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

SKRIPSI

PERENCANAAN DAN PEMBUATAN SISTEM PENGATURAN LAMPU LALULINTAS 4 JALUR DI LENGKAPI *KEYPAD* DAN *DOT MATRIK* BERBASIS MIKROKONTROLER RENESAS R8C/13 TINY DAN AT89C51

Disusun Oleh:
MOHAMMAD KURNIAWAN
02. 17. 157

MARET 2007

ВЫСШАЯ ШКОЛА

ББК 32.97-02
Г81
Рецензенты: кафедра вычислительной техники Московского энергетического института; доктор технических наук профессор Ю. М. Смирнов

Гуров В. В.
Г81 Архитектура микропроцессоров: Учеб. пособие для вузов. — М.: Высш. шк., 1990. — 191 с.: ил.
ISBN 5-06-001237-2

Рассмотрены принципы построения и функционирования микропроцессоров и микропроцессорных систем. Описаны архитектура, структура, система команд и программирование однокристальных микропроцессоров и микроЭВМ. Приведены примеры построения микропроцессорных систем.
Для студентов вузов, обучающихся по специальности «Вычислительные машины, системы и сети».

Г 2404000000-141 / 001(01)-90 — 141-90

ББК 32.97-02

LEMBAR PERSETUJUAN

PERENCANAAN DAN PEMBUATAN SISTEM PENGATURAN LAMPU LALULINTAS 4 JALUR DILENGKAPI *KEYPAD* DAN *DOT MATRIKS* BERBASIS MIKROKONTROLER RENESAS R8C/13 TINY DAN AT89C51.

SKRIPSI

Diajukan Guna Memenuhi Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik Pada Jurusan Teknik Elektro S-1 Konsentrasi Elektronika

Disusun Oleh :

MOHAMMAD KURNIAWAN

NIM : 02. 17. 157

Diperiksa dan Disetujui

Ketua Jurusan Teknik Elektro S-1

(Ir. F. Yudi Limpraptono, MT)

NIP.P 1039500274

Dosen Pembimbing

(Ir. Widodo Pudji M., MT)

NIP.P 1028700171

KONSENTRASI ELEKTRONIKA

JURUSAN TEKNIK ELEKTRO S - 1

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Karanglo KM. 2 Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

NAMA	: Mohammad Kurniawan
NIM	: 02 17 157
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik Elektronika
Masa Bimbingan	: 1 Januari 2007 s/d 1 Juli 2007
Judul Skripsi	: Perencanaan Dan Pembuatan Sistem Pengaturan Lampu Lalulintas 4 Jalur Dilengkapi Keypad dan Dot Matriks Berbasis Mikrokontroler Renesas R8C/13 Tiny dan AT89C51.

Dipertahankan di hadapan Tim Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari	: Sabtu
Tanggal	: 17 Maret 2007
Nilai	: 73.9 (B+)

Panitia Majelis Penguji,

(Ir. Mochtar Asroni, MSME)
NIP.Y. 1018100036

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

Anggota Penguji

(Joseph Dedy Irawan, ST. MT.)
NIP. 132 315 128

(Moh. Ashar, ST. MT.)
NIP.

ABSTRAKSI

PERANCANGAN DAN PEMBUATAN SISTEM PENGATURAN LAMPU LALULINTAS 4 JALUR DILENGKAPI *KEYPAD* DAN *DOT MATRIKS* BERBASIS MIKROKONTROLLER RENESAS R8C/13 TINY DAN AT89C51

(Mohammad Kurniawan, 02.17.157, Teknik Elektro S.1/Elektronika)
(Dosen Pembimbing : Ir. Widodo Pudji M., MT.)

Kata Kunci : Pengaturan Lalulintas (*Traffic Light*), LCD, *keypad,* Dot Matrik, Mikrokontroler.

Pada skripsi ini dirancang sebuah sistem *Pengaturan Lampu Lalulintas (Traffic Light)*, dimana sistem ini kerjanya adalah berdasarkan survei kepadatan kendaraan bermtor yang menggunakan timer sebagai inputan yang terbagi menjadi 4 (Empat) keadaan. Keadaan pertama dari jam 06.00-14.00, keadaan kedua jam 14.01-17.00, keadaan ketiga 17.01-22.00, keadaan keempat *stand bye* dengan tanda lampu warna kuning nyala berkedip sampai jam 05.59.

Kemudian menyeting lama *delay time* dari keempat keadan tersebut dan disetiap keadaan *delay time*nya tidak sama, bila pada keadaan pertama dan ketiga *delay time* agak lama kerena pada saat itu kendaraan padat, keadaan kedua agak padat. Lama *delay time* diatur sesuai kapadatan kendaran bermotor dan sesuai dengan ketentuan DLLAJ (Dinas Lalulintas Angkutan Jalan).

Dalam sistem ini timer sebagai inputan diambil mikrokontroler RENESAS dan ditampilkan ke *LCD*,dan *delay time* yang suadah diatur pada setiap keadaanya akan ditampilkan melalu *Dot Matrik* pada setiap jalurnya sesuai keadaan. Lampu lalulintas (merah, kuning, hijau) akan berubah dengan sendirinya sesuai dengan *delay time*, apabila nyala lampu merah pada setiap jalurnya delay time akan aktif dari hitungan tertinggi sampai 0 (Nol) yang ditampilakan oleh *Dot Matrik*, pada 0 (Nol) maka lampu yang nyala berganti hijau.

KATA PENGANTAR

Alhamdulillah, puji syukur kehadirat-Mu Ya Allah yang telah memberikan rahmat dan hidayah-Nya, sehingga dapat menyelesaikan skripsi yang berjudul **"Perancangan Dan Pembuatan Sistem Pengaturan Lampu Lalulintas 4 Jalur dilengkapi *Keypad* dan *Dot Matriks* Berbasis Mikrokontroller RENESAS R8C/13 TINY dan AT89C51"** ini dengan lancar. Skripsi ini merupakan persyaratan kelulusan Studi di Jurusan Teknik Elektro S-1 Konsentrasi Teknik Elektronika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan peyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

- Allah SWT, penguasa alam semesta.
- Nabi Muhammad SAW, penerang jalan hidupku.
- Ramanda Moh. Hattayani, Ibunda St. Amaniyah dan Dian Utama Sari tercinta serta keluargaku dirumah yang senantiasa memberikan dukungan Do'a dan Moril selama penulis menyelesaikan skripsi ini..
- Bapak Dr. Ir .Abraham Lomi, MSEE selaku Rektor Institut Teknologi Nasional Malang.
- Bapak Ir. Widodo Pudji M., MT. selaku Dosen Pembimbing, atas segala bantuan ilmu, arahan, dorongan, serta dukungannya sehingga penulis dapat menyelesaikan tugas akhir ini.

- Bapak Ir. F. Yudi Limpraptono, MT, selaku Ketua Jurusan Tehnik Elektro Fakultas Tehnik Industri Institut Teknologi Nasional Malang.
- Semua teman – teman yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penyusun telah berusaha semaksimal mungkin dan meyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelasaikan laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penyusun semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, Maret 2007

Penyusun

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengaturan lampu lalulintas pada awalnya menggunakan teknologi rangkaian logika yang berfungsi mengatur atau mengontrol lampu berdasarkan beberapa daftar waktu tetap yang sudah diset terlebih dahulu. Metode ini disebut dengan istilah *fixed time.* Kinerja pengatur lampu lalulintas seperti ini dalam hubungannya dengan fungsi untuk mengatur waktu tunda (*delay time*) kendaraan bermotor sangatlah ditentukan oleh survey yang diperoleh, yang sangat tergantung sekali dengan kondisi statistic lalulintas kendaraan yang ada pada suatu persimpangan atau perempatan jalan tersebut. Kinerja pengontrol lampu lalulintas ini akan dirasakan sangat optimal sekali jika setiap terjadi perubahan kondisi lalulintas dilakukan penyesuaian dengan secara otomatis mengaktifkan lampu lalulintas yang dilengkapi dengan *keypad* sebagai seting waktu jika pada suatu saat kepadatan kendaraan bertambah padat dan *dot matriks* sebagai tampilan hitung waktu mundur.

Lampu lalulintas memegang peranan penting dalam pengaturan kelancaran lalulintas. Sistem pengturan lampu lalu lintas yang baik akan secara otomatis menyesuaikan diri dengan kepadatan arus lalulintas pada jalur yang diatur. Dengan penerapan mikrokontroler renesas R8C/13 TINY maka system

pengaturan lampu lalulintas sangat memungkinkan untuk dapat dilakukan pada setiap keadaan.

1.2 Rumusan Masalah

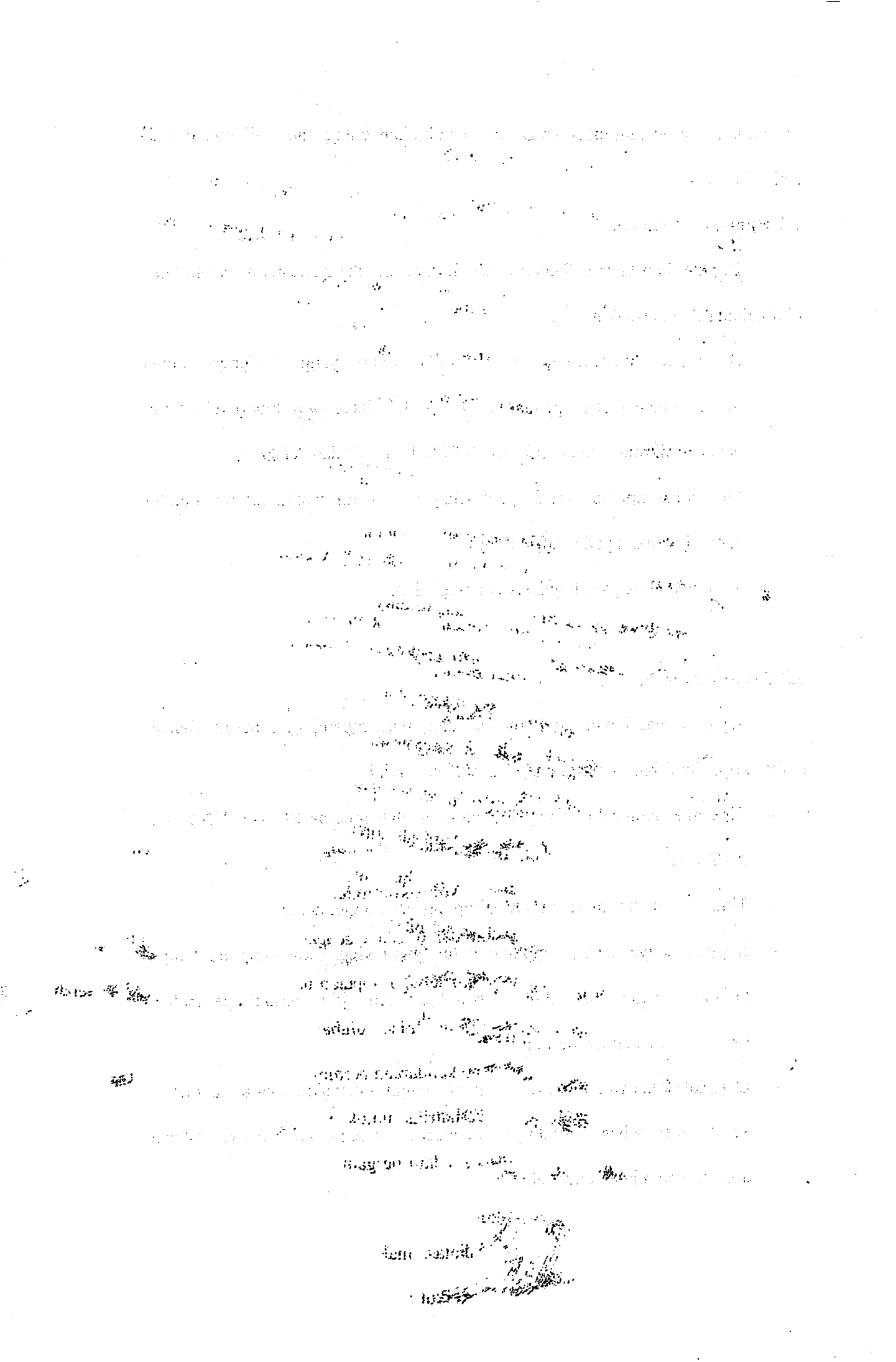
Berdasarkan latar belakang masalah diatas, maka rumusan masalah dapat dijabarkan sebagai berikut :

- Bagaimana merancang dan membuat alat yang berfungsi untuk menampilkan system pengaturan lampu lalulintas yang kinerjanya diatur dengan mikrokontroler RENESAS R8C/13 TINY dan AT89C51.
- Menerima inputan dari *keypad* sebagai pengatur waktu hitung mundur yang dijalankan pada traffic light.
- Membuat perangkat lunak sistem rangkaian.

1.3 Batasan Masalah

Agar permasalahan tidak meluas dan menyimpang dari permasalahan, maka skripsi ini dibatasi hanya pada hal-hal berikut ini :

- Tipe mikrokontroller yang digunakan adalah renesas R8C/13 TINY dan AT89C51.
- Tidak membahas masalah power supplay secara mendetail.
- Keypad hanya sebagai inputan untuk menyeting lama tampilan lampu lalulintas yang bisa diubah sewaktu-waktu jika terjadi perubahan kepadatan kendaraan bermotor.
- Dot Matrik hanya untuk menampilkan penghitung waktu secara mundur.
- Tidak membahas terjadinya kemacetan karena adanya kecelakaan diperempatan jalan tersebut..



1.4 Tujuan

Tujuan dari penulisan skripsi ini adalah untuk merencanakan dan membuat alat yang dapat mengatur lampu lalulintas 4 jalur yang dilengkapi dengan *keypad* sebagai seting atau mengubah waktu mundurnya (*delay time*) sesuai kepadatan lalulintas saat ini dan *dot matriks* untuk menampilkan hitung waktu mundur (*delay time*) yang kinerjanya diatur oleh mikrokontroller.

1.5 Metodologi

Untuk merealisasikan system pengaturan lampu lalulintas, dilalui tahapan-tahapan sebagai berikut :

- Studi literatur, diperlukan untuk mempelajari teori dasar yang berhubungan dengan alat yang dirancang.
- Perencanaan alat, sebelum dilakukan pembuatan alat terlebih dahulu dilakukan penentuan spesifikasi alat yang akan dirancang tersebut, yaitu :
    1. Menggunakan IC Mikrokontroller renesas R8C/13 TINY untuk menjalankan system dan di Slave dengan Mikrokontroler AT89C51.
    2. Menggunakan *keypad* sebagai inputan untuk menyeting lama tampilan lampu lalulintas yang bisa diubah sewaktu-waktu jika terjadi perubahan kepadatan kendaraan bermotor.
    3. Menggunakan dot matriks untuk menampilkan penghitung waktu mundur (*delay time*) dalam pergantian nyala lampu lalulintas (merah, kuning, hijau).

Berdasarkan spesifikasi diatas, maka dilakukan perancangan alat dengan langkah-langkah sebagai berikut :

1. Perencanaan blok diagram sistem
2. Perancangan rangkaian untuk masing-masing blok
3. Perancangan perangkat lunak

1.6 Sistematika Penulisan

Sistematika pembahasan dari skripsi ini terdiri dari pokok pembahasan yang saling berkaitan antara satu dengan lainnya, yaitu :

Bab I Pendahuluan

Pada bab ini dibahas tentang latar belakang permasalahan, rumusan masalah, batasan masalah, sistematika pembahasan dari alat yang direncanakan.

Bab II Teori Penunjang

Pada bab ini dibahas tentang teori-teori yang mendukung dalam perancangan dan pembuatan alat ini yang meliputi rangkaian driver, LCD M1632, dot matriks dan Mikrokontroler.

Bab III Perancangan dan Pembuatan Alat

Pada bab ini dibahas tentang perancangan dan pembuatan keseluruhan sistem perangkat keras (hardware) dan perangkat lunak (software).

Bab IV Pengujian Alat

Pada bab ini dibahas tentang proses serta hasil dari pengujian alat, yang didasarkan oleh pengukuran-pengukuran yang diperlukan.

Bab V Penutup

Pada bab ini akan disampaikan kesimpulan dari perancangan dan pembuatan sistem ini.

BAB II

LANDASAN TEORI

2.1 Pendahuluan

Pada bab ini akan dibahas mengenai landasan teori dari peralatan yang direncanakan, landasan teori ini akan membahas tentang komponen dan peralatan pendukung pada alat yang dibuat.

2.2 Keypad

Keypad merupakan salah satu devais memasukkan suatu data ke LCD (Liquid Cristal Display). Prinsip kerja dari keypad ini adalah menggunakan metode matriks scaning baris dan kolom, pada system scaning sinyal atau tegangan diberikan secara bergantian pada masing-masing pin, keypad ini digunakan sebagai inputan untuk mengganti atau menyeting berapa lama waktu nyala lampu lalulintas pada traffic light.

Kelengkapan yang ada pada IC74C922 adalah:

1. 4 bit counter internal
2. Osilator internal untuk keperluan deteksi tombol yang ditekan
3. Rangkaian anti debouncing
4. Tristate output 4 bit

Dengan penyemat X1, X2, X3, X4 untuk kolom dan Y1, Y2, Y3, Y4 untuk baris dapat dibuat keypad 4x4 sehingga keseluruhan ada 16 tombol. Tabel kebenaran untuk ke16 tombol tersebut terhadap logika keluaran pada penyemat A,B,C dan D dapat dilihat dalam Tabel 2.1.

Tabel 2-1 Kombinasi Keluaran Keypad Matrik 4x4 dari IC 74C922 [8]

Truth Tables

(Pins 0 through 11)

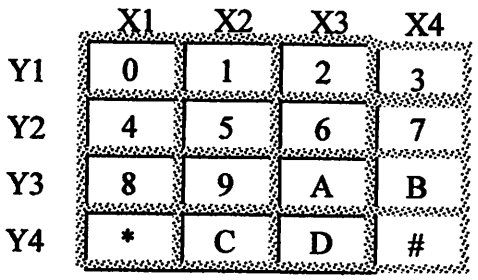
Switch Position	0 Y1, X1	1 Y1, X2	2 Y1, X3	3 Y1, X4	4 Y2, X1	5 Y2, X2	6 Y2, X3	7 Y2, X4	8 Y3, X1	9 Y3, X2	10 Y3, X3	11 Y3, X4
DATA A	0	1	0	1	0	1	0	1	0	1	0	1
DATA B	0	0	1	1	0	0	1	1	0	0	1	1
DATA C	0	0	0	0	1	1	1	1	0	0	0	0
DATA D	0	0	0	0	0	0	0	0	1	1	1	1
DATA E (Note 1)	0	0	0	0	0	0	0	0	0	0	0	0

(Pins 12 through 19)

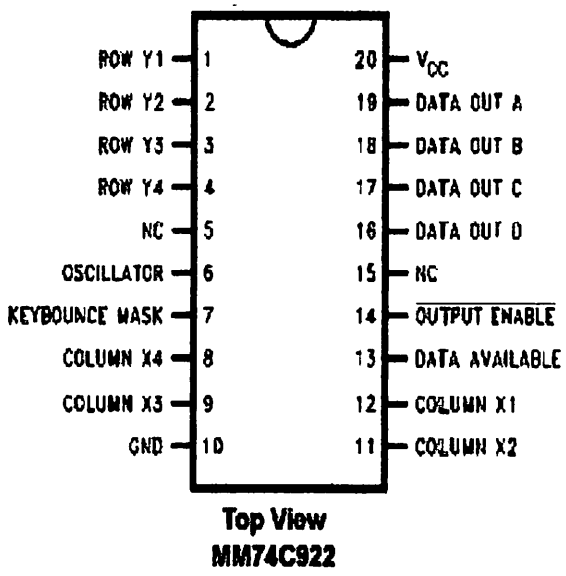
Switch Position	12 Y4, X1	13 Y4, X2	14 Y4, X3	15 Y4, X4	16 Y5(Note 1), X1	17 Y5 (Note 1), X2	18 Y5 (Note 1), X3	19 Y5 (Note 1), X4
DATA A	0	1	0	1	0	1	0	1
DATA B	0	0	1	1	0	0	1	1
DATA C	1	1	1	1	0	0	0	0
DATA D	1	1	1	1	0	0	0	0
DATA E (Note 1)	0	0	0	0	1	1	1	1

Note 1: Only for MM74C922

Posisi tombol terhadap kolom (Xn) dan barisnya (Ym) serta pin yang tersedia ditunjukkan dalam Gambar 2.1.



Gambar 2-1 Posisi tombol terhadap kolom (Xn) dan barisnya (Ym) serta pin yang tersedia n dan m nomor kolom dan baris.



Gambar 2-2 Konfigurasi PIN 74C922 [8]

Keterangan:

a. Pin 1-4 (Row Y1-Row Y4), sebagai input baris.

b. Pin 6 (Oscillator), sebagai Clock input

c. Pin 7 (Keybounce Mask), sebagai kunci input.

d. Pin 8-9, sebagai input dari colom.

e. Pin 10, sebagai ground untuk rangkaian yang dipakai.

f. Pin 11-12, sebagai input colom.

g. Pin 13, sebagai output data available.

h. Pin 14, sebagai output enable.

i. Pin 16-19, sebagai output data.

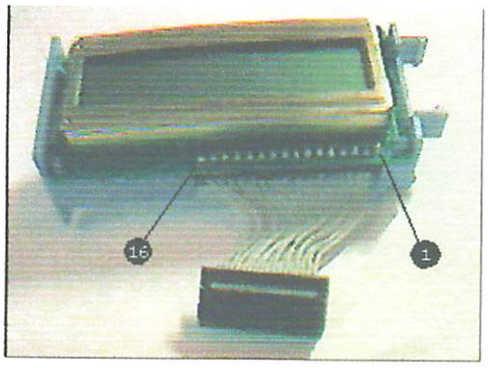
j. Pin 20, VCC

2.3 LCD (Liquid Cristal Display)

LCD display model M1632 buatan Seiko Inc. terdiri dari bagian, yang pertama panel LCD sebagai media penampil informasi dalam bentuk huruf atau angka, dua baris masing-masing baris dapat menampung 16 huruf atau angka.

Bagian kedua merupakan sebuah system yang dibentuk dengan mikrokontroler yang ditempelkan dibalik pada panel LCD, berfungsi mengatur tampilan informasi serta berfungsi mengatur komunikasi dengan mikrokontroler yang memakai tampilan LCD itu. Dengan demikian pemakain M1632 menjadi sederhana.

LCD diperlukan untuk menempilkan informasi dari berapa lama waktu yang diatur atau diseting pada mikrokontroler melalui keypad, LCD M1632 pada skripsi ini digunakan sebagai tampilan saja sehingga LCD hanyan nerima inputan dan kemudian dikimkan ke mikrokontroler. Hal ini dilakukan dengan memberi tegangan positif pada pin no.5 (R/W) untuk menyalakan back light maka pin no.15 (V + BL) haruus diberi tegangan 4.6 Volt dan pin no.16 (V - BL) harus diground.



Gambar 2-3 Pin Out LCD M1632 Standard

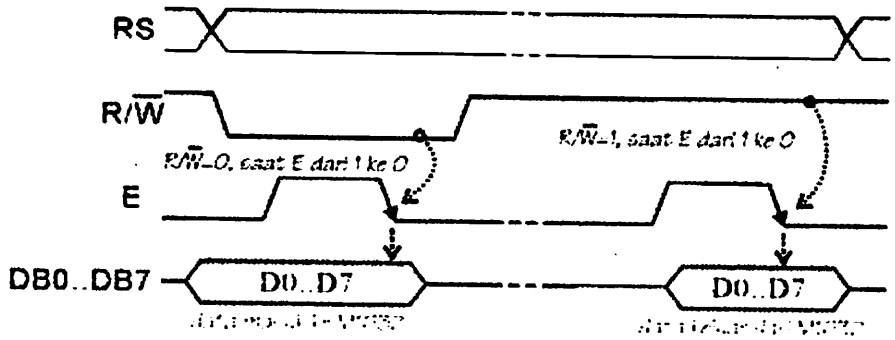
2.3.1. Sinyal interface M1632

Untuk berhubungan dengan mikrokontroler pemakai, M1632 dilengkapi dengan 8 jalur data (**DB0..DB7**) yang dipakai untuk menyalurkan kode ASCII maupun perintah pengatur kerjanya M1632. Selain itu dilengkapi pula dengan **E**, **R/W** dan **RS** seperti layaknya komponen yang kompatibel dengan mikroprosesor. Kombinasi lainya **E** dan **R/W** merupakan sinyal standar pada komponen buatan Motorola. Sebaliknya sinyal-sinyal dari MCS51 merupakan sinyal khas Intel dengan kombinasi sinyal **WR** dan **RD**..

RS, singkatan dari *Register Select,* dipakai untuk membedakan jenis data yang dikirim ke M1632, kalau **RS=0** data yang dikirim adalah perintah untuk mengatur kerja M1632, sebaliknya kalau **RS=1** data yang dikirim adalah kode ASCII yang ditampilkan.

Demikian pula saat pengambilan data, saat **RS=0** data yang diambil dari M1632 merupakan data status yang mewakili aktivitas M1632, dan saat **RS=1** maka data yang diambil merupakan kode ASCII dari data yang ditampilkan.

Proses mengirim/mengambil data ke/dari M1632 digambarkan dalam gambar 2-26 bisa dijabarkan sebagai berikut :



Gambar 2-4 Mengirim/Mengambil Data Ke/Dari M1632 [11]

1. **RS** harus dipersiapkan dulu, untuk menentukan jenis data seperti yang telah dibicarakan di atas.
2. **R/W** di-nol-kan untuk menandakan akan diadakan pengiriman data ke M1632. Data yang akan dikirim disiapkan di **DB0..DB7**, sesaat kemudian sinyal **E** di-satu-kan dan di-nol-kan kembali. Sinyal **E** merupakan sinyal sinkronisasi, saat **E** berubah dari 1 menjadi 0 data di **DB0 .. DB7** diterima oleh M1632.
3. Untuk mengambil data dari M1632 sinyal **R/W** di-satu-kan, menyusul sinyal **E** di-satu-kan. Pada saatu **E** menjadi 1, M1632 akan meletakkan datanya di **DB0 .. DB7**, data ini harus diambil sebelum sinyal **E** di-nol-kan kembali.

2.3.2 Mengatur tampilan M1632

M1632 mempunyai seperangkat perintah untuk mengatur tata kerjanya, perangkat perintah tersebut meliputi perintah untuk menghapus tampilan, meletakkan kembali *cursor* pada barishuruf pertama baris pertama, menghidup/matikan tampilan dan lain sebagainya, semua itu dibahas secara terperinci dalam lembar data M1632.

Setelah diberi sumber daya, ada beberapa langkah persiapan yang harus dikerjakan dulu agar M1632 bisa dipakai, langkah-langkah tersebut antara lain adalah:

1. Tunggu dulu selama 15 mili-detik atau lebih.
2. Kirimkan perintah 30h, artinya trasfer data antar M1632 dan mikrokontroler dilakukan dengan mode 8 bit

3. Tunggu selama 4.1 mili-detik

4. Kirimkan sekali lagi perintah 30h

5. Tunggu lagi selama 100 mikro-detik

Setelah langkah-langkah tersebut di atas M1632 barulah bisa menerima data dan menampilkannya dengan baik. Pada awalnya tampilan akan nampak kacau, dengan demikian perlu segera dikirim perintah menghapus tampilan dan lain sebagainya, sesuai dengan petunjuk yang ada di Lembar Data.

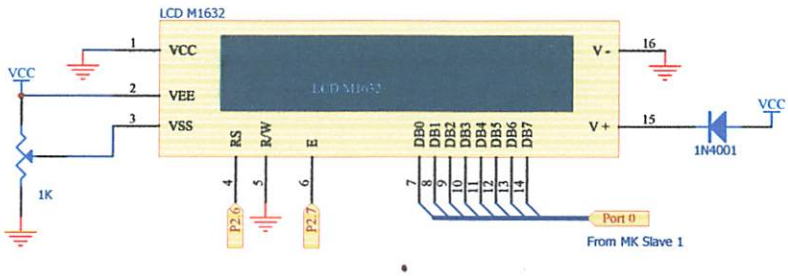
Di atas dipakai Renesas R8C/13Tiny sebagai contoh, meskipun demikian semua yang dibahas di atas sepenuhnya bisa dipakai pada mikrokontroler MCS 51. Dalam pemakaiannya karena berbagai macam alasan, bisa saja sinyal **E**; **RW** dan **RS** tidak disimulasikan di **P3.4**; **P3.5** dan **P3.7**. Hal ini bisa diselesaikan dengan melakukan beberapa penyesuaian, yakni tentukan dulu perubahan rangkaian sesuai dengan keadaan yang ada, dan perubahan rangkaian itu harus di sesuaikan di baris 1 sampai 3 pada potongan program di atas.

M1632 mempunyai 8 jalur data dan memerlukan 3 jalur kontrol, dalam suatu rangkaian yang memakai banyak port dari MC-S51, bisa terjadi kekurangan port untuk menghubungkan MCS51 ke M1632. Jika sampai terjadi hal semacam ini bisa ditempuh hal hal berikut :

1. M1632 dipakai dalam mode data 4 bit, yakni hanya memakai jalur data **D0..D3**

2. Dengan sedikit tambahan rangkaian sinyal **WR** dan **RD** diubah menjadi sinyal **E** dan **R/W** gaya Motorola, sehingga tidak perlu menyediakan port untuk men-simulasikan sinyal-sinyal tersebut.

Berikut adalah gambar rangkaian LCD dengan komponen-komponen pendukung dengan pin-pin yang akan dihubungkan pada mikrokontroller MCS 51 :



Gambar 2-5 Rangkaian LCD M1632

LCD modul M1632 mempunyai 16 pin dengan fungsi sebagai berikut :

Tabel 2-2 Fungsi Pin – Pin LCD [11]

No. PIN	Nama PIN	Fungsi
1	Vss	Terminal Ground
2	Vcc	Tegangan Catu + 5 volt
3	Vee	Mengendalikan kecerahan LCD
4	RS	Sinyal pemilihan register 0 = Tulis 1 = Baca
5	R/W	Sinyal seleksi tulis atau baca 0 = Tulis 1 = Baca
6	E	Sinyal operasi awal yang mengaktifkan data tulis atau baca
7 - 14	DB0 – DB7	Merupakan saluran data berisi perintah data yang akan ditampilkan
15	V + BL	Back Light Supply 4 - 4.2 Volt (Volt)
16	V - BL	Back Ligth Supply 0 (Ground)

LCD M1632 mempunyai spesifikasi sebagai berikut :

1. Memiliki 16 karakter dan dua baris tampilan yang terdiri dari 5 x 7 dot matrik ditambah dengan kursor.
2. Pembangkit karakter ROM untuk 192 jenis karakter.
3. Pembangkit karakter RAM untuk 8 jenis karakter.
4. 80 x 8 display data RAM (max 80 karakter).
5. Isolator didalam modul.

1. Memerlukan catu daya ± volt.

7. Otomatis reset saat catu daya dinyalakan.

Pada LCD juga terdapat instruksi – instruksi sebagai berikut :

- *Display clear* : membersihkan tampilan yang ada pada LCD serta menyimpan, sedangkan kursor kembali ke posisi semula.
- *Cursor home* : hanya membersihkan tampilan dan kursor kembali ke semula.
- *Empty mode Set* : layar beraksi sebagai tampilan tulis.

  S : 1/0 = menggeser layar.

  1/0 : 1 = kursor bergerak ke kanan dan layar bergerak ke kiri.

  1/0 : 0 = kursor bergerak ke kiri dan layar bergerak ke kanan

- *Display On/Off* kontrol.

  D : 1 = layar on

  D : 0 = layar off

  C : 1 = kursor on

  C : 0 = kursor off

  B : 1 = kursor berkedip-kedip

  B : 0 = kursor tidak berkedip – kedip

❖ *Cursor Display Shift*

S/C : 1 = LCD diidentifikasikan sebagai layar

S/C : 0 = LCD diidentifikasikan sebagai kursor

R/L : 1 = menggeser satu spasi ke kanan

R/L : 0 = menggeser satu spasi ke kiri

❖ *Fuction Set*

DL : 1 = panjang data LCD pada 8 bit

DL : 0 = panjang data LCD pada 4 bit

Bit upper ditransfer terlebih dahulu kemudian diikuti dengan 4 bit lower.

N : 1/0 = LCD menggunakan 2 atau 1 baris karakter

P : 1/0 = LCD menggunakan 5 x 10 dot matrik

❖ *CG RAM address set* : menulis alamat RAM ke karakter

❖ *DD RAM address set* : menulis alamat RAM ke tampilan

❖ *BF/address set* : BF = 1/0, LCD dalam keadaan sibuk atau tidak sibuk.

❖ *Data write to CG RAM or DD RAM* : membaca byte dari alamat terakhir RAM yang dipilih.

2.4 Resistor

Pada dasarnya semua bahan memiliki sifat resistif namun beberapa bahan seperti tembaga, perak, emas dan bahan metal umumnya memiliki resistansi yang sangat kecil. Bahan-bahan tersebut menghantar arus listrik dengan baik, sehingga dinamakan konduktor. Kebalikan dari bahan yang konduktif, bahan material seperti karet, gelas, karbon memiliki resistansi yang lebih besar menahan aliran elektron dan disebut sebagai insulator.

Tabel 2-3 Tabel Gelang Warna Resistor

Warna	Nilai	Faktor Pengali	Toleransi
Hitam	0	1	
Coklat	1	10	1%
Merah	2	100	2%
Jingga	3	1.000	
Kuning	4	10.000	
Hijau	5	100.000	
Biru	6	10^6	
Violet	7	10^7	
Abu-abu	8	10^8	
Putih	9	10^9	
Emas	-	0.1	5%
Perak	-	0.01	10%
Tanpa warna	-	-	20%

Resistor adalah komponen dasar elektronika yang digunakan untuk membatasi jumlah arus yang mengalir dalam satu rangkaian. Sesuai dengan namanya resistor bersifat resistif dan umumnya terbuat dari bahan karbon. Dari hukum Ohm diketahui, resistansi berbanding terbalik dengan jumlah arus yang mengalir melaluinya. Satuan resistansi dari suatu resistor disebut Ohm atau dilambangkan dengan simbol W (Omega).

Tipe resistor yang umum adalah berbentuk tabung dengan dua kaki tembaga di kiri dan kanan. Pada badannya terdapat lingkaran membentuk gelang kode warna untuk memudahkan pemakai mengenali besar resistansi tanpa mengukur besarnya dengan Ohm meter. Kode warna tersebut adalah standar manufaktur yang dikeluarkan oleh *EIA (Electronic Industries Association)* seperti yang ditunjukkan pada tabel 2-3.

Resistansi dibaca dari warna gelang yang paling depan ke arah gelang toleransi berwarna coklat, merah, emas atau perak. Biasanya warna gelang toleransi ini berada pada badan resistor yang paling pojok atau juga dengan lebar yang lebih menonjol, sedangkan warna gelang yang pertama agak sedikit ke dalam. Dengan demikian pemakai sudah langsung mengetahui berapa toleransi dari resistor tersebut. Kalau kita telah bisa menentukan mana gelang yang pertama selanjutnya adalah membaca nilai resistansinya.

Jumlah gelang yang melingkar pada resistor umumnya sesuai dengan besar toleransinya. Biasanya resistor dengan toleransi 5%, 10% atau 20% memiliki 3 gelang (tidak termasuk gelang toleransi). Tetapi resistor dengan toleransi 1% atau 2% (toleransi kecil) memiliki 4 gelang (tidak termasuk gelang toleransi). Gelang

pertama dan seterusnya berturut-turut menunjukkan besar nilai satuan, dan gelang terakhir adalah faktor pengalinya.

Misalnya resistor dengan gelang kuning, violet, merah dan emas. Gelang berwarna emas adalah gelang toleransi. Dengan demikian urutan warna gelang resitor ini adalah, gelang pertama berwarna kuning, gelang kedua berwana violet dan gelang ke tiga berwarna merah. Gelang ke empat tentu saja yang berwarna emas dan ini adalah gelang toleransi. Dari tabel 2-3 diketahui jika gelang toleransi berwarna emas, berarti resitor ini memiliki toleransi 5%. Nilai resistansisnya dihitung sesuai dengan urutan warnanya. Pertama yang dilakukan adalah menentukan nilai satuan dari resistor ini. Karena resitor ini resistor 5% (yang biasanya memiliki tiga gelang selain gelang toleransi), maka nilai satuannya ditentukan oleh gelang pertama dan gelang kedua. Masih dari tabel 2-3 diketahui gelang kuning nilainya = 4 dan gelang violet nilainya = 7. Jadi gelang pertama dan kedua atau kuning dan violet berurutan, nilai satuannya adalah 47. Gelang ketiga adalah faktor pengali, dan jika warna gelangnya merah berarti faktor pengalinya adalah 100. Sehingga dengan ini diketahui nilai resistansi resistor tersebut adalah nilai satuan x faktor pengali atau 47 x 100 = 4.7K Ohm dan toleransinya adalah 5%.

Spesifikasi lain yang perlu diperhatikan dalam memilih resitor pada suatu rancangan selain besar resistansi adalah besar watt-nya. Karena resistor bekerja dengan dialiri arus listrik, maka akan terjadi disipasi daya berupa panas sebesar $W=I^2R$ watt. Semakin besar ukuran fisik suatu resistor bisa menunjukkan semakin besar kemampuan disipasi daya resistor tersebut.

Umumnya di pasar tersedia ukuran 1/8, 1/4, 1, 2, 5, 10 dan 20 watt. Resistor yang memiliki disipasi daya 5, 10 dan 20 watt umumnya berbentuk kubik memanjang persegi empat berwarna putih, namun ada juga yang berbentuk silinder. Tetapi biasanya untuk resistor ukuran jumbo ini nilai resistansi dicetak langsung dibadannya, misalnya 100W 5W.

2.5. Transistor Bipolar

Prinsip kerja transistor adalah arus bias base-emiter yang kecil mengatur besar arus kolektor-emiter.

Bagian penting berikutnya adalah bagaimana caranya memberi arus bias yang tepat sehingga transistor dapat bekerja optimal.

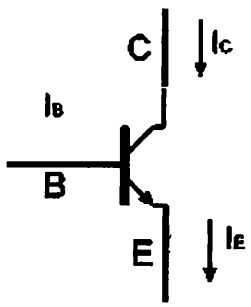
2.5.1. Arus bias

Ada tiga cara yang umum untuk memberi arus bias pada transistor, yaitu rangkaian *CE (Common Emitter), CC (Common Collector)* dan *CB (Common Base).* Namun dalam hal ini akan lebih detail dijelaskan bias transistor rangkaian CE. Dengan menganalisa rangkaian CE akan dapat diketahui beberapa parameter penting dan berguna terutama untuk memilih transistor yang tepat untuk berbagai aplikasi.

2.5.2. Arus *Emiter*

Dari hukum Kirchhoff diketahui bahwa jumlah arus yang masuk ke satu titik akan sama jumlahnya dengan arus yang keluar. Jika teorema tersebut diaplikasikan pada transistor, maka hukum itu menjelaskan hubungan :

$$I_E = I_C + I_B \quad \text{.....(2-2)}$$



Gambar 2-6 Arus Emitor

Persamanaan (2-2) tersebut mengatakan arus *emiter* I_E adalah jumlah dari arus kolektor I_C dengan arus base I_B. Karena arus I_B sangat kecil sekali atau disebutkan $I_B << I_C$, maka dapat dinyatakan :

$$I_E = I_C \quad \text{.....(2-3)}$$

2.5.3. *Alpha* (a)

Pada tabel data transistor (*databook*) sering dijumpai spesifikasi $_{dc}$ *(alpha dc)* yang tidak lain adalah :

$$a_{dc} = I_C/I_E \quad \text{.....(2-4)}$$

Definisinya adalah perbandingan arus kolektor terhadap arus emitor. Karena besar arus kolektor umumnya hampir sama dengan besar arus emiter maka idealnya besar a_{dc} adalah = 1 (satu). Namun umumnya transistor yang ada memiliki a_{dc} kurang lebih antara 0.95 sampai 0.99.

2.5.4. *Beta* (b)

Beta didefenisikan sebagai besar perbandingan antara arus kolektor dengan arus *base*.

$$b = I_C/I_B \quad \text{.....(2-5)}$$

Dengan kata lain, b adalah parameter yang menunjukkan kemampuan penguatan arus *(current gain)* dari suatu transistor. Parameter ini ada tertera di *data book* transistor dan sangat membantu para perancang rangkaian elektronika dalam merencanakan rangkaiannya.

Misalnya jika suatu transistor diketahui besar b=250 dan diinginkan arus kolektor sebesar 10 mA, maka berapakah arus bias base yang diperlukan. Maka :

$$\mathbf{I_B = I_C/b = 10mA/250 = 40\ uA}$$

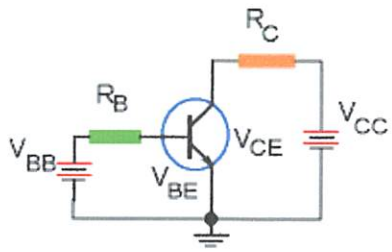
Arus yang terjadi pada kolektor transistor yang memiliki b = 200 jika diberi arus bias base sebesar 0.1 mA adalah :

$$\mathbf{I_C = b\ I_B = 200 \times 0.1mA = 20\ mA}$$

Dari rumusan ini lebih terlihat definisi penguatan arus transistor, yaitu sekali lagi, arus *base* yang kecil menjadi arus kolektor yang lebih besar.

2.5.5. *Common Emitter* (CE)

Rangkaian CE adalah rangkain yang paling sering digunakan untuk berbagai aplikasi yang mengunakan transistor. Dinamakan rangkaian CE, sebab titik *ground* atau titik tegangan 0 volt dihubungkan pada titik *emiter*.



Gambar 2-7 Rangkaian CE

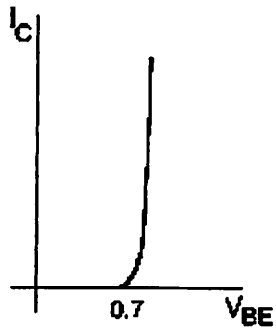
Sekilas tentang notasi, ada beberapa notasi yang sering digunakan untuk menunjukkan besar tegangan pada suatu titik maupun antar titik. Notasi dengan 1 *subscript* adalah untuk menunjukkan besar tegangan pada satu titik, misalnya V_C = tegangan kolektor, V_B = tegangan *base* dan V_E = tegangan *emiter*.

Ada juga notasi dengan 2 *subscript* yang dipakai untuk menunjukkan besar tegangan antar 2 titik, yang disebut juga dengan tegangan jepit. Diantaranya adalah :

- V_{CE} = tegangan jepit kolektor-emitor.
- V_{BE} = tegangan jepit *base*-emitor.
- V_{CB} = tegangan jepit kolektor-*base*.

Notasi seperti V_{BB}, V_{CC}, V_{EE} berturut-turut adalah besar sumber tegangan yang masuk ke titik base, kolektor dan emitor.

2.5.6. Kurva *Base*



Gambar 2-8 Kurva I_B -V_{BE}

Hubungan antara IB dan VBE tentu saja akan berupa kurva dioda. Karena memang telah diketahui bahwa junction *base-emitor* tidak lain adalah sebuah dioda. Jika hukum Ohm diterapkan pada loop base diketahui adalah :

$$I_B = (V_{BB} - V_{BE}) / R_B \quad \text{.....(2-6)}$$

V_{BE} adalah tegangan jepit dioda *junction base-emitor*. Arus hanya akan mengalir jika tegangan antara base-emitor lebih besar dari V_{BE}. Sehingga arus I_B mulai aktif mengalir pada saat nilai V_{BE} tertentu. Besar V_{BE} umumnya tercantum di dalam *databook*. Tetapi untuk penyerdehanaan umumnya diketahui V_{BE} = 0.7 volt untuk transistor silikon dan V_{BE} = 0.3 volt untuk transistor germanium.

Sampai disini akan sangat mudah mengetahui arus I_B dan arus I_C dari rangkaian berikut ini, jika diketahui besar b = 200. Katakanlah yang digunakan adalah transistor yang dibuat dari bahan silikon.

$$I_B = (V_{BB} - V_{BE}) / R_B$$

$$= (2V - 0.7V) / 100\ K$$

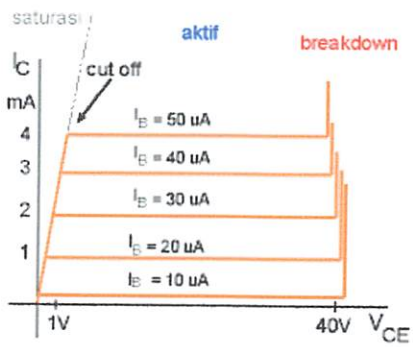
$$= 13\ uA$$

Dengan b = 200, maka arus kolektor adalah :

$$I_C = b\ I_B = 200 \times 13uA = 2.6\ mA$$

2.5.7. Kurva Kolektor

Sekarang sudah diketahui konsep arus base dan arus kolektor. Satu hal lain yang menarik adalah bagaimana hubungan antara arus base I_B, arus kolektor I_C dan tegangan kolektor-emiter V_{CE}. Pada grafik berikut telah diplot beberapa kurva kolektor arus I_C terhadap V_{CE} dimana arus I_B dibuat konstan.



Gambar 2-9 Kurva Kolektor

Dari kurva ini terlihat ada beberapa *region* yang menunjukkan daerah kerja transistor. Pertama adalah daerah saturasi, lalu daerah *cut-off*, kemudian daerah aktif dan seterusnya daerah *breakdown*.

2.5.8. Daerah Aktif

Daerah kerja transistor yang normal adalah pada daerah aktif, dimana arus I_C konstan terhadap berapapun nilai V_{CE}. Dari kurva ini diperlihatkan bahwa arus I_C hanya tergantung dari besar arus I_B. Daerah kerja ini biasa juga disebut daerah linier (*linear region*).

Jika hukum Kirchhoff mengenai tegangan dan arus diterapkan pada loop kolektor (rangkaian CE), maka dapat diperoleh hubungan :

$$V_{CE} = V_{CC} - I_C R_C \quad \text{.....(2-7)}$$

Dapat dihitung disipasi daya transistor adalah :

$$P_D = V_{CE}\, I_C \quad \text{.....(2-8)}$$

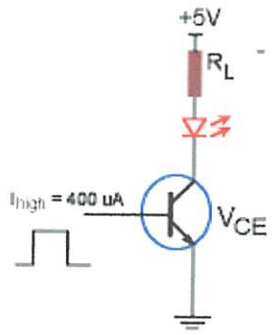
Rumus ini mengatakan jumlah disipasi daya transistor adalah tegangan kolektor-emitor dikali jumlah arus yang melewatinya. Disipasi daya ini berupa panas yang menyebabkan naiknya temperatur transistor. Umumnya untuk transistor power sangat perlu untuk mengetahui spesifikasi P_{Dmax}. Spesifikasi ini menunjukkan temperatur kerja maksimum yang diperbolehkan agar transistor masih bekerja normal. Sebab jika transistor bekerja melebihi kapasitas daya P_{Dmax}, maka transistor dapat rusak atau terbakar.

2.5.9. Daerah Saturasi

Daerah saturasi adalah mulai dari V_{CE} = 0 volt sampai kira-kira 0.7 volt (transistor silikon), yaitu akibat dari efek dioda kolektor-*base* yang mana tegangan V_{CE} belum mencukupi untuk dapat menyebabkan aliran elektron.

2.5.10. Daerah *Cut-Off*

Jika kemudian tegangan V_{CC} dinaikkan perlahan-lahan, sampai tegangan VCE tertentu tiba-tiba arus IC mulai konstan. Pada saat perubahan ini, daerah kerja transistor berada pada daerah *cut-off* yaitu dari keadaan saturasi *(OFF)* lalu menjadi aktif *(ON)*. Perubahan ini dipakai pada system digital yang hanya mengenal angka biner 1 dan 0 yang tidak lain dapat direpresentasikan oleh status transistor *OFF* dan *ON*.



Gambar 2-10 Rangkaian *Driver* LED

Misalkan pada rangkaian *driver* LED di atas, transistor yang digunakan adalah transistor dengan b = 50. Penyalaan LED diatur oleh sebuah gerbang logika (*logic gate)* dengan arus *output high* = 400 uA dan diketahui tegangan

forward LED, V_{LED} = 2.4 volt. Kemudian pertanyaannya adalah, berapakah seharusnya resistansi R_L yang dipakai.

$$\mathbf{I_C = bI_B = 50 \times 400\ uA = 20\ mA}$$

Arus sebesar ini cukup untuk menyalakan LED pada saat transistor *cut-off*. Tegangan VCE pada saat *cut-off* idealnya = 0, dan aproksimasi ini sudah cukup untuk rangkaian ini.

$$R_L = (V_{CC} - V_{LED} - V_{CE}) / I_C$$

$$= (5 - 2.4 - 0)V / 20\ mA$$

$$= 2.6V / 20\ mA$$

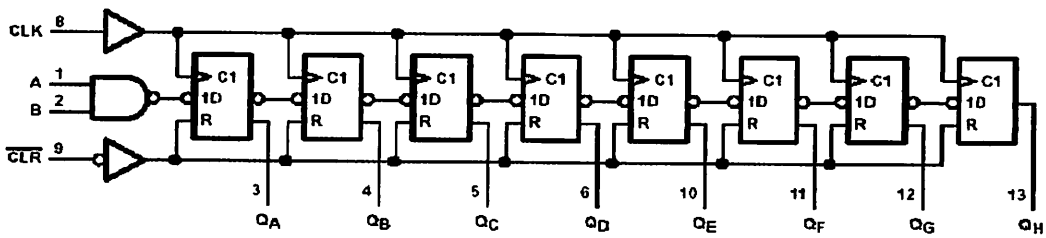
$$= 130\ Ohm$$

2.5.11. Daerah *Breakdown*

Dari kurva kolektor, terlihat jika tegangan V_{CE} lebih dari 40V, arus I_C menanjak naik dengan cepat. Transistor pada daerah ini disebut berada pada daerah *breakdown*. Seharusnya transistor tidak boleh bekerja pada daerah ini, karena akan dapat merusak transistor tersebut. Untuk berbagai jenis transistor nilai tegangan V_{CEmax} yang diperbolehkan sebelum *breakdown* bervariasi. V_{CEmax} pada *data book* transistor selalu dicantumkan juga.

2.6. Register Geser 8 bit *Serial In Pararel Out* (SIPO)

Bagian ini akan merinci salah satu dari sekian banyak register geser, yaitu register yang masukannya seri dan keluarannya paralel 8 bit. IC yang digunakan adalah IC SN74HC595 yang merupakan register 8 bit yang terpicu sentuh dengan masukan serial. Semua keluaran paralel tersedia atas setiap *flip-flop D internal.* Diagram rinci tersedia dalam gambar 2-13, memperlihatkan penggunaan delapan *flip-flop D internal*, masing-masing dengan keluaran data paralelnya (Q_A sampai Q_B).



Gambar 2-11 Diagram Detail Register Geser 8 Bit SN74HC595

IC SN74HC595 terlihat memiliki sebuah masukan serial. Data dimasukkan secara serial melalui salah satu dari dua masukan (A dan B). Dalam Gambar 2-13 kedua masukan data (A dan B) di NAND-kan. Kedua masukan ini bisa digabung menjadi satu masukan atau bisa juga salah satu ditentukan sebagai logika tinggi, sedangkan yang satu lagi untuk pemasukan data.

Masukan pengatur ulang utama *Clear* untuk IC SN74HC595 merupakan *input* aktif rendah. Dalam tabel kebenaran (tabel 2-3) memperlihatkan bahwa bila diaktifkan, masukan *Clear* akan mematikan semua input yang lainnya, dan mengembalikan semua *flip-flop* ke 0. IC SN74HC595 menggeser data suatu tempat ke kanan dalam setiap peralihan rendah ke tinggi dalam masukan *Clock*.

Tabel 2-4 Tabel Kebenaran IC 74HC595 [7]

INPUT					OUTPUT		FUNCTION
SH CP	ST CP	$\overline{OE}$	$\overline{MR}$	DS	Q7'	Qn	
X	X	-	L	X	L	n.c	a LOW on $\overline{MR}$ only affects the shift registers
X		-	L	X	L	-	empty shift register loaded into storage register
X	X	H	L	X	L	Z	shift register clear; parallel outputs in high-impedance OFF-state
	X	-	H	l	Qn	n.c	logic high level shifted into shift register stage 0; contents of all shift register stages shifted through, e.g. previous state of stage 6 (internal Qn) appears on the serial output (Q7')
X		-	H	X	n.c.	Qn	contents of shift register stages (internal Qn) are transferred to the storage register and parallel output stages
		-	H	X	Qn	Qn	contents of shift register shifted through previous contents of the shift register is transferred to the storage register and the parallel output stages

Note

1. H = HIGH voltage level;
   L = LOW voltage level;
   = LOW-to-HIGH transition;
   = HIGH-to-LOW transition;
   Z = high-impedance OFF-state
   n.c = no change
   X = don't care.

Dalam register geser *serial-in paralel out* (SIPO), data disajikan satu bit dalam satu saat lalu digeserkan masuk dalam setiap pulsa *clock*. Sesudah seperangkat pulsa *clock* lengkap, register menjadi penuh dan kandungannya dapat dibaca diterminal Q atau dikeluarkan melalui seperangkat saluran paralel.

2.7. *Light Emitting Diode* (LED)

Dioda pemancar cahaya *light-emitting diode* (LED) merupakan sebuah dioda yang dapat memancarkan cahaya dengan memanfaatkan adanya penurunan elektron dari *level* konduksi ke *level valensi* yang menimbulkan pelepasan energi dalam bentuk panas atau cahaya. Sebuah elektron yang bergerak melintasi sembarang pertemuan PN akan bergerak ke daerah lubang *(hole*). Hal ini akan menyebabkan sebuah elektron konduksi yang berdekatan turun ke level valensinya, sambil meradiasikan energi. Dalam dioda dan transistor biasa, yang terbuat dari germanium, silikon, atau galium arsenida.

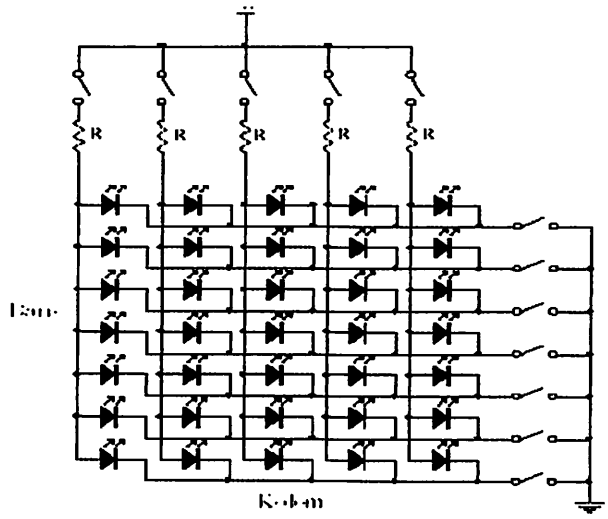
Arus maju berkisar antara 10 mA sampai dengan 20 mA untuk kecerahan maksimum. Sebuah resistor akan selalu dihubungkan seri dengan LED untuk membatasi arus sesuai dengan yang dikehendaki.

LED menggantikan lampu pilot yang dulu digunakan sebagai penunjuk visual dalam peralatan. Sebuah LED ditambah sebuah resistor dapat digunakan dalam sembarang tegangan dari 2 sampai 10 volt. LED menggunakan arus yang jauh lebih kecil daripada lampu pilot.

2.7.1. LED Sebagai *Dot Matriks*

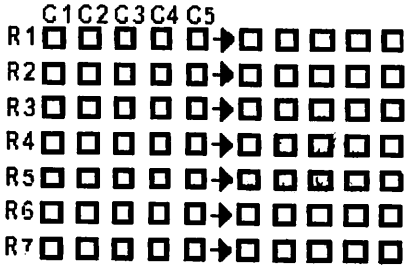
Piranti *dot matriks* mempunyai sejumlah besar sumber cahaya yang berbentuk seperti titik. Contoh umum adalah matriks LED 5x8 yang terlihat dalam gambar 2-14.. Untuk menghidupkan sebuah LED dalam matriks ini, harus menerapkan tegangan dalam anoda-nya dan men-*ground*-kan katoda-nya. Dengan menerapkan tegangan lebih dari sebuah kolom dan men-ground-kan lebih dari satu baris, maka dapat

memperagakan setiap angka desimal, setiap huruf abjad, serta berbagai lambang-lambang lain.



Gambar 2-12 Skema LED Dot Matriks 5x7

Tampilan matriks titik *(dot matriks)* terdiri atas sejumlah LED yang disusun dalam baris dan kolom. Susunan yang paling sering digunakan adalah matriks 5x7, yaitu lima kolom dan tujuh baris, seperti dalam gambar 2-15.



Gambar 2-13 Dot Matriks 5x7

Matriks 5×7 dapat digunakan untuk menyajikan karakter *alfanumerik* yang lengkap. Proses pembangkitan karakter antara lain melibatkan proses *scanning* baris (atau kolom), memilih LED yang tepat dalam baris (kolom) dan menyalakannya. Proses

yang ini diulang untuk baris atau kolom berikutnya. Setelah semua baris atau kolom dipilih dengan urutan tertentu, proses diatas diulang mulai dari baris paling atas (atau kolom pertama).

Jika frekuensi *scanning*-nya cukup cepat (sekitar 50 Hz) saja, akan diperoleh karakter bebas kedip. Jika matriks di-*scan* dari kiri ke kanan, kolom demi kolom, disebut vertical *scanning*. Jika lakukan baris demi baris, disebut horisontal *scanning*.

2.7.2. *Dekode Matriks* LED

Untuk menyalakan huruf *alfanumerik* dengan pembacaan dot matriks, LED yang dibutuhkan tidak dinyalakan serentak (lihat gambar 2-15). Bila proses ini diulang dengan cepat suatu *alfanumerik* akan ditampilkan tanpa kedipan.

Rangkaian yang men-*driver* matriks LED cukup rumit, karena harus melakukan *scan* baris-baris horisontal dan pada saat yang sama memasang tegangan pada kolom yang sesuai. Secara umum pencacah lingkar *(ring counter)* melakukan *scan* dalam baris horisontal, sedangkan memori berisi data kolom LED yang menyala memberikan tegangan dalam kolom-kolom vertikal

2.8. Mikrokontroler AT89C51

Mikrokontroller bisa dipandang sebagai sebuah mini komputer yang terintegrasi dalam sebuah chip. Didalam satu chip mikrokontroller sudah terdapat bagian-bagian seperti dalam sebuah komputer. Bagian-bagian itu antara lain ; ALU (*Arithmetic Logic Unit*), PC (*Program Counter*), SP (*Stack Pointer*), Register, ROM (*Read Only Memory*), RAM (*Random Acces Memory*), Paralel I/O, Serial I/O, *Counter* dan sebuah rangkaian *Clock*.

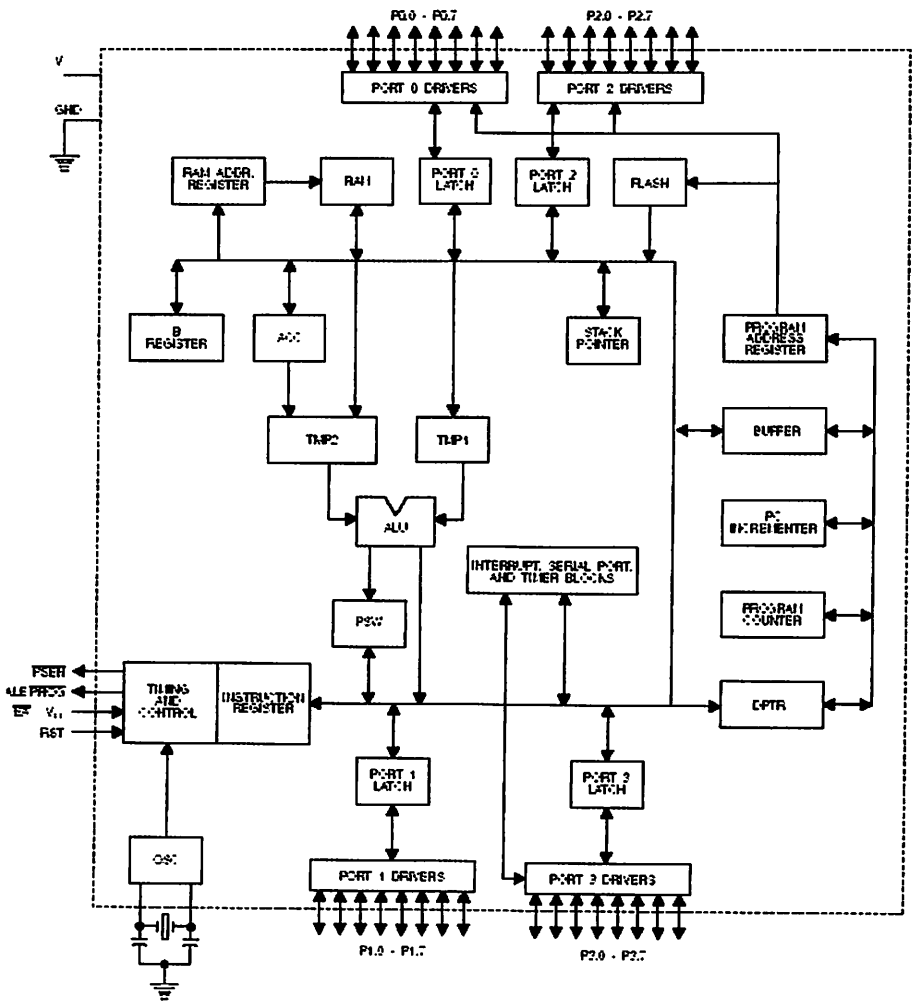
Seperti sebuah mikroprosessor, mikrokontroller adalah sebuah perangkat serbaguna, yang fungsi kerjanya dapat ditentukan melalui sebuah perangkat lunak yang mendeskripsikan sebuah sistem yang diinginkan.

Pada saat ini terdapat banyak keluarga mikrokontroller salah satunya adalah keluarga MCS51. Salah satu tipe mikrokontroller yang termasuk dalam keluarga MCS51 adalah AT89C51 buatan Atmel.

AT89C51 adalah mikrokontroler keluaran atmel dengan 4K byte Flash PEROM (*Programmable and Erasable Read Only Memory*), AT89C51 merupakan memori dengan teknologi nonvolatile memori, artinya isi memori tersebut dapat diisi ulang ataupun dihapus berulang kali.

Memori ini biasa digunakan untuk menyimpan instruksi (Perintah) berstandar MCS – 51 code sehingga memungkinkan mikrokontroler ini untuk bekerja dalam mode *Single Chip Operation* (Mode Operasi Keping Tunggal) yang tidak memerlukan *Eksternal Memori* (Memori luar) untuk menyimpan source code tersebut.

2.8.1. Arsitektur AT89C51



Gambar 2-14 Blok Diagram AT89C51 [6]

AT89C51 adalah mikrokontroler keluaran atmel dengan 4K byte Flash PEROM (*Programmable and Erasable Read Only Memory*), AT89C51 merupakan memori dengan teknologi nonvolatile memori, artinya isi memori tersebut dapat diisi ulang ataupun dihapus berulang kali.

Memori ini biasa digunakan untuk menyimpan instruksi (Perintah) berstandar MCS – 51 code sehingga memungkinkan mikrokontroler ini untuk bekerja dalam mode *Single Chip Operation* (Mode Operasi Keping Tunggal) yang

tidak memerlukan *Eksternal Memori* (Memori luar) untuk menyimpan source code tersebut.

IC ATMEL AT89C51 menyediakan standart berikut:

⇒ 4K Bytes memori yang dapat diprogram ulang

⇒ 128 Bytes internal RAM

⇒ 32 jalur I/O (Input dan Output) yang dapat diprogram

⇒ Sepasang 16 bit Timer dan Counter

⇒ Dual data Pointer (DPTR)

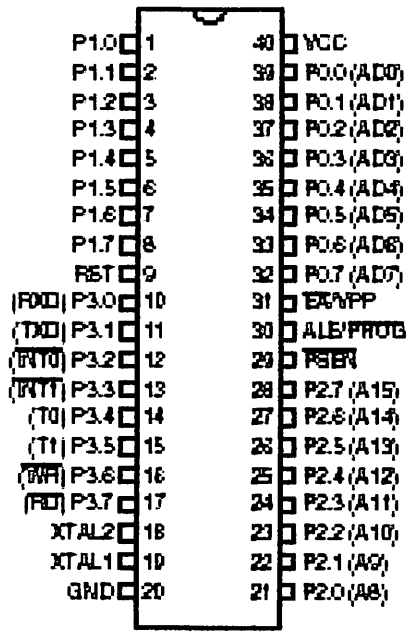
⇒ Watchdog Timer

⇒ ISP Port

⇒ Mendukung serial Port secara penuh

⇒ Waktu Pemrograman yang singkat

Sebagai tambahan AT89C51 dirancang menggunakan logika yang statis untuk mode pengoperasian yang menuju ke frekwensi dasar dan pendukungan terhadap dua Software, serta dapat memilih model Power Savingnya. Mode idle akan berhenti ketika CPU sedang menjalankan RAM, Timer/Counter, Serial Port dan Interrupt System untuk terus melanjutkan fungsinya. Model power down akan menyimpan isi dari RAM tapi akan memberhentikan ossilator dan akan menghentikan semua chip lain yang sedang berfungsi sampai terdapat adanya gangguan dari luar atau hardware di reset.



Gambar 2-15 Pin – Pin AT89C51 [6]

2.8.2. Pin Deskripsi

VCC : Power Supply

GND : Ground

Port 0 : Port 0 berfungsi sebagai 8 bit I/O bertipe *open drain bi-directional.* Sebagai port keluaran masing – masing pin dapat menyerap arus sebesar 8 masukan TTL (sekitar 3,8 mA). Ketika diberikan logika '1' pada pin port 0 ini maka pin – pin port 0 ini akan dapat digunakan sebagai inputan berimpedansi tinggi.

Port 0 juga dapat dikonfigurasikan pada sebagai bus alamat/data selama proses pengaksesan data memori dan program eksternal. Jika digunakan dalam mode ini port 0 memiliki internal Pull Up.

Port 0 juga menerima kode – kode data yang diberkan padanya selama proses pemrograman dan memberikan kode – kode selama proses verifikasi program yang telah tersimpan didalam memori. Dalam hal ini dibutuhkan eksternal Pull Up selama proses verifikasi program.

Port 1 : Port 1 berfungsi sebagai 8 bit I/O Bi-directional yang dilengkapi dengan internal Pull Up. Ketika diberikan logika '1' pin ini akan di Pull Up secara internal sehingga dapat digunakan sebagai input. Sebagai inputan jika pin – pin ini dihubungkan ke ground maka masing – masing pin ini dapat menghantarkan arus karena di Pull High secara internal. Port 1 juga menerima *Low Order Address Bytes* selama melakukan verifikasi program.

Port 2 : Port 2 berfungsi sebagai 8 bit I/O Bi-directional yang dilengkapi dengan internal Pull Up Penyangga keluaran port 2 dapat memberikan atau menyerap arus empat masukan TTL (sekitar 1,6 mA)

Jika diberikan logika '1' pada pin – pin port 2, maka masing – masing pin akan di Pull High secara internal sehingga dapat digunakan sebagai inputan. Sebagai inputan jika pin – pin port 2 dihubungkan ke ground (di Pull Low), maka , masing – masing pin dapat menghantarkan arus karena di Pull High secara internal.

