

**PERANCANGAN DAN PEMBUATAN SIMULASI LAMPU
LALU LINTAS DENGAN MENGGUNAKAN REAL TIME
CLOCK DAN ALAT PENGONTROL JARAK JAUH
BERBASIS MIKROKONTROLER AT89S52**

TUGAS AKHIR

Sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains
dalam bidang Fisika

Oleh:

TANTO ADI WALUYO
0001100164-93



JURUSAN FISIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN
ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2007

LEMBAR PENGESAHAN TUGAS AKHIR

PERANCANGAN DAN PEMBUATAN SIMULASI LAMPU LALU LINTAS DENGAN MENGGUNAKAN REAL TIME CLOCK DAN ALAT PENGONTROL JARAK JAUH BERBASIS MIKROKONTROLER AT89S52

Oleh :

TANTO ADI WALUYO
0001100164-93

Setelah dipertahankan di depan Majelis Penguji

Pada Tanggal

dan dinyatakan memenuhi syarat untuk memperoleh gelar Sarjana
Sains dalam bidang Fisika

Pembimbing I

Pembimbing II

Dr.Ing. Setyawan P.S,M.Eng
NIP. 131 879 401

Ahmad Nadhir, S.Si., M.T.
NIP. 132 232 182

Mengetahui,

Ketua Jurusan Fisika
Fakultas MIPA Universitas Brawijaya

Drs. Adi Susilo, PhD.
NIP. 131 960 447

LEMBAR PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama :TANTO ADI WALUYO
NIM : 0001100164-93
Jurusan : Fisika
Penulis Tugas Akhir : Perancangan dan Pembuatan
Simulasi Lampu Lalu Lintas
Dengan Menggunakan Real
Time Clock dan Pengontrol
Jarak Jauh berbasis
Mikrokontroler AT89S52.

Dengan ini menyatakan bahwa :

1. Isi dari Tugas Akhir yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Tugas Akhir ini.
2. Apabila dikemudian hari ternyata Tugas Akhir yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, Juli 2007

Yang Menyatakan,

TANTO ADI WALUYO
NIM. 0001100164-93

PERANCANGAN DAN PEMBUATAN SIMULASI LAMPU LALU LINTAS DENGAN MENGGUNAKAN REAL TIME CLOCK DAN ALAT PENGONTROL JARAK JAUH BERBASIS MIKROKONTROLER AT89S52

ABSTRAK

Tugas akhir ini berisi tentang perancangan dan pembuatan simulasi lampu lalu lintas dengan menggunakan *real time clock* dan alat pengontrol jarak jauh berbasis mikrokontroler AT89S52. Sistem ini terdiri dari sebuah pemancar dan penerima. Secara umum sistem bekerja dengan memasukan waktu tunda ke sistem pemancar. Sinyal dari sistem pemancar akan dipancarkan ke sistem penerima melalui radio pengirim penerima 144 MHz. Sinyal dari sistem pemancar akan diolah oleh sistem penerima untuk mengontrol pewaktuan dari *traffic light*.

Dari hasil pengujian, sistem pemancar dan sistem penerima telah dapat berkomunikasi satu arah dengan baik. Perangkat lunak yang digunakan pada sistem pemancar dan sistem penerima telah bekerja dengan baik.

DESIGN AND MAKING SIMULATION OF TRAFFIC LIGHT BASED ON REAL TIME CLOCK AND LONG DISTANCE CONTROLLER USING MICROCONTROLLER AT89S52

ABSTRACT

It has been designed a system to simulate a traffic light based on real time clock and long distance controller using a microcontroller AT89S52. The system contained of a transmitter and a receiver. The system worked based on adjustment of a delay time on the transmitter. The signal from the transmitter was transmitted to the receiver through a handy talkie. The signal was proceed by the receiver to control traffic light.

The result showed the transmitter and the receiver could communicate well in an direction. The software used to control between the transmitter and receiver worked successfully.

KATA PENGANTAR

Alhamdulillahillobbi ‘alamin, segala puji dan syukur kepada Allah SWT, Tuhan semesta alam. Karena rahmat dan kurnia-Nya penulis dapat menyelesaikan tugas akhir yang berjudul **“Perancangan dan Pembuatan Simulasi Lampu Lalu Lintas Dengan Menggunakan Real Time Clock dan Alat Pengontrol Jarak Jauh Berbasis Mikrokontroler AT89S52”**.

Tugas akhir ini disusun sebagai syarat dalam meraih gelar Sarjana Sains (S.si) di Jurusan Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya.

Ungkapan terimakasih saya sampaikan sebesar-besarnya kepada:

1. Bapak Dr. Ing. Setyawan P.S.,M.Eng, selaku dosen pembimbing I, atas ide-ide yang membuat skripsi ini dapat diselesaikan.
2. Ahmad Nadhir, S.Si., M.T, selaku dosen pembimbing II, atas saran, kritik dan bimbingannya.
3. Bapak Drs. Adi Susilo,Phd, selaku ketua jurusan Fisika FMIPA-UB
4. Juga kepada semua bapak dan Ibu dosen Fisika, yang telah banyak memberikan ilmu pada penulis.

Penulis menyadari bahwa penulisan tugas akhir ini masih jauh dari sempurna. Untuk itu penulis mengharapkan saran dan kritik yang dapat mengembangkan tugas akhir ini.

