

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



INSTITUT TEKNOLOGI NASIONAL

SKRIPSI

**PERANCANGAN DAN PEMBUATAN SISTEM PRABAYAR
UNTUK PENUMPANG KERETA API MENGGUNAKAN KARTU RFID
BERBASIS MIKROKONTROLLER RENESAS**

MALANG

Disusun Oleh :

**BRENA SURANTA KABAN
00.17.267**



SEPTEMBER 2006

LEMBAR PERSETUJUAN



PERANCANGAN DAN PEMBUATAN SISTEM PRABAYAR UNTUK PENUMPANG KERETA API MENGGUNAKAN KARTU RFID BERBASIS MIKROKONTROLLER RENESAS

SKRIPSI

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

Disusun oleh:

BRENA SURANTA KABAN
NIM: 00.17.267

Diperiksa dan Disetujui



Ketua Jurusan Teknik Elektro S-1

(Ir. F. Yudi Limpraptono, MT.)
NIP. Y. 103 9500 274

Dosen Pembimbing

(Ir. F. Yudi Limpraptono, MT.)
NIP. Y. 103 9500 274

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



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

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama Mahasiswa : BRENA SURANTA KABAN
NIM : 00.17.267
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Judul Skripsi :

**“PERANCANGAN DAN PEMBUATAN SISTEM PRABAYAR UNTUK
PENUMPANG KERETA API MENGGUNAKAN KARTU RFID
BERBASIS MIKROKONTROLLER RENESAS“**

Dipertahankan Dihadapan Team Penguji Skripsi Jenjang Strata Satu (S-1), pada :

Hari : Kamis
Tanggal : 21 September 2006
Dengan nilai : 73,4 (B+)



Panitia Ujian Skripsi:

(Ir. Mochtar Asroni, MSME)
NIP. 1018100036

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

Anggota Penguji:

(Dr.Cahyo Crysdiان)
Penguji I

(Mohammad Ashar, ST, MT)
Penguji II

ABSTRAKSI

PERANCANGAN DAN PEMBUATAN SISTEM PRABAYAR UNTUK PENUMPANG KERETA API MENGGUNAKAN KARTU RFID BERBASIS MIKROKONTROLLER RENESAS

(Brena Suranta Kaban, 0017267, Jurusan Teknik Elektro S-1, Konsentrasi Teknik Elektronika, 53 halaman)

(Dosen Pembimbing Ir. F. Yudi Limpraptono, MT.)

Kata kunci : RFID, Mikrokontroler, RS 232, LCD, Keypad, Sistem Prabayar.

Seiring dengan perkembangan ilmu pengetahuan dan teknologi, manusia dituntut untuk melaksanakan segala sesuatu bukan hanya cepat, tetapi juga harus tepat. Untuk itu manusia senantiasa mencari cara untuk mempermudah pekerjaannya sehari-hari. Berkaitan dengan hal tersebut, manusia juga cenderung untuk mencari alternative pemecahan masalah bagi pekerjaan yang dilakukan secara manual sehingga pekerjaan tersebut dapat dikerjakan secara otomatis. Pada saat ini sistem penjualan karcis kereta api masih banyak dilakukan secara manual, dimana masih membutuhkan waktu yang lama dalam pelayanan penjualan karcis belum lagi dikarenakan oleh faktor kesalahan manusia atau human error.

Perancangan *sistem prabayar* untuk penumpang kereta api menggunakan kartu RFID terdiri atas minimum sistem *R8C/tiny* yang digunakan sebagai pengontrol utama dari keseluruhan sistem, dimana *RFID, Keypad, LCD, RS 232* dan PC yang digunakan untuk pengaplikasian dari keseluruhan sistem kerja alat prabayar untuk kereta api .

Alat ini diharapkan mampu menggantikan sistem penjualan karcis secara manual yang telah ada, sehingga efisiensi dan ketepatan sistem penjualan karcis kereta api dalam penggantian dari manual ke otomatisasi dapat tercapai.

KATA PENGANTAR

Puji dan syukur penyusun panjatkan ke hadirat Tuhan Yang Maha Esa yang telah melimpahkan rahmat dan petunjukNya sehingga penyusun dapat menyelesaikan skripsi yang berjudul “Perancangan dan Pembuatan Sistem Prabayar untuk Penumpang Kereta Api Menggunakan RFID Berbasis Mikrokontroler RENESAS”, laporan skripsi ini merupakan salah satu persyaratan kelulusan Program Strata 1 Teknik Elektro/Konsentrasi Elektronika Institut Teknologi Nasional Malang.

Keberhasilan penyusunan laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terimakasih kepada:

1. Bapak dan Ibu tercinta yang selalu memberikan kasih sayang, do'a restu, dukungan dan bantuan secara moril maupun materiil kepada penyusun, dengan segala dukungan orang tua dapat membesarkan hati penyusun menyelesaikan skripsi ini.
2. Bapak Dr. Ir. Abraham Lomi, MSEE. selaku Rektor Institut Teknologi Nasional Malang.
3. Bapak Ir. Mochtar Asroni, MSME. selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
4. Bapak Ir. F. Yudi Limpraptono, MT. selaku Ketua Jurusan Teknik Elektro S-1 Institut Teknologi Nasional Malang.
5. Bapak Ir. F. Yudi Limpraptono, MT. selaku Dosen Pembimbing.

6. Teman-teman yang telah memberikan motivasi serta bantuan baik berupa tenaga maupun pikiran dalam proses penyelesaian skripsi ini (*special thank's to Yoga Anggoro, ST*).

Penyusun telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam penyelesaian laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang bersifat membangun demi kesempurnaan laporan ini.

Malang, September 2006

Penyusun

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
BERITA ACARA UJIAN SKRIPSI.....	iii
ABSTRAKSI.....	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii

BAB I PENDAHULUAN

1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah	2
1.4. Tujuan	3
1.5. Metologi Penulisan	3
1.6. Sistematika Penulisan	4

BAB II LANDASAN TEORI

2.1. Pendahuluan.....	5
2.2. Mikrokontroler RENESAS R8C / Tiny	5

2.3. RFID	11
2.3.1. Overview RFID	11
2.3.2. Tag RFID.....	13
2.3.3. Pembaca (<i>reader</i>) RFID	15
2.3.4. Format Pembacaan ASCII.....	16
2.4. Komunikasi Serial	18
2.4.1. Metode Transmisi Serial	19
2.5. Sistem Komunikasi RS 232	20
2.5.1. Sinyal-Sinyal pada Interface RS 232	20
2.5.2. Interface EIA 232	22
2.5.2.1. Konfigurasi Pin	22
2.5.2.2. Sinyal Transmisi Data	23
2.6. <i>Liquid Crystal Display</i> (LCD).....	24
2.7. Keypad	27

BAB III PERENCANAAN DAN PEMBUATAN ALAT

3.1. Pendahuluan.....	28
3.2. Perancangan dan Pembuatan Perangkat Keras (<i>Hardware</i>)	30
3.2.1. Perencanaan MCU Renesas R8C / Tiny	30
3.2.2. Rangkaian Antarmuka RFID.....	32
3.2.3. Rangkaian RS 232	32
3.2.4. LCD (<i>Liquid Crystal Display</i>).....	33
3.2.5. Keypad	34

3.3. Perancangan Software	35
BAB IV PENGUJIAN ALAT	
4.1. Pendahuluan.....	38
4.2. Pengujian RFID	38
4.2.1. Tujuan Pengujian RFID	38
4.2.2. Peralatan yang digunakan.....	38
4.2.3. Langkah-langkah Pengujian RFID.....	39
4.2.4. Hasil Pengujian RFID	39
4.2.5. Analisa Hasil Pengujian RFID	40
4.3. Pengujian Komunikasi Serial	41
4.3.1. Tujuan.....	41
4.3.2. Peralatan yang digunakan.....	41
4.3.3. Langkah-langkah Pengujian.....	41
4.3.4. Hasil Pengujian	42
4.3.5. Analisa Hasil Pengujian	43
4.4. Pengujian LCD	43
4.4.1. Tujuan.....	43
4.4.2. Peralatan yang digunakan.....	43
4.4.3. Langkah-langkah Pengujian.....	44
4.4.4. Hasil Pengujian	44
4.4.5. Analisa Hasil Pengujian	45
4.5. Pengujian Keypad.....	45
4.5.1. Tujuan.....	45

4.5.2. Peralatan yang digunakan.....	45
4.5.3. Langkah-langkah Pengujian.....	45
4.5.4. Hasil Pengujian	46
4.5.5. Analisa Hasil Pengujian	46
4.6. Pengujian secara keseluruhan	46

BAB V PENUTUP

5.1. Kesimpulan	52
5.2. Saran	53

DAFTAR PUSTAKA

LAMPIRAN-LAMPIRAN

DAFTAR GAMBAR

2-1	Blok Diagram MCU Renesas.....	6
2-2	Memori Map	7
2-3	Konfigurasi Pin R8C / Tiny	8
2-4	Komunikasi antara Reader dan Transmitter (Tag).....	11
2-5	Bagian-bagian Tag RFID	14
2-6	Konfigurasi Pin ID-10 (RFID Reader).....	15
2-7	Konektor DB9	22
2-8	Rangkaian Interface RS 232	24
2-9	IC RS 232.....	24
2-10	Konfigurasi Kaki LCD.....	26
2-11	Keypad	27
3-1	Blok Diagram Perancangan	28
3-2	Rangkaian Minimum Sistem Mikrokontroler Renesas R8C/Tiny	31
3-3	Rangkaian RFID	32
3-4	Rangkaian RS 232.....	33
3-5	Perencanaan Rangkain LCD	34
3-6	Perencanaan Rangkain Keypad Matrik 4x4.....	35
4-1	Blok Diagram Pengujian RFID.....	39

4-2	Pengujian RFID	40
4-3	Blok Pengujian Komunikasi Serial dari Mikrokontroller ke PC	41
4-4	Blok Pengujian Komunikasi Serial dari PC ke Mikrokontroller	42
4-5	Tampilan Komunikasi Serial dari Mikrokontroller ke PC.....	42
4-6	Tampilan Komunikasi Serial dari PC ke Mikrokontroller.....	43
4-7	Flowchart Pengujian LCD	44
4-8	Tampilan LCD pada Pengujian.....	44
4-9	Rangkaian pengujian keypad	45
4-10	Blok Diagram Pengujian Sistem Keseluruhan.....	47
4-11	Tampilan PC Saat Pengujian Memasukkan identitas pemilik kartu prabayar	48
4-12	Tampilan LCD saat pengujian memasukkan password	48
4-13	Tampilan LCD Saat Pengujian Memasukkan password benar	49
4-14	Tampilan LCD Saat Pengujian Memasukkan password salah.....	49
4-15	Tampilan LCD Saat Pengujian menampilkan sisa jumlah saldo	49
4-16	Tampilan LCD saat pengujian memilih kota tujuan	50
4-17	Tampilan LCD saat pengujian saldo tidak mencukupi	50
4-18	Tampilan pada PC saat transaksi berlangsung.....	50
4-19	Laporan transaksi setiap pemilik kartu	51
4-20	Hasil cetakan tiket.....	51

DAFTAR TABEL

2-1	Perbandingan Teknologi Barcode dengan RFID	15
2-2	Fungsi Pin dan Format Data	16
2-3	Konfigurasi pin DB9	22
2-4	Nama dan Fungsi Penyemat pensinyalan pada modul LCD	25
4-1	Pengujian RFID	39
4-2	Hasil Pengujian Komunikasi Serial dari Mikrokontroler ke PC	42
4-3	Hasil Pengujian Komunikasi Serial dari PC ke Mikrokontroler	43
4-4	Hasil pengujian keypad	46

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi dewasa ini sangatlah pesat. Seiring dengan perkembangan tersebut, manusia dituntut untuk melaksanakan segala sesuatu bukan hanya cepat, tetapi juga harus tepat. Untuk itu manusia senantiasa mencari cara untuk mempermudah pekerjaannya sehari-hari. Berkaitan dengan hal tersebut, manusia juga cenderung untuk mencari alternative pemecahan masalah bagi pekerjaan yang dilakukan secara manual sehingga pekerjaan tersebut dapat dikerjakan secara otomatis. Salah satu perkembangan yang paling menonjol saat ini adalah perkembangan dibidang komputerisasi. Suatu system yang ditangani oleh computer, semuanya akan terasa lebih canggih, lebih pintar, lebih otomatis lebih praktis dan lebih efisien.

Saat ini sistem penjualan karcis kereta api masih banyak dilakukan secara manual, dimana masih membutuhkan waktu yang lama dalam pelayanan penjualan karcis belum lagi dikarenakan oleh faktor kesalahan manusia atau human error. Disamping itu penumpang tidak perlu lagi membawa uang tunai untuk membayar tiket kereta api, karena biasanya distasiun rawan sekali pencopetan.

Oleh karena hal tersebut diatas, maka diciptakan alat karcis prabayar pada stasiun kereta api yang diharapkan mampu menggantikan sistem penjualan karcis secara manual yang telah ada.

1.2. Rumusan Masalah

Dalam “Perancangan Dan Pembuatan Sistem Prabayar Untuk Penumpang Kereta Api Menggunakan Kartu RFID Berbasis Mikrokontroller RENESAS” terdapat beberapa permasalahan yang dihadapi, antara lain:

1. Bagaimana merancang dan membuat perangkat keras dan perangkat lunak alat karcis prabayar dengan RFID.
2. Bagaimana menerapkan mikrokontroller sebagai alat yang dapat berkomunikasi dengan computer menggunakan jaringan standart RS-232
3. Bagaimana cara mengolah data sehingga menghasilkan keluaran yang diinginkan
4. Bagaimana sistem dapat bekerja maksimal secara keseluruhan.
5. Data keluaran berupa apa saja

1.3. Batasan Masalah

Dalam penyusunan skripsi ini diperlukan batasan masalah agar tidak menyimpang dari ruang lingkup yang akan dibahas. Adapun batasan masalah adalah sebagai berikut:

1. Unit Pemroses utama (CPU) yang digunakan adalah mikrokontroler RENESAS
2. Kelas Kereta Api hanya terbatas untuk kelas ekonomi
3. Database diolah dengan menggunakan delphi 7
4. Tidak membahas tentang arsitektur komputer secara mendetail.

1.4. Tujuan

Adapun tujuan penulisan skripsi ini adalah untuk **Perancangan Dan Pembuatan Sistem Prabayar Untuk Penumpang Kereta Api Menggunakan Kartu RFID Berbasis Mikrokontroller RENESAS** Sehingga didapat suatu peralatan yang dapat menggantikan sistem penjualan karcis manual supaya lebih efisien.

1.5. Metodologi Penulisan

Metode yang digunakan dalam penulisan tugas akhir ini adalah

1. Studi Pustaka

Memperoleh data dengan cara membaca dan mempelajari buku literature yang berhubungan dengan penyusunan skripsi ini.

2. Studi Lapangan

Memperoleh data dengan cara praktek secara langsung untuk menunjang pembuatan alat.

3. Pengolahan Data

Mengolah data dengan jalan membuat analisa dan menarik kesimpulan dari hasil pengujian yang ada.

1.6. Sistematika Penulisan

Sistematika penulisan dari laporan ini adalah sebagai berikut :

BAB I PENDAHULUAN

Berisi latar belakang permasalahan, batasan masalah, metodologi dan sistematika penulisan.

BAB II TEORI PENUNJANG

Berisi landasan teori yang berhubungan dengan pembuatan alat

BAB III PERENCANAAN DAN PEMBUATAN ALAT

Meliputi tentang penjelasan dan tata cara perencanaan dan pembuatan alat.

BAB IV PENGUJIAN ALAT

Meliputi proses pengujian alat yang terdiri dari peralatan yang digunakan, langkah kerja dan analisa hasil pengujian.

BAB V PENUTUP

Meliputi kesimpulan dan saran

BAB II

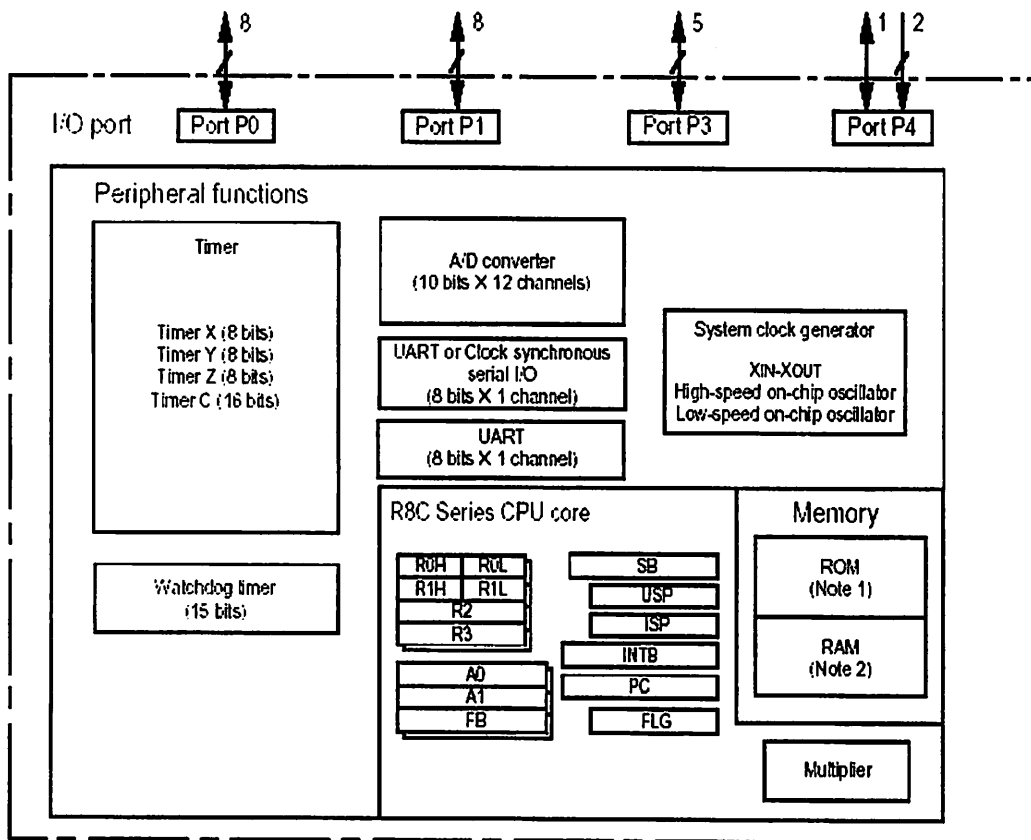
LANDASAN TEORI

2.1. Pendahuluan

Bab ini akan membahas teori-teori yang menunjang perencanaan dan pembuatan alat. Diawali dengan membahas tentang mikrokontroler RENESAS yang diterapkan untuk unit kontrol utama selain *Personal Computer (PC)*. Kemudian pada bagian lain juga dibahas tentang komunikasi data serial, IC MAX 232, RFID, keypad, LCD dan bahasa pemrograman Delphi 7 untuk tampilan pada PC.

2.2. Mikrokontroler RENESAS R8C / Tiny

Mikrokontroler RENESAS dibangun menggunakan proses gerbang silicon CMOS dengan kemampuan tinggi menggunakan CPU seri R8C/Tiny dan dikemas dalam modul plastic dengan jumlah pin sebanyak 32. Mikrokontroler ini beroperasi menggunakan perintah canggih khususnya efisiensi perintah dengan level tinggi. Mikrokontroler ini mempunyai 1 Mbytes kapasitas alamat, yang bisa digunakan untuk mengeksekusi perintah dengan kecepatan tinggi. Data flash ROM sebesar 2 KB x 2 blocks.

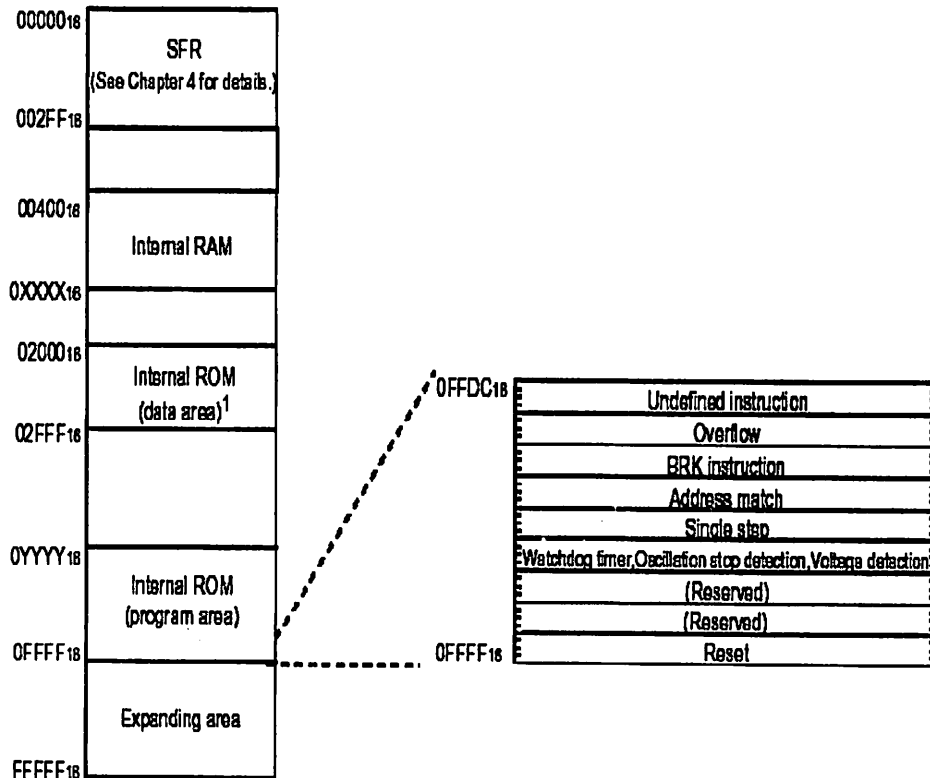


Gambar 2-1 Blok Diagram MCU Renesas
(Sumber : Datasheet Renesas)

Mikrokontroler Renesas R8C/Tiny mempunyai struktur memori yang terdiri atas :

- Space alamat hingga 1 Mbytes dari alamat 00000_{16} sampai $FFFFFF_{16}$.
- ROM internal (*program area*) dialokasikan pada alamat terendah dimulai dari alamat $0FFFF_{16}$. Misalnya, 16 Kbyte ROM internal dialokasikan pada alamat yang dimulai dari $0C000_{16}$ sampai $0FFFF_{16}$.
- ROM internal untuk *data area* dialokasikan pada alamat 02000_{16} sampai $02FFF_{16}$.
- Sedangkan RAM internal dialokasikan pada arah alamat yang lebih tinggi dimulai dari alamat 00400_{16} .

- Special function register (SFR) dialokasikan pada alamat mulai dari 00000₁₆ sampai 002FF₁₆. Fungsi register control peripheral dialokasikan disini.

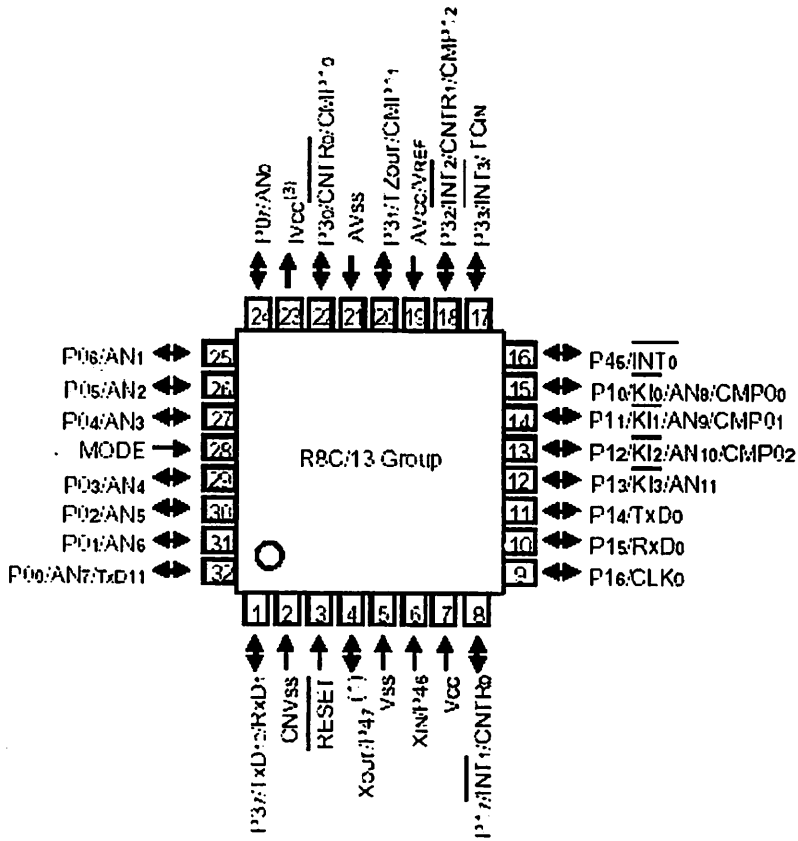


NOTES:

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 0YYYY ₁₆	Size	Address 0XXXX ₁₆
R5F21134FP, R5F21134DFP	16K bytes	0C000 ₁₆	1K bytes	007FF ₁₆
R5F21133FP, R5F21133DFP	12K bytes	0D000 ₁₆	768 bytes	006FF ₁₆
R5F21132FP, R5F21132DFP	8K bytes	0E000 ₁₆	512 bytes	005FF ₁₆

Gambar 2-2 Memory Map
(Sumber : Datasheet Renesas)



Gambar 2-3 Konfigurasi Pin R8C/Tiny
(Sumber : Datasheet Renesas)

Keterangan fungsi masing-masing pin :

1. V_{CC}

Digunakan untuk sumber tegangan dengan range nilai antara 2,7 – 5,5 Volt.

2. V_{SS}

Range tegangan 0 Volt

3. IV_{CC}

Pin ini digunakan untuk menyetabilkan sumber tegangan internal. Pin ini dihubungkan ke V_{SS} melalui kapasitor $0,1 \mu F$.

4. AV_{CC}, AV_{SS}

Pin ini merupakan input power supply untuk A/D Converter. Pin ini dihubungkan ke pin V_{CC} , sedangkan AV_{SS} dihubungkan ke V_{SS} . Hubungkan pin AV_{SS} dan AV_{CC} dengan kapasitor.

5. Reset

Merupakan input reset pada MCU.

6. CNV_{SS}

Pin ini dihubungkan ke V_{SS} melalui resistor.

7. Mode

Pin ini dihubungkan ke V_{CC} melalui resistor.

8. X_{IN}, X_{OUT}

Pin ini disediakan untuk pembangkitan rangkaian I/O pada clock utama.

Hubungkan resonator keramik atau osilator kristal antara X_{IN} dan X_{OUT} .

Untuk menggunakan clock derived external, masukkan ke pin X_{IN} dan pin X_{OUT} dibiarkan terbuka.

9. $INT_0 - INT_3$

Merupakan pin input interrupt

10. $KI_0 - KI_3$

Merupakan pin Key Input interrupt.

11. CNTR₀ (I/O)

Merupakan timer pin X I/O

12. CNTR₀ (O)

Merupakan timer pin X output.

13. CNTR₁

Merupakan timer pin Y I/O

14. TZ_{OUT}

Merupakan timer pin Z output.

15. TC_{IN}

Merupakan timer pin C input.

16. CMPO₀ – CMPO₃ dan CMPO₁₀ – CMPO₁₃

Merupakan timer pin C output.

17. CLK₀

Merupakan transfer clock untuk pin I/O.

18. RxD₀ dan RxD₁

Pin serial data input.

19. TxD₀, TxD₁₀ dan TxD₁₁

Pin serial data output.

20. V_{REF}

Referensi pin input tegangan untuk A/D Converter. Hubungkan V_{REF} ke

V_{CC}.

21. AN₀ – AN₁₁

Pin input analog untuk A/D Converter.

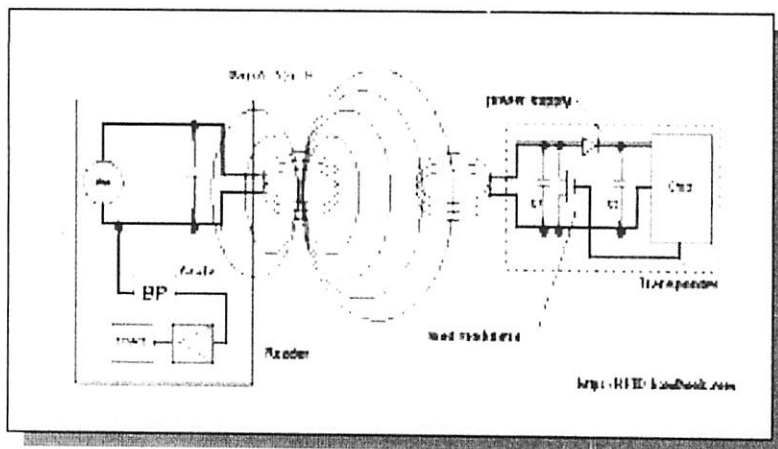
22. P0₀ – P0₁, P1₀ – P1₇, P3₀ – P3₃, P3₇ dan P4₅

Merupakan port 8 bit CMOS I/O. P1₀ – P1₇ juga berfungsi sebagai port LED driver.

2.3. RFID (Radio Frequency Identification)

2.3.1. Overview RFID

RFID adalah proses identifikasi seseorang atau objek dengan menggunakan frekuensi transmisi radio. RFID menggunakan frekuensi radio untuk membaca informasi dari sebuah devais kecil yang disebut tag atau *transponder* (*Transmitter + Responder*). Tag RFID akan mengenali diri sendiri ketika mendeteksi sinyal dari devais yang kompatibel, yaitu pembaca RFID (*RFID Reader*) dengan range kisaran pembacaan 8 cm serta bekerja pada frekuensi 125 KHz.



Gambar 2-4
Komunikasi Antara Reader dan Transmitter (Tag)
(Sumber: www.digiware.com/RFID)

RFID dapat disediakan dalam piranti (devais) yang hanya dapat dibaca saja (*Read Only*) atau dapat dibaca dan ditulis (*Read/Write*), tidak memerlukan kontak langsung maupun jalur cahaya untuk dapat beroperasi, dapat berfungsi pada berbagai variasi kondisi lingkungan, dan menyediakan tingkat integritas data yang tinggi. Sebagai tambahan, karena teknologi ini sulit untuk dipalsukan, maka RFID dapat menyediakan tingkat keamanan yang tinggi.

Pada sistem RFID umumnya, tag atau *transponder* ditempelkan pada suatu objek. Setiap tag membawa dapat membawa informasi yang unik, di antaranya: serial number, model, warna, tempat perakitan, dan data lain dari objek tersebut. Ketika tag ini melalui medan yang dihasilkan oleh pembaca RFID yang kompatibel, tag akan mentransmisikan informasi yang ada pada tag kepada pembaca RFID, sehingga proses identifikasi objek dapat dilakukan.

Sistem RFID terdiri dari empat komponen:

- Tag : Ini adalah devais yang menyimpan informasi untuk identifikasi objek. Tag RFID sering juga disebut sebagai *transponder*. Format dari tag pada perancangan ini adalah EM4001 atau tag kompatibel lainnya.
- Antena : untuk mentransmisikan sinyal frekuensi radio antara pembaca RFID dengan tag RFID.
- Pembaca RFID: adalah devais yang kompatibel dengan tag RFID yang akan berkomunikasi secara *wireless* dengan

tag. Digunakan Tipe ID-10 sebagai RFID reader pada perancangan ini.

- Software Aplikasi: adalah aplikasi pada sebuah workstation atau PC yang dapat membaca data dari tag melalui pembaca RFID. Baik tag dan pembaca RFID diperlengkapi dengan antena sehingga dapat menerima dan memancarkan gelombang elektromagnetik.

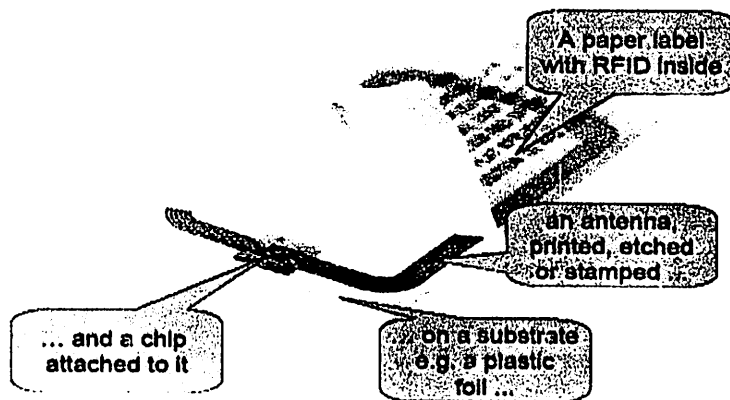
2.3.2. Tag RFID

Tag RFID adalah devais yang dibuat dari rangkaian elektronika dan antena yang terintegrasi di dalam rangkaian tersebut. Rangkaian elektronik dari tag RFID umumnya memiliki memori sehingga tag ini mempunyai kemampuan untuk menyimpan data. Memori pada tag secara dibagi menjadi sel-sel. Beberapa sel menyimpan data *Read Only*, misalnya *serial number* yang unik yang disimpan pada saat tag tersebut diproduksi. Sel lain pada RFID mungkin juga dapat ditulis dan dibaca secara berulang.

Berdasarkan cara daya tag, tag RFID dapat digolongkan menjadi:

- Tag Aktif: yaitu tag yang daya diperoleh dari baterai, sehingga akan mengurangi daya yang diperlukan oleh pembaca RFID dan tag dapat mengirimkan informasi dalam jarak yang lebih jauh. Kelemahan dari tipe tag ini adalah harganya yang mahal dan ukurannya yang lebih besar karena lebih kompleks. Semakin banyak fungsi yang dapat dilakukan oleh tag RFID maka rangkaianannya akan semakin kompleks dan ukurannya akan semakin besar.

- Tag Pasif: yaitu tag yang catu dayanya diperoleh dari medan yang dihasilkan oleh pembaca RFID. Rangkaianannya lebih sederhana, harganya jauh lebih murah, ukurannya kecil, dan lebih ringan. Kelemahannya adalah tag hanya dapat mengirimkan informasi dalam jarak yang dekat dan pembaca RFID harus menyediakan daya tambahan untuk tag RFID.



Gambar 2-5
Bagian bagian Tag RFID
(Sumber: Rivas, Mario, *RFID – its Applications and Benefit*)

Tag RFID telah sering dipertimbangkan untuk digunakan sebagai barcode pada masa yang akan datang. Pembacaan informasi pada tag RFID tidak memerlukan kontak sama sekali. Karena kemampuan rangkaian terintegrasi yang modern, maka tag RFID dapat menyimpan jauh lebih banyak informasi dibandingkan dengan barcode.

Pada tabel 2-1 diilustrasikan perbedaan utama antara teknologi barcode dengan RFID.

