

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



**PERANCANGAN DAN PEMBUATAN ALAT UKUR SUHU
DAN KADAR KARBON MONOKSIDA (CO) DI JALAN RAYA
YANG DILENGKAPI TAMPILAN WAKTU DAN INFORMASI**

SKRIPSI

**Disusun Oleh :
Ali Machfudin
NIM 0217014**

MARET 2007

OMAH JANGAN KONGSI TUTTIKI

SEKITAR KEMAYOGAN WARTU

18 APRIL 1962

KEMAYOGAN

WALAU SUDAH TAJA KATAKUNYAN MAS HADIMADHAN
AVAN KALAU (OO) ABERKONON HESKAN KADAN MAS
KONKONON MAS UTAWA MAJUNAT PANGONZINDI DMAY

2000000

: 1000 000000

000000000000

4 700 000 000

0000000000

LEMBAR PERSETUJUAN

PERANCANGAN DAN PEMBUATAN ALAT UKUR SUHU DAN KADAR
KARBON MONOKSIDA (CO) DI JALAN RAYA YANG DILENGKAPI
TAMPILAN WAKTU DAN INFORMASI

SKRIPSI

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

Disusun Oleh :

ALI MACHFUDIN
NIM : 0217014

Diperiksa dan Disetujui

Dosen Pembimbing I


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

Dosen Pembimbing II


M. Ashar, ST, MT

Mengetahui



Jurusan Teknik Elektro S-1


Yudi Limpraptono, MT
NIP. Y. 1039500274

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

2007



BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama Mahasiswa : Ali Machfudin
NIM : 02.17.014
Jurusan : Teknik Elektro S1
Konsentrasi : Teknik Elektronika
Judul Skripsi : Perancangan dan Pembuatan Alat Ukur Suhu dan Kadar Karbon Monoksida (CO) Di Jalan Raya Yang Dilengkapi Tampilan Waktu dan Informasi
Dipertahankan dihadapan Tim Penguji Skripsi Jenjang Strata Satu (S-1) pada :
Hari : Jum'at
Tanggal : 16 Maret 2007
Dengan Nilai : A (81,4) *By*

PANITIA UJIAN SKRIPSI



KEPUA

Ir. Mochtar Asroni, MSME
NIP.Y. 1018100036

SEKRETARIS

Ir. F. Yudi Limpraptono, M.T.
NIP.Y. 1039500274

ANGGOTA PENGUJI

PENGUJI I

Ir. Eko Nurcahyo
NIP.Y.1028700172

PENGUJI II

I Komang Somawirata, ST, MT
NIP.P 1030100361

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas semua limpahan rahmat dan hidayah-Nya yang diberikan kepada penulis, sehingga penulis bisa menyelesaikan skripsi dengan judul *”Perancangan dan Pembuatan Alat Ukur Suhu dan Kadar Karbon Monoksida (CO) Di Jalan Raya Yang Dilengkapi Tampilan Waktu dan Informasi ”* dengan baik dan tepat pada waktunya.

Banyak pihak yang telah membantu dalam penulisan skripsi ini, oleh karena itu penulis ingin mengucapkan rasa terima kasih yang sebesar – besarnya kepada :

1. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE. selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Mochtar Asroni, MSME selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
3. Bapak Ir. F.Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S1 Konsentrasi Teknik Elektronika.
4. Bapak Ir. F.Yudi Limpraptono, MT dan Bapak M.Ashar, ST, MT selaku Dosen Pembimbing yang telah memberikan bimbingan, petunjuk, dan saran sehingga skripsi ini bisa selesai dengan baik dan tepat pada waktunya.
5. Bapak Ir. Yusuf Ismail Nahkoda, MT. selaku Sekretaris Jurusan Teknik Elektro S1.
6. Seluruh dosen di Institut Teknologi Nasional Malang yang telah memberikan bekal ilmu kepada penulis selama penulis mengikuti kuliah.

Akhir kata, penulis berharap semoga skripsi ini bermanfaat, untuk pengembangan penelitian, khususnya di Jurusan Teknik Elektro Institut Teknologi Nasional Malang dan bisa bermanfaat bagi seluruh masyarakat pada umumnya.

Malang, Maret 2007

Penulis

**PERANCANGAN DAN PEMBUATAN ALAT UKUR SUHU DAN KADAR
KARBON MONOKSIDA (CO) DIJALAN RAYA YANG DILENGKAPI
TAMPILAN WAKTU DAN INFORMASI**

Ali Machfudin, Konsentrasi Teknik Elektronika, Jurusan Teknik Elektro S1,
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang, 2007.
Dosen Pembimbing I : Ir. F.Yudi Limpraptono, MT
Dosen Pembimbing II : M. Ashar, ST, MT

Abstrak

Pada skripsi dirancang dan dibuat suatu alat yang mampu mengukur suhu dan kadar karbon monoksida (CO) di jalan raya yang dilengkapi tampilan waktu dan informasi.

Alat ini menggunakan dua sensor sebagai pendeteksi suhu dan kadar karbon monoksida, output dari sensor yang berupa data analog diubah ke digital oleh ADC (analog to digital converter) agar dapat diolah datanya oleh mikrokontroler. Data yang telah diproses tersebut kemudian dikirim ke mikrokontroler berikutnya secara serial untuk ditampilkan pada display dot matrik. Pada alat ini juga dilengkapi dengan tampilan waktu (jam dan tanggal) dengan menggunakan RTC (real time clock) dan juga pesan atau informasi.

Dari perancangan dan pembuatan alat tersebut setelah dilakukan kalibrasi dan pengukuran dengan membandingkan dengan alat yang sudah ada diperoleh selisih (error) 1,928 % untuk pembacaan suhu dan 6,61 % untuk pembacaan CO.

Kata Kunci : *Alat Ukur CO, Sensor, ADC, Mikrokontroler, RTC*

DAFTAR ISI

HALAMAN JUDUL

LEMBAR PERSETUJUAN

BERITA ACARA UJIAN SKRIPSI

| | |
|-----------------------------|-------------|
| KATA PENGANTAR | i |
| ABSTRAK | iii |
| DAFTAR ISI | iv |
| DAFTAR GAMBAR | viii |
| DAFTAR TABEL | x |
| DAFTAR GRAFIK | xi |

BAB I PENDAHULUAN

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

BAB II LANDASAN TEORI

| | |
|---------------------------------|---|
| 2.1 Pendahuluan | 5 |
| 2.2 Karbon Monoksida (CO) | 5 |
| 2.3 Sensor Yang Digunakan | 7 |

| | |
|---|----|
| 2.3.1 Sensor Karbon Monoksida (CO) AF56 | 7 |
| 2.3.2 Sensor Suhu (LM 35) | 10 |
| 2.4 Mikrokontroler | 11 |
| 2.4.1 Mikrokontroler AT89C89S51 | 11 |
| 2.4.1.1 Struktur Memori AT89C51 | 14 |
| 2.4.1.1.1 RAM Internal | 15 |
| 2.4.1.1.2 Register Fungsi Khusus (<i>Special Function Register</i>) | 16 |
| 2.4.1.2 <i>Timer</i> dan <i>Counter</i> Dalam MCS51 | 19 |
| 2.4.1.3 Komunikasi Data Serial | 20 |
| 2.4.2 Mikrokontroler AT89S8252..... | 21 |
| 2.4.2.1 Perangkat keras Mikrokontroller AT89S8252 | 21 |
| 2.4.2.2 SFR Tambahan Pada Atmel AT89S8252..... | 23 |
| 2.4.2.3 Data Memori (EEPROM) dan RAM..... | 23 |
| 2.4.2.4 Programable Watchdog Timer (WDT) | 24 |
| 2.4.2.5 Timer 2 | 24 |
| 2.5 <i>Real Time Clock</i> (RTC) IC DS12887 | 26 |
| 2.5.1 Fungsi Pin-pin RTC DS12C887..... | 27 |
| 2.5.2 Register-register RTC DS12C887..... | 29 |
| 2.6 <i>Analog Digital To Digital</i> (ADC) 0808 | 33 |
| 2.7 Komponen Pendukung Tampilan Dot Matrik | 34 |
| 2.7.1 Dot Matrik | 34 |
| 2.7.2 Dekoder Matrik LED | 35 |
| 2.7.3 <i>Register Geser 8 Bit Serial In Parallel Out</i> (SIPO) | 36 |
| 2.8 Penguat (Op-Amp) | 37 |

| | |
|--------------------------------|----|
| 2.8.1 Inverting Amplifier..... | 38 |
| 2.9 Keypad | 39 |

**BAB III PERANCANAAN DAN PEMBUATAN ALAT UKUR SUHU DAN
KADAR KARBON MONOKSIDA**

| | |
|---|----|
| 3.1 Pendahuluan | 41 |
| 3.2 Diagram Blok | 41 |
| 3.3 Prinsip Kerja..... | 42 |
| 3.4 Perancangan Perangkat Keras | 43 |
| 3.4.1 Sensor Suhu | 43 |
| 3.4.2 Sensor CO | 45 |
| 3.4.3 ADC0808 | 46 |
| 3.4.4 Mikrokontroler | 50 |
| 3.4.5 RTC (<i>Real Time Clock</i>)..... | 52 |
| 3.4.6 Dot Matrik | 53 |
| 3.5 Perancangan Perangkat Lunak (<i>Software</i>) | 55 |
| 3.5.1 Flowchat Program Utama..... | 56 |
| 3.5.2 Flowchat Untuk Menampilkan Suhu dan Kadar CO..... | 57 |
| 3.5.3 Flowchat Mengambil Data dan Menampilkan Waktu | 58 |
| 3.5.4 Flowchat Untuk <i>Setting</i> Tanggal, Bulan dan Tahun | 59 |
| 3.5.5 Flowchat Untuk <i>Setting</i> Jam | 60 |
| 3.5.6 Flowchart Untuk Mengambil dan Mengirim Data ADC dan RTC | 61 |

BAB IV PENGUJIAN DAN PERCOBAAN ALAT

| | |
|--|----|
| 4.1 Pendahuluan | 62 |
| 4.2 Pengujian Sensor Suhu | 62 |
| 4.2.1 Pengujian Untuk Mencari Besarnya Penguatan Vout Sensor | 62 |
| 4.2.2 Pengujian Untuk Mencari Kesalahan Rata-rata Dari Vout Sensor | 63 |
| 4.2.3 Pengujian Untuk Mencari Kesalahan Rata-rata Vout Op-Amp | 64 |
| 4.2.4 Pengujian Pembacaan Suhu Pada Tampilan (<i>Display</i>) | 65 |
| 4.3 Pengujian Sensor CO | 66 |
| 4.4 Pengujian ADC | 67 |
| 4.5 Pengujian Dot Matrik | 68 |
| 4.6 Pengujian RTC | 68 |
| 4.7 Pengujian Keseluruhan..... | 69 |

BAB V PENUTUP

| | |
|----------------------|----|
| 5.1 Kesimpulan | 74 |
| 5.2 Saran | 76 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1 Sensor Gas AF-56 | 8 |
| Gambar 2.2 Tampak Bawah Sensor LM35 | 10 |
| Gambar 2.3 Susunan Pin Dari AT89S51 | 11 |
| Gambar 2.4 Alamat RAM Internal dan <i>Flash</i> PEROM | 14 |
| Gambar 2.5 Peta Memori dan <i>Special Function Register</i> | 16 |
| Gambar 2.6 Konsep Dasar Timer / Counter Sebagai Input..... | 20 |
| Gambar 2.7 Susunan Pin Dari AT89S8252 | 22 |
| Gambar 2.8 Konfigurasi Kaki RTC DS12887 | 27 |
| Gambar 2.9 Konfigurasi Kaki ADC0808 | 34 |
| Gambar 2.10 <i>Display</i> Dot Matrik 5x7 | 35 |
| Gambar 2.11 Diagram Detail <i>Register</i> Geser IC 74HC164 | 36 |
| Gambar 2.12 Konfigurasi Kaki 74HC164 | 37 |
| Gambar 2.13 Penguat <i>Non Inverting</i> | 38 |
| Gambar 2.14 Keypad 4x3..... | 40 |
| Gambar 3.1 Diagram Blok Sistem Alat | 41 |
| Gambar 3.2 Rangkaian Sensor Suhu dan Pengkondisi Sinya | 44 |
| Gambar 3.3 Rangkaian Sensor Gas AF-56 | 45 |
| Gambar 3.4 Rangkaian ADC0808 | 47 |
| Gambar 3.5 Rangkaian <i>Clock</i> Eksternal ADC0808 | 48 |
| Gambar 3.6 Rangkaian Mikrokontroler AT89S51 | 51 |
| Gambar 3.7 Rangkaian Mikrokontroler AT89S8252 | 52 |
| Gambar 3.8 Rangkaian RTC | 53 |

Gambar 3.9 Rangkaian *Driver* Dot Matrik 54

Gambar 4.1 Pengukuran Arus Basis (I_B) dan Arus Kolektor (I_C) 68

DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1 Pengaruh Konsentrasi COHb Dalam Darah Terhadap Manusia | 7 |
| Tabel 2.2 Fungsi – Fungsi Khusus Port3 MCS-51 | 12 |
| Tabel 2.3 Fungsi Khusus Pada Port 1 AT89S8252 | 22 |
| Tabel 2.4 Mode Operasi Timer 2 | 25 |
| Tabel 3.1 <i>Address Decoder</i> ADC0808..... | 47 |
| Tabel 3.2 Hasil Perhitungan Output ADC0808 | 50 |
| Tabel 4.1 Pengukuran Vout Sensor dan Vout Op-Amp..... | 63 |
| Tabel 4.2 Perbandingan Vout Sensor Data Sheet Dengan Pengukuran | 64 |
| Tabel 4.3 Perbandingan Vout Op-Amp Perhitungan Dengan Pengukuran | 65 |
| Tabel 4.4 Pembacaan Suhu Termometer Dengan <i>Display</i> Dot Matrik | 66 |
| Tabel 4.5 Pengukuran Vout Sensor CO | 66 |
| Tabel 4.6 Pembacaan Alat ukur CO <i>Display</i> Dot Matrik | 67 |
| Tabel 4.7 Perbandingan Output Perhitungan Dengan Pengukuran ADC0808 | 67 |
| Tabel 4.8 Perbandingan Waktu Jam Digital Dengan <i>Display</i> Dot Matrik..... | 69 |

DAFTAR GRAFIK

| | |
|--|----|
| Grafik 4.1 Perbandingan Temperatur Terhadap Vout Sensor Suhu..... | 70 |
| Grafik 4.2 Perbandingan Kadar CO Terhadap Vout SensorCO..... | 71 |
| Grafik 4.3 Perbandingan Pembacaan Suhu Termometer Dengan Alat Yang Dibuat | 72 |
| Grafik 4.4 Perbandingan Pembacaan Alat Ukur CO Dengan Alat Yang Dibuat.. | 73 |

BAB I

PENDAHULUAN

1. Latar Belakang

Dalam kehidupan sehari-hari peranan informasi dirasa sangat penting baik dalam kondisi apapun dan dimanapun, tidak terkecuali di jalan raya. Masalah yang sangat nyata di jalan raya kita selain semakin banyaknya kecelakaan juga masalah polusi udara dari hari kehari dirasa semakin bertambah parah. Jalan raya yang merupakan tempat dari sumber polusi udara yang paling parah, berdasarkan data yang diperoleh dari beberapa sumber yang dapat dipercaya bahwa tingkat polusi udara di kota-kota besar di Indonesia diperkirakan 70 % berasal dari jalan raya yang disebabkan oleh asap kendaraan dan sisanya berasal dari industri atau pabrik^[1].

Pertambahan jumlah kendaraan yang cukup tinggi seiring dengan tingkat pertumbuhan ekonomi nasional mengakibatkan beberapa kota besar, seperti : Jakarta, Medan, Bandung, Semarang dan Surabaya menerima beban polusi udara yang sangat tinggi, yang tentu saja berpengaruh cukup besar bagi tingkat kesehatan masyarakat di kota besar. Salah satunya adalah polusi udara dari gas karbon monoksida (CO). Disamping itu akibat yang ditimbulkan dari gas tersebut suhu disekitar kita juga ikut berubah.

Atas dasar itulah diperlukan suatu alat yang dapat mengukur besarnya suhu dan kadar karbon monoksida (CO) di udara pada jalan raya secara otomatis. Pembuatan alat tersebut dapat membantu sekali dalam memberikan informasi pada masyarakat tentang tingkat polusi yang terjadi di jalan raya kita.

2. Rumusan Masalah

Dalam penulisan skripsi ini, masalah-masalah yang dibahas adalah sebagai berikut :

1. Bagaimana cara mendeteksi suhu dan kadar CO udara kemudian mengolah data sehingga dapat dilihat hasilnya secara nyata.
2. Merancang dan membuat hardware dan software yang dibutuhkan dalam proses tersebut.
3. Bagaimana cara memberikan informasi pada masyarakat tentang keadaan udara disuatu jalan raya seperti suhu, kadar karbon monoksida (CO) dan menampilkan waktu serta pesan / informasi.

3. Tujuan

Tujuan penulis memilih judul ini adalah merancang dan membuat alat ukur suhu dan karbon monoksida (CO) di jalan raya yang dilengkapi dengan tampilan waktu dan informasi.

4. Batasan Masalah

Didalam penulisan skripsi ini, agar pembahasan masalah sesuai dengan yang diinginkan dan tidak menyimpang dari apa yang semula dirumuskan, maka penulis menyertakan batasan masalah sebagai berikut :

1. Alat ini menggunakan dua buah mikrokontroller dari jenis AT89S51 dan AT89S8252.
2. Sensor yang digunakan adalah LM35 untuk sensor suhu dan AF-56 untuk sensor CO.

3. Untuk sensor gas AF-56 hanya digunakan untuk mendeteksi karbon monoksida (CO).
4. Pada alat ini display yang digunakan yaitu dot matrik 5 x 7 titik sebanyak 10 karakter.
5. Alat yang dibuat bukan ukuran sebenarnya hanyalah miniatur atau model.

5. Metodologi

Metode yang digunakan dalam perencanaan dan pembuatan laporan tugas akhir ini adalah :

1. Studi Literatur

Dengan mempelajari teori dasar serta aplikasinya yang mempunyai hubungan dengan alat yang dibuat.

2. Percobaan Laboratorium

Melakukan percobaan dan membandingkan data hasil percobaan dengan kondisi sebenarnya di lapangan yang merupakan data primer sebagai pembuktian kebenaran dari data sekunder hasil studi literature.

3. Merancang dan membuat perangkat keras dan perangkat lunaknya.

- 1) Membuat gambaran umum
- 2) Analisa kebutuhan, baik kebutuhan perangkat keras maupun perangkat lunak.
- 3) Perancangan alat
 - a) Pembuatan diagarm

- b) Penentuan data flow diagram
 - c) Membuat blok diagram
 - d) Membuat rancangan fisik alat yang dibuat
 - e) Membuat rancangan software
- 4) Membuat alat sesuai dengan rancangan yang dibuat.
- 4. Ujicoba terhadap alat yang telah di buat.**
- 5. Menyusun laporan skripsi.**

6. Sistematika

Adapun sistematika dari penyusunan laporan skripsi ini adalah :

BAB I PENDAHULUAN

Berisi latar belakang belakang, rumusan masalah, tujuan, pembatasan masalah, metodologi penulisan, serta sistematika penyusunan dan pembuatan alat.

BAB II TEORI PENUNJANG

Berisi tentang teori – teori dasar yang memiliki relevansi sebagai dasar perencanaan dan pembuatan.

BAB III PERENCANAAN DAN PEMBUATAN

Berisi tentang perencanaan hardware dan software.

BAB IV PENGUJIAN ALAT

Berisi tentang data hasil pengujian peralatan yang telah dibuat secara keseluruhan.

BAB V PENUTUP

Berisi kesimpulan dan saran dari tugas akhir ini

BAB II

LANDASAN TEORI

2.1. Pendahuluan

Untuk dapat memahami alat yang akan dirancang ini maka dalam bab II ini dijelaskan mengenai teori dasar yang berkaitan dengan sistem ini, diantaranya meliputi: Mikrokontroler AT89C51, teori tentang karbon monoksida dan sensor-sensor yang digunakan, *Real Time Clock* (RTC), *Analog to Digital Converter* (ADC), display dot matrik dan komponen pendukungnya dan komponen-komponen yang lainnya.

2.2. Karbon Monoksida (CO)

Karbon monoksida (CO) adalah suatu komponen tidak berwarna, tidak berbau dan tidak mempunyai rasa yang terdapat dalam gas pada suhu diatas 192 °C. komponen yang mempunyai berat sebesar 96,5 % dari berat air dan tidak larut dalam air. Karbon monoksida (CO) yang terdapat di alam ini terbentuk dari salah satu proses sebagai berikut :

1. Pembakaran tidak lengkap terhadap karbon (C) atau komponen mengandung karbon.
2. Reaksi antara karbon monoksida (CO) dan komponen yang mengandung karbon pada suhu tinggi.
3. Pada suhu tinggi, karbon dioksida (CO₂) terurai menjadi karbon monoksida (CO) dan oksida (O).

Oksidasi tidak lengkap pada karbon atau komponen yang mengandung karbon terjadi jika jumlah oksigen yang tersedia kurang dari jumlah yang dibutuhkan untuk pembakaran sempurna dimana dihasilkan karbon dioksida (CO_2). Pembentukan karbon monoksida (CO) hanya terjadi jika reaktan yang ada terdiri atas karbon (C) dan oksigen (O_2) murni. Jika yang terjadi adalah pembakaran komponen yang mengandung karbon di udara, prosesnya lebih kompleks dan terdiri dari beberapa tahap reaksi. Reaksi ini sering terjadi pada suhu tinggi yang umumnya terdapat pada industri.

Pada umumnya konsentrasi karbon monoksida (CO) di udara kurang dari 110 ppm. Jika terjadi kontak antara manusia dengan karbon monoksida (CO) pada konsentrasi tinggi dapat mengakibatkan kematian, sedangkan pada konsentrasi yang relatif rendah (110 ppm atau kurang) dapat mengganggu kesehatan. Faktor yang penting yang menentukan pengaruh karbon monoksida (CO) terhadap manusia adalah konsentrasi COHb yang terdapat dalam darah, dimana semakin tinggi prosentase hemoglobin (Hb) yang terikat dalam bentuk CoHb maka semakin buruk pengaruhnya terhadap kesehatan manusia^[1]. Hubungan antara konsentrasi CoHb didalam darah dan pengaruhnya terhadap kesehatan dapat dilihat pada table dibawah ini:

Tabel 2.1 Pengaruh Konsentrasi COHb Di Dalam Darah Terhadap Manusia^[2]

| Jumlah COHb Dalam Darah (%) | Gejala-gejala (Akibat) |
|------------------------------------|--|
| 0 - 10 | Tidak ada gejala (normal) |
| 10 – 20 | Sedikit sakit kepala, pengembangan pembuluh-pembuluh darah kulit |
| 20 - 30 | Sakit kepala menghentak-hentak dipelipis, mudah terangsang, emosi labil, cepat letih. |
| 30 – 40 | Sakit kepala hebat, badan lemas, pandangan kabur, mual, muntah dan jatuh |
| 40 - 50 | Sakit kepala hebat, badan lemas, pandangan kabur, mual, muntah, jatuh dan pingsan, denyut nadi dan pernapasan bertambah. |
| 50 - 60 | Sinkope, pernapasan dan denyut nadi bertambah, koma dengan sebentar kejang-kejang. |
| 60 - 70 | Koma dengan kejang-kejang, kerja jantung dan pernapasan tertekan dan bisa mati. |
| 70 - 80 | Denyut jantung sangat lemah, pernapasan lambat, kegagalan pernapasan dan menyebabkan kematian. |

2.3. Sensor yang Digunakan

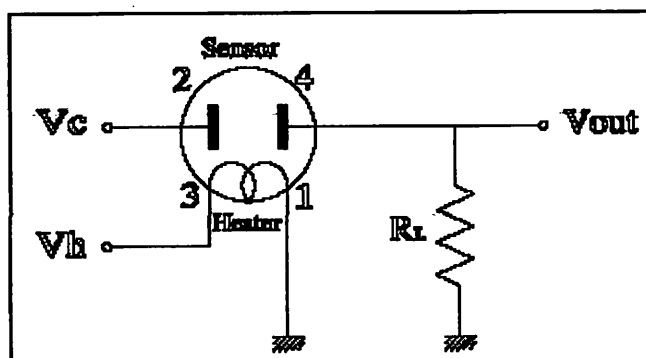
2.3.1. Sensor Karbon Monoksida (CO) AF56

Sesor gas karbon monoksida (CO) yang digunakan adalah sensor gas AF-seri (Scemarec) yang terbuat dari bahan semikonduktor yang memiliki kinerja, kualitas dan ketahanan jangka panjang. Elemen semiconductor yang mendeteksi gas dipanaskan pada salah satu sisi bahan keramik, selanjutnya ditutup dengan keramik film untuk melindungi terhadap debu. Sebuah *micro-heater* dipasang

pada bagian belakang lapisan, dengan demikian cepat memanaskan elemen pendeteksi gas pada suhu operasi yang diinginkan.

Elemen yang merasakan gas adalah suatu semikonduktor oksida yang dipanaskan sampai 350 °C. Jika berhadapan dengan gas yang sesuai, oksigen di permukaan akan diserap untuk digunakan dan tahanan elektriknya menurun. Penurunan tahanan ini mengikuti pola yang dapat diprediksi dan berulang-ulang. Dimana keluaran sensor tersebut adalah berupa selisih tegangan, yaitu saat sensor dipengaruhi oleh gas karbon monoksida (CO) tegangan tersebut akan naik sesuai dengan banyaknya gas karbon monoksida (CO) yang diterima oleh sensor.

Sensor karbon monoksida yang digunakan adalah sensor gas tipe AF56 (scemarec), yang merupakan sensor gas yang dapat bekerja pada temperatur -10°C sampai 55 °C, serta tegangan kerja pada rangkaian berupa DC atau AC sebesar 5 V (tegangan maksimal 12 V) dan tegangan kerja untuk pemanas (heater) berupa tegangan DC atau AC sebesar 5 V dengan bias $\pm 0,2$ V.



Gambar 2.1 Sensor Gas AF-56 ^[3]

Struktur gas sensor karbon monoksida (CO) ini memiliki 4 kaki dimana kaki 3 dan kaki 1 berfungsi sebagai *heater* sedangkan kaki 2 dan kaki 4 berfungsi sebagai sensor karbon monoksida (CO). Sensor gas ini dirancang untuk beroperasi dengan tegangan 5 V. Ketika sensor mendeteksi gas, tahanan sensor akan menurun.

Sensitivitas menunjukkan rasio (R_{GAS}/R_{UDARA}) dari sensor gas yang diperoleh dalam udara yang berisi gas (R_{GAS}) untuk tahanan sensor yang diperoleh di udara bersih tanpa pencemaran gas (R_{UDARA}). Sedangkan daya sensor (P_S) yang dibutuhkan pada saat bekerja adalah $P_S \leq 15 \text{ mW}$.

Dari gambar diatas dapat diketahui bahwa tahanan sensor (R_S) dapat dihitung dari tegangan output (V_o) dengan menggunakan rumus sebagai berikut :

$$R_S = \frac{V_c - V_{out}}{V_{out}} \cdot R_L$$

sedangkan daya sensor (P_S) yang dibutuhkan pada saat bekerja :

$$P_S = \frac{V_c^2 - R_S}{(R_S + R_L)}$$

Dimana :

R_S = Tahanan sensor

R_L = Tahanan beban (variabel)

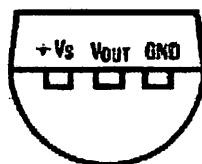
V_c = Tegangan kerja pada rangkaian

V_{out} = Tegangan keluaran (V_o)

P_S = Daya yang dibutuhkan oleh sensor ($P_S \leq 15 \text{ mW}$)^[3]

2.3.2. Sensor Suhu (LM 35)

Seri LM35 adalah sensor temperatur atau suhu yang sangat presisi, itu dapat di lihat dari tegangan output yang mempunyai sifat kelinieran yang sesuai yaitu dapat dinyatakan langsung dalam derajat celcius. LM35 mempunyai keuntungan yaitu lebih linier bila dikalibrasi ke kelvin, untuk memperoleh skala celcius tidak perlu mengurangi tegangan konstan dari output. Untuk pengkalibrasian tidak memerlukan kalibrasi eksternal atau penyesuaian terhadap tipikal akurasi yaitu pada $\pm 0,25$ °C untuk temperatur ruang dan $0,75$ °C untuk temperatur yang lebih tinggi, range temperatur antara -55 sampai $+150$ °C. Seri LM35 mempunyai impedansi output yang rendah, output yang linier dan juga ketepatan kalibrasi yang merupakan kelebihan dari LM35 yang membuat penginterfaccan untuk pembacaan keluar atau kontrol rangkaian sangat mudah. LM35 juga dapat digunakan dengan menggunakan power supply tunggal yaitu maksudnya supply tegangan plus (+) atau minus (-). Selain itu hanya membawa 60 uA dari supplynya, itu membuat *self-heating*-nya sangat rendah, lebih kecil dari $0,1$ °C pada udara konstan.



Gambar 2.2 Tampak Bawah Dari Sensor LM35^[4]

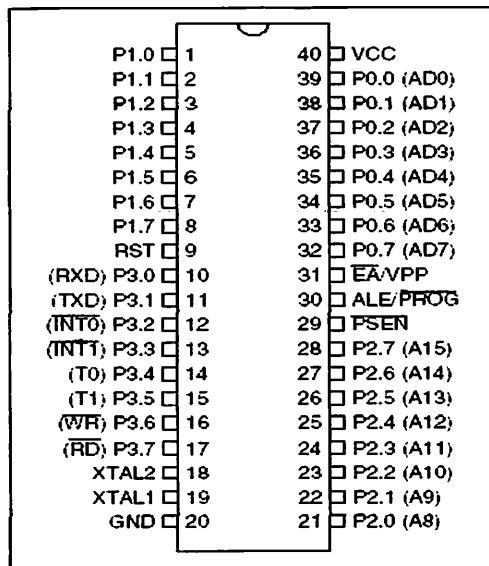
2.4. Mikrokontroler

2.4.1. Mikrokontroler AT 89C51

Mikrokontroler AT89C51 merupakan salah satu rangkaian terintegrasi (*Integrated Circuits*) mikrokontroler yang tergabung dalam keluarga MCS-51.

Beberapa fitur yang terdapat pada mikrokontroler ini ialah sebagai berikut :

1. 4 buah I/O (*input/output*) port 8 bit.
2. 4 Kbyte memori program (*Flash EEPROM*).
3. 128 byte RAM internal.
4. 2 buah *timer* 16 bit.
5. 64 kbyte maksimum eksternal *memory address*.
6. 210 bit *addressable memory location*.
7. Terdapat *interface* untuk komunikasi serial.



Gambar 2.3 Susunan Pin Dari AT89C51^[5]

Keterangan Pin mikrokontroler AT89C51

1. Port 0

Dalam perancangan konvensional Port ini hanya digunakan sebagai port I/O serbaguna, namun dalam sistem yang melibatkan memori eksternal port ini dapat digunakan secara bergantian sebagai port data maupun alamat (A0-A7).

2. Port 1 : Port I/O serba guna

3. Port 2 : Port dua fungsi.

Fungsinya sama dengan port 0, namun pada penggunaan memori eksternal, port ini digunakan untuk *bus* alamat *high* (A8-A15).

4. Port 3 : Port dua fungsi.

Selain dapat digunakan sebagai port I/O serbaguna dapat pula digunakan untuk fungsi-fungsi khusus lainnya seperti yang ditunjukkan pada tabel 2.1 berikut:

Tabel 2.2. Fungsi Khusus Port 3 MCS-51^[6]

| Bit | Nama pin | Alamat bit | Fungsi alternatif |
|------|----------|------------|----------------------------------|
| P3.0 | RXD | B0H | Comm serial receiver |
| P3.1 | TXD | B1H | Comm serial tranceiver |
| P3.2 | INT 0 | B2H | Eksternal interrupt 0 |
| P3.4 | T 0 | B4H | Eksternal Timer /Counter Input 0 |
| P3.5 | T 1 | B5H | Eksternal Timer /Counter Input 1 |
| P3.6 | WR | B6H | Eksternal memori write enable |

5. PSEN (*Program store enable*)

Merupakan sebuah sinyal *output* yang berfungsi sebagai kontrol dalam membaca program maupun kode dari memori eksternal. Biasanya pin ini dihubungkan dengan *output enable* dari EPROM. Jika program yang dieksekusi

berasal dari memori internal maka Pin ini akan berada pada kondisi tidak aktif (*high*).

6. ALE (*Address Latch Enable*)

Sinyal *output* ALE berfungsi dalam memisahkan waktu pengiriman data dengan alamat dalam penggunaan memori eksternal. ALE hanya akan aktif pada saat mengakses memori eksternal.

7. EA (Eksternal akses)

Jika Pin EA diberi logika rendah maka mikrokontroler akan mengakses memori program eksternal. Sebaliknya jika Pin EA diberi logika tinggi, mikrokontroler akan mengakses memori program internal. Pin ini juga dipakai sebagai tegangan pemrograman EPROM atau *Flash Memori Internal*.

8. RST (Reset)

Pin ini berfungsi untuk mereset logika program dari awal. Biasanya untuk penggunaan *Power on reset*, sebuah resistor *pullup* ke VCC dan sebuah kapasitor yang terhubung ke *ground* dihubungkan pada pin ini.

9. *Oscillator*

Oscillator pada mikrokontroler ini diatur oleh besarnya nilai X'tal yang terhubung pada pin 18 dan pin 19. Besar nilai X'tal sekitar 12 -24 Mhz.

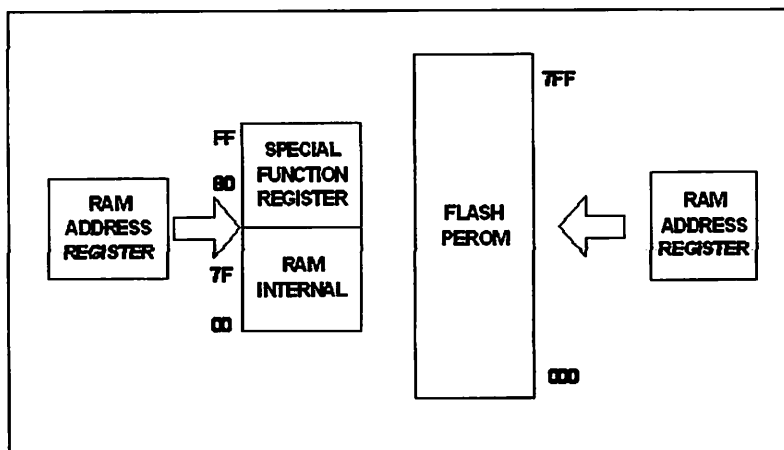
10. *Power*

AT89C51 memerlukan tegangan catu sebesar 5 V DC pada pin 40 dan 0 V (*ground*) sebagai titik acuan pada pin 20.

2.4.1.1. Struktur Memori AT89C51

Memori yang ada pada AT89C51 terdiri atas : (1) RAM Internal, memori sebesar 128 byte biasa difunakan untuk menyimpan variable atau data yang bersifat sementara. (2) *Special Function Register*(Register Fungsi Khusus), memori ini berisi register-register yang mempunyai fungsi-fungsi khusus yang disediakan oleh mikrokontroler tersebut seperti timer, serial dan lain-lain. (3) *Flash PEROM*, memori yang digunakan untuk menyimpan instruksi-instruksi dalam AT89C51.

Dalam bukunya Paulus Andi Nalwan (2003) menyatakan: “AT89C51 mempunyai struktur memori yang terpisah antara RAM internal dan *Flash PEROM*-nya”. RAM Internal dialamati oleh *RAM Address Register* (Register Alamat RAM) sedangkan *Flash PEROM* dialamati oleh *Program Address Register* (Register Alamat Program). Dengan adanya struktur memori yang terpisah tersebut, maka walaupun RAM Internal dan *Flash PEROM* memiliki alamat yang sama yaitu alamat 00, namun secara fisiknya kedua memori tersebut tidak saling berhubungan. Untuk lebih jelasnya dapat dilihat pada gambar 2.4.



Gambar 2.4. Alamat RAM Internal dan *Flash PEROM* ^[7]

2.4.1.1.1. RAM Internal

RAM Internal pada AT89C51 terdiri atas :

1. Register Banks

AT89C51 mempunyai delapan buah register , terdiri dari R0-R7. Kedelapan buah register ini selalu terletak pada alamat 00H-07H saat sistem direset.

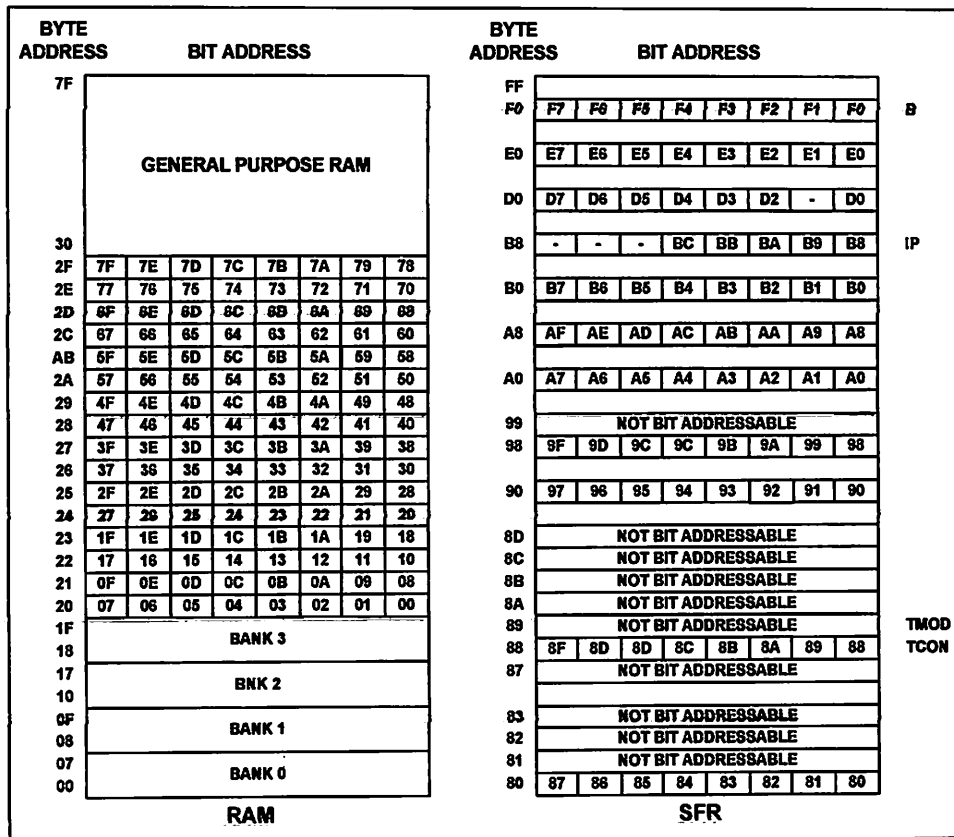
2. Bit Addressable RAM

RAM pada alamat 20H-2FH dapat diakses secara pengalamatan bit (*bit addressable*) sehingga hanya dengan sebuah instruksi saja setiap bit dalam area ini dapat diset, clear, AND dan OR.

3. *General purpose RAM* (RAM Keperluan Umum)

RAM ini dimulai pada alamat 30H-7FH yang dapat diakses dengan pengalamatan langsung maupun pengalamatan tak langsung. Pengalamatan langsung dilakukan ketika salah satu operand merupakan bilangan yang menunjukkan lokasi yang dialamati. Sedangkan pengalamatan secara tak langsung pada lokasi RAM Internal ini yaitu akses data dari memori ketika alamat memori tersebut tersimpan dalam suatu register R0 atau R1. Register R0 dan R1 merupakan dua buah register pada mikrokontroler AT89C51 yang dapat digunakan sebagai pointer dari sebuah lokasi memori pada RAM Internal.

Untuk gambar peta memori RAM dan *Special Funton Register* dapat dilihat pada gambar 2.5.



Gambar 2.5 Peta Memori dan *Special Function Register*^[8]

2.4.1.1.2. Register Fungsi Khusus (*Special Function Register*)

AT89C51 memiliki 21 *Special Function Register* (Register Fungsi Khusus). Terletak pada alamat 80H-FFH (lihat Gambar 2.5). Beberapa dari register-register ini dapat dialamatkan dengan pengalamatan bit.

Berikut ini adalah register-register yang ada pada *Special Function Register*.

1. Akumulator

Register ini terletak pada alamat E0H, fungsinya yaitu untuk operasi aritmatik dan operasi logika serta untuk pengiriman data ke memori eksternal.

2. Port I/O

AT89C51 memiliki empat buah port, yaitu port 0, port 1, port 2 dan port 3 terletak pada alamat 80H, 90H, A0H dan B0H. Namun jika digunakan kita menggunakan eksternal memori ataupun fungsi-fungsi spesial seperti *external interrupt*, serial ataupun *external timer*, port 0, port 2 dan port 3 tidak dapat digunakan sebagai port dengan fungsi umum. Untuk itu disediakan port 1 yang dikhususkan untuk port dengan fungsi umum.

3. PSW (*Program Status Word*)

Program Status Word atau PSW terletak pada alamat D0H (gambar 2.5).

4. Register B

Dipakai bersama-sama dengan akumulator untuk proses aritmatik, register ini berfungsi juga register biasa, sifatnya *bit addressable*.

4. Stack Pointer

Merupakan sebuah register 8 bit, terletak pada alamat 81H, berisi alamat dari data yang disimpan di-*stack*.

Stack pointer dapat diedit atau dibiarkan saja mengikuti standar sesudah terjadi reset (Nalwan ; 14).

Instruksi-instruksi yang biasa dipakai adalah *push*, *POP*, *Acsll* dan *Lcall*.

6. Data Pointer

Data pointer (DPTR) merupakan register 16 bit dan terletak pada alamat 82H untuk DPL da 83H untuk DPH. Berfungsi untuk mengakses *source code* ataupun data yang terletak dimemori eksternal. Sebagai contoh lihat *listing* program dibawah ini :

```
Mov  A,#01H
Mov  DPTR,#2000H
```


Mov @DPTR,A

Listing diatas berfungsi untuk menuliskan data 01H kealamat 2000H, langkahnya pertama data diisikan ke akumulator kemudian DPTR yang berfungsi untuk menunjukkan alamat penyimpanan data diisi dengan 2000H. kemudian isi data dari akumulator akan disimpan ke lokasi memori yang ditunjuk oleh DPTR.

7. Register Timer

AT89C51 terdiri dari dua buah 16 bit Timer/counter, yaitu Timer 0 dan Timer 1. Timer terletak di alamat 8AH untuk TL0 dan 8CH untuk TH0 dan Timer 1 terletak di alamat 8BH untuk TL1 dan 8DH untuk TH1.

8. Register Port Serial

Dalam AT89C51 terdapat sebuah *on chip serial port* (Port Serial di dalam keeping), berfungsi untuk berkomunikasi dengan peralatan lain yang menggunakan serial port juga seperti modem, *shift register* dan lain-lain.

Buffer (penyangga) dalam proses komunikasi terletak pada register SBUF di alamat 99H, sedangkan untuk mengatur mode serial dapat dilakukan dengan mengubah isi dari SCON yang terletak pada alamat 98H.

9. Register Interupsi

AT89C51 memiliki lima buah interupsi dengan dua level prioritas interupsi. Interupsi ini selalu nonaktif setiap sistem dalam AT89C51 di reset. Register-register yang berhubungan dengan *interrupti* adalah *interrupt enable register* (IE) atau register pengaktif interupsi terletak pada alamat A8H, berfungsi untuk mengatur aktifnya tiap-tiap *interrupt* dan yang terakhir adalah *interrupt priority register* (IP) atau register prioritas interupsi terletak pada alamat B8H.

10. Register Kontrol Power

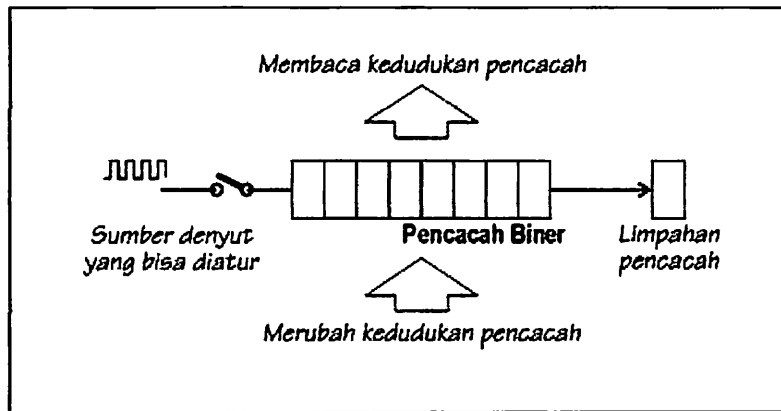
Register ini terdiri atas SMOD, berfungsi untuk melipat dua *baud rate* dari port serial, dua buah bit untuk *flag* fungsi umum pada bit ketiga dan bit kedua, *power down* (PD) bit dan *Idle* (IDL) bit. pada *mode idle* hubungan antara CPU (*central processing unit*) dan *internal clock* terputus, tapi port tetap pada kondisi terakhir, ALE dan PSEN menjadi *high*, timer masih tetap bekerja. Mode *idle* berakhir pada saat terjadi interupsi, reset ataupun kondisi-kondisi lain yang mereset IDL bit.

2.4.1.2. *Timer dan Counter* dalam MCS51

Dalam mikrokontroler AT89C51 terdapat dua buah timer yaitu Timer 0 dan Timer 1, keduanya berfungsi sebagai *counter* maupun *timer*. Pada dasarnya sarana *input* yang satu ini merupakan seperangkat pencacah biner (*binary counter*) yang terhubung langsung ke saluran-data mikrokontroler, sehingga mikrokontroler bisa membaca kedudukan pancakah, bila diperlukan mikrokontroler dapat pula merubah kedudukan pencacah tersebut.

Seperti layaknya pencacah biner, bilamana sinyal *clock* (denyut) yang diumpankan sudah melebihi kapasitas pencacah, maka pada bagian akhir rangkaian pencacah akan timbul sinyal limpahan, sinyal ini merupakan suatu hal yang penting sekali dalam pemakaian pencacah. Terjadinya limpahan pencacah ini dicatat dalam sebuah *flip-flop* tersendiri.

Di samping itu, sinyal *clock* yang diumpankan ke pencacah harus bisa dikendalikan dengan mudah. Hal-hal di atas diringkas dalam Gambar 2.5



Gambar 2.6 Konsep Dasar Timer/Counter Sebagai Sarana Input^[9]

Sinyal *clock* yang diumpankan ke pencacah bisa dibedakan menjadi 2 macam, yang pertama sinyal *clock* dengan frekuensi tetap yang sudah diketahui besarnya dan yang kedua adalah sinyal *clock* dengan frekuensi tidak tetap.

Jika sebuah pencacah bekerja dengan frekuensi tetap yang sudah diketahui besarnya, dikatakan pencacah tersebut bekerja sebagai *timer*, karena keadaan pencacah tersebut setara dengan waktu yang bisa ditentukan dengan pasti.

Jika sebuah pencacah bekerja dengan frekuensi yang tidak tetap, dikatakan pencacah tersebut bekerja sebagai *counter*, keadaan pencacah tersebut hanyalah menyatakan banyaknya pulsa yang sudah diterima pencacah.

2.4.1.3. Komunikasi Data Serial

Untuk komunikasi data serial AT89C51 dilengkapi dengan *on chip serial port* (Port Serial di dalam keeping), sistemnya secara *full duplex* sehingga port serial ini masih dapat menerima data pada saat proses pengiriman data terjadi (Nalwan; 39). Saat proses pengiriman ataupun penerimaan data AT89C51 menggunakan SBUF sebagai penyangga (*buffer*), SBUF ini letaknya pada alamat 99H, sehingga jika AT89C51 sedang melakukan proses pembacaan data pertama

sedangkan data yang kedua belum diterima secara penuh maka data ini tidak akan hilang.

SBUFF terdiri atas dua buah register yang terletak di alamat yang sama yaitu 99H, kedua register tersebut adalah *transmite register* sifatnya *write only* (hanya dapat ditulis) dan *receive register* sifatnya *read only* (hanya dapat dibaca). Sistem kerjanya yaitu saat proses penerimaan data dari port serial, data yang masuk ke dalam port serial akan ditampung pada *receive register* dulu, kemudian diteruskan ke jalur *bus* internal pada saat pembacaan register SBUF. Sedangkan saat pengiriman data ke port serial data yang dituliskan dari *bus* internal ditampung pada *transmite register* dulu sebelum dikirim ke port serial.

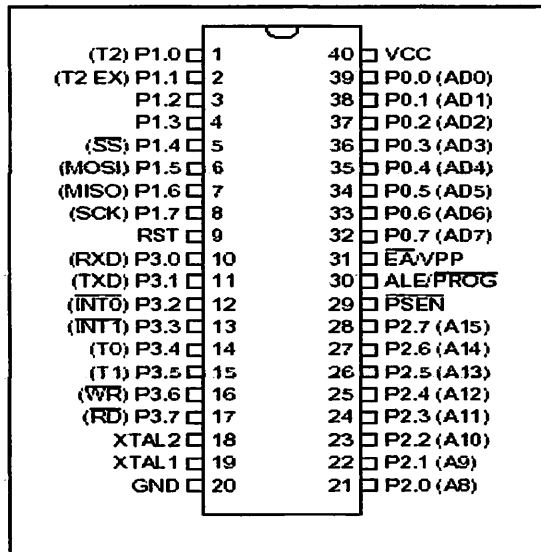
2.4.2. Mikrokontroler AT89S8252

Mikrokontroler AT89S8252 adalah mikrokontroler ATMEL kompetibel penuh dengan mikrokontroler keluarga MCS-51, dengan supply daya rendah, memiliki performance yang tinggi, dan merupakan mikrokontroler 8 k byte Downloadable Flash Memori yang dilengkapi 2 Kbyte EEPROM (*Electrical Erasable and programmable read Only Memory*) dan 256 Byte RAM internal. Dalam sistem mikrokontroler terdapat dua hal yang mendasar, yaitu: perangkat keras dan perangkat lunak yang keduanya saling terkait dan mendukung.

2.4.2.1. Perangkat keras Mikrokontroler AT89S8252

Mikrokontroler AT89S8252 merupakan pengembangan dari mikrokontroler standard MCS-51, dengan banyak kelebihan yang ditawarkan. Dipakainya downloadable flash memori memungkinkan mikrokontroler ini

bekerja sendiri tanpa diperlukan tambahan chip lainnya, dan flash memori dapat diprogram hingga seribu kali.



Gambar 2.7 Susunan Pin Dari AT 89S8252^[10]

Untuk pin-pin pada mikrokontoler AT89S825 berdasarkan penggunaan dan fungsinya banyak memiliki kesamaan dengan MCS-51. ada beberapa fungsi tambahan pada beberapa pin seperti :

1 Port 1

Pada port 1 mikrokontroler AT 89S8252 terdapat beberapa fungsi khusus, seperti pada tabel dibawah ini :

Tabel 2.3 Fungsi Khusus Pada Port 1 AT89S8252^[11]

| Port PIN | Fungsi Khusus |
|----------|---|
| P 1.0 | T2 (masukan luar untuk timer / counter 2) |
| P 1.1 | T2 EX (timer / counter capture / reload trigger dan control arah) |
| P 1.2 | - |
| P 1.3 | - |
| P 1.4 | SS (slave port select input) |
| P 1.5 | MOSI (master data ouput, slave dan input untuk kanal SPI) |

| | |
|-------|--|
| P 1.6 | MISO (master data input, slave data output untuk kanal SPI) |
| P 1.7 | SCK (master clock output, slave clock input untuk kanal SPI) |

2 Pin 31 (EA/VPP (*Externall Access/Program Supplay Voltage*))

Jika Pin EA diberi logika rendah maka mikrokontroler akan mengakses memori program eksternal. Sebaliknya jika Pin EA diberi logika tinggi, mikrokontroler akan mengakses memori program internal. Pin ini juga dipakai sebagai tegangan pemrograman EPROM atau *Flash Memori Internal*

2.4.2.2. SFR tambahan pada Atmel AT89S8252

Selain memiliki SFR (*Special Function Register*) seperti halnya pada MCS-51, mikrokontroler Atmel AT89S8252 memiliki tambahan SFR. Hal ini tak lain adalah karena adanya fitur tambahan pada mikrokontroler Atmel AT89S8252. SFR tambahan ini meliputi: T2CON (Timer 2 Register dengan alamat 0C8H), T2MOD (*Timer 2 Mode* dengan alamaat 0C9H), WMCON (*Watchdog and Memory Control Register* dengan alamat 96H), SPCR(*SPI Control Register* dengan alamat D5H), SPSR (*SPI Status Refister* dengan alamat AAH), SPDR(*SPI Data Register* dengan alamat 86H).

2.4.2.3. Data Memory (EEPROM) dan RAM

Berbeda dengan mokrokontroler standart MCS-51, mikrokontroler Atmel AT89S8252 juga dilengkapi dengan data memori yang berupa EEPROM (*Electrically Erasable Programmable Read Only Memory*). EEPROM yang ditanamkan ini besarnya 2 Kbyte dan dipakai untuk penyimpanan data.

EEPROM ini diakses dengan mengeset bit EEMEN pada register WMCON pada alamat 96H. Alamat EEPROM ini adalah 000H sampai 7FFH. Instruksi MOVX digunakan untuk mengakses EEPROM internal ini. Namun jika

ingin mengakses data memori luar (diluar mikrokontroler Atmel AT89S8252) dengan menggunakan instruksi MOVX ini maka bit EEMEN harus dibuat "0".

Bit EEMWE pada register WMCON harus di-*set* ke 1 sebelum proses penulisan pada lokasi EEPROM. Pada EEMWE di-*reser* ke "0" jika proses penulisan ke EEPROM telah selesai. Proses penulisan pada EEPROM dapat dilihat dengan membaca bit RDY/BSY pada SFR WMCON. Jika bit ini berlogika rendah maka penulisan EEPROM sedang berlangsung, jika bit ini berlogika tinggi maka penulisan sudah selesai dan penulisan lain dapat dimulai lagi. Sedangkan RAM yang ada pada mikrokontroler Atmel AT89S8252 adalah berkapasitas 256 byte. Penjelasan mengenai RAM ini adalah sama dengan RAM yang ada pada mikrokontroler standart MCS-51.

2.4.2.4. Programmable Watchdog Timer (WDT)

Pada mikrokontroler Atmel AT89S8252 juga dilkengkapi oleh *watchdog Timer*. Untuk mengatur rentang waktu (periode) pada WDT ini maka terdapat bit *prescaler* yang dapat mengatur rentang waktu yang dibutuhkan. Bit *prescaler* ini adalah bit PS0, PS1 dan PS2 pada register WMCON. Periode waktu pada WDT ini berkisar dari 16 mS sampai 2048 mS.

WDT dinonaktifkan oleh Power on Reset (POR) dan selama Power Down. WDT diaktifkan dengan men-*setting* bit WD TEN pada SFR WMCON (alamat 96H). Jika perhitungan waktu WDT telah selesai tanpa ada reset atau dilumpuhkan, maka suatu pulsa reset internal akan dihasilkan untuk mereset CPU.

2.4.2.5. Timer 2

Pada mikrokontroler Atmel AT89S8252 terdapat tambahan *Timer 2. Timer* yang lain adalah *timer 0* dan *timer 1*. Pada mikrokontroler Atmel AT89S8252 selain

(The following text is extremely faint and largely illegible due to the quality of the scan. It appears to be a multi-paragraph document, possibly a report or a letter, containing various sentences and phrases. The text is scattered across the page and is difficult to transcribe accurately.)

menggunakan *timer* 1 sebagai *baudrate* (untuk menjaga kompatibility dengan MCS-51) juga dapat menggunakan *Timer* 2 sebagai penghasil *baudrate* untuk *serial port*. *Timer* 2 ini merupakan *Timer/Counter* yang berukuran 16 bit yang dapat beroperasi sebagai *timer* atau dapat beroperasi sebagai penghitung kejadian dengan detak dari luar. Untuk mengatur fungsi ini dilakukan dengan mengatur bit C/T2 pada SFR T2CON. Terlihat bahwa jika bit logika *high* maka akan terpilih fungsi *counter*, dan sebaliknya jika bit ini logika *low* maka akan terpilih fungsi *Timer* 2.

Timer 2 ini memiliki 3 mode operasi yaitu: *capture*, *auto reload (up dan down counting)* dan *baud rate generator*. Untuk memilih mode ini dilakukan dengan mengatur bit pada SFR T2CON.

Timer 2 ini terdiri dari 2 buah *timer* 8 bit *register* yaitu TH2 dan TL2 dinaikkan tiap siklus mesin. Karena siklus mesin terdiri dari 1 *periode* osilasi, maka *count rate* menjadi 1/12 dari frekuensi osilator.

Tabel 2.4 Mode Operasi Timer 2^[12]

| RCLK+TCLK | CP/RL2 | TR2 | MODE |
|-----------|--------|-----|---------------------|
| 0 | 0 | 1 | 16 bit auto reload |
| 0 | 1 | 1 | 16 bit capture |
| 1 | X | 1 | Baud rate generator |
| X | X | 0 | Off |

2.5. Real Time Clock (RTC) DS12C887

Chip DS12C887 buatan Dallas Semikonduktor merupakan alat pencatat waktu yang baik, dalam *chip* itu tersimpan baterai litium sehingga pencatatan waktu tidak terhenti jika catu daya alat pemakainya terhenti dan proses dapat mengambil kapanpun hasil kerjanya tanpa harus mengatur.

