instruksi pada program utama.



Gambar 2.4 Siklus interupsi pada ATMega8535[1]

Pada ATMega 8535 terdapat 21 sumber interupsi seperti yang dijelaskan pada Tabel 2.2.

Tabel 2.2 Vektor Interupsi dan Reset[1]

Vector No.	Program Address	Source	Interrupt Definition
1	0x000	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	TIMER2 COMP	Timer/Counter2 Compare Match
5	0x004	TIMER2 OVF	Timer/Counter2 Overflow
6	0x005	TIMER1 CAPT	Timer/Counter1 Capture Event
7	0x006	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	0x007	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	0x008	TIMER1 OVF	Timer/Counter1 Overflow
10	0x009	TIMER0 OVF	Timer/Counter0 Overflow
11	0x00A	SPI, STC	Serial Transfer Complete
12	0x00B	USART, RXC	USART, Rx Complete
13	0x00C	USART, UDRE	USART Data Register Empty
14	0x00D	USART, TXC	USART, Tx Complete
15	0x00E	ADC	ADC Conversion Complete
16	0x00F	EE_RDY	EEPROM Ready
17	0x010	ANA_COMP	Analog Comparator
18	0x011	TWI	Two-wire Serial Interface
19	0x012	INT2	External Interrupt Request 2
20	0x013	TIMER0 COMP	Timer/Counter0 Compare Match
21	0x014	SPM_RDY	Store Program Memory Ready

Interupsi Eksternal

Pada ATMega8535 terdapat 3 pin untuk interupsi eksternal, yaitu INT0, INT1, dan INT2. Interupsi eksternal dapat dibangkitkan apabila terdapat perubahan logika0 pada pin INT0, INT1, dan INT2. Pengaturan kondisi keadaan yang menyebabkan terjadinya interupsi eksternal diatur oleh *register* MCUCR (MCU *Control Register*), seperti Gambar 2.5 berikut.

Bit	7	6	5	4	3	2	1	0
	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00
Read/Write	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

Gambar 2.5 *Register* MCUCR[1]

- Bit ISC11 dan ISC10 menentukan kondisi yang dapat menyebabkan interupsi eksternal pada pin INT1. Konfigurasi *bit* ISC11 dan ISC10 dapat dilihat pada Tabel 2.3 berikut.

Tabel 2.3 Konfigurasi bit ISC11 dan ISC10[1]

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

- Bit ISC01 dan ISC00 menentukan kondisi yang dapat menyebabkan interupsi eksternal pada pin INT0. Konfigurasi *bit* ISC01 dan ISC00 dapat dilihat pada Tabel 2.4 berikut.

Tabel 2.4 Konfigurasi bit ISC01 dan ISC00[1]

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

Pemilihan interupsi eksternal diatur oleh *register* GICR (*General Interupt Control Register*), seperti dapat dilihat pada Gambar 2.6 berikut ini.

Bit	7	6	5	4	3	2	1	0
	INT1	INT0	INT2	-	-	-	IVSEL	IVCE
Read/Write	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

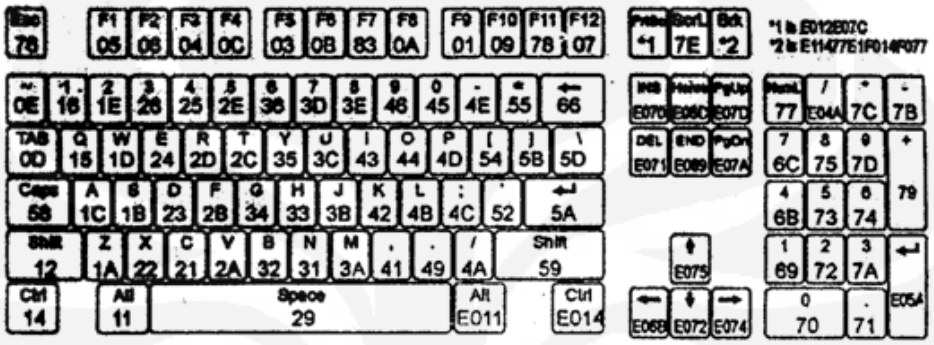
Gambar 2.6 *General interupt control register*[1]

Bit-bit *INT0, INT1, dan INT2 pada* register *GICR digunakan untuk mengaktifkan masing-masing interupsi eksternal. Ketika* bit-bit *itu diset 1(aktif) maka interupsi eksternal akan aktif jika* bit *1 (*interrupt*) pada SREG (*status register*) diset 1 juga (*enable interrupt*), instruksi untuk mengaktifkan* interrupt *yaitu* sei. *Program interupsi dari masing-masing interupsi akan dimulai dari* vektor *interupsi pada masing-masing jenis interupsi eksternal.*

2.2 Keyboard

Dalam beberapa proyek mikrokontroler, pemasangan *keypad* untuk input sudah cukup untuk dapat memasukkan data[4]. Akan tetapi, penggunaan *push-button* ataupun *keypad* ini memiliki keterbatasan, yaitu kedua fasilitas atau komponen memiliki jumlah tombol yang sedikit dan hanya dapat digunakan untuk pengambilan input data dalam jumlah yang terbatas.

Untuk mengatasi hal tersebut digunakan *keyboard*. Aplikasi penggunaan *keyboard* ini biasanya digunakan jika sebuah *keypad* sudah tidak mampu memenuhi kebutuhan karakter dari sistem yang dirancang. Misalnya pembuatan sebuah mesin ketik elektronik, *moving character display* (penampil karakter berjalan) dengan input *keyboard*, dan lain-lain.



Gambar 2.7 Konfigurasi data dari tombol *keyboard*[9]

Gambar 2.7 menunjukkan data dari masing-masing karakter tombol pada *keyboard* dalam heksadesimal. Jika karakter dari tombol di *keyboard* ditekan, maka data dikirimkan dalam bentuk sinyal data dan sinyal *clock*. Pengiriman satu paket data selalu diawali dengan *start bit* berlogika nol diikuti sinyal *clock*, dilanjutkan dengan 8 *bit* data yang dimulai dari bit 0 hingga bit ke 7, dan diakhiri dengan *parity bit* serta sebuah *stop bit* berlogika '1'. Setiap pengiriman *bit* data, baik pada *start bit*, 8 bit data, *parity* dan *stop bit* selalu diikuti dengan sebuah sinyal *clock* yang digunakan bagi bagian penerima bahwa satu *bit* data telah terkirim, *keyboard* membangkitkan sinyal *clock* dan pulsa *clock* pada umumnya yaitu 60-100µs sebagaimana tercantum dalam pedoman menghubungkan mikrokontroller dengan keyboard[9]. Perhatikan Gambar 2.8 berikut ini.

Gambar 2.8 Sinyal *clock* dan data dari *keyboard*[9]

Setelah bentuk data dikenali, kemudian dilanjutkan dengan pengenalan bentuk-bentuk kode (*scancode*) yang digunakan pada komunikasi data PC *keyboard* sebagai berikut:

- 01H hingga 83H adalah *scancode*.
- F0H sebagai awalan dari *scancode* menandakan ada tombol yang dilepas.
- E0H sebagai awalan dari *scancode* tombol tambahan.
- FAH, AAH, EEH, FFH, 00H yaitu kode-kode yang dipakai untuk menjawab perintah dari perangkat yang terhubung dengan *keyboard*.

Setelah *scancode* dikenali, dilakukan perancangan program untuk *keyboard*. Tahap awal merancang diagram alir (*flowchart*) dan program untuk pengambilan sebuah *scancode* yang berupa data PC *keyboard* serta program untuk konversi *scancode* tersebut ke dalam bentuk kode ASCII.

Tipe *socket keyboard* yang digunakan dalam pembuatan tugas akhir ini adalah tipe PS2 / DIN6. Konfigurasi *pin-pin* daripada *socket* PS2 / DIN6 ini ditunjukkan oleh Tabel 2.5 berikut.

Tabel 2.5 Konfigurasi pin Keyboard[9]

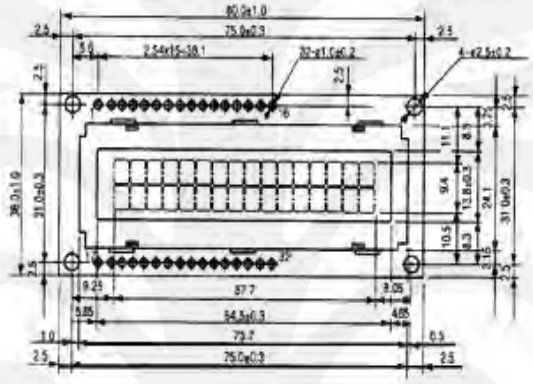
AT Computer Signals	 DIN41524, Female at Computer, 5-pin DIN 180°	 6-pin Mini DIN PS2 Style Female at Computer
Clock	1	5
Data	2	1
nc	3	2,6
GND	4	3
+5V	5	4
Shield	Shell	Shell

2.3 Liquid Crystal Display (LCD)

Sarana peraga saat ini sudah sangat banyak, antara lain berupa *Cathode Ray Tube* (CRT), *Liquid Crystal Display* (LCD), *dot matrix*, *seven-segment*, dan lain sebagainya. Masing-masing peraga tersebut memiliki kelebihan dan kekurangan tergantung dari aplikasinya[12]. Teknologi LCD sudah banyak digunakan dibandingkan teknologi CRT. Dibandingkan dengan CRT, LCD memiliki bentuk yang lebih ringan dan ringkas, konsumsi daya yang lebih kecil,

radiasi yang lebih kecil, dan lebih nyaman dimata. Namun LCD juga memiliki kekurangan dibandingkan dengan CRT, diantaranya ketajaman gambar atau tulisan yang dihasilkan lebih kurang, hanya dapat bekerja pada satu macam resolusi, dan memiliki sudut pandang yang lebih kecil.

LCD dapat digunakan untuk menampilkan karakter baik berupa huruf maupun angka. LCD memiliki ukuran yang bermacam-macam, seperti LCD dengan jumlah 1 – 4 baris, 16 – 40 karakter per baris, dan sebagainya. Salah satu contoh LCD tersebut adalah LCD 2x16. Gambar daripada LCD 2x16 ini ditunjukkan pada Gambar 2.9 berikut.



Gambar 2.9 LCD 2X16[7]

Pada umumnya, LCD memiliki 16 *pin* yang terdiri dari delapan pin jalur data (D0 – D7), tiga *pin* jalur kontrol (RS, E, dan RW), *pin* sumber tegangan dan *ground*, dan sebuah *pin* driver LCD dan dua *pin backlight*. Tabel 2.6 berikut menunjukkan konfigurasi dari *pin-pin* LCD tersebut.

Tabel 2.6 Konfigurasi *pin* LCD 2X16[7]

Pin Functions		
No.	**Name**	**Function**
1	Vss	GND
2	Vdd	Power supply voltage +5V
3	Vlc	Liquid crystal driving voltage
4	RS	L : Instruction code input. H: Data input
5	R/W	L : Data write from MPU to LCM. H: Data read from LCM to MPU
6	E	Enable
7	DB0	Data bus line
8	DB1	Data bus line
9	DB2	Data bus line
10	DB3	Data bus line
11	DB4	Data bus line
12	DB5	Data bus line
13	DB6	Data bus line
14	DB7	Data bus line
15	Va	Anode
16	Vc	Cathode

Untuk dapat mengatur tampilan LCD ini diperlukan karakter generator, yaitu bentuk-bentuk karakter yang dapat ditampilkan. Urutan dan posisi dari karakter yang akan ditampilkan dan pergantian ke *display* harus disimpan dan digabungkan disimpan di RAM. Semua pengontrol tampilan ini telah dibentuk dalam satu IC modul LCD yang berfungsi menerima kode-kode karakter (8-bit per karakter) dari suatu mikroprosesor atau komputer dan menyimpannya di *display* data RAM (DD RAM). Karakter ini akan dihubungkan dengan pola karakter yang tersedia pada karakter generator ROM (CG ROM). Jadi pola karakter yang dapat ditampilkan hanya pola karakter yang tersedia pada CG ROM. Namun, pemakai dapat mendefinisikan 8 pola (8-*bit*) tambahan yang disimpan di CG RAM. Module ini dapat menerima data atau instruksi dari mikroprosesor atau komputer dengan konfigurasi koneksi (*interface*) 4-*bit* atau 8-*bit* dan *supply* 5 volt.

CG ROM mempunyai 160 (1280 byte) karakter yang disimpan dalam bentuk 7x5 *dot matrix* sehingga pola satu karakter disimpan dalam 8-*bit*. CG RAM dapat menyimpan 8 pola (64 *byte*) karakter tambahan. Sementara CG RAM mempunyai kapasitas 80 kode karakter (80 *byte*).

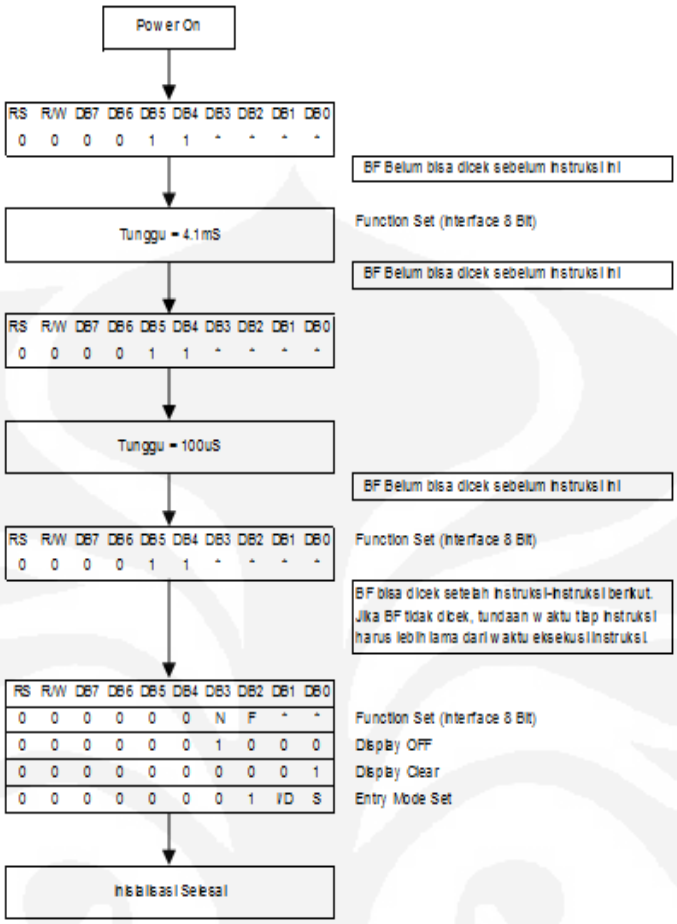
Untuk menampilkan satu karakter, posisi data pada tampilan dikirim ke *register* instruksi diikuti kode karakter ke *register* data. Module LCD akan menghubungkan karakter dengan pola karakter pada CG ROM dam mengirimkan pola karakter pada *display* sesuai dengan posisinya. Posisi dari tampilan dapat dikurangi atau ditambah secara otomatis tergantung dari inisialisasi yang dilakukan sebelum mengisi karakter sehingga dapat mengirimkan karakter yang

berurutan (*string* yang lebih dari satu karakter) dan akan ditampilkan satu *string* yang kontinyu. Penjelasan yang lebih detail mengenai LCD ini dapat dilihat pada lampiran yang berisikan datasheet LCD.

Langkah yang perlu dilakukan sebelum menampilkan karakter pada LCD adalah melakukan inisialisasi LCD terlebih dahulu. Inisialisasi LCD adalah hal yang terpenting, jika inisialisasi gagal, maka tidak ada tampilan atau yang tampil pada LCD adalah karakter-karakter aneh. Pada tahap inisialisasi ini berisi konfigurasi dari LCD yang akan digunakan. Adapun konfigurasi yang harus diatur pada tahap inisialisasi ini adalah :

1. Banyaknya *bit data interface* dengan MPU yang digunakan (8 *bit* atau 4 *bit*).
2. Jumlah baris pada LCD yang digunakan.
3. Pergeseran *cursor*.
4. Pergeseran tampilan.
5. *Cursor* atau tanpa *cursor*, berkedip atau tidak berkedip.

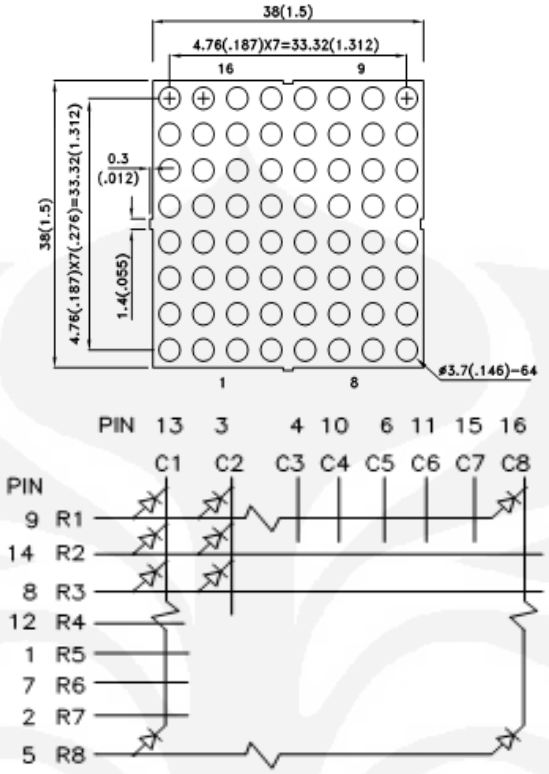
Diagram alir (*flowchart*) daripada inisialisasi LCD tersebut dapat dilihat pada Gambar 2.10. Contoh program untuk inisialisasi dan menampilkan *string* pada LCD dapat dilihat pada lampiran yang berisikan program untuk pengetesan LCD.



Gambar 2.10 Flowchart Inisialisasi LCD[8]

2.4 *Dot Matrix* LED 8x8

Sama halnya dengan LCD, *dot matrix* ini juga terdiri dari berbagai macam ukuran, salah satunya adalah *dot matrix* 8x8. Sesuai dengan namanya, komponen ini tersusun atas 64 buah LED (*Light Emitting Diode*), yang terdiri dari 8 baris dan 8 kolom. *Dot matrix* 8x8 ini memiliki rangkaian internal seperti yang ditunjukkan oleh Gambar 2.11 berikut.

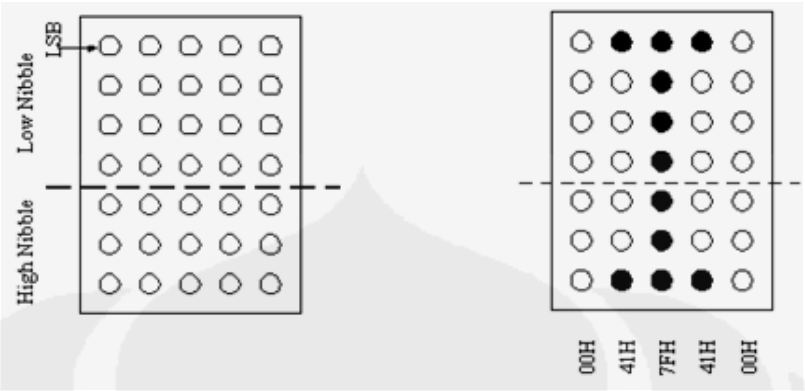


Gambar 2.11 Rangkaian internal *dot matrix* 8x8[10]

Dot matrix yang ditunjukkan oleh Gambar 2.11, tersusun atas 64 buah LED berwarna merah. Untuk menyalakan sebuah LED dibutuhkan tegangan 2,5V dengan arus 20mA.

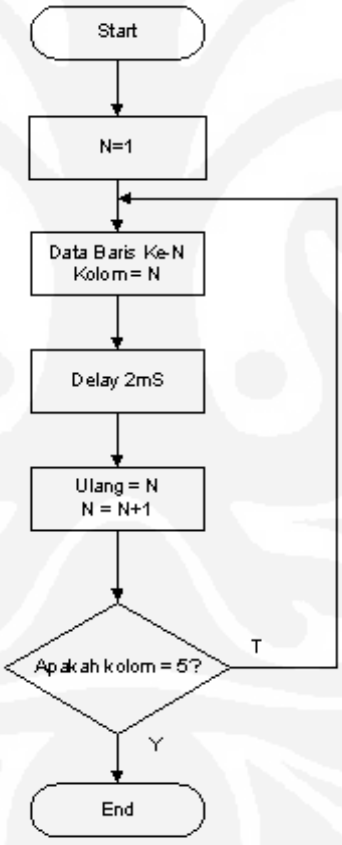
Untuk menampilkan suatu karakter pada *dot matrix* tersebut tidak dapat dilakukan sekaligus, melainkan melalui proses *scanning*. Susunan data yang harus dikirimkan ke *dot matrix* dapat dilihat pada Gambar 2.12 dan untuk menampilkan karakter “I” prosesnya adalah sebagai berikut:

1. Kirim data 00h ke PORTA (port yang mengendalikan baris) dan PORTB (port yang mengendalikan kolom) diberi data 00h untuk kolom pertama.
2. Kemudian kirim data 41h ke PORTA dan 01h pada PORTB agar kolom kedua aktif.
3. Dan seterusnya diulangi beberapa kali. Dengan kecepatan scanning yang sangat cepat, akan tampak berupa satu huruf.



Gambar 2.12 Scanning *dot matrix* LED[1]

Flowchart untuk menampilkan sebuah karakter pada *dot matrix* LED ditunjukkan pada Gambar 2.13 berikut:



Gambar 2.13 Flowchart scanning *dot matrix* LED

2.5 Pemrograman Mikrokontroler ATmega8535

Untuk memaksimalkan fungsi perangkat yang dibuat dibutuhkan suatu program khusus. Untuk membuat program dibutuhkan juga *software* khusus yang berfungsi untuk membuat, meng*compile* dan men*download* program ke mikrokontroller. AVRSTUDIO merupakan *software* khusus untuk bahasa *assembly* yang mempunyai fungsi sangat lengkap, yaitu digunakan untuk menulis program, kompilasi, simulasi dan *download* program ke IC mikrokontroler AVR. Sedangkan CodeVisionAVR merupakan *softwar*e *C-cross compiler*, dimana program dapat ditulis dalam bahasa C, Codevision memiliki IDE (*Integrated Development Environment*) yang lengkap, dimana penulisan program, *compile*, *link*, pembuatan kode mesin (assembler) dan *download* program ke *chip* AVR dapat dilakukan pada codevision, selain itu ada fasilitas terminal, yaitu untuk melakukan komunikasi serial dengan mikrokontroler yang sudah di program. Keuntungan dalam menggunakan bahasa pemrograman C pada mikrokontroler, yaitu:

- Waktu pemrograman dan tes program relatif lebih pendek.
- Pengetahuan terhadap instruksi *set* prosessor tidak terlalu dibutuhkan. Hanya pengetahuan dasar mengenai struktur memori CPU yang dibutuhkan, meskipun tidak terlalu penting.
- Perincian seperti alokasi *register*, pengalamatan memori, dan tipe data telah diatur oleh *compiler*.
- Memiliki kemampuan untuk mengkombinasikan *variabel* dengan operasi khusus sehingga dapat meningkatkan pembacaan program.
- Program menjadi lebih terstruktur dan mudah dipahami, dan program dapat dibagi menjadi beberapa fungsi yang terpisah.

Proses *download* program ke IC mikrokontroler AVR dapat menggunakan sistem *download* secara ISP (*In-System Programming*). *In-System Programmable Flash on-chip* mengizinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI.

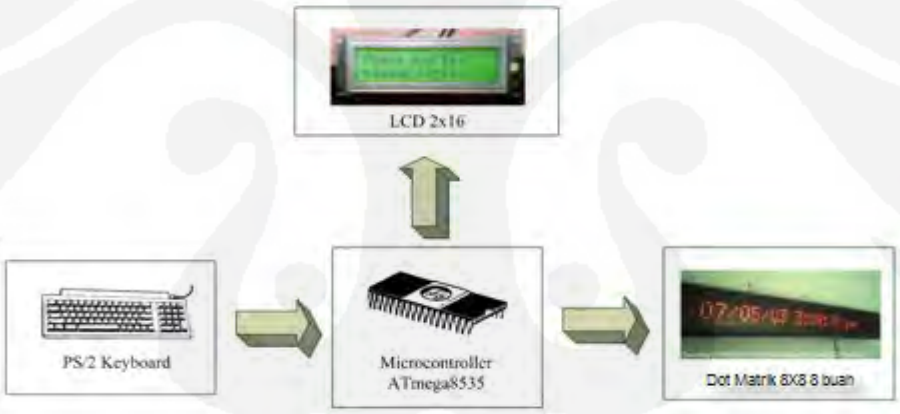
BAB III

Perancangan Papan Informasi Elektronik dengan PS/2 keyboard

Setelah mempelajari komponen-komponen yang digunakan seperti yang sudah dijelaskan pada Bab 2, selanjutnya dilakukan proses perancangan seperti dijabarkan berikut ini.

3.1 Perancangan Sistem

Pada perancangan ini komponen yang digunakan yaitu PS/2 *keyboard*, ATMega8535 sebagai sistem minimum, LCD dan *dot matrix*. Komponen-komponen tersebut memiliki fungsi tersendiri sesuai apa yang dijelaskan berikut ini. Tahap selanjutnya dilakukan perancangan seperti blok diagram dibawah ini.



Gambar 3.1 Blok diagram perancangan papan informasi elektronik dengan PS2 *keyboard*

Spesifikasi Alat

- Mikrokontroler ATMega8535 sebagai unit pemroses data.
- 8x8 *Dot Matrix segment* 8 buah.
- Data karakter dapat disimpan pada EEPROM ATMega8535.
- Tampilan keluaran bergeser ke kiri.
- LCD *alphanumeric* 2x16 sebagai *indicator* pada mode *edit*.

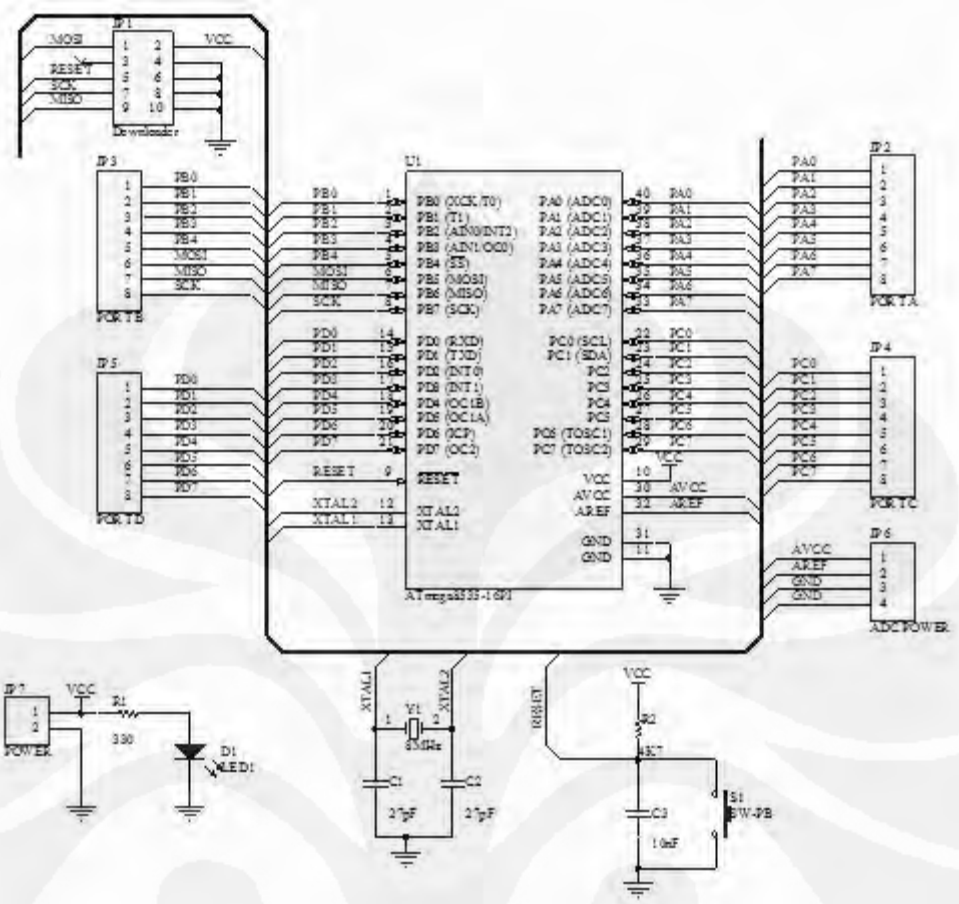
3.2 Rancang Bangun Perangkat

Rancang bangun perangkat keras meliputi beberapa bagian, yaitu:

- Perancangan sistem minimum menggunakan ATMega8535.
  Untuk menyimpan program dan sebagai pengendali terhadap keluaran yang diinginkan. Seperti telah diketahui, ATMega memiliki kemampuan untuk menyimpan data didalam EEPROM.
- Perancangan antarmuka LCD 2x16 dengan mikrokontroler ATMega8535.
  LCD 2X16 ini digunakan untuk membaca tulisan yang akan diinputkan oleh *keyboard*.
- Perancangan *shift left register* dan *dot matrix* LED
  Kedua komponen ini digunakan untuk menampilkan hasil tulisan yang diinput setelah disimpan ke EEPROM. Lalu tulisan akan digeser kekiri supaya terlihat lebih menarik.
- Perancangan antarmuka PS/2 *keyboard*.
  Komponen ini digunakan untuk memberikan karakter huruf yang terdapat pada tombol *keyboard* yang ditekan sehingga akan terbentuk suatu tulisan.

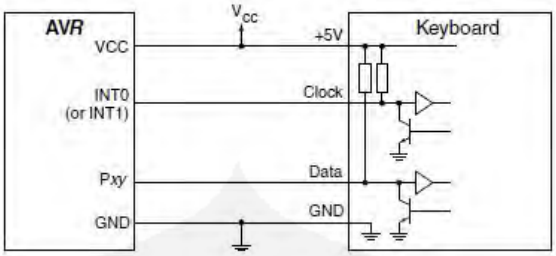
Rangkaian keseluruhan dari alat yang dibuat ditujukkan pada Gambar 3.2 sebagai lampiran A. Cara kerja dari rangkaian pada Gambar 3.2 adalah sebagai berikut:

ATMega8535 memiliki empat port yaitu Port A, Port B, Port C dan Port D. Pada skripsi ini PORTD disambungkan dengan *clock* dan data dari PS/2 *keyboard*, pengontrol LCD RS, R/W dan *Enable*. PORTA berfungsi sebagai data keluaran ke *dot matrix* LED, PORTC berfungsi sebagai 8-*bit* data keluaran ke LCD 2x16, sedangkan PORTB berfungsi untuk keluaran sinyal *shift left register* pada *dot matrix* LED. Gambar 3.3 berikut ini merupakan rangkaian sistem minimum ATMega8535.



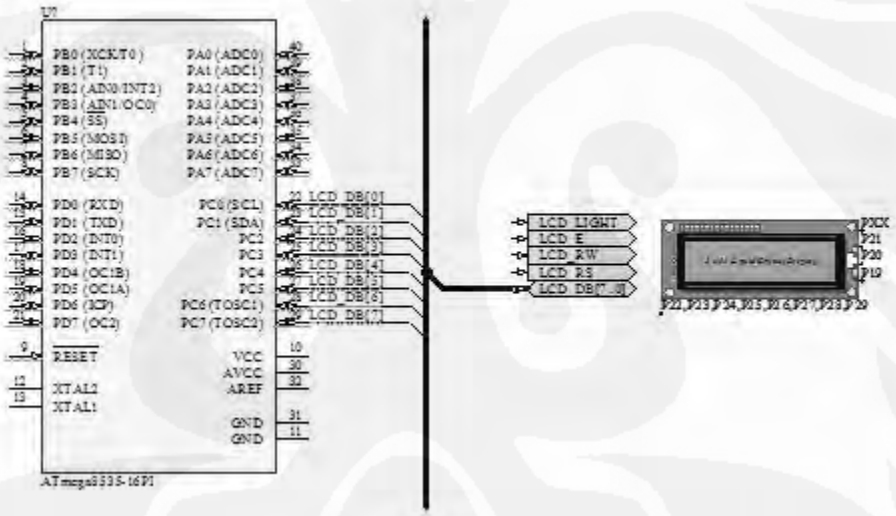
Gambar 3.3 Sistem Minimum mengunakan ATMega8535

Selanjutnya PS/2 *keyboard* akan mengirimkan 2 buah sinyal keluaran berupa *clock* dan data. Pada saat tombol tersebut ditekan, maka *clock* dan data mengeluarkan pulsa sebanyak 11 *bit*. Pada umumnya sinyal *clock* dan pulsa *clock* yang dihasilkan dari *keyboard* yaitu 30-50 µs untuk low logic dan 30-50 µs untuk *high logic*. *Pin clock* pada PS/2 *keyboard* ini terhubung dengan interupsi eksternal mikrokontroler ATMega8535 PORTD.2 untuk mendeteksi ada atau tidaknya tombol *keyboard* yang ditekan, sedangkan *pin* data terhubung dengan PORTD.4. PS/2 *keyboard* terdiri 6 buah *pin*, tetapi hanya 4 buah *pin* yang terpakai yaitu: VCC, GND, *clock*, dan data. *Pin* VCC dan GND mendapat *supply* 5V. Gambar 3.4 berikut ini merupakan antarmuka PS/2 *keyboard* dengan sistem minimum ATMega8535.



Gambar 3.4 Antarmuka PS2 *keyboard* dengan ATMega8535[5]

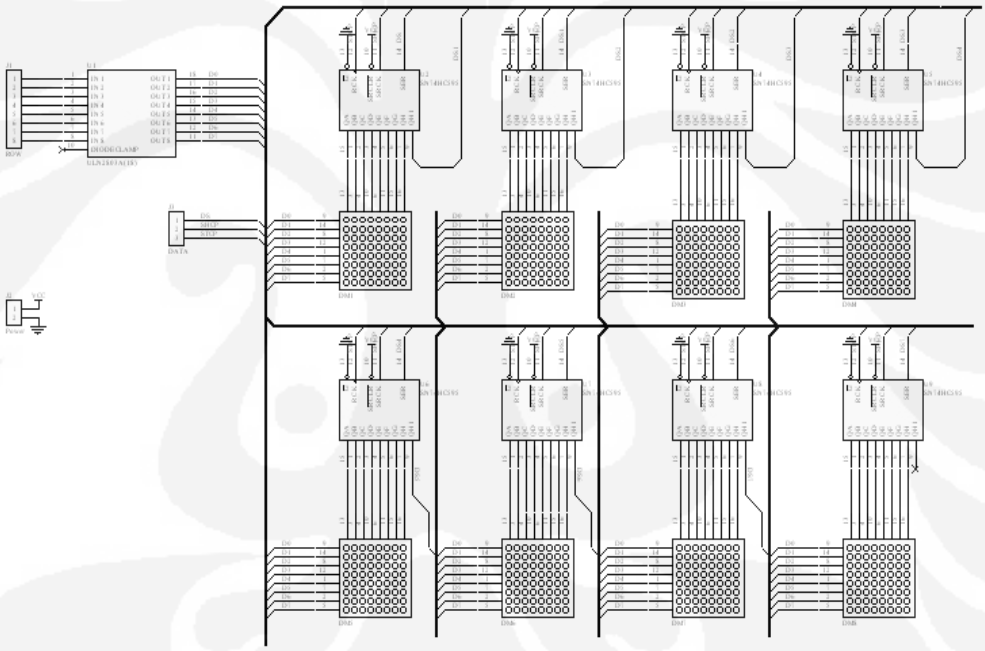
Mikrokontroler ATMega8535 yang telah menerima data dari PS/2 *keyboard* memproses data tersebut dari *scancode* ke bentuk ASCII, lalu menampilkan data karakter yang diinput dari *keyboard* ke LCD 2x16 yang dikonfigurasikan menggunakan pengalamatan data 8-*bit*, yaitu pada PORTC.0 ~ PORTC.7. Mode 8-*bit* ini dipilih karena proses untuk menampilkan data pada LCD lebih cepat bila dibandingkan dengan mode 4-*bit*. Gambar 3.5 berikut ini merupakan antarmuka LCD dengan mikrokontroller ATMega8535.



Gambar 3.5 Blok antarmuka LCD dengan ATMega8535

Karakter-karakter yang telah diinput melalui *keyboard* disimpan dalam suatu tempat penyimpanan sementara atau *register* sehingga kumpulan karakter-karakter tersebut menjadi sekumpulan kata atau kalimat. Ketika tombol *enter* pada PS/2 *keyboard* di tekan, sekumpulan karakter tersebut disalin ke memori EEPROM pada mikrokontroler ATMega8535, lalu data tersebut dikeluarkan melalui PORTA untuk ditampilkan pada *dot matrix* LED dan tulisan digeser menggunakan komponen *shift register*.

Shift register ini merupakan *serial to parallel shift register* yang dikonfigurasikan sebagai *shift left register* (*register* geser kiri). *Shift left register* ini digunakan untuk *scanning* kolom pada *dot matrix* LED. Tampilan pada *dot matrix* LED ini menggunakan teknik *scanning display*, yaitu menyalakan tiap-tiap kolom secara bergantian dengan kecepatan yang sangat tinggi. *Scanning display* ini bertujuan untuk menghindari pembebanan arus pada rangkaian. *Shift register* ini menerima data berupa *serial input*, STCP, dan SHCP dari mikrokontroler ATMega8535 untuk memulai proses *scanning display*. Apabila pada suatu kolom bernilai logika "1" dan pada suatu baris bernilai logika "1", maka *dot matrix* LED tersebut akan menyala. Untuk lebih jelas perhatikan Gambar 3.6 berikut ini.

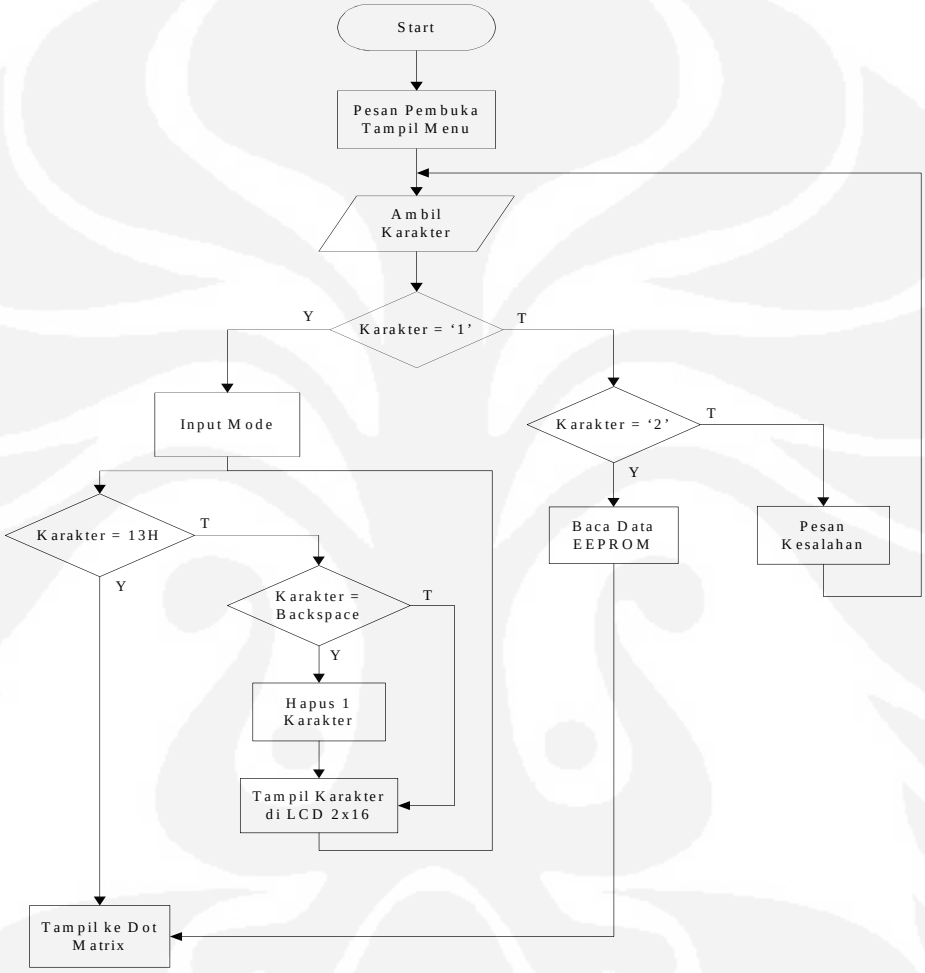


Gambar 3.6 *Shift register* dan *dot matrix* segmen

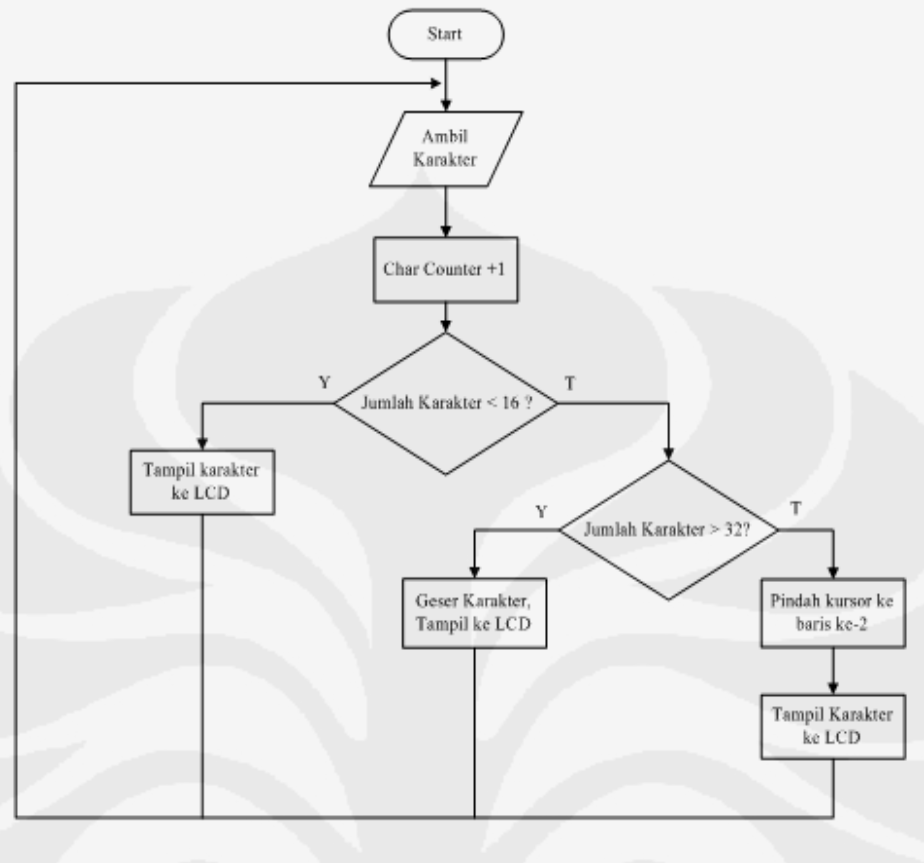
3.3 Perancangan dan Realisasi Perangkat Lunak (Software)

Perancangan perangkat lunak (*software*) dilakukan untuk mendukung kerja sistem berdasarkan *hardware*-nya, agar sistem dapat bekerja sesuai dengan fungsi dan aplikasinya. Bahasa pemrograman yang digunakan dalam perancangan dan pembuatan *software* sistem adalah bahasa tingkat tinggi, yaitu bahasa C untuk mikrokontroler seri Atmel AVR.

Software yang digunakan yaitu CodeVisionAVR 2.03.4 sebagai *compiler* bahasa C. Dengan *software* aplikasi ini, maka akan didapatkan *file* berekstensi (*.hex). *File* ini yang nanti akan di *download* pada *Flash* PEROM mikrokontroler ATMega8535, sebagai program untuk mengendalikan kinerja dari sistem yang dibuat.



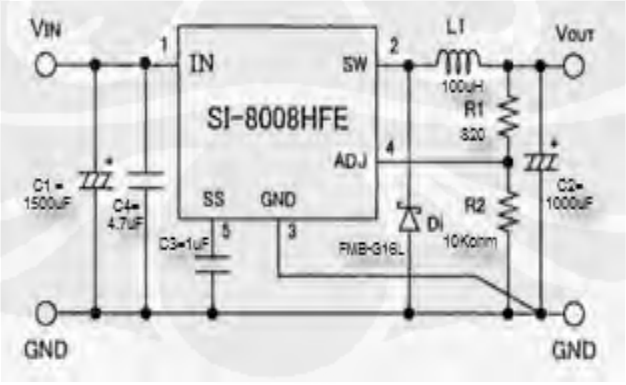
Gambar 3.7 Flowchart sistem secara umum



Gambar 3.8 Flowchart mode input

3.4 Power Supply

Untuk menyalakan sistem dari alat yang dibuat, dibutuhkan catu daya +5V dc dengan current diatas 1A. Pada skripsi ini, *power supply* yang dibuat menggunakan IC regulator SI-8088HFE. Input yang dipakai menggunakan adaptor +19V dc. Gambar 3.9 berikut ini merupakan rangkaian *power supply* .



`Gambar 3.9 Rangkaian *power supply*[6]

BAB IV

PENGUJIAN PAPAN INFORMASI ELEKTRONIK DENGAN PS2 *KEYBOARD*

Setelah dilakukan perancangan dan perakitan papan informasi ektronik dengan PS2 *keyboard*, tahap selanjutnya adalah menguji kinerja dari komponen maupun papan informasi elektronik dengan PS2 *keyboard* yang sudah dibuat.

Pengujian yang dilakukan meliputi :

1. Pengujian keluaran *power supply*.
2. Pengujian sistem minimum mikrokontroler ATMega8535.
3. Pengujian pengiriman data *scancode* dari PS/2 *keyboard* ke mikrokontroler.
4. Pengambilan data waktu respon untuk pemunculan setiap karakter.
5. Pengujian sistem secara keseluruhan menampilkan pesan bergeser pada *dot matrix* LED.

4.1 Pengujian *Power Supply*

Pengujian *power supply* dilakukan dengan mengukur tegangan +19Vdc pada bagian input dan tegangan +5Vdc pada bagian output seperti yang ditunjukkan pada Gambar 3.9. Pengukuran tegangan ini dimaksudkan untuk mengetahui kestabilan tegangan dioutput sehingga dapat mencatu sistem minimum dan *dot matrix* dengan maksimal.

Titik-titik pengukuran adalah :

a. Keluaran IC SI-8088HFE pin 2

   Nilai tegangan keluaran idealnya tergantung dari adjust di R1, karena diinginkan tegangan +5Vdc maka R1 diatur sampai didapatkan nilai +5Vdc. Nilai tegangan yang didapat adalah 5,02V.

b. Masukkan IC SI-8088HFE di pin 1

   Nilai tegangan masukan IC SI-8088HFE adalah 19Vdc yang diambil dari adaptor. Nilai tegangan yang didapat adalah 18,98V.

4.2 Pengujian sistem minimum mikrokontroler ATMega8535

Pengujian terhadap sistem minimum ATMega8535 dilakukan untuk mengetahui kinerja dari IC ATMega8535 terhadap program yang dibuat. Pengujian dilakukan dengan menjalankan program bahasa *assembly* atau bahasa tingkat tinggi lainnya seperti BASIC, C, PASCAL, dan lain-lain. Pada pengujian ini, digunakan bahasa C karena lebih *portable* dan *fleksibel*, memiliki daftar pustaka yang banyak, serta proses *executable* yang lebih cepat. Walaupun bagi pemula masih ada kesulitan dalam penggunaan *pointer*.

Sebelum proses *running* program maka yang perlu dilakukan adalah men*download* program ke *flash* PEROM pada mikrokontroler. Untuk men*download*, terlebih dahulu mikrokontroler ini harus terkoneksi dengan *downloader* yang terhubung dengan USB *port*. *Downloader* yang digunakan adalah *downloader* USBasp ISP (*In-System Programming*). Untuk mengecek kesiapan sistem minimum untuk proses *download*, maka pada konsol *window command prompt* diberikan instruksi dengan format berikut ini :

```
C:\avrdude -c <tipe ISP> -p <tipe IC uC> -U flash:w:<namafile.hex>

Contoh:

C:\avrdude -c usbasp -p m8535 -U flash:w:testlcd.hex
```

Pengujian sistem dilakukan terhadap *peripheral output* dan *input* pada *port* I/O mikrokontroler. Pada pengujian ini program pengujian dibuat untuk melakukan beberapa variasi penyalaan LED pada port B. Listing program pengujian adalah:

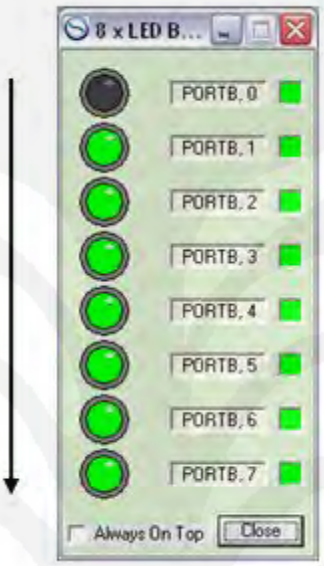
```
#include <mega8535.h>


#define xtal 8000000
#define fmove 2

// the LED on PORTC output 0 will be on
unsigned char led_status=0xfe;

// TIMER1 overflow interrupt service routine
// occurs every 0.5 seconds

interrupt [TIM1_OVF] void timer1_overflow(void)
{
// preset again TIMER1
TCNT1=0x10000-(xtal/1024/fmove);
// move the LED
led_status<<=1;
led_status|=1;
if (led_status==0xff) led_status=0xfe;
// turn on the LED
PORTB=led_status;
}
```

```
void main(void)
{
// set the I/O ports
// all PORTC pins are outputs
DDRB=0xff;
// turn on the first LED
PORTB=led_status;

TCCR1A=0x00;
TCCR1B=0x05;
TCNT1=0x10000-(xtal/1024/fmove);
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

MCUCR=0x00;
//EMCUCR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x80;

// global enable interrupts
#asm
    sei
#endasm

// the rest is done by TIMER1 overflow interrupts
while (1);
}
```

Dari program di atas PORTB dari mikrokontroler ATMega8535 terhubung dengan kondisi aktif rendah. Dengan memberikan logika “0” pada PORTB, maka LED akan menyala. Pada program diatas, LED akan menyala pada PORTB secara bergiliran dengan selang waktu 0.5 detik dan dimulai dari LSB

(*Least Significant Bit*) sampai MSB (*Most Significant Bit*). Pergeseran *bit* terjadi pada saat terjadinya *Timer Overflow* sehingga menyebabkan aktifnya fungsi interupsi untuk menghitung waktu selama 0.5 detik.

Dengan menggunakan *software* simulasi *AVR Simulator*, didapatkan keluaran berupa LED bergeser juga. Keluarannya dapat dilihat pada gambar 4.1berikut:



Gambar 4.1 Hasil simulasi dengan *AVR simulator* pada LED

Selanjutnya dilakukan pengujian terhadap LCD 2X16 untuk melihat apakah mikrokontroler juga sudah terkoneksi dengan benar dengan memberikan progam sehingga tampil tulisan "*Hello Word*".