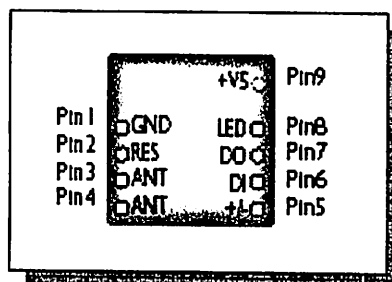
Tabel 2-1 Perbandingan Teknologi Barcode dengan RFID

Sistem	Barcode	RFID
Transmisi Data	Optik	Elektromagnetik
Ukuran Data	1 – 100 byte	128 – 8096 byte
Modifikasi Data	Tidak bisa	Bisa
Posisi Pembawa Data	Kontak cahaya	Tanpa kontak
Jarak Komunikasi	Beberapa meter	Dari cm sampai meter
Supseptibilitas Lingkungan	Debu	Dapat diabaikan
Pembacaan jamak	Tidak bisa	Bisa

2.3.3. Pembaca (reader) RFID

Pembaca RFID adalah merupakan penghubung antara software aplikasi dengan antena yang akan meradiasikan gelombang radio ke tag RFID. Gelombang radio yang diemisikan oleh antena berpropagasi pada ruangan di sekitarnya. Akibatnya data dapat berpindah secara *wireless* ke tag RFID yang berada berdekatan dengan antenna.

Berikut ini adalah konfigurasi pin yang terdapat pada RFID seperti yang terlihat pada gambar 2-6 di bawah ini :



Gambar 2-6
Konfigurasi Pin ID-10 (RFID Reader)
 (Sumber: www.digiware.com/RFID)

Adapun fungsi masing-masing pin dan format data pada RFID dapat dilihat pada Tabel 2-2 di bawah ini :

Tabel 2-2 Fungsi Pin dan Format Data

Pin No.	Description	ASCII	Wiegand26
Pin 1	Zero Volts and Tuning Capacitor Ground	GND 0V	GND 0V
Pin 2	Strap to +5V	Reset Bar	Reset Bar
Pin 3	To External Antenna and Tuning Capacitor	Antenna	Antenna
Pin 4	To External Antenna	Antenna	Antenna
Pin 5	Format Selector (+/-)	Strap to GND	Strap to +5V
Pin 6	Data 1	CMOS	One Output
Pin 7	Data 0	TTL Data (inverted)	Zero Output
Pin 8	3.1 kHz Logic	Beeper / LED	Beeper / LED
Pin 9	DC Voltage Supply	5V (+)	5V (+)

(Sumber: www.digiware.com/RFID)

2.3.4. Format Pembacaan ASCII

Salah satu tipe dari RFID reader ini yang digunakan pada alat ini adalah ID-10. RFID reader ini memiliki dua bentuk output serial yaitu: ASCII dan Wiegand 26-bit. Pada perancangan alat ini digunakan output dengan format ASCII, karena output ini sangat mudah untuk dihubungkan pada mikrokontroler.

Output yang memiliki format ASCII memiliki struktur sebagai berikut:

0 2	10 data karakter ASCII	checksum	CR	LF	0 3
-----	------------------------	----------	----	----	-----

Checksum merupakan hasil EXOR (Exclusive OR) dari 5 biner data byte.

Misalnya data output serial (dalam hexadesimal) yang kita tangkap adalah sebagai berikut:

O2	30	34	36	32	30	31	44	37	36	43	44	43	OD	OA	O3
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Langkah pertama adalah merubah semua nilai data diatas menjadi karakter ASCII.

Misalnya 30H menjadi karakter "0", 34H menjadi karakter "4", dst. Langkah kedua adalah menyusun data – data tersebut ke dalam Format Data ASCII seperti tabel 1. Kemudian ambil 10 data karakter ASCII. Dalam contoh ini berarti data tersebut adalah:

30	34	36	32	30	31	44	37	36	43	Data Hexsa
		6	2	0	1	D	7	6	C	Data ASCII

Untuk data dengan angka 30 dan 34 merupakan data untuk jenis-jenis kartu dan tidak digunakan dalam proses konversi, yang akan dipakai disini adalah data yang ke 3 s/d 10. Hasil konversi dari data heksa ke dalam data ASCII adalah "6201D76C". Gabungkan karakter data ASCII menjadi bilangan Hexadesimal, kemudian konversikan bilangan hexadesimal tersebut ke dalam desimal. Hasilnya sebagai berikut: 6201D76C H menjadi 1644287852. angka-angka ini merupakan

nomor kartu sebenarnya yang tertera pada badan kartu yang biasa disebut tag RFID.

2.4. Komunikasi Serial

Dalam dunia komunikasi ada dua cara pemindahan data yaitu secara paralel dan serial. Perbedaan dua cara tersebut terletak pada jumlah bit yang dipindahkan. Setiap data merupakan kelompok dari bit-bit tersebut, dikenal dengan istilah *byte* yang merupakan kelompok yang terdiri dari 8 bit, serta *word* yaitu kelompok yang terdiri dari 16 bit.

Pada komunikasi serial terjadi pemindahan data satu bit pada satuan waktu, sedangkan pada komunikasi paralel terjadi pemindahan data secara berurutan dari sekelompok bit pada satuan waktu.

Ditinjau dari arah komunikasi data, dikenal ada tiga cara yaitu:

1. Komunikasi *simplex* adalah sistem komunikasi data yang arah perpindahan datanya satu arah saja.
2. Komunikasi *half duplex* adalah sistem komunikasi data yang arah perpindahan datanya dua arah, namun proses pemindahannya tidak bersamaan.
3. Komunikasi *full duplex* adalah sistem komunikasi yang arah perpindahan datanya dua arah secara bersamaan.

Yang dimaksud dengan komunikasi secara serial adalah komunikasi dengan memanfaatkan hanya satu saluran sinyal untuk pengiriman dan penerimaan data. Untuk data digital yang merupakan serangkaian bit maka data dikirimkan bit per bit. Pengiriman akan dimulai dari LSB (*least Significant Bit*),

dan diakhiri dengan MSB (*Most Significant Bit*). Setiap karakter yang dikirimkan disusun sesuai dengan suatu urutan bit tertentu.

2.4.1. Metode Transmisi Serial

Ada dua metode transmisi serial yang digunakan yaitu:

2. Komunikasi Serial *Sinkron* (*Synchronous Serial Communication*) adalah clock dikirimkan bersama-sama dengan data serial.
3. Komunikasi Serial *Asinkron* (*Asynchronous Serial Communication*) adalah clock tidak dikirimkan bersama data serial, tetapi dibangkitkan sendiri-sendiri baik pada sisi pengirim (*transmitter*) maupun pada sisi penerima (*receiver*).

Pada IBM PC kompatibel port serialnya termasuk jenis asinkron. Komunikasi data serial ini dikerjakan oleh UART (*Universal Asynchronous Receiver/Transmitter*). IC UART dibuat khusus untuk mengubah data paralel menjadi data serial dan menerima data serial yang kemudian diubah kembali menjadi data paralel.

Pada UART, kecepatan pengiriman data (*baud rate*) dan fase clock pada sisi transmitter dan pada sisi receiver harus sinkron. Untuk itu diperlukan sinkronisasi antara transmitter dan receiver. Hal ini dilakukan oleh bit 'Start' dan bit 'stop'. Ketika saluran transmisi dalam keadaan idle, output UART adalah dalam keadaan logika '1'. Ketika transmitter ingin mengirimkan data, output UART akan diset lebih dulu ke logika '0' untuk waktu satu bit. Sinyal ini pada receiver akan dikenali sebagai sinyal 'start' yang digunakan untuk mensinkronkan fase clocknya sehingga dengan fase clock transmitter. Selanjutnya data akan

dikirimkan secara serial dari bit paling rendah (bit 0) sampai bit tertinggi. Selanjutnya, akan dikirim sinyal 'stop' sebagai akhir dari pengiriman data serial (Catur Edi dan Retna, 2004).

2.5. Sistem Komunikasi RS 232

Sistem komunikasi RS 232 merupakan sistem komunikasi serial standart pada setiap komputer. Ada 2 jenis keluaran yang tersedia pada komputer yaitu COM 1 (DB 9) dan COM 2 (DB 25). Komunikasi serial pada COM 1 untuk mouse dan COM 2 disediakan untuk komunikasi data.

2.5.1. Sinyal-Sinyal pada Interface RS 232

Sinyal yang dikeluarkan oleh komputer secara DCE maupun DTE adalah :

1. RLSD (*Received Line Signal Detect*)

Dengan saluran ini DCE memberitahukan ke DTE bahwa pada terminal masukan ada data yang masuk.

2. RxD (*Received Data*)

Sinyal yang digunakan DTE untuk menerima data dari DCE.

3. TxD (*Transmitted Data*)

Sinyal yang digunakan DTE untuk mengirimkan data ke DCE.

4. DTR (*Data Terminal Ready*)

Sinyal yang digunakan oleh PC untuk memberitahu kepada DCE bahwa PC siap menerima data dari DCE.

5. Signal Ground (GND)

Saluran untuk ground.

6. DSR (*Data Set Ready*)

Sinyal aktif pada saluran ini menunjukkan bahwa DTE sudah siap.

7. RTS (*Request To Send*)

Sinyal ini digunakan oleh PC Untuk memberitahu kepada DCE bahwa PC akan mengirim data. Apabila DCE dalam kondisi siap, maka modem akan memberikan sinyal CTS

8. CTS (*Clear To Send*)

Sinyal yang dikirimkan DCE untuk memberitahu kepada PC bahwa data data boleh dikirimkan.

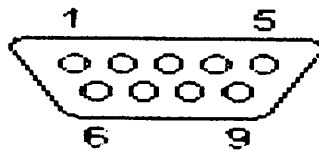
9. RI (*Ring Indikator*)

Pada saluran ini DCE memberitahu ke DTE bahwa stasiun menghendaki hubungan dengannya.

2.5.2. Interface EIA 232

2.5.2.1. Konfigurasi Pin

Sebuah komputer dapat berkomunikasi melalui *interface port I/O serial* yang disebut interface RS 232. Terdapat 9 pin dikenal dengan DB 9 yang terdapat pada bagian belakang komputer.



Gambar 2-7 Konektor DB 9

(Sumber : *Interfacing port paralel dan port serial dengan Delphi7*)

Fungsi dari pin-pin yang ada untuk konektor DB 9 adalah sebagai berikut :

Tabel 2-3 Konfigurasi Pin DB 9

EIA 232 (RS 232) Function	PIN
Data Carrier Detect (DCD)	1
Received Data (RxD)	2
Transmitted Data (TxD)	3
Data Terminal Ready (DTR)	4
Sinyal Ground (GND)	5
Data Set Ready (DSR)	6
Request To Send (RTS)	7
Clear To Send (CTS)	8
Ring Indicator (RI)	9

(Sumber : *B & B Electronics Mfg Co, USA, 2001, halaman 5*)

Jalur data (TxD dan RxD) digunakan untuk transport data, TxD adalah jalur output pada komputer, data dikirim dari pin ini sedangkan RxD adalah penerima untuk komputer, data yang datang akan diterima oleh pin ini. Pin

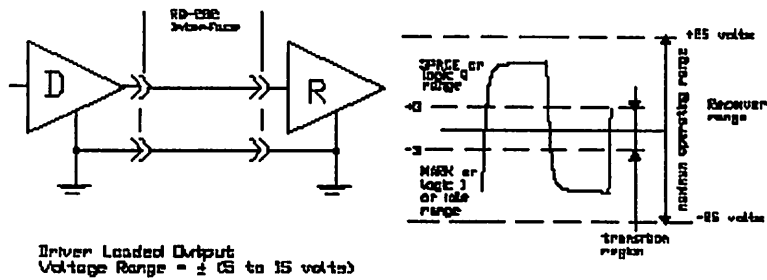
keempat adalah output (RTS) dimana sebuah sinyal akan diberikan pada alat yang dihubungkan dengan maksud meminta kiriman data. CTS adalah sinyal masukan yang menunggu sinyal dari alat yang terhubung. Ketika alat tersebut menerima sinyal RTS dan bisa menerima data maka ia akan mengirimkan sinyal balik yang merupakan CTS. DTR adalah sinyal keluaran yang memberi tanda bahwa ada alat yang terhubung dan akan mengirim data. DSR merupakan sinyal input yang mana jika alat yang terhubung menerima sinyal balik kemudian diterima sebagai sinyal DSR.

2.5.2.2. Sinyal Transmisi Data

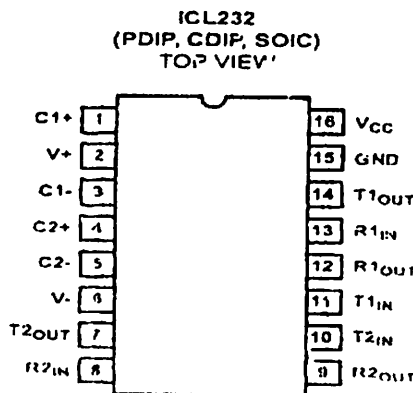
RS 232 dibuat untuk interface antara peralatan terminal data dan peralatan komunikasi data, dengan menggunakan data biner serial sebagai data yang ditransmisikan. Serial interface RS 232 memberi ketentuan logic level sebagai berikut:

1. Logic 1 disebut '*mark*' terletak antara -3 volt hingga -15 volt.
2. Logic 0 disebut '*space*' terletak antara $+3$ volt hingga $+15$ volt.
3. Daerah tegangan antara -3 volt hingga $+3$ volt adalah invalid level, yaitu daerah tegangan yang tidak memiliki logic sehingga daerah itu harus dihindari. Suatu saluran data RS 232 yang memberi keadaan tegangan ini berarti ada kesalahan. Demikian pula pada saluran pada daerah lebih dari -15 volt daerah lebih positif dari $+15$ volt.

Pada saat pengiriman data, sebelum mengirim data bit 0 harus diawali dengan start bit dahulu kemudian baru mengirim bit 0. Setelah mengirim bit 7 masih harus diakhiri dengan stop bit, jumlah stop bit dapat 1 bit, atau 2 stop bit atau 8 bit yang dibatasi oleh start bit dan stop bit disebut 1 frame.



Gambar 2-8
Rangkaian Interface RS 232
 (Sumber: B&B Electronics Manufacturing Company)



Gambar 2-9 IC RS 232
 (Sumber : Data Sheet Harris Semiconductor)

2.6. Liquid Crystal Display (LCD)

LCD atau *Liquid Crystal Display* merupakan sarana tampilan yang terdiri dari tumpukan tipis antar sel dari dua lembar kaca yang pinggirnya tertutup rapat. Diantara kedua lapisan tersebut diberi kristal cair yang tembus cahaya. Permukaan

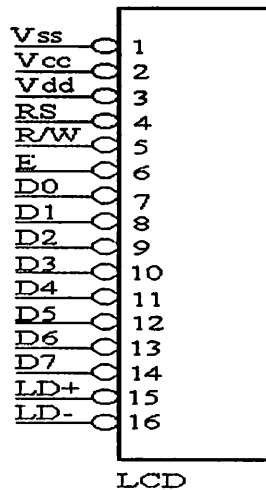
luar masing – masing keping kaca memiliki lapisan penghantar tembus cahaya. Sel memiliki ketebalan kira – kira 1×10^5 m yang diisi dengan kristal cair. Pada modul LCD TM162 keluaran Seiko Instrument, LCD memiliki 2 baris tampilan dengan 16 karakter setiap barisnya.

Tabel 2-4 Nama dan Fungsi Penyemat Pensinyalan pada Modul LCD

Nama	Fungsi
D0 – D7	Saluran data yang berisi perintah dan data LCD
Enable	Sinyal pengaktif komponen ‘1’ untuk mengaktifkan ‘0’ untuk tidak memilihnya
R/W	Selektor baca tulis ‘1’ untuk membaca ‘0’ untuk menulis
RS	Pemilih register ‘0’ untuk register inisialisasi (hanya tulis / write only) ‘1’ untuk register data (baca dan tulis)
VLC	Pengendali terang redupnya cahaya LCD
VCC	Catu daya positif (5 V)
VSS	Catu negatif (ground)

Sumber : Datasheet Seiko Instrument LCD, 1987

Konfigurasi kaki pada modul LCD dapat dijelaskan pada Gambar 2-10 di bawah ini :



Gambar 2-10 Konfigurasi Kaki LCD

Sumber : Datasheet Seiko Instrument LCD, 1987

Adapun karakteristik dari LCD TM162 adalah sebagai berikut:

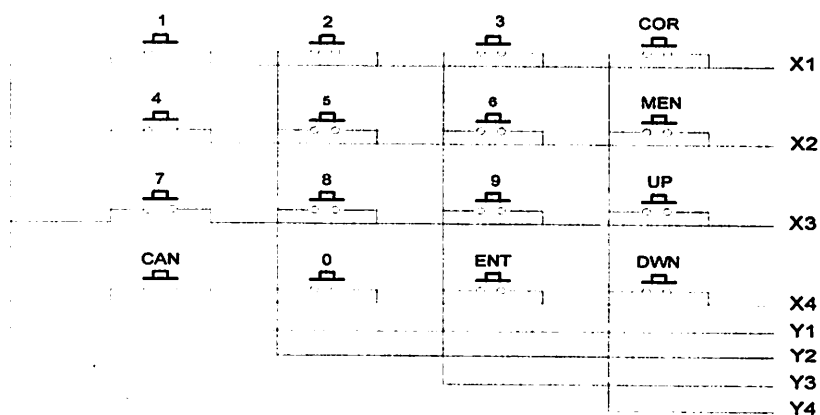
- 16x2 karakter dengan 5x7 dot matriks.
- ROM generator karakter dengan 192 tipe karakter
- RAM generator karakter dengan 8 tipe karakter (untuk program write).
- 80 x 8 bit RAM data display dengan 80 karakter maksimal
- Dapat diantarmukakan dengan MPU 4 atau 8 bit
- RAM data dan RAM generator karakter dapat dibaca dari MPU
- Rangkain osilator terpadu
- Catu daya tunggal 5V
- Reset otomatis terpadu
- Temperatur antara 0⁰C sampai 50⁰C

Adapun untuk penampilan karakter yang ada dilakukan dengan cara

memberikan kode karakter untuk tiap-tiap karakter yang diinginkan pada bus data dan menggunakan sinyal control E, RS, dan $\overline{R/W}$.

2.7. Keypad

Keypad digunakan sebagai sarana untuk memasukkan data ke minimum system. Keypad ini berfungsi untuk menampilkan data yang dimasukkan ke LCD. Keypad yang digunakan adalah keypad dengan 4 baris dan 4 kolom sehingga membentuk matriks 4 x 4. Pada keypad jenis ini, metode *scanning* tidak perlu digunakan. Untuk pengambilan data dari keypad dapat digunakan metode polling biasa. Pada kondisi tidak ada penekanan keypad akan berlogika 1 pada setiap bitnya dan sebaliknya apabila salah satu dari keypad ditekan, baris dan kolom yang berhubungan akan terhubung ke ground sehingga baris dan kolom tersebut akan berlogika 0. Gambar matriks keypad 4 x 4 terlihat pada gambar 2-11



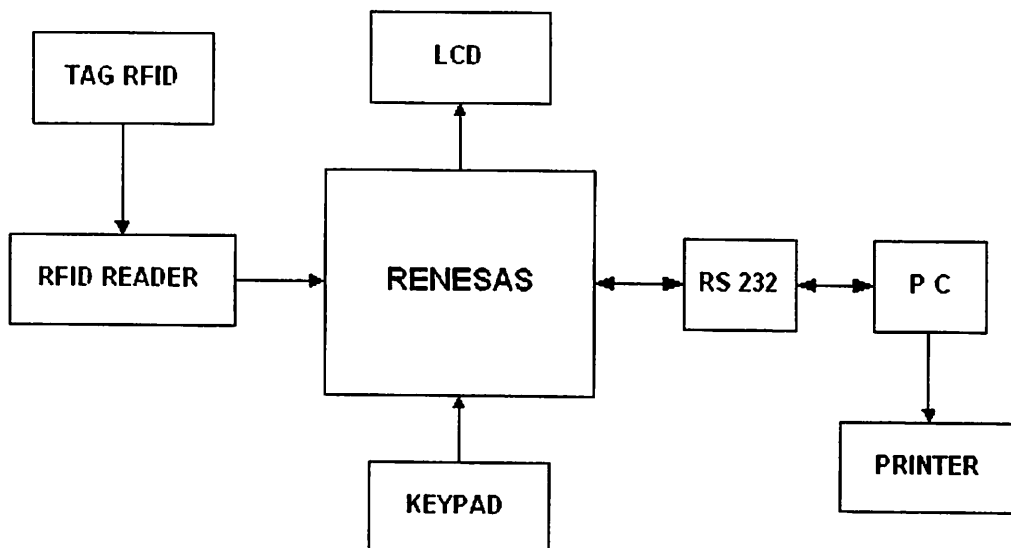
Gambar 2-11 Gambar Keypad

(Sumber: Teknik Antarmuka dan Pemrograman Mikrokontroler AT89C51)

BAB III**PERENCANAAN DAN PEMBUATAN ALAT****3.1. Pendahuluan**

Pada bab ini akan dibahas mengenai peralatan yang direncanakan dan akan direalisasikan sebagaimana fungsinya.

Perencanaan dan pembuatan alat ini menggunakan server berupa komputer dan memanfaatkan RS232 sebagai komunikasi *interface* antara alat dan komputer. Adapun blok diagram dari perancangan alat ini secara garis besar adalah sebagai berikut :



Gambar 3-1
Blok Diagram Perancangan

Berdasarkan blok diagram diatas, maka prinsip kerja dari alat tersebut dapat dijelaskan sebagai berikut.

- Tag RFID
Berfungsi memancarkan gelombang radio yang membawa serial number.
- RFID READER
Berfungsi untuk membaca serial number yang dikirim oleh tag RFID melalui gelombang radio.
- Mikrokontroler RENESAS
Berfungsi sebagai pengontrol masukan dan keluaran data.
- Keypad
Berfungsi untuk memasukan password, memilih kota tujuan, memasukkan jumlah tiket
- RS 232
Berfungsi sebagai interface antara mikrokontroler dengan PC.
- LCD
Berfungsi sebagai display atau tampilan
- PC (*Personal Computer*)
Berfungsi memproses dan menyimpan data yang dibutuhkan.
- Printer
Berfungsi sebagai pencetak karcis

Cara Kerja:

Cara kerja alat ini diawali dari tag RFID yang dibawa masing-masing penumpang memancarkan gelombang radio dan diterima oleh RFID reader. Gelombang radio tersebut membawa serial number yang akan dibaca oleh RFID reader. Data dari

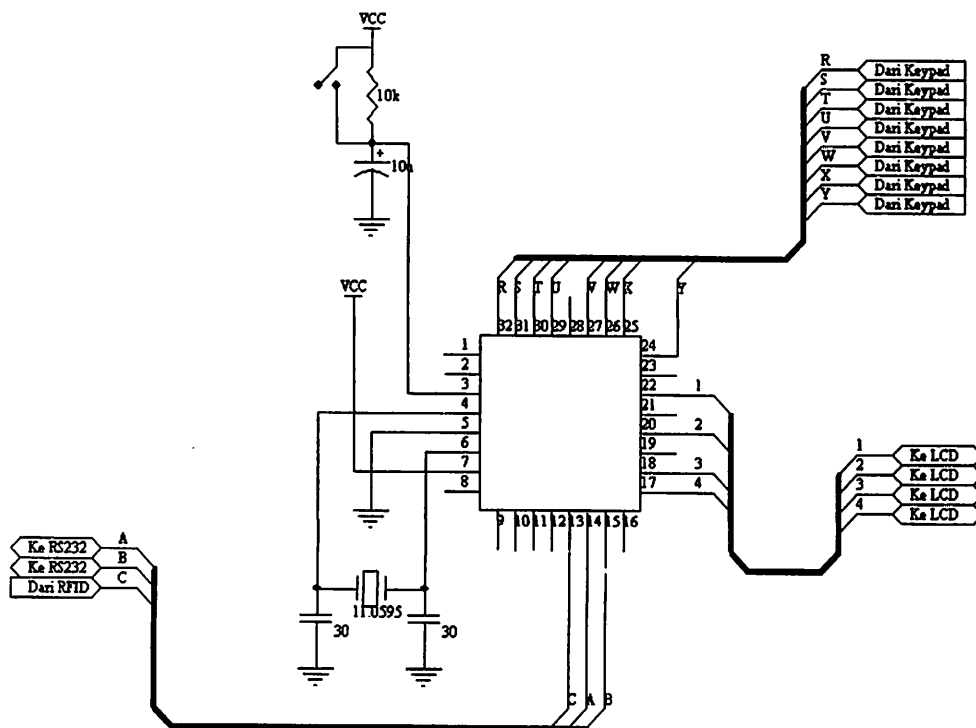
RFID reader akan diproses oleh mikrokontroler dan dikirim pada PC melalui komunikasi serial RS232. Selanjutnya data yang diterima oleh PC, diproses dan dicocokkan dengan data yang tersimpan pada PC. Setelah cocok, pemilik kartu RFID dapat menekan keypad untuk memasukkan password, setelah password benar, maka LCD akan menampilkan jumlah saldo kartu RFID yang dibawa, setelah itu dapat memilih tujuan yang akan tampil di layar LCD dengan menekan tombol arah panah atas atau ke bawah pada keypad, setelah cocok maka dapat menekan tombol enter pada keypad dengan demikian printer akan mencetak karcis berdasarkan tujuan dan banyak penumpangnya.

3.2. Perancangan dan Pembuatan Perangkat Keras (*Hardware*)

3.2.1. Perencanaan MCU Renesas R8C/Tiny

MCU merupakan rangkaian slave yang berfungsi sebagai penerima perintah-perintah dari PC. Beberapa alasan mengapa kita menggunakan MCU Renesas R8C/Tiny adalah rangkaiannya yang praktis karena tersusun dalam satu modul yang sangat mudah kita gunakan, sederhana dan tidak banyak memakan tempat.

Berikut ini gambar rangkaian MCU Renesas R8c/Tiny beserta hubungannya dengan rangkaian lain :



Gambar 3-2
Rangkaian Minimum Sistem Mikrokontroler Renesas R8C/Tiny

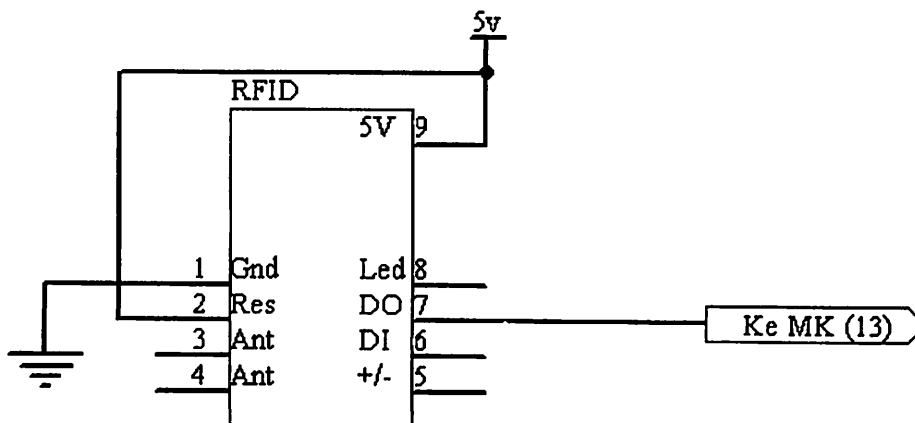
Mikrokontroler RENESAS adalah suatu modul yang terdiri dari 32 pin, dalam perancangan alat ini pin-pin yang digunakan dapat dilihat dalam Gambar 3.3, dan dijelaskan sebagai berikut:

- Pin 3 digunakan sebagai reset
- Pin 4 dan 6 digunakan sebagai *clock* mikrokontroler RENESAS
- Pin 7 (VCC) digunakan sebagai VCC Sumber untuk Digital
- Pin 13 terhubung dengan RFID *reader* yang digunakan sebagai alat pengenalan serial nomor tag RFID
- Pin 14 dan 15 digunakan sebagai komunikasi serial *interface* antara mikrokontroler RENESAS

- Pin 17, 18, 20 dan 22 terhubung pada LCD yang digunakan untuk tampilan.
- Pin 24, 25, 26, 27, 29, 30, 31, dan 32 terhubung dengan keypad

3.2.2. Rangkaian Antarmuka RFID

Kartu tag RFID ini akan digunakan sebagai media untuk mengenali ID dari kartu itu sendiri maupun identitas yang membawa kartu. Pada aplikasinya kartu tag RFID ini nantinya akan dibaca oleh RFID reader dan informasi yang termuat di dalamnya akan diteruskan ke mikrokontroler, setelah diproses akan diolah oleh PC. Gambar 3-4 dibawah ini adalah rangkaian dari RFID.

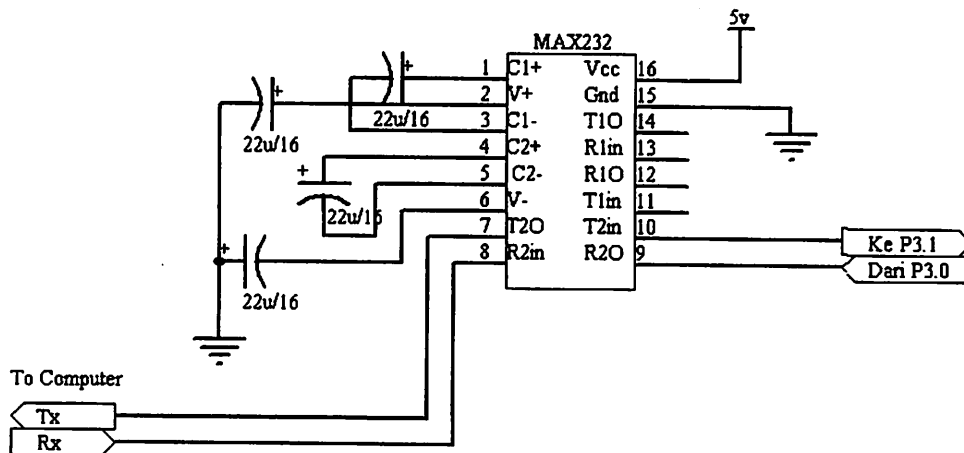


Gambar 3-3
Rangkaian RFID

3.2.3. RS 232

Sebelum diinputkan ke PC dibutuhkan rangkaian konverter tegangan. Mikrokontroler mempunyai output logika high dihasilkan dari tegangan 5 volt dan logika low sebesar 0 volt, tegangan ini akan sering mengakibatkan terjadinya

kesalahan pengiriman dan penerimaan data dikarenakan rugi-rugi dari kabel. RS 232 berfungsi untuk memperlebar range tegangan karena berada dikisaran +10 volt dan -10 volt, dengan range yang lebar ini kesalahan karena rugi-rugi sistem komunikasi dari mikrokontroler ke PC tidak mempengaruhi nilai data yang dikirimkan.



Gambar 3-4
Rangkaian RS 232

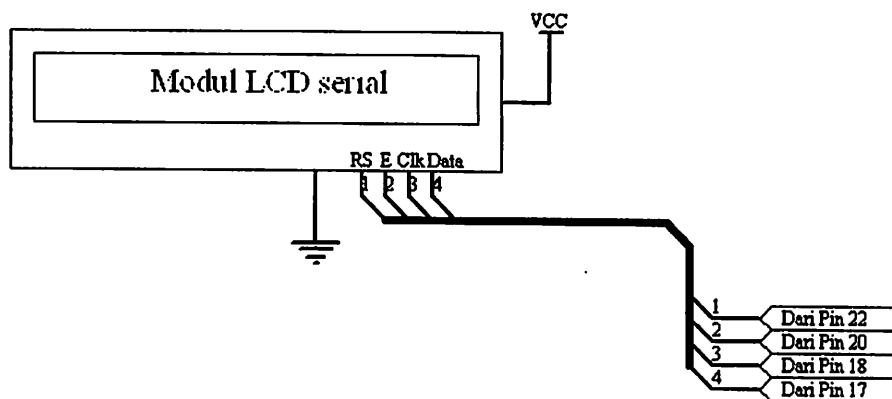
Untuk koneksi ke PC digunakan konektor DB9, dimana pin 15 dihubungkan dengan ground, pin 2 dan pin 3 pada port com dihubungkan dengan pin 7 dan pin 8 pada IC MAX 232 yang akan mengkonversikan tegangan

3.2.4. LCD (Liquid Crystal Display)

Sebagai tampilan perintah dan informasi. Disini menggunakan LCD M1632 (16 x 2), karena karakter yang ditampilkan dirancang tidak lebih dari 16 karakter per barisnya atau tidak lebih dari 32 karakter. Misalnya menampilkan

tulisan, "MASUKKAN PASSWORD:" atau menampilkan kota tujuan, jumlah saldo atau saldo tidak cukup yang dirancang tidak lebih dari 32 karakter.

Sinyal – sinyal yang dipergunakan oleh LCD adalah data bus, RS, R/W dan E. Sinyal E dihubungkan ke P2.1 untuk mengaktifkan LCD. LCD akan aktif jika Mikrokontroler memberikan instruksi tulis pada alamat LCD. Sedangkan P2.0 dipergunakan untuk memberikan sinyal RS yang membedakan data yang diberikan pada LCD. Sinyal RS diberikan ke LCD untuk membedakan sinyal antara instruksi program atau instruksi penulisan data. Untuk pin R/W akan berlogika low (0) apabila dihubungkan dengan ground maka LCD difungsikan hanya untuk menuliskan program atau data ke display. Untuk mengambil data dari Mikrokontroler maka Pin – Pin data dihubungkan dengan pin 17, 18, 20, dan 22 Pin – pin data dari Mikrokontroler.

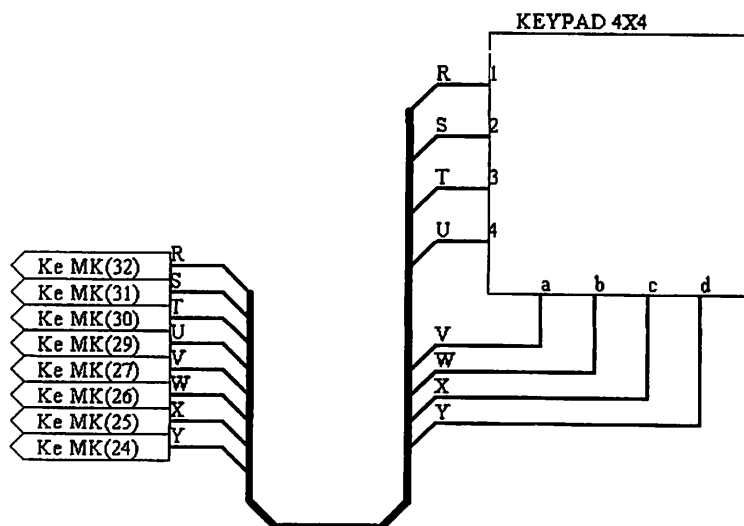


Gambar 3-5
Perencanaan Rangkain LCD

3.2.5. Keypad

Keypad digunakan untuk memasukkan *password*, pilihan kota tujuan. Keypad yang digunakan adalah keypad berukuran 4 x 4 (4 bagian kolom dan 4

bagian baris) yang terdiri dari saklar angka 0 sampai 9 sebagai penunjuk angka *password*, pilihan kota tujuan dan karakter *Enter*, *Cancel*, *Up*, *Down*, *Cord* dan *Men* yang digunakan sebagai proses menaikkan atau menurunkan *cursor* tiap baris pada LCD dan kompiramasi atau *exsekusi* pada terminal alat jika karakter yang ditampilkan lebih dari 32 karakter, pembatalan kesalahan penulisan, pembatalan proses inialisasi dan keluar dari proses inialisasi. Jumlah data yang dikirim oleh keypad pada mikrokontroler tergantung dari jumlah digit *password*, yaitu *password* sebanyak 4 digit. Gambar 3.7 memperlihatkan rangkain keypad matrik 4 x 4 yang terhubung dengan mikrokontroler.