Port 2 akan memberikan byte alamat bagian tinggi (High Byte) selama pengambilan instruksi dari memori program eksternal dan selama pengaksesan memori data eksternal yang menggunakan

perintah dengan alamat 16 bit (misalkan **MOVX@DPTR**). Dalam aplikasi ini , jika ingin mengirimkan '1', maka digunakan Pull Up internal yang sudah disediakan. Selama pengaksesan memori data eksternal yang menggunaka perintah 8 bit (misalkan **MOVX@RI**), port 2 akan mengirimkan isi dari SFR P2 (*Special Function Register Port 2*). Port 2 juga menerima alamat bagian tinggi (High Order Address) selama pemrograman dan verifikasi memori.

Port 3 : Port 3 sebagai 8 bit I/O Bi-directional yang dilengkapi dengan Pull Up Internal. Penyangga keluaran port 3 dapat memberikan atau menyerap arus empat masukan TTL (sekitar 1,6 mA).

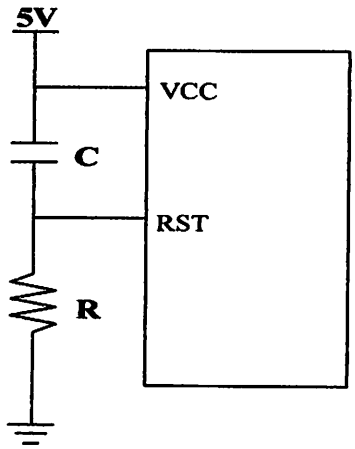
Jika diberikan logika '1' pada pin pin port 3, maka masing – masing pin akan di Pull High oleh Pull Up internal sehingga dapat digunakan sebagai inputan. Sebagai inputan, jika pin – pin port 3 dihubungkan ke ground, maka masing – masing kaki akan memberikan arus karena di Pull High secara internal.

Tabel 2 – 5 Fungsi – Fungsi Alternatif Port 3 [6]

Port Pin	Fungsi Alternatif
P3.0	RXD (Serial Input Port)
P3.1	TXD (Serial Output Port)
P3.2	INT0 (Eksternal Interrupt 0)
P3.3	$\overline{\text{INT1}}$ (Eksternal Interrupt 1)
P3.4	T0 (Timer 0 Eksternal Input)
P3.5	T1 (Timer 1 Eksternal Input)

P3.6	WR (Eksternal Data Memory Write Strobe)
P3.7	RD (Eksternal Data Memory Read Strobe)

Reset :Inputan Reset akan memberikan logika High '1' pada pin ini dengan jangka waktu yang ditentukan oleh lamanya pengosongan data muatan kapasitor. Jangka waktu minimal adalah 2 siklus mesin (24 periode frekwensi clock) ditambah waktu start On Osilator.



Gambar 2–16 Rangkaian Power On Reset

ALE/$\overline{\text{PROG}}$: Keluaran ALE (*Address Latch Enable*) menghasilkan pulsa – pulsa untuk menutup byte rendah (*Low Byte*) alamat selama mengakses memori eksternal. Pin ini juga berfungsi sebagai inputan pulsa program (*The Program Pulse Input*) atau PROG selama melakukan Flash Program. Pada operasi normal, ALE akan berpulsa dengan pewaktuan (*Timing*) atau pendetakan (*Clocking*) rangkaian eksternal. Sebagai catatan ada sebuah pulsa yang dilewati selama pengaksesan memori data eksternal. Jika dikehendaki operasi ALE dapat di nonaktifkan dengan cara mengatur bit 0 dari SFR (*Special*

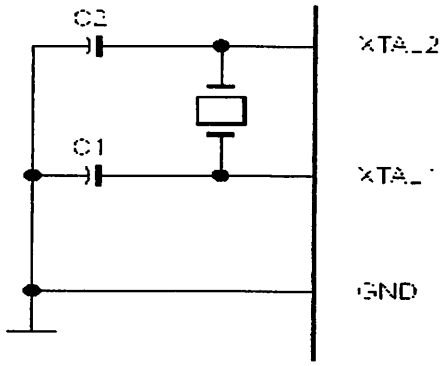
Function Register) lokasi 8Eh. Jika diberi logika '1' ALE hanya akan aktif selama menemui instruksi **MOVX** atau **MOVC**. Selain itu, pin ini secara perlahan akan di Pull High. Mematikan bit ALE tidak akan ada efeknya jika mikrokontroller mengeksekusi program secara eksternal.

$\overline{\text{PSEN}}$: **$\overline{\text{PSEN}}$** (*Program Store Enable*) merupakan sinyal baca untuk memori program eksternal. Ketika mikrokontroller AT89C51 menjalankan kode dari program eksternal, **PSEN** akan diaktifkan sebanyak 2 kali per siklusnya, kecuali dua aktivasi **PSEN** dilompati (Diabaikan) saat mengakses memori data eksternal.

$\overline{\text{EA}}$/VPP : **$\overline{\text{EA}}$/VPP** (*External Access Enable*). **$\overline{\text{EA}}$** harus selalu dihubungkan ke Ground karena digunakan untuk mengakses eksternal memori dengan lokasi 0000H sampai FFFFH. Catatan sekalipun bit '1' sudah terkunci dan terprogram, maka EA akan terkunci pada reset. **EA** juga harus dihubungkan ke **Vcc** untuk melakukan menjalankan program secara internal. Pada saat Flash Programming pin ini mendapatkan tegangan sebesar 12 Volt.

XTAL1 : Merupakan input ke penguat pembalik osilator dan ke rangkaian operasi Clock internal.

XTAL2 : Keluaran dari penguat pembalik osilator.



Gambar 2-17 Rangkaian Cristal [6]

Mikrokontroler AT89C51 memiliki rangkaian osilator internal dengan mengacu pada frekwensi referensi pada pin XTAL1 dan XTAL2.

2.8.3. Register Fungsi Khusus

AT89C51 mempunyai 21 ***Special Function Registers*** (Register Fungsi Khusus) yang terletak pada antara alamat 80H hingga FFH. Beberapa dari register – register ini juga bisa dialamati dengan pengalamatan bit sehingga dapat dioperasikan seperti yang ada pada RAM yang lokasinya dapat dialamati dengan pengalamatan bit.

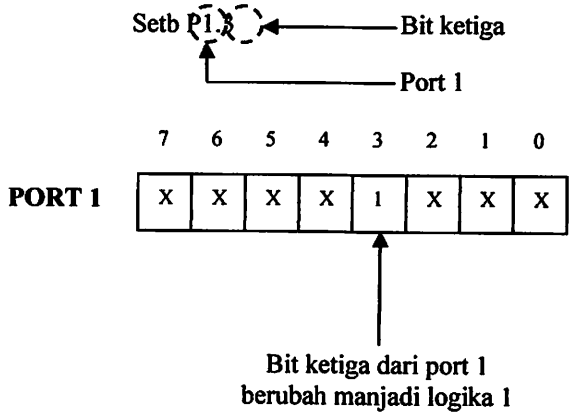
⇒ Accumulator

Register ini terletak pada alamat E0H. Hampir semua operasi aritmatik dan operasi logika selalu menggunakan register ini. Untuk proses pengambilan dan pengiriman data ke memori eksternal juga diperlukan register ini.

⇒ Port

AT89C51 mempunyai empat buah Port, yaitu Port 0, Port 1, Port 2 dan Port 3 yang terletak pada alamat 80H, 90H, A0H dan B0H. Namun, jika digunakan eksternal memoriataupun fungsi – fungsi special, seperti Eksternal Interrupt, Serial ataupun Eksternal Timer, Port 0, Port 2 dan Port 3 tidak dapat digunakan sebagai Port dengan fungsi umum. Semua Port ini dapat diakses dengan pengalamatan secara bit sehingga dapat dilakukan perubahan output pada tiap – tiap pin dari port ini tanpa mempengaruhi port – port yang lainnya.

Sebagai contoh, jika dilakukan instruksi Setb P1.3, maka bit ketiga dari port 1 akan berkondisi high (5V) tanpa mempengaruhi bit – bit yang lain pada port ini.



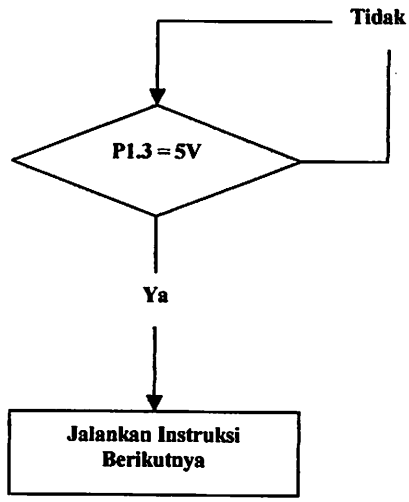
Gambar 2-18 Bit – Bit Port

Seperti yang tampak pada gambar 2 - 5, bit ketiga dari port 1 terletak pada alamat 93H oleh karena itu instruksi Setb P1.3 dapat diganti dengan instruksi Setb 93H.

Port ini digunakan untuk menunggu sinyal yang dikirim oleh komponen lain yang merupakan sinyal positif (5V) misalnya, dengan

```
Tunggu:
        Jnb     P1.3, tunggu
```

Selama kondisi pada port 1 pin ketiga masih low (0V), program akan terus melompat ke alamat yang ditunjukkan oleh label "tunggu" sehingga dapat diartikan bahwa program berhenti di alamat tersebut hingga terjadi sinyal positif (5V). Setelah sinyal positif (5V) muncul di bit ketiga dari port 1; program akan menuju ke alamat yang berikutnya:



Gambar 2-19 Diagram Alir Deteksi Bit Ketiga Port 1

⇒ Register B

Register B digunakan bersama accumulator untuk proses aritmatik selain dapat juga difungsikan sebagai register biasa. Register ini juga bersifat *Bit Addressable*.

⇒ Stack Pointer

Stack Pointer merupakan sebuah register 8 bit yang terletak di alamat 81H. Isi dari Stack Pointer ini merupakan alamat dari data yang disimpan di stack. Stack Pointer dapat diedit atau dibiarkan saja mengikuti standart sesudah terjadi reset. Jika Stack Pointer diisi data 5FH, area untuk proses penyimpanan dan pengambilan data dari dan ke stack adalah sebesar 32 byte, yaitu antara 60H hingga 7FH karena AT89C51 mempunyai Internal RAM sebesar 128 byte.

⇒ Data Pointer Two Byte Register (DPTR)

Data Pointer Two Byte Register atau DPTR merupakan register 16 bit dan terletak pada alamat 82H untuk DPL (Data Pointer Low) dan 83H untuk DPH (Data Pointer High). DPTR biasa digunakan untuk mengakses source code ataupun data yang terletak di memori eksternal.

Contoh:

```
MOV   A, #01h
MOV   DPTR, #2000H
MOVX  @Dptr,A
```

2000H. Pertama, data 01H diisikan ke accumulator. Kemudian, DPTR yang berfungsi untuk menunjukan alamat penyimpanan data diisi dengan 2000H. terakhir, isi dari accumulator A disimpan ke lokasi memori yang ditunjukan oleh DPTR (*Indirect Addressing*).

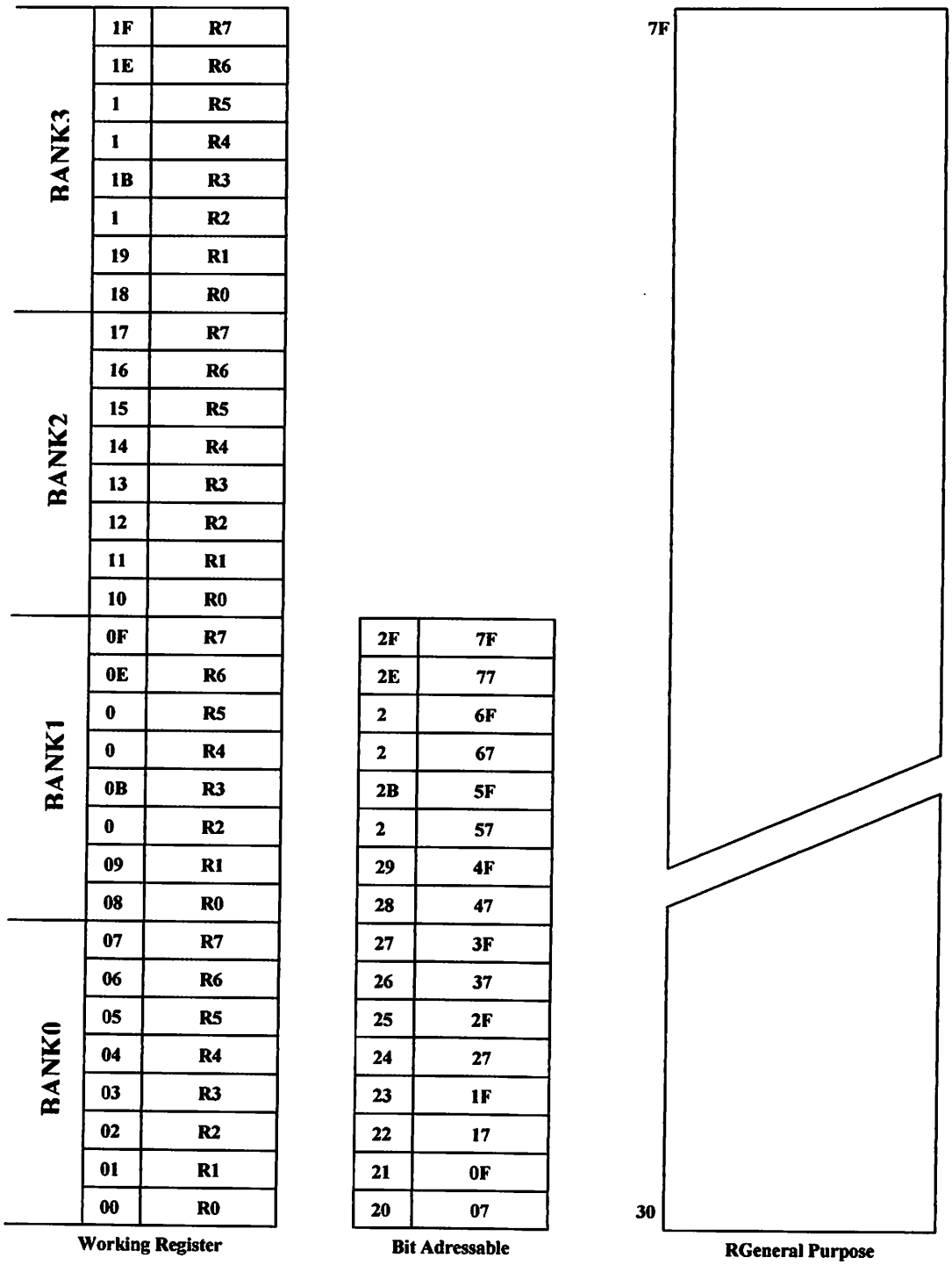
⇒ Register Port Serial

AT89C51 mempunyai sebuah *on chip serial port* (serial port dalam keping) yang dapat digunakan untuk berkomunikasi dengan peralatan

lain yang menggunakan serial port juga seperti modem, shift register dan lain – lain.

Buffer (Penyangga) untuk proses pengiriman maupun pengambilan data terletak pada register SBUF, yaitu pada alamat 99H. Sedangkan untuk mengatur mode serial dapat dilakukan dengan mengubah isi dari SCON yang terletak pada alamat 98H.

2.8.4. RAM Internal



Gambar 2-20 Ilustrasi Pembagian Ruang RAM AT89C51

⇒ RAM Internal, memori sebesar 128 byte yang biasanya digunakan untuk menyimpan variabel atau data yang bersifat sementara atau data akan hilang saat catu daya dimatikan.

⇒ *Special Function Register* (Register Fungsi Khusus), memori yang berisi register – register yang mempunyai fungsi – fungsi khusus yang disediakan oleh mikrokontroler tersebut, seperti timer, serial dan lain – lain.

⇒ *Flash PEROM*, memori yang digunakan untuk menyimpan instruksi – instruksi MCS – 51.

AT89C51 mempunyai struktur memori yang terpisah antara RAM Internal dan Flash PEROM – nya. RAM Internal dialamati oleh RAM *Address Register* (Register Alamat RAM) sedangkan Flash PEROM yang menyimpan perintah – perintah MCS – 51 dialamati oleh *Program Address Register* (Register Alamat Program). Dengan adanya struktur memori yang terpisah tersebut, walaupun RAM Internal dan Flash PEROM, mempunyai alamat awal yang sama, yaitu 00, namun secara fisik kedua memori tersebut tidak saling berhubungan.

RAM Internal terdiri atas:

⇒ Register BANKS

AT89C51 mempunyai delapan buah register yang terdiri atas R0 hingga R7. Kedelapan buah register ini selalu terletak pada alamat 00H hingga 07H pada setiap kali sistem direset. Namun, posisi R0 hingga R7 dapat dipindah ke BANK 1 (08 hingga 0FH), BANK 2

(10H hingga 17H), atau BANK 3 (18H hingga 1FH) dengan mengatur bit RS0 dan RS1.

⇒ Bit Addressable RAM

RAM pada alamat 20H hingga 2FH dapat diakses secara pengalamatan bit (*Bit Addressable*) sehingga hanya dengan sebuah instruksi saja setiap bit dalam area ini dapat diset, clear, AND dan OR. Sebagai contoh, pada saat terjadi instruksi Setb 67H, hal ini sama dengan

```
MOV     A,2CH          ; Pindahkan data dari alamat 2CH ke Acc
A
Orl     A,#10000000B  ; Set MSB Acc A
MOV     2CH,A          ; Pindahkan data dari Acc A ke alamat
```

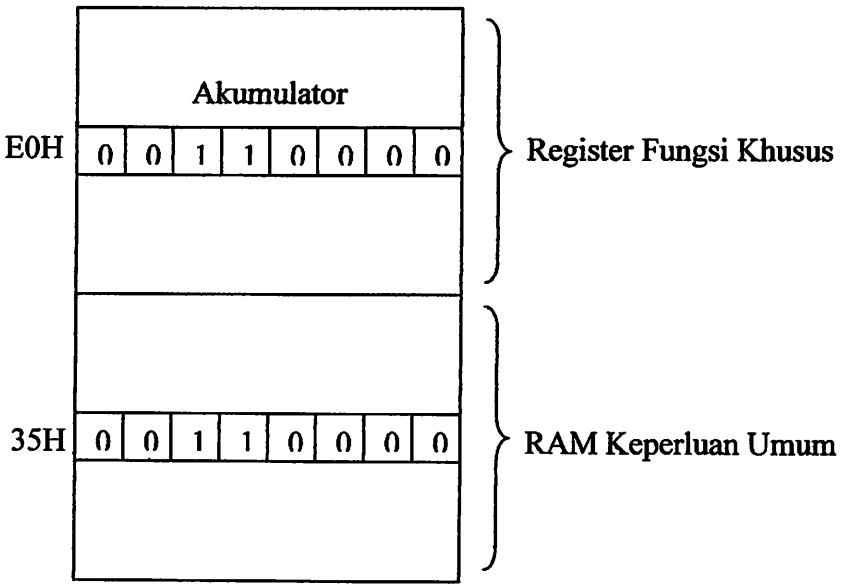
Dengan adanya sistem bit addressable RAM, proses yang seharusnya dijalankan dengan tiga cycle seperti listing diatas dapat digantikan dengan sebuah instruksi yang hanya membutuhkan satu instruksi saja. Dalam aplikasinya, lokasi yang dapat diakses dengan pengalamatan bit ini dapat juga digunakan untuk menandai suatu lokasi bit tertentu baik berupa Register Fungsi Khusus yang dapat dialamati secara bit (termasuk Register I/O) ataupun lokasi – lokasi tertentu yang dapat dialamati secara bit.

⇒ RAM Keperluan Umum

RAM keperluan umum dapat dimulai dari alamat 30H hingga 7FH dan dapat diakses dengan pengalamatan langsung maupun tak langsung. Pengalamatan langsung dilakukan ketika salah satu operand merupakan bilangan yang menunjukan lokasi yang dialamati seperti pada contoh berikut:

MOV A,35H ; Baca data dari alamat 35H dan disimpan di Accumulator.

Alamat yang
ditunjuk langsung



Gambar 2-21 Pengalamatan Secara Langsung

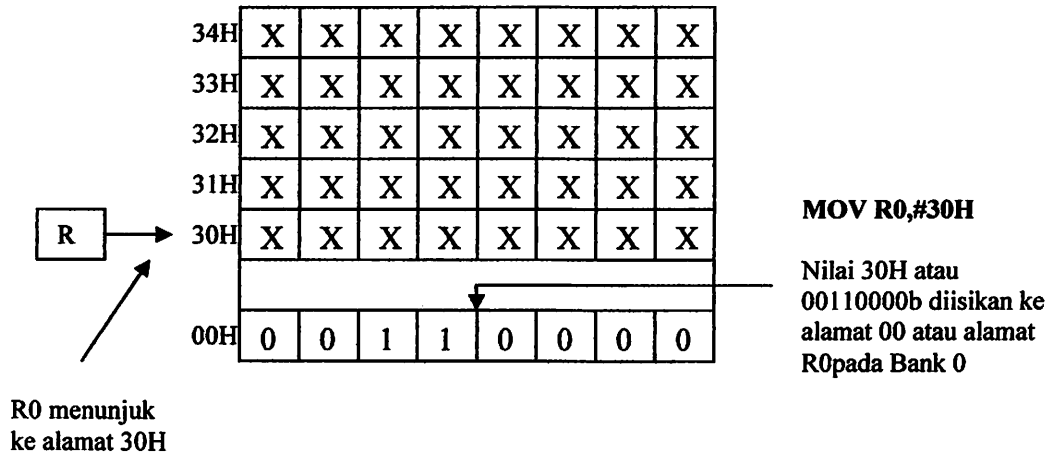
Sedangkan pengalamatan secara tak langsung pada lokasi dari RAM Internal ini adalah akses dari data memori ketika alamat memori tersebut tersimpan dalam suatu register R0 atau R1. R0 dan R1 adalah dua buah register pada mikrokontroler berarsitertur MCS – 51 yang dapat digunakan sebagai pointer dari sebuah lokasi memori pada RAM Internal. Sebagai Contoh:

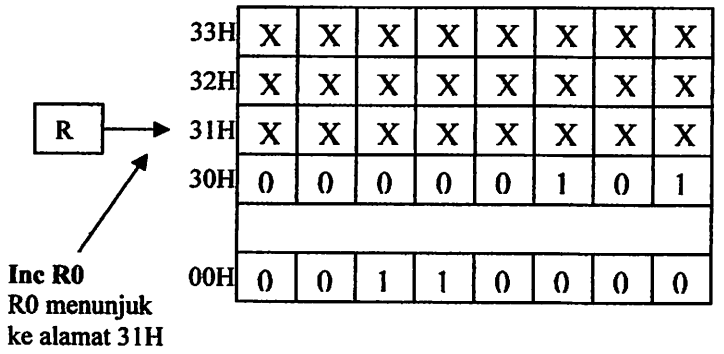
```
MOV  R0,#35H      ; R0 diisi dengan data 35H
MOV  A,@R0        ; Baca data di alamat yang dijuntuk oleh R0
```

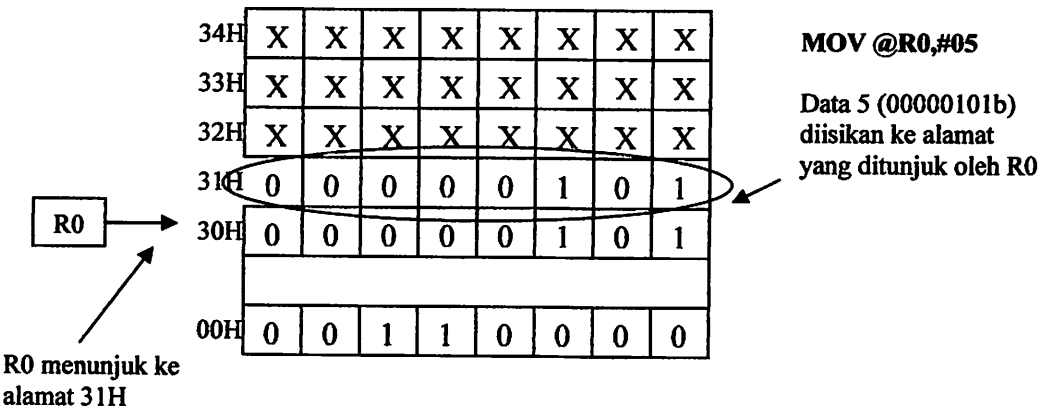
Pengalamatan secara tak langsung biasa digunakan untuk mengakses beberapa lokasi memori dengan letak yang beraturan seperti pada contoh berikut:

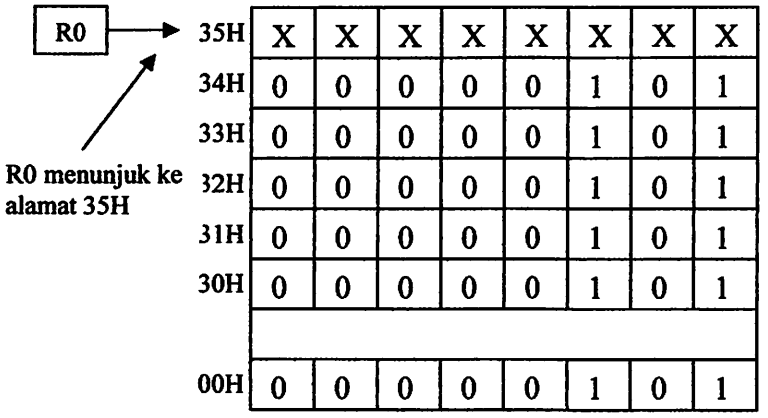
```
MOV     R0,#30H         ; R0 diisi dengan data 30H
Loop:
MOV     @R0,#05         ; Data 5 diisikan ke alamat yang
ditunjuk
                        ; oleh R0
Inc     R0              ; R0 menunjuk ke alamat selanjutnya
Cjne    R0,#35H,Loop    ; Jika R0 belum mencapai 35H,
lompat ke
                        ; label Loop
```

Pada gambar 2 - 8 yang merupakan step – step proses yang terjadi pada contoh program di atas, proses pemindahan data 5 ke alamat – alamat yang ditunjuk oleh R0 dilakukan berulang – ulang hingga R0, register yang berfungsi penyimpan nilai dari alamat yang diakses atau sebagai *pointer* (penunjuk) alamat yang diakses mencapai 35H


34H X X X X X X X X
33H X X X X X X X X
32H X X X X X X X X
31H X X X X X X X X
R
30H X X X X X X X X
00H 0 0 1 1 0 0 0 0
MOV R0,#30H
Nilai 30H atau 00110000b diisikan ke alamat 00 atau alamat R0pada Bank 0
R0 menunjuk ke alamat 30H


33H X X X X X X X X
32H X X X X X X X X
R
31H X X X X X X X X
30H 0 0 0 0 0 1 0 1
00H 0 0 1 1 0 0 0 0
Inc R0
R0 menunjuk ke alamat 31H


34H X X X X X X X X
33H X X X X X X X X
32H X X X X X X X X
31H 0 0 0 0 0 1 0 1
R0
30H 0 0 0 0 0 1 0 1
00H 0 0 1 1 0 0 0 0
MOV @R0,#05
Data 5 (00000101b) diisikan ke alamat yang ditunjuk oleh R0
R0 menunjuk ke alamat 31H



Gambar 2-22 Step – Step Yang Terjadi Pada Pemindahan Data 5 ke Alamat 30H hingga 34H

Langkah kerjanya: Pertama R0 diisi dengan data 30H sehingga register ini menunjuk ke alamat 30H dari RAM Internal. Kemudian data 5 diisikan alamat yang ditunjukan oleh R0 sehingga alamat 30H (alamat yang ditunjuk oleh R0 saat itu) akan berisi data 5. Perintah **Inc R0** menyebabkan nilai dalam R0 bertambah 1 menjadi 31H sehingga register ini menunjuk ke alamat 31H. Oleh karena R0 belum mencapai 35H (Cjne R0,#05). Demikian berlangsung seterusnya hingga pada saat R0 menunjuk ke alamat 35H proses pemindahan data 5 tersebut tidak dilakukanlagi sehingga alamat 30H hingga 34H terisi dengan data 5.

2.9. Mikrokontroler Renesas R8C/13 Tiny (R5F21134FP)

Renesas Technology adalah produsen semikonduktor tingkat internasional. Renesas terbangun dari gabungan dua produsen semikonduktor, yaitu Mitsubishi dan Hitachi. Sebagai produsen semikonduktor, renesas juga mengeluarkan berbagai jenis keluarga mikrokontroler (MK).

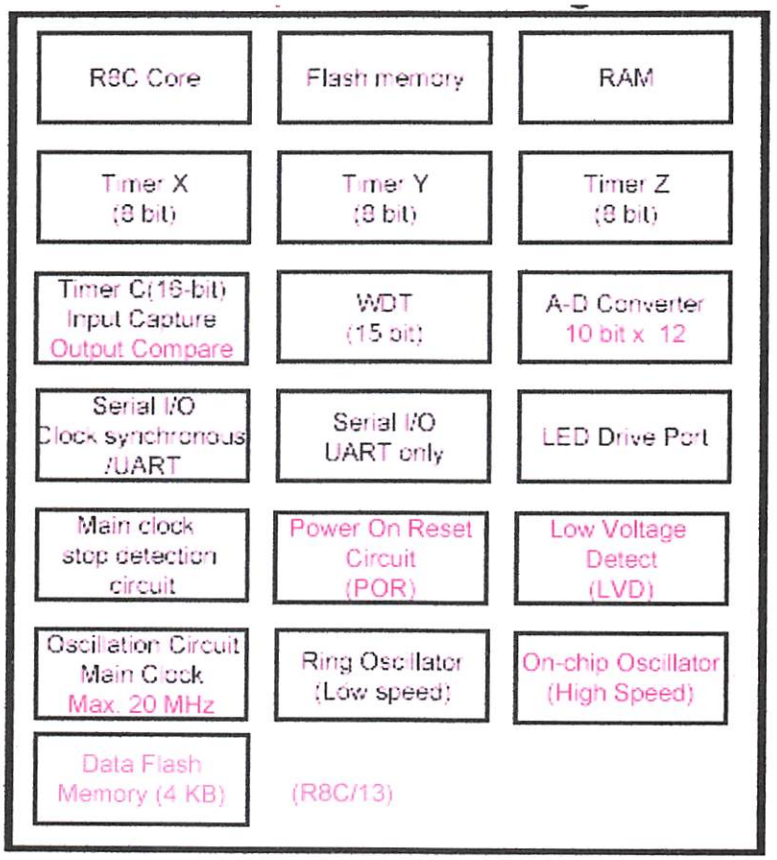
Renesas R8C adalah salah satu jenis seri dalam keluarga MK M16C. CPU R8C sama dengan CPU CISC 16-*bit* M16C, hanya saja lebar jalur data R8C adalah 8-*bit.* Karena menggunakan CPU yang sama maka R8C memiliki *instruction set* hampir sama dengan M16C. Perbedaannya hanya terletak pada 2 instruksi, yaitu R8C tidak memiliki instruksi JMPS *(Jump Special Page)* dan JSRS *(Jump Subroutine Special Page).* R8C/13 adalah salah satu tipe MK dalam seri R8C. MK ini memiliki kemasan 32-*pin* LQFP. Dalam perancangan pada skripsi ini menggunakan menggunakan MK seri R5F21134, yaitu R8C/13 yang memiliki Flash ROM 16 KB (1000 E/W *cycles*) dan RAM sebesar 1 KB.

2.9.1. Spesifikasi R5F21134FP

Berikut ini adalah spesifikasi *R5F21134FP* dengan peta peripheral dan memori-memorinya.

- Mempunyai *CPU Core* (16-bit) 1 – 20 MHz, 3.0 – 5.5 Volt dan 1 – 10MHz 2.7 – 5.5 Volt.
- Rangkaian Clock, kecepatan *Low/High On-Chip Oscillator. Clock* utama dengan Xin/Xout.

- Memory (ROM/SRAM) 16 Kbytes / 1 Kbytes, 2 x 2 K Bytes Data Flash pada R8C/12, 13.
- Kemasan 32 pin LQFP (7mm x 7mm)



Gambar 2-23 Blok Diagram R8C/11, 13 dan Peta *Peripheral*-nya [9]

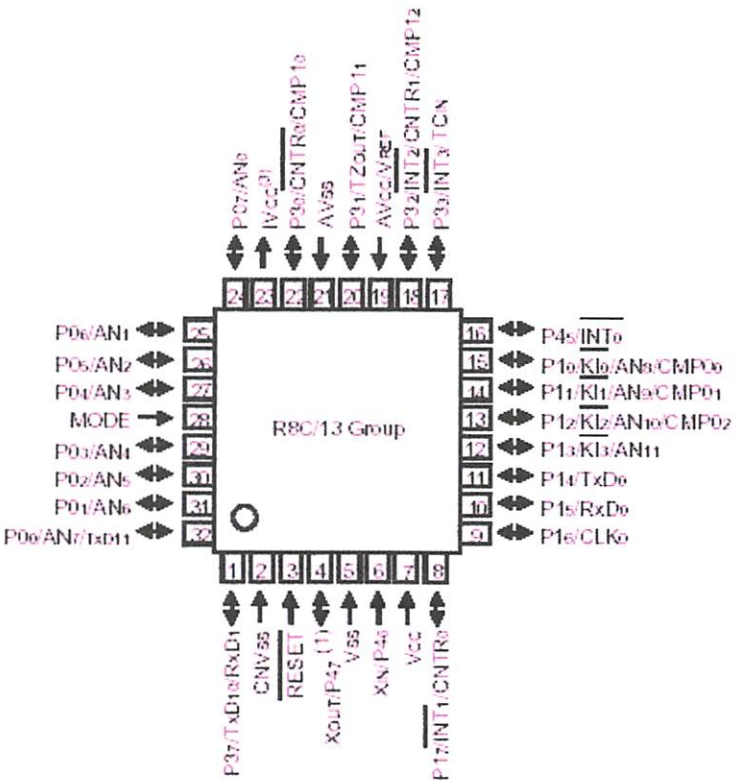
2.9.2. Kelebihan Kunci R8C/Tiny

Banyak kelebihan-kelebihan yang dimiliki R8C/Tiny diantaranya adalah :

- Kompatibel dengan M16C yaitu kompatibel dalam instruksi dan kode.
- *Peripheral* lebih terintegrasi jadi lebih hemat.
- *Electromagnetic Compatibility* (EMC) mempunyai EMI rendah, EMS tinggi.

- ❖ *Development Tool* (*Compiler* dan *Debugger*) didapat dengan murah dan difasilitasi *On-Chip Debugger*
- ❖ Mempunyai fitur *fail-safe* yaitu pengamanan terhadap kegagalan sistem.
- ❖ Konsumsi daya rendah.
- ❖ 16-bit CISC CPU dengan kecepatan maksimal 20 MHz (1:1).
- ❖ 89 instruksi CISC lebih hemat ROM kira-kra 20 %, RAM sampai 1 KB.
- ❖ Waktu konversi ADC hanya 3 uS.

2.9.3. Konfigurasi Pin R8C R5F21134FP



Gambar 2-24. Konfigurasi Pin R8C R5F21134FP [9]

Gambar diatas adalah kofigurasi pin-pin dari *R8C R5F21134FP* untuk lebih jelasnya dapat diamati pada tabel dekripsi pin-pin berikut ini :

Tabel 2-6 Kofigurasi pin-pin dari *R8C R5F21134FP* [9]

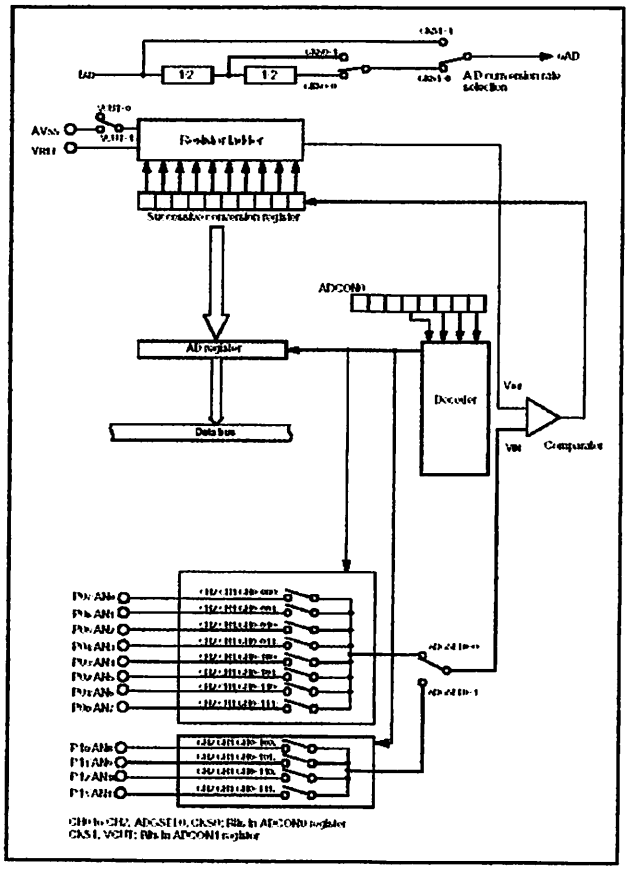
Nama Sinyal	Nama Pin	Type I/O	Fungsi
Masukan Catu Daya	Vcc, Vss	I	Tegangan 2.7 V – 5.5 V pada pin Vcc. Tegangan 0 V pada Vss pin
I Vcc	Ivcc	O	Pin ini untuk men-stabilkan catu daya *internal,* pin ini dihubungkan pada Vss melalui kapasitor 100nF. Jangan dihubungkan pada Vcc.
Input Catu Daya Analog	Avcc, Avss	I	Ini adalah untuk catu daya pada ADC. Avcc dihubungkan pada Vcc, A Vss dihubungkan ke Vss. Dianjurkan untuk menghubungkan kapasitor diantara pin A Vcc dan A Vss.
Input Reset	RESET	I	"L" untuk masukan ini mereset MCU
CNVss	CNVss	I	Pin ini dihubungkan pada Vss melalui sebuah resistor.
MODE	MODE	I	Pin ini dihubungkan pada Vcc melalui sebuah resistor.
Input Clock Utama	Xin	I	Pin-pin ini disediakan untuk membangkitkan rangkaian I/O Clock Utama. Dihubungkan dengan sebuah keramik resonator atau kristal diantara pin Xin dan Xout. Jika digunakan clock internal maka pin Xin dan Xout dalam keadaan terbuka.
Output Clock Utama	Xout	O	
Input Interupsi	INT0 –INT3	I	Pin ini sebagai masukan interupsi.
Input Kunci Interupsi	KI0 – KI3	I	Pin ini sebagai masukan kunci interupsi.
Timer X	CNTR 0	I/O	Pin I/O ini adalah untuk Timer X .
	CNTR 0	O	Pin Ouput untuk Timer X.

Timer Y	CNTR 1	I/O	Pin I/O untuk Timer Y.
Timer Z	TZout	O	Pin Ouput untuk Timer Z.
Timer C	TC in	I	Pin Input untuk Timer C.
	CMP00 – CMP03, CMP10 CMP13	O	Pin Output untuk Timer C.
Serial Interface	CLK 0	I/O	Pin I/O untuk memindahkan Clock.
	RXD0, RXD1	I	Pin input untuk data Serial.
	TXD0, TXD10, TXD11	O	Pin output untuk data Serial.
Input Tegangan Referensi	Vref	I	Tegangan referensi input ini untuk ADC. Vref pin dihubungkan ke Vcc.
ADC, pengubah dari analog ke digital	AN0–AN11	I	Pin analog input pada ADC.
Port I/O	P00-P07, P10-P17, P30-P33, P37, P45	I/O	Merupakan port I/O CMOS 8-bit . Setiap port mempunyai pilihan register pengarah sebagai input atau output. Tiap Port dapat dialamati per bit. Dapat di-set menggunakan pull up resistor dengan program. P10 – P17 mempunyai driver transistor.
Port Input	P46, P47	I	Pin ini hanya bisa digunakan sebagai input.

2.9.4. *Peripheral* R8C R5F21134FP

Mikrokontroler *R8C R5F21134FP* mempunyai beberapa *peripheral-peripheral* yang banyak digunakan pada beberapa aplikasi-aplikasi penting, diantaranya adalah sebagai berikut :

- ***Analog To Digital Converter*** **(ADC)**



Gambar 2-25 Diagram blok ADC [9]

Dengan 12 SAR ADC S/H yang mempunyai resolusi 8-bit atau 10-bit. Mode Operasinya menggunakan *One-Shot dan Repeat* dengan waktu konversi 2.8 uS (pada clock 10 MHz). Berikut gambar diagram blok ADC *built in* pada mikrokontroler ini :

❖ ***Timer Mode***

Mempunyai timer sebanyak 4 yaitu timer X, Y, Z, C. Berikut adalah mode-mode timernya :

Tabel 2-7 Mode-mode Timer [9]

Item		Timer X	Timer Y	Timer Z	Timer C
Configuration		8-bit timer with 8-bit prescaler	8-bit timer with 8-bit prescaler	8-bit timer with 8-bit prescaler	16-bit timer
Count		Down	Down	Down	Up
Count source		•f1 •f2 •f8 •f32	•f1 •f8 •fRING •Input from CNTR1 pin	•f1 •f2 •f8 •Timer Y underflow	•f1 •f8 •f32
Function	Timer mode	provided	provided	provided	not provided
	Pulse output mode	provided	not provided	not provided	not provided
	Event counter mode	provided	provided[1]	not provided	not provided
	Pulse width measurement mode	provided	not provided	not provided	not provided
	Pulse period measurement mode	provided	not provided	not provided	not provided
	Programmable waveform generation mode	not provided	provided	provided	not provided
	Programmable one-shot generation mode	not provided	not provided	provided	not provided
	Programmable wait one-shot generation mode	not provided	not provided	provided	not provided
	Capture	not provided	not provided	not provided	provided
Input pin		CNTR0	CNTR1	$\overline{INT0}$	TCIN
Output pin		CNTR0 $\overline{CNTR0}$	CNTR1	TZOUT	not provided
Related interrupt		Timer X int $\overline{INT1}$ int	Timer Y int $\overline{INT2}$ int	Timer Z int $\overline{INT0}$ int	Timer C int $\overline{INT3}$ int
Timer stop		provided	provided	provided	provided

❖ ***Low Voltage Detect*** **(LVD)**

LVD adalah untuk mendeteksi Vcc krang dari 3.8 V (± 0.5 V)

❖ ***Watchdog Timer***

Watchdog berfungsi untuk mendeteksi ketika program diluar kontrol.

❖ ***On Chip Debugger***

Fasilitas ini mempunyai fungsi untuk dapat di-*debug* pada waktu mikro sedang berjalan. Antara PC dan MK dapat berkomunikasi, PC akan mengetahui aktivitas MK saat itu. Syarat-syarat *On Chip Debugger* adalah:

- Vektor *Address Match interrupt* harus dihindari.
- *Single step interrupt* tidak dapat digunakan bersamaan interrupt lain.
- *UART1* tidak boleh dipakai.
- Instruksi BRK tidak boleh dipakai.
- Flash Address C000H – C7FFH.
- PD 3.7 harus "0".
- B5 FMR 0 harus "1"
- Menyiapkan 8 Byte untuk Stack.
- *On Chip Debugger* berpengaruh pada *timing run.*

❖ **Rangkaian Osilator**

Pada osilator utama menggunakan kristal luar sampai dengan 20 MHz, dengan memiliki fitur *Clock Stop Detect.* Kemudian untuk *On Chip* Osilator disediakan kecepatan *Low* 125 KHz dan *High* 8 MHz. Saat setelah reset, default clock adalah kecepatan rendah *On Chip* osilator 125 KHz.

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

3.1. Pendahuluan

Dalam bab ini akan dibahas perancangan dan pembuatan alat. Pembahasan akan dilakukan pada setiap blok rangkaian, cara kerja masing-masing blok rangkaian, perhitungan dan fungsi masing-masing blok rangkaian tersebut. Secara garis besar terdapat dua bagian perangkat yang ada yaitu :

- Perancangan perangkat keras *(Hardware).*
- Perancangan perangkat lunak *(Software).*

Pada perancangan perangkat keras akan meliputi seluruh *peripheral* yang digunakan pada sistem ini. Pada perancangan perangkat lunak akan meliputi diagram alir dan *software* secara umum. Akan tetapi kedua perangkat ini dalam kerjanya akan saling menunjang satu sama lain.

Secara umum sistem kerja dari keseluruhan sistem ini adalah untuk mengatur kelancaran lalulintas pada perempatan atau percabangan jalan agar tidak terjadi kemacetan, pada pengaturan lampu lalulintas ini dilengkapi dengan *dot matriks* pada setiap jalannya untuk mempermudah pengguna jalan pada saat pergantian lampu lalulintas dari merah ke hijau atau dari hijau ke kuning lalu ke merah. Sistem kerja dari dari pengaturan lampu lalulintas ini berdasarkan pada waktu yang terlebih dulu disurvei yaitu waktu padat, waktu normal, dan waktu tenggang (malam hari dari jam 22.00 sampai 05.30 WIB) dari ketiga waktu

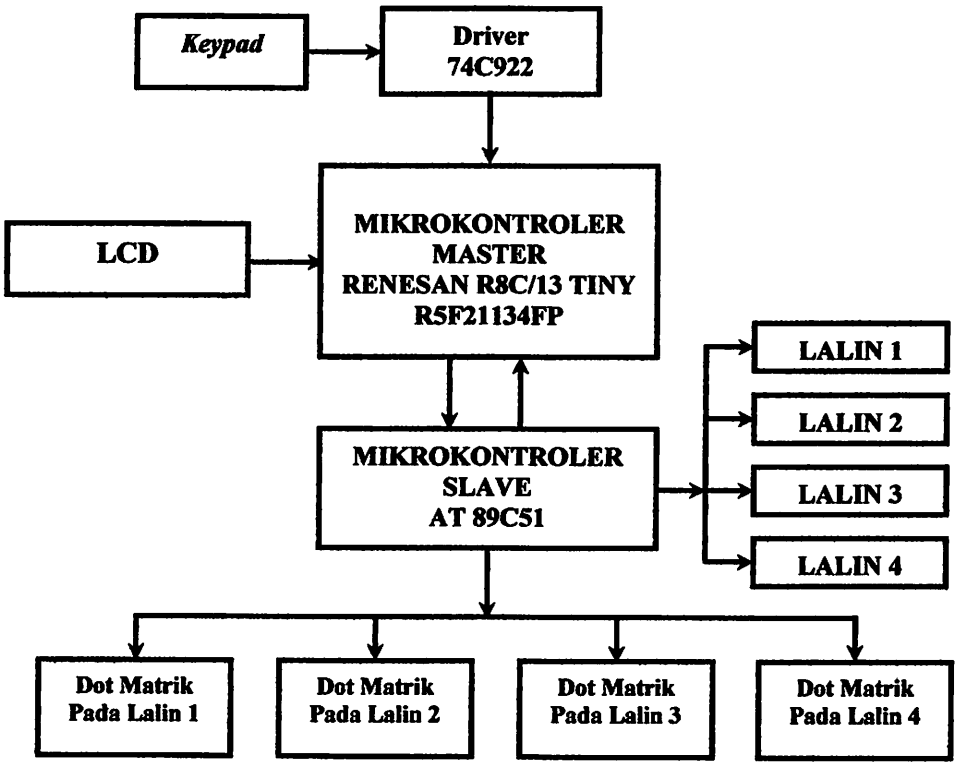
tersebut lama nyala lampu lalulintas tidak sama, nyala lampu lalulintas diset meleui *keypad* sesuai dengan sistem pada DLLAJ (Dinas Lalulintas Angkutan Jalan), inputan dari *keypad* ditampilka pada LCD pada setiap waktu yang telah ditetapkan. Inputan dari *keypad* diproses oleh mikrokontroler master terus ditampilkan pada LCD dan data hasil proses dari inputan *keypad* dikirim ke mikrokontroler slave untuk menjalankan sistem pengaturan lampu lalulintas dan lama nyala lampu lalulintas tersebut berdasarkan inputan data dari *keypad* sesuai dengan ketetapan pada DLLAJ (Dinas Lalulintas Angkutan Jalan). Setelah inputan selesai dan sistem berjalan sesuai dengan program, LCD menampilkan waktu keadaan sekaran sebagai inputan untuk menjalan sistem sesuai dengan waktu-waktu yang telah ditetapkan.

3.1.1. Diagram Blok Keseluruhan Sistem

Gambar 3-1 merupakan diagram blok keseluruahan sistem yang secara umum terdiri dari masukan-masukan dan keluaran-keluaran yang diproses oleh mikrokontroler. Mikrokontroler dalam diagram blok terdapat 2 bagian yaitu mikrokontroler *Master* Renesas R8C Tiny R5F21134FP dan mikrokontroler *Slave* AT89C51 yang bekerja saling berhubungan satu dengan yang lainnya.

Mikrokontroler dalam diagram blok terdapat 2 bagian yaitu mikrokontroler *Master* Renesas R8C /13 Tiny R5F21134FP dan mikrokontroler *Slave* AT89C51 yang bekerja saling berhubungan satu dengan yang lainnya, mikrokontroler *Master* disini bertugas sebagai pengendali utama dari mikrokontroler *Slave 1*.

Berikut penjelasan secara umum masing-masing diagram blok:



Gambar 3-1. Diagram Blok Keseluruhan Sistem

- **Mikrokontroler *Master* Renesas R8C Tiny R5F21134FP.**

  Mikrokontroler ini merupakan mikrokontroler pengendali utama yaitu menerima masukan-masukan dari *keypad, keypad* memberi inputan pada mikrokontroler mester kemudian diproses ditampilkan pada LCD hasil inputan dari *keypad* dan data hasil prosesnya dikirim kemikrokontroler slave untuk menjalankan proses *trafic light* (lampu lalulintas) yang disertai tampilan *dot matriks* sebagai tampilan hitung waktu mundur, agar pengendara mengetahui berapa lama perpindahan nyala lampu merah kehijau san berlaku sebaliknya. Disamping itu juga mikroontroler master ini memproses atau menampilkan timer sebagai waktu yang ditampilkan pada LCD. Banyak pertimbangan pada mikrokontroler master dipilih jenis

Renesas R8C /13 Tiny R5F21134FP, seperti yang dibahas pada bab sebelumnya.

- ***Keypad***

  Keypad berfugsi sebagai inputan data digital untuk mengubah lama nyala lampu lalulintas yang disertai dengan tampilan hitung waktu mundur pada *dot matriks*. Data tersebut bisa diubah sewaktu-waktu sesuai dengan perkembangan dan kepadatan kendaraan yang melewati jalan tersebut.

- **Driver (IC 74C922)**

  IC 74C922 merupakan jenis IC CMOS yang berfungsi untuk menerjemahkan dari inuputan *keypad* (bilangan biner) menjadi bilangan desimal yang hasil penerjemahan tersebut langsung dikirim kemikrokontroler sehingga mudah diproses oleh mikrokontroler dan ditampilkan ke LCD.

- **LCD M1632.**

  Menampilkan data-data dari masukan untuk dibaca oleh *user*/pengguna, dalam sistem ini menampilkan informasi inputan data yang dikirim dari *keypad* dan mempilkan *timer* sebagai fungsi waktu.

- ***Dot Matriks***

  Dot Matriks berfungsi untuk menampilkan hitungan waktu mundur dari nilai tertinggi sampai nol (0) sesuai dengan pergantian nyala lampu lalulintas, lampu lalulintas akan berubah dari lampu merah ke lampu hijau pada saat tampilan atau hitung waktu mundur pada *dot matriks* menampilkan nol (0). Lama waktu hitung mundur dan pergantian lampu lalulintas sesuai dengan inputan data yang telah diset oleh *keypad.*

- **LED (Lampu Lalulintas)**

  LED sebagai tampilan lampu lalulintas (merah, kuning dan hijau). Disaat nyala lampu merah, *dot matriks* menampilkan hitung waktu mundur dari nilai tertinggi sesuai dengan inputan *keypad* pada waktu tertentu juga sampai nol (0) dan berganti pada lampu hijau, *dot matriks* menampilkan hitung waktu mundur dari nilai tertinggi sesuai dengan inputan *keypad* pada waktu tertentu juga sampai nol (0) kemudian ke lampu kuning dan ke lampu merah lagi. Pada setiap jalurnya berlaku sistem yang sama, dengan sistem pergantian dari jalur 1 ke jalur 2 searah dengan jalur jam sampai ke jalur 4 secara terus menerus.

- **Mikrokontroler *Slave***

  Mikrokontroler ini merupakan mikrokontroler untuk *Dot Matrik LED* dan *traffic light* yaitu menampilkan hitungan waktu mundur pada *dot matriks* dan lampu merah, kuning dan hijau sebagai simstem *traffic light*, menginformasikan berapa lama nyala lampu merah dan hijau dengan cara perhitungan waktu mundur yang ditampilkan pada *dot matriks*. Mikrokontroler ini menerima data dari mikrokontroler master yaitu data inputan dari *keypad* sebagai seting berapa lama nyala lampu merah dan hijau dan kemudian ditampilkan pada *dot matrik* LED. Mikrokontroler ini memberikan pulsa *acknowledge* atau sinyal balasan ke mikrokontroler master bahwa sistem telah dioperasikan pada *traffic light* dan *dot matrik*, memberikan pesan pada mikrokontroler master untuk melanjutkan proses selanjutnya.

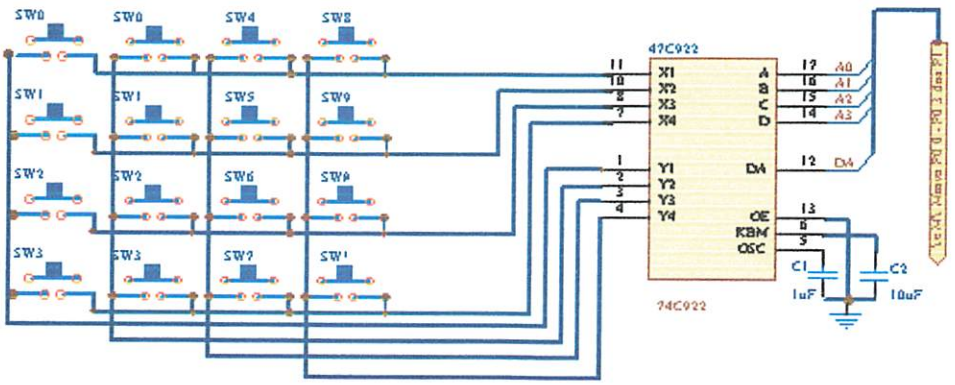
3.2. Perancangan Perangkat Keras

Perancangan perangkat keras terdiri dari beberapa bagian secara umum dapat dibagi menjadi 3 bagian utama yaitu :

- ❖ Bagian masukan yang terdiri dari *keypad* dan timer.
- ❖ Bagian pemroses data yaitu mikrokontroler.
- ❖ Bagian keluaran yang terdiri dari *driver-driver*, LCD dan dot matrik LED.

3.2.1. *Keypad 4 x 4*

Rangkaian *keypad* merupakan komponen yang digunakan sebagai sarana untuk kekomputer atau minimum sistem untuk rangkaian *keypad* ini menggunakan IC 74C922 (jenis CMOS) sebagai penerjemah kode desimal ke biner, karena IC jenis ini sudak memiliki beberapa kelengkapan. Didalam IC tersebut sudah memiliki cukup kelengkapan jika dibandingkan dengan IC jenis yang lainnya, IC lainnya memerlukan 1 kapasitor external, rangkaian internal register atau pengingat tombol terakhir yang ditekan meskipun tombol sudah dilepas. Selain itu *keypad* juga dapat dihubungkan langsung dengan mikrokontroler akan tetapi jika menggunakan cara ini maka harus sedia port lebih. Rangkaian *keypad* ini dihubungkan kemikrokontroler mulai dari P0.0 sampai P0.3 dan P1.0.



Gambar 3.2. Perancangan *Keypad*

3.2.2. *Perancangan* Rangkaian LCD

LCD Display Module M1632 buatan Seiko Instrument Inc. terdiri dari dua bagian, yang pertama merupakan panel LCD sebagai media penampil informasi dalam bentuk huruf atau angka dua baris, masing-masing baris bisa menampung 16 huruf atau angka.

Bagian kedua merupakan sebuah sistem yang dibentuk dengan mikrokontroler yang ditempelkan dibalik pada panel LCD, berfungsi mengatur tampilan informasi serta berfungsi mengatur komunikasi L1632 dengan mikrokontroler yang memakai tampilan LCD itu. Dengan demikian pemakaian M1632 menjadi sederhana, sistem lain yang M1632 cukup mengirimkan kode-kode ASCII dari informasi yang ditampilkan seperti layaknya memakai sebuah printer.

Untuk berhubungan dengan mikrokontroler pemakai, M1632 dilengkapi dengan 4 jalur data (**DB4..DB7**) yang dipakai untuk menyalurkan kode ASCII

maupun perintah pengatur kerjanya M1632. Selain itu dilengkapi pula dengan **E**, **R/W** dan **RS** seperti layaknya komponen yang kompatibel dengan mikroprosesor.

Kombinasi lainya **E** dan **R/W** merupakan sinyal standar pada komponen buatan Motorola. Sebaliknya sinyal-sinyal dari MCS51 merupakan sinyal khas Intel dengan kombinasi sinyal **WR** dan **RD.**

RS, singkatan dari Register Select, dipakai untuk membedakan jenis data yang dikirim ke M1632, kalau **RS=0** data yang dikirim adalah perintah untuk mengatur kerja M1632, sebaliknya kalau **RS=1** data yang dikirim adalah kode ASCII yang ditampilkan.

Demikian pula saat pengambilan data, saat **RS=0** data yang diambil dari M1632 merupakan data status yang mewakili aktivitas M1632, dan saat **RS=1** maka data yang diambil merupakan kode ASCII dari data yang ditampilkan.

Proses mengirim/mengambil data ke/dari M1632 bisa dijabarkan sebagai berikut :

1. **RS** harus dipersiapkan dulu, untuk menentukan jenis data seperti yang telah dibicarakan di atas.
2. **R/W** di-nol-kan untuk menandakan akan diadakan pengiriman data ke M1632. Data yang akan dikirim disiapkan di **DB4..DB7**, sesaat kemudian sinyal **E** di-satu-kan dan di-nol-kan kembali. Sinyal **E** merupakan sinyal sinkronisasi, saat **E** berubah dari 1 menjadi 0 data di **DB4..DB7** diterima oleh M1632.
3. Untuk mengambil data dari M1632 sinyal **R/W** di-satu-kan, menyusul sinyal **E** di-satu-kan. Pada saatu **E** menjadi 1, M1632

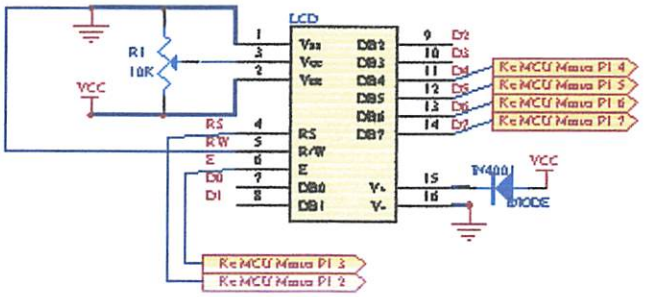
akan meletakkan datanya di **DB4..DB7**, data ini harus diambil sebelum sinyal **E** di-nol-kan kembali.

M1632 mempunyai seperangkat perintah untuk mengatur tata kerjanya, perangkat perintah tersebut meliputi perintah untuk menghapus tampilan, meletakkan kembali cursor pada baris huruf pertama baris pertama, menghidup/matikan tampilan dan lan sebagainya, semua itu dibahas secara terperinci dalam lembar data M1632.

Setelah diberi sumber daya, ada beberapa langkah persiapan yang harus dikerjakan dulu agar M1632 bisa dipakai, langkah-langkah tersebut antara lain adalah:

1. Tunggu dulu selama 15 mili-detik atau lebih.
2. Kirimkan perintah 30h, artinya trasfer data antar M1632 dan mikrokontroler dilakukan dengan mode 8 bit
3. Tunggu selama 4.1 mili-detik
4. Kirimkan sekali lagi perintah 30h
5. Tunggu lagi selama 100 mikro-detik

Setelah langkah-langkah tersebut di atas M1632 barulah bisa menerima data dan menampilkannya dengan baik. Pada awalnya tampilan akan nampak kacau, dengan demikian perlu segera dikirim perindah menghapus tampilan dan lain sebagainya, sesuai dengan petunjuk yang ada di lembar data.



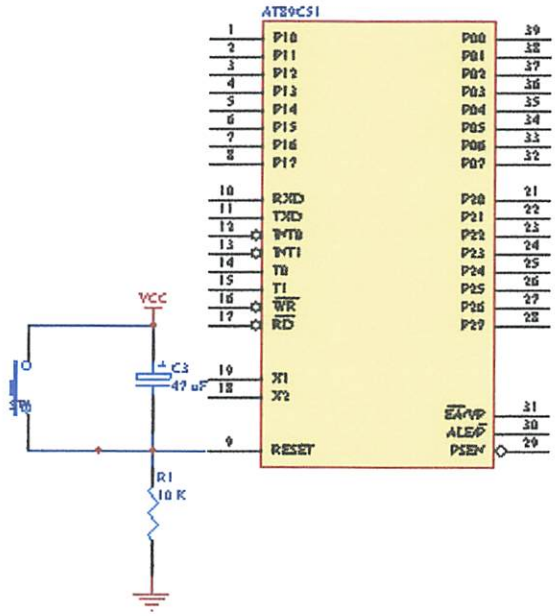
Gambar 3-3. Perancangan Rangkaian *Liquid Crystal Display* (LCD)

Untuk tampilan dipergunakan LCD Dot Matrik 2 x 16 karakter. Sinyal-sinyal yang diperlukan oleh LCD adalah RS dan Enable, sinyal RS dan Enable dipergunakan sebagai input yang outputnya dipakai untuk mengaktifkan LCD. LCD akan aktif apabila mikrokontroller memberikan instruksi tulis pada LCD. Saat kondisi RS don't care dan Enable 0 maka LCD tetap pada kondisi semula, pengiriman data ke LCD dilakukan saat RS berlogika 0 dan enable berlogika 1. Instruksi dikirim pada LCD bila keadaan RS 1 dan Enable 1. Pin LCD ini untuk data terkoneksi pada *Port* 1.4 sampai *Port* 1.7 mikrokontoler master. Kemudian untuk RS dihubungkan pada *Port* 1.2, tulis/baca (*Read/Write)* diberikan logika *low* karena disini LCD bersifat menulis data, dan yang terakhir *Enable* (E) dikendalikan dengan *Port* 1.3. Gambar rangkaian LCD ditunjukkan pada gambar 3-3.

3.2.3. *Perancangan* Rangkaian *Reset*

Untuk m*ereset* mikrokontroler AT89C51, maka pin RST diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan

sinyal *reset* kapasitor dihubungkan dengan Vcc dan sebuah resistor yang dihubungkan ke *ground*. Rangkaian *reset* ditunjukkan dalam gambar 3-4 sebagai berikut :



Gambar 3-4. Perancangan Rangkaian *Reset*

Karena kristal yang digunakan mempunyai frekuensi sebesar 11,0592 MHz, maka satu periode membutuhkan waktu sebesar :

$$T = \frac{1}{f_{XTAL}} = \frac{1}{11{,}0592\,MHz}\ S = 9{,}042 \times 10^{-8}\ s \quad \text{.....(3-2)}$$

Sehingga waktu minimal logika tinggi yang dibutuhkan untuk me-*reset* mikrokontroler adalah :

Reset (min) = T × periode yang dibutuhkan.....(3-3)

= 9,042 × 10 -8 × 24 = 2,17 μs

Jadi mikrokontroler membutuhkan waktu minimal 2,17 μs untuk m-*ereset*. Waktu minimal inilah yang dijadikan pedoman untuk menentukan nilai R dan C. Dari persamaan (3-3) dengan menentukan nilai R = 8,2 kΩ dan C = 10 μF, maka :

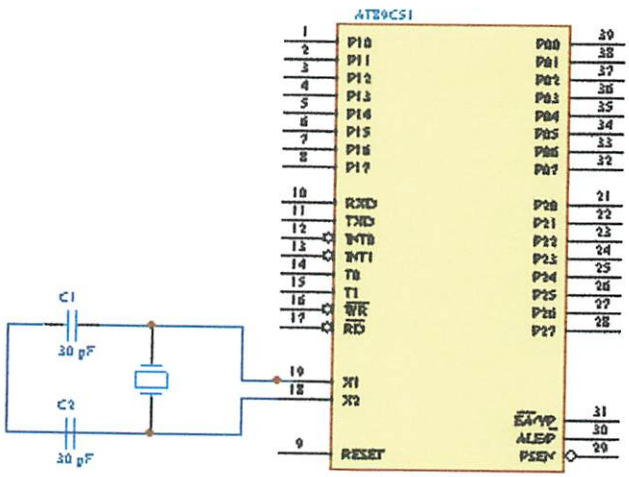
$$t = 0{,}357\ R.C = 0{,}357 \times 8200\Omega \times 10.10^{-6} = 29{,}274\ ms$$

Jadi dengan nilai komponen R = 8,2 kΩ dan C = 10μF dapat memenuhi syarat minimal untuk waktu yang dibutuhkan oleh mikrokontroler.

3.2.4. *Perancangan Clock*

Kecepatan proses yang dilakukan oleh mikrokontroler ditentukan oleh sumber *clock* yang mengendalikan mikrokontroler tersebut. Sistem yang dirancang ini menggunakan osilator internal yang telah tersedia dalam *chip* AT89C51. Untuk menentukan frekuensi osilatornya cukup dengan menghubungkan kristal dalam pin 19 (X_1) dan pin 18 (X_2) serta dua buah kapasitor ke *ground.*

Besarnya kapasitansinya disesuaikan dengan spesifikasi dalam lembar data AT89C51 yaitu 30 pF. Kristal yang digunakan adalah 11,0592 MHz. Gambar 3-5 memperlihatkan rangkaian *clock* yang dirancang.



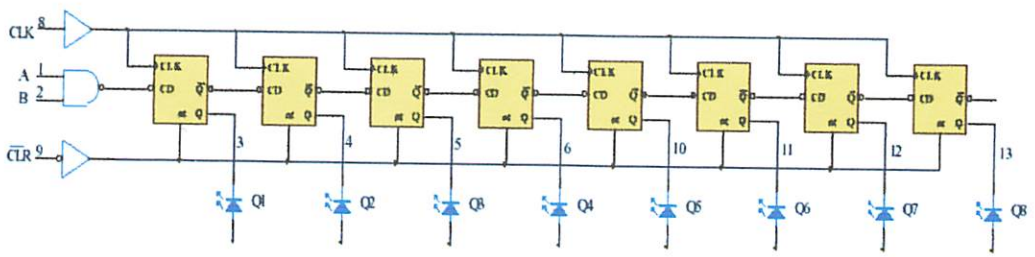
Gambar 3-5. Perancangan Rangkaian *Clock*

3.2.5. *Perancangan* Rangkaian *Dot Matrik LED* 5 x 7 8 Modul

Pada perancangan rangkaian Dot Matrik LED 5 x 7 ini dikendalikan oleh bagian mikrokontroler *Slave 2.* Mikrokontroler ini menerima masukan dari mikrokontroler *Master* berupa pulsa banyaknya orang yang masuk kedalam ruangan yang kemudian diproses oleh mikrokontroler *Slave 2* dan informasinya ditampilkan pada Dot Matrik LED. Mikrokontroler *Slave 2* juga memberikan pulsa balik *(acknowledge)* ke mikrokontroler *master* bahwa tulisan informasi sudah selesai dituliskan.

