

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



**PERENCANAAN DAN PEMBUATAN SISTEM MONITORING
KECEPATAN DAN ARAH ANGIN YANG DAPAT DIAKSES
VIA SMS BERBASIS AT89S8252**

SKRIPSI

**Disusun Oleh:
ZAINUL HASAN
01.17.117**

MARET 2006

LEMBAR PERSETUJUAN



PERENCANAAN DAN PEMBUATAN SISTEM MONITORING KECEPATAN DAN ARAH ANGIN YANG DAPAT DIAKSES VIA SMS BERBASIS AT89S8252

SKRIPSI

*Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Syarat
Guna Mencapai Gelar Sarjana Teknik Elektro Strata Satu (S-1)*

Disusun Oleh :

ZAINUL HASAN
01.17.117

Mengetahui,
Ketua Jurusan Teknik Elektro S-1



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

Diperiksa dan Disetujui,
Dosen Pembimbing



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

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



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

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Zainul Hasan
NIM : 01.17.117
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Judul Skripsi : Perencanaan dan Pembuatan Sistem Monitoring Kecepatan Dan Arah Angin Yang Dapat Diakses Via SMS Berbasis AT89S8252

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

pada :

Hari : Senin
Tanggal : 20 Maret 2006
Dengan Nilai : **86,775 (A)**

Panitia Ujian Skripsi



Ketua

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

Sekretaris



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

Anggota Penguji

Penguji I



(Ir. Usman Djuanda, MM)

Penguji II



(M. Ashar, ST, MT)

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dengan menyebut nama Allah Yang Maha Pemurah lagi Maha Penyayang.

(QS. Al Faatihah 1)

قُلْ إِن صَلَائِي وَإِشْرَاقِي وَمَحْيَايَ وَمَمَاتِي
إِلَهُ رَبِّ الْعَالَمِينَ

Katakanlah : "Sesungguhnya sholatku, ibadahku, hidupku dan matiku hanyalah untuk Allah, Tuhan semesta alam. Tiada sekutu baginya; dan demikian itulah yang diperintahkan kepadaku dan aku adalah orang yang pertama-tama menyerahkan diri (kepada Allah)".

(QS. Al An'am 162 - 163)

إِنَّمَا الْمُؤْمِنُونَ الَّذِينَ إِذَا ذُكِرَ اللَّهُ وَجِلَتْ قُلُوبُهُمْ وَإِذَا تَلَيَّكَ عَلَيْهِمْ آيَةٌ مِنْ رَبِّهِمْ زَادَتْهُمْ إِيمَانًا وَعَلَىٰ رَبِّهِمْ يَتَوَكَّلُونَ

Sesungguhnya orang-orang yang beriman ialah mereka yang bila disebut nama Allah gemetarlah hati mereka, dan apabila dibacakan ayat-ayatNya bertambahlah iman mereka (karenanya), dan hanya kepada Tuhanlah mereka bertawakkal.

(QS. Al Anfaal 2)

وَالْعَصْرُ ۝ إِنَّ الْإِنْسَانَ لِرَبِّهِ لَكَنُ خَسِرٌ ۝ إِلَّا الَّذِينَ
آمَنُوا وَعَمِلُوا الصَّالِحَاتِ وَتَوَّصُوا بِالحَقِّ وَتَوَّصُوا بِالصَّبْرِ ۝

Demi masa. Sesungguhnya manusia itu benar-benar dalam kerugian, kecuali orang-orang yang beriman dan mengerjakan amal saleh dan nasehat menasehati supaya mentaati kebenaran dan nasehat menasehati supaya menetapi kesabaran.

(QS. Al-Ashr 1-3)



Tak semua apa yang telah kita lewati bisa terwakili oleh goresan pena. Kumpulan warna, garis, titik, kurva membentuk sekumpulan gambar. Meskipun nampak diam namun sebenarnya mereka bisa berbicara seperti pena bicara lewat goresannya....

Kenangan dalam hidup yang telah kita lewati tentunya terlalu berharga untuk dilupakan. Bagaimanapun perjalanan hidup kita yang kadang silih berganti antara suka dan duka, itu semua adalah bagian dari lukisan kehidupan yang tidak di batasi oleh dimensi panjang dan lebar melainkan dibatasi oleh dimensi waktu. Otak ini terlalu kecil untuk menyimpan semua kenangan-kenangan tersebut tapi dengan bantuan sebuah pena kita bisa membantu mengingat itu semua.. oleh karena itu.....biarkan pena itu terus menari

Masa depan yang cerah berdasarkan pada masa lalu yang telah dilupakan. Kamu tidak akan dapat melangkah dengan baik dalam kehidupanmu sampai kamu melupakan kegagalan kamu di masa yang lalu.

Orang Yang Bijaksana Tidak Selalu Duduk Termenung Mearatapi Kegagalannya, Tapi Orang Yang Bijaksana Adalah Orang Yang Selalu Berusaha Untuk Bangkit Kembali Dan Selalu Belajar Dari Pengalaman Pahit Yang Pernah Dialami

Kebanyakan orang membuang banyak waktu dan tenaga hanya untuk memikirkan masalah dan bukan mencoba untuk menyelesaikannya.

Orang yang paling bahagia tidak selalu memiliki hal-hal terbaik, tapi bagaimana kita berusaha menjadikan yang terbaik dari setiap hal yang hadir dalam hidup kita.

Letak kebahagiaan itu sangat tinggi, namun kita harus menuju kesana. Kadang ada orang yang putus asa berjalan kearahnya karena disangkanya jalan kesana sangat sulit, padahal sebenarnya mudah, karena ia harus memulai dari dirinya sendiri.

Dunia ini umpama lautan yang luas. Kita adalah kapal yang berlayar dilautan yangb ramai kapal karam didalamnya....andai muatan kita adalah iman dan layarnya taqwa, niscaya kita akan selamat dari tersesat dilautan hidup ini.

Harta yang paling menguntungkan ialah SABAR. Teman yang paling akrab adalah AMAL. Pengawal peribadi yang paling waspada DIAM. Bahasa yang paling manis SENYUM. Dan ibadah yang paling indah tentunya KHUSYUK.



SPECIAL THANKS TO:

Alhamdulillah.....Segala Puji Syukur yang sedalam-dalamnya Saya Panjatkan Kepada Allah SWT, karena atas Pertolongan dan Ridho-Nyalah Skripsi ini Dapat Saya Selesaikan. Sholawat dan Salam Saya Haturkan Kepada Nabi Muhammad SAW, Para Keluarga Serta Para Sahabatnya. Pada Kesempatan ini Penulis Juga Sampaikan Rasa Hormat dan Terimakasih Kepada Semua Pihak Yang Telah Membantu Dalam Menyelesaikan Skripsi ini. Secara Khusus Saya Ucapkan Banyak – banyak Terimakasih Kepada :

1. Bapak dan Nyae' yang selalu mendoakan anakmu ini. Pak....Nyae'... anakmu sekarang sudah jadi Sarjana, semuanya berkat do'a dari Bapak&Nyae', kupersembahkan semuanya yang terbaik untukmu. Pak...Ye'...., engkau tidak pernah mengenal lelah dan putus asah dalam mendidik anakmu ini, membiayai, serta mengarahkan aku kejalan yang di Ridhoi Oleh Allah SWT. Ya.....Allah Yang Maha Kuasa, berikan aku kesempatan Ya....Allah agar selalu bisa membahagiakan kedua Orang Tuaku dan selalu bisa memberikan yang terbaik buat mereka.
2. Buat Ema'ku (Alm), aku sekarang udah jadi sarjana Ma'...., aku akan selalu berdo'a untuk Ema'. Buat Pak Enju', pak...engko' sateya lateddi Sarajana. Doakan aku agar semua cita-citaku tercapai dan menjadi orang yang sukses.
3. Buat kakakku Tommy M Mabrur yang selalu memberikan semangat untuk adikmu ini agar cepat selesai kuliah dan bisa cepet dapat kerja, kupersembahkan yang terbaik ini untukmu. Buat Mbakku Yumnawati, meskipun setiap kita ketemu kadang sering bertengkar. Untuk Mbak kupersembahkan semua yang terbaik untukmu, terimakasih banyak Mbak atas semua kasih sayangmu untuk adikmu ini.
4. Kak Didi (Sumatera), makasih banyak kak atas semuanya, meskipun kakak ada di sumatera, tapi komunikasi kita tetap lancar. Kapan-kapan aku main ke Sumatera ya....kak!!!Buat kak Rida di Ra'as, terimakasih yach..., saat telpon di rumah rusak kakak telah membantu mempermudah bapakku kalau mau menghubungi aku. Buat Mak Rasyidi di Sumenep,

Terimakasih banyak mak, yang telah meluangkan waktunya setiap bulan untuk membantu Bapak dalam mengirimkan duit buat aku. Buat YEYE (Kuta-Bali), aku ingin main ke Bali lagi, tunggu aku disana ya....!Ok Dech Aku doakan semoga usahanya lancar.

5. Bapak Ir. F. Yudi Limpraptono, MT selaku kepala jurusan Teknik Elektro S-1 sekaligus sebagai dosen pembimbing yang selalu sabar dalam membina saya baik yang bersifat teknis maupun non-teknis, sehingga penulisan skripsi ini dapat diselesaikan.
6. Ibu Ir. Mimin Mustikawati selaku Sekretaris Jurusan Teknik Elektro S-1, Institut Teknologi nasional Malang.
7. Bapak Ir. Usman Djuanda, MM, Bapak M. Ashar, ST, MT selaku dosen penguji yang telah memberikan saran dan kritik demi sempurnya Skripsi ini.
8. Ibu Puji(Recording Jurusan Teknik Elektro S-1), Pak Sugeng & Pak Jayeng, Makasih banyak ya pak....bu....., semua urusan administrasiku lancar berkat bantuan bapak & Ibu.
9. Bapak dan Ibu Dosen Jurusan Teknik Elektro S-1, yang telah memberikan ilmunya untukku. Doakan ya pak...bu....semoga ilmu yang saya dapatkan bermanfaat.
- 10.Chayankku "Any Sujarwati ", Sayang....kamu sangat berarti dalam hidupku, Apapun yang aku kerjakan, kemanapun aku melangkahkakan kaki, bayanganmu selalu hadir dalam setiap gerak langkahku. Thanks atas semua kesabaranmu yang selalu menemani aku, kamu telah mengisi hari-hariku dengan sesuatu yang indah. Makasih...atas cinta & kasih sayangmu untukku, bahasa rohanimu telah membangkitkan semangatku untuk menatap masa depan.
- 11.Sahabatku Hadaruddin, dar....kamu adalah temen baikku, makasih atas semuanya, aku tidak akan pernah melupakan semua jasa-jasa baikmu.
- 12.Sahabatku Dadang, dang...terimakasih banyak atas semuanya....., semua kebaikanmu tidak akan pernah aku lupakan.
- 13.Sahabatku Fajar(K-cit), Avan, Terimakasih banyak atas semuanya, aku terlalu sering merepotkan kamu, semoga persahabatan ini akan tetap langgeng untuk selamanya.
- 14.Untuk Temen-temen kontarakanku di Perum Puri Kartika Puskopad C-21 (Eko, Mas Asis, Feri, D-wang) Makasih banyak atas semuanya.
- 15.Buat Temen-temen baikku di ITS-Suarabaya, Asempayung 26 (Firman,hadi,pi2k,karno,wahyu). Thanks atas bantuan dan senda guraunya. Firman.....aku akan selalu ingat apa yang kamu katakan"Segera-disegerakan".

16. Buat Sahabat-sahabatku di Bali (Marwi, Sahru, Buhraini, Musarrafa/Tongkol/Chokie, Enjo, kolet). Terimakasih banyak atas semua kebaikanmu sobat. Kalau aku ke Bali kamu semua menyambutku dengan penuh kegembiraan. Aku tidak pernah melupakan Semua kebaikanmu selalu untuk selamanya.
17. Sahabat-sahabatku di Bendungan Sigura-gura 1A (Mas Herman, Om Didang, Pak De, Pak Hari, Jhon), Mas Herman, Om Didang(Terimakasih banyak atas semua kebaikanmu, semoga tali silaturahmi kita tidak akan pernah putus selamanya, Aku doakan semoga cepet Lulus ya...mas). Pak De, Pak Hari (Aku doakan Cepet Lulus ya....!!!, jangan Cuma gossip terus De....). Jhon (Kuliah yang rajin, kerjain tu...laporan, jangan males, kalau males kayak gitu ntar gak selesai-selesai donk kuliahnya).
18. Buat Temen baikkku Nanang Elka3, makasih banyak Nang atas printernya. Buat semua temen ELKA 2001 Thank's to all.



ZAINUL HASAN
PARA NYAE'

ABSTRAKSI

Zainul Hasan, 2006, Skripsi
Dosen Pembimbing : Ir. F. Yudi Limpraptono, MT

Cuaca sangat berpengaruh terhadap aktifitas manusia. Perubahan cuaca selain tergantung pada intensitas dan durasi harian dari energi radian matahari yang diterima di atmosfer diatas permukaan daerah, juga dipengaruhi oleh faktor kecepatan dan arah angin. Angin berfungsi sebagai penyebar suhu secara mendatar. Sehingga kecepatan angin beserta arah angin merupakan factor yang penting untuk mengetahui suhu di daerah tertentu. Untuk itu di butuhkan suatu sistem yang dapat mendeteksi kecepatan dan arah angin yang pembacaannya cepat dan mudah serta akurat.

Hal ini dapat direalisasikan dengan sistem monitoring kecepatan dan arah angin secara digital, yang dapat dibuat dengan menggunakan Mikrokontroler AT89S8252 sebagai Unit pemroses utama dan beberapa perangkat lain seperti sensor kecepatan dan sensor arah angin serta perangkat pendukung berupa PC yang digunakan untuk menampilkan dan menyimpan data perubahan kecepatan dan arah angin serta HP sebagai sarana komunikasi untuk mengirimkan informasi ke HP lain yang *me-request*.

Keluaran pada sistem untuk kecepatan angin berupa logika yang akan di counter per-detiknya dengan menggunakan satuan Km/Jam. Adapun kemampuan pengukuran kecepatan angin adalah dari 0 Km/Jam. Untuk arah angin, keluaran pada sistem juga berupa logika yang ditampilkan di PC dalam derajat yang mempunyai beda $11\frac{1}{4}^{\circ}$ tiap sudutnya. Dari hasil pengujian error untuk kecepatan angin 5,1% dan error untuk Arah Angin 1,67%.

Sistem ini dibuat untuk memberikan kemudahan bagi setiap orang yang dalam melakukan aktifitasnya sangat tergantung pada cuaca yang dipengaruhi oleh faktor kecepatan dan arah angin.

Kata Kunci : *Sistem Monitoring, Kecepatan dan Arah Angin, Mikrokontroler AT89S8252, Ms Visual Basic, Via SMS*

KATA PENGANTAR



Segala puji syukur saya panjatkan kehadiran Allah SWT, karena atas Petolongan dan Ridho-Nyalah, penyusun dapat menyelesaikan Skripsi ini dengan judul :

Perencanaan Dan Pembuatan Sistem Monitoring Kecepatan dan Arah Angin Yang Dapat Diakses Via SMS Berbasis AT89S8252

Skripsi ini merupakan syarat untuk menempuh program studi strata (S-1) Jurusan Teknik Elektro S-1, Konsentrasi Teknik Elektronika, Fakultas Teknologi Industri Di Institut Teknologi Nasional Malang.

Selama penyusunan Skripsi ini banyak masukan dan bantuan yang penyusun peroleh, sehingga pada kesempatan kali ini dengan segala kerendahan hati dan penuh rasa hormat penyusun menyampaikan terima kasih kepada yang terhormat:

1. Bapak Dr. Ir. Abraham Lomi, MSEE selaku Rektor ITN Malang.
2. Bapak Ir. Mochtar Asroni, MSME selaku Dekan Fakultas Teknologi Industri ITN Malang.
3. Bapak Ir. F. Yudi Limpraptono, MT selaku ketua Jurusan Teknik Elektro S-1 ITN Malang sekaligus sebagai Dosen Pembimbing.

4. Ibu Ir. Mimin Mustikawati selaku Sekretaris Jurusan Teknik Elektro S-1, Institut Teknologi nasional Malang.
5. Semua pihak yang tidak bisa disebutkan satu per satu dan telah memberikan dukungan kepada penulis atas terselesaikannya penyusunan skripsi ini.

Semoga Allah SWT memberikan balasan yang setimpal atas bimbingan, petunjuk serta dorongan yang telah diberikan pada kami selaku penyusun skripsi .

Selanjutnya dengan telah terselesaikannya skripsi ini, maka penyusun sadar bahwa skripsi ini masih perlu disempurnakan, maka berbagai masukan, saran, dan kritik yang sifatnya membangun sangat diharapkan untuk penyusunan skripsi berikutnya.

Akhirnya kepada Allahlah saya memohon, semoga Skripsi ini bermanfaat bagi kita semua serta mendapat Ridho Allah SWT sehingga kita mendapat petunjuk di bawah naungan bendera Nabi Muhammad SAW, Amiiin.

Malang, 13 Maret 2006

Penulis

Zainul Hasan

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
BERITA ACARA SKRIPSI	iii
LEMBAR PERSEMBAHAN	iv
ABSTRAKSI	ix
KATA PENGANTAR	x
DAFTAR ISI	xii
DAFTAR GAMBAR	xvi
DAFTAR TABEL	xviii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan	3
1.4. Batasan Masalah	3
1.5. Metodologi	3
1.6. Sistematika Penulisan	5
BAB II LANDASAN TEORI	6
2.1. Pendahuluan	6
2.2. Optocoupler	6
2.3.1. LED Infra Red	7
2.3.2. Photodiode	8
2.3.3. Piringan Optocoupler	8
2.3. Mikrokontroler AT89S8252	9
2.3.1. Fitur Mikrokontroler AT89S8252	10
2.3.2. Arsitektur AT89S8252	12
2.3.3. Konfigurasi Kaki-kaki MCU AT89S8252	13
2.3.4. Organisasi Memori MCU AT89S8252	16
2.3.4.1. Memori Program	16
2.3.4.2. Memori Data	18

2.3.4.3.	SFR Tambahan Pada AT89S8252.....	23
2.3.4.3.1.	SFR Untuk Timer 2	23
2.3.4.3.2.	SFR Untuk Watchdog dan Memori... ..	24
2.3.4.3.3.	SFR Pengontrol SPI	24
2.3.4.3.4.	Osilator	25
2.3.4.3.5.	Timer dan Counter	26
2.3.4.3.6.	Sistem Interrupt.....	28
2.3.5.	Metode Pengamatan	29
2.3.6.	Bahasa Assembler MCS-51	30
2.4.	Komunikasi Serial.....	32
2.5.	Metode Transmisi Serial	33
2.6.	Interface RS 232.....	34
2.7.	Pemrograman Visual Basic	40
2.7.1.	Memulai Visual Basic	41
2.7.2.	Tampilan Awal Visual Basic.....	41
2.7.3.	Komponen Visual Basic.....	41
2.7.4.	Mengolah Toolbox	49
2.8.	Hand Phone	51
2.9.	Protokol Komunikasi Data Serial Pada Siemens.....	52
2.9.1.	Karakteristik Level Signal Pada FBUS	52
2.9.2.	Protokol FBUS	52
2.9.3.	Format Data Pada Protokol FBUS	53
2.9.4.	PDU Untuk Kirim SMS Ke SMS-Centre.....	53
2.9.5.	PDU Untuk Terima SMS dari SMS-Centre	59
BAB III	PERENCANAAN DAN PEMBUATAN ALAT	61
3.1.	Gambaran Umum Sistem	61
3.2.	Perancangan Perangkat Keras	61
3.2.1.	Perancangan Rangkaian Sensor Arah Angin	65
3.2.1.1.	Perancangan Rangkaian Pemancar LED Infra Red	65
3.2.1.2.	Perancangan Rangkaian Penerima Infra Red ..	66

3.2.2.	Perancangan Rangkaian Sensor Kecepatan Angin.....	67
3.2.3.	Piringan Pada Sensor.....	70
3.2.3.1.	Piringan untuk Sensor Kecepatan Angin	70
3.2.3.2.	Piringan Untuk Sensor Arah Angin	72
3.2.4.	Sistem Mikrokontroller AT89S8252.....	75
3.2.5.	Interface RS-232	77
3.2.6.	Perencanaan Perangkat Lunak.....	78
BAB IV ANALISIS DAN PENGUJIAN ALAT.....		80
4.1.	Pengujian dan Pengukuran Optocoupler	80
4.1.1.	Tujuan.....	80
4.1.2.	Peralatan yang Digunakan.....	80
4.1.3.	Prosedur Pengujian.....	80
4.1.4.	Hasil Pengujian	80
4.1.5.	Analisis Hasil Pengujian	82
4.2.	Pengujian dan Pengukuran Sensor Arah Angin	82
4.2.1.	Tujuan.....	82
4.2.2.	Peralatan Yang Digunakan.....	82
4.2.3.	Prosedur Pengujian.....	82
4.2.4.	Hasil Pengujian	83
4.2.5.	Analisis Hasil Pengujian	84
4.3.	Pengujian mikrokontroller AT89S8252	84
4.3.1.	Tujuan.....	84
4.3.2.	Prosedur Pengujian.....	84
4.3.3.	Hasil Pengujian	85
4.3.4.	Analisi Hasil Pengujian	86
4.4.	Pengujian Serial 232.....	86
4.4.1.	Tujuan.....	86
4.4.2.	Peralatan Yang Digunakan.....	86
4.4.3.	Prosedur Pengujian.....	87
4.4.4.	Hasil Pengujian	88
4.4.5.	Analisis Hasil Pengujian	88

4.4.6.	Pengujian SMS Dikirim dan Diterima	88
4.4.7.	Prosedur Pengujian.....	88
4.4.8.	Hasil Pengujian	89
4.4.9.	Analisis Hasil Pengujian	89
4.5.	Pengujian Sistem Secara Keseluruhan	89
4.5.1.	Data Hasil Pengujian.....	91
BAB V	PENUTUP	92
5.1.	Kesimpulan.....	92
5.2.	Saran.....	93

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1	Rangkaian Dalam Optocoupler	6
Gambar 2.2	Penyemat LED Infra Red	7
Gambar 2.3	Simbol Photodiode	8
Gambar 2.4	Gambar Dari Piringan Optocoupler	9
Gambar 2.5	Blok Diagram AT89S8252.....	11
Gambar 2.6	Bentuk Fisik AT89S8252.....	13
Gambar 2.7	Memori Program MCS-51	16
Gambar 2.8	Memori Data MCS-51	18
Gambar 2.9	Memori Data External MCS-51	19
Gambar 2.10	Ruang Special Function Register	23
Gambar 2.11	SPI Master-slave Interconnection	25
Gambar 2.12	Karakteristik Osilator	26
Gambar 2.13	Asynchronous Serial Data Frame (8E1).....	34
Gambar 2.14	Konektor DB9	35
Gambar 2.15	Format data PDU untuk kirim SMS.....	53
Gambar 2.16	Format data PDU untuk Terima SMS	59
Gambar 3.1	Blok Diagram Sistem Monitoring Kecepatan dan Arah Angin .	62
Gambar 3.2	Rangkaian Pemancar LED Infra Red	65
Gambar 3.3	Rangkaian Sensor Arah Angin.....	66
Gambar 3.4	Rangkaian Optocoupler	68
Gambar 3.5	Piringan Untuk Sensor Kecepatan Angin.....	71
Gambar 3.6	Piringan Untuk Sensor Arah Angin	73
Gambar 3.7	Minimum Sistem AT89S8252	76
Gambar 3.8	Rangkaian Interface RS-232	77
Gambar 3.9	Flowchart Mikrokontroller	78
Gambar 3.10	Flowchart Pada PC	79
Gambar 4.1	Rangkaian Pengujian Optocoupler.....	81
Gambar 4.2	Rangkaian Pengujian Sensor Arah Angin	82

Gambar 4.3	Diagram Blok Pengujian Mikrokontroller	84
Gambar 4.4	Rangkaian Pengujian Transfer Data RS-232.....	86
Gambar 4.5	Rangkaian Pengujian SMS Dikirim Dan Diterima	88

DAFTAR TABEL

Tabel 2.1	Fungsi Alternatif Port 1	14
Tabel 2.2	Pembagian Alamat Pada SFR	22
Tabel 2.3	Mode Operasi Timer/Counter 0 dan 1	27
Tabel 2.4	Mode Operasi Timer 2	27
Tabel 2.5	Konfigurasi Pin DB9	36
Tabel 2.6	Tabel Karakteristik Sinyal Listrik Protokol RS-232	38
Tabel 2.7	Pilihan Tabulasi	41
Tabel 2.8	Fungsi Perintah/Tombol	42
Tabel 2.9	Fungsi Tombol Pada Toolbar Standard	44
Tabel 2.10	Fungsi Masing-masing Kontrol Pada Toolbox	45
Tabel 2.11	Tombol Pada Project Explorer	46
Tabel 2.12	Nomor SMS-Centre Operasi Selular Indonesia	54
Tabel 2.13	Jangka Waktu Validitas SMS	57
Tabel 3.1	Pengkodean Arah Angin Menggunakan Pencacah Bilangan Biner	74
Tabel 4.1	Hasil Pengujian Optocoupler	81
Tabel 4.2	Hasil Pengujian Rangkaian Sensor Arah Angin	83
Tabel 4.3	Hasil Pengujian Sistem Mikrokontroler	85
Tabel 4.4	Hasil Pengujian Transfer Data Serial	88
Tabel 4.5	Hasil Pengujian SMS Dikirim dan Diterima	89
Tabel 4.6	Hasil Pengamatan dan Pengujian Sistem Secara Keseluruhan	91

BAB I

PENDAHULUAN

1.1. LATAR BELAKANG

Cuaca sangat berpengaruh terhadap aktifitas manusia. Salah satunya kita melihat para pelaut atau nelayan yang sangat tergantung pada cuaca untuk melakukan kegiatan mereka. Selain para pelaut atau nelayan, masih terdapat banyak bidang-bidang kehidupan lainnya yang sangat tergantung pada kondisi cuaca, seperti bidang penerbangan, bidang meteorologi, pengawasan cuaca dan pertanian. Cuaca yang akan terjadi, selain tergantung pada intensitas dan durasi harian dari energi radian matahari yang diterima di atmosfer diatas permukaan daerah, juga dipengaruhi oleh faktor kecepatan dan arah angin. Angin berfungsi sebagai penyebar suhu secara mendatar.

Melihat fenomena diatas, memantau kondisi arah dan kecepatan angin sehari 24 jam non stop merupakan hal yang tidak mungkin dilakukan oleh manusia tanpa bantuan alat, apalagi dengan kesibukan yang dialami setiap hari. Oleh karena itu sangatlah diperlukan suatu alat yang dapat digunakan untuk mengetahui perubahan kecepatan dan arah angin yang terjadi setiap saat. Dengan semakin pesatnya perkembangan teknologi saat ini, hal itu merupakan hal yang sangat mungkin untuk dilakukan. Dengan Memanfaatkan *sensor* yang sudah ada dipasaran, kita dapat membuat alat yang bisa memberikan informasi tentang kecepatan dan arah angin tersebut.

Dengan perkembangan jaman , kita membutuhkan suatu alat yang dapat membaca dengan cepat dan mudah serta menghasilkan data yang akurat. Maka dibuatlah suatu sistem yang dapat mengukur kecepatan angin dan mengetahui darimana arah datangnya angin tersebut. Informasi yang ditangkap oleh Sensor kemudian akan dikirimkan secara serial ke PC (*Personal Computer*). Data yang diterima oleh PC dari mikrokontroller AT89S8252 akan diproses menggunakan Visual Basic 6.0 dan dikirimkan ke Handphone lain melalui fasilitas SMS.

1.2. RUMUSAN MASALAH

Masalah yang harus diperhatikan dalam pembuatan sistem pendeteksi kecepatan dan arah angin adalah tingkat sensitifitas dari sensor yang digunakan. Karena hal tersebut sangat menentukan kinerja dari sensor itu sendiri. Interface sensor disini difungsikan untuk menangkap informasi dari lingkungan sekitar dan mengubah informasi yang didapat menjadi data-data serial sebelum dikirimkan melalui SMS dengan menggunakan Handphone ke Handphone lain yang *Request*. Sehingga dapat diketahui perubahan kecepatan dan arah Angin. Dengan menggunakan mikrokontroller AT89S8252, Optocoupler sebagai sensor, data akan disimpan dan ditampilkan di PC dan HP sebagai sarana informasi ke HP lain via sms.

Dengan mengacu pada hal diatas, maka judul dari skripsi ini adalah:

“PERENCANAAN DAN PEMBUATAN SISTEM MONITORING KECEPATAN DAN ARAH ANGIN YANG DAPAT DIAKSES VIA SMS BERBASIS AT89S8252”

1.3. TUJUAN

Tujuan dari Tugas Akhir ini adalah untuk merencanakan dan membuat suatu sistem yang dapat memberikan informasi tentang kecepatan dan arah angin. Sehingga akan didapatkan data tentang berapa kecepatan dan darimana arah angin datang secara tepat dan tepat, dimana perubahan tersebut akan ditampilkan di PC dan disambungkan ke Handphone yang akan memberikan informasi ke Handphone lain yang me-*Request* melalui fasilitas SMS.

1.4. BATASAN MASALAH

1. Mikrokontroller yang digunakan AT89S8252.
2. Tidak membahas aspek teknis yang berhubungan dengan meteorologi dan geofisika.
2. Tidak membahas catu daya dan catu daya dianggap konstan.
3. Pesawat HP yang digunakan adalah buatan siemens dengan seri C35.
4. Menggunakan optocoupler sebagai sensor untuk mengukur kecepatan angin.
5. Menggunakan photodiode sebagai sensor arah angin.
6. Menggunakan bahasa pemrograman *Visual Basic 6.0*
7. Tidak membahas perhitungan pembuatan mekanik kincir angin

1.5. METODOLOGI

Tugas Akhir ini bersifat aplikatif, yaitu perencanaan dan pembuatan alat. Langkah-langkah yang perlu dilakukan untuk merencanakan alat yang akan dibuat adalah :

A. Library Research.

Yaitu penelitian yang dilakukan dengan jalan mengumpulkan data dan membaca buku-buku acuan yang ada hubungannya dengan masalah yang dibahas dalam skripsi ini.

B. Tahap perencanaan:

- Perencanaan tiap-tiap blok diagram dengan komponen yang sesuai.
- Perencanaan sistem secara keseluruhan.

C. Tahap pembuatan:

- Penggabungan tiap-tiap blok menjadi suatu kesatuan utuh yang diharapkan nantinya dapat mendukung keseluruhan sistem.
- Pembuatan perangkat lunak pada mikrokontroler AT89S8252 menggunakan *instruction set* yang telah disediakan oleh Intel.

D. Tahap pengujian, yang meliputi:

➤ Pengujian perangkat keras:

Pengujian ini dilakukan untuk mengetahui apakah sistem perangkat keras yang dibuat berjalan dengan baik ataukah tidak. Pengujian dilakukan dengan bantuan perangkat lunak.

➤ Pengujian perangkat lunak:

Pengujian perangkat lunak pada sistem mikrokontroler AT89S8252 dilakukan dengan melakukan simulasi pada program simulasi AT89S8252.

1.6. SISTEMATIKA PENULISAN

Adapun Sistematika Penulisan yang dipakai pada penyusunan Skripsi ini adalah sebagai berikut:

BAB 1 : Pendahuluan

Menguraikan secara singkat tentang latar belakang, tujuan, Batasan Masalah, metodologi dan sistematika pembahasan

BAB 2 : Dasar Teori

Menjelaskan Secara garis besar teori tentang mikrokontroller AT89S8252, IC Maxim 232, Sensor yang dipakai, sistem kerja SMS secara umum dan Hal-hal lain yang diperlukan dalam pembuatan sistem ini.

BAB 3 : Perencanaan dan Implementasi

Menguraikan tentang perencanaan dan implementasi dari sistem yang dibangun, yang meliputi proses pengiriman informasi dari Sensor, hingga proses pengiriman informasi melalui fasilitas SMS.

BAB 4 : Pengujian Sistem

Menguraikan tentang hasil dari pengujian system yang meliputi pengintegrasian perangkat keras (Hardware) dan perangkat lunak (Software), serta proses kerja dari sistem yang dibuat.

BAB 5 : Penutup

Menjelaskan tentang kesimpulan yang diambil dari keseluruhan Tugas Akhir ini dan sara-saran untuk memperbaiki kelemahan dan untuk penyempurnaan dan pengembangan diwaktu mendatang.

BAB II

LANDASAN TEORI

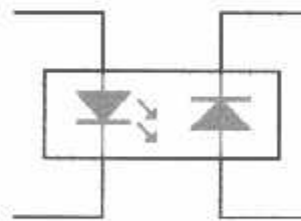
2.1. Pendahuluan

Dalam Pembuatan Sistem pendeteksi kecepatan dan arah angin ini di butuhkan beberapa teori yang mejadi landasan, yaitu teori yang berhubungan dengan komponen-komponen yang digunakan dalam perencanaan dan pembuatan sistem ini.

2.2. Optocoupler

Optocoupler atau optoisolator merupakan sebuah device yang memisahkan antara sumber eksitasi dan beban. Device ini terdiri dari sumber cahaya berupa LED pada sisi input dan photodiode pada sisi output. Diode pemancar cahaya (LED) dan photodiode terpisah oleh celah pada Optocoupler yang akan menghasilkan tegangan AC berupa pulsa-pulsa.^[8, 9]

Bentuk Optocoupler yang dijual-belikan secara umum, terdiri dari kotak plastik yang melindungi photodiode dan LED, seperti pada gambar 2.1 di bawah ini :



Gambar 2.1
Rangkaian dalam Optocoupler^[9]

2.2.1. LED Infra Red

LED Infra Red merupakan cahaya yang intensitasnya digandakan dan difokuskan pada arah tertentu. LED infra Red bersifat koheren dan mempunyai intensitas yang sangat tinggi. Tahun 1960 untuk pertama kalinya sinar laser He-Ne di demontrasikan oleh Javan, Bennet dan Heriot. Setelah itu berkembang sinar laser jenis gas seperti krypton dan sinar laser jenis zat cair seperti laser dyne. Salah satu dari jenis laser adalah laser diode atau disebut juga laser injeksi, karena pemompaannya dilakukan dengan injeksi arus listrik lewat sambungan PN semikonduktornya. Jadi laser ini tidak lain adalah sebuah diode dengan bias maju biasa.

Laser semikonduktor yang pertama diciptakan secara bersamaan oleh tiga kelompok pada tahun 1962. Mereka adalah R.H. Rediker dkk. (Lincoln Lab, MIT), M.I. Nathan dkk. (Yorktown Heights, IBM) dan R.N. Hall dkk. (General Electric Research Lab.). Diode-diode yang digunakan adalah galiun arsenida-flosfida GaAsP (sinar-tampak merah). Proses laser jenis ini mirip dengan kerja LED biasa. Pancaran fotonnya disebabkan oleh bergabungnya kembali elektron dan lubang (hole) di daerah sambungan PN-nya.



Gmabar 2.2
Penyemat Optocoupler (LED-Infra Red)^[3]

2.2.2. Photodiode

Photodiode masih masuk golongan Photoconductor yang terbuat dari silicon dan merupakan gabungan dari silicon jenis-P dan jenis-N, dimana keduanya merupakan perbedaan secara nyata, baik pembuatan maupun kegunaan. Ketika junction P/N dari Photodiode diberi sinar, maka aliran arus listrik yang mengalir pada photodiode sebanding dengan intensitas cahaya yang menyinari.^[8]

Apabila cahaya luar mengenai junction Photodiode yang dibias reverse, maka akan dihasilkan pasangan electron - hole dalam lapisan pengosongan. Makin kuat cahaya, makin banyak jumlah pembawa yang dihasilkan cahaya dan makin besar arus reverse. Karena itu Photodiode merupakan detector cahaya yang baik sekali.^[8]



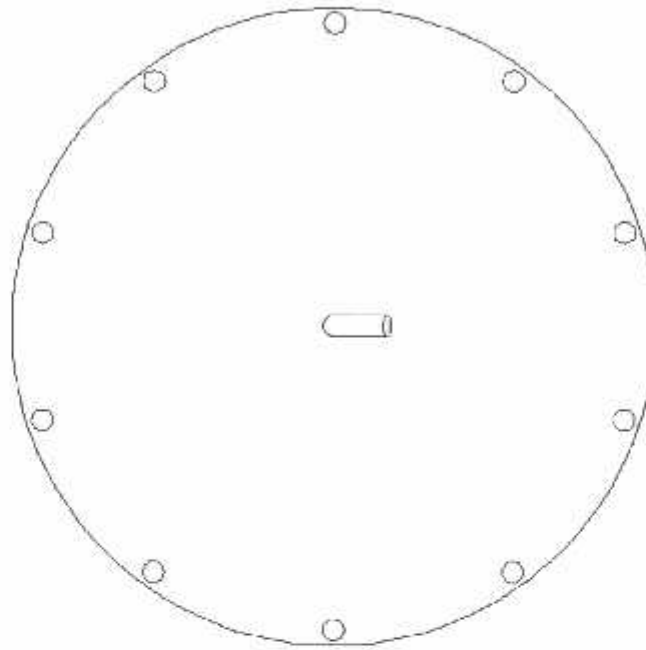
Gambar 2.3 Simbol Photodiode^[8]

2.2.3. Piringan Optocoupler

Pada Celah antara LED dan Phototransistor dipasang piringan yang berbentuk silindris dan mempunyai lubang n buah. Karena keluaran Phototransistor yang terkena sinar dari LED berupa sinar listrik yang sesuai dengan intensitas cahaya pantulan LED yang ditangkap oleh Phototransistor tersebut, maka dalam satu putaran akan dihasilkan n buah pulsa.

Frekuensi dari pulsa tersebut sebesar: ^[1]

$$f = (n \omega / 60) \text{ Hz.}$$



Gambar 2.4
Gambar dari piringan Optocoupler^[1]

2.3. Mikrokontroler AT89S8252

Mikrokontroler AT89S8252 merupakan versi CHMOS dari AT89S8252 yaitu versi NMOS. NMOS merupakan kependekan dari *N-Channel Metal Oxide Silicon* dan kompatibel dengan MCS-51 mikrokomputer yang merupakan produksi dari ATMEL. Seri AT89S8252 adalah mikrokontroller yang membutuhkan daya rendah, memiliki kemampuan yang tinggi, dan merupakan mikrokomputer 8 bit yang dilengkapi 8K byte Flash PEROM (*Programmable and Erasable Read Only Memory*) yaitu ROM yang dapat ditulis ulang atau

dihapus menggunakan sebuah perangkat programmer. Serta terdapat EEPROM *internal* sebesar 2K Byte.

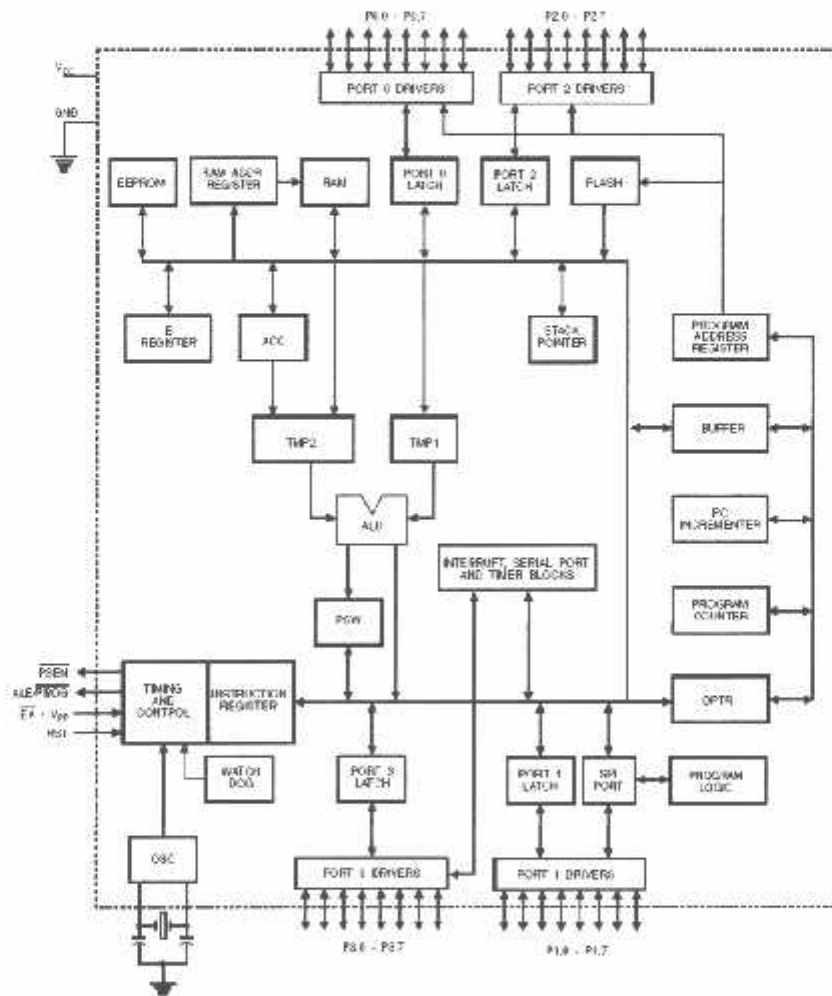
Flash PEROM dalam AT89S8252 menggunakan *Atmel's High-Density Non Volatile Technology* yang mempunyai kemampuan untuk ditulis ulang hingga 1000 kali dan berisikan perintah *standard* MCS-51. Selain itu juga dilengkapi RAM *internal* sebesar 256 byte. Dalam sistem Mikrokontroller terdapat dua hal yang mendasar, yaitu: perangkat keras dan perangkat lunak yang keduanya saling terkait dan mendukung. Mikrokontroller ini digunakan untuk beberapa keperluan mulai dari komersial, industri, otomotif, dan militer.

2.3.1. Fitur Mikrokontroller AT89S8252

Fitur Mikrokontroller AT89S8252 adalah sebagai berikut :

- Kompatibel dengan mikrokontroller MCS-51
 - 8K byte In-System Programmable Downloadable Flash Memori
 - SPI Serial Interface for Program Downloading
 - Endurance: 1,000 Write/Erase Cycles
 - 2K byte EEPROM
 - Edurance: 100,000 Write/Erase Cycles
 - 3 Level program memori lock
 - Memory 256 x 8 bit *Internal* RAM
 - 32 Port I/O Lines yang dapat dipakai semua
 - 3 buah Timer/Counter 16 bit
 - Nine Interrupt Source
-

- Programmable UART (serial port)
- SPI Serial Interface
- Programmable Watchdog Timer
- Dual Data Pointer
- Frekuensi kerja 0 sampai 24 MHz
- Tegangan Operasi 4 Volt sampai 6 Volt
- *Power-Off Flag*.^[6]



Gambar 2.5. Blok Diagram AT89S8252^[6]

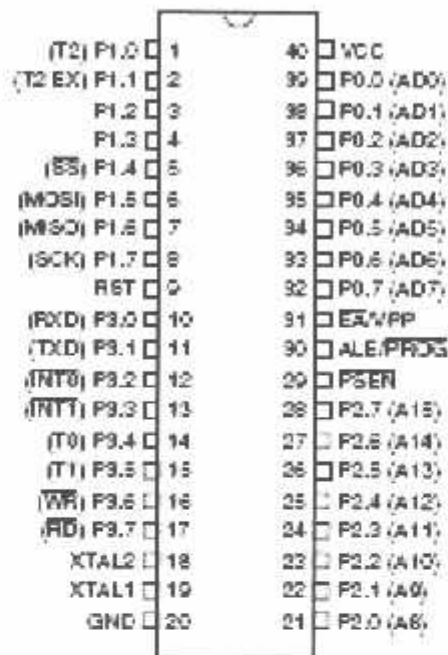
2.3.2. Arsitektur AT89S8252

Arsitektur Mikrokontroller AT89S8252 adalah sebagai berikut:

- 8 bit CPU (*Central Processing Unit*) dengan register A dan B.
 - 16 bit *Program Counter* (PC) dengan *Data Pointer* (DPTR).
 - 8 bit program status word (PSW).
 - 8 bit stack pointer (SP).
 - Internal EPROM dan ROM dari 0 sampai 4 Kb.
 - 256 byte *Internal RAM*
 - ⇒ 4 register bank masing-masing 8 register.
 - ⇒ 16 byte yang dapat dialamatkan pada *bit level*.
 - ⇒ 208 byte *memory general purpose memory data*.
 - 32 pin *input/output* tersusun sebagai 4 port masing-masing 8 bit (P0 – P3).
 - 3 buah Timer (T0 & T1) dengan masing-masing 16-bit timer/counter.
 - Data serial *receiver/transmitter full duplex* yaitu SBUF.
 - *Control register* antara lain TCON, TMOD, SCON, PCON, IP dan ME.
 - 2 eksternal dan 3 internal sumber *interrupt*.
 - Rangkaian *oscillator* dan *clock*.
-

2.3.3. Konfigurasi Kaki-kaki MCU AT89S8252

Berikut ini adalah bentuk fisik dari AT89S8252:



Gambar 2.6. Bentuk fisik AT89S8252^[6]

Fungsi-fungsi dari tiap-tiap pin sebagai berikut :

➤ **Pin 40: Vcc**

Dihubungkan dengan sumber tegangan + 5 V.

➤ **Pin 20: GND**

Dihubungkan dengan Ground rangkaian atau media pentanahan.

➤ **Pin 32-39 : Port 0 (P0.0 – P0.7)**

Port 0 merupakan port I/O 8 bit dua arah. Port ini digunakan sebagai multipleks bus alamat rendah dan bus data selama pengaksesan ke memori luar.

➤ **Pin 1-8 : Port 1 (P1.0 – P1.7)**

Port 1 dapat difungsikan sebagai masukan atau keluaran dan bekerja baik untuk operasi bit maupun *byte*, tergantung dari pengaturan program yang dibuat.

Tabel 2.1 Fungsi Alternatif Port 1

Port Pin	Fungsi
P1.0	T2 (masukan eksternal untuk Timer/Counter 2)
P1.1	T2EX (Timer/Counter 2 capture/reload trigger dan kontrol arah
P1.2	-
P1.3	-
P1.4	SS (slave port select input)
P1.5	MOSI (master data output, slave data input untuk chanel SPI)
P1.6	MISO (maser data input, slave data output untuk chanel SPI)
P1.7	SCK (master clock output, slave clock input untuk chanel SPI)

➤ **Pin 21-28 : Port 2 (P2.0 – P2.7)**

Port 2 merupakan port input-output dengan internal pull-up. Port ini dapat digunakan sebagai alamat bus baik *byte* tinggi selama adanya akses ke memori program atau memori data luar.Mengeluarkan address tinggi selama pengambilan (fetching) program memory external. Selama pengaksesan ke external data memory, port 2 mengeluarkan isi P2 SFR. Menerima address dan beberapa control selama pemrograman dan verifikasi.

➤ **Port 3 (P3.0 – P3.7)**

Port 3 mempunyai fungsi sebagai I/O juga mempunyai fungsi khusus sebagai berikut :

- RD (P3.7), sinyal pembacaan memori data luar.

- WR (P3.6), sinyal penulisan memori data luar.
- T1 (P3.5), masukan dari pewaktu/pencacah 1.
- T0 (P3.4), masukan dari pewaktu/pencacah 0.
- INT1 (P3.3), masukan interrupt 1.
- INT0 (P3.2), masukan interrupt 0.
- TXD (P3.1), keluaran pengiriman data untuk serial *port* (*asynchronous*) atau sebagai keluaran *clock* (*sybchronous*).
- RXD (P3.0), masukan data serial atau sebagai keluaran data.

➤ **Pin 9 : RST/VPD.**

Merupakan pin input yang aktif jika pin aktif tinggi selama dua siklus mesin maka ketika osilator bekerja akan mereset peralatan.

➤ **Pin 30 : ALE (*Address Latch Enable*).**

Pin ALE (aktif tinggi) mengeluarkan pulsa output untuk menyangga (*latch*) satu byte alamat rendah selama mengakses ke memori eksternal. ALE dapat mengendalikan 8 beban TTL. Pin ini juga merupakan input pulsa program yang aktif rendah selama pemrograman EPROM. Pada operasi normal, ALE dikeluarkan pada suatu kecepatan yang konstan yaitu 1/6 dari frekuensi osilator dan dapat digunakan untuk *timing* eksternal atau untuk tujuan membuat *clock*.

➤ **Pin 29 : PSEN (*Program Strobe Enable*).**

Pin ini aktif rendah yang merupakan *strobe* pembacaan ke program memori eksternal.

➤ **Pin 18-19 : XTAL1&XTAL2.**

Pin XTAL1 merupakan pin input ke penguat osilator pembalik dan XTAL2 merupakan pin output dari penguat osilator pembalik.