Gambar 3-6
Perencanaan Rangkain Keypad Matrik 4x4

3.3. Perancangan *Software*

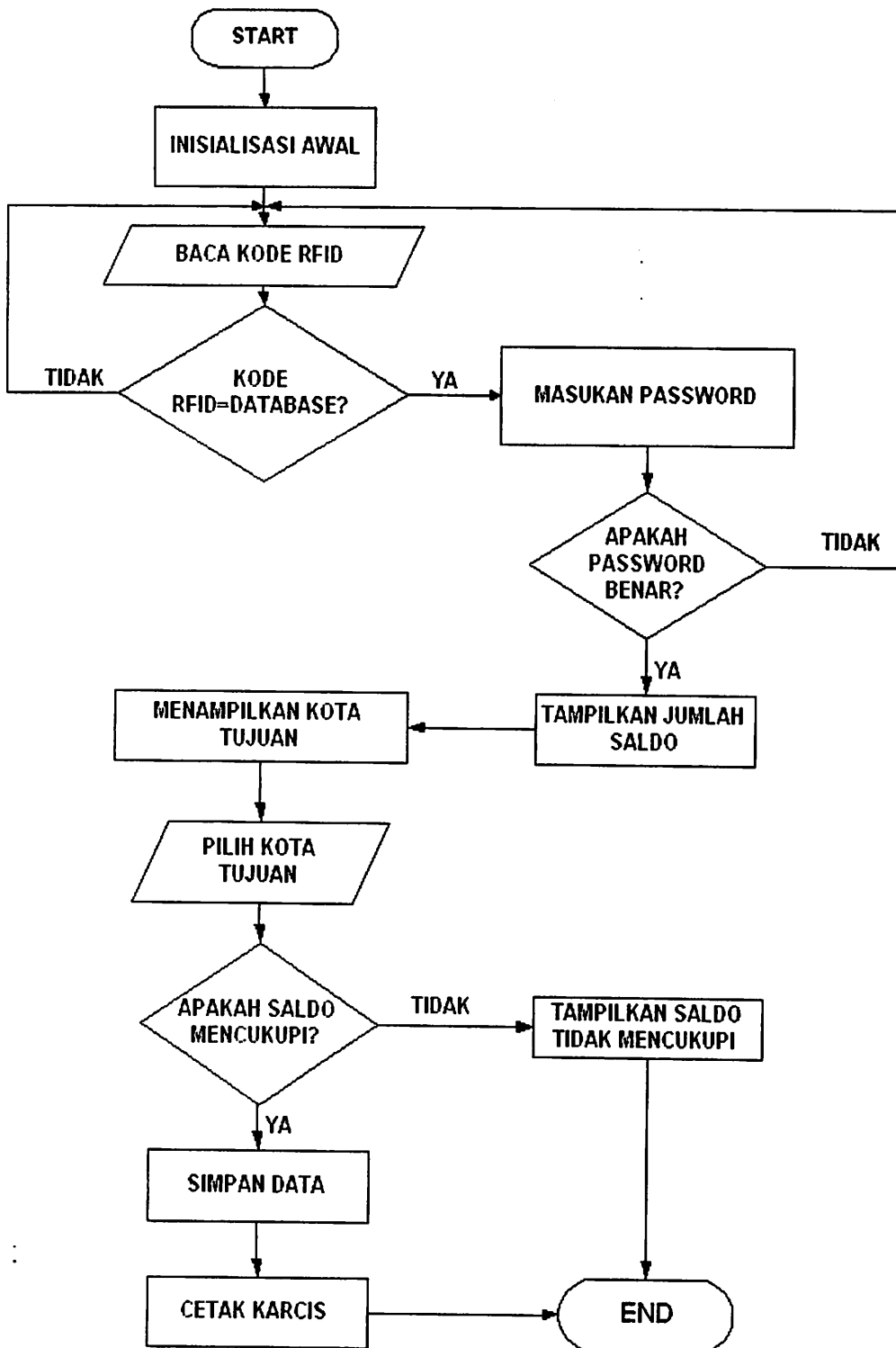
Komputer disini digunakan sebagai pengontrol pengiriman dan penerimaan data *password*, pemilihan kota tujuan, yang selanjutnya akan

mengolah permintaan dari *slave* mikrokontroller yang telah masuk dalam *data base* komputer. Komunikasi yang terjadi antara PC dan *slave* adalah komunikasi serial *full duplex*.

Langkah-langkah perencanaan *software*:

1. Perintah memasukkan *password*.
2. Konfirmasi *password.data base*.
3. Menampilkan jumlah saldo
4. Menampilkan kota tujuan
5. Memilih kota tujuan
6. Konfirmasi jumlah saldo dengan *database*.
7. Jika saldo tidak mencukupi, maka tampilkan saldo tidak mencukupi
8. Jika saldo mencukupi, maka simpan data yang masuk dan cetak karcis

FLOWCHART



BAB IV

PENGUJIAN ALAT

4.1. Pendahuluan

Bab ini akan membahas pengujian alat yang telah dirancang, dirakit serta direalisasikan. Tujuan pengujian alat ini adalah mengetahui kerja dari masing-masing sistem yang dibuat secara per-blok. Dengan demikian dapat diketahui kepresisian kerja dari alat yang direncanakan dan dibuat. Secara umum tujuan dari pengujian alat tersebut adalah sebagai berikut :

1. Mengetahui proses kerja dari masing-masing rangkaian.
2. Memudahkan pendataan spesifikasi alat.
3. Mengetahui hasil dari suatu perencanaan yang telah dibuat.
4. Memudahkan perawatan dan perbaikan apabila sewaktu-waktu terjadi kerusakan.

4.2. Pengujian RFID

4.2.1. Tujuan Pengujian RFID

Untuk mengetahui apakah pembaca RFID (*reader*) dapat menerima dan membaca informasi yang terdapat pada tag (kartu) RFID.

4.2.2. Peralatan yang digunakan

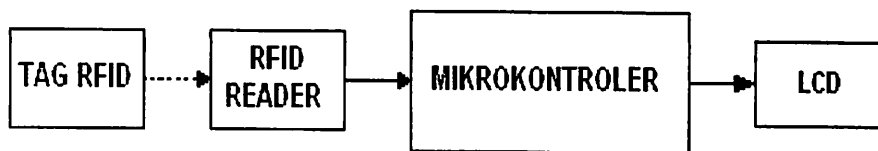
Peralatan yang digunakan adalah sebagai berikut :

- Pembaca RFID (*reader*)
- Sistem Mikrokontroller
- Tag RFID (kartu)

- Catu Daya

4.2.3. Langkah-langkah Pengujian RFID

1. Merangkai peralatan sesuai Gambar 4-2
2. Memberikan catu daya 5 volt pada rangkaian RFID
3. Mengaktifkan pembaca RFID
4. Mendekatkan Tag ke pembaca RFID dengan jarak 1 cm – 5cm.
5. Mengamati keluaran LCD



**Gambar 4-1
Blok Diagram Pengujian RFID**

4.2.4. Hasil Pengujian RFID

Hasil pengujian RFID ditunjukkan dalam tabel 4-2 berikut ini :

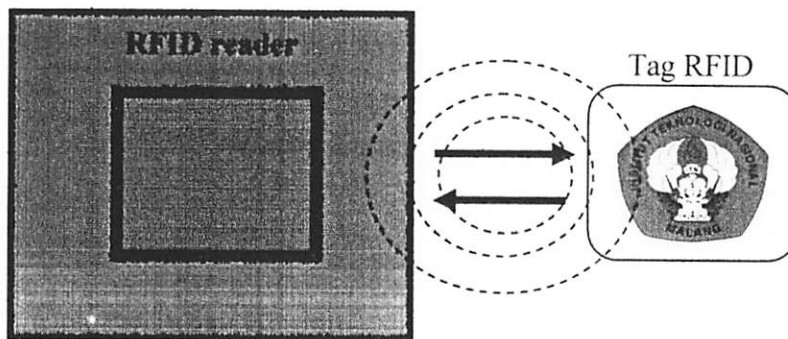
Tabel 4-1 Pengujian RFID

Tabel 4-1 Pengujian RFID Jarak jangkauan	Kepekaan Pembaca RFID
1 Cm	Sangat peka
2 Cm	Sangat Peka
3 Cm	Sangat Peka
4 Cm	Sangat Peka
5 Cm	Peka
6 Cm	Tidak peka
7 Cm	Tidak peka
8 Cm	Tidak peka

4.2.5. Analisa Hasil Pengujian RFID

Berdasarkan hasil pengujian di atas, terlihat bahwa pembaca RFID (*reader*) tersebut mampu mendeteksi dan membaca RFID dengan baik sampai sejauh 5 cm dan walaupun ada halangan non logam, RFID dapat bekerja dengan baik sesuai dengan yang direncanakan.

Pembaca (*reader*) tidak bisa mendeteksi dan membaca kartu RFID ketika terhalang oleh material berupa logam dalam hal ini besi dan aluminium dikarenakan komunikasi radio berkurang jika medan magnet harus menembus material yang mengurangi daya elektromagnetik, contohnya pada kasus objek dengan bahan logam. Tag RFID tidak akan terdeteksi ketika ditaruh di dalam logam, karena material logam akan meredam fluks magnet yang melalui tag secara drastis.



Gambar 4-2
Pengujian RFID

4.3. Pengujian Komunikasi Serial

4.3.1. Tujuan:

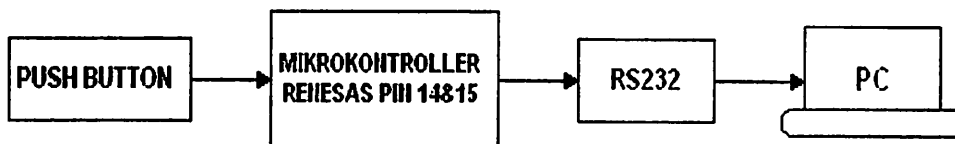
- Untuk mengetahui apakah komunikasi serial antara mikrokontroller dan PC dapat berjalan.

4.3.2. Peralatan yang digunakan:

- Power supply 5 volt.
- Mikrokontroller.
- PC.
- 8 LED.
- 8 Push Button.

4.3.3. Langkah-langkah pengujian:

1. Pengujian komunikasi serial dari mikrokontroller ke PC.
 - Membuat rangkaian seperti pada gambar dibawah ini:



Gambar 4-3

Blok Pengujian Komunikasi serial dari Mikrokontroller ke PC

- Membuat program pada mikrokontroller yang bisa mengambil data pada P2 mikrokontroller dan mengirimkan secara serial ke PC.
- Membuat program pada PC agar dapat membaca data yang dikirimkan melalui port serial oleh mikrokontroller.

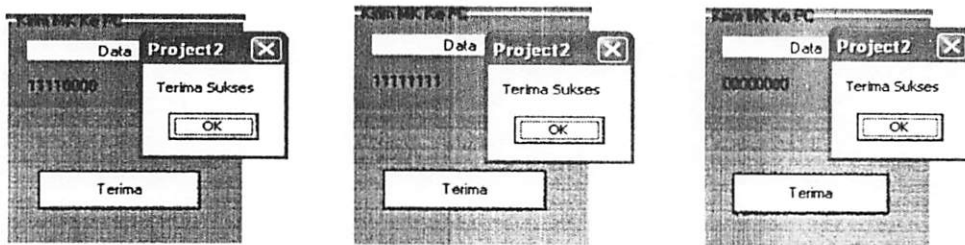
2. Pengujian komunikasi serial dari PC ke mikrokontroller.
 - Membuat rangkaian seperti pada gambar dibawah ini:



Gambar 4-4
Blok Pengujian Komunikasi serial dari PC ke Mikrokontroller

- Membuat program pada PC agar dapat mengirimkan data melalui port serial ke mikrokontroller.
- Membuat program pada mikrokontroller yang bisa mengambil data dari RXD dan mengirimkam pada P2.

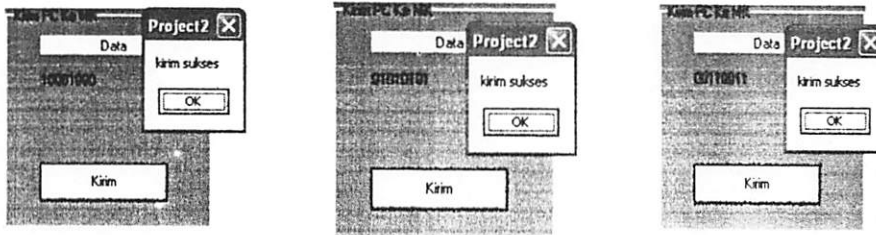
4.3.4. Hasil pengujian:



Gambar 4-5
Tampilan Komunikasi Serial dari Mikrokontroller ke PC

Tabel 4-2
Hasil Pengujian Komunikasi serial dari Mikrokontroller ke PC

DATA INPUT PADA P2								TAMPILAN PADA PC
PB1	PB2	PB3	PB4	PB5	PB6	PB7	PB8	
1	1	1	1	0	0	0	0	11110000
0	0	0	0	0	0	0	0	00000000
1	1	1	1	1	1	1	1	11111111



Gambar 4-6
Tampilan Komunikasi Serial dari PC ke Mikrokontroller

Tabel 4-3
Hasil Pengujian Komunikasi serial dari PC ke Mikrokontroller

DATA INPUT PC	LED 1	LED 2	LED 3	LED 4	LED 5	LED 6	LED 7	LED 8
01010101	0	1	0	1	0	1	0	1
10001000	1	0	0	0	1	0	0	0
00110011	0	0	1	1	0	0	1	1

4.3.5. Analisa Hasil Pengujian

Data yang dikirimkan oleh mikrokontroller dapat diterima dengan baik oleh PC, begitu juga sebaliknya. Ini berarti komunikasi serial antara mikrokontroller dan PC berjalan lancar.

4.4. Pengujian LCD

4.4.1. Tujuan Pengujian:

Untuk mengetahui apakah LCD dapat bekerja dengan baik, yaitu menampilkan karakter sesuai yang diinginkan berupa huruf dan angka.

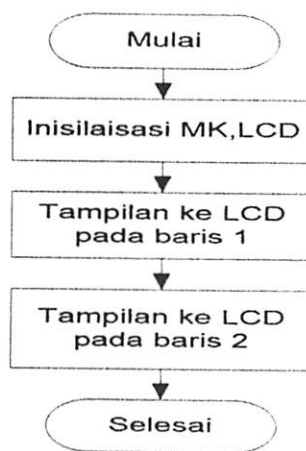
4.4.2. Peralatan yang digunakan

- o Mikrokontroler RENESAS

- LCD M1632.
- Catu Daya.

4.4.3. Langkah-langkah Pengujian

- Membuat program pada mikrokontroler untuk menampilkan karakter pada LCD dengan *flowchart* sebagai berikut:

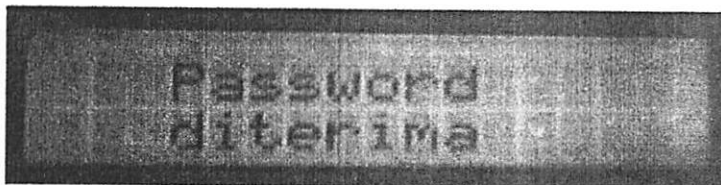


Gambar 4-7 Flowchart Pengujian LCD

- Mengamati tampilan pada LCD.

4.4.4. Hasil Pengujian

Hasil pengujian yang diperoleh adalah LCD dapat menampilkan karakter tersebut pada kedua baris tanpa ada kesalahan. Pada baris pertama "Password" dan pada baris kedua "diterima" seperti yang terlihat pada gambar.



Gambar 4-8 Tampilan LCD pada Pengujian

4.4.5. Analisa Hasil Pengujian

Dengan demikian rangkain interface modul LCD dapat berfungsi dengan baik dan dapat diakses oleh program.

4.5. Pengujian Keypad

4.5.1. Tujuan Pengujian

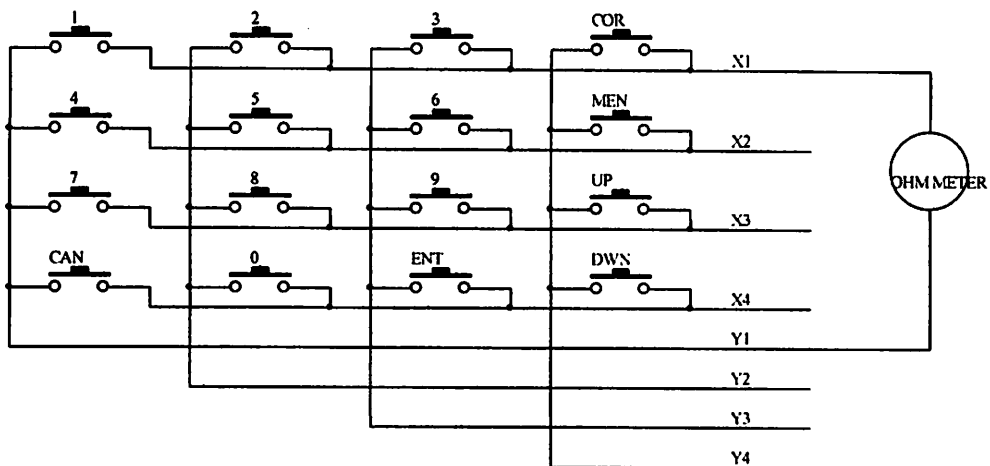
Agar dapat diketahui apakah keypad bisa digunakan sebagai input.

4.5.2. Peralatan yang digunakan

- Keypad
- Ohm Meter.

4.5.3. Langkah-langkah pengujian

- Merangkai keypad dan Ohm Meter seperti dibawah ini:



Gambar 4-9 Rangkaian Pengujian Keypad

- Menekan tombol keypad satu persatu. Pada saat menekan satu tombol, hubungkan satu *probe* Ohm Meter pada kaki X dan satu lagi pada kaki Y.
- Mengamati hasil pengujian pada Ohm Meter.

4.5.4. Hasil Pengujian

Tabel 4-4 Hasil Pengujian Keypad

TOMBOL	DATA DARI OHM METER							
	X1	X2	X3	X4	Y1	Y2	Y3	Y4
1	0	1	1	1	0	1	1	1
2	0	1	1	1	1	0	1	1
3	0	1	1	1	1	1	0	1
4	1	0	1	1	0	1	1	1
5	1	0	1	1	1	0	1	1
6	1	0	1	1	1	1	0	1
7	1	1	0	1	0	1	1	1
8	1	1	0	1	1	0	1	1
9	1	1	0	1	1	1	0	1
CAN	1	1	1	0	0	1	1	1
0	1	1	1	0	1	0	1	1
ENT	1	1	1	0	1	1	0	1
COR	0	1	1	1	1	1	1	0
MEN	1	0	1	1	1	1	1	0
UP	1	1	0	1	1	1	1	0
DWN	1	1	1	0	1	1	1	0

4.5.5. Analisis Hasil Pengujian

Keypad bekerja dengan baik dan sesuai dengan yang diharapkan

4.6. Pengujian Secara Keseluruhan

➤ Tujuan Pengujian

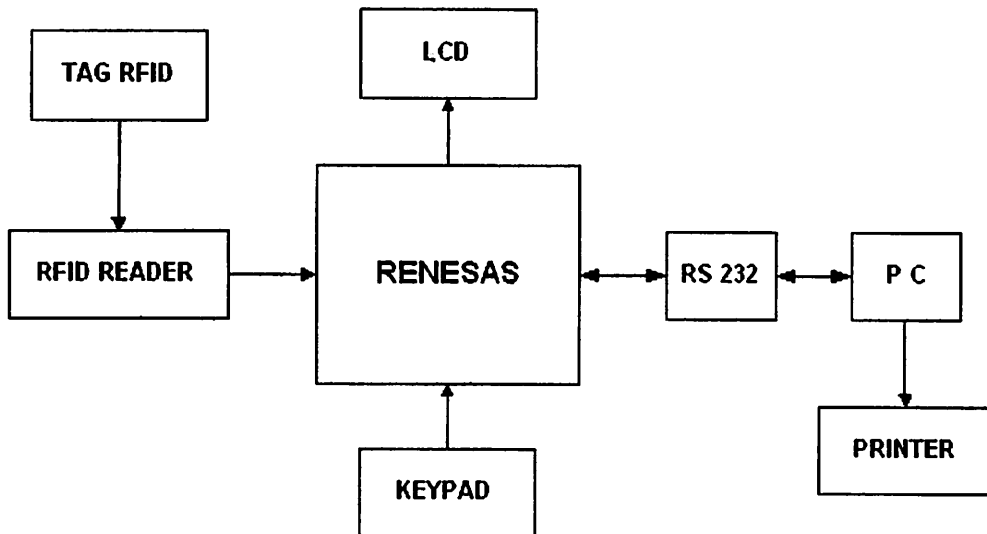
Untuk membuktikan bahwa sistem dapat bekerja dengan baik sesuai dengan perencanaan.

➤ Peralatan Yang digunakan

- PC
- Keseluruhan *Hardware*.

➤ **Prosedur Pengujian**

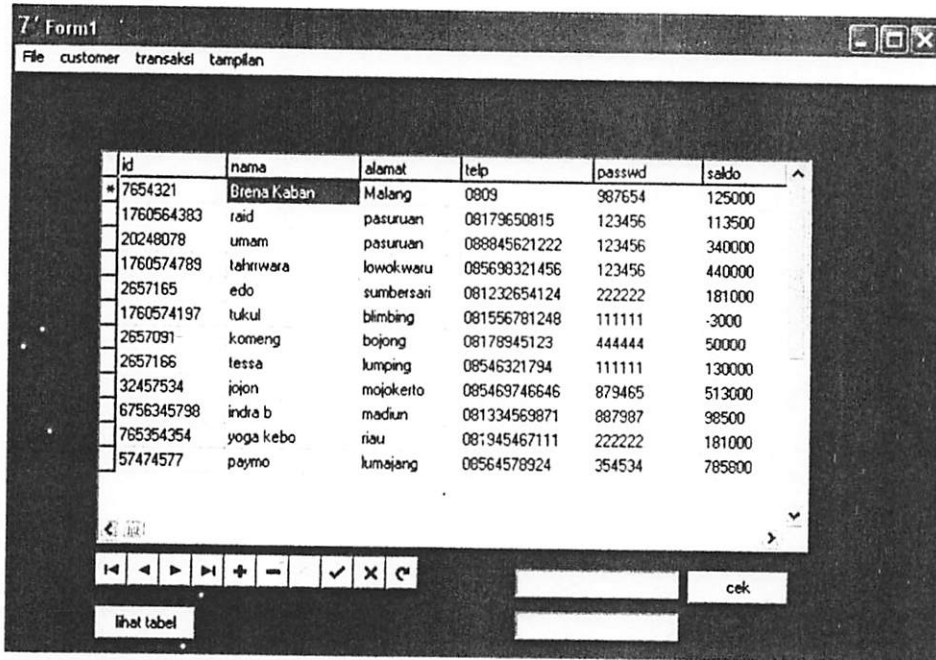
- Menginstalasi sebagaimana blok diagram di bawah



Gambar 4-10 Blok Diagram Pengujian Sistem Keseluruhan

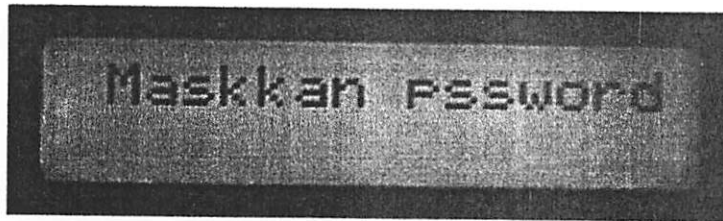
➤ **Memasukan data pada *Data Base***

- Memasukkan identitas pemilik kartu (Nama, Alamat, Telpon, dan Jumlah saldo)



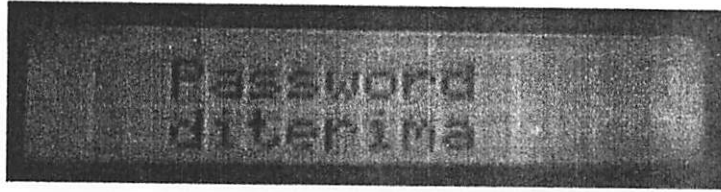
Gambar 4-11
Tampilan PC Saat Pengujian Memasukkan Identitas Pemilik Kartu Pra bayar

- o Memasukkan, *password* dari keypad



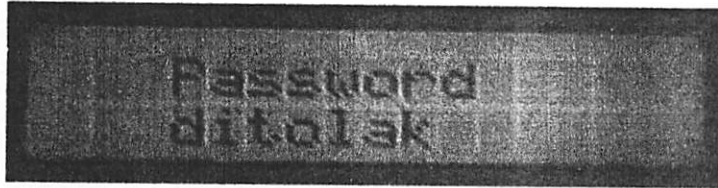
Gambar 4-12
Tampilan LCD Saat Pengujian Memasukkan *Password*

- o Memasukkan *Password* Benar pada Keypad, tampilan pada LCD seperti gambar 4-13

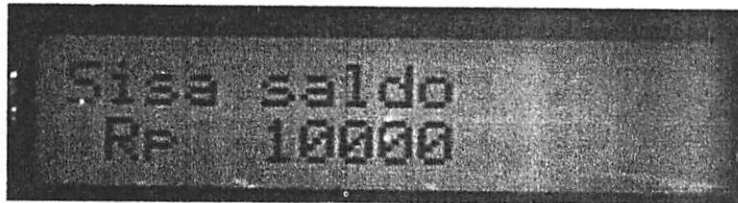


Gambar 4-13
Tampilan LCD Saat Pengujian Memasukkan *Password* Benar

- Memasukkan *Password* Salah pada *Keypad*, tampilan pada LCD seperti gambar 4-14

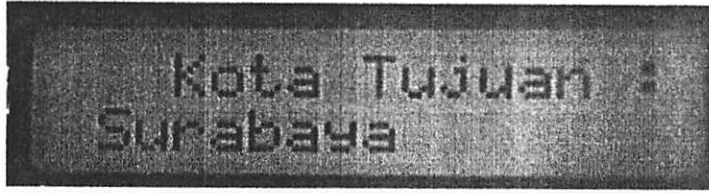


Gambar 4-14
Tampilan LCD Saat Pengujian Memasukkan *Password* Salah



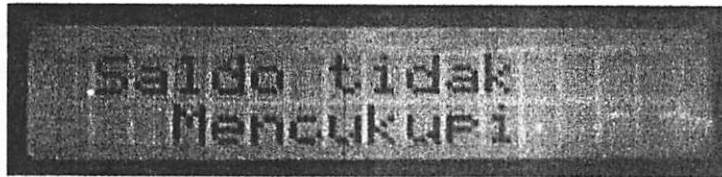
Gambar 4-15
Tampilan LCD Saat Pengujian Menampilkan Sisa Jumlah Saldo

- Memilih Kota Tujuan dengan menekan 'ENT' pada Kota Tujuan yang ditampilkan LCD.

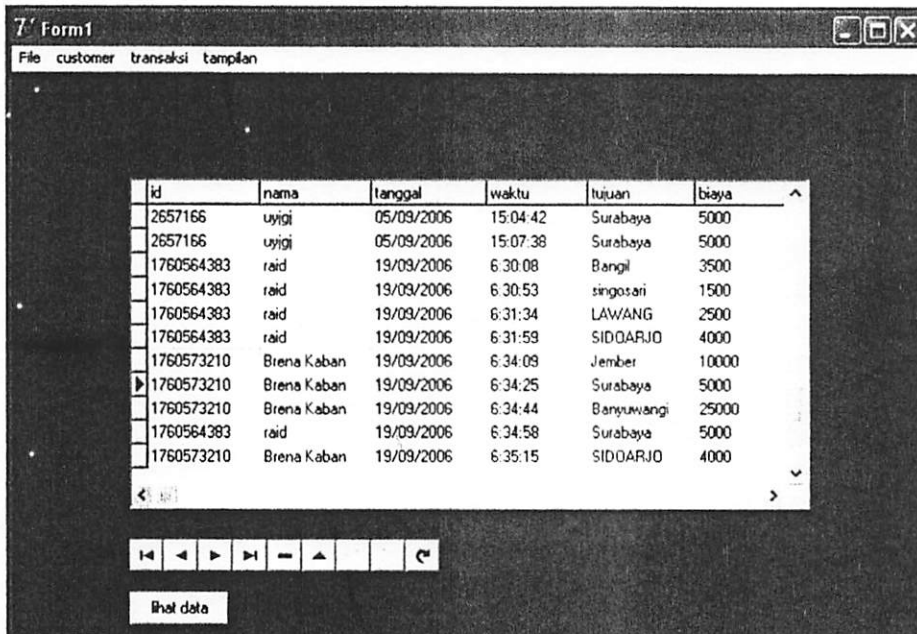


Gambar 4-16
Tampilan LCD Saat Pengujian Memilih Kota Tujuan

- o Menekan tombol krusor bawah pada *keypad* untuk kota selanjutnya dan menekan Krusor Atas untuk kembali ke kota sebelumnya.



Gambar 4-17
Tampilan LCD Saat Pengujian Saldo Tidak Mencukupi



Gambar 4-18
Tampilan PC Pada Saat Transaksi Berlangsung

7' Form1
File customer transaksi tampilan

ID	2657166
Nama	BRENA KABAN
Tanggal Transaksi	03/09/2006
Waktu Transaksi	0:51:20
Kota Tujuan	Surabaya
Biaya	5000
Saldo	176000

print

Gambar 4-19
Laporan Transaksi Setiap Pemilik Kartu

TIKET KERETA API PRABAYAR

Tanggal : 03/09/2006

Waktu : 0:51:20

Tujuan : Surabaya

Biaya : 5000

Gambar 4-20 hasil cetakan tiket

➤ **Analisa Hasil Pengujian**

Software dapat bekerja sesuai perancangan, yaitu dengan mengirim, menerima semua data dan menyimpan pada *data base*.

BAB V

PENUTUP

5.1. Kesimpulan

Dari hasil perancangan dan pembuatan alat maka didapat kesimpulan sebagai berikut :

1. Pembuatan alat ini menggunakan *Reader* tipe ID-10 dengan *internal* antena. Jarak maksimal komunikasi antara pembaca (*reader*) dengan kartu (*tag*) 5 cm, Kartu yang digunakan bersifat *pasif (read only)*
2. Kartu RFID merupakan media yang tepat untuk menjaga keasliannya karena setiap kartu hanya memiliki sebuah *serial number*. Dari 8 kali percobaan sebuah kartu, tingkat kebenaran data dari kartu RFID berupa *serial number* tersebut tidak ada kesalahan dalam pengiriman data, Sesuai dengan data yang terdapat dalam kartu RFID tersebut.
3. Dari hasil pengujian keseluruhan sistem, dimana *keypad* difungsikan untuk memasukkan *Password user*, LCD sebagai penampil *password* dari *keypad*, dan *database* pada PC (*Personal Computer*) diperoleh bahwa sistem berjalan dengan baik.
4. Tag RFID tidak akan terdeteksi ketika ditaruh di dalam logam, karena material logam akan meredam fluks magnet yang melalui *tag* secara drastis.

5.2. Saran

1. Alat kartu Prabayar ini masih menggunakan kartu RFID dan LCD, mungkin selanjutnya dapat dikembangkan dengan menggunakan sidik jari, yang mana tingkat keamanannya yang lebih bagus
2. Agar proses pencetakan tiket lebih cepat dan hasil yang bagus mungkin kedepan dapat digunakan printer yang lebih bagus, laser jet, dll.