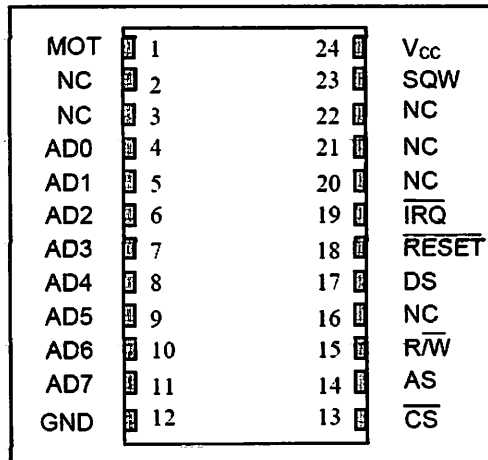
Waktu dicatat mulai dari detik, menit, jam, hari, bulan, tahun bahkan sampai abad dan penanggalanya dirancang bisa bekerja tanpa salah sampai tahun 2099. Pencatatan jam bisa dilakukan dalam format 24 jam perhari atau 12 jam AM/PM.

Selain dipakai sebagai pencatat waktu, DS12C887 bisa berfungsi sebagai alarm, bisa pula dipakai untuk membangkitkan sinyal periodik dengan frekuensi dari 2 HZ sampai dengan 8 KHZ^[10]. Karakteristik dari RTC DS12C887 adalah sebagai berikut :

1. Data tidak akan hilang maksimal sampai 10 tahun meskipun tidak ada catu daya.
2. Sub sistem tercakup sendiri meliputi litium, kwarsa, dan pendukung rangkaian.
3. Menghitung detik menit, jam, hari minggu, tanggal, bulan, dan tahun dengan lompatan pergantian tahun.
4. Data waktu kalender atau alarm dapat dipresentasikan dalam format biner / BCD
5. Format jam yang dipakai 24 atau 12 jam dengan AM dan PM dalam mode 12 jam.
6. *Timing bus* dapat dipilih antara Motorola dan Intel.

7. *Bus Multipleks* antara *address* dengan data untuk efisiensi 15 byte *clock register* kontrol dan 113 byte RAM untuk tujuan umum (bisa diprogram).

Secara lengkap struktur dari DS12C887 dapat dilihat pada gambar di bawah ini :



Gambar 2.8 Konfigurasi Kaki RTC DS12C887^[13]

2.5.1 Fungsi Pin – Pin RTC DS12C887

1. GND dan VCC

Merupakan pin catudaya, VCC dihubungkan pada catu daya +5 volt dan GND dihubungkan dengan *ground*.

2. MOT (*Mode Selection*)

Digunakan sebagai saklar pemilih mode diagram pewaktuan. Apabila dihubungkan dengan VCC berarti sistem diagram pewaktuan Motorola yang dipakai. Apabila dihubungkan dengan GND berarti sistem diagram pewaktuan Intel yang dipakai.

3. SQW (*Square Wave Output*)

Merupakan pin yang dapat dipilih untuk mengeluarkan satu dari tiga belas frekuensi keluaran yang tersedia. Besar frekuensi keluaran SQW dapat diubah

dengan program pada register A. Untuk mengaktifkan atau menonaktifkan sinyal SQW dipilih melalui bit SQWE pada register B.

4. AD0 – AD7 (*Multiplexed Bidirectional Address / Data Bus*)

Merupakan bus alamat / data dua arah.

5. AS (*Address Strobe Input*)

Merupakan masukan bagi sinyal yang digunakan untuk memisahkan bus data dan bus alamat (ALE). Tepi turun AS / ALE akan menyebabkan alamat ditahan dalam DS12C887. Tepi naik berikutnya yang terjadi pada AS akan meniadakan alamat tersebut tanpa memperhatikan kondisi penyemat CS aktif atau tidak.

6. DS (*Data Strobe or Read Input*)

Pin DS dapat diartikan sama dengan sinyal *Output Enable* (OE) pada komponen memori. Dihubungkan dengan sinyal RD yang berasal dari mikrokontroler untuk melakukan proses membaca data pada RAM internal RTC.

7. R / W (*Read / Write Input*)

Pin R / W dapat diartikan sama seperti sinyal *Write Enable* yang biasa dipakai pada komponen jenis RAM. Digunakan untuk proses penulisan pada RAM internal RTC.

8. CS (*Chip Select Input*)

Merupakan masukan untuk mengaktifkan peripheral RTC. Sinyal CS diperoleh dari dekoder alamat dengan alamat tertentu.

9. IRQ (*Interrupt Request Output*)

Sinyal IRQ merupakan sinyal aktif rendah, yang dipakai untuk interupsi ke mikrokontroler. Keluaran IRQ tetap rendah selama status bit yang menyebabkan

interupsi ada. Untuk mereset IRQ, mikrokontroler memberikan program pada register C RTC. Saat tidak terdapat interupsi, pin ini akan dalam kondisi impedansi tinggi (*high impedance*).

10. RSET (*Reset Input*)

Sinyal *RESET* diberikan dengan cara memberi logika rendah selama waktu yang dispesifikasikan, dan tidak berpengaruh pada unjuk kerja *clock*, kalender, dan fungsi RAM. Namun sinyal *RESET* dapat mengakibatkan beberapa *flag* direset menjadi nol.

2.5.2 Register – Register RTC DS12C887

RTC DS12C887 mempunyai 4 buah register kontrol dan 10 buah register data. Register data dipisahkan menjadi register waktu dan register alarm seperti yang tampak pada peta alamat RTC DS12C887. Sedangkan register kontrolnya adalah sebagai berikut :

1. Register A

| MSB | | | | | | LSB | |
|------|------|------|------|------|------|------|------|
| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
| UIP | DV2 | DV1 | DV0 | RS3 | RS2 | RS1 | RS0 |

1) UIP (*Update In Progress*)

Bit untuk memantau *flag update in progress*. Apabila bit ini bernilai 1, maka *update* transfer akan segera muncul. Akan tetapi bila bit ini bernilai 0, maka *update* transfer hanya akan muncul setelah 244 uS kemudian.

2) DV2, DV1, DV0

Ketiga bit ini berfungsi untuk mengatur osilator *on, off*, atau mereset perhitungan mundur. Konfigurasi 010 digunakan untuk mengatur osilator dalam keadaan *on*. Konfigurasi 11x digunakan untuk mengatur agar perhitungan mundur tertahan.

3) RS3, RS2, RS1, RS0

Bit – bit ini berfungsi sebagai 15 tingkat pembagi frekuensi yang dapat digunakan untuk membangkitkan sinyal output pada pin SQW atau *Periodic Interrupt Enable bit*.

2. Register B

| MSB | | | | LSB | | | |
|------|------|------|------|------|------|-------|------|
| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
| SET | PIE | AIE | UIE | SQWE | DM | 24/12 | DSE |

1) SET

Apabila bit ini bernilai 0, maka register data dapat *di-update* setiap detiknya. Akan tetapi apabila bit ini bernilai 1, maka register data tidak dapat *di-update*.

2) PIE (*Periodic Interrupt Enable*)

Bit yang berfungsi untuk membuat bit *periodic interrupt flag* dapat membangkitkan kondisi *low* pada pin IRQ.

3) AIE (*Alarm Interrupt Enable*)

Bit yang berfungsi untuk membuat bit *alarm flag* dapat membangkitkan kondisi *low* pada pin IRQ. Kondisi reset akan mengubah nilai bit ini menjadi 0.

4) UIE (*Update Ended Interrupt Enable*)

Bit yang berfungsi untuk membuat bit *update end flag* (UF) dapat membangkitkan kondisi *low* pada pin IRQ. Kondisi reset atau logika 1 pada bit SET akan mengubah nilai bit ini menjadi 0.

5) SQWE (*Square Wave Enable*)

Apabila bit ini bernilai 1, maka gelombang kotak dengan frekuensi yang ditentukan oleh pembagi frekuensi pada bit RS3, RS2, RS1, dan RS0 akan muncul di pin SQW.

6) DM (*Data Mode*)

Apabila bit ini bernilai 1, maka yang digunakan adalah mode biner. Apabila bit ini bernilai 0, maka yang digunakan adalah mode BCD.

7) 24 / 12

Apabila bit ini bernilai 1, maka yang digunakan adalah mode 24 jam, sedangkan apabila bit ini bernilai 0, maka yang digunakan adalah mode 12 jam.

8) DSE (*Daylight Saving Enable*)

Apabila bit ini bernilai 1, maka proses *update* akan terjadi pada saat 1:59:59 AM menjadi 3:00:00 AM pada hari minggu pertama di bulan april. Apabila bit ini bernilai 0, maka proses *update* akan terjadi pada saat 1:59:59 AM menjadi 1:00:00 AM pada hari minggu terakhir di bulan oktober.

3. Register C

| MSB | | | | | | LSB | |
|------|------|------|------|------|------|------|------|
| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
| IRQF | PF | AF | UF | 0 | 0 | 0 | 0 |

1) IRQF (*Interrupt Request Flag*)

Bit ini akan ber nilai 1 apabila salah satu kondisi di bawah ini terpenuhi :

$$PF = PIE = 1$$

$$AF = AIE = 1$$

$$UF = UIE = 1$$

$$\text{Contoh : } IRQF = (PF \times PIE) + (AF \times AIE) + (UF \times UIE)$$

2) PF (*Periodic Interrupt Flag*)

Bit ini akan bernilai 1 secara periodik yang sesuai dengan nilai bit RS3, RS2, RS1, dan RS0. Apabila bit PIE bernilai 1, maka logika 1 pada bit PF akan membangkitkan kondisi *low* pada pin IRQ. Bit ini akan bernilai 0 apabila register C telah dibaca atau kondisi reset.

3) AF (*Alarm Interrupt Flag*)

Bit ini akan bernilai 1 apabila waktu yang sedang berjalan sesuai dengan waktu yang diisikan pada register alarm. Apabila bit AIE bernilai 1, maka logika 1 pada bit AF akan membangkitkan kondisi *low* pada pin IRQ. Bit ini akan bernilai 0 apabila register C telah dibaca atau kondisi reset.

4) UF (*Update Ended Interrupt Flag*)

Bit ini akan bernilai 1 setelah siklus update. Apabila bit UIE bernilai 1, maka logika 1 pada bit UF akan membangkitkan kondisi *low* pada pin IRQ. Bit ini akan bernilai 0 apabila register C telah dibaca atau kondisi reset.

5) BIT3, BIT2, BIT1, dan BIT0

Keempat bit tersebut tidak digunakan di register C. Sifatnya adalah *read only* dan nilainya selalu 0.

4. Register D

| MSB | | | | LSB | | | |
|------|------|------|------|------|------|------|------|
| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
| VRT | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

1) VRT (*Valid RAM and Time*)

Bit ini akan selalu bernilai 1, yang menunjukkan bahwa *battery lithium* yang ada di dalam RTC DS12C887 masih bisa bekerja dengan baik. Apabila bit ini bernilai 0, maka *battery* internal sudah tidak layak digunakan dan isi dari data RTC dan data RAM patut dipertanyakan.

2) BIT6, BIT5, BIT4, BIT3, BIT2, BIT1, dan BIT0

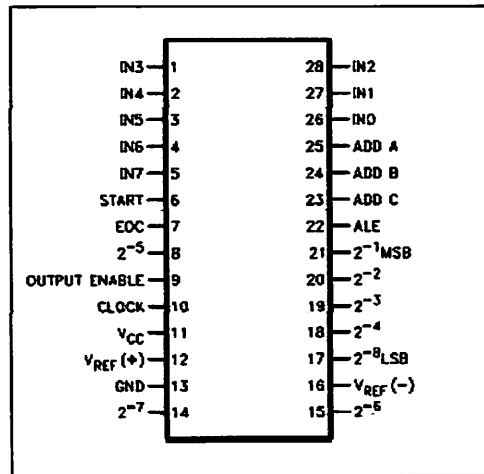
Ketujuh bit tersebut tidak digunakan di register D. Sifatnya adalah *read only* dan nilainya selalu 0. ^[14]

2.6. Analog To Digital Converter (ADC) 0808

Agar dapat mengukur atau menolah suatu fariabel fisik yang umumnya dalam bentuk besaran analog dengan piranti digital, variabel tersebut harus terlebih dahulu diubah menjadi variabel digital yang nilainya proposional dengan nilai variabel yang akan diukur atau diolah tersebut. Proses konversi ini dilakukan oleh konverte analok ke digital (*analog to digital converter*).

Spesifikasi penting lain selain ketelitian (akurasi) dan linieritas adalah waktu konversi (*conversion time*). Waktu konversi ADC adalah waktu yang diperlukan ADC untuk menghasilkan kode biner yang valid tegangan masukan yang diberikan. Semakin pendek waktu konversi berarti kecepatan konversi semakin tinggi.

Dalam pembuatan alat ini digunakan ADC jenis 0808 yang mempunyai 8 bit channel multiplekser dan menggunakan metode pendekatan *Successive Approximation Register* (SAR). Waktu konversi (*conversion time*) ADC ini adalah 100 μ s dengan rentang an input sebesar 0 sampai dengan 5 V dan supply tegangan sebesar 5 V.

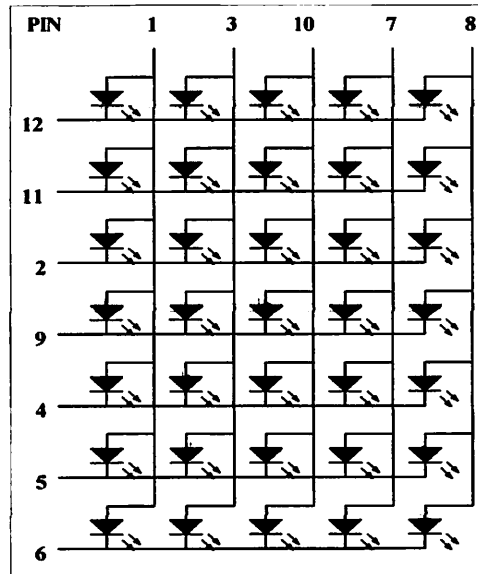


Gambar 2.9 Konfigurasi Kaki ADC0808^[15]

2.7. Komponen Pendukung Tampilan Dot Matrik

2.7.1. Dot matrik

Piranti dot matrik mempunyai sejumlah besar cahaya yang berbentuk seperti titik. Untuk menghidupkan sebuah LED dalam matrik ini, harus menghubungkan tegangan dengan anodanya dan *groundkan* katodanya. Dengan memberi tegangan lebih dari sebuah kolom dan *groundkan* lebih dari satu baris, maka dapat memperagakan setiap angka desimal, setiap huruf abjad, serta berbagai lambang lain. Tampilan matrik titik (*dot matrik*) terdiri dari sejumlah LED yang disusun pada baris dan kolom. Susunan yang terdapat di pasaran adalah matrik 5x7 titik, 5x8, dan 16x16 titik. Pada tampilan yang dipergunakan adalah tampilan 5x7 titik, seperti pada gambar dibawah ini.



Gambar 2.10 *Display Dot Matrik 5x7*^[16]

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

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

2.7.2. Dekoder Matrik LED

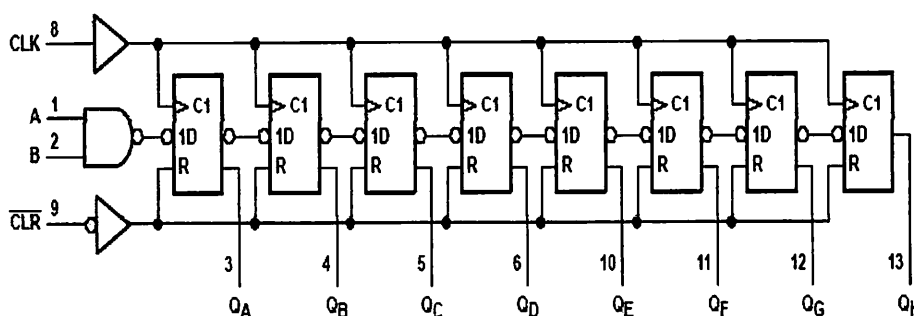
Untuk menyalakan huruf alfanumerik dengan pembacaan dot matik, LED yang dibutuhkan tidak dinyalakan serentak. Sebagai contoh , pada gambar 2.13

dimulai dengan menggroundkan baris pertama dan memasang tegangan pada kolom vertikal yang sesuai. Setelah itu, baris LED 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 tampil tanpa kedipan.

Rangkaian yang menjalankan matrik LED cukup rumit karena harus melakukan proses *scanning* baris-baris horizontal dan pada saat yang sama memasang tegangan pada kolom yang sesuai. Secara umum pencacah lingkaran (*ring counter*) melakukan *scan* pada baris horizontal, sedangkan memori berisi data kolom LED yang nyala memberikan tegangan pada kolom-kolom vertikal.

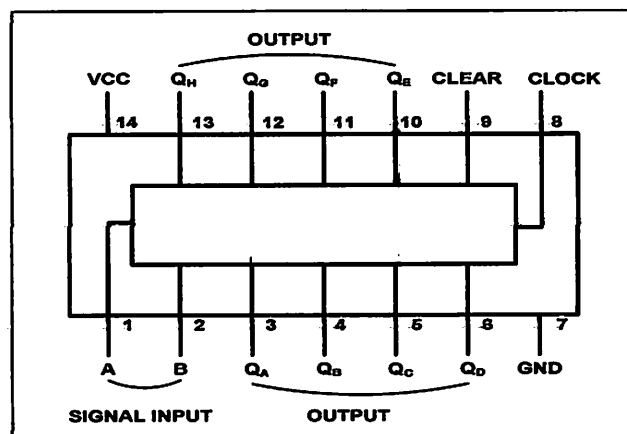
2.7.3. Register Geser 8 Bit *Serial In Parallel Out (SIPO)*

Bagian ini akan merinci salah satu dari sekian banyak register geser, yaitu register yang masuknya seri dan keluarannya parallel 8 bit. IC yang digunakan adalah IC 74HC164 yang merupakan register 8 bit yang terpicu sentuh dengan masukan serial. Semua keluaran parallel tersedia atas setiap *flip-flop D internal*. Diagram rinci tersedia dalam Gambar 2.14, memperlihatkan penggunaan delapan *flip-flop D internal*, masing-masing dengan keluaran data paralelnya (QA sampai QB).



Gambar 2.11 Diagram Detail Register Geser 8 Bit 74HC164^[17]

IC SN74HC164 terlihat memiliki sebuah masukan serial. Data dimasukkan secara serial melalui salah satu dari dua masukan (A dan B). perhatikan dalam gambar 2.12, kedua masukan (A dan B) di NAND-kan. Kedua masukan ini bisa digabung menjadi satu masukan atau bisa juga salah satu ditentukan sebagai logika tinggi, sedangkan yang satu lagi untuk pemasukan data. Masukan data pengatur ulang utama *clear* untuk IC 74HC164 merupakan input aktif rendah. Dalam tabel kebenaran (Tabel 2.5) memperlihatkan bahwa bila diaktifkan, masukan *clear* akan mematikan semua input yang lainnya, dan mengembalikan semua *flip-flop* ke 0. IC SN74HC164 menggeser data satu tempat ke kanan dalam setiap peralihan rendah ke tinggi dalam masukan *clock*.



Gambar 2.12 Konfigurasi Kaki 74HC164^[18]

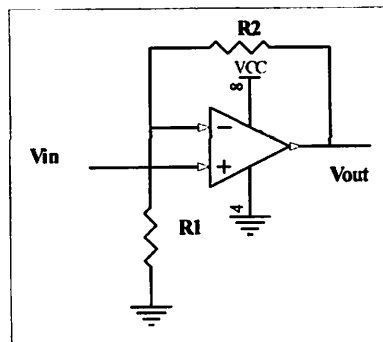
2.8. Penguat (Op-Amp)

Operational amplifier atau disingkat Op-Amp merupakan salah satu komponen analog yang populer digunakan dalam berbagai aplikasi rangkaian elektronika. Aplikasi Op-Amp populer yang paling sering dipakai dipakai antara

lain rangkaian *inverter*, *non-inverter*, *integrator* dan *differensiator*. Pada hal ini Op Am-yang dibahas adalah penguat *non-inverting*.

2.8.1. Non Inverting Amplifier

Prinsip utama rangkaian penguat non-inverting adalah seperti diperlihatkan pada gambar 2.13 berikut ini. Seperti namanya, penguat ini memiliki masukan yang dibuat melalui input non-inverting. Dengan demikian tegangan keluaran rangkaian ini akan satu fasa dengan tegangan inputnya. Untuk menganalisa rangkaian inverting.



Gambar 2.13 Penguat Non Inverting^[19]

Dengan menggunakan aturan 2, kita uraikan dulu beberapa fakta yang ada, antara lain :

$$V_{in} = V_{(+)}$$

$$V_{(+)} = V_{(-)} = V_{in} \dots\dots\text{lihat aturan 1}$$

Dari sini ketahuilah tegangan jepit pada R_2 adalah $V_{out} - V_{(-)} = V_{out} - V_{in}$, atau

$$I_{out} = \left(\frac{V_{out} - V_{in}}{R_2} \right). \text{ Lalu tegangan jepit pada } R_1 \text{ adalah } V_{-} = V_{in}, \text{ yang berarti arus}$$

$$I_{R1} = \frac{V_{in}}{R_1}.$$

Hukum kirchoff pada titik input inverting merupakan fakta yang mengatakan bahwa :

$$I_{out} + I_{(-)} = I_{R1}$$

Aturan 2 mengatakan bahwa $I_{(-)} = 0$ dan jika disubsitusikan ke rumus yang sebelumnya, maka diperoleh $I_{out} = I_{R1}$ dan jika ditulis dengan tegangan jepit

masing-masing maka diperoleh $\left(\frac{V_{out} - V_{in}}{R_2} \right) = \frac{V_{in}}{R_1}$ yang kemudian dapat

disederhanakan menjadi : $V_{out} = V_{in} \left(1 + \frac{R_2}{R_1} \right)$

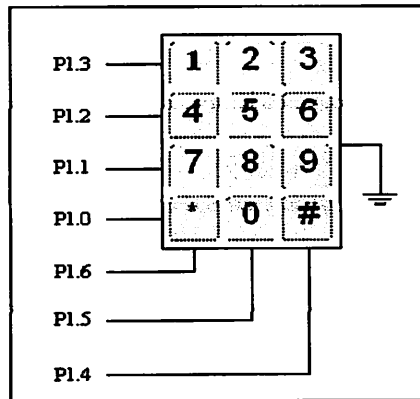
Jika penguatan G adalah perbandingan tegangan keluaran terhadap tegangan masukan, maka didapat penguatan op-amp non-inverting :

$$G = \frac{V_{out}}{V_{in}} = 1 + \left(\frac{R_2}{R_1} \right) \dots (2)$$

Impedansi untuk rangkaian op-amp non-inverting adalah impedansi dari input non-inverting op-amp tersebut.

2.9. Keypad

Karena bentuknya yang sederhana (menyerupai keyboard kecil), dan mudah pengoprasiannya, maka oleh penulis keypad ini digunakan untuk memasukkan pesan dan juga digunakan untuk mengeset waktu. Keypad yang ada umumnya jenis keypad 4 baris, 4 kolom (keypad 4X4) dan jenis 4 baris 3 kolom (keypad 4X3). Dalam hal ini keypad yang akan digunakan adalah keypad 4X3 dengan sebuah *common*. Gambar dari keypad adalah sebagai berikut :



Gambar 2.14 Keypad 4x3^[20]

Prinsip kerja keypad ini menggunakan metode matrik. Sinyal atau tegangan (sesuai dengan gambar diatas) diberikan secara bergantian pada masing-masing pin. Bila pin 1 mendapat sinyal 1 (tegangan 5V), maka pin lainnya mendapat sinyal 0 (tegangan 0V). metode pengiriman sinyal seperti ini dapat menggunakan bilangan biner. Bila data yang dikirim 1000, berate pin 1 mendapat sinyal 1, dan pin laninya mendapat sinyal 0. bila data dikirim 0010, bearti sinyal 1 diberikan pada pin3.

Pada saat data yang dikirim 0100 (pin 2 aktif), dan tombol yang ada padanya (4, 5 dan 6) ditekan, maka sinyal tersebut akan dikirim ke bagian detektor sesuai dengan tombol yang ditekan. Jika tombol 4 ditekan, maka sinyal diteruskan pada detector pin 1, jika tombol 4 dan 6 ditekan , maka sinyal akan diteruskan ke detector pin 1 dan pin 3. data pada detektor ini juga dapat dibaca secara biner. Bila data yang diterima 010, berarti pin 2 pada detektor mendapat sinyal (tombol 5 ditekan). Bila data yang diterima 011, berarti tombol 5 dan tombol 6 ditekan.

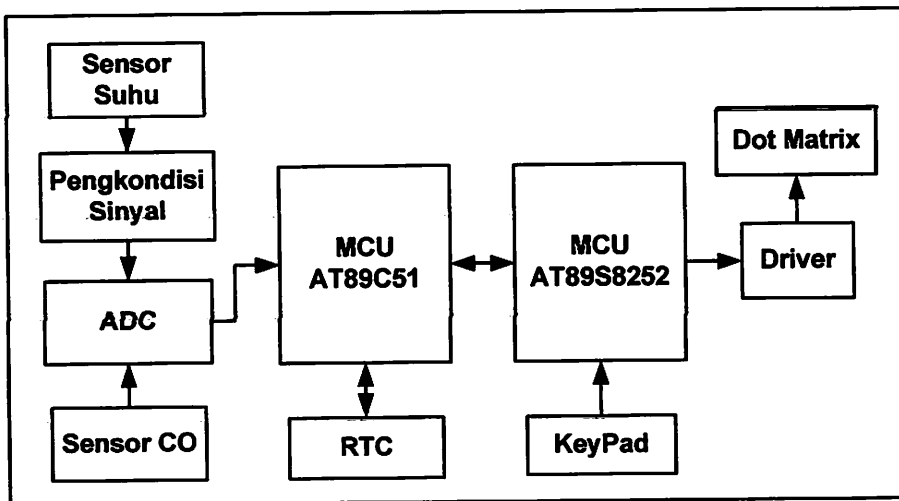
BAB III

PERANCANGAN DAN PEMBUATAN ALAT UKUR SUHU DAN KARBON MONOKSIDA

3.1. Pendahuluan

Dalam perancangan dan pembuatan alat ukur suhu dan karbon monoksida (CO) ini terdiri dari beberapa rangkaian yang saling berhubungan satu dengan yang lainnya. Diantaranya terdiri dari 2 buah mikrokontroler yaitu AT89C51 dan AT89S8252, ADC0808, RTC12DS887, Sensor CO(AF56), Sensor Suhu (LM35), penguat, dot matrik beserta drivernya. Dari beberapa komponen tersebut digabung menjadi beberapa rangkaian yang saling berhubungan.

3.2. Diagram Blok Sistem



Gambar 3.1 Diagram Blok Sistem Alat

3.3. Prinsip Kerja Alat

1. Dua buah sensor yaitu sensor gas dan sensor suhu, diletakkan dibagian utama dari alat yang merupakan pendeteksi besarnya kadar karbon monoksida dan suhu.
2. Untuk sensor gas (AF56) outputnya langsung dimasukkan ke ADC tanpa melalui rangkaian pengkondisi sinyal. Sedangkan untuk sensor suhu (LM35) outputnya dimasukkan ke pengkondisi sinyal agar outputnya sesuai dengan tegangan dalam ADC yaitu antara 0 – 5 volt.
3. Output yang telah dikondisikan dimasukkan ke ADC0808 pada pin IN0, sedangkan output dari sensor gas masuk ke pin IN1. dalam ADC output yang masih berupa data analog diubah menjadi data digital.
4. Data yang telah diubah menjadi digital oleh ADC masuk ke mikrokontroler yaitu pada port 1 sebagai input. Di dalam mikrokontroler data tersebut diolah atau diproses untuk mengetahui besarnya suhu dan kadar karbon monoksida yang terdeteksi.
5. RTC yang merupakan input untuk data waktu yang dihubungkan ke mikrokontroler, kemudian didalam mikrokontroler diprogram disesuaikan dengan keadaan waktu yang dikehendaki. Pada alat ini RTC digunakan untuk menunjukkan tanggal, bulan dan tahun.
6. Untuk mikrokontroler AT89S8252 khusus digunakan sebagai proses kontrol dan scanning dot matrik. Sehingga untuk menampilkan data hasil pengukuran dan data waktu maka antara kedua mikrokontroler dihubungkan melalui pin TX dan RX yang digunakan untuk mengirimkan data atau sebagai komunikasi serial.

7. Keypad digunakan untuk mengeset jam sesuai dengan kondisi sekarang.
8. Untuk menampilkan data pada dot matrik tidak langsung ditampilkan semua melainkan bergantian karena dot matrik hanya berjumlah 10 karakter, sehingga cara menampilkanya dengan cara menggeser atau menampilkan tulisan secara berjalan
9. Driver dot matrik, digunakan untuk memicu tegangan output dari mikrokontroler agar dapat menggerakkan atau menjalankan dot matrik.

3.4. Perancangan Perangkat Keras

3.4.1. Sensor Suhu

Pada dasarnya sensor suhu pada alat ini adalah untuk mengetahui suhu yang ada di jalan raya. Pada perancangan dan pembuatan alat ini menggunakan sensor LM35, yang memiliki kelebihan pada outputnya yang linier yaitu tegangan output sebesar 10mV setiap kenaikan suhu 1°C sehingga memudahkan dalam pengolahan datanya.

Untuk menghitung tegangan output dari sensor ini pada saat membaca temperature adalah :

$$V_{out} = \text{Temp} \times 10\text{mV}/^{\circ}\text{C}$$

Dimana :

Vout : Tegangan output dari sensor LM35 (Volt)

Temp : Besarnya suhu yang dibaca sensor (°C)

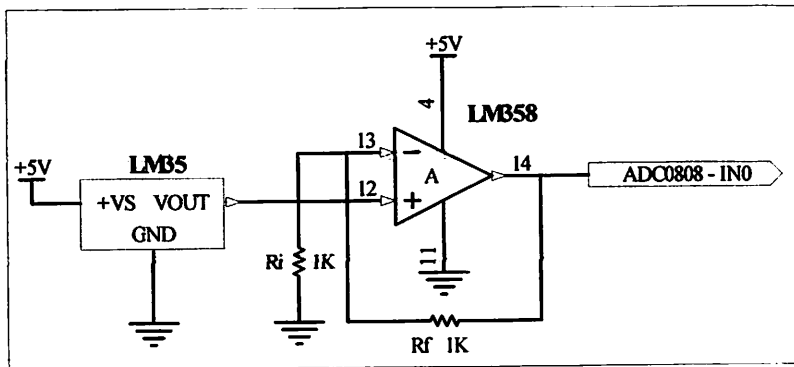
Sebagai contoh :

Jika suhu yang diukur 40 °C maka tegangan output sensor LM35 adalah :

$$V_{out} = 40\text{ }^{\circ}\text{C} \times 10\text{ mV}/^{\circ}\text{C}$$

$$V_{out} = 400 \text{ mV}$$

Untuk menaikkan resolusi pembacaan data suhu maka output dari sensor diumpankan kesuatu rangkaian penguat non inverting dengan menggunakan IC Op-Amp LM358. untuk lebih jelasnya dapat dilihat pada rangkaian dibawah ini :



Gambar 3.2 Rangkaian Sensor Suhu dan Pengkondisi Sinyal

Persamaan yang digunakan adalah sebagai berikut :

$$A = \left(1 + \frac{R_f}{R_i} \right)$$

Dari rumus tersebut dapat kita hitung besarnya penguatan dari output sensor LM35.

Diatas diketahui : $R_i = 1 \text{ k}$, $R_f = 1 \text{ k}$, maka penguatannya :

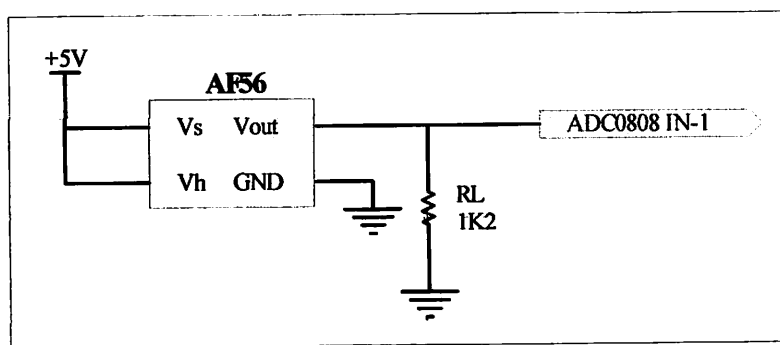
$$A = \left(1 + \frac{R_f}{R_i} \right)$$

$$A = \left(1 + \frac{1k}{1k} \right) = 2 \text{ kali}$$

Jadi dari perhitungan diatas diketahui penguatannya adalah 2 kali

3.4.2. Sensor CO

Pada perancangan ini sensor digunakan untuk mengukur berapa kadar gas karbon monoksida (CO) di suatu jalan raya. Sensor yang digunakan adalah Type (scimarec) AF-56 yang dapat bekerja pada temperatur -10°C sampai 55°C . Dari sensor ini bisa diketahui besarnya kadar CO sesuai dengan output yang dihasilkan ang berupa tegangan. Berikut adalah gambar rangkaian sensor gas AF56 :



Gambar 3.3 Rangkaian Sensor Gas AF-56

Besarnya tahanan sensor (R_s) dapat di ketahui dengan persamaan sebagai berikut :

$$R_s = \frac{V_c - V_{out}}{V_{out}} \cdot R_L$$

Maka dari situ kita dapat mencari besarnya daya yang dibutuhkan sensor (P_s) dengan rumus :

$$P_s = \frac{V_c^2 - R_s}{(R_s + R_L)^2}$$

Dimana :

R_s = Tahanan sensor

R_L = Tahanan beban (Variabel)

V_c = Tegangan kerja pada rangkaian

V_{out} = Tegangan keluaran (V_o)

P_s = Daya yang dibutuhkan oleh sensor ($P_s \leq 15 \text{ mW}$)

Sebagai contoh :

Diketahui hasil pengukuran diperoleh kadar CO adalah 1 % , $R_l = 1200 \text{ Ohm}$,

$V_{out} = 0,083 \text{ Volt}$, tahanan sensornya (R_s) adalah:

$$R_s = \frac{V_c - V_{out}}{V_{out}} \cdot R_l$$

$$R_s = \frac{5 - 0,083}{0,083} \cdot 1200 = 71089,15 \text{ Ohm} = 71,089 \text{ K}$$

Diketahui hasil pengukuran diperoleh kadar CO adalah 10 % , $R_l = 1200 \text{ Ohm}$,

$V_{out} = 0,838 \text{ Volt}$, tahanan sensornya (R_s) adalah:

$$R_s = \frac{V_c - V_{out}}{V_{out}} \cdot R_l$$

$$R_s = \frac{5 - 0,838}{0,838} \cdot 1200 = 5959,9 \text{ Ohm} = 5,959 \text{ K}$$

3.4.3. ADC (*Analog to Digital Converter*)

Dalam pembuatan alat ini juga dibutuhkan pengubah sinyal analog menjadi sinyal digital atau disebut *Analog to Digital Converter* (ADC), hal ini disebabkan karena sinyal-sinyal yang didapat dari sensor suhu dan sensor CO adalah berupa sinyal analog sedangkan rangkaian kontrolnya menggunakan system digital sehingga membutuhkan input berupa sinyal digital. Jenis ADC yang digunakan adalah ADC0808 yang memiliki kelebihan yaitu dapat menerima hingga 8 inputan. Pada alat ini hanya menggunakan 2 inputan saja yaitu ADD A (pin 25) sebagai inputan dari pengkondisi sinyal sensor suhu, sedangkan ADD B

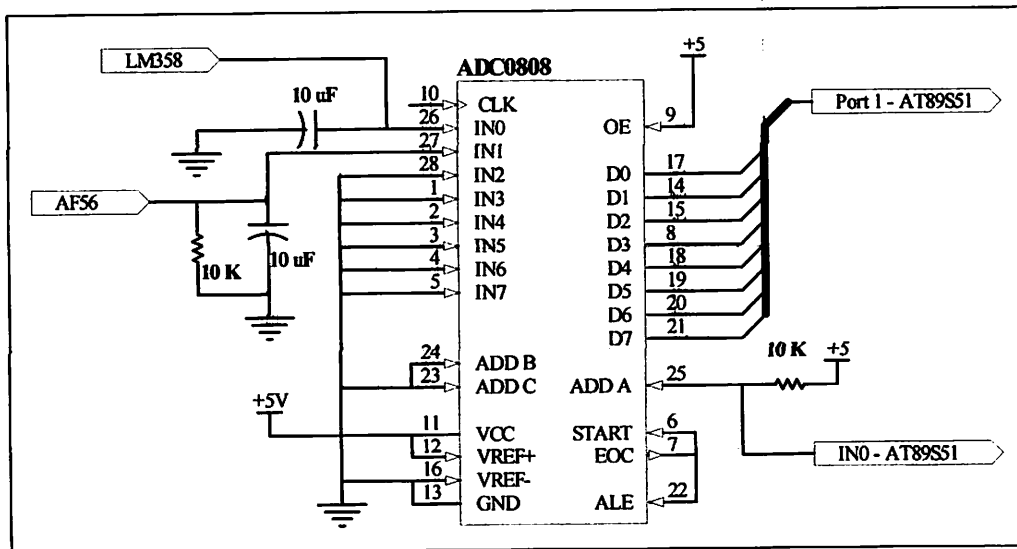
(pin 24) dan ADD C (pin 23) digroundkan karena input yang dipakai hanya 2 saja.

Untuk lebih jelasnya dapat kita lihat pada table dibawah ini :

Tabel 3.1 Address Decoder ADC0808^[22]

| Selected Analog Channel | Address Line | | |
|-------------------------|--------------|---|---|
| | C | B | A |
| IN0 | L | L | L |
| IN1 | L | L | H |
| IN2 | L | H | L |
| IN3 | L | H | H |
| IN4 | H | L | L |
| IN5 | H | L | H |
| IN6 | H | H | L |
| IN7 | H | H | H |

Berikut adalah gambar dari rangkaian ADC0808 :



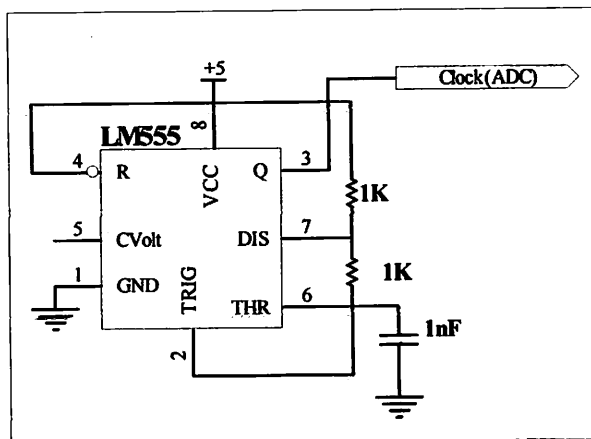
Gambar 3.4 Rangkaian ADC0808

Penjelasan dari gambar diatas :

1. IN0 (pin 26) merupakan inputan yang berupa sinyal analog dari pengkondisi sinyal sensor suhu LM35, IN1 (pin 27) merupakan inputan yang berupa sinyal analog yang berasal dari sensor CO (AF56), sedangkan IN2 – IN7 digroundkan karena tidak digunakan.
2. D0 – D7 digunakan untuk mengirim data digital ke mikrokontroller.
3. ADD A digunakan untuk menerima sinyal dari mikrokontroller untuk memilih data antara sensor suhu atau sensor CO.

Pada ADC0808 tidak memiliki *clock* internal sehingga untuk mengaktifkannya diperlukan *clock* eksternal. Dalam hal ini menggunakan IC 555.

Berikut ini adalah gambar dari rangkaian *clock* eksternal ADC0808 :



Gambar 3.5 Rangkaian *Clock* Eksternal Untuk ADC0808

Dari gambar rangkaian diatas dapat kita hitung besarnya frekuensi yang dihasilkan adalah sebagai berikut :

$$f = \frac{1}{1,1.R.C}$$

$$f = \frac{1}{1,1.(1.10^3 + 1.10^3).(10.10^{-9})}$$

$$f = \frac{1}{2,2 \cdot 10^{-5}}$$

$$f = \frac{1 \cdot 10^5}{2,2}$$

$$f = 45454,54 \text{ Hz}$$

$$f = 45,45 \text{ KHz}$$

Berdasarkan perhitungan diatas dapat diketahui frekuensi yang dihasilkan oleh rangkaian clock eksternal adalah 45,45 KHz. Jadi jika dilihat pada datasheet ADC0808, frekuensi *clock* yang dibutuhkan oleh ADC0808 adalah antara 10 – 280 KHz. Sehingga frekuensi yang dihasilkan oleh rangkaian clock eksternal ini sudah sesuai dan memenuhi syarat.

Pada ADC0808 *Vref* diset pada kondisi *Full Range* yaitu *Vref* (-) diset pada 0 volt, sedangkan untuk *Vref* (+) diset pada 5 volt. Pada saat 0 volt yaitu pada 0 sedangkan pada 5 volt pada 255. Karena kondisi tersebut maka *Vref*-nya dapat diketahui yaitu sebesar 5 volt. Dari data *Vref* tersebut dapat dicari *output* dari ADC dengan persamaan sebagai berikut :

$$V_{ref} = 5 \text{ volt}$$

$$\text{Resolusi per bit} = \frac{V_{ref}}{2^8} = \frac{5}{255} = 0.0196 \text{ mV}$$

Contoh :

$$V_{analog} = 0.5 \text{ volt}$$

$$\text{Data ADC} = \frac{V_{analog}}{\text{resolusi}} = \frac{0,5}{0,0196 \text{ mV}} = 25_{(10)}$$

Berikut hasil dari beberapa perhitungan dari output ADC :

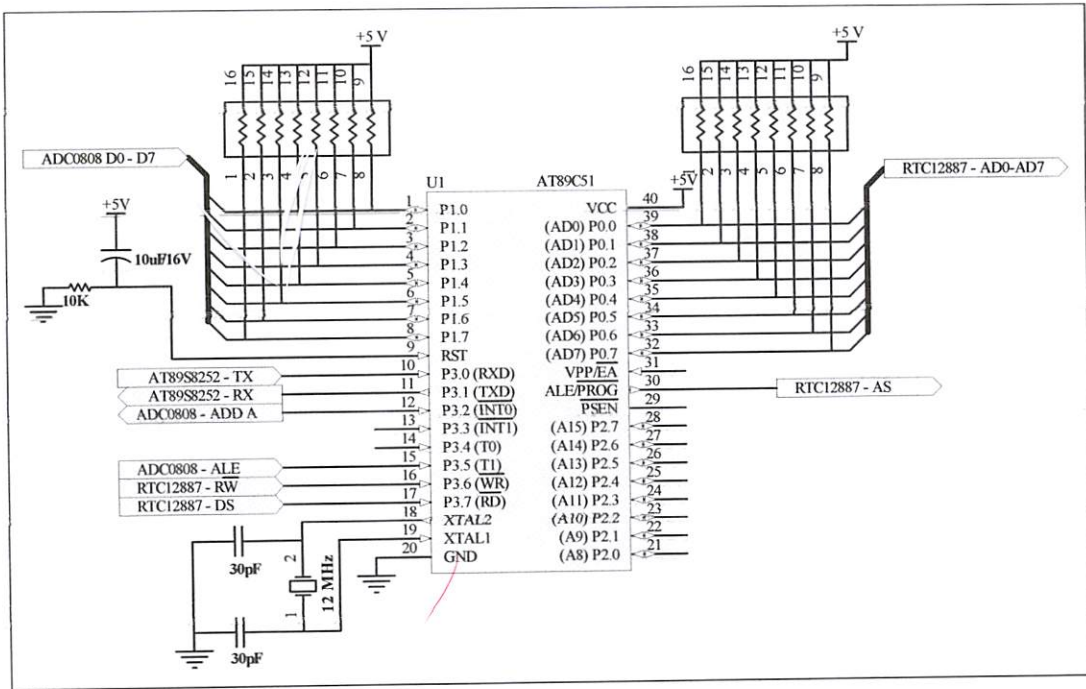
Tabel 3.2. Hasil Perhitungan *Output* ADC0808

| Tegangan Input ADC (V) | Output ADC | |
|------------------------|------------|----------|
| | Desimal | Biner |
| 0,5 | 25 | 00011001 |
| 1,0 | 51 | 00110011 |
| 1,5 | 76 | 01001100 |
| 2,0 | 102 | 01100110 |
| 2,5 | 127 | 01111111 |
| 3,0 | 153 | 10011001 |
| 3,5 | 178 | 10110010 |
| 4,0 | 204 | 11001100 |
| 4,5 | 229 | 11100101 |
| 5,0 | 255 | 11111111 |

3.4.4. Mikrokontroler

Pada perancangan dan pembuatan alat ini menggunakan dua buah mikrokontroler 8 bit buatan ATMEL yaitu AT89C51 dan AT89S8252. AT89C51 digunakan untuk mengolah data dan menyimpan data dari ADC dan RTC, sedangkan AT89S8252 digunakan untuk menyimpan karakter-karakter yang akan ditampilkan di display dot matrik. Penggunaan dua buah mikrokontroler karena disamping kekurangan port juga untuk mempercepat akses data yang akan ditampilkan di display dot matrik.

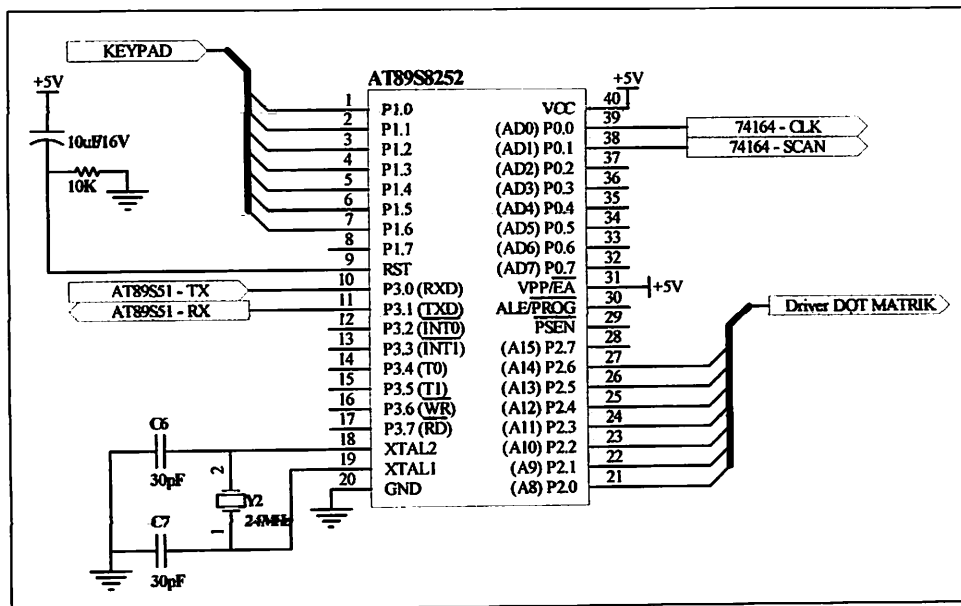
Berikut ini adalah gambar rangkaian mikrokontroller AT89C51 dan AT89S8252 :



Gambar 3.6 Rangkaian Mikrokontroler AT89C51

Berikut penjelasan dari gambar diatas :

1. Port 1 digunakan sebagai port data dari ADC0808.
2. Port 0, digunakan sebagai port data dari RTC DS12887.
3. Port 3.0 (RXD) dan port 3.1 (TXD) digunakan sebagai komunikasi data yaitu untuk menerima dan mengirim data ke mikrokontroler AT89S8252.
4. Port 3.3 digunakan untuk mengirimkan data ke ADC pada port ADD A yang berupa logika *High* atau *Low*. Sedangkan port 3.5, digunakan sebagai port pengontrol terhadap proses di ADC0808.
5. Pada port 3.6 (WR), port 3.7(DS) dan pin 30 (ALE/PROG) digunakan sebagai pengontrol terhadap semua proses yang terjadi di RTC.



Gambar 3.7 Rangkaian Mikrokontroler AT89S8252

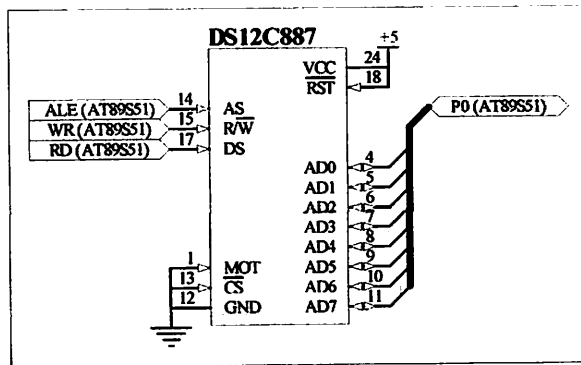
Berikut penjelasan dari gambar diatas :

1. Port 3.0 (RXD) dan port 3.1 (TXD), sebagai port untuk menerima dan mengirimkan data secara serial ke mikrokontroler AT89C51
2. Port 1, sebagai port data dari keypad untuk pengesetan jam.
3. Port 2, digunakan sebagai port data ke driver dot matrik.
4. Port 0.0 dan port 0.1, digunakan sebagai data dan *Clock* pada 74164 *shift register*.

3.4.5. RTC (*Real Time Clock*)

RTC (*Real Time Clock*) yang digunakan adalah DS12C887 sebagai inputan data waktu. Untuk menampilkan waktu (jam dan tanggal) mikrokontroler meminta data waktu pada RTC dengan perintah tertentu kemudian RTC baru mengirimkan data waktu pada mikrokontroler untuk diolah dan ditampilkan.

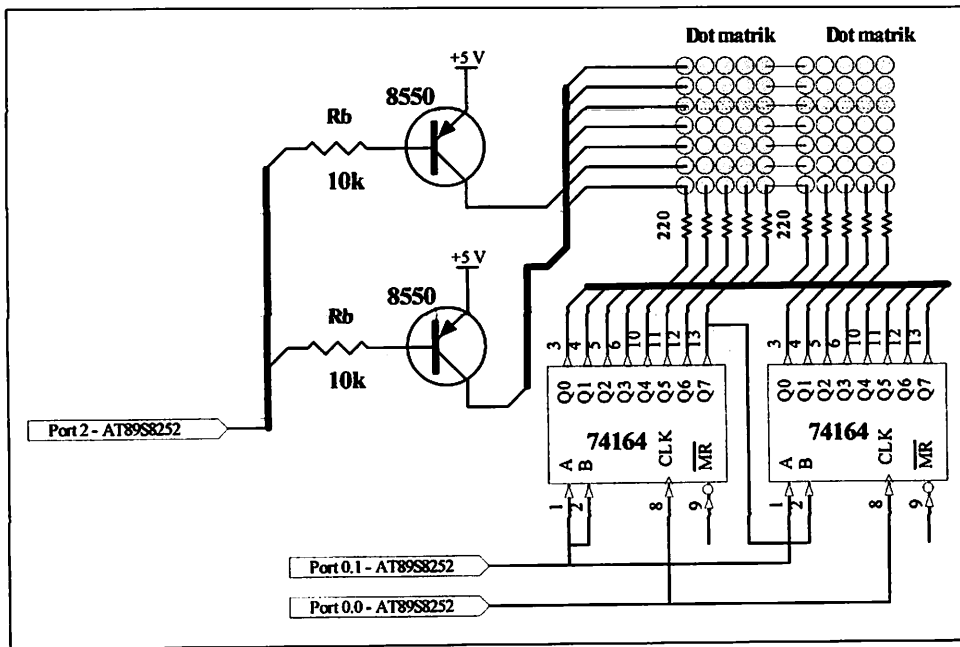
Proses Inisialisasi RTC diatur oleh mikrokontroler. Ada empat sinyal pengatur untuk menentukan kerja dari RTC ini yaitu AS (*Address Strobe*), WR (*Write*), DS (*Data Strobe*), dan CS (*Chip Select*). CS berfungsi untuk mengaktifkan dan menonaktif RTC. Berikut ini rangkaian dari RTC :



Gambar 3.8 Rangkaian RTC

3.4.6. Dot Matrik

Display LED dot matrik yang digunakan adalah dengan ukuran 5x7 sebanyak 12 buah. Pada setiap kaki anoda LED dihubungkan semua. Sedangkan pada katoda LED dihubungkan dengan keluaran IC 74164 yang berfungsi sebagai scan kolom. Untuk 10 buah LED dot matrik membentuk 50 titik kolom dan 7 titik baris, berikut adalah gambar dari rangkaian display dot matrik :



Gambar 3.9 Rangkaian *Driver* Dot matrik

Dari perancangan rangkaian diatas kita dapat menentukan nilai nilai R_b dan R_c . Untuk mencari nilai R_b dengan menggunakan persamaan sebagai berikut :

$$R_b = \frac{V_{EE} - V_{BE} - V_{ol}}{I_B} \quad I_B = \frac{I_C}{h_{fe}}$$

Diketahui dari *data sheet transistor 8550* besarnya $I_C = 80 \text{ mA}$, $h_{fe} = 190$, $V_{BE} = 0,7 \text{ Volt}$, sehingga :

$$\begin{aligned} I_B &= \frac{80 \text{ mA}}{190} = 0,42 \text{ mA} & R_b &= \frac{5 - 0,7 - 0,5}{0,42 \text{ mA}} \\ & & &= \frac{3,8}{0,42 \text{ mA}} \\ & & &= 9,04 \text{ K} \end{aligned}$$

Sedangkan untuk menentukan besarnya nilai R_c yaitu dengan menggunakan persamaan sebagai berikut :

$$R_c = \frac{V_{EE} - V_{EC}}{I_{LED}}$$

Diketahui dari *data sheet* LED, $I_{LED} = 20 \text{ mA}$. Kemudian dari *data sheet* transistor 8550, $V_{EC} = 0,28 \text{ volt}$. Sehingga R_c adalah :

$$\begin{aligned} R_c &= \frac{5 - 0,28}{20 \text{ mA}} = \frac{4,72}{20 \text{ mA}} \\ &= 0,236 \text{ K} \\ &= 236 \Omega \end{aligned}$$

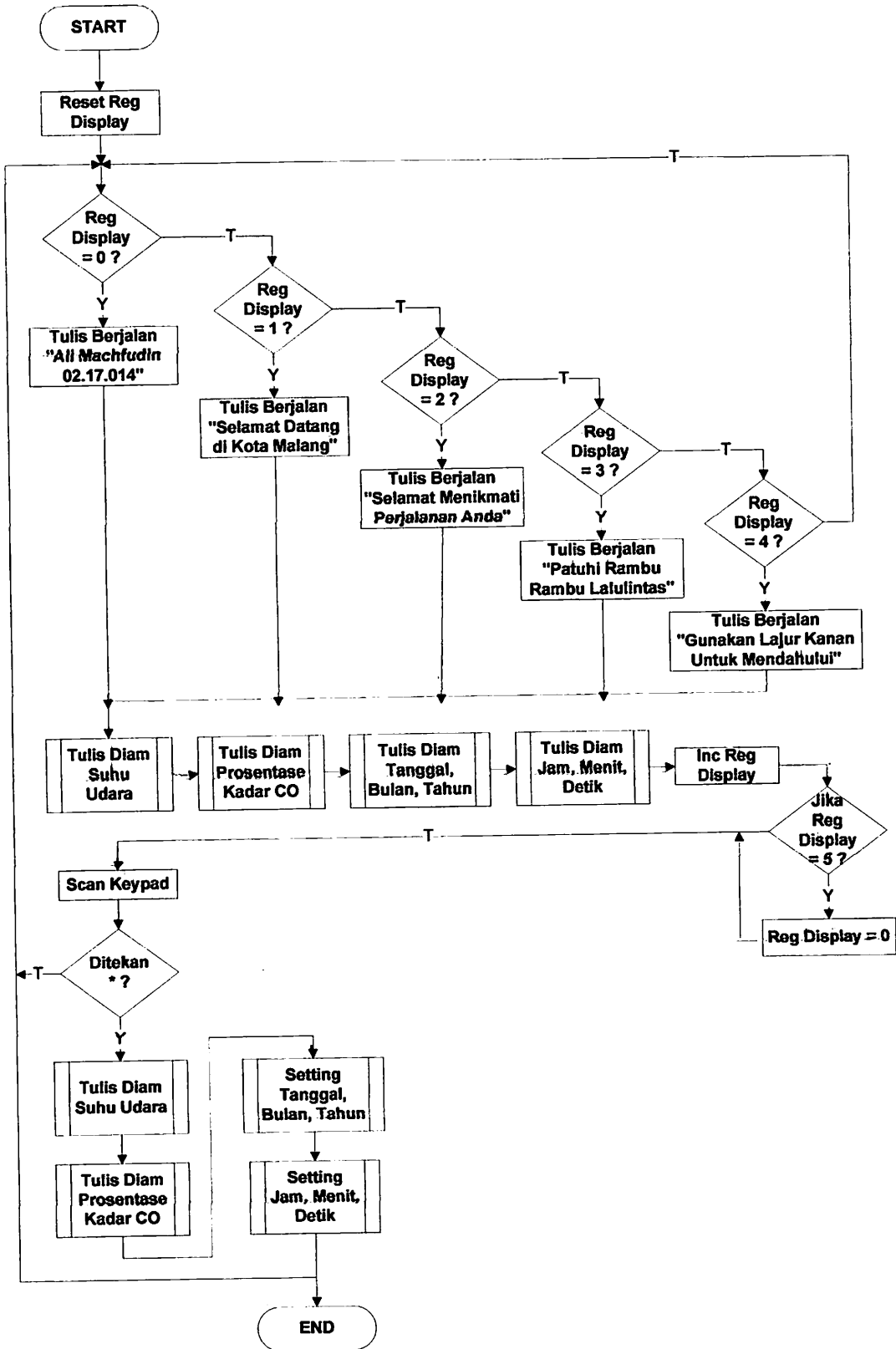
Jadi dari hasil perhitungan diatas diperoleh nilai R_B sebesar 9,04 K, karena dipasaran tidak ada maka digunakan R_b sebesar 10 K. sedangkan untuk nilai R_c diperoleh nilai sebesar 23,6 Ω . Sama seperti nilai R_b karen tidak tersedia dipasaran maka digunakan R_c sebesar 220 Ω .