3.2.5.1. Rangkaian Register Geser 8 bit

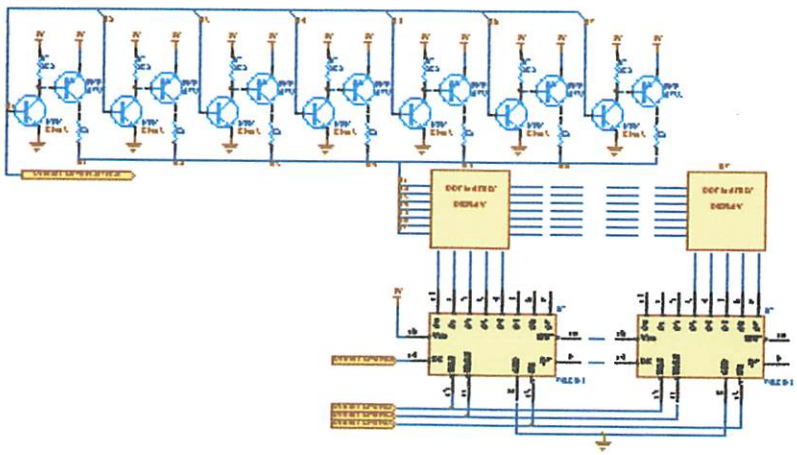
Rangkaian register geser digunakan untuk menggeser data masukan dari mikrokontroler yang berfungsi sebagai scan kolom dalam LED dot matriks. Masukan register geser SN74HC595 (A, B, dan *clock*) berasal dari port 3.3 dan *port* 3.4. Data yang keluar dari keluaran paralel register geser digunakan dalam proses *scanning* untuk pergeseran tampilan per kolom. Berikut ini gambar rangkaian dari register geser yang dirancang untuk m*enscan* kolom LED dot matriks.



Gambar 3-6. Rangkaian Register Geser Untuk *Scan* Kolom

3.2.5.2. Perancangan Rangkaian Tampilan dan Rangkaian *Driver*

Tampilan yang digunakan adalah susunan dari modul-modul LED dot matriks 5x7 sebanyak 8 modul yang nantinya dipasang di setiap perempatan jalan sebanyak 2 modul setiap jalurnya. Dalam setiap modul dot matriks anoda LED dihubungkan ke VCC melalui transistor dan tahanan 82Ω dalam katoda LED dihubungkan pada keluaran IC SN74HC595 yang berfungsi sebagai *scan* kolom. Gambar 3-7 memperlihatkan rangkaian tampilan dot matriks beserta *driver*nya.



Gambar 3-7. Rangkaian Tampilan dan *Driver*

Dalam perencanaan rangkaian tampilan diperlukan rangkaian driver yang dihubungkan ke port 0.0 – port 0.3, dikarenakan arus yang keluar dari *port* mikrokontroler kecil yaitu 1,6 mA saat logika rendah. Transistor yang digunakan untuk *driver* tampilan Port 1.0 – port 1.6. dot matriks (suplay arus dalam LED dot matriks) adalah transistor jenis PNP tipe 9013 yang mempunyai nilai $V_{CE(sat)}$ = 0,6 V (*Fairchild Semiconductor, 2000: 1*) dan NPN tipe A733 yang mempunyai nilai $V_{CE(sat)}$ = 0,18 V (*Fairchild Semiconductor, 2000: 1*).

Nilai $I_{C(sat)} = I_{LED}$ = 15 mA, V_{LED} = 1,7 V (Ledtech Electronics, 2002:1) sehingga:

$$V_{EE} - V_{CE(sat)} - I_{C(sat)}.R_C - V_{LED} = 0$$

$$R_C = \frac{V_{EE} - V_{CE(sat)} - V_{LED}}{I_{C(sat)}} = \frac{(5-0,6-1,7)V}{15mA} = 180\Omega$$

Perencanaan nilai menggunakan nilai β_{dc} minimal untuk menjamin I_C saturasi. Karena dengan nilai β_{dc} minimal, maka arus $I_{B(sat)}$ yang diperoleh adalah besar dan jika terjadi perubahan β_{dc} maka arus I_C akan tetap dalam kondisi saturasi. $I_{C(sat)}$ merupakan harga maksimum dari arus kolektor Sesuai *data sheet* transistor 9013 nilai β_{dc} minimal adalah 64. Arus $I_{C(Sat)} = I_{LED}$ sebesar 15 mA, sehingga dapat dicari nilai R_B sebesar :

$$I_{B(sat)} = \frac{I_{C(sat)}}{\beta_{dc}} = \frac{15\,mA}{64} = 0,234375\,mA$$

$$V_{EE} - V_{BE} - V_{OL} - I_E . R_E = 0$$

$$R_E = \frac{V_{EE} - V_{BE} - V_{OL}}{I_E} = \frac{(5-1.2-0,45)V}{50mA} = 67\Omega$$

Dalam perhitungan diatas diperoleh nilai $R_C = 180\Omega$ dan $R_E = 67\Omega$ karena nilai tersebut tidak ada dipasaran maka digunakan nilai R_E sebesar 82 Ω.

Nilai R_B yang digunakan dalam perancangan adalah R_E 67 Ω, maka dengan persamaan nilai ini arus basis dapat dihitung kembali.

$$I_{B(sat)} = \frac{V_{EE} - V_{BE} - V_{OL}}{R_E} = \frac{(5-1,2-0,45)V}{67\Omega} = 0,05mA$$

Harga $I_{B(sat)}$ = 0,05 mA ini tidak melebihi dari arus maksimum yang dapat diterima oleh *port* 2 mikrokontroler saat aktif rendah yaitu sebesar 1,6 mA.

Dengan menggunakan persamaan dan nilai R_C sebesar 180Ω, maka diperoleh nilai I_{LED} yang baru sebesar :

$$I_{LED} = \frac{5-(0,18+1,7)V}{180\Omega} = 17.3mA$$

Sehingga nilai I_{LED} = 17.3 mA dengan R_C = 180Ω melebihi I_{LED} tipikal yaitu sebesar 15 mA.

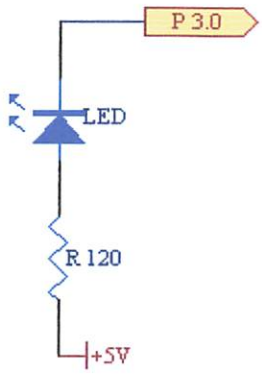
3.2.6. Perancangan Rangkaian *Traffic Light*

Dalam rangkaian indikator menggunakan LED yang diseri dengan sebuah resistor yang berfungsi untuk membatasi arus yang melalui LED sebesar 15 mA, sehingga R_S dapat diketahui :

$$R_S = \frac{V_{CC} - V_{LED} - V_{OI}}{I_{LED}} = \frac{(5-2{,}1-0{,}45)\,V}{15\,mA} = 163{,}33\,\Omega$$

Dalam perencanaan R_S yang digunakan sebesar 120Ω sehingga dapat dihitung kembali :

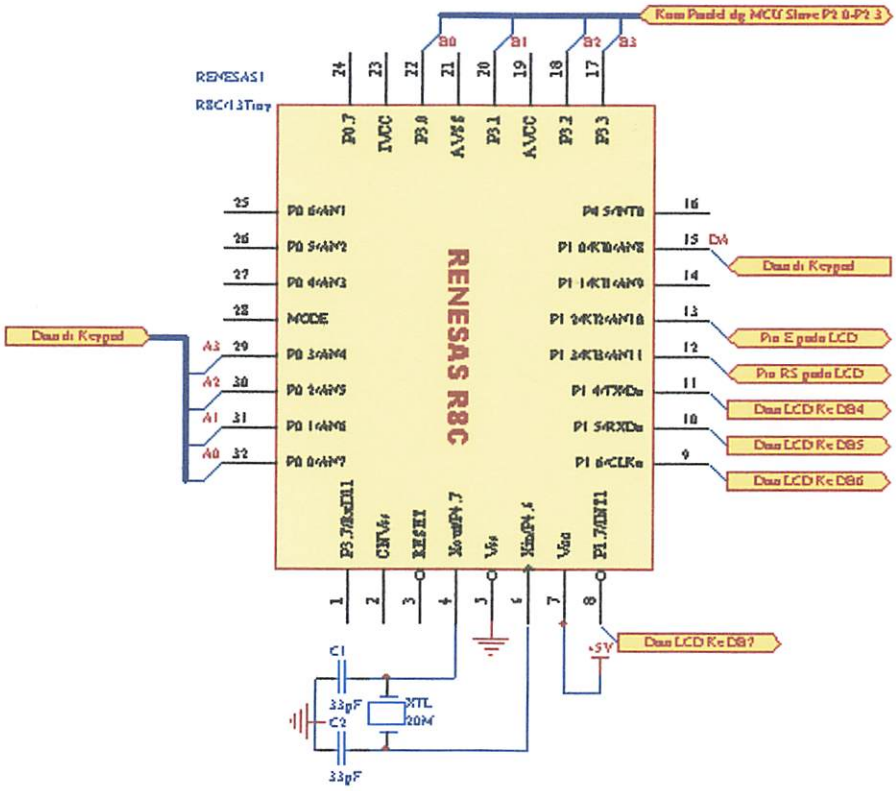
$$I_{LED} = \frac{(5-2{,}1-0{,}45)\,V}{120\,\Omega} = 20.42\,mA$$



Gambar 3-8. Rangkaian *Traffic Light*

Gambar 3-8 memperlihatkan rangkaian *Traffic Light* yang dihubungkan dengan mikrokontroler.

3.2.7. *Perancangan Minimum System* Mikrokontroler Master Renesas R8C Tiny R5F21134FP.



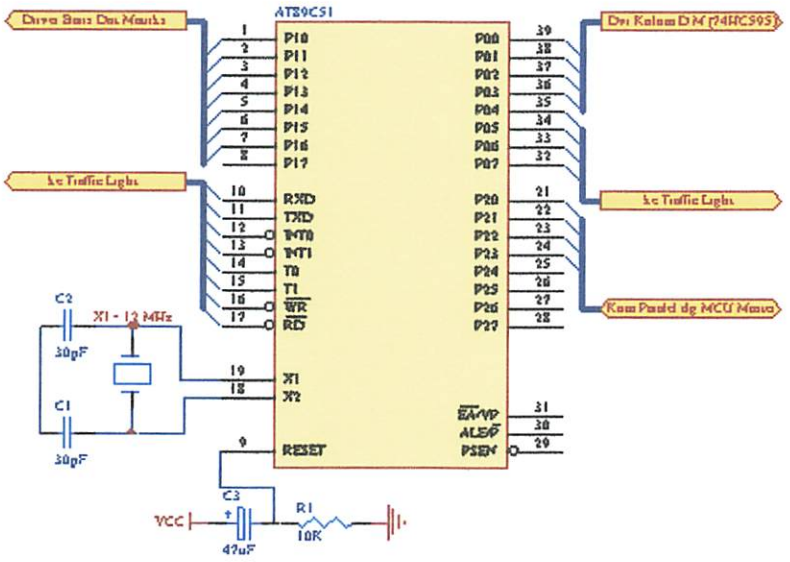
Gambar 3-9. *Minimum System* Mikrokontroler Renesas R8C Tiny R5F21134FP.

Mikrokontroler *Master* merupakan mikrokontroler pengendali utama. Dalam hal ini menggunakan mikrokontroler Renesas R8C Tiny R5F21134FP karena banyak dengan mempertimbangkan keunggulan-keunggulannya dan fasilitas-fasilitas yang dimiliki IC ini. Pembahasan ini telah dijelaskan pada bab-bab sebelumnya. Mikrokontroler *master* ini mempunyai I/O Port yaitu P0.0 – P0.7, P1.0 – P1.7, P3.0 – P3.3, P3.7 dan P4.5 sedangkan P4.6, P4.7 hanya bisa digunakan sebagai *input* saja, bila konfigurasi kristal memakai kristal *internal*. Dalam hal ini yang digunakan Port I/O saja. Berikut adalah konfigurasi pin-pin mikrokontroler *master* :

- P0.0 s.d. P0.3 digunakan sebagai masukan keypad untuk memberikan data berapa lama nyala lampu traffic light.
- P1.0 digunakan sebagai masukan keypad untuk memberikan data berapa lama nyala lampu traffic light.
- P1.4 s.d P1.7 digunakan sebagai pengiriman data pada LCD.
- P1.2 digunakan untuk E pada LCD.
- P1.3 digunakan untuk RS pada LCD
- P3.0 s.d P3.3 digunakan untuk komunikasi pararel ke mikrokontroler slave sebagai pengiriman data.

3.2.8. Perancangan Minimum System Mikrokontroler *Slave.*

Mikrokontroler *Slave* menggunakan mikrokontroler AT89C51. Mikrokontroler ini menerima pulsa *trigger* naik dan turun dari mikrokontroler *master* untuk ditampilkan pada deretan dot matrik led. Mikrokontroler ini juga memberikan pulsa balsan *(akcnowledge)* ke mikrokontroler *master* sebagai tanda bahwa penulisan pada deretan dot matrik telah diproses. Gambar berikut adalah konfigurasi pin-pin yang terpakai :



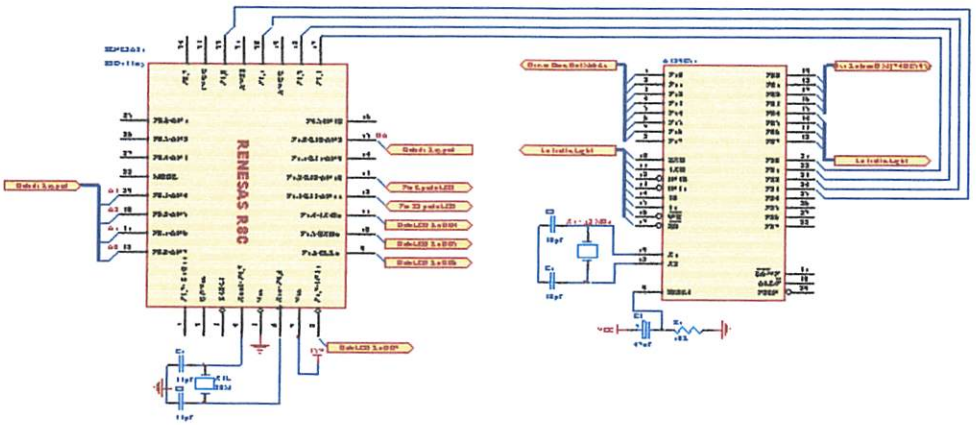
Gambar 3-10. *Minimum System* Mikrokontroler *Slave* AT89C51

- P1.0 s.d. P1.6 digunakan untuk pengontrol driver tampilan pada *dot matriks.*
- P0.4 s.d. P0.7 digunakan untuk pengontrol tampilan pada *traffic light*.
- RXD digunakan untuk pengontrol tampilan pada *traffic ligh.*
- TXD digunakan untuk pengontrol tampilan pada *traffic ligh.*
- INT0 dan INT1digunakan untuk pengontrol tampilan pada *traffic ligh.*
- T0 dan T1 digunakan untuk pengontrol tampilan pada *traffic ligh*
- WR digunakan untuk pengontrol tampilan pada *traffic ligh.*
- RD digunakan untuk pengontrol tampilan pada *traffic ligh.*
- P2.0 s.d. P2.3 digunakan untuk komunikasi pararel dari mikrokontroler master.
- P0.0 s.d. P0.3 digunakan untuk *scan* data dalam masukan IC SN74HC595.

- ❖ X_1 dan X_2 sebagai masukan dari rangkaian osilator kristal. Rangkaian osilator kristal terdiri atas osilator 11,0592 MHZ, kapasitor C_1 dan C_2 yang masing-masing bernilai 30 pF yang akan membangkitkan pulsa *clock* yang digunakan sebagai penggerak bagi sejumlah operasi internal CPU.

3.2.9. Perancangan Komunikasi 2 Mikrokontroler (Master dan Slave)

Dalam perancangan komunikasi 2 mikrokontroler master dan mikrokontroler slave mengunakan komunikasi secara paralel 4 bit karena komunikasi paralel lebih cepat dibandingkan komunikasi serial dalam pengiriman datanya.



Gambar 3-11. Komunikasi paralel antara 2 Mikrokontroler

Pada mikrokontroler master port yang digunakan untuk mengirimkan data pada komunikasi paralel yaitu port 3.0 sampai port 3.3. dan pada mikrokontroler salve untuk menerima data yang dikirim dari mikrokontroler master terhubung pada port 2.0 sampai port 2.3.

Bagian dari program mikrokontroler master merupakan data yang dikirim kemikrokontroler slave.

```
if    (adh2 <= 0x04)
{
   if ( pillih == 0x01)
    {
    Tulis_LCD(0xC0,"TRAFFIC BERKEDIP");
    p3 = 0x05;
      p1_1 = 0;
    }else
   if ( pillih == 0x02)
    {
    Tulis_LCD(0xC0,"Pilihan timer 5d");
    p3 = 0x01;
      p1_1 = 0;
    }else
   if ( pillih == 0x03)
    {
    Tulis_LCD(0xC0,"Pilih timer 10dt");
    p3 = 0x02;
      p1_1 = 0;
    }else
   if ( pillih == 0x04)
    {
    Tulis_LCD(0xC0,"Pilih timer 15dt");
    p3 = 0x03;
      p1_1 = 0;
    }else
   if ( pillih == 0x05)
    {
    Tulis_LCD(0xC0,"Pilih timer 20dt");
    p3 = 0x04;
      p1_1 = 0;
    }
      }else
```

Pada mikrokontroler master data dikirimkan melalui port 3.0 sampai dengan port 3.3. pada saat keypad ditekan angka 2 maka akan menjalankan timer 5 detik dan data yang dikirim dari mikrokontroler master yaitu 01H kemudian diproses ke biner menjadi 0001B.

Bagian dari program mikrokontroler slave.

```
MULAI:

mov   a,p2
anl   a,#0fh              (Meng ANDkn Nilai a dg 0f H)
mov   b,a
cjne  a,#01h,pilih2       (Compare and Jump If Not Equel)
jmp   pilihan_1
```

```
pilih2:
      cjne  a,#02h,pilih3
      jmp   pilihan_2

pilih3:
      cjne  a,#03h,pilih4
      jmp   pilihan_3

pilih4:
      cjne  a,#04h,pilih5
      jmp   pilihan_4

pilih5:
      cjne  a,#05h,pilih2
      jmp   pilihan_5
```

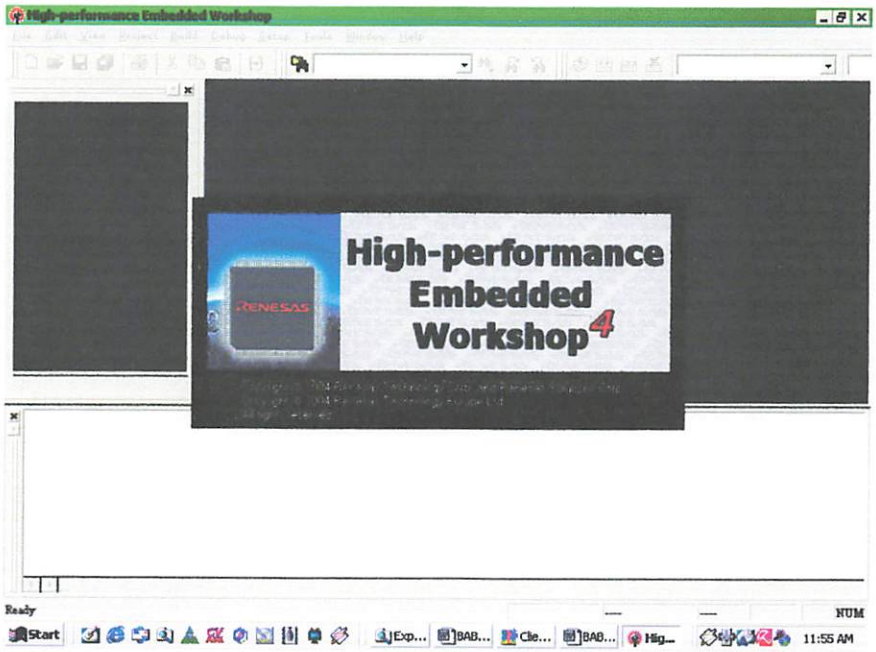
Pada mikrokontroler salve (saat nerima data) ditanyakan apakah data yang dikirim 01H jika benar maka mikrokontroler slave akan menjalankan sistem traffic ligh selama 5 detik, jika tidak mikrokontroler slave mengambil data lagi.

3.3. Perancangan Perangkat Lunak *(Software)*.

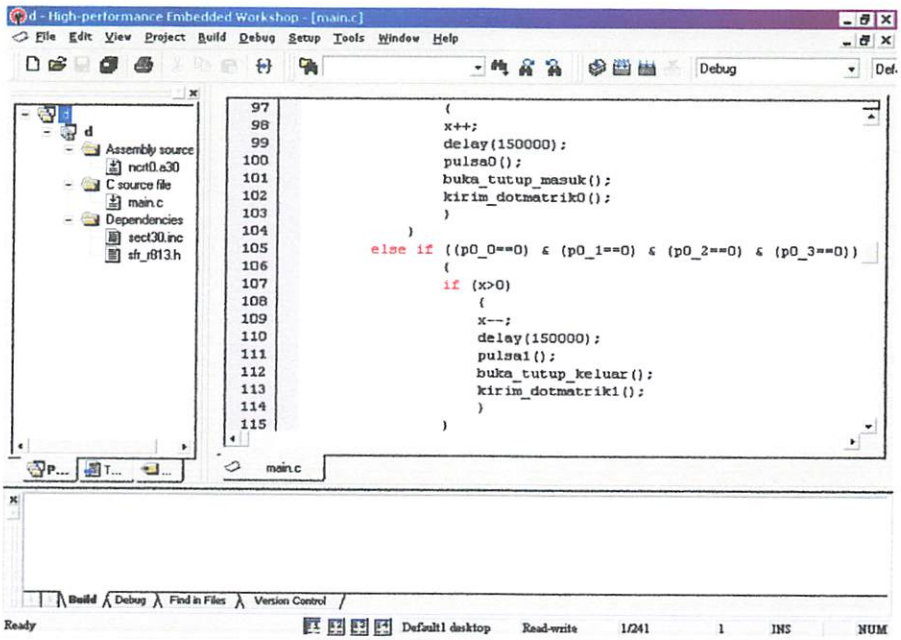
Pada perancangan perangkat lunak ini dipaparkan dalam diagram alir secara keseluruhan sistem namun pada sistem perangkat lunak *Dot Matrik* dijelaskan secara sub program pada *scan* datanya. Untuk bahasa pemrograman dalam hal ini dibagi menjadi 2 bagian karena disini menggunakan 2 mikrokontroler yang berbeda yakni :

1. Mikrokontroler *Master* Renesas R8C/Tiny R5F21134FP menggunakan pemrograman bahasa C dengan *Compiler* yang dipaket bersama pada suatu IDE yaitu HEW *(High-performance Embedded Workshop)*. Fasilitas lainnya yang dibawakan Renesas adalah *software emulator* KD30 dengan menggunakan fasilitas *On-Chip Debugger* R8C yang mempunyai kehandalan mengeliminasi kebutuhan akan *simulator software* dan dapat melakukan *debug* langsung pada *hardware*. *Development Tool Software* ini disediakan *freeware*-nya

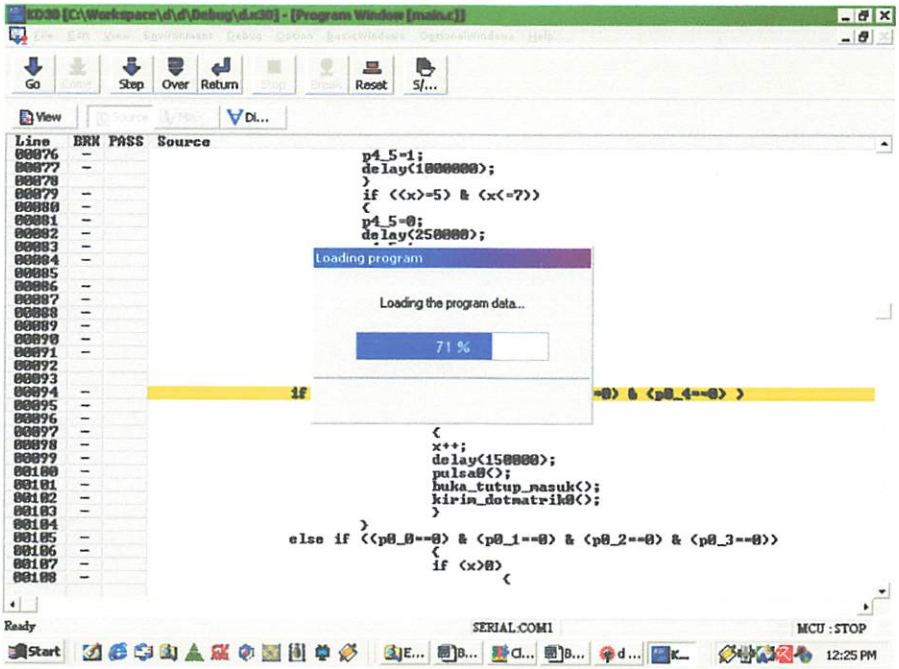
beserta *application-application note*-nya pada situs Renesas. Berikut adalah tampilan *window* IDE EW *(High-performance Embedded Workshop)* dan *software emulator* KD30.



Gambar 3-12. IDE HEW *(High-performance Embedded Workshop)*.

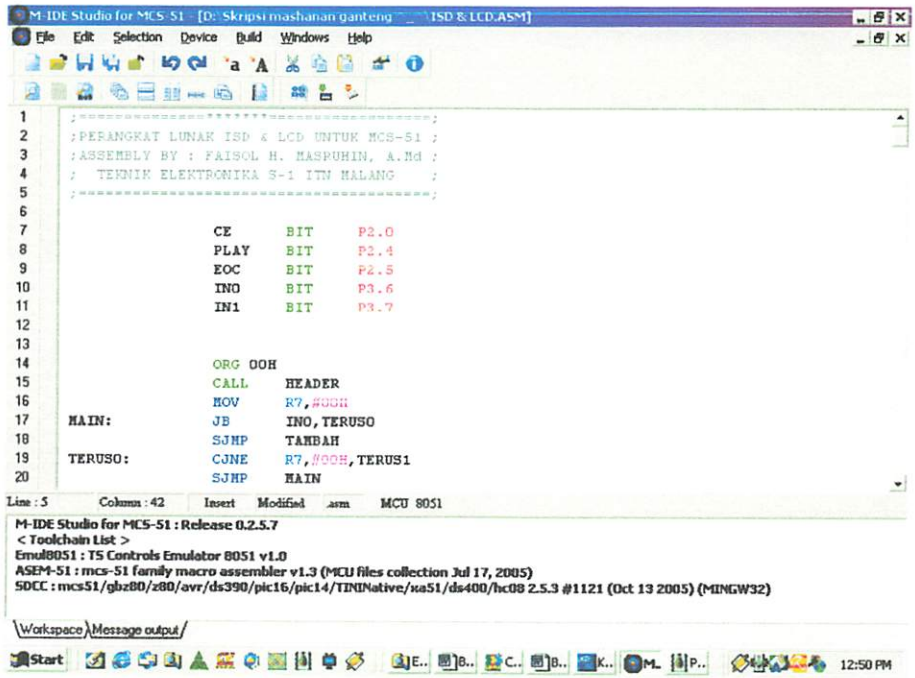


Gambar 3-13. Editor pada HEW *(High-performance Embedded Workshop)*.

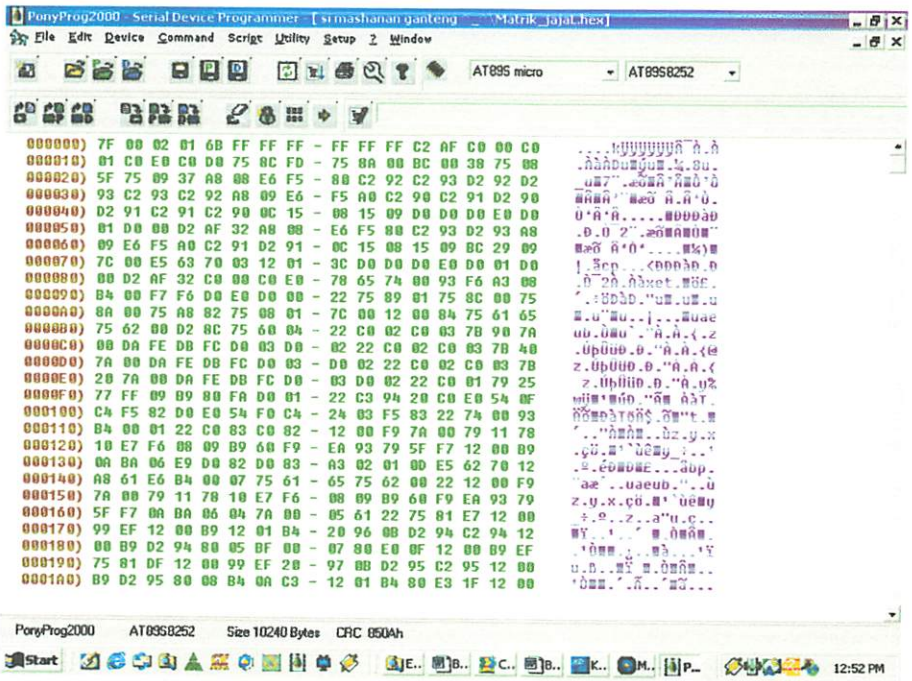


Gambar 3-14. *Programmer dan Emulator KD30*

2. Mikrokontroler *Slave* AT89C51 menggunakan bahasa *assembly* MCS-51 karena kedua mikrokontroler ini adalah masih keluarga MCS-51. Untuk IDE menggunakan *M-IDE Studio for* MCS-51, *compiler* SDCC *(Small Device C Compiler)* dan untuk *software downloading programmer*-nya menggunakan *Pony-Prog 2000 – Serial Device Programmer,* kedua software ini dapat *didownload* di internet secara *freeware*. Berikut adalah tampilan *window* dari kedua *software Development Tool* ini.



Gambar 3-15. *M-IDE Studio*

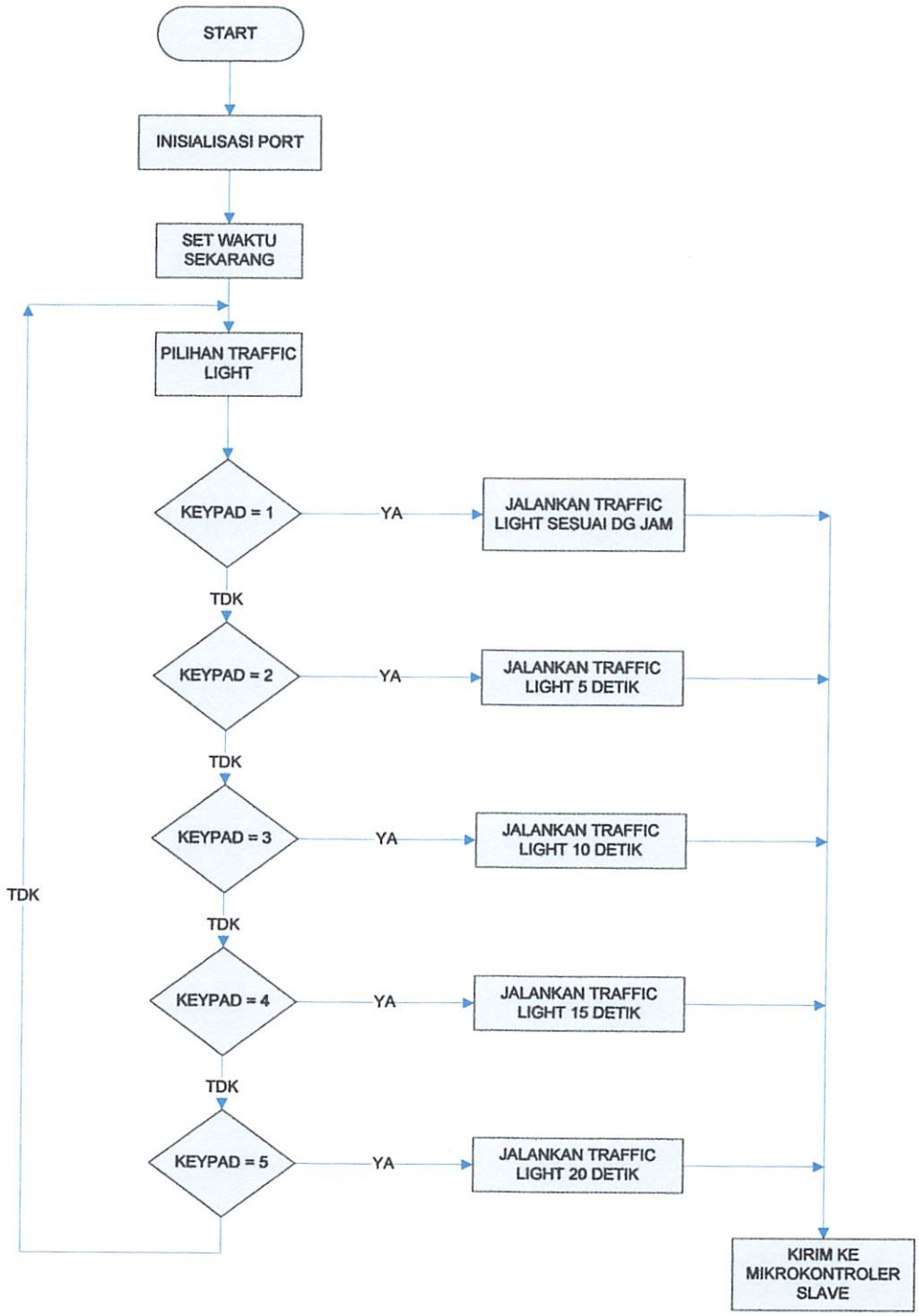


Gambar 3-16. *Pony Prog 2000 Serial Device Programmer*

3.3.1 Diagram Alir (*Flow Chart*) Keseluruhan Sistem

3.3.1.1. Diagram Alir (*Flow Chart*) Pada Mikrokontroler Master

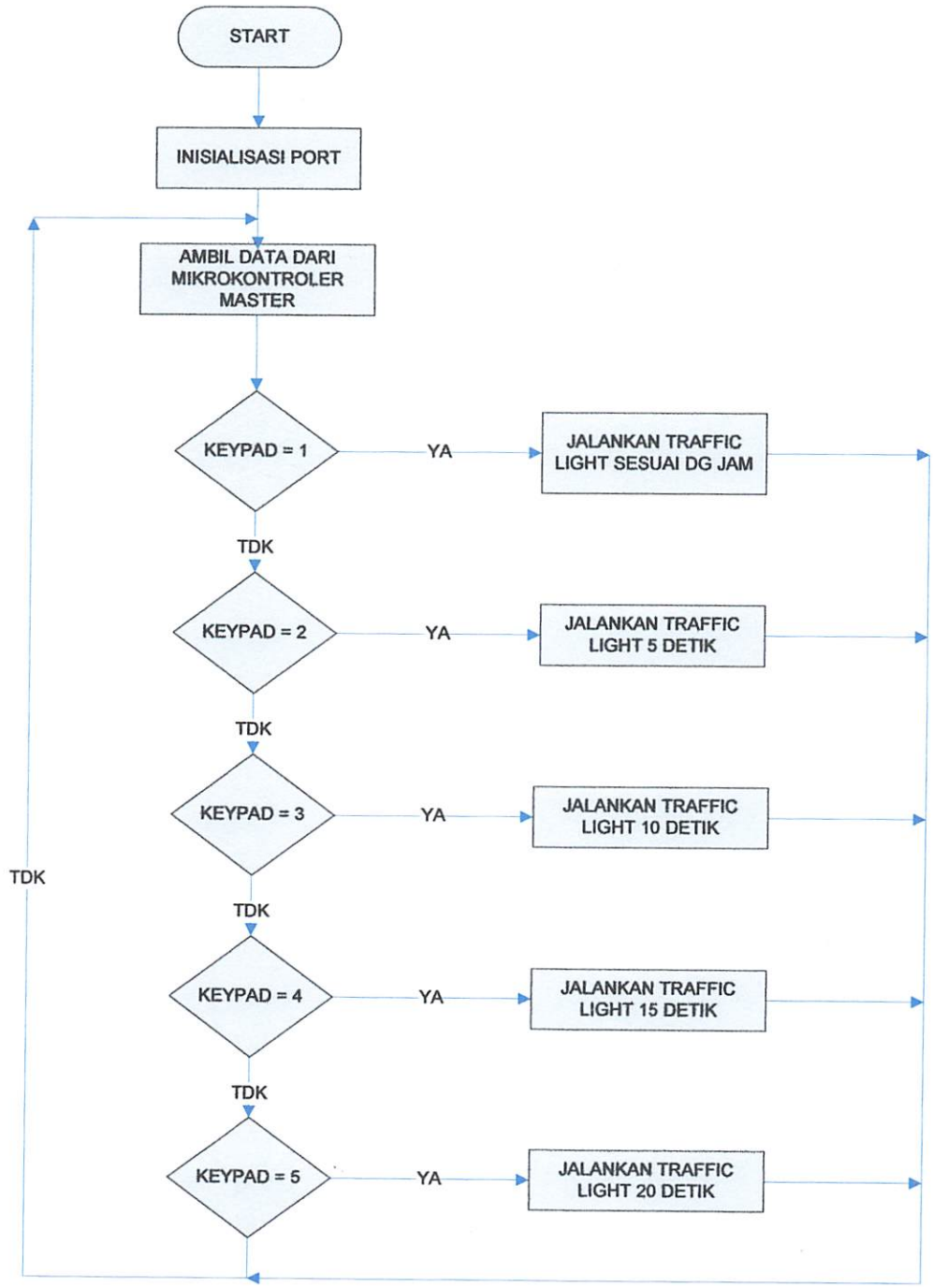
Sub program scan data digunakan untuk menampilkan karakter ke tampilan dot matriks dan traffic light. Sub program ditunjukkan dalam gambar 3-17.



Gambar 3-17. Diagram Alir Sub Program pada Mikrokontroler Master

3.3.1.2. Diagram Alir (*Flow Chart*) Pada Mikrokontroler Slave

Sub program scan data digunakan untuk menampilkan karakter ke tampilan dot matriks dan traffic light. Sub program ditunjukkan dalam gambar 3-18.



Gambar 3-18. Diagram Alir Sub Program pada Mikrokontroler Master

BAB IV

PENGUJIAN ALAT

Untuk mendapatkan hasil yang maksimal setelah melaksanakan perancangan dan pembuatan alat, maka perlu dilakukan suatu pengujian terhadap alat yang telah dibuat. Pengujian ini bertujuan untuk mengetahui apakah alat yang telah dibuat dapat bekerja sesuai yang dengan perencanaan.

Bagian-bagian yang diuji dari peralatan ini adalah :

1. Rangkaian *keypad.*
2. Rangkaian Liquid Cristal Display (LCD) .
3. Rangkaian *Traffic Light* (lampu lalulintas).
4. Rangkaian Dot Matrik LED.

4.1. Pengujian rangkaian *keypad.*

4.1.1. Tujuan

Untuk mengetahui hasil konversi bilangan biner ke digital setiap setiap tombol yang ada pada *keypad.* Pada rangkaian keypad yang digunakan adalah matrik keypad 4x4 sehingga ada 16 fasilitas tombol yang tersedia. Tombol yang tersedia adalah 0, 1, 2, 3 pada baris pertama, tombol 4, 5, 6, 7 pada baris ke dua, tombol 8, 9, A, B pada baris ke tiga, serta tombol *, C, D, # pada baris ke empat. Pada masing-masing tombol keypad ini prinsipnya hanya menghubungkan atau menggabungkan tegangan jalur baris dengan tegangan jalur kolom.

4.1.2. Pengujian Keypad.

Peralatan yang digunakan:

1) Avometer

2) Keypad

4.1.3. Langkah pengujian

1) Atur posisi Avometer pada posisi Voltase DC.

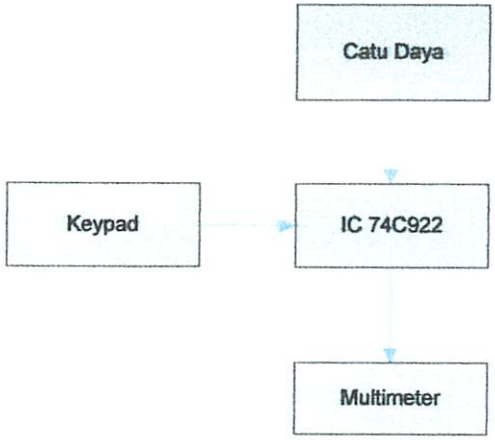
2) Lakukan penekanan pada tombol pada keypad lalu ukur dengan menggunakan avometer untuk pada LED yang nyala (logika 1) dan padam (logika 0).

4.1.4. Hasil pengujian

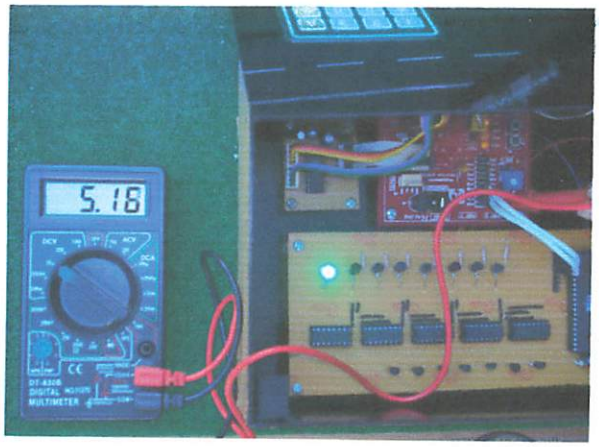
Hasil pengujian keypad ditunjukkan pada tabel dibawah ini.

Tabel 4-1. Data hasil pengujian Keypad

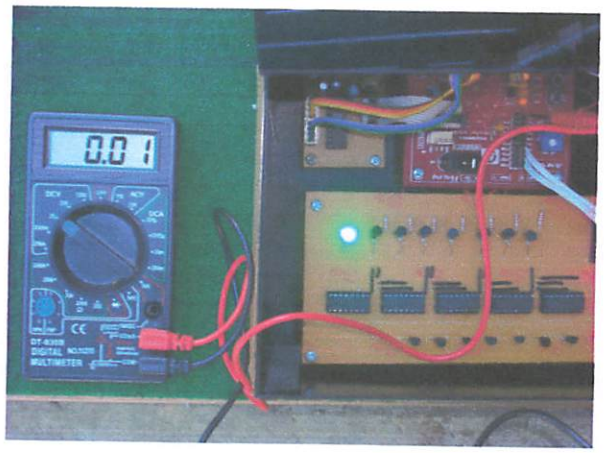
No	Inputan Keypad	Out Put dari IC 74C922				Keterangan (Avometer)
		D0	D1	D2	D3	
1	0	0	0	0	0	
2	1	1	0	0	0	
3	2	0	1	0	0	
4	3	1	1	0	0	
5	4	0	0	1	0	
6	5	1	0	1	0	Tegangan 0 = 0.01 V
7	6	0	1	1	0	
8	7	1	1	1	0	
9	8	0	0	0	1	Tegangan 1 = 5.16 V
10	9	1	0	0	1	
11	A	0	1	0	1	
12	B	1	1	0	1	
13	*	0	0	1	1	
14	C	1	0	1	1	
15	D	0	1	1	1	
16	#	1	1	1	1	



Gambar 4-1. Diagram Blok Pengujian Keypad



Gambar 4-2. Hasil Pengukuran Out Put Rangkaian Keypad High (Logika 1)



Gambar 4-3. Hasil Pengukuran Out Put Rangkaian Keypad Low (Logika 0)

4.1.5. Analisa Pengujian

Keypad yang ditekan akan menghubungkan antara baris dan kolom, yang mempunyai nilai bilangan biner '0'.

Berdasarkan hasil pengujian:

- ✓ Jika baris 1 dan kolom 1 ditekan akan menghasilkan angka 0 dan data yang dikeluarkan dari IC 74C922 adalah bilangan biner 4 bit yaitu 0000
- ✓ Jika baris 1 dan kolom 2 ditekan akan menghasilkan angka 1 dan data yang dikeluarkan dari IC 74C922 adalah bilangan biner 4 bit yaitu 1000
- ✓ Jika baris 1 dan kolom 3 ditekan akan menghasilkan angka 2 dan data yang dikeluarkan dari IC 74C922 adalah bilangan biner 4 bit yaitu 0100
- ✓ Jika baris 1 dan kolom 4 ditekan akan menghasilkan angka 3 dan data yang dikeluarkan dari IC 74C922 adalah bilangan biner 4 bit yaitu 1100
- ✓ Jika baris 2 dan kolom 1 ditekan akan menghasilkan angka 4 dan data yang dikeluarkan dari IC 74C922 adalah bilangan biner 4 bit yaitu 0010
- ✓ Jika baris 2 dan kolom 2 ditekan akan menghasilkan angka 5 dan data yang dikeluarkan dari IC 74C922 adalah bilangan biner 4 bit yaitu 1010
- ✓ Jika baris 2 dan kolom 3 ditekan akan menghasilkan angka 6 dan data yang dikeluarkan dari IC 74C922 adalah bilangan biner 4 bit yaitu 0110
- ✓ Jika baris 2 dan kolom 4 ditekan akan menghasilkan angka 7 dan data yang dikeluarkan dari IC 74C922 adalah bilangan biner 4 bit yaitu 1110
- ✓ Jika baris 3 dan kolom 1 ditekan akan menghasilkan angka 8 dan data yang dikeluarkan dari IC 74C922 adalah bilangan biner 4 bit yaitu 0001
- ✓ Jika baris 3 dan kolom 2 ditekan akan menghasilkan angka 9 dan data yang dikeluarkan dari IC 74C922 adalah bilangan biner 4 bit yaitu 1001

- ✓ Jika baris 3 dan kolom 3 ditekan akan menghasilkan huruf A dan data yang dikeluarkan dari IC 74C922 adalah bilangan biner 4 bit yaitu 0101
- ✓ Jika baris 3 dan kolom 4 ditekan akan menghasilkan huruf B dan data yang dikeluarkan dari IC 74C922 adalah bilangan biner 4 bit yaitu 1101
- ✓ Jika baris 4 dan kolom 1 ditekan akan menghasilkan * dan data yang dikeluarkan dari IC 74C922 adalah bilangan biner 4 bit yaitu 0011
- ✓ Jika baris 4 dan kolom 2 ditekan akan menghasilkan huruf C dan data yang dikeluarkan dari IC 74C922 adalah bilangan biner 4 bit yaitu 1011
- ✓ Jika baris 4 dan kolom 3 ditekan akan menghasilkan huruf D dan data yang dikeluarkan dari IC 74C922 adalah bilangan biner 4 bit yaitu 0111
- ✓ Jika baris 4 dan kolom 4 ditekan akan menghasilkan # dan data yang dikeluarkan dari IC 74C922 adalah bilangan biner 4 bit yaitu 1111

4.2. Pengujian rangkaian LCD (Liquid Cristal Display).

4.2.1. Tujuan

Tujuan pengujian rangkaian LCD ini untuk dapat mengetahui kinerja dari pada LCD tersebut, dimana LCD yang diuji adalah LCD dengan tipe M 162 A yang terdiri dari 16 pin dengan fungsi masing-masing adapun alat yang digunakan adalah sebagai berikut:

4.2.2. Pengujian Rangkaian LCD.

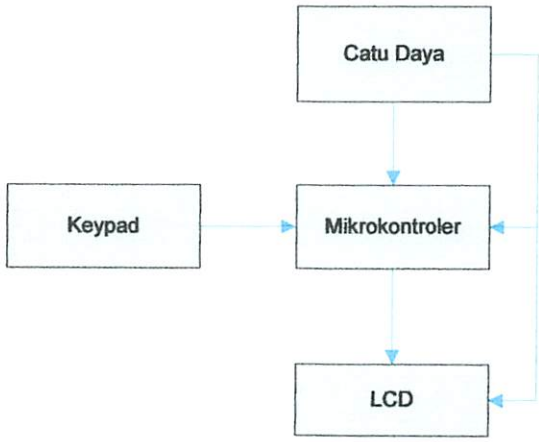
Alat yang digunakan:

1). Avometer

2). LCD

4.2.3. Langkah pengujian

Sesuai dengan perencanaan alat dan dari hasil pengujian yang dilakukan, RS dari LCD dihubungkan Mikrokontroler master diPort 1.3 sementara itu Enable dihubungkan dengan Port 1.2. LCD apabila membaca instruksi maka Port 1.3 harus berlogika 0 (Low) secara program dengan perintah CLR P2.7. Pada saat membaca data dari MCU master maka P1.3 harus berlogika 1 (high) dengan perintah SETB P1.3. Untuk pin Enable dihubungkan dengan Port 1.2 diberikan logika 1 pada saat penulisan atau pembacaan data. Untuk D4 sampai dengan D7 merupakan tempat masuknya data dan instruksi yang diberikan melalui port MCU master yaitu Port 1.4 sampai dengan Port 1.7.

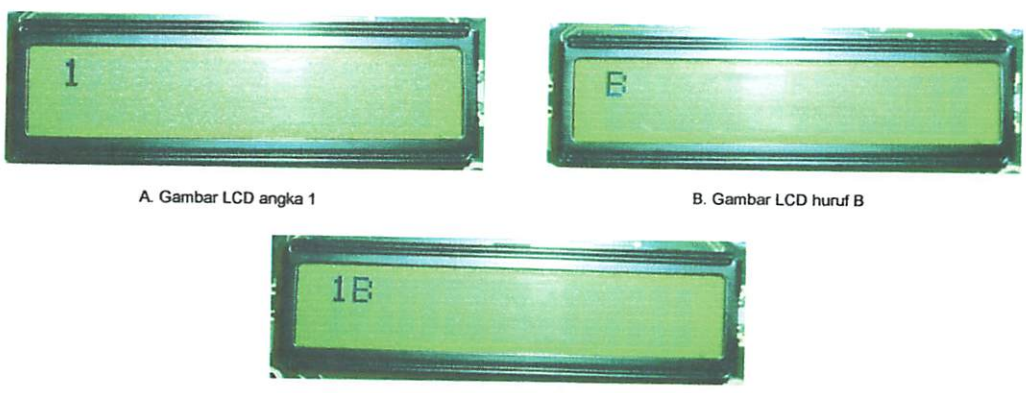


Gambar 4-4. Diagram Blok Pengujian LCD

4.2.4. Hasil Pengujian

Tabel 4-2. Data Hasil Pengujian LCD

Angka/Huruf	D7	D6	D5	D4	D3	D2	D1	D0
1 (UpperBit)	0	0	1	1	*	*	*	*
(LowerBit)	0	0	0	1				
B (UpperBit)	0	1	0	0	*	*	*	*
(LowerBit)	0	0	1	0				



A. Gambar LCD angka 1

B. Gambar LCD huruf B

C. Gambar LCD angka dan huruf

Gambar 4-5. Hasil Pengujian LCD

4.3. Pengujian Rangkaian *Traffic Light*

4.3.1. Tujuan

Untuk mengetahui apakah rangkaian *traffic light* dapat bekerja baik sesuai dengan perancangan.

4.3.2. Peralatan Yang Digunakan

1. Catu Daya 5 V
2. Avometer.
3. Modul Rangkaian *traffic light*

4.3.3. Langkah Pengujian

1. Memberikan catu daya 5 V pada modul rangkaian *Traffic Light.*
2. Atur posisi avometer pada voltase DC dan lakukan pengukuran dimodul rangkaian *taffic ligh.*
3. Lihat hasil pengukuran pada avometer apakah sesuai dengan perancangan atau mendekati perancangan.

4.3.4. Hasil Pengujian

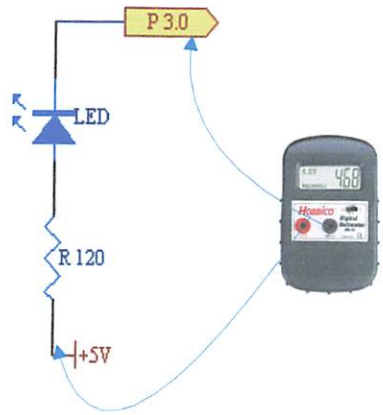
Terlihat pada hasil pengukuran diavometer pada rangkain modul traffic light dengan resistansi 120Ω hasilnya mendekati antara hasil perhitungan perancangan dengan hasil pengukuran pada avometer dengan selisih error yang sangan kecil, itu dipengaruhi oleh hambatan atau resistasi dalam pada LED. Arus minimal untuk menyalakan LED sebesra 15mA dan maksimalnya 60 mA.

$V = I_{LED} * R$

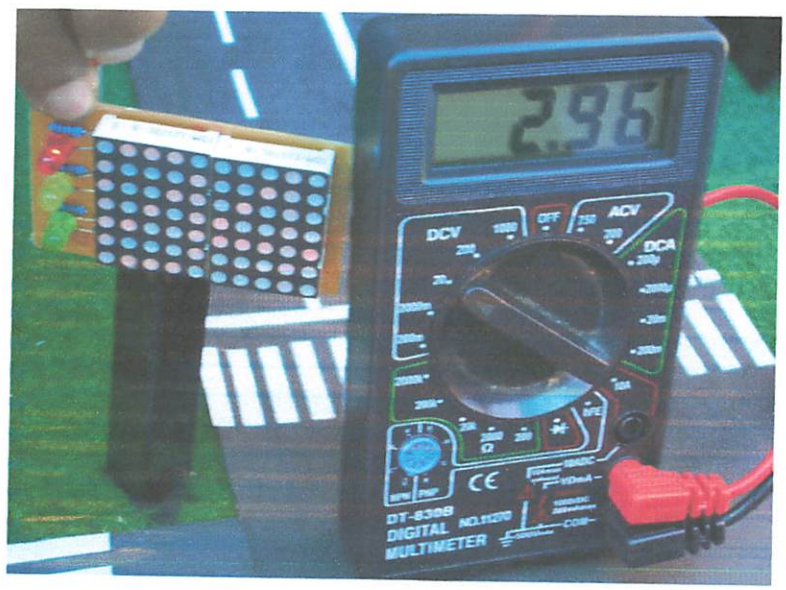
$R = 120\Omega$

$V = 5Volt$

$$I_{LED} = \frac{(5 - 2.1 - 0.45)}{120\Omega} = 20.42mA$$



Gambar 4-6. Rangkaian Pengukuran Traffic Light



Gambar 4-7. Hasil Pengukuran pada Rangkaian *Traffic Light*

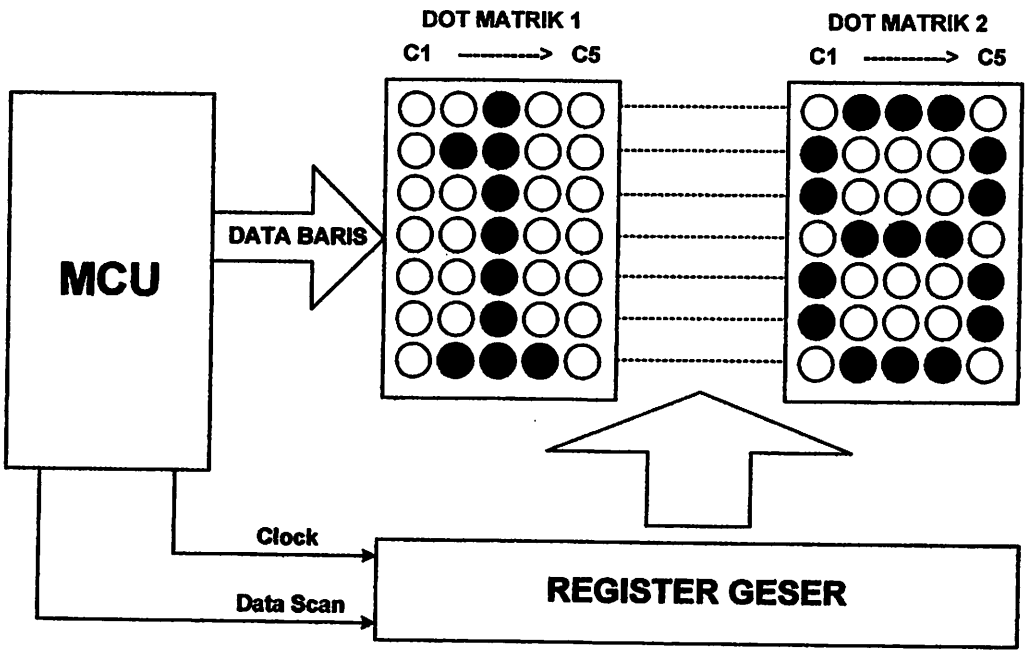
4.4. Pengujian Hubungan Antara Mikrokontroler dengan Dot Matrik Led

4.4.1. Tujuan

Untuk mengetahui apakah LED Dot Matrik dapat bekerja baik sesuai dengan perancangan.

4.4.2. Peralatan Yang Digunakan

1. Catu Daya 5 V
2. Modul Rangkaian Dot Matrik
3. *Programmer* Mikrokontroler MCS-51



Gambar 4-8. Diagram Blok Pengujian Dot Matrik LED 5 x 7

4.4.3. Langkah Pengujian

1. Memberikan catu daya 5 V pada modul rangkaian DOT Matrik LED 5 x 7.

2. Men-*download*-kan program *assembly* Dot Matrik LED seperti yang terlampir dalam lampiran.

4.4.4. Hasil Pengujian

4.4.4.1. Hasil Pengujian Pada Tampilan Dot Matriks

Terlihat bahwa pada 10 bit pertama LED Dot Matrik terus menyala seperti diagram tangga. Selanjutnya pada 10 bit terakhir LED Dot Matrik mati satu per satu dengan clock yang diberikan. Tampilan Dot Matrik ini dikontrol oleh keluaran register geser sebagai scan data dan keluaran Port 1.0 s.d. Port 1.6 mikrokontroler sebagai data masukan. Model scan yang digunakan adalah vertikal *scanning*. Jika data scan untuk kolom diberikan logika 0 dan data baris diberikan logika (untuk mengaktifkan transistor) dan clock diberikan terus menerus, makak tampilan DOT Matrik akan menyala satu per satu. Sebaliknya jika data scan untuk kolom diberikan logika 1 dan data baris diberikan logika 0 kemudian clock diberikan terus menerus, maka tampilan DOT Matrik akan padam 1 per satu.

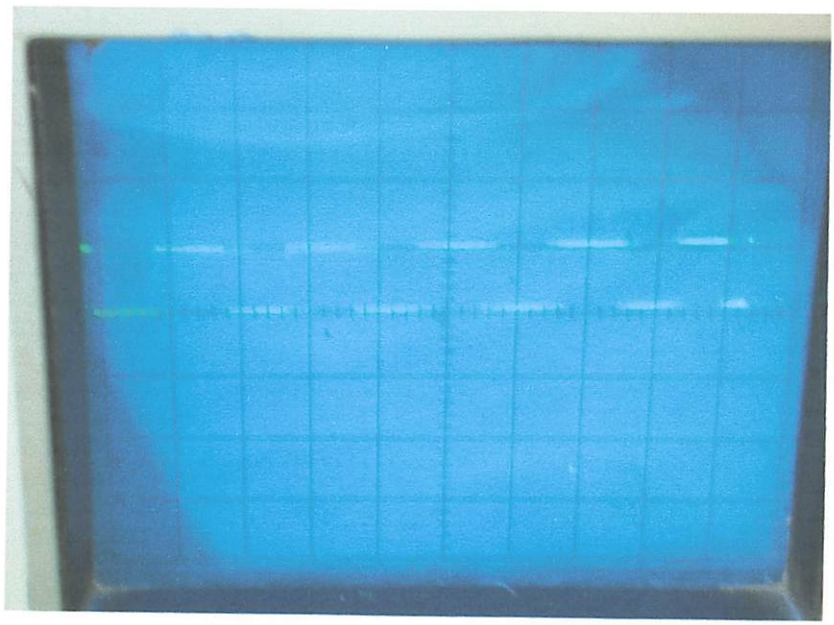
4.4.4.2.Hasil Pengujian pada Driver Kolom

Terlihat pada tampilan osciloscope frekuensi scaning dari driver sangat cepat sehingga data yang dikirim pun juga sangat cepat.

$T / div = 1\mu s$

$V / div = 5V$

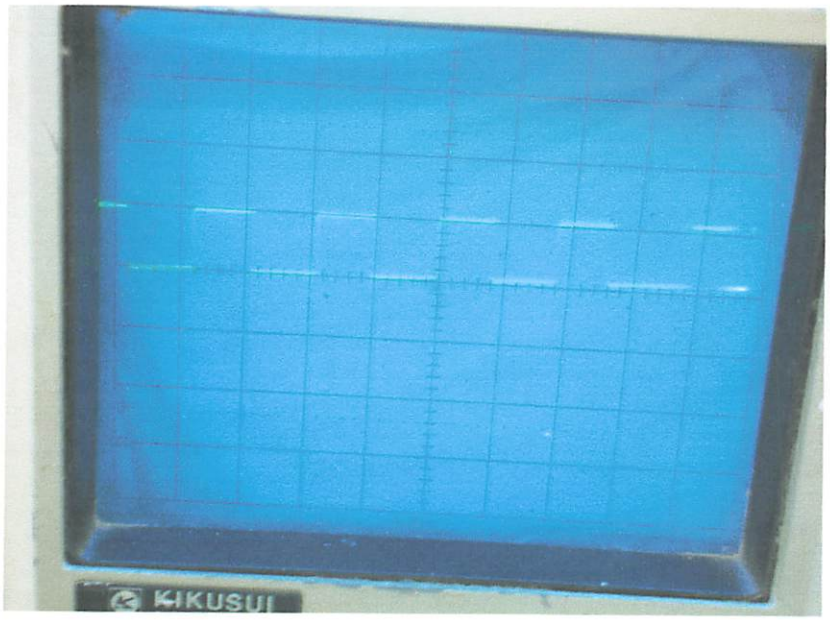
Gambar 4-9. Frekuensi Out Put dari IC74HC595 pada Kaki 1



$T / div = 50\mu s$

$V / div = 5V$

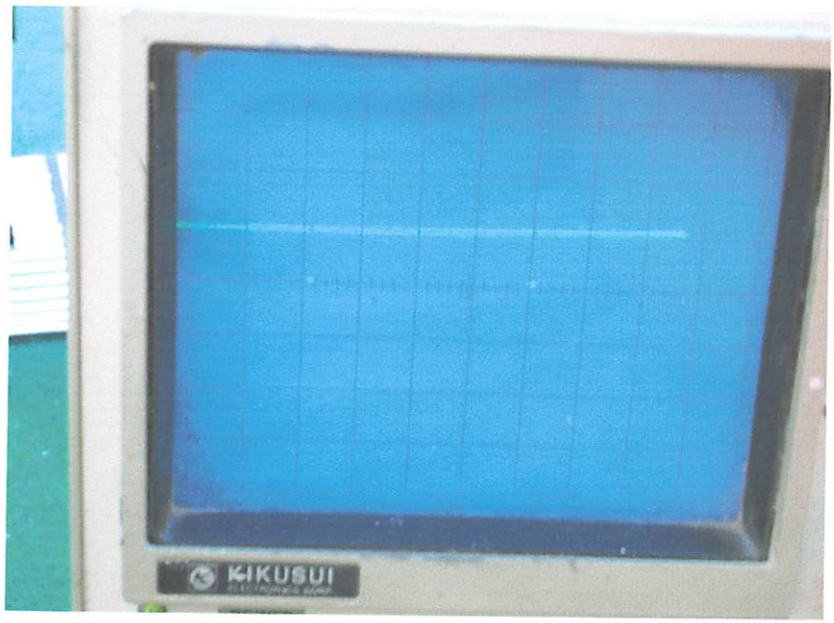
Gambar 4-10. Frekuensi Out Put dari IC74HC595 pada Kaki 11



$T / div = 50\mu s$

$V / div = 5V$

Gambar 4-11. Frekuensi Out Put dari IC74HC595 pada Kaki 12



$T / div = 1\eta s$

$V / div = 5V$

Gambar 4-12. Frekuensi Out Put dari IC74HC595 pada Kaki 14

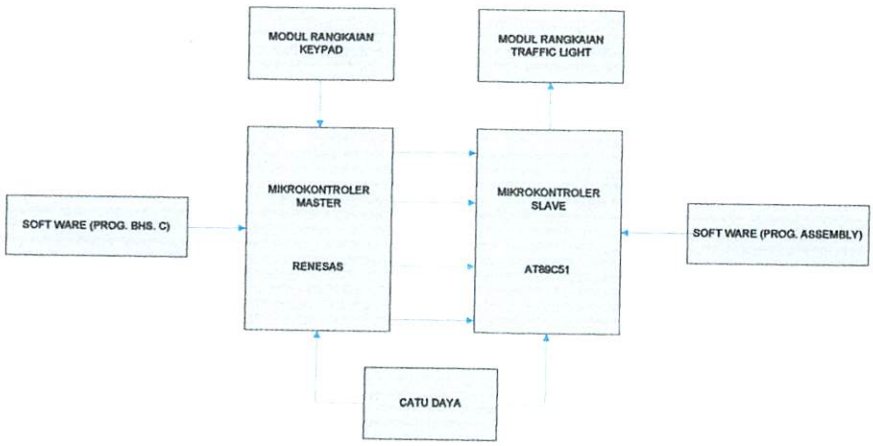
4.5. Pengujian Komunikasi 2 Mikrokontroler (Master Dan Slave) secara Paralel.

4.5.1. Tujuan

Untuk mengetahui apakah rangkaian ke-2 mikrokontroler sudah dapat bekerja baik sesuai dengan perancangan.

4.5.2. Peralatan Yang Digunakan

1. Catu Daya 5 V
2. Modul rangkaian keypad.
3. Modul Rangkaian Mikrokonteroler Master dan Mikrokontroler Slave.
4. Programmer Mikrokontroler Renesas HEW *(High-performance Embedded Workshop)* dan Mikrokontroler AT89C51 MCS-51



Gambar 4-13. Blok Diagram Pengujian Komunikasi 2 Mikrokontroler

4.5.3. Langkah Pengujian

Memberikan catu daya pada kedua mikrokontroler master dan slave kemudian Men-*download*-kan program bahasa C dengan *Compiler* yang dipaket bersama pada suatu IDE yaitu HEW *(High-performance Embedded Workshop)* dan program *assembly* untuk mikrokontroler slave seperti yang

terlampir dalam lampiran. Sesuai dengan perencanaan alat kedua mikrokontroler tersebut dihubungkan, dari port 3.0 mikrokontroler master dihubungkan dengan port 2.3 mikrokontroler slave, port 3.1 mikrokontroler master dihubungkan dengan port 2.2 mikrokontroler slave, port 3.2 mikrokontroler master dihubungkan dengan port 2.1 mikrokontroler slave dan port 3.3 mikrokontroler master dihubungkan dengan port 2.0 mikrokontroler slave.

4.5.4. Hasil Pengujian

Terlihat bahwa pada alat system pengaturan traffic light antara kedua mikrokontroler master dan mikrokontroler salve dapat berkomunikasi secara paralel dengan baik. Data-data yang dikirim dari mikrokontroler master dapat diterima dengan baik dan bisa dijalankan sesuai dengan perancangan.