Daftar Pustaka

1. Ibnu Malik. *Moh. Belajar Mikrontroller RENESAS*: Gava Media Yogyakarta, 2003
2. Green, DC. *Komunikasi Data*. Yogyakarta, 2000
3. Malvino, Hanapi Gunawan, *Prinsip Prinsip Elektronik*, (Edisi Kedua) Erlangga, Jakarta, 1999.
4. <http://www.my-siemens.com>
5. <http://www.atmel.com>
6. <http://www.nasional.com>
7. MAX 232 *Data Sheet*.
8. <http://www.digiware.com/RFID>
9. ID-10 (RFID Reader) *Datasheet*
10. LCD *Datasheet* (<http://www.senet.com.au/~cpeacock>)



INSTITUT TEKNOLOGI NASIONAL
Kampus I : Jl. Bendungan Sigura-gura No 2
Kampus II : Jl. Raya Karanglo Km 2

FORMULIR BIMBINGAN SKRIPSI

Nama : Brena Suranta Kaban
Nim : 00.17.267
Masa Bimbingan : 17 Juli 2006 s/d 17 Januari 2007/8
Judul : Perancangan dan Pembuatan Sistem Prabayar untuk Penumpang Kereta Api Menggunakan Kartu RFID Berbasis Mikrocontroller Renesas

No	Tanggal	Uraian	Paraf Pembimbing
1	5/9 2006	Bab I, II, III	
2		Demu	
3		Bab IV & V	
4			
5			
6			
7			
8			
9			
10			

Malang,

Dosen Pembimbing

(Ir. F. Yudi Limpraptono, MT)

Nip. 1039500274

Form S-4a



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

LEMBAR BIMBINGAN SKRIPSI


1. Nama : BRENA SURANTA KABAN
2. NIM : 00.17.267
3. Jurusan : Teknik Elektro S-1
4. Konsentrasi : Teknik Elektronika
5. Judul Skripsi :

“PERANCANGAN DAN PEMBUATAN SISTEM PRABAYAR UNTUK
PENUMPANG KERETA API MENGGUNAKAN KARTU RFID
BERBASIS MIKROKONTROLLER RENESAS “


6. Tanggal Pengajuan Skripsi : 3 Juni 2006
7. Selesai Menulis Skripsi : 21 September 2006
8. Dosen Pembimbing : Ir. F. Yudi Limpraptono, MT
9. Telah Dievaluasi Dengan Nilai : 80

Diperiksa dan Disetujui:

Ketua Jurusan Teknik Elektro S-1


(Ir. F. Yudi Limpraptono, MT)
NIP.Y. 103 9500 274

Dosen Pembimbing


(Ir. F. Yudi Limpraptono, MT)
NIP.Y. 103 9500 274




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

FORMULIR PERBAIKAN SKRIPSI


Nama : BRENA SURANTA KABAN
NIM : 00.17.267
Masa Bimbingan : 17 Juli 2006 s/d 17 Januari 2007
Judul :

“PERANCANGAN DAN PEMBUATAN SISTEM PRABAYAR UNTUK
PENUMPANG KERETA API MENGGUNAKAN KARTU RFID
BERBASIS MIKROKONTROLLER RENESAS “

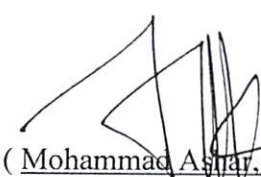
Tanggal	Uraian	Paraf
21 September 2006	<ul style="list-style-type: none">• Mekanisme komunikasi RFID reader dengan Mikrokontroller Renesas• Kecepatan transmisi data antara RFID reader ke Mikrokontroller• Jarak terjauh komunikasi RFID reader ke Mikrokontroller• Impedansi dari sistem• Arus total saat sistem beroperasi• Pengujian RFID• Kesimpulan	

Disetujui:

Penguji I


(Dr. Cahyo Crysdiان)

Penguji II


(Mohammad Asrar, ST, MT)

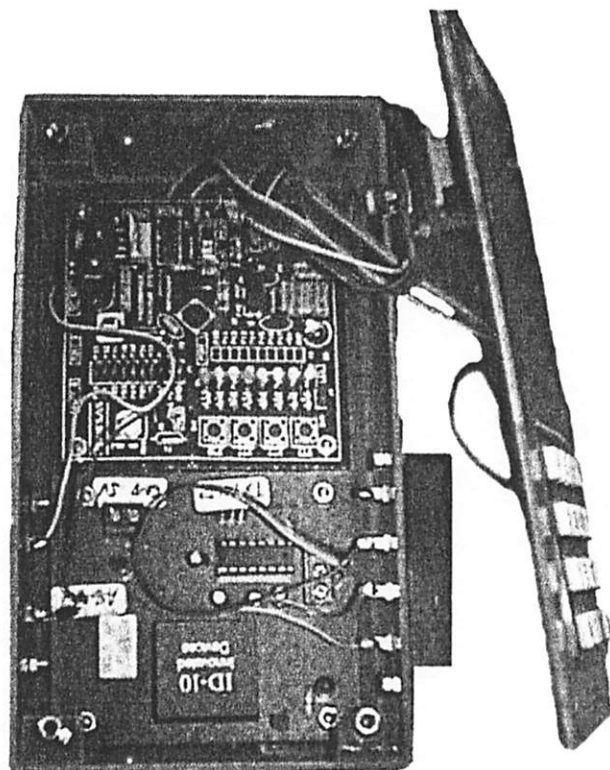
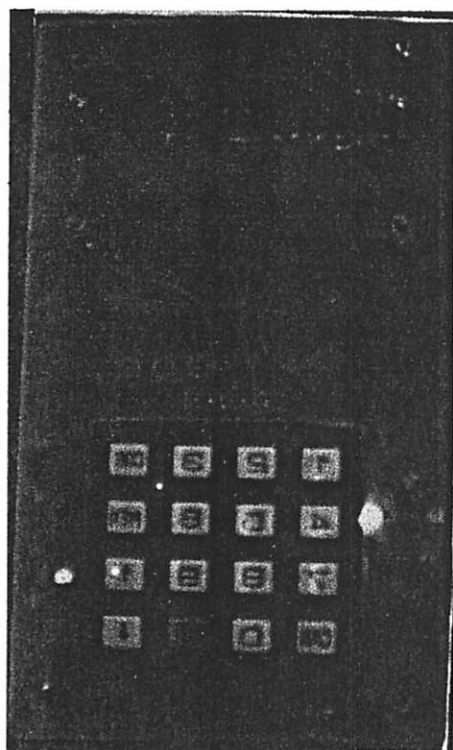
Mengetahui,

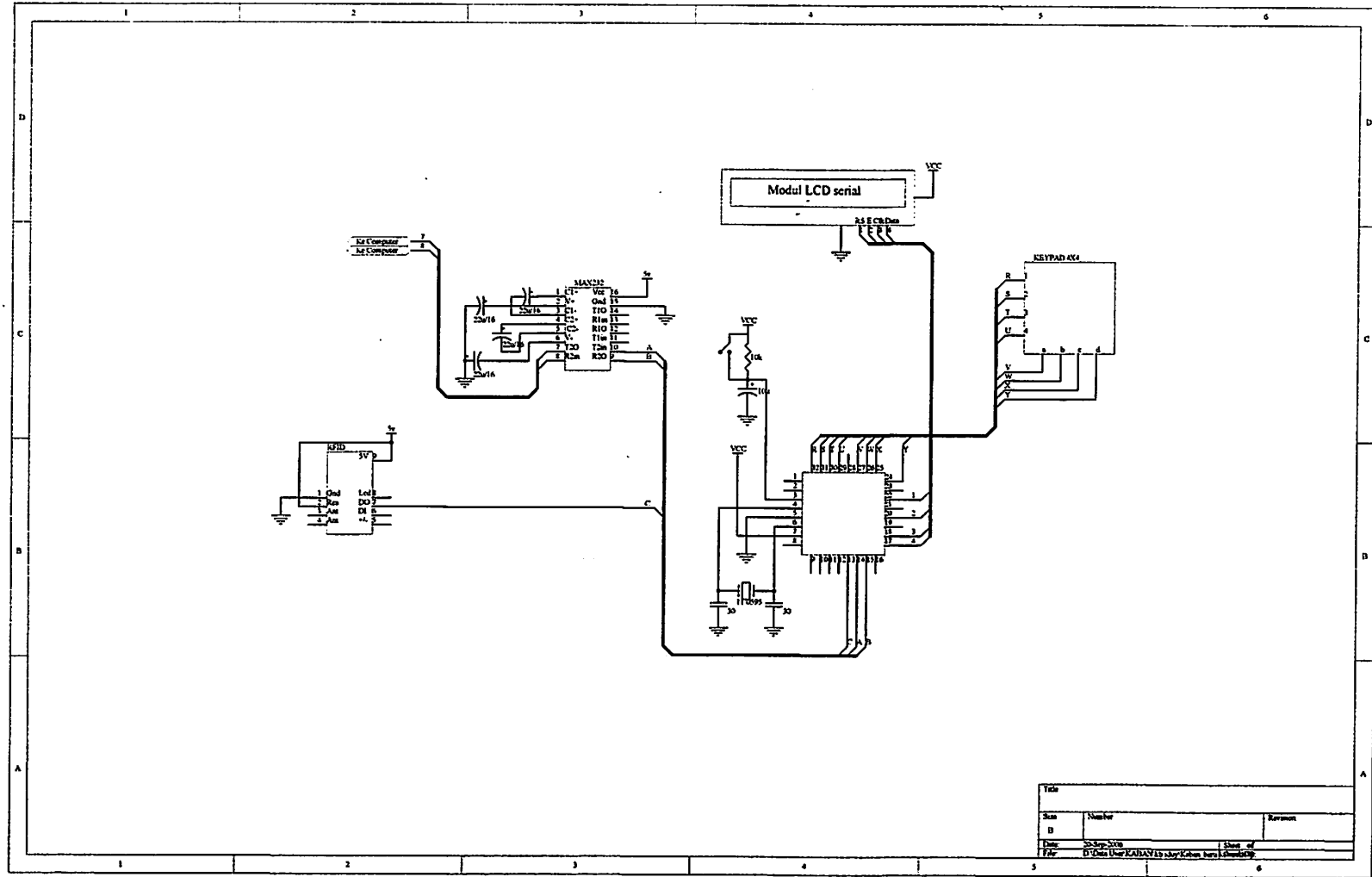
Dosen Pembimbing


(Ir. F. Yudi Limpraptono, MT)

LEMPER

GAMBAR ALAT





Tanda		
Skala	Number	Keterangan
B		
Date	20-Sep-2016	Sheet of
File	D:\Garis Ulang KAJIAN 13 sby\Kaban baru\skema.docx	

LISTING PROGRAM DELPHI 7

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, Menus, ExtCtrls, Grids, DBGrids, DB, ADODB, CPort,
  DBCtrls,
  StdCtrls;

type
  TForm1 = class(TForm)
    Notebook1: TNotebook;
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    close1: TMenuItem;
    ADOConnection1: TADOConnection;
    ADOQuerycust: TADOQuery;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    ComPort1: TComPort;
    Edit1: TEdit;
    Edit2: TEdit;
    edcekid: TEdit;
    btcek: TButton;
    edpasscek: TEdit;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    LbId: TLabel;
    lbnama: TLabel;
    lbtgl: TLabel;
    lbwaktu: TLabel;
    lbtuj: TLabel;
    lbbiaya: TLabel;
    ADOQuerytransaksi: TADOQuery;
    DataSource2: TDataSource;
    DBGrid2: TDBGrid;
    DBNavigator2: TDBNavigator;
    btlihattransaksi: TButton;
    customer1: TMenuItem;
    lihattabel1: TMenuItem;
    transaksil: TMenuItem;
    lihattabel2: TMenuItem;
    tampilan1: TMenuItem;
    btlihatcust: TButton;
    Label7: TLabel;
    lbsisasaldo: TLabel;
    btprint: TButton;
    ADOConnection2: TADOConnection;
    ADOQuery1: TADOQuery;
    ADOConnection3: TADOConnection;
    ADOQuery2: TADOQuery;
  end;
end;
```

```

    lbciptat: TLabel;
    button: TImage;
    GroupBox2: TGroupBox;
    lbpassciptat: TLabel;
    Label8: TLabel;
    edbt: TEdit;
    buton: TButton;
    procedure kirimdata(data:string);
    procedure closelClick(Sender: TObject);
    procedure ComPort1RxChar(Sender: TObject; Count: Integer);
    procedure FormCreate(Sender: TObject);
    procedure btcekClick(Sender: TObject);
    procedure btlihattransaksiClick(Sender: TObject);
    procedure lihattabel1Click(Sender: TObject);
    procedure lihattabel2Click(Sender: TObject);
    procedure tampilan1Click(Sender: TObject);
    procedure btlihatcustClick(Sender: TObject);
    procedure btprintClick(Sender: TObject);
    procedure butonClick(Sender: TObject);
    procedure buttonClick(Sender: TObject);

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

var
    Form1: TForm1;
    dtserial, s1, waktu:string;
    i1, i2, bd:integer;

implementation

uses Math, Unit2, Unit6, Unit4;

{$R *.dfm}

procedure TForm1.kirimdata(data:string);
begin
    ComPort1.WriteStr(data);
    sleep(100);
end;

procedure TForm1.closelClick(Sender: TObject);
begin
    close;
end;

procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
//var
//iddes:string;
begin
    ComPort1.ReadStr(dtserial,Count);

    s1:=s1+dtserial;
    i1:=i1+Count;
    Edit1.Text:=s1;
    Edit2.Text:=Edit2.Text+' '+IntToStr(Count);

```

```

try
  if il=16 then
  begin
  //Edit1.Text:=s1;
  edcekid.Text:=IntToStr(StrToInt('$'+copy(s1,2,8)));
  edpasscek.Text:=copy(s1,11,6);
  il:=0;
  s1:='';
  bd:=1;
  btcek.Click;
  end;

  If il=1 then
  begin
  i2:=byte(dtserial[1]);
  //Edit1.Text:=dtserial;
  //Edit1.Text:=IntToStr(i2);
  il:=0;
  s1:='';
  btcek.Click;
  end;
except
end;

end;

procedure TForm1.FormCreate(Sender: TObject);
begin
Notebook1.ActivePage:='tampilan';
il:=0;
s1:='';
i2:=0;
end;

procedure TForm1.btcekClick(Sender: TObject);
var
ssaldo:string;
i,biaya,saldo,sby,jbr,bny,bgl,jkt,lwg,sdj,sgsr:integer;
begin
ADOQuerycust.Close;
ADOQuerycust.SQL.Clear;
ADOQuerycust.SQL.Add('select * from customer where
id='+QuotedStr(edcekid.Text));
ADOQuerycust.Open;

if
(edcekid.Text=ADOQuerycust.FieldByName('id').AsString)and(edcekid.Text<>'')and(edpasscek.Text=ADOQuerycust.FieldByName('passwd').AsString)
then
begin

  if bd=1 then
  begin
  kirimdata('A');
  sleep(1000);
  kirimdata('C');
  ssaldo:=ADOQuerycust.FieldByName('saldo').AsString;
  //Edit1.Text:='saya';
  //Edit1.Text:=IntToStr(length(ssaldo));
  for i := 1 to length(ssaldo) do

```

```

begin
    kirimdata(ssaldo[i]);
end;
kirimdata('Z');
sleep(2000);
kirimdata('D');
bd:=0;
end;
if i2<>0 then
begin
//Edit1.Text:=IntToStr(i2);
sby:=5000;
jbr:=10000;
bny:=25000;
bgl:=3500;
jkt:=60000;
lwg:=2500;
sdj:=4000;
sgsr:=1500;
waktu:=TimeToStr(Time);
//btprint.Click;
// Application.CreateForm(TForm2, Form2);
//Form2.QuickRepl.Preview;
// Form2.QuickRepl.Print;

//sleep(1000);
case i2 of
    1: begin
        if (ADOQuerycust.FieldByName('saldo').AsInteger>=sby) then
            begin
                LbId.Caption:=edcekid.Text;
                lbnama.Caption:=ADOQuerycust.FieldByName('nama').AsString;
                lbtgl.Caption:=DateToStr(date);
                lbwaktu.Caption:=waktu;
                lbtuj.Caption:='Surabaya';
                biaya:=sby;
                lbbiaya.Caption:=IntToStr(biaya);

                saldo:=ADOQuerycust.FieldByName('saldo').AsInteger-biaya;

                ADOQuerytransaksi.SQL.Clear;
                ADOQuerytransaksi.SQL.Add('insert into
transaksi(id,nama,tanggal,waktu,tujuan,biaya)values(:a,:b,:c,:d,:e,:f
)');
                ADOQuerytransaksi.Parameters.ParamByName('a').Value:=
edcekid.Text;
                ADOQuerytransaksi.Parameters.ParamByName('b').Value:=
lbnama.Caption;
                ADOQuerytransaksi.Parameters.ParamByName('c').Value:=
lbtgl.Caption;
                ADOQuerytransaksi.Parameters.ParamByName('d').Value:=
lbwaktu.Caption;
                ADOQuerytransaksi.Parameters.ParamByName('e').Value:=
lbtuj.Caption;
                ADOQuerytransaksi.Parameters.ParamByName('f').Value:=
lbbiaya.Caption;
                ADOQuerytransaksi.ExecSQL;

                ADOQuerycust.Close;
                ADOQuerycust.SQL.Clear;

```

```

        ADOQuerycust.SQL.Add('update customer set
saldo='+QuotedStr(IntToStr(saldo))+'where
id='+QuotedStr(edcekid.Text));
        ADOQuerycust.ExecSQL;

        edcekid.Text:='';
        edpasscek.Text:='';
        i2:=0;
        kirimdata('E');

    end
    else
    begin
        kirimdata('F');
        sleep(500);
        kirimdata('E');
    end;
end;
2: begin
    if (ADOQuerycust.FieldByName('saldo').AsInteger>=jbr) then
        begin

            LbId.Caption:=edcekid.Text;
            lbnama.Caption:=ADOQuerycust.FieldByName('nama').AsString;
            lbtgl.Caption:=DateToStr(date);
            lbwaktu.Caption:=waktu;
            lbtuj.Caption:='Jember';
            biaya:=jbr;
            lbbiaya.Caption:=IntToStr(biaya);

            saldo:=ADOQuerycust.FieldByName('saldo').AsInteger-biaya;
            ADOQuerytransaksi.SQL.Clear;
            ADOQuerytransaksi.SQL.Add('insert into
transaksi(id,nama,tanggal,waktu,tujuan,biaya)values(:a,:b,:c,:d,:e,:f
)');
            ADOQuerytransaksi.Parameters.ParamByName('a').Value:=
edcekid.Text;
            ADOQuerytransaksi.Parameters.ParamByName('b').Value:=
lbnama.Caption;
            ADOQuerytransaksi.Parameters.ParamByName('c').Value:=
lbtgl.Caption;
            ADOQuerytransaksi.Parameters.ParamByName('d').Value:=
lbwaktu.Caption;
            ADOQuerytransaksi.Parameters.ParamByName('e').Value:=
lbtuj.Caption;
            ADOQuerytransaksi.Parameters.ParamByName('f').Value:=
lbbiaya.Caption;
            ADOQuerytransaksi.ExecSQL;

            ADOQuerycust.Close;
            ADOQuerycust.SQL.Clear;
            ADOQuerycust.SQL.Add('update customer set
saldo='+QuotedStr(IntToStr(saldo))+'where
id='+QuotedStr(edcekid.Text));
            ADOQuerycust.ExecSQL;
            edcekid.Text:='';
            edpasscek.Text:='';
            i2:=0;
            kirimdata('E');
        end
    else

```

```

begin
    kirimdata('F');
    sleep(500);
    kirimdata('E');
end;
end;
3:begin
    if (ADOQuerycust.FieldByName('saldo').AsInteger>=bny) then
        begin

LbId.Caption:=edcekid.Text;
    lbnama.Caption:=ADOQuerycust.FieldByName('nama').AsString;
    lbtgl.Caption:=DateToStr(date);
    lbwaktu.Caption:=waktu;
    lbtuj.Caption:='Banyuwangi';
    biaya:=bny;
    lbbiaya.Caption:=IntToStr(biaya);

        saldo:=ADOQuerycust.FieldByName('saldo').AsInteger-biaya;
        ADOQuerytransaksi.SQL.Clear;
        ADOQuerytransaksi.SQL.Add('insert into
transaksi(id,nama,tanggal,waktu,tujuan,biaya)values(:a,:b,:c,:d,:e,:f
)');
        ADOQuerytransaksi.Parameters.ParamByName('a').Value:=
edcekid.Text;
        ADOQuerytransaksi.Parameters.ParamByName('b').Value:=
lbnama.Caption;
        ADOQuerytransaksi.Parameters.ParamByName('c').Value:=
lbtgl.Caption;
        ADOQuerytransaksi.Parameters.ParamByName('d').Value:=
lbwaktu.Caption;
        ADOQuerytransaksi.Parameters.ParamByName('e').Value:=
lbtuj.Caption;
        ADOQuerytransaksi.Parameters.ParamByName('f').Value:=
lbbiaya.Caption;
        ADOQuerytransaksi.ExecSQL;

        ADOQuerycust.Close;
        ADOQuerycust.SQL.Clear;
        ADOQuerycust.SQL.Add('update customer set
saldo='+QuotedStr(IntToStr(saldo))+ 'where
id='+QuotedStr(edcekid.Text));
        ADOQuerycust.ExecSQL;
        edcekid.Text:='';
        edpasscek.Text:='';
        i2:=0;
        kirimdata('E');
        end
    else
        begin
            kirimdata('F');
            sleep(500);
            kirimdata('E');
        end;
    :
end;
4:begin
    if (ADOQuerycust.FieldByName('saldo').AsInteger>=bgl) then
        begin

LbId.Caption:=edcekid.Text;
    lbnama.Caption:=ADOQuerycust.FieldByName('nama').AsString;

```

```

lbtgl.Caption:=DateToStr(date);
lbwaktu.Caption:=waktu;
lbtuj.Caption:='Bangil';
biaya:=bgl;
lbbiaya.Caption:=IntToStr(biaya);

saldo:=ADOQuerycust.FieldByName('saldo').AsInteger-biaya;
ADOQuerytransaksi.SQL.Clear;
ADOQuerytransaksi.SQL.Add('insert into
transaksi(id,nama,tanggal,waktu,tujuan,biaya)values(:a,:b,:c,:d,:e,:f
)');
ADOQuerytransaksi.Parameters.ParamByName('a').Value:=
edcekid.Text;
ADOQuerytransaksi.Parameters.ParamByName('b').Value:=
lbnama.Caption;
ADOQuerytransaksi.Parameters.ParamByName('c').Value:=
lbtgl.Caption;
ADOQuerytransaksi.Parameters.ParamByName('d').Value:=
lbwaktu.Caption;
ADOQuerytransaksi.Parameters.ParamByName('e').Value:=
lbtuj.Caption;
ADOQuerytransaksi.Parameters.ParamByName('f').Value:=
lbbiaya.Caption;
ADOQuerytransaksi.ExecSQL;

ADOQuerycust.Close;
ADOQuerycust.SQL.Clear;
ADOQuerycust.SQL.Add('update customer set
saldo='+QuotedStr(IntToStr(saldo))+ 'where
id='+QuotedStr(edcekid.Text));
ADOQuerycust.ExecSQL;
edcekid.Text:='';
edpasscek.Text:='';
i2:=0;
irimdata('E');
end
else
begin
irimdata('F');
sleep(500);
irimdata('E');
end;
end;
5:begin
if (ADOQuerycust.FieldByName('saldo').AsInteger>=jkt) then
begin

LbId.Caption:=edcekid.Text;
lbnama.Caption:=ADOQuerycust.FieldByName('nama').AsString;
lbtgl.Caption:=DateToStr(date);
lbwaktu.Caption:=waktu;
lbtuj.Caption:='Jakarta';
biaya:=jkt;
lbbiaya.Caption:=IntToStr(biaya);

saldo:=ADOQuerycust.FieldByName('saldo').AsInteger-biaya;
ADOQuerytransaksi.SQL.Clear;
ADOQuerytransaksi.SQL.Add('insert into
transaksi(id,nama,tanggal,waktu,tujuan,biaya)values(:a,:b,:c,:d,:e,:f
)');

```



```

        ADOQuerytransaksi.Parameters.ParamByName('a').Value:=
edcekid.Text;
        ADOQuerytransaksi.Parameters.ParamByName('b').Value:=
lbnama.Caption;
        ADOQuerytransaksi.Parameters.ParamByName('c').Value:=
lbtgl.Caption;
        ADOQuerytransaksi.Parameters.ParamByName('d').Value:=
lbwaktu.Caption;
        ADOQuerytransaksi.Parameters.ParamByName('e').Value:=
lbtuj.Caption;
        ADOQuerytransaksi.Parameters.ParamByName('f').Value:=
lbbiaya.Caption;
        ADOQuerytransaksi.ExecSQL;

        ADOQuerycust.Close;
        ADOQuerycust.SQL.Clear;
        ADOQuerycust.SQL.Add('update customer set
saldo='+QuotedStr(IntToStr(saldo))+ 'where
id='+QuotedStr(edcekid.Text));
        ADOQuerycust.ExecSQL;
        edcekid.Text:='';
        edpasscek.Text:='';
        i2:=0;
        kirimdata('E');
        end
        else
        begin
        kirimdata('F');
        sleep(500);
        kirimdata('E');
        end;
    end;
6:begin
        if (ADOQuerycust.FieldByName('saldo').AsInteger>=lwg) then
        begin

LbId.Caption:=edcekid.Text;
        lbnama.Caption:=ADOQuerycust.FieldByName('nama').AsString;
        lbtgl.Caption:=DateToStr(date);
        lbwaktu.Caption:=waktu;
        lbtuj.Caption:='LAWANG';
        biaya:=lwg;
        lbbiaya.Caption:=IntToStr(biaya);

        saldo:=ADOQuerycust.FieldByName('saldo').AsInteger-biaya;
        ADOQuerytransaksi.SQL.Clear;
        ADOQuerytransaksi.SQL.Add('insert into
transaksi(id,nama,tanggal,waktu,tujuan,biaya) values (:a,:b,:c,:d,:e,:f
)');
        ADOQuerytransaksi.Parameters.ParamByName('a').Value:=
edcekid.Text;
        ADOQuerytransaksi.Parameters.ParamByName('b').Value:=
lbnama.Caption;
        ADOQuerytransaksi.Parameters.ParamByName('c').Value:=
lbtgl.Caption;
        ADOQuerytransaksi.Parameters.ParamByName('d').Value:=
lbwaktu.Caption;
        ADOQuerytransaksi.Parameters.ParamByName('e').Value:=
lbtuj.Caption;
        ADOQuerytransaksi.Parameters.ParamByName('f').Value:=
lbbiaya.Caption;

```

```

ADOQuerytransaksi.ExecSQL;

ADOQuerycust.Close;
ADOQuerycust.SQL.Clear;
ADOQuerycust.SQL.Add('update customer set
saldo='+QuotedStr(IntToStr(saldo))+'where
id='+QuotedStr(edcekid.Text));
ADOQuerycust.ExecSQL;
edcekid.Text:='';
edpasscek.Text:='';
i2:=0;
kirimdata('E');
end
else
begin
kirimdata('F');
sleep(500);
kirimdata('E');
end;
end;
7:begin
if (ADOQuerycust.FieldByName('saldo').AsInteger>=sdj) then
begin
LbId.Caption:=edcekid.Text;
lbnama.Caption:=ADOQuerycust.FieldByName('nama').AsString;
lbtgl.Caption:=DateToStr(date);
lbwaktu.Caption:=waktu;
lbtuj.Caption:='SIDOARJO';
biaya:=sdj;
lbbiaya.Caption:=IntToStr(biaya);

saldo:=ADOQuerycust.FieldByName('saldo').AsInteger-biaya;
ADOQuerytransaksi.SQL.Clear;
ADOQuerytransaksi.SQL.Add('insert into
transaksi(id,nama,tanggal,waktu,tujuan,biaya)values(:a,:b,:c,:d,:e,:f
)');
ADOQuerytransaksi.Parameters.ParamByName('a').Value:=
edcekid.Text;
ADOQuerytransaksi.Parameters.ParamByName('b').Value:=
lbnama.Caption;
ADOQuerytransaksi.Parameters.ParamByName('c').Value:=
lbtgl.Caption;
ADOQuerytransaksi.Parameters.ParamByName('d').Value:=
lbwaktu.Caption;
ADOQuerytransaksi.Parameters.ParamByName('e').Value:=
lbtuj.Caption;
ADOQuerytransaksi.Parameters.ParamByName('f').Value:=
lbbiaya.Caption;
ADOQuerytransaksi.ExecSQL;

ADOQuerycust.Close;
ADOQuerycust.SQL.Clear;
ADOQuerycust.SQL.Add('update customer set
saldo='+QuotedStr(IntToStr(saldo))+'where
id: '+QuotedStr(edcekid.Text));
ADOQuerycust.ExecSQL;
edcekid.Text:='';
edpasscek.Text:='';
i2:=0;
kirimdata('E');
end

```

```

        else
        begin
        kirimdata('F');
        sleep(500);
        kirimdata('E');
        end;
    end;
8:begin
    if (ADOQuerycust.FieldByName('saldo').AsInteger>=sgsr) then
    begin
    LbId.Caption:=edcekid.Text;
    lbnama.Caption:=ADOQuerycust.FieldByName('nama').AsString;
    lbtgl.Caption:=DateToStr(date);
    lbwaktu.Caption:=waktu;
    lbtuj.Caption:='singosari';
    biaya:=sgsr;
    lbbiaya.Caption:=IntToStr(biaya);

    saldo:=ADOQuerycust.FieldByName('saldo').AsInteger-biaya;
    ADOQuerytransaksi.SQL.Clear;
    ADOQuerytransaksi.SQL.Add('insert into
transaksi(id,nama,tanggal,waktu,tujuan,biaya)values(:a,:b,:c,:d,:e,:f
)');
        ADOQuerytransaksi.Parameters.ParamByName('a').Value:=
edcekid.Text;
        ADOQuerytransaksi.Parameters.ParamByName('b').Value:=
lbnama.Caption;
        ADOQuerytransaksi.Parameters.ParamByName('c').Value:=
lbtgl.Caption;
        ADOQuerytransaksi.Parameters.ParamByName('d').Value:=
lbwaktu.Caption;
        ADOQuerytransaksi.Parameters.ParamByName('e').Value:=
lbtuj.Caption;
        ADOQuerytransaksi.Parameters.ParamByName('f').Value:=
lbbiaya.Caption;
        ADOQuerytransaksi.ExecSQL;

        ADOQuerycust.Close;
        ADOQuerycust.SQL.Clear;
        ADOQuerycust.SQL.Add('update customer set
saldo='+QuotedStr(IntToStr(saldo))+ 'where
id='+QuotedStr(edcekid.Text));
        ADOQuerycust.ExecSQL;
        edcekid.Text:='';
        edpasscek.Text:='';
        i2:=0;
        kirimdata('E');
        end
    else
    begin
    kirimdata('F');
    sleep(500);
    kirimdata('E');
    end;
    end;
    lbsisasaldo.Caption:=IntToStr(saldo);
    btprint.Click;
    //edcekid.Text:='';
    // edpasscek.Text:='';
end;

```

```

end
else
if (edcekid.Text=ADOQuerycust.FieldByName('id').AsString) and (edcekid.Text<>'') and (edpasscek.Text<>ADOQuerycust.FieldByName('passwd').AsString) then
begin
    kirimdata('B');
    edcekid.Text:='';
    edpasscek.Text:='';
end
else
if (edcekid.Text<>ADOQuerycust.FieldByName('id').AsString) and (edcekid.Text<>'') and (edpasscek.Text<>ADOQuerycust.FieldByName('passwd').AsString) then
begin
    kirimdata('B');
    edcekid.Text:='';
    edpasscek.Text:='';
end;
{else if (edcekid.Text<>ADOQuerycust.FieldByName('id').AsString) then
begin
    kirimdata('B');
end;
}
//edcekid.Text:='';
//edpasscek.Text:='';
end;

procedure TForm1.btlihattransaksiClick(Sender: TObject);
begin
ADOQuerytransaksi.Close;
ADOQuerytransaksi.SQL.Clear;
ADOQuerytransaksi.SQL.Add('select * from transaksi');
ADOQuerytransaksi.Open;
end;

procedure TForm1.lihattabel1Click(Sender: TObject);
begin
Notebook1.ActivePage:='customer';
end;

procedure TForm1.lihattabel2Click(Sender: TObject);
begin
Notebook1.ActivePage:='transaksi';
end;

procedure TForm1.tampilan1Click(Sender: TObject);
begin
Notebook1.ActivePage:='tampilan';
end;

procedure TForm1.btlihatcustClick(Sender: TObject);
begin
    ADOQuerycust.Close;
    ADOQuerycust.SQL.Clear;
    ADOQuerycust.SQL.Add('select * from customer');
    ADOQuerycust.Open;
end;

procedure TForm1.btprintClick(Sender: TObject);

```

```
begin
Application.CreateForm(TForm2, Form2);
  //Form2.QuickRepl.Preview;
  Form2.QuickRepl.Print;
end;

procedure TForm1.butonClick(Sender: TObject);
begin
DM6.g;

end;

procedure TForm1.buttonClick(Sender: TObject);
begin
DM4.a;
end;

end.
```

```

/*****
/*
/* FILE      :kaban.c
/* DATE      :Wed, Aug 09, 2006
/* DESCRIPTION :Main Program
/* CPU TYPE   :Other
/*
/* This file is generated by Renesas Project Generator (Ver.4.0).
/*
/*****
#include <sfr_r813.h>
#include "lcd.h"
#include "keypad.h"
#include "usart.h"

#define max_list 9 //list maximum = max_list - 2
#define control p0_2

char mode, status;
char rf_data[16];
char pass_data[6];
char rx_buff;
unsigned char counter, counter_p, pointer;

#pragma INTERRUPT rx_int
void rx_int(void)
{
    /*if(u0rb == 0x02)
    {
        mode = 1; //rfid mode
        counter = 0;
    }else if(u0rb == 0x05)
    {
        mode = 0; //comp mode
    }*/
    if(mode == 1)
    {
        rf_data[counter] = u0rb;
        counter++;
    }else if(mode == 0)
    {
        rx_buff = u0rb;
        status = 1;
    }
}

void Delay(char lama);
void baca_pwd(void);
void baca_sisa(void);
void pilih_kota(void);
void kirim_pwd(void);

void main(void)
{
    - asm("FCLR I"); // Interrupt disable */
    prcr = 1; // Protect off */
    cml3 = 1; // X-in X-out */
    cml5 = 1; // XCIN-XCOUT drivecapacity select bit : HIGH */

```

```

cm05 = 0;
cm16 = 0;
cm17 = 0;
cm06 = 0;
asm("nop");
asm("nop");
asm("nop");
asm("nop");
oed2 = 0;
prcr = 0;
prc2 = 1;
pd0_2 = 1;
pd3_7 = 0;

control = 0;
initLCD();
initSerial();
init_keypad();

mode = 1;
status = 0;
rx_buff = 0;
counter = 0;
asm("FSET I");
 kirimLCD_perintah(0x01);
 kirimLCD_string(" KA Prabayar");
 while(1)
 {
     if(mode == 1)
     {
         while(counter < 16){;}
         asm("FCLR I"); //disable int

         //convert id
         kirimLCD_perintah(0x01);
         kirimLCD_perintah(0x81);
         kirimLCD_string("Maskkan pssword");
         kirimLCD_perintah(0xC1);
         baca_pwd();
         counter = 3;
         counter_p = 0;
         control = 1;
         Delay(0xFFFF);
         kirimSeri('I');
         Delay(0xFFFF);
         while(counter < 11)
         {
             kirimSeri(rf_data[counter]);
             Delay(0xFFFF);
             counter++;
         }
         kirimSeri('P');
         Delay(0xFFFF);
         kirim_pwd();
         kirimLCD_perintah(0x01);
     }
 }

```

```

counter = 0;
mode = 0;
asm("FSET I");
}else if(mode == 0)
{
switch(rx_buff)
{
case 'A' : kirimLCD_perintah(0x01);
            kirimLCD_perintah(0x83);
            kirimLCD_string("Password ");
            kirimLCD_perintah(0xC3);
            kirimLCD_string("diterima ");
            Delay(0xFFFF);
            while (status == 0){;}
            status = 0;
            rx_buff = 0;
            break;
case 'B' : kirimLCD_perintah(0x01);
            mode = 1;
            Delay(0xFFFF);
            control = 0;
            kirimLCD_perintah(0x83);
            kirimLCD_string("Password ");
            kirimLCD_perintah(0xC3);
            kirimLCD_string("ditolak ");
            pm03 = 1;
            rx_buff = 0;
            break;
case 'C' : kirimLCD_perintah(0x01);
            kirimLCD_string("Sisa saldo");
            kirimLCD_perintah(0xC1);
            kirimLCD_string("Rp ");
            baca_sisa();
            rx_buff = 0;
            break;
case 'D' : pilih_kota();
            rx_buff = 0;
            break;
case 'E' : mode = 1;
            Delay(0xFFFF);
            control = 0;
            rx_buff = 0;
            pm03 = 1;
            kirimLCD_perintah(0x01);
            kirimLCD_perintah(0x81);
            kirimLCD_string(" KA Prabayar");
            break;
case 'F' : kirimLCD_perintah(0x01);
            kirimLCD_perintah(0x81);
            kirimLCD_string("Saldo tidak");
            kirimLCD_perintah(0xC3);
            kirimLCD_string("Mencukupi");
            rx_buff = 0;
            break;
}
}
}

```



```

}
void Delay(char lama)
{
    while (lama > 0)
    {
        lama--;    // lama=lama-1, ulangi terus sampai lama = 0
    }
}
void baca_pwd(void)
{
    counter_p = 0;
    while(counter_p < 6)
    {
        while(Tombolnya1() != ' ')
        {
            if((Tombolnya1() >= '0') && (Tombolnya1() <= '9'))
            {
                pass_data[counter_p] = Tombolnya1();
                kirimLCD_data('*');
                counter_p++;
            }
            while(Tombolnya1() != ' '){;}
        }
        while(Tombolnya1() != 'E');
    }
}
void kirim_pwd(void)
{
    counter_p = 0;
    while(counter_p < 6)
    {
        kirimSeri(pass_data[counter_p]);
        Delay(0xFFFF);
        counter_p++;
    }
}
void baca_sisa(void)
{
    //  ulc1 |= 0x04;
    //  while((ulc1 & 0x08) == 0x00 ){;}
    //  rx_buff = ulrb;    //pindahkan data

    while(rx_buff != 'Z')
    {
        status = 0;
        if((rx_buff >= 0x30) && (rx_buff <= 0x39))
        {
            kirimLCD_data(rx_buff);
        }
        Delay(0xFFFF);
        //  ulc1 |= 0x04;
        //  while((ulc1 & 0x08) == 0x00 ){;}
        while (status == 0){;}
        //  rx_buff = ulrb;    //pindahkan data
    }
}

```