3.5. Perancangan Perangkat Lunak (*software*)

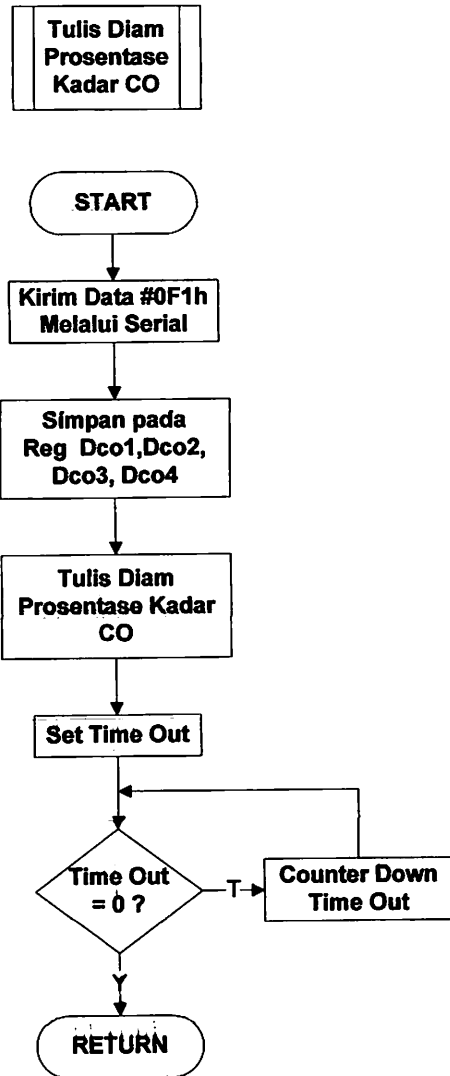
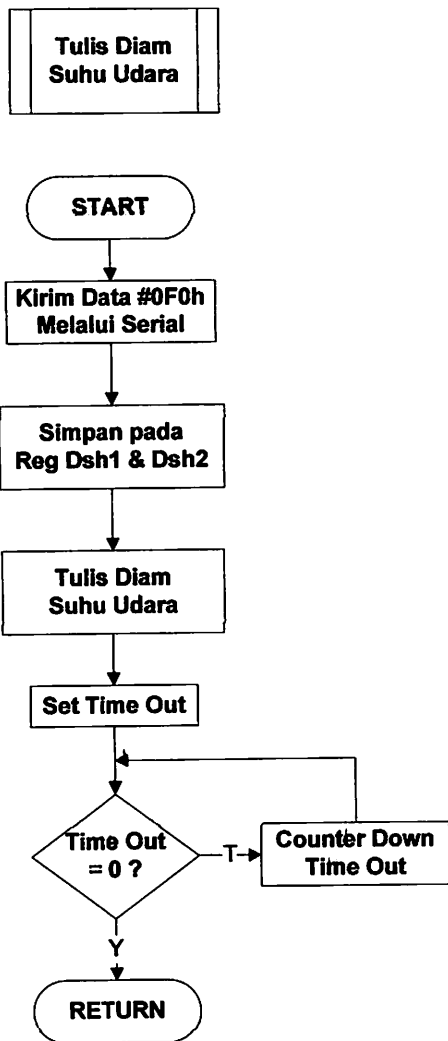
Untuk mendukung *hardware* yang sudah dibuat, maka dibutuhkan perangkat lunak (*software*) supaya perangkat keras tersebut bisa berjalan sesuai dengan tujuan. Mikrokontroler dapat mengendalikan seluruh system apabila ada urutan instruksi yang mendefinisikan secara jelas urutan kerja yang harus dilaksanakan. Dalam perancangan alat ini perangkat lunak yang digunakan adalah bahasa pemrograman *assembler*.

Sebelum membuat perangkat lunak, terlebih dahulu dibuat diagram alir (*flowchart*) dari proses yang akan dibuat supaya memudahkan dalam pembuatan perangkat lunak (*software*).

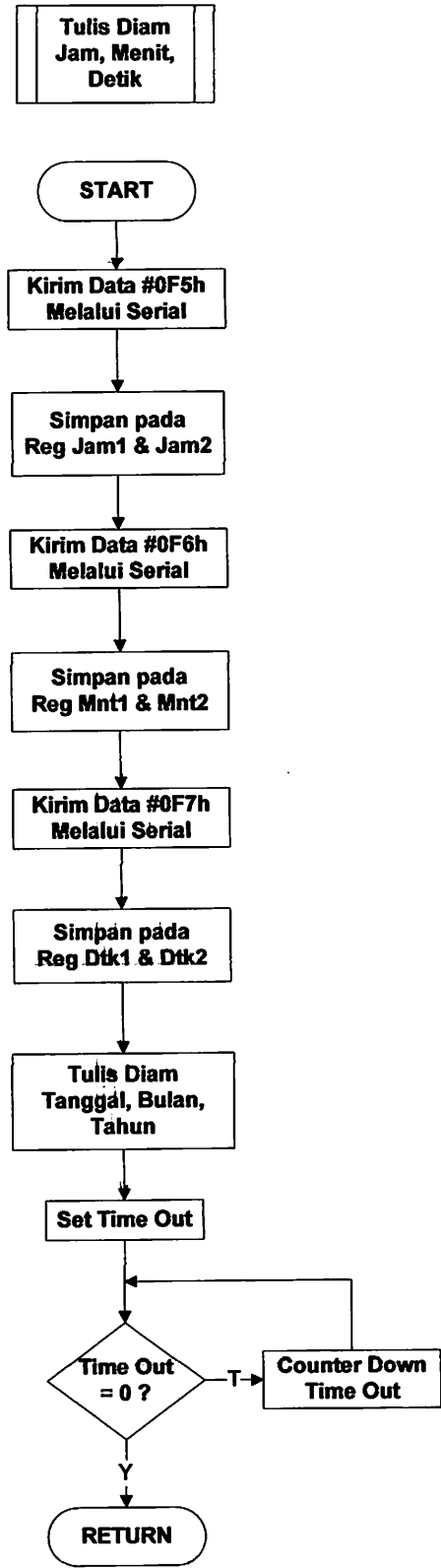
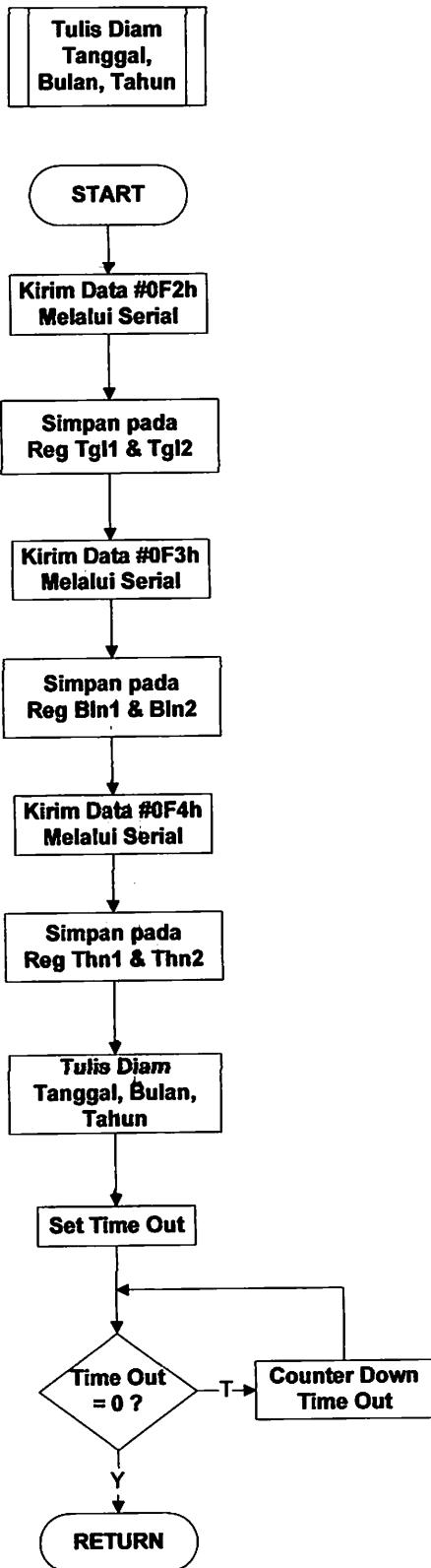
3.5.1. flowchart Program Utama



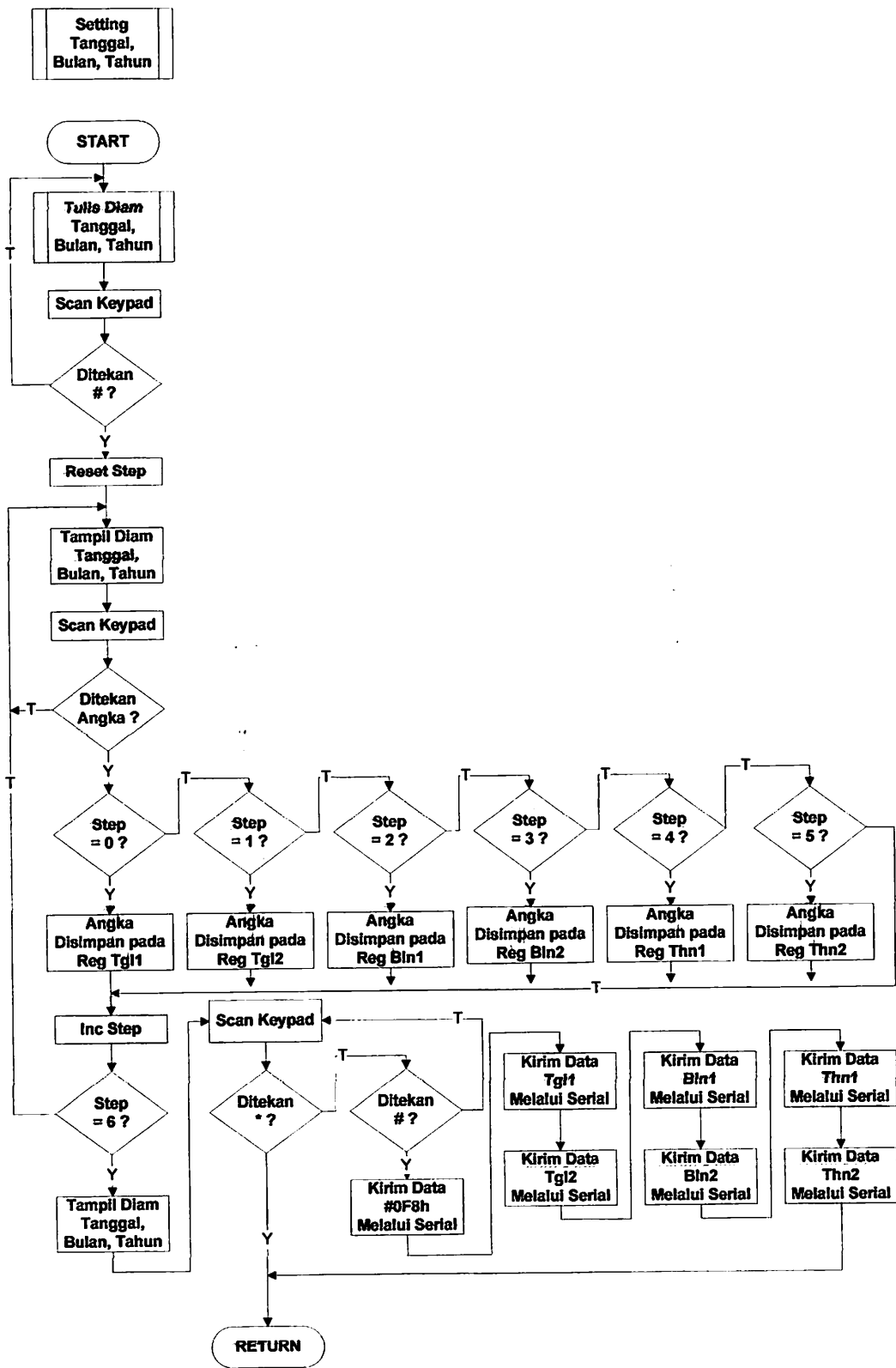
3.5.2. Flowchart Untuk Menampilkan Suhu dan Kadar CO



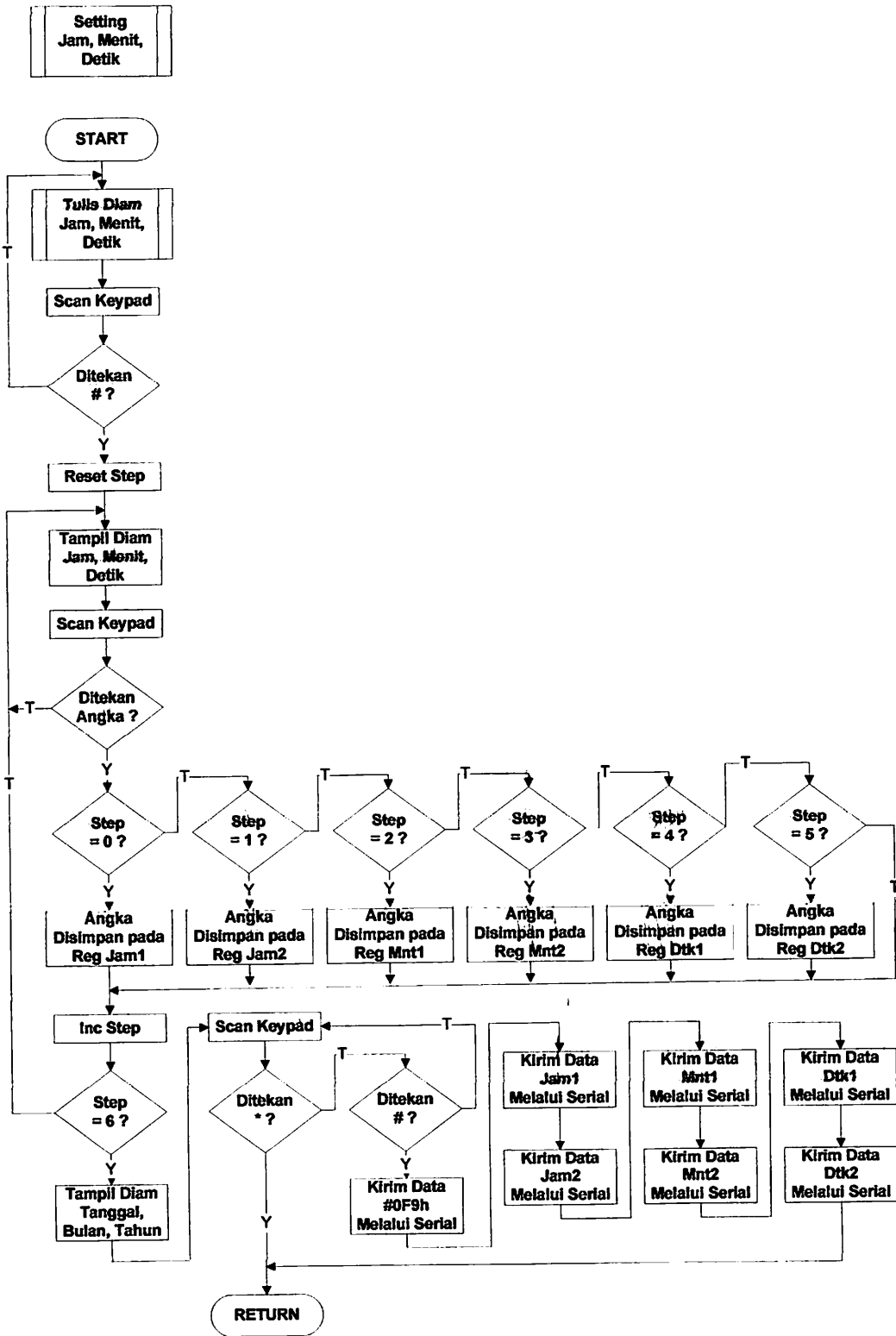
3.5.3. Flowchart Untuk Mengambil Data dan Menampilkan Waktu



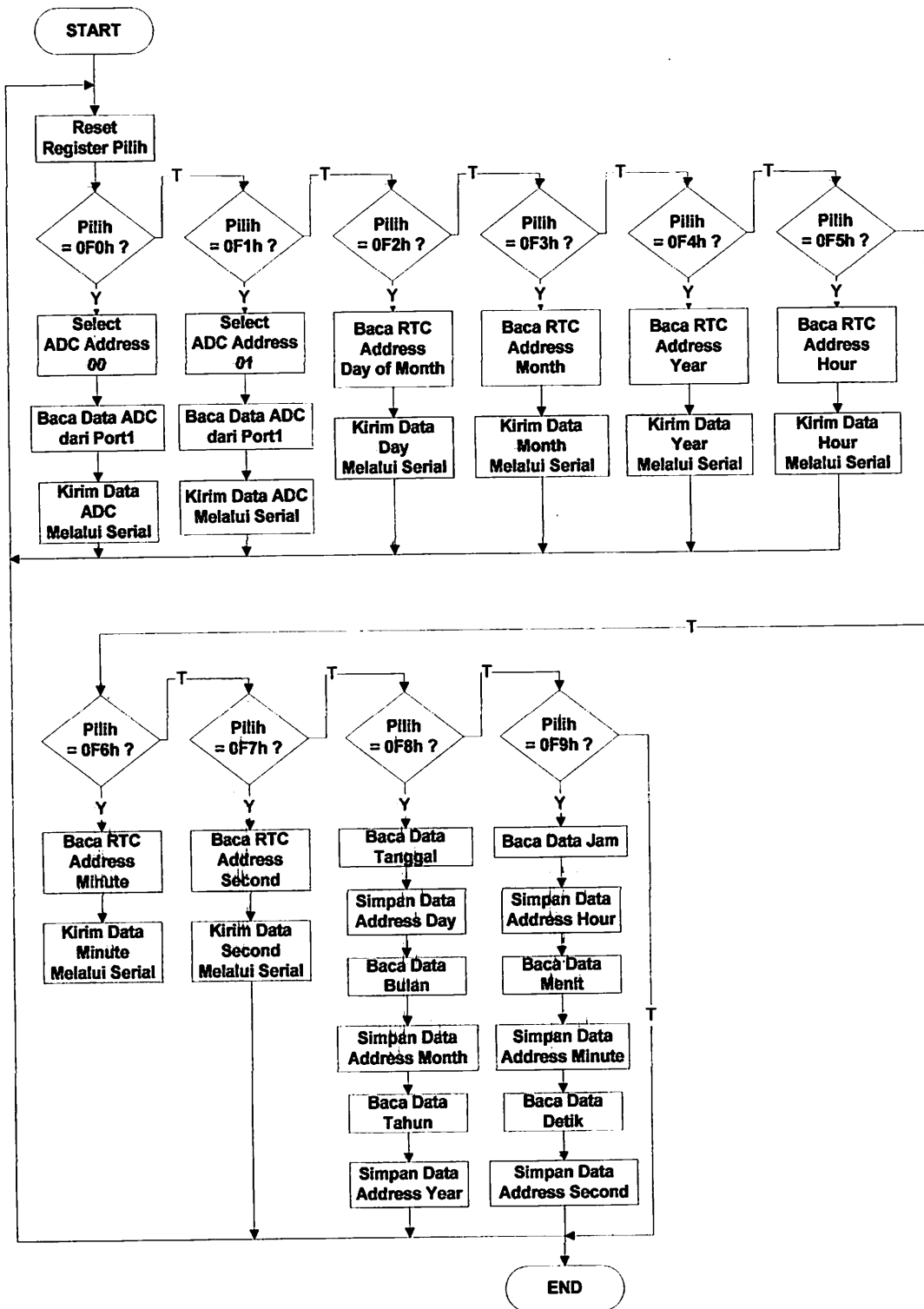
3.5.4. Flowchart Untuk Setting Tanggal, Bulan dan Tahun



3.5.5. Flowchart Untuk Setting Jam



3.5.6. Flowchart Untuk Mengambil dan Mengirim Data ADC dan RTC



BAB IV

PENGUJIAN DAN PERCOBAAN ALAT

4.1. Pendahuluan

Dalam bab ini akan dibahas tentang pengujian alat yang telah dirancang dan dibuat secara keseluruhan. Adapun tujuan dari pengujian alat adalah untuk mengetahui apakah alat yang sudah direncanakan dan dibuat dapat bekerja dengan baik sesuai dengan perencanaan. Pengujian alat ini dengan pengambilan data secara langsung melalui pengujian respon alat terhadap gas karbon monoksida (CO) dan suhu.

4.2. Pengujian Sensor Suhu

Pada pengujian sensor suhu dilakukan pada suhu ruangan dengan menggunakan termometer air raksa. Tujuan dilakukan percobaan ini adalah untuk mengetahui apakah besarnya keluaran tegangan dari sensor suhu sesuai dengan tegangan keluaran sensor suhu yaitu $10 \text{ mV}/^\circ\text{C}$. Untuk menaikkan resolusi pembacaan data suhu, maka output dari sensor suhu diumpankan ke suatu rangkaian penguat *non-inverting*. Pada Pengujian sensor suhu ada beberapa pengukuran yang harus dilakukan antara lain sebagai berikut :

4.2.1. Pengukuran Untuk Mencari Besarnya Penguatan Vout Sensor Suhu

Untuk mengetahui besarnya penguatan dari penguat *non-inverting* dapat dicari dengan membandingkan antara jumlah Vout Op-Amp dan Vout sensor.

Berikut ini persamaan untuk menghitung penguatan rata-rata dari output sensor beserta hasilnya :

$$A_{rata-rata} = \frac{\sum V_{out}}{\sum V_{in}} = \frac{\sum V_{op-Amp}}{\sum V_{out_sensor}}$$

Tabel 4.1 Pengukuran Vout Sensor dan Vout Op-amp

| Temperature (°C) | Vout Sensor (mV) | Vout Op-Amp (mV) |
|--------------------------------|------------------|------------------|
| 25 | 26,0 | 48,10 |
| 30 | 31,0 | 57,35 |
| 35 | 36,0 | 66,60 |
| 40 | 41,0 | 74,00 |
| 45 | 46,0 | 85,10 |
| 50 | 51,0 | 94,35 |
| Jumlah | 231 | 425,5 |
| Penguatan (A) rata-rata | 1,84 Kali | |

4.2.2. Pengujian Untuk Mencari Kesalahan Rata-rata Dari Vout Sensor

Pada pengujian ini untuk mencari kesalahan Vout sensor yaitu dengan menggunakan persamaan sebagai berikut :

$$\Delta t = |Vout_Op - Amp(pengukuran) - Vout_Op - Amp(perhitungan)|$$

$$\Delta\% = \left| \frac{Vout_sensor(pengukuran) - Vout_sensor(datasheet)}{Vout_sensor(datasheet)} \times 100\% \right|$$

$$Kesalahan_rata - rata = \left| \frac{\sum \Delta\%}{\sum pengukuran} \right|$$

Tabel 4.2 Perbandingan Vout Sensor Data Sheet Dengan Vout Sensor

Pengukuran

| Temperatur (°C) | Vout Sensor (mV) | | Δt | Δ% |
|--------------------------------------|------------------|------------|-------|------|
| | Data Sheet | Pengukuran | | |
| 25 | 25 | 26,0 | 1 | 4 |
| 30 | 30 | 31,0 | 1 | 3,33 |
| 35 | 35 | 36,0 | 1 | 2,85 |
| 40 | 40 | 41,0 | 1 | 2,5 |
| 45 | 45 | 46,0 | 1 | 2,22 |
| 50 | 50 | 51,0 | 1 | 2 |
| Kesalahan Rata-rata (<i>error</i>) | | | 0,81% | |

4.2.3. Pengujian Untuk Mencari Kesalahan Rata-rata Vout Op-Amp

Pada pengujian ini untuk mencari kesalahan Vout Op-Amp yaitu dengan persamaan sebagai berikut :

$$\Delta t = |V_{out_Op - Amp}(pengukuran) - V_{out_Op - Amp}(perhitungan)|$$

$$\Delta \% = \left| \frac{V_{out_Op - Amp}(pengukuran) - V_{out_Op - Amp}(perhitungan)}{V_{out_Op - Amp}(perhitungan)} \times 100\% \right|$$

$$Kesalahan_rata - rata = \left| \frac{\sum \Delta \%}{\sum pengukuran} \right|$$

Tabel 4.3 Perbandingan Vout Op-Amp Perhitungan Dengan Vout Op-Amp Pengukuran

| Temperatur (°C) | Vout Op-Amp (mV) | | Δt | Δ% |
|--------------------------------------|------------------|------------|--------|------|
| | Perhitungan | Pengukuran | | |
| 25 | 50 | 48,10 | 1,90 | 3,8 |
| 30 | 60 | 57,35 | 2,65 | 4,41 |
| 35 | 70 | 66,60 | 3,40 | 4,85 |
| 40 | 80 | 74,00 | 6,00 | 7,50 |
| 45 | 90 | 85,10 | 4,90 | 5,44 |
| 50 | 100 | 94,35 | 5,65 | 5,65 |
| Kesalahan Rata-rata (<i>error</i>) | | | 5,27 % | |

4.2.4. Pengujian Pembacaan Suhu Pada Tampilan (*Display*)

Pada pengujian ini untuk mencari kesalahan pembacaan suhu pada termometer dengan pembacaan pada dot matrik dengan menggunakan persamaan sebagai berikut :

$$\Delta t = |Pembacaan(dot_matrik) - Pembacaan(termometer)|$$

$$\Delta \% = \left| \frac{Pembacaan(dot_matrik) - Pembacaan(termometer)}{Pembacaan(termometer)} \times 100\% \right|$$

$$Kesalahan_rata - rata = \left| \frac{\sum \Delta \%}{\sum Pembacaan} \right|$$

Tabel 4.4 Pembacaan Suhu Termometer Dengan *Display* Dot Matrik

| Temperatur (°C) | Pembacaan | | Δt | $\Delta\%$ |
|--------------------------------------|------------|------------|------------|------------|
| | Termometer | Dot Matrik | | |
| 25 | 25 | 25 | 0 | 0 |
| 30 | 30 | 30 | 0 | 0 |
| 35 | 35 | 36 | 1 | 2.85 |
| 40 | 40 | 41 | 1 | 2.50 |
| 45 | 45 | 46 | 1 | 2.22 |
| 50 | 50 | 52 | 2 | 4 |
| Kesalahan Rata-rata (<i>error</i>) | | | 1,928 % | |

4.3. Pengujian Sensor CO

Pada pengujian sensor CO dilakukan dengan cara mendekatkan sensor dengan kenalpot kendaraan dalam keadaan mesin hidup. Untuk proses pengkalibrasian dilakukan dengan menyamakan hasil pengujian antara alat yang dibuat dengan alat uji emisi. Setelah proses pengkalibrasian selesai antara kedua alat tersebut di bandingkan apakah sesuai atau ada selisih antara kedua alat tersebut. Berikut adalah tabel pengukuran V_{out} sensor CO :

Tabel 4.5 Pengukuran V_{out} Sensor CO

| Kadar CO (%) | V_{out} Sensor CO (V) |
|--------------|-------------------------|
| 0,10 | 0,36 |
| 0,11 | 0,40 |
| 0,14 | 0,51 |
| 0,19 | 0,69 |
| 0,24 | 0,88 |

Tabel 4.6 Pembacaan Alat Ukur CO Dengan Disply Dot Matrik

| RPM Motor | Pembacaan | | Δt | $\Delta \%$ |
|--------------------------------------|----------------|------------|------------|-------------|
| | Alat Uji Emisi | Dot Matrik | | |
| 750 | 0,22 | 0,20 | 0,02 | 9 |
| 1500 | 0,28 | 0,30 | 0,02 | 9 |
| 2500 | 0,54 | 0,55 | 0,01 | 1.85 |
| Kesalahan Rata-rata (<i>error</i>) | | | 6,61 % | |

4.4. Pengujian ADC

Pengujian terhadap ADC0808 ini dapat dilakukan dengan menggunakan LED. Input pengujian ADC dibuat dari pencatu daya tegangan DC dari 0 – 5 volt, kemudian potensiometer kita putar berlahan-lahan, kita sesuaikan nilainya pada tegangan tertentu. LED sebagai indikator output dari ADC akan menyala sesuai dengan numerik dari 0 sampai dengan 255 dalam decimal.

Untuk hasil perhitungan dan pengukuran (pembacaan) ADC0808 adalah sebagai berikut :

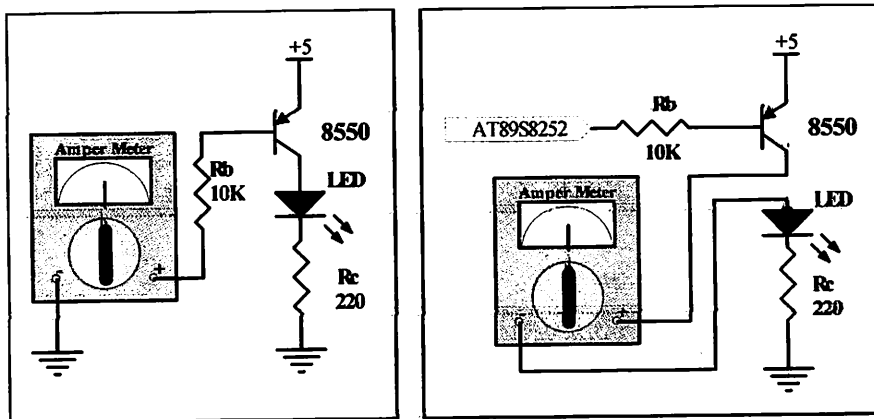
Tabel 4.7 Perbandingan Perhitungan dan Pembacaan ADC0808

| Tegangan Input ADC (V) | Output ADC (Perhitungan) | | Output ADC (Pembacaan) |
|---------------------------|--------------------------|----------|---------------------------|
| | Desimal | Biner | Biner |
| 0,5 | 25 | 00011001 | 00011001 |
| 1,0 | 51 | 00110011 | 00110011 |
| 1,5 | 76 | 01001100 | 01001100 |
| 2,0 | 102 | 01100110 | 01100110 |
| 2,5 | 127 | 01111111 | 01111111 |
| 3,0 | 153 | 10011001 | 10011001 |

| | | | |
|-----|-----|----------|----------|
| 3,5 | 178 | 10110010 | 10110010 |
| 4,0 | 204 | 11001100 | 11001100 |
| 4,5 | 229 | 11100101 | 11100101 |
| 5,0 | 255 | 11111111 | 11111111 |

4.5. Pengujian Dot Matrik

Dalam pengujian ini menggunakan rangkaian yang terdiri dari transistor 8550, resistor, LED dan sebuah multimeter. Dalam hal ini besaran yang diukur adalah besarnya arus basis (I_B) dan arus kolektor (I_c), seperti yang terlihat pada gambar dibawah ini :



Gambar 4.1 Pengukuran Arus Basis (I_B) dan Arus Kolektor (I_c)

Pada pengukuran diatas didapatkan hasil sebagai berikut :

Arus basis (I_B) = 0,44 mA

Arus kolektor (I_c) = 13,5 mA

4.6. Pengujian RTC

Pada pengujian ini bertujuan untuk membandingkan tampilan waktu (jam, menit, detik) dengan alat sebenarnya dalam hal ini yang digunakan adalah jam digital. Dalam pengujian, waktu pengujian yaitu selama 24 jam sehingga dapat

diperoleh data dari hasil pengujian tersebut. Hasil yang diperoleh dari pengujian tersebut adalah sebagai berikut :

Tabel 4.8 Pengujian Waktu Jam Digital Dan Display Dot Matrik

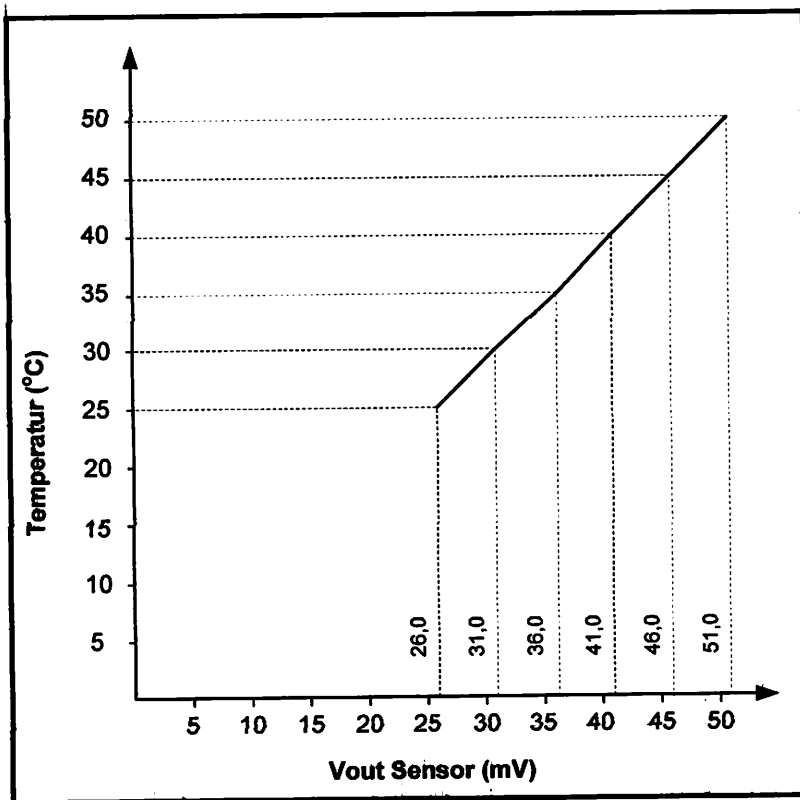
| Waktu Pada Jam Digital | Dot Matrik |
|-------------------------------|-------------------|
| 18 : 00 : 00 | 18 : 00 : 00 |
| 22 : 00 : 00 | 22 : 00 : 00 |
| 24 : 00 : 00 | 24 : 00 : 00 |
| 02 : 00 : 00 | 02 : 00 : 00 |
| 05 : 00 : 00 | 05 : 00 : 00 |
| 07 : 00 : 00 | 07 : 00 : 00 |
| 10 : 00 : 00 | 10 : 00 : 00 |
| 13 : 00 : 00 | 13 : 00 : 00 |
| 15 : 00 : 00 | 15 : 00 : 00 |
| 18 : 00 : 00 | 18 : 00 : 00 |

Jadi dari data hasil pengujian diatas, dapat disimpulkan bahwa perbandingan antara waktu pada jam digital dan waktu pada alat yang dibuat selama 24 jam, sudah sesuai atau tidak ditemukan *error* (selisih).

4.7. Pengujian Keseluruhan

Setelah pengujian alat setiap blok selesai dan sudah sesuai dengan perencanaan, maka untuk selanjutnya akan dilakukan pengujian sistem secara keseluruhan. Pengujian ini dimaksudkan untuk mengetahui apakah alat yang dibuat sudah sesuai dengan perencanaan atau belum.

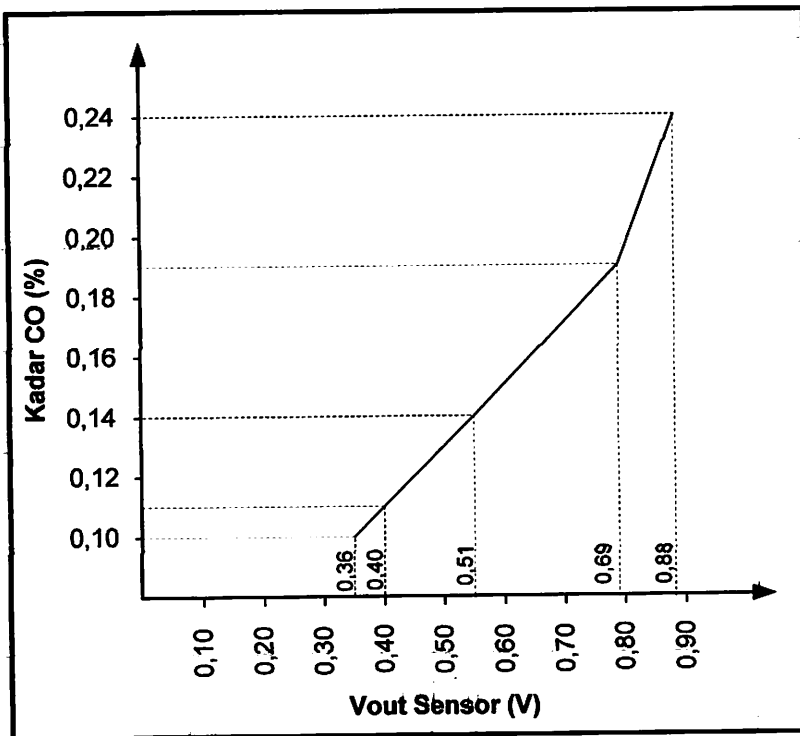
Hasil dari pengujian secara keseluruhan dapat dilihat pada grafik perbandingan antara temperatur terhadap tegangan output sensor dan grafik perbandingan kadar karbon monoksida terhadap tegangan out sensor.



Grafik 4.1 Perbandingan Temperatur Terhadap Vout Sensor Suhu

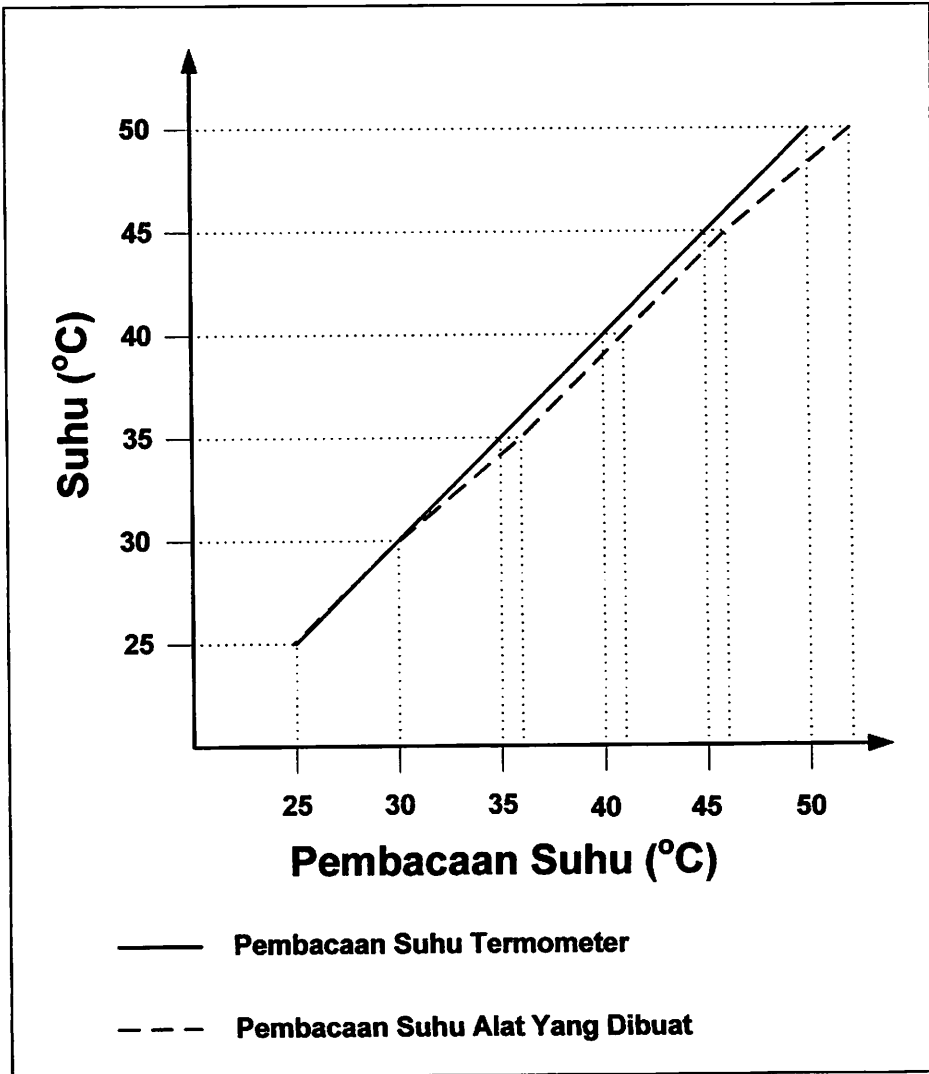
Berdasarkan grafik perbandingan temperatur terhadap tegangan output sensor dapat ditarik kesimpulan bahwa setiap kenaikan suhu sebesar 1°C , akan mempengaruhi tegangan output dari sensor. Dari hasil pengujian yang telah dilakukan, setiap kenaikan suhu sebesar 1°C , tegangan output sensor akan berubah atau naik sebesar 11°C . Berdasarkan hasil pengujian tersebut, nilai tegangan output sensor memiliki sesilsh atau perbedaan antara pengukuran dan teori. Berdasarkan teori, tegangan output sensor akan mengalami kenaikan sebesar

10 mV setiap kenaikan suhu sebesar 1 °C. Jadi selisih antara teori dan pengukuran pada tegangan output sensor suhu adalah 1 mV.



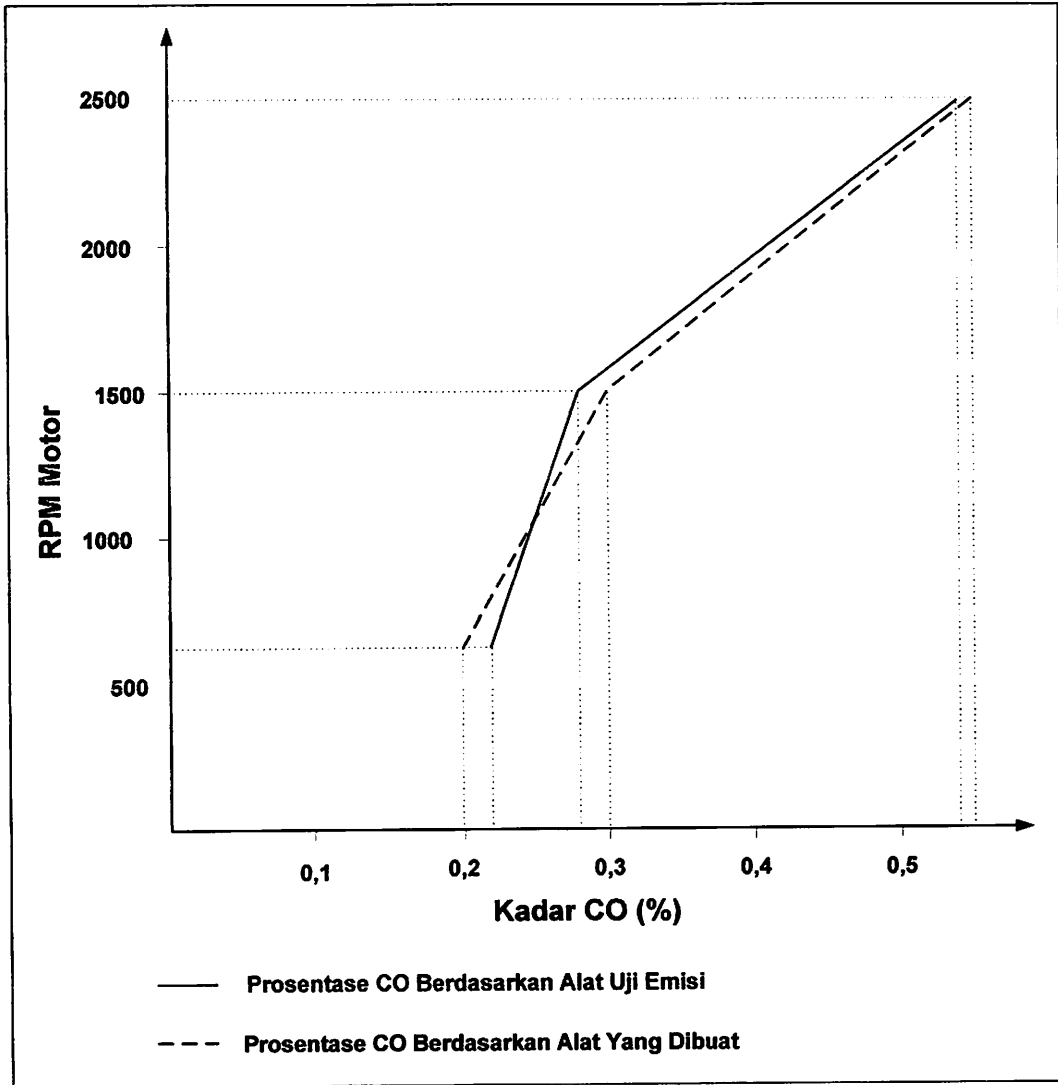
Grafik 4.2 Perbandingan Kadar CO Terhadap Vout Sensor CO

Pada grafik perbandingan kadar karbon monoksida (CO) terhadap Vout sensor dapat disimpulkan bahwa setiap kenaikan kadar CO akan mempengaruhi tegangan output dari sensor (Vout sensor). Dari hasil pengujian yang telah dilakukan, setiap kenaikan kadar CO sebesar 0,01 %, tegangan output sensor akan berubah atau naik sebesar 0,04 V.



Grafik 4.3 Perbandingan Pembacaan Suhu Termometer Dengan Alat Yang Dibuat

Berdasarkan Garafik diatas dapat diketahui selisih antara pembacaan suhu pada termometer dengan pembacaan suhu pada alat yang dibuat. Selisih (error) dari pembacaan suhu tersebut dapat dilihat pada tabel 4.4, yaitu sebesar 1,928 %.



Grafik 4.4 Pembacaan Alat Ukur CO Dengan Alat yang Dibuat

Pada garfik perbandingan prosentase kadar karbon monoksida (CO) antara alat uji emisi dengan alat yang dibuat diatas dapat ditarik kesimpulan yaitu berdasarkan juga dari hasil pengujian pada tabel 4.6, yaitu terdapat selisih (error) sebesar 6,61 %.

BAB V

PENUTUP

Pada bab ini penulis akan menyampaikan kesimpulan dan saran setelah melakukan pengujian pada alat ukur suhu dan kadar karbon monoksida (CO) yang di jalan raya yang dilengkapi tampilan waktu dan informasi.

5.1 Kesimpulan

Setelah melakukan perencanaan, pembuatan, dan pengujian pada alat ukur suhu dan kadar karbon monoksida (CO) yang di jalan raya yang dilengkapi dengan tampilan waktu dan informasi, dapat diambil beberapa kesimpulan sebagai berikut :

1. Pada proses pengkalibrasian diperoleh penguatan sebesar 2 kali itu terbukti dari hasil pengukuran *output* sensor dibandingkan dengan *output* Op-Amp diperoleh nilai penguatan (A) sebesar 1,984 kali. setiap kenaikan suhu sebesar 1°C, tegangan *output* sensor akan berubah atau naik sebesar 11°C.
2. Proses pengkalibrasian CO dilakukan dengan cara membandingkan dengan alat uji emisi kendaraan dengan alat ukur yang dibuat. Dari hasil pengujian yang telah dilakukan, setiap kenaikan kadar CO sebesar 0,01 %, tegangan output sensor akan berubah atau naik sebesar 0,04 V. Sedangkan dari hasil perbandingan alat ukur CO dengan *display* dot matrik diperoleh kesalahan (*error*) rata-rata sebesar 6,61 %.
3. Pada ADC0808 tegangan referensi di-*set* pada kondisi *full range* yaitu $V_{ref(-)}$ pada 0 volt sedangkan $V_{ref(+)}$ pada 5 volt. Dari pengesetan tersebut

nilai V_{ref} adalah 5 volt sehingga diperoleh resolusi per bitnya adalah sebesar 0,0196 mV. Dari perbandingan perhitungan dan pengukuran output ADC0808 ternyata tidak ditemukan kesalahan (*error*) atau dapat dikatakan sudah sesuai.

4. Untuk display digunakan dot matrik 5 x7 sebanyak 10 buah. Untuk *drive* dari display dot matrik diperlukan sebuah rangkaian driver yang terdiri dari IC SN74HC164 dan juga transistor 8550. Dari hasil perhitungan pada perancangan arus basis (I_B) sebesar 0,42 mA dan arus kolektor (I_c) sebesar 80 mA. sedangkan dari hasil pengukuran I_B sebesar 0,44 mA dan untuk I_c sebesar 13,5 mA.
5. Untuk menampilkan waktu digunakan sebuah RTC (*Real Time Clock*). Dari RTC data waktu masuk ke mikrokontroler kemudian diproses sehingga dapat ditampilkan pada display dot matrik. Dari hasil pengujian , yaitu perbandingan antara display waktu pada alat dengan jam digital diperoleh selama 24 jam didapatkan hasil yang sudah sesuai atau tidak ada *error* (selisih).
6. untuk suhu, dapat dikalibrasi langsung pada penguatnya sedangkan untuk CO tidak bisa dikalibrasi secara langsung karena pengkalibrasian dilakukan pada *software*. Kemudian untuk waktu (Jam dan tanggal) dapat di kalibrasi atau diset dengan menggunakan keypad.

5.2 Saran

1. Pada saat melakukan pengkalibrasian, untuk mencari alat ukur CO di Indonesia sangat kesulitan sehingga untuk memperoleh data secara langsung dari alat ukur sebenarnya sangat sulit misalnya adapun biayanya sangat mahal. Alat yang sudah ada di jalan raya hanya menampilkan kisaran kadar CO saja, jadi kita kesulitan mengetahui tepatnya berapa.
2. Karena bentuknya terlalu kecil dan jumlah *display* hanya 10 karakter maka untuk pengembangan lebih lanjut sebaiknya digunakan *display* yang lebih besar sehingga pada saat dipasang di jalan raya dapat dilihat dengan baik oleh masyarakat yang melintasi jalan tersebut.
3. Pada tampilan suhu pada dot matrik, yang hanya menampilkan 2 karakter saja, sebaiknya untuk memperoleh hasil tepat dan sesuai dengan pembacaan sensor sebaiknya pada *display* jumlah karakter yang dipakai harus ditambah minimal 4 karakter.
4. Untuk peletakan sensor sebaiknya diletakkan dekat dengan media yang diukur atau dengan kata lain tidak jauh dari sumber gas dari asap kendaraan bermotor.
5. Pada alat ini hanya menampilkan keadaan suhu dan kadar CO dalam bentuk angka sesuai pembacaan sensor, tidak menampilkan keterangan dari hasil pengukuran seperti kondisi baik, sedang dan jelek.
6. Pada alat ini terutama untuk alat ukur CO tidak dilengkapi *device* untuk pengkalibrasi secara langsung, untuk pengembangannya sebaiknya dilengkapi dengan *device* untuk pengkalibrasian secara langsung.

Daftar Pustaka

- [1] BPS-Statistik Lingkungan Hidup Indonesia. Tahun 2001.
- [2] D.E, Daniel. 2002. Perancangan Dan Pembuatan Alat Pengontrol Ruang Ber-AC. Hal.32.
- [3] Scimarec. Thick Film Gas Sensor AF-Series. Data Sheet AF-56. Hal.1.
- [4] National Semiconductor. December 1994. LM-35 Data Sheet. Hal.1.
- [5], [6], [7], [8], [9] Chairizal, Ricardo.2003. Pencatat Konsumsi Daya Listrik Jarak Jauh Berbasis Mikrokontroler AT89C51. Hal11-23. Malang : Universita Negeri Malang
- [10], [11], [12] Sundyatmoko, Uun. Maret 2006. Perancangan dan Pembuatan *Controlling* Miniatur Mobil Rekreasi Yang Berjalan Otomatis Diaplikasikan Pada Tempat Wisata Alam Bagi Keluarga. Hal 22-34. Malang : ITN Malang.
- [13] Dallas Semiconductor. 2000. DS12C887 Data Sheet. Hal.7.
- [14]
- [15] National Semiconductor. November 1995. ADC0808 Data Sheet. Hal.1.
- [16]
- [17], [18] Fairchild Semiconductor.January 2003. 74HC164 Data Sheet
- [19]
- [20] Nalwan, Andi Paulus.2003. *Teknik Antar Muka dan Pemrograman*. Jakarta : Gramedia.
- [21] National Semiconductor. November 1995. ADC0808 Data Sheet. Hal.5.

LAMPIRAN



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

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

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

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Nomor : ITN - 618 /III.TA-2/7/07
Lampiran : -
Perihal : Survey

Malang, 18 Januari 2007

Kepada : Yth. Pimpinan
PPPGT/VEDC MALANG
Jl. Teluk Mandar
Di - Arjosari - Malang

Bersama ini dengan hormat kami mohon kebijaksanaan Saudara agar Mahasiswa kami dari Fakultas Teknologi Industri Jurusan Teknik Elektro S-1 Konsentrasi Teknik Elektronika dapat diijinkan untuk melaksanakan Survey pada PPPGT/VEDC yang saudara pimpin untuk mendapatkan data-data guna penyusunan Skripsi dengan judul : Perancangan dan Pembuatan Alat Ukur Suhu dan Kadar Karbon Monoksida (CO) Di Jalan Raya Yang Dilengkapi Tampilan Waktu dan Informasi

Mahasiswa tersebut adalah :

Ali Machfudin Nim. 02.17.014

Adapun lamanya survey adalah : 1 Hari

Demikian agar maklum dan atas perhatian serta bantuannya kami ucapkan terima kasih.