Malang, Juli 2007

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
HALAMAN PERNYATAAN	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
DAFTAR LAMPIRAN	xi
 BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
 BAB II DASAR TEORI	
2.1 Gelombang Radio.....	5
2.2 Mikrokontroler AT89S52	5
2.2.1 Memori Mikrokontroler AT89S52.....	7
2.2.1.1 Memori Program.....	9
2.2.1.2 Memori Data.....	9
2.2.2 Register-Register AT89S52.....	9
2.2.2.1 Register R0-R7	9
2.2.2.2 <i>Special Function Register (SFR)</i>	9
2.2.3 Pewaktuan.....	13
2.2.4 Interupsi	13
2.2.5 <i>Reset</i>	14
2.3 <i>Dual Tone Multiple Frequency (DTMF)</i>	14
2.3.1 <i>Encoder Dual Tone Multiple Frequency TP 5088N</i> . ..	15
2.3.2 <i>DecoderDual Tone Multiple Frequency</i>	17
2.4 Radio Pemancar Penerima (RPP) FM 144 MHz	18
2.5 <i>Liquid Crystal Display (LCD)</i>	18

2.6	<i>Keypad dan Encoder MM 74C922</i>	19
2.7	<i>Real Time Clock DS1302</i>	20
2.8	<i>Relay</i>	21

BAB III METODE PENELITIAN

3.1	Perencanaan Perangkat Keras	23
3.1.1	Rangkaian Pengirim.....	24
3.1.1.1	<i>Encoder DTMF TP 5088N</i>	24
3.1.1.2	<i>Keypad dan Encoder MM 74C922</i>	25
3.1.1.3	Mikrokontroler AT89S52	26
3.1.1.4	LCD	28
3.1.2	Rangkaian Penerima	29
3.1.2.1	<i>Decoder DTMF MT5570</i>	29
3.1.2.2	Mikrokontroler AT89S52	30
3.1.2.3	<i>Real Time Clock DS 1302</i>	32
3.2	Perencanaan Perangkat Lunak.....	32
3.2.1	Perangkat Lunak Mikrokontroler Pengirim.....	32
3.2.2	Perangkat Lunak Mikrokontroler Penerima.....	33
3.2.2.1	Jalankan Miniatur.....	33
3.2.2.2	Pembacaan Mode Masukan	36
3.3	Pengujian Alat.....	37
3.3.1	Pengujian <i>Real Time Clock DS1302</i>	38
3.3.2	Pengujian <i>Relay</i>	38

BAB IV ANALISA DAN PEMBAHASAN

4.1	Hasil Pengujian	39
4.2	Pembahasan.....	40
4.2.1	Keseluruhan Sistem	41
4.2.2	Perangkat Keras.....	41
4.2.3	Perangkat Lunak.....	43

BAB V KESIMPULAN DAN SARAN

5.1	Kesimpulan.....	45
5.2	Saran.....	45

DAFTAR PUSTAKA	47
-----------------------------	----

DAFTAR GAMBAR

Gambar 2.1	Konfigurasi pin AT89S52.....	6
Gambar 2.2	Peta Memori Internal Mikrokontroler AT89S52...	8
Gambar 2.3	Peta Memori <i>Special Function Register</i> (SFR) AT89S52.....	10
Gambar 2.4	Pewaktuan.....	13
Gambar 2.5	Reset.....	14
Gambar 2.6	Konfigurasi Pin TP 5088	16
Gambar 2.7	Konfigurasi pin MT 8870	17
Gambar 2.8	Konfigurasi pin LCD	19
Gambar 2.9	Konfigurasi pin MM74C922.....	20
Gambar 2.10	Konfigurasi pin RTC DS1302.....	21
Gambar 2.11	Jenis-jenis Relay	22
Gambar 3.1.	Diagram Blok Pengirim	23
Gambar 3.2.	Diagram Blok Penerima.....	24
Gambar 3.3	Perencanaan rangkaian TP 5088N	25
Gambar 3.4	Perencanaan rangkaian keypad dengan encoder MM 74C922.....	26
Gambar 3.5	Perencanaan penggunaan port pada AT89S52 sebagai modul pengirim.....	27
Gambar 3.6	Perencanaan rangkaian LCD Seiko M 1632.....	28
Gambar 3.7	Perencanaan rangkaian CM8870.....	29
Gambar 3.8	Perencanaan penggunaan port pada AT89S52 sebagai modul penerima.....	30
Gambar 3.9	Perencanaan rangkaian DS1302.....	32
Gambar 3.10	Diagram alir prosedur pada mikrokontroler Pengirim	33
Gambar 3.11	Diagram alir jalankan miniatur.....	35
Gambar 3.12	Diagram alir penentuan waktu tunda.....	36
Gambar 3.13	Diagram alir rutin cek waktu.....	36
Gambar 3.14	Diagram alir prosedur interupsi.....	37
Gambar 3.15	Diagram blok pengujian RTC.....	38
Gambar 3.16	Diagram blok pengujian Relay.....	38

DAFTAR TABEL

Tabel 2.1. Alokasi Frekuensi Radio	5
Tabel 2.2 Alokasi Frekuensi DTMF	15

UNIVERSITAS BRAWIJAYA



DAFTAR LAMPIRAN

Lampiran 1 <i>Schematic</i> Rangkaian Pengirim	49
Lampiran 2 <i>Schematic</i> Rangkaian Penerima	51
Lampiran 3 <i>Source Code</i> Rangkaian Pengirim.....	53
Lampiran 4 <i>Source Code</i> Rangkaian Penerima.....	57
Lampiran 5 <i>Source Code</i> Pengujian RTC DS1302 dan Relay....	71
Lampiran 6 <i>Source Code Driver</i> RTC DS1302.....	73



BAB I

PENDAHULUAN

1.1 Latar Belakang

Kelancaran arus lalu-lintas dipengaruhi oleh ketertiban pengguna jalan dalam mematuhi peraturan lalu lintas. Peraturan ini dapat berupa rambu-rambu lalu lintas. Salah satu rambu-rambu lalu-lintas adalah lampu lalu-lintas yang dipasang di persimpangan jalan di mana kepadatan arus kendaraan yang melintas sangat tinggi. Lampu lalu-lintas berfungsi sebagai pengatur kesempatan bergerak kendaraan dengan urutan tertentu.

