SKRIPSI

PERANCANGAN DAN PEMBUATAN SISTEM SERVER UNTUK KWH METER PRABAYAR PADA RUMAH SUSUN YANG MENGGUNAKAN JALA-JALA LISTRIK SEBAGAI MEDIA KOMUNIKASI DATA

Disusun Oleh :

SLAMET SETIYABUDI

NIM : 02.17.084

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

2008

SKRIPSI

PERANCANGAN DAN PEMBUATAN SISTEM [illegible] UNTUK KWH METER PRABAYAR PADA RUMAH SUSUN YANG MENGGUNAKAN JALA-JALA LISTRIK SEBAGAI MEDIA KOMUNIKASI DATA

[illegible]
[illegible]
NRP : [illegible]

JURUSAN TEKNIK ELEKTRO D-3
[illegible] ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
[illegible]

LEMBAR PERSETUJUAN

PERANCANGAN DAN PEMBUATAN SISTEM SERVER UNTUK KWH METER PRABAYAR PADA RUMAH SUSUN YANG MENGGUNAKAN JALA-JALA LISTRIK SEBAGAI MEDIA KOMUNIKASI DATA

SKRIPSI

Disusun dan Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik Program Studi Teknik Elektronika

Disusun Oleh :

SLAMET SETIYABUDI
02.17.084

Diperiksa dan Disetujui

Dosen Pembimbing I	Dosen Pembimbing II
Ir. Widodo Pudji M, MT	**Ir. Yusuf Ismail Nakhoda, MT**
NIP.Y 1028700171	**NIP. Y 1018800189**

Mengetahui
Ketua Jurusan Teknik Elektro S-1

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

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2008

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

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama	: Slamet Setiyabudi
NIM	: 02.17.084
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik Elektronika
Masa Bimbingan	: 14 Juni 2008 – 14 Desember 2008
Judul Skripsi	: Perancangan dan Pembuatan Sistem Server untuk kWh Meter Prabayar pada Rumah Susun yang Menggunakan Jala-Jala listrik sebagai Media Komunikasi Data

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

Hari	: Jum'at
Tanggal	: 26 September 2008
Nilai	: 85,25 (A)

Panitia Majelis Penguji

(Ir. Mochtar Asroni, MSME)	(Ir. F. Yudi Limpraptono, MT)
NIP.Y. 1018100036	NIP.Y. 1039500274

Anggota Penguji

(Ir. Mimien Mustikawati, MT)	(M. Ibrahim Ashari, ST, MT)
NIP. P. 1030000352	NIP. P. 1030100358

ABSTRAKSI

PERANCANGAN DAN PEMBUATAN SISTEM SERVER UNTUK KWH METER RPABAYAR PADA RUMAH SUSUN YANG MENGGUNAKAN JALA-JALA LISTRIK SEBAGAI MEDIA KOMUNIKASI DATA

(Slamet Setiyabudi, 02.17.084, Teknik Elektro S-1/Elektronika)
(Dosen Pembimbing I: Ir. Widodo Pudji M, MT)
(Dosen Pembimbing II: Ir. Yusuf Ismail Nakhoda, MT)

Sistem listrik prabayar telah diujicobakan di beberapa kota besar di Indonesia dengan cukup berhasil. Untuk dapat menggunakan layanan baru ini dibutuhkan kWh meter digital prabayar dengan *Smart Card.* Untuk pengembangan ke sistem otomatisasi pengisian ulang pulsa dan pembayaran, akan sulit jika menggunakan sistem kartu tersebut karena tidak terjadi komunikasi antara kartu tersebut dengan tempat pengisisan pulsa. Dalam studi kasus pengelolaan energi listrik pada rumah susun dengan sistem prabayar dapat memanfaatkan jala-jala listrik yang telah tersedia sebagai media komunikasinya yang disebut sebagai sistem komunikasi *Power Line Carrier(PLC).*

Dalam penerapannya dibutuhkan sistem *server* yang dapat berkomunikasi melalui jala-jala listrik dengan kWh meter prabayar rumah susun. Bagian penyusunnya terdiri dari rangkaian konverter level tegangan RS232-TTL, mikrokontroler, rangkaian selektor dan *Modem Power Line Carrier(PLC).* Untuk pemrogramannya dirancang agar dapat digunakan untuk melayani proses pengisian pulsa, pengecekan status dan pengendalian aliran listrik *(on/off)* ke pelanggan.

Dengan *Chip Modem PLC* LM1893 dapat dihasilkan rangkaian *Modem PLC* yang menggunakan teknik modulasi FSK dan bekerja cukup baik pada frekuensi *carrier* 125kHz. Dari pengujian keseluruhan sistem pada *server*, komunikasi dapat dilakukan dengan baik melalui jala-jala listrik dan *server* dapat melayani pelanggan listrik rumah susun dengan sistem kWh meter prabayar.

Kata Kunci: Komunikasi, *Modem PLC*, kWh Meter Prabayar

KATA PENGANTAR

Dengan memanjatkan rasa syukur kehadirat Allah, Tuhan Tuhan Yang Maha Pengasih dan Maha Penyayang yang telah mencurahkan ni'mat, petunjuk, pertolongan dan ilmu-Nya sehingga penyusun dapat menyelesaikan laporan Skripsi yang berjudul "Perancangan dan Pembuatan Sistem Server untuk kWh Meter Prabayar pada Rumah Susun yang Menggunakan Jala-jala Listrik sebagai media Komunikasi data" dengan baik.

Laporan Skripsi ini disusun sebagai syarat untuk mendapatkan gelar Sarjana Teknik (ST) dan sebagai bentuk dedikasi profesi mahasiswa teknik elektro.

Dalam kesempatan ini penyusun mengucapkan terima kasih kepada :

1. Bapak, ibu, adik dan seluruh keluarga tercinta yang senantiasa memberikan dukungan, kasih sayang dan pengorbanan yang berharga.
2. Bapak Dr.Ir.Abraham Lomi, MSEE, selaku Rektor Institut Teknologi Nasional Malang.
3. Bapak Ir. F. Yudi Limpraptono, MT selaku Dosen Pembimbing dan Kajur Teknik Elektro S-1.
4. Bapak Ir. Widodo Pudji M, MT selaku Dosen Pembimbing I.
5. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Dosen Pembimbing II.
6. Seluruh Staf Jurusan Teknik Elektro S-1 yang telah banyak membantu.
7. Saudara-saudara mahasiswa yang telah banyak membantu memberikan dorongan dan kerjasamanya.
8. Dan semua pihak yang telah turut membantu dalam penyelesaian skripsi ini.

Penyusun berharap semoga laporan Skripsi ini dapat bermanfaat. Kritik dan saran yang membangun akan sangat membantu dalam penyempurnaan laporan ini.

Malang, September 2008

Penyusun

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1. LATAR BELAKANG

Keberadaan rumah susun (rusun) di kota-kota besar di Indonesia yang diharapkan dapat membantu masyarakat menengah bawah dalam mendapatkan tempat tinggal yang layak, ternyata memunculkan permasalahan lain seperti dilaporkan adanya penunggakan pembayaran listrik 1,2 milyar pada rusun Penjaringan, Jakarta Utara (Ibnu Rusyidi/Tempointeraktif, 2005). Untuk mengantisipasi hal yang sama terjadi dimasa yang akan datang dapat digunakan pelayanan listrik prabayar sebagaimana dilakukan PT. Sarana Pembangunan Palembang Jaya (SP2J) yang menawarkan pengembangan Rumah Susun Sederhana Sewa (Rusunawa) di Jl. Kasnariansyah Palembang dengan mengembangkan konsep listrik prabayar menggunakan kartu (Siera Syailendra/SINDO, 2008).

Mengingat krisis energi listrik yang terjadi di Indonesia saat ini, diperlukan manajemen baru bagi PLN sebagai pengelola tunggal sektor ketenagalistrikan. Dengan sistem listrik prabayar tampaknya akan lebih banyak memberi keuntungan bagi PT. PLN. Alasan yang pertama, PLN akan mendapatkan suntikan dana segar lebih cepat, karena konsumen membayar dimuka (prinsip prabayar adalah "bayar dulu, baru pake"). Kedua, sistem prabayar ini bisa menjadi instrumen untuk menurunkan losses non-teknis, yaitu menekan pencurian listrik. Pasalnya, konsumen tidak lagi bisa mengutak-atik kWh meter

atau sambungan langsung ke tiang listrik. Ketiga, penghematan sumber daya manusia, khususnya petugas pencatat meter. Keempat, manfaat utama yang diperoleh adalah peningkatan citra pelayanan karena tidak akan ada lagi masalah kesalahan baca meter yang sering menjadi keluhan pelanggan saat ini. (Tulus Abadi/Koran Tempo, 2007). Sedangkan keuntungan atau nilai tambah yang dapat diperoleh bagi konsumen dengan sistem prabayar ini diantaranya tidak diwajibkan membayar uang jaminan langganan (UJL), terhindar dari kesalahan pembacaan meter dan terhindar dari ancaman pemutusan sementara, pelanggan juga dapat menggunakan listrik tanpa batasan daya, tidak ada ketentuan peruntukan listrik bagi rumah tangga, bisnis atau lainnya serta tidak perlu membayar rekening bulanan (Dar/kapanlagi.com, 2005). Dengan sistem prabayar ini pelanggan akan terlatih lebih berhemat dengan mengontrol sendiri pemakaian energi listrik.

Namun demikian pada sistem prabayar dengan menggunakan kartu masih cukup merepotkan bagi pelanggan karena kartu harus dibawa ke loket pengisian ulang yang jumlah dan waktu operasionalnya yang terbatas. Dan Agar sistem prabayar dapat dikembangkan lebih lanjut untuk otomatisasi pengisian ulang seperti pengisian melalui jaringan internet (*Internet-Banking*), jaringan telepon seluler (*Mobile-banking* atau *SMS-Banking*) atau jaringan ATM maka diperlukan bentuk sistem pengisian ulang secara *online*. Artinya antara pelanggan prabayar dan tempat pengisian pulsa prabayar terjadi komunikasi secara langsung. Karena konsumen yang dilayani adalah pelanggan listrik maka komunikasi yang dilakukan dapat memanfaatkan jaringan listrik ke pelanggan. Komunikasi dengan media jaringan listrik ini biasa disebut dengan PLC (*Power Line Communication/Carrier*).

Pada Skripsi ini diambil studi kasus tentang pengelolaan energi listrik pada rumah susun dengan memanfaatkan sistem prabayar. Karena rumah susun biasanya dibangun dalam area tertentu dengan beberapa blok, maka pelayanan listrik sistem prabayar memanfaatkan jaringan listrik untuk pengisian ulang pulsa prabayar akan lebih mudah untuk direalisasikan.

1.2. TUJUAN

Merancang dan membuat *prototype* sistem *server* untuk pelanggan listrik prabayar pada rumah susun yang menggunakan jala-jala listrik sebagai media komunikasi data.

1.3. RUMUSAN MASALAH

1. Bagaimana merancang dan membuat sistem *server* sehingga dapat berkomunikasi dengan kWh meter prabayar melalui jala-jala listrik.
2. Bagaimana merancang dan membuat perangkat lunak untuk mengatur kerja sistem dan untuk melayani pelanggan prabayar pada rumah susun.

1.4. BATASAN MASALAH

1. Komunikasi dilakukan dengan menggunakan jala-jala listrik 220V pada rumah susun pelanggan 1 fasa.
2. Modem PLC (*Power Line Carrier*) yang digunakan berbasis *chip* PLC dengan tipe LM1893.
3. Tidak membahas analisis pengaruh luar yang dapat mengganggu pada media komunikasi data.

4. Permasalahan yang dibahas tidak mencakup keamanan komunikasi data, pengaruh jenis atau panjang penghantar.
5. Parameter keberhasilan perancangan dan pembuatan sistem adalah dapat berkomunikasinya *server* dengan kWh meter prabayar.

1.5. METODE PENELITIAN

1. Studi Literatur

Yaitu dengan mempelajari berbagai materi yang berkaitan dengan alat yang akan dirancang baik dari diktat kuliah, buku-buku referensi, laporan skripsi/tugas akhir, situs internet maupun sumber lain yang memungkinkan.

2. Perancangan Alat

Yaitu membuat perencanaan alat yang akan dibuat, baik untuk perangkat keras maupun perangkat lunak. Perencanaan perangkat keras dapat berupa model rangkaian yang akan digunakan dan pemilihan komponen atau bahan yang sesuai. Untuk perencanaan perangkat lunak yaitu dengan membuat diagram alir (*flowchart*) untuk proses yang akan akan dijalankan pada alat. Perencanaan perangkat lunak sebagai acuan untuk pemrograman pada mikrokontroller dan PC.

3. Pembuatan Alat

Yaitu dengan membuat alat sesuai dengan perencanaan yang terdiri dari perangkat keras dan perangkat lunak.

4. Analisis dan Pengujian

Adalah dengan melakukan pengamatan terhadap kinerja alat, pengukuran besaran listrik terhadap parameter yang ditentukan dan melakukan

pengujian terhadap subsistem maupun sistem secara keseluruhan untuk mengevaluasi hasil yang didapatkan.

5. Kesimpulan

Adalah pengambilan kesimpulan berdasarkan hasil analisa dan pengujian.

1.6. SISTEMATIKA PENULISAN

Sistematika penulisan dari laporan skripsi ini adalah sebagai berikut:

Bab I : Pendahuluan

Memuat latar belakang, tujuan, rumusan masalah, batasan masalah, metode penelitian dan sistematika penulisan.

Bab II : Teori Dasar

Membahas teori-teori dasar penunjang perancangan dan pembuatan alat.

Bab III: Perancangan dan Pembuatan Alat

Memuat prinsip kerja, perancangan dan pembuatan alat.

Bab IV: Pengujian dan Analisis Alat

Memuat hasil pengujian dan analisis alat atau sistem yang telah dibuat.

Bab V : Kesimpulan dan Saran

Berisi kesimpulan dan saran dari skripsi ini.

BAB II

TEORI DASAR

2.1. Rumah Susun

Dalam undang-undang no. 16 tahun 1985 tentang rumah susun dijelaskan, bahwa rumah susun (rusun) adalah bangunan gedung bertingkat yang dibangun dalam suatu lingkungan, yang terbagi dalam bagian-bagian yang distrukturkan secara fungsional dalam arah horizontal maupun vertikal dan merupakan satuan-satuan yang masing-masing dapat dimiliki dan digunakan secara terpisah, terutama untuk tempat hunian, yang dilengkapi dengan bagian-bersama, benda-bersama dan tanah-bersama.

Pembangunan Rusun berlandaskan pada azas kesejahteraan umum, keadilan dan pemerataan, serta keserasian dan keseimbangan dalam perikehidupan. Tujuan pertama dari pembangunan Rusun adalah untuk memenuhi kebutuhan perumahan yang layak bagi rakyat, terutama golongan masyarakat yang berpenghasilan rendah, yang menjamin kepastian hukum dalam pemanfaatannya. Yang kedua adalah meningkatkan daya guna dan hasil guna tanah di daerah perkotaan dengan memperhatikan kelestarian sumber daya alam dan menciptakan lingkungan pemukiman yang lengkap, serasi, dan seimbang. Serta memenuhi kebutuhan untuk kepentingan lainnya yang berguna bagi kehidupan masyarakat.

Fasilitas yang disediakan untuk rumah susun sesuai dengan yang diamanatkan undang-undang haruslah memadai antara lain mencakup : jaringan

air bersih, jaringan listrik, jaringan gas, saluran pembuangan air hujan, saluran pembuangan air limbah, saluran dan tempat pembuangan sampah, tempat untuk kemungkinan pemasangan jaringan telepon dan alat komunikasi lainnya, alat transportasi yang berupa tangga, *lift* atau eskalator sesuai dengan tingkat keperluan, pintu dan tangga darurat kebakaran, tempat jemuran, alat pemadam kebakaran, penangkal petir, alat/sistem alarm, pintu kedap asap pada jarak-jarak tertentu serta generator listrik yang disediakan untuk rumah susun yang menggunakan lift.

Saat ini di berbagai kota besar di Indonesia telah banyak didirikan Rusun sederhana yang dapat dihuni dengan hak sewa maupun hak milik. Rumah susun sederhana yang dihuni dengan hak sewa biasa disebut dengan Rusunawa dan Rumah susun sederhana yang dihuni dengan hak milik biasa disebut dengan Rusunami.

Bagi penghuni rusunawa, mereka harus membayar biaya sewa bulanan atau tahunan. Sedangkan untuk rusunami, pemilik diberikan kemudahan dengan cara mencicil pembayaran. Disamping itu, mereka juga harus membayar tagihan listrik dan air tiap bulan yang besarnya sesuai dengan pemakaian.

Dalam perkembangannya muncul permasalahan dalam hal pembayaran seperti disampaikan oleh Unit Pengelola Terpadu (UPT) Pengelola Rusun Dinas Perumahan DKI Jakarta bahwa tunggakan listrik dan air penghuni Rusun di DKI mencapai Rp. 4 Milyar (Aji/2007). Agar tidak terjadi hal yang sama di masa mendatang, seharusnya Pengelola Rusun membuat peraturan yang tegas dan pengelolaan/manajemen yang lebih baik.

Untuk meningkatkan pelayanan khususnya untuk kebutuhan energi listrik kepada penghuni Rusun dan untuk memperbaiki pengelolaan, munculah ide untuk menerapkan sistem listrik prabayar sebagaimana yang diutarakan Direktur Sarana Pembangunan Palembang Jaya (SP2J), Bahder Johan yang telah melakukan kontak dengan pihak PLN cabang Palembang untuk mendapatkan pelayanan listrik sistem prabayar di setiap unit Rusunawa (mg17/2008).

2.2. Listrik Prabayar

Sampai saat ini masyarakat sudah terbiasa menikmati layanan listrik dari PLN dengan sistem pascabayar. Dengan sistem ini pelanggan harus membayar tagihan listrik tiap bulan sesuai dengan jumlah energi yang digunakan dan daya yang terpasang. Sistem ini sebenarnya menguntungkan bagi pelanggan karena pembayarannya dilakukan diakhir pemakaian listrik. Namun demikian tidak dipungkiri bahwa seringkali terdapat laporan adanya kesalahan pembacaan meter oleh petugas sehingga membuat pelanggan kaget dengan tagihan listriknya yang membengkak.

Listrik prabayar merupakan bentuk layanan baru dari PLN yang ditujukan bagi daerah yang potensial untuk diterapkannya sistem ini. Perubahan sistem pembayaran ini dilatarbelakangi oleh sebuah paradigma dari era *product driven*--produk dibuat tanpa memikirkan kemauan konsumen--menjadi *costumer driven*--produsen menciptakan produk untuk memberikan kepuasan kepada konsumen (inforial PT PLN, koran tempo, 9 April 2007).

Sistem listrik prabayar ini sebenarnya sudah diujicobakan di beberapa kota di Indonesia, seperti Denpasar, Pontianak serta Tarakan, dan konon berjalan

sukses (Tulus Abadi, reformasihukum.org, 1 Mei 2007). Sebagai produk baru, diharapkan sistem ini dapat memberikan lebih banyak keuntungan atau nilai tambah baik bagi PLN maupun bagi pelanggan listrik.

Beberapa keuntungan yang dapat diperoleh bagi produsen atau PLN diantaranya yang pertama, PLN akan mendapatkan suntikan dana segar lebih cepat karena konsumen membayar dimuka. Kedua, sistem prabayar ini dapat menjadi instrumen untuk menurunkan losses non-teknis, yaitu menekan pencurian listrik. Pasalnya, konsumen tidak lagi bisa mengutak-atik kWh meter atau sambungan langsung ke tiang listrik. Ketiga, penghematan sumber daya manusia, khususnya petugas pencatat meter (Tulus Abadi, reformasihukum.org, 1 Mei 2007). Keempat, manfaat utama yang diperoleh adalah peningkatan citra pelayanan karena tidak ada lagi masalah kesalahan pembacaan meter yang sering menjadi keluhan pelanggan saat ini (Dar, kapanlagi.com, 7 April 2005).

Bagi konsumen ada beberapa keuntungan atau nilai tambah yang diperoleh dengan sistem prabayar diantaranya yang pertama, tidak diwajibkan membayar uang jaminan langganan (UJL). Kedua terhindar dari resiko kesalahan pembacaan meter. Ketiga, terhindar dari ancaman pemutusan sementara. Keempat, pelanggan dapat menggunakan listrik tanpa batasan daya, karena tidak ada ketentuan peruntukan listrik bagi rumah tangga, bisnis atau lainnya. Kelima, tidak perlu membayar rekening bulanan (Dar, kapanlagi.com, 7 April 2005). Dan yang keenam tidak menanggung biaya beban/abonemen seperti pada sistem pascabayar.

Untuk dapat menikmati layanan dari sistem yang baru ini pelanggan harus menggunakan kWh meter prabayar dengan kartu atau voucher. Kartu tersebut berguna untuk menyimpan data energi listrik (kWh) yang besarnya sesuai dengan

besaran nilai yang dibeli misalnya paket 300 kWh, 400 kWh dan 500 kWh. Berikut adalah bentuk dari kWh meter prabayar yang digunakan saat ini.

Gambar 2.1 KWh meter prabayar dengan *Smart Card* [1]

Untuk pengembangan ke arah otomatisasi pengisian pulsa tanpa menggunakan voucher atau secara elektronik sebagaimana yang sudah diterapkan pada sistem telepon seluler kemungkinan bisa dilakukan jika ada komunikasi langsung antar kWh meter prabayar dengan tempat pengisian pulsa. Dan hal ini tidak bisa bila diterapkan pada kWh meter prabayar *Smart Card* yang ada saat ini karena tidak dilengkapi perangkat pendukung komunikasi dengan tempat isi ulang pulsa.

Sistem komunikasi yang tepat untuk digunakan melayani pelanggan listrik adalah dengan memanfaatkan infrastruktur jaringan listrik yang sudah tersedia sebagai media komunikasi data. Komunikasi dengan menggunakan media jala-jala listrik biasa disebut *Power line Carrier Communication.*

2.3. *Power Line Carrier (PLC)*

Singkatan PLC berasal dari istilah asli *Power Line Carrier*. Akhir-akhir ini PLC juga diartikan sebagai *Power Line Communication*. Konsep dasar dari PLC adalah mengirimkan informasi dan listrik secara serempak (*simultaneously*) sepanjang saluran listrik sebagai alternatif pada pembangunan infrastruktur yang berguna dalam komunikasi. Secara umum, saluran listrik atau jala-jala listrik dibuat dari material penghantar yang sama sebagaimana yang digunakan dalam telekomuniksi. [6]

Pada dasarnya, PLC menawarkan penggunakan jaringan listrik untuk membawa sinyal komunikasi antara penyedia layanan (Tempat isi ulang pulsa listrik prabayar, *Internet Provider*, dll) dan pelanggan. Prinsip dibalik setiap solusi PLC adalah dengan memodulasi frekuensi pembawa (*carrier frequency*) diatas frekuensi 50/60 Hertz pada saluran listrik. Frekuensi dan pengkodean algoritma yang digunakan ini kemudian menentukan jumlah yang sebenarnya dari data yang dapat dibawa.

Data atau sinyal yang ditransmisikan dapat berupa sinyal analog atau digital. Untuk sinyal analog dapat langsung ditumpangkan ke frekuensi pembawa untuk ditransmisikan ke jaringan jala-jala listrik. Sedangkan untuk sinyal digital perlu dilakukan modulasi sebelum ditumpangkan ke frekuensi pembawa untuk dipancarkan melalui jala-jala listrik. Modulasi untuk sinyal digital dapat dengan teknik modulasi FSK (*Frequency Shift Keying*), ASK (*Amplitude Shift Keying*) dan PSK(*Phase Shift Keying*).

Sinyal yang telah termodulasi dapat ditumpangkan pada frekuensi pembawa dan dihubungkan dengan jala-jala listrik melalui 2 metode, yaitu:

1. PLC terhubung induktif, yaitu menempatkan penghantar yang berfungsi sebagai antena dengan arah paralel terhadap penghantar jaringan tenaga listrik untuk jarak tetentu.
2. PLC terhubung kapasitif, yaitu menghubungkan peralatan komunikasi dengan jaringan listrik lewat kapasitor.

Metode yang sering digunakan adalah PLC terhubung kapasitif. Fungsi kapasitor adalah sebagai penahan arus AC dari jala-jala listrik yang memiliki frekuensi berbeda dengan frekuensi pembawa dari peralatan komunikasi. Impedansi kapasitif ditentukan oleh persamaan 2-1.

$$X_c = \frac{1}{2\pi \cdot f \cdot C} \quad \text{..... (2.1)}$$

Dengan: Xc = Impedansi kapasitif (Ω)

f = Frekuensi (Hz)

C = Kapasitansi (F)

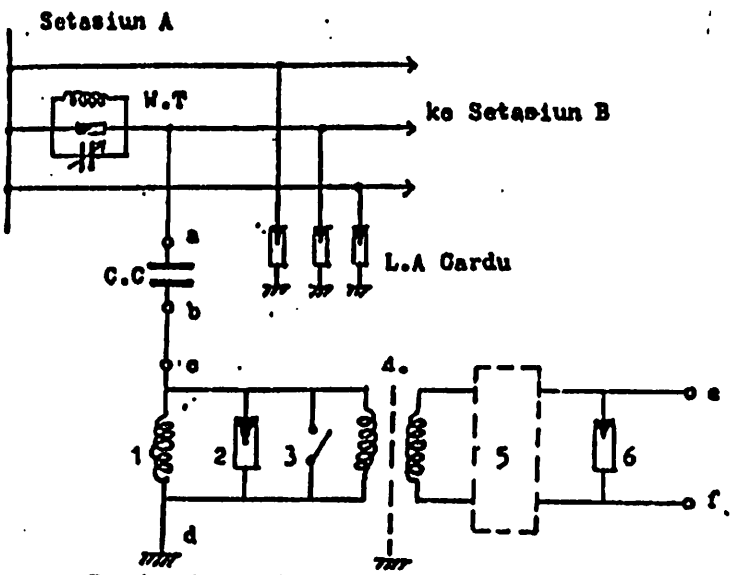
Nilai X_C dapat ditentukan dengan cara mengatur nilai C sehingga didapatkan nilai X_C yang sesuai dengan kebutuhan. Untuk frekuensi jala-jala sebesar 50Hz, X_C bernilai sangat besar sehingga arus yang mengalir ke peralatan komunikasi sangat kecil dan tidak merusak rangkaian. Untuk frekuensi pembawa dari peralatan komunikasi bagian pemancar yang bernilai sangat tinggi, maka X_C bernilai sangat rendah sehingga arus gelombang pembawa dari bagian pemancar dapat ditransmisikan ke rangkaian peralatan komunikasi bagian penerima.

Ada dua jenis hubungan dengan kapasitor:

1. Penghubung kapasitor jenis penala (*tunning type*), yaitu kapasitor merupakan bagian dari alat penala yang dihubungkan seri dengan jaringan tenaga listrik.

2. Penghubung kapasitor jenis penyaring (*filter type*), yaitu kapasitor penggandeng merupakan jaringan berkutub 4 dan menggandengkan peralatan komunikasi dengan jaringan tenaga listrik. IC tipe LM1893 merupakan komponen utama *Modem PLC* yang menggunakan penghubung kapasitor jenis ini.

Gambar 2.2 menunjukkan rangkaian penghubung antara peralatan komunikasi dengan jala-jala listrik.



Gambar 2. Bagian-bagian peralatan kopling.

Keterangan gambar :
1. drain coil
2. arrester pertama.
3. kontak pentanahan
5. peralatan penala
4. transformator penyeimbang dan pengisolasi.
6. arrester kedua.

a. terminal tegangan tinggi kapasitor kopling
b. terminal tegangan rendah CC
c. terminal utama peralatan kopling
d. terminal pentanahan
e,f. terminal peralatan kopling ke terminal PLC (SSB).

Gambar 2.2 Bagian-bagian dari peralatan penghubung/kopling[2]

2.4. Modem PLC dengan IC LM1893

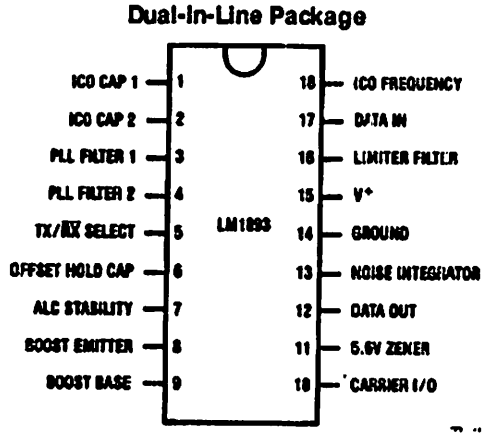
2.4.1. Prinsip Kerja

Sistem PLC menggunakan saluran listrik untuk mengirimkan informasi antara lokasi yang berjauhan. *Chip* PLC LM1893 buatan National Semiconductor ini berfungsi sebagai antarmuka dengan saluran listrik untuk komunikasi *half-duplex* (*bi-directional*) aliran bit serial dari pengkodean manapun. Pada pengiriman data, sinyal *carrier* sinusoida dimodulasi oleh sinyal data yang telah termodulasi FSK dan dipancarkan pada jaringan listrik melalui *driver on-chip* yang berat. Untuk penerimaan, demodulator berbasis PLL dan *impulse noise filter* berkombinasi untuk memberikan jangkauan yang maksimum. Konfigurasi pin dari IC LM1893 ditunjukkan pada gambar 2.3.

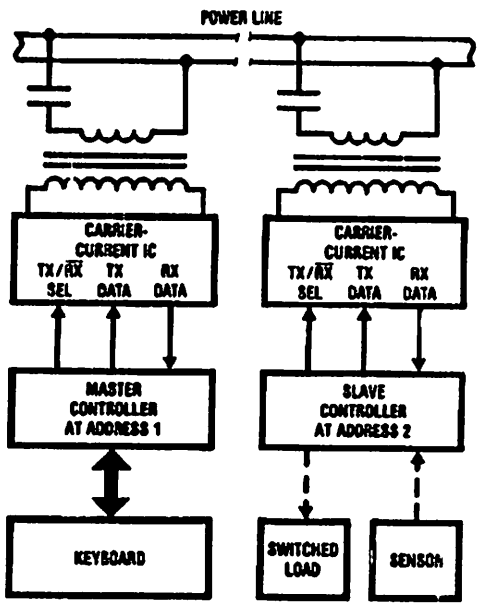
Gambar 2.4 mengilustrasikan sistem komunikasi PLC dengan *Chip Bi-Line* LM1893 untuk antarmuka kontroller digital. *Master Controller* dan *Slave Controller* berfungsi sebagai pengirim dan penerima data serta untuk mengubah mode kerja LM1893 menjadi mode *transmit* atau mode *receive*. *Controller* dapat berupa mikrikontroller atau personal komputer.

Untuk mengirim data, *Controller* mengubah mode kerja *Modem PLC* menjadi mode transmit dengan memberi logika 1 pada pin 5 *Chip* LM1893 (pin TX/RX Select). Kemudian data dikirim secara serial ke pin 17 *Chip* LM1893 (pin Data In). Data serial tersebut akan membangkitkan arus untuk mengontrol osilator. Kemudian keluaran osilator dikuatkan dan dikeluarkan melalui pin 10. Keluaran dari pin 10 ini merupakan data digital yang termodulasi. Keluarannya berupa arus dengan frekuensi tertentu yang dapat membangkitkan tegangan

berosilasi setelah dihubungkan dengan dengan rangkaian penghubung kapasitor jenis penyaring.



Gambar 2.3 Konfigurasi pin LM1893[3]



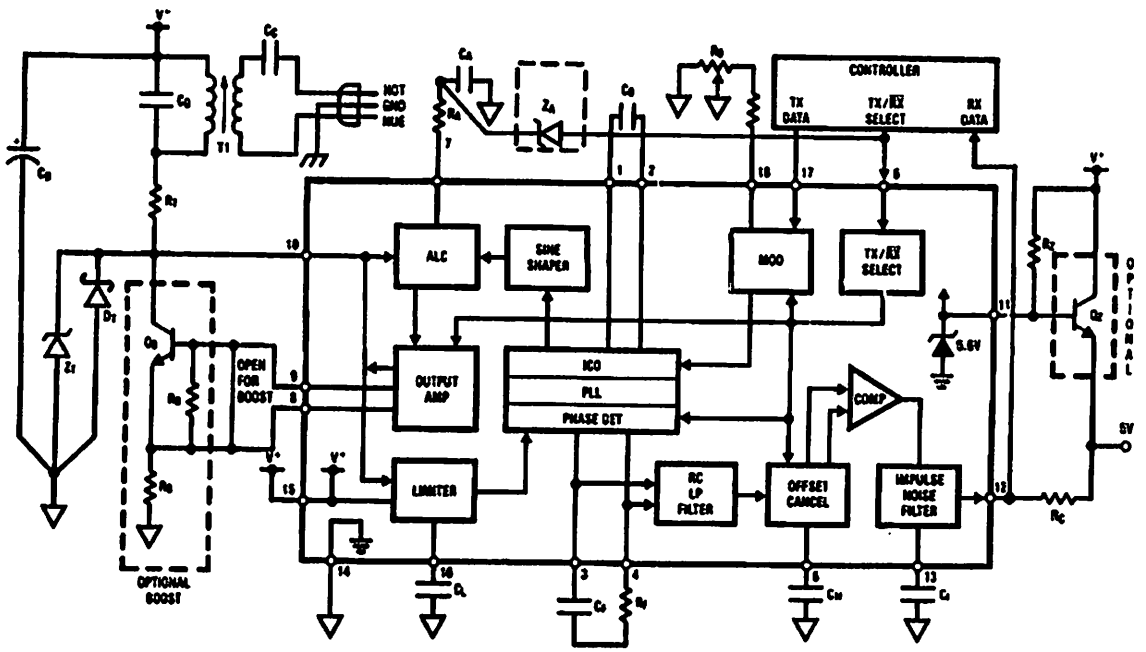
Gambar 2.4 Blok diagram sistem PLC dengan *Chip Bi-Line* LM1893 untuk antarmuka kontroller digital melalui jala-jala listrik[3]

Untuk menerima data, *controller* mengubah mode kerja *modem PLC* menjadi mode *receive* dengan memberi logika 0 pada pin 5 *Chip* LM1893.

Sinyal dari jala-jala listrik dikopel untuk melewatkan frekuensi sinyal dan menahan frekuensi jala-jala, serta PLL (*Phase Lock Loop*) pada *Chip* untuk memisahkan frekuensi sinyal dengan frekuensi pembawa. Sinyal yang telah terpisah dari gelombang pembawa masuk ke bagian demodulator FSK untuk mendapatkan data digital.

2.4.2. Rangkaian Aplikasi

Rangkaian *Modem PLC* terdiri dari IC LM1893 dengan beberapa komponen pendukung seperti gambar dibawah ini.



Gambar 2.5 Diagram blok sistem PLC dengan komponen pendukung[3]

Pada rangkaian tersebut juga terlihat antarmuka dari *modem PLC* dengan *controller*. Untuk mode TX, data serial dari controller digunakan oleh sinyal pembawa (*carrier*) 50 hingga 300 kHz yang termodulasi FSK pada saluran listrik.

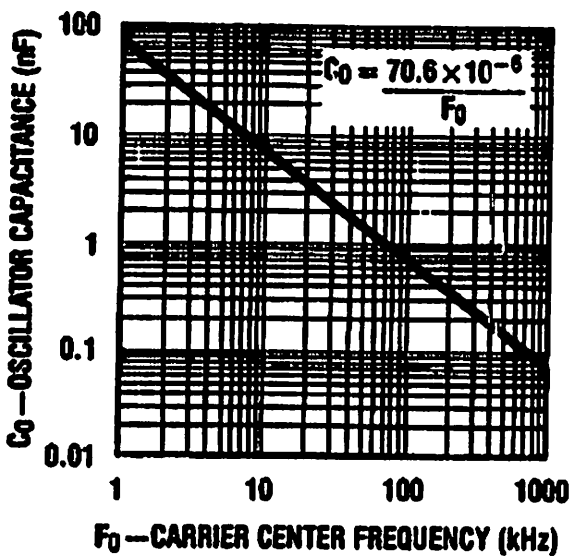
Pada mode RX, sinyal pada saluran akan melalui *coupling transformer* ke dalam penerima berbasis PLL. Data serial yang dihasilkan dikirim ke controller.

2.4.2.1. Bagian Pemancar (*Transmitter*)

Bagian pemancar adalah bagian dari sistem PLC yang berfungsi pada proses pengiriman data. Komponen pendukung untuk bagian pemancar terdiri dari C_O, R_O, C_A, R_A, T_1, C_Q, Z_T, dan R_T. Berikut adalah penjelasan dari fungsi masing-masing komponen tersebut.

- **C_O**

C_O berfungsi untuk menentukan frekuensi pembawa (*carrier*) yang digunakan dalam komunikasi. Penentuan nilai dari C_O didasarkan pada *datasheet* IC LM1893 yang ditunjukkan pada Gambar 2.6.



Gambar 2.6 Penentuan nilai C_O untuk F_O diketahui[3]

Persamaan untuk mendapatkan nilai C_O adalah:

$$C_O = \frac{70.6 \times 10^{-6}}{F_O} \quad \text{.....(2-2)}$$

C_O adalah nilai kapasitansi yang dicari dan F_O adalah nilai dari frekuensi osilasi pembawa.

- **R_O**

Resistor R_O berfungsi untuk menghasilkan arus V_{BE}/R yang akan dikalikan 2 untuk menghasilkan arus kendali ICO 200μA yang menentukan nilai F_O. Semakin kecil nilai R_O akan meningkatkan F_O. Nilai yang dianjurkan untuk R_O adalah antara 5,6kΩ sampai 7,6 kΩ. Karena potensiometer dengan koefisien temperatur lebih rendah relatif mahal, dianjurkan R_O dibuat dari Resistor 5,6kΩ yang dirangkai seri dengan Potensio 2kΩ.

- **C_A dan R_A**

Besarnya nilai C_A dan R_A akan mempengaruhi ALC IC LM1893. Semakin kecil nilai C_A dan R_A maka ALC akan tidak stabil, sedangkan semakin besar nilai C_A dan R_A respon akan lambat tetapi semakin stabil. Nilai yang dianjurkan untuk $C_A = 0{,}1$ μFdan $R_A = 10$ kΩ.

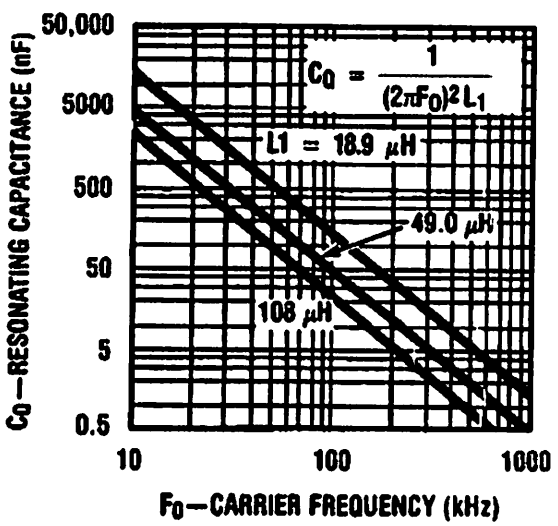
- **T_1**

T_1 digunakan bersama C_Q untuk membentuk frekuensi resonansi F_Q. T_1 juga digunakan sebagai *transformer coupling* pada penumpangan sinyal informasi ke jala-jala listrik. Nilai induktansi kumparan primer L_1 bersama-sama dengan C_Q

akan menentukan frekuensi resonansi. T_1 juga harus mempunyai perbandingan jumlah lilitan kumparan primer dan sekunder sebesar mungkin agar dapat menumpangkan sinyal ke jala-jala dengan baik.

- **C_Q**

C_Q digunakan bersama T_1 sebagai frekuensi resonansi yang sama dengan F_O supaya sinyal informasi dapat ditumpangkan ke jala-jala listrik. Frekuensi resonansi yang dihasilkan oleh C_Q dan induktansi T_1 harus sama dengan F_O. Penentuan nilai C_Q didasarkan pada gambar 2.7.



Gambar 2.7 Hubungan antara nilai C_Q dengan F_O[3]

Persamaan untuk menentukan C_Q adalah :

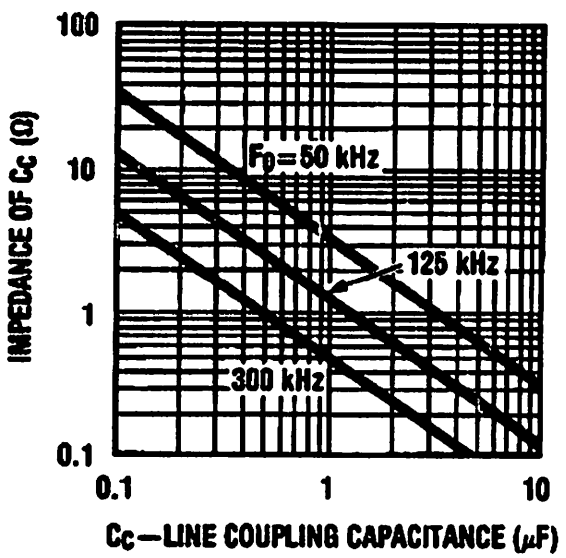
$$C_Q = \frac{1}{(2\pi F_O)^2 L_1} \quad \text{.....(2-3)}$$

Dimana L_1 adalah induktansi primer dari trafo T_1.

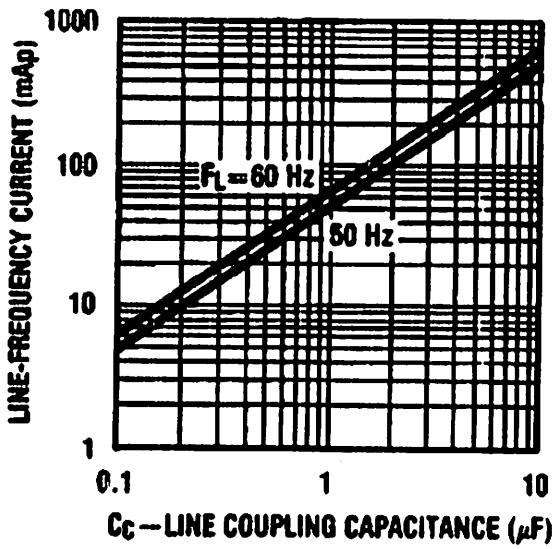
- C_C

Fungsi utama dari kapasitor C_C adalah untuk menghalangi tegangan jala-jala dari lilitan T_1 sisi saluran listrik. Kapasitor C_C dan lilitan T_1 juga berfungsi sebagai LC highpass filter. C_C harus mempunyai impedansi yang cukup rendah pada saat F_O untuk memperbolehkan T_1 mengarahkan energi yang ditransmisikan ke dalam saluran. Untuk mengarahkan saluran listrik 14Ω, impedansi dari C_C haruslah dibawah 14Ω.

Gambar 2.8 digunakan untuk mendapatkan impedansi reaktif dari C_C, untuk memeriksa bahwa nilainya lebih kecil dari impedansi saluran. Kemudian Gambar 2.9 digunakan untuk mengetahui bahwa arus jala-jala listrik cukup rendah untuk menjaga T_1 tidak saturasi. Trafo yang dianjurkan yaitu yang dapat menahan energi magnetisasi 10 Amp-turn (1 Amp melalui 10 lilitan).



Gambar 2.8 Penentuan kapasitansi C_C agar impedansi C_C lebih kecil dari impedansi saluran listrik[3]



Gambar 2.9 penentuan nilai C_C agar T_1 tidak saturasi[3]

- Z_T

Z_T merupakan dioda zener yang digunakan untuk mencegah adanya sinyal transien yang melalui transformator T_1 menuju ke pin *I/O Carrier*. Sinyal transien dapat berasal transient saluran listrik atau transien yang disebabkan dari pengosongan muatan kapasitor C_C pada saat menghubungkan atau melepas rangkaian dari jala-jala listrik. Untuk proteksi diri, *I/O Carrier* mempunyai tegangan clamp internal 44V dengan resistansi seri 20Ω. Z_T adalah dioda avalance yang dirancang khusus untuk mencegah sinyal transien. Z_T dianjurkan mempunyai tegangan dadal (*Breakdown Voltage*) antara 44–49V (sumber: *datasheet* IC LM1893, 1995), karena pin *carrier* I/O mempunyai dioda zener internal dengan *Breakdown Voltage* 44V.

- **R_T**

R_T berfungsi sebagai pembagi tegangan dengan Z_T, yang akan menyerap tegangan transien supaya tegangan yang masuk pada pin *I/O Carrier* tetap dibawah 44V. Nilai yang dianjurkan sebesar 47Ω, ¼ Watt.

- **D_T**

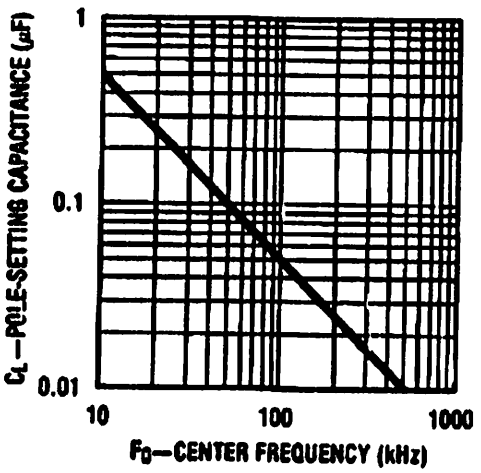
Dioda schottky ini diletakkan secara paralel dengan dioda substrat *chip* PLC untuk melewatkan sebagian besar arus yang berasal dari ground ketika *Carrier Input* atau *Carrier Output* ditarik dibawah ground hingga lebih besar dari dua kali *supply-swing* pada tangki. Z_T juga paralel dengan dioda substrat, namun kurang efektif terkait dengan *forward voltage drop*-nya yang tinggi dan kapasitansi difusi tinggi yang disebabkan oleh *forward speed*-nya yang rendah. Dianjurkan untuk menggunakan 1N5818 yang terbukti menjaga fungsional alur penerimaan dengan penguatan *Transmitter* 20X untuk trafo 7:1 berusaha untuk mengayunkan *carrier* I/O *Receiver* hingga ±100V (300mA arus peak ground pada *Receiver*).

2.4.2.2. Bagian Penerima (*Receiver*)

Komponen yang berbagi baik untuk *Transmitter* maupun *Receiver* adalah C_C, T_1, C_Q, R_T, Z_T, C_O, R_O dan periperal supply serta komponen-komponen bias yang tidak perlu perubahan untuk mode operasi R_X (penerimaan). Komponen pendukung khusus bagian penerima terdiri dari C_L, C_F, R_F, C_I dan Z_A. Penjelasan fungsi dan nilai masing-masing komponen pendukung bagian penerima adalah sebagai berikut:

- C_L

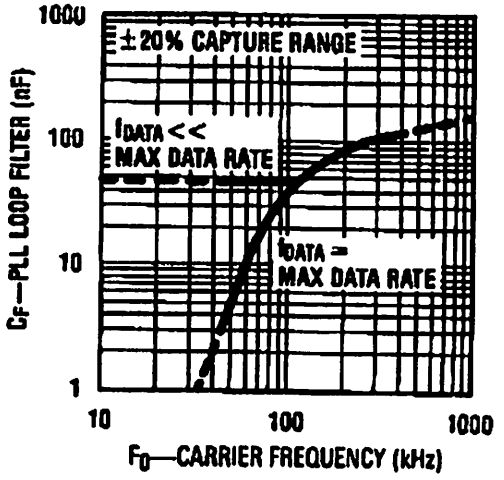
Norton Input Limiter Amplifier pada LM1893 mempunyai *band pass filter* untuk meningkatkan sensitifitas (selektifitas, kekebalan terhadap noise dan penolakan frekuensi jala-jala) penerimaan sinyal. Frekuensi *cut off* atas ditentukan sebesar 300kHz. Sedangkan frekuensi *cut off* bawah ditentukan oleh C_L.



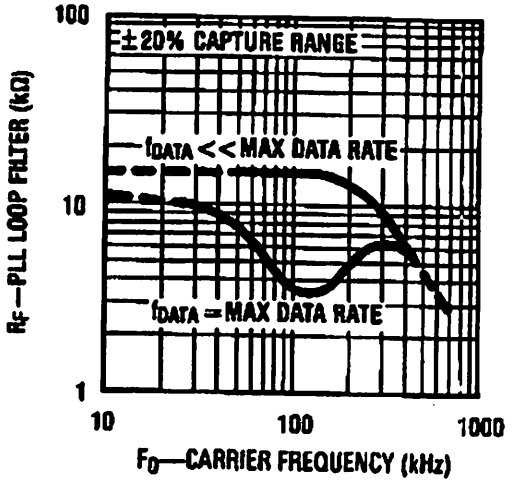
Gambar 2.10 Hubungan antara nilai C_L dengan F_O[3]

- **C_F dan R_F**

C_F dan R_F adalah komponen yang digunakan pada PLL IC LM1893 sebagai filter untuk menghilangkan noise. Besarnya nilai C_F dan R_F ditentukan oleh frekuensi pembawa (FO) yang digunakan. Gambar 2.11 dan 2.12 merupakan hubungan nilai C_F dan R_F.



Gambar 2.11 Penentuan nilai C_F berdasarkan F_O[3]

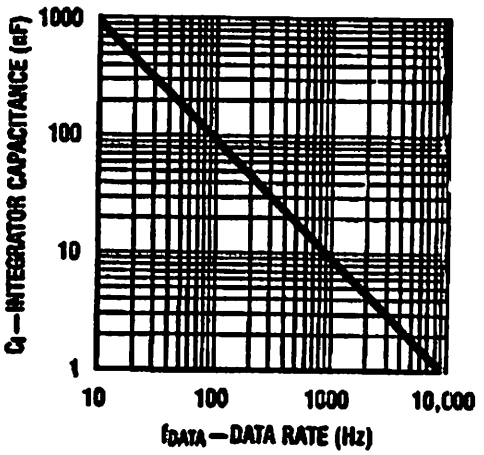


Gambar 2.12 Penentuan nilai R_F berdasarkan F_O[3]

- C_I

Kapasitor integrator *impulse noise filter* C_I digunakan untuk menghalangi jalan lintasan dari pulsa manapun yang lebih pendek daripada waktu *integrator charge.*

Waktu *charge* tersebut, diatur pada nominal waktu ½ bit, merupakan waktu yang diperlukan arus charge ± 50μA untuk mengayun C_I diatas jangkauan 2 V_{BE}. Waktu *charge* dibawah kondisi kasus yang terburuk tidak boleh lebih besar dari waktu bit ketika tidak ada sinyal yang lewat kemudian. Waktu *charge* nominal maksimum dianjurkan ½ bit untuk penggunaan kapasitor ± 10%, jangkauan suhu simpangan penuh, dan jangkauan arus ketetapan penuh. Gambar 2.13 menunjukkan keadaan sebagaimana tersebut diatas.



Gambar 2.13 Hubungan kapasitor integrator *impulse noise filter* C_I dengan F_{DATA} untuk waktu charge adalah waktu ½ bit[3]

- Z_A

Dioda zener silikon 5,1V Z_A diperlukan ketika waktu perpindahan yang cepat RX-ke-TX dibutuhkan pada waktu yang sama bahwa *Chip* beroperasi pada mode RX dengan ayunan sinyal pin 10 mendekati atau melebihi dua kali tegangan *supply*.

2.5. Delphi dan *Database*

Untuk melakukan komunikasi antara mikrokontroller dengan PC, mengolah data, dan menyimpan data pada komputer, diperlukan suatu program aplikasi dan *database*. Program aplikasi yang digunakan adalah Borland Delphi 7.

2.5.1. Delphi

Delphi merupakan perangkat pengembangan aplikasi yang populer dilingkungan Windows. Delphi menggunakan bahasa Pascal sebagai bahasa dasarnya dan merupakan visual dari Pascal. Dengan Delphi dapat diciptakan aplikasi antarmuka (*interfacing*) dengan tampilan yang menarik dan atraktif. Delphi juga telah dilengkapi dengan komponen-komponen untuk mengakses *database*.

Beberapa istilah dasar pada Delphi antara lain *application, form* dan *component application* adalah sederetan kode yang digunakan untuk mengatur komputer agar melakukan sesuatu sesuai dengan keinginan pembuat program. Secara umum, sebuah *application* melibatkan satu atau lebih *form. Form* adalah tampilan berbentuk jendela untuk interaksi antara pengguna program dengan komputer. Sebuah *form* melibatkan satu atau lebih komponen. Beberapa contoh *component* antara lain tombol tekan, tombol radio, edit, memo, cek box dan lain-lain.

2.5.2. *Database*

Basis data (bahasa Inggris: *database*), atau sering pula dieja basisdata, adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik

sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system*, DBMS). Sistem basis data dipelajari dalam ilmu informasi.

Istilah "basis data" berawal dari ilmu komputer. Meskipun kemudian artinya semakin luas, memasukkan hal-hal di luar bidang elektronika, artikel ini mengenai basis data komputer. Catatan yang mirip dengan basis data sebenarnya sudah ada sebelum revolusi industri yaitu dalam bentuk buku besar, kwitansi dan kumpulan data yang berhubungan dengan bisnis.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur atau skema dari jenis fakta yang tersimpan di dalamnya. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan di antara obyek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel.