DEKAN
Fakultas Teknologi Industri

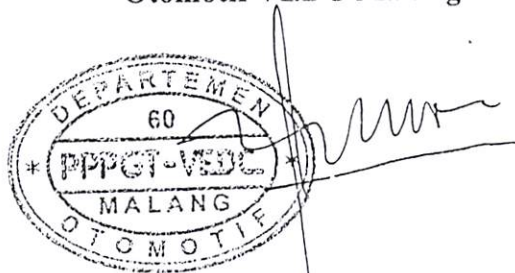
DEKAN Ir. Mochtar Asroni, MSME
Nip.Y. 1018100036

**Perbandingan Pembacaan Alat Uji Emisi Dengan Pembacaan
Pada Display Dot Matrik Alat Ukur CO**

| No | RPM Motor | Alat Uji Emisi | Display Dot Matrik |
|----|-----------|----------------|--------------------|
| 1 | 750 | 0,22 | 0,20 |
| 2 | 1500 | 0,28 | 0,30 |
| 3 | 2500 | 0,54 | 0,55 |

Pengukuran dilakukan di Laboratorium Mesin VEDC Malang

**Mengetahui Pimpinan Departemen
Otomotif VEDC Malang**



(Sukma Tjatroa)



FORMULIR BIMBINGAN SKRIPSI

Nama : ALI MACHFUDIN
NIM : 02 17 014
Masa Bimbingan : 9 Juni 2006 - 9 Desember 2006
Judul Skripsi : PERANCANGAN DAN PEMBUATAN ALAT UKUR SUHU
DAN KADAR KARBON MONOKSIDA (CO) DI JALAN RAYA
YANG DILENGKAPI TAMPILAN WAKTU DAN INFORMASI

| No | Tanggal | Uraian | Paraf Pembimbing |
|----|----------------|---------------------------------|------------------|
| 1 | 29/2006 /12 | Bab I, II, III perbaikan refer. | |
| 2 | 12/2007 /12 | | |
| 3 | 15/2007 /1 | Bab IV & Bab V | |
| 4 | 30/2007 /1 | Bab IV & V | |
| 5 | 2/2007 /2 | Bab V + Seminar | |
| 6 | 27/2007 /2 | Seminar | |
| 7 | 28/2007 /2 | Demo | |
| 8 | | | |
| 9 | | | |
| 10 | | | |

Malang,
Dosen Pembimbing I

Ir. F. Yudi Limpraptono, MT
NIP : 103 950 0274



FORMULIR BIMBINGAN SKRIPSI

Nama : ALI MACHFUDIN
NIM : 02 17 014
Masa Bimbingan : 9 Juni 2006 - 9 Desember 2006
Judul Skripsi : PERANCANGAN DAN PEMBUATAN ALAT UKUR SUHU
DAN KADAR KARBON MONOKSIDA (CO) DI JALAN RAYA
YANG DILENGKAPI TAMPILAN WAKTU DAN INFORMASI

| No | Tanggal | Uraian | Paraf Pembimbing |
|----|----------|---------------------|------------------|
| 1 | 18/09'06 | Revisi Bab I & II | |
| 2 | 29/09'06 | Diagram blok + III | |
| 3 | 7/02'07 | Bab. III + IV | |
| 4 | 7/02'07 | Analisa system. | |
| 5 | 7/02'07 | Makalah Seminar Acc | |
| 6 | 13/02'07 | Bab III + IV, + V | |
| 7 | 26/02'07 | Acc Kompri | |
| 8 | | | |
| 9 | | | |
| 10 | | | |

Malang,
Dosen Pembimbing II

M. Ashar, ST, MT



LEMBAR BIMBINGAN SKRIPSI

Nama Mahasiswa : Ali Machfudin
NIM : 02.17.014
Jurusan : Teknik Elektro S1
Konsentrasi : Teknik Elektronika
Judul Skripsi : Perancangan dan Pembuatan Alat Ukur Suhu
dan Kadar Karbon Monoksida (CO) Di Jalan
Raya Yang Dilengkapi Tampilan Waktu dan
Informasi
Masa Penulisan Skripsi : 9 Juni 2006 s/d 9 Desember 2006
Dievaluasi Dengan Nilai : 90 (Sembilan Puluh) *f*

Mengetahui,

Ketua Jurusan Teknik Elektro S1

Ir. F. Yudi Limpraptono, M.T.
NIP.Y. 1039500274

Malang, 22 Maret 2007

Dosen Pembimbing I

Ir. F. Yudi Limpraptono, M.T.
NIP.Y. 1039500274



LEMBAR BIMBINGAN SKRIPSI

Nama Mahasiswa : Ali Machfudin
NIM : 02.17.014
Jurusan : Teknik Elektro S1
Konsentrasi : Teknik Elektronika
Judul Skripsi : Perancangan dan Pembuatan Alat Ukur Suhu
dan Kadar Karbon Monoksida (CO) Di Jalan
Raya Yang Dilengkapi Tampilan Waktu dan
Informasi
Masa Penulisan Skripsi : 9 Juni 2006 s/d 9 Desember 2006
Dievaluasi Dengan Nilai : 80 (Delapan Puluh)

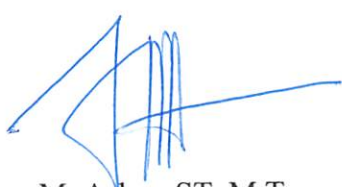
Mengetahui,

Ketua Jurusan Teknik Elektro S1


Ir. F. Yudi Limpraptono, M.T.
NIP.Y. 1039500274

Malang, 22 Maret 2007

Dosen Pembimbing I I


M. Ashar, ST, M.T.



LEMBAR PERBAIKAN SKRIPSI

Nama Mahasiswa : Ali Machfudin
NIM : 02.17.014
Jurusan : Teknik Elektro S1
Konsentrasi : Teknik Elektronika
Hari / Tanggal Ujian Skripsi : Jum'at / 16 Maret 2007

| Materi Perbaikan | Paraf |
|---|-------|
| Grafik hasil pengujian dan hasil perhitungan antara alat standart dengan alat yang dibuat | |

Diperiksa / Disetujui:

PENGUJI II

I Komang Somawirata, ST, MT
NIP.P 1030100361

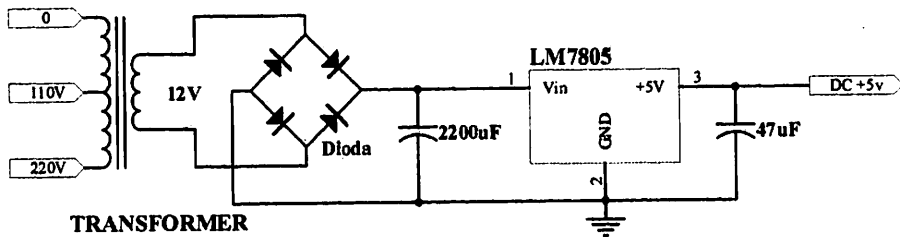
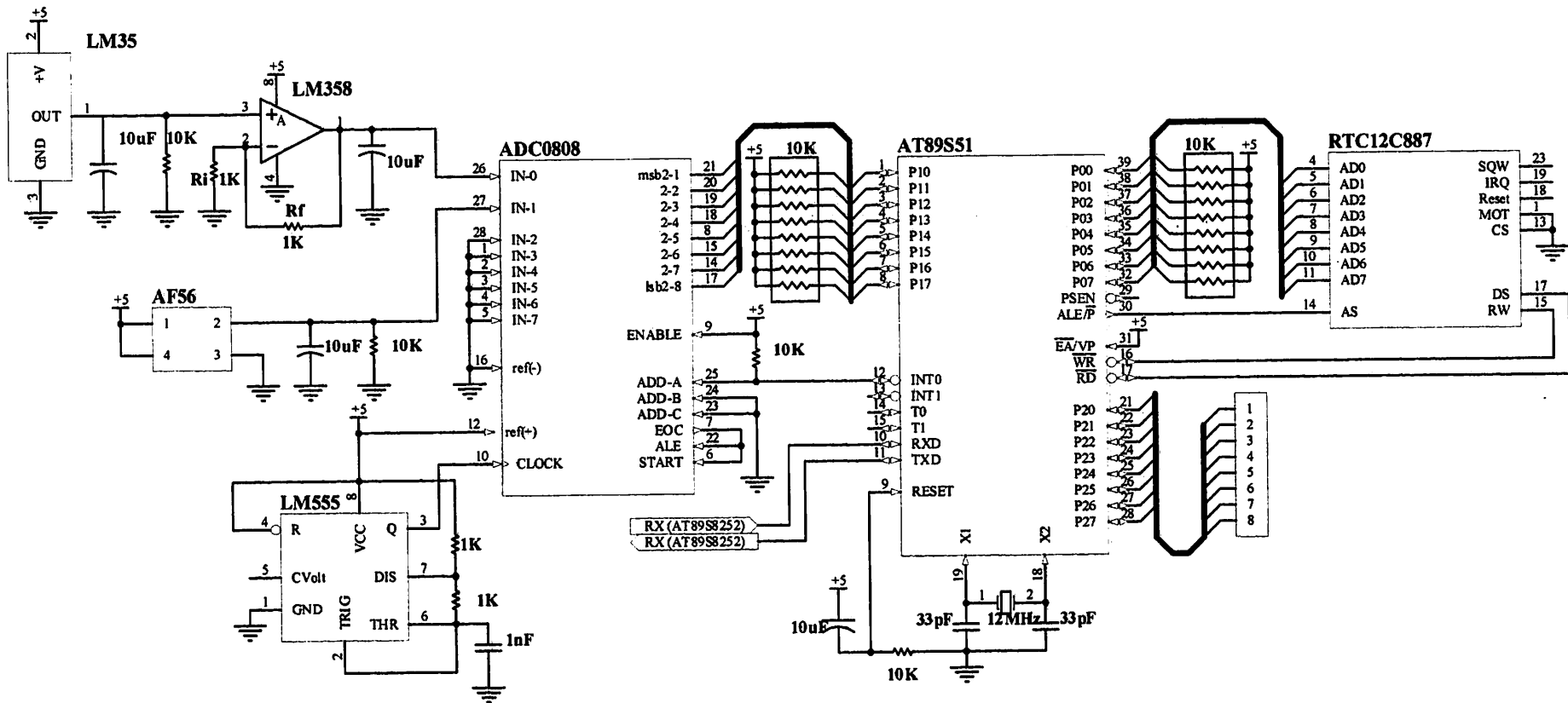
Mengetahui :

Dosen Pembimbing I

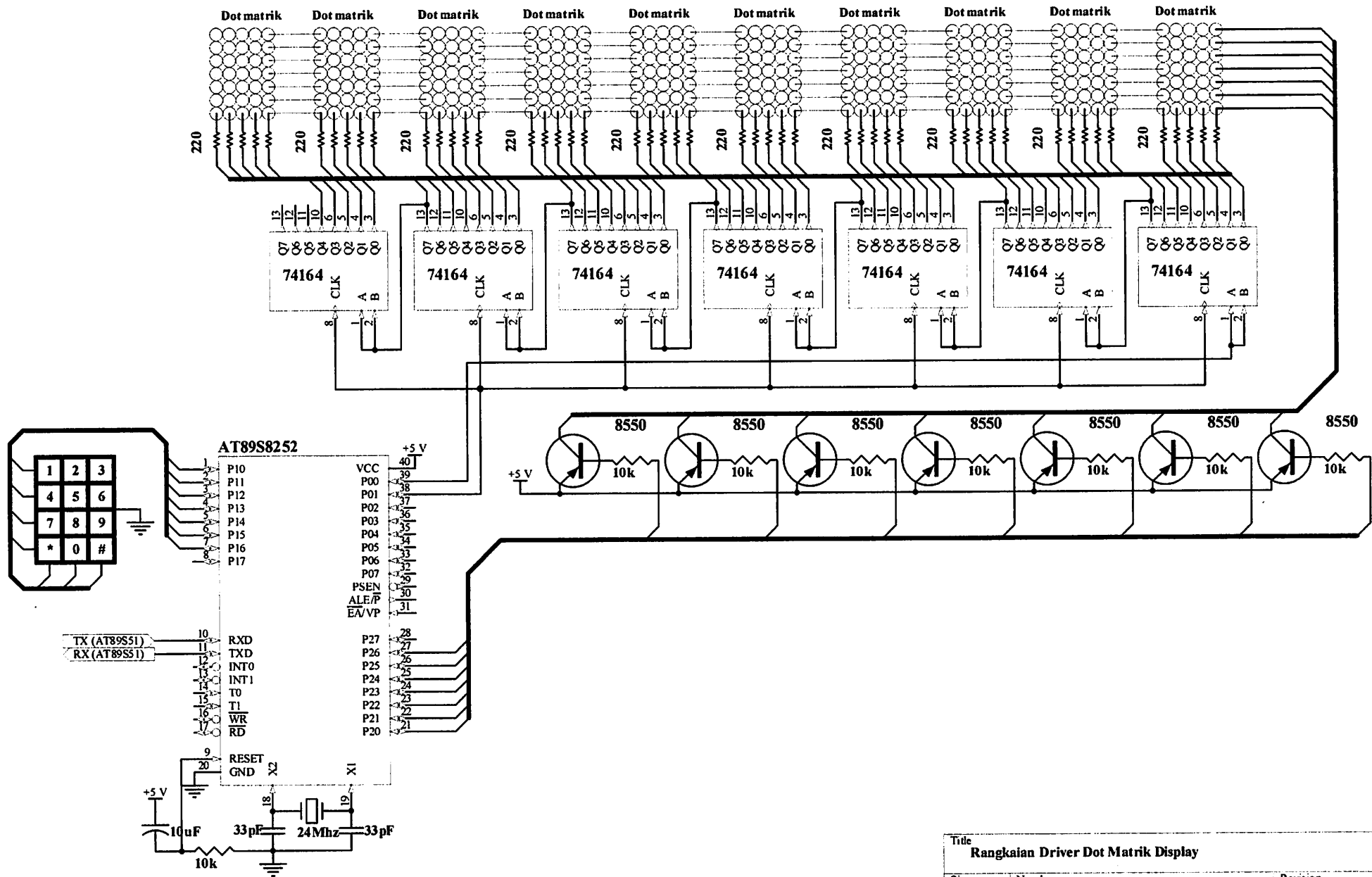
Ir. F Yudi Limpraptono, MT.
NIP.Y 1039500274

Dosen Pembimbing II

M Ashar, ST, MT



| Title | | | |
|---|---|----------|--|
| Rangkaian Utama Alat Ukur Suhu dan kadar CO | | | |
| Size | Number | Revision | |
| A4 | Ali Machfudin 02.17.014 | | |
| Date: | 1/18/2006 | Sheet of | |
| File: | E:\alexandro\...rangkaian utamaA.SCHDOC Drawn By: Ali Machfudin | | |



| Title | | |
|-------------------------------------|-------------------------------------|-------------------------|
| Rangkaian Driver Dot Matrik Display | | |
| Size | Number | Revision |
| A4 | Ali Machfudin 02.17.014 | |
| Date: | 3/15/2007 | Sheet of |
| File: | E:\ale file\make PCB\Rang D M.SCHDO | Drawn By: Ali Machfudin |


```

1: ;-----;
2: ; Program Assembler Perancangan dan Pembuatan Alat Ukur Suhu Dan Kadar Karbon ;
3: ; Monoksida (CO) Di jalan Raya Yang Diliengkapi Tampilan Waktu dan Informasi ;
4: ;                               Ali Machfudin   02.17.014                               ;
5: ;-----;
6:
7:      org      0h
8:      ljmp     init          ;long jump ke init
9: ;-----;
10: ;      Interrupt Serial          ;
11: ;-----;
12:      org      23h          ; menunjukkan alamat serial interrupt
13:      clr      ES
14:      jnb     RI,$          ; lompat ke dia sendiri jika belum 0
15:      clr      RI          ; jadikan RI = 0
16:      mov     A,SBUF        ; pindahkan data SBUF ke accumulator
17:      setb    ES          ; jadikan ES = 1
18:      reti
19: ;-----;
20: ;      Equate Definition          ;
21: ;-----;
22:      DtAB    Bit P0.0      ; Alamat AB (scan)
23:      Clck    Bit P0.1      ; Alamat clock
24:      Dsh1    Equ 30h       ; Alamat data suhu 1
25:      Dsh2    Equ 31h       ; Alamat data suhu 2
26:      Dco1    Equ 32h       ; Alamat data CO 1
27:      Dco2    Equ 33h       ; Alamat data CO 2
28:      Dco3    Equ 34h       ; Alamat data CO 3
29:      Dco4    Equ 35h       ; Alamat data CO 4
30:      Jam0    Equ 36h       ; Alamat data jam 1
31:      Jam1    Equ 37h       ; Alamat data jam 2
32:      Jam2    Equ 38h       ; Alamat data jam 3
33:      Mnt0    Equ 39h       ; Alamat data menit 0
34:      Mnt1    Equ 3Ah       ; Alamat data menit 1
35:      Mnt2    Equ 3Bh       ; Alamat data menit 2
36:      Dtk0    Equ 3Ch       ; Alamat data detik 0
37:      Dtk1    Equ 3Dh       ; Alamat data detik 1
38:      Dtk2    Equ 3Eh       ; Alamat data detik 2
39:      Tgl0    Equ 3Fh       ; Alamat data detik 0
40:      Tgl1    Equ 40h       ; Alamat data detik 1
41:      Tgl2    Equ 41h       ; Alamat data detik 2
42:      Bln0    Equ 42h       ; Alamat data bulan 0
43:      Bln1    Equ 43h       ; Alamat data detik 1
44:      Bln2    Equ 44h       ; Alamat data detik 2
45:      Thn0    Equ 45h
46:      Thn1    Equ 46h
47:      Thn2    Equ 47h
48:      Cnt0    Equ 52h
49:      Cnt1    Equ 53h
50:      Dspl    Equ 54h          ; counter display

```

```

51:      Stds   Equ 55h           ; status display
52:      Tstd   Equ 56h
53:      Tjln   Equ 57h
54:      Dots   Equ 58h
55:      Jbrs   Equ 59h
56:      Dly0   Equ 5Ah
57:      Dly1   Equ 5Bh
58: ;-----;
59: ;      Program Utama           ;
60: ;-----;
61: init:  lcall  ser_in          ; panggil subroutine ser_in
62:      mov   Dspl,#0           ; isi Dspl dengan data 0
63:      mov   Stds,#0          ; isi Stds dengan data 0
64: ;
65: mulai: mov   A,Dspl          ; isi A dengan Dspl
66:      cjne  A,#0,dsp1        ; bandingkan A dengan 0 jika belum sama lompat ke dsp1
67:      lcall  tmpali          ; tampil ali machfudin
68: dsp1:  cjne  A,#1,dsp2        ; bandingkan A dengan 1 jika belum sama lompat ke dsp2
69:      lcall  tmpdkm          ; tampil selamat datang di kota malang
70: dsp2:  cjne  A,#2,dsp3        ; bandingkan A dengan 2 jika belum sama lompat ke dsp3
71:      lcall  tmpnkm          ; tampil selamat menikmati perjalanan anda
72: dsp3:  cjne  A,#3,dsp4        ; bandingkan A dengan 3 jika belum sama lompat ke dsp4
73:      lcall  tmppth          ; tampil patuhi rambu-rambu lalulintas
74: dsp4:  cjne  A,#4,dsp5        ; bandingkan A dengan 4 jika belum sama lompat ke dsp5
75:      lcall  tmpglj          ; tampil gunakan lajur kanan untuk mendahului
76: dsp5:  lcall  tmpsh          ; tampil suhu
77:      lcall  tmpco          ; tampil kadar co
78:      lcall  tmptg          ; tampil tanggal
79:      lcall  tmpjm          ; tampil jam
80:      inc   Dspl             ; Dspl kurangi 1
81:      mov   A,Dspl          ; isi A dengan Dspl
82:      cjne  A,#5,dsp6        ; andingkan A dengan 5 jika belum sama lompat ke dsp6
83:      mov   Dspl,#0          ; isi Dspl dengan data 0
84: dsp6:  ljmp  mulai          ; lompat ke mulai
85: ;-----;
86: ;      Program Menampilkan Karakter Tulisan Suhu           ;
87: ;-----;
88: tlsru: mov   Tstd,#0          ; isi Tstd dengan data 0
89:      mov   Dots,Tstd         ; isi Dots dengan isi dari Tstd
90:      mov   DPTR,#suhu        ; isi DPTR dengan suhu
91:      mov   Cnt0,#28          ; isi Cnt0 dengan data 28
92:      lcall  tulis0           ; tampilkan tulis0
93:      mov   Dots,Tstd         ; angka pertama
94:      mov   Cnt0,#6           ; isi Cnt0 dengan data 6
95:      mov   A,Dsh1           ; isi A dengan isi dari Dsh1
96:      lcall  scnagk          ; tampil scnagk
97:      lcall  tulis0           ; tampil tulis0
98:      mov   Dots,Tstd         ; isi Dots dengan isi dari Tstd (angka kedua)
99:      mov   Cnt0,#6           ; isi Cnt0 dengan data 6
100:     mov   A,Dsh2           ; isi A dengan isi dari Dsh2

```

```

101:    lcall  scnagk      ; tampil scnagk
102:    lcall  tulis0     ; tampil tulis0
103:    mov   Dots,Tstd   ; isi Dots dengan isi dari Tstd (char derajat)
104:    mov   DPTR,#drajt ; isi DPTR dengan drajt
105:    mov   Cnt0,#5     ; isi Cnt0 dengan data 5
106:    lcall  tulis0     ; tampil tulis0
107:    mov   Dots,Tstd   ; isi Dots dengan isi dari Tstd (char celcius)
108:    mov   DPTR,#clcius ; isi DPTR dengan clcius
109:    mov   Cnt0,#5     ; isi Cnt0 dengan data 5
110:    lcall  tulis0     ; tampil tulis0
111:    lcall  tmp0      ; tampil tmp0
112:    mov   Dots,Tstd   ; isi Dots dengan isi dari Tstd (suhu)
113:    mov   DPTR,#suhu  ; isi DPTR dengan suhu
114:    mov   Cnt0,#28    ; isi Cnt0 dengan data 28
115:    lcall  tulis1     ; tampil tulis1
116:    mov   Dots,Tstd   ; isi Dots dengan isi dari Tstd (angka pertama)
117:    mov   Cnt0,#6     ; isi Cnt0 dengan data 6
118:    mov   A,Dsh1     ; isi A dengan Dsh1
119:    lcall  scnagk     ; tampil scnagk
120:    lcall  tulis1     ; tampil tulis1
121:    mov   Dots,Tstd   ; isi Dots dengan isi dari Tstd (angka kedua)
122:    mov   Cnt0,#6     ; isi Cnt0 dengan data 6
123:    mov   A,Dsh2     ; isi A dengan Dsh2
124:    lcall  scnagk     ; tampil scnagk
125:    lcall  tulis1     ; tampil tulis1
126:    mov   Dots,Tstd   ; isi Dots dengan isi dari Tstd (char derajat)
127:    mov   DPTR,#drajt ; isi DPTR dengan drajt
128:    mov   Cnt0,#5     ; isi Cnt0 dengan data 5
129:    lcall  tulis1     ; tampil tulis1
130:    mov   Dots,Tstd   ; isi Dots dengan isi dari Tstd (char celcius)
131:    mov   DPTR,#clcius ; isi DPTR dengan clcius
132:    mov   Cnt0,#5     ; isi Cnt0 dengan data 5
133:    lcall  tulis1     ; tampil tulis1
134:    lcall  tmp1      ; tampil tmp1
135:    mov   Dots,Tstd   ; isi Dots dengan isi dari Tstd (suhu)
136:    mov   DPTR,#suhu  ; isi DPTR dengan suhu
137:    mov   Cnt0,#28    ; isi Cnt0 dengan data 28
138:    lcall  tulis2     ; tampil tulis2
139:    mov   Dots,Tstd   ; isi Dots dengan isi dari Tstd (angka pertama)
140:    mov   Cnt0,#6     ; isi Cnt0 dengan data 6
141:    mov   A,Dsh1     ; isi A dengan Dsh1
142:    lcall  scnagk     ; tampil scnagk
143:    lcall  tulis2     ; tampil tulis2
144:    mov   Dots,Tstd   ; isi Dots dengan isi dari Tstd (angka kedua)
145:    mov   Cnt0,#6     ; isi Cnt0 dengan data 6
146:    mov   A,Dsh2     ; isi A dengan Dsh2
147:    lcall  scnagk     ; tampil scnagk
148:    lcall  tulis2     ; tampil tulis2
149:    mov   Dots,Tstd   ; isi Dots dengan isi dari Tstd (char derajat)
150:    mov   DPTR,#drajt ; isi DPTR dengan drajt

```

```

251: ;-----;
252: ;----- Program Menampilkan karakter Tulisan CO -----;
253: ;-----;
254: tikdco: mov     Tstd,#0          ; isi Tstd dengan data 0
255:         mov     Dots,Tstd       ; isi Dots dengan isi dari Tstd {kdrco}
256:         mov     DPTR,#kdrco     ; isi DPTR dengan tulisan CO
257:         mov     Cnt0,#17        ; Cnt0 isi dengan data 17
258:         lcall   tulis0          ; tampil tulis0
259:         mov     Dots,Tstd       ; isi Dots dengan isi dari Tstd (angka pertama)
260:         mov     Cnt0,#6         ; Cnt0 isi dengan data 6
261:         mov     A,Dco1          ; isi A dengan isi dari Dco1
262:         lcall   scnagk          ; tampil scnagk
263:         lcall   tulis0          ; tampil tulis0
264:         mov     Dots,Tstd       ; isi Dots dengan isi dari Tstd (angka kedua)
265:         mov     Cnt0,#6         ; Cnt0 isi dengan data 6
266:         mov     A,Dco2          ; isi A dengan isi dari Dco2
267:         lcall   scnagk          ; tampil scnagk
268:         lcall   tulis0          ; tampil tulis0
269:         mov     Dots,Tstd       ; isi Dots dengan isi dari Tstd (angka titik)
270:         mov     Cnt0,#2         ; Cnt0 isi dengan data 2
271:         mov     A,#11           ; isi A dengan data 11
272:         lcall   scnagk          ; tampil scnagk
273:         lcall   tulis0          ; tampil tulis0
274:         mov     Dots,Tstd       ; isi Dots dengan isi dari Tstd (angka ketiga)
275:         mov     Cnt0,#6         ; Cnt0 isi dengan data 6
276:         mov     A,Dco3          ; isi A dengan isi dari Dco3
277:         lcall   scnagk          ; tampil scnagk
278:         lcall   tulis0          ; tampil tulis0
279:         mov     Dots,Tstd       ; isi Dots dengan isi dari Tstd (angka keempat)
280:         mov     Cnt0,#7         ; Cnt0 isi dengan data 7
281:         mov     A,Dco4          ; isi A dengan isi dari Dco4
282:         lcall   scnagk          ; tampil scnagk
283:         lcall   tulis0          ; tampil tulis0
284:         mov     Dots,Tstd       ; isi Dots dengan isi dari Tstd (char persen)
285:         mov     DPTR,#prcnt     ; isi DPTR dengan tulisan prcnt
286:         mov     Cnt0,#6         ; Cnt0 isi dengan data 6
287:         lcall   tulis0          ; tampil tulis0
288:         lcall   tmp10          ; tampil tmp0
289:         mov     Dots,Tstd       ; isi Dots dengan isi dari Tstd {kdrco}
290:         mov     DPTR,#kdrco     ; isi DPTR dengan tulisan CO
291:         mov     Cnt0,#17        ; Cnt0 isi dengan data 17
292:         lcall   tulis1          ; tampil tulis1
293:         mov     Dots,Tstd       ; isi Dots dengan isi dari Tstd (angka pertama)
294:         mov     Cnt0,#6         ; Cnt0 isi dengan data 6
295:         mov     A,Dco1          ; isi A dengan isi dari Dco1
296:         lcall   scnagk          ; tampil scnagk
297:         lcall   tulis1          ; tampil tulis1
298:         mov     Dots,Tstd       ; isi Dots dengan isi dari Tstd (angka kedua)
299:         mov     Cnt0,#6         ; Cnt0 isi dengan data 6
300:         mov     A,Dco2          ; isi A dengan isi dari Dco2

```

```

451:      mov     A,Dco4           ; isi A dengan isi dari Dco4
452:      lcall  scnagk           ; tampil scnagk
453:      lcall  tulis5           ; tampil tulis5
454:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (char persen)
455:      mov     DPTR,#prcnt      ; isi DPTR dengan tulisan prcnt
456:      mov     Cnt0,#6          ; Cnt0 isi dengan data 6
457:      lcall  tulis5           ; tampil tulis5
458:      lcall  tmp15            ; tampil tmp15
459:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (CO
460:      mov     DPTR,#kdrco      ; isi DPTR dengan tulisan CO
461:      mov     Cnt0,#17         ; Cnt0 isi dengan data 17
462:      lcall  tulis6           ; tampil tulis6
463:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (angka pertama)
464:      mov     Cnt0,#6          ; Cnt0 isi dengan data 6
465:      mov     A,Dco1           ; isi A dengan isi dari Dco1
466:      lcall  scnagk           ; tampil scnagk
467:      lcall  tulis6           ; tampil tulis6
468:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstda (angka kedua)
469:      mov     Cnt0,#6          ; Cnt0 isi dengan data 6
470:      mov     A,Dco2           ; isi A dengan isi dari Dco2
471:      lcall  scnagk           ; tampil scnagk
472:      lcall  tulis6           ; tampil tulis6
473:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstda (angka titik)
474:      mov     Cnt0,#2          ; Cnt0 isi dengan data 2
475:      mov     A,#11            ; isi A dengan data 11
476:      lcall  scnagk           ; tampil scnagk
477:      lcall  tulis6           ; tampil tulis6
478:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstda (angka ketiga)
479:      mov     Cnt0,#6          ; Cnt0 isi dengan data 6
480:      mov     A,Dco3           ; isi A dengan isi dari Dco3
481:      lcall  scnagk           ; tampil scnagk
482:      lcall  tulis6           ; tampil tulis6
483:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstda (angka keempat)
484:      mov     Cnt0,#7          ; Cnt0 isi dengan data 7
485:      mov     A,Dco4           ; isi A dengan isi dari Dco4
486:      lcall  scnagk0          ; tampil scnagk0
487:      lcall  tulis6           ; tampil tulis6
488:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstda (char persen)
489:      mov     DPTR,#prcnt      ; isi DPTR dengan tulisan prcnt
490:      mov     Cnt0,#6          ; Cnt0 isi dengan data 6
491:      lcall  tulis6           ; tampil tulis6
492:      lcall  tmp16            ; tampil tmp16
493:      ret                     ; kembali ke program utama
494: ;-----;
495: ;      Program Menampilkan Tanggal      ;
496: ;-----;
497: t1stgl: mov     Tstd,#0        ; isi Tstd dengan data 0
498:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (tgl angka pertama)
499:      mov     Cnt0,#6          ; isi Cnt0 dengan data 6
500:      mov     A,Tgl1           ; isi A dengan isi dari Tgl1

```

```

501:      lcall  scnagk      ; tampil scnagk
502:      lcall  tulis0     ; tampil tulis0
503:      mov    Dots,Tstd  ; isi Dots dengan isi dari Tstd (tgl angka kedua)
504:      mov    Cnt0,#6    ; isi Cnt0 dengan data 6
505:      mov    A,Tgl2     ; isi A dengan isi dari Tgl2
506:      lcall  scnagk     ; tampil scnagk
507:      lcall  tulis0     ; tampil tulis0
508:      mov    Dots,Tstd  ; isi Dots dengan isi dari Tstd (strip)
509:      mov    Cnt0,#6    ; isi Cnt0 dengan data 6
510:      mov    A,#12     ; isi A dengan data 12
511:      lcall  scnagk     ; tampil scnagk
512:      lcall  tulis0     ; tampil tulis0
513:      mov    Dots,Tstd  ; isi Dots dengan isi dari Tstd (bln angka pertama)
514:      mov    Cnt0,#6    ; isi Cnt0 dengan data 6
515:      mov    A,Bln1    ; isi A dengan isi dari Bln1
516:      lcall  scnagk     ; tampil scnagk
517:      lcall  tulis0     ; tampil tulis0
518:      mov    Dots,Tstd  ; isi Dots dengan isi dari Tstd (bln angka kedua)
519:      mov    Cnt0,#6    ; isi Cnt0 dengan data 6
520:      mov    A,Bln2    ; isi A dengan isi dari Bln2
521:      lcall  scnagk     ; tampil scnagk
522:      lcall  tulis0     ; tampil tulis0
523:      mov    Dots,Tstd  ; isi Dots dengan isi dari Tstd (strip)
524:      mov    Cnt0,#6    ; isi Cnt0 dengan data 6
525:      mov    A,#12     ; isi A dengan data 12
526:      lcall  scnagk     ; tampil scnagk
527:      lcall  tulis0     ; tampil tulis0
528:      mov    Dots,Tstd  ; isi Dots dengan isi dari Tstd (thn angka pertama)
529:      mov    Cnt0,#6    ; isi Cnt0 dengan data 6
530:      mov    A,Thn1    ; isi A dengan isi dari Thn1
531:      lcall  scnagk     ; tampil scnagk
532:      lcall  tulis0     ; tampil tulis0
533:      mov    Dots,Tstd  ; isi Dots dengan isi dari Tstd (thn angka kedua)
534:      mov    Cnt0,#7    ; isi Cnt0 dengan data 7
535:      mov    A,Thn2    ; isi A dengan isi dari Thn2
536:      lcall  scnagk     ; tampil scnagk
537:      lcall  tulis0     ; tampil tulis0
538:      lcall  tmp10     ; tampil tmp10
539: ;
540:      mov    Dots,Tstd  ; isi Dots dengan isi dari Tstd (tgl angka pertama)
541:      mov    Cnt0,#6    ; isi Cnt0 dengan data 6
542:      mov    A,Tgl1     ; isi A dengan isi dari Tgl1
543:      lcall  scnagk     ; tampil scnagk
544:      lcall  tulis1     ; tampil tulis1
545:      mov    Dots,Tstd  ; isi Dots dengan isi dari Tstd (tgl angka kedua)
546:      mov    Cnt0,#6    ; isi Cnt0 dengan data 6
547:      mov    A,Tgl2     ; isi A dengan isi dari Tgl2
548:      lcall  scnagk     ; tampil scnagk
549:      lcall  tulis1     ; tampil tulis1
550:      mov    Dots,Tstd  ; isi Dots dengan isi dari Tstd (strip)

```

```

651:      mov     A,#12           ; isi A dengan data 12
652:      lcall  scnagk          ; tampil scnagk
653:      lcall  tulis3          ; tampil tulis3
654:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (thn angka pertama)
655:      mov     Cnt0,#6        ; isi Cnt0 dengan data 6
656:      mov     A,Thn1         ; isi A dengan isi dari Thn1
657:      lcall  scnagk          ; tampil scnagk
658:      lcall  tulis3          ; tampil tulis3
659:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (thn angka kedua)
660:      mov     Cnt0,#7        ; isi Cnt0 dengan data 7
661:      mov     A,Thn2         ; isi A dengan isi dari Thn1
662:      lcall  scnagk          ; tampil scnagk
663:      lcall  tulis3          ; tampil tulis3
664:      lcall  tmp13           ; tampil tmp13
665: ;
666:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (tgl angka pertama)
667:      mov     Cnt0,#6        ; isi Cnt0 dengan data 6
668:      mov     A,Tgl1         ; isi A dengan isi dari Tgl1
669:      lcall  scnagk          ; tampil scnagk
670:      lcall  tulis4          ; tampil tulis4
671:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (tgl angka kedua)
672:      mov     Cnt0,#6        ; isi Cnt0 dengan data 6
673:      mov     A,Tgl2         ; isi A dengan isi dari Tgl2
674:      lcall  scnagk          ; tampil scnagk
675:      lcall  tulis4          ; tampil tulis4
676:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (strip)
677:      mov     Cnt0,#6        ; isi Cnt0 dengan data 6
678:      mov     A,#12          ; isi A dengan data 12
679:      lcall  scnagk          ; tampil scnagk
680:      lcall  tulis4          ; tampil tulis4
681:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (bln angka pertama)
682:      mov     Cnt0,#6        ; isi Cnt0 dengan data 6
683:      mov     A,Bln1         ; isi A dengan isi dari Bln1
684:      lcall  scnagk          ; tampil scnagk
685:      lcall  tulis4          ; tampil tulis4
686:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (bln angka kedua)
687:      mov     Cnt0,#6        ; isi Cnt0 dengan data 6
688:      mov     A,Bln2         ; isi A dengan isi dari Bln2
689:      lcall  scnagk          ; tampil scnagk
690:      lcall  tulis4          ; tampil tulis4
691:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (strip)
692:      mov     Cnt0,#6        ; isi Cnt0 dengan data 6
693:      mov     A,#12          ; isi A dengan data 12
694:      lcall  scnagk          ; tampil scnagk
695:      lcall  tulis4          ; tampil tulis4
696:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (thn angka pertama)
697:      mov     Cnt0,#6        ; isi Cnt0 dengan data 6
698:      mov     A,Thn1         ; isi A dengan isi dari Thn1
699:      lcall  scnagk          ; tampil scnagk
700:      lcall  tulis4          ; tampil tulis4

```

```

751:      mov     Cnt0,#6           ; isi Cnt0 dengan data 6
752:      mov     A,Tgl1           ; isi A dengan isi dari Tgl1
753:      lcall   scnagk           ; tampil scnagk
754:      lcall   tulis6           ; tampil tulis6
755:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (tgl angka kedua)
756:      mov     Cnt0,#6           ; isi Cnt0 dengan data 6
757:      mov     A,Tgl2           ; isi A dengan isi dari Tgl2
758:      lcall   scnagk           ; tampil scnagk
759:      lcall   tulis6           ; tampil tulis6
760:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (strip)
761:      mov     Cnt0,#6           ; isi Cnt0 dengan data 6
762:      mov     A,#12            ; isi A dengan data 12
763:      lcall   scnagk           ; tampil scnagk
764:      lcall   tulis6           ; tampil tulis6
765:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (bln angka pertama)
766:      mov     Cnt0,#6           ; isi Cnt0 dengan data 6
767:      mov     A,Bln1           ; isi A dengan isi dari Bln1
768:      lcall   scnagk           ; tampil scnagk
769:      lcall   tulis6           ; tampil tulis6
770:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (bln angka kedua)
771:      mov     Cnt0,#6           ; isi Cnt0 dengan data 6
772:      mov     A,Bln2           ; isi A dengan isi dari Bln2
773:      lcall   scnagk           ; tampil scnagk
774:      lcall   tulis6           ; tampil tulis6
775:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (strip)
776:      mov     Cnt0,#6           ; isi Cnt0 dengan data 6
777:      mov     A,#12            ; isi A dengan data 12
778:      lcall   scnagk           ; tampil scnagk
779:      lcall   tulis6           ; tampil tulis6
780:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (thn angka pertama)
781:      mov     Cnt0,#6           ; isi Cnt0 dengan data 6
782:      mov     A,Thn1           ; isi A dengan isi dari Thn1
783:      lcall   scnagk           ; tampil scnagk
784:      lcall   tulis6           ; tampil tulis6
785:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (thn angka kedua)
786:      mov     Cnt0,#7           ; isi Cnt0 dengan data 7
787:      mov     A,Thn2           ; isi A dengan isi dari Thn2
788:      lcall   scnagk           ; tampil scnagk
789:      lcall   tulis6           ; tampil tulis6
790:      lcall   tmlp6            ; tampil tmlp6
791:      ret                      ; kembali ke program utama
792: ;-----;
793: ;      Program Menulis Karakter Jam      ;
794: ;-----;
795: tlsjam: mov     Tstd,#0         ; isi Tstd dengan data 0
796:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (jam angka pertama)
797:      mov     Cnt0,#6           ; isi Cnt0 dengan data 6
798:      mov     A,Jam1           ; isi A dengan isi dari jam1
799:      lcall   scnagk           ; tampil scnagk
800:      lcall   tulis0           ; tampil tulis0

```



```

801:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (jam angka kedua)
802:      mov     Cnt0,#6       ; isi Cnt0 dengan data 6
803:      mov     A,Jam2        ; isi A dengan isi dari jam2
804:      lcall   scnagk        ; tampil scnagk
805:      lcall   tulis0        ; tampil tulis0
806:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (strip)
807:      mov     Cnt0,#6       ; isi Cnt0 dengan data 6
808:      mov     A,#13         ; isi A dengan data 13
809:      lcall   scnagk        ; tampil scnagk
810:      lcall   tulis0        ; tampil tulis0
811:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (mnt angka pertama)
812:      mov     Cnt0,#6       ; isi Cnt0 dengan data 6
813:      mov     A,Mnt1        ; isi A dengan isi dari Mnt1
814:      lcall   scnagk        ; tampil scnagk
815:      lcall   tulis0        ; tampil tulis0
816:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (mnt angka kedua)
817:      mov     Cnt0,#6       ; isi Cnt0 dengan data 6
818:      mov     A,Mnt2        ; isi A dengan isi dari Mnt2
819:      lcall   scnagk        ; tampil scnagk
820:      lcall   tulis0        ; tampil tulis0
821:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (strip)
822:      mov     Cnt0,#6       ; isi Cnt0 dengan data 6
823:      mov     A,#13         ; isi A dengan data 13
824:      lcall   scnagk        ; tampil scnagk
825:      lcall   tulis0        ; tampil tulis0
826:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (dtk angka pertama)
827:      mov     Cnt0,#6       ; isi Cnt0 dengan data 6
828:      mov     A,Dtk1        ; isi A dengan isi dari Dtk1
829:      lcall   scnagk        ; tampil scnagk
830:      lcall   tulis0        ; tampil tulis0
831:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (dtk angka kedua)
832:      mov     Cnt0,#7       ; isi Cnt0 dengan data 7
833:      mov     A,Dtk2        ; isi A dengan isi dari Dtk2
834:      lcall   scnagk        ; tampil scnagk
835:      lcall   tulis0        ; tampil tulis0
836:      lcall   tmp10         ; tampil tmp10
837: ;
838:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (jam angka pertama)
839:      mov     Cnt0,#6       ; isi Cnt0 dengan data 6
840:      mov     A,Jam1        ; isi A dengan isi dari jam1
841:      lcall   scnagk        ; tampil scnagk
842:      lcall   tulis1        ; tampil tulis1
843:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (jam angka kedua)
844:      mov     Cnt0,#6       ; isi Cnt0 dengan data 6
845:      mov     A,Jam2        ; isi A dengan isi dari jam2
846:      lcall   scnagk        ; tampil scnagk
847:      lcall   tulis1        ; tampil tulis1
848:      mov     Dots,Tstd      ; isi Dots dengan isi dari Tstd (strip)
849:      mov     Cnt0,#6       ; isi Cnt0 dengan data 6
850:      mov     A,#13         ; isi A dengan data 13

```

```

1001:      mov     Cnt0,#7           ; isi Cnt0 dengan data 7
1002:      mov     A,Dtk2          ; isi A dengan isi dari Dtk2
1003:      lcall   scnagk          ; tampil scnagk
1004:      lcall   tulis4          ; tampil tulis4
1005:      lcall   tmp14           ; tampil tmp14
1006: ;
1007:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (jam angka pertama)
1008:      mov     Cnt0,#6          ; isi Cnt0 dengan data 6
1009:      mov     A,Jam1           ; isi A dengan isi dari jam1
1010:      lcall   scnagk          ; tampil scnagk
1011:      lcall   tulis5          ; tampil tulis5
1012:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (jam angka kedua)
1013:      mov     Cnt0,#6          ; isi Cnt0 dengan data 6
1014:      mov     A,Jam2           ; isi A dengan isi dari jam2
1015:      lcall   scnagk          ; tampil scnagk
1016:      lcall   tulis5          ; tampil tulis5
1017:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (strip)
1018:      mov     Cnt0,#6          ; isi Cnt0 dengan data 6
1019:      mov     A,#13            ; isi A dengan data 13
1020:      lcall   scnagk          ; tampil scnagk
1021:      lcall   tulis5          ; tampil tulis5
1022:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (mnt angka pertama)
1023:      mov     Cnt0,#6          ; isi Cnt0 dengan data 6
1024:      mov     A,Mnt1           ; isi A dengan isi dari Mnt1
1025:      lcall   scnagk          ; tampil scnagk
1026:      lcall   tulis5          ; tampil tulis5
1027:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (mnt angka kedua)
1028:      mov     Cnt0,#6          ; isi Cnt0 dengan data 6
1029:      mov     A,Mnt2           ; isi A dengan isi dari Mnt2
1030:      lcall   scnagk          ; tampil scnagk
1031:      lcall   tulis5          ; tampil tulis5
1032:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (strip)
1033:      mov     Cnt0,#6          ; isi Cnt0 dengan data 6
1034:      mov     A,#13            ; isi A dengan data 13
1035:      lcall   scnagk          ; tampil scnagk
1036:      lcall   tulis5          ; tampil tulis5
1037:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (dtk angka pertama)
1038:      mov     Cnt0,#6          ; isi Cnt0 dengan data 6
1039:      mov     A,Dtk1           ; isi A dengan isi dari Dtk1
1040:      lcall   scnagk          ; tampil scnagk
1041:      lcall   tulis5          ; tampil tulis5
1042:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (dtk angka kedua)
1043:      mov     Cnt0,#7          ; isi Cnt0 dengan data 7
1044:      mov     A,Dtk2          ; isi A dengan isi dari Dtk2
1045:      lcall   scnagk          ; tampil scnagk
1046:      lcall   tulis5          ; tampil tulis5
1047:      lcall   tmp15           ; tampil tmp15
1048: ;
1049:      mov     Dots,Tstd        ; isi Dots dengan isi dari Tstd (jam angka pertama)
1050:      mov     Cnt0,#6          ; isi Cnt0 dengan data 6

```

```

1051:      mov     A,Jam1      ; isi A dengan isi dari jam1
1052:      lcall   scnagk      ; tampil scnagk
1053:      lcall   tulis6      ; tampil tulis6
1054:      mov     Dots,Tstd   ; isi Dots dengan isi dari Tstd (jam angka kedua)
1055:      mov     Cnt0,#6     ; isi Cnt0 dengan data 6
1056:      mov     A,Jam2      ; isi A dengan isi dari jam2
1057:      lcall   scnagk      ; tampil scnagk
1058:      lcall   tulis6      ; tampil tulis5
1059:      mov     Dots,Tstd   ; isi Dots dengan isi dari Tstd (strip)
1060:      mov     Cnt0,#6     ; isi Cnt0 dengan data 6
1061:      mov     A,#13       ; isi A dengan data 13
1062:      lcall   scnagk      ;tampil scnagk
1063:      lcall   tulis6      ; tampil tulis6
1064:      mov     Dots,Tstd   ; isi Dots dengan isi dari Tstd (mnt angka pertama)
1065:      mov     Cnt0,#6     ; isi Cnt0 dengan data 6
1066:      mov     A,Mnt1      ; isi A dengan isi dari Mnt1
1067:      lcall   scnagk      ; tampil scnagk
1068:      lcall   tulis6      ; tampil tulis6
1069:      mov     Dots,Tstd   ; isi Dots dengan isi dari Tstd (mnt angka kedua)
1070:      mov     Cnt0,#6     ; isi Cnt0 dengan data 6
1071:      mov     A,Mnt2      ; isi A dengan isi dari Mnt2
1072:      lcall   scnagk      ; tampil scnagk
1073:      lcall   tulis6      ; tampil tulis6
1074:      mov     Dots,Tstd   ; isi Dots dengan isi dari Tstd (strip)
1075:      mov     Cnt0,#6     ; isi Cnt0 dengan data 6
1076:      mov     A,#13       ; isi A dengan data 13
1077:      lcall   scnagk      ; tampil scnagk
1078:      lcall   tulis6      ; tampil tulis6
1079:      mov     Dots,Tstd   ; isi Dots dengan isi dari Tstd (dtk angka pertama)
1080:      mov     Cnt0,#6     ; isi Cnt0 dengan data 6
1081:      mov     A,Dtk1      ; isi A dengan isi dari Dtk1
1082:      lcall   scnagk      ; tampil scnagk
1083:      lcall   tulis6      ; tampil tulis6
1084:      mov     Dots,Tstd   ; isi Dots dengan isi dari Tstd (dtk angka kedua)
1085:      mov     Cnt0,#7     ; isi Cnt0 dengan data 7
1086:      mov     A,Dtk2      ; isi A dengan isi dari Dtk2
1087:      lcall   scnagk      ; tampil scnagk
1088:      lcall   tulis6      ; tampil tulis6
1089:      lcall   tmp16      ; tampil tmp16
1090:      ret                ; kembali ke program utama
1091: ;-----;
1092: ;      Program Menampilkan Tulisan Berjalan      ;
1093: ;-----;
1094: t1sj1n: mov     Tjln,#0   ; isi Tjln dengan data 0
1095: t1sj10: mov     Dots,Tjln ; isi Dots dengan isi dari Tjln
1096:      mov     Cnt0,#50    ; isi Cnt0 dengan data 50
1097:      lcall   tulis0      ; tampil tulis0
1098:      lcall   tmp10      ; tampil tmp10
1099:      mov     Dots,Tjln   ; isi Dots dengan isi dari Tjln
1100:      mov     Cnt0,#50    ; isi Cnt0 dengan data 50

```

```

1251: ;-----Program Scanning Baris-----;
1252: ;-----;
1253: tmp10: mov    P2,#1111110b  ; isi P2 dengan data 1111110b
1254:        lcall  delays        ; panggil subroutine delays
1255:        mov    P2,#1111111b  ; isi P2 dengan data 1111111b
1256:        ret                    ; panggil subroutine delays
1257: ;
1258: tmp11: mov    P2,#11111101b ; isi P2 dengan data 11111101b
1259:        lcall  delays        ; panggil subroutine delays
1260:        mov    P2,#1111111b  ; isi P2 dengan data 1111111b
1261:        ret                    ; panggil subroutine delays
1262: ;
1263: tmp12: mov    P2,#11111011b ; isi P2 dengan data 11111011b
1264:        lcall  delays        ; panggil subroutine delays
1265:        mov    P2,#1111111b  ; isi P2 dengan data 1111111b
1266:        ret                    ; panggil subroutine delays
1267: ;
1268: tmp13: mov    P2,#11110111b ; isi P2 dengan data 11110111b
1269:        lcall  delays        ; panggil subroutine delays
1270:        mov    P2,#1111111b  ; isi P2 dengan data 1111111b
1271:        ret                    ; panggil subroutine delays
1272: ;
1273: tmp14: mov    P2,#11101111b ; isi P2 dengan data 11101111b
1274:        lcall  delays        ; panggil subroutine delays
1275:        mov    P2,#1111111b  ; isi P2 dengan data 1111111b
1276:        ret                    ; panggil subroutine delays
1277: ;
1278: tmp15: mov    P2,#11011111b ; isi P2 dengan data 11011111b
1279:        lcall  delays        ; panggil subroutine delays
1280:        mov    P2,#1111111b  ; isi P2 dengan data 1111111b
1281:        ret                    ; panggil subroutine delays
1282: ;
1283: tmp16: mov    P2,#10111111b ; isi P2 dengan data 10111111b
1284:        lcall  delays        ; panggil subroutine delays
1285:        mov    P2,#1111111b  ; isi P2 dengan data 1111111b
1286:        ret                    ; kembali ke program utama
1287: ;-----;
1288: ;-----Program Menampilkan Pesan---;
1289: ;-----;
1290: tmpali: mov    DPTR,#alimcf  ; isi DPTR dengan alamat alimcf
1291:        mov    Jbrs,#192      ; isi Jbrs dengan data 192
1292:        lcall  tlsjln        ; tampil tlsjln
1293:        ret                    ; kembali ke program utama
1294: ;
1295: tmpdkm: mov    DPTR,#sdkmlg  ; isi DPTR dengan alamat sdkmlg
1296:        mov    Jbrs,#225      ; isi Jbrs dengan data 255
1297:        lcall  tlsjln        ; tampil tlsjln
1298:        ret                    ; kembali ke program utama
1299: ;
1300: tmpnkm: mov    DPTR,#nikmat  ; isi DPTR dengan alamat nikmat

```

```

1301:      mov     Jbrs,#250      ; isi Jbrs dengan data 250
1302:      lcall  tlsjln          ; tampil tlsjln
1303:      ret                      ; kembali ke program utama
1304: ;
1305: tmppth: mov     DPTR,#ptrr11  ; isi DPTR dengan alamat ptrr11
1306:      mov     Jbrs,#250      ; isi Jbrs dengan data 250
1307:      lcall  tlsjln          ; tampil tlsjln
1308:      ret                      ; kembali ke program utama
1309: ;
1310: tmpglj: mov     DPTR,#glkumd   ; isi DPTR dengan alamat glkumd
1311:      mov     Jbrs,#255      ; isi Jbrs dengan data 255
1312:      lcall  tlsjln          ; tampil tlsjln
1313:      ret                      ; kembali ke program utama
1314: ;-----;
1315: ;-----Program Menampilkan Suhu-----;
1316: ;-----;
1317: tmpsh:  mov     A,#0F0h       ; isi A dengan data F0h
1318:      lcall  kr_srl          ; panggil kr_srl
1319:      lcall  delayk          ; panggil subroutine delayk
1320:      mov     B,#10           ; isi B dengan data 10
1321:      div     AB              ;
1322:      mov     Dsh1,A          ; isi Dsh1 dengan isi dari A
1323:      mov     Dsh2,B          ; isi Dsh2 dengan isi dari B
1324:      mov     Cnt1,#255      ; isi Cnt1 dengan data 255
1325: tshlp:  lcall  tlsruu        ; tampil tlsruu
1326:      djnz   Cnt1,tshlp      ; Cnt1 kurangi 1 jika belum 0 lompat ke tshlp
1327:      ret                      ; kembali ke program utama
1328: ;-----;
1329: ;      Program Menampilkan CO      ;
1330: ;-----;
1331: tmpco:  mov     A,#0F1h       ; isi A dengan data F1h
1332:      lcall  kr_srl          ; panggil kr_srl
1333:      lcall  delayk          ; panggil subroutine delayk
1334:      lcall  k1b00           ; panggil k1b00
1335:      mov     Cnt1,#255      ; isi Cnt1 dengan data 255
1336: tcolp:  lcall  tikdco        ; tampil tikdco
1337:      djnz   Cnt1,tcolp      ; Cnt1 kurangi 1 jika belum 0 lompat ke tcolp
1338:      ret                      ; kembali ke program utama
1339: ;-----;
1340: ;      Program Menampilkan Tanggal  ;
1341: ;-----;
1342: tmptg:  mov     A,#0F2h       ; isi A dengan data F2h
1343:      lcall  kr_srl          ; panggil kr_srl
1344:      lcall  delayk          ; panggil subroutine delayk
1345:      mov     B,#10           ; isi B dengan data 10
1346:      div     AB              ; A dibagi B
1347:      mov     Tg11,A          ; isi Tg11 dengan isi dari A
1348:      mov     Tg12,B          ; isi Tg12 dengan isi dari B
1349:      mov     A,#0F3h       ; isi A dengan data F3h
1350:      lcall  kr_srl          ; panggil kr_srl

```

```

1351:      lcall  delayk      ; panggil subroutine delayk
1352:      mov   B,#10        ; isi B dengan data 10
1353:      div   AB           ; A dibagi B
1354:      mov   Bin1,A       ; isi Bin1 dengan isi dari A
1355:      mov   Bln2,B       ; isi Bln2 dengan isi dari B
1356:      mov   A,#0F4h     ; isi A dengan data F4h
1357:      lcall  kr_srl      ; panggil kr_srl
1358:      lcall  delayk      ; panggil subroutine delayk
1359:      mov   B,#10        ; isi B dengan data 10
1360:      div   AB           ; A dibagi B
1361:      mov   Thn1,A       ; isi Thn1 dengan isi dari A
1362:      mov   Thn2,B       ; isi Thn2 dengan isi dari B
1363:      mov   Cnt1,#255    ; isi Cnt1 dengan data 255
1364: loptg: lcall  tlstgl    ; tampil tlstgl
1365:      djnz  Cnt1,loptg   ; Cnt1 kurangi 1 jika belum nol lompat ke loptg
1366:      ret                ; kembali ke program utama
1367: ;-----;
1368: ;      Program Menampilkan Jam      ;
1369: ;-----;
1370: tmpjm: mov   A,#0F5h     ; isi A dengan data F5h
1371:      lcall  kr_srl      ; panggil kr_srl
1372:      lcall  delayk      ; panggil subroutine delayk
1373:      mov   B,#10        ; isi B dengan data 10
1374:      div   AB           ; A dibagi B
1375:      mov   Jam1,A       ; isi Jam1 dengan isi dari A
1376:      mov   Jam2,B       ; isi Jam2 dengan isi dari B
1377:      mov   A,#0F6h     ; isi A dengan data F6h
1378:      lcall  kr_srl      ; panggil kr_srl
1379:      lcall  delayk      ; panggil subroutine delayk
1380:      mov   B,#10        ; isi B dengan data 10
1381:      div   AB           ; A dibagi B
1382:      mov   Mnt1,A       ; isi Mnt1 dengan isi dari A
1383:      mov   Mnt2,B       ; isi Mnt2 dengan isi dari B
1384:      mov   A,#0F7h     ; isi A dengan data F7h
1385:      lcall  kr_srl      ; panggil kr_srl
1386:      lcall  delayk      ; panggil subroutine delayk
1387:      mov   B,#10        ; isi B dengan data 10
1388:      div   AB           ; A dibagi B
1389:      mov   Dtk1,A       ; isi Dtk1 dengan isi dari A
1390:      mov   Dtk2,B       ; isi Dtk2 dengan isi dari B
1391:      mov   Cnt1,#255    ; isi Cnt1 dengan data 255
1392: lopjm: lcall  tlsjam     ; tampil tlsjam
1393:      djnz  Cnt1,lopjm   ; Cnt1 kurangi 1 jika belum nol lompat ke lopjam
1394:      ret                ; kembali ke program utama
1395: ;-----;
1396: ;      Program Kalibrasi Suhu, Co, Tanggal      ;
1397: ;-----;
1398: klbsh: mov   A,#0F0h     ; isi A dengan data F0h (Kalibrasi Suhu)
1399:      lcall  kr_srl      ; panggil kr_srl
1400:      lcall  delayk      ; panggil subroutine delayk

```

```

1101:      lcall  tulis1      ; tampil tulis1
1102:      lcall  tmp11       ; tampil tmp11
1103:      mov    Dots,Tjln   ; isi Dots dengan isi dari Tjln
1104:      mov    Cnt0,#50    ; isi Cnt0 dengan data 50
1105:      lcall  tulis2      ; tampil tulis2
1106:      lcall  tmp12       ; tampil tmp12
1107:      mov    Dots,Tjln   ; isi Dots dengan isi dari Tjln
1108:      mov    Cnt0,#50    ; isi Cnt0 dengan data 50
1109:      lcall  tulis3      ; tampil tulis3
1110:      lcall  tmp13       ; tampil tmp13
1111:      mov    Dots,Tjln   ; isi Dots dengan isi dari Tjln
1112:      mov    Cnt0,#50    ; isi Cnt0 dengan data 50
1113:      lcall  tulis4      ; tampil tulis4
1114:      lcall  tmp14       ; tampil tmp14
1115:      mov    Dots,Tjln   ; isi Dots dengan isi dari Tjln
1116:      mov    Cnt0,#50    ; isi Cnt0 dengan data 50
1117:      lcall  tulis5      ; tampil tulis5
1118:      lcall  tmp15       ; tampil tmp15
1119:      mov    Dots,Tjln   ; isi Dots dengan isi dari Tjln
1120:      mov    Cnt0,#50    ; isi Cnt0 dengan data 50
1121:      lcall  tulis6      ; tampil tulis6
1122:      lcall  tmp16       ; tampil tmp16
1123:      inc   Tjln        ; Tjln ditambah 1
1124:      djnz  Jbrs,tlsl11 ; Jbrs kurangi 1 jika belum nol lompat ke tlsl11
1125:      ljmp  tlslj12     ; lompat ke tlslj12
1126: tlslj11: ljmp  tlslj10  ; lompat ke tlslj10
1127:      ret                ; kembali ke program utama
1128: ;-----;
1129: ;      Program Menampilkan Tulisan Pada Dot Matrik      ;
1130: ;-----;
1131: tulis0: mov    A,Dots    ; isi A dengan isi dari Dots
1132:      movc  A,@A+DPTR    ; isi A dengan isi alamat yang ditunjukkan oleh DPTR
1133:      jnb  Acc.1,mtdts0  ; lompat ke mtdts0 jika Acc.1 = 0
1134:      setb DtAB          ; DtAB = 1
1135:      ljmp nydts0        ; lompat ke nydts0
1136: mtdts0: clr    DtAB     ; DtAB = 0
1137: nydts0: lcall  clock    ; tampil clock
1138:      setb DtAB          ; DtAB = 1
1139:      inc  Dots          ; Dots ditambah 1
1140:      djnz Cnt0,tulis0   ; Cnt0 kurangi 1 jika belum 0 lompat ke tulis0
1141:      ret                ; kembali ke program utama
1142: ;
1143: tulis1: mov    A,Dots    ; isi A dengan isi dari Dots
1144:      movc  A,@A+DPTR    ; isi A dengan isi alamat yang ditunjukkan oleh DPTR
1145:      jnb  Acc.2,mtdts1  ; lompat ke mtdts1 jika Acc.2 = 0
1146:      setb DtAB          ; DtAB = 0
1147:      ljmp nydts1        ; lompat ke nydts1
1148: mtdts1: clr    DtAB     ; DtAB = 0
1149: nydts1: lcall  clock    ; tampil clock
1150:      setb DtAB          ; DtAB = 0