- **Pengujian menampilkan karakter pada LCD 2x16**

Pada pengujian ini, digunakan fungsi pustaka LCD.H pada perangkat lunak CodevisionAVR 2.03.4 untuk inisialisasi LCD dan menambahkan intruksi lain untuk dapat menampilkan sebuah kalimat. Pengujian LCD 2x16 ini dilakukan dengan menampilkan kalimat "*Hello World*" pada LCD 2x16 dengan membuat program singkat berikut ini:

```
// the LCD is connected to PORTC outputs
// see the file lcd.h in the ..\inc directory
#asm
.equ __lcd_port=0x15 ;PORTC
#endasm

#include <mega8535.h>

// include the LCD driver routines
#include <lcd.h>

void main(void)
{
// initialize the LCD for
// 2 lines & 16 columns
lcd_init(16);

// go on the second LCD line
lcd_gotoxy(0,1);

// display the message
lcd_putsf("Hello world");

// stop here
while (1);
}
```

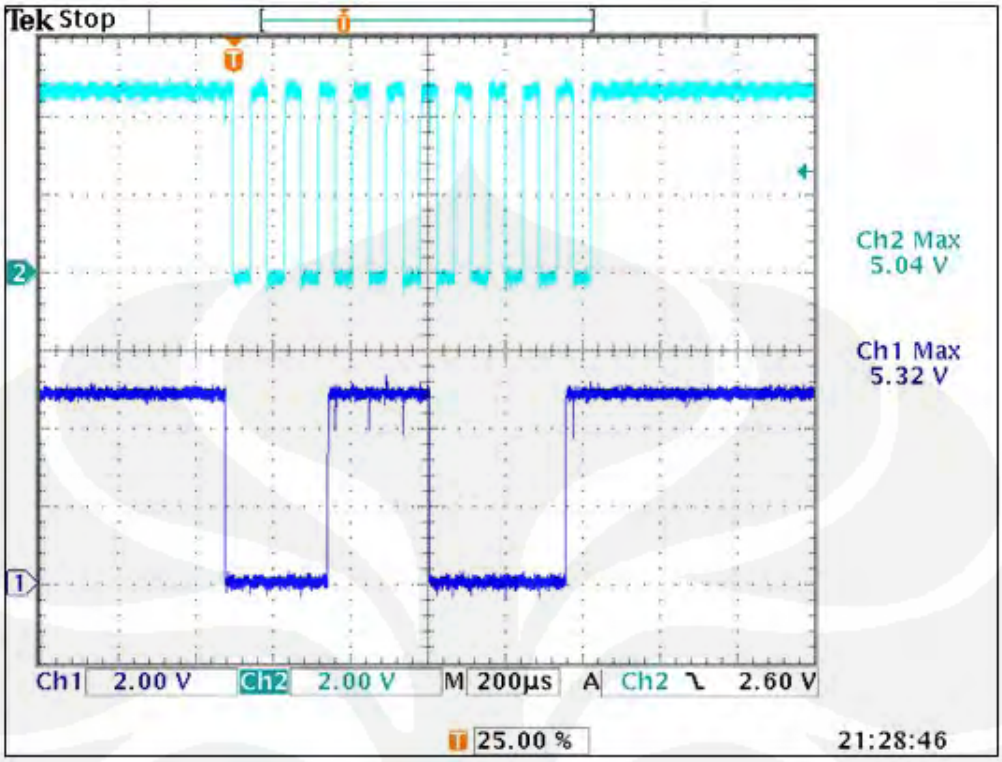
Dari program diatas, kalimat “Hello world” ditampilkan pada LCD 2x16 dengan posisi pada baris ke-2 dan dimulai dari kolom ke-1. Setelah program diatas di *compile* dan di *download* ke ATmega8535, ditunjukkan bahwa output kalimat tersebut beserta tata letaknya sesuai dengan yang diinputkan.

Dengan menggunakan *software AVR Simulator* juga didapat hasil keluaran yang sama pada LCD. Hasil keluarannya dapat dilihat pada Gambar 4.2 dibawah ini:

Gambar 4.2 Hasil output dengan menggunakan *avr simulator*

4.3 Pengujian PS/2 keyboard

Pengujian PS/2 *keyboard* ini bertujuan untuk mengetahui kinerja dari *keyboard* terhadap karakter yang ditekan pada tombol *keyboard*. Pengujian PS/2 *keyboard* dilakukan dengan mengukur parameter gelombang *clock* dan *data* yang berasal dari PS/2 *keyboard*. Pada pengujian ini, dilakukan dengan menekan tombol huruf “A” pada *keyboard* serta merekam gelombang output dari PS/2 *keyboard* menggunakan osiloskop dan ditunjukkan pada Gambar 4.3 berikut ini:

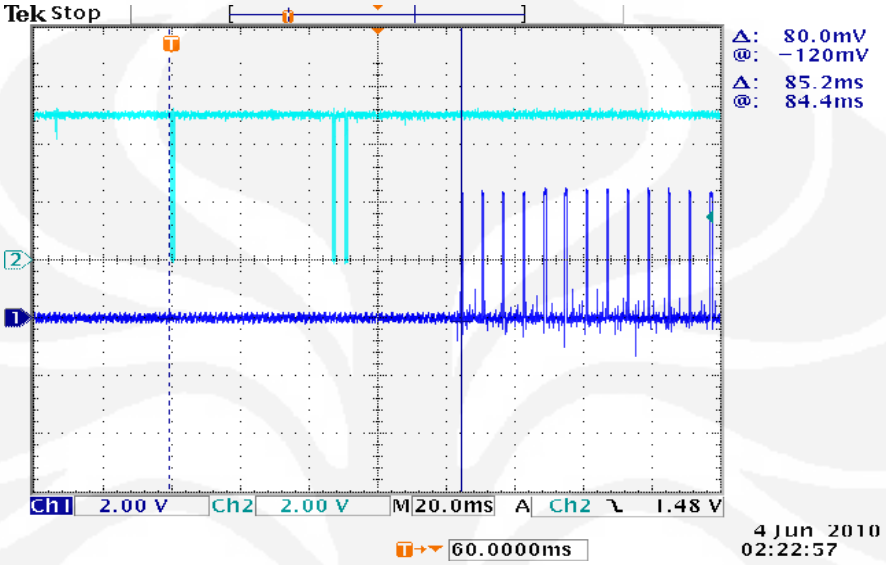


Gambar 4.3 Gambar gelombang antara data dengan *clock* pada *keyboard*

Dari gambar di atas dapat dilihat bahwa pada saat tombol "A" ditekan, gelombang *clock* (*channel-2*) berdetak sebanyak 11 *clock* sesuai dengan teori yaitu data tersebut dikirim secara serial dengan format: *Start bit – Data – Parity check bit – stop bit*. Selain itu, dari gambar 4.3 didapat bahwa *clock* yang dibutuhkan setiap karakter yang ditekan yaitu 920µs. Untuk satu *clock* dibutuhkan 83,63µs atau 11 KHz. Kemudian gelombang data (*channel-1*) menunjukkan nilai heksadesimal 0x1C (0001 1100). Jika dibandingkan dengan penjelasan prinsip kerja *clock* pada *keyboard*, hal ini masih memenuhi karakteristik *keyboard* dengan *clock* 60-100µs. Selain itu pula pada data tombol konfigurasi *keyboard* untuk huruf A adalah 1C (gambar 2.7). Data yang dikirim pertama kali yaitu mulai dari LSB hingga MSB. Data dengan nilai heksa 0x1C tersebut bukanlah merupakan data ASCII, tetapi data *scancode*. Oleh karena itu untuk dapat mengolah dan menampilkan karakter "A" tersebut pada LCD, maka *scancode* tersebut harus dikonversikan terlebih dahulu ke bentuk ASCII. Pengkonversian tersebut dilakukan dengan menggunakan program PS2 *keyboard*.

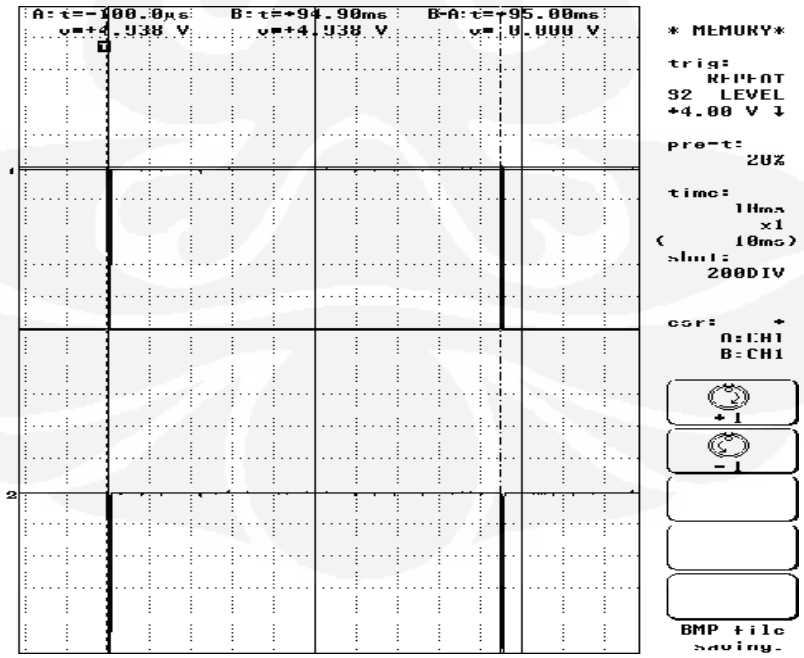
4.4 Data Waktu Respon

Berikutnya adalah pengujian respon waktu. Data *respon time* diukur untuk mengetahui kecepatan waktu muncul diantara setiap karakter yang tampil pada *display dot matrix* baik karakter yang sama maupun yang berbeda. Dibawah ini merupakan data *respon time* yang diukur dengan osiloskop.

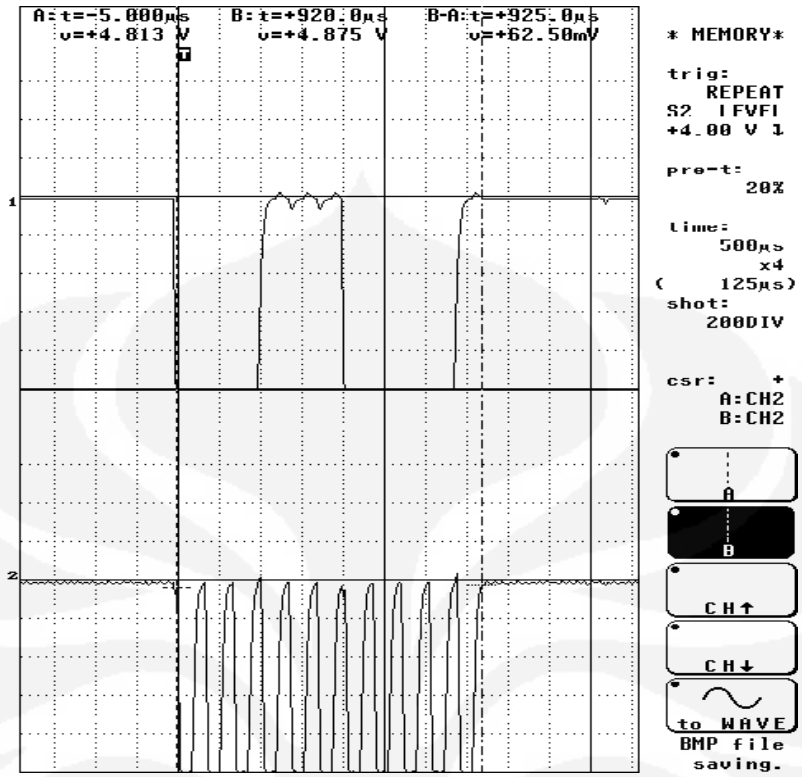


Gambar 4.4 Gelombang antara data pada *keyboard* dengan *display dot matrik*

Dari gambar 4.4 didapatkan bahwa *respon time* yang didapat untuk jeda waktu antara tombol enter ditekan dengan pemunculan karakter di *display dot matrix* adalah 85,2 ms.



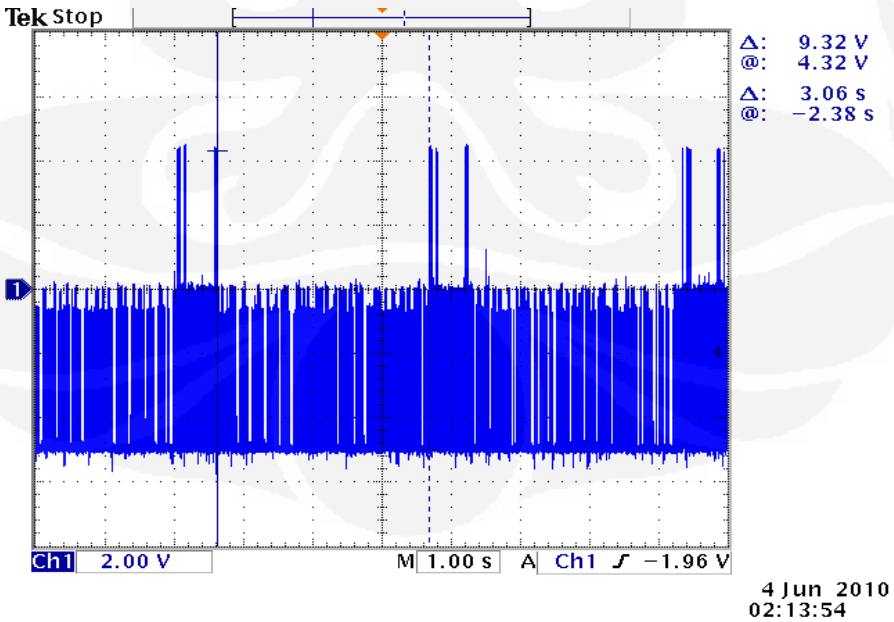
Gambar 4.5 Respon waktu pemunculan karakter dari *keyboard*



Gambar 4.6 *clock* data untuk satu karakter

Dari hasil pengukuran data waktu respon antara pemunculan karakter pertama dengan karakter berikutnya pada *keyboard* memiliki *respons time* yang sama yaitu 95ms.

Gambar 4.7 di bawah ini merupakan lamanya waktu untuk pemunculan dalam 1 siklus atau sering disebut dengan *looping time*.



Gambar 4.7 *Looping time* dari program yang dibuat

Dari gambar diatas diatas didapatkan bahwa *looping time* yang didapatkan adalah 3.06s.

Selain itu Gambar 4.7 dibawah ini merupakan waktu respon saat tombol *enter* ditekan pada *keyboard* sampai karakter itu muncul pada *display dot matrix*. Pengukuran ini diambil untuk mengetahui kecepatan waktu respon saat ditampilkan ke *dot matrix*.

4.5 Pengujian sistem secara keseluruhan

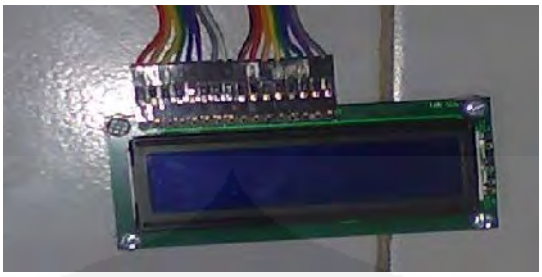
Tahap akhir adalah pengujian sistem secara keseluruhan untuk mengetahui kinerja dari alat dan program keseluruhan yang dibuat sesuai pada lampiran A. Pada pengujian ini, alat yang telah dibuat dioperasikan dengan cara membuat string menu pilihan "1:Input Mode" dan "2:Read EEPROM".

Gambar 4.8 Menu pilihan saat program dijalankan

Dengan menggunakan *software AVR Simulator* didapat hasil keluaran sesuai dengan program yang dibuat. Hasil keluarannya adalah muncul dua mode pada tampilan LCD yaitu mode input dan *read* EEPROM. Saat mode input, dapat dilakukan pengisian karakter dengan PS2 *keyboard*. Ketika ditekan tombol *Enter*, pada saat itu karakter yang telah diinput secara otomatis di simpan ke EEPROM dan pesan langsung ditampilkan ke *dot matrix display*.

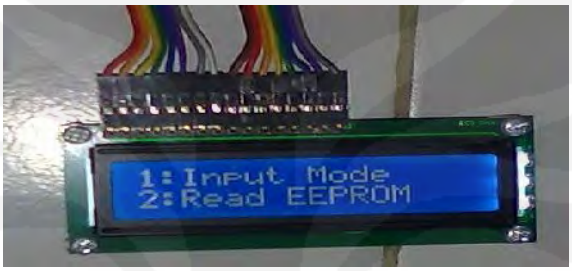
Untuk mengetahui data yang di *input* benar-benar telah disimpan dalam EEPROM, maka dilakukan pengujian dengan cara memutuskan hubungan sumber listrik dari alat ini, lalu kemudian menghubungkan sumber kembali. Pada menu awal tampilan LCD dipilih mode "Read EEPROM" dan lihat pada *dot matrix display* data terakhir yang disimpan tampil atau tidak.

Dari hasil pengujian ini, ditunjukkan bahwa perangkat dapat menampilkan data terakhir yang tersimpan walau telah terjadi putusnya hubungan arus listrik seperti dapat dilihat pada Gambar 4.9 hingga 4.11.



Gambar 4.9 Tampilan LCD saat tidak menyala

Saat sumber dimatikan, LCD tidak akan menampilkan tulisan seperti yang ditunjukkan pada gambar 4.9



Gambar 4.10 Tampilan setelah LCD menyala

Gambar 4.11 Tampilan setelah mode “2” dipilih

Saat sumber dinyalakan kembali tampilan seperti Gambar 4.10 akan tampil dan setelah dipilih mode “2” hasil tulisan yang diinput sebelum dimatikan sumber masih bisa terbaca.

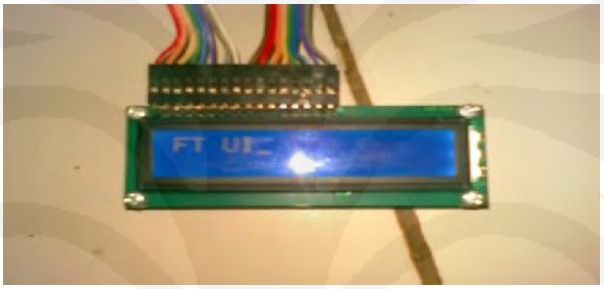
Urut-urutan tampilan perangkat secara keseluruhan dapat dilihat seperti pada Gambar 4.12 hingga 4.13.

Gambar 4.12 Tampilan judul setelah power on

Gambar 4.12 Menunjukkan bahwa saat *power on*, tampilan LCD menampilkan pesan pembuka.

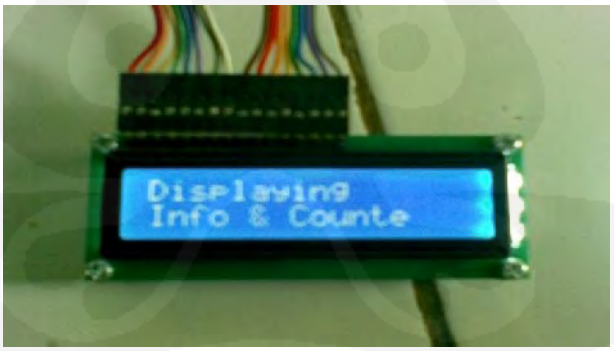
Gambar 4.13 Tampilan menu

Setelah 0,5 s, tampilan pada LCD berubah menjadi mode pilihan yaitu mode *input* dan mode *read* EEPROM. Gambar 4.13 Menunjukkan tampilan menu setelah pesan pembuka.



Gambar 4.14 Tampilan menu edit “1”

Menu edit “1” atau pilihan mode input digunakan untuk menuliskan pesan atau informasi sesuai yang ditunjukkan pada Gambar 4.14.



Gambar 4.15 Tampilan baca EEPROM

Gambar 4.15 Menunjukkan tampilan setelah disimpan ke EEPROM atau setelah tombol *enter* ditekan.

Gambar 4.16 Tampilan di display *dot matrik*

Setelah itu hasil ditampilkan pada dot matrix display, sesuai dengan pesan yang diinputkan pada saat mode *input* dipilih. Gambar 4.16 menunjukkan hasil tampilan di *dot matrix display*.

Dengan demikian, pengujian alat dan program secara keseluruhan dapat bekerja dengan baik dan sesuai dengan yang diharapkan.

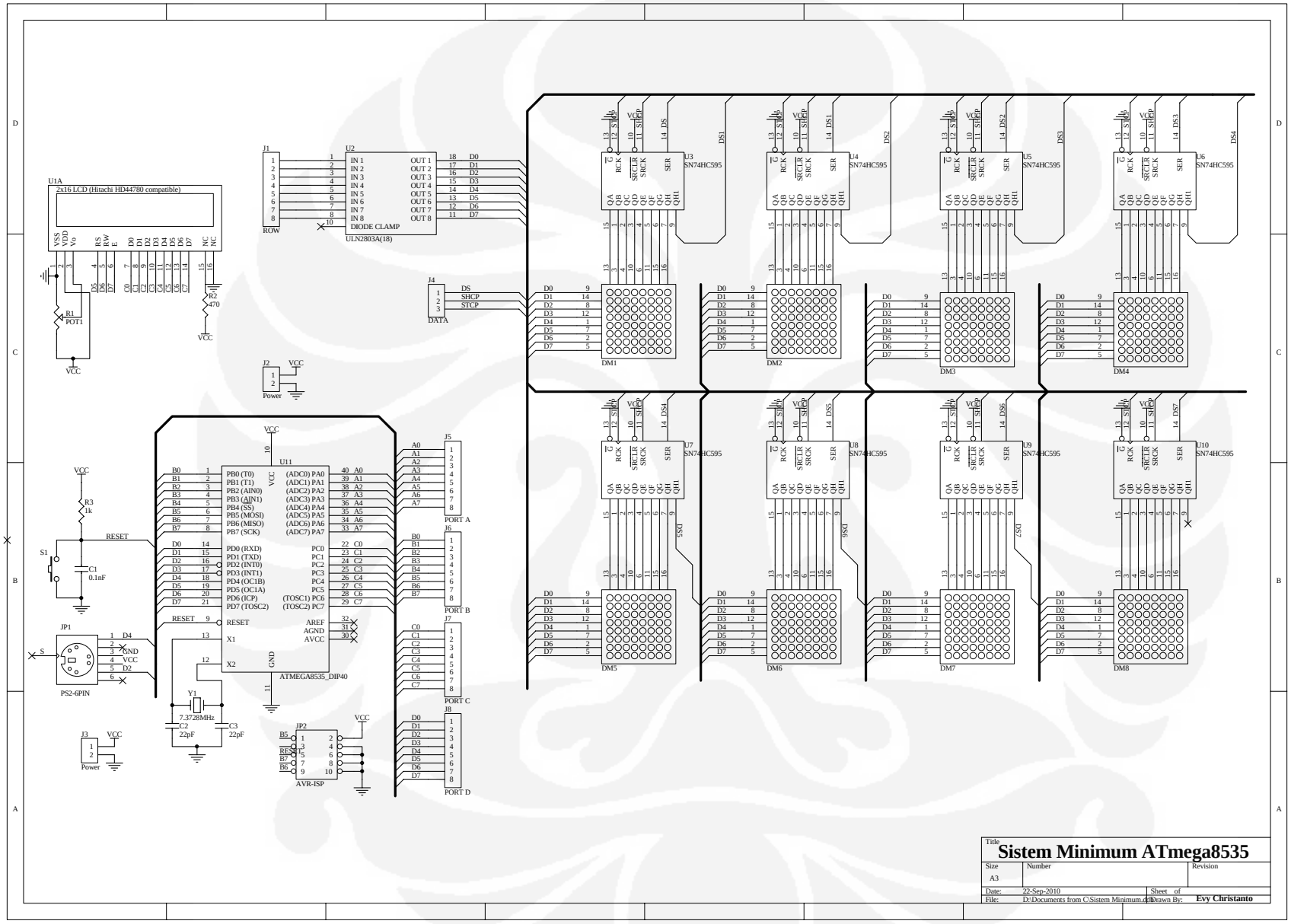
BAB V
KESIMPULAN

Setelah melakukan rancangan bangun dan pengujian papan informasi elektronik dengan PS2 *keyboard* dapat disimpulkan bahwa :

1. Telah berhasil dilakukan rancang bangun papan informasi elektronik yang *compact*, mudah dioperasikan dan ekonomis yang didesain khusus untuk kebutuhan produksi PT Sanken Indonesia.
2. Papan pesan elektronik ini membutuhkan daya 1,25 watt, mampu menampilkan karakter ke LCD dari *keyboard* dengan kecepatan 95ms.
3. Waktu yang dibutuhkan untuk menampilkan tulisan ke display dot-matrik sekitar 85,2ms sedangkan untuk *looping* pemunculan karakter sekitar 3.06s.

DAFTAR ACUAN

1. Andrianto, Heri. (2008). *Pemrograman Mikrokontroler AVR ATmega16 Menggunakan Bahasa C (CodeVision AVR)*. Bandung: Informatika.
2. Pappas, C.H., & Murray, W.H. *Borland C++ Handbook* (3rd ed.). Osborne: McGraw-Hill.
3. Glibota, Zvonko. (2004). *DIY Moving Message Display*. January 3, 2004. http://www.edaboard.com/ftopic58756.html/
4. Hartanto, Budi. (2007). *Memahami Logika Pembuatn Program C Secara Mudah*. Yogyakarta: Andi Offset.
5. Soebhakti, Hendawan. (2007). *Basic AVR Microcontroller Tutorial ATmega8535L*. Batam: Politeknik Batam.
6. Haiduc, Pavel. (2008). *CodeVisionAVR Version 2.03.4 User Manual*. HP InfoTech
7. http://www.dipmicro.com/store/LCD-1602A-B
8. Datasheet LCD HD744840
9. Datasheet *Keyboard*
10. Datasheet *Dot Matrix*
11. http://indonetwork.or.id/alloffers/LED-MATRIX.html
12. http://home.iae.nl/users/pouweha/lcd/lcd0.shtml#pin_assignment


Title
Sistem Minimum ATmega8535
Size
A3
Number
Revision
Date: 22-Sep-2010
Sheet of
File: D:\Documents from C\Sistem Minimum.ddb
Drawn By: Evy Christanto
2x16 LCD (Hitachi HD44780 compatible)
ROW
DIODE CLAMP
ULN2803A(18)
DATA
Power
PS2-6PIN
ATMEGA8535_DIP40
7.3728MHz
AVR-ISP
PORT A
PORT B
PORT C
PORT D
SN74HC595
DM1
DM2
DM3
DM4
DM5
DM6
DM7
DM8

Listing Program

```
// PAPAN INFORMASI ELEKTRONIK DENGAN PS/2 KEYBOARD
SEBAGAI INPUT

#include <mega8535.h>
#include <delay.h>
#include <matrix.h>

//referensi keyboard
#define KBD_CLOCK PORTD.2
#define KBD_DATA PIND.4
#define BUFF_SIZE 64

// referensi LCD
#define LCD_RS  PORTD.5
#define LCD_RW  PORTD.6
#define LCD_E   PORTD.7
#define COMMAND_MODE    0
#define DATA_MODE       1
#define WRITE_MODE      0
#define READ_MODE       1
#define LCD_PORT        PORTC

// LED Display
#define SCROLL_DELAY    140             // Konstanta Kecepatan

// fungsi-fungsi
void lcd_init(void);
void lcd_clear(void);
void lcd_putc(unsigned char ch);
void lcd_putsf(unsigned char flash *);
void lcd_puts(unsigned char *str);
void put_char_kbbuff(unsigned char);
void put_scancode_kbbuff(unsigned char);
int getchar(void);
int kbhit(void);
void decode(unsigned char);
void enable(void); // fungsi untuk enable LCD
void goto_address(unsigned char a);
void display_char(unsigned char data);
void welcome_message(void);
void input_mode(void);
void menu_display(void);
```

```
void wrong_option(void);
void displaying_message(void);
void display_matrix(void);

// variabel global
unsigned char bitcount;
unsigned char kb_buffer[BUFF_SIZE];
unsigned char buffcnt = 0;
unsigned char *inpt, *outpt;
unsigned char chcount;     // Character counter
// char message_matrix[];
// unsigned int matrix_content;
// int char_count_disp, col_matrix;
unsigned char message[100];
```

```
eeprom unsigned char message_matrix[100];
eeprom unsigned char last_address;



// keyboard scan codes (without & with shift key pressed)
flash unsigned char unshifted[67][2] = {
//0x0d,9,
0x0e,'`',0x15,'q',0x16,'1',0x1a,'z',0x1b,'s',0x1c,'a',0x1d,'w',0x1e,'2',0x21,'c',0x22,'x'
,0x23,'d',0x24,'e',
0x25,'4',0x26,'3',0x29,'
',0x2a,'v',0x2b,'f',0x2c,'t',0x2d,'r',0x2e,'5',0x31,'n',0x32,'b',0x33,'h',0x34,'g',
0x35,'y',0x36,'6',0x39,',',0x3a,'m',0x3b,'j',0x3c,'u',0x3d,'7',0x3e,'8',0x41,',',0x42,'k',
0x43,'i',0x44,'o',
0x45,'0',0x46,'9',0x49,'.',0x4a,'/',0x4b,'l',0x4c,';',0x4d,'p',0x4e,'-
',0x52,'`',0x54,'[',0x55,'=',0x5a,13,
0x5b,']',0x5d,'/',0x61,'<',0x66,8,
0x69,'1',0x6b,'4',0x6c,'7',0x70,'0',0x71,',',0x72,'2',0x73,'5',0x74,'6',
0x75,'8',0x79,'+',0x7a,'3',0x7b,'-',0x7c,'*',0x7d,'9',0,0 };


flash unsigned char shifted[67][2] = {
//0x0d,9,
0x0e,'`',0x15,'Q',0x16,'!',0x1a,'Z',0x1b,'S',0x1c,'A',0x1d,'W',0x1e,'@',0x21,'C',0x2
2,'X',0x23,'D',0x24,'E',
0x25,'$',0x26,'#',0x29,'
',0x2a,'V',0x2b,'F',0x2c,'T',0x2d,'R',0x2e,'%',0x31,'N',0x32,'B',0x33,'H',0x34,'G',
0x35,'Y',0x36,'^',0x39,'L',0x3a,'M',0x3b,'J',0x3c,'U',0x3d,'&',0x3e,'*',0x41,'<',0x4
2,'K',0x43,'I',0x44,'O',
0x45,')',0x46,'(',0x49,'>',0x4a,'?',0x4b,'L',0x4c,':',0x4d,'P',0x4e,'_',0x52,'"',0x54,'{',
0x55,'+',0x5a,13,
0x5b,'}',0x5d,'|',0x61,'>',0x66,8,
0x69,'1',0x6b,'4',0x6c,'7',0x70,'0',0x71,',',0x72,'2',0x73,'5',0x74,'6',
0x75,'8',0x79,'+',0x7a,'3',0x7b,'-',0x7c,'*',0x7d,'9',0,0 };

// fungsi inisialiasi LCD
void lcd_init(void)
{
 delay_ms(50);
 LCD_E = 0;
 LCD_RS = COMMAND_MODE;
 LCD_RW = WRITE_MODE;
 delay_ms(1);
 LCD_PORT = 0x38;       // jalur data 8 bit, 2 lines, huruf ukuran 5x8  ==>
function set
 enable();
 LCD_PORT = 0x0f;       // display ON, kursor ON, kursor berkedip
 enable();
```

```
 LCD_PORT = 0x06;       // entry mode, increment, not shifted
 enable();
 lcd_clear();
}

// fungsi menghapus layar LCD
void lcd_clear(void)
{
 LCD_RS = COMMAND_MODE;
 LCD_PORT = 0x01;       // clear LCD
 enable();
}

void enable(void)
{
 LCD_E = 1; delay_ms(1);
 LCD_E = 0; delay_ms(2);
}

//Rutin service interupsi untuk eksternal interrupt 0 (EXT_INT0)
interrupt [2] void keyboard_isr(void)
{
 static unsigned char data;     // Rutin entered at falling edge
 // fungsi dipanggil pada saat sinyal KBD_CLOCK transisi turun
 // jika data bit adalah bit berikutnya yang akan dibaca
 // (bit 3 sampai 10 adalah data, start, stop & parity bis diabaikan
 if((bitcount < 11) && (bitcount > 2))
 {
  data = (data >> 1);
  if (KBD_DATA)                         // jika bit berikutnya adalah 1
     data = data | 0x80;  // simpan bit '1'
  else
     data = data & 0x7f;        // jika tidak simpan bit '0'
 }
 if(--bitcount == 0)
 {                              // Apakah semua bit sudah diterima ?
  decode(data);                 // decode byte yang diterima
  bitcount = 11;
 }
}

//***********************************************
// return 1 if a key is pressed (non blocking)
// else return 0
int kbhit(void)
{
 if (buffcnt)
 {
```

```
 // reset buffer variables (flush the buffer)
 inpt = kb_buffer;
 outpt = kb_buffer;
 buffcnt = 0;
 bitcount = 11;
 return 1;
 }
 return 0;
}

// puts scan code (in hex format) into keyboard buffer
// (used for debugging purposes)
//************************************************
void put_scancode_kbbuff(unsigned char sc)
{
 unsigned char h,l;

 // convert hi and low nibbles of the scancode
 // into ascii and store them into keyboard buffer
 h =  ((sc & 0xf0 ) >> 0x04) & 0x0f;
 if ( h > 9)
    h = h + 7;
    h = h + 0x30;
    put_char_kbbuff(h);

    l = sc & 0x0f;
 if ( l > 9)
    l = l + 7;
    l = l + 0x30;
    put_char_kbbuff(l);
}

//************************************************
// store character in the keyboard ring buffer
void put_char_kbbuff(unsigned char c) {
        if (buffcnt < BUFF_SIZE) { // if buffer is not full
                *(inpt++) = c;
                buffcnt++;
                if (inpt >= kb_buffer + BUFF_SIZE) // pointer wrapping
                        inpt = kb_buffer;
        }
}

//************************************************
// get next available character from the keyboard ring buffer
// (waits until a character is available in the buffer)
int getchar(void) {
        int byte;
```

```
        while (buffcnt == 0); // wait for data
        byte = *outpt; // get byte
        outpt++;
        if (outpt >= kb_buffer + BUFF_SIZE) // pointer wrapping
                outpt = kb_buffer;
        buffcnt--; // decrement buffer count
        return byte;
}

//************************************************
// decode scan code
void decode(unsigned char sc) {
        static unsigned char is_up=0, shift = 0, mode = 0;
        unsigned char i;

        if (!is_up) {
                switch (sc) {
                        case 0xF0 :// The up-key identifier
                                is_up = 1;
                                break;
                        case 0x12 :// Left SHIFT
                                shift = 1;
                                break;
                        case 0x59 :// Right SHIFT
                                shift = 1;
                                break;
                        case 0x05 :// F1
                                if(mode == 0)
                                        mode = 1;// Enter scan code mode
                                if(mode == 2)
                                        mode = 3;// Leave scan code mode
                                break;
                        default:
                                if(mode == 0 || mode == 3) {// If ASCII mode

                                        if(!shift) {// If shift not pressed, do a table
look-up
                                                for(i = 0; unshifted[i][0]!=sc &&
unshifted[i][0]; i++);
                                                        if (unshifted[i][0] == sc) {

        put_char_kbbuff(unshifted[i][1]);

                                                        }

                                        }
                                        else {// If shift pressed
```

```
                                        for(i = 0; shifted[i][0]!=sc &&
shifted[i][0]; i++);
                                                if (shifted[i][0] == sc) {

        put_char_kbbuff(shifted[i][1]);
                                                }
                                        }
                                }
                                else put_scancode_kbbuff(sc);          // scan code
mode (debugging mode)
                                break;
                        }
                }
                else {
                        is_up = 0;// Two 0xF0 in a row not allowed
                        switch (sc) {
                                case 0x12 :// Left SHIFT
                                        shift = 0;
                                        break;
                                case 0x59 :// Right SHIFT
                                        shift = 0;
                                        break;
                                case 0x05 :// F1 --  F1 puts you in debugging mode
                                          // pressing F1 again gets you out of debugging
mode
                                          // in debugging mode hex code of the scan codes
                                          // are stored in the buffer instead of their ascii
codes
                                        if(mode == 1)
                                        mode = 2;
                                        if(mode == 3)
                                                mode = 0;
                                        break;
                        }
                }
}

void goto_address(unsigned char a)
{
 LCD_RS = COMMAND_MODE;
 LCD_RW = WRITE_MODE;
 if (a>15)
   a = a + 0x30;
 LCD_PORT = a + 0x80;
 LCD_E = 1; delay_ms(1);
 LCD_E = 0; delay_ms(2);
}
```

```
void display_char(unsigned char data)
{
 LCD_RS = DATA_MODE;
 LCD_RW = WRITE_MODE;
 LCD_PORT = data;
 enable();
}

//***********************************************
// display char on lcd
void lcd_putc(unsigned char ch)
{
 LCD_RS = DATA_MODE;
 LCD_PORT = ch;
 LCD_E = 1;   delay_us(40);
 LCD_E = 0;   delay_us(40);
}
//***********************************************
// display string on lcd
// input: pointer to string stored in flash ROM
void lcd_putsf(unsigned char flash *str)
{
 while (*str!= '\0')
 {
  lcd_putc(*(str++));
 }
}

//***********************************************
//  same as above except the string is in RAM
void lcd_puts(unsigned char *str)
{
 while (*str!= '\0')
 {
  lcd_putc(*(str++));
 }
}

//***********************************************
// Welcome Message
void welcome_message(void)
{
 unsigned char welcome1[] = "INFO BOARD";
 unsigned char welcome2[] = "By:Evy.CH";
 unsigned char i, j;

 i = 0;
 goto_address(1);
```

```
 for (j=0; j<13; j++)
 {
  display_char(welcome1[i++]);
 }
 i = 0;
 goto_address(23);
 for (j=0; j<9; j++)
 {
  display_char(welcome2[i++]);
 }
 delay_ms(5000);
 lcd_clear();
}

//*************************************************
// Display Menu
void menu_display(void)
{
 unsigned char menu1[] = "1:Input Mode";
 unsigned char menu2[] = "2:Read EEPROM";
 unsigned char i, j;

 LCD_RS = COMMAND_MODE;
 LCD_RW = WRITE_MODE;
 LCD_PORT = 0x0c;
 enable();

 i = 0;
 goto_address(0);
 for (j=0; j<12; j++)
 {
  display_char(menu1[i++]);
 }
 i = 0;
 goto_address(16);
 for (j=0; j<13; j++)
 {
  display_char(menu2[i++]);
 }
}

void wrong_option(void)
{
 unsigned char wrong[] = "Sorry,";
 unsigned char wrong1[] = "Wrong Option...";
 unsigned char i, j;

 LCD_RS = COMMAND_MODE;
```

```
 LCD_RW = WRITE_MODE;
 LCD_PORT = 0x0c;
 enable();

 lcd_clear();

 i = 0;
 goto_address(0);
 for (j=0; j<6; j++)
 {
  display_char(wrong[i++]);
 }
 i = 0;
 goto_address(16);
 for (j=0; j<15; j++)
 {
  display_char(wrong1[i++]);
 }
 delay_ms(2000);
}

void displaying_message(void)
{
 unsigned char display[] = "Displaying";
 unsigned char display1[] = "Message.....";
 unsigned char i;

 LCD_RS = COMMAND_MODE;
 LCD_RW = WRITE_MODE;
 LCD_PORT = 0x0c;
 enable();

 lcd_clear();

 // i = 0;
 goto_address(0);
 for (i=0; i<10; i++)
 {
  display_char(display[i]);
 }
 // i = 0;
 goto_address(16);
 for (i=0; i<13; i++)
 {
  display_char(display1[i]);
 }
}
```

```
void input_mode(void)
{
 // unsigned char message[100];
 unsigned char ch, b, c, x, y, d;
 unsigned char i, j;

 lcd_init();

 while(1)
 {
  ch = 0x00;
  chcount = 0;
  i = 0;
  b = 0;
  x = 0;
  d = 0;

  // fill the array with 4 empty spaces at the start and the end of the message
  // as the screen has 4 chars
  // (it looks better if we start scrolling with empty screen and end
  // with empty screen , rather than displaying four chars immediatelly and
  // then start scrolling)
  // message[i++] = ' ';
  // message[i++] = ' ';
  // message[i++] = ' ';
  // message[i++] = ' ';

  // enter message until 300 characters entered or <enter> key pressed
  while((ch != 13) || (i < 100))
  {
   ch = getchar();
   if (ch != 13)
   {
    if (ch == 8)
    {
     if (chcount == 0)
     {
      lcd_clear();
      x = 0;
      i = 0;
     }
     else if (chcount>32)
     {
      y = 0;
      message[--i] = ' ';
      for (c=33;c>=2;c--)
      {
       goto_address(y);
```

```
   display_char(message[chcount-c]);
   y++;
  }
  chcount--;
  x = chcount;
  --b;
 }
 else
 {
  message[--i] = ' ';
  goto_address(chcount-1);
  display_char(message[i]);
  goto_address(chcount-1);
  x--;
  chcount--;
 }
}

else

if (chcount<32)
{
 goto_address(x);
 message[i++] = ch;
 display_char(message[--i]);
 i++;
 x++;
 chcount++;
}
else
{
 x = 0;
 b++;
 i = b;
 for (d=0;d<31;d++)
 {
  goto_address(x);
  display_char(message[i++]);
  x++;
 }
 goto_address(x);
 message[i++] = ch;
 display_char(message[--i]);
 i++;
 //delay_ms(10);
 chcount++;
}
}
```

```
  // if q is pressed clear the LCD
  // (code only writes to the first LCD line so
  // if the line is longer than 16 chars i cant see the rest of the chars
  // , there will be a menu system in the future software version
  // and the line on the LCD will be scrolled
  // to the left automatically as the cursor gets to the end of the line
  //, for now use this dirty solution:))
  if (ch == 13)
  {
   i = 0;
   message_matrix[i++] = ' ';
   message_matrix[i++] = ' ';
   message_matrix[i++] = ' ';
   message_matrix[i++] = ' ';
   for (j=0; j<chcount; j++)
   {
    message_matrix[i++] = message[j];
   }
   message_matrix[i++] = ' ';
   message_matrix[i++] = ' ';
   message_matrix[i++] = ' ';
   message_matrix[i++] = ' ';

   last_address = i;

   displaying_message();

   display_matrix();


  }
 }
 // message[i++] = ' ';
 // message[i++] = ' ';
 // message[i++] = ' ';
 // message[i++] = ' ';
 }
}

void display_matrix(void)
{
 unsigned char row, i, currentChar, charOffset, tmpCurrentChar, tmpCharOffset,
ledArray[20], messageLength;
 flash unsigned char *tmpDataPtr, *dataPtr;
 unsigned int j;

 messageLength = last_address - 4;
```

```
charOffset = 0;
currentChar = 0;
dataPtr = &led_chars[message_matrix[0] - 0x20][0];

while (!kbhit())
{
 tmpDataPtr = dataPtr;
 tmpCharOffset = charOffset;
 tmpCurrentChar = currentChar;
 for (i = 0; i<=19 ; i++)
 {
  ledArray[i] = *tmpDataPtr++;
  if (++tmpCharOffset==0x06)
  {
   tmpCharOffset = 0;
   if (++tmpCurrentChar == messageLength)
         tmpCurrentChar = 0;
         tmpDataPtr = &led_chars[message_matrix[tmpCurrentChar] - 0x20][0];
  }
 }
 if (++charOffset == 0x06)
 {
  charOffset = 0;
  if (++currentChar == messageLength)
     currentChar = 0;
 }
 dataPtr = &led_chars[message_matrix[currentChar] - 0x20][charOffset];

 row = 0x02;
 for (j=0; j<SCROLL_DELAY; j++)
 {
  for (i=0; i<=19 ; i++)
  {
   PORTB.2 = (ledArray[i] & row) ? 1 : 0;
   PORTB.3 = 0;
   PORTB.4 = 1;
   PORTB.3 = 1;
   PORTB.4 = 0;
  }
  PORTA = row;
  row <<= 1;
  if (!row)
   row = 0x02;
  delay_us(800);
  PORTA = 0x00;
 }
}
}
```

```
// Program Utama
void main(void)
{
 unsigned char i;

 #asm("cli");              // disable global interrupt

 // inisialisasi port
 DDRA = 0xff;
 DDRB = 0xff;
 DDRC = 0xff;            // PORTC sebagai output
 DDRD = 0xeb;            // PORTC sebagai output kecuali PIND.2 & PIND.4
sebagai input

 PORTA = 0x00;
 PORTB = 0x00;
 PORTC = 0x00;
 PORTD = 0X00;

 // inisialisasi usart
 UCSRA=0x00;
 UCSRB=0xD8;
 UCSRC=0x86;
 UBRRH=0x00;
 UBRRL=0x47;

 // inisialisasi keyboard
 inpt = kb_buffer;
 outpt = kb_buffer;
 buffcnt = 0;
 bitcount = 11;

 // inisialisasi LCD
 lcd_init();

 // inisialisasi register interupsi
 MCUCR = 0x02;          // interrupsi pada INT0 dengan transisi turun (falling
edge)
 GICR = 0x40;           // interrupsi INT0 enable

 #asm("sei");              // enable global interrupt

 // welcome message
 welcome_message();

 while(1)
```

```
{
 lcd_clear();
 menu_display();

 i = getchar();

 if (i == '1')
 {
  lcd_clear();
  input_mode();
 }
 else if (i == '2')
 {
  displaying_message();
  display_matrix();
 }
 else
  wrong_option();
}
}
```

Pustaka <matrix.h>

```
flash unsigned char led_chars[94][6] = {
0x00,0x00,0x00,0x00,0x00,0x00,        // space
0x00,0x00,0xfa,0x00,0x00,0x00,        // !
0x00,0xe0,0x00,0xe0,0x00,0x00,        // "
0x28,0xfe,0x28,0xfe,0x28,0x00,        // #
0x24,0x54,0xfe,0x54,0x48,0x00,  // $
0xc4,0xc8,0x10,0x26,0x46,0x00,  // %
0x6c,0x92,0xaa,0x44,0x0a,0x00,  // &
0x00,0xa0,0xc0,0x00,0x00,0x00,  // '
0x00,0x38,0x44,0x82,0x00,0x00,        // (
0x00,0x82,0x44,0x38,0x00,0x00,  // )
0x28,0x10,0x7c,0x10,0x28,0x00,  // *
0x10,0x10,0x7c,0x10,0x10,0x00,  // +
0x00,0x0a,0x0c,0x00,0x00,0x00,  // ,
0x10,0x10,0x10,0x10,0x10,0x00,  // -
0x00,0x06,0x06,0x00,0x00,0x00,  // .
0x04,0x08,0x10,0x20,0x40,0x00,  // /
0x7c,0x8a,0x92,0xa2,0x7c,0x00,  // 0
0x00,0x42,0xfe,0x02,0x00,0x00,  // 1
0x42,0x86,0x8a,0x92,0x62,0x00,  // 2
0x84,0x82,0xa2,0xd2,0x8c,0x00,        // 3
0x18,0x28,0x48,0xfe,0x08,0x00,        // 4
0xe5,0xa2,0xa2,0xa2,0x9c,0x00,        // 5
0x3c,0x52,0x92,0x92,0x0c,0x00,        // 6
0x80,0x8e,0x90,0xa0,0xc0,0x00,        // 7
0x6c,0x92,0x92,0x92,0x6c,0x00,        // 8
0x60,0x92,0x92,0x94,0x78,0x00,        // 9
0x00,0x6c,0x6c,0x00,0x00,0x00,        // :
0x00,0x6a,0x6c,0x00,0x00,0x00,        // ;
0x10,0x28,0x44,0x82,0x00,0x00,        // <
0x28,0x28,0x28,0x28,0x28,0x00,        // =
0x00,0x82,0x44,0x28,0x10,0x00,        // >
0x40,0x80,0x8a,0x90,0x60,0x00,        // ?
0x4c,0x92,0x9e,0x82,0x7c,0x00,        // @
0x7e,0x88,0x88,0x88,0x7e,0x00,        // A
0xfe,0x92,0x92,0x92,0x6c,0x00,        // B
0x7c,0x82,0x82,0x82,0x44,0x00,        // C
0xfe,0x82,0x82,0x44,0x38,0x00,        // D
0xfe,0x92,0x92,0x92,0x82,0x00,        // E
```

```
0xfe,0x90,0x90,0x90,0x80,0x00,        // F
0x7c,0x82,0x92,0x92,0x5e,0x00,        // G
0xfe,0x10,0x10,0x10,0xfe,0x00,        // H
0x00,0x82,0xfe,0x82,0x00,0x00,        // I
0x04,0x02,0x82,0xfc,0x80,0x00,        // J
0xfe,0x10,0x28,0x44,0x82,0x00,        // K
0xfe,0x02,0x02,0x02,0x02,0x00,        // L
0xfe,0x40,0x30,0x40,0xfe,0x00,        // M
0xfe,0x20,0x10,0x08,0xfe,0x00,        // N
0x7c,0x82,0x82,0x82,0x7c,0x00,        // O
0xfe,0x90,0x90,0x90,0x60,0x00,        // P
0x7c,0x82,0x8a,0x84,0x7a,0x00,        // Q
0xfe,0x90,0x98,0x94,0x62,0x00,        // R
0x62,0x92,0x92,0x92,0x8c,0x00,        // S
0x80,0x80,0xfe,0x80,0x80,0x00,        // T
0xfc,0x02,0x02,0x02,0xfc,0x00,        // U
0xf8,0x04,0x02,0x04,0xf8,0x00,        // V
0xfc,0x02,0x1c,0x02,0xfc,0x00,        // W
0xc6,0x28,0x10,0x28,0xc6,0x00,        // X
0xe0,0x10,0x0e,0x10,0xe0,0x00,        // Y
0x86,0x8b,0x92,0xa2,0xc2,0x00,        // Z
0x00,0xfe,0x82,0x82,0x00,0x00,        // [
0x00,0x00,0x00,0x00,0x00,0x00,        // *** do not remove this empty char ***
0x00,0x82,0x82,0xfe,0x00,0x00,        // ]
0x20,0x40,0x80,0x40,0x20,0x00,        // ^
0x02,0x02,0x02,0x02,0x02,0x00,        // _
0x00,0x80,0x40,0x20,0x00,0x00,        // `
0x04,0x2a,0x2a,0x2a,0x1e,0x00,        // a
0xfe,0x12,0x22,0x22,0x1c,0x00,        // b
0x1c,0x22,0x22,0x22,0x04,0x00,        // c
0x1c,0x22,0x22,0x12,0xfe,0x00,        // d
0x1c,0x2a,0x2a,0x2a,0x18,0x00,        // e
0x10,0x7e,0x90,0x80,0x40,0x00,        // f
0x30,0x4a,0x4a,0x4a,0x7c,0x00,        // g
0xfe,0x10,0x20,0x20,0x1e,0x00,        // h
0x00,0x22,0xbe,0x02,0x00,0x00,        // i
0x04,0x02,0x22,0xbc,0x00,0x00,        // j
0xfe,0x08,0x14,0x22,0x00,0x00,        // k
0x00,0x82,0xfe,0x02,0x00,0x00,        // l
0x3e,0x20,0x18,0x20,0x1e,0x00,        // m
0x3e,0x10,0x20,0x20,0x1e,0x00,        // n
0x1c,0x22,0x22,0x22,0x1c,0x00,        // o
0x3e,0x28,0x28,0x28,0x10,0x00,        // p
0x10,0x28,0x28,0x18,0x3e,0x00,        // q
0x3e,0x10,0x20,0x20,0x10,0x00,        // r
0x12,0x2a,0x2a,0x2a,0x04,0x00,        // s
0x20,0xfc,0x22,0x02,0x04,0x00,        // t
0x3c,0x02,0x02,0x04,0x3e,0x00,        // u
```

```
0x38,0x04,0x02,0x04,0x38,0x00,        // v
0x3c,0x02,0x0c,0x02,0x3c,0x00,        // w
0x22,0x14,0x08,0x14,0x22,0x00,        // x
0x30,0x0a,0x0a,0x0a,0x3c,0x00,       // y
0x22,0x26,0x2a,0x32,0x22,0x00,         // z
0x00,0x10,0x6c,0x82,0x00,0x00,        // {
0x00,0x00,0xfe,0x00,0x00,0x00,       // |
0x00,0x82,0x6c,0x10,0x00,0x00 };
```

Features

- High-performance, Low-power AVR® 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® Microcontroller with 8K Bytes In-System Programmable Flash

ATmega8535
ATmega8535L

Advance Information

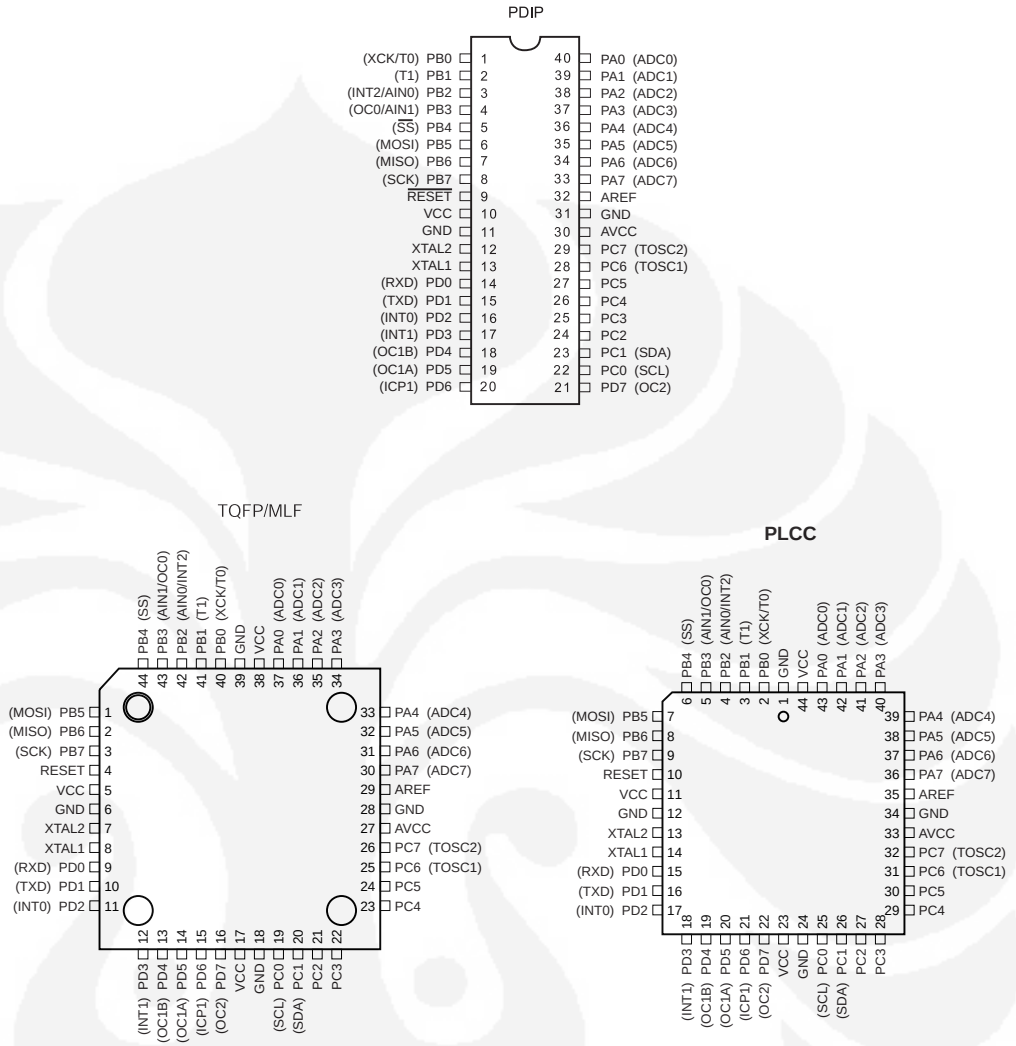
Summary

Rev. 2502CS–AVR–04/03

Note: This is a summary document. A complete document is available on our web site at *www.atmel.com*.