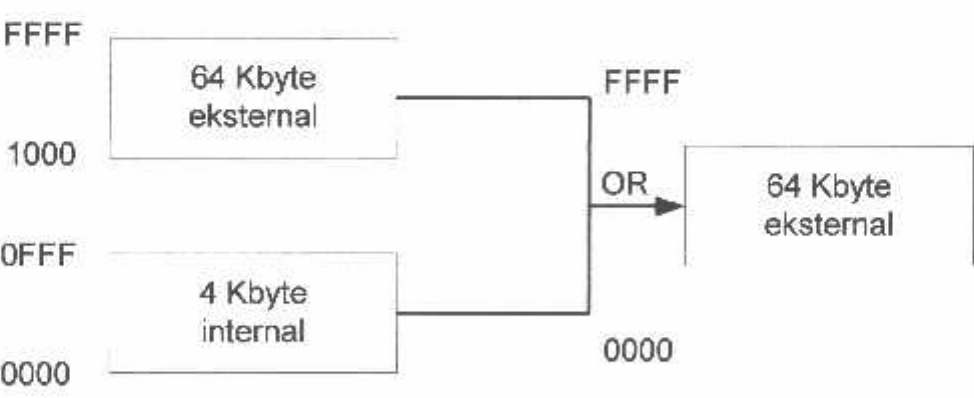
➤ **Pin 30 : EA,VPP (*External Access/Programming Supply Voltage*).**

Pin EA di Vcc agar 89S8252 dapat mengakses kode mesin dari program memori.

2.3.4. Organisasi Memori MCU AT89S8252

Mikrokontroler AT89S8252 termasuk keluarga MCS-51 yang memiliki memori program dan memori data yang terpisah. Pemisahan ini dilakukan secara logika schingga CPU dapat mengakses sampai 64 *Kbyte* memori program dan 64 *Kbyte* memori data. Lebar memori data internal adalah 8 bit dan 16 bit (register PC dan register DPTR).

2.3.4.1. Memori Program



Gambar 2.7. Memori program MCS-51^[5]

Memori program menggunakan alamat sepanjang 64 *Kbyte* dengan 4 *Kybe* (alamat \$0000 sampai dengan \$0FFF) yang merupakan memori internal sehingga 60 *Kbyte* merupakan memori eksternal. Dapat menggunakan 64 *Kbyte* memori eksternal sebagaimana yang ditunjukkan pada gambar memori data.

Memori program merupakan tempat penyimpanan data permanen. Memori program lebih dikenal dengan nama *Read Only Memory* (ROM). Data dalam ROM tidak akan terhapus meskipun catu daya dimatikan atau dikenal sebagai sifat *non-volatile*. Karena sifatnya yang demikian ROM dapat digunakan untuk menyimpan program.

Ada beberapa tipe ROM, antara lain :

- ROM (*Read Only Memory*)

Merupakan memori yang sudah diprogram oleh pabrik (ROM murni).

- PROM (*Programmable Read Only Memory*)

Merupakan memori yang dapat diprogram oleh pemakai tetapi tidak dapat diprogram ulang.

- EPROM (*Erasable Programmable Read Only Memory*)

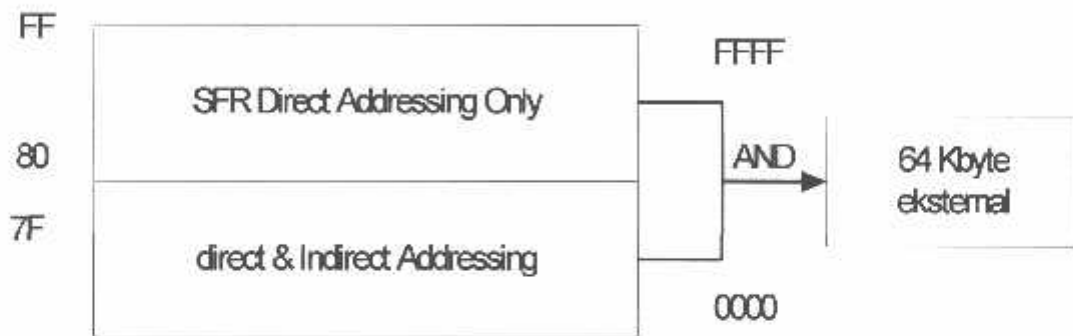
Merupakan PROM yang dapat diulang. ROM ini juga terdapat pada mikrokontroler 8751, hal ini ditandai dengan adanya jendela kaca pada konstruksi IC 8751 yang digunakan untuk menghapus atau memperbaiki program yang sudah ada.

- EEPROM (*Electrical Erasable Programmable Read Only Memory*)

Pada prinsipnya hampir sama dengan EPROM, tetapi perbedaanya terletak pada pengosongan atau penghapusan program. Untuk EPROM

dapat dihapus dengan menggunakan sinar *ultra violet*, sedangkan pada EEPROM pengisian program dapat dilakukan langsung atau menumpuk program lama dengan program yang baru. EEPROM lebih fleksibel dibandingkan EPROM.

2.3.4.2. Memori Data

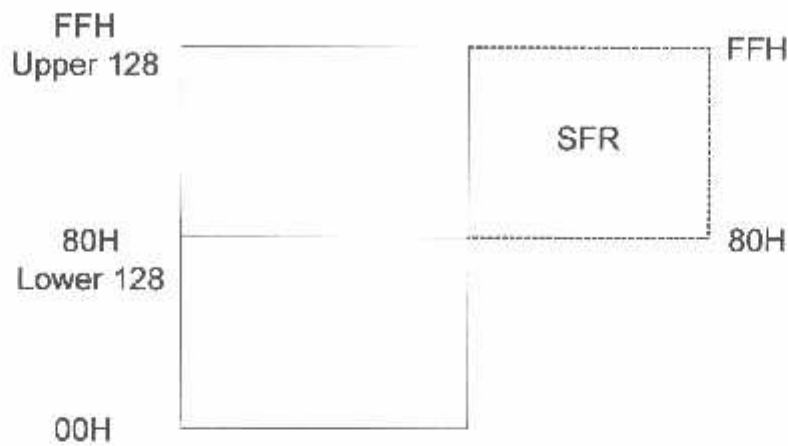


Gambar 2.8. Memori data MCS-51^[5]

Memori data merupakan tempat penyimpanan data yang bersifat sementara atau *volatile*. Dengan kata lain data akan hilang bila tidak dicatu. Memori data lebih dikenal dengan nama RAM (*Random Access Memory*), yaitu dapat dilakukan pembacaan dan penulisan data alamat yang tersedia.

Memori MCS-51 mempunyai 128 *bytes* - 256 *byte* RAM internal ditambah sejumlah register fungsi khusus atau *Special Function Register* (SFR). (*Advanced Mikrodevidees*, 1988.1.6). Selain mempunyai memori internal, MCU AT89S8252 mempunyai memori eksternal yang memiliki pengalamatan sampai 64 *Kbytes*.

Pada keluarga mikrokontroler MCS-51, ruang memori data eksternal terbagi menjadi 3 blok yang disebut *lower 128*, *upper 128* dan ruang SFR, sebagaimana ditunjukkan pada gambar di bawah ini.



Gambar 2.9. Memori Data Eksternal ^[5]

Pada *lower 128* lokasi memori dibagi menjadi 3 bagian :

1. Register bank 0 – 3

Lokasi bank register dimulai dari alamat 00h – 1h yang terdiri dari 32 *bytes*. Register bank ini terdiri dari 4 buah register 8 bit yang dapat dipilih melalui pengaturan *program status word* register.

2. Bit Addressing

Terdiri dari 16 *bytes* yang dimulai dari 20h – 2fh. Masing-masing dari 128 bit lokasi ini dapat dialamati secara langsung yaitu dari \$00h sampai \$7fh.

3. Scratch Pad Area

Lokasi dari alamat \$30h sampai \$7fh atau sebanyak 80 *bytes* yang dapat digunakan sebagai alamat bagi RAM.

Pada 128 *bytes* atas (*upper* 128) ditempati oleh register yang mempunyai fungsi khusus yang disebut dengan *Special Function Register* (SFR). Ruang dari register fungsi khusus ini adalah dari 80h sampai FFh. Berikut ini adalah contoh isi vector alamat pada *Special Function Register*.

- Akumulator (Acc) atau register A dan register B.

Kedua register tersebut digunakan untuk operasi perkalian dan pembagian.

- *Program Status Word*

Register ini meliputi bit-bit : CY (*Carry*), AC (*Auxillary Carry*), FO sebagai flag, RS0 dan RS1 untuk pemilih register bank, OV (*Over Flow*), dan *parity flag*.

- *Stack Pointer* (SP)

SP merupakan register yang digunakan untuk penunjuk alamat. Register ini berguna apabila digunakan suatu *routine* pada program utama.

- *Data Pointer High* (DPH) dan *Data Pointer Lower* (DPL)

DPTR adalah register yang digunakan untuk pengalamatan tidak langsung. Register ini digunakan untuk mengakses memori program baik internal maupun eksternal. DPTR dikontrol oleh 2 buah register 8 bit yaitu DPH dan DPL.

- *Port 0, Port 1, Port 2, Port 3*

Pada keluarga 8051 masing-masing *port* dapat dialamati langsung baik secara *byte* atau bit. Masing-masing *port* merupakan *port bi-directional* (*input/output*) :

1. *Port 0* digunakan sebagai pengalamatan memori dari luar.
2. *Port 1* digunakan sebagai I/O dari mikrokontroler.
3. *Port 2* digunakan sebagai pengalamatan memori dari luar.
4. *Port 3* berisi sinyal kontrol seperti *interrupt serial*, WR, dan RD.

- *Register Prioritas Interrupt (Interrupt Priority Register /IP).*

Merupakan register yang berisi bit-bit untuk mengaktifkan prioritas dari suatu *interrupt* yang ada pada mikrokontroler pada taraf yang diinginkan.

- *Interrupt Enable Register*

Merupakan register yang berisi bit-bit untuk menghidupkan atau mematikan sumber-sumber *interrupt*.

- *Timer/Counter Control Register*

TCON merupakan register yang berisi bit-bit memulai atau menghentikan pencacah atau pewaktu.

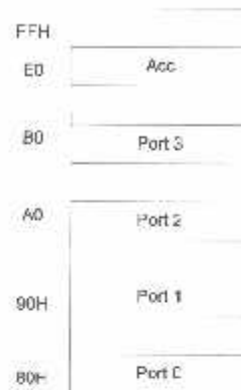
- *Serial Control Buffer*

Register ini digunakan untuk menampung data masukan (SBUF *in*) atau keluaran (SBUF *out*) dari *serial port*.

SYMBOL	NAME	ADDRESS
ACC	ACCUMULATOR	0E0H
B	B REGISTER	0F0H
PSW	PROGRAM STATUS WORD	0D0H
SP	STACK POINTER	81H
DPTR	DATA POINTER 2 BYTE	
DPL	LOW BYTE	82H
DPH	HIGH BYTE	83H
P0	PORT 0	80H
P1	PORT 1	90H
P2	PORT 2	0A0H
P3	PORT 3	080H
IP	INTERRUPT PERIORITY CONTROL	088H
IE	INTERRUPT ENABLE CONTROL	0ABH
TMOD	TIMER/COUNTER MODE CONTROL	89H
TCON	TIMER/COUNTER CONTROL	88H
+TCON	TIMER/COUNTER 2 CONTROL	0CBH
TH0	TIMER/COUNTER 0 HIGH CONTROL	8CH
TL0	TIMER/COUNTER 0 LOW CONTROL	8DH
TH1	TIMER/COUNTER 1 HIGH CONTROL	8DH
TL1	TIMER/COUNTER 1 LOW CONTROL	8CH
TH2	TIMER/COUNTER 2 HIGH CONTROL	0CDH
TL2	TIMER/COUNTER 2 LOW CONTROL	0CCH
RCAP2H	T/C 2 CAPTURE REG. HIGH BYTE	0CBH
+RCAP2L	T/C 2 CAPTURE REG. LOW BYTE	0CAH
SCON	SERIAL CONTROL	98H
SBUF	SERIAL DATA BUFFER	99H
PCON	POWER CONTROL	87H

Tabel 2.2 Pembagian alamat pada SFR^[6]

Adapun diagram blok dari SFR adalah sebagai berikut :



Gambar 2.10. Ruang *Special Function Register*^[5]

2.3.4.3. SFR Tambahan Pada AT89S8252

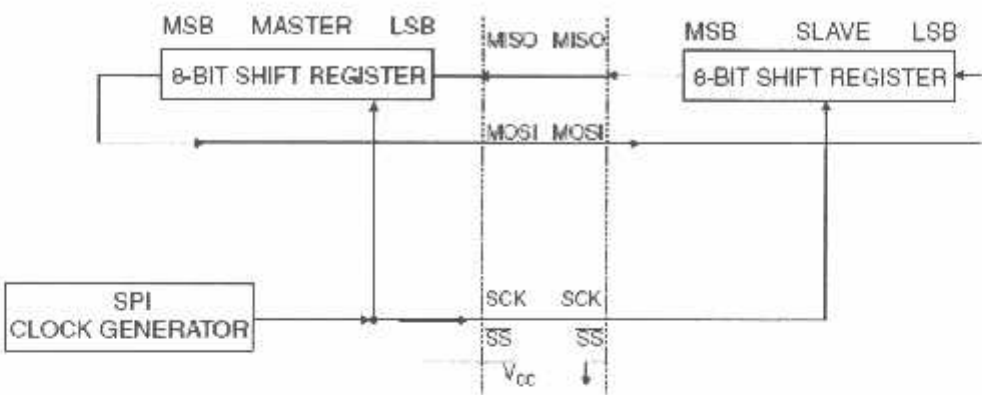
Selain memiliki SFR (*Special Function Register*) seperti halnya pada MCS-51, mikrokontroller AT89S8252 memiliki tambahan SFR. Hal ini tak lain karena adanya fitur tambahan pada mikrokontroller AT89S8252.

SFR tambahan ini meliputi : T2CON (Timer 2 Register dengan alamat 0C81H), T2MOD (Timer 2 Mode dengan alamat 0C9H), WMCON (Watchdog and Memory Control Register dengan alamat 96H), SPCR (SPI Control Register dengan alamat D5H), SPSR (SPI Status Register dengan alamat AAH), SPDR (SPI Data Register dengan alamat 86H)

2.3.4.3.1. SFR Untuk Timer 2

Mikrokontroler AT89S8252 terdapat tambahan sebuah Timer/Counter yang diberi nama Timer 2 (sehingga AT89S8252 memiliki 3 Timer/Counter yaitu Timer/Counter 0, Timer/Counter 1, Timer/Counter 2). Pada Timer/Counter 2 ini dikendalikan oleh SFR yang bernama T2CON (Timer 2 Control), T2MOD (Timer

Gambar berikut menunjukkan hubungan antara CPU master dan slave.

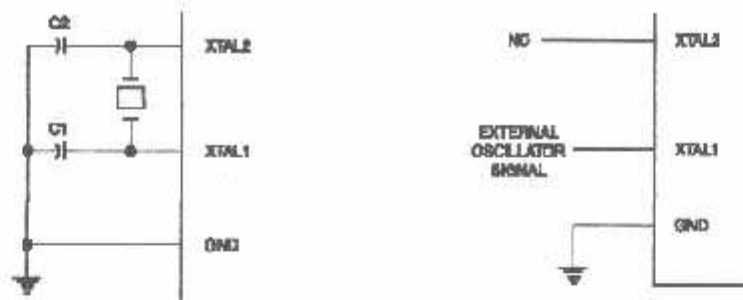


Gambar 2.11. SPI Master-slave Interconnection^[6]

2.3.4.3.4. Osilator

Jantung dari AT89S8252 adalah rangkaian yang membangkitkan pulsa clock yang mensinkronkan semua operasi internal. Mikrokontroler AT89S8252 memiliki osilator internal (*on-chip oscillator*) yang dapat digunakan sebagai sumber pewaktu (clock) bagi CPU. Untuk menggunakan osilator internal diperlukan sebuah kristal atau resonator keramik antara pin XTAL₁ dan XTAL₂ dan sebuah kapasitor ke ground. XTAL₂ dan XTAL₁ secara berurutan merupakan input dan output dari sebuah inverting amplifier yang dapat dikonfigurasi penggunaannya sebagai *on-chip oscillator* seperti yang ditunjukkan pada gambar 2.12a.

Untuk memberikan IC AT89S8252 sumber clock eksternal, maka pin XTAL₂ dibiarkan tidak berhubungan dengan sumber clock eksternal dan XTAL₁ dihubungkan dengan sumber clock eksternal seperti pada gambar 2.12b.

a). *Oscillator Connector*b). *External Clock Drive Configuration***Gambar 2.12 Karakteristik Osilator^[6]**

2.3.4.3.5. Timer dan Counter

Dalam mikrokontroler AT89S8252 terdapat 3 buah pewaktu/pencacah (Timer/Counter) 16-bit yang dapat diatur melalui perangkat lunak, yaitu pewaktu/pencacah 0 dan pewaktu/pencacah 1. Timer/Counter ini diatur oleh SFR (*Special Function Register*) yaitu Timer/Counter Control (TCON dengan alamat 88_H) dan Timer/Counter Mode Control (TMOD dengan alamat 89_H). Selain itu nilai byte bawah dan byte atas dari Timer/Counter disimpan dalam register TL dan TH.

Jika difungsikan sebagai Timer, maka akan menggunakan sistem clock sebagai sumber masukan pulsanya. Jika sebagai Counter (pencacah), maka akan menggunakan pulsa dari luar (eksternal) sebagai masukan pulsanya. Pada port 3 terdapat fungsi khusus yaitu T0 (masukan luar untuk Timer/Counter 0) dan T1 (masukan luar untuk Timer/Counter 1). Pemilihan mode Timer/Counter dikontrol oleh register TMOD. Dengan memberikan nilai tertentu pada register TMOD,

dapat dipilih mode operasi untuk Timer/Counter 0 dan Timer/Counter 1 seperti terlihat dalam tabel.

Tabel 2.3. Mode Operasi Timer/Counter 0 dan 1^[4]

Mode	Timer/Counter 0	Timer/Counter 1
0	13-bit Timer	13-bit Timer
1	16-bit Timer	16-bit Timer
2	8-bit auto-reload	8-bit auto-reload
3	Dua 8-bit Timer	Tidak bekerja

Pada mikrokontroler terdapat tambahan Timer 2. Timer yang lain adalah Timer 0 dan Timer 1. Timer 2 ini merupakan Timer/Counter 16-bit dan memiliki 3 mode operasi yaitu *capture*, *auto-reload (up-down counting)* dan *baund rate generator*. Untuk memilih mode ini dilakukan dengan mengatur bit pada *SLR T2CON* (Timer 2 Control Register). Timer 2 ini terdiri dari 2 buah Timer 8-bit register yaitu TH2 dan TL2. Pada fungsi Timer, register TL2 dinaikkan (*increament*) tiap siklus mesin. Karena siklus mesin terdiri dari 12 periode osilasi, maka *count rate* menjadi 1/12 dari frekuensi osilator. Sedangkan pada fungsi Counter, register dinaikkan berdasarkan tanggapan adanya transisi tinggi ke rendah pada pin yang bersesuaian (dalam hal ini pin T2 atau P1.0). Tabel berikut menunjukkan mode operasi yang dapat dijalankan pada Timer 2.

Tabel 2.4 Mode Operasi Timer 2^[4]

RCLK + TCLK	CP/RL2	TR2	MODE
0	0	1	16-bit auto-reload
0	1	1	16-bit capture
1	X	1	Baund Rate Generator
X	X	0	Off

Keterangan^{||}:

RCLK = *Receive Clock Enable*. Jika diset menyebabkan serial port menggunakan pulsa *overflow* Timer 2 sebagai detak penerimaan pada serial port. Jika RCLK = 0, maka Timer 1 yang digunakan

TCLK = *Transmit Clock Enable*. Jika diset menyebabkan serial port menggunakan pulsa *overflow* Timer 2 sebagai detak pengiriman. Jika TCLK = 0, maka pulsa *overflow* Timer 1 yang digunakan

CP/RL2 = *Pemilihan Capture/Reload*. Jika diset maka proses *capture* yang terjadi sedangkan jika bit ini dideclear maka proses *reload*

2.3.4.3.6. Sistem Interrupt

AT89S8252 memiliki 6 buah sumber interupsi, 2 eksternal interupsi (INT0 dan INT1), 3 Timer interupsi (Timer 0, 1 dan 2) dan satu serial port interupsi.

INT0 = interrupt pada P3.2 (kaki 12)

INT1 = interrupt pada P3.3 (kaki 13)

Timer 0 = Timer pada P3.4 (kaki 14)

Timer 1 = Timer pada P3.5 (kaki 15)

Port serial = jika pengiriman/penerimaan suatu frame telah lengkap

Saat terjadinya interupsi, mikrokontroler secara otomatis akan menuju *subroutine* pada alamat tersebut. Setelah interupsi servis selesai dikerjakan, mikrokontroler akan mengerjakan program semula. Dua sumber eksternal adalah INT0 dan INT1, kedua interupsi eksternal akan aktif, transisi tergantung isi IT0

dan IT1 pada register TCON. Interrupt T0 dan T1 aktif pada saat Timer yang sesuai mengalami *roll over*. Interupsi serial akan dibangkitkan dengan melakukan operasi OR pada R1 dan T1 tiap-tiap sumber interupsi dapat *enable* atau *disable* secara *software*. Tingkat prioritas semua sumber interupsi dapat diprogram sendiri-sendiri dengan set atau *clear bit* pada SFR IP (*interrupt priority*). Register yang akan berperan dalam mengatur aktif tidaknya interupsi adalah Interrupt Enable Register.

2.3.5. Metode Pengalamatan

Metode pengalamatan yang digunakan pada MCS-51 terbagi menjadi dua jenis, yaitu pengalamatan langsung dan pengalamatan tidak langsung. (Moh. Ibnu Malik, 1997:36)

φ *Pengalamatan Tak Langsung*

Operasi pengalamatan tak langsung menunjukkan ke sebuah register yang berisi lokasi alamat memori yang akan digunakan dalam suatu operasi. Lokasi yang nyata tergantung dari isi register saat instruksi dijalankan. Untuk melakukan pengalamatan tak langsung digunakan simbol @. Misalnya :

- ADD A, @R0 : tambahkan isi R0 dengan Acc dan hasilnya di Acc
 - DEC @R1 : kurangi isi dari alamat R1
-

φ *Pengalamatan Langsung*

Pengalamatan langsung dilakukan dengan memberikan nilai ke suatu register secara langsung. Untuk melakukan hal tersebut digunakan tanda #.

Misalnya:

- `MOV A,#01H` : isi Acc dengan data 01H
- `MOV DPTR,#19H` : isi DPTR dengan data 19H

Pengalamatan data langsung dari 0 sampai 127 akan mengakses RAM internal, sedangkan pengalamatan dari 128 sampai 255 akan mengakses register perangkat keras. Misalnya :

- `MOV P3,A` : pindahkan isi Acc ke alamat *Port* 3 (B0H)
- `INC 50` : naikan lokasi 50 (desimal) dalam memori

2.3.6. Bahasa Assembler MCS-51

Bahasa assembler digunakan dalam setiap operasi CPU dalam bentuk bahasa simbol yang disusun berurutan dalam pernyataannya. Masing-masing pernyataan akan diterjemahkan ke dalam instruksi bahasa mesin atau sering disebut *operation code/opcode*. Dalam penulisan bahasa mesin ini, terdapat berbagai macam kelompok instruksi, diantaranya :

- **Perpindahan Data**

Instruksi ini digunakan untuk memindahkan data antar register, memori, register-memori, antar muka register dan antar muka memori.

Contonya: `MOV A,R0` : memindahkan isi register R0 ke Acc.

MOV A,@R0 : memindahkan isi alamat R0 ke Acc.

- Operasi Aritmatika

Instruksi ini melaksanakan operasi aritmatika yang meliputi penjumlahan, pengurangan, perkalian, maupun pembagian.

Contohnya: ADD A,#data : menambah Acc dengan data.

ADC A,#data : menambah Acc dengan data dan carry.

INC R6 : menambah isi R6 dengan 1.

DEC R7 : mengurangi isi R5 dengan 1.

MUL AB : mengalikan isi Acc dengan isi register B.

DIV AB : membagi isi Acc dengan isi register B.

- Operasi Percabangan

Instruksi ini mengubah urutan normal pelaksanaan suatu program untuk melaksanakan pada lain tempat yang kita perlukan pada saat itu.

Contohnya:

1. CJNE (*Compare Jump Not Equal*)

Instruksi ini membandingkan isi lokasi memori tertentu dengan isi Acc, jika sama instruksi ini selanjutnya akan dieksekusi. Jika tidak sama eksekusi akan kembali ke alamat kode yang telah ditunjuk.

2. JB (*Jump if Bit Set*)

Instruksi ini akan menguji suatu alamat bit isi satu, eksekusi akan menuju ke alamat kode dan jika tidak instruksi akan dilanjutkan.

3. JNB (*Jump if Bit Not Set*)

Instruksi ini menguji suatu alamat bit. Jika berisi 0 maka eksekusi akan menuju ke alamat kode. Jika berisi 1 maka instruksi selanjutnya yang akan dieksekusi.

2.4. Komunikasi Serial

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

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

Ditinjau dari arah komunikasi data, dikenal ada tiga cara, yaitu: komunikasi *simplex*, *half duplex* dan *full duplex*.

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

Yang dimaksud dengan komunikasi secara serial adalah komunikasi dengan memanfaatkan hanya satu saluran sinyal untuk pengiriman dan penerimaan data. Untuk data digital yang merupakan serangkaian bit maka data dikirimkan bit per bit. Pengiriman akan dimulai dari LSB (*Least Significant Bit*), dan diakhiri dengan MSB (*Most Significant Bit*). Setiap karakter yang dikirimkan disusun sesuai dengan suatu urutan bit tertentu.

2.5. Metode Transmisi Serial

Ada dua metode transmisi serial yang digunakan yaitu:

1. Komunikasi Serial Sinkron (*Synchronous Serial Communication*)

Pada komunikasi metode ini, pengiriman dan penerimaan akhir komunikasi adalah secara sinkron atau serempak menggunakan *clock* yang tepat, waktu dalam periode terpisah tiap bit dengan pengecekan *clock* pada penerimaan akhir. Salah satu aspek penting pada metode ini jika akhir dari komunikasi kehilangan sinyal *clock*, maka komunikasi akan *terminasi*.

2. Komunikasi Serial Asinkron (*Asynchronous Serial Communication*)

Komunikasi metode ini sinyal *clock* pengirim dan penerima data tidak harus *sinkron* untuk setiap bit data yang ditransmisikan. Karena itu sinyal *clock* tidak perlu dikirimkan. Hal ini bukan berarti tidak ada usaha *sinkronisasi* sama sekali antara pengirim dan penerima data. *Sinkronisasi* dilakukan dengan pengiriman sebuah bit awal (*bit start*) sebelum serangkaian bit data dikirimkan. Agar selama penerimaan data, *clock* penerima tidak melesat jauh, maka jumlah bit

Yang dimaksud dengan komunikasi secara serial adalah komunikasi dengan memanfaatkan hanya satu saluran sinyal untuk pengiriman dan penerimaan data. Untuk data digital yang merupakan serangkaian bit maka data dikirimkan bit per bit. Pengiriman akan dimulai dari LSB (*Least Significant Bit*), dan diakhiri dengan MSB (*Most Significant Bit*). Setiap karakter yang dikirimkan disusun sesuai dengan suatu urutan bit tertentu.

2.5. Metode Transmisi Serial

Ada dua metode transmisi serial yang digunakan yaitu:

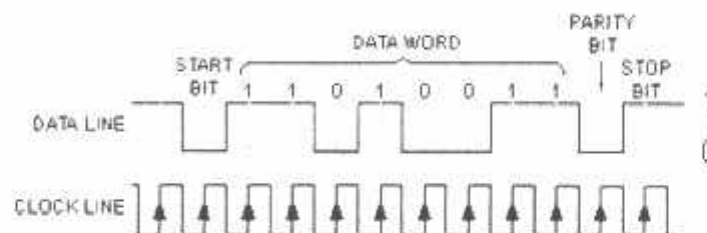
1. Komunikasi Serial Sinkron (*Synchronous Serial Communication*)

Pada komunikasi metode ini, pengiriman dan penerimaan akhir komunikasi adalah secara sinkron atau serempak menggunakan *clock* yang tepat, waktu dalam periode terpisah tiap bit dengan pengecekan *clock* pada penerimaan akhir. Salah satu aspek penting pada metode ini jika akhir dari komunikasi kehilangan sinyal *clock*, maka komunikasi akan *terminasi*.

2. Komunikasi Serial Asinkron (*Asynchronous Serial Communication*)

Komunikasi metode ini sinyal *clock* pengirim dan penerima data tidak harus *sinkron* untuk setiap bit data yang ditransmisikan. Karena itu sinyal *clock* tidak perlu dikirimkan. Hal ini bukan berarti tidak ada usaha *sinkronisasi* sama sekali antara pengirim dan penerima data. *Sinkronisasi* dilakukan dengan pengiriman sebuah bit awal (*bit start*) sebelum serangkaian bit data dikirimkan. Agar selama penerimaan data, *clock* penerima tidak melesat jauh, maka jumlah bit

data dalam satu kali pengiriman harus dibatasi. Umumnya batasan maksimal 8 bit data (*1 byte*) dalam satu kali pengiriman. Oleh sebab itu komunikasi serial *asinkron* biasanya digunakan pada perangkat yang menghasilkan data dengan laju rendah. Kemudian untuk menjamin adanya transisi sinyal minimal satu kali diantara dua pengiriman rangkaian bit data, maka pengiriman data perlu diakhiri dengan bit penutup (*bit stop*) yang memiliki polaritas yang berbeda dengan bit pembuka.



Gambar 2.13
Asynchronous Serial Data Frame (8E1)^[14]

Bagian penting yang lain pada setiap sinyal serial *asinkronous* adalah *bit rate* pada data yang dikirimkan. Kecepatan data yang diterima didasarkan pada kecepatan minimum 300bps (*bit per second*).

2.6. Interface RS-232

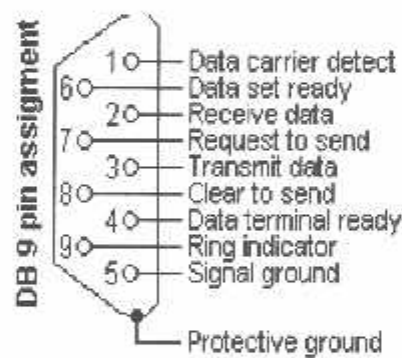
Interface

Interface jika diterjemahkan, mungkin akan mengandung arti sebagai antarmuka. Dalam suatu sistem komputer apabila ada alat yang dihubungkan akan dapat mengganggu komputer tersebut. Sinyal-sinyal yang tidak dikenal akan merusak rangkaian komputer. Untuk itu, dibutuhkan suatu *Interface* yang

berfungsi sebagai antarmuka dua lingkungan yang berbeda. *Interface* merupakan semacam pintu gerbang dikenal sebagai *I/O Port*.

a). Konfigurasi Pin

Sebuah komputer atau peralatan lain akan dapat bertukar informasi melalui *Interface port I/O serial* yang disebut *Interface RS 232*. RS 232 merupakan suatu *Interface* antara terminal data dan peralatan komunikasi data yang menggunakan data biner serial sebagai data yang ditransmisikan. RS 232 mengirim dan menerima data dengan kecepatan di atas 120kbps, bahkan kecepatannya mampu mencapai 250 kbps. Salah satu konektor yang biasa digunakan adalah konektor DB 9 seperti ditunjukkan gambar 2.2.



Gambar 2.14
Konektor DB 9^[18]

Pada dasarnya semua pin-pin memiliki fungsi yang berbeda, tetapi tidak semua digunakan. Fungsi-fungsi pin konektor DB 9 tertera pada tabel 2.1. berikut.

Tabel 2.5
Konfigurasi Pin DB 9⁽¹⁴⁾

DB-9 Pin	Signal	Function
2	RD	Received data
3	TD	Transmitted data
8	CTS	Clear to send
7	RTS	Request to send
6	DSR	Data set ready
5	SG	Signal ground
4	DTR	Data terminal ready
1	DCD	Data carrier detect

Sinyal yang dikeluarkan oleh komputer serial secara DCE maupun DTE adalah:

- *RTS (Request To Send)*

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

- *CTS (Clear To Send)*

Sinyal yang dikirimkan modem untuk memberitahu kepada PC bahwa telah dapat dilakukan pengiriman data.

- *DTR (Data Terminal Ready)*

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

- *DSR (Data Set Ready)*

Sinyal yang digunakan modem sebagai tanggapan dari sinyal DTR, dengan sinyal ini PC akan tahu modem dalam kondisi ON.

- RI (*Ring Indication*)

Sinyal indikator yang digunakan untuk memberitahu pada PC bahwa ada sinyal dering dari line telepon.

- TxD (*Transmitted Data*)

Sinyal yang digunakan modem untuk memberitahu bahwa data sudah dikirim.

- RxD (*Receive Data*)

Sinyal yang digunakan modem untuk memberitahu bahwa data sudah diterima.

b). Karakteristik Sinyal Listrik *Interface* RS 232

CCITT telah merekomendasikan karakteristik sinyal listrik pada rangkaian *Interface* V-24. Karakteristik sinyal listrik yang dimaksudkan adalah batas-batas tegangan yang digunakan. Dalam rekomendasi tersebut dibedakan antara sinyal data dan sinyal kontrol, sinyal data sebagai logika '0' dan '1', sedangkan sinyal kontrol dinyatakan dengan ON dan OFF. Sinyal-sinyal tersebut mempunyai batas-batas (daerah) tegangan yang ditunjukkan pada tabel 2.6.

Tabel 2.6
Tabel Karakteristik Sinyal Listrik Protokol RS 232^[6]

Sinyal	Nilai	Dacrah Tegangan
Sinyal data	1	$-3V \geq V_p \geq -15V$
	0	$+3V \leq V_p \leq +15V$
Sinyal kontrol	ON	$+3V \leq V_p \leq +15V$
	OFF	$-3V \geq V_p \geq -15V$

Dalam standar RS232, tegangan antara -3 sampai +15 Volt pada input *Line Receiver* dianggap sebagai *level* tegangan ‘0’, dan tegangan antara -3 sampai -15 Volt dianggap sebagai *level* tegangan ‘1’.

Untuk mengurangi kemungkinan terjadinya gangguan ‘*cross talk*’ antara kabel saluran sinyal RS 232, kecuraman perubahan tegangan sinyal diatasi tidak boleh lebih dari 30 Volt/mikro-detik, makin besar kecuraman sinyal, makin besar pula kemungkinan terjadi ‘*cross talk*’. Di samping itu ditentukan pula kecepatan transmisi data scri tidak boleh lebih besar dari 20 KiloBit/Detik. Impedansi saluran dibatasi antara 3 Kilo-Ohm sampai 7 Kilo-Ohm, dalam standar RS232 yang pertama ditentukan pula panjang kabel tidak boleh lebih dari 15 Meter (50 feet), tapi ketentuan ini sudah di-revisi pada standar RS232 versi ‘D’. Dalam ketentuan baru tidak lagi ditentukan panjang kabel maksimum, tapi ditentukan nilai kapasitan dari kabel tidak boleh lebih besar dari 2500 pF, sehingga dengan menggunakan kabel kualitas baik bisa dicapai jarak yang lebih dari 50 *feet*.

e). Protokol Komunikasi pada RS 232

Beberapa protokol pada *Interface* RS 232 adalah:

- *Start Bits*
Merupakan sebuah bit dengan logic '0', bit ini yang menandakan bahwa akan ada karakter atau data yang mengikutinya. Bit ini langsung diberikan oleh sinyal *device* tanpa harus mengeset terlebih dahulu.
 - *Data Bits*
Merupakan bit yang mewakili dari karakter yang diikutinya, data bit ini dapat diset sepanjang antara 5 sampai 8 bit.
 - *Parity Bits*
Merupakan bit yang digunakan sebagai *error checking* pada *receiver*. *Parity bit* akan menghitung jumlah data yang berlogika '1' pada data bit. Perhitungan jumlah data bit tersebut tergantung dari jenis *parity* yang diset. Untuk *parity 'even'*, jumlah data bit yang berlogika '1' ditambah dengan *parity bit* akan menghasilkan jumlah yang ganjil. Sedangkan untuk *parity 'mark'*, merupakan *parity bit* yang selalu berlogika '1' begitu pula pada space, *parity bit* selalu berlogika '0' dan *parity 'none'* merupakan *parity bit* yang diabaikan.
 - *Stop Bits*
Merupakan bit yang menandakan akhir dari suatu paket data (biasanya 1 *byte* data). Seperti pada data bit, bit ini langsung diberikan dari serial *device*. *Stop bit* ini dapat diset menjadi satu bit, satu setengah dan dua bit.
 - *Boud Rate*
Baud Rate digunakan untuk menunjukkan kecepatan dari transmisi (bits per second).
-

2.7. Pemrograman Visual Basic 6.0

Visual Basic 6.0 merupakan bahasa pemrograman yang cukup populer dan mudah untuk dipelajari. pengguna dapat membuat program dengan aplikasi GUI (*Graphical User Interface*) atau program yang memungkinkan pemakai komputer berkomunikasi dengan komputer tersebut dengan menggunakan modus grafik atau gambar.

Visual Basic 6.0 berawal dari bahasa pemrograman BASIC (*Beginners All-purpose Symbolic Instruction Code*). Karena bahasa BASIC cukup mudah dipelajari dan populer maka hampir setiap programmer menguasai bahasa ini.

Tahun 1980-an sistem operasi DOS cukup populer di kalangan pemakai PC karena di dalamnya disertakan bahasa BASIC yang dikenal dengan QBASIC (QuickBasic). Sistem tersebut sekarang sudah jarang digunakan. Di era Windows, Microsoft menciptakan Visual Basic yang terus mengalami penyempurnaan hingga Visual Basic 6.0 ini.

2.7.1. Memulai Visual Basic

Pada bagian ini akan dijelaskan bagaimana cara menjalankan Visual Basic 6.0 pada sistem operasi Windows. Cara pertama yang dapat dilakukan untuk memulai Microsoft Visual Basic 6.0 adalah:

1. Klik tombol Start pada Taskbar, kemudian pilih Program dari tampilan menu utama
 2. Dari tampilan menu yang ada, pilih Visual Basic 6.0. Untuk memudahkan pemahaman. Perhatikan Gambar 1. yang menunjukkan
-

proses cara menjalankan Visual Basic 6.0 dengan menggunakan Windows.

2.7.2. Tampilan Awal Visual Basic

Secara otomatis, pada saat pertama kali menjalankan Visual Basic, akan tampil kotak dialog New Project. Pada kotak dialog tersebut terdapat tiga pilihan tabulasi dengan keterangan sebagai berikut:

Tabel 2.7
Pilihan Tabulasi

Tabulasi	Keterangan
New	Pilihan ini digunakan untuk membuat project baru dengan berbagai macam pilihan.
Existing	Pilihan ini digunakan untuk membuka project yang pernah dibuat sebelumnya dengan menentukan folder sekaligus nama file.
Recent	Pilihan ini digunakan untuk membuka project yang telah dibuat dan terakhir kali dibuka.

2.7.3. Komponen Visual Basic

1. Title Bar

Title bar merupakan batang judul dari program Visual Basic 6.0 yang terletak pada bagian paling atas dari jendela program yang berfungsi untuk menampilkan judul atau nama jendela. Selain itu title bar juga berfungsi untuk:

- Memindah posisi jendela dengan menggunakan proses drag and drop pada posisi title bar tersebut.
- Mengatur ukuran jendela dari ukuran Maximize ke ukuran Restore ataupun sebaliknya dengan melakukan klik ganda pada posisi title bar tersebut.

2. Control Menu

Control Menu merupakan sebuah elemen yang terletak pada bagian sudut kiri atas dari jendela Visual Basic. Dalam sistem operasi Windows, elemen ini tampil dalam bentuk ikon program. Ketika meng-klik Control Menu, akan tampil daftar menu perintah yang digunakan untuk mengubah ukuran jendela, memindah letak jendela dan juga dapat untuk keluar dari program Microsoft Visual Basic 6.0. Untuk lebih jelasnya, dapat dilihat fungsi dari masing-masing perintah pada tabel di bawah ini:

Tabel 2.8.
Fungsi Perintah/Tombol

Tombol	Fungsi
Restore	Mengubah ukuran jendela Visual Basic ke ukuran relatif
Minimize	Mengubah ukuran jendela Visual Basic ke ukuran minimal.
Maximize	Mengubah ukuran jendela Visual Basic ke ukuran maksimal.
Move	Memindah posisi jendela Visual Basic.
Size	Mengatur ukuran jendela Visual Basic.
Close	Keluar dari program Visual Basic.

3. Menu Bar

Menu bar merupakan batang menu yang terletak di bawah title bar yang berfungsi untuk menampilkan pilihan menu atau perintah untuk mengoperasikan program Visual Basic. Saat pertama kali jendela program Visual Basic terbuka, ada tiga belas menu utama, yaitu: File, Edit, View, Project, Format, Debug, Run, Query, Diagram, Tools, Add-Ins, Windows dan Help.

Tampilan pilihan menu dalam program Visual Basic memiliki beberapa variasi yang masing-masing mempunyai pengertian yang berbeda dengan

penjelasan sebagai berikut:

- a. Menu dengan tanda segitiga yang terletak pada bagian kanan, di mana apabila menu tersebut dipilih maka akan menampilkan submenu pilihan berikutnya.
- b. Menu yang secara langsung akan menjalankan suatu perintah yang dipilih.
- c. Menu dengan tanda titik tiga pada bagian kanan, apabila dipilih maka akan menampilkan kotak dialog perintah.
- d. Menu yang tampil redup yang mengindikasikan bahwa menu tersebut tidak dapat untuk dipilih.
- e. Menu dengan gambar ikon di sebelah kirinya yang menandakan bahwa selain dapat dengan menggunakan menu tersebut untuk menjalankan suatu perintah, Anda dapat menekan tombol yang ada di toolbar.
- f. Menu dengan Shortcut key di sebelah kanannya yang menandakan bahwa selain dengan menggunakan menu tersebut untuk menjalankan suatu perintah, Anda dapat menekan tombol kombinasi pada keyboard. Misalnya kombinasi antara tombol Shift + F7 untuk mewakili perintah menu View - Object.

4. Toolbar

Toolbar merupakan sebuah batang yang berisi kumpulan tombol yang terletak di bagian bawah menu bar yang dapat digunakan untuk menjalankan suatu perintah. Pada kondisi default program Visual Basic hanya menampilkan toolbar Standard.

Untuk lebih jelasnya tentang fungsi masing-masing tombol pada toolbar standard, perhatikan penjelasan berikut:

Tabel 2.9.
Fungsi tombol pada toolbar Standard

Nama Tombol	Fungsi
Add Project	Menambahkan project baru,Dengan pilihan: • Standard EXE • ActiveX DLL • ActiveX EXE • ActiveX Control
Add Form	Menambahkan item, dengan pilthan: • Form • MDI Form • Module • Class Module • User Control • Property Page • Add File • User Document
Menu Editor	Menampilkan kotak dialog Menu Editor.
Open Project	Membuka project yang sudah pernah dibuat ebelumnya.
Save Project Group	Menyimpan project.
Cut	Memotong kontrol yang ada di jendela form atau teks yang ada di jendela code.
Copy	Menyalin kontrol yang ada di jendela form atau teks yang ada di Jendela code.
Paste	Menempelkan kontrol atau teks yang sudah dipotong dengan perintah Cut atau disalin dengan perintah Copy.
Find	Mencari teks pada jendela code.
Undo	Membatalkan suatu perintah an dialankan sebelumn a..
Redo	Mengulangi suatu perintah yang pernah dibatalkan
Start	Menjalankan program.
Break	Menghentikan program yang sedang jalan untuk Sementara.
End	Menghentikan program yang sedang dijalankan.
Project Explorer	Menampilkan jendela ProjectExplorer.
Properties Window	Menampilkan jendela Properties.
	Menampilkan jendela Form
Window	Layout.
Object	Menampilkan jendela Object
Browser	Browser.
Toolbox	Menampilkan jendela Toolbox.

5. Toolbox

Toolbox merupakan kotak perangkat yang berisi kumpulan tombol objek atau kontrol untuk mengatur desain dari aplikasi yang akan dibuat. Pada kondisi

default, toolbox menampilkan tabulasi General dengan 21 tombol kontrol yang dapat ditampilkan dengan menggunakan prosedur:

- 1. Klik tombol Toolbox di bagian Toolbar Standard.
- 2. Pilih perintah View - Toolbox.

Untuk penjelasan tentang fungsi masing-masing kontrol, perhatikan tabel berikut:

Tabel 2.10.
Fungsi masing-masing Kontrol pada Toolbox

Nama Pengontrol	Fungsi
Pointer	Memilih, mengatur ukuran danmemindah posisi kontrol yang terpasang pada bagian form.
PictureBox	Menampilkan file gambar.
Label	Menambahkan label atau teks tambahan.
TextBox	Menambahkan kotak teks.
Frame	Menambahkan kontrol yang dapat diisi dengan kontrol Option Button atau CheckBox.
Command Button	Menambahkan kontrol tombol perintah.
CheckBox	Menambahkan kontrol kotak periksa.
OptionButton	Menambahkan kontrol tombol pilihan.
ComboBox	Menambahkan kontrol kotak combo yang merupakan kontrol gabungan antara TextBox dan ListBox.
ListBox	Menambahkan kontrol daftar pilihan.
HscrollBar	Menambahkan kontrol batang penggulung horisontal.
VscrollBar	Menambahkan kontrol batang penggulung vertikal.
Timer	Menambahkan kontrol sebagai kontrol pencacah waktu.
DriveListBox	Menambahkan kontrol daftar disk drive pada kompuer.
DirListBox	Menambahkan kontrol daftar direktori pada drive aktif.
FileListBox	Menambahkan kontrol daftar file pada direktori aktif.
Shape	Menambahkan kontrol gambar berupa lingkaran, oval, persegi panjang, bujur sangkar dan lain-lain.
Line	Menambahkan kontrol gambar garis lurus.
Image	Menambahkan file gambar dengan pilihan properti yang lebih sedikit dibandingkan kontrol pictureBox.
Data	Menambahkan kontrol yang berupa database
OLE	Menambahkan kontrol yang berhubungan dengan proses relasi antar program aplikasi.

6. Project

Project merupakan suatu kumpulan module atau merupakan program aplikasi itu sendiri. Dalant Visual Basic, file project disimpan dengan nama file berakhiran.VBP, di mana file ini berfungsi untuk menyimpan seluruh komponen program.

Apabila kita membuat suatu program aplikasi baru, maka secara otomatis project tersebut akan diisi dengan objek Form 1. Dalam jendela Project Explorer ditampilkan suatu struktur hirarki dari sebuah project itu sendiri yang berisi semua item yang terkandung di dalamnya. Dengan Project Explorer dapat dipilih objek yang akan dibuat dengan mudah. Untuk Menampilkan jendela Project Explorer, Gunakan Prosedur berikut:

- 1. Klik tombol Project Explorer pada bagian toolbar Standard.
- 2. Pilih perintah View-.Project Explorer.
- 3. Shorcut key Ctrl+R.

Selain menampilkan nama project dan form, pada jendela Project Explorer terdapat tiga tombol dengan penjelasan sebagai berikut:

Tabel 2.11
Tombol pada Project Explorer

Nama Tombol	Fungsi
View Code	Menampilkan jendela Code yang digunakan untuk menulis kode program yang terhubung dengan objek yang terpilih pada jendela form.
View Object	Menampilkan jendela Object untuk item yang terpilih pada form aktif.
Toggle Folders	Menampilkan atau menyembu nyikan folder yang menam pung nama form dari suatu project.

7. Properties Window

Properties Window merupakan sebuah jendela digunakan untuk menampung nama properti dari yang terpilih. Pengaturan properti pada program Basic merupakan hal yang sangat penting membedakan objek yang satu dengan yang lainnya.

Pada jendela properti ditampilkan jenis dan nama objek yang bisa dipilihurut berdasarkan abjad pada tab Alphabetic atau berdasarkan kategori pada tab Categorized. Untuk menampilkan jendela Properties, dapat menggunakan prosedur berikut:

1. Klik tombol Properties Window pada toolbar Standard.
2. Pilih perintah View - Properties Window.
3. Shortcut key F4.

8. Form Layout Window

Form Layout Window merupakan sebuah jendela yang digunakan untuk mengatur posisi dari form pada form saat program dijalankan. Pada saat mengarahkan pointer mouse ke bagian form, maka pinter mouse akan berubah menjadi anak panah empat arah (pointer pengatur posisi). Untuk memindah posisi form pada layar monitor dapat dilakukan dengan proses drag and drop.

Untuk menampilkan jendela Form Layout, dapat menggunakan prosedur berikut:

1. Klik tombol Form Layout Window pada toolbar Standard.
 2. Pilih perintah View - Form Layout Window.
-

9. Immediate Window

Immediate Window merupakan sebuah jendela yang digunakan untuk mencoba beberapa perintah dengan mengetikkan baris program dan dapat secara langsung melihat hasilnya. Hal tersebut biasa dilakukan dan sangat membantu proses pengujian suatu perintah sebelum dipasang di dalam program.