```

```

1201:      ret                ; kembali ke program utama
1202: ;
1203: tulis6: mov      A,Dots          ; isi A dengan isi dari Dots
1204:      movc     A,@A+DPTR        ; isi A dengan isi alamat yang ditunjukkan oleh DPTR
1205:      jb       Acc.7,mtdts6     ; lompat ke mtdts6 jika Acc.7 = 1
1206:      setb     DtAB             ; DtAB = 0
1207:      ljmp     nydts6          ; lompat ke nydts6
1208: mtdts6: clr      DtAB          ; DtAB = 0
1209: nydts6: lcall   clock         ; tampil clock
1210:      setb     DtAB             ; DtAB = 0
1211:      inc      Dots             ; Dots ditambah 1
1212:      djnz     Cnt0,tulis6      ; Cnt0 kurangi 1 jika belum 0 lompat ke tulis6
1213:      ret                ; kembali ke program utama
1214: ;-----;
1215: ;      Program Untuk Scanning Angka      ;
1216: ;-----;
1217: scnakg: cjne    A,#0,scnakg0    ; bandingkan A dengan 0 jika tidak sama lompat ke scnakg0
1218:      mov      DPTR,#nol        ; isi DPTR dengan alamat nol
1219: scnakg0: cjne   A,#1,scnakg1    ; bandingkan A dengan 1 jika tidak sama lompat ke scnakg1
1220:      mov      DPTR,#satu       ; isi DPTR dengan alamat satu
1221: scnakg1: cjne   A,#2,scnakg2    ; bandingkan A dengan 2 jika tidak sama lompat ke scnakg2
1222:      mov      DPTR,#dua        ; isi DPTR dengan alamat dua
1223: scnakg2: cjne   A,#3,scnakg3    ; bandingkan A dengan 3 jika tidak sama lompat ke scnakg3
1224:      mov      DPTR,#tiga       ; isi DPTR dengan alamat tiga
1225: scnakg3: cjne   A,#4,scnakg4    ; bandingkan A dengan 4 jika tidak sama lompat ke scnakg4
1226:      mov      DPTR,#empat      ; isi DPTR dengan alamat empat
1227: scnakg4: cjne   A,#5,scnakg5    ; bandingkan A dengan 5 jika tidak sama lompat ke scnakg5
1228:      mov      DPTR,#lima       ; isi DPTR dengan alamat lima
1229: scnakg5: cjne   A,#6,scnakg6    ; bandingkan A dengan 6 jika tidak sama lompat ke scnakg6
1230:      mov      DPTR,#enam       ; isi DPTR dengan alamat enam
1231: scnakg6: cjne   A,#7,scnakg7    ; bandingkan A dengan 7 jika tidak sama lompat ke scnakg7
1232:      mov      DPTR,#tujuh      ; isi DPTR dengan alamat tujuh
1233: scnakg7: cjne   A,#8,scnakg8    ; bandingkan A dengan 8 jika tidak sama lompat ke scnakg8
1234:      mov      DPTR,#dlpan      ; isi DPTR dengan alamat dlpan
1235: scnakg8: cjne   A,#9,scnakg9    ; bandingkan A dengan 9 jika tidak sama lompat ke scnakg9
1236:      mov      DPTR,#smbln      ; isi DPTR dengan alamat smbln
1237: scnakg9: cjne   A,#11,scnakgA   ; bandingkan A dengan 11 jika tidak sama lompat ke scnakgA
1238:      mov      DPTR,#titik      ; isi DPTR dengan alamat titik
1239: scnakgA: cjne   A,#12,scnakgB   ; bandingkan A dengan 12 jika tidak sama lompat ke scnakgB
1240:      mov      DPTR,#strip      ; isi DPTR dengan alamat strip
1241: scnakgB: cjne   A,#13,scnakgC   ; bandingkan A dengan 13 jika tidak sama lompat ke scnakgC
1242:      mov      DPTR,#petik      ; isi DPTR dengan alamat petik
1243: scnakgC: ret
1244: ;-----;
1245: ;-----Subroutine Clock-----;
1246: ;-----;
1247: clock: setb     Clck           ; Clock = 1
1248:      clr      Clck           ; Clock = 0
1249:      ret                ; kembali ke program utama
1250: ;-----;

```



```

1501:      mov     A,#0F8h      ; isi A dengan data F0h
1502:      lcall   kr_srl       ; tampil kr_srl
1503:      mov     A,#0DDh      ; isi A dengan data DDh
1504: ktgl4:  cjne   A,#0DDh,ktgl5 ; bandingkan A dengan DDh jika tidak sama lompat ke ktgl5
1505:      ljmp    ktgl4        ; lompat ke ktgl4
1506: ktgl5:  lcall   jeda       ; panggil jeda
1507:      mov     A,Tgl0       ; isi A dengan isi dari Tgl0
1508:      lcall   kr_srl       ; tampil kr_srl
1509:      mov     A,#0DDh      ; isi A dengan data DDh
1510: ktgl6:  cjne   A,#0DDh,ktgl7 ; bandingkan A dengan DDh jika tidak sama lompat ke ktgl7
1511:      ljmp    ktgl6        ; lompat ke ktgl6
1512: ktgl7:  lcall   jeda       ; panggil jeda
1513:      mov     A,Bln0       ; isi A dengan isi dari Bln0
1514:      lcall   kr_srl       ; tampil kr_srl
1515:      mov     A,#0DDh      ; isi A dengan data DDh
1516: ktgl8:  cjne   A,#0DDh,ktgl9 ; bandingkan A dengan DDh jika tidak sama lompat ke ktgl9
1517:      ljmp    ktgl8        ; lompat ke ktgl8
1518: ktgl9:  lcall   jeda       ; panggil jeda
1519:      mov     A,Thn0       ; isi A dengan isi dari Thn0
1520:      lcall   kr_srl       ; tampil kr_srl
1521:      ljmp    mulai        ; lompat ke mulai
1522: ;-----;
1523: ;      Program Kalibrasi Jam      ;
1524: ;-----;
1525: kljam:  mov     A,#0F5h      ; isi A dengan data 0F5h
1526:      lcall   kr_srl       ; panggil kr_srl
1527:      lcall   delayk       ; panggil delayk
1528:      mov     B,#10        ; isi B dengan data 10
1529:      div     AB           ; A dibagi B
1530:      mov     Jam1,A       ; isi Jam1 dengan isi dari A
1531:      mov     Jam2,B       ; isi Jam2 dengan isi dari B
1532:      mov     A,#0F6h      ; isi A dengan data 0F6h
1533:      lcall   kr_srl       ; panggil kr_srl
1534:      lcall   delayk       ; panggil delayk
1535:      mov     B,#10        ; isi B dengan data 10
1536:      div     AB           ; A dibagi B
1537:      mov     Mnt1,A       ; isi Mnt1 dengan isi dari A
1538:      mov     Mnt2,B       ; isi Mnt2 dengan isi dari B
1539:      mov     A,#0F7h      ; isi A dengan data 0F7h
1540:      lcall   kr_srl       ; panggil kr_srl
1541:      lcall   delayk       ; panggil delayk
1542:      mov     B,#10        ; isi B dengan data 10
1543:      div     AB           ; A dibagi B
1544:      mov     Dtk1,A       ; isi Dtk1 dengan isi dari A
1545:      mov     Dtk2,B       ; isi Dtk2 dengan isi dari B
1546:      mov     Cnt1,#10     ; isi Cnt1 dengan data 10
1547: kljm0:  lcall   tlsjam     ; tampil tlsjam
1548:      djnz   Cnt1,kljm0    ; Cnt1 kurangi 1 jika belum nol lompat ke kljm0
1549:      lcall   scnkpdp      ; tampil scnkpdp
1550:      cjne   R0,#11,kljm1 ; bandingkan R0 dengan 11 jika tidak sama lompat ke kljm1

```

```

1601: kljm6:  cjne   A,#0DDh,kljm7  ; bandingkan A dengan 0DDh jika tidak sama lompat ke kljm7
1602:          ljmp   kljm6          ; lompat ke kljm6
1603: kljm7:  lcall  jeda            ; panggil jeda
1604:          mov   A,Mnt0         ; isi A dengan isi dari Mnt0
1605:          lcall  kr_srl        ; tampil kr_srl
1606:          ljmp   mulai         ; lompat ke mulai
1607: ;-----;
1608: ;      Program Initial Serial      ;
1609: ;-----;
1610: ser_in:  setb   EA              ; aktifkan semua interrupt
1611:          mov   TMOD,#20h       ; isi TMOD dengan data 20h
1612:          mov   TH1,#0F3h      ; isi TH1 dengan data 0F3h
1613:          setb  TR1             ; TR1=1
1614:          mov   SCON,#50h      ; isi SCON dengan data 50h
1615:          setb  ES              ; enable serial = 1
1616:          ret                    ; kembali ke program utama
1617: ;-----;
1618: ;      Program Kirim Serial        ;
1619: ;-----;
1620: kr_srl:  clr    ES              ; enable serial=0
1621:          mov   SBUF,A          ; isi SBUF dengan isi dari A
1622:          jnb   TI,$            ; lompat jika TI nol
1623:          clr   TI              ; TI=0
1624:          setb  ES              ; enable serial=1
1625:          ret                    ; kembali ke program utama
1626: ;-----;
1627: ;      Program Scanning Keypad     ;
1628: ;-----;
1629: scnkbd:  mov    R0,#10          ; status tombol
1630: brs1:   mov    P1,#11111101b
1631:          mov    R1,P1
1632: b1c1:   cjne   R1,#11111001b,b1c2
1633:          mov    R0,#1
1634: b1c2:   cjne   R1,#11111100b,b1c3
1635:          mov    R0,#2
1636: b1c3:   cjne   R1,#11101101b,brs2
1637:          mov    R0,#3
1638: ;
1639: brs2:   mov    P1,#10111111b
1640:          mov    R1,P1
1641: b2c1:   cjne   R1,#10111011b,b2c2
1642:          mov    R0,#4
1643: b2c2:   cjne   R1,#10111110b,b2c3
1644:          mov    R0,#5
1645: b2c3:   cjne   R1,#10101111b,brs3
1646:          mov    R0,#6
1647: ;
1648: brs3:   mov    P1,#11011111b
1649:          mov    R1,P1
1650: b3c1:   cjne   R1,#11011011b,b3c2

```

```

1651:      mov      R0,#7
1652: b3c2:  cjne     R1,#11011110b,b3c3
1653:      mov      R0,#8
1654: b3c3:  cjne     R1,#11001111b,brs4
1655:      mov      R0,#9
1656: ;
1657: brs4:  mov      P1,#11110111b
1658:      mov      R1,P1
1659: b4c1:  cjne     R1,#11110011b,b4c2
1660:      mov      R0,#11
1661: b4c2:  cjne     R1,#11110110b,b4c3
1662:      mov      R0,#0
1663: b4c3:  cjne     R1,#11100111b,back
1664:      mov      R0,#12
1665: back:  ret
1666: ;
1667: tg_lps: lcall   scnkpdc      ; tampil scnkpdc
1668:      mov      P2,#11111111b ; isi P2 dengan data 11111111b
1669:      cjne     R0,#10,tg_lps ; bandingkan R0 dengan 10 jika tidak sama lompat ke tg_lps
1670:      ret
1671: ;
1672: tg_tkn: mov      A,Stds      ; isi A dengan isi dari Stds
1673:      cjne     A,#1,tg_tk1   ; bandingkan A dengan 1 jika tidak sama lompat ke tg_tk1
1674:      lcall   tlstg1         ; tampil tlstg1
1675: tg_tk1: cjne     A,#2,tg_tk2 ; bandingkan A dengan 2 jika tidak sama lompat ke tg_tk2
1676:      lcall   tlsjam        ; tampil tlsjam
1677: tg_tk2: lcall   scnkpdc      ; tampil scnkpdc
1678:      cjne     R0,#11,tg_tk3  ; bandingkan R0 dengan 11 jika tidak sama lompat ke tg_tk3
1679:      ljmp    tg_tkn         ; lompat tg_tkn
1680: tg_tk3: cjne     R0,#12,tg_tk4 ; bandingkan R0 dengan 12 jika tidak sama lompat ke tg_tk4
1681:      ljmp    tg_tkn         ; lompat tg_tkn
1682: tg_tk4: cjne     R0,#10,tg_tk5 ; bandingkan R0 dengan 10 jika tidak sama lompat ke tg_tk5
1683:      ljmp    tg_tkn         ; lompat tg_tkn
1684: tg_tk5: ret
1685:
1686: ;
1687: jeda:  djnz    Dly0,$        ; Dly0 kurangi 1 jika belum nol kembalilagi
1688:      ret
1689: ;-----;
1690: ;      Sub Routine Delay      ;
1691: ;-----;
1692: delays: mov      Dly1,#4      ; isi Dly1 dengan data 4
1693: dlys:  lcall   jeda          ; panggil jeda
1694:      djnz    Dly1,dlys      ; Dly1 kurangi 1 jika belum nol lompat ke dlys
1695:      ret
1696: ;
1697: delayk: mov      Dly1,#10     ; isi Dly1 dengan data 10
1698: dlyk:  lcall   jeda          ; panggil jeda
1699:      lcall   scnkpdc        ; tampil scnkpdc
1700:      cjne     R0,#11,dlkx    ; bandingkan R0 dengan 11 jika tidak sama lompat ke dlkx

```

```

1701:      lcall  tg_lps      ; tampil tg_lps
1702:      ljmp   klbsh      ; lompat ke klbsh
1703: dlkx: djnz   Dly1,dlyk ; Dly1 kurangi 1 jika belum nol lompat ke dlys
1704:      ret                ; kembali ke program utama
1705: ;-----;
1706: ;      Kalibrasi CO      ;
1707: ;-----;
1708: klb00: cjne   A,#28,klb01      ; 00.05
1709:      mov    Dco1,#0
1710:      mov    Dco2,#0
1711:      mov    Dco3,#0
1712:      mov    Dco4,#5
1713: klb01: cjne   A,#29,klb02      ; 00.06
1714:      mov    Dco1,#0
1715:      mov    Dco2,#0
1716:      mov    Dco3,#0
1717:      mov    Dco4,#6
1718: klb02: cjne   A,#30,klb03      ; 00.07
1719:      mov    Dco1,#0
1720:      mov    Dco2,#0
1721:      mov    Dco3,#0
1722:      mov    Dco4,#7
1723: klb03: cjne   A,#31,klb04      ; 00.08
1724:      mov    Dco2,#0
1725:      mov    Dco3,#0
1726:      mov    Dco4,#8
1727: klb04: cjne   A,#32,klb05      ; 00.09
1728:      mov    Dco1,#0
1729:      mov    Dco2,#0
1730:      mov    Dco3,#0
1731:      mov    Dco4,#9
1732: klb05: cjne   A,#33,klb06      ; 00.10
1733:      mov    Dco1,#0
1734:      mov    Dco2,#0
1735:      mov    Dco3,#1
1736:      mov    Dco4,#0
1737: klb06: cjne   A,#34,klb07      ; 00.11
1738:      mov    Dco1,#0
1739:      mov    Dco2,#0
1740:      mov    Dco3,#1
1741:      mov    Dco4,#1
1742: klb07: cjne   A,#35,klb08      ; 00.12
1743:      mov    Dco1,#0
1744:      mov    Dco2,#0
1745:      mov    Dco3,#1
1746:      mov    Dco4,#2
1747: klb08: cjne   A,#36,klb09      ; 00.13
1748:      mov    Dco1,#0
1749:      mov    Dco2,#0
1750:      mov    Dco3,#1

```

```

1751:      mov      Dco4,#3
1752:  k1b09:  cjne   A,#37,k1b10      ; 00.14
1753:      mov      Dco1,#0
1754:      mov      Dco2,#0
1755:      mov      Dco3,#1
1756:      mov      Dco4,#4
1757:  k1b10:  cjne   A,#38,k1b11      ; 00.15
1758:      mov      Dco1,#0
1759:      mov      Dco2,#0
1760:      mov      Dco3,#1
1761:      mov      Dco4,#5
1762:  k1b11:  cjne   A,#39,k1b12      ; 00.16
1763:      mov      Dco1,#0
1764:      mov      Dco2,#0
1765:      mov      Dco3,#1
1766:      mov      Dco4,#6
1767:  k1b12:  cjne   A,#40,k1b13      ; 00.17
1768:      mov      Dco1,#0
1769:      mov      Dco2,#0
1770:      mov      Dco3,#1
1771:      mov      Dco4,#7
1772:  k1b13:  cjne   A,#41,k1b14      ; 00.18
1773:      mov      Dco1,#0
1774:      mov      Dco2,#0
1775:      mov      Dco3,#1
1776:      mov      Dco4,#8
1777:  k1b14:  cjne   A,#42,k1b15      ; 00.19
1778:      mov      Dco1,#0
1779:      mov      Dco2,#0
1780:      mov      Dco3,#1
1781:      mov      Dco4,#9
1782:  k1b15:  cjne   A,#43,k1b16      ; 00.20
1783:      mov      Dco1,#0
1784:      mov      Dco2,#0
1785:      mov      Dco3,#2
1786:      mov      Dco4,#0
1787:  k1b16:  cjne   A,#44,k1b17      ; 00.21
1788:      mov      Dco1,#0
1789:      mov      Dco2,#0
1790:      mov      Dco3,#2
1791:      mov      Dco4,#1
1792:  k1b17:  cjne   A,#45,k1b18      ; 00.22
1793:      mov      Dco1,#0
1794:      mov      Dco2,#0
1795:      mov      Dco3,#2
1796:      mov      Dco4,#2
1797:  k1b18:  cjne   A,#46,k1b19      ; 00.23
1798:      mov      Dco1,#0
1799:      mov      Dco2,#0
1800:      mov      Dco3,#2

```

| | | | |
|-------|--------|----------|-----------------------|
| 1951: | mov | Dco4, #3 | |
| 1952: | klb49: | cjne | A, #77, k1b50 ; 00.54 |
| 1953: | mov | Dco1, #0 | |
| 1954: | mov | Dco2, #0 | |
| 1955: | mov | Dco3, #5 | |
| 1956: | mov | Dco4, #4 | |
| 1957: | klb50: | cjne | A, #78, k1b51 ; 00.55 |
| 1958: | mov | Dco1, #0 | |
| 1959: | mov | Dco2, #0 | |
| 1960: | mov | Dco3, #5 | |
| 1961: | mov | Dco4, #5 | |
| 1962: | klb51: | cjne | A, #79, k1b52 ; 00.56 |
| 1963: | mov | Dco1, #0 | |
| 1964: | mov | Dco2, #0 | |
| 1965: | mov | Dco3, #5 | |
| 1966: | mov | Dco4, #6 | |
| 1967: | klb52: | cjne | A, #80, k1b53 ; 00.57 |
| 1968: | mov | Dco1, #0 | |
| 1969: | mov | Dco2, #0 | |
| 1970: | mov | Dco3, #5 | |
| 1971: | mov | Dco4, #7 | |
| 1972: | klb53: | cjne | A, #81, k1b54 ; 00.58 |
| 1973: | mov | Dco1, #0 | |
| 1974: | mov | Dco2, #0 | |
| 1975: | mov | Dco3, #5 | |
| 1976: | mov | Dco4, #8 | |
| 1977: | klb54: | cjne | A, #82, k1b55 ; 00.59 |
| 1978: | mov | Dco1, #0 | |
| 1979: | mov | Dco2, #0 | |
| 1980: | mov | Dco3, #5 | |
| 1981: | mov | Dco4, #9 | |
| 1982: | klb55: | cjne | A, #83, k1b56 ; 00.60 |
| 1983: | mov | Dco1, #0 | |
| 1984: | mov | Dco2, #0 | |
| 1985: | mov | Dco3, #6 | |
| 1986: | mov | Dco4, #0 | |
| 1987: | klb56: | cjne | A, #84, k1b57 ; 00.61 |
| 1988: | mov | Dco1, #0 | |
| 1989: | mov | Dco2, #0 | |
| 1990: | mov | Dco3, #6 | |
| 1991: | mov | Dco4, #1 | |
| 1992: | klb57: | cjne | A, #85, k1b58 ; 00.62 |
| 1993: | mov | Dco1, #0 | |
| 1994: | mov | Dco2, #0 | |
| 1995: | mov | Dco3, #6 | |
| 1996: | mov | Dco4, #2 | |
| 1997: | klb58: | cjne | A, #86, k1b59 ; 00.63 |
| 1998: | mov | Dco1, #0 | |
| 1999: | mov | Dco2, #0 | |
| 2000: | mov | Dco3, #6 | |

```

2001:      mov      Dco4,#3
2002: k1b59: cjne   A,#87,k1b60      ; 00.64
2003:      mov      Dco1,#0
2004:      mov      Dco2,#0
2005:      mov      Dco3,#6
2006:      mov      Dco4,#4
2007: k1b60: cjne   A,#88,k1b61      ; 00.65
2008:      mov      Dco1,#0
2009:      mov      Dco2,#0
2010:      mov      Dco3,#6
2011:      mov      Dco4,#5
2012: k1b61: cjne   A,#89,k1b62      ; 00.66
2013:      mov      Dco1,#0
2014:      mov      Dco2,#0
2015:      mov      Dco3,#6
2016:      mov      Dco4,#6
2017: k1b62: cjne   A,#90,k1b63      ; 00.67
2018:      mov      Dco1,#0
2019:      mov      Dco2,#0
2020:      mov      Dco3,#6
2021:      mov      Dco4,#7
2022: k1b63: cjne   A,#91,k1b64      ; 00.68
2023:      mov      Dco1,#0
2024:      mov      Dco2,#0
2025:      mov      Dco3,#6
2026:      mov      Dco4,#8
2027: k1b64: cjne   A,#92,k1b65      ; 00.69
2028:      mov      Dco1,#0
2029:      mov      Dco2,#0
2030:      mov      Dco3,#6
2031:      mov      Dco4,#9
2032: k1b65: cjne   A,#93,k1b66      ; 00.70
2033:      mov      Dco1,#0
2034:      mov      Dco2,#0
2035:      mov      Dco3,#7
2036:      mov      Dco4,#0
2037: k1b66: cjne   A,#94,k1b67      ; 00.71
2038:      mov      Dco1,#0
2039:      mov      Dco2,#0
2040:      mov      Dco3,#7
2041:      mov      Dco4,#1
2042: k1b67: cjne   A,#95,k1b68      ; 00.72
2043:      mov      Dco1,#0
2044:      mov      Dco2,#0
2045:      mov      Dco3,#7
2046:      mov      Dco4,#2
2047: k1b68: cjne   A,#96,k1b69      ; 00.73
2048:      mov      Dco1,#0
2049:      mov      Dco2,#0
2050:      mov      Dco3,#7

```

```

2051:      mov      Dco4,#3
2052: k1b69:  cjne     A,#97,k1b70      ; 00.74
2053:      mov      Dco1,#0
2054:      mov      Dco2,#0
2055:      mov      Dco3,#7
2056:      mov      Dco4,#4
2057: k1b70:  cjne     A,#98,k1b71      ; 00.75
2058:      mov      Dco1,#0
2059:      mov      Dco2,#0
2060:      mov      Dco3,#7
2061:      mov      Dco4,#5
2062: k1b71:  cjne     A,#99,k1b72      ; 00.76
2063:      mov      Dco1,#0
2064:      mov      Dco2,#0
2065:      mov      Dco3,#7
2066:      mov      Dco4,#6
2067: k1b72:  cjne     A,#100,k1b73     ; 00.77
2068:      mov      Dco1,#0
2069:      mov      Dco2,#0
2070:      mov      Dco3,#7
2071:      mov      Dco4,#7
2072: k1b73:  cjne     A,#101,k1b74     ; 00.78
2073:      mov      Dco1,#0
2074:      mov      Dco2,#0
2075:      mov      Dco3,#7
2076:      mov      Dco4,#8
2077: k1b74:  cjne     A,#102,k1b75     ; 00.79
2078:      mov      Dco1,#0
2079:      mov      Dco2,#0
2080:      mov      Dco3,#7
2081:      mov      Dco4,#9
2082: k1b75:  cjne     A,#103,k1b76     ; 00.80
2083:      mov      Dco1,#0
2084:      mov      Dco2,#0
2085:      mov      Dco3,#8
2086:      mov      Dco4,#0
2087: k1b76:  cjne     A,#104,k1b77     ; 00.81
2088:      mov      Dco1,#0
2089:      mov      Dco2,#0
2090:      mov      Dco3,#8
2091:      mov      Dco4,#1
2092: k1b77:  cjne     A,#105,k1b78     ; 00.82
2093:      mov      Dco1,#0
2094:      mov      Dco2,#0
2095:      mov      Dco3,#8
2096:      mov      Dco4,#2
2097: k1b78:  cjne     A,#106,k1b79     ; 00.83
2098:      mov      Dco1,#0
2099:      mov      Dco2,#0
2100:      mov      Dco3,#8

```



```

2251:      mov      Dco4,#3
2252: k1b109: cjne    A,#137,k1b110      ; 01.14
2253:      mov      Dco1,#0
2254:      mov      Dco2,#1
2255:      mov      Dco3,#1
2256:      mov      Dco4,#4
2257: k1b110: cjne    A,#138,k1b111      ; 01.15
2258:      mov      Dco1,#0
2259:      mov      Dco2,#1
2260:      mov      Dco3,#1
2261:      mov      Dco4,#5
2262: k1b111: cjne    A,#139,k1b112      ; 01.16
2263:      mov      Dco1,#0
2264:      mov      Dco2,#1
2265:      mov      Dco3,#1
2266:      mov      Dco4,#6
2267: k1b112: cjne    A,#140,k1b113      ; 01.17
2268:      mov      Dco1,#0
2269:      mov      Dco2,#1
2270:      mov      Dco3,#1
2271:      mov      Dco4,#7
2272: k1b113: cjne    A,#141,k1b114      ; 01.18
2273:      mov      Dco1,#0
2274:      mov      Dco2,#1
2275:      mov      Dco3,#1
2276:      mov      Dco4,#8
2277: k1b114: cjne    A,#142,k1b115      ; 01.19
2278:      mov      Dco1,#0
2279:      mov      Dco2,#1
2280:      mov      Dco3,#1
2281:      mov      Dco4,#9
2282: k1b115: cjne    A,#143,k1b116      ; 01.20
2283:      mov      Dco1,#0
2284:      mov      Dco2,#1
2285:      mov      Dco3,#2
2286:      mov      Dco4,#0
2287: k1b116: cjne    A,#144,k1b117      ; 01.21
2288:      mov      Dco1,#0
2289:      mov      Dco2,#1
2290:      mov      Dco3,#2
2291:      mov      Dco4,#1
2292: k1b117: cjne    A,#145,k1b118      ; 01.22
2293:      mov      Dco1,#0
2294:      mov      Dco2,#1
2295:      mov      Dco3,#2
2296:      mov      Dco4,#2
2297: k1b118: cjne    A,#146,k1b119      ; 01.23
2298:      mov      Dco1,#0
2299:      mov      Dco2,#1
2300:      mov      Dco3,#2

```

```

2301:      mov      Dco4,#3
2302: k1b119: cjne   A,#147,k1b120      ; 01.24
2303:      mov      Dco1,#0
2304:      mov      Dco2,#1
2305:      mov      Dco3,#2
2306:      mov      Dco4,#4
2307: k1b120: cjne   A,#148,k1b121      ; 01.25
2308:      mov      Dco1,#0
2309:      mov      Dco2,#1
2310:      mov      Dco3,#2
2311:      mov      Dco4,#5
2312: k1b121: cjne   A,#149,k1b122      ; 01.26
2313:      mov      Dco1,#0
2314:      mov      Dco2,#1
2315:      mov      Dco3,#2
2316:      mov      Dco4,#6
2317: k1b122: cjne   A,#150,k1b123      ; 01.27
2318:      mov      Dco1,#0
2319:      mov      Dco2,#1
2320:      mov      Dco3,#2
2321:      mov      Dco4,#7
2322: k1b123: cjne   A,#151,k1b124      ; 01.28
2323:      mov      Dco1,#0
2324:      mov      Dco2,#1
2325:      mov      Dco3,#2
2326:      mov      Dco4,#8
2327:
2328: k1b124: cjne   A,#152,k1b125      ; 01.29
2329:      mov      Dco1,#0
2330:      mov      Dco2,#1
2331:      mov      Dco3,#2
2332:      mov      Dco4,#9
2333: k1b125: cjne   A,#153,k1b126      ; 01.30
2334:      mov      Dco1,#0
2335:      mov      Dco2,#1
2336:      mov      Dco3,#3
2337:      mov      Dco4,#0
2338: k1b126: cjne   A,#154,k1b127      ; 01.31
2339:      mov      Dco1,#0
2340:      mov      Dco2,#1
2341:      mov      Dco3,#3
2342:      mov      Dco4,#1
2343: k1b127: cjne   A,#155,k1b128      ; 01.32
2344:      mov      Dco1,#0
2345:      mov      Dco2,#1
2346:      mov      Dco3,#3
2347:      mov      Dco4,#2
2348: k1b128: cjne   A,#156,k1b129      ; 01.33
2349:      mov      Dco1,#0
2350:      mov      Dco2,#1

```

```

2351:      mov      Dco3,#3
2352:      mov      Dco4,#3
2353: k1b129:  cjne   A,#157,k1b130      ; 01.34
2354:      mov      Dco1,#0
2355:      mov      Dco2,#1
2356:      mov      Dco3,#3
2357:      mov      Dco4,#4
2358: k1b130:  cjne   A,#158,k1b131      ; 01.35
2359:      mov      Dco1,#0
2360:      mov      Dco2,#1
2361:      mov      Dco3,#3
2362:      mov      Dco4,#5
2363: k1b131:  cjne   A,#159,k1b132      ; 01.36
2364:      mov      Dco1,#0
2365:      mov      Dco2,#1
2366:      mov      Dco3,#3
2367:      mov      Dco4,#6
2368: k1b132:  cjne   A,#160,k1b133      ; 01.37
2369:      mov      Dco1,#0
2370:      mov      Dco2,#1
2371:      mov      Dco3,#3
2372:      mov      Dco4,#7
2373: k1b133:  cjne   A,#161,k1b134      ; 01.38
2374:      mov      Dco1,#0
2375:      mov      Dco2,#1
2376:      mov      Dco3,#3
2377:      mov      Dco4,#8
2378: k1b134:  cjne   A,#162,k1b135      ; 01.39
2379:      mov      Dco1,#0
2380:      mov      Dco2,#1
2381:      mov      Dco3,#3
2382:      mov      Dco4,#9
2383: k1b135:  cjne   A,#163,k1b136      ; 01.40
2384:      mov      Dco1,#0
2385:      mov      Dco2,#1
2386:      mov      Dco3,#4
2387:      mov      Dco4,#0
2388: k1b136:  cjne   A,#164,k1b137      ; 01.41
2389:      mov      Dco1,#0
2390:      mov      Dco2,#1
2391:      mov      Dco3,#4
2392:      mov      Dco4,#1
2393: k1b137:  cjne   A,#165,k1b138      ; 01.42
2394:      mov      Dco1,#0
2395:      mov      Dco2,#1
2396:      mov      Dco3,#4
2397:      mov      Dco4,#2
2398: k1b138:  cjne   A,#166,k1b139      ; 01.43
2399:      mov      Dco1,#0
2400:      mov      Dco2,#1

```

```

2401:      mov      Dco3,#4
2402:      mov      Dco4,#3
2403: k1b139:  cjne   A,#167,k1b140      ; 01.44
2404:      mov      Dco1,#0
2405:      mov      Dco2,#1
2406:      mov      Dco3,#4
2407:      mov      Dco4,#4
2408: k1b140:  cjne   A,#168,k1b141      ; 01.45
2409:      mov      Dco1,#0
2410:      mov      Dco2,#1
2411:      mov      Dco3,#4
2412:      mov      Dco4,#5
2413: k1b141:  cjne   A,#169,k1b142      ; 01.46
2414:      mov      Dco1,#0
2415:      mov      Dco2,#1
2416:      mov      Dco3,#4
2417:      mov      Dco4,#6
2418: k1b142:  cjne   A,#170,k1b143      ; 01.47
2419:      mov      Dco1,#0
2420:      mov      Dco2,#1
2421:      mov      Dco3,#4
2422:      mov      Dco4,#7
2423: k1b143:  cjne   A,#171,k1b144      ; 01.48
2424:      mov      Dco1,#0
2425:      mov      Dco2,#1
2426:      mov      Dco3,#4
2427:      mov      Dco4,#8
2428: k1b144:  cjne   A,#172,k1b145      ; 01.49
2429:      mov      Dco1,#0
2430:      mov      Dco2,#1
2431:      mov      Dco3,#4
2432:      mov      Dco4,#9
2433: k1b145:  cjne   A,#173,k1b146      ; 01.50
2434:      mov      Dco1,#0
2435:      mov      Dco2,#1
2436:      mov      Dco3,#5
2437:      mov      Dco4,#0
2438: k1b146:  cjne   A,#174,k1b147      ; 01.51
2439:      mov      Dco1,#0
2440:      mov      Dco2,#1
2441:      mov      Dco3,#5
2442:      mov      Dco4,#1
2443: k1b147:  cjne   A,#175,k1b148      ; 01.52
2444:      mov      Dco1,#0
2445:      mov      Dco2,#1
2446:      mov      Dco3,#5
2447:      mov      Dco4,#2
2448: k1b148:  cjne   A,#176,k1b149      ; 01.53
2449:      mov      Dco1,#0
2450:      mov      Dco2,#1

```

```

2451:      mov     Dco3,#5
2452:      mov     Dco4,#3
2453: k1b149:  cjne   A,#177,k1b150      ; 01.54
2454:      mov     Dco1,#0
2455:      mov     Dco2,#1
2456:      mov     Dco3,#5
2457:      mov     Dco4,#4
2458: k1b150:  cjne   A,#178,k1b151      ; 01.55
2459:      mov     Dco1,#0
2460:      mov     Dco2,#1
2461:      mov     Dco3,#5
2462:      mov     Dco4,#5
2463:
2464:
2465: k1b151:  ret
2466: ;-----;
2467: ;                      Define Byte                      ;
2468: ;-----;
2469: nol:    DB    07Ch,082h,082h,082h,07Ch,000h,000h,000h,000h,000h
2470: satu:   DB    000h,042h,0FEh,002h,000h,000h,000h,000h,000h,000h
2471: dua:    DB    042h,086h,08Ah,092h,062h,000h,000h,000h,000h,000h
2472: tiga:   DB    084h,082h,0A2h,0D2h,08Ch,000h,000h,000h,000h,000h
2473: empat:  DB    018h,028h,048h,0FEh,008h,000h,000h,000h,000h,000h
2474: lima:   DB    0E4h,0A2h,0A2h,0A2h,09Ch,000h,000h,000h,000h,000h
2475: enam:   DB    03Ch,052h,092h,092h,00Ch,000h,000h,000h,000h,000h
2476: tujuh:  DB    080h,080h,09Eh,0A0h,0C0h,000h,000h,000h,000h,000h
2477: dlpan:  DB    06Ch,092h,092h,092h,06Ch,000h,000h,000h,000h,000h
2478: smb1n:  DB    060h,092h,092h,094h,078h,000h,000h,000h,000h,000h
2479: titik:  DB    002h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2480: strip:  DB    000h,010h,010h,010h,000h,000h,000h,000h,000h,000h
2481: petik:  DB    000h,000h,024h,000h,000h,000h,000h,000h,000h,000h
2482: drajt:  DB    000h,0E0h,0A0h,0E0h,000h,000h,000h,000h,000h,000h
2483: clcius: DB    07Ch,082h,082h,082h,044h,000h,000h,000h,000h,000h
2484: prcnt:  DB    0C4h,0C8h,010h,026h,046h,000h,000h,000h,000h,000h
2485: suhu:   DB    064h,092h,092h,092h,04Ch,000h,03Ch,002h,002h,004h
2486:         DB    03Eh,000h,0FEh,010h,020h,020h,01Eh,000h,03Ch,002h
2487:         DB    002h,004h,03Eh,000h,000h,000h,000h,000h,000h,000h
2488: kdrco:  DB    000h,07Ch,082h,082h,082h,044h,000h,01Ch,022h,022h
2489:         DB    022h,01Ch,000h,000h,000h,000h,000h,000h,000h,000h
2490:         DB    000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2491: alimcf: DB    000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2492:         DB    000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2493:         DB    000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2494:         DB    000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2495:         DB    000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2496:         DB    07Eh,090h,090h,090h,07Eh,000h,000h,082h,0FEh,002h
2497:         DB    000h,000h,000h,022h,0BEh,002h,000h,000h,000h,000h
2498:         DB    000h,000h,000h,000h,0FEh,040h,030h,040h,0FEh,000h
2499:         DB    004h,02Ah,02Ah,02Ah,01Eh,000h,01Ch,022h,022h,022h
2500:         DB    004h,000h,0FEh,010h,020h,020h,01Eh,000h,010h,07Eh

```

2501: DB 090h,080h,040h,000h,03Ch,002h,002h,004h,03Eh,000h
2502: DB 000h,01Ch,022h,022h,012h,0FEh,000h,000h,022h,0BEh
2503: DB 002h,000h,000h,03Eh,010h,020h,020h,01Eh,000h,000h
2504: DB 000h,000h,000h,000h,000h,07Ch,08Ah,092h,0A2h,07Ch
2505: DB 000h,042h,086h,08Ah,092h,062h,000h,000h,002h,000h
2506: DB 000h,000h,000h,000h,042h,0FEh,002h,000h,000h,080h
2507: DB 080h,09Eh,0A0h,0C0h,000h,000h,002h,000h,000h,000h
2508: DB 000h,07Ch,08Ah,092h,0A2h,07Ch,000h,000h,042h,0FEh
2509: DB 002h,000h,000h,018h,028h,048h,0FEh,008h,000h,000h
2510: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2511: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2512: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2513: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2514: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2515: ;
2516: sdkmlg: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2517: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2518: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2519: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2520: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2521: DB 064h,092h,092h,092h,04Ch,000h,01Ch,02Ah,02Ah,02Ah
2522: DB 018h,000h,000h,082h,0FEh,002h,000h,000h,004h,02Ah
2523: DB 02Ah,02Ah,01Eh,000h,03Eh,020h,018h,020h,01Eh,000h
2524: DB 004h,02Ah,02Ah,02Ah,01Eh,000h,020h,0FCh,022h,002h
2525: DB 004h,000h,000h,000h,000h,000h,000h,000h,0FEh,082h
2526: DB 082h,044h,038h,000h,004h,02Ah,02Ah,02Ah,01Eh,000h
2527: DB 020h,0FCh,022h,002h,004h,000h,004h,02Ah,02Ah,02Ah
2528: DB 01Eh,000h,03Eh,010h,020h,020h,01Eh,000h,010h,02Ah
2529: DB 02Ah,02Ah,03Ch,000h,000h,000h,000h,000h,000h,000h
2530: DB 01Ch,022h,022h,012h,0FEh,000h,000h,022h,0BEh,002h
2531: DB 000h,000h,000h,000h,000h,000h,000h,000h,0FEh,010h
2532: DB 028h,044h,082h,000h,01Ch,022h,022h,022h,01Ch,000h
2533: DB 020h,0FCh,022h,002h,004h,000h,004h,02Ah,02Ah,02Ah
2534: DB 01Eh,000h,000h,000h,000h,000h,000h,000h,0FEh,040h
2535: DB 030h,040h,0FEh,000h,004h,02Ah,02Ah,02Ah,01Eh,000h
2536: DB 000h,082h,0FEh,002h,000h,000h,004h,02Ah,02Ah,02Ah
2537: DB 01Eh,000h,03Eh,010h,020h,020h,01Eh,000h,010h,02Ah
2538: DB 02Ah,02Ah,03Ch,000h,000h,000h,000h,000h,000h,000h
2539: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2540: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2541: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2542: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2543: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2544: ;
2545: nikmat: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2546: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2547: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2548: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2549: DB 000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2550: DB 064h,092h,092h,092h,04Ch,000h,01Ch,02Ah,02Ah,02Ah

```

2551:      DB  018h,000h,000h,082h,0FEh,002h,000h,000h,004h,02Ah
2552:      DB  02Ah,02Ah,01Eh,000h,03Eh,020h,018h,020h,01Eh,000h
2553:      DB  004h,02Ah,02Ah,02Ah,01Eh,000h,020h,0FCh,022h,002h
2554:      DB  004h,000h,000h,000h,000h,000h,000h,000h,0FEh,040h
2555:      DB  030h,040h,0FEh,000h,01Ch,02Ah,02Ah,02Ah,018h,000h
2556:      DB  03Eh,010h,020h,020h,01Eh,000h,000h,022h,0BEh,002h
2557:      DB  000h,000h,0FEh,008h,014h,022h,000h,000h,03Eh,020h
2558:      DB  018h,020h,01Eh,000h,004h,02Ah,02Ah,02Ah,01Eh,000h
2559:      DB  020h,0FCh,022h,002h,004h,000h,000h,022h,0BEh,002h
2560:      DB  000h,000h,000h,000h,000h,000h,000h,000h,0FEh,090h
2561:      DB  090h,090h,060h,000h,01Ch,02Ah,02Ah,02Ah,018h,000h
2562:      DB  03Eh,010h,020h,020h,010h,000h,004h,002h,022h,0BCh
2563:      DB  000h,000h,004h,02Ah,02Ah,02Ah,01Eh,000h,000h,082h
2564:      DB  0FEh,002h,000h,000h,004h,02Ah,02Ah,02Ah,01Eh,000h
2565:      DB  03Eh,010h,020h,020h,01Eh,000h,004h,02Ah,02Ah,02Ah
2566:      DB  01Eh,000h,03Eh,010h,020h,020h,01Eh,000h,000h,000h
2567:      DB  000h,000h,000h,000h,07Eh,090h,090h,090h,07Eh,000h
2568:      DB  03Eh,010h,020h,020h,01Eh,000h,01Ch,022h,022h,012h
2569:      DB  0FEh,000h,004h,02Ah,02Ah,02Ah,01Eh,000h,000h,000h
2570:      DB  000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2571:      DB  000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2572:      DB  000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2573:      DB  000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2574:      DB  000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2575:      ;
2576: ptrrll: DB  000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2577:      DB  000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2578:      DB  000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2579:      DB  000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2580:      DB  000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2581:      DB  0FEh,090h,090h,090h,060h,000h,004h,02Ah,02Ah,02Ah
2582:      DB  01Eh,000h,020h,0FCh,022h,002h,004h,000h,03Ch,002h
2583:      DB  002h,004h,03Eh,000h,0FEh,010h,020h,020h,01Eh,000h
2584:      DB  000h,022h,0BEh,002h,000h,000h,000h,000h,000h,000h
2585:      DB  000h,000h,0FEh,090h,098h,094h,062h,000h,004h,02Ah
2586:      DB  02Ah,02Ah,01Eh,000h,03Eh,020h,018h,020h,01Eh,000h
2587:      DB  0FEh,012h,022h,022h,01Ch,000h,03Ch,002h,002h,004h
2588:      DB  03Eh,000h,010h,010h,010h,010h,010h,000h,0FEh,090h
2589:      DB  098h,094h,062h,000h,004h,02Ah,02Ah,02Ah,01Eh,000h
2590:      DB  03Eh,020h,018h,020h,01Eh,000h,0FEh,012h,022h,022h
2591:      DB  01Ch,000h,03Ch,002h,002h,004h,03Eh,000h,000h,000h
2592:      DB  000h,000h,000h,000h,0FEh,002h,002h,002h,002h,000h
2593:      DB  004h,02Ah,02Ah,02Ah,01Eh,000h,000h,082h,0FEh,002h
2594:      DB  000h,000h,03Ch,002h,002h,004h,03Eh,000h,000h,082h
2595:      DB  0FEh,002h,000h,000h,000h,022h,0BEh,002h,000h,000h
2596:      DB  03Eh,010h,020h,020h,01Eh,000h,020h,0FCh,022h,002h
2597:      DB  004h,000h,004h,02Ah,02Ah,02Ah,01Eh,000h,012h,02Ah
2598:      DB  02Ah,02Ah,024h,000h,000h,000h,000h,000h,000h,000h
2599:      DB  000h,000h,000h,000h,000h,000h,000h,000h,000h,000h
2600:      DB  000h,000h,000h,000h,000h,000h,000h,000h,000h,000h

```



```

1: ;-----;
2: ;----- Program Untuk Membaca data Suhu, CO, Waktu dan untuk mengirimkan data -----;
3: ;----- ke Mikrokontroler AT89S8252 secara serial-----;
4: ;----- BY : ALI MACHFUDIN 02.17.014-----;
5: ;-----;
6:      org      0h
7:      sjmp     init
8: ;-----;
9: ;      Interrupt Serial      ;
10: ;-----;
11:      org      23h           ; alamat serial interrupt
12:      clr      ES           ; enable serial=0
13:      jnb      RI,$         ; RI=0 maka lompat
14:      clr      RI           ; start interrupt=0
15:      mov      R0,SBUF      ; pindahkan isi SBUF ke R0
16:      setb     ES           ; enable serial=1
17:      reti                    ; proses interrupt selesai
18: ;-----;
19: ;      Equate Definition      ;
20: ;-----;
21:      Slct     Bit P3.2     ; memilih data dari ADC
22:      RegA     Equ 0Ah      ; Register A
23:      RegB     Equ 0Bh      ; Register B
24:      RegC     Equ 0Ch      ; Register C
25:      RegD     Equ 0Dh      ; Register D
26:      Secd     Equ 00h
27:      Mint     Equ 02h
28:      Hour     Equ 04h
29:      Date     Equ 07h
30:      Mont     Equ 08h
31:      Year     Equ 09h
32:      Tgl0     Equ 30h
33:      Bln0     Equ 31h
34:      Thn0     Equ 32h
35:      Jam0     Equ 33h
36:      Mnt0     Equ 34h
37:      Dtk0     Equ 35h
38:      Dly0     Equ 40h
39:      Dly1     Equ 41h
40: ;-----;
41: ;      Program Inisialisai      ;
42: ;-----;
43: init:  acall   jeda         ; Panggil subroutine delay
44:        acall   ser_in       ; Panggil subroutine inisialisasi serial
45:        acall   rtc_in       ; Panggil subroutine inisialisasi RTC
46:        mov     R0,#0        ; isi R0 dengan data 0
47: ;-----;
48: ; program kirim data CO ke mikrokontroler berikutnya ;
49: ;-----;
50: cek0:  cjne    R0,#0F0h,cek1 ; bandingkan R0 dengan F0h jika tidak sama lompat ke cek1

```

```

51:      mov     R0,#0           ; isi R0 dengan data 0
52:      clr     Slct           ; Baca data CO pada ADC
53:      acall   jeda           ; panggil subroutine jeda
54:      acall   jeda           ; panggil subroutine jeda
55:      mov     A,P1           ; simpan data dari P1 pada Accumulator
56:      acall   kr_srl         ; Panggil subroutine kirim serial
57: ;-----;
58: ; Program kirim data suhu ke mikrokontroler berikutnya ;
59: ;-----;
60: cek1:  cjne   R0,#0F1h,cek2  ; bandingkan R0 dengan F0h jika tidak sama lompat ke cek1
61:      mov     R0,#0           ; isi R0 dengan data 0
62:      setb    Slct           ; Baca data suhu pada ADC
63:      acall   jeda           ; panggil subroutine jeda
64:      acall   jeda           ; panggil subroutine jeda
65:      mov     A,P1           ; simpan data dari P1 pada Accumulator
66:      acall   kr_srl         ; Panggil subroutine kirim serial
67: ;-----;
68: ; Program kirim data waktu ke mikrokontroler berikutnya ;
69: ;-----;
70: cek2:  cjne   R0,#0F2h,cek3  ; bandingkan R0 dengan F2h jika tidak sama lompat ke cek3
71:      mov     R0,#0           ; isi R0 dengan data 0
72:      acall   bc_tgl         ; panggil subroutine bc_tgl
73:      acall   kr_srl         ; panggil subroutine kr_srl
74: cek3:  cjne   R0,#0F3h,cek4  ; bandingkan R0 dengan F3h jika tidak sama lompat ke cek4
75:      mov     R0,#0           ; isi R0 dengan data 0
76:      acall   bc_bin         ; panggil subroutine bc_tgl
77:      acall   kr_srl         ; panggil subroutine kr_srl
78: cek4:  cjne   R0,#0F4h,cek5  ; bandingkan R0 dengan F4h jika tidak sama lompat ke cek5
79:      mov     R0,#0           ; isi R0 dengan data 0
80:      acall   bc_thn         ; panggil subroutine bc_tgl
81:      acall   kr_srl         ; panggil subroutine kr_srl
82: cek5:  cjne   R0,#0F5h,cek6  ; bandingkan R0 dengan F5h jika tidak sama lompat ke cek6
83:      mov     R0,#0           ; isi R0 dengan data 0
84:      acall   bc_jam         ; panggil subroutine bc_tgl
85:      acall   kr_srl         ; panggil subroutine kr_srl
86: cek6:  cjne   R0,#0F6h,cek7  ; bandingkan R0 dengan F6h jika tidak sama lompat ke cek7
87:      mov     R0,#0           ; isi R0 dengan data 0
88:      acall   bc_mnt         ; panggil subroutine bc_tgl
89:      acall   kr_srl         ; panggil subroutine kr_srl
90: cek7:  cjne   R0,#0F7h,cek8  ; bandingkan R0 dengan F7h jika tidak sama lompat ke cek8
91:      mov     R0,#0           ; isi R0 dengan data 0
92:      acall   bc_dtk         ; panggil subroutine bc_tgl
93:      acall   kr_srl         ; panggil subroutine kr_srl
94:
95: cek8:  cjne   R0,#0F8h,cek15 ; bandingkan R0 dengan F8h jika tidak sama lompat ke cek15
96:      mov     R0,#0           ; isi R0 dengan data 0
97:      acall   jeda           ; panggil subroutine jeda
98:      mov     A,#0EEh        ; isi A dengan data EEh
99:      acall   kr_srl         ; panggil subroutine kr_srl
100:     mov     R0,#0EEh        ; isi R0 dengan data EEh

```

```

201:         ret                ; kembali ke program utama
202: ;
203: spndtk: mov    DPTR,#RegB    ; isi DPTR dengan RegB
204:         mov    A,#96h        ; isikan alamat 96h ke A(untuk menulis)
205:         movx   @DPTR,A       ; Pindahkan A ke alamat yang ditunjuk oleh DPTR
206:         mov    DPTR,#Secd    ; pindahkan isi alamat Secd0 pada A
207:         mov    A,#00         ; isikan A dengan alamat 00
208:         movx   @DPTR,A       ; Pindahkan A ke alamat yang ditunjuk oleh DPTR
209:         mov    DPTR,#RegB    ; isi DPTR dengan RegB
210:         mov    A,#16h        ; isikan alamat 16h ke A(untuk mengembalikan)
211:         movx   @DPTR,A       ; pindahkan A ke alamat yang ditunjukkan oleh DPTR
212:         ret                ; kembali ke program utama
213: ;-----;
214: ;      Program Inisialisasi Serial      ;
215: ;-----;
216: ser_in: setb   EA            ; Aktifkan semua interrupt
217:         mov    TMOD,#20h     ; isikan TMOD dengan data 20h
218:         mov    TH1,#0F3h    ; isikan TH1 dengan data F3h
219:         setb   TR1          ; start timer1 = 1
220:         mov    SCON,#50h    ; isikan TCON dengan data 50h
221:         setb   ES           ; enable interrupt = 1
222:         ret                ; kembali ke program utama
223: ;-----;
224: ;      Program Kirim data serial      ;
225: ;-----;
226: kr_srl: clr    ES            ; enable serial = 0 (non aktif)
227:         mov    SBUF,A        ; pindahkan isi A ke SBUF
228:         jnb   TI,$          ; tunggu TI sampai berlogika 1 (selesai mengirim)
229:         clr   TI            ; TI = 0
230:         setb   ES           ; enable serial = 1 (normal)
231:         ret                ; kembali ke program utama
232: ;-----;
233: ;      Program Baca Data Waktu Dari RTC      ;
234: ;-----;
235: bc_dtk: mov    DPTR,#Secd    ; isikan DPTR dengan alamat Secd
236:         movx   A,@DPTR      ; isikan A dengan isi dari DPTR (secd)
237:         ret                ; kembali ke program utama
238: ;
239: bc_mnt: mov    DPTR,#Mint    ; isikan DPTR dengan alamat Mint
240:         movx   A,@DPTR      ; isikan A dengan isi dari DPTR (Mint)
241:         ret                ; kembali ke program utama
242: ;
243: bc_jam: mov    DPTR,#Hour    ; isikan DPTR dengan alamat Hour
244:         movx   A,@DPTR      ; isikan A dengan isi dari DPTR (Hour)
245:         ret                ; kembali ke program utama
246: ;
247: bc_tgl: mov    DPTR,#Date    ; isikan DPTR dengan alamat Hour
248:         movx   A,@DPTR      ; isikan A dengan isi dari DPTR (Date)
249:         ret                ; kembali ke program utama
250: ;

```

```

251: bc_bln: mov    DPTR,#Mont    ; isikan DPTR dengan alamat Hour
252:      movx   A,@DPTR        ; isikan A dengan isi dari DPTR (Mont)
253:      ret                      ; kembali ke program utama
254: ;
255: bc_thn: mov    DPTR,#Year    ; isikan DPTR dengan alamat Hour
256:      movx   A,@DPTR        ; isikan A dengan isi dari DPTR (Year)
257:      ret                      ; kembali ke program utama
258: ;-----;
259: ;      Program Inisialisasi RTC      ;
260: ;-----;
261: rtc_in: mov    DPTR,#RegA    ; isi DPTR dengan RegA
262:      mov    A,#20h          ; isikan alamat 20h ke A(osilator on)
263:      movx   @DPTR,A        ; pindahkan A ke alamat yang ditunjukkan oleh DPTR
264:      mov    DPTR,#RegB    ; isi DPTR dengan RegB
265:      mov    A,#16h          ; isikan A dengan alamat 16h
266:      movx   @DPTR,A        ; pindahkan A ke alamat yang ditunjukkan oleh DPTR
267:      ret                      ; kembali ke program utama
268: ;-----;
269: ;      Program Subroutine Delay      ;
270: ;-----;
271: jeda:  djnz   Dly0,$         ; kurangi Dly0 dengan 1 jika belum 0 kembali lagi
272:      ret                      ; kembali ke program utama
273: ;
274: delays: mov    Dly1,#255    ; isi Dly1 dengan 255
275: dlys:  acall  jeda          ; panggil jeda
276:      djnz   Dly1,dlys      ; kurangi Dly1 dengan 1 jika belum 0 lompat ke dlys
277:      ret                      ; kembali ke program utama
278: ;
279:      end

```

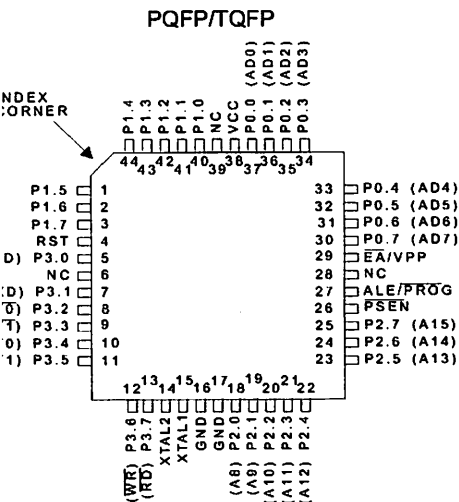
Features

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
- Endurance: 1,000 Write/Erase Cycles
- Supply Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 8 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- 5 Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

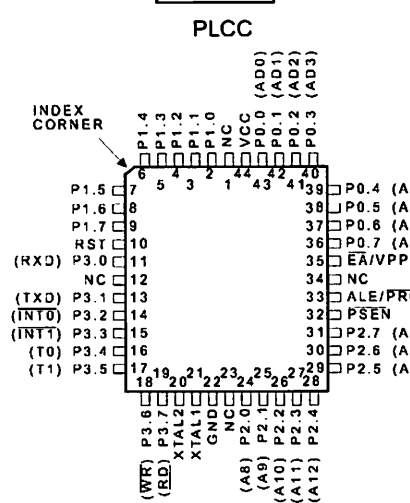
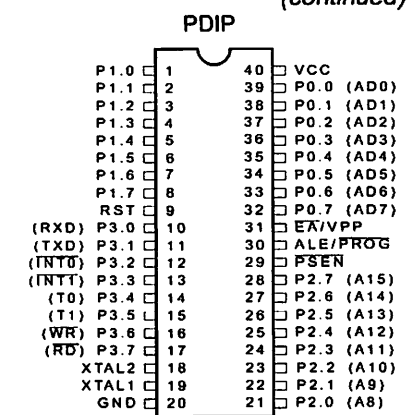
Description

AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K Bytes of Flash Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

Pin Configurations



(continued)



8-Bit Microcontroller with 4K Bytes Flash

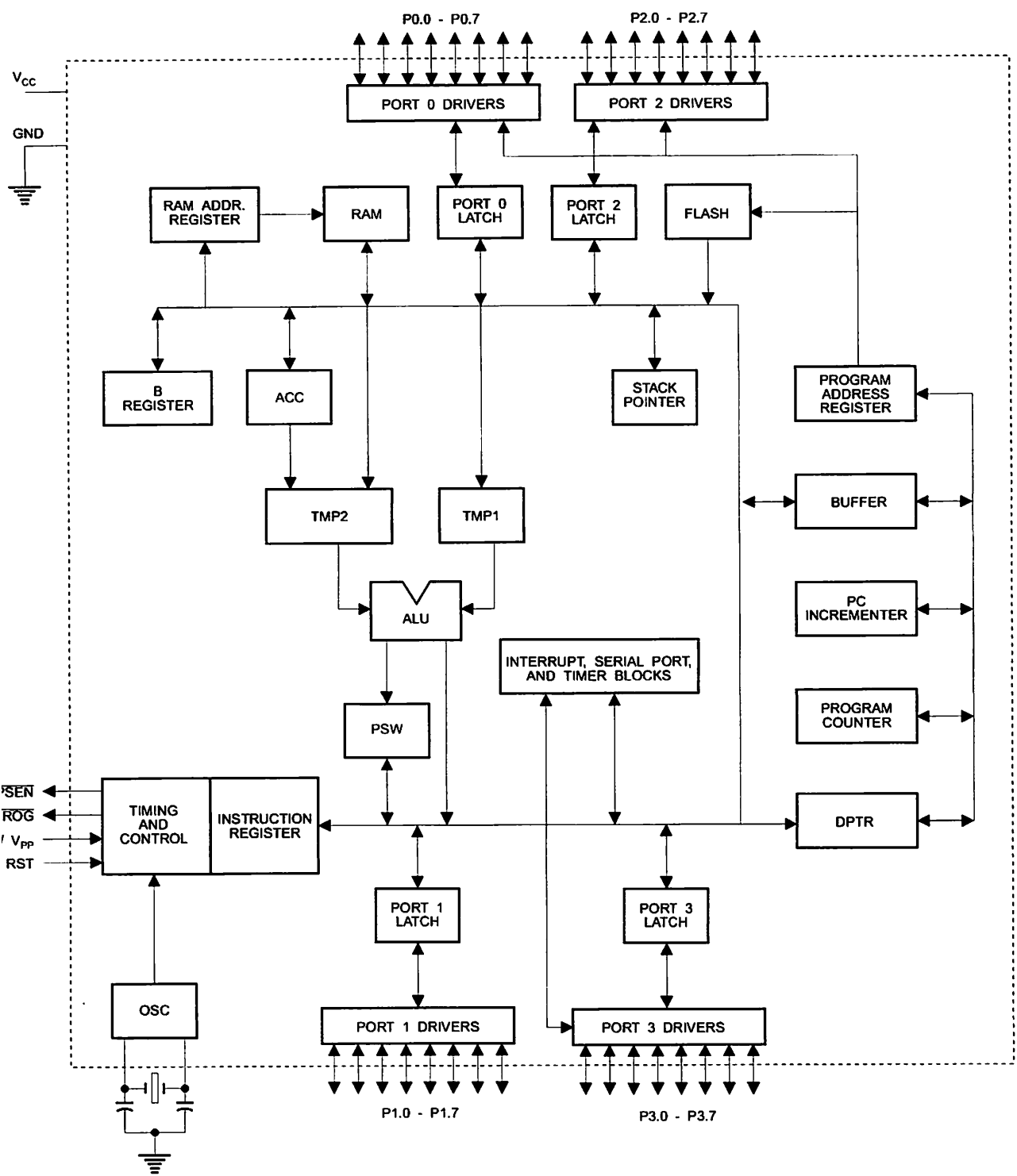
AT89C51

0265F-A-12/97





Diagram



AT89C51

AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next software reset.