4.5.5. Analisa Pengujian

Bagian dari program mikrokontroler master merupakan data yang dikirim kemikrokontroler slave.

```
if    (adh2 <= 0x04)
{
   if ( pillih == 0x01)
    {
    Tulis_LCD(0xC0,"TRAFFIC BERKEDIP");
    p3 = 0x05;
      p1_1 = 0;
    }else
   if ( pillih == 0x02)
    {
    Tulis_LCD(0xC0,"Pilihan timer 5d");
    p3 = 0x01;
      p1_1 = 0;
    }else
   if ( pillih == 0x03)
    {
    Tulis_LCD(0xC0,"Pilih timer 10dt");
```

```
 p3 = 0x02;
   p1_1 = 0;
 }else
if ( pillih == 0x04)
 {
 Tulis_LCD(0xC0,"Pilih timer 15dt");
 p3 = 0x03;
   p1_1 = 0;
 }else
if ( pillih == 0x05)
 {
 Tulis_LCD(0xC0,"Pilih timer 20dt");
 p3 = 0x04;
   p1_1 = 0;
 }
   }else
```

Pada mikrokontroler master data dikirimkan melalui port 3.0 sampai dengan port 3.3. pada saat keypad ditekan angka 2 maka akan menjalankan timer 5detik dan data yang dikirim dari mikrokontroler master yaitu 01H kemudian diproses ke biner menjadi 0001B.

Bagian dari program mikrokontroler slave.

```
      MULAI:

      mov   a,p2
      anl   a,#0fh            (Meng ANDkn Nilai a dg 0f H)
      mov   b,a
      cjne  a,#01h,pilih2     (Compare and Jump If Not Equel)
      jmp   pilihan_1

pilih2:
      cjne  a,#02h,pilih3
      jmp   pilihan_2

pilih3:
      cjne  a,#03h,pilih4
      jmp   pilihan_3

pilih4:
      cjne  a,#04h,pilih5
      jmp   pilihan_4

pilih5:
      cjne  a,#05h,pilih2
      jmp   pilihan_5
```

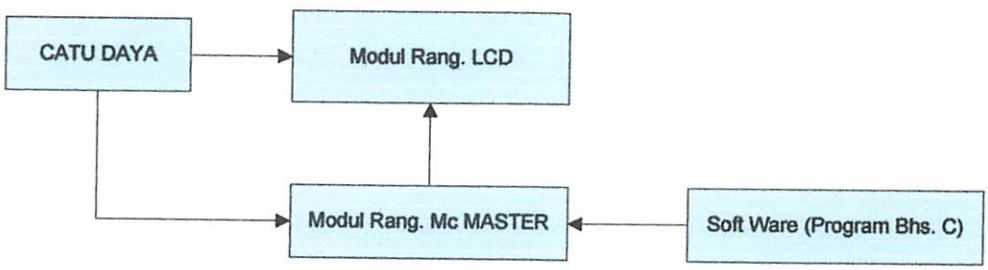
4.6. Pengujian Waktu (diambil dari timernya mikrokontroler Renesas)

4.6.1. Tujuan

Untuk mengetahui apakah jam atau waktu sudah sesuai dengan waktu yang sebenarnya.

4.6.2. Peralatan Yang Digunakan

1. Catu Daya 5 V
2. Modul rangkaian LCD.
3. Modul Rangkaian Mikrokonteroler Master.
4. Programmer Mikrokontroler Renesas HEW *(High-performance Embedded Workshop).*



Gambar 4-14. Blok Diagram Pengujian Timer

4.6.3. Langkah Pengujian

Pada pengujian timer yang dibangkitkan oleh mikrokontroler renesas ini menggunakan delay secara coba-coba (trenen error) untuk mendapatkan satu detik. Memberikan catu daya pada modul rangkaian mikrokontroler master dan modul rangkaian LCD kemudian men- *download*kan program bahasa C dengan *Compiler* yang dipaket bersama pada suatu IDE yaitu HEW *(High-performance Embedded Workshop).*

Bagian dari program bahasa C untuk membangkitkan timer internal mikrokontroler renesas.

```
Tulis_LCD_Byte1(0x8f,adNil5);
adNil5 = adNil5 + 1;
Delay(500000);
if(adNil5==0x0a)
  {adNil5 = 0;
      adNil4 = adNil4 + 1;
      Tulis_LCD_Byte1(0x8f,0x00);
  }else
  if(adNil5<=0x0a)
{
Tulis_LCD_Byte1(0x8f,adNil5);
}
/*if (p1_1 == DATA_0)
{pillih = pillih + 1;
Tulis_LCD_Hex (0x83,pillih);
}*/
if(adNil4==0x06)
  {adNil4 = 0;
  adNil3 = adNil3 + 1;
  Tulis_LCD_Byte1(0x8e,0x00);
  }else
  if(adNil4<=0x05)
{
Tulis_LCD_Byte1(0x8e,adNil4);
}
if(adNil3==0x0a)
  {adNil3 = 0;
      adNil2 = adNil2 + 1;
  Tulis_LCD_Byte1(0x8b,0x00);
  }else
  if(adNil3<=0x0a)
{
Tulis_LCD_Byte1(0x8b,adNil3);
}
if(adNil2==0x06)
  {adNil2 = 0;
  adNil1 = adNil1 + 1;
  Tulis_LCD_Byte1(0x8a,0x00);
  }else
  if(adNil2<=0x06)
{
Tulis_LCD_Byte1(0x8a,adNil2);
}

if(adNil1==0x0a)
  {adNil1 = 0;
      adNil = adNil + 1;
  /*Tulis_LCD_Byte1(0x87,0x00);*/
  }else
  if(adNil==0x02)
  {if(adNil1==0x05)
   {adNil1 = 1;
   adNil = 0;
   /*Tulis_LCD_Byte1(0x87,0x01);
```

```
Tulis_LCD_Byte1(0x86,0x00);*/}
}else
```

4.6.4. Hasil Pengujian

Tabel 4-3. Data Hasil Pengujian Waktu

No	Waktu Mikrokontroler	Detik	Waktu Sebenarnya Stopwatch (Detik)	Error (%)
1	14.00 - 15.00	3600	3602.56	0.071
2	15.30 – 16.30	3600	3602.29	0.064
3	17.00 – 19.00	7200	7205.79	0.08
4	20.00 – 21.30	5400	5404.38	0.081
5	07.00 – 07.30	1800	1801.54	0.085
6	08.00 – 08.25	1500	1501.22	0.081

Terlihat bahwa pada alat system pengaturan traffic light mikrokontroler master dapat menampilkan waktu pada LCD meskipun terdapat selisih atau error. Dari hasil pengujian ini terdapat selisih antara waktu yang dibangkitkan oleh mikrokontroler renesas dengan waktu yang sebenarnya. Selisih antara waktu yang dibangkitkan oleh mikrokontroler renesas dengan waktu yang sebenarnya dapat dilihat dalam tabel 4-3.

$$Error = \frac{Selisih}{WaktuSebenarnya} x100\%$$

$$\overline{X}Error = \frac{JumlahError}{BanyaknyaPercobaan} = \frac{0.462\%}{6} = 0.077\%$$

Jadi antara waktu sebenarnya dan waktu yang dibangkitkan oleh mikrokontroler renesas terdapat Error rata-rata sebesar 0.077%.

BAB V

PENUTUP

5.1. Kesimpulan

Dari analisa dan percobaan yang dilakukan dengan alat ini, maka dapat diambil kesimpulan bahwa :

1. Penggunaan mikrokontroler *master* Renesas R8C sebagai pengendali utama sangatlah tepat, karena dengan tegangan yang tidak stabil dapat beroperasi dengan baik dan tahan terhadap gangguan sinyal elektris lainnya.
2. Pada LCD ditampilkan inputan lama nyala lampu lalulintas dan jam untuk mempermudah seorang pengendara kendaraan bermotor mengetahui waktu sekarang dan juga sebagai inputan otomatis pada mikrokontroler master untuk menjalankan sistem traffic light secara otomatis sesuai dengan jam padat atau tidak.
3. Pada saat inputan dari keypad diproses oleh mikrokontroler master kemudian ditampilkan pada LCD dan data dikirim ke mikrokontroler slave sistem sudah bisa dijalankan pada traffic light sesuai dengan inputan dari keypad dan timer untuk membaca data apakah dalam keadaan padat atau tidak.
4. Informasi lama nyala lampu lalulitas ditampilkan melalui *Dot Matrik Led.*

5. Komunikasi antara mikrokontroler master dan mikrokontroler salve yaitu dengan komunikasi secara paralel 4 bit, konunikasi paralel lebih cepat pengiriman datanya dibandingkan dengan komunikasi secara serial.
6. Waktu yang dibangkitkan oleh mikrokontroler renesas terdapat error rata-rata sebesar 0.077% jika dikalibarasi dengan waktu yang sebenarnya.
7. Alat yang dibuat perlu adanya pengkalibrasian dengan *traffic light* sesungguhnya.

5.2. Saran-saran

Untuk antisipasi pada sistem *taffic light* diMalang, dengan bertambah mahasiswa yang kuliah diMalang otomatis kepadatan kendaran diMalang bertambah dengan pesat. Sering kali terjadi kemacetan pada *taffic light* dari Univ. Brawijaya, ITN I, UMM II dan dari Sumbersari. Jika suatu waktu terjadi kemacetan cuma tinggal mencet tombol *keypad* yang sesuai, sistem *traffic light* akan berubah secara otomatis lama nyala *traffic light*.

DAFTAR PUSTAKA

[1] Hafindo, Pelatihan Microcontroller MCS-51 Programming and Interfacing, Hafindo Electronic & Education, Malang, 2001.

[2] Afianto Eko Putra, Belajar Mikrokontroler At89C51/52/55, Gava Media, Yogyakarta, 2002.

[3] Moh. Ibnu Malik & Anistardi, Bereksperiman dengan Mikrokontroler 8031, PT. Elex Media Komputindo, Jakarta, 1997.

[4] G. Jong Bloed, Elektronika Merencanakan dan Merakit sendiri, Angkasa, Bandung, 1998.

[5] www.innovativeelectronics.com

[6] www.atmel.com

[7] www.Philips Semiconductors.com

[8] www.fairchildsemi.net

[9] www.renesas.com

[10] www.data sheet.com

[11] www.seiko Instruments Inc.com

LAMPIRAN

sPeciaL thank's tO

Aku Persembahkan Karya Tulis Ini Untuk....

Ramanda MOHAMMAD HATTAYANI &

Ibunda SITI ASMANIYAH

Yang Selalu menyayangi, Mendo'akan, Membiayai dan Membesarkan Aku dengan penuh kasih sayang. Hanya Do'a yang masih bisa aku berikan pada beliau mudah-mudahan Allah memberikan kenikmatan di Dunia & Akhirat.

Especially For My honey DJAN UTAMAJ SARJ (cepet selesain ya sekripsinya, jgn plinplan ya? Moga bahagia, eh ya yang teges, gunakan akal sehat untuk menGambil semua kepuTusan, jgn cpt eMosi pikirkan dulu semua resikonya sebelum mengambil kepusan dan jgn mudah menyerah cayo cayo n buktikan kalo DJ biSa)

koNco....temen.....SahAbat.??????????? angkatan BONGEWU LOBO cepet lulus bagi yang lg selesein TA mbek Cepet ndang dapet kerjo. Terakir buat semuanya sopo ae wes yang telah bantu aku sampe mari
sssssssssssssuuuuuuuuwwwwwwuuuuuuunnn!!!!!!!!

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Karanglo KM. 2 Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Elektronika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama	: Mohammad Kurniawan
NIM	: 02 17 157
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik Elektronika
Masa Bimbingan	: 1 Januari 2007 s/d 1 Juli 2007
Judul Skripsi	: Perencanaan Dan Pembuatan Sistem Pengaturan Lampu Lalulintas 4 Jalur Dilengkapi Keypad dan Dot Matriks Berbasis Mikrokontroler Renesas R8C/13 Tiny dan AT89C51.

Tanggal	Uraian	Paraf
17 Maret 2007	❖ Pengujian dan Kesimpulan ❖ Visualisasi Waktu	
	❖ Keypad dan Foto LCD ❖ Komunikasi Antara 2 Mikrokontroler ❖ Pengujian Waktu	

Mengetahui,

Dosen Pembimbing

(Ir. Widodo Pudji M., MT)
NIP. P. 1028700171

Dosen Penguji,

(Joseph Dedy I., ST. MT.)
NIP. 132 315 128

(Moh. Ashar, ST. MT.)
NIP.

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

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

NAMA :
N I M :
Perbaikan meliputi :

1) REV PAD
2) FOTO (CD)
3) KOMUNIKASI ANTAR 2 RTC. → BAB III
4) analisa perhitungan waktu dgn nilai.
5) PENGUJIAN JAM (WAKTU)

Malang,

( )

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

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

NAMA : Muh. Furniawan.
NIM : 0217157
Perbaikan meliputi :

+ Pengujian + Kesimpulan
+ Visualisasi waktu.

Malang,

(_____)

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Karanglo KM. 2 Malang

LEMBAR BIMBINGAN SKRIPSI

NAMA	: Mohammad Kurniawan
NIM	: 02 17 157
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik Elektronika
Masa Bimbingan	: 1 Januari 2007 s/d 1 Juli 2007
Judul Skripsi	: Perencanaan Dan Pembuatan Sistem Pengaturan Lampu Lalulintas 4 Jalur Dilengkapi Keypad dan Dot Matriks Berbasis Mikrokontroler Renesas R8C/13 Tiny dan AT89C51.
Dosem Pembimbing	: Ir. Widodo Pujdi M., MT.
Telah Dievaluasi	: 79 (Tujuh Puluh Sembilan)

Diperiksa dan Disetujui,
Dosen Pembimbing

(Ir. Widodo Pujdi M., MT)
NIP.P. 1028700171

Mengatahui,
Ketua Jurusan Teknik Elektro S-1

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

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR BIMBINGAN SKRIPSI

Nama : Mohammad Kurniawan

NIM : 02 17 157

Masa Bimbingan : 1 Januari 2007 s/d 1 Juli 2007

Judul Skripsi : Perencanaan dan Pembuatan Sistem Pengaturan Lampu Lalulintas 4 Jalur Dilengkapi *Keypad* dan *Dot Matriks* Berbasis Mikrokontroller Renesas R8C/13 Tiny dan AT89C51.

No	Tanggal	Uraian	Paraf Pembimbing
1	10/02 07	Konsultasi Pendahuluan dan BAB II	
2	12/02 07	BAB II, revisi gambar dan tabel	
3	15/02 07	Konsultasi Bab III	
4	16/02 07	BAB III, revisi gambar, cara kerja rangkaian.	
5	17/02 07	ACC BAB II, Teori Dasar dan BAB III	
6	20/02 07	Konsultasi BAB IV	
7	23/02 07	Revisi BAB IV Pengujian Alat, penambahan tabel uji.	
8	24/02 07	ACC BAB IV, Pengujian Alat.	
9	25/02 07	BAB V Penutup, revisi kesimpulan, *finishing* alat.	

Malang, 2007
Dosen Pembimbing

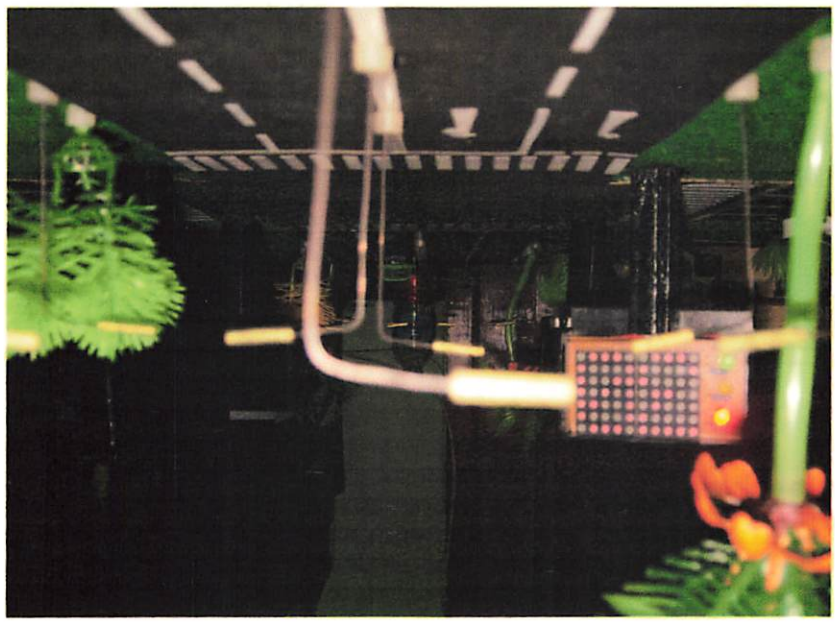
(Ir. Widodo Pujdi M., MT.)
NIP. P. 1028700171

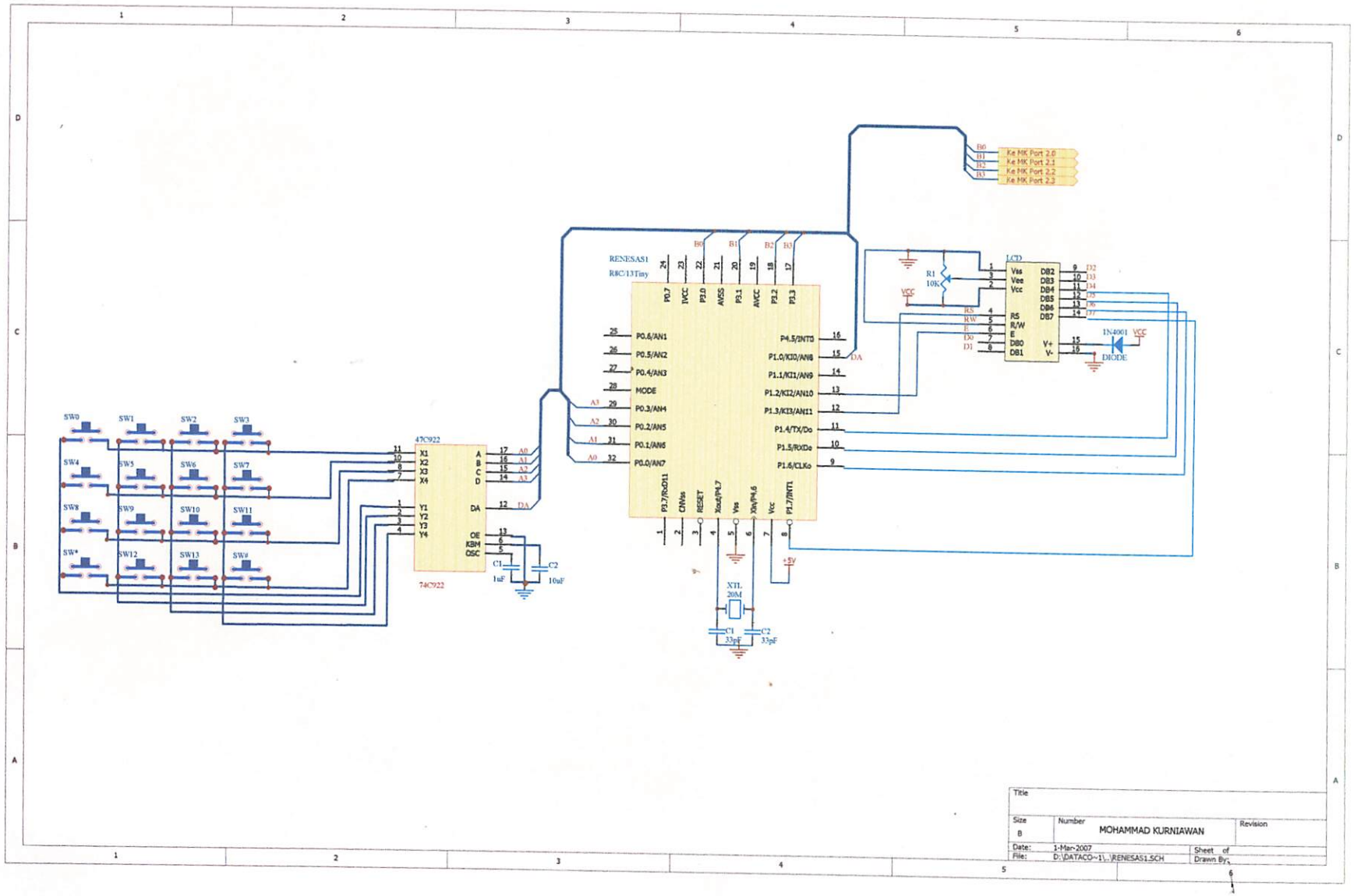
Form S-4a

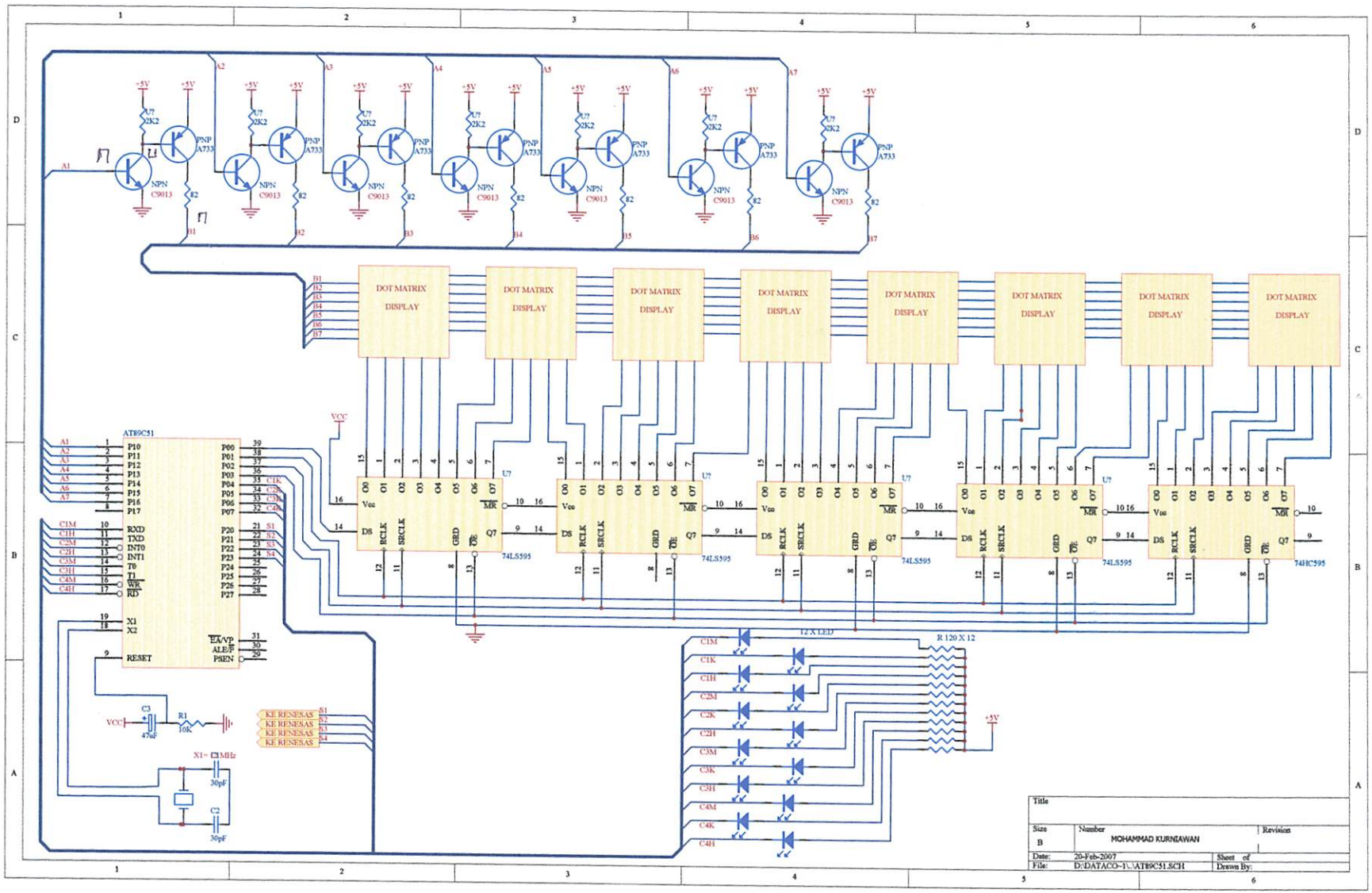
Miniatur Sistem Pengaturan Lampu Lalulintas

Spesifikasi Alat :

1. Input Tegangan : 220 Volt AC /60 Hz
2. Input Tegangan/Arus DC : + 5V/6.5 A.
3. Ukuran :
   - Panjang : 0.92 M
   - Lebar : 0.92 M
   - Tinggi : 0.25 M
4. Input :
   - Keypad
   - Timer
5. Output :
   - Lampu Lalulintas
   - LCD M1632.
   - Dot Matrik Led.




B0
B1
B2
B3
Ke MK Port 2.0
Ke MK Port 2.1
Ke MK Port 2.2
Ke MK Port 2.3
RENESAS1
R8C/13Tiny
P0.7
IVCC
P3.0
AVSS
P3.1
AVCC
P3.2
P3.3
P0.6/AN1
P0.5/AN2
P0.4/AN3
MODE
P0.3/AN4
P0.2/AN5
P0.1/AN6
P0.0/AN7
P4.5/INT0
P1.0/KI0/AN8
P1.1/KI1/AN9
P1.2/KI2/AN10
P1.3/KI3/AN11
P1.4/TX/Do
P1.5/RXDo
P1.6/CLKo
P1.7/INT1
P1.7/RxD11
CNVss
RESET
Xout/P4.7
Vss
Xin/P4.6
Vcc
A3
A2
A1
A0
DA
R1
10K
VCC
LCD
Vss
Vee
Vcc
RS
R/W
E
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
V+
V-
D2
D3
D4
D5
D6
D7
1N4001
DIODE
SW0
SW1
SW2
SW3
SW4
SW5
SW6
SW7
SW8
SW9
SW10
SW11
SW*
SW12
SW13
SW#
47C922
74C922
X1
X2
X3
X4
Y1
Y2
Y3
Y4
A
B
C
D
DA
OE
KBM
OSC
C1
1uF
C2
10uF
XTL
20M
C1
33pF
C2
33pF
+5V
Title
Size
B
Number
MOHAMMAD KURNIAWAN
Revision
Date:
1-Mar-2007
Sheet of
File:
D:\DATACO~1\..\RENESAS1.SCH
Drawn By:


DOT MATRIX DISPLAY
74LS595
74HC595
AT89C51
12 X LED
R 120 X 12
MOHAMMAD KURNIAWAN
20-Feb-2007
D:\DATACO~1\...\AT89C51.SCH

```
;*************************************************;
;                SOFTWARE TRAFFIC LIGHT           ;
; (MIKROKONTROLER MASTER RENESAS R8C/13 TINY)     ;
;                        BY                       ;
;              MOHAMMAD KURNIAWAN                 ;
;               MOBILE : 081805083534             ;
;              TEKNIK ELEKTRONIKA S-1             ;
;       INSTITUT TEKNOLOGI NASIONAL MALANG        ;
;*************************************************;


#include   <stdio.h>
#include          "sfr_r813.h"

#define DATA_0 0
#define DATA_1 1
#define LOW    0
#define HIGH   1
#define detik1    0x00
#define detik2    0x01
#define menit1    0x02
#define menit2    0x03
#define jam1      0x04
#define jam2      0x05

/* Definition of port */
#define INT               p1_0
#define INT1      p1_1                    /* Definition of the R8C/13 SFR */
int Konst = 0;
int adNil = 0;
int adNil1 = 0;
int adNil2 = 0;
int adNil3 = 0;
int adNil4 = 0;
int adNil5 = 0;
int adhil = 0;
int adhil1 = 0;
int adhil2 = 0;
int adhil3 = 0;
int adhil4 = 0;
int adhil5 = 0;

int adhu = 0;
int adh1 = 0;
int adh2 = 0;
int adh3 = 0;
int adh4 = 0;
int adh5 = 0;

int waktu = 0;
int pillih = 0;


int Ratusan = 0;
int Puluhan = 0;
int Satuan = 0;
int cout = 0;
char hit=0;
```

```
char hit1=0;


/*******************************************
*  proto-type declaration
*******************************************/
void init()
{
        asm("FCLR I");                          /* Interrupt disable */
        prcr = 1;                               /* Protect off */
        cm13 = 1;                                       /* X-in X-out */
        cm15 = 1;                                       /* XCIN-XCOUT drive capacity
select bit : HIGH */
        cm05 = 0;                                       /* X-in on */
        cm16 = 0;                                       /* Main clock = No division mode
*/
        cm17 = 0;
        cm06 = 0;                                       /* CM16 and CM17 enable */
        asm("nop");
        asm("nop");
        asm("nop");
        asm("nop");
        ocd2 = 0;                                       /* Main clock change */
        prcr = 0;                               /* Protect on */

        p1 = 0xFF;
  pd1 = 0xFE;
        p0 = 0x00;
  pd0 = 0x00;
        p3 = 0xFF;
  pd3 = 0xFF;

        }

void Delay(long tunggu)
{
        while(tunggu--);
}

void LCD_data(char c,char dat)
{
        p1_2 = c;
        if ((dat & 0x80)==0x80) p1_7=1; else p1_7=0;
        if ((dat & 0x40)==0x40) p1_6=1; else p1_6=0;
        if ((dat & 0x20)==0x20) p1_5=1; else p1_5=0;
        if ((dat & 0x10)==0x10) p1_4=1; else p1_4=0;
        p1_3 = 1;                       p1_3 = 0;
        if ((dat & 0x08)==0x08) p1_7=1; else p1_7=0;
        if ((dat & 0x04)==0x04) p1_6=1; else p1_6=0;
        if ((dat & 0x02)==0x02) p1_5=1; else p1_5=0;
        if ((dat & 0x01)==0x01) p1_4=1; else p1_4=0;
        p1_3 = 1;                       p1_3 = 0;
        Delay(100);
}

void Tulis_LCD(char a, char* dat)
{
        char i = 0;
```

```
        LCD_data(0,a);
        while(dat[i] != 0)
        {
                LCD_data(1,dat[i]); i++;
        }
}
void Tulis_LCD_Hex(char posisi,char dat)
{
        unsigned int temp;
        LCD_data(0,posisi);
        temp=((dat>>4)&0x0f)|0x30;
        if (temp> 0x39)temp=temp+7;
        LCD_data(1,temp);
        temp=dat&0x0f|0x30;
        if (temp>0x39)temp=temp+7;
        LCD_data(1,temp);
}
void Tulis_LCD_Byte(char posisi,char dat)
{
        unsigned int temp;
        LCD_data(0,posisi);
        temp=((dat/100)&0x0f)|0x30;
        if (temp> 0x39)temp=temp+7;
        LCD_data(1,temp);
        temp=(((dat%100)/10)&0x0f)|0x30;
        if (temp> 0x39)temp=temp+7;
        LCD_data(1,temp);
        temp=(((dat%100)%10)&0x0f)|0x30;
        if (temp> 0x39)temp=temp+7;
        LCD_data(1,temp);
        }
void Tulis_LCD_Byte2(char posisi,char dat)
{
        unsigned int temp;
        LCD_data(0,posisi);
        temp=((dat/10)& 0x0f)|0x30;
        if (temp> 0x39)temp=temp+7;
        LCD_data(1,temp);
        temp=((dat%10)&0x0f)|0x30;
        if (temp> 0x39)temp=temp+7;
        LCD_data(1,temp);
        }
void Tulis_LCD_Byte1(char posisi,char dat)
{
        unsigned int temp;
        LCD_data(0,posisi);
        temp=((dat)&0x0f)|0x30;
        if (temp> 0x39)temp=temp+7;
        LCD_data(1,temp);
        }

        void posisi0(void)
{
        unsigned int temp1;
        temp1 = p0 & 0x0f;
        adNil = temp1;
        if(temp1>=0x0a)
          {hit=0;
```

```
          }else
         if(temp1>=0x0d)
          {;
          }else
          if(temp1<=0x09)
         {Tulis_LCD_Byte1(0x86,temp1);
         Delay(100000);
         }

}
void posisi1(void)
{
         unsigned int temp6;
         temp6 = p0 & 0x0f ;
         adNil1 = temp6;
                  if(temp6==0x0c)
          {hit=0;
          }else
         if(temp6==0x0f)
          {Tulis_LCD(0xC0,"Pilihan WAKTU   ");
          }else
          if(temp6<=0x0a)
         {Tulis_LCD_Byte1(0x87,temp6);
         Delay(100000);
         }

}
void posisi2(void)
{
         unsigned int temp2;
         temp2 = p0 & 0x0f ;
         adNil2 = temp2;
                  if(temp2==0x0c)
          {hit=0;
          }else
         if(temp2==0x0f)
          {Tulis_LCD(0xC0,"Pilihan WAKTU   ");
          }else
          if(temp2<=0x0a)
         {Tulis_LCD_Byte1(0x8a,temp2);
         Delay(100000);
 }
}
void posisi3(void)
{
         unsigned int temp3;
         temp3 = p0 & 0x0f;
         adNil3 = temp3;
                  if(temp3==0x0c)
          {hit=0;
          }else
         if(temp3==0x0f)
          {Tulis_LCD(0xC0,"Pilihan WAKTU  ");
          }else
          if(temp3<=0x0a)
         {Tulis_LCD_Byte1(0x8b,temp3);
         Delay(100000);
         }
```

```
}
void posisi4(void)
{
        unsigned int temp7;
        temp7 =p0 & 0x0f;
        adNil4 = temp7;
                if(temp7==0x0c)
         {hit=0;
         }else
        if(temp7==0x0f)
         {Tulis_LCD(0xC0,"Pilihan WAKTU   ");
         }else
         if(temp7<=0x0a)
        {Tulis_LCD_Byte1(0x8e,temp7);
        Delay(100000);
        }

}
void posisi5(void)
{

        unsigned int temp8;
        temp8 = p0 & 0x0f;
        adNil5 = temp8;
                if(temp8==0x0c)
         {hit=0;
         }else
        if(temp8==0x0f)
         {Tulis_LCD(0xC0,"Pilihan WAKTU   ");
         }else
         if(temp8<=0x0a)
        {Tulis_LCD_Byte1(0x8f,temp8);
        /*adNil4 = ((adNil4<<4)|(adNil4>>4));
        adhil4 = adNil4 & 0xf0;
        adhil5 = adhil4 | adNil5;

        adNil2 = ((adNil2<<4)|(adNil2>>4));
        adhil2 = adNil2 & 0xf0;
        adhil3 = adhil2 | adNil3;

        adNil = ((adNil<<4)|(adNil>>4));
        adhil = adNil & 0xf0;
        adhil1 = adhil | adNil1;*/
        Delay(100000);
}
        }


void posisi6(void)

{  unsigned int temp11;
        Tulis_LCD(0xC0,"PERINTAH PILIH M");
        temp11 = p0 & 0x0f;
        if(temp11 <= 0x06)
         {pillih = temp11;
         /*Tulis_LCD_Hex (0x83,pillih);*/
         p1_1 = 1;
```

```
            hit=7;
             }else
           if(temp11>=0x0a)
            {hit=0;
            }
Delay(100000);
          }
void posisi7(void)
{

          Tulis_LCD_Byte1(0x8f,adNil5);
          adNil5 = adNil5 + 1;
          Delay(500000);
          if(adNil5==0x0a)
            {adNil5 = 0;
                    adNil4 = adNil4 + 1;
                    Tulis_LCD_Byte1(0x8f,0x00);
            }else
            if(adNil5<=0x0a)
          {
          Tulis_LCD_Byte1(0x8f,adNil5);
          }
          /*if (p1_1 == DATA_0)
          {pillih = pillih + 1;
          Tulis_LCD_Hex (0x83,pillih);
          }*/
          if(adNil4==0x06)
            {adNil4 = 0;
            adNil3 = adNil3 + 1;
            Tulis_LCD_Byte1(0x8e,0x00);
            }else
            if(adNil4<=0x05)
          {
          Tulis_LCD_Byte1(0x8e,adNil4);
          }
          if(adNil3==0x0a)
            {adNil3 = 0;
                    adNil2 = adNil2 + 1;
            Tulis_LCD_Byte1(0x8b,0x00);
            }else
            if(adNil3<=0x0a)
          {
          Tulis_LCD_Byte1(0x8b,adNil3);
          }
          if(adNil2==0x06)
            {adNil2 = 0;
            adNil1 = adNil1 + 1;
            Tulis_LCD_Byte1(0x8a,0x00);
            }else
            if(adNil2<=0x06)
          {
          Tulis_LCD_Byte1(0x8a,adNil2);
          }

          if(adNil1==0x0a)
            {adNil1 = 0;
                    adNil = adNil + 1;
            /*Tulis_LCD_Byte1(0x87,0x00);*/
```

```
  }else
  if(adNil==0x02)
  {if(adNil1==0x05)
   {adNil1 = 1;
   adNil = 0;
   /*Tulis_LCD_Byte1(0x87,0x01);
   Tulis_LCD_Byte1(0x86,0x00);*/}
   }else
  if(adNil1<=0x0a)
{
/*Tulis_LCD_Byte1(0x87,adNil1);*/
}
if(adNil==0x03)
  {adNil = 0;
  /*Tulis_LCD_Byte1(0x86,0x00);*/
  }else
  if(adNil<=0x02)
{

/*Tulis_LCD_Byte1(0x86,adNil);*/

adhu = ((adNil<<4)|(adNil>>4));
adh1 = adhu & 0xf0;
adh2 = adh1 | adNil1;
Tulis_LCD_Hex (0x86,adh2);
/*Tulis_LCD_Hex (0x83,pillih);*/

if        (adh2 <= 0x04)
          {
            if ( pillih == 0x01)
             {
             Tulis_LCD(0xC0,"TRAFFIC BERKEDIP");
             p3 = 0x05;
                   p1_1 = 0;
             }else
            if ( pillih == 0x02)
             {
             Tulis_LCD(0xC0,"Pilihan timer 5d");
             p3 = 0x01;
                   p1_1 = 0;
             }else
            if ( pillih == 0x03)
             {
             Tulis_LCD(0xC0,"Pilih timer 10dt");
             p3 = 0x02;
                   p1_1 = 0;
             }else
            if ( pillih == 0x04)
             {
             Tulis_LCD(0xC0,"Pilih timer 15dt");
             p3 = 0x03;
                   p1_1 = 0;
             }else
            if ( pillih == 0x05)
             {
             Tulis_LCD(0xC0,"Pilih timer 20dt");
             p3 = 0x04;
                   p1_1 = 0;
```

```
        }
            }else

if      (adh2 <= 0x13)
        {
         if ( pillih == 0x01)
          {
          Tulis_LCD(0xC0,"TRAFFIC PADAT   ");
          p3 = 0x02;
                p1_1 = 0;
          }else
         if ( pillih == 0x02)
          {
          Tulis_LCD(0xC0,"Pilihan timer 5d");
          p3 = 0x01;
                p1_1 = 0;
          }else
         if ( pillih == 0x03)
          {
          Tulis_LCD(0xC0,"Pilih timer 10dt");
          p3 = 0x02;
                p1_1 = 0;
          }else
         if ( pillih == 0x04)
          {
          Tulis_LCD(0xC0,"Pilih timer 15dt");
          p3 = 0x03;
                p1_1 = 0;
          }else
         if ( pillih == 0x05)
          {
          Tulis_LCD(0xC0,"Pilih timer 20dt");
          p3 = 0x04;
                p1_1 = 0;
          }
                }else

       if       (adh2 <= 0x16)
       {
         if ( pillih == 0x01)
          {
          Tulis_LCD(0xC0,"TRAFFIC TDK PDT ");
          p3 = 0x04;
                p1_1 = 0;
          }else
         if ( pillih == 0x02)
          {
          Tulis_LCD(0xC0,"Pilihan timer 5d");
          p3 = 0x01;
                p1_1 = 0;
          }else
         if ( pillih == 0x03)
          {
          Tulis_LCD(0xC0,"Pilih timer 10dt");
          p3 = 0x02;
                p1_1 = 0;
          }else
         if ( pillih == 0x04)
```

```
    {
    Tulis_LCD(0xC0,"Pilih timer 15dt");
    p3 = 0x03;
          p1_1 = 0;
    }else
   if ( pillih == 0x05)
    {
    Tulis_LCD(0xC0,"Pilih timer 20dt");
    p3 = 0x04;
          p1_1 = 0;
    }
          }else
 if       (adh2 <= 0x22)
 {
   if ( pillih == 0x01)
    {
    Tulis_LCD(0xC0,"TRAFFIC PADAT   ");
    p3 = 0x02;
          p1_1 = 0;
    }else
   if ( pillih == 0x02)
    {
    Tulis_LCD(0xC0,"Pilihan timer 5d");
    p3 = 0x01;
          p1_1 = 0;
    }else
   if ( pillih == 0x03)
    {
    Tulis_LCD(0xC0,"Pilih timer 10dt");
    p3 = 0x02;
          p1_1 = 0;
    }else
   if ( pillih == 0x04)
    {
    Tulis_LCD(0xC0,"Pilih timer 15dt");
    p3 = 0x03;
          p1_1 = 0;
    }else
   if ( pillih == 0x05)
    {
    Tulis_LCD(0xC0,"Pilih timer 20dt");
    p3 = 0x04;
          p1_1 = 0;
    }
          }else
if       (adh2 <= 0x24)
{
   if ( pillih == 0x01)
    {
    Tulis_LCD(0xC0,"TRAFFIC BERKEDIP");
    p3 = 0x05;
          p1_1 = 0;
    }else
   if ( pillih == 0x02)
    {
    Tulis_LCD(0xC0,"Pilihan timer 5d");
    p3 = 0x01;
          p1_1 = 0;
```

```
                    }else
                    if ( pillih == 0x03)
                    {
                    Tulis_LCD(0xC0,"Pilih timer 10dt");
                    p3 = 0x02;
                          p1_1 = 0;
                    }else
                    if ( pillih == 0x04)
                    {
                    Tulis_LCD(0xC0,"Pilih timer 15dt");
                    p3 = 0x03;
                          p1_1 = 0;
                    }else
                    if ( pillih == 0x05)
                    {
                    Tulis_LCD(0xC0,"Pilih timer 20dt");
                    p3 = 0x04;
                          p1_1 = 0;
                    }
                          }

        }


}
void posisi8(void)
{
        Tulis_LCD(0xC0,"PERINTAH PILIH U");
        Delay(500000);
        hit=6;
        }
```

```
/*************************************************************************
******
Name:       sfr_init
Parameters: None
Returns:    None
Description: Initial setting of SFR registers
*************************************************************************
*****/
void main()
{ unsigned int data1;
  unsigned int data2;
        init();

        Delay(50000);
        LCD_data(0,0x33); LCD_data(0,0x32); LCD_data(0,0x2F);
        LCD_data(0,0x0E); LCD_data(0,0x06); LCD_data(0,0x06);

                Tulis_LCD(0x80," MOH KURNIAWAN  ");
        Tulis_LCD(0xC0," NIM: 02.17.157 ");
        Delay(500000);
        Delay(500000);
        Tulis_LCD(0x80,"  TRAFFIC LIGHT ");
        Tulis_LCD(0xC0,"  ************* ");
        Delay(500000);
        Delay(500000);
        Tulis_LCD(0x80,"WAKTU= :: :: ");
        Tulis_LCD(0xC0,"Pilihan TRAFFIC ");
        Delay(500000);
        Delay(500000);



/*************************************************************************
*************************************************************************
* PERHATIAN!!!!!!!!!!!!!!!!
* Program ini tidak berjalan dengan baik jika Jumper LED di Board HRS8000
* terpasang, tegangan LED akan mempengaruhi masukan ADC
*************************************************************************
*************************************************************************/

while (1)
        {

        if (INT == DATA_1)
                {
                Delay(5);
                if      (INT == DATA_1)
                        hit++;

                        }
                        if      (hit==9)
                                hit=0;
                        switch(hit)
                        {

                        case 0: posisi0();break;
                        case 1: posisi1();break;
                        case 2: posisi2();break;
```

```
                case 3: posisi3();break;
                case 4: posisi4();break;
                case 5: posisi5();break;
                case 6: posisi6();break;
                case 7: posisi7();break;
                case 8: posisi8();break;
                }

            }

}
```

```
;*********************************************;
;            SOFTWARE TRAFFIC LIGHT           ;
;        (MIKROKONTROLER SLAVE AT89C51)       ;
;                     BY                      ;
;            MOHAMMAD KURNIAWAN               ;
;            MOBILE : 081805083534            ;
;            TEKNIK ELEKTRONIKA S-1           ;
;      INSTITUT TEKNOLOGI NASIONAL MALANG     ;
;*********************************************;

flag            equ     20h
head            bit     20h.0
data_r          bit     20h.1
finish          bit     20h.2
kode            Equ     24h
kode1           Equ     25h
kode2           Equ     26h
kode3           Equ     27h
kode4           Equ     28h
kode5           Equ     29h
kode6           Equ     30h
kode7           Equ     31h
data1           Equ     32h
data2           Equ     33h
data3           Equ     34h
data4           Equ     35h
dt1             EQU     36H
dtk             EQU     37H
dt1a            EQU     38H
dtka            EQU     39H
dt1b            EQU     40H
dtkb            EQU     41H
dt1c            EQU     42H
dtkc            EQU     43H

DATAKEY3        EQU     44H

DATA5           EQU     45H
DATA6           EQU     46H
DATA7           EQU     47H
DATA8           EQU     48H
DATA9           EQU     49H
DATA10          EQU     50H
DATA11          EQU     51H
DATA12          EQU     52H
DATA13          EQU     53H
DATA14          EQU     54H
DATA15          EQU     55H
DATA16          EQU     56H
DATA17          EQU     57H
DATA18          EQU     58H
DATA19          EQU     59H
DATA20          EQU     60H
```

```
JMP   MULAI

MULAI:

          mov     a,p2
          anl     a,#0fh
          mov     b,a
          cjne    a,#01h,pilih2
          jmp     pilihan_1
pilih2:
          cjne    a,#02h,pilih3
          jmp     pilihan_2
pilih3:
          cjne    a,#03h,pilih4
          jmp     pilihan_3
pilih4:
          cjne    a,#04h,pilih5
          jmp     pilihan_4
pilih5:
          cjne    a,#05h,pilih2
          jmp     pilihan_5


pilihan_1:

          mov     data10,#00h
          mov     data11,#01h
          mov     data12,#06h
          mov     data13,#08h
          mov     Dtk,#06H
          MOV     DT1,#00H
          mov     Dtka,#06H
          MOV     DT1a,#00H
          mov     Dtkb,#02H
          MOV     DT1b,#01H
          mov     Dtkc,#08H
          MOV     DT1c,#01H
          jmp     terus

pilihan_2:

          mov     data10,#01h
          mov     data11,#03h
          mov     data12,#01h
          mov     data13,#03h
          mov     Dtk,#01H
          MOV     DT1,#01H
          mov     Dtka,#01H
          MOV     DT1a,#01H
          mov     Dtkb,#02H
          MOV     DT1b,#02H
          mov     Dtkc,#03H
          MOV     DT1c,#03H
          jmp     terus
```

```
pilihan_3:

        mov     data10,#01h
        mov     data11,#04h
        mov     data12,#06h
        mov     data13,#08h
        mov     Dtk,#06H
        MOV     DT1,#01H
        mov     Dtka,#06H
        MOV     DT1a,#01H
        mov     Dtkb,#02H
        MOV     DT1b,#03H
        mov     Dtkc,#08H
        MOV     DT1c,#04H
        jmp     terus

pilihan_4:

        mov     data10,#02h
        mov     data11,#06h
        mov     data12,#01h
        mov     data13,#03h
        mov     Dtk,#01H
        MOV     DT1,#02H
        mov     Dtka,#01H
        MOV     DT1a,#02H
        mov     Dtkb,#02H
        MOV     DT1b,#04H
        mov     Dtkc,#03H
        MOV     DT1c,#06H
        jmp     terus

pilihan_5:

        setb    p3.0
        setb    p3.1
        clr     p0.7
        setb    p3.3
        setb    p3.2
        clr     p0.6
        setb    p3.5
        setb    p3.4
        clr     p0.5
        setb    p3.7
        setb    p3.6
        clr     p0.4
        mov     kode,#00h
        mov     kode1,#00h
        mov     data10,#00h
        mov     kode2,#00h
        mov     kode3,#00h
        mov     kode4,#00h
        mov     kode5,#00h
        mov     kode6,#00h
        mov     kode7,#00h
        mov     DATAKEY3,#9
        ajmp    display
put1q:
```

```
        mov     a,p2
        anl     a,#0fh
        mov     b,a
        cjne    a,#05h,mulaiq
        cpl     p0.7
        cpl     p0.6
        cpl     p0.5
        cpl     p0.4
        mov     DATAKEY3,#9
        ajmp    display
mulaiq:
        jmp     mulai
terus:
        mov     data1,#0
        mov     data2,#0
        mov     data3,#0
        mov     data4,#0
        clr     p3.0
        setb    p3.1
        setb    p0.7
        clr     p3.3
        setb    p3.2
        setb    p0.6
        clr     p3.5
        setb    p3.4
        setb    p0.5
        clr     p3.7
        setb    p3.6
        setb    p0.4
        mov     kode,#00h
        mov     kode1,#00h
        mov     data10,#00h
        mov     kode2,#00h
        mov     kode3,#00h
        mov     kode4,#00h
        mov     kode5,#00h
        mov     kode6,#00h
        mov     kode7,#00h
        mov     DATAKEY3,#0
        jmp     put1

hit1:
        mov     a,data1
        inc     a
        mov     data1,a

        cjne    a,#01h,isia
        mov     Dtk,data13
        MOV     DT1,data11
        clr     p3.1
        setb    p3.0
        setb    p0.7
        ret
```

```
isial:
        cjne    a,#02h,isia1
        mov     a,p2
        anl     a,#0fh
        XRL A,B
        JZ CEKMNT
        ajmp    mulai
cekmnt:
        mov     Dtk,data12
        MOV     DT1,data10
        clr     p3.0
        setb    p3.1
        setb    p0.7
        ret
isia1:
        mov     data1,#0
        jmp     hit1
put1:
        mov     DATAKEY3,#9

HITUNG_WAKTU:
        ;PUSH   ACC
        ;CLR    C
        MOV     A,DTK  ;INC DTK
        DEC     A
        DA      A
        MOV     DTK,A
        CJNE    A,#02H,lanjut
        MOV     A,data1
        CJNE    A,#02H,lanjut

        setb    p3.0
        setb    p3.1
        clr     p0.7
lanjut:
        MOV     A,DTK
        CJNE    A,#00H,RETURN_HW
        mov     a,kode
        cjne    a,#0,jalan
        call    hit1
        mov     kode1,#0
        mov     kode1,#0
        mov     kode,#0
        mov     kode,#0
        jmp     put2
jalan:
        jmp     RETURN_HW1


RETURN_HW:
        MOV     A,DT1
        ANL     A,#0FH
        mov     kode,a
        mov     kode,a
        MOV     A,DTK
        ANL     A,#0FH
        mov     kode1,a
        mov     kode1,a
```

```
        jmp     put2

RETURN_HW1:
        MOV     A,DT1
        ANL     A,#0FH
        mov     kode,a
        mov     kode,a
        MOV     A,DTK
        ANL     A,#0FH
        mov     kode1,#0
        mov     kode1,#0
        mov     Dtk,#0aH
        MOV     A,DT1   ;INC DTK
        DEC     A
        DA      A
        MOV     DT1,A
        CJNE    A,#00H,put2
        jmp     put2

;*********************************************************************
hit2:
        mov     a,data2
        inc     a
        mov     data2,a

        cjne    a,#01h,isib
        mov     Dtka,data12
        MOV     DT1a,data10
        clr     p3.2
        setb    p3.3
        setb    p0.6
        ret
isib:
        cjne    a,#02h,isib1
        mov     Dtka,data13
        MOV     DT1a,data11
        clr     p3.3
        setb    p3.2
        setb    p0.6
        ret
isib1:
        mov     data2,#0
        jmp     hit2
put2:

        MOV     A,DTKa;INC DTK
        DEC     A
        DA      A
        MOV     DTKa,A

        CJNE    A,#02H,lanjut1
        MOV     A,data2
        CJNE    A,#01H,lanjut1

        setb    p3.2
        setb    p3.3
        clr     p0.6
lanjut1:
```

```
        MOV   A,DTKa

        CJNE  A,#00H,RETURN_HWa
        mov   a,kode2
        cjne  a,#0,jalana
        call  hit2
        mov   kode3,#0
        mov   kode3,#0
        mov   kode2,#0
        mov   kode2,#0
        jmp   put3
jalana:
        jmp   RETURN_HW1a

RETURN_HWa:
        MOV   A,DT1a
        ANL   A,#0FH
        mov   kode2,a
        mov   kode2,a
        MOV   A,DTKa
        ANL   A,#0FH
        mov   kode3,a
        mov   kode3,a
        jmp   put3

RETURN_HW1a:
        MOV   A,DT1a
        ANL   A,#0FH
        mov   kode2,a
        mov   kode2,a
        MOV   A,DTKa
        ANL   A,#0FH
        mov   kode3,#0
        mov   kode3,#0
        mov   Dtka,#0aH
        MOV   A,DT1a ;INC DTK
        DEC   A
        DA    A
        MOV   DT1a,A
        CJNE  A,#00H,put3
        jmp   put3

;*************************************************************************
hit3:
        mov   a,data3
        inc   a
        mov   data3,a

        cjne  a,#01h,isic
        mov   Dtkb,data12
        MOV   DT1b,data10
        clr   p3.4
        setb  p3.5
        setb  p0.5
        ret
isic:
        cjne  a,#02h,isic1
        mov   Dtkb,data13
```

```
        MOV     DT1b,data11
        clr     p3.5
        setb    p3.4
        setb    p0.5
        ret
isic1:
        mov     data3,#0
        jmp     hit3
put3:

        MOV     A,DTKb;INC DTK
        DEC     A
        DA      A
        MOV     DTKb,A

        CJNE    A,#02H,lanjut2
        MOV     A,data3
        CJNE    A,#01H,lanjut2

        setb    p3.4
        setb    p3.5
        clr     p0.5
lanjut2:
        MOV     A,DTKb
        CJNE    A,#00H,RETURN_HWb
        mov     a,kode4
        cjne    a,#0,jalanb
        call    hit3
        mov     kode5,#0
        mov     kode5,#0
        mov     kode4,#0
        mov     kode4,#0
        jmp     put4
jalanb:
        jmp     RETURN_HW1b

RETURN_HWb:
        MOV     A,DT1b
        ANL     A,#0FH
        mov     kode4,a
        mov     kode4,a
        MOV     A,DTKb
        ANL     A,#0FH
        mov     kode5,a
        mov     kode5,a
        jmp     put4
```

```
RETURN_HW1b:
        MOV     A,DT1b
        ANL     A,#0FH
        mov     kode4,a
        mov     kode4,a
        MOV     A,DTKb
        ANL     A,#0FH
        mov     kode5,#0
        mov     kode5,#0
        mov     Dtkb,#0aH
        MOV     A,DT1b ;INC DTK
        DEC     A
        DA      A
        MOV     DT1b,A
        CJNE    A,#00H,put4
        jmp     put4

;*************************************************************************
;*************************************************************************
hit4:
        mov     a,data4
        inc     a
        mov     data4,a

        cjne    a,#01h,isid
        mov     Dtkc,data12
        MOV     DT1c,data10
        clr     p3.6
        setb    p3.7
        setb    p0.4
        ret
isid:
        cjne    a,#02h,isid1
        mov     Dtkc,data13
        MOV     DT1c,data11
        clr     p3.7
        setb    p3.6
        setb    p0.4
        ret
isid1:
        mov     data4,#0
        jmp     hit4
put4:

        MOV     A,DTKc;INC DTK
        DEC     A
        DA      A
        MOV     DTKc,A

        CJNE    A,#02H,lanjut3
        MOV     A,data4
        CJNE    A,#01H,lanjut3

        setb    p3.6
        setb    p3.7
        clr     p0.4
lanjut3:
        MOV     A,DTKc
```

```
        CJNE    A,#00H,RETURN_HWc
        mov     a,kode6
        cjne    a,#0,jalanc
        call    hit4
        mov     kode7,#0
        mov     kode7,#0
        mov     kode6,#0
        mov     kode6,#0
        jmp     display
jalanc:
        jmp     RETURN_HW1c

RETURN_HWc:
        MOV     A,DT1c
        ANL     A,#0FH
        mov     kode6,a
        mov     kode6,a
        MOV     A,DTKc
        ANL     A,#0FH
        mov     kode7,a
        mov     kode7,a
        jmp     display
RETURN_HW1c:
        MOV     A,DT1c
        ANL     A,#0FH
        mov     kode6,a
        mov     kode6,a
        MOV     A,DTKc
        ANL     A,#0FH
        mov     kode7,#0
        mov     kode7,#0
        mov     Dtkc,#0aH
        MOV     A,DT1c ;INC DTK
        DEC     A
        DA      A
        MOV     DT1c,A
        CJNE    A,#00H,display
        jmp     display

;*************************************************************************

put:
        mov     r0,datakey3
        djnz    r0,display
        mov     a,b
        CJNE    A,#05H,put1a
        jmp     put1q
put1a:
        ajmp    put1
display:
        mov     datakey3,r0
        mov     a,kode
        cjne    a,#0,satuA
        mov     p1,#3eh
        ACALL PULSA
        call    no1
        ajmp    display1
satuA:
```

```
        cjne    a,#1,duaA
        mov     p1,#00h
        ACALL PULSA
        call    satu
        ajmp    display1
duaA:
        cjne    a,#2,tigaA
        mov     p1,#23h
        ACALL PULSA
        call    dua
        JMP     display1
tigaA:
        cjne    a,#3,empatA
        mov     p1,#42h
        ACALL PULSA
        call    tiga
        jmp     display1
empatA:
        cjne    a,#4,limaA
        mov     p1,#0ch
        ACALL PULSA
        call    empat
        jmp     display1
limaA:
        cjne    a,#5,enamA
        mov     p1,#72h
        ACALL PULSA
        call    lima
        jmp     display1
enamA:
        cjne    a,#6,tujuA
        mov     p1,#1eh
        ACALL PULSA
        call    enam
        jmp     display1
tujuA:
        cjne    a,#7,delapanA
        mov     p1,#40h
        ACALL PULSA
        call    tujuh
        jmp     display1
delapanA:
        cjne    a,#8,sembilanA
        mov     p1,#36h
        ACALL PULSA
        call    delapan
        Ajmp    display1
sembilanA:
        cjne    a,#9,display1
        mov     p1,#31h
        ACALL PULSA
        call    sembilan
display1:
        mov     a,kode1
        cjne    a,#0,satu1
        call    nol
        Ajmp    display2
satu1:
```

```
        cjne    a,#1,dua1
        call    satu
        Ajmp    display2
dua1:
        cjne    a,#2,tiga1
        call    dua
        Ajmp    display2
tiga1:
        cjne    a,#3,empat1
        call    tiga
        Ajmp    display2
empat1:
        cjne    a,#4,lima1
        call    empat
        Ajmp    display2
lima1:
        cjne    a,#5,enam1
        call    lima
        Ajmp    display2
enam1:
        cjne    a,#6,tuju1
        call    enam
        Ajmp    display2
tuju1:
        cjne    a,#7,delapan1
        call    tujuh
        Ajmp    display2
delapan1:
        cjne    a,#8,sembilan1
        call    delapan
        Ajmp    display2
sembilan1:
        cjne    a,#9,display2
        call    sembilan

;*************************************1

display2:
        mov     a,kode2
        cjne    a,#00h,satu2
        call    nol
        Ajmp    display3
satu2:
        cjne    a,#01h,dua2
        call    satu
        Ajmp    display3
dua2:
        cjne    a,#02h,tiga2
        call    dua
        Ajmp    display3
tiga2:
        cjne    a,#03h,empat2
        call    tiga
        Ajmp    display3
empat2:
        cjne    a,#04h,lima2
        call    empat
        Ajmp    display3
```

```
lima2:
        cjne    a,#05h,enam2
        call    lima
        Ajmp    display3
enam2:
        cjne    a,#06h,tuju2
        call    enam
        Ajmp    display3
tuju2:
        cjne    a,#07h,delapan2
        call    tujuh
        Ajmp    display3
delapan2:
        cjne    a,#08h,sembilan2
        call    delapan
        Ajmp    display3
sembilan2:
        cjne    a,#09h,display3
        call    sembilan
        Ajmp    display3


display3:
        mov     a,kode3
        cjne    a,#00h,satu3
        call    nol
        Ajmp    display4
satu3:
        cjne    a,#01h,dua3
        call    satu
        Ajmp    display4
dua3:
        cjne    a,#02h,tiga3
        call    dua
        Ajmp    display4
tiga3:
        cjne    a,#03h,empat3
        call    tiga
        Ajmp    display4
empat3:
        cjne    a,#04h,lima3
        call    empat
        Ajmp    display4
lima3:
        cjne    a,#05h,enam3
        call    lima
        Ajmp    display4
```

```
enam3:
        cjne    a,#06h,tuju3
        call    enam
        Ajmp    display4
tuju3:
        cjne    a,#07h,delapan3
        call    tujuh
        Ajmp    display4
delapan3:
        cjne    a,#08h,sembilan3
        call    delapan
        Ajmp    display4
sembilan3:
        cjne    a,#09h,display4
        call    sembilan

;*************************************3
display4:
        mov     p1,#00h
        mov     a,kode4
        cjne    a,#00h,satu4
        call    nol
        Ajmp    display5
satu4:
        cjne    a,#01h,dua4
        call    satu
        Ajmp    display5
dua4:
        cjne    a,#02h,tiga4
        call    dua
        Ajmp    display5
tiga4:
        cjne    a,#03h,empat4
        call    tiga
        Ajmp    display5
empat4:
        cjne    a,#04h,lima4
        call    empat
        Ajmp    display5
lima4:
        cjne    a,#05h,enam4
        call    lima
        Ajmp    display5
enam4:
        cjne    a,#06h,tuju4
        call    enam
        Ajmp    display5
tuju4:
        cjne    a,#07h,delapan4
        call    tujuh
        Ajmp    display5
delapan4:
        cjne    a,#08h,sembilan4
        call    delapan
        Ajmp    display5
```

```
sembilan4:
        cjne    a,#09h,display5
        call    sembilan

display5:
        mov     a,kode5
        cjne    a,#00h,satu5
        call    nol
        jmp     display6
satu5:
        cjne    a,#01h,dua5
        call    satu
        jmp     display6
dua5:
        cjne    a,#02h,tiga5
        call    dua
        jmp     display6
tiga5:
        cjne    a,#03h,empat5
        call    tiga
        jmp     display6
empat5:
        cjne    a,#04h,lima5
        call    empat
        jmp     display6
lima5:
        cjne    a,#05h,enam5
        call    lima
        jmp     display6
enam5:
        cjne    a,#06h,tuju5
        call    enam
        jmp     display6
tuju5:
        cjne    a,#07h,delapan5
        call    tujuh
        jmp     display6
delapan5:
        cjne    a,#08h,sembilan5
        call    delapan
        jmp     display6
sembilan5:
        cjne    a,#09h,display6
        call    sembilan

;*************************************4
display6:
        mov     p1,#00h
        mov     a,kode6
        cjne    a,#00h,satu6
        call    nol
        jmp     display7
satu6:
        cjne    a,#01h,dua6
        call    satu
        ajmp    display7
dua6:
        cjne    a,#02h,tiga6
```

```
        call    dua
        Ajmp    display7
tiga6:
        cjne    a,#03h,empat6
        call    tiga
        Ajmp    display7
empat6:
        cjne    a,#04h,lima6
        call    empat
        Ajmp    display7
lima6:
        cjne    a,#05h,enam6
        call    lima
        Ajmp    display7
enam6:
        cjne    a,#06h,tuju6
        call    enam
        Ajmp    display7
tuju6:
        cjne    a,#07h,delapan6
        call    tujuh
        Ajmp    display7
delapan6:
        cjne    a,#08h,sembilan6
        call    delapan
        Ajmp    display7
sembilan6:
        cjne    a,#09h,display7
        call    sembilan

display7:
        mov     p1,#00h
        mov     a,kode7
        cjne    a,#00h,satu7
        call    nol
        Ajmp    display8
satu7:
        cjne    a,#01h,dua7
        call    satu
        Ajmp    display8
dua7:
        cjne    a,#02h,tiga7
        call    dua
        Ajmp    display8
tiga7:
        cjne    a,#03h,empat7
        call    tiga
        Ajmp    display8
empat7:
        cjne    a,#04h,lima7
        call    empat
        Ajmp    display8
lima7:
        cjne    a,#05h,enam7
        call    lima
        Ajmp    display8
enam7:
        cjne    a,#06h,tuju7
```

```
        call    enam
        Ajmp    display8
tuju7:
        cjne    a,#07h,delapan7
        call    tujuh
        Ajmp    display8
delapan7:
        cjne    a,#08h,sembilan7
        call    delapan
        Ajmp    display8
sembilan7:
        cjne    a,#09h,display8
        call    sembilan
        Ajmp    display8

;*************************************4
DISPLAY8:
        ajmp    put


nol:
        mov     p1,#3eh
        acall   tampil
        mov     p1,#45h
        acall   tampil
        mov     p1,#49h
        acall   tampil
        mov     p1,#51h
        acall   tampil
        mov     p1,#3eh
        acall   tampil
        ret
satu:
        acall   tampil
        mov     p1,#21h
        acall   tampil
        mov     p1,#7fh
        acall   tampil
        mov     p1,#01h
        acall   tampil
        mov     p1,#00h
        acall   tampil
        ret
dua:
        mov     p1,#23h
        acall   tampil
        mov     p1,#45h
        acall   tampil
        mov     p1,#49h
        acall   tampil
        mov     p1,#49h
        acall   tampil
        mov     p1,#31h
        acall   tampil
        ret
tiga:
        mov     p1,#42h
        acall   tampil
```

```
        mov     p1,#41h
        acall   tampil
        mov     p1,#49h
        acall   tampil
        mov     p1,#59h
        acall   tampil
        mov     p1,#66h
        acall   tampil
        RET
empat:
        mov     p1,#0ch
        acall   tampil
        mov     p1,#14h
        acall   tampil
        mov     p1,#24h
        acall   tampil
        mov     p1,#7fh
        acall   tampil
        mov     p1,#04h
        acall   tampil
        RET
lima:
        mov     p1,#72h
        acall   tampil
        mov     p1,#51h
        acall   tampil
        mov     p1,#51h
        acall   tampil
        mov     p1,#51h
        acall   tampil
        mov     p1,#4eh
        acall   tampil
        ret
enam:
        mov     p1,#1eh
        acall   tampil
        mov     p1,#29h
        acall   tampil
        mov     p1,#49h
        acall   tampil
        mov     p1,#49h
        acall   tampil
        mov     p1,#46h
        acall   tampil
        ret
tujuh:
        mov     p1,#40h
        acall   tampil
        mov     p1,#47h
        acall   tampil
        mov     p1,#48h
        acall   tampil
        mov     p1,#50h
        acall   tampil
        mov     p1,#60h
        acall   tampil
        ret
delapan:
```

```
        mov     p1,#36h
        acall   tampil
        mov     p1,#49h
        acall   tampil
        mov     p1,#49h
        acall   tampil
        mov     p1,#49h
        acall   tampil
        mov     p1,#36h
        acall   tampil
        ret
sembilan:
        mov     p1,#31h
        acall   tampil
        mov     p1,#49h
        acall   tampil
        mov     p1,#49h
        acall   tampil
        mov     p1,#4ah
        acall   tampil
        mov     p1,#3ch
        acall   tampil
        ret

PULSA:
        mov     b,p0
        mov     a,#00001111b
        orl     a,b
        mov     b,#11110001b
        anl     a,b
        mov     p0,a
        MOV     R7,#02
LOOP5:  MOV     R6,#02
LOOP6:  MOV     R5,#02
        DJNZ    R5,$
        DJNZ    R6,LOOP6
        DJNZ    R7,LOOP5

        mov     b,p0
        mov     a,#00001111b
        orl     a,b
        mov     b,#11110101b
        anl     a,b
        mov     p0,a
        MOV     R7,#02
LOOP7:  MOV     R6,#02
LOOP8:  MOV     R5,#02
        DJNZ    R5,$
        DJNZ    R6,LOOP8
        DJNZ    R7,LOOP7

        mov     b,p0
        mov     a,#00001111b
        orl     a,b
        mov     b,#11110000b
        anl     a,b
        mov     p0,a
        MOV     R7,#02
```

```
LOOP9: MOV   R6,#02
LOOP10:      MOV   R5,#02
       DJNZ  R5,$
       DJNZ  R6,LOOP10
       DJNZ  R7,LOOP9
       mov   b,p0
       mov   a,#00001111b
       orl   a,b
       mov   b,#11110100b
       anl   a,b
       mov   p0,a
       MOV   R7,#02
LOOP11:      MOV   R6,#02
LOOP12:      MOV   R5,#02
       DJNZ  R5,$
       DJNZ  R6,LOOP12
       DJNZ  R7,LOOP11
       RET

tampil:
       mov   b,p0
       mov   a,#00001111b
       orl   a,b
       mov   b,#11110011b
       anl   a,b

       mov   p0,a
       MOV   R7,#02
LOOP1: MOV   R6,#02
LOOP2: MOV   R5,#02
       DJNZ  R5,$
       DJNZ  R6,LOOP2
       DJNZ  R7,LOOP1

       mov   b,p0
       mov   a,#00001111b
       orl   a,b
       mov   b,#11110101b
       anl   a,b

       mov   p0,a
       MOV   R7,#02
LOOP3: MOV   R6,#02
LOOP4: MOV   R5,#02
       DJNZ  R5,$
       DJNZ  R6,LOOP4
       DJNZ  R7,LOOP3
       ret


       END
```

. Overview

This MCU is built using the high-performance silicon gate CMOS process using a R8C/Tiny Series CPU core and is packaged in a 32-pin plastic molded LQFP. This MCU operates using sophisticated instructions featuring a high level of instruction efficiency. With 1M bytes of address space, it is capable of executing instructions at high speed.
The data flash ROM (2 KB X 2 blocks) is embedded.

