# SKRIPSI

## APLIKASI PENGIRIM DATA WEB ONLINE PENGUKUR KADAR POLUSI DI JALAN RAYA MENGGUNAKAN MIKROKONTROLLER AT89S52 DAN SMS GATE WAY


**Disusun Oleh :**

**NANANG SETIYA BUDI**

**NIM: 05. 12. 219**



**JURUSAN TEKNIK ELEKTRO S-1**

**KONSENTRASI TEKNIK ELEKTRONIKA**

**FAKULTAS TEKNOLOGI INDUSTRI**

**INSTITUT TEKNOLOGI NASIONAL MALANG**



**MARET 2010**

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

## LEMBAR PERSETUJUAN

# APLIKASI PENGIRIM DATA WEB ONLINE PENGUKUR KADAR POLUSI DIJALAN RAYA MENGGUNAKAN MIKROKONTROLLER AT89S52 DAN SMS GATE WAY

# SKRIPSI

*Disusun dan diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik Elektronika Strata Satu (S-1)*

**Disusun Oleh :**

**NANANG SETIYA BUDI**

**05.12.219**

**Diperiksa dan disetujui :**

MILIK PERPUSTAKAAN ITN MALANG

**Mengetahui :**

**Ketua Jurusan Teknik Elektro S-1**

**(Ir. F. Yudi Limpraptono, MT.)**

**NIP. Y. 103 950 0274**

**Dosen Pembimbing,**

**( Joseph Dedy Irawan, ST. MT )**

**NIP. 132315178**

**JURUSAN TEKNIK ELEKTRO S-1**

**KONSENTRASI TEKNIK ELEKTRONIKA**

**FAKULTAS TEKNOLOGI INDUSTRI**

**INSTITUT TEKNOLOGI NASIONAL MALANG**

**MARET 2010**

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO
JL.Raya Karanglo Km 2
MALANG

**BERITA ACARA UJIAN SKRIPSI**
**FAKULTAS TEKNOLOGI INDUSTRI**

| | |
|---|---|
| Nama | : NANANG SETIYA BUDI |
| Nim | : 05.12.219 |
| Jurusan | : Teknik Elektro |
| Kosentrasi | : Teknik Elektronika S-1 |
| Masa Bimbingan | : 01 Desember 2009 s/d 01 Juni 2010 |
| Judul Skripsi | : Aplikasi pengirim data web online pengukur kadar polusi di jalan raya menggunakan mikrokontroller AT89S52 dan SMS gate way. |

Dipertahankan dihadapan Team Penguji Skripsi Jenjang Strata Satu ( S–1 ) pada :

| | |
|---|---|
| Hari | : Selasa |
| Tanggal | : 09 Februari 2010 |
| Nilai | : 84,6 (A) |

**Panitia Majelis Penguji :**

Ketua

(Ir. Sidik Noertjahjono, MT)
NIP. Y. 102 870 0163

Sekretaris

(Ir. F. Yudi Limpraptono, MT)
NIP. Y. 103 950 0274

**Anggota penguji :**

Penguji pertama

(Ir. F. Yudi Limpraptono, MT)
NIP. Y. 103 950 0274

Penguji kedua

(Ir. TH. Mimien Mustikawati, MT)
NIP. P. 103 000 0352

# LEMBAR PERSEMBAHAN

## Gelar sarjana ini khusus Ku persembahkan kepada :

*Abah dan Ibu H. Moh. Mukmin dan Hj. Siti Rokhaniyah, Kakakku berdua Nur Wahyuniah dan Nur Wahyu Dewi Kusrini sekeluarga, kedua ponakanku Ana dan Akbar* yang tersayang, terimakasih kuucapkan atas semua kasih sayang, doa, perhatian, dan nasehat serta pengorbadan dukungan moril maupun spritual yang tak henti-tentiNya tercurahkan kepada Ku. Begitu besar jasa yang engkau berikan sehingga tak mungking tuk bisa sepenuh membalasnya.

Yang ku sayang Fitri Kristinah yang selalu memberikan semangat dan peringatan serta ocehan-ocehan sehingga dapat memperoleh gelar ini

wisuda periode oktober 2009; Fitri kristina,Ssi

# Ucapkan Terima kasih Ku Kepada

Keluarga besar KSR PMI Unit ITN Malng yang telah memberikan banyak pelajaan dan pengalaman sehingga dapat menjadikan aku yang bukan apa-apa menjadi orang yang berguna dan pekerja keras. Semoga KSR PMI Unit ITN Malang selalu menciptakan generasi - generasi yang peduli terhadap sesama.

Buat Pembimbing Skripsiku bapak Joseph Dedi Irawan, ST. MT yang dengan sabar dan telaten membimbingku menyelesaikan laporan skripsi ini. Dan terima kasih atas bantuannya ketika seminar hasil kemarin.

Buat dosen - dosen ITN, Bu Siswi Astuti, Pak Taufik, Pak Aryuanto, Pak Yudi, Pak Yusuf, Pak Komang, Pak Sidik, Pak Ibrahim, Pak Sumanto serta dosen - dosen semuanya.

Mas Yudi sawajajar, yang telah membantu terselesainya hardware dan software sehinnga gelar ini dapat aku raih. Semoga Mas sekeluarga selalu sehat selalu dan dilimpahkan rahmat oleh-Nya.

Buat arek - arek asistan laboratorium kendali industry, Guruh, Andry, Heri, Viki, Rini, Reni, Darwis dan arek baru yang tidak tau namanya. Semoga kalian semua mendapatkan ilmu yang bermanfaat.

Buat arek kos SUTERDAM ( Sunan Ampel Terus Kedalam), Rony, Ditok, zain, Riski, Dika, Enong, Wahyu, kuriman, Deni, Bul - Bul, bapak dan ibu bandi dan arek - arek yang lainnya. Terima kasih atas dukungannya. Kalau ada waktu aku pasti akan main - main ke SUTERDAM.

Buat *Arek - arek Elektro S-1 angkatan 2005 ( Ponda, Elsa, Pakde Pras, Tompez bayu, Zona, Gopal, agung, Nasrudin, Ari merauke, Suryo, Rofi, Rizal Bli, Yuda, Arik Mario Band, Fauzi Gendut, Miki, Drajat, Braman, Heru PKN, semuanya ).*

Yang telah memberikan dukungan dan ejekan serta pengalaman jalan - jalan di Malang Raya. Kapan - kapan ayo kumpul dan reoni serta menapak tilas perjalanan kita selama menjadi mahasiswa ITN. Makasih atas kekompakan dan dukungan selama ini dalam memperjuangkan sarjana S-1, semoga persabatan kita selalu terjaga.

Perjalanan ini bukanlah yang terakhir
Perjalanan baru dimulai dengan babak baru
Terjang dan hadapi semua lika-liku kehidupan
Dengan senyum dan semangat

*Malang, 02 maret 2010*

NANANG SETIYA BUDI

# ABSTRAKSI


## APLIKASI WEB ONLINE PENGUKUR KADAR POLUSI DI JALAN RAYA MENGGUNAKAN MIKROKONTROLLER AT89S52 DAN SMS GATE WAY

**(Nanang Setiya Budi, 05.12.219**, Jurusan Teknik ElektroS-1/Elektronika)
(Dosen Pembimbing : Joseph Dedy Irawan, ST. MT)

Dari tahun ke tahun tingkat polusi udara semakin meningkat tajam dan semakin membahayakan kesehatan makhluk hidup terutama kesehatan manusia, ini ditandai dengan semakin panasnya suhu bumi sekarang dan mencairnya es dikutub utara dan selatan (*global warming).* Sehingga diperlukan alat pemantau kadar polusi agar masyarakat bisa mengetahui dan menghentikan kebiasaan menciptakan polusi udara.

Garis besar prinsip kerjanya adalah kadar polusi gas karbon dioksida dideteksi oleh sensor gas karbon dioksida yaitu sensor TGS 4161 dan otputan sensor tersebut akan dikondisikan sinyalnya oleh op-amp sehinggan stabil sinyalnya. Karena sinyal yang dihasilkan oleh sensor TGS 4161 masih berupa sinyal analog maka akan dikonversikan menjadi sinyal digital oleh ADC, ADC yang kami pakai adalah ADC PCF 8591. Setelah sinyal berubah menjadi sinyal digital maka akan diolah/diproses oleh mikrokontroller AT89S52 dan akan ditampilkan ke display dot matriks. Untuk jam dan tanggal digital ditampilkan oleh display seven segmen dari hasil proses mikrokontroller AT89S52 dan RTC DS 1307. Alat ini juga sebagai pengirim data tentang kadar polusi udara ke web online melalui komunikasi AT comment dengan SMS gate way. Sehingga kadar polusi bisa ditampilkan ke display dialat tersebut dan web dan semua orang dapat melihatnya di web.

Alat ini menggunakan mikrokontroller AT89S52 sebagai basisnya dan perangkat pendukungnya meliputi : Sensor gas karbon dioksida (TGS 4161), Rangkaian ADC, RTC sebagai informasi jam/tanggal dan display seven segmen & dot matriks. Serta menggunakan handphone sebagai komunikasi dengan web.



**Kata kunci : mikrokontroller AT89S52, TGS 4161 dan RTC serta karbon dioksida**

# KATA PENGANTAR

Segala puji dan syukur kami panjatkan kehadirat Allah SWT atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan skripsi ini dengan judul, ***aplikasi pengirim data web online pengukur kadar polusi di jalan raya menggunakan mikrokontroller AT89S52 dan SMS gate way.***

Skripsi ini merupakan salah satu syarat yang harus ditempuh dalam menyelesaikan program studi (S-1) Jurusan Teknik Elektro Konsentrasi Elektronika di Institut Teknologi Nasional Malang. Dalam penyusunan skripsi ini penulis tidak lepas dari bantuan dari berbagai banyak pihak, oleh karena itu pada kesempatan ini penulis ingin mengucapkan banyak terima kasih kepada:

1. Bapak Prof. Dr. Ir. Eng. Abraham Lomi, MSEE selaku Rektor ITN Malang.
2. Bapak Ir. Sidik Noertjahjono, MT selaku Dekan Fakultas Teknologi Industri ITN Malang.
3. Bapak Ir. F Yudi Limpratono,MT selaku ketua jurusan Teknik Elektro S-1.
4. Bapak Joseph Dedy Irawan, ST. MT selaku dosen Pembimbing atas bantuan, arahan dan bimbingan dalam penyusunan skripsi ini.
5. Bapak dan Ibu dosen Teknik Elektro yang telah banyak memberi saran dan kritik.
6. Abah, ibu dan mbak - mbakku yang selalu berdoa dan memberi dukungan moral dalam penyusunan skripsi ini
7. Teman – teman seangkatan jurusan Tektnik Elektro S-1 angkatan 2005 yang senantiasa memberi motivasi serta semangat.
8. Rekan – rekan mahasiswa dan semuanya yang telah membantu untuk sampai terselesaikannya skripsi ini.

Penyusun menyadari bahwa skripsi ini masih terdapat kekurangan. Oleh karena itu, penyusun mengharapkan saran dan kritik yang bersifat membangun dari pembaca. dan semoga skripsi ini bermanfaat bagi mahasiswa Institut Teknologi Nasional Malang.

Malang, Maret 2010

Penyusun

## DAFTAR ISI

Halaman

**LEMBAR PERSETUJUAN** ..... **I**
**ABSTRAK** ..... **ii**
**KATA PENGANTAR** ..... **iv**
**DAFTAR ISI** ..... **vi**
**DAFTAR TABEL** ..... **xi**
**DAFTAR GAMBAR** ..... **xiii**
**DAFTAR GRAFIK** ..... **xiv**

**BAB I PENDAHULUAN**

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

**BAB II LANDASAN TEORI**

2.1. Gas beracun ..... 6
2.1.1 Gas karbon diolsida (CO2) ..... 6
2.2. Mikrokontroler AT 89S52 ..... 11
2.2.1. Fitur mikrokontroler AT 89S52 ..... 11
2.2.2. Organisasi Memori ..... 16
2.2.3. Special Fungsi Register ..... 16
2.4.3.1. Accumulator ..... 17
2.4.3.2. Register B ..... 17
2.4.3.3. Register Port ..... 17

2.4.3.4. Register Timer ..... 17

2.4.3.5. Register Control ..... 18

2.4.3.6. Program Status Word ..... 20

2.4.3.7. Stack Pointer ..... 21

2.4.3.8. Data Pointer ..... 21

2.4.3.9. Serial Data Buffer ..... 21

2.4.4. Mode Pengalamatan ..... 21

2.4.5. Set Intruksi ..... 23

2.3 Sensor Gas Karbon Dioksida *( TGS 4161 )* ..... 26

2.4 Display 7 segmen ..... 28

2.5 Display Dot Matriks ..... 30

2.5.1 Dekode matriks LED ..... 30

2.6 Shift register 8 bit IC 74HCT164 ..... 31

2.7 RTC (Real Time Clock) DS1307 ..... 33

2.7.1. Fungsi kaki-kaki (pin) RTC DS1307 ..... 34

2.7.2. Cara kerja RTC DS1307 ..... 34

2.8. Pengkondisi sinyal ..... 35

2.8.1. Penguat non inverting ..... 37

2.8.2. Penguat inverting ..... 37

2.8.3. Penguat differensial ..... 38

2.8.4 Penguat instrumentasi ..... 39

2.8.5. Integrator ..... 39

2.8.6. Differensiator ..... 40

2.9. Transistor ..... 41

2.10. Dioda. ..... 41

2.11. ADC PCF 8591..... 43

2.12. Hp Siemens..... 44

2.12.1. Pin out Hp Siemens..... 44

2.12.2. Kabel data ..... 45

2.12.3. SMS ..... 46

2.12.4. PDU ..... 46

**BAB III PERENCANAAN DAN PEMBUATAN ALAT**

3.1. Perancangan sistem ..... 49

3.2. Prinsip kerja ..... 51

3.3. Perencanaan sensor ..... 51

3.3.1. Sensor gas karbon dioksida (TGS 4161)..... 51

3.4. Perancangan rangkaian ADC PCF 8591 ..... 52

3.5. Perancangan rangkaian mikrokontroller AT89S52..... 54

3.6. Perancangan rangkaian RTC DS 1307..... 55

3.7. Perancangan rangkaian komunikasi serial handpone..... 56

3.8. Perancangan rangkaian 7-segmen..... 56

3.9. Perancangan rangkaian dot matriks ..... 57

3.10. flowchart ..... 59

**BAB IV PENGUJIAN ALAT**

4.1. Tujuan ..... 60

4.2. Pengujian perangkat keras ..... 60

4.2.1. Pengujian sensor TGS 4161 ..... 61

2.4.3.2 Tujuan . ..... 61

2.4.3.2 Prosedur pengujian..... 61

2.4.3.3 Hasil pengujian ..... 67
4.2.2. Pengujian ADC PCF 8591 ..... 68
4.2.2.1 Tujuan ..... 68
4.2.2.2 Alat-alat Yang Digunakan ..... 68
4.2.2.3. Prosedur Pengujian ..... 68
4.2.2.4. Hasil Pengujian ..... 69
4.2.3. Pengujian RTC DS 1307 ..... 70
4.2.3.1. Tujuan ..... 70
4.2.3.2 Alat-alat Yang Digunakan ..... 70
4.2.3.3. Langkah Pengujian ..... 70
4.2.3.4. Hasil Pengujian ..... 71
4.2.4. Pengujian handphone ..... 71
4.2.4.1. Tujuan ..... 71
4.2.4.2. Alat-alat Yang Digunakan ..... 71
4.2.4.3 Langkah Pengujian ..... 72
4.2.4.4. Hasil Pengujian ..... 72
4.2.5. Pengujian 7-segmen ..... 74
4.2.5.1. Tujuan ..... 74
4.2.5.2. Alat-alat Yang Digunakan ..... 74
4.2.5.3. Langkah Pengujian ..... 74
4.2.5.4. Hasil Pengujian ..... 74
4.2.6. Pengujian system secara keseluruhan ..... 75
4.2.6.1. Hasil pengujian ..... 76

**BAB V PENUTUP**

5.1. Kesimpulan ..... 77

5.2. Saran..... 78

**DAFTAR PUSTAKA**..... **79**

**LAMPIRAN**.....

## DAFTAR TABEL

Halaman


Tabel 2-1. Fungsi khusus port 1 pada AT89S52 ..... 14

Tabel 2-2. Fungsi khusus port 3 pada AT89S52 ..... 15

Tabel 2-3. Register IP pada mikrokontroler AT89S52 ..... 18

Tabel 2-4. Register IE pada mikrokontroler AT89S52 ..... 19

Tabel 2-5. Display 7-segment ..... 29

Tabel 2-6. Pin-pin IC 74HCT164 ..... 32

Tabel 2-7. Pinout hp siemens ..... 45

Tabel 2-8. Kabel data siemens ..... 48

Tabel 3-1. Hubungan antara ADC PCF 8591 dengan AT89S52 ..... 53

Tabel 4-1. Data hasil perhitungan ΔEMF ..... 62

Tabel 4-2. Data hasil perhitungan tegangan keluaran sensor yang telah dikuatkan oleh op-amp ..... 64

Tabel 4-3. Data hasil pengukuran tegangan keluaran sensor yang telah dikuatkan oleh op-amp ..... 66

Table 4-4. Perbandingan data hasil perhitungan dan pengukuran tegangan keluaran sensor yang telah dikuatkan oleh op-amp ..... 67

Tabel 4-5. Hasil pembacaan dan perhitungan ADC PCF 8591 ..... 69

Tabel 4-6. Hasil pengujian waktu pada jam digital dengan waktu pada alat ..... 71

Tabel 4-7. Hasil pengujian tanggal pada jam digital dengan tanggal pada alat .... 71

Tabel 4-8. Data hasil pengujian SMS gate way ..... 73

Tabel 4-9. Hasil pengujian waktu pada jam digital dengan waktu pada alat ..... 74

Tabel 4-10. Hasil pengujian tanggal pada jam digital dengan tanggal pada alat .... 75

Tabel 4-11. Hasil pengujian keseluruahan alat ..... 76

## DAFTAR GAMBAR

Halaman


Gambar 2-1. Diagram blok mikrokontroller AT89S52 ..... 12

Gambar 2-2. Mikrokontroller Atmel AT89S52 ..... 13

Gambar 2-3. Rangkaian osilator ..... 15

Gambar 2-4. Struktur memori program dan data ..... 16

Gambar 2-5. Register IP pada mikrokontroler AT89S52 ..... 18

Gambar 2-6. Register IE pada mikrokontroler AT89S52 ..... 19

Gambar 2-7. Register PSW dalam mikrokontroler AT89S52 ..... 20

Gambar 2-8. Sensor TGS 4161/sensor CO2 ..... 27

Gambar 2-9. Basic measuring circuit ..... 28

Gambar 2-10. Display 7-segmen ..... 29

Gambar 2-11. Display dot matriks 5x7 titik ..... 30

Gambar 2-12. Dekode LED matriks dalam tampilan karakter ..... 31

Gambar 2-13. IC 74HCT164 ..... 32

Gambar 2-14. Pin-pin RTC DS1307 ..... 34

Gambar 2-15. Simbol Op-amp ..... 35

Gambar 2.16. Rangkaian penguat non inverting ..... 37

Gambar 2-17. Rangkaian penguat inverting ..... 38

Gambar 2-18. Rangkaian penguat diferensial ..... 38

Gambar 2-19. Rangkaian penguat instrumentasi ..... 39

Gambar 2-20. Rangkaian integrator ..... 40

Gambar 2-21. Rangkaian diferensiator ..... 40

Gambar 2-22. Transistor ..... 41

Gambar 2-23. Dioda ..... 42

Gambar 2-24. Kurva dioda ..... 42
Gambar 2-25. Pin-pin PCF 8591 ..... 43
Gambar 3-1. Blok diagram perencanaan ..... 49
Gambar 3-2. Rangkaian skematik TGS4161 ..... 51
Gambar 3-3. Rangkaian skematik PCF 8591 ..... 53
Gambar 3-3. Rangkaian skematik mikrokontroller AT89S52 ..... 54
Gambar 3-4. Rangkaian skematik RTC DS 1307 ..... 55
Gambar 3-5. Rangkaian komunikasi hp dengan AT89S52 ..... 56
Gambar 3-6. Rangkaian skematik 7-segmen ..... 57
Gambar 3-7. Rangkaian skematik dot matriks ..... 58
Gambar 4-1. Hasil pengukuran tegangan keluaran sensor ..... 65
Gambar 4-2. Hasil pengukuran tegangan keluaran sensor ..... 65
Gambar 4-3. Setting jam digital ..... 72
Gambar 4-4. SMS meminta data sensor TGS 4161 ..... 72
Gambar 4-5. Pesan balasan dari alat ..... 73

## DAFTAR GRAFIK

Halaman

Grafik 2-1. Sensitivitu characteristics TGS 4161 ..... 27

Grafik 4-1. Hasil perhitungan antara PPM dan Vout..... 64

Grafik 4-2 hasil pengukuran antara PPM dan Vout..... 66

# BAB I

# PENDAHULUAN

## 1.1 Latar Belakang Permasalahan

Masalah pencemaran lingkungan bukanlah hal yang baru lagi dan telah menjadi isu dimana-mana. Sepintas masalah lingkungan sederhana saja, seperti sampah, limba industry, pencemaran udara, dan sebagainya. Akan tetapi bila ditinjau lebih dalam permasalahannya tidak sesederhana itu. Permasalahan lingkungan seperti bagian dari pandangan hidup, sebuah reaksi krisis atas pemujaan tehnologi, pengurasan energi, sikap hidup komsutif dan usaha mengejar pertumbuhan di atas segala hal (*Eksekutif*, Nomor 01 Edisi Maret 1986 dalam Syamsul 2000:1). Hal ini ditunjang kenyataan bahwa himbauan ini pertama kali terdengar dinegara-negara maju, yang sudah lebih dulu mengalami pahit getirnya teknologi tinggi, revolusi industri maupun urbanisasi.

Kebisingan jutaan mobil, mesin-mesin industri yang membakar minyak dan mengeluarkan $CO_2$, hutan terbakar melepas $CO_2$, pembusukan limbah pertanian juga memproduksi gas karbon dioksida, dan pemukiman-pemukiman orang modern banyak menyerap energi alam juga merupakan penyumbang pencemaran lingkungan. Masalah ini menambah panjang daftar permasalahan alam yang sudah muncul seperti fenomena "bumi makin panas". Ini muncul akibat penambahan gas-gas atmosfer, terutama $CO_2$ yang menyerap panas matahari meskipun secara tidak langsung.

Dimalang kendaraan bermotor merupakan salah satu factor terbesar yang menyumbangkan banyak polusi udara dikota tersebut, apalagi malang dikenal sebagai kota pelajar maka semakin banyak orang yang datang ke malang untuk

menimbah ilmu maka sumbangan polusi udara oleh kendaraan bermotor yang mereka bawa ke malang semakin besar pula.

Kendaraan bermotor sangat mempengaruhi masalah polusi udara. Setiap kndaraan bermotor mengeluarkan karbon dioksida, yang sifatnya sangat merugikan bagi kesehatan manusia. Semakain banyak kadar karbon dioksida dalam tubuh maka akan mempengaruhi kesehatan manusia, sehingga otak akan kekurangan oksigen yang membuat fungsi otak akan terganggu. Dan, apabila otak tidak berfungsi seperti yang semestinya maka organ-organ yang lain juga akan terganngu kinerjanya sehingga tubuh akan menjadi kurang sehat.

Oleh karena bahayanya masalah diatas, maka kita akan membuat sebuah alat yang dapat memantau berapa kadar polusi udara. Khususnya, karbon dioksida yang ada dijalan raya sehingga orang dapat mengetahui seberapa besar kadar polusi karbon dioksida yang ada disekitar kita.

## 1.2 Rumusan Masalah Penelitian

Berdasarkan latar belakang tersebut, dapat dirumuskan rumusan masalah sebangai berikut :

1. Bagaimana merancang dan membuat identifikasi karbon dioksida dengan sensor TGS 4161 menggunakan mikrokontroler AT89S52.
2. Bagaimana cara menampilkan hasil identifikasi karbon dioksida ke display.
3. Bagaimana agar mikrokontroler AT89S52 dapat mengirimkan SMS agar gapat ditampilkan pada web.
4. Bagaimana cara menampilkam jam digital menggunakan mikrokontroler AT89S52 ke display.

## 1.3 Tujuan

Adapun tujuan yang ingin dicapai dalam perancancangan dan pembuatan alat ini sebagai berikut :

1. Memberikan informasi kepada masyarakat tentang kadar polusi yang terkandung dalam udara.
2. Memberikan informasi kepada masyarakat pengguna jalan raya tentang waktu dan tanggal.
3. Memberikan data ke web tentang kadar polusi sebagai data base melalui handphone.

## 1.4 Batasan Masalah.

Untuk mencapai tujuan penyelesaian skripsi ini secara maksimal, maka diperlukan batasan masalah yang diharapkan agar permasalahan tidak meluas dan tetap terfokus pada tujuan utama. Adapun batasan-batasan masalah pada skripsi ini yaitu :

1. Membahas sampai pengiriman data pada web online.
2. Menggunakan mikrokontroller AT98S52.
3. Menggunakan dot matrik dan seven segmen.
4. Menggunakan sensor karbon dioksida TGS 4161.
5. Handphone siemen C35.
6. Hanya membahas polusi udara gas karbon dioksida.

## 1.5 Metodologi pembahasan.

Untuk dapat menyelesaikan skripsi ini, maka dapat mengambil metode-metode untuk menyelesaikan pemecahan masalah pada skripi ini, meliputi:

a. Studi literatur

Mempelajari teori-teori yang terkait melalui literatur yang telah ada, yang berhubungan dengan pembahasan masalah.

b. Perencanaan dan pembuatan alat

Membuat digram blok rangkaian yang sesuai dengan rencana kerja, yang kemudian direalisasikan dengan masalah perencanaan dan pembuatan berdasarkan diagram blok rangkaian yang telah disusun.

c. Studi analisa alat

Dimaksudkan untuk melakukan analisa dan pengujian alat yang telah dirancang apakah sesuai antara fungsi dengan kerja yang diharapkan.

d. Pengambilan Kesimpulan

Dilakukan setelah mendapatkan hasil dari perancangan dan pengujian alat. Jika hasil yang diperoleh telah sesuai dengan sfesifikasi yang ditentukan saat dilakukan perancangan, berarti alat tersebut telah dianggap selesai dan sesuai dengan harapan.

e. Penyusunan buku laporan

Bertujuan untuk menyusun data laporan yang berpedoman pada alat yang telah selesai dibuat beserta kesimpulan dan cara kerja alat.

## 1.6 Sistematika

Pembahasan dalam Skripsi ini akan diuraikan dalam lima bab, yang penjabarannya adalah sebagai berikut :

**Bab I : PENDAHULUAN**

Membahas tentang latar belakang, rumusan masalah, tujuan, metodologi pembahasan dan sistematika pembahasan yang akan dipaparkan dalam skripsi ini.

**Bab II : LANDASAN TEORI**

Membahas tentang berbagai macam teori yang mendukung dalam menjalankan waktu digital dan sensor gas karbon dioksida dapat bekerja menggunakan AT89LS52 sehingga dapat ditampilkan pada display serta dikirimkan ke web server melalui sms.

**Bab III : PERENCANAAN DAN PEMBUATAN ALAT**

Membahas tentang proses kerja sensor gas karbon dioksida dan waktu digital menggunakan mikrokontroler AT89LS52 serta dapat mengirimkan data ke web server melalui sms

**Bab IV : PENGUJIAN SISTEM**

Membahas tentang pengujian waktu digital, sensor gas karbon dioksida apakah berjalan dengan menggunakan mikrokontroler AT89LS52 dan dapat ditampilkan ke display yaitu dapat menampilkan kondisi kadar polusi rendah, sedang atau, tinggi sesuai dengan kondisi udara yang ada. Serta menguji apakah sms yang dikirimkan ke handphone bisa diterima oleh PC sehingga dapat dikirimkan ke web server yang telah ditentukan.

**Bab V : PENUTUP**

Merupakan bagian akhir dari laporan skripsi yang terdiri dari kesimpulan dan saran.

# BAB II

# LANDASAN TEORI

Landasan teori sangat membantu untuk dapat memahami suatu sistem. Disamping itu dapat juga dijadikan acuan didalam merencanakan suatu sistem. Dengan pertimbangan hal-hal tersebut, maka landasan teori merupakan bagian yang harus dipahami untuk pembahasan selanjutnya.

Pada bab ini akan diuraikan mengenai penggunaan teknologi mikrokontroler AT89S52 beserta piranti penunjang lainya. Dimana teknologi AT89S52 sendiri merupakan pengembangan teknologi mikroprosesor dan mikrokontroler sebelumnya. Disini akan diuraikan mengenai segala aspek komponen perangkat keras yang akan dibuat.

## 2.1 Gas Beracun

### 2.1.1 Gas Karbon Dioksida *($CO_2$)*

*Karbon dioksida* atau zat asam arang adalah sejenis senyawa kimia yang terdiri dari dua atom *oksigen* yang teikat secara *kovalen* dengan sebuah atom *karbon*. *Karbon dioksida* dihasilkan oleh semua heawan, tumbuh-tumbuhan, fungi dan *mikroorganisme* pada *proses respirasi* dan digunakan pada tumbuhan pada *proses fotosintesis*.

Pada abad ke-17, seorang kimiawan *Fleming, Jan Baptist Van Helmont*, menemukan bahwa arang yang dibakar pada bejana tertutup akan menghasilkan abu yang massanya daripada massa arang semula. Dia berkesimpulan bahwa sebagian arang tersebut telah ditransmutasikan menjadi zat yang tidak terlihat, yang kemudian dinamakan zat tersebut sebagai "gas" atau *spiritus sylvestre* ( Bahasa Indonesia: arwah liar )

Sifat-sifat *karbon dioksida* dipelajari lebih lanjut pada tahun 1750 oleh fisikawan Skotlandia *Joseph Black*. Dia menemukan bahwa batu kapur (kalsium karbonat) dapat dibakar atau diberikan asam dan menghasilkan gas yang dia namakan sebagai *"fixed air"*. Dia juga menemukan bahwa gas ini lebih berat daripada udara dan ketika digelembungkan dalam *larutan kapur (kalsium hidroksida)* akan mengendapkan *kalsium karbonat*. Dia menggunakan fenomena ini untuk mengilustrasikan bahwa *karbon dioksida* dihasilkan dari pernafasan hewan dan *fermentasi mikrob*. Pada tahun 1772, seorang kimiawan Inggris *Joseph Priestley* mempublikasikan sebuah jurnal yang berjudul *Impregnating Water with Fixed Air*. Dalam jurnal tersebut, dia menjelaskan proses penetesan *asam sulfat* atau *minyak vitriol* seperti yang Priestley sebut ke kapur untuk menghasilkan *karbon dioksida* dan memaksa gas itu untuk larut dengan menggoncangkan semangkuk air yang berkontak dengan gas.

*Karbon dioksida* secara garis besar dihasilkan dari enam proses:

1. Sebagai hasil samping dari pengilangan *ammonia* dan *hidrogen*, di mana *metana* dikonversikan menjadi $CO_2$.
2. Dari pembakaran kayu dan bahan bakar fosil;
3. Sebagai hasil samping dari *fermentasi* gula pada proses peragian bir, wiski, dan minuman beralkohol lainnya.
4. Dari proses penguraian termal batu kapur, $CaCO_3$
5. Sebagai produk samping dari pembuatan *natrium fosfat*
6. Secara langsung di ambil dari mata air yang karbon dioksidanya dihasilkan dari pengasaman air pada batu kapur atau *dolomit*.

Kandungan *karbon dioksida* di udara segar bervariasi antara *0,03% (300ppm)* sampai dengan *0,06% (600 ppm)* bergantung pada lokasi.

Menurut *Otoritas Keselamatan Maritim Australia*, "Paparan berkepanjangan terhadap *konsentrasi karbon dioksida* yang sedang dapat menyebabkan *asidosis* dan efek-efek merugikan pada *metabolisme* kalsium *fosforus* yang menyebabkan peningkatan endapan kalsium pada jaringan lunak. *Karbon dioksida* beracun kepada jantung dan menyebabkan menurunnya gaya *kontraktil*. Pada konsentrasi tiga persen berdasarkan volume di udara, ia bersifat *narkotik* ringan dan menyebabkan peningkatan tekanan darah dan denyut nadi, dan menyebabkan penurunan daya dengar. Pada konsentrasi sekitar lima persen berdasarkan volume, ia menyebabkan *stimulasi* pusat pernafasan, pusing-pusing, kebingungan, dan kesulitan pernafasan yang diikuti sakit kepala dan sesak nafas. Pada konsentrasi delapan persen, ia menyebabkan sakit kepala, keringatan, penglihatan buram, tremor, dan kehilangan kesadaran setelah paparan selama lima sampai sepuluh menit."

Oleh karena bahaya kesehatan yang diasosiasikan dengan paparan karbon dioksida, *Administrasi Kesehatan dan Keselamatan Kerja Amerika Serikat* menyatakan bahwa paparan rata-rata untuk orang dewasa yang sehat selama waktu kerja 8 jam sehari tidak boleh melebihi *5.000 ppm (0,5%)*. Batas aman maksimum untuk balita, anak-anak, orang tua, dan individu dengan masalah kesehatan *kardiopulmonari* (jatung dan paru-paru) secara signifikan lebih kecil. Untuk paparan dalam jangka waktu pendek (di bawah 10 menit), batasan dari *Institut Nasional untuk Kesehatan dan Keamanan Kerja Amerika Serikat (NIOSH)* adalah *30.000 ppm (3%)*. NIOSH juga menyatakan bahwa konsentrasi *karbon dioksida* yang melebihi 4% adalah langsung berbahaya bagi keselamatan jiwa dan kesehatan.

Adaptasi terhadap peningkatan kadar *karbon dioksida* dapat terjadi pada manusia. *Inhalasi karbon dioksida* yang berkelanjutan dapat ditoleransi pada konsentrasi inspirasi tiga persen paling sedikit selama satu bulan dan empat persen konsentrasi insiparsi selama lebih dari satu minggu. Diajukan juga bahwa konsentrasi insipirasi sebesar 2,0 persen dapat digunakan untuk ruangan tertutup (seperti kapal selam) oleh karena adaptasi ini bersifat *fisiologis* dan *reversibel*. Penurunan kinerja atau pada aktivitas fisik yang normal tidak terjadi pada tingkat konsentrasi ini.

Gambaran-gambaran ini berlaku untuk *karbon dioksida* murni. Dalam ruangan tertutup yang dipenuhi orang, konsentrasi *karbon dioksida* akan mencapai tingkat yang lebih tinggi daripada konsentrasi di udara bebas. Konsentrasi yang lebih besar dari *1.000 ppm* akan menyebabkan ketidaknyamanan terhadap 20% penghuni dan ketidaknyamanan ini akan meningkat seiring dengan meningkatnya konsentrasi *karbon dioksida*. Ketidaknyamanan ini diakibatkan oleh gas-gas yang dikeluarkan sewaktu pernafasan dan keringatan manusia, bukan oleh karbon dioksida. Pada konsentrasi *2.000 ppm*, mayoritas penghuni akan merasakan ketidaknyamanan yang signifikan dan banyak yang akan mual-mual dan sakit kepala. Konsentrasi *karbon dioksida* antara *300 ppm* sampai dengan *2.500 ppm* digunakan sebagai indikator kualitas udara dalam ruangan.

Keracunan *karbon dioksida* akut dikenal sebagai lembap hitam. Para penambang biasanya akan membawa sesangkar burung kenari ketika mereka sedang bekerja untuk memperingati mereka ketika kadar *karbon dioksida* mencapat tingkat yang berbahaya. Burung kenari akan terlebih dahulu mati sebelum kadar *karbon dioksida* mencapai tingkat yang berbahaya untuk manusia. *Karbon dioksida* menyebabkan kematian yang luas di Danau Nyos di Kamerun pada tahun 1996.

Karbon dioksida yang lebih berat yang dikeluarkan mendorong *oksigen* keluar, menyebabkan kematian hampir 2000 orang.

*Hemoglobin*, molekul pengangkut oksigen yang utama pada sel darah merah, mengangkut baik *oksigen* maupun *karbon dioksida*. Namun *karbon dioksida* yang diangkut *hemoglobin* tidak terikat pada tempat yang sama dengan *oksigen*. Ia bergabung dengan gugus terminal-N pada empat rantai *globin*. Namun, karena efek *alosterik* pada molekul *hemoglobin*, pengikatan *karbon dioksida* mengurangi jumlah *oksigen* yang dapat diikat. Penurunan pengikatan *karbon dioksida* oleh karena peningkatan kadar *oksigen* dikenal sebagai *efek Haldane* dan penting dalam *trasportasi karbon dioksida* dari jaringan ke paru-paru. Sebaliknya, peningkatan tekanan parsial $CO_2$ atau penurunan *pH* akan menyebabkan pelepasan *oksigen* dari *hemoglobin*, dikenal sebagai *efek Bohr*

$CO_2$ diangkut di darah dengan tiga cara yang berbeda:

- Kebanyakan (sekitar 70% - 80%) dikonversikan menjadi *ion bikarbonat* $HCO_3^-$ oleh enzim *karbonat anhidrase* di sel-sel darah merah dengan reaksi

$$CO_2 + H_2O \rightarrow H_2CO_3 \rightarrow H^+ + HCO_3^-.$$

- 5% - 10% larut di *plasma*

- 5% - 10% diikat oleh *hemoglobin* sebagai senyawa *karbamino*

*Karbon dioksida* adalah salah satu *mediator autoregulasi* setempat suplai darah. Apabila kadar *karbon dioksida* tinggi, *kapiler* akan mengembang untuk mengijinkan arus darah yang lebih besar ke jaringan yang dituju.

*Ion bikarbonat* sangatlah penting dalam meregulasi *pH* darah. Laju pernafasan seseorang dipengaruhi oleh kadar *karbon dioksida* dalam darahnya. Pernafasan yang terlalu lambat akan menyebabkan *asidosis* pernafasan, sedangkan pernafasan yang terlalu cepat akan menimbulkan *hiperventilasi* yang bisa menyebabkan *alkalosis* pernafasan.

Walaupun tubuh memerlukan *oksigen* untuk *metabolisme*, kadar *oksigen* yang rendah tidak akan menstimulasi pernafasan. Sebaliknya pernafasan distimulasi oleh kadar *karbon dioksida* yang tinggi. Akibatnya, bernafas pada udara bertekanan rendah atau campuran gas tanpa *oksigen* (seperti *nitrogen* murni) dapat menyebabkan kehilangan kesadaran. Hal ini sangatlah berbahaya bagi pilot tempur. Ini juga adalah alasan mengapa penumpang pesawat diinstruksikan untuk memakai masker *oksigen* ke dirinya sendiri terlebih dahulu sebelum membantu orang lain ketika tekanan kabin berkurang, jika tidak maka terjadi resiko tidak sadarkan diri.

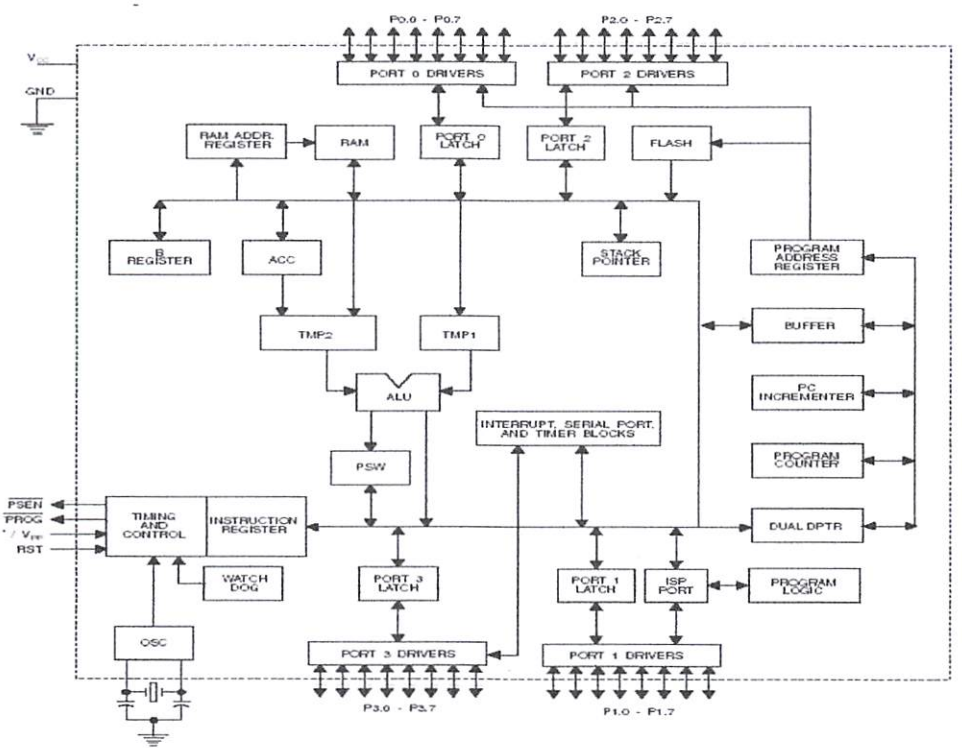
Menurut salah satu kajian dari *Departemen Pertanian Amerika Serikat*, pernafasan orang pada umumnya menghasilkan kira-kira 450 liter (sekitar 900 gram) *karbon dioksida* perhari.

## 2.2. Mikrokontroler AT89S52

### 2.2.1. Fitur mikrokontroler AT89S52

Mikrokontroler AT89S52 merupakan salah satu keluarga MCS-51 keluaran Atmel yang mempunyai 40 kaki, 32 kaki diantaranya digunakan sebagai port paralel. Satu port paralel terdiri dari 8 kaki dengan demikian 32 kaki membentuk 4 buah port paralel, yang masing-masing dikenal dengan port 0, port 1, port 2, port 3. Jenis mikrokontroler ini pada prinsipnya dapat digunakan untuk mengolah data per bit ataupun data 8 bit secara bersamaan.

Sebuah mikrokontroler dapat bekerja bila dalam mikrokontroler tersebut terdapat sebuah program yang berisi instruksi-instruksi yang akan digunakan untuk menjalankan sistem mikrokontroler tersebut. Pada prinsipnya program pada mikrokontroler dijalankan secara bertahap, jadi dalam program itu sendiri terdapat beberapa set instruksi dan tiap instruksi itu dijalankan secara bertahap atau berurutan.

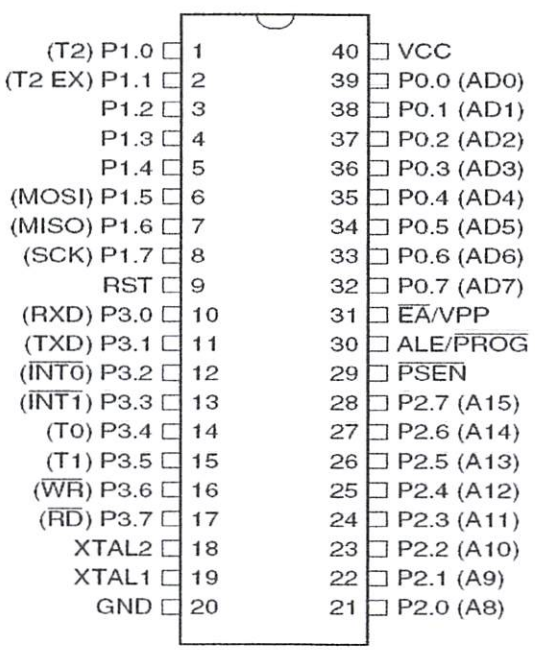



**Gambar 2-1**. Diagram Blok mikrokontroler AT89S52
Sumber : Datasheet mikrokontroller AT89S52

Beberapa spesifikasi yang dimiliki oleh mikrokontroler AT89S52 adalah sebagai berikut :

- Kompatibel dengan semua produk MCS-51
- 8K byte ISP *flash memory*
- 3 level program memori *lock*
- 256 byte RAM *internal*
- Jalur I/0 dua arah 32 buah
- 3 buah *timer/counter* 16-bit
- *Programmable* UART (serial port)

- *Programmable watchdog timer*
- *Dual data pointer*
- Frekuensi kerja 0 sampai 33 MHz
- Teganan operasi 4,0 Volt sampai 5,5Volt




**Gambar 2-2.** Mikrokontroler Atmel AT89S52
Sumber : Datasheet mikrokontroller AT89S52

Semua pin pada mikrokontroler Atmel AT89S52 adalah sama dengan mikrokontroler MCS-51. Penjelasan untuk masing-masing pin dari mikrokontroler adalah sebagai berikut :

- **VCC** digunakan sebagai catu daya
- **Gnd** digunakan sebagai ground
- **Port 0** (P0) merupakan port paralel 8-bit dua arah dan memiliki alamat 80H. Posisi *low significant* bit (LSB) terletak pada pin 39 dan *most significant* (MSB) terletak pada pin 32.
- **Port 1** (P1) merupakan port paralel 8-bit dua arah dan memiliki alamat 90H. Posisi LSB terletak pada pin 1 dan LSB terletak pada pin 8.

Penyangga keluaran port 1 mampu memberikan menyerap arus empat masukan TTL (sekitar 1,6 mA). Port ini mempunyai beberapa fungsi khusus ditunjukkan pada tabel dibawah ini :

**Tabel 2-1.** Fungsi khusus port 1 pada AT89S52

| Pin-pin pada port 1 | Fungsi pengganti |
|---|---|
| P1.0 | T2 (masukan luar untuk timer/counter 2) |
| P1.1 | T2 EX (timer/counter 2 capture/ reload triger dan control arah) |
| P1.2 | - |
| P1.3 | - |
| P1.4 | - |
| P1.5 | MOSI (Master Data Output, Slave Data Input untuk kanal SPI) |
| P1.6 | MISO (Master Data Input, Slave Data Output untuk kanal SPI) |
| P1.7 | SCK (Master clock output, Slave clock input untuk kanal SPI) |

Sumber : Datasheet mikrokontroller AT89S52

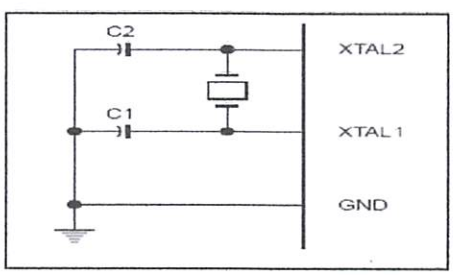
- **Port 2** (P2) merupakan port paralel 8-bit dua arah dan memiliki alamat A0H. Posisi *low significant* bit (LSB) terletak pada pin 21 dan *most significant* (MSB) terletak pada pin 28. Penyangga keluaran port 2 mampu memberikan menyerap arus empat masukan TTL (sekitar 1,6 mA).
- **Port 3** (P3) merupakan port paralel 8-bit dua arah dan memiliki alamat B0H. Posisi LSB terletak pada pin 10 dan LSB terletak pada pin 17. Penyangga keluaran port 3 mampu memberikan menyerap arus empat masukan TTL (sekitar 1,6 mA). Port ini mempunyai beberapa fungsi khusus ditunjukkan pada tabel dibawah ini :

**Tabel 2-2.** Fungsi khusus port 3 pada AT89S52

| Pin-pin pada port 3 | Fungsi pengganti |
|---|---|
| P3.0 | RXD (port input serial) |
| P3.1 | TXD (port output serial) |
| P3.2 | INT0 (interrupt eksternal 0) |
| P3.3 | INT1 (interrupt eksternal 1) |
| P3.4 | T0 (input eksternal timer 0) |
| P3.5 | T1 (input eksternal timer 1) |
| P3.6 | WR (perintah write pada memori eksternal) |
| P3.7 | RD (perintah read pada memori eksternal) |

Sumber : Datasheet mikrokontroller AT89S52

- **RST** (reset) pada kondisi high akan aktif selama dua siklus.
- **ALE/PROG** atau *Adreess Latch Enable* menghasilkan pulsa-pulsa untuk mengancing byte rendah (*low byte*) alamat selama pengaksesan memori eksternal.
- **EA/Vpp** atau *External Access Enable* pada kondisi low maka mikrokontroler menjalankan instruksi-instruksi yang ada pada memori internal.
- **XTAL 1** sebagai masukan dari rangkaian osilator.
- **XTAL 2** sebagai keluaran dari rangkaian osilator.

