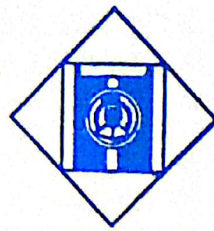


PENGHITUNG VOLUME AIR DENGAN MENGGUNAKAN MIKROKONTROLER 8031



TUGAS AKHIR

Disusun oleh :

NAMA : MIZWAR YS
N.R.P : 011880055
N.I.R.M : 883206710250052

**JURUSAN TEKNIK ELEKTRO - PEMINATAN ELEKTRONIKA
FAKULTAS TEKNIK INDUSTRI
INSTITUT TEKNOLOGI INDONESIA
SERPONG 1996**

PENGHITUNG VOLUME AIR DENGAN MENGGUNAKAN MIKROKONTROLER 8031

TUGAS AKHIR

Diajukan untuk memenuhi persyaratan kurikulum sarjana Strata Satu (S1)
Jurusan Teknik Elektro - Fakultas Teknologi Industri
Institut Teknologi Indonesia

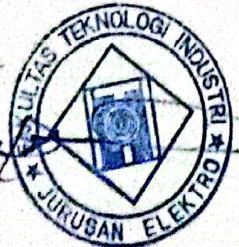
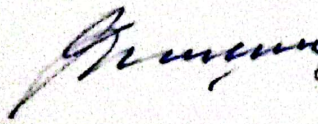
Disusun oleh :

MIZWAR YS

Jurusan Teknik Elektro - Peminatan Elektronika
Fakultas Teknologi Industri
Institut Teknologi Indonesia

Serpong, November 1996

Mengetahui,



Ir. Imam Sugandi

Ketua Jurusan

Menyetujui,



Ir. Sutisno M.Sc. MSTs

Dosen Pembimbing

ABSTRAK

Tugas Akhir ini membahas tentang perancangan alat penghitung volume air dengan menggunakan mikrokontroler 8031. Konstruksi alat ini terdiri dari : sensor aliran fluida, mikrokontroler 8031, valve selenoid dan gelas ukur.

Penghitung volume air ini menggunakan sensor aliran fluida yang di dalamnya terdapat baling-baling yang digerakkan oleh aliran fluida, sehingga sensor mengeluarkan pulsa yang akan dikirim ke alat pengolah data (mikrokontroler 8031).

Hasil perhitungan ditampilkan pada layar peraga dengan tampilan tujuh ruas, angka yang ditampilkan pada papan tampilan akan disertakan dengan perhitungan air pada gelas ukur.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah swt, yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan makalah tugas akhir ini, yang dimaksudkan guna melengkapi persyaratan mengikuti sidang sarjana strata 1 (S-1) di jurusan Teknik Elektro peminatan Elektronika, Fakultas Teknologi Industri, Institut Teknologi Indonesia, Serpong.

Adapun makalah tugas akhir ini berjudul :

“ Penghitung Volume Air Dengan Menggunakan Mikrokontroler 8031 ”

Dalam menyelesaikan makalah tugas akhir ini, penulis memperoleh banyak bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada :

- * Bapak Ir. Imam Sugandi, selaku ketua jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Indonesia.
- * Ibu Dr. Ir. Fatimah Z.P. selaku koordinator tugas akhir jurusan Teknik Elektro, peminatan Elektronika, Fakultas Teknologi Industri, Institut Teknologi Indonesia.
- * Bapak Ir. Sutisno M.Sc, selaku dosen pembimbing yang telah memberikan bimbingan dan saran kepada penulis dalam penyusunan makalah tugas akhir ini.
- * Orang tua dan saudara serta saudari penulis yang tercinta, yang telah banyak memberikan dorongan moril dan materiil.

Rekan-rekan mahasiswa seperti : Ijoel, Ir. Arief, Khi-q, yose, Budi, Halik dan rekan-rekan elektro yang lain yang telah memberikan dukungan kepada penulis, baik secara langsung maupun secara tidak langsung, serta rekan serumah Ajie, Bonny, Ryan, Teguh, Wadie.

Akhir kata, semoga makalah tugas akhir yang penulis anggap masih banyak kekurangan ini, dapat memberikan manfaat bagi para pembaca sekalian.

Serpong, November 1996

Penulis

DAFTAR ISI

Lembar Pengesahan	
Abstrak	i
Kata Pengantar	ii
Daftar isi	iv
BAB I PENDAHULUAN	I
1.1 Latar Belakang Masalah	1
1.2 Tujuan	2
1.3 Batasan Masalah	2
1.4 Sistematika Penulisan	3
BAB II LANDASAN TEORI	5
2.1 Fluida	5
2.2 Momentum Baling-baling	5
2.2.1 Hubungan Aliran Fluida Dengan Pulsa	6
2.3 Opto-Coupler	8
2.4 Sensor Turbin	9
BAB III PERANCANGAN DAN PEMBUATAN ALAT	11

3.1. Uraian Umum	11
3.1.1. Aliran Fluida	12
3.2. Rangkaian Sensor Fluida Cair	13
3.3. Mikrokontroler 8031	17
3.3.1. Memory	20
3.3.1.1 Internal RAM	20
3.3.1.2 Special Function Register	21
3.3.1.3. Memory Program	21
3.3.2. Arsitektur Internal	21
3.3.2.1 Akumulator	21
3.3.2.2. Base Register	22
3.3.2.3. Register Fungsi Umum	22
3.3.2.4. Program Status Word	22
3.3.2.5 Stack Pointer	23
3.3.2.6 Port-Port I/O	23
3.3.2.7. Interupsi	24
3.3.3. Rangkaian Mikrokontroler 8031	24
3.3.3.1 Rangkaian Clock	25
3.3.3.2 Rangkaian Power On Reset	26
3.3.3.3 Rangkaian Dekoder	28
3.3.3.4 Rangkaian Latch	30
3.3.3.5 Rangkaian Memory	31

3.3.3.6. Programmable Peripheral Interface.....	32
3.3.3.7. Rangkaian Papan Kunci (key pad).....	37
3.3.4. Rangkaian Peraga Tujuh Ruas.....	39
3.3.5. Rangkaian Valve Selenoid.....	40
3.4. Perangkat Lunak.....	41
3.4.1. Program Utama.....	42
3.4.2. Subrutin Pelayanan Interupsi Papan Kunci.....	44
 BAB IV PENGUJIAN ALAT.....	 45
4.1. Langkah Penggunaan Alat.....	45
4.1.1. Pemilihan Mode.....	45
4.1.2. Proses Memasukan Data.....	46
4.2. Pelaksanaan Pengukuran.....	46
4.3. Hasil Pengukuran.....	47
 BAB V KESIMPULAN.....	 49
DAFTAR PUSTAKA.....	50
LAMPIRAN.....	51
Lampiran A : Skema Rangkaian	
Lampiran B : Perangkat Lunak	
Lampiran C : Foto-foto Sistem	
Lampiran D : Data Sheet Sensor	

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Dewasa ini perkembangan teknologi begitu pesat, seperti yang kita ketahui terlebih dalam perkembangan teknologi elektronika. Kebanyakan seluruh produk yang dihasilkan pada dewasa ini telah bersifat otomatis, misalnya mesin cuci otomatis, pagar rumah otomatis, kompor listrik, dan pengatur pendingin ruangan otomatis, serta banyak lagi yang lainnya.

Oleh karena itu disini penulis bermaksud untuk membuat suatu perangkat yang sederhana untuk penghitung volume air atau banyaknya air yang kita perlukan dengan menggunakan sensor yang dihitung oleh mikrokontroler 8031, sesuai dengan kalibrasi dan dapat dikontrol sesuai dengan yang diinginkan pada penggunaan papan kunci (*key pad*) dan ditampilkan pada tujuh ruas (*seven segment*).

Satu hal yang harus diingat, bahwa sensor yang digunakan hanya untuk jenis cairan yaitu yang berupa air. Oleh karena itu juga penghitung volume air ini diharapkan dapat membantu tugas atau hal yang berhubungan dengan hitungan atau ukuran volume dari beberapa jenis satuan volume, seperti dalam satuan milliliter dan liter serta gelas ukur dan satuan volume lainnya, sehingga penghitung volume air ini dapat dimanfaatkan untuk penggunaan satuan volume yang relatif kecil, seperti di laboratorium.

1.2. Tujuan

Perancangan perangkat keras dan perangkat lunak bertujuan untuk membuat alat yang dapat digunakan untuk mengukur volume air, dapat diset sesuai dengan jumlah volume air yang dibutuhkan.

1.3. Batasan Masalah

Peralatan yang dibuat mencakup perangkat keras dan perangkat lunak, sistem perangkat kerasnya adalah rangkaian minimum 8031. Layar peraga berupa tujuh ruas (*seven segment*) serta papan kunci (*key pad*), sedangkan perangkat lunaknya adalah realisasi dari algoritma pengontrol volume dengan menggunakan bahasa pemrograman tingkat rendah (*Assembly 8031*), yang digunakan sebagai program

untuk melayani papan kunci (*key pad*), layar peraga (*seven segment*) yang menghitung banyaknya pulsa dari sensor dan program untuk melayani pemrosesan data.

Pembahasan prinsip kerja perangkat keras dilakukan dengan mengacu pada diagram blok dari sistem yang dibuat, sedangkan untuk perangkat lunak dilakukan pada pembahasan unjuk kerja alat ini, dibahas berdasarkan hasil yang didapat melalui uji coba alat. Kalibrasi dilakukan terhadap alat penghitung yang telah dikalibrasi, sehingga alat yang dibuat ini dapat melakukan penghitungan volume sebenarnya.

1.4. Sistematika Penulisan

Metodologi penulisan yang dilakukan dalam pembuatan makalah tugas akhir ini merupakan studi kepustakaan, yaitu mengenai hal - hal yang terkait dengan konsep alat yang dimaksud dalam materi penulisan serta menjelaskan cara kerja alat yang dibuat. Untuk mempermudah makalah ini, ditulis dalam 5 (lima) bab, yaitu :

BAB I : PENDAHULUAN

Bab ini menguraikan latar belakang masalah, tujuan, pembatasan masalah, dan sistematika penulisan

BAB II : LANDASAN TEORI

Bab ini berisi mengenai teori pustaka yang berhubungan dengan fluida, serta jenis sensor yang digunakan

BAB III : PERANCANGAN DAN PEMBUATAALAT

Bab ini berisi realisasi perangkat keras dan perangkat lunak alat kontrol volume air, yang terdiri dari pembahasan mengenai mikrokontroler 8031 itu sendiri, rancangan dan implementasi sistem yang meliputi perancangan CPUmikrokontroler 8031, layar peraga rangkaian key pad (papan kunci), serta pembahasan diagram alur program dan juga perhitungan banyaknya pulsa dengan volume air yang mengalir.

BAB IV : PENGUJIAN ALAT

Bab ini menjelaskan tentang pengujian sistem secara keseluruhan.

BAB V : KESIMPULAN

Kesimpulan merupakan penutup makalah, yang mana berisi kesimpulan mengenai hal-hal yang penting sehubungan dengan alat penghitung volume air, dan hasil uji coba.

BAB II

LANDASAN TEORI

2.1. Fluida

Fluida adalah merupakan suatu zat yang berubah bentuk secara kontinu, bila terkena tegangan geser. Fluida terbagi dari dua jenis, yaitu : fluida cair dan fluida gas, sedangkan ciri -ciri dari aliran dapat diklasifikasikan dalam beberapa cara yaitu cara turbulen laminar nyata, ideal, mampu balik, tak mampu balik, stedi dan tak stedi.

2.2. Momentum Baling - Baling

Baling - baling dari sensor ini berfungsi untuk mengubah momentum fluida yaitu tempat baling - baling terendam dengan mengembangkan gaya dorong yang digunakan untuk pergerakan, sedangkan baling - baling tersebut akan tetap berada

pada porosnya atau tetap ditempatnya didalam suatu aliran. Faktor tekanan fluida akan mempengaruhi kecepatan dari aliran fluida yang akan menimbulkan perputaran dari baling - baling. Aliran air yang melewati atau yang melintasi baling - baling mengalami kecepatan yang tetap.

2.2.1 Hubungan aliran Fluida dengan Pulsa Frekuensi

Fluida mengalir melalui pipa sensor, sehingga memutarakan baling-baling yang ada didalamnya, yang akan menghasilkan pulsa-pulsa frekuensi, dimana hubungan antara kecepatan aliran volume air dengan pulsa frekwensi adalah :

$$Q = V \times A \dots\dots\dots (1)$$

Dimana :

Q = Debit aliran air (m^3 / detik)

V = Kecepatan aliran air/fluida (m / detik)

A = Luas penampang pipa (m^2)

Jika sensor aliran fluida yang berupa baling-baling dihubungkan dengan aliran air yang mengalir dengan kecepatan tertentu, maka baling-baling akan berputar dengan kecepatan putaran tertentu pula.

Hubungan antara volume air dengan kecepatan putaran baling-baling adalah sebagai berikut :

$$U/t = C \times U \dots\dots\dots (2)$$

dimana

U = Volume air (m^3)

t = Waktu (detik)

Dari persamaan (1), hubungan antara kecepatan aliran air dengan kecepatan putar baling-baling adalah

$$V \times A = C \times U \dots\dots\dots (3)$$

atau

$$V = C1 \times U \dots\dots\dots (4)$$

dimana :

C = konstanta (m^3/rad)

U = kecepatan putar baling-baling ($rad/detik$)

$C1$ = konstanta (m/rad)

Selanjutnya baling-baling yang berputar dengan kecepatan putar tertentu akan memutar piringan sensor yaitu piringan yang bercelah N buah.

Hubungan kecepatan putar dengan pulsa - pulsa frekuensi adalah sebagai berikut :

$$Q = V \times A \times C \dots\dots\dots (5)$$

$$Q = \tau \times f \dots\dots\dots (6)$$

$$V \times A \times C = \frac{\tau \times f}{K} \dots\dots\dots (7)$$

dimana :

τ = konstanta waktu (60 detik/m)

K = pulsa / satuan volume (pulsa/m³)

Akhirnya diperoleh hubungan antara kecepatan dengan frekuensi yaitu sebagai berikut :

$$V = K1 \times N \times f \dots\dots\dots(8)$$

$$V = K2 \times f \dots\dots\dots(9)$$

dimana;

f = frekuensi (Hz)

r = jari - jari sensor aliran fluida (cm)

Ψ = kecepatan putar sensor aliran fluida (rad/detik)

$K2$ = konstanta (cm) yaitu jarak antar celah

N = jumlah celah pada piringan sensor

Hal ini menunjukkan, bahwa kecepatan aliran berbanding lurus dengan frekwensi yang dihasilkan oleh putaran baling - baling.

Misalkan : Jumlah celah pada piringan sensor sebanyak N dan jari-jari sensor aliran fluida = r , maka nilai dari $K2$ adalah :

$$K2 = 2\pi r/N = \text{ millimeter.}$$

2.3 Opto-Coupler

Penghitung volume air ini, menggunakan sensor dengan sistem opto-coupler. Dengan menempatkan piringan diantara celah opto-coupler. Didalam opto-coupler itu sendiri terdapat pemancar (*transmitter*) dan penerima (*receiver*). Pemancar yang berupa LED dan penerima berupa photo-dioda.

Sinar yang seharusnya diterima oleh photo-dioda kemudian terpotong - potong oleh putaran piringan yang bercelah tersebut, sehingga menghasilkan pulsa - pulsa sebagai data penghitungan.

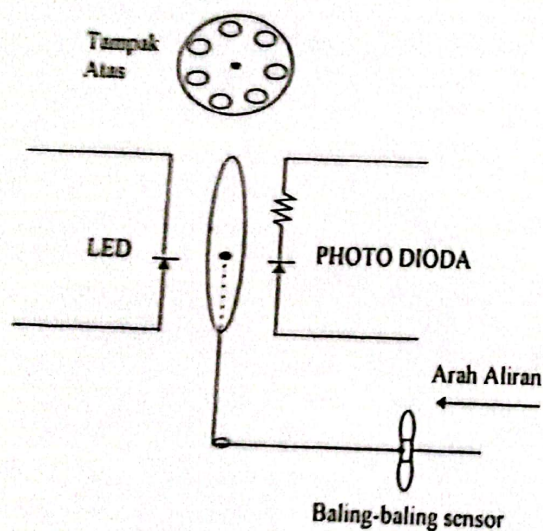
Apabila penerima menerima cahaya atau sinar yang berasal dari pemancar maka resistansinya akan mengecil, sehingga arus dapat melewati penerima atau photo dioda tersebut. Dan sebaliknya resistansinya akan membesar apabila tidak menerima cahaya atau sinar dari pemancar (LED), sehingga arus tidak dapat melewati penerima atau photo-dioda tersebut.