1.1 Applications

Electric household appliance, office equipment, housing equipment (sensor, security), general industrial equipment, audio, etc.

1.2 Performance Outline

Table 1.1. lists the performance outline of this MCU.

Table 1.1 Performance outline

Item		Performance
CPU	Number of basic instructions	89 instructions
	Shortest instruction execution time	50 ns (f(XIN) = 20 MHz, VCC = 3.0 to 5.5 V) 100 ns (f(XIN) = 10 MHz, VCC = 2.7 to 5.5 V)
	Operating mode	Single-chip
	Address space	1M bytes
	Memory capacity	See Table 1.2.
Peripheral function	Interrupt	Internal: 11 factors, External: 5 factors, Software: 4 factors, Priority level: 7 levels
	Watchdog timer	15 bits x 1 (with prescaler) Reset start function selectable
	Timer	Timer X: 8 bits x 1 channel, Timer Y: 8 bits x 1 channel, Timer Z: 8 bits x 1 channel (Each timer equipped with 8-bit prescaler) Timer C: 16 bits x 1 channel Circuits of input capture and output compare.
	Serial interface	•1 channel Clock synchronous, UART •1 channel UART
	A/D converter	10-bit A/D converter: 1 circuit, 12 channels
	Clock generation circuit	2 circuits •Main clock generation circuit (Equipped with a built-in feedback resistor) •On-chip oscillator (high-speed, low-speed) On high-speed on-chip oscillator the frequency adjustment function is usable.
	Oscillation stop detection function	Stop detection of main clock oscillation
	Voltage detection circuit	Included
	Power on reset circuit	Included
	Port	Input/Output: 22 (including LED drive port), Input: 2 (LED drive I/O port: 8)
Electrical characteristics	Power supply voltage	VCC = 3.0 to 5.5V (f(XIN) = 20MHz) VCC = 2.7 to 5.5V (f(XIN) = 10MHz)
	Power consumption	Typ.9 mA (VCC = 5.0V, (f(XIN) = 20MHz,High-speed mode) Typ.5 mA (VCC = 3.0V, (f(XIN) = 10MHz,High-speed mode) Typ.35 µA (VCC = 3.0V, Wait mode, Peripheral clock stops) Typ.0.7 µA (VCC = 3.0V, Stop mode)
Flash memory	Program/erase voltage	VCC = 2.7 to 5.5 V
	Number of program/erase	10,000 times (Data area)
		1,000 times (Program area)
Operating ambient temperature		-20 to 85°C -40 to 85°C (D-version)
Package		32-pin plastic mold LQFP

1.3 Block Diagram

Figure 1.1 shows this MCU block diagram.

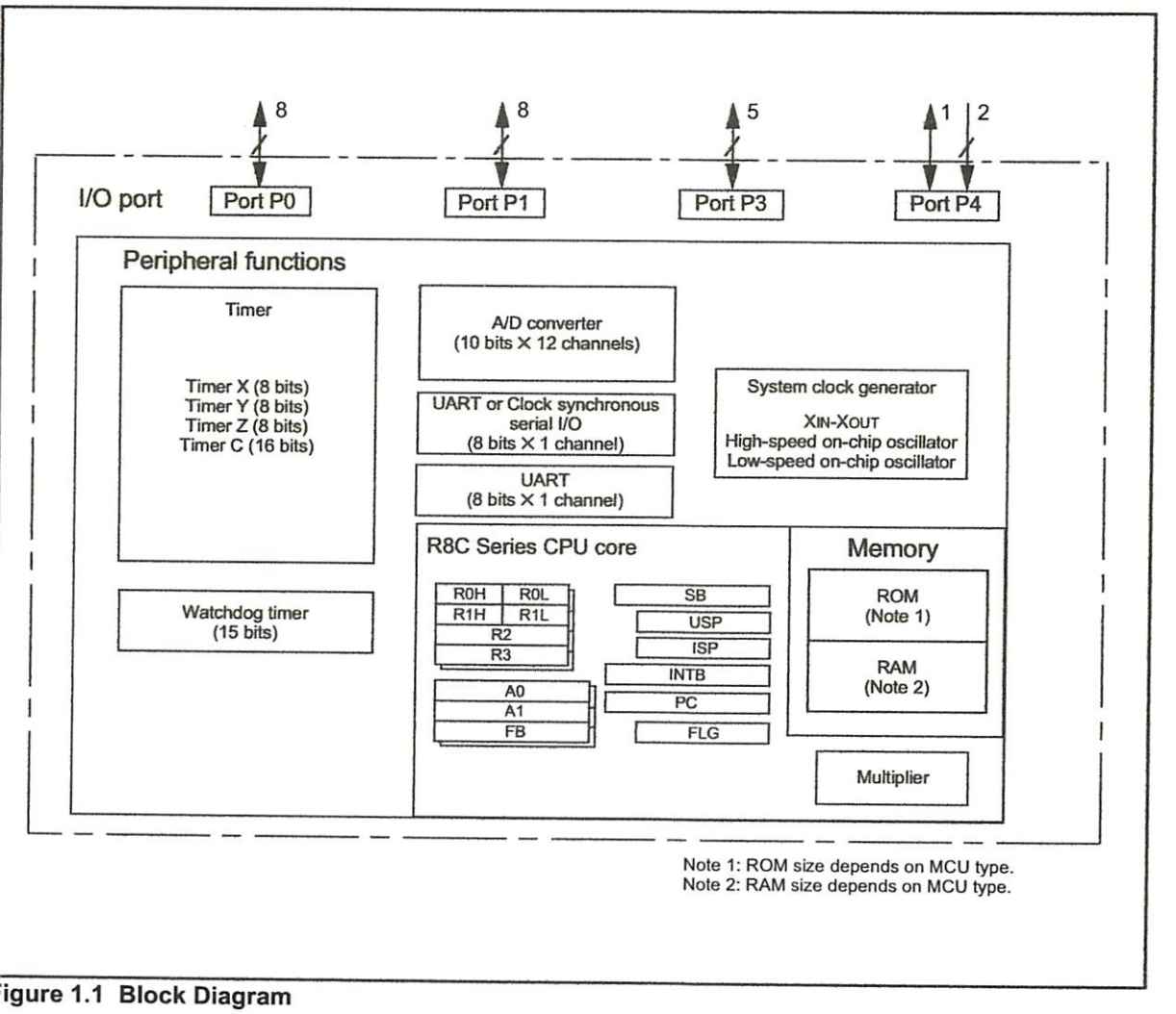


Figure 1.1 Block Diagram

1.4 Product Information

Table 1.2 lists the products.

Table 1.2 Product List

As of April 2005

Type No.	ROM capacity		RAM capacity	Package type	Remarks
	Program area	Data area			
R5F21132FP	8K bytes	2K bytes x 2	512 bytes	PLQP0032GB-A	Flash memory version
R5F21133FP	12K bytes	2K bytes x 2	768 bytes	PLQP0032GB-A	
R5F21134FP	16K bytes	2K bytes x 2	1K bytes	PLQP0032GB-A	
R5F21132DFP	8K bytes	2K bytes x 2	512 bytes	PLQP0032GB-A	D version
R5F21133DFP	12K bytes	2K bytes x 2	768 bytes	PLQP0032GB-A	
R5F21134DFP	16K bytes	2K bytes x 2	1K bytes	PLQP0032GB-A	

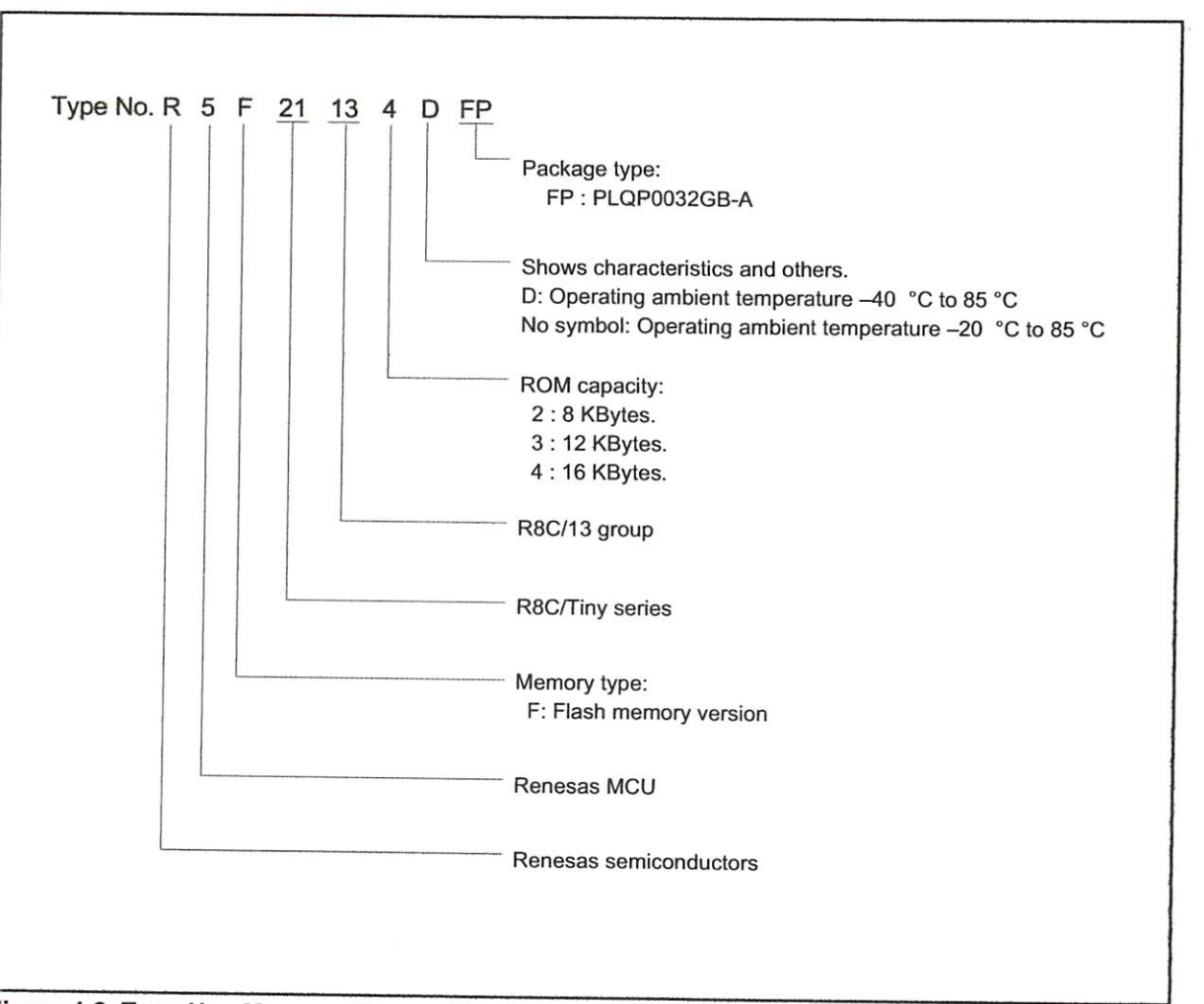
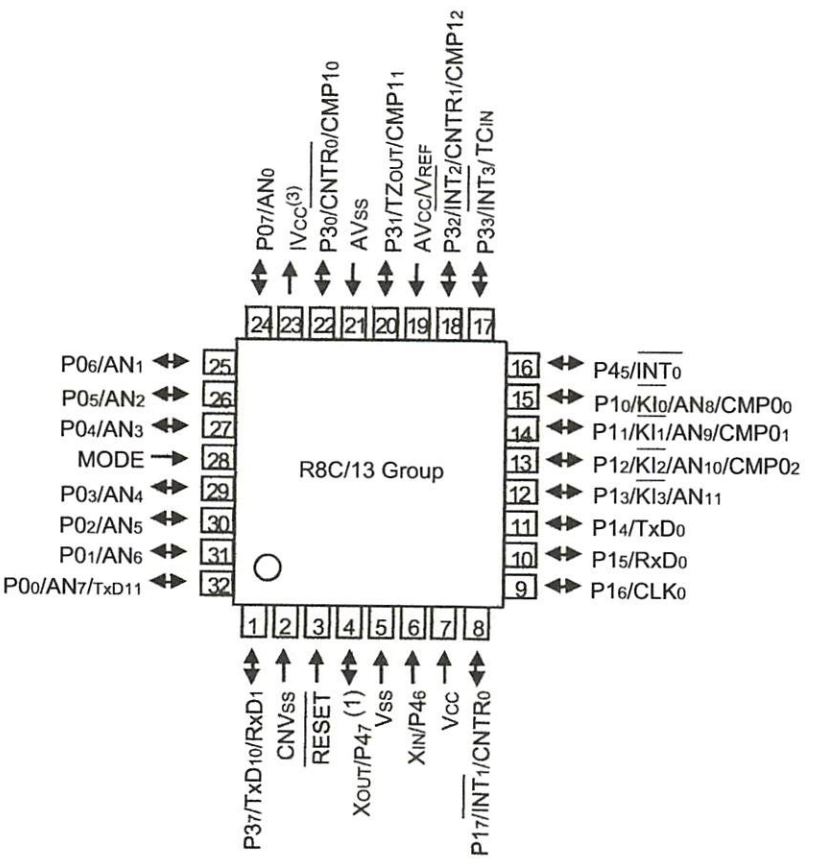


Figure 1.2 Type No., Memory Size, and Package

.5 Pin Assignments

Figure 1.3 shows the pin configuration (top view).

PIN Assignments (top view)



NOTES:
1. P47 functions only as an input port.
2. When using On-chip debugger, do not use P00/AN7/TxD11 and P37/TxD10/RxD1 pins.
3. Do not connect IVcc to Vcc.

Package: PLQP0032GB-A (32P6U-A)

igure 1.3 Pin Assignments (Top View)

1.6 Pin Description

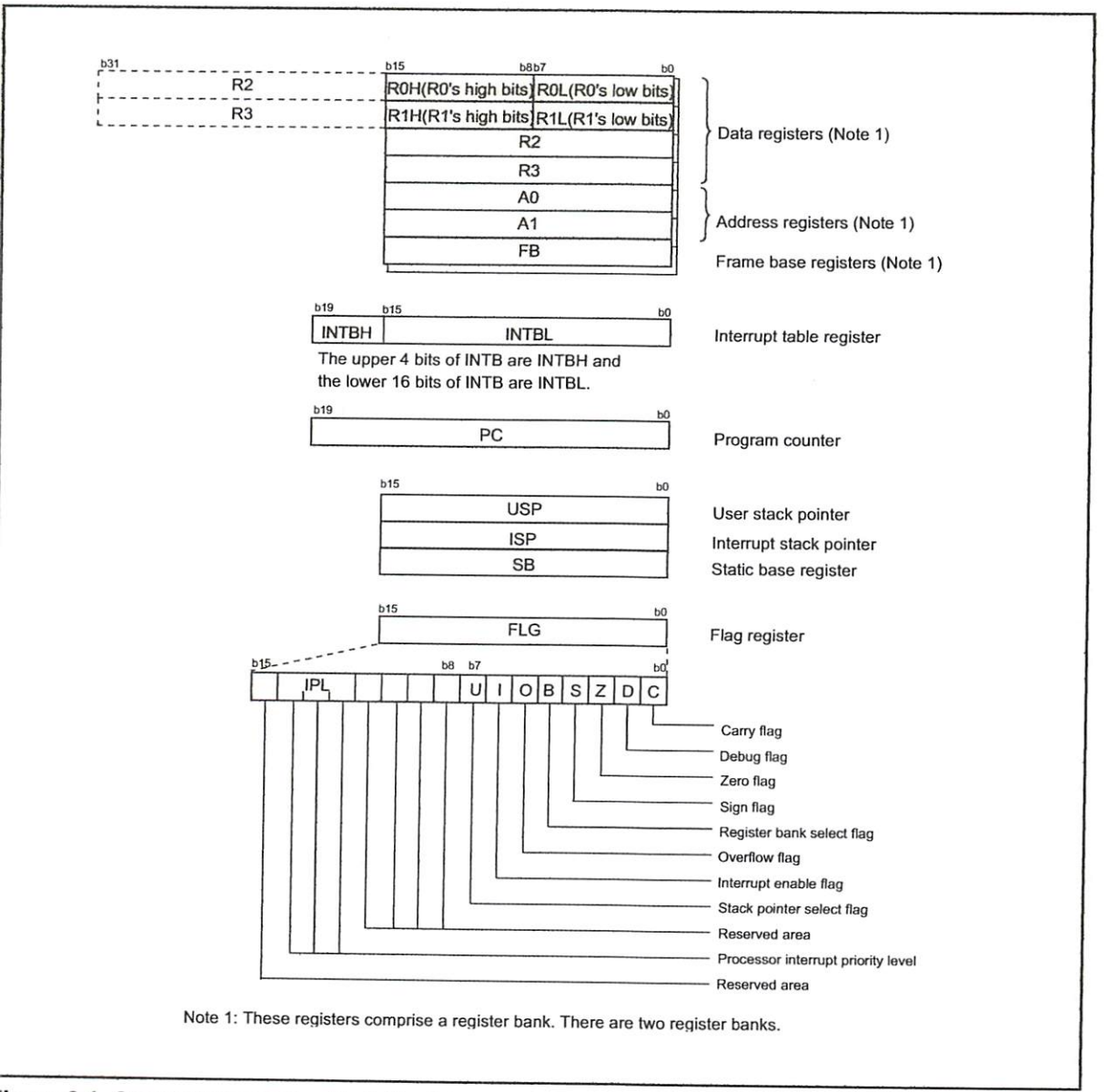
Table 1.3 shows the pin description

Table 1.3 Pin description

Signal name	Pin name	I/O type	Function
Power supply input	Vcc, Vss	I	Apply 2.7 V to 5.5 V to the Vcc pin. Apply 0 V to the Vss pin.
IVcc	IVcc	O	This pin is to stabilize internal power supply Connect this pin to Vss via a capacitor (0.1 µF) Do not connect to Vcc
Analog power supply input	AVcc, AVss	I	These are power supply input pins for A/D converter. Connect the AVcc pin to Vcc. Connect the AVss pin to Vss. Connect a capacitor between pins AVcc and AVss.
Reset input	$\overline{\text{RESET}}$	I	"L" on this input resets the MCU.
CNVss	CNVss	I	Connect this pin to Vss via a resistor(1)
MODE	MODE	I	Connect this pin to Vcc via a resistor
Main clock input	XIN	I	These pins are provided for the main clock generating circuit I/O. Connect a ceramic resonator or a crystal oscillator between the XIN and XOUT pins. To use an externally derived clock, input it to the XIN pin and leave the XOUT pin open.
Main clock output	XOUT	O	
$\overline{\text{INT}}$ interrupt input	$\overline{\text{INT0}}$ to $\overline{\text{INT3}}$	I	These are $\overline{\text{INT}}$ interrupt input pins.
Key input interrupt input	$\overline{\text{KI0}}$ to $\overline{\text{KI3}}$	I	These are key input interrupt pins.
Timer X	CNTR0	I/O	This is the timer X I/O pin.
	$\overline{\text{CNTR0}}$	O	This is the timer X output pin.
Timer Y	CNTR1	I/O	This is the timer Y I/O pin.
Timer Z	TZOUT	O	This is the timer Z output pin.
Timer C	TCIN	I	This is the timer C input pin.
	CMP00 to CMP03, CMP10 to CMP13	O	These are the timer C output pins.
Serial interface	CLK0	I/O	This is a transfer clock I/O pin.
	RxD0, RxD1	I	These are serial data input pins.
	TxD0, TxD10, TxD11	O	These are serial data output pins.
Reference voltage input	VREF	I	This is a reference voltage input pin for A/D converter. Connect the VREF pin to Vcc.
A/D converter	AN0 to AN11	I	These are analog input pins for A/D converter.
I/O port	P00 to P07, P10 to P17, P30 to P33, P37, P45	I/O	These are 8-bit CMOS I/O ports. Each port has an I/O select direction register, allowing each pin in that port to be directed for input or output individually. Any port set to input can select whether to use a pull-up resistor or not by program. P10 to P17 also function as LED drive ports.
nput port	P46, P47	I	These are input only pins.

2. Central Processing Unit (CPU)

Figure 2.1 shows the CPU registers. The CPU has 13 registers. Of these, R0, R1, R2, R3, A0, A1 and FB comprise a register bank. There are two register banks.



Note 1: These registers comprise a register bank. There are two register banks.

Figure 2.1 Central Processing Unit Register

2.1 Data Registers (R0, R1, R2 and R3)

The R0 register consists of 16 bits, and is used mainly for transfers and arithmetic/logic operations. R1 to R3 are the same as R0.

The R0 register can be separated between high (R0H) and low (R0L) for use as two 8-bit data registers. R1H and R1L are the same as R0H and R0L. Conversely, R2 and R0 can be combined for use as a 32-bit data register (R2R0). R3R1 is the same as R2R0.

2.2 Address Registers (A0 and A1)

The register A0 consists of 16 bits, and is used for address register indirect addressing and address register relative addressing. They also are used for transfers and logic/logic operations. A1 is the same as A0.

In some instructions, registers A1 and A0 can be combined for use as a 32-bit address register (A1A0).

2.3 Frame Base Register (FB)

FB is configured with 16 bits, and is used for FB relative addressing.

2.4 Interrupt Table Register (INTB)

INTB is configured with 20 bits, indicating the start address of an interrupt vector table.

2.5 Program Counter (PC)

PC is configured with 20 bits, indicating the address of an instruction to be executed.

2.6 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP)

Stack pointer (SP) comes in two types: USP and ISP, each configured with 16 bits.

Your desired type of stack pointer (USP or ISP) can be selected by the U flag of FLG.

2.7 Static Base Register (SB)

SB is configured with 16 bits, and is used for SB relative addressing.

2.8 Flag Register (FLG)

FLG consists of 11 bits, indicating the CPU status.

2.8.1 Carry Flag (C Flag)

This flag retains a carry, borrow, or shift-out bit that has occurred in the arithmetic/logic unit.

2.8.2 Debug Flag (D Flag)

The D flag is used exclusively for debugging purpose. During normal use, it must be set to "0".

2.8.3 Zero Flag (Z Flag)

This flag is set to "1" when an arithmetic operation resulted in 0; otherwise, it is "0".

2.8.4 Sign Flag (S Flag)

This flag is set to "1" when an arithmetic operation resulted in a negative value; otherwise, it is "0".

2.8.5 Register Bank Select Flag (B Flag)

Register bank 0 is selected when this flag is "0" ; register bank 1 is selected when this flag is "1".

2.8.6 Overflow Flag (O Flag)

This flag is set to "1" when the operation resulted in an overflow; otherwise, it is "0".

2.8.7 Interrupt Enable Flag (I Flag)

This flag enables a maskable interrupt.

Maskable interrupts are disabled when the I flag is "0", and are enabled when the I flag is "1". The I flag is cleared to "0" when the interrupt request is accepted.

2.8.8 Stack Pointer Select Flag (U Flag)

ISP is selected when the U flag is "0"; USP is selected when the U flag is "1".

The U flag is cleared to "0" when a hardware interrupt request is accepted or an INT instruction for software interrupt Nos. 0 to 31 is executed.

2.8.9 Processor Interrupt Priority Level (IPL)

IPL is configured with three bits, for specification of up to eight processor interrupt priority levels from level 0 to level 7.

If a requested interrupt has priority greater than IPL, the interrupt is enabled.

2.8.10 Reserved Area

When write to this bit, write "0". When read, its content is indeterminate.

3. Memory

Figure 3.1 is a memory map of this MCU. The address space extends the 1M bytes from address 0000016 to FFFFF16.

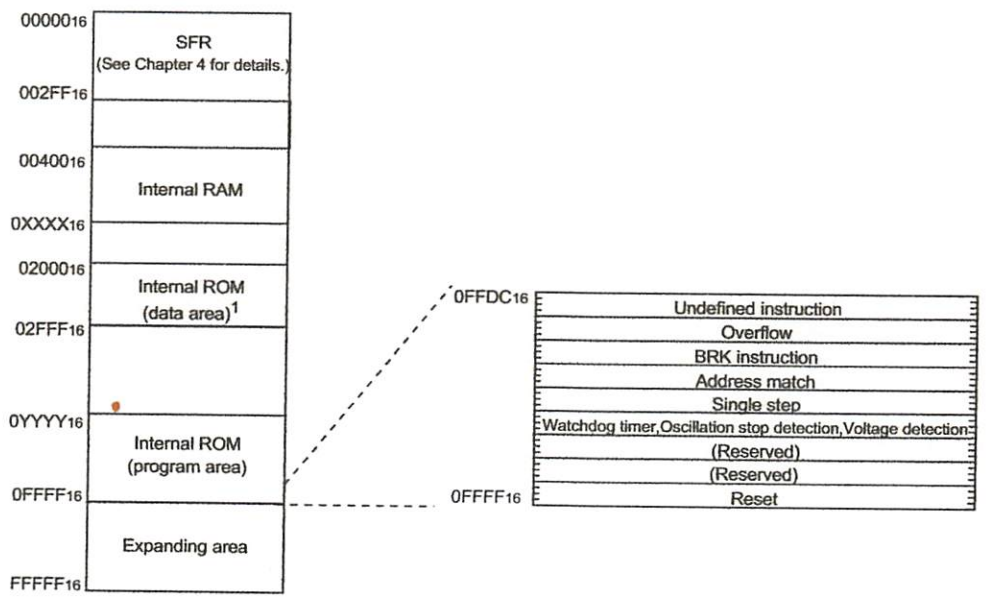
The internal ROM (program area) is allocated in a lower address direction beginning with address 0FFFF16. For example, a 16-Kbyte internal ROM is allocated to the addresses from 0C00016 to 0FFFF16.

The fixed interrupt vector table is allocated to the addresses from 0FFDC16 to 0FFFF16. Therefore, store the start address of each interrupt routine here.

The internal ROM (data area) is allocated to the addresses from 0200016 to 02FFF16.

The internal RAM is allocated in an upper address direction beginning with address 0040016. For example, a 1-Kbyte internal RAM is allocated to the addresses from 0040016 to 007FF16. In addition to storing data, the internal RAM also stores the stack used when calling subroutines and when interrupts are generated.

Special function registers (SFR) are allocated to the addresses from 0000016 to 002FF16. Peripheral function control registers are located here. Of the SFR, any space which has no functions allocated is reserved for future use and cannot be used by users.



NOTES:
1. The data flash ROM block A (2K bytes) and block B (2K bytes) are shown.
2. Blank spaces are reserved. No access is allowed.

Type name	Internal ROM		Internal RAM	
	Size	Address 0YYYY16	Size	Address 0XXXX16
R5F21134FP, R5F21134DFP	16K bytes	0C00016	1K bytes	007FF16
R5F21133FP, R5F21133DFP	12K bytes	0D00016	768 bytes	006FF16
R5F21132FP, R5F21132DFP	8K bytes	0E00016	512 bytes	005FF16

Figure 3.1 Memory Map

4. Special Function Register (SFR)

SFR(Special Function Register) is the control register of peripheral functions. Tables 4.1 to 4.4 list the SFR information

Table 4.1 SFR Information(1)(1)

Address	Register	Symbol	After reset
000016			
000116			
000216			
000316			
000416	Processor mode register 0 [1]	PM0	0016
000516	Processor mode register 1	PM1	0016
000616	System clock control register 0	CM0	011010002
000716	System clock control register 1	CM1	001000002
000816	High-speed on-chip oscillator control register 0	HR0	0016
000916	Address match interrupt enable register	AIER	XXXXXX002
000A16	Protect register	PRCR	00XXX0002
000B16	High-speed on-chip oscillator control register 1	HR1	4016
000C16	Oscillation stop detection register	OCD	000001002
000D16	Watchdog timer reset register	WDTR	XX16
000E16	Watchdog timer start register	WDTS	XX16
000F16	Watchdog timer control register	WDC	000111112
001016 001116 001216	Address match interrupt register 0	RMAD0	0016 0016 X016
001316			
001416 001516 001616	Address match interrupt register 1	RMAD1	0016 0016 X016
001716			
001816			
001916	Voltage detection register 1 [2]	VCR1	000010002
001A16	Voltage detection register 2 [2]	VCR2	0016 [3] 100000002 [4]
001B16			
001C16			
001D16			
001E16	INT0 input filter select register	INT0F	XXXXX0002
001F16	Voltage detection interrupt register [2]	D4INT	0016 [3] 010000012 [4]
002016			
002116			
002216			
002316			
002416			
002516			
002616			
002716			
002816			
002916			
002A16			
002B16			
002C16			
002D16			
002E16			
002F16			
003016			
003116			
003216			
003316			
003416			
003516			
003616			
003716			
003816			
003916			
003A16			
003B16			
003C16			
003D16			
003E16			
003F16			

: Undefined

OTES:

1. Blank columns are all reserved space. No access is allowed.
2. Software reset or the watchdog timer reset does not affect this register.
3. Owing to Reset input.
4. In the case of RESET pin = H retaining.

Table 4.2 SFR Information(2)(1)

Address	Register	Symbol	After reset
0040₁₆			
0041₁₆			
0042₁₆			
0043₁₆			
0044₁₆			
0045₁₆			
0046₁₆			
0047₁₆			
0048₁₆			
0049₁₆			
004A₁₆			
004B₁₆			
004C₁₆			
004D₁₆	Key input interrupt control register	KUPIC	XXXXX0002
004E₁₆	AD conversion interrupt control register	ADIC	XXXXX0002
004F₁₆			
0050₁₆	Compare 1 interrupt control register	CMP1IC	XXXXX0002
0051₁₆	UART0 transmit interrupt control register	S0TIC	XXXXX0002
0052₁₆	UART0 receive interrupt control register	S0RIC	XXXXX0002
0053₁₆	UART1 transmit interrupt control register	S1TIC	XXXXX0002
0054₁₆	UART1 receive interrupt control register	S1RIC	XXXXX0002
0055₁₆	INT2 interrupt control register	INT2IC	XXXXX0002
0056₁₆	Timer X interrupt control register	TXIC	XXXXX0002
0057₁₆	Timer Y interrupt control register	TYIC	XXXXX0002
0058₁₆	Timer Z interrupt control register	TZIC	XXXXX0002
0059₁₆	INT1 interrupt control register	INT1IC	XXXXX0002
005A₁₆	INT3 interrupt control register	INT3IC	XXXXX0002
005B₁₆	Timer C interrupt control register	TCIC	XXXXX0002
005C₁₆	Compare 0 interrupt control register	CMP0IC	XXXXX0002
005D₁₆	INT0 interrupt control register	INT0IC	XX00X0002
005E₁₆			
005F₁₆			
0060₁₆			
0061₁₆			
0062₁₆			
0063₁₆			
0064₁₆			
0065₁₆			
0066₁₆			
0067₁₆			
0068₁₆			
0069₁₆			
006A₁₆			
006B₁₆			
006C₁₆			
006D₁₆			
006E₁₆			
006F₁₆			
0070₁₆			
0071₁₆			
0072₁₆			
0073₁₆			
0074₁₆			
0075₁₆			
0076₁₆			
0077₁₆			
0078₁₆			
0079₁₆			
007A₁₆			
007B₁₆			
007C₁₆			
007D₁₆			
007E₁₆			
007F₁₆			

: Undefined

OTES:

1. Blank columns are all reserved space. No access is allowed.

Table 4.3 SFR Information(3)[1]

Address	Register	Symbol	After reset
008016	Timer Y, Z mode register	TYZMR	0016
008116	Prescaler Y	PREY	FF16
008216	Timer Y secondary	TYSC	FF16
008316	Timer Y primary	TYPR	FF16
008416	Timer Y, Z waveform output control register	PUM	0016
008516	Prescaler Z	PREZ	FF16
008616	Timer Z secondary	TZSC	FF16
008716	Timer Z primary	TZPR	FF16
008816			
008916			
008A16	Timer Y, Z output control register	TYZOC	0016
008B16	Timer X mode register	TXMR	0016
008C16	Prescaler X	PREX	FF16
008D16	Timer X register	TX	FF16
008E16	Count source set register	TCSS	0016
008F16			
009016 009116	Timer C register	TC	0016 0016
009216			
009316			
009416			
009516			
009616	External input enable register	INTEN	0016
009716			
009816	Key input enable register	KIEN	0016
009916			
009A16	Timer C control register 0	TCC0	0016
009B16	Timer C control register 1	TCC1	0016
009C16 009D16	Capture, compare 0 register	TM0	0016 0016[2]
009E16 009F16	Compare 1 register	TM1	FF16 FF16
00A016	UART0 transmit/receive mode register	U0MR	0016
00A116	UART0 bit rate register	U0BRG	XX16
00A216 00A316	UART0 transmit buffer register	U0TB	XX16 XX16
00A416	UART0 transmit/receive control register 0	U0C0	000010002
00A516	UART0 transmit/receive control register 1	U0C1	000000102
00A616 00A716	UART0 receive buffer register	U0RB	XX16 XX16
00A816	UART1 transmit/receive mode register	U1MR	0016
00A916	UART1 bit rate register	U1BRG	XX16
00AA16 00AB16	UART1 transmit buffer register	U1TB	XX16 XX16
00AC16	UART1 transmit/receive control register 0	U1C0	000010002
00AD16	UART1 transmit/receive control register 1	U1C1	000000102
00AE16 00AF16	UART1 receive buffer register	U1RB	XX16 XX16
00B016	UART transmit/receive control register 2	UCON	0016
00B116			
00B216			
00B316			
00B416			
00B516			
00B616			
00B716			
00B816			
00B916			
00BA16			
00BB16			
00BC16			
00BD16			
00BE16			
00BF16			

X : Undefined

NOTES:

1. Blank columns are all reserved space. No access is allowed.
2. When the output compare mode is selected (the TCC13 bit in the TCC1 register = 1), the value is set to FFFF16.

Table 4.4 SFR Information(4)(1)

Address	Register	Symbol	After reset
00C016	AD register	AD	XX16
00C116			XX16
00C216			
00C316			
00C416			
00C516			
00C616			
00C716			
00C816			
00C916			
00CA16			
00CB16			
00CC16			
00CD16			
00CE16			
00CF16			
00D016			
00D116			
00D216			
00D316			
00D416	AD control register 2	ADCON2	0016
00D516			
00D616	AD control register 0	ADCON0	00000XXX2
00D716	AD control register 1	ADCON1	0016
00D816			
00D916			
00DA16			
00DB16			
00DC16			
00DD16			
00DE16			
00DF16			
00E016	Port P0 register	P0	XX16
00E116	Port P1 register	P1	XX16
00E216	Port P0 direction register	PD0	0016
00E316	Port P1 direction register	PD1	0016
00E416			
00E516	Port P3 register	P3	XX16
00E616			
00E716	Port P3 direction register	PD3	0016
00E816	Port P4 register	P4	XX16
00E916			
00EA16	Port P4 direction register	PD4	0016
00EB16			
00EC16			
00ED16			
00EE16			
00EF16			
00F016			
00F116			
00F216			
00F316			
00F416			
00F516			
00F616			
00F716			
00F816			
00F916			
03FA16			
00FB16			
00FC16	Pull-up control register 0	PUR0	00XX00002
00FD16	Pull-up control register 1	PUR1	XXXXXX0X2
00FE16	Port P1 drive capacity control register	DRR	0016
00FF16	Timer C output control register	TCOUT	0016

01B316	Flash memory control register 4	FMR4	010000002
01B416			
01B516	Flash memory control register 1	FMR1	1000000X2
01B616			
01B716	Flash memory control register 0	FMR0	000000012

0FFFF16	Option function select register (2)	OFS	Note 2

: Undefined

OTES:

. The blank areas, 010016 to 01B216 and 01B816 to 02FF16 are reserved and cannot be used by users.

. The watchdog timer control bit is assigned. Refer to "Figure11.2 OFS, WDC, WDTR and WDTS registers" of Hardware Manual for details

5. Electrical Characteristics

Table 5.1 Absolute Maximum Ratings

Symbol	Parameter	Condition	Rated value	Unit
V_{CC}	Supply voltage	V_{CC}=AVcc	-0.3 to 6.5	V
AVcc	Analog supply voltage	V_{CC}=AVcc	-0.3 to 6.5	V
V_I	Input voltage		-0.3 to V_{CC}+0.3	V
V_O	Output voltage		-0.3 to V_{CC}+0.3	V
P_d	Power dissipation	Topr=25 °C	300	mW
T_{opr}	Operating ambient temperature		-20 to 85 / -40 to 85 (D version)	°C
T_{stg}	Storage temperature		-65 to 150	°C

Table 5.2 Recommended Operating Conditions

Symbol	Parameter	Conditions	Standard			Unit
			Min.	Typ.	Max.	
V_{CC}	Supply voltage		2.7		5.5	V
AVcc	Analog supply voltage			V_{CC}[3]		V
Vss	Supply voltage			0	—	V
AVss	Analog supply voltage			0		V
V_{IH}	"H" input voltage		0.8V_{CC}		V_{CC}	V
V_{IL}	"L" input voltage		0		0.2V_{CC}	V
$I_{OH\ (sum)}$	"H" peak all output currents: Sum of all pins' IOH (peak)				-60.0	mA
$I_{OH\ (peak)}$	"H" peak output current				-10.0	mA
$I_{OH\ (avg)}$	"H" average output current				-5.0	mA
$I_{OL\ (sum)}$	"L" peak all output currents: Sum of all pins' IOL (peak)				60	mA
$I_{OL\ (peak)}$	"L" peak output current: Except P1$_0$ to P1$_7$				10	mA
	P1$_0$ to P1$_7$	Drive ability HIGH		—	30	mA
		Drive ability LOW			10	mA
$I_{OL\ (avg)}$	"L" average output current: Except P1$_0$ to P1$_7$				5	mA
	P1$_0$ to P1$_7$	Drive ability HIGH			15	mA
		Drive ability LOW			5	mA
f(X_{IN})	Main clock input oscillation frequency	3.0V ≤ V_{CC} ≤ 5.5V	0		20	MHz
		2.7V ≤ V_{CC} < 3.0V	0		10	MHz

Note
1: Referenced to V_{CC} = AVcc = 2.7 to 5.5V at Topr = -20 to 85 °C / -40 to 85 °C unless otherwise specified.
2: The mean output current is the mean value within 100ms.
3: Set V_{CC}=AVcc

Table 5.3 A/D Conversion Characteristics

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
–	Resolution		V_{ref} =V_{CC}		—	10	Bit
–	Absolute accuracy	10 bit mode	øAD=10 MHz, Vref=Vcc=5.0V			±3	LSB
		8 bit mode	øAD=10 MHz, Vref=Vcc=5.0V			±2	LSB
		10 bit mode	øAD=10 MHz, Vref=Vcc=3.3V[3]			±5	LSB
		8 bit mode	øAD=10 MHz, Vref=Vcc=3.3V[3]			±2	LSB
R_{LADDER}	Ladder resistance		V_{REF}=V_{CC}	10		40	kΩ
t_{CONV}	Conversion time	10 bit mode	øAD=10 MHz, Vref=Vcc=5.0V	3.3	—		µs
		8 bit mode	øAD=10 MHz, Vref=Vcc=5.0V	2.8			µs
V_{REF}	Reference voltage				V_{CC}[4]		V
V_{IA}	Analog input voltage			0		V_{ref}	V
–	A/D operation clock frequency[2]	Without sample & hold		0.25	—	10	MHz
		With sample & hold		1.0		10	MHz

Note
1: Referenced to V_{CC}=AV_{CC}=2.7 to 5.5V at Topr = -20 to 85 °C / -40 to 85 °C unless otherwise specified.
2: When f_{AD} is 10 MHz more, divide the f_{AD} and make A/D operation clock frequency ($Ø_{AD}$) lower than 10 MHz.
3: When the AVcc is less than 4.2V, divide the f_{AD} and make A/D operation clock frequency ($Ø_{AD}$) lower than f_{AD}/2.
4: Set Vcc=Vref

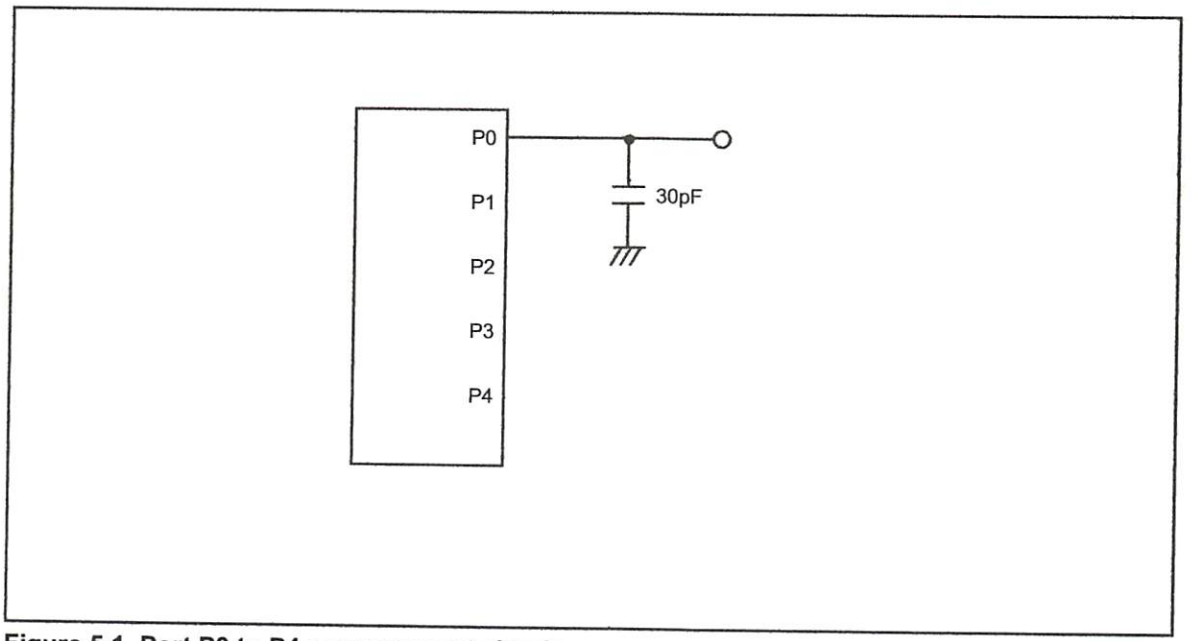


Figure 5.1 Port P0 to P4 measurement circuit

Table 5.4 Flash Memory (Program area) Electrical Characteristics

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max	
–	Program/Erase cycle [2]		1000[3]	—	—	cycle
–	Byte program time	Vcc = 5.0 V at Topr = 25 °C	—	50	—	µs
–	Block erase time	Vcc = 5.0 V at Topr = 25 °C	—	0.4	—	s
td(SR-ES)	Time delay from Suspend Request until Erase Suspend		—	—	8	ms
–	Erase Suspend Request Interval		10	—	—	ms
–	Program, Erase Voltage		2.7	—	5.5	V
–	Read Voltage		2.7	—	5.5	V
–	Program, Erase Temperature		0	—	60	°C
–	Data-retention duration	Topr = 55 °C	20	—	—	year

Table 5.5 Flash Memory (Data area Block A, Block B) Electrical Characteristics [4]

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max	
–	Program/Erase endurance[2]		10000 [3]	—	—	times
–	Byte program time(program/erase endurance ≤1000 times)	Vcc = 5.0 V at Topr = 25 °C	—	50	400	µs
–	Byte program time(program/erase endurance >1000 times)	Vcc = 5.0 V at Topr = 25 °C	—	65	—	µs
–	Block erase time(program/erase endurance ≤1000 times)	Vcc = 5.0 V at Topr = 25 °C	—	0.2	9	s
–	Block erase time(program/erase endurance >1000 times)	Vcc = 5.0 V at Topr = 25 °C	—	0.3	—	s
td(SR-ES)	Time delay from Suspend Request until Erase Suspend		—	—	8	ms
–	Erase Suspend Request Interval		10	—	—	ms
–	Program, Erase Voltage		2.7	—	5.5	V
–	Read Voltage		2.7	—	5.5	V
–	Program/Erase Temperature		-20(-40)[8]	—	85	°C
–	Data-retention duration	Topr = 55 °C	20	—	—	year

Note
1: Referenced to Vcc=AVcc=2.7 to 5.5V at Topr = 0°C to 60°C unless otherwise specified.
2: Definition of Program/Erase
The cycle of Program/Erase shows a cycle for each block.
If the program/erase number is "n" (n = 1000, 10000), "n" times erase can be performed for each block.
For example, if performing one-byte write to the distinct addresses on Block A of 2K-byte block 2048 times and then erasing that block, the number of Program/Erase cycles is one time.
However, performing multiple writes to the same address before an erase operation is prohibited (overwriting prohibited).
3: Maximum numbers of Program/Erase cycles for which all electrical characteristics is guaranteed.
4: Table 16.5 applies for Block A or B when the Program/Erase cycles are more than 1000. The byte program time up to 1000 cycles are the same as that of the program area (see Table 5.4).
5: To reduce the number of Program/Erase cycles, a block erase should ideally be performed after writing in series as many distinct addresses (only one time each) as possible. If programming a set of 16 bytes, write up to 128 sets and then erase them one time. This will result in ideally reducing the number of Program/Erase cycles. Additionally, averaging the number of Program/Erase cycles for Block A and B will be more effective. It is important to track the total number of block erases and restrict the number.
6: If error occurs during block erase, attempt to execute the clear status register command, then the block erase command at least three times until the erase error disappears.
7: Customers desiring Program/Erase failure rate information should contact their Renesas technical support representative.
8: -40 °C for D version.

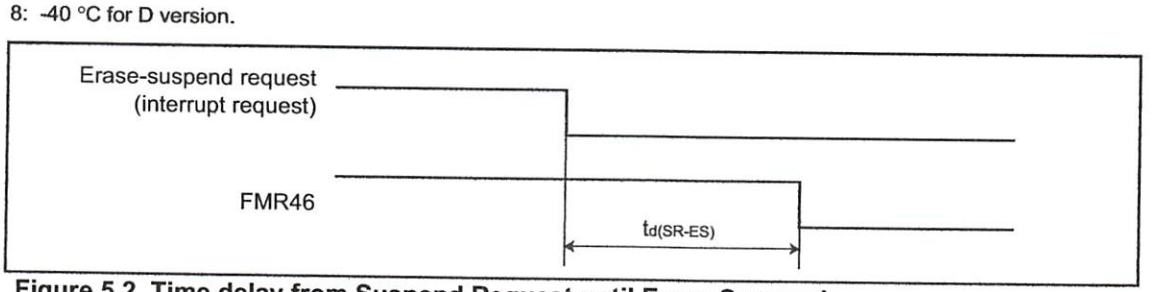


Figure 5.2 Time delay from Suspend Request until Erase Suspend

Table 5.6 Voltage Detection Circuit Electrical Characteristics

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
Vdet	Voltage detection level		3.3	3.8	4.3	V
	Voltage detection interrupt request generating time[2]			40		µs
	Voltage detection circuit self consumption current	VC27=1, VCC=5.0V		600		nA
td(E-A)	Waiting time until voltage detection circuit operation starts[3]				20	µs
Vccmin	Microcomputer operation voltage minimum value		2.7			V

NOTES:
1. The measuring condition is Vcc=AVcc=2.7V to 5.5V and Topr=-40°C to 85°C.
2. This shows the time until the voltage detection interrupt request is generated since the voltage passes Vdet.
3. This shows the required time until the voltage detection circuit operates when setting to "1" again after setting the VC27 bit in the VCR2 register to "0".

Table 5.7 Reset Circuit Electrical Characteristics (When Using Hardware Reset 2[1, 3])

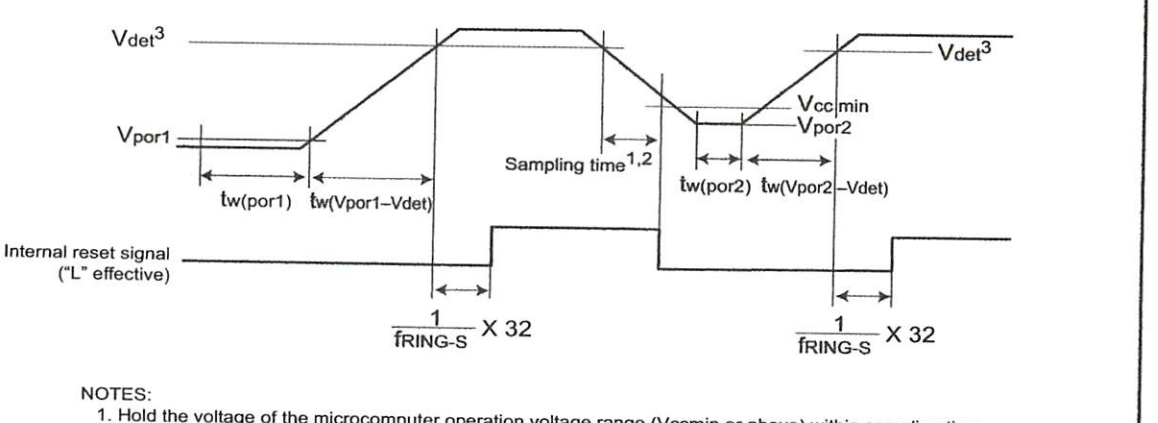
Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
Vpor2	Power-on reset valid voltage	–20°C ≤ Topr < 85°C	—	—	Vdet	V
tw(Vpor2-Vdet)	Supply voltage rising time when power-on reset is canceled[2]	–20°C ≤ Topr < 85°C, tw(por2) ≥ 0s[4]	—	—	100	ms

NOTES:
1. The voltage detection circuit which is embedded in a microcomputer is a factor to generate the hardware reset 2. Refer to 5.1.2 Hardware Reset 2.
2. This condition is not applicable when using Vcc ≥ 1.0V.
3. When turning power on after the external power has been held below the valid voltage for greater than 10 seconds, refer to Table 16.8 Reset Circuit Electrical Characteristics (When Not Using Hardware Reset 2).
4. tw(por2) is time to hold the external power below effective voltage (Vpor2).

Table 5.8 Reset Circuit Electrical Characteristics (When Not Using Hardware Reset 2)

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
Vpor1	Power-on reset valid voltage	–20°C ≤ Topr < 85°C	—	—	0.1	V
tW(Vpor1-Vdet)	Supply voltage rising time when power-on reset is canceled	0°C ≤ Topr ≤ 85°C, tw(por1) ≥ 10s[2]	—	—	100	ms
tW(Vpor1-Vdet)	Supply voltage rising time when power-on reset is canceled	–20°C ≤ Topr < 0°C, tw(por1) ≥ 30s[2]	—	—	100	ms
tW(Vpor1-Vdet)	Supply voltage rising time when power-on reset is canceled	–20°C ≤ Topr < 0°C, tw(por1) ≥ 10s[2]	—	—	1	ms
tW(Vpor1-Vdet)	Supply voltage rising time when power-on reset is canceled	0°C ≤ Topr ≤ 85°C, tw(por1) ≥ 1s[2]	—	—	0.5	ms

NOTES:
1. When not using hardware reset 2, use with Vcc ≥ 2.7V.
2. tw(por1) is time to hold the external power below effective voltage (Vpor1).



NOTES:
1. Hold the voltage of the microcomputer operation voltage range (Vccmin or above) within sampling time.
2. A sampling clock is selectable. Refer to "5.4 Voltage Detection Circuit" for details.
3. Vdet shows the voltage detection level of the voltage detection circuit. Refer to "5.4 Voltage Detection Circuit" for details.

Figure 5.3 Reset Circuit Electrical Characteristics

Table 5.9 High-speed On-Chip Oscillator Circuit Electrical Characteristics

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
—	High-speed on-chip oscillator frequency 1 / {td(HRoffset)+td(HR)} when the reset is released	VCC=5.0V, Topr=25 °C Set "40$_{16}$" in the HR1 register	—	8	—	MHz
td(HRoffset)	Settable high-speed on-chip oscillator minimum period	VCC=5.0V, Topr=25 °C Set "00$_{16}$" in the HR1 register	—	61	—	ns
td(HR)	High-speed on-chip oscillator period adjusted unit	Differences when setting "01$_{16}$" and "00$_{16}$" in the HR register	—	1	—	ns
—	High-speed on-chip oscillator temperature dependence(1)	Frequency fluctuation in temperature range of -10 °C to 50 °C	—	±5	—	%
—	High-speed on-chip oscillator temperature dependence(2)	Frequency fluctuation in temperature range of -40 °C to 85 °C	—	±10	—	%

NOTES:
1. The measuring condition is Vcc=AVcc=5.0 V and Topr=25 °C.

Table 5.10 Power Circuit Timing Characteristics

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
td(P-R)	Time for internal power supply stabilization during powering-on[2]		1		2000	µs
td(R-S)	STOP release time[3]				150	µs

Note
1: The measuring condition is Vcc=AVcc=2.7 to 5.5 V and Topr=25 °C.
2: This shows the wait time until the internal power supply generating circuit is stabilized during power-on.
3: This shows the time until BCLK starts from the interrupt acknowledgement to cancel stop mode.

Table 5.11 Electrical Characteristics (1) [Vcc=5V]

Symbol	Parameter		Measuring condition		Standard			Unit
					Min.	Typ.	Max.	
V_{OH}	"H" output voltage	Except X_{OUT}	I_{OH}=-5mA		Vcc-2.0	—	Vcc	V
			I_{OH}=-200µA		Vcc-0.3	—	Vcc	V
		X_{OUT}	Drive capacity HIGH	I_{OH}=-1 mA	Vcc-2.0	—	Vcc	V
			Drive capacity LOW	I_{OH}=-500µA	Vcc-2.0	—	Vcc	V
V_{OL}	"L" output voltage	P10 to P17 Except X_{OUT}	I_{OL}= 5 mA		—	—	2.0	V
			I_{OL}= 200 µA		—	—	0.45	V
		P10 to P17	Drive capacity HIGH	I_{OL}= 15 mA	—	—	2.0	V
			Drive capacity LOW	I_{OL}= 5 mA	—	—	2.0	V
			Drive capacity LOW	I_{OL}= 200 µA	—	—	0.45	V
		X_{OUT}	Drive capacity HIGH	I_{OL}= 1 mA	—	—	2.0	V
			Drive capacity LOW	I_{OL}=500 µA	—	—	2.0	V
V_{T+}-V_{T-}	Hysteresis	$\overline{INT0}$, $\overline{INT1}$, $\overline{INT2}$, $\overline{INT3}$, KI0, KI1, KI2, KI3, CNTR0, CNTR1, TCIN, RxD0, RxD1, P45			0.2	—	1.0	V
		$\overline{RESET}$			0.2	—	2.2	V
I_{IH}	"H" input current		V_I=5V		—	—	5.0	µA
I_{IL}	"L" input current		V_I=0V		—	—	-5.0	µA
R_{PULLUP}	Pull-up resistance		V_I=0V		30	50	167	kΩ
R_{fXIN}	Feedback resistance	X_{IN}			—	1.0	—	MΩ
f_{RING-S}	Low-speed on-chip oscillator frequency				40	125	250	kHz
V_{RAM}	RAM retention voltage		At stop mode		2.0	—	—	V

Note
1 : Referenced to Vcc=AVcc=4.2 to 5.5V at Topr = -20 to 85 °C / -40 to 85 °C, f(X_{IN})=20MHz unless otherwise specified.

Table 5.12 Electrical Characteristics (2) [Vcc=5V]

Symbol	Parameter	Measuring condition		Standard Min.	Standard Typ.	Standard Max.	Unit
Icc	Power supply current (Vcc=3.3 to 5.5V) In single-chip mode, the output pins are open and other pins are Vss	High-speed mode	X_{IN}=20 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz No division		9	15	mA
			X_{IN}=16 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz No division		8	14	mA
			X_{IN}=10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz No division		5		mA
		Medium-speed mode	X_{IN}=20 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz Division by 8		4		mA
			X_{IN}=16 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz Division by 8		3		mA
			X_{IN}=10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz Division by 8		2		mA
		High-speed on-chip oscillator mode	Main clock off High-speed on-chip oscillator on=8 MHz Low-speed on-chip oscillator on=125 kHz No division		4	8	mA
			Main clock off High-speed on-chip oscillator on=8 MHz Low-speed on-chip oscillator on=125 kHz Division by 8		1.5		mA
		Low-speed on-chip oscillator mode	Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz Division by 8		470	900	µA
		Wait mode	Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz When a WAIT instruction is executed[2] Peripheral clock operation VC27="0"		40	80	µA
		Wait mode	Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz When a WAIT instruction is executed[2] Peripheral clock off VC27="0"		38	76	µA
		Stop mode	Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10="1" Peripheral clock off VC27="0"		0.8	3.0	µA

NOTES
1: The power supply current measuring is executed using the measuring program on frash memory.
2: Timer Y is operated with timer mode.

Timing requirements (Unless otherwise noted: Vcc = 5V, Vss = 0V at Ta = 25 °C) [Vcc=5V]

Table 5.13 XIN input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(XIN)	XIN input cycle time	50		ns
twH(XIN)	XIN input HIGH pulse width	25		ns
twL(XIN)	XIN input LOW pulse width	25		ns

Table 5.14 CNTR0 input, CNTR1 input, $\overline{INT2}$ input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(CNTR0)	CNTR0 input cycle time	100		ns
twH(CNTR0)	CNTR0 input HIGH pulse width	40		ns
twL(CNTR0)	CNTR0 input LOW pulse width	40		ns

Table 5.15 TCIN input, $\overline{INT3}$ input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TCIN)	TCIN input cycle time	400 [1]		ns
twH(TCIN)	TCIN input HIGH pulse width	200 [2]		ns
twL(TCIN)	TCIN input LOW pulse width	200 [2]		ns

NOTES

1 :When using the Timer C input capture mode, adjust the cycle time above (1/ Timer C count source frequency x 3).

2 : When using the Timer C input capture mode, adjust the pulse width above (1/ Timer C count source frequency x 1.5).

Table 5.16 Serial Interface

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(CK)	CLKi input cycle time	200		ns
tw(CKH)	CLKi input HIGH pulse width	100		ns
tw(CKL)	CLKi input LOW pulse width	100		ns
td(C-Q)	TxDi output delay time		80	ns
th(C-Q)	TxDi hold time	0		ns
tsu(D-C)	RxDi input setup time	35		ns
th(C-D)	RxDi input hold time	90		ns

Table 5.17 External interrupt $\overline{INT0}$ input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tw(INH)	$\overline{INT0}$ input HIGH pulse width	250 [1]		ns
tw(INL)	$\overline{INT0}$ input LOW pulse width	250 [2]		ns

NOTES

1 : When selecting the digital filter by the $\overline{INT0}$ input filter select bit, use the $\overline{INT0}$ input HIGH pulse width to the greater value,either (1/ digital filter clock frequency x 3) or the minimum value of standard.

2 : When selecting the digital filter by the $\overline{INT0}$ input filter select bit, use the $\overline{INT0}$ input LOW pusle width to the greater value,either (1/ digital filter clock frequency x 3) or the minimum value of standard.

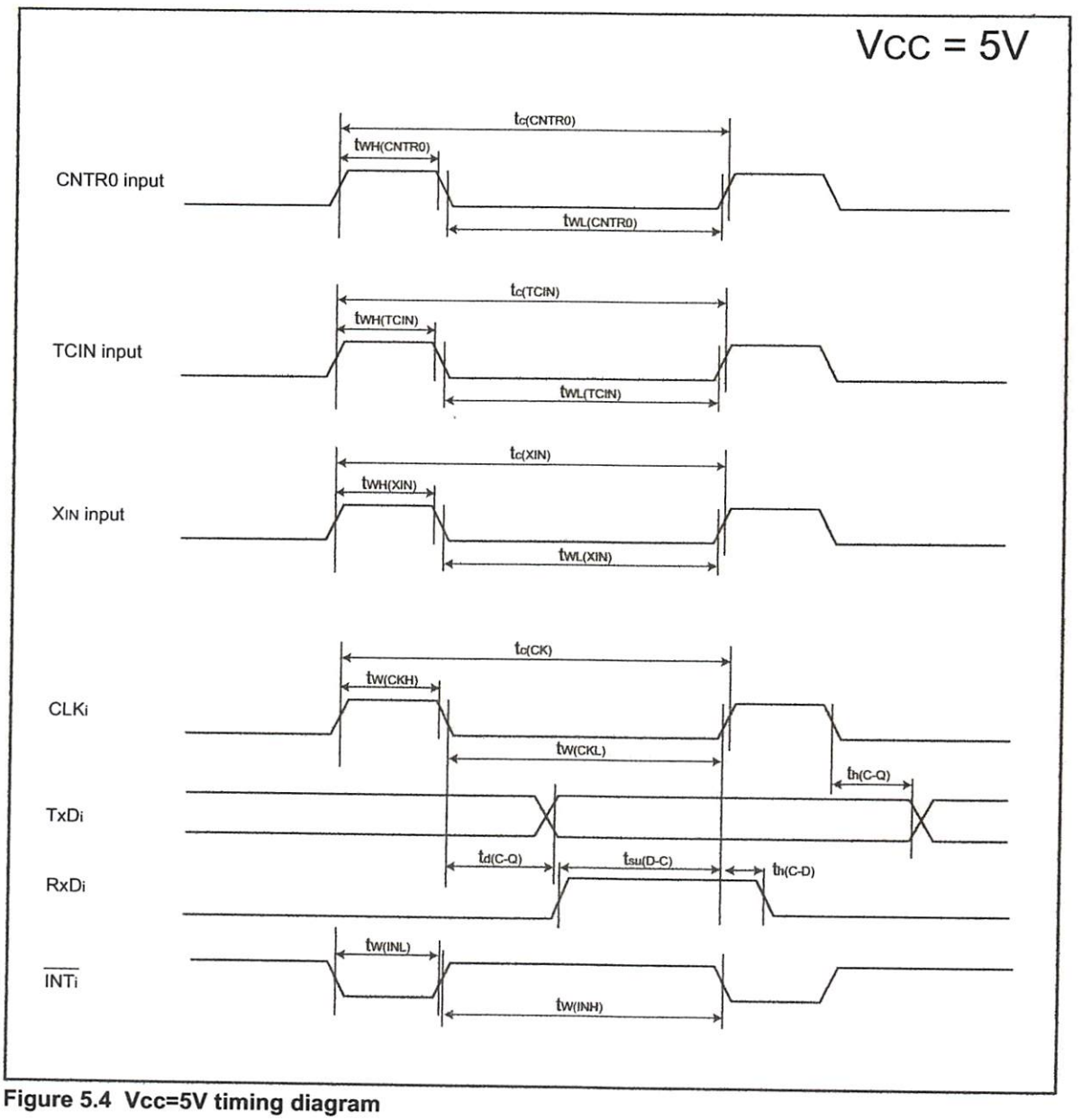


Figure 5.4 Vcc=5V timing diagram

Table 5.18 Electrical Characteristics (3) [Vcc=3V]

Symbol	Parameter		Measuring condition		Standard			Unit
					Min.	Typ.	Max.	
V_{OH}	"H" output voltage	Except X_{OUT}	I_{OH}=-1mA		Vcc-0.5	—	Vcc	V
		X_{OUT}	Drive capacity HIGH	I_{OH}=-0.1 mA	Vcc-0.5	—	Vcc	V
			Drive capacity LOW	I_{OH}=-50 µA	Vcc-0.5	—	Vcc	V
V_{OL}	"L" output voltage	P10 to P17 Except X_{OUT}	I_{OL}= 1 mA		—	—	0.5	V
		P10 to P17	Drive capacity HIGH	I_{OL}= 2 mA	—	—	0.5	V
			Drive capacity LOW	I_{OL}= 1 mA	—	—	0.5	V
		X_{OUT}	Drive capacity HIGH	I_{OL}= 0.1 mA	—	—	0.5	V
			Drive capacity LOW	I_{OL}=50 µA	—	—	0.5	V
V_{T+}-V_{T-}	Hysteresis	$\overline{INT_0}$, $\overline{INT_1}$, $\overline{INT_2}$, $\overline{INT_3}$, $\overline{KI_0}$, $\overline{KI_1}$, $\overline{KI_2}$, $\overline{KI_3}$, CNTR0, CNTR1, TCIN, RxD0, RxD1, P45			0.2	—	0.8	V
		$\overline{RESET}$			0.2	—	1.8	V
I_{IH}	"H" input current		V_I=3V		—	—	4.0	µA
I_{IL}	"L" input current		V_I=0V		—	—	-4.0	µA
R_{PULLUP}	Pull-up resistance		V_I=0V		66	160	500	kΩ
R_{fXIN}	Feedback resistance	X_{IN}			—	3.0	—	MΩ
f_{RING-S}	Low-speed on-chip oscillator frequency				40	125	250	kHz
V_{RAM}	RAM retention voltage		At stop mode		2.0	—	—	V

Note
1 : Referenced to Vcc=AVcc=2.7 to 3.3V at Topr = -20 to 85 °C / -40 to 85 °C, f(X_{IN})=10MHz unless otherwise specified.

Table 5.19 Electrical Characteristics (4) [Vcc=3V]

Symbol	Parameter	Measuring condition		Standard Min.	Standard Typ.	Standard Max.	Unit
Icc	Power supply current (Vcc=2.7 to 3.3V) In single-chip mode, the output pins are open and other pins are Vss	High-speed mode	XIN=20 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz No division		8	13	mA
			XIN=16 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz No division		7	12	mA
			XIN=10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz No division		5		mA
		Medium-speed mode	XIN=20 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz Division by 8		3		mA
			XIN=16 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz Division by 8		2.5		mA
			XIN=10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz Division by 8		1.6		mA
		High-speed on-chip oscillator mode	Main clock off High-speed on-chip oscillator on=8 MHz Low-speed on-chip oscillator on=125 kHz No division		3.5	7.5	mA
			Main clock off High-speed on-chip oscillator on=8 MHz Low-speed on-chip oscillator on=125 kHz Division by 8		1.5	—	mA
		Low-speed on-chip oscillator mode	Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz Division by 8		420	800	µA
		Wait mode	Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz When a WAIT instruction is executed[2] Peripheral clock operation VC27="0"		37	74	µA
		Wait mode	Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on=125 kHz When a WAIT instruction is executed[2] Peripheral clock off VC27="0"		35	70	µA
		Stop mode	Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10="1" Peripheral clock off VC27="0"		0.7	3.0	µA

NOTES
1: The power supply current measuring is executed using the measuring program on frash memory.
2: Timer Y is operated with timer mode.