```

}
void pilih_kota(void)
{
    kirimLCD_perintah(0x80);
    kirimLCD_string(" Kota Tujuan :");
    // kirimLCD_perintah(0xC1);
    pointer = 1;
    while(Tombolnyal() != 'E')
    {
        if(Tombolnyal() == 'U')
            pointer++;
        if(Tombolnyal() == 'D')
            pointer--;
        while(Tombolnyal() != ' '){;}
        if(pointer == 0)
            pointer = max_list - 1;
        if(pointer == max_list)
            pointer = 1;
        kirimLCD_perintah(0xC1);
        switch(pointer)
        {
            case 1 : //kirimLCD_perintah(0x01);
                    kirimLCD_string("Surabaya");
                    break;
            case 2 : //kirimLCD_perintah(0x01);
                    //kirimLCD_perintah(0xC1);
                    kirimLCD_string("Jember");
                    break;
            case 3 : //kirimLCD_perintah(0x01);
                    //kirimLCD_perintah(0xC1);
                    kirimLCD_string("Banyuwangi");
                    break;
            case 4 : //kirimLCD_perintah(0x01);
                    //kirimLCD_perintah(0xC1);
                    kirimLCD_string("Bangil");
                    break;
            case 5 : //kirimLCD_perintah(0x01);
                    //kirimLCD_perintah(0xC1);
                    kirimLCD_string("Jakarta");
                    break;
            case 6 : kirimLCD_string("Lawang");
                    break;
            case 7 : kirimLCD_string("Sidoarjo");
                    break;
            case 8 : kirimLCD_string("Singosari");
                    break;
        }
    }
    kirimSeri(pointer);;
}

```



+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

General Description

The MAX220/MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where $\pm 12V$ is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5 μ W. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

- Portable Computers
- Low-Power Modems
- Interface Translation
- Battery-Powered RS-232 Systems
- Multi-Drop RS-232 Networks

Features

Superior to Bipolar

- ◆ Operate from Single +5V Power Supply (+5V and +12V—MAX231/MAX239)
- ◆ Low-Power Receive Mode in Shutdown (MAX223/MAX242)
- ◆ Meet All EIA/TIA-232E and V.28 Specifications
- ◆ Multiple Drivers and Receivers
- ◆ 3-State Driver and Receiver Outputs
- ◆ Open-Line Detection (MAX243)

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering information continued at end of data sheet.

*Contact factory for dice specifications.

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (nF)	SHDN & Three-State	Rx Active In SHDN	Data Rate (kbps)	Features
MAX220	+5V	1	0	1.0	Yes	Yes	1.5	Differential output, industry standard output
MAX221	+5V	1	0	1.0	Yes	Yes	2.0	Low-power shutdown
MAX222	+5V	1	0	1.0	Yes	Yes	1.5	MAX221 and receivers active in shutdown
MAX223	+5V	1	0	1.0	Yes	Yes	1.5	Open-line detect
MAX224	+5V	1	0	1.0	Yes	Yes	1.5	3-state drivers with shutdown
MAX225	+5V	1	0	1.0	Yes	Yes	1.5	Standard $\pm 5V$ battery supplies
MAX226	+5V	1	0	1.0	Yes	Yes	1.5	Same functions as MAX225
MAX227	+5V	1	0	1.0	Yes	Yes	1.5	Industry standard
MAX228	+5V	1	0	1.0	Yes	Yes	2.0	High slew rate, small caps
MAX229	+5V	1	0	1.0	Yes	Yes	1.5	Hexadecimal caps
MAX230	+5V	1	0	1.0	Yes	Yes	2.0	Hexadecimal caps, high slew rate
MAX231	+5V	1	0	1.0	Yes	Yes	1.5	Replaces T162
MAX232	+5V	1	0	1.0	Yes	Yes	1.5	Hexadecimal caps
MAX233	+5V	1	0	1.0	Yes	Yes	1.5	Shutdown, three-state
MAX234	+5V	1	0	1.0	Yes	Yes	1.5	Complements IBM PC [™] serial port
MAX235	+5V	1	0	1.0	Yes	Yes	1.5	Replaces T162 and T162
MAX236	+5V	1	0	1.0	Yes	Yes	1.5	Standard $\pm 5V$ battery supplies
MAX237	+5V	1	0	1.0	Yes	Yes	1.5	Single package solution for IBM PC [™] serial port
MAX238	+5V	1	0	1.0	Yes	Yes	1.5	DIP or flipchip package
MAX239	+5V	1	0	1.0	Yes	Yes	1.5	Complete IBM PC [™] serial port
MAX240	+5V	1	0	1.0	Yes	Yes	2.0	Separate shutdown and enable
MAX241	+5V	1	0	1.0	Yes	Yes	2.0	Open-line detection simplifies cabling
MAX242	+5V	1	0	1.0	Yes	Yes	1.5	High slew rate
MAX243	+5V	1	0	1.0	Yes	Yes	1.5	High slew rate, caps, two shutdown modes
MAX244	+5V	1	0	1.0	Yes	Yes	1.5	High slew rate, caps, three shutdown modes
MAX245	+5V	1	0	1.0	Yes	Yes	1.5	High slew rate, caps, nine operating modes
MAX246	+5V	1	0	1.0	Yes	Yes	1.5	High slew rate, subpackage half chip enables
MAX247	+5V	1	0	1.0	Yes	Yes	1.5	Open-line, pad flipchip package



Maxim Integrated Products 1

For free samples & the latest literature: <http://www.maxim-ic.com>, or phone 1-800-998-8800. For small orders, phone 408-737-7600 ext. 3468.

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V_{CC})	-0.3V to +6V	15-Pin Narrow SO (derate 8.70mW/°C above +70°C) ... 696mW
Input Voltages		15-Pin Wide SO (derate 9.52mW/°C above +70°C) ... 762mW
V_{IH}	-0.3V to ($V_{CC} + 0.3$ V)	18-Pin Wide SO (derate 9.52mW/°C above +70°C) ... 762mW
V_{IL}	± 30 V	20-Pin Wide SO (derate 10.00mW/°C above +70°C) ... 800mW
I_{IH} (Note 1)	± 15 V	20-Pin SSOP (derate 8.00mW/°C above +70°C) ... 640mW
Output Voltages		15-Pin CERDIP (derate 10.00mW/°C above +70°C) ... 800mW
V_{OH}	± 15 V	18-Pin CERDIP (derate 10.53mW/°C above +70°C) ... 842mW
V_{OL}	-0.3V to ($V_{CC} + 0.3$ V)	
Driver/Receiver Output Short-Circuited to GND	Continuous	
Continuous Power Dissipation ($T_A = +70^\circ\text{C}$)		Operating Temperature Ranges
15-Pin Plastic DIP (derate 10.93mW/°C above +70°C) ... 842mW		MAX2__AC__MAX2__C__
18-Pin Plastic DIP (derate 11.11mW/°C above +70°C) ... 889mW		MAX2__AE__MAX2__E__
20-Pin Plastic DIP (derate 8.00mW/°C above +70°C) ... 440mW		MAX2__AM__MAX2__M__
		Storage Temperature Range
		Lead Temperature (soldering, 10sec)

Note 1: Input voltage measured with I_{IH} in high-impedance state. $\overline{\text{SHDN}}$ or $V_{CC} = 0\text{V}$.

Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device is not guaranteed beyond these ratings. Exposure to absolute maximum conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

($V_{CC} = +5\text{V} \pm 10\%$, C1-C4 = 0.1 μF , $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Output Voltage Swing	All transmitter outputs loaded with 3k Ω to GND		± 5	± 8		V
Input Logic Threshold Low				1.4	0.8	V
Input Logic Threshold High			2	1.4		V
Logic Full-Output Current	Normal operation			5	40	μA
	$\overline{\text{SHDN}} = 0\text{V}$, MAX222/242, shutdown		± 0.01		± 1	
Output Leakage Current	$V_{CC} = 5.5\text{V}$, $\overline{\text{SHDN}} = 0\text{V}$, $V_{IH} = \pm 15\text{V}$, MAX222/242		± 0.01		± 10	μA
	$V_{CC} = \overline{\text{SHDN}} = 0\text{V}$, $V_{IH} = \pm 15\text{V}$		± 0.01		± 10	
Data Rate	All except MAX220, normal operation			200	116	kbts/ sec
	MAX220			22	20	
Transmitter Output Resistance	$V_{CC} = V_+ = V_- = 0\text{V}$, $V_{IH} = \pm 2\text{V}$		300	10M		Ω
Output Short-Circuit Current	$V_{IH} = 0\text{V}$		± 7	± 22		mA
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					± 30	V
RS-232 Input Threshold Low	$V_{CC} = 5\text{V}$	All except MAX243 R2IN	0.8	1.3		V
		MAX243 R2IN (Note 2)	-3			
RS-232 Input Threshold High	$V_{CC} = 5\text{V}$	All except MAX243 R2IN		1.8	2.4	V
		MAX243 R2IN (Note 2)		-0.5	-0.1	
RS-232 Input Hysteresis	All except MAX243, $V_{CC} = 5\text{V}$, no hysteresis in shdn.		0.2	0.5	1	V
	MAX243			1		
RS-232 Input Resistance			3	5	7	k Ω
TTL-CMOS Output Voltage Low	$I_{OH} = 3.2\text{mA}$			0.2	0.4	V
TTL-CMOS Output Voltage High	$I_{OH} = -1.0\text{mA}$		3.5	$V_{CC} - 0.2$		V
TTL-CMOS Output Short-Circuit Current	Sourcing $V_{OH} = \text{GND}$		-2	-10		mA
	Sinking $V_{OH} = V_{CC}$		10	30		
TTL-CMOS Output Leakage Current	$\overline{\text{SHDN}} = V_{IH}$ or $\overline{\text{EN}} = V_{IH}$ ($\overline{\text{SHDN}} = 0\text{V}$ for MAX222), $0\text{V} \leq V_{IH} \leq V_{CC}$		± 0.05		± 10	μA

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

V_{CC} = +5V, V_{EE} = 0V, I_{CC} = 1.4mA, I_Q = 1mA, t_{PLH}, t_{PHL}, t_{PL}, t_{PH}, unless otherwise noted.

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
EN input threshold voltage	MAX243			1.4	0.8	V
EN input threshold voltage	MAX243		2.0	1.4		V
Supply voltage			4.5		5.5	V
Maximum sink current (I _{OL}) V _{OL} = 0.4V, V _{CC} = 5V, I _{CC} = 1mA	MAX220	MAX220		0.5	2	mA
		MAX220/232A/233A/242/243		4	10	
	MAX220	MAX220		12		
		MAX220/232A/233A/242/243		15		
Maximum source current (I _{OH}) V _{OH} = 4.5V, V _{CC} = 5V, I _{CC} = 1mA	MAX220/243	I _A = +20°C		0.1	10	μA
		I _A = 0°C to +10°C		2	50	
		I _A = +40°C to +85°C		2	50	
		I _A = -20°C to +10°C		30	100	
MAX220/243 output voltage	MAX220/243				±1	μA
MAX220/243 output voltage	MAX220/243			1.4	0.8	V
MAX220/243 output voltage	MAX220/243		2.0	1.4		V
Propagation delay t _{PLH} , t _{PHL} V _{OL} = 0.4V, V _{OH} = 4.5V R _L = 10kΩ, C _L = 50pF	MAX220	MAX220/232A/233A/242/243	0	12	30	ns
		MAX220	1.5	3	30	
Propagation delay t _{PL} , t _{PH} V _{OL} = 0.4V, V _{OH} = 4.5V R _L = 10kΩ, C _L = 50pF	MAX220	MAX220/232A/233A/242/243		1.5	3.5	ns
		MAX220		4	10	
	MAX220	MAX220/232A/233A/242/243		1.5	3.5	
		MAX220		5	10	
Propagation delay t _{PL} , t _{PH} V _{OL} = 0.4V, V _{OH} = 4.5V R _L = 10kΩ, C _L = 50pF	MAX220	MAX220/232A/233A/242/243		0.5	1	ns
		MAX220		0.5	3	
	MAX220	MAX220/232A/233A/242/243		0.5	1	
		MAX220		0.5	3	
Propagation delay t _{PL} , t _{PH} V _{OL} = 0.4V, V _{OH} = 4.5V R _L = 10kΩ, C _L = 50pF	MAX243	MAX243		0.5	10	ns
		MAX243		2.5	10	
Propagation delay t _{PL} , t _{PH} V _{OL} = 0.4V, V _{OH} = 4.5V R _L = 10kΩ, C _L = 50pF	MAX243	MAX243		125	500	ns
Propagation delay t _{PL} , t _{PH} V _{OL} = 0.4V, V _{OH} = 4.5V R _L = 10kΩ, C _L = 50pF	MAX243	MAX243		150	500	ns
Propagation delay t _{PL} , t _{PH} V _{OL} = 0.4V, V _{OH} = 4.5V R _L = 10kΩ, C _L = 50pF	MAX243	MAX220/242, 0.1μF caps in both channels (at 25°C)		200		ns
Propagation delay t _{PL} , t _{PH} V _{OL} = 0.4V, V _{OH} = 4.5V R _L = 10kΩ, C _L = 50pF	MAX243	MAX220/242, 0.1μF caps		500		ns
Propagation delay t _{PL} , t _{PH} V _{OL} = 0.4V, V _{OH} = 4.5V R _L = 10kΩ, C _L = 50pF	MAX220/243	MAX220/232A/233A/242/243		300		ns
		MAX220		2000		
Propagation delay t _{PL} , t _{PH} V _{OL} = 0.4V, V _{OH} = 4.5V R _L = 10kΩ, C _L = 50pF	MAX220/243	MAX220/232A/233A/242/243		100		ns
		MAX220		225		

Note 2: MAX243 output current is limited when R_L is 10kΩ or less.

+5V Powered Dual RS-232 Transmitter/Receiver

December 1993

Features

- Meets All RS-232C Specifications
- Requires Only Single +5V Power Supply
- Onboard Voltage Doubler/Inverter
- Low Power Consumption
- 2 Drivers
 - :9V Output Swing for +5V Input
 - 300 Ω Power-off Source Impedance
 - Output Current Limiting
 - TTL/CMOS Compatible
 - 30V/ μ s Maximum Slew Rate
- 2 Receivers
 - :30V Input Voltage Range
 - 3k Ω to 7k Ω Input Impedance
 - 0.5V Hysteresis to Improve Noise Rejection
- All Critical Parameters are Guaranteed Over the Entire Commercial, Industrial and Military Temperature Ranges

Applications

- Any System Requiring RS-232 Communications Port
 - Computer - Portable and Mainframe
 - Peripheral - Printers and Terminals
 - Portable Instrumentation
 - Modems
 - Dataloggers

Description

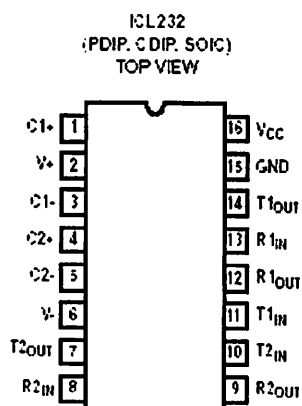
The ICL232 is a dual RS-232 transmitter/receiver interface circuit that meets all EIA RS-232C specifications. It requires a single +5V power supply, and features two onboard charge pump voltage converters which generate +10V and -10V supplies from the 5V supply.

The drivers feature true TTL/CMOS input compatibility, slew-rate-limited output, and 300 Ω power-off source impedance. The receivers can handle up to +30V, and have a 3k Ω to 7k Ω input impedance. The receivers also have hysteresis to improve noise rejection.

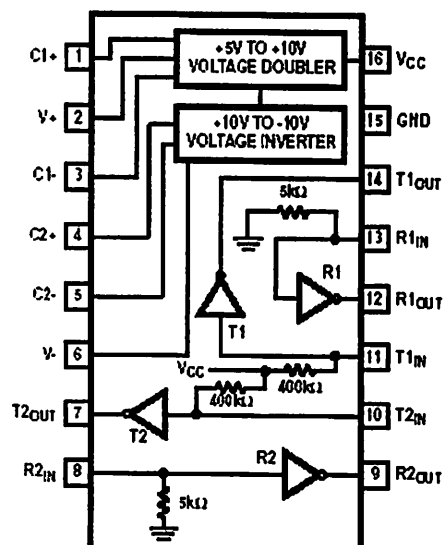
Ordering Information

PART NUMBER	TEMPERATURE RANGE	PACKAGE
ICL232CPE	0°C to +70°C	16 Lead Plastic DIP
ICL232CJE	0°C to +70°C	16 Lead Ceramic DIP
ICL232CBE	0°C to +70°C	16 Lead SOIC (W)
ICL232IPE	-40°C to +85°C	16 Lead Plastic DIP
ICL232IJE	-40°C to +85°C	16 Lead Ceramic DIP
ICL232IBE	-40°C to +85°C	16 Lead SOIC (W)
ICL232MJE	-55°C to +125°C	16 Lead Ceramic DIP

Pinouts



Functional Diagram



CAUTION These devices are sensitive to electrostatic discharge. Users should follow proper I.C. Handling Procedures.
Copyright © Harris Corporation 1993

File Number 3020.2

Specifications ICL232

Absolute Maximum Ratings

V_{CC} to Ground	(GND -0.3V) < V_{CC} < 6V
$V+$ to Ground	(V_{CC} -0.3V) < $V+$ < 12V
$V-$ to Ground	-12V < $V-$ < (GND +0.3V)
Input Voltages	
T_{1II} , T_{2II}	(V- -0.3V) < V_{II} < (V+ +0.3V)
R_{1II} , R_{2II}	$\pm 30V$
Output Voltages	
T_{1OII} , T_{2OII}	(V- -0.3V) < V_{IOUI} < (V+ +0.3V)
R_{1OII} , R_{2OII}	(GND -0.3V) < V_{RXOUI} < (V_{CC} +0.3V)
Short Circuit Duration	
T_{1OII} , T_{2OII}	Continuous
R_{1OII} , R_{2OII}	Continuous
Storage Temperature Range	-65°C to +150°C
Lead Temperature (Soldering 10s)	+300°C

Thermal Information

Thermal Resistance	θ_{JA}	θ_{JC}
Ceramic DIP Package	80°C/W	24°C/W
Plastic DIP Package	100°C/W	-
SOIC Package	100°C/W	-
Maximum Power Dissipation	.250mW	
Operating Temperature Range		
ICL232C	0°C to +70°C	
ICL232I	-40°C to +95°C	
ICL232M	-55°C to +125°C	

CAUTION: Stresses above those listed in Absolute Maximum Ratings may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

Electrical Specifications

Test Conditions: $V_{CC} = +5V \pm 10\%$, $T_A =$ Operating Temperature Range. Test Circuit as in Figure 8 Unless Otherwise Specified

PARAMETER	TEST CONDITIONS	LIMITS			UNITS
		MIN	TYP	MAX	
Transmitter Output Voltage Swing, T_{OII}	T_{1OUI} and T_{2OUI} loaded with 3k Ω to Ground	± 5	± 9	± 10	V
Power Supply Current, I_{CC}	Outputs Unloaded, $T_A = +25^\circ C$	-	5	10	mA
T_{II} Input Logic Low, V_{II}		-	-	0.8	V
T_{II} Input Logic High, V_{II}		2.0	-	-	V
Logic Pullup Current, I_p	T_{1II} , $T_{2II} = 0V$	-	15	200	μA
RS-232 Input Voltage Range, V_{II}		-30	-	+30	V
Receiver Input Impedance, R_{II}	$V_{II} = \pm 3V$	3.0	5.0	7.0	k Ω
Receiver Input Low Threshold, V_{II} (H-L)	$V_{CC} = 5.0V$, $T_A = +25^\circ C$	0.8	1.2	-	V
Receiver Input High Threshold, V_{II} (L-H)	$V_{CC} = 5.0V$, $T_A = +25^\circ C$	-	1.7	2.4	V
Receiver Input Hysteresis, V_{IYSI}		0.2	0.5	1.0	V
TTL/CMOS Receiver Output Voltage Low, V_{OI}	$I_{CUI} = 3.2mA$	-	0.1	0.4	V
TTL/CMOS Receiver Output Voltage High, V_{OH}	$I_{CUI} = -1.0mA$	3.5	4.6	-	V
Propagation Delay, t_{PD}	RS-232 to TTL	-	0.5	-	μs
Instantaneous Slew Rate, SR	$C_L = 10pF$, $R_L = 3k\Omega$, $T_A = +25^\circ C$ (Notes 1, 2)	-	-	30	V/ μs
Transition Region Slew Rate, SR_I	$R_L = 3k\Omega$, $C_L = 2500pF$ Measured from +3V to -3V or -3V to +3V	-	3	-	V/ μs
Output Resistance, $R_{(OII)}$	$V_{CC} = V+ = V- = 0V$, $V_{OUI} = \pm 2V$	300	-	-	Ω
RS-232 Output Short Circuit Current, I_{SC}	T_{1OUI} or T_{2OUI} shorted to GND	-	± 10	-	mA

NOTES:

1. Guaranteed by design.
2. See Figure 4 for definition.

ICL232

Typical Performance Curves

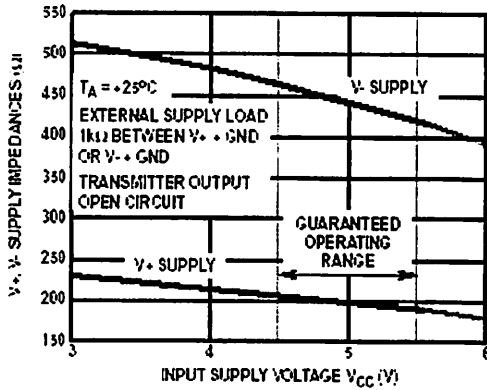


FIGURE 1. V+, V- OUTPUT IMPEDANCES vs V_{CC}

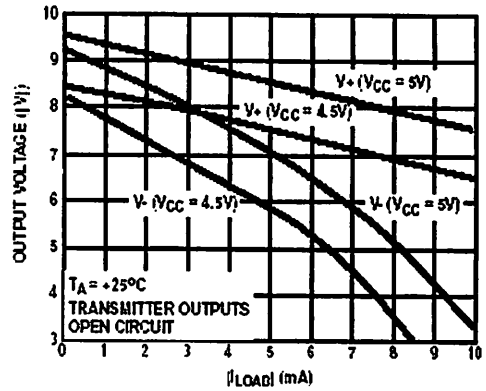


FIGURE 2. V+, V- OUTPUT VOLTAGES vs LOAD CURRENT

Pin Descriptions

PLASTIC DIP, CERAMIC DIP	SOIC	PIN NAME	DESCRIPTION
1	1	C1+	External capacitor "+" for internal voltage doubler.
2	2	V+	Internally generated +10V (typical) supply.
3	3	C1-	External capacitor "-" for internal voltage doubler.
4	4	C2+	External capacitor "+" internal voltage inverter.
5	5	C2-	External capacitor "-" internal voltage inverter.
6	6	V-	Internally generated -10V (typical) supply.
7	7	T2 _{OUT}	RS-232 Transmitter 2 output ±10V (typical).
8	8	R2 _{IN}	RS-232 Receiver 2 input, with internal 5K pulldown resistor to GND.
9	9	R2 _{OUT}	Receiver 2 TTL/CMOS output.
10	10	T2 _{IN}	Transmitter 2 TTL/CMOS input, with internal 400K pullup resistor to V _{CC} .
11	11	T1 _{IN}	Transmitter 1 TTL/CMOS input, with internal 400K pullup resistor to V _{CC} .
12	12	R1 _{OUT}	Receiver 1 TTL/CMOS output.
13	13	R1 _{IN}	RS-232 Receiver 1 input, with internal 5K pulldown resistor to GND.
14	14	T1 _{OUT}	RS-232 Transmitter 1 output ±10V (typical).
15	15	GND	Supply Ground.
16	16	VCC	Positive Power Supply +5V ± 10%

ICL232

Detailed Description

The ICL232 is a dual RS-232 transmitter/receiver powered by a single +5V power supply which meets all EIA RS232C specifications and features low power consumption. The functional diagram illustrates the major elements of the ICL232. The circuit is divided into three sections: a voltage doubler/inverter, dual transmitters, and dual receivers.

Voltage Converter

An equivalent circuit of the dual charge pump is illustrated in Figure 3.

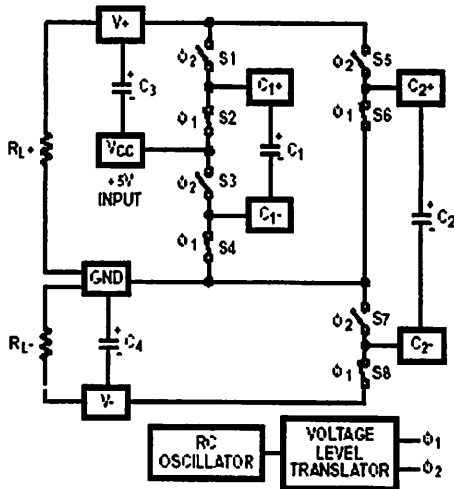


FIGURE 3. DUAL CHARGE PUMP

The voltage quadrupler contains two charge pumps which use two phases of an internally generated clock to generate +10V and -10V. The nominal clock frequency is 16kHz. During phase one of the clock, capacitor C1 is charged to V_{CC} . During phase two, the voltage on C1 is added to V_{CC} , producing a signal across C2 equal to twice V_{CC} . At the same time, C3 is also charged to $2V_{CC}$, and then during phase one, it is inverted with respect to ground to produce a signal across C4 equal to $-2V_{CC}$. The voltage converter accepts input voltages up to 5.5V. The output impedance of the doubler ($V+$) is approximately 200 Ω , and the output impedance of the inverter ($V-$) is approximately 450 Ω . Typical graphs are presented which show the voltage converters output vs input voltage and output voltages vs load characteristics. The test circuit (Figure 8) uses 1 μ F capacitors for C1-C4, however, the value is not critical. Increasing the values of C1 and C2 will lower the output impedance of the voltage doubler and inverter, and increasing the values of the reservoir capacitors, C3 and C4, lowers the ripple on the $V+$ and $V-$ supplies.

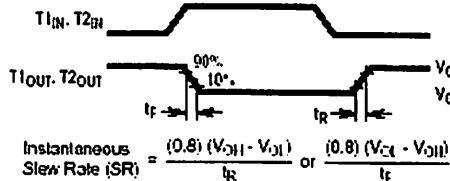


FIGURE 4. SLEW RATE DEFINITION

Transmitters

The transmitters are TTL/CMOS compatible inverters which translate the inputs to RS-232 outputs. The input logic threshold is about 26% of V_{CC} , or 1.3V for $V_{CC} = 5V$. A logic 1 at the input results in a voltage of between -5V and $V-$ at the output, and a logic 0 results in a voltage between +5V and ($V+ - 0.6V$). Each transmitter input has an internal 400k Ω pullup resistor so any unused input can be left unconnected and its output remains in its low state. The output voltage swing meets the RS-232 specification of $\pm 5V$ minimum with the worst case conditions of: both transmitters driving 3k Ω minimum load impedance, $V_{CC} = 4.5V$, and maximum allowable operating temperature. The transmitters have an internally limited output slew rate which is less than 30V/ μ s. The outputs are short circuit protected and can be shorted to ground indefinitely. The powered down output impedance is a minimum of 300 Ω with $\pm 2V$ applied to the outputs and $V_{CC} = 0V$.

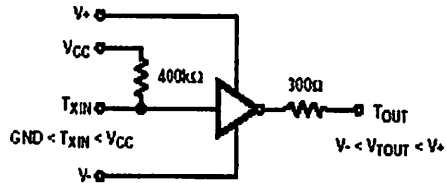


FIGURE 5. TRANSMITTER

Receivers

The receiver inputs accept up to $\pm 30V$ while presenting the required 3k Ω to 7k Ω input impedance even if the power is off ($V_{CC} = 0V$). The receivers have a typical input threshold of 1.3V which is within the $\pm 3V$ limits, known as the transition region, of the RS-232 specification. The receiver output is 0V to V_{CC} . The output will be low whenever the input is greater than 2.4V and high whenever the input is floating or driven between +0.8V and -30V. The receivers feature 0.5V hysteresis to improve noise rejection.

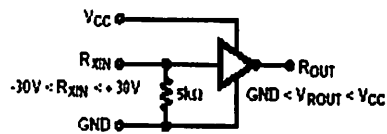


FIGURE 6. RECEIVER

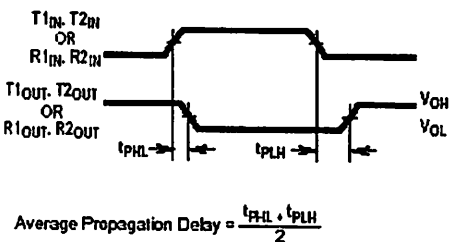


FIGURE 7. PROPAGATION DELAY DEFINITION

Test Circuits

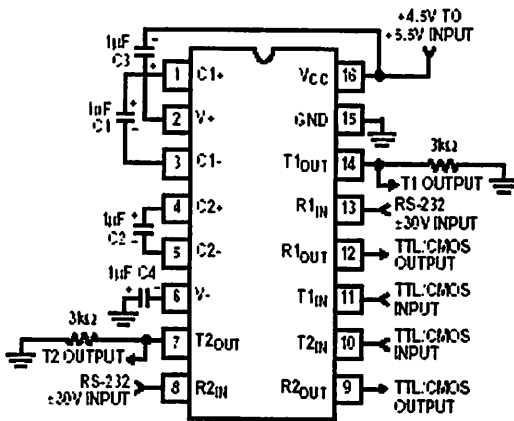


FIGURE 8. GENERAL TEST CIRCUIT

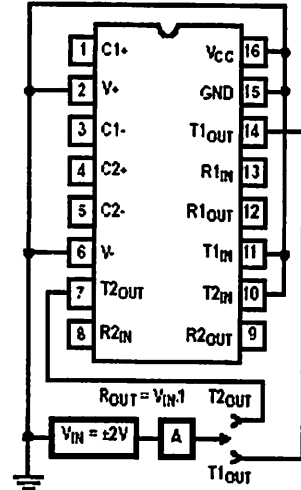


FIGURE 9. POWER-OFF SOURCE RESISTANCE CONFIGURATION

Applications

The ICL232 may be used for all RS-232 data terminal and communication links. It is particularly useful in applications where +12V power supplies are not available for conventional RS-232 interface circuits. The applications presented represent typical interface configurations.

A simple duplex RS-232 port with CTS/RTS handshaking is illustrated in Figure 10. Fixed output signals such as DTR (data terminal ready) and DSRs (data signaling rate select) is generated by driving them through a 5kΩ resistor connected to V+.