2.4 Sensor Turbin

Sensor aliran merupakan bagian yang utama dalam penghitungan atau pengukuran volume air. Jenis sensor aliran yang mempunyai karakteristik serta kelebihan terhadap kemampuan kecepatan rata-rata aliran atau debit aliran.

Sensor aliran fluida yang dibahas pada makalah ini adalah sensor aliran tipe baling-baling atau sensor turbin.

Fungsi sebuah baling-baling pada sensor turbin ialah untuk menggerakkan piringan yang bercelah, piringan bercelah berada di antara LED dan photo dioda dan dihubungkan ke rangkaian sensor (dibahas pada Bab III). Rata-rata aliran pada sensor turbin berada pada selang aliran tertentu, artinya baling-baling akan berputar dengan kecepatan tertentu sesuai rata-rata aliran karakteristik sensor, bila rata-rata aliran tidak sesuai dengan karakteristik sensor maka baling-baling tidak akan berputar atau mengalami kerusakan secara fisik.



Gambar 2.1 Baling-baling sensor.

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

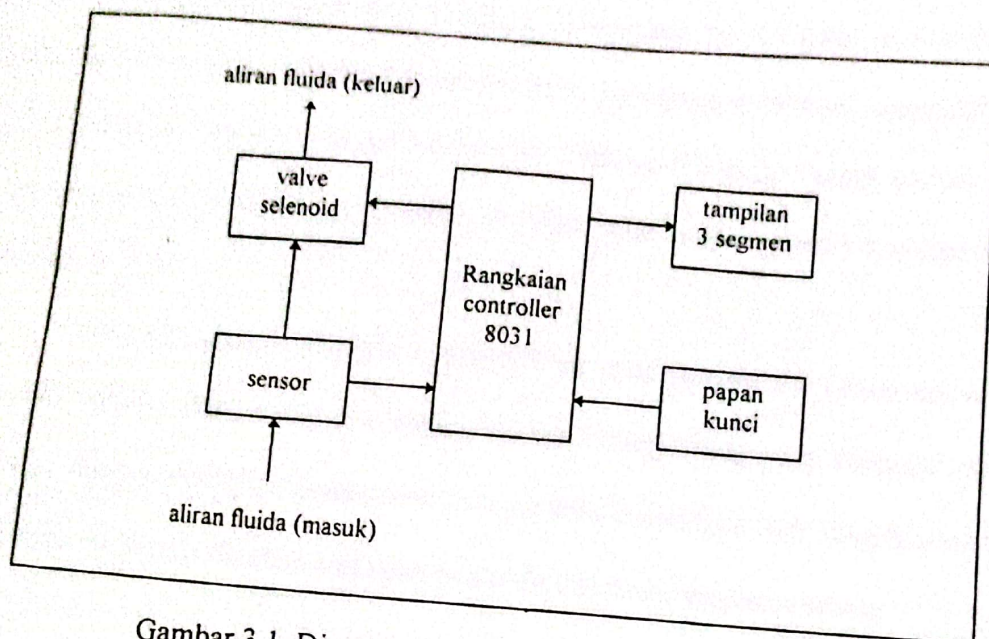
3.1. Uraian Umum

Sistem penghitung volume air ini, terdiri dari rangkaian mikrokontroler 8031, peraga tujuh ruas (*seven segment*) dengan bantuan PPI 8255, rangkaian penyimpan data menggunakan Internal RAM. Diagram blok sistem penghitung volume air ditunjukkan pada gambar 3.1.

Perangkat keras yang akan dibahas dalam sistem penghitung volume air ini meliputi :

- Sensor fluida cair
- Rangkaian mikrokontroler 8031
- Rangkaian peraga tujuh ruas

- Rangkaian papan kunci (*key pad*)
- Rangkaian valve selenoid



Gambar 3.1. Diagram Blok Penghitung Volume Air

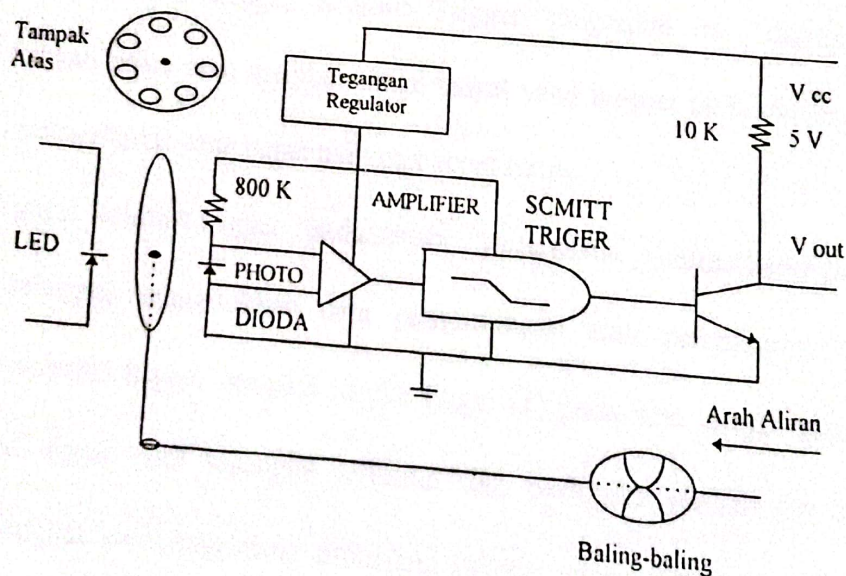
3.1.1. Aliran Fluida

Aliran air yang mengalir melalui saluran pipa sensor, hanya terbatas pada jenis fluida dan temperatur. Menurut data sensor yang digunakan, bahwa hanya jenis cairan yang dapat digunakan, sedangkan untuk jenis gas tidak dapat digunakan sensor ini. Untuk aliran cairan yang digunakan mempunyai kadar asam yang lemah, diantaranya adalah air (H_2O), Hidrokarbon, cairan Oxihidrocarbon, cairan asam organik pada konsentrasi tertentu.

3.2. Rangkaian Sensor fluida Cair

Sensor ini termasuk jenis kerangka turbin yang berada pada titik vertikal, input dari sensor jenis ini yaitu adanya aliran air, sedangkan outputnya menghasilkan pulsa frekuensi. Didalam kerangka sensor ini, terdiri dari baling - baling, lempengan tipis yang berongga, LED, photo-dioda, amplifier, serta rangkaian schmitt trigger dan resistor.

Cara kerja sensor ini, seperti yang terlihat pada gambar 3.2. Yaitu air yang mengalir melalui pipa sensor dan menggerakkan baling-baling yang berputar pada porosnya yang dihubungkan pada poros piringan atau lempengan tipis yang berongga, lempengan ini berada diantara LED dan photo-dioda pada rangkaian sensor.

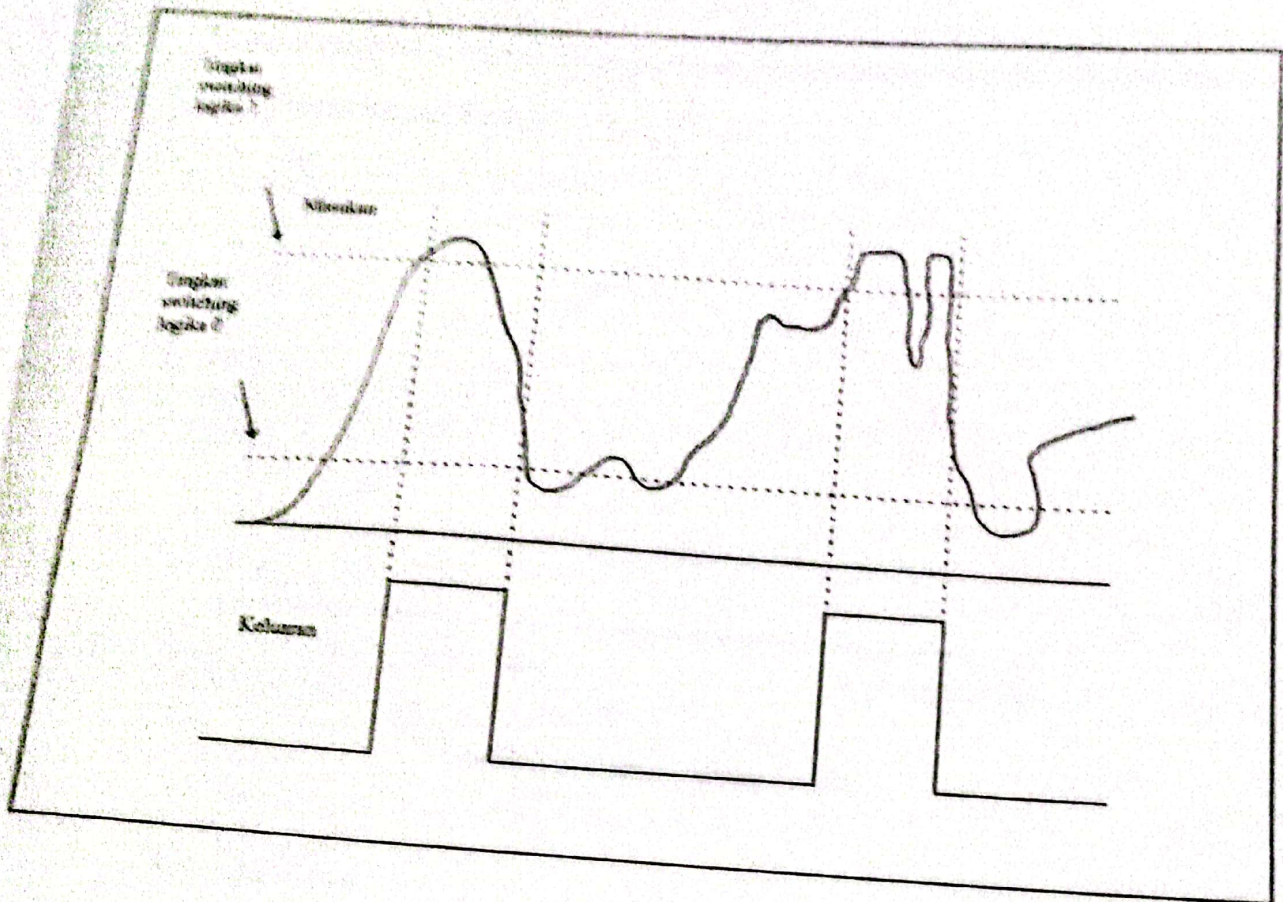


Gambar 3.2. Rangkaian Sensor

Lempengan berongga dihimpit oleh LED dan photo-dioda, bila sinar yang berasal dari led tepat menembus lubang, maka sinar akan diterima di photo-dioda dan akan terjadi hubungan singkat. Dengan ini, maka input rangkaian sensor akan menghasilkan tegangan 0 volt pada penguat amplifier, kemudian akan dibentuk menjadi gelombang yang halus di schmitt trigger dengan keluaran pulsa 0 dan dihubungkan ke komponen transistor yang mendapatkan tegangan sumber (V_{cc}), maka keluaran menghasilkan pulsa 1. Sebaliknya, bila sinar tidak menembus rongga lempengan, maka tegangan pada penguat amplifier akan menghasilkan 5 Volt, dan akan diubah menjadi gelombang kotak yang halus di schmitt trigger dengan keluaran pulsa 1 dan di hubungkan ke komponen transistor, sehingga terjadi hubungan singkat, maka tegangan keluaran menjadi 0 volt atau pulsa 0.

Yang dikenal dengan Schmitt Trigger, rangkaian ini digunakan untuk membersihkan noise atau membersihkan sinyal yang lambat naik atau lambat turun menjadi pulsa digital yang cepat naik dan cepat turun.

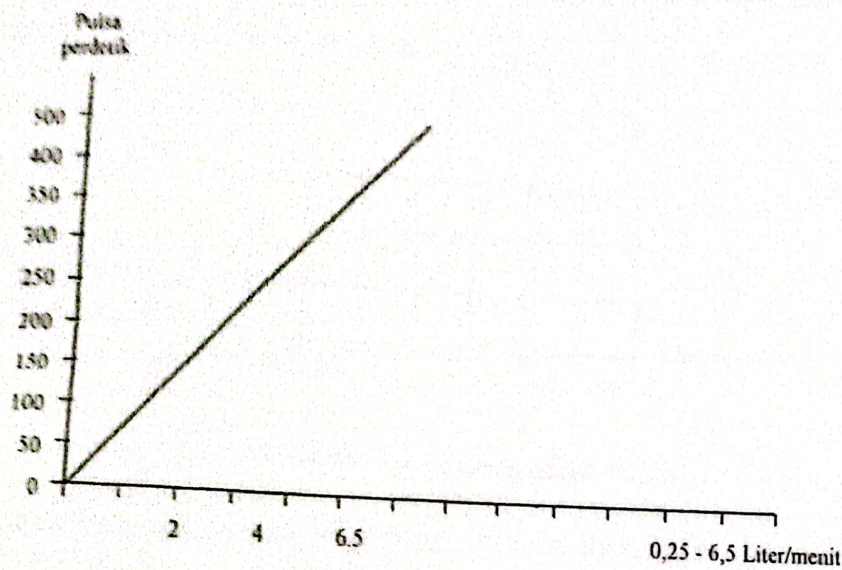
Fungsi schmitt trigger pada sensor yaitu untuk memperbaiki pulsa - pulsa persegi sehingga penunjukkan data penghitungan atau pengukuran bisa stabil. Keluaran schmitt trigger menjadi logika tinggi (1) pada saat isyarat masukan lebih besar, dari suatu nilai tegangan tertentu, dan berlogika rendah (0) bila berada dibawah tingkat *switching* yang diberikan (sistem pintu AND). Hal ini membuat schmitt trigger tahan terhadap desah seperti kita lihat pada gambar 3.3.



Gambar 3.3. Gelombang Pulsa Scmitt Trigger

Menurut data kalibrasi sensor yang digunakan, grafik aliran air pada sensor menunjukkan garis linier yang berbanding lurus antara pulsa perdetik dengan volume permenit, seperti pada grafik 3.1.

Menurut grafik sensor aliran air diatas dapat diuraikan tiap satu pulsa menghasilkan besar volume sebagai berikut :



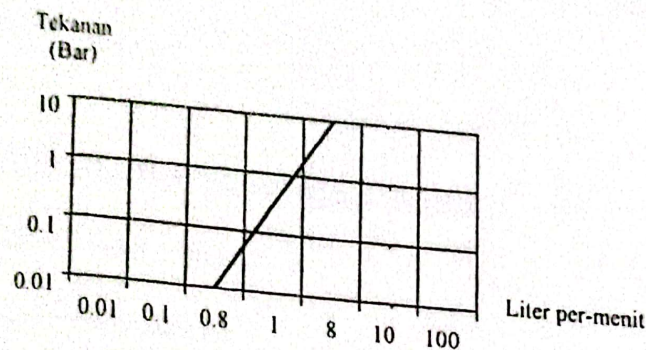
Grafik 3.1. Sensor Aliran Fluida.

- 4600 pulsa = 1 liter
- 1 pulsa = $\frac{1000}{4600}$ milli liter

Keterangan volume per menit di atas menunjukkan bahwa data yang digunakan pada saat kalibrasi di pabrik 0,25 hingga 6,5 liter per menit.

Selain jumlah pulsa yang harus diperhitungkan, juga daya tekanan dari sensor perlu diperhatikan. Menurut dari data spesifikasi sensor yang digunakan. Tekanan aliran fluida pada sensor seperti pada grafik 3.2 dapat diketahui tekanan minimum

dan maksimum aliran fluida yang melalui sensor. Grafik tekanan aliran pada sensor menunjukkan grafik yang mendekati linier yang merupakan perbandingan tekanan dengan rata aliran fluida pada sensor.