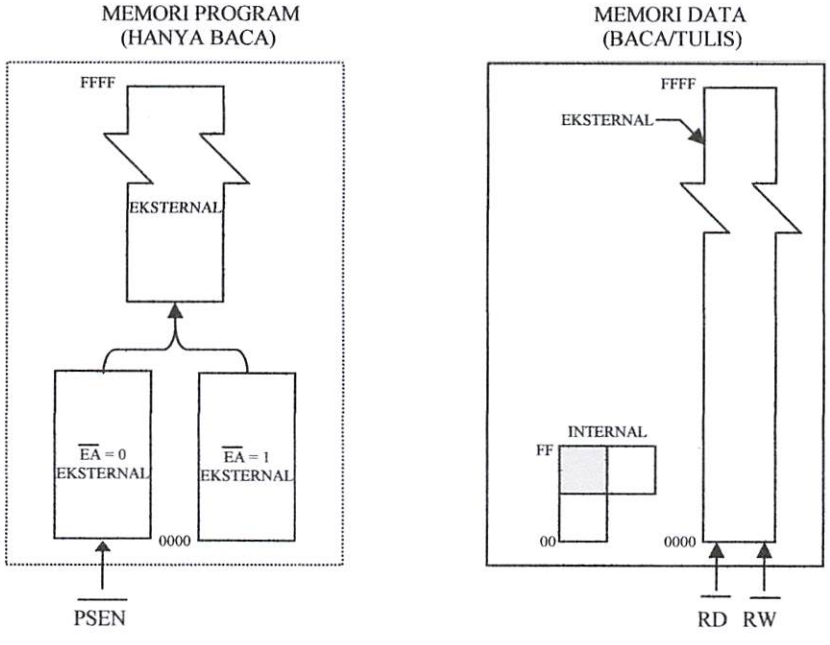



**Gambar 2-3.** Rangkaian osilator
Sumber : Datasheet mikrokontroller AT89S52

### 2.2.2. Organisasi memori

Mikrokontroler AT89S52 mengimplementasikan ruang memori yang terpisah antara program dan data. Kapasitas memori internal 8 Kb, tetapi keduanya dapat diperluas dengan memori *eksternal* sampai 64 Kb memori program dan 64 Kb memori data. Memori *eksternal* ini dapat diaktifkan dengan sinyal PSEN, sedangkan pin EA dinolkan ke ground, maka akan menghasilkan sinyal *Write* (WR) dan *read* (RD).

Namun AT89S52 telah memiliki ROM *internal* sebanyak 8 Kb yang dipakai sebagai memori program serta RAM *internal* sebanyak 256 byte yang digunakan sebagai memori data. Penunjukkan alamat kedua memori ini dihasilkan melalui register *Data Pointer* (DPTR).




**Gambar 2-4.** Struktur memori program dan data
Sumber : Datasheet mikrokontroller AT89S52

### 2.2.3. Special Function register (SFR)

*Special Function Register* atau SFR adalah register-register yang mempunyai fungsi khusus, di antaranya ada yang digunakan untuk mengatur input output data dari mikrokontroler. Misalnya register P0, P1, P2, dan P3 digunakan sebagai register

untuk menampung data input/output. Selain itu juga digunakan untuk mengatur dan memantau kondisi UART, yaitu pada register SCON. Register yang mengatur kerja *timer* adalah TCON.

#### 2.2.3.1. Accumulator

*Accumulator* menempati lokasi E0H merupakan register yang berfungsi menyimpan data sementara dalam program. Register accumulator ini sering digunakan dalam proses operasi aritmatika, logika, pengambilan dan penerimaan data.

#### 2.2.3.2. Register B

Register B menempati lokasi F0H digunakan untuk operasi perkalian dan pembagian untuk instruksi lain dapat diperlakukan sebagai register *scratch pad* (Papan coret-coret).

#### 2.2.3.3. Register Port

Register ini terdapat 4 buah yaitu register P0, P1, P2, dan P3 masing-masing menempati lokasi 80H, 90H, A0H, dan B0H digunakan sebagai sarana input/output untuk menyimpan data dari atau ke masing-masing port 0, port 1, port 2, dan port 3.

#### 2.2.3.4. Register Timer

Mikrokontroler AT89S52 mempunyai tiga buah 16-bit timer , yaitu timer 0, timer 1, dan timer 2. Timer 0 dibentuk oleh register TH0 dan TL0 di lokasi 8CH dan 8AH. Timer 1 dibentuk oleh register TH1 dan TL1 di lokasi 8DH dan 8BH. Timer 2 dibentuk oleh register TH2 dan TL2 di lokasi CDH dan CCH. Perilaku dari register TH0, TH1, TH2, TL0, TL1, dan TL2 diatur oleh register TMOD dan register TCON.

### 2.2.3.5. Register Control

Ada beberapa register yang berisi bit-bit kontrol dan status untuk sistem interupsi, pencacah atau pewaktu, dan port serial antara lain :

- Register IP (*Interrupt Priority*) digunakan untuk mengatur prioritas dari masing-masing interupsi. Tingkat prioritas semua sumber *interupt* dapat diprogram sendiri-sendiri dengan *set* atau *clear* bit pada (*Interupt Priority*). Jika dua permintaan interupsi dengan tingkat prioritas yang berbeda diterima secara bersamaan, permintaan interupsi dengan prioritas tertinggi yang akan dilayani. Jika permintaan interupsi dengan prioritas yang sama diterima bersamaan, akan dilakukan polling untuk menentukan mana yang akan dilayani. Bit-bit pada IP adalah sebagai berikut:

MSB LSB

| - | - | - | PS | PT1 | PX1 | PT0 | PX0 |
|---|---|---|---|---|---|---|---|

Priority bit = 1 menandakan prioritas tinggi (*high priority*)

Priority bit = 0 menandakan prioritas rendah (*low priority*)

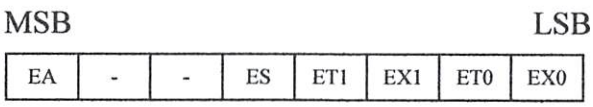
**Gambar 2-5.** Register IP pada mikrokontroler AT89S52
Sumber : Datasheet mikrokontroller AT89S52

**Tabel 2-3.** Register IP pada mikrokontroler AT89S52

| Simbol | Posisi | Fungsi |
|---|---|---|
| - | IP.7 | Cadangan |
| - | IP.6 | Cadangan |
| - | IP.5 | Cadangan |
| PS | IP.4 | Bit interupsi port serial |
| PT1 | IP.3 | Bit interupsi timer 1 overflow |
| PX1 | IP.2 | Bit interupsi eksternal 1 |
| PT0 | IP.1 | Bit interupsi timer 0 overflow |
| PX0 | IP.0 | Bit interupsi eksternal 0 |

Sumber : Datasheet mikrokontroller AT89S52

- Register IE (*Interrupt Enable*) digunakan untuk mengaktifkan atau menonaktifkan sarana interupsi. IE.0 sampai IE.6 mengatur masing-masing sumber interupsi, sedangkan IE.7 mengatur interupsi secara keseluruan. Jika IE.7 bernilai 0 maka sistem interupsi akan non aktif atau keadaan dari IE.0 sampai IE.6 tidak diperhatikan. Masing-masing sumber interupsi dapat diaktifkan dan dimatiakan secara individual atau me-nol-kan bit-bit IE (*interrupt Enable*) dalam SFR. Register IE ini juga mengandung sebuah bit untuk aktivasi interupsi secara global, yang dapat digunakan untuk mengaktifkan dan mematikan interupsi secara keseluruan.




Jika isinya = 1 artinya bit aktif (*enable*)

Jika isinya = 0 artinya bit pasif (*disable*)

**Gambar 2-6.** Register IE pada mikrokontroler AT89S52
Sumber : Datasheet mikrokontroller AT89S52

**Tabel 2-4.** Register IE pada mikrokontroler AT89S52

| Simbol | Posisi | Fungsi |
|---|---|---|
| EA | IE.7 | Untuk menghidupkan (IE=1)dan mematikan (IE=0) |
| - | IE.6 | Cadangan |
| - | IE.5 | Cadangan |
| ES | IE.4 | Bit interupsi port serial |
| ET1 | IE.3 | Bit interupsi timer 1 overflow |
| EX1 | IE.2 | Bit interupsi eksternal 1 |
| ET0 | IE.1 | Bit interupsi timer 0 overflow |
| EX0 | IE.0 | Bit interupsi eksternal 0 |

Sumber : Datasheet mikrokontroller AT89S52

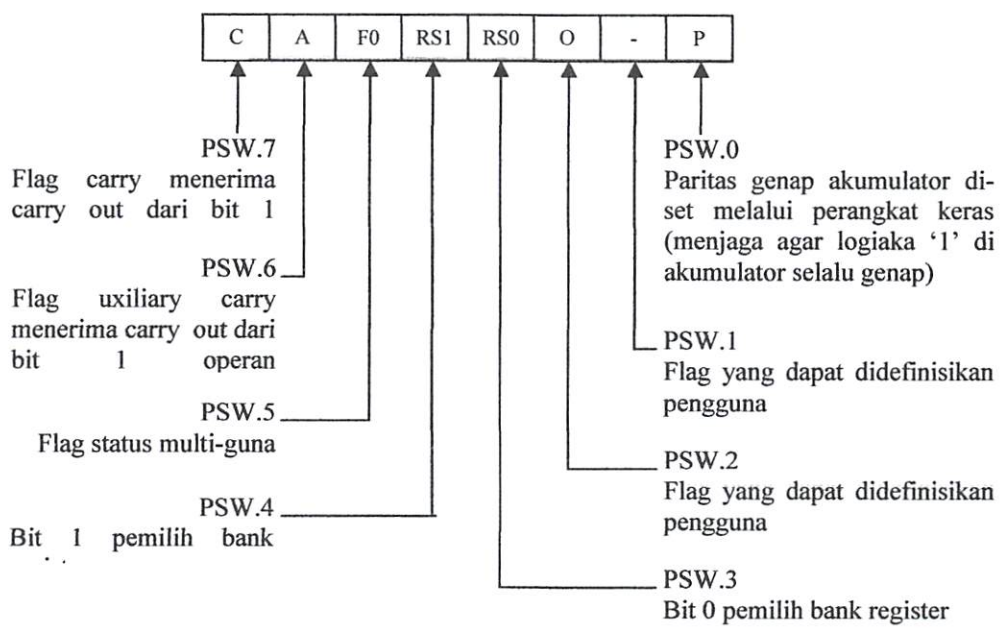
- Register TMOD (*Timer Mode*) digunakan untuk mengatur mode kerja timer. Dengan mengatur mode kerja timer untuk diatur menjadi timer 16-bit, timer13-bit, atau timer 8-bit. Selain itu register ini juga dapat mengatur agar

proses pencacah timer dapat dikendalikan malalui sinyal dari luar mikrokontroler.

- Register TCON (*Timer Control*) digunakan untuk memulai atau menghentikan proses pencacah timer, mengatur sinyal interupsi dari INT0 atau INT1.
- Register SCON (*Serial Control*) digunakan untuk mangatur perilaku dari UART yang diantaranya memantau proses pengiriman dan penerimaan data seri.
- Register PCON (*Power Control*) digunakan untuk mengatur pemakaian daya pada IC.

### 2.2.3.6. Program Status Word

Register *Program Status Word* (PSW) menempati lokasi D0H ini berisi informasi status program yang mana masing-masing bit menunjukkan kondisi *Central Processing Unit* (CPU) setelah operasi dijalankan. Terdiri atas beberapa bit sebagai berikut:




**Gambar 2-7**. Register PSW dalam mikrokontroler AT89S52
Sumber : Datasheet mikrokontroller AT89S52

### 2.2.3.7. Stack Pointer

Register *Stack Pointer* (SP) menempati lokasi 81H merupakan register 8-bit yang terletak pada alamat 81H yang mempunyai fungsi untuk menyimpan data pada saat terjadi interrupt.

### 2.2.3.8. Data Pointer

Register *Data Pointer* (DPTR) merupakan register 16-bit atau sebagai dua register 8-bit yang terpisah yaitu data pointer high byte (DPH) dan data pointer low byte (DPL) yang masing-masing menempati lokasi 83H dan 82H.

### 2.2.3.9. Serial Data Buffer

*Serial Data Buffer* (SBUF) terletak pada lokasi 99H yang dibagi menjadi dua register terpisah, yaitu penyangga pengirim atau *transmit buffer* dan penerima atau *receive buffer*. Pada saat data disalin ke SBUF, maka data sesunggunya dikirim ke penyangga pengirim dan sekaligus mengawali transmisi data serial. Sedangkan pada saat data disalin dari SBUF, maka sebenarnya data tersebut berasal dari penyangga penerima.

## 2.2.4. Mode Pengalamatan

Data atau operan bisa berada ditempat yang berbeda sehingga dikenal beberapa cara untuk mengakses data tersebut yang dinamakan sebagai mode pengalamatan (*addressing mode*).

**1. Pengalamatan langsung (*Direct Addressing*)**

Dalam pengalamatan langsung, operan-operan ditentukan berdasarkan alamat 8-bit dalam suatu instruksi. Hanya RAM data internal dan SFR yang bisa diakses secara langsung.

**2. Pengalamatan tak langsung (*Indirect Addressing*)**

Dalam pengalamatan tak langsung, instruksi menentukan suatu register yang digunakan untuk menyimpan alamat operan. Baik RAM internal maupun eksternal dapat diakses secara tak langsung.

Register alamat untuk alamat-alamat 8-bit bisa menggunakan *stack pointer* atau R0 atau R1 dari bank register yang dipilih. Sedangkan untuk alamat 16-bit hanya bisa menggunakan register pointer 16-bit atau DPTR.

**3. Instruksi-instruksi register**

Bank-bank register yang masing-masing R0 hingga R7 atau 8 register dapat diakses melalui instruksi yang OP-kodenya mengandung 3-bit spesifikasi register (000 untuk R0, 001 untuk R1 hingga 111 untuk R7). Pengaksesan register dengan cara demikian bisa menghemat penggunaan kode instruksi, karena tidak memerlukan sebuah byte untuk alamat. Saat instruksi tersebut dijalankan satu dari delapan register pada bank yang terpilih yang diakses.

**4. Instruksi-instruksi register khusus**

Beberapa instruksi hanya dikhususkan untuk suatu register tertentu. Misalnya suatu instruksi yang hanya bekerja pada akumulator saja, sehingga tidak memerlukan alamat byte untuk menunjuk ke akumulator tersebut. Dalam hal ini op-kodenya sendiri telah mengandung penunjuk ke register yang benar. Instruksi yang mengacu akumulator sebagai A akan dikodekan dengan op-kode spesifik akumulator.

**5. Konstata langsung (*Immediate Constants*)**

Nilai dari suatu konstanta dapat segera menyatu dengan op-kode dalam memori program. Misalnya instruksi : MOV A,#100, yang akan menyimpan

konstanta 100 (desimal) ke dalam akumulator. Bilangan yang sama tersebut bisa juga dituliskan dengan format heksa sebagai 64H.

**6. Pengalamatan terindeks (*Indexed Addressing*)**

Memori program hanya dapat diakses melalui pengalamatan terindeks. Mode pengalamatan ini ditunjukan untuk membaca tabel (look-up tables) yang tersimpan dalam memori program. Sebuah register dasar 16-bit menunjuk ke awal atau dasar tabel dan akumulator di-set dengan angka indek tabel yang akan diakses. Alamat dari entri tabel dalam memori program dibentuk dengan menjumlahkan data akumulator dengan penunjuk awal tabel.

Type lain dari pengalamatan terindeks digunakan dalam interuksi-interuksi "lompat bersyarat". Dalam hal ini alamat tujuan dari interuksi lompat (*jump*) dihitung sebagai jumlah penunjuk dasar (*base pointer*) dengan data akumulator.

### 2.2.5. Set instruksi

Secara keseluruan MCS51 mempunyai 255 macam instruksi, yang dibentuk dengan mengkombinasikan dan operan. Instruksi tersebut dikelompokkan menjadi berikut :

**1. Kelompok penyalin data**

Instruksi dasar untuk kelompok ini adalah MOV, singkatan dari move yang artinya memindahkan, meskipun demikian lebih tepat dikatakan instruksi ini mempunyai arti menyalin data. Misalnya, instruksi-instruksi berikut ini :

```
MOV  A, 20H      ; salin isi lokasi memori 20H ke akumulator A
MOV  A, @R1      ; salin isi lokasi yang ditunjuk R1 ke A
MOV  A, P1       ; salin isi latch port 1 ke A
MOV  P3, A       ; salin isi A ke latch port 3
```

Untuk pemakaian pada memori program , instruksi ini ditulis menjadi MOVC, hanya ada 2 jenis instruksi yang menggunakanya yaitu :

```
MOVC  A, @A+DPTR      ; DPTR sebagai register tak langsung
MOVC  A, @ A+PC       ; PC sebagai register tak langsung
```

Selain itu masih dikenal pula instruksi MOVX yang digunakan untuk mengakses memori data eksternal. Hanya ada 6 macam instruksi yang memakai MOVX, antara lain :

```
MOVX  A,@DPTR      ; salin data eksternal pada lokasi DPTR ke A
MOVX  A,@R0        ; salin data eksternal pada lokasi R0 ke A
MOVX  A,@R1        ; salin data eksternal pada lokasi R1 ke A
MOVX  @DPTR, A     ; salin data eksternal dari A ke lokasi DPTR
MOVX  @R0, A       ; salin data eksternal dari A ke lokasi R0
MOVX  @R1,A        ; salin data eksternal dari A ke lokasi R1
```

**2. Kelompok instruksi aritmatik**

Instruksi-instruksi dalam aritmatik ini selalu melibatkan akumulator, hanya beberapa yang juga melibatkan register lainya. Instruksi yang termasuk dalam kelompok ini, antara lain :

A. Instruksi ADD dan ADDC

Isi akumulator A ditambahkan dengan suatu bilangan (1 byte), maka hasil penjumlahannya akan ditampung kembali dalam akumulator. Dalam operasi ini bit *Carry* berfungsi sebagai penampung limpahan hasil penjumlahan. Jika hasil penjumlahan tersebut melimpah maka bit *Carry* akan bernilai '1' jika tidak *Carry* akan selalu bernilai '0'.

B. Instruksi SUBB

Isi akumulator dikurangi dengan bilangan (1 byte) besertadengan nilai bit *Carry*, hasil pengurangan akan ditampung kembali dalam akumulator. Dalam hal ini bit *Carry* juga berfungsi sebagai penampung limpahan dari hasil pengurangan. Jika hasil pengurangan tersebut melimpah maka bit *Carry* akan bernilai '1' jika tidak akan bernilai '0'.

C. Instruksi DA A

Instruksi DA A (*Decimal Adjust*) dipakai setelah instruksi ADD, ADDC atau SUBB, digunakan untuk merubah nilai biner8-bit yang tersimpan dalam akumulator menjadi 2 digit bilangan dalam format BCD (*Binary Code Decimal*).

D. Instruksi MUL AB

Bilangan biner 8-bit dalam akumulator A dikalikan dengan bilangan biner 8-bit dalam register B. Hasil perkalian berupa bilangan biner 16-bit, 8-bit bilangan biner yang bagian atas (*high byte*) disimpan di register B. Sedangkan 8-bit lainya (*low byte*) disimpan di akumulator A.

E. Instruksi DIV AB

Bilangan biner 8-bit dalam akumulator A dibagi dengan bilangan biner 8-bit dalam register B. Hasil perkalian berupa bilangan biner 16-bit, 8-bit bilangan biner yang bagian atas (*high byte*) disimpan di register B. Sedangkan 8-bit lainya (*low byte*) disimpan di akumulator A.

F. Instruksi DEC dan INC

Interuksi DEC digunakan untuk menurunkan satu nilai (1 byte). Jika nilai awal adalah 00H, maka setelah dilaksanakan instruksi ini hasilnya adalah FFh. Tidak ada flag yang terpengaruh. Sedangkan instruksi INC digunakan untuk

menaikkn nilai (1 byte). Jika nilai awalnya adalah FFH maka setelah dilaksanakan instruksi ini menjadi 00H.

G. Instruksi INC DPTR

Instruksi penaikkan (*increment*) yang bekerja pada data 16-bit yaitu DPTR. Yaitu menaikkan penunjuk data sebesar 1. suatu limpahan dari byte rendah (*low order*) dari DPTR atau DPL akan menaikkan byte tinggi (*high order*) yaitu yang tersimpan dalam DPH sebesar 1.

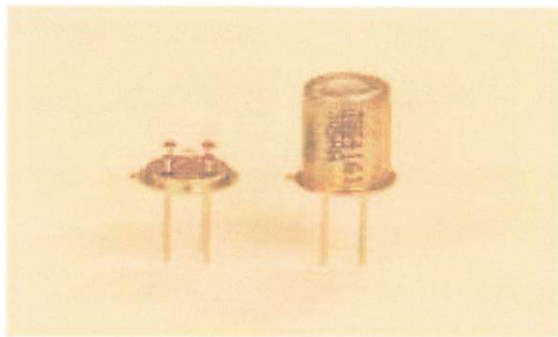
**3. Kelompok instruksi logika**

Kelompok instruksi ini dipakai untuk melakukan operasi logika, yaitu operasi AND (instruksi ANL), operasi OR (Instrukai ORL),operasi exlusive OR (instruksi XRL), operasi clear (instruksi CLR), instruksi negasi atau komplemen (instruksi CPL), operasi pergeseran kanan atau kiri (instruksi RR, RRC, RL dan RLC) serta operasi pertukaran data (instruksi SWAP).

## 2.3 Sensor Gas Karbon Dioksida *( TGS 4161 )*

Sensor yang digunakan untuk mendeteksi *karbon dioksida* adalah *TGS 4161.* Sensor ini dapat mendeteksi karbon dioksida dari *350-8.000 ppm* dalam udara sehingga sensor ini sangat sensitif sekali dan sangat mendukung untuk mendeteksi karbon dioksida di dalam ruangan.

Sensor ini menggunakan unsur *solid elektrolid* yang dibentuk diantara dua *electrodes* yang kemudian bersamaan dicetak *heater* ($RuO_2$) *substrate.* Dengan perubahan pada *electromotive force (EMF)* yang dihasilkan dua *electrodes* memungkinkan untuk mengukur konsentrasi gas *karbon dioksida.*



**Gambar 2-8.** Sensor TGS 4161 / Sensor $CO_2$
Sumber : Datasheet TGS 4161

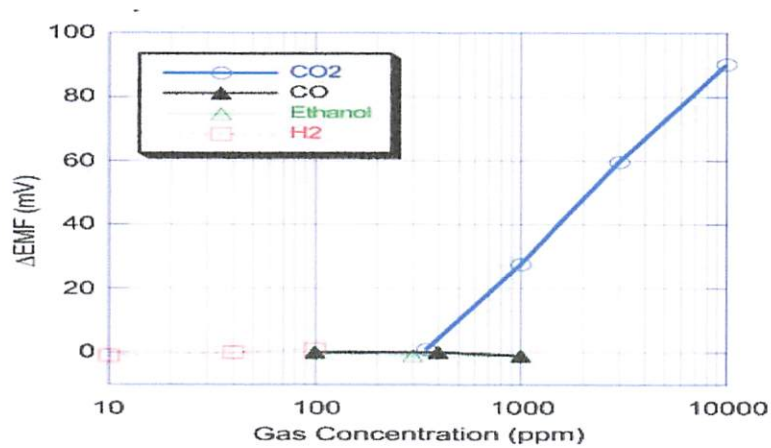
Grafik di bawah ini merupakan ciri khas dari sensitivitas sensor. *Y-axis* yang diindikasikan *ΔEMF* yang didefinisikan sebagai berikut :

*ΔEMF = EMF1-EMF2*

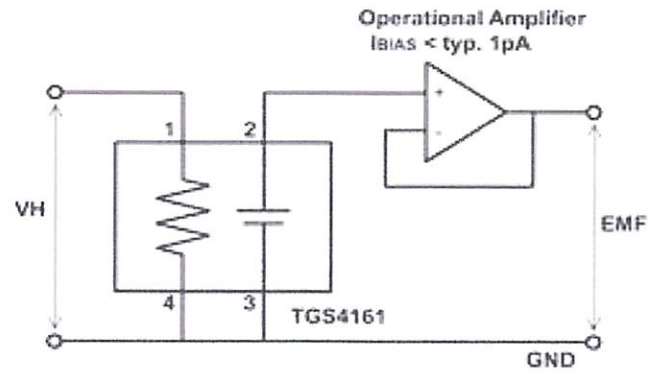
dimana

*EMF1* = *EMF* di *350 ppm CO2*

*EMF2* = *EMF* gas yang terdapat dalam konsentrasi




**Grafik 2-1.** *Sensitivity characteristics TGS 4161*
Sumber : Datasheet TGS 4161

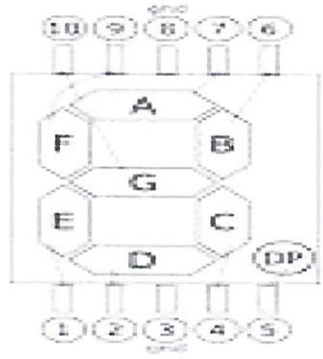



**Gambar 2-9.** *Basic measuring circuit*
Sumber : Datasheet TGS 4161

Sensor membutuhkan masukan *heater voltage* ($V_H$). *Heater voltage* adalah aplikasi yang terintegrasi pada heater yang berfungsi untuk mempertahankan deteksi secara spesifik pada suhu,yang dimana optimal untuk mendeteksi. *Electromotive Force (EMF)* sensor harus diukur dengan menggunakan impedansi tinggi *(>100GΩ) Op-Amp* dengan *current bias < 1 pA*. Sensor ini menggunakan bahan electrolyte yang membuat sensor ini berfungsi seperti baterai.

## 2.4 Display 7 segmen.

Display 7 segmen merupakan display yang terbentuk oleh 7 buah LED (ditambah 1 LED untuk menampilkan titik) yang dirangkai untuk menampilkan bilangan heksadesimal. Kedelapan LED tersebut bias dirangkai membentuk rangkaian common anoda atau common katoda. Pada skripsi ini yang dipakai adalah 7 segmen common anoda karena jenis ini bias langsung dihubungkan dengan port mikrokontroller melalui sebuah transistor. Gambar 2.3 memperlihatkan konfigurasi segmen dan rangkaian ekuivalen 7 segmen jenis common anoda.




**Gambar 2-10.** Display 7 segmen
Sumber : Teknik antarmuka + pemprograman mikrokontroller AT89S52, 2008

Display 7 segmen, seperti ditunjukkan oleh gambar 2.3, terdiri atas segmen-segmen yang diberi nama segmen A,B,C,D,E,F dan,G serta DP untuk menampilkan tanda titik. Dengan mengatur segmen-segmen mana yang menyala, display 7 segmen bias menampilkan bilangan heksadesimal (0,1,2,3,….,A,B,C,D,E,F). Tabel 2.1 memperlihatkan segmen mana saja yang harus dinyalakan untuk menampilkan bilangan heksadesimal ini, ON berarti nyala dan OFF berarti mati.

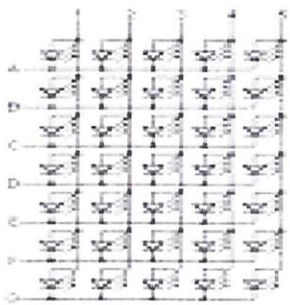
**Tabel 2-5** Display 7 segmen

| Angka | Segmen | | | | | | |
|---|---|---|---|---|---|---|---|
| | A | B | C | D | E | F | G |
| 0 | ON | ON | ON | ON | ON | ON | ON |
| 1 | | ON | ON | | | | |
| 2 | ON | ON | | ON | ON | | ON |
| 3 | ON | ON | ON | ON | | | ON |
| 4 | | ON | ON | | | ON | ON |
| 5 | ON | | ON | ON | | ON | ON |
| 6 | ON | | ON | ON | ON | ON | ON |
| 7 | ON | ON | ON | | | | |
| 8 | ON | ON | ON | ON | ON | ON | ON |
| 9 | ON | ON | ON | ON | | ON | ON |
| A | ON | ON | ON | | ON | ON | ON |
| D | | | ON | ON | ON | ON | ON |
| C | ON | | | ON | ON | ON | |
| D | | ON | ON | ON | ON | | ON |
| E | ON | | | ON | ON | ON | ON |
| F | ON | | | | ON | ON | ON |

Sumber : Teknik antarmuka + pemprograman mikrokontroller AT89S52, 2008

## 2.5 Display Dot Matriks.

Tampilan matriks titik *(dot matriks)* terdiri dari sejumlah LED yang disusun pada baris dan kolom. Susunan yang terdapat dipasaran adalah matriks 5x7 titik, 5x8 titik, 16x16 titik dan masih banyak lagi. Pada tampilan ini yang digunakan adalah tampilan 5x7 titik, seperti digambarkan dalam gambar 2.9 dibawah ini.



**Gambar 2-11.**. Display dot matriks 5x7 titik.
Sumber Teknik antarmuka + pemprograman mikrokontroller AT89S52, 2008

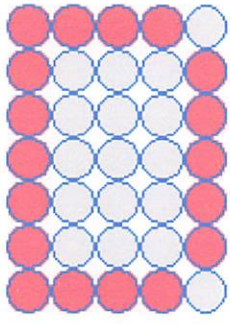
Matrik 5x7 titik mampu menyajikan karakter alfanumerik yang lengkap. Proses pembangkitan karakter antara lain melibatkan proses *scanning* baris atau kolom, memilih LED yang tepat pada baris (kolom), dan menyalakannya. Proses ini diulang untuk baris atau kolom berikutnya. Setelah semua baris atau kolom yang dipilih dengan urutan tertentu, proses diatas diulang mulai dari baris paling atas atau kolom pertama.

Jika frekuensi *scanning*nya cukup cepat (sekitar 100Hz), maka akan didapatkan karakter bebas kedip. Jika matrik di*scan* dari kiri kekanan, kolom demi kolom, disebut *vertical scanning*. Jika dilakukan baris demi baris, disebut *horizontal scanning.*

### 2.5.1 Dekode matriks LED

Untuk menyalakan angka decimal dengan pembacaan matriks titik, LED yang dibutuhkan tidak dinyalakan serentak. Biasanya LED diaktifkan baris demi baris secara cepat. Sebagai contoh pada gambar 2.10 dimulai dengan mentanahkan

(ground) baris pertama dan memasang tegangan pada kolom vertical yang sesuai. Setelah suatu perangkat singkat, baris kedua digroundkan dan tegangan dipasang pada kolom-kolom tertentu, hal ini akan dilanjutkan pada baris-baris berikutnya dan akan diulang lagi. Bila proses ini diulang secara cepat, suatu alfanumerik akan tertampil tanpa kedipan.

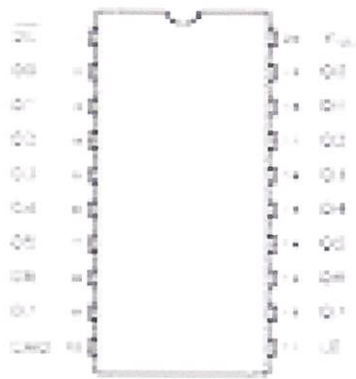


**Gambar 2-12.** Dekode LED matriks dalam tampilan karakter.
Sumber : Teknik antarmuka + pemprograman mikrokontroller AT89S52, 2008

Rangkaian yang menjalankan matriks LED cukup rumit karena harus melakukan proses *scanning* baris-baris horizontal dan pada saat yang sama memesang tegangan pada kolom yang sesuai. Secara umum pencacahan lingkar *(ring counter)* melakukan *scan* pada baris horizontal, sedangkam memori berisi data kolom LED yang menyala memberikan tegangan pada kolom-kolom vertical.

## 2.6. Shift register 8 bit IC 74HCT164.

IC 74HCT164 merupakan *shift register* dengan input-input serial yang mana mempunyai susunan output yang pararel pada setiap tingkatnya. *Shift register* 8 bit ini digunakan untuk proses *scan* data kolom pada LED dot matriks. Keluaran dari tiap-tiap pin dihubungkan ketiap-tiap kolom dot matriks. Untuk konfigurasi dari kaki-kaki IC 74HCT164 dapat diliat pada gambar 2.11.



**Gambar 2-13.** IC 74HCT164
Sumber : Datasheet 74HCT164

**Tabel 2-6.** Pin-pin IC 74HCT164

| Simbol | Pin | Diskripsi |
|---|---|---|
| DSA | 1 | Data input |
| DSB | 2 | Data input |
| Q0 | 3 | Output |
| Q1 | 4 | Output |
| Q2 | 5 | Output |
| Q3 | 6 | Output |
| GND | 7 | Ground |
| CP | 8 | Clock (low to hight) |
| MR | 9 | Reset |
| Q4 | 10 | Output |
| Q5 | 11 | Output |
| Q6 | 12 | Output |
| Q7 | 13 | Output |
| VCC | 14 | Tegangan supplay |

Sumber : Datasheet 74HCT164

## 2.7. RTC (Real Time Clock) DS1307

*Real Time clock* (RTC) merupakan alat pencatat sekaligus pemberi informasi waktu, RTC yang kami gunakan disini merupakan RTC serial tipe DS1307. Chip DS1307 buatan *Dallas semiconductor* merupakan alat pencatat waktu yang baik, dalam chip itu tersimpan batteray *Litium* sehingga pencatatan waktu tidak terhenti jika catu daya alat pemakaian terhenti dan proses dapat mengambil kapanpun hasil kerjanya tanpa harus mengatur. Waktu dicatat mulai dari detik, menit, jam, hari, bulan, tahun bahkan sampai abad, dan penanggalannya dirancang bisa bekerja tanpa salah sampai tahun 2099. Pencatatan jam bisa dilakukan dalam format 24 jam perhari atau 12 jam AM/PM.

DS1307 memiliki built-in yang berfungsi sebagai sirkuit yang mendeteksi gangguan listrik dan secara otomatis beralih ke baterai pasokan. Karakteristik dari DS1307 adalah sebagai berikut :

1. RTC menyimpan data-data detik, menit, jam, tanggal, bulan, hari dalam seminggu, dan tahun valid hingga 2100.
2. 56-byte, baterai cadangan, RAM nonvolatile (NV) RAM untuk penyimpanan.
3. Antarmuka serial *two-wire*(I2C).
4. Sinyal keluaran berupa gelombang kotak terprogram *(programmable squarewave).*
5. Deteksi otomatis kegagalan daya *(power-fail)* dan rangkaian switch.
6. Konsumsi daya kurang dari 500nA menggunakan mode baterai cadangan dengan operasional osilator.
7. Tersedia fitur industry dengan ketahanan suhu: -40°C hingga +85°C.
8. Tersedia dalam kemasa 8-pin DIP atau SOIC.

### 2.7.1. Fungsi kaki-kaki (pin) RTC DS1307 :

1. SCL (*Seial Clock Input*).

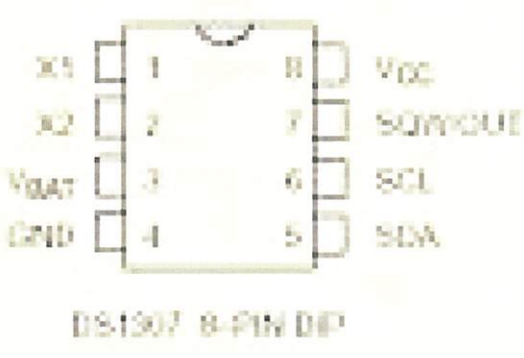
   SCL digunakan untuk singkronisasi pertukaran data pada *serial interface.*

2. SDA *(Serial Data Input/Output)*

   SDA merupakan pin input/output untuk 2 kabel serial *interface.*

3. SQW/OUT *(Square Wave/Output Driver)*

   Ketika *enable*, SQWE diset ke "1". SQW/OUT akan berfungsi ketika diberi tegangan Vcc atau Vbat. SQW/OUT ini juga membutuhkan resistor *pull-up eksternal.* Berikut ini adalah gambar kaki-kaki RTC DS1307 :




**Gambar 2-14.** pin-pin RTC DS1307

Sumber : www.dalsemi.com

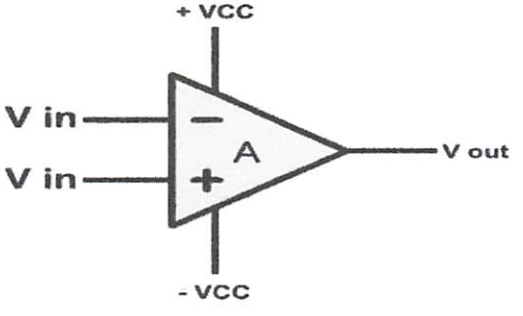
### 2.7.2. Cara kerja RTC DS1307.

DS1307 bekerja sebagai *slave* pada serial bus. Akses yang ada pada saat ada kondisi START dan tersedia identifikasi yang ada pada register alamat. Selanjutnya register-register dapat diakses secara berurutan sampai suatu kondisi STOP terjadi.

Saat Vcc lebih kecil dari 1,25 x Vbat, DS1307 akan mengakhiri suatu akses yang sedang terjadi dan me*reset address counter*. Input yang masuk tidak akan diakses untuk mencegah kesalahan penulisan data. Saat Vcc lebih kecil dari Vbat, DS1307 akan *switch* ke mode baterai arus rendah. Ketika *power-up* DS1307 akan *switch* dari baterai ke Vcc saat Vcc lebih besar dari Vbat+0,2V dan akan mengakses input yang masuk ketika Vcc lebih besar 1,25 x Vbat.

## 2.8. Pengkondisi sinyal

Di dalam sistem kontrol sering kali keluaran dari sensor nilainya tidak sesuai yang diharapkan yaitu nilainya mudah untuk diolah. Oleh karena itu perlu adanya pengolah sinyal agar sinyal keluaran dari sensor dapat kita olah terlebih dahulu agar keluarannya seperti yang diharapkan. Penguat operasional atau sering disebut *op-amp (operasional amplifier)* merupakan komponen elektronika yang berfungsi untuk memperkuat sinyal arus searah *(DC)* maupun arus bolak-balik *(AC)*. Pada prinsipnya penguat operasional hanya bekerja sebagai penguat sinyal bukan penguat daya. Penguat operasional terdiri atas transistor, resistor dan kapasitor yang dirangkai dan dikemas dalam rangkaian terpadu *(integrated circuit)*. Simbol *op-amp* ditunjukkan pada gambar di bawah ini. Vin merupakan masukan sinyal, *Vout* keluaran sinyal, *A* besar penguatan dan *VCC* sumber tegangan.




**Gambar 2-15.** Simbol Op-Amp

Sumber : http://franzaditya.blogspot.com/2009/02/penguat-operasional-op-amp.html

Karakteristik *op-amp* ideal adalah kondisi *op-amp* sesuai dengan teori. Karakteristik **op-amp** ideal adalah sebagai berikut:

1. Faktor penguat tidak terhingga.
2. Tidak memiliki *offset*, maksudnya adalah apabila masukan nol maka keluaran juga nol.
3. *Impedansi* masukan tidak terhingga.
4. *Impedansi* keluaran nol.
5. Lebar *bandwidth* tidak terhingga.
6. *Rise time* nol.
7. Tidak mudah terpengaruh oleh perubahan tegangan sumber maupun perubahan suhu.

Pada kenyataannya dalam pembuatan *op-amp* memiliki keterbatasan sehingga tidak ada *op-amp* yang ideal. *Op-amp* yang ada hanyalah *op-amp* yang mendekati ideal karena karakteristik *op-amp* adalah sebagai berikut :

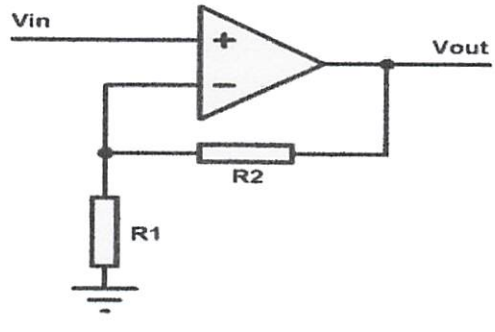
1. Faktor penguat terbatas kurang lebih 100.000 kali
2. Terdapat *offset* dimana saat masukan bernilai nol tegangan keluaran tidak nol.
3. *Impedansi* masukan cukup tinggi namun terbatas sampai kira-kira ratusan kilo *ohm* saja.
4. *Impedansi* keluaran rendah namun terbatas puluhan sampai ratusan *ohm*.
5. *Rise time* tidak nol.
6. Kerja *op-amp* terpengarruh perubahan sumber tegangan dan perubahan pada suhu.

Dalam penggunaannya *op-amp* dibagi menjadi dua jenis yaitu penguat linier dan penguat tidak linier. Penguat linier merupakan penguat yang tetap mempertahankan bentuk sinyal masukan, yang termasuk dalam penguat ini antara lain penguat *non inverting*, penguat *inverting*, penjumlah, penguat *diferensial* dan penguat *instrumentasi*. Sedangkan penguat tidak linier merupakan penguat yang bentuk sinyal keluarannya tidak sama dengan bentuk sinyal masukannya, diantaranya *komparator*, *integrator, diferensiator,* pengubah bentuk gelombang dan pembangkit gelombang. Untuk menangani penguatan dari sensor biasanya digunakan penguat linier yang tidak mengubah bentuk sinyal namun hanya memperkuat sinyal saja.

### 2.8.1 Penguat Non Inverting

Merupakan penguat yang berfungsi memperkuat sinyal masukan tanpa membalik sinyal masukan. Rangkaian penguat dan rumusnya adalah sebagai berikut:

$$Vout = \left(\frac{R2}{R1} + 1\right) \cdot Vin$$



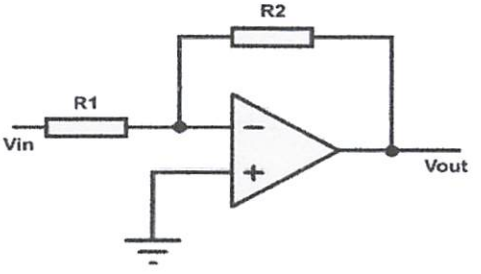

**Gambar 2-16.** Rangkaian Penguat Non Inverting

Sumber : http://franzaditya.blogspot.com/2009/02/penguat-operasional-op-amp.html

### 2.8.2 Penguat Inverting

Rangkaian penguat ini berfungsi untuk memperkuat sinyal masukan dan menggeser sinyal keluaran sebesar $180^0$ sehingga masukan yang positif akan

menghasilkan keluaran negatif, demikian juga dengan masukan negatif akan menghasilkan keluaran positif. rangkaian dan rumusnya adalah sebagai berikut:



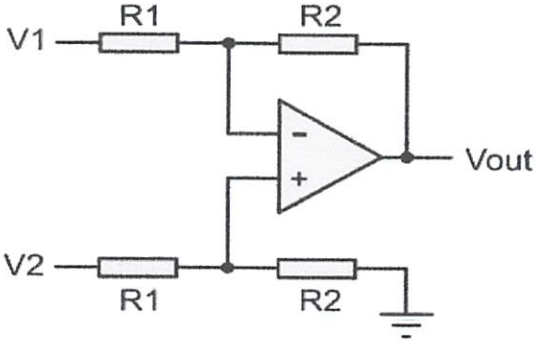

**Gambar 2-17.** Rangkaian Penguat Inverting

Sumber : http://franzaditya.blogspot.com/2009/02/penguat-operasional-op-amp.html

### 2.8.3 Penguat Diferensial

Penguat ini mampu memperkuat sinyal kecil yang berada dalam sinyal yang jauh lebih besar. Keluaran dari penguat ini sebanding dengan perbedaan tegangan kedua masukannya. Rangkaian penguat ini digambarkan dan dirumuskan sebagai berikut:

$$Vout = \left(\frac{R2}{R1}\right) \cdot (V2 - V1)$$



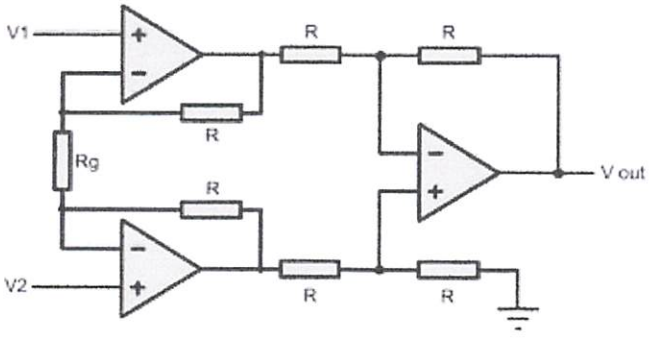

**Gambar 2-18..** Rangkaian Penguat Diferensial

Sumber : http://franzaditya.blogspot.com/2009/02/penguat-operasional-op-amp.html

### 2.8.4 Penguat Instrumentasi

Penguat ini merupakan penguat serba guna dan bermanfaat yang terdiri atas tiga op-amp dan tujuh buah tahanan. Rangkaian ini tersusun atas rangkaian penguat differensial dan penguat penyangga. Untuk mengatur penguatan yang diinginkan diatur dengan mengubah-ubah nilai *Rg*. Rumusan dan gambar dari penguat *instrumentasi* adalah sebagai berikut :

$$Vout = (v2 - v1) \cdot \left(1 + \left(\frac{2 \cdot R1}{Rg}\right)\right)$$



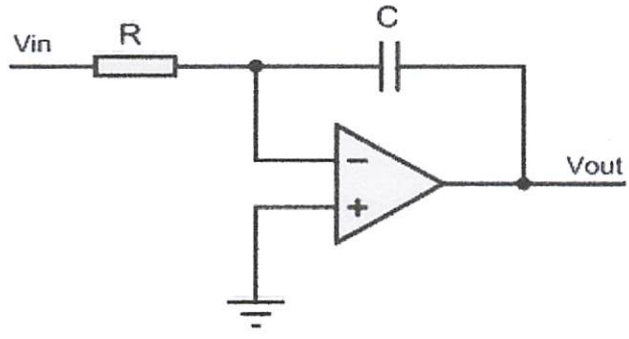

**Gambar 2-19..** Rangkaian Penguat Instrumentasi

Sumber : http://franzaditya.blogspot.com/2009/02/penguat-operasional-op-amp.html

### 2.8.5 Integrator

Merupakan penguat tidak linier yang berfungsi sebagai operator integrasi secara matematik, keluaran dari rangkaian ini menghasilkan tegangan keluaran yang sebanding dengan integrasi masukannya. Rangkaian dan rumus dari *integrator* dapat dilihat di bawah ini:

$$Vout = \frac{1}{R \cdot C} \cdot \int Vin \cdot dt$$



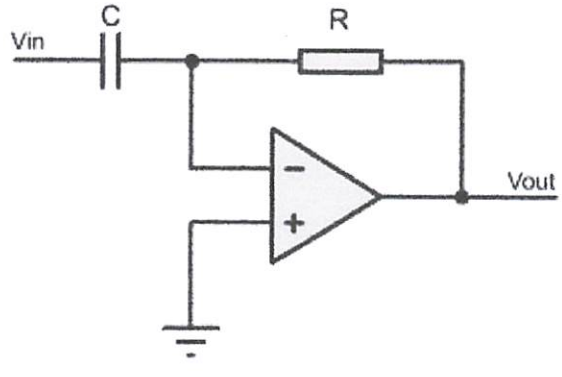

**Gambar 2-20.** Rangkaian Integrator

Sumber : http://franzaditya.blogspot.com/2009/02/penguat-operasional-op-amp.html

### 2.8.6 Diferensiator

Penguat *diferensiator* ini menyediakan operasi matematik dan menghasilkan tegangan keluaran yang sebanding dengan kemiringan tegangan masukannya. Biasanya rangkaian ini digunakan untuk menghasilkan keluaran persegi dari masukan lereng.

$$Vout = R \cdot C \cdot \frac{d(Vin)}{dt}$$




**Gambar 2-21.** Rangkaian Diferensiator

Sumber : http://franzaditya.blogspot.com/2009/02/penguat-operasional-op-amp.html

## 2.9 Transistor

Transistor merupakan salah satu semi konduktor yang dapat dipergunakan untuk perataan arus, menahan sebagian arus, menguatkan arus, membangkitkan frekwensi rendah maupun frekwensi tinggi. Transistor di buat dari bahan semikonduktor, yaitu silikon dan germanium. Transistor yang banyak di gunakan dalam teknik elektronika antara lain transistor PNP ( positive-Negative-Positive) dan transistor NPN ( Negative-Positive-Negative ).

a b

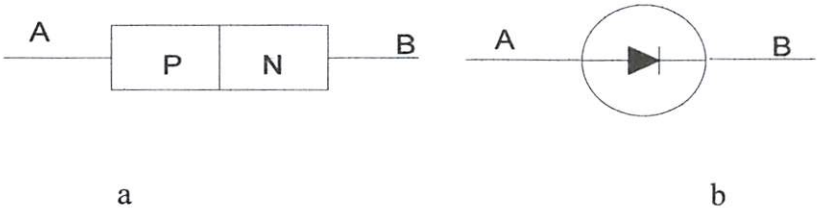
**Gambar 2-22.** *a. Transistor PNP b. Transistor NPN*
Sumber : Teknik dasar-dasar elektronika, 1983

## 2.10 Dioda

Dioda semikonduktor adalah komponen elektronika yang terdiri atas pertemuan tipe P dan semikonduktor N ( P – N junction ). Elektroda yang di hubungkan dengan tipe P disebut anoda, sedangkan yang dihubungkan dengan tipe N disebut katoda.