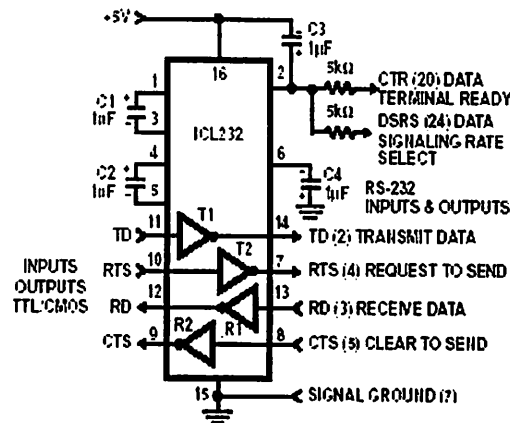


FIGURE 10. SIMPLE DUPLEX RS-232 PORT WITH CTS/RTS HANDSHAKING

In applications requiring four RS-232 inputs and outputs (Figure 11), note that each circuit requires two charge pump capacitors (C1 and C2) but can share common reservoir

capacitors (C3 and C4). The benefit of sharing common reservoir capacitors is the elimination of two capacitors and the reduction of the charge pump source impedance which effectively increases the output swing of the transmitters.

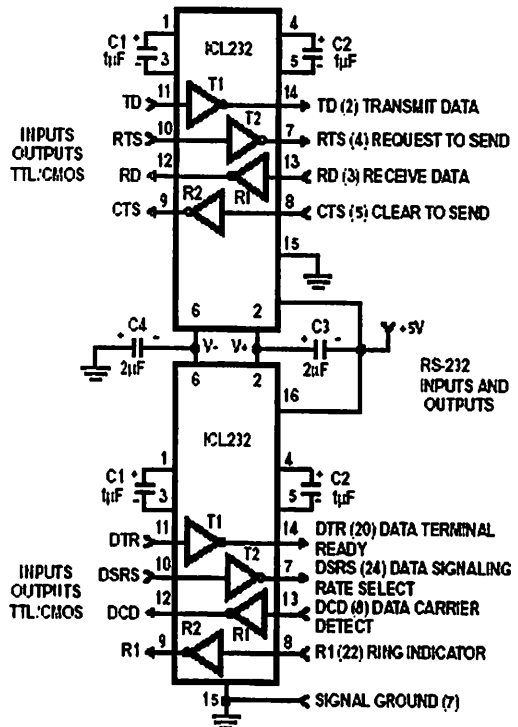


FIGURE 11. COMBINING TWO ICL232s FOR 4 PAIRS OF RS-232 INPUTS AND OUTPUTS

DALLAS
SEMICONDUCTOR

Application Note 83 Fundamentals of RS-232 Serial Communications

Due to its relative simplicity and low hardware overhead (as compared to parallel interfacing), serial communications is used extensively within the electronics industry. Today, the most popular serial communications standard in use is certainly the EIA/TIA-232-E specification. This standard, which has been developed by the Electronic Industry Association and the Telecommunications Industry Association (EIA/TIA), is more popularly referred to simply as "RS-232" where "RS" stands for "recommended standard". In recent years, this suffix has been replaced with "EIA/TIA" to help identify the source of the standard. This paper will use the common notation of "RS-232" in its discussion of the topic.

The official name of the EIA/TIA-232-E standard is "Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange". Although the name may sound intimidating, the standard is simply concerned with serial data communication between a host system (Data Terminal Equipment, or "DTE") and a peripheral system (Data Circuit-Terminating Equipment, or "DCE").

The EIA/TIA-232-E standard which was introduced in 1962 has been updated four times since its introduction in order to better meet the needs of serial communication applications. The letter "E" in the standard's name indicates that this is the fifth revision of the standard.

RS-232 SPECIFICATIONS

RS-232 is a "complete" standard. This means that the standard sets out to ensure compatibility between the host and peripheral systems by specifying 1) common voltage and signal levels, 2) common pin wiring configurations, and 3) a minimal amount of control information between the host and peripheral systems. Unlike many standards which simply specify the electrical characteristics of a given interface, RS-232 specifies electrical, functional, and mechanical characteristics in order to meet the above three criteria. Each of these aspects of the RS-232 standard is discussed below.

ELECTRICAL CHARACTERISTICS

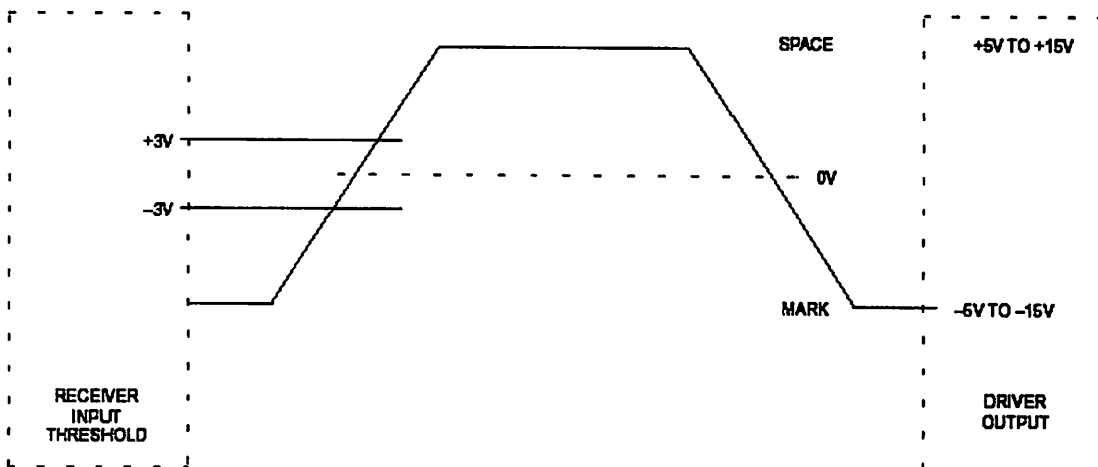
The electrical characteristics section of the RS-232 standard includes specifications on voltage levels, rate of change of signal levels, and line impedance.

The original RS-232 standard was defined in 1962. As this was before the days of TTL logic, it should not be surprising that the standard does not use 5 volt and ground logic levels. Instead, a high level for the driver output is defined as being +5 to +15 volts and a low level for the driver output is defined as being between -5 and -15 volts. The receiver logic levels were defined to provide a 2 volt noise margin. As such, a high level for the receiver is defined as +3 to +15 volts and a low level is -3 to -15 volts. Figure 1 illustrates the logic levels defined by the RS-232 standard. It is necessary to note that, for RS-232 communication, a low level (-3 to -15 volts) is defined as a logic 1 and is historically referred to as "marking". Likewise a high level (+3 to +15 volts) is defined as a logic 0 and is referred to as "spacing".

The RS-232 standard also limits the maximum slew rate at the driver output. This limitation was included to help reduce the likelihood of cross-talk between adjacent signals. The slower the rise and fall time, the smaller the chance of cross talk. With this in mind, the maximum slew rate allowed is 30 V/us. Additionally, a maximum data rate of 20k bits/second has been defined by the standard. Again with the purpose of reducing the chance of cross talk.

The impedance of the interface between the driver and receiver has also been defined. The load seen by the driver is specified to be 3k Ω to 7k Ω . For the original RS-232 standard, the cable between the driver and the receiver was also specified to be a maximum of 15 meters in length. This part of the standard was changed in revision "D" (EIA/TIA-232-D). Instead of specifying the maximum length of cable, a maximum capacitive load of 2500 pF was specified which is clearly a more adequate specification. The maximum cable length is determined by the capacitance per unit length of the cable which is provided in the cable specifications.

RS-232 LOGIC LEVEL SPECIFICATIONS Figure 1

**FUNCTIONAL CHARACTERISTICS**

Since RS-232 is a "complete" standard, it includes more than just specifications on electrical characteristics. The second aspect of operation that is covered by the standard concerns the functional characteristics of the interface. This essentially means that RS-232 has defined the function of the different signals that are used in the interface. These signals are divided into four different categories: common, data, control, and timing. Table 1 illustrates the signals that are defined by the RS-232 standard. As can be seen from the table there is an overwhelming number of signals defined by the standard. The standard provides an abundance of control signals and supports a primary and secondary communications channel. Fortunately few applications, if any, require all of these defined signals. For example, only eight signals are used for a typical modem. Some simple applications may require only four signals (two for data and two for handshaking) while others may require only data signals with no handshaking. Examples of how the RS-232 standard is used in some "real world" applications are discussed later in this paper. The complete list of defined signals is included here as a reference, but it is beyond the scope of this paper to review the functionality of all of these signals.

MECHANICAL INTERFACE CHARACTERISTICS

The third area covered by RS-232 concerns the mechanical interface. In particular, RS-232 specifies a 25-pin connector. This is the minimum connector size that can accommodate all of the signals defined in the functional portion of the standard. The pin assignment for this connector is shown in Figure 2. The connector for DCE equipment is male for the connector housing and female for the connection pins. Likewise, the DTE connector is a female housing with male connection pins. Although RS-232 specifies a 25-position connector, it should be noted that often this connector is not used. This is due to the fact that most applications do not require all of the defined signals and therefore a 25-pin connector is larger than necessary. This being the case, it is very common for other connector types to be used. Perhaps the most popular is the 9-position DB9S connector which is also illustrated in Figure 2. This connector provides the means to transmit and receive the necessary signals for modem applications, for example. This will be discussed in more detail later.

RS-232 DEFINED SIGNALS Table 1

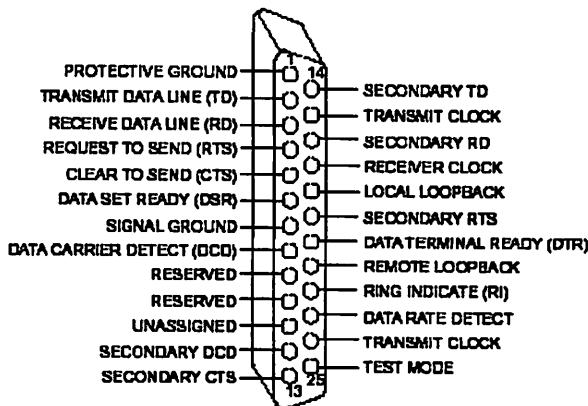
CIRCUIT MNEMONIC	CIRCUIT NAME*	CIRCUIT DIRECTION	CIRCUIT TYPE
AB	Signal Common	-	Common
BA BB	Transmitted Data (TD) Received Data (RD)	To DCE From DCE	Data
CA CB CC CD CE CF CG CH CI CJ RL LL TM	Request to Send (RTS) Clear to Send (CTS) DCE Ready (DSR) DTE Ready (DTR) Ring Indicator (RI) Received Line Signal Detector** (DCD) Signal Quality Detector Data Signal Rate Detector from DTE Data Signal Rate Detector from DCE Ready for Receiving Remote Loopback Local Loopback Test Mode	To DCE From DCE From DCE To DCE From DCE From DCE From DCE To DCE From DCE To DCE To DCE To DCE From DCE	Control
DA	Transmitter Signal Element Timing from DTE	To DCE	
DB DD	Transmitter Signal Element Timing from DCE Receiver Signal Element Timing From DCE	From DCE From DCE	Timing
SBA SBB	Secondary Transmitted Data Secondary Received Data	To DCE From DCE	Data
SCA SCB SCF	Secondary Request to Send Secondary Clear to Send Secondary Received Line Signal Detector	To DCE From DCE From DCE	Control

*Signals with abbreviations in parentheses are the eight most commonly used signals.

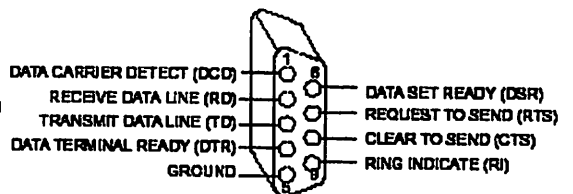
**This signal is more commonly referred to as Data Carrier Detect (DCD).

RS-232 CONNECTOR PIN ASSIGNMENTS Figure 2

25-PIN CONNECTOR



9-PIN CONNECTOR



PRACTICAL RS-232 IMPLEMENTATION

Most systems designed today do not operate using RS-232 voltage levels. Since this is the case, level conversion is necessary to implement RS-232 communication. Level conversion is performed by special RS-232 IC's. These IC's typically have line drivers that generate the voltage levels required by RS-232 and line receivers that can receive RS-232 voltage levels without being damaged. These line drivers and receivers typically invert the signal as well since a logic 1 is represented by a low voltage level for RS-232 communication and likewise a logic 0 is represented by a high logic level. Figure 3 illustrates the function of an RS-232 line driver/receiver in a typical modem application. In this particular example, the signals necessary for serial communication are generated and received by the Universal Asynchronous Receiver/Transmitter (UART). The RS-232 line driver/receiver IC performs the level translation necessary between the CMOS/TTL and RS-232 interface.

The UART just mentioned performs the "overhead" tasks necessary for asynchronous serial communication. For example, the asynchronous nature of this type of communication usually requires that start and stop bits be initiated by the host system to indicate to the peripheral system when communication will start and stop. Parity bits are also often employed to ensure that the data sent has not been corrupted. The UART usually generates the start, stop, and parity bits when transmitting data and can detect communication errors upon receiving data. The UART also functions as the intermediary between byte-wide (parallel) and bit-wide (serial) communication; it converts a byte of data into a

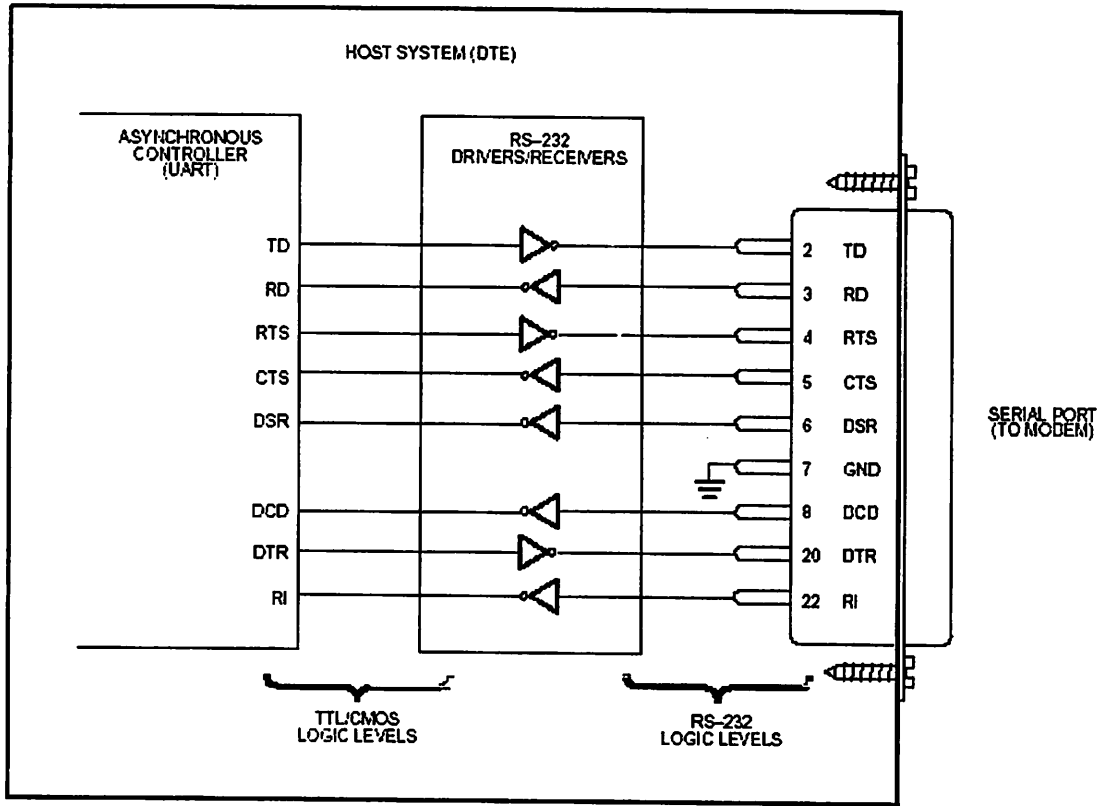
serial bit stream for transmitting and converts a serial bit stream into a byte of data when receiving.

Now that an elementary explanation of the TTL/CMOS to RS-232 interface has been provided we can consider some "real world" RS-232 applications. It has already been noted that RS-232 applications rarely follow the RS-232 standard precisely. Perhaps the most significant reason this is true is due to the fact that many of the defined signals are not necessary for most applications. As such, the unnecessary signals are omitted. Many applications, such as a modem, require only nine signals (two data signals, six control signals, and ground). Other applications may require only five signals (two for data, two for handshaking, and ground), while others may require only data signals with no handshake control. We will begin our investigation of "real world" implementations by first considering the typical modem application.

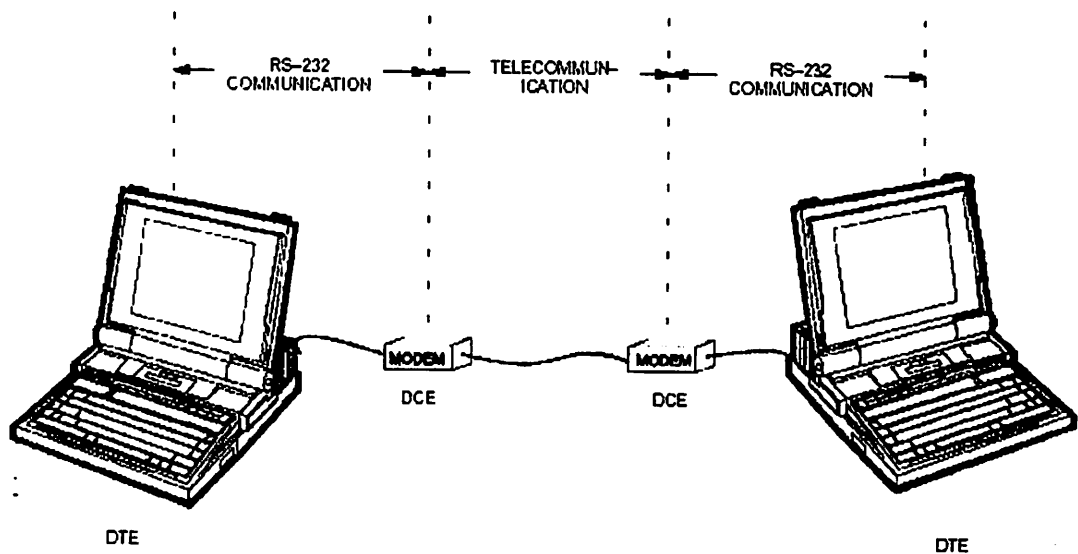
RS-232 IN MODEM APPLICATIONS

Modem applications are one of the most popular uses for the RS-232 standard. Figure 4 illustrates a typical modem application utilizing the RS-232 interface standard. As can be seen in the diagram, the PC is the DTE and the modem is the DCE. Communication between each PC and its associated modem is accomplished using the RS-232 standard. Communication between the two modems is accomplished via telecommunication. It should be noted that although a microcomputer is usually the DTE in RS-232 applications, this is not mandatory according to a strict interpretation of the standard.

TYPICAL RS-232 MODEM APPLICATION Figure 3



MODEM COMMUNICATION BETWEEN TWO PC'S Figure 4



Many modem applications require only nine signals (including ground). Although some designers choose to use a 25-pin connector, it is not necessary since there are only nine interface signals between the DTE and DCE. With this in mind, many have chosen to use 9- or 15-pin connectors (see Figure 2 for 9-pin connector pin assignment). The "basic nine" signals used in modem communication are illustrated in Figure 3. Note that with respect to the DTE, three RS-232 drivers and five receivers are necessary. The functionality of these signals is described below. Note that for the following signal descriptions, "ON" refers to a high RS-232 voltage level (+5 to +15 volts) and "OFF" refers to a low RS-232 voltage level (-5 to -15 volts). Keep in mind that a high RS-232 voltage level actually represents a logic 0 and a low RS-232 voltage level refers to a logic 1.

Transmitted Data (TD): One of two separate data signals. This signal is generated by the DTE and received by the DCE.

Received Data (RD): The second of two separate data signals. This signal is generated by the DCE and received by the DTE.

Request to Send (RTS): When the host system (DTE) is ready to transmit data to the peripheral system (DCE), RTS is turned ON. In simplex and duplex systems, this condition maintains the DCE in receive mode. In half-duplex systems, this condition maintains the DCE in receive mode and disables transmit mode. The OFF condition maintains the DCE in transmit mode. After RTS is asserted, the DCE must assert CTS before communication can commence.

Clear to Send (CTS): CTS is used along with RTS to provide handshaking between the DTE and the DCE. After the DCE sees an asserted RTS, it turns CTS ON when it is ready to begin communication.

Data Set Ready (DSR): This signal is turned on by the DCE to indicate that it is connected to the telecommunications line.

Data Carrier Detect (DCD): This signal is turned ON when the DCE is receiving a signal from a remote DCE which meets its suitable signal criteria. This signal remains ON as long as a suitable carrier signal can be detected.

Data Terminal Ready (DTR): DTR indicates the readiness of the DTE. This signal is turned ON by the DTE

when it is ready to transmit or receive data from the DCE. DTR must be ON before the DCE can assert DSR.

Ring Indicator (RI): RI, when asserted, indicates that a ringing signal is being received on the communications channel.

The signals described above form the basis for modem communication. Perhaps the best way to understand how these signals interact is to give a brief step by step example of a modem interfacing with a PC. The following steps describe a transaction in which a remote modem calls a local modem.

1. The local PC monitors the RI (Ring Indicator) signal via software.
2. When the remote modem wants to communicate with the local modem, it generates an RI signal. This signal is transferred by the local modem to the local PC.
3. The local PC responds to the RI signal by asserting the DTR (Data Terminal Ready) signal when it is ready to communicate.
4. After recognizing the asserted DTR signal, the modem responds by asserting DSR (Data Set Ready) after it is connected to the communications line. DSR indicates to the PC that the modem is ready to exchange further control signals with the DTE to commence communication. When DSR is asserted, the PC begins monitoring DCD for indication that data is being sent over the communication line.
5. The modem asserts DCD (Data Carrier Detect) after it has received a carrier signal from the remote modem that meets the suitable signal criteria.
6. At this point data transfer can begin. If the local modem has full-duplex capability, the CTS (Clear to Send) and RTS (Request to Send) signals are held in the asserted state. If the modem has only half-duplex capability, CTS and RTS provide the handshaking necessary for controlling the direction of the data flow. Data is transferred over the RD and TD signals.
7. When the transfer of data has been completed, the PC disables the DTR signal. The modem follows by inhibiting the DSR and DCD signals. At this point the PC and modem are in the original state described in step number 1.

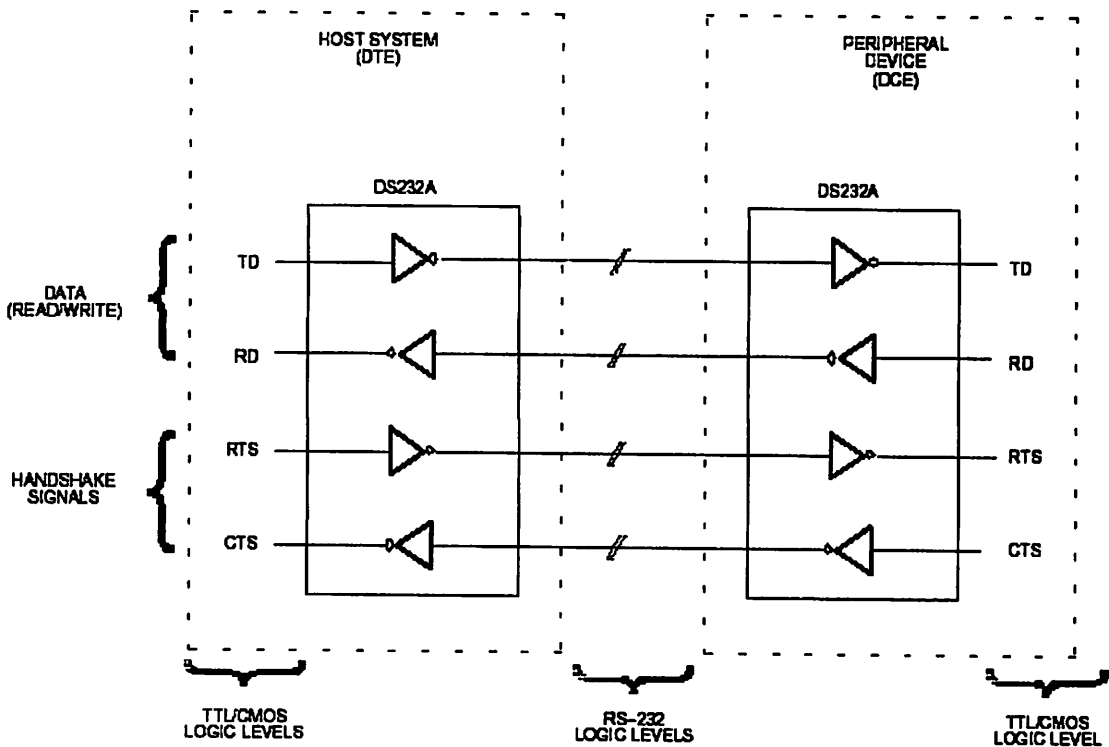
RS-232 IN MINIMAL HANDSHAKE APPLICATIONS

Even though the modem application discussed above is simplified from the RS-232 standard in terms of the number of signals needed, it is still more complex than the requirements of many systems. For many applications, two data lines and two handshake control lines are all that is necessary to establish and control communication between a host system and a peripheral system. For example, an environmental control system may need to interface with a thermostat using a half-duplex communication scheme. At times the control systems may desire to read the temperature from the thermostat and at other times may need to load temperature trip points to the thermostat. In this type of simple

application, five signals may be all that is necessary (two for data, two for handshake control, and ground).

Figure 5 illustrates a simple half-duplex communication interface. As can be seen in this diagram, data is transferred over the TD (Transmit Data) and RD (Receive Data) pins and handshake control is provided by the RTS (Ready to Send) and CTS (Clear to Send) pins. RTS is driven by the DTE to control the direction of data. When it is asserted, the DTE is placed in transmit mode. When RTS is inhibited, the DTE is placed in receive mode. CTS, which is generated by the DCE, controls the flow of data. When asserted, data can flow. However, when CTS is inhibited, the transfer of data is interrupted. The transmission of data is halted until CTS is reasserted.

HALF-DUPLEX COMMUNICATION SCHEME Figure 5



RS-232 APPLICATION LIMITATIONS

As mentioned earlier in this paper, the RS-232 standard was first introduced in 1962. In the more than three decades since, the electronics industry has changed immensely and therefore there are some limitations in the RS-232 standard. One limitation, the fact that over twenty signals have been defined by the standard, has already been addressed – simply do not use all of the signals or the 25-pin connector if they are not necessary. Other limitations in the standard are not necessarily as easy to correct, however.

GENERATION OF RS-232 VOLTAGE LEVELS

As we saw in the section on RS-232 electrical characteristics, RS-232 does not use the conventional 0 and 5 volt levels implemented in TTL and CMOS designs. Drivers have to supply +5 to +15 volts for a logic 0 and -5 to -15 volts for a logic 1. This means that extra power supplies are needed to drive the RS-232 voltage levels. Typically, a +12 volt and a -12 volt power supply are used to drive the RS-232 outputs. This is a great inconvenience for systems that have no other requirements for these power supplies. With this in mind, RS-232 products manufactured by Dallas Semiconductor have on-chip charge-pump circuits that generate the necessary voltage levels for RS-232 communication. The first charge pump essentially doubles the standard +5 volt power supply to provide the voltage level necessary for driving a logic 0. A second charge pump, inverts this voltage and provides the voltage level necessary for driving a logic 1. These two charge pumps allow the RS-232 interface products to operate from a single +5 volt supply.

MAXIMUM DATA RATE

Another limitation in the RS-232 standard is the maximum data rate. The standard defines a maximum data

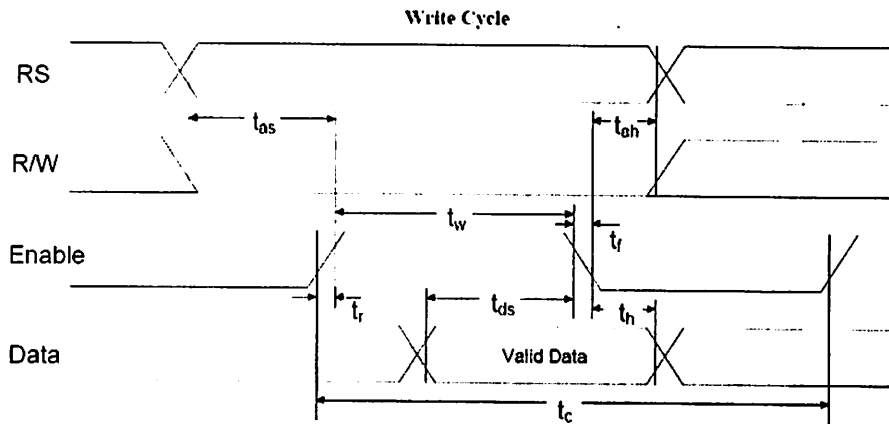
rate of 20k bits/second. This is unnecessarily slow for many of today's applications. RS-232 products manufactured by Dallas Semiconductor guarantee up to 250k bits/second and typically can communicate up to 350k bits/second. While providing a communication rate at this frequency, the devices still maintain a maximum 30V/μs maximum slew rate to reduce the likelihood of cross-talk between adjacent signals.

MAXIMUM CABLE LENGTH