Timing requirements (Unless otherwise noted: Vcc = 3V, Vss = 0V at Ta = 25 °C) [Vcc=3V]

Table 5.20 XIN input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(XIN)	XIN input cycle time	100		ns
tWH(XIN)	XIN input HIGH pulse width	40		ns
tWL(XIN)	XIN input LOW pulse width	40		ns

Table 5.21 CNTR0 input, CNTR1 input, $\overline{INT2}$ input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(CNTR0)	CNTR0 input cycle time	300		ns
tWH(CNTR0)	CNTR0 input HIGH pulse width	120		ns
tWL(CNTR0)	CNTR0 input LOW pulse width	120		ns

Table 5.22 TCIN input, $\overline{INT3}$ input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TCIN)	TCIN input cycle time	1200 [1]		ns
tWH(TCIN)	TCIN input HIGH pulse width	600 [2]		ns
tWL(TCIN)	TCIN input LOW pulse width	600 [2]		ns

NOTES

1 :When using the Timer C input capture mode, adjust the cycle time above (1/ Timer C count source frequency x 3).

2 : When using the Timer C input capture mode, adjust the pulse width above (1/ Timer C count source frequency x 1.5).

Table 5.23 Serial Interface

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(CK)	CLKi input cycle time	300		ns
tW(CKH)	CLKi input HIGH pulse width	150		ns
tW(CKL)	CLKi input LOW pulse width	150		ns
td(C-Q)	TxDi output delay time		160	ns
th(C-Q)	TxDi hold time	0		ns
tsu(D-C)	RxDi input setup time	55		ns
th(C-D)	RxDi input hold time	90		ns

Table 5.24 External interrupt $\overline{INT0}$ input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tW(INH)	$\overline{INT0}$ input HIGH pulse width	380 [1]		ns
tW(INL)	$\overline{INT0}$ input LOW pulse width	380 [2]		ns

NOTES

1 : When selecting the digital filter by the $\overline{INT0}$ input filter select bit, use the $\overline{INT0}$ input HIGH pulse width to the greater value,either (1/ digital filter clock frequency x 3) or the minimum value of standard.

2 : When selecting the digital filter by the $\overline{INT0}$ input filter select bit, use the $\overline{INT0}$ input LOW pusle width to the greater value,either (1/ digital filter clock frequency x 3) or the minimum value of standard.

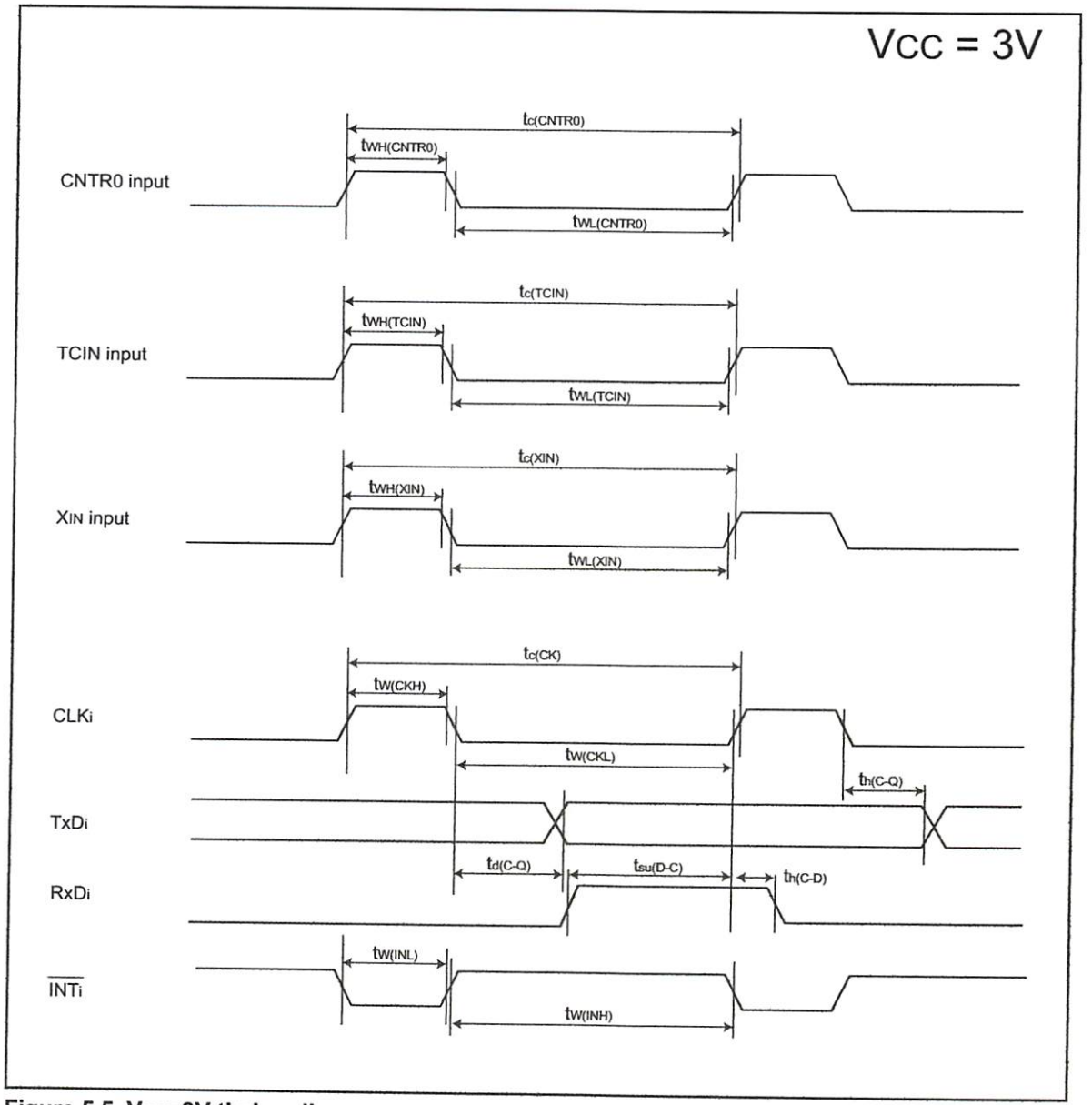
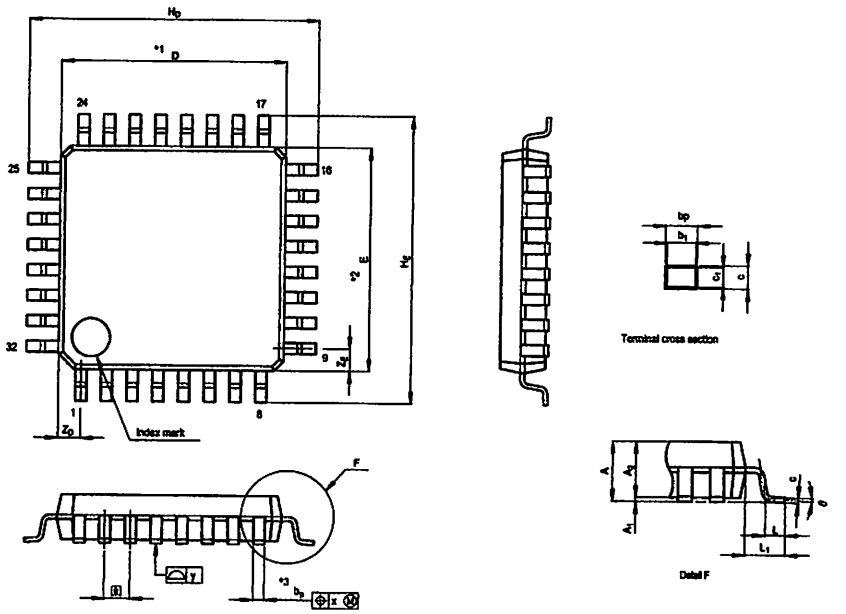


Figure 5.5 Vcc=3V timing diagram

Package Dimensions

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP32-7x7-0.80	PLQP0032GB-A	32P6U-A	0.2g



NOTE)
1. DIMENSIONS "I" AND "2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A_2	—	1.4	—
H_D	8.8	9.0	9.2
H_E	8.8	9.0	9.2
A	—	—	1.7
A_1	0	0.1	0.2
b_p	0.32	0.37	0.42
b_1	—	0.35	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.8	—
x	—	—	0.20
y	—	—	0.10
Z_D	—	0.7	—
Z_E	—	0.7	—
L	0.3	0.5	0.7
L_1	—	1.0	—

ev.	Date	Description	
		Page	Summary
10	Oct 28, 2003		First edition issued
20	Dec05, 2003	5	Figure 1.3 revised
		10	Chapter 4, NOTES revised
		16	Table 5.4 revised Table 5.5 revised
		17	Table 5.6 revised Figure 5.3 added
		18	Table 5.8 revised Table 5.10 revised
		21	Figure 5.3 revised to Figure 5.4
		22	Table 5.17 revised
		25	Figure 5.4 revised to Figure 5.5
)0	Sep 30, 2004	All pages	Words standardized (on-chip oscillator, serial interface, A/D)
		2	Table 1.1 revised
		5	Figure 1.3, NOTES 3 added
		6	Table 1.3 revised
		9	Figure 3.1, NOTES added
		10-13	One body sentence in chapter 4 added ; Titles of Table 4.1 to 4.4 added
		12	Table 4.3 revised ; Table 4.4 revised
		14	Table 5.2 revised
		15	Table 5.3 revised
		16	Table 5.4 and Table 5.5 revised
		17	Table 5.6, 5.7 and 5.8 revised ; Figure 5.3 revised
		18	Table 5.9 and 5.11 revised
		19	Table 5.12 revised
		20	Table 5.13 revised
		22	Table 5.18 revised
		23	Table 5.19 revised
		24	Table 5.20 and Table 5.24 revised
)	Apr.27.2005	4	Table 1.2, Figure 1.2 package name revised
		5	Figure 1.3 package name revised
		10	Table 4.1 revised
		12	Table 4.3 revised
		15	Table 5.3 partly revised
		16	Table 5.4, Table 5.5 partly added

ev.	Date	Description	
		Page	Summary
10	Apr.27.2005	17	Table 5.7, 5.8 revised
		18	Table 5.10, Table 5.11 partly revised
		22	Table 5.18 partly revised
		26	Package Dimensions revised

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

RENESAS

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology (Shanghai) Co., Ltd.
Unit2607 Ruijing Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

© 2005. Renesas Technology Corp., All rights reserved. Printed in Japan.

Colophon 2.0

ires

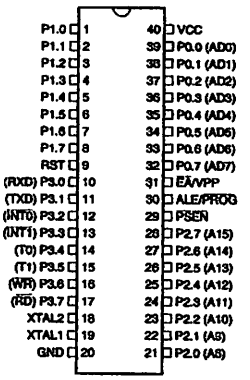
- atible with MCS-51™ Products
- les of In-System Reprogrammable Flash Memory
- ıdurance: 1,000 Write/Erase Cycles
- ɜtatic Operation: 0 Hz to 24 MHz
- level Program Memory Lock
- ɜ-bit Internal RAM
- grammable I/O Lines
- ɜ-bit Timer/Counters
- errupt Sources
- ımmable Serial Channel
- ɔwer Idle and Power-down Modes

ɜiption

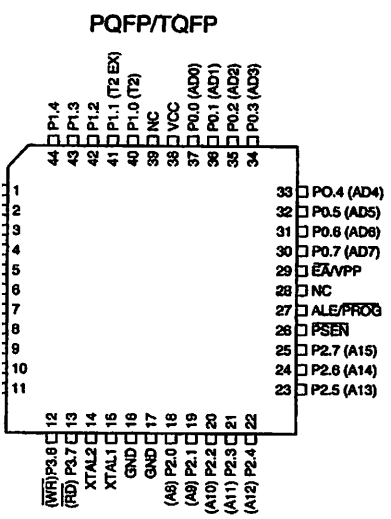
ɜ9C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K Flash programmable and erasable read only memory (PEROM). The device factured using Atmel's high-density nonvolatile memory technology and is ɔle with the industry-standard MCS-51 instruction set and pinout. The on-chip lows the program memory to be reprogrammed in-system or by a conven- nvolatile memory programmer. By combining a versatile 8-bit CPU with Flash ıolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides flexible and cost-effective solution to many embedded control applications.

ɔnfigurations

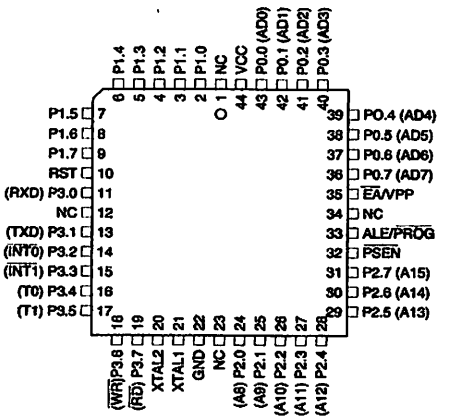
PDIP



PQFP/TQFP



PLCC



ATMEL®

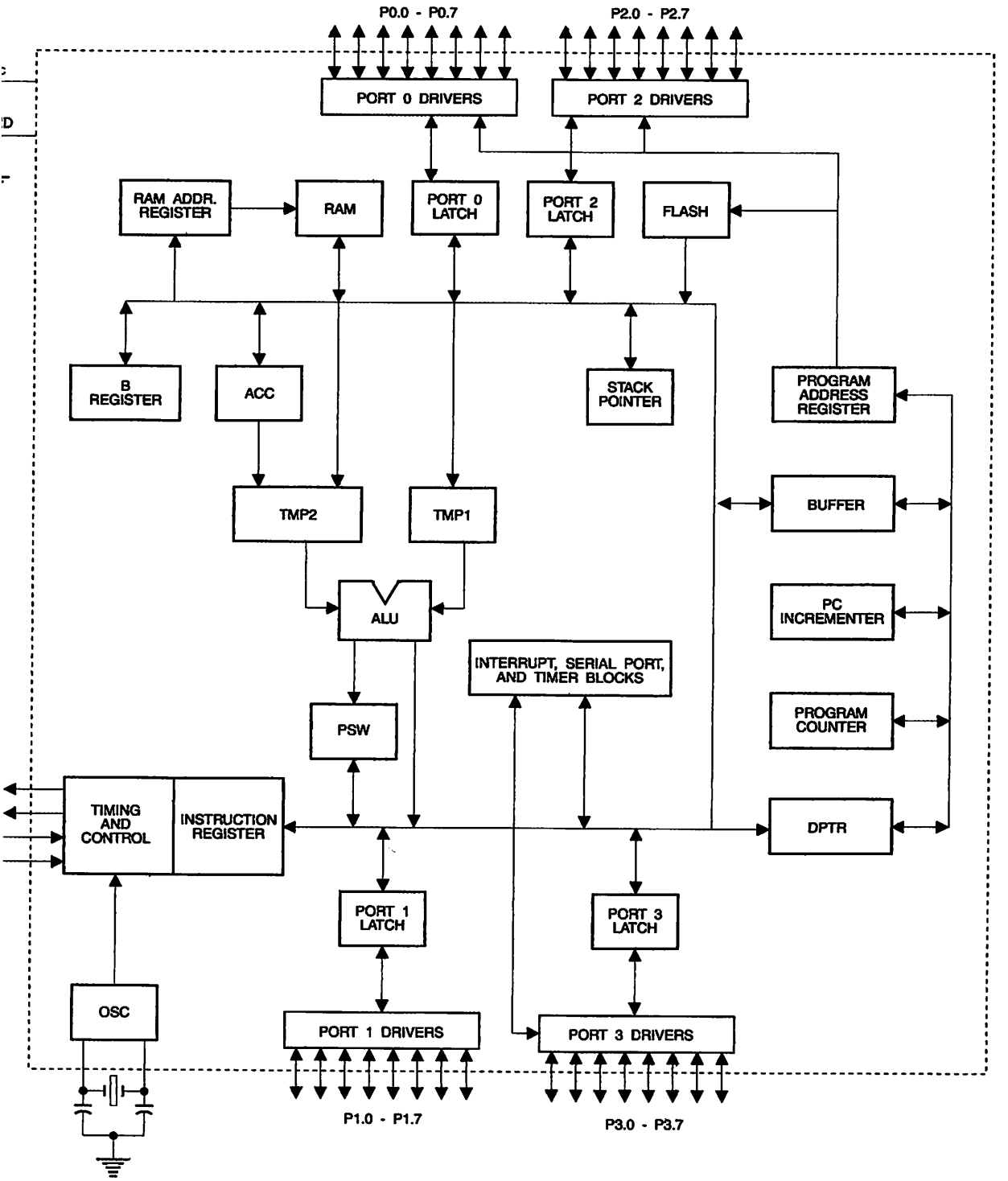
8-bit Microcontroller with 4K Bytes Flash

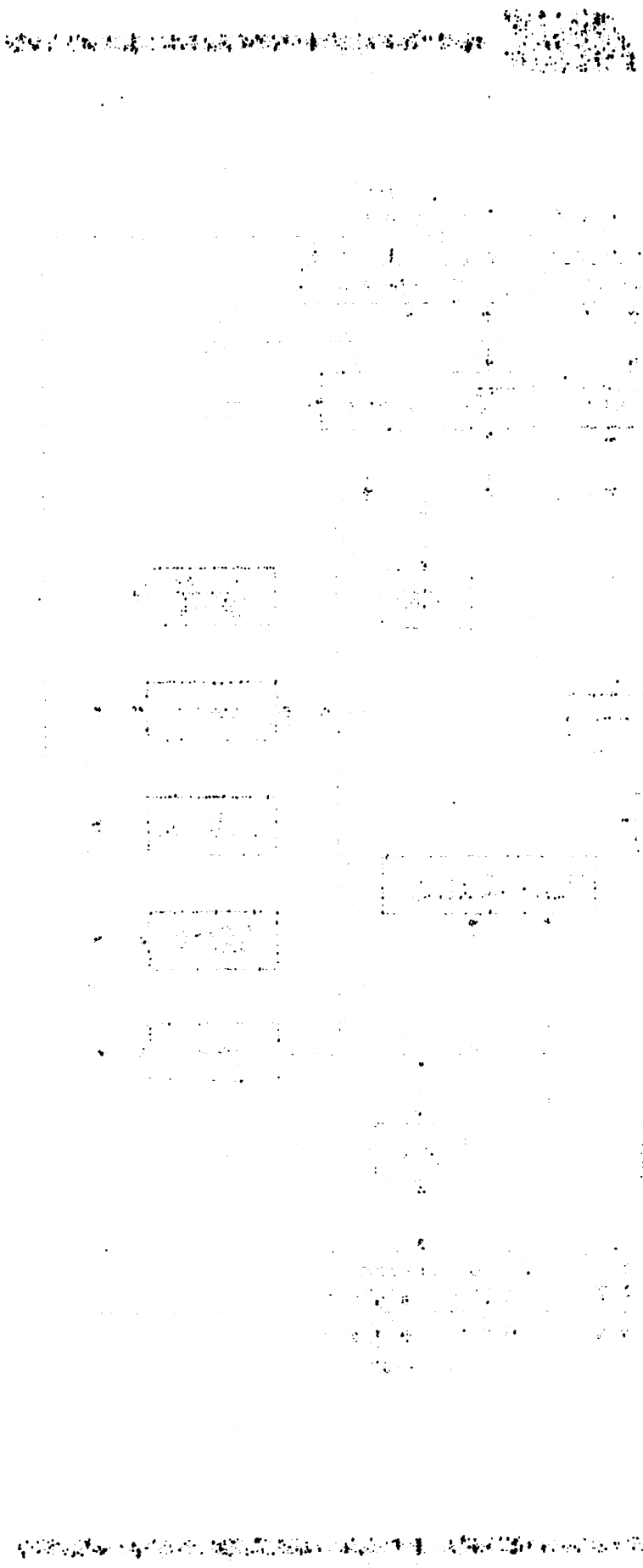
AT89C51

Rev. 0265G–02/00

ATMEL

Diagram


P0.0 - P0.7
P2.0 - P2.7
PORT 0 DRIVERS
PORT 2 DRIVERS
RAM ADDR. REGISTER
RAM
PORT 0 LATCH
PORT 2 LATCH
FLASH
B REGISTER
ACC
STACK POINTER
PROGRAM ADDRESS REGISTER
TMP2
TMP1
BUFFER
ALU
PC INCREMENTER
INTERRUPT, SERIAL PORT, AND TIMER BLOCKS
PSW
PROGRAM COUNTER
TIMING AND CONTROL
INSTRUCTION REGISTER
DPTR
PORT 1 LATCH
PORT 3 LATCH
OSC
PORT 1 DRIVERS
PORT 3 DRIVERS
P1.0 - P1.7
P3.0 - P3.7



89C51 provides the following standard features: 4K f Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit unters, a five vector two-level interrupt architecture, uplex serial port, on-chip oscillator and clock cir- n addition, the AT89C51 is designed with static logic ration down to zero frequency and supports two e selectable power saving modes. The Idle Mode ne CPU while allowing the RAM, timer/counters, ort and interrupt system to continue functioning. The down Mode saves the RAM contents but freezes llator disabling all other chip functions until the next e reset.

escription

voltage.

s an 8-bit open-drain bi-directional I/O port. As an ort, each pin can sink eight TTL inputs. When 1s ten to port 0 pins, the pins can be used as high- nce inputs.

nay also be configured to be the multiplexed low- ldress/data bus during accesses to external pro- nd data memory. In this mode P0 has internal

lso receives the code bytes during Flash program- nd outputs the code bytes during program on. External pullups are required during program on.

an 8-bit bi-directional I/O port with internal pullups. 1 output buffers can sink/source four TTL inputs. s are written to Port 1 pins they are pulled high by nal pullups and can be used as inputs. As inputs, ns that are externally being pulled low will source I_{IL}) because of the internal pullups.

lso receives the low-order address bytes during ogramming and verification.

an 8-bit bi-directional I/O port with internal pullups. 2 output buffers can sink/source four TTL inputs. s are written to Port 2 pins they are pulled high by nal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE

; skipped during each access to external Data
:.

d, ALE operation can be disabled by setting bit 0 of
ation 8EH. With the bit set, ALE is active only dur-
OVX or MOVC instruction. Otherwise, the pin is
pulled high. Setting the ALE-disable bit has no
the microcontroller is in external execution mode.

ı Store Enable is the read strobe to external pro-
ɜmory.

ıe AT89C51 is executing code from external pro-
emory, $\overline{PSEN}$ is activated twice each machine
(cept that two $\overline{PSEN}$ activations are skipped during
:ess to external data memory.

Access Enable. $\overline{EA}$ must be strapped to GND in
enable the device to fetch code from external pro-
ɜmory locations starting at 0000H up to FFFFH.
ıwever, that if lock bit 1 is programmed, $\overline{EA}$ will be
/ latched on reset.

uld be strapped to V_{CC} for internal program
ns.

also receives the 12-volt programming enable volt-
,) during Flash programming, for parts that require
V_{PP}.

the inverting oscillator amplifier and input to the
:lock operating circuit.

ɔm the inverting oscillator amplifier.

ator Characteristics

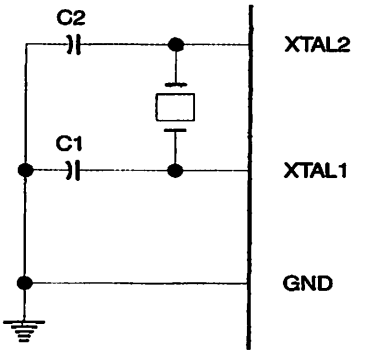
ınd XTAL2 are the input and output, respectively,
erting amplifier which can be configured for use as
ip oscillator, as shown in Figure 1. Either a quartz
ır ceramic resonator may be used. To drive the
ɔm an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

It should be noted that when idle is terminated by a hard ware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

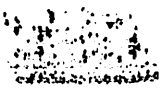
Figure 1. Oscillator Connections



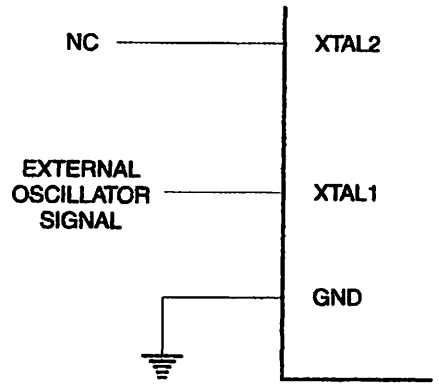
Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

; of External Pins During Idle and Power-down Modes

	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
	Internal	1	1	Data	Data	Data	Data
	External	1	1	Float	Data	Address	Data
ɔwn	Internal	0	0	Data	Data	Data	Data
ɔwn	External	0	0	Float	Data	Data	Data



2. External Clock Drive Configuration



r-down Mode

ower-down mode, the oscillator is stopped, and the on that invokes power-down is the last instruction d. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below.

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of $\overline{EA}$ be in agreement with the current logic level at that pin in order for the device to function properly.

Bit Protection Modes

Program Lock Bits				
	LB1	LB2	LB3	Protection Type
	U	U	U	No program lock features
	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash is disabled
	P	P	U	Same as mode 2, also verify is disabled
	P	P	P	Same as mode 3, also external execution is disabled

·amming the Flash

89C51 is normally shipped with the on-chip Flash ' array in the erased state (that is, contents = FFH) dy to be programmed. The programming interface ; either a high-voltage (12-volt) or a low-voltage rogram enable signal. The low-voltage program- ode provides a convenient way to program the ;1 inside the user's system, while the high-voltage ıming mode is compatible with conventional third- ısh or EPROM programmers.

89C51 is shipped with either the high-voltage or age programming mode enabled. The respective marking and device signature codes are listed in wing table.

	V_{PP} = 12V	V_{PP} = 5V
e Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
re	(030H) = 1EH (031H) = 51H (032H) =F FH	(030H) = 1EH (031H) = 51H (032H) = 05H

!9C51 code memory array is programmed byte-by- ıither programming mode. *To program any non- te in the on-chip Flash Memory, the entire memory erased using the Chip Erase Mode.*

mming Algorithm: Before programming the 1, the address, data and control signals should be ɔcording to the Flash programming mode table and and Figure 4. To program the AT89C51, take the ı steps.

t the desired memory location on the address

t the appropriate data byte on the data lines.

'ate the correct combination of control signals.

e $\overline{EA}/V_{PP}$ to 12V for the high-voltage program- ı mode.

ə ALE/$\overline{PROG}$ once to program a byte in the ı array or the lock bits. The byte-write cycle is ımed and typically takes no more than 1.5 ms. ıat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

Data Polling: The AT89C51 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/$\overline{PROG}$ low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

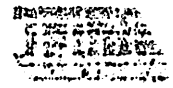
Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 51H indicates 89C51
(032H) = FFH indicates 12V programming
(032H) = 05H indicates 5V programming

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.



Programming Modes

		RST	PSEN	ALE/PROG	EA/V_{PP}	P2.6	P2.7	P3.6	P3.7
ode Data		H	L		H/12V	L	H	H	H
ode Data		H	L	H	H	L	L	H	H
ɔck	Bit - 1	H	L		H/12V	H	H	H	H
	Bit - 2	H	L		H/12V	H	H	L	L
	Bit - 3	H	L		H/12V	H	L	H	L
ıse		H	L	(1)	H/12V	H	L	L	L
gnature Byte		H	L	H	H	L	L	L	L

1. Chip Erase requires a 10 ms PROG pulse.

. Programming the Flash

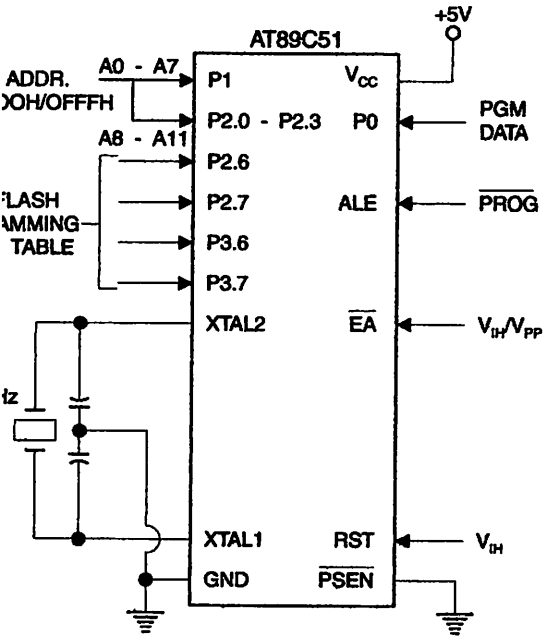
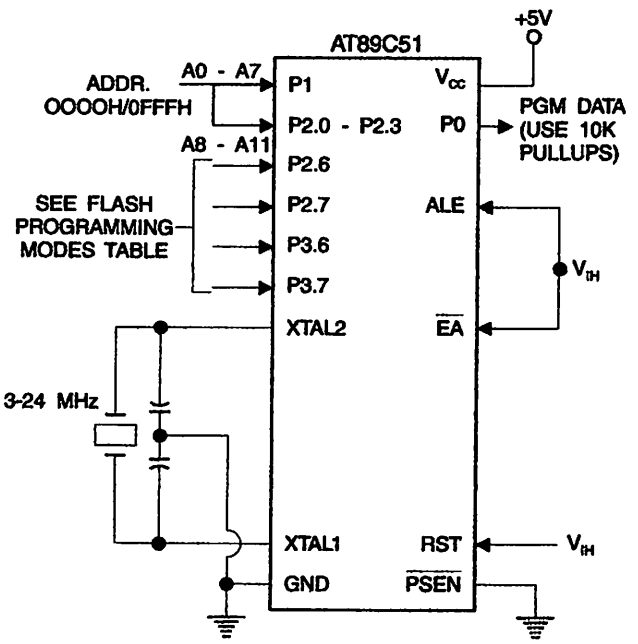
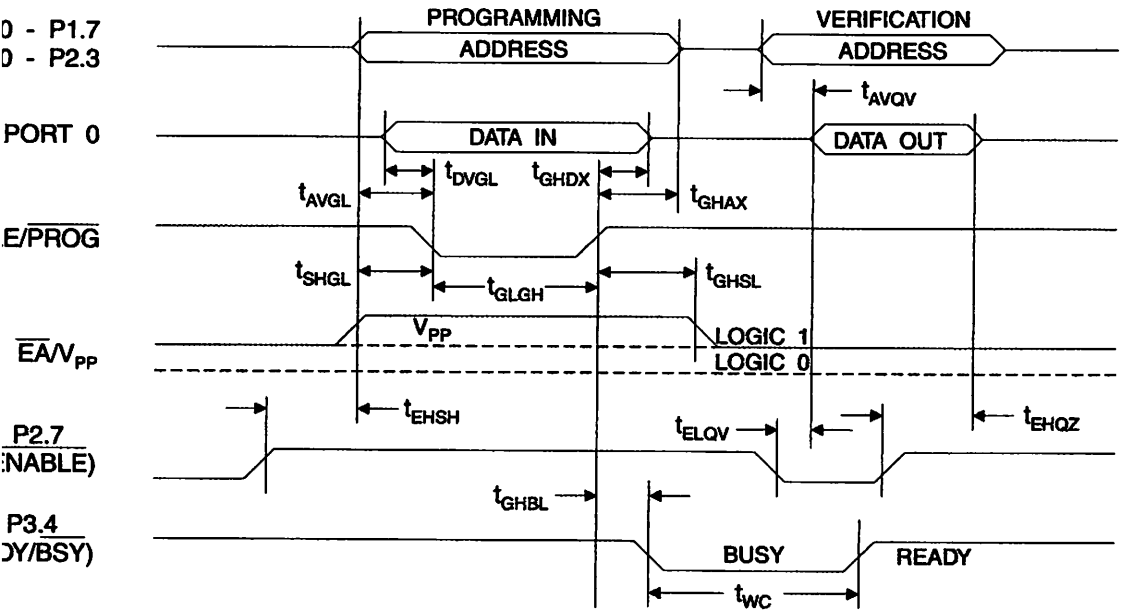


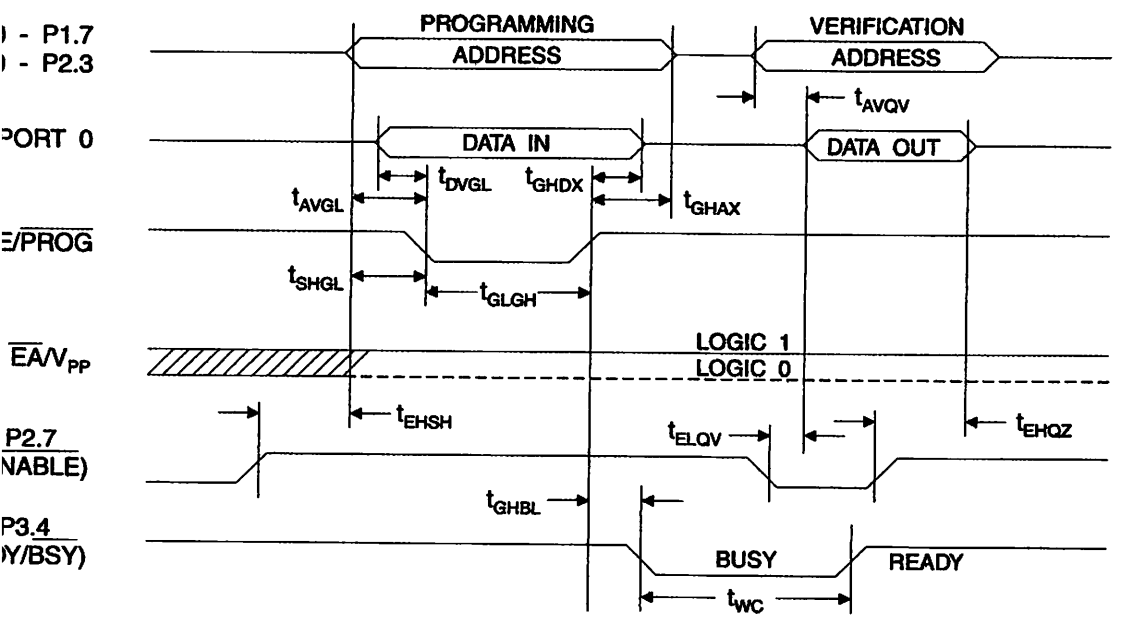
Figure 4. Verifying the Flash



Programming and Verification Waveforms - High-voltage Mode (V_{PP} = 12V)



Programming and Verification Waveforms - Low-voltage Mode (V_{PP} = 5V)

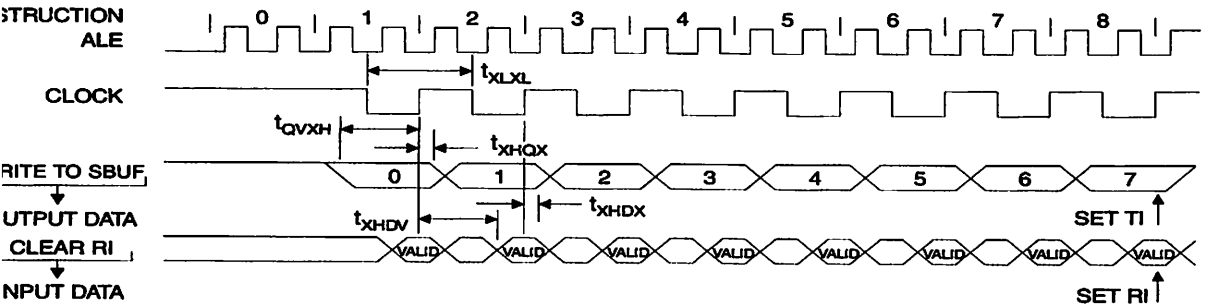


Port Timing: Shift Register Mode Test Conditions

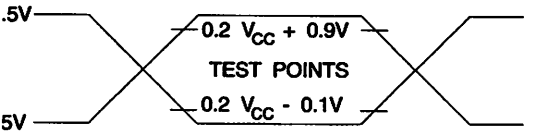
0 V ± 20%; Load Capacitance = 80 pF)

	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-117		ns
	Input Data Hold After Clock Rising Edge	0		0		ns
	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

Register Mode Timing Waveforms

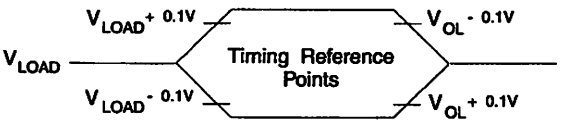


sting Input/Output Waveforms[1]

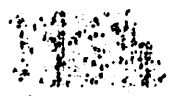


. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms[1]



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded V_{OH}/V_{OL} level occurs.



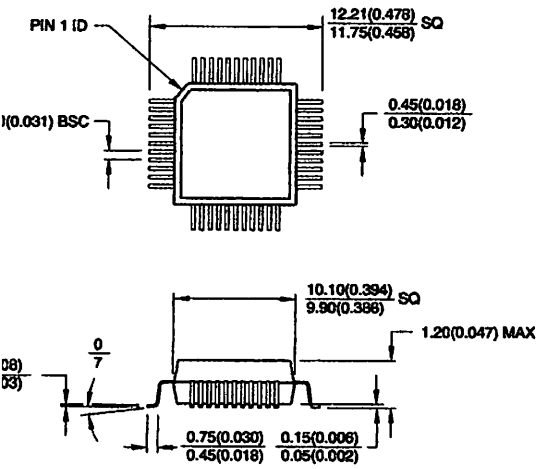
ing Information

d)	Power Supply	Ordering Code	Package	Operation Range
	5V ± 20%	AT89C51-12AC	44A	Commercial
		AT89C51-12JC	44J	(0°C to 70°C)
		AT89C51-12PC	40P6	
		AT89C51-12QC	44Q	
		AT89C51-12AI	44A	Industrial
		AT89C51-12JI	44J	(-40°C to 85°C)
		AT89C51-12PI	40P6	
		AT89C51-12QI	44Q	
	5V ± 20%	AT89C51-16AC	44A	Commercial
		AT89C51-16JC	44J	(0°C to 70°C)
		AT89C51-16PC	40P6	
		AT89C51-16QC	44Q	
		AT89C51-16AI	44A	Industrial
		AT89C51-16JI	44J	(-40°C to 85°C)
		AT89C51-16PI	40P6	
		AT89C51-16QI	44Q	
	5V ± 20%	AT89C51-20AC	44A	Commercial
		AT89C51-20JC	44J	(0°C to 70°C)
		AT89C51-20PC	40P6	
		AT89C51-20QC	44Q	
		AT89C51-20AI	44A	Industrial
		AT89C51-20JI	44J	(-40°C to 85°C)
		AT89C51-20PI	40P6	
		AT89C51-20QI	44Q	
	5V ± 20%	AT89C51-24AC	44A	Commercial
		AT89C51-24JC	44J	(0°C to 70°C)
		AT89C51-24PC	40P6	
		AT89C51-24QC	44Q	
		AT89C51-24AI	44A	Industrial
		AT89C51-24JI	44J	(-40°C to 85°C)
		AT89C51-24PI	40P6	
		AT89C51-24QI	44Q	

	Package Type
	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
	44-lead, Plastic J-leaded Chip Carrier (PLCC)
	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
	44-lead, Plastic Gull Wing Quad Flatpack (PQFP)

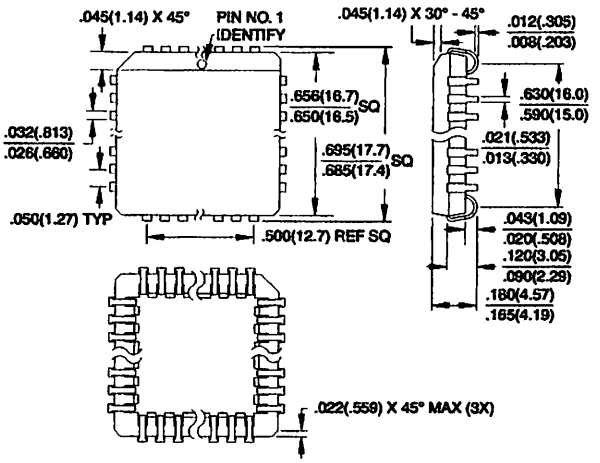
ıging Information

, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad ɔack (TQFP)
ənsions in Millimeters and (Inches)*
:C STANDARD MS-026 ACB

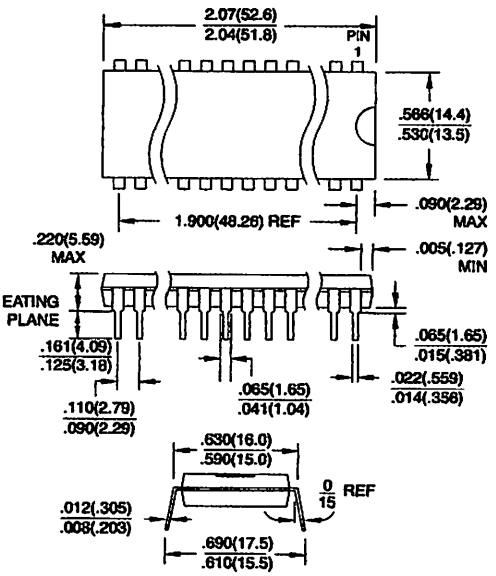


rolling dimension: millimeters

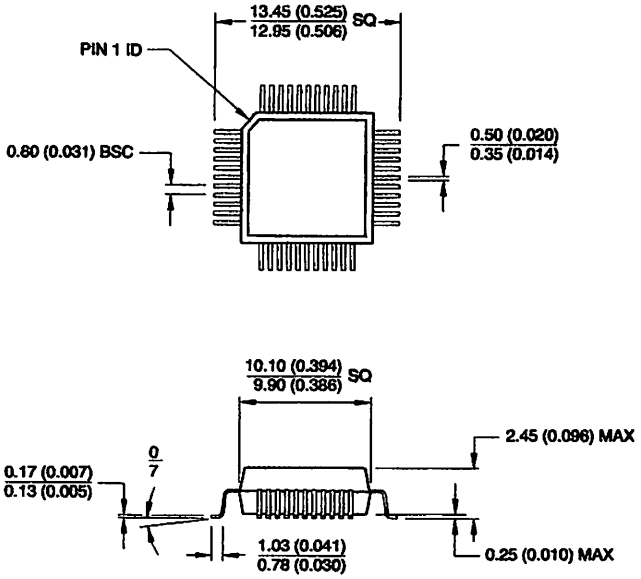
44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-018 AC



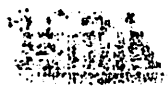
, 40-lead, 0.600" Wide, Plastic Dual Inline
age (PDIP)
ınsions in Inches and (Millimeters)



44Q, 44-lead, Plastic Quad Flat Package (PQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-022 AB



Controlling dimension: millimeters



l Headquarters

rate Headquarters
Orchard Parkway
Jose, CA 95131
(408) 441-0311
(408) 487-2600

l U.K., Ltd.
eum Business Centre
'side Way
)erley, Surrey GU15 3YL
and
(44) 1276-686-677
(44) 1276-686-697

l Asia, Ltd.
1 1219
achem Golden Plaza
)dy Road Tsimhatsui
Kowloon
Kong
(852) 2721-9778
(852) 2722-1369

l Japan K.K.
onetsu Shinkawa Bldg.
8 Shinkawa
-ku, Tokyo 104-0033
1
81) 3-3523-3551
81) 3-3523-7581

Atmel Operations

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

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

orporation 2000.

)oration makes no warranty for the use of its products, other than those expressly contained in the Company's standard war-
) is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for
which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without
does not make any commitment to update the information contained herein. No licenses to patents or other intellectual prop-
el are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are
zed for use as critical components in life support devices or systems.

'ing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

product names in this document may be trademarks of others.

Printed on recycled paper.
0265G–02/00/xM

FAIRCHILD
SEMICONDUCTOR™

October 1987
Revised April 2001

MM74C922 • MM74C923
16-Key Encoder • 20-Key Encoder

General Description

The MM74C922 and MM74C923 CMOS key encoders provide all the necessary logic to fully encode an array of SPST switches. The keyboard scan can be implemented by either an external clock or external capacitor. These encoders also have on-chip pull-up devices which permit switches with up to 50 kΩ on resistance to be used. No diodes in the switch array are needed to eliminate ghost switches. The internal debounce circuit needs only a single external capacitor and can be defeated by omitting the capacitor. A Data Available output goes to a high level when a valid keyboard entry has been made. The Data Available output returns to a low level when the entered key is released, even if another key is depressed. The Data Available will return high to indicate acceptance of the new key after a normal debounce period; this two-key roll-over is provided between any two switches.

An internal register remembers the last key pressed even after the key is released. The 3-STATE outputs provide for easy expansion and bus operation and are LPTTL compatible.

Features

- 50 kΩ maximum switch on resistance
- On or off chip clock
- On-chip row pull-up devices
- 2 key roll-over
- Keybounce elimination with single capacitor
- Last key register at outputs
- 3-STATE output LPTTL compatible
- Wide supply range: 3V to 15V
- Low power consumption

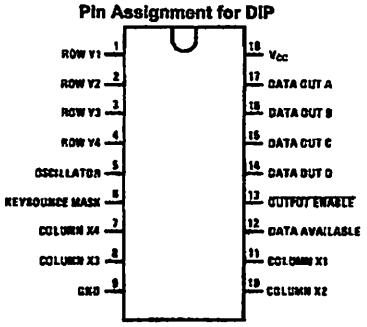
Ordering Code:

Order Number	Package Number	Package Description
MM74C922WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
MM74C922N	N18B	18-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide
MM74C923WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
MM74C923N	N20A	20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Device also available in Tape and Reel. Specify by appending suffix letter "X" to the ordering code.

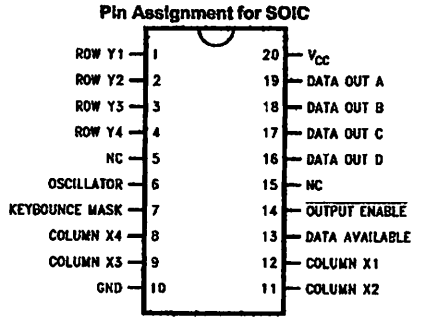
Connection Diagrams

Pin Assignment for DIP



Top View
MM74C922

Pin Assignment for SOIC

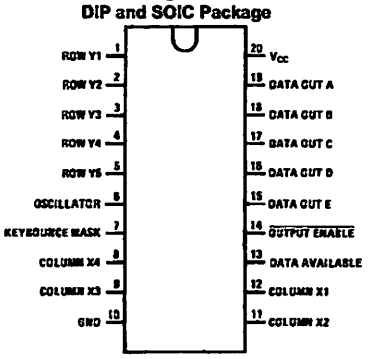


Top View
MM74C922

© 2001 Fairchild Semiconductor Corporation

Connection Diagrams (Continued)

Pin Assignment for
DIP and SOIC Package



Top View
MM74C923

Truth Tables

(Pins 0 through 11)

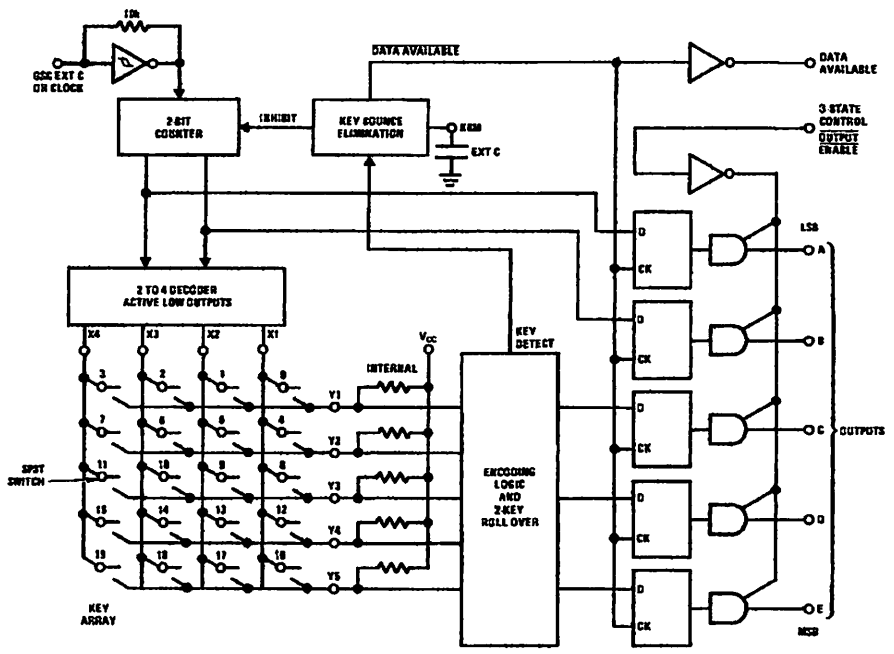
Switch Position		0 Y1, X1	1 Y1, X2	2 Y1, X3	3 Y1, X4	4 Y2, X1	5 Y2, X2	6 Y2, X3	7 Y2, X4	8 Y3, X1	9 Y3, X2	10 Y3, X3	11 Y3, X4
DATA OUT	A	0	1	0	1	0	1	0	1	0	1	0	1
	B	0	0	1	1	0	0	1	1	0	0	1	1
	C	0	0	0	0	1	1	1	1	0	0	0	0
	D	0	0	0	0	0	0	0	0	1	1	1	1
	E (Note 1)	0	0	0	0	0	0	0	0	0	0	0	0

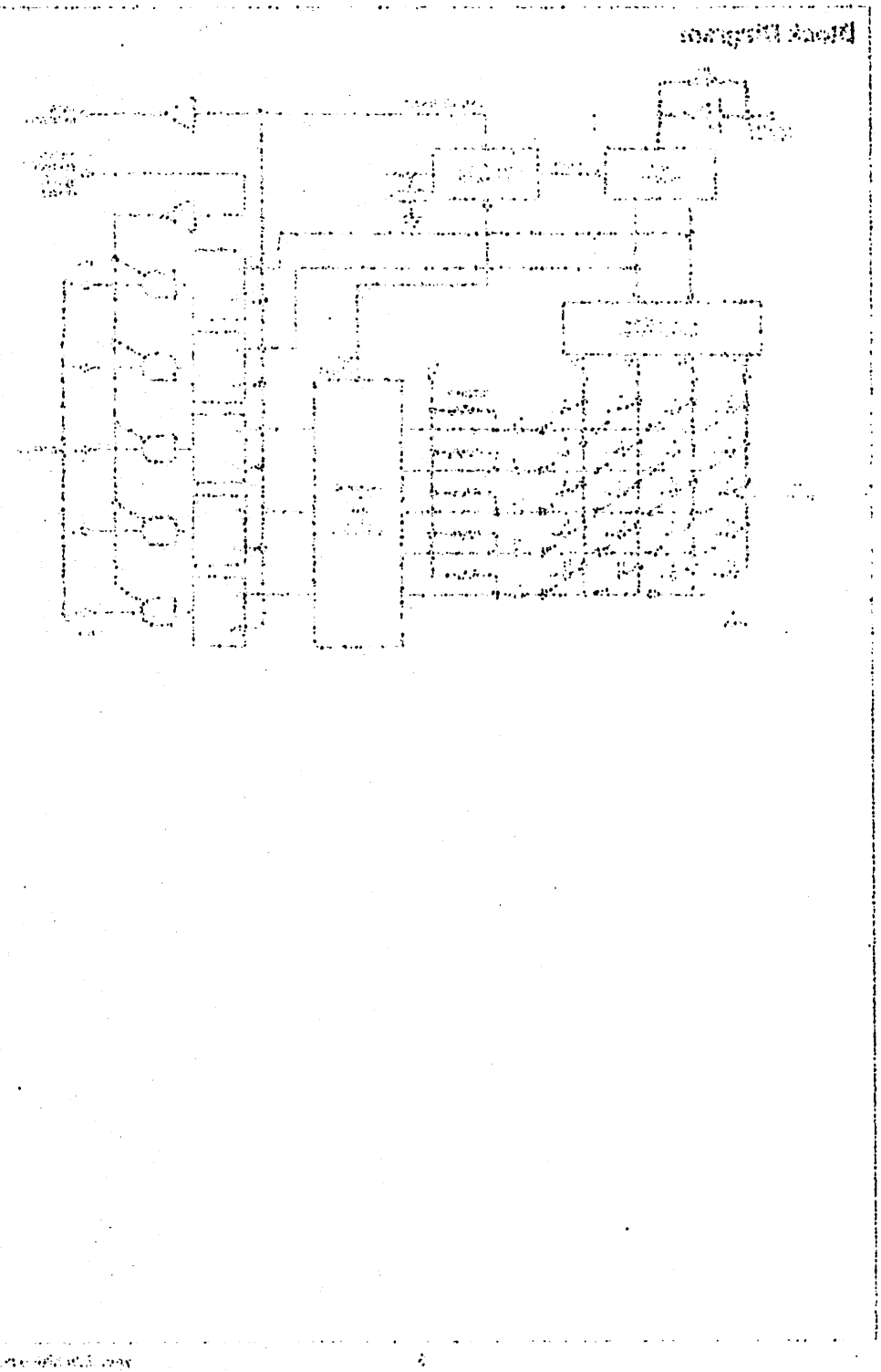
(Pins 12 through 19)

Switch Position		12 Y4, X1	13 Y4, X2	14 Y4, X3	15 Y4, X4	16 Y5(Note 1), X1	17 Y5 (Note 1), X2	18 Y5 (Note 1), X3	19 Y5 (Note 1), X4
DATA OUT	A	0	1	0	1	0	1	0	1
	B	0	0	1	1	0	0	1	1
	C	1	1	1	1	0	0	0	0
	D	1	1	1	1	0	0	0	0
	E (Note 1)	0	0	0	0	1	1	1	1

Note 1: Omit for MM74C922

Block Diagram





Absolute Maximum Ratings(Note 2)

Voltage at Any Pin	V_{CC} − 0.3V to V_{CC} + 0.3V
Operating Temperature Range	
MM74C922, MM74C923	−40°C to +85°C
Storage Temperature Range	−65°C to +150°C
Power Dissipation (P_D)	
Dual-In-Line	700 mW
Small Outline	500 mW
Operating V_{CC} Range	3V to 15V
V_{CC}	18V
Lead Temperature	
(Soldering, 10 seconds)	260°C

Note 2: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range" they are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

DC Electrical Characteristics

Min/Max limits apply across temperature range unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
CMOS TO CMOS						
V_{T+}	Positive-Going Threshold Voltage at Osc and KBM Inputs	V_{CC} = 5V, $I_{IN} \geq$ 0.7 mA	3.0	3.6	4.3	V
		V_{CC} = 10V, $I_{IN} \geq$ 1.4 mA	6.0	6.8	8.6	V
		V_{CC} = 15V, $I_{IN} \geq$ 2.1 mA	9.0	10	12.9	V
V_{T-}	Negative-Going Threshold Voltage at Osc and KBM Inputs	V_{CC} = 5V, $I_{IN} \geq$ 0.7 mA	0.7	1.4	2.0	V
		V_{CC} = 10V, $I_{IN} \geq$ 1.4 mA	1.4	3.2	4.0	V
		V_{CC} = 15V, $I_{IN} \geq$ 2.1 mA	2.1	5	6.0	V
$V_{IN(1)}$	Logical "1" Input Voltage, Except Osc and KBM Inputs	V_{CC} = 5V	3.5	4.5		V
		V_{CC} = 10V	8.0	9		V
		V_{CC} = 15V	12.5	13.5		V
$V_{IN(0)}$	Logical "0" Input Voltage, Except Osc and KBM Inputs	V_{CC} = 5V		0.5	1.5	V
		V_{CC} = 10V		1	2	V
		V_{CC} = 15V		1.5	2.5	V
I_{rp}	Row Pull-Up Current at Y1, Y2, Y3, Y4 and Y5 Inputs	V_{CC} = 5V, V_{IN} = 0.1 V_{CC}		−2	−5	µA
		V_{CC} = 10V		−10	−20	µA
		V_{CC} = 15V		−22	−45	µA
$V_{OUT(1)}$	Logical "1" Output Voltage	V_{CC} = 5V, I_O = −10 µA	4.5			V
		V_{CC} = 10V, I_O = −10 µA	9			V
		V_{CC} = 15V, I_O = −10 µA	13.5			V
$V_{OUT(0)}$	Logical "0" Output Voltage	V_{CC} = 5V, I_O = 10 µA			0.5	V
		V_{CC} = 10V, I_O = 10 µA			1	V
		V_{CC} = 15V, I_O = 10 µA			1.5	V
R_{on}	Column "ON" Resistance at X1, X2, X3 and X4 Outputs	V_{CC} = 5V, V_O = 0.5V		500	1400	Ω
		V_{CC} = 10V, V_O = 1V		300	700	Ω
		V_{CC} = 15V, V_O = 1.5V		200	500	Ω
I_{CC}	Supply Current Osc at 0V, (one Y low)	V_{CC} = 5V		0.55	1.1	mA
		V_{CC} = 10V		1.1	1.9	mA
		V_{CC} = 15V		1.7	2.6	mA
$I_{IN(1)}$	Logical "1" Input Current at $\overline{\text{Output Enable}}$	V_{CC} = 15V, V_{IN} = 15V		0.005	1.0	µA
$I_{IN(0)}$	Logical "0" Input Current at $\overline{\text{Output Enable}}$	V_{CC} = 15V, V_{IN} = 0V	−1.0	−0.005		µA
CMOS/LPTTL INTERFACE						
$V_{IN(1)}$	Except Osc and KBM Inputs	V_{CC} = 4.75V	V_{CC} − 1.5			V
$V_{IN(0)}$	Except Osc and KBM Inputs	V_{CC} = 4.75V			0.8	V
$V_{OUT(1)}$	Logical "1" Output Voltage	I_O = −360 µA, V_{CC} = 4.75V, I_O = −360 µA	2.4			V

DC Electrical Characteristics (Continued)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
$V_{OUT(0)}$	Logical "0" Output Voltage	$I_O = -360\ \mu A$ $V_{CC} = 4.75V$ $I_O = -360\ \mu A$			0.4	V
OUTPUT DRIVE (See Family Characteristics Data Sheet) (Short Circuit Current)						
I_{SOURCE}	Output Source Current (P-Channel)	$V_{CC} = 5V$, $V_{OUT} = 0V$, $T_A = 25°C$	−1.75	−3.3		mA
I_{SOURCE}	Output Source Current (P-Channel)	$V_{CC} = 10V$, $V_{OUT} = 0V$, $T_A = 25°C$	−8	−15		mA
I_{SINK}	Output Sink Current (N-Channel)	$V_{CC} = 5V$, $V_{OUT} = V_{CC}$, $T_A = 25°C$	1.75	3.6		mA
I_{SINK}	Output Sink Current (N-Channel)	$V_{CC} = 10V$, $V_{OUT} = V_{CC}$, $T_A = 25°C$	8	16		mA

AC Electrical Characteristics (Note 3)

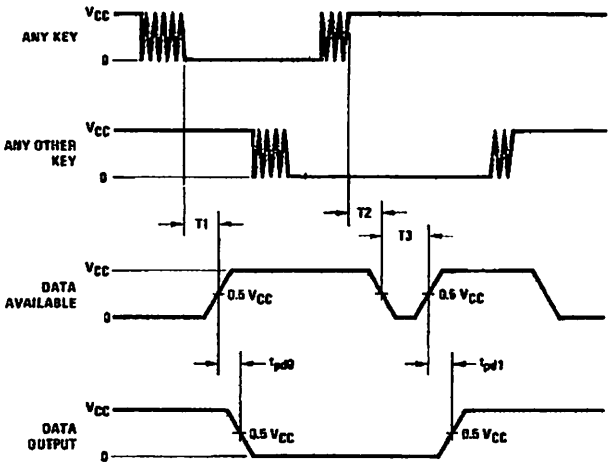
$T_A = 25°C$, $C_L = 50$ pF, unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{pd0}, t_{pd1}	Propagation Delay Time to	$C_L = 50$ pF (Figure 1)				
	Logical "0" or Logical "1"	$V_{CC} = 5V$		60	150	ns
	from D.A.	$V_{CC} = 10V$		35	80	ns
		$V_{CC} = 15V$		25	60	ns
t_{0H}, t_{1H}	Propagation Delay Time from	$R_L = 10k$, $C_L = 10$ pF (Figure 2)				
	Logical "0" or Logical "1"	$V_{CC} = 5V$, $R_L = 10k$		80	200	ns
	Into High Impedance State	$V_{CC} = 10V$, $C_L = 10$ pF		65	150	ns
		$V_{CC} = 15V$		50	110	ns
t_{H0}, t_{H1}	Propagation Delay Time from	$R_L = 10k$, $C_L = 50$ pF (Figure 2)				
	High Impedance State to a	$V_{CC} = 5V$, $R_L = 10k$		100	250	ns
	Logical "0" or Logical "1"	$V_{CC} = 10V$, $C_L = 50$ pF		55	125	ns
		$V_{CC} = 15V$		40	90	ns
C_{IN}	Input Capacitance	Any Input (Note 4)		5	7.5	pF
C_{OUT}	3-STATE Output Capacitance	Any Output (Note 4)		10		pF

Note 3: AC Parameters are guaranteed by DC correlated testing.

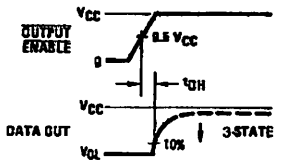
Note 4: Capacitance is guaranteed by periodic testing.

Switching Time Waveforms



T1 = T2 = RC, T3 = 0.7 RC, where R = 10k and C is external capacitor at KBM input.

FIGURE 1.



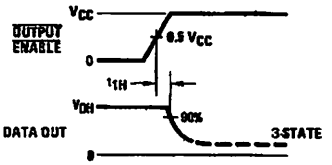
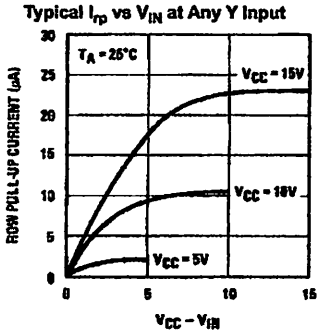


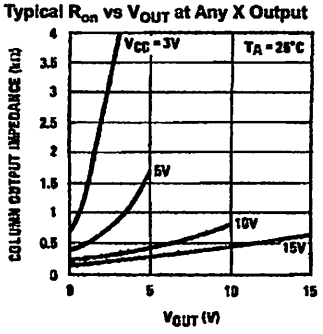
FIGURE 2.

Typical Performance Characteristics

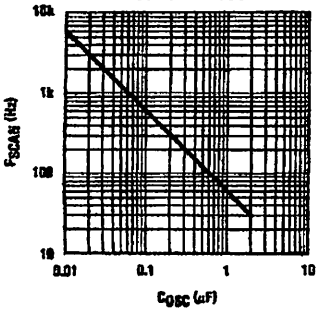
Typical I_{rp} vs V_{IN} at Any Y Input



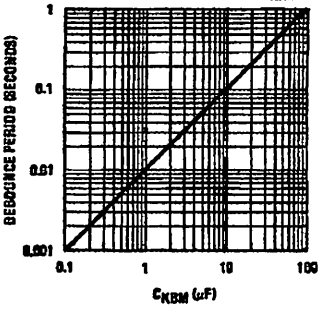
Typical R_{on} vs V_{OUT} at Any X Output



Typical F_{SCAN} vs C_{OSC}

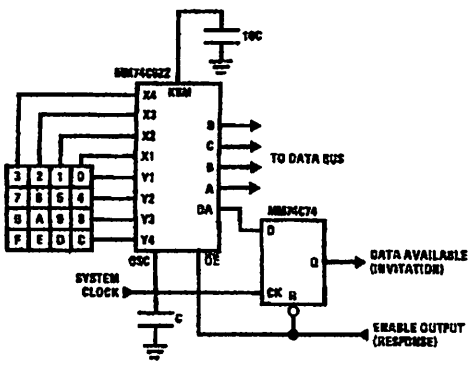


Typical Debounce Period vs C_{KBM}



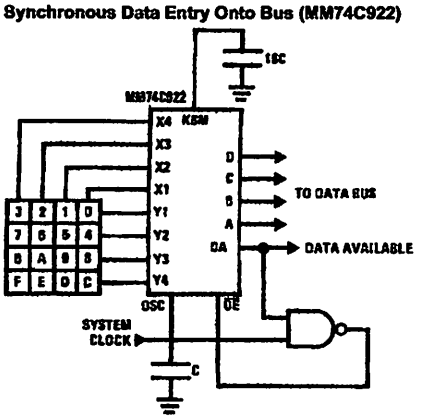
Typical Applications

Synchronous Handshake (MM74C922)



The keyboard may be synchronously scanned by omitting the capacitor at osc. and driving osc. directly if the system clock rate is lower than 10 kHz

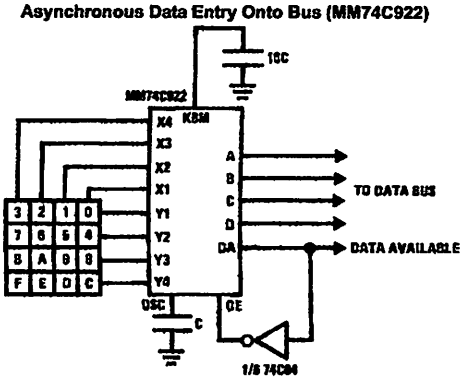
Synchronous Data Entry Onto Bus (MM74C922)



Outputs are enabled when valid entry is made and go into 3-STATE when key is released.

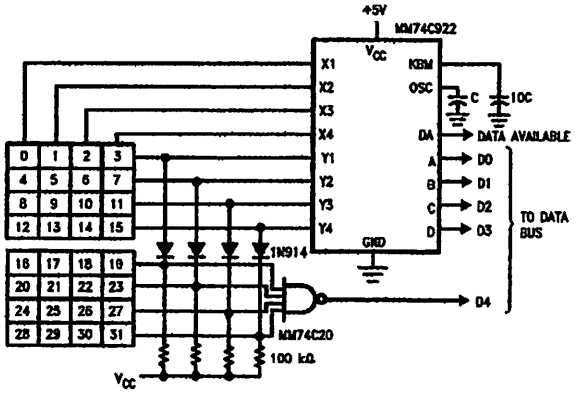
The keyboard may be synchronously scanned by omitting the capacitor at osc. and driving osc. directly if the system clock rate is lower than 10 kHz

Asynchronous Data Entry Onto Bus (MM74C922)



Outputs are in 3-STATE until key is pressed, then data is placed on bus. When key is released, outputs return to 3-STATE.

Expansion to 32 Key Encoder (MM74C922)



Theory of Operation

The MM74C922/MM74C923 Keyboard Encoders implement all the logic necessary to interface a 16 or 20 SPST key switch matrix to a digital system. The encoder will convert a key switch closer to a 4(MM74C922) or 5(MM74C923) bit nibble. The designer can control both the keyboard scan rate and the key debounce period by altering the oscillator capacitor, C_{OSE}, and the key bounce mask capacitor, C_{MSK}. Thus, the MM74C922/MM74C923's performance can be optimized for many keyboards.

The keyboard encoders connect to a switch matrix that is 4 rows by 4 columns (MM74C922) or 5 rows by 4 columns (MM74C923). When no keys are depressed, the row inputs are pulled high by internal pull-ups and the column outputs sequentially output a logic "0". These outputs are open drain and are therefore low for 25% of the time and otherwise off. The column scan rate is controlled by the oscillator input, which consists of a Schmitt trigger oscillator, a 2-bit counter, and a 2–4-bit decoder.