Dioda dapat berfungsi sebagai saklar elektronik karena dioda hanya akan melewatkan arus dalam satu arah saja dan simbol dioda pada gambar berikut :

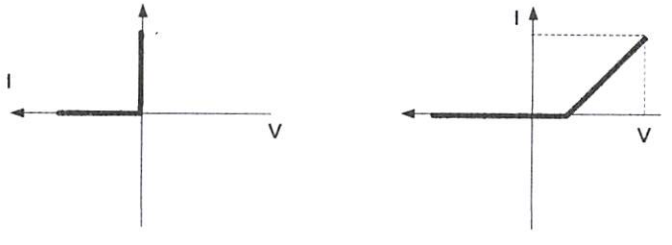
Anoda ( A ) dan Katoda ( K )




a b

**Gambar 2-23** *a kontroksi Dioda* ***Gambar 2.23*** *b syimbol dioda*
Sumber : Teknik dasar-dasar elektronika, 1983

Dioda akan mengalirkan arus ( terjadi konduksi ) bila diberi bias maju, yaitu anoda mendapatkan tegangan positif dan katoda mendapat tegangan negatif, sebaliknya jika diberi reverse bias, maka dioda akan mempuanyai resistansi tinggi, artinya dioda tidak mengalami konduksi. Karena dioda dapat dipergunakan sebagai saklar elektronik mestinya pada saat dioda diberi bias maju akan terjadi aliran arus ( saklar tertutup ) dan apabila diberi bias mundur, maka dioda seperti saklar yang terbuka. Keadaan konduksi dioda jika diberi tegangan maju sebesar 0,7V ( untuk jenis silikon dan 0,2V germanium ). Setelah mencapai tegangan ini ( Knee Voltage ), maka setiap kenaikan arus diikuti kenaikan tegangan, artinya dioda pada saat konduksi mempunyai resistansi tertentu.




**Gambar 2-24.** *a Kurva Dioda Ideal* ***Gambar 2.24*** *b Kurva Pendekatan*

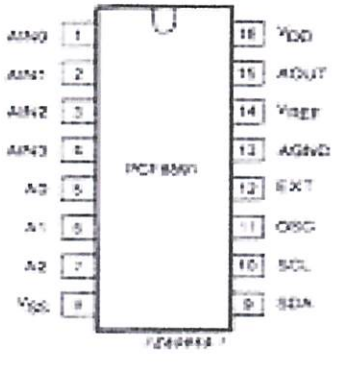
Sumber : Teknik dasar-dasar elektronika, 1983

## 2.11. ADC PCF 8591.

PCF 8591 merupakan chip tunggal buatan *Pilliph Semiconduktor*, mempunyai daya rendah sendiri. Data 8 bit CMOS dengan 4 analog input dan 1 analog output dan 1 bus serial I2C. Tiga alamat pin A0, A!, dan A2 digunakan untuk pemprograman alamat hardware, sehingga memungkinkan penggunaan hingga delapan perangkat yang terhubung ke I2C bus tanpa peranggkat tambahan. Alamat, control dan data ked an dari perangkat ini akan di transfer secara serial memalui dua baris bidirectional I2C bus.

Fungsi perangkat input analog meliputi multiplek, 8 bit analog ke digital dan konversi 8 bit digital ke analog. Maksimum tingkat konversi yang diberikan sesuai dengan kecepatan maksimum I2C bus .Spesifikasi yang dimiliki ADC PCF8991 adalah:

1. Kompatibel penuh dengan 89C52.
2. Hanya perlu 2 jalur kabel utuk interface dengan mikrokontroller.
3. Analog input 4 chanel 8 bit.
4. Analog output 1 chanel 8 bit.
5. Input range tegangan 0 V – 2,5 V.
6. Spesifikasi ADDA PCF8591 adalah 10mV/bit.



**Gambar 2-25.** Pin-pin ADC PCF 8591
Sumber : Datasheet ADDA PCF8591.

Diskripsi pin-pin PCF 8591 :

1. AIN0 : Analog input (A/D converter).
2. AIN1 : Analog input (A/D converter).
3. AIN2 : Analog input (A/D converter).
4. AIN3 : Analog input (A/D converter).
5. A0 : Alamat
6. A1 : Alamat
7. A2 : Alamat
8. Vss : Tegangan suplay negarif.
9. SDA : Data input/output dari bus I2c.
10. SCL : clock input dari bus I2C.
11. OSC : osilator.
12. EXT : Saklar eksternal/internal untuk osilator.
13. AGND : ground.
14. Vref : Tegangan masukan.
15. AOUT : output analog (D/A converter).
16. VDD : Tegangan suplay positif

## 2.12. HP Siemens.

### 2.12.1. Pinout Hp Siemens.

Untuk melekukan koneksing peralatan lain seperti computer dengan mikrokontroller, pada hp Siemens terdapat pinout dengan susunan seperti yang ditunjukkan dalam table 2.10.1. berikut ini:

**Tabel 2-7.** Pinout hp Siemens

| No | Nama | Fungsi | In/Out |
|---|---|---|---|
| 1 | GND | Ground | |
| 2 | SELF SERVICE | Recognition/control baterai | In |
| 3 | LOAD | Charging voltage | Out |
| 4 | BATERAI | Baterai | IN |
| 5 | DATA OUT | Data sent | Out |
| 6 | DATA IN | Data reseived | Out |
| 7 | Z_CLK | Recognition/control accessories | In |
| 8 | Z_DATA | Recognition/control accessories | |
| 9 | MICG | Ground for microphone | |
| 10 | MIC | Mikrophone input | In |
| 11 | AUD | Loadspeaker | |
| 12 | AUDG | Ground for eksternal speaker | Out |

Sumber: www.siemens.com

### 2.12.2. Kabel data Siemens C45

Siemens serie C45 dilengkapi kabel data untuk melakukan komunikasi data secara serial dengan terminal lain. Kecepatan transfer data sebesar 19200. Untuk mengadakan komunikasi serial, pin-pin yang digunakan adalah :

**Tabel 2-8.** Kabel data Siemens

| Nomor Pin | Nama | Fungsi |
|---|---|---|
| 1 | GND | Data ground |
| 5 | TMMS/DFMS – Terminal adaptor equipment from mobile station data mobile station. | Serial data out (out) |
| 6 | TTMS/DTMS – Terminal adaptor equipment to mobile ststion data to mobile ststion | Serial data in (Rx) |

Sumber : www.siemens.com

### 2.12.3. SMS (Short Message Service).

Short Message Service adalah sebuah mekanisme pengiriman pesan singkat melalui jaringan bergerak (*mobile netwok*). Pesan yang dikirim oleh sebuah telephone seluler, ditampung di sentral SMS yang biasanya disebut SMSC *(Sort Message Service Centre)*. Oleh SMSC pesan ini ditransmisikan ke telephone seluler yang dituju.

### 2.12.4. PDU.

Data yang mengalir kea tau dari SMS-Central harus berbentuk PDU. PD berisi bilangan-bilangan heksadesimal yang mencerminkan bahasa I/O. PDU terdiri dari beberapa *header*. *Header* untuk mengirim SMS ke SMS-Central berbeda dengan SMS yang diterima dari SMS-Central.

PUD untuk mengirim SMS terdiri atas 8 *header*, yaitu sebagai berikut :

1. Nomor SMS-Central.
2. Tipe SMS.

3. Nomor referensi SMS.
4. Nomor telephone penerima.
5. Bentuk SMS.
6. Skema enconding data I/O.
7. Jangka waktu, sebelum SMS expired.
8. Isi SMS.

Delapab *header* untuk SMS_terima, adalah sebagai berikut :

1. Nomor SMS-Central.
2. Tipe SMS (untuk SMS_terima : 4 atau 04 dalam heksadesimal).
3. Nomor ponsel pengirim.
4. Bentuk SMS.
5. Skema enconding.
6. Tanggal dan waktu SMS distamp di SMS-Central.
7. Batas waktu validasi, jika tidak dibatasi dilambangkan dengan 00.
8. Isi SMS.

# BAB III

# PERANCANGAN DAN PEMBUATAN ALAT

Perancangan alat yang dibuat dalam skripsi ini meliputi perencanaan perangkat keras dan perencanaan perangkat lunak. Komponen yang dipakai dalam perencanaan ini antara lain mikrokontroller AT89S52 sebagai kontrol utama, dengan komponen pendukung meliputi sensor gas karbon dioksida TGS 4161, RTC DS1307, pengkondisia sinyal, ADC PCF 8591, seven segmen dan dot matrik,dan siemen C45.

Cara kerja alat ini secara garis besar dapat dijelaskan sebagai berikut :

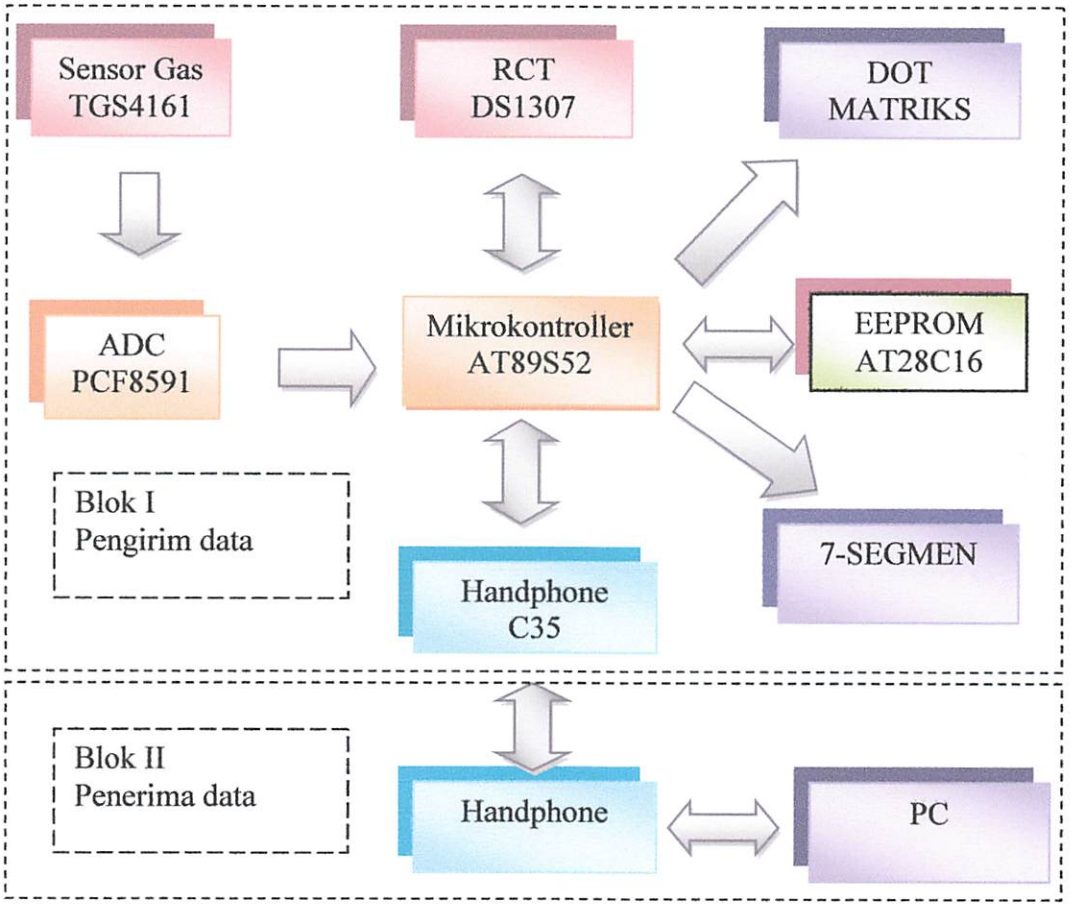
Sensor gas TGS4161 digunakan untuk mendeteksi gas beracun yang terkandung dalam udara yaitu karbon dioksida. Keluaran dari sensor tersebut masih berbentuk sinyal analog dan selanjutnya sinyal analog tersebut menjadi inputan ADC PCF 8591 sehingga keluaran dari ADC PCF 8591 berbentuk sinyal digital (biner) yang dapat diproses oleh mikrokontroller AT89S52. Sinyak digital keluaran dari ADC PCF8591 tersebut diolah oleh mikrokontroller AT89S52 melalui program *software* sehingga data olahan tersebut dapat ditampilkan di display dot matrik.

Sedangkan RTC digunakan untuk me-set jam dan tanggal digital secara otomatis yang langsung dihubungkan ke mikrokontroller AT89S52 dan diproses agar dapat ditampilkan pada display seven segmen.

Pada kedua alat tersebut, mikrokontroller AT89S52 tidak hanya memproses agar dapat ditampilkan pada display dot matrik dan seven segmen saja. Melainkan mikrokontroller AT89S52 juga akan memproses agar dapat dikirimkan ke internet melalui handphone siement C35 menggunakan cara SMS gate way.

### 3.1. Perancangan Sistem

Perencanaan alat pendeteksi gas beracun karbon dioksidapada di jalan raya, dimana dikendalikan oleh mikrokontroller AT89S52 sebagai kendali utama dan menggunakan komponen lain sebagai komponen pendukung. Sebelum membuat perangkat keras terlebih dahulu direncanakan blok diagram yang akan dibuat, dan membahasnya sesuai dengan blok diagram. Disini kita hanya membahas sampai pengiriman data untuk inputan yang ada pada web yaitu pada diagram blok I yang diberi tanda putus-putus, sedangkan blok II yang sebagai penerima data tidak dibahas karena sudah dibahas oleh teman kelompok yang membahas tentang web. Adapun blok diagram yang direncanakan adalah sebagai berikut:




**Gambar 3-1. Blok Diagram Perencanaan**

Dari diagram blok di atas dapat dijelaskan sebagai berikut :

1. **Sensor $CO_2$ *(TGS 4161)***

   Mengubah besaran non eleketris menjadi besaran elektris, dalam hal ini adalah gas karbon dioksida ($CO_2$)

2. **Rangakain pengkondisi sinyal (Op-Amp)**

   Untuk mengkondisikan tegangan dari out put sensor yang bervariasi agar dapat dibaca oleh ADC PCF8591

3. **ADC PCF8591**

   Merubah atau mengkonversikan sinyal analog dari rangkaian pengkondisi sinyal menjadi sinyal digital

4. **Mikrokontroller AT89S52**

   Digunakan sebagai penghitung dan pengolah data serta memprosesnya yang kemudian memberikan instruksi pada dot matriks dan seven segmen

5. **RTC DS1307**

   Digunakan sebagai setting detik, menit, jam, hari, bulan, tahun secara digital.

6. **Dot matriks**

   Akan menampilkan hasil dari proses mikrokontroller AT89S52 tentang kadar gas beracun karbon dioksida dalam udara.

7. **Seven segmen**

   Untuk menampilkan jam dan tanggal secara digital.

8. **Siemen C35**

   Untuk mengirimkan data dari mikrokontroller AT89S52 ke web internet.

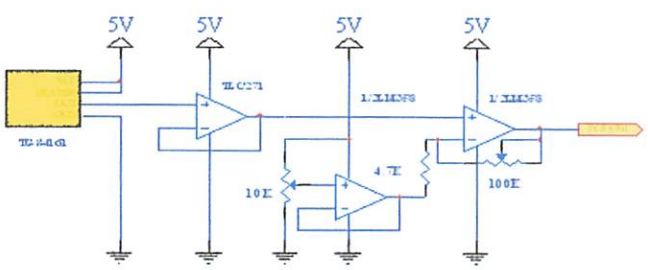
## 3.2. Prinsip Kerja

Pada saat pertama alat dinyalakan, alat akan meminta *inputan setting point* kadar gas beracun, inputan ini sebagai patokan atau batas. Setelah itu sensor karbon dioksida(TGS 4161) akan mendeteksi gas beracun (karbon dioksida), apakah kadar gas beracun yang berbahaya bagi kesehatan, maka mikrokontroller AT89S52 akan mengirimkan data ke dot matrik yang berfungsi sebagai display sehingga orang disekitar alat tersebut dapat melihat data tersebut apakah kadarnya ringan, sedang atau tinggi (berbahaya) kadar karbon dioksida dalam udara disekitar.

## 3.3. Perancangan Sensor

### 3.3.1. Sensor Gas Karbon Dioksida (TGS 4161)

Sensor yang digunakan pada perancangan alat ini untuk medeteksi gas karbon dioksida adalah TGS 4164. Dalam kemampuan pendeteksian gas kondiktifitas sensor bertambah tergantung konsentrasi gas pada udara. Sebuah rangkaian listrik sederhana dapat mengkonversikan perubahan konduktivitas kedalam sebuah signal output yang sesuai dengan kosentrasi gas.




**Gambar 3-2. Rangakaian skematik TGS 4161 / Sensor $CO_2$**

Pada rangkaian TGS4161 diatas, pin 1 dan pin 2 TGS4161 vdi hubungkan ke VCC dengan nilai sebesar +5V, pin 4 dihubungkan ke ground. Pin 3 dihubungkan pada op-amp TLC271 yang berfungsi sebagai memperkuat sinyal karena sinyal dari sensor TGS4161 nilainya kecil. Output dari rangkaian sensor TGS4161 langsung dikirimkan ke ADC PCF8591 yang masih berupa sinyal analog.

### 3.4. Perancangan Rangkaian ADC PCF 8591

Pengubahan analog ke digital mengambil masukan analog melalui kaki AIN 0 dan AIN 1 yang mencupliknya, kemudian mengubah amplitudo dari setiap cuplikan menjadi sandi digital. Keluaran melalui kaki 9 SDA *(Serial Data Line)* yang merupakan jalur pengiriman dan penerimaan data bit dan kaki 10 SCL *(Serial Clock Line)* yang merupakan jalur pengiriman data penerima sinyal *clock.* Keluaran dari kaki SDA dan SCL adalah sejumlah bit bit digital yang status logikanya menunjukkan amplitudonya dari setiap cuplikan. Tegangan yang keluar dari sensor suhu merupakan tegangan analog sehingga akan dikonversikan oleh ADC menjadi data digital yang dibaca oleh mikrokontroller dan tergantung juga terhadap jumlah bit dari ADC yang digunakan. Resolusi ADC menjadi jumlah bit keluarannya, semakin banyak bit keluaran maka makin kecil nilai resolusi yang dihasilkan. Faktor yang terpenting dalam konversi adalah resolusi, kecepatan konversi dan pewaktu. Resolusi yang tinggi di dapat dari tegangan output maksimal (tegangan referensi) dibagi dua pangkat n dan dikurangi 1, dimana n sama dengan jumlah bit , dengan rumus sebagai berikut:

$$\text{Vperstep} = \frac{V\,\text{max}}{2^n - 1}$$

$$= \frac{5}{2^8 - 1}$$

$$= \frac{5}{256-1}$$

$$= \frac{5}{255}$$

$$= 0.0196 \text{ volt}$$

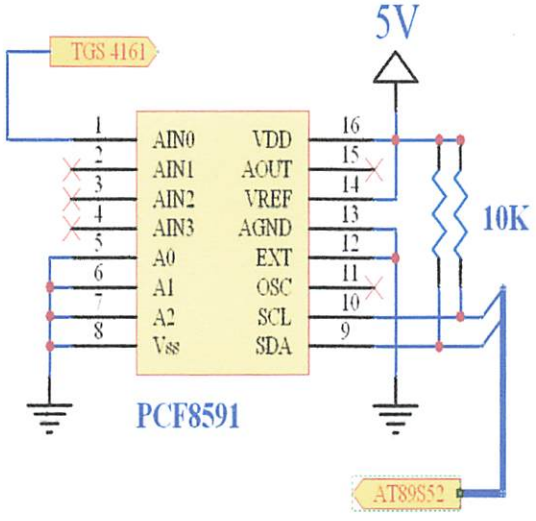
Untuk Vout ADC dapat diketahui yaitu :

$$Vout = \frac{Vanalog}{Vperstep}$$

$$= \frac{3}{0{,}0196}$$

$$= 153{,}06 = 153$$

$$= 10011001 = 99H$$



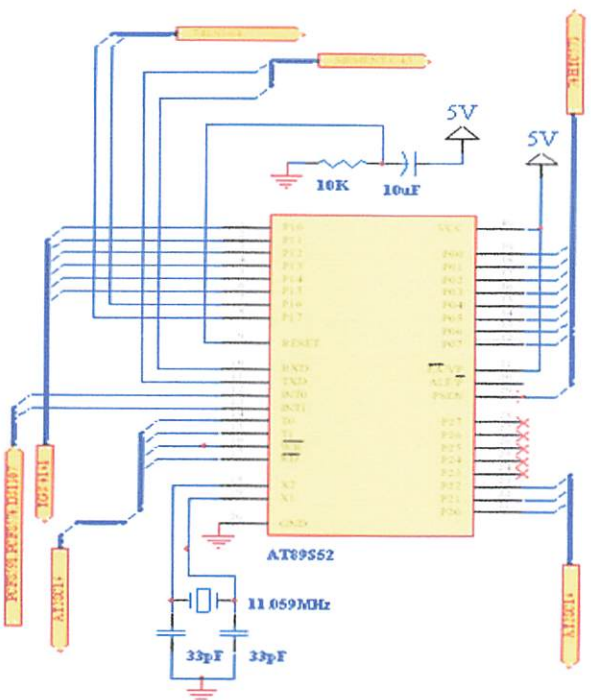

**Gambar 3-3. Rangkaian skematik PCF8591**

**Tabel 3-1. Hubungan antara ADD PCF8591 dengan AT89S52**

| ADDA PCF8591 | AT89S52 |
|---|---|
| SCL | INT0 |
| SDA | INT1 |

### 3.5. Perancangan Rangkaian Mikrokontroller AT59S52

Pengendali mikro merupakan modul utama dalam tugas akhir ini. Rangkaian pengendali mikro ini terdiri dari IC pengendali mikro AT89s52, rangkaian osilator pengendali mikro AT89S53, rangkaian reset pengendali mikro AT89S52.




**Gambar 3-4. Rangkaian skematik mikrokontroller AT89S52**

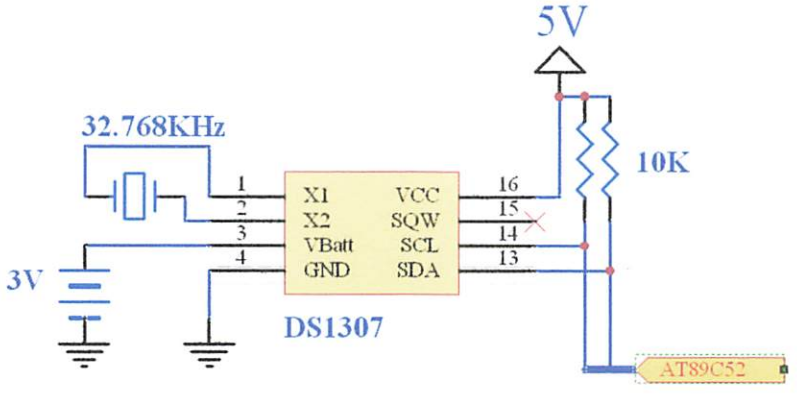
Rangkaian osilator terdiri dari sebuah kristal dan 2 buah kapasitor, rangkaian ini dihubungkan dengan pin XTAL1 dan XTAL2. Nilai kapasitor yang dipakai sebesar 33pF dan kristal yang digunakan memiliki nilai 11,6592 MHz. Rangkaian reset yang direalisasikan memiliki kemampuan power-on reset, yang disertai juga dengan tombl reset. Rangkaian ini terdiri dari sebuah kapasitor, sebuah resistor *push-button.* Nilai kapasitor yang dipakai sebesar 10μF, dan nilai resistor yang dipakai sebesar 10KΩ.

Port INT1 dan INT 2 digunakan sebagai jalur komunikasi antara pengendali mikro dengan RTC. Karena tidak memiliki internal *pull-up* resistor maka harus di *pull-up* terlebih dahulu menggunakan resistor *array* sebesar 10KΩ. Port INT1 dan INT2 tidak hanya dihubungakn ke RTC melainkan juga dihubungkan ke perangkat lain yaitu ADDA PCF89S52 dan I/O expantion dari dot matriks dan 7-segmen.

Port 1.0 (P1.0) sampai Port 1.5 (P1.5) dihubungkan ke seven segmen dan menjadi outputan dari mikrokontroller, port 1.6 (P1.6) dan port 1.7 (P.7) dihubungkan ke 74LS164 yang merupakan driver colom dari display dot matriks. Sedangkan Pin Rx dan Tx dihubungkan pada Hp siemen C45 sebagai komunikasi serial handphone dengan mikrokontroller AT89S52.

### 3.6. Perancangan Rangkaian RTC DS1307

RTC *(Real Time Clock)* merupakan suatu IC yang digunakan sebagai pencatat waktu dari keseluruhan alat. Dan IC ini sudah dilengkapi dengan baterai sendiri, maka bila terjadi pemutusan powes supply, data dari waktu yang telah catat tidak akan hilang dan IC akan tetap mencatat penambahan waktu.



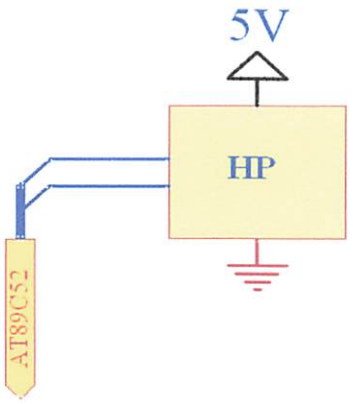

**Gambar 3-5. Rangkaian skematik RTC DS307**

Pin X1 dan X2 dihubungkan dengan sebuah kristal yang besar nilainya 32,768 KHz. Pin Vbat dihubungkan dengan baterai cadangan sebesar 3V. Pin GND

dihubungkan pada ground, sedangkan pin Vcc dihubungkan dengan power supply sebesar +5V. Pin SCL dan SDA dihubungkan dengan mikrokontrollel AT89S52 (pin INT0 dan Pin INT1).

### 3.7. Perancangan Rangkaian komunikasi serial handphone ke mikrokontroller AT98S52.

Mikrokontroller AT98S52 dilengkapi dengan port serial, port serial memungkinkan untuk interface dengan hardware lain dalam format serial. Komunikasi serial antara AT89S52 dengan Hp Siemens C35 hanya tinggal menghubungkan pin-pin serial dari masing-masing hardware. Pin TX dari mikrokontoller AT89S52 akan dihubungkan dengan pin RX yang dad di handphone Siemens C35. Sedangkan pin RX yang ada dimikrokontroller A89S52 dihubungkan ke pin TX yang ada di handphone Siemens C45.

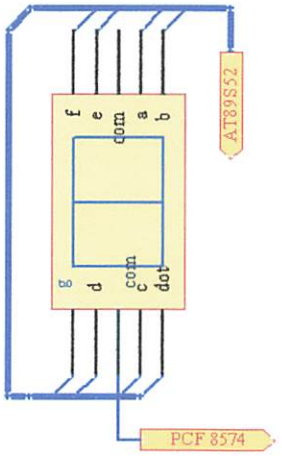



**Gambar 3-6 Rangkaian komunikasi Hp dengan AT89S52.**

### 3.8. Perancangan Rangkaian seven segment.

Display 7 segmen merupakan display yang terbentuk oleh 7 buah LED (ditambah 1 LED untuk menampilkan titik) yang dirangkai untuk menampilkan bilangan heksadesimal. Kedelapan LED tersebut bias dirangkai membentuk rangkaian

common anoda atau common katoda. Pada perencanaan ini yang dipakai adalah 7 segmen common anoda karena jenis ini bisa langsung dihubungkan dengan port mikrokontroller melalui sebuah transistor.

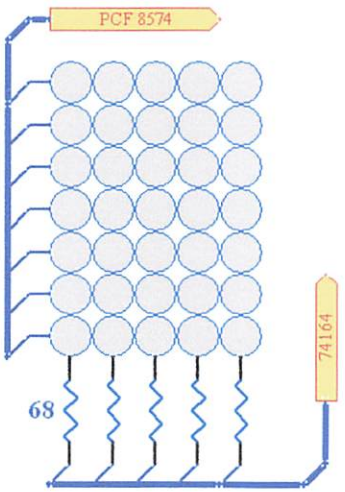



**Gambar 3-7. Rangkaian skematik 7-segment.**

Pada perencanaan ini, seven segmen bisa langsung dihubungkan ke mikrokontroller melainkan ke I/O ekspantion PCF 8574 supaya dapat menghemat pin yang dihubungkan ke mikrokontroller AT89S52. Pin a, b, c, d, e, f, g, h dihubungkan dengan driver kemudian input driver dihubungkan ke mikrokontroller AT89S52 pada port 1.0 (P1.0) sampai port 1.5 (P1.5). sedangkan salah satu dari ground 7-segmen dihubungkan ke PCF8574 sebagai I/O ekspantion (pin P0, P1, P2, P3).

## 3.9. Perancangan Rangkaian dot matriks

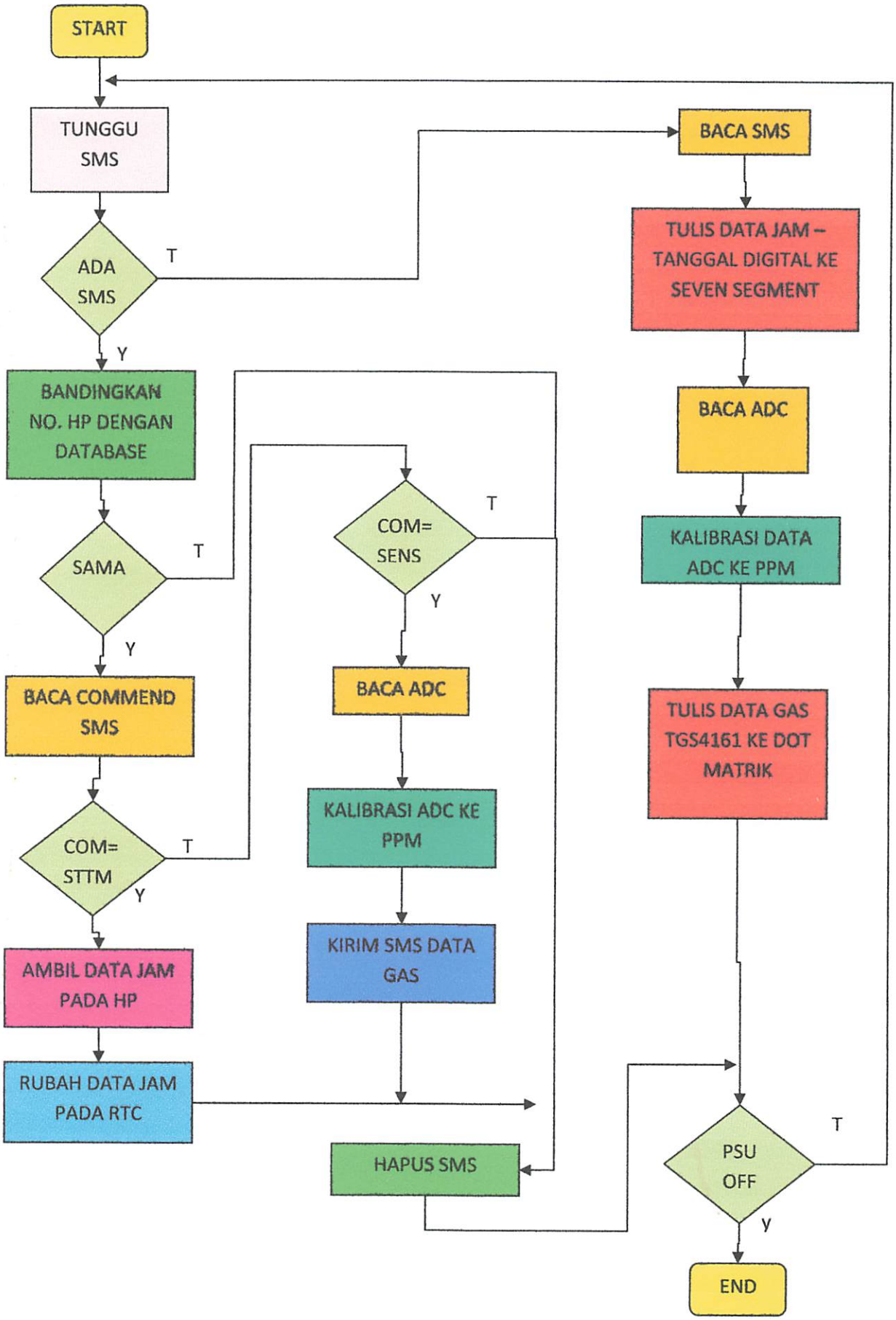
Dot matriks pada perencanaan ini mengunakan dot matriks tipe 5x7, LED warna merah, ukurannya 3 inci. Dot matrik ini digunakan sebagai tampilan dari hasil sensor TGS4161 yang telah diproses oleh mikrokontroller AT89S52. Tampilannya meliputi apakah ringan, sedang dan, tinggi kadar polusi karbon dioksida serta berapa besar nilainya dalam bentuk ppm.




**Gambar 3-8. rangkaian skematik dot matrik**

LED yang berada pada baris dihubungkan dengan I/O expantion PCF8574 supaya dapat menghemat pin yang hubungkan ke mikrokontroller AT89S52 menjadi 2 buah. Sedangkan LED yang berada pada kolom dihubungkan pada register geser 74LS164 yang input juga menjadi 2 buah kemudian dihubungkan pada Mikrokontroller AT89S52 pada port 1.6 (P1.6) dan Port 1.7 (P1.7).

### 3.10. Flowchart



# BAB IV

# PENGUJIAN ALAT

## 4.1. Tujuan

Bab ini akan membahas tentang pengujian alat yang telah dirancang. Adapun tujuan dari pengujian ini adalah untuk mengetahui apakah *hardware* dan *software* dapat bekerja sesuai dengan kondisi yang diinginkan, maka dilakukan pengujian pada alat dan sistem kerja alat, yang mana prosedur pengujian meliputi:

1. Pengujian *Hardware.*
2. Pengujian sistem secara keseluruhan.

## 4.2. Pengujian Perangkat Keras (*Hardware).*

Dalam pengujian alat dibagi dalam beberapa sub sistem dari instrumen dan peralatan, diantaranya adalah :

1. Pengujian Sensor TGS 4161 (sensor CO2)
2. Pengujian ADC PCF 8591
3. Pengujian RTC DS 1307
4. Pengujian 7-segment
5. Pengujian SMS-gate way.
6. Pengujian Keseluruhan Alat

Pengujian perangkat keras ini mencakup pengujian rangkaian elektronika pada masing-masing blok maupun blok secara keseluruhan.

### 4.2.1 Pengujian Sensor TGS 4161

#### 4.2.1.1. Tujuan

Untuk mengetahui berapa kadar gas karbon dioksida ( CO2 ) yang dideteksi oleh sensor TGS 4161.

#### 4.2.1.2. Prosedur pengujian

Pada rangkaian sensor menggunakan op-amp IC TLC 271 sebagai *buffer* atau penyanggah karena impedansi sensor terlalu besar ( > 100 GΩ ). Diketahui sesuai dengan *data sheet* :

ΔEMF = EMF1 – EMF2

EMF1 = EMF pada 350 ppm CO2

EMF2 = EMF pada konsentrasi gas yang terdeteksi

Dari rumus di atas dapat diketahui ( misalkan X = 350 ppm ( range sensor 350 ppm – 8000 ppm ) ):

ΔEMF = EMF1 – EMF2

ΔEMF ( 350 ppm ) [mv] = 350 ppm [mv] – X ppm [mv]

ΔEMF ( 350 ppm ) [mv] = 350 ppm [mv] – 350 ppm [mv]

ΔEMF ( 350 ppm ) [mv] = 0 [mv]

Jadi pada saat 350 ppm ΔEMF sebesar 0 [mv]

1. Jika EMF2 = 455 ppm maka,

ΔEMF = EMF1 – EMF2

= 350 – 455

= -105 ppm

2. Jika EMF2 = 4010 ppm maka,

$\Delta EMF = EMF1 - EMF2$

$= 350 - 4010$

$= -3660$ ppm

**Tabel 4-1.** Data hasil perhitungan ΔEMF

| EMF1 | EMF2 | ΔEMF |
|---|---|---|
| 350 | 350 | 0 |
| 350 | 455 | -105 |
| 350 | 960 | -610 |
| 350 | 1265 | -915 |
| 350 | 1570 | -1220 |
| 350 | 1875 | -1525 |
| 350 | 2180 | -1830 |
| 350 | 2485 | -2135 |
| 350 | 2790 | -2440 |
| 350 | 3095 | -2745 |
| 350 | 3400 | -3050 |
| 350 | 3705 | -3355 |
| 350 | 4010 | -3660 |
| 350 | 4315 | -3965 |
| 350 | 4620 | -4270 |
| 350 | 4925 | -4575 |
| 350 | 5230 | -4880 |
| 350 | 5535 | -5185 |
| 350 | 5840 | -5490 |
| 350 | 6145 | -5795 |
| 350 | 6450 | -6100 |
| 350 | 6755 | -6405 |
| 350 | 7060 | -6710 |
| 350 | 7365 | -7015 |
| 350 | 7670 | -7320 |
| 350 | 7975 | -7625 |

Pada rangkaian sensor ini menggunakan pengkondisi sinyal non inverting yang di ketahui rumusnya :

$$\text{Vout} = (\text{Vin} - \text{Vref})\left(\frac{R2}{R1}+1\right) + \text{Vref}$$

Dari salah satu pengukuran yang dilakukan pada TLC271 diketahui nilai Vin = 750 mV dan Vref = 763,5 mV serta $\left(\frac{R2}{R1}+1\right) = 56{,}56$ x, maka dapat diketahui :

1. $\text{Vout} = (750\text{ V} - 0763.5)\text{ mV x }\left(\frac{R2}{R1}+1\right) + 763.5\text{ mV}$

   Vout = (-13,5) mV x (56.56 x) + 763,5 mV

   Vout = 0 mV

2. Jika Vin = 776.25 mV

   $\text{Vout} = (776.25 - 763{,}5)\text{ mV x }\left(\frac{R2}{R1}+1\right) + 763.5\text{ mV}$

   Vout = (12,75) mV x (56.56 x) + 763.5 mV

   Vout = 1484,64 mV = 1.48 V

3. Jika Vin = 838.9 mV

   $\text{Vout} = (838.9 - 763.5)\text{ mV x}\left(\frac{R2}{R1}+1\right) + 763.5\text{ mV}$

   Vout = (75.4) V x (56,56 x) + 763,5

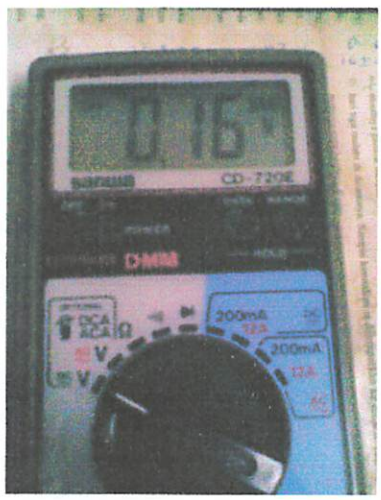
   Vout = 5028,124 mV = 5,03 V

**Tabel 4-2.** Data hasil perhitungan tegangan keluaran sensor Yang telah dikuatkan oleh op-amp

| Vin | Vout perhitungan | PPM |
|---|---|---|
| 750 | 0 | 350 |
| 753.15 | 0.18 | 602 |
| 773.45 | 1.33 | 2360 |
| 785.35 | 1.99 | 3380 |
| 794.45 | 2.52 | 4160 |
| 805.65 | 3.15 | 5150 |
| 826.3 | 4.32 | 6890 |
| 830.85 | 4.58 | 7280 |
| 832.25 | 4.65 | 7400 |
| 838.9 | 5.03 | 7970 |

**Grafik 4-1.** Hasil perhitungan antara PPM dan Vout

Pengukuran tegangan keluaran dari rangkaian sensor (Vout) dapat diukur menggunakan alat ukur digital multimeter dan Hasil pengukuran tegangan keluaran op-amp dan akan menjadi masukan dari ADC PCF 8591.




**Gambar 4-1.** Hasil pengukuran tegangan keluaran sensor

**Gambar 4-2.** Hasil pengukuran tegangan keluaran sensor

**Tabel 4-3.** Data hasil pengukuran tegangan keluaran sensor yang telah dikuatkan oleh op-amp

| Vin | Vout pengukuran | PPM |
|---|---|---|
| 750 | 0 | 350 |
| 753.15 | 0.16 | 602 |
| 773.45 | 1.32 | 2360 |
| 785.35 | 1.98 | 3380 |
| 794.45 | 2.51 | 4160 |
| 805.65 | 3.14 | 5150 |
| 826.3 | 4.32 | 6890 |
| 830.85 | 4.57 | 7280 |
| 832.25 | 4.64 | 7400 |
| 838.9 | 5.01 | 7970 |

**Grafik 4-2.** Hasil pengukuran antara PPM dan Vout yang telah dikuatkan oleh op-amp

**Tabel 4-4.** Perbandingan data hasil perhitungan dan pengukuran tegangan keluaran sensor yang telah dikuatkan oleh op-amp

| Vin | Vout perhitungan | Vout pengukuran | PPM | ERROR |
|---|---|---|---|---|
| 750 | 0 | 0 | 350 | 0 |
| 753.15 | 0.18 | 0.16 | 602 | 11.11 |
| 773.45 | 1.33 | 1.32 | 2360 | 0.75 |
| 785.35 | 1.99 | 1.98 | 3380 | 0.5 |
| 794.45 | 2.52 | 2.51 | 4160 | 0.39 |
| 805.65 | 3.15 | 3.14 | 5150 | 0.32 |
| 826.3 | 4.32 | 4.32 | 6890 | 0 |
| 830.85 | 4.58 | 4.57 | 7280 | 0.22 |
| 832.25 | 4.65 | 4.64 | 7400 | 0.22 |
| 838.9 | 5.03 | 5.01 | 7970 | 0.39 |

Untuk menghitung kesalahan pada percobaan yaitu :

$$Error = \frac{HasilPerhitungan - Hasilpengukuran}{HasilPerhitungan} \times 100\%$$

$$Error = \frac{0.18 - 0.16}{0.18} x100\%$$

$$= \frac{0.02}{0.18} x100\%$$

$$= 11{,}11\%$$

Error rata-rata sebesar :

$$\sum errorrata - rata = \frac{\sum error}{n}$$

$$= \frac{13.9}{10}$$

$$= 1{,}39\%$$

### 4.2.2 Pengujian ADC PFC 8591

#### 4.2.2.1. Tujuan

Untuk mengetahui cara kerja dari rangkaian ADC PCF 8591, apakah sesuai dengan yang direncanakan atau belom maka dilakukan pengujian.

#### 4.2.2.2. Peralatan yang Digunakan

1. Rangkaian ADC PCF 8591
2. Sumber tegangan 5 volt
3. Rangkaian mikrokontroller AT89S8252
4. Resistor 220 Ω
5. LED

#### 4.2.2.3. Prosedur Pengujian

1. Merangkai rangkaian ADC PCF 8591 dengan rangkaian Mikrokontroller AT89S52.
2. Menghubungkan pin 9 (SDA) dan pin 10 (SCL) ADC PCF 8591 dengan rangkaian mikrokontroller AT89S52.
3. Menghubungkan rangkaian pengujian ADC PCF 8591 dengan sumber tegangan 5 volt
4. Ketika ADC itu diberikan clock pada kaki SCL dengan SDA diberi perintah untuk menerjemahkan data konversi analog ke digital secara serial melalui pin SDA ke mikrokontroller AT89S52, kemudian mikrokontroller akan mengeluarkan konversi LED melalui nyala LED
5. Mengamati LED yang menyala dan mati sebagai indikator keluaran 8 bit dari mikrokontroller, dan mencatatat hasilnya pada tabel.

#### 4.2.2.4. Hasil Pengujian

Untuk mengetahui perubahan tegangan tiap bit (Vperstep) ADC adalah dengan menggunakan perhitungan sebagai berikut

Diketahui V = 5 Volt Step ADC = $2^8$ ( 255 ), maka

$$\text{Vperstep} = \frac{V\,\text{max}}{255}$$

$$= \frac{5}{255} = 0{,}0196\ \text{V}$$

**Tabel 4-5.** Hasil Pembacaan Dan Perhitungan ADC PCF 8591

| Input ADC ( Volt ) | Output ADC ( pembacaan ) | Output ADC Perhitungan | | |
|---|---|---|---|---|
| | Biner | Biner | Desimal | Heksa Desimal |
| 0 | 00000000 | 00000000 | 0 | 00h |
| 0.25 | 00001101 | 00001101 | 13 | 0Dh |
| 0.5 | 00011010 | 00011010 | 26 | 1Ah |
| 0.75 | 00100110 | 00100110 | 38 | 26h |
| 1.01 | 00110100 | 00110100 | 52 | 34h |
| 1.26 | 01000000 | 01000000 | 64 | 40h |
| 1.5 | 01001101 | 01001101 | 77 | 4Dh |
| 1.75 | 01011001 | 01011001 | 89 | 59h |
| 2.02 | 01100111 | 01100111 | 103 | 67h |
| 2.25 | 01110011 | 01110011 | 115 | 73h |
| 2.5 | 10000000 | 10000000 | 128 | 80h |
| 2.75 | 10001100 | 10001100 | 140 | 8Ch |
| 3 | 10011001 | 10011001 | 153 | 99h |
| 3.25 | 10100110 | 10100110 | 166 | A6h |
| 3.5 | 10110011 | 10110011 | 179 | B3h |
| 3.75 | 10111111 | 10111111 | 191 | BFh |
| 4 | 11001100 | 11001100 | 204 | CCh |
| 4.25 | 11011001 | 11011001 | 217 | D9h |
| 4.5 | 11100110 | 11100110 | 230 | E6h |
| 4.75 | 11110010 | 11110010 | 242 | F2h |
| 4.99 | 11111111 | 11111111 | 255 | FFh |

Sedangkan untuk mengetahui data tegangan keluaran dari ADC ditentukan dengan perhitungan sebagai berikut:

Diket V analog = 0.25 Volt, maka

$$\text{Vout} = \frac{Vanalog}{Vperstep}$$

$$= \frac{3}{0.0196} = 153.03 = 153$$

$$= 10011001_{(2)}$$

$$= 99_{(16)}$$

### 4.2.3. Pengujian RTC DS 1307

#### 4.2.3.1 Tujuan

Untuk mengetahui bahwa RTC SD 1307bisa bekerja dengan baik serta sesuai dengan jam digital.

#### 4.2.3.1. Alat dan Bahan Yang digunakan

1. Sumber tegangan +5 Volt
2. Modul RTC DS 1307.
3. Rangkaian mikrokontroller AT89S52

#### 4.2.3.3. Langkah Pengujian

1. Merangkai modul RTC DS 1307 dengan mikrokontroller AT89S52
2. Mencatat hasil pengujian dalam table

#### 4.2.3.4 Hasil Pengujian

**Tabel 4.6. Hasil Pengujian Waktu Pada Jam Digital Dengan Waktu Pada Alat**

| No | Waktu Pada Jam Digital | Waktu Pada Alat |
|---|---|---|
| 1 | 07.00.03 | 07.00.03 |
| 2 | 07.30.32 | 07.30.32 |
| 3 | 08.00.21 | 08.00.21 |
| 4 | 08.30.45 | 08.30.45 |
| 5 | 09.00.23 | 09.00.23 |
| 6 | 09.30.06 | 09.30.06 |
| 7 | 10.00.44 | 10.00.44 |

**Tabel 4-7. Hasil Pengujian tanggal Pada Jam Digital Dengan tanggal Pada Alat**

| No | Tanggal,bulan,tahun pada jam digital | Tanggal,bulan,tahun pada jam alat |
|---|---|---|
| 1 | 28/12/2009 | 28.12.09 |
| 2 | 29/12/2009 | 29.12.09 |
| 3 | 30/12/2009 | 30.12.09 |
| 4 | 31/12/2009 | 31.12.09 |
| 5 | 01/01/2010 | 01.01.10 |
| 6 | 02/01/2010 | 02.01.10 |
| 7 | 03/01/2010 | 03.01.10 |

### 4.2.4 Pengujian handphone (SMS gate way)

#### 4.2.4.1 Tujuan

Untuk mengetahui bahwa proses SMS gate way dapat bekerja dengan baik dengan mengirimkan SMS dengan benar.

#### 4.2.4.2 Alat dan bahan.

1. Handpone Siemens C45
2. RTC DS 1307.
3. Rangkaian sensor TGS 4161
4. Rangkaian mikrokontroller AT89S52.

#### 4.2.4.3 Langkah pengujian.

1. Rangkai RTC, sensor TGS 4161, Handphone ke mikrokontroller AT89S52.
2. Kirim SMS ke nomor handphone yang ada di alat.
3. Mencatat hasil dalam table.

#### 4.2.4.4 Hasil pengujian.

1. Men-*setting* jam digital.

**Gambar 4-3.** *Setting* jam digital

Untuk dapat men-*setting* jam dan tanggal pada display maka langkah pertama harus diliat apakah jam dan tanggal pada handphone alat sudah benar. Apabila sudah benar, maka lanjutkan ke langkah yang kedua yaitu mengirimkan SMS ke handphone yang ada pada alat. Yang isi SMS-nya adalah "STTM".