Istilah *basis data* mengacu pada koleksi dari data-data yang saling berhubungan, dan perangkat lunaknya seharusnya mengacu sebagai *sistem manajemen basis data* (*database management system/DBMS*). Jika konteksnya sudah jelas, banyak administrator dan programer menggunakan istilah basis data untuk kedua arti tersebut.

2.6. Komunikasi Serial

2.6.1. Metode Transfer Data Serial

Pada transmisi data serial, data dikirimkan satu bit demi satu bit melalui kanal komunikasi yang telah dipilih misalnya data dikirimkan dalam bentuk kode ASCII dengan 7 bit untuk tiap karakter, atau dalam bentuk data heksa dengan jumlah data 8 bit untuk setiap karakter. Maka penerima juga harus menerima data, bit demi bit sesuai dengan jumlah bit dalam karakter yang telah ditentukan, sehingga kecepatan data menjadi lebih lambat dibandingkan dengan transmisi data paralel.

Berdasarkan arahnya, transfer data serial dibagi menjadi tiga, yaitu :

1. *Simplex*, pada sistem ini transfer terjadi satu arah saja dimana salah satu obyek dalam sistem hanya bisa bertindak sebagai pengirim saja dan yang satu sebagai penerima saja.
2. *Half duplex*, pada sistem ini transfer terjadi dalam dua arah. Misalkan komputer A dan komputer B. Pada saat komputer A mengirimkan data maka komputer B hanya dapat menerima saja, begitu juga sebaliknya.
3. *Full duplex*, pada sistem ini setiap komputer/obyek yang melakukan transfer data dapat bertindak sebagai pengirim atau penerima pada saat yang bersamaan.

Dalam pengiriman data secara serial harus ada Sinkronisasi atau penyesuaian antara pengirim dengan penerima agar data yang dikirim dapat diterjemahkan dengan tepat oleh penerima. Sehingga dapat dikatakan fungsi Sinkronisasi adalah agar penerima mengetahui dengan tepat bahwa sinyal yang diterimanya merupakan bit dari suatu data (Sinkronisasi bit) dan agar penerima mengetahui dengan tepat bit data yang membentuk sebuah karakter (Sinkronisasi karakter).

Berdasarkan cara Sinkronisasi dikenal 2 mode transmisi serial, yaitu Sinkron dan tak Sinkron.

1. Sinkron

Pada pengiriman data Sinkron sejumlah blok data dikirimkan secara terus-menerus tanpa bit awal atau akhir. Transmisi serial Sinkron digunakan untuk transmisi kecepatan tinggi. Dalam sistem ini baik pengirim maupun penerima bekerja bersama-sama dan Sinkronisasi dilakukan setiap sekian ribu data. Sinkronisasi terjadi dengan jalan mengirimkan pola data tertentu antara pengirim dan penerima. Pola data tertentu ini disebut karakter Sinkronisasi (*synchronization character*). Pengirim akan mengirimkan sejumlah besar data. Pengirim yang mengetahui kode yang digunakan, akan memenggal data tersebut dan meneruskan ke komputer. Transmisi ini lebih efisien karena untuk Sinkronisasi hanya membutuhkan 16 sampai 32 bit. Sementara data dapat mencapai beberapa ribu bit panjangnya. Bentuk transimisi diperlihatkan pada gambar 2.14.

Gambar 2.14 Format pengiriman data Sinkron[4]

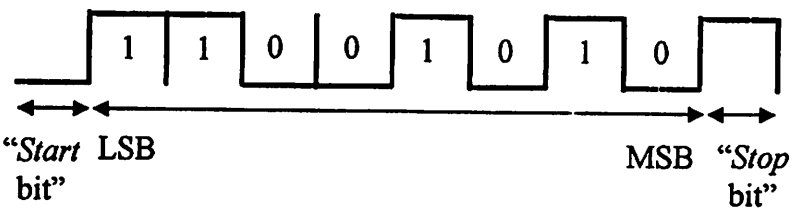
2. Tak Sinkron

Pada mode tak Sinkron, setiap *Byte* data diawali oleh bit "*start*" dan diakhiri dengan bit "*stop*". Jumlah bit *stop* dapat 1 atau 2. Pada deretan bit data dapat juga ditambahkan bit "*parity*". Karakter dapat dikirimkan sekaligus ataupun beberapa karakter kemudian berhenti untuk waktu yang tidak tentu, lalu dikirimkan sisanya. Akibatnya setiap kali penerima harus melakukan Sinkronisasi agar bit data yang dikirimkan diterima dengan benar. Dengan demikian penerima harus mengetahui mulainya bit pertama dari sinyal data. Caranya dengan memberikan suatu pulsa yang disebut *start pulse* pada awal setiap karakter. Pulsa ini memberitahukan penerima untuk mulai menerima bit data. Umumnya keadaan ini disebut *idle*, yaitu keadaan tanpa transmisi sinyal, dalam keadaan tinggi (*mark*). Sehingga dapat dikatakan bahwa selama keadaan *idle* pengirim (*Transmitter)* mengirimkan deretan "1" secara terus-menerus. Keadaan sebaliknya, yaitu "0" disebut *space*. Format pengiriman data serial tak Sinkron diperlihatkan pada gambar 2.15.

Gambar 2.15 Format pengiriman data serial tak Sinkron[4]

Kondisi bit paritas ditentukan oleh sistem paritas yang digunakan (ganjil atau genap). Agar tidak terjadi kesalahan interpretasi antar pengirim dan penerima, maka sistem paritas yang hendak dipakai perlu disetujui bersama, paritas genap atau paritas ganjil. Bit ini akan dipasang pada 1 atau 0 untuk meyakinkan cacah bit 1 pada setiap karakter adalah genap untuk paritas genap, ganjil untuk paritas ganjil. Sehingga, setiap karakter mempunyai panjang 10 bit.

Setiap sistem paritas ini memungkinkan adanya deteksi kesalahan tunggal pada setiap karakter.



Gambar 2.16 Format pengiriman data serial tak Sinkron (1100 1010)[4]

Misalkan kita akan mengirim data 01010011, paritas genap dan bit *stop* 1. karena menggunakan paritas genap dan jumlah bit 1 yang akan dikirimkan jumlahnya sudah genap, bit paritasnya menjadi 0, sehingga format data serial yang dikirimkan dapat dilihat pada gambar 2.16.

2.6.2. *Port* Komunikasi Serial *Personal Computer*

Pada semua komputer pribadi sudah dilengkapi dengan *port-port* komunikasi untuk berhubungan dengan perangkat luar. Pada komputer IBM PC terdapat dua jenis *port* komunikasi yang dapat digunakan yaitu *port* paralel dan *port* serial. *Port* paralel biasanya digunakan untuk berhubungan dengan pencetak *(Printer)* dan peralatan lain yang mampu melakukan hubungan data secara paralel. Sedangkan *port* serial, IBM PC hanya dilengkapi oleh dua *port* serial, yaitu masing-masing disebut COM1 dan COM2. *Port* serial COM1, secara fisik dihubungkan dengan menggunakan konektor DB9 *male* atau *female.* Hampir semua komputer pribadi menggunakan *Chip* pengontrol data *serial western digital* 8250, *Chip* pengontrol terbaru 16550A, atau komponen yang sesuai dari pabrik lain. *Chip* pengontrol menghubungkan secara langsung ke bus ekspansi komputer

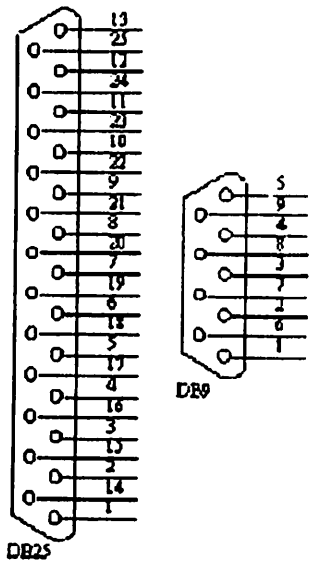
pribadi. Karena *Chip* pengontrol data serial menghubungkan secara langsung ke bus ekspansi, setiap *port* serial menggunakan beberapa alamat dalam ruang alamat masukan/keluaran CPU, dan masing-masing dihubungkan ke sebuah saluran permintaan interupsi (IRQ). Penataan ini memungkinkan CPU membaca dan menulis data secara langsung ke dan dari *port*. Dan setiap *port* menempati 8 alamat *port* yang berdampingan dan satu saluran IRQ. *Port* COM komputer pribadi standar dilokasikan pada alamat 3F8 H dan 2F8 H. COM1 menggunakan saluran IRQ 4, dan COM2 mengunakan saluran IRQ 3.

2.6.2.1. Karakteristik Fisik

Secara fisik, adapter serial pada komputer pribadi adalah berupa konektor DB9 *male* dan DB25 *male*. Fungsi dari masing-masing pena pada konektor dapat dilihat pada tabel 2.1. Dari fungsi masing-masing pena kedua konektor tersebut, maka dapat dirancang perkawatannya untuk mentransfer data. Sedangkan bentuk kedua konektor tersebut ditunjukkan pada gambar 2.17.

Tabel 2.1 Fungsi pin pada DB9 dan DB25[4]

25 Pin	9 Pin	Sinyal
1	-	*Chasis Ground*
2	3	*Transmit Data (TxD)*
3	2	*Receive Data (RxD)*
4	7	*Request To Send (RTS)*
5	8	*Clear To Send (CTS)*
6	6	*Data Set Ready (DSR)*
7	5	*Signal Ground*
8	1	*Carrier Detect (CD)*
20	4	*Data Terminal Ready (DTR)*
22	9	*Ring Indicator (RI)*



Gambar 2.17 Konektor DB 9 dan DB 25 *male*[4]

2.6.2.2. Karakteristik Sinyal Listrik

Karena *port* serial komputer pribadi sesuai dengan standar RS-232, maka sesuai ketentuannya, *port* serial komputer pribadi harus mempunyai karakteristik sinyal listrik sebagai berikut :

a. Tegangan rangkaian terbuka tidak boleh lebih dari 25 V.

b. Keadaan logika "1" (*mark*) ditandai dengan tegangan yang lebih negatif dari –3 V. Pada logika "1" tegangan harus diantara –3 V dan –25 V.

c. Keadaan logika "0" (*space*) ditandai dengan tegangan yang lebih positif dari +3 V. Pada logika "0" tegangan harus diantara +3 V dan +25 V.

d. Hambatan keluaran DC harus lebih kecil dari 7 kΩ jika diberi tegangan antara 3 - 25 V dan harus lebih dari 3 kΩ jika tegangan kurang dari 25 V.

e. *Slew Rate* (perubahan tegangan keluaran per satuan waktu) harus lebih kecil dari 30 V/s. Waktu yang diperlukan untuk melewati daerah peralihan –3 V sampai +3 V tidak boleh lebih dari 1 ms.

2.7. Mikrokontroller ATmega8515

2.7.1. Gambaran Umum

ATmega8515 adalah mikrokontroller CMOS 8-bit dengan konsumsi daya-rendah berbasis pada AVR dengan arsitektur RISC yang ditingkatkan. Dengan kemampuan menjalankan instruksi yang bertenaga dalam siklus *clock* tunggal, ATmega8515 mencapai *throughputs* mendekati 1 MIPS (Million Instruction Per Second) tiap MHz yang mengijinkan perancang sistem untuk mengoptimalkan konsumsi energi terhadap kecepatan pemrosesan.

AVR *core* mengkombinasikan set instruksi yang beragam dengan 32 *general purpose working Register*. Kesemuanya dari 32 *Register* tersebut secara langsung dihubungkan pada Unit Logika dan Aritmatika (ALU), mengijinkan 2 *Register* mandiri untuk diakses dalam satu instruksi tunggal yang mengeksekusi dalam satu siklus *clock*. Arsitektur yang dihasilkan adalah penggunaan kode yang lebih efisien pada saat menuju keberhasilan throughputs hingga sepuluh kali lebih cepat dari mikrokontroller CISC konvensional.

ATmega8515 menyediakan fitur-fitur berikut : 8K *Bytes* In-Sistem Programmable Flash dengan kemampuan *Read-While-Write*, 512 *Bytes* EEPROM, 512 *Bytes* SRAM, sebuah antarmuka Memori eksternal, 35 *general purpose I/O lines*, 32 *general purpose working Register*, dua Timer/*Counters* fleksibel dengan mode-mode pembanding, *interrupt* internal dan eksternal, sebuah USART yang dapat diprogram secara serial, sebuah programmable watchdog timer yang dapat diprogram dengan Osilator internal, sebuah *port* serial SPI, dan tiga software yang bisa dipilih untuk *power saving modes*. Mode *Idle* menghentikan CPU dan membiarkan SRAM, Timer/*Counters*, SPI *port*, dan

sistem *interrupt* untuk terus bekerja. Mode *Power-Down* menyimpan isi *Register* CPU tetapi membekukan Osilator, menghentikan semua fungsi *Chip* sampai *interrupt* berikutnya atau *hardware reset*. Pada mode *Standby*, Osilator kristal/resonator tetap berjalan ketika sebagian sisa perangkat/*device* berhenti. Ini mengijinkan *start-up* yang sangat cepat dikombinasikan dengan komsumsi daya yang rendah.

Perangkat dibuat dengan menggunakan teknologi memori *nonvolatile* kepadatan tinggi yang dimiliki Atmel. *On-chip ISP flash* mengijinkan memori program untuk diprogramkan kembali dalam sistem melalui suatu antarmuka serial SPI, dengan suatu pemrogram memori *nonvolatile* konvensional, atau dengan suatu program boot *on-chip* yang berjalan pada AVR *core*. Program boot dapat menggunakan antarmuka apapun untuk men-*download* program aplikasi ke dalam memori Flash aplikasi. *Software* di dalam bagian *boot flash* akan terus berjalan ketika bagian Flash Aplikasi di-*update*, menyediakan operasi *true Read-While-Write*. Dengan kombinasi suatu CPU RISC 8-Bit dengan Flash *In-Sistem Self-Programmable* pada suatu *Chip* monolitis, Atmel ATmega8515 adalah suatu mikrokontroller kuat yang menyediakan sesuatu yang sangat fleksibel dan solusi hemat biaya untuk banyak aplikasi kendali *embedded*.

ATmega8515 didukung dengan suatu deretan perangkat pengembangan program dan sistem secara penuh mencakup : *C Compiler, Macro Asembler, Program debugger/simulator, In-Circuit Emulator*, dan Kit Evaluasi.

2.7.2. Fitur Mikrokontroler ATmega8515

Mikrokontroller ATmega8515 merupakan produk dari Atmel Corporation yang berkantor pusat di 2325 Orchard Parkway San Jose, CA 95131, USA. ATmega8515 adalah mikrokontroller dari keluarga AVR yang memiliki fitur-fitur sebagai berikut :

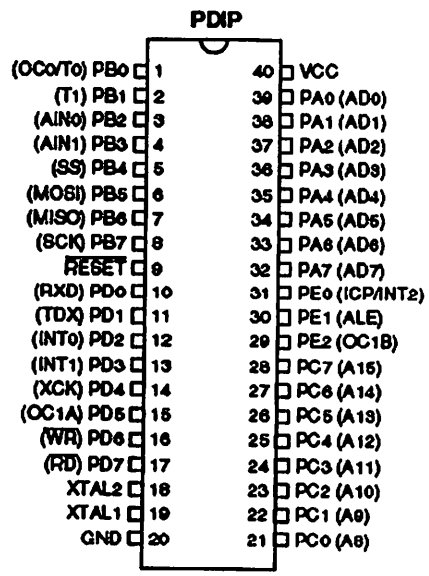
a. Mikrokontroller 8-bit AVR low-power dengan performa tinggi.
b. Arsitektur RISC : 32 x 8 *General Purpose Working Register*, thoroughput hingga 16 MIPS pada 16 MHz, Operasi statis penuh, Multiplier 2 siklus *on-chip*.
c. Memori data dan program yang bersifat *nonvolatile* terdiri dari : 8 K*Bytes* memori program flash dengan daya tahan 10.000 siklus *write/erase*, 512 *Bytes* EEPROM dengan daya tahan 100.000 siklus *write/erase*, 512 *Bytes* memori data SRAM internal, ruang memori eksternal hingga 64 K*Bytes*, penguncian program untuk keamanan software.
d. Fitur Peripheral : sebuah Timer/*Counter* 8-bit dan 16-bit, 3 saluran PWM, USART serial yang dapat diprogram, antarmuka serial SPI *master/slave*, *watchdog timer* yang dapat diprogram dengan *on-chip oscillator* yang terpisah, komparator analog *on-chip*.
e. Fitur Mikrokontroller khusus : Oscilator RC terkalibrasi internal, sumber interupsi internal dan eksternal, 3 mode *sleep* (*idle, power-down, standby*).
f. 35 saluran I/O yang dapat diprogram yang terdiri dari 4 buat saluran I/O 8-bit dan 1 buah I/O 3-bit.

g. Tegangan operasi : 4,5V – 5,5V.

h. Nilai kecepatan : 0 – 16 MHz.

2.7.3. Konfigurasi Pin

Konfigurasi Pin dari mikrokontroller ATmega8515 dengan kemasan 40-pin PDIP ditunjukkan pada gambar berikut :



Gambar 2.18 Pinout ATmega8515[5]

Penjelasan dari susunan pin ATmega8515 :

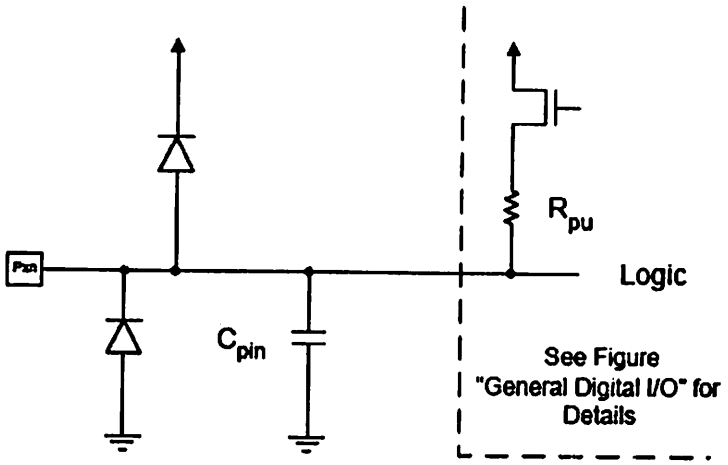
1. VCC (pin 40) : untuk *supply* tegangan digital.
2. GND (pin 20) : untuk *ground.*
3. *Port* A (pin 39... 32) : PA7 – PA0 merupakan *port* I/O *bi-directional* dengan resistor *pull-up* internal.
4. *Port* B (pin 1... 8) : PB0 – PB7 merupakan *port* I/O *bi-directional* dengan resistor *pull-up* internal.

5. *Port* C (pin 21... 28) : PC0 – PC7 merupakan *port* I/O *bi-directional* dengan resistor *pull-up* internal.
6. *Port* D (pin 10... 17) : PD0 – PD7 merupakan *port* I/O *bi-directional* dengan resistor *pull-up* internal.
7. *Port* E (pin 29, 30 dan 31) : PE2, PE1 dan PE0 merupakan *port* I/O *bi-directional* dengan resistor *pull-up* internal.
8. $\overline{\text{RESET}}$ (pin 9) : untuk input reset aktif low.
9. XTAL1 (pin 19) : masukan untuk penguat *inverting Oscillator* masukan untuk rangkaian operasi *clock* internal.
10. XTAL2 (pin 18) : keluaran dari penguat *inverting Oscillator*.

2.7.4. Port I/O

Semua port-port AVR fungsionalitas *true Read-Modify-Write* ketika digunakan sebagai *general digital I/O ports*. Ini berarti bahwa arah dari satu pin port dapat berubah tanpa perubahan arah yang tidak disengaja dari pin manapun dengan instruksi-instruksi SBI (*Set Bit in I/O Register*) dan CBI (*Clear Bit in I/O Register*). Penerapan yang sama ketika mengubah nilai pengarah (*drive value*) jika diatur sebagai keluaran atau *enable/disable* resistor *pull-up* jika diatur sebagai masukan. Masing-masing keluaran penyangga (*buffer output*) mempunyai karakteristik pengarah simetris dengan kedua-duanya yang *high sink* dan *source capability*. Pengarah (*driver*) pin cukup kuat untuk men-*drive* tampilan LED secara langsung. Semua pin port mempunyai resistor pull-up yang bisa dipilih sendiri dengan resistansi *invariant* sumber tegangan. Semua pin-pin I/O

mempunyai dioda proteksi untuk VCC dan Ground sebagaimana ditunjukkan pada Gambar 2.19.



Gambar 2.19 Skema ekivalent pin I/O[5]

Semua referensi register dan bit pada bagian ini ditulis dalam bentuk umum. Huruf kecil "x" mewakili bentuk penomeran untuk port, dan huruf kecil "n" mewakili nomer bit. Bagaimanapun, ketika menggunakan bit atau register yang didefinisikan dalam suatu program, format yang tepat harus digunakan. Sebagai contoh, PORTB3 untuk bit nomer 3 pada port B, di sini biasanya didokumentasikan sebagai PORTxn.

Kebanyakan pin-pin pada port mempunyai fungsi alternatif sebagai tambahan disamping menjadi general digital I/O. Sub bagian berikut menguraikan fungsi alternatif untuk masing-masing port, dan menghubungkan sinyal penolakan untuk fungsi alternatif. Mengacu pada uraian fungsi alternatif untuk detil lebih lanjut.

1. Fungsi Alternatif dari Port A

Port A mempunyai fungsi alternatif sebagai alamat *byte* rendah dan saluran data untuk antarmuka memori eksternal.

Table 2.2 Fungsi alternatif pin-pin Port A[5]

Port Pin	Alternate Function
PA7	AD7 (External memory interface address and data bit 7)
PA6	AD6 (External memory interface address and data bit 6)
PA5	AD5 (External memory interface address and data bit 5)
PA4	AD4 (External memory interface address and data bit 4)
PA3	AD3 (External memory interface address and data bit 3)
PA2	AD2 (External memory interface address and data bit 2)
PA1	AD1 (External memory interface address and data bit 1)
PA0	AD0 (External memory interface address and data bit 0)

2. Fungsi Alternatif dari Port B

Pin-pin Port B dengan fungsi alternatifnya ditunjukkan pada tabel dibawah ini.

Table 2.3 Fungsi alternatif pin-pin Port B[5]

Port Pin	Alternate Functions
PB7	SCK (SPI Bus Serial Clock)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB5	MOSI (SPI Bus Master Output/Slave Input)
PB4	$\overline{SS}$ (SPI Slave Select Input)
PB3	AIN1 (Analog Comparator Negative Input)
PB2	AIN0 (Analog Comparator Positive Input)
PB1	T1 (Timer/Counter1 External Counter Input)
PB0	T0 (Timer/Counter0 External Counter Input) OC0 (Timer/Counter0 Output Compare Match Output)

3. Fungsi Alternatif dari Port C

Pin-pin Port C dengan fungsi alternatifnya ditunjukkan pada tabel berikut ini.

Table 2.4 Fungsi alternatif pin-pin Port C[5]

Port Pin	Alternate Function
PC7	A15 (External memory interface address bit 15)
PC6	A14 (External memory interface address bit 14)
PC5	A13 (External memory interface address bit 13)
PC4	A12 (External memory interface address bit 12)
PC3	A11 (External memory interface address bit 11)
PC2	A10 (External memory interface address bit 10)
PC1	A9 (External memory interface address bit 9)
PC0	A8 (External memory interface address bit 8)

4. Fungsi Alternatif dari Port D

Pin-pin port D dengan fungsi alternatif ditunjukkan pada table 2.5

Table 2.5 Fungsi alternatif pin-pin port D[5]

Port Pin	Alternate Function
PD7	$\overline{RD}$ (Read Strobe to External Memory)
PD6	$\overline{WR}$ (Write Strobe to External Memory)
PD5	OC1A (Timer/Counter1 Output Compare A Match Output)
PD4	XCK (USART External Clock Input/Output)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

5. Fungsi Alternatif dari Port E

Pin-pin port E dengan fungsi alternatif ditunjukkan pada tabel dibawah ini.

Table 2.6 Fungsi alternatif pin-pin port E[5]

Port Pin	Alternate Function
PE2	OC1B (Timer/Counter1 Output Compare B Match Output)
PE1	ALE (Address Latch Enable to External Memory)
PE0	ICP (Timer/Counter1 Input Capture Pin) INT2 (External Interrupt 2 Input)

2.7.5. Interupsi (*interrupt*)

AVR menyediakan beberapa sumber interupsi yang berbeda. Vektor Interupsi dan Reset yang terpisah masing-masing mempunyai vektor program yang terpisah di dalam ruang memori program. Bergantung pada *Program Counter value*, interupsi kemungkinan akan tidak diperbolehkan *(disabled)* ketika *Boot Lock bit* BLB02 atau BLB12 diprogram. Fitur ini meningkatkan keamanan software.

Alamat-alamat terendah pada ruang memori program secara *default* didefinisikan sebagai vektor reset dan interupsi. Daftar lengkap dari vektor-vektor ditunjukkan pada tabel 2.7. Daftar pada tabel tersebut juga menunjukkan tingkatan prioritas dari interupsi-interupsi yang berbeda. Semakin rendah alamatnya semakin tinggi tingkat prioritas. RESET mempunyai prioritas yang tertinggi dan berikutnya adalah INT0 – permintaan interupsi eksternal 0. Vektor interupsi dapat dipindah ke awal bagian *Boot Flash* dengan mengatur bit IVSEL pada *General Interrupt Control Register* (GICR). Vektor Reset dapat juga dipindahkan pada awal bagian Boot Flash dengan pemrograman BOOTRST Fuse.

Tabel 2.7 Vektor Reset dan Interupsi[5]

Vector No.	Program Address(2)	Source	Interrupt Definition
1	$000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog Reset
2	$001	INT0	External Interrupt Request 0
3	$002	INT1	External Interrupt Request 1
4	$003	TIMER1 CAPT	Timer/Counter1 Capture Event
5	$004	TIMER1 COMPA	Timer/Counter1 Compare Match A
6	$005	TIMER1 COMPB	Timer/Counter1 Compare Match B
7	$006	TIMER1 OVF	Timer/Counter1 Overflow
8	$007	TIMER0 OVF	Timer/Counter0 Overflow
9	$008	SPI, STC	Serial Transfer Complete
10	$009	USART, RXC	USART, Rx Complete
11	$00A	USART, UDRE	USART Data Register Empty
12	$00B	USART, TXC	USART, Tx Complete
13	$00C	ANA_COMP	Analog Comparator
14	$00D	INT2	External Interrupt Request 2
15	$00E	TIMER0 COMP	Timer/Counter0 Compare Match
16	$00F	EE_RDY	EEPROM Ready
17	$010	SPM_RDY	Store Program memory Ready

Catatan :

1. Ketika BOOTRST fuse diprogram, *device* akan berpindah ke alamat *Boot Loader* pada saat reset.

2. Ketika bit IVSEL pada GICR di-set, vektor interupsi akan dipindahkan pada awal dari bagian *Boot Flash*. Alamat dari tiap-tiap vektor interupsi kemudian akan menjadi alamat pada tabel tambahan berikut pada alamat awal dari bagian *Boot Flash*.

Tabel 2.8 menunjukkan penempatan vektor reset dan interupsi untuk berbagai kombinasi dari seting BOOTRST dan IVSEL. Jika program tidak pernah memperbolehkan sumber interupsi, vektor interupsi tidak digunakan, dan kode program regular dapat ditempatkan pada lokasi ini.

Ini juga merupakan kasus jika vektor reset ada pada bagian aplikasi ketika vektor interupsi berada pada bagian Boot dan sebaliknya.

Tabel 2.8 Penempatan Vektor Reset dan Interupsi[5]

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	$0000	$0001
1	1	$0000	Boot Reset Address + $0001
0	0	Boot Reset Address	$0001
0	1	Boot Reset Address	Boot Reset Address + $0001

2.7.6. *Port* Serial Mikrokontroller ATmega8515

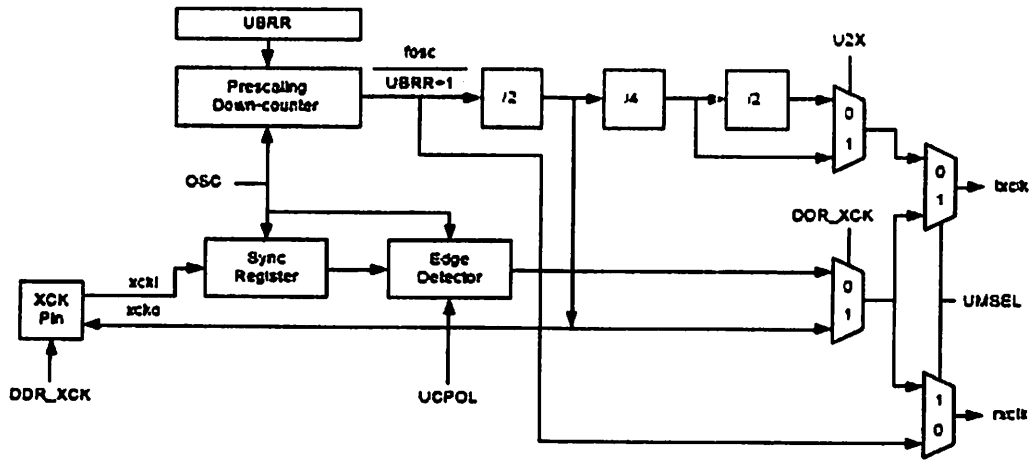
Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) merupakan suatu alat komunikasi serial yang sangat fleksibel. Fitur utamanya adalah :

- Operasi *Full Duplex* (*Register* pengirim dan penerima serial yang berdiri sendiri).
- Operasi Sinkron atau Tak Sinkron.
- *Master* atau *Slave* mendapat *clock* dengan operasi *Synchronous*.
- Pembangkit *Baud Rate* dengan resolusi tinggi.
- Mendukung *Serial Frames* dengan 5, 6, 7, 8, atau 9 Bit Data dan 1 atau 2 *Stop* Bit.
- Pembangkitan paritas genap atau ganjil serta pemeriksaan paritas yang didukung oleh *Hardware*.
- Pendeteksian *Data OverRun*.
- Pendeteksian *Framing Error*.

- Pemfilteran *Noise* mencakup pendeteksian kesalahan *Start Bit* dan *Low Pass Filter* Digital.
- 3 interupsi yang terpisah untuk *TX Complete*, *TX Data Register Empty*, dan *RX Complete*.
- Mode komunikasi *Multi-processor*.
- Kecepatan ganda (*Double Speed*)untuk mode komunikasi Tak Sinkron.

2.7.6.1. Pembangkitan *clock*

Logika pembangkitan *clock* menghasilkan *clock* dasar untuk *Transmitter* dan *Receiver*. USART mendukung 4 mode untuk operasi *clock* : Mode Tak Sinkron Normal, Tak Sinkron kecepatan ganda, Sinkron *Master* dan Sinkron *Slave*. Bit *UMSL USART Control and Status Register C* (UCSRC) untuk memilih antara operasi Sinkron dan tak Sinkron. Kecepatan ganda (hanya untuk mode tak Sinkron) diatur oleh U2X yang dapat ditemukan pada *Register* UCSRA. Ketika menggunakan mode Sinkron (UMSEL=1), *Data Direction Register* untuk XCK pada pin (DDR_XCK) mengatur apakah sumber *clock*-nya internal (*Master Mode*) atau eksternal (*Slave Mode*). Pin XCK. hanya akan aktif pada saat menggunakan mode tak Sinkron.



Gambar 2.20 Diagram blok logika pembangkitan *Clock* [5]

Keterangan sinyal :

txclk : *Clock Transmitter.* (sinyal internal)

rxclk : Basis *clock* untuk *Receiver.* (sinyal eksternal)

xcki : Masukan dari pin xck (sinyal internal). Digunakan untruk operasi *Slave* Sinkron.

xcko : *Port* keluaran untuk pin XCK (sinyal internal).digunakan untuk operasi *Master* Sinkron.

fosc : Frekuensi pin XTAL. (*clock* sistem)

A. Pembangkitan *Clock* Internal - Pembangkit *Baud rate*

Pembangkitan *clock* internal digunakan untuk operasi mode *master* Sinkron dan tak Sinkron. USART *Baud Rate Register* (UBRR) dan *down-counter* berkaitan dengan fungsinya sebagai *programmable prescaler* atau pembangkit *baud rate*. *Down-counter* berjalan sebagai *clock* sistem (fosc), dimuati dengan nilai UBRR untuk setiap saat *counter* telah menghitung mundur ke nol atau ketika *Register* UBRR ditulis. *Clock* dibangkitkan tiap saat *counter* mencapai nol. *Clock* merupakan keluaran *clock* pembangkit *baud rate* (=fosc/(UBRR+1)). *Transmitter* membagi keluaran *clock* pembangkit *baud rate* dengan 2, 8 atau 18 bergantung pada mode.

Keluaran pembangkit *baud rate* digunakan secara langsung dengan *clock Receiver* dan *recovery data unit.* Bagaimanapun, unit *recovery* menggunakan mesin status yang menggunakan status 2, 8 atau 18 bergantung pada set mode pada bit UMSEL, U2X dan DDR_XCK.

Pada table dibawah terdapat persamaan untuk menghitung *baud rate* (dalam bit per second) dan untuk menghitung nilai UBRR untuk setiap mode operasi yang menggunakan sumber *clock* yang dibangkitkan secara internal.

Tabel 2.9 Persamaan untuk perhitungan pengaturan *Register baud rate*[5]

Operating Mode	Equation for Calculating Baud Rate[1]	Equation for Calculating UBRR Value
Asynchronous Normal mode (U2X = 0)	$BAUD = \frac{f_{OSC}}{16(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed mode (U2X = 1)	$BAUD = \frac{f_{OSC}}{8(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master mode	$BAUD = \frac{f_{OSC}}{2(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

Note: 1. The baud rate is defined to be the transfer rate in bit per second (bps).

Keterangan:

BAUD Baut rate (dalam bit per detik, bps)

f_{osc} Frekuensi *clock* Osilator sistem

UBRR Isi dari Register UBRRH dan UBRRL, (0-4095)

B. *Clock* Eksternal

Clock eksternal digunakan pada operasi Sikron mode *slave.* Masukan *clock* eksternal dari pin XCK di-sample oleh register sinkronisasi untuk memperkecil kesempatan meta-stabilitas.

Keluaran dari register sinkronisasi kemudian harus lewat melalui suatu detektor tepi (*edge detector*) sebelum dapat digunakan oleh *Transmitter* dan

Receiver . Proses ini memperkenalkan dua periode waktu tunda (*clock delay*) CPU dan oleh karena itu frekuensi *clock* XCK eksternal maksimum dibatasi oleh persamaan berikut:

$$f_{XCK} < \frac{f_{OSC}}{4}$$

Yang menjadi catatan bahwa F_{OSC} bergantung pada stabilitas dari sumber *clock* sistem. Oleh karena itu dianjurkan untuk menambahkan beberapa garis tepi untuk menghindari kemungkinan kehilangan data berkaitan dengan variasi frekuensi.

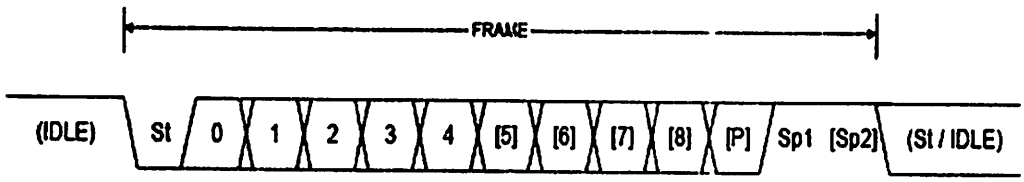
2.7.6.2. Format Bingkai (*Frame*) Serial

Bingkai serial ditentukan menjadi satu karakter dari bit-bit data dengan bit-bit sinkronisasi (bit *start* dan *stop*), dan secara bebas pilih bit paritas untuk mengecek kesalahan. USART menerima semua 30 kombinasi berikut sebagai seperti format bingkai yang sah :

- 1 bit *start*
- 5, 6, 7, 8 atau 9 bit data
- Tanpa bit paritas, bit paritas ganjil dan genap
- 1 atau 2 bit *stop*

Bingkai (*frame*) dimulai dengan bit *start* diikuti oleh *least significant data bit*. Kemudian bit-bit data berikutnya, sampai total enam, adalah berhasil, diakhiri dengan *most significant bit*. Jika diperbolehkan, bit paritas disisipkan setelah bit data sebelum bit *stop*. Ketika bingkai yang lengkap terkirim, dapat diikuti secara langsung *frame* yang baru atau saluran komunikasi dapat diatur pada keadaan *idle*.

Gambar 2.21 menggambarkan kemungkinan kombinasi dari format bingkai (*frame*). Bit-bit di dalam tanda kurung merupakan pilihan.



Gambar 2.21 Format bingkai (*Frame*)[5]

Keterangan:

St Bit *start*, selalu berlogika rendah

(n) Bit-bit data (0-8)

P Bit paritas. Bisa ganjil atau genap

Sp Bit *stop*, selalu berlogika tinggi

IDLE Tidak ada *transfer* data pada saluran komunikasi (TXD atau RXD). Saluran pada keadaan IDLE harus berlogika tinggi.

Format bingkai digunakan oleh USART di-set dengan bit-bit UCSZ2:0, UPM1:0 dan USBS didalam UCSRB dan UCSRC. *Receiver* dan *Transmitter* menggunakan pengaturan yang sama. Catatan bahwa perubahan pengaturan apapun pada bit ini akan merusak semua komunikasi yang sedang berlangsung baik pada *Transmitter* atau *Receiver.*

Ukuran karakter USART bit (UCSZ2:0) memilih sejumlah bit data di dalam bingkai. Bit-bit (UPM1:0) mode paritas USART memperbolehkan dan men-set bit paritas. Pemilihan antara satu atau dua bit *stop* dilakukan oleh bit USART *Stop* Bit Select (USBS).

Receiver mengabaikan bit *stop* yang kedua. Sebuah FE (*Frame* Error) oleh karena itu hanya akan dideteksi didalam hal yang mana bit *stop* yang pertama adalah nol.

Bit paritas dihitung dengan melakukan *exclusive-or* dari semua bit-bit data. Jika paritas ganjil digunakan, hasil dari operasi *exclusive-or* dibalik. Hubungan antara bit paritas dan bit data adalah sebagai berikut :

$$P_{even} = d_{n-1} \oplus \ldots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 0$$
$$P_{odd} = d_{n-1} \oplus \ldots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 1$$

Keterangan:

P_{even}	Bit paritas menggunakan paritas genap (*even parity*)
P_{odd}	Bit paritas menggunakan paritas ganjil (odd parity)
d_n	n bit data dari karakter

Jika digunakan, bit paritas diletakkan diantara bit data pertama dan bit data terakhir dari bingkai (*frame*) serial.

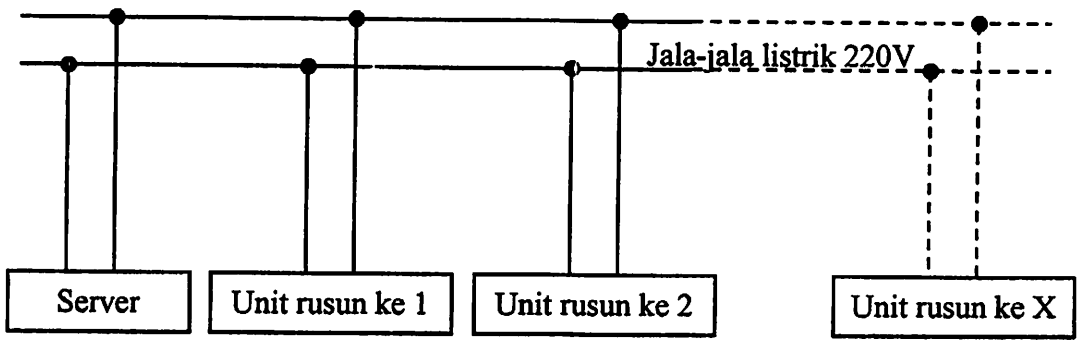
BAB III

PERANCANGAN DAN PEMBUATAN ALAT

Pada bagian ini, dijelaskan perancangan dan pembuatan sistem *server* untuk kWh meter prabayar pada rumah susun yang menggunakan jala-jala listrik sebagai media komunikasi data. Perancangan dan pembuatannya meliputi perencanaan aplikasi sistem, penggambaran blok diagram, perancangan perangkat keras dan perancangan perangkat lunak.

3.1. Perencanaan Aplikasi Sistem *Server*

Berikut ini merupakan gambaran umum komunikasi antara sistem *server* dengan kWh prabayar pada tiap unit rumah susun menggunakan media jala-jala listrik 220Volt.



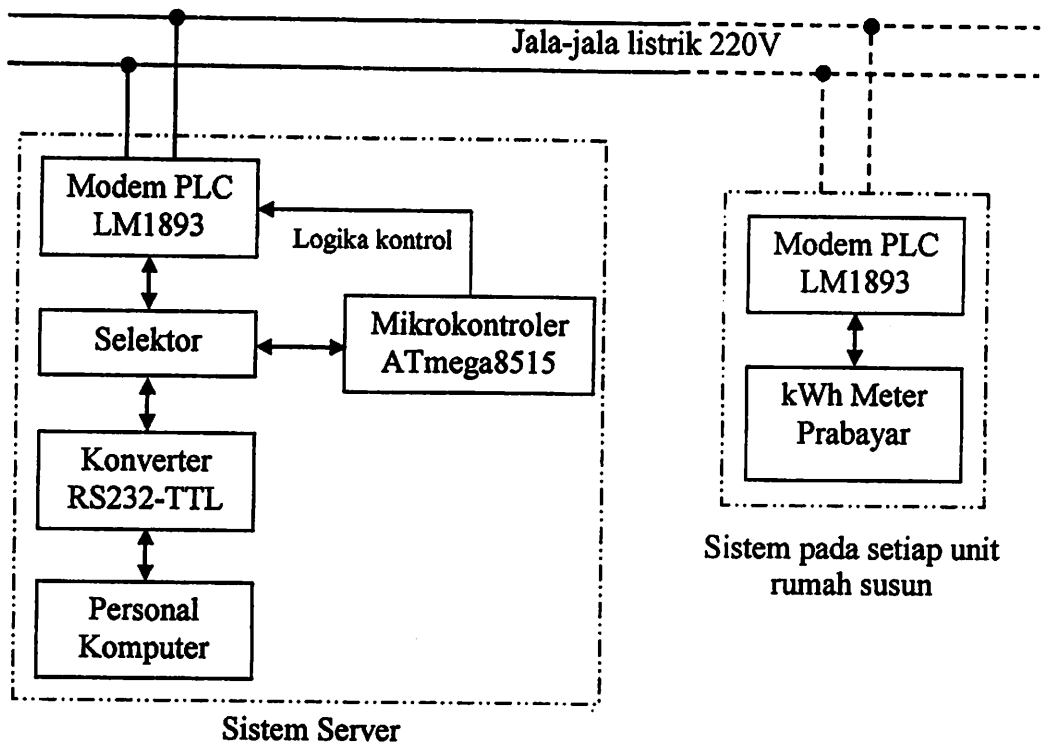
Gambar 3.1 Diagram Blok Aplikasi Sistem *Server*

Server antara lain digunakan untuk melayani pengisian pulsa prabayar, pengecekan pulsa dan status pelanggan, serta mengontrol *supply* energi listrik ke pelanggan. Sistem *server* berkomunikasi dengan kWh meter prabayar pada setiap

unit rusun melalui media jala-jala listrik. Agar komunikasi tersebut dapat berlangsung, pada *server* maupun kWh meter prabayar diperlukan *modem* komunikasi jala-jala listrik atau *Modem Power Line Carrier (PLC).*

Pada kWh meter prabayar terdapat informasi yang tersimpan pada memori *controller* yang setiap saat siap digunakan apabila diminta oleh *server.* Data atau informasi tersebut diantaranya yaitu identitas/ID Pelanggan, status pelanggan dan jumlah saldo pulsa. Untuk mempermudah pemrosesan data, besaran pulsa yang digunakan adalah dalam bentuk satuan energi listrik yaitu kilowatt hour (kWh). Sehingga dalam pengisian pulsa data yang dikirim oleh *server* nantinya adalah nominal energi listrik atau kWh.

3.2. Diagram Blok dan Spesifikasi Sistem



Gambar 3.2 Diagram blok sistem *server*

Pada gambar diatas terlihat bahwa sistem *server* terdiri dari beberapa bagian yaitu personal komputer (PC), konverter atau pengubah level tegangan RS232-TTL, Selektor, Mikrokontroller ATmega8515 dan *Modem PLC* LM1893. Untuk spesifikasi dari masing-masing bagian tersebut dapat dijelaskan sebagai berikut :

1. Personal Komputer

Personal Komputer digunakan sebagai komputer *server* dalam melayani pelanggan listrik prabayar pada area rumah susun. Spesifikasi komputer yang dapat digunakan sebagai *server* harus mendukung OS Microsoft Windows XP dan mempunyai sebuah port serial untuk antarmuka peralatan komunikasi *PLC*. Komputer menggunakan pemrograman Delphi 7 dan Database Paradoks untuk menghasilkan program aplikasi yang berfungsi untuk melayani pelanggan listrik rumah susun.

2. Konverter Level Tegangan RS232-TTL

Konverter level tegangan RS232-TTL berfungsi untuk merubah data serial yang dikirim dari PC dengan level tegangan RS232 menjadi data serial yang dikirim ke mikrokontroller dengan level tegangan TTL. Rangkaian ini berbasis IC Maxim MAX232.

3. Selektor

Selektor berfungsi untuk memilih hubungan antara PC dengan Mikrokontroler atau Mikrokontroler dengan *Modem*. Selektor menggunakan IC *Bus Buffer* 74LS126 dan *inverter* 74LS04.

4. Mikrokontroler ATmega8515

Mikrokontroler ini berfungsi untuk mengatur mengatur selektor, menentukan mode TX/RX *Modem PLC*, serta mengirim dan menerima data serial baik dari/ke PC maupun *Modem PLC*. Mikrokontroler yang digunakan adalah ATmega8515 dari keluarga Atmel AVR.

5. *Modem PLC* LM1893

Modem PLC berfungsi sebagai modulator untuk data digital dari mikrokontroler dan demodulator untuk data yang masuk dari kWh meter prabayar pelanggan, sehingga dapat terjadi komunikasi antara *server* dengan kWh meter prabayar melalui jala-jala listrik. *Modem* ini berbasis IC LM1893 yang merupakan produk dari National Semiconductor.

3.3. Prinsip Kerja dari Sistem *Server*

Untuk melayani pengisian pulsa pelanggan listrik prabayar pada rumah susun, *server* terlebih dahulu meminta data pelanggan (nama atau ID kWh meter). Kemudian menentukan jenis perintah yang akan dikerjakan antara lain cek status, isi ulang pulsa prabayar dan kendali kWh meter (On/Off). Untuk cek status, *server* akan meminta data saldo pulsa dan status kWh meter (On/Off). Untuk isi ulang pulsa prabayar, *server* memasukkan besarnya pulsa yang akan dikirim ke kWh meter prabayar. *Server* juga dapat mengendalikan *supply* energi listrik ke pelanggan dengan memberikan perintah untuk mengatur "*switch*" pada kWh meter prabayar, sehingga apabila dikehendaki listrik ke pelanggan yang dituju dapat dipadamkan.

Berbagai bentuk *service* ke pelanggan sebagaimana tersebut di atas membutuhkan format data komunikasi untuk mengantisipasi adanya kesalahan atau *miss communication.* Pada mikrokontroler, format data yang diterima dari PC (berupa kode perintah, ID Tujuan dan jumlah pulsa) ditambahi dengan *header* dan ID *Server* sebelum dikirim ke kWh meter prabayar. Penjelasan lebih lanjut mengenai format data komunikasi dapat dilihat pada sub bab 3.5 mengenai protokol komunikasi.

Untuk proses pengiriman data ke pelanggan, selektor menghubungkan PC dengan Mikrokontroler (merupakan kondisi *default*). Data serial yang dikirim dari PC ditambahi dengan *header* dan alamat *server.* Mikrokontroller kemudian mengirimkan logika high ke pin *TX/RX Select Modem PLC* untuk memilih mode pengiriman atau TX. Mikrokontroler juga mengontrol selektor agar menghubungkan Mikrokontroler dengan *Modem PLC* sehingga terjadi komunikasi serial. Data serial tersebut dimodulasi digital dengan teknik modulasi *FSK (Frequency Shift Keying).* Data yang dihasilkan kemudian akan dimodulasikan dengan frekuensi pembawa (*carrier*) 125kHz pada jala-jala atau saluran listrik pelanggan rumah susun.

Untuk proses penerimaan data dari kWh meter pelanggan, selektor dibiarkan untuk tetap menghubungkan Mikrokontroler dengan *Modem PLC* dengan memberikan waktu delay tertentu setelah pengiriman data serial dilakukan. Sinyal yang diterima dari kWh meter didemodulasi oleh *Modem PLC* hingga didapatkan data serial. Data ini kemudian akan diterima Mikrokontroler dan dikirimkan ke PC dengan mengontrol selektor untuk menghubungkan Mikrokontroler dengan PC.

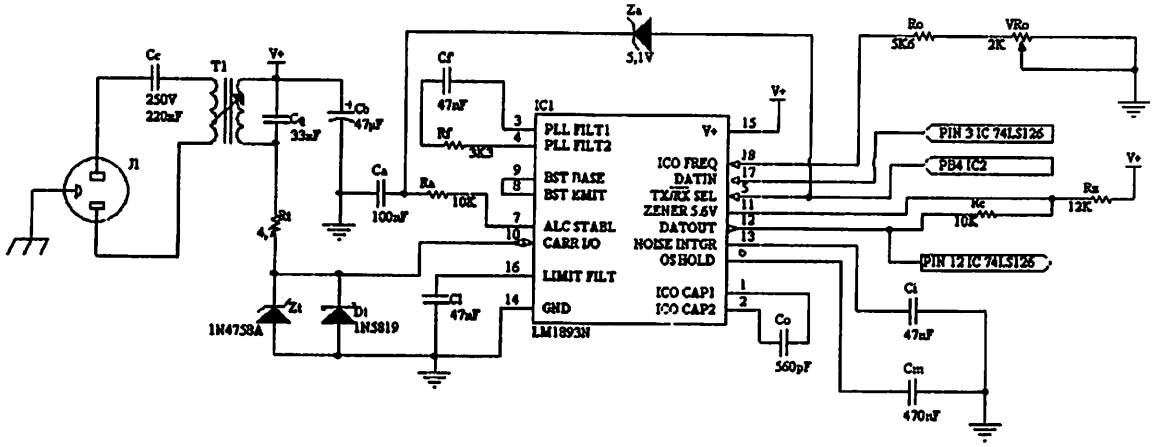
3.4. Perancangan Perangkat Keras

Perangkat keras yang digunakan pada sistem *server* terdiri dari beberapa rangkaian berikut:

- Rangkaian *Modem Power Line Carrier (PLC)*
- Rangkaian Mikrokontroler ATmega8515
- Rangkaian Konverter Level Tegangan
- Rangkaian Selektor

3.4.1. Rangkaian *Modem Power Line Carrier (PLC)*

Rangkaian *Modem Power Line Carrier (PLC)* ini berbasis IC LM1893. Pada gambar dibawah ini ditunjukkan rangkaian *Modem PLC* dengan beberapa komponen pendukungnya. Besarnya nilai dari masing-masing komponen ditentukan berdasarkan *datasheet* IC LM1893.



Gambar 3.3 Rangkaian *Modem PLC*

3.4.1.1. Pemilihan komponen bagian pemancar (*Transmitter*)

Komponen yang mendukung *Modem PLC* saat berfungsi sebagai pemancar antara lain C_O, R_O, C_A, R_A, T_1, C_Q, C_C, Z_T, R_T dan D_T. Penentuan nilai dari masing-masing komponen tersebut dijelaskan sebagai berikut:

- **C_O**

C_O berfungsi untuk menentukan frekuensi pembawa (*carrier*) yang digunakan dalam komunikasi. Penentuan nilai C_O berdasarkan pada *datasheet* yang ditunjukkan pada Gambar 2.5.

Frekuensi *carrier* yang digunakan adalah 125kHz. Dengan persamaan 2-2 nilai C_O dapat diperoleh dengan cara berikut:

$$C_O = \frac{70.6 \times 10^{-6}}{F_O}$$

$$C_O = \frac{70.6 \times 10^{-6}}{125 \times 10^{3}} = 0{,}5648 \times 10^{-9} F = 564{,}8 pF$$

Dari hasil perhitungan tersebut, nilai C_O yang digunakan sebesar 560pF disesuaikan dengan komponen yang ada dipasaran.

- **R_O**

Resistor R_O berfungsi untuk menghasilkan arus V_{BE}/R yang akan dikalikan 2 untuk menghasilkan arus kendali ICO 200μA yang menentukan nilai F_O. Semakin kecil nilai R_O akan meningkatkan F_O. Nilai yang dianjurkan untuk R_O adalah antara 5,6kΩ sampai 7,6 kΩ. Karena potensiometer dengan koefisien temperatur lebih rendah relatif mahal, dianjurkan R_O dibuat dari Resistor 5,6kΩ seri dengan potensiometer 2kΩ.