Pin Description

V_{CC}
Supply voltage.

GND
Ground.

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

Port 0 may also be configured to be the multiplexed lower address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

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

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

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

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

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application it uses strong internal pullups

when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

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

Port 3

Port 3 is an 8-bit 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 AT89C51 as listed below:

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

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

RST

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

ALE/ \overline{PROG}

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

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

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

\overline{PSEN}

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



When the AT89C51 is executing code from external program memory, \overline{PSEN} is activated twice each machine cycle, except that two \overline{PSEN} activations are skipped during each access to external data memory.

V_{PP}
Internal Access Enable. \overline{EA} must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, \overline{EA} will be normally 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, for parts that require 12-volt V_{PP} .

AL1
 Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

AL2
 Output from the inverting oscillator amplifier.

Oscillator Characteristics

AL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry goes through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be served.

Idle Mode

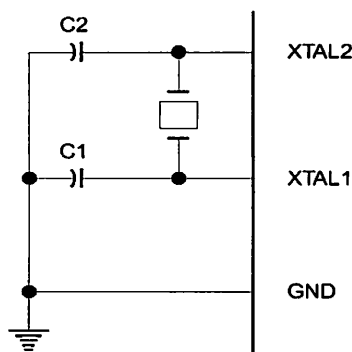
In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware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

Status of External Pins During Idle and Power Down Modes

| Mode | Program Memory | ALE | \overline{PSEN} | PORT0 | PORT1 | PORT2 | PORT3 |
|------------|----------------|-----|-------------------|-------|-------|---------|-------|
| Idle | Internal | 1 | 1 | Data | Data | Data | Data |
| Idle | External | 1 | 1 | Float | Data | Address | Data |
| Power Down | Internal | 0 | 0 | Data | Data | Data | Data |
| Power Down | External | 0 | 0 | Float | Data | Data | Data |

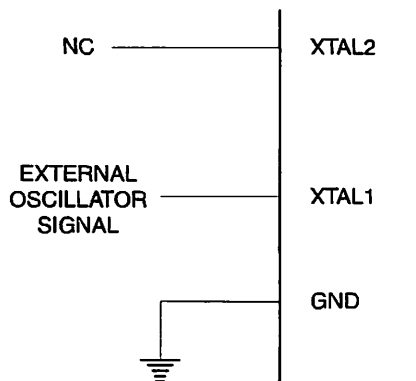
It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port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

Figure 1. Oscillator Connections



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

Figure 2. External Clock Drive Configuration



Power Down Mode

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

Program Memory Lock Bit Protection Modes

| | Program Lock Bits | | | Protection Type |
|---|-------------------|-----|-----|--|
| | LB1 | LB2 | LB3 | |
| 1 | U | U | U | No program lock features. |
| 2 | P | U | U | MOV _C instructions executed from external program memory are disabled from fetching code bytes from internal memory, \overline{EA} is sampled and latched on reset, and further programming of the Flash is disabled. |
| 3 | P | P | U | Same as mode 2, also verify is disabled. |
| 4 | P | P | P | Same as mode 3, also external execution is disabled. |

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

| | $V_{PP} = 12V$ | $V_{PP} = 5V$ |
|---------------|--|--|
| Top-Side Mark | AT89C51 xxxx yyww | AT89C51 xxxx-5 yyww |
| Signature | (030H)=1EH (031H)=51H (032H)=FFH | (030H)=1EH (031H)=51H (032H)=05H |

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-link byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Program Memory Lock Bits

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

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

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figures 3 and 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise \overline{EA}/V_{PP} to 12V for the high-voltage programming mode.
5. Pulse ALE/\overline{PROG} once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. 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 AT89C51 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. 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. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.



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

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

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of operations 030H,

031H, and 032H, except that P3.6 and P3.7 must be pulled to logic low. The values returned are as follows.

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

Programming Interface

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

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

Flash Programming Modes

| Mode | RST | PSEN | ALE/PROG | EA/V _{pp} | P2.6 | P2.7 | P3.6 | P3.7 |
|---------------------|---------|------|----------|--------------------|-------|------|------|------|
| Write Code Data | H | L | | H/12V | L | H | H | H |
| Read Code Data | H | L | H | H | L | L | H | H |
| Write Lock | Bit - 1 | H | L | | H/12V | H | H | H |
| | Bit - 2 | H | L | | H/12V | H | H | L |
| | Bit - 3 | H | L | | H/12V | H | L | L |
| Chip Erase | H | L | (1) | H/12V | H | L | L | L |
| Read Signature Byte | H | L | H | H | L | L | L | L |

Note: 1. Chip Erase requires a 10-ms PROG pulse.

Figure 3. Programming the Flash

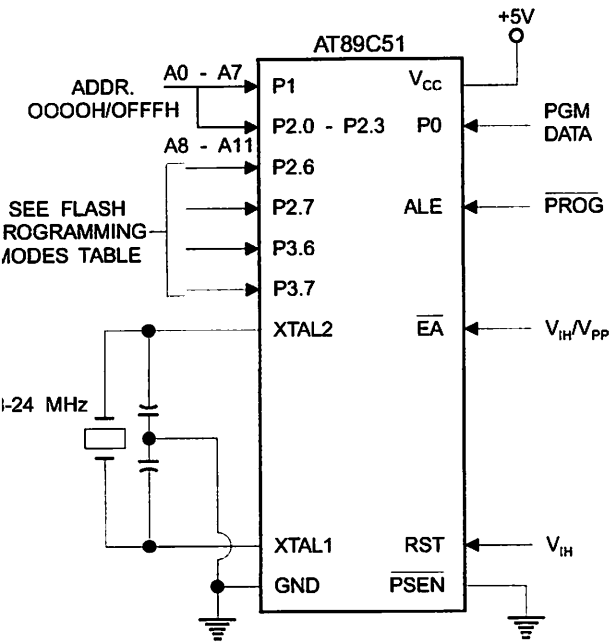
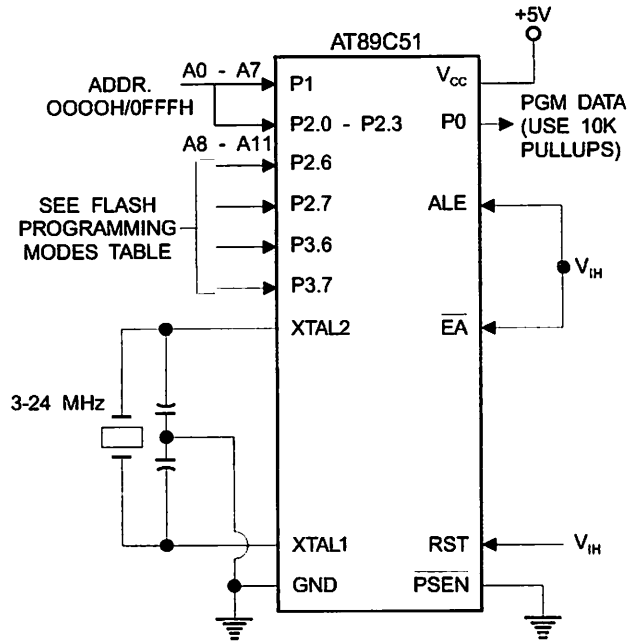


Figure 4. Verifying the Flash



Flash Programming and Verification Characteristics

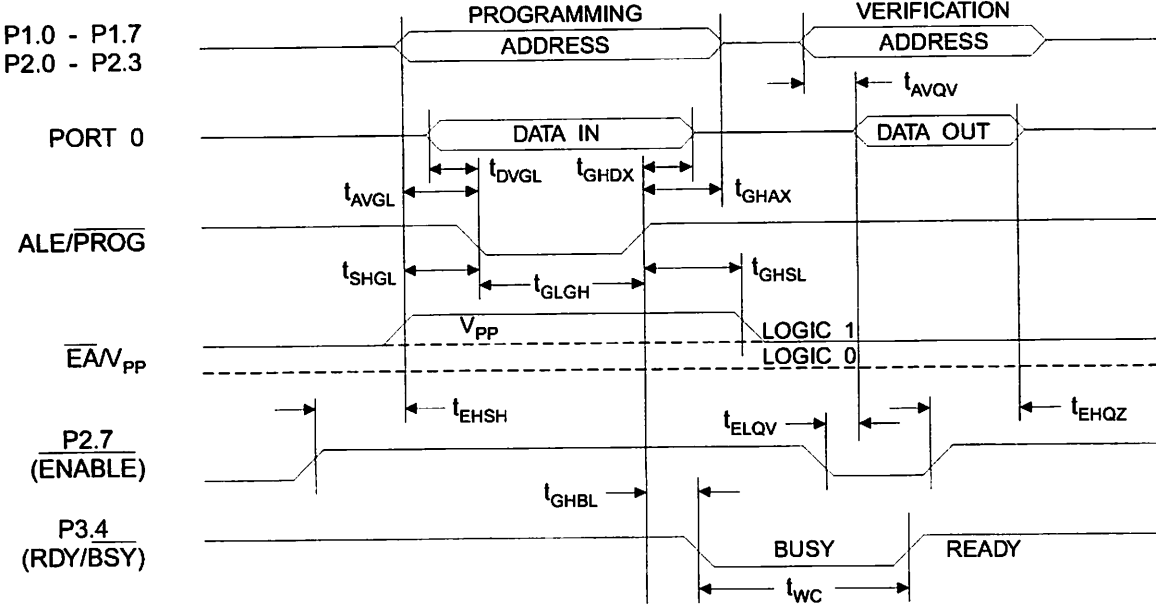
$T = 0^{\circ}\text{C to } 70^{\circ}\text{C}, V_{CC} = 5.0 \pm 10\%$

| Symbol | Parameter | Min | Max | Units |
|-----------------|---|--------------|--------------|---------------|
| $V_{PP}^{(1)}$ | Programming Enable Voltage | 11.5 | 12.5 | V |
| $I_{PP}^{(1)}$ | Programming Enable Current | | 1.0 | mA |
| f_{CLCL} | Oscillator Frequency | 3 | 24 | MHz |
| t_{VGL} | Address Setup to $\overline{\text{PROG}}$ Low | $48t_{CLCL}$ | | |
| t_{HAX} | Address Hold After $\overline{\text{PROG}}$ | $48t_{CLCL}$ | | |
| t_{VGL} | Data Setup to $\overline{\text{PROG}}$ Low | $48t_{CLCL}$ | | |
| t_{HDX} | Data Hold After $\overline{\text{PROG}}$ | $48t_{CLCL}$ | | |
| t_{HSH} | P2.7 ($\overline{\text{ENABLE}}$) High to V_{PP} | $48t_{CLCL}$ | | |
| t_{HGL} | V_{PP} Setup to $\overline{\text{PROG}}$ Low | 10 | | μs |
| $t_{HSL}^{(1)}$ | V_{PP} Hold After $\overline{\text{PROG}}$ | 10 | | μs |
| t_{PLGH} | $\overline{\text{PROG}}$ Width | 1 | 110 | μs |
| t_{WQV} | Address to Data Valid | | $48t_{CLCL}$ | |
| t_{ELQV} | $\overline{\text{ENABLE}}$ Low to Data Valid | | $48t_{CLCL}$ | |
| t_{FHQZ} | Data Float After $\overline{\text{ENABLE}}$ | 0 | $48t_{CLCL}$ | |
| t_{HBL} | $\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low | | 1.0 | μs |
| t_{WC} | Byte Write Cycle Time | | 2.0 | ms |

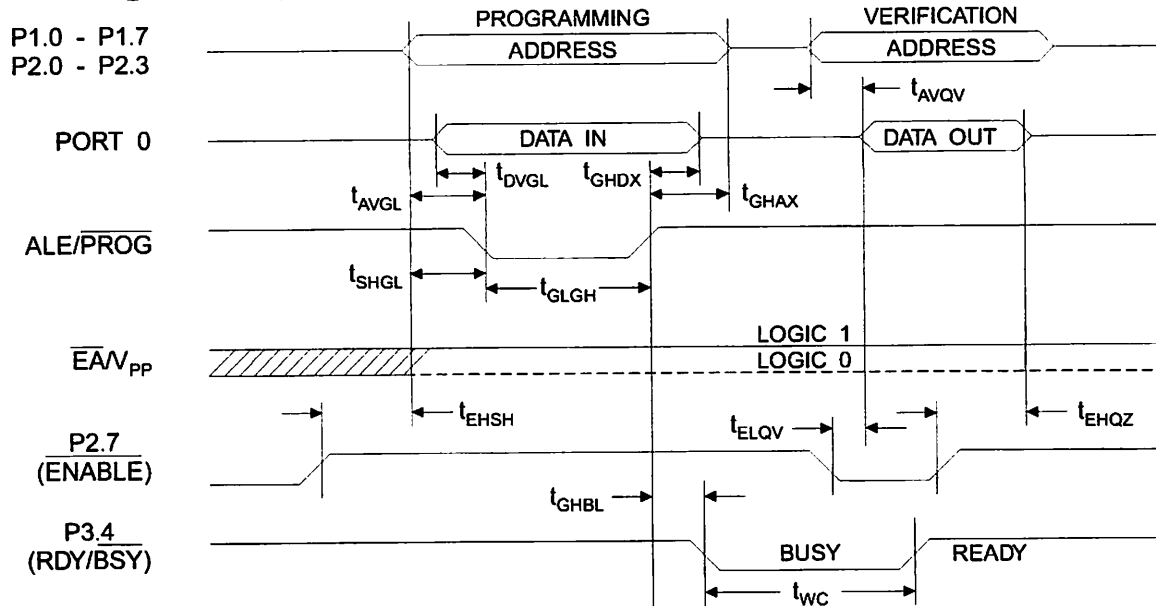
Note: 1. Only used in 12-volt programming mode.



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



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



Absolute Maximum Ratings*

| | |
|---|-----------------|
| Operating Temperature..... | -55°C to +125°C |
| Storage Temperature..... | -65°C to +150°C |
| Voltage on Any Pin with Respect to Ground..... | -1.0V to +7.0V |
| Maximum Operating Voltage..... | 6.6V |
| Maximum Output Current..... | 15.0 mA |

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Characteristics

$T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 5.0\text{V} \pm 20\%$ (unless otherwise noted)

| Symbol | 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 |
| I_{OL} | Output Low Voltage ⁽¹⁾ (Ports 1,2,3) | $I_{OL} = 1.6 \text{ mA}$ | | 0.45 | V |
| I_{OL1} | Output Low Voltage ⁽¹⁾ (Port 0, ALE, \overline{PSEN}) | $I_{OL} = 3.2 \text{ mA}$ | | 0.45 | V |
| I_{OH} | Output High Voltage (Ports 1,2,3, ALE, \overline{PSEN}) | $I_{OH} = -60 \mu\text{A}$, $V_{CC} = 5\text{V} \pm 10\%$ | 2.4 | | V |
| | | $I_{OH} = -25 \mu\text{A}$ | $0.75 V_{CC}$ | | V |
| | | $I_{OH} = -10 \mu\text{A}$ | $0.9 V_{CC}$ | | V |
| I_{OH1} | Output High Voltage (Port 0 in External Bus Mode) | $I_{OH} = -800 \mu\text{A}$, $V_{CC} = 5\text{V} \pm 10\%$ | 2.4 | | V |
| | | $I_{OH} = -300 \mu\text{A}$ | $0.75 V_{CC}$ | | V |
| | | $I_{OH} = -80 \mu\text{A}$ | $0.9 V_{CC}$ | | V |
| | Logical 0 Input Current (Ports 1,2,3) | $V_{IN} = 0.45\text{V}$ | | -50 | μA |
| | Logical 1 to 0 Transition Current (Ports 1,2,3) | $V_{IN} = 2\text{V}$, $V_{CC} = 5\text{V} \pm 10\%$ | | -650 | μA |
| | Input Leakage Current (Port 0, \overline{EA}) | $0.45 < V_{IN} < V_{CC}$ | | ± 10 | μA |
| RST | Reset Pulldown Resistor | | 50 | 300 | $\text{K}\Omega$ |
| CO | Pin Capacitance | Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$ | | 10 | pF |
| IC | Power Supply Current | Active Mode, 12 MHz | | 20 | mA |
| | | Idle Mode, 12 MHz | | 5 | mA |
| | Power Down Mode ⁽²⁾ | $V_{CC} = 6\text{V}$ | | 100 | μA |
| | | $V_{CC} = 3\text{V}$ | | 40 | μA |

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:

Maximum I_{OL} per port pin: 10 mA

Maximum I_{OL} per 8-bit port: Port 0: 26 mA

Ports 1, 2, 3: 15 mA

Maximum total I_{OL} for all output pins: 71 mA

If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum V_{CC} for Power Down is 2V.





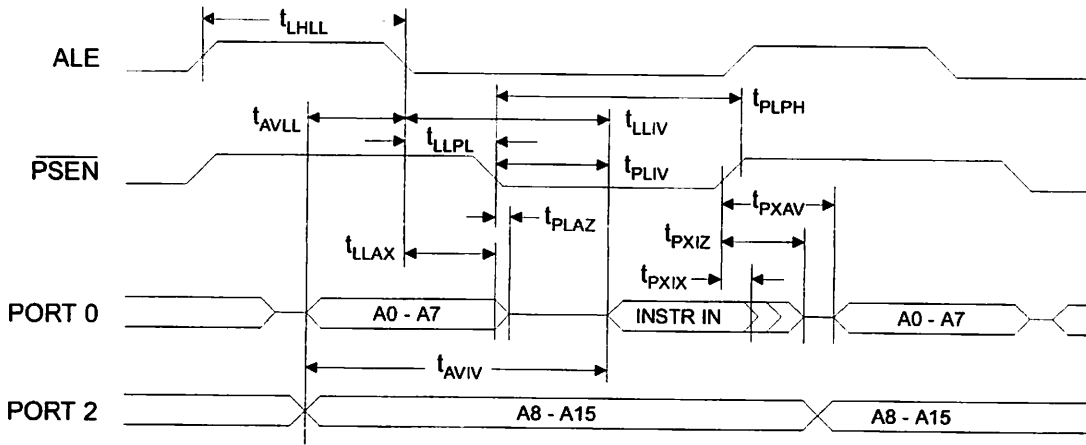
Characteristics

Under Operating Conditions; Load Capacitance for Port 0, ALE/ $\overline{\text{PROG}}$, and $\overline{\text{PSEN}}$ = 100 pF; Load Capacitance for all other ports = 80 pF)

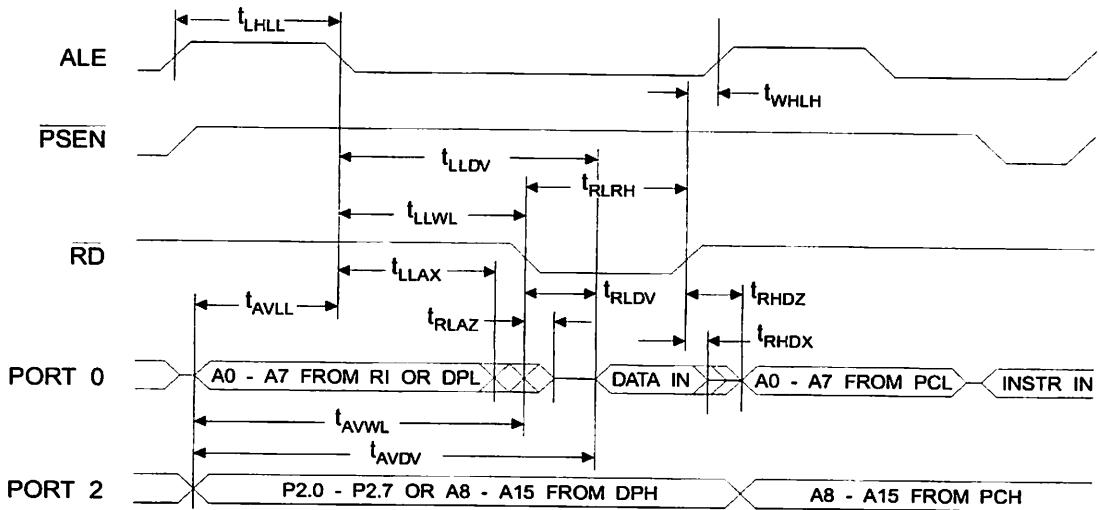
Internal Program and Data Memory Characteristics

| Symbol | Parameter | 12 MHz Oscillator | | 16 to 24 MHz Oscillator | | Units |
|--------|---|-------------------|-----|-------------------------|------------------------|-------|
| | | Min | Max | Min | Max | |
| CLCL | Oscillator Frequency | | | 0 | 24 | MHz |
| AL | ALE Pulse Width | 127 | | $2t_{\text{CLCL}}-40$ | | ns |
| AVL | Address Valid to ALE Low | 43 | | $t_{\text{CLCL}}-13$ | | ns |
| AHX | Address Hold After ALE Low | 48 | | $t_{\text{CLCL}}-20$ | | ns |
| AV | ALE Low to Valid Instruction In | | 233 | | $4t_{\text{CLCL}}-65$ | ns |
| APL | ALE Low to $\overline{\text{PSEN}}$ Low | 43 | | $t_{\text{CLCL}}-13$ | | ns |
| APH | $\overline{\text{PSEN}}$ Pulse Width | 205 | | $3t_{\text{CLCL}}-20$ | | ns |
| APV | $\overline{\text{PSEN}}$ Low to Valid Instruction In | | 145 | | $3t_{\text{CLCL}}-45$ | ns |
| AIHX | Input Instruction Hold After $\overline{\text{PSEN}}$ | 0 | | 0 | | ns |
| AIIZ | Input Instruction Float After $\overline{\text{PSEN}}$ | | 59 | | $t_{\text{CLCL}}-10$ | ns |
| APAV | $\overline{\text{PSEN}}$ to Address Valid | 75 | | $t_{\text{CLCL}}-8$ | | ns |
| AV | Address to Valid Instruction In | | 312 | | $5t_{\text{CLCL}}-55$ | ns |
| APAZ | $\overline{\text{PSEN}}$ Low to Address Float | | 10 | | 10 | ns |
| ARDH | $\overline{\text{RD}}$ Pulse Width | 400 | | $6t_{\text{CLCL}}-100$ | | ns |
| ARDWH | $\overline{\text{WR}}$ Pulse Width | 400 | | $6t_{\text{CLCL}}-100$ | | ns |
| ARDV | $\overline{\text{RD}}$ Low to Valid Data In | | 252 | | $5t_{\text{CLCL}}-90$ | ns |
| ARDX | Data Hold After $\overline{\text{RD}}$ | 0 | | 0 | | ns |
| ARDZ | Data Float After $\overline{\text{RD}}$ | | 97 | | $2t_{\text{CLCL}}-28$ | ns |
| ARDV | ALE Low to Valid Data In | | 517 | | $8t_{\text{CLCL}}-150$ | ns |
| ARDV | Address to Valid Data In | | 585 | | $9t_{\text{CLCL}}-165$ | ns |
| ARDVL | ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low | 200 | 300 | $3t_{\text{CLCL}}-50$ | $3t_{\text{CLCL}}+50$ | ns |
| ARDVL | Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low | 203 | | $4t_{\text{CLCL}}-75$ | | ns |
| ARDWX | Data Valid to $\overline{\text{WR}}$ Transition | 23 | | $t_{\text{CLCL}}-20$ | | ns |
| ARDWH | Data Valid to $\overline{\text{WR}}$ High | 433 | | $7t_{\text{CLCL}}-120$ | | ns |
| ARDQX | Data Hold After $\overline{\text{WR}}$ | 33 | | $t_{\text{CLCL}}-20$ | | ns |
| ARDVZ | $\overline{\text{RD}}$ Low to Address Float | | 0 | | 0 | ns |
| ARDLH | $\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High | 43 | 123 | $t_{\text{CLCL}}-20$ | $t_{\text{CLCL}}+25$ | ns |

Internal Program Memory Read Cycle

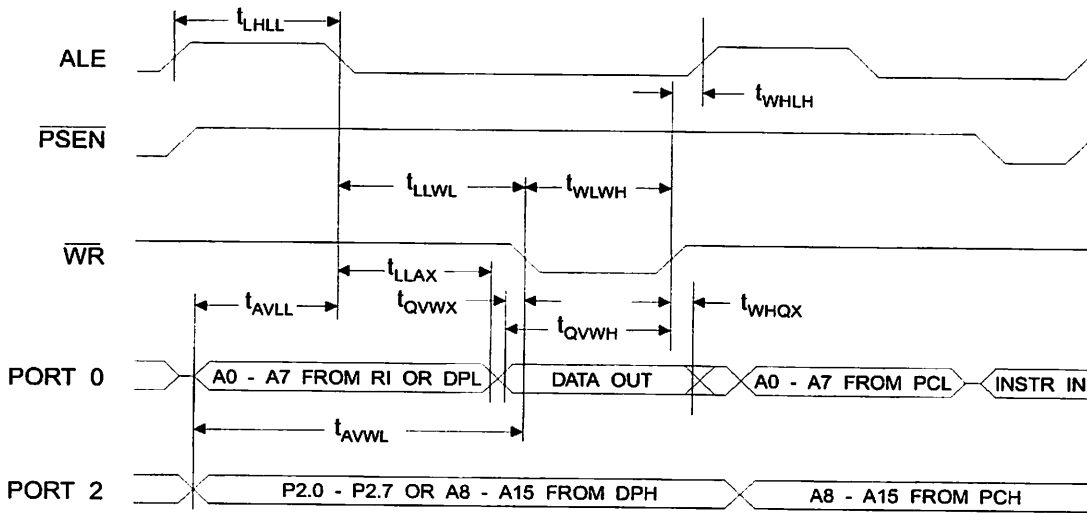


Internal Data Memory Read Cycle

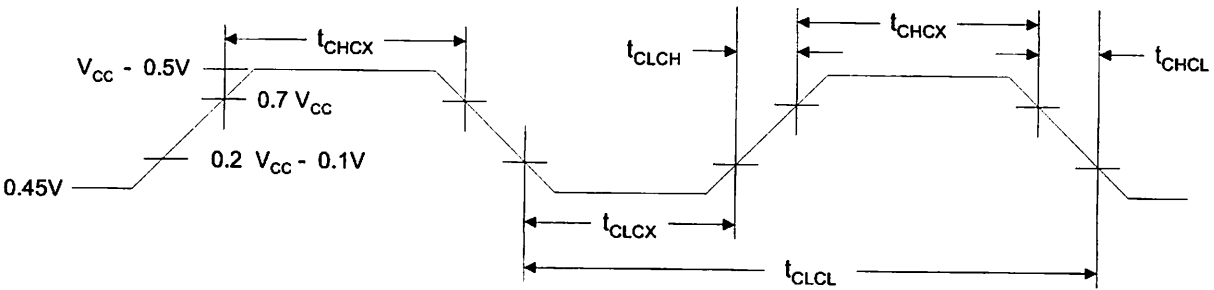




External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

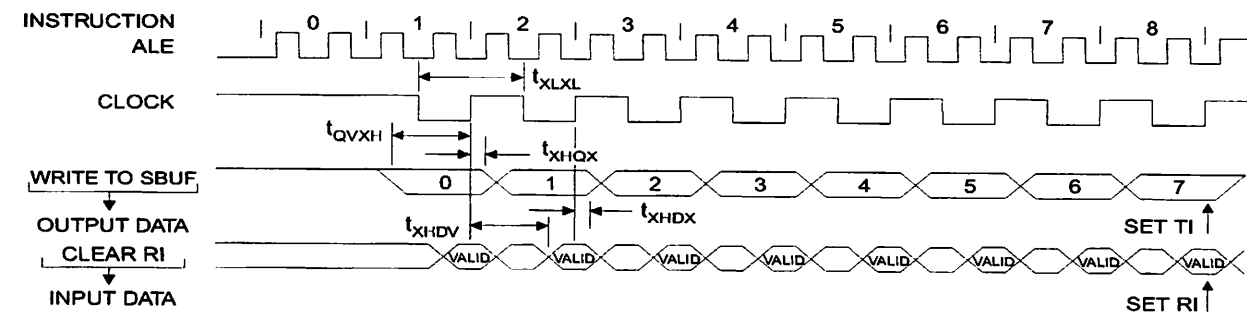
| Symbol | Parameter | Min | Max | Units |
|--------|----------------------|------|-----|-------|
| CLCL | Oscillator Frequency | 0 | 24 | MHz |
| CL | Clock Period | 41.6 | | ns |
| CHCX | High Time | 15 | | ns |
| CLCX | Low Time | 15 | | ns |
| CH | Rise Time | | 20 | ns |
| CL | Fall Time | | 20 | ns |

Serial Port Timing: Shift Register Mode Test Conditions

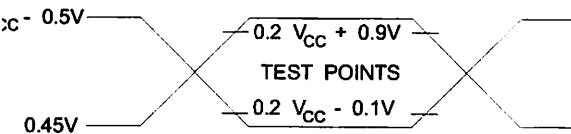
$V_{CC} = 5.0\text{ V} \pm 20\%$; Load Capacitance = 80 pF)

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

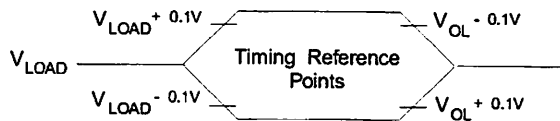
Shift Register Mode Timing Waveforms



Testing Input/Output Waveforms⁽¹⁾



Float Waveforms⁽¹⁾



- AC Inputs during testing are driven at $V_{CC} - 0.5\text{V}$ 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.

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

FEATURES

- Drop-in replacement for IBM AT computer clock/calendar
- Pin compatible with the MC146818B and DS1287
- Totally nonvolatile with over 10 years of operation in the absence of power
- Self-contained subsystem includes lithium, quartz, and support circuitry.
- Counts seconds, minutes, hours, days, day of the week, date, month, and year with leap year compensation valid up to 2100
- Binary or BCD representation of time, calendar, and alarm
- 12- or 24-hour clock with AM and PM in 12-hour mode
- Daylight Savings Time option
- Selectable between Motorola and Intel bus timing
- Multiplex bus for pin efficiency
- Interfaced with software as 128 RAM locations
 - 15 bytes of clock and control registers
 - 113 bytes of general purpose RAM
- Programmable square wave output signal
- Bus-compatible interrupt signals (IRQ)
- Three interrupts are separately software maskable and testable
 - Time-of-day alarm once/second to once/day
 - Periodic rates from 122 ms to 500 ms
 - End of clock update cycle
- Century register

PIN ASSIGNMENT

| | | | |
|-----|----|----|-----------------|
| MOT | 1 | 24 | V _{CC} |
| NC | 2 | 23 | SQW |
| NC | 3 | 22 | NC |
| AD0 | 4 | 21 | NC |
| AD1 | 5 | 20 | NC |
| AD2 | 6 | 19 | IRQ |
| AD3 | 7 | 18 | RESET |
| AD4 | 8 | 17 | DS |
| AD5 | 9 | 16 | NC |
| AD6 | 10 | 15 | R/W |
| AD7 | 11 | 14 | AS |
| GND | 12 | 13 | CS |

DS12C887 24-Pin
ENCAPSULATED PACKAGE

PIN DESCRIPTION

- AD0-AD7 - Multiplexed Address/Data Bus
- NC - No Connect
- MOT - Bus Type Selection
- CS - RTC Chip Select Input
- AS - Address Strobe
- R/W - Read/Write Input
- DS - Data Strobe
- RESET - Reset Input
- IRQ - Interrupt Request Output
- SQW - Square Wave Output
- V_{CC} - +5 Volt Main Supply
- GND - Ground

DESCRIPTION

The DS12C887 Real Time Clock plus RAM is designed as a direct upgrade replacement for the DS12887 existing IBM compatible personal computers to add hardware year 2000 compliance. A century byte is added to memory location 50, 32h, as called out by the PC AT specification. A lithium energy source, quartz crystal, and write-protection circuitry are contained within a 24-pin dual in-line package. In such, the DS12C887 is a complete subsystem replacing 16 components in a typical application. The functions include a nonvolatile time-of-day clock, an alarm, a one-hundred-year calendar, programmable interrupt, square wave generator, and 113 bytes of nonvolatile static RAM. The real time clock is inactive in that time-of-day and memory are maintained even in the absence of power.

OPERATION

The block diagram in Figure 1 shows the pin connections with the major internal functions of the DS12C887. The following paragraphs describe the function of each pin.

SIGNAL DESCRIPTIONS

V_{CC}, V_{CC} - DC power is provided to the device on these pins. V_{CC} is the +5 volt input. When 5 volts are applied within normal limits, the device is fully accessible and data can be written and read. When V_{CC} is below 4.25 volts typical, reads and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage. As V_{CC} falls below 3 volts typical, the RAM and timekeeper are switched over to an internal lithium energy source. The timekeeping function maintains an accuracy of ±1 minute per month at 25°C regardless of the voltage input on the V_{CC} pin.

MOT (Mode Select) – The MOT pin offers the flexibility to choose between two bus types. When connected to V_{CC}, Motorola bus timing is selected. When connected to GND or left disconnected, Intel bus timing is selected. The pin has an internal pull-down resistance of approximately 20KΩ.

SQW (Square Wave Output) – The SQW pin can output a signal from one of 13 taps provided by the 5 internal divider stages of the Real Time Clock. The frequency of the SQW pin can be changed by programming Register A as shown in Table 1. The SQW signal can be turned on and off using the SQWE bit in Register B. The SQW signal is not available when V_{CC} is less than 4.25 volts typical.

AD0-AD7 (Multiplexed Bidirectional Address/Data Bus) – Multiplexed buses save pins because address information and data information time share the same signal paths. The addresses are present during the first portion of the bus cycle and the same pins and signal paths are used for data in the second portion of the cycle. Address/data multiplexing does not slow the access time of the DS12C887 since the bus change from address to data occurs during the internal RAM access time. Addresses must be valid prior to the falling edge of AS/ALE, at which time the DS12C887 latches the address from AD0 to AD6. Valid write data must be present and held stable during the latter portion of the DS or WR pulses. In a read cycle the DS12C887 outputs 8 bits of data during the latter portion of the DS or RD pulses. The read cycle is terminated and the bus returns to a high impedance state as DS transitions low in the case of Motorola timing or as RD transitions high in the case of Intel timing.

AS (Address Strobe Input) – A positive going address strobe pulse serves to demultiplex the bus. The rising edge of AS/ALE causes the address to be latched within the DS12C887. The next rising edge that occurs on the AS bus will clear the address regardless of whether \overline{CS} is asserted. Access commands could be sent in pairs

DS (Data Strobe or Read Input) – The DS/RD pin has two modes of operation depending on the level of the MOT pin. When the MOT pin is connected to V_{CC}, Motorola bus timing is selected. In this mode DS is a positive pulse during the latter portion of the bus cycle and is called Data Strobe. During read cycles, DS signifies the time that the DS12C887 is to drive the bidirectional bus. In write cycles the falling edge of DS causes the DS12C887 to latch the written data. When the MOT pin is connected to GND, Intel bus timing is selected. In this mode the DS pin is called Read(RD). RD identifies the time period when the DS12C887 drives the bus with read data. The RD signal is the same definition as the Output Enable (OE) signal on a typical memory.

$\overline{R/\overline{W}}$ (Read/Write Input) – The $\overline{R/\overline{W}}$ pin also has two modes of operation. When the MOT pin is connected to V_{CC} for Motorola timing, $\overline{R/\overline{W}}$ is at a level which indicates whether the current cycle is a read or write. A read cycle is indicated with a high level on $\overline{R/\overline{W}}$ while DS is high. A write cycle is indicated when $\overline{R/\overline{W}}$ is low during DS. When the MOT pin is connected to GND for Intel timing, the $\overline{R/\overline{W}}$ signal is an active low signal called WR. In this mode the $\overline{R/\overline{W}}$ pin has the same meaning as the Write Enable signal (WE) on generic RAMs.

\overline{CS} (Chip Select Input) – The Chip Select signal must be asserted low for a bus cycle in the DS12C887 to be accessed. \overline{CS} must be kept in the active state during DS and AS for Motorola timing and during RD and WR for Intel timing. Bus cycles which take place without asserting \overline{CS} will latch addresses but no access will occur. When V_{CC} is below 4.25 volts, the DS12C887 internally inhibits access cycles by internally disabling the \overline{CS} input. This action protects both the real time clock data and RAM data during power outages.

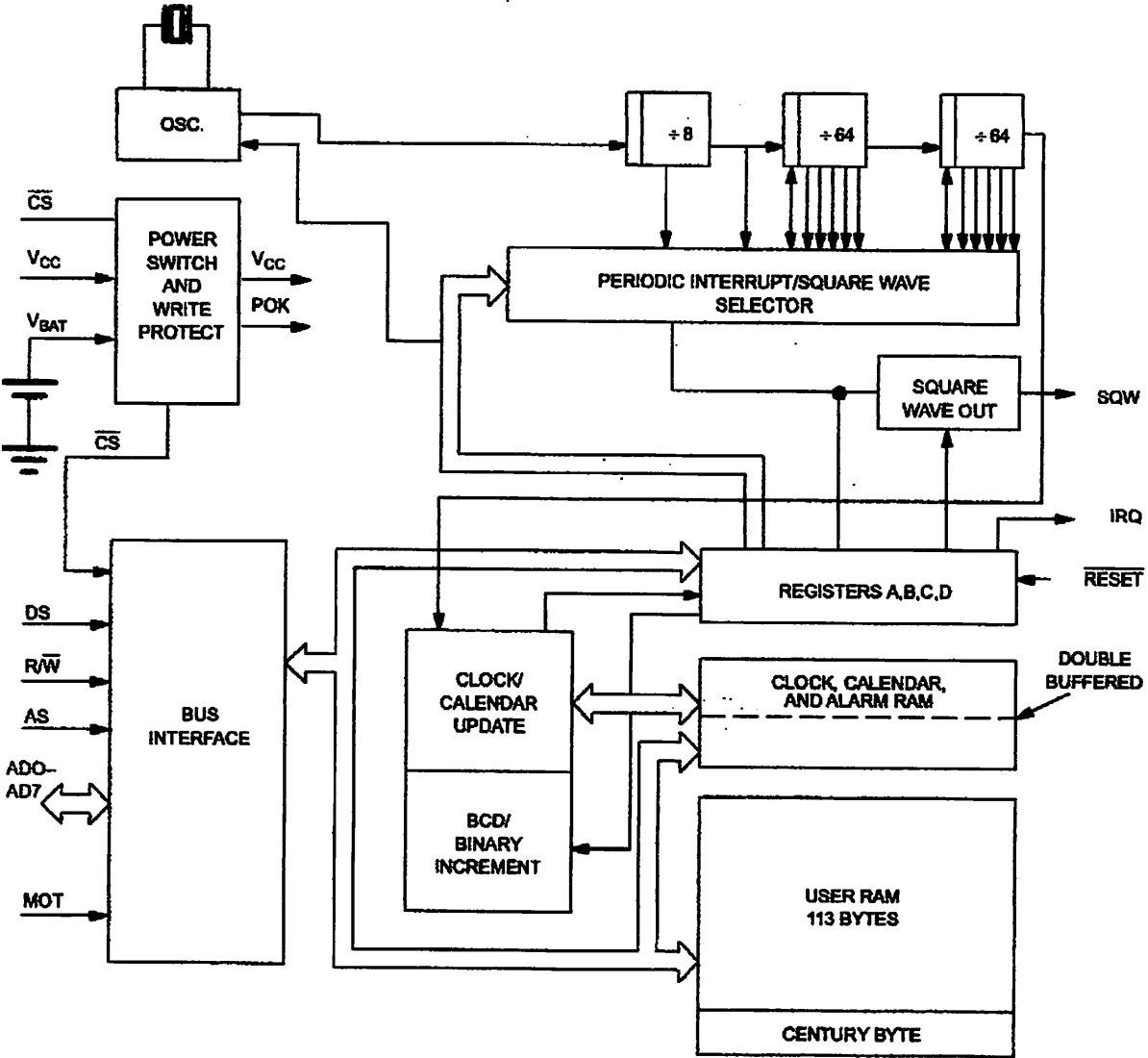
\overline{IRQ} (Interrupt Request Output) - The \overline{IRQ} pin is an active low output of the DS12C887 that can be used as an interrupt input to a processor. The \overline{IRQ} output remains low as long as the status bit causing the interrupt is present and the corresponding interrupt-enable bit is set. To clear the \overline{IRQ} pin the processor program normally reads the C register. The RESET pin also clears pending interrupts. When no interrupt conditions are present, the \overline{IRQ} level is in the high impedance state. Multiple interrupting devices can be connected to an \overline{IRQ} bus. The \overline{IRQ} bus is an open drain output and requires an external pull-up resistor.

\overline{RESET} (Reset Input) – The \overline{RESET} pin has no effect on the clock, calendar, or RAM. On power-up the \overline{RESET} pin can be held low for a time in order to allow the power supply to stabilize. The amount of time that \overline{RESET} is held low is dependent on the application. However, if \overline{RESET} is used on power-up, the time \overline{RESET} is low should exceed 200 ms to make sure that the internal timer that controls the DS12C887 on power-up has timed out. When \overline{RESET} is low and V_{CC} is above 4.25 volts, the following occurs:

- A. Periodic Interrupt Enable (PEI) bit is cleared to zero.
- B. Alarm Interrupt Enable (AIE) bit is cleared to zero.
- C. Update Ended Interrupt Flag (UF) bit is cleared to zero.
- D. Interrupt Request Status Flag (IRQF) bit is cleared to zero.
- E. Periodic Interrupt Flag (PF) bit is cleared to zero.
- F. The device is not accessible until \overline{RESET} is returned high.
- G. Alarm Interrupt Flag (AF) bit is cleared to zero.
- H. \overline{IRQ} pin is in the high impedance state.
- I. Square Wave Output Enable (SQWE) bit is cleared to zero.
- J. Update Ended Interrupt Enable (UIE) is cleared to zero.

In a typical application \overline{RESET} can be connected to V_{CC} . This connection will allow the DS12C887 to go in and out of power fail without affecting any of the control registers.

S12C887 BLOCK DIAGRAM Figure 1



POWER-DOWN/POWER-UP CONSIDERATIONS

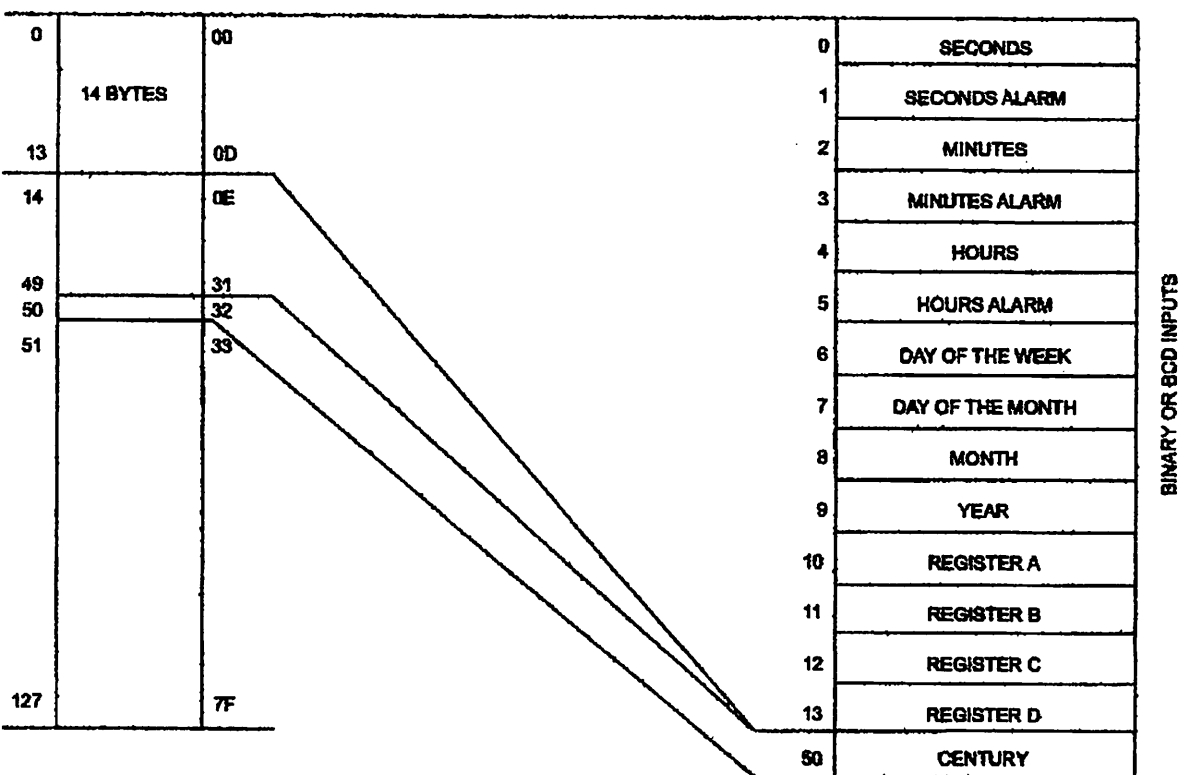
The Real Time Clock function will continue to operate and all of the RAM, time, calendar, and alarm memory locations remain nonvolatile regardless of the level of the V_{CC} input. When V_{CC} is applied to the DS12C887 and reaches a level of greater than 4.25 volts, the device becomes accessible after 200 ms, provided that the oscillator is running and the oscillator countdown chain is not in reset (see Register A). This time period allows the system to stabilize after power is applied. When V_{CC} falls below 4.25 volts, the chip select input is internally forced to an inactive level regardless of the value of \overline{CS} at the input pin. The DS12C887 is, therefore, write-protected. When the DS12C887 is in a write-protected state, all inputs are ignored and all outputs are in a high impedance state. When V_{CC} falls below a level of approximately 1.5 volts, the external V_{CC} supply is switched off and an internal lithium energy source supplies power to the Real Time Clock and the RAM memory.

RTC ADDRESS MAP

The address map for the DS12C887 is shown in Figure 2. The address map consists of 113 bytes of user RAM, 11 bytes of RAM that contain the RTC time, calendar, and alarm data, and 4 bytes which are used for control and status. All 128 bytes can be directly written or read except for the following:

1. Registers C and D are read-only.
2. Bit-7 of Register A is read-only.
3. The high order bit of the seconds byte is read-only.

DS12C887 REAL TIME CLOCK ADDRESS MAP Figure 2



TIME, CALENDAR AND ALARM LOCATIONS

The time and calendar information is obtained by reading the appropriate memory bytes. The time, calendar, and alarm are set or initialized by writing the appropriate RAM bytes. The contents of the ten time, calendar, and alarm bytes can be either Binary or Binary-Coded Decimal (BCD) format. Before writing the internal time, calendar, and alarm registers, the SET bit in Register B should be written to a logic one to prevent updates from occurring while access is being attempted. In addition to writing the ten time, calendar, and alarm registers in a selected format (binary or BCD), the data mode bit (DM) of Register B must be set to the appropriate logic level. All ten time, calendar, and alarm bytes must use the time data mode. The set bit in Register B should be cleared after the data mode bit has been written to allow the real time clock to update the time and calendar bytes. Once initialized, the real time clock makes all updates in the selected mode. The data mode cannot be changed without reinitializing the ten data bytes. Table 2 shows the binary and BCD formats of the ten time, calendar, and alarm locations. The 11-12 bit cannot be changed without reinitializing the hour locations. When the 12-hour format is selected, the high order bit of the hours byte represents PM when it is a logic one. The time, calendar, and alarm bytes are always accessible because they are double buffered. Once per second the eleven bytes are advanced by one second and checked for an alarm condition. If a read of the time and calendar data occurs during an update, a problem exists where seconds, minutes, hours, etc. may not correlate. The probability of reading incorrect time and calendar data is low. Several methods of avoiding any possible incorrect time and calendar reads are covered later in this text. The three alarm bytes can be used in two ways. First, when the alarm time is written in the appropriate hours, minutes, and seconds alarm locations, the alarm interrupt is initiated at the specified time each day if the alarm enable bit is high. The second use condition is to insert a "don't care" state in one or more of the three alarm bytes. The "don't care" code is any hexadecimal value from C0 to FF. The two most significant bits of each byte set the "don't care" condition when at logic 1. An alarm will be generated each hour when the "don't care" bits are set in the hours byte. Similarly, an alarm is generated every minute with "don't care" codes in the hours and minute alarm bytes. The "don't care" codes in all three alarm bytes create an interrupt every second.

TIME, CALENDAR AND ALARM DATA MODES Table 1

| ADDRESS LOCATION | FUNCTION | DECIMAL RANGE | RANGE | |
|------------------|--------------------------|---------------|--------------------|--------------------|
| | | | BINARY DATA MODE | BCD DATA MODE |
| 0 | Seconds | 0-59 | 00-3B | 00-59 |
| 1 | Seconds Alarm | 0-59 | 00-3B | 00-59 |
| 2 | Minutes | 0-59 | 00-3B | 00-59 |
| 3 | Minutes Alarm | 0-59 | 00-3B | 00-59 |
| 4 | Hours 12-hr, Mode | 1-12 | 01-0C AM, 81-8C PM | 01-12 AM, 81-92 PM |
| | Hours 24-hr, Mode | 0-23 | 00-17 | 00-23 |
| 5 | Hours Alarm 12-hr, Mode | 1-12 | 01-0C AM, 81-8C PM | 01-12 AM, 81-92 PM |
| | Hours Alarm 24-hr, Mode | 0-23 | 00-17 | 00-23 |
| 6 | Day of the week Sunday=1 | 1-7 | 01-07 | 01-07 |
| 7 | Date of Month | 1-31 | 01-1F | 01-31 |
| 8 | Month | 1-12 | 01-0C | 01-12 |
| 9 | Year | 0-99 | 00-63 | 00-99 |
| 50 | Century | 0-99 | NA | 19,20 |

CONTROL REGISTERS

The DS12C887 has four control registers which are accessible at all times, even during the update cycle.

REGISTER A**MSB****LSB**

| BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| UIP | DV2 | DV1 | DV0 | RS3 | RS2 | RS1 | RS0 |

UIP - The Update In Progress (UIP) bit is a status flag that can be monitored. When the UIP bit is a 1, the update transfer will soon occur. When UIP is a 0, the update transfer will not occur for at least 244 μ s. The time, calendar, and alarm information in RAM is fully available for access when the UIP bit is 0. The UIP bit is read-only and is not affected by RESET. Writing the SET bit in Register B to a 1 inhibits any update transfer and clears the UIP status bit.

DV2, DV1, DV0 - These three bits are used to turn the oscillator on or off and to reset the countdown chain. A pattern of 010 is the only combination of bits that will turn the oscillator on and allow the RTC to keep time. A pattern of 11X will enable the oscillator but holds the countdown chain in reset. The next update will occur at 500 ms after a pattern of 010 is written to DV0, DV1, and DV2.

RS3, RS2, RS1, RS0 - These four rate-selection bits select one of the 13 taps on the 15-stage divider or disable the divider output. The tap selected can be used to generate an output square wave (SQW pin) and/or a periodic interrupt. The user can do one of the following:

1. Enable the interrupt with the PIE bit;
2. Enable the SQW output pin with the SQWE bit;
3. Enable both at the same time and the same rate; or
4. Enable neither

Table 1 lists the periodic interrupt rates and the square wave frequencies that can be chosen with the RS bits. These four read/write bits are not affected by RESET.

REGISTER B
LSB

| BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| SET | PIE | AIE | UIE | SQWE | DM | 24/12 | DSE |

SET - When the SET bit is a 0, the update transfer functions normally by advancing the counts once per second. When the SET bit is written to a 1, any update transfer is inhibited and the program can initialize the time and calendar bytes without an update occurring in the midst of initializing. Read cycles can be executed in a similar manner. SET is a read/write bit and is not affected by $\overline{\text{RESET}}$ or internal functions of the DS12C887.

PIE - The Periodic Interrupt Enable bit is a read/write bit which allows the Periodic Interrupt Flag (PF) bit in Register C to drive the $\overline{\text{IRQ}}$ pin low. When the PIE bit is set to 1, periodic interrupts are generated by driving the $\overline{\text{IRQ}}$ pin low at a rate specified by the RS3-RS0 bits of Register A. A 0 in the PIE bit blocks the $\overline{\text{IRQ}}$ output from being driven by a periodic interrupt, but the Periodic Flag (PF) bit is still set at the periodic rate. PIE is not modified by any internal DS12C887 functions but is cleared to 0 on $\overline{\text{ESET}}$.

AIE - The Alarm Interrupt Enable (AIE) bit is a read/write bit which, when set to a 1, permits the Alarm Flag (AF) bit in register C to assert $\overline{\text{IRQ}}$. An alarm interrupt occurs for each second that the 3 time bytes equal the 3 alarm bytes including a “don’t care” alarm code of binary 11XXXXXX. When the AIE bit is set to 0, the AF bit does not initiate the $\overline{\text{IRQ}}$ signal. The internal functions of the DS12C887 not affect the AIE bit.

UIE - The Update Ended Interrupt Enable (UIE) bit is a read/write bit that enables the Update End Flag (UEF) bit in Register C to assert $\overline{\text{IRQ}}$. The $\overline{\text{RESET}}$ pin going low or the SET bit going high clears the UIE bit.

SQWE - When the Square Wave Enable (SQWE) bit is set to a 1, a square wave signal at the frequency specified by the rate-selection bits RS3 through RS0 is driven out on the SQW pin. When the SQWE bit is set to 0, the SQW pin is held low. SQWE is a read/write bit and is cleared by $\overline{\text{RESET}}$. SQWE is set to a 1 when V_{CC} is powered up.

DM - The Data Mode (DM) bit indicates whether time and calendar information is in binary or BCD format. The DM bit is set by the program to the appropriate format and can be read as required. This bit is not modified by internal functions or $\overline{\text{RESET}}$. A 1 in DM signifies binary data while a 0 in DM specifies Binary Coded Decimal (BCD) data.

24/12 - The 24/12 control bit establishes the format of the hours byte. A 1 indicates the 24-hour mode and a 0 indicates the 12-hour mode. This bit is read/write and is not affected by internal functions or $\overline{\text{RESET}}$.

DSE - The Daylight Savings Enable (DSE) bit is a read/write bit which enables two special updates when DSE is set to 1. On the first Sunday in April the time increments from 1:59:59 AM to 3:00:00 AM. On the last Sunday in October when the time first reaches 1:59:59 AM it changes to 1:00:00 AM. These special updates do not occur when the DSE bit is a zero. This bit is not affected by internal functions or $\overline{\text{RESET}}$.

REGISTER C**MSB****LSB**

| BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| IRQF | PF | AF | UF | 0 | 0 | 0 | 0 |

IRQF - The Interrupt Request Flag (IRQF) bit is set to a 1 when one or more of the following are true:

PIE = 1

AIE = 1

UIE = 1

$$\text{IRQF} = (\text{PF} \bullet \text{PIE}) + (\text{AF} \bullet \text{AIE}) + (\text{UF} \bullet \text{UIE})$$

Any time the IRQF bit is a 1, the $\overline{\text{IRQ}}$ pin is driven low. Flag bits PF, AF, and UF are cleared after Register C is read by the program or when the $\overline{\text{RESET}}$ pin is low.