2. Mengirim SMS meminta data sensor TGS 4161

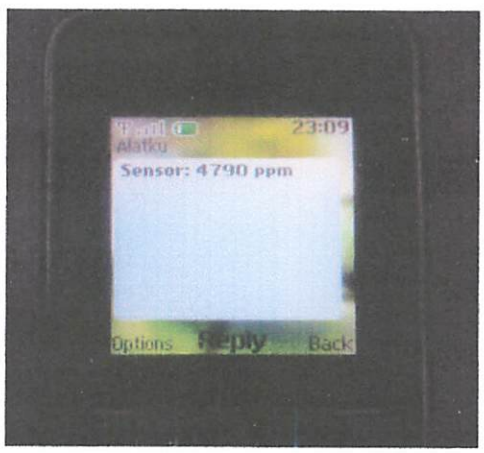
**Gambar 4-4.** SMS meminta data sensor TGS 4161

Untuk mendapatkan data berapa besar kadar gas karbon dioksida pada saat ini, maka kita dapat melakukan langkah-langkah berikut ini. Yang pertama, kita kirimkan SMS ke No. handphone yang ada pada alat. Yang sebelumnya No, pengirim sudah ditulis di database agar tidak semua No. handphone bisa mengakses alat ini. Kemudian isi pesan dengan "SENS". Setelah itu kirimkan ke No. handphone yang ada pada alat. Setelah SMS sudah diterima oleh Handphone yang ada pada alat maka mikrokontroller akan memproses berapa kadar gas karbon dioksida yang dideteksi oleh sensor TGS4161 saat SMS diterima oleh alat.

Setelah mikrokontroller sudah mendapat datanya maka mikrokontroller akan memproses untuk mengirimkan data menggunakan handphone yang ada pada alat sebagai media pengirim pesan balasan.

Contoh : pesan balasannya adalah "Sensor: 4790 ppm".

Pesan balasan dari alat.




**Gambar 4-5.** Pesan Balasan dari alat

**Tabel 4-8.** Data hasil pengujian SMS gate way.

| Pesan SMS | Keterangan |
|---|---|
| STTM | Setting jam digital |
| SENS | Meminta data sensor TGS 4161 |
| Sensor: 4790 | Pesan balasan berupa data sensor TGS 4161 |

### 4.2.5. Pengujian 7-segment

#### 4.2.5.1 Tujuan

Untuk mengetahui bahwa seven segmen bisa bekerja dengan baik dengan menampilkan waktu dan tanggal display seven segmen.

#### 4.2.5.2 Alat dan Bahan Yang digunakan

1. Sumber tegangan +5 Volt
2. RTC DS 1307.
3. Rangkaian mikrokontroller AT89S52
4. 7-segment

#### 4.2.5.3. Langkah Pengujian

1. Merangkai modul RTC DS 1307 dengan mikrokontroller AT89S52
2. Merangkai 7-segment dengan mikrokontroller AT89S52
3. Mencatat hasil pengujian dalam table

#### 4.2.5.4. Hasil Pengujian

**Tabel 4-9.** Hasil pengujian waktu pada jam digital dengan waktu pada alat

| No | Waktu Pada Jam Digital | Waktu Pada Alat |
|---|---|---|
| 1 | 11.00.23 | 11.00.23 |
| 2 | 11.30.09 | 11.30.09 |
| 3 | 12.00.42 | 12.00.42 |
| 4 | 12.30.18 | 12.30.18 |
| 5 | 13.00.21 | 13.00.21 |
| 6 | 13.30.32 | 13.30.32 |
| 7 | 14.00,11 | 14.00.11 |

**Tabel 4-10.** Hasil pengujian tanggal pada jam digital dengan tanggal pada alat

| No | Tanggal,bulan,tahun pada jam digital | Tanggal,bulan,tahun pada jam alat |
|---|---|---|
| 1 | 28/12/2009 | 28.12.09 |
| 2 | 29/12/2009 | 29.12.09 |
| 3 | 30/12/2009 | 30.12.09 |
| 4 | 31/12/2009 | 31.12.09 |
| 5 | 01/01/2010 | 01.01.10 |
| 6 | 02/01/2010 | 02.01.10 |
| 7 | 03/01/2010 | 03.01.10 |

### 4.2.6 Pengujian Sistem Secara Keseluruhan

Pengujian rangkaian secara keseluruhan dilakukan dengan menghubungkan masing-masing rangkaian atau blok dan menjalankan perangkat lunak yang dibuat. Pengujian ini dimaksudkan untuk mengetahui apakah peralatan yang dibuat telah sesuai dengan perencanaan

1. Memberi tegangan catu daya sebesar 12 volt DC, dapat berupa baterai, aki dan lain – lain.
2. Nyalakan saklar *ON/OFF*
3. *Input*kan setting point
4. Berikan inputan CO2 untuk TGS 4161.
5. Mengamati nilai pengukuran pada dot matriks.

#### 4.2.6.1 Hasil pengujian keseluruhan alat

**Tabel 4-11.** Hasil pengujian keseluruhan alat.

| No | Vin (mV) | Vout (mV) | ADC | PPM | Golongan |
|---|---|---|---|---|---|
| 1 | 750 | 0 | 0 | 350 | Rendah |
| 2 | 750.35 | 19.736 | 1 | 380 | Rendah |
| 3 | 753.5 | 197.9 | 10 | 650 | Rendah |
| 4 | 756.3 | 356.268 | 18 | 890 | Rendah |
| 5 | 758.75 | 494.84 | 25 | 1100 | Rendah |
| 6 | 767.15 | 969.944 | 49 | 1820 | Rendah |
| 7 | 779.75 | 1682.6 | 85 | 2900 | Rendah |
| 8 | 780.1 | 1702.396 | 86 | 2930 | Sedang |
| 9 | 783.6 | 1900.356 | 96 | 3230 | Sedang |
| 10 | 789.9 | 2256.684 | 114 | 3770 | Sedang |
| 11 | 793.4 | 2454.644 | 124 | 4070 | Sedang |
| 12 | 799 | 2771.38 | 140 | 4550 | Sedang |
| 13 | 804.25 | 3068.32 | 155 | 5000 | Sedang |
| 14 | 808.1 | 3286.076 | 166 | 5330 | Sedang |
| 15 | 809.15 | 3345.464 | 169 | 5420 | Sedang |
| 16 | 809.5 | 3365.26 | 170 | 5450 | Sedang |
| 17 | 809.85 | 3385.056 | 171 | 5480 | Tinggi |
| 18 | 810.2 | 3404.852 | 172 | 5510 | Tinggi |
| 19 | 820 | 3959.14 | 200 | 6350 | Tinggi |
| 20 | 823.15 | 4137.304 | 209 | 6620 | Tinggi |
| 21 | 831.55 | 4612.408 | 233 | 7340 | Tinggi |
| 22 | 833.65 | 4731.184 | 239 | 7520 | Tinggi |
| 23 | 836.1 | 4869.756 | 246 | 7730 | Tinggi |
| 24 | 838.2 | 4988.532 | 252 | 7910 | Tinggi |
| 25 | 838.9 | 5028.124 | 254 | 7970 | Tinggi |

# BAB V

# PENUTUP

## 5.1. Kesimpulan

Dari pembahasan Perancangan dan pembuatan alat pendeteksi gas karbon diksida dijalan raya, dapat diambil kesimpulan, yaitu :

1. Pada sensor TGS 4161 semakin besar nilai Vin maka nilai PPM akan semakin tinggi.
2. Penguat op-amp untuk sensor TGS4161 adalah sebesar 56,56 kali.
3. Penyimpangan rata rata pada pengukuran dan perhitungan sensor TGS 4161 adalah sebesar 1,39% .
4. Hasil RTC DS1307 yang ditampilkan pada display seven segmen yang berupa jam dan tanggal digital sesuai dengan waktu dan tanggal yang ada pada jam handphone yang ada pada alat.
5. Hasil RTC DS 1307 yang berupa jam digital dan tangal digital yang ditampilkan oleh display seven segmen muncul/tampil secara bergantian sekitar 6 detik.

## 5.2. Saran

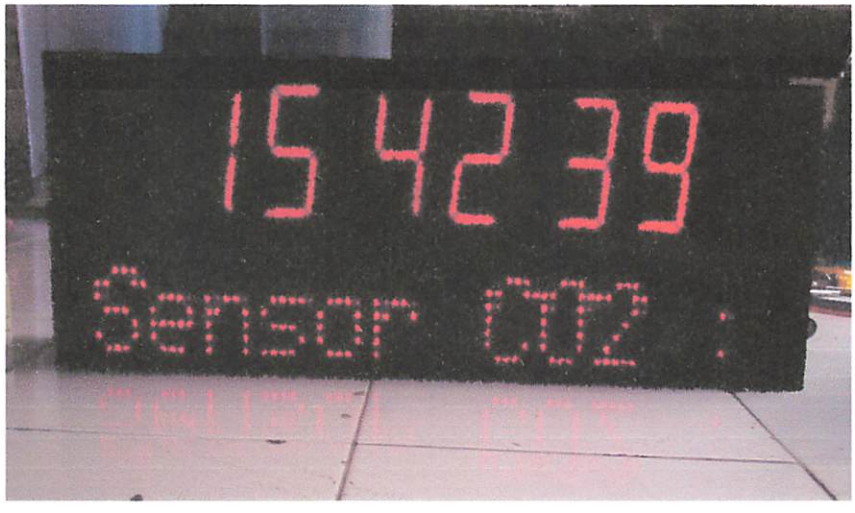
Alat yang dibuat ini masih memiliki keterbatasan, nantinya diharapkan dapat dikembangkan untuk mengatasi keterbatasan itu. Sehingga mendapatkan alat yang diharapkan dapat mendekati alat yang ideal.

1. Bagi peneliti selanjutnya diharapkan bisa mengaplikasikan TGS 4161 pada bidang industri karena sangat bahayanya gas karbon dioksida bagi tubuh.
2. Diharapkan kedepannya bisa menggunakan sensor yang lebih baik dan sensivitasnya lebih besar untuk kadar karbon dioksida udara luar.
3. diharapkan kedepannya dapat dilakukan pengujian dengan membandingkan dengan alat yang sudah ada. Supaya hasilnya dapat lebih akurat.
4. Diharapkan peneliti selanjutnya bisa mengetahui kadar polusi karbon dioksida standar diudara.

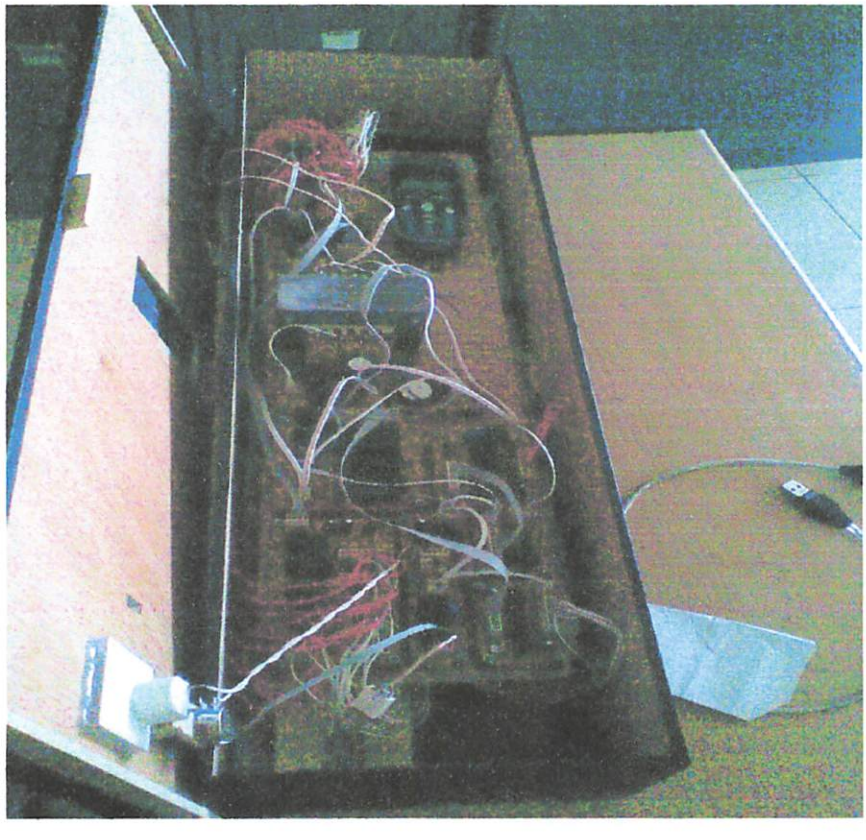
# DAFTAR PUSTAKA

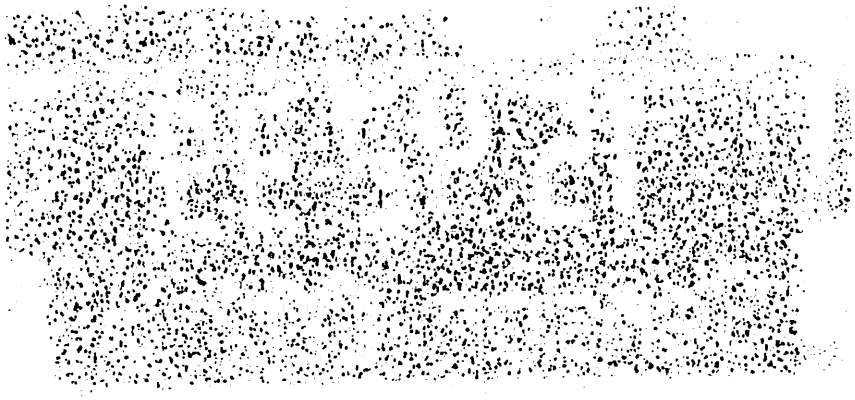

1. Usman, *teknik antarmuka + pemrograman mikrokontroller AT89S52*, penerbit andi, Yogyakarta, 2008
2. *Data sheet* IC AT89S52, (http://www.atmel.com),
3. *Data sheet* TGS 4161, (http://www.figaro.com)
4. *Data sheet ADC PCF 8591,* (http://www.phillipsemiconductor.com)
5. *Data sheet* IC PCF 8574, , (http://www.phillipsemiconductor.com)
6. *Datasheet* RTC DS 1307, (http://www.dalsemi.com)
7. http:/www.polusiudara/karbondioksida.com
8. Wasito S, *Vademekum elektonika edisi kedua,* penerbit PT gramedia pustaka utama, Jakarta, 2001.
9. George Clayton, Steve Winder, *Operational Amplifiers edisi kelima*, Penerbit ERLANGGA, Jakarta, 2003.
10. Siemens, *AT Command Set Reference Manual*, Agustus, 1999.
11. Syam Hardi, Teknik Dasar-Dasar Elektronika, penerbit PT. Bina Aksara, Jakarta,1983.

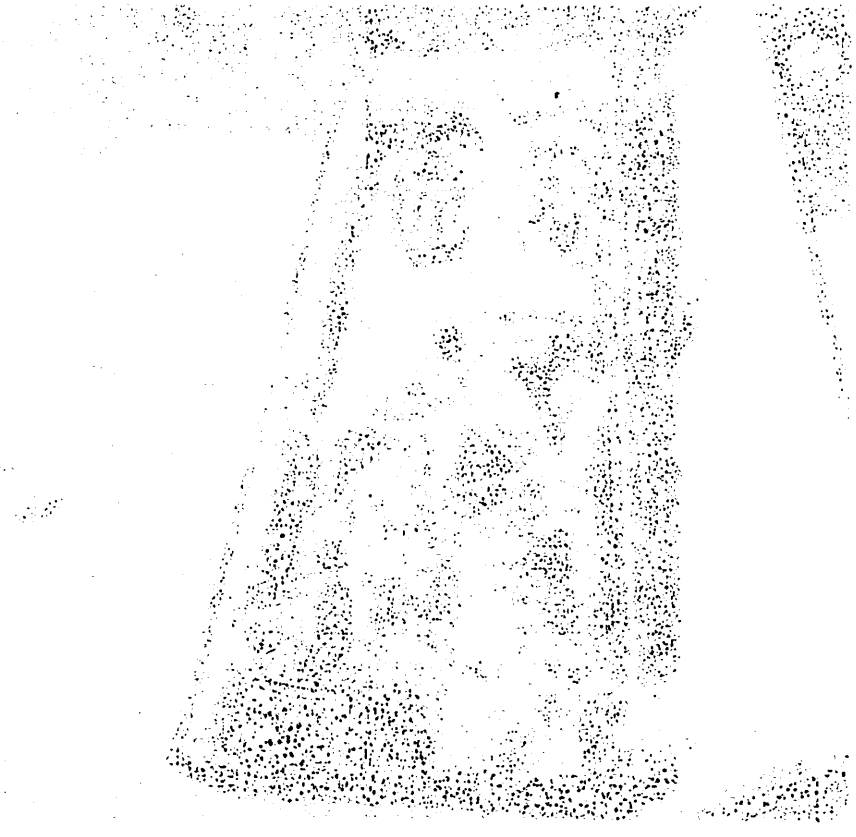
## Foto alat

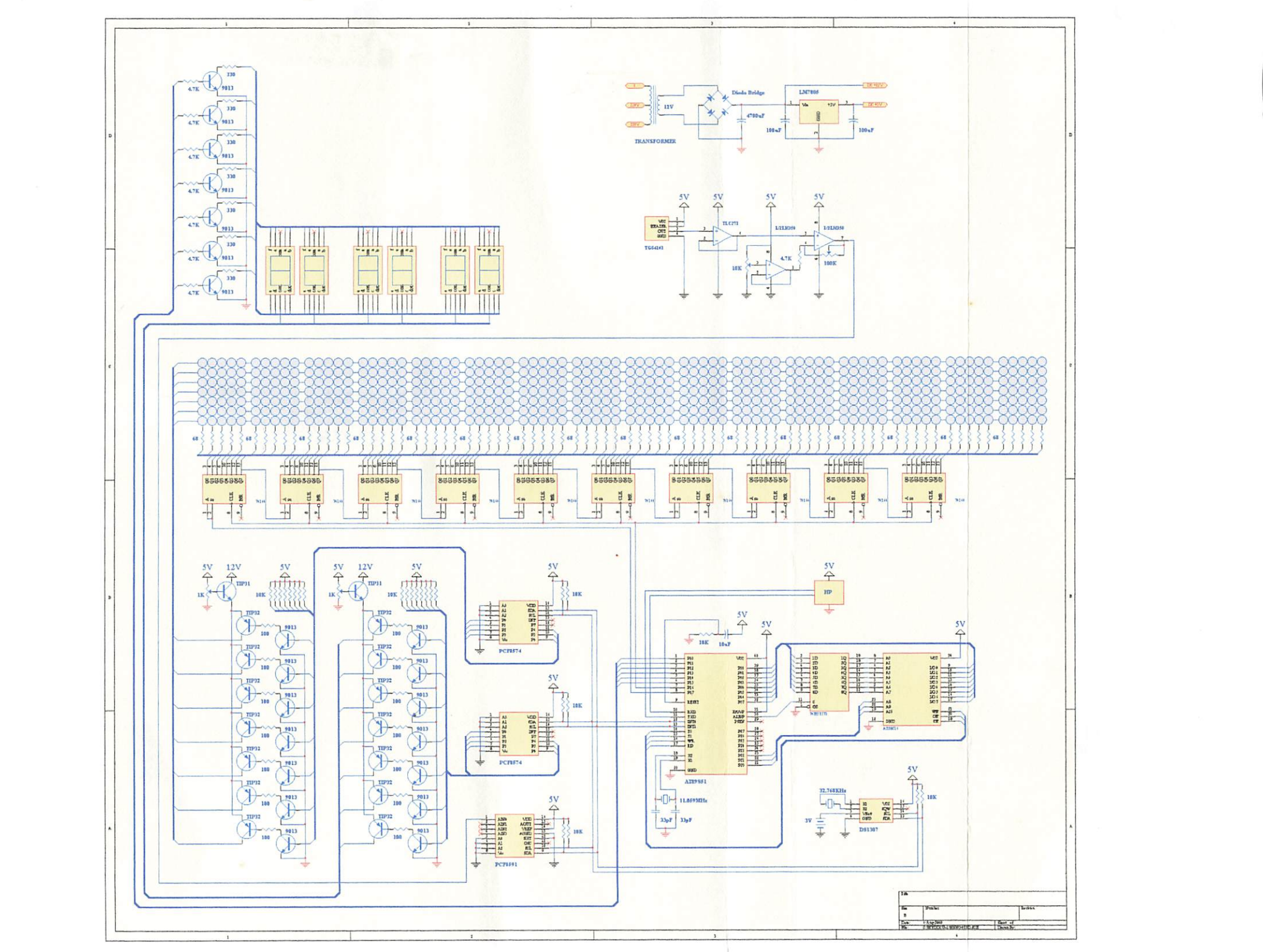


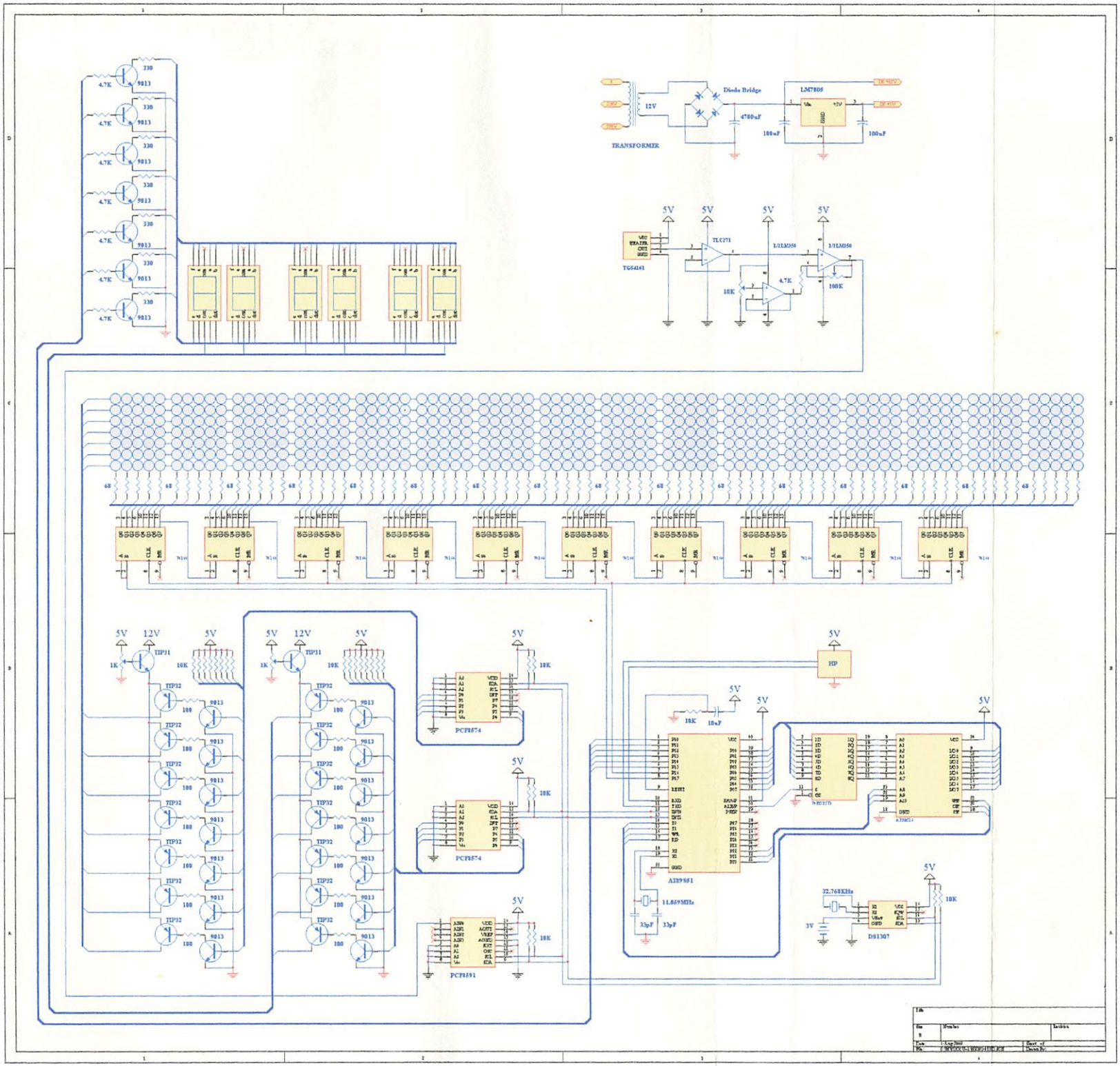

## Foto hardware








TRANSFORMER
Diode Bridge
LM7805
12V
5V
4700uF
100uF
100nF
TL071
LM358
TG54243
4.7K
10K
100K
TIP31
TIP32
9013
1K
100
PCF8574
PCF8591
AT89S51
11.0592MHz
33pF
32.768KHz
3V
DS1307
HP


TRANSFORMER
Diode Bridge
LM7805
TLC271
TLC4541
PCF8574
PCF8591
AT89S51
DS1307
TIP31
TIP32
9013

# LAMPIRAN

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO
JL.Raya Karanglo Km 2
MALANG

# FORMULIR PERBAIKAN UJIAN SKRIPSI

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

| | |
|---|---|
| Nama | : Nanang Setiya Budi |
| Nim | : 05.12.219 |
| Masa Bimbingan | : 01 Desember 2009 s/d 01 Juni 2010 |
| Judul Skripsi | : Aplikasi pengirim data web online pengukur kadar polusi udara di jalan raya menggunakan mikrokontroller AT98S52 dan SMS gate way. |

| No | Tanggal | Uraian | Paraf |
|---|---|---|---|
| 1 | 01-03-2010 | Kesimpulan | |

Malang, Februari 2010
Dosen Penguji I

(Ir. F. Yudi Limpraptono, MT)
NIP. Y. 103 950 0274

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO
JL.Raya Karanglo Km 2
MALANG

# FORMULIR PERBAIKAN UJIAN SKRIPSI

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

Nama : Nanang Setiya Budi
Nim : 05.12.219
Masa Bimbingan : 01 Desember 2009 s/d 01 Juni 2010
Judul Skripsi : Aplikasi pengirim data web online pengukur kadar polusi udara di jalan raya menggunakan mikrokontroller AT98S52 dan SMS gate way.

| No | Tanggal | Uraian | Paraf |
|---|---|---|---|
| 1 | 20-02-2010 | Abstrak | |
| 2 | 20-02-2010 | Batasan masalah : hanya membahas CO2 | |
| 3 | 20-02-2010 | Tabel dan gambar harus ada sumbernya | |
| 4 | 20-02-2010 | Blok diagram tambahkan memori eksternal | |
| 5 | 20-02-2010 | Kesimpulan | |
| 6 | 20-02-2010 | Flowchart | |

Malang, Februari 2010
Dosen Penguji II

(Ir. TH. Mimien Mustikawati, MT)
NIP. P. 103 000 0352

```
        org    00h
        ljmp   init
;
        org    23h
        clr    ES
        jnb    RI,$
        clr    RI
        mov    R7,SBUF
        setb   ES
        reti
;
        ISCL   Bit P3.2          ; I2C clock
        ISDA   Bit P3.3          ; I2C data
        Sclk   Bit P3.4
        Slct   Bit P3.5
        Sdta   Bit P1.7
        Sttc   Bit 20h.0         ; status command
        Sttm   Bit 20h.1         ; status time
        Jchr   Equ 30h                  ; jumlah character ascii
        Jcd0   Equ 31h                  ; jumlah colom dot low
        Jcd1   Equ 32h                  ; jumlah colom dot high
        Cnt0   Equ 33h
        Cnt1   Equ 34h
        Cnt2   Equ 35h
        DpcL   Equ 36h                  ; data pointer character low
        DpcH   Equ 37h                  ; data pointer character high
        DpdL   Equ 38h                  ; data pointer dot low
        DpdH   Equ 39h                  ; data pointer dot high
        Buf0   Equ 3Ah
        Buf1   Equ 3Bh
        Buf2   Equ 3Ch
        Buf3   Equ 3Dh
        Buf4   Equ 3Eh
        Buf5   Equ 3Fh
        Djam   Equ 40h                  ; data jam
        Dmnt   Equ 41h                  ; data menit
        Ddtk   Equ 42h                  ; data detik
        Dtgl   Equ 43h                  ; data tanggal
        Dbln   Equ 44h                  ; data bulan
        Dcen   Equ 45h                  ; data century
        Dthn   Equ 46h                  ; data tahun
        Dadc   Equ 47h
        Dgas   Equ 48h
        Dgs0   Equ 49h
        Dgs1   Equ 4Ah
        Hex0   Equ 4Bh
        Hex1   Equ 4Ch

        Chr0   Equ 50h
        Chr1   Equ 51h
```

**INSTITUT TEKNOLOGI NASIONAL**
Jl. Bendungan Sigura – gura No.2
**MALANG**

# FORMULIR BIMBINGAN SKRIPSI

Nama : Nanang Setiya Budi
Nim : 05.12.219
Masa Bimbingan : 01 Desember 2009 s/d 01 Juni 2010
Judul Skripsi : Aplikasi web online pengukur kadar polusi di jalan raya menggunakan mikrokontroler AT89S52 dan SMS gate way .

| No | Tanggal | Uraian | Paraf Pembimbing |
|---|---|---|---|
| 1 | 15/12 '09 | Revisi Bab I : Batasan masalah | |
| 2 | 23/12 '09 | Revisi Bab II : mikrokontroller AT89S52 & ADC PCF 8591 | |
| 3 | 27/12 '09 | ACC bab I & II ; Revisi bab III : Perhitungan sensor CO2. | |
| 4 | 08/'10 01 | Revisi : Diagram blok & flow chart. | |
| 5 | 20/01 '10 | ACC bab III ; Revisi perhitungan ADC | |
| 6 | 27/01 '10 | Revisi tabel hasil pengukuran kadar $CO_2$ | |
| 7 | 01/02 '10 | ACC bab III & IV. | |
| 8 | 5/2 '10 | Nu kompre | |
| 9 | | | |
| 10 | | | |