Grafik 3.2 Tekanan aliran fluida

Menurut data sensor yang digunakan, tekanan maksimum dan minimum dapat dideteksi oleh sensor, yaitu :

- Tekanan Maksimum = $\longrightarrow$ 10 bar 8 liter per-menit
 - Tekanan Minimum = $\longrightarrow$ 0.01 bar 0.8 liter per-menit
- (1 bar = 10^5 pa ; 1 pa = 1 N/m)

3.3. Mikrokontroler 8031

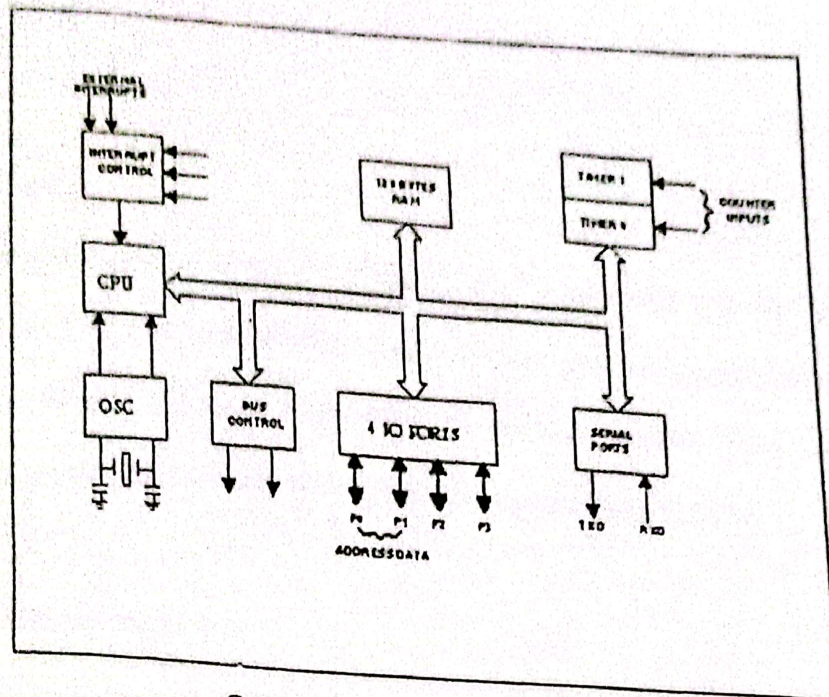
Mikrokontroler 8031 merupakan anggota MCS - 51 keluaran Intel, mikrokontroler merupakan mikroprosesor yang dirancang untuk aplikasi kontrol yang dilengkapi dengan ROM, RAM dan fasilitas I/O pada suatu chip (*one chip*) mikrokontroler. Seri 8051 terdiri dari dua macam yaitu 8031 tanpa ROM dan 8751

dengan EPROM. Alasan menggunakan mikrokontroler 8031 adalah karena kebutuhan kontroler yang akan direalisasikan dapat dipenuhi oleh mikrokontroler ini, yaitu lebar jalur data, kecepatan operasi, operasi boolean dan adanya kemampuan pengalamatan satu bit.

Fasilitas yang dimiliki oleh 8031 adalah sebagai berikut :

- processor boolean (variabel 1 bit)
- 64 kbyte program memory address
- 64 kbyte data memory address
- Internal RAM 128 byte
- 32 port I/O yang diatur sebagai 4 - 8 bit port :P0 - P3
- Control register : TCON, TMOD, SCON, PCON, IP, dan IE
- On chip oscillator

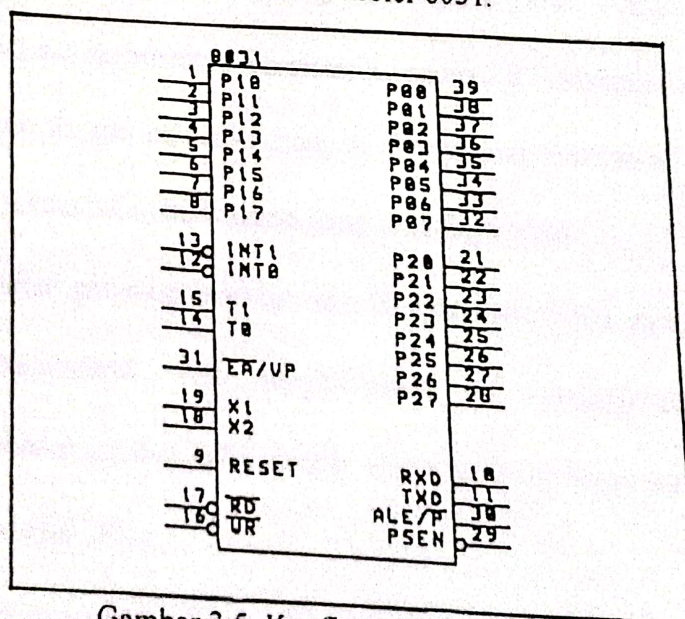
Diagram blok 8031 dapat di lihat pada gambar.



Gambar 3.4 diagram blok 8031.

Mikrokontroler 8031 mempunyai 40 pin, terlihat pada gambar 3.5.

Menunjukkan konfigurasi pin dari mikrokontroler 8031.



Gambar 3.5. Konfigurasi pin dari 8031.

3.3.1. Memory

Sebuah komputer harus memiliki memory untuk kode byte dari program, biasanya di ROM, Dan memory RAM untuk variabel data yang dapat diubah ketika program berjalan. Untuk itu 8031 memiliki internal memory RAM, sedangkan internal memory ROM dimiliki oleh seri 8751 yang memiliki EPROM.

3.3.1.1. Internal RAM

128 byte internal RAM terbagi tiga yaitu :

1. 32 byte dari alamat 00H sampai 1FH yang membentuk 32 register yang bekerja diatur sebagai 4 bank dari masing - masing 8 register. 4 register bank diberi nomor 0 sampai 3 dan membentuk 8 register R0 sampai R7. Masing - masing register dapat dialamatkan dengan namanya maupun oleh alamat ROM-nya. Byte RS0 dan RS1 pada PSW menentukan bank mana yang akan digunakan.
2. Daerah pengalamatan bit dari 16 byte pada RAM dengan alamat 20H sampai 2FH membentuk 128 pengalamatan bit. Pengalamatan bit dapat menggunakan alamat bit dari 00H sampai 7FH atau 8 bit dapat membentuk alamat byte dari 20H sampai 2FH.
3. Daerah penggunaan umum RAM terletak diatas daerah bit dari 30H sampai 7FH pengalamatan sebagai byte.

3.3.1.2. Special Function Register

Pengoperasian 8031 yang tidak menggunakan alamat internal 128 byte RAM dilakukan kelompok internal register khusus yang dinamakan *special function register* (SFR) yang dialamatkan mirip internal RAM, menggunakan alamat dari 80H sampai FFH.

3.3.1.3. Memory Program

Pada memory program hanya dapat dilakukan proses membaca agar 8031 dapat menggunakan external program memory, maka pin Eksternal Acces (EA) harus diaktifkan dengan memberikan tegangan rendah (ground). Gerbang 0 berfungsi sebagai bus alamat yang digandakan dengan bus data. Gerbang 0 akan mengeluarkan low byte dari program counter sebagai informasi alamat. Fungsi gerbang 0 akan menjadi bus data, apabila pin address latch enable (ALE) berubah menjadi tinggi, sementara itu gerbang 2 mengeluarkan informasi highbyte dari program counter sebagai informasi alamat. Setelah urutan tersebut terpenuhi, maka PSEN akan berubah kelevel tegangan rendah.

3.3.2. Arsitektur Internal

3.3.2.1. Akumulator

Akumulator merupakan register 8 bit yang banyak digunakan, terutama dalam instruksi - instruksi pada operasi aritmatika, logik, dan rotate.

3.3.2.2. Base Register

Register base merupakan register yang khusus dipakai untuk operasi perkalian dan pembagian bersama akumulator. Register ini juga dapat digunakan untuk instruksi - instruksi lain seperti register biasa.

3.3.2.3. Register Fungsi Umum

Register fungsi umum ini di kelompokkan dalam 4 bank register yang dialokasikan pada 32 byte pertama dalam internal RAM mikrokontroler 8031. Untuk mengakses register - register ini, banknya terlebih dahulu diaktifkan dengan mengisi RS0 dan RS1 pada Program Status Word (PSW). Tabel 3.1 menunjukkan cara memilih bank register yang mengisi bit - bit yang sesuai untuk RS0 dan RS1.

Tabel 3.1. Pemilihan Bank Register

RS0	RS1	Bank	Lokasi Memori
0	0	0	00H - 07H
0	1	1	08H - 0FH
1	0	2	10H - 17H
1	1	3	18H - 1FH

3.3.2.4. Program Status Word (PWS)

PSW terdiri atas beberapa bit status atau keadaan yang menggambarkan keadaan CPU pada suatu saat. Bit - bit status PSW adalah : bit Carry, Auxiliary Carry. Dua bit pengatur bank register, dan dua status flag.

3.3.2.5. Stack Pointer (SP)

Stack pointer adalah register 8 bit. Isi awal register ini bergantung pada bank register yang digunakan dan akan selalu bertambah apabila ada pemanggilan interupsi atau instruksi PUSH dan CALL. Mekanisme penyimpanan data pada register ini adalah *Last In First Out* (LIFO), yaitu data yang masuk terakhir akan di keluarkan lebih dahulu. Dalam 8031, isi stack pointer tidak boleh lebih dari 7FH, karena alamat setelah 7FH telah di isi oleh register fungsi khusus (SFR) seperti : port, register base, akumulator, PSW dan fungsi kontrol lainnya. Stack pointer akan berisi data 07H apabila prosesor di reset.

3.3.2.6. Port - Port I/O

8031 mempunyai 4 port I/O 8 bit. Port-0 dan port-2 digunakan untuk program eksternal dan pengalamatan data eksternal. Bila tidak digunakan sebagai bus saluran alamat atau bus saluran data, port 0 dan port 2 dapat digunakan sebagai port I/O bidirectional. Port 1 adalah port I/O bidirectional yang digunakan untuk fungsi umum. Pin 3 berisi pin - pin interupsi eksternal, timer eksternal, mengaktifkan baca tulis (RD dan WR) memori data eksternal dan port I/O serial untuk mengirim dan menerima data. Port 3 juga dapat digunakan sebagai port I/O bidirectional.

3.3.2. Interupsi

Terdapat lima interupsi dalam mikrokontroler 8031 yaitu : INT0, INT1, TF0, TF1, TI dan RI. INT0 dan INT1 adalah interupsi eksternal yang dapat diaktifkan dengan mengeset IT0 dan IT1 pada register TCON. Flag yang menghasilkan interupsi adalah IE0 dan IE1 pada TCON. Ketika interupsi eksternal dihasilkan, maka flag tadi akan dihapus melalui perangkat keras (*hard ware*).

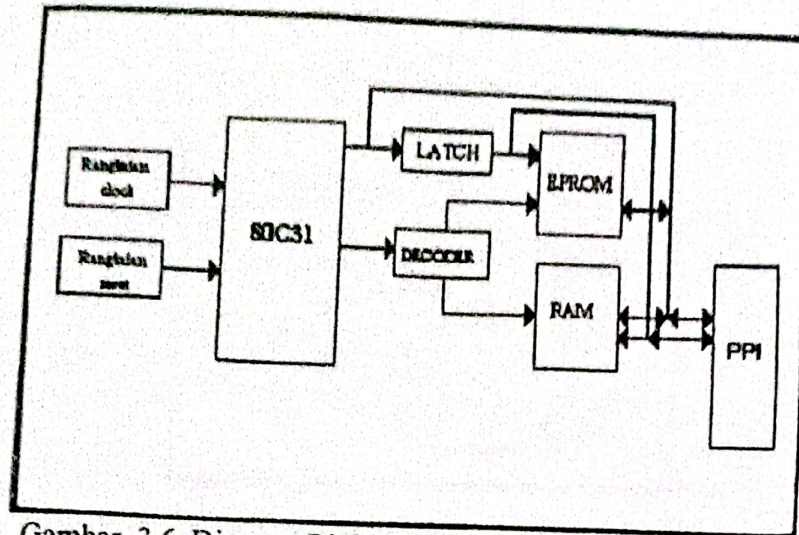
TF0 dan TF1 adalah interupsi yang menghasilkan atau mengaktifkan timer 0 dan timer 1, sedangkan RI dan TI adalah interupsi yang menghasilkan interupsi port serial. Jika terjadi interupsi maka flag - flag tadi tidak dihapus oleh perangkat keras, tetapi harus dihapus melalui perangkat lunak.

3.3.3. Rangkaian Mikrokontroler 8031

Selain chip 8031 , rangkaian mikrokontroler 8031 terdiri dari :

- Rangkaian clock
- Latch
- Memory
- Rangkaian reset
- Dekoder alamat
- PPI 8255

Keterangan dari masing - masing unit akan dijelaskan pada bagian selanjutnya. Diagram blok rangkaian mikrokontroler 8031 dapat dilihat pada gambar 3.6.

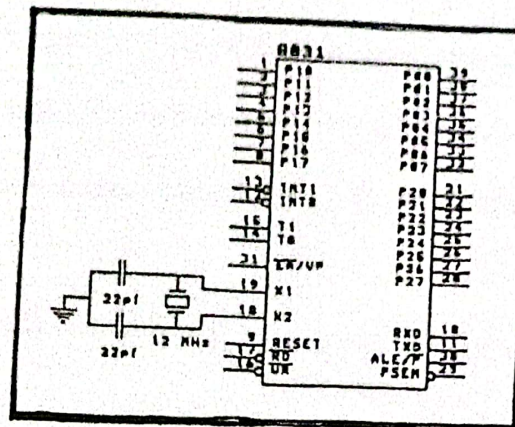


Gambar 3.6. Diagram Blok Rangkaian Mikrokontroler 8031

3.3.3.1. Rangkaian Clock

Mikrokontroler 8031 mempunyai rangkaian osilator didalamnya (*on - chip clock oscillator*), yang dapat digunakan sebagai sumber clock dari CPU. Untuk dapat menggunakan rangkaian osilator dalam chip tersebut, maka pada pin XTAL1 dan XTAL2 (pin 19 dan pin 18) harus dihubungkan dengan sebuah kristal serta dua buah kapasitor yang dihubungkan dengan tanah. Rangkaian clock 8031 dapat dilihat pada gambar 3.7.

Generator clock internal dari 8031 menentukan siklus mesin CPU. Satu siklus mesin (*machine cycle*) terdiri dari enam keadaan dan setiap keadaan sama dengan dua kali perioda osilator. Jadi satu siklus mesin adalah sepanjang 12 perioda osilator.



Gambar 3.7. Rangkaian Clock 8031

3.3.3.2. Rangkaian Power On Reset

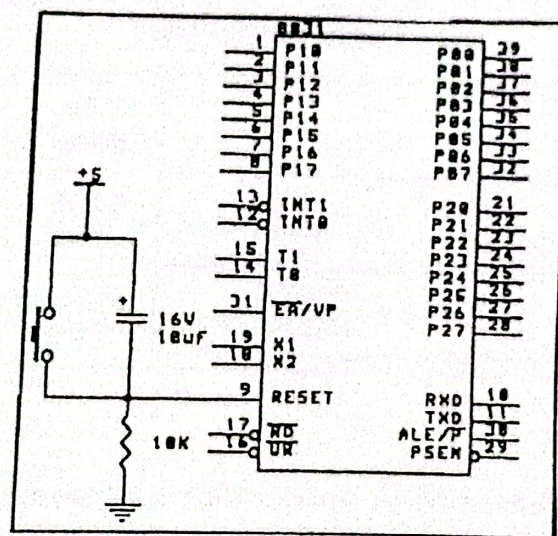
Masukkan reset dilakukan melalui pin reset (pin 9). Reset dilakukan selama dua siklus mesin, dan CPU akan mengaktifkan internal reset.