Pembagian waktu bergerak kendaraan merupakan salah satu unsur yang penting dalam mengatur kelancaran arus lalu-lintas. Semakin tinggi kepadatan arus kendaraan yang melewati sebuah persimpangan jalan, kesempatan untuk bergerak kendaraan harus dibuat lebih lama.

Jumlah kendaraan yang melewati persimpangan jalan tidaklah selalu sama antara arah yang satu dengan arah lainnya. Oleh karena itu, jika kesempatan bergerak pada setiap arah dibuat sama, maka akan ada pengguna jalan yang dirugikan karena menunggu kesempatan untuk bergerak padahal arah yang ditunggu tidak ada kendaraan yang melintas.

Di setiap persimpangan jalan juga memiliki kepadatan kendaraan dan keadaan yang berbeda-beda tergantung pada waktu. Pada saat orang berangkat ke tempat kerja atau sekolah, maka kepadatan jalan raya akan tinggi.

Pada suatu keadaan tertentu, dibutuhkan pemblokiran ruas jalan untuk kepentingan tertentu. Solusi selama ini untuk keadaan seperti ini adalah dengan memberikan tanda dilarang melintas atau pengaturan secara manual oleh polisi tanpa mempedulikan lampu lalu-lintas.

Berdasarkan alasan-alasan di atas, diperlukan seperangkat sistem kontrol lampu lalu-lintas yang dinamis. Untuk mengganti jeda waktu kesempatan bergerak kendaraan, digunakan pengendali jarak jauh.

Pengaturan sistem menggunakan sebuah mikrokontroler AT89S52, *Dual Tone Mutiple Frequency* (DTMF) sebagai piranti antar muka sistem dan media transmisi datanya menggunakan *handy talkie* 144 MHz yang merupakan peralatan yang biasa digunakan oleh polisi.

Sebenarnya, alat untuk mengganti waktu tunda pada sebuah lampu lalu lintas pernah dibuat. Namun, alat itu masih mempunyai kelemahan sebagai berikut :

- Hanya mampu merubah waktu tunda lampu hijau dari 1 sampai 99 detik sehingga diperlukan perbaikan agar waktu tunda yang dimasukkan bisa lebih lama.
- Setiap lampu hijau harus memiliki waktu tunda sehingga tidak bisa melakukan pemblokiran jalan pada arah tertentu.
- Lampu yang digunakan masih berupa LED sehingga tidak bisa menggunakan tegangan 220 V yang biasa digunakan pada lampu lalu lintas yang sesungguhnya.

1.2 Perumusan Masalah

Permasalahan yang diteliti adalah :

1. Bagaimana merancang dan membuat sistem berbasis mikrokontroler untuk mengatur simulasi siklus lalu lintas.
2. Bagaimana memanfaatkan DTMF sebagai piranti antar muka untuk mengatur sistem yang diatur dari jarak jauh untuk mengirim data sebanyak 5 kali dalam setiap proses.
3. Bagaimana merencanakan dan membuat perangkat lunak pada mikrokontroler pemancar (untuk mengirim data) dan penerima (untuk menerjemahkan data yang diterima).
4. Bagaimana memanfaatkan *Real Time Clock* (RTC) untuk referensi pewaktuan

1.3 Batasan Masalah

Untuk menyederhanakan pembahasan dalam perencanaan dan pembuatan alat ini diambil batasan masalah antara lain:

1. Tidak dibahas tentang pesawat transmitter dan receiver yang digunakan (Radio Pengirim Penerima 144 MHz).
2. Modem yang digunakan DTMF.

3. Digunakan lampu 2,5 Watt sebagai pengganti lampu lalu lintas.
4. Setiap pengiriman data, hanya salah satu arah yang diatur.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah merancang dan membuat simulasi lampu lalu lintas dengan alat pengendali jarak jauh untuk mengganti waktu tunda pada lampu lalu lintas dengan menggunakan RTC sebagai referensi pewaktuan.

1.5 Manfaat Penelitian

Alat yang dihasilkan pada penelitian ini diharapkan dapat digunakan sebagai lampu lalu lintas yang dinamis sehingga dapat membantu polisi untuk mengatur pergerakan kendaraan tanpa harus mengatur secara manual di jalan raya.



UNIVERSITAS BRAWIJAYA



BAB II

TINJAUAN PUSTAKA

2.1 Gelombang Radio

Komunikasi radio menggunakan gelombang elektromagnetik yang dipancarkan lewat atmosfer bumi atau ruang bebas untuk membawa informasi. Gelombang elektromagnetik dipancarkan pada ruang yang dipergunakan bersama sehingga besar kemungkinan untuk saling mengganggu. Gangguan ini dapat dicegah dengan menentukan alokasi frekuensi perhubungan radio. Pembagian dan alokasi frekuensi radio diperlihatkan Tabel 2.1 (Roody,1996).