Malang,

Dosen pembimbing

**Joseph Dedy Irawan, ST,MT**
**NIP.132315178**

Form S-4b

```
        Chr2    Equ 52h
        Chr3    Equ 53h
        Chr4    Equ 54h
        Chr5    Equ 55h
        Chr6    Equ 56h
        Chr7    Equ 57h
        Chr8    Equ 58h
        Chr9    Equ 59h
        ChrA    Equ 5Ah
        ChrB    Equ 5Bh
        ChrC    Equ 5Ch
        ChrD    Equ 5Dh
        ChrE    Equ 5Eh
        ChrF    Equ 5Fh

        ChrH    Equ 60h
        Cnsm    Equ 61h
        Cndt    Equ 62h
        Cntr    Equ 63h
        Tmo0    Equ 64h
        Tmo1    Equ 65h
        Bufr    Equ 66h

        Dly0    Equ 70h
        Dly1    Equ 71h
        Dly2    Equ 72h
        Dly3    Equ 73h
;
init:   clr     Slct
        lcall   clrdot
        mov     DPTR,#00h
        mov     A,#'S'                  ; 0
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,#'e'                  ; 1
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,#'n'                  ; 2
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,#'s'                  ; 3
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,#'o'                  ; 4
        movx    @DPTR,A
        lcall   wt_wr
```

```
inc   DPTR
mov   A,#'r'              ; 5
movx  @DPTR,A
lcall wt_wr
inc   DPTR
mov   A,#' '              ; 6
movx  @DPTR,A
lcall wt_wr
inc   DPTR
mov   A,#'C'              ; 7
movx  @DPTR,A
lcall wt_wr
inc   DPTR
mov   A,#'O'              ; 8
movx  @DPTR,A
lcall wt_wr
inc   DPTR
mov   A,#'2'              ; 9
movx  @DPTR,A
lcall wt_wr
inc   DPTR
mov   A,#' '              ; 10
movx  @DPTR,A
lcall wt_wr
inc   DPTR
mov   A,#':'              ; 11
movx  @DPTR,A
lcall wt_wr
inc   DPTR
mov   A,#' '              ; 12
movx  @DPTR,A
lcall wt_wr
inc   DPTR
mov   A,#'R'              ; 13
movx  @DPTR,A
lcall wt_wr
inc   DPTR
mov   A,#'e'              ; 14
movx  @DPTR,A
lcall wt_wr
inc   DPTR
mov   A,#'n'              ; 15
movx  @DPTR,A
lcall wt_wr
inc   DPTR
mov   A,#'d'              ; 16
movx  @DPTR,A
lcall wt_wr
inc   DPTR
mov   A,#'a'              ; 17
```

```
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,#'h'                ; 18
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,#','                ; 19
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,#' '                ; 20
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,#'0'                ; 21
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,#'0'                ; 22
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,#'0'                ; 23
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,#'0'                ; 24
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,#' '                ; 25
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,#'p'                ; 26
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,#'p'                ; 27
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,#'m'                ; 28
        movx   @DPTR,A
        lcall  wt_wr
;
        lcall  srl_in                ; inisialisasi serial
        lcall  noecho                ; komunikasi no-echo
        lcall  dsminc                ; display incoming sms
```

```
;      lcall  rstcmd                ; reset command
;
mulai: mov    Jchr,#29              ; jumlah character ascii
       mov    DpcL,#00h             ; ambil dari data pointer
       mov    DpcH,#00h             ; address 0000h
       lcall  cchchr                ; cacah character
loop0: mov    Jcd0,#254             ; jalankan terus menerus
       mov    Jcd1,#000             ; dalam 254 colom dot matrix led
       lcall  tljln1                ; oke
;
loop1: lcall  bc_gas                ;\
       mov    A,Dgas                ; |
       mov    B,#50                 ; | if kurang dari 1/3
       div    AB              ; |
       jnz    loop2                 ;/
       mov    DPTR,#013             ;\
       mov    A,#'R'                ; |
       movx   @DPTR,A                    ; |
       lcall  wt_wr                 ; |
       inc    DPTR                  ; |
       mov    A,#'e'                ; |
       movx   @DPTR,A                    ; |
       lcall  wt_wr                 ; |
       inc    DPTR                  ; |
       mov    A,#'n'                ; |
       movx   @DPTR,A                    ; |
       lcall  wt_wr                 ; |
       inc    DPTR                  ; |
       mov    A,#'d'                ; | tulis rendah
       movx   @DPTR,A                    ; |
       lcall  wt_wr                 ; |
       inc    DPTR                  ; |
       mov    A,#'a'                ; |
       movx   @DPTR,A                    ; |
       lcall  wt_wr                 ; |
       inc    DPTR                  ; |
       mov    A,#'h'                ; |
       movx   @DPTR,A                    ; |
       lcall  wt_wr                 ; |
       ljmp   loop4                 ;/
;
loop2: mov    A,Dgas                        ;\
       mov    B,#100                ; | if antara
       div    AB              ; | 1/3 - 2/3
       jnz    loop3                 ;/
       mov    DPTR,#013             ;\
       mov    A,#'S'                ; |
       movx   @DPTR,A                    ; |
       lcall  wt_wr                 ; |
       inc    DPTR                  ; |
```

```
        mov    A,#'e'              ;|
        movx   @DPTR,A                 ;|
        lcall  wt_wr               ;|
        inc    DPTR                ;|
        mov    A,#'d'              ;|
        movx   @DPTR,A                 ;|
        lcall  wt_wr               ;|
        inc    DPTR                ;|
        mov    A,#'a'              ;| tulis sedang
        movx   @DPTR,A                 ;|
        lcall  wt_wr               ;|
        inc    DPTR                ;|
        mov    A,#'n'              ;|
        movx   @DPTR,A                 ;|
        lcall  wt_wr               ;|
        inc    DPTR                ;|
        mov    A,#'g'              ;|
        movx   @DPTR,A                 ;|
        lcall  wt_wr               ;|
        ljmp   loop4               ;/
;
loop3:  mov    DPTR,#013           ;\
        mov    A,#'T'              ;|
        movx   @DPTR,A                 ;|
        lcall  wt_wr               ;|
        inc    DPTR                ;|
        mov    A,#'i'              ;|
        movx   @DPTR,A                 ;|
        lcall  wt_wr               ;|
        inc    DPTR                ;|
        mov    A,#'n'              ;|
        movx   @DPTR,A                 ;|
        lcall  wt_wr               ;|
        inc    DPTR                ;| if lebih dari 2/3
        mov    A,#'g'              ;| tulis tinggi
        movx   @DPTR,A                 ;|
        lcall  wt_wr               ;|
        inc    DPTR                ;|
        mov    A,#'g'              ;|
        movx   @DPTR,A                 ;|
        lcall  wt_wr               ;|
        inc    DPTR                ;|
        mov    A,#'i'              ;|
        movx   @DPTR,A                 ;|
        lcall  wt_wr               ;/
;
loop4:  mov    DpcL,#013           ; ambil dari data pointer
        mov    DpcH,#000           ; address 13
        lcall  cchrng              ; cacah range
;
```

```
        mov     DPTR,#lokup0            ;\
        mov     A,Dgas                  ; |
        movc    A,@A+DPTR               ; |
        mov     Dgs0,A                  ; |
        mov     DPTR,#lokup1            ; | kalibrasi tabel
        mov     A,Dgas                  ; |
        movc    A,@A+DPTR               ; |
        mov     Dgs1,A                  ; |
        mov     Hex0,Dgs0               ; |
        mov     Hex1,Dgs1               ; |
        lcall   hexdec                  ;/
        mov     DPTR,#angka             ;\
        mov     A,R3                    ; |
        movc    A,@A+DPTR               ; |
        mov     R3,A                    ; |
        mov     A,R2                    ; |
        movc    A,@A+DPTR               ; | rubah nilai -> character
        mov     R2,A                    ; |
        mov     A,R1                    ; |
        movc    A,@A+DPTR               ; |
        mov     R1,A                    ; |
        mov     A,R0                    ; |
        movc    A,@A+DPTR               ; |
        mov     R0,A                    ;/
;
        mov     DPTR,#021               ;\
        mov     A,R3                    ; |
        movx    @DPTR,A                         ; |
        lcall   wt_wr                   ; |
        inc     DPTR                    ; |
        mov     A,R2                    ; |
        movx    @DPTR,A                         ; |
        lcall   wt_wr                   ; |
        inc     DPTR                    ; | simpan memory
        mov     A,R1                    ; |
        movx    @DPTR,A                         ; |
        lcall   wt_wr                   ; |
        inc     DPTR                    ; |
        mov     A,R0                    ; |
        movx    @DPTR,A                         ; |
        lcall   wt_wr                   ;/
        mov     DpcL,#021               ; ambil dari data pointer
        mov     DpcH,#000               ; address 21
        lcall   cchnil                  ; cacah nilai
;
        cjne    R7,#0FFh,loop5
        ljmp    loop6
loop5:  lcall   delay2
        lcall   bc_sms
        lcall   rstcmd
```

```
      mov    SP,#07h
loop6: ljmp   loop0
;
clock: setb   Sclk
      clr    Sclk
      ret
;
tljln1: mov    DpdL,#100          ; jalan terus
      mov    DpdH,#000
      mov    Cnt1,#0
      mov    Cnt2,#1
tlj110: lcall  bc_rtc             ;\
      jb     Sttm,tpljam
      mov    R0,Djam            ; |
      mov    R1,Dmnt            ; |
      mov    R2,Ddtk            ;/
tpljam: jnb    Sttm,tpltgl
      mov    R0,Dtgl            ; |
      mov    R1,Dbln            ; |
      mov    R2,Dthn            ;/
tpltgl: mov    DPL,DpdL
      mov    DPH,DpdH
      mov    Cnt0,#80
      lcall  tulis0
      mov    DPL,DpdL
      mov    DPH,DpdH
      mov    Cnt0,#80
      lcall  tulis1
      mov    DPL,DpdL
      mov    DPH,DpdH
      mov    Cnt0,#80
      lcall  tulis2
      mov    DPL,DpdL
      mov    DPH,DpdH
      mov    Cnt0,#80
      lcall  tulis3
      mov    DPL,DpdL
      mov    DPH,DpdH
      mov    Cnt0,#80
      lcall  tulis4
      mov    DPL,DpdL
      mov    DPH,DpdH
      mov    Cnt0,#80
      lcall  tulis5
      mov    DPL,DpdL
      mov    DPH,DpdH
      mov    Cnt0,#80
      lcall  tulis6
      djnz   Cnt2,tlj112
      mov    Cnt2,#1
```

```
        lcall   gsrdot
        mov     A,Jcd1
        jz      tljl11
        djnz    Cnt1,tljl12
        djnz    Jcd1,tljl12
tljl11: mov     A,Jcd0
        jz      tljl15
        djnz    Jcd0,tljl12
        ljmp    tljl15
tljl12: ljmp    tljl10
tljl15: jb      Sttm,tljl16
        setb    Sttm
        ljmp    tljl17
tljl16: jnb     Sttm,tljl16
        clr     Sttm
tljl17: ret
;
gsrdot: inc     DpdL
        mov     A,DpdL
        jnz     gsrdt
        inc     DpdH
gsrdt:  ret
;
tulis0: movx    A,@DPTR
        mov     C,Acc.1
        cpl     C
        mov     Sdta,C
        lcall   clock
        inc     DPTR
        djnz    Cnt0,tulis0
        mov     A,#72h                  ;\
        lcall   adrtx                   ; |
        mov     A,#00010000b            ; | nyalakan
        lcall   dtatx                   ; |
        lcall   i2cstp                  ;/
        mov     DPTR,#angka0
        mov     A,R0                    ;\
        mov     B,#10                   ; |
        div     AB              ; | ambil puluhan
        movc    A,@A+DPTR               ; |
        mov     P1,A                    ;/
        mov     A,#70h                  ;\
        lcall   adrtx                   ; |
        mov     A,#00000100b            ; | nyalakan
        lcall   dtatx                   ; |
        lcall   i2cstp                  ;/
        lcall   delay1                  ; wait
        mov     A,#72h                  ;\
        lcall   adrtx                   ; |
        mov     A,#00000000b            ; | matikan
```

```
        lcall  dtatx                    ;|
        lcall  i2cstp                   ;/
        mov    A,#70h                   ;\
        lcall  adrtx                    ;|
        mov    A,#00000000b             ;| matikan
        lcall  dtatx                    ;|
        lcall  i2cstp                   ;/
        ret
;
tulis1: movx   A,@DPTR
        mov    C,Acc.2
        cpl    C
        mov    Sdta,C
        lcall  clock
        inc    DPTR
        djnz   Cnt0,tulis1
        mov    A,#72h                   ;\
        lcall  adrtx                    ;|
        mov    A,#00100000b             ;| nyalakan
        lcall  dtatx                    ;|
        lcall  i2cstp                   ;/
        mov    DPTR,#angka0
        mov    A,R0                     ;\
        mov    B,#10                    ;|
        div    AB                 ;| ambil
        mov    A,B                      ;| satuan
        movc   A,@A+DPTR                ;|
        mov    P1,A                     ;/
        mov    A,#70h                   ;\
        lcall  adrtx                    ;|
        mov    A,#00001000b             ;| nyalakan
        lcall  dtatx                    ;|
        lcall  i2cstp                   ;/
        lcall  delay1                   ; wait
        mov    A,#72h                   ;\
        lcall  adrtx                    ;|
        mov    A,#00000000b             ;| matikan
        lcall  dtatx                    ;|
        lcall  i2cstp                   ;/
        mov    A,#70h                   ;\
        lcall  adrtx                    ;|
        mov    A,#00000000b             ;| matikan
        lcall  dtatx                    ;|
        lcall  i2cstp                   ;/
        ret
;
tulis2: movx   A,@DPTR
        mov    C,Acc.3
        cpl    C
        mov    Sdta,C
```

```
        lcall   clock
        inc     DPTR
        djnz    Cnt0,tulis2
        mov     A,#72h                  ;\
        lcall   adrtx                   ; |
        mov     A,#01000000b            ; | nyalakan
        lcall   dtatx                   ; |
        lcall   i2cstp                  ;/
        mov     DPTR,#angka0
        mov     A,R1                    ;\
        mov     B,#10                   ; |
        div     AB              ; | ambil puluhan
        movc    A,@A+DPTR               ; |
        mov     P1,A                    ;/
        mov     A,#70h                  ;\
        lcall   adrtx                   ; |
        mov     A,#10000000b            ; | nyalakan
        lcall   dtatx                   ; |
        lcall   i2cstp                  ;/
        lcall   delay1                  ; wait
        mov     A,#72h                  ;\
        lcall   adrtx                   ; |
        mov     A,#00000000b            ; | matikan
        lcall   dtatx                   ; |
        lcall   i2cstp                  ;/
        mov     A,#70h                  ;\
        lcall   adrtx                   ; |
        mov     A,#00000000b            ; | matikan
        lcall   dtatx                   ; |
        lcall   i2cstp                  ;/
        ret
;
tulis3: movx    A,@DPTR
        mov     C,Acc.4
        cpl     C
        mov     Sdta,C
        lcall   clock
        inc     DPTR
        djnz    Cnt0,tulis3
        mov     A,#72h                  ;\
        lcall   adrtx                   ; |
        mov     A,#10000000b            ; | nyalakan
        lcall   dtatx                   ; |
        lcall   i2cstp                  ;/
        mov     DPTR,#angka0
        mov     A,R1                    ;\
        mov     B,#10                   ; |
        div     AB              ; | ambil
        mov     A,B                     ; | satuan
        movc    A,@A+DPTR               ; |
```

```
        mov    P1,A                ;/
        mov    A,#70h              ;\
        lcall  adrtx               ; |
        mov    A,#01000000b        ; | nyalakan
        lcall  dtatx               ; |
        lcall  i2cstp              ;/
        lcall  delay1              ; wait
        mov    A,#72h              ;\
        lcall  adrtx               ; |
        mov    A,#00000000b        ; | matikan
        lcall  dtatx               ; |
        lcall  i2cstp              ;/
        mov    A,#70h              ;\
        lcall  adrtx               ; |
        mov    A,#00000000b        ; | matikan
        lcall  dtatx               ; |
        lcall  i2cstp              ;/
        ret
;
tulis4: movx   A,@DPTR
        mov    C,Acc.5
        cpl    C
        mov    Sdta,C
        lcall  clock
        inc    DPTR
        djnz   Cnt0,tulis4
        mov    A,#72h              ;\
        lcall  adrtx               ; |
        mov    A,#00001000b        ; | nyalakan
        lcall  dtatx               ; |
        lcall  i2cstp              ;/
        mov    DPTR,#angka0
        mov    A,R2                ;\
        mov    B,#10               ; |
        div    AB                  ; | ambil puluhan
        movc   A,@A+DPTR           ; |
        mov    P1,A                ;/
        mov    A,#70h              ;\
        lcall  adrtx               ; |
        mov    A,#00100000b        ; | nyalakan
        lcall  dtatx               ; |
        lcall  i2cstp              ;/
        lcall  delay1              ; wait
        mov    A,#72h              ;\
        lcall  adrtx               ; |
        mov    A,#00000000b        ; | matikan
        lcall  dtatx               ; |
        lcall  i2cstp              ;/
        mov    A,#70h              ;\
        lcall  adrtx               ; |
```

```
        mov    A,#00000000b          ; | matikan
        lcall  dtatx                 ; |
        lcall  i2cstp                ;/
        ret
;
tulis5: movx   A,@DPTR
        mov    C,Acc.6
        cpl    C
        mov    Sdta,C
        lcall  clock
        inc    DPTR
        djnz   Cnt0,tulis5
        mov    A,#72h                ;\
        lcall  adrtx                 ; |
        mov    A,#00000100b          ; | nyalakan
        lcall  dtatx                 ; |
        lcall  i2cstp                ;/
        mov    DPTR,#angka0
        mov    A,R2                  ;\
        mov    B,#10                 ; |
        div    AB                ; | ambil
        mov    A,B                   ; | satuan
        movc   A,@A+DPTR             ; |
        mov    P1,A                  ;/
        mov    A,#70h                ;\
        lcall  adrtx                 ; |
        mov    A,#00010000b          ; | nyalakan
        lcall  dtatx                 ; |
        lcall  i2cstp                ;/
        lcall  delay1                ; wait
        mov    A,#72h                ;\
        lcall  adrtx                 ; |
        mov    A,#00000000b          ; | matikan
        lcall  dtatx                 ; |
        lcall  i2cstp                ;/
        mov    A,#70h                ;\
        lcall  adrtx                 ; |
        mov    A,#00000000b          ; | matikan
        lcall  dtatx                 ; |
        lcall  i2cstp                ;/
        ret
;
tulis6: movx   A,@DPTR
        mov    C,Acc.7
        cpl    C
        mov    Sdta,C
        lcall  clock
        inc    DPTR
        djnz   Cnt0,tulis6
        mov    A,#72h                ;\
```

```
      lcall  adrtx                ; |
      mov    A,#00000010b         ; | nyalakan
      lcall  dtatx                ; |
      lcall  i2cstp               ;/
      lcall  delay1               ; wait
      mov    A,#72h               ;\
      lcall  adrtx                ; |
      mov    A,#00000000b         ; | matikan
      lcall  dtatx                ; |
      lcall  i2cstp               ;/
      ret
;
cchchr: mov    DpdL,#180          ; cacah character
      mov    DpdH,#000
      mov    DPL,DpcL
      mov    DPH,DpcH
cchch: movx    A,@DPTR
      lcall  chrdot
      mov    DPL,DpcL
      mov    DPH,DpcH
      inc    DPTR
      mov    DpcL,DPL
      mov    DpcH,DPH
      djnz   Jchr,cchch
      ret
;
cchrng: mov    Jchr,#6                  ; jumlah character ascii
      mov    DpdL,#002          ; start = 180 + (13chr * 6dot)
      mov    DpdH,#001          ;       = 180 + 78
      mov    DPL,DpcL           ;       = 258
      mov    DPH,DpcH
ccrng: movx    A,@DPTR
      lcall  chrdot
      mov    DPL,DpcL
      mov    DPH,DpcH
      inc    DPTR
      mov    DpcL,DPL
      mov    DpcH,DPH
      djnz   Jchr,ccrng
      ret
;
cchnil: mov    Jchr,#4            ; jumlah character ascii
      mov    DpdL,#050          ; start = 180 + (21chr * 6dot)
      mov    DpdH,#001          ;       = 180 + 126
      mov    DPL,DpcL           ;       = 306
      mov    DPH,DpcH
ccnil: movx    A,@DPTR
      lcall  chrdot
      mov    DPL,DpcL
      mov    DPH,DpcH
```

```
        inc    DPTR
        mov    DpcL,DPL
        mov    DpcH,DPH
        djnz   Jchr,ccnil
        ret
;
chrdot: mov    DPTR,#chrksg
        cjne   A,#'A',chdt00
        mov    DPTR,#charAb
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt00: cjne   A,#'B',chdt01
        mov    DPTR,#charBb
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt01: cjne   A,#'C',chdt02
        mov    DPTR,#charCb
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt02: cjne   A,#'D',chdt03
        mov    DPTR,#charDb
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt03: cjne   A,#'E',chdt04
        mov    DPTR,#charEb
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt04: cjne   A,#'F',chdt05
        mov    DPTR,#charFb
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt05: cjne   A,#'G',chdt06
        mov    DPTR,#charGb
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt06: cjne   A,#'H',chdt07
        mov    DPTR,#charHb
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt07: cjne   A,#'I',chdt08
        mov    DPTR,#charIb
        lcall  ambdta
```

```
        lcall   tlsmem
        ljmp    chdt99
chdt08: cjne    A,#'J',chdt09
        mov     DPTR,#charJb
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt09: cjne    A,#'K',chdt10
        mov     DPTR,#charKb
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt10: cjne    A,#'L',chdt11
        mov     DPTR,#charLb
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt11: cjne    A,#'M',chdt12
        mov     DPTR,#charMb
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt12: cjne    A,#'N',chdt13
        mov     DPTR,#charNb
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt13: cjne    A,#'O',chdt14
        mov     DPTR,#charOb
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt14: cjne    A,#'P',chdt15
        mov     DPTR,#charPb
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt15: cjne    A,#'Q',chdt16
        mov     DPTR,#charQb
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt16: cjne    A,#'R',chdt17
        mov     DPTR,#charRb
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt17: cjne    A,#'S',chdt18
        mov     DPTR,#charSb
        lcall   ambdta
```

```
        lcall   tlsmem
        ljmp    chdt99
chdt18: cjne    A,#'T',chdt19
        mov     DPTR,#charTb
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt19: cjne    A,#'U',chdt20
        mov     DPTR,#charUb
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt20: cjne    A,#'V',chdt21
        mov     DPTR,#charVb
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt21: cjne    A,#'W',chdt22
        mov     DPTR,#charWb
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt22: cjne    A,#'X',chdt23
        mov     DPTR,#charXb
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt23: cjne    A,#'Y',chdt24
        mov     DPTR,#charYb
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt24: cjne    A,#'Z',chdt25
        mov     DPTR,#charZb
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt25: cjne    A,#'a',chdt26
        mov     DPTR,#charak
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt26: cjne    A,#'b',chdt27
        mov     DPTR,#charbk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt27: cjne    A,#'c',chdt28
        mov     DPTR,#charck
        lcall   ambdta
```

```
        lcall   tlsmem
        ljmp    chdt99
chdt28: cjne    A,#'d',chdt29
        mov     DPTR,#chardk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt29: cjne    A,#'e',chdt30
        mov     DPTR,#charek
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt30: cjne    A,#'f',chdt31
        mov     DPTR,#charfk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt31: cjne    A,#'g',chdt32
        mov     DPTR,#chargk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt32: cjne    A,#'h',chdt33
        mov     DPTR,#charhk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt33: cjne    A,#'i',chdt34
        mov     DPTR,#charik
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt34: cjne    A,#'j',chdt35
        mov     DPTR,#charjk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt35: cjne    A,#'k',chdt36
        mov     DPTR,#charkk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt36: cjne    A,#'l',chdt37
        mov     DPTR,#charlk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt37: cjne    A,#'m',chdt38
        mov     DPTR,#charmk
        lcall   ambdta
```

```
        lcall   tlsmem
        ljmp    chdt99
chdt38: cjne    A,#'n',chdt39
        mov     DPTR,#charnk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt39: cjne    A,#'o',chdt40
        mov     DPTR,#charok
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt40: cjne    A,#'p',chdt41
        mov     DPTR,#charpk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt41: cjne    A,#'q',chdt42
        mov     DPTR,#charqk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt42: cjne    A,#'r',chdt43
        mov     DPTR,#charrk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt43: cjne    A,#'s',chdt44
        mov     DPTR,#charsk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt44: cjne    A,#'t',chdt45
        mov     DPTR,#chartk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt45: cjne    A,#'u',chdt46
        mov     DPTR,#charuk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt46: cjne    A,#'v',chdt47
        mov     DPTR,#charvk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt47: cjne    A,#'w',chdt48
        mov     DPTR,#charwk
        lcall   ambdta
```

```
        lcall  tlsmem
        ljmp   chdt99
chdt48: cjne   A,#'x',chdt49
        mov    DPTR,#charxk
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt49: cjne   A,#'y',chdt50
        mov    DPTR,#charyk
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt50: cjne   A,#'z',chdt51
        mov    DPTR,#charzk
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt51: cjne   A,#'0',chdt52
        mov    DPTR,#charn0
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt52: cjne   A,#'1',chdt53
        mov    DPTR,#charn1
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt53: cjne   A,#'2',chdt54
        mov    DPTR,#charn2
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt54: cjne   A,#'3',chdt55
        mov    DPTR,#charn3
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt55: cjne   A,#'4',chdt56
        mov    DPTR,#charn4
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt56: cjne   A,#'5',chdt57
        mov    DPTR,#charn5
        lcall  ambdta
        lcall  tlsmem
        ljmp   chdt99
chdt57: cjne   A,#'6',chdt58
        mov    DPTR,#charn6
        lcall  ambdta
```

```
        lcall   tlsmem
        ljmp    chdt99
chdt58: cjne    A,#'7',chdt59
        mov     DPTR,#charn7
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt59: cjne    A,#'8',chdt60
        mov     DPTR,#charn8
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt60: cjne    A,#'9',chdt61
        mov     DPTR,#charn9
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt61: cjne    A,#':',chdt62
        mov     DPTR,#chrtt2
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt62: cjne    A,#'/',chdt63
        mov     DPTR,#chrgrm
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt63: cjne    A,#'.',chdt64
        mov     DPTR,#chrttk
        lcall   ambdta
        lcall   tlsmem
        ljmp    chdt99
chdt64: cjne    A,#' ',chdt65
        mov     DPTR,#chrsps
        lcall   ambdta
        lcall   tlsmem
chdt65: cjne    A,#',',chdt66
        mov     DPTR,#chrkom
        lcall   ambdta
        lcall   tlsmem
chdt66: cjne    A,#';',chdt67
        mov     DPTR,#chrtkm
        lcall   ambdta
        lcall   tlsmem
chdt67: cjne    A,#'[',chdt68
        mov     DPTR,#chrbkr
        lcall   ambdta
        lcall   tlsmem
chdt68: cjne    A,#']',chdt69
        mov     DPTR,#chrtpr
```

```
        lcall   ambdta
        lcall   tlsmem
chdt69: cjne    A,#'=',chdt70
        mov     DPTR,#chrsdg
        lcall   ambdta
        lcall   tlsmem
chdt70: cjne    A,#'-',chdt99
        mov     DPTR,#chrstr
        lcall   ambdta
        lcall   tlsmem
chdt99: ret
;
ambdta: clr     A
        movc    A,@A+DPTR
        mov     Buf0,A
        inc     DPTR
        clr     A
        movc    A,@A+DPTR
        mov     Buf1,A
        inc     DPTR
        clr     A
        movc    A,@A+DPTR
        mov     Buf2,A
        inc     DPTR
        clr     A
        movc    A,@A+DPTR
        mov     Buf3,A
        inc     DPTR
        clr     A
        movc    A,@A+DPTR
        mov     Buf4,A
        inc     DPTR
        clr     A
        movc    A,@A+DPTR
        mov     Buf5,A
        ret
;
tlsmem: mov     DPL,DpdL
        mov     DPH,DpdH
        mov     A,Buf0
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Buf1
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Buf2
        movx    @DPTR,A
        lcall   wt_wr
```

```
      inc   DPTR
      mov   A,Buf3
      movx  @DPTR,A
      lcall wt_wr
      inc   DPTR
      mov   A,Buf4
      movx  @DPTR,A
      lcall wt_wr
      inc   DPTR
      mov   A,Buf5
      movx  @DPTR,A
      lcall wt_wr
      inc   DPTR
      mov   DpdL,DPL
      mov   DpdH,DPH
      ret
;
clrdot: mov   DPTR,#100          ; clear colom dot matrix
      mov   Cnt0,#4
      mov   Cnt1,#80
cldot:  mov   A,#0
      movx  @DPTR,A
      lcall wt_wr
      inc   DPTR
      djnz  Cnt1,cldot
      djnz  Cnt0,cldot
      ret
;
wt_wr:  lcall  delay0
      lcall delay0
      lcall delay0
      lcall delay0
      lcall delay0
      lcall delay0
      lcall delay0
      lcall delay0
      ret
;
charAb: DB    07Eh,090h,090h,090h,07Eh,000h
charBb: DB    0FEh,092h,092h,092h,06Ch,000h
charCb: DB    07Ch,082h,082h,082h,044h,000h
charDb: DB    0FEh,082h,082h,044h,038h,000h
charEb: DB    0FEh,092h,092h,092h,082h,000h
charFb: DB    0FEh,090h,090h,090h,080h,000h
charGb: DB    07Ch,082h,092h,092h,05Ch,000h
charHb: DB    0FEh,010h,010h,010h,0FEh,000h
charIb: DB    000h,082h,0FEh,082h,000h,000h
charJb: DB    004h,002h,082h,0FCh,080h,000h
charKb: DB    0FEh,010h,028h,044h,082h,000h
charLb: DB    0FEh,002h,002h,002h,002h,000h
```

```
charMb: DB    0FEh,040h,030h,040h,0FEh,000h
charNb: DB    0FEh,020h,010h,008h,0FEh,000h
charOb: DB    07Ch,082h,082h,082h,07Ch,000h
charPb: DB    0FEh,090h,090h,090h,060h,000h
charQb: DB    07Ch,082h,08Ah,084h,07Ah,000h
charRb: DB    0FEh,090h,098h,094h,062h,000h
charSb: DB    064h,092h,092h,092h,04Ch,000h
charTb: DB    080h,080h,0FEh,080h,080h,000h
charUb: DB    0FCh,002h,002h,002h,0FCh,000h
charVb: DB    0F8h,004h,002h,004h,0F8h,000h
charWb: DB    0FCh,002h,01Ch,002h,0FCh,000h
charXb: DB    0C6h,028h,010h,028h,0C6h,000h
charYb: DB    0E0h,010h,00Eh,010h,0E0h,000h
charZb: DB    086h,08Ah,092h,0A2h,0C2h,000h
charak: DB    004h,02Ah,02Ah,02Ah,01Eh,000h
charbk: DB    0FEh,012h,022h,022h,01Ch,000h
charck: DB    01Ch,022h,022h,022h,004h,000h
chardk: DB    01Ch,022h,022h,012h,0FEh,000h
charek: DB    01Ch,02Ah,02Ah,02Ah,018h,000h
charfk: DB    010h,07Eh,090h,080h,040h,000h
chargk: DB    010h,02Ah,02Ah,02Ah,03Ch,000h
charhk: DB    0FEh,010h,020h,020h,01Eh,000h
charik: DB    000h,022h,0BEh,002h,000h,000h
charjk: DB    004h,002h,022h,0BCh,000h,000h
charkk: DB    0FEh,008h,014h,022h,000h,000h
charlk: DB    000h,082h,0FEh,002h,000h,000h
charmk: DB    03Eh,020h,018h,020h,01Eh,000h
charnk: DB    03Eh,010h,020h,020h,01Eh,000h
charok: DB    01Ch,022h,022h,022h,01Ch,000h
charpk: DB    03Eh,028h,028h,028h,010h,000h
charqk: DB    010h,028h,028h,018h,03Eh,000h
charrk: DB    03Eh,010h,020h,020h,010h,000h
charsk: DB    012h,02Ah,02Ah,02Ah,024h,000h
chartk: DB    020h,0FCh,022h,002h,004h,000h
charuk: DB    03Ch,002h,002h,004h,03Eh,000h
charvk: DB    038h,004h,002h,004h,038h,000h
charwk: DB    03Ch,002h,004h,002h,03Ch,000h
charxk: DB    022h,014h,008h,014h,022h,000h
charyk: DB    030h,00Ah,00Ah,00Ah,03Ch,000h
charzk: DB    022h,026h,02Ah,032h,022h,000h

charn0: DB    07Ch,082h,082h,082h,07Ch,000h
charn1: DB    000h,042h,0FEh,002h,000h,000h
charn2: DB    042h,086h,08Ah,092h,062h,000h
charn3: DB    084h,082h,0A2h,0D2h,08Ch,000h
charn4: DB    018h,028h,048h,0FEh,008h,000h
charn5: DB    0E4h,0A2h,0A2h,0A2h,09Ch,000h
charn6: DB    03Ch,052h,092h,092h,00Ch,000h
charn7: DB    080h,080h,09Eh,0A0h,0C0h,000h
charn8: DB    06Ch,092h,092h,092h,06Ch,000h
```

```
charn9: DB    060h,092h,092h,094h,078h,000h

chrtt2: DB    000h,000h,024h,000h,000h,000h          ; titik dua
chrgrm: DB     004h,008h,010h,020h,040h,000h         ; garis miring
chrttk: DB    000h,000h,002h,000h,000h,000h          ; titik

chrkom: DB     000h,00Ah,00Ch,000h,000h,000h         ; koma
chrtkm: DB     000h,06Ah,06Ch,000h,000h,000h         ; titik koma
chrbkr: DB     000h,0FEh,082h,082h,000h,000h         ; buka kurung
chrtpr: DB    000h,082h,082h,0FEh,000h,000h          ; tutup kurung
chrsdg: DB     028h,028h,028h,028h,028h,000h         ; sama dengan
chrstr: DB    010h,010h,010h,010h,010h,000h          ; strip

chrsps: DB    000h,000h,000h,000h,000h,000h          ; spasi
chrksg: DB    000h,000h,000h,000h,000h,000h          ; kosong
;
bc_rtc: mov    A,#0D0h                  ; address DS1307 write address
     lcall  adrtx                      ; send
     mov    A,#000h                    ; register second
     lcall  dtatx                      ; send
     mov    A,#0D1h                    ; address DS1307 read address
     lcall  adrtx                      ; send
     lcall  dtarx                      ; read data
     lcall  bcddec                     ; convert bcd to dec
     mov    Ddtk,A                     ; save second (time)
     lcall  givack                     ; send ack
     lcall  dtarx                      ; read data
     lcall  bcddec                     ; convert bcd to dec
     mov    Dmnt,A                     ; save minute (time)
     lcall  givack                     ; send ack
     lcall  dtarx                      ; read data
     lcall  bcddec                     ; convert bcd to dec
     mov    Djam,A                     ; save hour (time)
     lcall  givack                     ; send ack
     lcall  dtarx                      ; read data
     lcall  givack                     ; send ack
     lcall  dtarx                      ; read data
     lcall  bcddec                     ; convert bcd to dec
     mov    Dtgl,A                     ; save date (time)
     lcall  givack                     ; send ack
     lcall  dtarx                      ; read data
     lcall  bcddec                     ; convert bcd to dec
     mov    Dbln,A                     ; save month (time)
     lcall  givack                     ; send ack
     lcall  dtarx                      ; read data
     lcall  bcddec                     ; convert bcd to dec
     mov    Dthn,A                     ; save year (time)
     lcall  i2cstp                     ; i2c stop
     ret
;
```

```
bc_gas: mov    A,#90h                  ; address PCF8591 write address
        lcall  adrtx                   ; kirim
        mov    A,#00h                  ; adc ch-1
        lcall  dtatx                   ; kirim
        mov    A,#91h                  ; address PCF8591 read address
        lcall  adrtx                   ; kirim
        lcall  dtarx                   ; baca data
        lcall  givack                  ; beri ack
        lcall  dtarx                   ; baca data
        lcall  i2cstp                  ; i2c stop
        mov    Dadc,A                  ; simpan data ADC sensor gas
        mov    A,#150                  ;\
        mov    B,Dadc                  ; |
        clr    C               ; |
        subb   A,B                     ; |
        jnc    bcgas                   ; |
        mov    A,#0                    ; |
bcgas:  mov    Dgas,A                      ;/
        ret
;
hexdec: mov    R0,#0                   ; cacah urut
        mov    R1,#0
        mov    R2,#0
        mov    R3,#0
        mov    A,Hex1
        jz     hexdc1
        mov    Buf0,#0
hexdc0: lcall  incdec
        djnz   Buf0,hexdc0
        djnz   Hex1,hexdc0
hexdc1: mov    A,Hex0
        jz     hexdc3
hexdc2: lcall  incdec
        djnz   Hex0,hexdc2
hexdc3: ret
;
incdec: inc    R0                      ; increment decimal
        cjne   R0,#10,indec
        mov    R0,#0
        inc    R1
        cjne   R1,#10,indec
        mov    R1,#0
        inc    R2
        cjne   R2,#10,indec
        mov    R2,#0
        inc    R3
        cjne   R3,#10,indec
        mov    R3,#0
indec:  ret
;
```

```
adrtx: lcall  i2cstr                ; kirim address
       lcall  putbit                ; kirim data
       ret                          ; back
;
dtatx: lcall  putbit                ; kirim data
       ret                          ; back
;
dtarx: lcall  getbit                ; terima data
       ret                          ; back
;
putbit: mov   R6,#8                 ;\
putbt: RLC    A                     ; |
       mov    ISDA,C                ; |
       setb   ISCL                  ; |
       clr    ISCL                  ; | kirim bit
       djnz   R6,putbt        ; |
       setb   ISDA                  ; |
       lcall  getack                ; |
       ret                          ;/
;
getbit: mov   R6,#8                 ;\
getbt: setb   ISCL                  ; |
       mov    C,ISDA                ; |
       RLC    A               ; | terima bit
       clr    ISCL                  ; |
       djnz   R6,getbt        ; |
       setb   ISDA                  ; |
       ret                          ;/
;
getack: setb  ISDA                  ;\
       setb   ISCL                  ; |
ackbit: mov   C,ISDA                        ; | tunggu ack
       jc     ackbit                ; | D=1, C=1D=0, C=0
       clr    ISCL                  ; |
       ret                          ;/
;
givack: clr   ISDA                  ;\
       setb   ISCL                  ; |
       clr    ISCL                  ; | kirim ack -> D=0, C=1, C=0, D=1
       setb   ISDA                  ; |
       ret                          ;/
;
i2cstr: setb  ISCL                  ;\
       setb   ISDA                  ; |
       clr    ISDA                  ; | i2c start -> C=1, D=1, D=0, C=0
       clr    ISCL                  ; |
       ret                          ;/
;
i2cstp: clr   ISDA                  ;\
       setb   ISCL                  ; |
```

```
      setb   ISDA                ; | i2c stop -> D=0, C=1, D=1, C=0
      clr    ISCL                ; |
      ret                        ;/
;
bcddec: mov    R1,A
      anl    A,#11110000b
      RR     A
      RR     A
      RR     A
      RR     A
bcddc0: cjne   A,#0,bcddc1
      mov    A,R1
      ljmp   bcddc6
bcddc1: cjne   A,#1,bcddc2
      mov    A,R1
      mov    B,#6
      clr    C
      subb   A,B
      ljmp   bcddc6
bcddc2: cjne   A,#2,bcddc3
      mov    A,R1
      mov    B,#12
      clr    C
      subb   A,B
      ljmp   bcddc6
bcddc3: cjne   A,#3,bcddc4
      mov    A,R1
      mov    B,#18
      clr    C
      subb   A,B
      ljmp   bcddc6
bcddc4: cjne   A,#4,bcddc5
      mov    A,R1
      mov    B,#24
      clr    C
      subb   A,B
      ljmp   bcddc6
bcddc5: cjne   A,#5,bcddc6
      mov    A,R1
      mov    B,#30
      clr    C
      subb   A,B
bcddc6: ret
;
bc_sms: lcall  bcnohp             ;\
      jnb    Sttc,bcsms0         ; | baca no HP
      lcall  bccmnd              ; | cek status
      lcall  hpssms              ; | 0=hapus sms, 1=baca command
      ljmp   bcsms1              ; | hapus sms
bcsms0: lcall  hpssms             ; | cek command
```

```
      ret                              ;/
;
bcsms1: mov    DPTR,#comnd0                ;\
      lcall  ckcmnd                    ; | ambil data command
      jnb    Sttc,bcsms2               ; | cek command
      lcall  krmsns                    ; | kirim data sensor
      ljmp   bcsms3                    ;/
;
bcsms2: mov    DPTR,#comnd1                ;\
      lcall  ckcmnd                    ; | ambil data command
      jnb    Sttc,bcsms3               ; | cek command
      lcall  bc_tim                    ; | kirim data sensor
      ljmp   bcsms3                    ;/
;
bcsms3: ret
;
krmsns: lcall  nilpdu                  ; rubah nilai sensor -> PDU
      mov    DPTR,#smskrm              ; ambil data PDU kirim 20 oktet
      mov    ChrH,#10                  ; ATCommand 10 character
      lcall  kr_ins                    ; kirim data diakhiri enter
      lcall  bc_fbk                    ; tunggu feedback dari HP
      mov    Dly3,#1
      lcall  delay3
      mov    DPTR,#sensor              ; ambil data PDU "sensor"
      mov    ChrH,#44                  ; ATCommand 44 character
      lcall  kr_dta                    ; kirim data setengah yg sudah pasti
      lcall  kr_pdu                    ; kirim data diakhiri CTRL+Z
      lcall  bc_fbk                    ; tunggu feedback dari HP
      mov    Dly3,#1
      lcall  delay3
      ret
;
nilpdu: mov    Chr0,R3                        ; nilai -> PDU
      mov    Chr1,R2
      mov    Chr2,R1
      mov    Chr3,R0
      mov    Chr4,#' '
      mov    Chr5,#'p'
      mov    Chr6,#'p'
      mov    Chr7,#'m'
      lcall  jd_pdu
      ret
;
bcnohp: mov    DPTR,#smsred                ;\
      mov    ChrH,#9                   ; |
      lcall  kr_ins                    ; | baca no hp
      mov    Cntr,#35          ; | header sms + service center
      lcall  bc_fbk                    ; | dihilangkan
bcnolp: lcall  bc_srl                  ; |
      djnz   Cntr,bcnolp               ; |
```

```
      lcall  bc_dta                    ;/
;
cknohp: mov    DPTR,#nohpku             ;\
      mov    Cntr,#12          ; | jumlah no hp = 14
      mov    Cnsm,#0                   ; | reset counter sms
      mov    Cndt,#0                   ; | reset counter no hp
cknhp0: lcall  Dtasms                  ; | ambil data counter sms
      lcall  Dtanhp                    ; | ambil data counter no hp
      clr    C                 ; |
      subb   A,B                       ; | samakan
      cjne   A,#0,cknhp1               ; | jika sama
      inc    Cnsm                      ; | ambil data counter sms selanjutnya
      inc    Cndt                      ; | ambil data counter no hp selajnutnya
      djnz   Cntr,cknhp0               ; | cek sampai no hp habis
      setb   Sttc                      ; | set status = 1
      ljmp   cknhp2                    ; | jika tidak
cknhp1: clr    Sttc                    ; | set status = 0
cknhp2: ret                            ;/
;
dtanhp: mov    A,Cndt                        ;\
      movc   A,@A+DPTR         ; | ambil data pointer yang ke cndt
      ret                              ;/
;
bccmnd: mov    DPTR,#smsred              ;\
      mov    ChrH,#9                   ; |
      lcall  kr_ins                    ; | baca command
      mov    Cntr,#67          ; | header sms + service center
      lcall  bc_fbk                    ; | + no HP
bccmlp: lcall  bc_srl                  ; | dihilangkan
      djnz   Cntr,bccmlp               ; |
      lcall  bc_dta                    ; |
;      lcall  tl_dta                   ; |
      ret                              ;/
;
ckcmnd: mov    Cntr,#08                ;\
      mov    Cnsm,#0                   ; | jumlah command
      mov    Cndt,#0                   ; | reset counter sms
ckcnd0: lcall  Dtasms                  ; | reset counter command
      lcall  Dtacmd                    ; | ambil data counter sms
      clr    C                 ; | ambil data counter command
      subb   A,B                       ; | samakan
      cjne   A,#0,ckcnd1               ; | jika sama
      inc    Cnsm                      ; | ambil data counter sms selanjutnya
      inc    Cndt                      ; | ambil data counter command selanjutnya
      djnz   Cntr,ckcnd0               ; | cek sampai command habis
      setb   Sttc                      ; | set status = 1
      ljmp   ckcnd2                    ; | jika tidak
ckcnd1: clr    Sttc                    ; | set status = 0
ckcnd2: ret                            ;/
;
```

```
dtacmd: mov    A,Cndt                    ;\
        movc   A,@A+DPTR          ; | ambil data pointer yang ke cndt
        ret                       ;/
;
dtasms: clr    C                  ;\
        mov    A,Cnsm             ; |
dtsm00: cjne   A,#00,dtsm01       ; |
        mov    B,Chr0             ; |
dtsm01: cjne   A,#01,dtsm02       ; |
        mov    B,Chr1             ; |
dtsm02: cjne   A,#02,dtsm03       ; |
        mov    B,Chr2             ; |
dtsm03: cjne   A,#03,dtsm04       ; |
        mov    B,Chr3             ; |
dtsm04: cjne   A,#04,dtsm05       ; |
        mov    B,Chr4             ; |
dtsm05: cjne   A,#05,dtsm06       ; |
        mov    B,Chr5             ; |
dtsm06: cjne   A,#06,dtsm07       ; |
        mov    B,Chr6             ; |
dtsm07: cjne   A,#07,dtsm08       ; |
        mov    B,Chr7             ; | samakan data counter
dtsm08: cjne   A,#08,dtsm09       ; | dengan character
        mov    B,Chr8             ; |
dtsm09: cjne   A,#09,dtsm0A              ; |
        mov    B,Chr9             ; |
dtsm0A: cjne   A,#10,dtsm0B              ; |
        mov    B,ChrA             ; |
dtsm0B: cjne   A,#11,dtsm0C              ; |
        mov    B,ChrB             ; |
dtsm0C: cjne   A,#12,dtsm0D              ; |
        mov    B,ChrC             ; |
dtsm0D: cjne   A,#13,dtsm0E              ; |
        mov    B,ChrD             ; |
dtsm0E: cjne   A,#14,dtsm0F       ; |
        mov    B,ChrE             ; |
dtsm0F: cjne   A,#15,dtsmsx       ; |
        mov    B,ChrF             ; |
dtsmsx: ret                       ;/
;
bc_dta: lcall  bc_srl             ;\
        mov    Chr0,R7            ; |
        lcall  bc_srl             ; |
        mov    Chr1,R7            ; |
        lcall  bc_srl             ; |
        mov    Chr2,R7            ; |
        lcall  bc_srl             ; |
        mov    Chr3,R7            ; |
        lcall  bc_srl             ; |
        mov    Chr4,R7            ; |
```

```
      lcall  bc_srl                ;|
      mov    Chr5,R7               ;|
      lcall  bc_srl                ;|
      mov    Chr6,R7               ;|
      lcall  bc_srl                ;|
      mov    Chr7,R7               ;|
      lcall  bc_srl                ;|
      mov    Chr8,R7               ;| baca serial & simpan
      lcall  bc_srl                ;| ke memory character
      mov    Chr9,R7               ;|
      lcall  bc_srl                ;|
      mov    ChrA,R7               ;|
      lcall  bc_srl                ;|
      mov    ChrB,R7               ;|
      lcall  bc_srl                ;|
      mov    ChrC,R7               ;|
      lcall  bc_srl                ;|
      mov    ChrD,R7               ;|
      lcall  bc_srl                ;|
      mov    ChrE,R7               ;|
      lcall  bc_srl                ;|
      mov    ChrF,R7               ;|
      mov    Dly3,#5               ;|
      lcall  delay3                ;|
      ret                          ;/
;
kr_ins: clr    A                   ;\
        movc   A,@A+DPTR           ;|
        lcall  kr_srl              ;|
        inc    DPTR                ;| kirim instruksi ke hp
        djnz   ChrH,kr_ins         ;| sebagai awal pengiriman data
      mov    A,#0Dh                ;|
        lcall  kr_srl              ;|
      ret                          ;/
;
kr_dta: clr    A                   ;\
        movc   A,@A+DPTR           ;|
        lcall  kr_srl              ;| kirim setengah data pdu
        inc    DPTR                ;| yang sudah pasti
        djnz   ChrH,kr_dta         ;|
      ret                          ;/
;
kr_srl: clr    ES                  ;\
      mov    SBUF,A                ;|
      jnb    TI,$                  ;|
      clr    TI              ;| kirim serial ke hp
      setb   ES              ;|
      lcall  delay0                ;|
      ret                          ;/
;
```

```
bc_tim: mov   DPTR,#smstim          ;\
        mov   ChrH,#8               ; | kirim
        lcall kr_ins                ; | instruksi
        mov   Cntr,#9               ; |
        lcall bc_fbk                ; |
bctim:  lcall bc_srl                ; | dihilangkan
        djnz  Cntr,bctim            ; |
        lcall bc_srl                ; |
        mov   Chr0,R7               ; |
        lcall bc_srl                ; |
        mov   Chr1,R7               ; |
        lcall bc_srl                ; |
        lcall bc_srl                ; |
        mov   Chr2,R7               ; |
        lcall bc_srl                ; |
        mov   Chr3,R7               ; |
        lcall bc_srl                ; |
        lcall bc_srl                ; |
        mov   Chr4,R7               ; |
        lcall bc_srl                ; |
        mov   Chr5,R7               ; |
        lcall bc_srl                ; |
        lcall bc_srl                ; |
        mov   Chr6,R7               ; |
        lcall bc_srl                ; |
        mov   Chr7,R7               ; |
        lcall bc_srl                ; |
        lcall bc_srl                ; |
        mov   Chr8,R7               ; |
        lcall bc_srl                ; |
        mov   Chr9,R7               ; |
        lcall bc_srl                ; |
        lcall bc_srl                ; |
        mov   ChrA,R7               ; |
        lcall bc_srl                ; |
        mov   ChrB,R7               ; |
        mov   Dly3,#2
        lcall delay3
;
        mov   A,Chr0
        lcall ascdec
        RL    A
        RL    A
        RL    A
        RL    A
        mov   Chr0,A
        mov   A,Chr1
        lcall ascdec
        orl   A,Chr0
        mov   Dthn,A
```

```
;
    mov   A,Chr2
    lcall ascdec
    RL    A
    RL    A
    RL    A
    RL    A
    mov   Chr2,A
    mov   A,Chr3
    lcall ascdec
    orl   A,Chr2
    mov   Dbln,A
;
    mov   A,Chr4
    lcall ascdec
    RL    A
    RL    A
    RL    A
    RL    A
    mov   Chr4,A
    mov   A,Chr5
    lcall ascdec
    orl   A,Chr4
    mov   Dtgl,A
;
    mov   A,Chr6
    lcall ascdec
    RL    A
    RL    A
    RL    A
    RL    A
    mov   Chr6,A
    mov   A,Chr7
    lcall ascdec
    orl   A,Chr6
    mov   Djam,A
;
    mov   A,Chr8
    lcall ascdec
    RL    A
    RL    A
    RL    A
    RL    A
    mov   Chr8,A
    mov   A,Chr9
    lcall ascdec
    orl   A,Chr8
    mov   Dmnt,A
;
    mov   A,ChrA
```

```
      lcall  ascdec
      RL     A
      RL     A
      RL     A
      RL     A
      mov    ChrA,A
      mov    A,ChrB
      lcall  ascdec
      orl    A,ChrA
      mov    Ddtk,A
;
sttime: mov    A,#0D0h                    ; address DS1307 write address
      lcall  adrtx                  ; send
      mov    A,#000h                ; register second
      lcall  dtatx                  ; send
      mov    A,Ddtk                 ; data second (time)
      lcall  dtatx                  ; send
      mov    A,Dmnt                 ; data minute (time)
      lcall  dtatx                  ; send
      mov    A,Djam                 ; data hour (time)
      lcall  dtatx                  ; send
      mov    A,#01                  ; data day (time)
      lcall  dtatx                  ; send
      mov    A,Dtgl                 ; data date (time)
      lcall  dtatx                  ; send
      mov    A,Dbln                 ; data month (time)
      lcall  dtatx                  ; send
      mov    A,Dthn                 ; data year (time)
      lcall  dtatx                  ; send
      lcall  i2cstp                 ; i2c stop
      ret                           ;/
;
ascdec: mov    B,#30h                     ; ascii to decimal
      clr    C
      subb   A,B
      ret
;
noecho: mov    Dly3,#1                    ;\
      lcall  delay3                 ; |
      mov    DPTR,#smseco           ; |
      mov    ChrH,#4                ; | kirim
      lcall  kr_ins                 ; | instruksi
      lcall  bc_fbk                 ; | no-echo ke hp
      mov    Dly3,#1                ; |
      lcall  delay3                 ; |
      ret                           ;/
;
dsminc: mov    Dly3,#1                    ;\
      lcall  delay3                 ; |
      mov    DPTR,#smsfrm           ; |
```

```
        mov    ChrH,#17              ; | kirim instruksi
        lcall  kr_ins                ; | display-incoming-sms
        lcall  bc_fbk                ; | ke hp
        mov    Dly3,#1               ; |
        lcall  delay3                ; |
        ret                          ;/
;
hpssms: mov    DPTR,#smshps                ;\
        mov    ChrH,#9               ; |
        lcall  kr_ins                ; | hapus sms
        lcall  bc_fbk                ; | pada memory 1
        mov    Dly3,#1               ; |
        lcall  delay3                ; |
        ret                          ;/
;
bc_fbk: mov    R7,#0FFh              ;\
bc_fb0: cjne   R7,#0FFh,bc_fb1               ; | tunggu feedback
        ljmp   bc_fb0                ; | data dari hp
bc_fb1: ret                          ;/
;
bc_srl: mov    R7,#0FFh              ;\
        mov    Tmo0,#0               ; |
        mov    Tmo1,#10              ; | baca serial
bc_sr0: cjne   R7,#0FFh,bc_srl               ; | tunggu data bukan FF
        djnz   Tmo0,bc_sr0           ; | sebelum time out
        djnz   Tmo1,bc_sr0           ; |
bc_srl: ret                          ;/
;
rstcmd: mov    R7,#0FFh
        ret
;
srl_in: mov    TMOD,#20h             ;\
        mov    TH1,#0FDh             ; |
        mov    SCON,#50h             ; |
        mov    A,#80h                ; | set baudrate
        orl    87h,A                 ; | 19200bps
        setb   TR1                   ; |
        setb   EA                ; |
        setb   ES                ; |
        ret                          ;/
;
delay0: djnz   Dly0,delay0
        ret
;
delay1: mov    Dly1,#2
dely10: lcall  delay0
        djnz   Dly1,dely10
        ret
;
delay2: mov    Dly2,#15
```

```
dely20: lcall  delay0
      djnz   Dly1,dely20
      djnz   Dly2,dely20
      ret
;
delay3: lcall  delay0
      djnz   Dly1,delay3
      djnz   Dly3,delay3
      ret
;
smseco: DB     'ATE0'
smsred: DB     'AT+CMGR=1'
smshps: DB     'AT+CMGD=1'
smskrm: DB     'AT+CMGS=28'
smstim: DB     'AT+CCLK?'
smsfrm: DB     'AT+CNMI=1,1,0,0,1'
comnd0: DB     'D3A2730A'          ; SENS
comnd1: DB     '532AB509'          ; STTM
;nohpku: DB    '265857151649F6'    ; HP
nohpku: DB     '261831066448'      ; HP
angka:  DB     '0123456789ABCDEF'        ; character angka
sensor: DB     '0001000D91265857151649F6000010D3B27BFE96EB40'
pdulkp: DB
'0001000D91265857151649F6000010D3B27BFE96EB4030180C0682C3DB'
ascii:  DB     'Sensor: 0000 ppm'
;
angka0: DB     00111111b
      DB     00000110b
      DB     01011011b
      DB     01001111b
      DB     01100110b
      DB     01101101b
      DB     01111101b
      DB     00000111b
      DB     01111111b
      DB     01101111b
;
kr_pdu: mov    DPTR,#angka                ;\
      mov    R0,Chr0                    ; |
        lcall    cchpdu                 ; |
      mov    R0,Chr1                    ; |
        lcall    cchpdu                 ; |
      mov    R0,Chr2                    ; |
        lcall    cchpdu                 ; | kirim PDU
      mov    R0,Chr3                    ; |
        lcall    cchpdu                 ; |
      mov    R0,Chr4                    ; |
        lcall    cchpdu                 ; |
      mov    R0,Chr5                    ; |
        lcall    cchpdu                 ; |
```

```
      mov    R0,Chr6                ;|
        lcall    cchpdu             ;|
      mov    R0,Chr7                ;|
        lcall    cchpdu             ;|
      mov    A,#26                  ;|
        lcall    kr_srl             ;|
      ret                           ;/
;
cchpdu: mov    A,R0                 ;\
      anl    A,#0F0h                ;|
      RR     A              ;|
      RR     A              ;|
      RR     A              ;|
      RR     A              ;| cacah character
      movc   A,@A+DPTR              ;| menjadi data hexa
      lcall  kr_srl                 ;|
      mov    A,R0                   ;|
      anl    A,#00Fh                ;|
      movc   A,@A+DPTR              ;|
      lcall  kr_srl                 ;|
      ret                           ;/
;
jd_pdu: mov    A,Chr1                       ;\
      RR     A              ;| data1 geser 7x
      anl    A,#080h                ;| and dg 1000 0000
      orl    A,Chr0                 ;| or dg data0
      mov    Chr0,A                 ;/
      mov    A,Chr1                 ;\
      anl    A,#11111110b           ;|
      RR     A              ;| data1 hilangkan bit0
      mov    Bufr,A                 ;| data1 geser kanan 1x
      mov    A,Chr2                 ;| simpan sementara
      RR     A              ;| and dg 1100 0000
      RR     A              ;| or dg bufr
      anl    A,#0C0h                ;|
      orl    A,Bufr                 ;|
      mov    Chr1,A                 ;/
      mov    A,Chr2                 ;\
      anl    A,#11111100b           ;|
      RR     A              ;|
      RR     A              ;| data1 hilangkan bit0 & bit1
      mov    Bufr,A                 ;| data1 geser kanan 2x
      mov    A,Chr3                 ;| simpan sementara
      RR     A              ;| and dg 1110 0000
      RR     A              ;| or dg bufr
      RR     A              ;|
      anl    A,#0E0h                ;|
      orl    A,Bufr                 ;|
      mov    Chr2,A                 ;/
      mov    A,Chr3                 ;\
```

```
anl   A,#11111000b          ;|
RR    A                     ;|
RR    A                     ;|
RR    A                     ;| data1 hilangkan bit0,bit1 & bit2
mov   Bufr,A                ;| data1 geser kanan 3x
mov   A,Chr4                ;| simpan sementara
RR    A                     ;| and dg 1111 0000
RR    A                     ;| or dg bufr
RR    A                     ;|
RR    A                     ;|
anl   A,#0F0h               ;|
orl   A,Bufr                ;|
mov   Chr3,A                ;/
mov   A,Chr4                ;\
anl   A,#11110000b          ;|
RR    A                     ;|
RR    A                     ;|
RR    A                     ;|
RR    A                     ;|
mov   Bufr,A                ;|
mov   A,Chr5                ;|
RR    A                     ;|
RR    A                     ;|
RR    A                     ;|
RR    A                     ;|
RR    A                     ;|
anl   A,#0F8h               ;|
orl   A,Bufr                ;|
mov   Chr4,A                ;/
mov   A,Chr5                ;\
anl   A,#11100000b          ;|
RR    A                     ;|
RR    A                     ;|
RR    A                     ;|
RR    A                     ;|
RR    A                     ;|
mov   Bufr,A                ;|
mov   A,Chr6                ;|
RR    A                     ;|
RR    A                     ;|
RR    A                     ;|
RR    A                     ;|
RR    A                     ;|
RR    A                     ;|
anl   A,#0FCh               ;|
orl   A,Bufr                ;|
mov   Chr5,A                ;/
mov   A,Chr6                ;\
anl   A,#11000000b          ;|
RR    A                     ;|
```

```
        RR    A               ;|
        RR    A               ;|
        RR    A               ;|
        RR    A               ;|
        RR    A               ;|
        mov   Bufr,A                ;|
        mov   A,Chr7                ;|
        RR    A               ;|
        RR    A               ;|
        RR    A               ;|
        RR    A               ;|
        RR    A               ;|
        RR    A               ;|
        RR    A               ;|
        anl   A,#0FEh               ;|
        orl   A,Bufr                ;|
        mov   Chr6,A                ;/

        ret
;
lokup0: DB    094,124,154,184,214,244,018,048,078,108       ; 000-009
        DB    138,168,198,228,002,032,062,092,122,152       ; 010-019
        DB    182,212,242,016,046,076,106,136,166,196       ; 020-029
        DB    226,000,030,060,090,120,150,180,210,240       ; 030-039
        DB    014,044,074,104,134,164,194,224,254,028       ; 040-049
        DB    058,088,118,148,178,208,238,012,042,072       ; 050-059
        DB    112,132,162,192,222,252,026,056,086,116       ; 060-069
        DB    146,176,206,236,010,040,070,100,130,160       ; 070-079
        DB    190,220,250,024,054,084,114,144,174,204       ; 080-089
        DB    234,008,038,068,098,128,158,188,218,248       ; 090-099
        DB    022,052,082,112,142,172,202,232,006,036       ; 100-109
        DB    066,096,126,156,186,216,246,020,050,080       ; 110-119
        DB    110,140,170,200,230,004,034,064,094,124       ; 120-129
        DB    154,184,214,244,018,048,078,108,138,168       ; 130-139
        DB    198,228,002,032,062,092,122,152,182,212       ; 140-149
        DB    242,016,046,076,106,136,166,196,226,000       ; 150-159
        DB    030,060,090,120,150,180,210,240,014,044       ; 160-169
        DB    074,104,134,164,194,224,254,028,058,088       ; 170-179
        DB    118,184,178,208,238,012,042,072,102,132       ; 180-189
        DB    162,192,222,252,026,056,086,116,146,176       ; 190-199
        DB    206,236,010,040,070,100,130,160,190,220       ; 200-209
        DB    250,024,054,084,114,144,174,204,238,008       ; 210-219
        DB    038,068,098,128,158,188,218,248,022,052       ; 220-229
        DB    082,112,142,172,202,232,006,036,066,096       ; 230-239
        DB    126,156,186,216,246,020,050,080,110,140       ; 240-249
        DB    170,200,230,004,034,064                       ; 250-255
;
lokup1: DB    001,001,001,001,001,001,002,002,002,002       ; 000-009
        DB    002,002,002,002,003,003,003,003,003,003       ; 010-019
        DB    003,003,003,004,004,004,004,004,004,004       ; 020-029
```

```
    DB    004,005,005,005,005,005,005,005,005,005          ; 030-039
    DB    006,006,006,006,006,006,006,006,006,007          ; 040-049
    DB    007,007,007,007,007,007,007,008,008,008          ; 050-059
    DB    008,008,008,008,008,008,008,009,009,009          ; 060-069
    DB    009,009,009,009,010,010,010,010,010,010          ; 070-079
    DB    010,010,010,011,011,011,011,011,011,011          ; 080-089
    DB    011,012,012,012,012,012,012,012,012,012          ; 090-099
    DB    013,013,013,013,013,013,013,013,014,014          ; 100-109
    DB    014,014,014,014,014,014,014,015,015,015          ; 110-119
    DB    015,015,015,015,015,016,016,016,016,016          ; 120-129
    DB    016,016,016,016,017,017,017,017,017,017          ; 130-139
    DB    017,017,018,018,018,018,018,018,018,018          ; 140-149
    DB    018,019,019,019,019,019,019,019,019,020          ; 150-159
    DB    020,020,020,020,020,020,020,020,021,021          ; 160-169
    DB    021,021,021,021,021,021,021,022,022,022          ; 170-179
    DB    022,022,022,022,022,023,023,023,023,023          ; 180-189
    DB    023,023,023,023,024,024,024,024,024,024          ; 190-199
    DB    024,024,025,025,025,025,025,025,025,025          ; 200-209
    DB    025,026,026,026,026,026,026,026,026,027          ; 210-219
    DB    027,027,027,027,027,027,027,027,028,028          ; 220-229
    DB    028,028,028,028,028,028,029,029,029,029          ; 230-239
    DB    029,029,029,029,029,030,030,030,030,030          ; 240-249
    DB    030,030,030,031,031,031                          ; 250-255
;
tl_dta: mov   DPTR,#00h
    mov   A,Chr0              ; 0
    movx  @DPTR,A
    lcall wt_wr
    inc   DPTR
    mov   A,Chr1              ; 1
    movx  @DPTR,A
    lcall wt_wr
    inc   DPTR
    mov   A,Chr2              ; 2
    movx  @DPTR,A
    lcall wt_wr
    inc   DPTR
    mov   A,Chr3              ; 3
    movx  @DPTR,A
    lcall wt_wr
    inc   DPTR
    mov   A,Chr4              ; 4
    movx  @DPTR,A
    lcall wt_wr
    inc   DPTR
    mov   A,Chr5              ; 5
    movx  @DPTR,A
    lcall wt_wr
    inc   DPTR
    mov   A,Chr6              ; 6
```

```
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,Chr7                ; 7
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,Chr8                ; 8
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,Chr9                ; 9
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,ChrA                ; A
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,ChrB                ; 0
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,ChrC                ; C
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,ChrD                ; D
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,ChrE                ; E
        movx   @DPTR,A
        lcall  wt_wr
        inc    DPTR
        mov    A,ChrF                ; F
        movx   @DPTR,A
        lcall  wt_wr
;
        mov    Jchr,#29        ; jumlah character ascii
        mov    DpcL,#00h               ; ambil dari data pointer
        mov    DpcH,#00h               ; address 0000h
        lcall  cchchr                  ; cacah character
smntr0: mov    Jcd0,#254               ; jalankan terus menerus
        mov    Jcd1,#000               ; dalam 254 colom dot matrix led
        lcall  tljln1                  ; oke
        ljmp   smntr0
;
        end
```

IGARO

PRODUCT INFORMATION

# TGS 4161 - for the detection of Carbon Dioxide

## Features:

* High selectivity to $CO_2$
* Compact size
* Low dependency on humidity
* Long life
* Low power consumption

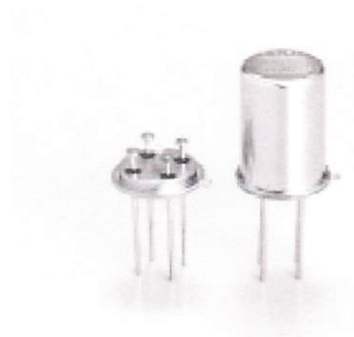
## Applications:

* Air quality control
* Fermentation process control

**GS4161** is a new solid electrolyte $CO_2$ sensor which offers miniaturization and ow power consumption. The $CO_2$ sensitive element consists of a solid electrolyte ormed between two electrodes, together with a printed heater ($RuO_2$) substrate. y monitoring the change in electromotive force (EMF) generated between the vo electrodes, it is possible to measure $CO_2$ gas concentration.

he top of the sensor cap contains adsorbent (zeolite) for the purpose of educing the influence of interference gases.

**GS4161** exhibits a linear relationship between ΔEMF and $CO_2$ gas concentration n a logarithmic scale. The sensor displays good long term stability and shows xcellent durability against the effects of high humidity through the application f innovative technology in the sensor's electrode design.