Rangkaian power on reset diperlukan agar CPU dapat di-reset secara otomatis pada saat pertama kali dihidupkan. Keadaan reset diperoleh apabila pin reset pada mikrokontroler 8031 diberi logika high dalam beberapa milli detik setelah catu daya dihidupkan. Hal ini diberikan untuk memberikan waktu pada rangkaian osilator agar mencapai keadaan stabil.

Waktu power on reset dapat dihitung dengan rumus konstanta yaitu $\tau = RC$. Dengan mengambil nilai $C = 10 \mu F$ dan $R = 10 k\Omega$, maka didapat $\tau = 100$ milli detik.

Cara kerja rangkaian reset adalah sebagai berikut. Bila tegangan catu V_{cc} diberikan, maka arus akan melewati kapasitor sehingga akan menimbulkan beda tegangan pada resistor, tegangan pada pin reset merupakan beda tagangan antara V_{cc} dengan kapasitor, karena sifat dari kapasifor maka tegangan pada pin reset akan menurun sehingga akan mencapai logik 0. Rangkaian reset dapat dilihat pada gambar 3.8.

Sistem ini dirancang juga untuk reset secara manual dengan menggunakan saklar push on reset, bila saklar ditekan maka pin reset akan mendapat tegangan sebesar V_{cc} yang menyebabkan mikrokontroler berada dalam keadaan reset.



Gambar 3.8. Rangkaian Reset Mikrokontroler 8031

3.1.1.1. Rangkaian Dekoder

Pada rangkaian mikrokontroler 8031, setiap komponen harus mempunyai alamat yang unik, salah satu cara untuk mendapatkan alamat yang unik adalah dengan menggunakan dekoder. Mikrokontroler 8031 mempunyai lokasi memori sebesar 64 Kb yang dapat dibagi menjadi 8 bagian yang besarnya masing - masing Kb. Pembagian memori ini menggunakan decoder 74LS138.

Pembagian lokasi memori dapat dilihat pada tabel 3.2. Pada gambar tersebut dapat dilihat alamat 0000H - 1FFFFH digunakan untuk lokasi EPROM 2764 yang besarnya 8Kb, alamat 2000H - 3FFFFH untuk lokasi RAM, alamat 4000H - 4FFFFH digunakan untuk lokasi PPI 8255.

Tabel 3.2. Peta lokasi memori

5FFFH 4000H	PPI 255
3FFFH 2000H	RAM 6264
1FFFH 0000H	EPROM 2764

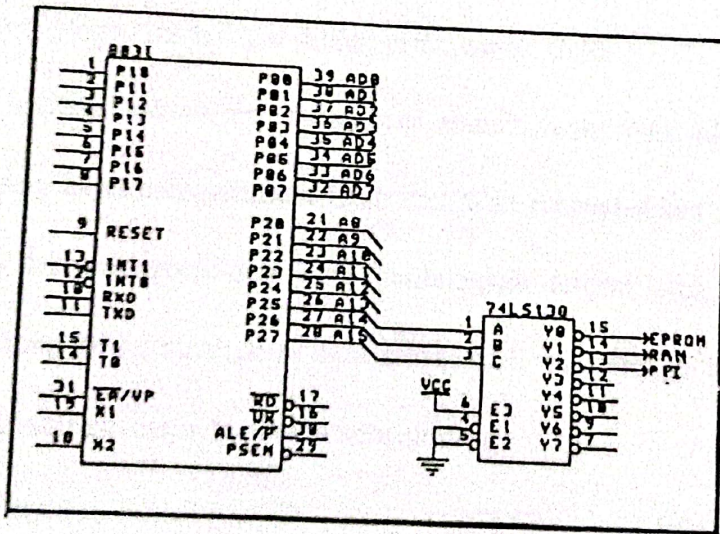
Tabel 3.3. merupakan tabel kebenaran decoder. Masukkan untuk decoder alamat adalah pin A13, A14 A15 pada mikrokontroler 8031. Keluaran dari decoder alamat dihubungkan dengan chip select masing - masing divais.

Tabel 3.4 Tabel Kebenaran *Decoder*

[illegible]

Keterangan x = menyatakan bisa 0 bisa 1

Pada tabel kebenaran dapat dilihat pada keluaran dari decoder alamat Y0 dihubungkan dengan EPROM, dan Y2 dihubungkan dengan PPI - 8255.



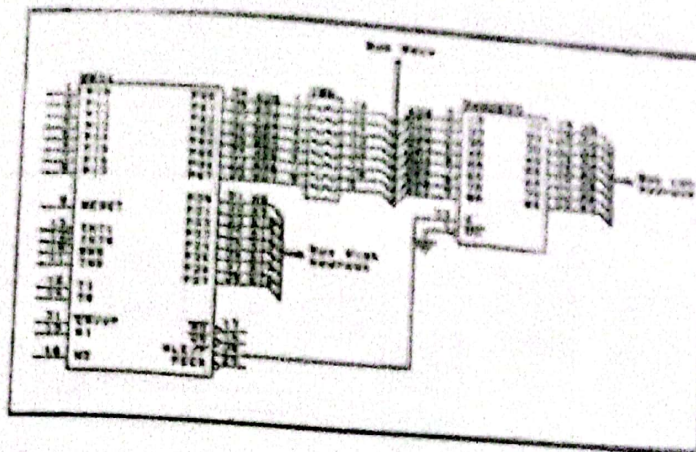
Gambar 3.9. Rangkaian Decoder

Agar decoder 74LS138 selalu berada dalam keadaan aktif, maka masukan enable E3 dihubungkan dengan Vcc sedangkan dua masukan enable lainnya yaitu E1 dan E2 dihubungkan dengan ground. Rangkaian decoder dapat dilihat pada gambar 3.9.

3.3.3.4. Rangkaian Latch

Pada mikrokontroler 8031 low byte alamat (AD0 - AD7) dan satu berada pada satu gerbang. Untuk dapat memisahkan data dan alamat maka digunakan latch. Apabila sinyal ALE (*Address Latch Enable*) dari mikrokontroler berada pada posisi 1, maka menandakan alamat ruang akses. ALE (pin 30) dari mikrokontroler dihubungkan dengan C (pin 11) dari latch (74HCT573). Hubungan antara mikrokontroler dengan latch dapat dilihat pada gambar 3.10.

Pada saat mikrokontroler mengakses alamat, maka ALE akan memberikan sinyal high yang akan mengaktifkan latch dan akan menyebabkan low byte alamat (AD0 - AD7) dari mikrokontroler yang dihubungkan dengan latch akan tersimpan. Pada saat mikrokontroler mengakses data, sinyal ALE = 0 sehingga latch disable dan latch tetap menyimpan low byte alamat sebelumnya.



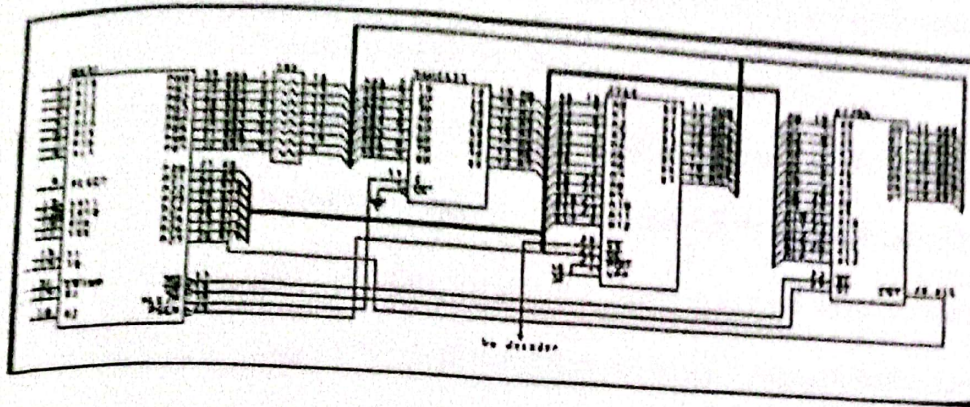
Gambar 3.10. Hubungan latch dengan 8031

3.3.3.5. Rangkaian Memori

Mikrokontroler 8031 mempunyai 16 bit alamat untuk pengalamatan luar. Informasi low byte alamat (A0- A7) disediakan oleh gerbang 0 yang juga mengandung informasi data, sedangkan informasi high byte alamat (A8- A15) disediakan oleh gerbang 2.

Untuk program memori (27C64), low byte address dihubungkan dengan keluaran dari latch (74HC573), high byte address dihubungkan langsung ke gerbang 2. Bus data pada memori program dihubungkan langsung dengan gerbang 0. Pin CE pada program memori dihubungkan dengan keluaran dari decoker 74LS138 (Y0).

sedangkan OE dihubungkan dengan PSEN dari mikrokontroler 8031. Hubungan antara mikrokontroler 8031 dengan memori dapat dilihat pada gambar 3.11



Gambar 3.11. Hubungan 8031 dengan memori

3.3.3.6. Programmable Peripheral Interface (PPI) 8255

PPI 8255 memiliki tiga buah gerbang register dan sebuah register kontrol. Setiap register tersebut terdiri dari 8 bit yang dapat diatur melalui perangkat lunak untuk melaksanakan fungsi input atau out put.

Beberapa pin yang perlu diketahui adalah :

- CS : Bila masukan pin CS dalam keadaan low , maka 8255 dapat berkomunikasi dengan 8031.
- RD : Logika "0" pada pin ini bersama-sama dengan logika "0" pada CS kemungkinan 8255 untuk mengirim data ke

mikrokontroler, jadi mikrokontroler melakukan pembacaan data dari 8255

- WR : Logika "0" pada pin ini bersama-sama dengan logika "0" pada CS kemungkinan mikrokontroler untuk mengirimkan data atau control word ke PPI 8255
- A0 dan A1 : Ada empat buah alamat yang dapat dibentuk dari kombinasi masukan A0 dan A1. Jadi A0 dan A1 menentukan pemilihan salah satu dari tiga gerbang dan satu control word register
- Reset : Logika "1" pada masukan pin RESET mengakibatkan control word register terhapus dan ketiga gerbang diatas sebagai mode masukan.

Fungsi dari ketiga gerbang ditentukan oleh mode operasi PPI, yang dikontrol melalui program pada register kontrol. Ada tiga mode pemrograman PPI 8255, yaitu :

1. Masukan dan Keluaran Dasar (Mode-0)

Dalam mode ini 24 jalur dibagi menjadi dua kelompok (*group*) 8 jalur (gerbang A) dan (gerbang B) dan dua group 4 jalur (gerbang C *upper* dan gerbang C *lower*). Masing - masing gerbang ini dapat diprogram sebagai masukan dan keluaran.

2. Masukan dan Keluaran dengan Strobe (Mode-1)

Dalam mode ini PPI menggunakan dua gerbang 8 bit, gerbang A dan gerbang B sebagai masukan dan keluaran satu arah. Masing - masing gerbang transfer data dengan *strobe* atau *handshaking* sinyal pada gerbang C.

3. Masukan atau Keluaran Dua arah dengan strobe (Mode 2)

Dalam mode ini gerbang A sebagai gerbang dua arah. Lima bit dari gerbang C dipakai sebagai *handshaking* sinyal

Tabel 3.3. Operasi dasar PPI 8255

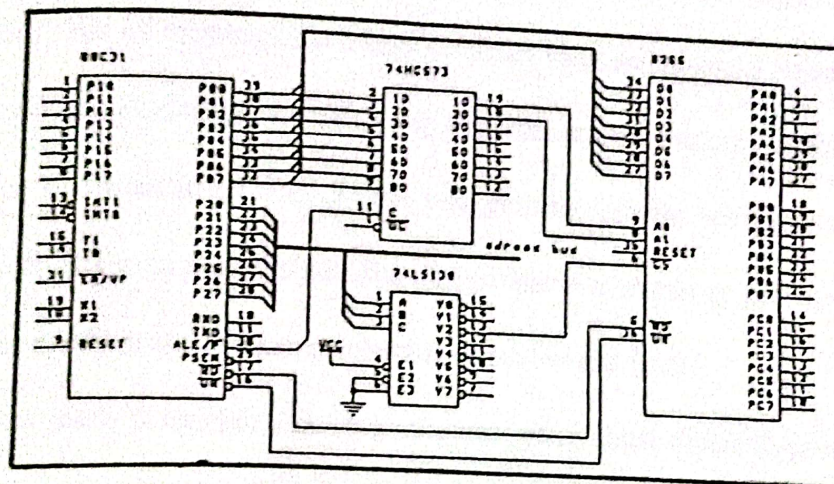
A1	A0	RD	WR	CS	Perpindahan	Operasi
0	0	0	1	0	Port A → data bus	Input (READ)
0	1	0	1	0	Port B → data bus	
1	0	0	1	0	Port C → data bus	
0	0	1	0	0	Data bus → Port A	Output (WRITE)
0	1	1	0	0	Data bus → Port B	
1	0	1	0	0	Data bus → Port C	
1	1	1	0	0	Data bus → Control	
X	X	X	X	1		Data bus 3-state
X	X	1	1	0		Data bus 3-state
1	1	0	1	0		Kondisi ilegal

Keterangan : X menyatakan keadaan bisa 0 atau 1

Operasi dasar PPI 8255 sehubungan dengan pin - pin inputnya dapat dilihat pada tabel diatas. Untuk menghubungkan PPI 8255 ke mikrokontroler 8031 dapat

dilakukan dengan menghubungkan data bus 8031 ke jalur data dari PPI WR dan RD dari 8031 kemasukan WR dan RD dari PPI, A0 dan A1 dari 8031 ke pin - pin yang bersesuaian. Sedangkan untuk pin CS dihubungkan dengan keluaran dari decoder (Y2). Hubungan antara PPI 8255 dengan 8031 dapat dilihat pada gambar 3.12.

Untuk pemetaan I/O ini perlu diperhatikan bahwa PPI 8255 menempati empat lokasi I/O sesuai dengan jumlah registernya, yaitu register PA, PB, PC dan register kontrol. Ke-empat register ini dapat diakses dengan pengaturan pin A0, A1.



Gambar 3.12. Hubungan PPI 8255 dengan 8031

Mode operasi dari gerbang A sampai gerbang C dan operasi set- reset pada gerbang C diatur dengan mode set control atau gerbang C bit set reset control word yang dikirim ke register kontrol.

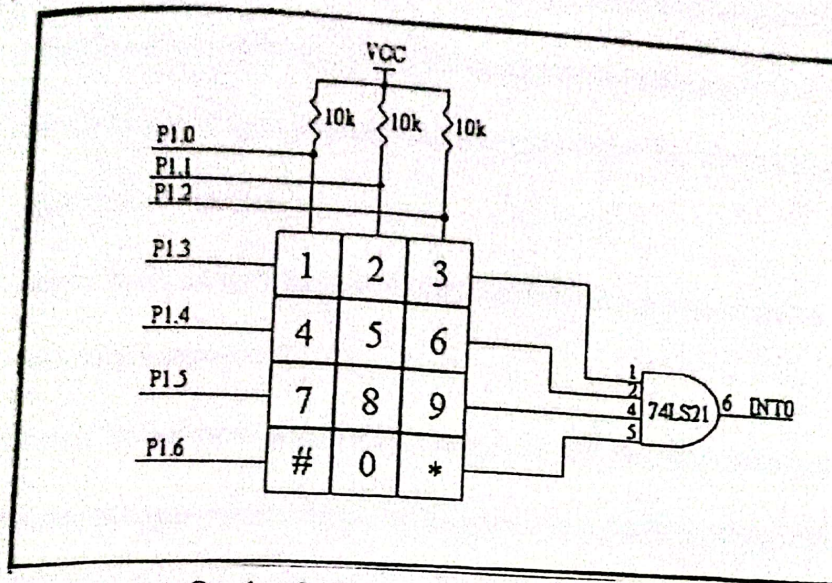
Dalam perencanaan perangkat keras, PPI 8255 yang dipergunakan dioperasikan dalam mode 0 untuk semua gerbang, sehingga pembahasan hanya dibatasi untuk mode 0 saja, untuk mode - mode lainnya dapat dilihat pada lampiran data 8255.