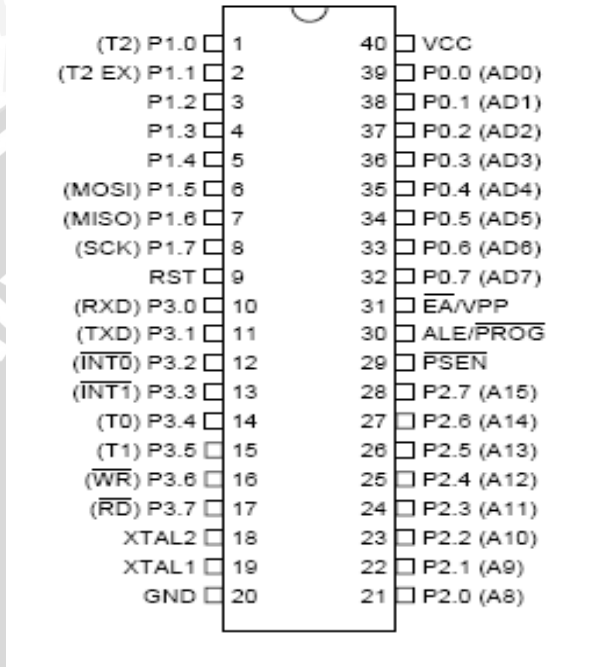
Tabel 2.1. Alokasi Frekuensi Radio

Nama	Frekuensi
Extremely Low Frequency	Kurang dari 3 KHz
Very Low Frequency	3 – 30 KHz
Low Frequency	30 – 300 KHz
Medium Frequency	300 – 3000 KHz
High Frequency	3 – 30 MHz
Very High Frequency	30 – 300 MHz
Ultra High Frequency	300 – 3000 MHz
Super High Frequency	3 – 30 GHz
Extremely High Frequency	30 – 300 GHz

2.2 Mikrokontroler AT89S52

Mikrokontroler AT89S52 adalah mikrokontroler yang memiliki memori program dalam satu chipnya (internal chip program memory) dengan kapasitas memori 8 kilo byte. AT89S52 mempunyai 256 byte RAM internal, 4 buah I/O port 8 bit, 3 timer

16 bit. AT89S52 dapat di pakai dalam mode *single chip* atau mode *expande*. AT89S52 dapat digunakan dengan menambahkan kristal untuk pewaktu (clock) dan power on reset.



Gambar 2.1. Konfigurasi pin AT89S52

(Sumber: www.keil.com/dd/docs/datashts/atmel/at89s52_ds.pdf)

Fungsi dari masing-masing pin dapat dikelompokkan menjadi tiga bagian yaitu bagian catu daya, bagian port I/O 8 bit dan bagian jalur interupsi. Bagian catu daya meliputi Vcc sebagai pin positif sumber tegangan 5V DC dan Vss sebagai pin ground sumber tegangan. Bagian port I/O 8 bit terdiri dari port 0 (P0), port 1 (P1) port 2 (P2) dan port 3 (P3). Port 3 dapat juga difungsikan sebagai port masukan serial (RxD pada P3.0), port keluaran serial (TxD pada P3.1), untuk intrupsi eksternal 0 (INT0 pada P3.2) untuk interupsi eksternal 1 (INT1 pada P3.3), strobe untuk penulisan memori data eksternal WR pada P3.6), masukan eksternal timer 0 (T0 pada P3.4), strobe untuk pembacaan memori data eksternal (RD pada P3.7). Bagian jalur interupsi terdiri dari Reset, ALE , PSEN,

EA/Vpp, XTAL1, XTAL2. Reset berfungsi memberikan logika tinggi (high) pada pin ini selama 2 siklus mesin selama osilator berjalan akan mengakibatkan IC ini menjadi reset. ALE berfungsi mengeluarkan pulsa untuk menahan (latch) alamat byte rendah selama mengakses memori eksternal. PSEN berfungsi memberikan pulsa *strobe* ke memori program eksternal ketika menjalankan kode dari memori program eksternal. PSEN tidak aktif selama mengambil/menjalankan program internal. EA/Vpp, EA harus berlogika rendah jika menjalankan program mulai dari alamat 0000H sampai alamat terbesar dari EEPROM internal. Jika EA berlogika tinggi, maka program yang dijalankan adalah program yang terdapat di ROM internal selama program counter mengeluarkan alamat 0000H sampai dengan 0FFFH. XTAL1 berfungsi sebagai masukan ke *inverting oscillator amplifier* dan masukan ke rangkai pembangkit clock internal. XTAL2 berfungsi sebagai keluaran dari *inverting oscillator amplifier*.