A final limitation to discuss concerning RS-232 communication is cable length. As we have already seen, the cable length specification that was once included in the RS-232 standard has been replaced by a maximum load capacitance specification of 2500 pF. To determine the total length of cable allowed, one must determine the total line capacitance. Figure 6 shows a simple approximation for the total line capacitance of a conductor. As can be seen in the diagram, the total capacitance is approximated by the sum of the mutual capacitance between the signal conductors and the conductor-to-shield capacitance (or stray capacitance in the case of unshielded cable).

As an example, let's assume that the user has decided to use non-shielded cable when interconnecting the equipment. The cable mutual capacitance (C_m) of the cable is found in the cable's specifications to be 20 pF per foot. If we assume that the input capacitance of the receiver is 20 pF, this leaves the user with 2480 pF for the interconnecting cable. From the equation in Figure 6, the total capacitance per foot is found to be 30 pF. Dividing 2480 pF by 30 pF reveals that the maximum cable length is approximately 80 feet. If a longer cable length is required, the user would need to find a cable with a smaller mutual capacitance.

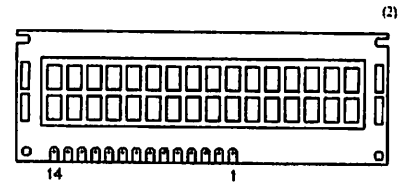
Instruction	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Description	Clocks
NOP	0	0	0	0	0	0	0	0	0	0	No Operation	0
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears display & sets address counter to zero.	165
Cursor Home	0	0	0	0	0	0	0	0	1	0	Sets address counter to zero. returns shifted display to original position. DDRAM contents remains unchanged.	3
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction. and specifies automatic shift.	3
Display Control	0	0	0	0	0	0	1	D	C	B	Turns display (D), cursor on/off (C) or cursor blinking(B).	3
Cursor/display shift	0	0	0	0	0	1	S/C	R/L	0	0	Moves cursor and shift display. DDRAM contents remains unchanged.	3
Function Set	0	0	0	0	1	DL	N	M	G	0	Sets interface data width(DL), number of display lines (N.M) and voltage generator control (G).	3
Set CGRAM Addr	0	0	0	1	Character Generator RAM					Sets CGRAM Address	3	
Set DDRAM Addr	0	0	1	Display Data RAM Address					Sets DDRAM Address	3		
Busy Flag & Addr	0	1	BF	Address Counter					Reads Busy Flag & Address Counter	0		
Read Data	1	0	Read Data					Reads data from CGRAM or DDRAM	3			
Write Data	1	1	Write Data					Writes data from CGRAM or DDRAM	3			



Parameter	Symbol	Min ⁽¹⁾	Typ ⁽¹⁾	Max ⁽¹⁾	Unit
Enable Cycle Time	t_c	500	-	-	ns
Enable Pulse Width (High)	t_w	230	-	-	ns
Enable Rise/Fall Time	t_r, t_f	-	-	20	ns
Address Setup Time	t_{as}	40	-	-	ns
Address Hold Time	t_{ah}	10	-	-	ns
Data Setup Time	t_{ds}	80	-	-	ns
Data Hold Time	t_h	10	-	-	ns

Note¹ The above specifications are a indication only. Timing will vary from manufacturer to manufacturer.

Note² A 2 line by 16 Character LCD Module is Pictured. Data will work on most 1 line x 16 character, 1 line x 20 character, 2 line x 16 character, 2 line x 20 character, 4 lines x 20 character, 2 lines x 40 character etc. modules compatible with the HD44780 LCD Module.



Pin No	Name	I/O	Description
1	V _{ss}	Power	GND
2	V _{dd}	Power	-5v
3	V _o	Analog	Contrast Control
4	RS	Input	Register Select
5	RW	Input	Read Write
6	E	Input	Enable (Strobe)
7	D0	I/O	Data LSB
8	D1	I/O	Data
9	D2	I/O	Data
10	D3	I/O	Data
11	D4	I/O	Data
12	D5	I/O	Data
13	D6	I/O	Data
14	D7	I/O	Data MSB

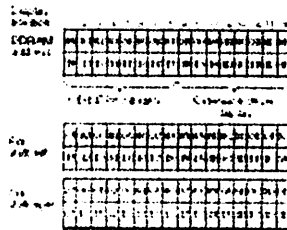
M1632 MODULE LCD 16 X 2 BARIS (M1632)

Deskripsi:

M1632 adalah module modul LCD dengan tampilan 16 x 2 baris dengan kecerahan daya rendah. Modul ini dilengkapi dengan mikrokontroler yang didesain khusus untuk mengendalikan IC Mikrokontroler HD44780 bagian Hardswara yang berfungsi sebagai pengontrol LCD ini mempunyai CGROM (Character Generator) dan PCAL Only Memory (CGRAM) (Character Generator Random Access Memory) (CGRAM) Display Data Random Access Memory.

DDRAM

DDRAM adalah merupakan memori untuk karakter yang ditayangkan berbaris. Contoh: jika karakter 'A' atau 'H' yang ditulis pada alamat 00, maka karakter tersebut akan tampil pada baris pertama dan kolom pertama dan LCD. Apabila karakter tersebut ditulis di alamat 01, maka karakter tersebut akan tampil pada baris kedua kolom pertama dan LCD.



Gambar 1
DDRAM M1632 diambil dari data sheet HD44780

CGROM

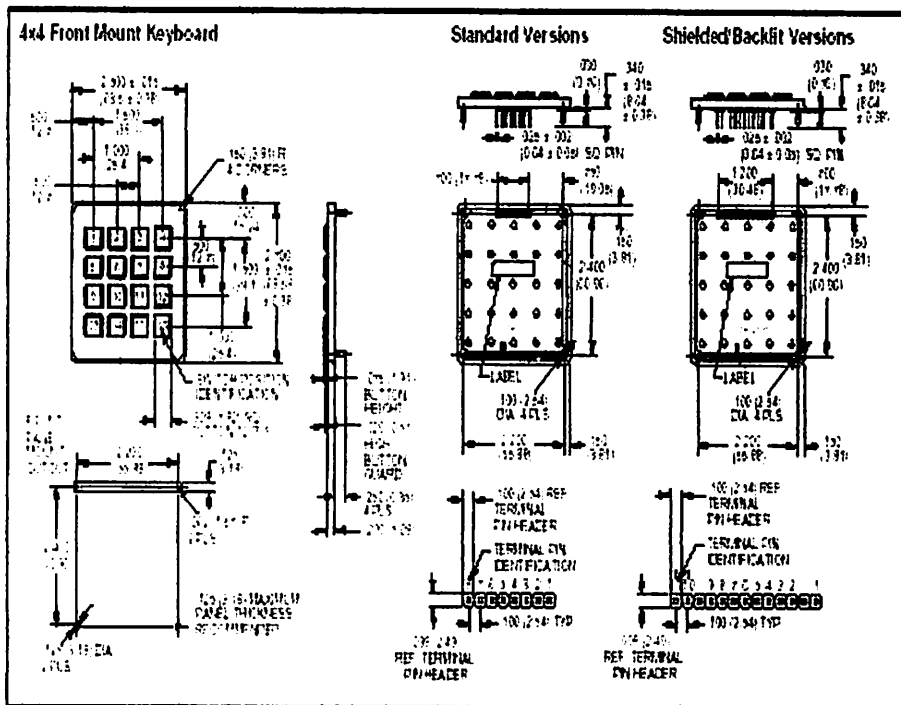
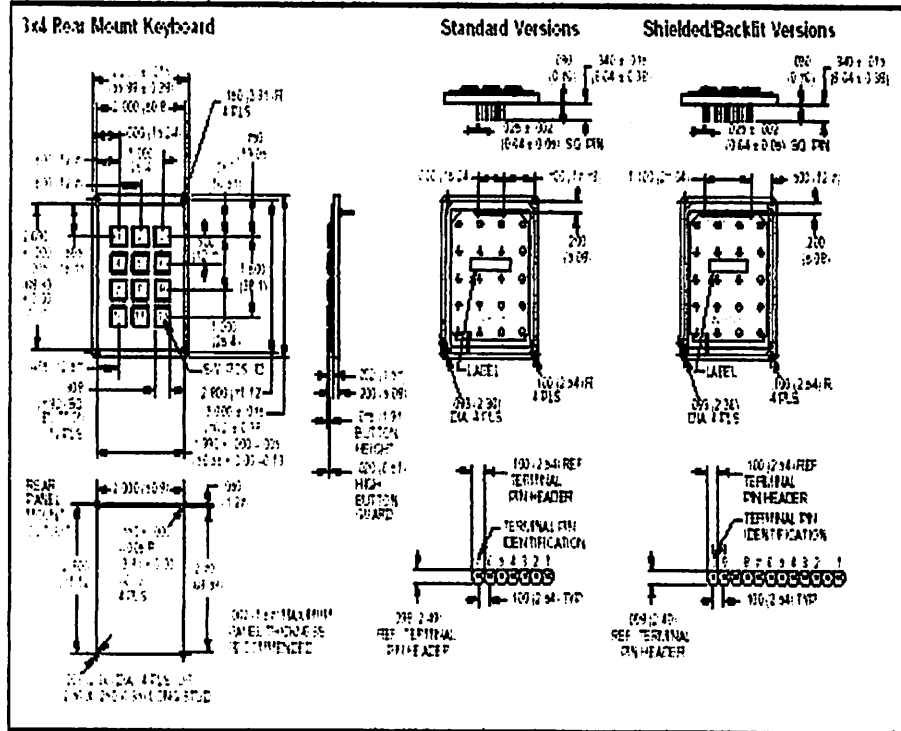
CGROM adalah merupakan memori untuk menampung karakter pada baris dan kolom. Jika dituliskan ke dalam memori. Namun memori akan hilang saat power off maka akan hilang jika karakter akan hilang.

CGROM

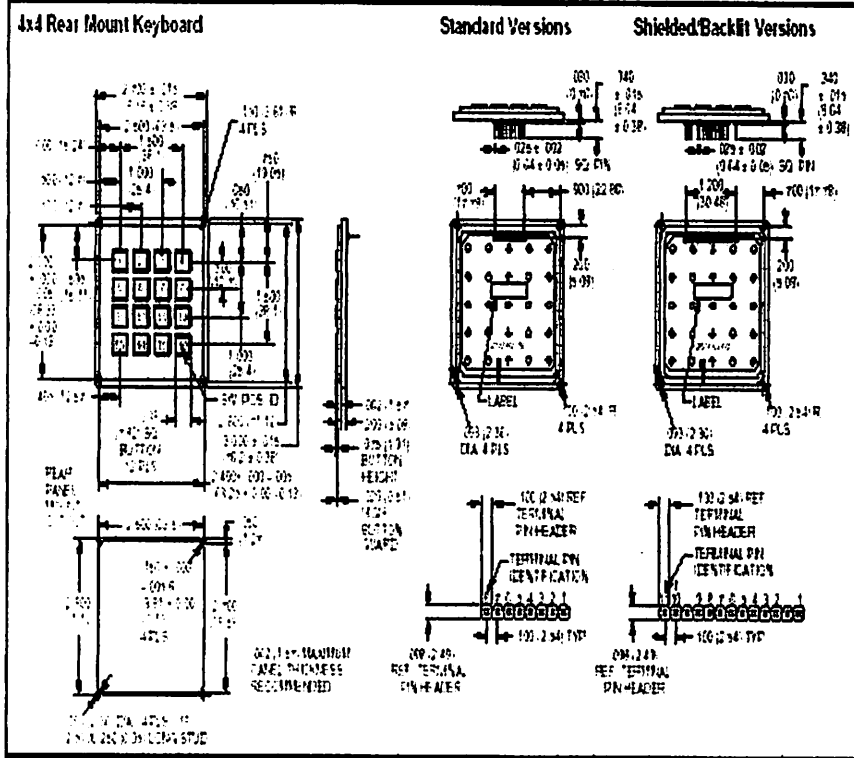
CGROM adalah merupakan memori untuk menampung karakter pada baris dan kolom. Jika dituliskan ke dalam memori. Namun memori akan hilang saat power off maka akan hilang jika karakter akan hilang.

Pada gambar 2, tampak terlihat pada-pada karakter yang tersimpan dalam memori dengan cara ini. Pada saat HD44780 akan menuliskan data 41H yang tersimpan pada DDRAM in HD44780 akan menuliskan data di alamat 41H (01000001) yang ada pada CGROM yaitu pada karakter A.

DIMENSIONS in inches (and millimeters)



DIMENSIONS in inches (and millimeters)



CODE AND TRUTH TABLES

Dots in the chart indicate connected terminals when switch is closed. Terminals are identified on the keyboard.

12 Button Keypads

BUTTON LOCATION	MATRIX CODES												
	Standard						Shielded/Backlit						
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
	1	2	3	4	5	6	7	8	9	10	11	12	
	A	B	C	D	E	F	G	H	I	J	K	L	
	M	N	O	P	Q	R	S	T	U	V	W	X	
	Y	Z	1	2	3	4	5	6	7	8	9	0	
	1	2	3	4	5	6	7	8	9	0	1	2	
	A	B	C	D	E	F	G	H	I	J	K	L	
	M	N	O	P	Q	R	S	T	U	V	W	X	
	Y	Z	1	2	3	4	5	6	7	8	9	0	

TERMINAL LOCATION

16 Button Keypads

BUTTON LOCATION	MATRIX CODES															
	Standard								Shielded/Backlit							
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																
16																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5	6
	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5	6
	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2

TERMINAL LOCATION

Pembacaan format data RFID

AN-05

Oleh: Tim Digiware

Saat ini model alat identifikasi sangatlah bermacam – macam ada yang berupa kartu dengan lubang, barcode, RFID, dll. RFID (RF Identification) merupakan suatu alat untuk identifikasi yang biasanya ditempelkan pada barang atau dibuat menjadi kartu. Di dalam artikel ini akan dibahas mengenai cara membaca format data yang dikeluarkan oleh RFID reader dengan format output ASCII.

Pendahuluan

RFID reader mempunyai banyak sekali tipe, antara lain: ID-10, ID-19, EM-13, dll. Biasanya RFID reader ini memiliki dua bentuk output serial yaitu: ASCII dan Wiegand 26-bit. Yang paling banyak digunakan adalah output dengan format ASCII, karena output ini sangat mudah untuk dihubungkan pada mikrokontroler atau PC menggunakan komunikasi serial UART.

Format Data ASCII

Output yang memiliki format ASCII memiliki struktur sebagai berikut:

02	10 data karakter ASCII	checksum	CR	LF	03
----	------------------------	----------	----	----	----

Gb 1. Format data ASCII

Checksum merupakan hasil EXOR (Exclusive OR) dari 5 biner data byte. Untuk lebih jelasnya tentang cara pembacaan format ASCII, lihat contoh berikut.

Misalnya data output serial (dalam hexadesimal) yang kita tangkap adalah sebagai berikut:

02	30	34	36	32	30	31	44	37	36	43	44	43	0D	0A	03
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Langkah pertama adalah merubah semua nilai data diatas menjadi karakter ASCII. Misalnya 30H menjadi karakter "0", 34H menjadi karakter "4", dst. Langkah kedua adalah menyusun data – data tersebut ke dalam Format Data ASCII seperti gb 1. Kemudian ambil 10 data karakter ASCII. Dalam contoh ini berarti data tersebut adalah:

30	34	36	32	30	31	44	37	36	43	Data Heksa
		6	2	0	1	D	7	6	C	Data ASCII

Untuk data dengan warna biru merupakan data untuk jenis – jenis kartu dan tidak digunakan dalam proses konversi. yang akan dipakai disini adalah data yang ke 3 s/d 10. Hasil konversi dari data heksa ke dalam data ASCII adalah "6201D76C". Gabungkan karakter data ASCII menjadi bilangan Hexadesimal, kemudian konversikan bilangan hexadesimal tsb ke dalam desimal. Hasilnya sebagai berikut: 6201D76C H menjadi 1644287852 (ini merupakan nomor kartu sebenarnya yang tertera pada badan kartu tsb). Cara ini hanya berlaku pada kartu yang tidak dienkrpsi.

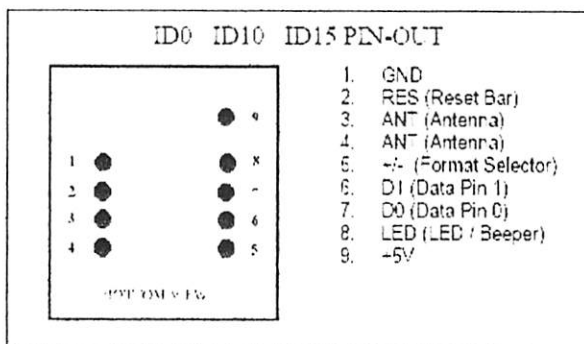
Selamat mencoba.

Low Cost Short-Range Proximity Readers

ID-0 / ID-10 / ID-15 MK(ii) Series

Maintenance only: Consider using the ID-2, ID-12 and the ID-20 that have improved performance.

The ID Series short-range readers come in three different sizes and read ranges. Both the ID-10 and ID-15 come with internal antennas, and have read ranges of 12+ cm and 16+ cm, respectively. With an external antenna, the ID-0 MK(ii) can deliver read ranges of up to 25 cm. All three readers support ASCII and Wiegand26 data formats.



Operational and Physical Characteristics

Parameters	ID-0	ID-10	ID-15
Read Range	N/A (no internal antenna)	12+ cm	15+ cm
Dimensions	21 mm x 19 mm x 6 mm	26 mm x 25 mm x 7 mm	40 mm x 40 mm x 9 mm
Frequency	125 kHz	125 kHz	125 kHz
Card Format	EM 4001 or compatible	EM 4001 or compatible	EM 4001 or compatible
Encoding	Manchester 64-bit, modulus 64	Manchester 64-bit, modulus 64	Manchester 64-bit, modulus 64
Power Requirement	5 VDC @ 13mA nominal	5 VDC @ 30mA nominal	5 VDC @ 50mA nominal
I/O Output Current	+/-200mA PK	-	-
Voltage Supply Range	+4.6V through +5.4V	+4.6V through +5.4V	+4.6V through +5.4V

Pin Description & Output Data Formats

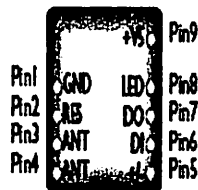
Pin No.	Description	ASCII	Wiegand26
Pin 1	Zero Volts and Tuning Capacitor Ground	GND 0V	GND 0V
Pin 2	Strap to +5V	Reset Bar	Reset Bar
Pin 3	To External Antenna and Tuning Capacitor	Antenna	Antenna
Pin 4	To External Antenna	Antenna	Antenna
Pin 5	Format Selector (+/-)	Strap to GND	Strap to +5V
Pin 6	Data 1	CMOS	One Output
Pin 7	Data 0	TTL Data (inverted)	Zero Output
Pin 8	3.1 kHz Logic	Beeper / LED	Beeper / LED
Pin 9	DC Voltage Supply	+5V	+5V

Advanced Digital Reader Technology

---Better In Design---

ID-10 SERIES LOW COST PROXIMITY READER

BOOTTOM VIEW



Power Requirement : 5V@13mA nominal I/O Output Current : 20mA sink/source
 Card Format : EM4001 or compatible Drive Current : 300mA
 Frequency : 125KHz Antenna Volt : 100 Volt PKPK
 Encoding : Manchester 62bit, modulus 64

OUTPUT FORMAT-ASCII



The checksum is the result of the 'exclusive or' of the 5 Binary Data bytes (the 10ASCII data characters)

Data baud rate: 9600

ASCII (RS232)

Pin1	Ground 0V	Zero volts and Tuning Capacitor Ground
Pin2	Reset Bar	Strap to +5V
Pin3	Antenna	NC
Pin4	Antenna	NC
Pin5	Strap to Ground	
Pin6	CMOS	Serial ASCII
Pin7	TTL Data	Serial ASCII Inverted
Pin8	BEEP/LED	2.7KHz Logic
Pin9	+4.6 through +5.5V	Supply DC volts

DATA STRUCTURE WIEGAND 26 BIT

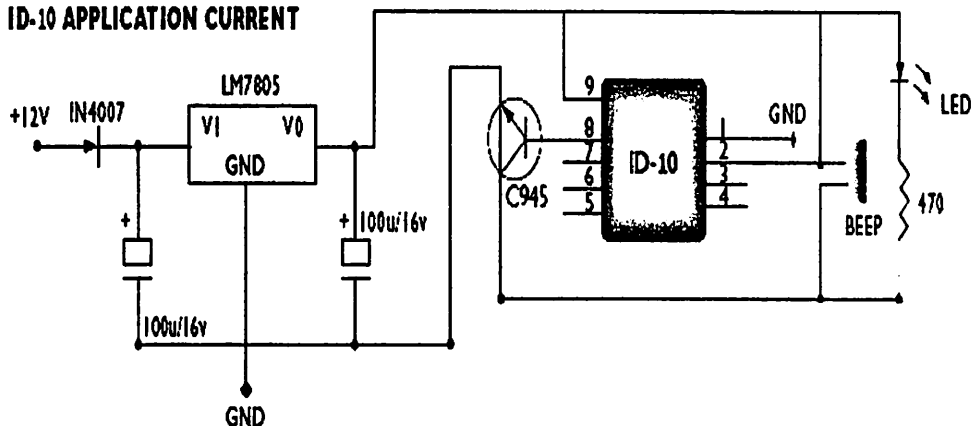


P=Parity Start Bit and Stop Bit

WIEGAND 26

Pin1	Ground 0V	Zero volts and Tuning Capacitor Ground
Pin2	Reset Bar	Strap to +5V
Pin3	Antenna	NC
Pin4	Antenna	NC
Pin5	Strap to Pin +5V	
Pin6	One Output	
Pin7	Zero Output	
Pin8	BEEP/LED	2.7KHz Logic
Pin9	+4.6 through +5.5V	Supply DC volts

ID-10 APPLICATION CURRENT



1. 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 LOFP. 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 uA (VCC = 3.0V, Wait mode, Peripheral clock stops) Typ. 0.7 uA (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 model LQFP	

1.3 Block Diagram

Figure 1.1 shows this MCU block diagram.

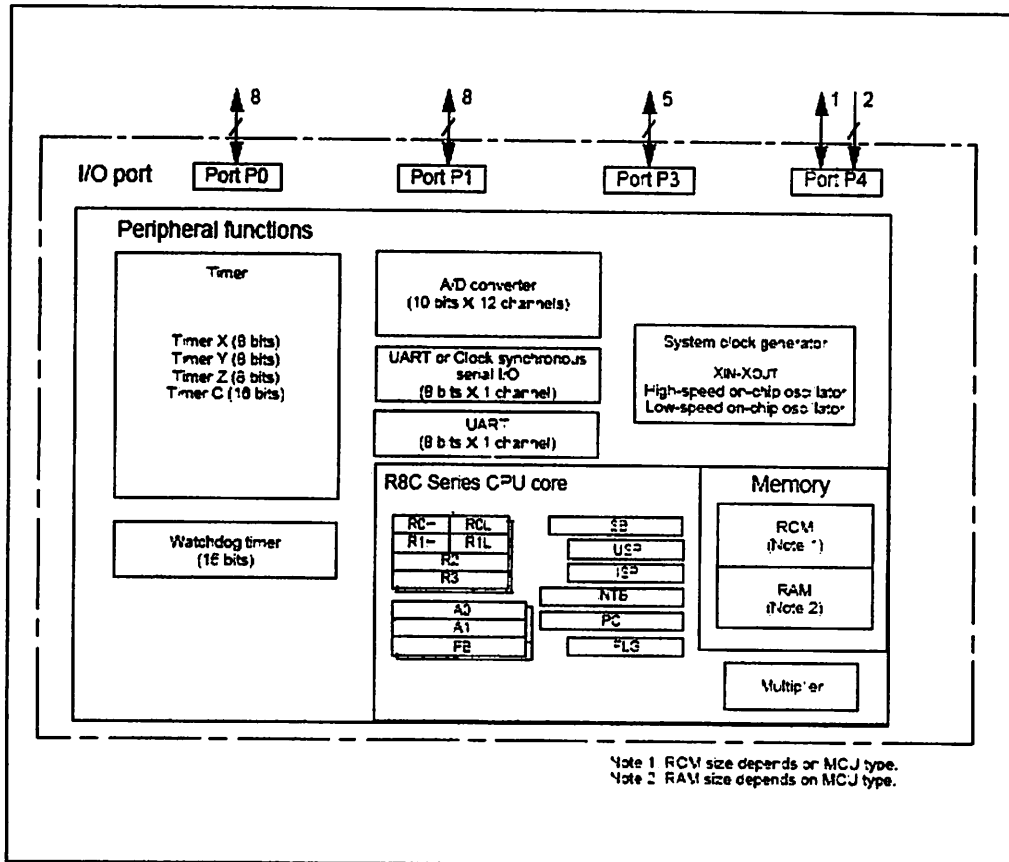


Figure 1.1 Block Diagram

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	

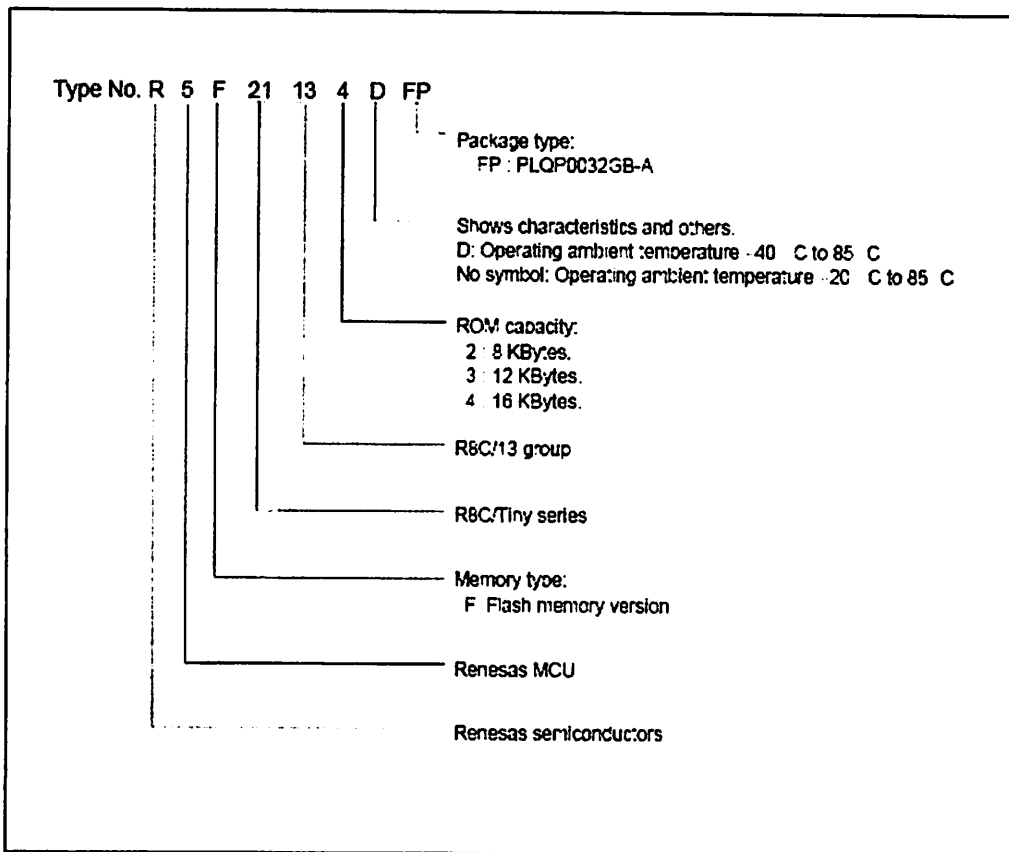


Figure 1.2 Type No., Memory Size, and Package

1.5 Pin Assignments

Figure 1.3 shows the pin configuration (top view).

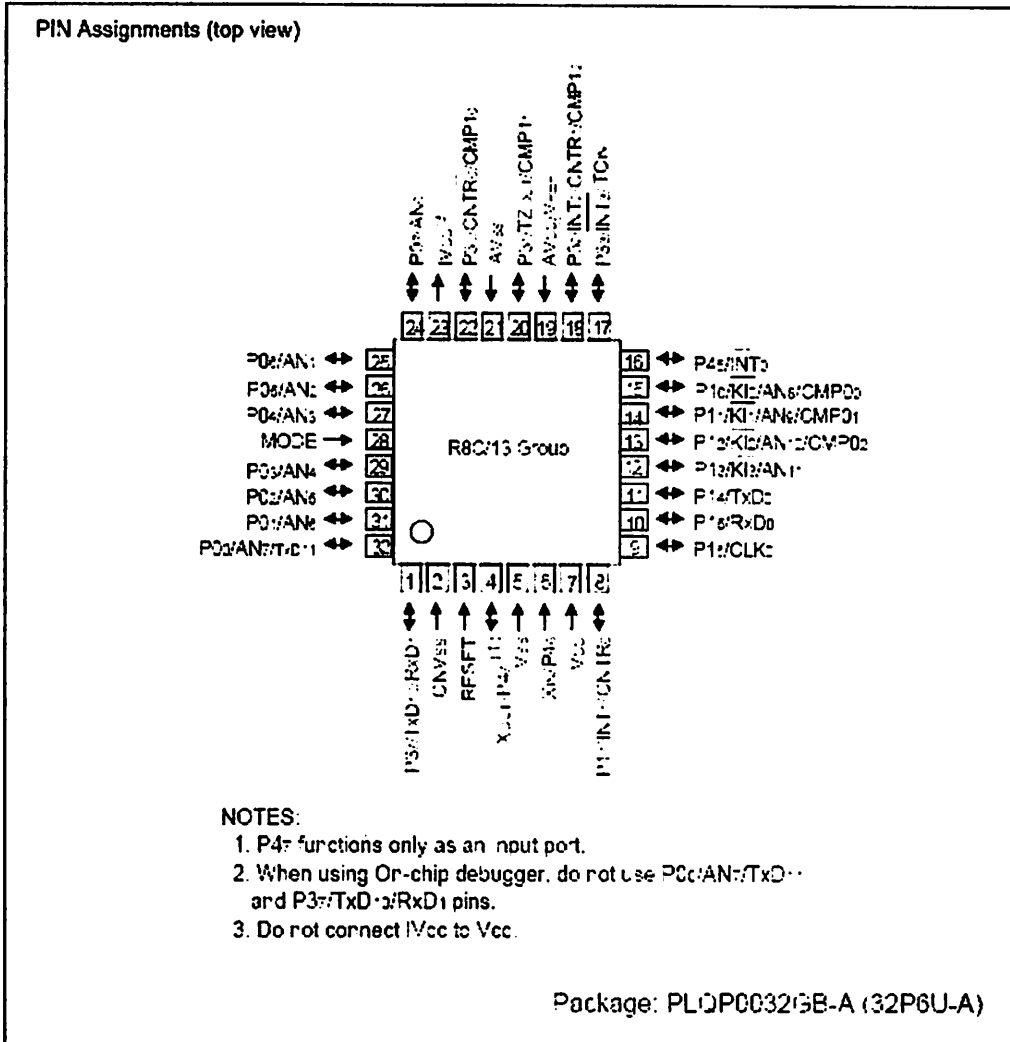


Figure 1.3 Pin Assignments (Top View)

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.
Vcc	Vcc	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	RESET	I	_ on this input resets the MCU.
CNVss	CNVss	I	Connect this pin to Vss via a resistor.
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	
INT interrupt input	INT0 to INT5	I	These are INT interrupt input pins.
Key input interrupt input	KI0 to KI7	I	These are key input interrupt pins.
Timer X	CNTR0	I/O	This is the timer X I/O pin.
	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	CLK2	I/O	This is a transfer clock I/O pin.
	RxD0, RxD1	I	These are serial data input pins.
	TxD0, TxD1	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	PC0 to PC7, 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.
Input 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.

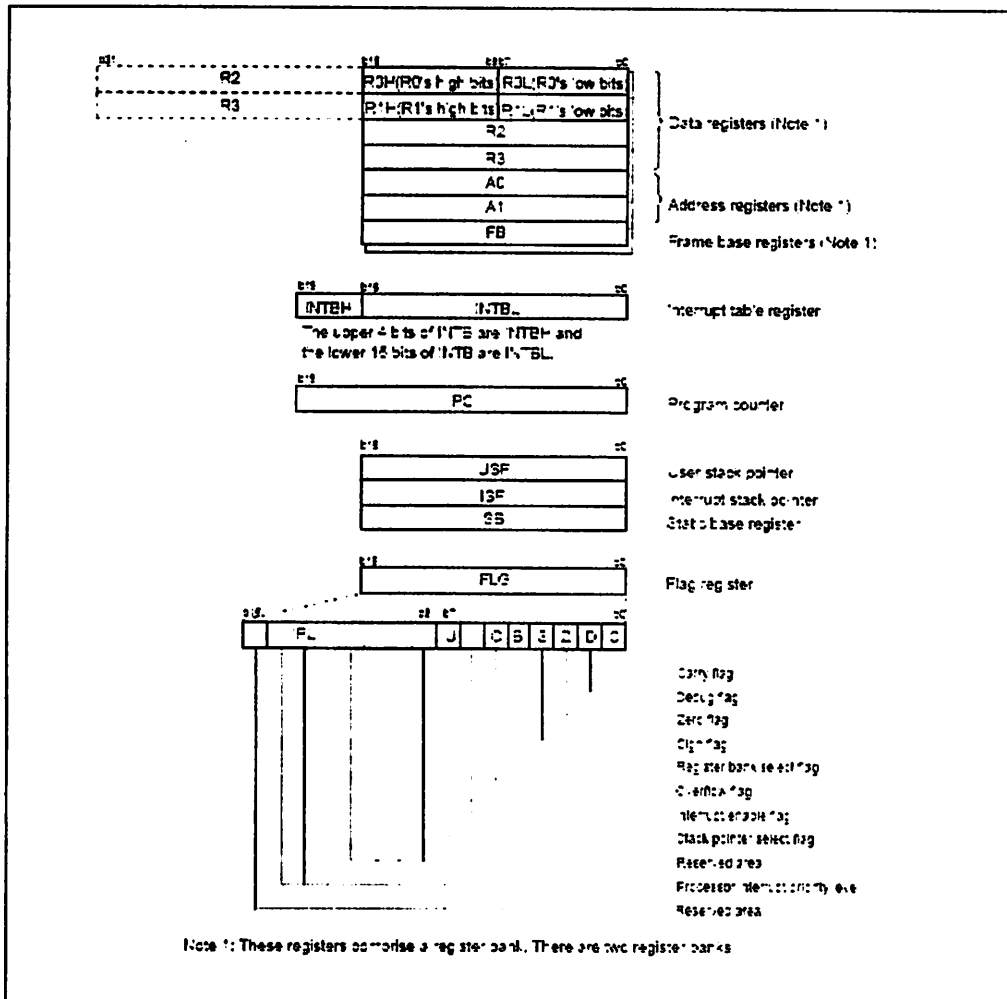


Figure 2.1 Central Processing Unit Register

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.

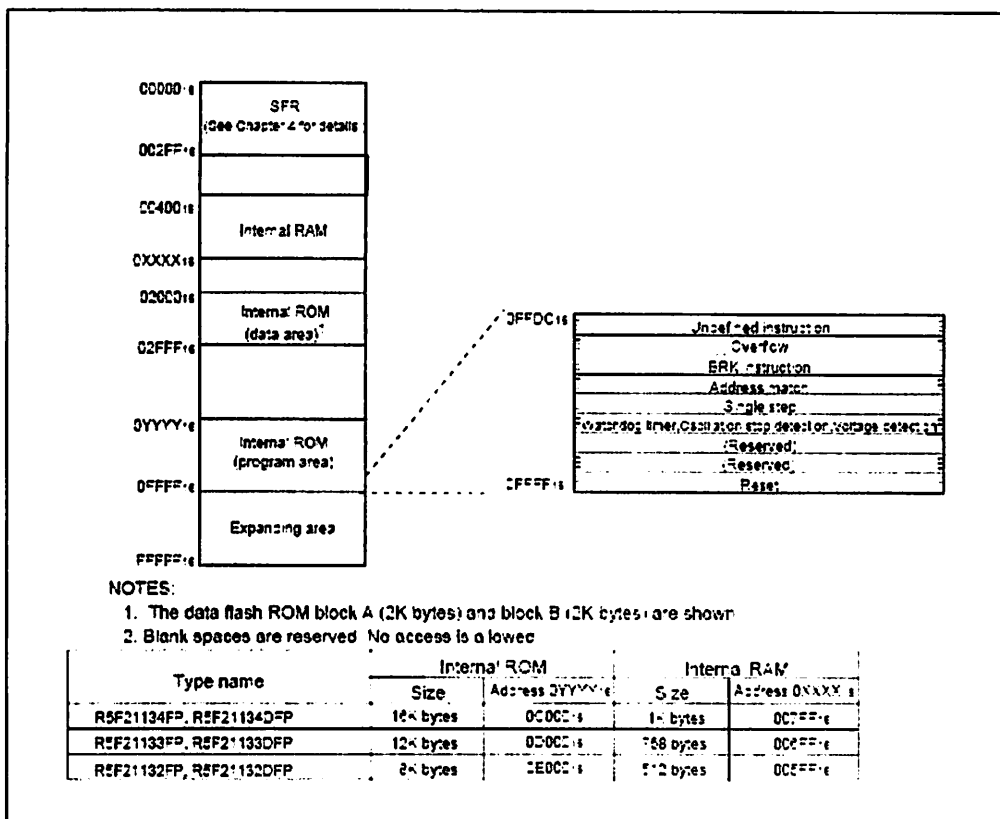


Figure 3.1 Memory Map

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
00C0:0			
00C1:0			
00C2:0			
00C3:0			
00C4:0	Processor mode register C 1	PVC	C318
00C5:0	Processor mode register 1	PV1	C318
00C6:0	System clock control register 0	CMS	C11D16000
00C7:0	System clock control register 1	CM1	C31DC620C
00C8:0	High-speed on-chip oscillator control register 0	HPL	C318
00C9:0	Address match interrupt enable register	AIER	XXXXXXXX000
00CA:0	Protect register	FRCR	00XXXX000
00CB:0	High-speed on-chip oscillator control register 1	HP1	C318
00CC:0	Oscillation stop detection register	OSD	C3003*000
00CD:0	Watchdog timer reset register	WDR	XXXX
00CE:0	Watchdog timer start register	WDS	XXXX
00CF:0	Watchdog timer control register	WDC	C33*1*1*0
0C10:0	Address match interrupt register 0	RMAD0	C318 C318 XXXX
0C11:0			
0C12:0			
0C13:0			
0C14:0	Address match interrupt register 1	RMAD1	C318 C318 XXXX
0C15:0			
0C16:0			
0C17:0			
0C18:0			
0C19:0	Voltage detection register 1 2	VCR1	C3C310000
0C1A:0	Voltage detection register 2 2	VCR2	C3183 13C3C0000 4
0C1B:0			
0C1C:0			
0C1D:0			
0C1E:0	INT0 register select register	INTS	XXXXXXXX000
0C1F:0	Voltage detection interrupt register 2	DAINT	C3183 01000000 2 4
0C20:0			
0C21:0			
0C22:0			
0C23:0			
0C24:0			
0C25:0			
0C26:0			
0C27:0			
0C28:0			
0C29:0			
0C2A:0			
0C2B:0			
0C2C:0			
0C2D:0			
0C2E:0			
0C2F:0			
0C30:0			
0C31:0			
0C32:0			
0C33:0			
0C34:0			
0C35:0			
0C36:0			
0C37:0			
0C38:0			
0C39:0			
0C3A:0			
0C3B:0			
0C3C:0			
0C3D:0			
0C3E:0			
0C3F:0			

X: Undefined

NOTES

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 RESE pin = H retaining.

Table 4.2 SFR Information(2)(1)

Address	Register	Symbol	After reset
0040 ^h			
0041 ^h			
0042 ^h			
0043 ^h			
0044 ^h			
0045 ^h			
0046 ^h			
0047 ^h			
0048 ^h			
0049 ^h			
004A ^h			
004B ^h			
004C ^h			
004D ^h	Key input interrupt control register	KIIC	XXXX0000
004E ^h	AD conversion interrupt control register	ADIC	XXXX0000
004F ^h			
0050 ^h	Compare 1 interrupt control register	CMP1IC	XXXX0000
0051 ^h	UART0 transmit interrupt control register	SOIC	XXXX0000
0052 ^h	UART0 receive interrupt control register	SOBIC	XXXX0000
0053 ^h	UART1 transmit interrupt control register	SITC	XXXX0000
0054 ^h	UART1 receive interrupt control register	SIRIC	XXXX0000
0055 ^h	INT2 interrupt control register	INT2IC	XXXX0000
0056 ^h	Timer X interrupt control register	TXIC	XXXX0000
0057 ^h	Timer Y interrupt control register	TYIC	XXXX0000
0058 ^h	Timer Z interrupt control register	TZIC	XXXX0000
0059 ^h	INT1 interrupt control register	INT1IC	XXXX0000
005A ^h	INT3 interrupt control register	INT3IC	XXXX0000
005B ^h	Timer C interrupt control register	TCIC	XXXX0000
005C ^h	Compare 0 interrupt control register	CMP0IC	XXXX0000
005D ^h	INT0 interrupt control register	INT0IC	XXXX0000
005E ^h			
005F ^h			
0060 ^h			
0061 ^h			
0062 ^h			
0063 ^h			
0064 ^h			
0065 ^h			
0066 ^h			
0067 ^h			
0068 ^h			
0069 ^h			
006A ^h			
006B ^h			
006C ^h			
006D ^h			
006E ^h			
006F ^h			
0070 ^h			
0071 ^h			
0072 ^h			
0073 ^h			
0074 ^h			
0075 ^h			
0076 ^h			
0077 ^h			
0078 ^h			
0079 ^h			
007A ^h			
007B ^h			
007C ^h			
007D ^h			
007E ^h			
007F ^h			

X: Undefined

NOTES

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

Table 4.3 SFR Information(3)(1)

Address	Register	Symbol	After reset
00E0 ₁₆	Timer Y, Z mode register	TYZMR	00 ₁₆
00E1 ₁₆	Prescaler Y	PREY	FF ₁₆
00E2 ₁₆	Timer Y secondary	TYSC	FF ₁₆
00E3 ₁₆	Timer Y primary	TYPR	FF ₁₆
00E4 ₁₆	Timer Y, Z waveform output control register	TYZM	00 ₁₆
00E5 ₁₆	Prescaler Z	PREZ	FF ₁₆
00E6 ₁₆	Timer Z secondary	TZSC	FF ₁₆
00E7 ₁₆	Timer Z primary	TZPR	FF ₁₆
00E8 ₁₆			
00E9 ₁₆			
00EA ₁₆	Timer Y, Z output control register	TYZOC	00 ₁₆
00EB ₁₆	Timer X mode register	TXMR	00 ₁₆
00EC ₁₆	Prescaler X	PREX	FF ₁₆
00ED ₁₆	Timer X register	TX	FF ₁₆
00EE ₁₆	Count source set register	TCSS	00 ₁₆
00EF ₁₆			
00F0 ₁₆	Timer C register	TC	00 ₁₆
00F1 ₁₆			00 ₁₆
00F2 ₁₆			
00F3 ₁₆			
00F4 ₁₆			
00F5 ₁₆			
00F6 ₁₆	External input enable register	XTEN	00 ₁₆
00F7 ₁₆			
00F8 ₁₆	Key input enable register	KEN	00 ₁₆
00F9 ₁₆			
00FA ₁₆	Timer C control register 0	TCC0	00 ₁₆
00FB ₁₆	Timer C control register 1	TCC1	00 ₁₆
00FC ₁₆	Capture, compare 0 register	TMD	00 ₁₆
00FD ₁₆			00 ₁₆
00FE ₁₆	Compare 1 register	TM1	FF ₁₆
00FF ₁₆			FF ₁₆
02A0 ₁₆	UART0 transmit/receive mode register	U0MR	00 ₁₆
02A1 ₁₆	UART0 bit rate register	U0BRG	XX ₁₆
02A2 ₁₆	UART0 transmit buffer register	U0TB	XX ₁₆
02A3 ₁₆			XX ₁₆
02A4 ₁₆	UART0 transmit/receive control register 0	U0CC	0000 ₁₆
02A5 ₁₆	UART0 transmit/receive control register 1	U0C1	000000 ₁₆
02A6 ₁₆	UART0 receive buffer register	U0RB	XX ₁₆
02A7 ₁₆			XX ₁₆
02A8 ₁₆	UART1 transmit/receive mode register	U1MR	00 ₁₆
02A9 ₁₆	UART1 bit rate register	U1BRG	XX ₁₆
02AA ₁₆	UART1 transmit buffer register	U1TB	XX ₁₆
02AB ₁₆			XX ₁₆
02AC ₁₆	UART1 transmit/receive control register 0	U1CC	0000 ₁₆
02AD ₁₆	UART1 transmit/receive control register 1	U1C1	000000 ₁₆
02AE ₁₆	UART1 receive buffer register	U1RB	XX ₁₆
02AF ₁₆			XX ₁₆
02B0 ₁₆	UART1 transmit/receive control register 2	U1CC1	00 ₁₆
02B1 ₁₆			
02B2 ₁₆			
02B3 ₁₆			
02B4 ₁₆			
02B5 ₁₆			
02B6 ₁₆			
02B7 ₁₆			
02B8 ₁₆			
02B9 ₁₆			
02BA ₁₆			
02BB ₁₆			
02BC ₁₆			
02BD ₁₆			
02BE ₁₆			
02BF ₁₆			

X: Undefined

NOTES

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

2 When the output compare mode is selected (the "CC13 bit in the "CC1 register" = 1), the value is set to "FFFF".

Table 4.4 SFR Information(4)(1)

Address	Register	Symbol	After reset
00C0 ₁₆	AD register	AD	XX ¹⁶
00C1 ₁₆			XX ¹⁶
00C2 ₁₆			
00C3 ₁₆			
00C4 ₁₆			
00C5 ₁₆			
00C6 ₁₆			
00C7 ₁₆			
00C8 ₁₆			
00C9 ₁₆			
00CA ₁₆			
00CB ₁₆			
00CC ₁₆			
00CD ₁₆			
00CE ₁₆			
00CF ₁₆			
00D0 ₁₆			
00D1 ₁₆			
00D2 ₁₆			
00D3 ₁₆			
00D4 ₁₆	AD control register 2	ADCON2	0010
00D5 ₁₆			
00D6 ₁₆	AD control register 0	ADCON0	0000XX ¹⁶
00D7 ₁₆	AD control register 1	ADCON1	0010
00D8 ₁₆			
00D9 ₁₆			
00DA ₁₆			
00DB ₁₆			
00DC ₁₆			
00DD ₁₆			
00DE ₁₆			
00DF ₁₆			
00E0 ₁₆	Port P0 register	P0	XX ¹⁶
00E1 ₁₆	Port P1 register	P1	XX ¹⁶
00E2 ₁₆	Port P0 direction register	P0D	0010
00E3 ₁₆	Port P1 direction register	P0D	0010
00E4 ₁₆			
00E5 ₁₆	Port P3 register	P3	XX ¹⁶
00E6 ₁₆			
00E7 ₁₆	Port P3 direction register	P0D	0010
00E8 ₁₆	Port P4 register	P4	XX ¹⁶
00E9 ₁₆			
00EA ₁₆	Port P4 direction register	P04	0010
00EB ₁₆			
00EC ₁₆			
00ED ₁₆			
00EE ₁₆			
00EF ₁₆			
00F0 ₁₆			
00F1 ₁₆			
00F2 ₁₆			
00F3 ₁₆			
00F4 ₁₆			
00F5 ₁₆			
00F6 ₁₆			
00F7 ₁₆			
00F8 ₁₆			
00F9 ₁₆			
00FA ₁₆			
00FB ₁₆			
00FC ₁₆	Pull-up control register 0	PUP0	00XX0000
00FD ₁₆	Pull-up control register 1	PUP1	XXXXXX ¹⁶
00FE ₁₆	Port P1 drive capacity control register	DRR	0010
00FF ₁₆	Timer C output control register	CCOUT	0010
01B3 ₁₆	Flash memory control register 4	FVR4	01000000
01B4 ₁₆			
01B5 ₁₆	Flash memory control register 1	FVR1	100000XX
01B6 ₁₆			
01B7 ₁₆	Flash memory control register 3	FVR3	00000010
0FFF ₁₆	Option function select register (2)	OPS	Note 2

X Undefined

NOTES
 1. The blank areas, 0100₁₆ to 01B2₁₆ and 01B2₁₆ to 02FF₁₆ are reserved and cannot be used by users.
 2. The watchdog timer control bit is assigned. Refer to "Figure 11.2. OWS, WCC, WDTR and WDT3 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} =AV _{CC}	-0.3 to 6.5	V