Pin Configurations

Figure 1. Pinout ATmega8535



Disclaimer

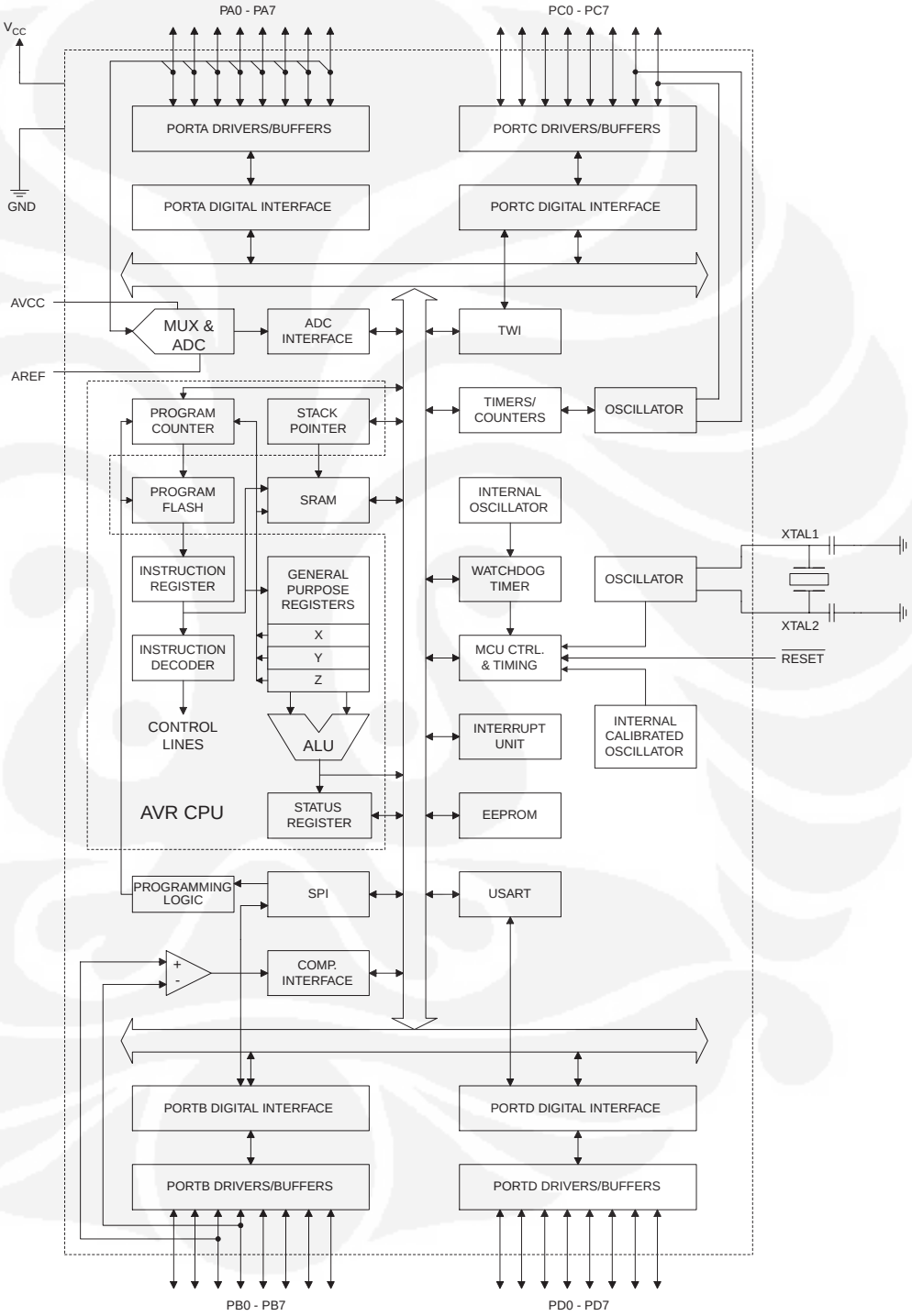
Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

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

Figure 2. Block Diagram



AVR313: Interfacing the PC AT Keyboard

8-bit AVR® Microcontroller

Application Note

Features

- **Interfacing Standard PC AT Keyboards**
- **Requires Only Two I/O Pins. One of them must be an External Interrupt Pin**
- **No Extra Hardware Required**
- **Complete Example in C, Implementing a Keyboard to Serial Converter**

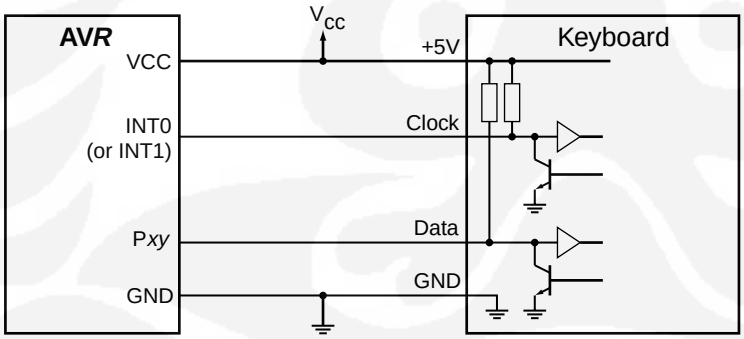
Introduction

Most microcontrollers requires some kind of a human interface. This application note describes one way of doing this using a standard PC AT keyboard.

The Physical Interface

The physical interface between the keyboard and the host is shown in Figure 1. Two signal lines are used, clock and data. The signal lines are open connector, with pullup resistors located in the keyboard. This allows either the keyboard or the host system to force a line to low level. Two connector types are available, the 5-pin DIN connector of “5D” type, and the smaller six-pin mini-DIN. The pin assignments are shown in Table 1.

Figure 1. The Interface.



Rev. 1235B–AVR–05/02

Table 1. AT Keyboard Connector Pin Assignments

AT Computer		
Signals	**DIN41524, Female at Computer, 5-pin DIN 180°**	**6-pin Mini DIN PS2 Style Female at Computer**
Clock	1	5
Data	2	1
nc	3	2,6
GND	4	3
+5V	5	4
Shield	Shell	Shell

Timing

The timing for the data transferred from the keyboard to the host is shown in Figure 2. The protocol is: one start bit (always 0), eight data bits, one odd parity bit and one stop bit (always 1). The data is valid during the low period of the clock pulse. The keyboard is generating the clock signal, and the clock pulses are typically 30-50 µs low and 30-50 µs high.

The host system can send commands to the keyboard by forcing the clock line low. It then pulls the data line low (the start bit). Now, the clock line must be released. The keyboard will count 10 clock pulses. The data line must be set up to the right level by the host before the trailing edge of the clock pulse. After the tenth bit, the keyboard checks for a high level on the data line (the stop bit), and if it is high, it forces it low. This tells the host that the data is received by the keyboard. The software in this design note will not send any commands to the keyboard.

Scan Codes

The AT keyboard has a scan code associated with each key. When a key is pressed, this code is transmitted. If a key is held down for a while, it starts repeating. The repeat rate is typically 10 per second. When a key is released, a “break” code ($F0) is transmitted followed by the key scan code. For most of the keys, the scan code is one byte. Some keys like the *Home*, *Insert* and *Delete* keys have an extended scan code, from two to five bytes. The first byte is always $E0. This is also true for the “break” sequence, e.g., E0 F0 xx…

AT keyboards are capable of handling three sets of scan codes, where set two is default. This example will only use set two.

The Software

The code supplied with this application note is a simple keyboard to RS-232 interface. The scan codes received from the keyboard are translated into appropriate ASCII characters and transmitted by the UART. The source code is written in C, and is easily modified and adaptable to all AVR microconrollers with SRAM.

Note: The linkerfile (AVR313.xcl) included in the software archive has to be included instead of the standard linkerfile. This is done from the include menu under XLINK – Options. The linker file applies to AT90S8515 only.

The Algorithm

Keyboard reception is handled by the interrupt function **INT0_interrupt**. The reception will operate independent of the rest of the program.

The algorithm is quite simple: Store the value of the data line at the leading edge of the clock pulse. This is easily handled if the clock line is connected to the INT0 or INT1 pin. The interrupt function will be executed at every edge of the clock cycle, and data will be stored at the falling edge. After all bits are received, the data can be decoded. This is done by calling the **decode** function. For character keys, this function will store an ASCII character in a buffer. It will take into account if the shift key is held down when a key is pressed. Other keys like function keys, navigation keys (arrow keys, page up/down keys etc.) and modifier keys like Ctrl and Alt are ignored.

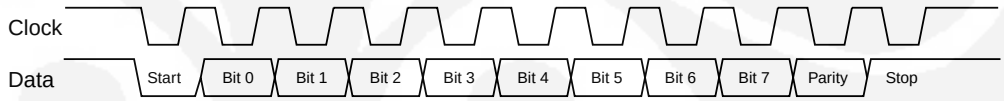
The mapping from scan codes to ascii characters are handled with table look-ups, one table for shifted characters and one for un-shifted.

Modifications and Improvements

If the host falls out of sync with the keyboard, all subsequent data received will be wrong. One way to solve this is to use a time out. If 11 bits are not received within 1.5 ms, some error have occurred. The bit counter should be reset and the faulty data discarded.

If keyboard parameters like typematic rate and delay are to be set, data must be sent to the keyboard. This can be done as described earlier. For the commands, see the keyboard manufacturer's specifications.

Figure 2. Timing for Keyboard to Host Transfer



Main.c

```
#include <pgmspace.h>
#include <stdio.h>
#include <stdlib.h>
#include "io8515.h"

#include "serial.h"
#include "gpr.h"
#include "kb.h"

void main(void)
{
  unsigned char key;

  init_uart();                  // Initializes the UART transmit buffer
  init_kb();                    // Initialize keyboard reception

  while(1)
  {
    key=getchar();
    putchar(key);
    delay(100);
  }
}
```

Low_level_init.c

```
#include <ina90.h>
#include <io8515.h>

int __low_level_init(void)
{
    UBRR = 12;          // 19200bps @ 4 MHz
    UCR  = 0x08;        // TX enable
    GIMSK= 0x40;        // Enable INT0 interrupt

    _SEI();
    return 1;
}
```

Serial.c

```
#include <stdio.h>
#include <pgmspace.h>
#include <io8515.h>                 /* SFR declarations */
#include "serial.h"

#define ESC 0x1b
#define BUFF_SIZE 64

flash char CLR[] = {ESC, '[','H', ESC, '[', '2', 'J',0};

unsigned char UART_buffer[BUFF_SIZE];
unsigned char *inptr, *outptr;
unsigned char buff_cnt;

void init_uart(void)
{
    inptr =  UART_buffer;
    outptr = UART_buffer;
    buff_cnt = 0;
}

void clr(void)
{
    puts_P(CLR);                            // Send a 'clear screen' to a
VT100 terminal
}

int putchar(int c)
{
    if (buff_cnt<BUFF_SIZE)
    {
        *inptr = c;                         // Put character into buffer
        inptr++;                            // Increment pointer

        buff_cnt++;

        if (inptr >= UART_buffer + BUFF_SIZE)   // Pointer wrapping
```

```
            inptr = UART_buffer;

        UCR = 0x28;                             // Enable UART Data register
                                                // empty interrupt

        return 1;
    } else {
        return 0;                               // Buffer is full
    }

}

   // Interrupt driven transmitter

interrupt [UART_UDRE_vect] void UART_UDRE_interrupt(void)
{
    UDR = *outptr;                              // Send next byte
    outptr++;                                   // Increment pointer

    if (outptr >= UART_buffer + BUFF_SIZE)      // Pointer wrapping
        outptr = UART_buffer;

    if(--buff_cnt == 0)                         // If buffer is empty:
        UCR = UCR && (1<<UDRIE);                //    disabled interrupt
}
```

Kb.c

```
#include <pgmspace.h>
#include "kb.h"
#include "serial.h"
#include "gpr.h"

#include "scancodes.h"

#define BUFF_SIZE 64

unsigned char edge, bitcount;// 0 = neg.  1 = pos.

unsigned char kb_buffer[BUFF_SIZE];
unsigned char *inpt, *outpt;
unsigned char buffcnt;


void init_kb(void)
{
    inpt =  kb_buffer;// Initialize buffer
    outpt = kb_buffer;
    buffcnt = 0;

  MCUCR = 2; // INT0 interrupt on falling edge
```

```
  edge = 0;  // 0 = falling edge  1 = rising edge
  bitcount = 11;
}

interrupt [INT0_vect] void INT0_interrupt(void)
{
  static unsigned char data;// Holds the received scan code

  if (!edge) // Routine entered at falling edge
  {
    if(bitcount < 11 && bitcount > 2)// Bit 3 to 10 is data. Parity bit,
    {          // start and stop bits are ignored.
      data = (data >> 1);
      if(PIND & 8)
        data = data | 0x80;// Store a '1'
    }

    MCUCR = 3;// Set interrupt on rising edge
    edge = 1;

  } else {   // Routine entered at rising edge

    MCUCR = 2;// Set interrupt on falling edge
    edge = 0;

    if(--bitcount == 0)// All bits received
    {
      decode(data);
      bitcount = 11;
    }
  }
}


void decode(unsigned char sc)
{
  static unsigned char is_up=0, shift = 0, mode = 0;
  unsigned char i;
  if (!is_up)// Last data received was the up-key identifier
  {
    switch (sc)
    {
      case 0xF0 :// The up-key identifier
      is_up = 1;
      break;

      case 0x12 :// Left SHIFT
      shift = 1;
      break;
```

```
      case 0x59 :// Right SHIFT
      shift = 1;
      break;

      case 0x05 :// F1
      if(mode == 0)
        mode = 1;// Enter scan code mode
      if(mode == 2)
        mode = 3;// Leave scan code mode
      break;

      default:
      if(mode == 0 || mode == 3)// If ASCII mode
      {
        if(!shift)// If shift not pressed,
        {     // do a table look-up
          for(i = 0; unshifted[i][0]!=sc && unshifted[i][0]; i++);
          if (unshifted[i][0] == sc) {
            put_kbbuff(unshifted[i][1]);
          }
        } else {// If shift pressed
          for(i = 0; shifted[i][0]!=sc && shifted[i][0]; i++);
          if (shifted[i][0] == sc) {
            put_kbbuff(shifted[i][1]);
          }
        }
      } else{ // Scan code mode
        print_hexbyte(sc);// Print scan code
        put_kbbuff(' ');
        put_kbbuff(' ');
      }
      break;
    }
  } else {
    is_up = 0;// Two 0xF0 in a row not allowed
    switch (sc)
    {
      case 0x12 :// Left SHIFT
      shift = 0;
      break;

      case 0x59 :// Right SHIFT
      shift = 0;
      break;

      case 0x05 :// F1
      if(mode == 1)
        mode = 2;
      if(mode == 3)
        mode = 0;
```

```
      break;
      case 0x06 :// F2
      clr();
      break;
    }
  }
}

void put_kbbuff(unsigned char c)
{
  if (buffcnt<BUFF_SIZE)// If buffer not full
  {
    *inpt = c;// Put character into buffer
    inpt++;    // Increment pointer

    buffcnt++;

    if (inpt >= kb_buffer + BUFF_SIZE)// Pointer wrapping
      inpt = kb_buffer;
    }
}

int getchar(void)
{
  int byte;
  while(buffcnt == 0);// Wait for data

  byte = *outpt;// Get byte
  outpt++;   // Increment pointer

  if (outpt >= kb_buffer + BUFF_SIZE)// Pointer wrapping
    outpt = kb_buffer;

  buffcnt--; // Decrement buffer count

  return byte;
}
```

Gpr.c

```
#include "gpr.h"

void print_hexbyte(unsigned char i)
{
    unsigned char h, l;

    h = i & 0xF0;               // High nibble
    h = h>>4;
    h = h + '0';

    if (h > '9')
        h = h + 7;

    l = (i & 0x0F)+'0';         // Low nibble
    if (l > '9')
        l = l + 7;


    putchar(h);
    putchar(l);
}


void delay(char d)
{
    char i,j,k;
    for(i=0; i<d; i++)
        for(j=0; j<40; j++)
            for(k=0; k<176; k++);
}
```

Pindefs.h

```
//*********************
// Pin definition file
//*********************


// Keyboard konnections
#define PIN_KB  PIND
#define PORT_KB PORTD
#define CLOCK   2
#define DATAPIN 3
```

Scancodes.h

```
// Unshifted characters
flash unsigned char unshifted[][2] = {
0x0d,9,
0x0e,'|',
0x15,'q',
0x16,'1',
0x1a,'z',
0x1b,'s',
0x1c,'a',
0x1d,'w',
0x1e,'2',
0x21,'c',
0x22,'x',
0x23,'d',
0x24,'e',
0x25,'4',
0x26,'3',
0x29,' ',
0x2a,'v',
0x2b,'f',
0x2c,'t',
0x2d,'r',
0x2e,'5',
0x31,'n',
0x32,'b',
0x33,'h',
0x34,'g',
0x35,'y',
0x36,'6',
0x39,',',
0x3a,'m',
0x3b,'j',
0x3c,'u',
0x3d,'7',
0x3e,'8',
0x41,',',
0x42,'k',
0x43,'i',
0x44,'o',
0x45,'0',
0x46,'9',
0x49,'.',
0x4a,'-',
0x4b,'l',
0x4c,'ø',
0x4d,'p',
0x4e,'+',
0x52,'æ',
0x54,'å',
0x55,'\\',
```

```
0x5a,13,
0x5b,'¨',
0x5d,'\'',
0x61,'<',
0x66,8,
0x69,'1',
0x6b,'4',
0x6c,'7',
0x70,'0',
0x71,',',
0x72,'2',
0x73,'5',
0x74,'6',
0x75,'8',
0x79,'+',
0x7a,'3',
0x7b,'-',
0x7c,'*',
0x7d,'9',
0,0
};
// Shifted characters
flash unsigned char shifted[][2] = {
0x0d,9,
0x0e,'§',
0x15,'Q',
0x16,'!',
0x1a,'Z',
0x1b,'S',
0x1c,'A',
0x1d,'W',
0x1e,'"',
0x21,'C',
0x22,'X',
0x23,'D',
0x24,'E',
0x25,'¤',
0x26,'#',
0x29,' ',
0x2a,'V',
0x2b,'F',
0x2c,'T',
0x2d,'R',
0x2e,'%',
0x31,'N',
0x32,'B',
0x33,'H',
0x34,'G',
0x35,'Y',
0x36,'&',
```

```
0x39,'L',
0x3a,'M',
0x3b,'J',
0x3c,'U',
0x3d,'/',
0x3e,'(',
0x41,';',
0x42,'K',
0x43,'I',
0x44,'O',
0x45,'=',
0x46,')',
0x49,':',
0x4a,'_',
0x4b,'L',
0x4c,'Ø',
0x4d,'P',
0x4e,'?',
0x52,'Æ',
0x54,'Å',
0x55,'`',
0x5a,13,
0x5b,'^',
0x5d,'*',
0x61,'>',
0x66,8,
0x69,'1',
0x6b,'4',
0x6c,'7',
0x70,'0',
0x71,',',
0x72,'2',
0x73,'5',
0x74,'6',
0x75,'8',
0x79,'+',
0x7a,'3',
0x7b,'-',
0x7c,'*',
0x7d,'9',
0,0
};
```

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 487-2600

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-76-58-30-00
FAX (33) 4-76-58-34-80

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

1235B–AVR–05/02 0M

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 142 for details.

Pin Descriptions

V_{CC} — 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 57.

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 61.

$\overline{\text{RESET}}$ — 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.

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

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

Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x3F (0x5F)	SREG	I	T	H	S	V	N	Z	C	8
0x3E (0x5E)	SPH	–	–	–	–	–	SP10	SP9	SP8	10
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	10
0x3C (0x5C)	OCR0	Timer/Counter0 Output Compare Register								82
0x3B (0x5B)	GICR	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	47, 66
0x3A (0x5A)	GIFR	INTF1	INTF0	INTF2	–	–	–	–	–	67
0x39 (0x59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	82, 112, 130
0x38 (0x58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	83, 113, 131
0x37 (0x57)	SPMCR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	224
0x36 (0x56)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	177
0x35 (0x55)	MCUCR	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	30, 65
0x34 (0x54)	MCUCSR	–	ISC2	–	–	WDRF	BORF	EXTRF	PORF	38, 66
0x33 (0x53)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	80
0x32 (0x52)	TCNT0	Timer/Counter0 (8 Bits)								82
0x31 (0x51)	OSCCAL	Oscillator Calibration Register								28
0x30 (0x50)	SFIOR	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	56,85,132,199,219
0x2F (0x4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	107
0x2E (0x4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	110
0x2D (0x4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								111
0x2C (0x4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								111
0x2B (0x4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte								111
0x2A (0x4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte								111
0x29 (0x49)	OCR1BH	Timer/Counter1 – Output Compare Register B High Byte								111
0x28 (0x48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low Byte								111
0x27 (0x47)	ICR1H	Timer/Counter1 – Input Capture Register High Byte								111
0x26 (0x46)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte								111
0x25 (0x45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	125
0x24 (0x44)	TCNT2	Timer/Counter2 (8 Bits)								127
0x23 (0x43)	OCR2	Timer/Counter2 Output Compare Register								128
0x22 (0x42)	ASSR	–	–	–	–	AS2	TCN2UB	OCR2UB	TCR2UB	128
0x21 (0x41)	WDTCR	–	–	–	WDCE	WDE	WDP2	WDP1	WDP0	40
0x20[1] (0x40)[1]	UBRRH	URSEL	–	–	–	UBRR[11:8]				165
	UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	163
0x1F (0x3F)	EEARH	–	–	–	–	–	–	–	EEAR8	17
0x1E (0x3E)	EEARL	EEPROM Address Register Low Byte								17
0x1D (0x3D)	EEDR	EEPROM Data Register								17
0x1C (0x3C)	EECR	–	–	–	–	EERIE	EEMWE	EEWE	EERE	17
0x1B (0x3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	63
0x1A (0x3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	63
0x19 (0x39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	63
0x18 (0x38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	63
0x17 (0x37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	63
0x16 (0x36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	64
0x15 (0x35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	64
0x14 (0x34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	64
0x13 (0x33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	64
0x12 (0x32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	64
0x11 (0x31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	64
0x10 (0x30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	64
0x0F (0x2F)	SPDR	SPI Data Register								139
0x0E (0x2E)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	139
0x0D (0x2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	137
0x0C (0x2C)	UDR	USART I/O Data Register								160
0x0B (0x2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	161
0x0A (0x2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	162
0x09 (0x29)	UBRRL	USART Baud Rate Register Low Byte								165
0x08 (0x28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	199
0x07 (0x27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	215
0x06 (0x26)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	217
0x05 (0x25)	ADCH	ADC Data Register High Byte								218
0x04 (0x24)	ADCL	ADC Data Register Low Byte								218
0x03 (0x23)	TWDR	Two-wire Serial Interface Data Register								179
0x02 (0x22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	180
0x01 (0x21)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	179

Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x00 (0x20)	TWBR	Two-wire Serial Interface Bit Rate Register								177

Notes:
1. Refer to the USART description for details on how to access UBRRH and UCSRC.
2. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
3. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rdl,K	Add Immediate to Word	Rdh:Rdl ← Rdh:Rdl + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract Immediate from Word	Rdh:Rdl ← Rdh:Rdl - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← 0xFF – Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← 0x00 – Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd • (0xFF - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd – 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← 0xFF	None	1
MUL	Rd, Rr	Multiply Unsigned	R1:R0 ← Rd x Rr	Z,C	2
MULS	Rd, Rr	Multiply Signed	R1:R0 ← Rd x Rr	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	R1:R0 ← Rd x Rr	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	R1:R0 ← (Rd x Rr) << 1	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	R1:R0 ← (Rd x Rr) << 1	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	R1:R0 ← (Rd x Rr) << 1	Z,C	2
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC ← Z	None	2
JMP	k	Direct Jump	PC ← k	None	3
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	PC ← Z	None	3
CALL	k	Direct Subroutine Call	PC ← k	None	4
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1 / 2 / 3
CP	Rd,Rr	Compare	Rd – Rr	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	Rd – Rr – C	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	Rd – K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC←PC+k + 1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC←PC+k + 1	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1 / 2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1

Mnemonics	Operands	Description	Operation	Flags	#Clocks
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/Timer)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A

Ordering Information[1]

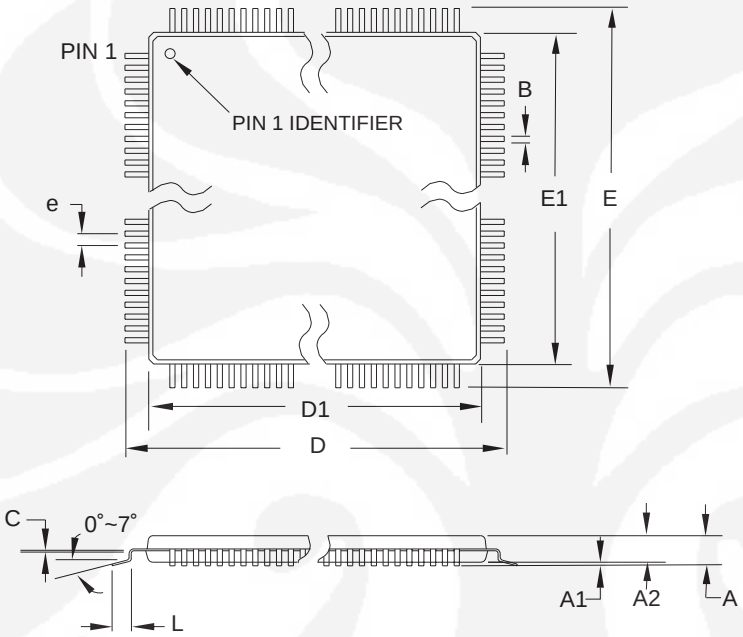
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
8	2.7 - 5.5V	ATmega8535L-8AC	44A	Commercial (0°C to 70°C)
		ATmega8535L-8PC	40P6	
		ATmega8535L-8JC	44J	
		ATmega8535L-8MC	44M1	
		ATmega8535L-8AI	44A	Industrial (-40°C to 85°C)
		ATmega8535L-8PI	40P6	
		ATmega8535L-8JI	44J	
		ATmega8535L-8MI	44M1	
16	4.5 - 5.5V	ATmega8535-16AC	44A	Commercial (0°C to 70°C)
		ATmega8535-16PC	40P6	
		ATmega8535-16JC	44J	
		ATmega8535-16MC	44M1	
		ATmega8535-16AI	44A	Industrial (-40°C to 85°C)
		ATmega8535-16PI	40P6	
		ATmega8535-16JI	44J	
		ATmega8535-16MI	44M1	

Note: 1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.

Package Type	
44A	44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
40P6	40-pin, 0.600” Wide, Plastic Dual Inline Package (PDIP)
44J	44-lead, Plastic J-ledded Chip Carrier (PLCC)
44M1-A	44-pad, 7 x 7 x 1.0 mm body, lead pitch 0.50 mm, Micro Lead Frame Package (MLF)

Packaging Information

44A



COMMON DIMENSIONS
(Unit of Measure = mm)

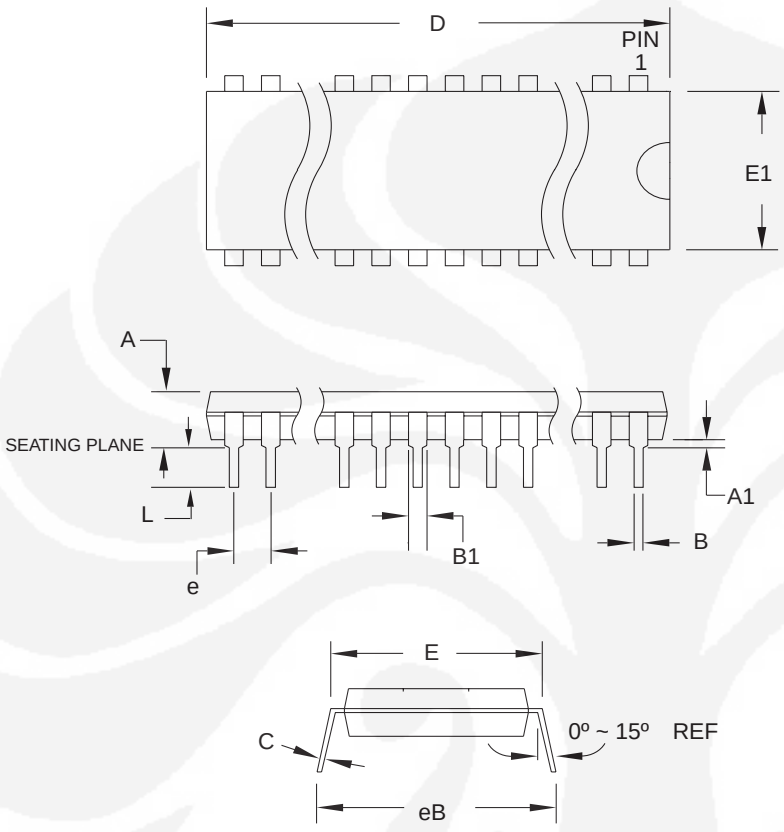
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	11.75	12.00	12.25	
D1	9.90	10.00	10.10	Note 2
E	11.75	12.00	12.25	
E1	9.90	10.00	10.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

Notes:
1. This package conforms to JEDEC reference MS-026, Variation ACB.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10 mm maximum.

10/5/2001

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**44A,** 44-lead, 10 x 10 mm Body Size, 1.0 mm Body Thickness, 0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)	44A	B

40P6



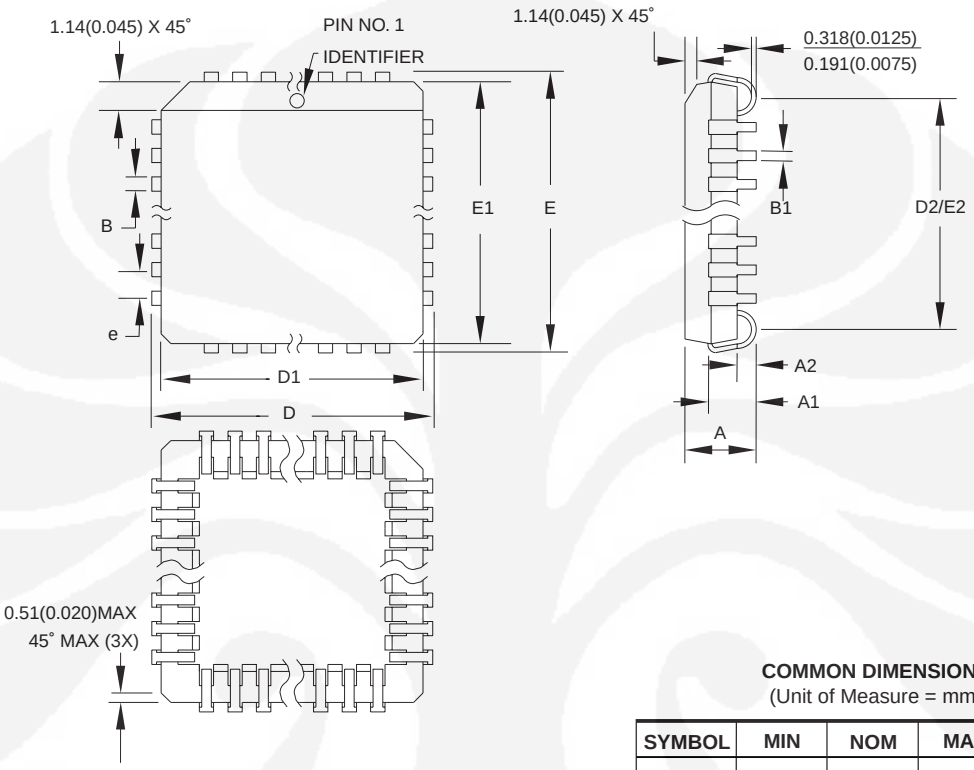
COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.826	
A1	0.381	–	–	
D	52.070	–	52.578	Note 2
E	15.240	–	15.875	
E1	13.462	–	13.970	Note 2
B	0.356	–	0.559	
B1	1.041	–	1.651	
L	3.048	–	3.556	
C	0.203	–	0.381	
eB	15.494	–	17.526	
e	2.540 TYP			

Notes:
1. This package conforms to JEDEC reference MS-011, Variation AC.
2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

09/28/01

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**40P6**, 40-lead (0.600"/15.24 mm Wide) Plastic Dual Inline Package (PDIP)	40P6	B



COMMON DIMENSIONS
(Unit of Measure = mm)

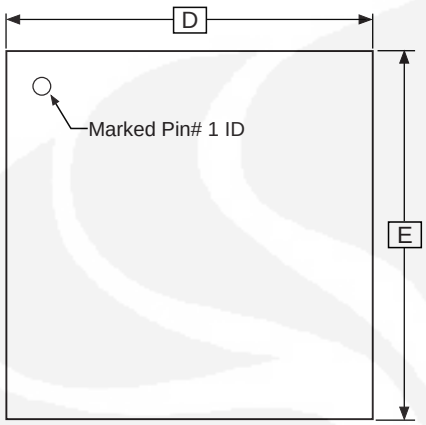
SYMBOL	MIN	NOM	MAX	NOTE
A	4.191	–	4.572	
A1	2.286	–	3.048	
A2	0.508	–	–	
D	17.399	–	17.653	
D1	16.510	–	16.662	Note 2
E	17.399	–	17.653	
E1	16.510	–	16.662	Note 2
D2/E2	14.986	–	16.002	
B	0.660	–	0.813	
B1	0.330	–	0.533	
e	1.270 TYP			

Notes:
1. This package conforms to JEDEC reference MS-018, Variation AC.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is .010"(0.254 mm) per side. Dimension D1 and E1 include mold mismatch and are measured at the extreme material condition at the upper or lower parting line.
3. Lead coplanarity is 0.004" (0.102 mm) maximum.

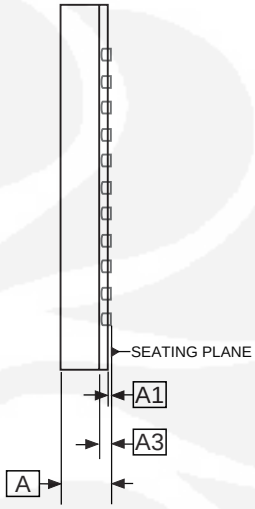
10/04/01

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**44J**, 44-lead, Plastic J-leaded Chip Carrier (PLCC)	44J	B

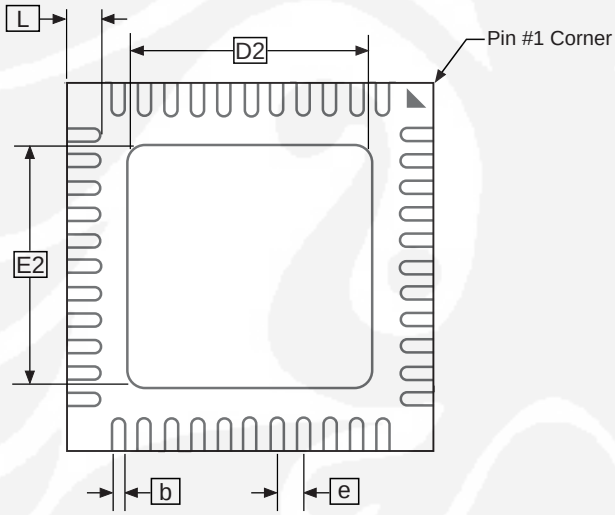
44M1-A



TOP VIEW



SIDE VIEW



BOTTOM VIEW

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.90	1.00	
A1	–	0.02	0.05	
A3		0.25 REF		
b	0.18	0.23	0.30	
D		7.00 BSC		
D2	5.00	5.20	5.40	
E		7.00 BSC		
E2	5.00	5.20	5.40	
e		0.50 BSC		
L	0.35	0.55	0.75	

Note: JEDEC Standard MO-220, Fig. 1 (SAW Singulation) VKKD-1.

01/15/03

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**44M1**, 44-pad, 7 x 7 x 1.0 mm Body, Lead Pitch 0.50 mm Micro Lead Frame Package (MLF)	44M1	C

Data Sheet Change Log for ATmega8535

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

Changes from Rev. 2502A-06/02 to Rev. 2502B-09/02

1. **Canged the Endurance on the Flash to 10,000 Write/Erase Cycles.**

Changes from Rev. 2502B-09/02 to Rev. 2502C-05/03

1. **Updated “Packaging Information” on page 12.**
2. **Updated Figure 1 on page 2, Figure 84 on page 175, Figure 85 on page 181, Figure 87 on page 187, Figure 98 on page 203.**
3. **Added the section “EEPROM Write During Power-down Sleep Mode” on page 20.**
4. **Removed the references to the application notes “Multi-purpose Oscillator” and “32 kHz Crystal Oscillator”, which do not exist.**
5. **Updated code examples on page 42.**
6. **Removed ADHSM bit.**
7. **Renamed Port D pin ICP to ICP1. See “Alternate Functions of Port D” on page 61.**
8. **Added information about PWM symmetry for Timer 0 on page 76 and Timer 2 on page 123.**
9. **Updated Table 68 on page 165, Table 75 on page 186, Table 76 on page 189, Table 77 on page 192, Table 108 on page 249, Table 113 on page 256.**
10. **Updated description on “Bit 5 – TWSTA: TWI START Condition Bit” on page 178.**
11. **Updated the description in “Filling the Temporary Buffer (Page Loading)” and “Performing a Page Write” on page 227.**
12. **Removed the section description in “SPI Serial Programming Characteristics” on page 250.**
13. **Updated “Electrical Characteristics” on page 251.**
14. **Updated “ADC Characteristics – Preliminary Data” on page 258.**
14. **Updated “Register Summary” on page 6.**
15. **Various Timer 1 corrections.**
16. **Added WD_FUSE period in Table 108 on page 249.**

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

2502CS–AVR–04/03 0M

Part Number: TC15-11YWA Yellow

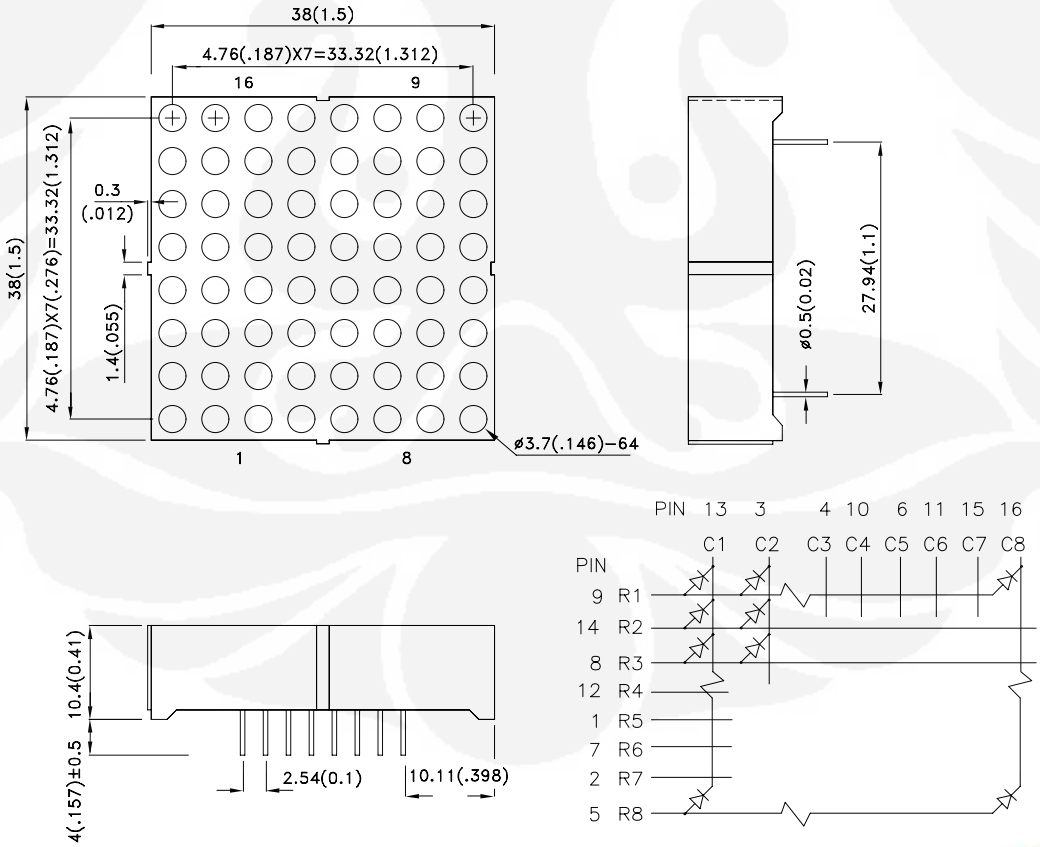
Features

- 1.5 INCH MATRIX HEIGHT.
- DOT SIZE 3.7mm.
- LOW CURRENT OPERATION.
- HIGH CONTRAST AND LIGHT OUTPUT.
- COMPATIBLE WITH ASCII AND EBCDIC CODES.
- STACKABLE HORIZONTALLY AND VERTICALLY.
- COLUMN CATHODE AND COLUMN ANODE AVAILABLE.
- EASY MOUNTING ON P.C. BOARDS OR SOCKETS.
- MULTICOLOR AVAILABLE.
- MECHANICALLY RUGGED.
- STANDARD : GRAY FACE, WHITE DOT.
- RoHS COMPLIANT.

Description

The Yellow source color devices are made with Gallium Arsenide Phosphide on Gallium Phosphide Yellow Light Emitting Diode.

Package Dimensions& Internal Circuit Diagram



Notes:
1. All dimensions are in millimeters (inches), Tolerance is ±0.25(0.01")unless otherwise noted.
2. Specifications are subject to change without notice.

Selection Guide

Part No.	Dice	Lens Type	Iv (ucd) [1] @ 10mA		Description
			Min.	Typ.	
TC15-11YWA	Yellow (GaAsP/GaP)	WHITE DIFFUSED	1900	8000	Column Cathode

Note:
1. Luminous intensity/ luminous Flux: +/-15%.

Electrical / Optical Characteristics at TA=25°C

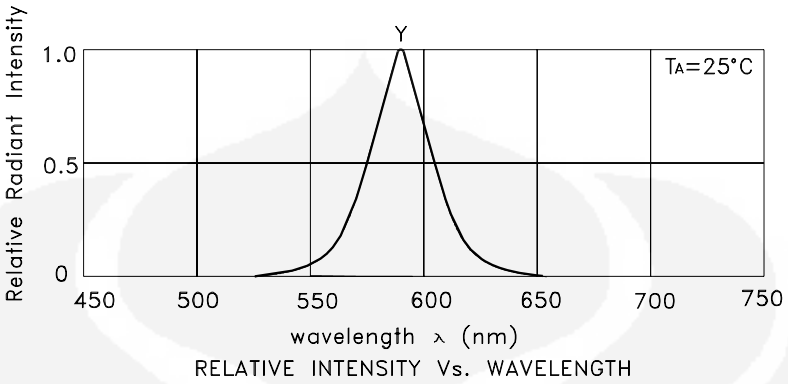
Symbol	Parameter	Device	Typ.	Max.	Units	Test Conditions
λpeak	Peak Wavelength	Yellow	590		nm	I_F=20mA
λD [1]	Dominant Wavelength	Yellow	588		nm	I_F=20mA
Δλ1/2	Spectral Line Half-width	Yellow	35		nm	I_F=20mA
C	Capacitance	Yellow	20		pF	V_F=0V;f=1MHz
V_F [2]	Forward Voltage	Yellow	2.1	2.5	V	I_F=20mA
I_R	Reverse Current	Yellow		10	uA	V_R=5V

Notes:
1.Wavelength: +/-1nm.
2. Forward Voltage: +/-0.1V.

Absolute Maximum Ratings at TA=25°C

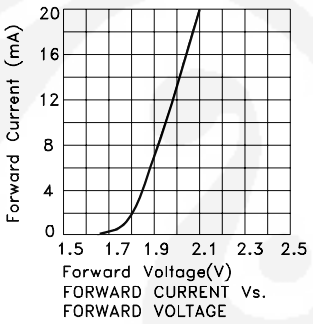
Parameter	Yellow	Units
Power dissipation	75	mW
DC Forward Current	30	mA
Peak Forward Current [1]	140	mA
Reverse Voltage	5	V
Operating / Storage Temperature	-40°C To +85°C	
Lead Solder Temperature[2]	260°C For 3-5 Seconds	

Notes:
1. 1/10 Duty Cycle, 0.1ms Pulse Width.
2. 2mm below package base.

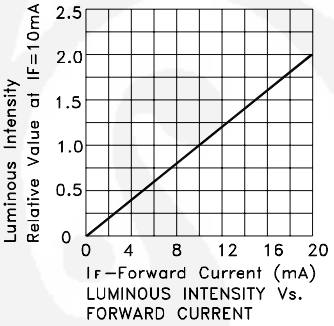


RELATIVE INTENSITY Vs. WAVELENGTH

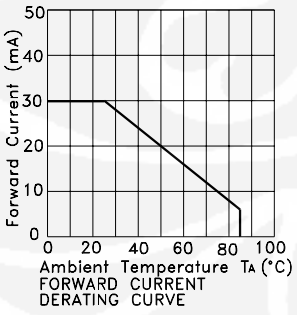
Yellow **TC15-11YWA**



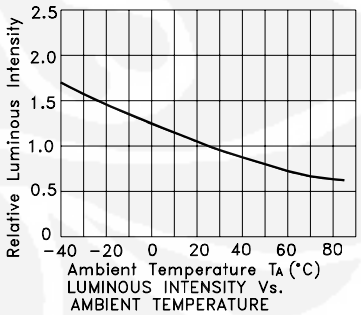
FORWARD CURRENT Vs. FORWARD VOLTAGE



LUMINOUS INTENSITY Vs. FORWARD CURRENT

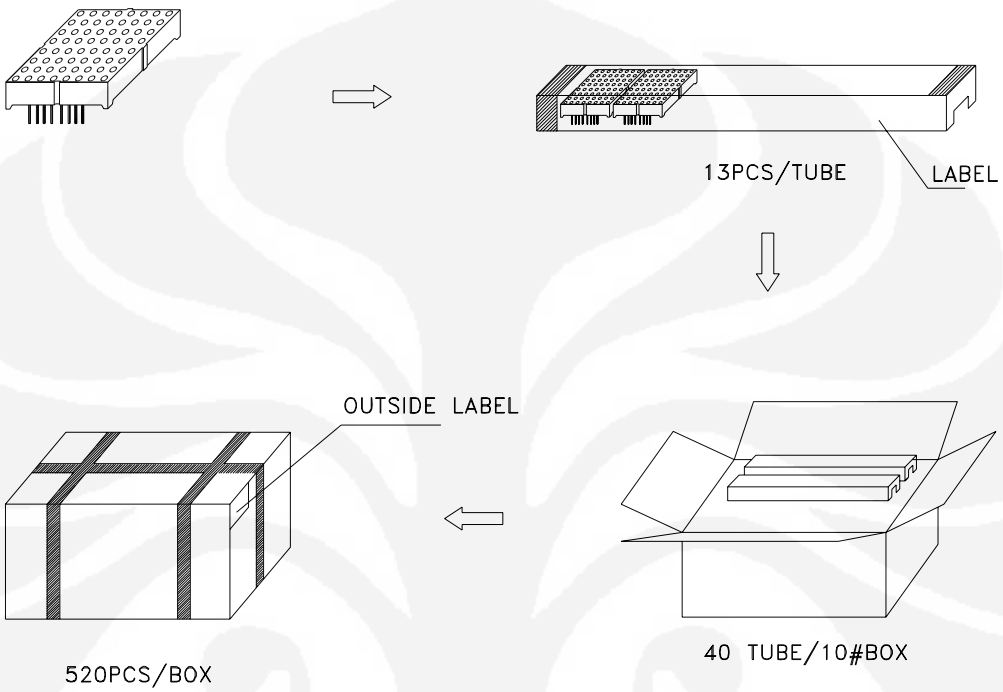


FORWARD CURRENT DERATING CURVE

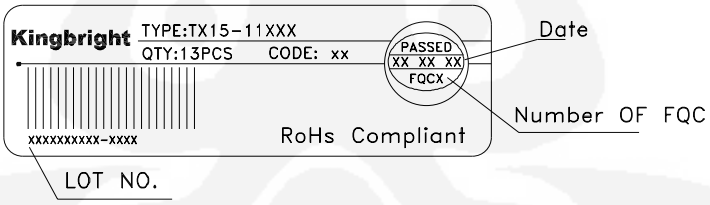


LUMINOUS INTENSITY Vs. AMBIENT TEMPERATURE

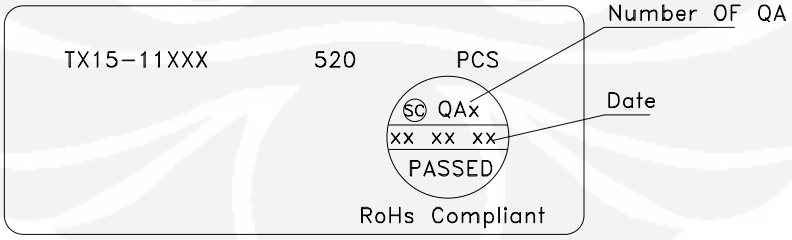
PACKING & LABEL SPECIFICATIONS **TC15-11YWA**



Inside LABEL Paste On The IC-tube



Outside LABEL Paste On The Box



HD44780U (LCD-II)

(Dot Matrix Liquid Crystal Display Controller/Driver)

HITACHI

ADE-207-272(Z)
'99.9
Rev. 0.0

Description

The HD44780U dot-matrix liquid crystal display controller and driver LSI displays alphanumerics, Japanese kana characters, and symbols. It can be configured to drive a dot-matrix liquid crystal display under the control of a 4- or 8-bit microprocessor. Since all the functions such as display RAM, character generator, and liquid crystal driver, required for driving a dot-matrix liquid crystal display are internally provided on one chip, a minimal system can be interfaced with this controller/driver.

A single HD44780U can display up to one 8-character line or two 8-character lines.

The HD44780U has pin function compatibility with the HD44780S which allows the user to easily replace an LCD-II with an HD44780U. The HD44780U character generator ROM is extended to generate 208 5 × 8 dot character fonts and 32 5 × 10 dot character fonts for a total of 240 different character fonts.