2.2.1. Memori Mikrokontroler AT89S52

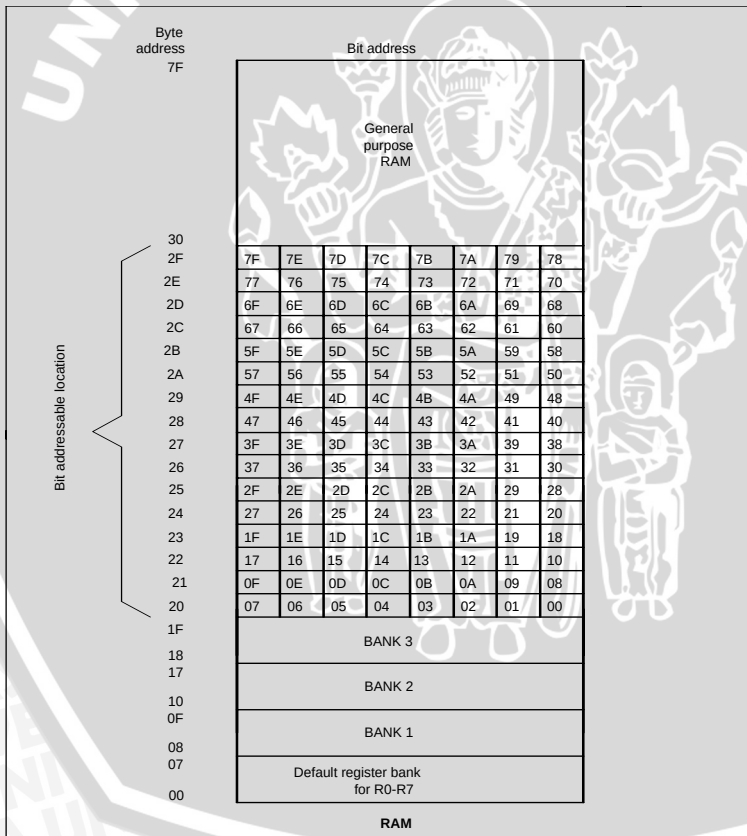
Organisasi memori mikrokontroler AT89S52 dibagi atas dua bagian yang berbeda, yaitu memori program dan memori data. Pembagian itu berdasarkan fungsinya dalam penyimpanan data atau program. Memori program digunakan untuk instruksi yang akan dijalankan oleh mikrokontroler. Sedangkan memori data digunakan sebagai tempat penyimpanan data-data yang sedang dan akan diakses oleh mikrokontroler.

Untuk pemisahan memori program dan memori data dilakukan dengan memakai memori data pada 8 bit dan 16 bit yang dihasilkan melalui DPTR. Mikrokontroler AT89S52 mempunyai 5 buah ruang alamat yaitu:

1. Ruang alamat kode sebanyak 64 Kbyte yang semuanya merupakan ruang alamat kode eksternal (*off-bit*).
2. Ruang alamat data internal yang dialamati secara langsung seperti RAM sebanyak 128 byte dan hardware register sebanyak 128 byte.
3. Ruang data internal yang dialamati tidak langsung sebanyak 128 byte.
4. Ruang alamat data eksternal 64 byte yang didapat ditambahkan oleh pemakai.

- Ruang alamat bit dapat diakses dengan pengalamatan secara langsung.

Banyaknya *stack* dibatasi oleh kemampuan RAM data internal. Lokasinya ditentukan lewat 8 bit *stack pointer*. Semua register kecuali *program counter* dan *register bank* berada dalam ruang alamat register fungsi khusus (SFR). Register lain termasuk dalam SFR adalah register aritmatika, *pointer*, *port input/output* dan register untuk *interrupt*, *timer*, dan *serial chanel*. Lokasi-lokasi 128 byte dalam alamat SFR dapat dialamati secara per bit. Secara keseluruhan AT89S52 mempunyai 256 byte RAM data internal dan 20 register fungsi khusus (SFR). Instruksi-instruksi dalam AT89S52 terdiri dari 49 buah *single byte*, 45 buah *two byte*, dan 17 buah *three byte*.



Gambar 2.2 Peta Memori Internal pada Mikrokontroler AT89S52

2.2.1.1. Memori Program

Pada saat pin EA dihubungkan dengan ground, maka semua program akan berjalan di memori eksternal. Pada saat Pin EA dihubungkan dengan pin Vcc, maka dari alamat 0000H-1FFFFH akan berjalan di memori internal.

2.2.1.2. Memori Data

Mikrokontroler AT89S52 memiliki memori data berupa RAM internal sebesar 256 byte. Memori data ini dapat diakses baik langsung atau tidak langsung. Selain mengakses memori data internal mikrokontroler AT89S52 juga dapat mengakses memori data eksternal. Lebar jalur alamat yang dapat diakses adalah 16 bit yaitu mulai alamat 0000H – FFFFH. Konfigurasi perangkat kerasnya sama dengan memori program eksternal. Hanya saja sinyal PSEN tidak digunakan untuk membaca data tetapi yang digunakan adalah sinyal RD, sedangkan untuk penulisan digunakan sinyal WR.

2.2.2. Register-register pada AT89S52.

AT89S52 ini mempunyai 8 bit register R0 – R7 dengan 4 buah tempat penampungan berbeda alamat yang disebut bank dan mempunyai 21 buah SFR (*Special Function Register*) yang terletak pada alamat 80H – FFH, semua register terletak pada internal RAM.

2.2.2.1. Register R₀ – R₇

Setelah reset *default register* terletak pada bank 0 yaitu alamat 00H – 07H. Register ini mempunyai 4 bank yaitu bank 0 – bank 3 yang masing-masing bank mempunyai ruang 8 byte. Jadi alamat bank untuk register ini 00H – 1FH. Register R0 dan R1 adalah register istimewa karena hanya register ini saja yang dapat dioperasikan dalam berbagai macam mode pengalamatan.

2.2.2.2. Register Fungsi Khusus (SFR)

Register internal AT89C51 tersusun sebagai bagian dari RAM internal mikrokontroler. *Special Function Register* (SFR) berjumlah 21 yang terletak pada bagian atas RAM internal, yaitu yang beralamat 80H – FFH. Gambar 2.3. menampilkan secara jelas peta pembagian alamat memori pada SFR.