PF - The Periodic Interrupt Flag (PF) is a read-only bit which is set to a 1 when an edge is detected on the selected tap of the divider chain. The RS3 through RS0 bits establish the periodic rate. PF is set to a 1 independent of the state of the PIE bit. When both PF and PIE are 1's, the $\overline{\text{IRQ}}$ signal is active and will set the IRQF bit. The PF bit is cleared by a software read of Register C or a $\overline{\text{RESET}}$.

AF - A 1 in the Alarm Interrupt Flag (AF) bit indicates that the current time has matched the alarm time. If the AIE bit is also a 1, the $\overline{\text{IRQ}}$ pin will go low and a 1 will appear in the IRQF bit. A $\overline{\text{RESET}}$ or a read of Register C will clear AF.

UF - The Update Ended Interrupt Flag (UF) bit is set after each update cycle. When the UIE bit is set to a 1, a 1 in UF causes the IRQF bit to be a 1, which will assert the $\overline{\text{IRQ}}$ pin. UF is cleared by reading Register C or a $\overline{\text{RESET}}$.

BIT 3 THROUGH BIT 0 - These are unused bits of the status Register C. These bits always read 0 and cannot be written.

REGISTER D**MSB****LSB**

| BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| VRT | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

VRT - The Valid RAM and Time (VRT) bit indicates the condition of the battery connected to the V_{BAT} pin. This bit is not writeable and should always be a 1 when read. If a 0 is ever present, an exhausted external lithium energy source is indicated and both the contents of the RTC data and RAM data are questionable. This bit is unaffected by $\overline{\text{RESET}}$.

BIT 6 THROUGH BIT 0 - The remaining bits of Register D are not usable. They cannot be written and, when read, they will always read 0.

CENTURY REGISTER

The century register at location 32h, is a BCD register designed to automatically load the BCD value 20 when the year register changes from 99 to 00. The MSB of this register will not be affected when the load of 0 occurs and will remain at the value written by the user.

NONVOLATILE RAM

The 113 general purpose nonvolatile RAM bytes are not dedicated to any special function within the DS12C887. They can be used by the processor program as nonvolatile memory and are fully available during the update cycle.

INTERRUPTS

The RTC plus RAM includes three separate, fully automatic sources of interrupt for a processor. The alarm interrupt can be programmed to occur at rates from once per second to once per day. The periodic interrupt can be selected for rates from 500ms to 122 μ s. The update-ended interrupt can be used to indicate to the program that an update cycle is complete. Each of these independent interrupt conditions is described in greater detail in other sections of this text.

The processor program can select which interrupts, if any, are going to be used. Three bits in Register B enable the interrupts. Writing a logic 1 to an interrupt-enable bit permits that interrupt to be initiated when the event occurs. A zero in an interrupt-enable bit prohibits the $\overline{\text{IRQ}}$ pin from being asserted from that interrupt condition. If an interrupt flag is already set when an interrupt is enabled, $\overline{\text{IRQ}}$ is immediately set at an active level, although the interrupt initiating the event may have occurred much earlier. As a result, there are cases where the program should clear such earlier initiated interrupts before enabling new interrupts.

When an interrupt event occurs, the relating flag bit is set to logic 1 in Register C. These flag bits are set independent of the state of the corresponding enable bit in Register B. The flag bit can be used in polling mode without enabling the corresponding enable bits. The interrupt flag bit is a status bit which software can interrogate as necessary. When a flag is set, an indication is given to software that an interrupt event has occurred since the flag bit was last read; however, care should be taken when using the flag bits as they are cleared each time Register C is read. Double latching is included with Register C so that bits which are set remain stable throughout the read cycle. All bits which are set (high) are cleared when read and new interrupts which are pending during the read cycle are held until after the cycle is completed. One, two, or three bits can be set when reading Register C. Each utilized flag bit should be examined when read to ensure that no interrupts are lost.

The second flag bit usage method is with fully enabled interrupts. When an interrupt flag bit is set and the corresponding interrupt enable bit is also set, the $\overline{\text{IRQ}}$ pin is asserted low. $\overline{\text{IRQ}}$ is asserted as long as at least one of the three interrupt sources has its flag and enable bits both set. The IRQF bit in Register C is set whenever the $\overline{\text{IRQ}}$ pin is being driven low. Determination that the RTC initiated an interrupt is accomplished by reading Register C. A logic one in bit 7 (IRQF bit) indicates that one or more interrupts have been initiated by the DS12C887. The act of reading Register C clears all active flag bits and the IRQF bit.

OSCILLATOR CONTROL BITS

When the DS12C887 is shipped from the factory, the internal oscillator is turned off. This feature prevents the lithium energy cell from being used until it is installed in a system. A pattern of 010 in bits 4 through 6 of Register A will turn the oscillator on and enable the countdown chain. A pattern of 11X will turn the oscillator on, but holds the countdown chain of the oscillator in reset. All other combinations of bits 4 through 6 keep the oscillator off.

SQUARE WAVE OUTPUT SELECTION

Thirteen of the 15 divider taps are made available to a 1-of-15 selector, as shown in the block diagram of Figure 1. The first purpose of selecting a divider tap is to generate a square wave output signal on the SQW pin. The RS0–RS3 bits in Register A establish the square wave output frequency. These frequencies are listed in Table 1. The SQW frequency selection shares its 1-of-15 selector with the periodic interrupt generator. Once the frequency is selected, the output of the SQW pin can be turned on and off under program control with the square wave enable bit (SQWE).

PERIODIC INTERRUPT SELECTION

The periodic interrupt will cause the $\overline{\text{IRQ}}$ pin to go to an active state from once every 500ms to once every 122 μ s. This function is separate from the alarm interrupt which can be output from once per second to once per day. The periodic interrupt rate is selected using the same Register A bits which select the square wave frequency (see Table 1). Changing the Register A bits affects both the square wave frequency and the periodic interrupt output. However, each function has a separate enable bit in Register B. The SQWE bit controls the square wave output. Similarly, the periodic interrupt is enabled by the PIE bit in Register B. The periodic interrupt can be used with software counters to measure inputs, create output intervals, or await the next needed software function.

PERIODIC INTERRUPT RATE AND SQUARE WAVE OUTPUT FREQUENCY

Table 2

| EXT. REG. B E32K | SELECT BITS REGISTER A | | | | t_{PI} PERIODIC INTERRUPT RATE | SQW OUTPUT FREQUENCY |
|---------------------|------------------------|-----|-----|-----|-------------------------------------|-------------------------|
| | RS3 | RS2 | RS1 | RS0 | | |
| 0 | 0 | 0 | 0 | 0 | None | None |
| 0 | 0 | 0 | 0 | 1 | 3.90625 ms | 256 Hz |
| 0 | 0 | 0 | 1 | 0 | 7.8125 ms | 128 Hz |
| 0 | 0 | 0 | 1 | 1 | 122.070 μ s | 8.192 kHz |
| 0 | 0 | 1 | 0 | 0 | 244.141 μ s | 4.096 kHz |
| 0 | 0 | 1 | 0 | 1 | 488.281 μ s | 2.048 kHz |
| 0 | 0 | 1 | 1 | 0 | 976.5625 μ s | 1.024 kHz |
| 0 | 0 | 1 | 1 | 1 | 1.953125 ms | 512 Hz |
| 0 | 1 | 0 | 0 | 0 | 3.90625 ms | 256 Hz |
| 0 | 1 | 0 | 0 | 1 | 7.8125 ms | 128 Hz |
| 0 | 1 | 0 | 1 | 0 | 15.625 ms | 64 Hz |
| 0 | 1 | 0 | 1 | 1 | 31.25 ms | 32 Hz |
| 0 | 1 | 1 | 0 | 0 | 62.5 ms | 16 Hz |
| 0 | 1 | 1 | 0 | 1 | 125 ms | 8 Hz |
| 0 | 1 | 1 | 1 | 0 | 250 ms | 4 Hz |
| 0 | 1 | 1 | 1 | 1 | 500 ms | 2 Hz |

UPDATE CYCLE

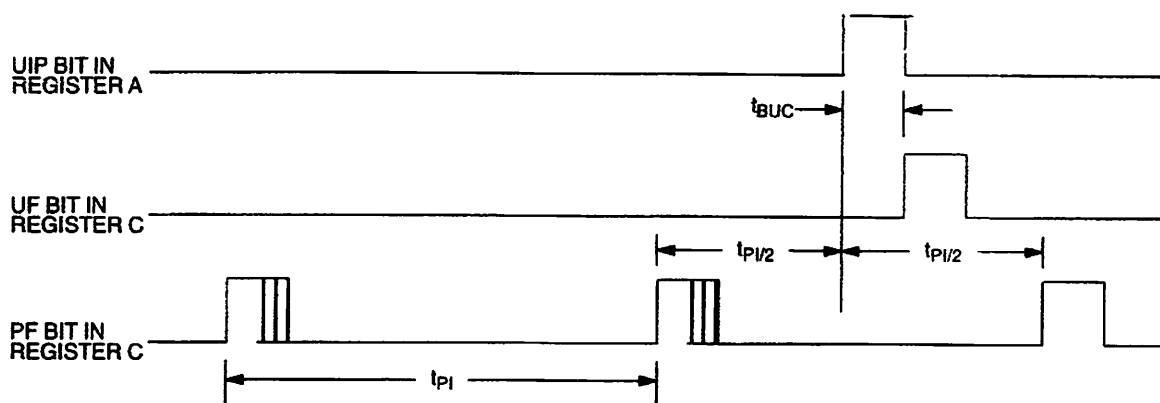
The DS12C887 executes an update cycle once per second regardless of the SET bit in Register B. When the SET bit in Register B is set to one, the user copy of the double buffered time, calendar, and alarm bytes is frozen and will not update as the time increments. However, the time countdown chain continues to update the internal copy of the buffer. This feature allows time to maintain accuracy independent of reading or writing the time, calendar, and alarm buffers and also guarantees that time and calendar information is consistent. The update cycle also compares each alarm byte with the corresponding time byte and issues an alarm if a match or if a “don’t care” code is present in all three positions.

There are three methods that can handle access of the real time clock that avoid any possibility of accessing inconsistent time and calendar data. The first method uses the update-ended interrupt. If enabled, an interrupt occurs after every update cycle that indicates that over 999 ms are available to read valid time and date information. If this interrupt is used, the IRQF bit in Register C should be cleared before leaving the interrupt routine.

The second method uses the update-in-progress bit (UIP) in Register A to determine if the update cycle is in progress. The UIP bit will pulse once per second. After the UIP bit goes high, the update transfer occurs 244 μ s later. If a low is read on the UIP bit, the user has at least 244 μ s before the time/calendar data will be changed. Therefore, the user should avoid interrupt service routines that would cause the time needed to read valid time/calendar data to exceed 244 μ s.

The third method uses a periodic interrupt to determine if an update cycle is in progress. The UIP bit in Register A is set high between the setting of the PF bit in Register C (see Figure 3). Periodic interrupts occur at a rate of greater than t_{BUC} allow valid time and date information to be reached at each occurrence of the periodic interrupt. The reads should be complete within $1 (t_{PI/2} + t_{BUC})$ to ensure that data is not read during the update cycle.

UPDATE-ENDED AND PERIODIC INTERRUPT RELATIONSHIP Figure 3



t_{PI} = PERIODIC INTERRUPT TIME INTERNAL PER TABLE 1
 t_{BUC} = DELAY TIME BEFORE UPDATE CYCLE = 244 μ s

ABSOLUTE MAXIMUM RATINGS*

| | |
|---------------------------------------|-----------------------------------|
| Voltage on Any Pin Relative to Ground | -0.3V to +7.0V |
| Operating Temperature | 0° to 70°C |
| Storage Temperature | -40°C to +70°C |
| Soldering Temperature | 260°C for 10 seconds (See Note 7) |

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

RECOMMENDED DC OPERATING CONDITIONS (0°C to 70°C)

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS | NOTES |
|----------------------|-----------------|------|-----|----------------------|-------|-------|
| Power Supply Voltage | V _{CC} | 4.5 | 5.0 | 5.5 | V | 1 |
| Input Logic 1 | V _{IH} | 2.3 | | V _{CC} +0.3 | V | 1 |
| Input Logic 0 | V _{IL} | -0.3 | | 0.8 | V | 1 |

DC ELECTRICAL CHARACTERISTICS (0°C to 70°C; V_{CC} = 5.0V ± 10%)

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS | NOTES |
|--|------------------|------|------|------|-------|-------|
| Average V _{CC} Power Supply Current | I _{CC1} | | 7 | 15 | mA | 2 |
| Input Leakage | I _{IL} | -1.0 | | +1.0 | μA | 3 |
| /O Leakage | I _{OL} | -1.0 | | +1.0 | μA | 4 |
| Input Current | I _{MOT} | -1.0 | | +500 | μA | 3 |
| Output @ 2.4V | I _{OH} | -1.0 | | | mA | 1,5 |
| Output @ 0.4V | I _{OL} | | | 4.0 | mA | 1 |
| Write Protect Voltage | V _{TP} | 4.0 | 4.25 | 4.5 | V | |

CAPACITANCE (t_A = 25°C)

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS | NOTES |
|--------------------|------------------|-----|-----|-----|-------|-------|
| Input Capacitance | C _{IN} | | | 5 | pF | |
| Output Capacitance | C _{OUT} | | | 7 | pF | |

DC ELECTRICAL CHARACTERISTICS (0°C to 70°C; VCC = 5.0V ± 10%)

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS | NOTES |
|---|---------------------------------|-----|-----|-----|-------|-------|
| Cycle Time | t _{CYC} | 385 | | DC | ns | |
| Pulse Width, DS/E Low or RD/WR High | PW _{EL} | 150 | | | ns | |
| Pulse Width, DS/E High or RD/WR Low | PW _{RH} | 125 | | | ns | |
| Input Rise and Fall | t _R , t _F | | | 30 | ns | |
| R/W Hold Time | t _{RWH} | 10 | | | ns | |
| R/W Setup Time Before DS/E | t _{RWS} | 50 | | | ns | |
| Chip Select Setup Time Before DS, WR, or RD | t _{CS} | 20 | | | ns | |
| Chip Select Hold Time | t _{CH} | 0 | | | ns | |
| Read Data Hold Time | t _{DHR} | 10 | | 80 | ns | |
| Write Data Hold Time | t _{DHW} | 0 | | | ns | |
| Mux'ed Address Valid Time to ALE Fall | t _{ASL} | 30 | | | ns | |
| Mused Address Hold Time to ALE Fall | t _{AHL} | 10 | | | ns | |
| Delay Time DS/E to AS/ALE Rise | t _{ASD} | 20 | | | ns | |
| Pulse Width AS/ALE High | PW _{ASH} | 60 | | | ns | |
| Delay Time, AS/ALE to DS/E Rise | t _{ASED} | 40 | | | ns | |
| Output Data Delay Time from DS/E or RD | t _{DDR} | 20 | | 120 | ns | 6 |
| Data Setup Time | t _{DSW} | 100 | | | ns | |
| Reset Pulse Width | t _{RWL} | 5 | | | μs | |
| RQ Release from DS | t _{IRDS} | | | 2 | μs | |
| RQ Release from RESET | t _{IRR} | | | 2 | μs | |

NOTES:

All voltages are referenced to ground.

All Outputs are open.

The MOT pin has an internal pull-down of 20KΩ.

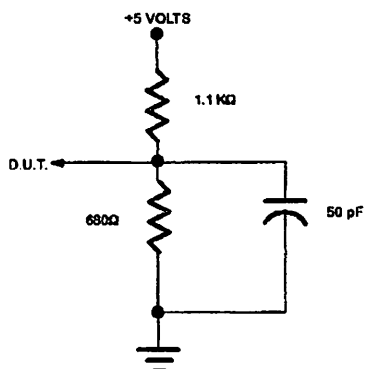
Applies to the AD0-AD7 pins, the IRQ pin, and the SQW pin when each is in a high impedance state.

The IRQ pin is open drain.

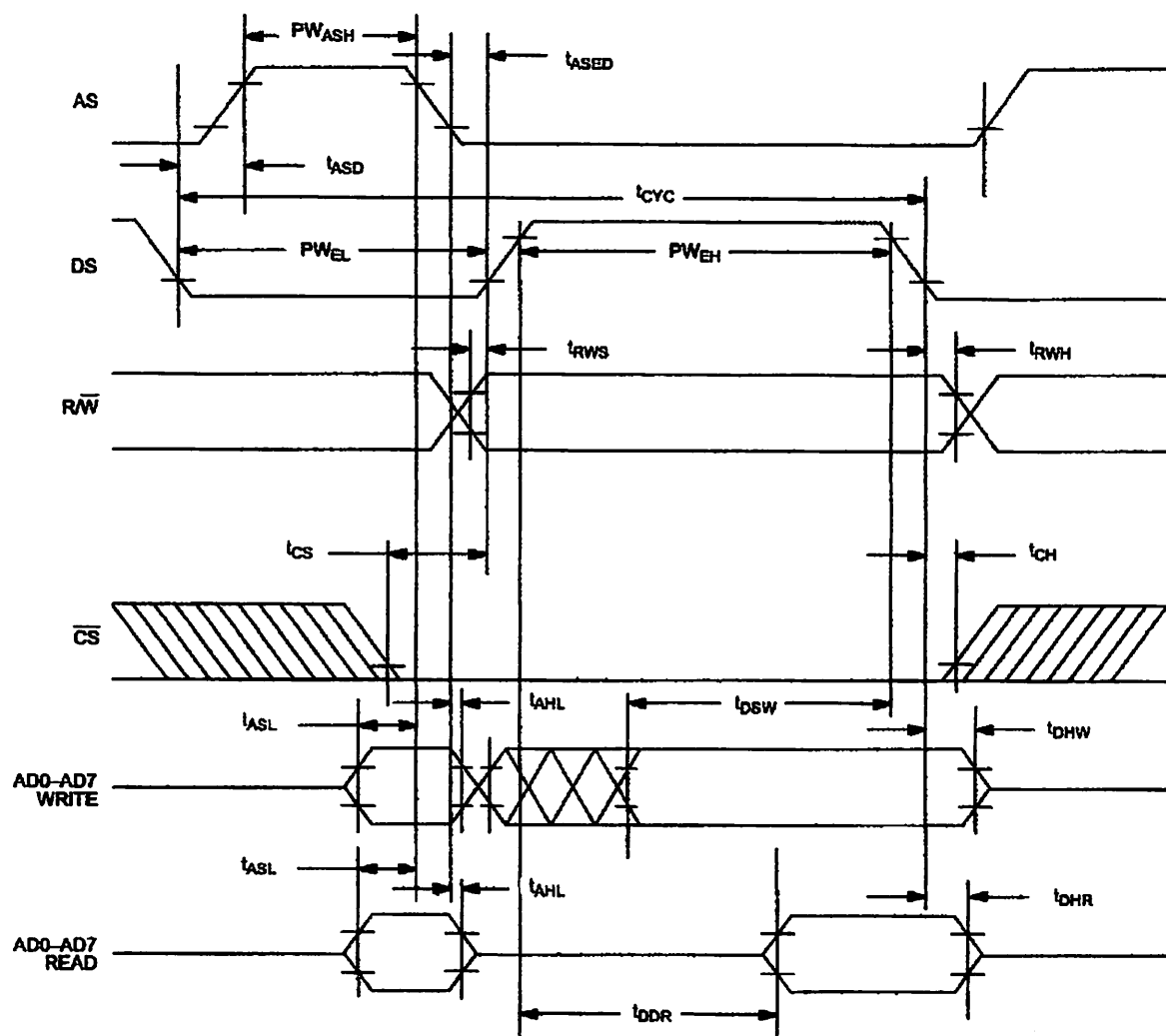
Measured with a load as shown in Figure 4.

Real-Time Clock Modules can be successfully processed through conventional wave-soldering techniques as long as temperature exposure to the lithium energy source contained within does not exceed +85°C. Post solder cleaning with water washing techniques is acceptable, provided that ultrasonic vibration is not used. Such cleaning can damage the crystal.

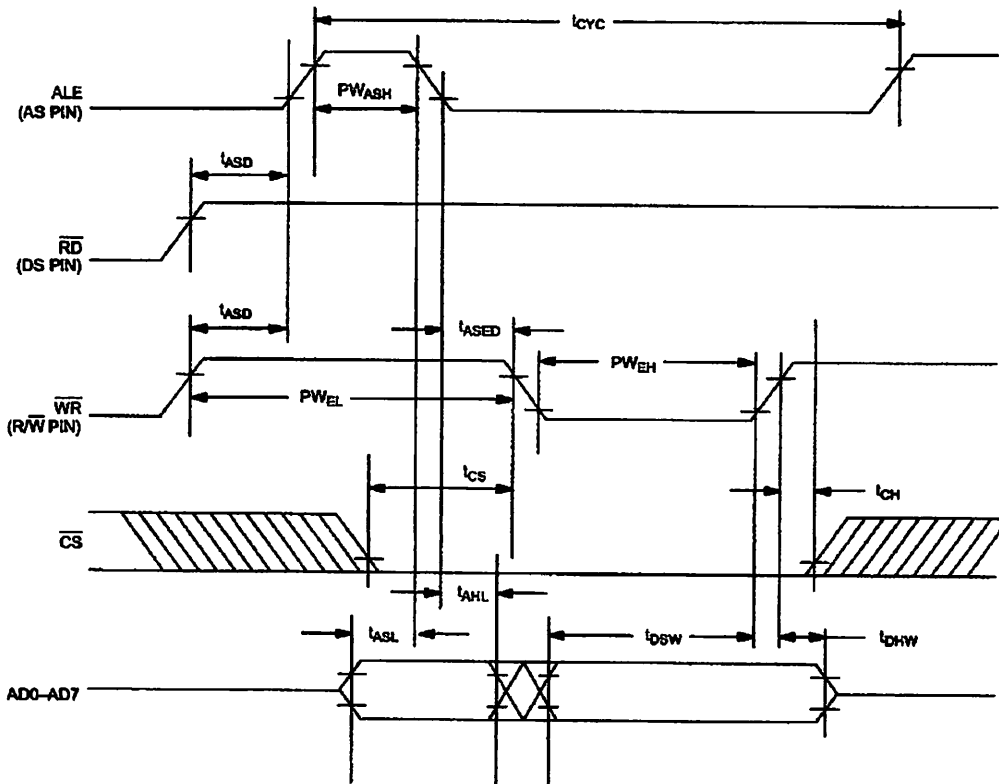
OUTPUT LOAD Figure 4



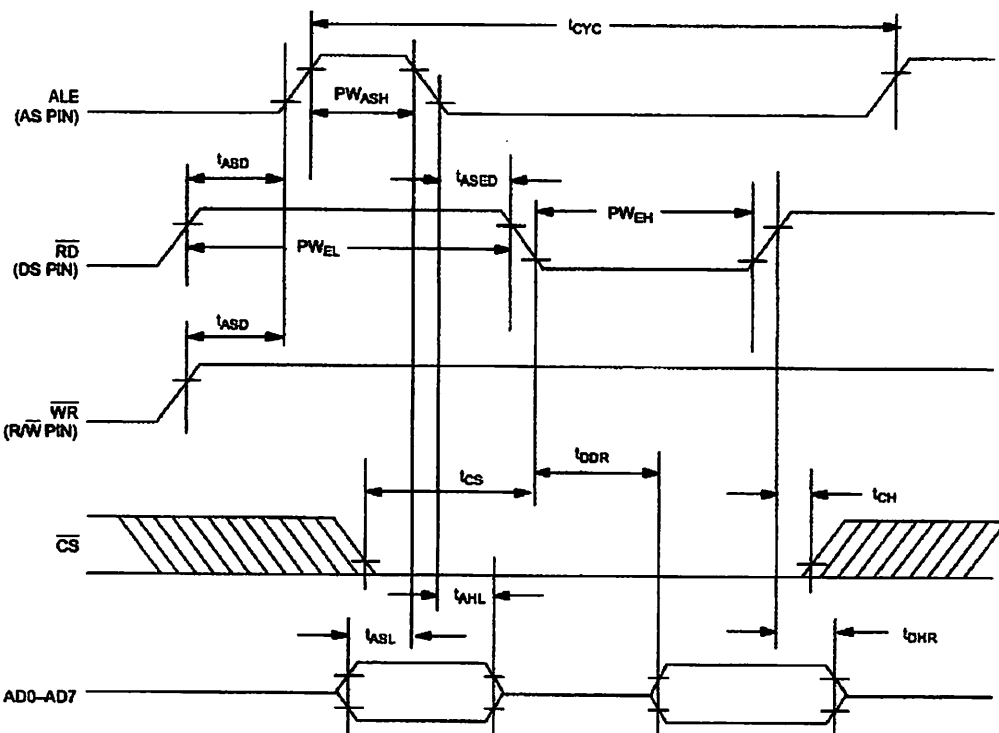
DS12C887 BUS TIMING FOR MOTOROLA INTERFACE



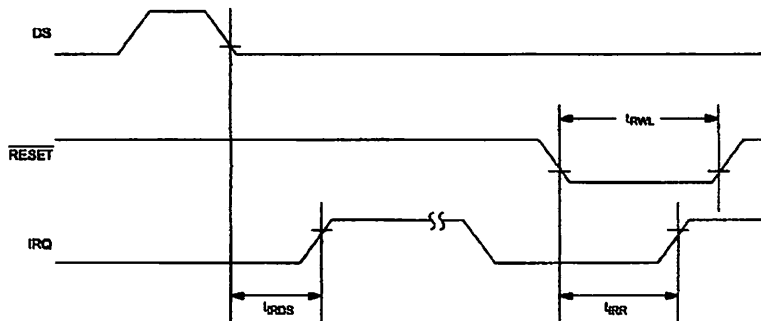
S12C887 BUS TIMING FOR INTEL INTERFACE WRITE CYCLE



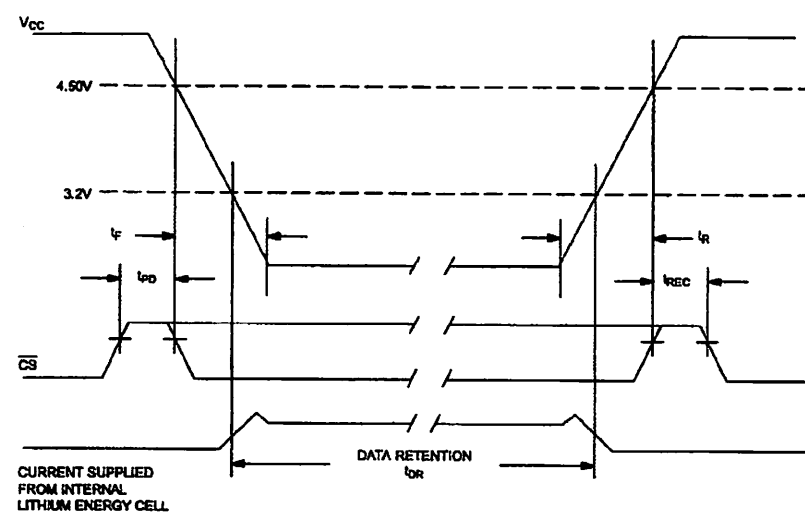
S12C887 BUS TIMING FOR INTEL INTERFACE READ CYCLE



S12C887 IRQ RELEASE DELAY TIMING



POWER DOWN / POWER UP TIMING



POWER DOWN / POWER UP TIMING

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS | NOTES |
|---|--------------------------------------|-----|-----|-----|---------|-------|
| \overline{CS} at V_{IH} before Power-Down | t_{PD} | | | 0 | μs | |
| $\overline{V_{CC}}$ slew from 4.5V to 0V (\overline{CS} at V_{IH}) | t_F $4.0 \leq V_{CC} \leq 4.5V$ | 300 | | | μs | |
| $\overline{V_{CC}}$ slew from 0V to 4.5V (\overline{CS} at V_{IH}) | t_R | 100 | | | μs | |
| \overline{CS} at V_{IH} after Power-Up | t_{REC} | 20 | | 200 | ms | |
| Expected Data Retention | t_{DR} | 10 | | | years | 10,11 |

($t_A=25^\circ C$)

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS | NOTES |
|-------------------------|----------|-----|-----|-----|-------|-------|
| Expected Data Retention | t_{DR} | 10 | | | years | 10,11 |

Note:

The real time clock will keep time to an accuracy of ± 1 minute per month during data retention time for the period of t_{DR} .

Warning:

Under no circumstances are negative undershoots, of any amplitude, allowed when device is in battery back-up mode.

ADC0808/ADC0809

8-Bit μ P Compatible A/D Converters with 8-Channel Multiplexer

General Description

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals.

The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.)

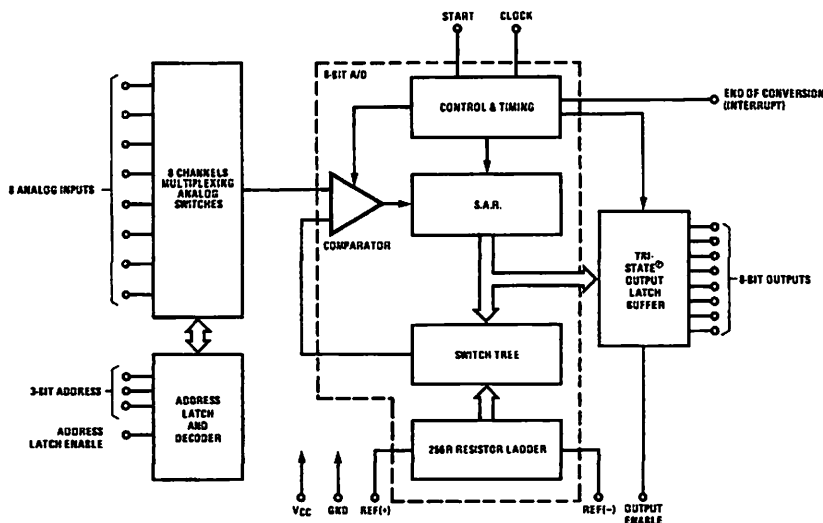
Features

- Easy interface to all microprocessors
- Operates ratiometrically or with 5 V_{DC} or analog span adjusted voltage reference
- No zero or full-scale adjust required
- 8-channel multiplexer with address logic
- 0V to 5V input range with single 5V power supply
- Outputs meet TTL voltage level specifications
- ADC0808 equivalent to MM74C949
- ADC0809 equivalent to MM74C949-1

Key Specifications

- | | |
|--------------------------|-------------------------------|
| ■ Resolution | 8 Bits |
| ■ Total Unadjusted Error | $\pm 1/2$ LSB and ± 1 LSB |
| ■ Single Supply | 5 V _{DC} |
| ■ Low Power | 15 mW |
| ■ Conversion Time | 100 μ s |

Block Diagram



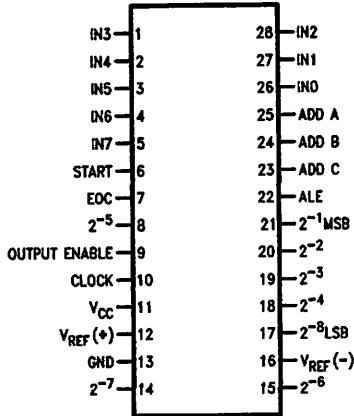
See Ordering Information

00567201

ADC0808/ADC0809 8-Bit μ P Compatible A/D Converters with 8-Channel Multiplexer

Connection Diagrams

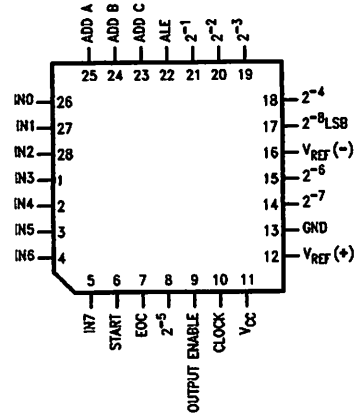
Dual-In-Line Package



00667211

Order Number ADC0808CCN or ADC0809CCN
See NS Package J28A or N28A

Molded Chip Carrier Package



00667212

Order Number ADC0808CCV or ADC0809CCV
See NS Package V28A

Ordering Information

| | | | |
|-------------------|-------------------|-----------------|--------------------------|
| TEMPERATURE RANGE | | -40°C to +85°C | |
| Error | ±½ LSB Unadjusted | ADC0808CCN | ADC0808CCV |
| | ±1 LSB Unadjusted | ADC0809CCN | ADC0809CCV |
| Package Outline | | N28A Molded DIP | V28A Molded Chip Carrier |

Absolute Maximum Ratings (Notes 2,

1)
If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

| | |
|--|--------------------------|
| Supply Voltage (V_{CC}) (Note 3) | 6.5V |
| Voltage at Any Pin | -0.3V to $(V_{CC}+0.3V)$ |
| Except Control Inputs | |
| Voltage at Control Inputs (START, OE, CLOCK, ALE, ADD A, ADD B, ADD C) | -0.3V to +15V |
| Storage Temperature Range | -65°C to +150°C |
| Package Dissipation at $T_A=25^\circ\text{C}$ | 875 mW |
| Lead Temp. (Soldering, 10 seconds) | |

| | |
|--------------------------------|-------|
| Dual-In-Line Package (plastic) | 260°C |
| Molded Chip Carrier Package | |
| Vapor Phase (60 seconds) | 215°C |
| Infrared (15 seconds) | 220°C |
| ESD Susceptibility (Note 8) | 400V |

Operating Conditions (Notes 1, 2)

| | |
|----------------------------|---|
| Temperature Range (Note 1) | $T_{MIN} \leq T_A \leq T_{MAX}$ |
| ADC0808CCN, ADC0809CCN | $-40^\circ\text{C} \leq T_A \leq +85^\circ\text{C}$ |
| ADC0808CCV, ADC0809CCV | $-40^\circ\text{C} \leq T_A \leq +85^\circ\text{C}$ |
| Range of V_{CC} (Note 1) | 4.5 V_{DC} to 6.0 V_{DC} |

Electrical Characteristics

Converter Specifications: $V_{CC}=5$ $V_{DC}=V_{REF+}$, $V_{REF(-)}=GND$, $T_{MIN} \leq T_A \leq T_{MAX}$ and $f_{CLK}=640$ kHz unless otherwise stated.

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|-------------------------------------|---------------------------------|-------------------------|----------------|------------|----------------|---------------|
| | ADC0808 | | | | | |
| | Total Unadjusted Error (Note 5) | 25°C | | | $\pm 1/2$ | LSB |
| | | T_{MIN} to T_{MAX} | | | $\pm 3/4$ | LSB |
| | ADC0809 | | | | | |
| | Total Unadjusted Error (Note 5) | 0°C to 70°C | | | ± 1 | LSB |
| | | T_{MIN} to T_{MAX} | | | $\pm 1 1/4$ | LSB |
| | Input Resistance | From Ref(+) to Ref(-) | 1.0 | 2.5 | | k Ω |
| | Analog Input Voltage Range | (Note 4) V(+) or V(-) | GND-0.10 | | $V_{CC}+0.10$ | V_{DC} |
| REF(+) | Voltage, Top of Ladder | Measured at Ref(+) | | V_{CC} | $V_{CC}+0.1$ | V |
| $\frac{V_{REF(+)} + V_{REF(-)}}{2}$ | Voltage, Center of Ladder | | $V_{CC}/2-0.1$ | $V_{CC}/2$ | $V_{CC}/2+0.1$ | V |
| REF(-) | Voltage, Bottom of Ladder | Measured at Ref(-) | -0.1 | 0 | | V |
| I _V | Comparator Input Current | $f_c=640$ kHz, (Note 6) | -2 | ± 0.5 | 2 | μA |

Electrical Characteristics

Digital Levels and DC Specifications: ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, $4.75 \leq V_{CC} \leq 5.25\text{V}$, $-40^\circ\text{C} \leq T_A \leq +85^\circ\text{C}$ unless otherwise noted

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|---------------------------|--|---|--------------|-----|------------|---------------------|
| ANALOG MULTIPLEXER | | | | | | |
| OFF(+) | OFF Channel Leakage Current | $V_{CC}=5\text{V}$, $V_{IN}=5\text{V}$, $T_A=25^\circ\text{C}$ T_{MIN} to T_{MAX} | | 10 | 200 1.0 | nA μA |
| OFF(-) | OFF Channel Leakage Current | $V_{CC}=5\text{V}$, $V_{IN}=0$, $T_A=25^\circ\text{C}$ T_{MIN} to T_{MAX} | -200 -1.0 | -10 | | nA μA |
| CONTROL INPUTS | | | | | | |
| IN(1) | Logical "1" Input Voltage | | $V_{CC}-1.5$ | | | V |
| IN(0) | Logical "0" Input Voltage | | | | 1.5 | V |
| IN(1) | Logical "1" Input Current (The Control Inputs) | $V_{IN}=15\text{V}$ | | | 1.0 | μA |
| IN(0) | Logical "0" Input Current (The Control Inputs) | $V_{IN}=0$ | -1.0 | | | μA |
| I _{CC} | Supply Current | $f_{CLK}=640$ kHz | | 0.3 | 3.0 | mA |

Electrical Characteristics (Continued)

Digital Levels and DC Specifications: ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, $4.75 \leq V_{CC} \leq 5.25V$, $-40^\circ C \leq T_A \leq +85^\circ C$ unless otherwise noted

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|---|--------------------------------|---|-----|------------|------|--------------------|
| DATA OUTPUTS AND EOC (INTERRUPT) | | | | | | |
| $V_{OUT(1)}$ | Logical "1" Output Voltage | $V_{CC} = 4.75V$ $I_{OUT} = -360\mu A$ $I_{OUT} = -10\mu A$ | | 2.4 4.5 | | V(min) V(min) |
| $V_{OUT(0)}$ | Logical "0" Output Voltage | $I_O = 1.6 \text{ mA}$ | | | 0.45 | V |
| $V_{OUT(0)}$ | Logical "0" Output Voltage EOC | $I_O = 1.2 \text{ mA}$ | | | 0.45 | V |
| I_{OUT} | TRI-STATE Output Current | $V_O = 5V$ $V_O = 0$ | -3 | | 3 | μA μA |

Electrical Characteristics

Timing Specifications $V_{CC} = V_{REF(+)} = 5V$, $V_{REF(-)} = GND$, $t_r = t_f = 20 \text{ ns}$ and $T_A = 25^\circ C$ unless otherwise noted.

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|------------------|-----------------------------------|--|-----|-----|-------------|------------------|
| t_{WS} | Minimum Start Pulse Width | (Figure 5) | | 100 | 200 | ns |
| t_{WALE} | Minimum ALE Pulse Width | (Figure 5) | | 100 | 200 | ns |
| t_s | Minimum Address Set-Up Time | (Figure 5) | | 25 | 50 | ns |
| t_H | Minimum Address Hold Time | (Figure 5) | | 25 | 50 | ns |
| t_D | Analog MUX Delay Time From ALE | $R_S = 0\Omega$ (Figure 5) | | 1 | 2.5 | μs |
| t_{HI}, t_{HO} | OE Control to Q Logic State | $C_L = 50 \text{ pF}$, $R_L = 10k$ (Figure 8) | | 125 | 250 | ns |
| t_{HI}, t_{OH} | OE Control to Hi-Z | $C_L = 10 \text{ pF}$, $R_L = 10k$ (Figure 8) | | 125 | 250 | ns |
| t_c | Conversion Time | $f_c = 640 \text{ kHz}$, (Figure 5) (Note 7) | 90 | 100 | 116 | μs |
| f_c | Clock Frequency | | 10 | 640 | 1280 | kHz |
| t_{EOC} | EOC Delay Time | (Figure 5) | 0 | | 8+2 μs | Clock Periods |
| C_{IN} | Input Capacitance | At Control Inputs | | 10 | 15 | pF |
| C_{OUT} | TRI-STATE Output Capacitance | At TRI-STATE Outputs | | 10 | 15 | pF |

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: All voltages are measured with respect to GND, unless otherwise specified.

Note 3: A zener diode exists, internally, from V_{CC} to GND and has a typical breakdown voltage of $7 V_{DC}$.

Note 4: Two on-chip diodes are tied to each analog input which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_{CC} supply. The spec allows 100 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 100 mV, the output code will be correct. To achieve an absolute $0V_{DC}$ to $5V_{DC}$ input voltage range will therefore require a minimum supply voltage of $\approx 1.900 V_{DC}$ over temperature variations, initial tolerance and loading.

Note 5: Total unadjusted error includes offset, full-scale, linearity, and multiplexer errors. See Figure 3. None of these A/Ds requires a zero or full-scale adjust. However, if an all zero code is desired for an analog input other than 0.0V, or if a narrow full-scale span exists (for example: 0.5V to 4.5V full-scale) the reference voltages can be adjusted to achieve this. See Figure 13.

Note 6: Comparator input current is a bias current into or out of the chopper stabilized comparator. The bias current varies directly with clock frequency and has a typical value of $\pm 10 \mu A$.

Note 7: The outputs of the data register are updated one clock cycle before the rising edge of EOC.

Note 8: Human body model, 100 pF discharged through a 1.5 k Ω resistor.

Functional Description

Multiplexer. The device contains an 8-channel single-ended analog signal multiplexer. A particular input channel is selected by using the address decoder. *Table 1* shows the input states for the address lines to select any channel. The address is latched into the decoder on the low-to-high transition of the address latch enable signal.

TABLE 1.

| SELECTED ANALOG CHANNEL | ADDRESS LINE | | |
|----------------------------|--------------|---|---|
| | C | B | A |
| IN0 | L | L | L |
| IN1 | L | L | H |
| IN2 | L | H | L |
| IN3 | L | H | H |
| IN4 | H | L | L |
| IN5 | H | L | H |
| IN6 | H | H | L |
| IN7 | H | H | H |

CONVERTER CHARACTERISTICS

The Converter

The heart of this single chip data acquisition system is its 8-bit analog-to-digital converter. The converter is designed to give fast, accurate, and repeatable conversions over a wide range of temperatures. The converter is partitioned into 3 major sections: the 256R ladder network, the successive approximation register, and the comparator. The converter's digital outputs are positive true.

The 256R ladder network approach (*Figure 1*) was chosen over the conventional R/2R ladder because of its inherent monotonicity, which guarantees no missing digital codes. Monotonicity is particularly important in closed loop feedback control systems. A non-monotonic relationship can cause oscillations that will be catastrophic for the system. Additionally, the 256R network does not cause load variations on the reference voltage.

The bottom resistor and the top resistor of the ladder network in *Figure 1* are not the same value as the remainder of the network. The difference in these resistors causes the output characteristic to be symmetrical with the zero and full-scale points of the transfer curve. The first output transition occurs when the analog signal has reached $+1/2$ LSB and succeeding output transitions occur every 1 LSB later up to full-scale.

The successive approximation register (SAR) performs 8 iterations to approximate the input voltage. For any SAR type converter, n-iterations are required for an n-bit converter. *Figure 2* shows a typical example of a 3-bit converter. In the ADC0808, ADC0809, the approximation technique is extended to 8 bits using the 256R network.

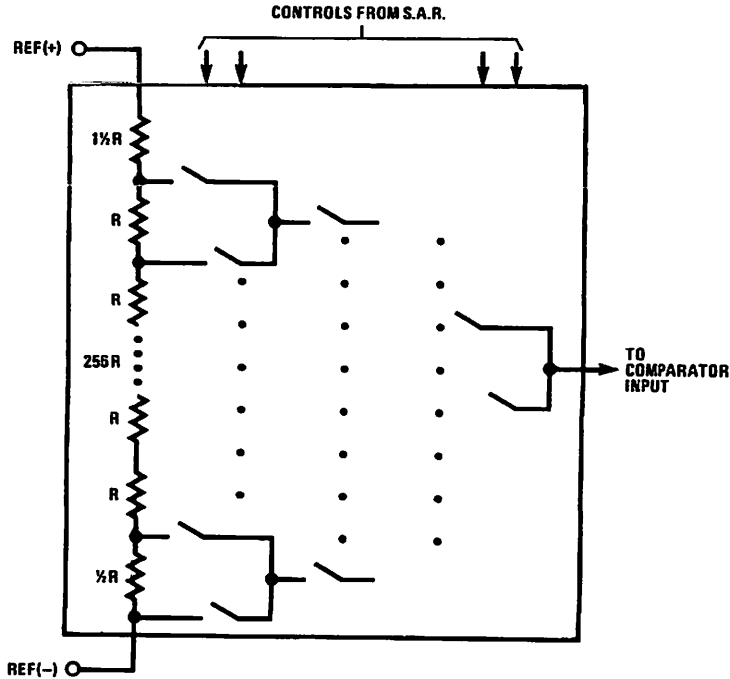
The A/D converter's successive approximation register (SAR) is reset on the positive edge of the start conversion start pulse. The conversion is begun on the falling edge of the start conversion pulse. A conversion in process will be interrupted by receipt of a new start conversion pulse. Continuous conversion may be accomplished by tying the end-of-conversion (EOC) output to the SC input. If used in this mode, an external start conversion pulse should be applied after power up. End-of-conversion will go low between 0 and 8 clock pulses after the rising edge of start conversion.

The most important section of the A/D converter is the comparator. It is this section which is responsible for the ultimate accuracy of the entire converter. It is also the comparator drift which has the greatest influence on the repeatability of the device. A chopper-stabilized comparator provides the most effective method of satisfying all the converter requirements.

The chopper-stabilized comparator converts the DC input signal into an AC signal. This signal is then fed through a high gain AC amplifier and has the DC level restored. This technique limits the drift component of the amplifier since the drift is a DC component which is not passed by the AC amplifier. This makes the entire A/D converter extremely insensitive to temperature, long term drift and input offset errors.

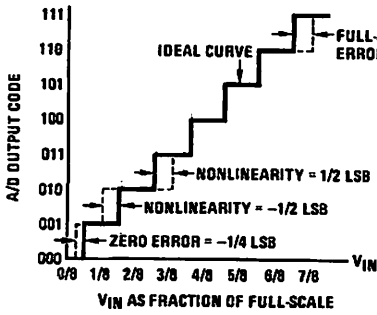
Figure 4 shows a typical error curve for the ADC0808 as measured using the procedure outlined in AN-179.

Functional Description (Continued)



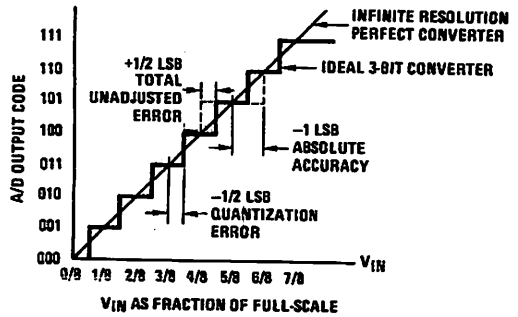
00567202

FIGURE 1. Resistor Ladder and Switch Tree



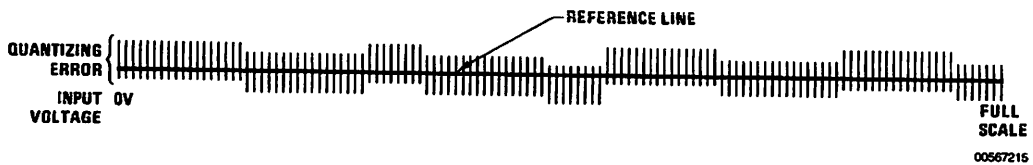
00567213

FIGURE 2. 3-Bit A/D Transfer Curve



00567214

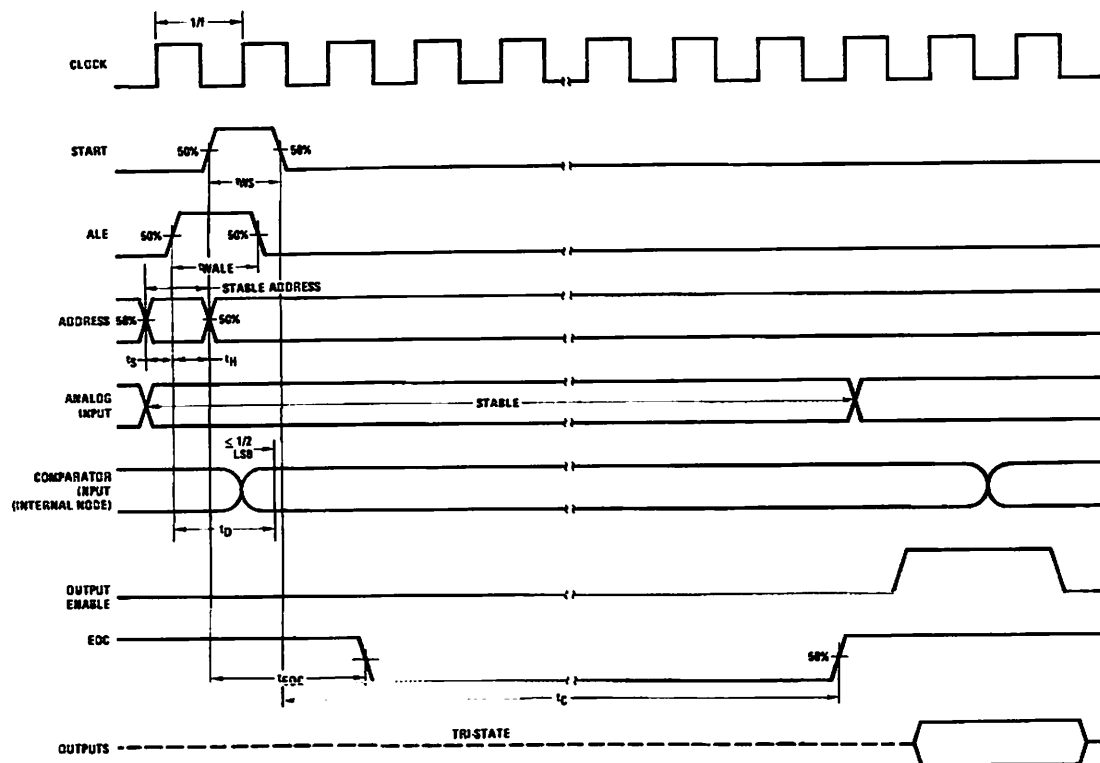
FIGURE 3. 3-Bit A/D Absolute Accuracy Curve



00567215

FIGURE 4. Typical Error Curve

Timing Diagram



00567204

FIGURE 5.

00000000000000000000

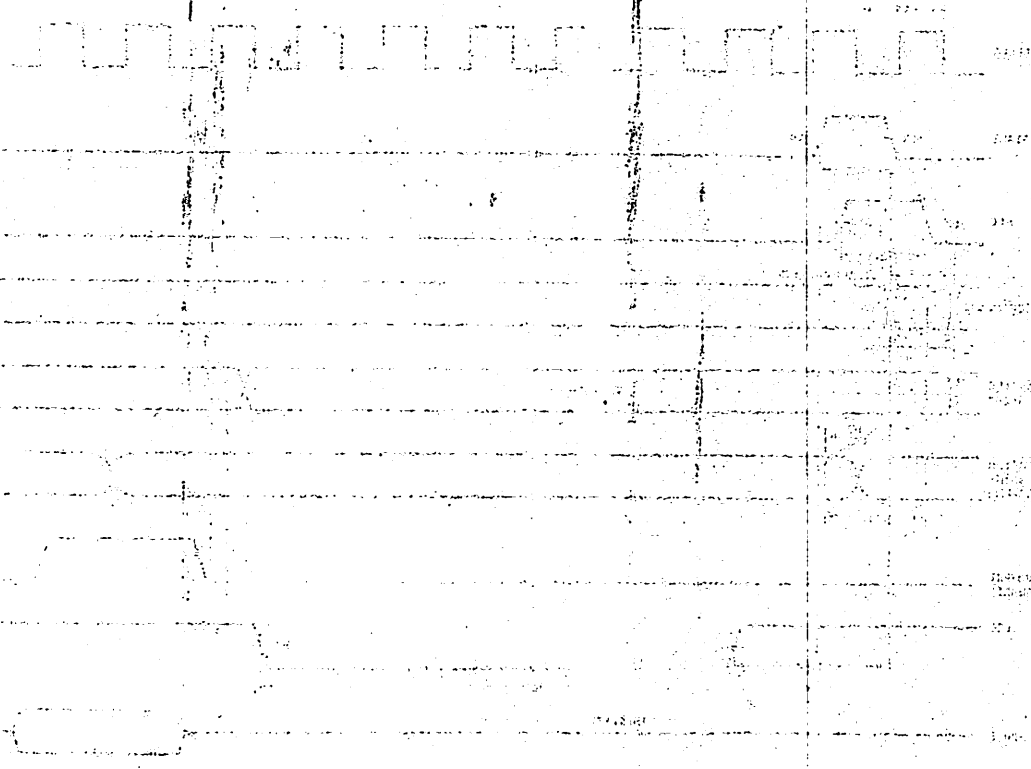
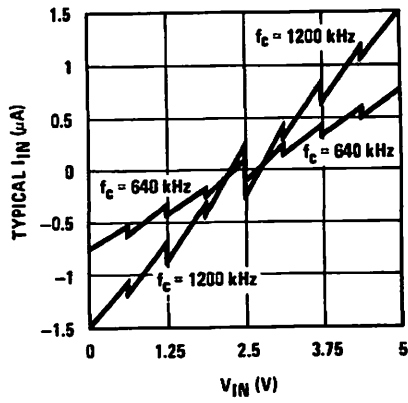


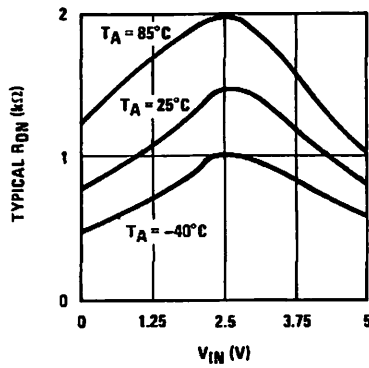
FIGURE 1

Typical Performance Characteristics



00567216

FIGURE 6. Comparator I_{IN} vs V_{IN}
($V_{CC}=V_{REF}=5V$)



00567217

FIGURE 7. Multiplexer R_{ON} vs V_{IN}
($V_{CC}=V_{REF}=5V$)

TRI-STATE Test Circuits and Timing Diagrams

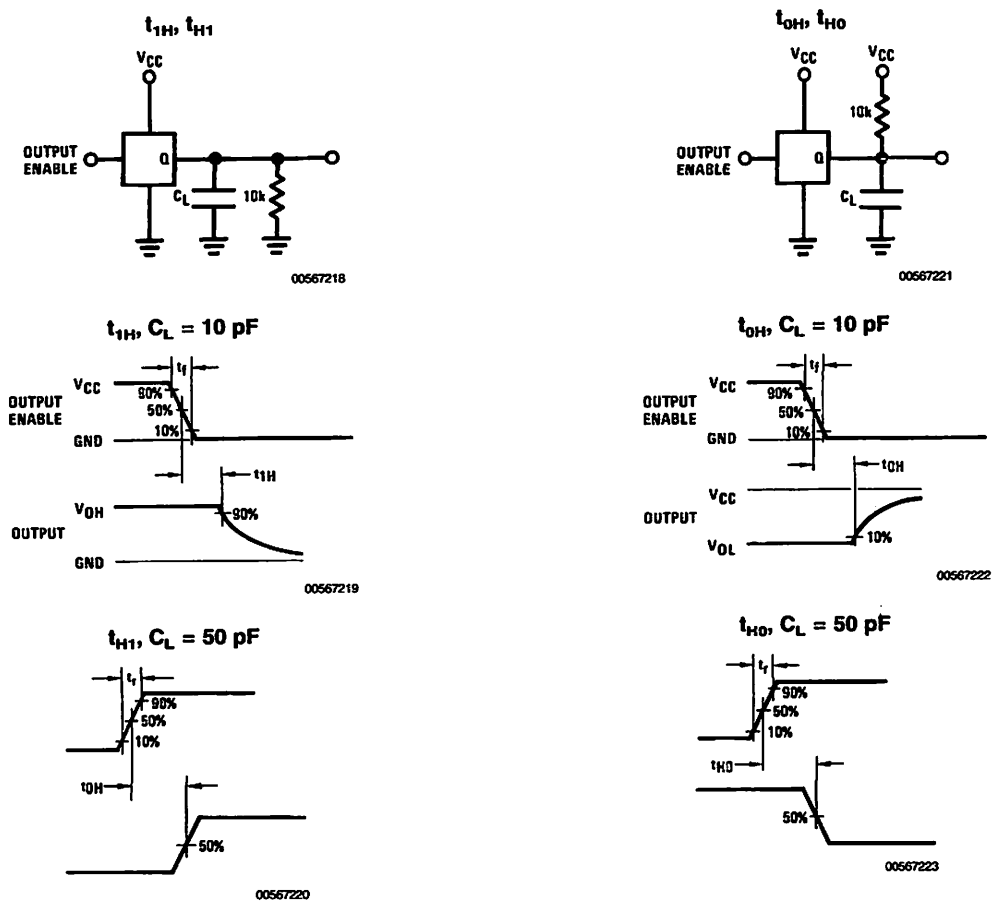


FIGURE 8.

Applications Information

OPERATION

1.0 RATIOMETRIC CONVERSION

The ADC0808, ADC0809 is designed as a complete Data Acquisition System (DAS) for ratiometric conversion systems. In ratiometric systems, the physical variable being measured is expressed as a percentage of full-scale which is not necessarily related to an absolute standard. The voltage input to the ADC0808 is expressed by the equation

$$\frac{V_{IN} - V_Z}{V_{FS} - V_Z} = \frac{D_X}{D_{MAX} - D_{MIN}} \quad (1)$$

V_{IN} = Input voltage into the ADC0808

V_{FS} = Full scale voltage

V_Z = Zero voltage

D_X = Data point being measured

D_{MAX} = Maximum data limit

D_{MIN} = Minimum data limit

A good example of a ratiometric transducer is a potentiometer used as a position sensor. The position of the wiper is directly proportional to the output voltage which is a ratio of the full-scale voltage across it. Since the data is represented as a proportion of full-scale, reference requirements are greatly reduced, eliminating a large source of error and cost for many applications. A major advantage of the ADC0808, ADC0809 is that the input voltage range is equal to the supply range so the transducers can be connected directly across the supply and their outputs connected directly into the multiplexer inputs, (Figure 9).