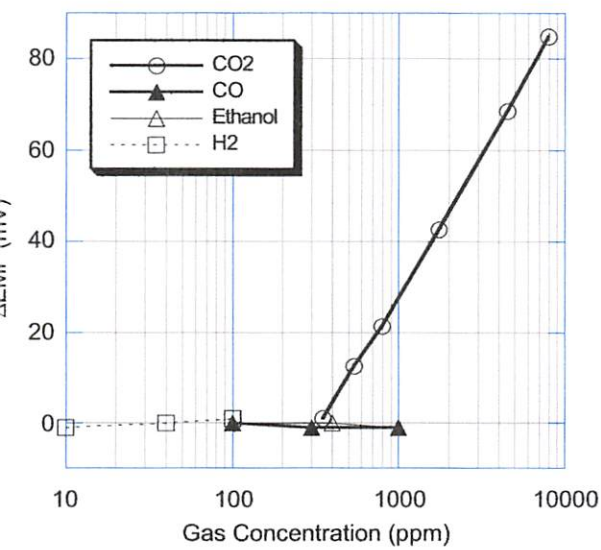
he figure below represents typical sensitivity characteristics of GS4161. The Y-axis is indicated as ΔEMF which is defined as llows:

ΔEMF=EMF1 - EMF2
*where*
EMF1=EMF in 350 ppm $CO_2$
EMF2=EMF in listed gas concentration

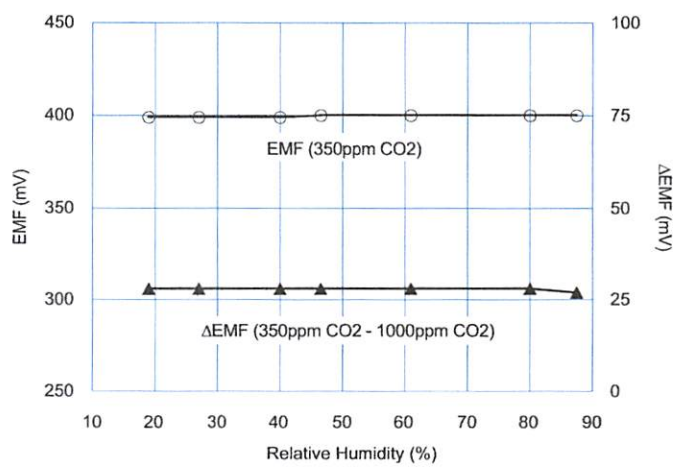
The figure below shows typical humidity dependency of TGS4161. Again, the Y-axis is indicated as ΔEMF which is defined as follows:

ΔEMF=EMF1 - EMF2
*where*
EMF1=EMF in 350 ppm $CO_2$
EMF2=EMF in 1000ppm $CO_2$

### ensitivity Characteristics:



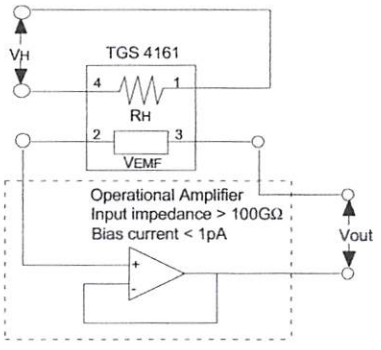

### Humidity Dependency:




**IPORTANT NOTE:** OPERATING CONDITIONS IN WHICH FIGARO SENSORS ARE USED WILL VARY WITH EACH CUSTOMER'S SPECIFIC APPLICATIONS. FIGARO STRONGLY ECOMMENDS CONSULTING OUR TECHNICAL STAFF BEFORE DEPLOYING FIGARO SENSORS IN YOUR APPLICATION AND, IN PARTICULAR, WHEN CUSTOMER'S TARGET GASES RE NOT LISTED HEREIN. FIGARO CANNOT ASSUME ANY RESPONSIBILITY FOR ANY USE OF ITS SENSORS IN A PRODUCT OR APPLICATION FOR WHICH SENSOR HAS NOT BEEN PECIFICALLY TESTED BY FIGARO.

## Basic Measuring Circuit:

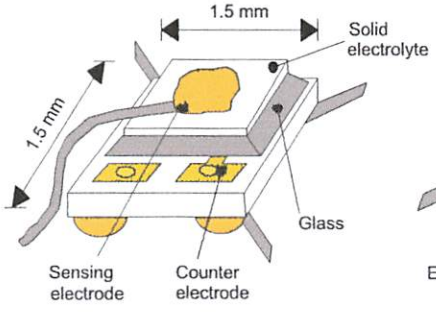
The TGS4161 sensor requires heater voltage ($V_H$) input. The heater voltage is applied to the integrated heater in order to maintain the sensing element at a specific temperature which is optimal for sensing. Electromotive force (EMF) of the sensor should be measured using a high impedance (>100 GΩ) operational amplifier with bias current < 1pA (e.g. Texas Instruments' model #TLC271). Since the solid electrolyte type sensor functions as a kind of battery, the EMF value itself would drift using this basic measuring circuit. However, the change of EMF value (ΔEMF) shows a stable relationship with the change of $CO_2$ concentration. Therefore, in order to obtain an accurate measurement of $CO_2$, a special microprocessor for signal processing should be used with TGS4161. Figaro can provide a special evaluation sensor module (AM-4) for TGS4161.



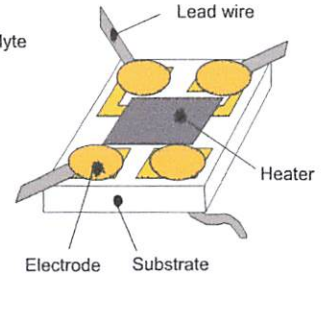

## Specifications:

| | | | | |
|---|---|---|---|---|
| Model number | | | TGS 4161 | |
| Sensing element type | | | Solid electrolyte | |
| Target gases | | | Carbon dioxide | |
| Typical detection range | | | 350 ~ 8000 ppm | |
| Electrical characteristics | Heater resistance | $R_H$ | 70 ± 7Ω at room temp. | |
| | Heater current | $I_H$ | approx. 50mA | |
| | Heater power consumption | $P_H$ | approx. 250mW | |
| | Electromotive force | EMF | 220~490mV in 350ppm $CO_2$ | |
| | Sensitivity | ΔEMF | 44~72mV | EMF(350ppm $CO_2$)-EMF(3500ppm $CO_2$) |
| | Heater voltage | $V_H$ | 5.0 ± 0.2V (DC) | |
| Sensor characteristics | Response time | | approx. 1.5 min. (to 90% of final value) | |
| | Measurement accuracy | | approx. ±20% at 1,000ppm $CO_2$ | |
| Operating conditions | | | -10~50°C, 5~95%RH | |
| Strorage conditions | | | -20~60°C, 5~90%RH (store in moisture proof bag with silica gel) | |
| Standard test conditions | Test gas condition | | $CO_2$ in air at 20±2°C, 65±5%RH | |
| | Circuit condition | | $V_H$ = 5.0±0.05V DC | |
| | Conditioning period before test | | 2 days | |

## Sensing Element Structure:

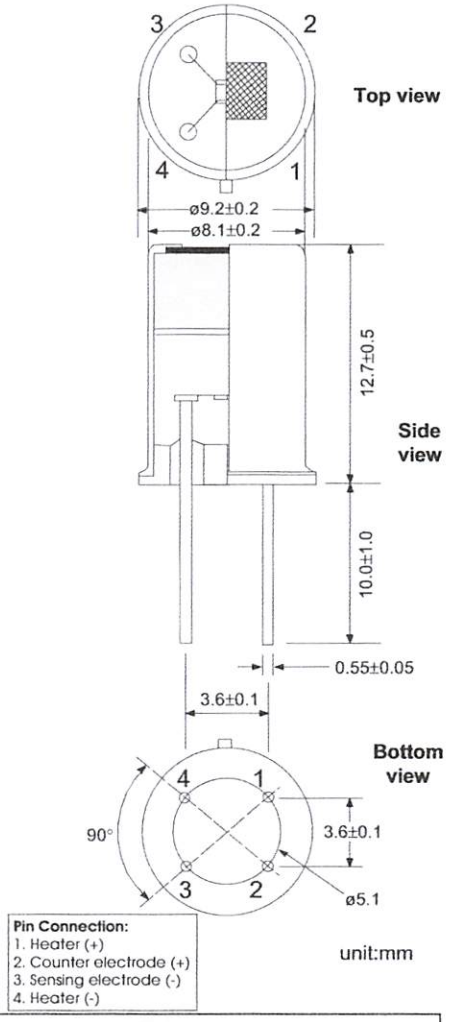



Solid electrolyte side




Heater side

## Structure and Dimensions:




**Pin Connection:**
1. Heater (+)
2. Counter electrode (+)
3. Sensing electrode (-)
4. Heater (-)


**FIGARO ENGINEERING INC.**
1-5-11 Senba-nishi
Mino, Osaka 562-8505 JAPAN
Phone: (81)-727-28-2561
Fax: (81)-727-28-0467
email: figaro@figaro.co.jp

**FIGARO USA, INC.**
3703 West Lake Ave. Suite 203
Glenview, Illinois 60025 USA
Phone: (1)-847-832-1701
Fax: (1)-847-832-1705
email: figarousa@figarosensor.com


REV: 08/01

DALLAS SEMICONDUCTOR

# DS1307/DS1308
# 64 X 8 Serial Real Time Clock

www.dalsemi.com

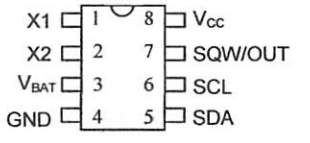
## FEATURES

- Real time clock counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap year compensation valid up to 2100
- 56 byte nonvolatile RAM for data storage
- 2-wire serial interface
- Programmable squarewave output signal
- Automatic power-fail detect and switch circuitry
- Consumes less than 500 nA in battery backup mode with oscillator running
- Optional industrial temperature range -40°C to +85°C (IND) available for DS1307 and DS1308
- DS1307 available in 8-pin DIP or SOIC
- DS1308 available in 36-pin SMD BGA (Ball Grid Array)
- DS1308 accuracy is better than ±2 minute/month at 25°C
- Recognized by Underwriters Laboratory

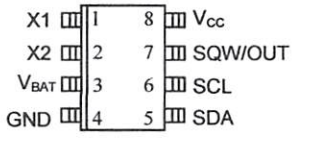
## ORDERING INFORMATION

| | |
|---|---|
| 1307 | 8-Pin DIP |
| 1307Z | 8-Pin SOIC (150 mil) |
| 1307N | 8-Pin DIP (Industrial) |
| 1307ZN | 8-Pin SOIC (Industrial) |
| 1308 | 36-Pin BGA |
| 1308N | 36-Pin BGA (Industrial) |

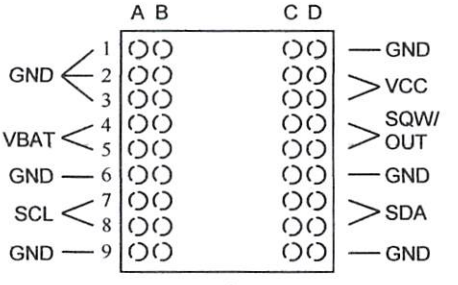
## PIN ASSIGNMENT




DS1307 8-Pin DIP (300 mil)




DS1307Z 8-Pin SOIC (150 mil)




DS1308 36-Pin SMD BGA
**(Top View)**

## PIN DESCRIPTION DS1307/DS1308

| | |
|---|---|
| $V_{CC}$ | - Primary Power Supply |
| X1, X2 | - 32.768 kHz Crystal Connection |
| $V_{BAT}$ | - +3V Battery Input |
| GND | - Ground |
| SDA | - Serial Data |
| SCL | - Serial Clock |
| SQW/OUT | - Square wave/Output Driver |

## DS1308 PIN IDENTIFIER

| | |
|---|---|
| $V_{CC}$ | - C2, C3, D2, D3 |
| $V_{BAT}$ | - A4, A5, B4, B5 |
| SDA | - C7, C8, D7, D8 |
| SCL | - A7, A8, B7, B8 |
| SQW/OUT | - C4, C5, D4, D5 |
| GND | - All Remaining Balls |

## ESCRIPTION

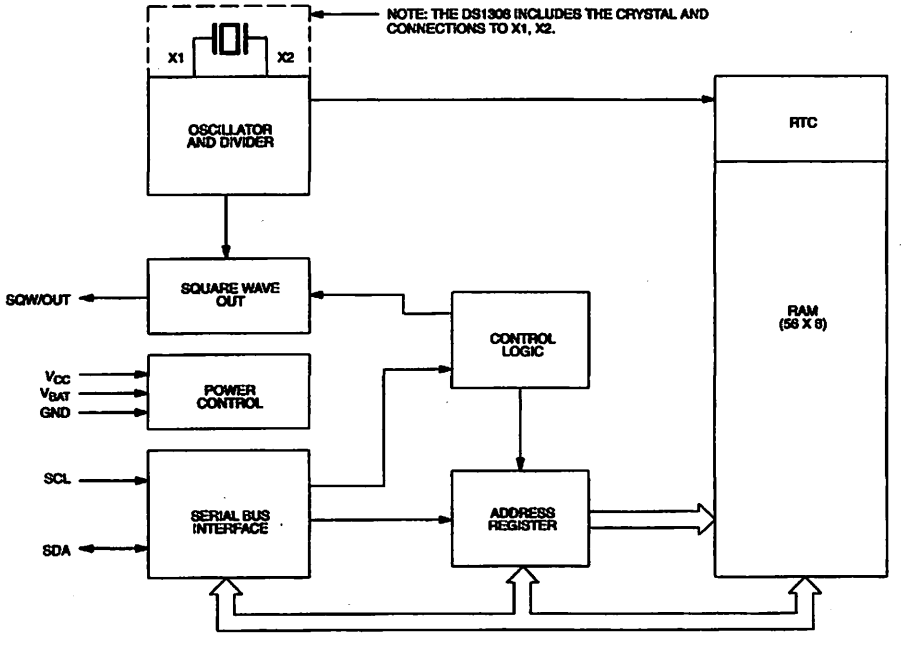
DS1307 Serial Real Time Clock is a low power, full BCD clock/calendar plus 56 bytes of volatile SRAM. Address and data are transferred serially via a 2-wire bi-directional bus. The ck/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the nth date is automatically adjusted for months with less than 31 days, including corrections for leap r. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 a built-in power sense circuit which detects power failures and automatically switches to the battery ply.

DS1308 incorporates the DS1307 chip with a 32.768 kHz crystal in a surface mountable, 36-pin ball l array package (BGA). The close proximity of the embedded crystal to the high impedance crystal ut pins on the DS1307 minimizes capacitive loading and noise injection problems associated with ny other oscillator designs. The total area required for installation is less than that of one United States ze: thus, minimizing PCB space required.

## PERATION

DS1307/1308 operates as a slave device on the serial bus. Access is obtained by implementing a ART condition and providing a device identification code followed by a register address. Subsequent isters can be accessed sequentially until a STOP condition is executed. When $V_{CC}$ falls below 1.25 x $_{BAT}$ the device terminates an access in progress and resets the device address counter. Inputs to the ice will not be recognized at this time to prevent erroneous data from being written to the device from out of tolerance system. When $V_{CC}$ falls below $V_{BAT}$ the device switches into a low current battery kup mode. Upon power up, the device switches from battery to $V_{CC}$ when $V_{CC}$ is greater than $V_{BAT}$ 2V and recognizes inputs when $V_{CC}$ is greater than 1.25 x $V_{BAT}$. The block diagram in Figure 1 shows main elements of the Serial Real Time Clock.

## 1307/DS1308 BLOCK DIAGRAM Figure 1



## GNAL DESCRIPTIONS

:, **GND** - DC power is provided to the device on these pins. $V_{CC}$ is the +5 volt input. When 5 volts is lied within normal limits, the device is fully accessible and data can be written and read. When a )lt battery is connected to the device and $V_{CC}$ is below 1.25 x $V_{BAT}$, reads and writes are inhibited. vever, the Timekeeping function continues unaffected by the lower input voltage. As $V_{CC}$ falls below .T the RAM and timekeeper are switched over to the external power supply (nominal 3.0V DC) at .T.

.T - Battery input for any standard 3-volt lithium cell or other energy source. Battery voltage must be l between 2.0 and 3.5 volts for proper operation. The nominal write protect trip point voltage at which :ss to the real time clock and user RAM is denied is set by the internal circuitry as 1.25 x $V_{BAT}$ iinal. A lithium battery with 48 mAhr or greater will back up the DS1307/DS1308 for more than 10 rs in the absence of power at 25 degrees C.

L **(Serial Clock Input)** - SCL is used to synchronize data movement on the serial interface.

A **(Serial Data Input/Output)** - SDA is the input/output pin for the 2-wire serial interface. The SDA is open drain which requires an external pullup resistor.

W/OUT **(Square Wave/ Output Driver)** - When enabled, the SQWE bit set to 1, the SQW/OUT pin )uts one of four square wave frequencies (1 Hz, 4 kHz, 8 kHz, 32 kHz). The SQW/OUT pin is open n which requires an external pullup resistor. SQW/OUT will operate with either Vcc or Vbat applied.

TE: **X1, X2 are not applicable for the DS1308 or DS1308N.**

**X2** - Connections for a standard 32.768 kHz quartz crystal. The internal oscillator circuitry is gned for operation with a crystal having a specified load capacitance (CL) of 12.5 pF.

more information on crystal selection and crystal layout considerations, please consult Application e 58, "Crystal Considerations with Dallas Real Time Clocks." The DS1307 can also be driven by an rnal 32.768 kHz oscillator. In this configuration, the X1 pin is connected to the external oscillator al and the X2 pin is floated.

se review Application Note 95, "Interfacing the DS1307/DS1308 with a 8051-Compatible rocontroller" for additional information.

## C AND RAM ADDRESS MAP

address map for the RTC and RAM registers of the DS1307/DS1308 is shown in Figure 2. The real e clock registers are located in address locations 00h to 07h. The RAM registers are located in address tions 08h to 3Fh. During a multi-byte access, when the address pointer reaches 3Fh, the end of RAM ce, it wraps around to location 00h, the beginning of the clock space.

## 1307 ADDRESS MAP Figure 2

| | |
|---|---|
| 00H | SECONDS |
| | MINUTES |
| | HOURS |
| | DAY |
| | DATE |
| | MONTH |
| | YEAR |
| 07H | CONTROL |
| 08H | RAM 56 x 8 |
| 3FH | |

## OCK AND CALENDAR

time and calendar information is obtained by reading the appropriate register bytes. The real time k registers are illustrated in Figure 3. The time and calendar are set or initialized by writing the ropriate register bytes. The contents of the time and calendar registers are in the Binary-Coded imal (BCD) format. Bit 7 of Register 0 is the Clock Halt (CH) bit. When this bit is set to a 1, the illator is disabled. When cleared to a 0, the oscillator is enabled.

**ase note that the initial power on state of all registers is not defined. Therefore it is important to ble the oscillator (CH bit=0) during initial configuration.**

DS1307/DS1308 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined he 12- or 24-hour mode select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10 hour bit (20-23 rs).

a 2-wire START, the current time is transferred to a second set of registers. The time information is l from these secondary registers, while the clock may continue to run. This eliminates the need to re- l the registers in case of an update of the main registers during a read.

## 1307/DS1308 TIMEKEEPER REGISTERS Figure 3

| | BIT7 | | | | | | | BIT0 | |
|---|---|---|---|---|---|---|---|---|---|
| 00H | CH | 10 SECONDS | | | SECONDS | | | | 00-59 |
| | X | 10 MINUTES | | | MINUTES | | | | 00-59 |
| | X | 12/24 | 10 HR / A/P | 10 HR | HOURS | | | | 01-12 / 00-23 |
| | X | X | X | X | X | DAY | | | 1-7 |
| | X | X | 10 DATE | | DATE | | | | 01-28/29 / 01-30 / 01-31 |
| | X | X | X | 10 MONTH | MONTH | | | | 01-12 |
| | 10 YEAR | | | | YEAR | | | | 00-99 |
| 07H | OUT | X | X | SQWE | X | X | RS1 | RS0 | |

## )NTROL REGISTER

: DS1307/DS1308 Control Register is used to control the operation of the SQW/OUT pin.

| BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|---|---|---|---|---|---|---|---|
| OUT | X | X | SQWE | X | X | RS1 | RS0 |

T (Output control): This bit controls the output level of the SQW/OUT pin when the square wave put is disabled. If SQWE=0, the logic level on the SQW/OUT pin is 1 if OUT=1 and is 0 if OUT=0.

WE (Square Wave Enable): This bit, when set to a logic 1, will enable the oscillator output. The uency of the square wave output depends upon the value of the RS0 and RS1 bits.

(Rate Select): These bits control the frequency of the square wave output when the square wave put has been enabled. Table 1 lists the square wave frequencies that can be selected with the RS bits.

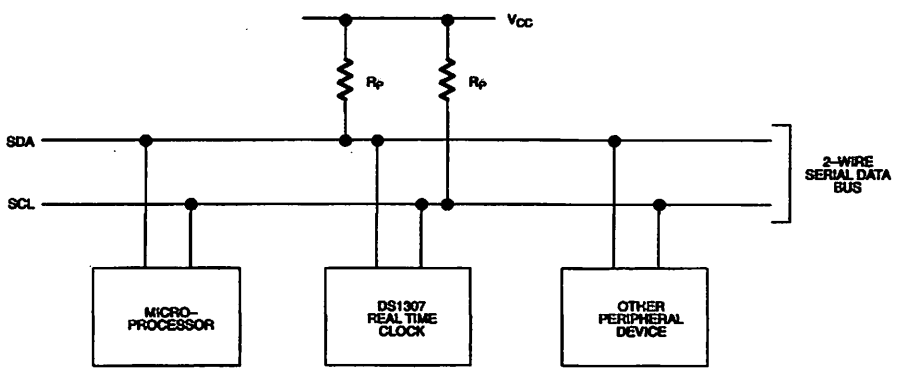
## UAREWAVE OUTPUT FREQUENCY Table 1

| RS1 | RS0 | SQW OUTPUT FREQUENCY |
|---|---|---|
| 0 | 0 | 1 Hz |
| 0 | 1 | 4.096 kHz |
| 1 | 0 | 8.192 kHz |
| 1 | 1 | 32.768 kHz |

## WIRE SERIAL DATA BUS

: DS1307 supports a bi-directional 2-wire bus and data transmission protocol. A device that sends a onto the bus is defined as a transmitter and a device receiving data as a receiver. The device that trols the message is called a master. The devices that are controlled by the master are referred to as es. The bus must be controlled by a master device which generates the serial clock (SCL), controls bus access, and generates the START and STOP conditions. The DS1307/DS1308 operates as a slave he 2-wire bus. A typical bus configuration using this 2-wire protocol is show in Figure 4.

## PICAL 2-WIRE BUS CONFIGURATION Figure 4



ures 5, 6, and 7 detail how data is transferred on the 2-wire bus.

Data transfer may be initiated only when the bus is not busy.
During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is high will be interpreted as control signals.

ordingly, the following bus conditions have been defined:

**not busy:** Both data and clock lines remain HIGH.

**rt data transfer:** A change in the state of the data line, from HIGH to LOW, while the clock is HIGH, nes a START condition.

**p data transfer:** A change in the state of the data line, from LOW to HIGH, while the clock line is GH, defines the STOP condition.

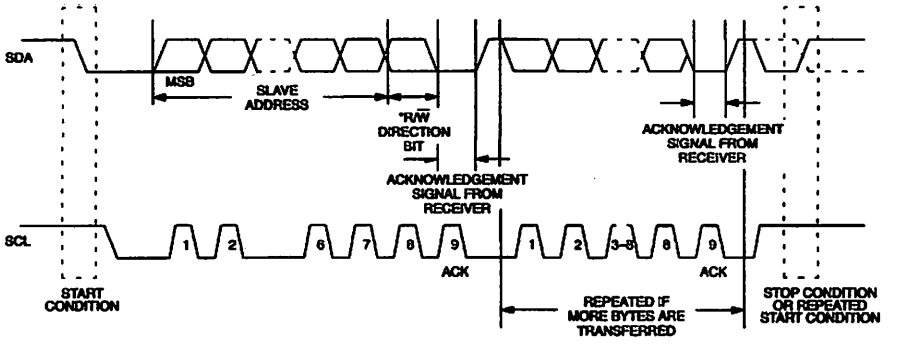
**a valid:** The state of the data line represents valid data when, after a START condition, the data line table for the duration of the HIGH period of the clock signal. The data on the line must be changed ing the LOW period of the clock signal. There is one clock pulse per bit of data.

h data transfer is initiated with a START condition and terminated with a STOP condition. The ber of data bytes transferred between START and STOP conditions is not limited, and is determined he master device. The information is transferred byte-wise and each receiver acknowledges with a h bit. Within the 2-wire bus specifications a regular mode (100 kHz clock rate) and a fast mode ) kHz clock rate) are defined. The DS1307/DS1308 operates in the regular mode (100 kHz) only.

**nowledge:** Each receiving device, when addressed, is obliged to generate an acknowledge after the ption of each byte. The master device must generate an extra clock pulse which is associated with acknowledge bit.

evice that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of rse, setup and hold times must be taken into account. A master must signal an end of data to the slave ot generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, slave must leave the data line HIGH to enable the master to generate the STOP condition.

## TA TRANSFER ON 2-WIRE SERIAL BUS Figure 5



)ending upon the state of the R/$\overline{W}$ bit, two types of data transfer are possible:

**Data transfer from a master transmitter to a slave receiver.** The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.

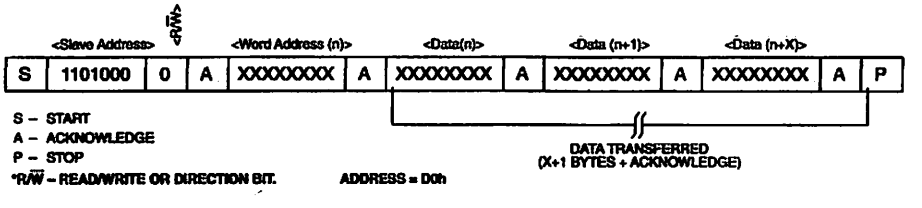
**Data transfer from a slave transmitter to a master receiver.** The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. This is followed by the slave transmitting a number of data bytes. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a 'not acknowledge' is returned.

: master device generates all of the serial clock pulses and the START and STOP conditions. A ısfer is ended with a STOP condition or with a repeated START condition. Since a repeated START dition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred ı the most significant bit (MSB) first.

: DS1307/DS1308 may operate in the following two modes:

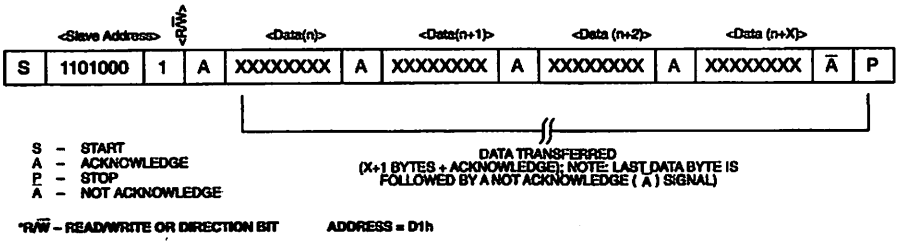
**Slave receiver mode (DS1307/DS1308 write mode):** Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and *direction bit (See Figure 6). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7 bit DS1307/DS1308 address, which is 1101000, followed by the *direction bit (R/$\overline{W}$) which, for a write, is a 0. After receiving and decoding the address byte the device outputs an acknowledge on the SDA line. After the DS1307/DS1308 acknowledges the slave address + write bit, the master transmits a register address to the DS1307/DS1308 This will set the register pointer on the DS1307/DS1308. The master will then begin transmitting each byte of data with the DS1307/DS1308 acknowledging each byte received. The master will generate a stop condition to terminate the data write.

## ıTA WRITE - SLAVE RECEIVER MODE Figure 6




**Slave transmitter mode (DS1307/DS1308 read mode):** The first byte is received and handled as in the slave receiver mode. However, in this mode, the *direction bit will indicate that the transfer direction is reversed. Serial data is transmitted on SDA by the DS1307/DS1308 while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer (See Figure 7). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7-bit DS1307/DS1308 address, which is 1101000, followed by the *direction bit (R/$\overline{W}$) which, for a read, is a 1. After receiving and decoding the address byte the device inputs an acknowledge on the SDA line. The DS1307/DS1308 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode the first address that is read is the last one stored in the register pointer. The DS1307/DS1308 must receive a Not Acknowledge to end a read.

## ATA READ - SLAVE TRANSMITTER MODE Figure 7



## ISOLUTE MAXIMUM RATINGS*

| | |
|---|---|
| tage on Any Pin Relative to Ground | -0.5V to +7.0V |
| erating Temperature | 0°C to 70°C (-40°C to 85°C for industrial) |
| rage Temperature | -55°C to +125°C |
| dering Temperature | 260°C for 10 seconds DIP (See NOTE 12)<br>See JPC/JEDEC Standard J-STD-020A for Surface Mount Devices |

his is a stress rating only and functional operation of the device at these or any other conditions above ose indicated in the operation sections of this specification is not implied. Exposure to absolute aximum rating conditions for extended periods of time may affect reliability.

Dallas Semiconductor DS1307/DS1308 is built to the highest quality standards and manufactured for g term reliability. All Dallas Semiconductor devices are made using the same quality materials and nufacturing methods. However, standard versions of the DS1307/DS1308 are not exposed to ironmental stresses, such as burn-in, that some industrial applications require. Products which have cessfully passed through this series of environmental stresses are marked IND or N, denoting their nded operating temperature and reliability rating. For specific reliability information on this product, ase contact the factory at (972) 371-4448.

## COMMENDED DC OPERATING CONDITIONS

(0°C to 70°C or -40°C to +85°C)

| RAMETER | SYMBOL | MIN | TYP | MAX | UNITS | NOTES |
|---|---|---|---|---|---|---|
| pply Voltage | $V_{CC}$ | 4.5 | 5.0 | 5.5 | V | 1 |
| gic 1 | $V_{IH}$ | 2.2 | | $V_{CC}$+0.3 | V | 1 |
| gic 0 | $V_{IL}$ | -0.3 | | +0.8 | V | 1 |
| AT Battery Voltage | $V_{BAT}$ | 2.0 | | 3.5 | V | 1 |

## ELECTRICAL CHARACTERISTICS

(0°C to 70°C or -40°C to +85°C; $V_{CC}$=4.5V to 5.5V)

| RAMETER | SYMBOL | MIN | TYP | MAX | UNITS | NOTES |
|---|---|---|---|---|---|---|
| ut Leakage | $I_{LI}$ | | | 1 | µA | 10 |
| ) Leakage | $I_{LO}$ | | | 1 | µA | 11 |
| gic 0 Output | $V_{OL}$ | | | 0.4 | V | 2 |
| tive Supply Current | $I_{CCA}$ | | | 1.5 | mA | 9 |
| ndby Current | $I_{CCS}$ | | | 200 | µA | 3 |
| ttery Current (OSC ON); W/OUT OFF | $I_{BAT1}$ | | 300 | 500 | nA | 4 |
| ttery Current (OSC ON); W/OUT ON (32 kHz) | $I_{BAT2}$ | | 480 | 800 | nA | 4 |

## ; ELECTRICAL CHARACTERISTICS

(0°C to 70°C or -40°C to +85°C; $V_{CC}$=4.5V to 5.5V)

| RAMETER | SYMBOL | MIN | TYP | MAX | UNITS | NOTES |
|---|---|---|---|---|---|---|
| Clock Frequency | $f_{SCL}$ | 0 | | 100 | kHz | |
| Free Time Between a STOP and ART Condition | $t_{BUF}$ | 4.7 | | | µs | |
| d Time (Repeated) START Condition | $t_{HD:STA}$ | 4.0 | | | µs | 5 |
| W Period of SCL Clock | $t_{LOW}$ | 4.7 | | | µs | |
| H Period of SCL Clock | $t_{HIGH}$ | 4.0 | | | µs | |
| up Time for a Repeated START dition | $t_{SU:STA}$ | 4.7 | | | µs | |
| a Hold Time | $t_{HD:DAT}$ | 0 | | | µs | 6, 7 |
| a Set-up Time | $t_{SU:DAT}$ | 250 | | | ns | |
| Time of Both SDA and SCL Signals | $t_R$ | | | 1000 | ns | |
| Time of Both SDA and SCL Signals | $t_F$ | | | 300 | ns | |
| up Time for STOP Condition | $t_{SU:STO}$ | 4.7 | | | µs | |
| acitive Load for each Bus Line | $C_B$ | | | 400 | pF | 8 |
| Capacitance | $C_{I/O}$ | | 10 | | pF | |
| stal Specified Load Capacitance | | | 12.5 | | pF | |

### TES:

All voltages are referenced to ground.

Logic zero voltages are specified at a sink current of 5 mA at $V_{CC}$=4.5V, $V_{OL}$=GND for capacitive loads.

$I_{CCS}$ specified with $V_{CC}$=5.0V and SDA, SCL=5.0V.

$V_{CC}$=0V, $V_{BAT}$=3V.

After this period, the first clock pulse is generated.

A device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the $V_{IHMIN}$ of the SCL signal) in order to bridge the undefined region of the falling edge of SCL.

The maximum $t_{HD:DAT}$ has only to be met if the device does not stretch the LOW period ($t_{LOW}$) of the SCL signal.

$C_B$ - total capacitance of one bus line in pF.

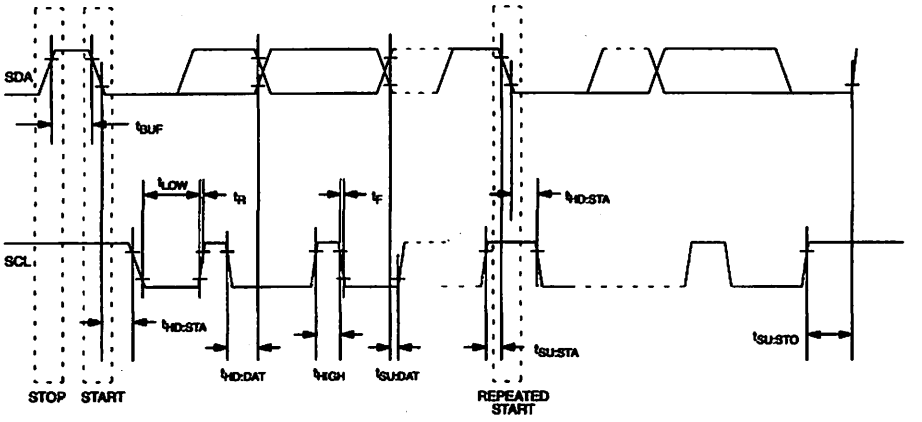
$I_{CCA}$ - SCL clocking at max frequency = 100 kHz.

SCL only.

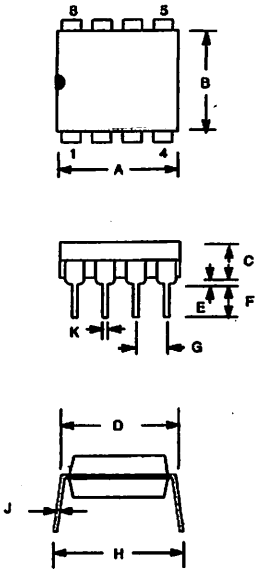
SDA and SQW/OUT

The DS1308 is designed to be subjected to no more than two passes through a solder reflow process to limit premature crystal aging effects and maintain a reasonable accuracy of ±2 minutes/month at 25 degrees C (worst case).

## MING DIAGRAM Figure 8

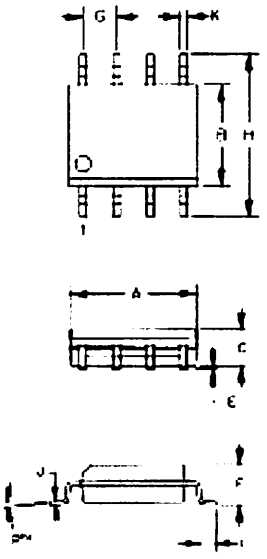



## 1307 64 X 8 SERIAL REAL TIME CLOCK
## PIN DIP MECHANICAL DIMENSIONS




| PKG | 8-PIN | |
|---|---|---|
| DIM | MIN | MAX |
| A IN. | 0.360 | 0.400 |
| MM | 9.14 | 10.16 |
| B IN. | 0.240 | 0.260 |
| MM | 6.10 | 6.60 |
| C IN. | 0.120 | 0.140 |
| MM | 3.05 | 3.56 |
| D IN. | 0.300 | 0.325 |
| MM | 7.62 | 8.26 |
| E IN. | 0.015 | 0.040 |
| MM | 0.38 | 1.02 |
| F IN. | 0.120 | 0.140 |
| MM | 3.04 | 3.56 |
| G IN. | 0.090 | 0.110 |
| MM | 2.29 | 2.79 |
| H IN. | 0.320 | 0.370 |
| MM | 8.13 | 9.40 |
| J IN. | 0.008 | 0.012 |
| MM | 0.20 | 0.30 |
| K IN. | 0.015 | 0.021 |
| MM | 0.38 | 0.53 |

# 1307Z 64 X 8 SERIAL REAL TIME CLOCK
# PIN SOIC (150-MIL) MECHANICAL DIMENSIONS

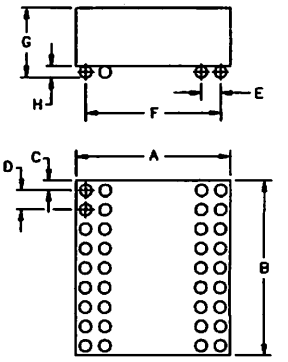



| PKG | 8-PIN (150 MIL) | |
|---|---|---|
| DIM | MIN | MAX |
| A IN. | 0.188 | 0.196 |
| MM | 4.78 | 4.98 |
| B IN. | 0.150 | 0.158 |
| MM | 3.81 | 4.01 |
| C IN. | 0.048 | 0.062 |
| MM | 1.22 | 1.57 |
| E IN. | 0.004 | 0.010 |
| MM | 0.10 | 0.25 |
| F IN. | 0.053 | 0.069 |
| MM | 1.35 | 1.75 |
| G IN. | 0.050 BSC | |
| MM | 1.27 BSC | |
| H IN. | 0.230 | 0.244 |
| MM | 5.84 | 6.20 |
| J IN. | 0.007 | 0.011 |
| MM | 0.18 | 0.28 |
| K IN. | 0.012 | 0.020 |
| MM | 0.30 | 0.51 |
| L IN. | 0.016 | 0.050 |
| MM | 0.41 | 1.27 |
| phi | 0° | 8° |

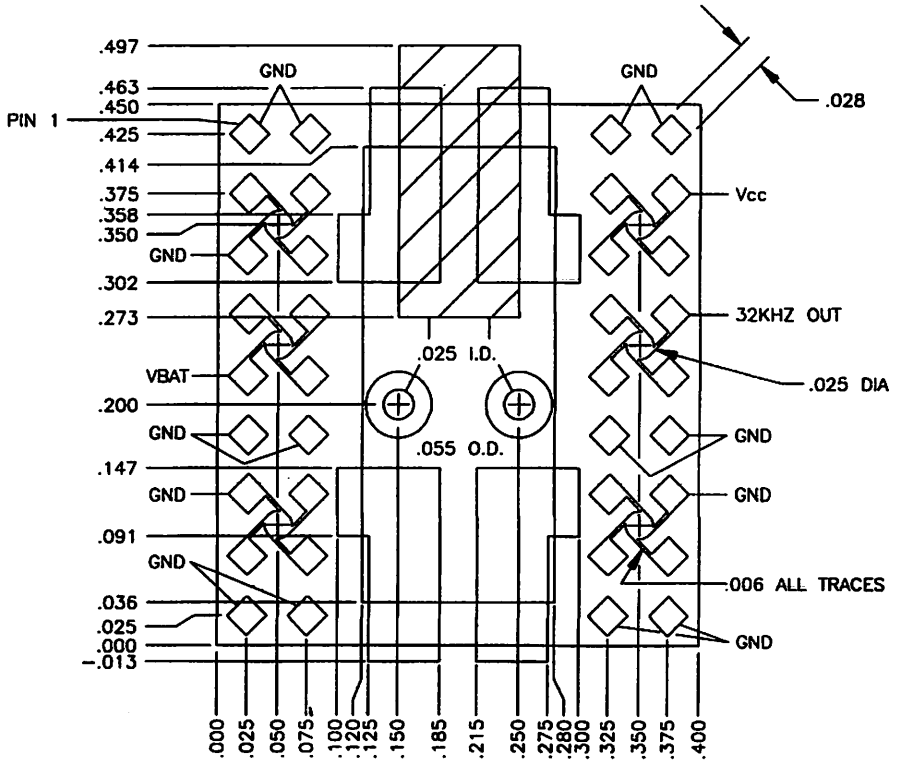
56-G2008-001

# 1308 MECHANICAL DIMENSIONS

| PKG | 36-PIN BALL GRID | |
|---|---|---|
| DIM | MIN | MAX |
| A IN. MM | 0.395 | 0.405 |
| B IN. MM | 0.445 | 0.455 |
| C IN. MM | 0.022 | 0.028 |
| D IN. MM | 0.047 | 0.053 |
| E IN. MM | 0.047 | 0.053 |
| F IN. MM | 0.347 | 0.353 |
| G IN. MM | 0.170 | 0.190 |
| H IN. MM | 0.025 | 0.030 |



# 1308 RECOMMENDED LAYOUT LAND PATTERN



## ures

- patible with MCS-51® Products
- ytes of In-System Programmable (ISP) Flash Memory
- Endurance: 1000 Write/Erase Cycles
- to 5.5V Operating Range
- Static Operation: 0 Hz to 33 MHz
- e-level Program Memory Lock
- 8-bit Internal RAM
- rogrammable I/O Lines
- e 16-bit Timer/Counters
- t Interrupt Sources
- Duplex UART Serial Channel
- power Idle and Power-down Modes
- rupt Recovery from Power-down Mode
- hdog Timer
- Data Pointer
- er-off Flag

## cription

T89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K of in-system programmable Flash memory. The device is manufactured using s high-density nonvolatile memory technology and is compatible with the indus-ndard 80C51 instruction set and pinout. The on-chip Flash allows the program ry to be reprogrammed in-system or by a conventional nonvolatile memory pro-ner. By combining a versatile 8-bit CPU with in-system programmable Flash on olithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a flexible and cost-effective solution to many embedded control applications.

T89S52 provides the following standard features: 8K bytes of Flash, 256 bytes M, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a ctor two-level interrupt architecture, a full duplex serial port, on-chip oscillator, ock circuitry. In addition, the AT89S52 is designed with static logic for operation to zero frequency and supports two software selectable power saving modes. le Mode stops the CPU while allowing the RAM, timer/counters, serial port, and pt system to continue functioning. The Power-down mode saves the RAM con-ut freezes the oscillator, disabling all other chip functions until the next interrupt dware reset.

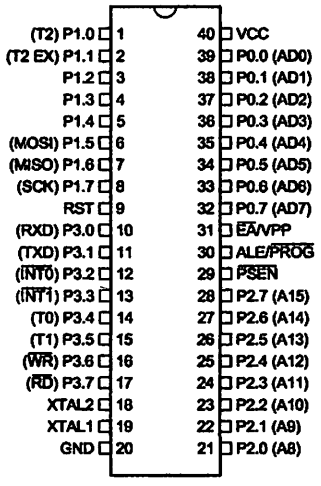
# 8-bit Microcontroller with 8K Bytes In-System Programmable Flash

# AT89S52

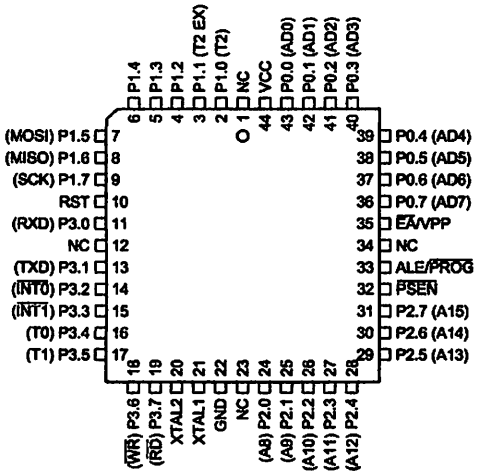
Rev. 1919A-07/01

## Configurations

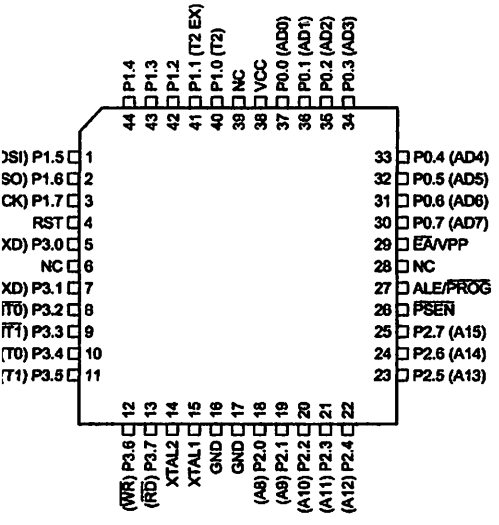
PDIP



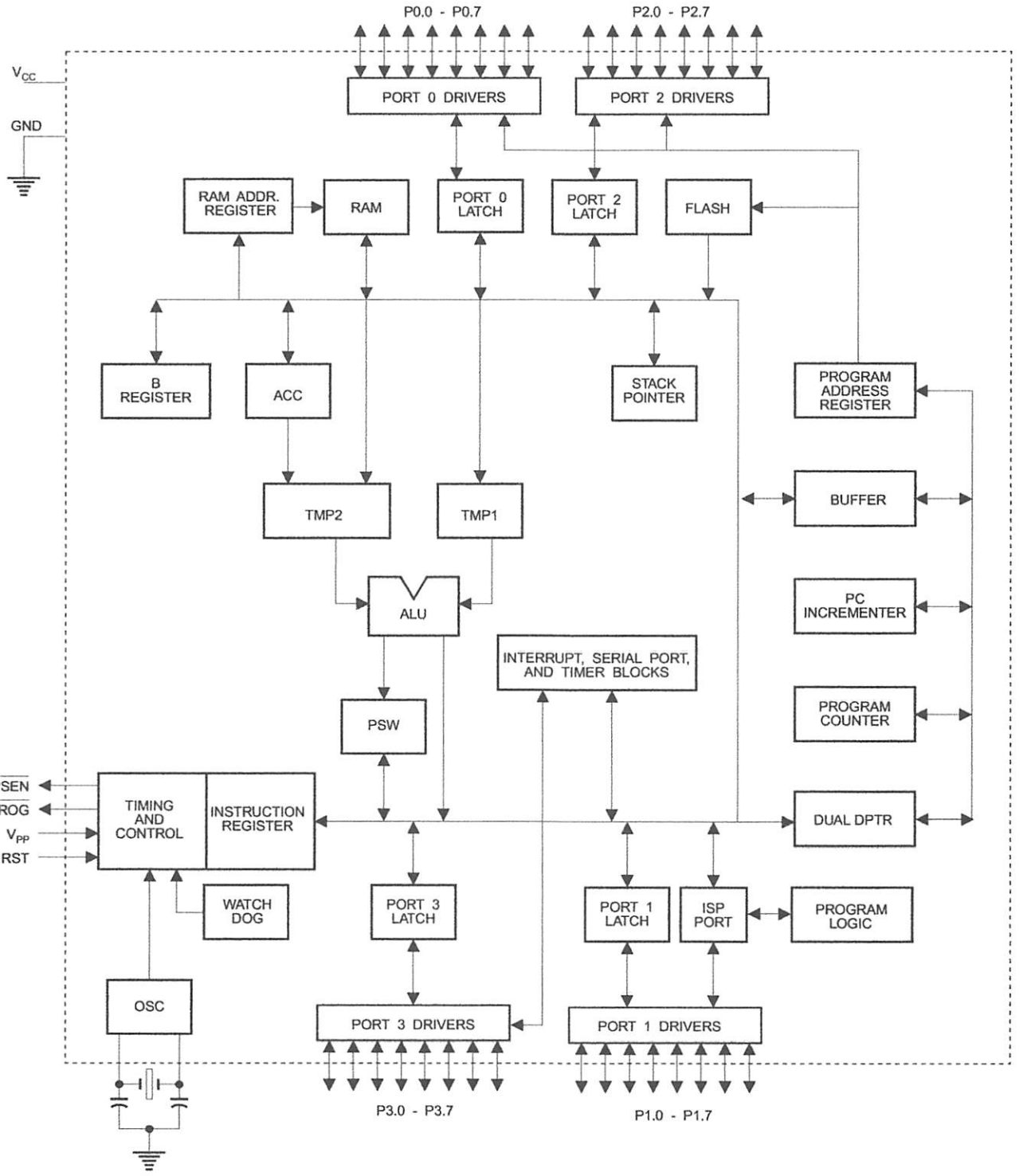

PLCC




TQFP



## ck Diagram


P0.0 - P0.7
P2.0 - P2.7
$V_{CC}$
GND
PORT 0 DRIVERS
PORT 2 DRIVERS
RAM ADDR. REGISTER
RAM
PORT 0 LATCH
PORT 2 LATCH
FLASH
B REGISTER
ACC
STACK POINTER
PROGRAM ADDRESS REGISTER
BUFFER
TMP2
TMP1
ALU
PC INCREMENTER
INTERRUPT, SERIAL PORT, AND TIMER BLOCKS
PSW
PROGRAM COUNTER
SEN
ROG
$V_{PP}$
RST
TIMING AND CONTROL
INSTRUCTION REGISTER
DUAL DPTR
WATCH DOG
PORT 3 LATCH
PORT 1 LATCH
ISP PORT
PROGRAM LOGIC
OSC
PORT 3 DRIVERS
PORT 1 DRIVERS
P3.0 - P3.7
P1.0 - P1.7

## Description

r voltage.

d.

is an 8-bit open drain bidirectional I/O port. As an port, each pin can sink eight TTL inputs. When 1s itten to port 0 pins, the pins can be used as high- ance inputs.

can also be configured to be the multiplexed low- address/data bus during accesses to external m and data memory. In this mode, P0 has internal s.

also receives the code bytes during Flash program- nd outputs the code bytes during program verifica- External pullups are required during program tion.