Byte address	Bit address								
FF									
F0	F7	F6	F5	F4	F3	F2	F1	F0	B
E0	E7	E6	E5	E4	E3	E2	E1	E0	ACC
D0	D7	D6	D5	D4	D3	D2	-	D0	PSW
B8	-	-	-	BC	BB	BA	B9	B8	IP
B0	B7	B6	B5	B4	B3	B2	B1	B0	P3
A8	AF	-	-	AC	AB	AA	A9	A8	IE
A0	A7	A6	A5	A4	A3	A2	A1	A0	P2
99	not bit addressable								SBUF
98	9F	9E	9D	9C	9B	9A	99	98	SCON
90	97	96	95	94	93	92	91	90	P1
8D	not bit addressable								TH1
8C	not bit addressable								TH0
8B	not bit addressable								TL1
8A	not bit addressable								TL0
89	not bit addressable								TMOD
88	8F	8E	8D	8C	8B	8A	89	88	TCON
87	not bit addressable								PCON
83	not bit addressable								DPH
82	not bit addressable								DPL
81	not bit addressable								SP
80	87	86	85	84	83	82	81	80	P0
SPECIAL FUNCTION REGISTER									

Gambar 2.3. Peta Memori pada SFR AT89S52

Beberapa fungsi dari SFR yaitu :

- *Accumulator (Acc)* :
Register ini merupakan register serbaguna dan umumnya digunakan untuk operasi matematika dan logika. Register ini memiliki panjang 8 bit dan terletak pada alamat E0H.
- *Register B* :
Register ini terletak pada alamat F0H memiliki panjang 8 bit dan biasanya digunakan bersama dengan *accumulator* perkalian (Mul) dan pembagian (Div). Pada operasi perkalian , hasil perkalian *low byte* diletakkan pada *accumulator* sedang *high byte*

pada register B dan hasil pembagian ada di Acc, sedang sisanya pada register B.

- **Stack Pointer (SP) :**

Stack Pointer merupakan register 8 bit yang terdapat pada alamat 81H. Saat operasi *push*, data ke *stack* akan ditambah sebelum data ditulis, saat operasi *pop* data dari *stack* diambil setelah itu *stack* dikurangi. Biasanya pemakai menginisialisasi ulang *stack* tetapi jika tidak diinisialisasi ulang maka *default* dari *stack* adalah \$07. Ini berarti *bank 1 – bank 3* digunakan oleh *stack* sehingga register R0 – R7 harus diletakkan pada bank 0 sesuai dengan *default*nya.

- **Data Pointer (DPTR)**

Data pointer merupakan register 16 bit yang dibagi menjadi DPL alamat 82 H untuk byte rendah (*low-byte*) dan DPH alamat 83 H untuk byte tinggi (*high-byte*). Fungsi utama dari DPTR digunakan untuk mengakses eksternal data/kode memori dan juga menampung data 2 byte.

- **Port Register**

AT89S52 memiliki register port 0 pada alamat 80H, register port 1 pada alamat 90H, register port 2 alamat A0H, register port 3 pada alamat B0H.

- **Timer Register**

AT89S52 memiliki 3 buah timer 16 bit. Register timer 0 berada pada alamat 8AH untuk TL0 (*low byte*) dan alamat 8CH untuk TH0 (*high byte*). Register timer 1 terletak pada alamat 8BH untuk TL1 dan alamat 80H untuk TH1. Operasi dari timer ditentukan oleh register TMOD (*Timer/counter Mode Control*), dengan alamat 89H untuk menekan mode operasi dari register TCON (*Timer/Counter Control*), dengan alamat 88H sebagai kontrol operasi.

- **Serial Port Register**

AT89S52 memiliki kaki untuk komunikasi serial oleh sebab itu diperlukan sebuah register untuk menampung data komunikasi. Register tersebut adalah Serial Data Buffer (SBUF) pada alamat 99H. Register ini menampung data untuk mengirim ataupun data yang diterima. Berbagai macam mode operasi untuk serial komunikasi terletak pada *Serial Port Control Register (SCON)* pada alamat 98H.

- *Interrupt Register*

AT89S52 memiliki 6 sumber interupsi (serial port interupsi, timer 0, timer 1 dan timer 2 interupsi, eksternal interupsi 0 dan 1). Interupsi tidak aktif setelah sistem reset, maka interupsi harus diaktifkan dengan software pada *Interrupt Enable Register* (IE) alamat A8H. Prioritas interupsi dapat diatur pada *Interrupt Priority Register* (IP) B8H.

- *Power Control Register* (PCON) :

Didalam register PCON terdapat kontrol bit untuk catu daya IC sehingga dapat diatur pemakaian dayanya. Didalam register terdapat kontrol bit SMOD untuk menggandakan *baud rate* serial port mode 1,2,3 dan kontrol bit untuk *power down* (PD) juga untuk *idle mode* (IDL).

- *Program Status Word* (PSW).

Terletak pada alamat D0H dan merupakan bit status yang menunjukkan keadaan CPU pada saat itu.

Status terdiri atas :

- *Carry Flag* (CY)

Bit ini akan bernilai satu apabila bit 8 terlampaui saat dilakukan operasi penjumlahan/nilai pengurang lebih besar saat operasi pengurangan.

- *Auxiliary Carry Flag* (AC)

Bit ini akan bernilai 1 jika nilai suatu data dialuar range \$00 - \$09 saat dilakukan operasi penjumlahan bilangan B BCD dan diikuti dengan perintah DAA maka data yang keluar *range* dapat dikembalikan dalam *range*.