Untuk menampilkan jendela Immediate, menggunakan prosedur berikut:

1. Pilih perintah View - Immediate Window.
2. Shortcut key Ctrl+G.
3. Klik tombol Immediate Window pada toolbar Debug.

10. Form Window

Form Window merupakan jendela desain dari sebuah program aplikasi. Kita dapat mendesain sebuah program aplikasi dengan menempatkan kontrol-kontrol yang ada di bagian toolbox pada area form.

Pada jendela form juga terdapat beberapa elemen yang dapat digunakan untuk mengatur tampilan.

11. Code Window

Code Window merupakan sebuah jendela yang digunakan untuk menuliskan kode program dari kontrol yang dipasang pada jendela form dengan cara memilih terlebih dahulu kontrol tersebut pada kotak objek.

12. Event

Event merupakan suatu kejadian yang akan diterima oleh suatu objek. Event yang diterima oleh objek berfungsi untuk menjalankan kode program yang ada didalam objek tersebut.

13. Method

Method adalah suatu kumpulan perintah yang memiliki kegunaan yang hampir sama dengan suatu fungsi atau prosedur, tetapi perintah-perintah tersebut sudah disesuaikan dalam suatu objek.

Suatu method dapat dipanggil dengan cara menyebutkan nama objek dan diikuti dengan tanda titik dan nama metodenya. Method umumnya digunakan untuk menjalankan perintah khusus pada suatu objek tertentu.

14. Module

Module hampir sama fungsinya dengan form, tetapi module tidak berisi objek dan bentuk standar. Dan module berisi kode program atau prosedur yang dapat digunakan oleh program aplikasi.

2.7.4. Mengolah Toolbox

Pada kondisi default, pada bagian toolbox terdapat satu tabulasi yaitu tabulasi General. Saat menambahkan item kontrol baru secara otomatis akan ditampilkan pada tab yang aktif (General). Kita dapat mengelompokkan item-item yang akan ditambahkan ke bagian toolbox dengan menggunakan tabulasi buatan sendiri. Diharapkan semua item yang ditambahkan ke toolbox dapat terorganisasi sesuai dengan aplikasi yang dibuat.

1. Menambah Tabulasi

Untuk menambahkan tabulasi ke bagian toolbox, dapat menggunakan prosedur berikut:

1. Klik kanan pada bagian toolbox.
2. Pilih perintah Add Tab sehingga akan tampil sebuah kotak dialog.
3. Ketikkan nama tabulasi pada kotak New tab Name dan klik tombol OK.

2. Menghapus Tabulasi

Untuk menghapus tabulasi dari bagian toolbox. Dapat menggunakan prosedur berikut:

1. Klik kanan bagian tabulasi yang ingin dihapus.
2. Pilih perintah Delete Tab.

3. Mengganti Nama Tabulasi

Untuk mengganti nama tabulasi, dapat menggunakan prosedur berikut:

1. Klik kanan pada bagian tabulasi yang ingin diganti namanya.
2. Pilih perintah Rename Tab.
3. Ketikkan nama tabulasi yang baru dan klik tombol OK.

4. Memindah Posisi Tabulasi

Untuk memindah posisi tabulasi, dapat menggunakan prosedur berikut:

1. Klik kanan pada bagian tabulasi yang ingin Anda pindah posisinya.
 2. Pilih perintah Move Up untuk memindah posisi ke atas atau Move Down untuk memindali posisi ke bawah.
-

5. Menambahkan Item Kontrol

Untuk menambahkan item kontrol, dapat menggunakan prosedur berikut:

1. Aktifkan tabulasi untuk menambahkan item pada bagian kontrol tersebut dengan cara klik.
2. Berikan salah satu perintah berikut:
 - a. Pilih perintah Project .Components.
 - b. Klik kanan pada bagian tabulasi toolbox dan pilih perintah Components.
 - c. Shortcut key Ctrl + y
3. Pada tabulasi Controls, aktifkan kotak periksa dari beberapa pilihan kontrol item yang ingin ditambahkan dan klik tombol Close.

6. Menghapus Item Kontrol

Untuk menghapus item kontrol, dapat menggunakan prosedur berikut:

1. Berikan salah satu perintah berikut:
 - a. Pilih perintah Project - Components.
 - b. Klik kanan pada bagian tabulasi toolbox dan pilih perintah Components.
 - c. Shorcut Key Ctrl + T.
2. Pada tabulasi controls, matikan kotak periksa dari beberapa pilihan kontrol item yang akan dihilangkan dari toolbox dan klik tombol close

2.8. *Hand Phone* (HP)

HP merupakan jenis telepon bergerak yang menggunakan teknologi sel sebagai akses komunikasinya, sehingga lebih efektif dan efisien. Jenis HP yang

digunakan dalam Tugas Akhir adalah Siemens C35 dan serinya. Selain mengirim data suara, HP juga dapat mengirim data yang berupa karakter-karakter yang disebut SMS (*Short Service Message*). HP tidak hanya dapat berkomunikasi antar HP, tetapi juga dapat berkomunikasi dengan PC (*Personal Computer*). Untuk dapat berkomunikasi dengan PC, HP dilengkapi saluran khusus yang dinamakan saluran FBUS.

2.9. Protokol Komunikasi Data Serial Pada Siemens.

2.9.1. Karakteristik Level Sinyal Pada FBUS

Level tegangan perlu diperhatikan karena terdapat perbedaan *level* tegangan antara protokol FBUS dengan protokol RS232. *Level* tegangan protokol FBUS ini berbeda dengan *level* tegangan TTL maupun UART. *Level* sinyal FBUS adalah 0 volt untuk logika '0' dan 2.7 volt untuk logika '1'.

Oleh karena itu, diperlukan suatu konverter *level* tegangan dari RS232 ke protokol FBUS. Hal ini dilakukan oleh suatu *Interface* yang diletakkan antara komputer dan ponsel, pada *Interface* tersebut terdapat IC MAX232 yang mampu merubah *level* tegangan RS232 (pada serial PC) ke *level* tegangan TTL ini sudah ada pada kabel data pada kabel FBUS yang ada didalamnya.

2.9.2. Protokol FBUS

Saluran komunikasi FBUS dipakai sebagai saluran komunikasi data seri *asinkron* dengan format data model UART, yang dipakai dalam serial *port* PC. FBUS pada Siemens memakai 2 jalur komunikasi yaitu Rx dan Tx, yang dipakai

untuk transmisi data secara *full duplex* (komunikasi dua arah secara bersamaan). Kecepatan transfer data lewat FBUS dapat mencapai 115.200 *baud*. FBUS digunakan untuk pertukaran data yang mencakup pertukaran data phone book, logo, ring tone.

2.9.3. Format Data SMS Pada Siemens

Data yang mengalir ke atau dari SMS-Centre harus berbentuk PDU (*Protokol Data Unit*). PDU berisi bilangan-bilangan heksadesimal yang mencerminkan bahasa I/O. PDU terdiri atas beberapa Header. Header untuk kirim SMS ke SMS-Centre berbeda dengan SMS yang diterima dari SMS-Centre.

2.9.4. PDU Untuk kirim SMS ke SMS-Centre

No. SMS-C	Tipe SMS	No. Referensi	No. Penerima	Bentuk SMS	Skema Encoding	Waktu Validitas	Isi SMS
-----------	----------	---------------	--------------	------------	----------------	-----------------	---------

Gambar 2.15 Format data PDU untuk kirim SMS^[13]

Format PDU untuk mengirim SMS terdiri atas delapan Header, yaitu sebagai berikut :

1. Nomor SMS-Centre

Header pertama ini terbagi atas tiga sub-header, yaitu :

- a. Jumlah pasangan Heksadesimal SMS-Centre dalam bilangan heksa.
- b. National/International Code.
 - Untuk *National*, kode subhcader-nya yaitu 81.
 - Untuk *International*, kode subheader-nya yaitu 91.

- c. Nomor SMS-Centrenya sendiri, dalam pasangan heksa dibalik-balik. Jika tertinggal 1 angka heksa yang tidak memiliki pasangan, angka tersebut akan dipasangkan dengan huruf F di depannya. Contoh : untuk nomor SMS-Centre Telkomsel dapat ditulis dengan dua cara yaitu :

- Cara 1, 081100000 diubah menjadi :

- 1). 06 → 6 pasang
2). 81 → 1 pasang
3). 80-11-00-00-F0 → 5 pasang

Total ada 6 pasang, digabung menjadi 068180110000F0.

- Cara 2, 62811000000 diubah menjadi :

- 1). 07 → 7 pasang
2). 91 → 1 pasang
3). 26-18-01-00-00-F0 → 6 pasang

Total ada 6 pasang, digabung menjadi : 07912618010000F0.

Berikut ini nomor beberapa SMS-Centre Operator selular di Indonesia

Tabel 2.12

Nomor SMS-Centre Operasi Selular di Indonesia^[13]

No.	Operator Selular	SMS-Centre No	Kode PDU
1	Telkomsel	081100000	068180110000F0
2	Satelindo	0816125	0581806121F5
3	Excelcom	0818445009	06818081440590
4	Indosat-M3	0855000000	06818055000000

No.	Operator Selular	SMS-Centre No	Kode PDU
1	Telkomsel	62811000000	07912618010000F0
2	Satelindo	62816125	059126181652
3	Excelcom	62818445009	07912618485400F9
4	Indosat-M3	62855000000	07912658050000F0

2. Tipe SMS

Untuk tipe SMS kirim=1, jadi bilangan heksanya adalah 01.

3. Nomor Referensi

Nomor Referensi ini dibiarkan dulu 0, jadi bilang heksanya adalah 00. Nomor referensi ini akan diberikan secara otomatis oleh ponsel tersebut.

4. Nomor ponsel penerima

Header ini juga terbagi menjadi tiga bagian, yaitu :

- Jumlah bilangan decimal nomor ponsel yang dituju dalam bilang heksa.
- National/International Code.
 - Untuk *National*, kode subheader-nya yaitu 81
 - Untuk *International*, kode subheader-nya yaitu 91
- Nomor ponsel yang dituju, dalam pasangan heksa dibalik-balik. Jika tertinggal satu angka heksa atau tidak memiliki pasangan, maka angka tersebut dipasangkan dengan huruf F didepannya. Contoh untuk nomor ponsel yang dituju **6281334465659** dapat ditulis dengan dua cara, yaitu :
 - Cara 1, 081334465659 diubah menjadi :

1). 0C $\longrightarrow$ 12 angka

2). 81 sebagai kode *National*

3). 80-31-43-64-65-95

Sehingga digabung menjadi 0C81803143646595

- Cara 2, 6281334465659 diubah menjadi :

1). 0D $\longrightarrow$ 13 angka

2). 91 sebagai kode *International*

3). 26-18-33-44-56-56-F9

Sehingga digabung menjadi 0D91261833445656F9

5. Bentuk SMS, antara lain :

0 $\rightarrow$ 00 $\rightarrow$ dikirim sebagai SMS

1 $\rightarrow$ 01 $\rightarrow$ dikirim sebagai telex

2 $\rightarrow$ 02 $\rightarrow$ dikirim sebagai fax

Dalam hal ini, untuk mengirim dalam bentuk SMS tentu saja harus memakai kode heksa 00

6. Skema Encoding Data I/O

Ada dua skema, yaitu :

- Skema 7 bit, ditandai dengan angka 0 $\rightarrow$ 00
- Skema 8 bit, ditandai dengan angka lebih besar dari 0 kemudian diubah ke heksa

Kebanyakan Ponsel/SMS Gateway yang ada di pasaran sekarang menggunakan skema 7 bit sehingga kita menggunakan kode 00

7. Jangka Waktu Validitas

Jika jangka waktu validitas diabaikan (kode 00), itu berarti tidak membatasi waktu berlakunya SMS. Sedangkan jika diisi dengan suatu bilangan integer yang kemudian diubah ke pasangan heksa tertentu, bilangan yang diberikan tersebut akan mewakili jumlah waktu validitas SMS tersebut. Rumus untuk menghitung jangka waktu validitas SMS adalah sebagai berikut :

Tabel 2.13
Jangka Waktu Validitas SMS^[13]

Integer (INT)	Jangka Waktu Validitas SMS
0-144	$(INT+1) \times 5$ menit (berarti : 5 menit s.d 12 jam)
144-167	12 jam + $[(INT-143) \times 30 \text{ menit}]$
168-196	$(INT-166) \times 1$ hari
197-255	$(INT-192) \times 1$ minggu

8. Isi SMS

Header ini terdiri atas dua subheader, yaitu :

- Panjang isi (jumlah huruf dari isi)

Misalnya : Untuk kata “hello”, ada 5 huruf maka penulisannya 05.

- Isi berupa pasangan bilangan heksa

Untuk ponsel/SMS Gateway berskema encoding 7 bit, jika mengetik suatu huruf dari keypadnya, berarti telah membuat 7 I/O berurutan.

Ada dua langkah yang harus dilakukan untuk mengkonversikan isi SMS, yaitu:

Langkah pertama : mengubah menjadi kode 7 bit

Langkah kedua : mengubah kode 7 bit menjadi 8 bit yang diwakili oleh pasangan heksa

Contoh untuk kata "hello"

o Langkah pertama :

Bit	7	1
h	110	1000
e	110	101
l	110	1100
l	110	1100
o	110	1111

o Langkah kedua :

h	<u>1</u>	^E 110	⁸ 1000
e	<u>00</u>	³ 11	² 0010 1
l	<u>100</u>	⁹ 1	^B 1011 00
l	<u>1111</u>	^F 1101	^D 100
o	<u>0000</u>	⁰ 0	⁶ 110 1111

Oleh karena total 7 bit x 5 huruf = 35 bit, sedangkan yang kita perlukan adalah 8 bit x 5 huruf = 40 bit, maka diperlukan 5 bit dummy (sisanya) yang diisi dengan bilangan 0. Setiap 8 bit mewakili suatu pasangan heksa. Tiap 4 bit mewakili suatu angka heksa. Dengan demikian kata “hello” hasil konversinya menjadi E8329BFD06.

➤ **Penggabungan Kedelapan Header**

Setelah kita mempelajari masing-masing header maupun subheader untuk mengirim SMS, kita dapat menggabungkannya menjadi sebuah PDU yang lengkap.

Untuk mengirim kata “hello” ke ponsel nomor 6281334465659 lewat SMS-Centre Telkomsel, tanpa membatasi jangka waktu valid, maka PDU lengkapnya adalah **07912618010000F001000D91261833445656F9000005E8329BFD06**

2.9.5. PDU Untuk SMS Terima Dari SMS-Centre

No. SMS-C	Tipe SMS	No. Pengirim	Bentuk SMS	Skema Encoding	Tgl&Waktu SMS	Waktu Validitas	Isi SMS
-----------	----------	--------------	------------	----------------	---------------	-----------------	---------

Gambar 2.16 Format data PDU untuk Terima SMS^[13]

Kebanyakan format Header SMS terima telah dibahas pada Header SMS kirim, kecuali beberapa yang berbeda.

- 1. Nomor SMS-Centre
- 2. Tipe SMS untuk terima = 4 menjadi 04 atau 24
- 3. Nomor ponsel pengirim
- 4. Bentuk SMS

5. Skema Encoding
6. Tanggal dan waktu SMS di SMS-Centre
 Diawali oleh 12 bilangan heksa (6 pasang) yang berarti : yy/mm/dd
 Hh:mm:ss. Misalnya 603050512380 yang berarti : 06/03/05 15:32:08 atau
 05 maret 2006 15:32:08 WIB.
7. Batas Waktu Validitas, jika tidak tidak dibatasi dilambangkan dengan 00
8. Isi SMS

➤ **Membedah Kedelapan Header SMS Terima**

Setelah mengupas satu dcmi satu Header untuk SMS terima ini, maka untuk PDU sebagai berikut

**07912618010000F0,04,0D91261833445656F9,00,00,603050512380,00,05,E
 8329BFD06**

Dapat kita artikan sebagai berikut:

1. SMS tersebut dikirim lewat SMS-Centre : 62811000000
2. SMS tersebut merupakan SMS terima
3. SMS tersebut dikirim dari ponsel nomor 6281334465659
4. SMS tersebut diterima dalam bentuk SMS
5. SMS tersebut memiliki skema encoding 7 bit
6. SMS tersebut sampai di SMS-Centre pada tanggal 05-03-06, pukul
 15:32:08
7. SMS tersebut tidak memiliki batas waktu valid
8. SMS tersebut isinya adalah "hello"

BAB III

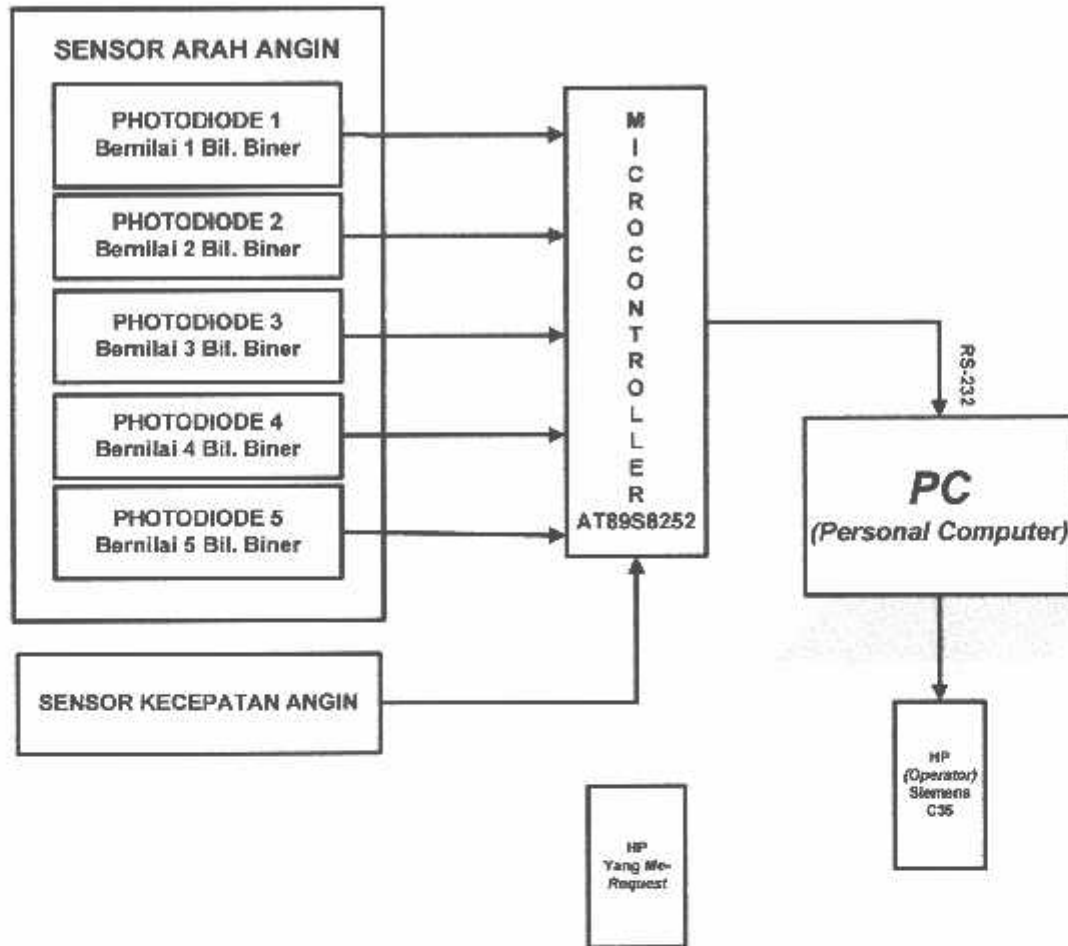
PERENCANAAN DAN PEMBUATAN ALAT

3.1. Gambaran Umum Sistem

Pada bab ini akan di bahas mengenai perencanaan dan pembuatan sistem untuk mendeteksi perubahan kecepatan dan arah angin yang terjadi setiap saat. Informasi yang didapat oleh sensor kemudian akan dikirimkan secara serial ke PC (Personal Computer). Data yang diterima oleh PC dari mikrokontroller AT89S8252 akan diproses menggunakan Visual Basic dan dikirimkan ke handphone lain melalui SMS

3.2. Perancangan Perangkat Keras

Pada bagian ini akan di jelaskan secara garis besar tentang Perencanaan dan Pembuatan sistem monitoring kecepatan dan arah angin dengan menggunakan optocoupler sebagai sensor kecepatan dan photodiode sebagai sensor arah angin yang akan ditampilkan di PC dan dapat diakses via sms.



Gambar 3.1

Block Diagram Sistem Monitoring Kecepatan dan Arah Angin

Alat ini dirancang untuk memberikan informasi kecepatan dan arah angin setiap detiknya. Secara Block diagram perencanaan perangkat keras alat ditunjukkan seperti diatas.

Prinsip Kerja dari block diagram diatas sebagai berikut:

- Sensor kecepatan angin yang dipakai disini adalah optocoupler yang berfungsi untuk menghasilkan pulsa dari piringan optocoupler yang berputar pada kincir angin.

- Dan Sensor arah angin berfungsi sebagai pendeteksi dari mana datangnya angin. Untuk mengetahui hal itu disini digunakan 5 photodiode. Masing-masing photodiode tersebut mewakili bilangan biner yang berbeda-beda pada piringan konsentris yang dipasang anatara photodiode dengan infra red. Bilangan-bilangan biner ini akan membentuk kode-kode dimana setiap kode-kode yang dihasilkan menunjukkan berapa besar sudut arah angin dengan acuan pertama kali adalah arah utara.

1. Photodiode 1

Kondisi logika dari output photodiode ini mempunyai nilai biner terendah ($2^0 = 1$ Decimal) dan terletak dalam cincin piringan konsentris terdalam.

2. Photodiode 2

Kondisi logika dari output photodiode ini mempunyai nilai biner terendah ($2^1 = 2$ Decimal) dan terletak dalam cincin piringan konsentris kedua dari dalam.

3. Photodiode 3

Kondisi logika dari output photodiode ini mempunyai nilai biner terendah ($2^2 = 4$ Decimal) dan terletak dalam cincin piringan konsentris ketiga dari dalam.

4. Photodiode 4

Kondisi logika dari output photodiode ini mempunyai nilai biner terendah ($2^3 = 8$ Decimal) dan terletak dalam cincin piringan konsentris keempat dari dalam.

5. Photodiode 5

Kondisi logika dari output photodiode ini mempunyai nilai biner terendah ($2^4 = 16$ Decimal) dan terletak dalam cincin piringan konsentris terluar.

- Mikrokontroler AT89S8252 berfungsi sebagai pengatur system secara keseluruhan. Dimana system minimum ini merupakan otak dari system yang akan mengendalikan kerja dari optocoupler, dan pengiriman data secara serial melalui RS-232. Mikrokontroler akan membaca pulsa sebagai logika high-low yang kemudian akan dicounter.
 - Sistem Komunikasi Serial RS-232, Digunakan sebagai antarmuka serial antara mikrokontroler dengan PC (*Personal Computer*) untuk melihat hasil data yang ditangkap oleh sensor.
 - PC (*Personal Computer*), berfungsi mengambil data kecepatan dan arah angin dari mikrokontroler untuk disimpan dan kemudian komputer menunggu *Re-quest*
 - Handphone digunakan sebagai sarana untuk mengirimkan informasi ke Handphone lain Via SMS.
-

3.2.1. Perancangan Rangkaian Sensor Arah Angin

3.2.1.1. Perencanaan Rangkaian Pemancar LED Infra Red

Led Infra Red dalam hal ini berfungsi sebagai sumber cahaya yang memancarkan sinyal cahaya. Rangkaian dari pemancar LED Infra Red seperti gambar 3.2 berikut :



Gambar 3.2
Rangkaian Pemancar LED Infra Red

Rangkaian Diode pemancar cahaya (LED Infra Red) Dengan $V_{cc} = 5$ Volt dan arus yang mengalir direncanakan sebesar 8 mA, maka dapat diketahui besar resistor R adalah :

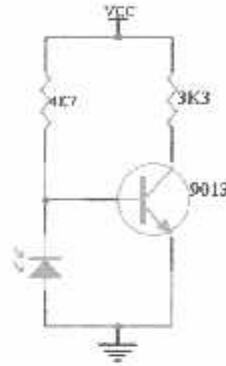
$$R = \frac{V_{CC} - V_{LED}}{I}$$

$$R = \frac{5 - 1,5}{8 \cdot 10^{-3}} = 437 \, \Omega$$

Maka nilai R yang digunakan sebesar 470 Ω sesuai dengan yang ada di pasaran.

3.2.1.2. Perencanaan Rangkaian Penerima Infra Red

Pada sensor arah angin digunakan Photodiode sebagai penerima sinyal cahaya yang dipancarkan LED Infra Red. Diantara LED Infra Red dan Photodiode dipasang piringan dimana piringan tersebut terdapat sisi gelap-terang.



Gambar 3.3
Rangkaian Penerima Infra Red

1. Kondisi saat Photodiode tidak mendapat cahaya/terhalang.

Pada saat seperti ini Photodiode mempunyai hambatan besar sekali sehingga tegangan dari Vcc dibias ke transistor. Arus yang melewati photodiode max 10mA, untuk rangkaian diberikan arus sebesar 0,5mA supaya tidak cepat rusak karena rangkaian ini aktif dalam waktu yang lama. Menentukan nilai Rc sebesar 3K3. Dengan V_{Photo} sebesar 2,3 V. Maka :

$$V_{CC} = V_R + V_{\text{Photo}}$$

$$V_R = V_{CC} - V_{\text{Photo}}$$

$$V_R = 5 - 2,3$$

$$V_R = 2,7 \text{ Volt}$$

$$V_R = I_{\text{Photo}} \cdot R_I$$

$$2,7 = 5 \cdot 10^{-4} \cdot R_1$$

$$R_1 = \frac{2,7}{5 \cdot 10^{-4}}$$

$$R_1 = 5K4$$

Karena dipasaran tidak ada nilai resistor sebesar 5K4 maka diganti dengan nilai terdekat yaitu 4K7.

Dengan tegangan sebesar 2,7 Volt ini, agar dapat mengkondisikan transistor C9013 berada dalam kondisi saturasi ($V_{ce} = 0$). Dengan transistor dalam keadaan saturasi maka tegangan Vcc akan mengalir ke ground, sehingga praktis dapat dikatakan tegangan yang mengalir ke mikrokontroller adalah 0V.

2. Kondisi saat Photodiode mendapat cahaya/tidak terhalang.

Pada saat kondisi ini Photodiode mempunyai hambatan kecil, sehingga tegangan tidak dibias, melainkan mengalir ke ground, akibatnya transistor berada dalam kondisi cut-off, sehingga tegangan Vcc transistor akan mengalir ke mikrokontroller.

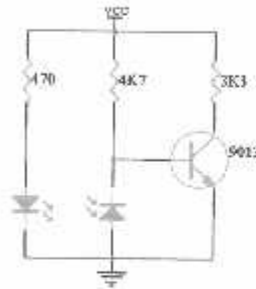
Jadi dapat disimpulkan bahwa Photodiode tidak mendapat cahaya maka logika ke mikrokontroller adalah 0 (low). Pada saat Photodiode mendapat cahaya maka logika ke mikrokontroller adalah 1 (high).

3.2.2. Perencanaan Rangkaian Sensor Kecepatan Angin

Rangkaian Sensor yang digunakan adalah Optocoupler yang berfungsi untuk mendeteksi jumlah pulsa yang masuk dari kincir angin yang dipasang pada poros kincir angin. Rangkaian Optocoupler ini terdiri dari diode pemancar

cahaya dan photodiode yang terpisah oleh celah, dan apabila diberi tegangan maka akan menghasilkan tegangan output berupa pulsa.

Photodiode pada optocoupler berfungsi untuk menerima sinar dari diode pemancar cahaya.



Gambar 3.4
Rangkaian Optocoupler

Led Infra Red dalam hal ini berfungsi sebagai sumber cahaya yang memancarkan sinyal cahaya. Rangkaian dari pemancar LED Infra Red seperti pada gambar 3.4

Rangkaian Diode pemancar cahaya (LED Infra Red) Dengan $V_{CC} = 5$ Volt dan arus yang mengalir direncanakan sebesar 8 mA, maka dapat diketahui besar resistor R adalah :

$$R = \frac{V_{CC} - V_{LED}}{I}$$

$$R = \frac{5 - 1,5}{8 \cdot 10^{-3}} = 437 \, \Omega$$

Maka nilai R yang digunakan sebesar 470 Ω sesuai dengan yang ada di pasaran.

1. Kondisi saat Photodiode tidak mendapat cahaya/terhalang.

Pada saat seperti ini Photodiode mempunyai hambatan besar sekali sehingga tegangan dari V_{cc} dibias ke transistor. Arus yang melewati photodiode max 10mA, untuk rangkaian diberikan arus sebesar 0,5mA supaya tidak cepat rusak karena rangkaian ini aktif dalam waktu yang lama. Menentukan nilai R_c sebesar 3K3. Dengan V_{Photo} sebesar 2,3 V. Maka :

$$V_{cc} = V_B + V_{Photo}$$

$$V_R = V_{CC} - V_{Photo}$$

$$V_R = 5 - 2,3$$

$$V_R = 2,7 \text{ Volt}$$

$$V_R = I_{Photo} \cdot R_1$$

$$2,7 = 5 \cdot 10^{-4} \cdot R_1$$

$$R_1 = \frac{2,7}{5 \cdot 10^{-4}}$$

$$R_1 = 5K4$$

Karena dipasaran tidak ada nilai resistor sebesar 5K4 maka diganti dengan nilai terdekat yaitu 4K7.

Dengan tegangan sebesar 2,7 Volt ini, agar dapat mengkondisikan transistor C9013 berada dalam kondisi saturasi ($V_{ce} = 0$). Dengan transistor dalam keadaan saturasi maka tegangan V_{cc} akan mengalir ke ground, sehingga praktis dapat dikatakan tegangan yang mengalir ke mikrokontroller adalah 0V.

2. Kondisi saat Photodiode mendapat cahaya/tidak terhalang.

Pada saat kondisi ini Photodiode mempunyai hambatan kecil, sehingga tegangan tidak dibias, melainkan mengalir ke ground, akibatnya transistor

berada dalam kondisi cut-off, sehingga tegangan V_{cc} transistor akan mengalir ke mikrokontroler.

Jadi dapat disimpulkan bahwa Photodiode tidak mendapat cahaya maka logika ke mikrokontroler adalah 0 (low). Pada saat Photodiode mendapat cahaya maka logika ke mikrokontroler adalah 1 (high).

3.2.3. Piringan Pada Sensor

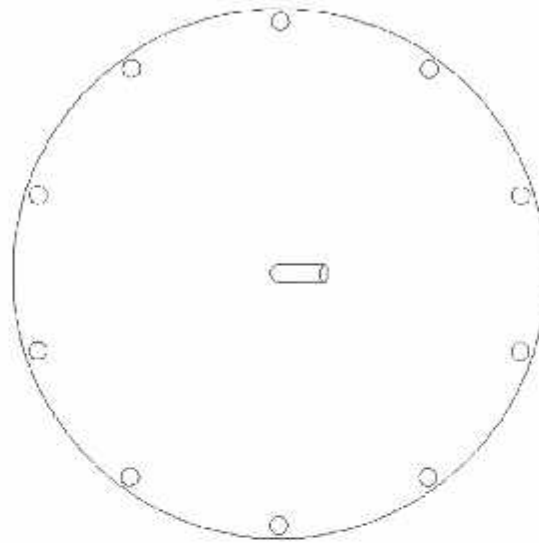
Piringan pada sensor kecepatan dan arah sensor angin sangat berbeda. Pada sensor arah angin, piringan dibuat dengan memperhitungkan selisih sudut yang sama antara lubang satu dengan yang lainnya pada piringannya. Untuk mendapatkan hasil yang Presisi, diperlukan ketelitian dalam pembuatan lubang pada piringan sensor arah angin.

3.2.3.1. Piringan Untuk Sensor Kecepatan Angin

Pada Optocoupler yang difungsikan sebagai pengukur kecepatan angin, dipasang piringan yang berbentuk silindris dengan jari-jari 5 cm dan diameter tiap lubangnya 3mm sebanyak 10 lubang yang akan menghasilkan 10 buah pulsa keluaran dalam satu putaran. Banyaknya putaran tergantung pada kencang lambatnya angin, semakin angin bertiup, maka semakin banyak putaran dan pulsa yang dihasilkan.

Pulsa yang dihasilkan oleh Optocoupler ini akan discounter oleh mikrokontroler. Dari hasil counter ini akan diketahui berapa banyak pulsa yang dihasilkan oleh piringan Optocoupler yang diasumsikan sebagai kecepatan angin.

Pada piringan Optocoupler, lubang satu dengan lainnya mempunyai selisih perbedaan sudut 36° . Misalkan dalam satu lubang pada piringan 1/10 putaran kecepatan dalam satuan Km dan pengukuran ini dilakukan per detik, maka apabila dalam satuan detik hanya dihasilkan 2 pulsa keluaran, maka kecepatan angin diasumsikan sebesar 0,2 m/det.



Gambar 3.5
Piringan Untuk Sensor Kecepatan Angin

Frekuensi dari pulsa yang dihasilkan oleh piringan Optocoupler dan dengan mengabaikan gaya gesek pada poros, maka :

$$f = \frac{n\omega}{60} \dots\dots\dots (3-1)$$

$$f = \frac{1}{T}$$

$$T = \frac{60}{n\omega} \dots\dots\dots (3-2)$$

$$K = 2 \cdot \pi \cdot r \dots\dots\dots (3-3)$$

Dimana :

f = Frekuensi (Hz)

n = Jumlah lubang pada piringan

ω = Kecepatan Sudut (rpm)

T = Periode (detik)

K = Keliling Lingkaran (m)

$\pi = 3,14$

r = Jari-jari Piringan (m)

Sedangkan untuk persamaan Kecepatan dinyatakan :

$$V = \frac{S}{t} \dots\dots\dots (3-4)$$

Karena jarak yang ditempuh berupa lingkaran, maka dalam persamaan diatas $S =$

K dan $t = T$. Akan didapat persamaan :

$$V = \frac{2.\pi.r}{T} \dots\dots\dots (3-5) [\quad]$$

Atau

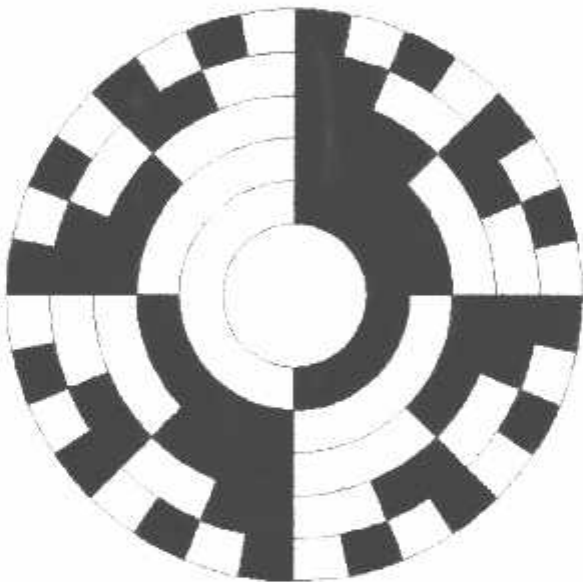
$$V = 2.\pi.f$$

3.2.3.2. Piringan Pada Sensor Arah Angin

Piringan yang dipasang pada sensor arah angina berbentuk silindris seperti pada sensor kecepatan angin, hanya bedanya piringan dibuat pola-pola pembentuk kode yang tersusun secara konsentris pada sisi-sisi piringan tersebut. Pola-pola tersebut terbentuk dari segment-segment yang bergantian antara gelap dan terang pada piringan. ¹¹¹

Piringan ini terdiri dari 5 cincin yang disusun secara dari pusat menuju kearah luar jumlah bagian gelap-terang pada cincin konsentris bertambah dalam cacahan biner, dari cincin yang paling dalam bernilai 16 cacahan biner sampai 1 cacahan biner pada cincin terluar. Maka masing-masing posisi sudut dari poros penggerak akan mempunyai kombinasi kode yang berbeda-beda.

Karena pada piringan ini dipasang 5 photodiode yang akan mendeteksi setiap bilangan biner yang ada pada piringan, maka $360^{\circ} : 2^5 = 11 \frac{1}{4}^{\circ}$. Ini berarti lubang satu dengan yang laiannya pada piringan selisih beda sudut sebesar $11 \frac{1}{4}^{\circ}$.



Gambar 3.6
Piringan Pada Sensor Arah Angin

Dari piringan diatas, maka akan didapatkan pengkodean arah mata angin seperti terlihat pada table 3.1 dabwah ini :

Tabel 3.1
Pengkodean Arah Angin Menggunakan Pencacah Bilangan Biner

DESIMAL	BILANGAN BINER	ARAH	SUDUT
0	0 0 0 0 0	UTARA	0 ^o
1	0 0 0 0 1		11,25 ^o
2	0 0 0 1 0		22,5 ^o
3	0 0 0 1 1	BARAT LAUT	33,75 ^o
4	0 0 1 0 0		45 ^o
5	0 0 1 0 1		56,25 ^o
6	0 0 1 1 0	BARAT	67,5 ^o
7	0 0 1 1 1		78,75 ^o
8	0 1 0 0 0		90 ^o
9	0 1 0 0 1		100,25 ^o
10	0 1 0 1 0		112,5 ^o
11	0 1 0 1 1	BARAT DAYA	123,75 ^o
12	0 1 1 0 0		135 ^o
13	0 1 1 0 1		146,25 ^o
14	0 1 1 1 0	SELATAN	157,5 ^o
15	0 1 1 1 1		168,75 ^o
16	1 0 0 0 0		180 ^o
17	1 0 0 0 1		191,25 ^o
18	1 0 0 1 0		205,5 ^o
19	1 0 0 1 1	TENGGAH	213,75 ^o
20	1 0 1 0 0		225 ^o
21	1 0 1 0 1		236,25 ^o
22	1 0 1 1 0	TIMUR	247,5 ^o
23	1 0 1 1 1		258,75 ^o
24	1 1 0 0 0		270 ^o

25	1 1 0 0 1		281,25 ⁰
26	1 1 0 1 0		292,5 ⁰
27	1 1 0 1 1	TIMUR LAUT	303,75 ⁰
28	1 1 1 0 0		315 ⁰
29	1 1 1 0 1		326,25 ⁰
30	1 1 1 1 0	UTARA	337,5 ⁰
31	1 1 1 1 1		348,75 ⁰

3.2.4. Sistem Mikrokontroller AT89S8252

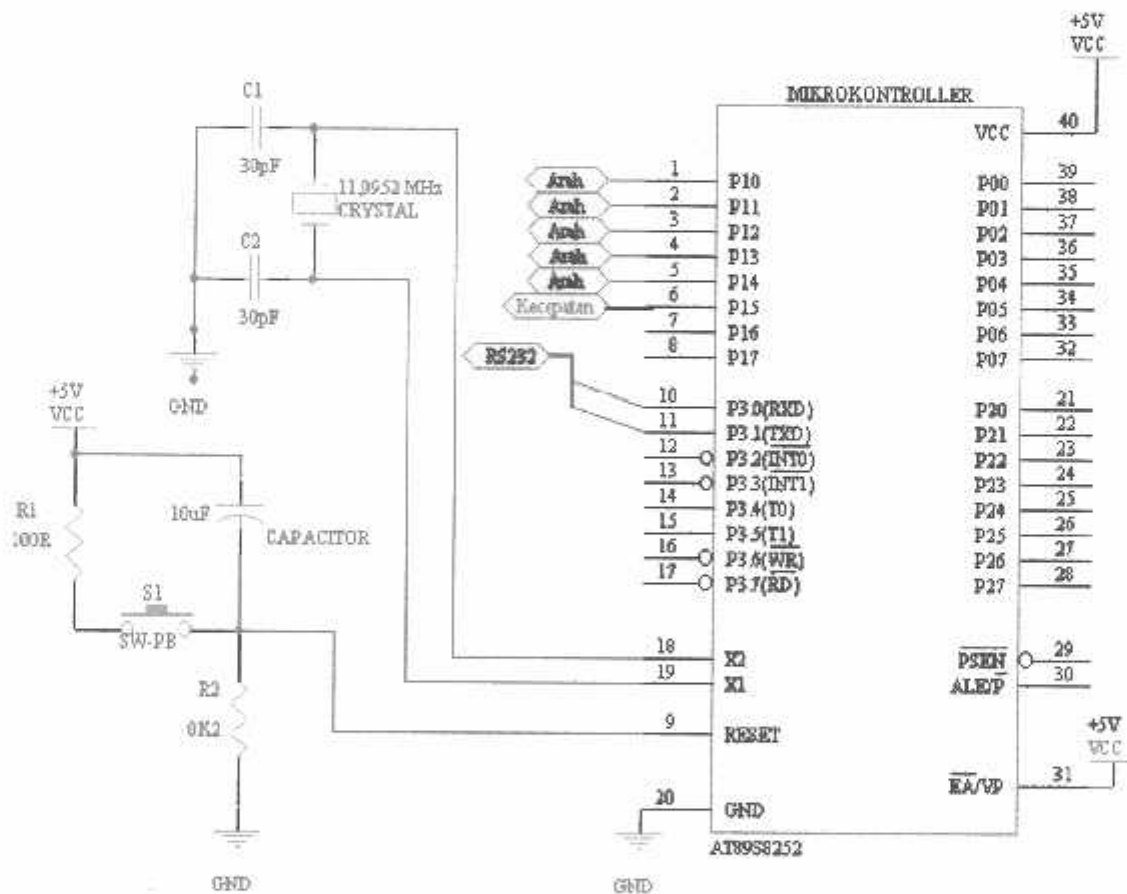
Mikrokontroller AT89S8252 digunakan untuk mendapatkan rangkaian yang praktis dan ringkas. Maka untuk mengolah data rangkaian pengukur kecepatan dan arah angin digunakan mikrokontroller AT89S8252 yang merupakan IC mikrokontroller dengan fasilitas EEPROM sebagai tempat penyimpanan program, sehingga sistem alat ukur menjadi lebih sederhana. Mikrokontroller AT89S8252 memiliki 4 port yang masing-masing port digunakan sebagai sebagai berikut:

- Port 1.5 digunakan untuk sensor kecepatan angin.
- Port 1.0, 1.1, 1.2, 1.3, 1.4 digunakan untuk input sensor arah angin.

Secara berurutan:

- P1.0 digunakan untuk photodiode terluar dengan nilai biner 2^0 .
- P1.1 digunakan untuk photodiode kedua dari yang terluar. nilai biner 2^1 .
- P1.2 digunakan untuk photodiode ketiga dari yang terluar, dengan nilai biner 2^2 .

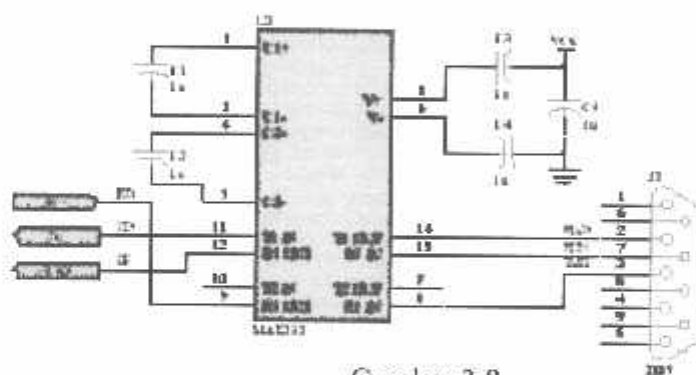
- P1.3 digunakan untuk photodiode kedua dari yang terdalam, dengan nilai biner 2^3 .
- P1.4 digunakan untuk photodiode terdalam dengan nilai biner 2^4 .
- Port 3.0 dan port 3.1 berfungsi sebagai Tx dan Rx untuk komunikasi serial RS-232 dengan komputer
- Kristal yang digunakan dalam mikrokontroller ini adalah 11,0952 MHz, hal ini berpengaruh pada kecepatan pengiriman sinyal data komunikasi serial.



Gambar 3.7
Minimum Sistem AT89S8252

3.2.5. Interface RS-232

Pada perencanaan hubungan antara MC dengan PC adalah dipergunakan komunikasi data secara serial yaitu port 1 atau yang sering dikenal dengan COM 1. Adapun kaki atau pin-pin yang dipakai adalah pada pin no. 2 yang berfungsi untuk sambungan *receive data*, pin no. 3 untuk sambungan *transmit data* yang berguna sebagai input data pada PC yang sebelumnya telah disesuaikan dulu level tegangannya dari RS-232 ke level tegangan TTL melalui sebuah IC MAX 232, dan pin no. 5 untuk *signal ground*. Dalam hal ini kecepatan transfer data per *bit* menggunakan 9600 bps. Rangkaian interface RS-232 diperlihatkan pada gambar 3.8.



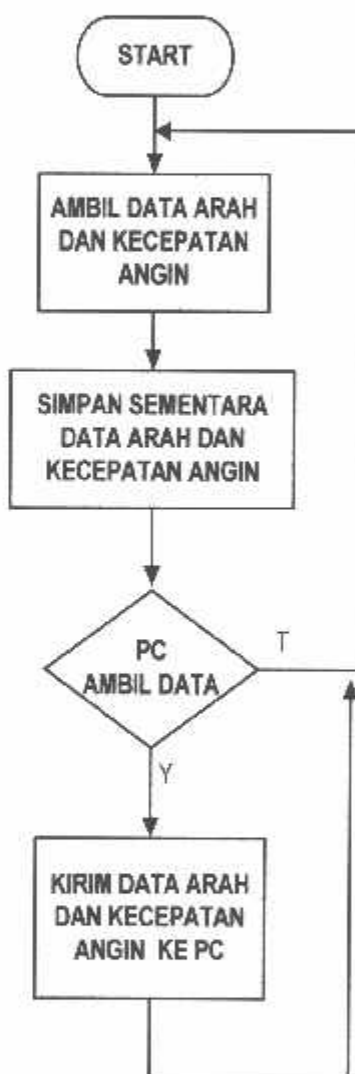
Gambar 3.8

Rangkaian Interface RS-232

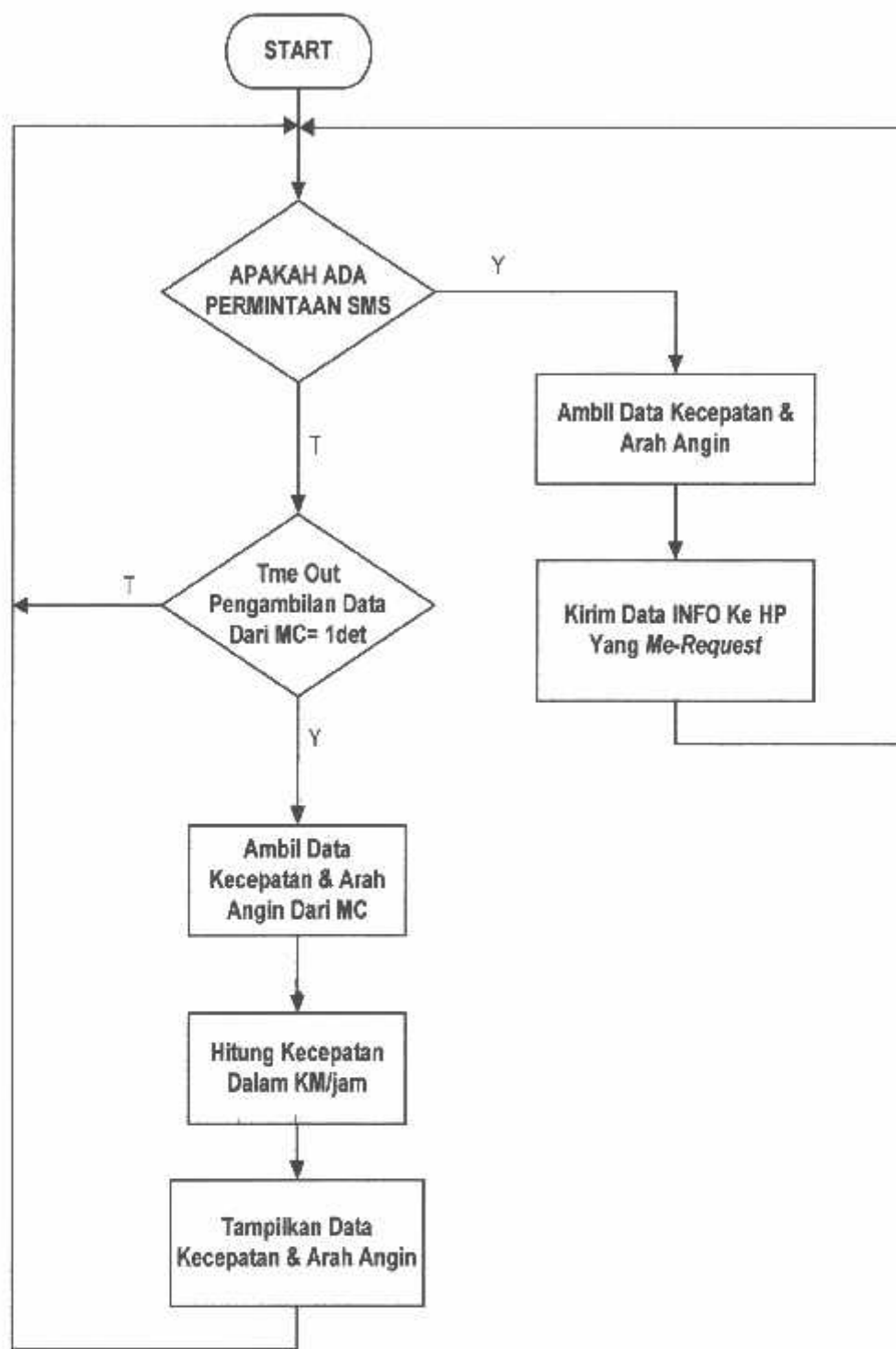
Data biner yang berasal dari komputer mempunyai level tegangan antara +3V sampai +15V dan -3V sampai -15V, sebelum masuk pada rangkaian modulator demodulator terlebih dahulu tegangannya melalui sebuah IC MAX 232 dan diubah menjadi tegangan TTL sebesar 0V sampai 5V yang sesuai dengan tegangan untuk mencatu kerja rangkaian modulator demodulator tersebut.