The low power supply (2.7V to 5.5V) of the HD44780U is suitable for any portable battery-driven product requiring low power dissipation.

Features

- 5 × 8 and 5 × 10 dot matrix possible
- Low power operation support:
  - 2.7 to 5.5V
- Wide range of liquid crystal display driver power
  - 3.0 to 11V
- Liquid crystal drive waveform
  - A (One line frequency AC waveform)
- Correspond to high speed MPU bus interface
  - 2 MHz (when V_{CC} = 5V)
- 4-bit or 8-bit MPU interface enabled
- 80 × 8-bit display RAM (80 characters max.)
- 9,920-bit character generator ROM for a total of 240 character fonts
  - 208 character fonts (5 × 8 dot)
  - 32 character fonts (5 × 10 dot)

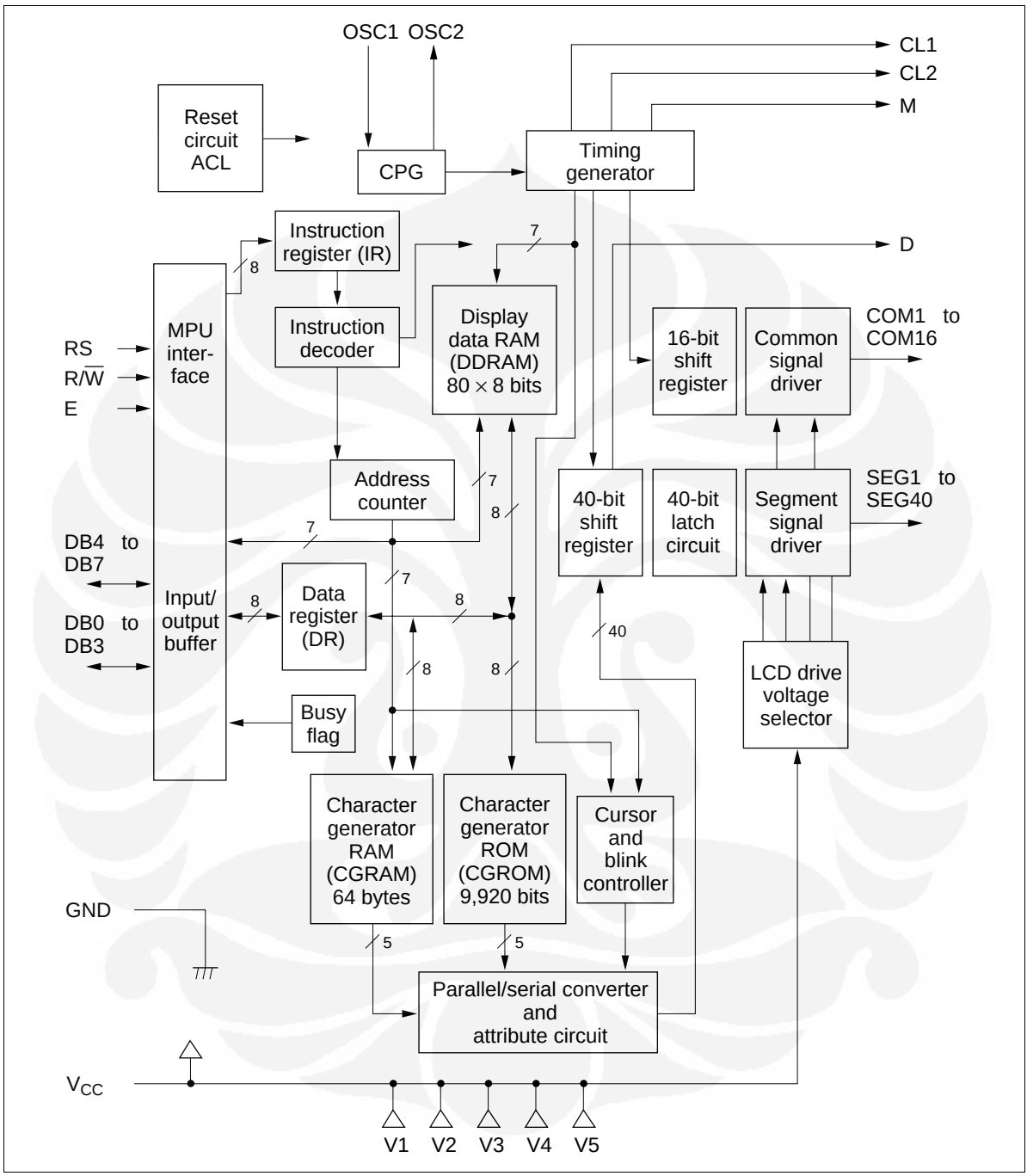
- 64 × 8-bit character generator RAM
  - 8 character fonts (5 × 8 dot)
  - 4 character fonts (5 × 10 dot)
- 16-common × 40-segment liquid crystal display driver
- Programmable duty cycles
  - 1/8 for one line of 5 × 8 dots with cursor
  - 1/11 for one line of 5 × 10 dots with cursor
  - 1/16 for two lines of 5 × 8 dots with cursor
- Wide range of instruction functions:
  - Display clear, cursor home, display on/off, cursor on/off, display character blink, cursor shift, display shift
- Pin function compatibility with HD44780S
- Automatic reset circuit that initializes the controller/driver after power on
- Internal oscillator with external resistors
- Low power consumption

Ordering Information

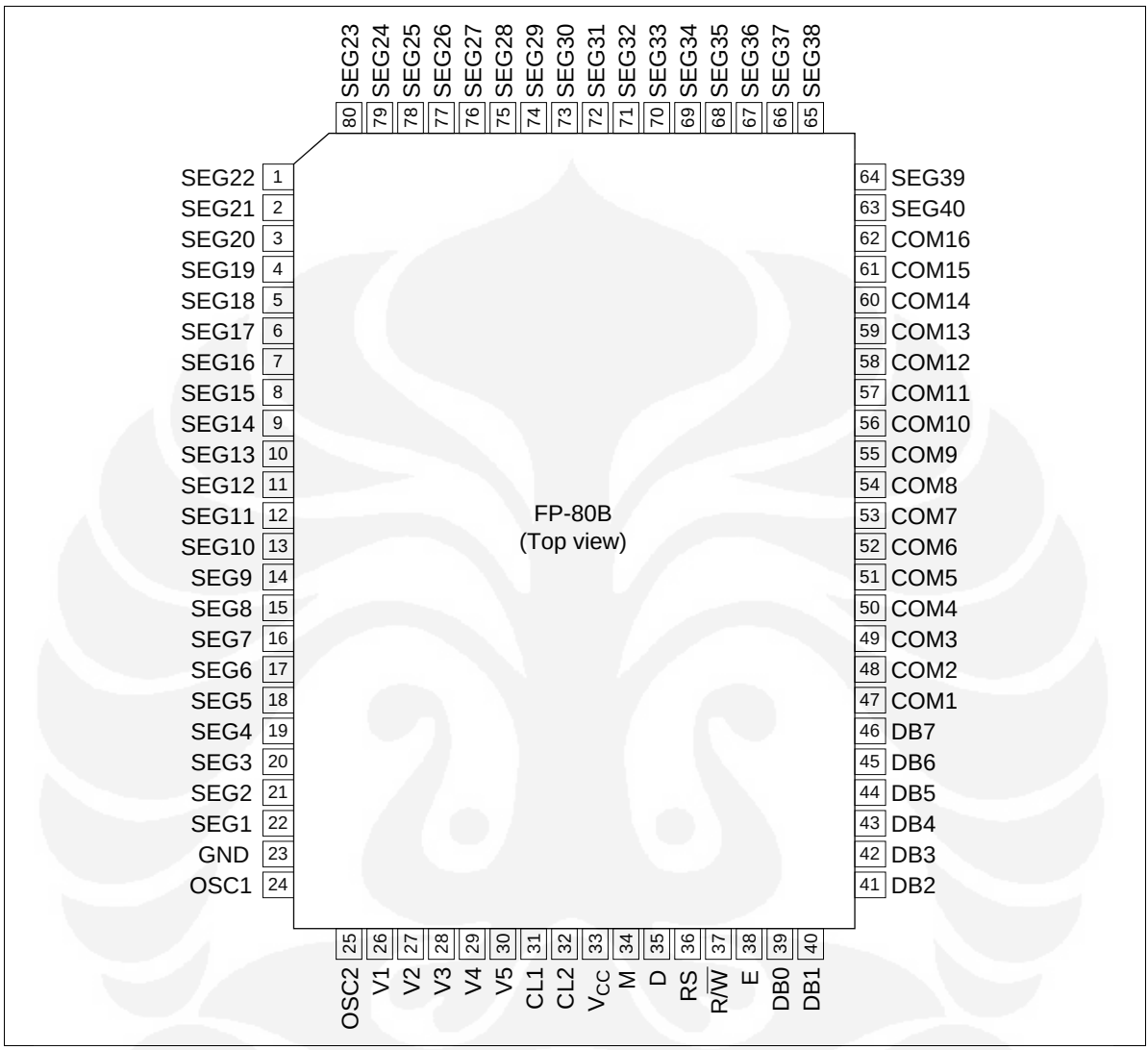
Type No.	Package	CGROM
HD44780UA00FS	FP-80B	Japanese standard font
HCD44780UA00	Chip	
HD44780UA00TF	TFP-80F	
HD44780UA02FS	FP-80B	European standard font
HCD44780UA02	Chip	
HD44780UA02TF	TFP-80F	
HD44780UBxxFS	FP-80B	Custom font
HCD44780UBxx	Chip	
HD44780UBxxTF	TFP-80F	

Note: xx: ROM code No.

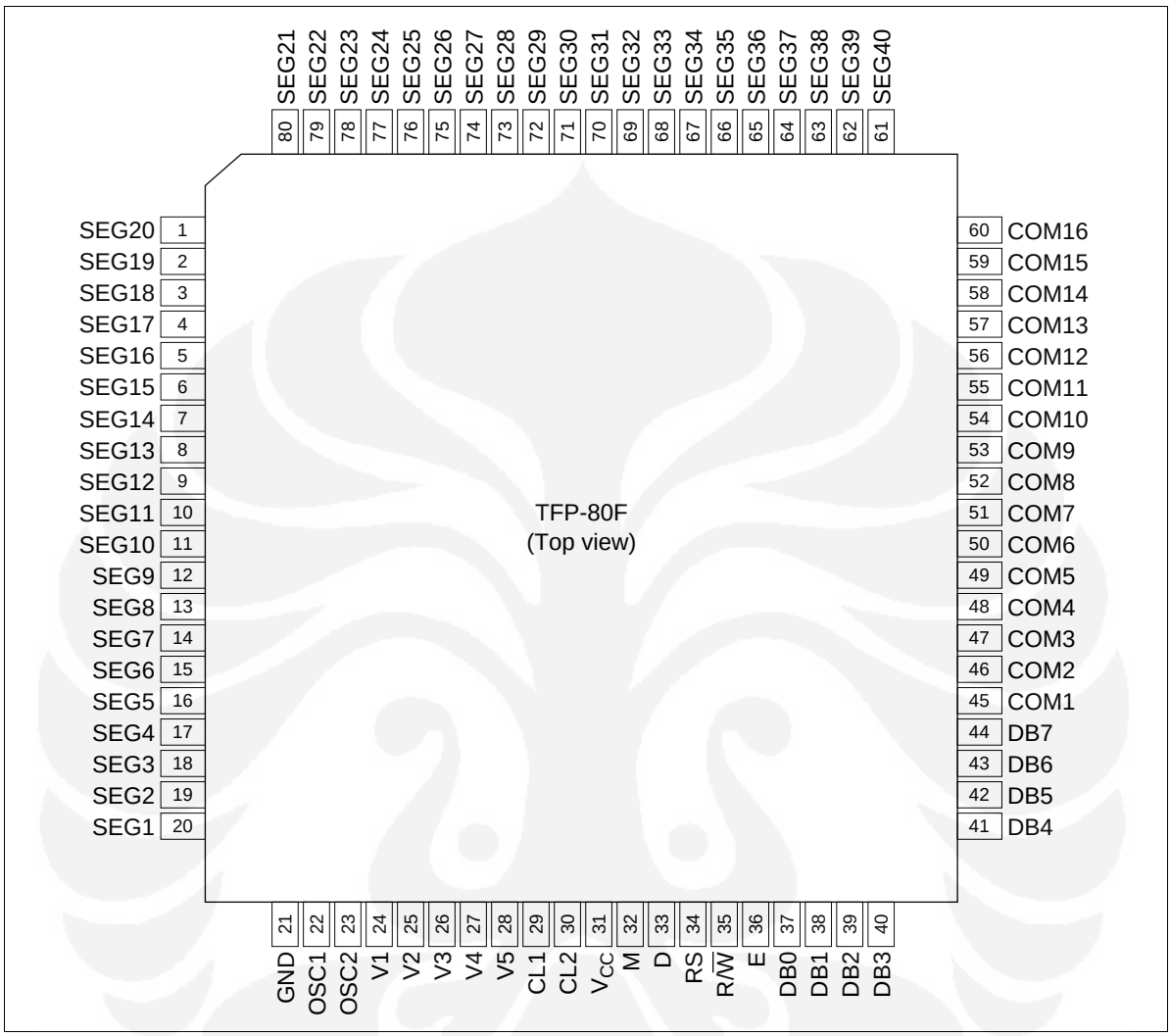
HD44780U Block Diagram



HD44780U Pin Arrangement (FP-80B)

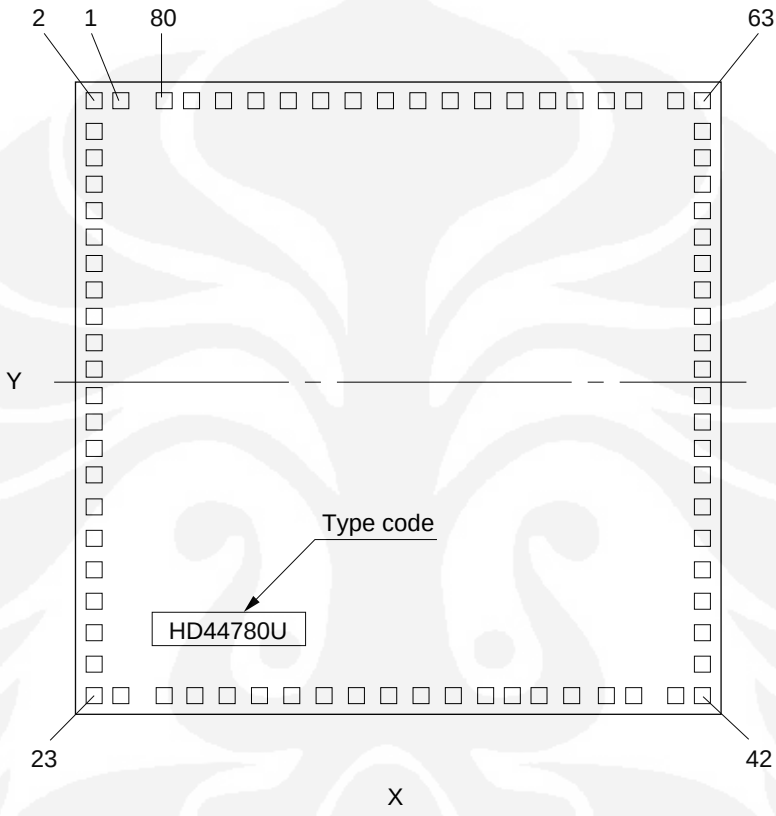


HD44780U Pin Arrangement (TFP-80F)



HD44780U Pad Arrangement

Chip size: 4.90×4.90 mm^2
Coordinate: Pad center (μm)
Origin: Chip center
Pad size: 114×114 μm^2



HCD44780U Pad Location Coordinates

		Coordinate				Coordinate	
Pad No.	Function	X (um)	Y (um)	Pad No.	Function	X (um)	Y (um)
1	SEG22	–2100	2313	41	DB2	2070	–2290
2	SEG21	–2280	2313	42	DB3	2260	–2290
3	SEG20	–2313	2089	43	DB4	2290	–2099
4	SEG19	–2313	1833	44	DB5	2290	–1883
5	SEG18	–2313	1617	45	DB6	2290	–1667
6	SEG17	–2313	1401	46	DB7	2290	–1452
7	SEG16	–2313	1186	47	COM1	2313	–1186
8	SEG15	–2313	970	48	COM2	2313	–970
9	SEG14	–2313	755	49	COM3	2313	–755
10	SEG13	–2313	539	50	COM4	2313	–539
11	SEG12	–2313	323	51	COM5	2313	–323
12	SEG11	–2313	108	52	COM6	2313	–108
13	SEG10	–2313	–108	53	COM7	2313	108
14	SEG9	–2313	–323	54	COM8	2313	323
15	SEG8	–2313	–539	55	COM9	2313	539
16	SEG7	–2313	–755	56	COM10	2313	755
17	SEG6	–2313	–970	57	COM11	2313	970
18	SEG5	–2313	–1186	58	COM12	2313	1186
19	SEG4	–2313	–1401	59	COM13	2313	1401
20	SEG3	–2313	–1617	60	COM14	2313	1617
21	SEG2	–2313	–1833	61	COM15	2313	1833
22	SEG1	–2313	–2073	62	COM16	2313	2095
23	GND	–2280	–2290	63	SEG40	2296	2313
24	OSC1	–2080	–2290	64	SEG39	2100	2313
25	OSC2	–1749	–2290	65	SEG38	1617	2313
26	V1	–1550	–2290	66	SEG37	1401	2313
27	V2	–1268	–2290	67	SEG36	1186	2313
28	V3	–941	–2290	68	SEG35	970	2313
29	V4	–623	–2290	69	SEG34	755	2313
30	V5	–304	–2290	70	SEG33	539	2313
31	CL1	–48	–2290	71	SEG32	323	2313
32	CL2	142	–2290	72	SEG31	108	2313
33	V_{CC}	309	–2290	73	SEG30	–108	2313
34	M	475	–2290	74	SEG29	–323	2313
35	D	665	–2290	75	SEG28	–539	2313
36	RS	832	–2290	76	SEG27	–755	2313
37	R/$\overline{W}$	1022	–2290	77	SEG26	–970	2313
38	E	1204	–2290	78	SEG25	–1186	2313
39	DB0	1454	–2290	79	SEG24	–1401	2313
40	DB1	1684	–2290	80	SEG23	–1617	2313

Pin Functions

Signal	No. of Lines	I/O	Device Interfaced with	Function
RS	1	I	MPU	Selects registers. 0: Instruction register (for write) Busy flag: address counter (for read) 1: Data register (for write and read)
R/$\overline{W}$	1	I	MPU	Selects read or write. 0: Write 1: Read
E	1	I	MPU	Starts data read/write.
DB4 to DB7	4	I/O	MPU	Four high order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. DB7 can be used as a busy flag.
DB0 to DB3	4	I/O	MPU	Four low order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. These pins are not used during 4-bit operation.
CL1	1	O	Extension driver	Clock to latch serial data D sent to the extension driver
CL2	1	O	Extension driver	Clock to shift serial data D
M	1	O	Extension driver	Switch signal for converting the liquid crystal drive waveform to AC
D	1	O	Extension driver	Character pattern data corresponding to each segment signal
COM1 to COM16	16	O	LCD	Common signals that are not used are changed to non-selection waveforms. COM9 to COM16 are non-selection waveforms at 1/8 duty factor and COM12 to COM16 are non-selection waveforms at 1/11 duty factor.
SEG1 to SEG40	40	O	LCD	Segment signals
V1 to V5	5	—	Power supply	Power supply for LCD drive V_{cc} –V5 = 11 V (max)
V_{cc}, GND	2	—	Power supply	V_{cc}: 2.7V to 5.5V, GND: 0V
OSC1, OSC2	2	—	Oscillation resistor clock	When crystal oscillation is performed, a resistor must be connected externally. When the pin input is an external clock, it must be input to OSC1.

Function Description

Registers

The HD44780U has two 8-bit registers, an instruction register (IR) and a data register (DR).

The IR stores instruction codes, such as display clear and cursor shift, and address information for display data RAM (DDRAM) and character generator RAM (CGRAM). The IR can only be written from the MPU.

The DR temporarily stores data to be written into DDRAM or CGRAM and temporarily stores data to be read from DDRAM or CGRAM. Data written into the DR from the MPU is automatically written into DDRAM or CGRAM by an internal operation. The DR is also used for data storage when reading data from DDRAM or CGRAM. When address information is written into the IR, data is read and then stored into the DR from DDRAM or CGRAM by an internal operation. Data transfer between the MPU is then completed when the MPU reads the DR. After the read, data in DDRAM or CGRAM at the next address is sent to the DR for the next read from the MPU. By the register selector (RS) signal, these two registers can be selected (Table 1).

Busy Flag (BF)

When the busy flag is 1, the HD44780U is in the internal operation mode, and the next instruction will not be accepted. When RS = 0 and R/$\overline{W}$ = 1 (Table 1), the busy flag is output to DB7. The next instruction must be written after ensuring that the busy flag is 0.

Address Counter (AC)

The address counter (AC) assigns addresses to both DDRAM and CGRAM. When an address of an instruction is written into the IR, the address information is sent from the IR to the AC. Selection of either DDRAM or CGRAM is also determined concurrently by the instruction.

After writing into (reading from) DDRAM or CGRAM, the AC is automatically incremented by 1 (decremented by 1). The AC contents are then output to DB0 to DB6 when RS = 0 and R/$\overline{W}$ = 1 (Table 1).

Table 1 Register Selection

RS	R/$\overline{W}$	Operation
0	0	IR write as an internal operation (display clear, etc.)
0	1	Read busy flag (DB7) and address counter (DB0 to DB6)
1	0	DR write as an internal operation (DR to DDRAM or CGRAM)
1	1	DR read as an internal operation (DDRAM or CGRAM to DR)

Display Data RAM (DDRAM)

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 × 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. See Figure 1 for the relationships between DDRAM addresses and positions on the liquid crystal display.

The DDRAM address (A_{DD}) is set in the address counter (AC) as hexadecimal.

- 1-line display (N = 0) (Figure 2)
  - When there are fewer than 80 display characters, the display begins at the head position. For example, if using only the HD44780, 8 characters are displayed. See Figure 3.

    When the display shift operation is performed, the DDRAM address shifts. See Figure 3.

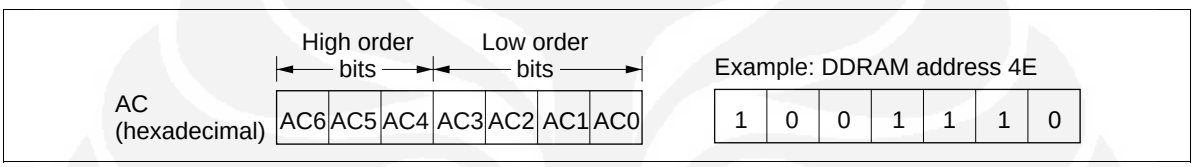


Figure 1 DDRAM Address

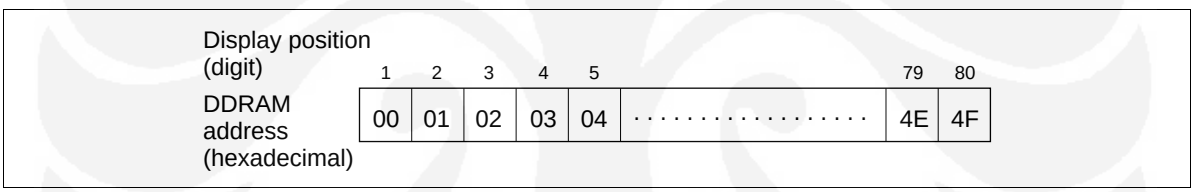


Figure 2 1-Line Display

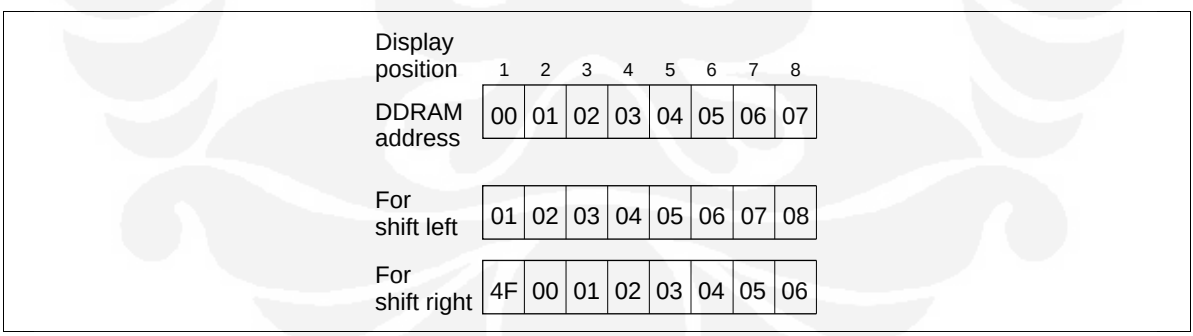


Figure 3 1-Line by 8-Character Display Example

- 2-line display (N = 1) (Figure 4)

  — Case 1: When the number of display characters is less than 40 × 2 lines, the two lines are displayed from the head. Note that the first line end address and the second line start address are not consecutive. For example, when just the HD44780 is used, 8 characters × 2 lines are displayed. See Figure 5.

  When display shift operation is performed, the DDRAM address shifts. See Figure 5.

Display position	1	2	3	4	5		39	40
DDRAM address (hexadecimal)	00	01	02	03	04	· · · · · · · · · · · · · · · · · ·	26	27
	40	41	42	43	44	· · · · · · · · · · · · · · · · · ·	66	67

Figure 4 2-Line Display

Display position	1	2	3	4	5	6	7	8
DDRAM address	00	01	02	03	04	05	06	07
	40	41	42	43	44	45	46	47
For shift left	01	02	03	04	05	06	07	08
	41	42	43	44	45	46	47	48
For shift right	27	00	01	02	03	04	05	06
	67	40	41	42	43	44	45	46

Figure 5 2-Line by 8-Character Display Example

— Case 2: For a 16-character × 2-line display, the HD44780 can be extended using one 40-output extension driver. See Figure 6.

When display shift operation is performed, the DDRAM address shifts. See Figure 6.

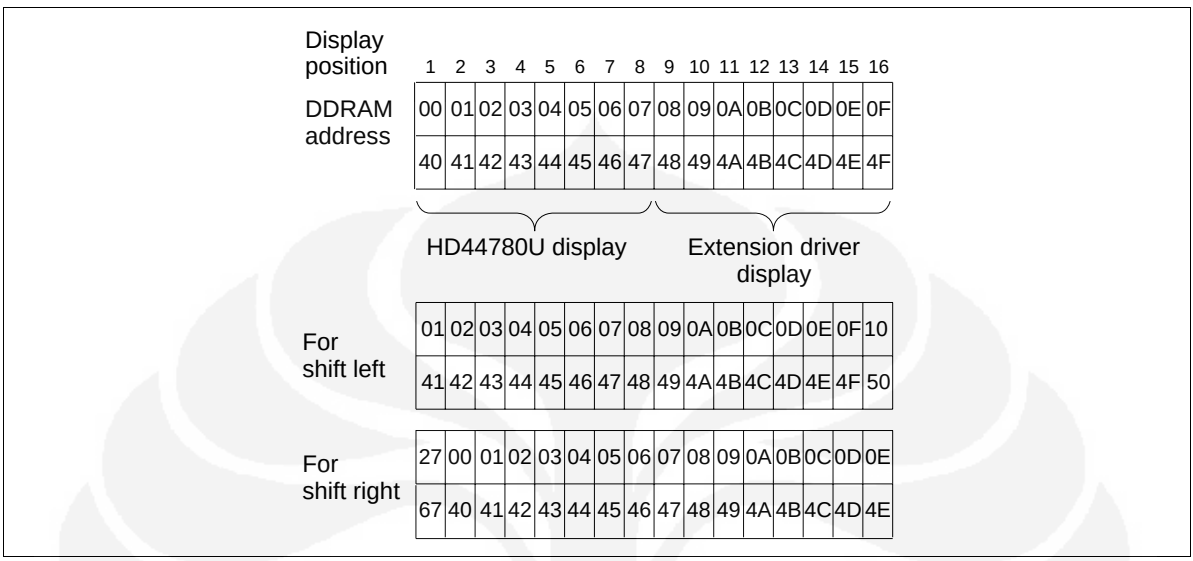


Figure 6 2-Line by 16-Character Display Example

Character Generator ROM (CGROM)

The character generator ROM generates 5 × 8 dot or 5 × 10 dot character patterns from 8-bit character codes (Table 4). It can generate 208 5 × 8 dot character patterns and 32 5 × 10 dot character patterns. User-defined character patterns are also available by mask-programmed ROM.

Character Generator RAM (CGRAM)

In the character generator RAM, the user can rewrite character patterns by program. For 5 × 8 dots, eight character patterns can be written, and for 5 × 10 dots, four character patterns can be written.

Write into DDRAM the character codes at the addresses shown as the left column of Table 4 to show the character patterns stored in CGRAM.

See Table 5 for the relationship between CGRAM addresses and data and display patterns.

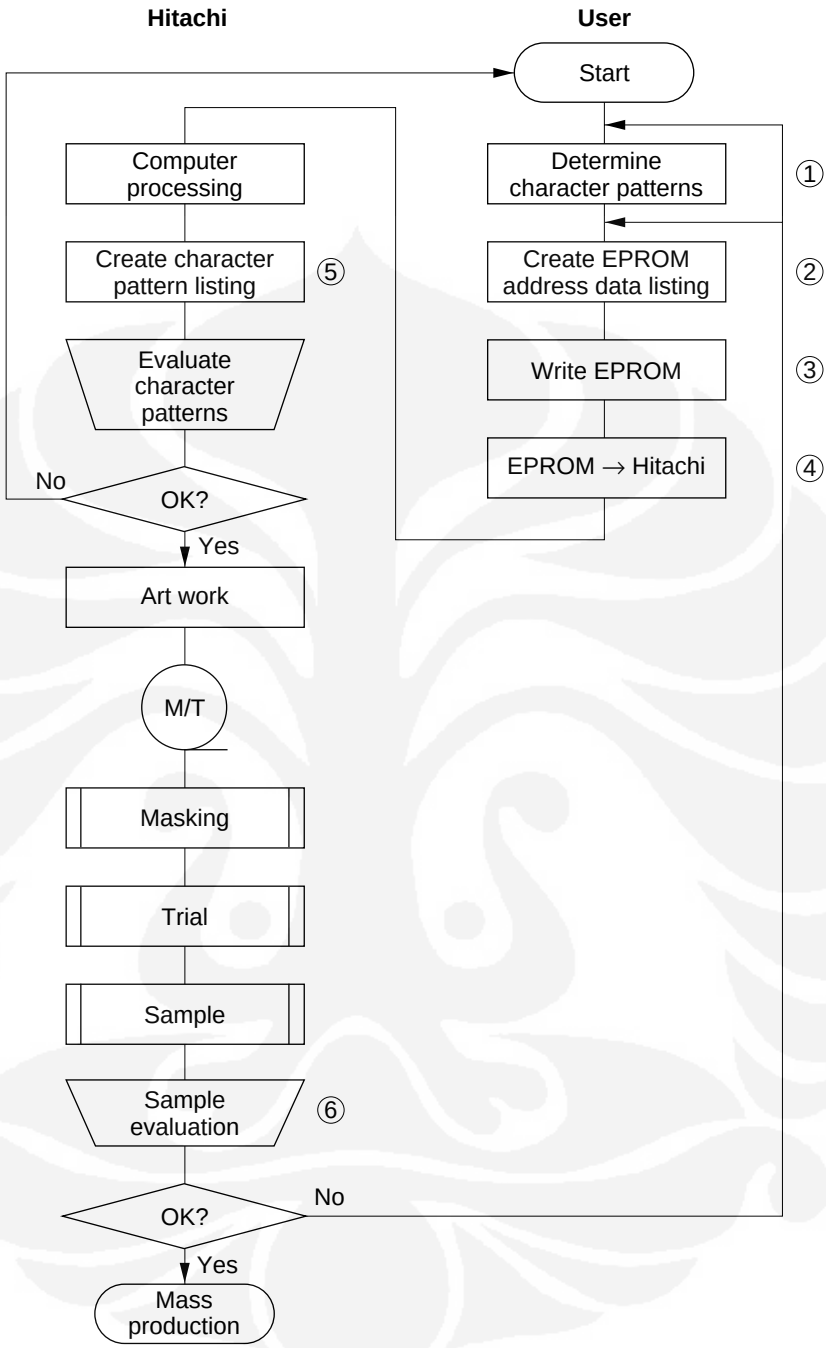
Areas that are not used for display can be used as general data RAM.

Modifying Character Patterns

- Character pattern development procedure

The following operations correspond to the numbers listed in Figure 7:

1. Determine the correspondence between character codes and character patterns.
2. Create a listing indicating the correspondence between EPROM addresses and data.
3. Program the character patterns into the EPROM.
4. Send the EPROM to Hitachi.
5. Computer processing on the EPROM is performed at Hitachi to create a character pattern listing, which is sent to the user.
6. If there are no problems within the character pattern listing, a trial LSI is created at Hitachi and samples are sent to the user for evaluation. When it is confirmed by the user that the character patterns are correctly written, mass production of the LSI proceeds at Hitachi.



Note: For a description of the numbers used in this figure, refer to the preceding page.

Figure 7 Character Pattern Development Procedure

- Programming character patterns

  This section explains the correspondence between addresses and data used to program character patterns in EPROM. The HD44780U character generator ROM can generate 208 5 × 8 dot character patterns and 32 5 × 10 dot character patterns for a total of 240 different character patterns.

  — Character patterns

  EPROM address data and character pattern data correspond with each other to form a 5 × 8 or 5 × 10 dot character pattern (Tables 2 and 3).

Table 2 Example of Correspondence between EPROM Address Data and Character Pattern (5 × 8 Dots)

EPROM Address												Data					
A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	O4	O3	O2	O1	LSB O0	
								0	0	0	0	1	0	0	0	0	
								0	0	0	1	1	0	0	0	0	
								0	0	1	0	1	0	1	1	0	
								0	0	1	1	1	1	0	0	1	
								0	1	0	0	1	0	0	0	1	
								0	1	0	1	1	0	0	0	1	
								0	1	1	0	1	1	1	1	0	
0	1	1	0	0	0	1	0	0	1	1	1	0	0	0	0	0	← Cursor position
								1	0	0	0	0	0	0	0	0	
								1	0	0	1	0	0	0	0	0	
								1	0	1	0	0	0	0	0	0	
								1	0	1	1	0	0	0	0	0	
								1	1	0	0	0	0	0	0	0	
								1	1	0	1	0	0	0	0	0	
								1	1	1	0	0	0	0	0	0	
								1	1	1	1	0	0	0	0	0	

Character code — Line position

Notes:
1. EPROM addresses A11 to A4 correspond to a character code.
2. EPROM addresses A3 to A0 specify a line position of the character pattern.
3. EPROM data O4 to O0 correspond to character pattern data.
4. EPROM data O5 to O7 must be specified as 0.
5. A lit display position (black) corresponds to a 1.
6. Line 9 and the following lines must be blanked with 0s for a 5 × 8 dot character fonts.

— Handling unused character patterns

1. EPROM data outside the character pattern area: Always input 0s.
2. EPROM data in CGRAM area: Always input 0s. (Input 0s to EPROM addresses 00H to FFH.)
3. EPROM data used when the user does not use any HD44780U character pattern: According to the user application, handled in one of the two ways listed as follows.
   a. When unused character patterns are not programmed: If an unused character code is written into DDRAM, all its dots are lit. By not programing a character pattern, all of its bits become lit. (This is due to the EPROM being filled with 1s after it is erased.)
   b. When unused character patterns are programmed as 0s: Nothing is displayed even if unused character codes are written into DDRAM. (This is equivalent to a space.)

Table 3 Example of Correspondence between EPROM Address Data and Character Pattern (5 × 10 Dots)

EPROM Address												Data					
	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	O4	O3	O2	O1	LSB O0
									0	0	0	0	0	0	0	0	0
									0	0	0	1	0	0	0	0	0
									0	0	1	0	0	1	1	0	1
									0	0	1	1	1	0	0	1	1
									0	1	0	0	1	0	0	0	1
									0	1	0	1	1	0	0	0	1
									0	1	1	0	0	1	1	1	1
	0	1	0	1	0	0	1	0	0	1	1	1	0	0	0	0	1
									1	0	0	0	0	0	0	0	1
									1	0	0	1	0	0	0	0	1
									1	0	1	0	0	0	0	0	0
									1	0	1	1	0	0	0	0	0
									1	1	0	0	0	0	0	0	0
									1	1	0	1	0	0	0	0	0
									1	1	1	0	0	0	0	0	0
									1	1	1	1	0	0	0	0	0

← Cursor position

Character code

Line position

Notes:
1. EPROM addresses A11 to A3 correspond to a character code.
2. EPROM addresses A3 to A0 specify a line position of the character pattern.
3. EPROM data O4 to O0 correspond to character pattern data.
4. EPROM data O5 to O7 must be specified as 0.
5. A lit display position (black) corresponds to a 1.
6. Line 11 and the following lines must be blanked with 0s for a 5 × 10 dot character fonts.

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A00)

Upper 4 Bits / Lower 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)															
xxxx0001	(2)															
xxxx0010	(3)															
xxxx0011	(4)															
xxxx0100	(5)															
xxxx0101	(6)															
xxxx0110	(7)															
xxxx0111	(8)															
xxxx1000	(1)															
xxxx1001	(2)															
xxxx1010	(3)															
xxxx1011	(4)															
xxxx1100	(5)															
xxxx1101	(6)															
xxxx1110	(7)															
xxxx1111	(8)															

Note: The user can specify any pattern for character-generator RAM.

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A02)

Upper 4 Bits / Lower 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)															
xxxx0001	(2)															
xxxx0010	(3)															
xxxx0011	(4)															
xxxx0100	(5)															
xxxx0101	(6)															
xxxx0110	(7)															
xxxx0111	(8)															
xxxx1000	(1)															
xxxx1001	(2)															
xxxx1010	(3)															
xxxx1011	(4)															
xxxx1100	(5)															
xxxx1101	(6)															
xxxx1110	(7)															
xxxx1111	(8)															

Table 5 Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data)

For 5 × 8 dot character patterns

Character Codes (DDRAM data) 7 6 5 4 3 2 1 0 (High … Low)	CGRAM Address 5 4 3 2 1 0 (High … Low)	Character Patterns (CGRAM data) 7 6 5 4 3 2 1 0 (High … Low)	
0 0 0 0 * 0 0 0	0 0 0	0 0 0	* * * 1 1 1 1 0
			0 0 1 * * * 1 0 0 0 1
			0 1 0 * * * 1 0 0 0 1
			0 1 1 * * * 1 1 1 1 0
			1 0 0 * * * 1 0 1 0 0
			1 0 1 * * * 1 0 0 1 0
			1 1 0 * * * 1 0 0 0 1
			1 1 1 * * * 0 0 0 0 0
0 0 0 0 * 0 0 1	0 0 1	0 0 0	* * * 1 0 0 0 1
			0 0 1 * * * 0 1 0 1 0
			0 1 0 * * * 1 1 1 1 1
			0 1 1 * * * 0 0 1 0 0
			1 0 0 * * * 1 1 1 1 1
			1 0 1 * * * 0 0 1 0 0
			1 1 0 * * * 0 0 1 0 0
			1 1 1 * * * 0 0 0 0 0
		0 0 0	* * *
		0 0 1	* * *
0 0 0 0 * 1 1 1	1 1 1	1 0 0	* * *
		1 0 1	* * *
		1 1 0	* * *
		1 1 1	* * *

Character pattern (1)

Cursor position

Character pattern (2)

Cursor position

Notes: 1. Character code bits 0 to 2 correspond to CGRAM address bits 3 to 5 (3 bits: 8 types).

2. CGRAM address bits 0 to 2 designate the character pattern line position. The 8th line is the cursor position and its display is formed by a logical OR with the cursor.

   Maintain the 8th line data, corresponding to the cursor display position, at 0 as the cursor display.

   If the 8th line data is 1, 1 bits will light up the 8th line regardless of the cursor presence.

3. Character pattern row positions correspond to CGRAM data bits 0 to 4 (bit 4 being at the left).

4. As shown Table 5, CGRAM character patterns are selected when character code bits 4 to 7 are all 0. However, since character code bit 3 has no effect, the R display example above can be selected by either character code 00H or 08H.

5. 1 for CGRAM data corresponds to display selection and 0 to non-selection.

* Indicates no effect.

Table 5 Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data) (cont)

For 5 × 10 dot character patterns

Character Codes (DDRAM data)	CGRAM Address	Character Patterns (CGRAM data)	
7 6 5 4 3 2 1 0	5 4 3 2 1 0	7 6 5 4 3 2 1 0	
High Low	High Low	High Low	
0 0 0 0 * 0 0 *	0 0 0 1 0 0	* * * 0 0 0 0 0	Character pattern
		* * * 0 0 0 0 0	
		* * * 1 0 1 1 0	
		* * * 1 1 0 0 1	
		* * * 1 0 0 0 1	
		* * * 1 0 0 0 1	
		* * * 1 1 1 1 0	
		* * * 1 0 0 0 0	
		* * * 1 0 0 0 0	
		* * * 1 0 0 0 0	
		* * * 0 0 0 0 0	Cursor position
	0 0 1 1 1 0	* * * * * * * *	
	0 1 0 0 1 0	* * *	
	0 1 0 1 1 0	* * *	
	0 1 1 0 1 0	* * *	
	0 1 1 1 1 0	* * * * * * * *	
	1 0 0 0 0 0	* * *	
	1 0 0 0 1 0	* * *	
0 0 0 0 * 1 1 *	1 1 1 0 0 1	* * *	
	1 0 1 0	* * *	
	1 0 1 1 1 0	* * * * * * * *	
	1 1 0 0 1 0	* * *	
	1 1 0 1 1 0	* * *	
	1 1 1 0 1 0	* * * * * * * *	
	1 1 1 1 1 0	* * *	

Notes: 1. Character code bits 1 and 2 correspond to CGRAM address bits 4 and 5 (2 bits: 4 types).

2. CGRAM address bits 0 to 3 designate the character pattern line position. The 11th line is the cursor position and its display is formed by a logical OR with the cursor.
   Maintain the 11th line data corresponding to the cursor display positon at 0 as the cursor display.
   If the 11th line data is “1”, “1” bits will light up the 11th line regardless of the cursor presence.
   Since lines 12 to 16 are not used for display, they can be used for general data RAM.

3. Character pattern row positions are the same as 5 × 8 dot character pattern positions.

4. CGRAM character patterns are selected when character code bits 4 to 7 are all 0.
   However, since character code bits 0 and 3 have no effect, the P display example above can be selected by character codes 00H, 01H, 08H, and 09H.

5. 1 for CGRAM data corresponds to display selection and 0 to non-selection.

* Indicates no effect.

Timing Generation Circuit

The timing generation circuit generates timing signals for the operation of internal circuits such as DDRAM, CGROM and CGRAM. RAM read timing for display and internal operation timing by MPU access are generated separately to avoid interfering with each other. Therefore, when writing data to DDRAM, for example, there will be no undesirable interferences, such as flickering, in areas other than the display area.

Liquid Crystal Display Driver Circuit

The liquid crystal display driver circuit consists of 16 common signal drivers and 40 segment signal drivers. When the character font and number of lines are selected by a program, the required common signal drivers automatically output drive waveforms, while the other common signal drivers continue to output non-selection waveforms.

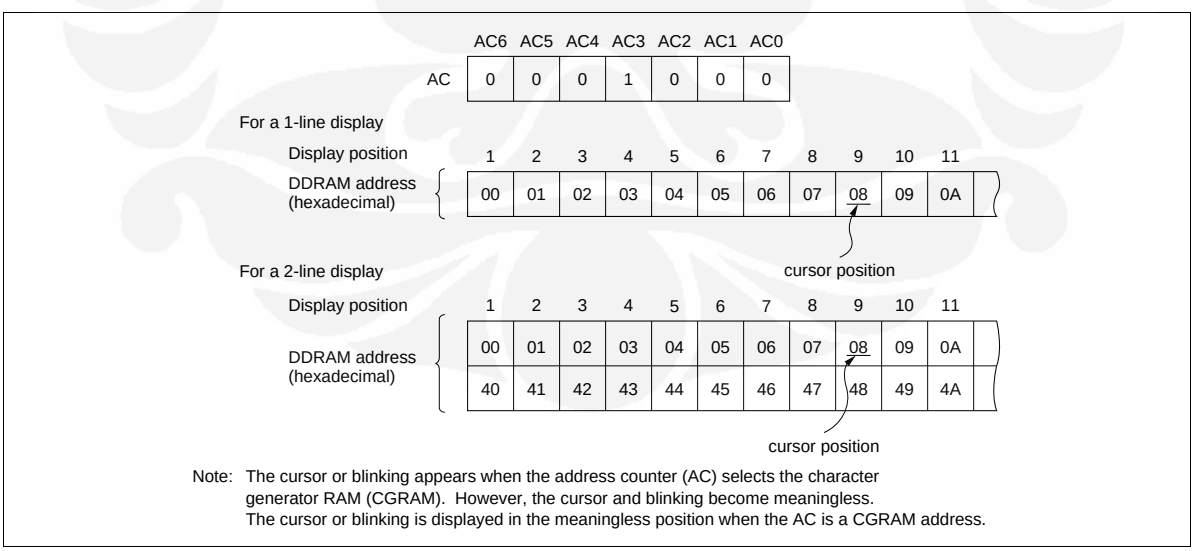
Sending serial data always starts at the display data character pattern corresponding to the last address of the display data RAM (DDRAM).

Since serial data is latched when the display data character pattern corresponding to the starting address enters the internal shift register, the HD44780U drives from the head display.

Cursor/Blink Control Circuit

The cursor/blink control circuit generates the cursor or character blinking. The cursor or the blinking will appear with the digit located at the display data RAM (DDRAM) address set in the address counter (AC).

For example (Figure 8), when the address counter is 08H, the cursor position is displayed at DDRAM address 08H.



Note: The cursor or blinking appears when the address counter (AC) selects the character generator RAM (CGRAM). However, the cursor and blinking become meaningless. The cursor or blinking is displayed in the meaningless position when the AC is a CGRAM address.

Figure 8 Cursor/Blink Display Example

Interfacing to the MPU

The HD44780U can send data in either two 4-bit operations or one 8-bit operation, thus allowing interfacing with 4- or 8-bit MPUs.

- For 4-bit interface data, only four bus lines (DB4 to DB7) are used for transfer. Bus lines DB0 to DB3 are disabled. The data transfer between the HD44780U and the MPU is completed after the 4-bit data has been transferred twice. As for the order of data transfer, the four high order bits (for 8-bit operation, DB4 to DB7) are transferred before the four low order bits (for 8-bit operation, DB0 to DB3).

  The busy flag must be checked (one instruction) after the 4-bit data has been transferred twice. Two more 4-bit operations then transfer the busy flag and address counter data.
- For 8-bit interface data, all eight bus lines (DB0 to DB7) are used.

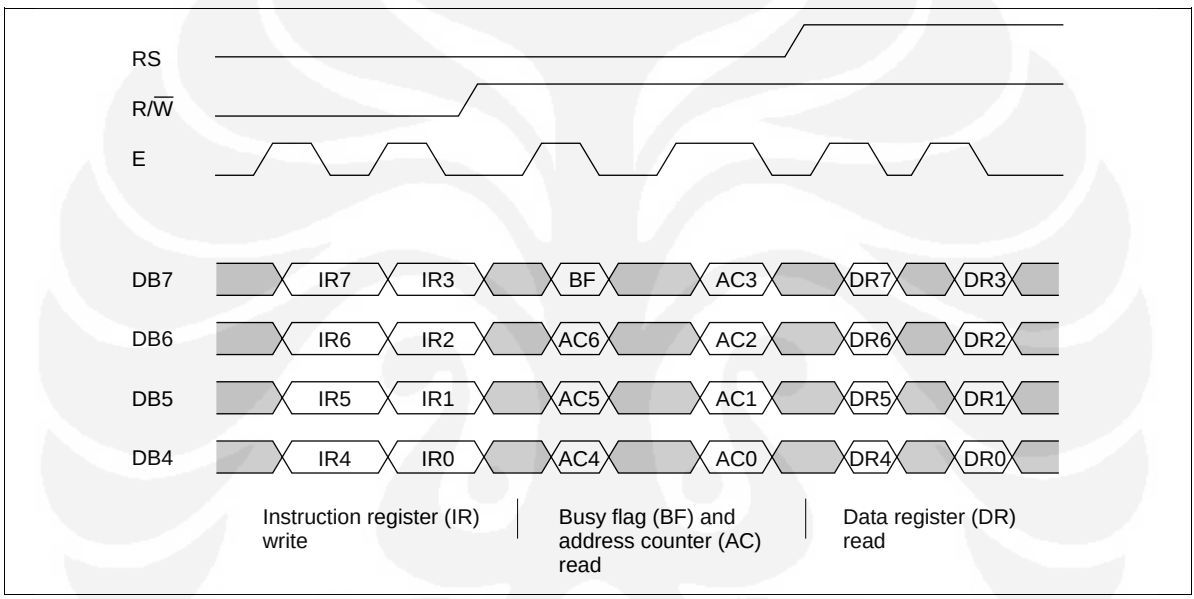


Figure 9 4-Bit Transfer Example

Reset Function

Initializing by Internal Reset Circuit

An internal reset circuit automatically initializes the HD44780U when the power is turned on. The following instructions are executed during the initialization. The busy flag (BF) is kept in the busy state until the initialization ends (BF = 1). The busy state lasts for 10 ms after V_{CC} rises to 4.5 V.

1. Display clear
2. Function set:
   DL = 1; 8-bit interface data
   N = 0; 1-line display
   F = 0; 5 × 8 dot character font
3. Display on/off control:
   D = 0; Display off
   C = 0; Cursor off
   B = 0; Blinking off
4. Entry mode set:
   I/D = 1; Increment by 1
   S = 0; No shift

Note: If the electrical characteristics conditions listed under the table Power Supply Conditions Using Internal Reset Circuit are not met, the internal reset circuit will not operate normally and will fail to initialize the HD44780U. For such a case, initial-ization must be performed by the MPU as explained in the section, Initializing by Instruction.

Instructions

Outline

Only the instruction register (IR) and the data register (DR) of the HD44780U can be controlled by the MPU. Before starting the internal operation of the HD44780U, control information is temporarily stored into these registers to allow interfacing with various MPUs, which operate at different speeds, or various peripheral control devices. The internal operation of the HD44780U is determined by signals sent from the MPU. These signals, which include register selection signal (RS), read/

write signal (R/$\overline{W}$), and the data bus (DB0 to DB7), make up the HD44780U instructions (Table 6). There are four categories of instructions that:

- Designate HD44780U functions, such as display format, data length, etc.
- Set internal RAM addresses
- Perform data transfer with internal RAM
- Perform miscellaneous functions

Normally, instructions that perform data transfer with internal RAM are used the most. However, auto-incrementation by 1 (or auto-decrementation by 1) of internal HD44780U RAM addresses after each data write can lighten the program load of the MPU. Since the display shift instruction (Table 11) can perform concurrently with display data write, the user can minimize system development time with maximum programming efficiency.

When an instruction is being executed for internal operation, no instruction other than the busy flag/address read instruction can be executed.

Because the busy flag is set to 1 while an instruction is being executed, check it to make sure it is 0 before sending another instruction from the MPU.

Note: Be sure the HD44780U is not in the busy state (BF = 0) before sending an instruction from the MPU to the HD44780U. If an instruction is sent without checking the busy flag, the time between the first instruction and next instruction will take much longer than the instruction time itself. Refer to Table 6 for the list of each instruc-tion execution time.