Dalam mode 0, ketiga gerbang 8255 hanya berfungsi sebagai input dan output gerbang biasa. Semua gerbang yang bekerja sebagai keluaran gerbang akan bersifat latch, artinya data yang sudah diletakkan ke output gerbang masih tetap ada selama tidak diubah oleh data lain yang ditulis ke gerbang yang sama atau akibat adanya sinyal *RESET*. Hal ini memungkinkan bagi peralatan yang berhubungan ke keluaran gerbang untuk melakukan pembacaan yang tertunda, artinya tidak pada saat data tersebut ada di keluaran gerbang. Tetapi bila gerbang bekerja sebagai masukan gerbang maka data yang akan dipindahkan dari peralatan masukan ke gerbang 8255 harus sudah tersedia saat dilakukan pembacaan (tidak bersifat *latch*).

Dalam mode 0 tersebut ada 16 konfigurasi yang dapat dibentuk yaitu dari kombinasi bit-bit D0, D1, D3, dan D4 pada *control word register* seperti gambar 3.12. Untuk operasi mode 0 pada semua gerbang, maka bit D7=1, sedangkan bit D6, D5, dan D2 sama dengan 0.

3.3.3.7. Rangkaian Papan Kunci (Keypad)

Apabila sebuah papan kunci dihubungkan, pada umumnya dihubungkan secara matriks dalam bentuk baris dan kolom. Papan kunci yang akan dihubungkan dengan rangkaian mikrokontroler adalah jenis papan kunci kontak mekanis dengan konfigurasi matriks 3 x 4 (3 kolom, 4 baris). Papan kunci ini sebagai piranti masukan data kelembaban yang akan diinginkan, rangkaian papan kunci ini dapat dilihat pada gambar 3.13.



Gambar 3.13. Rangkaian papan kunci

Bila rangkaian papan kunci dihubungkan ke suatu mikroprosesor, maka ada tiga hal yang harus diperhatikan antara lain :

- Mendeteksi adanya kunci yang ditekan.
- Melakukan *debounce* pada papan kunci yang ditekan.

- Melakukan proses pengkodean terhadap kunci yang ditekan.

Ketiga hal di atas dapat dilakukan baik oleh perangkat keras, perangkat lunak, atau kombinasi keduanya, tergantung perancangan sistem yang dibuat. Pada sistem ini, penanganan ketiga hal di atas dipilih dengan menggunakan perangkat lunak. Pemilihan ini berdasarkan pertimbangan bahwa penanganan dengan perangkat lunak termasuk sederhana dan bila terjadi kerusakan pada rangkaian mudah dilacak.

- Papan kunci dengan konfigurasi 3 x 4 ini dihubungkan ke mikrokontroler 80C31 melalui gerbang (port) 1 dengan susunan sebagai berikut :
- Setiap kolom dengan jumlah 3 dihubungkan secara berurutan ke P1.0-P1.2 dari mikrokontroler.
- Setiap baris dengan jumlah 4 dihubungkan secara berurutan ke P1.3-P1.6 dari mikrokontroler.

Susunan kolom pada papan kunci berperan sebagai *scan lines*, dan susunan baris berperan sebagai *return lines*. Perangkat lunak pada mikrokontroler akan memberikan logika high pada *return lines*, dan secara bergantian menulis logika low kepada semua *scan lines*. Untuk menstabilkan logika high pada *scan lines* dipasang resistor *pull up* ke Vcc. Dan keempat *return lines* dihubungkan dengan sebuah gerbang AND empat (4) masukan (74LS21), dan keluaran gerbang AND dihubungkan pada INT0 (Interupsi eksternal 0) pada mikrokontroler. INT0 8031 aktif low, artinya interupsi akan terjadi, bila kaki INT0 berlogika low. Bila tidak ada kunci

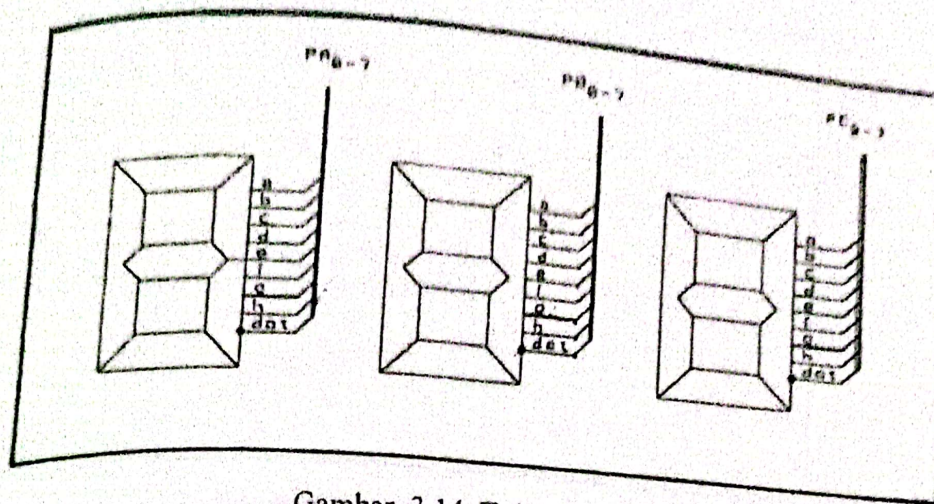
yang ditekan, semua return lines berlogika high, maka keluaran gerbang AND akan selalu high, yang berarti tidak terjadi interupsi.

Bila salah satu papan kunci ditekan, maka salah satu scan lines akan terhubung ke salah satu return lines, sehingga mengakibatkan return lines yang bersangkutan akan bernilai logika low. Hal ini menyebabkan keluaran 74LS21 menjadi low, sehingga akan terjadi pulsa interupsi pada mikrokontroler. Dengan demikian adanya penekanan papan kunci telah terdeteksi oleh mikrokontroler. Selanjutnya rutin interupsi pada mikrokontroler bertugas mengenali dan pengkodean kunci yang ditekan tersebut. Namun sebelum melakukan pengkodean ini, perangkat lunak harus melakukan proses *debouncing* terlebih dahulu.

3.3.4. Rangkaian Peraga Tujuh Ruas (*Seven Segment*)

Untuk menampilkan nilai volume air yang terukur, digunakan peraga tujuh ruas. Jenis peraga yang digunakan adalah *common anoda*. Karena keluaran dari mikrokontroler berupa data BCD (*Binary Code Decimal*) dan tidak digunakannya dekoder BCD-7 ruas maka pada program diberikan tabel konversi ke penampil tujuh ruas. Dalam sistem ini digunakan tiga peraga tujuh ruas yang dihubungkan ke PPI 8255 (gerbang _A, gerbang _B, dan gerbang _C). Untuk PPI 8255 data terakhir yang akan dikirim tidak akan berubah, digunakan sebagai gerbang keluaran. Gerbang _A

dihubungkan dengan tujuh ruas pertama, gerbang B dihubungkan dengan tujuh ruas kedua dan gerbang C dihubungkan dengan tujuh segmen yang ketiga



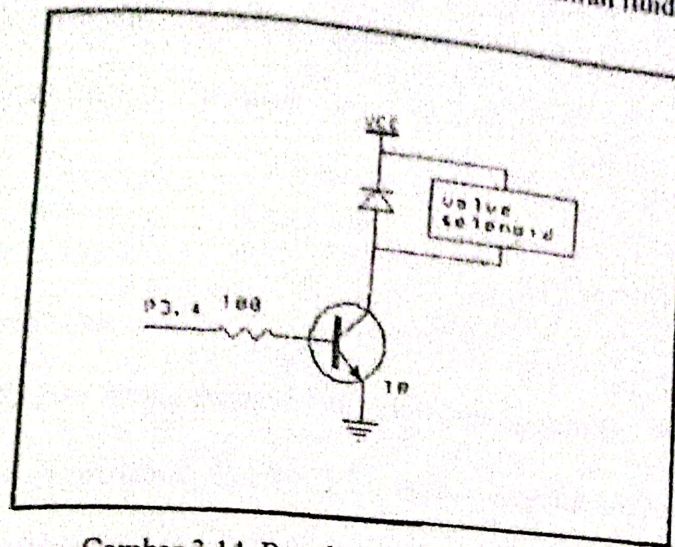
Gambar. 3.14. Tujuh segmen

3.3.5. Rangkaian Valve Selenoid

Rangkaian ini digunakan untuk membuka valve selenoid yang digunakan sebagai katub aliran fluida cair. Rangkaian valve selenoid ini dapat kita lihat pada gambar 3.15.

Apabila Port 3.4 dari mikrokontroler diberi sinyal high, maka basis dari transistor akan mendapatkan bias tegangan 3 volt, sedangkan tahanan yang terdapat pada selenoid adalah sebesar 50 ohm pada saat ini katub akan terbuka. Dengan keadaan ini akan menyebabkan adanya arus yang mengalir pada valve selenoid,

sehingga valve akan terbuka dan air dapat mengalir. Kemudian sebaliknya apabila port 3.4 dari mikrokontroler 8031 diberi sinyal low, maka aliran fluida akan terhenti.



Gambar 3.14. Rangkaian Valve Selenoid

3.4. Perangkat Lunak

Perangkat lunak (*software*) yang digunakan pada pengontrol volume adalah bahasa pemrograman *assembly* ASM51. Fungsi utama dari perangkat lunak ini adalah

- Menerima masukan data dari papan kunci berupa data penghitungan volume yang diinginkan pada setiap langkah.
- Menghitung pulsa yang dihasilkan sensor.

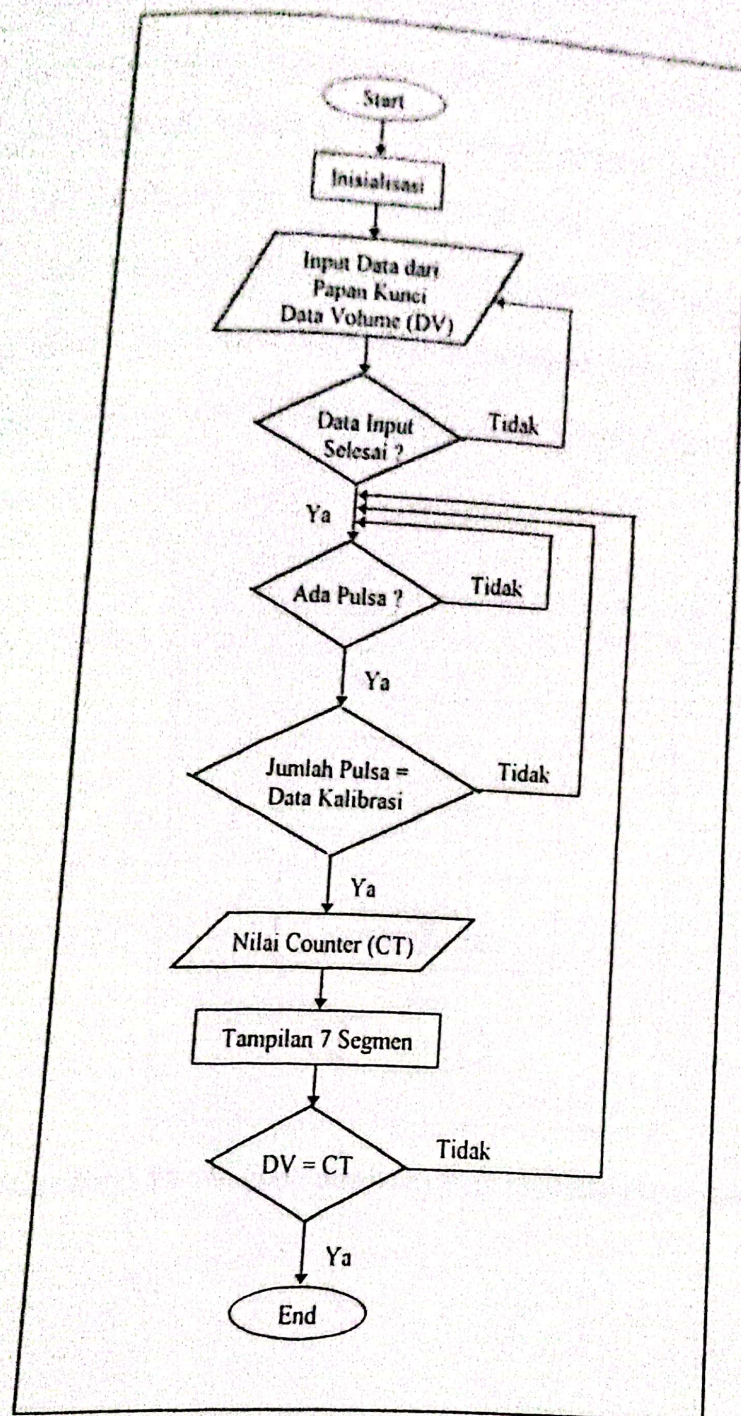
membandingkan data yang diinputkan dengan jumlah air yang telah diinputkan
dan menampilkannya pada screen program.

Membuka dan menutup katup.

3.4.1. Program Utama

Pada program utama pertama kali dilakukan inisialisasi, dengan menerima masukan dari papan kunci. Gambar 3.15. adalah diagram alir utama pengontrol penghitung volume. Inisialisasi yang dilakukan antara lain :

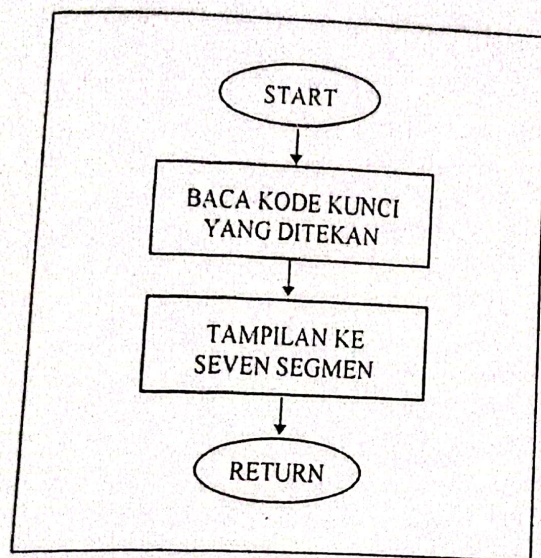
- Inisialisasi ekuasi variabel - variabel dan lokasi alamat yang digunakan
- Inisialisasi eksternal interrupt 0 (INT0) sebagai masukan papan kunci.
- Inisialisasi eksternal interrupt 1 (INT1) sebagai masukan pulsa dari sensor.
- Inisialisasi PPI 8255 dengan PA , PB dan PC sebagai keluaran.
- Inisialisasi papan kunci dengan memberi harga logika "1" pada setiap bit di port_1 mikrokontroler 8031.



Gambar 3.15. Diagram alir program utama penghitung volume

3.4.2. Subrutin pelayanan interupsi papan kunci

Subrutin pelayanan interupsi papan kunci ini pertama-tama membaca kode papan kunci yang masuk, kemudian diberikan kepada penampil tujuh segmen, setelah itu kembali ke program utama. Kode papan kunci yang masuk dibaca dulu, apakah enter (#), escape (*) atau tombol numerik. Bila enter yang ditekan maka data numerik yang sudah terlebih masuk akan ditampilkan ke penampil tujuh segmen, bila escape yang ditekan maka data numerik harus diinput ulang. Diagram alir ini diperlihatkan pada gambar 3.16.



Gambar 3.16. Diagram alir pelayanan interupsi papan kunci

BAB IV

PENGUJIAN ALAT

4.1. Langkah Penggunaan Alat

4.1.1. Pemilihan Mode

Pada setiap permulaan pengukuran volume air akan diminta untuk memasukkan mode yang akan digunakan. Mode yang digunakan ada beberapa macam yaitu :

Mode 1, adalah untuk setiap 1 milliliter terhadap kenaikan setiap angka pada papan tampilan.

Mode 2, digunakan untuk setiap 10 milliliter.

Mode 3, digunakan untuk setiap 100 milliliter

Mode 4, digunakan untuk setiap 1000 milliliter atau 1 liter

4.1.3. Proses Memasukan Data

Pemilihan mode yang diinginkan telah dipilih maka, data volume yang diinginkan tersebut dapat dimasukan. Data yang diinginkan tersebut maksimum 999. Dalam pemasukan data yang salah dapat diganti dengan cara menekan tombol (*), setelah itu data dimasukan kembali.