3.2.6. Perencanaan Perangkat Lunak

Perangkat lunak dirancang agar sistem dapat bekerja sesuai dengan perencanaan sistem secara keseluruhan. Bahasa pemrograman yang digunakan adalah *Bahasa Assembly Mikrokontroller Standard MCS-51*. Flowchart untuk pembuatan perangkat lunak dapat dilihat pada gambar di bawah ini



Gambar 3.9
Flowchart MK



Gambar 3.10
Flowchart PC

BAB IV

ANALISIS DAN PENGUJIAN ALAT

Dalam bab ini dibahas pengujian dan analisis alat pada masing-masing blok yang telah direncanakan dan pengujian sistem secara keseluruhan. Pengujian ini meliputi perangkat keras dan perangkat lunak sebagai penunjangnya. Hasil pengujian ini kemudian dianalisa dengan membandingkannya terhadap perencanaan. Pengujian dilakukan terhadap blok-blok sistem yang meliputi:

- Rangkaian Optocoupler
- Rangkaian Sensor Arah Angin
- Sistem Mikontroller AT89S8252.
- Komunikasi Serial Mikrokontroller dengan PC
- Pengujian Sistem Secara Keseluruhan.

4.1. Pengujian dan Pengukuran Optocoupler

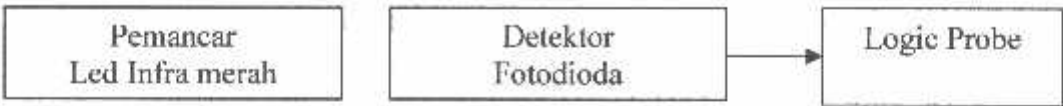
4.1.1. Tujuan:

Untuk mengetahui apakah optocoupler dapat bekerja dengan baik sesuai dengan perencanaan ataupun tidak.

4.1.2. Peralatan yang digunakan:

1. Logic Probe.
2. Rangkaian Optocoupler
3. Catu daya 5 volt.

4.1.3. Prosedur pengujian



Gambar 4.1 Rangkaian Pengujian Optocoupler
Sumber : Perancangan

1. Merangkai peralatan yang digunakan sesuai Gambar 4.1.
2. Memberikan catu daya 5 volt pada rangkaian Optocoupler berupa Led Infra Red dan fotodioda.
3. Mengarahkan berkas sinar ke Photodiode pada jarak sekitar 2 cm
4. Mengamati keluaran logic probe.
5. Berkas sinar ditutup dengan suatu penghalang.
6. Mengamati keluaran logic probe.

4.1.4. Hasil Pengujian

Hasil pengujian Optocoupler ditunjukkan dalam Tabel 4.2 berikut ini :

Tabel 4.1 Hasil Pengujian Optocoupler

Ada/Tidak Halangan	Keluar Logic Probe
Ada	Low
Tidak	High

4.1.5. Analisis Hasil Pengujian

Photodiode akan berlogika 0 (low) jika terdapat halangan antara pemancar (Tx) dan penerima (Rx), dan photodiode akan berlogika 1 (high) jika tidak terdapat penghalang antara Tx dan Rx.

Berdasarkan hasil pengujian di atas, terlihat bahwa Optocoupler tersebut dapat bekerja sesuai dengan perencanaan.

4.2. Pengujian dan Pengukuran Sensor Arah Angin

4.2.1. Tujuan

Untuk mengetahui apakah Rangkaian pemancar dan penerima dapat bekerja dengan baik sesuai dengan perencanaan ataukah tidak.

4.2.2. Peralatan yang digunakan

1. Rangkaian Pemancar dan Penerima Sensor arah angin
2. Logic probe
3. Catu Daya 5 Volt

4.2.3. Prosedur Pengujian



Gambar 4.2 Rangkaian Pengujian sensor arah angin
Sumber: Perancangan

1. Merangkai peralatan yang digunakan sesuai dengan gambar 4.2

- 2. Memberikan catu daya 5 Volt pada rangkaian sensor arah angin.
- 3. Mengarahkan berkas sinar ke photodiode pada sekitar 2 cm
- 4. Mengamati keluaran logic probe.
- 5. Berkas sinar ditutup dengan suatu penghalang.
- 6. Mengamati keluaran logic probe

4.2.4. Hasil Pengujian

Hasil pengujian Optocoupler ditunjukkan dalam Tabel 4.2 berikut ini :

Tabel 4.2 Hasil Pengujian Optocoupler

Ada/Tidak Halangan	Keluar Logic Probe
Ada	Low
Tidak	High

4.2.5. Analisis Hasil Pengujian

Photodiode akan berlogika 0 (low) jika terdapat halangan antara pemancar (Tx) dan penerima (Rx), dan photodiode akan berlogika 1 (high) jika tidak terdapat penghalang antara Tx dan Rx.

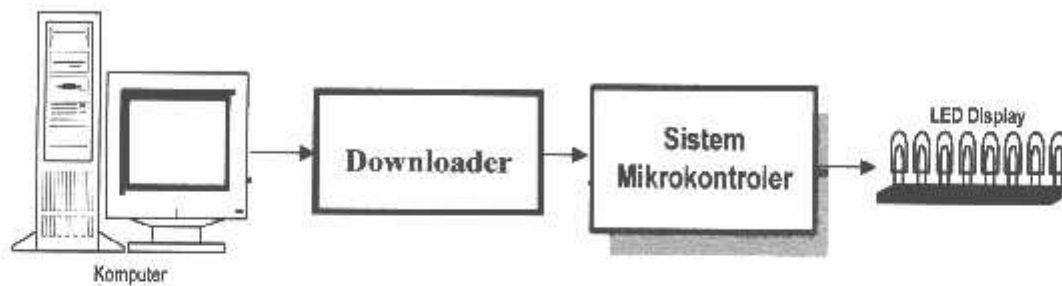
Berdasarkan hasil pengujian di atas, terlihat bahwa Optocoupler tersebut dapat bekerja sesuai dengan perencanaan.

4.3. Pengujian Mikrokontroller AT89S8252

4.3.1. Tujuan

Untuk mengetahui kondisi awal dari mikrokontroller apakah sudah sesuai dengan yang direncanakan.

4.3.2. Prosedur Pengujian



Gambar 4-3 Diagram blok Pengujian Mikrokontroler

1. Rangkaian dibuat seperti gambar 4.3
2. Memberikan catu daya 5 Volt.
3. Membuat program yang akan digunakan dalam pengujian mikrokontroller.

Program yang digunakan dalam pengujian mikrokontroler ini merupakan program sederhana yang meletakkan $0F_H$ dan $F0_H$ pada ACC secara bergantian kemudian memindahkannya pada *Port 1* AT89S8252. Program yang dibuat adalah sebagai berikut :

```

      ORG 0000H
      JMP  START
START: MOV  A,#0FH
      MOV  P1,A
      CALL TUNDA
  
```

```
MOV    A,#F0H
MOV    P1,A
JMP    START
TUNDA: MOV    R3,#0FFH
TUNDA1: MOV    R2,#0FFH
        DJNZ   R2,$
        MOV    R1,#0FH
        DJNZ   R1,$
        DJNZ   R3,TUNDA1
        RET
        END
```

- 4. Download program diatas
- 5. Mengamati keluaran pada LED display

4.3.3. Hasil Pengujian

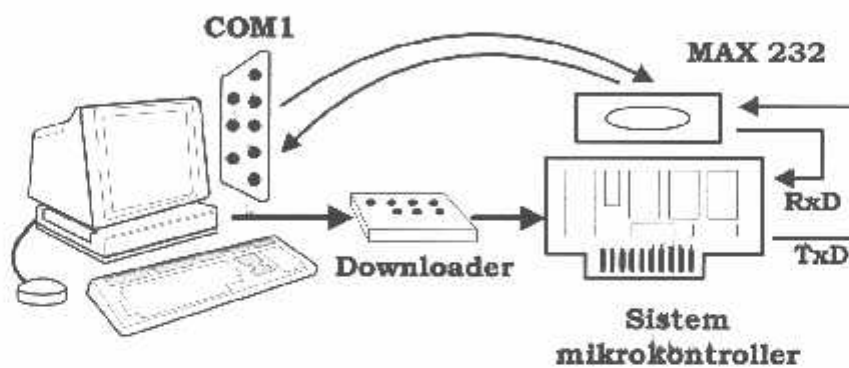
Tabel 4.3 Hasil Pengujian Sistem Mikrokontroler

Kondisi	Keluaran pada LED Display							
	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Satu	1	1	1	1	0	0	0	0
Dua	0	0	0	0	1	1	1	1

4.3.4. Analisis Hasil Pengujian

Dari hasil pengujian dalam tabel 4-1 dapat dilihat bahwa *port 1* memberikan logika 0F_H dan F0_H secara bergantian sesuai dengan isi program.

4.4. Pengujian Serial 232



Gambar 4-4 Rangkaian Pengujian Transfer Data

4.5.1. Tujuan

Untuk mengetahui apakah data yang dikirim dari MCU ke *PC* dapat diterima dengan benar dengan melakukan simulasi pada computer

4.5.2. Peralatan yang digunakan

- Komputer
- Sistem mikrokontroller dengan antarmuka RS232.

4.5.3. Prosedur pengujian

1. Menyusun rangkaian seperti pada Gambar 4-2.
2. Membuat program transfer data pada sistem mikrokontroller seperti yang ditunjukkan dalam Gambar 4-3 dan Gambar 4-4. Dengan program tersebut komputer mengirim data '1234567890' ke alat dan oleh alat akan dikembalikan lagi ke komputer.
3. Download program ke Mikrokontroler dan eksekusi program.
4. Mencatat hasil yang terlihat dalam layar komputer.

➤ Program Uji Komunikasi Data di Mikrokontroller

```

B2400 EQU 232
MULAI: CALL INIT232
      CALL SERIAL
      JMP MULAI
PCON EQU 87H
      ORG 0000H
      JMP MULAI
      ORG 0023H
SERIAL: JBC TI, OUT232
      PUSH ACC
      MOV A, SBUF
      MOV SBUF, A
      POP ACC
OUT232: CLR TI
      CLR RI
      RETI
INIT232: MOV TH1, #B1200
      MOV TMOD, #22H
      ANL PCON, #7FH
      MOV SCON, #50H
      SETB PS
      SETB ES

```

➤ Program Uji Komunikasi Data di Komputer

‘ Cuplikan program uji komunikasi data di komputer menggunakan program VB 6.0

```
MSComm1.Output = '1234567890'
```

```
Text1.Text = MSComm1.Input
```

4.5.4. Hasil Pengujian

Hasil pengujian transfer data serial ini ditunjukkan Tabel 4.4 dibawah ini :

Tabel 4.4 Hasil Pengujian Transfer Data Serial

Data yang dikirim komputer	Data yang diterima komputer
1234567890	1234567890

4.5.5. Analisis Hasil Pengujian

Hasil pengujian dalam Tabel 4.4 menunjukkan bahwa proses pengiriman data serial dengan menggunakan RS 232 ke alat telah benar.

4.5. Pengujian SMS Dikirim dan Diterima



Gambar 4-5. Rangkaian Pengujian SMS Dikirim Dan Diterima

4.5.1. Tujuan

Untuk mengetahui kelancaran sms terkirim dan sms diterima dari HP, baik Hp Operator ataupun Hp tujuan Request

4.5.2. Prosedur Pengujian

1. Merangkai seperti seperti pada gambar 4-5
2. Buka Kotak Pesan
3. Ketik INFO kirim ke no 081334465659
4. Kemudian Anda akan mendapatkan pesan baru berupa data Kecepatan dan Arah Angin

4.5.3. Hasil Pengujian

Hasil pengujian sms dikirim dan diterima ditunjukkan pada tabel 4-5

Banyaknya	Sukses	Gagal
30 kali	28	2

4.5.4. Analisis Hasil Pengujian

Keberhasilan SMS dikirim dan diterima ditentukan oleh jaringan masing-masing GSM. Percobaan dilakukan 30 kali, 28 sukses karena sinyal kuat/lancar dan 2 kali gagal karena tidak ada sinyal pada waktu itu

4.6. Pengujian Sistem Secara Keseluruhan

Pengujian ini dapat dilakukan setelah semua tahap pengujian perbagian telah dilakukan dengan baik. Adapun prosedur pengujian secara keseluruhan dapat dijelaskan sebagai berikut:

1. Pasang semua piranti perangkat keras.
2. Buka program aplikasi di komputer.
3. Lakukan seting pada port serial.

4. Ambil data

5. Kirim Data Via SMS

Jika semua langkah telah dilakukan dan tidak terdapat kesalahan maka pengujian secara keseluruhan telah dapat dilakukan dengan baik.

DATA HASIL PENGAMATAN DAN PENGUKURAN
 Sabtu, 25 Februari 2006

NO	Jam Pengamatan	Arah Angin Dari BMG (°C)	Kecepatan Angin Dari BMG (m/s)	Arah Angin Dari Alat (°C)	Kecepatan Angin Dari Alat (m/s)	Error (%)	
						Arah	Kecepatan
1	14:40	360	3.4	348,75	3.3	3.12	5.8
2	14:50	216	4.4	213,75	4.2	1.04	4.5
3	15:00	188	4.3	180	4.1	4.25	4.6
4	15:10	193	4.4	191,25	4.2	0.9	4.5
5	15:20	216	6.2	213,75	6	1.04	3.2
6	15:30	360	5.0	348,75	4.6	3.12	8
7	15:40	209	4.1	205,5	3.9	1.67	4.8
8	15:50	212	3.0	205,5	2.9	3.06	3.3
9	16:00	215	3.8	213,75	3.6	0.58	5.2
10	16:10	215	2.2	213,75	2.0	0.58	9
11	16:20	214	2.7	213,75	2.6	0.46	3.7
12	16:30	205	2.0	205,5	1.9	0.2	5
13	16:40	216	2.0	213,75	1.9	1.04	5
14	16:50	210	1.9	205,5	1.9	2.14	0
15	17:00	208	2.3	205,5	2.1	1.2	8.6
16	17:10	203	2.5	205,5	2.4	1.2	4
17	17:20	195	1.8	191,25	1.7	1.92	5.5
18	17:30	212	1.3	213,75	1.25	0.82	3.8
19	17:40	199	1.4	191,25	1.35	3.8	3.5
20	17:50	194	1.0	191,25	0.9	1.41	10
Rata- rata Error						1,67	5,1

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan Perancangan serta pengujian dari keseluruhan sistem, maka dapat disimpulkan sebagai berikut :

1. Keakurasian pengukuran Arah Angin dipengaruhi oleh pemasangan kincir angin dengan memperhatikan kode piringan arah angin tepat menunjukkan arah utara Utara sebagai acuan pertama.
2. Error pada pengukuran kecepatan dan Arah Angin dikalibrasi dengan sistem pengukuran yang dimiliki BMG adalah: untuk kecepatan 5,1% dan untuk Arah 1,67%.
3. Error pada kecepatan angin dipengaruhi oleh perbedaan tinggi kincir angin yang dimiliki BMG dengan kincir angin pada Alat
4. Piringan untuk sensor Arah angin dibagi dalam 32 segment (16 sisi gelap, 16 sisi terang). Sehingga memiliki beda sudut $11,25^{\circ}$. Semakin kecil beda sudutnya, data yang dihasilkan akan lebih akurat.

5.2. Saran

- Bahan yang digunakan untuk kincir angin diharapkan dari bahan yang ringan tapi kuat. Agar kecepatan yang kecilpun dapat dideteksi. Dan apabila ada angin yang kecepatannya besar, kincir angin bisa menahan tidak goyang atau rusak.
 - Untuk membuat kincir angin yang standard diperlukan perhitungan yang teliti.
-

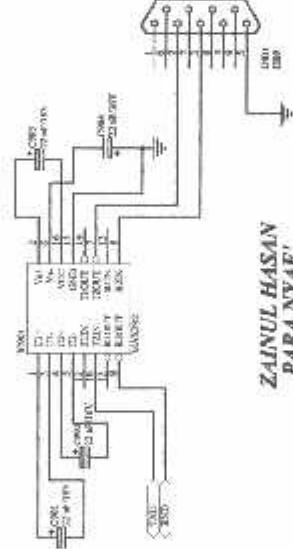
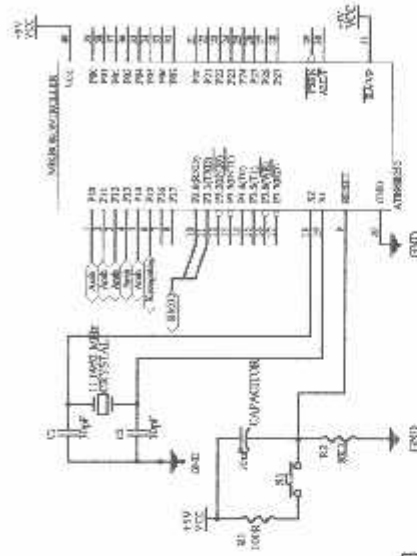
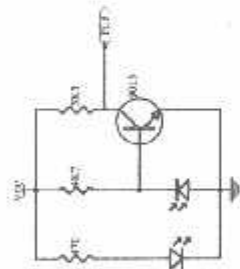
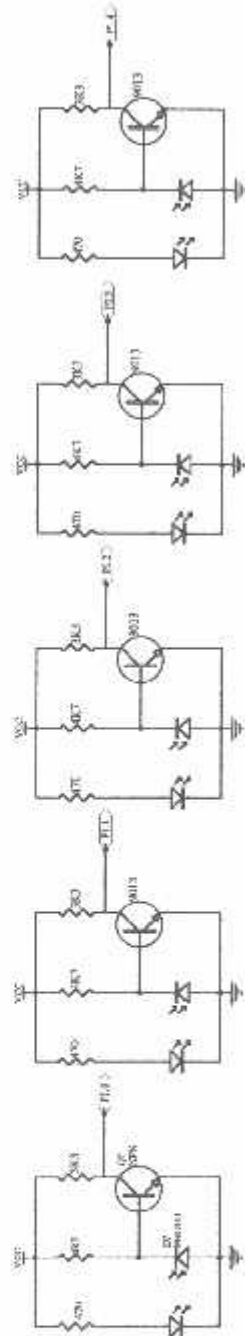
DAFTAR PUSTAKA

1. William David Coover, *Instrumentasi Elektronika dan Teknik Pengukuran*, Erlangga, Edisi ke-2, 1999
 2. William H. Hayt, Jr. Gerold W. Neudeck, *Electronic Circuit Analysis and Design*, Houghton Millin Company, Boston, 1976
 3. William H. Sahm III, Marvin W. Smith, *Optoelectronics*, General Electronic Company, 1982
 4. Moh. Ibnu Malik, ST, *Belajar Mikrokontroller AT89S8252*, Gaya Media , Edisi Pertama, 2003
 5. Eko Putra, Agfianto, 2002, *Belajar Mikrokontroler AT89C51/52/55*, Yogyakarta, Gramedia
 6. *Data Sheet*
 7. Roger L. Tokheim, *Elektronika Digital*, Erlangga, Edisi Kedua, 1995
 8. Malvino, *Prinsip-prinsip Elektronik*, Erlangga, Edisi Kedua, 1994
 9. wasito, *Vademikum Elektronika*, Gramedia, Edisi kedua, 2001
 10. Abdul Yadi, *Aplikasi Visual Basic dalam Industri Manufaktur*, PT Elex Media Komputindo, 2000
 11. Ario Suryo Kusumo, *Buku Latihan Microsoft Visual Basic 6.0*, PT Elex Media Komputindo, 2000
 12. Wiryanto Dewobroto, *Aplikasi Sain dan Teknik Dengan Visual Basic 6.0*. PT Elex Media Komputindo, 2002
 13. Bustam, *Aplikasi Berbasis SMS*, 2002
-

14. <http://www.google.com/>
15. <http://www.motorola.com/literature.pdf>
16. <http://www.maxim-ic.com.pdf>
17. www.siemens-mobile.com
18. www.dallas%20semiconductor.com

LAMPIRAN

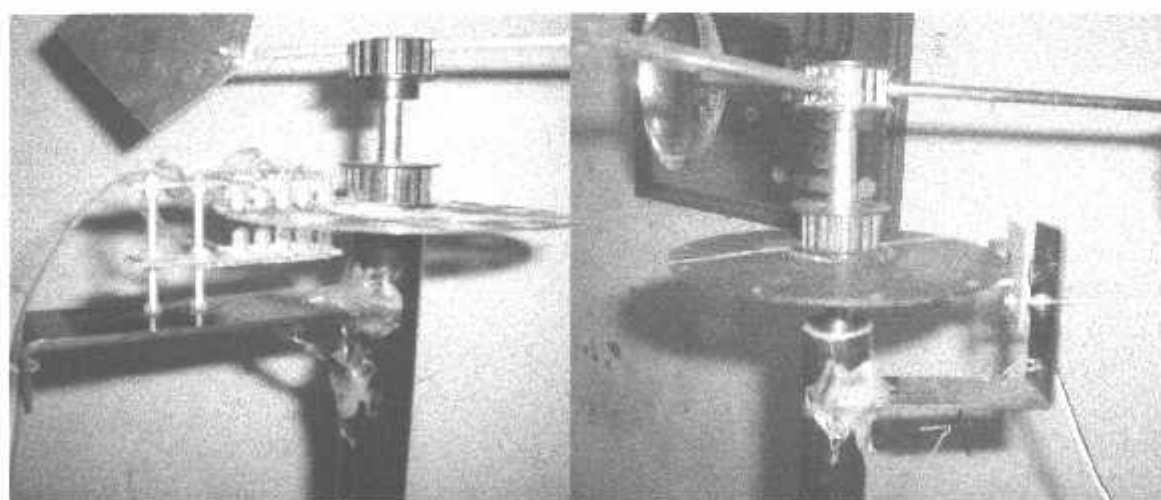
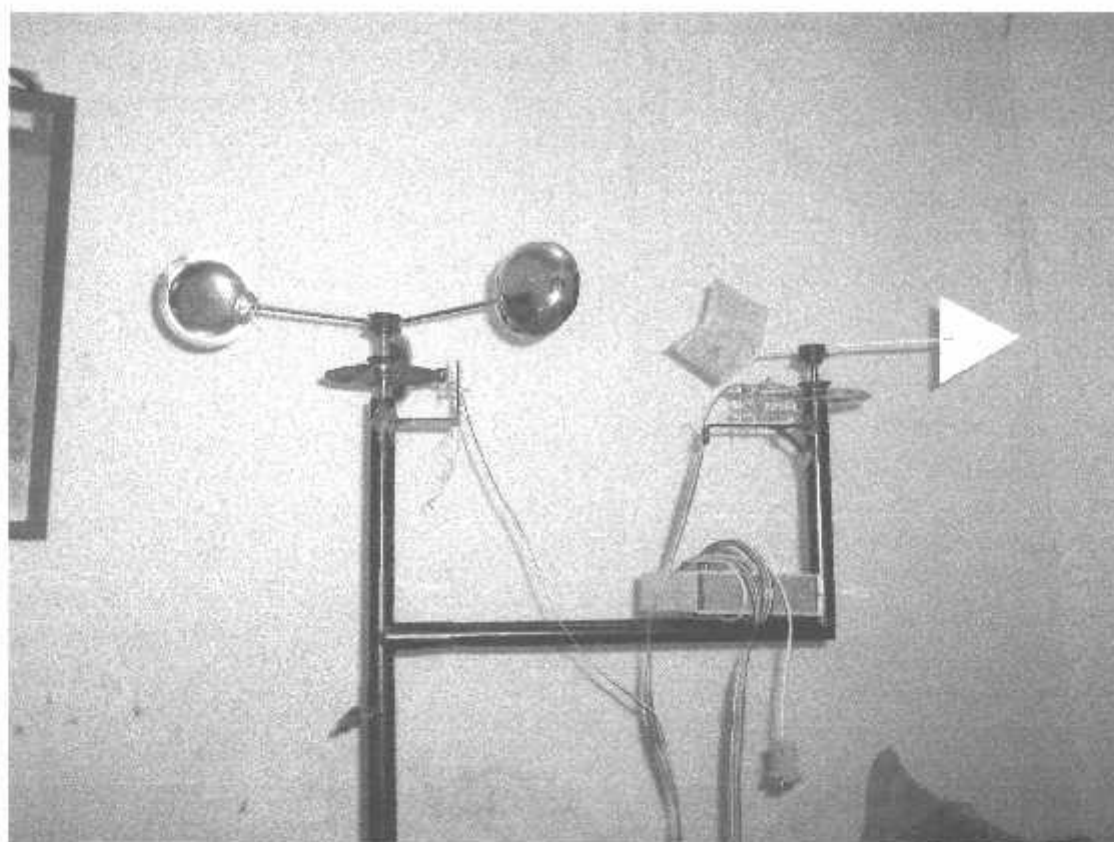
RANGKAIAN SISTEM MONITORING KECEPATAN DAN ARAH ANGIN YANG DAPAT DIAKSES VIA SMS BERBASIS AT89S8252



ZAINUL HASAN
PARA NYAE'

Revisi	
No.	Revisi
1	Revisi
2	Revisi
3	Revisi
4	Revisi
5	Revisi
6	Revisi
7	Revisi
8	Revisi
9	Revisi
10	Revisi

Disusun oleh: ZAINUL HASAN
Dibimbing oleh: PARA NYAE'





FORMULIR BIMBINGAN SKRIPSI

Nama : ZAINUL HASAN
Nim : 01.17.117
Masa Bimbingan : Mulai Tanggal 16 December 2005 s/d 16 Mei 2006
Judul : PERENCANAAN DAN PEMBUATAN SISTEM MONITORING
KECEPATAN DAN ARAH ANGIN YANG DAPAT DIAKSES VIA
SMS BERBASIS AT89S8252

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1	27/06	Bab I (perbaikan)	
2		Bab II	
3		Bab III	
4		Bab IV	
5		Bab V	
6		Penutup	
7		Bab VI	
8	13/3	Revisi	
9			
10			

Malang,

200...

Dosen Pembimbing

(Ir. F. Yudi I. Impraptono, MT)
NIP. Y 1039500274



FORMULIR PERBAIKAN SKRIPSI

Nama : Zainul Hasan
NIM : 01.17.117
Masa Bimbingan : 16 Desember 2005 s/d 16 Mei 2006
Judul : Perencanaan dan Pembuatan Sistem Monitoring Kecepatan
dan Arah Angin Yang Dapat Diakses Via SMS Berbasis
AT89S8252

NO	Tanggal	Uraian	Paraf
1	20 Maret 2006	Prosedur Pengujian Alat	
2	20 Maret 2006	Tabel Hasil Pengujian dan Pengukuran	
3	20 Maret 2006	Kesimpulan	

Disetujui

Penguji I

(Ir. Usman Djuanda, MM)

Penguji II

(M. Ashar, ST, MT)

Mengetahui,
Dosen Pembimbing

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



LEMBAR BIMBINGAN SKRIPSI

Nama : Zainul Hasan
NIM : 01.17.117
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Judul Skripsi : Perencanaan dan Pembuatan Sistem Monitoring
Kecepatan dan Arah Angin Yang Dapat Diakses
Via SMS Berbasis AT89S8252

Tanggal Pengajuan Skripsi : 20 Maret 2006
Selesai Penulisan Skripsi : 13 Maret 2006
Dosen Pembimbing : Ir. F.Yudi Limpraptono, MT
Telah Dievaluasi Dengan Nilai : 90 (Sembilan Puluh) 

Mengetahui,
Ketua Jurusan Teknik Elektro S-1



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

Diperiksa dan Disetujui,
Dosen Pembimbing



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



BADAN METEOROLOGI DAN GEOFISIKA
STASIUN KLIMATOLOGI KARANGPLOSO

Jl. Zentana 33. Karangploso - Malang Telp. (0341) 464827 - 461595 Fax. (0341) 464827
E-mail : zentana33@yahoo.com

SURAT PERNYATAAN

Nomor : UM. 207/1/12/KMlg-2006

Yang bertanda tangan di bawah ini :

Nama	:	HERY KOESNANTO
N I P	:	120 142 188
Pangkat/Gol.Ruang	:	PENATA MUDA / III a
Unit Kerja	:	STASIUN KLIMATOLOGI KARANGPLOSO MALANG

Menyatakan bahwa :

Nama	:	ZAINUL HASAN
Nim	:	01.17.117
Fakultas	:	Teknik Industri
Jurusan	:	Teknik Elektro S-1
Konsentrasi	:	Teknik Elektro
Universitas	:	Institut Teknologi Nasional Malang

Mahasiswa tersebut diatas telah melakukan pengujian alat di Stasiun Klimatologi Karangploso Malang pada tanggal 25 Februari 2006 dengan judul SISTEM MONITORING KECEPATAN DAN ARAH ANGIN YANG DAPAT DIAKSES VIA SMS BERBASIS AT89S8252 guna untuk mendapatkan data sebagai pembandingan dengan alat yang telah dibuat oleh mahasiswa yang bersangkutan.

Demikian surat ini dibuat dengan sebenarnya agar dapat dipergunakan sebagaimana mestinya.

Mengetahui
Kepala Stasiun Klimatologi
Karangploso Malang


Ir. ANTOYO SETYADIPRATIKTO
NIP. 120 106 886

Malang, 27 Februari 2006
Pendamping Lapangan,


HERY KOESNANTO
NIP. 120 142 188

LISTING MICRO

```
DataArah      Data      P1
PortKecepatan Bit      P1.5
CntPutar      Data      20h

      Mov      SP,#40h
      Call     Serial_Initialization
      Clr      RI
      Mov      CntPutar,#0
      Mov      P2,CntPutar

CekPutar:
      Mov      P2,CntPutar
      Jnb      PortKecepatan,CekPutar1
      Jmp      CekPC1
CekPutar1:
      Jb       RI,KirimDataToPC
      Jmp      CekPutar
;-----
CekPC1:
      Jb       RI,KirimDataToPC
      Jb       PortKecepatan,CekPC1
      Inc      CntPutar
      Mov      P2,CntPutar
      Call     Delay_Fix_10ms
      Call     Serial_Initialization
      Jmp      CekPutar

KirimDataToPC:
      Clr      RI
      Mov      A,SBUF
      Cjnc     A,'#@',CekPutar
      Mov      A,CntPutar
      Call     Serial_HEX
      Mov      A,'#8'
      Call     Serial_Transmit
      Mov      A>DataArah
      Anl      A,#15h
      Cpl      A
      Call     Serial_HEX
      Mov      A,'#$'
      Call     Serial_Transmit
      Call     Delay_Fix_10ms
      Mov      CntPutar,#0
      Clr      RI
      Call     Serial_Initialization
      Jmp      CekPutar

Serial_Receive:
      Jnb      RI,$
      Clr      RI
      Mov      A,SBUF
      Clr      RI
      ret
Serial_Transmit:
      Mov      SBUF,A
      Jnb      TI,$
```

```

        Clr      TI
    ret
Serial_Initialization:
    Mov    SCON,#50h
    Mov    TH1,#250      ; BAUDRATE 9600
    ;Mov    TH1,#208      ; BAUDRATE 1200
    Mov    R7h,#00h      ; PCON
    Mov    TMOD,#21h
    Mov    TCON,#01010000B ; RUN T1 AND T0
    ;Mov    IE,#10000010B
    ;Mov    IP,#00001000B
    ret
Serial_HEX:
    Push    07h
    Push    ACC

    Mov     7,A
    Anl     A,#0F0h
    Swap    A
    Orl     A,#30h
    Call    Tes_Huruf
    Call    Serial_Transmit
    Mov     A,7
    Anl     A,#0Fh
    Orl     A,#30h
    Lcall   Tes_Huruf
    Lcall   Serial_Transmit

    Pop     ACC
    Pop     07h
    Ret

    ret
Tes_Huruf:
    Cjne    A,#3Ah,tes_huruf_1
    Mov     A,#'A'
    Ret
tes_huruf_1:
    Cjne    A,#3Bh,tes_huruf_2
    Mov     A,#'B'
    Ret
tes_huruf_2:
    Cjne    A,#3Ch,tes_huruf_3
    Mov     A,#'C'
    Ret
tes_huruf_3:
    Cjne    A,#3Dh,tes_huruf_4
    Mov     A,#'D'
    Ret
tes_huruf_4:
    Cjne    A,#3Eh,tes_huruf_5
    Mov     A,#'E'
    Ret
tes_huruf_5:
    Cjne    A,#3Fh,tes_huruf_6
    Mov     A,#'F'
    Ret
tes_huruf_6:
    Ret

```

```

    ret

Serial_Transmit_String:
    Push    ACC
    Push    DPL
    Push    DPH
xgetcarl:
    Clr     A           ; mengambil data dari eprom
    Movc    A,@A+DPTR
    Cjne    A,#0,xtamppill ; tes apakah data habis?
    Ljmp    xmettul

xtamppill:
           ; keluarkan data ke lcd
    Lcall   Serial_Transmit
    Inc     DPTR         ; naikkan dptr
    Ljmp    xgetcarl
xmettul:
    Pop     DPH
    Pop     DPL
    Pop     ACC
    ret

Serial_Transmit_String_Ram:
    Push    ACC
    Push    DPL
    Push    DPH
axgetcarl:
    Movx    A,@DPTR

    Cjne    A,#0,axtamppill ; tes apakah data habis?
    Ljmp    axmettul

axtamppill:
           ; keluarkan data ke lcd
    Call    Serial_Transmit
    Inc     DPTR         ; naikkan dptr
    Ljmp    axgetcarl
axmettul:
    Pop     DPH
    Pop     DPL
    Pop     ACC
    ret

;===== DELAY =====
Delay_Var_1ms:
    Call    Delay_Fix_1ms
    Djnz    R0,Delay_Var_1ms
    ret
Delay_Var_10ms:
    Call    Delay_Fix_10ms
    Djnz    R0,Delay_Var_10ms
    ret
Delay_Var_100ms:
    Call    Delay_Fix_100ms
    Djnz    R0,Delay_Var_100ms
    ret
Delay_Var_1s:
    Call    Delay_Fix_1s
    Djnz    R0,Delay_Var_1s
    ret
Delay_Var_10s:

```

```

        Call    Delay_Fix_1Cs
        Djnz    R0,Delay_Var_1Cs
    ret
Delay_Var_10us:
    Call    Delay_Fix_10us
    Djnz    R0,Delay_Var_10us
    ret
Delay_Fix_10us:
    Push    1
    Mov     1,#20
    Djnz    1,$
    Pop     1
    ret
Delay_Fix_10s:
    Push    1
    Mov     1,#100
delay_fix_10s_1:
    Call    Delay_Fix_100ms
    Djnz    1,delay_fix_10s_1
    Pop     1
    ret
Delay_Fix_1s:
    Push    1
    Mov     1,#100
delay_fix_1000ms_1:
    Call    Delay_Fix_10ms
    Djnz    1,delay_fix_1000ms_1
    Pop     1
    ret
Delay_Fix_100ms:
    Push    1
    Mov     1,#10
delay_fix_100ms_1:
    Call    Delay_Fix_10ms
    Djnz    1,delay_fix_100ms_1
    Pop     1
    ret
Delay_Fix_10ms:
    Mov     TMOD,#00000001b    ; Timer 1 bekerja pada mode 1
    Mov     TL0,#3Dh    ; siapkan waktu tunda 5C mili-detik
    Mov     TR0,#0B0h
    Clr     TF0                ; me-nol-kan bit limpahan
    Setb    TR0                ; timer mulai bekerja
    Jnb     TF0,$             ; tunggu di sini sampai melimpah
    Clr     TR0                ; timer berhenti kerja
    Ret
    ret
Delay_Fix_1ms:
    Mov     TMOD,#00000001b    ; Timer 1 bekerja pada mode 1
    Mov     TL0,#0EDh    ; siapkan waktu tunda 5C mili-detik
    Mov     TH0,#78h
    Clr     TF0                ; me-nol-kan bit limpahan
    Setb    TR0                ; timer mulai bekerja
    Jnb     TF0,$             ; tunggu di sini sampai melimpah
    Clr     TR0                ; timer berhenti kerja
    Ret
    ret

```

LISTING VB

```
Dim StrInfo As String
Dim SerialTampung As String
Dim Mengirim As Boolean
Dim Membedah As Boolean
Dim AbaikanChk As Boolean
Dim FlagComm As Boolean
Dim FlagToggleDSR As Boolean
Dim FlagDSR As Boolean
Dim CntDSR As Long
Dim Ret As Integer      ' Scratch integer.
Dim Temp As String      ' Scratch string.
Dim hLogFile As Integer ' Handle of open log file.
Dim StartTime As Date   ' Stores starting time for port timer

Private Sub CmdConnect_Click()

    MSComm2.CommPort = cmbPortMC.Text
    CmdDisconnect.Enabled = True
    CmdConnect.Enabled = False
    MSComm2.CommPort = 4
    MSComm2.PortOpen = True
    tmrGetMC.Enabled = True

End Sub

Private Sub cmdDelete_Click()

    CMSave.Text = ""
    CMSave.SaveFile App.Path & "\data.txt", True

End Sub

Private Sub CmdDisconnect_Click()

    tmrGetMC.Enabled = False
    CmdDisconnect.Enabled = False
    CmdConnect.Enabled = True
    MSComm2.PortOpen = False

End Sub

Private Sub cmdSave_Click()
    CMSave.SaveFile App.Path & "\data.txt", True
End Sub

Private Sub Command1_Click()
Dim a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u,
v, w, X, Y, z
Dim s0, s1, s2, s3, s4, s5, s6, s7, s8, s9
Dim xx As MSComctlLib.ListItem
Dim s10, s12, s13, s14, s15, s16, s17, s18, s19
Dim s20, s22, s23, s24, s25, s26, s27, s28, s29
Dim s30, s32, s33, s34, s35, s36, s37, s38, s39
Dim s40, s42, s43, s44, s45, s46, s47, s48, s49
Dim who
Dim EA As Variant
```

```

Dim SMSMasuk As SMSIn
Set fs = CreateObject("Scripting.FileSystemObject")
Dim sPath As String
Dim OldFile, NewFile
Dim rsImage As Recordset
Dim campo As Field
Dim campo1 As Field
Dim Temp As String
Dim SSq1
Dim XX1 As Variant
Dim JmlMK() As String
Dim NamaMK() As String
Dim JmlSKS() As Integer
Dim KodeMK() As String
Dim LokasiAwal, LokasiAkhir
Dim KodeMikrolet
Dim CntMikrolet
Dim PesanBalik

CMM.OpenFile App.Path + "\data.txt"
a = Text4.Text
lblSMS.Caption = a
b = InStr(1, LCase(a), "info")
If b = 0 Then
    Exit Sub
End If
'----- CEK DATA Lokasi -----
'

StrInfo = "Kec: " & txtKecepatan.Text & " km/jam, arah: " &
txtArah.Text & " derajat."
lblHP.Caption = Text3.Text

lblTanggal.Caption = Text5.Text & ", " & Text6.Text
Text1.Text = StrInfo
KirimSMS
End Sub
Private Function cekDataSMS(SMS) As Boolean
Dim a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u,
v, w, X, Y, z
Dim s0, s1, s2, s3, s4, s5, s6, s7, s8, s9
Dim xx As MSComctlLib.ListItem
Dim s10, s12, s13, s14, s15, s16, s17, s18, s19
Dim s20, s22, s23, s24, s25, s26, s27, s28, s29
Dim s30, s32, s33, s34, s35, s36, s37, s38, s39
Dim s40, s42, s43, s44, s45, s46, s47, s48, s49
Dim who
Set fs = CreateObject("Scripting.FileSystemObject")
Dim sPath As String
Dim OldFile, NewFile
Dim rsImage As Recordset
Dim campo As Field
Dim Temp As String
Dim SSq1
End Function
Private Sub Command2_Click()
Dim a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u,
v, w, X, Y, z
Dim s0, s1, s2, s3, s4, s5, s6, s7, s8, s9

```

```

Dim xx As MSComctlLib.ListItem
Dim s10, s12, s13, s14, s15, s16, s17, s18, s19
Dim s20, s22, s23, s24, s25, s26, s27, s28, s29
Dim s30, s32, s33, s34, s35, s36, s37, s38, s39
Dim s40, s42, s43, s44, s45, s46, s47, s48, s49
'----- PDU Isi SMS -----

a = CDec(Len(Text1.Text))
If a > 155 Then Text1.Text = Mid(Text1.Text, 1, 150)
a = CDec(Len(Text1.Text))
s30 = a Mod 8
If s30 = 1 Then
    Text1.Text = Text1.Text & " "
ElseIf s30 = 0 Then
    Text1.Text = Text1.Text & " "
End If
a = CDec(Len(Text1.Text))
b = a * 8 - (a * 7)
d = b Mod 8
c = a
X = d - 1
Text2.Text = ""
For i = 1 To a
    s1 = Mid(Text1.Text, c, 1)
    s2 = Asc(s1)
    s3 = Hex(s2)
    s4 = BINER7(s2)
    If i = 1 Then
        s10 = ""
        For Y = 1 To d
            s10 = s10 & "0"
        Next Y

        s5 = s10 & s4
        s6 = BinToHex(Mid(s5, 1, 8))
        Text2.Text = s6 & Text2.Text
        If X < 0 Then
            Else
                s7 = Right(s5, X)
            End If
        Else
            s5 = s7 + s4
            s6 = BinToHex(Mid(s5, 1, 8))
            s7 = Right(s5, X)
            If s7 = "" Then
                s7 = "0"
            End If
            If X <> 7 Then Text2.Text = s6 & Text2.Text
        End If
        X = X - 1
        If X < 0 Then X = 7
        c = c - 1
    Next i
a = Hex(Len(Text1.Text))
If Len(a) = 1 Then a = "0" + a
Text2.Text = a & Text2.Text
'----- PDU SMS Center -----
a = Len(txtSMSCenter.Text)
b = a Mod 2

```

```

c = a \ 2
If b <> 0 Then c = c + 1
txtSMSCenterPDU.Text = "91"
For i = 1 To c
    d = Mid(txtSMSCenterPDU.Text, 1 + (2 * (i - 1)), 2)
    If Len(d) = 1 Then
        e = "F" & d
    Else
        e = balik(d)
    End If
Next i
a = Len(txtSMSCenterPDU.Text) \ 2
b = Hex(a)
If Len(b) = 1 Then b = "0" & a
txtSMSCenterPDU.Text = b & txtSMSCenterPDU.Text
'----- PDU HP Penerima -----
a = Len(txtHPPenerimaPDU.Text)
b = a Mod 2
c = a \ 2
If b <> 0 Then c = c + 1
txtHPPenerimaPDU.Text = "91"
For i = 1 To c
    d = Mid(txtHPPenerimaPDU.Text, 1 + (2 * (i - 1)), 2)
    If Len(d) = 1 Then
        e = "F" & d
    Else
        e = balik(d)
    End If
    txtHPPenerimaPDU.Text = txtHPPenerimaPDU.Text & e
Next i
a = Len(txtHPPenerimaPDU.Text)
b = Hex(a)
txtHPPenerimaPDU.Text = b & txtHPPenerimaPDU.Text
'----- Gabung PDU -----
a = txtSMSCenterPDU.Text & "01" & "00" & txtHPPenerimaPDU.Text &
"00" & "00" & Text2.Text
txtPDUTotal.Text = a
lblCMGS.Caption = (Len(a) - Len(txtSMSCenterPDU.Text)) \ 2
End Sub
Private Function BinToHex(Biner As String) As String
Dim a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u,
v, w, X, Y, z
Dim s0, s1, s2, s3, s4, s5, s6, s7, s8, s9
s1 = 0
j = 0
For i = 7 To 0 Step -1
    a = CDec(Mid(Biner, 8 - i, 1))
    b = a * 2 ^ i
    s1 = s1 + b
Next i
BinToHex = Hex(s1)
If Len(BinToHex) = 1 Then
    BinToHex = "0" & BinToHex
End If
End Function

```



```

Private Sub Command3_Click()
Dim a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u,
v, w, X, Y, z
Dim s0, s1, s2, s3, s4, s5, s6, s7, s8, s9
Dim s10, s11, s12, s13, s14, s15, s16, s17, s18, s19
Dim s20, s21, s22, s23, s24, s25, s26, s27, s28, s29
Dim s30, s31, s32, s33, s34, s35, s36, s37, s38, s39
Dim s40, s41, s42, s43, s44, s45, s46, s47, s48, s49
Dim str1
Dim Lokasi

Dim xx As MSComctlLib.ListItem
Dim waktu As Date
Dim Buffer$
Dim EA As Variant
Dim Data
Dim data1

Dim l1, l2 As Long
lblStatusKirim.Caption = "Send..."
txtTerm.Text = ""
SerialTampung = ""
a = a
a = MSComm1.CommPort
MSComm1.DTREnable = True
MSComm1.InputLen = 0
waktu = Now
Do
    txtTerm.Text = ""
    txtTerm.Text = ""
    MSComm1.Output = "ATE1" & Chr$(13)
    Do
        DoEvents
    Loop Until InStr(txtTerm.Text, "OK") Or InStr(txtTerm.Text,
"ERROR")
    Sleep 100
Loop Until InStr(txtTerm.Text, "OK") Or InStr(txtTerm.Text,
"ERROR")
If InStr(txtTerm.Text, "ERROR") Then
    MsgBox "Error pada komunikasi PC dengan HP. Ulangi lagi!",
vbCritical, "Error"
    Membedah = False
    Exit Sub
ElseIf InStr(txtTerm.Text, "OK") Then
    txtTerm.Text = ""
    txtTerm.Text = ""
    SerialTampung = ""
    waktu = Now
    Do
        txtTerm.Text = ""
        txtTerm.Text = ""
        MSComm1.Output = "AT+CMGS=" & lblCMGS.Caption & Chr$(13)
        Do
            DoEvents
        Loop Until InStr(txtTerm.Text, ">") Or InStr(txtTerm.Text,
"ERROR")
        Loop Until InStr(txtTerm.Text, ">") Or InStr(txtTerm.Text,
"ERROR")
        If InStr(txtTerm.Text, "ERROR") Then

```

```

        lblStatusKirim.Caption = "Not Sent"
        Membedah = False
        Exit Sub
    End If
    txtTerm.Text = ""
    txtTerm.Text = ""
    SerialTampung = ""
    waktu = Now
    Do
        txtTerm.Text = ""
        txtTerm.Text = ""
        a = Len(txtPDUTotal.Text)
        MSComm1.Output = txtPDUTotal.Text & Chr$(26)
        Do
            DoEvents
            Loop Until InStr(txtTerm.Text, "OK") Or
InStr(txtTerm.Text, "ERROR")
            Loop Until InStr(txtTerm.Text, "OK") Or InStr(txtTerm.Text,
"ERROR")
            If InStr(txtTerm.Text, "ERROR") Then
                lblStatusKirim.Caption = "Not Sent"
                Membedah = False
                Exit Sub
            End If
            lblStatusKirim.Caption = "Delivered"
        Else
            MsgBox "Error pada perangkat keras antara PC dan HP. Periksa
kabel dan ulangi lagi!", vbCritical, "Error"
            Membedah = False
            Exit Sub
        End If
        'txtTerm.Text = SerialTampung
        CMTerm.Text = txtTerm.Text
    End Sub

Private Sub Command4_Click()
    CMM1.SaveFile App.Path + "\data.txt", True
End Sub

Private Sub Command5_Click()
    MSComm1.PortOpen = True
End Sub

Private Sub Form_Load()
    Dim EA As Variant
    Dim a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u,
v, w, X, Y, z
    Dim s0, s1, s2, s3, s4, s5, s6, s7, s8, s9
    Dim xx As MSComctlLib.ListItem
    Dim s10, s12, s13, s14, s15, s16, s17, s18, s19
    Dim s20, s22, s23, s24, s25, s26, s27, s28, s29
    Dim s30, s32, s33, s34, s35, s36, s37, s38, s39
    Dim s40, s42, s43, s44, s45, s46, s47, s48, s49
    Dim who
    Set fs = CreateObject("Scripting.FileSystemObject")
    Dim sPath As String
    Dim OldFile, NewFile
    Dim rsImage As Recordset
    Dim campo As Field