When a key is depressed, key 0, for example, nothing will happen when the X1 input is off, since Y1 will remain high. When the X1 column is scanned, X1 goes low and Y1 will go low. This disables the counter and keeps X1 low. Y1 going low also initiates the key bounce circuit timing and locks out the other Y inputs. The key code to be output is a combination of the frozen counter value and the decoded Y inputs. Once the key bounce circuit times out, the data is latched, and the Data Available (DAV) output goes high.

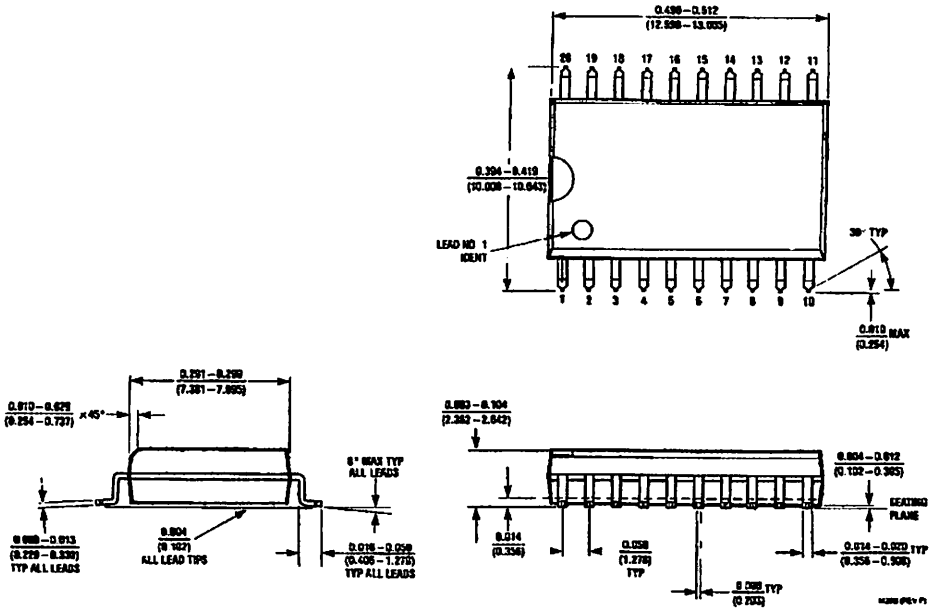
If, during the key closure the switch bounces, Y1 input will go high again, restarting the scan and resetting the key bounce circuitry. The key may bounce several times, but as soon as the switch stays low for a debounce period, the closure is assumed valid and the data is latched.

A key may also bounce when it is released. To ensure that the encoder does not recognize this bounce as another key closure, the debounce circuit must time out before another closure is recognized.

The two-key roll-over feature can be illustrated by assuming a key is depressed, and then a second key is depressed. Since all scanning has stopped, and all other Y inputs are disabled, the second key is not recognized until the first key is lifted and the key bounce circuitry has reset.

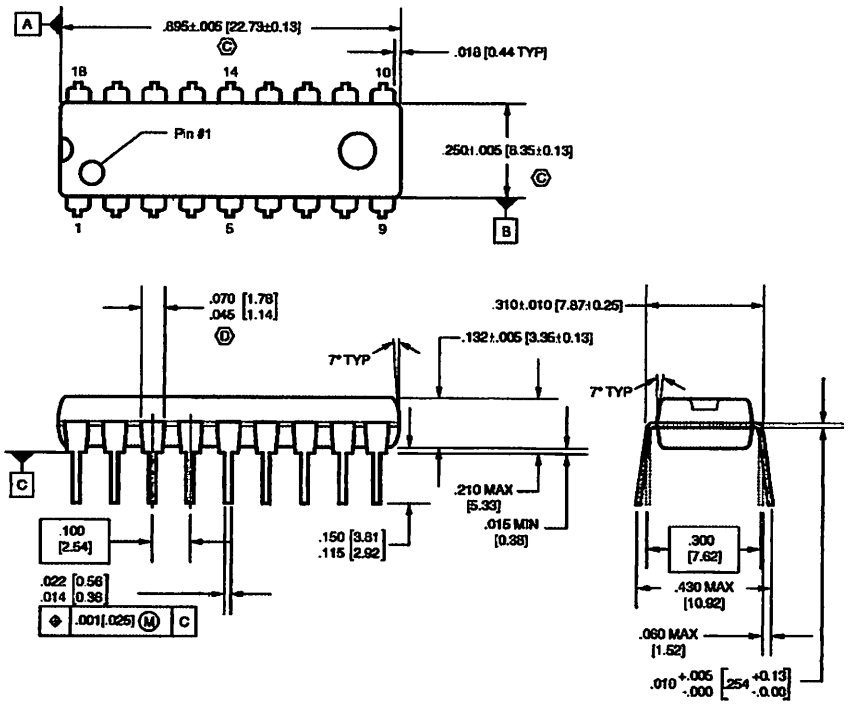
The output latches feed 3-STATE, which is enabled when the Output Enable ($\overline{OE}$) input is taken low.

Physical Dimensions inches (millimeters) unless otherwise noted



20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
Package Number M20B

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



NOTES:

A. CONFORMS TO JEDEC REGISTRATION MS-001, VARIATIONS AC, DATED 6/1993.

B. CONTROLLING DIMENSIONS ARE IN INCHES REFERENCE DIMENSIONS ARE IN MILLIMETERS.

Ⓒ DOES NOT INCLUDE MOLD FLASH OR PROTRUSIONS. MOLD FLASH OR PROTRUSIONS SHALL NOT EXCEED .010 INCHES OR 0.25MM.

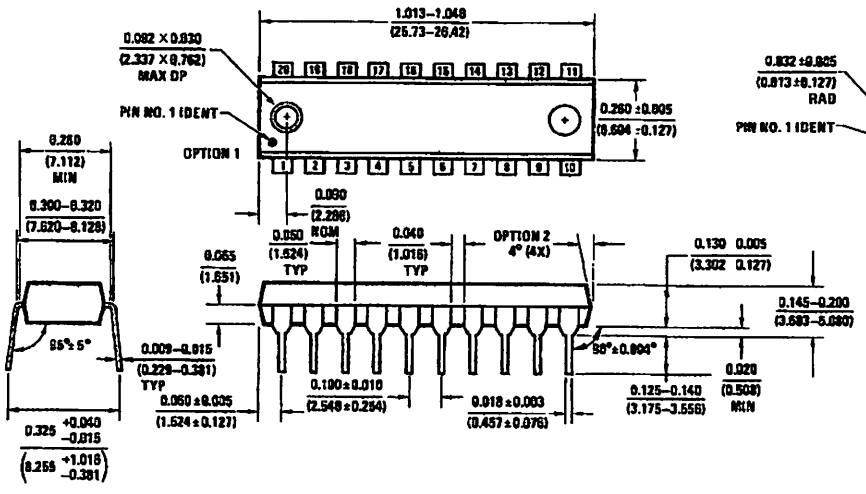
Ⓓ DOES NOT INCLUDE DAMBAR PROTRUSIONS. DAMBAR PROTRUSIONS SHALL NOT EXCEED .010 INCHES OR 0.25MM.

E. DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994.

N18BrevA

18-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide
Package Number N18B

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide
Package Number N20A

Fairchild does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and Fairchild reserves the right at any time without notice to change said circuitry and specifications.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

www.fairchildsemi.com

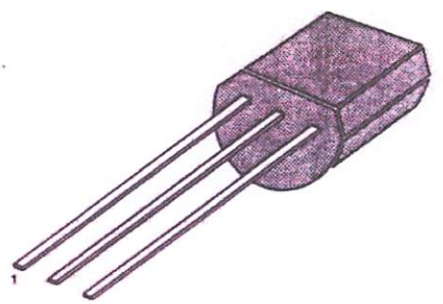
lite

A733
PNP Epitaxial Silicon Transistor

FREQUENCY AMPLIFIER

llector-Emitter Voltage: V_{CEO}=-50V
llector Dissipation: P_C(max)=250mW

TO - 92



1. Emitter 2. Collector 3. Base

lute Maximum Ratings (TA=25°C)

haracteristic	Symbol	Rating	Unit
r-Base Voltage	V_{CBO}	-60	V
r-Emitter Voltage	V_{CEO}	-50	V
Base Voltage	V_{EBO}	-5	V
r Current	I_C	-150	mA
r Dissipation	P_C	250	mW
Temperature	T_J	150	°C
Temperature	T_{STG}	-55~+150	°C

rical Characteristics (TA=25°C)

Characteristic	Symbol	Test Conditions	Min	Typ	Max	Unit
r-Base Breakdown Voltage	BV_{CBO}	I_C= -5μA, I_E=0	-60			V
r-Emitter Breakdown Voltage	BV_{CEO}	I_C= -1mA, I_B=0	-50			V
Base Breakdown Voltage	BV_{EBO}	I_E= -50μA, I_C=0	-5			V
r Cut-off Current	I_{CBO}	V_{CB}= -60V, I_E=0			0.1	μA
Cut-off Current	I_{EBO}	V_{EB}= -5V, I_C=0			01	μA
ent Gain	h_{FE}	V_{CE}= -6V, I_C= -1mA	90	200	600	
r-Emitter Saturation Voltage	$V_{CE(sat)}$	I_C= -100mA, I_B= -10mA		-0.18	-0.3	V
n Frequency	f_T	V_{CE}= -6V, I_C= -10mA				
		f= 30MHz	50	180		MHz

LASSIFICATION

sification	R	Q	P	K
h_{FE}	90-180	135-270	200-400	300-600

erprises (H.K.) Co., Ltd.
25/F., Nanyang Plaza, 57 Hung To Road, Kwun Tong, H.K.
723-3122 Fax: (852) 2723-3990 Email: info@elite-ent.com.hk

TO-92

1. Emitter 2. Collector 3. Base

Characteristic	Symbol	Rating	Unit
Collector-Base Voltage	V_{CBO}	-60	V
Collector-Emitter Voltage	V_{CEO}	-60	V
Emitter-Base Voltage	V_{EBO}	-5	V
Collector Current	I_C	-150	mA
Collector Dissipation	P_C	250	mW
Junction Temperature	T_J	150	°C
Storage Temperature	T_{STG}	-55~+150	°C

Characteristic	Symbol	Test Conditions	Min	Typ	Max	Unit
Collector-Base Breakdown Voltage	BV_{CBO}	I_C= -10μA, I_E=0	-60			V
Collector-Emitter Breakdown Voltage	BV_{CEO}	I_C= -1mA, I_B=0	-60			V
Emitter-Base Breakdown Voltage	BV_{EBO}	I_E= -10μA, I_C=0	-5			V
Collector Cut-off Current	I_{CBO}	V_{CB}= -50V, I_E=0			-0.1	μA
Emitter Cut-off Current	I_{EBO}	V_{EB}= -5V, I_C=0			-0.1	μA
DC Current Gain	h_{FE}	V_{CE}= -6V, I_C= -1mA	60	200	600	
Collector-Emitter Saturation Voltage	$V_{CE(sat)}$	I_C= -100mA, I_B= -10mA		-0.18	-0.3	V
Current Gain Bandwidth Product	f_T	V_{CE}= -6V, I_C= -10mA, f= 30MHz	50	180		MHz

CLASSIFICATION

Classification	P	Q	P	K
h_{FE}	60~120	110~270	200~400	300~600

9013 NPN SILICON TRANSISTOR

TO—92

1.EMITTER
发 射 极
2.BASE
基 极
3.COLLECTOR
集 电 极

1 2 3

TURES
征

ower dissipation （最大耗散功率）

P_{CM} : 0.625 W（Tamb=25℃）

llector current （最大集电极电流）

I_{CM} : 0.5 A

llector-base voltage （集电极--基极击穿电压）

$V_{(BR)CBO}$: 45 V

CTRICAL CHARACTERISTICS（Tamb=25℃ unless otherwise specified）
电 特 性 （环境温度 除 非 另 有 规 定）

Parameter 参 数	Symbol 符 号	Test conditions 测 试 条 件	MIN 最小值	TYP 典型值	MAX 最大值	UNIT 单位
ctor-base breakdown voltage 电 极 - - 基 极 击 穿 电 压	$V(BR)_{CBO}$	Ic= 100 μ A， I_E=0	45			V
ctor-emitter breakdown voltage 极 - - 发 射 极 击 穿 电 压	$V(BR)_{CEO}$	Ic= 0. 1 mA， I_B=0	25			V
ter-base breakdown voltage 射 极 - - 基 极 击 穿 电 压	$V(BR)_{EBO}$	I_E= 100 μ A， I_C=0	5			V
ctor cut-off current 极 - - 基 极 截 止 电 流	I_{CBO}	V_{CB}= 40 V I_E=0			0.1	μ A
ctor cut-off current 极 - - 发 射 极 截 止 电 流	I_{CEO}	V_{CE}= 20 V I_B=0			0.1	μ A
er cut-off current 极 - - 基 极 截 止 电 流	I_{EBO}	V_{EB}= 5 V， I_C=0			0.1	μ A
current gain(note) 流 电 流 增 益	$H_{FE\ (1)}$	V_{CE}= 1 V, I_C= 50 mA	64		300	
	$H_{FE\ (2)}$	V_{CE}= 1V, I_C =500 mA	40			
ctor-emitter saturation voltage 极 - - 发 射 极 饱 和 压 降	V_{CE}(sat)	I_C= 500 mA, I_B=50 mA			0.6	V
emitter saturation voltage 发射极饱和压降	V_{BE}(sat)	I_C= 500mA, I_B= 50 mA			1.2	V
emitter voltage -发射极正向电压	V_{BE}	I_E=100mA			1.4	V
ition frequency 征 频 率	f_T	V_{CE}= 6 V, I_C= 20 mA f =30MHz	150			MHz

SSIFICATION OF $H_{FE(1)}$ （分类）

	D	E	F	G	H	I
e	64-91	78-112	96-135	112-166	144-220	190-300

ing Computer Components Co., (H.K.)Ltd.
ge: http://www.wingshing.com
Tel:(852)2341 9276 Fax:(852)2797 8153
E-mail: wsccltd@hkstar.com

DATA SHEET

74HC595; 74HCT595
8-bit serial-in, serial or parallel-out shift register with output latches; 3-state

·oduct specification
ıpersedes data of 1998 Jun 04

2003 Jun 25

ıilips
:miconductors

PHILIPS

·bit serial-in, serial or parallel-out shift ¡gister with output latches; 3-state

74HC595; 74HCT595

'URES

- ¡it serial input
- ¡it serial or parallel output
- ¡rage register with 3-state outputs
- ¡ft register with direct clear
-) MHz (typical) shift out frequency
- D protection:
  M EIA/JESD22-A114-A exceeds 2000 V
  1 EIA/JESD22-A115-A exceeds 200 V.

.ICATIONS

- ¡ial-to-parallel data conversion
- mote control holding register.

DESCRIPTION

The 74HC/HCT595 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT595 is an 8-stage serial shift register with a storage register and 3-state outputs. The shift register and storage register have separate clocks.

Data is shifted on the positive-going transitions of the SH_CP input. The data in each register is transferred to the storage register on a positive-going transition of the ST_CP input. If both clocks are connected together, the shift register will always be one clock pulse ahead of the storage register.

The shift register has a serial input (DS) and a serial standard output (Q7') for cascading. It is also provided with asynchronous reset (active LOW) for all 8 shift register stages. The storage register has 8 parallel 3-state bus driver outputs. Data in the storage register appears at the output whenever the output enable input (OE) is LOW.

:K REFERENCE DATA

= 0 V; T_{amb} = 25 °C; $t_r = t_f$ = 6 ns.

IBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			74HC	74HCT	
t_{PLH}	propagation delay	C_L = 50 pF; V_{CC} = 4.5 V			
	SH_CP to Q7'		19	25	ns
	SH_CP to Qn		20	24	ns
	$\overline{MR}$ to Q7'		100	52	ns
	maximum clock frequency SH_CP and ST_CP		100	57	MHz
	input capacitance		3.5	3.5	pF
	power dissipation capacitance per package	notes 1 and 2	115	130	pF

s

- ;PD is used to determine the dynamic power dissipation (P_D in µW).
- 'D = $C_{PD} \times V_{CC}^2 \times f_i \times N + \Sigma(C_L \times V_{CC}^2 \times f_o)$ where:
- = input frequency in MHz;
-) = output frequency in MHz;
- ;L = output load capacitance in pF;
- /CC = supply voltage in Volts;
- I = total load switching outputs;
- ;($C_L \times V_{CC}^2 \times f_o$) = sum of the outputs.
- ;or 74HC595 the condition is V_I = GND to V_{CC}.
- ;or 74HCT595 the condition is V_I = GND to V_{CC} – 1.5 V.

CTION TABLE

ıote 1.

INPUT					OUTPUT		FUNCTION
CP	ST_CP	$\overline{OE}$	$\overline{MR}$	DS	Q7'	Qn	
:	X	L	L	X	L	n.c.	a LOW level on $\overline{MR}$ only affects the shift registers
:	↑	L	L	X	L	L	empty shift register loaded into storage register
:	X	H	L	X	L	Z	shift register clear; parallel outputs in high-impedance OFF-state
:	X	L	H	H	Q6'	n.c.	logic high level shifted into shift register stage 0; contents of all shift register stages shifted through, e.g. previous state of stage 6 (internal Q6') appears on the serial output (Q7')
:	↑	L	H	X	n.c.	Qn'	contents of shift register stages (internal Qn') are transferred to the storage register and parallel output stages
:	↑	L	H	X	Q6'	Qn'	contents of shift register shifted through; previous contents of the shift register is transferred to the storage register and the parallel output stages

I = HIGH voltage level;
= LOW voltage level;
= LOW-to-HIGH transition;
= HIGH-to-LOW transition;
= high-impedance OFF-state;
.c. = no change;
= don't care.

ERING INFORMATION

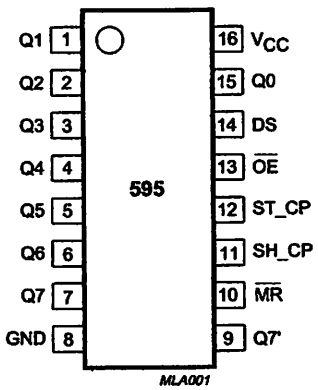
PE NUMBER	PACKAGE				
	TEMPERATURE RANGE	PINS	PACKAGE	MATERIAL	CODE
C595N	–40 to +125 °C	16	DIP16	plastic	SOT38-4
CT595N	–40 to +125 °C	16	DIP16	plastic	SOT38-4
C595D	–40 to +125 °C	16	SO16	plastic	SOT109-1
CT595D	–40 to +125 °C	16	SO16	plastic	SOT109-1
C595DB	–40 to +125 °C	16	SSOP16	plastic	SOT338-1
CT595DB	–40 to +125 °C	16	SSOP16	plastic	SOT338-1
C595PW	–40 to +125 °C	16	TSSOP16	plastic	SOT403-1
CT595PW	–40 to +125 °C	16	TSSOP16	plastic	SOT403-1
C595BQ	–40 to +125 °C	16	DHVQFN16	plastic	SOT763-1
CT595BQ	–40 to +125 °C	16	DHVQFN16	plastic	SOT763-1

bit serial-in, serial or parallel-out shift gister with output latches; 3-state

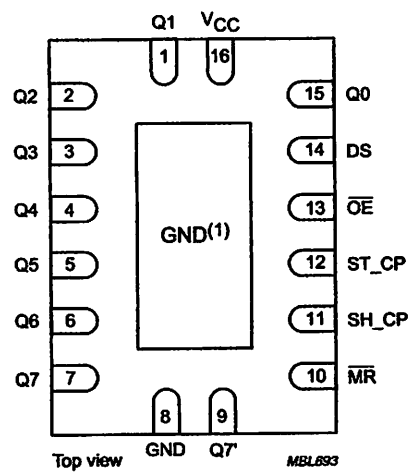
74HC595; 74HCT595

ING

PIN	SYMBOL	DESCRIPTION
1	Q1	parallel data output
2	Q2	parallel data output
3	Q3	parallel data output
4	Q4	parallel data output
5	Q5	parallel data output
6	Q6	parallel data output
7	Q7	parallel data output
8	GND	ground (0 V)
9	Q7'	serial data output
10	$\overline{MR}$	master reset (active LOW)
11	SH_CP	shift register clock input
12	ST_CP	storage register clock input
13	$\overline{OE}$	output enable (active LOW)
14	DS	serial data input
15	Q0	parallel data output
16	V_{CC}	positive supply voltage



ig.1 Pin configuration DIP16, SO16 and (T)SSOP16.



(1) The die substrate is attached to this pad using conductive die attach material. It can not be used as a supply pin or input.

Fig.2 Pin configuration DHVQFN16.

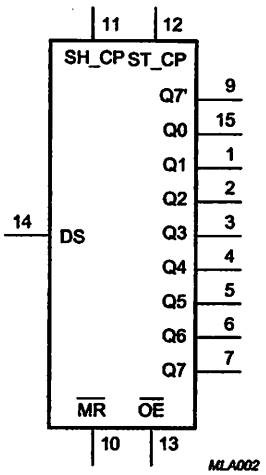


Fig.3 Logic symbol.

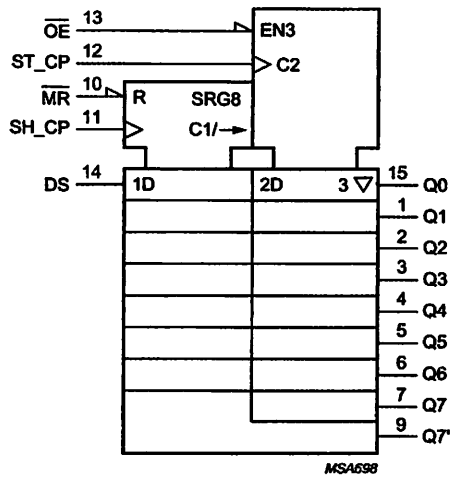


Fig.4 IEC logic symbol.

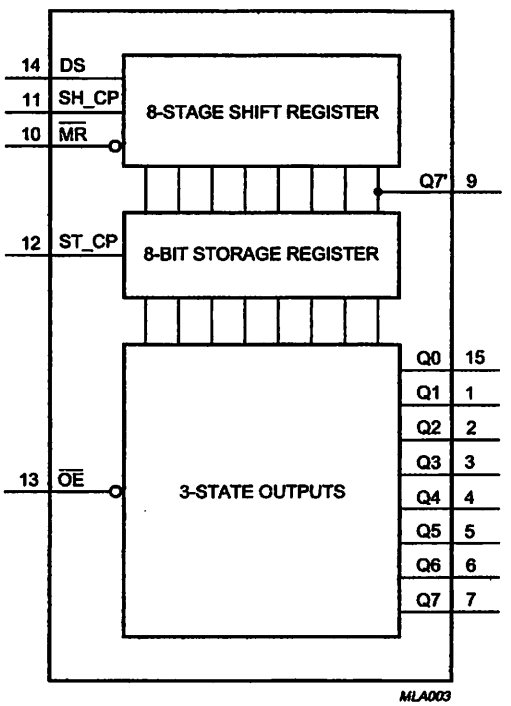


Fig.5 Functional diagram.

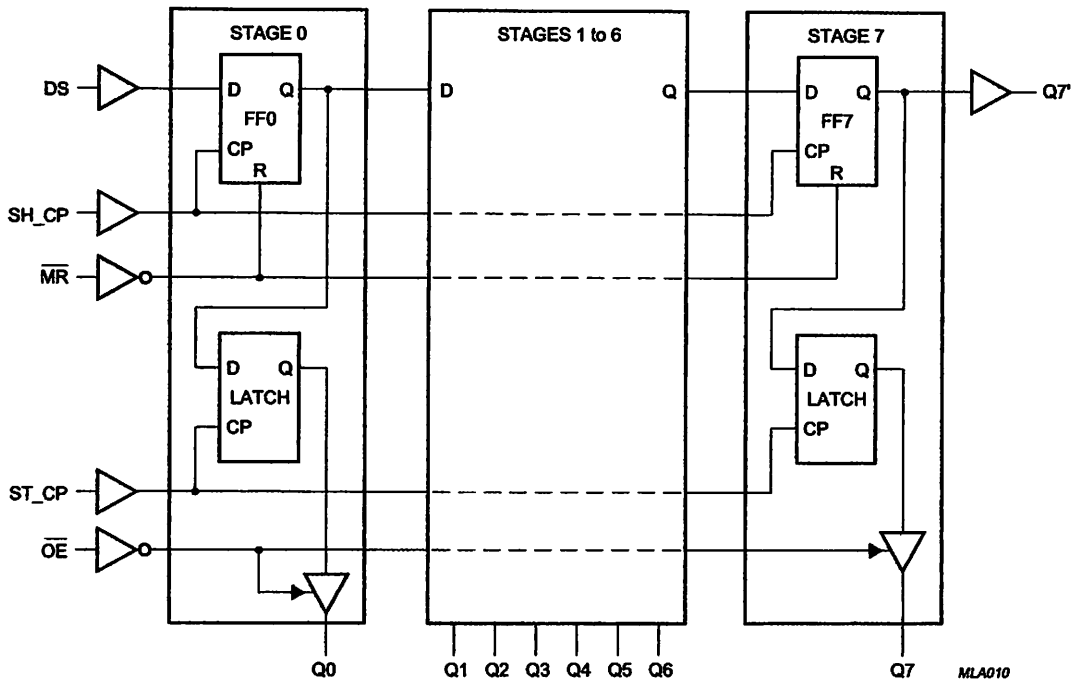


Fig.6 Logic diagram.

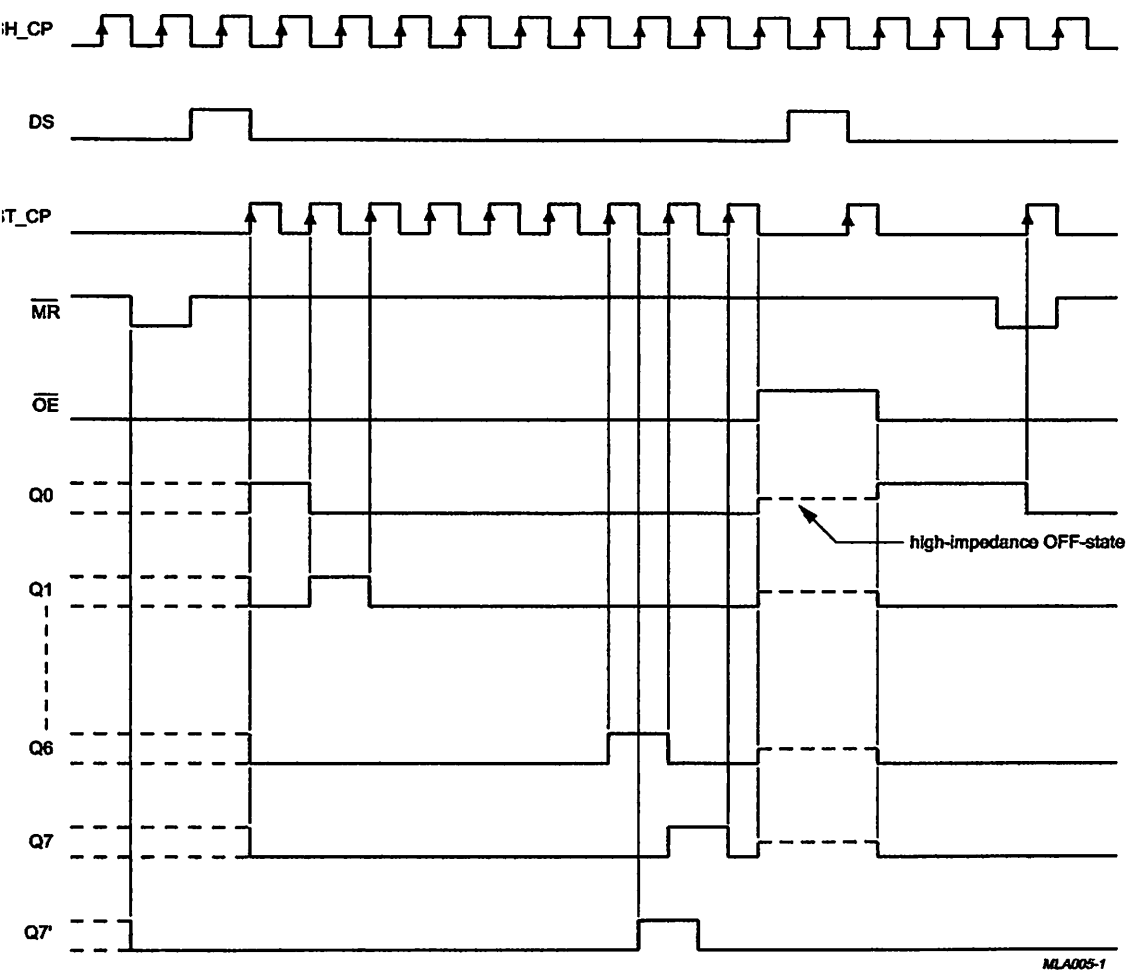


Fig.6 Timing diagram.

OMMENDED OPERATING CONDITIONS

IBOL	PARAMETER	CONDITIONS	74HC			74HCT			UNIT
			MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
	supply voltage		2.0	5.0	6.0	4.5	5.0	5.5	V
	input voltage		0	–	V_{CC}	0	–	V_{CC}	V
	output voltage		0	–	V_{CC}	0	–	V_{CC}	V
)	ambient temperature		–40	–	+125	–40	–	+125	°C
	input rise and fall time	V_{CC} = 2.0 V	–	–	1000	–	–	–	ns
		V_{CC} = 4.5 V	–	6.0	500	–	6.0	500	ns
		V_{CC} = 6.0 V	–	–	400	–	–	–	ns

ΓED VALUES

cordance with the Absolute Maximum Rating System (IEC 60134); voltages are referenced to GND (ground = 0 V).

IBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
	supply voltage		–0.5	+7.0	V
	input diode current	V_I < –0.5 V to V_I > V_{CC} + 0.5 V	–	±20	mA
	output diode current	V_O < –0.5 V to V_O > V_{CC} + 0.5 V	–	±20	mA
	output source or sink current	V_O = –0.5 V to V_{CC} + 0.5 V			
		Q7' standard output	–	±25	mA
		Qn bus driver outputs	–	±35	mA
I_{GND}	V_{CC} or GND current		–	±70	mA
	storage temperature		–65	+150	°C
	power dissipation	T_{amb} = –40 to +125 °C; note 1	–	500	mW

or DIP16 packages: above 70 °C derate linearly with 12 mW/K.
or SO16 packages: above 70 °C derate linearly with 8 mW/K.
or SSOP16 packages: above 60 °C derate linearly with 5.5 mW/K.
or TSSOP16 packages: above 60 °C derate linearly with 5.5 mW/K.
or DHVQFN16 packages: above 60 °C derate linearly with 4.5 mW/K.

HARACTERISTICS

74HC

ommended operating conditions; voltages are referenced to GND (ground = 0 V).

MBOL	PARAMETER	TEST CONDITIONS		MIN.	TYP.	MAX.	UNIT
		OTHER	V_{CC} (V)				
= –40 to +85 °C; note 1							
	HIGH-level input voltage		2.0	1.5	1.2	–	V
			4.5	3.15	2.4	–	V
			6.0	4.2	3.2	–	V
	LOW-level input voltage		2.0	–	0.8	0.5	V
			4.5	–	2.1	1.35	V
			6.0	–	2.8	1.8	V
	HIGH-level output voltage	$V_I = V_{IH}$ or V_{IL}					
		all outputs					
		I_O = –20 µA	2.0	1.9	2.0	–	V
			4.5	4.4	4.5	–	V
			6.0	5.9	6.0	–	V
		Q7' standard output					
		I_O = –4.0 mA	4.5	3.84	4.32	–	V
		I_O = –5.2 mA	6.0	5.34	5.81	–	V
		Qn bus driver outputs					
		I_O = –6.0 mA	4.5	3.84	4.32	–	V
		I_O = –7.8 mA	6.0	5.34	5.81	–	V
	LOW-level output voltage	$V_I = V_{IH}$ or V_{IL}					
		all outputs					
		I_O = 20 µA	2.0	–	0	0.1	V
			4.5	–	0	0.1	V
			6.0	–	0	0.1	V
		Q7' standard output					
		I_O = 4.0 mA	4.5	–	0.15	0.33	V
		I_O = 5.2 mA	6.0	–	0.16	0.33	V
		Qn bus driver outputs					
		I_O = 6.0 mA	4.5	–	0.16	0.33	V
		I_O = 7.8 mA	6.0	–	0.16	0.33	V
	input leakage current	$V_I = V_{CC}$ or GND	6.0	–	–	±1.0	µA
	3-state output OFF-state current	$V_I = V_{IH}$ or V_{IL}; $V_O = V_{CC}$ or GND	6.0	–	–	±5.0	µA
	quiescent supply current	$V_I = V_{CC}$ or GND; I_O = 0	6.0	–	–	80	µA

MBOL	PARAMETER	TEST CONDITIONS		MIN.	TYP.	MAX.	UNIT
		OTHER	V_{CC} (V)				
= –40 to +125 °C							
	HIGH-level input voltage		2.0	1.5	–	–	V
			4.5	3.15	–	–	V
			6.0	4.2	–	–	V
	LOW-level input voltage		2.0	–	–	0.5	V
			4.5	–	–	1.35	V
			6.0	–	–	1.8	V
	HIGH-level output voltage	$V_I = V_{IH}$ or V_{IL}					
		all outputs					
		$I_O = -20$ µA	2.0	1.9	–	–	V
			4.5	4.4	–	–	V
			6.0	5.9	–	–	V
		Q7' standard output					
		$I_O = -4.0$ mA	4.5	3.7	–	–	V
		$I_O = -5.2$ mA	6.0	5.2	–	–	V
		Qn bus driver outputs					
		$I_O = -6.0$ mA	4.5	3.7	–	–	V
		$I_O = -7.8$ mA	6.0	5.2	–	–	V
	LOW-level output voltage	$V_I = V_{IH}$ or V_{IL}					
		all outputs					
		$I_O = 20$ µA	4.5	–	–	0.1	V
		Q7' standard output					
		$I_O = 4.0$ mA	4.5	–	–	0.4	V
		Qn bus driver outputs					
		$I_O = 6.0$ mA	4.5	–	–	0.4	V
	input leakage current	$V_I = V_{CC}$ or GND	5.5	–	–	±1.0	µA
	3-state output OFF-state current	$V_I = V_{IH}$ or V_{IL}; $V_O = V_{CC}$ or GND	5.5	–	–	±10.0	µA
	quiescent supply current	$V_I = V_{CC}$ or GND; $I_O = 0$	5.5	–	–	160	µA

ll typical values are measured at T_{amb} = 25 °C.

74HCT

commended operating conditions; voltages are referenced to GND (ground = 0 V); $t_r = t_f = 6$ ns; $C_L = 50$ pF.

MBOL	PARAMETER	TEST CONDITIONS		MIN.	TYP.	MAX.	UNIT
		OTHER	V_{CC} (V)				
= –40 to +85 °C; note 1							
	HIGH-level input voltage		4.5 to 5.5	2.0	1.6	–	V
	LOW-level input voltage		4.5 to 5.5	–	1.2	0.8	V
	HIGH-level output voltage	$V_I = V_{IH}$ or V_{IL}					
		all outputs $I_O = -20$ µA	4.5	4.4	4.5	–	V
		Q7' standard output $I_O = -4.0$ mA	4.5	3.84	4.32	–	V
		Qn bus driver outputs $I_O = -6.0$ mA	4.5	3.7	4.32	–	V
	LOW-level output voltage	$V_I = V_{IH}$ or V_{IL}					
		all outputs $I_O = 20$ µA	4.5	–	0	0.33	V
		Q7' standard output $I_O = 4.0$ mA	4.5	–	0.15	0.33	V
		Qn bus driver outputs $I_O = 6.0$ mA	4.5	–	0.16	0.33	V
	input leakage current	$V_I = V_{CC}$ or GND	5.5	–	–	±1.0	µA
	3-state output OFF-state current	$V_I = V_{IH}$ or V_{IL}; $V_O = V_{CC}$ or GND	5.5	–	–	±5.0	µA
	quiescent supply current	$V_I = V_{CC}$ or GND; $I_O = 0$	5.5	–	–	80	µA
	additional supply current per input	$V_I = V_{CC} - 2.1$ V; $I_O = 0$; note 2	4.5 to 5.5	–	100	450	µA

rMBOL	PARAMETER	TEST CONDITIONS		MIN.	TYP.	MAX.	UNIT
		OTHER	V_{CC} (V)				
₀ = –40 to +125 °C							
	HIGH-level input voltage		4.5 to 5.5	2.0	–	–	V
	LOW-level input voltage		4.5 to 5.5	–	–	0.8	V
	HIGH-level output voltage	$V_I = V_{IH}$ or V_{IL}					
		all outputs I_O = –20 μA	4.5	4.4	–	–	V
		Q7' standard output I_O = –4.0 mA	4.5	3.7	–	–	V
		Qn bus driver outputs I_O = –6.0 mA	4.5	3.7	–	–	V
	LOW-level output voltage	$V_I = V_{IH}$ or V_{IL}					
		all outputs I_O = 20 μA	4.5	–	–	0.1	V
		Q7' standard output I_O = 4.0 mA	4.5	–	–	0.4	V
		Qn bus driver outputs I_O = 6.0 mA	4.5	–	–	0.4	V
	input leakage current	$V_I = V_{CC}$ or GND	5.5	–	–	±1.0	μA
	3-state output OFF-state current	$V_I = V_{IH}$ or V_{IL}; $V_O = V_{CC}$ or GND	5.5	–	–	±10.0	μA
	quiescent supply current	$V_I = V_{CC}$ or GND; I_O = 0	5.5	–	–	160	μA
	additional supply current per input	$V_I = V_{CC}$ – 2.1 V; I_O = 0; note 2	4.5 to 5.5	–	–	490	μA

ɜ

ll typical values are measured at T_{amb} = 25 °C.

he value of additional quiescent supply current (ΔI_{CC}) for a unit load of 1 is given here. To determine ΔI_{CC} per input, ıultiply this value by the unit load coefficient per input pin:

. pin DS: 0.25

. pins $\overline{MR}$, SH_CP, ST_CP and $\overline{OE}$: 1.50.

HARACTERISTICS

ly 74HC

= 0 V; $t_r = t_f$ = 6 ns; C_L = 50 pF.

MBOL	PARAMETER	TEST CONDITIONS		MIN.	TYP.	MAX.	UNIT
		WAVEFORMS	V_{CC} (V)				
= 25 °C							
t_{PLH}	propagation delay SH_CP to Q7'	see Fig.7	2.0	–	52	160	ns
			4.5	–	19	32	ns
			6.0	–	15	27	ns
	propagation delay ST_CP to Qn	see Fig.8	2.0	–	55	175	ns
			4.5	–	20	35	ns
			6.0	–	16	30	ns
	propagation delay $\overline{MR}$ to Q7'	see Fig.10	2.0	–	47	175	ns
			4.5	–	17	35	ns
			6.0	–	14	30	ns
t_{PZL}	3-state output enable time $\overline{OE}$ to Qn	see Fig.11	2.0	–	47	150	ns
			4.5	–	17	30	ns
			6.0	–	14	26	ns
t_{PLZ}	3-state output disable time $\overline{OE}$ to Qn	see Fig.11	2.0	–	41	150	ns
			4.5	–	15	30	ns
			6.0	–	12	26	ns
	shift clock pulse width HIGH or LOW	see Fig.7	2.0	75	17	–	ns
			4.5	15	6	–	ns
			6.0	13	5	–	ns
	storage clock pulse width HIGH or LOW	see Fig.8	2.0	75	11	–	ns
			4.5	15	4	–	ns
			6.0	13	3	–	ns
	master reset pulse width LOW	see Fig.10	2.0	75	17	–	ns
			4.5	15	6.0	–	ns
			6.0	13	5.0	–	ns
	set-up time DS to SH_CP	see Fig.9	2.0	50	11	–	ns
			4.5	10	4.0	–	ns
			6.0	9.0	3.0	–	ns
	set-up time SH_CP to ST_CP	see Fig.8	2.0	75	22	–	ns
			4.5	15	8	–	ns
			6.0	13	7	–	ns
	hold time DS to SH_CP	see Fig.9	2.0	+3	–6	–	ns
			4.5	+3	–2	–	ns
			6.0	+3	–2	–	ns

'MBOL	PARAMETER	TEST CONDITIONS		MIN.	TYP.	MAX.	UNIT
		WAVEFORMS	V_{CC} (V)				
	removal time $\overline{MR}$ to SH_CP	see Fig.10	2.0	+50	–19	–	ns
			4.5	+10	–7	–	ns
			6.0	+9	–6	–	ns
	maximum clock pulse frequency SH_CP or ST_CP	see Figs 7 and 8	2.0	9	30	–	MHz
			4.5	30	91	–	MHz
			6.0	35	108	–	MHz
, = –40 to +85 °C							
t_{PLH}	propagation delay SH_CP to Q7'	see Fig.7	2.0	–	–	200	ns
			4.5	–	–	40	ns
			6.0	–	–	34	ns
	propagation delay ST_CP to An	see Fig.8	2.0	–	–	220	ns
			4.5	–	–	44	ns
			6.0	–	–	37	ns
	propagation delay $\overline{MR}$ to Q7'	see Fig.10	2.0	–	–	220	ns
			4.5	–	–	44	ns
			6.0	–	–	37	ns
t_{PZL}	3-state output enable time $\overline{OE}$ to Qn	see Fig.11	2.0	–	–	190	ns
			4.5	–	–	38	ns
			6.0	–	–	33	ns
t_{PLZ}	3-state output disable time $\overline{OE}$ to Qn	see Fig.11	2.0	–	–	190	ns
			4.5	–	–	38	ns
			6.0	–	–	33	ns
	shift clock pulse width HIGH or LOW	see Fig.7	2.0	95	–	–	ns
			4.5	19	–	–	ns
			6.0	16	–	–	ns
	storage clock pulse width HIGH or LOW	see Fig.8	2.0	95	–	–	ns
			4.5	19	–	–	ns
			6.0	16	–	–	ns
	master reset pulse width LOW	see Fig.10	2.0	95	–	–	ns
			4.5	19	–	–	ns
			6.0	16	–	–	ns
	set-up time DS to SH_CP	see Fig.9	2.0	65	–	–	ns
			4.5	13	–	–	ns
			6.0	11	–	–	ns
	set-up time SH_CP to ST_CP	see Fig.8	2.0	95	–	–	ns
			4.5	19	–	–	ns
			6.0	16	–	–	ns

MBOL	PARAMETER	TEST CONDITIONS		MIN.	TYP.	MAX.	UNIT
		WAVEFORMS	V_{CC} (V)				
	hold time DS to SH_CP	see Fig.9	2.0	3	–	–	ns
			4.5	3	–	–	ns
			6.0	3	–	–	ns
	removal time $\overline{MR}$ to SH_CP	see Fig.10	2.0	65	–	–	ns
			4.5	13	–	–	ns
			6.0	11	–	–	ns
	maximum clock pulse frequency SH_CP or ST_CP	see Figs 7 and 8	2.0	4.8	–	–	MHz
			4.5	24	–	–	MHz
			6.0	28	–	–	MHz
, = –40 to +125 °C							
t_{PLH}	propagation delay SH_CP to Q7'	see Fig.7	2.0	–	–	240	ns
			4.5	–	–	48	ns
			6.0	–	–	41	ns
	propagation delay ST_CP to Qn	see Fig.8	2.0	–	–	265	ns
			4.5	–	–	53	ns
			6.0	–	–	45	ns
	propagation delay $\overline{MR}$ to Q7'	see Fig.10	2.0	–	–	265	ns
			4.5	–	–	53	ns
			6.0	–	–	45	ns
t_{PZL}	3-state output enable time $\overline{OE}$ to Qn	see Fig.11	2.0	–	–	225	ns
			4.5	–	–	45	ns
			6.0	–	–	38	ns
t_{PLZ}	3-state output disable time $\overline{OE}$ to Qn	see Fig.11	2.0	–	–	225	ns
			4.5	–	–	45	ns
			6.0	–	–	38	ns
	shift clock pulse width HIGH or LOW	see Fig.7	2.0	110	–	–	ns
			4.5	22	–	–	ns
			6.0	19	–	–	ns
	storage clock pulse width HIGH or LOW	see Fig.8	2.0	110	–	–	ns
			4.5	22	–	–	ns
			6.0	19	–	–	ns
	master reset pulse width LOW	see Fig.10	2.0	110	–	–	ns
			4.5	22	–	–	ns
			6.0	19	–	–	ns

MBOL.	PARAMETER	TEST CONDITIONS		MIN.	TYP.	MAX.	UNIT
		WAVEFORMS	V_{CC} (V)				
	set-up time DS to SH_CP	see Fig.9	2.0	75	–	–	ns
			4.5	15	–	–	ns
			6.0	13	–	–	ns
	set-up time SH_CP to ST_CP	see Fig.8	2.0	110	–	–	ns
			4.5	22	–	–	ns
			6.0	19	–	–	ns
	hold time DS to SH_CP	see Fig.9	2.0	3	–	–	ns
			4.5	3	–	–	ns
			6.0	3	–	–	ns
	removal time $\overline{MR}$ to SH_CP	see Fig.10	2.0	75	–	–	ns
			4.5	15	–	–	ns
			6.0	13	–	–	ns
	maximum clock pulse frequency SH_CP or ST_CP	see Figs 7 and 8	2.0	4	–	–	MHz
			4.5	20	–	–	MHz
			6.0	24	–	–	MHz

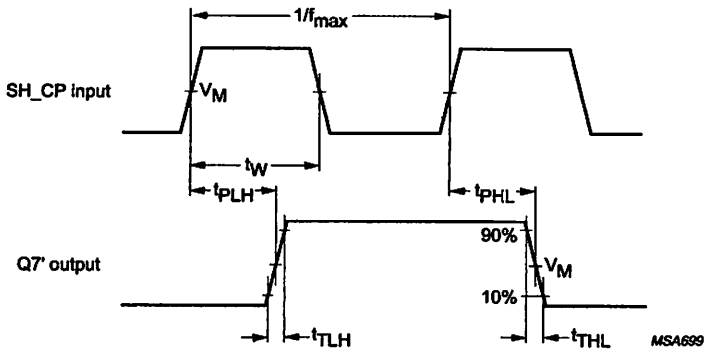
ly 74HCT

= 0 V; $t_r = t_f$ = 6 ns; C_L = 50 pF.

'MBOL	PARAMETER	TEST CONDITIONS		MIN.	TYP.	MAX.	UNIT
		WAVEFORMS	V_{CC} (V)				
, = 25 °C							
t_{PLH}	propagation delay SH_CP to Q7'	see Fig.7	4.5	–	25	42	ns
	propagation delay ST_CP to Qn	see Fig.8	4.5	–	24	40	ns
	propagation delay $\overline{MR}$ to Q7'	see Fig.10	4.5	–	23	40	ns
t_{PZL}	3-state output enable time $\overline{OE}$ to Qn	see Fig.11	4.5	–	21	35	ns
t_{PLZ}	3-state output disable time $\overline{OE}$ to Qn	see Fig.11	4.5	–	18	30	ns
	shift clock pulse width HIGH or LOW	see Fig.7	4.5	16	6	–	ns
	storage clock pulse width HIGH or LOW	see Fig.8	4.5	16	5	–	ns
	master reset pulse width LOW	see Fig.10	4.5	20	8	–	ns
	set-up time DS to SH_CP	see Fig.9	4.5	16	5	–	ns
	set-up time SH_CP to ST_CP	see Fig.8	4.5	16	8	–	ns
	hold time DS to SH_CP	see Fig.9	4.5	+3	–2	–	ns
	removal time $\overline{MR}$ to SH_CP	see Fig.10	4.5	+10	–7	–	ns
	maximum clock pulse frequency SH_CP or ST_CP	see Figs 7 and 8	4.5	30	52	–	MHz
= –40 to +85 °C							
PLH	propagation delay SH_CP to Q7'	see Fig.7	4.5	–	–	53	ns
	propagation delay ST_CP to Qn	see Fig.8	4.5	–	–	50	ns
	propagation delay $\overline{MR}$ to Q7'	see Fig.10	4.5	–	–	50	ns
PZL	3-state output enable time $\overline{OE}$ to Qn	see Fig.11	4.5	–	–	44	ns
PLZ	3-state output disable time $\overline{OE}$ to Qn	see Fig.11	4.5	–	–	38	ns

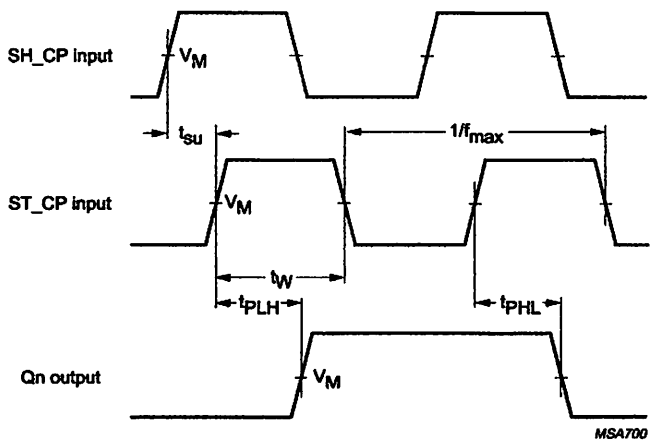
MBOL	PARAMETER	TEST CONDITIONS		MIN.	TYP.	MAX.	UNIT
		WAVEFORMS	V_{CC} (V)				
	shift clock pulse width HIGH or LOW	see Fig.7	4.5	20	–	–	ns
	storage clock pulse width HIGH or LOW	see Fig.8	4.5	20	–	–	ns
	master reset pulse width LOW	see Fig.10	4.5	25	–	–	ns
	set-up time DS to SH_CP	see Fig.9	4.5	20	–	–	ns
	set-up time SH_CP to ST_CP	see Fig.8	4.5	20	–	–	ns
	hold time DS to SH_CP	see Fig.9	4.5	3	–	–	ns
	removal time $\overline{MR}$ to SH_CP	see Fig.10	4.5	13	–	–	ns
	maximum clock pulse frequency SH_CP or ST_CP	see Figs 7 and 8	4.5	24	–	–	MHz
= –40 to +125 °C							
t_{PLH}	propagation delay SH_CP to Q7'	see Fig.7	4.5	–	–	63	ns
	propagation delay ST_CP to Qn	see Fig.8	4.5	–	–	60	ns
	propagation delay $\overline{MR}$ to Q7'	see Fig.10	4.5	–	–	60	ns
t_{PZL}	3-state output enable time $\overline{OE}$ to Qn	see Fig.11	4.5	–	–	53	ns
t_{PLZ}	3-state output disable time $\overline{OE}$ to Qn	see Fig.11	4.5	–	–	45	ns
	shift clock pulse width HIGH or LOW	see Fig.7	4.5	24	–	–	ns
	storage clock pulse width HIGH or LOW	see Fig.8	4.5	24	–	–	ns
	master reset pulse width LOW	see Fig.10	4.5	30	–	–	ns
	set-up time DS to SH_CP	see Fig.9	4.5	24	–	–	ns
	set-up time SH_CP to ST_CP	see Fig.8	4.5	24	–	–	ns
	hold time DS to SH_CP	see Fig.9	4.5	3	–	–	ns
	removal time $\overline{MR}$ to SH_CP	see Fig.10	4.5	15	–	–	ns
	maximum clock pulse frequency SH_CP or ST_CP	see Figs 7 and 8	4.5	20	–	–	MHz

/AVEFORMS



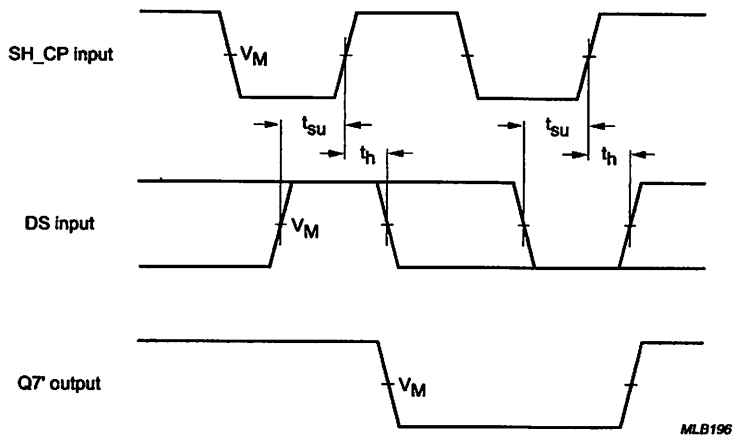
HC595: V_M = 50%; V_I = GND to V_{CC}.
HCT595: V_M = 1.3 V; V_I = GND to 3 V.

g.7 Waveforms showing the clock (SH_CP) to output (Q7') propagation delays, the shift clock pulse width and maximum shift clock frequency.



HC595: V_M = 50%; V_I = GND to V_{CC}.
HCT595: V_M = 1.3 V; V_I = GND to 3 V.

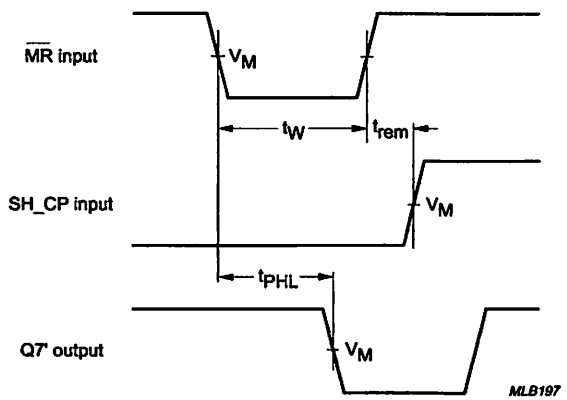
g.8 Waveforms showing the storage clock (ST_CP) to output (Qn) propagation delays, the storage clock pulse width and the shift clock to storage clock set-up time.



HC595: V_M = 50%; V_I = GND to V_{CC}.
HCT595: V_M = 1.3 V; V_I = GND to 3 V.

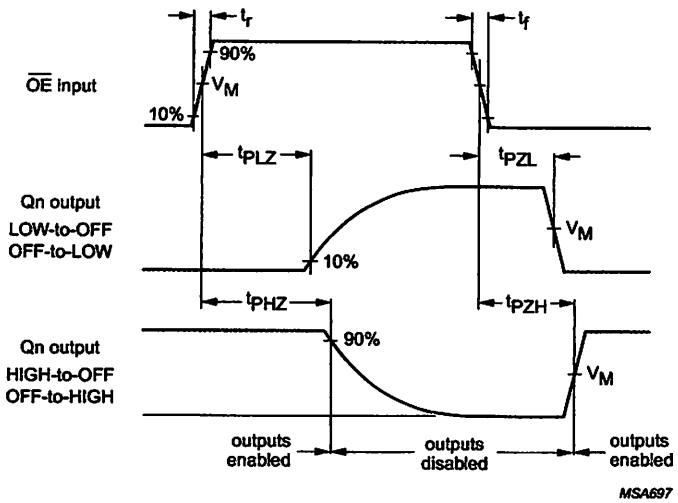
e shaded areas indicate when the input is permitted to change for predictable output performance.

Fig.9 Waveforms showing the data set-up and hold times for the DS input.



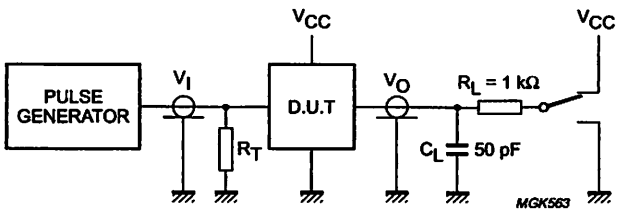
HC595: V_M = 50%; V_I = GND to V_{CC}.
HCT595: V_M = 1.3 V; V_I = GND to 3 V.

g.10 Waveforms showing the Master Reset ($\overline{MR}$) pulse width, the master reset to output (Q7') propagation delay and the master reset to shift clock (SH_CP) removal time.



HC595: V_M = 50%; V_I = GND to V_{CC}.
HCT595: V_M = 1.3 V; V_I = GND to 3 V.

Fig.11 Waveforms showing the 3-state enable and disable times for input $\overline{OE}$.



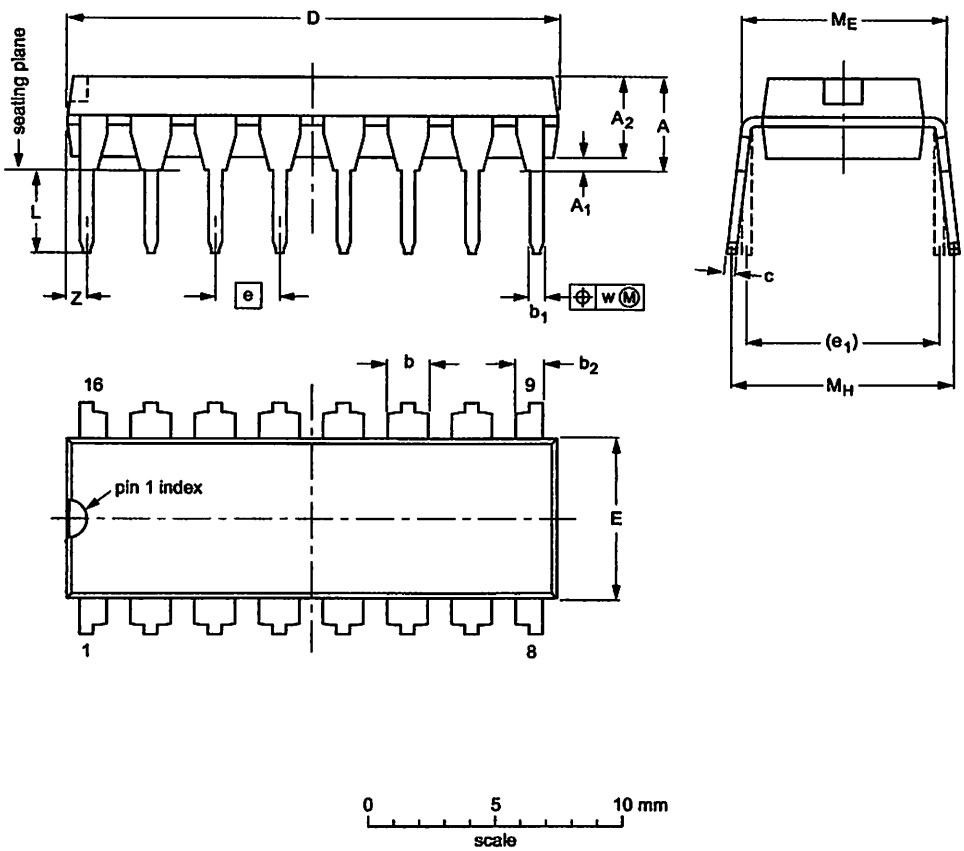
TEST	SWITCH
t_{PLH}/t_{PHL}	open
t_{PLZ}/t_{PZL}	V_{CC}
t_{PHZ}/t_{PZH}	GND

Definitions for test circuit:
R_L = Load resistor.
C_L = Load capacitance including jig and probe capacitance.
R_T = Termination resistance should be equal to the output impedance Z_o of the pulse generator.

Fig.12 Test circuit for 3-state outputs.

KAGE OUTLINES

6: plastic dual in-line package; 16 leads (300 mil) **SOT38-4**



ENSIONS (inch dimensions are derived from the original mm dimensions)

NIT	A max.	A_1 min.	A_2 max.	b	b_1	b_2	c	$D^{(1)}$	$E^{(1)}$	e	e_1	L	M_E	M_H	w	$Z^{(1)}$ max.
nm	4.2	0.51	3.2	1.73 1.30	0.53 0.38	1.25 0.85	0.36 0.23	19.50 18.55	6.48 6.20	2.54	7.62	3.60 3.05	8.25 7.80	10.0 8.3	0.254	0.76
ches	0.17	0.02	0.13	0.068 0.051	0.021 0.015	0.049 0.033	0.014 0.009	0.77 0.73	0.26 0.24	0.1	0.3	0.14 0.12	0.32 0.31	0.39 0.33	0.01	0.03

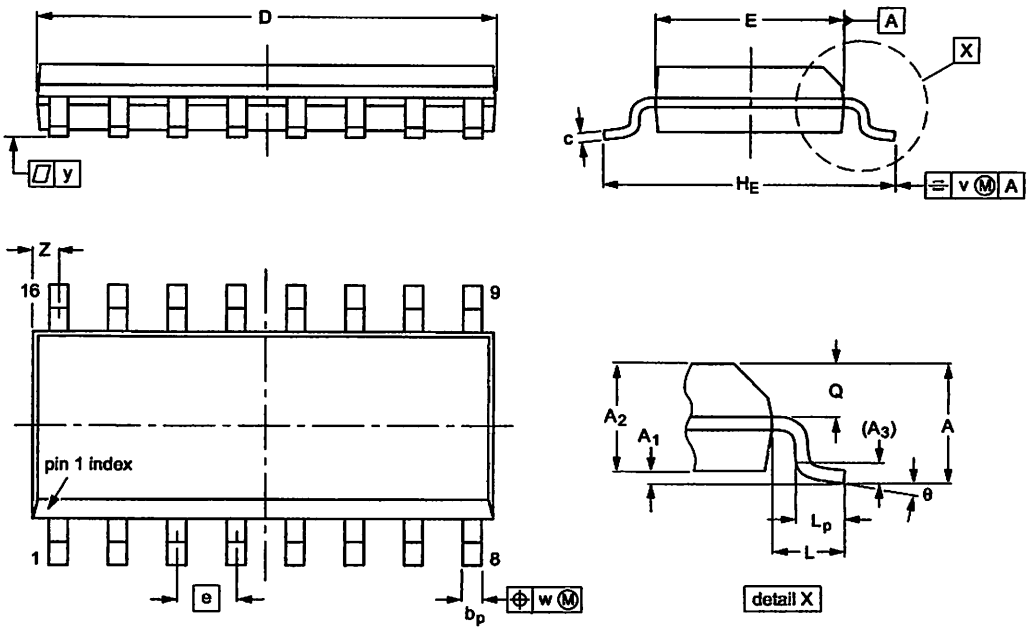
e

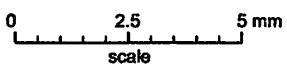
lastic or metal protrusions of 0.25 mm (0.01 inch) maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT38-4						~~95-01-14~~ 03-02-13

): plastic small outline package; 16 leads; body width 3.9 mm

SOT109-1





IENSIONS (inch dimensions are derived from the original mm dimensions)

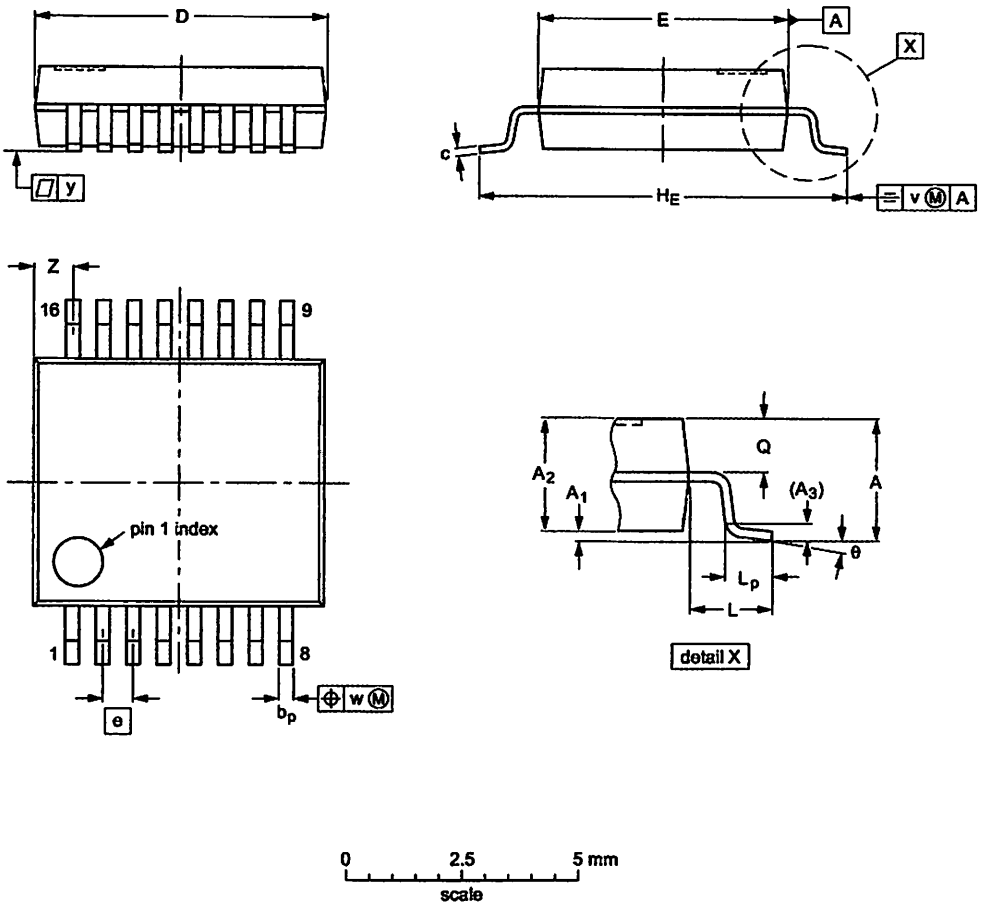
NIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
ım	1.75	0.25 0.10	1.45 1.25	0.25	0.49 0.36	0.25 0.19	10.0 9.8	4.0 3.8	1.27	6.2 5.8	1.05	1.0 0.4	0.7 0.6	0.25	0.25	0.1	0.7 0.3	8° 0°
ches	0.069	0.010 0.004	0.057 0.049	0.01	0.019 0.014	0.0100 0.0075	0.39 0.38	0.16 0.15	0.05	0.244 0.228	0.041	0.039 0.016	0.028 0.020	0.01	0.01	0.004	0.028 0.012	

e

'lastic or metal protrusions of 0.15 mm (0.006 inch) maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT109-1	076E07	MS-012				~~99-12-27~~ 03-02-19

P16: plastic shrink small outline package; 16 leads; body width 5.3 mm **SOT338-1**



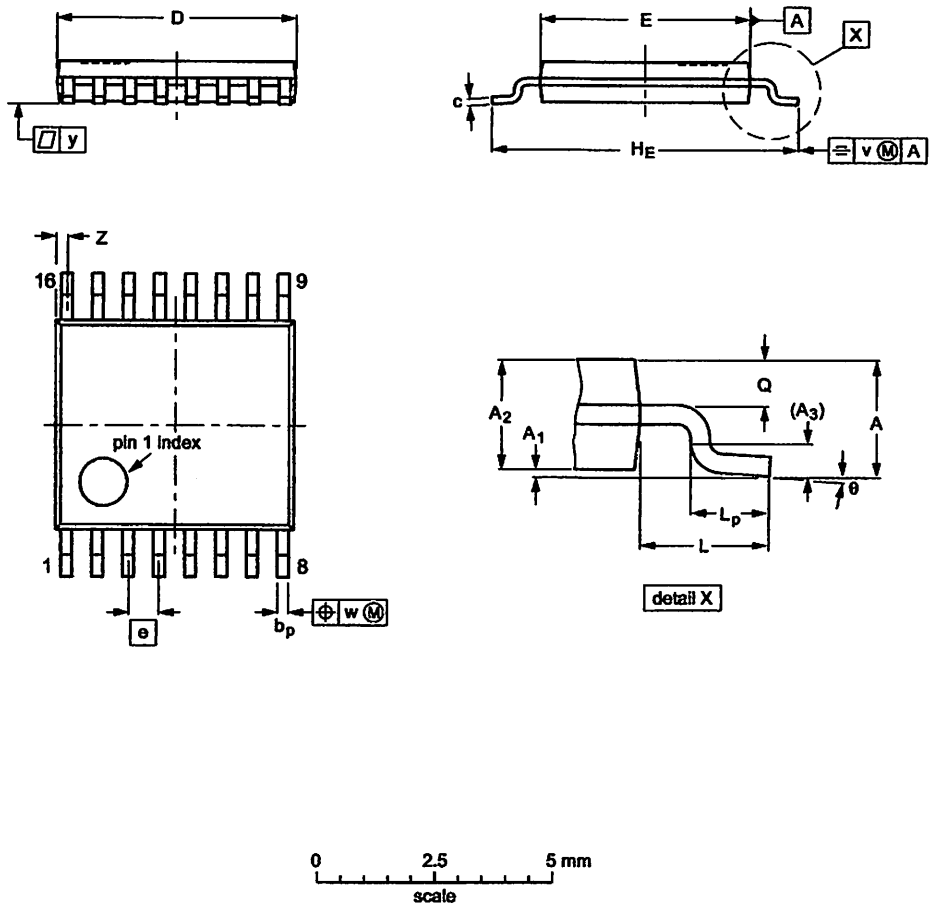
ENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
mm	2	0.21 0.05	1.80 1.65	0.25	0.38 0.25	0.20 0.09	6.4 6.0	5.4 5.2	0.65	7.9 7.6	1.25	1.03 0.63	0.9 0.7	0.2	0.13	0.1	1.00 0.55	8° 0°

e
lastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT338-1		MO-150				~~99-12-27~~ 03-02-19

DP16: plastic thin shrink small outline package; 16 leads; body width 4.4 mm **SOT403-1**



ENSIONS (mm are the original dimensions)

NIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(2)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
im	1.1	0.15 0.05	0.95 0.80	0.25	0.30 0.19	0.2 0.1	5.1 4.9	4.5 4.3	0.65	6.6 6.2	1	0.75 0.50	0.4 0.3	0.2	0.13	0.1	0.40 0.06	8° 0°

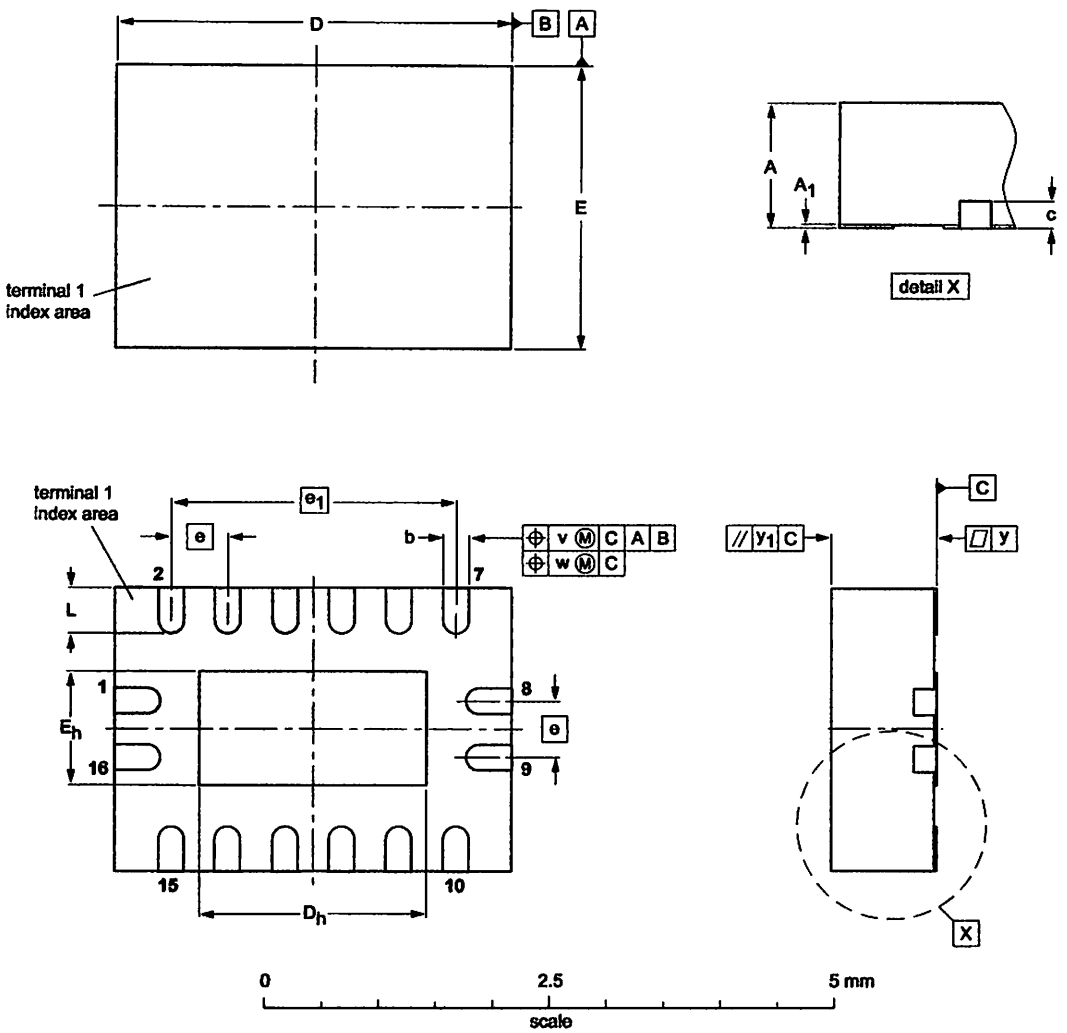
es

lastic or metal protrusions of 0.15 mm maximum per side are not included.

lastic interlead protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT403-1		MO-153				~~99-12-27~~ 03-02-18

QFN16: plastic dual in-line compatible thermal enhanced very thin quad flat package; no leads; :rminals; body 2.5 x 3.5 x 0.85 mm **SOT763-1**



IENSIONS (mm are the original dimensions)

NIT	A(1) max.	A1	b	c	D(1)	Dh	E(1)	Eh	e	e1	L	v	w	y	y1
im	1	0.05 0.00	0.30 0.18	0.2	3.6 3.4	2.15 1.85	2.6 2.4	1.15 0.85	0.5	2.5	0.5 0.3	0.1	0.05	0.05	0.1

:e
Plastic or metal protrusions of 0.075 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT763-1	---	MO-241	---			~~02-10-17~~ 03-01-27

\ SHEET STATUS

EL	DATA SHEET STATUS[1]	PRODUCT STATUS[2][3]	DEFINITION
	Objective data	Development	This data sheet contains data from the objective specification for product development. Philips Semiconductors reserves the right to change the specification in any manner without notice.
	Preliminary data	Qualification	This data sheet contains data from the preliminary specification. Supplementary data will be published at a later date. Philips Semiconductors reserves the right to change the specification without notice, in order to improve the design and supply the best possible product.
	Product data	Production	This data sheet contains data from the product specification. Philips Semiconductors reserves the right to make changes at any time in order to improve the design, manufacturing and supply. Relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN).

s

lease consult the most recently issued data sheet before initiating or completing a design.

he product status of the device(s) described in this data sheet may have changed since this data sheet was ublished. The latest information is available on the Internet at URL http://www.semiconductors.philips.com.

or data sheets describing multiple type numbers, the highest-level product status determines the data sheet status.