4.2. Pelaksanaan Pengukuran

Menghubungkan kabel catu daya 5 volt untuk pengolah data sensor dan juga menghubungkan kabel catu daya ke rangkaian valve selenoid sebesar 12 volt. Kemudian menghubungkan keluaran sinyal dari sensor ke INT Ø 8031.

Mereset pengolah data sebelum pengukuran dilakukan agar tampilan menunjukkan angka nol, kemudian memasukkan angka yang diinginkan dengan menekan papan kunci, maka pengukuran dapat dilakukan sesuai dengan angka yang dikehendaki.

Pengukuran dapat dilakukan dengan beberapa satuan volume yang dikehendaki sesuai dengan kalibrasi pengolah data, misalnya satuan milliliter dan

liter, dan satuan lainnya. Hasil pengukuran ini dapat dilihat pada papan tampilan yang akan dibandingkan dengan hasil pengukuran yang ada pada gelas ukur.

4.3. Hasil Pengukuran

Hasil besaran yang didapat dari pengukuran akan disetarakan dengan volume yang ditampung pada gelas ukur. Satuan yang digunakan yaitu, satuan pada gelas ukur, liter dan milliliter yang dilakukan pada pengkalibrasian alat pengolah data, dimana pengolah data melakukan beberapa kali perhitungan pada saat pengukuran, hal ini setelah alat pengolah data dikalibrasi.

Berdasarkan pada perhitungan dibawah ini, maka pada tabel 4.1. dapat diketahui persen kesalahan sebagai berikut :

$$\begin{aligned}\% \text{ Kesalahan} &= \frac{300 - 200}{300} \times 100 \% \\ &= \frac{100}{300} \times 100 \% \\ &= 33.3 \%\end{aligned}$$

Tabel 4.1 Hasil Pengukuran

No.	Mode	Tampilan	Volume terukur milliliter (ml)	Kesalahan (%)
1	1	60	60	0
	1	300	200	33.3
	1	500	300	40
2	2	010	100	0
	2	100	1000	0
	2	900	8990	0.1
3	3	005	500	0
	3	020	2000	0
	3	100	9990	0.1
4	4	001	1000	0
	4	005	5000	0
	4	015	14990	0.1

BAB V

KESIMPULAN

- Ketepatan penghitungan volume air dengan menggunakan sensor dan alat pengolah data, tergantung kepada data yang digunakan untuk mengkonversi banyaknya pulsa yang dihasilkan oleh sensor.
- Pengukuran aliran fluida dengan menggunakan sensor ini, dapat mengukur aliran fluida dengan pengkalibrasian satuan volume yang dikehendaki.
- Dari hasil rancangan perangkat lunak, alat pengukur ini dapat digunakan untuk kelipatan 1 milliliter, 10 milliliter, 100 milliliter dan 1000 milliliter (1 liter).

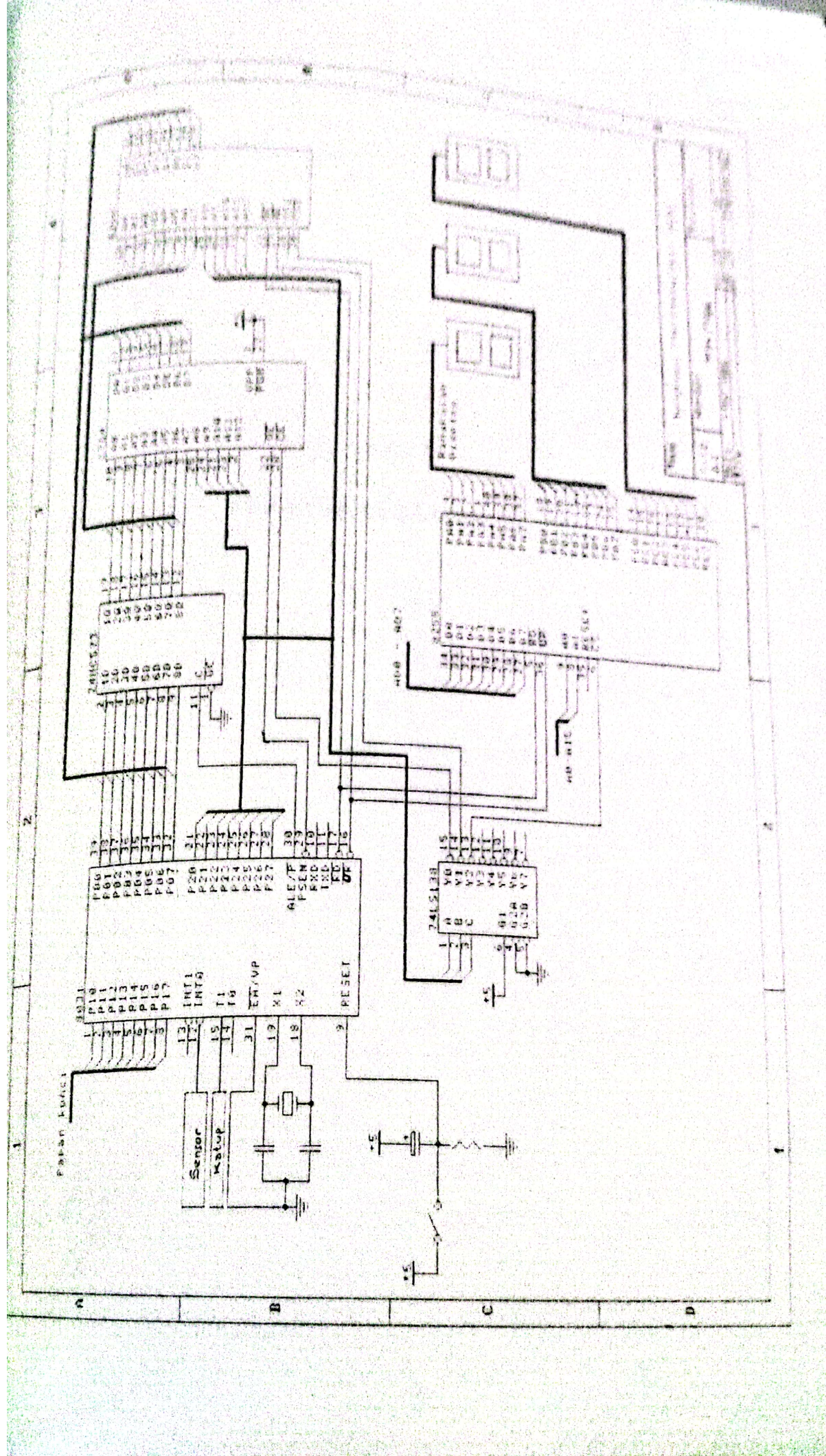
Alat ini mempunyai kelemahan untuk pengukuran dalam mode 1 yaitu dengan ketelitian sekitar 60 %. Sedangkan dalam mode 2, 3 dan mode 4 mempunyai ketelitian sebesar 99 %.

DAFTAR PUSTAKA

1. Spector, *Mekanika Fluida*, Gramedia, Juni 1982
2. Wajidi, S., *Vademecum Elektronika I*, Gramedia, 9187
3. Asala, Kenneth I., *The 8051 Microcontroller Architecture, Programming, and Applications*, 8th edition, West Publishing Company, St. Paul, MN, 1991
4. Hall, Douglas V., *Microprocessor and Interfacing (Programming and Hardware)*, Mc. Grawhill Inc, Singapore, 1986
5. Intel, *Memory Component Hand book*, Intel Corp, St. Clara, CA, 1987
6. Tokheim, Roger L., *Elektronika Digital*, Edisi kedua, Penerbit Erlangga, 1993

LAMPIRAN

LAMPIRAN A
SARANA RANGKAIAN



LAMPIRAN B

PERANGKAT LUNAK


```

simpan_satuan:    call    tampil1
                  mov     r0,#071h
                  mov     a,r0
                  mov     a,#0f0h
                  and     r7,a
                  orl     r0,a
                  mov     r0,b.3
                  ret

simpan:           clr     b.5
                  ret

simpan:           setb    ie.2
                  clr     r2,ss.3
                  djnz    r2,75h
                  mov     r3,ss.3
                  djnz    r3,74h
                  mov     last,ie.2
                  jmp     setb
                  reti

set:              last:  clr     ie.2
                  mov     r0,#0h
                  clr     c
                  mov     a,77h
                  add     a,#1h
                  da
                  mov     77h,a
                  jnc     last1
                  clr     c
                  mov     77,#0h
                  mov     a,76h
                  add     a,#1h
                  da
                  mov     76h,a
                  jnc     last1
                  mov     76h,#0h

last1:            clr     c
                  jb      b.7,last2
                  mov     a,70h
                  subb    a,76h
                  cjne    a,#0h,last4
                  setb    b.7

last2:            clr     c
                  jb      b.6,qq
                  mov     a,71h
                  subb    a,77h
                  cjne    a,#0h,last4
                  setb    b.6
                  mov     p3.4
                  clr     p3.1
                  setb    b.5
                  clr     ie.7
                  clr     ie.2
                  mov     60h,#7fh
                  mov     76h,70h
                  mov     77h,71h
                  call    udah
                  reti

last4:            clr     c
                  setb    ie.2
                  call    udah
                  reti

```

udah: push acc

```

mov     a,77h      ;ambil data satuan BCD
and     a,#0fh     ;simpan ke internal RAM
mov     r7,a       ;ambil data 4 bit LSB
call    checkn     ;simpan data ke register r7
                        ;panggil prosedur cek data pada r7

```

menampilkan data yang ditukan

data pertama dan kedua digabung
data disimpan pada memori
set bit tanda data sudah ada

```

segmen:          call    tampil1
                  mov     a,77h
                  and     a,#0f0h
                  swap    a
                  mov     r7,a
                  call    checkn
                  call    tampil1

```

```

segmen:          mov     a,76h
                  and     a,#0fh
                  mov     r7,a
                  call    checkn
                  call    tampil2
                  ret

```

CHECKN:

```

cjne    r7,#00h,No1 ; Prosedur memeriksa data
mov     r0,#82H      ; periksa apakah data = 0
ret              ; data 0 pada 7 segmen

No1:       cjne    r7,#01h,No2 ; periksa apakah data = 1
mov     r0,#9fH      ; data 1 pada 7 segmen
ret

No2:       cjne    r7,#02h,No3 ; periksa apakah data = 2
mov     r0,#0a8H     ; data 2 pada 7 segmen
ret

No3:       cjne    r7,#03h,No4 ; periksa apakah data = 3
mov     r0,#89H      ; data 3 pada 7 segmen
ret

No4:       cjne    r7,#04h,No5 ; periksa apakah data = 4
mov     r0,#95H      ; data 4 pada 7 segmen
ret

No5:       cjne    r7,#05h,No6 ; periksa apakah data = 5
mov     r0,#0e1H     ; data 5 pada 7 segmen
ret

No6:       cjne    r7,#06h,No7 ; periksa apakah data = 6
mov     r0,#0c0H     ; data 6 pada 7 segmen
ret

No7:       cjne    r7,#07h,No8 ; periksa apakah data = 7
mov     r0,#8fH      ; data 7 pada 7 segmen
ret

No8:       cjne    r7,#08h,No9 ; periksa apakah data = 8
mov     r0,#080H     ; data 8 pada 7 segmen
ret

No9:       mov     r0,#81H      ; periksa apakah data = 9
ret              ; data 9 pada 7 segmen

```

tampil1:

```

mov     dptr,#4002h ; Prosedur penampilan data
mov     a,r0        ; pilih port a pada PPI
movx    dptr,a      ; untuk menampilkan
ret              ; data

```

tampil2:

```

mov     dptr,#4000h ; Prosedur penampilan data
mov     a,r0        ; pilih port a pada PPI
movx    dptr,a      ; untuk menampilkan
ret              ; data

```

tampil3:

```

mov     dptr,#4001h ; Prosedur penampilan data
mov     a,r0        ; pilih port a pada PPI
movx    dptr,a      ; untuk menampilkan
ret              ; data

```

rutin loop scanning keypad

```

setb    ie.0
clr     b.0
clr     b.2

loopscan:
clr     p1.0
call    delay1
setb    p1.0

```



```

        b.0
        p.1
        delay 1
        p.1
        b.0
        p.2
        delay 1
        p.2
        delay
        p.2
        delay
        p.2
        b.0
        h.2,loopscan
        ie.0
        ret

```

```

keyin:  mov a,#0h
        mov r7,#0h
        clr ie.0
        call delay2
        call b.0,keluarkey1
        jb delay2
        call delay2
        call h.0
        setb delay2
        call delay2
        call ie.0
        clr r7.p1
        mov ie.0
        clr keytable
        call

```

```

keluarkey1: setb b.2
            clr b.1
            setb ie.0
            setb b.2
            reti

```

```

keytable:  clr b.1

```

```

satu:     cjne r7,#0f6h,dua
        mov a,#01h
        mov r7,a;#09fh
        mov r6,a
        call simpan
        call delay
        jmp keluar2

```

```

dua:      cjne r7,#0f3h,tiga
        mov a,#02h
        mov r6,a
        mov r7,a
        call simpan
        call delay
        jmp keluar2

```

```

a:        cjne r7,#0f3h,empat
        mov a,#03h
        mov r6,a
        mov r7,a
        call simpan
        call delay
        jmp keluar2

```

```

at:       cjne r7,#0ech,lima
        mov a,#04h
        mov r6,a
        mov r7,a
        call simpan
        call delay
        jmp keluar2

```

```

        cjne r7,#0edh,enam
        mov a,#05h
        mov r6,a
        mov r7,a

```

```

        call simpan
        call delay
        jmp keluar2

```

```

enam:     cjne r7,#0ebh,keluar2
        mov a,#06h
        mov r6,a
        mov r7,a
        call simpan
        call delay
        jmp keluar2

```

```

tujuh:    cjne r7,#0deh,delapan
        mov a,#07h
        mov r6,a
        mov r7,a
        call simpan
        call delay
        jmp keluar2

```

```

delapan:  cjne r7,#0ddh,sembilan
        mov a,#08h
        mov r6,a
        mov r7,a
        call simpan
        call delay
        jmp keluar2

```

```

sembilan: cjne r7,#0dbh,enol
        mov a,#09h
        mov r6,a
        mov r7,a
        call simpan
        call delay
        jmp keluar2

```

```

enol:     cjne r7,#0bdh,kross
        mov a,#0h
        mov r6,a
        mov r7,a
        call simpan
        call delay
        jmp keluar2

```

```

kross:    cjne r7,#0beh,kress
        setb b.4
        call delay
        jmp keluar2

```

```

kress:    cjne r7,#0bbh,keluar2
        call delay
        setb b.3

```

```

keluar2:  setb b.1
        setb b.2
        call delay
        ret

```

```

delay:    mov r3,#0fh
delaya:   mov r2,#0fh
delayb:   djnz r2,delayb
        djnz r3,delaya
        ret

```

```

delay1:   mov r3,#0fh
delay1a:  mov r2,#04h
delay1b:  djnz r2,delay1b
        djnz r3,delay1a
        ret

```

```

delay2:   mov r3,#09h
delay2a:  mov r2,#0fh
delay2b:  djnz r2,delay2b
        djnz r3,delay2a
        ret
        end

```




PRELIMINARY

80C51BH/80C51BH-1/80C51BH-2 CHMOS SINGLE-CHIP 8-BIT MICROCOMPUTER With Factory Mask-Programmable ROM

80C31BH/80C31BH-1/80C31BH-2 CHMOS SINGLE-CHIP 8-BIT CONTROL-ORIENTED CPU WITH RAM AND I/O

80C51BH/80C31BH — 3.5 TO 12 MHz, $V_{CC} = 5\text{ V} \pm 20\%$
80C51BH-1/80C31BH-1 — 3.5 TO 16 MHz, $V_{CC} = 5\text{ V} \pm 20\%$
80C51BH-2/80C31BH-2 — 0.5 TO 12 MHz, $V_{CC} = 5\text{ V} \pm 20\%$

CHMOS III Process
16K x 8-bit ROM
32 Programmable I/O Lines
Two 16-bit Timer/Counters
Full Duplex Serial Port

- High Performance CHMOS Process
- Boolean Processor
- 5 Interrupt Sources
- Programmable Serial Port
- 64K Data Memory Space

80C51BH/80C31BH products are fabricated on Intel's CHMOS III process and are functionally compatible with the standard MCS-51 HMOS and EPROM products. CHMOS III is a technology which combines the high performance characteristics of HMOS with the low power attributes of CHMOS. This combination expands the capabilities of the powerful MCS-51 architecture and instruction set.