```

```

Dim Temp As String
Dim SSq1

'On Error Resume Next
CheckDelete.Value = 1

CMM1.OpenFile App.Path + "\data.txt"
cmbSMS.ListIndex = 0

'-----
Settings = GetSetting(App.Title, "Properties", "Settings", "")
' frmTerminal.MSComm1.Settings\
If Settings <> "" Then
    MSComm1.Settings = Settings
    If Err Then
        MsgBox Error$, 48
        Exit Sub
    End If
End If

CommPort = GetSetting(App.Title, "Properties", "CommPort", "")
' frmTerminal.MSComm1.CommPort
If CommPort <> "" Then MSComm1.CommPort = CommPort

Handshaking = GetSetting(App.Title, "Properties",
"Handshaking", "") 'frmTerminal.MSComm1.Handshaking
If Handshaking <> "" Then
    MSComm1.Handshaking = Handshaking
    If Err Then
        MsgBox Error$, 48
        Exit Sub
    End If
End If

Echo = GetSetting(App.Title, "Properties", "Echo", "") ' Echo
On Error GoTo 0

MSComm1.Settings = "19200,n,8,1"

If Dir(App.Path & "\data.txt") = "" Then
    CMSave.SaveFile App.Path & "\data.txt", True
    a = a
Else
    CMSave.OpenFile App.Path & "\data.txt"
End If

End Sub

Private Sub MSComm1_OnComm()
    Select Case MSComm1.CommEvent
        Case comEvReceive
            Dim Buffer As Variant
            Buffer = MSComm1.Input
            'Debug.Print "Receive - " & StrConv(Buffer, vbUnicode)
            ShowData txtTerm, (StrConv(Buffer, vbUnicode))
            'SerialTampung = SerialTampung & (StrConv(Buffer,
vbUnicode))
            'ShowData txtTerm, (StrConv(buffer, vbUnicode))
    End Select

```

```

End Sub
Private Sub MSComm2_OnComm()
    Select Case MSComm2.CommEvent
        Case comEvReceive
            Dim Buffer As Variant
            Buffer = MSComm2.Input
            'Debug.Print "Receive - " & StrConv(Buffer, vbUnicode)
            ShowData txtTerm1, (StrConv(Buffer, vbUnicode))
            'SerialTampung = SerialTampung & (StrConv(Buffer,
vbUnicode))
            'ShowData txtTerm, (StrConv(buffer, vbUnicode))
    End Select
End Sub

Private Sub Text1_Change()
    lblCountIsi.Caption = Len(Text1.Text)
End Sub

Private Sub Text3_Change()
    txtHPPenerima.Text = Text3.Text
End Sub

Private Sub Timer1_Timer()
    Dim a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u,
v, w, X, Y, z
    Dim s0, s1, s2, s3, s4, s5, s6, s7, s8, s9
    Dim s10, s11, s12, s13, s14, s15, s16, s17, s18, s19
    Dim s20, s21, s22, s23, s24, s25, s26, s27, s28, s29
    Dim s30, s31, s32, s33, s34, s35, s36, s37, s38, s39
    Dim s40, s41, s42, s43, s44, s45, s46, s47, s48, s49
    Dim str1
    Dim Lokasi

    Dim xx As MSComctlLib.ListItem
    Dim waktu As Date
    Dim Buffer$
    Dim EA As Variant
    Dim Data
    Dim data1

    Dim l1, l2 As Long

    a = MSComm1.DTREnable

    Membedah = True
    If Mengirim = True Then Exit Sub
    Timer1.Enabled = False
    txtTerm.Text = ""
    SerialTampung = ""
    a = a
    a = MSComm1.Settings
    MSComm1.DTREnable = True
    MSComm1.InputLen = 0
    waktu = Now
    lblSMS.Caption = ""
    Sleep 10
    If XPButton3.Enabled = False Then Exit Sub
    Do
        MSComm1.Output = "ATZ" & Chr$(13)

```

```

Do
    DoEvents
    Loop Until InStr(txtTerm.Text, "OK") Or InStr(txtTerm.Text,
"ERROR") Or Now > DateAdd("s", 5, waktu)
    Sleep 100
Loop Until InStr(txtTerm.Text, "OK") Or InStr(txtTerm.Text,
"ERROR") Or Now > DateAdd("s", 5, waktu)
If InStr(txtTerm.Text, "ERROR") Then
    MsgBox "Error pada komunikasi PC dengan HP. Ulangi lagi!",
vbCritical, "Error"
    Membedah = False
    Exit Sub
ElseIf InStr(txtTerm.Text, "OK") Then
    txtTerm.Text = ""
    txtTerm.Text = ""
    SerialTampung = ""
    waktu = Now
    Do
        MSComm1.Output = "AT+CMGL=" & cmbSMS.ListIndex & Chr$(13)
        Do
            DoEvents
            Loop Until InStr(txtTerm.Text, "OK") Or
InStr(txtTerm.Text, "ERROR") Or Now > DateAdd("s", 5, waktu)
            Loop Until InStr(txtTerm.Text, "OK") Or InStr(txtTerm.Text,
"ERROR") Or Now > DateAdd("s", 5, waktu)
        Else
            MsgBox "Error pada perangkat keras antara PC dan HP. Periksa
kabel dan ulang lagi!", vbCritical, "Error"
            Membedah = False
            Exit Sub
        End If
        'txtTerm.Text = SerialTampung
        CMTerm.Text = txtTerm.Text
        shapeTerima.BackColor = vbRed
        a = CMTerm.Text
        EA = Split(a, "+CMGL:")
        s1 = UBound(EA)
        s30 = 0
        If s1 <> 0 Then
            For i = 1 To s1
                Membedah = True
                Do
                    DoEvents
                    Loop Until Mengirim = False
                    shapeTerima.BackColor = vbRed
                    DoEvents
                    Lokasi = CMTerm.GetLine((2 * i))
                    If Lokasi = "" Then
                        MsgBox "Terdapat kesalahan pembacaan SMS."
                        Exit For
                    End If
                    data1 = CMTerm.GetLine((2 * i) + 1)
                    s40 = ExtractSMS(data1)
                    Command1_Click
                    s20 = InStr(1, Lokasi, ":")
                    s21 = InStr(1, Lokasi, ",")
                    s22 = Trim(Mid(Lokasi, s20 + 1, s21 - s20 - 1))
                    If CheckDelete.Value = 1 Then
                        Else

```

```

        txtTerm.Text = ""
        SerialTampung = ""
        'Sleep 1000
        txtTerm.Text = ""
        SerialTampung = ""
        a = a
        a = MSComm1.CommPort
        Do
            MSComm1.Output = "ATE1" & Chr$(13)
            Do
                DoEvents
                Loop Until InStr(txtTerm.Text, "OK") Or
                InStr(txtTerm.Text, "ERROR") Or Now > DateAdd("s", 5, waktu)
                Sleep 100
                Loop Until InStr(txtTerm.Text, "OK") Or
                InStr(txtTerm.Text, "ERROR") Or Now > DateAdd("s", 5, waktu)
                If InStr(txtTerm.Text, "OK") Then
                    'MSComm1.Output = "AT+CMGD=" & s22
                End If

                'MSComm1.Output = "AT+CMGD=" & s22
            End If
            s30 = s12 + 1
        Next i
    End If
    Timer1.Interval = 500
    Timer1.Enabled = True
End Sub

Function Swap(str1) As String
    Dim a, b, c, d, e, f, g, n, i
    Dim j, k, l, m, n, o, p, q, r, s, t, u, v, w, X, Y, z
    Dim hasil
    a = str1
    b = Len(a) / 2
    c = 0
    hasil = ""
    For i = 1 To b
        d = Mid(a, c + 1, 2)
        e = Mid(d, 1, 1)
        f = Mid(d, 2, 1)
        w = f & e
        hasil = hasil & w
        c = c + 2
    Next i
    Swap = hasil
End Function
'function ini untuk mengubah integer menjadi string 7 bit
Function BINER7(ByVal angka As Integer) As String
    If angka > 1 Then
        Dim i As Integer
        Dim hasil(8) As Integer
        Dim SISA(8) As Integer
        i = 1
        Do
            hasil(i) = Int(angka / 2)
            SISA(i) = angka Mod 2
            angka = hasil(i)
            i = i + 1
        Loop
    End If
    BINER7 = ""
    For i = 8 To 1 Step -1
        BINER7 = SISA(i) & BINER7
    Next i
End Function

```

```

        BINER7 = SISA(i - 1) & BINER7
    Loop Until hasil(i - 1) < 2
    BINER7 = hasil(i - 1) & BINER7
    BINER7 = String$(7 - Len(BINER7), "0") & BINER7
ElseIf angka = 1 Then
    BINER7 = "0000001"
ElseIf angka = 0 Then
    BINER7 = "0000000"
End If
End Function

'function ini untuk mengubah integer menjadi string 4 bit
Function BINER4(ByVal angka As Integer) As String
If angka > 1 Then
    Dim i As Integer
    Dim hasil(5) As Integer
    Dim SISA(5) As Integer
    i = 1
    Do
        hasil(i) = Int(angka / 2)
        SISA(i) = angka Mod 2
        angka = hasil(i)
        i = i + 1
        BINER4 = SISA(i - 1) & BINER4
    Loop Until hasil(i - 1) < 2
    BINER4 = hasil(i - 1) & BINER4
    BINER4 = String$(4 - Len(BINER4), "0") & BINER4
ElseIf angka = 1 Then
    BINER4 = "0001"
ElseIf angka = 0 Then
    BINER4 = "0000"
End If
End Function

'function ini untuk mengubah string 7 bit menjadi integer
Function DEBINER7(ByVal huruf As String) As Integer
Dim angka(8) As Integer
Dim i
For i = 1 To 7
    angka(i) = Val(Mid(huruf, i, 1))
    DEBINER7 = DEBINER7 + angka(i) * 2 ^ (7 - i)
Next i
End Function

'function ini untuk mengubah string 4 bit menjadi integer
Function DEBINER4(ByVal huruf As String) As Integer
Dim angka(5) As Integer
Dim i
For i = 1 To 4
    angka(i) = Val(Mid(huruf, i, 1))
    DEBINER4 = DEBINER4 + angka(i) * 2 ^ (4 - i)
Next i
End Function

'function ini untuk mengubah suatu karakter menjadi bilangan
'ASCII. untuk beberapa karakter tertentu, oleh forum SMS sedunia
'diadakan perubahan.
Function T00TH(ByVal karakter As String) As Integer
Dim pc As Integer

```

```

pc = Asc(karakter)
If pc = 64 Then
    TOOTH = 0
ElseIf pc = 36 Then
    TOOTH = 2
ElseIf pc = 223 Then
    TOOTH = 30
ElseIf pc >= 32 And pc <= 35 Then
    TOOTH = pc
ElseIf pc >= 37 And pc <= 63 Then
    TOOTH = pc
ElseIf pc = 95 Then
    TOOTH = 64
ElseIf pc >= 65 And pc <= 90 Then
    TOOTH = pc
ElseIf pc = 196 Then
    TOOTH = 91
ElseIf pc = 214 Then
    TOOTH = 92
ElseIf pc = 220 Then
    TOOTH = 94
ElseIf pc = 168 Then
    TOOTH = 96
ElseIf pc >= 97 And pc <= 122 Then
    TOOTH = pc
ElseIf pc = 228 Then
    TOOTH = 123
ElseIf pc = 246 Then
    TOOTH = 124
ElseIf pc = 252 Then
    TOOTH = 126
End If
End Function

```

'function ini untuk mengubah suatu bilangan ascii menjadi
'suatu karakter. Untuk beberapa karakter tertentu, oleh
'forum SMS sedunia, diadakan perubahan

```

Function DETOOTH(ByVal pc As Integer) As String
Dim TOOTH As Integer

```

```

If pc = 0 Then
    TOOTH = 64
ElseIf pc = 2 Then
    TOOTH = 36
ElseIf pc = 4 Then
    TOOTH = 232
ElseIf pc = 30 Then
    TOOTH = 223
ElseIf pc >= 32 And pc <= 35 Then
    TOOTH = pc
ElseIf pc >= 37 And pc <= 63 Then
    TOOTH = pc
ElseIf pc = 64 Then
    TOOTH = 95
ElseIf pc >= 65 And pc <= 90 Then
    TOOTH = pc
ElseIf pc = 91 Then
    TOOTH = 196

```



```

ElseIf pc = 92 Then
    TOOTH = 214
ElseIf pc = 94 Then
    TOOTH = 220
ElseIf pc = 96 Then
    TOOTH = 168
ElseIf pc >= 97 And pc <= 122 Then
    TOOTH = pc
ElseIf pc = 123 Then
    TOOTH = 228
ElseIf pc = 124 Then
    TOOTH = 246
ElseIf pc = 125 Then
    TOOTH = 241
ElseIf pc = 126 Then
    TOOTH = 252
ElseIf pc = 127 Then
    TOOTH = 224
End If
DETOOTH = Chr(TOOTH)
End Function

```

'function ini untuk mengubah string isi dalam teks
'biasa menjadi string isi dalam pasangan heksa

```

Function UbahIsi(ByVal strIsi As String) As String
Dim pj As String
Dim potongan() As String
Dim empatan() As String
Dim hasil() As String
Dim tamp As String
pj = Len(strIsi)
ReDim potongan(pj + 1)
ReDim empatan(pj + 1, 3)
ReDim hasil(pj + 1, 3)
Dim i
tamp = String$(pj, "0")
For i = 1 To pj
    tamp = tamp & BINER7(TOOTH(Mid(strIsi, pj - i + 1, 1)))
Next i
For i = pj To 1 Step -1
    potongan(i) = Mid(tamp, (8 * (i - 1)) + 1, 8)
    empatan(i, 1) = Mid(potongan(i), 1, 4)
    empatan(i, 2) = Mid(potongan(i), 5, 4)
    hasil(i, 1) = Hex$(DEBINER4(empatan(i, 1)))
    hasil(i, 2) = Hex$(DEBINER4(empatan(i, 2)))
    UbahIsi = UbahIsi & hasil(i, 1) & hasil(i, 2)
Next i
End Function

```

'function ini untuk mengubah string nomor ponsel yang dituju
'dalam teks biasa menjadi string dalam pasangan heksa

```

Function UbahNo(ByVal strNomor As String) As String
Dim pj As String
Dim tamp() As String
Dim hasil() As String

pj = Len(strNomor)
If pj Mod 2 = 1 Then strNomor = strNomor & "F"
pj = Len(strNomor)

```

```

ReDim tamp(pj / 2 + 1)
ReDim hasil(pj / 2 + 1, 3)
Dim i
For i = pj To pj / 2
    tamp(i) = Mid(strNomer, (i - 1) * 2 + 1, 2)
    hasil(i, 1) = Right(tamp(i), 1)
    hasil(i, 2) = Left(tamp(i), 1)
    UbahNo = UbahNo & hasil(i, 1) & hasil(i, 2)
Next i

End Function

'function ini untuk mengubah string isi dalam pasangan
'heksa menjadi string isi dalam teks biasa
Function balikinIsi(ByVal isi As String) As String
Dim b, c, d, e
Dim f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, X, Y, z
Dim pj As Integer
Dim splits() As String
Dim tmp As String, tmp1 As String
Dim bnr() As String
Dim d7() As String
Dim i7() As Integer
Dim str1, str2
Dim str3, str4
Dim str5, str6, str7, str8
pj = Len(isi)
tmp1 = ""
tmp = ""
balikinIsi = ""
str1 = ""
str2 = ""
ReDim splits(pj / 2)
ReDim bnr(pj)
For i = 0 To pj / 2 - 1
    splits(i) = balik(Mid(isi, 2 * i + 1, 2))
    tmp = tmp + splits(i)
Next i
tmp = StrReverse(tmp)
For i = 0 To pj - 1
    bnr(i) = BINER4(antiHex(Mid(tmp, i + 1, 1))) * 467724
    tmp1 = tmp1 & bnr(i)
Next i
b = Len(tmp1)
str1 = tmp1
tmp1 = Right(tmp1, pj / 2 * 7)
c = Len(tmp1)
str2 = Mid(str1, 1, b - c)
d = Len(str2)
e = d Mod 7
str5 = Mid(str2, e + 1)
f = Len(str5)
str2 = str5
ReDim d7(Len(str2) / 7)
ReDim i7(Len(str2) / 7)
For i = 0 To Len(str2) / 7 - 1
    d7(i) = Mid(str2, 7 * i + 1, 7)
    If d7(i) = "0000000" And i = 0 Then
        Else

```

```

        i7(i) = DEBINER7(d7(i))
        d7(i) = DETOOTH(i7(i))
        balikinIsi = d7(i) & balikinIsi
    End If
Next i
str3 = balikinIsi
balikinIsi = ""
ReDim d7(Len(tmp1) / 7)
ReDim i7(Len(tmp1) / 7)
For i = 0 To Len(tmp1) / 7 - 1
    d7(i) = Mid(tmp1, 7 * i + 1, 7)
    i7(i) = DEBINER7(d7(i))
    d7(i) = DETOOTH(i7(i))
    balikinIsi = d7(i) & balikinIsi
Next i
str4 = balikinIsi & str3
balikinIsi = str4
End Function

'function ini untuk mengubah string nomor ponsel dalam
'pasangan heksa menjadi string dalam teks biasa
Function balikinNo(ByVal strNo As String) As String
    Dim pj As Integer
    Dim splits() As String
    pj = Len(strNo)
    ReDim splits(pj / 2 - 1)
    Dim i
    If Left(strNo, 1) = "9" Then
        For i = 0 To pj / 2 - 2
            splits(i) = balik(Mid(strNo, 2 * i + 3, 2))
            balikinNo = balikinNo & splits(i)
            balikinNo = Replace(balikinNo, "F", "")
        Next i
    Else
        balikinNo = 62
        For i = 0 To pj / 2 - 2
            If i = 0 Then
                splits(i) = Left((Mid(strNo, 2 * i + 3, 2)), 1)
            Else
                splits(i) = balik(Mid(strNo, 2 * i + 3, 2))
            End If
            balikinNo = balikinNo & splits(i)
            balikinNo = Replace(balikinNo, "F", "")
        Next i
    End If
End Function

'function ini untuk mengubah string tanggal dalam pasangan
'heksa menjadi string dalam teks biasa
Function balikinDt(ByVal strDt As String) As String
    Dim splits(6) As String
    Dim i
    For i = 0 To 5
        splits(i) = balik(Mid(strDt, 2 * i + 1, 2))
        balikinDt = balikinDt & splits(i)
    Next i
End Function

'function ini untuk mengubah string menjadi tanggal
Function str2dt(ByVal strDt As String) As Date

```

```

Dim spl(6) As String
spl(0) = Mid(strDt, 3, 2)
spl(1) = Mid(strDt, 5, 2)
spl(2) = Mid(strDt, 7, 2)
spl(3) = Mid(strDt, 9, 2)
spl(4) = Mid(strDt, 11, 2)
spl(5) = Mid(strDt, 13, 2)
str2dt = CDate(spl(0) & "/" & spl(1) & "/" & spl(2) & " " & _
    spl(3) & ":" & spl(4) & ":" & spl(5))
End Function

'function ini untuk membalikkan posisi kedua huruf dalam
'sebuah pasangan heksa
Function balik(ByVal strDuo As String) As String
    balik = Right(strDuo, 1) & Left(strDuo, 1)
End Function

'function ini untuk mengubah long integer menjadi string hexa
Function lng2hex(ByVal lngBil As Long) As String
    lng2hex = Hex$(lngBil)
    If Len(lng2hex) = 1 Then
        lng2hex = "0" & lng2hex
    End If
End Function

'function ini untuk mengubah string hexa menjadi long integer
Function hex2lng(ByVal strHex As String) As Long
    Dim strH(2) As String
    Dim intL(2) As Integer
    strH(0) = Left(strHex, 1)
    strH(1) = Right(strHex, 1)
    Dim i
    For i = 0 To 1
        If Asc(strH(i)) >= 48 And Asc(strH(i)) <= 57 Then
            intL(i) = Asc(strH(i)) - 48
        ElseIf Asc(strH(i)) >= 65 And Asc(strH(i)) <= 70 Then
            intL(i) = Asc(strH(i)) - 55
        End If
    Next i
    hex2lng = 16 * intL(0) + intL(1)
End Function

'function ini untuk mengubah string bilangan heksa desimal
'menjadi integer
Function antiHex(ByVal strHex As String) As Integer
    If Asc(strHex) >= 48 And Asc(strHex) <= 57 Then
        antiHex = Asc(strHex) - 48
    ElseIf Asc(strHex) >= 65 And Asc(strHex) <= 70 Then
        antiHex = Asc(strHex) - 55
    End If
End Function

Function Hex2Bin(Heksa) As String
    Dim a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u,
    v, w, X, Y, z
    Dim s0, s1, s2, s3, s4, s5, s6, s7, s8, s9
    Dim s10, s12, s13, s14, s15, s16, s17, s18, s19

```

```

Dim s20, s22, s23, s24, s25, s26, s27, s28, s29
Dim s30, s32, s33, s34, s35, s36, s37, s38, s39
Dim s40, s42, s43, s44, s45, s46, s47, s48, s49

```

```

a = Heksa
s0 = Mid(a, 1, 1)
s1 = HB(s0)
s0 = Mid(a, 2, 1)
s2 = HB(s0)
Hex2Bin = s1 & s2
End Function

```

```

Function HB(Heksa) As String
Select Case Heksa
Case "0"
    HB = "0000"
Case "1"
    HB = "0001"
Case "2"
    HB = "0010"
Case "3"
    HB = "0011"
Case "4"
    HB = "0100"
Case "5"
    HB = "0101"
Case "6"
    HB = "0110"
Case "7"
    HB = "0111"
Case "8"
    HB = "1000"
Case "9"
    HB = "1001"
Case "A"
    HB = "1010"
Case "B"
    HB = "1011"
Case "C"
    HB = "1100"
Case "D"
    HB = "1101"
Case "E"
    HB = "1110"
Case "F"
    HB = "1111"
End Select
End Function

```

```

Function ExtractSMS(Data)
Dim a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u,
v, w, X, Y, z
Dim s0, s1, s2, s3, s4, s5, s6, s7, s8, s9
Dim s10, s12, s13, s14, s15, s16, s17, s18, s19
Dim s20, s22, s23, s24, s25, s26, s27, s28, s29
Dim s21
Dim s2xx
Dim s21xx, s21xxx, s21xxxx

```

```

Dim s30, s32, s33, s34, s35, s36, s37, s38, s39
Dim s40, s42, s43, s44, s45, s46, s47, s48, s49
Dim Encoding
Dim Indeks

```

```

Dim xx As MSComctlLib.ListItem
Dim yy As MSComctlLib.ListItem
Dim EA As Variant
Dim str1 As String
Dim str2 As String
Dim str3 As String
Dim str4 As String
Dim str5 As String

```

```

On Error Resume Next

```

```

'GoTo YX
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""

```

```

str1 = Data
s0 = CDec(Mid(str1, 1, 2))
s1 = Mid(str1, 3 + (s0 * 2) + 2)
'----- ambil nomor HP pengirim -----
a = CDec("&H" & Mid(s1, 1, 2))
s2 = Mid(s1, 5, a)
s2xx = Mid(s2, Len(s2))
If s2xx = "F" Then
    s2l = Mid(s1, 5, a + 1)
    s2lxxx = Mid(s2l, Len(s2l))
    s2 = Mid(s1, 5, a - 1) & s2lxxx
    a = a + 1
    s3 = Swap(s2) & s2lxxx
Else
    s3 = Swap(s2)
End If

s4 = "0" + Mid(s3, 3, Len(s3) - 2)
Text3.Text = s4
RTRecord.Text = RTRecord.Text & "No Urut: " & (MF1.Rows - 1) &
vbCrLf
RTRecord.Text = RTRecord.Text & "No HP pengirim: " & s4 & ", "
'----- ambil tanggal dan jam sms -----
Encoding = Mid(s1, a + 7, 2)
s5 = Mid(s1, a + 9)
s6 = Mid(s5, 1, 12)
s7 = Swap(s6)
'--- tanggal ---
m = Mid(s7, 5, 2)
n = Mid(s7, 3, 2)
o = "20" & Mid(s7, 1, 2)
p = m & "-" & n & "-" & o
Text5.Text = p
RTRecord.Text = RTRecord.Text & "Tanggal, jam kirim: " & p & ", "
'--- jam ---
m = Mid(s7, 7, 2)
n = Mid(s7, 9, 2)

```

```

o = Mid(s7, 11, 2)
p = m & ":" & n & ":" & o
Text6.Text = p
RTRecord.Text = RTRecord.Text & p & vbCrLf
'----- ambil skema encoding pesan -----
' perhatikan, jika = 00 maka gunakan seperti yang ada dibuku
' jika = 08 maka...
' if FF = status Report
If Encoding = "08" Then
    s8 = CDec("&H" & Mid(s1, a + 9 + 12 + 2, 2))
    s9 = s8 \ 2
    s10 = Mid(s1, a + 9 + 12 + 4)
    z = 0
    Y = ""
    For i = 1 To s9
        m = Mid(s10, (i * 3) + (i - 1), 2)
        Y = Y + Chr(CDec("&h" & m))
    Next i
    Text4.Text = Y
ElseIf Encoding = "00" Then
    s8 = CDec("&H" & Mid(s1, a + 9 + 12 + 2, 2))
    str1 = UCase(Mid(s1, a + 9 + 12 + 2 + 2))
    str2 = balikinIsi(str1)
    a = Len(str2)
    Text4.Text = str2
    'Text7.Text = str1
    'Text8.Text = str2
ElseIf Encoding = "FF" Then
    ' Format: Status, Nomor HP, Tgl Kirim, Jam Kirim, Tgl Terima,
    Jam Terima
    '4D = Not Delivered
    s8 = CDec("&H" & Mid(s1, a + 9 + 12 + 2, 2))
    txtHPFenerima.Text = s1
    str1 = UCase(Mid(s1, a + 9 + 12 + 2 + 2))
    str2 = balikinIsi(str1)
End If
RTRecord.Text = RTRecord.Text & "Isi Pesan: " & vbCrLf &
Text4.Text & vbCrLf
RTRecord.Text = RTRecord.Text & "-----
-----" & vbCrLf & vbCrLf
RTRecord.SelStart = Len(RTRecord.Text)
a = a

YX:
a = MF1.Rows
MF1.Row = MF1.Rows - 1
MF1.Col = 0
MF1.Text = CStr(MF1.Rows - 1)
MF1.Col = 1
MF1.Text = Text3.Text
MF1.Col = 2
MF1.Text = Text4.Text

If Text3.Text <> "" Or Text4.Text <> "" Then
    MF1.Col = 3
    MF1.Text = "Antri"
    MF1.CellAlignment = 4
End If

```

```

MF1.Col = 3
MF1.Text = "Batal"
MF1.CellAlignment = 4
End If

End Function

Private Static Sub ShowData(Term As Control, Data As String)
    On Error GoTo Handler
    Const MAXTERMSIZE = 16000
    Dim TermSize As Long, i

    ' Make sure the existing text doesn't get too large.
    TermSize = Len(Term.Text)
    If TermSize > MAXTERMSIZE Then
        Term.Text = Mid$(Term.Text, 4097)
        TermSize = Len(Term.Text)
    End If

    ' Point to the end of Term's data.
    Term.SelStart = TermSize

    ' Filter/handle BACKSPACE characters.
    Do
        i = InStr(Data, Chr$(8))
        If i Then
            If i = 1 Then
                Term.SelStart = TermSize - 1
                Term.SelLength = 1
                Data = Mid$(Data, i + 1)
            Else
                Data = Left$(Data, i - 2) & Mid$(Data, i + 1)
            End If
        End If
    Loop While i

    ' Eliminate line feeds.
    Do
        i = InStr(Data, Chr$(10))
        If i Then
            Data = Left$(Data, i - 1) & Mid$(Data, i + 1)
        End If
    Loop While i

    ' Make sure all carriage returns have a line feed.
    i = 1
    Do
        i = InStr(i, Data, Chr$(13))
        If i Then
            Data = Left$(Data, i) & Chr$(10) & Mid$(Data, i + 1)
            i = i + 1
        End If
    Loop While i

    ' Add the filtered data to the SelText property.
    Term.SelText = Data

    ' Log data to file if requested.
    If hLogFile Then

```

```

i = 2
Do
    Err = 0
    Put hLogFile, , Data
    If Err Then
        i = MsgBox(Error$, 21)
        If i = 2 Then
            'mruCloseLog_Click
        End If
    End If
    Loop While i <> 2
End If
Term.SelStart = Len(Term.Text)
Exit Sub

Handler:
    MsgBox Error$
    Resume Next
End Sub

Private Sub Timer2_Timer()

End Sub

Private Sub tmrGetMC_Timer()
    If MSComm2.PortOpen = True Then

        MSComm2.Output = "@"

    Else
        MsgBox "Port ke MC belum terhubung"
        Exit Sub
    End If
End Sub

Private Sub txtTerm1_Change()
    Dim a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u,
    v, w, X, Y, z
    Dim s0, s1, s2, s3, s4, s5, s6, s7, s8, s9
    Dim Kecepatan
    Dim Arah

    If txtTerm1.Text = "" Then Exit Sub
    a = InStr(1, txtTerm1.Text, "$")
    If a > 5 Then
        b = Mid(txtTerm1.Text, a - 5, 5)
        s0 = Val("&H" & Mid(b, 1, 2))
        k = (2 * (22 / 7) * 6) / 100
        j = (k * s0 / 10) * 3.6
        Kecepatan = Round(j, 2)
        txtKecepatan.Text = Kecepatan
        w = Val("&H" & Mid(b, 4, 2)) And 31
        Arah = w * 11.25
        txtArah.Text = Arah & Chr(176)
        txtTerm1.Text = ""
        CMSave.InsertLine CMSave.LineCount, Time & " " & Date & "
-> arah = " & txtArah.Text & Chr(9) & ", v = " & txtKecepatan.Text
& " km/jam"

```

```

        End If
    End Sub
    Private Sub XPButton1_Click()
        Unload Me
    End Sub

    Private Sub XPButton2_Click()

        MSComm1.CommPort = cmbPortHP.Text
        MSComm1.PortOpen = True
        Timer1.Enabled = True
        XPButton3.Enabled = True
        XPButton2.Enabled = False

    End Sub

    Private Sub XPButton3_Click()
        MSComm1.PortOpen = False
        Timer1.Enabled = False
        XPButton2.Enabled = True
        XPButton3.Enabled = False
    End Sub

    Private Sub ShowLV(RecSet As Recordset)
        Dim EA As Variant
        Dim a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u,
        v, w, X, Y, z
        Dim s0, s1, s2, s3, s4, s5, s6, s7, s8, s9
        Dim xx As MSComctlLib.ListItem
        Dim s10, s12, s13, s14, s15, s16, s17, s18, s19
        Dim s20, s22, s23, s24, s25, s26, s27, s28, s29
        Dim s30, s32, s33, s34, s35, s36, s37, s38, s39
        Dim s40, s42, s43, s44, s45, s46, s47, s48, s49

        Dim campo As Field
        Dim Temp As String
        Dim Data(8)

        On Error Resume Next
        LV2.ListItems.Clear
        s30 = rcd.RecordCount
        s31 = rcd.Fields.Count
        i = 1

        For i = 1 To s30
            c = 1
            For Each campo In rcd.Fields
                a = campo.Value
                Data(c) = a
                If c = 1 Then
                    Set xx = LV2.FindItem(a)
                End If
                c = c + 1
            Next campo
            If xx Is Nothing Then
                Set xx = LV2.ListItems.Add()
                xx.Text = Data(1)
                xx.Key = Data(1) & Data(2)
                xx.ListSubItems.Add , xx.Key & "1", Data(2)
            End If
        Next i
    End Sub

```

```

        xx.ListSubItems.Add , xx.Key + "2", Data(5)
        xx.ListSubItems.Add , xx.Key + "3", Data(6)
        xx.ListSubItems.Add , xx.Key + "4", Data(7)
    End If
    rcd.MoveNext
Next i
End Sub

Private Sub XFPButton4_Click()
    If DataNIM <> "" Then
        SSql = "Delete * from datasemua where [nim]='" & DataNIM &
        "';"
        rcd.Open SSql, cnn, adOpenStatic, adCmdTable
        SSql = "Select * from DataSemua ;"
        rcd.Open SSql, cnn, adOpenStatic, adCmdTable
        Call ShowLV(rcd)
        rcd.Close
    End If
End Sub

Private Sub KirimSMS()
    Dim a
    txtHPPenerima.Text = Text3.Text
    a = Mid(txtHPPenerima.Text, 2, Len(txtHPPenerima.Text) - 1)
    txtHPPenerima.Text = "62" + a
End Sub

Command2_Click

Command3_Click
End Sub

```

Manual Reference

AT Command Set

(GSM 07.07, GSM 07.05,
Siemens specific commands)

for the SIEMENS Mobile Phones

S35i, C35i, M35i

The command descriptions or example sequences in this document imply no liability or warranty in any way. The author therefore will take no responsibility and will accept no liability which results of using the content of this document in any way.

All rights reserved. No part of this work covered by the copyrights hereof may be reproduced or copied in any form or by any means (graphic, electronic, or mechanical, including photocopying, taping, or information storage and retrieval systems) without written permission of the publisher.

Revisions Overview

Date	Version	Name	Description of revision
15-03-2000	1.0	Kel	created

Table of Contents

Revisions Overview.....2

Table of Contents.....2

1. Software Interface.....3

1.1. OVERVIEW OF THE SUPPORTED AT COMMAND SET.....3

1.2. AT COMMAND SET.....5

1.2.1. *Hayes-Standard Commands*.....6

1.2.2. *Acknowledgments for Normal Data Communication*.....6

1.3. AT COMMANDS AND RESPONSES ACCORDING TO GSM 07.07 and GSM 07.05.....7

1.3.1. *AT Cellular Commands According to GSM 07.07*.....7

1.3.2. *AT Commands According to GSM 07.05 for SMS*.....26

1.3.3. *User-Defined Commands for Controlling the GSM Mobile Phone*.....33

1.3.4. *Summary of All Unexpected Messages*.....44

APPENDIX A.....45

Factory settings made by AT&F.....45

Features of the Telephone-Book Memory.....45

Writing to the FDN Phonebook / FDN Replacement.....47

Special hints for using +CPBR/+CPBW command.....48

APPENDIX B.....50

Example for creating/interrogation of an organizer entry.....50

1. Software Interface

1.1. Overview of the Supported AT Command Set

Page	Commands 07.07	Function
7	AT+CGMI	Issue manufacturer ID code
7	AT+CGMM	Issue model ID code
7	AT+CGMR	Output the GSM telephone version
8	AT+CGSN	Output the serial number (IMEI)
8	AT+CSN	Output the serial number (IMEI)
8	AT+CHUP	Terminate call
8	AT+CEER	Query the reason for disconnection of last call
9	AT+CREG	Network registration
9	AT+COPS	Commands concerning selection of network operator
10	AT+CLCK	Switch locks on and off
10	AT+CPWD	Change password to a lock
11	AT+CLIP	Display telephone number of calling party
11	AT+CCFC	Call forwarding
12	AT+CHLD	Call hold and multiparty
12	AT+CPAS	Query the telephone status
13	AT+CPIN	Enter PIN and query lock
13	AT+CBC	Battery charge
14	AT+CSQ	Output signal quality
14	AT+CPBS	Select a telephone book
15	Fehler! Kein gültiges Resultat für Tabelle.	Read a telephone-book entry
15	AT+CPBW	Write a telephone-book entry
16	AT+CMEE	Expanded error messages according to GSM 07.07
17	AT+VTS	Send a DTMF tone
18	AT+VTD	Set duration of a DTMF tone
18	AT+WS46	Select wireless network
18	AT+CSCS	Select TE character set
19	AT+CAOC	Advice of charge
19	AT+CSSN	Supplementary service notifications
20	AT+CRSM	Restricted SIM access
20	AT+CIMI	Output of IMSI
21	AT+CACM	Accumulated call meter
21	AT+CAMM	Accumulated call meter maximum
22	AT+CLCC	List Current Calls
23	AT+CCLK	Clock
23	AT+COPN	Read operator names
23	AT+CPUC	Price per unit and currency table
24	AT+CALM	Alert sound mode
24	AT+CRSL	Ringer sound level
24	AT+CLVL	Loudspeaker volume level
24	AT+CMUT	Mute control
25	AT+CVIB	Vibrator mode

Page	Commands 07.05	Function
26	AT+CSMS	Selection of message service
27	AT+CPMS	Selection of SMS memory
27	AT+CMGF	SMS format
28	AT+CSCA	Address of the SMS service center
28	AT+CNMI	Display new incoming SMS
29	AT+CNMA	Acknowledgment of a short message directly output
30	AT+CMGL	List SMS
31	AT+CMGR	Read in an SMS
31	AT+CMGS	Send an SMS
31	AT+CMSS	Send an SMS from the SMS memory
32	AT+CMGW	Write an SMS to the SMS memory
32	AT+CMGD	Delete an SMS in the SMS memory
32	AT+CSCB	Select cell broadcast messages
32	AT+CMGC	Send an SMS command

Page	Siemens- specific commands	Function
33	AT+SPBS	Select a telephone book (including Siemens-specific books)
33	AT+SDLD	Delete the "last number redial" memory
34	AT+SPBC	Seek the first entry in the sorted telephone book which begins with the selected (or next available) letter
34	AT+SPBG	Read entry from the sorted telephone book via the sorted index
35	AT+SLOCK	Switch locks (including user-defined locks) on and off
35	AT+SPWD	Change password to a lock (including user-defined locks)
36	AT+SACM	Output ACM (accumulated call meter) and ACMmax
36	AT+SPLM	Read the PLMN
36	AT+SPLR	Read an entry from the preferred-operator list
36	AT+SPLW	Write an entry to the preferred-operator list
37	AT+SCNI	Output call number information
37	AT+SNFV	Set the volume
37	AT+SNFS	Select NF hardware
38	AT+SRTC	Set the ringing tone
38	AT+SCID	Output card ID
38	AT+SCKS	Output SIM card status
39	AT+SPIC	Output PIN counter
39	AT+SMGO	SMS overflow indicator
40	AT+SMGL	List SMS (without status change from unread to read)
40	AT+SMGR	Read SMS record without Changing unread->read
40	AT+SMSC	Switch device off
41	AT+SLNG	Language settings
41	AT+SSTK	SIM Toolkit
41	AT+SBNW	Binary Write
43	AT+SBNR	Binary Read

1.2. AT Command Set

Remote control operation of the GSM mobile telephone runs via a serial interface (data cable or infrared connection), where AT+C commands according to ETSI GSM 07.07 and GSM 07.05 specification as well as several manufacturer specific AT commands are available. These commands are described in more detail later on.

The modem guideline V.25ter applies to the sequence of the interface commands. According to this guideline, commands should begin with the character string "AT" and end with "<CR>" (= 0x0D). The input of a command is acknowledged by the display of "OK" or "ERROR". **A command currently in process is interrupted by each additional character entered.** This means that you should not enter the next command until you have received the acknowledgment; otherwise the current command is interrupted.

The commands supported are listed in the following tables:

1.2.1. Hayes-Standard Commands

The Hayes-standard commands correspond to the commands of AT Hayes-compatible modems.

Command	Function
A/	Repeat last command
AT...	Prefix for all other commands
ATA	Accept call
ATD<str>;	Dial the dialing string <str> with the voice utility Valid dial modifiers: "T" (tone dialing), "P" (pulse dialing) is ignored. The character ";" is important, for this tells the phone that the call should be set up with the voice utility. Otherwise an attempt is made to set up a data call, which the phone immediately acknowledges with "ERROR". The dial command responds with OK to the user right after starting a voice call. Other behavior like *# sequences in the dial command and also data calls remain unchanged.
ATD><n>;	Dial the telephone number from the current telephone book location number <n> The telephone book is selected with the command at+cpbs (or at^spbs).
ATD><mem><n>;	Dial the telephone number from the telephone book <mem> location number <n>
ATDL	Dial last telephone number
ATE0	Deactivate command echo
ATE1	Activate command echo
ATH[0]	Separate connection
ATQ0	Display acknowledgments
ATQ1	Suppress acknowledgments
ATV0	Output acknowledgments as numbers
ATV1	Output acknowledgments as text
AT&F[0]	Reset to factory profile
ATZ	Set to default configuration
AT+GCAP	Output the capabilities list

1.2.2. Acknowledgments for Normal Data Communication

Response	Numeric	Meaning
OK	0	Command executed, no errors
RING	2	Ring detected
NO CARRIER	3	Link not established or disconnected
ERROR	4	Invalid command or command line too long
NO DIALTONE	6	No dial tone, dialing impossible, wrong mode
BUSY	7	Remote station busy

1.3. AT Commands and Responses According to GSM 07.07 and GSM 07.05

According to GSM, it is possible to execute an AT command in various forms.

Test command	AT+CXXX=?	The telephone responds by sending the list of parameters and value ranges; these can be set using the affiliated Write command or by means of internal processes.
Read command	AT+CXXX?	This command tells you the current value setting of the parameter(s).
Write command	AT+CXXX=<...>	This command is used to set parameters that can be set.
Execute command	AT+CXXX	The Execute command reads non-settable parameters which are influenced by internal processes in the telephone.

1.3.1. AT Cellular Commands According to GSM 07.07

AT+CGMI	Issue manufacturer ID code
Test command AT+CGMI=?	Response OK
Execute command AT+CGMI	Response <manufacturer> Parameter <manufacturer> Name of manufacturer (SIEMENS) Important: There is a leading output prefix +CGMI in models before the S25.

AT+CGMM	Issue model ID code
Test command AT+CGMM=?	Response OK
Execute command AT+CGMM	Response <model> Parameter <model> Name of telephone (MOBILE) Important: There is a leading output prefix +CGMM in models before the S25.

AT+CGMR	Output the GSM telephone version
Test command AT+CGMR=?	Response OK
Execute command AT+CGMR	Response <revision> Parameter <revision> Version of the telephone software Important: There is a leading output prefix +CGMR in models before the S25.

AT+CGSN	Output the serial number (IMEI)
Test command AT+CGSN=?	Response OK
Execute command AT+CGSN	Response <sn> Parameter <sn> IMEI of the telephone Important: There is a leading output prefix +CGMI in models before the S25.

AT+GSN	Output the serial number (IMEI)
Test command AT+GSN=?	Response OK
Execute command AT+GSN	Response +GSN: <sn> Parameter <sn> IMEI of the telephone Important: The output prefix +GSN may be missing in future versions.

AT+CHUP	Terminate call
Test command AT+CHUP=?	Response OK
Execute command AT+CHUP	Response OK/ERROR Description: All active calls and all calls on hold are terminated.

AT+CEER	Query the reason for disconnection of last call
Test command AT+CEER=?	Response OK
Execute command AT+CEER	Response +CEER: <report> Parameter <report> Disconnection reason reported as number

AT+CREG		Network registration
Test command	Response	
AT+CREG=?	+CREG: (list of supported <n>s) OK/ERROR/+CME ERROR	
	Parameter	
	<n>	0 Suppresses the unexpected network-status messages 1 Displays the unexpected network-status messages OK/ERROR/+CME ERROR
Read command	Response	
AT+CREG?	+CREG: <n>, <stat>[, <lac>, <ci>] OK/ERROR/+CME ERROR	
	Parameter	
	<n>	See Test command
	<stat>	0 Not checked in, not seeking 1 Checked in 2 Not checked in, but seeking a network 3 Check-in denied by network 4 Unknown 5 Registered, roaming
	<lac>	Hexadecimal 2-byte string type of location area code
	<ci>	Hexadecimal 2-byte string type of cell ID
Write command	Parameter	
AT+CREG=<n>	<n> See Test command	
	Response	
	OK/ERROR/+CME ERROR	
	Unexpected message	
	+CREG: <stat>	

AT+COPS		Commands concerning selection of network operator
Test command	Response	
AT+COPS=?	+COPS: [list of supported (<stat>, long alphanumeric <oper>, numeric <oper>)s][, (list of supported <mode>s), (list of supported <format>s)] OK/ERROR/+CME ERROR	
	Parameter	
	<stat>	0 Unknown 1 Useful network operator 2 Used network operator 3 Prohibited network operator
	<oper>	Operator in the format according to <mode>
	<mode>	0 Automatic mode 1 Manual selection of network operator 3 Setting of format 4 Automatic, manual selected
	<format>	0 Long alphanumeric 2 Numeric <oper>
Read command	Response	
AT+COPS?	+COPS: <mode>[, <format>, <oper> OK/ERROR/+CME ERROR	
	Parameter	
	<mode>	See Test command
	<format>	See Test command
	<oper>	Network operator
Write command	Parameter	
AT+COPS=<mode> [, <format>[, <oper>]]	<mode> See Test command	
	<format> See Test command	
	If <mode> = 1, <format> can only = 2	
	<oper> In numeric form only	
	Response	

OK/ERROR/+CME ERROR

AT+CLCK	Switch locking on and off Revision to GSM 07.07 according to CR TDOC ETSI/SMG4 187/96
Test command AT+CLCK=?	Response +CLCK: (list of supported <fac>s) OK/ERROR/+CME ERROR Parameter <fac> "CS" Keyboard lock "PS" Phone locked to SIM (device code) "SC" SIM card (PIN) "FD" FDN lock "AO" BAOC (bar all outgoing calls) "OI" BOIC (bar outgoing international calls) "OX" BOIC-exHC (bar outgoing international calls except to home country) "AI" BAIC (bar all incoming calls) "IR" BIC-Roam (bar incoming calls when roaming outside the home country) "AB" All Barring services "AG" All outgoing barring services "AC" All incoming barring services
Write command AT+CLCK=<fac>, <mode>[, <passwd> [, <class>]]	Parameter <fac> See Test command <mode> 0 Cancels lock 1 Activates lock 2 Queries lock status <passwd> Password <class> 1 Voice 2 Data 4 Fax 7 All classes (default value) Response If <mode>=2 and command is successful +CLCK: <status>[, <class1>][<CR><LF> +CLCK: <status>, class2....] Parameter <status> 0 Off 1 On OK/ERROR/+CME ERROR

AT+CPWD	Change password to a lock
Test command AT+CPWD=?	Response +CPWD: list of supported (<fac>, <pwdlength>s) OK/ERROR/+CME ERROR Parameter <fac> "P2" PIN2 otherwise See Test command for AT+CLCK command, without "FD" <pwdlength> Password length
Write command AT+CPWD= <fac>, <oldpwd>, <newpwd>	Parameter <fac> See Test command for AT+CLCK command <oldpwd>, <newpwd> Old and new password Response OK/ERROR/+CME ERROR

AT+CLIP		Display telephone number of calling party	
Test command AT+CLIP=?	Response +CLIP: (list of supported <n>s) OK/ERROR/+CME ERROR	Parameter <n>	0 Suppresses the unexpected messages 1 Displays the unexpected messages
Read command AT+CLIP?	Response +CLIP: <n>, <m> OK/ERROR/+CME ERROR	Parameter <n>	See Test command
		<m>	0 CLIP not booked 1 CLIP booked 2 Unknown
Write command AT+CLIP=[<n>]	Parameter <n>	See Read command	
	Response OK/ERROR/+CME ERROR		
	Unexpected message +CLIP: <num>,<type>	Telephone number of caller	

AT+CCFC		Call forwarding	
Test command AT+CCFC=?	Response +CCFC: (list of supported <reas>s) OK/ERROR/+CME ERROR	Parameter <reas>	0 Always 1 If busy 2 If no answer 3 If not available 4 All reasons (0-3) 5 All conditional reasons (1-3)
Write command AT+CCFC=<reas>, <mode>[, <num> [, <type>[, <class> [, <time>]]]]	Parameter <reas>	See Test command	
	<mode>	0 Deactivate 1 Activate 2 Query 3 Install 4 Delete	
	<num>	Telephone number	
	<type>	Type of telephone number	
	<class>	1 Voice 2 Data 4 Fax 7 All classes	
	<time>	1-30 Time, rounded to a multiple of five seconds	
	Response	If <mode>=2 and command is successful +CCFC: <status>, <class1>[, <num>, <type>[, , <time>]] [<CR><LF>+CCFC:]	
	OK/ERROR/+CME ERROR		
	Parameter <status>	0 Not active 1 Active	

AT+CHLD		Call hold and multiparty																											
Test command AT+CHLD=?		Response +CHLD: (list of supported <n>s) OK/ERROR/+CME ERROR																											
Write command AT+CHLD= [<n>]		Parameter <table><tr><td><n></td><td>0</td><td>Terminates all held calls or sets UDUB (User Determined User Busy) for a waiting call</td></tr><tr><td></td><td>1</td><td>Terminates all active calls (if there are any) and accepts the other call (waiting call or held call)</td></tr><tr><td></td><td>1X</td><td>Terminates call number X (X= 1-7)</td></tr><tr><td></td><td>2</td><td>Puts all active calls on hold (if there are any) and accepts the other call (waiting call or held call) as active</td></tr><tr><td></td><td>2X</td><td>Puts all active calls except call X (X= 1-7) on hold</td></tr><tr><td></td><td>3</td><td>Connects the call put on hold to the active call</td></tr><tr><td colspan="2">For terminating</td><td colspan="2">Terminating all calls except waiting calls is done with 'AT+CHUP'</td></tr><tr><td colspan="2">Note:</td><td colspan="2">Command scope depends on the SIM clearing and/or on the network support</td></tr></table> Response OK/ERROR/+CME ERROR		<n>	0	Terminates all held calls or sets UDUB (User Determined User Busy) for a waiting call		1	Terminates all active calls (if there are any) and accepts the other call (waiting call or held call)		1X	Terminates call number X (X= 1-7)		2	Puts all active calls on hold (if there are any) and accepts the other call (waiting call or held call) as active		2X	Puts all active calls except call X (X= 1-7) on hold		3	Connects the call put on hold to the active call	For terminating		Terminating all calls except waiting calls is done with 'AT+CHUP'		Note:		Command scope depends on the SIM clearing and/or on the network support	
<n>	0	Terminates all held calls or sets UDUB (User Determined User Busy) for a waiting call																											
	1	Terminates all active calls (if there are any) and accepts the other call (waiting call or held call)																											
	1X	Terminates call number X (X= 1-7)																											
	2	Puts all active calls on hold (if there are any) and accepts the other call (waiting call or held call) as active																											
	2X	Puts all active calls except call X (X= 1-7) on hold																											
	3	Connects the call put on hold to the active call																											
For terminating		Terminating all calls except waiting calls is done with 'AT+CHUP'																											
Note:		Command scope depends on the SIM clearing and/or on the network support																											

AT+CPAS		Query the telephone status										
Test command AT+CPAS=?		Response +CPAS: (list of supported <pas>s) OK/ERROR/+CME ERROR										
		Parameter <table><tr><td><pas></td><td>0</td><td>Ready</td></tr><tr><td></td><td>3</td><td>Incoming call (phone is ringing)</td></tr><tr><td></td><td>4</td><td>Call is active</td></tr></table>		<pas>	0	Ready		3	Incoming call (phone is ringing)		4	Call is active
<pas>	0	Ready										
	3	Incoming call (phone is ringing)										
	4	Call is active										
Execute command AT+CPAS		Response +CPAS: <pas> OK/ERROR/+CME ERROR										
		Parameter <pas> See Test command OK/ERROR/+CME ERROR										

AT+CPIN		Enter PIN and query lock
Test command	Response	
AT+CPIN=?	OK	
Read command	Response	
AT+CPIN?	+CPIN: <code> OK/ERROR/+CME ERROR	
	Parameter	
	<code>	
	READY	No further input necessary
	SIM PIN	SIM PIN input necessary
	SIM PUK	SIM PUK input necessary
	PH-SIM PIN	Device-code (theft protection) input necessary
	PH-SIM PUK	Device-code PUK (theft protection) input necessary
	SIM PIN2	PIN2, e.g. for editing the FDN book; only possible if previous command was acknowledged with +CME ERROR:17
	SIM PUK2	Only possible if previous command was acknowledged with error +CME ERROR:18
	The required error message can (must) be provoked by an attempted Write command.	
Write command	Parameter	
AT+CPIN=<pin> [,<new pin>]	<pin>	
	Password for appropriate lock; if the lock is a PUK, then a <new pin> is necessary.	
	<new pin>	
	New password for the lock	
	Response	
	OK/ERROR/+CME ERROR	

AT+CBC		Battery charge
Test command	Response	
AT+CBC=?	+CBC: (list of supported <bcs>s),(list of supported <bcd>s) OK/ERROR/+CME ERROR	
	Parameter	
	<bcs>	
	0	ME is supplied from battery
	1	ME has battery but is not supplied from there
	2	ME has no battery connected
	3	Error
	<bcd>	
	0	Battery is flat, but no more actions possible
	1-100: charge in per cent	
Execute command	Response	
AT+CBC	+CBC: <bcs>,<bcd>	

AT+CSQ	Output signal quality
Test command AT+CSQ=?	Response +CSQ: (list of supported <rss>s), list of supported <ber>) OK/ERROR/+CME ERROR Parameter <rss> Reception level: 0 -113 dBm or less 1 -111 dBm 2-30 -109 to -53 dBm 31 -51 dBm or more 99 Unknown <ber> Bit error rate: 0-7 Like RXQUAL values from Table GSM 05.08 in Section 8.2.4 99 Unknown
Execute command AT+CSQ	Response +CSQ: <rss>, <ber> OK/ERROR/+CME ERROR Parameter <rss> See Test command <ber> See Test command

AT+CPBS	Select a telephone book
Test command AT+CPBS=?	Response +CPBS: (list of supported <sto>s) OK/ERROR/+CME ERROR Parameter <sto> "FD" SIM fix-dialing phonebook "SM" SIM phonebook "ME" ME phonebook "DC" ME Dialed Calls List "ON" SIM (or ME) own numbers (MSISDNs) list "LD" SIM last-dialing phonebook "MC" ME missed (unanswered received) calls list "RC" ME received calls list *For description of telephone-book features, see Appendix A Note: "DC" and "LD" are never both available.
Read command AT+CPBS?	Response +CPBS: <sto> OK/ERROR/+CME ERROR Parameter <sto> See Test command
Write command AT+CPBS=<sto>	Parameter <sto> See Test command Response OK/ERROR/+CME ERROR

AT+CPBR	Read a telephone-book entry
Test command AT+CPBR=?	Response +CPBR: (list of supported <index>s), <nlength>, <tlength> OK/ERROR/+CME ERROR Parameter <index> Location number <nlength> Max. length of telephone number <tlength> Max. length of text corresponding to the number
Write command AT+CPBR= <index1> [,<index2>]	Response +CPBR: <index1>, <nummer>, <typ>, <text>[<CR><LF> +CPBR: +CPBR: <index2>, <nummer>, <typ>, <text>] OK/ERROR/+CME ERROR Parameter <index1> Location number where the read of the entry starts <index2> Location number where the read of the entry ends <nummer> Telephone number <typ> Type of number <text> Text corresponding to the telephone number NOTE: In the <text> field, there may appear special characters like "" (0x22), "@" (0x00), "ø" (0x08), "Ö" (0x5c). (See also +CPBW and Appendix A: Special hints for using +CPBR/+CPBW command) In models before the S25, empty phonebook records are reported as follows: +CPBR: <index1>,empty In S25ff, those empty entries don't produce any output.