- **C_A dan R_A**

Besarnya nilai C_A dan R_A akan mempengaruhi ALC IC LM1893. Semakin kecil nilai C_A dan R_A maka ALC akan kurang stabil. Sedangkan semakin besar nilai C_A dan R_A ALC semakin stabil tetapi respon akan lambat. Nilai yang dianjurkan untuk $C_A = 0{,}1\ \mu F$ dan $R_A = 10\ k\Omega$.

- **T_1**

T_1 digunakan bersama C_Q untuk membentuk frekuensi resonansi F_Q. T_1 juga digunakan sebagai *transformator coupling* pada penumpangan sinyal informasi ke jala-jala listrik. Nilai induktansi kumparan primer L_1 bersama-sama dengan C_Q akan menentukan frekuensi resonansi. T_1 juga harus mempunyai perbandingan jumlah lilitan kumparan primer dan sekunder sebesar mungkin agar dapat menumpangkan sinyal ke jala-jala listrik dengan baik. Dengan pertimbangan tersebut maka untuk T_1 dianjurkan menggunakan *transformator* tipe Toko 707VX-A042YUK yang mempunyai induktansi kumparan primer sebesar 49 μH dan induktansi kumparan sekunder sebesar 0,98 μH. Karena tipe *transformator* tersebut susah didapatkan di pasaran, maka untuk T_1 menggunakan trafo IF warna putih yang mempunyai induktansi kumparan primer sebesar 60 μH dan induktansi kumparan sekunder sebesar 1 μH. Nilai induktansi trafo IF tersebut diperoleh dari hasil pengukuran menggunakan LCR meter.

- **C_Q**

C_Q digunakan bersama T_1 sebagai frekuensi resonansi yang sama dengan F_O supaya sinyal informasi dapat ditumpangkan ke jala-jala listrik. Frekuensi

resonansi yang dihasilkan oleh C_Q dan induktansi T_1 harus sama dengan F_O. Penentuan nilai C_Q didasarkan pada gambar 2.6.

Dari gambar 2.6 dapat dihitung nilai C_Q menggunakan persamaan 2-3. dengan L_1 adalah nilai induktansi primer *transformator* T_1 sebesar 49μH, maka nilai C_Q adalah:

$$C_Q = \frac{1}{(2\pi F_O)^2 L_1}$$

$$C_Q = \frac{1}{(2\pi \times 125.10^3)^2 \times 49.10^{-6}}$$

$$C_Q = 33{,}09.10^{-9} F = 33{,}09nF$$

Sehingga nilai kapasitor tangki C_Q yang digunakan sebesar 33nF.

- **C_C**

Fungsi utama dari kapasitor C_C adalah untuk menghalangi tegangan jala-jala dari lilitan T_1 sisi saluran listrik. Kapasitor C_C dan lilitan T_1 juga berfungsi sebagai LC *highpass filter.* C_C harus mempunyai impedansi yang cukup rendah pada saat F_O untuk memperbolehkan T_1 mengarahkan energi yang ditransmisikan ke dalam saluran. Untuk mengarahkan saluran listrik 14Ω, impedansi dari C_C haruslah dibawah 14Ω.

Nilai C_C yang dianjurkan sebesar 0,22μF. Dengan persamaan 2-1 dapat diketahui nilai Impedansi Reaktif dari C_C dengan perhitungan sebagai berikut:

-Nilai Impedansi Reaktif Z_C untuk frekuensi jala-jala 50Hz:

$$Z_R = \frac{1}{2\pi \times f \times C_C}$$

$$Z_R = \frac{1}{2\pi \times 50 \times 0{,}22.10^{-6}}$$

$$Z_R = 14468{,}631\Omega = 14{,}47K\Omega$$

-Nilai Impedansi Reaktif Z_R untuk frekuensi *carrier* 125kHz:

$$Z_R = \frac{1}{2\pi \times f \times C_C}$$

$$Z_R = \frac{1}{2\pi \times 125.10^3 \times 0{,}22.10^{-6}}$$

$$Z_R = 5{,}79\Omega$$

Dari perhitungan tersebut dapat diartikan bahwa dengan nilai C_C sebesar 0,22μF, tegangan jala-jala listrik dapat diblok karena tingginya nilai impedansi pada saat frekuensi jala-jala 50Hz. Dan dengan nilai kapasitor tersebut, impedansi reaktif dari C_C pada saat frekuensi osilasi cukup rendah (5,79Ω) atau lebih kecil daripada impedansi saluran (14Ω). Berdasarkan Gambar 2.8, untuk nilai C_C sebesar 0,22μF diperoleh arus frekuensi-saluran listrik (*line-frequency current*) sebesar 10mA sesuai dengan rekomendasi dalam *datasheet*

- **Z_T**

Z_T merupakan dioda zener yang digunakan untuk mencegah adanya sinyal transien yang melalui *transformator* T_1 menuju ke pin *I/O Carrier*. Sinyal transien dapat berasal transien saluran listrik atau transien yang disebabkan dari pengosongan muatan kapasitor C_C pada saat menghubungkan atau melepaskan rangkaian dari jala-jala listrik. Untuk proteksi diri, *I/O Carrier* mempunyai tegangan *clamp* internal 44V dengan resistansi seri 20Ω. Z_T adalah dioda *avalance* yang dirancang khusus untuk mencegah sinyal transien. Z_T dianjurkan mempunyai tegangan dadal (*Breakdown Voltage*) antara 44–49V, karena pin *carrier* I/O mempunyai dioda zener internal dengan *Breakdown Voltage* 44V.

- **R_T**

R_T berfungsi sebagai pembagi tegangan dengan Z_T, yang akan menyerap tegangan transien supaya tegangan yang masuk pada pin *I/O Carrier* tetap dibawah 44V. Nilai yang dianjurkan sebesar 47Ω, ¼ Watt.

- **D_T**

Dioda schottky ini diletakkan secara paralel dengan dioda substrat *Chip PLC* untuk melewatkan sebagian besar arus yang berasal dari ground ketika *Carrier Input* atau *Carrier Output* ditarik dibawah ground hingga lebih besar dari dua kali *supply-swing* pada tangki. Z_T juga paralel dengan dioda substrat, namun kurang efektif terkait dengan *forward voltage drop*-nya yang tinggi dan kapasitansi difusi tinggi, yang disebabkan oleh *forward speed*-nya yang rendah. Dianjurkan untuk menggunakan 1N5818 yang terbukti menjaga fungsional alur penerimaan dengan penguatan *transmitter* 20X untuk trafo 7:1 yang berusaha untuk mengayunkan *carrier* I/O *receiver* hingga ±100V (300mA arus *peak ground* pada *receiver*).

3.4.1.2. Pemilihan komponen bagian penerima (*Receiver*)

Komponen yang sama-sama digunakan baik untuk *transmitter* maupun *receiver* adalah C_C, T_1, C_Q, R_T, Z_T, C_O, R_O dan periperal *supply* serta komponen-komponen bias yang tidak perlu perubahan untuk mode operasi R_X (penerimaan). Komponen pendukung khusus bagian penerima terdiri dari C_L, C_F, R_F, C_I dan Z_A. Penjelasan fungsi dan nilai masing-masing komponen pendukung bagian penerima adalah sebagai berikut:

- **C_L**

Norton Input Limiter Amplifier pada LM1893 mempunyai *band pass filter* untuk meningkatkan sensitifitas (selektifitas, kekebalan terhadap noise dan penolakan frekuensi jala-jala) penerimaan sinyal. Frekuensi *cut off* atas ditentukan sebesar 300kHz. Sedangkan frekuensi *cut off* bawah ditentukan oleh C_L. Berdasarkan Gambar 2.9 untuk frekuensi osilasi F_O =125kHz, maka nilai C_L sebesar 0,047μF.

- **C_F dan R_F**

C_F dan R_F adalah komponen yang digunakan pada PLL IC LM1893 sebagai filter untuk menghilangkan noise. Besarnya nilai C_F dan R_F ditentukan oleh frekuensi pembawa (F_O) yang digunakan. Berdasarkan Gambar 2.11 dapat diketahui bahwa nilai C_F sebesar 47nF dan berdasarkan Gambar 2.12 nilai R_F sebesar 3,3kΩ.

- **C_I**

Kapasitor integrator *impulse noise filter* C_I digunakan untuk menghalangi jalan lintasan dari pulsa manapun yang lebih pendek daripada waktu pengisian integrator (*integrator charge)*. Waktu pengisian tersebut, diatur pada nominal waktu ½ bit, merupakan waktu yang diperlukan arus pengisian ± 50μA untuk mengayunkan C_I diatas jangkauan $2V_{BE}$. Waktu pengisian nominal maksimum dianjurkan ½ bit untuk penggunaan kapasitor ±10%, jangkauan suhu simpangan penuh, dan jangkauan arus ketetapan penuh. Gambar 2.12 menunjukkan hubungan

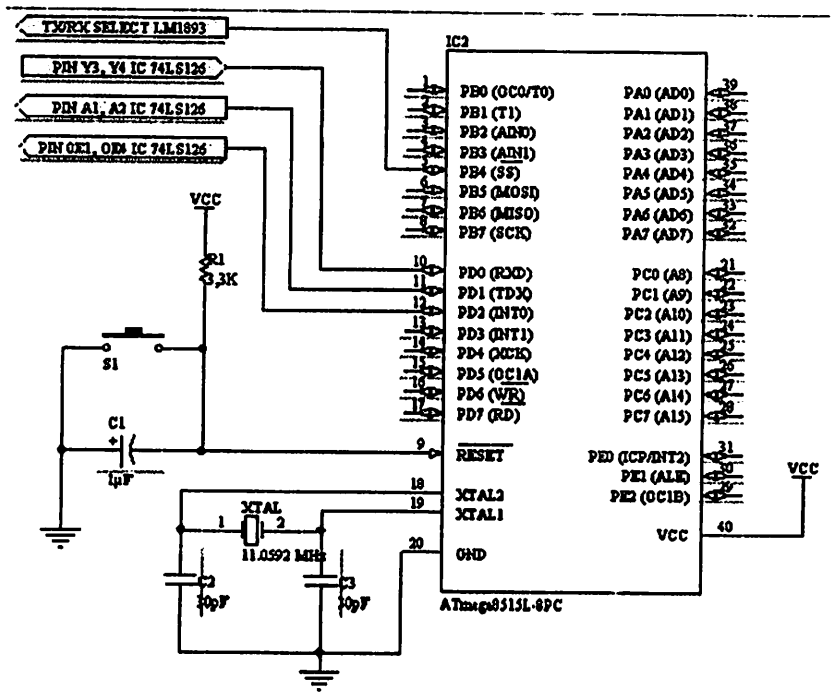
C_I terhadap *datarate* sebagaimana kondisi tersebut di atas. Berdasarkan gambar tersebut untuk frekuensi *datarate* 180Hz diperoleh nilai C_I sebesar 47nF.

- **Z_A**

Dioda zener silikon Z_A sebesar 5,1V diperlukan ketika waktu perpindahan dari RX-ke-TX yang cepat diperlukan pada waktu yang sama bahwa *chip* beroperasi pada mode RX dengan ayunan sinyal pin 10 mendekati atau melebihi dua kali tegangan *supply*. Penyebab yang paling mempengaruhi berkenaan dengan ayunan yang begitu besar pada input RX adalah Tegangan supply *transmitter* lebih tinggi dari pada tegangan *supply receiver*, pasangan TX dan RX yang secara elektris berdekatan, rasio perubahan *step-up* trafo T_1 pada TX lebih tinggi daripada rasio *step-down* pada RX.

3.4.2. Rangkaian Mikrokontroler ATmega8515

Mikrokontroler yang digunakan pada rangkaian ini adalah mikrokontroler dengan tipe ATmega8515 yang merupakan keluarga dari Atmel AVR. Komponen ini merupakan sebuah *chip* tunggal yang berfungsi sebagai pengontrol komunikasi data antara *server* dengan kWh meter prabayar. Pemilihan mikrokontroller ini karena mudah diperoleh di pasaran. Pin-pin mikrokontroler ATmega8515 dihubungkan pada rangkaian seperti yang ditunjukkan dalam Gambar 3.4 di bawah ini.



Gambar 3.4 Rangkaian Mikrokontroller ATmega8515

Pin-pin yang digunakan yaitu:

1. Port B

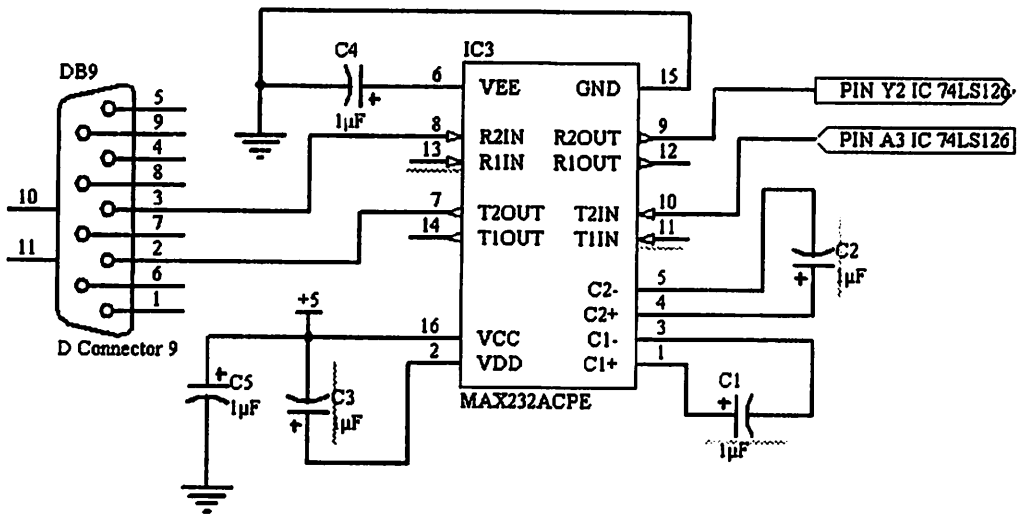
- Port B.4 (Pin 5) digunakan sebagai pemilih mode *transmit (TX)* atau *receive (RX)* pada *Modem Power Line Carrier (PLC).*
- Port B.7 (Pin 8) digunakan sebagai pengontrol selektor untuk memilih hubungan antara mikrokontroler dengan PC atau mikrokontroler dengan *Modem Power Line Carrier (PLC).*

2. Port D

- Port D.0 (pin10)digunakan sebagai masukan data serial baik data dari PC atau data dari kWh meter pelanggan prabayar pada rumah susun.
- Port D.1 (pin 11) digunakan sebagai keluaran data serial baik data dari PC atau data dari kWh meter pelanggan prabayar pada rumah susun.

3.4.3. Rangkaian Konverter Level Tegangan

Port serial komputer bekerja pada level tegangan RS-232, sehingga membutuhkan rangkaian pengubah level tegangan ke TTL agar sesuai dengan level tegangan yang digunakan oleh mikrokontroler. Rangkaian konverter atau pengubah level tegangan ini berbasis IC *multichannel driver/receiver* RS-232 MAX232A buatan Maxim Integrated Product. MAX232A memiliki konfigurasi pin yang sama dengan MAX220 dan MAX232. IC MAX232A memiliki 2 buah *channel driver/receiver* RS-232. Bentuk rangkaian konverter level tegangan RS232-TTL ditunjukkan pada Gambar 3.5.

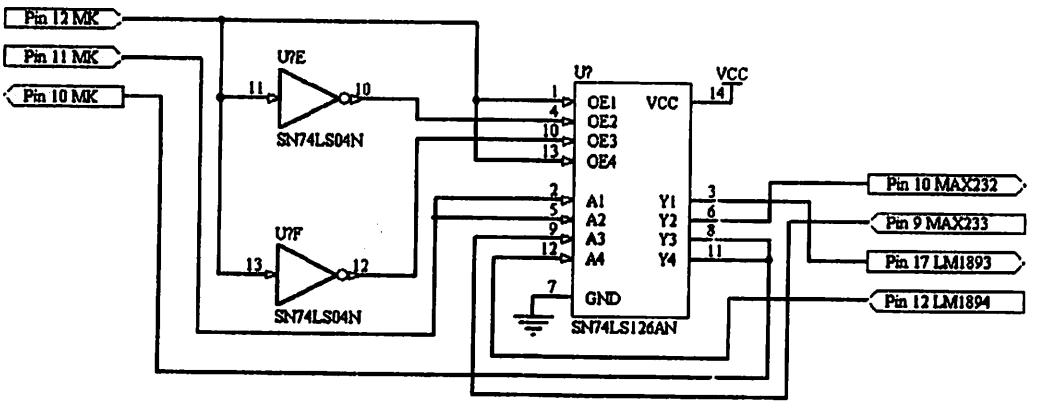


Gambar 3.5 Rangkaian konverter level tegangan RS232-TTL

Pada gambar tersebut terlihat bahwa pin 8 (R2IN) IC MAX232A terhubung dengan pin 3 (TX) Port Serial PC agar sinyal dengan level tegangan RS232 dirubah ke level tegangan TTL pada pin R2OUT untuk dikirimkan ke mikrokontroler. Pin 7 (T2OUT) IC MAX232A terhubung dengan pin 2 (RX) port serial PC agar dapat menerima sinyal dengan level tegangan RS232 dari hasil

perubahan sinyal masukan dari mikrokontroler dengan level tegangan TTL pada pin 10 (T2IN).

3.4.4. Rangkaian Selektor



Gambar 3.6 Rangkaian Selektor

Rangkaian selektor berfungsi untuk memilih hubungan mikrokontroler dengan PC atau mikrokontroler dengan *modem Power Line Carrier* (*PLC*). Pada Gambar 3.6 terlihat bahwa rangkaian selektor ini menggunakan IC *Bus Buffer* 74LS126 dan *Inverter* 74LS04.

Pin A1 – A4 IC 74LS126 berfungsi sebagai jalur masukan data serial yang akan dikirim dan pin Y1 – Y4 berfungsi sebagai jalur keluaran data serial apabila pin pengontrol gerbang OE1 – OE4 berlogika 1. Pin OE(Output Enable) yang terhubung dengan pin 12 mikrokontroler adalah pin 1 dan 13. Sedangkan pin pin 4 dan 10 dilewatkan sebuah inverter terlebih dahulu untuk menghasilkan logika yang berlawanan.

Sehingga apabila logika 1 dikirimkan dari pin 12 mikrokontroler maka akan memperbolehkan jalur data serial antara mikrokontroler dengan IC LM1893.

sebaliknya apabila logika 0 dikirimkan dari pin 12 mikrokontroler maka akan memperbolehkan jalur data serial antara mikrokontroler dengan IC MAX232 atau PC.

3.5. Perancangan Protokol Komunikasi

Protokol komunikasi adalah sebuah aturan atau standar yang mengatur atau mengijinkan terjadinya hubungan, komunikasi, dan perpindahan data antara dua atau lebih titik komputer. Protokol dapat diterapkan pada perangkat keras, perangkat lunak atau kombinasi dari keduanya. Pada tingkatan yang terendah, protokol mendefinisikan koneksi perangkat keras.

Agar terjadi komunikasi antara sistem *server* dengan kWh meter prabayar pada rumah susun, dibutuhkan suatu protokol komunikasi. Protokol komunikasi dirancang sebagai berikut:

1. Proses komunikasi dimulai dengan PC mengirim data ke mikrokontroler yang terdiri dari nomor ID kWh meter pelanggan yang dituju, kode perintah, nilai nominal kWh dan validasi.
2. PC menunggu respon dari mikrokontroler berupa kode apakah mikrokontroler dapat menerima data dengan benar atau tidak. Jika dalam waktu tertentu tidak ada respon, maka PC menganggap terjadi *error* pada hubungan komunikasi dengan mikrokontroler dan proses tidak dilanjutkan.
3. Jika data yang dikirim oleh PC tidak sama dengan data yang diterima oleh mikrokontroler yang ditandai dengan ketidaksamaan data validasi, maka mikrokontroler akan memberikan kode kepada PC yang menyatakan bahwa data yang diterima mikrokontroler salah, dan PC akan mengirim data kembali.

Jika dalam dua kali pengiriman, data yang diterima mikrokontroler masih tidak sama dengan dengan yang dikirim oleh PC, maka PC menganggap *error* pada hubungan komunikasi dengan mikrokontroler dan proses tidak dilanjutkan.

4. Jika data dapat diterima oleh mikrokontroler dengan benar, maka mikrokontroler mengirim data tersebut dikirim ke kWh meter pelanggan ditambah dengan *header* dan nomor ID *Server*. Saat ini PC dalam kondisi menunggu pengiriman data dari mikrokontroler berupa hasil proses komunikasi dengan kWh meter pelanggan.
5. Mikrokontroler menunggu jawaban dari kWh meter pelanggan berupa kode yang menyatakan bahwa data sudah diterima dengan benar atau tidak, serta saldo pulsa atau nominal kWh yang tersimpan. Jika dalam waktu tertentu tidak ada jawaban, maka mikrokontroler menganggap terjadi *error* pada komunikasi dengan kwh meter pelanggan dan akan mengirim data ke PC berupa kode yang menyatakan bahwa terjadi *error* dalam komunikasi agar proses tidak dilanjutkan.
6. Jika data yang dikirim oleh mikrokontroler tidak sama dengan data yang diterima oleh kWh meter pelanggan maka dapat ditandai dengan adanya perbedaan kode validasi, sehingga kWh meter pelanggan akan mengirimkan kode ke mikrokontroler yang menyatakan bahwa data yang diterima salah dan mikrokontroler akan mengirimkan data lagi. Jika dalam 2 kali pengiriman data yang dikirim yang diterima oleh kWh meter pelanggan masih salah maka mikrokontroler menganggap terjadi *error* dalam hubungan komunikasi. Sehingga mikrokontroler akan mengirim data ke PC berupa kode yang

menyatakan bahawa terjadi *error* dalam komunikasi dengan kWh meter pelanggan dan proses dihentikan.

7. Jika data dapat diterima dengan benar oleh kWh meter pelanggan maka data saldo pulsa pelanggan dikirimkan ke mikrokontroler pada sistem *server*. Jika data yang diterima mikrokontroler tidak sama dengan data yang dikirim kWh meter pelanggan akan ditandai dengan perbedaan validasi dan mikrokontroler akan mengirimkan kode ke kWh meter pelanggan yang menyatakan bahwa data yang yang diterima mikrokontroler salah sehingga akan dilakukan pengiriman data sekali lagi. Jika dalam 2 kali pengiriman data yang diterima mikrokontroler salah maka dianggap terjadi *error* pada hubungan komunikasi. Sehingga mikrokontroler akan megirimkan data ke PC berupa kode yang menyatakan terjadi *error* pada hubungan komunikasi dengan kWh meter pelanggan.
8. Jika mikrokontroler sudah menerima data yang dikirim dari kWh meter pelanggan dengan benar, maka mikrokontroler mengirim data data tersebut ke PC untuk disimpan kedalam database. Data terdiri dari nomor ID kWh meter pelanggan prabayar, kode validasi (*error* atau tidak) dan saldo pulsa (nominal kWh). Sampai disini proses telah selesai.

Format data yang digunakan dalam komunikasi antara PC *server* dengan mikrokontroler ditunjukkan pada gambar dibawah ini.

Tujuan	Kode perintah	Atas	Tengah	Bawah	Validasi

Gambar 4.8 Format data untuk komunikasi PC *server* dengan mikrokontroler

Keterangan:

Tujuan = 8 bit alamat tujuan pengiriman data

Kode perintah = 8 bit kode perintah untuk kWh meter prabayar

Atas,tengah dan bawah = 24 bit nominal kWh

Validasi = 8 bit penanda kebenaran pengiriman data

Format data yang digunakan dalam komunikasi antara mikrokontroler dengan kWh meter pelanggan ditunjukkan pada gambar dibawah ini.

3 byte header	Kode perintah	Tujuan	Sumber	Atas	Tengah	Bawah	Validasi

Gambar 4.9 Format data untuk komunikasi mikrokontroler dengan kWh meter

Keterangan:

Kode perintah = 8 bit kode perintah untuk kWh meter prabayar

Tujuan = 8 bit nomor ID kWh meter pelanggan

Sumber = 8 bit nomor ID *server*

Atas,tengah dan bawah = 24 bit nominal kWh

Validasi = 8 bit penanda kebenaran pengiriman data

Format data yang digunakan dalam komunikasi antara mikrokontroler dengan kWh meter pelanggan ditunjukkan pada gambar dibawah ini.

1 byte header	'i'/'e'	Sumber	'n'/'e'	Nominal pulsa (kWh)

Gambar 4.10 Format data untuk komunikasi mikrokontroler dengan PC *server*

Keterangan:

'i/e' = sebagai kode adanya *error* atau tidak di sistem *server*

Sumber = 8 bit nomor ID kWh meter pelanggan

'n'/e = sebagai kode adanya *error* atau tidak pada kWh meter pelanggan

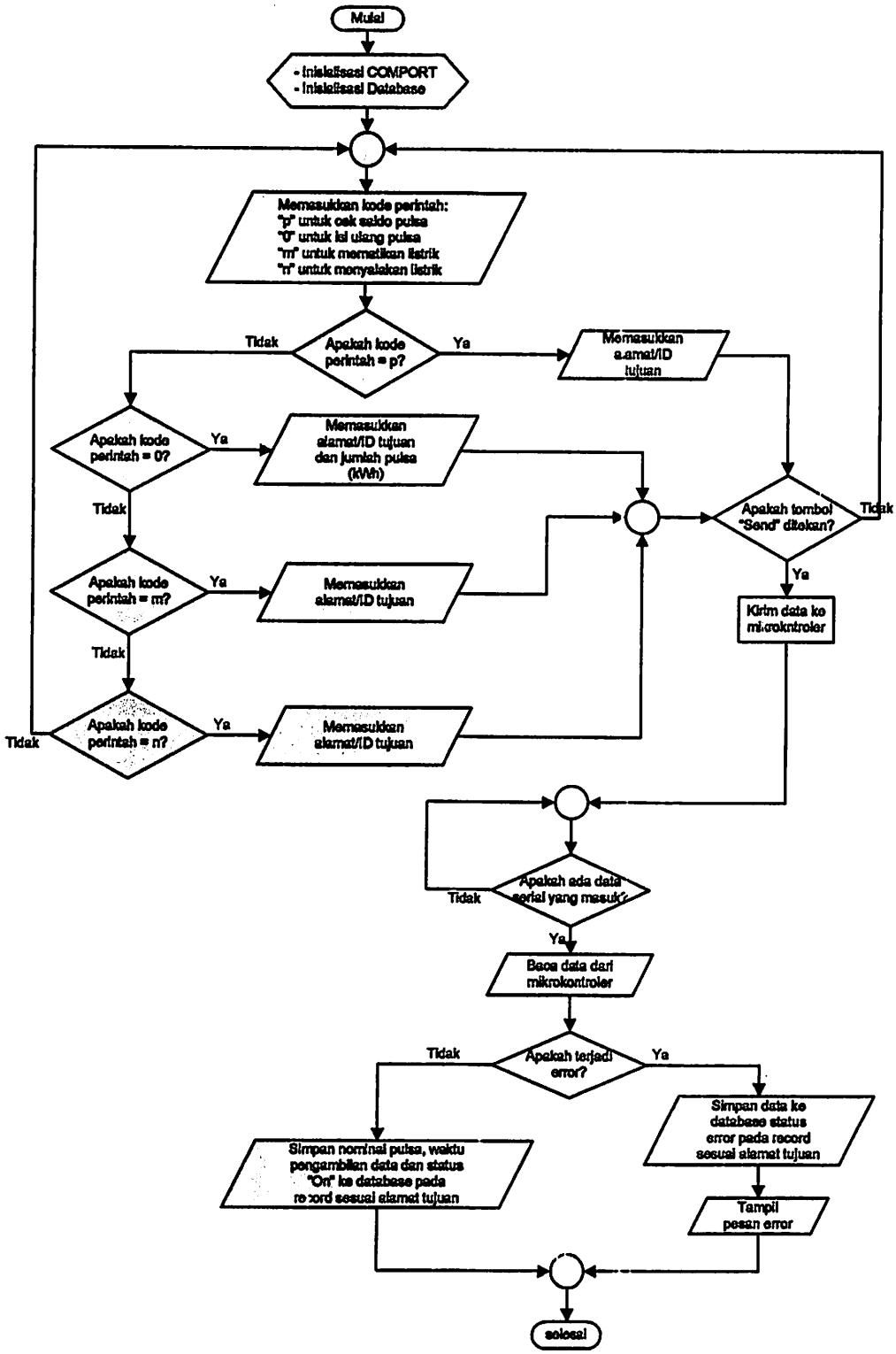
Nominal pulsa(kWh) = nominal pulsa(kWh) dalam bentuk string

3.6. Perancangan Perangkat Lunak

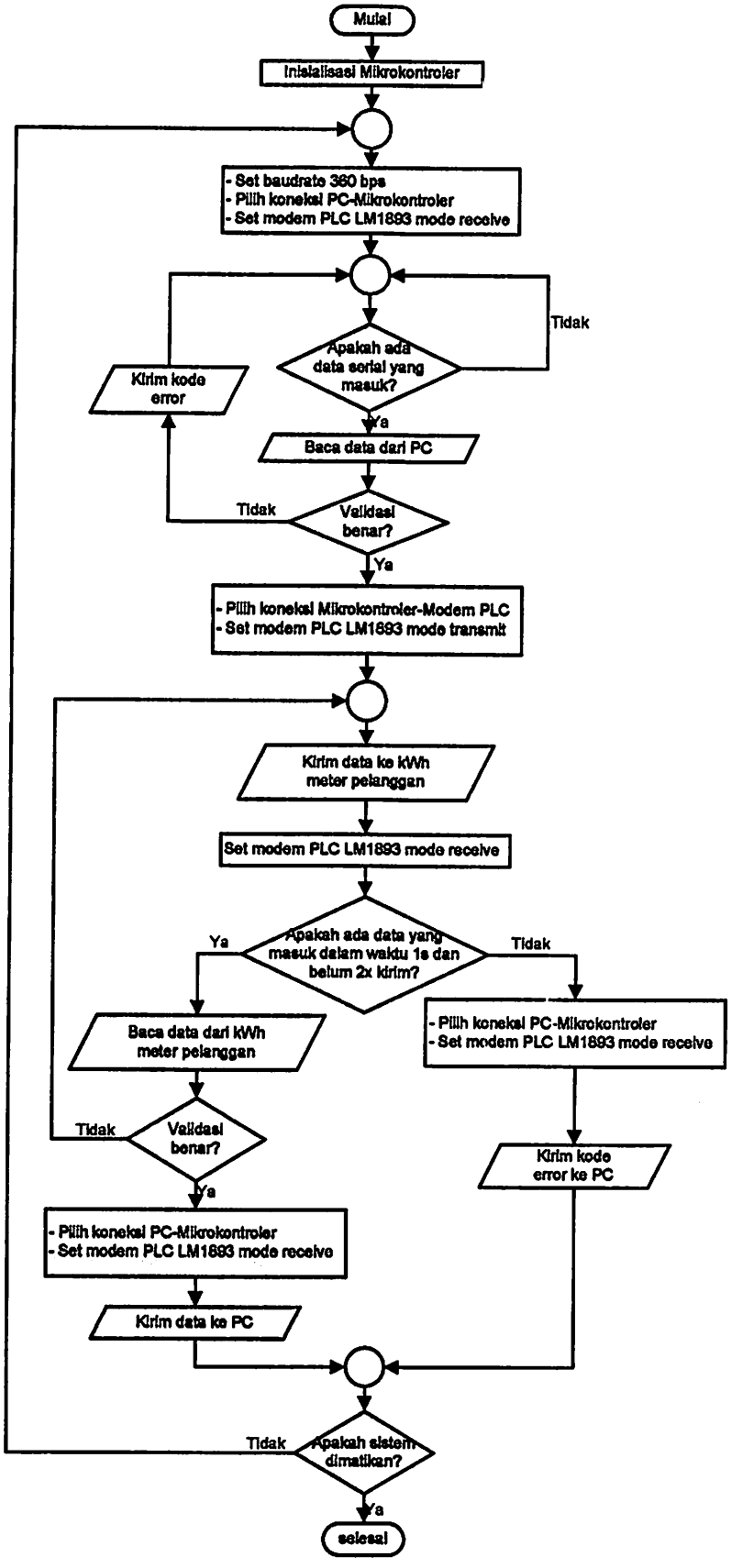
Perangkat lunak yang dirancang terdiri dari perangkat lunak untuk PC dan perangkat lunak untuk mikrokontroler. Pada PC, perangkat lunak digunakan untuk antarmuka peralatan komunikasi dan melayani pelanggan listrik prabayar pada rumah susun. Sedang ada mikrokontroler, perangkat lunak digunakan untuk mengatur komunikasi data dengan kWh meter pelanggan.

Perangkat lunak ini tersusun dari instruksi-instruksi yang membentuk sebuah listing program. Program dibuat secara terstruktur dalam beberapa subrutin yang secara khusus menangani fungsi tertentu agar memudahkan dalam pembuatan dan pencarian kesalahan serta pengujian program sehingga dapat bekerja dengan baik.

Perangkat lunak untuk PC dibuat dengan menggunakan lingkungan pengembangan dan bahasa pemrograman Delphi 7. Untuk perangkat lunak mikrokontroler dibuat dengan menggunakan Code Vision AVR buatan HP info tech. Bahasa pemrograman yang digunakan adalah bahasa C. Perancangan perangkat lunak untuk PC dan mikrokontroler ditunjukkan oleh diagram alir pada Gambar 3.7 dan 3.8.



Gambar 3.7 Diagram alir perangkat lunak untuk PC



Gambar 3.8 Diagram alir untuk perangkat lunak mikrokontroler

BAB IV

PENGUJIAN DAN ANALISIS

Pada bagian ini, dijelaskan mengenai pengujian dan analisis sistem yang telah dibuat untuk mengetahui sejauh mana keberhasilan dari perancangan dan pembuatan sistem. Pengujian dilakukan dengan memberikan masukan sinyal pada blok rangkaian dan mengamati keluarannya. Data hasil pengujian tersebut kemudian dianalisis untuk dievaluasi dan sebagai acuan dalam menarik kesimpulan.

4.1. Pengujian Rangkaian *Modem PLC*

4.1.1. Tujuan

Secara umum, tujuan dari pengujian rangkaian *Modem PLC* LM1893 adalah untuk mengetahui bahwa rangkaian tersebut dapat bekerja dengan baik sesuai dengan spesifikasi yang terdapat dalam *datasheet*. Untuk mencapai tujuan ini, pengujian dilakukan dalam dua keadaan yaitu pengujian pada mode *transmit* dan mode *receive*.

Tujuan dari pengujian rangkaian *Modem PLC* mode *transmit* adalah untuk mengetahui bentuk sinyal, besar tegangan dan frekuensi dari keluaran pada pin 10 ketika pin 17 mendapat masukan berupa logika 0 atau 1. Sedangkan tujuan dari pengujian rangkaian *Modem PLC* dengan mode *receive* adalah untuk mengetahui bentuk logika dari sinyal yang dibangkitkan pada pin 12 ketika mendapat

masukan pada pin 10 berupa sinyal sinusoida dengan frekuensi 122,25kHz dan 127,75kHz.

4.1.2. Peralatan yang digunakan

Peralatan yang digunakan terdiri dari:

1. Osciloscope
2. Voltmeter
3. Frequency Counter
4. Logic Probe
5. Function Generator
6. Catu daya 5V dan 18V

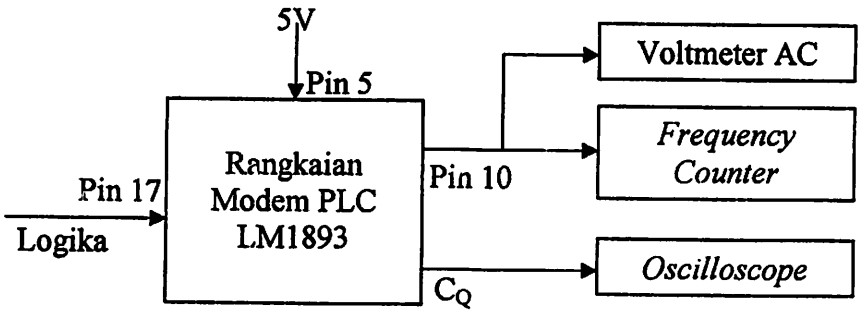
4.1.3. Prosedur Pengujian

Langkah-langkah yang dilakukan untuk pengujian rangkaian *Modem PLC* LM1893 adalah sebagai berikut:

1. Rangkaian di-*tuning* dengan prosedur seperti yang terdapat dalam *datasheet*, yaitu sebagai berikut:
    1. Membuat Pin 17 berlogika rendah.
    2. Membuat Pin 5 berlogika tinggi.
    3. Meletakkan *Frequency counter main probe* pada pin 10 dan *frequency counter common probe* diletakkan pada pin ground.
    4. Mengatur Variabel Resistor (R_O) pada pin 18 sampai *frequency counter* menunjukkan angka sebesar $1{,}022F_O$ yaitu sebesar 127,75kHz.

2. Untuk pengujian rangkaian *Modem PLC* saat mode *transmit*, langkah-langkah yang dilakukan adalah sebagai berikut:

   1. Rangkaian disusun seperti gambar dibawah ini

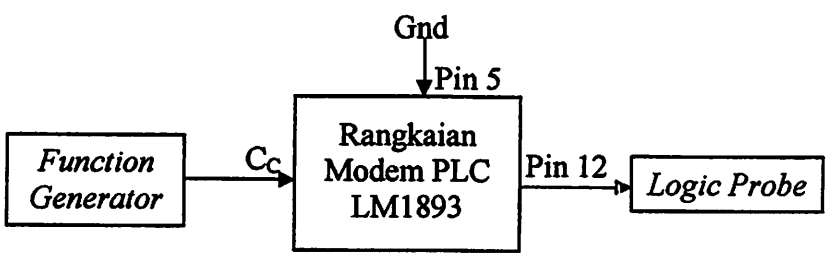


Gambar 4.1 Diagram blok pengujian rangkaian *Modem PLC* mode *transmit*

   2. Membuat Pin 5 berlogika tinggi (1)
   3. Pin 17 diberi masukan berupa logika tinggi (1) dan logika rendah (0)
   4. Mengamati bentuk sinyal keluaran pada pin 10 dengan *osciloscope*, mengukur besar tegangan keluaran dengan *voltmeter* dan mengukur besar frekuensi dengan *frequency counter*.

3. Untuk pengujian rangkaian *Modem PLC* saat mode *receive*, langkah-langkah yang dilakukan adalah sebagai berikut:

   1. Rangkaian disusun seperti gambar berikut.



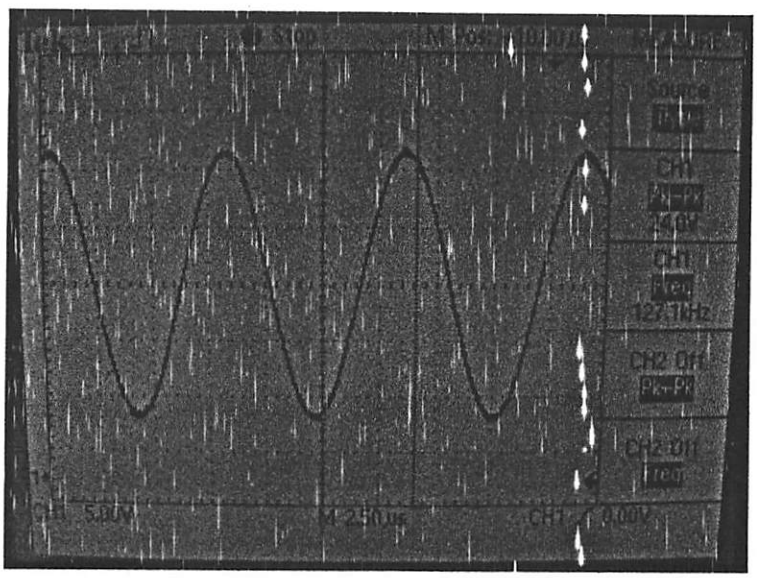
Gambar 4.2 Diagram blok pengujian rangkaian *Modem PLC* mode *receive*

2. Membuat Pin 5 berlogika rendah (0).
3. Memberi masukan pin 10 (*I/O Carrier*) rangkaian *Modem PLC* sinyal sinusoida menggunakan *function generator* dengan tegangan maksimum sebesar 3V berfrekuensi 122,25kHz dan 127,75kHz.
4. Mengamati logika keluaran pada pin 12 dengan *logic probe*.

4.1.4. Data Hasil Pengujian

Data dari hasil pengujian rangkaian *Modem PLC* saat mode *transmit* adalah sebagai berikut:

1. Bentuk sinyal keluaran *transmitter* rangkaian *Modem PLC* ditunjukkan pada gambar dibawah ini.



Gambar 4.3 Sinyal keluaran pin 10 saat mode *transmit*

2. Tegangan keluaran yang diukur melalui pin 10 adalah 17,50 Volt.
3. Frekuensi sinyal keluran pemancar LM1893 dapat dilihat dalam tabel 4.1

Tabel 4.1 Data hasil pengujian rangkaian *Modem PLC* saat mode *transmit*

No	Logika masukan pada pin 17	Frekuensi keluaran (kHz)
1.	1	122,00
2.	0	127,10

Data hasil pengujian rangkaian *Modem PLC* saat mode *receive* dapat dilihat pada Tabel 4.2.

Tabel 4.2 Data hasil pengujian rangkaian *Modem PLC* saat mode *receive*

No	Frekuensi sinyal masukan (kHz)	Logika keluaran pada pin 12
1.	122,25	1
2.	127,75	0

4.1.5. Analisis Data Hasil Pengujian

Saat mode *transmit*, data pengujian menunjukkan bahwa sinyal keluarannya berupa sinyal sinusoida, tegangan sinyal keluarannya sebesar 17,50V dan frekuensi sinyal keluarannya adalah sebesar 127,10kHz saat diberi masukan logika 0 dan 122,00kHz saat diberi masukan logika 1. Sedangkan spesifikasi yang ditunjukkan oleh *datasheet* adalah sinyal keluaran berupa sinusoida, tegangan sinyal minimal sebesar 4,0V dan maksimal 5,7V. Untuk frekuensi sinyal keluaran sebesar 127,75kHz saat diberi masukan logika 0 dan 122,25kHz saat diberi masukan logika 1. Toleransi penyimpangan frekuensi sinyal keluaran yang diijinkan adalah sebesar 20%. Oleh karena itu dapat disimpulkan bahwa saat mode *transmit*, rangkaian ini dapat bekerja dengan spesifikasi yang ditunjukkan oleh *datasheet*.

Saat mode *receive*, data pengujian menunjukkan bahwa logika keluarannya adalah *high* saat diberi masukan berupa sinyal sinusoida dengan frekuensi sebesar 122,25kHz dan *low* saat diberi masukan berupa sinyal sinusoida dengan frekuensi 127,75kHz. Data hasil pengujian tersebut sama dengan spesifikasi yang ditunjukkan oleh *datasheet* sehingga dapat disimpulkan bahwa rangkaian dapat bekerja dengan baik saat mode *receive*.

4.2. Pengujian komunikasi antar rangkaian mikrokontroler menggunakan rangkaian *Modem PLC*

4.2.1. Tujuan

Tujuan pengujian komunikasi antar mikrokontroler menggunakan rangkaian *Modem PLC* adalah untuk mengetahui nilai *baudrate* yang bisa digunakan agar tercapai keberhasilan komunikasi data antara mikrokontroler *master* dan mikrokontroler *slave*.

4.2.2. Peralatan yang digunakan

Peralatan yang digunakan terdiri dari:

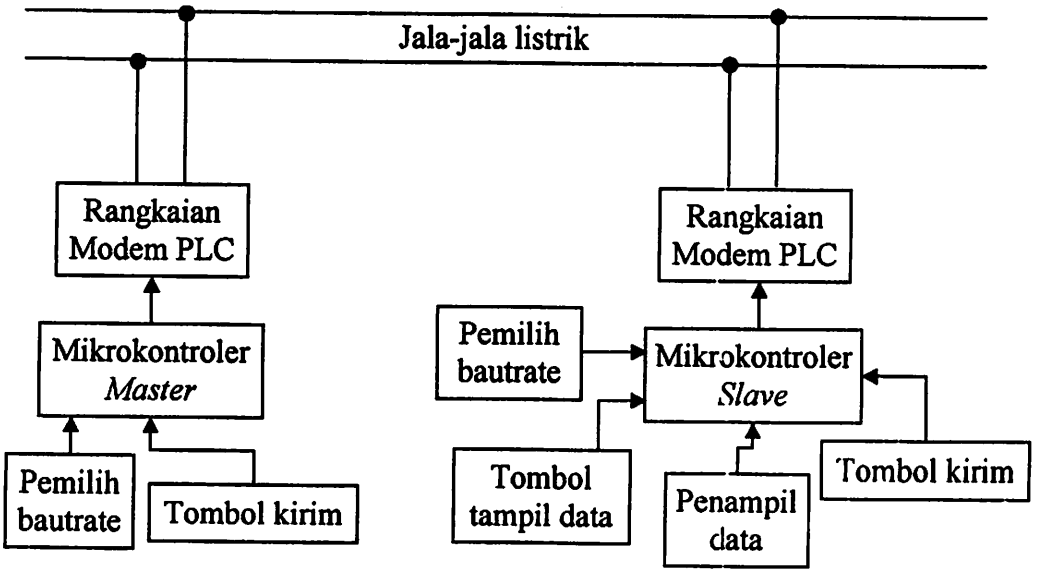
1. Rangkaian *Modem PLC* yang telah di-*tuning*
2. *Minimum system* mikrokontroler ATMEGA8515
3. Rangkaian indikator LED
4. Rangkaian *Dip Switch-8*
5. Rangkaian *push button*
6. Catu daya 5V dan 18V
7. *Software* CodeVisionAVR

8. *Software* PonyProg2000

4.2.3. Langkah-langkah pengujian

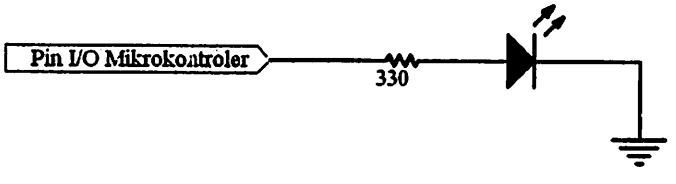
Langkah-langkah yang dilakukan untuk pengujian komunikasi antar mikrokontroler menggunakan rangkaian *Modem PLC* adalah sebagai berikut:

1. Rangkaian disusun seperti Gambar 4.4



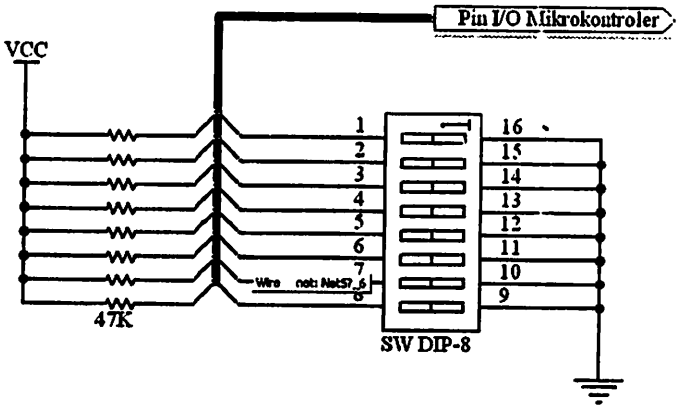
Gambar 4.4 Diagram blok pengujian komunikasi antar mikrokontroler menggunakan rangkaian *Modem PLC*

Penampil data dan indikator nomor urut data masing-masing terdiri dari 8 buah rangkaian LED. Skema rangkaian LED yang digunakan ditunjukkan pada gambar 4.5.



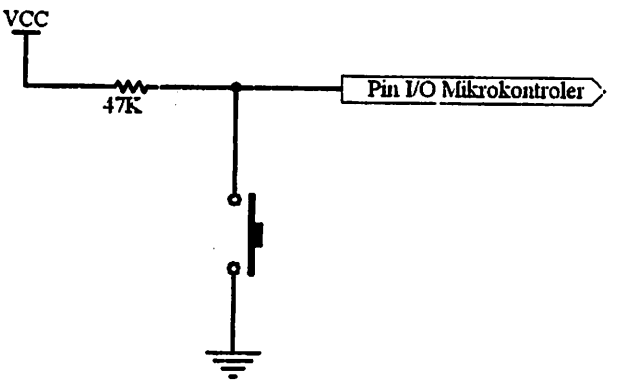
Gambar 4.5 Rangkaian LED

Pemilih *bautrate* menggunakan rangkaian DIP SW-8. Rangkaian ini ditunjukkan pada Gambar 4.6 dibawah ini.



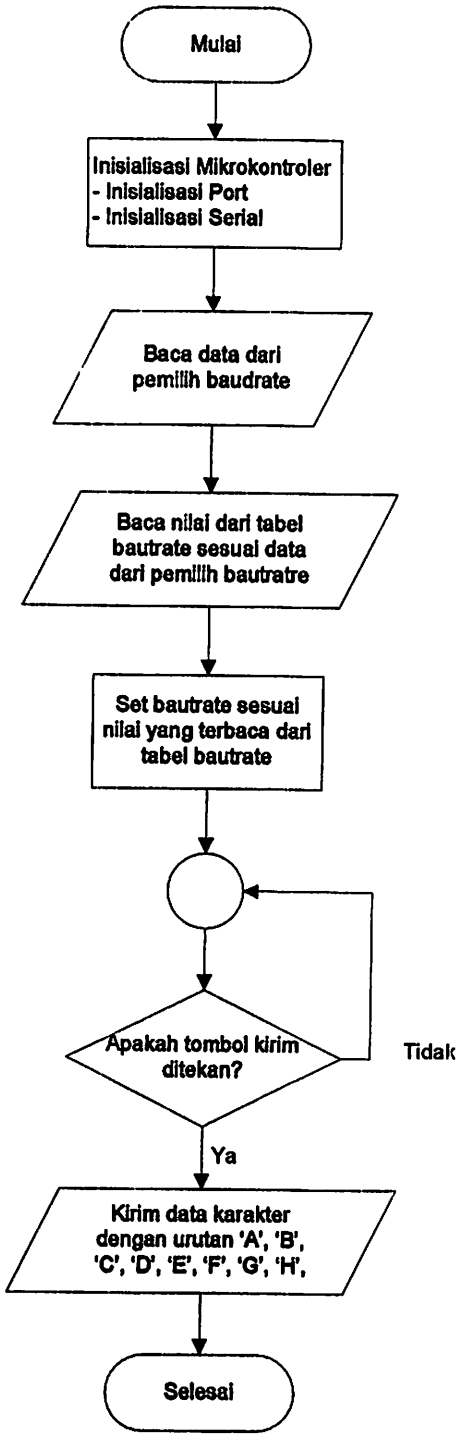
Gambar 4.6 Rangkaian SW DIP-8

Tombol kirim dan tombol tampil data menggunakan rangkaian *Push Button* seperti yang ditunjukkan pada gambar 4.7 berikut ini.

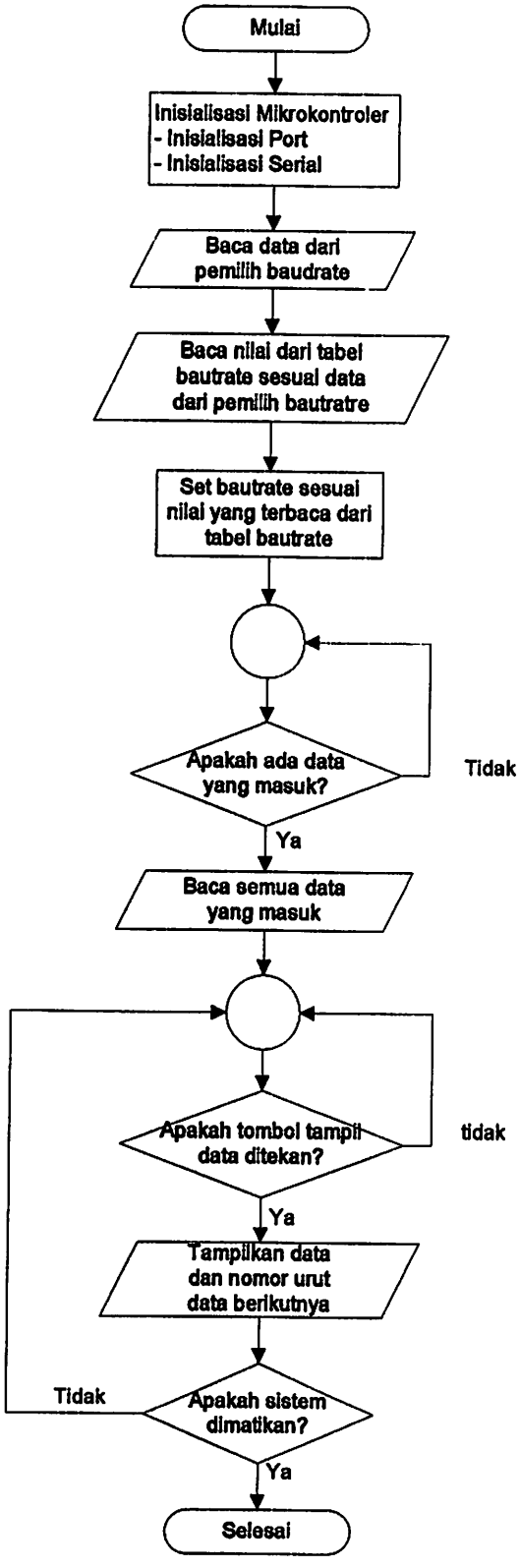


Gambar 4.7 Rangkaian *Push Button*

2. Program pada mikrokontroler *master* dibuat dengan algoritma seperti ditunjukkan pada Gambar 4.8. Sedangkan program pada mikrokontroer *slave* dibuat dengan algoritma seperti yang ditunjukkan oleh Gambar 4.9.