In the 80C51BH/80C31BH versions, the MCS-51 CHMOS products have the following features: 4K byte of ROM (80C51BH-1/80C31BH-2 only); 128 bytes of RAM; 32 I/O lines; two 16-bit timer/counters; a five-pin serial port; a full duplex serial port; and on-chip oscillator and clock circuitry. In addition, 80C51BH/80C31BH products have two software selectable modes of reduced activity for further power saving: Power Down and Sleep.

In Power Down mode, the CPU is inactive while allowing the RAM, timer/counters, serial port and interrupt system to remain active. The Power Down mode saves the RAM contents but freezes the oscillator, causing all on-chip logic to be inoperative.

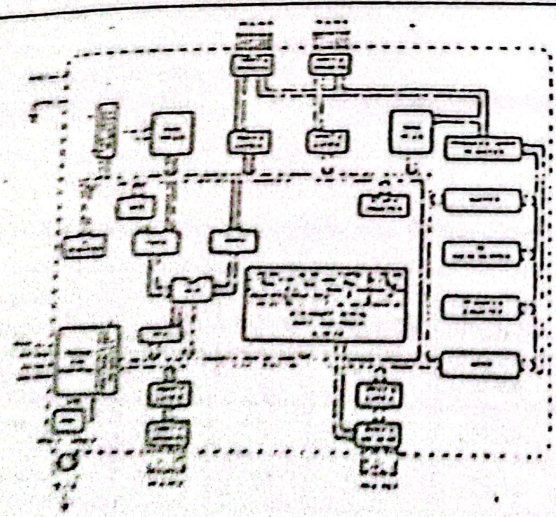


Figure 1. Block Diagram

Idle Mode

In the Idle mode, the CPU puts itself to sleep while the on-chip peripherals stay active. The instruction that invokes the Idle mode is the last instruction executed in the normal operating mode before Idle mode is activated. The content of CPU, the on-chip RAM, and all the Special Function Registers remain intact during this mode. The Idle mode can be terminated either by any enabled interrupt, at which time the process is picked up at the interrupt service routine and continued, or by a hardware reset which puts the processor the same as a power on reset.

Power Down Mode

In the Power Down mode the oscillator is stopped, and the instruction that invokes Power Down is the last instruction executed. Only the contents of the on-chip RAM is preserved. A hardware reset is the only way to terminate Power Down.

The control bits for the reduced power modes are in the Special Function Register PCON.

NOTE: For more detailed information on these reduced power modes refer to the 1985 Microcontroller Handbook.

DESCRIPTIONS

Supply voltage during normal, Idle, and Power Down operations.

to ground.

Port 0

Port 0 is an 8-bit open drain bi-directional I/O port. Pins that have 1's written to them float, and in

that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1s. Port 0 also outputs the code bytes during program verification in the 80C51BH. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during program verification.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL, on the data sheet) 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 3

Port 3 is an 8-bit bidirectional I/O port with internal

Table 1. Status of the external pins during Idle and Power Down modes

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

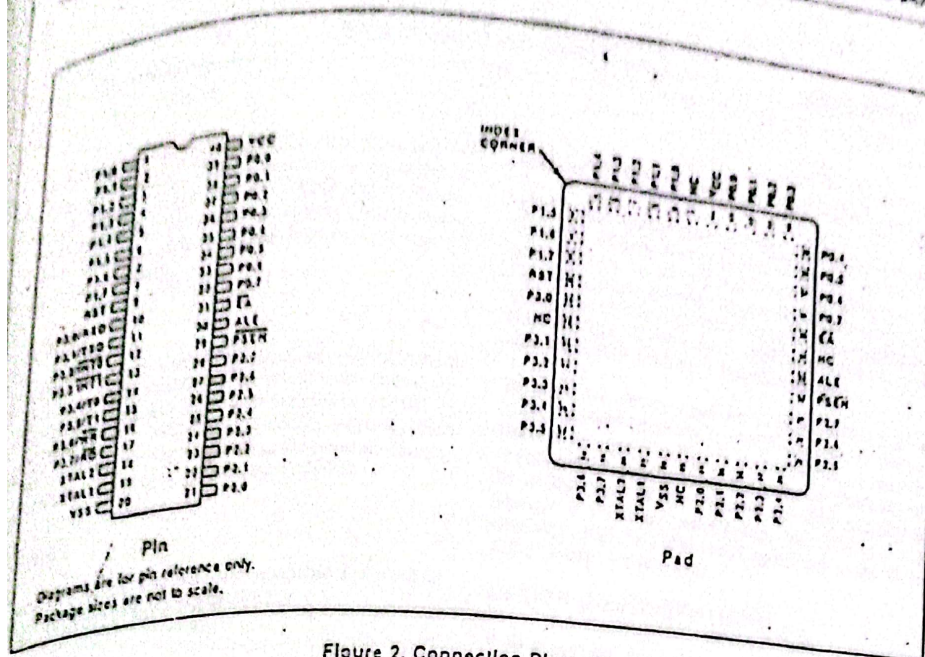


Figure 2. Connection Diagram

pullups. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL, on data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the MCS-51 Family, as listed below:

Port Pin	Alternate Function
P3.0	RXD [®] (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external Interrupt 0)
P3.3	INT1 (external Interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. An internal diffused resistor to VSS permits Power-On reset using only an external capacitor to VCC.

ALE

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory.

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.

PSEN

Program Store Enable is the read strobe to external Program Memory.

When the 80C51BH is executing code from external Program Memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external Data Memory. PSEN is not activated during latches from internal program memory.

EA

External Access enable. EA must be externally held low in order to enable the device to fetch code from external Program Memory locations 0000H to 0FFFH. If EA is held high the device executes from internal Program Memory unless the program counter contains an address greater than 0FFFH.

XTAL1

Input to the inverting oscillator amplifier and input to the internal block generator circuits.

XTAL2

Output from the inverting oscillator amplifier.

OSCILLATOR CHARACTERISTICS

XTAL1 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 3. More detailed information concerning the use of the on-chip oscillator is available in Application Note AP-155, "Oscillators for Microcontrollers," available from your Intel sales representative.

To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

DESIGN CONSIDERATIONS

- At power on, the voltage on VCC and RST must come up at the same time for a proper start-up.
- Before entering the Power Down mode the contents of the Carry Bit and B. 7 must be equal.

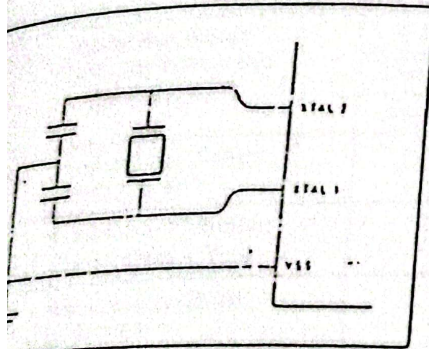


Figure 3. Crystal Oscillator

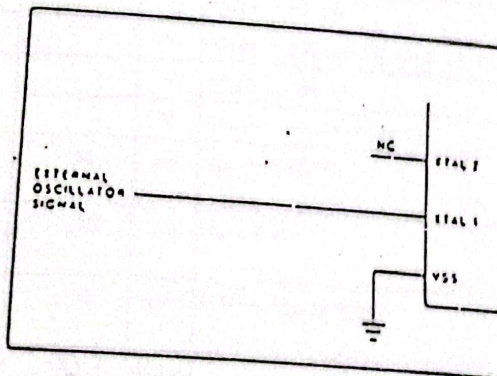


Figure 4. External Drive Configuration

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias	0°C to 70°C
Storage Temperature	-85°C to +150°C
Voltage on Any Pin to VSS	-0.5 V to VCC + 0.5 V
Voltage on VCC to VSS	-0.5 V to 6.5 V
Power Dissipation	1.0 W

*NOTICE: Stresses above 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 above 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.

D.C. CHARACTERISTICS: (TA = 0°C to 70°C; VCC = 5 V ± 20%; VSS = 0 V)

Symbol	Parameter	Min	Max	Unit	Test Conditions
VIL	Input Low Voltage (Except EA)	-0.5	.2VCC - .1	V	
VIL1	Input Low Voltage (EA)	-0.5	.2VCC - .3	V	
VIH	Input High Voltage (Except XTAL1, RST)	.2VCC + .9	VCC + 0.5	V	
VIH1	Input High Voltage (XTAL1, RST)	.7VCC	VCC + 0.5	V	
VOL	Output Low Voltage (Ports 1, 2, 3)		0.45	V	IOL = 1.6 mA (Note 1)
VOL1	Output Low Voltage (Port 0, ALE, PSEN)		0.45	V	IOL = 3.2 mA (Note 1)
VOH	Output High Voltage (Ports 1, 2, 3)	2.4		V	IOH = -60 µA VCC = 5 V ± 10%
		.75VCC		V	IOH = -25 µA
		.9VCC		V	IOH = -10 µA
VOH1	Output High Voltage (Port 0 in External Bus Mode, ALE, PSEN)	2.4		V	IOH = -400 µA VCC = 5 V ± 10%
		.75VCC		V	IOH = -150 µA
		.9VCC		V	IOH = -40 µA (Note 2)
IIL	Logical 0 Input Current (Ports 1, 2, 3)		-50	µA	Vin = 0.45 V
ITL	Logical 1 to 0 Transition Current (Ports 1, 2, 3)		-650	µA	Vin = 2.0 V
II	Input Leakage Current (Port 0, EA)		± 10	µA	0.45 < Vin < VCC
RST	Reset Pulldown Resistor	50	150	KOhm	
C	Pin Capacitance		10	pF	Test Freq = 1 MHz, TA = 25°C
ID	Power Down Current		50	µA	VCC = 2 to 6 V (Note 3)

*Value is based on the maximum allowable die temperature and the thermal resistance of the package.

Maximum static supply current operating at 5 V and $V_{DD} = 5.0 \text{ V}$ is 0.8 mA, and varies with frequency of the clock and data signals.

MAXIMUM ICC (mA)

Freq. V_{DD}	Operating (Mode 4)			Idle (Mode 5)	
	4 V	5 V	6 V	4 V	5 V
0.1 MHz	1.8	2.2	3	0.8	1.2
0.5 MHz	4.3	5.7	7.5	1.1	1.6
1.0 MHz	6.3	11	14	1.6	2.7
1.5 MHz	12	16	20	2.7	3.5
18 MHz	18	20.5	25	3.5	5

- Note 1: Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE and PSEN. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when those pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8 V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger STROBE input.
- Note 2: Capacitive loading on Ports 0 and 2 may cause the V_{OH} on ALE and PSEN to momentarily fall below the 5 V_{CC} specification when the address bits are stabilizing.
- Note 3: Power Down ICC is measured with all output pins disconnected; EA = Port 0 = V_{CC}; XTAL2 N.C.; RST = V_{SS}.
- Note 4: ICC is measured with all output pins disconnected; XTAL1 driven with TC_{LCH}, TC_{HCL} = 5 ns; V_A = V_{SS} = 5 V; V_{IN} = V_{CC} = 5 V; XTAL2 N.C.; EA = RST = Port 0 = V_{CC}. ICC would be slightly higher if a crystal oscillator is used.
- Note 5: Idle ICC is measured with all output pins disconnected; XTAL1 driven with TC_{LCH}, TC_{HCL} = 5 ns; V_A = V_{SS} = 5 V; V_{IN} = V_{CC} = 5 V; XTAL2 N.C.; Port 0 = V_{CC}; EA = RST = V_{SS}.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

- A: Address.
- Cl: Clock.
- I: Input data.
- H: Logic level HIGH.
- Inst: Instruction (program memory contents).
- L: Logic level LOW, or ALE.

- P: PSEN.
- O: Output data.
- R: READ signal.
- T: Time.
- V: Valid.
- W: WRITE signal.
- X: No longer a valid logic level.
- Z: Float.

EXAMPLE:
 TAVLL = Time for Address Valid to ALE Low.
 TLLPL = Time for ALE Low to PSEN Low.

intel

80C51BH/80C51BH-1/80C51BH-2
80C31BH/80C31BH-1/80C31BH-2

PRELIMINARY

A.C. CHARACTERISTICS: (TA = 0°C to 70°C, VCC = 5 V ± 20%, VSS = 0 V, Load Capacitance for Port 0, ALE, and PSEN = 100 pF, Load Capacitance for All Other Outputs = 50 pF)

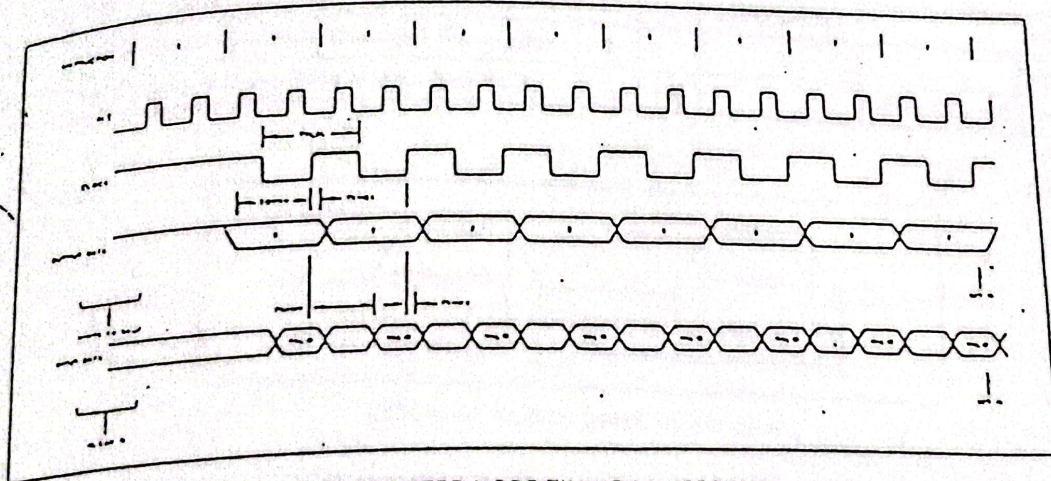
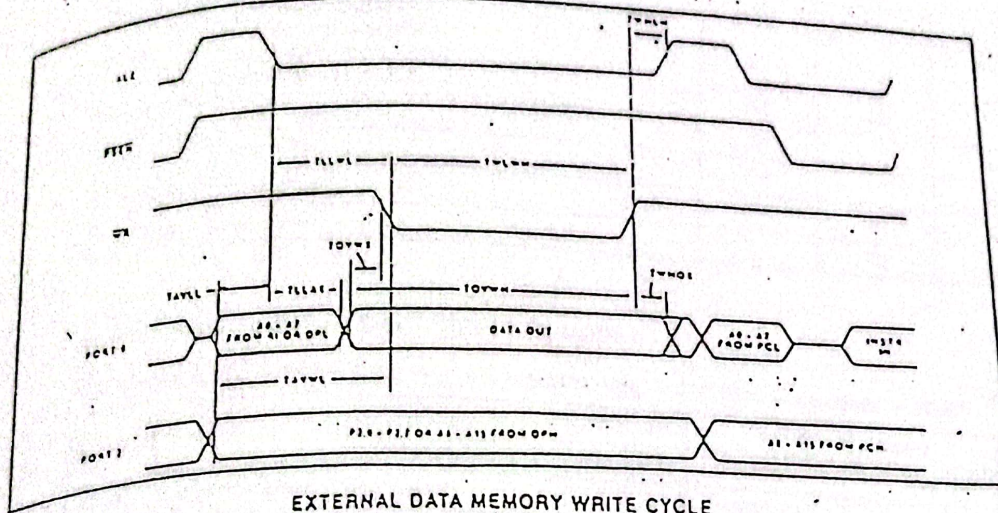
EXTERNAL PROGRAM AND DATA MEMORY CHARACTERISTICS