AT+CPBW	Write a telephone-book entry																														
Test command AT+CPBW=?	Response +CPBW: (list of supported <index>s), <nlength>,(list of supported <type>s), <tlength> OK/ERROR/+CME ERROR Parameter <index> Location number <nlength> Max. length of telephone number <tlength> Max. length of text corresponding to the number																														
Write command AT+CPBW= [<index>] [,<nummer> [,<typ>[,<text>]]]	Parameter <index> Location number at which the entry is written <nummer> Telephone number <typ> Type of number <text> Text corresponding to the telephone number Response OK/ERROR/+CME ERROR Note: The following characters in <text> must be entered via the escape sequence (see also <i>Appendix A: Special hints for using +CPBR/+CPBW command</i>) <table><tr><th>GSM Char</th><th>Hex char.</th><th>ASCII</th><th>GSM Esc Seq</th><th>Seq.(hex)</th><th>Note</th></tr><tr><td>Ö</td><td>5C</td><td>\</td><td>Ö5C</td><td>5C 35 43</td><td>Backslash</td></tr><tr><td>"</td><td>22</td><td>"</td><td>Ö22</td><td>5C 32 32</td><td>String delim</td></tr><tr><td>ø</td><td>08</td><td>BSP</td><td>Ö08</td><td>5C 30 38</td><td>Backspace</td></tr><tr><td>@</td><td>00</td><td>NULL</td><td>Ö00</td><td>5C 30 30</td><td>GSM Null</td></tr></table> y cause problems on application level when using the function strlen() and should thus be represented by an escape sequence	GSM Char	Hex char.	ASCII	GSM Esc Seq	Seq.(hex)	Note	Ö	5C	\	Ö5C	5C 35 43	Backslash	"	22	"	Ö22	5C 32 32	String delim	ø	08	BSP	Ö08	5C 30 38	Backspace	@	00	NULL	Ö00	5C 30 30	GSM Null
GSM Char	Hex char.	ASCII	GSM Esc Seq	Seq.(hex)	Note																										
Ö	5C	\	Ö5C	5C 35 43	Backslash																										
"	22	"	Ö22	5C 32 32	String delim																										
ø	08	BSP	Ö08	5C 30 38	Backspace																										
@	00	NULL	Ö00	5C 30 30	GSM Null																										

AT+CMEE		Expanded error messages according to GSM 07.07																																																																																	
Test command	AT+CMEE=?	Response	+CMEE: (list of supported <n>s)																																																																																
		Parameter																																																																																	
		<n>	0 Suppresses the expanded error format																																																																																
			1 Expanded error messages as number																																																																																
			2 Expanded error messages as text																																																																																
Read command	AT+CMEE?	Response	+CMEE: <n>																																																																																
		Parameter																																																																																	
		<n>	See Read command																																																																																
Write command	AT+CMEE=<n>	Parameter																																																																																	
		<n>	See Read command																																																																																
		Response	OK/ERROR/+CME ERROR																																																																																
		Description:	The following CME errors are possible: <table><tr><td>0</td><td>PHONE FAILURE</td></tr><tr><td>1</td><td>NO CONNECTION TO PHONE</td></tr><tr><td>2</td><td>PH-TA LINK RESERVED</td></tr><tr><td>3</td><td>OPERATION NOT ALLOWED</td></tr><tr><td>4</td><td>OPERATION NOT SUPPORT</td></tr><tr><td>5</td><td>PH-SIM PIN REQUIRED</td></tr><tr><td>10</td><td>SIM NOT INSERTED</td></tr><tr><td>11</td><td>SIM PIN REQUIRED</td></tr><tr><td>12</td><td>SIM PUK REQUIRED</td></tr><tr><td>13</td><td>SIM FAILURE</td></tr><tr><td>14</td><td>SIM BUSY</td></tr><tr><td>15</td><td>SIM WRONG</td></tr><tr><td>16</td><td>INCORRECT PASSWORD</td></tr><tr><td>17</td><td>SIM PIN2 REQUIRED</td></tr><tr><td>18</td><td>SIM PUK2 REQUIRED</td></tr><tr><td>20</td><td>MEMORY FULL</td></tr><tr><td>21</td><td>INVALID INDEX</td></tr><tr><td>22</td><td>NOT FOUND</td></tr><tr><td>23</td><td>MEMORY FAILURE</td></tr><tr><td>24</td><td>TEXT TOO LONG</td></tr><tr><td>25</td><td>INV CHAR IN TEXT</td></tr><tr><td>26</td><td>DIAL STRING TOO LONG</td></tr><tr><td>27</td><td>INV CHAR IN DIAL</td></tr><tr><td>30</td><td>NO NETWORK SERVICE</td></tr><tr><td>31</td><td>NETWORK TIMEOUT</td></tr><tr><td>100</td><td>UNKNOWN</td></tr><tr><td>512</td><td>CALL BARRED BY BLACKLIST</td></tr><tr><td>513</td><td>PHONE LINK RESERVED</td></tr><tr><td>514</td><td>INVALID DIAL STRING</td></tr><tr><td>515</td><td>PHONE BUSY</td></tr><tr><td>550</td><td>PH-SIM PUK REQUIRED</td></tr><tr><td>551</td><td>NTF-SIM PIN REQUIRED</td></tr><tr><td>552</td><td>NTF-SIM PUK REQUIRED</td></tr><tr><td>553</td><td>PH-NET PIN REQUIRED</td></tr><tr><td>554</td><td>PH-NET PUK REQUIRED</td></tr><tr><td>555</td><td>PH-SP PIN REQUIRED</td></tr><tr><td>556</td><td>PH-SP PUK REQUIRED</td></tr><tr><td>557</td><td>PH-CP PIN REQUIRED</td></tr><tr><td>558</td><td>PH-CP PUK REQUIRED</td></tr><tr><td>559</td><td>FEATURE PIN REQUIRED</td></tr></table>	0	PHONE FAILURE	1	NO CONNECTION TO PHONE	2	PH-TA LINK RESERVED	3	OPERATION NOT ALLOWED	4	OPERATION NOT SUPPORT	5	PH-SIM PIN REQUIRED	10	SIM NOT INSERTED	11	SIM PIN REQUIRED	12	SIM PUK REQUIRED	13	SIM FAILURE	14	SIM BUSY	15	SIM WRONG	16	INCORRECT PASSWORD	17	SIM PIN2 REQUIRED	18	SIM PUK2 REQUIRED	20	MEMORY FULL	21	INVALID INDEX	22	NOT FOUND	23	MEMORY FAILURE	24	TEXT TOO LONG	25	INV CHAR IN TEXT	26	DIAL STRING TOO LONG	27	INV CHAR IN DIAL	30	NO NETWORK SERVICE	31	NETWORK TIMEOUT	100	UNKNOWN	512	CALL BARRED BY BLACKLIST	513	PHONE LINK RESERVED	514	INVALID DIAL STRING	515	PHONE BUSY	550	PH-SIM PUK REQUIRED	551	NTF-SIM PIN REQUIRED	552	NTF-SIM PUK REQUIRED	553	PH-NET PIN REQUIRED	554	PH-NET PUK REQUIRED	555	PH-SP PIN REQUIRED	556	PH-SP PUK REQUIRED	557	PH-CP PIN REQUIRED	558	PH-CP PUK REQUIRED	559	FEATURE PIN REQUIRED
0	PHONE FAILURE																																																																																		
1	NO CONNECTION TO PHONE																																																																																		
2	PH-TA LINK RESERVED																																																																																		
3	OPERATION NOT ALLOWED																																																																																		
4	OPERATION NOT SUPPORT																																																																																		
5	PH-SIM PIN REQUIRED																																																																																		
10	SIM NOT INSERTED																																																																																		
11	SIM PIN REQUIRED																																																																																		
12	SIM PUK REQUIRED																																																																																		
13	SIM FAILURE																																																																																		
14	SIM BUSY																																																																																		
15	SIM WRONG																																																																																		
16	INCORRECT PASSWORD																																																																																		
17	SIM PIN2 REQUIRED																																																																																		
18	SIM PUK2 REQUIRED																																																																																		
20	MEMORY FULL																																																																																		
21	INVALID INDEX																																																																																		
22	NOT FOUND																																																																																		
23	MEMORY FAILURE																																																																																		
24	TEXT TOO LONG																																																																																		
25	INV CHAR IN TEXT																																																																																		
26	DIAL STRING TOO LONG																																																																																		
27	INV CHAR IN DIAL																																																																																		
30	NO NETWORK SERVICE																																																																																		
31	NETWORK TIMEOUT																																																																																		
100	UNKNOWN																																																																																		
512	CALL BARRED BY BLACKLIST																																																																																		
513	PHONE LINK RESERVED																																																																																		
514	INVALID DIAL STRING																																																																																		
515	PHONE BUSY																																																																																		
550	PH-SIM PUK REQUIRED																																																																																		
551	NTF-SIM PIN REQUIRED																																																																																		
552	NTF-SIM PUK REQUIRED																																																																																		
553	PH-NET PIN REQUIRED																																																																																		
554	PH-NET PUK REQUIRED																																																																																		
555	PH-SP PIN REQUIRED																																																																																		
556	PH-SP PUK REQUIRED																																																																																		
557	PH-CP PIN REQUIRED																																																																																		
558	PH-CP PUK REQUIRED																																																																																		
559	FEATURE PIN REQUIRED																																																																																		

	560	FEATURE PUK REQUIRED
The following CMS errors have been defined for SMS:		
	300	ME failure
	301	SMS service of ME reserved
	302	operation not allowed
	303	operation not supported
	304	invalid PDU parameter
	305	invalid TEXT mode
	310	SIM not inserted
	311	SIM PIN necessary
	312	PH-SIM PIN necessary
	313	SIM failure
	314	SIM busy
	315	SIM wrong
	320	memory failure
	321	invalid memory failure
	322	memory full
	330	SMSC address unknown
	331	no network service
	332	network timeout
	340	NO +CNMA ACK EXPECTED
	500	unknown error

AT+VTS	Send a DTMF tone
Test command AT+VTS=?	Response (list of supported <dtmf>s), (list of supported <duration>s) OK/ERROR/+CME ERROR Parameter <dtmf> 0-9,#,*,A-D, exactly one character <duration> Duration of tone in (duration/10) seconds
Write command AT+VTS= <dtmf> [,<duration>] or AT+VTS= <dtmf-string>	Parameter <dtmf> One character from the list, see Test command<duration> See Test command <dtmf-string> max. 29 characters in quotation marks ("..."), then a duration cannot be specified Response OK/ERROR/+CME ERROR Important: There is a leading output prefix +VTS in models before the S25.

AT+VTD	Set duration of a DTMF tone
Test command AT+VTD=?	Response +VTD: (list of supported <duration>s) OK/ERROR/+CME ERROR Parameter <duration> 1-255 Duration of tone in (duration/10) seconds
Read command AT+VTD?	Response +VTD: <duration> OK/ERROR/+CME ERROR
Write command AT+VTD= <duration>	Parameter <duration> See Test command Response OK/ERROR Important: There is a leading output prefix +VTD in models before the S25.

AT+WS46	Select wireless network
Test command AT+WS46=?	Response (list of supported <n>s) OK
Read command AT+WS46?	Response <n> OK/ERROR/+CME ERROR Parameter <n> Integer; WDS side stack 12 GSM digital cellular
Write command AT+WS46=[<n>]	Response OK/ERROR/+CME ERROR Important: There is a leading output prefix +WS46 in models before the S25.

AT+CSCS	Select TE character set
Test command AT+CSCS=?	Response +CSCS: (list of supported <chset>s) OK
Read command AT+CSCS?	Response +CSCS: <chset> OK/ERROR/+CME ERROR Parameter <chset> String; determines which TE character set is used
Write command AT+CSCS= [<chset>]	Response OK/ERROR/+CME ERROR

AT+CAOC	Advice of charge
Test command AT+CAOC=?	Response +CAOC: (list of supported <mode>s) Parameter <mode> 0 query CCM value
Read command AT+CAOC?	Response +CAOC: <mode> Parameter <mode> 0 See Test command
Write command AT+CAOC=<mode>	Response OK Parameter <mode> 0 See Test command
Execute command AT+CAOC	Response +CAOC: <ccm> OK/ERROR/+CME ERROR Parameter <ccm> Updated hexadecimal call meter, measured in home units; coding analogous to ACMmax on the SIM

AT+CSSN	Supplementary service notifications Revision according to GSM 07.07 Version 5.0.0
Test command AT+CSSN=?	Response +CSSN: (list of supported <n>s), (list of supported <m>s) Parameter <n> 0 Suppresses the +CSSI messages 1 Activates the +CSSI messages <m> 0 Suppresses the +CSSU messages 1 Activates the +CSSU messages For supported +CSSI/+CSSU messages, see also 1.3.4. Summary of All Unexpected Messages
Read command AT+CSSN?	Response +CSSN: <n>, <m> Parameter <n> See Test command <m> See Test command
Write command AT+CSSN=<n>[, <m>]	Parameter <n> See Read command <m> See Read command
	Unexpected message +CSSI: <code1> +CSSU: <code2> Parameter <code1> Intermediate result code 3 Waiting call is pending <code2> Unsolicited result code 5 Held call was terminated

AT+CRSM	
Restricted SIM access	
Test command AT+CRSM=?	Response OK
Write command +CRSM=<command> [,<fileid> [,<P1>,<P2>,<P3> [,<data>]]]	Response +CRSM: <sw1>,<sw2>[,<response>] OK/ERROR/+CME ERROR Parameter <command>: 176 READ BINARY 178 READ RECORD 192 GET RESPONSE 214 UPDATE BINARY 220 UPDATE RECORD 242 STATUS <fileid>: Integer, identifier of the data file on the SIM, mandatory for every command except STATUS (see GSM 11.11) <P1>,<P2>,<P3>: Integer, transferal parameter from ME to SIM, mandatory for every command except GET RESPONSE,STATUS (see GSM 11.11) <data>: Hexadecimal string; information that is to be written to the SIM <sw1>,<sw2>: Integer; information from the SIM as to how/whether the command was executed <response>: Hexadecimal string; given when a command was successfully processed Note: The write access to CK boxes receives only limited support and differs from device to device.

AT+CIMI	
Output of IMSI	
Test command AT+CIMI=?	Response OK
Execute command AT+CIMI	Response <imsi> Parameter <imsi> International Mobile Subscriber Identity (IMSI)

AT+CCLK	Clock
Test command AT+CCLK=?	Response OK
Read command AT+SCLK?	Response +CCLK: <time> OK/ERROR/+CME ERROR Parameter: <time>: string type value; format is "yy/MM/dd,hh:mm:ss", where characters indicate year (two last digits), month, day, hour, minutes; E.g. 6th of May 1994, 22:10:00 hours equals to „94/05/06,22:10:00"
Write command AT+CCLK=<time>	Response OK/ERROR/+CME ERROR Parameter: <time> see Test command

AT+COPN	Read operator names
Test command AT+COPN=?	Response OK
Execute command AT+COPN	Response +COPN:numeric <oper>,long alphanumc <oper><CR><LF> +COPN:..... OK/ERROR/+CME ERROR Parameter <oper> Network operator in numeric and alphanumeric notation see AT+SPLM

AT+CPUC	Price per unit and currency table
Test command AT+CPUC=?	Response OK
Read command AT+CPUC?	Response +CPUC: <currency>,<ppu> OK/ERROR/+CME ERROR Parameter <currency> three-character currency code (e.g. "FRA", "DEM") <ppu> price per unit; dot is used as a decimal separator (e.g. "1.33")
Write command AT+CPUC= <currency>,<ppu>[, <passwd>]	Response OK/ERROR/+CME ERROR Parameter <passwd> String type; usually PIN2

AT+Calm	Alert sound mode
Test command AT+Calm=?	Response +Calm: (list of supported <mode>s) OK
Read command AT+Calm?	Response +Calm: <mode> OK/ERROR/+CME ERROR
Write command AT+Calm=<mode>	Response OK/ERROR/+CME ERROR Parameter <mode>: 0 normal mode 1 silent mode (all sounds are prevented) 2 beep (only a short beep indicates an incoming call)

AT+CRSL	Ringer sound level
Test command AT+CRSL=?	Response +CRSL: (list of supported <level>s) OK
Read command AT+CRSL?	Response +CRSL: <level> OK/ERROR/+CME ERROR
Write command AT+CRSL=<level>	Response OK/ERROR/+CME ERROR Parameter <level>: Ringer Sound Level

AT+CLVL	Loudspeaker volume level
Test command AT+CLVL=?	Response +CLVL: (list of supported <level>s) OK
Read command AT+CLVL?	Response +CLVL: <level> OK/ERROR/+CME ERROR
Write command AT+CLVL=<level>	Response OK/ERROR/+CME ERROR Parameter <level>: Loudspeaker Volume Level

AT+CMUT	Mute control
Test command AT+CMUT=?	Response +CMUT: (list of supported <n>s) OK
Read command AT+CMUT?	Response +CMUT: <n> OK/ERROR/+CME ERROR
Write command AT+CMUT=<n>	Response OK/ERROR/+CME ERROR Parameter <n>: 0 mute off 1 mute on

AT+CVIB	Vibrator mode
Test command AT+CVIB=?	Response +CVIB: (list of supported <mode>s) OK
Execute command AT+CVIB	Response +CVIB: <mode> OK/ERROR/+CME ERROR
Write command AT+CVIB=<mode>	Response OK/ERROR/+CME ERROR Parameter <mode>: Vibrator mode 0 disable 1 enable 16 vibrate then ring (not available in every model)

1.3.2. AT Commands According to GSM 07.05 for SMS

The GSM 07.05 commands are used for operating the SMS functions of the GSM mobile phone. The GSM module MOBILE supports the SMS PDU mode.

AT+CSMS	Selection of message service Revision according to GSM 07.05 Version 5.0.0
Test command AT+CSMS=?	Response +CSMS: (list of supported <service>s) Parameter <service> 0 GSM 3.40 and 3.41 1 GSM 3.40 and 3.41 and compatibility of the AT command syntax for phase 2+ NOTE: Deactivating the phase 2+ compatibility is only possible if the direct output of short messages +CNMI=1,2 or +CNMI=1,3 is not activated. If necessary, the latter should be deactivated first.
Read command AT+CSMS?	Response +CSMS: <service>, <mt>, <mo>, <bm> Parameter <service> 0 GSM 3.40 and 3.41 <mt> Mobile terminated messages 1 Type supported <mo> Mobile originated messages 1 Type supported <bm> Broadcast type messages 0 Type not supported
Write command AT+CSMS= <service>	Parameter <service> 0 GSM 3.40 and 3.41 Response +CSMS: <mt>, <mo>, <bm> OK/ERROR/+CMS ERROR

AT+CPMS	
Selection of SMS memory Revision according to GSM 07.05 Version 4.7.0	
Test command AT+CPMS=?	Response +CPMS: (list of supported <mem1>s),(list of supported <mem2>s) (list of supported <mem3>s) Parameter <mem1> Memory from which messages are read and deleted "SM" SIM-messages memory <mem2> Memory to which messages are written and sent "SM" SIM-messages memory <mem3> Memory in which received messages are stored, if forwarding to the PC is not set ("+CNMI") "SM" SIM-messages memory
Read command AT+CPMS?	Response +CPMS: <mem1>,<used1>,<total1>,<mem2>,<used2>,<total2> <mem3>,<used3>,<total3> Parameter <memx> Memory from which messages are read and deleted <usedx> Number of messages currently in <memx> <totalx> Number of storable messages in <memx>
Write command AT+CPMS= <mem1> [,<mem2> [,<mem3>]]	Parameter <mem1> See Test command <mem2> See Test command <mem3> See Test command Response +CPMS: <used1>,<total1>,<used2>,<total3>,<used3>,<total3> OK/ERROR/+CMS ERROR

AT+CMGF	
SMS format	
Test command AT+CMGF=?	Response +CMGF: (list of supported <mode>s) Parameter <mode>: 0 PDU mode
Read command AT+CMGF?	Response +CMGF: <mode> Parameter <mode>: 0 PDU mode
Write command AT+CMGF=[<mode>]	Parameter <mode>: 0 PDU mode Response OK/ERROR

AT+CSCA	Address of the SMS service center
Test command AT+CSCA=?	Response OK
Read command AT+CSCA?	Response +CSCA: <sca>,<tosca> Parameter <sca> Service-center address in string format <tosca> Service-center address format
Write command AT+CSCA= <sca>[, <tosca>]	Parameter <sca> Service-center address in string format <tosca> Service-center address format Response OK/ERROR

AT+CNMI	Display new incoming SMS Revision according to GSM 07.05 Version 4.7.0
Test command AT+CNMI=?	Response +CNMI: (list of supported <mode>s),(list of supported <mt>s),(list of supported <bm>s),(list of supported <ds>s),(list of supported <bfr>s) Parameter <mode> 0 Buffers unexpected messages (but is equivalent to rejecting; see <bfr>) 1 Discard indication and reject new received message unsolicited result codes when TA-TE link is reserved. Otherwise forward them directly to the TE. (only with S25F) 2 Buffers unexpected messages if serial interface is occupied, otherwise they are output (only models before S25) <mt> 0 Suppresses unexpected messages for incoming short messages 1 Unexpected messages of a received short message (SMS-DELIVER) that is stored on a chip card are output in the form +CMTI: <mem>,<index> 2 Unexpected messages of a received short message (SMS-DELIVER) (except class 2 and the message "Waiting Indication Group: store message") are output in the form +CMT: [<alpha>] <length><CR><LF><pdu> (<alpha> is not supported) 3 Class 2 and the message "Waiting Indication Group: store message" are output as <mt>=1 3 Unexpected messages of a received short message (SMS-DELIVER) class 3 are output as <mt>=2. Messages with other data coding schemes are output as <mt>=1. NOTE: <mt>=2 and <mt>=3 are not possible unless the Phase 2+ compatibility has been activated by means of +CSMS=1 <bm> 0 Suppresses unexpected messages for incoming cell broadcast messages 2 Outputs unexpected messages for cell broadcast messages in the form +CBM: <length><CR><LF><pdu> <ds> 0 Suppresses unexpected messages for incoming SMS status reports 2 Outputs unexpected messages for SMS status reports in the form +CDS: <length><CR><LF><pdu>

	<bfr> 1 Buffered unexpected messages are rejected when switching from <mode> 0 to <mode> 2. <mem> See +CPMS <index> Index of the record on the chip card <alpha> alphanumeric representation of the sender address <length> Length of <pdu> <pdu> See +CMGL
Read command AT+CNMI=?	Response +CNMI: <mode>,<mt>,<bm>,<ds>,<bfr> Parameter <mode> See Test command <mt> See Test command <bm> See Test command <ds> See Test command <bfr> See Test command
Write command AT+CNMI= [<mode> [<mt>[,<bm> [,<ds>[,<bfr>]]]]]	Parameter <mode> See Test command <mt> See Test command <bm> See Test command <ds> See Test command <bfr> See Test command Response OK/ERROR/+CMS ERROR
	Unexpected message +CMTI: <mem>,<index> Indication that new message has arrived +CMT: <length><CR><LF><pdu> Direct output of the short message +CDS: <length><CR><LF><pdu> Direct output of the status report +CBM: <length><CR><LF><pdu> Direct output of the cell broadcast message

AT+CNMA	Acknowledgment of a short message directly output (without storing on the chip card) Revision according to GSM 07.05 Version 5.0.0 (NOTE: This command is not possible unless the Phase 2+ compatibility has been activated by means of +CSMS=1)
Test command AT+CNMA=?	Response +CNMA: (list of supported <n>s) Parameter <n> 0 Mode of functioning analogous to GSM 07.05 text mode
Write command AT+CNMA[=<n>]	Parameter <n> See Test command Response OK/ERROR/+CMS ERROR: <err>

AT+CMGL	List SMS Revision according to GSM 07.05 Version 4.7.0
Test command AT+CMGL=?	Response +CMGL: (list of supported <stat>s) Parameter <stat> 0 "REC UNREAD": received unread messages (default) 1 "REC READ": received read messages 2 "STO UNSENT": stored unsent messages 3 "STO SENT": stored sent messages 4 "ALL": all messages
Write command AT+CMGL [=<stat>]	Parameter <stat> See Test command Response If PDU mode (+CMGF=0) and command are successful: +CMGL: <index>, <stat>[, <alpha>], <length> <CR><LF><pdu>[<CR><LF> +CMGL: <index>, <stat>[, <alpha>], <length> <CR><LF><pdu><CR><LF> [...]]
	Parameter <pdu> The PDU begins with the service-center address (according to GSM04.11), followed by the TPDU according to GSM03.40 hexadecimal format otherwise: +CMS ERROR: <err> in

AT+CMGR	Read in an SMS Revision according to GSM 07.05 Version 4.7.0
Test command AT+CMGR=?	Response OK
Write command AT+CMGR= <index>	Parameter <index> Index of message in selected memory <mem1> Response If PDU mode (+CMGF=0) and command are successful: +CMGR: <stat>,[<alpha>],<length><CR><LF><pdu> Parameter <pdu> Siehe "AT+CMGL" otherwise: +CMS ERROR: <err>

AT+CMGS	Send an SMS
Test command AT+CMGS=?	Response OK
Write command If PDU mode (+CMGF=0) +CMGS=<length><CR>PDU is given <ctrl-Z/ESC>	Parameter <length> Length of PDU <pdu> See "AT+CMGL" <mr> Message reference Response If sending is successful: +CMGS: <mr> If sending is not successful: +CMS ERROR: <err>

AT+CMSS	Send an SMS from the SMS memory
Test command AT+CMSS=?	Response OK
Write command +CMSS=<index>[,<da>[,<toda>]]	Parameter <index> Index of message in selected memory <mem1> <da> Destination address in string format <toda> Format of destination address <mr> Message reference Response If sending is successful: +CMSS: <mr> If sending is not successful: +CMS ERROR: <err>

AT+CMGW	Write an SMS to the SMS memory
Test command AT+CMGW=?	Response OK
Write command If PDU mode (+CMGF=0) AT+CMGW=<length>[,<stat>]<CR> PDU is given <ctrl-Z/ESC>	Parameter <length> Length of PDU <stat> See command +CMGL <pdu> See "AT+CMGL" <index> Index of message in selected memory <mem1> Response +CMGW: <index> +CMS ERROR: <err>

AT+CMGD	Delete an SMS in the SMS memory
Test command AT+CMGD=?	Response OK
Write command AT+CMGD=<index>	Parameter <index> Index of message in the selected memory <mem1> Response OK/ERROR/+CMS ERROR

AT+CSCB	Select cell broadcast messages
Test command AT+CSCB=?	Response +CSCB: (list of supported <mode>s) Parameter <mode> 0 Accepts messages that are defined in <mids> and <dcss> 1 Does not accept messages that are defined in <mids> and <dcss>
Read command AT+CSCB?	Response +CSCB: <mode>,<mids>,<dcss> Parameter <mode> See Test command <mids> String type; combinations of CBM message IDs <dcss> String type; combinations of CBM data coding schemes
Write command AT+CSCB=[<mode>[,<mids>[,<dcss>]]]	

AT+CMGC	Send an SMS command
Test command AT+CMGC=?	Response OK
Write command If PDU mode (+CMGF=0) +CMGC=<length><CR> PDU is given <ctrl-Z/ESC>	Parameter <length> Length of PDU <pdu> See "AT+CMGL" <mr> Message reference Response If sending is successful: +CMGC: <mr> If sending is not successful: +CMS ERROR: <err>

1.3.3. User-Defined Commands for Controlling the GSM Mobile Phone

Since user-defined commands cannot be implemented according to official syntax, the character string "+C" is replaced by "^S" ("A" = 0x5E). For the future: if a user-defined command is accepted in the same syntax in GSM recommendations, the command can be addressed using both command strings.

AT^SPBS	
Select a telephone book (including Siemens-specific books)	
Test command AT^SPBS=?	Response ^SPBS: (list of supported <sto>s) OK/ERROR/+CME ERROR Parameter <sto> "FD" SIM fix-dialing telephone book "SM" SIM telephone book "ME" Telephone book in device "DC" ME Dialed Calls List "ON" Own telephone numbers "LD" SIM last dialing number "MC" ME Missed Calls List "RC" ME Received Calls List "MD" Last number redial memory in telephone device "OW" Own numbers "BD" Barred dialing numbers "SD" Service dialing numbers "MS" Missed dialing numbers (unanswered calls) "CD" Callback dialing numbers (answered calls) "BL" Blacklist dialing numbers (barred numbers from remote) "MB" Mailbox dialing numbers (network-operator mailbox) "CS" Common sortable telephone book (sorted combination of "SM", "ME", "FD"; access only via ^SPBC, ^SPBG) "RD" Red book (all entries in "CS" whose name portions have an exclamation point (!) as the final character) *For a description of the telephone-book features, see Appendix A
Read command AT^SPBS?	Response ^SPBS: <sto> OK/ERROR/+CME ERROR Parameter <sto> See Test command
Write command AT^SPBS= <sto>	Parameter <sto> See Test command Response OK/ERROR/+CME ERROR

AT^SDLD	
Delete the "last number redial" memory	
Test command AT^SDLD=?	Response OK
Execute command AT^SDLD	Response OK/ERROR/+CME ERROR

AT^SPBC	Seek the first entry in the sorted telephone book which begins with the selected (or next available) letter
Test command AT^SPBC=?	Response ^SPBC: (list of sorted telephone books supported <mem>s) See AT+CPBS/AT^SPBS OK/ERROR/+CME ERROR
Write command AT^SPBC=<char>	Parameter <char> First letter of sought entry "A" to "Z" (with any other character, the index of the first entry that begins with a special character is sent back) <index> Index in the sorted telephone book (access via AT^SPBG) Response ^SPBC: <index> OK/ERROR/+CME ERROR

AT^SPBG	Read entry from the sorted telephone book via the sorted index
Test command AT^SPBG=?	Response ^SPBG: (list of supported <index>s), <nlength>, <tlength> OK/ERROR/+CME ERROR Parameter <index> Location number <nlength> Max. length of telephone number <tlength> Max. length of the text corresponding to the number
Write command AT^SPBG= <index1> [, <index2>]	Response ^SPBG: <index1>, <nummer>, <typ>, <text>[<CR><CL> ^SPBG: ^SPBG: <index2>, <nummer>, <typ>, <text>] OK/ERROR/+CME ERROR Parameter <index1> Location number where the read of the entry starts <index2> Location number where the read of the entry ends <nummer> Telephone number <typ> Type of number <text> Text corresponding to the telephone number

AT^SLCK	Switch locks (including user-defined locks) on and off
Test command AT^SLCK=?	Response ^SLCK: (list of supported <fac>s) OK/ERROR/+CME ERROR Parameter <fac> "PS" Phone locked to SIM (device code) "SC" SIM card (PIN) "FD" FDN lock "AO" BAOC (bar all outgoing calls) "OI" BOIC (bar outgoing international calls) "OX" BOIC-exHC (bar outgoing international calls except to home country) "AI" BAIC (bar all incoming calls) "IR" BIC-Roam (bar incoming calls when roaming outside the home country) "AB" All barring services "AG" All outgoing barring services "AC" All incoming barring services "PN" Network personalization (GSM 02.22) "PC" Corporate personalization (GSM 02.22) "PU" Network subset personalization (GSM 02.22) "PP" Service provider personalization (GSM 02.22) "PF" Phone locked to very first inserted SIM
Write command AT^SLCK = <fac> , <mode> [, <passwd> [, <class>]]	Parameter <fac> See Test command <mode> 0 Cancels lock 1 Activates lock 2 Queries lock status <passwd> Password <class> 1 Voice 2 Data 4 Fax 7 All classes (default value) Response If <mode>=2 and command is successful ^SLCK: <status>[, <class1>][<CR><LF> ^SLCK: <status>, class2....]] Parameter <status> 0 Off 1 On OK/ERROR/+CME ERROR

AT^SPWD	Change password to a lock (including user-defined locks)
Test command AT^SPWD=?	Response ^SPWD: list of supported (<fac>, <pwdlength>)s OK/ERROR/+CME ERROR Parameter <fac> "P2" PIN2 otherwise See Test command for the command AT^SLCK, without "FD" <pwdlength> Length of password
Write command AT^SPWD = <fac> ,<oldpwd> , <newpwd>	Parameter <fac> See Test command for the command AT^SLCK <oldpwd> , <newpwd> Old and new password Response OK/ERROR/+CME ERROR

AT^SACM	Output ACM (accumulated call meter) and ACMmax
Test command AT^SACM=?	Response ^SACM: (list of supported <n>s)
Execute command AT^SACM	Response ^SACM: <n>, <acm>, <acm_max> OK/ERROR/+CME ERROR Parameter <n> See Test command <acm> Accumulated call meter <acm_max> Maximum accumulated call meter
Write command AT^SACM=<n>	Parameter <n> 0 Suppresses the unexpected message 1 Outputs the unexpected message

AT^SPLM	Read the PLMN list
Test command AT^SPLM=?	Response OK
Execute command AT^SPLM	Response ^SPLM: numeric <oper>, long alphanumeric <oper><CR><LF> ^SPLM: OK/ERROR/+CME ERROR Parameter <oper> Network operator in numeric and alphanumeric notation

AT^SPLR	Read an entry from the preferred-operator list
Test command AT^SPLR=?	Response ^SPLR: (list of supported <index>s) OK/ERROR/+CME ERROR Parameter <index> Location numbers
Write command AT^SPLR=<index1> [, <index2>]	Response ^SPLR: <index1>, numeric <oper> ^SPLR: ^SPLR: <index2>, numeric <oper> OK/ERROR/+CME ERROR Parameter <index1> Location number where the read of the entry starts <index2> Location number where the read of the entry ends <oper> Network operator in numeric form

AT^SPLW	Write an entry to the preferred-operator list
Test command AT^SPLW=?	Response ^SPLW: (list of supported <index>s) OK/ERROR/+CME ERROR Parameter <index> Location number
Write command AT^SPLW=<index>[, <oper>]	Parameter <index> Location number at which the entry is written <oper> Network operator in numeric form Response OK/ERROR/+CME ERROR

AT^SCNI	Output call number information
Test command AT^SCNI=?	Response OK
Execute command AT^SCNI	Response ^SCNI: 1[, <cs>[, <number>, <type>]]<CR><LF> ^SCNI: 2[, <cs>[, <number>, <type>]]<CR><LF> ^SCNI: 3[, <cs>[, <number>, <type>]]<CR><LF> ^SCNI: 4[, <cs>[, <number>, <type>]]<CR><LF> ^SCNI: 5[, <cs>[, <number>, <type>]]<CR><LF> ^SCNI: 6[, <cs>[, <number>, <type>]]<CR><LF> ^SCNI: 7[, <cs>[, <number>, <type>]] OK/ERROR/+CME ERROR Parameter <cs> Call status of affiliated call number (first parameter) 0 Call on hold 1 Active call 2 Waiting call <number> Telephone number <type> Type of number

AT^SNFV	Set the volume
Test command AT^SNFV=?	Response ^SNFV: (list of supported <vol>s) Parameter <vol> Value range of volume (0 to 4) (0 low, ..., 4 max. volume; approx. 3 dB/level)
Read command AT^SNFV?	Response ^SNFV: <vol> Parameter <vol> See Test command
Write command AT^SNFV=<vol>	Parameter <vol> See Test command Response OK/ERROR

AT^SNFS	Select NF hardware
Test command AT^SNFS=?	Response ^SNFS: (list of supported <dev>s) Parameter <dev> 0 Cell phone mode 1 Handsfree
Read command AT^SNFS?	Response ^SNFS: <dev> Parameter <dev> See Test command Note: Volume should be temporarily set to „0“ before NF hardware is changed.
Write command AT^SNFS=<dev>	Parameter <dev> See Test command Response OK/ERROR

AT^SRTC	
Set the ringing tone	
Test command AT^SRTC=?	Response ^SRTC: (list of supported <type>s), (list of supported <vol>s) Parameter <type> 1-X Number of ringing tone 0 Mutes the ringing tone; when MTC is set, the phone does not ring and the volume is ignored <vol> 0-Y Volume of ringing tone
Read command AT^SRTC?	Response ^SRTC: <type>, <vol>, <ringing> Parameter <type> See Test command <vol> See Test command <ringing> 0 Test-ring is switched off 1 Test-ring is switched on
Write command AT^SRTC=[<type>],[<vol>]	Parameter <type> See Test command <vol> See Test command Response OK/ERROR
Execute command AT^SRTC	Response The ringing tone sounds on the current NF device; it is selected using "AT+CNFS" until AT^SRTC is called up again OK/ERROR/+CME ERROR Note: If an MTC arrives while the test-ring is active, the latter is switched off and the "normal" ring is switched on.

AT^SCID	
Output card ID	
Test command AT^SCID=?	Response OK/ERROR/+CME ERROR
Execute command AT^SCID	Response ^SCID: <cid> OK/ERROR/+CME ERROR Parameter <cid> Number of SIM card

AT^SCKS	
Output SIM card status	
Test command AT^SCKS=?	Response ^SCKS: (list of supported <n>s) Parameter <n> 0 Suppresses the unexpected messages 1 Outputs the unexpected messages
Read command AT^SCKS?	Response ^SCKS: <n>, <m> Parameter <m> 0 No card 1 Card in card reader
Write command AT^SCKS=<n>	Parameter <n> See Test command Response OK/ERROR
	Unexpected message ^SCKS: <m> See Read command

AT^SPIC	Output PIN counter
Test command AT^SPIC=?	Response: OK/ERROR/+CME ERROR
Execute command AT^SPIC	Response: ^SPIC: <counter> OK/ERROR/+CME ERROR Parameter: <counter> Number of tries still available to enter the <passwd. The command "AT+CPIN?" must be used to check which password is currently needed.

AT^SMGO	SMS overflow indicator
Test command AT^SMGO=?	Response: ^SMGO: (list of supported <n>s) OK/ERROR/+CME ERROR Parameter: <n> 0 Disable 1 Enable
Read command AT^SMGO?	Response: ^SMGO: <n>, <mode> OK/ERROR/+CME ERROR Parameter: <n> See Test command <mode> 0 Space still available 1 SMS buffer is full (chip card) 2 Buffer is full and new message that should be sent to the telephone is present in the SC
Write command AT^SMGO=<n>	Parameter: <n> See Test command Response: OK/ERROR/+CME ERROR
	Unexpected message: ^SMGO: <mode> See Read command

AT^SMGL	List SMS (without status change from <i>unread</i> to <i>read</i>) Revision according to GSM 07.05 Version 4.7.0
Test command AT^SMGL=?	Response ^SMGL: (list of supported <stat>s) Parameter <stat> 0 "REC UNREAD": received unread messages (default) 1 "REC READ": received read messages 2 "STO UNSENT": stored unsent messages 3 "STO SENT": stored sent messages 4 "ALL": all messages
Write command AT^SMGL [=<stat>]	Parameter <stat> See Test command Response If PDU mode (+CMGF=0) and command is successful: ^SMGL: <index>,<stat>,[<alpha>],<length> <CR><LF><pdu> [<CR><LF>^SMGL: <index>,<stat>,[<alpha>],<length> <CR><LF><pdu> [...]]
	Parameter <pdu> The PDU begins with the service-center address (according to GSM04.11), followed by the TPDU according to GSM03.40 in hexadecimal format otherwise: +CMS ERROR: <err>

AT^SMGR	Read SMS record without Changing unread->read Syntax like AT+CMGR
Test command AT^SMGR=?	Response OK
Write command AT^SMGR= <index>	Parameter <index> Index of message in selected memory <mem1> Response If PDU mode (+CMGF=0) and command are successful: ^SMGR: <stat>,[<alpha>],<length><CR><LF><pdu> Parameter <pdu> Siehe "AT+CMGL" otherwise: +CMS ERROR: <err>

AT^SMSO	Switch device off
Test command AT^SMSO=?	Response OK
Execute command AT^SMSO	Response OK Device switches off

	<subtype> 0 first entry of type "vcs" 1 entry of type "vcs" ... <actNumber> 0 deletes entry of the act. subtype other actual packet number <maxNumber> maximum number of packets
Write command AT^SBNW=<type>,<subtype>, [<actNumber>], <maxNumber>]]<CR>PDU is given<ctrl-Z/ESC>	Response OK/ERROR/+CME ERROR Parameter: <type> see Test command <subtype> see Test command <actNumber> see Test command <maxNumber> see Test command Note: -It is not possible to upload data when a call is active or in progress. -If a call is active the mobile responses with +CME ERROR: PHONE BUSY and the actual upload sequence is aborted and all data packets are discarded. -If uploaded data is not useable (e.g. wrong data format) the mobile responses with +CME ERROR: INV CHAR IN TEXT after the last packet is uploaded. -To get the extended +CME-ERROR-responses AT+CMEE=2 has to be sent before. Otherwise the mobile responses only with ERROR. (see GSM07.07) -If <actNumber> and <maxNumber> during the upload are omitted, the mobile aborts the whole input sequence for the current subtype. -If <actNumber> is 0 during the upload and <maxNumber> is omitted, the mobile deletes the actual record with index <subtype> -Packets have to be uploaded in the right order! -Limitation: The maximum pdu size is 176 bytes (or 352 characters) See Appendix B for examples.

AT^SBNR	Binary Read
Test command AT^SBNR=?	Response ^SBNR: (list of supported <types>s, (list of supported <subtype>s)) OK/ERROR/+CME ERROR Parameter: <type> see AT^SBNW command <subtype> see AT^SBNW command
Write command AT^SBNR=<type>, <subtype>	Response ^SBNR: <type>, <subtype>, 1, <maxNumber> <CR><LF><data><CR><LF> ^SBNR: <type>, <subtype>, 2, <maxNumber> <CR><LF><data><CR><LF>[...] OK/ERROR/+CME ERROR Parameter: <type> see AT^SBNW command <subtype> see AT^SBNW command <data> data in hexadecimal form (PDU) <maxNumber> see AT^SBNW command <i>See Appendix B for examples.</i>

1.3.4. Summary of All Unexpected Messages

Message	Meaning
+CREG: <stat>	Network registration
+CLIP: <num>	Telephone number of caller
+CMTI: <mem>, <index>	Indication of a new short message
+CMT: <length><CR><LF> <pdu>	Short message
+CSSI: <code1> +CSSU: <code2>	Supplementary service intermediate/unsolicited result code
^SMGO: <mode>	SMS overflow indicator
^SCKS: <m>	Message indicating whether card has been removed or inserted
^SACM: <m>	Message indicating if ACM has reached the maximum value ACMmax

Appendix A**Factory settings made by AT&F**

ATE1 (only in case of RCCP mode)

ATQ0

ATV1

AT+CREG=0

AT+CLIP=0

AT+CRC=0

AT+CAOC=0

AT+CMEE=0

AT+CPBS=SM (if available)

AT+COPS=0

AT+VTS=1

AT+CSCS="GSM"

AT+CSSN=0,0

AT^SCK=0

Reset pending locks (Phone Pin/Puk, Pin2/Puk2 ...)

which are give as answer of AT+CPIN?