Gambar 4.8 Diagram blok program mikrokontroler *master*



Gambar 4.9 Diagram blok program mikrokontroler *slave*

3. Menetapkan *bautrate* pada mikrokontroler *master* dan *slave* ditetapkan melalui pemilih *bautrate*.
4. Menekan tombol kirim data pada mikrokontroler *master*.
5. Menekan tombol tampil data pada mikrokontroler *slave* untuk melihat tampilan data pertama, kemudian menekan lagi untuk melihat tampilan data kedua dan seterusnya.
6. Mengamati indikator nomor data dan data yang diterima melalui LED indikator nomor data LED penampil data, dan memasukan data pada Tabel 4.4.

4.2.4. Data hasil pengujian

Pada pengujian dilakukan pengirimkan data berupa karakter A – H dari mikrokontroler *master* ke mikrokontroler *slave* menggunakan rangkaian *Modem PLC*.

Tabel 4.3 Data yang dikirim oleh mikrokontroler *master*

Nomor Urut	Karakter yang dikirim	Data yang dikirim (heksa)
1	A	41
2	B	42
3	C	43
4	D	44
5	E	45
6	F	46
7	G	47
8	H	48

Tabel 4.4 Data yang diterima mikrokontroler *slave*

No	*Bautrate* (bps)	Data yang diterima (Heksa) dengan nomor urut							
		1	2	3	4	5	6	7	8
1	60	41	42	43	44	45	46	47	48
2	120	41	42	43	44	45	46	47	48
3	180	41	42	43	44	45	46	47	48
3	240	41	42	43	44	45	46	47	48
5	300	41	42	43	44	45	46	47	48
6	360	41	42	43	44	45	46	47	48
7	480	41	42	43	44	45	46	47	48
8	540	41	42	43	44	45	46	47	48
9	600	41	42	43	44	45	46	47	48
10	720	41	42	43	44	45	46	47	48
11	900	41	42	43	44	45	46	47	48
12	960	41	42	43	44	45	46	47	48
13	1080	F0	3C	F0	3C	00	30	80	78
14	1200	C0	00	80	C0	00	00	00	80
15	1440	00	00	C0	00	00	00	C0	00
16	1880	00	00	C0	00	00	00	C0	00
17	1920	00	00	C0	00	00	00	C0	00
18	2160	00	00	C0	00	00	00	C0	00
19	2400	00	00	00	00	00	00	00	00

4.2.5. Analisis data hasil pengujian

Berdasarkan data hasil pengujian yang ditunjukkan pada tabel 4.3 dan 4.4, kesamaan data yang dikirim dan data yang diterima terjadi hanya sampai *baudrate* sebesar 960 bps. Sehingga dapat disimpulkan bahwa pemilihan *baudrate* sebesar 360 bps pada perancangan sudah sesuai untuk digunakan dalam komunikasi data. *Baudrate* yang dipilih adalah *baudrate* yang tidak terlalu cepat dan tidak terlalu lambat supaya dapat dihasilkan kinerja sistem yang paling optimal.

4.3. Pengujian Rangkaian Konverter Level Tegangan RS232-TTL

4.3.1. Tujuan

Tujuan pengujian rangkaian ini adalah untuk mengetahui besarnya level tegangan keluaran dari rangkaian konverter level tegangan RS232-TTL saat mendapatkan masukan berupa sinyal dengan level tegangan TTL atau RS232.

4.3.2. Peralatan yang digunakan

Peralatan yang digunakan antara lain:

1. Voltmeter
2. Rangkaian Konverter level tegangan dengan IC MAX232
3. Catu daya +5V, -15V dan +15V

4.3.3. Langkah-langkah pengujian

1. Untuk pengujian dengan level tegangan masukan logika TTL, rangkaian disusun seperti dalam Gambar 4.10.

Gambar 4.10 Diagram blok pengujian dengan level tegangan masukan logika TTL

2. Untuk pengujian dengan level tegangan masukan RS232, rangkaian disusun seperti berikut.

Gambar 4.11 Diagram blok pengujian dengan level tegangan masukan logika TTL

3. Mengukur tegangan keluaran IC MAX232 dari masing-masing pengujian dengan menggunakan Voltmeter.

4.3.4. Data Hasil Pengujian

Data pengujian dengan masukan logika TTL dapat dilihat pada tabel 4.5.

Tabel 4.5 Data pengujian dengan masukan logika TTL

Tegangan Masukan (Volt)	Tegangan Keluaran (Volt)
0	8,15
5	-8,51

Data pengujian dengan masukan logika RS232 dapat dilihat pada tabel 5.5.

Tabel 4.6 Data pengujian dengan masukan logika TTL

Tegangan Masukan (Volt)	Tegangan Keluaran (Volt)
+15	0
-15	4,99

4.3.5. Analisa data hasil pengujian

Data hasil pengujian pada tabel 4.5 menunjukkan bahwa saat diberi tegangan masukan sebesar 0V menghasilkan keluaran dengan nilai yang berada dalam *range* 3V – 15V, dan saat diberi tegangan masukan sebesar 5V menghasilkan tegangan keluaran dengan nilai yang berada dalam *range* (-3V) – (-15V). Data pengujian pada tabel 4.6 menunjukkan bahwa saat diberi tegangan masukan sebesar 15V menghasilkan tegangan keluaran berlogika rendah, dan saat diberi masukan sebesar -15V menghasilkan tegangan keluaran berlogika tinggi. Sehingga dapat disimpulkan bahwa rangkaian hasil perancangan mampu mengubah level tegangan RS232 menjadi tegangan TTL atau sebaliknya.

4.4. Pengujian komunikasi antara mikrokontroler dengan PC

4.4.1. Tujuan

Tujuan pengujian komunikasi antara mikrokontroler dengan PC adalah untuk mengetahui keberhasilan komunikasi data bertipe karakter antara mikrokontroler dengan PC.

4.4.2. Peralatan yang digunakan

Peralatan yang digunakan terdiri dari:

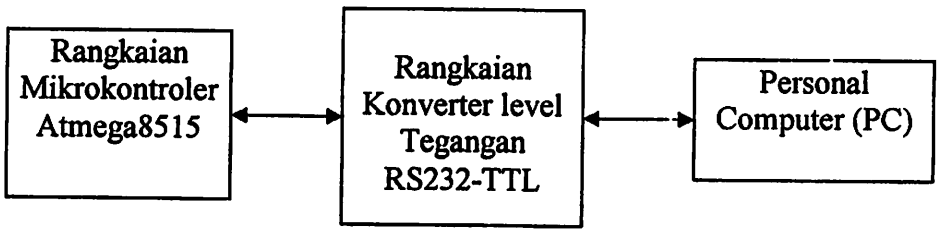
1. Personal komputer (PC)
2. *Minimum system* mikrokontroler ATmega8515
3. Rangkaian konverter level tegangan RS232-TTL
4. Program Hyperterminal
5. *Software* CodeVision AVR

6. *Software* Ponyprog2000

4.4.3. Langkah-langkah pengujian

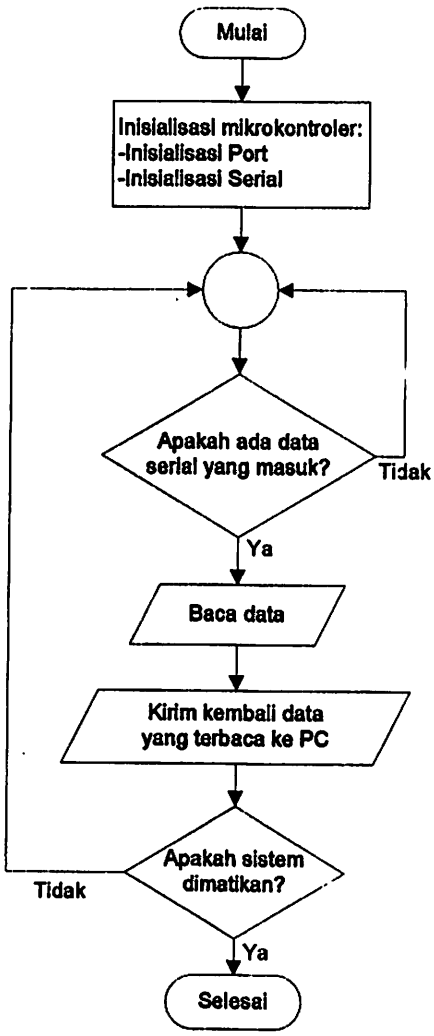
Langkah-langkah untuk menguji komunikasi antara PC dengan mikrokontroler adalah sebagai berikut:

1. Menyusun rangkaian seperti gambar di bawah ini.



Gambar 4.12 Diagram blok komunikasi antara mikrokontroler dengan PC

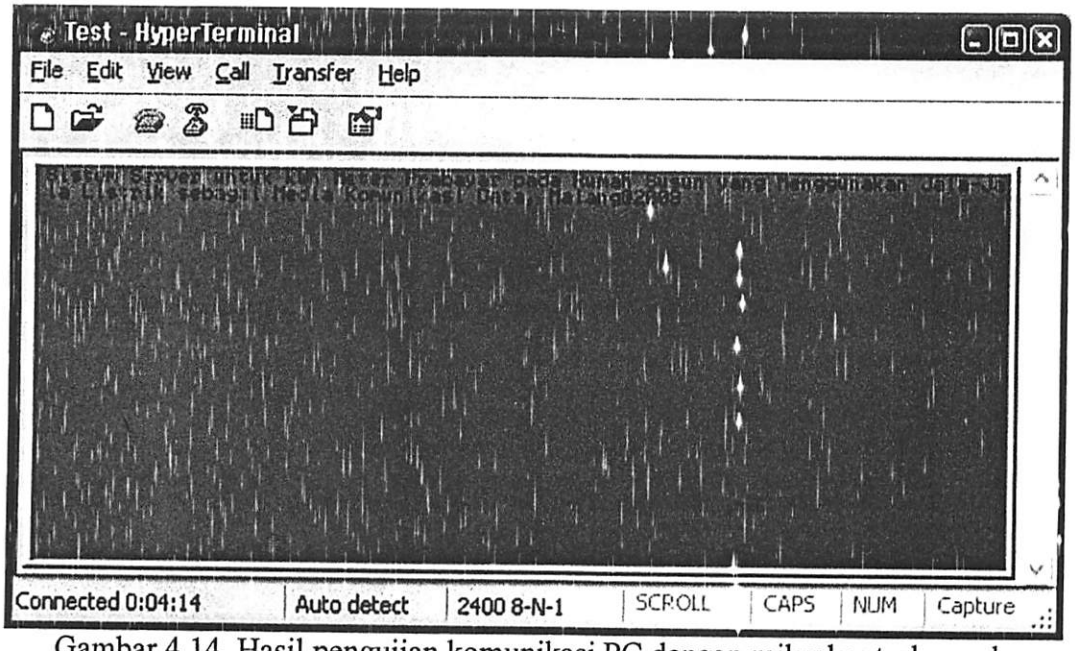
2. Membuat program mikrokontroler dengan algoritma seperti pada Gambar 4.13.
3. Mengaktifkan program Hyperterminal dengan *baudrate* 2400bps.
4. Mengetik karakter pada keyboard berupa tulisan "Sistem *Server* Untuk kWh Meter Prabayar pada Rumah Susun yang Menggunakan Jala-Jala Listrik sebagai Media Komunikasi Data, Malang@2008".
5. Mengamati hasilnya melalui tampilan Hyperterminal.



Gambar 4.13 Diagram alir program mikrokontroler untuk pengujian komunikasi antara mikrokontroler dengan PC

4.4.4. Data Hasil Pengujian

Hasil pengujian dari komunikasi antara PC dengan mikrokontroler ditunjukkan pada Gambar 4.14.



Gambar 4.14. Hasil pengujian komunikasi PC dengan mikrokontroler pada tampilan program Hyperterminal

4.4.5. Analisa Data Hasil Pengujian

Dalam pengujian komunikasi antara PC dengan mikrokontroler, karakter yang diketik melalui keyboard dikirim ke mikrokontroler secara serial melalui rangkaian konverter level tegangan RS232-TTL. Karakter yang diterima oleh mikrokontroler kemudian dikirim kembali ke PC dan hasilnya ditampilkan dengan program Hyperterminal. Pada gambar 4.14 ditunjukan bahwa karakter yang diketik melalui keyboard sama dengan karakter yang ditampilkan oleh program Hyperterminal, sehingga dapat disimpulkan bahwa komunikasi antara PC dan mikrokontroler dapat terjadi dengan baik.

4.5. Pengujian Rangkaian Selektor

4.5.1. Tujuan

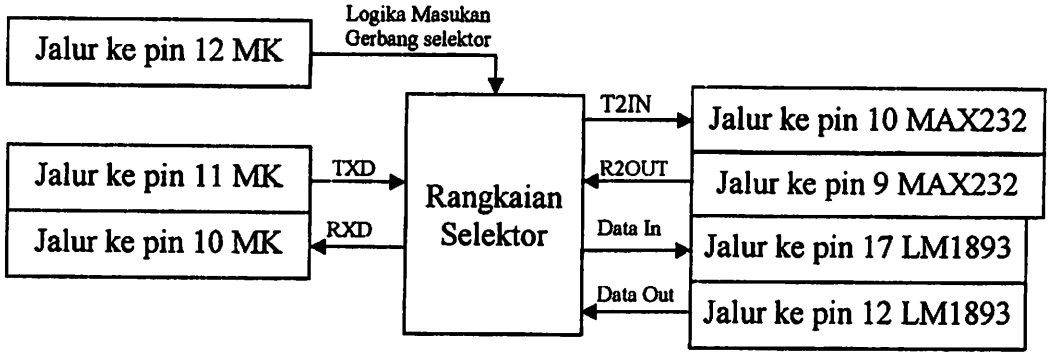
Tujuan dari pengujian rangkaian selektor adalah untuk mendapatkan rangkaian yang mampu digunakan untuk memilih hubungan komunikasi antara mikrokontroler dengan PC atau mikrokontroler dengan *Modem PLC*.

4.5.2. Peralatan yang Digunakan

1. Rangkaian Selektor berbasis IC 74LS126
2. Voltmeter
3. Catu daya 5V

4.5.3. Langkah-langkah Pengujian

1. Menyusun rangkaian seperti gambar diagram blok berikut:



Gambar 4.15 Diagram blok pengujian rangkaian selektor

2. Memberikan masukan logika 1 atau 0 pada pin pengontrol gerbang pada rangkaian selektor (pin 1, 13 IC 74LS126 dan pin 11, 13 IC 74LS04) yang terhubung ke pin 12 (PD2) mikrokontroler.

3. Mengukur besarnya tegangan pada jalur pin T2IN MAX232 dan Data In LM1893 saat ada masukan tegangan 5V pada jalur ke pin TXD mikrokontroler.
4. Mengukur besarnya tegangan pada jalur ke pin RXD mikrokontroler saat ada masukan tegangan 5V pada jalur pin R2OUT MAX232 dan jalur ke pin Data Out LM1893.

4.5.4. Data hasil pengujian

Tabel 4.7 Data hasil pengujian rangkaian selektor

Asal masukan tegangan 5V	Titik pengukuran	Tegangan keluaran (V) untuk tiap logika gerbang selektor	
		0	1
Jalur ke pin TXD mikrokontroler	T2IN MAX232	3,75	0,00
	Data In LM1893	0,01	3,74
R2OUT MAX232	Jalur ke pin RXD mikrokontroler	3,74	0,12
Data Out LM1893		0,12	3,70

4.5.5. Analisa data hasil pengujian

Dari hasil pengujian rangkaian selektor diketahui bahwa tegangan keluaran dari pin T2IN MAX232 3,75V dan Data In LM1893 0V saat logika gerbang selektor 0. Sebaliknya saat gerbang selektor berlogika 1, tegangan pada pin T2IN MAX232 0,01V dan Data In LM1893 3,74V. Sehingga pemilihan hubungan dengan bagian pengiriman data (TXD) mikrokontroler sudah dapat bekerja dengan baik. Untuk bagian penerimaan data (RXD) mikrokontroler terlihat bahwa saat logika 0 pada gerbang selektor, tegangan yang terukur sebesar 3,74V dan saat logika 1 pada gerbang selektor, tegangan yang terukur sebesar 0,12V untuk masukan tegangan yang berasal dari pin R2OUT MAX232. Selanjutnya untuk

tegangan masukan 5V berasal dari pin Data Out LM1893. saat logika 0 pada gerbang selektor, tegangan yang terukur sebesar 0,12V dan saat logika 1 pada gerbang selektor, tegangan yang terukur sebesar 3,70V. Artinya bahwa rangkaian selektor sudah dapat memilih data yang diterima oleh pin RXD mikrokontroler dengan merubah logika gerbang selektor. Sehingga logika gerbang selektor 0 untuk hubungan PC dengan mikrokontroler dan 1 untuk hubungan mikrokontroler dengan *Modem PLC.*

4.6. Pengujian Keseluruhan Sistem

4.6.1. Pengujian pengiriman pulsa prabayar (nominal kWh)

4.6.1.1. Tujuan

Untuk mengetahui keberhasilan sistem *server* dalam proses pengisian ulang pulsa prabayar ke kWh meter prabayar pada unit rumah susun.

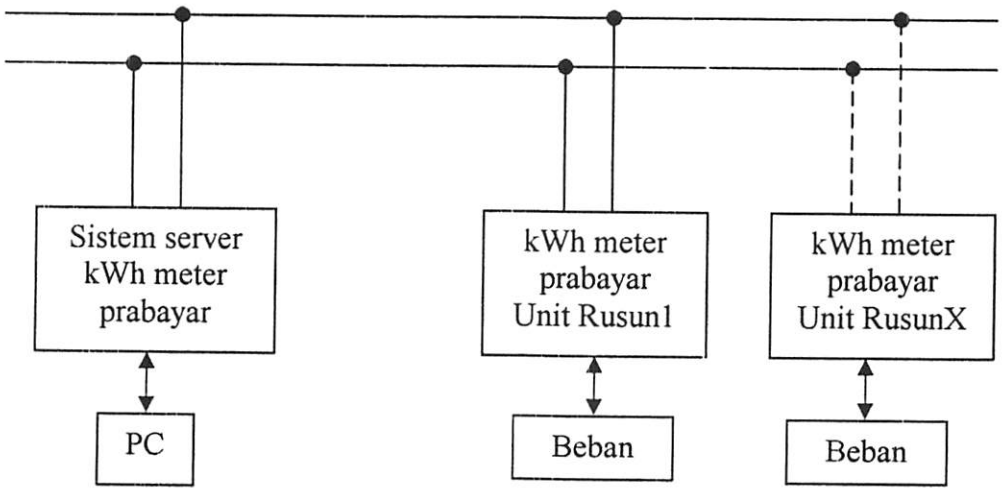
4.6.1.2. Peralatan yang digunakan

1. Rangkaian keseluruhan dari *server*
2. Sistem pada kWh meter prabayar
3. Program Apliksi untuk PC *Server.*

4.6.1.3. Langkah-langkah Pengujian

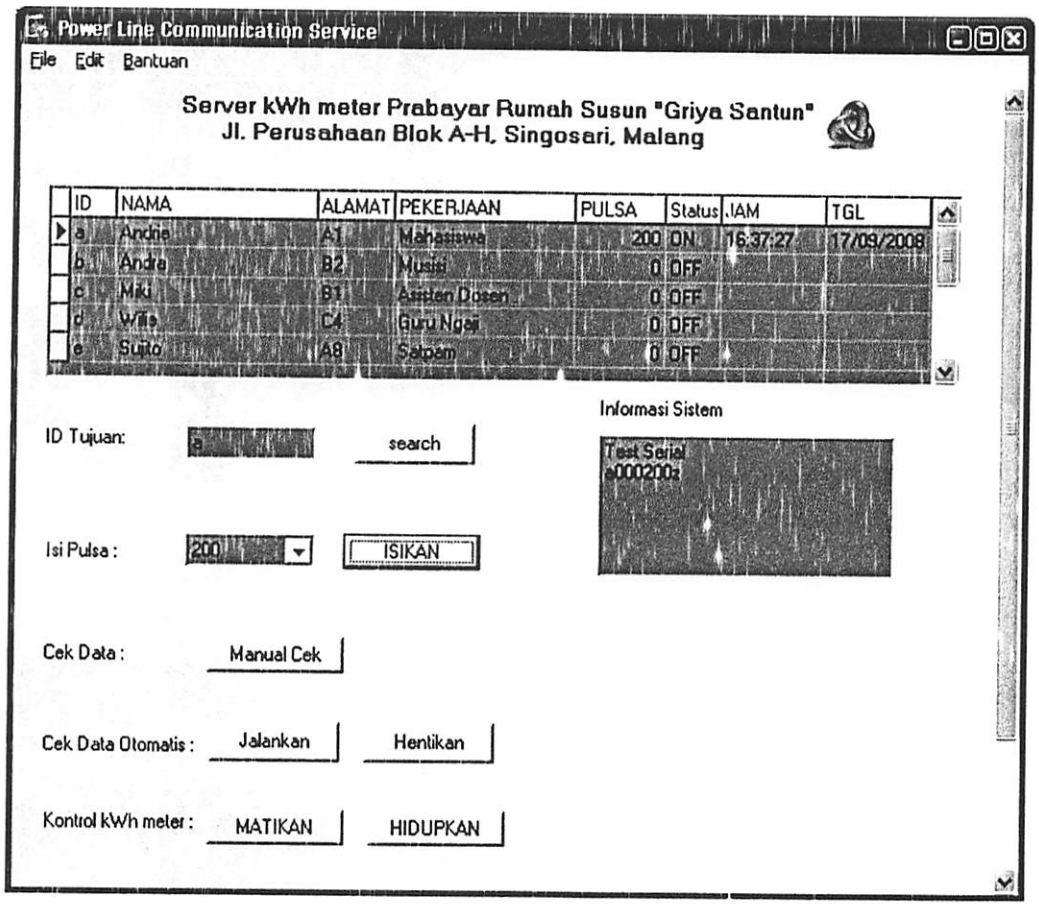
Langkah-langkah untuk melakukan pengujian dari pengisian ulang pulsa prabayar ke kWh meter prabayar adalah sebagai berikut:

1. Menyusun sistem yang digunakan seperti diagram blok dibawah ini:



Gambar 4.16 Diagram blok pengujian keseluruhan sistem

2. Mengaktifkan program pada PC *server* sehingga muncul tampilan berikut:



Gambar 4.17 Program aplikasi pada PC *server*

3. Memasukkan ID pelanggan yang dituju pada kotak edit "ID Tujuan" dan menekan tombol "*Search*".
4. Memilih nominal paket pulsa yang akan dikirimkan pada kotak "Isi Pulsa".
5. Menekan tombol "Isikan" untuk mengirimkan ke kWh meter pelanggan yang dituju.
6. Mengamati data yang diterima dari kWh Meter pelanggan dan memasukkan ke Tabel 4.8 dan Tabel 4.9.

4.6.1.4. Data Hasil Pengujian

Dari pengujian pengiriman paket pulsa dari *Server* ke kWh Meter pelanggan prabayar yang dituju untuk saldo awal masing-masing pelanggan 0kWh diperoleh data-data sebagai berikut :

Tabel 4.8 Data hasil pengujian pengiriman pulsa prabayar

ID Tujuan	Pulsa yang dikirim (kWh)	Pulsa yang diterima (kWh)
a	200	200
b	200	0
c	200	0

Tabel 4.9 Data hasil pengujian akumulasi pengisian ulang pulsa prabayar

No.	ID Tujuan	Saldo pulsa awal (kWh)	Jumlah isi ulang pulsa (kWh)	Saldo pulsa update (kWh)
1	a	200	100	300
2		300	50	350
3		350	25	375

4.6.1.5. Analisa Data Hasil Pengujian

Untuk pengujian pengiriman pulsa ke pelanggan degan saldo awal 0 terlihat bahwa penengiriman pulsa berhasil untuk pelanggan dengan ID "a". Sedangkan pelanggan dengan ID "b" dan "c" tidak dapat dikirimi pulsa. Hal ini

dikarenakan kWh Meter "a" terpasang dan dapat berkomunikasi dengan *server*, sedangkan untuk kWh meter pelanggan "b" dan "c" belum terpasang atau tidak dapat melakukan komunikasi dengan *server*.

Untuk pengujian akumulasi pengisian ulang pulsa prabayar, pelanggan dengan ID "a" berhasil pada pengisian ulang pulsa prabayar mulai dari saldo awal 200kWh dengan pengisian ulang sebesar 100kWh, saldonya bertambah menjadi 300kWh. Kemudian untuk pengisian ulang pulsa prabayar mulai dari saldo awal 300kWh dengan pengisian ulang sebesar 50kWh, saldonya bertambah menjadi 350kWh. Dan untuk pengisian ulang pulsa prabayar mulai dari saldo awal 350kWh dengan pengisian ulang sebesar 250kWh, saldonya bertambah menjadi 375kWh.

Sehingga dari pengujian pengisian pulsa dapat disimpulkan bahwa *server* telah dapat berkomunikasi dan melakukan pengisian ulang pulsa prabayar pada kWh meter pelanggan prabayar untuk rumah susun yang telah aktif.

4.6.2. Pengujian Pemeriksaan Data Pelanggan

4.6.2.1. Tujuan

Untuk mengetahui keberhasilan sistem *server* dalam pemeriksaan atau pengecekan jumlah pulsa dan status kWh meter prabayar pada setiap unit rumah susun.

4.6.2.2. Peralatan yang digunakan

1. Rangkaian keseluruhan dari *server*
2. Sistem pada kWh meter prabayar
3. Program Apliksi untuk PC *Server*.

4.6.2.3. Langkah-langkah pengujian

1. Menyusun sistem yang digunakan seperti Gambar 4.16.
2. Mengaktifkan program pada PC *server* sehingga muncul tampilan seperti Gambar 4.17.
3. Memasukkan ID pelanggan yang dituju pada kotak Edit "ID Tujuan" atau mengklik langsung ID tujuan pada tabel program aplikasi.
4. Menekan tombol "Jalankan" pada bagian pengecekan otomatis untuk melakukan pengecekan data pelanggan tiap 1 menit sekali.
5. Mengamati data yang diterima dari kWh Meter pelanggan dan memasukkan ke tabel 4.10.

4.6.2.4. Data hasil pengujian

Tabel 4.10 Data hasil pengujian pemeriksaan otomatis data pelanggan

ID Pelanggan	Status meteran	Status di *server*	Saldo pulsa (kWh) tercatat pada database *server*			Saldo pulsa (kWh) terlihat pada LCD kWh Meter		
			1 menit	2 menit	3 menit	1 menit	2 menit	3 menit
a	*ON*	*ON*	375	375	375	375	375	375
b	N/A	*OFF*	0	0	0	0	0	0
c	N/A	*OFF*	0	0	0	0	0	0

4.6.2.5.Analisa data hasil pengujian

Pada hasil pengujian pemeriksaan data pelanggan terlihat bahwa mulai dari menit pertama sampai menit ketiga data pulsa dapat diterima oleh *server*. Untuk pelanggan dengan ID "a" jumlah pulsa diterima tetap karena pada kWh meter pelanggan tidak ada beban yang terpasang dan statusnya *"On"* artinya listrik ke pelanggan tidak padam. Untuk pelanggan dengan ID "b" dan "c" saldo pulsanya tetap bernilai 0 karena *server* tidak dapat menerima data dari pelanggan tersebut

dan satusnya "*Off*" artinya listrik ke pelanggan padam atau pelanggan dengan ID tersebut tidak ada.

4.6.3. Pengujian Pengendalian *(On/Off)* Aliran Listrik Pelanggan

4.6.3.1. Tujuan

Untuk mengetahui keberhasilan sistem *server* dalam pengendalian atau pengontrolan aliran listrik ke pelanggan.

4.6.3.2. Peralatan yang digunakan

1. Rangkaian keseluruhan dari *server*
2. Sistem pada kWh meter prabayar
3. Program Apliksi untuk PC *Server*

4.6.3.3. Langkah-langkah pengujian

1. Menyusun sistem yang digunakan seperti Gambar 4.16.
2. Mengaktifkan program pada PC *server* sehingga muncul tampilan seperti Gambar 4.17.
3. Memasukkan ID pelanggan yang dituju pada kotak Edit "ID Tujuan" atau mengklik langsung ID tujuan pada tabel program aplikasi.
4. Menekan tombol "Hentikan" pada bagian pengecekan otomatis untuk menghentikan pengecekan data pelanggan jika masih berjalan.
5. Menekan tombol "Matikan" untuk memadamkan aliran listrik pada pelanggan yang dituju.

6. Mengamati data atau laporan yang diterima dari kWh Meter pelanggan dan memasukkan ke tabel 4.11.

4.6.3.4. Data hasil pengujian

Tabel 4.11 Data hasil pengujian pengontrolan aliran listrik ke pelanggan

ID Tujuan	Perintah pengontrolan	Listrik di Unit pelanggan	Status yang diterima *server*
a	Matikan	*OFF*	*OFF*
	Hidupkan	*ON*	*ON*
b	Matikan	N/A	*OFF*
	Hidupkan	N/A	*OFF*
c	Matikan	N/A	*OFF*
	Hidupkan	N/A	*OFF*

4.6.3.5. Analisa data hasil pengujian

Dari data hasil pengujian pengontrolan aliran listrik ke pelanggan, *server* telah dapat menontrol aliran listrik ke pelanggan. Untuk pelanggan dengan ID "a" dengan memberikan perintah "Matikan" listrik ke pelanggan tersebut padam dan *server* mendapatkan laporan status "*Off*". Dan dengan memberikan perintah "Hidupkan" listrik hidup kembali dan *server* mendapatkan laporan status "*On*". Untuk pelanggan dengan ID "b" dan "c" status yang diterima selalu "*Off*" karena kWh meter untuk pelanggan tersebut belum terpasang atau tidak ada.

BAB V

KESIMPULAN

5.1 Kesimpulan

Dari hasil pengujian tiap blok rangkaian dan pengujian sistem keseluruhan yang telah dilakukan pada bab IV, dapat ditarik kesimpulan sebagai berikut :

1. Untuk membuat *server* agar dapat berkomunikasi dengan sistem kWh meter prabayar pada rumah susun melalui jala-jala listrik dibutuhkan rangkaian konverter level tegangan, rangkaian kontroler, rangkaian selektor dan rangkaian *Modem PLC.*
2. Pembuatan perangkat lunak untuk mikrokontroler digunakan untuk mengatur kerja sistem dan dirancang agar dapat memfilter data yang masuk ke mikrokontroler dari *Modem PLC.* Dan perangkat lunak untuk PC digunakan untuk melayani pelanggan listrik prabayar pada rumah susun dibuat dengan bahasa pemrograman Delphi 7 yang dirancang agar dapat berkomunikasi dengan mikrokontroler dan mampu memfilter data mentah yang berasal dari mikrokontroler sehingga didapatkan data yang valid.
3. Dari pengujian rangkaian *Modem PLC* LM1893 dapat diketahui bahwa rangkaian tersebut dapat bekerja sesuai dengan spesifikasi yang ditunjukkan oleh *datasheet* baik pada saat mode *Transmit* maupun pada saat mode *Receive.*
4. Dari pengujian komunikasi antar mikrokontroler menggunakan rangkaian *Modem PLC* LM1893 dapat diketahui bahwa nilai *baudrate* terbesar yang bisa digunakan adalah 960 bps. Oleh karena itu dipilih *baudrate* sebesar 360 bps

agar komunikasi dapat terjadi secara optimal sesuai dengan frekuensi data yang dianjurkan dalam *datasheet*.

5. Dari pengujian komunikasi antara mikrokontroler dengan PC dapat diketahui bahwa komunikasi dapat terjadi dengan baik
6. Dari pengujian rangkaian selektor dapat diketahui bahwa rangkaian tersebut dapat digunakan untuk memilih hubungan antara mikrokontroler dengan PC atau mikrokontroler dengan *Modem PLC*.
7. *Server* untuk sistem kWh meter prabayar pada rumah susun yang menggunakan jala-jala listrik sebagai media komunikasi data dapat bekerja dengan baik karena dapat digunakan untuk melayani pengisian ulang pulsa prabayar ke kWh meter pelanggan, memeriksa data pelanggan dan mengontrol aliran listrik ke pelanggan.

5.2 Saran

1. Proses perancangan, pembuatan dan pengujian alat ini dilakukan sesuai dengan spesifikasi yang telah ditentukan sehingga masih ada faktor-faktor lain yang diabaikan. Faktor-faktor tersebut mungkin dapat mempengaruhi kinerja sistem jika alat diaplikasikan pada kondisi sesungguhnya. Faktor-faktor tersebut seperti panjang dan jenis kabel yang digunakan, adanya sambungan-sambungan, serta adanya harmonisa dan *noise*. Oleh sebab itu perlu dilakukan penelitian tentang pengaruh faktor-faktor tersebut terhadap kinerja sistem.
2. Besarnya daya pemancar pada alat ini sesuai dengan spesifikasi IC LM1893 yang memiliki jangkauan yang terbatas sehingga perlu dilakukan penelitian tentang daya pemancar agar dapat digunakan untuk proses pengiriman data

dalam jangkauan yang lebih jauh. Selain itu perlu juga dilakukan penambahan sistem keamanan data karena dalam perancangan dan pembuatan alat ini belum memperhitungkan masalah keamanan data yang ditransmisikan.

3. Untuk pengembangan lebih lanjut bisa ditambahkan *feature* berupa pembelian nominal kWh melalui SMS, ATM atau Internet sehingga pelanggan memiliki lebih banyak pilihan dan lebih mudah untuk memperoleh nominal kWh yang akan digunakan.

DAFTAR PUSTAKA

[1]. Osmond Consulting Indonesia, "*Kwh Meter Prabayar dengan Smart Card*" http://indonetwork.co.id/osmondconsulting/prod, Jakarta, 2008.

[2]. Agung Nugroho, "*Peralatan Kopling Power Line Carrier*", Semarang, 2005.

[3]. *Datasheet LM1893/LM2893 Carrier-Current Transceiver*, National Semiconductor, U.S.A, www.DigChip.com, 1995.

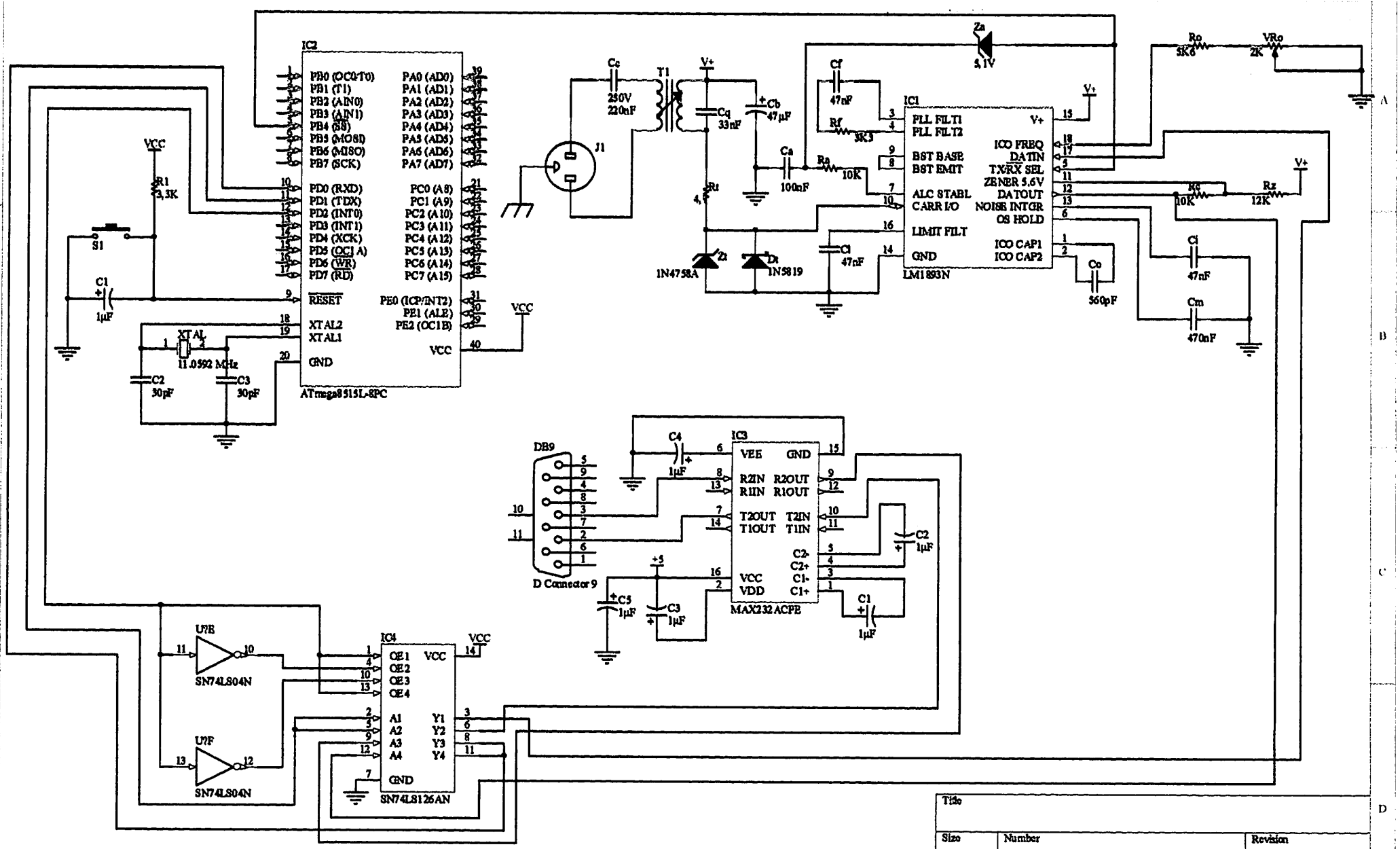
[4]. Yudha, "*Antar-Muka Serial dan Paralel Dalam Sistem Komputer*", 2006.

[5]. "*Datasheet ATmega8515*", Atmel Corporation. www.atmel.com, 2006.

[6]. Paul Topfer, *Technology Review of Powerline Communications (PLC) Technologies and Their Use in Australia*, Final Report, Australia, 2004.

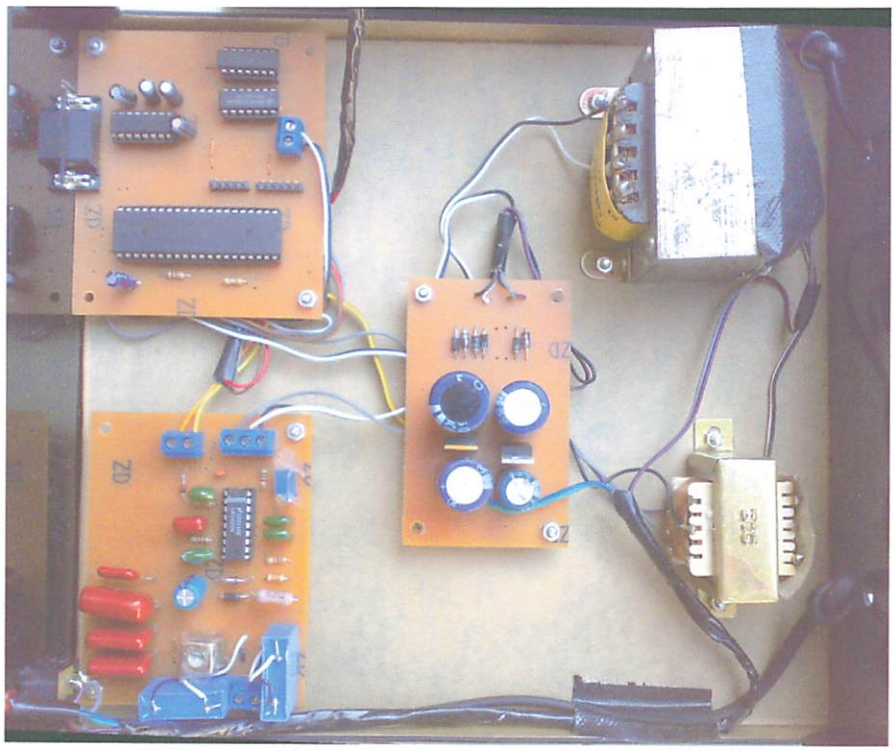
[7]. Widodo, Romy Budhi dan Joseph Dedy Irawan, "*Interfacing Paralel & Serial*", Yogyakarta: Graha Ilmu, 2007.


LAMP
i
RAN



Title		
Size: A4	Number	Revision
Date: 21/10/2008	Sheet of	
File: F:\Skripsi-TA\..\Rangk. Lengkap.SchDoc	Drawn By:	

FOTO ALAT



Rangkaian Sistem Server

Perangkat Server untuk melayani sistem kWh Meter prabayar pada rumah susun

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

NI (PERSERO) MALANG
ANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting). Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 24 Juni 2008

Nomor : ITN-089/1.TA/7/08
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr. **Ir. Widodo Pudji M, MT**
Dosen Institut Teknologi Nasional

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama	: Slamet Setiyabudi
Nim	: 0217084
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik Elektronika

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/I selama masa waktu (enam) bulan, terhitung mulai tanggal :

14 Juni 2008 s/d 14 Desember 2008

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian agar maklum atas perhatian serta bantuannya kami sampaikan terima kasih.

Ketua Jurusan
Teknik Elektro S-1

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

Tindasan Kepada Yth :
1. Mahasiswa yang bersangkutan
2. Arsip
3. *) coret yang tidak perlu

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
BNI (PERSERO) MALANG
BANK NIAGA MALANG
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo. Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 24 Juni 2008

Nomor : ITN-090/1.TA/7/08
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr. **Ir. Yusuf Ismail Nakhoda, MT**
Dosen Institut Teknologi Nasional

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat,
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama	: Slamet Setiyabudi
Nim	: 0217084
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik Elektronika

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/I selama masa waktu (enam) bulan, terhitung mulai tanggal :

14 Juni 2008 s/d 14 Desember 2008

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian agar maklum atas perhatian serta bantuannya kami sampaikan terima kasih.

Ketua Jurusan
Teknik Elektro S-1

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

Tindasan Kepada Yth :
1. Mahasiswa yang bersangkutan
2. Arsip
3. *) coret yang tidak perlu

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR BIMBINGAN SKRIPSI

Nama : Slamet Setiyabudi
NIM : 02.17.084
Masa Bimbingan : 14 Juni 2008 s/d 14 Desember 2008
Judul Skripsi : Perancangan dan Pembuatan Sistem Server untuk kWh Meter Prabayar pada Rumah Susun yang Menggunakan Jala-jala Listrik sebagai Media Komunikasi Data

No.	Tanggal	Uraian	Paraf Pembimbing
1.	15 - 8 - 08	Konsultasi bab I. II & III	
2.	19 - 8 - 08	Perbaikan bab I, II & III	
3.	2 - 9 - 08	Konsultasi bab IV & V	
4.	5 - 9 - 08	Perbaikan bab IV & V	
5.	15 - 9 - 08	Konsultasi Makalah Seminar Skripsi	
6.	22 - 9 - 08	Konsultasi Laporan Skripsi keseluruhan	
7.	23 - 9 - 08	Acc Kompre	
8.			
9.			
10.			

Malang, 24 September 2008
Dosen Pembimbing,

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

Form.S-4b

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR BIMBINGAN SKRIPSI

Nama : Slamet Setiyabudi
NIM : 02.17.084
Masa Bimbingan : 14 Juni 2008 s/d 14 Desember 2008
Judul Skripsi : Perancangan dan Pembuatan Sistem Server untuk kWh Meter Prabayar pada Rumah Susun yang Menggunakan Jala-jala Listrik sebagai Media Komunikasi Data

No.	Tanggal	Uraian	Paraf Pembimbing
1.	20/8/08	Konsultasi Bab I, II	
2.	21/8/08	Revisi Bab I – Tujuan, Rumusan & Batasan Masalah	
3.	25/8/08	Konsultasi Bab III	
4.	28/8/08	Revisi Bab III – Skematik rangkaian Modem PLC, Mikrokontroler & Selektor	
5.	5/9/08	Konsultasi Bab IV & V	
6.	9/9/08	Revisi Bab IV & V – Tabel dan Gambar hasil pengujian, Kesimpulan	
7.	15/9/08	Konsultasi Makalah Seminar Skripsi	
8.	22/9/08	Konsultasi Laporan Skripsi	
9.	23/9/08	ACC keseluruhan laporan & persiapan Ujian Skripsi	
10.			

Malang, 24 September 2008
Dosen Pembimbing

(Ir. Yusuf Ismail Nakhoda, MT)
NIP. Y. 1018800189

Form.S-4b

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

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

NAMA : Slamet setyabudi
NIM : 0217084
Perbaikan meliputi :

Rumusan masalah.
Rangkaian bagian-
per alat.

Malang, 25 Sept 2008.

(M. Ibrahim A ST, MT)

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR PERBAIKAN SKRIPSI

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

Nama	: Slamet Setiyabudi
NIM	: 02.17.084
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik Elektronika
Masa Bimbingan	: 14 Juni 2008 – 14 Desember 2008
Judul Skripsi	: Perancangan dan Pembuatan Sistem Server untuk kWh Meter Prabayar pada Rumah Susun yang Menggunakan Jala-Jala Listrik sebagai Media Komunikasi Data

Tanggal	Uraian	Paraf
17 Oktober 2008	✓ Rumusan Masalah	
	✓ Rangkaian Lengkap	
	✓ Foto Alat	

Dosen Penguji

M. Ibrahim Ashari ST, MT
NIP. P. 1030100358

Mengetahui

Dosen Pembimbing I

Ir. Widodo Pudji M, MT
NIP. 1028700171

Dosen Pembimbing II

Ir. Yusuf Ismail Nakhoda, MT
NIP. Y. 1018800189

//PROGRAM DELPHI

```
unit Unit1;

interface

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

type
  TForm1 = class(TForm)
    ComPort1: TComPort;
    tbid: TTable;
    DSid: TDataSource;
    Notebook1: TNotebook;
    DBEdit1: TDBEdit;
    Label1: TLabel;
    DBEdit2: TDBEdit;
    Label2: TLabel;
    Label3: TLabel;
    DBEdit3: TDBEdit;
    Label4: TLabel;
    DBEdit4: TDBEdit;
    Label5: TLabel;
    DBEdit5: TDBEdit;
    Button1: TButton;
    DBGrid1: TDBGrid;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Timer1: TTimer;
    DBGrid2: TDBGrid;
    Button5: TButton;
    Button6: TButton;
    Label6: TLabel;
    MainMenu1: TMainMenu;
    Pelanggan1: TMenuItem;
    Baru1: TMenuItem;
    Lama1: TMenuItem;
    ComboBox1: TComboBox;
    Button7: TButton;
    Memo1: TMemo;
    Button8: TButton;
    Memo2: TMemo;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure ComPort1RxChar(Sender: TObject; Count: Integer);
    procedure Button5Click(Sender: TObject);
    procedure Baru1Click(Sender: TObject);
    procedure Lama1Click(Sender: TObject);
    procedure Button7Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure Button8Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  data_tot,id:string;
```

```
implementation

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
tbid.Append;
Button1.Enabled:=FALSE;
Timer1.Enabled:=FALSE;
end;

procedure TForm1.Button2Click(Sender: TObject);
var uang:string;
begin
tbid.Edit;
tbid.FieldByName('status').AsString:='ON';
tbid.Post;
Appfication.ProcessMessages;
Button1.Enabled:=TRUE;
uang:=DBEdit5.Text;

while length(uang)<6 do uang:='0'+uang;

timer1.Enabled:=false;
timer1.Enabled:=true;
ComPort1.WriteStr(tbid.FIELDBYNAME('ID').AsString+uang+'z');
Timer1.Enabled:=TRUE;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
tbid.Cancel;
Button1.Enabled:=TRUE;
Timer1.Enabled:=TRUE;
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
IF tbid.RecordCount<>0 then tbid.Delete;
Button1.Enabled:=TRUE;
Timer1.Enabled:=TRUE;
end;

procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var data:string;
begin
ComPort1.ReadStr(data,count);
data_tot:=data_tot+data;
Memo1.Text:=Memo1.Text+data;
if (pos('a',data_tot)<>0) and (pos('z',data_tot)<>0) then
 begin
 data_tot:=copy(data_tot,pos('z',data_tot)-7,8);
 Memo2.Lines.Add(data_tot);
 // Memo1.Lines.Add(data_tot);
 IF tbid.FindKey([copy(data_tot,1,1)]) THEN
   BEGIN
   tbid.Edit;
   tbid.FieldByName('pulsa').AsString:=copy(data_tot,2,6);
   tbid.FieldByName('jam').AsDateTime:=time;
   tbid.FieldByName('tgl').AsDateTime:=date;
   IF StrToInt(copy(data_tot,2,6))=0 then tbid.FieldByName('status').AsString:='OFF';
   tbid.Post;
   END;
 data_tot:='';
 end;
end;

procedure TForm1.Button5Click(Sender: TObject);
```

```
begin
tbid.Edit;
tbid.FieldByName('status').AsString:='OFF';
tbid.Post;
timer1.Enabled:=false;
timer1.Enabled:=true;
Sleep(4000);
ComPort1.WriteStr(tbid.FIELDBYNAME('ID').AsString+'m12345z');
Timer1.Enabled:=false;
Timer1.Enabled:=true;
end;

procedure TForm1.Baru1Click(Sender: TObject);
begin
Notebook1.ActivePage:='baru';
end;

procedure TForm1.Lama1Click(Sender: TObject);
begin
Notebook1.ActivePage:='lama';
end;

procedure TForm1.Button7Click(Sender: TObject);
VAR UANG:STRING;
begin
UANG:=IntToStr(StrToInt(ComboBox1.Text)+tbid.FIELDBYNAME('PULSA').AsInteger);

while length(uang)<6 do uang:='0'+uang;

//ShowMessage(tbid.FIELDBYNAME('ID').AsString+uang+'z');
Timer1.Enabled:=false;
Timer1.Enabled:=true;
Sleep(4000);
ComPort1.WriteStr(tbid.FIELDBYNAME('ID').AsString+uang+'z');
Timer1.Enabled:=false;
Timer1.Enabled:=true;
end;

procedure TForm1.Button6Click(Sender: TObject);
begin
tbid.Edit;
tbid.FieldByName('status').AsString:='ON';
tbid.Post;
timer1.Enabled:=false;
timer1.Enabled:=true;
Sleep(4000);
ComPort1.WriteStr(tbid.FIELDBYNAME('ID').AsString+'n12345z');
timer1.Enabled:=false;
timer1.Enabled:=true;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
{if tbid.RecordCount<>0 then
begin
if id='a' then id:='b'
else id:='a';
ComPort1.WriteStr(id+'p12345z');
end; }
end;

procedure TForm1.Button8Click(Sender: TObject);
begin
ComPort1.WriteStr(tbid.FIELDBYNAME('ID').AsString+'p12345z');
end;

end.
```

/*PROGRAM MIKROKONTROLER*/

```
/* File include */
#include <mega8515.h>
#include <delay.h>
#include "lcdku.c"
#ifndef __lcdku_C
#define __lcdku_C

#define  enable      PORTB.1
#define  rs          PORTB.0
#define   clock      PORTB.2
#define  tx_LCD      PORTB.3


void dataout(unsigned char data_LCD, char kode)
{
 unsigned char i;
 clock=0;
 for(i=0;i<8;i++)
 {
  tx_LCD=(data_LCD & 0x1)==0x1 ? 1 : 0;
  delay_us(4);
  clock=1;
  delay_us(4);
  clock=0;
  data_LCD=data_LCD>>1;
 }
 rs=kode;
 delay_us(40);
 enable=1;
 delay_us(40);
 enable=0;
}

void pos(unsigned char i,unsigned char n)
{
 if(i==1)
 {
  dataout (0x80+n-1,0);
  delay_us(40);
 }
 else if (i==2)
 {
  dataout (0xc0+n-1,0);
  delay_us(40);
 }
 else;
}


void busek()
{
      dataout(0x01,0);
      delay_ms(2);
}


void initlcd()
{
      delay_ms(1000);
      dataout(0x30,0);
      delay_ms(500);
      dataout(0x30,0);
      delay_us(10000);
      dataout(0x30,0);
      delay_us(4000);
      dataout(0x38,0);
```

```
        delay_us(4000);
        dataout(0x08,0);
        delay_us(4000);
        dataout(0x01,0);
        delay_us(4000);
        dataout(0x0c,0);
        delay_us(4000);
        dataout(0x06,0);
        delay_us(4000);
}

void cetak(int i,int n,flash char *text)
{
        pos(i,n);
        while(*text)
        {
                dataout(*text++,1);
                delay_us(40);
        }
}

#endif
#include "i2c.c"


#asm
  .equ __i2c_port=0x15 ;PORTB
  .equ __sda_bit=1
  .equ __scl_bit=0
#endasm

#include <i2c.h>
//#include <stdio.h>


#define ADDA_ADDR      0x90
#define EEPROM_ADDR    0xA0
#define RTC_ADDR       0xD0


unsigned char read_adc(unsigned char channel)
{
    unsigned char datax;
    i2c_start();
    i2c_write(ADDA_ADDR);
    //i2c_write((channel+0x10));
            i2c_write((channel+0x0));
    i2c_stop();
    i2c_start();
    i2c_write(ADDA_ADDR | 1);
    datax = i2c_read(0);
    i2c_stop();
    return datax;
}

void write_dac(unsigned char tmp1, unsigned char tmp2)
{
    i2c_start();
    i2c_write(ADDA_ADDR);
    i2c_write(tmp1+0x10);
    i2c_write(tmp2);
    i2c_stop();
}

unsigned char bcd2dec(unsigned char input)
{
    unsigned char tmp_data, tmp1;
```

```
    tmp_data = input;
    tmp1 = tmp_data % 16;
    if (tmp_data > 15) tmp_data = tmp_data / 16;
    else tmp_data = 0;
    tmp_data = (tmp_data * 10)+tmp1;
    return tmp_data;
}

unsigned char dec2bcd(unsigned char input)
{
    unsigned char tmp_data;

    if (input > 9) tmp_data = ((input / 10)*16) + (input % 10);
    else
      tmp_data = input;
    return tmp_data;
}

unsigned char read_rtc(unsigned char alamat)
{
    unsigned char tmp_data;

    i2c_start();
    i2c_write(RTC_ADDR);
    i2c_write(alamat);
    i2c_stop();
    i2c_start();
    i2c_write(RTC_ADDR | 1);
    tmp_data = i2c_read(0);
    i2c_stop();
    return bcd2dec(tmp_data);
}

void write_rtc(unsigned char alamat, unsigned char datax)
{
            i2c_start();
    i2c_write(RTC_ADDR);
    i2c_write(alamat);
    if (alamat < 7)
      i2c_write(dec2bcd(datax));
    else
      i2c_write(datax);
    i2c_stop();
}
unsigned char bacamem(unsigned int alamat)
{   unsigned char aa;
    unsigned char datax;
    aa=alamat>>8;aa=aa<<1;
    i2c_start();
    i2c_write(EEPROM_ADDR |aa);
    i2c_write(alamat);
    i2c_stop();
    i2c_start();
    i2c_write(EEPROM_ADDR | (aa|1));
    datax = i2c_read(0);
    i2c_stop();
    return datax;
}

void tulismem(unsigned int alamat, unsigned char nilai)
{   unsigned char aa;
    aa=alamat>>8;aa=aa<<1;
    i2c_start();
    i2c_write(EEPROM_ADDR|aa);
    i2c_write(alamat);
    i2c_write(nilai);
    i2c_stop();
```

```
    delay_ms(10);
}
/*
unsigned char read_EEPROM(unsigned char alamat)
{
    unsigned char datax;
    i2c_start();
    i2c_write(EEPROM_ADDR);
    i2c_write(alamat);
    i2c_stop();
    i2c_start();
    i2c_write(EEPROM_ADDR | 1);
    datax = i2c_read(0);
    i2c_stop();
    return datax;
}

void write_EEPROM(unsigned char alamat, unsigned char nilai)
{
    i2c_start();
    i2c_write(EEPROM_ADDR);
    i2c_write(alamat);
    i2c_write(nilai);
    i2c_stop();
    delay_ms(10);
}

unsigned char read_EEPROM2(unsigned char chip_addr,unsigned int address)
{
    unsigned char datax;
    unsigned char addresshi;
    unsigned char addresslo;
    unsigned char chip;
            addresshi       = address >> 8;
            addresslo       = address;
            chip            = chip_addr << 1;
    i2c_start();
    i2c_write(EEPROM_ADDR | chip );
    i2c_write(addresshi);
    i2c_write(addresslo);
            i2c_stop();
    i2c_start();
    i2c_write(EEPROM_ADDR | (chip | 1 ));
    datax = i2c_read(0);
    i2c_stop();
    return datax;
}

void write_EEPROM2(unsigned char chip_addr,unsigned int address, unsigned char nilai)
{
    unsigned char addresshi;
    unsigned char addresslo;
    unsigned char chip;
            addresshi       = address >> 8;
            addresslo       = address;
            chip            = chip_addr << 1;
            i2c_start();
    i2c_write(EEPROM_ADDR | chip_addr );
    i2c_write(addresshi);
    i2c_write(addresslo);
    i2c_write(nilai);
    i2c_stop();
    delay_ms(10);
}
*/


/* Pendefinisian */
```

```
#define control  PORTB.4
#define pc  PORTD.2
#define triac  PORTC.0
#define pulsa  PIND.2


/* Inisialisasi variabel global */
unsigned char a,d[10];
long int te;
unsigned int i,b,im;
bit t,trx,tm;

/*Inisialisasi input output */
void init_port()
{
      DDRC=0b00000011;
      DDRB=0b00011111;
      DDRD=0b00000110;
      DDRA=0b00000000;
      PORTD.2=1;
      //scld=1;sdad=1;
}


void initser()
{
    UBRRL=64;
    UBRRH=3;
    UCSRA=0x80;//aslinya 0
    UCSRB=0x98; //Txd,Rxd Enabled aslinya 18
    UCSRC=0x86; //8 bit data
}

unsigned char terimaser()
{
      //te=0;
    while((UCSRA & 0x80) == 0x00);//{te++;if(te>100000)break;}
    return UDR;
}

void kirimser(char TxData)
{
    while((UCSRA & 0x20) == 0x00);
    //while ( !( UCSRA & (1<<UDRE)) );
    UDR = TxData;
    delay_ms(10);
}
void kirimtext(flash char *text)
{

      while(*text)
      {
            kirimser(*text++);
      }
}
void pancingan()
{
for(i=0;i<10;i++)
{kirimser('%');delay_ms(100);}
}


/* Program Utama */
void main()
{
```

```
        /* Inisialisasi */
        init_port();
        //init_timer1();
        //init_timer2();
        //init_ext_interrupt();
        initlcd();
        //init_ADC();
        initser();
        //i2c_init();

       pc=0;control=0;
        delay_ms(10);
        kirimtext("Test Serial");
        cetak(1,1,"goal");
        delay_ms(1000);busek();
        #asm("cli")
        i=0;
        do
        {
        pc=0;control=0;pos(1,1);
        for(i=0;i<8;i++)
        {d[i]=terimaser();dataout(d[i],1);}
        pc=1;control=1;//delay_ms(1000);
        pancingan();
        for(i=0;i<8;i++)
        kirimser(d[i]);


        delay_ms(500);
        control=0;
        pos(1,1);i=0;tm=0;im=0;
        while(a<'a' || a>'j')
        {
         //while((UCSRA & 0x80) == 0x00);
        a=terimaser();im++;
        if(im>100)tm=1;
        if(im>100)break;
        }
        if(tm==0)
        {
        d[i]=a;dataout(a,1);i++;
        while(a!='z'){a=terimaser();d[i]=a;dataout(a,1);i++;}
        pc=0;control=0;delay_ms(1000);
        for(i=0;i<8;i++)
        kirimser(d[i]);
        }


       }while(1);
}
```

National Semiconductor

April 1995

LM1893/LM2893 Carrier-Current Transceiver†

General Description

Carrier-current systems use the power mains to transfer information between remote locations. This bipolar carrier-current chip performs as a power line interface for half-duplex (bi-directional) communication of serial bit streams of virtually any coding. In transmission, a sinusoidal carrier is FSK modulated and impressed on most any power line via a rugged on-chip driver. In reception, a PLL-based demodulator and impulse noise filter combine to give maximum range. A complete system may consist of the LM1893, a COPS™ controller, and discrete components.