- *Flag 0*

Bit ini disediakan untuk status bit pemakai untuk berbagai keperluan. *Register bank Select bits* : 2 bit ini digunakan untuk memilih *bank* mana yang akan dipakai oleh register R0 –R7 dengan memilih kombinasi dari RS0 dan RS1. Jumlah kombinasinya sesuai dengan jumlah *bank* yang ada. *Default* dari RS0 dan RS1 adalah *bank 0*.

- *Over Flow Flag*

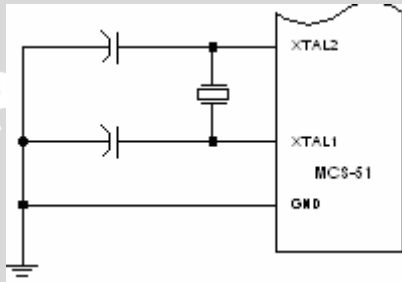
Bit akan bernilai satu jika terjadi nilai *overflow* dari operasi penjumlahan ataupun pengurangan.

- *Parity Bit*

Bit ini secara otomatis bernilai satu atau nol untuk menentukan nilai paritas.

2.2.3. Pewaktuan

Mikrokontroler AT89S52 memiliki rangkaian osilator internal yang mengacu pada frekuensi berupa kristal dan kapasitor yang dihubungkan ke ground. Penggunaan kristal sebagai referensi dapat memberikan ketelitian yang baik. Frekuensi kerja osilator pada mikrokontroler AT89S52 adalah 3 MHz sampai 24 MHz.



Gambar 2.4. Pewaktuan melalui Osilator Internal

2.2.4. Interupsi

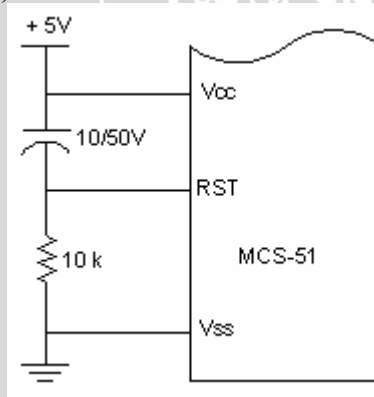
Saat CPU pada mikrokontroler AT89S52 sedang mengadakan suatu program kita dapat menghentikan pelaksanaan program tersebut secara sementara dengan meminta interupsi. Bila CPU mendapat permintaan interupsi program counter (PC) akan diisi dengan alamat dari vektor interupsi. CPU kemudian melaksanakan rutin pelayanan interupsi mulai dari alamat tersebut. Bila rutin pelayanan interupsi selesai dilaksanakan, CPU AT89S52 kembali melaksanakan program utama yang ditinggalkan.

Interupsi pada AT89S52 dibangkitkan dari tujuh sumber. Dua sumber dibangkitkan secara eksternal melalui pin INT0 dan INT1. Logika 0 pada pin INT0 akan mengeset bit IE0 pada register TCON. Logika 0 pada pin INT1 akan mengeset bit IE1 pada register TCON. Bit IE0 dan IE1 akan menghasilkan interupsi pada AT89S52. Jadi ada dua cara pengaktifan interupsi eksternal ini. Cara pertama adalah *Transition-activated Interrupt*. Pada teknik ini interupsi akan terjadi pada saat AT89S52 mendeteksi transisi logika 1 ke 0 pada pin INT0 atau INT1. Cara kedua adalah *level-activated interrupt*. Terjadi saat ada logika 0 pada pin INT0 dan INT1. Interupsi terjadi karena pin INT0 akan menyebabkan alamat program lompat ke alamat 03H, sedangkan INT1 alamat 13H.

Mikrokontroler AT89S52 mempunyai dua sumber interupsi lainnya yaitu *Timer 0* dan *Timer 1*. Saat timer *overflow*, bit-bit TF0, TF1 dan TF2 pada TCON akan diset. Pengesetan bit-bit ini akan menyebabkan terjadinya *timer interrupt*. Nilai 1 pada TF0 akan menyebabkan mikrokontroler lompat ke alamat 0BH, sedangkan TF1 akan menyebabkan alamat program melompat ke alamat 23H. Tiap-tiap sumber interupsi dapat dienable atau didisable secara *software*.

2.2.5. Reset

Rangkaian *Power-On Reset* diperlukan untuk mereset mikrokontroler secara otomatis setiap catu daya dinyalakan. Ketika catu daya diaktifkan, rangkaian reset akan menahan logika tinggi pada pin RST dengan jangka waktu yang ditentukan oleh lama pengosongan muatan kapasitor. Jangka waktu minimal adalah 2 siklus mesin (24 periode frekuensi clock) ditambah waktu *start on osilator*. Gambar 2.5 menunjukkan rangkaian *Power-On Reset*. (Anonim,2003).



Gambar 2.5. Rangkaian Power-on Reset

2.3 Dual Tone Mutiple Frequency

Sistem DTMF merupakan salah satu sistem modem. Sistem ini mengubah suatu informasi digital pada karakter 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, #, A, B, C, D yang akan dikirim melalui gelombang elektromagnetik menjadi frekuensi tertentu atau sebaliknya. Setiap satu frekuensi yang dipancarkan mewakili 4 bit kombinasi biner

karakter masukan yang diberikan. Kombinasi biner tersebut berbeda-beda untuk setiap karakter. DTMF menggunakan frekuensi antara 697 – 1633 Hz yang terdiri dari kelompok frekuensi rendah dan frekuensi tinggi. Frekuensi rendah dibagi menjadi 4 dan begitu pula frekuensi tinggi. Setiap karakter akan diwakili oleh satu frekuensi rendah dan satu frekuensi tinggi (Widiarti,2004).