AV _{CC}	Analog supply voltage	V _{CC} =AV _{CC}	-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 _c	Power dissipation	T _{op} =25 °C	200	mW
T _{op}	Operating ambient temperature		-20 to 85 / -40 to 85 (Divers on)	°C
T _{stg}	Storage temperature		-55 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
AV _{CC}	Analog supply voltage			V _{CC} ¹		V
V _{SS}	Supply voltage					V
AV _{SS}	Analog supply voltage					V
V _I	Input voltage		0.5V _{CC}		V _{CC}	V
V _{IL}	Low input voltage				0.2V _{CC}	V
I _{OH} (sum)	"H" peak all output currents (peak)	Sum of 3 pins' OH	0		-50.0	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 (peak)	Sum of 3 pins' OL			10	mA
I _{OL} (peak)	"L" peak output current	Except P1 to P1-			10	mA
I _{OL} (avg)	"L" average output current	P1 to P1-	Drive ability HIGH		10	mA
			Drive ability LOW		10	mA
I _{OL} (avg)	"L" average output current	Except P1 to P1-	Drive ability HIGH		5	mA
		P1 to P1-	Drive ability LOW		5	mA
f _(XIN)	Master clock input oscillator frequency	3.0V ≤ V _{CC} ≤ 5.5V 2.7V ≤ V _{CC} ≤ 3.0V			10	MHz

Note

1. Referenced to V_{CC} = AV_{CC} = 2.7 to 5.5V at T_{op} = -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}=AV_{CC}.

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	$f_{AD}=10\text{ MHz}$, $V_{ref}=V_{CC}=5.0\text{ V}$	± 5	LSB
		8 bit mode	$f_{AD}=10\text{ MHz}$, $V_{ref}=V_{CC}=5.0\text{ V}$	± 2	LSB
		10 bit mode	$f_{AD}=10\text{ MHz}$, $V_{ref}=V_{CC}=3.3\text{ V}^3$	± 5	LSB
		8 bit mode	$f_{AD}=10\text{ MHz}$, $V_{ref}=V_{CC}=3.3\text{ V}^3$	± 2	LSB
R_{LADDER}	Ladder resistance	$V_{REF}=V_{CC}$	10	40	k Ω
t_{CONV}	Conversion time	10 bit mode $f_{AD}=10\text{ MHz}$, $V_{ref}=V_{CC}=5.0\text{ V}$	3.2		μs
		8 bit mode $f_{AD}=10\text{ MHz}$, $V_{ref}=V_{CC}=5.0\text{ V}$	2.9		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 5.5 V at $T_{opr} = -20$ to $85\text{ }^\circ\text{C}$ / -40 to $85\text{ }^\circ\text{C}$ unless otherwise specified.
- 2: When f_{AD} is 10 MHz more, divide the f_{AD} and make A/D operation clock frequency (f_{OADC}) lower than 10 MHz.
- 3: When the AV_{CC} is less than 4.2V, divide the f_{AD} and make A/D operation clock frequency (f_{OADC}) lower than $f_{AD}/2$.
- 4: Set $V_{CC}=V_{ref}$.

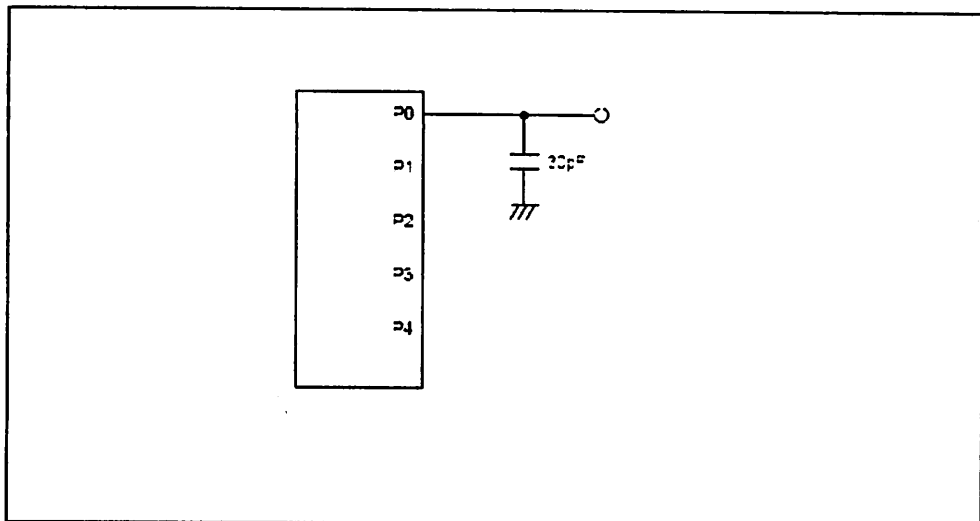


Figure 5.1 Port P0 to P4 measurement circuit

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
(tISR-ES)	Time delay from Suspend Request until Erase Suspend		—	—	6	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 Characteristics⁴

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max	
-	Program/Erase endurance ²		10000 ⁵	—	—	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	—	85	—	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	—	0.2	—	s
(tISR-ES)	Time delay from Suspend Request until Erase Suspend		—	—	3	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	—	95	°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 2548 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 5.5 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.

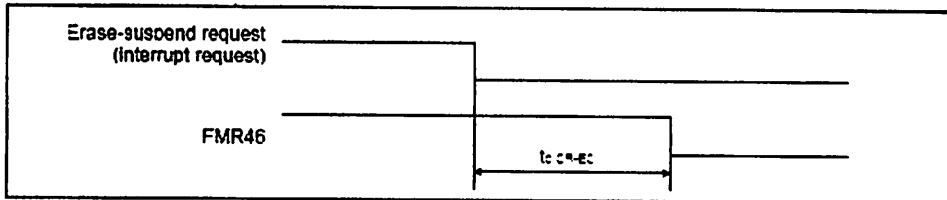


Figure 5.2 Time delay from Suspend Request until Erase Suspend

Table 5.6 Voltage Detection Circuit Electrical Characteristics

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
Vdet	Voltage detector level	V _{CC7} = V _{CC} ± 5V	3.9	4.2	—	V
	Voltage detection interrupt request generating time ²		—	46	—	μs
tk(EA)	Voltage detector circuit reset consumption current	V _{CC7} = V _{CC} ± 5V	—	602	—	μA
tw(Vdet)	Waiting time until voltage detector circuit operation starts ³		—	—	50	μs
V _{opmin}	Microcomputer operation voltage minimum value	—	—	—	V	

NOTES:

- The measuring condition is V_{CC7}=V_{CC}±2.7V to ±5.0V and T_{opr}=40°C to 35°C.
- This shows the time until the voltage detection interrupt request is generated since the voltage passes V_{det}.
- This shows the required time until the voltage detection circuit operates when setting to '1' again after setting the VCC7 bit in the VCP2 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.	
V _{por2}	Power-on reset valid voltage	20°C ± 10ppm/°C	—	—	V _{det}	V
tw(Vpor2)	Supply voltage rising time when power-on reset is canceled ²	20°C ± 10ppm/°C, tw(Vpor2) ≥ 10μs	—	—	100	ns

NOTES:

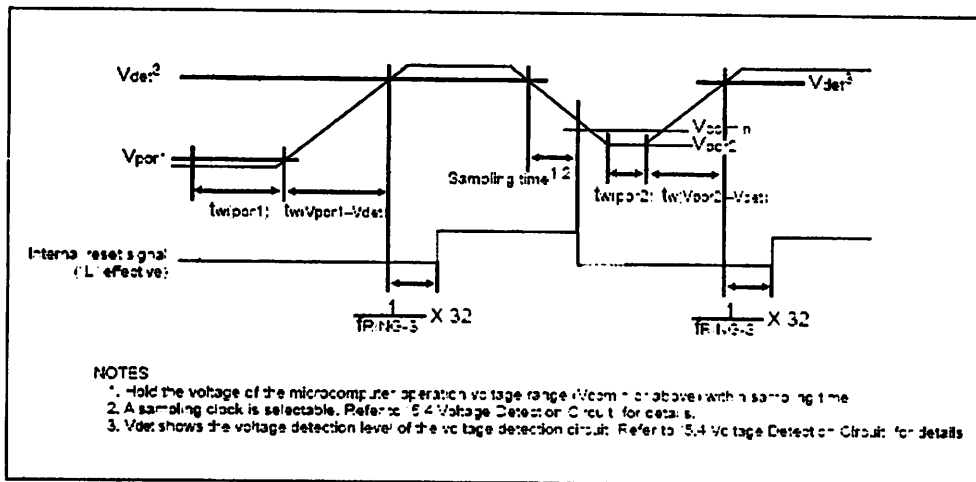
- The voltage detection circuit which is embedded in a microcomputer is a factor to generate the hardware reset 2. Refer to 5.1.2 Hardware Reset 2.
- This condition is not applicable when using V_{CC} ≥ 1.0V.
- When turning power on after the external power has been held below the valid voltage for greater than 10 seconds, refer to Table 16.2 Reset Circuit Electrical Characteristics (When Not Using Hardware Reset 2).
- tw(Vpor2) is time to hold the external power below effective voltage (V_{por2}).

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

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
V _{por}	Power-on reset valid voltage	20°C ± 10ppm/°C	—	—	2.1	V
tw(Vpor1-V _{det})	Supply voltage rising time when power-on reset is canceled	0°C ± 100ppm/°C, tw(Vpor1) ≥ 10μs	—	—	100	ns
tw(Vpor1-V _{det})	Supply voltage rising time when power-on reset is canceled	20°C ± 10ppm/°C, tw(Vpor1) ≥ 30μs	—	—	100	ns
tw(Vpor1-V _{det})	Supply voltage rising time when power-on reset is canceled	20°C ± 10ppm/°C, tw(Vpor1) ≥ 10μs	—	—	—	ns
tw(Vpor1-V _{det})	Supply voltage rising time when power-on reset is canceled	0°C ± 100ppm/°C, tw(Vpor1) ≥ 15μs	—	—	2.5	ns

NOTES:

- When not using hardware reset 2, use with V_{CC} ≥ 2.7V.
- tw(Vpor1) is time to hold the external power below effective voltage (V_{por1}).



NOTES:

- Hold the voltage of the microcomputer operation voltage range (V_{opmin}) or above with a sampling time.
- A sampling clock is selectable. Refer to 5.4 Voltage Detection Circuit for details.
- V_{det}² 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 (f _{osc} = R _{osc} × (1/R)) when the reset is released	V _{CC} =5.0V, Top=25°C Def: 4011116 = M1 register	—	±	—	Hz
to = R _{osc}	Default high-speed on-chip oscillator initial number	V _{CC} =5.0V, Top=25°C Def: 4011116 = M1 register	—	±	—	15
to = R _{osc}	High-speed on-chip oscillator period adjusted unit	Differences after setting 011 and 02111176 = R _{osc} register	—	—	—	15
—	High-speed on-chip oscillator temperature dependence 1	Frequency fluctuation in temperature range of -40 to +85°C	—	±	—	%
—	High-speed on-chip oscillator temperature dependence 2	Frequency fluctuation in temperature range of -40 to +85°C	—	±0	—	%

NOTES

1. The measuring condition is V_{CC}=AV_{CC}=5.0V and Top=25°C.

Table 5.10 Power Circuit Timing Characteristics

Symbol	Parameter	Measuring condition	Standard			Unit
			Min	Typ	Max	
to = P _{st}	Time to start power supply stabilization during power-on ¹				2000	μs
to = P _{st}	Power release time ²				100	μs

Note

1. The measuring condition is V_{CC}=AV_{CC}=2.7 to 5.5V and Top=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) [V_{CC}=5V]

Symbol	Parameter		Measuring condition	Standard			Unit
				Min	Typ	Max	
V _{OH}	I _{OH} output voltage	Except I _{OH}	I _{OH} =5mA	2.4	—	V _{OH}	V
		I _{OH} =200μA	—	—	V _{OH}	V	
V _{OL}	I _{OL} output voltage	Except I _{OL}	Drive capacity HIGH I _{OL} =15mA	0.4	—	V _{OL}	V
			Drive capacity LOW I _{OL} =500μA	0.1	—	V _{OL}	V
		I _{OL} = 10mA	Drive capacity HIGH I _{OL} =15mA	—	—	2.0	V
			Drive capacity LOW I _{OL} =500μA	—	—	0.45	V
		I _{OL} = 5mA	Drive capacity HIGH I _{OL} =15mA	—	—	2.0	V
			Drive capacity LOW I _{OL} =500μA	—	—	0.45	V
V _{IL}	Input voltage	I _{IL} = 1mA	Drive capacity HIGH I _{IL} =15mA	—	—	2.0	V
			Drive capacity LOW I _{IL} =500μA	—	—	0.4	V
I _{IN}	I _{IN} input current		0.0	—	0.0	μA	
I _{IN}	I _{IN} input current		0.0	—	5.0	μA	
R _{IN} (in)	Input resistance		—	—	8.0	kΩ	
R _{IN} (out)	Output resistance		10	25	150	Ω	
R _{IN} (in)	Input resistance (oscillator frequency)		—	10	—	MΩ	
V _{REF}	RAM reference voltage	At stop mode	2.0	2.1	2.2	V	

Note

1. Referenced to V_{CC}=AV_{CC}=4.0 to 5.5V; Top = +20 to +55°C, -40 to +55°C, 41 to 80V = 0, unless otherwise specified.

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

Symbol	Parameter	Measuring condition	Standard		Unit	
			Min.	Typ.		Max.
I _{CC}	Power supply current (V _{CC} =5.0 to 5.5V) in single-chip mode (the CPU, I/Os, DRAM, ROM and others is 300 kHz)	High-speed mode	Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode	2	4	mA
		1000/1000000 mode	Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode	4		mA
		High-speed mode	High-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode	3		mA
		High-speed mode	High-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode	2		mA
		High-speed mode	High-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode	4	8	mA
		High-speed mode	High-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode	10		mA
		Low-speed mode	High-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode	400	900	μA
		Wait mode	High-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode	40	80	μA
		Wait mode	High-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode	35	70	μA
		Wait mode	High-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode Low-speed source mode High-speed source mode	20	30	μA

NOTES
 1. The power supply current measuring is executed using the measuring program on measurement.
 2. The chip is operated with drive 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
tW-(XIN)	XIN input HIGH pulse width	25		ns
tWL(XIN)	XIN input LOW pulse width	25		ns

Table 5.14 CNTR0 input, CNTR1 input, INT2 input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tC(CNTRC)	CNTR0 input cycle time	100		ns
tW-(CNTRC)	CNTR0 input HIGH pulse width	40		ns
tWL(CNTRC)	CNTR0 input LOW pulse width	40		ns

Table 5.15 TCIN input, INT3 input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tC(TCIN)	TCIN input cycle time	400 ¹		ns
tW-(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(CLK)	CLKi input cycle time	200		ns
tW-(CLKH)	CLKi input HIGH pulse width	100		ns
tW-(CLKL)	CLKi input LOW pulse width	100		ns
tD(C-C)	TxD output delay time		80	ns
tH(C-C)	TxD hold time	0		ns
tS(D-C)	RxDi input setup time	35		ns
tH(D-C)	RxDi input hold time	90		ns

Table 5.17 External interrupt INT0 input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tW-(NH)	INT0 input HIGH pulse width	250 ¹		ns
tW-(NL)	INT0 input LOW pulse width	250 ²		ns

NOTES

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

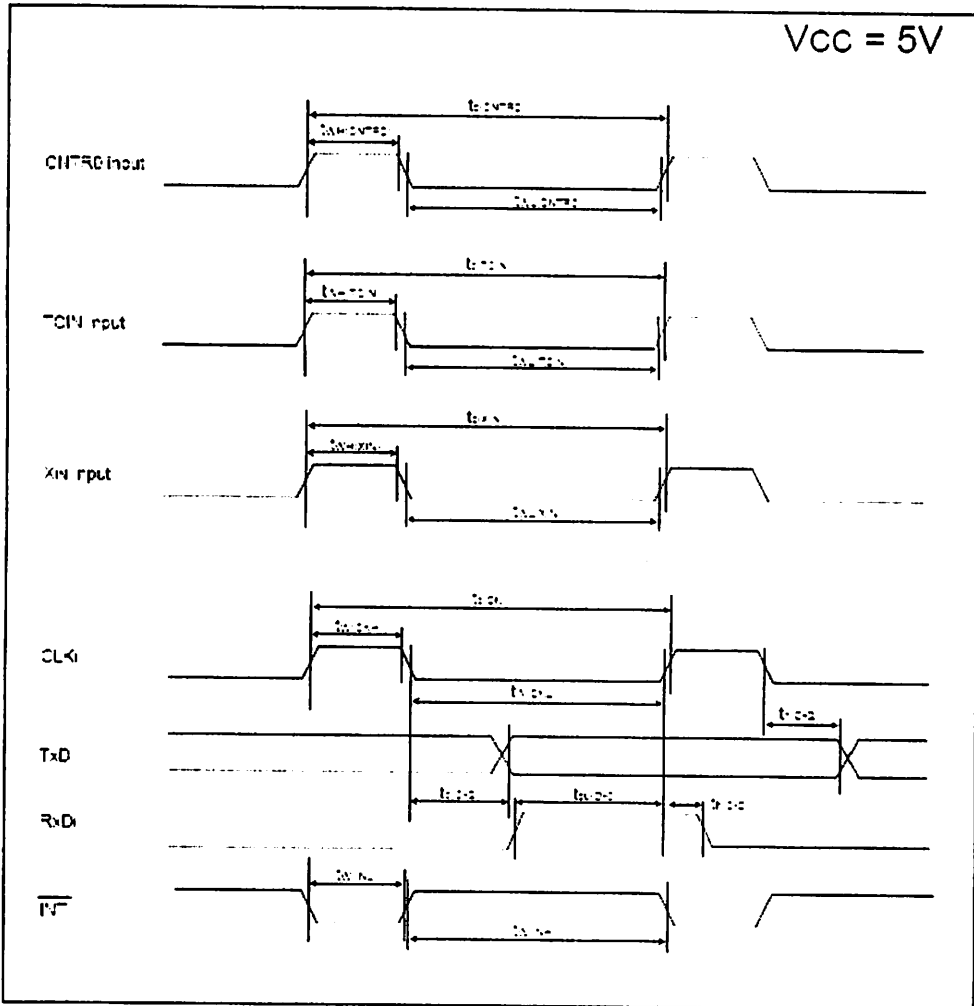


Figure 5.4 Vcc=5V timing diagram

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

Symbol	Parameter	Measuring condition	Standard			Unit		
			Min.	Typ.	Max.			
V _{OH}	Output voltage	Except I_{OL}	I _{OH} =1mA	2.4	—	V		
		I_{OL}	Drive capacity HIGH Drive capacity LOW	I _{OH} =20mA I _{OH} =50µA	2.0	—	V	
V _{OL}	Output voltage	Except I_{OH}	I _{OH} =1mA	—	—	0.8	V	
		Except I_{OL}	Drive capacity HIGH Drive capacity LOW	I _{OL} =20mA I _{OL} =1mA	—	—	0.8	V
		I_{OH}	Drive capacity HIGH Drive capacity LOW	I _{OH} =20mA I _{OH} =50µA	—	—	0.8	V
		I_{OL}	Drive capacity HIGH Drive capacity LOW	I _{OL} =20mA I _{OL} =50µA	—	—	0.8	V
V _{IL}	Input level	Except I_{IH}	V _{CC} - 0.1V, V _{CC} - 0.3V, V _{CC} - 0.5V, V _{CC} - 0.7V, V _{CC} - 0.9V, V _{CC} - 1.1V, V _{CC} - 1.3V, V _{CC} - 1.5V, V _{CC} - 1.7V, V _{CC} - 1.9V, V _{CC} - 2.1V, V _{CC} - 2.3V, V _{CC} - 2.5V, V _{CC} - 2.7V, V _{CC} - 2.9V, V _{CC} - 3.1V, V _{CC} - 3.3V	0.0	—	0.8	V	
		I_{IH}	0.0	—	—	0.8	V	
I _{OH}	Output current	V _{OH} =2V	—	—	40	µA		
I _{OL}	Output current	V _{OL} =0V	—	—	40	µA		
R _{OL}	Output resistance	V _{OH} =2V	16	160	500	Ω		
R _{IN}	Feedback resistance	V _{CC}	—	30	—	kΩ		
f _{MAX}	Maximum operating frequency		—	100	200	kHz		
V _{RAV}	RAV error voltage	All input pins	0.0	—	—	V		

Note 1: Referenced to Vcc=AVcc=0.7 to 3.3V at T_{OP} = -20 to 65 °C, C_{IN} = 40 to 25 pF, C_{OUT} = 10kΩ unless otherwise specified.

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

Symbol	Parameter	Measuring condition	Vin	Standard Typ	Max	Unit
Power supply current Icc=2.7 to 3.3V, single-chip mode, the output data is open and other pins are Vcc		High-speed mode	1. 100% duty square wave 2. High-speed output collector off 3. Load capacitance: 50 pF 4. Input signal: 1.5V, 100 ns 5. Input signal: 1.5V, 100 ns 6. Input signal: 1.5V, 100 ns 7. Input signal: 1.5V, 100 ns 8. Input signal: 1.5V, 100 ns 9. Input signal: 1.5V, 100 ns 10. Input signal: 1.5V, 100 ns 11. Input signal: 1.5V, 100 ns 12. Input signal: 1.5V, 100 ns	0	3	mA
		Medi-High-speed mode	1. 100% duty square wave 2. High-speed output collector off 3. Load capacitance: 50 pF 4. Input signal: 1.5V, 100 ns 5. Input signal: 1.5V, 100 ns 6. Input signal: 1.5V, 100 ns 7. Input signal: 1.5V, 100 ns 8. Input signal: 1.5V, 100 ns 9. Input signal: 1.5V, 100 ns 10. Input signal: 1.5V, 100 ns 11. Input signal: 1.5V, 100 ns 12. Input signal: 1.5V, 100 ns	0	4	mA
		Low-speed mode	1. 100% duty square wave 2. High-speed output collector off 3. Load capacitance: 50 pF 4. Input signal: 1.5V, 100 ns 5. Input signal: 1.5V, 100 ns 6. Input signal: 1.5V, 100 ns 7. Input signal: 1.5V, 100 ns 8. Input signal: 1.5V, 100 ns 9. Input signal: 1.5V, 100 ns 10. Input signal: 1.5V, 100 ns 11. Input signal: 1.5V, 100 ns 12. Input signal: 1.5V, 100 ns	0	1.5	mA
		High-speed dynamic operation mode	1. 100% duty square wave 2. High-speed output collector off 3. Load capacitance: 50 pF 4. Input signal: 1.5V, 100 ns 5. Input signal: 1.5V, 100 ns 6. Input signal: 1.5V, 100 ns 7. Input signal: 1.5V, 100 ns 8. Input signal: 1.5V, 100 ns 9. Input signal: 1.5V, 100 ns 10. Input signal: 1.5V, 100 ns 11. Input signal: 1.5V, 100 ns 12. Input signal: 1.5V, 100 ns	0	400	μA
		Wait mode	1. High speed off 2. High-speed output collector off 3. Load capacitance: 50 pF 4. Input signal: 1.5V, 100 ns 5. Input signal: 1.5V, 100 ns 6. Input signal: 1.5V, 100 ns 7. Input signal: 1.5V, 100 ns 8. Input signal: 1.5V, 100 ns 9. Input signal: 1.5V, 100 ns 10. Input signal: 1.5V, 100 ns 11. Input signal: 1.5V, 100 ns 12. Input signal: 1.5V, 100 ns	0	17	μA
		Wait mode	1. High speed off 2. High-speed output collector off 3. Load capacitance: 50 pF 4. Input signal: 1.5V, 100 ns 5. Input signal: 1.5V, 100 ns 6. Input signal: 1.5V, 100 ns 7. Input signal: 1.5V, 100 ns 8. Input signal: 1.5V, 100 ns 9. Input signal: 1.5V, 100 ns 10. Input signal: 1.5V, 100 ns 11. Input signal: 1.5V, 100 ns 12. Input signal: 1.5V, 100 ns	0	15	μA
		Stop mode	1. High speed off 2. High-speed output collector off 3. Load capacitance: 50 pF 4. Input signal: 1.5V, 100 ns 5. Input signal: 1.5V, 100 ns 6. Input signal: 1.5V, 100 ns 7. Input signal: 1.5V, 100 ns 8. Input signal: 1.5V, 100 ns 9. Input signal: 1.5V, 100 ns 10. Input signal: 1.5V, 100 ns 11. Input signal: 1.5V, 100 ns 12. Input signal: 1.5V, 100 ns	0	27	μA

NOTES
 1. The power supply current measuring is executed using the measuring program on flash memory.
 2. The Vcc 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, 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, INT3 input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tC(TCIN)	TCIN input cycle time	1200		ns
tWH(TCIN)	TCIN input HIGH pulse width	600		ns
tWL(TCIN)	TCIN input LOW pulse width	600		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(CLK)	CLKi input cycle time	300		ns
tW(CLK-)	CLKi input HIGH pulse width	150		ns
tW(CLK)	CLKi input LOW pulse width	150		ns
tD(C-Q)	TxDi output delay time		160	ns
tH(C-Q)	TxDi hold time	3		ns
tSU(D-C)	RxDi input setup time	55		ns
tH(C-D)	RxDi input hold time	90		ns

Table 5.24 External interrupt INT0 input

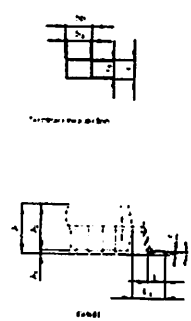
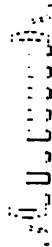
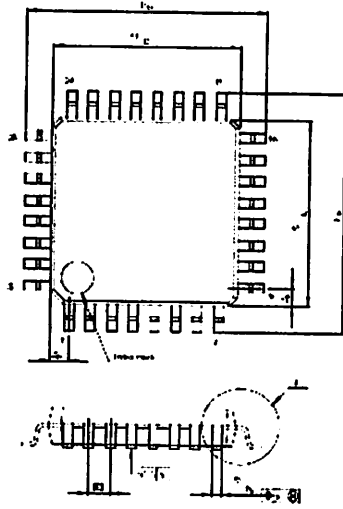
Symbol	Parameter	Standard		Unit
		Min.	Max.	
tW(IN-)	INT0 input HIGH pulse width	380		ns
tW(INL)	INT0 input LOW pulse width	380		ns

NOTES

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

Package Dimensions

JATA Package Code P48F30-14-000	PLAL SIG Code P48F30C00A	Pinout Code NPN-AB	MANUFACT P-1
------------------------------------	-----------------------------	-----------------------	-----------------



010
000-0111-0112
010-0111-0113
010-0111-0114
010-0111-0115

Pin No.	Symbol	Pin No.	Symbol
1	01	14	01
2	02	15	02
3	03	16	03
4	04	17	04
5	05	18	05
6	06	19	06
7	07	20	07
8	08	21	08
9	09	22	09
10	10	23	10
11	11	24	11
12	12	25	12
13	13	26	13
14	14	27	14
15	15	28	15
16	16	29	16
17	17	30	17
18	18	31	18
19	19	32	19
20	20	33	20
21	21	34	21
22	22	35	22
23	23	36	23
24	24	37	24
25	25	38	25
26	26	39	26
27	27	40	27
28	28	41	28
29	29	42	29
30	30	43	30
31	31	44	31
32	32	45	32
33	33	46	33
34	34	47	34
35	35	48	35
36	36	49	36
37	37	50	37
38	38	51	38
39	39	52	39
40	40	53	40
41	41	54	41
42	42	55	42
43	43	56	43
44	44	57	44
45	45	58	45
46	46	59	46
47	47	60	47
48	48	61	48
49	49	62	49
50	50	63	50
51	51	64	51
52	52	65	52
53	53	66	53
54	54	67	54
55	55	68	55
56	56	69	56
57	57	70	57
58	58	71	58
59	59	72	59
60	60	73	60
61	61	74	61
62	62	75	62
63	63	76	63
64	64	77	64
65	65	78	65
66	66	79	66
67	67	80	67
68	68	81	68
69	69	82	69
70	70	83	70
71	71	84	71
72	72	85	72
73	73	86	73
74	74	87	74
75	75	88	75
76	76	89	76
77	77	90	77
78	78	91	78
79	79	92	79
80	80	93	80
81	81	94	81
82	82	95	82
83	83	96	83
84	84	97	84
85	85	98	85
86	86	99	86
87	87	100	87

LAMPIRAN REVISI

1. Mekanisme komunikasi antara RFID *reader* dengan mikrokontroler Renesas:
Menggunakan komunikasi serial UART (*Universal Asynchronous Receiver Transmitter*) dengan kecepatan 9600 *bps*.

Output dari RFID reader berupa format ASCII.

Output yang memiliki format ASCII memiliki struktur sebagai berikut:

0 2	10 data karakter ASCII	checksum	CR	LF	0 3
-----	------------------------	----------	----	----	-----

Checksum merupakan hasil EXOR (*Exclusive OR*) dari 5 biner data byte.

Misalnya data output serial (dalam hexadesimal) yang kita tangkap adalah sebagai berikut:

02	30	34	36	32	30	31	44	37	36	43	44	43	0D	0A	03
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Langkah pertama adalah merubah semua nilai data diatas menjadi karakter ASCII. Misalnya 30H menjadi karakter "0", 34H menjadi karakter "4", dst. Langkah kedua adalah menyusun data – data tersebut ke dalam Format Data ASCII seperti tabel 1. Kemudian ambil 10 data karakter ASCII. Dalam contoh ini berarti data tersebut adalah:

30	34	36	32	30	31	44	37	36	43	Data Hexsa
		6	2	0	1	D	7	6	C	Data ASCII

Untuk data dengan angka 30 dan 34 merupakan data untuk jenis-jenis kartu dan tidak digunakan dalam proses konversi, yang akan dipakai disini

adalah data yang ke 3 s/d 10. Hasil konversi dari data heksa ke dalam data ASCII adalah "6201D76C". Gabungkan karakter data ASCII menjadi bilangan Hexadesimal, kemudian konversikan bilangan hexadesimal tsb ke dalam desimal. Hasilnya sebagai berikut: 6201D76C H menjadi 1644287852. angka-angka ini merupakan nomor kartu sebenarnya yang tertera pada badan kartu yang biasa disebut tag RFID.

2. Kecepatan transmisi data antara RFID reader ke Mikrokontroller Renesas:

Sesuai dengan ketentuan (*Datasheet*) dari RFID reader menggunakan 9600 *bps*.

Kecepatan pengiriman data (*baudrate*) dan fase clock pada sisi transmitter dan pada sisi receiver harus sinkron. Untuk itu diperlukan sinkronisasi antara transmitter dan receiver, maka Mikrokontroller Renesas juga harus diset dengan jalan mengisi register u0brg/u1brg, dengan rumus:

$$\frac{f_{EXT}}{Bitrate \times 16} - 1$$
$$\frac{11MHz}{9600 \times 16} - 1 = \frac{11000000}{153600} - 1 = 70,614 \approx 71$$

Bitrate diisi sesuai dengan *boudrate* yang diinginkan. Hasilnya berupa suatu nilai yang harus dibulatkan sebelum dimasukkan ke register u0brg/u1brg.

3. Jarak terjauh yang dapat dijangkau oleh komunikasi RFID *reader* ke Mikrokontroller Renesas adalah 5 meter karena pengirimannya berupa tegangan TTL (*Transisitor-Transistor Logic*) sehingga apabila lebih dari 5 meter data akan error yang disebabkan oleh rugi-rugi tegangan.

4. Dari hasil pengukuran Arus (1) total yang diperlukan sistem saat beroperasi sebesar 85 mA

5. Impedansi dari sistem:

$$V = 5 \text{ Volt} \quad ; I = 85 \text{ mA}$$

$$Z = \frac{V}{I} = \frac{5}{85 \times 10^{-3}}$$

$$= 58,82 \Omega$$