Table 6 Instructions

	Code											Execution Time (max) (when f_{cp} or f_{osc} is 270 kHz)
Instruction	**RS**	**R/$\overline{W}$**	**DB7**	**DB6**	**DB5**	**DB4**	**DB3**	**DB2**	**DB1**	**DB0**	**Description**	
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	
Return home	0	0	0	0	0	0	0	0	1	—	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.52 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 µs
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	37 µs
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.	37 µs
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (N), and character font (F).	37 µs
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.	37 µs
Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	37 µs
Read busy flag & address	0	1	BF	AC	AC	AC	AC	AC	AC	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 µs

Table 6 Instructions (cont)

Instruction	Code RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description	Execution Time (max) (when f_{cp} or f_{OSC} is 270 kHz)
Write data to CG or DDRAM	1	0	Write data								Writes data into DDRAM or CGRAM.	37 µs t_{ADD} = 4 µs*
Read data from CG or DDRAM	1	1	Read data								Reads data from DDRAM or CGRAM.	37 µs t_{ADD} = 4 µs*
	I/D = 1: Increment I/D = 0: Decrement S = 1: Accompanies display shift S/C = 1: Display shift S/C = 0: Cursor move R/L = 1: Shift to the right R/L = 0: Shift to the left DL = 1: 8 bits, DL = 0: 4 bits N = 1: 2 lines, N = 0: 1 line F = 1: 5 × 10 dots, F = 0: 5 × 8 dots BF = 1: Internally operating BF = 0: Instructions acceptable										DDRAM: Display data RAM CGRAM: Character generator RAM ACG: CGRAM address ADD: DDRAM address (corresponds to cursor address) AC: Address counter used for both DD and CGRAM addresses	Execution time changes when frequency changes Example: When f_{cp} or f_{OSC} is 250 kHz, $37\ \mu s \times \frac{270}{250} = 40\ \mu s$

Note: — indicates no effect.

* After execution of the CGRAM/DDRAM data write or read instruction, the RAM address counter is incremented or decremented by 1. The RAM address counter is updated after the busy flag turns off. In Figure 10, t_{ADD} is the time elapsed after the busy flag turns off until the address counter is updated.

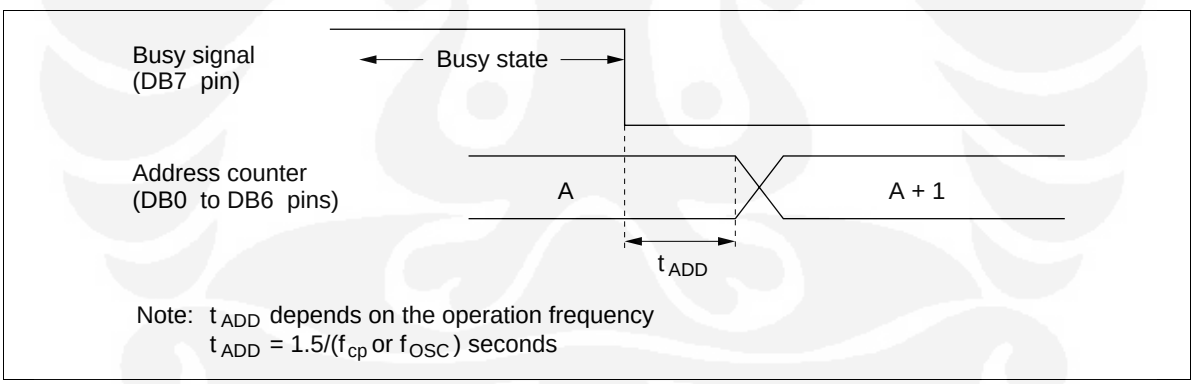


Figure 10 Address Counter Update

Instruction Description

Clear Display

Clear display writes space code 20H (character pattern for character code 20H must be a blank pattern) into all DDRAM addresses. It then sets DDRAM address 0 into the address counter, and returns the display to its original status if it was shifted. In other words, the display disappears and the cursor or blinking goes to the left edge of the display (in the first line if 2 lines are displayed). It also sets I/D to 1 (increment mode) in entry mode. S of entry mode does not change.

Return Home

Return home sets DDRAM address 0 into the address counter, and returns the display to its original status if it was shifted. The DDRAM contents do not change.

The cursor or blinking go to the left edge of the display (in the first line if 2 lines are displayed).

Entry Mode Set

I/D: Increments (I/D = 1) or decrements (I/D = 0) the DDRAM address by 1 when a character code is written into or read from DDRAM.

The cursor or blinking moves to the right when incremented by 1 and to the left when decremented by 1. The same applies to writing and reading of CGRAM.

S: Shifts the entire display either to the right (I/D = 0) or to the left (I/D = 1) when S is 1. The display does not shift if S is 0.

If S is 1, it will seem as if the cursor does not move but the display does. The display does not shift when reading from DDRAM. Also, writing into or reading out from CGRAM does not shift the display.

Display On/Off Control

D: The display is on when D is 1 and off when D is 0. When off, the display data remains in DDRAM, but can be displayed instantly by setting D to 1.

C: The cursor is displayed when C is 1 and not displayed when C is 0. Even if the cursor disappears, the function of I/D or other specifications will not change during display data write. The cursor is displayed using 5 dots in the 8th line for 5 × 8 dot character font selection and in the 11th line for the 5 × 10 dot character font selection (Figure 13).

B: The character indicated by the cursor blinks when B is 1 (Figure 13). The blinking is displayed as switching between all blank dots and displayed characters at a speed of 409.6-ms intervals when f_{cp} or f_{OSC} is 250 kHz. The cursor and blinking can be set to display simultaneously. (The blinking frequency changes according to f_{OSC} or the reciprocal of f_{cp}. For example, when f_{cp} is 270 kHz, 409.6 × 250/270 = 379.2 ms.)

Cursor or Display Shift

Cursor or display shift shifts the cursor position or display to the right or left without writing or reading display data (Table 7). This function is used to correct or search the display. In a 2-line display, the cursor moves to the second line when it passes the 40th digit of the first line. Note that the first and second line displays will shift at the same time.

When the displayed data is shifted repeatedly each line moves only horizontally. The second line display does not shift into the first line position.

The address counter (AC) contents will not change if the only action performed is a display shift.

Function Set

DL: Sets the interface data length. Data is sent or received in 8-bit lengths (DB7 to DB0) when DL is 1, and in 4-bit lengths (DB7 to DB4) when DL is 0.When 4-bit length is selected, data must be sent or received twice.

N: Sets the number of display lines.

F: Sets the character font.

Note: Perform the function at the head of the program before executing any instructions (except for the read busy flag and address instruction). From this point, the function set instruction cannot be executed unless the interface data length is changed.

Set CGRAM Address

Set CGRAM address sets the CGRAM address binary AAAAAA into the address counter.

Data is then written to or read from the MPU for CGRAM.

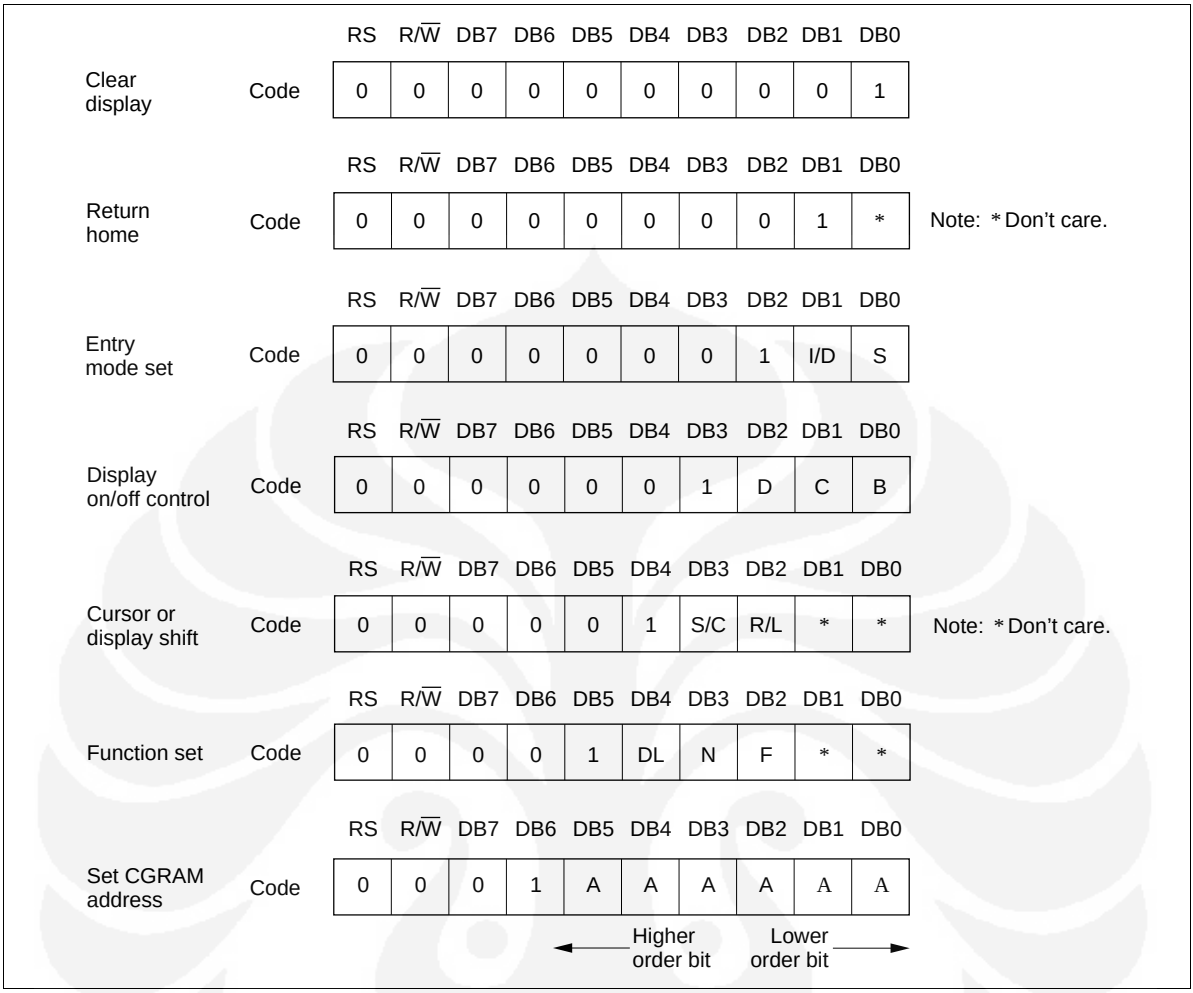


Figure 11 Instruction (1)

Set DDRAM Address

Set DDRAM address sets the DDRAM address binary AAAAAAA into the address counter.

Data is then written to or read from the MPU for DDRAM.

However, when N is 0 (1-line display), AAAAAAA can be 00H to 4FH. When N is 1 (2-line display), AAAAAAA can be 00H to 27H for the first line, and 40H to 67H for the second line.

Read Busy Flag and Address

Read busy flag and address reads the busy flag (BF) indicating that the system is now internally operating on a previously received instruction. If BF is 1, the internal operation is in progress. The next instruction will not be accepted until BF is reset to 0. Check the BF status before the next write operation. At the same time, the value of the address counter in binary AAAAAAA is read out. This address counter is used by both CG and DDRAM addresses, and its value is determined by the previous instruction. The address contents are the same as for instructions set CGRAM address and set DDRAM address.

Table 7 Shift Function

S/C	R/L	
0	0	Shifts the cursor position to the left. (AC is decremented by one.)
0	1	Shifts the cursor position to the right. (AC is incremented by one.)
1	0	Shifts the entire display to the left. The cursor follows the display shift.
1	1	Shifts the entire display to the right. The cursor follows the display shift.

Table 8 Function Set

N	F	No. of Display Lines	Character Font	Duty Factor	Remarks
0	0	1	5 × 8 dots	1/8	
0	1	1	5 × 10 dots	1/11	
1	*	2	5 × 8 dots	1/16	Cannot display two lines for 5 × 10 dot character font

Note: * Indicates don't care.

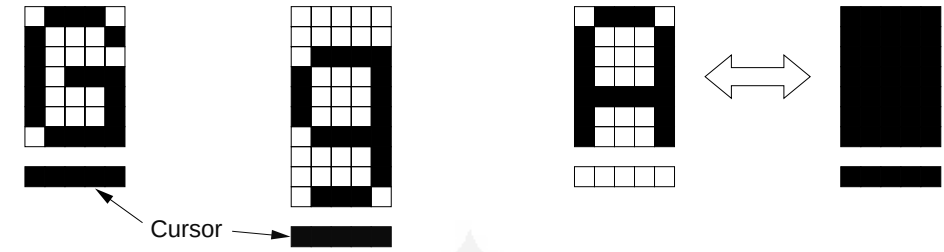


Figure 12 Cursor and Blinking

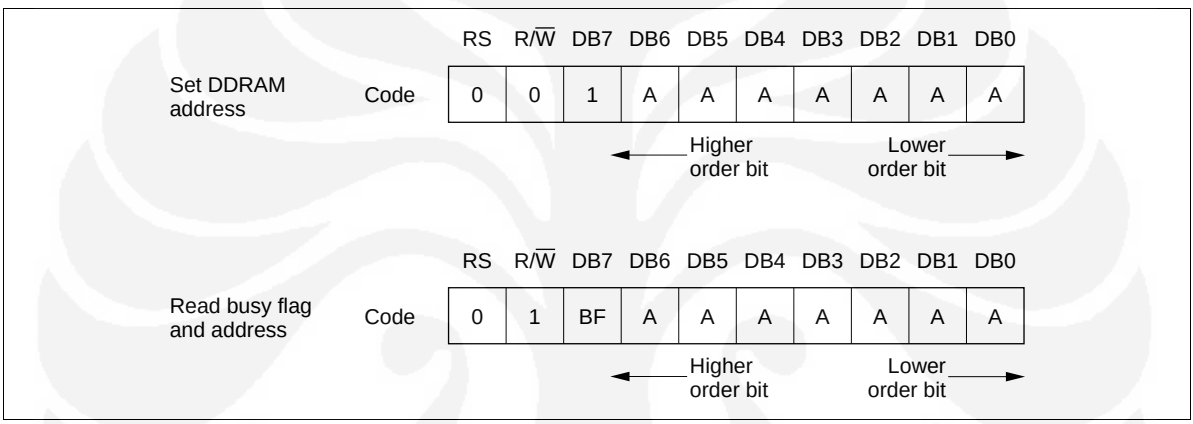


Figure 13 Instruction (2)

Write Data to CG or DDRAM

Write data to CG or DDRAM writes 8-bit binary data DDDDDDDD to CG or DDRAM.

To write into CG or DDRAM is determined by the previous specification of the CGRAM or DDRAM address setting. After a write, the address is automatically incremented or decremented by 1 according to the entry mode. The entry mode also determines the display shift.

Read Data from CG or DDRAM

Read data from CG or DDRAM reads 8-bit binary data DDDDDDDD from CG or DDRAM.

The previous designation determines whether CG or DDRAM is to be read. Before entering this read instruction, either CGRAM or DDRAM address set instruction must be executed. If not executed, the first read data will be invalid. When serially executing read instructions, the next address data is normally read from the second read. The address set instructions need not be executed just before this read instruction when shifting the cursor by the cursor shift instruction (when reading out DDRAM). The operation of the cursor shift instruction is the same as the set DDRAM address instruction.

After a read, the entry mode automatically increases or decreases the address by 1. However, display shift is not executed regardless of the entry mode.

Note: The address counter (AC) is automatically incremented or decremented by 1 after the write instructions to CGRAM or DDRAM are executed. The RAM data selected by the AC cannot be read out at this time even if read instructions are executed. Therefore, to correctly read data, execute either the address set instruction or cursor shift instruction (only with DDRAM), then just before reading the desired data, execute the read instruction from the second time the read instruction is sent.

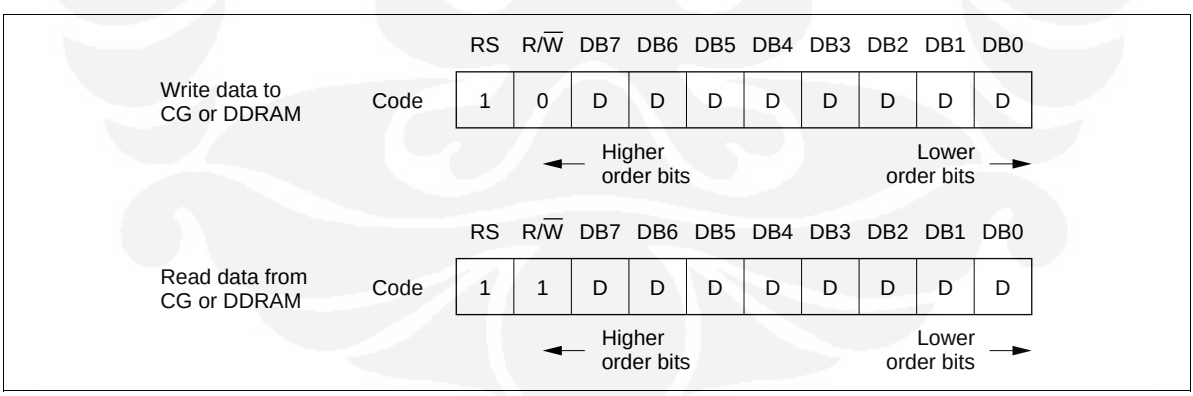


Figure 14 Instruction (3)

Interfacing the HD44780U

Interface to MPUs

- Interfacing to an 8-bit MPU

  See Figure 16 for an example of using a I/O port (for a single-chip microcomputer) as an interface device.

  In this example, P30 to P37 are connected to the data bus DB0 to DB7, and P75 to P77 are connected to E, R/$\overline{W}$, and RS, respectively.

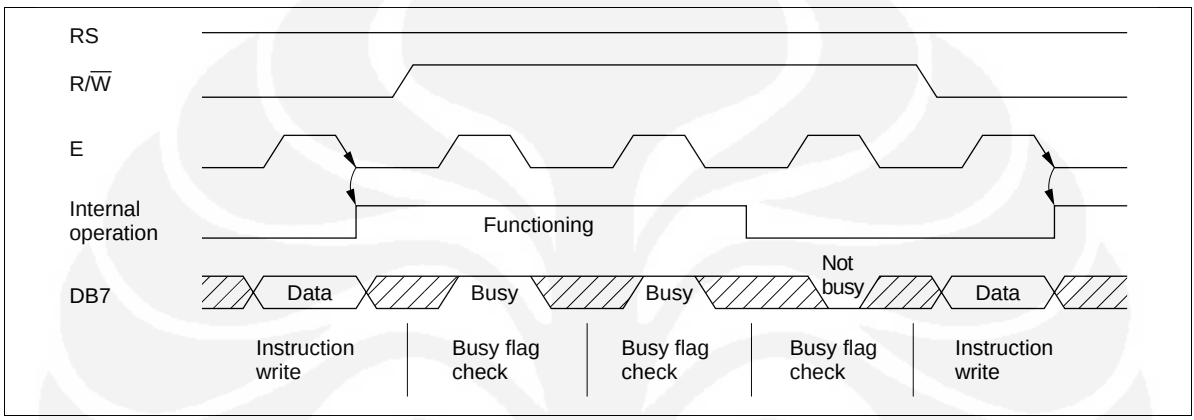


Figure 15 Example of Busy Flag Check Timing Sequence

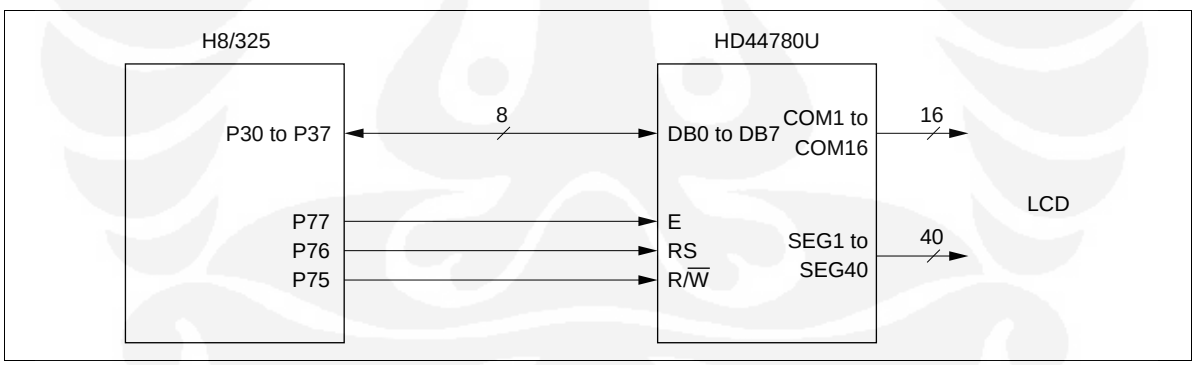


Figure 16 H8/325 Interface (Single-Chip Mode)

- Interfacing to a 4-bit MPU

  The HD44780U can be connected to the I/O port of a 4-bit MPU. If the I/O port has enough bits, 8-bit data can be transferred. Otherwise, one data transfer must be made in two operations for 4-bit data. In this case, the timing sequence becomes somewhat complex. (See Figure 17.)

  See Figure 18 for an interface example to the HMCS4019R.

  Note that two cycles are needed for the busy flag check as well as for the data transfer. The 4-bit operation is selected by the program.

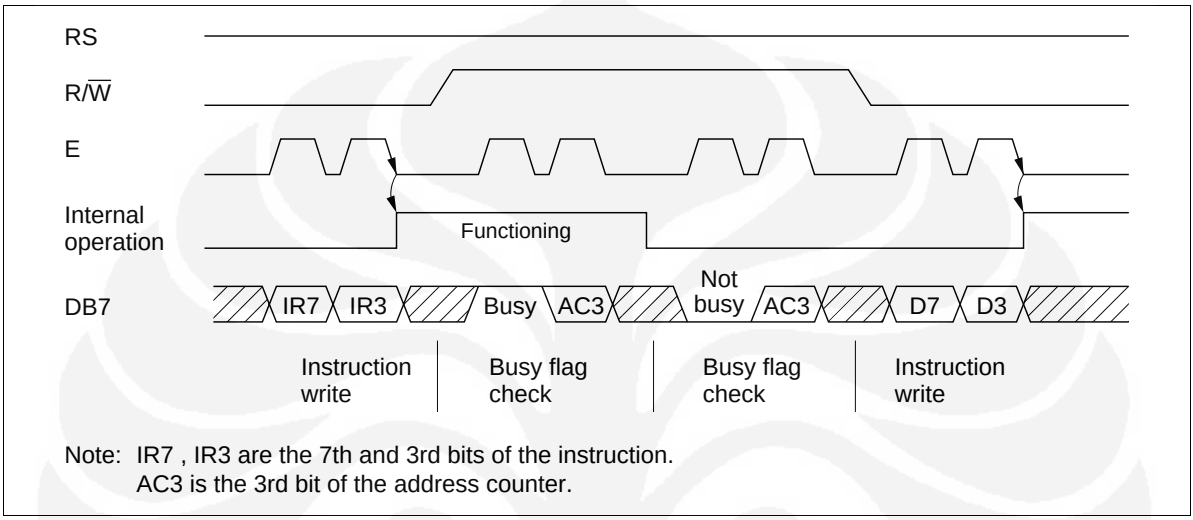


Figure 17 Example of 4-Bit Data Transfer Timing Sequence

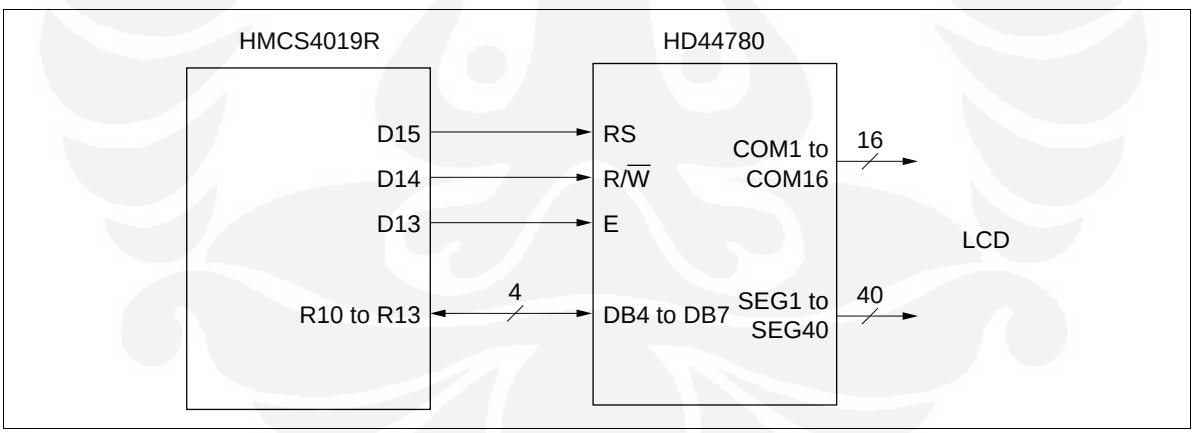


Figure 18 Example of Interface to HMCS4019R

Interface to Liquid Crystal Display

Character Font and Number of Lines: The HD44780U can perform two types of displays, 5 × 8 dot and 5 × 10 dot character fonts, each with a cursor.

Up to two lines are displayed for 5 × 8 dots and one line for 5 × 10 dots. Therefore, a total of three types of common signals are available (Table 9).

The number of lines and font types can be selected by the program. (See Table 6, Instructions.)

Connection to HD44780 and Liquid Crystal Display: See Figure 19 for the connection examples.

Table 9 Common Signals

Number of Lines	Character Font	Number of Common Signals	Duty Factor
1	5 × 8 dots + cursor	8	1/8
1	5 × 10 dots + cursor	11	1/11
2	5 × 8 dots + cursor	16	1/16

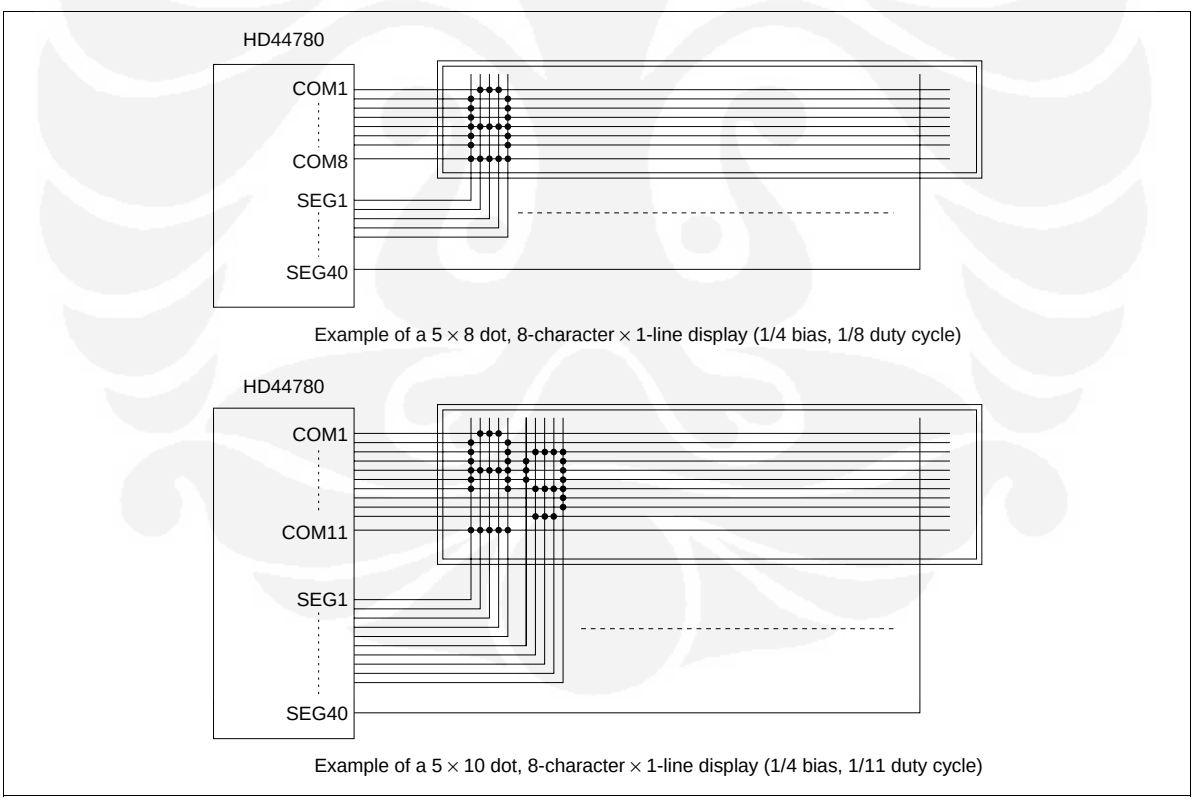


Figure 19 Liquid Crystal Display and HD44780 Connections

Since five segment signal lines can display one digit, one HD44780U can display up to 8 digits for a 1-line display and 16 digits for a 2-line display.

The examples in Figure 19 have unused common signal pins, which always output non-selection waveforms. When the liquid crystal display panel has unused extra scanning lines, connect the extra scanning lines to these common signal pins to avoid any undesirable effects due to crosstalk during the floating state.

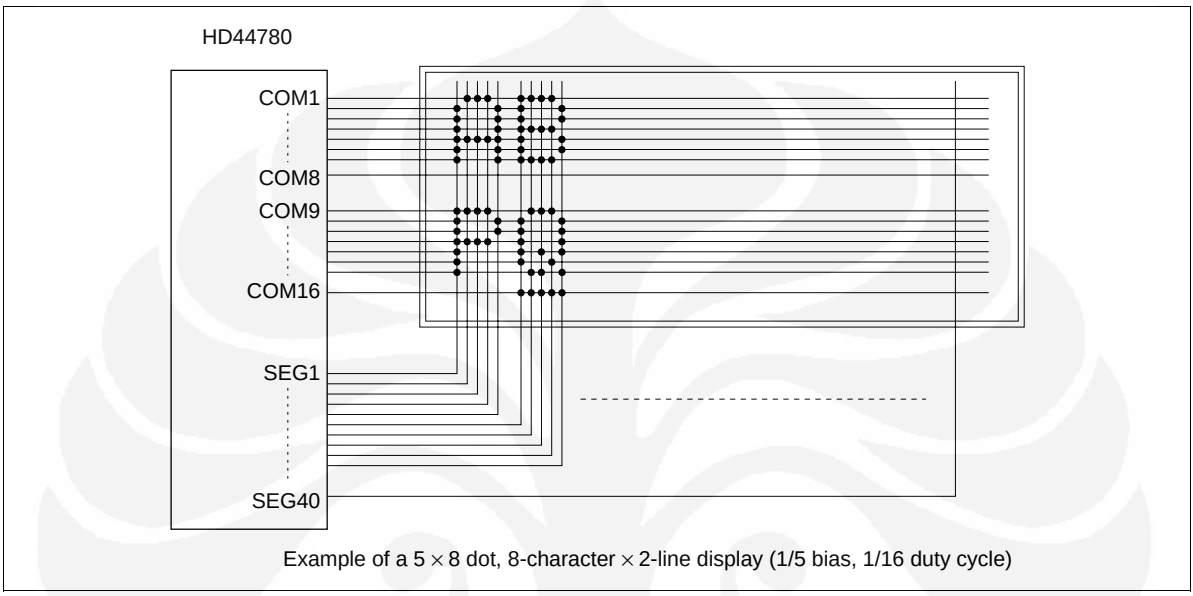


Figure 19 Liquid Crystal Display and HD44780 Connections (cont)

Connection of Changed Matrix Layout: In the preceding examples, the number of lines correspond to the scanning lines. However, the following display examples (Figure 20) are made possible by altering the matrix layout of the liquid crystal display panel. In either case, the only change is the layout. The display characteristics and the number of liquid crystal display characters depend on the number of common signals or on duty factor. Note that the display data RAM (DDRAM) addresses for 4 characters × 2 lines and for 16 characters × 1 line are the same as in Figure 19.

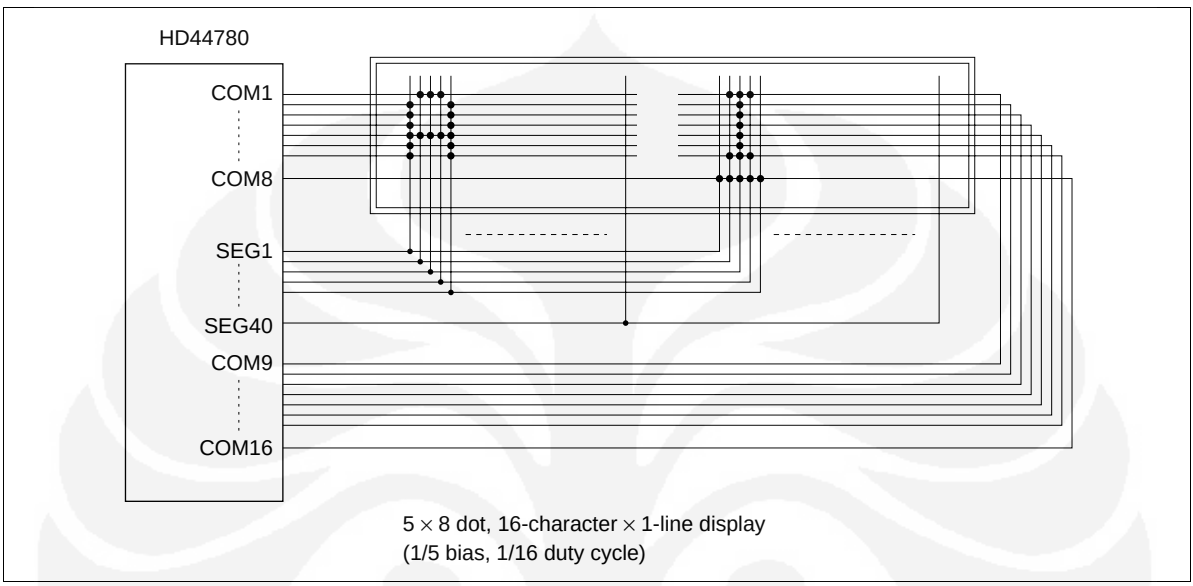


Figure 20 Changed Matrix Layout Displays

Power Supply for Liquid Crystal Display Drive

Various voltage levels must be applied to pins V1 to V5 of the HD44780U to obtain the liquid crystal display drive waveforms. The voltages must be changed according to the duty factor (Table 10).

VLCD is the peak value for the liquid crystal display drive waveforms, and resistance dividing provides voltages V1 to V5 (Figure 21).

Table 10 Duty Factor and Power Supply for Liquid Crystal Display Drive

	Duty Factor	
	1/8, 1/11	1/16
	Bias	
Power Supply	1/4	1/5
V1	V_{cc}–1/4 VLCD	V_{cc}–1/5 VLCD
V2	V_{cc}–1/2 VLCD	V_{cc}–2/5 VLCD
V3	V_{cc}–1/2 VLCD	V_{cc}–3/5 VLCD
V4	V_{cc}–3/4 VLCD	V_{cc}–4/5 VLCD
V5	V_{cc}–VLCD	V_{cc}–VLCD

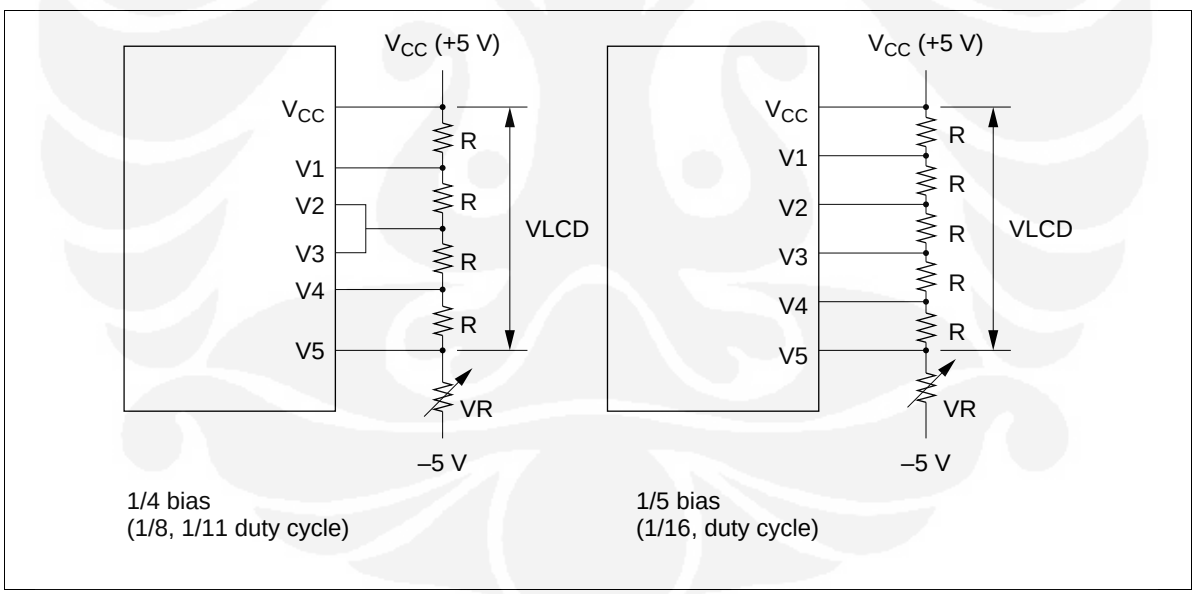


Figure 21 Drive Voltage Supply Example

Relationship between Oscillation Frequency and Liquid Crystal Display Frame Frequency

The liquid crystal display frame frequencies of Figure 22 apply only when the oscillation frequency is 270 kHz (one clock pulse of 3.7 μs).

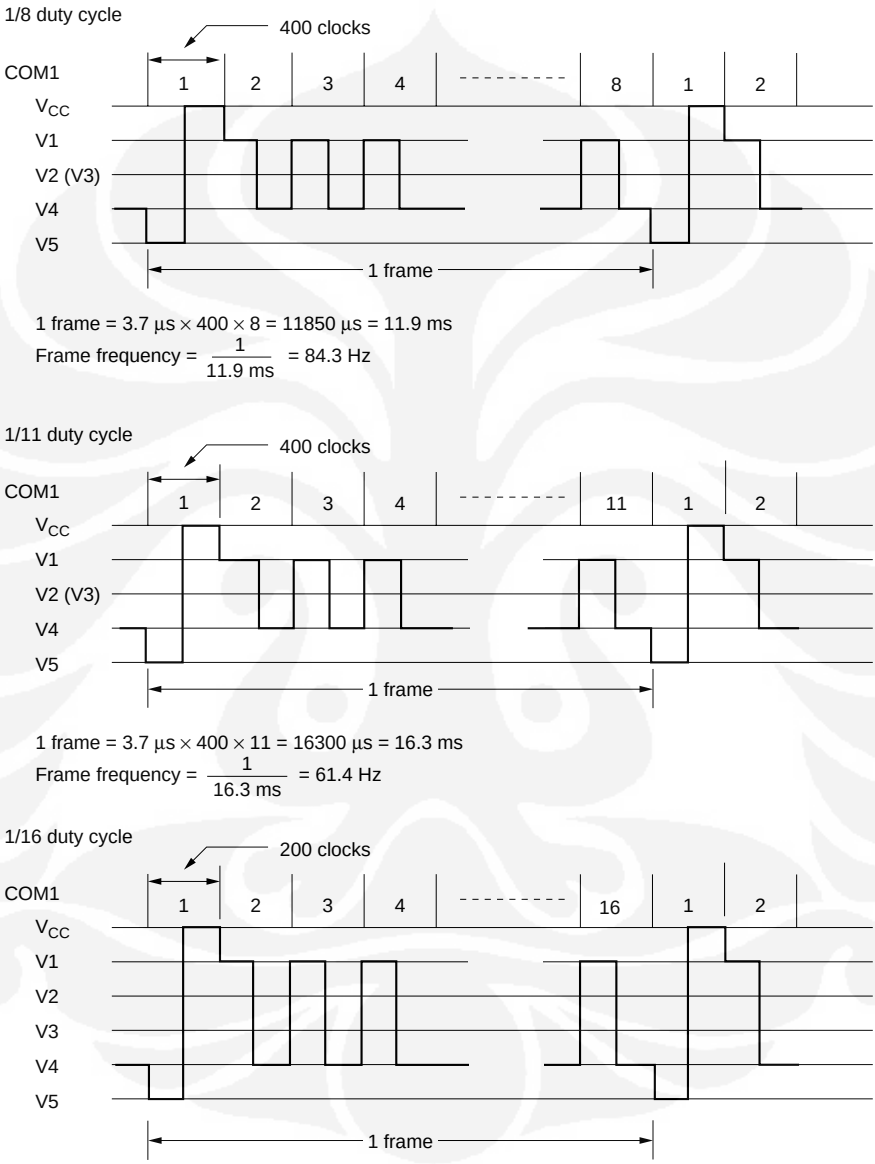


Figure 22 Frame Frequency

Instruction and Display Correspondence

- 8-bit operation, 8-digit × 1-line display with internal reset

  Refer to Table 11 for an example of an 8-digit × 1-line display in 8-bit operation. The HD44780U functions must be set by the function set instruction prior to the display. Since the display data RAM can store data for 80 characters, as explained before, the RAM can be used for displays such as for advertising when combined with the display shift operation.

  Since the display shift operation changes only the display position with DDRAM contents unchanged, the first display data entered into DDRAM can be output when the return home operation is performed.

- 4-bit operation, 8-digit × 1-line display with internal reset

  The program must set all functions prior to the 4-bit operation (Table 12). When the power is turned on, 8-bit operation is automatically selected and the first write is performed as an 8-bit operation. Since DB0 to DB3 are not connected, a rewrite is then required. However, since one operation is completed in two accesses for 4-bit operation, a rewrite is needed to set the functions (see Table 12). Thus, DB4 to DB7 of the function set instruction is written twice.

- 8-bit operation, 8-digit × 2-line display

  For a 2-line display, the cursor automatically moves from the first to the second line after the 40th digit of the first line has been written. Thus, if there are only 8 characters in the first line, the DDRAM address must be again set after the 8th character is completed. (See Table 13.) Note that the display shift operation is performed for the first and second lines. In the example of Table 13, the display shift is performed when the cursor is on the second line. However, if the shift operation is performed when the cursor is on the first line, both the first and second lines move together. If the shift is repeated, the display of the second line will not move to the first line. The same display will only shift within its own line for the number of times the shift is repeated.

Note: When using the internal reset, the electrical characteristics in the Power Supply Conditions Using Internal Reset Circuit table must be satisfied. If not, the HD44780U must be initialized by instructions. See the section, Initializing by Instruction.

Table 11 8-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset

Step No.	Instruction RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	Operation
1	Power supply on (the HD44780U is initialized by the internal reset circuit)											Initialized. No display.
2	Function set	0	0	0	0	1	1	0	0	*	*		Sets to 8-bit operation and selects 1-line display and 5 × 8 dot character font. (Number of display lines and character fonts cannot be changed after step #2.)
3	Display on/off control	0	0	0	0	0	0	1	1	1	0	_	Turns on display and cursor. Entire display is in space mode because of initialization.
4	Entry mode set	0	0	0	0	0	0	0	1	1	0	_	Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
5	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	0	0	0	H_	Writes H. DDRAM has already been selected by initialization when the power was turned on. The cursor is incremented by one and shifted to the right.
6	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	0	0	1	HI_	Writes I.
7	. . . . .					.						. . . . .	
8	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	0	0	1	HITACHI_	Writes I.
9	Entry mode set	0	0	0	0	0	0	0	1	1	1	HITACHI_	Sets mode to shift display at the time of write.
10	Write data to CGRAM/DDRAM	1	0	0	0	1	0	0	0	0	0	ITACHI _	Writes a space.

Table 11 8-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset (cont)

Step No.	Instruction RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	Operation
11	Write data to CGRAM/DDRAM 1	0	0	1	0	0	1	1	0	1	TACHI M_	Writes M.
12	. . . . .										. . . . .	
13	Write data to CGRAM/DDRAM 1	0	0	1	0	0	1	1	1	1	MICROKO_	Writes O.
14	Cursor or display shift 0	0	0	0	0	1	0	0	*	*	MICROKO	Shifts only the cursor position to the left.
15	Cursor or display shift 0	0	0	0	0	1	0	0	*	*	MICROKO	Shifts only the cursor position to the left.
16	Write data to CGRAM/DDRAM 1	0	0	1	0	0	0	0	1	1	ICROCO	Writes C over K. The display moves to the left.
17	Cursor or display shift 0	0	0	0	0	1	1	1	*	*	MICROCO	Shifts the display and cursor position to the right.
18	Cursor or display shift 0	0	0	0	0	1	0	1	*	*	MICROCO_	Shifts the display and cursor position to the right.
19	Write data to CGRAM/DDRAM 1	0	0	1	0	0	1	1	0	1	ICROCOM_	Writes M.
20	. . . . .										. . . . .	
21	Return home 0	0	0	0	0	0	0	0	1	0	HITACHI	Returns both display and cursor to the original position (address 0).

Table 12 4-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset

Step No.	Instruction RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	Display	Operation
1	Power supply on (the HD44780U is initialized by the internal reset circuit)							Initialized. No display.
2	Function set							Sets to 4-bit operation. In this case, operation is handled as 8 bits by initialization, and only this instruction completes with one write.
	0	0	0	0	1	0		
3	Function set							Sets 4-bit operation and selects 1-line display and 5 × 8 dot character font. 4-bit operation starts from this step and resetting is necessary. (Number of display lines and character fonts cannot be changed after step #3.)
	0	0	0	0	1	0		
	0	0	0	0	*	*		
4	Display on/off control						_	Turns on display and cursor. Entire display is in space mode because of initialization.
	0	0	0	0	0	0		
	0	0	1	1	1	0		
5	Entry mode set						_	Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
	0	0	0	0	0	0		
	0	0	0	1	1	0		
6	Write data to CGRAM/DDRAM						H_	Writes H. The cursor is incremented by one and shifts to the right.
	1	0	0	1	0	0		
	1	0	1	0	0	0		

Note: The control is the same as for 8-bit operation beyond step #6.

Table 13 8-Bit Operation, 8-Digit × 2-Line Display Example with Internal Reset

Step No.	Instruction RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	Operation
1	Power supply on (the HD44780U is initialized by the internal reset circuit)											Initialized. No display.
2	Function set											Sets to 8-bit operation and selects 2-line display and 5 × 8 dot character font.
	0	0	0	0	1	1	1	0	*	*		
3	Display on/off control											Turns on display and cursor. All display is in space mode because of initialization.
	0	0	0	0	0	0	1	1	1	0	_	
4	Entry mode set											Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
	0	0	0	0	0	0	0	1	1	0	_	
5	Write data to CGRAM/DDRAM											Writes H. DDRAM has already been selected by initialization when the power was turned on. The cursor is incremented by one and shifted to the right.
	1	0	0	1	0	0	1	0	0	0	H_	
6	. . . . .					. . . . .					. . . . .	
7	Write data to CGRAM/DDRAM											Writes I.
	1	0	0	1	0	0	1	0	0	1	HITACHI_	
8	Set DDRAM address											Sets DDRAM address so that the cursor is positioned at the head of the second line.
	0	0	1	1	0	0	0	0	0	0	HITACHI _	

Table 13 8-Bit Operation, 8-Digit × 2-Line Display Example with Internal Reset (cont)

Step No.	Instruction RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	Operation
9	Write data to CGRAM/DDRAM 1	0	0	1	0	0	1	1	0	1	HITACHI M_	Writes M.
10	.	.	.	.	.	.	.	.	.	.	.	
11	Write data to CGRAM/DDRAM 1	0	0	1	0	0	1	1	1	1	HITACHI MICROCO_	Writes O.
12	Entry mode set 0	0	0	0	0	0	0	1	1	1	HITACHI MICROCO_	Sets mode to shift display at the time of write.
13	Write data to CGRAM/DDRAM 1	0	0	1	0	0	1	1	0	1	ITACHI ICROCOM_	Writes M. Display is shifted to the left. The first and second lines both shift at the same time.
14	.	.	.	.	.	.	.	.	.	.	.	
15	Return home 0	0	0	0	0	0	0	0	1	0	HITACHI MICROCOM	Returns both display and cursor to the original position (address 0).

Initializing by Instruction

If the power supply conditions for correctly operating the internal reset circuit are not met, initialization by instructions becomes necessary.

Refer to Figures 23 and 24 for the procedures on 8-bit and 4-bit initializations, respectively.

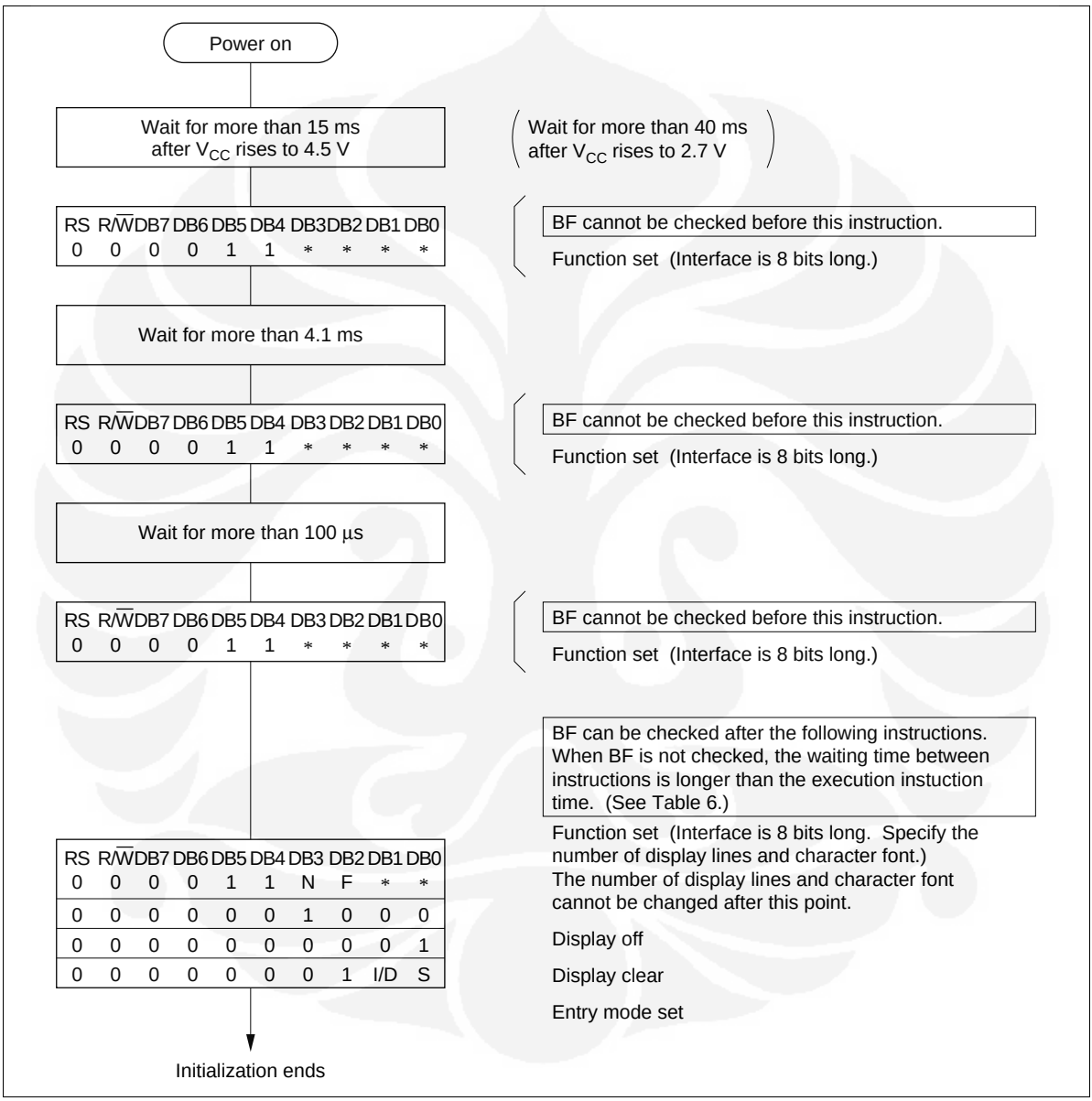


Figure 23 8-Bit Interface

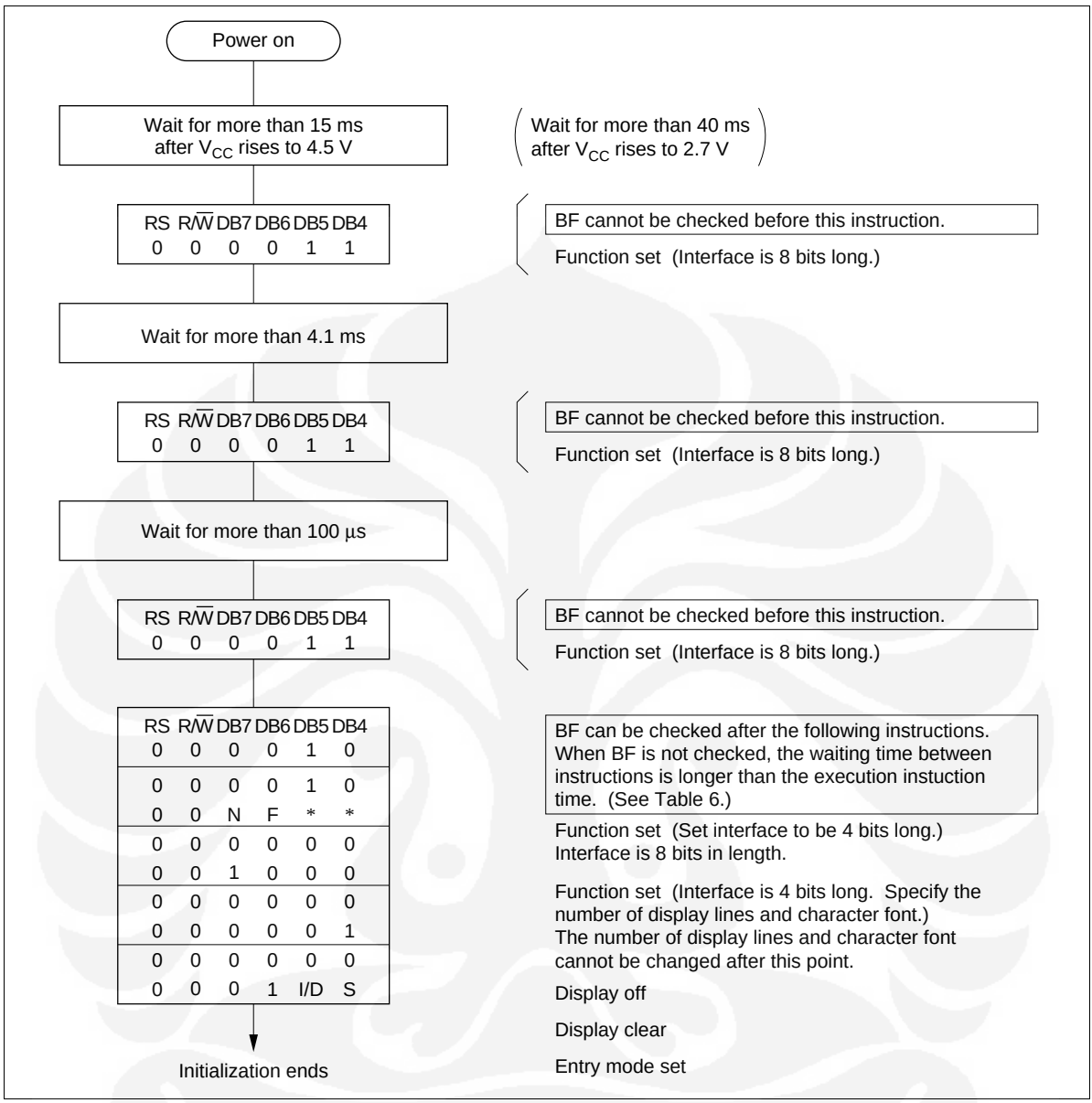


Figure 24 4-Bit Interface

Absolute Maximum Ratings*

Item	Symbol	Value	Unit	Notes
Power supply voltage (1)	V_{CC}–GND	–0.3 to +7.0	V	1
Power supply voltage (2)	V_{CC}–V5	–0.3 to +13.0	V	1, 2
Input voltage	Vt	–0.3 to V_{CC} +0.3	V	1
Operating temperature	T_{opr}	–30 to +75	°C	
Storage temperature	T_{stg}	–55 to +125	°C	4

Note: * If the LSI is used above these absolute maximum ratings, it may become permanently damaged. Using the LSI within the following electrical characteristic limits is strongly recommended for normal operation. If these electrical characteristic conditions are also exceeded, the LSI will malfunction and cause poor reliability.