NITIONS

t-form specification — The data in a short-form fication is extracted from a full data sheet with the type number and title. For detailed information see levant data sheet or data handbook.

ing values definition — Limiting values given are in dance with the Absolute Maximum Rating System 60134). Stress above one or more of the limiting s may cause permanent damage to the device. e are stress ratings only and operation of the device se or at any other conditions above those given in the acteristics sections of the specification is not implied. sure to limiting values for extended periods may device reliability.

ication information — Applications that are ibed herein for any of these products are for ative purposes only. Philips Semiconductors make presentation or warranty that such applications will be le for the specified use without further testing or ication.

DISCLAIMERS

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

Right to make changes — Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

hilips Semiconductors – a worldwide company

ntact information

additional information please visit **http://www.semiconductors.philips.com**. Fax: **+31 40 27 24825**
sales offices addresses send e-mail to: **sales.addresses@www.semiconductors.philips.com**.

oninklijke Philips Electronics N.V. 2003 SCA75

ights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner.
information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed
out notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license
er patent- or other industrial or intellectual property rights.

ted in The Netherlands 613508/04/pp28 Date of release: 2003 Jun 25 Document order number: 9397 750 11263

Let's make things better.

ilips
miconductors

PHILIPS

AN.No.1632-711E

LIQUID CRYSTAL DISPLAY MODULE

M 1 6 3 2

USER MANUAL

Seiko Instruments Inc.

PREFACE

This manual describes technical informations on functions and instructions of M1632 from Seiko Instruments Inc. Please read this instruction manual carefully to understand all the module functions and make the best use of them. Description details may be changed without notice.

Revision Record

Edition	Revision	Date
1	Original	April 1985
2	Completely revised	Jan. 1987

©Seiko Instruments Inc. 1987
Printed in Japan

NERAL

General

The M1632 is a low-power-consumption dot-matrix liquid crystal display (LCD) module with a high-contrast wide-view TN LCD panel and a CMOS LCD drive controller built in. The controller has a built-in character generator ROM/RAM, and display data RAM. All the display functions are controlled by instructions and the module can easily be interfaced with an MPU. This makes the module applicable to a wide range of purposes including terminal display units for microcomputers and display units for measuring gages.

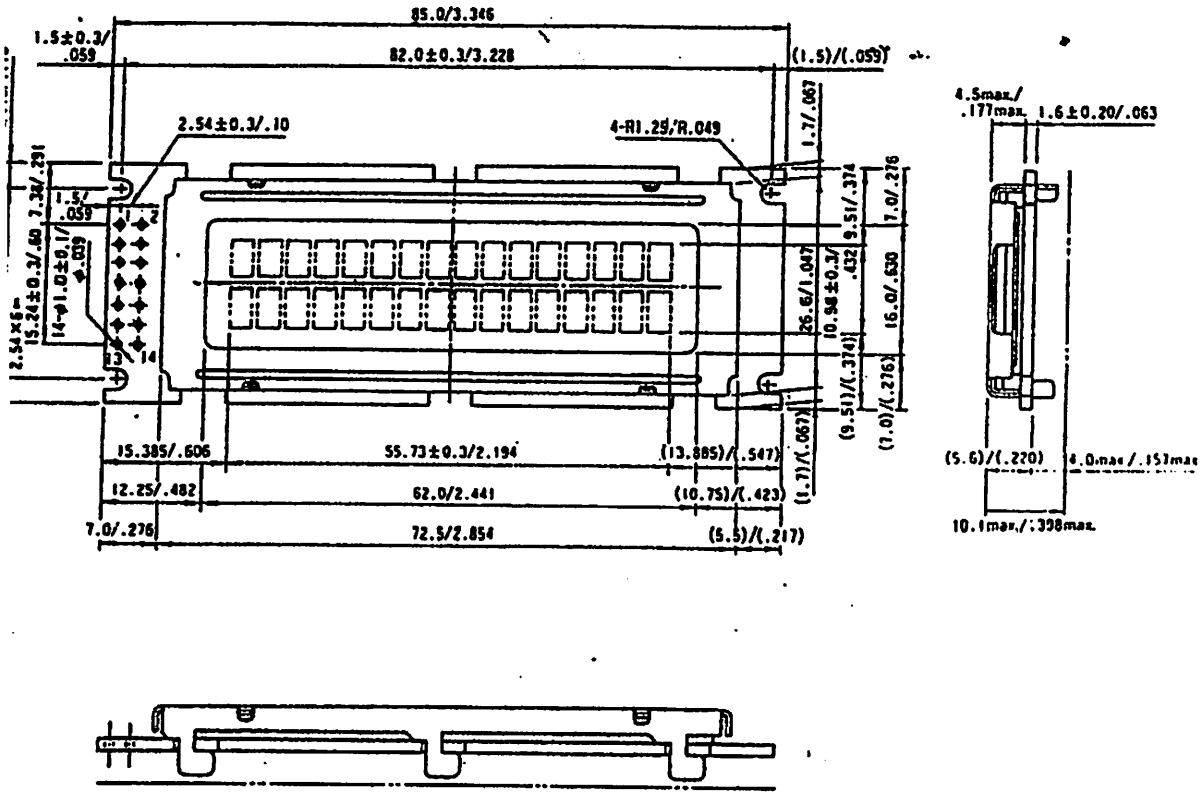
Features

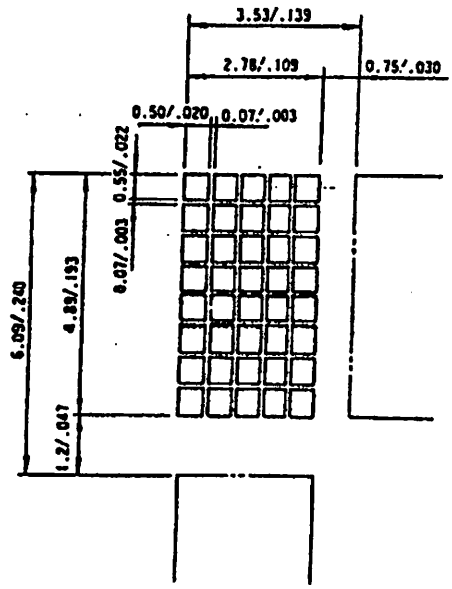
- 16-character, two-line TN liquid crystal display of 5 x 7 dot matrix + cursor
- Duty ratio: 1/16
- Character generator ROM for 192 character types.
  (character font: 5 x 7 dot matrix)
- Character generator RAM for eight character types (program write)
  (character font: 5 x 7 dot matrix)
- 80 x 8 bit display data RAM (80 characters maximum)
- Interface with four-bit and eight-bit MPUs possible
- Display data RAM and character generator RAM readable from MPU
- Many instruction functions

  Display Clear, Cursor Home, Display ON/OFF, Cursor ON/OFF, Display Character Blink, Cursor Shift, and Display Shift

- Built-in oscillator circuit
- +5 V single power supply
- Built-in automatic reset circuit at power-on
- CMOS process
- Operating temperature range: 0°C to 50°C

3 Dimensions Diagram



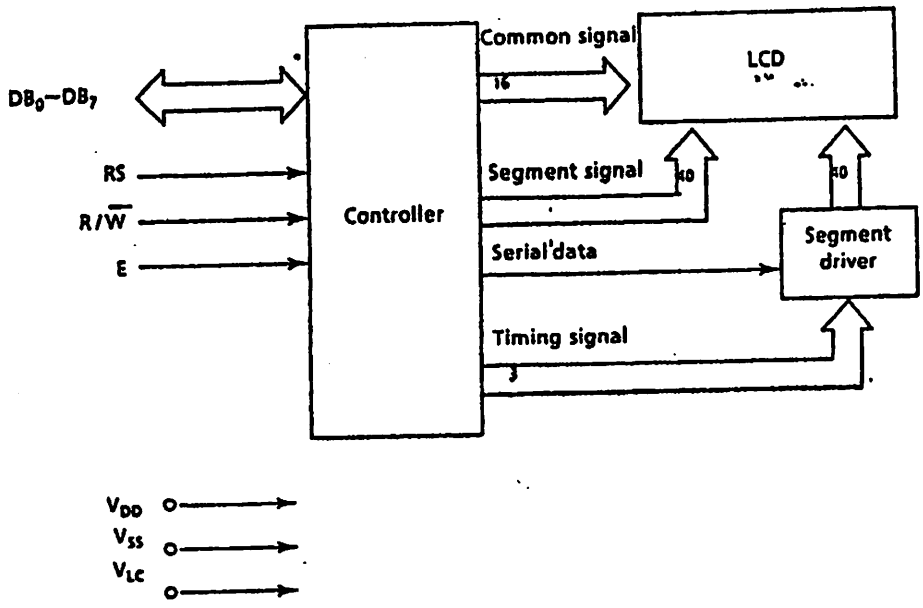


Unit : mm/inch
General tolerance : ± 0.5 mm

No.	Symbol	Level	Function	
1	Vss	-	Power Supply	0V (GND)
2	Vcc	-		5V ±10%
3	Vee	-		for LCD Drive
4	RS	H/L	H: Data Input L: Instruction Input	
5	R/W	H/L	H:READ L:WRITE	
6	E	H, ↴	Enable Signal	
7	DB0	H/L	Data Bus	
8	DB1	H/L		
9	DB2	H/L		
10	DB3	H/L		
11	DB4	H/L		
12	DB5	H/L		
13	DB6	H/L		
14	DB7	H/L		
15	V+ BL	-	Back Light Supply	4 - 4.2V 50-200mA
16	V- BL	-		0V (GND)

Figure 1 Dimensions diagram

ock Diagram


DB0–DB7
RS
R/W
E
Controller
Common signal
16
LCD
Segment signal
40
Serial data
Timing signal
3
40
Segment driver
VDD
VSS
VLC

bsolute Maximum Ratings

V_{SS} = 0 V

Item	Symbol	Standard	Unit	Remarks
Power supply voltage	V_{DD}	− 0.3 to + 7.0	V	
	V_{LC}	V_{DD} − 13.5 to V_{DD} + 0.3	V	
Input voltage	V_{in}	− 0.3 to V_{DD} + 0.3,	V	
Operating temperature	T_{opr}	0 to + 50	°C	
Storage temperature	T_{stg}	− 20 to + 60	°C	At 50% RH

ectrical Characteristics

V_{DD} = 5 V ± 5%, V_{SS} = 0 V, T_A = 0°C to 50°C

Item		Symbol	Conditions	Standard			Unit
				Min.	Typ.	Max.	
Input voltage	High	V_{IH1}		2.2	−	V_{DD}	V
	Low	V_{IL1}		0	−	0.6	V
Output voltage (TTL)	High	V_{OH1}	$-I_{OH}$ = 0.205 mA	2.4	−	−	V
	Low	V_{OL1}	I_{OL} = 1.2 mA	−	−	0.4	V
Output voltage (CMOS)	High	V_{OH2}	$-I_{OH}$ = 0.04 mA	0.9V_{DD}	−	−	V
	Low	V_{OL2}	I_{OL} = 0.04 mA	−	−	0.1V_{DD}	V
Power supply voltage		V_{DD}		4.75	5.00	5.25	V
		$-V_{LC}$	V_{DD} = 5 V, T_A = 25°C	−	0.25	−	V
Current consumption		I_{DD}		−	2.0	3.0	mA
		I_{LC}	V_{LC} = 0.25V	−	−	1.0	mA
Clock oscillation freq.		f_{osc}	Resistance oscillation	190	270	350	kHz

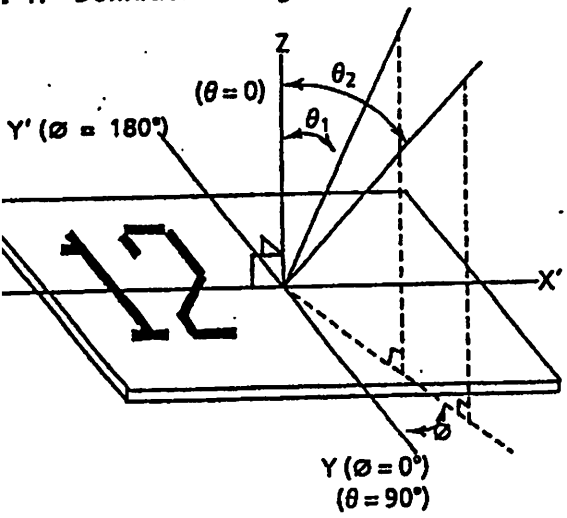
Optical Characteristics

7.1 Optical characteristics

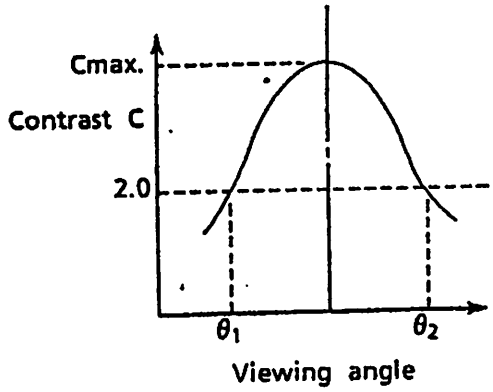
Maximum viewing angle: 6 o'clock ($\varnothing$ = 0°)
T_A = 25°C, V_{opr} = 4.75 V

Item	Symbol	Conditions	Min.	Typ.	Max.	Remarks
Viewing angle	$\theta_2 - \theta_1$	$C \geq 2.0$, $\varnothing = 0°$	35	–	–	See Notes 1 and 2.
Contrast	C	$\theta = 25°$, $\varnothing = 0°$	5	8	–	See Note 3.
Rise time	t_{on}	$\theta = 25°$, $\varnothing = 0°$	–	60 ms	70 ms	See Note 4.
Fall time	t_{off}	$\theta = 25°$, $\varnothing = 0°$	–	150 ms	170 ms	See Note 4.

e 1: Definition of angles $\varnothing$ and θ

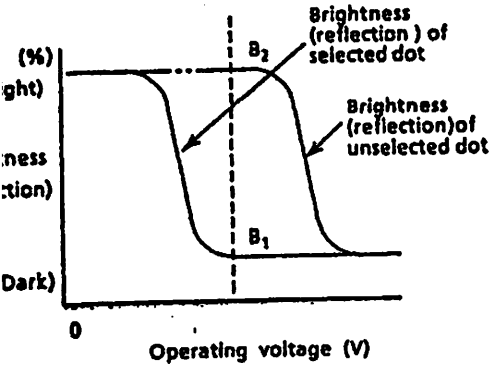


Note 2: Definition of viewing angles θ_1 and θ_2

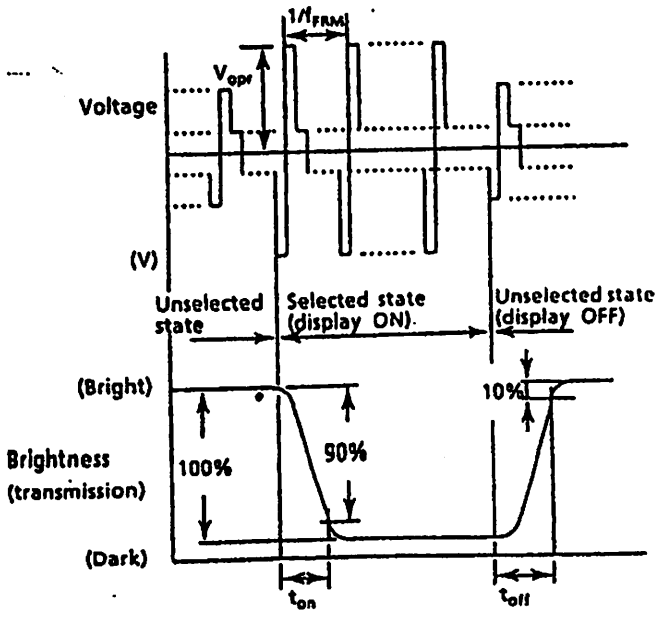


Note 3: Definition of contrast C

$$C = \frac{\text{Brightness (reflection) of unselected dot (B2)}}{\text{Brightness (reflection) of selected dot (B1)}}$$



Note 4: Definition of response time



V_{opr} : Operating voltage (V)
f_{FRM} : Frame frequency (Hz)
t_{on} : Response time (rise)(ms)
t_{off} : Response time (fall)(ms)

Recommended operating voltage

The viewing angle and screen contrast of the LCD panel can be varied by changing the liquid crystal operating voltage (V_{opr}), that is V_{LC}.

The optical characteristics is influenced by an ambient temperature. The recommended value of V_{opr} for an ambient temperatures are shown below.

Temperature (℃)	0	10	25	40	50
Voltage V_{opr} (V)	5.00	4.90	4.75	4.60	4.50

$V_{opr} = V_{DD} - V_{LC}$

RATING INSTRUCTIONS

Terminal Functions

Table 1 Terminal functions

name	No. of terminals	I/O	Destination	Function
o DB_3	4	I/O	MPU	Tristate bidirectional lower four data buses: Data is read from the module to the MPU or written to the module from the MPU through the buses. If the interface data is 4 bits, the signals are not used.
o DB_7	4	I/O	MPU	Tristate bidirectional upper four data buses: Data is read from the module to the MPU or written to the module from the MPU through the buses. DB_7 is also used as a busy flag.
E	1	Input	MPU	Operation start signal: The signal activates data write or read.
$\overline{W}$	1	Input	MPU	Read (R) and Write ($\overline{W}$) selection signals 0 : Write 1 : Read
S	1	Input	MPU	Register selection signals 0 : Instruction register (Write) Busy flag and address counter (Read) 1 : Data register (Write and Read)
LC	1	–	Power supply	Power supply terminal for driving liquid crystal display: The screen contrast can be varied by changing V_{LC}.
DD	1	–	Power supply	+5 V
SS	1	–	Power supply	Ground terminal: 0 V

Basic Operations

.1 Registers

The controller has two kinds of eight-bit registers: the instruction register (IR) and the data register (DR). They are selected by the register select (RS) signal as shown in Table 2.

The IR stores instruction codes such as Display Clear and Cursor Shift, and the address information of display data RAM (DD RAM) and character generator RAM (CG RAM). They can be written from the MPU, but cannot be read to the MPU.

The DR temporarily stores data to be written into DD RAM or CG RAM, or read from DD RAM or CG RAM. When data is written into DD RAM or CG RAM from the MPU, the data in the DR is automatically written into DD RAM or CG RAM by internal operation. However, when data is read from DD RAM or CG RAM, the necessary data address is written into the IR. The specified data is read out to the DR and then the MPU reads it from the DR. After the read operation, the next address is set and DD RAM or CG RAM data at the address is read into the DR for the next read operation.

Table 2 Register selection

S	R/$\overline{W}$	Operation
)	0	IR selection, IR write. Internal operation : Display clear
)	1	Busy flag (DB_7) and address counter (DB_0 to DB_6) read
	0	DR selection, DR write. Internal operation : DR to DD RAM or CG RAM
	1	DR selection, DR read. Internal operation : DD RAM or CG RAM to DR

2.2 Busy flag (BF)

The flag indicates whether the module is ready to accept the next instruction. As shown in Table 2, the signal is output to DB_7 if RS = 0 and R/$\overline{W}$ = 1. If the value is 1, the module is working internally and the instruction cannot be accepted. If the value is 0, the next instruction can be written. Therefore, the flag status needs to be checked before executing an instruction. If an instruction is executed without checking the flag status, wait for more than the execution time shown by 2.4 Instruction Outline.

.3 Address counter (AC)

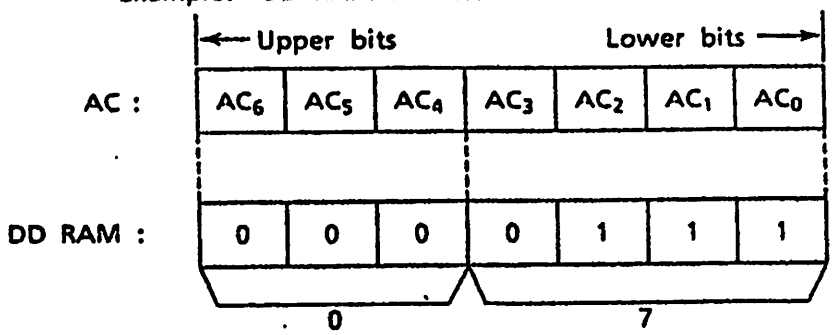
The counter specifies an address when data is written into DD RAM or CG RAM and the data stored in DD RAM or CG RAM is read out. If an Address Set instruction (for DD RAM or CG RAM) is written in the IR, the address information is transferred from the IR to the AC. When display data is written into or read from DD RAM or CG RAM, the AC is automatically incremented or decremented by one according to the Entry Mode Set. The contents of the AC are output to DB_0 to DB_6 as shown in Table 2 if RS = 0 and $R/\overline{W}$ = 1.

.4 Display data RAM (DD RAM)

DD RAM has a capacity of up to 80 × 8 bits and stores display data of 80 eight-bit character codes. Some storage areas of DD RAM which are not used for display can be used as general data RAM.

A DD RAM address to be set in the AC is expressed in hexadecimal form as follows.

Example: DD RAM address = 07



00_H to $0F_H$ of the DD RAM address is set in the line 1, and 40_H to $4F_H$ in the line 2.

Note : The addresses in the digit 16 of line 1 and the digit 1 of line 2 are not consecutive.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Display digit
e 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	DD RAM address
e 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	

If the display is shifted, DD RAM address 00_H to 27_H are displayed in line 1 and 40_H to 67_H in line 2. The following figures are examples of display shifts.

*Left shift

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Display digit
1	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	DD RAM address
2	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	

*Right shift

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Display digit
1	27	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	DD RAM address
2	67	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	

Character generator ROM (CG ROM)

Character generator ROM generates 192 types of 5 x 7 dot-matrix character patterns from eight-bit character codes.

Table 3 shows the correspondence between the CG ROM character codes and character patterns.

Character generator RAM (CG RAM)

CG RAM is used to create character patterns freely by programming. Eight types of character patterns can be written.

Table 4 shows the character patterns created from CG RAM addresses and data. To display a created character pattern, the character code in the left column of the table is written into DD RAM corresponding to the display position (digit). The areas not used for display are available as general data RAM.

Table 3 Correspondence between character codes and character patterns

bit / 4 bit	0 0000	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	1010	1011	1100	1101	1110	1111
000	CG RAM (1)												
001	(2)												
010	(3)												
011	(4)												
100	(5)												
101	(6)												
110	(7)												
111	(8)												
000	(1)												
001	(2)												
010	(3)												
011	(4)												
100	(5)												
101	(6)												
110	(7)												
111	(8)												

Table 4 Relationships between CG RAM addresses and character codes (DD RAM) and character patterns (CG RAM data)

haracter code)D RAM data)		CG RAM address		Character pattern (CG RAM data)	
5 5 4 3 2 1 0 r bit Lower bit →		5 4 3 2 1 0 ←Upper bit Lower bit →		7 6 5 4 3 2 1 0 ←Upper bit Lower bit →	
) 0 0 * 0 0 0		0 0 0 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 0 1 1 1		* * * 1 1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0 1 0 1 0 0 1 0 0 1 0 1 0 0 0 1 * * * 0 0 0 0 0	Example of character pattern (R) ← Cursor position
) 0 0 * 0 0 1		0 0 1 0 0 0 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 0 1 1 1		* * * 1 0 0 0 1 0 1 0 1 0 1 1 1 1 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 0 0 1 0 0 * * * 0 0 0 0 0	Example of character pattern (¥)
0 0 * 1 1 1		1 1 1 0 0 0 0 0 1 1 0 0 1 0 1 1 1 0 1 1 1		* * * * * *	

lotes: • In CG RAM data, 1 corresponds to Selection and 0 to Non-selection on the display.

- Character code bits 0 to 2 and CG RAM address bits 3 to 5 correspond with each other (three bits, eight types).
- CG RAM address bits 0 to 2 specify a line position for a character pattern. Line 8 of a character pattern is the cursor position where the logical sum of the cursor and CG RAM data is displayed. Set the data of line 8 to 0 to display the cursor. If the data is changed to 1, one bit lights, regardless of the cursor.

The character pattern column positions correspond to CG RAM data bits 0 to 4 and bit 4 comes to the left end. CG RAM data bits 5 to 7 are not displayed but can be used as general data RAM.

When reading a character pattern from CG RAM, set to 0 all of character code bits 4 to 7. Bits 0 to 2 determine which pattern will be read out. Since bit 3 is not valid, 00_H and 08_H select the same character.

iming Characteristics

1 Write timing characteristics

V_{DD} = 5.0 V ± 5%, V_{SS} = 0 V, T_A = 0°C to 50°C

Item		Symbol	Standard		Unit
			Min.	Max.	
Enable cycle time		t_{CYCE}	1000	–	ns
Enable pulse width	High level	PW_{EH}	450	–	ns
Enable rise and fall time		t_{Er},t_{Ef}	–	25	ns
Setup time	RS, R/$\overline{W}$ — E	t_{AS}	140	–	ns
Address hold time		t_{AH}	10	–	ns
Data setup time		t_{DSW}	195	–	ns
Data hold time		t_H	10	–	ns

ite operation

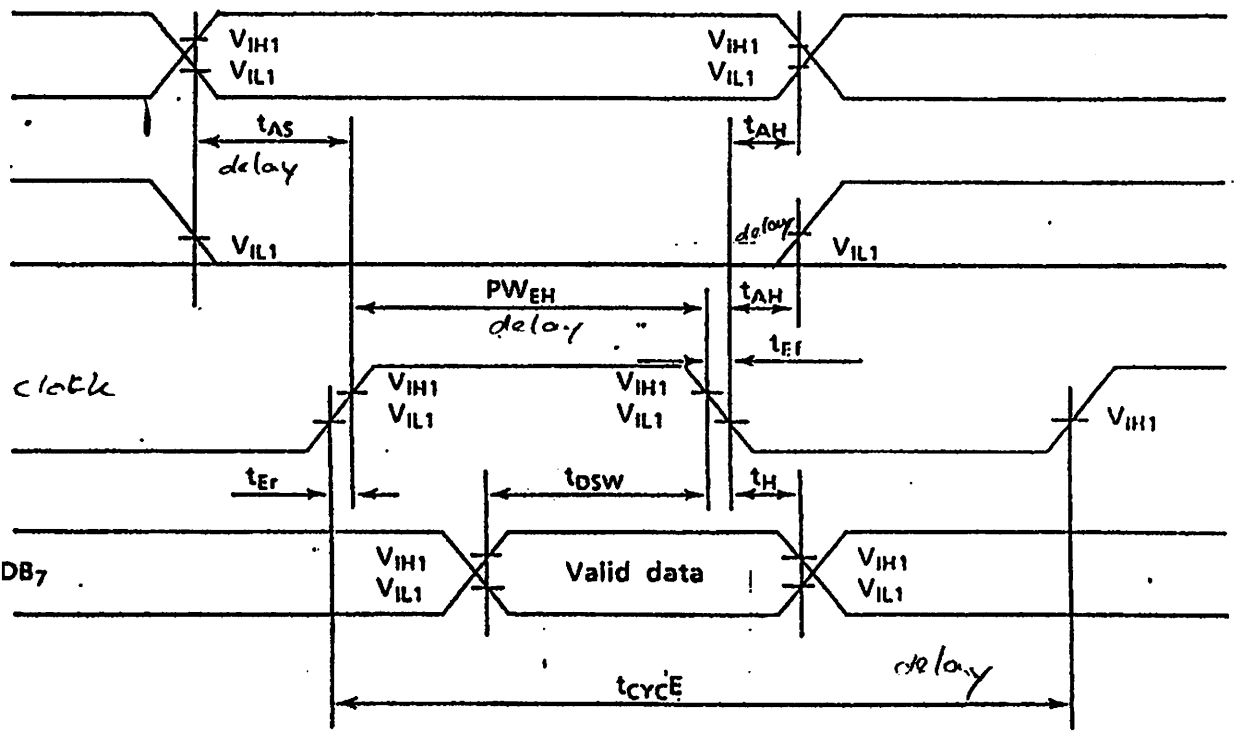


Figure 3 Data write from MPU to module

Read timing characteristics

V_{DD} = 5.0 V ± 5%, V_{SS} = 0 V, T_A = 0°C to 50°C

Item		Symbol	Standard		Unit
			Min.	Max.	
Enable cycle time		t_{CYCE}	1000	–	ns
Enable pulse width	High level	PW_{EH}	450	–	ns
Enable rise and fall time		t_{Er},t_{Ef}	–	25	ns
Setup time	RS, R/$\overline{W}$—E	t_{AS}	140	–	ns
Address hold time		t_{AH}	10	–	ns
Data delay time		t_{DDR}	–	320	ns
Data hold time		t_H	20	–	ns

d operation

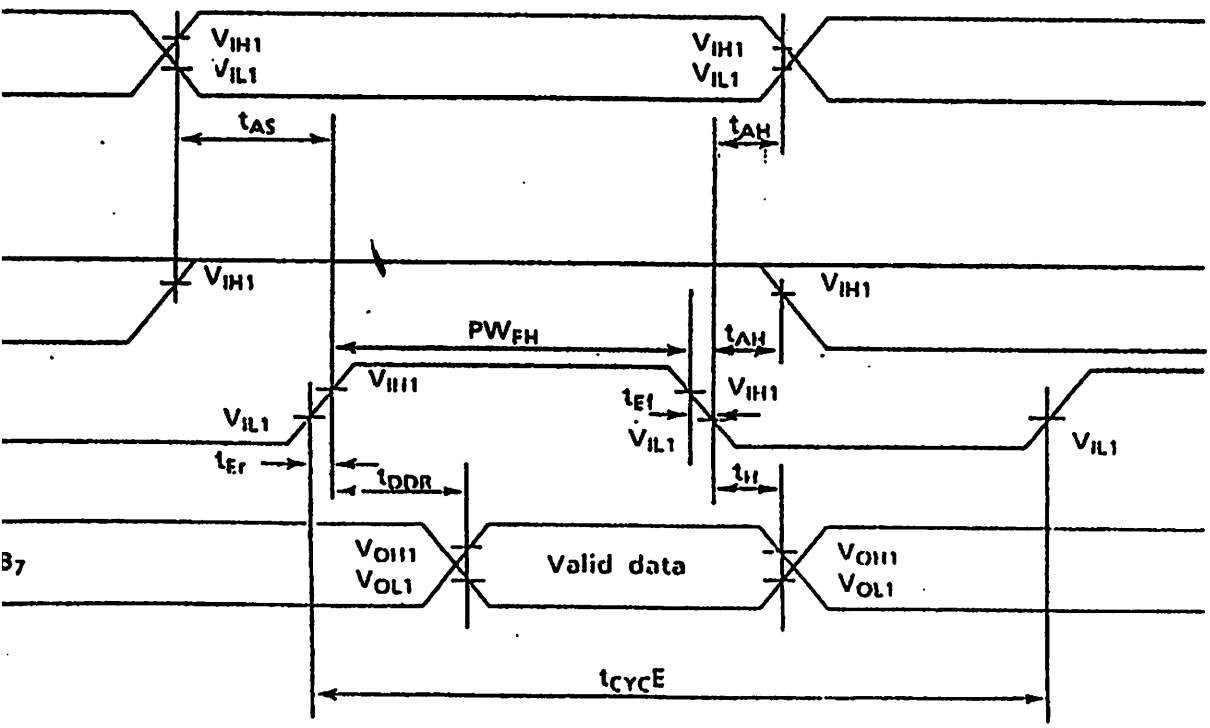


Figure 4 Data read from module to MPU

ıstruction Outline

Table 5 List of instructions

uction	Code										Function	Execu-tion time
	RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0		
clear	0	0	0	0	0	0	0	0	0	1	Clears all display and returns cursor to home position (address 0)	1.64 ms
Home	0	0	0	0	0	0	0	0	1	•	Returns cursor to home position. Shifted display returns to home position and DD RAM contents do not change.	1.64 ms
Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets direction of cursor movement and whether display will be shifted when data is written or read	40 µs
ON / OFF	0	0	0	0	0	0	1	D	C	B	Turns ON/OFF total display (D) and cursor (C), and makes cursor position column start blinking (B)	40 µs
Display Shift	0	0	0	0	0	1	S/C	R/L	•	•	Moves cursor and shifts display without changing DD RAM contents	40 µs
n Set	0	0	0	0	1	DL	1	•	•	•	Sets interface data length (DL)	40 µs
AM Address	0	0	0	1	A_{CG}						Sets CG RAM address to start transmitting or receiving CG RAM data	40 µs
AM Address	0	0	1	A_{DD}							Sets DD RAM address to start transmitting or receiving DD RAM data	40 µs
ress Read	0	1	BF	AC							Reads BF indicating module in internal operation and AC contents (used for both CG RAM and DD RAM)	0 µs
Write to CG or DD RAM	1	0	Write Data								Writes data into DD RAM or CG RAM	40 µs
Read from AM or DD	1	1	Read Data								Reads data from DD RAM or CG RAM	40 µs

l bit	I/D = 1 : Increment	C = 1 : Cursor ON	R/L = 1 : Right shift
RAM address	I/D = 0 : Decrement	C = 0 : Cursor OFF	R/L = 0 : Left shift
RAM address			
	S = 1 : Display shift	B = 1 : Blink ON	DL = 1 : 8 bits
	S = 0 : No display shift	B = 0 : Blink OFF	DL = 0 : 4 bits
	D = 1 : Display ON	S/C = 1 : Display shift	BF = 1 : Internal operation in progress
	D = 0 : Display OFF	S/C = 0 : Cursor movement	BF = 0 : Instruction can be accepted

uction Details

splay Clear

	RS	R/$\overline{W}$	DB_7							DB_0
Code	0	0	0	0	0	0	0	0	0	1

Display Clear clears all display and returns cursor to home position (address 0). Space code 20 (hexadecimal) is written into all the addresses of DD RAM, and DD AM address 0 is set to the AC. If shifted, the display returns to the original sition. After execution of the Display Clear instruction, the entry mode is cremented.

Note : When executing the Display Clear instruction, follow the restrictions listed in Table 6.

ursor Home

	RS	R/$\overline{W}$	DB_7							DB_0	
Code	0	0	0	0	0	0	0	0	1.	*	* : Invalid bit

Cursor Home returns cursor to home position (address 0).

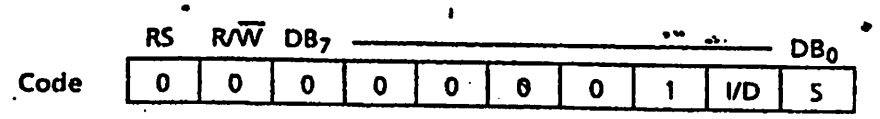
DD RAM address 0 is set to the AC. The cursor returns to the home position. If ifted, the display returns to the original position. The DD RAM contents do not hange. If the cursor or blinking is ON, it returns to the left side.

Note : When executing the Cursor Home instruction, follow the restrictions listed in Table 6.

ble 6 Restrictions on execution of Display Clear and Cursor Home instructions

Conditions of use	Restrictions
en executing the Display Clear or Cursor ne instruction when the display is ed (after execution of Display Shift uction)	The Cursor Home instruction should be executed again immediately after the Display Clear or Cursor Home instruction is executed. Do not leave an interval of a multiple of $400/f_{osc}$* second after the first execution. Example: 1.5 ms, 3 ms, 4.5 ms for f_{osc} = 270 kHz *f_{osc}: Oscillation frequency
en 23_H, 27_H, 63_H, or 67_H is used as a DD M address to execute Cursor Home ruction	Before executing the Cursor Home instruction, the data of the four DD RAM addresses given at the left should be read and saved. After execution, write the data again in DD RAM.(This restriction is necessary to prevent the contents of the DD RAM addresses from being destroyed after the Cursor Home instruction has been executed.)

Entry Mode Set



Entry Mode Set sets the direction of cursor movement and whether display will be shifted.

I/D : The DD RAM address is incremented or decremented by one when a character code is written into or read from DD RAM. This is also true for writing into or reading from CG RAM.

When I/D = 1, the address is incremented by one and the cursor or blink moves to the right.

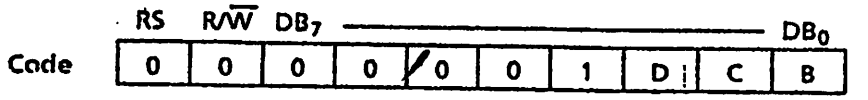
When I/D = 0, the address is decremented by one and the cursor or blink moves to the left.

S : If S = 1, the entire display is shifted either to the right or left for writing into DD RAM. The cursor position does not change, only the display moves. There is no display shift for reading from DD RAM.

When S = 1 and I/D = 1, the display shifts to the left.
When S = 1 and I/D = 0, the display shifts to the right.

If S = 0, the display does not shift.

Display ON/OFF Control



Display ON/OFF Control turns the total display and the cursor ON and OFF, and makes the cursor position start blinking. Cursor ON/OFF and blinking is done at the column indicated by the specified DD RAM address by the AC.

D : When D = 1, the display is turned ON.
When D = 0, the display is turned OFF.
If D = 0 is used, display data remains in DD RAM. Change 0 to 1 to display data.

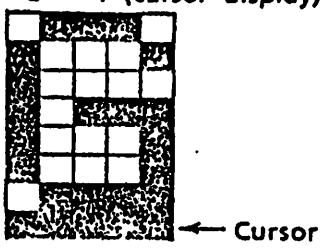
C : When C = 1, the cursor is displayed.
When C = 0, the cursor is not displayed.

The cursor is displayed in the dot line below the 5 x 7 dot-matrix character fonts. If the cursor is OFF, display data is written into DD RAM in the order specified by I/D.

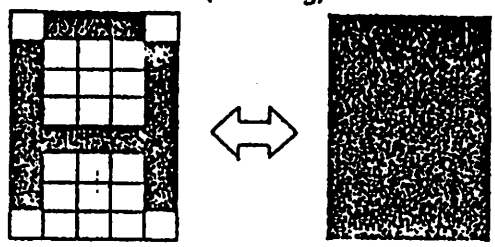
B : When B = 1, the character at the cursor position starts blinking.
When B = 0, it does not blink.

For blinking, all-black dots and the character are switched about every 0.4 seconds. The cursor and blinking can be set at the same time.

Example: C = 1 (cursor display)



B = 1 (blinking)



ursor/Display Shift

	RS	R/W	DB_7							DB_0
Code	0	0	0	0	0	1	S/C	R/L	*	*

* :Invalid bit

Cursor/Display Shift moves the cursor and shifts the display without changing the D RAM contents.
The cursor position and the AC contents match. This instruction is available for splay correction and retrieval because the cursor position or display can be shifted ithout writing or reading display data. Since the DD RAM capacity is 40-character d two lines, the cursor is shifted from digit 40 of line 1 to digit 1 of line 2. splays of lines 1 and 2 are shifted at the same time. Therefore, the display ttern of line 2 is not shifted to line 1.

S/C	R/L	Operation
0	0	The cursor position is shifted to the left (the AC decrements one).
0	1	The cursor position is shifted to the right (the AC increments one).
1	0	The entire display is shifted to the left with the cursor.
1	1	The entire display is shifted to the right with the cursor.

Note: If only display shift is done, the AC contents do not change.

'unction Set

	RS	R/W	DB_7						DB_0
Code	0	0	0	0	1	DL	1	*	*	*

* : Invalid bit

unction Set sets the interface data length.

DL : Interface data length

When DL = 1, the data length is set at eight bits (DB_7 to DB_0).

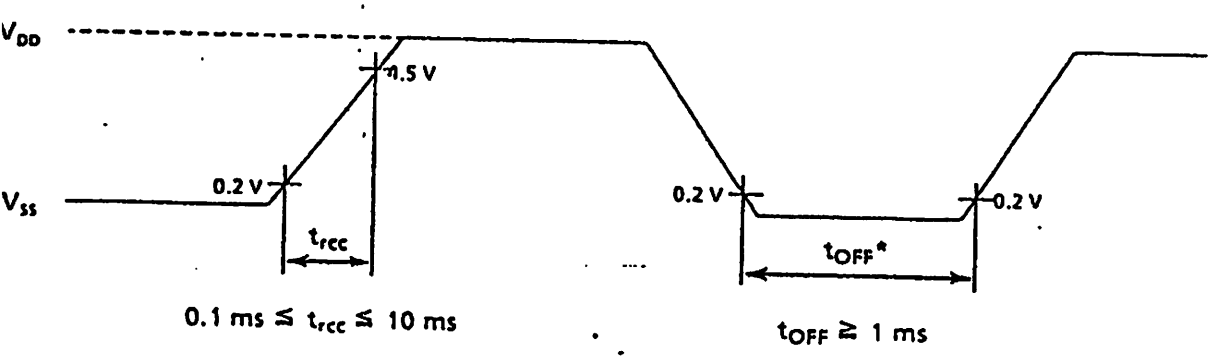
When DL = 0, the data length is set at four bits (DB_7 to DB_4).

The upper four bits are transferred first, then the lower four bits follow.

The Function Set instruction must be executed prior to all other instructions
:cept for Busy Flag/Address Read. If another instruction is executed first, no
nction instruction except changing the interface data length can be executed.

:marks: Initialization

The system is automatically initialized at power-on if the following power supply
nditions are satisfied.



*t_{OFF}: Time when power supply is OFF if cut instantaneously or turned ON and OFF repeatedly

following instructions are executed for initialization.

5 x 7 dot-matrix character font: 1/8 duty

Display clear

Function Set — DL = 1: Interface data length: 8 bits

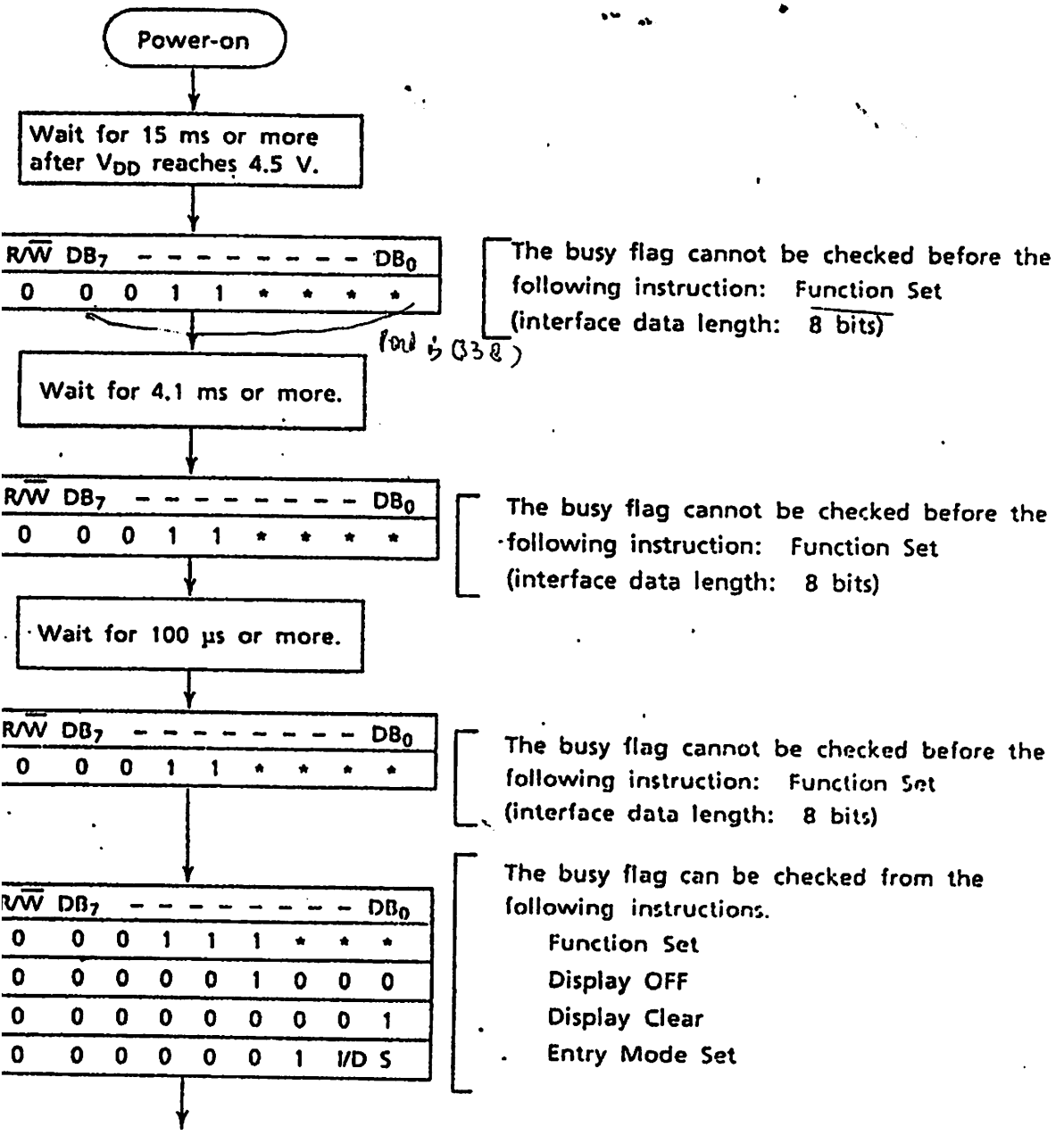
Display ON/OFF Control — D = 0: Display OFF
C = 0: Cursor OFF
B = 0: Blink OFF

Entry mode — I/O = 1: Increment
S = 0: No display shift

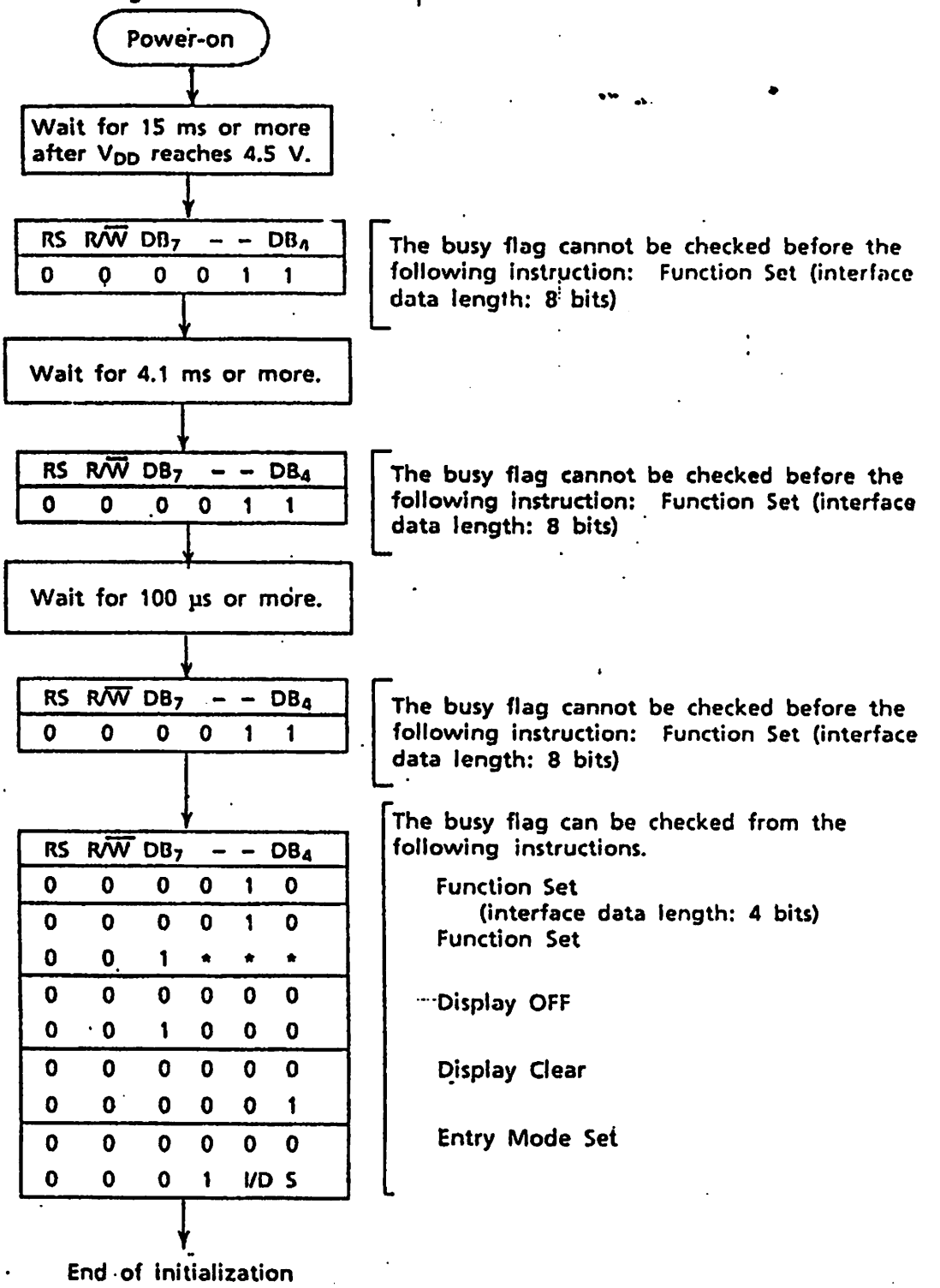
ince the condition is not suitable for the M1632, further function setting is essary.

automatic initialization is not executed because the above power supply ditions are not satisfied, use the instruction from next page on.

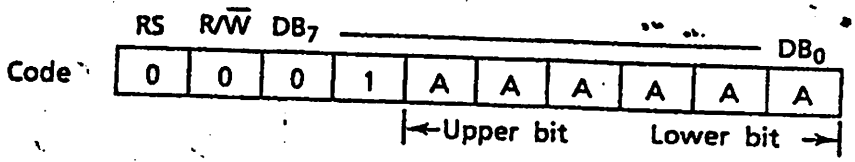
Interface data length : Eight bits



terface data length: Four bits

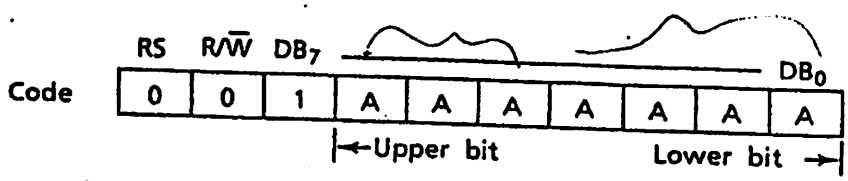


G RAM Address Set



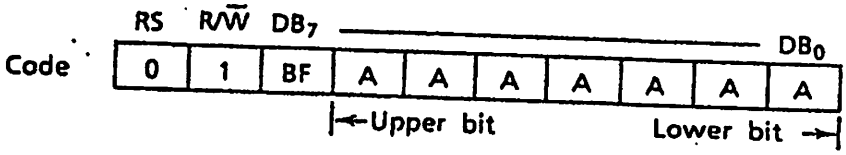
CG RAM addresses expressed as binary AAAAAA are set to the AC. Then data in
G RAM is written from or read to the MPU.

) RAM Address Set



DD RAM addresses expressed as binary AAAAAAA are set to the AC. Then data
DD RAM is written from or read to the MPU. The addresses used for display in
1 (AAAAAAA) are 00_H to 27_H and those for line 2 (AAAAAAA) are 40_H to 67_H.

y Flag/Address Read



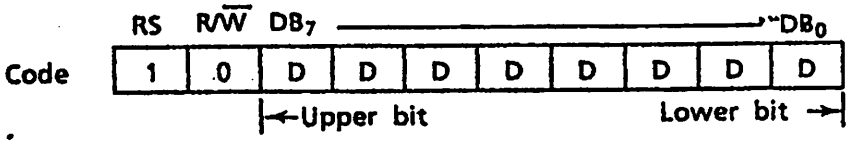
ne BF signal is read out, indicating that the module is working internally because
ne previous instruction.

hen BF = 1, the module is working internally and the next instruction cannot be accepted until the BF value becomes 0.

hen BF = 0, the next instruction can be accepted.

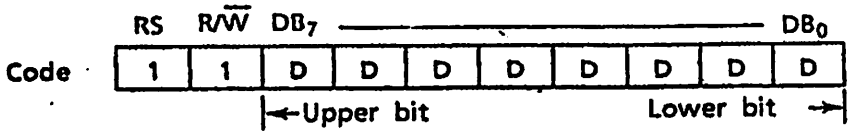
ierefore, make sure that BF = 0 before writing the next instruction. The AC
s of binary AAAAAAA are read out at the same time as reading the busy flag.
AC addresses are used for both CG RAM and DD RAM but the address set
e execution of the instruction determines which address is to be used.

ta Write to CG RAM or DD RAM



Binary eight-bit data DDDDDDDD is written into CG RAM or DD RAM. The CG AM Address Set instruction of (7) or the DD RAM Address Set instruction of (8) fore this instruction selects either RAM. After the write operation, the address d display shift are determined by the entry mode setting.

ata Read from CG RAM or DD RAM



Binary eight-bit data DDDDDDDD is read from CG RAM or DD RAM. The CG AM Address Set instruction of (7) or the DD RAM Address Set instruction of (8) efore this instruction selects either RAM. In addition, either instruction (7) or (8) ust be executed immediately before this instruction. If no address set instruction executed before a read instruction, the first data read becomes invalid. If read structions are executed consecutively, data is normally read from the second time. owever, if the cursor is shifted by the Cursor Shift instruction when reading DD AM, there is no need to execute an address set instruction because the Cursor Shift struction does this.
fter the read operation, the address is automatically incremented or decremented y one according to the entry mode, but the display is not shifted.

Note : The AC is automatically incremented or decremented by one according to the entry mode after a write instruction is executed to write data in CG RAM or DD RAM. However, the data of the RAM selected by the AC are not read out even if a read instruction is executed immediately afterwards.

Correct data is read out under the following conditions.

- An address set instruction is executed immediately before readout.
- For DD RAM, the Cursor Shift instruction is executed immediately before readout.
- The second, or later, instruction is executed in consecutive execution of read instructions.

imples of Instruction Use

nterface data length: Eight bits

Instruction	Display	Operation
Power-on RS | R/W | DB_7 —— DB_0 / | / |		The built-in reset circuit initializes the module.
unction Set ✓ RS | R/W | DB_7 —— DB_0 0 | 0 | 0 0 1 1 1 * * *		The interface data length is set to 8 bits. The character format becomes 5 x 7 dot-matrix at 1/16 duty cycle.
)isplay ON/OFF Control RS | R/W | DB_7 —— DB_0 0 | 0 | 0 0 0 0 1 1 1 0		The display and cursor are turned ON, but nothing is displayed.
ntry Mode Set RS | R/W | DB_7 —— DB_0 0 | 0 | 0 0 0 0 0 1 1 0		The address is incremented by one and the cursor shifts to the right in a write operation to internal RAM. The display is not shifted.
Vrite to CG RAM or DD RAM RS | R/W | DB_7 —— DB_0 1 | 0 | 0 1 0 0 1 1 0 0	L_	L is written. The AC is incremented by one and the cursor shifts to the right.
Vrite to CG RAM or DD RAM RS | R/W | DB_7 —— DB_0 1 | 0 | 0 1 0 0 0 0 1 1	LC_	C is written.
⋮	⋮	
Vrite to CG RAM or DD RAM RS | R/W | DB_7 —— DB_0 1 | 0 | 0 0 1 1 0 0 1 0	LCD MODULE M1632	2 is written in digit 16. Cursor disappears.

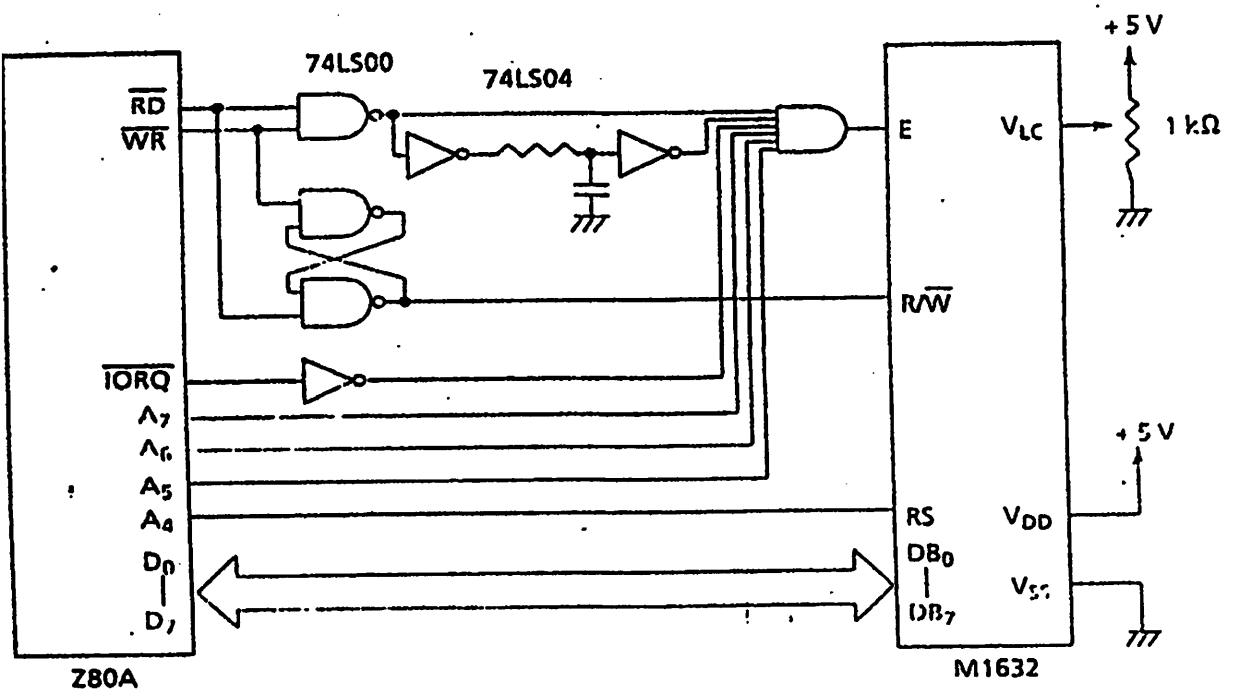
Instruction	Display	Operation
DD RAM address set RS: 0, R/W: 0, DB_7 —— DB_0: 1 1 0 0 0 0 0 0	LCD MODULE M1632 _	The DD RAM address is set so that the cursor appears at digit 1 of line 2.
Write to CG RAM or DD RAM RS: 1, R/W: 0, DB_7 —— DB_0: 0 0 1 1 0 0 0 1	LCD MODULE M1632 1_	1 is written.
Write to CG RAM or DD RAM RS: 1, R/W: 0, DB_7 —— DB_0: 0 0 1 1 0 1 1 0	LCD MODULE M1632 16_	6 is written.
⋮	⋮	
Write to CG RAM or DD RAM RS: 1, R/W: 0, DB_7 —— DB_0: 0 1 0 1 0 0 1 1	LCD MODULE M1632 16DIGITS,2LINES	S is written.
DD RAM address set RS: 0, R/W: 0, DB_7 —— DB_0: 1 0 0 0 0 0 0 0	LCD MODULE M1632 16DIGITS,2LINES	The cursor returns to the home position.
Display clear RS: 0, R/W: 0, DB_7 —— DB_0: 0 0 0 0 0 0 0 1	_	All the display disappears and the cursor remains at the home position.
⋮	⋮	

Interface data length: Four bits

Instruction	Display	Operation
Power-on RS: —, R/W: —, DB_7 — DB_4: —		The built-in reset circuit initializes the module.
Function Set RS: 0, R/W: 0, DB_7 — DB_4: 0 0 1 0		Four-bit operation mode is set. *Eight-bit operation mode is set by initialization, and the instruction is executed only once.
Function Set RS: 0, R/W: 0, DB_7 — DB_4: 0 0 1 0 RS: 0, R/W: 0, DB_7 — DB_4: 1 * * *		The 4-bit operation mode, 1/16 duty cycle, and 5 x 7 dot-matrix character format are selected. Then 4-bit operation mode starts.
Display ON/OFF Control RS: 0, R/W: 0, DB_7 — DB_4: 0 0 0 0 RS: 0, R/W: 0, DB_7 — DB_4: 1 1 1 0	_	The display and cursor are turned ON, but nothing is displayed.
Entry Mode Set RS: 0, R/W: 0, DB_7 — DB_4: 0 0 0 0 RS: 0, R/W: 0, DB_7 — DB_4: 0 1 1 0	_	The address is incremented by one and the cursor shifts to the right in a write operation to internal RAM. The display is not shifted.
Write to CG RAM or DD RAM. RS: 1, R/W: 0, DB_7 — DB_4: 0 1 0 0 RS: 1, R/W: 0, DB_7 — DB_4: 1 1 0 0	L_	L is written. the AC is incremented by one and the cursor shifts to the right.

J Connection Diagrams

Z80A



Z80A and 8255A