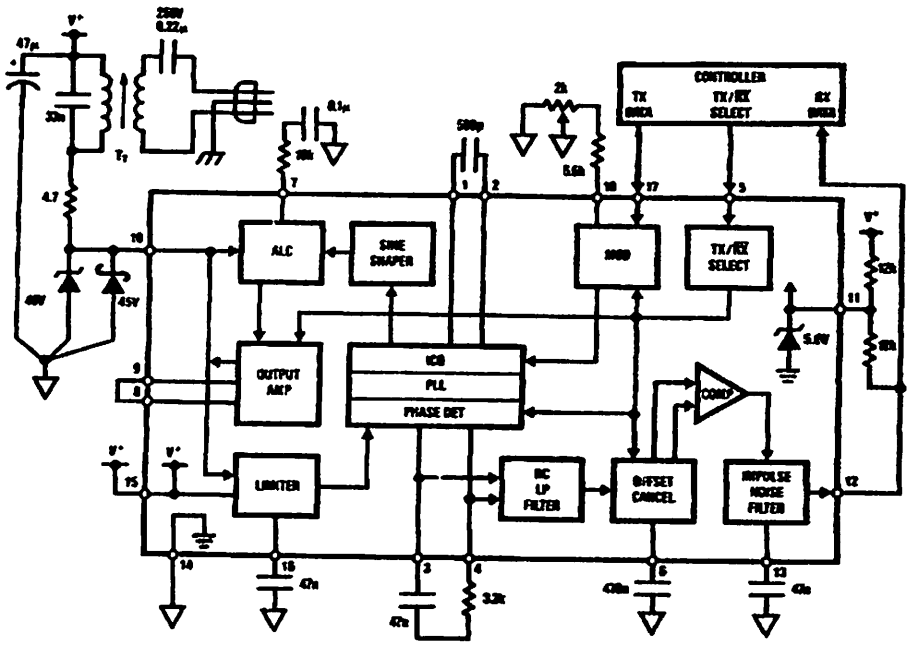
Features

- Noise resistant FSK modulation
- User-selected impulse noise filtering
- Up to 4.8 kBaud data transmission rate
- Strings of 0's or 1's in data allowed
- Sinusoidal line drive for low RFI
- Output power easily boosted 10-fold
- 50 to 300 kHz carrier frequency choice
- TTL and MOS compatible digital levels
- Regulated voltage to power logic
- Drives all conventional power lines

Applications

- Energy management systems
- Home convenience control
- Inter-office communication
- Appliance control
- Fire alarm systems
- Security systems
- Telemetry
- Computer terminal interface

Typical Application



TL/H/6750–1

FIGURE 1. Block diagram of carrier—current chip with a complement of discrete components making a complete F_O = 125 kHz, f_{DATA} = 360 Baud transceiver. Use caution with this circuit—dangerous line voltage is present.

BI-LINE™ and COPS™ are trademarks of National Semiconductor Corp.
†Carrier-Current Transceivers are also called Power Line Carrier (PLC) transceivers.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply voltage		30 V
Voltage on pin 12		55 V
Voltage on pin 10 (Note 1)		41 V
Voltage on pins 5 and 17		40 V
5.6 V DC zener current		100 mA
Junction temperature:	transmit mode	150°C
	receive mode	125°C
Electro-Static Discharge (120 pF, 1500Ω)		1KV

Maximum continuous dissipation, T_A=25°C, plastic DIP N (Note 2):	transmit mode	1.66 W
	receive mode	1.33 W
Operating ambient temp. range		−40 to 85°C
Storage temperature range		−65 to 150°C
Lead temp., soldering, 7 seconds		260°C

Note: Absolute maximum ratings indicate limits beyond which damage to the device may occur. Electrical specifications are not ensured when operating the device above guaranteed limits but below absolute maximum limits, but there will be no device degradation.

General Electrical Characteristics

(Note 3). The test conditions are: $V^+ = 18V$ and $F_O = 125$ kHz, unless otherwise noted.

#	Parameter	Conditions	Typical	Test Limit (Note 4)	Design Limit (Note 5)	Limit Units
1	5.6 V Zener voltage, V_Z	Pin 11, I_Z = 2 mA	5.6	5.2 5.9		V min. V max.
2	5.6 V Zener resistance, R_Z	Pin 11, $R_Z = (V_Z@10\ mA - V_Z@1\ mA)/(10\ mA - 1\ mA)$	5			Ω
3	Carrier I/O peak survivable transient voltage, V_{OT}	Pin 10, discharge 1 μF cap. charged to V_{OT} thru < 1Ω	80	60		V max.
4	Carrier I/O clamp voltage, V_{OC}	Pin 10, I_{OC} = 10 mA, RX mode 2N2222 diode pin 8 to 9	44	41 50		V min. V max.
5	Carrier I/O clamp resistance, R_{10}	Pin 10, I_{OC} = 10 mA	20			Ω
6	TX/$\overline{RX}$ low input voltage, V_{IL}	Pin 5	1.8	0.8		V max.
7	TX/$\overline{RX}$ high input voltage, V_{IH}	Pin 5 (Note 9)	2.2	2.8		V min.
8	TX/$\overline{RX}$ low input current, I_{IL}	Pin 5 at 0.8 V	−2	−20 1		μA min. μA max.
9	TX/$\overline{RX}$ high input current, I_{IH}	Pin 5 at 40 V	10^{-4}	−1 10	0	μA min. μA max.
10	RX−TX switch-over time, T_{RT}	Time to develop 63% of full current drive thru pin 10	10			μs
11	TX−RX switch-over time, T_{TR}	1 bit time, $T_B = 1/(2F_{DATA})$. Time T_{TR} is user controlled with C_M, see Appls. Info.	2			bit
12	ICO initial accuracy of F_O	TX mode, R_O = 6.65 kΩ, C_O = 560 pF $F_O = (F_1 + F_2)/2$	125	113 137		kHz min. kHz max.
13	ICO temperature coefficient of F_O	TX or RX mode, $(F_{OMAX} - F_{OMIN})/(T_{JMAX} - T_{JMIN})$	−100			PPM/°C
14	Temperature drift of F_O	TX or RX mode, $-40 \le T_J \le T_{JMAX}$	±2.0		±5.0	% max.

Transmitter Electrical Characteristics (Note 3). The test conditions are: $V^+ = 18$ V and $F_O = 125$ kHz unless otherwise noted. The transmit center frequency is F_O, FSK low is F_1, and FSK high is F_2.

#	Parameter	Conditions	Typical	Test Limit (Note 4)	Design Limit (Note 5)	Limit Units
15	Supply voltage, V^+, range	Meets test 17 spec. at T_J = 25°C and: $\lvert(F_1[14V] - F_1[18V])/F_1[18V]\rvert < 0.01$ $\lvert(F_1[24V] - F_1[18V])/F_1[18V]\rvert < 0.01$	13 40	14 24	15 23	V min. V max.
16	Total supply current, I_{OT}	Pin 15. Pin 12 high. I_{OT} is I_O through pin 15 and the average current I_{ODC} of the Carrier I/O through pin 10	52	79		mA max.
17	Carrier I/O output current, I_O	100Ω load on pin 10	70	45		mApp min.
18	Carrier I/O lower swing limit, V_{ALC}	Pin 10. Set internally be ALC. 2N2222 diode pin 8 to 9	4.7	4.0 5.7		V min. V max.
19	THD of I_O (Note 6)	Q of 10 tank driving 10Ω line 100Ω load, no tank	0.6 5.5		5.0 9	% max. % max.
20	FSK deviation, $F_2 - F_1$	$(F_2 - F_1)/([F_2 + F_1]/2)$	4.4	3.7 5.2		% min. % max.
21	Data In, low input voltage, V_{IL}	Pin 17	1.7	0.8		V max.
22	Data In, high input voltage, V_{IH}	Pin 17 (Note 9)	2.1	2.8		V min.
23	Data In, low input current, I_{IL}	Pin 17 at 0.8 V	−1	−10 1		μA min. μA max.
24	Data In, high input current, I_{IH}	Pin 17 at 40 V	10^{-4}	−1 10	0	μA min. μA max.

Receiver Electrical Characteristics (Note 3). The test conditions are: $V^+ = 18$ V, $F_O = 125$ kHz, ±2.2% deviation FSK, $F_{DATA} = 2.4$ kHz, $V_{IN} = 100$ mVpp, in the receive mode, unless otherwise noted.

#	Parameter	Conditions	Typical	Test Limit (Note 4)	Design Limit (Note 5)	Limit Units
25	Supply voltage, V^+, range	Functional receiver (Note 7)	**12** **37**	13 30	**13.5** **28**	V min. V max.
26	Supply current, I_{DT}	I_{DT} is pin 15 (V^+) plus pin 10 (Carrier I/O) current. 2.4 kΩ Pin 13 to GND.	11	5 14		mA min. mA max.
27	Carrier I/O input resistance, R_{10}	Pin 10	19.5	14 30		kΩ min. kΩ max.
28	Max. data rate, F_{MD}	Functional receiver (Note 7), C_F = 100 pF, R_F = 0Ω, no tank, 2.4 kHz = 4.8 kBaud	10	4.8	**2.4**	kBaud
29	PLL capture range, F_C	C_F = 100 pF, R_F = 0 Ω	±40	±15	±10	% min.
30	PLL lock range, F_L	C_F = 100 pF, R_F = 0 Ω	±45	±15		% min.
31	Receiver input sensitivity, S_{IN}	For a functional receiver (Note 8) Referred to chip side (pin 10) of the line-coupling XFMR: F_O = 50 kHz F_O = 300 kHz Referred to line side of XFMR: (assuming a 7.07:1 XFMR) F_O = 50 kHz F_O = 300 kHz	1.8 2.0 1.4 0.25 0.29 0.20	10	**12**	mV_{RMS} mV_{RMS} mV_{RMS} mV_{RMS} mV_{RMS} mV_{RMS}
32	Tolerable input dc voltage offset range, V_{INDC}	Pin 10 lower than pin 15 by V_{INDC}	2	0.1		V max.
33	Data Out, breakdown voltage	Pin 12, leakage I ≤ 20 μA	70	55		V min.
34	Data Out, low output, V_{OL}	Pin 12, sat. voltage at I_{OL} = 2 mA	0.15	0.4		V max.
35	Impulse noise filter current, I_I	Pin 13 charge and discharge current	±55	±45 ±85		μA min. μA max.
36	Offset hold cap. bias voltage, V_{CM}	Pin 6	2.0	1.3 3.5		V min. V max.
37	Offset hold capacitor max. drive current, I_{HCM}	Pin 6. V(pin 3) − V(pin 4) = ±250 mV	±55	±25 ±80		μA min. μA max.
38	Offset hold bias current, I_{OHB}	Pin 6, TX mode. Bias pin 6 as it self-biased during test 31.	−0.5	−20	−40 40	nA min. nA max.
39	Phase comparator current, I_{PC}	Bias pins 3 and 4 at 6.5 V I_{PC} = I(pin 3) + I(pin 4), TX mode	100	50 200		μA min. μA max.
40	Phase detector output resistance, R_{PD}	Pins 3 and 4. R_{PD} = (V@100μA − V@50μA)/(50μA)	10	6 18		kΩ min. kΩ max.
41	Phase detector demodulated output voltage, V_{PD}	Pin 3 to 4, measured after filtering out the $2F_O$ component	100	60 180		mVpp min. mVpp max.
42	Fast offset cancel voltage "window" -to-V_{PD} ratio, V_W/V_{PD}	$V_{PIN3} - V_{PIN4} = \pm V_{WINDOW}$ + DC offset Drive for ±1 μA pin 6 current	0.95	0.70 1.20		V/V min. V/V max.
43	Power supply rejection, PSRR	C_L = 0.1 μF. PSRR = CMRR. 120 Hz	80			dB min.

Note 1: More accurately, the maximum voltage allowed on pin 10 is V_{OC}, and V_{OC} ranges from 41 to 50V. Also, transients may reach above 60V; see the transient peak voltage characteristic curve.

Note 2: The maximum power dissipation rating should be derated for device operation above 25°C to insure that the junction temperature remains below the maximum rating. Use a θ_{JA} of 75°C/W for the N package using a socket in still air (which is the worst case). Consult the Application Information section for more detail.

Note 3: The boldface values apply over the full junction temperature range for the specified supply voltage range. All other numbers apply at $T_A = T_J = 25$°C. Pin numbers refer to LM1893. LM2893 tested by shorting Carrier In to Carrier Out and testing it as an LM1893.

Note 4: Guaranteed and 100% production tested.

Note 5: Guaranteed (but not 100% production tested) over the temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.

Note 6: Total harmonic distortion is measured using THD = [I_{RMS} (all components at or above $2F_O$)/I_{RMS} (fundamental)].

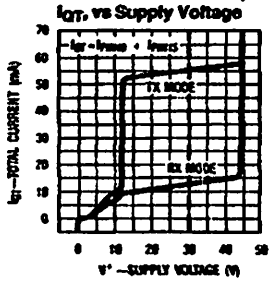
Note 7: Receiver function is defined as the error-free passage of 1 cycle of 50% duty-cycle 2.4 kHz square-wave data (2 sequential 208 μS bits), with the first bit being a "1." All of the data transitions (edges) must fall within ±10% (±20.8 μs) of their noise-free positions. RX time delay is minimized by using no impulse noise filter cap, C_I for this test.

Note 8: During the sensitivity check, note 7 requirements are followed with these exceptions: (1) data rate F_{DATA} = 1.2 kHz, (2) all of the data transitions must fall within ±20% (±41.6 μs) of their noise-free positions, and (3), a time-domain filter capacitor (C_I) is used. The time delay of C_I is ½ bit, or 208 μs. (C_I is approximately 6200 pF).

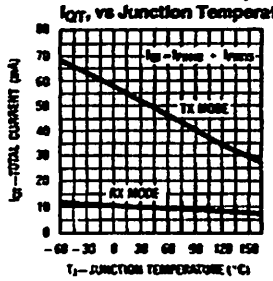
Note 9: For TTL compatibility use a pull-up resistor to increase min. V_{OH} to above 2.8 V.

Typical Performance Characteristics ($V^+ = 18V$, $F_O = 125$ kHz, circuit of *Figure 1*, pin numbers for LM1893)

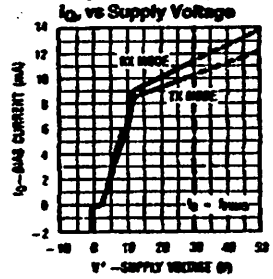
Total Current Consumption, I_{QT}, vs Supply Voltage



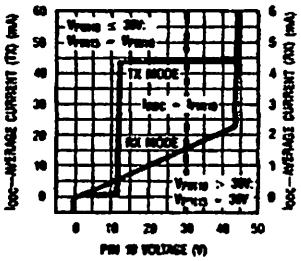
Total Current Consumption, I_{QT}, vs Junction Temperature



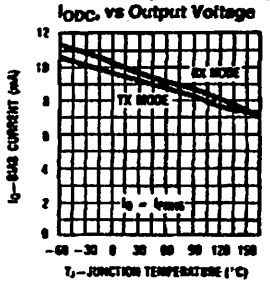
Chip Bias Current, I_Q, vs Supply Voltage



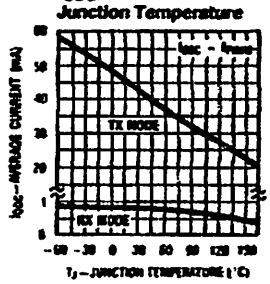
Chip Bias Current, I_Q, vs Junction Tempurature



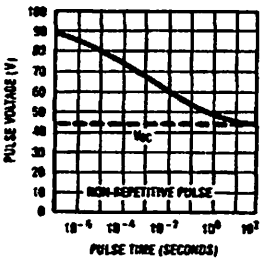
Output Stage DC Current, I_{ODC}, vs Output Voltage



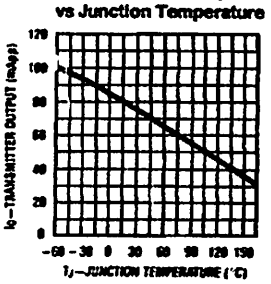
Output Stage DC Current, I_{ODC}, vs Junction Temperature



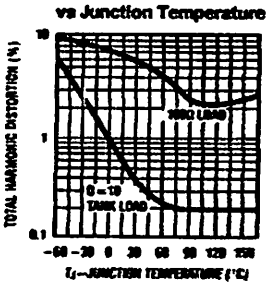
Transient Voltage Survival vs Pulse Time



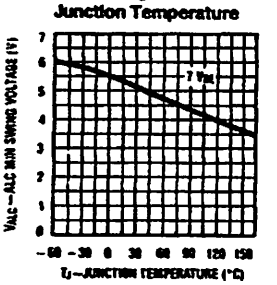
Transmitter AC Output Current vs Junction Temperature



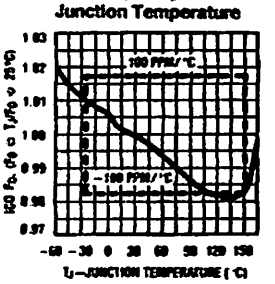
Transmitter Sinusoid THD vs Junction Temperature



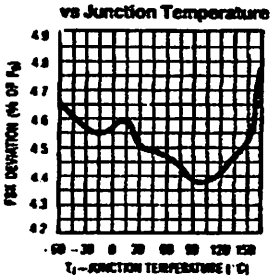
ALC Voltage vs Junction Temperature



fCO Frequency vs Junction Temperature



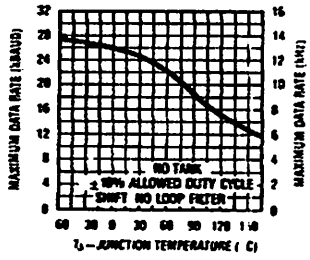
Transmitter FSK Deviation vs Junction Temperature



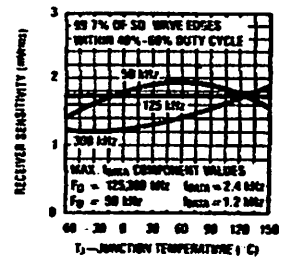
TL/H/10950–36

Typical Performance Characteristics (Continued)

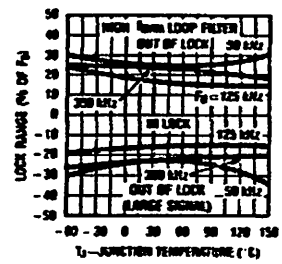
Maximum Data Rate vs Junction Temperature



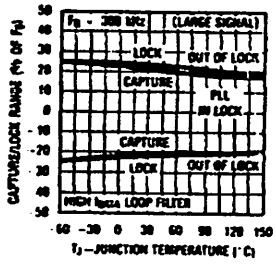
Receiver Sensitivity vs Junction Temperature



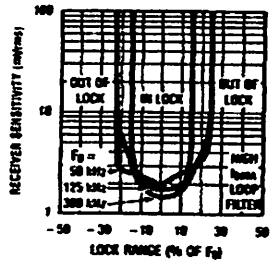
PLL Lock Range vs Junction Temperature and F_O



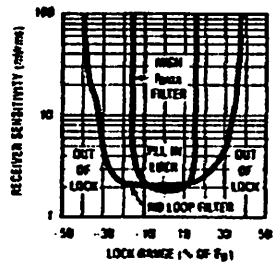
PLL Capture & Lock Range vs Junction Temperature



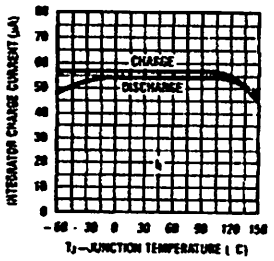
Receiver Sensitivity vs PLL Lock Range and F_O



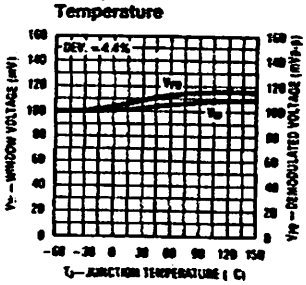
Receiver Sensitivity vs PLL Lock Range and Loop Filter



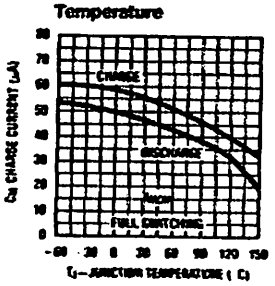
Impulse Noise Filter Current vs Junction Temperature



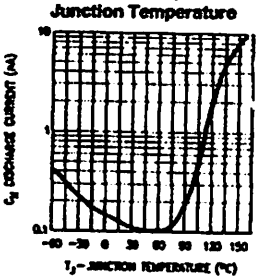
Phase Detector Output Voltage vs Junction Temperature



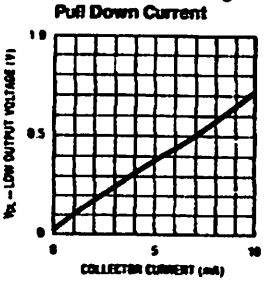
Offset Hold Cap. Charge Currents vs Junction Temperature



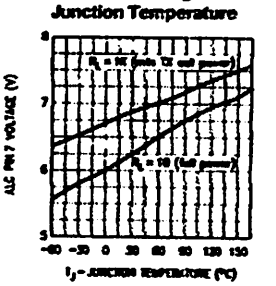
Offset Hold Cap. Bias Current vs Junction Temperature



Data Out Low Voltage vs Pull Down Current



Pin 7 Bias Voltage vs Junction Temperature



TL/H/6750–39

Application Information*

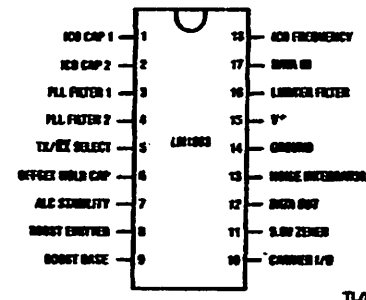
THE DATA PATH

The BI-LINE™ chip serves as a power line interface in the carrier-current transceiver (CCT) system of *Figure 3*. *Figure 4* shows the interface circuit now discussed. The controller may select either the transmit (TX) or receive (RX) mode. Serial data from the controller is used to generate a FSK-modulated 50 to 300 kHz carrier on the line in the TX mode. In the RX mode line signal passes through the coupling transformer into the PLL-based receiver. The recreated serial bit stream drives the controller.

With the IC in the TX mode (pin 5 a logic high), baseband data to 5 kHz drive the modulator's Data In pin to generate a switched 0.978I/1.022I control current to drive the low TC, triangle-wave, current-controlled oscillator to ±2.2% deviation. The tri-wave passes through a differential attenuator and sine shaper which deliver a current sinusoid through an automatic level control (ALC) circuit to the gain of 200 current output amplifier. Drive current from the Carrier I/O develops a voltage swing on T_1's (*Figure 4*) resonant tank proportional to line impedance, then passes through the step-down transformer and coupling capacitor C_C onto the line. Progressively smaller line impedances cause reduced signal swing, but never clipping-thus avoiding potential radio frequency interference. When large line impedances threaten to allow excessive output swing on pin 10, the ALC shunts current away from the output amplifier, holding the voltage swing constant and within the amp's compliance limit. The amplifier is stable with a load of any magnitude or phase angle.

In the RX mode (pin 5 a logic low), the TX sections on the chip are disabled. Carrier signal, broad-band noise, transient spikes, and power line component impinge of the receiver's input highpass filter, made up of C_C and T_1, and the tank bandpass filter. In-band carrier signal, band-limited noise, heavily attenuated line frequency component, and attenuated transient energy pass through to produce voltage swing on the tank, swinging about the positive supply to drive the Carrier I/O receiver input. The balanced Norton-input limiter amplifier removes DC offsets, attenuates line frequency, performs as a bandpass filter, and limits the signal to drive the PLL phase detector differentially. The differential demodulated output signal from the phase detector, containing AC and DC data signal, noise, system DC offsets, and a large twice-the-carrier-frequency component, passes through a 3-stage RC lowpass filter to drive the offset cancel circuit differentially. The offset cancelling circuit works by insuring that the (fixed) ±50 mV signal delivered to the data squaring ("slicing") comparator is centered around the 0 mV comparator switch point. Whenever the comparator signal plus DC offset and noise moves outside the carefully matched ±50 mV voltage "window" of the offset cancel circuit, it adjusts its DC correction voltage in series with the differential signal to force the signal back into the window. While the signal is within the ±50 mV window, the DC offset is stored on capacitor C_M. By grace of the highly non-linear offset hold capacitor charging during offset cancelling, the DC cancellation is done much more quickly than with an AC coupling capacitor normally used in place of the offset cancel circuit. Since impulse noise spikes normally ring the signal symmetrically around 0 V, the fully bilateral offset cancel topology affords excellent noise rejection. The switched current output of the comparator drives the impulse noise filter integrator capacitor that rejects all data pulses of less than the integrator charge time. Noise appears as duty-cycle jitter at the open collector serial data output.

Dual-In-Line Package

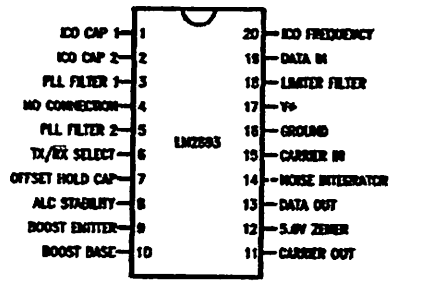


TL/H/6750–2

Top View

Order Number LM1893N
See NS Package Number N18A

Small Outline & Dual-In-Line Package

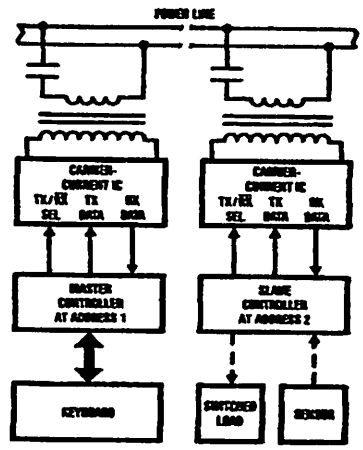


TL/H/6750–41

Top View

Order Number LM2893M or LM2893N
See NS Package Number M20B or N20A

FIGURE 2. Connection Diagrams

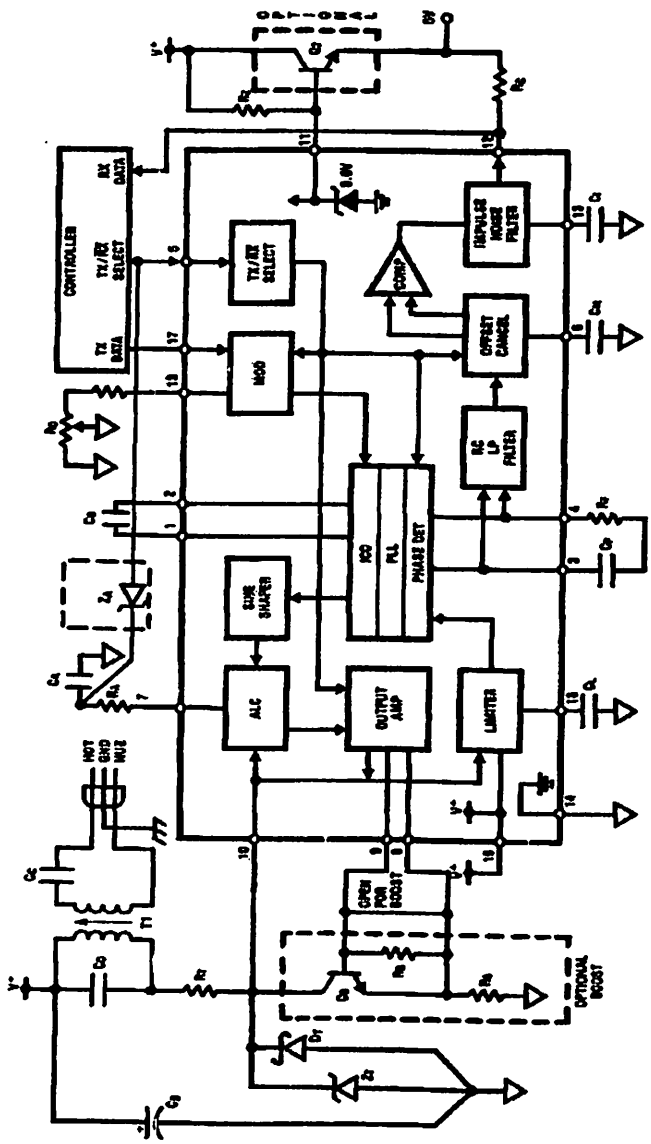


TL/H/6750–3

FIGURE 3. The block diagram of a carrier-current system using the Bi-Line chip to interface digital controllers via the power line

*Unless otherwise noted, all pin references refer to LM1893, but hold true for equivalent LM2893 pin.

Application Information (Continued)



TL/N/5750–4

FIGURE 4. Block diagram of a CCT system with the boost and 5V supply options shown in dashed boxes

Application Information (Continued)

#	Recommended Value	Purpose	Effect of making the component value:		Notes
			Smaller	Larger	
C_O R_O	560 pF 6.2 kΩ	Together, C_O and R_O set tCO F_O.	Increases F_O Increases F_O <5.6 k not recommended.	Decreases F_O Decreases F_O >7.6 k not recommended.	±5% NPO ceramic. Use low TC 2 k pot and 5.6 k fixed R. Poor F_O TC with <5.6 k R_O.
C_F R_F	0.047 µF 3.3 kΩ	PLL loop filter pole PLL loop filter zero	Less noise immune, higher f_{DATA}, more PLL stability. PLL less stable, allows less C_F. Less ringing.	More noise immune, lower f_{DATA}, less PLL stability. PLL more stable, allows more C_F. More ringing.	Depending on R_F value and F_O, PLL unstable with large C_F. See Apps. Info. C_F and R_F values not critical.
C_C	0.22 µF	Couples F_O to line, C_C and T_1 low-pass attenuates 60 Hz.	Low TX line amplitude. Less 60 Hz T_1 current. Less stored charge.	Drives lower line Z. More 60 Hz T_1 current. More stored charge.	≥250 V non-polar. Use $2C_C$ on hot and neutral for max. line isolation, safety.
C_O T_1	0.033 µF Use recommended XFMR	Tank matches line Z, bandpass filters, isolates from line, and attenuates transients.	Tank F_O up or increase L of T_1 for constant F_O. Smaller L: higher F_O or increase C_C; decreased F_O line pull.	Tank F_O down or decrease L of T_1 for constant F_O. Larger L: lower F_O or decrease C_C; increased F_O line pull.	100 V nonpolar, low TC, ±10% High large-signal Q needed. Optimize for low F_O line pull with control of F_O TC and Q.
C_A R_A	0.1 µF 10 kΩ	ALC pole ALC zero	Noise spikes turn ALC off. Less stable ALC.	Slower ALC response. More stable ALC.	R_A optional. ALC stable for C_A ≥ 100 pF.
C_L	0.047 µF	Limiter 50 kHz pole, 60 Hz rejection.	Higher pole F, more 60 Hz reject. F_O attenuation?	Lower pole F, less 60 Hz reject, more noise BW.	Any reasonably low TC cap. 300 pF guarantees stability.
C_M	0.47 µF	Holds RX path V_{OS}	Less noise immune, shorter V_{OS} hold, faster V_{OS} acquisition, shorter preamble.	More noise immune, longer V_{OS} hold, slower V_{OS} acquisition, longer preamble.	Low leakage ±20% cap. Scale with f_{DATA}.
C_I	0.047 µF	Rejects short pulses like impulse noise.	Less impulse reject, less delay, more pulse jitter.	More impulse reject, more delay, less pulse jitter.	C_I charge time ½ bit nom. Must be <1 bit worst-case.
R_C	10 kΩ	Open-col. pull-up	Less available sink I.	Less available source I.	R_C ≥ 1.5 kΩ on 5.6 V
R_Z	12 kΩ	5.6 V Zener bias	Larger shunt current, more chip dissipation.	Smaller shunt current, less V^+ current draw.	1 < I_Z < 30 mA recommended. (Chip power-up needs 5.6 V)
Z_T R_T D_T	≥44 V BV <60 V peak 4.7 Ω ≥44V BV	Transient clamp Transient I limit Over-drive Clamp	Z_T failure, higher series R-excess peak V, Zener and chip damage, less ruggedness. Damage Z_T, pull up V^+. Failure on Transient	Z_T costly, lower series R gives enhanced transient clamp, more ruggedness. Excessive TX attenuation. Costly	Recommend Zener rated for ≥ 500 W for 1 ms. Carbon comp. recommended. IRF 11DQ05 or 1N5819
R_B Q_B R_G	180 Ω Power NPN 1.1 Ω	Base bleed Boost gain device Current setting R	Faster, lower THD I_O. Excessive T_J and V_{SAT}. More I_O, need higher h_{fe}.	Inadequate turn-off speed. More rugged, but costly. Less I_O, lower min. h_{fe}.	Boost optional. Q_B F(−3 dB) of >200 MHz. R_B > 24 Ohm. $I_O = 70[(10 + R_G)/R_G]$ mApp.
C_B	≥4.7 µF	Supply bypass	Transients destroy chip.	Less supply spike.	V^+ never over abs. max.
Z_A	5.1V	Stop ALC charge in RX mode	Excess ALC current flow	ALC RX charging not inhibited over T_J	Z_A optional - 5.1V ±20% low leakage type

FIGURE 5. A quick explanation of the external component function using the circuit of *Figure 4*. Values given are for V^+ = 18 V, F_O = 125 kHz, f_{DATA} = 300 Baud (150 Hz), using a 115 V 60 Hz power line

Component Selection

Assuming the circuit of *Figure 4* is used with something other than the nominal 125 kHz carrier frequency, 150 Hz data rate, 18V supply voltage, etcetera, the component values listed in *Figure 5* will need changing. This section will help direct the CCT designer in finding the required component values with emphasis placed on look-up tables and charts. It is assumed that the designer has selected values for carrier center frequency, F_O; data rate, f_{DATA}; supply voltage, V^+; power line voltage, V_L; and power line frequency, F_L. If one or more of those parameters is not defined, one may read the data sheet and make an educated guess.

Maxims to keep in mind, based on CCT electrical performance considerations only, are: 1) the higher the F_O the better, 2) the lower the maximum data rate the better, and 3) the more time and frequency filtering the better.

Use *Figure 5* as a quick reference to the external component function.

THE TRANSMITTER

C_O

Central to chip operation is the low TC of F_O emitter-coupled oscillator. With proper C_O, the F_O of the $2V_{BE}$ amplitude triangle-wave oscillator output may vary from near DC to above 300 kHz. While C_O may have any value, C_O should

Component Selection (Continued)

be made above 10 pF so that parasitic capacitance is not dominant. Excessive or unbalanced common-mode-to-ground capacitance should be avoided. A low temperature coefficient (TC) of capacitance (<100 PPM/°C), such as a monolithic NPO ceramic multilayer type, preserves low TC of F_O. *Figure 6* finds a C_O value given F_O.

R_O

Resistor R_O is used by the IC to generate a V_{BE}/R related current that is multiplied by 2 to produce the 200 µA ICO control current that sets F_O. The control current TC "bucks" the V_{BE} related tri-wave amplitude across C_O to effect a low TC of F_O. Vary R_O to trim F_O, within limits. Raising F_O more than 20% above its untrimmed value by means of decreasing R_O more than 20% is not recommended. Low R_O, and so high control current, risks ICO saturation and poor TC under worst-case conditions. Raising R_O reduces the demodulated signal amplitude from the phase detector; raising R_O by more than a factor of 2 (1 octave) is not recommended.

Since lower TC pots are relatively costly, it is recommended that R_O be made up of a 5.6 k fixed (<100 PPM/°C) resistor with a 2 kΩ (<250 PPM/°C) series pot.

C_A and R_A

Components C_A and R_A control the dynamic characteristics of the transmitter output envelope. Their values are not critical. Use the values given in *Figure 5*. C_A and R_A are functions of loaded T_1 tank Q, R_O, f_{DATA}, and line impulse noise. Any changes made in C_A and R_A should be made based on empirical measurements of a CCT on the line. Roughly, C_A acts as an ALC pole and R_A an ALC zero.

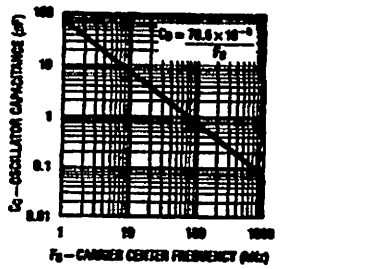
T_1

At this point, the CCT system designer may choose to use one of the recommended transformers or to design custom T_1. Consult "The Coupling Transformer" section to help with the design of T_1 if a new or boost-capable transformer is needed. The recommended 125 kHz transformer functions with an I_O of up to 600 mApp.

It is recommended that CCT systems use the recommended transformers, described in *Figure 7*, for T_1. The 3 transformers are optimized for use in the ranges of 50–100 kHz, 100–200 kHz, and 200–400 kHz with unloaded Q's (Q_U) of about 35, and loaded Q's (Q_L) of about 12. Three secondary taps are supplied with nominal 7.07, 10, and 14.1 turns ratios (N) to drive industrial and residential power line impedances of 3.5, 7, and 14Ω respectively. All are inexpensive, all have the same pin-outs for easy exchange in a PC board, and all are small - on the order of 10 mm diameter at the base.

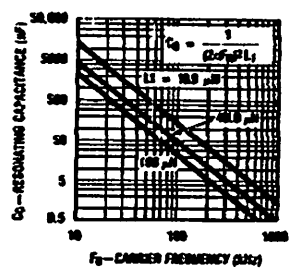
C_Q

Tank resonant frequency F_Q must be correct to allow passage of transmitter signal to the line. Use *Figure 8* to find C_Q's value. Trimming F_Q to equal F_O is done with T_1's trimming slug. The inductance of T_1 has a TC of +150 PPM/°C which may be cancelled by using a −150 PPM/°C cap such as polystyrene. Since circulating current in the tank is ⅓ A_{RMS}, C_Q should have a low series resistance (a 1 Ω series resistance is too much). Polypropelene caps are excellent, "orange drop" mylars are adequate, while many other mylars are inadequate. A 100V rating is needed for transient protection.



TL/H/6750–5

FIGURE 6. Find C_O's value knowing F_O



TL/H/6750–10

FIGURE 8. Find C_Q's value given F_O

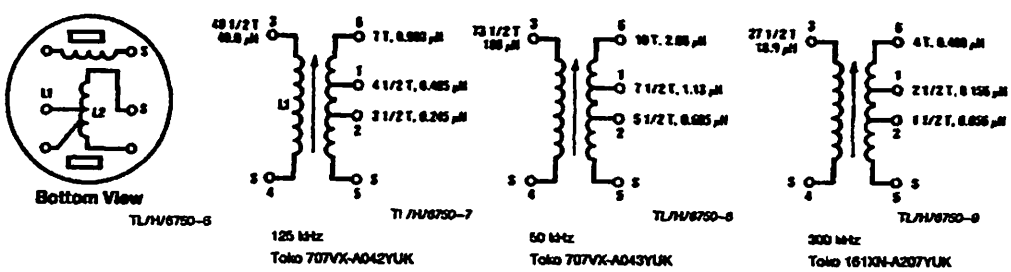


FIGURE 7. The recommended T_1 transformers, available through: Toko America, 1250 Feehanville Drive, Mount Prospect, IL, 60056, (312) 297-0070

Component Selection (Continued)

C_C

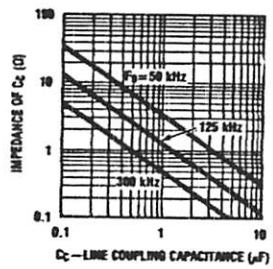
Capacitor C_C's primary function is to block the power line voltage from T_1's line-side winding. Also, C_C and T_1's line-side winding comprise a LC highpass filter. The self-inductance of T_1 is far too low to support a direct line connection. C_C must have a low enough impedance at F_O to allow T_1 to drive transmitted energy onto the line. To drive a 14Ω power line, the impedance of C_C should be below 14Ω.

Use *Figure 9* to find the reactive impedance of C_C to check that it is less than the line impedance. Then check *Figure 10* to see that the power line current is small enough to keep T_1 well out of saturation; the recommended transformers can withstand a 10 Amp-turn magnetizing force (1 Amp through the worst-case 10 turn line-side winding).

Caution is required when choosing C_C to avoid series resonance of the series combination of C_C, the transformer inductance, and the reflected tank impedance. The low resistance of the network under series resonance will load the line, possibly decreasing range. For your particular line coupling circuit, measure for series resonance using some expected line impedance load.

R_B

This base-bleed resistor turns Q_B off quickly - important since the amplifier output swing is about 200V/μs. An R_B below about 24Ω will conduct excessive current and overload the chip amplifier and is not recommended.



TL/H/6750–11

FIGURE 9. C_C's impedance should be, as a rule-of-thumb, smaller than the lowest expected line impedance

R_G

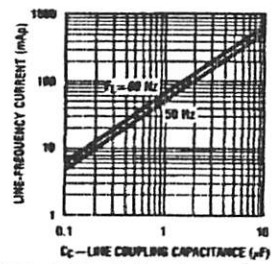
This resistor, in parallel with the internal 10Ω resistor, fixes the current gain of the output amplifier, and so the output current amplitude. *Figure 11* gives output current and minimum AC current gain h_{fe} for Q_B when R_G is used to boost output current.

Q_B

The boost gain transistor Q_B must be fast. Double-diffused devices with 50 MHz F_T's work, slower transistors (epi-base types) do not preserve a sinusoidal waveform when F_O is high or will cause the output amp. to oscillate. Q_B must have a certain minimum h_{fe} for given boost levels, as shown in *Figure 11*. *Figure 12* shows the power Q_B must dissipate continuously operating with a shorted output. BV_{CER} ($R = R_B$) must be 60V or greater and Q_B must have adequate SOA for transient survival.

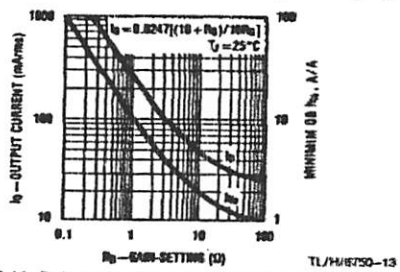
Z_T

Unfortunately, potentially damaging transient energy passes through transformer T_1 onto the Carrier I/O pin (instantaneous power of greater than 1 kW has been measured using the recommended transformers). For self protection, the Carrier I/O has an internal 44V voltage clamp with a 20Ω series resistance. A parallel low impedance 44V external transient suppression diode will then conduct the lion's share of any current when transients force the Carrier I/O to a high voltage.



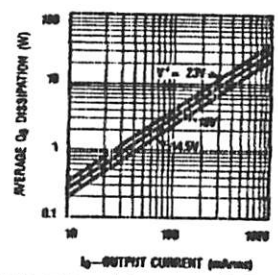
TL/H/6750–12

FIGURE 10. The AC line-induced current passed by C_C



TL/H/6750–13

FIGURE 11. Output amplifier current and required min. Q_B h_{fe} versus gain-setting resistor R_G



TL/H/6750–14

FIGURE 12. Boost transistor power dissipation versus amplifier output current

Z_T must be used unless some precaution is taken to protect the Carrier I/O pin from line transients or transients caused when stored line energy in C_C is discharged by the random phase of power line connection and disconnection. Worst case, C_C may discharge a full peak-to-peak line voltage into the tuned circuit. Another way to reduce the need for Z_T is by placing another magnetic circuit in the signal path that relies on a high, but easily saturated, permeability to couple a primary and secondary winding - a toroidal transformer for example. Toroids cost more than Z_T.

Use an avalanche diode designed specifically for transient suppression — they have orders of magnitude higher pulse

Component Selection (Continued)

power capability than standard avalanche diodes rated for equal DC dissipation. Metal oxide varistors have not proven useful because of their inferior clamping coefficient and are not recommended. Specifications for an example minimum diode are given in *Figure 13*.

Breakdown Voltage	44–49V @ 1 mA
Maximum Leakage	1 μA @ 40V
Capacitance	300 pF @ 8V
Maximum Clamp Voltage	64.5V @ 7.8A
Peak Non-Repetitive Pulse Power (REA Standard Exponential Pulse)	10 kW for 1 μs
Surge Current	70A for 1/120s

FIGURE 13. Key specifications for a recommended transient suppressor Z_T available from General Semiconductor, 2001 West Tenth Place, Tempe, AZ 85281, 602–968-3101, part no. SA40A

R_T

R_T acts as a voltage divider with Z_T, absorbing transient energy that attempts to pull the Carrier Input pin above 44V. Make the resistor a carbon composition 1/4W. When experiments discharging C_C charged to the peak-to-peak 620V AC thru a 1Ω power line were carried out, film resistors blew open-circuit.

D_T

This Schottky diode is placed in parallel with the CCT chip's substrate diode to pass the majority of the current drawn from ground when the Carrier Input or Carrier Output is pulled below ground by a larger-than-twice-the supply-swing on the tank. Note that Z_T is in parallel with the substrate diode, but is ineffective due to its high forward voltage drop and high diffusion capacitance caused by its low forward speed. Tests proved that a 1N5818 kept a receive-path functional with a 20X boost transmitter with a 7:1 transformer attempted to swing the receiver's Carrier I/O to ± 100V (300 mA peak ground current in the receiver). Without D_T, the receiver momentarily stops functioning at a 100 times lower ground current.

This diode is not needed if the Carrier I/O never swings below ground. If your CCT systems all run on the same regulated voltage with all matched transformers and turns ratios, it is not needed. Otherwise, it is.

THE RECEIVER

The receiver and transmitter share components C_C, T_1, C_O, R_T, Z_T, C_O, R_O, and peripheral supply and bias components that are not in need of change for RX mode operation. Values for the balance of the components are now found.