DC Characteristics (V_{CC} = 2.7 to 4.5 V, T_a = –30 to +75°C*[3])

Item	Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
Input high voltage (1) (except OSC1)	VIH1	$0.7V_{CC}$	—	V_{CC}	V		6
Input low voltage (1) (except OSC1)	VIL1	–0.3	—	0.55	V		6
Input high voltage (2) (OSC1)	VIH2	$0.7V_{CC}$	—	V_{CC}	V		15
Input low voltage (2) (OSC1)	VIL2	—	—	$0.2V_{CC}$	V		15
Output high voltage (1) (DB0–DB7)	VOH1	$0.75V_{CC}$	—	—	V	$–I_{OH}$ = 0.1 mA	7
Output low voltage (1) (DB0–DB7)	VOL1	—	—	$0.2V_{CC}$	V	I_{OL} = 0.1 mA	7
Output high voltage (2) (except DB0–DB7)	VOH2	$0.8V_{CC}$	—	—	V	$–I_{OH}$ = 0.04 mA	8
Output low voltage (2) (except DB0–DB7)	VOL2	—	—	$0.2V_{CC}$	V	I_{OL} = 0.04 mA	8
Driver on resistance (COM)	R_{COM}	—	2	20	kΩ	±Id = 0.05 mA, VLCD = 4 V	13
Driver on resistance (SEG)	R_{SEG}	—	2	30	kΩ	±Id = 0.05 mA, VLCD = 4 V	13
Input leakage current	I_{LI}	–1	—	1	µA	VIN = 0 to V_{CC}	9
Pull-up MOS current (DB0–DB7, RS, R/$\overline{W}$)	$–I_p$	10	50	120	µA	V_{CC} = 3 V	
Power supply current	I_{CC}	—	150	300	µA	R_f oscillation, external clock V_{CC} = 3 V, f_{OSC} = 270 kHz	10, 14
LCD voltage	VLCD1	3.0	—	11.0	V	V_{CC}–V5, 1/5 bias	16
	VLCD2	3.0	—	11.0	V	V_{CC}–V5, 1/4 bias	16

Note: * Refer to the Electrical Characteristics Notes section following these tables.

AC Characteristics (V_{CC} = 2.7 to 4.5 V, T_a = –30 to +75°C*[3])

Clock Characteristics

Item		Symbol	Min	Typ	Max	Unit	Test Condition	Note*
External clock operation	External clock frequency	f_{cp}	125	250	350	kHz		11
	External clock duty	Duty	45	50	55	%		
	External clock rise time	t_{rcp}	—	—	0.2	µs		
	External clock fall time	t_{fcp}	—	—	0.2	µs		
R_f oscillation	Clock oscillation frequency	f_{OSC}	190	270	350	kHz	R_f = 75 kΩ, V_{CC} = 3 V	12

Note: * Refer to the Electrical Characteristics Notes section following these tables.

Bus Timing Characteristics

Write Operation

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cycE}	1000	—	—	ns	Figure 25
Enable pulse width (high level)	PW_{EH}	450	—	—		
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	25		
Address set-up time (RS, R/$\overline{W}$ to E)	t_{AS}	60	—	—		
Address hold time	t_{AH}	20	—	—		
Data set-up time	t_{DSW}	195	—	—		
Data hold time	t_H	10	—	—		

Read Operation

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cycE}	1000	—	—	ns	Figure 26
Enable pulse width (high level)	PW_{EH}	450	—	—		
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	25		
Address set-up time (RS, R/$\overline{W}$ to E)	t_{AS}	60	—	—		
Address hold time	t_{AH}	20	—	—		
Data delay time	t_{DDR}	—	—	360		
Data hold time	t_{DHR}	5	—	—		

Interface Timing Characteristics with External Driver

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Clock pulse width	High level	t_{CWH}	800	—	—	ns	Figure 27
	Low level	t_{CWL}	800	—	—		
Clock set-up time		t_{CSU}	500	—	—		
Data set-up time		t_{SU}	300	—	—		
Data hold time		t_{DH}	300	—	—		
M delay time		t_{DM}	–1000	—	1000		
Clock rise/fall time		t_{ct}	—	—	200		

Power Supply Conditions Using Internal Reset Circuit

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Power supply rise time	t_{rCC}	0.1	—	10	ms	Figure 28
Power supply off time	t_{OFF}	1	—	—		

DC Characteristics (V_{CC} = 4.5 to 5.5 V, T_a = –30 to +75°C*[3])

Item	Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
Input high voltage (1) (except OSC1)	VIH1	2.2	—	V_{CC}	V		6
Input low voltage (1) (except OSC1)	VIL1	–0.3	—	0.6	V		6
Input high voltage (2) (OSC1)	VIH2	V_{CC}–1.0	—	V_{CC}	V		15
Input low voltage (2) (OSC1)	VIL2	—	—	1.0	V		15
Output high voltage (1) (DB0–DB7)	VOH1	2.4	—	—	V	$–I_{OH}$ = 0.205 mA	7
Output low voltage (1) (DB0–DB7)	VOL1	—	—	0.4	V	I_{OL} = 1.2 mA	7
Output high voltage (2) (except DB0–DB7)	VOH2	0.9 V_{CC}	—	—	V	$–I_{OH}$ = 0.04 mA	8
Output low voltage (2) (except DB0–DB7)	VOL2	—	—	0.1 V_{CC}	V	I_{OL} = 0.04 mA	8
Driver on resistance (COM)	RCOM	—	2	20	kΩ	±Id = 0.05 mA, VLCD = 4 V	13
Driver on resistance (SEG)	RSEG	—	2	30	kΩ	±Id = 0.05 mA, VLCD = 4 V	13
Input leakage current	I_{LI}	–1	—	1	µA	VIN = 0 to V_{CC}	9
Pull-up MOS current (DB0–DB7, RS, R/$\overline{W}$)	$–I_p$	50	125	250	µA	V_{CC} = 5 V	
Power supply current	I_{CC}	—	350	600	µA	R_f oscillation, external clock V_{CC} = 5 V, f_{OSC} = 270 kHz	10, 14
LCD voltage	VLCD1	3.0	—	11.0	V	V_{CC}–V5, 1/5 bias	16
	VLCD2	3.0	—	11.0	V	V_{CC}–V5, 1/4 bias	16

Note: * Refer to the Electrical Characteristics Notes section following these tables.

AC Characteristics (V_{CC} = 4.5 to 5.5 V, T_a = –30 to +75°C*[3])

Clock Characteristics

Item		Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
External clock operation	External clock frequency	f_{cp}	125	250	350	kHz		11
	External clock duty	Duty	45	50	55	%		11
	External clock rise time	t_{rcp}	—	—	0.2	µs		11
	External clock fall time	t_{fcp}	—	—	0.2	µs		11
R_f oscillation	Clock oscillation frequency	f_{OSC}	190	270	350	kHz	R_f = 91 kΩ V_{CC} = 5.0 V	12

Note: * Refer to the Electrical Characteristics Notes section following these tables.

Bus Timing Characteristics

Write Operation

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cycE}	500	—	—	ns	Figure 25
Enable pulse width (high level)	PW_{EH}	230	—	—		
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	20		
Address set-up time (RS, R/$\overline{W}$ to E)	t_{AS}	40	—	—		
Address hold time	t_{AH}	10	—	—		
Data set-up time	t_{DSW}	80	—	—		
Data hold time	t_H	10	—	—		

Read Operation

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cycE}	500	—	—	ns	Figure 26
Enable pulse width (high level)	PW_{EH}	230	—	—		
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	20		
Address set-up time (RS, R/$\overline{W}$ to E)	t_{AS}	40	—	—		
Address hold time	t_{AH}	10	—	—		
Data delay time	t_{DDR}	—	—	160		
Data hold time	t_{DHR}	5	—	—		

Interface Timing Characteristics with External Driver

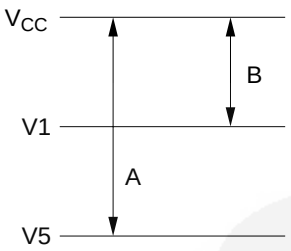
Item		Symbol	Min	Typ	Max	Unit	Test Condition
Clock pulse width	High level	t_{CWH}	800	—	—	ns	Figure 27
	Low level	t_{CWL}	800	—	—		
Clock set-up time		t_{CSU}	500	—	—		
Data set-up time		t_{SU}	300	—	—		
Data hold time		t_{DH}	300	—	—		
M delay time		t_{DM}	–1000	—	1000		
Clock rise/fall time		t_{ct}	—	—	100		

Power Supply Conditions Using Internal Reset Circuit

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Power supply rise time	t_{rCC}	0.1	—	10	ms	Figure 28
Power supply off time	t_{OFF}	1	—	—		

Electrical Characteristics Notes

1. All voltage values are referred to GND = 0 V.

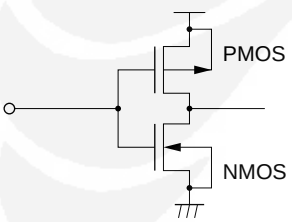


$A = V_{CC} - V5$
$B = V_{CC} - V1$
$A \geq 1.5$ V
$B \leq 0.25 \times A$

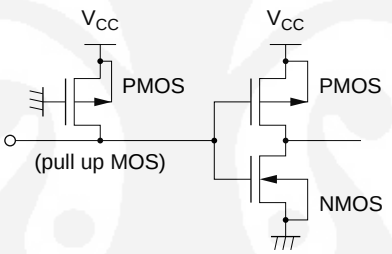
The conditions of V1 and V5 voltages are for proper operation of the LSI and not for the LCD output level. The LCD drive voltage condition for the LCD output level is specified as LCD voltage VLCD.

2. $V_{CC} \geq V1 \geq V2 \geq V3 \geq V4 \geq V5$ must be maintained.
3. For die products, specified at 75°C.
4. For die products, specified by the die shipment specification.
5. The following four circuits are I/O pin configurations except for liquid crystal display output.

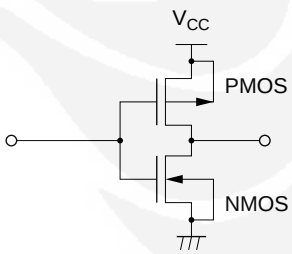
Input pin
Pin: E (MOS without pull-up)



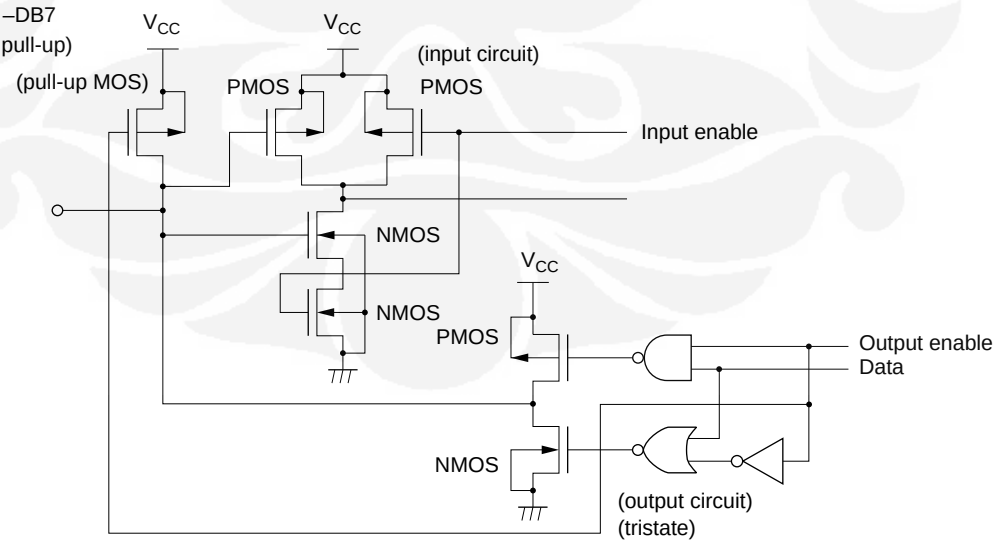
Pins: RS, R/$\overline{W}$ (MOS with pull-up)



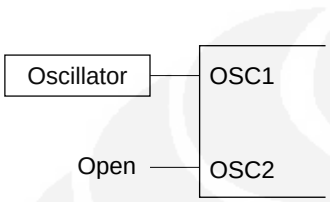
Output pin
Pins: CL1, CL2, M, D

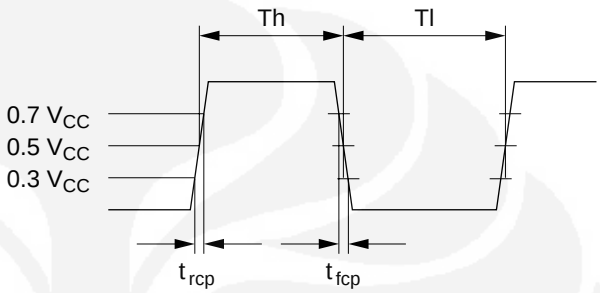


I/O Pin
Pins: DB0 –DB7
(MOS with pull-up)



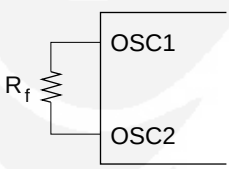
6. Applies to input pins and I/O pins, excluding the OSC1 pin.
7. Applies to I/O pins.
8. Applies to output pins.
9. Current flowing through pull–up MOSs, excluding output drive MOSs.
10. Input/output current is excluded. When input is at an intermediate level with CMOS, the excessive current flows through the input circuit to the power supply. To avoid this from happening, the input level must be fixed high or low.
11. Applies only to external clock operation.



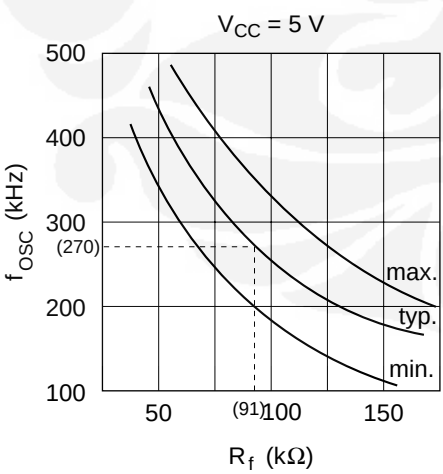


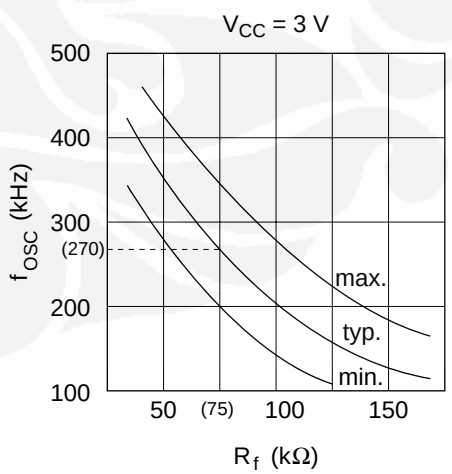
$$\text{Duty} = \frac{Th}{Th + Tl} \times 100\%$$

12. Applies only to the internal oscillator operation using oscillation resistor R_f.



R_f: 75 kΩ ± 2% (when V_{CC} = 3 V)
R_f: 91 kΩ ± 2% (when V_{CC} = 5 V)
Since the oscillation frequency varies depending on the OSC1 and OSC2 pin capacitance, the wiring length to these pins should be minimized.

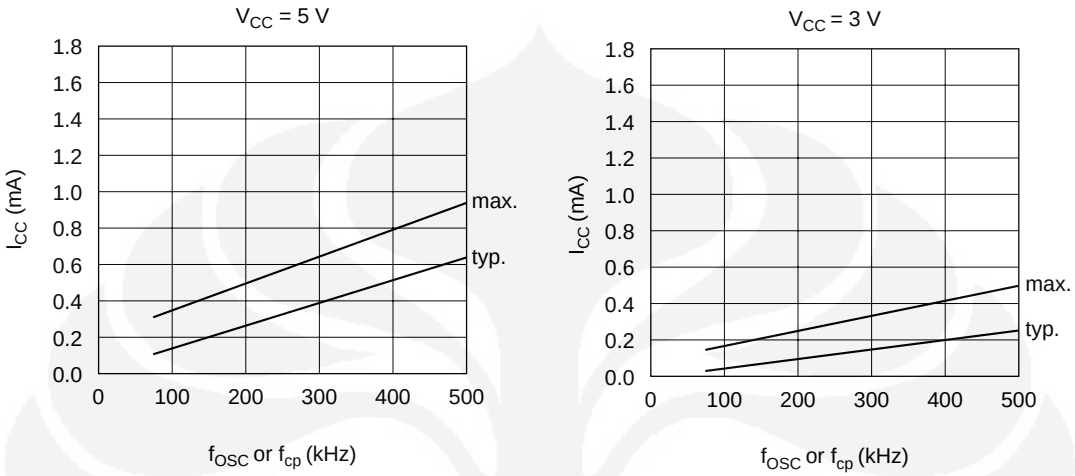




13. RCOM is the resistance between the power supply pins (V_{CC}, V1, V4, V5) and each common signal pin (COM1 to COM16).

    RSEG is the resistance between the power supply pins (V_{CC}, V2, V3, V5) and each segment signal pin (SEG1 to SEG40).

14. The following graphs show the relationship between operation frequency and current consumption.

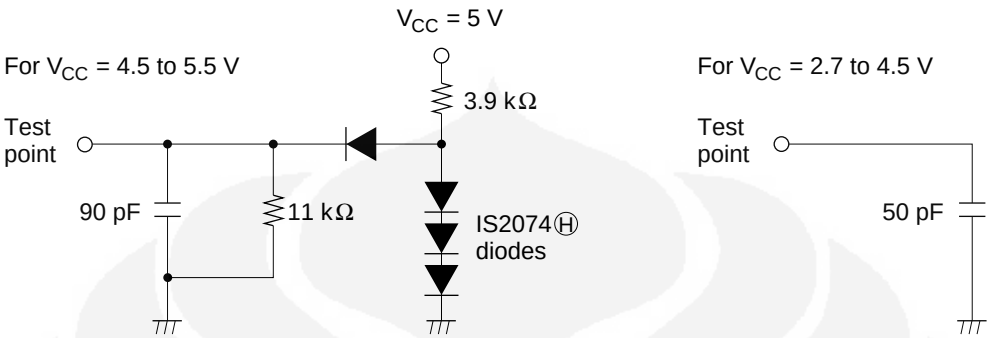


15. Applies to the OSC1 pin.

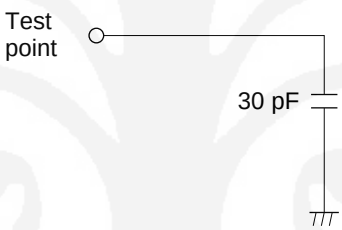
16. Each COM and SEG output voltage is within ±0.15 V of the LCD voltage (V_{CC}, V1, V2, V3, V4, V5) when there is no load.

Load Circuits

Data Bus DB0 to DB7



External Driver Control Signals: CL1, CL2, D, M



Timing Characteristics

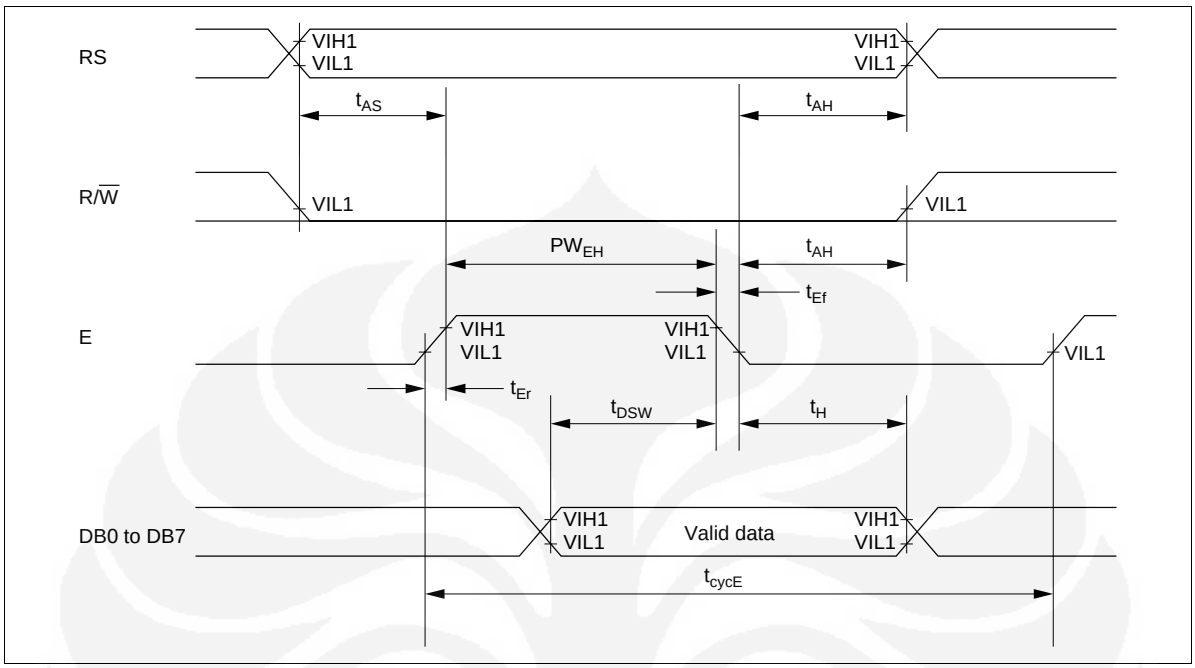
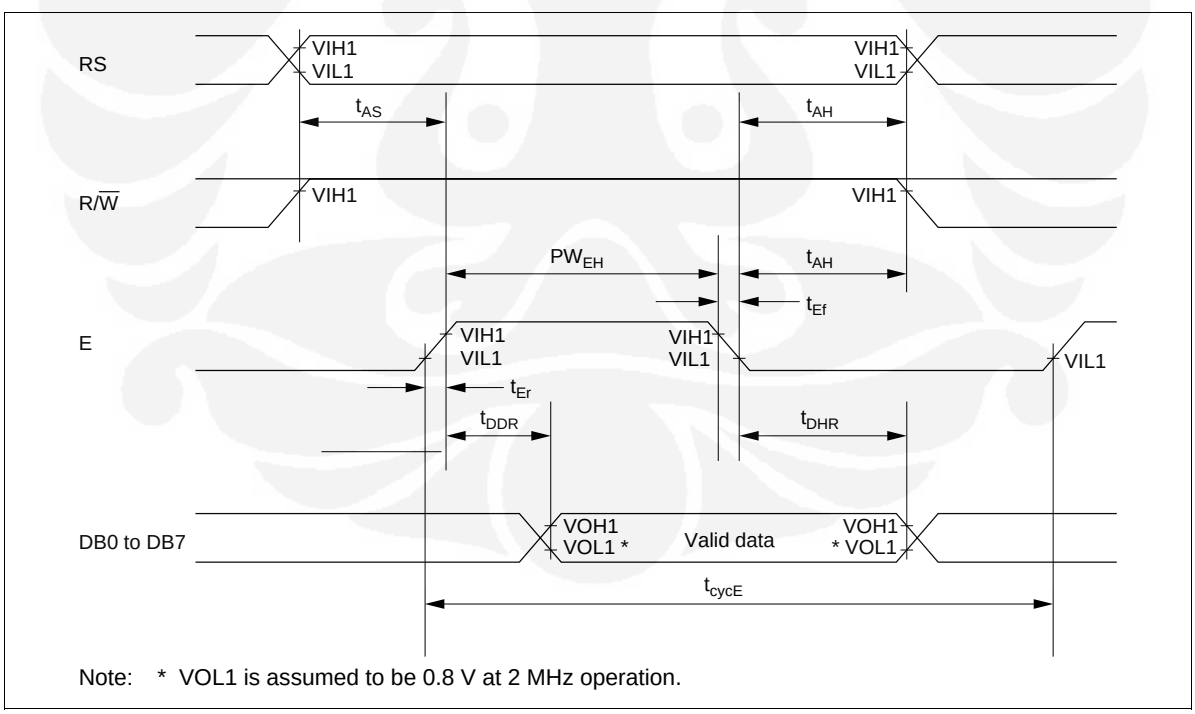


Figure 25 Write Operation



Note: * VOL1 is assumed to be 0.8 V at 2 MHz operation.

Figure 26 Read Operation

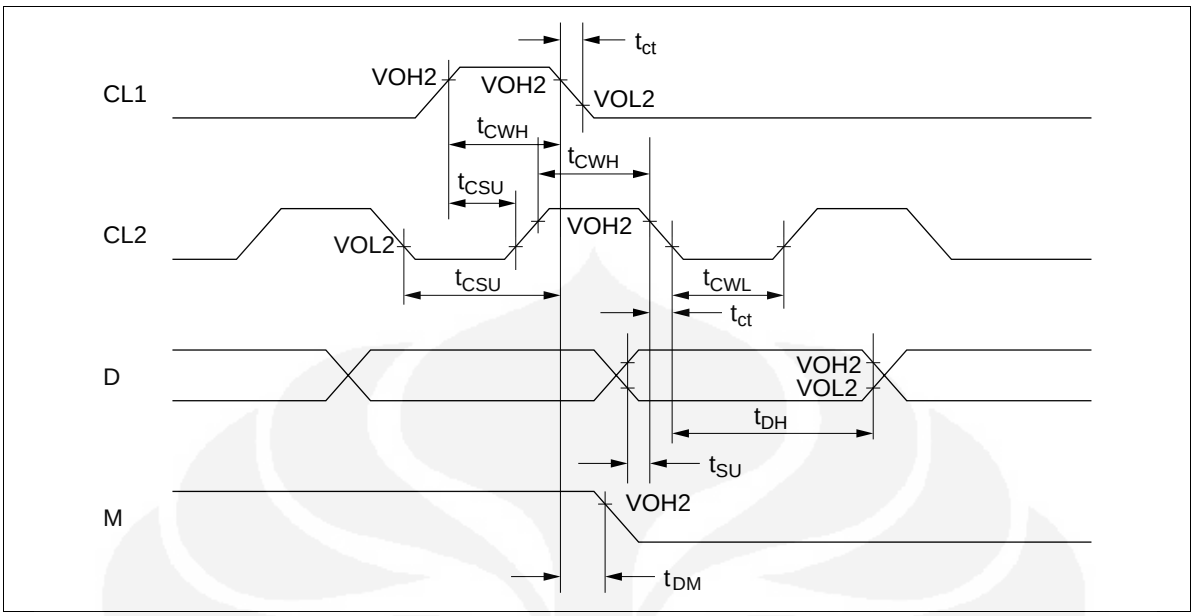
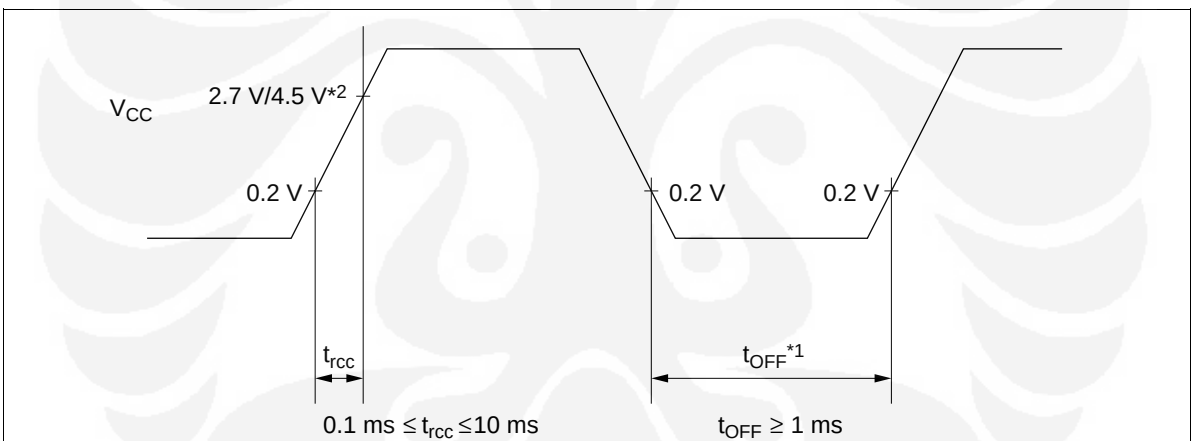


Figure 27 Interface Timing with External Driver



Notes: 1. t_{OFF} compensates for the power oscillation period caused by momentary power supply oscillations.
2. Specified at 4.5 V for 5-V operation, and at 2.7 V for 3-V operation.
3. For if 4.5 V is not reached during 5-V operation, the internal reset circuit will not operate normally.
In this case, the LSI must be initialized by software. (Refer to the Initializing by Instruction section.)

Figure 28 Internal Power Supply Reset

Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

HITACHI

Hitachi, Ltd.
Semiconductor & Integrated Circuits.
Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan
Tel: Tokyo (03) 3270-2111 Fax: (03) 3270-5109

URL
NorthAmerica : http:semiconductor.hitachi.com/
Europe : http://www.hitachi-eu.com/hel/ecg
Asia (Singapore) : http://www.has.hitachi.com.sg/grp3/sicd/index.htm
Asia (Taiwan) : http://www.hitachi.com.tw/E/Product/SICD_Frame.htm
Asia (HongKong) : http://www.hitachi.com.hk/eng/bo/grp3/index.htm
Japan : http://www.hitachi.co.jp/Sicd/indx.htm

For further information write to:

Hitachi Semiconductor (America) Inc.
179 East Tasman Drive,
San Jose,CA 95134
Tel: <1> (408) 433-1990
Fax: <1>(408) 433-0223

Hitachi Europe GmbH
Electronic components Group
Dornacher Straße 3
D-85622 Feldkirchen, Munich
Germany
Tel: <49> (89) 9 9180-0
Fax: <49> (89) 9 29 30 00

Hitachi Europe Ltd.
Electronic Components Group.
Whitebrook Park
Lower Cookham Road
Maidenhead
Berkshire SL6 8YA, United Kingdom
Tel: <44> (1628) 585000
Fax: <44> (1628) 778322

Hitachi Asia Pte. Ltd.
16 Collyer Quay #20-00
Hitachi Tower
Singapore 049318
Tel: 535-2100
Fax: 535-1533

Hitachi Asia Ltd.
Taipei Branch Office
3F, Hung Kuo Building. No.167,
Tun-Hwa North Road, Taipei (105)
Tel: <886> (2) 2718-3666
Fax: <886> (2) 2718-8180

Hitachi Asia (Hong Kong) Ltd.
Group III (Electronic Components)
7/F., North Tower, World Finance Centre,
Harbour City, Canton Road, Tsim Sha Tsui,
Kowloon, Hong Kong
Tel: <852> (2) 735 9218
Fax: <852> (2) 730 0281
Telex: 40815 HITEC HX

2981 and 2982

8-Channel Source Drivers

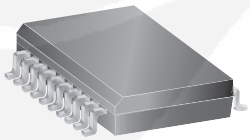
Features and Benefits

- TTL, DTL, PMOS, or CMOS compatible inputs
- 500 mA output source current capability
- Transient-protected outputs
- Output breakdown voltage to 50 V
- DIP or SOIC packaging

Packages:

Not to scale

18-pin DIP (Package A)



20-pin SOICW (package LW)
(drop-in replacement for discontinued 18-pin SOIC variants)

Description

Recommended for high-side switching applications that benefit from separate logic and load grounds, these devices encompass load supply voltages to 50 V and output currents to -500 mA. These 8-channel source drivers are useful for interfacing between low-level logic and high-current loads. Typical loads include relays, solenoids, lamps, stepper and/or servo motors, print hammers, and LEDs.

All devices may be used with 5 V logic systems — TTL, Schottky TTL, DTL, and 5 V CMOS. The device packages offered are electrically interchangeable, and will withstand a maximum output off voltage of 50 V, and operate to a minimum of 5 V. All devices in this series integrate input current limiting resistors and output transient suppression diodes, and are activated by an active high input.

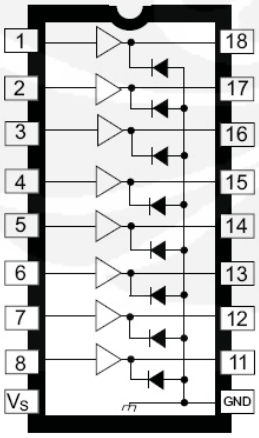
The suffix "A" indicates an 18-lead plastic dual in-line package with copper lead frame for optimum power dissipation. Under normal operating conditions, these devices will sustain 120 mA continuously for each of the eight outputs at an ambient temperature of +50°C and a supply of 15 V.

The suffix "LW" package is provided in a 20-pin wide-body SOIC package with improved thermal characteristics compared to the 18-pin SOIC version it replaces (100% pin-compatible electrically). The A2982ELW driver is available for operation over an extended temperature range, down to -40°C.

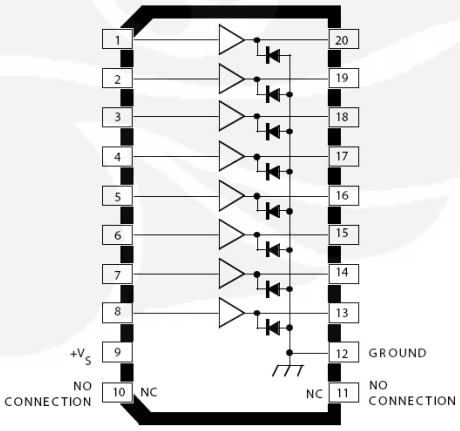
These packages are lead (Pb) free, with 100% matte-tin leadframe plating.

Simplified Block Diagrams

18-pin DIP (A Package)



20-pin SOICW (LW Package)



(NC pins, 10 and 11, not present on discontinued 18-pin LW package)

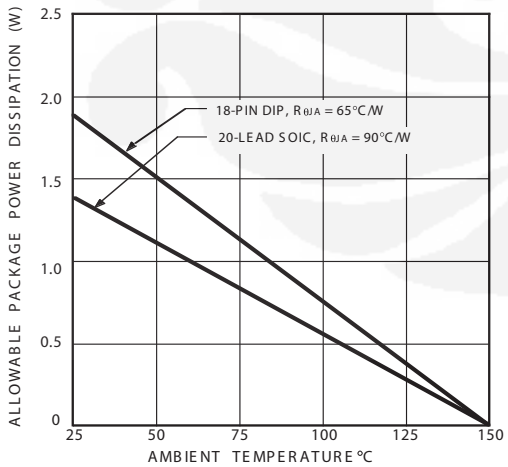
Selection Guide

Part Number	Package	Packing	Ambient Temperature T_A (°C)
A2982ELWTR-T*	20-pin SOICW	1000 per reel	–40 to 85
A2982SLWTR-T	20-pin SOICW	1000 per reel	–20 to 85
UDN2981A-T	18-pin DIP	21 per tube	
UDN2982A-T	18-pin DIP	21 per tube	

*Variant is in production but has been determined to be LAST TIME BUY. This classification indicates that the variant is obsolete and notice has been given. Sale of the variant is currently restricted to existing customer applications. The variant should not be purchased for new design applications because of obsolescence in the near future. Samples are no longer available. Status date change November 2, 2009. Deadline for receipt of LAST TIME BUY orders is April 30, 2010.

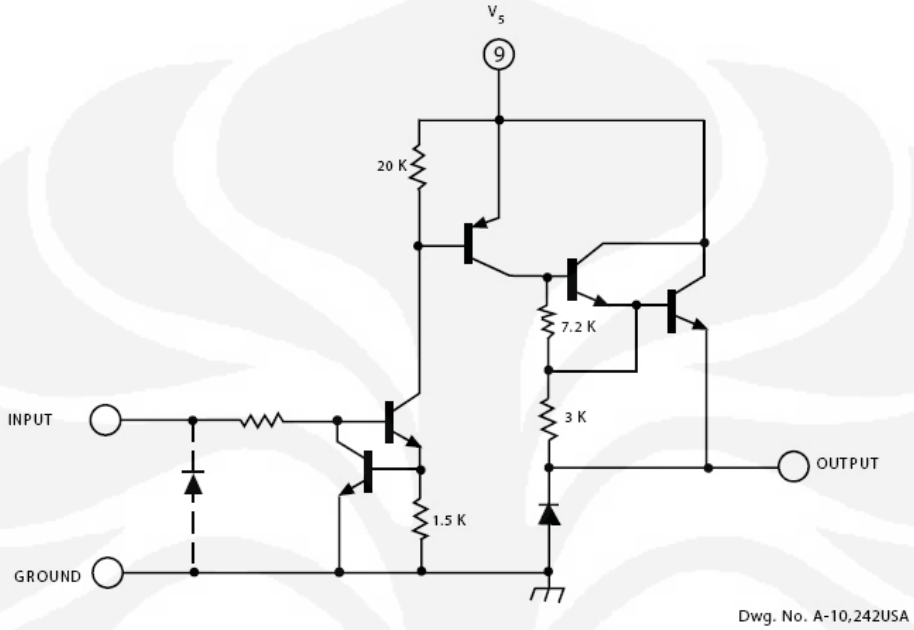
Absolute Maximum Ratings

Characteristic	Symbol	Notes	Rating	Units
Output Voltage Range	V_{CE}		5 to 50	V
Input Voltage	V_{IN}	UDN2981	20	V
		A2982, UDN2982	20	V
Output Current	I_{OUT}		–500	mA
Package Power Dissipation	P_D	See graph	–	–
Operating Ambient Temperature	T_A	Range E	–40 to 85	ºC
		Range S	–20 to 85	ºC
Maximum Junction Temperature	T_J(max)		150	ºC
Storage Temperature	T_{stg}		–55 to 150	ºC



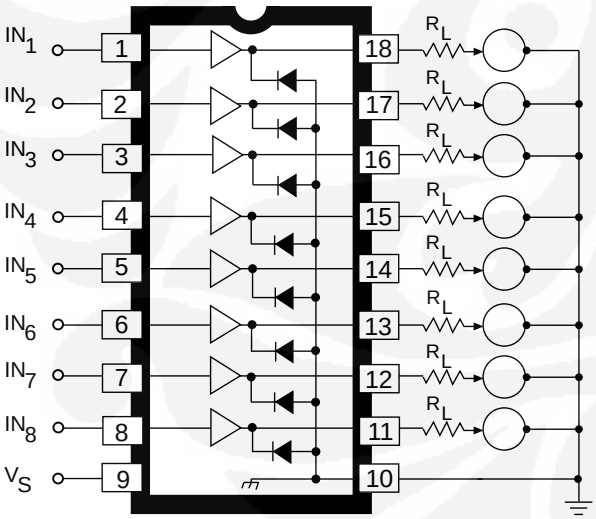
Allegro MicroSystems, Inc.
115 Northeast Cutoff
Worcester, Massachusetts 01615-0036 U.S.A.
1.508.853.5000; www.allegromicro.com

One of Eight Drivers

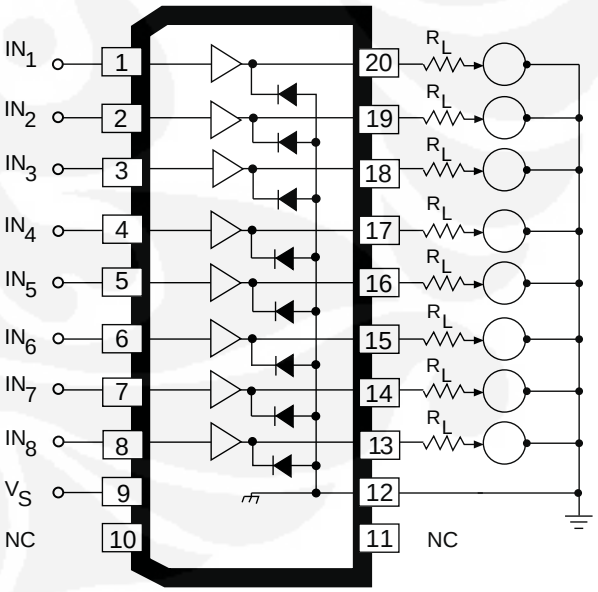


Typical electrosensitive printer application

18-pin DIP (A Package)



20-pin SOICW (LW Package)



Pins 10 and 11 can float; other pins match discontinued 18-pin SOIC: 1 to 9 same, pins 12 to 20 match pins 10 to 18

ELECTRICAL CHARACTERISTICS[1,2] at T_A = +25°C (unless otherwise specified).

Characteristic	Symbol	Variant	Test Conditions	Test Fig.	Min.	Typ.	Max.	Units
Output Leakage Current[3]	I_{CEX}	All	V_{IN} = 0.4 V, V_S = 50 V	1	—	—	20	µA
Output Sustaining Voltage	$V_{CE(SUS)}$	All	I_{OUT} = -45 mA	—	35	—	—	V
Collector-Emitter Saturation Voltage	$V_{CE(SAT)}$	All	V_{IN} = 2.4 V, I_{OUT} = -100 mA	2	—	1.6	1.8	V
			V_{IN} = 2.4 V, I_{OUT} = -225 mA	2	—	1.7	1.9	V
			V_{IN} = 2.4 V, I_{OUT} = -350 mA	2	—	1.8	2.0	V
Input Current	$I_{IN(ON)}$	2981	V_{IN} = 2.4 V	3	—	140	200	µA
			V_{IN} = 3.85 V	3	—	310	450	µA
		2982	V_{IN} = 2.4 V	3	—	140	200	µA
			V_{IN} = 12 V	3	—	1.25	1.93	mA
Output Source Current (Outputs Open)	I_{OUT}	2981	V_{IN} = 2.4 V, V_{CE} = 2.0 V	2	-350	—	—	mA
		2982	V_{IN} = 2.4 V, V_{CE} = 2.0 V	2	-350	—	—	mA
Supply Current Leakage Current	I_S	All	V_{IN} = 2.4 V*, V_S = 50 V	4	—	—	10	mA
Clamp Diode Current	I_R	All	V_R = 50 V, V_{IN} = 0.4 V*	5	—	—	50	µA
Clamp Diode Forward Voltage	V_F	All	I_F = 350 mA	6	—	1.5	2.0	V
Turn-On Delay	t_{ON}	All	0.5 E_{IN} to 0.5 E_{OUT}, R_L = 100Ω, V_S = 35 V	—	—	0.3	2.0	µs
Turn-Off Delay[4]	t_{OFF}	All	0.5 E_{IN} to 0.5 E_{OUT}, R_L = 100Ω, V_S = 35 V, See Note	—	—	2.0	10	µs

[1]Negative current is defined as coming out of (sourcing) the specified device terminal.

[2]All unused inputs must be connected to ground. Pull-down resistors (approximately 10 kΩ) are recommended for inputs that are allowed to float while power is being applied to V_S.

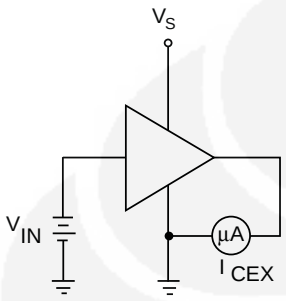
[3]All inputs simultaneously.

[4]Turn-off delay is influenced by load conditions. Systems applications well below the specified output loading may require timing considerations for some designs, i.e., multiplexed displays or when used in combination with sink drivers in a totem pole configuration.