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

lition, P1.0 and P1.1 can be configured to be the counter 2 external count input (P1.0/T2) and the ounter 2 trigger input (P1.1/T2EX), respectively, as in the following table.

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

| Pin | Alternate Functions |
|---|---|
| | T2 (external count input to Timer/Counter 2), clock-out |
| | T2EX (Timer/Counter 2 capture/reload trigger and direction control) |
| | MOSI (used for In-System Programming) |
| | MISO (used for In-System Programming) |
| | SCK (used for In-System Programming) |

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

emits the high-order address byte during fetches external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). 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 bidirectional 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 AT89S52, 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 | $\overline{INT0}$ (external interrupt 0) |
| P3.3 | $\overline{INT1}$ (external interrupt 1) |
| P3.4 | T0 (timer 0 external input) |
| P3.5 | T1 (timer 1 external input) |
| P3.6 | $\overline{WR}$ (external data memory write strobe) |
| P3.7 | $\overline{RD}$ (external data memory read strobe) |

**RST**

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 96 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

**ALE/$\overline{PROG}$**

Address Latch Enable (ALE) 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

y pulled high. Setting the ALE-disable bit has no if the microcontroller is in external execution mode.

am Store Enable ($\overline{PSEN}$) is the read strobe to exter- ogram memory.

the AT89S52 is executing code from external pro- memory, $\overline{PSEN}$ is activated twice each machine except that two $\overline{PSEN}$ activations are skipped during access to external data memory.

**PP**

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

**XTAL1**

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

**XTAL2**

Output from the inverting oscillator amplifier.

**1.** AT89S52 SFR Map and Reset Values

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| H | | | | | | | | 0FFH |
| H | B<br>00000000 | | | | | | | 0F7H |
| H | | | | | | | | 0EFH |
| H | ACC<br>00000000 | | | | | | | 0E7H |
| H | | | | | | | | 0DFH |
| H | PSW<br>00000000 | | | | | | | 0D7H |
| H | T2CON<br>00000000 | T2MOD<br>XXXXXX00 | RCAP2L<br>00000000 | RCAP2H<br>00000000 | TL2<br>00000000 | TH2<br>00000000 | | 0CFH |
| H | | | | | | | | 0C7H |
| H | IP<br>XX000000 | | | | | | | 0BFH |
| H | P3<br>11111111 | | | | | | | 0B7H |
| H | IE<br>0X000000 | | | | | | | 0AFH |
| H | P2<br>11111111 | | AUXR1<br>XXXXXXX0 | | | | WDTRST<br>XXXXXXXX | 0A7H |
| H | SCON<br>00000000 | SBUF<br>XXXXXXXX | | | | | | 9FH |
| H | P1<br>11111111 | | | | | | | 97H |
| H | TCON<br>00000000 | TMOD<br>00000000 | TL0<br>00000000 | TL1<br>00000000 | TH0<br>00000000 | TH1<br>00000000 | AUXR<br>XXX00XX0 | 8FH |
| H | P0<br>11111111 | SP<br>00000111 | DP0L<br>00000000 | DP0H<br>00000000 | DP1L<br>00000000 | DP1H<br>00000000 | PCON<br>0XXX0000 | 87H |

## cial Function Registers

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

hat not all of the addresses are occupied, and unoc- d addresses may not be implemented on the chip. accesses to these addresses will in general return m data, and write accesses will have an indetermi- ffect.

software should not write 1s to these unlisted loca- 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 3) 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.

**Interrupt Registers:** The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the six interrupt sources in the IP register.

**2.** T2CON – Timer/Counter 2 Control Register

T2CON Address = 0C8H — Reset Value = 0000 0000B

Bit Addressable

| Bit | TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | C/$\overline{T2}$ | CP/$\overline{RL2}$ |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| ol | Function |
|---|---|
| | 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. |
| | 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). |
| K | 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 overflow to be used for the receive clock. |
| K | 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. |
| N2 | 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. |
| | Start/Stop control for Timer 2. TR2 = 1 starts the timer. |
| | Timer or counter select for Timer 2. C/$\overline{T2}$ = 0 for timer function. C/$\overline{T2}$ = 1 for external event counter (falling edge triggered). |
| L2 | Capture/Reload select. CP/$\overline{RL2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/$\overline{RL2}$ = 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. |

**3a.** AUXR: Auxiliary Register

R Address = 8EH Reset Value = XXX00XX0B

Not Bit Addressable

| | – | – | – | WDIDLE | DISRTO | – | – | DISALE |
|---|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Reserved for future expansion

LE Disable/Enable ALE

| DISALE | Operating Mode |
|---|---|
| 0 | ALE is emitted at a constant rate of 1/6 the oscillator frequency |
| 1 | ALE is active only during a MOVX or MOVC instruction |

TO Disable/Enable Reset out

| DISRTO | |
|---|---|
| 0 | Reset pin is driven High after WDT times out |
| 1 | Reset pin is input only |

LE Disable/Enable WDT in IDLE mode

| WDIDLE | |
|---|---|
| 0 | WDT continues to count in IDLE mode |
| 1 | WDT halts counting in IDLE mode |

**Data Pointer Registers:** To facilitate accessing both al and external data memory, two banks of 16-bit Pointer Registers are provided: DP0 at SFR address ons 82H-83H and DP1 at 84H-85H. Bit DPS = 0 R AUXR1 selects DP0 and DPS = 1 selects DP1. ser 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 rest under software control and is not affected by reset.

**3b.** AUXR1: Auxiliary Register 1

R1 Address = A2H Reset Value = XXXXXXX0B

Not Bit Addressable

| | – | – | – | – | – | – | – | DPS |
|---|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Reserved for future expansion

Data Pointer Register Select

| DPS | |
|---|---|
| 0 | Selects DPTR Registers DP0L, DP0H |
| 1 | Selects DPTR Registers DP1L, DP1H |

## ıory Organization

51 devices have a separate address space for Pro- and Data Memory. Up to 64K bytes each of external ım and Data Memory can be addressed.

### ram Memory

$\overline{EA}$ pin is connected to GND, all program fetches are ed to external memory.

e AT89S52, if $\overline{EA}$ is connected to $V_{CC}$, program s to addresses 0000H through 1FFFH are directed to al memory and fetches to addresses 2000H through H are to external memory.

### Memory

T89S52 implements 256 bytes of on-chip RAM. The 128 bytes occupy a parallel address space to the al Function Registers. This means that the upper 128 have the same addresses as the SFR space but are ally 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 which use direct addressing access of the 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.

## chdog Timer
## e-time Enabled with Reset-out)

/DT is intended as a recovery method in situations the CPU may be subjected to software upsets. The consists of a 13-bit counter and the Watchdog Timer (WDTRST) SFR. The WDT is defaulted to disable xiting reset. To enable the WDT, a user must write and 0E1H in sequence to the WDTRST register location 0A6H). When the WDT is enabled, it will ment every machine cycle while the oscillator is run- The WDT timeout period is dependent on the external frequency. There is no way to disable the WDT t through reset (either hardware reset or WDT over- eset). When WDT overflows, it will drive an output T HIGH pulse at the RST pin.

## g the WDT

able the WDT, a user must write 01EH and 0E1H in nce to the WDTRST register (SFR location 0A6H). the WDT is enabled, the user needs to service it by 01EH and 0E1H to WDTRST to avoid a WDT over- The 13-bit counter overflows when it reaches 8191 H), and this will reset the device. When the WDT is ed, it will increment every machine cycle while the tor is running. This means the user must reset the at least every 8191 machine cycles. To reset the the user must write 01EH and 0E1H to WDTRST. RST is a write-only register. The WDT counter cannot d or written. When WDT overflows, it will generate an RESET pulse at the RST pin. The RESET pulse on is 96xTOSC, where TOSC=1/FOSC. To make the se of the WDT, it should be serviced in those sec- of code that will periodically be executed within the equired to prevent a WDT reset.

## T During Power-down and Idle

ver-down mode the oscillator stops, which means the also stops. While in Power-down mode, the user not need to service the WDT. There are two methods ing Power-down mode: by a hardware reset or via a activated external interrupt which is enabled prior to ng Power-down mode. When Power-down is exited ardware reset, servicing the WDT should occur as it lly does whenever the AT89S52 is reset. Exiting -down with an interrupt is significantly different. The pt is held low long enough for the oscillator to stabi- When the interrupt is brought high, the interrupt is ed. To prevent the WDT from resetting the device the interrupt pin is held low, the WDT is not started ne interrupt is pulled high. It is suggested that the be reset during the interrupt service for the interrupt o exit Power-down mode.

To ensure that the WDT does not overflow within a few states of exiting Power-down, it is best to reset the WDT just before entering Power-down mode.

Before going into the IDLE mode, the WDIDLE bit in SFR AUXR is used to determine whether the WDT continues to count if enabled. The WDT keeps counting during IDLE (WDIDLE bit = 0) as the default state. To prevent the WDT from resetting the AT89S52 while in IDLE mode, the user should always set up a timer that will periodically exit IDLE, service the WDT, and reenter IDLE mode.

With WDIDLE bit enabled, the WDT will stop to count in IDLE mode and resumes the count upon exit from IDLE.

## UART

The UART in the AT89S52 operates the same way as the UART in the AT89C51 and AT89C52. For further information on the UART operation, refer to the ATMEL Web site (http://www.atmel.com). From the home page, select 'Products', then '8051-Architecture Flash Microcontroller', then 'Product Overview'.

## Timer 0 and 1

Timer 0 and Timer 1 in the AT89S52 operate the same way as Timer 0 and Timer 1 in the AT89C51 and AT89C52. For further information on the timers' operation, refer to the ATMEL Web site (http://www.atmel.com). From the home page, select 'Products', then '8051-Architecture Flash Microcontroller', then 'Product Overview'.

## 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/$\overline{T2}$ 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 3. 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.

**Table 3.** Timer 2 Operating Modes

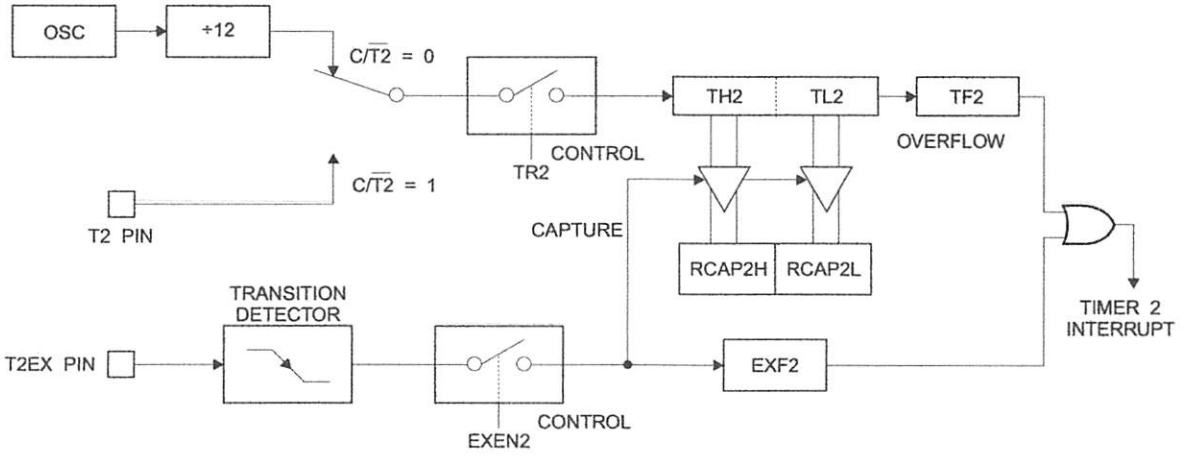
| RCLK +TCLK | CP/$\overline{RL2}$ | TR2 | MODE |
|---|---|---|---|
| 0 | 0 | 1 | 16-bit Auto-reload |
| 0 | 1 | 1 | 16-bit Capture |
| 1 | X | 1 | Baud Rate Generator |
| X | X | 0 | (Off) |

Counter function, the register is incremented in ıse to a 1-to-0 transition at its corresponding external ›in, T2. In this function, the external input is sampled S5P2 of every machine cycle. When the samples a high in one cycle and a low in the next cycle, the is incremented. The new count value appears in the er during S3P1 of the cycle following the one in which ınsition was detected. Since two machine cycles (24 tor periods) are required to recognize a 1-to-0 transi- ne maximum count rate is 1/24 of the oscillator fre- y. To ensure that a given level is sampled at least ›efore it changes, the level should be held for at least ll machine cycle.

### ure Mode

capture mode, two options are selected by bit 2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer ınter which upon overflow sets bit TF2 in T2CON.

**e 5.** Timer in Capture Mode




e 6 shows Timer 2 automatically counting up when l=0. In this mode, two options are selected by bit 2 in T2CON. If EXEN2 = 0, Timer 2 counts up to FH and then sets the TF2 bit upon overflow. The ›w also causes the timer registers to be reloaded with -bit value in RCAP2H and RCAP2L. The values in in Capture ModeRCAP2H and RCAP2L are preset tware. If EXEN2 = 1, a 16-bit reload can be triggered by an overflow or by a 1-to-0 transition at external T2EX. This transition also sets the EXF2 bit. Both the nd EXF2 bits can generate an interrupt if enabled.

g the DCEN bit enables Timer 2 to count up or down, ›wn in Figure 6. In this mode, the T2EX pin controls 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 5.

### Auto-reload (Up or Down Counter)

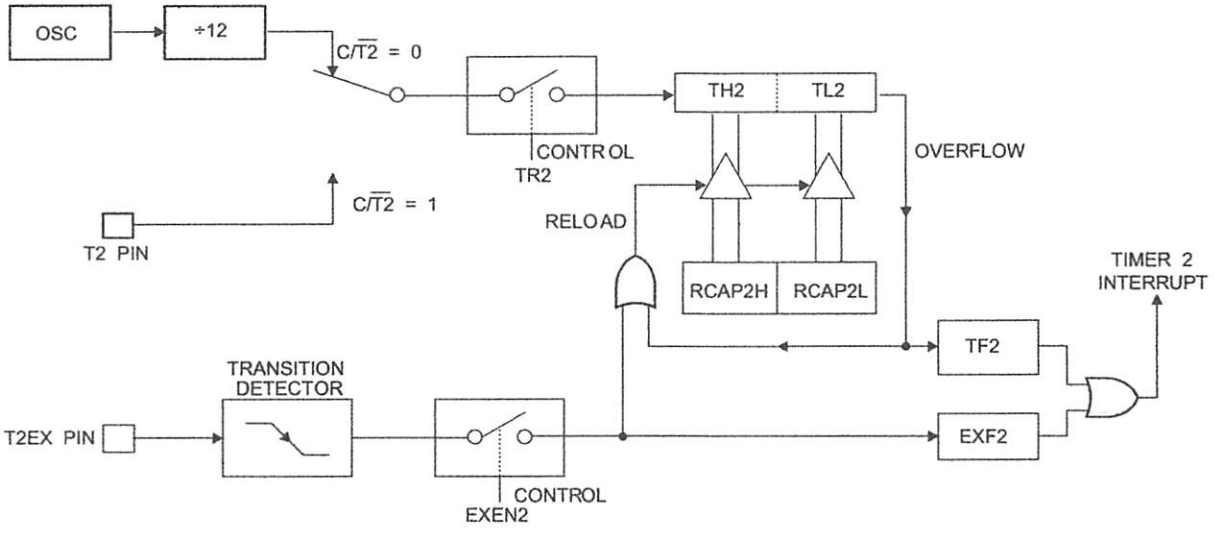
Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 4). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

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.

**e 6.** Timer 2 Auto Reload Mode (DCEN = 0)




**4.** T2MOD – Timer 2 Mode Control Register

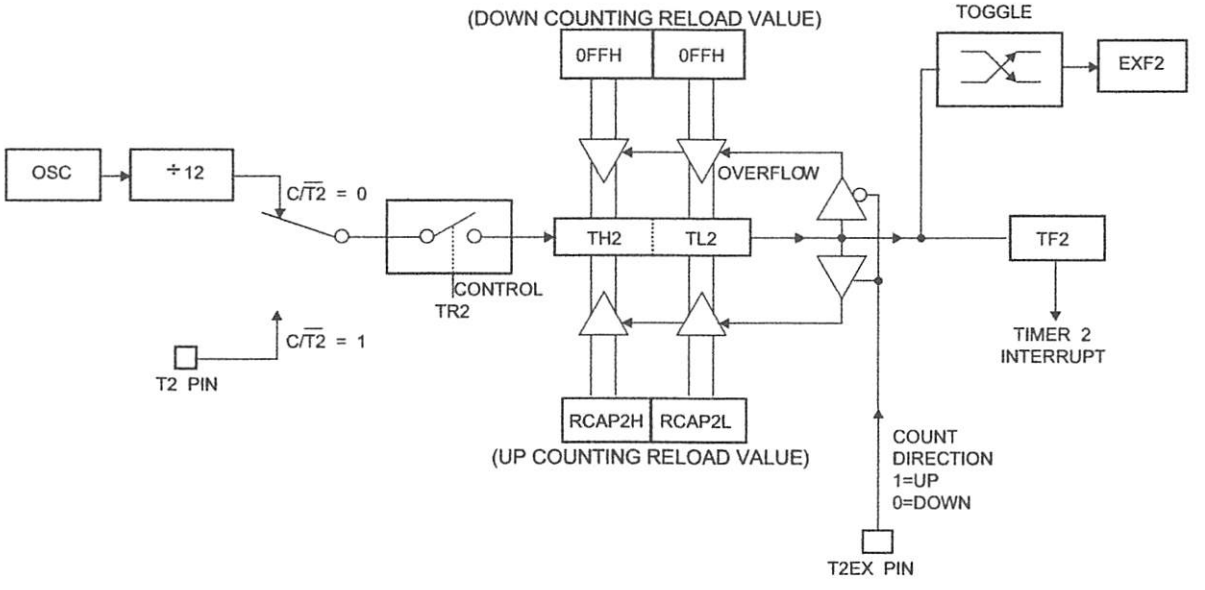
T2MOD Address = 0C9H — Reset Value = XXXX XX00B

Not Bit Addressable

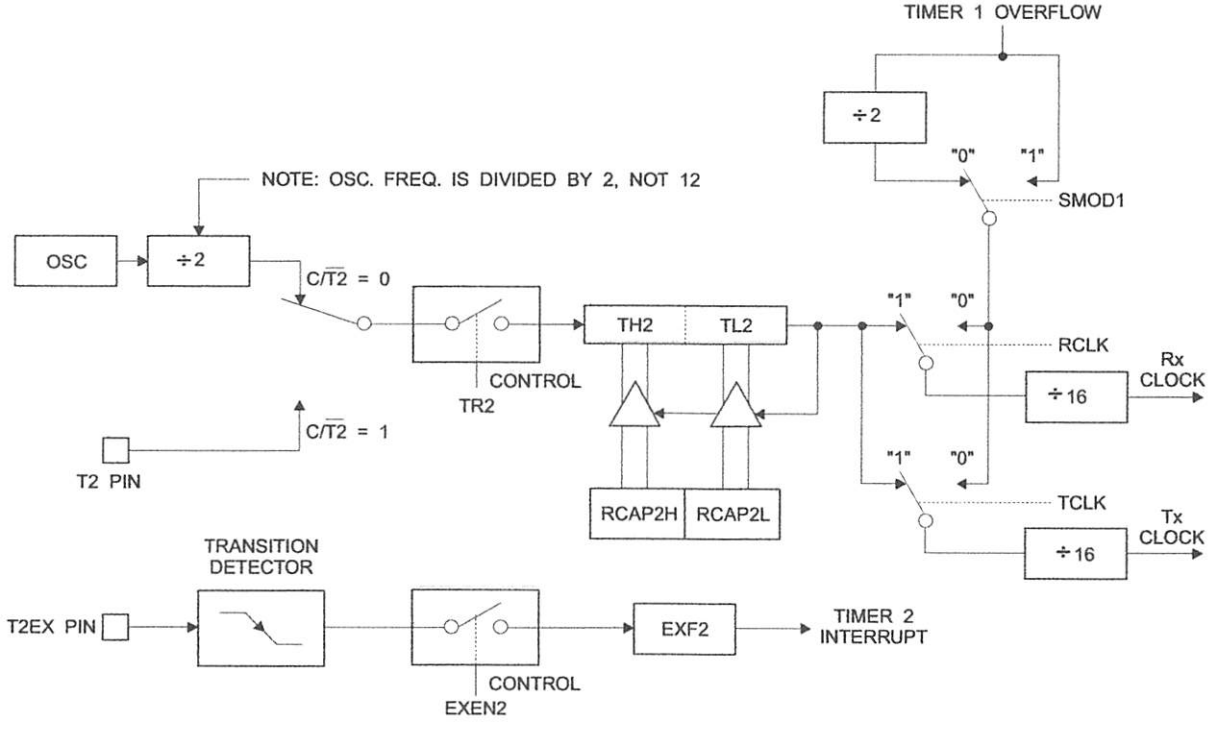
| Bit | – | – | – | – | – | – | T2OE | DCEN |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| | Not implemented, reserved for future |
| T | Timer 2 Output Enable bit |
| DCEN | When set, this bit allows Timer 2 to be configured as an up/down counter |

**e 7.** Timer 2 Auto Reload Mode (DCEN = 1)




**e 8.** Timer 2 in Baud Rate Generator Mode



## d Rate Generator

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

aud rate generator mode is similar to the auto-reload in that a rollover in TH2 causes the Timer 2 registers reloaded with the 16-bit value in registers RCAP2H CAP2L, which are preset by software.

aud rates in Modes 1 and 3 are determined by Timer erflow rate according to the following equation.

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

imer can be configured for either timer or counter tion. In most applications, it is configured for timer tion ($CP/\overline{T2} = 0$). The timer operation is different for 2 when it is used as a baud rate generator. Normally, mer, it increments every machine cycle (at 1/12 the tor 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.

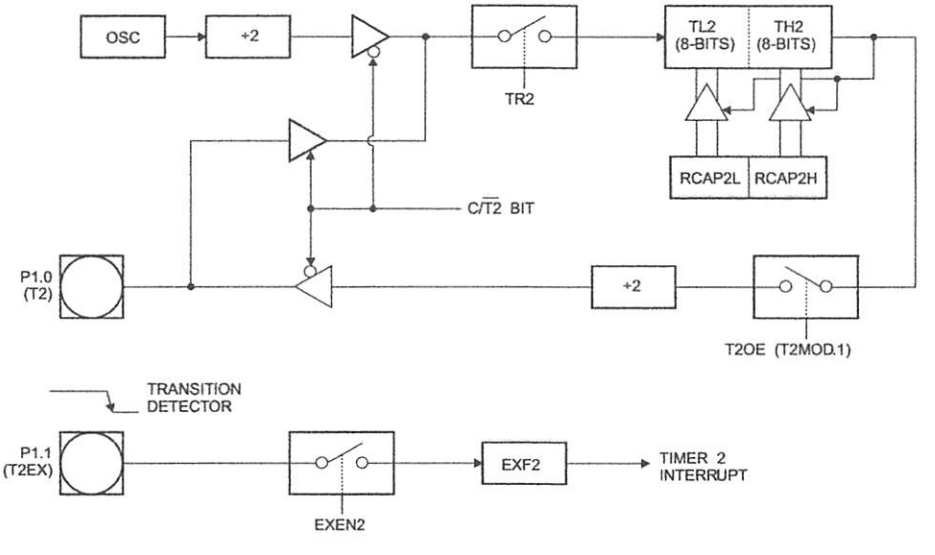
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536\text{-RCAP2H,RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 8. This figure 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 from (RCAP2H, RCAP2L) to (TH2, TL2). Thus, when Timer 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.

**9.** Timer 2 in Clock-Out Mode



## grammable Clock Out

% duty cycle clock can be programmed to come out on as shown in Figure 9. This pin, besides being a regu- O pin, has two alternate functions. It can be pro- ned to input the external clock for Timer/Counter 2 or put a 50% duty cycle clock ranging from 61 Hz to 4 t a 16 MHz operating frequency.

nfigure the Timer/Counter 2 as a clock generator, bit T2CON.1) must be cleared and bit T2OE (T2MOD.1) e set. Bit TR2 (T2CON.2) starts and stops the timer.

lock-out frequency depends on the oscillator fre- y and the reload value of Timer 2 capture registers P2H, RCAP2L), as shown in the following equation.

$$\text{k-Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536-(\text{RCAP2H,RCAP2L})]}$$

clock-out mode, Timer 2 roll-overs will not generate errupt. This behavior is similar to when Timer 2 is as a baud-rate generator. It is possible to use Timer 2 baud-rate generator and a clock generator simulta- ly. Note, however, that the baud-rate and clock-out ncies cannot be determined independently from one er since they both use RCAP2H and RCAP2L.

## rrupts

T89S52 has a total of six interrupt vectors: two exter- errupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (Tim- , 1, and 2), and the serial port interrupt. These pts are all shown in Figure 10.

of these interrupt sources can be individually enabled abled by setting or clearing a bit in Special Function er IE. IE also contains a global disable bit, EA, which es all interrupts at once.

hat Table 5 shows that bit position IE.6 is unimple- d. In the AT89S52, bit position IE.5 is also unimple- d. User software should not write 1s to these bit ns, since they may be used in future AT89 products.

2 interrupt is generated by the logical OR of bits TF2 XF2 in register T2CON. Neither of these flags is d by hardware when the service routine is vectored fact, the service routine may have to determine er it was TF2 or EXF2 that generated the interrupt, at bit will have to be cleared in software.

imer 0 and Timer 1 flags, TF0 and TF1, are set at of the cycle in which the timers overflow. The values en polled by the circuitry in the next cycle. However, mer 2 flag, TF2, is set at S2P2 and is polled in the cycle in which the timer overflows.

**Table 5.** Interrupt Enable (IE) Register

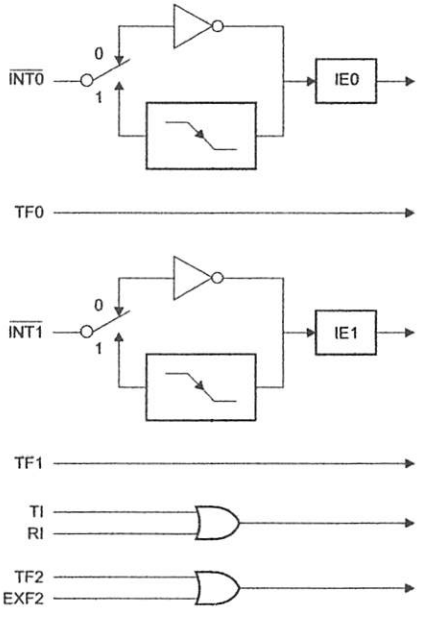
| (MSB) | | | | | | | (LSB) |
|---|---|---|---|---|---|---|---|
| EA | – | ET2 | ES | ET1 | EX1 | ET0 | EX0 |

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 | Serial Port 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



## illator Characteristics

l and XTAL2 are the input and output, respectively, nverting amplifier that can be configured for use as chip oscillator, as shown in Figure 11. Either a quartz l or ceramic resonator may be used. To drive the from an external clock source, XTAL2 should be left nected while XTAL1 is driven, as shown in Figure 12. are no requirements on the duty cycle of the external signal, since the input to the internal clocking circuitry ugh a divide-by-two flip-flop, but minimum and maxi- voltage high and low time specifications must be ed.

## Mode

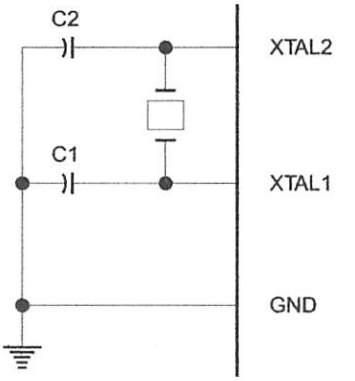
mode, the CPU puts itself to sleep while all the on- eripherals remain active. The mode is invoked by re. The content of the on-chip RAM and all the spe- nctions registers remain unchanged during this The idle mode can be terminated by any enabled pt or by a hardware reset.

hat when idle mode is terminated by a hardware the device normally resumes program execution here it left off, up to two machine cycles before the al reset algorithm takes control. On-chip hardware s access to internal RAM in this event, but access to rt pins is not inhibited. To eliminate the possibility of expected write to a port pin when idle mode is termi- by a reset, the instruction following the one that s idle mode should not write to a port pin or to exter- mory.

## er-down Mode

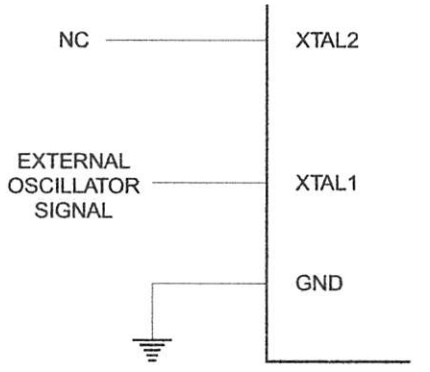
Power-down mode, the oscillator is stopped, and the tion that invokes Power-down is the last instruction ted. The on-chip RAM and Special Function Regis- tain their values until the Power-down mode is termi- Exit from Power-down mode can be initiated either ardware reset or by an enabled external interrupt. redefines the SFRs but does not change the on-chip The reset should not be activated before $V_{CC}$ is ed to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

**Figure 11.** Oscillator Connections




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

**Figure 12.** External Clock Drive Configuration




**6.** *Status of External Pins During Idle and Power-down Modes*

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

## ıram Memory Lock Bits

T89S52 has three lock bits that can be left unpro-ıed (U) or can be programmed (P) to obtain the addi-eatures listed in the following table.

7. Lock Bit Protection Modes

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

lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin pled and latched during reset. If the device is pow-p without a reset, the latch initializes to a random and holds that value until reset is activated. The l value of $\overline{EA}$ must agree with the current logic level pin in order for the device to function properly.

## ıramming the Flash – Parallel Mode

T89S52 is shipped with the on-chip Flash memory eady to be programmed. The programming interface a high-voltage (12-volt) program enable signal and ıpatible with conventional third-party Flash or M programmers.

T89S52 code memory array is programmed byte-by-

**amming Algorithm:** Before programming the 52, the address, data, and control signals should be according to the Flash programming mode table and s 13 and 14. To program the AT89S52, take the fol-steps:

ıut the desired memory location on the address ɔs.

ıut the appropriate data byte on the data lines.

tivate the correct combination of control signals.

ise $\overline{EA}/V_{PP}$ to 12V.

lse ALE/$\overline{PROG}$ once to program a byte in the ısh array or the lock bits. The byte-write cycle is lf-timed and typically takes no more than 50 µs.

Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

**Data Polling:** The AT89S52 features Data Polling to indicate the end of a byte write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P0.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

**Ready/Busy:** The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.0 is pulled low after ALE goes high during programming to indicate $\overline{BUSY}$. P3.0 is pulled high again when programming is done to indicate READY.

**Program Verify:** If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The status of the individual lock bits can be verified directly by reading them back.

**Reading the Signature Bytes:** The signature bytes are read by the same procedure as a normal verification of locations 000H, 100H, and 200H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(000H) = 1EH indicates manufactured by Atmel
(100H) = 52H indicates 89S52
(200H) = 06H

**Chip Erase:** In the parallel programming mode, a chip erase operation is initiated by using the proper combination of control signals and by pulsing ALE/$\overline{PROG}$ low for a duration of 200 ns - 500 ns.

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 500 ms.

During chip erase, a serial read from any address location will return 00H at the data output.

## Programming the Flash – Serial Mode

The Code memory array can be programmed using the serial ISP interface while RST is pulled to $V_{CC}$. 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 other operations can be executed. Before a reprogramming sequence can occur, a Chip Erase operation is required.

The Chip Erase operation turns the content of every memory location in the Code array into FFH.

Either an external system clock can be supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK)

ency should be less than 1/16 of the crystal fre-
y. With a 33 MHz oscillator clock, the maximum SCK
ncy is 2 MHz.

## l Programming Algorithm

gram and verify the AT89S52 in the serial program-
node, the following sequence is recommended:

wer-up sequence:

pply power between VCC and GND pins.

t RST pin to "H".

a crystal is not connected across pins XTAL1 and
AL2, apply a 3 MHz to 33 MHz clock to XTAL1 pin
d wait for at least 10 milliseconds.

able serial programming by sending the Pro-
amming Enable serial instruction to pin
SI/P1.5. The frequency of the shift clock sup-
ed at pin SCK/P1.7 needs to be less than the
PU clock at XTAL1 divided by 16.

e Code array is programmed one byte at a time
supplying the address and data together with the appropriate Write instruction. The write cycle is self-timed and typically takes less than 1 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 device operation.

Power-off sequence (if needed):

Set XTAL1 to "L" (if a crystal is not used).

Set RST to "L".

Turn $V_{CC}$ power off.

**Data Polling:** The Data Polling feature is also available in the serial mode. In this mode, during a write cycle an attempted read of the last byte written will result in the complement of the MSB of the serial output byte on MISO.

## Serial Programming Instruction Set

The Instruction Set for Serial Programming follows a 4-byte protocol and is shown in Table 10.

## gramming Interface – Parallel Mode

code byte in the Flash array can be programmed by the appropriate combination of control signals. The operation cycle is self-timed and once initiated, will atically time itself to completion.

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

**8.** Flash Programming Modes

| | $V_{CC}$ | RST | $\overline{PSEN}$ | ALE/ $\overline{PROG}$ | $\overline{EA}$/ $V_{PP}$ | P2.6 | P2.7 | P3.3 | P3.6 | P3.7 | P0.7-0 Data | P2.4-0 Address | P1.7-0 Address |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Code Data | 5V | H | L | (2) | 12V | L | H | H | H | H | $D_{IN}$ | A12-8 | A7-0 |
| Code Data | 5V | H | L | H | H | L | L | L | H | H | $D_{OUT}$ | A12-8 | A7-0 |
| Lock Bit 1 | 5V | H | L | (3) | 12V | H | H | H | H | H | X | X | X |
| Lock Bit 2 | 5V | H | L | (3) | 12V | H | H | H | L | L | X | X | X |
| Lock Bit 3 | 5V | H | L | (3) | 12V | H | L | H | H | L | X | X | X |
| Lock Bits | 5V | H | L | H | H | H | H | L | H | L | P0.2, P0.3, P0.4 | X | X |
| Erase | 5V | H | L | (1) | 12V | H | L | H | L | L | X | X | X |
| Atmel ID | 5V | H | L | H | H | L | L | L | L | L | 1EH | X 0000 | 00H |
| Device ID | 5V | H | L | H | H | L | L | L | L | L | 52H | X 0001 | 00H |
| Device ID | 5V | H | L | H | H | L | L | L | L | L | 06H | X 0010 | 00H |

1. Each $\overline{PROG}$ pulse is 200 ns - 500 ns for Chip Erase.
2. Each $\overline{PROG}$ pulse is 200 ns - 500 ns for Write Code Data.
3. Each $\overline{PROG}$ pulse is 200 ns - 500 ns for Write Lock Bits.
4. RDY/$\overline{BSY}$ signal is output on P3.0 during programming.
5. X = don't care.

**e 13.** Programming the Flash Memory (Parallel Mode)

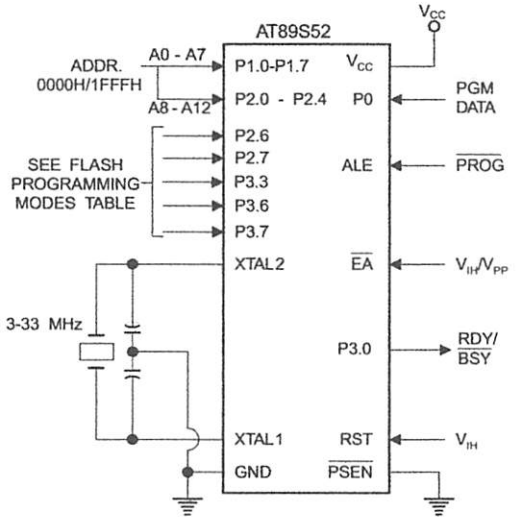
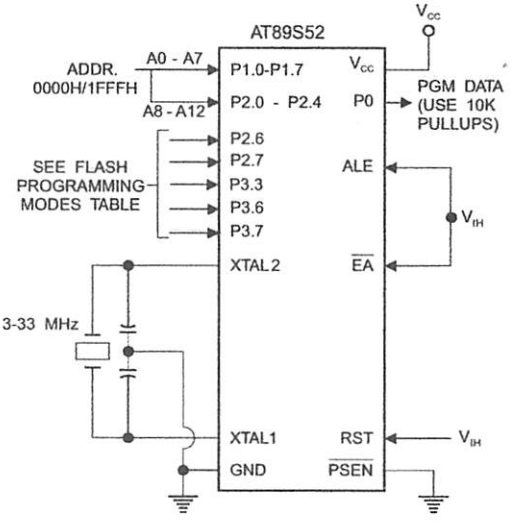



**Figure 14.** Verifying the Flash Memory (Parallel Mode)

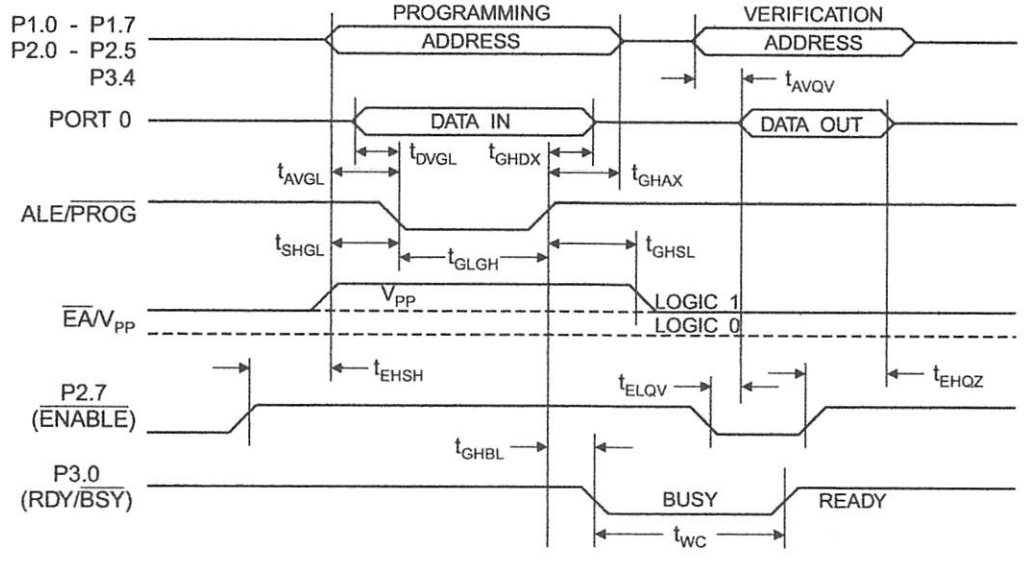


## h Programming and Verification Characteristics (Parallel Mode)

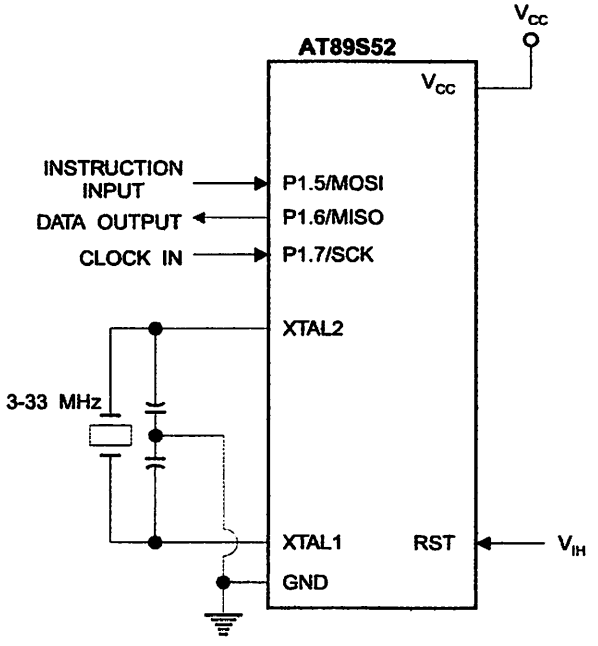
0°C to 30°C, $V_{CC}$ = 4.5 to 5.5V

| ol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| | Programming Supply Voltage | 11.5 | 12.5 | V |
| | Programming Supply Current | | 10 | mA |
| | $V_{CC}$ Supply Current | | 30 | mA |
| L | Oscillator Frequency | 3 | 33 | MHz |
| | Address Setup to $\overline{PROG}$ Low | $48t_{CLCL}$ | | |
| | Address Hold After $\overline{PROG}$ | $48t_{CLCL}$ | | |
| | Data Setup to $\overline{PROG}$ Low | $48t_{CLCL}$ | | |
| | Data Hold After $\overline{PROG}$ | $48t_{CLCL}$ | | |
| | P2.7 ($\overline{ENABLE}$) High to $V_{PP}$ | $48t_{CLCL}$ | | |
| | $V_{PP}$ Setup to $\overline{PROG}$ Low | 10 | | µs |
| | $V_{PP}$ Hold After $\overline{PROG}$ | 10 | | µs |
| | $\overline{PROG}$ Width | 0.2 | 1 | µs |
| | Address to Data Valid | | $48t_{CLCL}$ | |
| | $\overline{ENABLE}$ Low to Data Valid | | $48t_{CLCL}$ | |
| | Data Float After $\overline{ENABLE}$ | 0 | $48t_{CLCL}$ | |
| | $\overline{PROG}$ High to $\overline{BUSY}$ Low | | 1.0 | µs |
| | Byte Write Cycle Time | | 50 | µs |

**15.** Flash Programming and Verification Waveforms – Parallel Mode

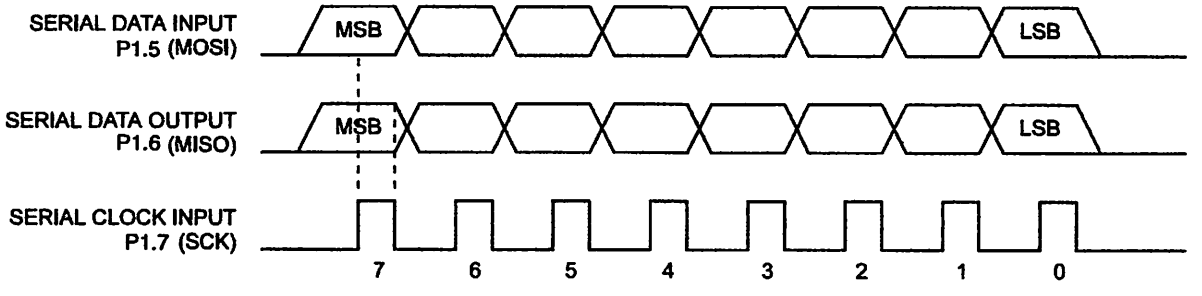


**16.** Flash Memory Serial Downloading




## h Programming and Verification Waveforms – Serial Mode

**17.** Serial Programming Waveforms



**9.** Serial Programming Instruction Set

| uction | Instruction Format<br>Byte 1 | Byte 2 | Byte 3 | Byte 4 | Operation |
|---|---|---|---|---|---|
| ramming Enable | 1010 1100 | 0101 0011 | xxxx xxxx | xxxx xxxx<br>0110 1001<br>(Output) | Enable Serial Programming while RST is high |
| Erase | 1010 1100 | 100x xxxx | xxxx xxxx | xxxx xxxx | Chip Erase Flash memory array |
| Program Memory Mode) | 0010 0000 | xxx A12 A11 A10 A9 A8 | A7 A6 A5 A4 A3 A2 A1 A0 | D7 D6 D5 D4 D3 D2 D1 D0 | Read data from Program memory in the byte mode |
| Program Memory Mode) | 0100 0000 | xxx A12 A11 A10 A9 A8 | A7 A6 A5 A4 A3 A2 A1 A0 | D7 D6 D5 D4 D3 D2 D1 D0 | Write data to Program memory in the byte mode |
| Lock Bits[2] | 1010 1100 | 1110 00 B1 B2 | xxxx xxxx | xxxx xxxx | Write Lock bits. See Note (2). |
| Lock Bits | 0010 0100 | xxxx xxxx | xxxx xxxx | xx LB3 LB2 LB1 xx | Read back current status of the lock bits (a programmed lock bit reads back as a '1') |
| Signature Bytes[1] | 0010 1000 | xxx A5 A4 A3 A2 A1 A0 | xxx xxxx | Signature Byte | Read Signature Byte |
| Program Memory Mode) | 0011 0000 | xxx A12 A11 A10 A9 A8 | Byte 0 | Byte 1...<br>Byte 255 | Read data from Program memory in the Page Mode (256 bytes) |
| Program Memory Mode) | 0101 0000 | xxx A12 A11 A10 A9 A8 | Byte 0 | Byte 1...<br>Byte 255 | Write data to Program memory in the Page Mode (256 bytes) |

1. The signature bytes are not readable in Lock Bit Modes 3 and 4.
2. B1 = 0, B2 = 0 ---> Mode 1, no lock protection
   B1 = 0, B2 = 1 ---> Mode 2, lock bit 1 activated
   B1 = 1, B2 = 0 ---> Mode 3, lock bit 2 activated
   B1 = 1, B1 = 1 ---> Mode 4, lock bit 3 activated

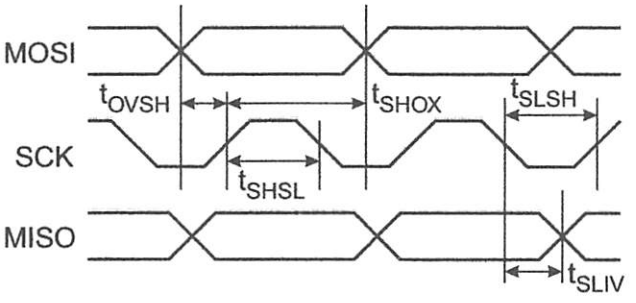
   Each of the lock bits needs to be activated sequentially before Mode 4 can be executed.

Reset signal is high, SCK should be low for at least 64 n clocks before it goes high to clock in the enable ytes. No pulsing of Reset signal is necessary. SCK d be no faster than 1/16 of the system clock at .

For Page Read/Write, the data always starts from byte 0 to 255. After the command byte and upper address byte are latched, each byte thereafter is treated as data until all 256 bytes are shifted in/out. Then the next instruction will be ready to be decoded.

## l Programming Characteristics

**18.** Serial Programming Timing




**10.** Serial Programming Characteristics, $T_A$ = -40° C to 85° C, $V_{CC}$ = 4.0 - 5.5V (Unless otherwise noted)

| ol | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| L | Oscillator Frequency | 0 | | 33 | MHz |
| | Oscillator Period | 30 | | | ns |
| | SCK Pulse Width High | 2 $t_{CLCL}$ | | | ns |
| | SCK Pulse Width Low | 2 $t_{CLCL}$ | | | ns |
| | MOSI Setup to SCK High | $t_{CLCL}$ | | | ns |
| | MOSI Hold after SCK High | 2 $t_{CLCL}$ | | | ns |
| | SCK Low to MISO Valid | 10 | 16 | 32 | ns |
| E | Chip Erase Instruction Cycle Time | | | 500 | ms |
| | Serial Byte Write Cycle Time | | | 64 $t_{CLCL}$ + 400 | µs |

## olute Maximum Ratings*

| | |
|---|---|
| ating Temperature | -55°C to +125°C |
| ge Temperature | -65°C to +150°C |
| ge on Any Pin Respect to Ground | -1.0V to +7.0V |
| num Operating Voltage | 6.6V |
| utput 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.