Line-Frequency Rejection

To use the ultimate sensitivity of the device, fully 110 dB of 115 V, 60 Hz attenuation is required between the line and the limiter amplifier output. Using the circuit topology of *Figure 4*, the combined attenuation of the C_C/T_1 highpass, the tuned transformer, and the bandpass filter attenuation of the limiter amplifier give far more line rejection than the above-stated minimum. However, if some other CCT line coupling circuit is used, line rejection will become important to the system designer.

Receiver input power supply rejection (PSRR) and common-mode rejection (CMRR) are one-in-the-same using the supply-referenced signal input of *Figure 4*. Ripple swings both differential inputs of the Norton amp. equally, while the single-ended input signal swings only the positive input. Overall PSRR consists of the input CMRR (set by the input stage component matching) and the ripple-frequency attenuation of the input amplifier bandpass response that passes carrier frequency but stops low frequencies. A typical 1% resistor and 1 *mV n-p-n mirror offsets give 26 dB of attenuation, the* bandpass gives 54 dB 120 Hz attenuation, for an overall 80 dB PSRR to allow tens of volts of ripple before impacting ultimate sensitivity.

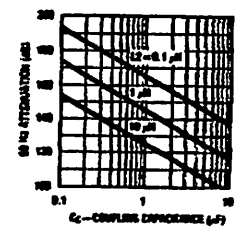
C_C

A value was chosen earlier. Knowing T_1's secondary inductance allows a check of LC line attenuation using *Figure 14*.

C_L

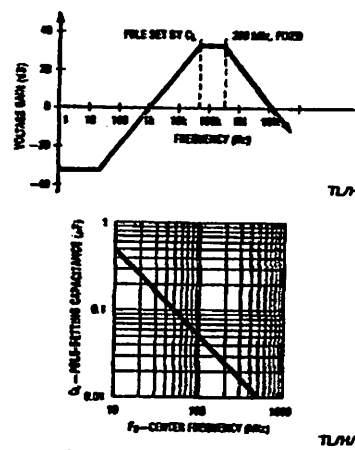
The Norton input limiter amplifier has a bandpass filter for enhanced receiver selectivity, noise immunity, and line frequency rejection. The nominal response curve for F_O = 50 kHz is shown in *Figure 15*. The 300 kHz pole is fixed. The 50 kHz pole is set by C_L's value. After C_L is found, the resulting line frequency attenuation is found for the bandpass filter.

Use *Figure 15* to find a C_L value given for F_O. The approximate line frequency attenuation of the bandpass filter may then be found in *Figure 16*. *Figure 15* returns a value for C_L 33% larger than nominal, giving a low frequency pole 33% low to allow for component tolerances.



TL/H/6750–15

FIGURE 14. The 60 Hz line rejection of the highpass filter made up of C_C and T_1's line-side winding (neglecting capacitive coupling)



TL/H/6750–16

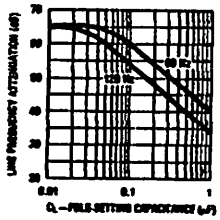
TL/H/6750–17

FIGURE 15. Given F_O, C_L is found. Also shown is the input amplifier's small signal amplitude response

Component Selection (Continued)

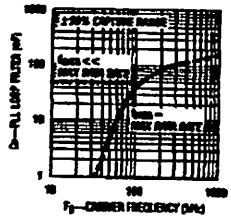
C_F and R_F

These phase-locked loop (PLL) loop filter components remove some of the noise and most of the $2F_O$ components present in the demodulated differential output voltage signal from the phase detector. They affect the PLL capture range, loop bandwidth, damping, and capture time. Because the PLL has an inherent loop pole due to the integrator action of the ICO (via C_O), the loop pole set by C_F and the zero set by R_F gives the loop filter a classical 2nd-order response.



TL/H/6750–18

FIGURE 16. The Norton-input limiter amplifier bandpass filter line-frequency signal attenuation given C_L



TL/H/6750–19

FIGURE 17. Find C_F given F_O. *Figure 19* gives the maximum data rate

No C_F and R_F give the most stable PLL with the fastest response. Large C_F's with a too-small R_F cause PLL loop instability leading to poor capture range and poor step response or oscillation.

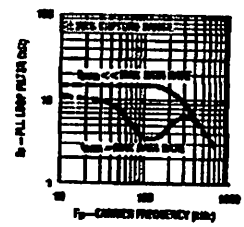
Calculation of C_F and R_F is quite difficult, involving not only the 2nd-order loop step response, but also the PLL non-dominant poles, the tuned transformer stepped-frequency response, and the RC lowpass step response (for data rates approaching 1 kHz). C_F and R_F values are best found empirically. Tolerance is not critical. Component values are selected to give the best possible impulse noise rejection while preserving a ±20% capture range and wide stability margin. *Figures 17* and *18* give C_F and R_F values versus F_O, where "f_{DATA} << MAX DATA RATE" means that f_{DATA} should be less than the maximum data rate, in kHz, from *Figure 19* divided by 10.

Note that C_F and R_F are a function of data rate only for high data rates and are not plotted against data rate - as one might expect. The reason for this is important to understand if the CCT system designer wishes to find C_F and R_F empirically. Data signal is, loosely speaking, passed through the PLL loop and is therefore potentially attenuated if the loop bandwidth is on the order of the 3rd harmonic of the data rate, or less. Overall loop bandwidth is held as low as possible for maximum noise rejection while passing the data. Loop bandwidth is roughly proportional to the geometric mean of the unfiltered loop bandwidth and the filter pole set by C_F. Therefore, C_F is related to data rate. Unfortunately, the loop capture range falls to critically low values when large enough values of C_F are used to reduce loop bandwidth down to the 100's of Hz range, for low data rates. The obvious way out is to then reduce the unfiltered loop bandwidth. That bandwidth is approximately proportional to the value of C_O. For a fixed F_O, unfiltered loop bandwidth reduction requires a larger C_O and larger control current. With this chip, changing the control current is not allowed. So one is forced to choose a C_F/R_F combination with some minimum capture range, say ±20%, that is within some guardband from the point of loop instability. Happily, impulse noise tends to last only fractions of a millisecond so that the lack of low bandwidth loop response with low data rates is not a heavy penalty. As long as there is adequate capture range, the impulse noise filter performs admirably. Note that reducing F_O will reduce the no-filter loop bandwidth, and indeed the maximum data rate falls below the limit set by the RC lowpass filter as F_O falls below 100 kHz (*Figure 19*).

The tuned transformer characteristics will affect the demodulated data waveform more than C_F and R_F at low data rates. Tank Q and off-tuning will affect overshoot during the FSK frequency steps. This is a property of tuned circuits. The maximum data rate of *Figure 19* is measured from the receiver input to the Data Out and does not include the data bandwidth reducing effects of T_1.

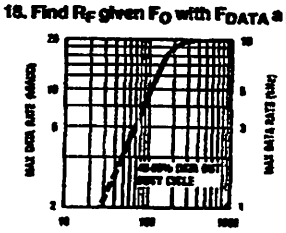
C_M

Capacitor C_M stores a voltage corresponding to a correction factor required to cancel the phase detector differential output DC offsets. The stored voltage is ½ of the DC offset plus some bias level of about 2.2 V. A large C_M value increases the time required to bias-up the receive path at the beginning of transmission. A large C_M does filter well and store its bias voltage long. Because of the initial random charge of C_M, the receiver must be given a data transition to charge to the proper bias voltage. Therefore, reducing C_M's value to one that may be charged in less than 2 bit-times will not save biasing time and is not recommended.



TL/H/6750–20

FIGURE 18. Find R_F given F_O with F_{DATA} a parameter

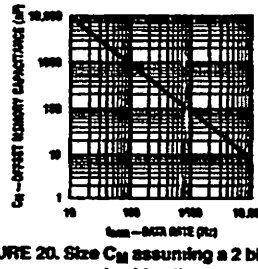


TL/H/6750–21

FIGURE 19. The maximum data rate versus F_O using loop filter components optimized for max. noise performance while retaining a min. ±20% capture range (large signal)

Use *Figure 20* to find C_M's value knowing f_{DATA}, assuming the standard 2 bit receive charge time is desired. The cap. value and TC are not critical, but the capacitor should have low leakage.

Component Selection (Continued)



TL/H/6750–22

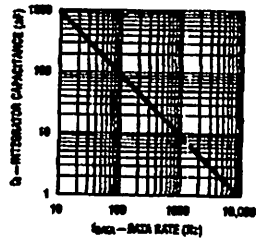
FIGURE 20. Size C_M assuming a 2 bit-time receive bias time

C_I

The impulse noise filter integrator capacitor C_I is used to disallow the passage of any pulse shorter than the integrator charge time. That charge time, set to a nominal ½ bit time, is the time required for a ±50 µA charge current to swing C_I over a 2 V_{BE} range. Charge time under worst case conditions must never be greater than a bit time since no signal could then pass. Using a ±10% capacitor, full junction temperature range, and full specified current range, a maximum nominal charge time of ½ bit is recommended. *Figure 21* gives C_I versus data rate under those conditions.

R_C

The collector pull-up resistor is sized to supply adequate pull-up current drive and speed while preserving adequate output low current drive.



TL/H/6750–24

FIGURE 21. Impulse noise filter cap. C_I versus F_{DATA} where the charge time is ½ bit time

Z_A

The 5.1V silicon zener diode Z_A is required when a short RX-to-TX switch-over time is needed at the same time that the chip is operating in the RX mode with a pin 10 input signal swing approaching or exceeding twice the supply voltage. Predominant causes of these large swings impinging on the RX input are: 1) a transmitter's supply voltage higher than the receiver's supply voltage, 2) a TX and RX pair that are electrically close, or, 3) a higher RX T_1 step-up turns ratio than the TX T_1 step-down ratio.

Normally, when in the RX mode with small incoming signal on pin 10, the ALC remains off with pin 7 at a 6V ($V_Z - 2V_{BE}$) bias voltage. C_A is then charged to 6V. TX mode may then be selected with 6V on C_A allowing 100% TX power to pump T_1's tuned circuit, and so the AC line, quickly for fast RX-to-TX switch time. As TX output swing increases so that pin 10 swings below V_{ALC} (4.7V typically), that ALC activates to charge C_A to about 6.6V to reduce TX output drive. However, if in the RX mode pin 10 ever swings below V_{ALC}, C_A will charge to above 6.6V. Now, when the TX mode is selected with C_A at 6.6V, somewhere from 0 to 100% TX output drive is available to pump T_1's tuned circuit resulting in a slower rising line signal - effectively reducing the RX-to-TX switch time.

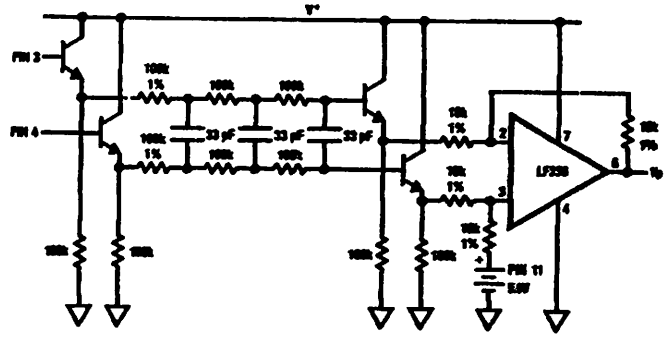
Use a 5.1V Z_A driven by a 0 to 0.6V logic low signal to guarantee over-temp. operation. R_A must be in series with Z_A to limit current flow and should never fall below 1 kΩ. If R_A is less than 1 kΩ, then put a 2 kΩ resistor in series with Z_A. Logic high voltages above 10V will cause current flow into pin 7 that must be limited to 1 mA (with R_A or a series R).

Breadboarding Tips

During CCT system evaluation, some techniques listed below will simplify certain measurements.

— Use caution when working on this circuit - dangerous line voltages may be present.

— When evaluating PLL operation, offset cancel circuit operation, and loop filter values, use the filter of *Figure 22* to view the demodulated signal minus the $2F_O$ and noise components. This filter models the RC lowpass filter on chip.



TL/H/6750–25

FIGURE 22. Circuit to view the differential demodulated data signal, minus the noise and $2F_O$ components, conveniently with a single-ended gain-of-one output

Breadboarding Tips (Continued)

— When evaluating CCT system noise performance on a real power line, it is desirable to vary the signal amplitude to the receiver. This is not easy. An in-line line-proof L-pad is fine except that the line impedance is unknown and variable and so the L-pad will rarely match. Instead, the power output of a chip transmitter may be controlled using the circuit of *Figure 23*. This circuit controls the ALC.

— It is sometimes desirable to place impulse noise on the line. A simple light dimmer with a 100 W light bulb load produces representative impulse noise.

— Do not allow peak currents of over 1 A through the 5.6 V Zener. In other words, don't short charged capacitors into this low-impedance device. Take care not to momentarily short pins 10 and 11 - chip damage may result.

— *Figure 24* shows some typical signals beginning with serial data transmitted to received signal.

Tuning Procedure

This procedure applies to circuits similar to *Figure 4* LM1893 or LM2893 circuit.

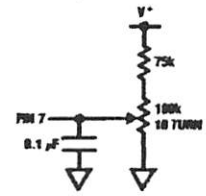
First, trim F_O by putting the chip in the TX mode, setting a logical high data input, and measuring the TX high frequency, 1.022 F_O, on the Carrier I/O using these steps:

1. Take pin 17 to a logic low.
2. Take pin 5 to a logic high.
3. Place a counter on pin 10.
4. Adjust R_O on pin 18 for $F = 1.022F_O$.

Second, the line transformer is tuned. The chip is placed in the TX mode, a resistive line load is connected to disable the ALC by reducing tank voltage swing below its limit. FSK data is then passed through the tank so that the tank envelope may be adjusted for equal amplitude for high and low data frequency.

1. Take pin 5 to a logic high.
2. Place a logic-level square wave at or below the receiver's maximum data rate on pin 17.
3. Temporarily place a 330 Ω resistor across the tank.
4. Place a scope on pin 10.
5. Adjust the transformer slug for the least envelope modulation.

In lieu of the 330 Ω resistive load, T_1 may be coupled to the power line to better simulate actual load and tank pull conditions during tank tuning. Alternatively, a passive network representing an average line impedance may be connected to the line side of T_1. The circuit of *Figure 23* should then be used to defeat the leveling effect of the ALC.



TL/H/6750–35

FIGURE 23. A means of transmitter output amplitude control is shown

Thermal Considerations

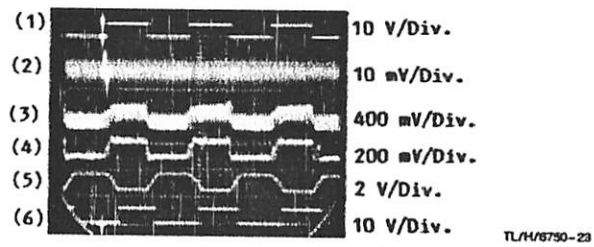
It is desirable to place the largest possible signal on the power line for maximum range, limited only by the chip power dissipation and maximum junction temperature T_J. The falling output power at elevated T_J allows a more optimal power output - high power at low T_J and lower power at high T_J for chip self-protection. However, it is still possible to exceed the maximum T_J within the specified ambient temperature limit (T_A = 85°C) under worst case conditions of 100% TX duty cyle, high supply, shorted load, poor PC board layout (with small copper foil area), and an above nominal current part. Under those conditions, a part may dissipate 2140 mW, reaching a T_J = 170°C worst-case (admittedly a rare occurrence). Proper system design includes the measurement or calculation of T_J max. to guarantee function under worst-case operation. Like all devices with failure modes modeled by the Arrhenius model, the high chip reliability is further enhanced by keeping the die temperature mercifully below the absolute maximum rating.

A direct method of measuring operating junction temperature is to measure the V_{BE} voltage on pin 18, which is always available under all operating modes. The graph of *Figure 25* may be used to find T_J, knowing V_{BE} at the operating point in question and V_{BE} at $T_A = T_J = 25°C$. V_{BE} is found by powering up a chip (in RX mode) that has been dissipating zero power at some T_A for some time and measuring V_{BE} in less than 1 s (for better than 5°C accuracy).

Alternately, T_J may be calculated using:

$$T_J = T_A + \theta_{JA}P_D \quad (1)$$

where θ_{JA} is 75°C/W for the plastic (N) package using a socket. That θ_{JA} value is for a high confidence level; nomi-

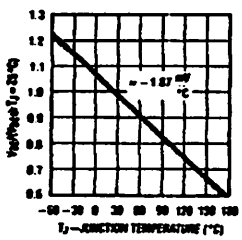


TL/H/6750–23

FIGURE 24. Oscillogram revealing signals at several important nodes under weak signal (0.5 mV_{RMS}) conditions with SCR spikes on an otherwise quiet 115 V, 60 Hz power line. The signals are: 1) transmitted data, 2) RX carrier on the tuned transformer, 3) demodulated signal from the PLL after passing thru circuit of *Figure 22*, 4) signal after RC lowpass, 5) data at impulse noise filter integrator, and 6) received data. Horizontal scale is 10 ms per div.

Thermal Considerations (Continued)

nal θ_{JA} for an N package is 60°C/W, lower with good PC board layout. Since P_D is a relatively strong function of T_J, an iterative solution process starting with an initial guess for T_J is used. With the estimated T_J, find the total supply current found in the typical performance characteristics.



TL/H/6750–27

FIGURE 25. T_J may be found by using the temperature coefficient of pin 18 V_{BE} if V_{BE} is known at 25°C

Transmit-To-Receive Switch-Over Time

An important figure-of-merit for a half-duplex CCT link, affecting effective data rate, is the TX-to-RX switch time T_{TR}. Using the recommended component values gives this part a nominal 2 bit-time (1 bit time = $1/[2f_{DATA}]$) over a wide range of operating conditions, where the receiver requires 1 data transition. T_{TR} cannot be decreased significantly but does increase as noise filtering, especially via C_M, is increased. Impulse noise at switch, signals near the limiting sensitivity, poor F_O match between receiver and transmitter because of poor trim or worst-case conditions, and the statistical nature of PLL signal acquisition may all contribute to increase T_{TR} to possibly 4 bit-times.

T_{TR} is lower when a pair of LM1893's handshake rapidly. The receiver was designed to "remember" the RX-mode DC operating points on C_M and C_F while in the TX mode. Under noisy worst case conditions, C_M will discharge to the point of false operation after 35 bit-times in the TX mode (1400 bit times with no noise and a nominal part, f_{DATA} = 180 Hz). T_{TR} is about 0.8 ms (proportional to the selected F_O) plus ½ bit-time.

The major components of T_{TR} are described below for a nominal 125 kHz F_O, 180 Hz f_{DATA}, lightly-loaded tank with a Q of 20, and the circuit of *Figure 4*. The remote CCT has been operating in the TX mode with a 26.6 V_{PP} tank swing and is now selected as a receiver. An incoming signal requiring the ultimate receiver sensitivity immediately is placed on the line.

First, the tank stored energy at the transmit frequency must decay to a level below the 2.8 mV_{PP} swing caused by the 0.14 mV_{RMS} incoming line signal containing the information to be received.

$$\text{decay time} = \frac{Q}{\pi F_O} \ln\left(\frac{V_1}{V_O}\right) =$$

$$\frac{20}{\pi \times 125\,000} \ln\left(\frac{26.6}{0.0028}\right) = 0.468 \text{ ms} \qquad (2)$$

That is 0.47 ms of delay (proportional to $1/F_O$ and Q).

Second, the PLL must acquire the signal; it must lock and settle. Acquisition time is statistical and may take any length of time, but average acquisition time depends on the loop filter components C_F and R_F and the difference in center frequencies, ΔF_O, of the TX/RX pair. Using the recommended C_F and R_F (47 nF and 6.2 kΩ) with a ± 4.4% ΔF_O (a ± 100 mV DC offset on C_F and R_F), lock was measured to take less than 50 cycles of F_O. That is a 0.40 ms delay (proportional to $1/F_O$).

Acquisition is incomplete until the second order PLL loop settles. For the above-mentioned C_F and R_F, the loop natural frequency F_N and damping factor are found to be 2.3 kHz and 1.0 respectively. Settling to within ± 25 mV of the ± 100 mV DC offset change requires 2.7 periods of F_N, or 1.2 ms (a function of C_F and R_F).

Third, the RC lowpass filter introduces a 0.12 ms delay.

Fourth, C_M must charge up to ± (¼)100 = 83 mV depending on the polarity of F_O. Borderline data squaring with zero noise immunity is possible with only ± (¼) 50 mV of charging. C_M charge current is an asymptotic function approximated by assuming a 50 µA charge current and the full 83 mV charge voltage. C_M charge time is then 1.7 ms (proportional to $1/f_{DATA}$).

Fifth, the impulse noise filter adds a ½ bit-time delay. Total T_{TR} is 3.9 ms plus ½ bit-time for a total of 1.9 bit-times at 360 Baud.

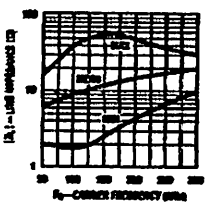
Receive-To-Transmit Switch-Over Time

Assume the chip has been in the RX mode and the TX mode is now selected. In less than 10 µs, full output current is exponentially building tank swing. 50% of full swing is achieved in less than 10 cycles - or under 80 µs at 125 kHz. In the same 10 µs that the output amp went on, the phase detector and loop filter are disconnected and the modulator input is enabled. FSK modulation is produced in 10 µs after switching to TX mode.

Power Line Impedance

Irrespective of how wide the limits on power line impedance Z_L are placed, there are no guarantees. However, since the CCT design requires an estimate of the lowest expected line impedance Z_{LN} encountered for the most efficient transmitter-to-line coupling, line impedance should be measured and Z_L limits fixed to a given confidence level. Reasonable values for T_1 turns ratio, loaded Q, and tank resonant frequency pull F_O may be found to enable a CCT system design that functions with the overwhelming majority of power lines.

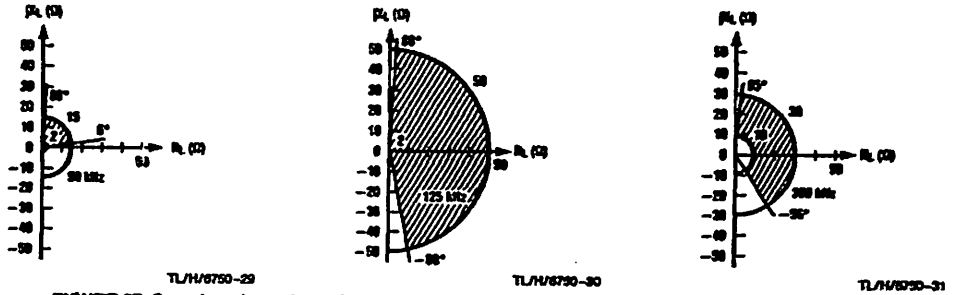
A limited sampling of Z_L was made, during the LM1893 design, of residential and commercial 115V 60 Hz power line. Data was also drawn from the research of Nicholson and Malack (reference 1), among others, to produce *Figures 26* and *27*. All measured impedances are contained within the shaded portions of *Figure 27*. A nominal 3.5, 7.0 and 14 Ω Z_{LN} is used throughout the application information with a nominal 45° phase angle (0° is sometimes used for simplicity).



TL/H/6750–28

FIGURE 26. Measured line impedance range for residential and commercial 115V, 60 Hz lines

Power Line Impedance (Continued)



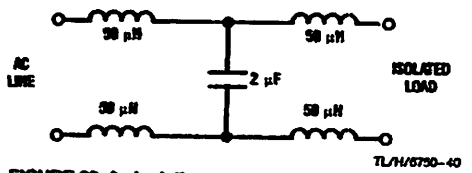
TL/H/5750–29 TL/H/5750–30 TL/H/5750–31

FIGURE 27. Complex-plane plots of measured 115V, 60 Hz line impedance where $Z_L = R_L + jX_L$

Power Line Attenuation

The wiring in most US buildings is a flat 3 conductor cable called Amerflex, BX, or Romex. All referenced line impedances refer to hot-to-neutral impedances with a grounded center conductor. The cable has a 100 Ω characteristic impedance, a 125 kHz quarter-wavelength of 600 m (250 m at 300 kHz), and a measured 7 dB attenuation for a 50 m run with a 10 Ω termination. Generally, line loads may be treated as lumped impedances. Instrument line cords exhibit about 0.7 μH and 30 pF per meter.

Limited tests of CCT link range using this chip show extensive coverage while remaining on one phase of a distribution transformer (100's of m), with link failure often occuring across transformer phases or through transformers unless coupling networks are utilized. Total line attenuation allowed from full signal to limiting sensitivity is more than 70 dB. Typically, signal is coupled across transformer phases by parasitic winding capacitance, typically giving 40 dB attenuation between phased 115 V windings. Coupling capacitors may be installed for improved link operation across phases. Power factor correcting capacitor banks on industrial lines or filter capacitors across the power lines of some electronic gear short carrier signal and should be isolated with inductors. Increasing range is sometimes accomplished by electing to install the isolating inductors (*Figure 28*) and coupling capacitors, as well as by electing to use the boost option. Frequency translating or time division multiplexed repeaters will also increase range.



TL/H/5750–40

FIGURE 28. An isolation network to prevent: 1) noise from some device from polluting the AC line, and 2) to stop some low impedance device (measured at F_O) from shorting carrier signal. Component values given as an example for F_O = 125 kHz on residential power lines

The Coupling Transformer

The design arrived at for T_1 is the result of an unhappy compromise - but a workable one. The goals of 1) building T_1 with a stable resonant frequency, F_O, that is little affected by the de-tuning effect of the line impedance Z_L, and of 2) building a tightly line-coupled transformer for transmitted carrier with loose coupling for transients, are somewhat mutually exclusive. The tradeoffs are exposed in the following example for the CCT designer attempting a new boost-capable, or different core, transformer design.

The compromises are eased by separating the TX output and RX input in the LM2893. An untuned TX coupling transformer with only core coupling (not air-coupled solenoid windings) would employ a high permeability, high magnetic field, low loss, square saturating, toroidal core. The resonant RX path would be isolated from line-pull problems by a unilateral amplifier that operates at line voltages with much more than 110 dB of dynamic range, or by a capacitively coupled pulse transformer driving a unilateral amplifier and filter, for increased selectivity. See the LM2893-specific applications section.

For a LM1893-style transformer application, first, choose the turns ratio N based on an estimated lowest Z_L likely encountered, Z_{LN}. *Figure 29* shows graphically how N affects line signal. N should be as large as possible to drive Z_{LN} with full signal. If T_1 has an unloaded Q, Q_U, of well less than 35, a guess of N somewhat high should be used and later checked for accuracy. The recommended transformers have secondary taps giving a choice of N = 7.07, 10, and 14.1 (nominally) for driving Z_{LN}'s of 14, 7.0, and 3.5 Ω respectively (at T_J = 25°C, V+ = 18V, and Q_U = 35).

The resonating inductance of the tuned primary, L_1, is sought. Note that, while standard transformer design gives a transformer self-inductance with an impedance at operating frequency well above load impedance, the tuned transformer requires a low L_1 for adequate Q_U and minimum line pull. Result: relatively poor mutual coupling.

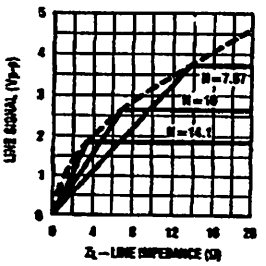
$$L_1 = \frac{R}{2\pi F_O Q} \qquad (3)$$

It is known that resonant frequency $F_O = F_O$ and some minimum bandwidth, or maximum Q, will be required to pass signal under full load conditions.

$$L_1 = \frac{R_O \,\|\, |Z_{LN}|'}{2\pi F_O Q_L} \qquad (4)$$

$|Z_{LN}|'$ is the reflected Z_{LN}, Q_L is the loaded Q, and parallel resistance R_O models all transformer losses and sets Q_O. $R_O \,\|\, |Z_{LN}|'$ is found knowing that it absorbs full rated power.

The Coupling Transformer (Continued)



TL/H/6750–32

FIGURE 29. Impressed line voltage for a given Z_L for each of the 3 taps available on the recommended transformers

$$P_O = I_O V_O = \frac{I_{OPP}}{2\sqrt{2}}\left[\frac{2(-V_{ALC}+V_+)}{2\sqrt{2}}\right] = \frac{(-4.7+V_+)I_O}{4} \quad (5)$$

where I_O is in amps peak-to-peak at an elevated T_J

$$P_O = \frac{(18 - 4.7)\,0.06}{4} = 0.200\ W \quad (6)$$

$$R_O \| |Z_{LN}|' = \frac{V_O^2}{P_O} = \frac{(-V_{ALC} + V_+)\sqrt{2}}{I_O} = 442\ \Omega \quad (7)$$

R_O is found using Z_{LN} and the value for N found when assuming $Q_U = 35$.

$$|Z_{LN}|' = N^2 Z_{LN} = (7.07)^2\,13.9 = 695\ \Omega \quad (8)$$

$$R_O = \frac{1}{\frac{1}{R_O \| |Z_{LN}|'} - \frac{1}{|Z_{LN}|'}} = \frac{1}{\frac{1}{442} - \frac{1}{695}} = 1210\ \Omega \quad (9)$$

$$R_{OS} = \frac{R_O}{1 + Q_U^2} = \frac{1210}{1 + 35^2} = 1\ \Omega \quad (10)$$

Only Q_L remains to be found to calculate L_1. Q_L is related to the −3 dB (half-power) bandwidth by

$$Q_L = \frac{1}{BW\ (\%\ of\ F_O)} \quad (11)$$

An iterative solution is forced where line pull, ΔF_O, must be guessed to find Q_L and L_1. L_1 is then used to check the line pull guess; a large error requires a new guess. Try a BW of 8.7% - that is 4.4% for deviation, 1% for TC of F_O, and 3.3% for ΔF_O - giving $Q_L = 11.5$.

$$L_1 = \frac{442}{2\pi \times 125\,000 \times 11.5} = 49.0\ \mu H \quad (12)$$

Knowing the core inductance per turn, L, and L_1, the number of turns is found.

$$T_1 = \sqrt{\frac{L_1}{L}} = \sqrt{\frac{49.0\ \mu H}{20\ nH/T}} = 49\ ½\ turns \quad (13)$$

T is normally an integer, but these transformers require so few turns that half-turns are specified, remembering that the remaining ½ turn is completed on the P.C. board and is loosely coupled. The secondary turns are calculated

$$T_2 = \frac{T_1}{N} = \frac{49.5}{7.07} = 7.00 = 7\ turns \quad (15)$$

giving an L_2 of 0.98 μH. Note that the recommended 125 kHz transformer mirrors these specifications. The resonating capacitor is

$$C_Q = \frac{1}{(2\pi F_O)^2 L_1} = 33.1 \times 10^{-9} = 33\ nF \quad (16)$$

Line pull ΔF_O was calculated (reference 3) for a Z_L magnitude of 14Ω and up with any phase angle from −90° to 90°. ΔF_O was 6.4% - well above the 3.3% estimate. Referring to (11), an 11.8% bandwidth is required, forcing L_1 to be reduced to reduce Q. That fix was not implemented; some signal attenuation under worst-case drift and ΔF_O is allowed. L_1 is already so small that the 31 gauge winding conducts a ¼ I_{RMS} circulating current.

Line Carrier Detection

While the addition of a carrier detection circuit (for a mute or squelch function) will only decrease receiver ultimate sensitivity, there is sometimes good reason to employ it to free the controller from watching for RX signal when no carrier is incoming, or to employ it to reduce the probability of line collisions (when multiple transmitters operate simultaneously to cause one or more transmissions to fail). Unless the detector is heavily filtered or uses a high carrier amplitude threshold, there will be false outputs that force the controller to have Data Out data checking capability just as is required when using no carrier detector. If false triggering is minimized, the probability of line collisions is increased due to the inability to sense low carrier amplitudes and because of sense delay. The property of the LM1893 to change output state infrequently (although the polarity is undefined) when in the RX mode, with no incoming carrier, reduces the desire to implement carrier detection and preserves the full ultimate sensitivity. Also, many impulse-noise insensitive transmission schemes, like handshaking, are easily modified to recover from line collisions.

Regarding this, it should be stated that for very complicated industrial systems with long signal runs and high line noise levels, it is probably wise to use a protocol which is inherently collision free so that no carrier detect hardware or software is needed. A token passing protocol is an example of such a system.

Figure 30 shows a low cost carrier amplitude detection circuit.

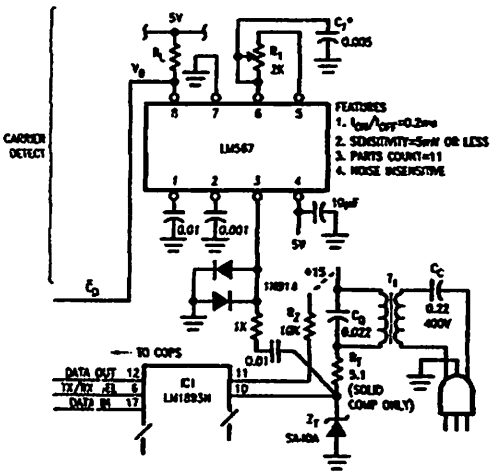
Audio Transmission

The LM1893 is designed to allow analog data transmission and reception. Base-band audio-bandwidth signals FM modulate the carrier passing through the tuned transformer (placing a limit on the usable percent modulation) onto the power line to be linearly demodulated by the receiver PLL. Because the receiver data path beyond the phase detector will pass only digital signal, external audio filtering and amplification is required. *Figure 31* shows a simple audio transmitter and receiver circuit utilizing a carrier detection mute circuit. A single LM339 quad. comparator may be used to build the carrier detect and mute. Filter bandwidth is held to a minimum to minimize noise, especially line-related correlated noise.

Communication and System Protocols

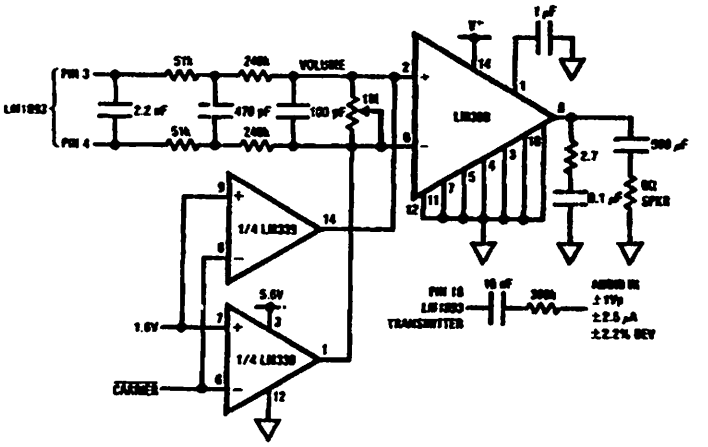
The development of communication and system protocols has historically been the single most time consuming element in design of carrier current systems. The protocols are defined as the following:

1. *Communication protocol*: a software method of encoding and decoding data that remains constant for every transmis-



TL/H/6750–33

FIGURE 30. A simple carrier amplitude detector with output low when carrier is detected



TL/H/6750–34

FIGURE 31. A simple linear analog audio transmitter and receiver are shown. The carrier and 1.6V inputs are derived from the carrier detector of *Figure 30*. The remaining 2 LM339 comparators may be used to build the carrier detector circuit.

Communication and System Protocols (Continued)

sion in a system. Its first purpose is to put data in a baseband digital form that is more easily recognized as a real message at the receive end. Secondly, it incorporates encoding techniques to ensure that noise induced errors do not easily occur; and when they do, they can always be detected. Lastly, the software algorithms that are used on the receive end to decode incoming data prevent the reception of noise induced "phantom" messages, and insure the recovery of real messages from an incoming bit stream that has been altered by noise.

2. *System protocol:* the manner in which messages are coordinated between nodes in a system. Its first purpose is to ensure message retransmission to correct errors (handshake). Secondly it coordinates messages for maximum utilization and efficiency on the network. Lastly, it ensures that messages do not collide on the network. Common system protocols include master-slave, carrier detect multiple access, and token passing. Token passing and master slave have been found to be the most useful since they are inherently collision free.

Both protocols usually reside as software in a single microcontroller that is connected to the LM1893/2893 I/O. In any case, some sort of intelligence is needed to process incoming and outgoing messages. *UARTs have no usefulness in*

Communication and System Protocols (Continued)

carrier current applications since they do not have the intelligence needed to distinguish between real messages and noise induced phantoms.

The difficulty in designing special protocols arises out of the special nature of the AC line, an environment laden with the worst imaginable noise conditions. The relatively low data rates possible over the AC line (typically less than 9600 baud) make it even more imperative that systems utilize the most sophisticated means available to ensure network efficiency.

With these facts in mind, the designer is referred to a publication intended to aid in the development of carrier current systems. This is literature #570075 The Bi-Line Carrier Current Networking System, a 200 pp. book that functions as the "bible" of Bi-Line system design. It has sections on LM1893 circuit optimization, protocol design, evaluation kit usage, critical component selection, and the Datachecker/DTS case study.

Basic Data Encoding (please refer to the previously mentioned publications for advanced techniques)

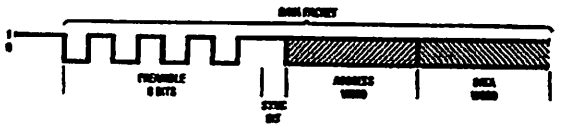
At the beginning of a received transmission, the first 0 to 2 bits may be lost while the chip's receiver settles to the DC bias point required for the given transmitter/receiver pair carrier frequency offset. With proper data encoding, dropped start bits can be tolerated and correct communication can take place. One simple data encoding scheme is now discussed.

Generally, a CCT system consists of many transceivers that normally listen to the line at all times (or during predetermined time windows), waiting for a transmission that directs one or more of the receivers to operate. If any receiver finds its address in the transmitted data packet, further action such as handshaking with the transmitter is initiated. The receiver might tell the transmitter, via retransmission, that it received this data, waiting for acknowledgement before acting on the received command. Error detecting and correcting codes may be employed throughout. The transmitter must have the capability to retransmit after a time if no response from the receiver is heard - under the assumption that the receiver didn't detect its address because of noise, or that the response was missed because of noise or a line collision. (A line collision happens when more than 1 transmitter operates at one time - causing one or more of the communications to fail). After many re-transmissions the transmitter might choose to give up. Collision recovery is achieved by waiting some variable amount of time before retransmission, using a random number of bits delay or a delay based on each transmitter's address, since each transceiver has a unique address.

An example of a simple transmission data packet is shown in *Figure 32*. The 8 bit 50% duty-cycle preamble is long enough to allow receiver biasing with enough bits left over to allow the receiver controller to detect the square-wave that signals the start of a transmission. If there had been no transmission for some time, the receiver would simply need to note that a data transition had occurred and begin its watch for a square-wave. If the receive controller detected the alternating-polarity data square-wave it would then use the sync. bit to signal that the address and data were immediately following. The address data would then be loaded, assuming the fixed format, and tested against its own. If the address was correct, the receiver would then load and store the data. If the address was not correct, either the transmission was not meant for this receiver or noise has fooled the receiver. In the former case, when the transmission was not meant for the receiver, the controller should immediately return to watching the incoming data for its address. If the later case were true, then the receive controller would continue to detect edges, tieing itself up by loading false data and being forced to handshake. The square-wave detection and address load and check routines should be fast to minimize the time spent in loops after being false-triggered by noise. If the controller detects an error (a received data bit that does not conform to the pre-defined encoding format) it should immediately resume watching the LM1893's Data Out for transmissions, the next bit would be shifted in and the process repeated.

A fine-synchronous CCT system passing 3 bits per half-cycle may replace the long 8 bit preamble and sync pulse with a 2 bit start-of-transmission bias preamble. The receive controller might then assume that preamble always starts after bit 1 (the first bit after zero-crossing) so that any data transition at a zero crossing must be the start of the address bits and is tested as such. The fine synchronous receiver operates with a simpler controller than an asynchronous system.

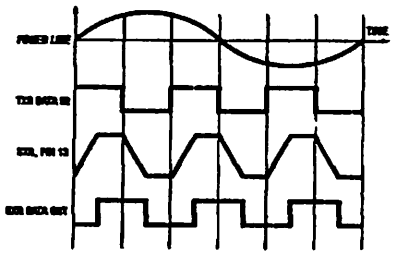
Discussion has assumed that the controller has always known when the Data Out is high or low. The controller must sample at the proper time to check the Data Out state. Since noise shows itself as pulse width jitter, symmetrically placed about the no-noise switch-points, optimum Data Out sampling is done in the center of the received data pulse. The receive data path has a time delay that, at low data rates, is dominated by the impulse noise filter integrator and is nominally ½ bit. At a 2 kHz data rate, an additional delay of approximately $^{1}/_{16}$ bit is added because of the cumulative delay of the remainder of the receiver. *Figure 33* shows that Data Out sampling occurs conveniently at the transmitted



TL/H/8950–35

FIGURE 32. A simple encoded data packet, generated by the transmit controller is shown. The horizontal axis is time where 1 bit time is $1/(2f_{DATA})$

Basic Data Encoding (Continued)



TL/H/5750–36

FIGURE 33. Operating waveforms of a line-synchronized transceiver pair are shown. The diagram shows how the transmitted data transitions may be used as received data sampling points

data edges for the line synchronous data transmission scheme mentioned in the previous paragraph. With the asynchronous system suggested, the receive controller must sample the Data Out pin often to determine, with several bits of accuracy, where the square-wave data transitions take place, average their positions assuming a known data rate, and calculate where the center of the data bits are and will continue to be as the address and data are read. A long preamble is helpful. Software that continuously updates the center-of-bit time estimate, as address and data are received, works even better. Alternatively, a coding scheme employing an embedded clock can be used.

LM2893 Application Hints

The LM2893 is intended for advanced applications where special circuitry is used in the transmit and receive paths. The LM2893 makes this possible by featuring separate transmit output and receive input pins.

Examples of enhancements that can be added to the basic LM1893/2893 circuit include separate transmit and receive windings on the coupling transformer, high quality ceramic or LC filters in the receive path, and simple impulse noise blanking circuits.

In many applications, the additional performance to be gained outweighs the extra cost of the additional circuitry. More than likely, high performance industrial applications such as building energy management will fit into this category, since they require the utmost in reliability.

Because of the specialized nature of individual LM2893 applications, it is not possible to give one circuit that will satisfy all requirements for performance and cost effectiveness. Therefore no specific application examples will be given. Instead the subsequent text describes in general terms the types of circuits that can be used to increase performance along with their advantages and disadvantages. It is intended to be a springboard for ideas.

LM2893 COUPLING NETWORKS

The main disadvantages of the typical LM1893 coupling network are that it functions as the bandpass filter, has loose coupling between primary and secondary, and has a single secondary. The LM1893 coupling network was designed this way mainly because of the restraint that the carrier input and output are tied together.

Because the coupling transformer is used as a filter, the LM1893 circuit is susceptible to pulling of the center frequency under conditions of changing line impedances or when several LM1893 circuits are close in proximity on the AC line. Because the tuned transformer has a high value of "Q", ringing also occurs in the presence of impulsive noise. This ringing occurs at the center frequency and increases the error rate of transmissions, especially at relatively high data rates (> 2000 baud). Because it is the only tuned circuit in the system, the selectivity characteristics leave a lot to be desired.

The LM2893, having separate receive input and transmit output pins, removes the limitations on coupling transformer design, allowing the design of circuits devoid of the previous limitations.

The first enhancement that can be made with the LM2893 circuit is the use of a high permeability ferrite toroid for line coupling along with a separate filter. The transformer would be of broadband design (untuned) with two secondaries, one for coupling to the transmit output and one for coupling to the receive input. This allows impedance matching of both the transmitter and receiver, with the result of quite a bit more receive sensitivity.

Because of the increased signal and separate receive signal path, a 3 or 6 db pad can be used before the selective stages to eliminate pulling of the center frequency due to changes in line impedance.

Another advantage of the toroidal transformer is that it can be designed for use at very low line impedances due to its inherent tight coupling.

SEPARATE FILTER

Because of the separate receive path of the LM2893, a relatively high quality bandpass filter can be used for selectivity. Inexpensive ceramic filters are available that have bandpass and center frequency characteristics compatible with carrier current operation. Furthermore, the use of these filters allows multichannel operation, previously made difficult by the single tuned network of the LM1893. These filters are easily cascaded for even more off-frequency rejection. If the pad is added before the filter, there will be negligible pulling due to changes in line impedance reflected through the coupling transformer.

Alternatively, a Butterworth/Chebyshev bandpass LC filter or an active filter can be used in place of the ceramic filter.

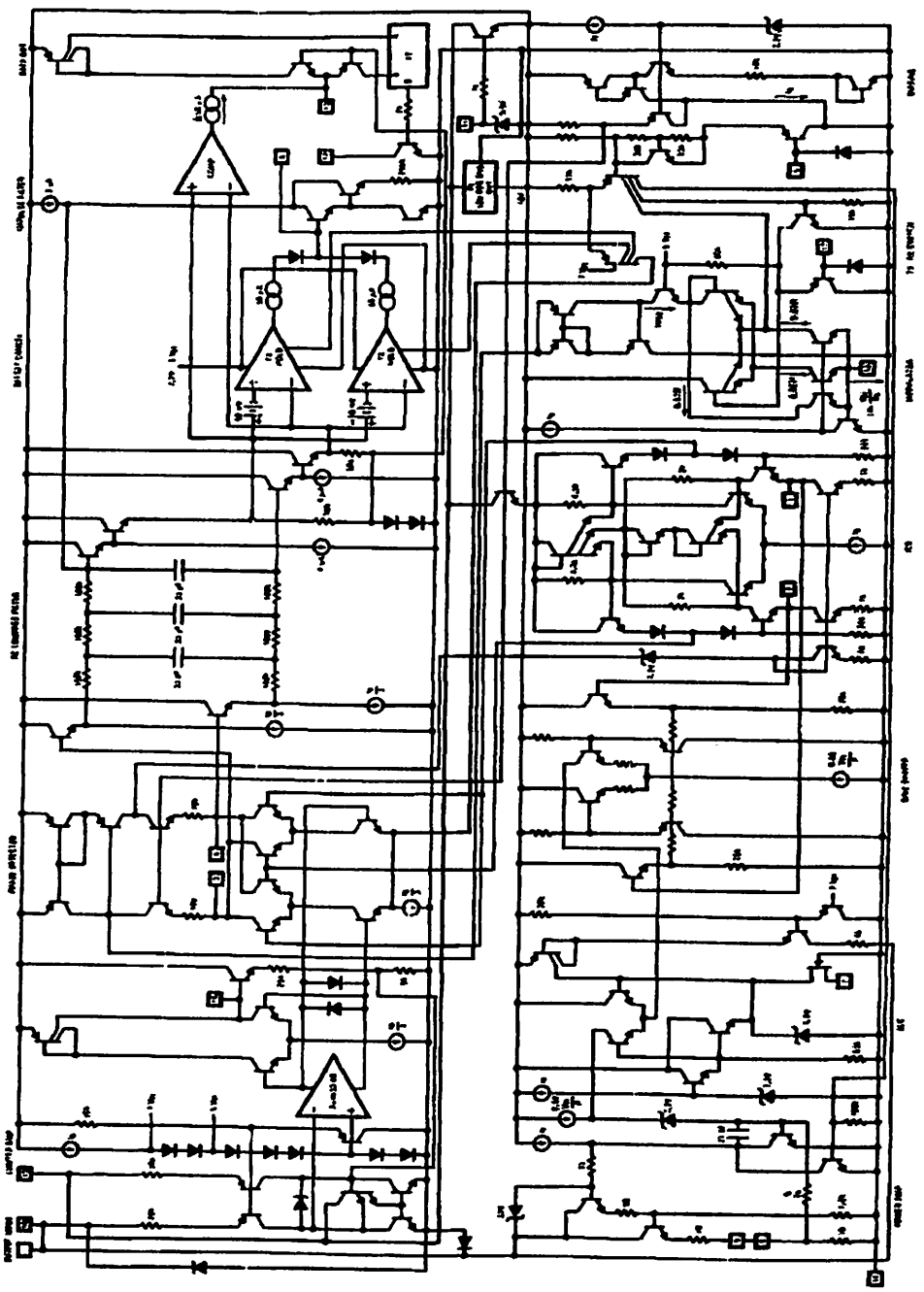
IMPULSE NOISE BLANKER

Although the LM2893 has adequate impulse noise rejection for most applications, there is reason to employ impulse blanking to improve error rates in severe AC line environments. Typically, errors occur due to pulse jitter in the LM1893/2893 data output that originates when the internal time domain filter smooths out an incoming noise pulse.

The solution involves removing the impulse completely and not simply trying to filter it. Moreover, the pulse should be removed in the receive signal path before the selective portions of the circuit to eliminate ringing. This also allows the receiver filter to smooth out the blanks that also occur in the desired incoming carrier signal.

If a carrier detect circuit is desired in conjunction with the LM2893 it can be located after the filter and impulse blanker. Because impulse noise is removed, the false triggering that plagues these circuits will be greatly reduced.