AT+CSMS=0

AT+CNMI=0,0,0,0,1

AT^SMGO=0

AT+CSCB=0

Features of the Telephone-Book Memory

Name	Description	Category / Access	Write	Delete completely
FD	Fix-dialing number (SIM fix-dialing telephone book)	GSM 07.07 / +CPBS or ^SPBS	Allowed (PIN2 required)	
SM	Abbreviate dialing number (SIM telephone book)	GSM 07.07 / +CPBS or ^SPBS	Allowed (device code required if FDN replacement is active)	
DC (MD)	Mobile last dialing number (last number redial memory; only if "LD" is not available)	GSM 07.07 / +CPBS or ^SPBS	Not allowed	By means of AT^SDLD
ON (OW)	Own Numbers (SIM own telephone numbers)	GSM 07.07 (Siemens) / +CPBS (historical)	Allowed	

LD	SIM last dialing number (last number redial memory on SIM)	GSM 07.07 / +CPBS or ^SPBS	Not allowed	By means of AT^SDLD
ME	Mobile-equipment telephone book (ME dialing numbers)	GSM 07.07 / +CPBS or ^SPBS	Allowed (device code required if FDN replacement is active)	
BD	Barred dialing numbers (blocked numbers)	Siemens / ^SPBS	Not allowed	
SD	Service dialing numbers (Service numbers)	Siemens / ^SPBS	Not allowed	
MC (MS)	Missed dialing numbers (unanswered calls)	GSM 07.07 (Siemens) / +CPBS, ^SPBS	Not allowed	
RC (CD)	Callback dialing numbers (answered calls)	GSM 07.07 (Siemens) / +CPBS, ^SPBS	Not allowed	
BL	Blacklist dialing numbers (numbers that are blocked for a certain time in order to prevent continuous accesses from remote control)	Siemens / ^SPBS	Not allowed	
MB	Mailbox dialing numbers (network-operator mailbox)	Siemens / ^SPBS	Not allowed	
CS	Common sortable numbers (sorted combination of "SM", "ME", "FD")	Siemens / ^SPBS /^SPBC / ^SPBG	Not allowed	
RD	Red book numbers ("CS" entries with 'I' at the end of the name portion)	Siemens / ^SPBS /^SPBC / ^SPBG	Not allowed	

Writing to the FDN Phonebook / FDN Replacement

Writing to the fix-dialing number phonebook is protected by PIN2.
A Write sequence (to e.g. record 5) runs as follows:

```

AT+CMEE=2                                //Activate expanded error message
OK

AT+CPBS=?                                // Listing of available telephone books
+CPBS: ("FD","SM","LD")
OK

AT+CPBS="FD"                              // Selection of the FDN telephone book
OK

AT+CPBW=5,"1234","test"                  // A Write to record 5 is attempted...
+CME ERROR: SIM PIN2 REQUIRED              // ... PIN2 is required for this purpose

AT+CPIN?                                  // Query of the PIN status...
+CPIN: SIM PIN2                          // ... PIN2 is to be entered

AT+CPIN="12345678"                        // Input of PIN2
OK

AT+CPBW=5,"1234","test"                  // A Write to record 5 is attempted...
OK                                         // PIN2 remains active as long as you use the commands
                                           // +CPIN, +CPBS, +CPBR, +CPBW, +CACM,
                                           // +CMM, +CPUC
                                           // ^SPIC, ^SPBS, ^SPBC, ^SPBG.:
                                           // If you use other commands or if none of the
                                           // above commands are executed within five
                                           // minutes, the validity of PIN2 is voided.

AT+CPBW=6,"5678","new test"              // A Write to record 6 is attempted...
OK

```

...

In addition, if there is no FDN phonebook available on the SIM, it is possible to activate a feature which activates FDN-like behavior for the "SM" and "ME" phonebooks (FDN replacement). (Currently this feature can only be activated via the MMI lock/device lock/excluding telephone book.)

In this case, the Write to the "SM" and "ME" phonebooks is ensured by the device code (PH-SIM PIN and PH-SIM PUK, respectively).

The sequence for entering the device code is analogous to the above example.

Special hints for using +CPBR/+CPBW command

String parameters ,like the <text> in +CPBW command shall be entered using quotation marks "" (Ascii=Windows=GSM=0x22).e.g. "Doe Joe"

It is possible to enter string parameters without "" but not recommended, because following problems may occur:

If no "" are used:

- SPACES (Space, Blank, Ascii=Windows=GSM=0x20) are skipped.
E.g. at+cpbw=1,"123",,K. H. results in "K.H." ☹
 at+cpbw=1,"123",,"K. H." SPACES are kept ☺
- No ` ` (Ascii=Windows=GSM=0x2C) and
 ` ` (Ascii=Windows=GSM=0x3B) in <text> is possible, because this characters
are used as separator of parameters/commands.
E.g. at+cpbw=1,"123",,Kurz,Helmut result in ERROR ☹
 at+cpbw=1,"123",,"Kurz,Helmut" ☺

But there are also some points to note when using quotation marks "":

There are some characters which cannot be entered in normal way:

e.g. quotation mark "" character itself, because this is interpreted as the end of the <text>

To make this (and some other special characters) possible to be entered, the character with hex value 0x5c is used as escape character. In the ASCII character set this is equal the ` ` ,like proposed in V.25ter. (Ascii=Windows=0x5C)
Unfortunately there is no ` ` in GSM character set. The 0x5C equals the `Ö`

The escape sequence has the following structure:

- The sequence begins with the escape character 0x5C
(ASCII=Windows=` ` , GSM=`Ö`)
- The special character follows and is entered 2 Byte representation of the GSM character set value .
e.g. the 2 Byte representation of the `@` (GSM=0x00) is `00`

Following special characters shall be entered by using the escape sequence:

GSM Char	Hex char.	ASCII	GSM Esc Seq	Seq.(hex)	Note
Ö	5C	\	Ö5C	5C 35 43	Backslash
"	22	"	Ö22	5C 32 32	String delimiter
ö	08	BSP	Ö08	5C 30 38	Backspace
@	00	NULL	Ö00	5C 30 30	GSM NULL

Examples:

GSM string wanted in Phonebook	String in AT+CPBW Command(GSM)	String in AT+CPBW Command (Hex)
Ölhändler	"Ö5CihÖ7Bndler"	22 5C 35 43 6C 68 7B 6E 64 6C 65 72 22
"Eddi" Kurz	"Ö22EddiÖ22 Kurz"	22 5C 32 32 45 64 64 69 5C 32 32 20 4B 75 72 7A 22
Oöö	*ÖÖ8ö" 22 4F 5C	30 38 6F 22
@Adr.	"ÖÖÖAdr."	22 5C 30 30 41 64 72 2E 22 [no problems with strlen()]
	"@Adr."	22 00 41 64 72 2E 22 [may cause problems with strlen() in application]

Note:

When reading phonebook records, there is NO replacement. Every character will appear in normal GSM character set notation (like the left column in the example above).

Appendix B

Example for creating/interrogation of an organizer entry

-vcs object which has to be uploaded:

```
BEGIN:VCALENDAR
VERSION:1.0
BEGIN:VEVENT
CATEGORIES:ANNIVERSARY
DTSTART:19991213T100000
DESCRIPTION:W. von Siemens
END:VEVENT
END:VCALENDAR
```

-hexadecimal representation of this object:

```
424547494E3A5643414C454E4441520D0A56455253494F4E3A312E300D0A4245
47494E3A564556454E540D0A43415445474F524945533A414E4E4956455253415
2590D0A445453544152543A3139393931323133543130303030300D0A44455343
52495054494F4E3A572E20766F6E205369656D656E730D0A454E443A56455645
4E540D0A454E443A5643414C454E4441520D0A
```

-upload of an entry on record 20

```
at^sbnw="vcs",20,1,3<CR>
<CR><LF> > <Space>
424547494E3A5643414C454E4441520D0A56455253494F4E3A312E300D0A4245
47494E3A564556454E540D0A43415445474F<Ctrl-Z>
<CR><LF>OK<CR><LF>
```

```
at^sbnw="vcs",20,2,3<CR>
<CR><LF> > <Space>
524945533A414E4E49564552534152590D0A445453544152543A3139393931323
133543130303030300D0A44455343524950<Ctrl-Z>
<CR><LF>OK<CR><LF>
```

```
at^sbnw="vcs",20,3,3<CR>
<CR><LF> > <Space>
54494F4E3A572E20766F6E205369656D656E730D0A454E443A564556454E540D
0A454E443A5643414C454E4441520D0A<Ctrl-Z>
<CR><LF>OK<CR><LF>
```

All characters are answered with an echo.

Echoing can be switched off with „ATE0“.

In this example the organizer entry is uploaded in 50 bytes packets (100 input characters in every pdu).

The blue painted characters characterize the responses of the mobile.

-Interrogation of the current <type>,<subtype>,<actNumber>,<maxNumber>

```
at^sbnw?<CR>
<CR><LF>^SBNW: "vcs",20,2,3<CR><LF>
<CR><LF>OK<CR><LF>
```

description: The actual object which is uploaded is an VCS object.
It has to be stored on record 20.
2 of 3 packets are already uploaded.

-deleting of record 20

```
at^sbnw="vcs",20,0<CR>
<CR><LF>OK<CR><LF>
```

-download entry from record 20

```
at^sbnr="vcs",20<CR>
<CR><LF>^SBNR: <space>"vcs",20,1,1<CR><LF>
424547494E3A5643414C454E4441520D0A56455253494F4E3A312E300D0A4245
47494E3A564556454E540D0A43415445474F524945533A414E4E4956455253415
2590D0A4445453544152543A31393939313231335431303030300D0A44455343
52495054494F4E3A572E20766F6E205369656D656E730D0A454E443A56455645
4E540D0A454E443A5643414C454E4441520D0A<CR><LF>
<CR><LF>OK<CR><LF>
```

The mobile segments the record entry in 176 byte (=176*2 characters) packets.

-Download of an empty record 20

```
at^sbnr="vcs",20<CR>
<CR><LF>OK<CR><LF>
```

-Test command of AT^SBNW

```
at^sbnw=?<CR>
<CR><LF>^SBNW: <space>("bmp",(0)),("mid",(0)),("vcs",(1-30)) <CR><LF>
<CR><LF>OK<CR><LF>
```

Features

- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Downloadable Flash Memory
 - SPI Serial Interface for Program Downloading
 - Endurance: 1,000 Write/Erase Cycles
- 2K Bytes EEPROM
 - Endurance: 100,000 Write/Erase Cycles
- 4V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low-power Idle and Power-down Modes
- Interrupt Recovery From Power-down
- Programmable Watchdog Timer
- Dual Data Pointer
- Power-off Flag

Description

The AT89S8252 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of downloadable Flash programmable and erasable read only memory and 2K bytes of EEPROM. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with downloadable Flash on a monolithic chip, the Atmel AT89S8252 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S8252 provides the following standard features: 8K bytes of downloadable Flash, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

The downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a serial programming interface and allows the program memory to be written to or read from unless Lock Bit 2 has been activated.



8-bit Microcontroller with 8K Bytes Flash

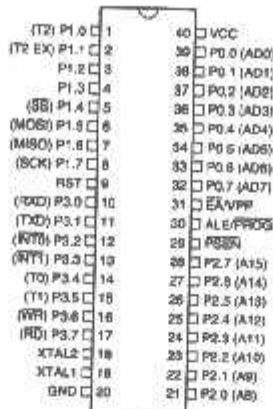
AT89S8252

Rev. 0401E-02/00

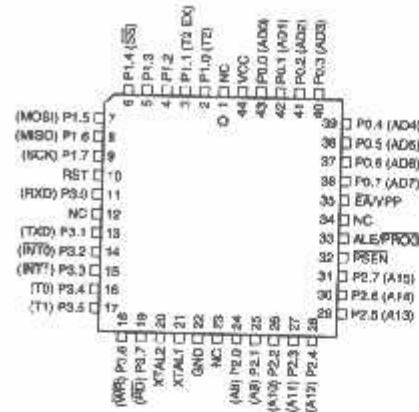


Pin Configurations

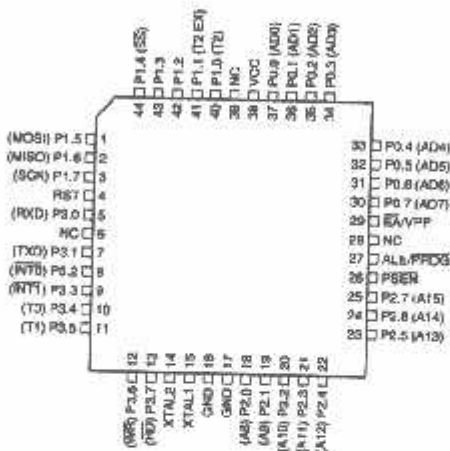
PDIP



PLCC



PQFP/TQFP



Pin Description

VCC
Supply voltage.

ND
Not used.

Port 0
Port 0 is an 8-bit open drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.
Port 0 can also be configured to be the multiplexed lower address/data bus during accesses to external

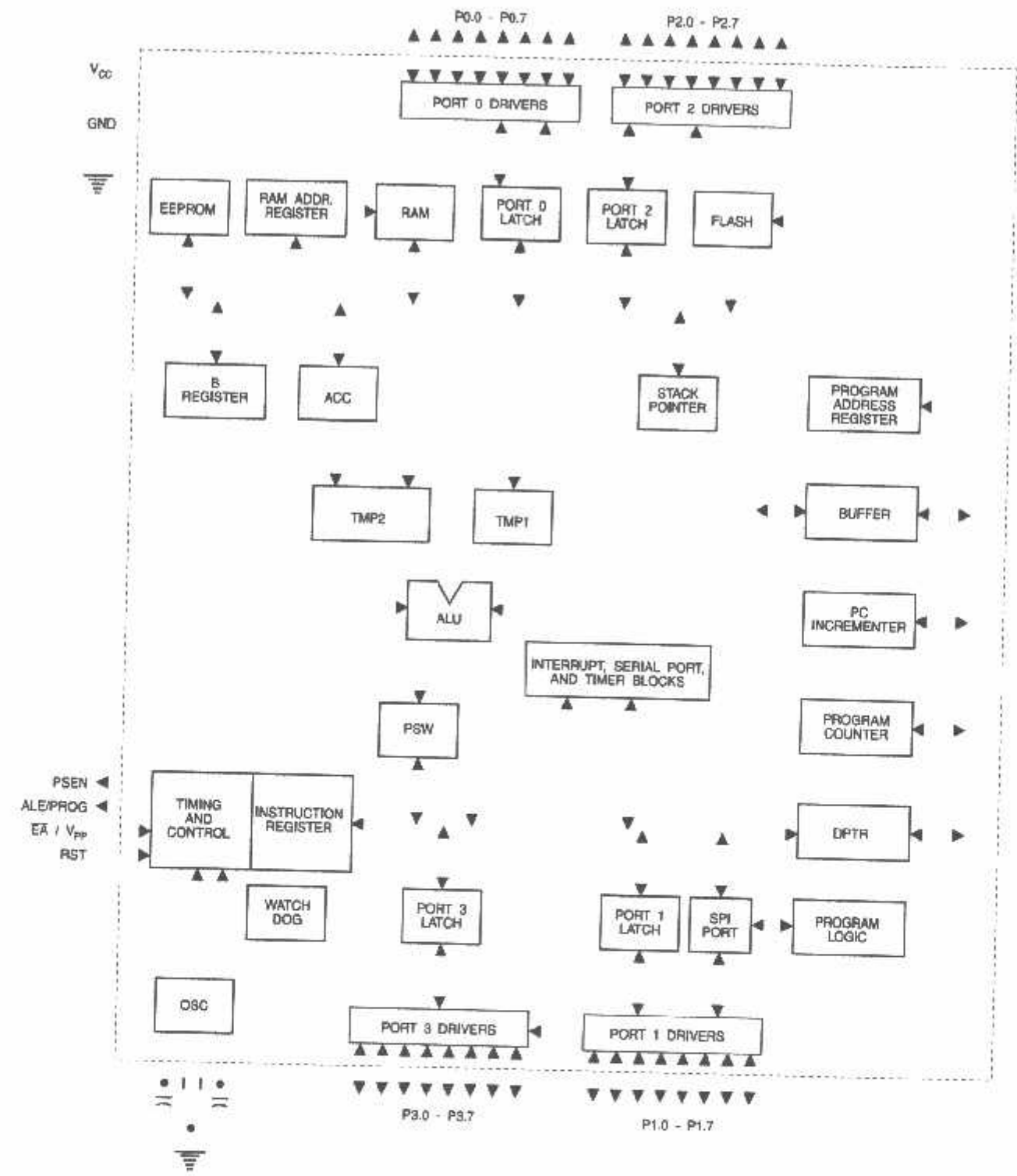
program and data memory. In this mode, P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

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

Block Diagram



Some Port 1 pins provide additional functions. P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively.

Pin Description

Furthermore, P1.4, P1.5, P1.6, and P1.7 can be configured as the SPI slave port select, data input/output and shift clock input/output pins as shown in the following table.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.4	$\overline{SS}$ (Slave port select input)
P1.5	MOSI (Master data output, slave data input pin for SPI channel)
P1.6	MISO (Master data input, slave data output pin for SPI channel)
P1.7	SCK (Master clock output, slave clock input pin for SPI channel)

Port 1 also receives the low-order address bytes during flash programming and verification.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

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

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

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs,

Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89S8252, as shown in the following table.

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

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

RST

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

ALE/PROG

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

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

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

PSEN

Program Store Enable is the read strobe to external program memory.

When the AT89S8252 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

$\overline{EA}/VPP$

External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external pro-

gram memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions. This pin also receives the 12-volt programming enable voltage (V_{pp}) during Flash programming when 12-volt programming is selected.

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Table 1. AT89S8252 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000					SPCR 000001XX		0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000		SPSR 00XXXXXX					0AFH
0A0H	P2 11111111							0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111						WMCON 00000010	97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR XXXXXXXX	PCON 0XXX0000 87H



Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted

locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 9) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16 bit capture mode or 16-bit auto-reload mode.

Table 2. T2CON—Timer/Counter 2 Control Register

T2CON Address = 0C8H

Reset Value = 0000 0000B

Bit Addressable

	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
Bit	7	6	5	4	3	2	1	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflows to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/T2	Timer or counter select for Timer 2. C/T2 = 0 for timer function. C/T2 = 1 for external event counter (falling edge triggered).
CP/RL2	Capture/Reload select. CP/RL2 = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/RL2 = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

Watchdog and Memory Control Register The WMCON register contains control bits for the Watchdog Timer (shown in Table 3). The EEMEN and EEMWE bits are used

to select the 2K bytes on-chip EEPROM, and to enable byte-write. The DPS bit selects one of two DPTR registers available.

Table 3. WMCON—Watchdog and Memory Control Register

WMCON Address = 96H					Reset Value = 0000 0010B			
Bit	PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN
	7	6	5	4	3	2	1	0
Symbol	Function							
PS2 PS1 PS0	Prescaler Bits for the Watchdog Timer. When all three bits are set to "0", the watchdog timer has a nominal period of 16 ms. When all three bits are set to "1", the nominal period is 2048 ms.							
EEMWE	EEPROM Data Memory Write Enable Bit. Set this bit to "1" before initiating byte write to on-chip EEPROM with the MOVX instruction. User software should set this bit to "0" after EEPROM write is completed.							
EEMEN	Internal EEPROM Access Enable. When EEMEN = 1, the MOVX instruction with DPTR will access on-chip EEPROM instead of external data memory. When EEMEN = 0, MOVX with DPTR accesses external data memory.							
DPS	Data Pointer Register Select. DPS = 0 selects the first bank of Data Pointer Register, DP0, and DPS = 1 selects the second bank, DP1.							
WDTRST RDY/BSY	Watchdog Timer Reset and EEPROM Ready/Busy Flag. Each time this bit is set to "1" by user software, a pulse is generated to reset the watchdog timer. The WDTRST bit is then automatically reset to "0" in the next instruction cycle. The WDTRST bit is Write-Only. This bit also serves as the RDY/BSY flag in a Read-Only mode during EEPROM write. RDY/BSY = 1 means that the EEPROM is ready to be programmed. While programming operations are being executed, the RDY/BSY bit equals "0" and is automatically reset to "1" when programming is completed.							
WDTEN	Watchdog Timer Enable Bit. WDTEN = 1 enables the watchdog timer and WDTEN = 0 disables the watchdog timer.							

PI Registers Control and status bits for the Serial Peripheral Interface are contained in registers SPCR (shown in Table 4) and SPSR (shown in Table 5). The SPI data bits are contained in the SPDR register. Writing the SPI data register during serial data transfer sets the Write Collision flag, WCOL, in the SPSR register. The SPDR is double buffered for writing and the values in SPDR are not changed by read.

Interrupt Registers The global interrupt enable bit and the individual interrupt enable bits are in the IE register. In addition, the individual interrupt enable bit for the SPI is in the SPCR register. Two priorities can be set for each of the interrupt sources in the IP register.

Dual Data Pointer Registers To facilitate accessing both internal EEPROM and external data memory, two banks of 16 bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR WMCON selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the appropriate value before accessing the respective Data Pointer Register.

Power Off Flag The Power Off Flag (POF) is located at bit_4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and reset under software control and is not affected by RESET.

Table 4. SPCR—SPI Control Register

SPCR Address = D5H

Reset Value = 0000 01XXB

Bit	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
	7	6	5	4	3	2	1	0

Symbol	Function												
SPIE	SPI Interrupt Enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts.												
SPE	SPI Enable. SPI = 1 enables the SPI channel and connects $\overline{SS}$, MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel.												
DORD	Data Order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission.												
MSTR	Master/Slave Select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects Slave SPI mode.												
CPOL	Clock Polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI Clock Phase and Polarity Control.												
CPHA	Clock Phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI Clock Phase and Polarity Control.												
SPR0 SPR1	SPI Clock Rate Select. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency, F_{osc}, is as follows: $SPR1SPR0 \quad SCK = F_{osc} \text{ divided by}$ <table border="1"> <tr> <td>0</td> <td>0</td> <td>4</td> </tr> <tr> <td>0</td> <td>1</td> <td>16</td> </tr> <tr> <td>1</td> <td>0</td> <td>64</td> </tr> <tr> <td>1</td> <td>1</td> <td>128</td> </tr> </table>	0	0	4	0	1	16	1	0	64	1	1	128
0	0	4											
0	1	16											
1	0	64											
1	1	128											

Table 5. SPSR – SPI Status Register

SPSR Address = AAH

Reset Value = 00XX XXXXB

	SPIF	WCOL	—	—	—	—	—	—
Bit	7	6	5	4	3	2	1	0

Symbol	Function
SPIF	SPI Interrupt Flag. When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE = 1 and ES = 1. The SPIF bit is cleared by reading the SPI status register with SPIF and WCOL bits set, and then accessing the SPI data register.
WCOL	Write Collision Flag. The WCOL bit is set if the SPI data register is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it has no effect. The WCOL bit (and the SPIF bit) are cleared by reading the SPI status register with SPIF and WCOL set, and then accessing the SPI data register.

Table 6. SPDR – SPI Data Register

PDR Address = 86H								Reset Value = unchanged							
Bit	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0							
	7	6	5	4	3	2	1	0							

AT89S8252

Data Memory – EEPROM and RAM

The AT89S8252 implements 2K bytes of on-chip EEPROM for data storage and 256 bytes of RAM. The upper 128 bytes of RAM occupy a parallel space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOV 0A0H, #data
```

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOV @R0, #data
```

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

The on-chip EEPROM data memory is selected by setting the EEMEN bit in the WMCON register at SFR address location 96H. The EEPROM address range is from 000H to FFH. The MOVX instructions are used to access the EPROM. To access off-chip data memory with the MOVX instructions, the EEMEN bit needs to be set to "0".

The EEMWE bit in the WMCON register needs to be set to "1" before any byte location in the EEPROM can be written. User software should reset EEMWE bit to "0" if no further EPROM write is required. EEPROM write cycles in the serial programming mode are self-timed and typically take 5 ms. The progress of EEPROM write can be monitored by reading the RDY/BSY bit (read-only) in SFR WMCON. RDY/BSY = 0 means programming is still in progress and RDY/BSY = 1 means EEPROM write cycle is completed and another write cycle can be initiated.

In addition, during EEPROM programming, an attempted read from the EEPROM will fetch the byte being written with the MSB complemented. Once the write cycle is completed, true data are valid at all bit locations.

Programmable Watchdog Timer

The programmable Watchdog Timer (WDT) operates from an independent oscillator. The prescaler bits, PS0, PS1 and PS2 in SFR WMCON are used to set the period of the Watchdog Timer from 16 ms to 2048 ms. The available timer periods are shown in the following table and the

actual timer periods (at $V_{CC} = 5V$) are within $\pm 30\%$ of the nominal.

The WDT is disabled by Power-on Reset and during Power-down. It is enabled by setting the WDTEN bit in SFR WMCON (address = 96H). The WDT is reset by setting the WDRST bit in WMCON. When the WDT times out without being reset or disabled, an internal RST pulse is generated to reset the CPU.

Table 7. Watchdog Timer Period Selection

WDT Prescaler Bits			Period (nominal)
PS2	PS1	PS0	
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms

Timer 0 and 1

Timer 0 and Timer 1 in the AT89S8252 operate the same way as Timer 0 and Timer 1 in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-45, section titled, "Timer/Counters."

Timer 2

Timer 2 is a 16 bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/T2 in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 8.

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which

he transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

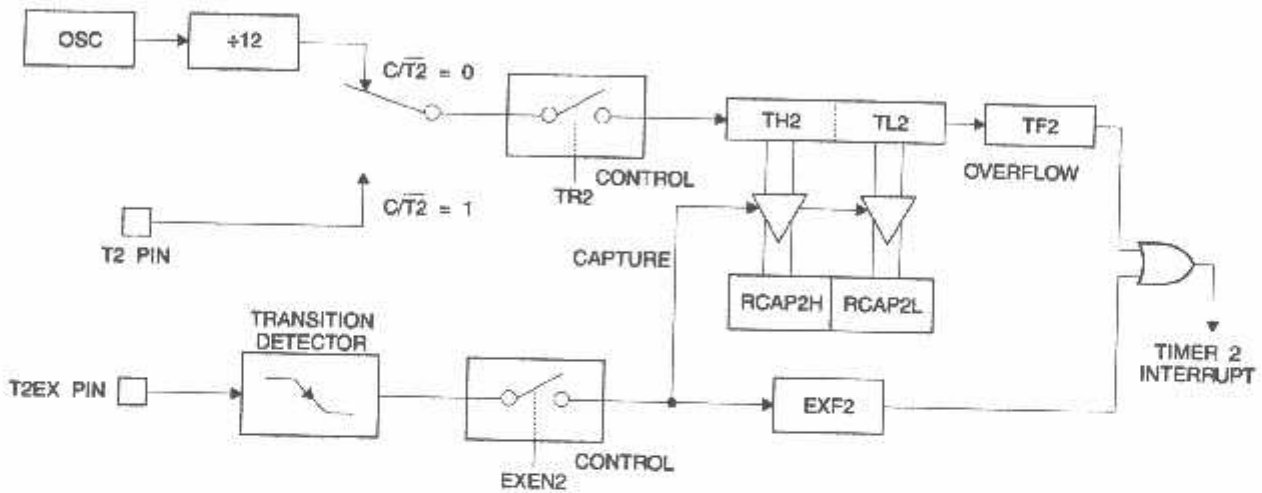
Table 8. Timer 2 Operating Modes

RCLK + TCLK	CP/RL2	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16 bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.

Figure 1. Timer 2 in Capture Mode



Auto-reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16 bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 9). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16 bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16 bit reload can be triggered either by an overflow or

by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16 bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

Figure 2. Timer 2 in Auto Reload Mode (DCEN = 0)

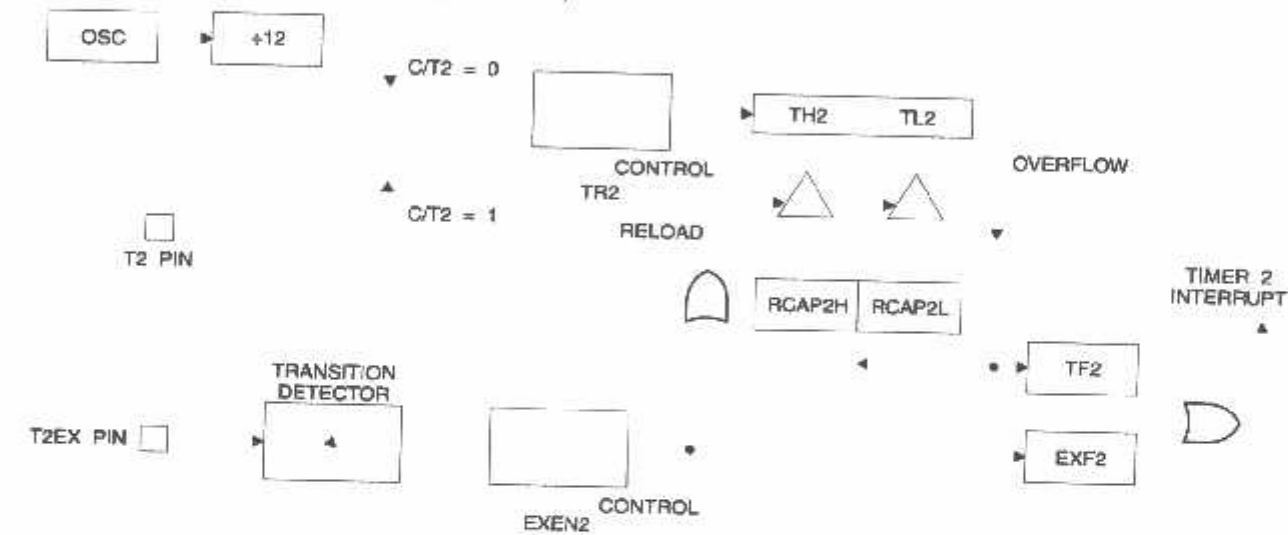


Table 9. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H								Reset Value = XXXX XX00B	
Not Bit Addressable									
Bit	7	6	5	4	3	2	1	0	
							T2OE	DCEN	
Symbol	Function								
	Not implemented, reserved for future use.								
T2OE	Timer 2 Output Enable bit.								
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter.								



Figure 3. Timer 2 Auto Reload Mode (DCEN = 1)

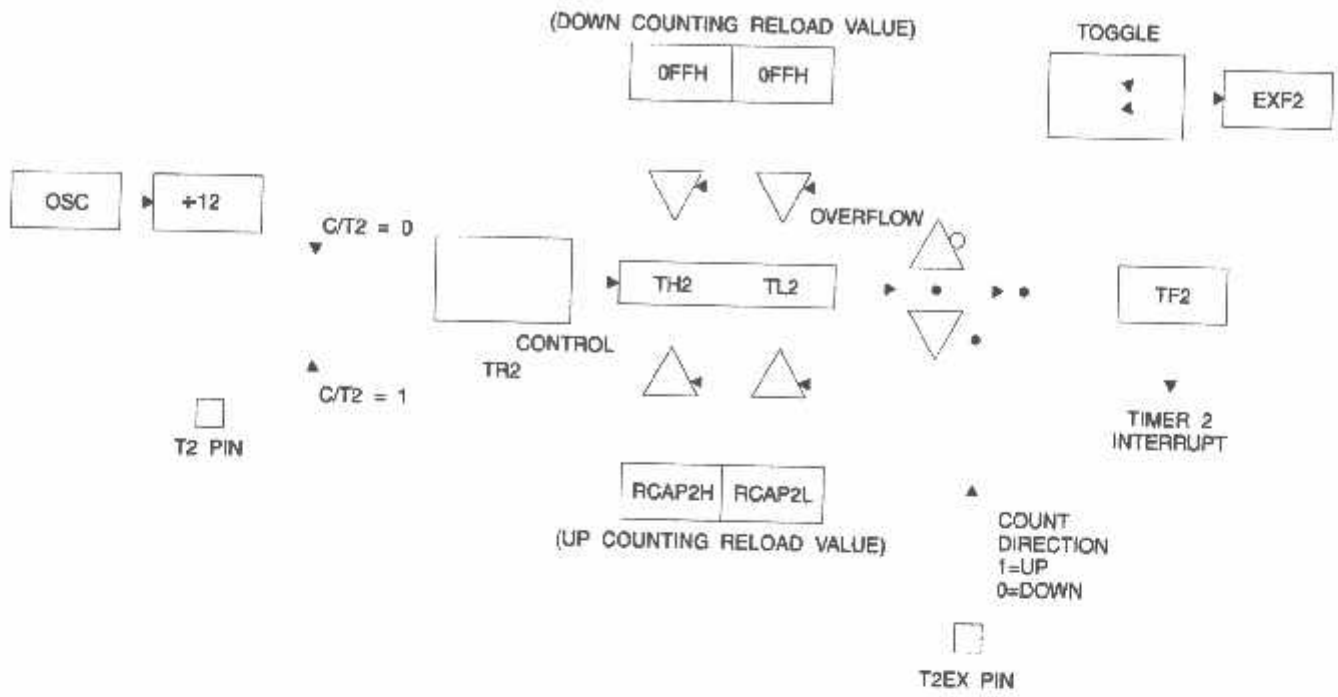
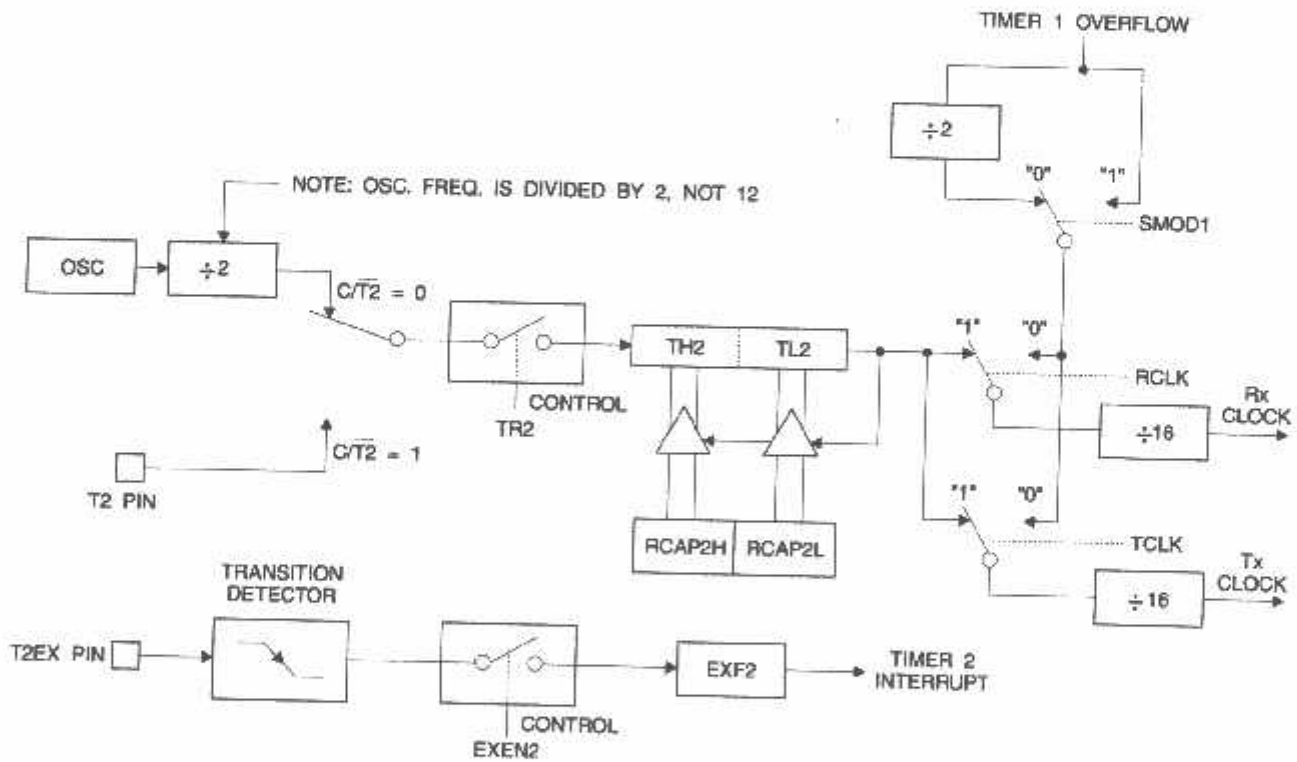


Figure 4. Timer 2 in Baud Rate Generator Mode



Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16 bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation (CP/T2 = 0). The timer operation is different for timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

$$\frac{\text{Modes 1 and 3 Baud Rate}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

here (RCAP2H, RCAP2L) is the content of RCAP2H and CAP2L taken as a 16 bit unsigned integer.

mer 2 as a baud rate generator is shown in Figure 4. This pure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload in (RCAP2H, RCAP2L) to (TH2, TL2). This means that

2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit C/T2 (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

In the clock-out mode, Timer 2 rollovers will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

Figure 5. Timer 2 in Clock-out Mode

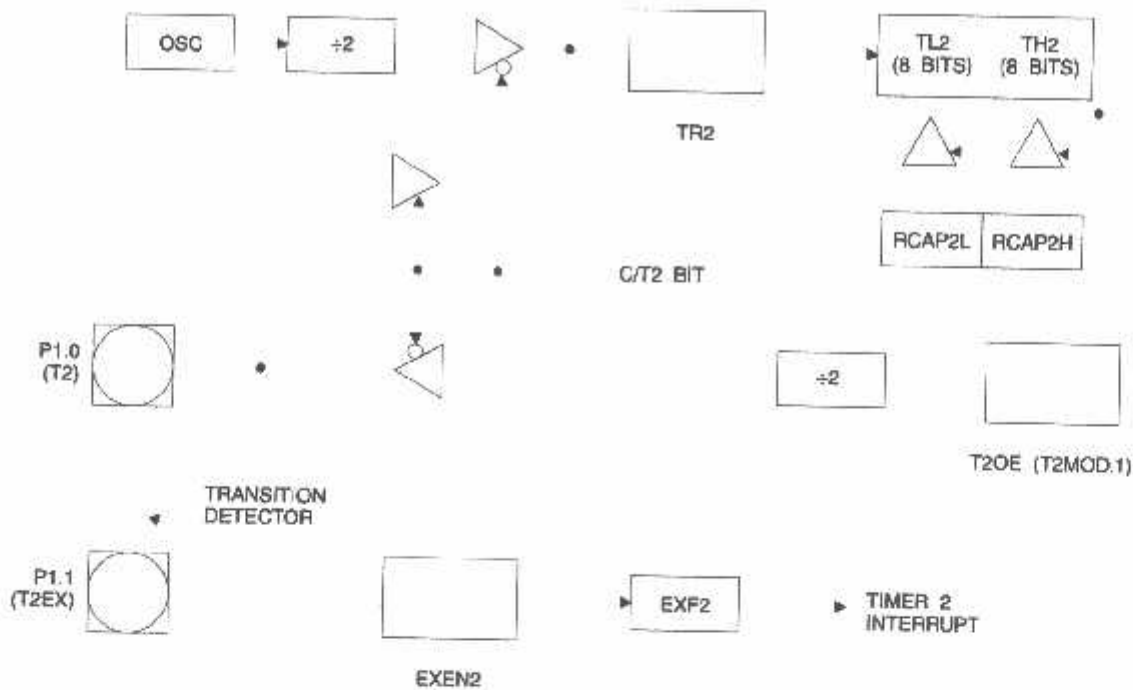
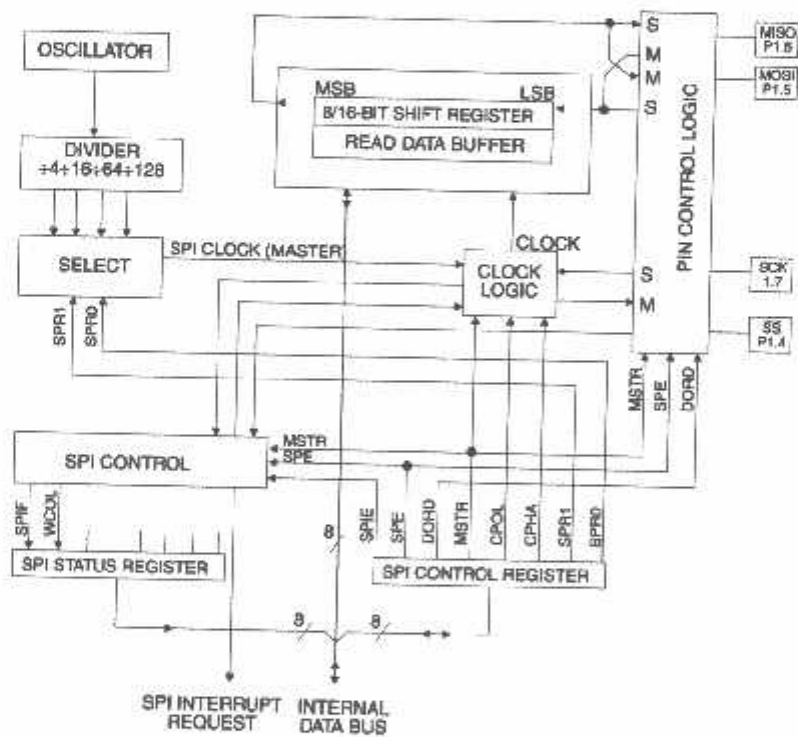


Figure 6. SPI Block Diagram



UART

The UART in the AT89S8252 operates the same way as the UART in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-49, section titled, "Serial Interface."

Serial Peripheral Interface

The serial peripheral interface (SPI) allows high-speed synchronous data transfer between the AT89S8252 and peripheral devices or between several AT89S8252 devices. The AT89S8252 SPI features include the following:

- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- 1.5 MHz Bit Frequency (max.)
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag

- Write Collision Flag Protection
- Wakeup from Idle Mode (Slave Mode Only)

The interconnection between master and slave CPUs with SPI is shown in the following figure. The SCK pin is the clock output in the master mode but is the clock input in the slave mode. Writing to the SPI data register of the master CPU starts the SPI clock generator, and the data written shifts out of the MOSI pin and into the MISO pin of the slave CPU. After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF). If both the SPI interrupt enable bit (SPIE) and the serial port interrupt enable bit (ES) are set, an interrupt is requested.

The Slave Select input, $\overline{SS}/P1.4$, is set low to select an individual SPI device as a slave. When $\overline{SS}/P1.4$ is set high, the SPI port is deactivated and the MOSI/P1.5 pin can be used as an input.

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 8 and Figure 9.

Figure 7. SPI Master-slave Interconnection

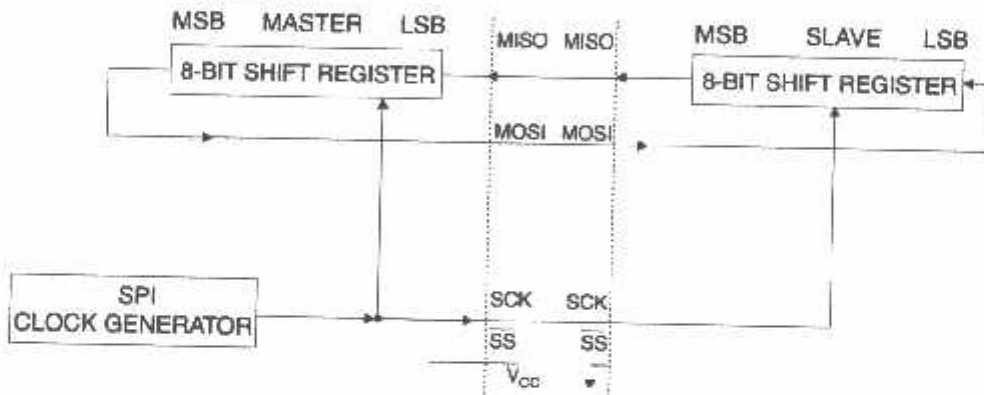
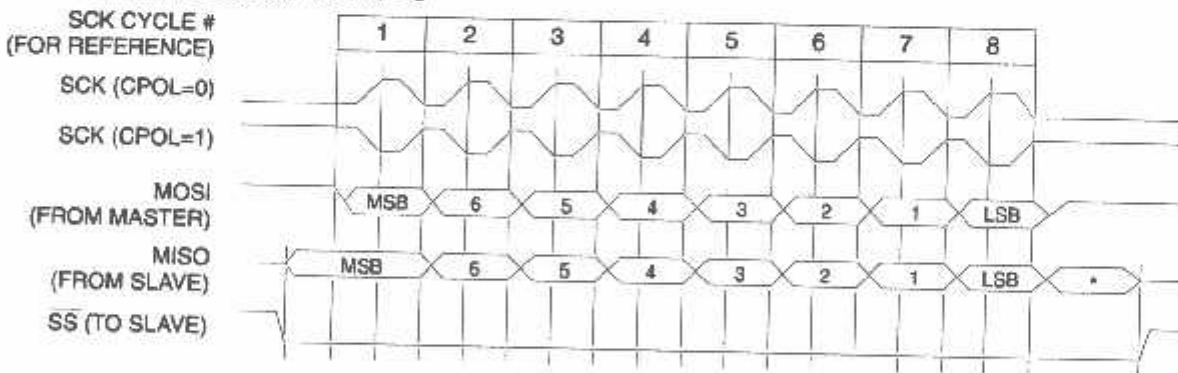
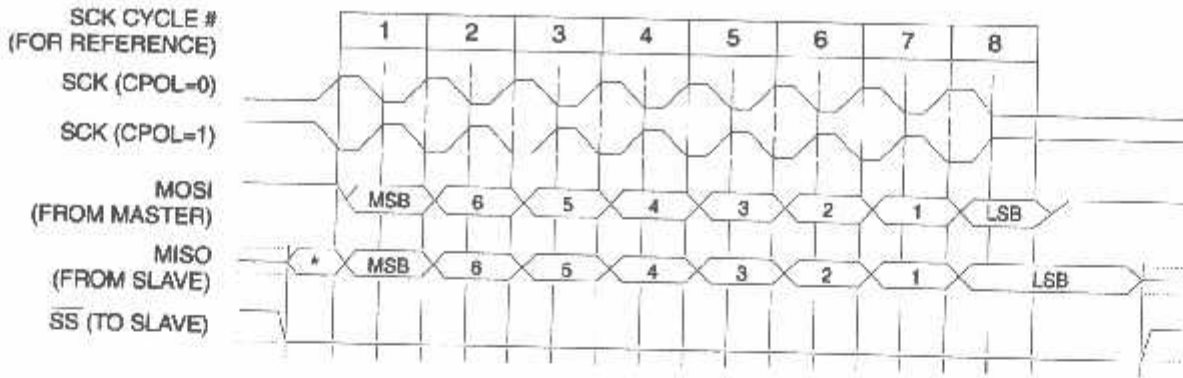


Figure 8. SPI transfer Format with CPHA = 0



It is defined but normally MSB of character just received

Figure 9. SPI Transfer Format with CPHA = 1



Not defined but normally LSB of previously transmitted character

Interrupts

The AT89S8252 has a total of six interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 10.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table 10 shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

Timer 0 and Timer 1 flags, TF0 and TF1, are set at the end of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, Timer 2 flag, TF2, is set at S2P2 and is polled in the next cycle in which the timer overflows.

Table 10. Interrupt Enable (IE) Register

(MSB)(LSB)

EA	—	ET2	ES	ET1	EX1	ET0	EX0
----	---	-----	----	-----	-----	-----	-----

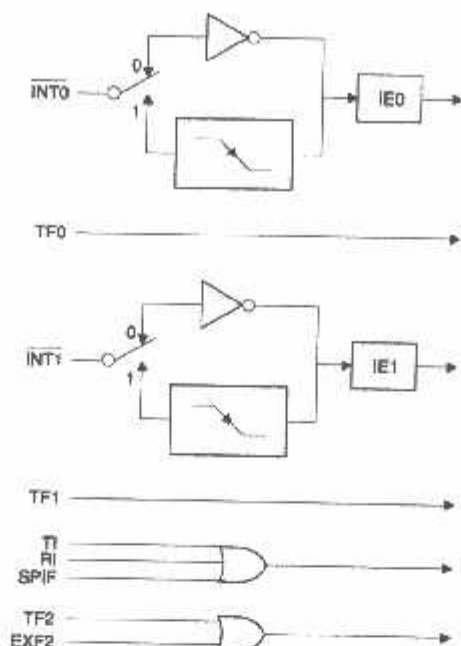
Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	SPI and UART interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.

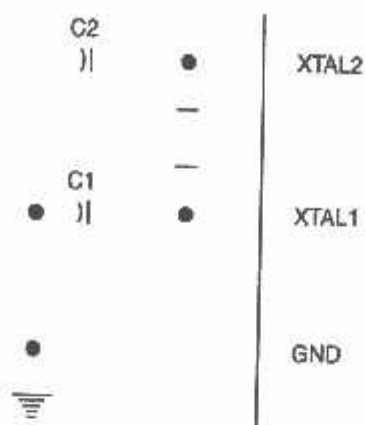
Figure 10. Interrupt Sources



Oscillator Characteristics

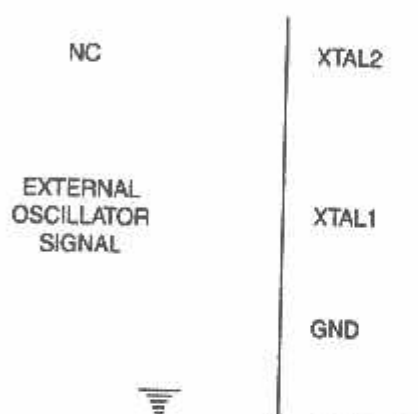
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 12. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 11. Oscillator Connections



Note: Note: C1, C2 = 30 pF $\pm$ 10 pF for Crystals
= 40 pF $\pm$ 10 pF for Ceramic Resonators

Figure 12. External Clock Drive Configuration



Idle Mode

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

Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution

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

Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

Power-down Mode

In the power-down mode, the oscillator is stopped and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. Exit from power-down can be initiated either by hardware reset or by an enabled external interrupt. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

To exit power-down via an interrupt, the external interrupt must be enabled as level sensitive before entering power-down. The interrupt service routine starts at 16 ms (nominal) after the enabled interrupt pin is activated.