Allegro MicroSystems, Inc.
115 Northeast Cutoff
Worcester, Massachusetts 01615-0036 U.S.A.
1.508.853.5000; www.allegromicro.com

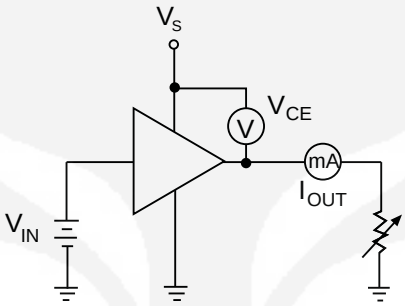
TEST FIGURES

Figure 1



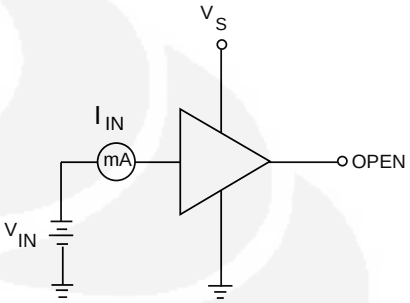
Dwg. No. A-11,083

Figure 2



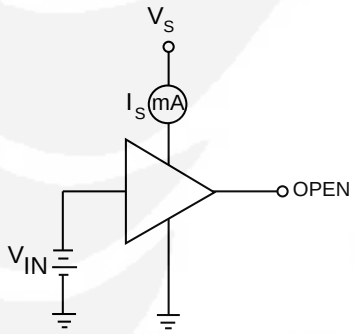
Dwg. No. A-11,084

Figure 3



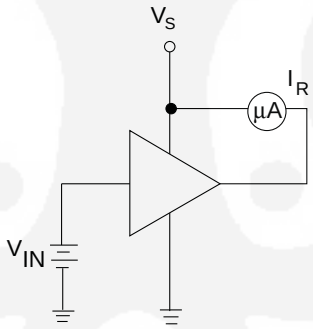
Dwg. No. A-11,085

Figure 4



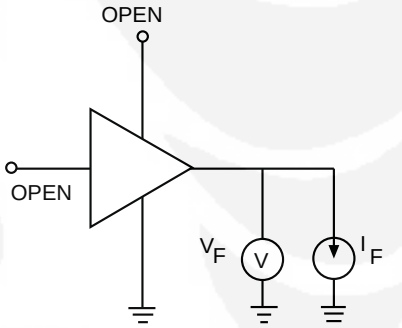
Dwg. No. A-11,086

Figure 5



Dwg. No. A-11,087

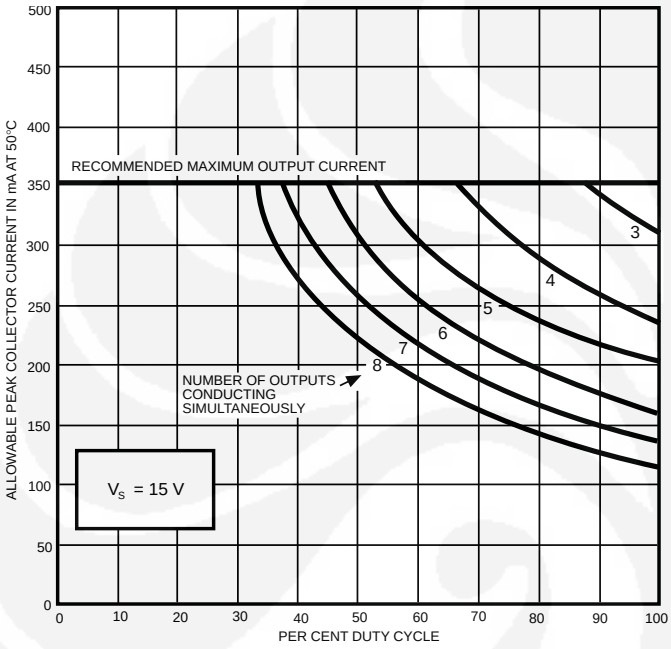
Figure 6



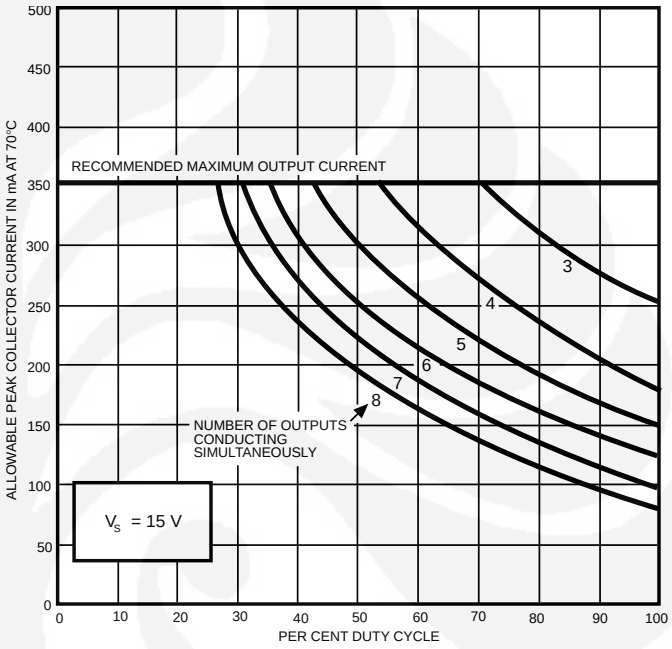
Dwg. No. A-11,088

Allowable peak collector current as a function of duty cycle

UDN2981A and UDN2982A

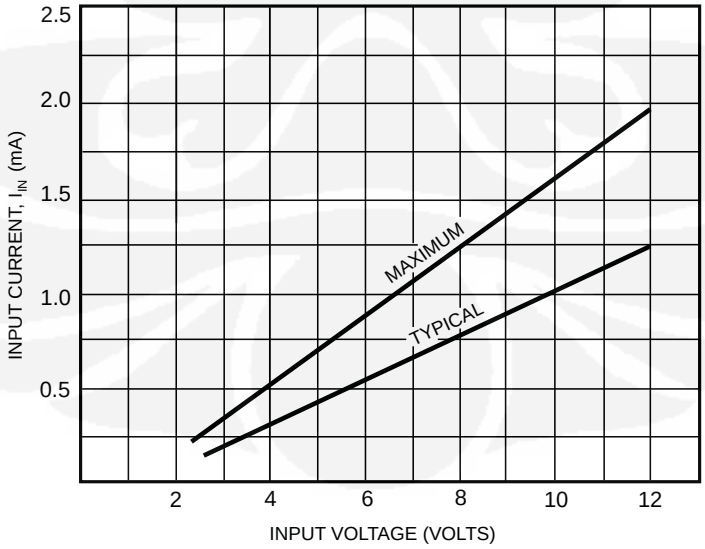


Dwg. No. A-11,107B



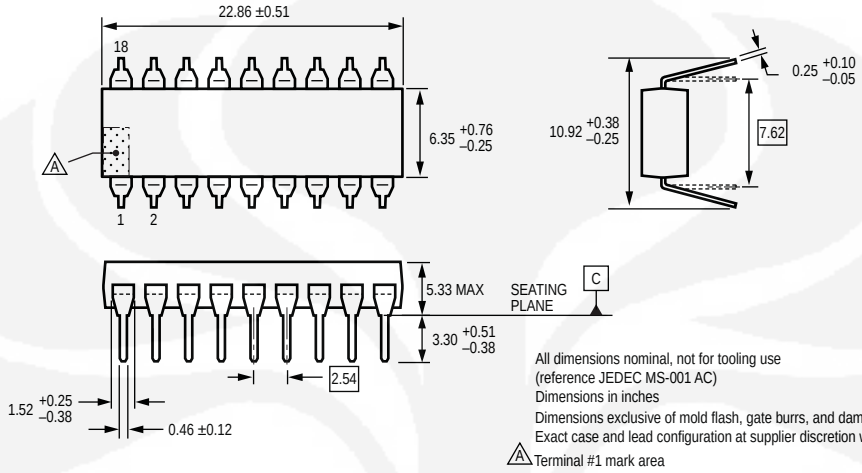
Dwg. No. A-11,108B

Input current as a function of input voltage



Dwg. No. A-11,115B

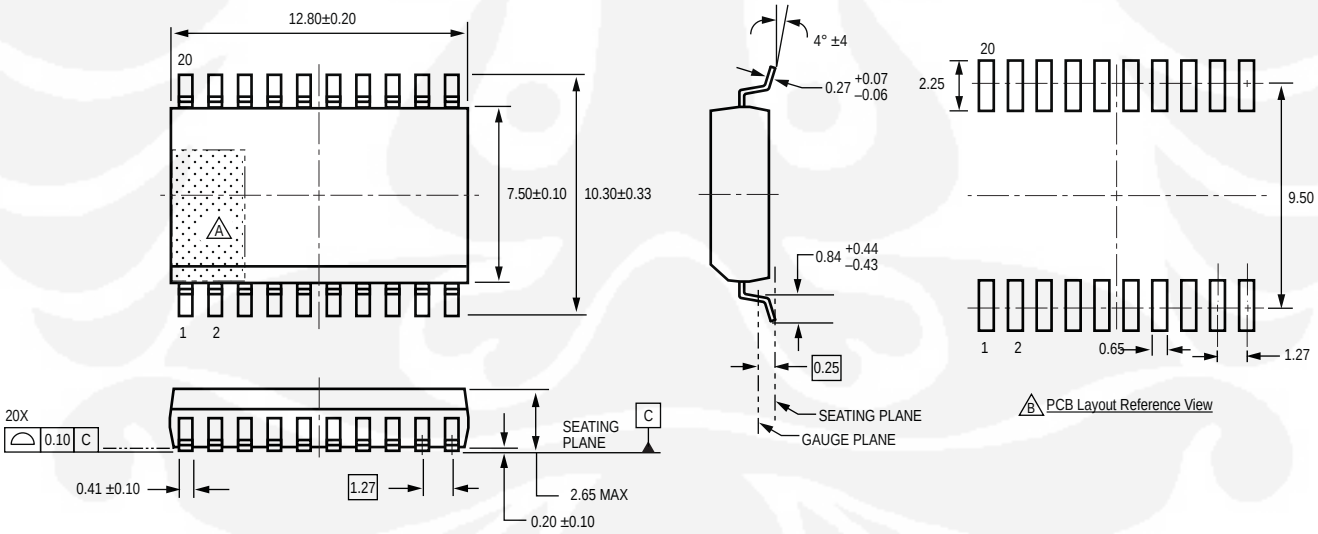
A Package, 18-Pin DIP



All dimensions nominal, not for tooling use
(reference JEDEC MS-001 AC)
Dimensions in inches
Dimensions exclusive of mold flash, gate burrs, and dambar protrusions
Exact case and lead configuration at supplier discretion within limits shown

A Terminal #1 mark area

LW Package, 20-Pin SOICW



For Reference Only
Dimensions in millimeters
(Reference JEDEC MS-013 AC)
Dimensions exclusive of mold flash, gate burrs, and dambar protrusions
Exact case and lead configuration at supplier discretion within limits shown

A Terminal #1 mark area

B Reference pad layout (reference IPC SOIC127P1030X265-20M)
All pads a minimum of 0.20 mm from all adjacent pads; adjust as necessary
to meet application process requirements and PCB layout tolerances

For the latest version of this document, visit our website:
www.allegromicro.com

Allegro MicroSystems, Inc.
115 Northeast Cutoff
Worcester, Massachusetts 01615-0036 U.S.A.
1.508.853.5000; www.allegromicro.com

サンケン電気株式会社
技術本部 PM 事業部
技術部技術 1 グループ殿

納　入　仕　様　書

品名 SI-8008HFE

LF No.1113

RoHS 指令対応
RoHS Directive Compliance

変更の事前連絡
本製品の仕様,材料,製造工程,及び管理システム等の変更を行う場合は事前に PM 事業部技術部技術 1 グループ殿に連絡することとする。

受領印欄：この書類を受領致しました。

RECEIVED		✓ New Item	Date: 2008. 6. 23
Check	Check	Check	Approval
宮原	猪狩	田澤	
SKI Goods in		✓ JSK ENGINEERING	
JSK Goods in		Return	
Important Notice for Goods in			Nothing

SANKEN ELECTRIC CO., LTD.
POWER MODULE DIVISION
ENGINEERING DEPARTMENT

34219079

承認	審査	作成
H. Nakamichi	Daiji Uehara	川邊良治
Hideki Nakamichi	Daiji Uehara	Ryouji Kawabe
サンケン電気株式会社 技術本部 PCD 事業部		
発行年月日	2008／06／13	
仕様書番号	SSJ-03557	

1. 適用範囲
Scope

この規格は、5端子フルモールド型降圧スイッチングレギュレータ IC SI-8008HFEについて適用する。
The present specifications shall apply to a DC-DC buck converter SI-8008HFE.

2. 概要
Outline

種別 Classificatio	半導体集積回路(モノリシックＩＣ) Semiconductor IC (monolithic IC)
構造 Structure	樹脂封止型(トランスファーモールド) Plastic package (transfer mold)
主用途 Applications	・直流安定化電源装置 For DC voltage regulator ・テレコムオンボードローカル電源 For power supplies for telecommunication equipment ・OA 機器 For power supplies for office equipment etc. ・オンボードローカル電源 For ON-board local regulator ・スイッチングレギュレータ２次側出力電圧安定化 For output voltage regulator at secondary stage of switch mode power supply

3. 絶対最大定格
Absolute maximum ratings

3-1 絶対最大定格
Absolute maximum ratings

項目 Characteristic	記号 Symbol	規格 Ratings	単位 Units	条件 Remarks
入力電圧 DC input voltage	V_{IN}	43	V	
無限大放熱時許容損失 Power dissipation with infinite heat-sink	Pd1-1	25	W	但し過熱保護により制限 Tjmax=150℃, But limited by thermal shut-down.
	Pd1-2	20	W	Tj=125 °C
放熱板未使用時許容損失 Power dissipation without heat-sink	Pd2-1	2.15	W	但し過熱保護により制限 Tjmax=150℃, But limited by thermal shut-down.
	Pd2-2	1.72	W	Tj=125 °C
接合温度 Junction temperature	Tj	+150max	°C	この製品は過熱保護回路を内蔵しており、接合部温度が 130℃以上になると、動作することがあります。 Thermal protection circuit is built-in in this product and when junction temperature rises to 130 °C or higher, it may be caused to operate 動作時のジャンクション温度としては125℃以下での設計を推奨いたします。 Recommended max. junction temperature at operation is 125°C.
保存温度 Storage temperature	Tstg	-40～150	°C	
熱抵抗(接合－ケース間) Thermal resistance (junction-case)	θ j-c	5	°C /W	
熱抵抗(接合－周囲間) Thermal resistance (junction-ambient air)	θ j-a	58	°C /W	

3-2 推奨動作条件
Recommended operating conditions

項目 Characteristic	記号 Symbol	規格 Ratings MIN	 MAX	単位 Units	条件 Remarks
入力電圧範囲 DC input voltage range	V_{IN}	*1 Vo+3	40	V	
出力電圧範囲 DC output voltage range	Vo	0.8～24		V	
出力電流範囲 DC output current range	Io	0～5.5		A	*2 V_{IN}≧Vo+3V
動作時接合温度範囲 Operating junction temperature range	T_{jop}	-30～125		°C	
動作温度範囲 Operating temperature range	T_{op}	-30～85		°C	*3

* 1 入力電圧範囲の最小値は、4.5V もしくは Vo+3 V のどちらか大きい値とします。
V_{IN}= Vo+2～Vo+3V の場合は Io=3A MAX となります。
*1 The minimum value of input voltage is taken as the larger one of either 4.5v or "Vout+3v."In the case of VIN=Vo+2～Vo+3V , it is set to Io=3A MAX.

*2 但し、熱減定格(P5 参照)以内で使用する必要があります。
*2 To be used within the allowable package power dissipation characteristics (refer to P5).

4．電気的特性
Electrical characteristics

4-1 電気的特性 (Ta=25℃、Vo=5V 設定時 R1=4.2kΩ,R2=0.8kΩ)
Electrical characteristics (Ta=25℃、Vo=5V adjusted R1=4.2kΩ,R2=0.8kΩ)

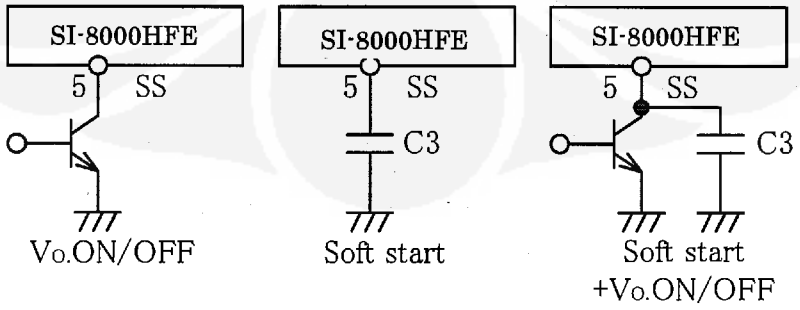
項目 Characteristic		記号 Symbol	規格値 Limits MIN	TYP	MAX	単位 Units	測定条件 Test conditions
設定基準電圧 Reference voltage		V_{ADJ}	0.784	0.800	0.816	V	V_{IN}=15V, I_O=1A
基準電圧温度係数 Reference voltage temperature coefficient		$\Delta V_{ADJ}/\Delta T$		±0.1		mV/℃	V_{IN}=15V, I_O=1A,Tc=0~100℃
効率 *3 Efficiency *3		η		83		%	V_{IN}=15V,I_O=3A
動作周波数 Operating frequency		fo		150		kHz	V_{IN}=15V, I_O=3A
ラインレギュレーション Line regulation		VLine		60	80	mV	V_{IN}=10～30V, I_O=3A
ロードレギュレーション Load regulation		VLoad		20	50	mV	V_{IN}=15V, I_O=0.2～5.5A
過電流保護開始電流 Over current protection starting current		Is	5.6	6.5	7.5	A	V_{IN}=15V
オンオフ端子 *4 ON/OFF terminal *4	Low レベル電圧 Low level voltage	V_{SSL}			0.5	V	
	Low 時流出電流 Flow-out current at low level voltage	I_{SSL}		10	30	μA	V_{SS}=0V
静止時回路電流 1 Quiescent current 1		Iq		6		mA	V_{IN}=15V, I_O=0A
静止時回路電流 2 Quiescent current 2		Iq(off)		200	400	μA	V_{IN}=15V、V_{SS}=0V

*3 効率は次式により算出されます。
*3 Efficiency is calculated by the following equation.

$$\eta\,(\%)=\frac{V_O \cdot I_O}{V_{IN} \cdot I_{IN}} \times 100$$

*4 5番端子は、SS 端子で、コンデンサーを接続することによりソフトスタートさせることが出来ます。また、SS 端子を用い、出力を ON/OFF することが可能です。
SS 端子電圧を V_{SSL} 以下にすることで出力は停止します。SS 端子の電位切り替えは、トランジスタのオープンコレクタ駆動等で行うことが出来ます。尚、ソフトスタートと、ON/OFF を併用した場合、ON/OFF 用トランジスタにはＣ３のディスチャージ電流が流れるため、Ｃ３の容量が大きい場合は、電流制限等の保護を行って下さい。また、SS 端子は IC 内部電源にプルアップ(3.7Vtyp)されていますので、外部からの電圧印加は出来ません。未使用の場合は、オープンとして下さい。

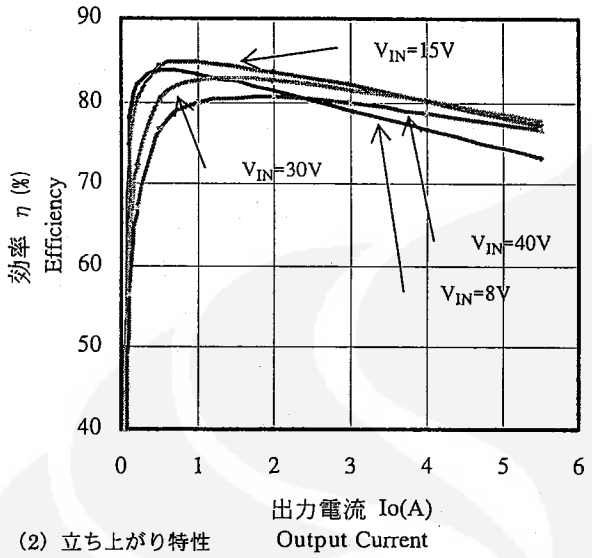
*4 No.5 terminal is a SS terminal to enable soft start by connecting a capacitor. The output can be turned on and off by using a SS terminal. The output is stopped by decreasing the SS terminal voltage below V_{SSL} and in order to perform ON/OFF operation of Vo, it is required to connect NPN transistor or the output of open collector type TTL between No.5 terminal and GND.
In case that both soft and Vout ON/OFF are used, a protection measure such as limitation of current is required, as the discharge current of C3 flows across a transistor for ON/OFF operation, if the capacitance of C3 large. As a pull-up type resistor is provided inside the IC, no external voltage can be applied. In case of no use of ON/OFF, please keep it open.



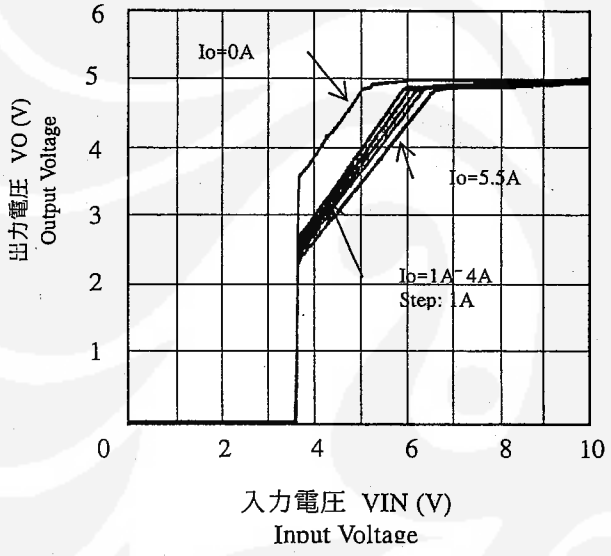
4-2 代表特性例 (Ta=25℃) 出力電圧 5V 設定時:

Typical characteristics (Ta=25℃) Vo=5V adjusted R1=4.2kΩ, R2=0.8kΩ

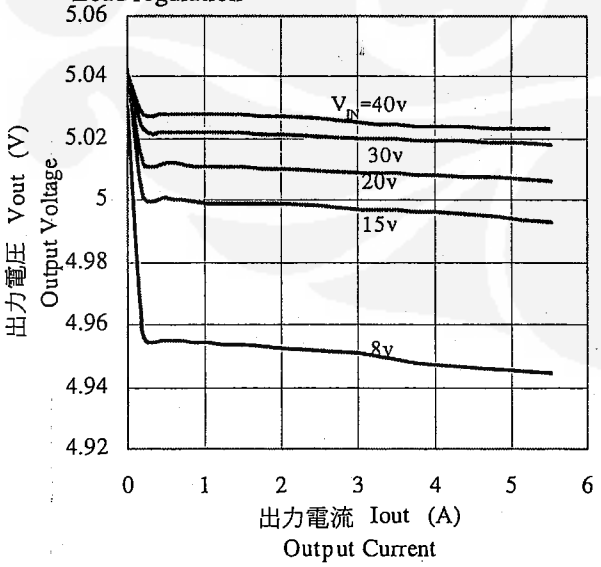
(1) 効率
Efficiency



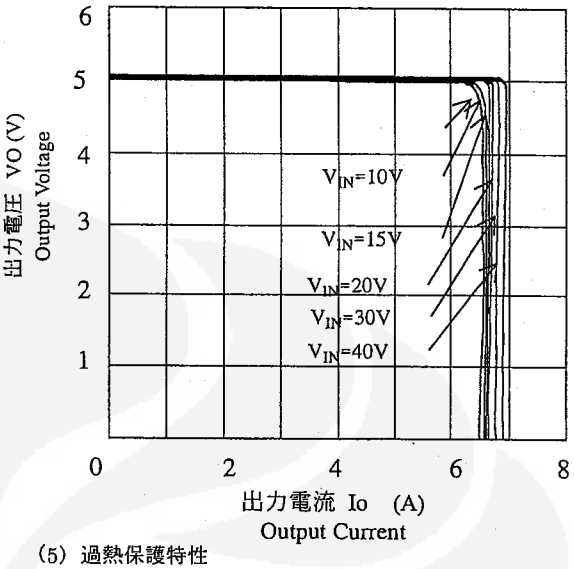
(2) 立ち上がり特性
Low voltage behavior



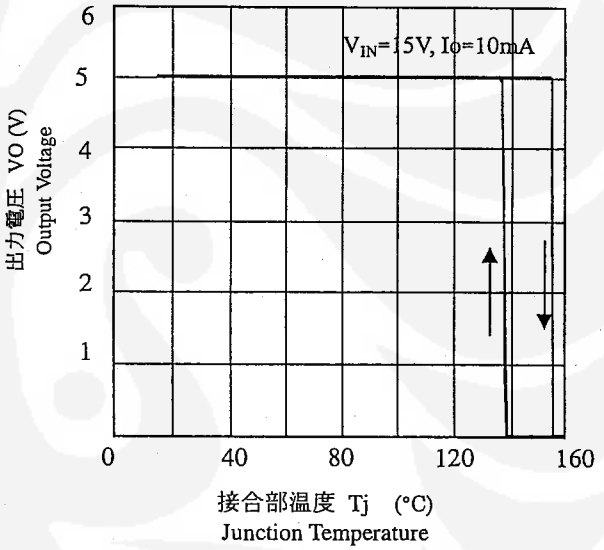
(3) ロードレギュレーション
Load regulation



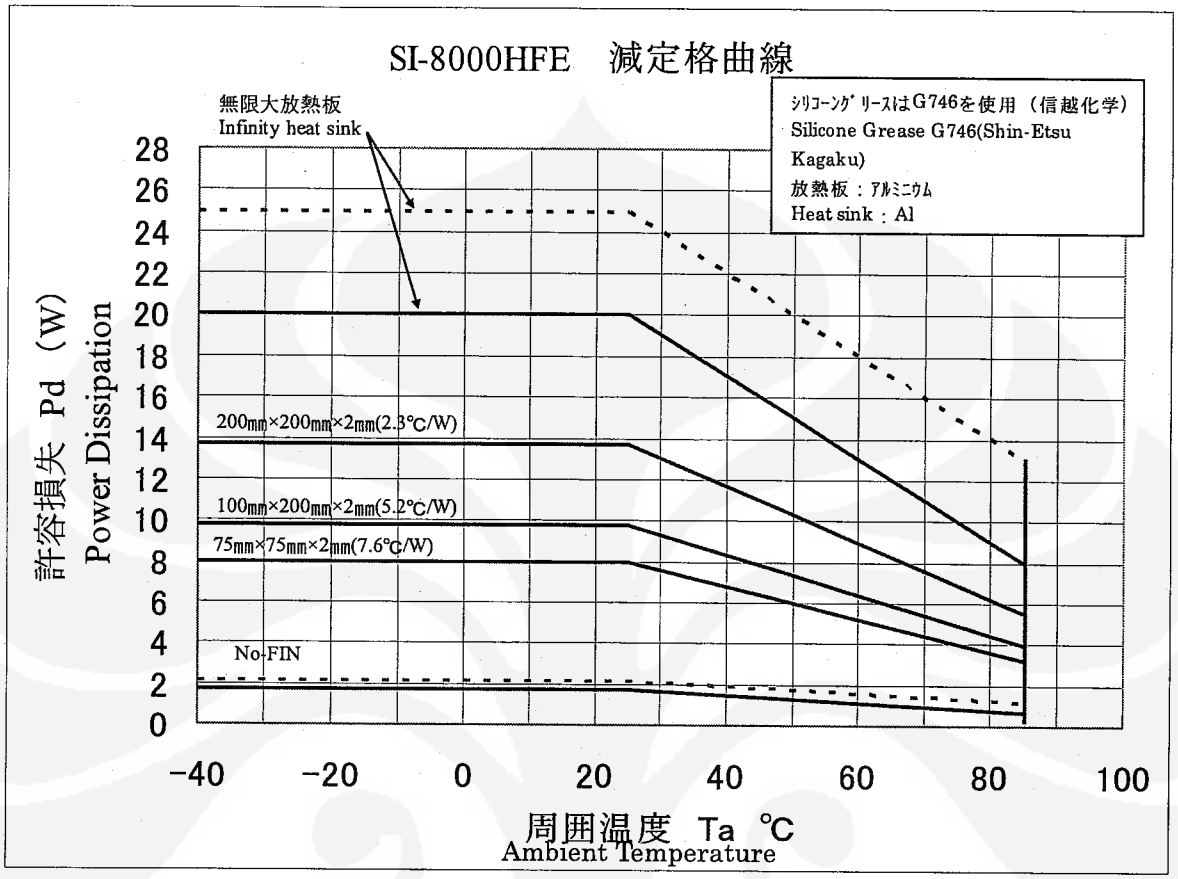
(4) 過電流保護特性
Over Current Protection



(5) 過熱保護特性
Thermal Protection



4-3熱減定格
Allowable package power dissipation



$$P_D = V_O \cdot I_O\left(\frac{100}{\eta_x} - 1\right) - V_F \cdot I_O\left(1 - \frac{V_O}{V_{IN}}\right)$$

V_O：出力電圧 Output voltage　V_{IN}：入力電圧 Input voltage
I_O：出力電流 Output current　η_x：効率（%）Efficiency（%）
V_F：Di 順方向電圧 Diode forward voltage　（FMB－G16L･･･0.55V at I_O=5.5A）

——— Tjmax=125℃
- - - - Tjmax=150℃

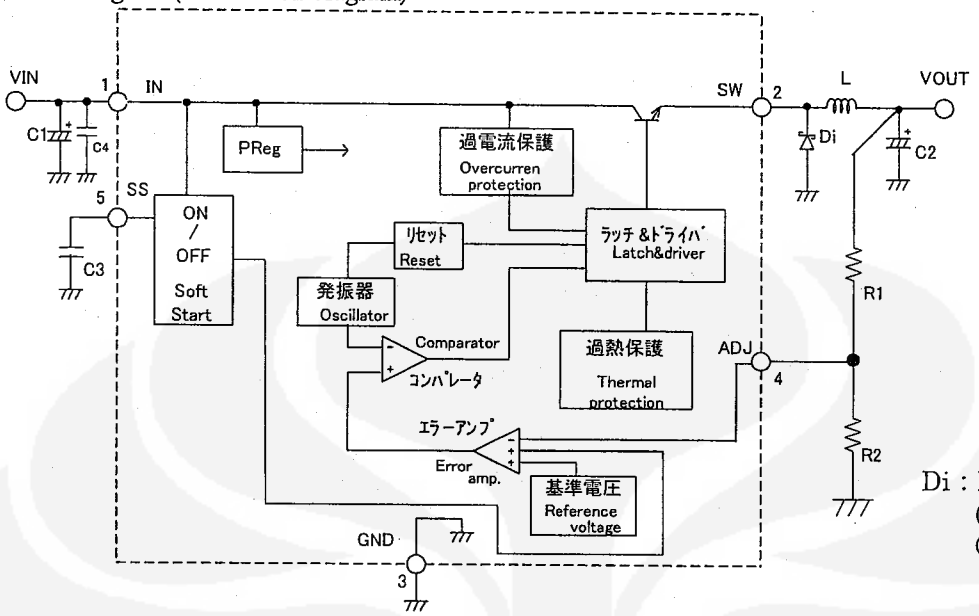
この製品は過熱保護回路を内蔵しており、接合部温度が130℃以上になると、動作することがあります。
Thermal protection circuit is built-in this product and when junction temperature rises to 130℃ or higher, it may be caused to operate.

注１：効率は、入力電圧、出力電流によって変化する為、４頁の効率曲線より求め、パーセント表示のまま代入する。
Note1：As the efficiency varies subject to the input voltage and output current, it shall be obtained from the efficiency curve in page 4 and substituted in percent.

注２：ダイオード熱設計は別途行う必要があります。
Note2：Thermal design for Diode shall be made separately.

5 ブロックダイアグラム(ピン配置)

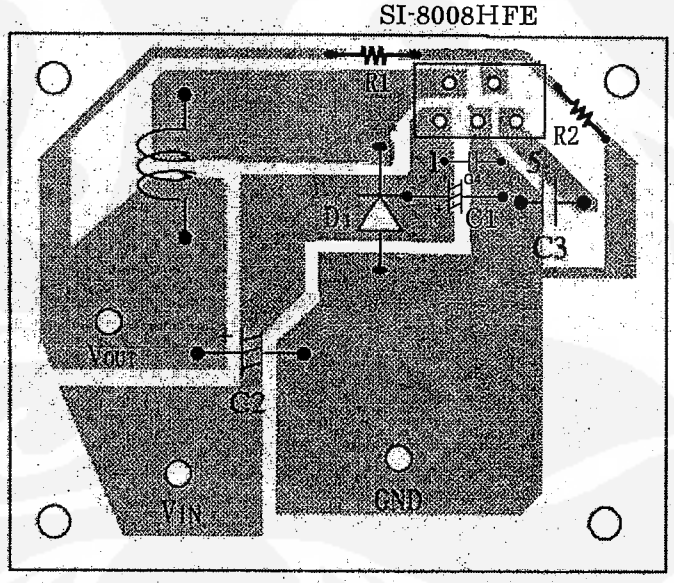
Block diagram (Connection diagram)



Di：FMB－G16L
(サンケン製)
(made by sanken)

推奨パターン

Recommended pattern



(top view ・シルク印刷面)
(top view・silk printing side)

*最適な動作条件とするためには、GND ラインは 3 番端子を中心にした 1 点 GND 配線とし、各部品を最短で配置することが必要です。

*The circuit board layout is recommended as follows:

①Other components are connected as close as possible to the SI-8008HFE.

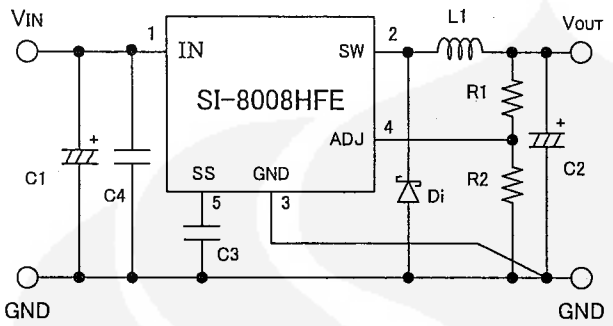
②Each ground of all components is connected at one point area.

6 応用回路例

Example application circuit

6-1 標準回路図

Standard circuit diagram



C1：1500μF
C2：1000μF
C3：1μF
(ソフトスタート機能使用時のみ)
(If soft start function is used)
C4：4.7μF (RPER11H475K5 (村田製作所製))
L1：100μH
Di：FMB－G16L
(サンケン製)
(made by sanken)

ダイオード Ｄｉ

・Ｄｉには、必ずショットキーバリアダイオードを使用して下さい。
ファーストリカバリダイオードを使用した場合、リカバリおよびオン電圧による逆電圧印加によりＩＣを破壊する恐れがあります。

Diode Ｄｉ

The shottky-barrier diode must be used for Di. If other diodes like fast recovery diodes are used, IC may be destroyed because of the reverse voltage applied by the recovery voltage or ON voltage

チョークコイル Ｌ1

・チョークコイルの巻き線抵抗が大きい場合、効率が低下し規格の値に達しない場合があります。
・ 過電流保護開始電流が 6.5A 程度のため、過負荷・負荷短絡時の磁気飽和によるチョークコイルの発熱に注意願います。

Choke coil Ｌ1

・If the winding resistance of the choke coil is too high, the efficiency may go down to the extent that it is out of the rating.
・As the over current protection start current is approx. 6.5A, attention must be paid to the heating of the choke coil by the magnetic saturation due to overload or short-circulated load.

コンデンサー Ｃ1，Ｃ2，Ｃ3，Ｃ４

・Ｃ1，Ｃ2 には大きなリップル電流が流れますので、スイッチング電源用高周波低インピーダンス品をご使用下さい。特にＣ2 のインピーダンスが高い場合、低温時にスイッチング波形に異常を起こすことがあります。又、Ｃ2 に OS コン、タンタルコンデンサー等直流等価抵抗（ＥＳＲ）が極端に小さいコンデンサーを使用した場合、異常発振となる可能性があるため使用しないで下さい。
・ Ｃ3 はソフトスタート用コンデンサーです。ソフトスタート機能を使用しない場合は５番端子をオープンとして下さい。ＩＣ内部でプルアップ済みです。
・ 安定して動作させる為には、C1 と C4 を製品の近傍にﾚｲｱｳﾄする事が極めて重要になります。
(P.6 推奨パターンを参照ください。)

Capacitor Ｃ1，Ｃ2，Ｃ3，Ｃ4

・As large ripple currents flow across C1 and C2, capacitors with high frequency and low impedance for SMPS must be used. Especially when the impedance of C2 is high, the switching waveform may not be normal at low temperature. Please use neither OS capacitor nor tantalum capacitor which causes an abnormal oscillation for the C2.
・C3 is a capacitor for soft start. In case soft start function is not used, please keep No.5 terminal open. A pull-up resistor is provided inside the IC.
・It is stabilized, and in order to make it operate, it becomes very important to arrange C1 and C4 near the product.
(Please refer to a 6-page recommendation pattern.)

抵抗　R1，R2

・R1，R2 は出力電圧を設定する為の抵抗です。I_{ADJ}が 1mA 程度となるよう設定して下さい。又、R1，R2 の値を求める式は以下のようになります。
・Vo=0.8V に設定する際も、安定動作の為 R2 は接続ください。
・出力電圧は入力電圧に対して８％以上になる様に設定する事を推奨します。

$$R1=\frac{(V_{OUT}-V_{ADJ})}{I_{ADJ}}=\frac{(V_{OUT}-0.8)}{1\times10^{-3}}(\Omega),\quad R2=\frac{V_{ADJ}}{I_{ADJ}}=\frac{0.8}{1\times10^{-3}}\fallingdotseq0.8k(\Omega)$$

◎最適な動作環境とするためには、各部品を最短で配置することが必要です。

Resister R1，R2

・R1，R2 is resistor to the Output Voltage. I_{ADJ} set to become 1mA. Moreover, R1，R2 is calculated by the following expression.
・Even in case you set output voltage as 0.8V, please connect R2, in order to make it stabilize and operate.
・It recommends setting up output voltage so that it may become 8% or more to input voltage.

$$R1=\frac{(V_{OUT}-V_{ADJ})}{I_{ADJ}}=\frac{(V_{OUT}-0.8)}{1\times10^{-3}}(\Omega),\quad R2=\frac{V_{ADJ}}{I_{ADJ}}=\frac{0.8}{1\times10^{-3}}\fallingdotseq0.8k(\Omega)$$

◎In order to have optimum operating condition, each component must be laid out with the minimum distance.

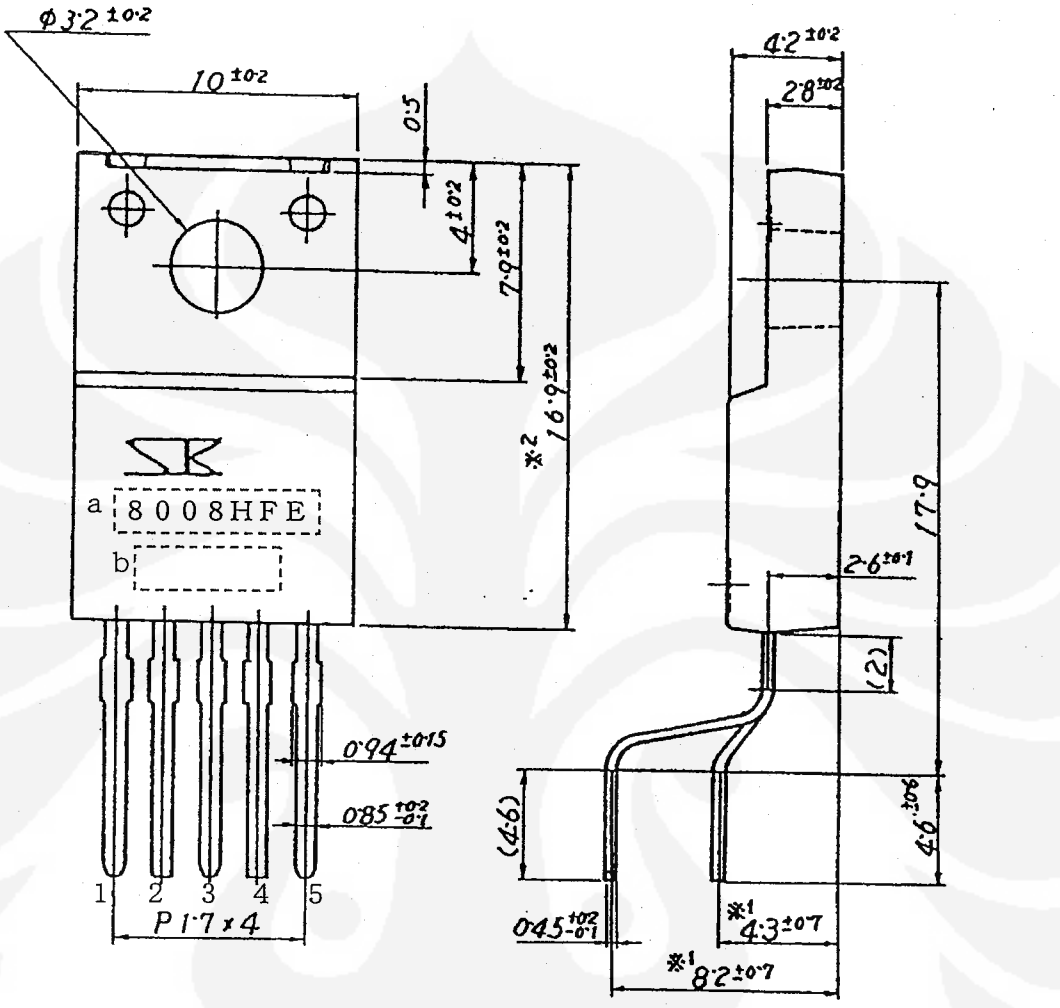
7．外形
Package information

7-2 寸法 (ﾘｰﾄﾞﾌｫｰﾐﾝｸﾞ No.LF1113)
Pphysical dimensions (Lead forming No.LF1113)

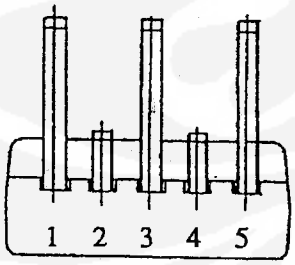
単位：㎜
Unit：㎜

外形寸法図 LF1113
Outline drawing of lead forming No.1113

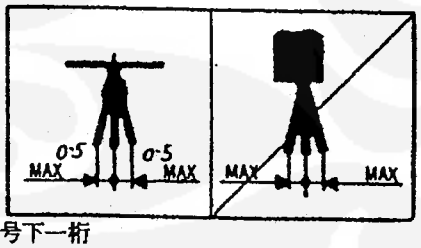
単位：mm
Dimensions in mm



●端子配列
Pin assignment
1：IN
2：SW
3：GND
4：ADJ
5：SS



a. 品名標示
Type Number
b. ロット番号
Lot Number
第1文字 1st letter 西暦年号下一桁 The last digit of year
第2文字 2nd letter 月 Month
1～9月：アラビア数字
10月：O
11月：N
12月：D
(1 to 9 for Jan. to Sept., O for Oct. N for Nov. D for Dec.)
第3,4文字 3rd & 4th letter 製造日 Day
01~31 アラビア数字 Arabic Numerical



●製品質量：約 2.3 g
Products Weight：Approx.2.3g

<注> ※2印寸法はゲートバリを含まず
The dimensions don't include the gate burr.
<注> ※1印寸法はリード先端部の寸法を示す。
Shows the dimensions measured at the top of lead.

DWG.NO.図番 TG3A-1113

● 外部端子処理：Sn-3Ag-0.5Cu ﾃﾞｨｯﾌﾟ
External terminal processing：Sn-3Ag-0.5Cu Dip

61426-01

7-2 外観
Appearance
本体は、汚れ、傷、亀裂等なく綺麗であること。
The body shall be clean and shall not bear at stain, rust or flaw.

7-3 標示
Marking
標示は本体に、品名及びロット番号を明瞭、かつ容易に消えぬようレーザーで捺印すること。
The type number and lot number shall be marked on the body by laser that shall not be unreadable easily.

7-4 推奨締め付けトルク
Screwing torque
ＳＩ－８０００ＨＦＥのトルクは、０．５８８～０．６８６Ｎ・ｍ（６．０～７．０ｋｇｆ・ｃｍ）として下さい。
The torque of screwing the SI-8000HFE to the heatsink shall be0.588~0.686N/m(6.0~7.0kgf/cm)

8 梱包
Packing

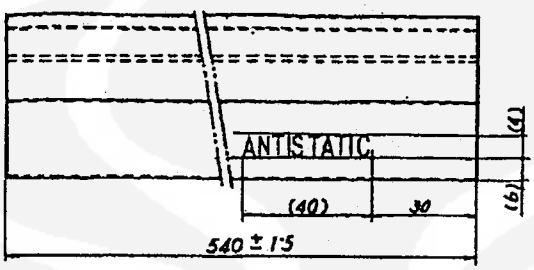
単位：㎜
Unit：㎜

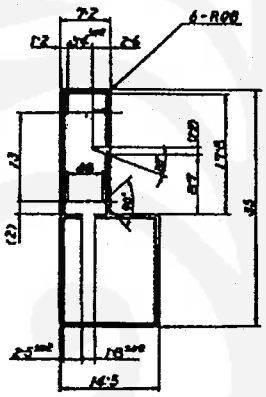
● SI-8000HFE シリーズ用　梱包
PACKING SPECIFICATIONS FOR SI-8000HFE SERIES

8-1 最小荷姿
Primary packing

8-1-1 スティック：スティック型式 …… FM-205 A
Stick : Type … FM-205 A

8-1-2 収納数：50個/1 スティック
Packing quantity : 50pieces per stick





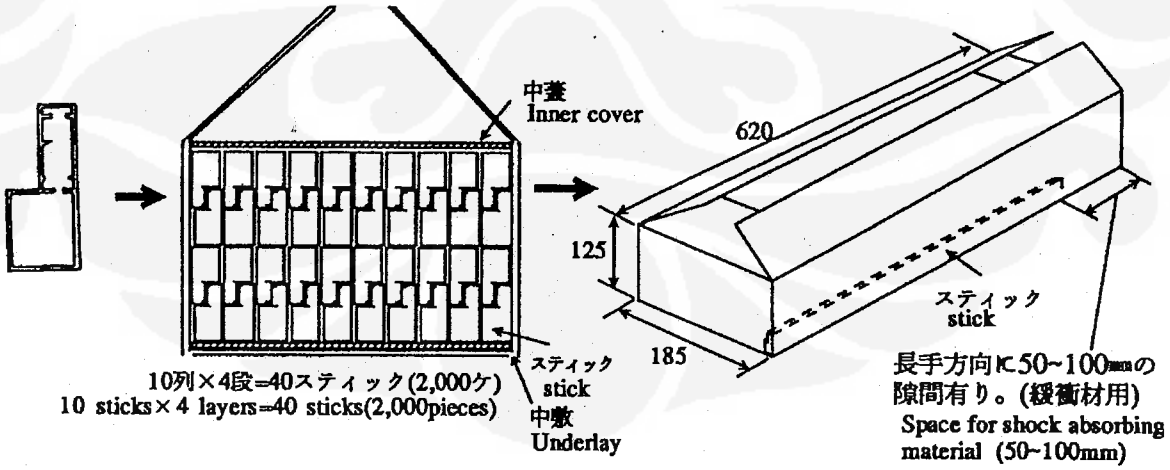
1. 製品肉厚：0.7mm±0.15
2. 材質：硬質塩化ビニル(透明)
3. ソリ：L字寸法に対し2mm以内
4. 印刷：色調は(青)とする。

1. Product thickness : 0.7mm± 0.15
2. Material : Hard polyvinyl(transparent)
3. Warp : within 2mm against L-shape dimensions
4. Printing color : Blue

8-2 単位荷姿
Secondary packing

8-2-1 梱包材：ダンボール箱
Packing material : Carton box

8-2-2 収納数：上記スティックを10列4段(40スティック)収納
Packing quantity : To pack 10 sticks by 4 layers (40 sticks) in a carton box



9. 使用上の注意
Cautions and warnings

9-1 並列運転について
Parallel operation
電流を増すための並列運転は出来ません。
The parallel operation to increase the current is not available.

9-2 過熱保護特性について
Thermal shut down
SI-8000HFE シリーズは過熱保護回路を内蔵しておりますが、これは瞬時短絡等の発熱に対し、ＩＣを保護する回路であり、長時間短絡等、発熱が継続状態での信頼性を含めた動作を保証するものではありません。
The SI-8000HFE series has a thermal protection circuit. This circuit keeps the IC from the fever by the over load. But this circuit cannot guarantee the long-term reliability against the continuously over load status.

9-3 放熱特性と信頼性
Heat radiation and reliability
一般に IC の信頼性は、その動作時の温度によって大きく左右されます。放熱には細心の注意を払い、放熱器の設計には充分余裕を設けて下さい。
また、その放熱器を SI-8000HFE に取り付ける際には、必ずシリコングリスを塗布してしっかりと締付けて下さい。シリコングリスには、当社推奨のものをご使用下さい。
The reliability of an IC is inseparable from the temperature in its operation. Careful consideration should be given to heat radiation and a sufficient safety margin must be allowed when designing a heat sink. When mounting the SI-8000HFE to the heat sink, be sure to apply silicone grease and securely screw it. Please use one of the following grease we suggest.

G746	信越化学工業(株)	SHIN-ETSU CHEMICAL CO., LTD.
SC102	東レ・ダウコーニング㈱	DOW CORNING TORAY CO.,LTD.
YG6260	モメンティブ・パフォーマンス・マテリアルズ・ジャパン合同会社	Momentive Performance Materials Inc SY6260SY6260

9-4 その他
Others
・本書に記載されている動作例及び回路例は、使用上の参考として示したもので、これらに起因する当社もしくは第三者の工業所有権、知的所有権、その他の権利の侵害問題について当社は一切責任を負いません。
Application and operation examples described in this document are quoted for the sole purpose of reference for the use of the products herein and Sanken can assume no responsibility for any infringement of industrial property right, intellectual property rights or any other rights of Sanken or any third party which may result form its use.

・本書に記載されている製品をご使用の場合は、これらの製品との組み合わせについて使用者の責任に於いて検討・判断を行って下さい。
When using the products herein, the applicability and suitability of such products for the intended purpose object shall be reviewed at the users responsibility.

・当社は品質、信頼性の向上に努めていますが、半導体製品では、ある確率での欠陥、故障の発生は避けられません。部品の故障により結果として、人身事故、火災事故、社会的な損害を発生させないよう、使用者の責任に於いて、装置やシステム上で十分な安全設計及び確認を行って下さい。
Although Sanken undertakes to enhance the quality and reliability of its products, the occurrence of failure and defect of semiconductor products at a certain rate is inevitable.
Users of Sanken products are requested to take, at their own risk, preventative measures including safety design of the equipment or systems against any possible injury, death, fires or damages to the society due to device failure or malfunction.

・本書に記載されている製品は、一般電子機器(家電製品、事務機器、通信端末機器、計測機器など)に使用されることを意図しております。
高い信頼性が要求される装置(輸送機器とその制御装置、交通信号制御装置、防災・防犯装置、各種安全装置など)への使用をご検討の際には、必ず当社販売窓口へご相談及び納入仕様書への記載をお願いします。極めて高い信頼性が要求される装置(航空宇宙機器、原子力制御、生命維持のための医療機器など)には当社の文書による合意がない限り使用しないで下さい。

Sanken products listed in this document are designed and intended for the use as components in general purpose electronic equipment or apparatus (home appliances, office equipment, telecommunication equipment, measuring equipment, etc.).
When considering the use of Sanken products in the applications where higher reliability is required (transportation equipment and its control systems, traffic signal control systems or equipment, fire/crime alarm systems, various safety devices, etc.), please contact your nearest Sanken sales representative to discuss and obtain written confirmation of your specifications.
The use of Sanken products without the written consent of Sanken in the applications where extremely high reliability is required (aerospace equipment, nuclear power control systems, life support systems, etc.) is strictly prohibited.

・本書に記載された製品は対放射線設計をしておりません。
Anti radioactive ray design is not considered for the products listed herein.

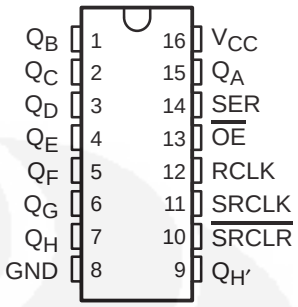
＜追　記＞

・サンケン電気株式会社の環境品質部品共通購入仕様書(CXA40000-505A)に適合しています。

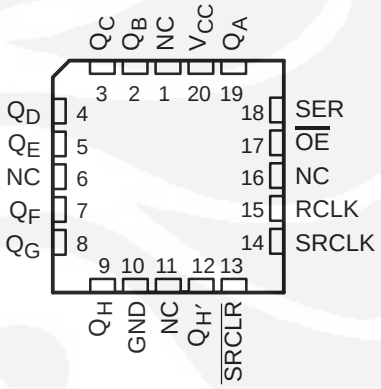
・RoHS 指令の禁止物質は含有しておりません。

- **8-Bit Serial-In, Parallel-Out Shift**
- **Wide Operating Voltage Range of 2 V to 6 V**
- **High-Current 3-State Outputs Can Drive Up To 15 LSTTL Loads**
- **Low Power Consumption, 80-μA Max I_{CC}**
- **Typical t_{pd} = 13 ns**
- **±6-mA Output Drive at 5 V**
- **Low Input Current of 1 μA Max**
- **Shift Register Has Direct Clear**

SN54HC595 . . . J OR W PACKAGE
SN74HC595 . . . D, DB, DW, N, OR NS PACKAGE
(TOP VIEW)



SN54HC595 . . . FK PACKAGE
(TOP VIEW)



NC − No internal connection

description/ordering information

The ’HC595 devices contain an 8-bit serial-in, parallel-out shift register that feeds an 8-bit D-type storage register. The storage register has parallel 3-state outputs. Separate clocks are provided for both the shift and storage register. The shift register has a direct overriding clear ($\overline{SRCLR}$) input, serial (SER) input, and serial outputs for cascading. When the output-enable ($\overline{OE}$) input is high, the outputs are in the high-impedance state.

Both the shift register clock (SRCLK) and storage register clock (RCLK) are positive-edge triggered. If both clocks are connected together, the shift register always is one clock pulse ahead of the storage register.