Ratiometric transducers such as potentiometers, strain gauges, thermistor bridges, pressure transducers, etc., are suitable for measuring proportional relationships; however, many types of measurements must be referred to an absolute standard such as voltage or current. This means a system reference must be used which relates the full-scale voltage to the standard volt. For example, if $V_{CC} = V_{REF} = 5.12V$, then the full-scale range is divided into 256 standard steps. The smallest standard step is 1 LSB which is then 20 mV.

Applications Information (Continued)

2.0 RESISTOR LADDER LIMITATIONS

The voltages from the resistor ladder are compared to the selected into 8 times in a conversion. These voltages are coupled to the comparator via an analog switch tree which is referenced to the supply. The voltages at the top, center and bottom of the ladder must be controlled to maintain proper operation.

The top of the ladder, Ref(+), should not be more positive than the supply, and the bottom of the ladder, Ref(-), should

not be more negative than ground. The center of the ladder voltage must also be near the center of the supply because the analog switch tree changes from N-channel switches to P-channel switches. These limitations are automatically satisfied in ratiometric systems and can be easily met in ground referenced systems.

Figure 10 shows a ground referenced system with a separate supply and reference. In this system, the supply must be trimmed to match the reference voltage. For instance, if a 5.12V is used, the supply should be adjusted to the same voltage within 0.1V.

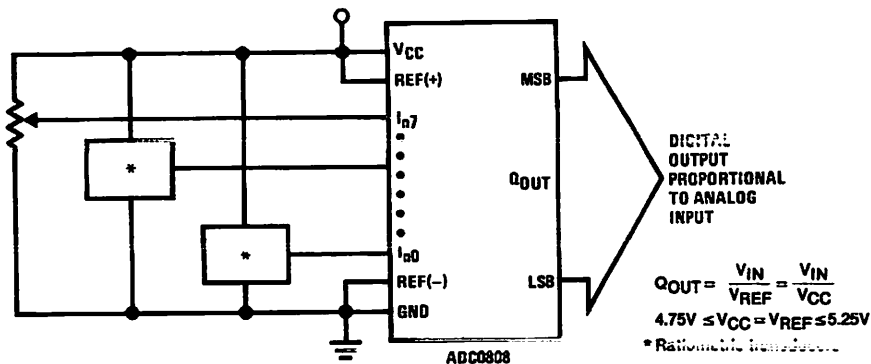
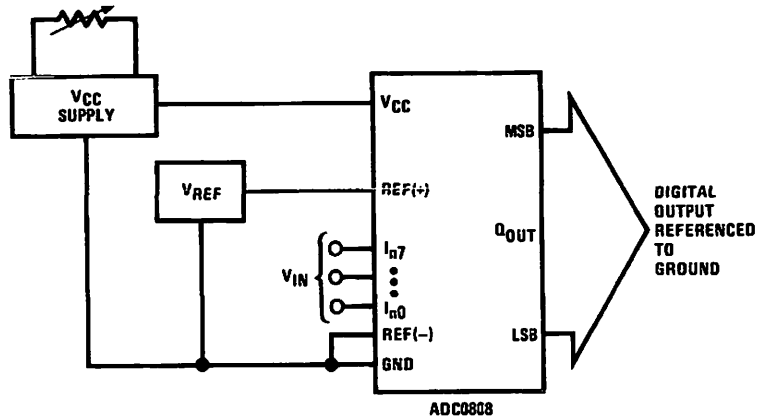


FIGURE 9. Ratiometric Conversion System

The ADC0808 needs less than a milliamp of supply current so developing the supply from the reference is readily accomplished. In Figure 11 a ground referenced system is shown which generates the supply from the reference. The buffer shown can be an op amp of sufficient drive to supply the milliamp of supply current and the desired bus drive, or if a capacitive bus is driven by the outputs a large capacitor will supply the transient supply current as seen in Figure 12. The LM301 is overcompensated to insure stability when loaded by the 10 μ F output capacitor.

The top and bottom ladder voltages cannot exceed V_{CC} and ground, respectively, but they can be symmetrically less than V_{CC} and greater than ground. The center of the ladder voltage should always be near the center of the supply. The sensitivity of the converter can be increased, (i.e., size of the LSB steps decreased) by using a symmetrical reference system. In Figure 10, a 2.5V reference is symmetrically centered about $V_{CC}/2$ since the same current flows in identical resistors. This system with a 2.5V reference allows the LSB bit to be half the size of a 5V reference system.

Applications Information (Continued)

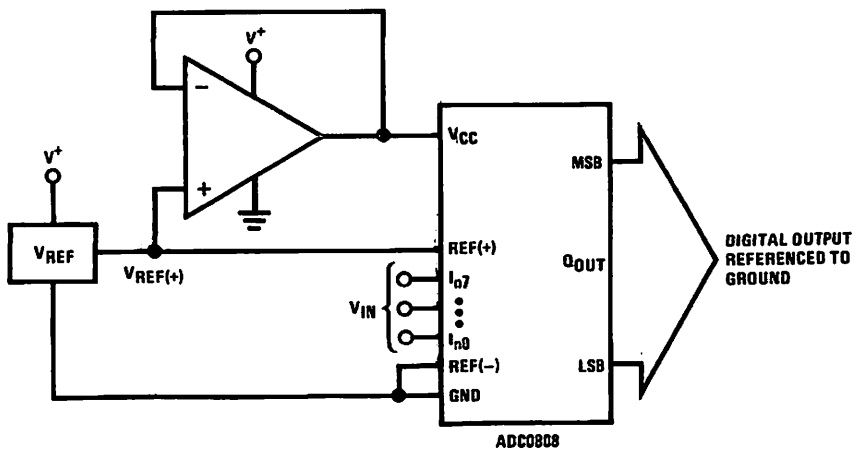


00567224

$$Q_{OUT} = \frac{V_{IN}}{V_{REF}}$$

$$4.75V \leq V_{CC} = V_{REF} \leq 5.25V$$

FIGURE 10. Ground Referenced Conversion System Using Trimmed Supply



00567225

$$Q_{OUT} = \frac{V_{IN}}{V_{REF}}$$

$$4.75V \leq V_{CC} = V_{REF} \leq 5.25V$$

FIGURE 11. Ground Referenced Conversion System with Reference Generating V_{CC} Supply

Applications Information (Continued)

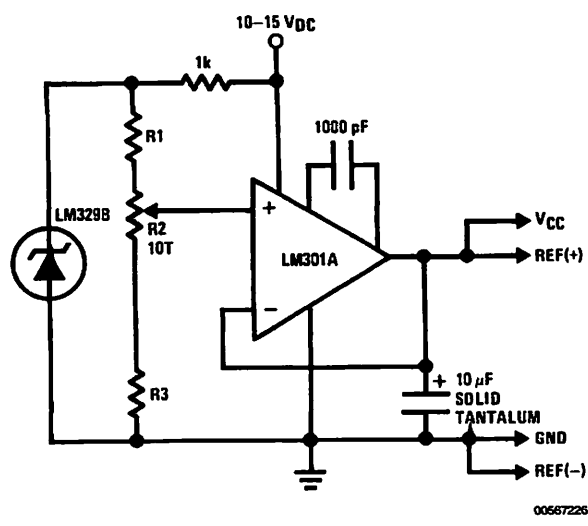
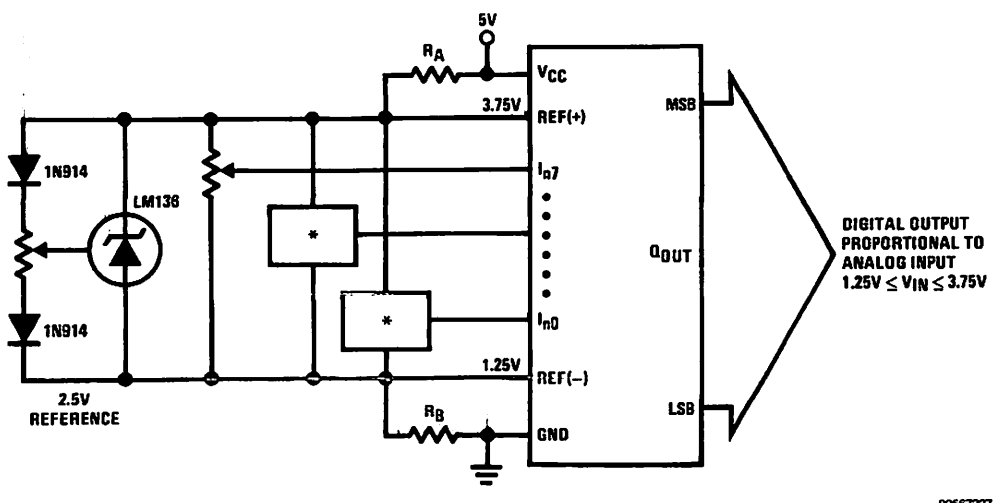


FIGURE 12. Typical Reference and Supply Circuit



$R_A = R_B$
 *Ratiometric transducers

FIGURE 13. Symmetrically Centered Reference

3.0 CONVERTER EQUATIONS

The transition between adjacent codes N and N+1 is given by:

$$V_{IN} = \left\{ (V_{REF(+)} - V_{REF(-)}) \left[\frac{N}{256} + \frac{1}{512} \right] \pm V_{TUE} \right\} + V_{REF(-)} \tag{2}$$

The center of an output code N is given by:

$$V_{IN} \left\{ (V_{REF(+)} - V_{REF(-)}) \left[\frac{N}{256} \right] \pm V_{TUE} \right\} + V_{REF(-)} \tag{3}$$

The output code N for an arbitrary input are the integers within the range:

$$N = \frac{V_{IN} - V_{REF(-)}}{V_{REF(+)} - V_{REF(-)}} \times 256 \pm \text{Absolute Accuracy} \tag{4}$$

- Where: V_{IN} = Voltage at comparator input
- $V_{REF(+)}$ = Voltage at Ref(+)
- $V_{REF(-)}$ = Voltage at Ref(-)
- V_{TUE} = Total unadjusted error voltage (typically $V_{REF(+)} \div 512$)

Applications Information (Continued)

4.0 ANALOG COMPARATOR INPUTS

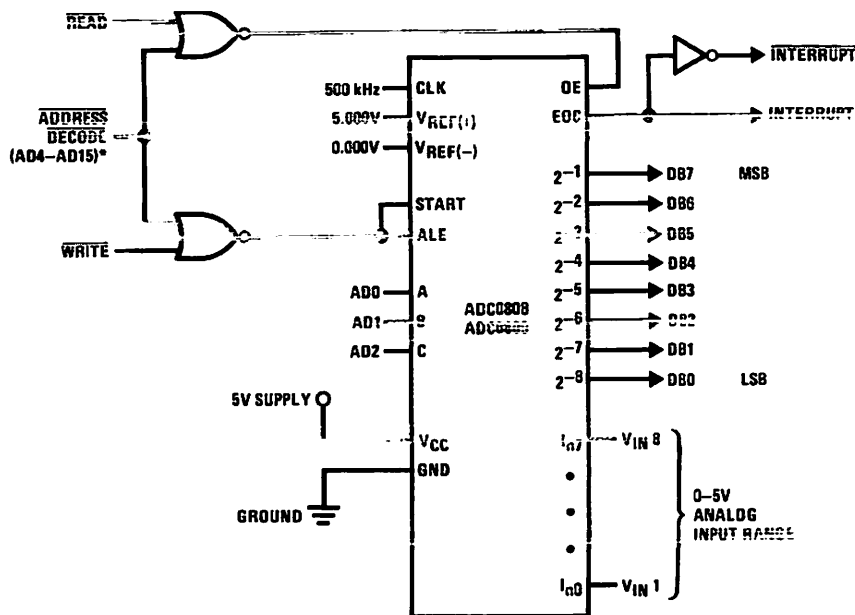
The dynamic comparator input current is caused by the periodic switching of on-chip stray capacitances. These are connected alternately to the output of the resistor ladder/switch tree network and to the comparator input as part of the operation of the chopper stabilized comparator.

The average value of the comparator input current varies directly with clock frequency and with V_{IN} as shown in Figure 6.

If no filter capacitors are used at the analog inputs and the signal source impedances are low, the comparator input current should not introduce converter errors, as the transient created by the capacitance discharge will die out before the comparator output is strobed.

If input filter capacitors are desired for noise reduction and signal conditioning they will tend to average out the dynamic comparator input current. It will then take on the characteristics of a DC bias current whose effect can be predicted conventionally.

Typical Application



*Address latches needed for 8085 and SC/MP interfacing the ADC0808 to a microprocessor

TABLE 2. Microprocessor Interface Table

| PROCESSOR | READ | WRITE | INTERRUPT (COMMENT) |
|-----------|---|---|---|
| 8080 | MEMR | MEMW | INTR (Thru RST Circuit) |
| 8085 | \overline{RD} | \overline{WR} | INTR (Thru RST Circuit) |
| Z-80 | \overline{RD} | \overline{WR} | \overline{INT} (Thru RST Circuit, Mode 0) |
| SC/MP | NRDS | NWDS | SA (Thru Sense A) |
| 6800 | \overline{VMA} , \overline{VQ} , \overline{DAV} | \overline{VMA} , \overline{VQ} , \overline{DAV} | INTR or INTR (Thru RST) |

LM35/LM35A/LM35C/LM35CA/LM35D Precision Centigrade Temperature Sensors

General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of $\pm 1/4^\circ\text{C}$ at room temperature and $\pm 3/4^\circ\text{C}$ over a full -55 to $+150^\circ\text{C}$ temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only $60\ \mu\text{A}$ from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a -55° to $+150^\circ\text{C}$ temperature range, while the LM35C is rated for a -40° to $+110^\circ\text{C}$ range (-10° with improved accuracy). The LM35 series is

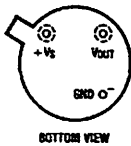
available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-202 package.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full -55° to $+150^\circ\text{C}$ range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than $60\ \mu\text{A}$ current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only $\pm 1/4^\circ\text{C}$ typical
- Low impedance output, $0.1\ \Omega$ for 1 mA load

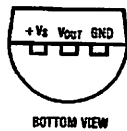
Connection Diagrams

**TO-46
Metal Can Package***



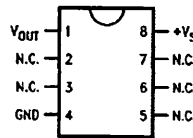
TL/H/5516-1

**TO-92
Plastic Package**



TL/H/5516-2

**SO-8
Small Outline Molded Package**



TL/H/5516-21

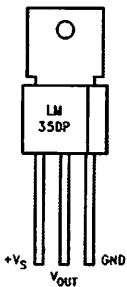
*Case is connected to negative pin (GND)

Order Number LM35H, LM35AH,
LM35CH, LM35CAH or LM35DH
See NS Package Number H03H

Order Number LM35CZ,
LM35CAZ or LM35DZ
See NS Package Number Z03A

N.C. = No Connection
Order Number LM35DM
See NS Package Number M08A

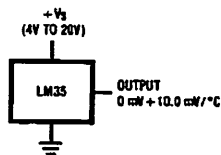
**TO-202
Plastic Package**



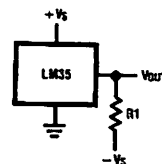
TL/H/5516-24

Order Number LM35DP
See NS Package Number P03A

Typical Applications



**FIGURE 1. Basic Centigrade
Temperature
Sensor ($+2^\circ\text{C}$ to $+150^\circ\text{C}$)**



Choose $R_1 = -V_S/50\ \mu\text{A}$

$V_{OUT} = +1,500\ \text{mV}$ at $+150^\circ\text{C}$
 $= +250\ \text{mV}$ at $+25^\circ\text{C}$
 $= -550\ \text{mV}$ at -55°C

**FIGURE 2. Full-Range Centigrade
Temperature Sensor**

TRI-STATE® is a registered trademark of National Semiconductor Corporation.

Absolute Maximum Ratings (Note 10)

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

| | |
|-------------------------------|-----------------|
| Supply Voltage | +35V to -0.2V |
| Output Voltage | +6V to -1.0V |
| Output Current | 10 mA |
| Storage Temp., TO-46 Package, | -60°C to +180°C |
| TO-92 Package, | -60°C to +150°C |
| SO-8 Package, | -65°C to +150°C |
| TO-202 Package, | -65°C to +150°C |

Lead Temp.:

| | |
|---|--------|
| TO-46 Package, (Soldering, 10 seconds) | 300°C |
| TO-92 Package, (Soldering, 10 seconds) | 260°C |
| TO-202 Package, (Soldering, 10 seconds) | +230°C |

SO Package (Note 12):

| | |
|---|-----------------|
| Vapor Phase (60 seconds) | 215°C |
| Infrared (15 seconds) | 220°C |
| ESD Susceptibility (Note 11) | 2500V |
| Specified Operating Temperature Range: T_{MIN} to T_{MAX} | |
| (Note 2) | |
| LM35, LM35A | -55°C to +150°C |
| LM35C, LM35CA | -40°C to +110°C |
| LM35D | 0°C to +100°C |

Electrical Characteristics (Note 1) (Note 6)

| Parameter | Conditions | LM35A | | | LM35CA | | | Units (Max.) |
|---|---|------------------------------|------------------------------|------------------------------|------------------------------|-----------------------|------------------------------|------------------------------|
| | | Typical | Tested Limit (Note 4) | Design Limit (Note 5) | Typical | Tested Limit (Note 4) | Design Limit (Note 5) | |
| Accuracy (Note 7) | $T_A = +25^\circ\text{C}$ | ± 0.2 | ± 0.5 | | ± 0.2 | ± 0.5 | | °C |
| | $T_A = -10^\circ\text{C}$ | ± 0.3 | | | ± 0.3 | | ± 1.0 | °C |
| | $T_A = T_{MAX}$ | ± 0.4 | ± 1.0 | | ± 0.4 | ± 1.0 | | °C |
| | $T_A = T_{MIN}$ | ± 0.4 | ± 1.0 | | ± 0.4 | | ± 1.5 | °C |
| Nonlinearity (Note 8) | $T_{MIN} \leq T_A \leq T_{MAX}$ | ± 0.18 | | ± 0.35 | ± 0.15 | | ± 0.3 | °C |
| Sensor Gain (Average Slope) | $T_{MIN} \leq T_A \leq T_{MAX}$ | +10.0 | +9.9, +10.1 | | +10.0 | | +9.9, +10.1 | mV/°C |
| Load Regulation (Note 3) $0 \leq I_L \leq 1$ mA | $T_A = +25^\circ\text{C}$ | ± 0.4 | ± 1.0 | | ± 0.4 | ± 1.0 | | mV/mA |
| | $T_{MIN} \leq T_A \leq T_{MAX}$ | ± 0.5 | | ± 3.0 | ± 0.5 | | ± 3.0 | mV/mA |
| Line Regulation (Note 3) | $T_A = +25^\circ\text{C}$ | ± 0.01 | ± 0.05 | | ± 0.01 | ± 0.05 | | mV/V |
| | $4V \leq V_S \leq 30V$ | ± 0.02 | | ± 0.1 | ± 0.02 | | ± 0.1 | mV/V |
| Quiescent Current (Note 9) | $V_S = +5V, +25^\circ\text{C}$ | 56 | 67 | | 56 | 67 | | μA |
| | $V_S = +5V$ | 105 | | 131 | 91 | | 114 | μA |
| | $V_S = +30V, +25^\circ\text{C}$ | 56.2 | 68 | | 56.2 | 68 | | μA |
| | $V_S = +30V$ | 105.5 | | 133 | 91.5 | | 116 | μA |
| Change of Quiescent Current (Note 3) | $4V \leq V_S \leq 30V, +25^\circ\text{C}$ | 0.2 | 1.0 | | 0.2 | 1.0 | | μA |
| | $4V \leq V_S \leq 30V$ | 0.5 | | 2.0 | 0.5 | | 2.0 | μA |
| Temperature Coefficient of Quiescent Current | | +0.39 | | +0.5 | +0.39 | | +0.5 | $\mu\text{A}/^\circ\text{C}$ |
| Minimum Temperature for Rated Accuracy | In circuit of Figure 1, $I_L = 0$ | +1.5 | | +2.0 | +1.5 | | +2.0 | °C |
| Long Term Stability | $T_J = T_{MAX}$, for 1000 hours | ± 0.08 | | | ± 0.08 | | | °C |

Note 1: Unless otherwise noted, these specifications apply: $-55^\circ\text{C} \leq T_J \leq +150^\circ\text{C}$ for the LM35 and LM35A; $-40^\circ\text{C} \leq T_J \leq +110^\circ\text{C}$ for the LM35C and LM35CA; and $0^\circ\text{C} \leq T_J \leq +100^\circ\text{C}$ for the LM35D. $V_S = +5\text{Vdc}$ and $I_{LOAD} = 50 \mu\text{A}$, in the circuit of Figure 2. These specifications also apply from $+2^\circ\text{C}$ to T_{MAX} in the circuit of Figure 1. Specifications in boldface apply over the full rated temperature range.

Note 2: Thermal resistance of the TO-46 package is $400^\circ\text{C}/\text{W}$, junction to ambient, and $24^\circ\text{C}/\text{W}$ junction to case. Thermal resistance of the TO-92 package is $180^\circ\text{C}/\text{W}$ junction to ambient. Thermal resistance of the small outline molded package is $220^\circ\text{C}/\text{W}$ junction to ambient. Thermal resistance of the TO-202 package is $65^\circ\text{C}/\text{W}$ junction to ambient. For additional thermal resistance information see table in the Applications section.

Electrical Characteristics (Note 1) (Note 6) (Continued)

| Parameter | Conditions | LM35 | | | LM35C, LM35D | | | Units (Max.) |
|---|---|---------------|--------------------------------|-----------------------|---------------|-----------------------|--------------------------------|------------------------------|
| | | Typical | Tested Limit (Note 4) | Design Limit (Note 5) | Typical | Tested Limit (Note 4) | Design Limit (Note 5) | |
| Accuracy, LM35, LM35C (Note 7) | $T_A = +25^\circ\text{C}$ | ± 0.4 | ± 1.0 | | ± 0.4 | ± 1.0 | | $^\circ\text{C}$ |
| | $T_A = -10^\circ\text{C}$ | ± 0.5 | | | ± 0.5 | | ± 1.5 | $^\circ\text{C}$ |
| | $T_A = T_{\text{MAX}}$ | ± 0.8 | ± 1.5 | | ± 0.8 | | ± 1.5 | $^\circ\text{C}$ |
| | $T_A = T_{\text{MIN}}$ | ± 0.8 | | ± 1.5 | ± 0.8 | | ± 2.0 | $^\circ\text{C}$ |
| Accuracy, LM35D (Note 7) | $T_A = +25^\circ\text{C}$ | | | | ± 0.6 | ± 1.5 | | $^\circ\text{C}$ |
| | $T_A = T_{\text{MAX}}$ | | | | ± 0.9 | | ± 2.0 | $^\circ\text{C}$ |
| | $T_A = T_{\text{MIN}}$ | | | | ± 0.9 | | ± 2.0 | $^\circ\text{C}$ |
| Nonlinearity (Note 8) | $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$ | ± 0.3 | | ± 0.5 | ± 0.2 | | ± 0.5 | $^\circ\text{C}$ |
| Sensor Gain (Average Slope) | $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$ | + 10.0 | + 9.8, + 10.2 | | + 10.0 | | + 9.8, + 10.2 | mV/ $^\circ\text{C}$ |
| Load Regulation (Note 3) $0 \leq I_L \leq 1 \text{ mA}$ | $T_A = +25^\circ\text{C}$ | ± 0.4 | ± 2.0 | | ± 0.4 | ± 2.0 | | mV/mA |
| | $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$ | ± 0.5 | | ± 5.0 | ± 0.5 | | ± 5.0 | mV/mA |
| Line Regulation (Note 3) | $T_A = +25^\circ\text{C}$ | ± 0.01 | ± 0.1 | | ± 0.01 | ± 0.1 | | mV/V |
| | $4\text{V} \leq V_S \leq 30\text{V}$ | ± 0.02 | | ± 0.2 | ± 0.02 | | ± 0.2 | mV/V |
| Quiescent Current (Note 9) | $V_S = +5\text{V}, +25^\circ\text{C}$ | 56 | 80 | | 56 | 80 | | μA |
| | $V_S = +5\text{V}$ | 105 | | 158 | 91 | | 138 | μA |
| | $V_S = +30\text{V}, +25^\circ\text{C}$ | 56.2 | 82 | | 56.2 | 82 | | μA |
| | $V_S = +30\text{V}$ | 105.5 | | 161 | 91.5 | | 141 | μA |
| Change of Quiescent Current (Note 3) | $4\text{V} \leq V_S \leq 30\text{V}, +25^\circ\text{C}$ | 0.2 | 2.0 | | 0.2 | 2.0 | | μA |
| | $4\text{V} \leq V_S \leq 30\text{V}$ | 0.5 | | 3.0 | 0.5 | | 3.0 | μA |
| Temperature Coefficient of Quiescent Current | | + 0.39 | | + 0.7 | + 0.39 | | + 0.7 | $\mu\text{A}/^\circ\text{C}$ |
| Minimum Temperature for Rated Accuracy | In circuit of <i>Figure 1</i> , $I_L = 0$ | + 1.5 | | + 2.0 | + 1.5 | | + 2.0 | $^\circ\text{C}$ |
| Long Term Stability | $T_J = T_{\text{MAX}}$, for 1000 hours | ± 0.08 | | | ± 0.08 | | | $^\circ\text{C}$ |

Note 3: Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

Note 4: Tested Limits are guaranteed and 100% tested in production.

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

Note 6: Specifications in boldface apply over the full rated temperature range.

Note 7: Accuracy is defined as the error between the output voltage and $10\text{mV}/^\circ\text{C}$ times the device's case temperature, at specified conditions of voltage, current, and temperature (expressed in $^\circ\text{C}$).

Note 8: Nonlinearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the device's rated temperature range.

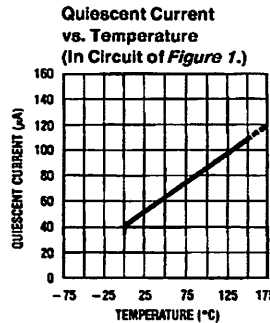
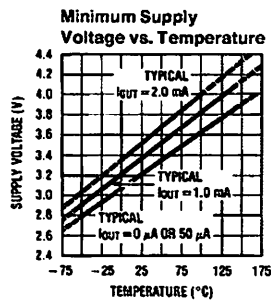
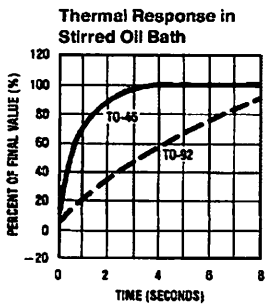
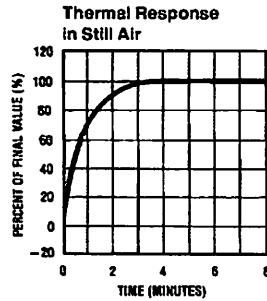
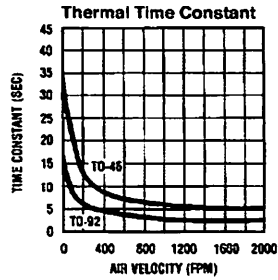
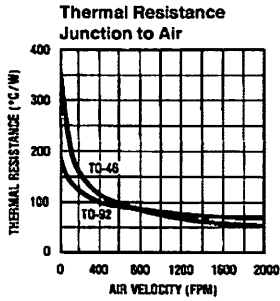
Note 9: Quiescent current is defined in the circuit of *Figure 1*.

Note 10: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions. See Note 1.

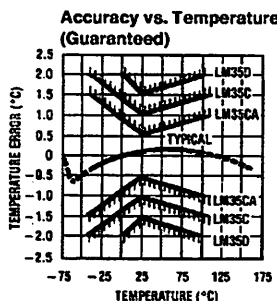
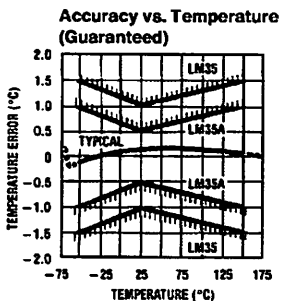
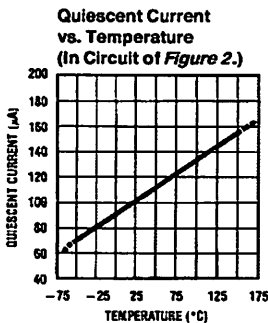
Note 11: Human body model, 100 pF discharged through a 1.5 k Ω resistor.

Note 12: See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" or the section titled "Surface Mount" found in a current National Semiconductor Linear Data Book for other methods of soldering surface mount devices.

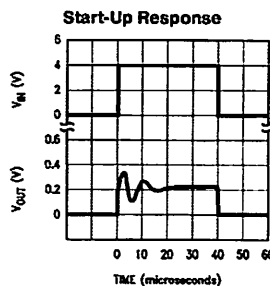
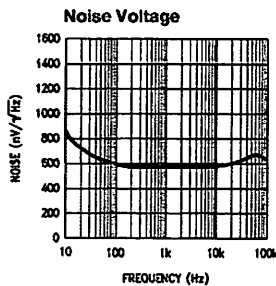
Typical Performance Characteristics



TL/H/5516-17



TL/H/5516-18



TL/H/5516-22

Applications

The LM35 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface and its temperature will be within about 0.01°C of the surface temperature.

This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature. This is especially true for the TO-92 plastic package, where the copper leads are the principal thermal path to carry heat into the device, so its temperature might be closer to the air temperature than to the surface temperature.

To minimize this problem, be sure that the wiring to the LM35, as it leaves the device, is held at the same temperature as the surface of interest. The easiest way to do this is to cover up these wires with a bead of epoxy which will insure that the leads and wires are all at the same temperature as the surface, and that the LM35 die's temperature will not be affected by the air temperature.

The TO-46 metal package can also be soldered to a metal surface or pipe without damage. Of course, in that case the V- terminal of the circuit will be grounded to that metal. Alternatively, the LM35 can be mounted inside a sealed-end metal tube, and can then be dipped into a bath or screwed into a threaded hole in a tank. As with any IC, the LM35 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperatures where condensation can occur. Printed-circuit coatings and varnishes such as Humiseal and epoxy paints or dips are often used to insure that moisture cannot corrode the LM35 or its connections.

These devices are sometimes soldered to a small lightweight heat fin, to decrease the thermal time constant and speed up the response in slowly-moving air. On the other hand, a small thermal mass may be added to the sensor, to give the steadiest reading despite small deviations in the air temperature.

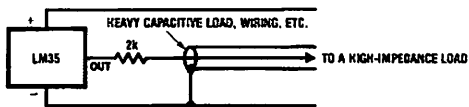
Temperature Rise of LM35 Due To Self-heating (Thermal Resistance)

| | TO-46, no heat sink | TO-46, small heat fin* | TO-92, no heat sink | TO-92, small heat fin** | SO-8 no heat sink | SO-8 small heat fin** | TO-202 no heat sink | TO-202 *** small heat fin |
|--|------------------------|---------------------------|------------------------|----------------------------|----------------------|--------------------------|------------------------|------------------------------|
| Still air | 400°C/W | 100°C/W | 180°C/W | 140°C/W | 220°C/W | 110°C/W | 85°C/W | 60°C/W |
| Moving air | 100°C/W | 40°C/W | 90°C/W | 70°C/W | 105°C/W | 90°C/W | 25°C/W | 40°C/W |
| Still oil | 100°C/W | 40°C/W | 90°C/W | 70°C/W | | | | |
| Stirred oil (Clamped to metal, infinite heat sink) | 50°C/W (24°C/W) | 30°C/W | 45°C/W | 40°C/W | | | | |
| | | | | | (55°C/W) | | (23°C/W) | |

* Wafertop type 201, or 1" disc of 0.020" sheet brass, soldered to case, or similar.

** TO-92 and SO-8 packages glued and leads soldered to 1" square of 1/16" printed circuit board with 2 oz. foil or similar.

Typical Applications (Continued)



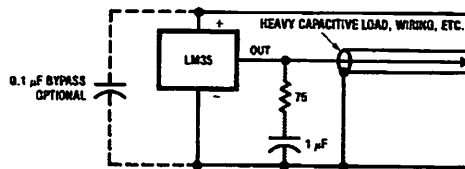
TL/H/5516-19

FIGURE 3. LM35 with Decoupling from Capacitive Load

CAPACITIVE LOADS

Like most micropower circuits, the LM35 has a limited ability to drive heavy capacitive loads. The LM35 by itself is able to drive 50 pf without special precautions. If heavier loads are anticipated, it is easy to isolate or decouple the load with a resistor; see Figure 3. Or you can improve the tolerance of capacitance with a series R-C damper from output to ground; see Figure 4.

When the LM35 is applied with a 200 Ω load resistor as shown in Figure 5, 6, or 8, it is relatively immune to wiring



TL/H/5516-20

FIGURE 4. LM35 with R-C Damper

capacitance because the capacitance forms a bypass from ground to input, not on the output. However, as with any linear circuit connected to wires in a hostile environment, its performance can be affected adversely by intense electromagnetic sources such as relays, radio transmitters, motors with arcing brushes, SCR transients, etc, as its wiring can act as a receiving antenna and its internal junctions can act as rectifiers. For best results in such cases, a bypass capacitor from V_{IN} to ground and a series R-C damper such as 75 Ω in series with 0.2 or 1 μ F from output to ground are often useful. These are shown in Figures 13, 14, and 16.

Typical Applications (Continued)

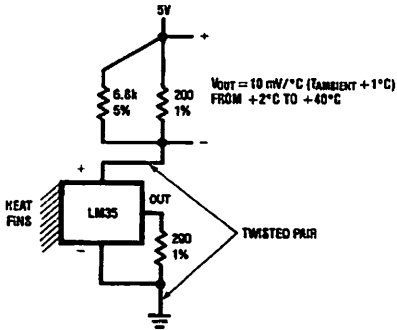


FIGURE 5. Two-Wire Remote Temperature Sensor (Grounded Sensor)
TL/H/5516-5

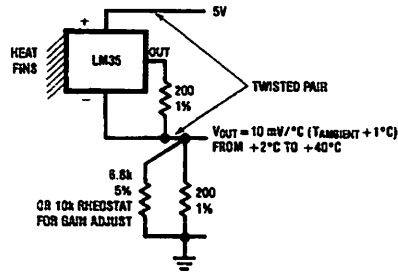


FIGURE 6. Two-Wire Remote Temperature Sensor (Output Referred to Ground)
TL/H/5516-6

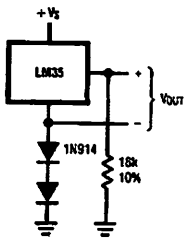


FIGURE 7. Temperature Sensor, Single Supply, -55°C to +150°C
TL/H/5516-7

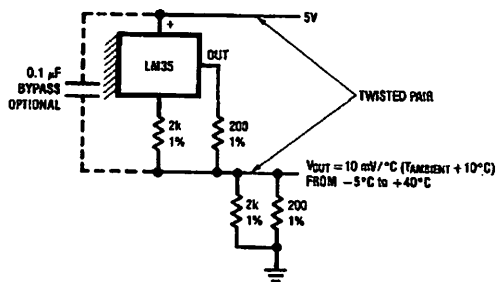


FIGURE 8. Two-Wire Remote Temperature Sensor (Output Referred to Ground)
TL/H/5516-8

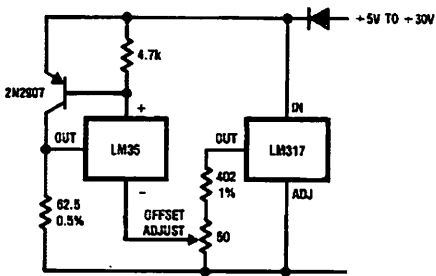


FIGURE 9. 4-To-20 mA Current Source (0°C to +100°C)
TL/H/5516-9

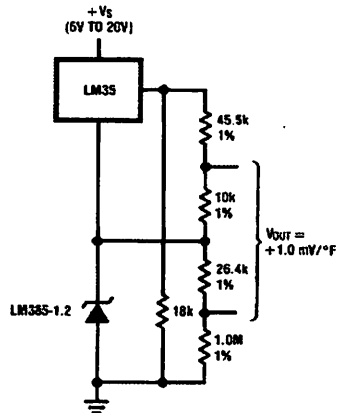
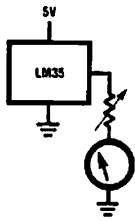


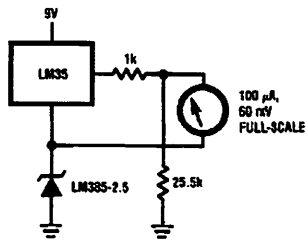
FIGURE 10. Fahrenheit Thermometer
TL/H/5516-10

Typical Applications (Continued)



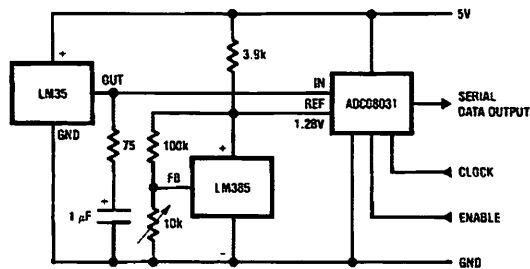
TL/H/5516-11

FIGURE 11. Centigrade Thermometer (Analog Meter)



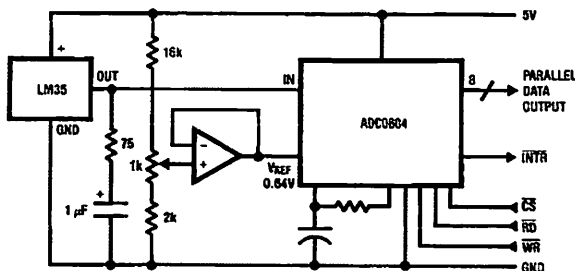
TL/H/5516-12

FIGURE 12. Expanded Scale Thermometer (50° to 80° Fahrenheit, for Example Shown)



TL/H/5516-13

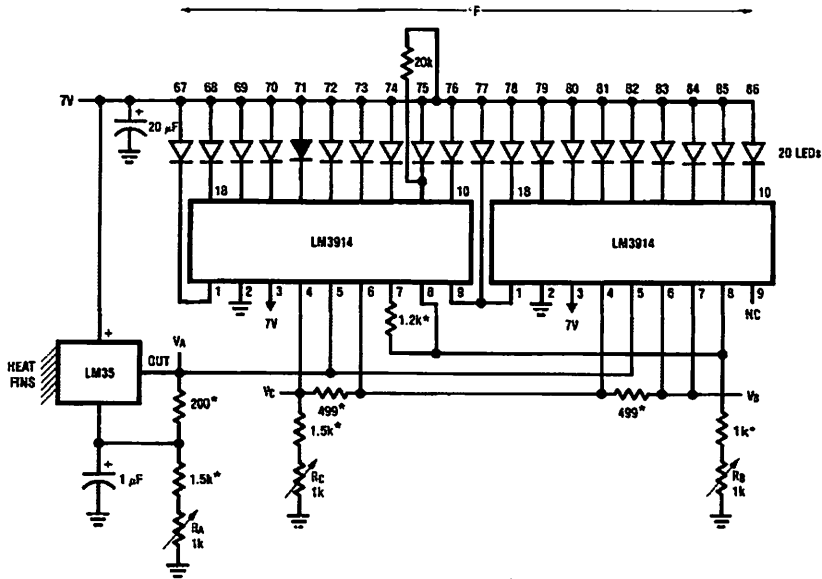
FIGURE 13. Temperature To Digital Converter (Serial Output) (+ 128°C Full Scale)



TL/H/5516-14

FIGURE 14. Temperature To Digital Converter (Parallel TRI-STATE® Outputs for Standard Data Bus to μP Interface) (128°C Full Scale)

Typical Applications (Continued)



- * = 1% or 2% film resistor
- Trim R_B for $V_B = 3.075V$
- Trim R_C for $V_C = 1.955V$
- Trim R_A for $V_A = 0.075V + 100mV/°C \times T_{ambient}$
- Example, $V_A = 2.275V$ at $22°C$

TL/H/5516-16

FIGURE 15. Bar-Graph Temperature Display (Dot Mode)

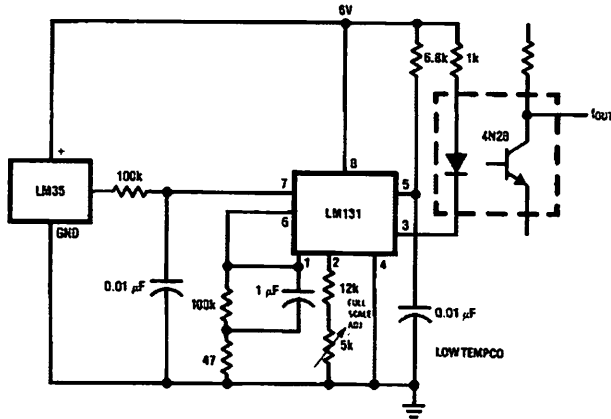
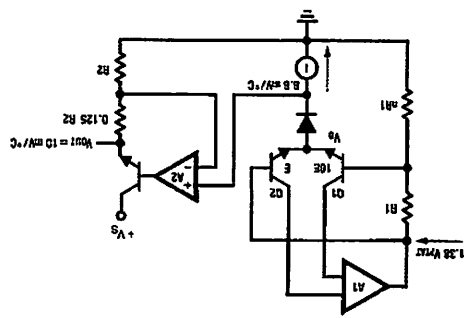


FIGURE 16. LM35 With Voltage-To-Frequency Converter And Isolated Output
($2°C$ to $+150°C$; 20 Hz to 1500 Hz)

TL/H/5516-15

Block Diagram



TL/H/5516-23

MM74HC164

8-Bit Serial-in/Parallel-out Shift Register

General Description

The MM74HC164 utilizes advanced silicon-gate CMOS technology. It has the high noise immunity and low consumption of standard CMOS integrated circuits. It also offers speeds comparable to low power Schottky devices.

This 8-Bit shift register has gated serial inputs and CLEAR. Each register bit is a D-type master/slave flip-flop. Inputs A & B permit complete control over the incoming data. A LOW at either or both inputs inhibits entry of new data and resets the first flip-flop to the low level at the next clock pulse. A high level on one input enables the other input which will then determine the state of the first flip-flop. Data at the serial inputs may be changed while the clock is HIGH or LOW, but only information meeting the setup and hold time requirements will be entered. Data is serially shifted in and out of the 8-Bit register during the positive going transi-

tion of the clock pulse. Clear is independent of the clock and accomplished by a low level at the CLEAR input.

The 74HC logic family is functionally as well as pin-out compatible with the standard 74LS logic family. All inputs are protected from damage due to static discharge by internal diode clamps to V_{CC} and ground.

Features

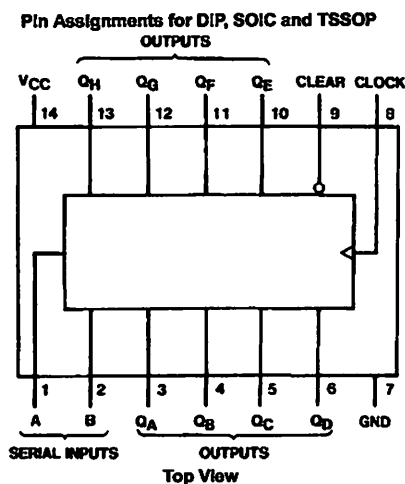
- Typical operating frequency: 50 MHz
- Typical propagation delay: 19 ns (clock to Q)
- Wide operating supply voltage range: 2-6V
- Low input current: 1 μA maximum
- Low quiescent supply current: 80 μA maximum (74HC Series)
- Fanout of 10 LS-TTL loads

Ordering Code:

| Order Number | Package Number | Package Description |
|--------------|----------------|--|
| MM74HC164M | M14A | 14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150" Narrow |
| MM74HC164MTC | MTC14 | 14-Lead Thin Shrink Small Outline Package (TSSOP), JEDEC MO-153, 4.4mm Wide |
| MM74HC164N | N14A | 14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide |

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

Connection Diagram

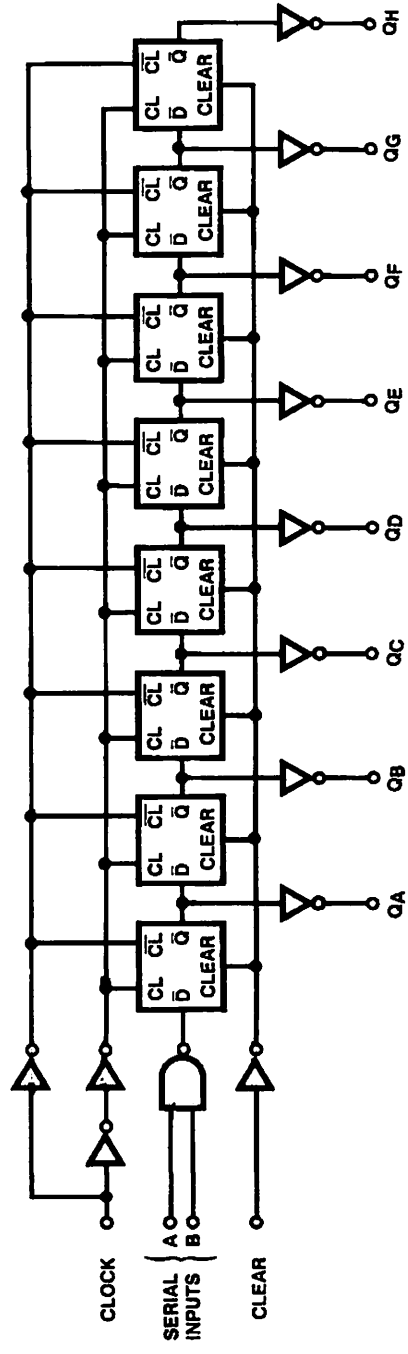


Truth Table

| Inputs | | | | Outputs | | | |
|--------|-------|---|---|-----------------|-----------------|-----|-----------------|
| Clear | Clock | A | B | Q _A | Q _B | ... | Q _H |
| L | X | X | X | L | L | | L |
| H | L | X | X | Q _{AO} | Q _{BO} | | Q _{HO} |
| H | ↑ | H | H | H | Q _{An} | | Q _{Gn} |
| H | ↑ | L | X | L | Q _{An} | | Q _{Gn} |
| H | ↑ | X | L | L | Q _{An} | | Q _{Gn} |

H = HIGH Level (steady state), L = LOW Level (steady state)
X = Irrelevant (any input, including transitions)
↑ = Transition from LOW-to-HIGH level.
Q_{AO}, Q_{BO}, Q_{HO} = the level of Q_A, Q_B, or Q_H, respectively, before the indicated steady state input conditions were established.
Q_{An}, Q_{Gn} = The level of Q_A or Q_G before the most recent ↑ transition of the clock; indicated a one-bit shift.

Logic Diagram



Absolute Maximum Ratings (Note 1)

(Note 2)

| | |
|--|-------------------------|
| Supply Voltage (V_{CC}) | -0.5 to +7.0V |
| DC Input Voltage (V_{IN}) | -1.5 to $V_{CC} + 1.5V$ |
| DC Output Voltage (V_{OUT}) | -0.5 to $V_{CC} + 0.5V$ |
| Clamp Diode Current (I_{IK}, I_{OK}) | ± 20 mA |
| DC Output Current, per pin (I_{OUT}) | ± 25 mA |
| DC V_{CC} or GND Current, per pin (I_{CC}) | ± 50 mA |
| Storage Temperature Range (T_{STG}) | -65°C to +150°C |
| Power Dissipation (P_D) | |
| (Note 3) | 600 mW |
| S.O. Package only | 500 mW |
| Lead Temperature (T_L) | |
| (Soldering 10 seconds) | 260°C |

Recommended Operating Conditions

| | | | |
|---|-----|----------|---------|
| Supply Voltage (V_{CC}) | Min | Max | Units |
| DC Input or Output Voltage (V_{IN}, V_{OUT}) | 2 | 6 | V |
| Operating Temperature Range (T_A) | 0 | V_{CC} | V |
| Input Rise or Fall Times (t_r, t_f) $V_{CC} = 2.0V$ | -40 | +85 | °C |
| $V_{CC} = 4.5V$ | | | 1000 ns |
| $V_{CC} = 6.0V$ | | | 500 ns |
| | | | 400 ns |

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: Power Dissipation temperature derating — plastic "N" package: -12 mW/°C from 65°C to 85°C.

DC Electrical Characteristics (Note 4)

| Symbol | Parameter | Conditions | V_{CC} | $T_A = 25^\circ C$ | | | | Units |
|----------|-----------------------------------|---|----------|--------------------|-------------------|-----------|-----------|---------|
| | | | | Typ | Guaranteed Limits | | | |
| V_{IH} | Minimum HIGH Level Input Voltage | | 2.0V | | 1.5 | 1.5 | 1.5 | V |
| | | | 4.5V | | 3.15 | 3.15 | 3.15 | V |
| | | | 6.0V | | 4.2 | 4.2 | 4.2 | V |
| V_{IL} | Maximum LOW Level Input Voltage | | 2.0V | | 0.5 | 0.5 | 0.5 | V |
| | | | 4.5V | | 1.35 | 1.35 | 1.35 | V |
| | | | 6.0V | | 1.8 | 1.8 | 1.8 | V |
| V_{OH} | Minimum HIGH Level Output Voltage | $V_{IN} = V_{IH}$ or V_{IL} $ I_{OUT} \leq 20 \mu A$ | 2.0V | 2.0 | 1.9 | 1.9 | 1.9 | V |
| | | | 4.5V | 4.5 | 4.4 | 4.4 | 4.4 | V |
| | | | 6.0V | 6.0 | 5.9 | 5.9 | 5.9 | V |
| | | $V_{IN} = V_{IH}$ or V_{IL} $ I_{OUT} \leq 4.0$ mA $ I_{OUT} \leq 5.2$ mA | 4.5V | 4.2 | 3.98 | 3.84 | 3.7 | V |
| | | | 6.0V | 5.7 | 5.48 | 5.34 | 5.2 | V |
| | | | | | | | | |
| V_{OL} | Maximum LOW Level Output Voltage | $V_{IN} = V_{IH}$ or V_{IL} $ I_{OUT} \leq 20 \mu A$ | 2.0V | 0 | 0.1 | 0.1 | 0.1 | V |
| | | | 4.5V | 0 | 0.1 | 0.1 | 0.1 | V |
| | | | 6.0V | 0 | 0.1 | 0.1 | 0.1 | V |
| | | $V_{IN} = V_{IH}$ or V_{IL} $ I_{OUT} \leq 4.0$ mA $ I_{OUT} \leq 5.2$ mA | 4.5V | 0.2 | 0.26 | 0.33 | 0.4 | V |
| | | | 6.0V | 0.2 | 0.26 | 0.33 | 0.4 | V |
| | | | | | | | | |
| I_{IN} | Maximum Input Current | $V_{IN} = V_{CC}$ or GND | 6.0V | | ± 0.1 | ± 1.0 | ± 1.0 | μA |
| I_{CC} | Maximum Quiescent Supply Current | $V_{IN} = V_{CC}$ or GND $I_{OUT} = 0 \mu A$ | 6.0V | | 8.0 | 80 | 160 | μA |

Note 4: For a power supply of 5V $\pm 10\%$ the worst case output voltages (V_{OH} and V_{OL}) occur for HC at 4.5V. Thus the 4.5V values should be used when designing with this supply. Worst case V_{IH} and V_{IL} occur at $V_{CC} = 5.5V$ and 4.5V respectively. (The V_{IH} value at 5.5V is 3.85V.) The worst case leakage current (I_{IN} , I_{CC} , and I_{OZ}) occur for CMOS at the higher voltage and so the 6.0V values should be used.

1942

| DATE | DESCRIPTION | AMOUNT |
|-------|-------------|--------|
| 1/1 | Balance | 100.00 |
| 1/15 | ... | ... |
| 2/1 | ... | ... |
| 2/15 | ... | ... |
| 3/1 | ... | ... |
| 3/15 | ... | ... |
| 4/1 | ... | ... |
| 4/15 | ... | ... |
| 5/1 | ... | ... |
| 5/15 | ... | ... |
| 6/1 | ... | ... |
| 6/15 | ... | ... |
| 7/1 | ... | ... |
| 7/15 | ... | ... |
| 8/1 | ... | ... |
| 8/15 | ... | ... |
| 9/1 | ... | ... |
| 9/15 | ... | ... |
| 10/1 | ... | ... |
| 10/15 | ... | ... |
| 11/1 | ... | ... |
| 11/15 | ... | ... |
| 12/1 | ... | ... |
| 12/15 | ... | ... |
| 12/31 | ... | ... |

1943

| DATE | DESCRIPTION | AMOUNT |
|-------|-------------|--------|
| 1/1 | Balance | 100.00 |
| 1/15 | ... | ... |
| 2/1 | ... | ... |
| 2/15 | ... | ... |
| 3/1 | ... | ... |
| 3/15 | ... | ... |
| 4/1 | ... | ... |
| 4/15 | ... | ... |
| 5/1 | ... | ... |
| 5/15 | ... | ... |
| 6/1 | ... | ... |
| 6/15 | ... | ... |
| 7/1 | ... | ... |
| 7/15 | ... | ... |
| 8/1 | ... | ... |
| 8/15 | ... | ... |
| 9/1 | ... | ... |
| 9/15 | ... | ... |
| 10/1 | ... | ... |
| 10/15 | ... | ... |
| 11/1 | ... | ... |
| 11/15 | ... | ... |
| 12/1 | ... | ... |
| 12/15 | ... | ... |
| 12/31 | ... | ... |

AC Electrical Characteristics $V_{CC} = 5V, T_A = 25^\circ C, C_L = 15 \text{ pF}, t_r = t_f = 6 \text{ ns}$

| Symbol | Parameter | Conditions | Typ | Guaranteed Limit | Units |
|--------------------|--|------------|-----|------------------|-------|
| f_{MAX} | Maximum Operating Frequency | | | 30 | MHz |
| t_{PHL}, t_{PLH} | Maximum Propagation Delay, Clock to Output | | 19 | 30 | ns |
| t_{PHL}, t_{PLH} | Maximum Propagation Delay, Clear to Output | | 23 | 35 | ns |
| t_{REM} | Minimum Removal Time, Clear to Clock | | -2 | 0 | ns |
| t_S | Minimum Setup Time Data to Clock | | 12 | 20 | ns |
| t_H | Minimum Hold Time Clock to Data | | 1 | 5 | ns |
| t_W | Minimum Pulse Width Clear or Clock | | 10 | 18 | ns |

AC Electrical Characteristics $C_L = 50 \text{ pF}, t_r = t_f = 6 \text{ ns}$ (unless otherwise specified)

| Symbol | Parameter | Conditions | V_{CC} | $T_A = 25^\circ C$ | | | Units | |
|--------------------|--|---------------|----------|--------------------|-------------------|------|-------|----|
| | | | | Typ | Guaranteed Limits | | | |
| f_{MAX} | Maximum Operating Frequency | | 2.0V | 5 | 4 | 3 | MHz | |
| | | | 4.5V | 27 | 21 | 18 | MHz | |
| | | | 6.0V | 31 | 24 | 20 | MHz | |
| t_{PHL}, t_{PLH} | Maximum Propagation Delay, Clock to Output | | 2.0V | 115 | 175 | 218 | 254 | ns |
| | | | 4.5V | 13 | 35 | 44 | 51 | ns |
| | | | 6.0V | 20 | 30 | 38 | 44 | ns |
| t_{PHL}, t_{PLH} | Maximum Propagation Delay, Clear to Output | | 2.0V | 140 | 205 | 256 | 297 | ns |
| | | | 4.5V | 28 | 41 | 51 | 59 | ns |
| | | | 6.0V | 24 | 35 | 44 | 51 | ns |
| t_{REM} | Minimum Removal Time Clear to Clock | | 2.0V | -7 | 0 | 0 | 0 | ns |
| | | | 4.5V | -3 | 0 | 0 | 0 | ns |
| | | | 6.0V | -2 | 0 | 0 | 0 | ns |
| t_S | Minimum Setup Time Data to Clock | | 2.0V | 25 | 100 | 125 | 150 | ns |
| | | | 4.5V | 14 | 20 | 25 | 30 | ns |
| | | | 6.0V | 12 | 17 | 21 | 25 | ns |
| t_H | Minimum Hold Time Clock to Data | | 2.0V | -2 | 5 | 5 | 5 | ns |
| | | | 4.5V | 0 | 5 | 5 | 5 | ns |
| | | | 6.0V | 1 | 5 | 5 | 5 | ns |
| t_W | Minimum Pulse Width Clear or Clock | | 2.0V | 22 | 80 | 100 | 120 | ns |
| | | | 4.5V | 11 | 16 | 20 | 24 | ns |
| | | | 6.0V | 10 | 14 | 18 | 20 | ns |
| t_{THL}, t_{TLH} | Maximum Output Rise and Fall Time | | 2.0V | 75 | 95 | 110 | ns | |
| | | | 4.5V | 15 | 19 | 22 | ns | |
| | | | 6.0V | 13 | 16 | 19 | ns | |
| t_r, t_f | Maximum Input Rise and Fall Time | | 2.0V | 1000 | 1000 | 1000 | ns | |
| | | | 4.5V | 500 | 500 | 500 | ns | |
| | | | 6.0V | 400 | 400 | 400 | ns | |
| C_{PD} | Power Dissipation Capacitance (Note 5) | (per package) | 5.0V | 150 | | | pF | |
| C_{IN} | Maximum Input Capacitance | | | 5 | 10 | 10 | 10 | pF |

Note 5: C_{PD} determines the no load dynamic power consumption, $P_D = C_{PD} V_{CC}^2 f + I_{CC} V_{CC}$, and the no load dynamic current consumption, $I_S = C_{PD} V_{CC} f + I_{CC}$.