Program Memory Lock Bits

The AT89S8252 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

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

Once programmed, the lock bits can only be unprogrammed with the Chip Erase operations in either the parallel or serial modes.

Lock Bit Protection Modes⁽¹⁾⁽²⁾

Program Lock Bits				Protection Type
	LB1	LB2	LB3	
1	U	U	U	No internal memory lock feature.
2	P	U	U	MOVX instructions executed from external program memory are disabled from fetching code bytes from internal memory. $\overline{EA}$ is sampled and latched on reset and further programming of the Flash memory (parallel or serial mode) is disabled.
3	P	P	U	Same as Mode 2, but parallel or serial verify are also disabled.
4	P	P	P	Same as Mode 3, but external execution is also disabled.

Notes: 1. U = Unprogrammed
2. P = Programmed

Programming the Flash and EEPROM

Atmel's AT89S8252 Flash Microcontroller offers 8K bytes of in-system reprogrammable Flash Code memory and 2K bytes of EEPROM Data memory.

The AT89S8252 is normally shipped with the on-chip Flash Code and EEPROM Data memory arrays in the erased state (i.e. contents = FFH) and ready to be programmed. This device supports a High-voltage (12V) Parallel programming mode and a Low-voltage (5V) Serial programming mode. The serial programming mode provides a convenient way to download the AT89S8252 inside the user's system. The parallel programming mode is compatible with conventional third party Flash or EPROM programmers.

The Code and Data memory arrays are mapped via separate address spaces in the serial programming mode. In the parallel programming mode, the two arrays occupy one contiguous address space: 0000H to 1FFFH for the Code array and 2000H to 27FFH for the Data array.

The Code and Data memory arrays on the AT89S8252 are programmed byte-by-byte in either programming mode. An auto-erase cycle is provided with the self-timed programming operation in the serial programming mode. There is no need to perform the Chip Erase operation to reprogram any memory location in the serial programming mode unless any of the lock bits have been programmed.

In the parallel programming mode, there is no auto-erase cycle. To reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

Parallel Programming Algorithm: To program and verify the AT89S8252 in the parallel programming mode, the following sequence is recommended:

- Power-up sequence:
 - Apply power between V_{CC} and GND pins.
 - Set RST pin to "H".
 - Apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
 - Set $\overline{PSEN}$ pin to "L".
 - ALE pin to "H".
 - $\overline{EA}$ pin to "H" and all other pins to "H".
 - Apply the appropriate combination of "H" or "L" logic levels to pins P2.6, P2.7, P3.6, P3.7 to select one of the programming operations shown in the Flash Programming Modes table.
 - Apply the desired byte address to pins P1.0 to P1.7 and P2.0 to P2.5.
 - Apply data to pins P0.0 to P0.7 for Write Code operation.

5. Raise $\overline{EA}/V_{PP}$ to 12V to enable Flash programming, erase or verification.
6. Pulse ALE/ $\overline{PROG}$ once to program a byte in the Code memory array, the Data memory array or the lock bits. The byte-write cycle is self-timed and typically takes 1.5 ms.
7. To verify the byte just programmed, bring pin P2.7 to "L" and read the programmed data at pins P0.0 to P0.7.
8. Repeat steps 3 through 7 changing the address and data for the entire 2K or 8K bytes array or until the end of the object file is reached.
9. Power-off sequence:
 - Set XTAL1 to "L".
 - Set RST and $\overline{EA}$ pins to "L".
 - Turn V_{CC} power off.

In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

Data Polling: The AT89S8252 features $\overline{DATA}$ Polling to indicate the end of a write cycle. During a write cycle in the parallel or serial programming mode, an attempted read of the last byte written will result in the complement of the written datum on P0.7 (parallel mode), and on the MSB of the serial output byte on MISO (serial mode). Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{DATA}$ Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The progress of byte programming in the parallel programming mode can also be monitored by the RDY/ $\overline{BSY}$ output signal. Pin P3.4 is pulled Low after ALE goes High during programming to indicate $\overline{BSY}$. P3.4 is pulled High again when programming is done to indicate $\overline{RDY}$.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed Code or Data byte can be read back via the address and data lines for verification. The state of the lock bits can also be verified directly in the parallel programming mode. In the serial programming mode, the state of the lock bits can only be verified indirectly by observing that the lock bit features are enabled.

Chip Erase: Both Flash and EEPROM arrays are erased electrically at the same time. In the parallel programming mode, chip erase is initiated by using the proper combination of control signals and by holding ALE/ $\overline{PROG}$ low for 10 ms. The Code and Data arrays are written with all "1"s in the Chip Erase operation.



In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase Instruction. In this mode, chip erase is self-timed and takes about 16 ms.

During chip erase, a serial read from any address location will return 00H at the data outputs.

Serial Programming Fuse: A programmable fuse is available to disable Serial Programming if the user needs maximum system security. The Serial Programming Fuse can only be programmed or erased in the Parallel Programming Mode.

The AT89S8252 is shipped with the Serial Programming Mode enabled.

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

(030H) = 1EH indicates manufactured by Atmel
(031H) = 72H indicates 89S8252

Programming Interface

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

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

Serial Downloading

Both the Code and Data memory arrays can be programmed using the serial SPI bus while RST is pulled to $\overline{L}$. The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before program/erase operations can be executed.

A full auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the Chip Erase instruction unless any of the 16k bits have been programmed. The Chip Erase operation turns the content of every memory location in both the Code and Data arrays into FFH.

The Code and Data memory arrays have separate address spaces:

0000H to 1FFFFH for Code memory and 000H to 7FFFH for Data memory.

Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/40 of the crystal frequency. With a 24 MHz oscillator clock, the maximum SCK frequency is 600 kHz.

Serial Programming Algorithm

To program and verify the AT89S8252 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:

Apply power between VCC and GND pins.
Set RST pin to "H".

If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.

2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 40.

3. The Code or Data array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The selected memory location is first automatically erased before new data is written. The write cycle is self-timed and typically takes less than 2.5 ms at 5V.

4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.

5. At the end of a programming session, RST can be set low to commence normal operation.

Power-off sequence (if needed):

Set XTAL1 to "L" (if a crystal is not used).

Set RST to "L".

Turn VCC power off.

Serial Programming Instruction

The Instruction Set for Serial Programming follows a 3-byte protocol and is shown in the following table:

Instruction Set

Instruction	Input Format			Operation
	Byte 1	Byte 2	Byte 3	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	Enable serial programming interface after RST goes high.
Chip Erase	1010 1100	xxxx x100	xxxx xxxx	Chip erase both 8K & 2K memory arrays.
Read Code Memory	aaaa a001	low addr	xxxx xxxx	Read data from Code memory array at the selected address. The 5 MSBs of the first byte are the high order address bits. The low order address bits are in the second byte. Data are available at pin MISO during the third byte.
Write Code Memory	aaaa a010	low addr	data in	Write data to Code memory location at selected address. The address bits are the 5 MSBs of the first byte together with the second byte.
Read Data Memory	00aa a101	low addr	xxxx xxxx	Read data from Data memory array at selected address. Data are available at pin MISO during the third byte.
Write Data Memory	00aa a110	low addr	data in	Write data to Data memory location at selected address.
Write Lock Bits	1010 1100	x x111	xxxx xxxx	Write lock bits. Set LB1, LB2 or LB3 = "0" to program lock bits.

- note:
1. DATA polling is used to indicate the end of a write cycle which typically takes less than 2.5 ms at 5V.
 2. "aaaaa" = high order address.
 3. "x" = don't care.

Flash and EEPROM Parallel Programming Modes

Mode	RST	PSEN	ALE/PROG	EA/V _{PP}	P2.6	P2.7	P3.6	P3.7	Data I/O P0.7:0	Address P2.5:0 P1.7:0
Serial Prog. Modes	H	h ⁽¹⁾	h ⁽¹⁾	x						
Chip Erase	H	L	 (2)	12V	H	L	L	L	X	X
Write (10K bytes) Memory	H	L		12V	L	H	H	H	DIN	ADDR
Read (10K bytes) Memory	H	L	H	12V	L	L	H	H	DOUT	ADDR
Write Lock Bits:	H	L		12V	H	L	H	L	DIN	X
Bit - 1									P0.7 = 0	X
Bit - 2									P0.6 = 0	X
Bit - 3									P0.5 = 0	X
Read Lock Bits:	H	L	H	12V	H	H	L	L	DOUT	X
Bit - 1									@P0.2	X
Bit - 2									@P0.1	X
Bit - 3									@P0.0	X
Read Atmel Code	H	L	H	12V	L	L	L	L	DOUT	30H
Read Device Code	H	L	H	12V	L	L	L	L	DOUT	31H
Serial Prog. Enable	H	L	 (2)	12V	L	H	L	H	P0.0 = 0	X
Serial Prog. Disable	H	L	 (2)	12V	L	H	L	H	P0.0 = 1	X
Read Serial Prog. Fuse	H	L	H	12V	H	H	L	H	@P0.0	X

- Notes:
- "h" = weakly pulled "High" internally.
 - Chip Erase and Serial Programming Fuse require a 10 ms PROG pulse. Chip Erase needs to be performed first before reprogramming any byte with a content other than FFH.
 - P3.4 is pulled Low during programming to indicate RDY/BSY.
 - "X" = don't care

Figure 13. Programming the Flash/EEPROM Memory

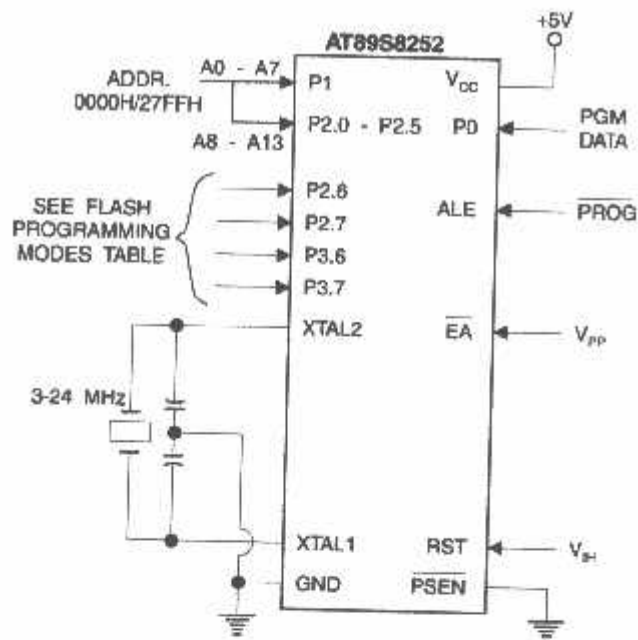


Figure 15. Flash/EEPROM Serial Downloading

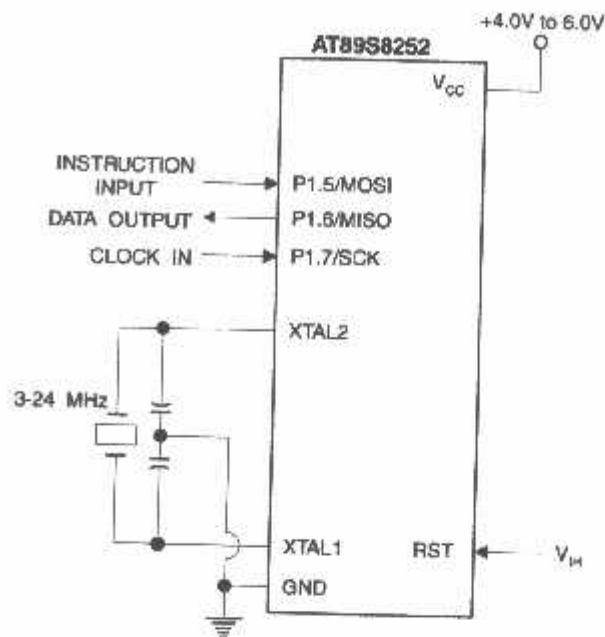
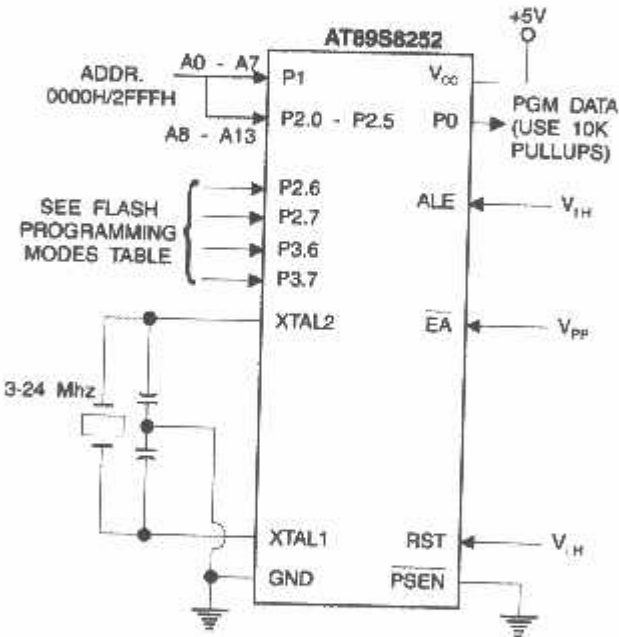


Figure 14. Verifying the Flash/EEPROM Memory

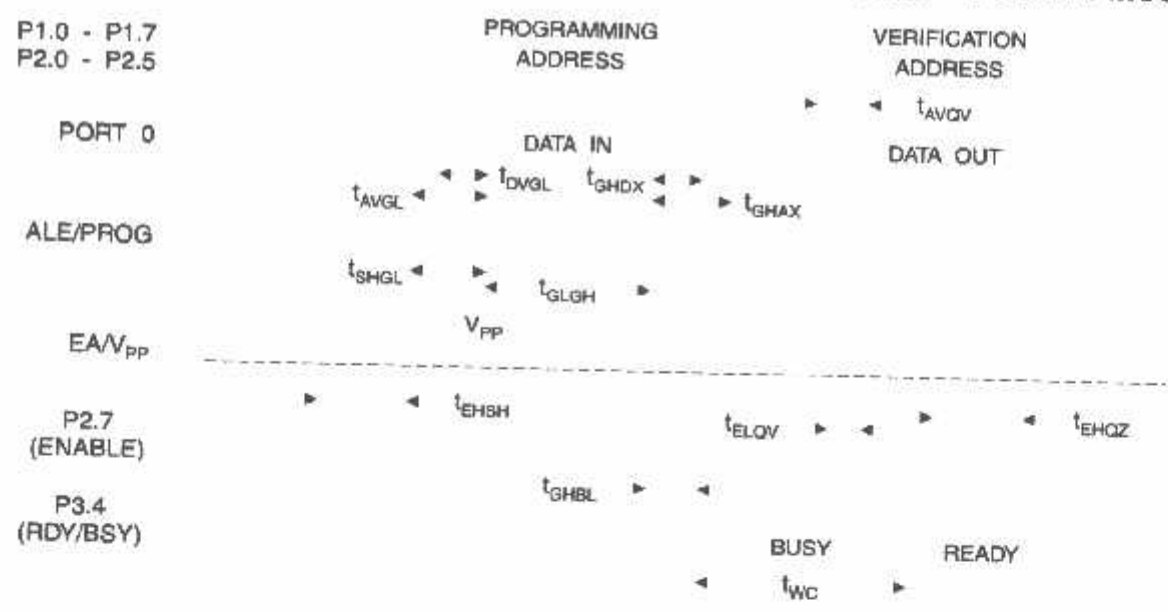


Flash Programming and Verification Characteristics – Parallel Mode

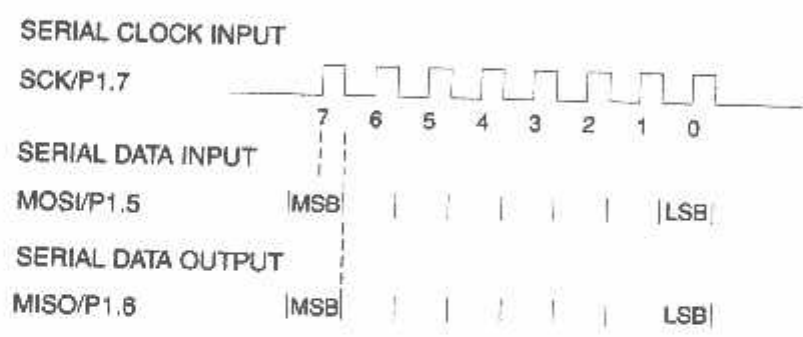
$T_A = 0^\circ\text{C to } 70^\circ\text{C}$, $V_{CC} = 5.0\text{V} \pm 10\%$

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold after $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{DVCL}	Data Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold after $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{ENSH}	P2.7 ($\overline{\text{ENABLE}}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
t_{GLGH}	$\overline{\text{PROG}}$ Width	1	110	μs
t_{AVGV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELOV}	$\overline{\text{ENABLE}}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float after $\overline{\text{ENABLE}}$	0	$48t_{CLCL}$	
t_{GHEL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	μs
t_{WC}	Byte Write Cycle Time		2.0	ms

Flash/EEPROM Programming and Verification Waveforms – Parallel Mode



Serial Downloading Waveforms



Absolute Maximum Ratings*

Operating Temperature.....	-55°C to +125°C
Storage Temperature.....	-65°C to +150°C
Voltage on Any Pin with Respect to Ground.....	-1.0V to +7.0V
Maximum Operating Voltage.....	6.6V
DC Output Current.....	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

The values shown in this table are valid for $T_A = -40^{\circ}\text{C}$ to 85°C and $V_{CC} = 5.0\text{V} \pm 20\%$, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage	(Except EA)	-0.5	$0.2 V_{CC} - 0.1$	V
V_{IL1}	Input Low-voltage (EA)		-0.5	$0.2 V_{CC} - 0.3$	V
V_{IH}	Input High-voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
V_{IH1}	Input High-voltage (XTAL1, RST)		$0.7 V_{CC}$	$V_{CC} + 0.5$	V
V_{OL}	Output Low-voltage ⁽¹⁾ (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.5	V
V_{OL1}	Output Low-voltage ⁽¹⁾ (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.5	V
V_{OH}	Output High-voltage (Ports 1,2,3, ALE, PSEN)	$I_{OH} = -60 \mu\text{A}$, $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
I_{OHI}	Output High-voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}$, $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
I_L	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	μA
I_L	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}$, $V_{CC} = 5\text{V} \pm 10\%$		-650	μA
I_I	Input Leakage Current (Port 0, EA)	$0.45 < V_{IN} < V_{CC}$		± 10	μA
R_{RST}	Reset Pull-down Resistor		50	300	$\text{K}\Omega$
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^{\circ}\text{C}$		10	pF
I_C	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode ⁽²⁾	$V_{CC} = 6\text{V}$		100	μA
		$V_{CC} = 3\text{V}$		40	μA

- Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: 10 mA
Maximum I_{OL} per 8-bit port:
Port 0: 26 mA
Ports 1, 2, 3: 15 mA

Maximum total I_{OL} for all output pins: 71 mA
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum V_{CC} for Power-down is 2V

AT89S8252

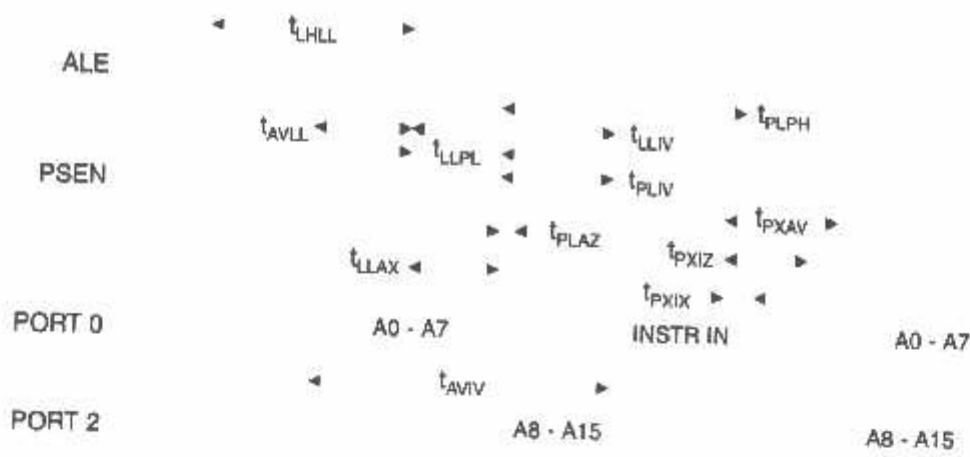
AC Characteristics

Under operating conditions, load capacitance for Port 0, ALE/PROG, and PSEN = 100 pF; load capacitance for all other outputs = 80 pF.

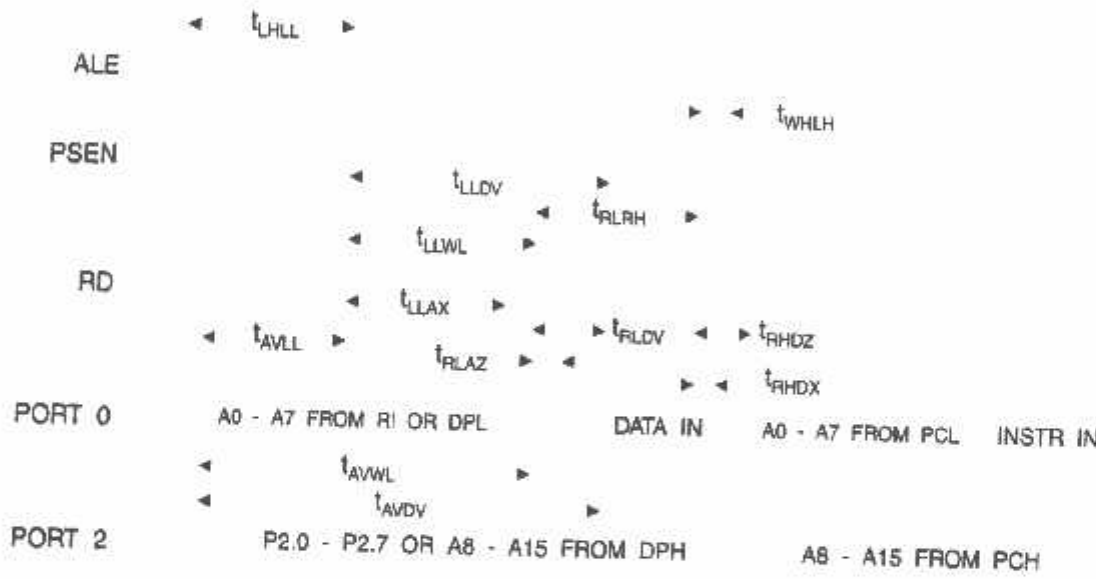
External Program and Data Memory Characteristics

Symbol	Parameter	Variable Oscillator		Units
		Min	Max	
$1/f_{\text{CLCL}}$	Oscillator Frequency	0	24	MHz
t_{LHL}	ALE Pulse Width	$2t_{\text{CLCL}} - 40$		ns
t_{AVLL}	Address Valid to ALE Low	$t_{\text{CLCL}} - 13$		ns
t_{LLAX}	Address Hold after ALE Low	$t_{\text{CLCL}} - 20$		ns
t_{LLIV}	ALE Low to Valid Instruction In		$4t_{\text{CLCL}} - 65$	ns
t_{LLPL}	ALE Low to PSEN Low	$t_{\text{CLCL}} - 13$		ns
t_{PLPH}	PSEN Pulse Width	$3t_{\text{CLCL}} - 20$		ns
t_{PLIV}	PSEN Low to Valid Instruction In		$3t_{\text{CLCL}} - 45$	ns
t_{PIXX}	Input Instruction Hold after PSEN	0		ns
t_{PIXZ}	Input Instruction Float after PSEN		$t_{\text{CLCL}} - 10$	ns
t_{PXAV}	PSEN to Address Valid	$t_{\text{CLCL}} - 8$		ns
t_{AVIV}	Address to Valid Instruction In		$5t_{\text{CLCL}} - 55$	ns
t_{PLAZ}	PSEN Low to Address Float		10	ns
t_{PLRH}	RD Pulse Width	$6t_{\text{CLCL}} - 100$		ns
t_{WLWH}	WR Pulse Width	$6t_{\text{CLCL}} - 100$		ns
t_{RLDV}	RD Low to Valid Data In		$5t_{\text{CLCL}} - 90$	ns
t_{RHDX}	Data Hold after RD	0		ns
t_{RHIZ}	Data Float after RD		$2t_{\text{CLCL}} - 28$	ns
t_{LLDV}	ALE Low to Valid Data In		$8t_{\text{CLCL}} - 150$	ns
t_{AVDV}	Address to Valid Data In		$9t_{\text{CLCL}} - 165$	ns
t_{LWL}	ALE Low to RD or WR Low	$3t_{\text{CLCL}} - 50$	$3t_{\text{CLCL}} + 50$	ns
t_{VWL}	Address to RD or WR Low	$4t_{\text{CLCL}} - 75$		ns
t_{VWX}	Data Valid to WR Transition	$t_{\text{CLCL}} - 20$		ns
t_{VWH}	Data Valid to WR High	$7t_{\text{CLCL}} - 120$		ns
t_{WHDX}	Data Hold after WR	$t_{\text{CLCL}} - 20$		ns
t_{LAZ}	RD Low to Address Float		0	ns
t_{HLH}	RD or WR High to ALE High	$t_{\text{CLCL}} - 20$	$t_{\text{CLCL}} + 25$	ns

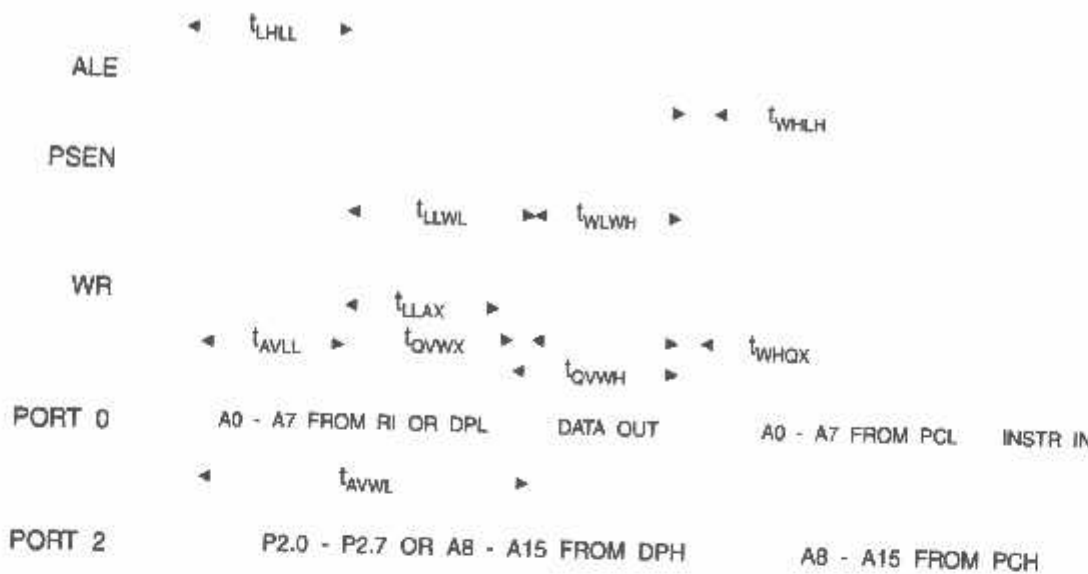
External Program Memory Read Cycle



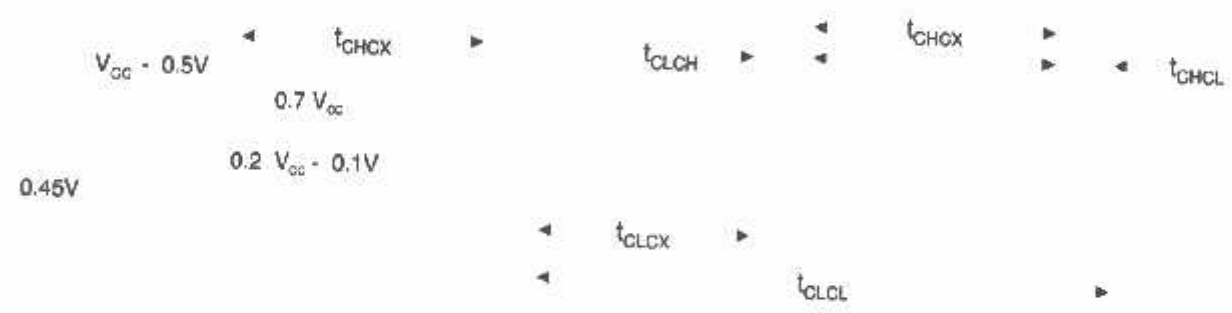
External Data Memory Read Cycle



External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	$V_{CC} = 4.0V \text{ to } 6.0V$		Units
		Min	Max	
f_{CLCL}	Oscillator Frequency	0	24	MHz
t_{LCL}	Clock Period	41.6		ns
t_{HCX}	High Time	15		ns
t_{LCX}	Low Time	15		ns
t_{LCH}	Rise Time		20	ns
t_{HCL}	Fall Time		20	ns

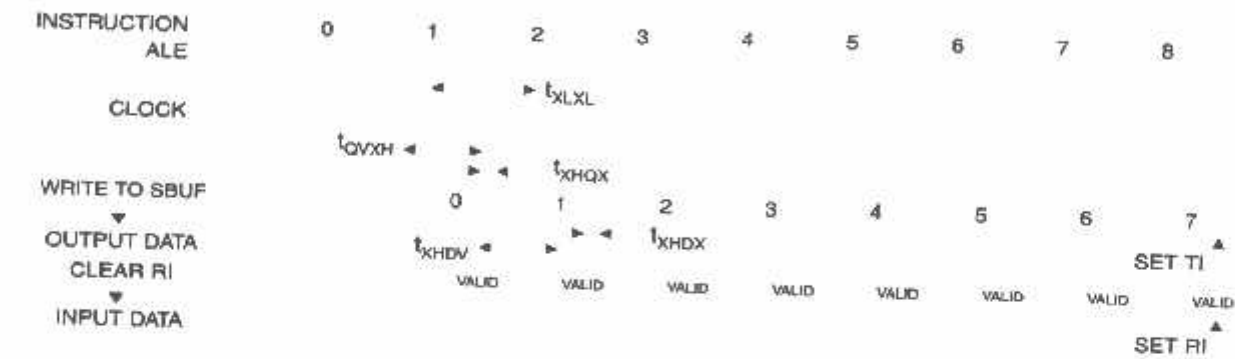


Serial Port Timing: Shift Register Mode Test Conditions

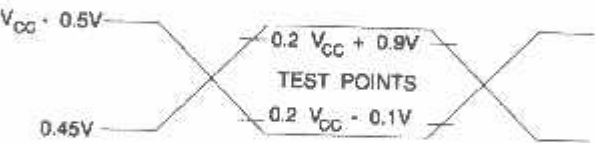
The values in this table are valid for $V_{CC} = 4.0V$ to $6V$ and Load Capacitance = 80 pF .

Symbol	Parameter	Variable Oscillator		Units
		Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	$12t_{CLCL}$		μs
t_{QVXH}	Output Data Setup to Clock Rising Edge	$10t_{CLCL} - 133$		ns
t_{XHGX}	Output Data Hold after Clock Rising Edge	$2t_{CLCL} - 117$		ns
t_{XHDX}	Input Data Hold after Clock Rising Edge	0		ns
t_{XHDX}	Clock Rising Edge to Input Data Valid		$10t_{CLCL} - 133$	ns

Shift Register Mode Timing Waveforms



AC Testing Input/Output Waveforms⁽¹⁾

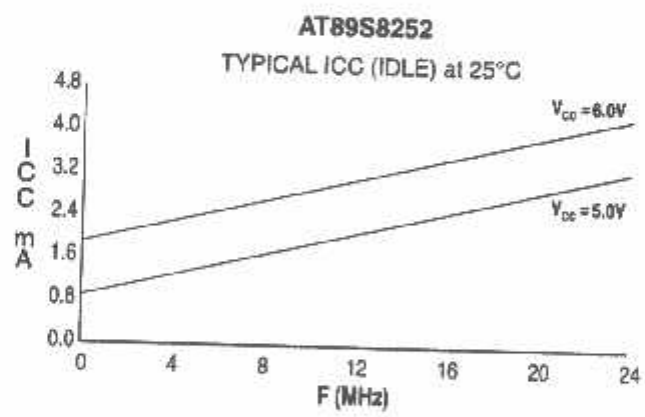
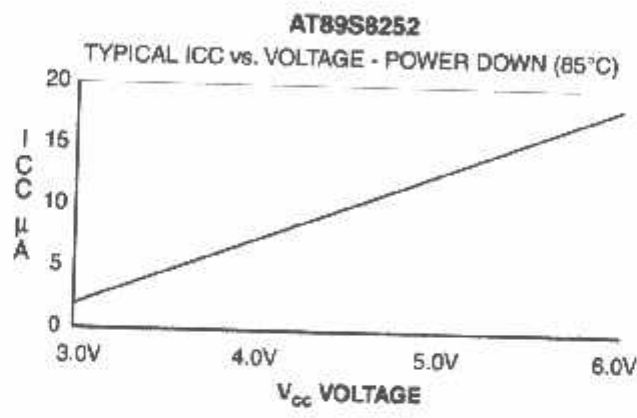
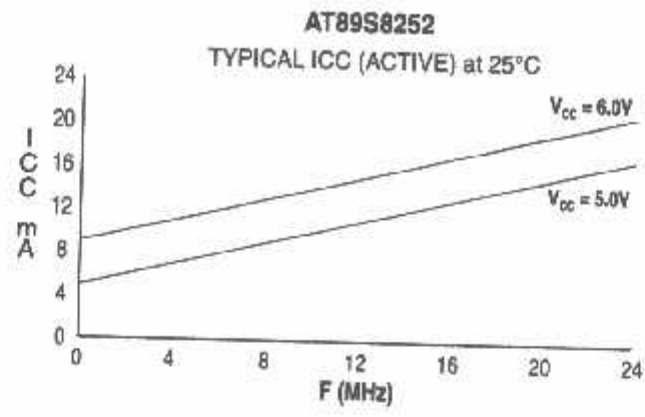


Notes: 1. AC Inputs during testing are driven at $V_{CC} - 0.5V$ for a logic 1 and $0.45V$ for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms⁽¹⁾



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



- Notes:
1. XTAL1 tied to GND for ICC (power-down)
 2. Lock bits programmed



Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 6.0V	AT89S8252-24AC	44A	Commercial (0°C to 70°C)
		AT89S8252-24JC	44J	
		AT89S8252-24PC	40P6	
		AT89S8252-24QC	44Q	
	4.0V to 6.0V	AT89S8252-24AI	44A	Industrial (-40°C to 85°C)
		AT89S8252-24JI	44J	
		AT89S8252-24PI	40P6	
		AT89S8252-24QI	44Q	
33	4.5V to 5.5V	AT89S8252-33AC	44A	Commercial (0°C to 70°C)
		AT89S8252-33JC	44J	
		AT89S8252-33PC	40P6	
		AT89S8252-33QC	44Q	

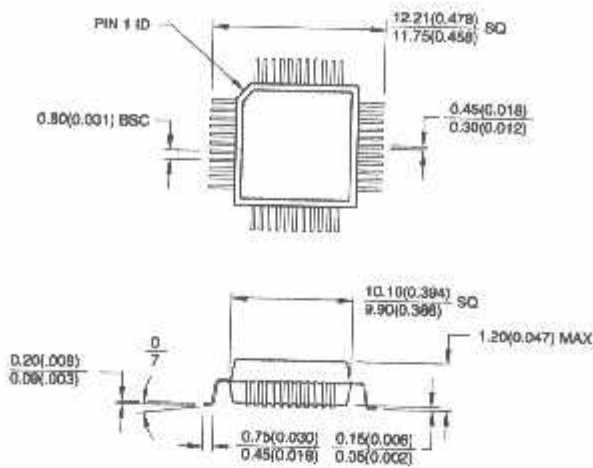
 = Preliminary Information

Package Type	
IA	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
IJ	44-lead, Plastic J-leaded Chip Carrier (PLCC)
IP6	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
IQ	44-lead, Plastic Gull Wing Quad Flatpack (PQFP)

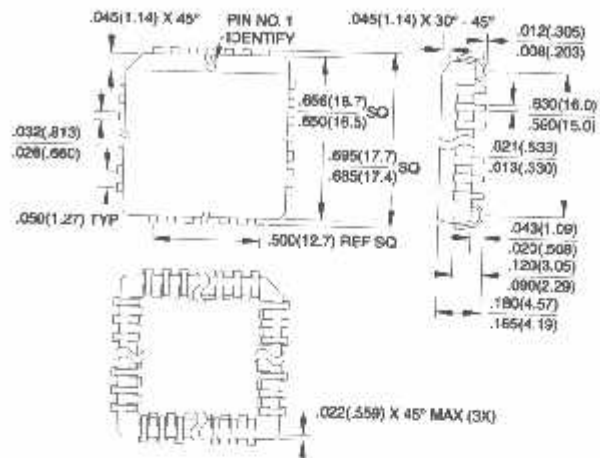
AT89S8252

Packaging Information

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flatpack (TQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-026 ACB

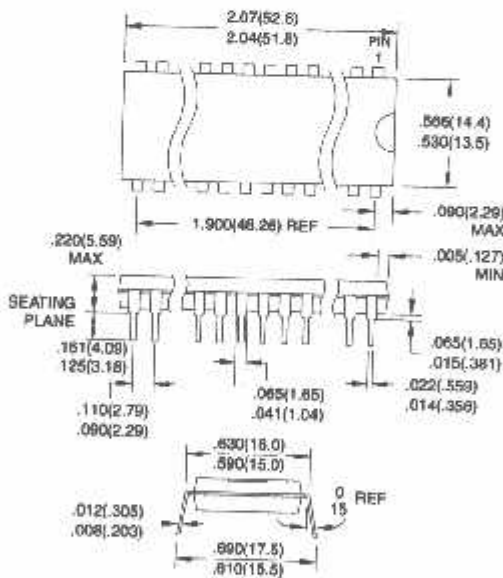


44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-018 AC

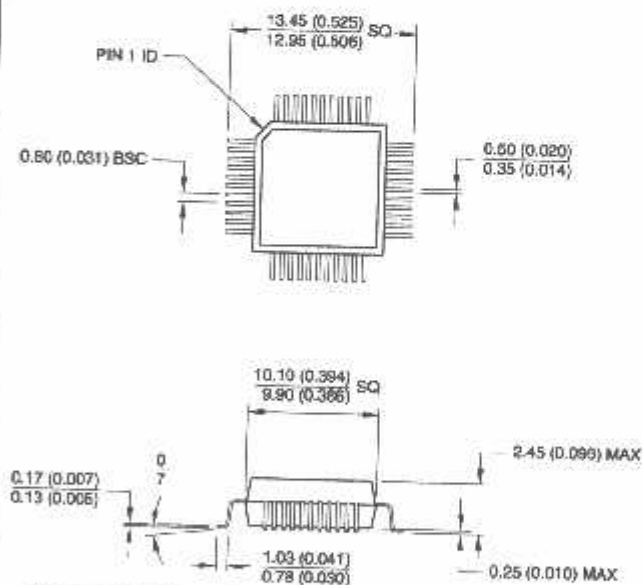


Controlling dimension: millimeters

40P6, 40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)



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



Controlling dimension: millimeters





Atmel Headquarters

Corporate Headquarters

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

Europe

Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia

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

Japan

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

Atmel Operations

Atmel Colorado Springs

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

Atmel Rousset

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

Fax-on-Demand

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

e-mail

literature@atmel.com

Web Site

<http://www.atmel.com>

BBS

1-(408) 436-4309

Atmel Corporation 2000.

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

marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

marks and product names in this document may be trademarks of others.

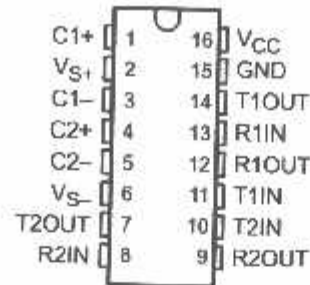


Printed on recycled paper.

0401E-02/00/xM

- Operates With Single 5-V Power Supply
- LinBiCMOS™ Process Technology
- Two Drivers and Two Receivers
- ± 30 -V Input Levels
- Low Supply Current . . . 8 mA Typical
- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- 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
- Package Options Include Plastic Small-Outline (D, DW, NS) Packages and Standard Plastic (N) DIPs

D, DW, N, OR NS PACKAGE
(TOP VIEW)



description

The MAX232 device 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.

The MAX232 is characterized for operation from 0°C to 70°C. The MAX232I is characterized for operation from -40°C to 85°C.

AVAILABLE OPTIONS

TA	PACKAGED DEVICES		
	SMALL OUTLINE (D, NS)	SMALL OUTLINE (DW)	PLASTIC DIP (N)
0°C to 70°C	MAX232D MAX232NS	MAX232DW	MAX232N
-40°C to 85°C	MAX232ID	MAX232IDW	MAX232IN

The D and DW packages are available taped and reeled by adding an R to the part number (i.e., MAX232DR). The NS package is only available taped and reeled.



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

LinASIC and LinBiCMOS are trademarks of Texas Instruments.

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



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2002, Texas Instruments Incorporated

MAX232, MAX232I
DUAL EIA-232 DRIVER/RECEIVER

SLLS047H – FEBRUARY 1989 – REVISED FEBRUARY 2002

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 range (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V _{OH}	High-level output voltage	T1OUT, T2OUT R _L = 3 kΩ to GND	5	7		V
		R1OUT, R2OUT I _{OH} = –1 mA	3.5			
V _{OL}	Low-level output voltage‡	T1OUT, T2OUT R _L = 3 kΩ to GND		–7	–5	V
		R1OUT, R2OUT I _{OL} = 3.2 mA			0.4	
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	0.3	1.2		V
V _{hys}	Input hysteresis voltage	R1IN, R2IN V _{CC} = 5 V	0.2	0.5	1	V
r _i	Receiver input resistance	R1IN, R2IN V _{CC} = 5, T _A = 25°C	3	5	7	kΩ
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
I _{CC}	Supply current	V _{CC} = 5.5 V, T _A = 25°C All outputs open,		8	10	mA

† 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.

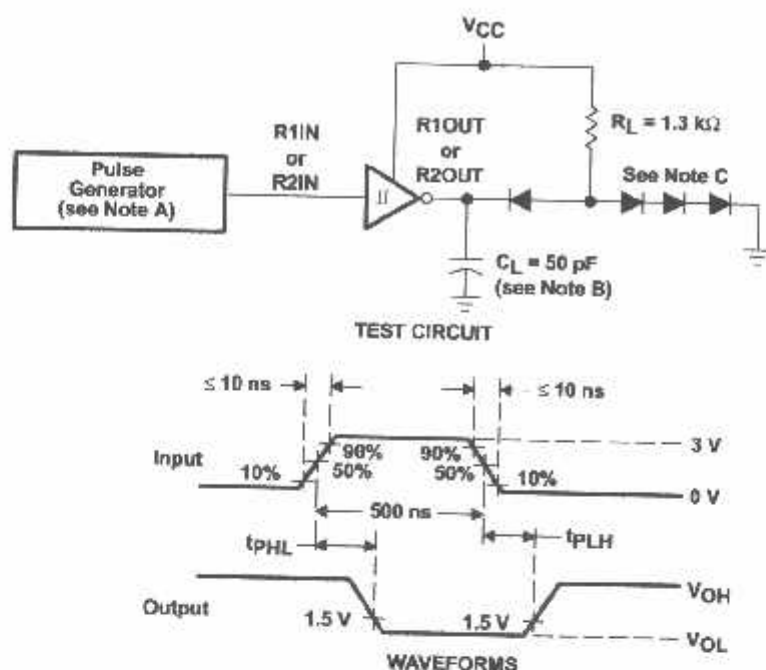
§ Not more than one output should be shorted at a time.

switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t _{PLH(R)}	Receiver propagation delay time, low- to high-level output	See Figure 1		500		ns
t _{PHL(R)}	Receiver propagation delay time, high- to low-level output	See Figure 1		500		ns
SR	Driver slew rate	R _L = 3 kΩ to 7 kΩ See Figure 2			30	V/μs
SR(tr)	Driver transition region slew rate	See Figure 3		3		V/μs



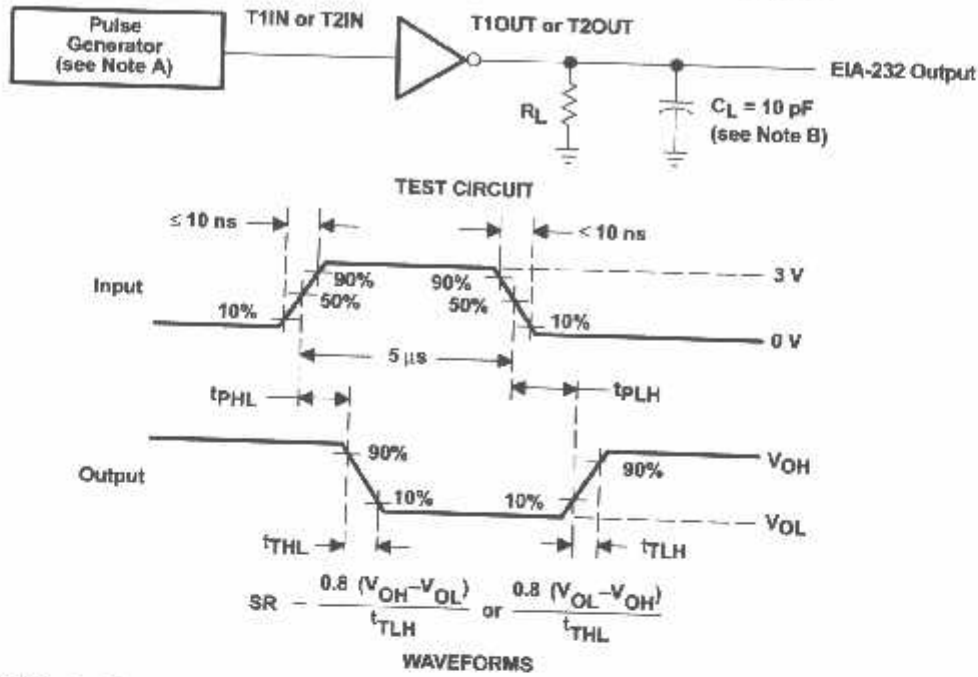
PARAMETER MEASUREMENT INFORMATION



- NOTES: A. The pulse generator has the following characteristics: $Z_0 = 50 \Omega$, duty cycle $\leq 50\%$.
B. C_L includes probe and jig capacitance.
C. All diodes are 1N3064 or equivalent.

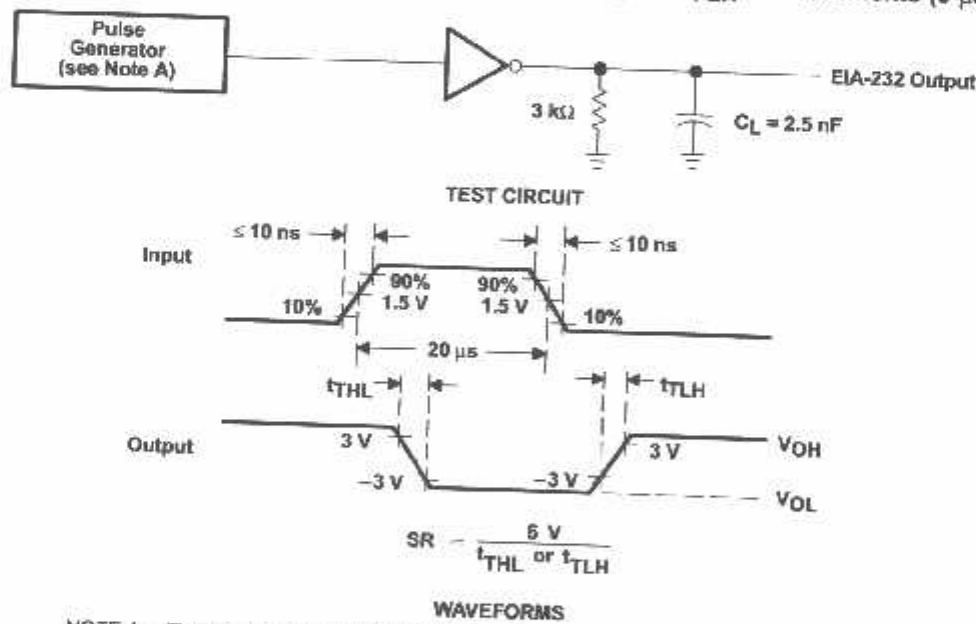
Figure 1. Receiver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements

PARAMETER MEASUREMENT INFORMATION



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

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



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

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