ORDERING INFORMATION

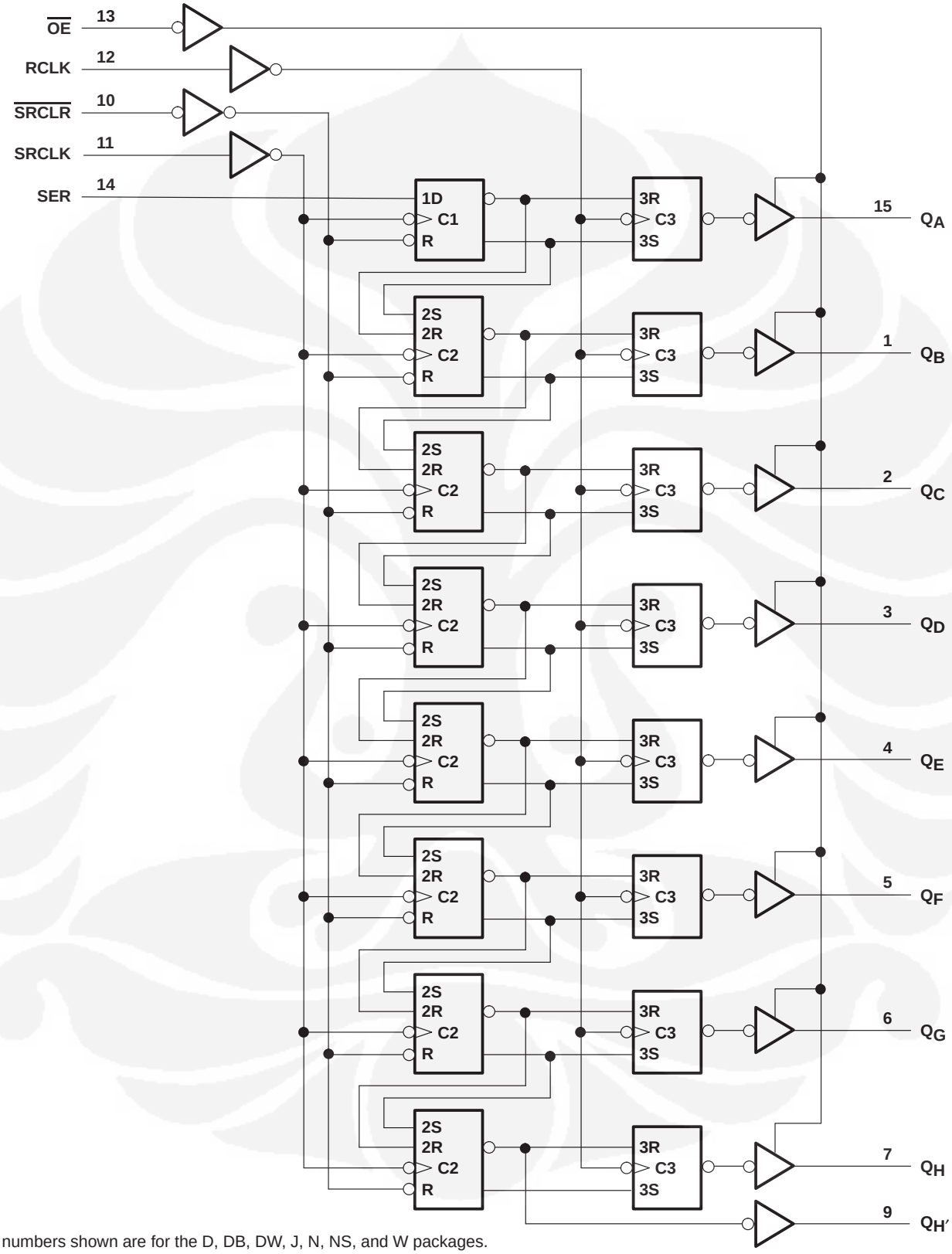
T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
−40°C to 85°C	PDIP − N	Tube of 25	SN74HC595N	SN74HC595N
	SOIC − D	Tube of 40	SN74HC595D	HC595
		Reel of 2500	SN74HC595DR	
		Reel of 250	SN74HC595DT	
	SOIC − DW	Tube of 40	SN74HC595DW	HC595
		Reel of 2000	SN74HC595DWR	
	SOP − NS	Reel of 2000	SN74HC595NSR	HC595
	SSOP − DB	Reel of 2000	SN74HC595DBR	HC595
−55°C to 125°C	CDIP − J	Tube of 25	SNJ54HC595J	SNJ54HC595J
	CFP − W	Tube of 150	SNJ54HC595W	SNJ54HC595W
	LCCC − FK	Tube of 55	SNJ54HC595FK	SNJ54HC595FK

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

FUNCTION TABLE

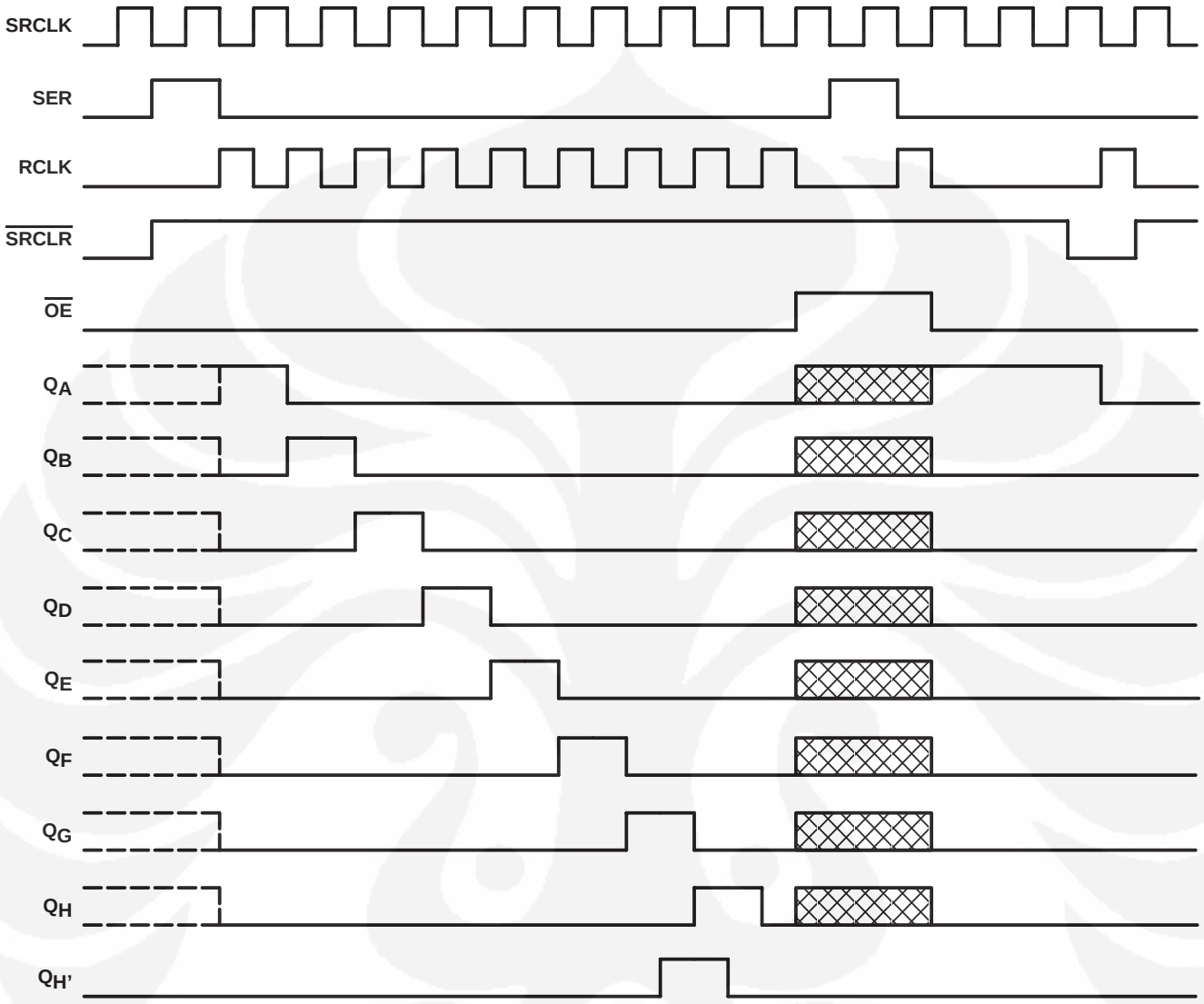
INPUTS					FUNCTION
SER	SRCLK	$\overline{\text{SRCLR}}$	RCLK	$\overline{\text{OE}}$	
X	X	X	X	H	Outputs Q_A–Q_H are disabled.
X	X	X	X	L	Outputs Q_A–Q_H are enabled.
X	X	L	X	X	Shift register is cleared.
L	↑	H	X	X	First stage of the shift register goes low. Other stages store the data of previous stage, respectively.
H	↑	H	X	X	First stage of the shift register goes high. Other stages store the data of previous stage, respectively.
X	X	X	↑	X	Shift-register data is stored in the storage register.

logic diagram (positive logic)



Pin numbers shown are for the D, DB, DW, J, N, NS, and W packages.

timing diagram



NOTE: ⊠ implies that the output is in 3-State mode.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage range, V_{CC} . . . . . −0.5 V to 7 V
Input clamp current, I_{IK} ($V_I < 0$ or $V_I > V_{CC}$) (see Note 1) . . . . . ±20 mA
Output clamp current, I_{OK} ($V_O < 0$ or $V_O > V_{CC}$) (see Note 1) . . . . . ±20 mA
Continuous output current, I_O (V_O = 0 to V_{CC}) . . . . . ±35 mA
Continuous current through V_{CC} or GND . . . . . ±70 mA
Package thermal impedance, θ_{JA} (see Note 2): D package . . . . . 73°C/W
DB package . . . . . 82°C/W
DW package . . . . . 57°C/W
N package . . . . . 67°C/W
NS package . . . . . 64°C/W
Storage temperature range, T_{stg} . . . . . −65°C to 150°C

† 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.

NOTES: 1. The input and output voltage ratings may be exceeded if the input and output current ratings are observed.
2. The package thermal impedance is calculated in accordance with JESD 51-7.

recommended operating conditions (see Note 3)

			SN54HC595			SN74HC595			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage		2	5	6	2	5	6	V
V_{IH}	High-level input voltage	V_{CC} = 2 V	1.5			1.5			V
		V_{CC} = 4.5 V	3.15			3.15			
		V_{CC} = 6 V	4.2			4.2			
V_{IL}	Low-level input voltage	V_{CC} = 2 V			0.5			0.5	V
		V_{CC} = 4.5 V			1.35			1.35	
		V_{CC} = 6 V			1.8			1.8	
V_I	Input voltage		0		V_{CC}	0		V_{CC}	V
V_O	Output voltage		0		V_{CC}	0		V_{CC}	V
$\Delta t/\Delta v$‡	Input transition rise/fall time	V_{CC} = 2 V			1000			1000	ns
		V_{CC} = 4.5 V			500			500	
		V_{CC} = 6 V			400			400	
T_A	Operating free-air temperature		−55		125	−40		85	°C

NOTE 3: All unused inputs of the device must be held at V_{CC} or GND to ensure proper device operation. Refer to the TI application report, *Implications of Slow or Floating CMOS Inputs*, literature number SCBA004.

‡ If this device is used in the threshold region (from V_{IL}max = 0.5 V to V_{IH}min = 1.5 V), there is a potential to go into the wrong state from induced grounding, causing double clocking. Operating with the inputs at t_t = 1000 ns and V_{CC} = 2 V does not damage the device; however, functionally, the CLK inputs are not ensured while in the shift, count, or toggle operating modes.

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS		V_{CC}	T_A = 25°C			SN54HC595		SN74HC595		UNIT
				MIN	TYP	MAX	MIN	MAX	MIN	MAX	
V_{OH}	V_I = V_{IH} or V_{IL}	I_{OH} = −20 μA	2 V	1.9	1.998		1.9		1.9		V
			4.5 V	4.4	4.499		4.4		4.4		
			6 V	5.9	5.999		5.9		5.9		
		$Q_{H'}$, I_{OH} = −4 mA	4.5 V	3.98	4.3		3.7		3.84		
		Q_A–Q_H, I_{OH} = −6 mA		3.98	4.3		3.7		3.84		
		$Q_{H'}$, I_{OH} = −5.2 mA	6 V	5.48	5.8		5.2		5.34		
		Q_A–Q_H, I_{OH} = −7.8 mA		5.48	5.8		5.2		5.34		
V_{OL}	V_I = V_{IH} or V_{IL}	I_{OL} = 20 μA	2 V		0.002	0.1		0.1		0.1	V
			4.5 V		0.001	0.1		0.1		0.1	
			6 V		0.001	0.1		0.1		0.1	
		$Q_{H'}$, I_{OL} = 4 mA	4.5 V		0.17	0.26		0.4		0.33	
		Q_A–Q_H, I_{OL} = 6 mA			0.17	0.26		0.4		0.33	
		$Q_{H'}$, I_{OL} = 5.2 mA	6 V		0.15	0.26		0.4		0.33	
		Q_A–Q_H, I_{OL} = 7.8 mA			0.15	0.26		0.4		0.33	
I_I	V_I = V_{CC} or 0		6 V		±0.1	±100		±1000		±1000	nA
I_{OZ}	V_O = V_{CC} or 0,	Q_A–Q_H	6 V		±0.01	±0.5		±10		±5	μA
I_{CC}	V_I = V_{CC} or 0,	I_O = 0	6 V			8		160		80	μA
C_i			2 V to 6 V		3	10		10		10	pF

timing requirements over recommended operating free-air temperature range (unless otherwise noted)

			V_{CC}	T_A = 25°C		SN54HC595		SN74HC595		UNIT
				MIN	MAX	MIN	MAX	MIN	MAX	
f_{clock}	Clock frequency		2 V		6		4.2		5	MHz
			4.5 V		31		21		25	
			6 V		36		25		29	
t_w	Pulse duration	SRCLK or RCLK high or low	2 V	80		120		100		ns
			4.5 V	16		24		20		
			6 V	14		20		17		
		$\overline{\text{SRCLR}}$ low	2 V	80		120		100		
			4.5 V	16		24		20		
			6 V	14		20		17		
t_{su}	Setup time	SER before SRCLK↑	2 V	100		150		125		ns
			4.5 V	20		30		25		
			6 V	17		25		21		
		SRCLK↑ before RCLK↑†	2 V	75		113		94		
			4.5 V	15		23		19		
			6 V	13		19		16		
		$\overline{\text{SRCLR}}$ low before RCLK↑	2 V	50		75		65		
			4.5 V	10		15		13		
			6 V	9		13		11		
		$\overline{\text{SRCLR}}$ high (inactive) before SRCLK↑	2 V	50		75		60		
			4.5 V	10		15		12		
			6 V	9		13		11		
t_h	Hold time, SER after SRCLK↑		2 V	0		0		0		ns
			4.5 V	0		0		0		
			6 V	0		0		0		

† This setup time allows the storage register to receive stable data from the shift register. The clocks can be tied together, in which case the shift register is one clock pulse ahead of the storage register.

switching characteristics over recommended operating free-air temperature range, C_L = 50 pF (unless otherwise noted) (see Figure 1)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	V_{CC}	T_A = 25°C			SN54HC595		SN74HC595		UNIT
				MIN	TYP	MAX	MIN	MAX	MIN	MAX	
f_{max}			2 V	6	26		4.2		5		MHz
			4.5 V	31	38		21		25		
			6 V	36	42		25		29		
t_{pd}	SRCLK	$Q_{H'}$	2 V		50	160		240		200	ns
			4.5 V		17	32		48		40	
			6 V		14	27		41		34	
	RCLK	Q_A–Q_H	2 V		50	150		225		187	
			4.5 V		17	30		45		37	
			6 V		14	26		38		32	
t_{PHL}	$\overline{SRCLR}$	$Q_{H'}$	2 V		51	175		261		219	ns
			4.5 V		18	35		52		44	
			6 V		15	30		44		37	
t_{en}	$\overline{OE}$	Q_A–Q_H	2 V		40	150		225		187	ns
			4.5 V		15	30		45		37	
			6 V		13	26		38		32	
t_{dis}	$\overline{OE}$	Q_A–Q_H	2 V		42	200		300		250	ns
			4.5 V		23	40		60		50	
			6 V		20	34		51		43	
t_t		Q_A–Q_H	2 V		28	60		90		75	ns
			4.5 V		8	12		18		15	
			6 V		6	10		15		13	
		$Q_{H'}$	2 V		28	75		110		95	
			4.5 V		8	15		22		19	
			6 V		6	13		19		16	

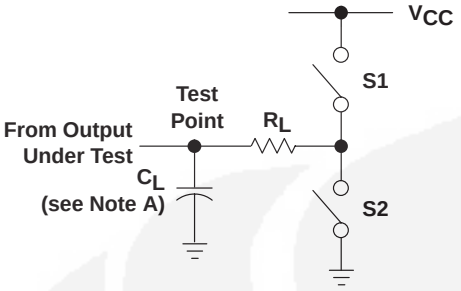
switching characteristics over recommended operating free-air temperature range, C_L = 150 pF (unless otherwise noted) (see Figure 1)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	V_{CC}	T_A = 25°C			SN54HC595		SN74HC595		UNIT
				MIN	TYP	MAX	MIN	MAX	MIN	MAX	
t_{pd}	RCLK	Q_A–Q_H	2 V		60	200		300		250	ns
			4.5 V		22	40		60		50	
			6 V		19	34		51		43	
t_{en}	$\overline{OE}$	Q_A–Q_H	2 V		70	200		298		250	ns
			4.5 V		23	40		60		50	
			6 V		19	34		51		43	
t_t		Q_A–Q_H	2 V		45	210		315		265	ns
			4.5 V		17	42		63		53	
			6 V		13	36		53		45	

operating characteristics, T_A = 25°C

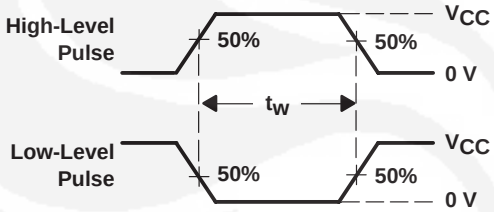
PARAMETER		TEST CONDITIONS	TYP	UNIT
C_{pd}	Power dissipation capacitance	No load	400	pF

PARAMETER MEASUREMENT INFORMATION

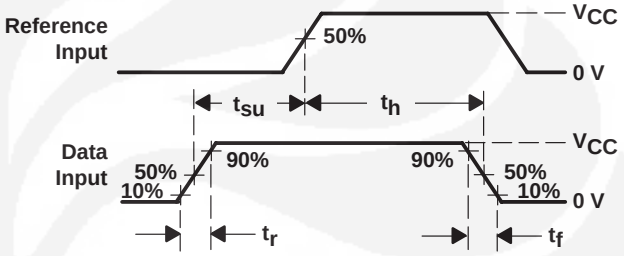


LOAD CIRCUIT

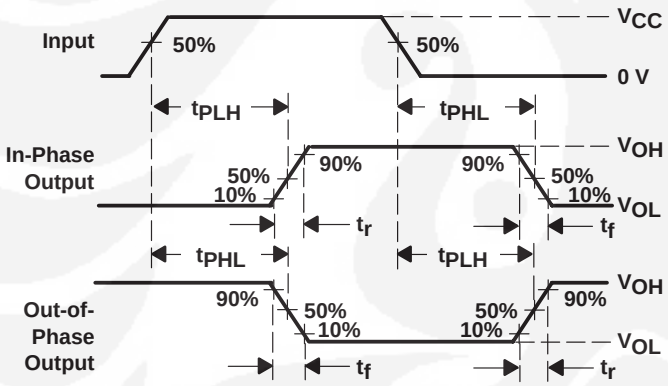
PARAMETER		R_L	C_L	S1	S2
t_{en}	t_{PZH}	1 kΩ	50 pF or 150 pF	Open	Closed
	t_{PZL}			Closed	Open
t_{dis}	t_{PHZ}	1 kΩ	50 pF	Open	Closed
	t_{PLZ}			Closed	Open
t_{pd} or t_t		−−	50 pF or 150 pF	Open	Open



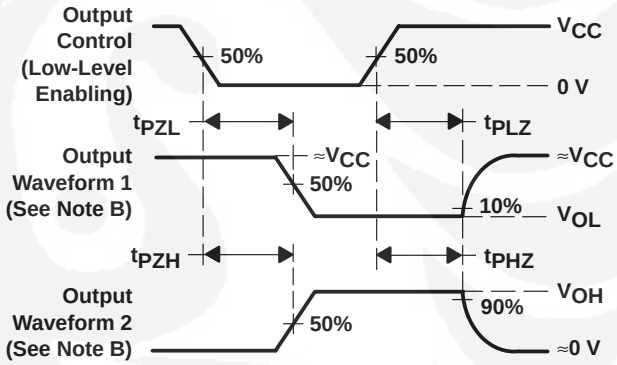
VOLTAGE WAVEFORMS
PULSE DURATIONS



VOLTAGE WAVEFORMS
SETUP AND HOLD AND INPUT RISE AND FALL TIMES



VOLTAGE WAVEFORMS
PROPAGATION DELAY AND OUTPUT TRANSITION TIMES



VOLTAGE WAVEFORMS
ENABLE AND DISABLE TIMES FOR 3-STATE OUTPUTS

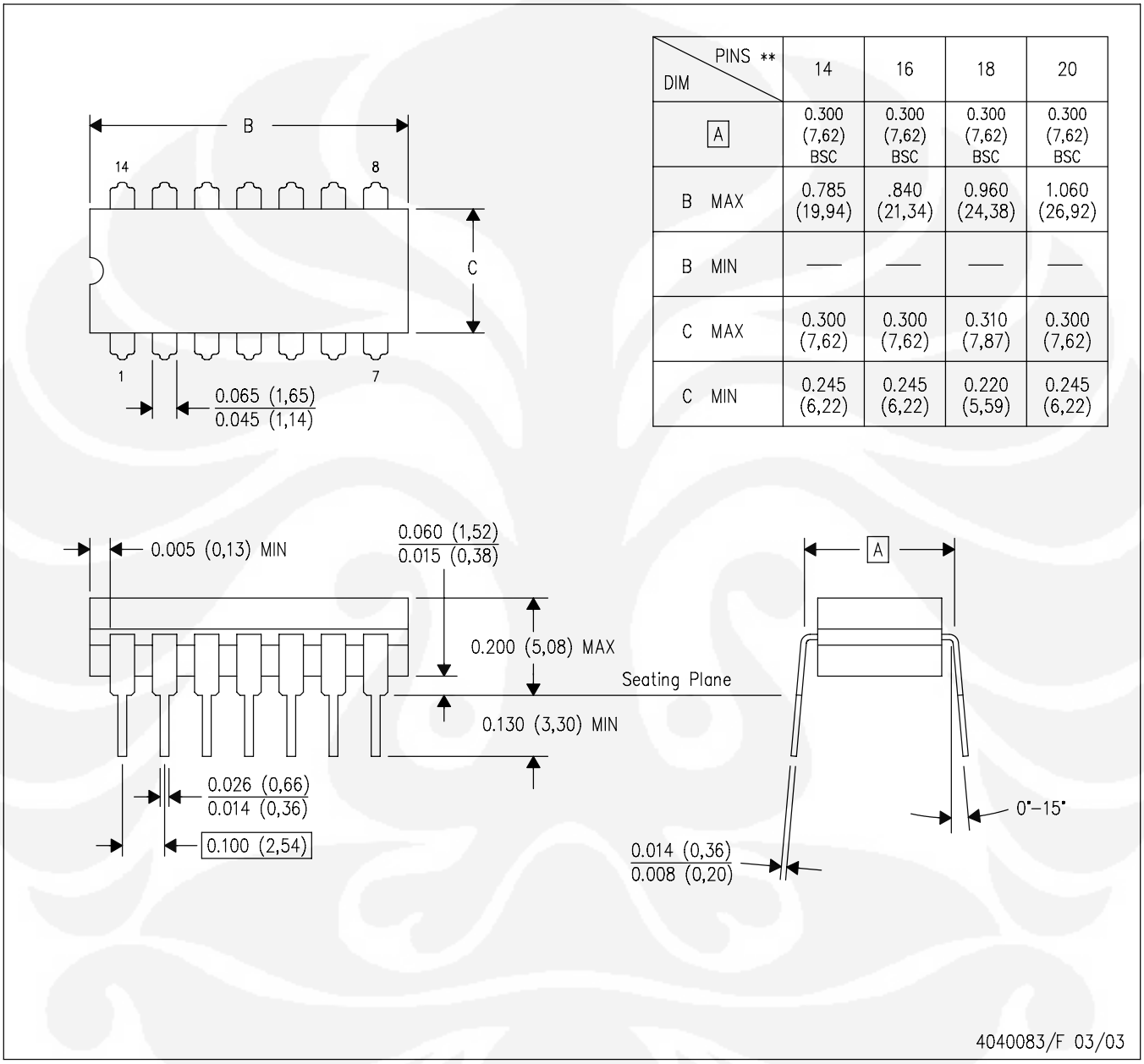
NOTES: A. C_L includes probe and test-fixture capacitance.
B. Waveform 1 is for an output with internal conditions such that the output is low, except when disabled by the output control. Waveform 2 is for an output with internal conditions such that the output is high, except when disabled by the output control.
C. Phase relationships between waveforms were chosen arbitrarily. All input pulses are supplied by generators having the following characteristics: PRR ≤ 1 MHz, Z_O = 50 Ω, t_r = 6 ns, t_f = 6 ns.
D. For clock inputs, f_{max} is measured when the input duty cycle is 50%.
E. The outputs are measured one at a time, with one input transition per measurement.
F. t_{PLZ} and t_{PHZ} are the same as t_{dis}.
G. t_{PZL} and t_{PZH} are the same as t_{en}.
H. t_{PLH} and t_{PHL} are the same as t_{pd}.

Figure 1. Load Circuit and Voltage Waveforms

J (R–GDIP–T**)

14 LEADS SHOWN

CERAMIC DUAL IN–LINE PACKAGE

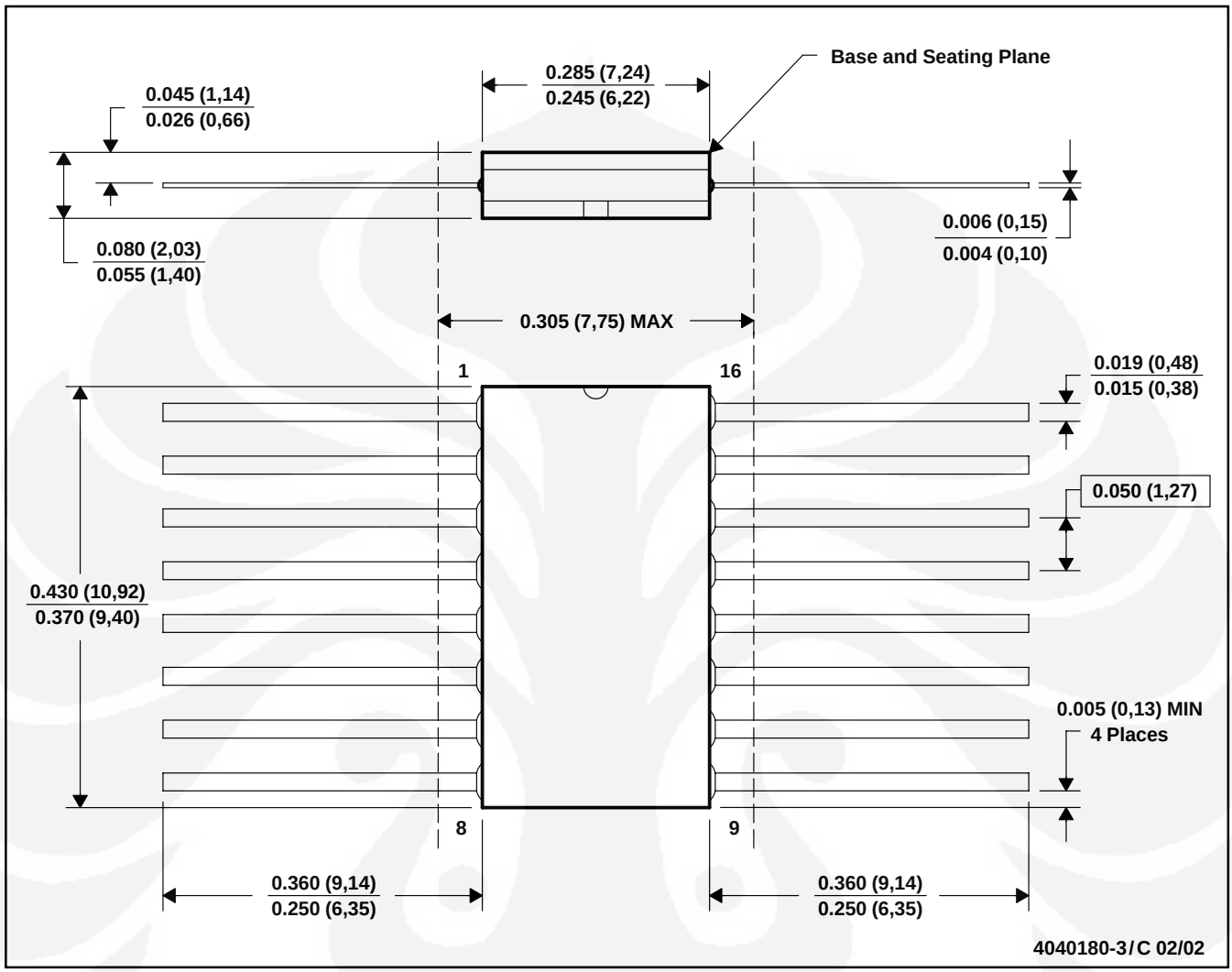


DIM \ PINS **	14	16	18	20
A	0.300 (7,62) BSC	0.300 (7,62) BSC	0.300 (7,62) BSC	0.300 (7,62) BSC
B MAX	0.785 (19,94)	.840 (21,34)	0.960 (24,38)	1.060 (26,92)
B MIN	—	—	—	—
C MAX	0.300 (7,62)	0.300 (7,62)	0.310 (7,87)	0.300 (7,62)
C MIN	0.245 (6,22)	0.245 (6,22)	0.220 (5,59)	0.245 (6,22)

NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package is hermetically sealed with a ceramic lid using glass frit.
D. Index point is provided on cap for terminal identification only on press ceramic glass frit seal only.
E. Falls within MIL STD 1835 GDIP1–T14, GDIP1–T16, GDIP1–T18 and GDIP1–T20.

MCFP004A– JANUARY 1995 – REVISED FEBRUARY 2002

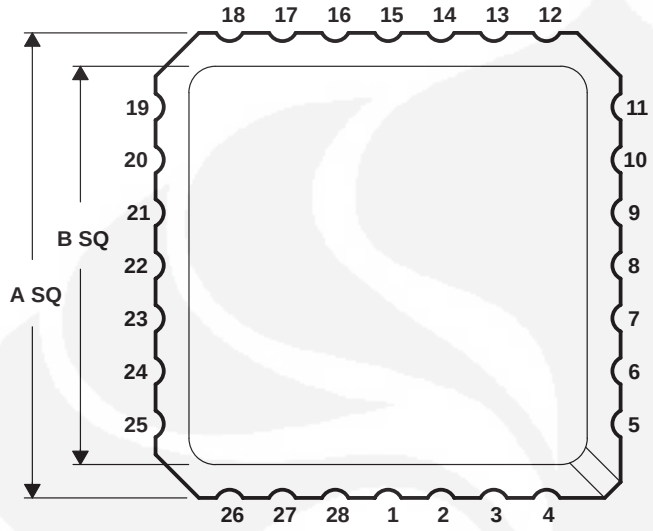
W (R-GDFP-F16) **CERAMIC DUAL FLATPACK**



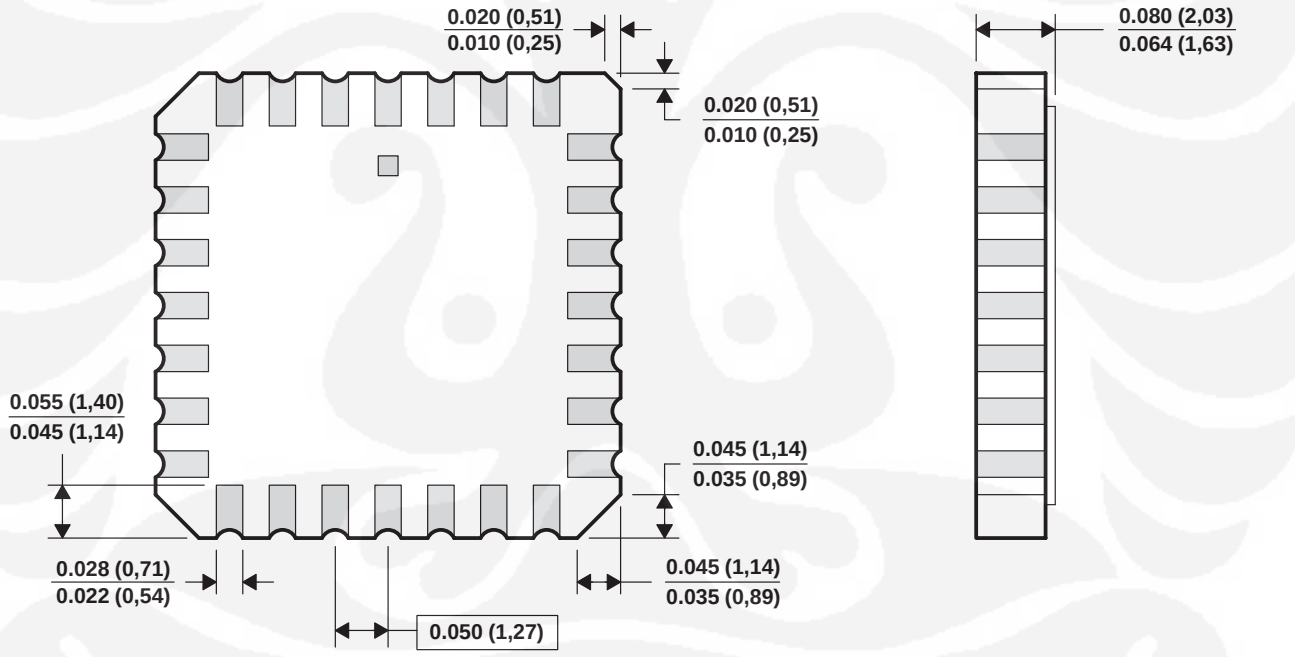
NOTES: A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package can be hermetically sealed with a ceramic lid using glass frit.
D. Index point is provided on cap for terminal identification only.
E. Falls within MIL STD 1835 GDFP-1F16 and JEDEC MO-092AC

FK (S-CQCC-N)** **LEADLESS CERAMIC CHIP CARRIER**

28 TERMINAL SHOWN



NO. OF TERMINALS **	A		B	
	MIN	MAX	MIN	MAX
20	0.342 (8,69)	0.358 (9,09)	0.307 (7,80)	0.358 (9,09)
28	0.442 (11,23)	0.458 (11,63)	0.406 (10,31)	0.458 (11,63)
44	0.640 (16,26)	0.660 (16,76)	0.495 (12,58)	0.560 (14,22)
52	0.739 (18,78)	0.761 (19,32)	0.495 (12,58)	0.560 (14,22)
68	0.938 (23,83)	0.962 (24,43)	0.850 (21,6)	0.858 (21,8)
84	1.141 (28,99)	1.165 (29,59)	1.047 (26,6)	1.063 (27,0)

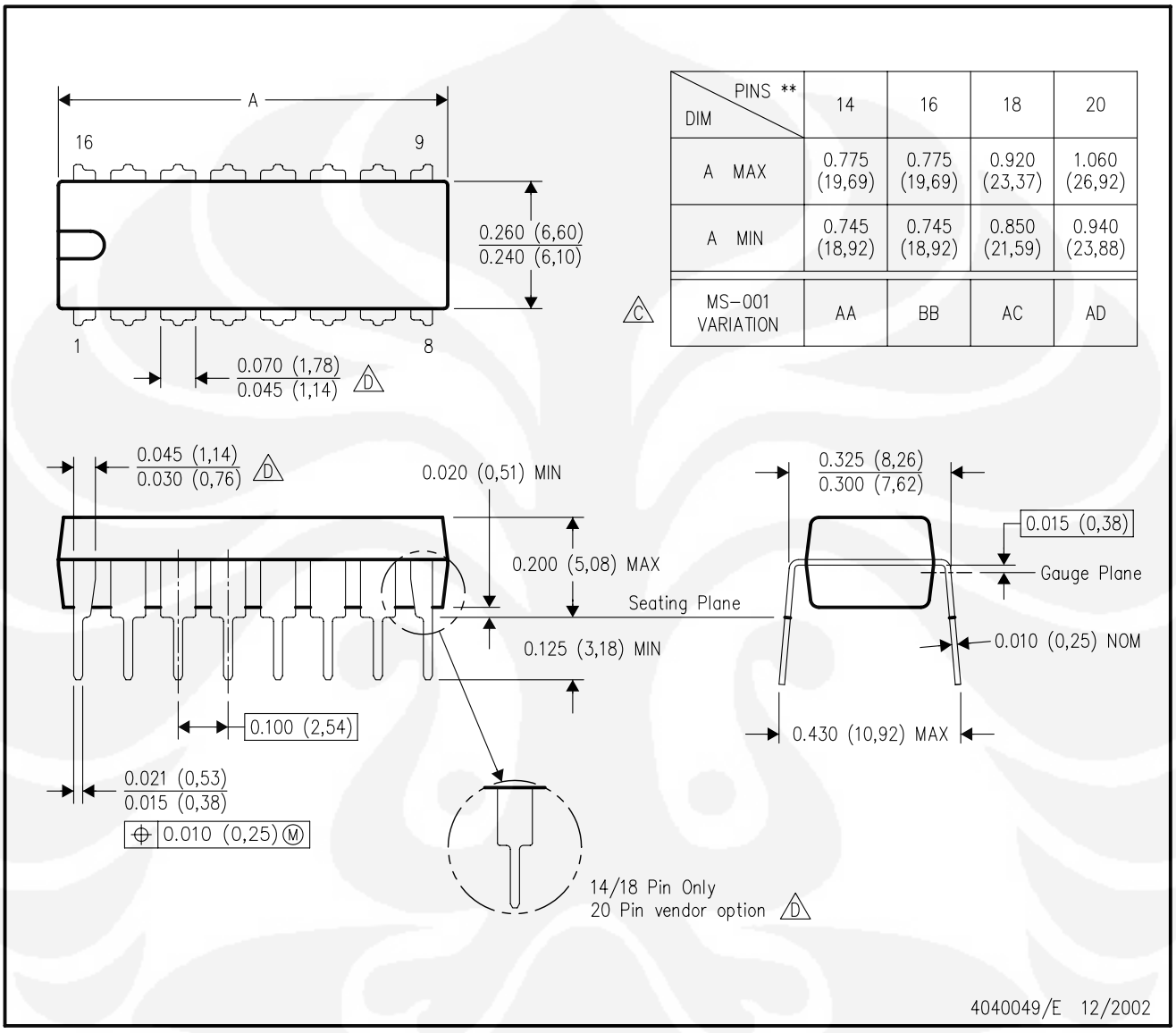


4040140/D 10/96

NOTES: A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package can be hermetically sealed with a metal lid.
D. The terminals are gold plated.
E. Falls within JEDEC MS-004

N (R–PDIP–T**) PLASTIC DUAL–IN–LINE PACKAGE

16 PINS SHOWN



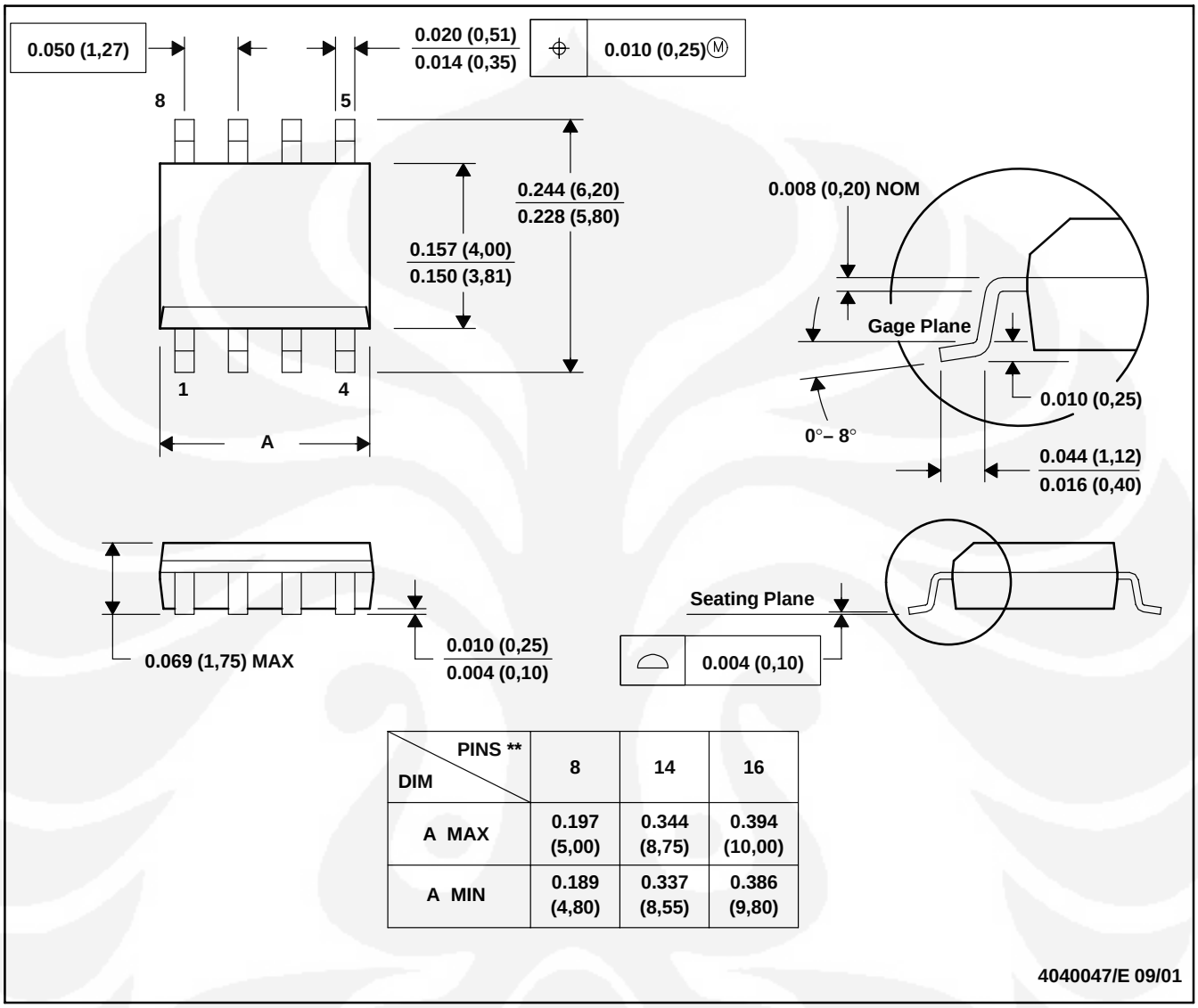
DIM \ PINS **	14	16	18	20
A MAX	0.775 (19,69)	0.775 (19,69)	0.920 (23,37)	1.060 (26,92)
A MIN	0.745 (18,92)	0.745 (18,92)	0.850 (21,59)	0.940 (23,88)
MS–001 VARIATION	AA	BB	AC	AD

NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS–001, except 18 and 20 pin minimum body length (Dim A).
D. The 20 pin end lead shoulder width is a vendor option, either half or full width.

MSOI002B – JANUARY 1995 – REVISED SEPTEMBER 2001

D (R-PDSO-G)** **PLASTIC SMALL-OUTLINE PACKAGE**

8 PINS SHOWN



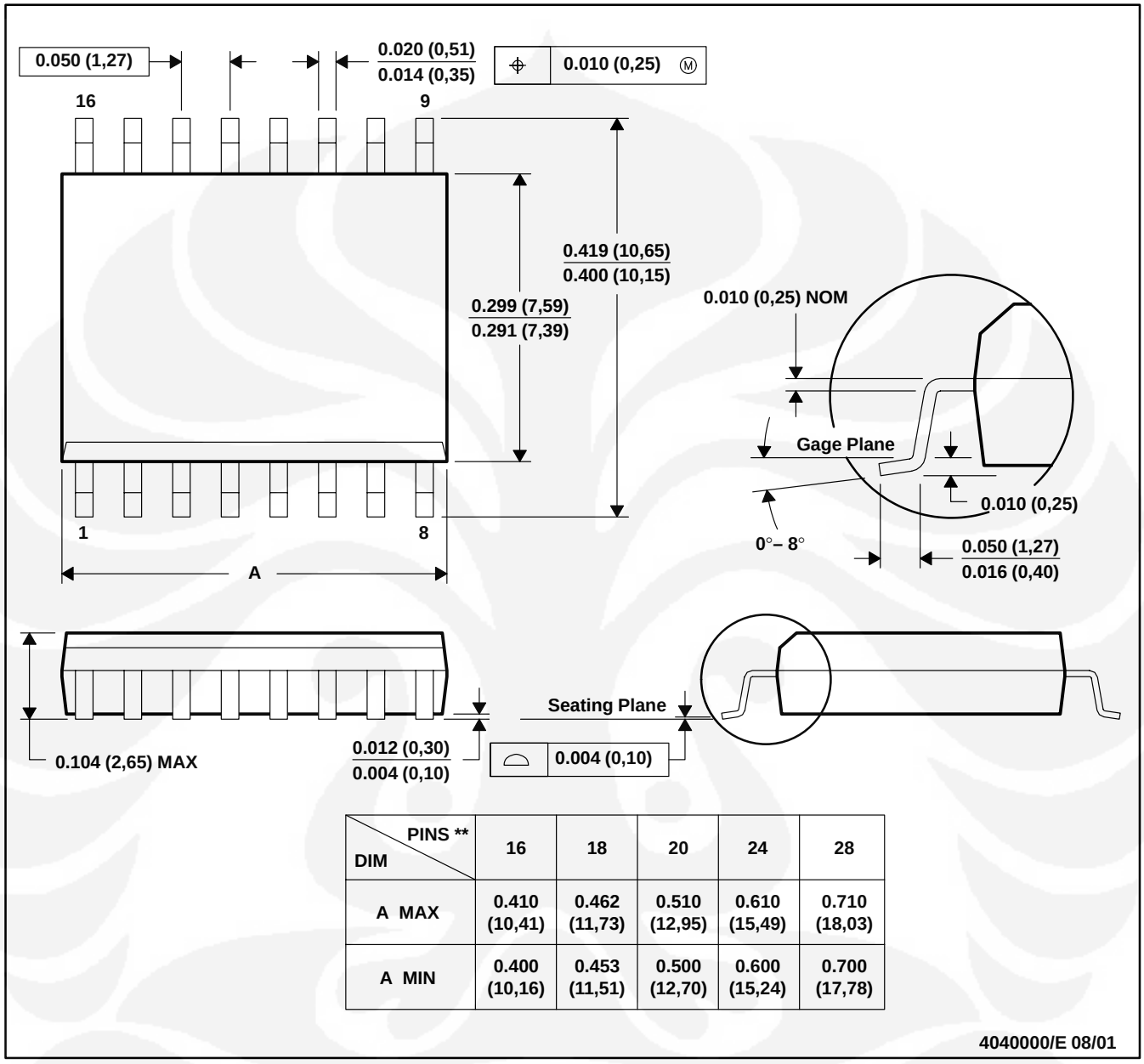
DIM \ PINS **	8	14	16
A MAX	0.197 (5,00)	0.344 (8,75)	0.394 (10,00)
A MIN	0.189 (4,80)	0.337 (8,55)	0.386 (9,80)

NOTES: A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion, not to exceed 0.006 (0,15).
D. Falls within JEDEC MS-012

MSOI003E – JANUARY 1995 – REVISED SEPTEMBER 2001

DW (R-PDSO-G)** **PLASTIC SMALL-OUTLINE PACKAGE**

16 PINS SHOWN



DIM \ PINS **	16	18	20	24	28
A MAX	0.410 (10,41)	0.462 (11,73)	0.510 (12,95)	0.610 (15,49)	0.710 (18,03)
A MIN	0.400 (10,16)	0.453 (11,51)	0.500 (12,70)	0.600 (15,24)	0.700 (17,78)

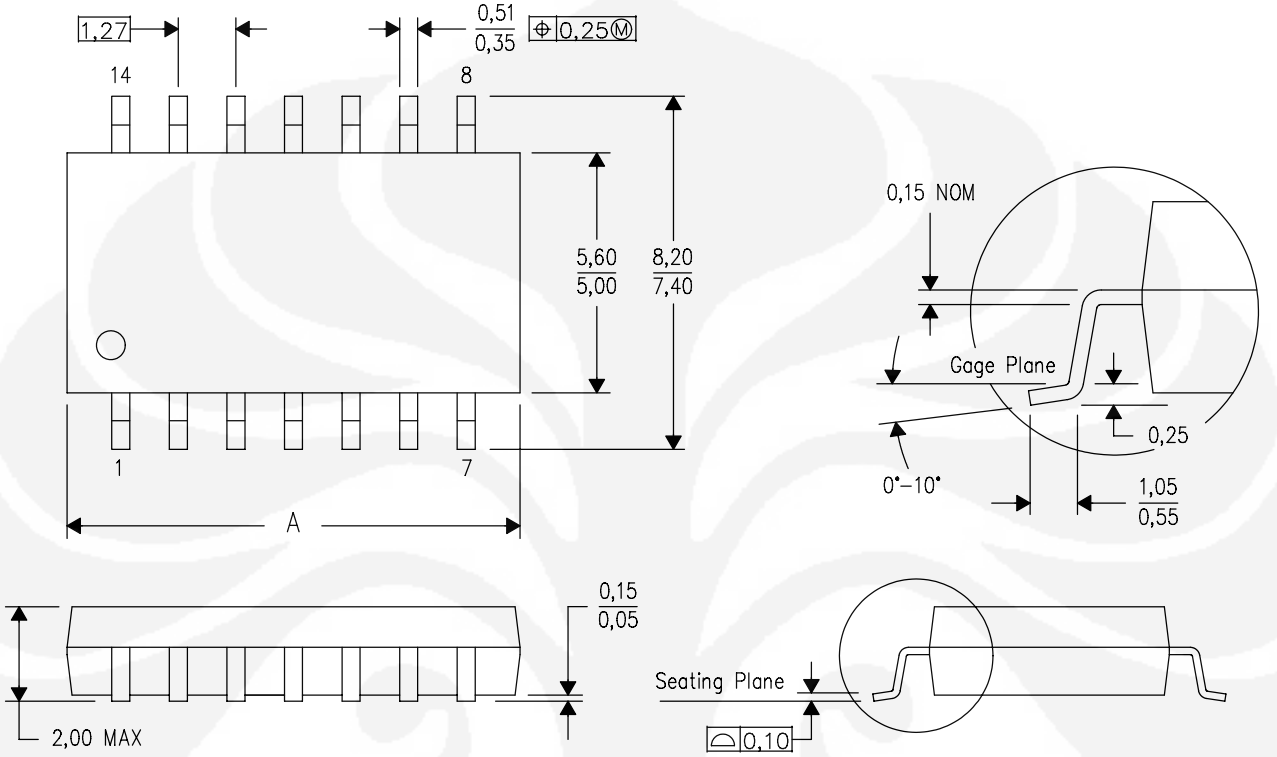
4040000/E 08/01

NOTES: A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion not to exceed 0.006 (0,15).
D. Falls within JEDEC MS-013

MECHANICAL DATA

NS (R-PDSO-G)** **PLASTIC SMALL-OUTLINE PACKAGE**

14-PINS SHOWN



DIM \ PINS **	14	16	20	24
A MAX	10,50	10,50	12,90	15,30
A MIN	9,90	9,90	12,30	14,70

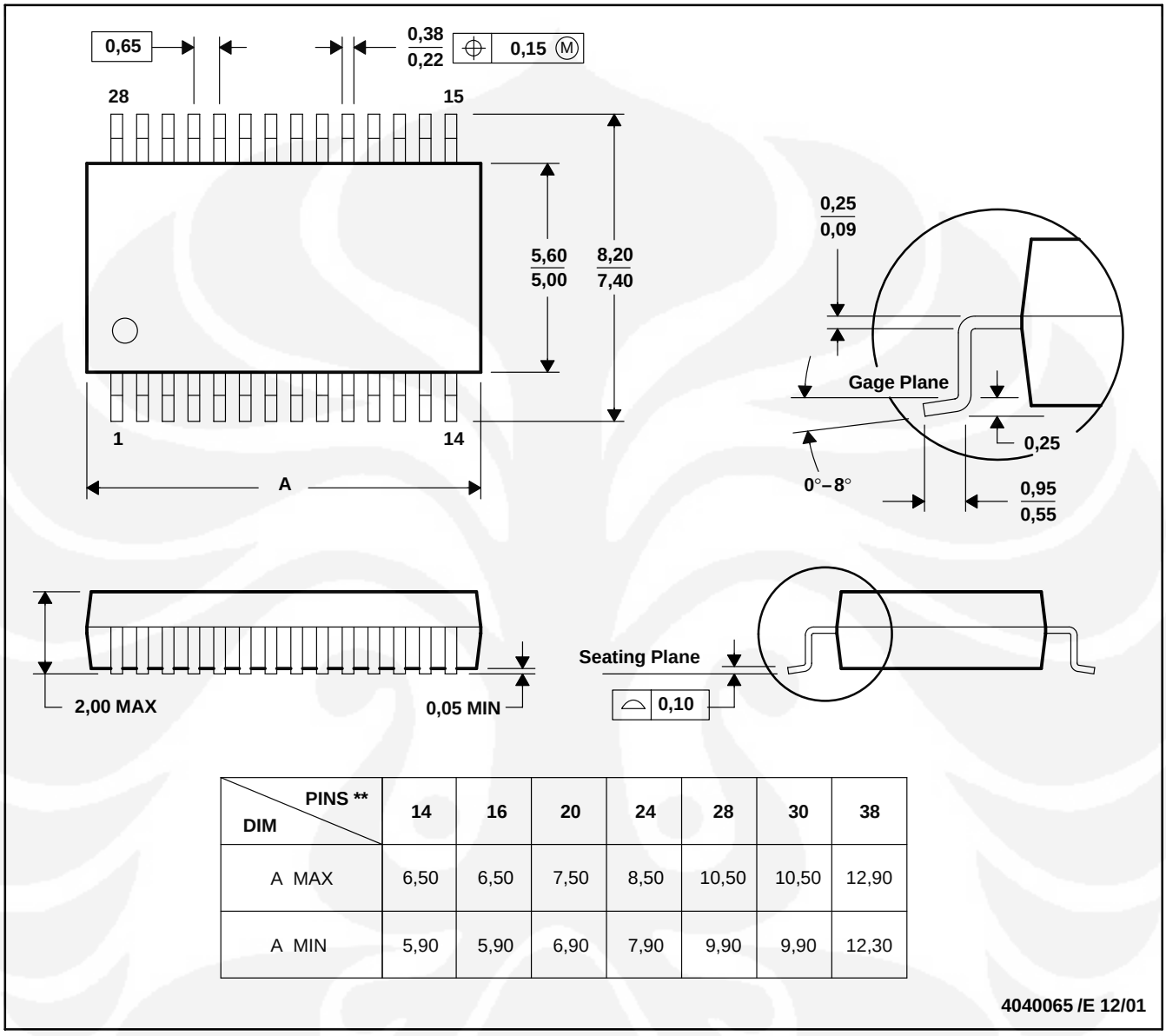
4040062/C 03/03

NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion, not to exceed 0,15.

MSSO002E – JANUARY 1995 – REVISED DECEMBER 2001

DB (R-PDSO-G)** **PLASTIC SMALL-OUTLINE**

28 PINS SHOWN



DIM \ PINS **	14	16	20	24	28	30	38
A MAX	6,50	6,50	7,50	8,50	10,50	10,50	12,90
A MIN	5,90	5,90	6,90	7,90	9,90	9,90	12,30

4040065 /E 12/01

NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion not to exceed 0,15.
D. Falls within JEDEC MO-150

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265