Simplified Schematic

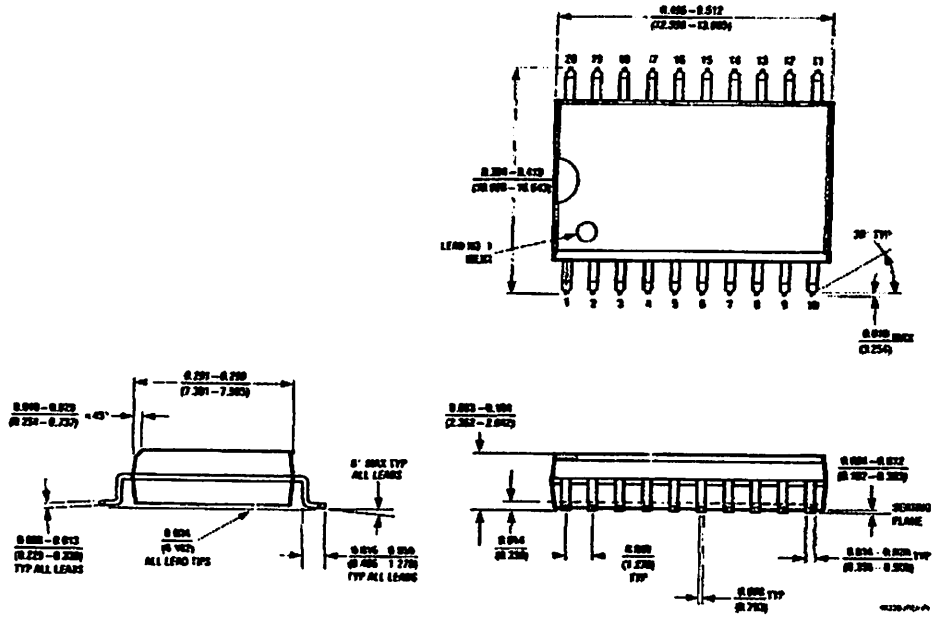


TL/H/6760–37

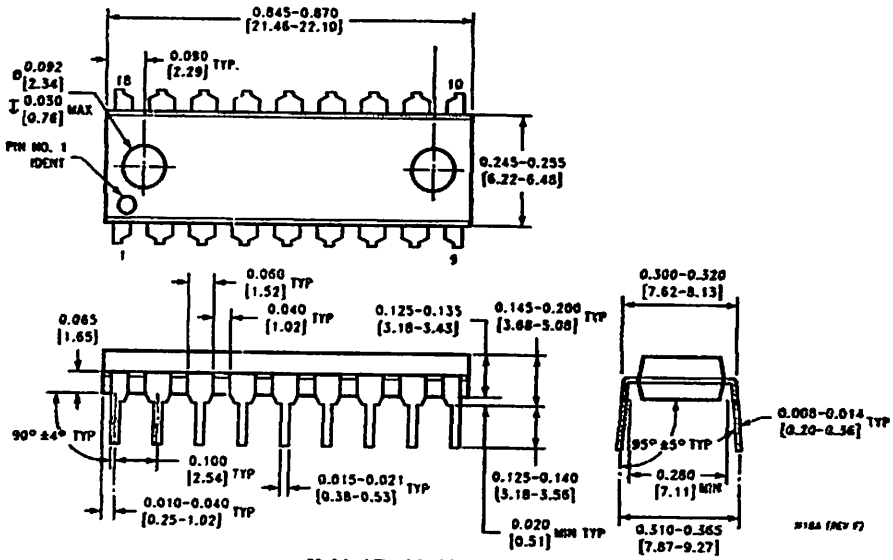
References

1. Nicholson, J.R. and J.A. Malack; "RF Impedance of Power Lines and Line Impedance Stabilization Network in Conducted Interference Measurements;" IEEE Transactions on Electromagnetic Compatibility; May 1973; (line impedance data)
2. Southwick, R.A.; "Impedance Characteristics of Single-Phase Power Lines;" Conference Rec.; 1973 IEEE Int. Symp. on Electromagnetic Compatibility; (line impedance data)
3. Hayt, William H. Jr. and Jack E. Kemmerly; "Engineering Circuit Analysis;" McGraw-Hill Books; 1971; pp. 447–453; (linear transformer reflected impedance)
4. FCC, "Notice of Proposed Rule Making," Docket 20780, adopted Apr. 14, 1976, (Proposed regulation)
5. Monticelli, Dennis M. and Michael E. Wright; "A Carrier Current Transceiver IC for Data Transmission Over the AC Power Lines;" IEEE J. Solid-State Circuits; vol. SC-17; Dec. 1982; pp. 1158-1165; (LM1893 circuit description)
6. Lee, Mitchell; "A New Carrier Current Transceiver IC;" IEEE Trans. on Consumer Electronics; vol. CE-28; Aug. 1982; pp. 409–414; (Application of LM1893)

Physical Dimensions inches (millimeters)



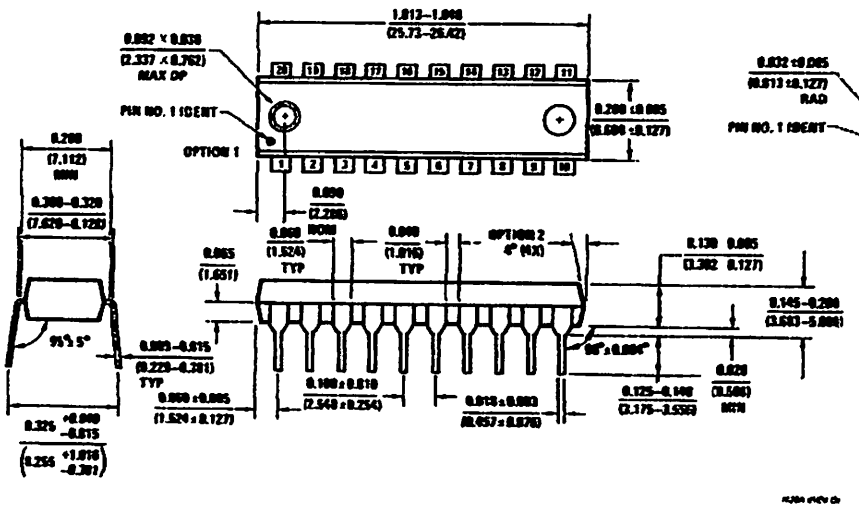
Molded Small Outline Package (M)
Order Part LM2893M
NS Package Number M20B



Molded Dual-In-Line Package (N)
Order Part LM1893N
NS Package Number N18A

Physical Dimensions inches (millimeters) (Continued)

Lit. # 107664



Molded Dual-In-Line Package (N)
Order Part LM2893N
NS Package Number N20A

LIFE SUPPORT POLICY

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

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italia to Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

Features

- High-performance, Low-power AVR® 8-bit Microcontroller
- RISC Architecture
  - 130 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 8K Bytes of In-System Self-programmable Flash
    Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - 512 Bytes EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - 512 Bytes Internal SRAM
  - Up to 64K Bytes Optional External Memory Space
  - Programming Lock for Software Security
- Peripheral Features
  - One 8-bit Timer/Counter with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Three PWM Channels
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Three Sleep Modes: Idle, Power-down and Standby
- I/O and Packages
  - 35 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad QFN/MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega8515L
  - 4.5 - 5.5V for ATmega8515
- Speed Grades
  - 0 - 8 MHz for ATmega8515L
  - 0 - 16 MHz for ATmega8515

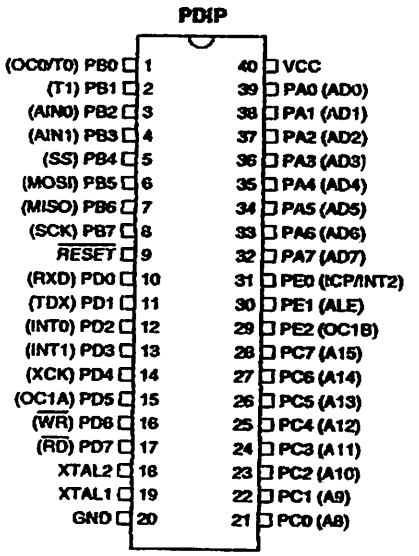
8-bit AVR® Microcontroller with 8K Bytes In-System Programmable Flash

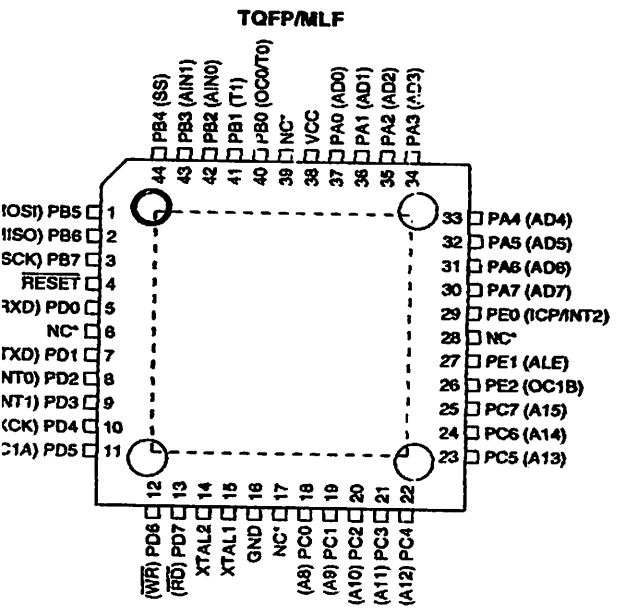
ATmega8515
ATmega8515L

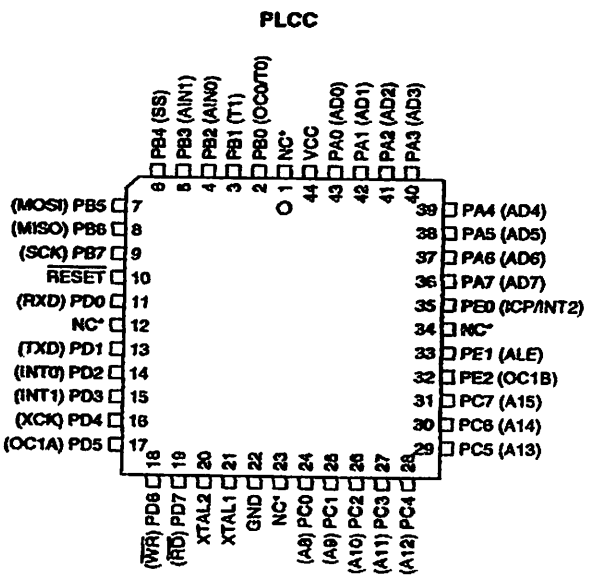
2512J–AVR–10/06

Pin Configurations

Figure 1. Pinout ATmega8515







NOTES:

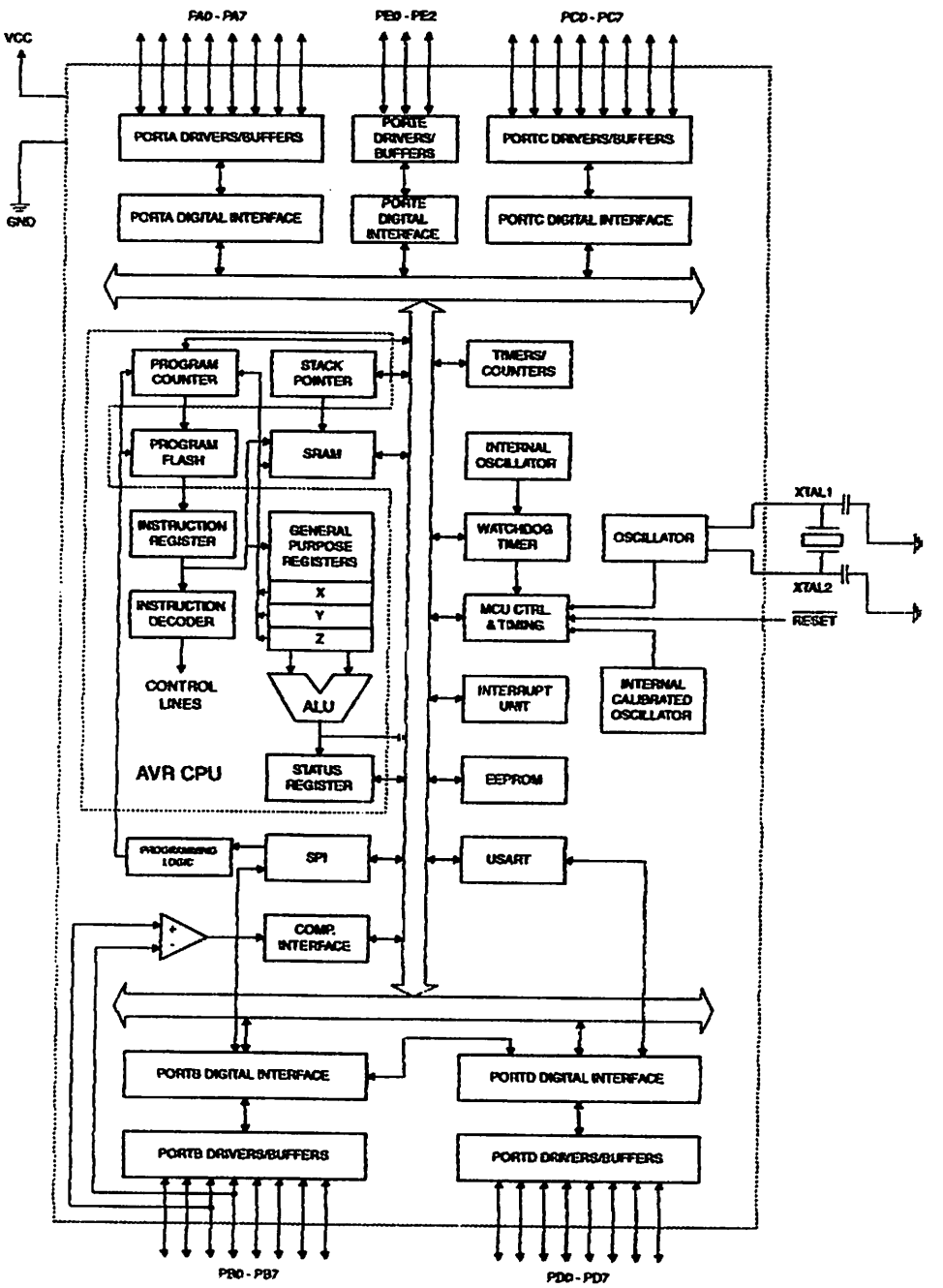
1. *MLF bottom pad should be soldered to ground.*
2. * NC = Do not connect (May be used in future devices)

verview

The ATmega8515 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega8515 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

lock Diagram

Figure 2. Block Diagram



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

The ATmega8515 provides the following features: 8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes EEPROM, 512 bytes SRAM, an External memory interface, 35 general purpose I/O lines, 32 general purpose working registers, two flexible Timer/Counters with compare modes, Internal and External interrupts, a Serial Programmable USART, a programmable Watchdog Timer with internal Oscillator, a SPI serial port, and three software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and Interrupt system to continue functioning. The Power-down mode saves the Register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the Program memory to be reprogrammed In-System through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-programmable Flash on a monolithic chip, the Atmel ATmega8515 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega8515 is supported with a full suite of program and system development tools including: C Compilers, Macro assemblers, Program debugger/simulators, In-circuit Emulators, and Evaluation kits.

sclaimer

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

90S4414/8515 and mega8515 mpatibility

The ATmega8515 provides all the features of the AT90S4414/8515. In addition, several new features are added. The ATmega8515 is backward compatible with AT90S4414/8515 in most cases. However, some incompatibilities between the two microcontrollers exist. To solve this problem, an AT90S4414/8515 compatibility mode can be selected by programming the S8515C Fuse. ATmega8515 is 100% pin compatible with AT90S4414/8515, and can replace the AT90S4414/8515 on current printed circuit boards. However, the location of Fuse bits and the electrical characteristics differs between the two devices.

0S4414/8515 Compatibility de

Programming the S8515C Fuse will change the following functionality:

- The timed sequence for changing the Watchdog Time-out period is disabled. See "Timed Sequences for Changing the Configuration of the Watchdog Timer" on page 53 for details.
- The double buffering of the USART Receive Registers is disabled. See "AVR USART vs. AVR UART – Compatibility" on page 137 for details.
- PORTE(2:1) will be set as output, and PORTE0 will be set as input.

in Descriptions

:C	Digital supply voltage.
ND	Ground.
rt A (PA7..PA0)	Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port A also serves the functions of various special features of the ATmega8515 as listed on page 67.
rt B (PB7..PB0)	Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port B also serves the functions of various special features of the ATmega8515 as listed on page 67.
rt C (PC7..PC0)	Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.
rt D (PD7..PD0)	Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port D also serves the functions of various special features of the ATmega8515 as listed on page 72.
rt E(PE2..PE0)	Port E is an 3-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port E also serves the functions of various special features of the ATmega8515 as listed on page 74.
SET	Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 18 on page 46. Shorter pulses are not guaranteed to generate a reset.
LL1	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
LL2	Output from the inverting Oscillator amplifier.

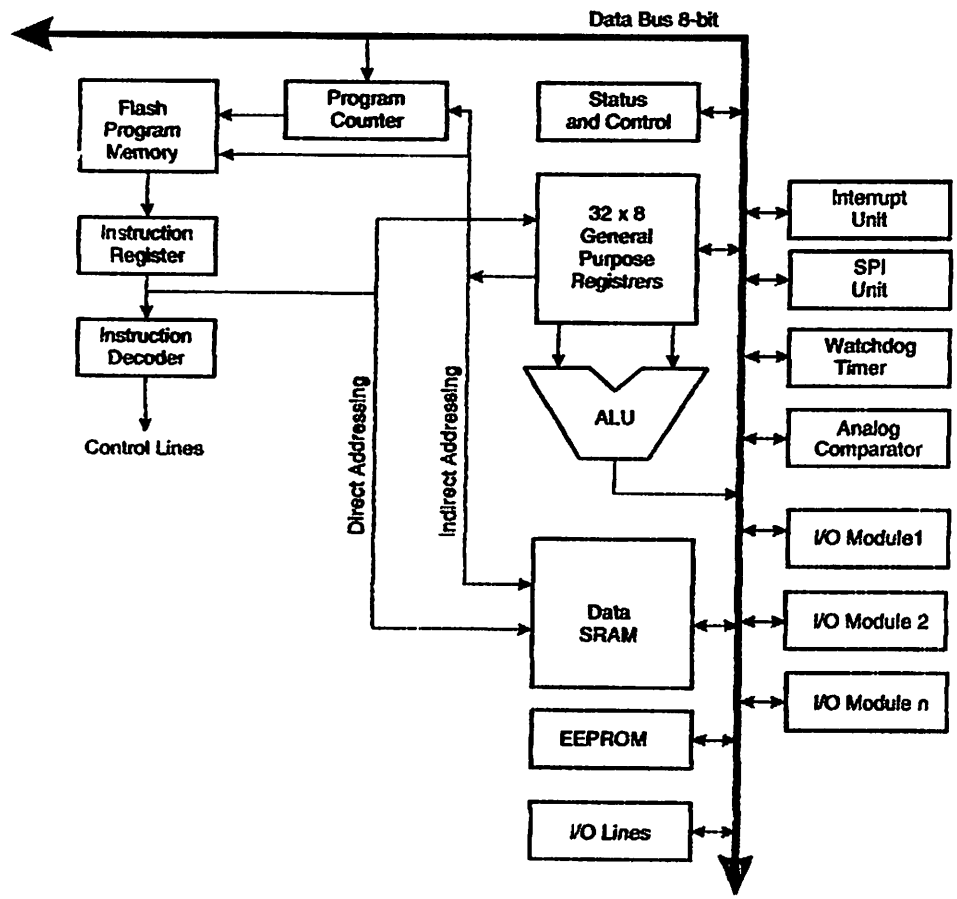
AVR CPU Core

Introduction

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

Architectural Overview

Figure 3. Block Diagram of the AVR Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the Program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the Program memory. This concept enables instructions to be executed in every clock cycle. The Program memory is In-System re programmable Flash memory.

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

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

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

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

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

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

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

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

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, $20 - $5F.

.U – Arithmetic Logic it

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

tatus Register

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

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

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

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

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

- **Bit 6 – T: Bit Copy Storage**

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

- **Bit 5 – H: Half Carry Flag**

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

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

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

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

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

- **Bit 2 – N: Negative Flag**

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

- **Bit 1 – Z: Zero Flag**

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

- **Bit 0 – C: Carry Flag**

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

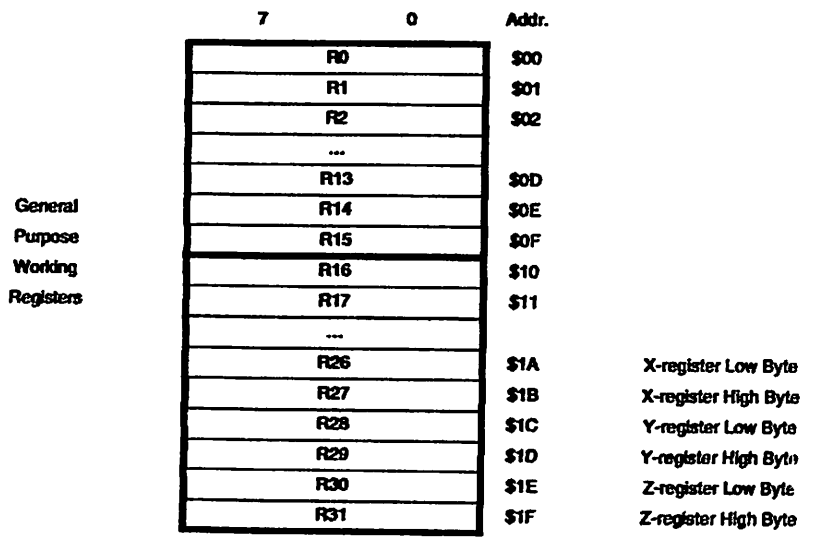
eneral Purpose egister File

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

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

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

Figure 4. AVR CPU General Purpose Working Registers



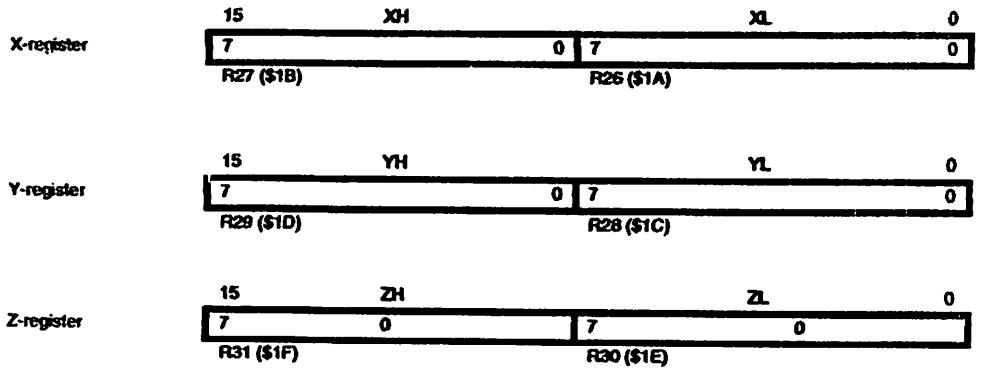
Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions.

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

he X-register, Y-register, and -register

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

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



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

ack Pointer

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

The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer must be set to point above $60. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when the return address is pushed onto the Stack with subroutine call or interrupt. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when address is popped from the Stack with return from subroutine RET or return from interrupt RETI.

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

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

istruction Execution iming

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

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

Figure 6. The Parallel Instruction Fetches and Instruction Executions

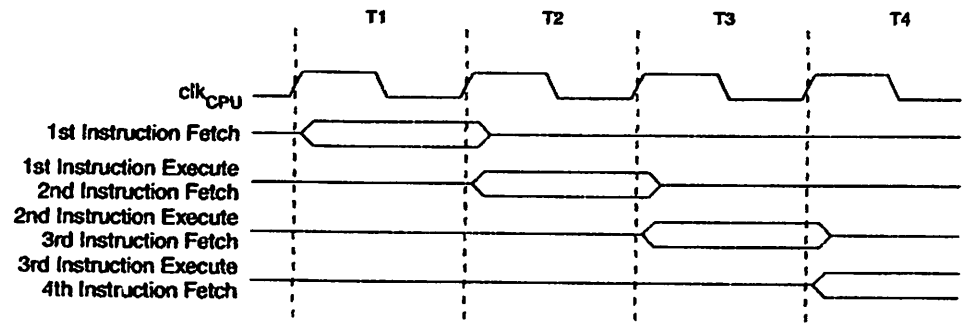
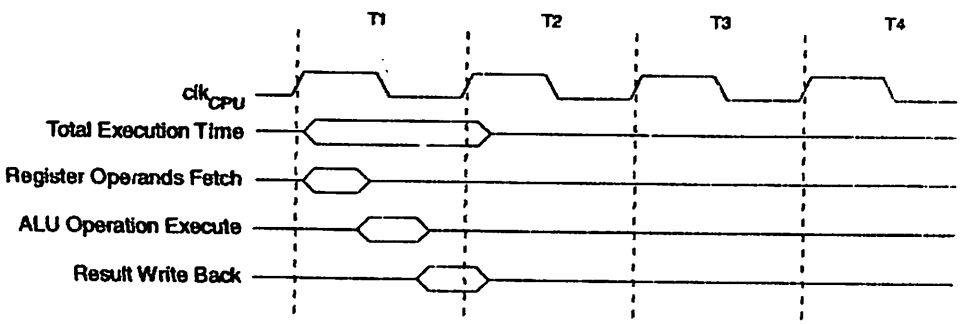


Figure 7 shows the internal timing concept for the Register File. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 7. Single Cycle ALU Operation



set and Interrupt ndling

The AVR provides several different interrupt sources. These interrupts and the separate Reset Vector each have a separate program vector in the Program memory space. All interrupts are assigned individual enable bits which must be written logic one together with the Global Interrupt Enable bit in the Status Register in order to enable the interrupt. Depending on the Program Counter value, interrupts may be automatically disabled when Boot Lock bits BLB02 or BLB12 are programmed. This feature improves software security. See the section "Memory Programming" on page 179 for details.

The lowest addresses in the Program memory space are by default defined as the Reset and Interrupt Vectors. The complete list of vectors is shown in "Interrupts" on page 54. The list also determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INT0 – the External Interrupt Request 0. The Interrupt Vectors can be moved to the start of the Boot Flash section by setting the IVSEL bit in the General Interrupt Control Register (GICR). Refer to "Interrupts" on page 54 for more information. The Reset Vector can

also be moved to the start of the Boot Flash section by programming the BOOTRST Fuse, see "Boot Loader Support – Read-While-Write Self-Programming" on page 166.

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

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

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

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

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

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

Assembly Code Example

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

C Code Example

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

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

Assembly Code Example

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

C Code Example

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

errupt Response Time

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

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

VR ATmega8515 lemories

This section describes the different memories in the ATmega8515. The AVR architecture has two main memory spaces, the Data Memory and the Program memory space. In addition, the ATmega8515 features an EEPROM Memory for data storage. All three memory spaces are linear and regular.

-System eprogrammable Flash rogram memory

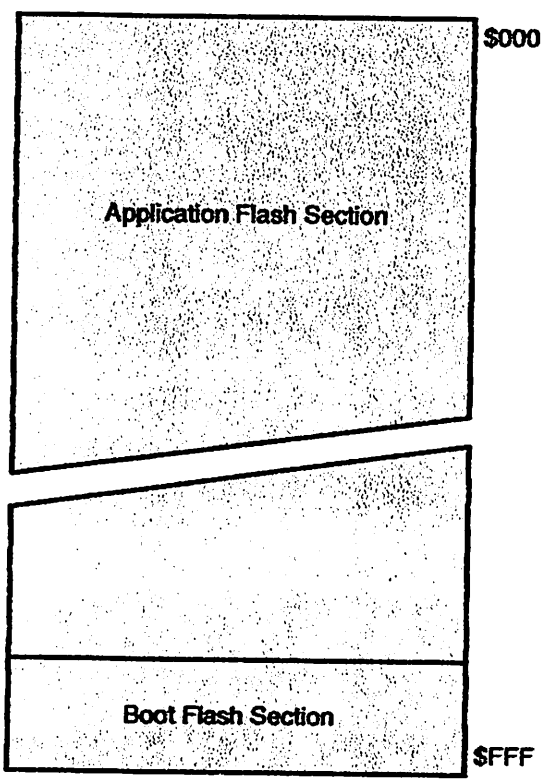
The ATmega8515 contains 8K bytes On-chip In-System Reprogrammable Flash memory for program storage. Since all AVR instructions are 16 or 32 bits wide, the Flash is organized as 4K x 16. For software security, the Flash Program memory space is divided into two sections, Boot Program section and Application Program section.

The Flash memory has an endurance of at least 10,000 write/erase cycles. The ATmega8515 Program Counter (PC) is 12 bits wide, thus addressing the 4K Program memory locations. The operation of Boot Program section and associated Boot Lock bits for software protection are described in detail in "Boot Loader Support – Read-While-Write Self-Programming" on page 166. "Memory Programming" on page 179 contains a detailed description on Flash data serial downloading using the SPI pins.

Constant tables can be allocated within the entire Program memory address space, see the LPM – Load Program memory instruction description.

Timing diagrams for instruction fetch and execution are presented in "Instruction Execution Timing" on page 13.

Figure 8. Program memory Map



RAM Data Memory

Figure 9 shows how the ATmega8515 SRAM Memory is organized.

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

An optional external data SRAM can be used with the ATmega8515. This SRAM will occupy an area in the remaining address locations in the 64K address space. This area starts at the address following the internal SRAM. The Register File, I/O, Extended I/O and Internal SRAM occupies the lowest 608 bytes in normal mode, so when using 64KB (65536 bytes) of External Memory, 64928 Bytes of External Memory are available. See "External Memory Interface" on page 25 for details on how to take advantage of the external memory map.

When the addresses accessing the SRAM memory space exceeds the internal Data memory locations, the external data SRAM is accessed using the same instructions as for the internal Data memory access. When the internal data memories are accessed, the read and write strobe pins (PD7 and PD6) are inactive during the whole access cycle. External SRAM operation is enabled by setting the SRE bit in the MCUCR Register.

Accessing external SRAM takes one additional clock cycle per byte compared to access of the internal SRAM. This means that the commands LD, ST, LDS, STS, LDD, STD, PUSH, and POP take one additional clock cycle. If the Stack is placed in external SRAM, interrupts, subroutine calls and returns take three clock cycles extra because the two-byte Program Counter is pushed and popped, and external memory access does not take advantage of the internal pipe-line memory access. When external SRAM interface is used with wait-state, one-byte external access takes two, three, or four additional clock cycles for one, two, and three wait-states respectively. Interrupts, subroutine calls and returns will need five, seven, or nine clock cycles more than specified in the instruction set manual for one, two, and three wait-states.

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

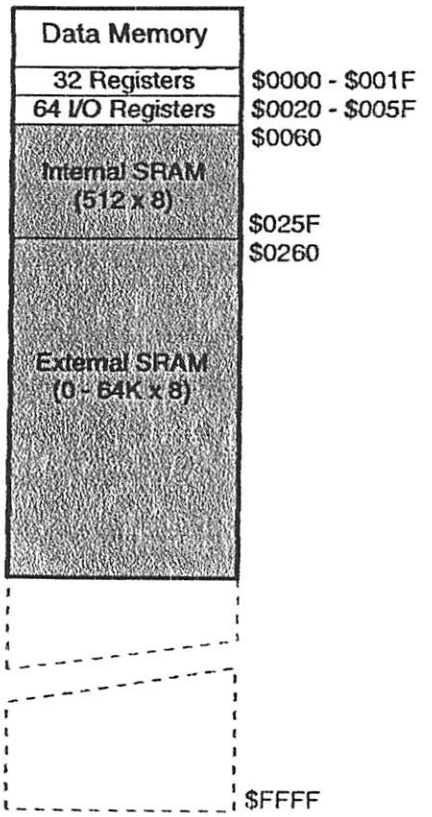
The direct addressing reaches the entire data space.

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

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

The 32 general purpose working registers, 64 I/O Registers, and the 512 bytes of internal data SRAM in the ATmega8515 are all accessible through all these addressing modes. The Register File is described in "General Purpose Register File" on page 11.

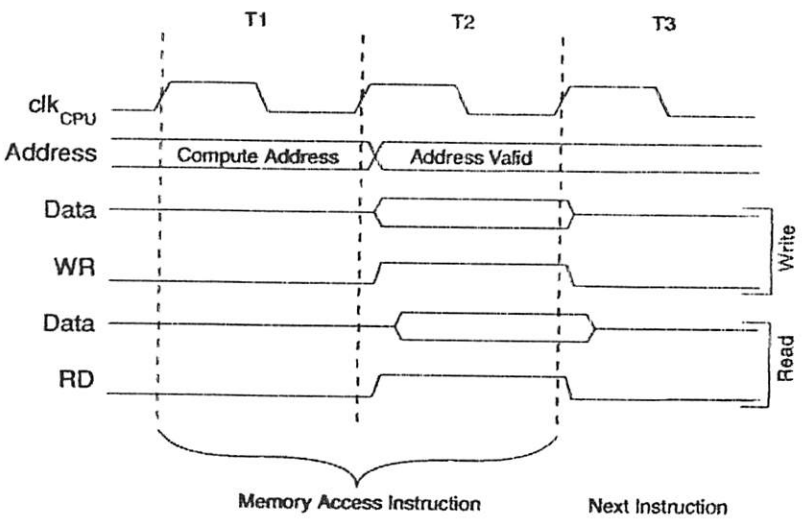
Figure 9. Data Memory Map



ta Memory Access Times

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

Figure 10. On-chip Data SRAM Access Cycles



EPROM Data Memory

The ATmega8515 contains 512 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described in the following, specifying the EEPROM Address Registers, the EEPROM Data Register, and the EEPROM Control Register.

"Memory Programming" on page 179 contains a detailed description on EEPROM Programming in SPI or Parallel Programming mode.

EPROM Read/Write Access

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

The write access time for the EEPROM is given in Table 1. A self-timing function, however, lets the user software detect when the next byte can be written. If the user code contains instructions that write the EEPROM, some precautions must be taken. In heavily filtered power supplies, V_{CC} is likely to rise or fall slowly on Power-up/down. This causes the device for some period of time to run at a voltage lower than specified as minimum for the clock frequency used. See "Preventing EEPROM Corruption" on page 24. for details on how to avoid problems in these situations.

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

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

e EEPROM Address gister – EEARH and EEARL

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

- **Bits 15..9 – Res: Reserved Bits**

These bits are reserved bits in the ATmega8515 and will always read as zero.

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

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

e EEPROM Data Register – EDR

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

- **Bits 7..0 – EEDR7.0: EEPROM Data**

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

e EEPROM Control Register ECR

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

- **Bits 7..4 – Res: Reserved Bits**

These bits are reserved bits in the ATmega8515 and will always read as zero.

- **Bit 3 – EERIE: EEPROM Ready Interrupt Enable**

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

- **Bit 2 – EEMWE: EEPROM Master Write Enable**

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

- **Bit 1 – EEWE: EEPROM Write Enable**

The EEPROM Write Enable Signal EEWE is the write strobe to the EEPROM. When address and data are correctly set up, the EEWE bit must be written to one to write the value into the EEPROM. The EEMWE bit must be written to one before a logical one is written to EEWE, otherwise no EEPROM write takes place. The following procedure should be followed when writing the EEPROM (the order of steps 3 and 4 is not essential):

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

The EEPROM can not be programmed during a CPU write to the Flash memory. The software must check that the Flash programming is completed before initiating a new EEPROM write. Step 2 is only relevant if the software contains a Boot Loader allowing the CPU to program the Flash. If the Flash is never being updated by the CPU, step 2 can be omitted. See "Boot Loader Support – Read-While-Write Self-Programming" on page 166 for details about boot programming.

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

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

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

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

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

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

Table 1. EEPROM Programming Time

Symbol	Number of Calibrated RC Oscillator Cycles[1]	Typ Programming Time
EEPROM Write (from CPU)	8448	8.5 ms

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

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

Assembly Code Example

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

C Code Example

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

The next code examples show assembly and C functions for reading the EEPROM. The examples assume that interrupts are controlled so that no interrupts will occur during execution of these functions.

Assembly Code Example

```
EEPROM_read:
  ; Wait for completion of previous write
  sbic EECR,EEWE
  rjmp EEPROM_read
  ; Set up address (r18:r17) in address register
  out EEARH, r18
  out EEARL, r17
  ; Start eeprom read by writing EERE
  sbi EECR,EERE
  ; Read data from data register
  in  r16,EEDR
  ret
```

C Code Example

```
unsigned char EEPROM_read(unsigned int uiAddress)
{
  /* Wait for completion of previous write */
  while(EECR & (1<<EEWE))
    ;
  /* Set up address register */
  EEAR = uiAddress;
  /* Start eeprom read by writing EERE */
  EECR |= (1<<EERE);
  /* Return data from data register */
  return EEDR;
}
```

PROM Write During Power-vn Sleep Mode

When entering Power-down Sleep mode while an EEPROM write operation is active, the EEPROM write operation will continue, and will complete before the Write Access time has passed. However, when the write operation is completed, the crystal Oscillator continues running, and as a consequence, the device does not enter Power-down entirely. It is therefore recommended to verify that the EEPROM write operation is completed before entering Power-down.

eventing EEPROM rruption

During periods of low V_{CC}, the EEPROM data can be corrupted because the supply voltage is too low for the CPU and the EEPROM to operate properly. These issues are the same as for board level systems using EEPROM, and the same design solutions should be applied.

An EEPROM data corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the EEPROM requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage is too low.

EEPROM data corruption can easily be avoided by following this design recommendation:

Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-out Detector (BOD). If the detection level of the internal BOD does not match the needed detection level, an external low V_{CC} Reset Protection circuit can be used. If a Reset occurs while a write operation is in progress, the write operation will be completed provided that the power supply voltage is sufficient.

) Memory

The I/O space definition of the ATmega8515 is shown in "Register Summary" on page 239.

All ATmega8515 I/Os and peripherals are placed in the I/O space. The I/O locations are accessed by the IN and OUT instructions, transferring data between the 32 general purpose working registers and the I/O space. I/O Registers within the address range $00 - $1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the instruction set section for more details. When using the I/O specific commands IN and OUT, the I/O addresses $00 - $3F must be used. When addressing I/O Registers as data space using LD and ST instructions, $20 must be added to these addresses.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers $00 to $1F only.

The I/O and Peripherals Control Registers are explained in later sections.

ɪternal Memory terface

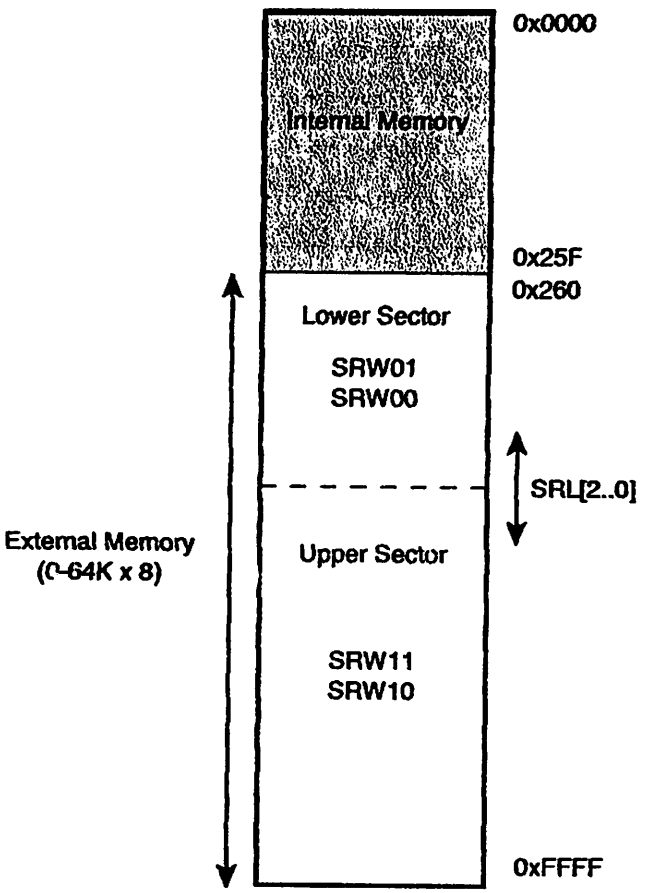
With all the features the External Memory Interface provides, it is well suited to operate as an interface to memory devices such as external SRAM and Flash, and peripherals such as LCD-display, A/D, and D/A. The main features are:

- **Four Different Wait State Settings (Including No wait State)**
- **Independent Wait State Setting for Different External Memory Sectors (Configurable Sector Size)**
- **The Number of Bits Dedicated to Address High Byte is Selectable**
- **Bus Keepers on Data Lines to Minimize Current Consumption (Optional)**

ɛrview

When the eXternal MEMory (XMEM) is enabled, address space outside the internal SRAM becomes available using the dedicated external memory pins (see Figure 1 on page 2, Table 26 on page 66, Table 32 on page 70, and Table 38 on page 74). The memory configuration is shown in Figure 11.

Figure 11. External Memory with Sector Select



ng the External Memory ɪrface

The interface consists of:

- AD7:0: Multiplexed low-order address bus and data bus
- A15:8: High-order address bus (configurable number of bits)
- ALE: Address latch enable
- $\overline{RD}$: Read strobe
- $\overline{WR}$: Write strobe

The control bits for the External Memory Interface are located in three registers, the MCU Control Register – MCUCR, the Extended MCU Control Register – EMCUCR, and the Special Function IO Register – SFIOR.

When the XMEM interface is enabled, it will override the settings in the data direction registers corresponding to the ports dedicated to the interface. For details about this port override, see the alternate functions in section "I/O Ports" on page 59. The XMEM interface will auto-detect whether an access is internal or external. If the access is external, the XMEM interface will output address, data, and the control signals on the ports according to Figure 13 (this figure shows the wave forms without wait states). When ALE goes from high to low, there is a valid address on AD7:0. ALE is low during a data transfer. When the XMEM interface is enabled, also an internal access will cause activity on address-, data-, and ALE ports, but the $\overline{RD}$ and $\overline{WR}$ strobes will not toggle during internal access. When the External Memory Interface is disabled, the normal pin and data direction settings are used. Note that when the XMEM interface is disabled, the address space above the internal SRAM boundary is not mapped into the internal SRAM. Figure 12 illustrates how to connect an external SRAM to the AVR using an octal latch (typically "74x573" or equivalent) which is transparent when G is high.

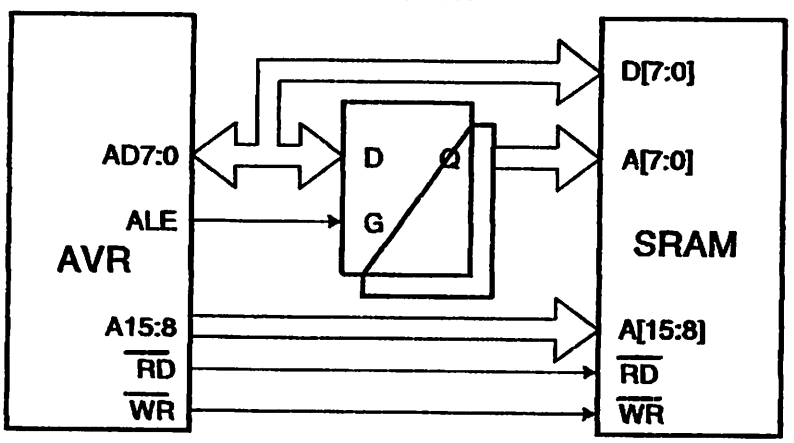
dress Latch Requirements

Due to the high-speed operation of the XRAM interface, the address latch must be selected with care for system frequencies above 8 MHz @ 4V and 4 MHz @ 2.7V. When operating at conditions above these frequencies, the typical old style 74HC series latch becomes inadequate. The external memory interface is designed in compliance to the 74AHC series latch. However, most latches can be used as long they comply with the main timing parameters. The main parameters for the address latch are:

- D to Q propagation delay (t_{pd})
- Data setup time before G low (t_{su})
- Data (address) hold time after G low (t_h)

The external memory interface is designed to guaranty minimum address hold time after G is asserted low of t_h = 5 ns (refer to t_{LAXX_LD}/t_{LLAXX_ST} in Table 98 to Table 105 on page 204). The D to Q propagation delay (t_{pd}) must be taken into consideration when calculating the access time requirement of the external component. The data setup time before G low (t_{su}) must not exceed address valid to ALE low (t_{AVLLC}) minus PCB wiring delay (dependent on the capacitive load).

Figure 12. External SRAM Connected to the AVR



ll-up and Bus Keeper

The pull-up resistors on the AD7:0 ports may be activated if the corresponding Port Register is written to one. To reduce power consumption in sleep mode, it is recommended to disable the pull-ups by writing the Port Register to zero before entering sleep.

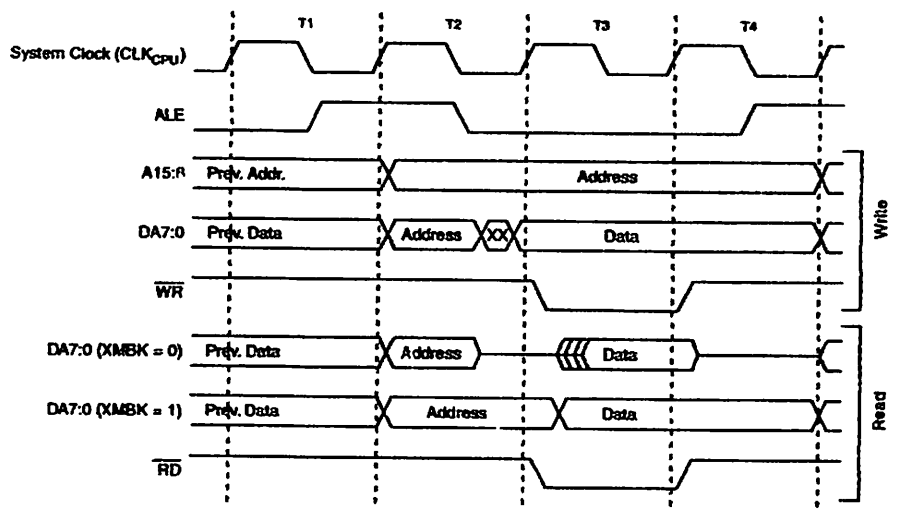
The XMEM interface also provides a bus keeper on the AD7:0 lines. The bus keeper can be disabled and enabled in software as described in "Special Function IO Register – SFIOR" on page 31. When enabled, the bus keeper will keep the previous value on the AD7:0 bus while these lines are tri-stated by the XMEM interface.

ning

External memory devices have various timing requirements. To meet these requirements, the ATmega8515 XMEM interface provides four different wait states as shown in Table 3. It is important to consider the timing specification of the external memory device before selecting the wait state. The most important parameters are the access time for the external memory in conjunction with the set-up requirement of the ATmega8515. The access time for the external memory is defined to be the time from receiving the chip select/address until the data of this address actually is driven on the bus. The access time cannot exceed the time from the ALE pulse is asserted low until data must be stable during a read sequence (t_{LLRL}+ t_{RLRH} - t_{DVRH} in Table 98 to Table 105 on page 204). The different wait states are set up in software. As an additional feature, it is possible to divide the external memory space in two sectors with individual wait state settings. This makes it possible to connect two different memory devices with different timing requirements to the same XMEM interface. For XMEM interface timing details, please refer to Figure 89 to Figure 92, and Table 98 to Table 105.

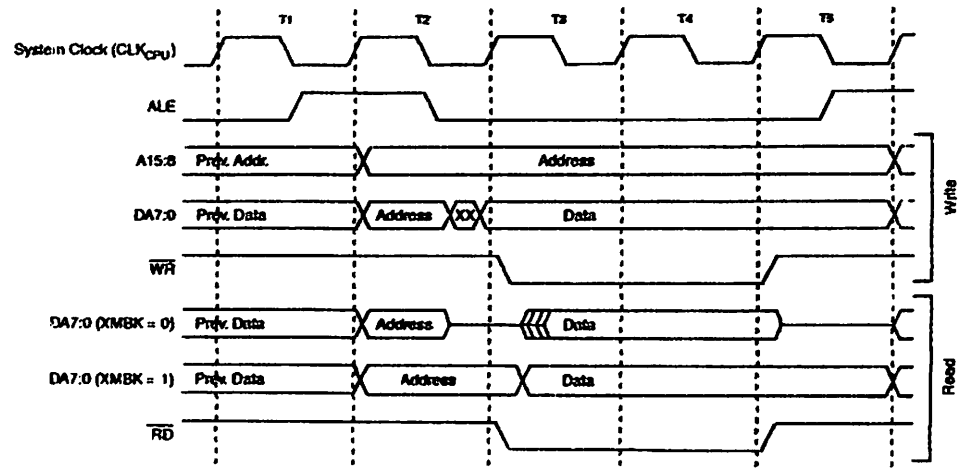
Note that the XMEM interface is asynchronous and that the waveforms in the figures below are related to the internal system clock. The skew between the internal and External clock (XTAL1) is not guaranteed (it varies between devices, temperature, and supply voltage). Consequently, the XMEM interface is not suited for synchronous operation.

Figure 13. External Data Memory Cycles without Wait State (SRWn1 = 0 and SRWn0 = 0)[1]



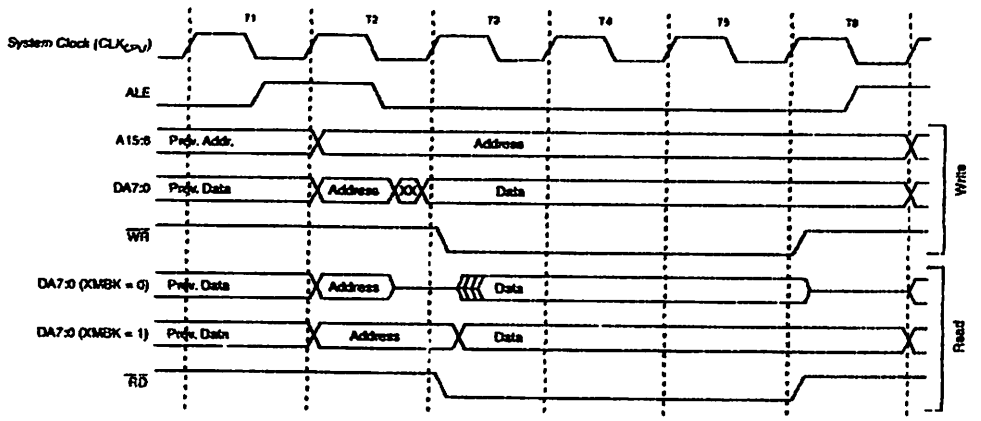
Note: 1. SRWn1 = SRW11 (upper sector) or SRW01 (lower sector), SRWn0 = SRW10 (upper sector) or SRW00 (lower sector)
The ALE pulse in period T4 is only present if the next instruction accesses the RAM (internal or external).

Figure 14. External Data Memory Cycles with SRWn1 = 0 and SRWn0 = 1[1]



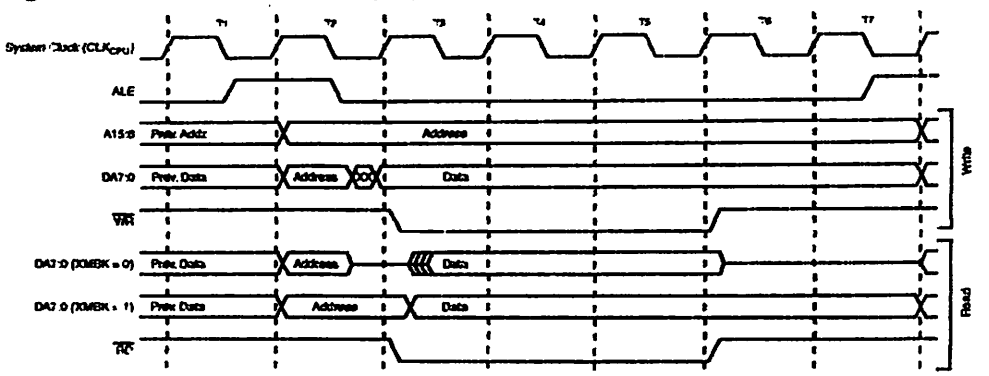
Note: 1. SRWn1 = SRW11 (upper sector) or SRW01 (lower sector), SRWn0 = SRW10 (upper sector) or SRW00 (lower sector)
The ALE pulse in period T5 is only present if the next instruction accesses the RAM (internal or external).

Figure 15. External Data Memory Cycles with SRWn1 = 1 and SRWn0 = 0[1]



Note: 1. SRWn1 = SRW11 (upper sector) or SRW01 (lower sector), SRWn0 = SRW10 (upper sector) or SRW00 (lower sector)
The ALE pulse in period T6 is only present if the next instruction accesses the RAM (internal or external).

Figure 16. External Data Memory Cycles with SRWn1 = 1 and SRWn0 = 1[1]



Note: 1. SRWn1 = SRW11 (upper sector) or SRW01 (lower sector), SRWn0 = SRW10 (upper sector) or SRW00 (lower sector)
The ALE pulse in period T7 is only present if the next instruction accesses the RAM (internal or external).

MEM Register Description

U Control Register – UCR

Bit	7	6	5	4	3	2	1	0	
	SRE	SRW10	SE	SM1	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SRE: External SRAM/XMEM Enable**

Writing SRE to one enables the External Memory Interface.The pin functions AD7:0, A15:8, ALE, $\overline{WR}$, and $\overline{RD}$ are activated as the alternate pin functions. The SRE bit overrides any pin direction settings in the respective Data Direction Registers. Writing SRE to zero, disables the External Memory Interface and the normal pin and data direction settings are used.

- **Bit 6 – SRW10: Wait State Select Bit**

For a detailed description, see common description for the SRWn bits below (EMCUCR description).

ended MCU Control ister – EMCUCR

Bit	7	6	5	4	3	2	1	0	
	SM0	SRL2	SRL1	SRL0	SRW01	SRW00	SRW11	ISC2	EMCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 6..4 – SRL2, SRL1, SRL0: Wait State Sector Limit**

It is possible to configure different wait states for different external memory addresses. The External Memory address space can be divided in two sectors that have separate wait state bits. The SRL2, SRL1, and SRL0 bits select the splitting of these sectors, see Table 2 and Figure 11. By default, the SRL2, SRL1, and SRL0 bits are set to zero and the entire External Memory address space is treated as one sector. When the entire

SRAM address space is configured as one sector, the wait states are configured by the SRW11 and SRW10 bits.

Table 2. Sector Limits with Different Settings of SRL2..0

SRL2	SRL1	SRL0	Sector Limits
0	0	0	Lower sector = N/A Upper sector = 0x0260 - 0xFFFF
0	0	1	Lower sector = 0x0260 - 0x1FFF Upper sector = 0x2000 - 0xFFFF
0	1	0	Lower sector = 0x0260 - 0x3FFF Upper sector = 0x4000 - 0xFFFF
0	1	1	Lower sector = 0x0260 - 0x5FFF Upper sector = 0x6000 - 0xFFFF
1	0	0	Lower sector = 0x0260 - 0x7FFF Upper sector = 0x8000 - 0xFFFF
1	0	1	Lower sector = 0x0260 - 0x9FFF Upper sector = 0xA000 - 0xFFFF
1	1	0	Lower sector = 0x0260 - 0xBFFF Upper sector = 0xC000 - 0xFFFF
1	1	1	Lower sector = 0x0260 - 0xDFFF Upper sector = 0xE000 - 0xFFFF

- **Bit 1 and Bit 6 MCUCR – SRW11, SRW10: Wait State Select Bits for Upper Sector**

The SRW11 and SRW10 bits control the number of wait states for the upper sector of the External Memory address space, see Table 3.

- **Bit 3..2 – SRW01, SRW00: Wait State Select Bits for Lower Sector**

The SRW01 and SRW00 bits control the number of wait states for the lower sector of the External Memory address space, see Table 3.

Table 3. Wait States[1]

SRWn1	SRWn0	Wait States
0	0	No wait states.
0	1	Wait one cycle during read/write strobe.
1	0	Wait two cycles during read/write strobe.
1	1	Wait two cycles during read/write and wait one cycle before driving out new address.

Note: 1. n = 0 or 1 (lower/upper sector).
For further details of the timing and wait states of the External Memory Interface, see Figure 13 to Figure 16 how the setting of the SRW bits affects the timing.

ecial Function IO Register – IOR

Bit	7	6	5	4	3	2	1	0	
	–	XMBK	XMM2	XMM1	XMM0	PUD	–	PSR10	SFIOR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 6 – XMBK: External Memory Bus Keeper Enable**

Writing XMBK to one enables the Bus Keeper on the AD7:0 lines. When the Bus Keeper is enabled, AD7:0 will keep the last driven value on the lines even if the XMEM interface has tri-stated the lines. Writing XMBK to zero disables the Bus Keeper. XMBK is not qualified with SRE, so even if the XMEM interface is disabled, the Bus Keepers are still activated as long as XMBK is one.

- **Bit 5..3 – XMM2, XMM1, XMM0: External Memory High Mask**

When the External Memory is enabled, all Port C pins are used for the high address byte by default. If the full 64,928 bytes address space is not required to access the External Memory, some, or all, Port C pins can be released for normal Port Pin function as described in Table 4. As described in "Using all 64KB Locations of External Memory" on page 33, it is possible to use the XMMn bits to access all 64KB locations of the External Memory.