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency 80C51BH/80C31BH 80C51BH-1/80C31BH-1 80C51BH-2/80C31BH-2			3.5 3.5 0.5	12 16 12	MHz
TLHLL	ALE Pulse Width	127		2TCLCL - 40		ns
TAVLL	Address Valid to ALE Low	29		TCLCL - 55		ns
TLLAX	Address Hold After ALE Low	48		TCLCL - 35		ns
TLLIV	ALE Low to Valid Instr In		234		4TCLCL - 100	ns
TLLPL	ALE Low to PSEN Low	43		TCLCL - 40		ns
TPLPH	PSEN Pulse Width	205		3TCLCL - 45		ns
TPLIV	PSEN Low to Valid Instr In		145		3TCLCL - 105	ns
TPXIX	Input Instr Hold After PSEN	0		0		ns
TPXIZ	Input Instr Float After PSEN		59			ns
TAVIV	Address to Valid Instr In		112		TCLCL - 25	ns
TPLAZ	PSEN Low to Address Float		10		5TCLCL - 105	ns
TRLRH	RD Pulse Width	400		6TCLCL - 100	10	ns
TWLWH	WR Pulse Width	400		6TCLCL - 100		ns
TRLDV	RD Low to Valid Data In		252		5TCLCL - 165	ns
TRHOX	Data Hold After RD	0		0		ns
TRHOZ	Data Float After RD		97			ns
TLLDV	ALE Low to Valid Data In		517		2TCLCL - 70	ns
TAVDV	Address to Valid Data In		505		8TCLCL - 150	ns
TLLWL	ALE Low to RD or WR Low	200	300	3TCLCL - 50	9TCLCL - 165	ns
TAVWL	Address Valid to Write Low	203		4TCLCL - 130	3TCLCL + 50	ns
TOVWX	Data Valid to WR Transition	23		TCLCL - 60		ns
TWHOX	Data Hold After WR	33		TCLCL - 50		ns
TALAZ	RD Low to Address Float		0		0	ns
TWHLH	RD or WR High to ALE High	43	123	TCLCL - 40	TCLCL + 40	ns

intel®

80C51BH/80C51BH-1/80C51BH-2
80C31BH/80C31BH-1/80C31BH-2

PRELIMINARY



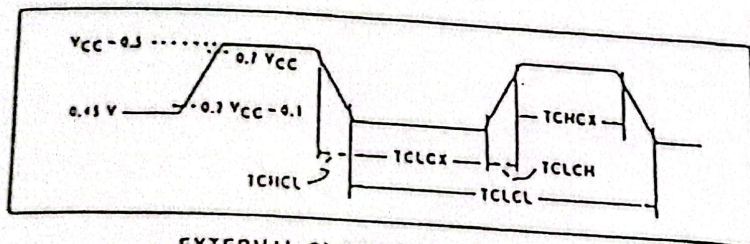
EXTERNAL CLOCK DRIVE

Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency 80C51BH/80C31BH 80C51BH-1/80C31BH-1 80C51BH-2/80C31BH-2	3.5 3.5 0.5	12 16 12	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

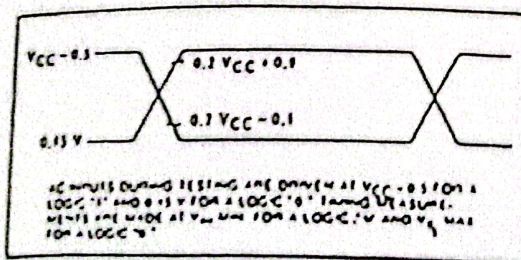
SERIAL PORT TIMING — SHIFT REGISTER MODE

Test Conditions: $T_A = 0^\circ\text{C to } 70^\circ\text{C}$; $V_{CC} = 5\text{ V} \pm 20\%$; $V_{SS} = 0\text{ V}$; Load Capacitance = 80 pF

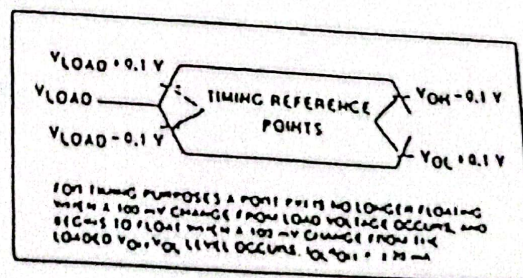
Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1.0		12TCLCL		μs
TOVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL - 133		ns
TXHOX	Output Data Hold After Clock Rising Edge	50		2TCLCL - 117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL - 133	ns



EXTERNAL CLOCK DRIVE WAVEFORM



AC TESTING INPUT, OUTPUT WAVEFORMS

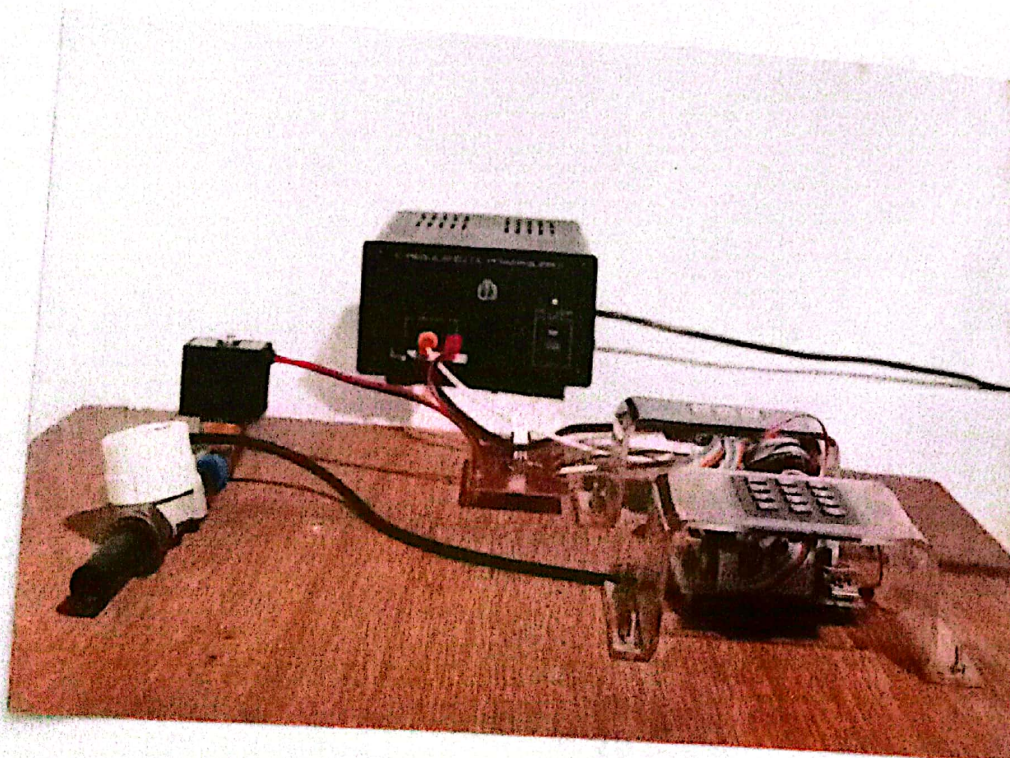


FLOAT WAVEFORMS

LAMPIRAN C
FOTO - FOTO SISTEM



Tampak Alat Bagian Atas



Tampak Alat dari Depan

LAMPIRAN D
DATA SHEET SENSOR



FLOW SENSORS - 15mm DIA. PIPE

Stock Nos. 257-133, 257-149

GENERAL

These flow transducers have been designed for use with mains heating system waters. Subject to customer evaluation, it is also possible to use them with demineralised water, dilute or dilute alkaline solutions, hydrocarbon or mild oxygenated solvents (including light fuel oils and spirits) and weak organic acids depending on concentration (generally up to 5% at ambient temperature is acceptable) and temperature.

It is not possible to use these flow transducers with strong mineral acids or acid solutions at any temperature or pressure, nor with strong or searching solvents (e.g. ethylene chloride, acetone, alcohols, chloroform).

INSTALLATION

At installation check the free running of the sensor by passing fluid through it. DO NOT USE AN AIR LINE. The flow sensor should be installed with the arrow on the housing pointing vertically upwards. The unit may, however, be installed in any attitude providing the pipes are full of liquid. Pockets of vapour or bubbles of air may affect some of the light and so alter the number of pulses received from the sensor. On the lower flow model these bubbles may take some time to clear because of the low fluid velocity inside the chamber. A large back pressure will overcome any tendency the liquid has to form vapour pockets. As the detection system is optical it is undesirable to install the sensor near a strong light source or in direct sunlight, as this may "swamp" the detection system.

PULSATIONS

Pulsations cause pulsations in the fluid, centrifugal pumps have the lowest disturbance, and reciprocating pumps the highest. With a centrifugal pump the pulsations reduce after a short pipe run so if the flow sensor is positioned as far from the pump as possible, the effects will be minimized. With reciprocating pumps more positive isolation is desirable and a damper or accumulator is probably required. If pulsations cannot be removed the unit must be recalibrated to the desired flow range.

CONSIDERATIONS

All flow sensors should only be installed with the following in mind: bends, valves, flow regulators, tee junctions and other fittings which cause the flow to travel faster at one side of the pipe relative to the other. This asymmetry in the flow can seriously affect the calibration and the disturbance should be as far removed from the sensor as possible and if at all practical, put after the flow sensor. Recalibration on site will, of course, remove any doubt about the installation. In all cases an 80 mesh filter should be fitted upstream of the flow sensor.

VISCOSITY

Viscosity effects:- all turbine transducers are affected by viscosity and where possible the viscosity (temperature) of the liquid should be kept fairly constant. Viscous drag causes the turbine to be slowed down quicker at the lower flows, as viscosity increases so does the threshold to operation. If the fluid is lubricating and a higher pressure drop is acceptable, the turbine can be run at up to 50% over range with no detrimental effects.

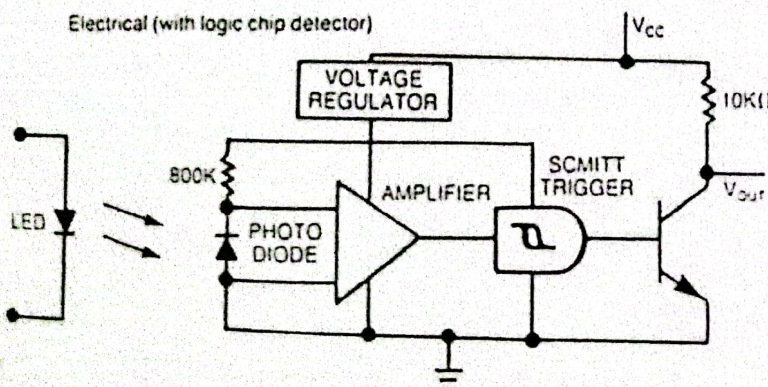
ELECTRONIC

Inside the housing is an infra-red light emitting diode. The LED is directed at the receiver which has a built-in voltage regulator, photo diode, amplifier, schmitt trigger and output stage. Every turbine blade passage blocks the light beam and so reduces the level of the output signal to typically 200mV (maximum 400mV). The "unblocked" level is the supply voltage and the output is tied to this through a 10k resistor. The output rise and fall times are typically 60ns and 6ns respectively.

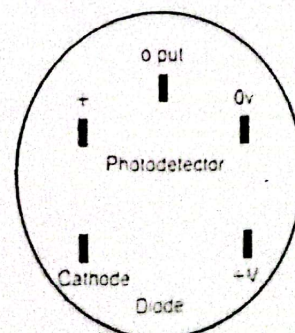
At the maximum continuous operating temperature (70°C) the power dissipation of the detector output stage is 2.5mW, care must therefore be taken not to overload the output. At ambient temperatures the unit can dissipate up to 250mW.

The power to the LED is dependent on supply current, and this MUST be limited with a series resistor and potentiometer. Current required is subject to optical density of medium being measured. Maximum recommended current = 30mA.

Electrical (with logic chip detector)



Electrical Connection



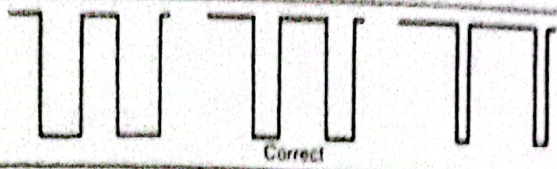
CODE **Maximum Ratings**

Operating voltage @ 25°C free air temperature	3.0V
Continuous forward current at (or below) 25°C	60mA
Operating temperature	30mA
Recommended current	-40°C to 70°C
Operating free air temperature range	-40°C to 70°C
Storage temperature range	260°C
Lead temperature 1/16th inch from case for 5 sec	100mW
Power dissipation	

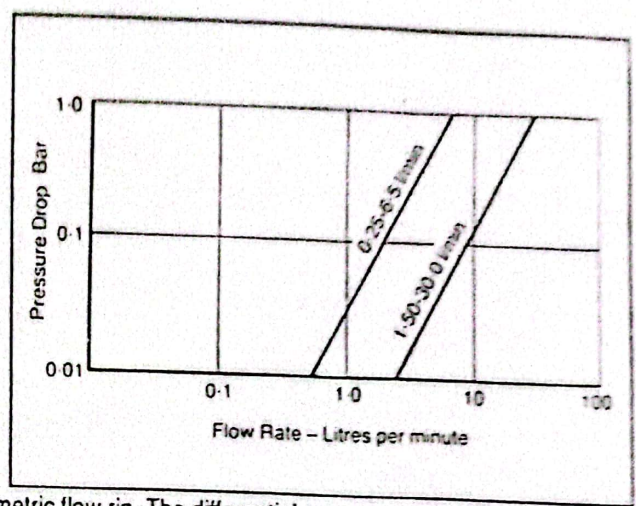
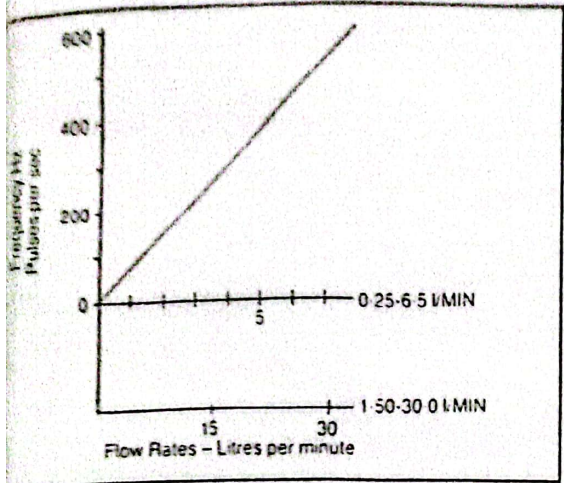
PHOTODETECTOR **(Typical)**

Max. dissipation	250mW @ 25°C
Operating supply voltage	4.5 to 10Vdc supply
Output low	5mA
Supply current	7mA
Output high	5.4mA
Supply current	5mA
Output voltage low	0.2V
Output voltage high	2V
Rise time	50ns
Fall time	5ns

At the maximum continuous operating temperature (70°C) the power dissipation of the detector output stage is 100mW. Care must therefore be taken not to overload the output.



Pulses from the sensor when viewed on an oscilloscope should look like the central drawing above. If LED current is too high, an increase in ambient light levels could cause the signal to stay high. A 50:50 mark space ratio is possible as the blades are thinner than the spaces.



Graphs show results of flowtransducers calibrated on a gravimetric flow rig. The differential pressure is set using a rising manometer. Water temperature 8°C

LOUR CODE	FLOW RANGE	LINEARITY at FSD	APPROXIMATE FULL SCALE FREQUENCY	APPROXIMATE PULSES PER LITRE AT FULL SCALE
ey body	0.25-6.5 l/m	±1.0%	500Hz	4600 pulses/litre
ige body	1.50-30.0 l/m	±1.5%	600Hz	1200 pulses/litre

Materials of Construction

- Acetal Copolymer
- ng Cap - Acetal Copolymer
- Acetal Copolymer
- ousing - Clear Nylon
- Stainless Steel
- Sapphire
- Nitile Rubber

Standard Connections

- 15mm Cu x 15mm Cu (compression fittings)
- Operating Pressure
- Max. 300lbs/sq ins
- Dimensions mm
- Length =89 (Over fittings)
- Height =51
- Breadth =37