## Characteristics

alues shown in this table are valid for $T_A$ = -40°C to 85°C and $V_{CC}$ = 4.0V to 5.5V, unless otherwise noted.

| ol | Parameter | Condition | Min | Max | Units |
|---|---|---|---|---|---|
| | Input Low Voltage | (Except $\overline{EA}$) | -0.5 | 0.2 $V_{CC}$-0.1 | V |
| | Input Low Voltage ($\overline{EA}$) | | -0.5 | 0.2 $V_{CC}$-0.3 | V |
| | Input High Voltage | (Except XTAL1, RST) | 0.2 $V_{CC}$+0.9 | $V_{CC}$+0.5 | V |
| | Input High Voltage | (XTAL1, RST) | 0.7 $V_{CC}$ | $V_{CC}$+0.5 | V |
| | Output Low Voltage[1] (Ports 1,2,3) | $I_{OL}$ = 1.6 mA | | 0.45 | V |
| | Output Low Voltage[1] (Port 0, ALE, $\overline{PSEN}$) | $I_{OL}$ = 3.2 mA | | 0.45 | V |
| | Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$) | $I_{OH}$ = -60 µA, $V_{CC}$ = 5V ± 10% | 2.4 | | V |
| | | $I_{OH}$ = -25 µA | 0.75 $V_{CC}$ | | V |
| | | $I_{OH}$ = -10 µA | 0.9 $V_{CC}$ | | V |
| | Output High Voltage (Port 0 in External Bus Mode) | $I_{OH}$ = -800 µA, $V_{CC}$ = 5V ± 10% | 2.4 | | V |
| | | $I_{OH}$ = -300 µA | 0.75 $V_{CC}$ | | V |
| | | $I_{OH}$ = -80 µA | 0.9 $V_{CC}$ | | V |
| | Logical 0 Input Current (Ports 1,2,3) | $V_{IN}$ = 0.45V | | -50 | µA |
| | Logical 1 to 0 Transition Current (Ports 1,2,3) | $V_{IN}$ = 2V, $V_{CC}$ = 5V ± 10% | | -650 | µA |
| | Input Leakage Current (Port 0, $\overline{EA}$) | 0.45 < $V_{IN}$ < $V_{CC}$ | | ±10 | µA |
| | Reset Pulldown Resistor | | 10 | 30 | KΩ |
| | Pin Capacitance | Test Freq. = 1 MHz, $T_A$ = 25°C | | 10 | pF |
| | Power Supply Current | Active Mode, 12 MHz | | 25 | mA |
| | | Idle Mode, 12 MHz | | 6.5 | mA |
| | Power-down Mode[1] | $V_{CC}$ = 5.5V | | 50 | µA |

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.

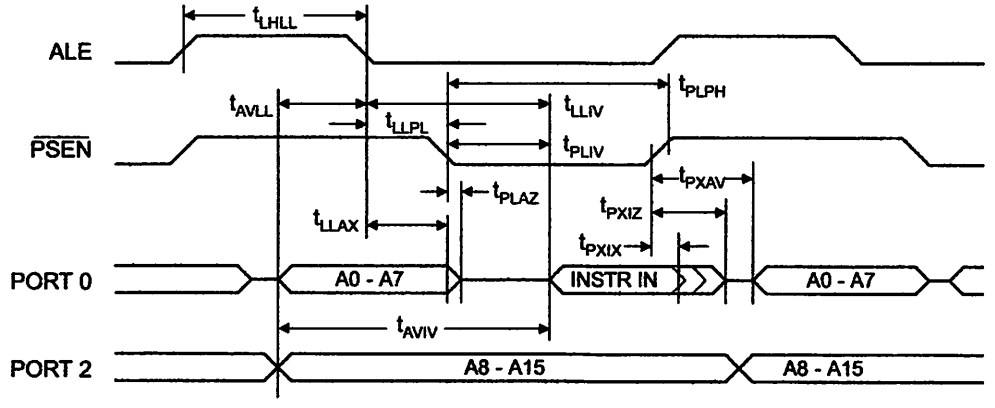
## Characteristics

operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other s = 80 pF.

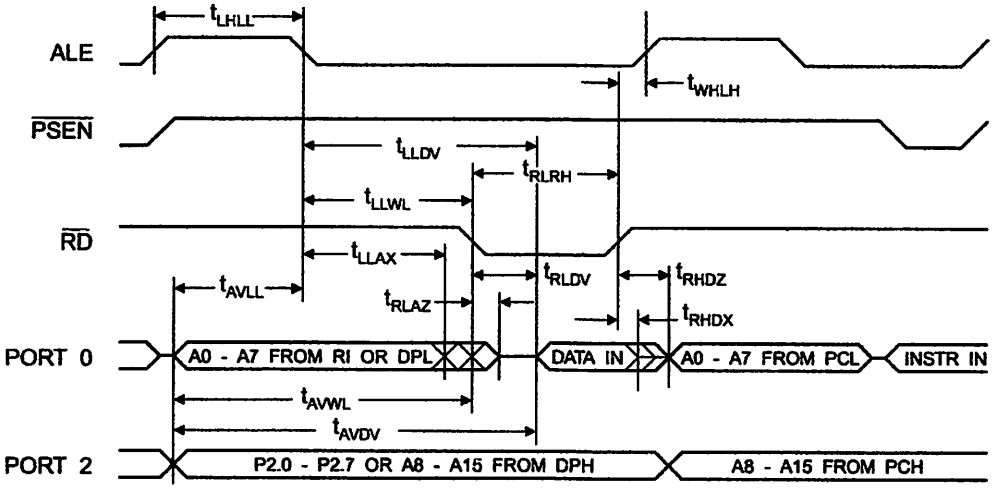
## rnal Program and Data Memory Characteristics

| ol | Parameter | 12 MHz Oscillator | | Variable Oscillator | | Units |
|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | |
| L | Oscillator Frequency | | | 0 | 33 | MHz |
| | ALE Pulse Width | 127 | | $2t_{CLCL}$-40 | | ns |
| | Address Valid to ALE Low | 43 | | $t_{CLCL}$-25 | | ns |
| | Address Hold After ALE Low | 48 | | $t_{CLCL}$-25 | | ns |
| | ALE Low to Valid Instruction In | | 233 | | $4t_{CLCL}$-65 | ns |
| | ALE Low to $\overline{PSEN}$ Low | 43 | | $t_{CLCL}$-25 | | ns |
| | $\overline{PSEN}$ Pulse Width | 205 | | $3t_{CLCL}$-45 | | ns |
| | $\overline{PSEN}$ Low to Valid Instruction In | | 145 | | $3t_{CLCL}$-60 | ns |
| | Input Instruction Hold After $\overline{PSEN}$ | 0 | | 0 | | ns |
| | Input Instruction Float After $\overline{PSEN}$ | | 59 | | $t_{CLCL}$-25 | ns |
| | $\overline{PSEN}$ to Address Valid | 75 | | $t_{CLCL}$-8 | | ns |
| | Address to Valid Instruction In | | 312 | | $5t_{CLCL}$-80 | ns |
| | $\overline{PSEN}$ Low to Address Float | | 10 | | 10 | ns |
| | $\overline{RD}$ Pulse Width | 400 | | $6t_{CLCL}$-100 | | ns |
| | $\overline{WR}$ Pulse Width | 400 | | $6t_{CLCL}$-100 | | ns |
| | $\overline{RD}$ Low to Valid Data In | | 252 | | $5t_{CLCL}$-90 | ns |
| | Data Hold After $\overline{RD}$ | 0 | | 0 | | ns |
| | Data Float After $\overline{RD}$ | | 97 | | $2t_{CLCL}$-28 | ns |
| | ALE Low to Valid Data In | | 517 | | $8t_{CLCL}$-150 | ns |
| | Address to Valid Data In | | 585 | | $9t_{CLCL}$-165 | ns |
| | ALE Low to $\overline{RD}$ or $\overline{WR}$ Low | 200 | 300 | $3t_{CLCL}$-50 | $3t_{CLCL}$+50 | ns |
| | Address to $\overline{RD}$ or $\overline{WR}$ Low | 203 | | $4t_{CLCL}$-75 | | ns |
| | Data Valid to $\overline{WR}$ Transition | 23 | | $t_{CLCL}$-30 | | ns |
| | Data Valid to $\overline{WR}$ High | 433 | | $7t_{CLCL}$-130 | | ns |
| | Data Hold After $\overline{WR}$ | 33 | | $t_{CLCL}$-25 | | ns |
| | $\overline{RD}$ Low to Address Float | | 0 | | 0 | ns |
| | $\overline{RD}$ or $\overline{WR}$ High to ALE High | 43 | 123 | $t_{CLCL}$-25 | $t_{CLCL}$+25 | ns |

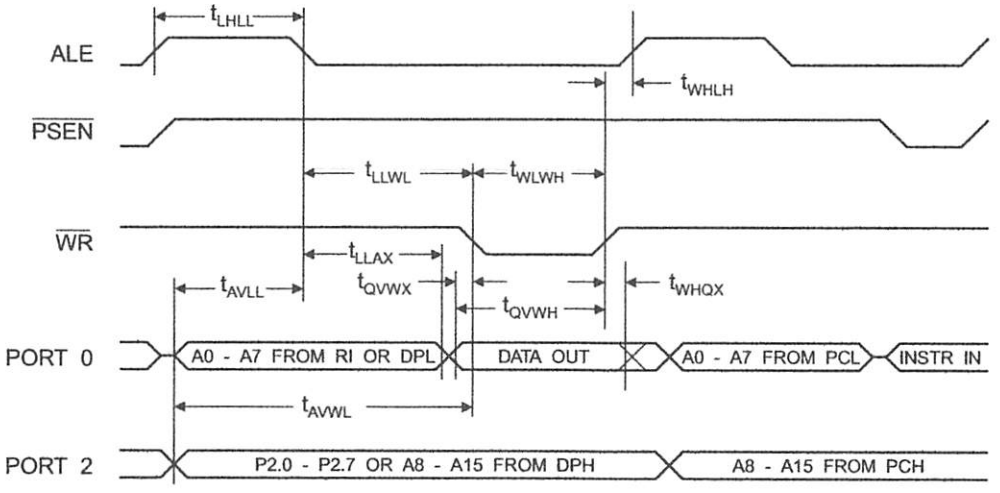
## rnal Program Memory Read Cycle



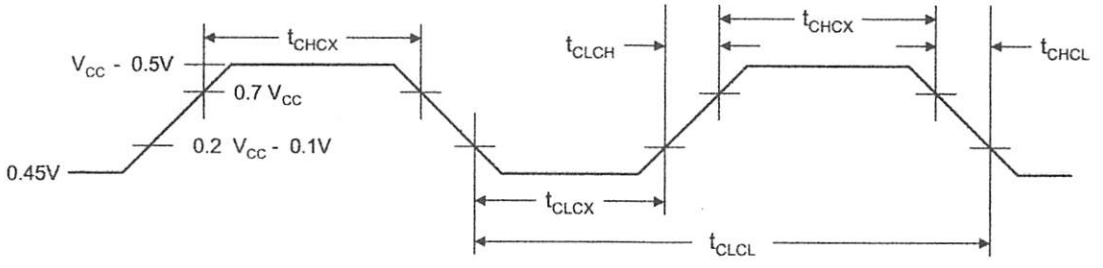

## rnal Data Memory Read Cycle



## rnal Data Memory Write Cycle




## rnal Clock Drive Waveforms




## rnal Clock Drive

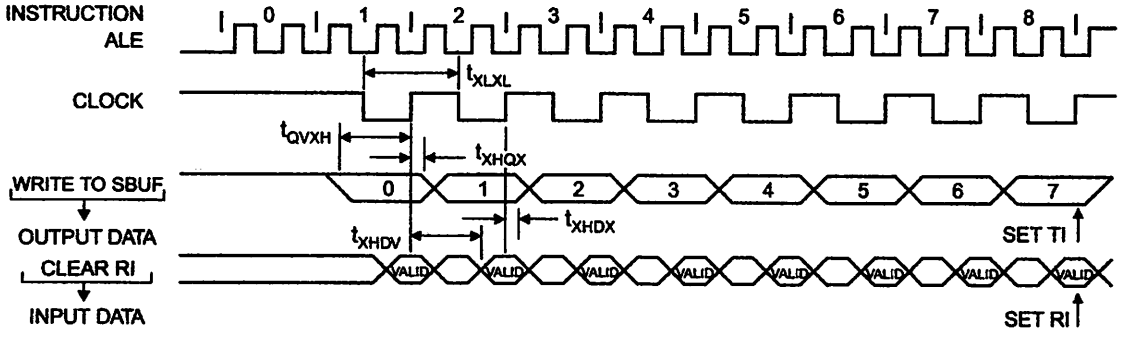
| ol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| L | Oscillator Frequency | 0 | 33 | MHz |
| | Clock Period | 30 | | ns |
| | High Time | 12 | | ns |
| | Low Time | 12 | | ns |
| | Rise Time | | 5 | ns |
| | Fall Time | | 5 | ns |

## al Port Timing: Shift Register Mode Test Conditions

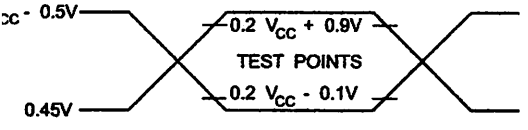
ilues in this table are valid for $V_{CC}$ = 4.0V to 5.5V and Load Capacitance = 80 pF.

| ol | Parameter | 12 MHz Osc | | Variable Oscillator | | Units |
|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | |
| | Serial Port Clock Cycle Time | 1.0 | | $12t_{CLCL}$ | | µs |
| | Output Data Setup to Clock Rising Edge | 700 | | $10t_{CLCL}$-133 | | ns |
| | Output Data Hold After Clock Rising Edge | 50 | | $2t_{CLCL}$-80 | | ns |
| | Input Data Hold After Clock Rising Edge | 0 | | 0 | | ns |
| | Clock Rising Edge to Input Data Valid | | 700 | | $10t_{CLCL}$-133 | ns |

## : Register Mode Timing Waveforms

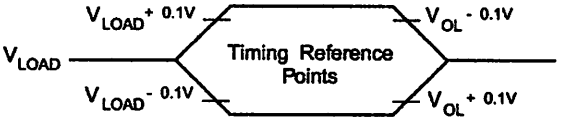



## esting Input/Output Waveforms[1]




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

## Float Waveforms[1]




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

## ering Information

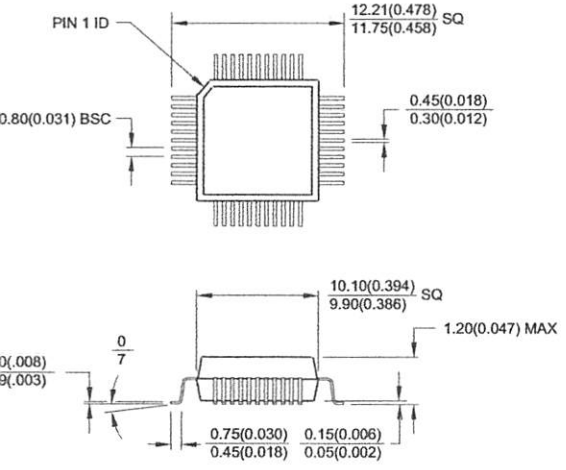
| eed Hz) | Power Supply | Ordering Code | Package | Operation Range |
|---|---|---|---|---|
| 4 | 4.0V to 5.5V | AT89S52-24AC<br>AT89S52-24JC<br>AT89S52-24PC | 44A<br>44J<br>40P6 | Commercial<br>(0° C to 70° C) |
| | | AT89S52-24AI<br>AT89S52-24JI<br>AT89S52-24PI | 44A<br>44J<br>40P6 | Industrial<br>(-40° C to 85° C) |
| 3 | 4.5V to 5.5V | AT89S52-33AC<br>AT89S52-33JC<br>AT89S52-33PC | 44A<br>44J<br>40P6 | Commercial<br>(0° C to 70° C) |

= Preliminary Availability

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

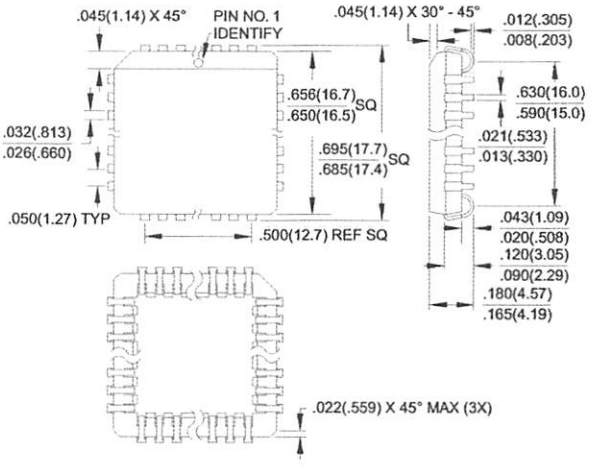
## kaging Information

**A,** 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad
at Package (TQFP)
mensions in Millimeters and (Inches)*

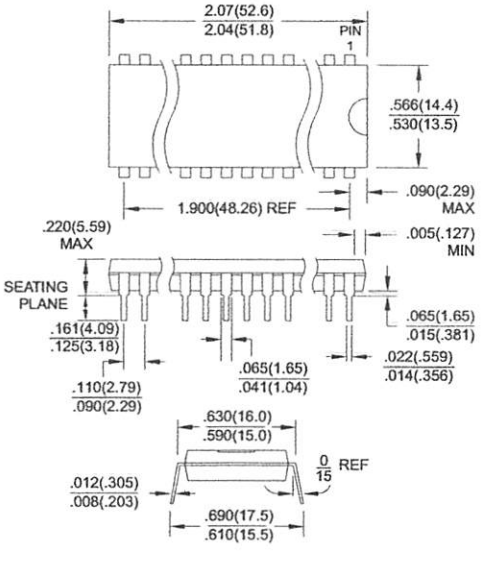



ontrolling dimension: millimeters

**44J,** 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)




**P6,** 40-pin, 0.600" Wide, Plastic Dual Inline
ckage (PDIP)
mensions in Inches and (Millimeters)
DEC STANDARD MS-011 AC



## el Headquarters

*orate Headquarters*
!5 Orchard Parkway
ı Jose, CA 95131
. (408) 441-0311
K (408) 487-2600

*pe*
ıel SarL
ıte des Arsenaux 41
ıa Postale 80
-1705 Fribourg
itzerland
. (41) 26-426-5555
K (41) 26-426-5500

ıel Asia, Ltd.
ɔm 1219
nachem Golden Plaza
Mody Road Tsimhatsui
ıt Kowloon
ıg Kong
. (852) 2721-9778
K (852) 2722-1369

ıel Japan K.K.
Tonetsu Shinkawa Bldg.
4-8 Shinkawa
ıo-ku, Tokyo 104-0033
an
. (81) 3-3523-3551
K (81) 3-3523-7581

## Atmel Product Operations

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

*Atmel Grenoble*
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-7658-3000
FAX (33) 4-7658-3480

*Atmel Heilbronn*
Theresienstrasse 2
POB 3535
D-74025 Heilbronn, Germany
TEL (49) 71 31 67 25 94
FAX (49) 71 31 67 24 23

*Atmel Nantes*
La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 0 2 40 18 18 18
FAX (33) 0 2 40 18 19 60

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

*Atmel Smart Card ICs*
Scottish Enterprise Technology Park
East Kilbride, Scotland G75 0QR
TEL (44) 1355-357-000
FAX (44) 1355-242-743


---

*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


**l Corporation 2001.**
orporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors ıay appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does e any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted ompany in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical ents in life support devices or systems.

' is the registered trademark of Atmel.

® is the registered trademark of Intel Corporation. Terms and product names in this document may be rks of others.

Printed on recycled paper.



Rev.1919A-07/01/xM

## tures

- Read Access Time - 150 ns
- Byte Write - 200 µs or 1 ms
- -Timed Byte Write Cycle
- Internal Address and Data Latches
- Internal Control Timer
- Automatic Clear Before Write
- ct Microprocessor Control
- DATA POLLING
- Power
- 30 mA Active Current
- 100 µA CMOS Standby Current
- ı Reliability
- *Endurance: $10^4$ or $10^5$ Cycles*
- Data Retention: 10 Years
- 10% Supply
- )S & TTL Compatible Inputs and Outputs
- EC Approved Byte Wide Pinout
- ımercial and Industrial Temperature Ranges

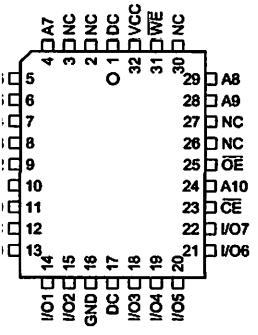
## cription

T28C16 is a low-power, high-performance Electrically Erasable and Program- Read Only Memory with easy to use features. The AT28C16 is a 16K memory zed as 2,048 words by 8 bits. The device is manufactured with Atmel's reliable latile CMOS technology. *(continued)*

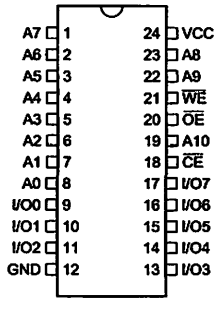
## Configurations

| lame | Function |
|---|---|
| A10 | Addresses |
| | Chip Enable |
| | Output Enable |
| | *Write Enable* |
| I/O7 | Data Inputs/Outputs |
| | No Connect |
| | Don't Connect |

PLCC
Top View




PDIP, SOIC
Top View




PLCC package pins 1 and 17
N'T CONNECT.

# 16K (2K x 8) Parallel EEPROMs

# AT28C16

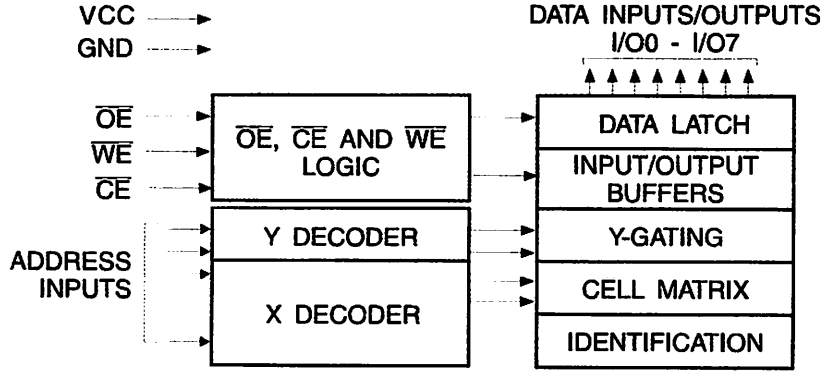
Rev. 0540B–10/98

,T28C16 is accessed like a static RAM for the read or cycles without the need of external components. Dur- byte write, the address and data are latched inter- freeing the microprocessor address and data bus for operations. Following the initiation of a write cycle, evice will go to a busy state and automatically clear rite the latched data using an internal control timer. nd of a write cycle can be determined by $\overline{DATA}$ ING of $I/O_7$. Once the end of a write cycle has been ted, a new access for a read or a write can begin.

The CMOS technology offers fast access times of 150 ns at low power dissipation. When the chip is deselected the standby current is less than 100 µA.

Atmel's 28C16 has additional features to ensure high quality and manufacturability. The device utilizes error correction internally for extended endurance and for improved data retention characteristics. An extra 32 bytes of EEPROM are available for device identification or tracking.

## ck Diagram




## olute Maximum Ratings*

| | |
|---|---|
| erature Under Bias | -55°C to +125°C |
| ge Temperature | -65°C to +150°C |
| put Voltages (including NC Pins) Respect to Ground | -0.6V to +6.25V |
| utput Voltages Respect to Ground | -0.6V to $V_{CC}$ + 0.6V |
| ge on $\overline{OE}$ and A9 Respect to Ground | -0.6V to +13.5V |

*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

## ice Operation

**):** The AT28C16 is accessed like a Static RAM. $\overline{CE}$ and $\overline{OE}$ are low and $\overline{WE}$ is high, the data stored memory location determined by the address pins is ted on the outputs. The outputs are put in a high lance state whenever $\overline{CE}$ or $\overline{OE}$ is high. This dual line ol gives designers increased flexibility in preventing ontention.

**WRITE:** Writing data into the AT28C16 is similar to g into a Static RAM. A low pulse on the $\overline{WE}$ or $\overline{CE}$ with $\overline{OE}$ high and $\overline{CE}$ or $\overline{WE}$ low (respectively) ini- a byte write. The address location is latched on the lling edge of $\overline{WE}$ (or $\overline{CE}$); the new data is latched on st rising edge. Internally, the device performs a self- before write. Once a byte write has been started, it utomatically time itself to completion. Once a pro- ning operation has been initiated and for the duration , a read operation will effectively be a polling opera-

**BYTE WRITE:** The AT28C16E offers a byte write of 200 µs maximum. This feature allows the entire to be rewritten in 0.4 seconds.

**POLLING:** The AT28C16 provides $\overline{DATA}$ POLLING nal the completion of a write cycle. During a write cycle, an attempted read of the data being written results in the complement of that data for $I/O_7$ (the other outputs are indeterminate). When the write cycle is finished, true data appears on all outputs.

**WRITE PROTECTION:** Inadvertent writes to the device are protected against in the following ways: (a) $V_{CC}$ sense—if $V_{CC}$ is below 3.8V (typical) the write function is inhibited; (b) $V_{CC}$ power on delay—once $V_{CC}$ has reached 3.8V the device will automatically time out 5 ms (typical) before allowing a byte write; and (c) write inhibit—holding any one of $\overline{OE}$ low, $\overline{CE}$ high or $\overline{WE}$ high inhibits byte write cycles.

**CHIP CLEAR:** The contents of the entire memory of the AT28C16 may be set to the high state by the CHIP CLEAR operation. By setting $\overline{CE}$ low and $\overline{OE}$ to 12 volts, the chip is cleared when a 10 msec low pulse is applied to $\overline{WE}$.

**DEVICE IDENTIFICATION:** An extra 32 bytes of EEPROM memory are available to the user for device identification. By raising A9 to 12 ± 0.5V and using address locations 7E0H to 7FFH the additional bytes may be written to or read from in the same manner as the regular memory array.

## and AC Operating Range

| | | AT28C16-15 |
|---|---|---|
| ating<br>erature (Case) | Com. | 0°C - 70°C |
| | Ind. | -40°C - 85°C |
| Power Supply | | 5V ± 10% |

## rating Modes

| e | $\overline{CE}$ | $\overline{OE}$ | $\overline{WE}$ | I/O |
|---|---|---|---|---|
| l | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | $D_{OUT}$ |
| (2) | $V_{IL}$ | $V_{IH}$ | $V_{IL}$ | $D_{IN}$ |
| dby/Write Inhibit | $V_{IH}$ | X(1) | X | High Z |
| Inhibit | X | X | $V_{IH}$ | |
| Inhibit | X | $V_{IL}$ | X | |
| ut Disable | X | $V_{IH}$ | X | High Z |
| Erase | $V_{IL}$ | $V_H$(3) | $V_{IL}$ | High Z |

1. X can be $V_{IL}$ or $V_{IH}$.
2. Refer to AC Programming Waveforms.
3. $V_H$ = 12.0V ± 0.5V

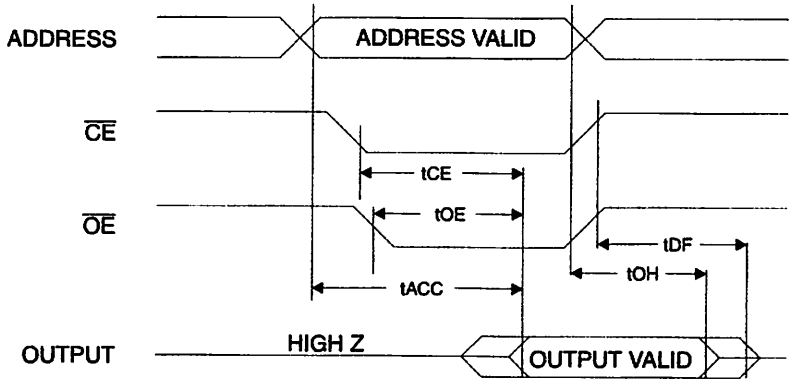
## Characteristics

| bol | Parameter | Condition | | Min | Max | Units |
|---|---|---|---|---|---|---|
| | Input Load Current | $V_{IN}$ = 0V to $V_{CC}$ + 1V | | | 10 | µA |
| | Output Leakage Current | $V_{I/O}$ = 0V to $V_{CC}$ | | | 10 | µA |
| | $V_{CC}$ Standby Current CMOS | $\overline{CE}$ = $V_{CC}$ - 0.3V to $V_{CC}$ + 1.0V | | | 100 | µA |
| | $V_{CC}$ Standby Current TTL | $\overline{CE}$ = 2.0V to $V_{CC}$ + 1.0V | Com. | | 2 | mA |
| | | | Ind. | | 3 | mA |
| | $V_{CC}$ Active Current AC | f = 5 MHz; $I_{OUT}$ = 0 mA<br>$\overline{CE}$ = $V_{IL}$ | Com. | | 30 | mA |
| | | | Ind. | | 45 | mA |
| | Input Low Voltage | | | | 0.8 | V |
| | Input High Voltage | | | 2.0 | | V |
| | Output Low Voltage | $I_{OL}$ = 2.1 mA | | | .4 | V |
| | Output High Voltage | $I_{OH}$ = -400 µA | | 2.4 | | V |

## Read Characteristics

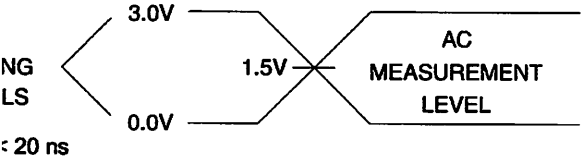
| bol | Parameter | AT28C16-15 Min | AT28C16-15 Max | Units |
|---|---|---|---|---|
| | Address to Output Delay | | 150 | ns |
| | $\overline{CE}$ to Output Delay | | 150 | ns |
| | $\overline{OE}$ to Output Delay | 10 | 70 | ns |
| 4) | $\overline{CE}$ or $\overline{OE}$ High to Output Float | 0 | 50 | ns |
| | Output Hold from $\overline{OE}$, $\overline{CE}$ or Address, whichever occurred first | 0 | | ns |

## Read Waveforms(1)(2)(3)(4)



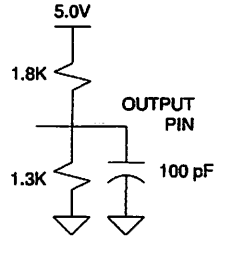

1. $\overline{CE}$ may be delayed up to $t_{ACC} - t_{CE}$ after the address transition without impact on $t_{ACC}$.
2. $\overline{OE}$ may be delayed up to $t_{CE} - t_{OE}$ after the falling edge of $\overline{CE}$ without impact on $t_{CE}$ or by $t_{ACC} - t_{OE}$ after an address change without impact on $t_{ACC}$.
3. $t_{DF}$ is specified from $\overline{OE}$ or $\overline{CE}$ whichever occurs first ($C_L$ = 5 pF).
4. This parameter is characterized and is not 100% tested.

## It Test Waveforms and surement Level




NG
LS
: 20 ns

## Output Test Load




## Capacitance

IHz, T = 25°C(1)

| ol | Typ | Max | Units | Conditions |
|---|---|---|---|---|
| | 4 | 6 | pF | $V_{IN}$ = 0V |
| | 8 | 12 | pF | $V_{OUT}$ = 0V |

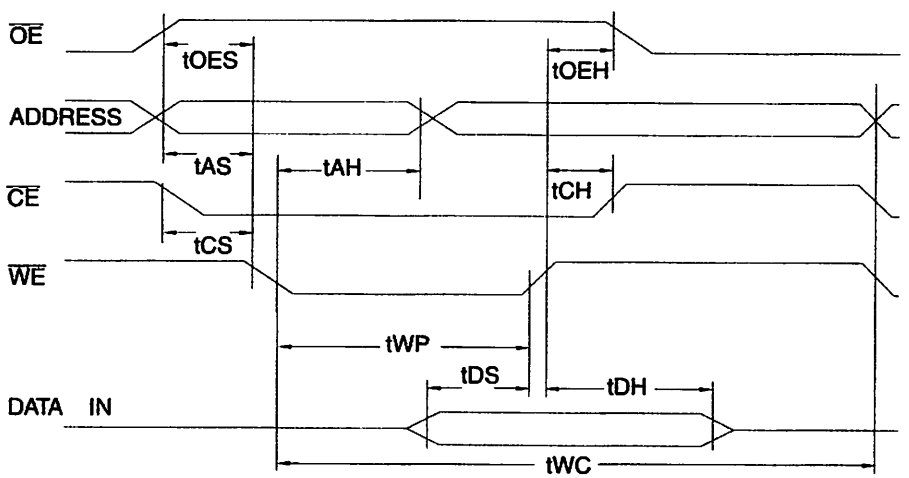
1. This parameter is characterized and is not 100% tested.

## Write Characteristics

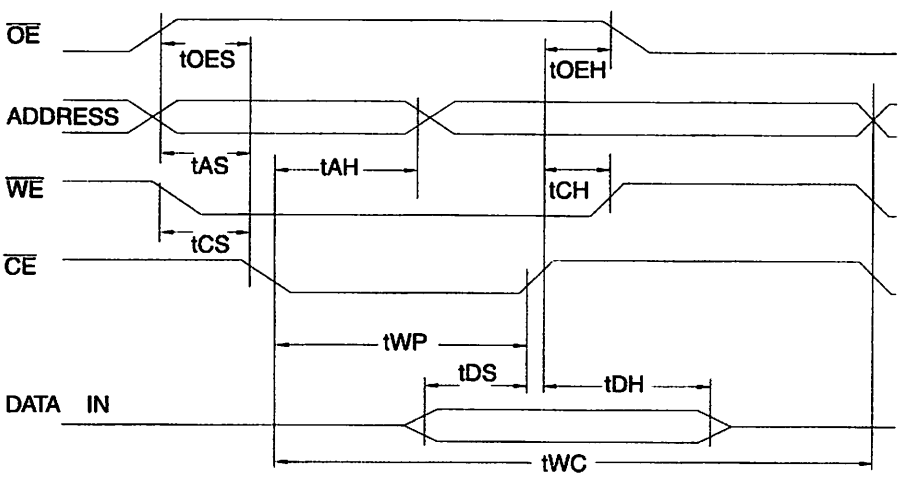
| bol | Parameter | | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| DES | Address, $\overline{OE}$ Set-up Time | | 10 | | | ns |
| | Address Hold Time | | 50 | | | ns |
| | Write Pulse Width ($\overline{WE}$ or $\overline{CE}$) | | 100 | | 1000 | ns |
| | Data Set-up Time | | 50 | | | ns |
| DEH | Data, $\overline{OE}$ Hold Time | | 10 | | | ns |
| CH | $\overline{CE}$ to $\overline{WE}$ and $\overline{WE}$ to $\overline{CE}$ Set-up and Hold Time | | 0 | | | ns |
| | Write Cycle Time | AT28C16 | | 0.5 | 1.0 | ms |
| | | AT28C16E | | 100 | 200 | µs |

## Write Waveforms

### Controlled




### ontrolled

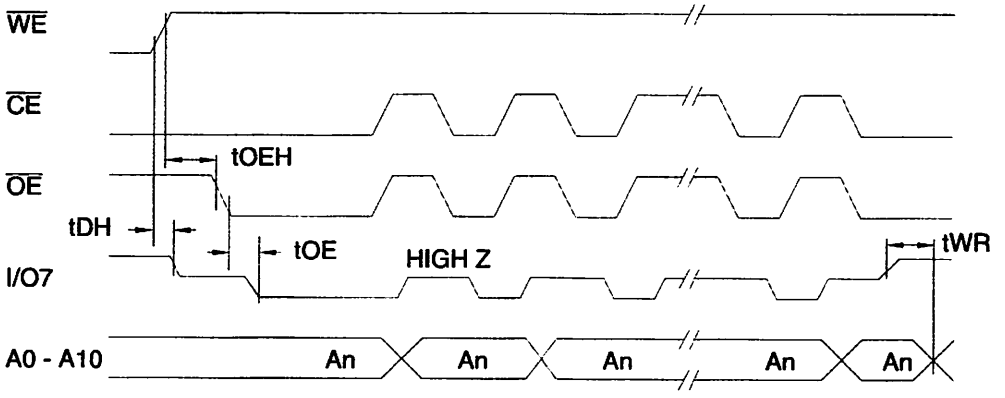


## ā Polling Characteristics[1]

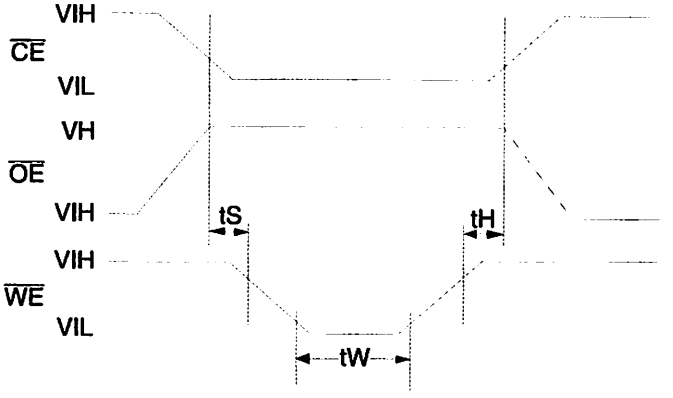
| bol | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| | Data Hold Time | 10 | | | ns |
| | $\overline{OE}$ Hold Time | 10 | | | ns |
| | $\overline{OE}$ to Output Delay[2] | | | | ns |
| | Write Recovery Time | 0 | | | ns |

1. These parameters are characterized and not 100% tested.
2. See AC Characteristics.

## ā Polling Waveforms

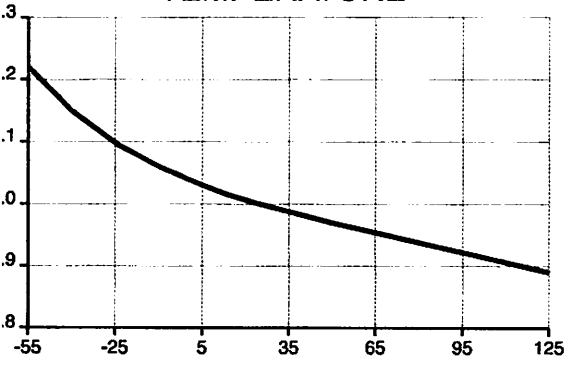



## › Erase Waveforms

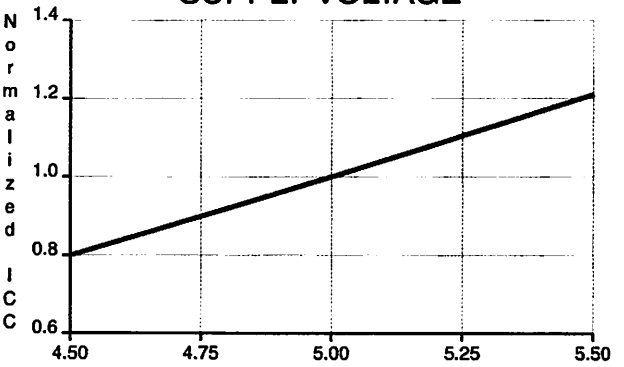



= 1 µsec (min.)
msec (min.)
2.0V ± 0.5V

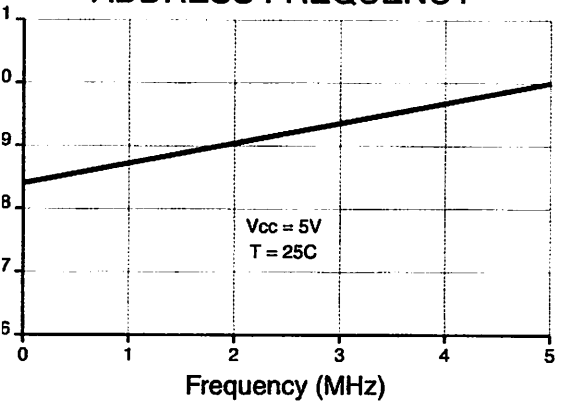
## NORMALIZED SUPPLY CURRENT vs. TEMPERATURE



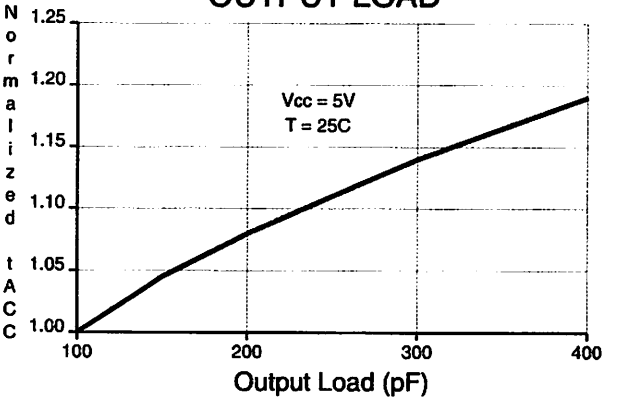

## NORMALIZED SUPPLY CURRENT vs. SUPPLY VOLTAGE



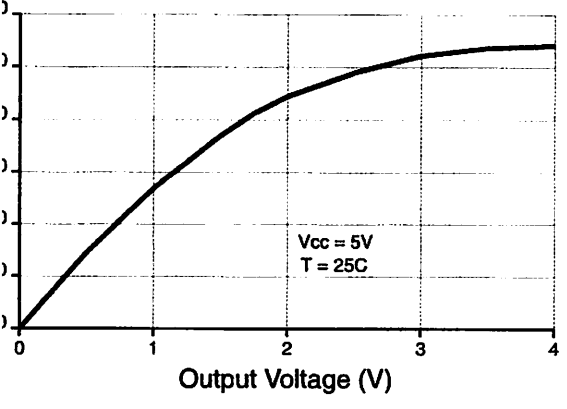

## NORMALIZED SUPPLY CURRENT vs. ADDRESS FREQUENCY



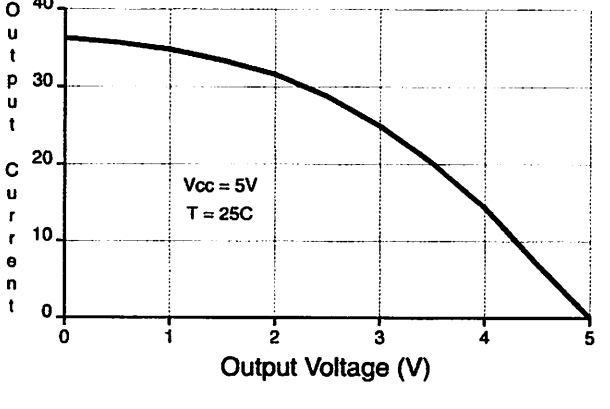

## NORMALIZED ACCESS TIME vs. OUTPUT LOAD




## OUTPUT SINK CURRENT vs. OUTPUT VOLTAGE




## OUTPUT SOURCE CURRENT vs. OUTPUT VOLTAGE



## ering Information[1]

| cc s) | $I_{CC}$ (mA) Active | $I_{CC}$ (mA) Standby | Ordering Code | Package | Operation Range |
|---|---|---|---|---|---|
| 50 | 30 | 0.1 | AT28C16(E)-15JC<br>AT28C16(E)-15PC<br>AT28C16(E)-15SC | 32J<br>24P6<br>24S | Commercial<br>(0°C to 70°C) |
| | 45 | 0.1 | AT28C16(E)-15JI<br>AT28C16(E)-15PI<br>AT28C16(E)-15SI | 32J<br>24P6<br>24S | Industrial<br>(-40°C to 85°C) |

1. See Valid Part Numbers table below.
2. The 28C16 200 ns and 250 ns speed selections have been removed from valid selections table and are replaced by the faster 150 ns $T_{AA}$ offering.
3. The 28C16 ceramic package offerings have been removed. New designs should utilize the 28C256 ceramic offerings.

## d Part Numbers

llowing table lists standard Atmel products that can be ordered.

| ce Numbers | Speed | Package and Temperature Combinations |
|---|---|---|
| C16 | 15 | JC, JI, PC, PI, SC, SI |
| C16E | 15 | JC, JI, PC, PI, SC, SI |
| C16 | - | W |

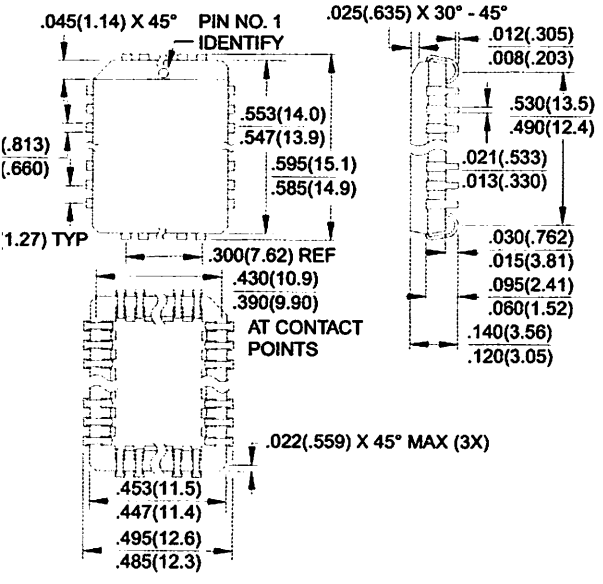
## Products

ence Section: Parallel EEPROM Die Products

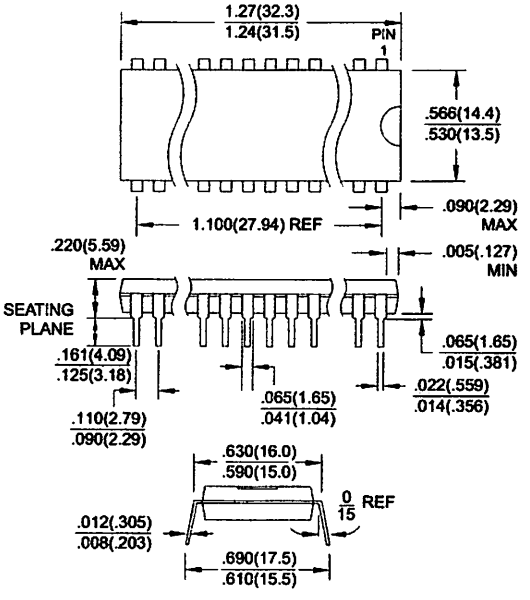
| | Package Type |
|---|---|
| | 32 Lead, Plastic J-Leaded Chip Carrier (PLCC) |
| | 24 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP) |
| | 24 Lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC) |
| | Die |
| | **Options** |
| t | Standard Device: Endurance = 10K Write Cycles; Write Time = 1 ms |
| | High Endurance Option: Endurance = 100K Write Cycles; Write Time = 200 µs |

# kaging Information

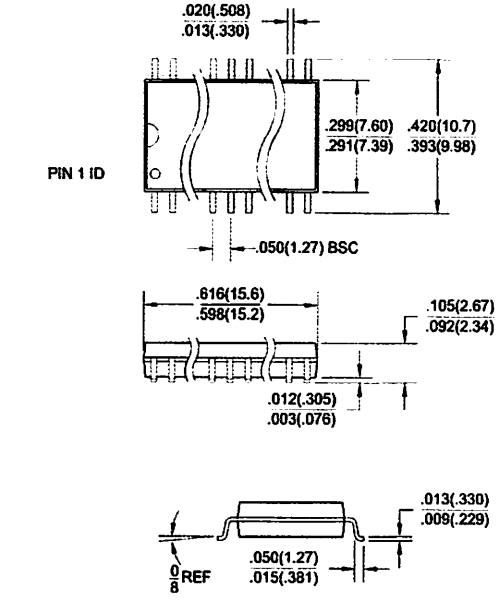
**2J**, 32-Lead, Plastic J-Leaded Chip Carrier (PLCC)
imensions in Inches and (Millimeters)
:DEC STANDARD MS-018 AA




**24P6**, 24-Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-011 AA




**IS**, 24-Lead, 0.300" Wide, Plastic Gull Wing Small
utline (SOIC)
mensions in Inches and (Millimeters)



## ıel Headquarters

*ıorate Headquarters*
25 Orchard Parkway
n Jose, CA 95131
L (408) 441-0311
X (408) 487-2600

*ıpe*
nel U.K., Ltd.
liseum Business Centre
ɾerside Way
mberley, Surrey GU15 3YL
gland
L (44) 1276-686677
X (44) 1276-686697

nel Asia, Ltd.
om 1219
inachem Golden Plaza
Mody Road
imshatsui East
wloon, Hong Kong
L (852) 27219778
X (852) 27221369

*n*
nel Japan K.K.
netsu Shinkawa Bldg., 9F
4-8 Shinkawa
uo-ku, Tokyo 104-0033
ɔan
L (81) 3-3523-3551
X (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 42 53 60 00
FAX (33) 4 42 53 60 01

*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



**ıl Corporation 1998.**
Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard war-
hich is detailed in Atmel's Terms and Conditions located on the Company's website. The Company assumes no responsibility for
ors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without
and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual prop-
Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are
horized for use as critical components in life support devices or systems.

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

ınd product names in this document may be trademarks of others.

Printed on recycled paper.



0540B–10/98/xM