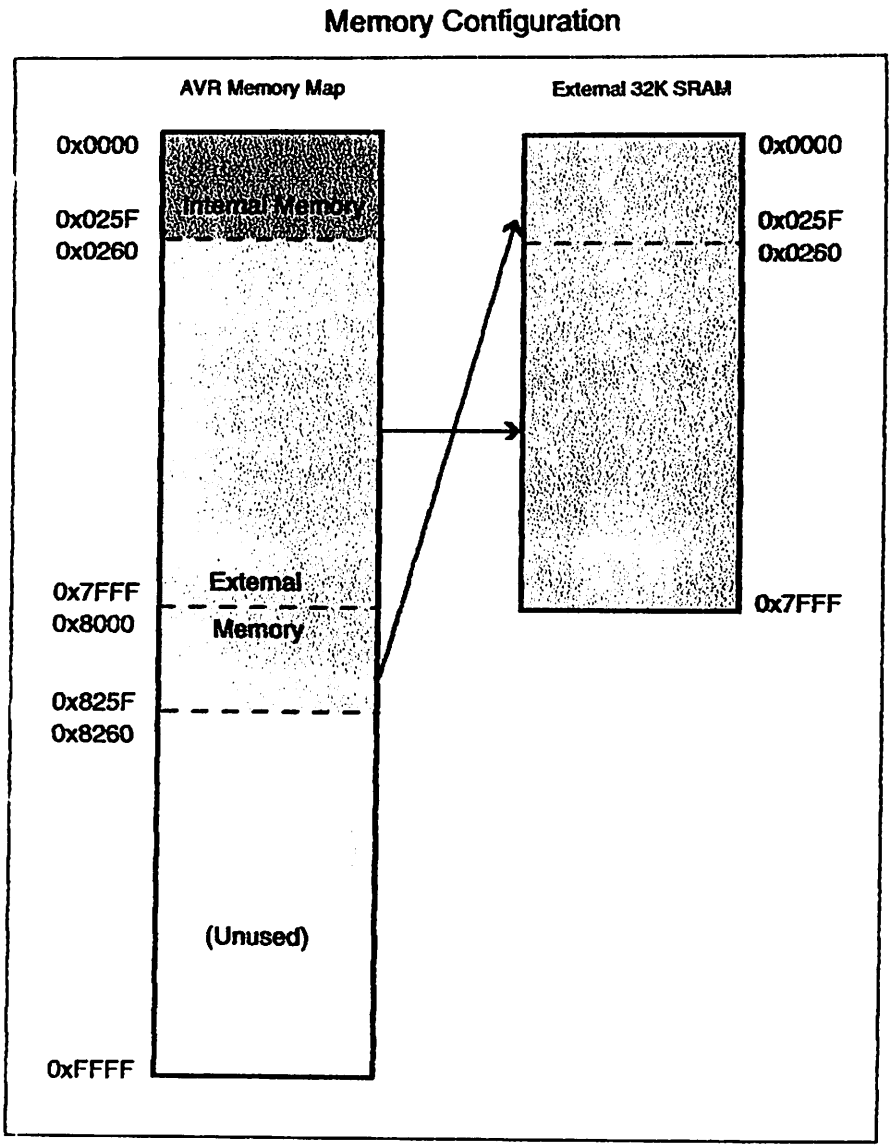
Table 4. Port C Pins Released as Normal Port Pins when the External Memory is Enabled

XMM2	XMM1	XMM0	# Bits for External Memory Address	Released Port Pins
0	0	0	8 (Full 64,928 Bytes Space)	None
0	0	1	7	PC7
0	1	0	6	PC7 - PC6
0	1	1	5	PC7 - PC5
1	0	0	4	PC7 - PC4
1	0	1	3	PC7 - PC3
1	1	0	2	PC7 - PC2
1	1	1	No Address High bits	Full Port C

ng all Locations of ernal Memory Smaller than KB

Since the external memory is mapped after the internal memory as shown in Figure 11, the external memory is not addressed when addressing the first 608 bytes of data space. It may appear that the first 608 bytes of the external memory are inaccessible (external memory addresses 0x0000 to 0x025F). However, when connecting an external memory smaller than 64 KB, for example 32 KB, these locations are easily accessed simply by addressing from address 0x8000 to 0x825F. Since the External Memory Address bit A15 is not connected to the external memory, addresses 0x8000 to 0x825F will appear as addresses 0x0000 to 0x025F for the external memory. Addressing above address 0x825F is not recommended, since this will address an external memory location that is already accessed by another (lower) address. To the Application software, the external 32 KB memory will appear as one linear 32 KB address space from 0x0260 to 0x825F. This is illustrated in Figure 17.

Figure 17. Address Map with 32 KB External Memory



ing all 64KB Locations of ternal Memory

Since the External Memory is mapped after the Internal Memory as shown in Figure 11, only 64,928 bytes of External Memory is available by default (address space 0x0000 to 0x025F is reserved for Internal Memory). However, it is possible to take advantage of the entire External Memory by masking the higher address bits to zero. This can be done by using the XMMn bits and control by software the most significant bits of the address. By setting Port C to output 0x00, and releasing the most significant bits for normal Port Pin operation, the Memory Interface will address 0x0000 - 0x1FFF. See code example below.

Assembly Code Example[1]

```
; OFFSET is defined to 0x2000 to ensure
; external memory access
; Configure Port C (address high byte) to
; output 0x00 when the pins are released
; for normal Port Pin operation
ldi  r16, 0xFF
out  DDRC, r16
ldi  r16, 0x00
out  PORTC, r16
; release PC7:5
ldi  r16, (1<<XMM1)|(1<<XMM0)
out  SFIOR, r16
; write 0xAA to address 0x0001 of external
; memory
ldi  r16, 0xaa
sts  0x0001+OFFSET, r16
; re-enable PC7:5 for external memory
ldi  r16, (0<<XMM1)|(0<<XMM0)
out  SFIOR, r16
; store 0x55 to address (OFFSET + 1) of
; external memory
ldi  r16, 0x55
sts  0x0001+OFFSET, r16
```

C Code Example[1]

```
#define OFFSET 0x2000

void XRAM_example(void)
{
unsigned char *p = (unsigned char *) (OFFSET + 1);

DDRC = 0xFF;
PORTC = 0x00;

SFIOR = (1<<XMM1) | (1<<XMM0);

*p = 0xaa;

SFIOR = 0x00;

*p = 0x55;
}
```

Note: 1. See "About Code Examples" on page 7.

Care must be exercised using this option as most of the memory is masked away.

terrupts

This section describes the specifics of the interrupt handling as performed in ATmega8515. For a general explanation of the AVR interrupt handling, refer to "Reset and Interrupt Handling" on page 13.

errupt Vectors in mega8515

Table 22. Reset and Interrupt Vectors

Vector No.	Program Address(2)	Source	Interrupt Definition
1	$000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog Reset
2	$001	INT0	External Interrupt Request 0
3	$002	INT1	External Interrupt Request 1
4	$003	TIMER1 CAPT	Timer/Counter1 Capture Event
5	$004	TIMER1 COMPA	Timer/Counter1 Compare Match A
6	$005	TIMER1 COMPB	Timer/Counter1 Compare Match B
7	$006	TIMER1 OVF	Timer/Counter1 Overflow
8	$007	TIMER0 OVF	Timer/Counter0 Overflow
9	$008	SPI, STC	Serial Transfer Complete
10	$009	USART, RXC	USART, Rx Complete
11	$00A	USART, UDRE	USART Data Register Empty
12	$00B	USART, TXC	USART, Tx Complete
13	$00C	ANA_COMP	Analog Comparator
14	$00D	INT2	External Interrupt Request 2
15	$00E	TIMER0 COMP	Timer/Counter0 Compare Match
16	$00F	EE_RDY	EEPROM Ready
17	$010	SPM_RDY	Store Program memory Ready

Notes: 1. When the BOOTRST Fuse is programmed, the device will jump to the Boot Loader address at reset, see "Boot Loader Support – Read-While-Write Self-Programming" on page 166.

2. When the IVSEL bit in GICR is set, Interrupt Vectors will be moved to the start of the Boot Flash section. The address of each Interrupt Vector will then be the address in this table added to the start address of the Boot Flash section.

Table 23 shows Reset and Interrupt Vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa.

Table 23. Reset and Interrupt Vectors Placement[1]

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	$0000	$0001
1	1	$0000	Boot Reset Address + $0001
0	0	Boot Reset Address	$0001
0	1	Boot Reset Address	Boot Reset Address + $0001

Note: 1. The Boot Reset Address is shown in Table 78 on page 177. For the BOOTRST Fuse "1" means unprogrammed while "0" means programmed.

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega8515 is:

```
Address  Labels   Code              Comments
$000              rjmp RESET        ; Reset Handler
$001              rjmp EXT_INT0     ; IRQ0 Handler
$002              rjmp EXT_INT1     ; IRQ1 Handler
$003              rjmp TIM1_CAPT    ; Timer1 Capture Handler
$004              rjmp TIM1_COMPA   ; Timer1 Compare A Handler
$005              rjmp TIM1_COMPB   ; Timer1 Compare B Handler
$006              rjmp TIM1_OVF     ; Timer1 Overflow Handler
$007              rjmp TIM0_OVF     ; Timer0 Overflow Handler
$008              rjmp SPI_STC      ; SPI Transfer Complete Handler
$009              rjmp USART_RXC    ; USART RX Complete Handler
$00a              rjmp USART_UDRE   ; UDR0 Empty Handler
$00b              rjmp USART_TXC    ; USART TX Complete Handler
$00c              rjmp ANA_COMP     ; Analog Comparator Handler
$00d              rjmp EXT_INT2     ; IRQ2 Handler
$00e              rjmp TIM0_COMP    ; Timer0 Compare Handler
$00f              rjmp EE_RDY       ; EEPROM Ready Handler
$010              rjmp SPM_RDY      ; Store Program memory Ready
Handler

$011     RESET:   ldi  r16,high(RAMEND); Main program start
$012              out  SPH,r16      ; Set Stack Pointer to top of RAM
$013              ldi  r16,low(RAMEND)
$014              out  SPL,r16
$015              sei               ; Enable interrupts
$016              <instr>  xxx
...      ...      ...
```

When the BOOTRST Fuse is unprogrammed, the Boot section size set to 2K bytes and the IVSEL bit in the GICR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

```
Address  Labels   Code                  Comments
$000     RESET:  ldi  r16,high(RAMEND); Main program start
$001             out  SPH,r16       ; Set Stack Pointer to top of RAM
$002             ldi  r16,low(RAMEND)
$003             out  SPL,r16
$004             sei                ; Enable interrupts
$005             <instr>  xxx
;
.org $C02
$C02             rjmp EXT_INT0      ; IRQ0 Handler
$C04             rjmp EXT_INT1      ; IRQ1 Handler
...      ....    ..                 ;
$C2A             rjmp SPM_RDY       ; Store Program memory Ready
Handler
```

When the BOOTRST Fuse is programmed and the Boot section size set to 2K bytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

```
Address  Labels   Code                Comments
.org $002
$001             rjmp EXT_INT0      ; IRQ0 Handler
$002             rjmp EXT_INT1      ; IRQ1 Handler
...      ....    ..                 ;
$010             rjmp SPM_RDY       ; Store Program memory Ready
Handler
;
.org $C00
$C00     RESET:  ldi  r16,high(RAMEND); Main program start
$C01             out  SPH,r16       ; Set Stack Pointer to top of RAM
$C02             ldi  r16,low(RAMEND)
$C03             out  SPL,r16
$C04             sei                ; Enable interrupts
$C05             <instr>  xxx
```

When the BOOTRST Fuse is programmed, the Boot section size set to 2K bytes and the IVSEL bit in the GICR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

```
Address  Labels   Code                  Comments
.org $C00
$C00             rjmp RESET         ; Reset handler
$C01             rjmp EXT_INT0      ; IRQ0 Handler
$C02             rjmp EXT_INT1      ; IRQ1 Handler
...      ....    ..                 ;
$C10             rjmp SPM_RDY       ; Store Program memory Ready
Handler
;
$C11     RESET:  ldi  r16,high(RAMEND); Main program start
```

```
$C12            out  SPH,r16      ; Set Stack Pointer to top of RAM
$C13            ldi  r16,low(RAMEND)
$C14            out  SPL,r16
$C15            sei               ; Enable interrupts
$C16            <instr>  xxx
```

ving Interrupts between plication and Boot Space

The General Interrupt Control Register controls the placement of the Interrupt Vector table.

neral Interrupt Control gister – GICR

Bit	7	6	5	4	3	2	1	0	
	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	GICR
Read/Write	R/W	R/W	R/W	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 1 – IVSEL: Interrupt Vector Select**

When the IVSEL bit is cleared (zero), the Interrupt Vectors are placed at the start of the Flash memory. When this bit is set (one), the Interrupt Vectors are moved to the beginning of the Boot Loader section of the Flash. The actual address of the start of the Boot Flash section is determined by the BOOTSZ Fuses. Refer to the section "Boot Loader Support – Read-While-Write Self-Programming" on page 166 for details. To avoid unintentional changes of Interrupt Vector tables, a special write procedure must be followed to change the IVSEL bit:

1. Write the Interrupt Vector Change Enable (IVCE) bit to one.
2. Within four cycles, write the desired value to IVSEL while writing a zero to IVCE.

Interrupts will automatically be disabled while this sequence is executed. Interrupts are disabled in the cycle IVCE is set, and they remain disabled until after the instruction following the write to IVSEL. If IVSEL is not written, interrupts remain disabled for four cycles. The I-bit in the Status Register is unaffected by the automatic disabling.

Note: If Interrupt Vectors are placed in the Boot Loader section and Boot Lock bit BLB02 is programmed, interrupts are disabled while executing from the Application section. If Interrupt Vectors are placed in the Application section and Boot Lock bit BLB12 is programed, interrupts are disabled while executing from the Boot Loader section. Refer to the section "Boot Loader Support – Read-While-Write Self-Programming" on page 166 for details on Boot Lock bits.

- **Bit 0 – IVCE: Interrupt Vector Change Enable**

The IVCE bit must be written to logic one to enable change of the IVSEL bit. IVCE is cleared by hardware four cycles after it is written or when IVSEL is written. Setting the IVCE bit will disable interrupts, as explained in the IVSEL description above. See Code Example below.

Assembly Code Example

```
Move_interrupts:
  ; Enable change of interrupt vectors
  ldi r16, (1<<IVCE)
  out GICR, r16
  ; Move interrupts to boot flash section
  ldi r16, (1<<IVSEL)
  out GICR, r16
  ret
```

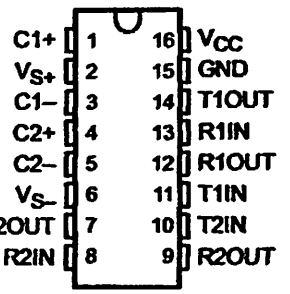
C Code Example

```
void Move_interrupts(void)
{
  /* Enable change of interrupt vectors */
  GICR = (1<<IVCE);
  /* Move interrupts to boot flash section */
  GICR = (1<<IVSEL);
}
```

SLLS047I – FEBRUARY 1989 – REVISED OCTOBER 2002

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

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



description/ordering information

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

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube	MAX232N	MAX232N
	SOIC (D)	Tube	MAX232D	MAX232
		Tape and reel	MAX232DR	
	SOIC (DW)	Tube	MAX232DW	MAX232
		Tape and reel	MAX232DWR	
	SOP (NS)	Tape and reel	MAX232NSR	MAX232
–40°C to 85°C	PDIP (N)	Tube	MAX232IN	MAX232IN
	SOIC (D)	Tube	MAX232ID	MAX232I
		Tape and reel	MAX232IDR	
	SOIC (DW)	Tube	MAX232IDW	MAX232I
		Tape and reel	MAX232IDWR	

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

Function Tables

EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

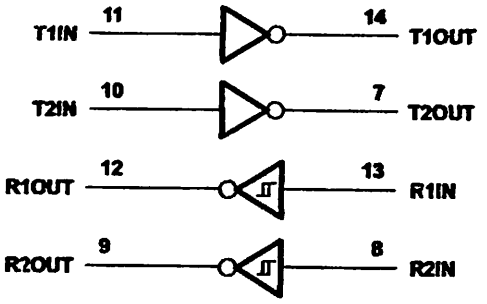
H = high level, L = low level

EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

Logic diagram (positive logic)



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

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

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

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

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

recommended operating conditions

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage		4.5	5	5.5	V
V_{IH}	High-level input voltage (T1IN,T2IN)		2			V
V_{IL}	Low-level input voltage (T1IN, T2IN)				0.8	V
R1IN, R2IN	Receiver input voltage				±30	V
T_A	Operating free-air temperature	MAX232	0		70	°C
		MAX232I	–40		85	

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

PARAMETER		TEST CONDITIONS	MIN	TYP‡	MAX	UNIT
I_{CC}	Supply current	V_{CC} = 5.5 V, All outputs open, T_A = 25°C		8	10	mA

‡ All typical values are at V_{CC} = 5 V and T_A = 25°C.

NOTE 3: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

DRIVER SECTION

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

PARAMETER			TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage	T1OUT, T2OUT	R_L = 3 kΩ to GND		5	7		V
V_{OL}	Low-level output voltage‡	T1OUT, T2OUT	R_L = 3 kΩ to GND			–7	–5	V
r_o	Output resistance	T1OUT, T2OUT	V_{S+} = V_{S-} = 0,	V_O = ±2 V	300			Ω
I_{OS}§	Short-circuit output current	T1OUT, T2OUT	V_{CC} = 5.5 V,	V_O = 0		±10		mA
I_{IS}	Short-circuit input current	T1IN, T2IN	V_I = 0				200	µA

All typical values are at V_{CC} = 5 V, T_A = 25°C.
The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.
Not more than one output should be shorted at a time.
NOTE 3: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

Switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 3)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
SR	Driver slew rate	R_L = 3 kΩ to 7 kΩ, See Figure 2			30	V/µs
SR(t)	Driver transition region slew rate	See Figure 3		3		V/µs
	Data rate	One TOUT switching		120		kbit/s

NOTE 3: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

RECEIVER SECTION

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

PARAMETER			TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage	R1OUT, R2OUT	I_{OH} = –1 mA		3.5			V
V_{OL}	Low-level output voltage‡	R1OUT, R2OUT	I_{OL} = 3.2 mA				0.4	V
V_{IT+}	Receiver positive-going input threshold voltage	R1IN, R2IN	V_{CC} = 5 V,	T_A = 25°C		1.7	2.4	V
V_{IT-}	Receiver negative-going input threshold voltage	R1IN, R2IN	V_{CC} = 5 V,	T_A = 25°C	0.8	1.2		V
V_{hys}	Input hysteresis voltage	R1IN, R2IN	V_{CC} = 5 V		0.2	0.5	1	V
	Receiver input resistance	R1IN, R2IN	V_{CC} = 5,	T_A = 25°C	3	5	7	kΩ

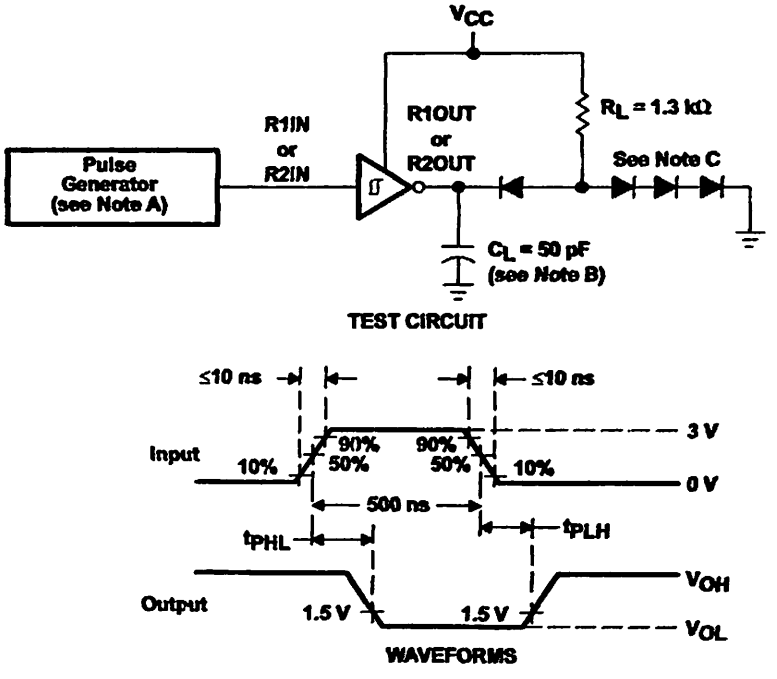
All typical values are at V_{CC} = 5 V, T_A = 25°C.
The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.
NOTE 3: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

Switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 3 and Figure 1)

PARAMETER		TYP	UNIT
$t_{PLH(R)}$	Receiver propagation delay time, low- to high-level output	500	ns
$t_{PHL(R)}$	Receiver propagation delay time, high- to low-level output	500	ns

NOTE 3: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

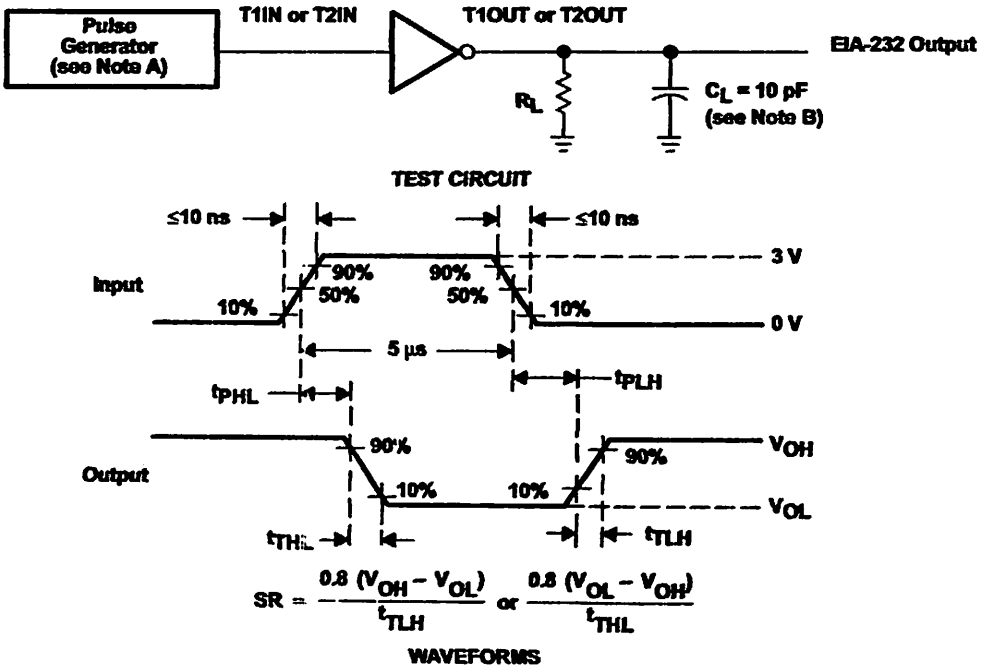
PARAMETER MEASUREMENT INFORMATION



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

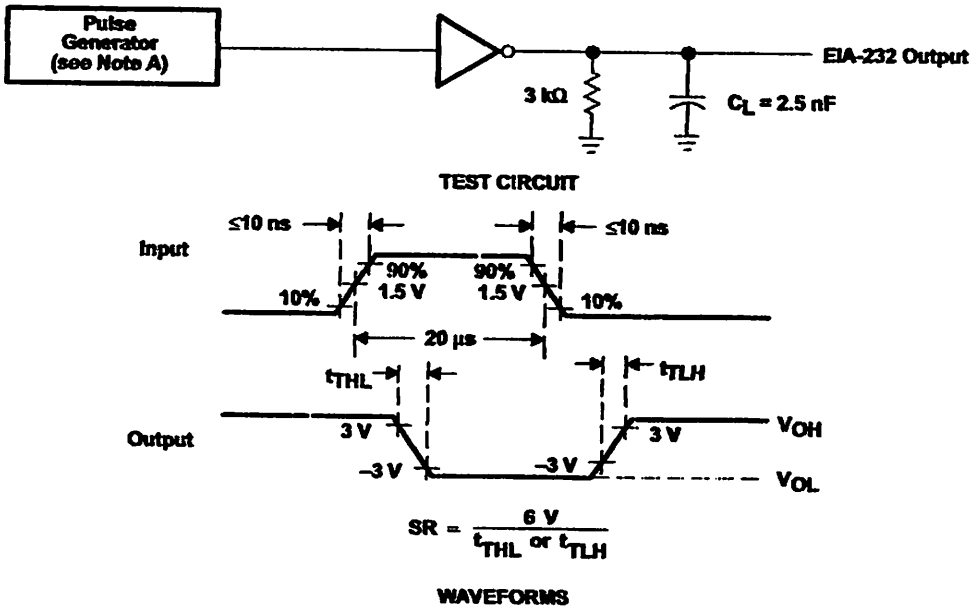
Figure 1. Receiver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements

PARAMETER MEASUREMENT INFORMATION



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

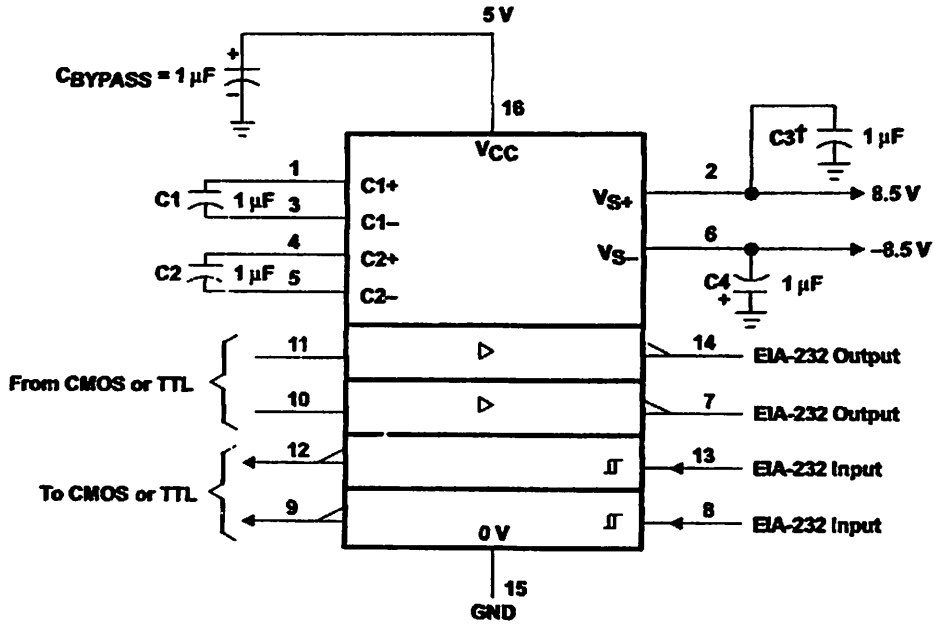
Figure 2. Driver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements (5-µs Input)



NOTE A: The pulse generator has the following characteristics: Z_O = 50 Ω, duty cycle ≤ 50%.

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

APPLICATION INFORMATION



† C3 can be connected to V_{CC} or GND.

Figure 4. Typical Operating Circuit

The SN54125, SN54126, SN74125, SN74126, and SN54LS126A are obsolete and are no longer supplied.

SN54125, SN54126, SN54LS125A, SN54LS126A, SN74125, SN74126, SN74LS125A, SN74LS126A
QUADRUPLE BUS BUFFERS WITH 3-STATE OUTPUTS

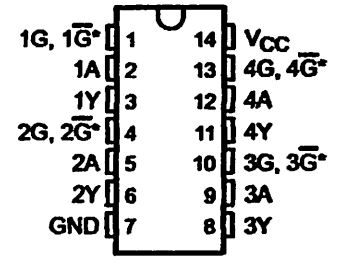
SDLS044A – DECEMBER 1983 – REVISED MARCH 2002

- Quad Bus Buffers
- 3-State Outputs
- Separate Control for Each Channel

description

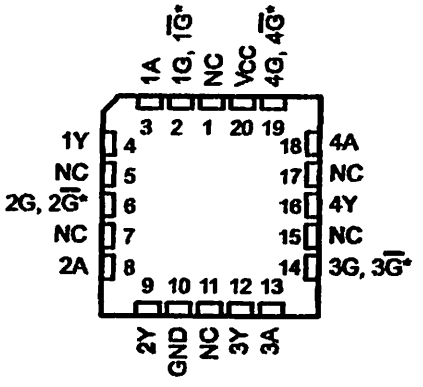
These bus buffers feature three-state outputs that, when enabled, have the low impedance characteristics of a TTL output with additional drive capability at high logic levels to permit driving heavily loaded bus lines without external pullup resistors. When disabled, both output transistors are turned off, presenting a high-impedance state to the bus so the output will act neither as a significant load nor as a driver. The '125 and 'LS125A devices' outputs are disabled when $\overline{G}$ is high. The '126 and 'LS126A devices' outputs are disabled when G is low.

SN54125, SN54126, SN54LS125A, SN54LS126A . . . J OR W PACKAGE
SN74125, SN74126 . . . N PACKAGE
SN74LS125A, SN74LS126A . . . D, N, OR NS PACKAGE
(TOP VIEW)



*$\overline{G}$ on '125 and 'LS125A devices;
G on 126 and 'LS126A devices

SN54LS125A, SN54LS126A . . . FK PACKAGE
(TOP VIEW)



*$\overline{G}$ on '125 and 'LS125A devices;
G on 126 and 'LS126A devices
NC – No internal connection

The SN54125, SN54126, SN74125, SN74126, and SN54LS126A are obsolete and are no longer supplied.

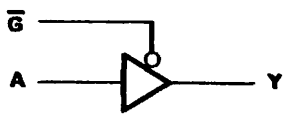
ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP – N	Tube	SN74LS125AN	SN74LS125AN
		Tube	SN74LS126AN	SN74LS126AN
	SOIC – D	Tube	SN74LS125AD	LS125A
		Tape and reel	SN74LS125ADR	
		Tube	SN74LS126AD	LS126A
		Tape and reel	SN74LS126ADR	
	SOP – NS	Tape and reel	SN74LS125ANSR	74LS125A
		Tape and reel	SN74LS126ANSR	74LS126A
–55°C to 125°C	CDIP – J	Tube	SN54LS125AJ	SN54LS125AJ
		Tube	SNJ54LS125AJ	SNJ54LS125AJ
	CFP – W	Tube	SNJ54LS125AW	SNJ54LS125AW
	LCCC – FK	Tube	SNJ54LS125AFK	SNJ54LS125AFK

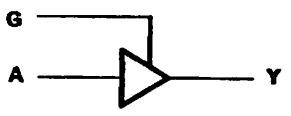
† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

gic diagram (each gate)

'125, 'LS125A



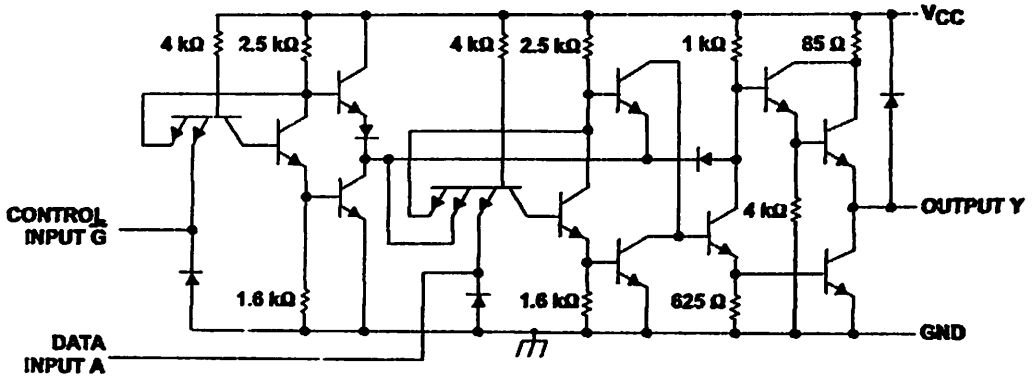
'126, 'LS126A



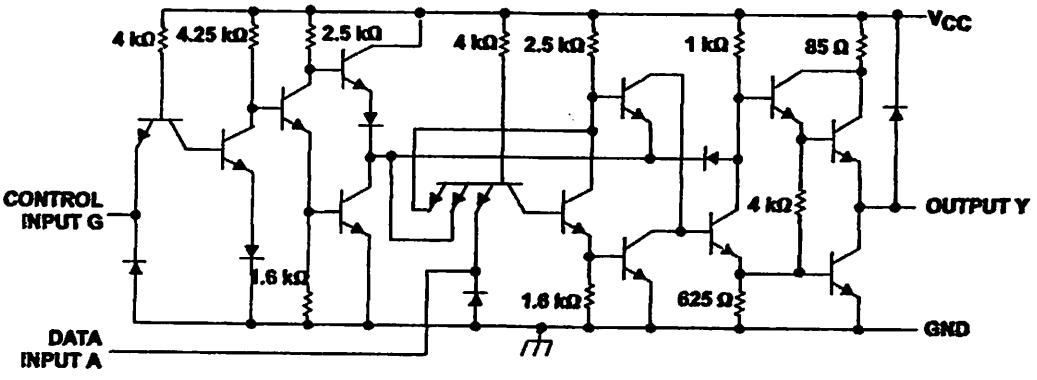
$Y = A$

schematics (each gate)

'125 CIRCUITS



'126 CIRCUITS



absolute maximum ratings over operating free-air temperature (unless otherwise noted)† ('125 and '126)

Supply voltage, V_{CC} (see Note 1) ..... 7 V
Input voltage, V_I ..... 5.5 V
Package thermal impedance, θ_{JA} (see Note 2): N package ..... 80°C/W
Storage temperature range, T_{stg} ..... –65°C to 150°C

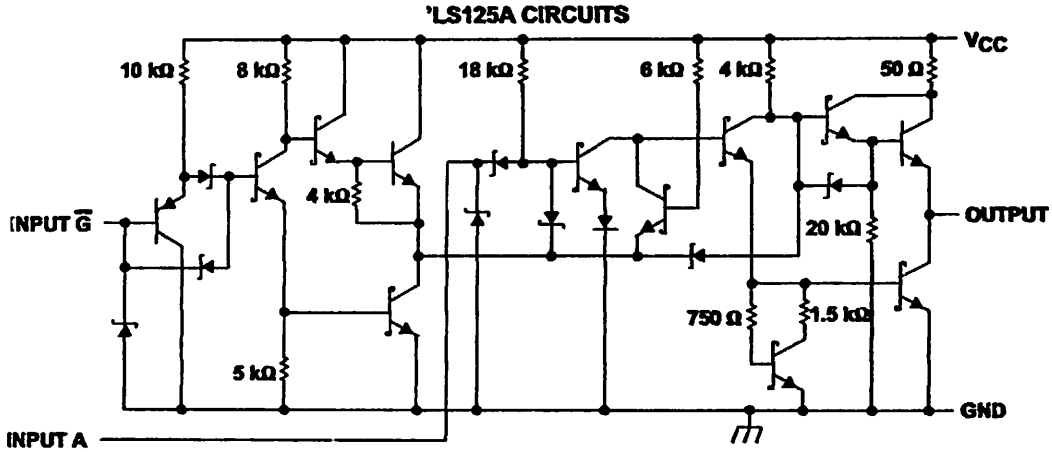
† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to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

NOTES: 1. Voltage values are with respect to network ground terminal.
2. The package termal impedance is calculated in accordance with JESD 51-7.

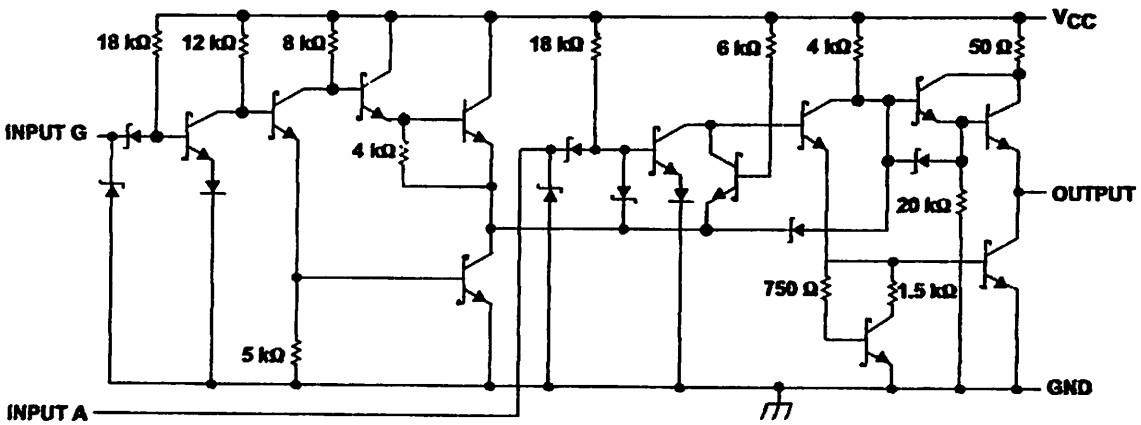
The SN54125, SN54126, SN74125, SN74126, and SN54LS126A are obsolete and are no longer supplied.

chematics (each gate)

'LS125A CIRCUITS



'LS126A CIRCUITS



Resistor values shown are nominal.

bsolute maximum ratings over operating free-air temperature (unless otherwise noted)† S125A and 'LS126A)

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage, V_I	7 V
Package thermal impedance, θ_{JA} (see Note 2): D package	86°C/W
N package	80°C/W
NS package	76°C/W
Storage temperature range, T_{stg}	–65°C to 150°C

tresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and nctional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not plied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

TES: 1. Voltage values are with respect to network ground terminal.
2. The package termal impedance is calculated in accordance with JESD 51-7.

recommended operating conditions

		SN54125 SN54126			SN74125 SN74126			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High-level input voltage	2			2			V
V_{IL}	Low-level input voltage			0.8			0.8	V
I_{OH}	High-level output current			–2			–5.2	mA
I_{OL}	Low-level output current			16			16	mA
T_A	Operating free-air temperature	–55		125	0		70	°C

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

PARAMETER	TEST CONDITIONS†			SN54125 SN54126			SN74125 SN74126			UNIT
				MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IK}	V_{CC} = MIN,	I_I = –12 mA				–1.5			–1.5	V
V_{OH}	V_{CC} = MIN, V_{IL} = 0.8 V	V_{IH} = 2 V,	I_{OH} = –2 mA	2.4	3.3					V
			I_{OH} = –5.2 mA				2.4	3.1		
V_{OL}	V_{CC} = MIN, I_{OL} = 16 mA	V_{IH} = 2 V,	V_{IL} = 0.8 V,			0.4			0.4	V
I_{OZ}	V_{CC} = MAX V_{IL} = 0.8 V	V_{IH} = 2 V,	V_O = 2.4 V			40			40	µA
			V_O = 0.4 V			–40			–40	
I_I	V_{CC} = MAX,	V_I = 6.5 V				1			1	mA
I_{IH}	V_{CC} = MAX,	V_I = 2.4 V				40			40	µA
I_{IL}	V_{CC} = MAX,	V_I = 0.4 V				–1.6			–1.6	mA
I_{OS}§	V_{CC} = MAX			–30		–70	–28		–70	mA
I_{CC}	V_{CC} = MAX (see Note 3)		'125		32	54		32	54	mA
			'126		36	62		36	62	

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡ All typical values are at V_{CC} = 5 V, T_A = 25°C.
§ Not more than one output should be shorted at a time.
NOTE 3: Data inputs = 0 V; output control = 4.5 V for '125 and 0 V for '126.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Figure 1)

PARAMETER	TEST CONDITIONS		SN54125 SN74125			SN54126 SN74126			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	R_L = 400 Ω,	C_L = 50 pF		8	13		8	13	ns
t_{PHL}				12	18		12	18	
t_{PZH}	R_L = 400 Ω,	C_L = 50 pF		11	17		11	18	ns
t_{PZL}				16	25		16	25	
t_{PHZ}	R_L = 400 Ω,	C_L = 5 pF		5	8		10	16	ns
t_{PLZ}				7	12		12	18	

The SN54125, SN54126, SN74125, SN74126, and SN54LS126A are obsolete and are no longer supplied.

ecommended operating conditions

		SN54LS125A SN54LS126A			SN74LS125A SN74LS126A			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High-level input voltage	2			2			V
V_{IL}	Low-level input voltage			0.7			0.8	V
I_{OH}	High-level output current			–1			–2.6	mA
I_{OL}	Low-level output current			12			24	mA
T_A	Operating free-air temperature	–55		125	0		70	°C

lectrical characteristics over recommended operating free-air temperature range (unless therwise noted)

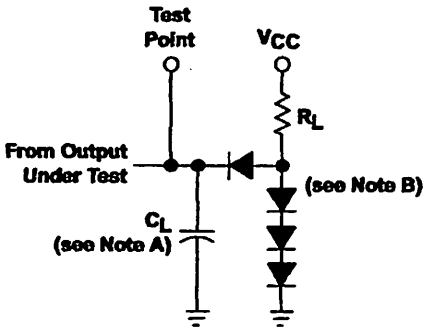
PARAMETER	TEST CONDITIONS†			SN54LS125A SN54LS126A			SN74LS125A SN74LS126A			UNIT
				MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IK}	V_{CC} = MIN,	I_I = –18 mA				–1.5			–1.5	V
V_{OH}	V_{CC} = MIN, V_{IH} = 2 V	V_{IL} = 0.7 V,	I_{OH} = –1 mA	2.4						V
		V_{IL} = 0.8 V	I_{OH} = –2.6 mA				2.4			
V_{OL}	V_{CC} = MIN, V_{IH} = 2 V	V_{IL} = 0.7 V,	I_{OL} = 12 mA		0.25	0.4				V
		V_{IL} = 0.8 V,	I_{OL} = 12 mA					0.25	0.4	
		V_{IL} = 0.8 V,	I_{OL} = 24 mA					0.35	0.5	
I_{OZ}	V_{CC} = MAX, V_{IH} = 2 V,	V_{IL} = 0.7 V	V_O = 2.4 V			20				µA
			V_O = 0.4 V			–20				
		V_{IL} = 0.8 V	V_O = 2.4 V						20	
			V_O = 0.4 V						–20	
I_I	V_{CC} = MAX,	V_I = 7 V				0.1			0.1	mA
I_{IH}	V_{CC} = MAX,	V_I = 2.7 V				20			20	µA
I_{IL}	V_{CC} = MAX, V_I = 0.4 V	'LS125A-G inputs				–0.2			–0.2	mA
		'LS125A-A inputs; 'LS126A All inputs				–0.4			–0.4	mA
I_{OS}§	V_{CC} = MAX			–40		–225	–40		–225	mA
I_{CC}	V_{CC} = MAX (see Note 4)		'LS125A		11	20		11	20	mA
			'LS126A		12	22		12	22	

For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
All typical values are at V_{CC} = 5 V, T_A = 25°C.
Not more than one output should be shorted at a time, and the duration of the short circuit should not exceed one second.
NOTE 4: Data inputs = 0 V; output control = 4.5 V for 'LS125A and 0 V for 'LS126A.

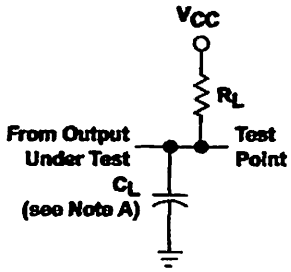
witching characteristics, V_{CC} = 5 V, T_A = 25°C (see Figure 1)

PARAMETER	TEST CONDITIONS		SN54LS125A SN74LS125A			SN54LS126A SN74LS126A			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	R_L = 667 Ω,	C_L = 45 pF		9	15		9	15	ns
t_{PHL}				7	18		8	18	
t_{PZH}	R_L = 667 Ω,	C_L = 45 pF		12	20		16	25	ns
t_{PZL}				15	25		21	35	
t_{PHZ}	R_L = 667 Ω,	C_L = 5 pF			20			25	ns
t_{PLZ}					20			25	

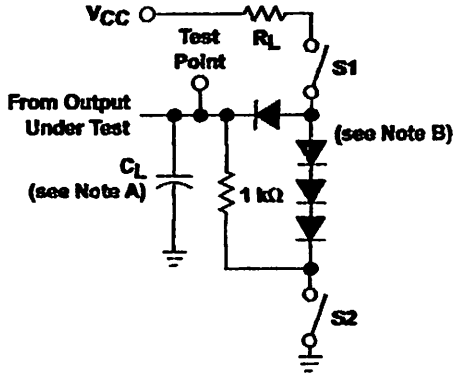
PARAMETER MEASUREMENT INFORMATION
SERIES 54/74 DEVICES



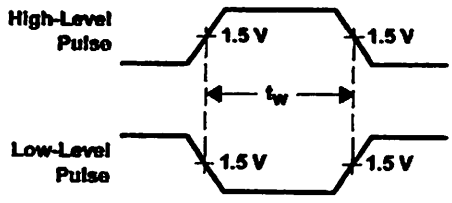
LOAD CIRCUIT
FOR 2-STATE TOTEM-POLE OUTPUTS



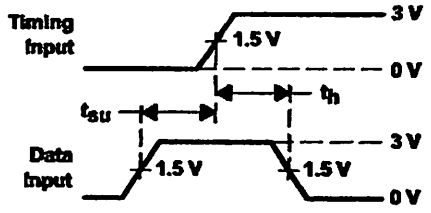
LOAD CIRCUIT
FOR OPEN-COLLECTOR OUTPUTS



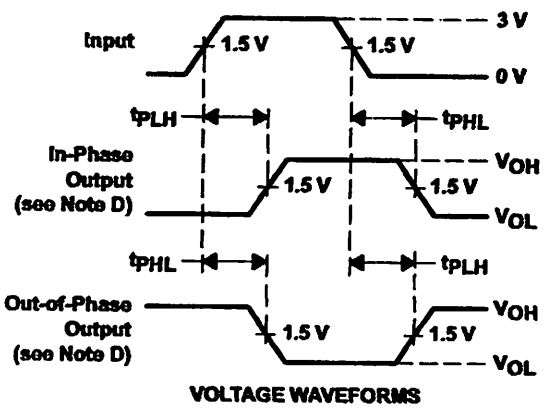
LOAD CIRCUIT
FOR 3-STATE OUTPUTS



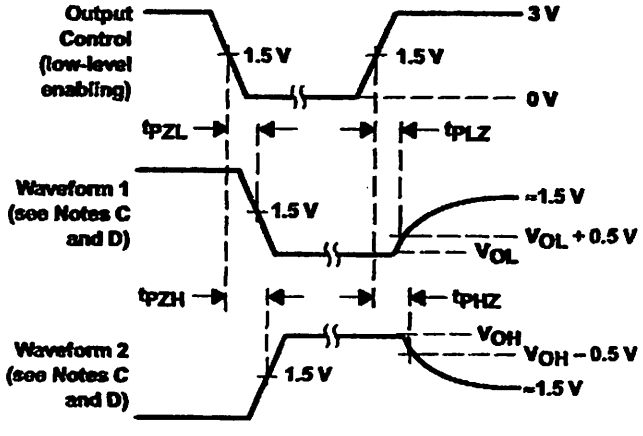
VOLTAGE WAVEFORMS
PULSE DURATIONS



VOLTAGE WAVEFORMS
SETUP AND HOLD TIMES



VOLTAGE WAVEFORMS
PROPAGATION DELAY TIMES



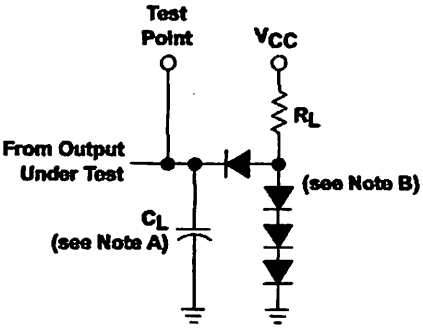
VOLTAGE WAVEFORMS
ENABLE AND DISABLE TIMES, 3-STATE OUTPUTS

NOTES: A. C_L includes probe and jig capacitance.
B. All diodes are 1N3064 or equivalent.
C. Waveform 1 is for an output with internal conditions such that the output is low except when disabled by the output control. Waveform 2 is for an output with internal conditions such that the output is high except when disabled by the output control.
D. S1 and S2 are closed for t_{PLH}, t_{PHL}, t_{PHZ}, and t_{PLZ}; S1 is open and S2 is closed for t_{PZH}; S1 is closed and S2 is open for t_{PZL}.
E. All input pulses are supplied by generators having the following characteristics: PRR ≤ 1 MHz, Z_O = 50 Ω, t_r and t_f ≤ 7 ns for Series 54/74 devices and t_r and t_f ≤ 2.5 ns for Series 54S/74S devices.
F. The outputs are measured one at a time with one input transition per measurement.

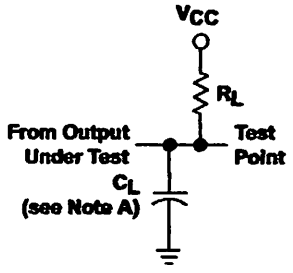
Figure 1. Load Circuits and Voltage Waveforms

The SN54125, SN54126, SN74125, SN74126, and SN54LS126A are obsolete and are no longer supplied.

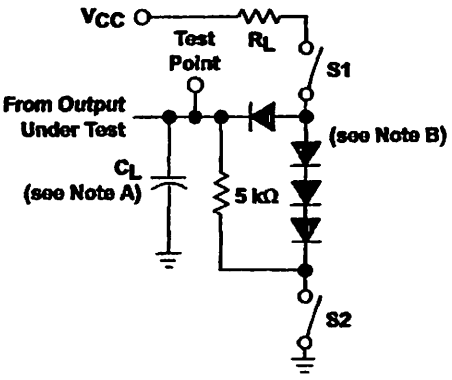
PARAMETER MEASUREMENT INFORMATION
SERIES 54LS/74LS DEVICES



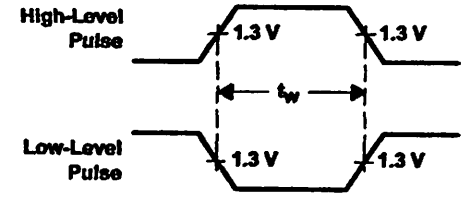
LOAD CIRCUIT
FOR 2-STATE TOTEM-POLE OUTPUTS



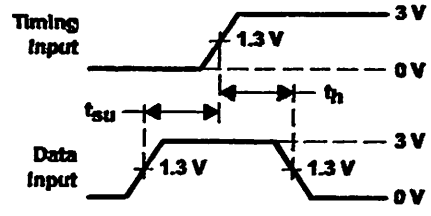
LOAD CIRCUIT
FOR OPEN-COLLECTOR OUTPUTS



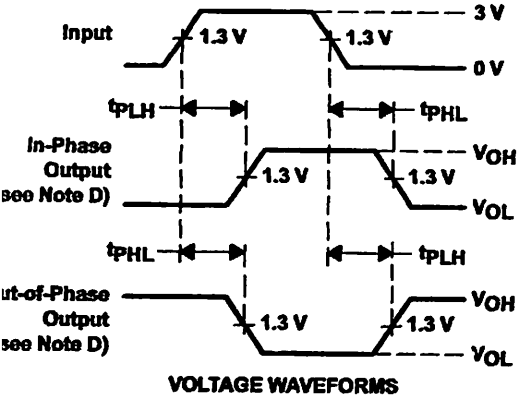
LOAD CIRCUIT
FOR 3-STATE OUTPUTS



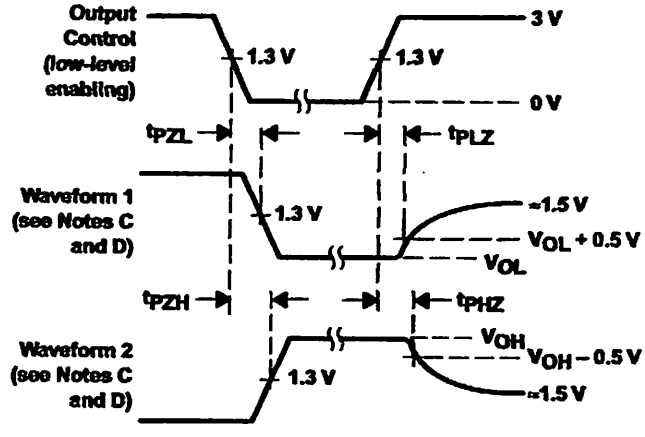
VOLTAGE WAVEFORMS
PULSE DURATIONS



VOLTAGE WAVEFORMS
SETUP AND HOLD TIMES



VOLTAGE WAVEFORMS
PROPAGATION DELAY TIMES



VOLTAGE WAVEFORMS
ENABLE AND DISABLE TIMES, 3-STATE OUTPUTS

OTES: A. C_L includes probe and jig capacitance.
B. All diodes are 1N3064 or equivalent.
C. Waveform 1 is for an output with internal conditions such that the output is low except when disabled by the output control. Waveform 2 is for an output with internal conditions such that the output is high except when disabled by the output control.
D. S1 and S2 are closed for t_{PLH}, t_{PHL}, t_{PHZ}, and t_{PLZ}; S1 is open and S2 is closed for t_{PZH}; S1 is closed and S2 is open for t_{PZL}.
E. Phase relationships between inputs and outputs have been chosen arbitrarily for these examples.
F. All input pulses are supplied by generators having the following characteristics: PRR ≤ 1 MHz, $Z_O \approx 50\ \Omega$, $t_r \leq 1.5$ ns, $t_f \leq 2.6$ ns.
G. The outputs are measured one at a time with one input transition per measurement.

Figure 2. Load Circuits and Voltage Waveforms