Adapun alokasi frekuensi DTMF lebih lengkapnya dapat dilihat pada tabel di bawah ini:

Tabel 2.2. Alokasi Frekuensi DTMF

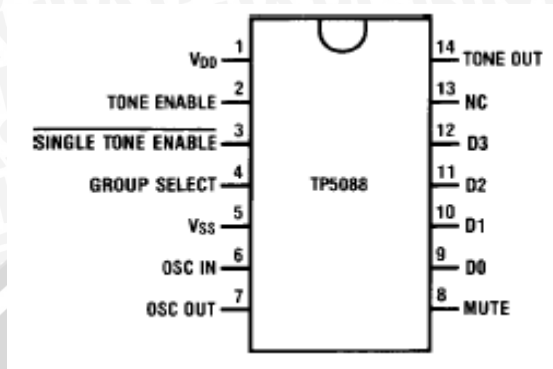
Frekuensi	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

(Sumber: <http://en.wikipedia.org/wiki/DTMF>)

2.3.1 Encoder Dual Tone Multiple Frequency

Dalam rangkaian terintegrasi *tone encoder* banyak tersedia. Salah satunya adalah TP 5088N yang merupakan IC CMOS untuk menghasilkan nada pilih yang digunakan dalam sistem telepon DTMF.

TP 5088N menghasilkan satu sinyal *dual-tone* dengan mengaktifkan frekuensi pada salah satu kolom dan salah satu baris. *Tone-out* berimpedansi tinggi ketika satu dari beberapa kolom diaaktifkan dan berlogika tinggi saat semua masukan kolom tidak diaktifkan.



Gambar 2.6. Konfigurasi Pin TP5088

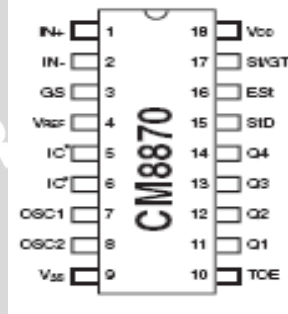
(Sumber: http://www.datasheetcatalog.com/datasheets_pdf/T/P/5/0/T/P5088N.shtml)

Adapun fungsi dari masing-masing pin dapat dijelaskan sebagai berikut:

- VDD merupakan pin untuk catu daya positif
- Tone Enable merupakan pin yang berfungsi untuk pengambilan data. Pada saat tone enable diset tinggi, maka data dapat diambil dan pada saat tone enable diset rendah, data mengambang dan tidak dapat diambil
- Vss merupakan pin untuk catu daya negatif.
- D0-D4 merupakan saluran data yang membentuk kombinasi 4 bit.
- Tone out merupakan pin untuk keluaran dari DTMF
- Single Tone Enable merupakan pin yang berfungsi untuk pembuatan tone keluaran dalam bentuk satu (hanya frekuensi tinggi saja atau frekuensi rendah saja).
- Group Select merupakan pin yang berfungsi untuk pemilihan frekuensi keluaran pin *tone enable* pada saat pin *single tone enable* di hubungkan dengan Vss. Pada saat pin ini dibiarkan terbuka, maka pin *tone out* akan mengeluarkan frekuensi tinggi. Pada saat dengan Vss, maka pin *tone out* akan mengeluarkan frekuensi rendah. (Anonim, 1991)

2.3.2 Decoder Dual Tone Multiple Frequency

Dekoder DTMF merupakan rangkaian yang dapat merespon kode-kode sinyal DTMF ke dalam bentuk biner. Di pasaran telah tersedia dalam bentuk terintegrasi yaitu MT 8870. Adapun gambar konfigurasi pin MT 8870 dapat dilihat pada gambar di bawah ini.



Gambar 2.7. Konfigurasi pin MT 8870

(Sumber: http://www.datasheetcatalog.com/datasheets_pdf/C/M/8/8/CM8870.shtml)

Adapun fungsi dari masing-masing pin dijelaskan sebagai berikut:

- IN + dan IN – merupakan pin masukan. IN + merupakan *non inverting input op-amp* dan IN – merupakan pin *inverting input op-amp*
- Gs merupakan pin yang berfungsi untuk mengatur besarnya penguatan untuk sinyal dari luar.
- Vref merupakan pin untuk tegangan teferensi dan biasanya besarnya setengah dari VDD.
- INH merupakan pin yang berfungsi untuk mengeluarkan data DTMF A,B,C,D
- OSC 1 dan OSC 2 dihubungkan dengan kristal 3,57945 MHz. OSC 1 merupakan *clock input* dan OSC 2 merupakan *clock output*.
- VSS dan VDD merupakan pin catu daya. VSS dihubungkan dengan *ground* sedangkan VDD dihubungkan dengan catu daya positif.

- St/Gt untuk menentukan kecepatan nada yang dapat diterjemahkan
- Est merupakan pin yang akan berlogika tinggi saat menerima nilai masukan yang benar.
- Std merupakan pin yang berlogika tinggi ketika menerima pasangan nada yang dikenali
- D0-D3 merupakan nilai keluaran.(Anonim, 1997)

2.4 Radio Pemancar Penerima (RPP) FM 144 MHz

Radio pemancar penerima adalah gabungan dari pemancar dan penerima yang berfungsi sebagai piranti komunikasi. Pemancar berfungsi memodulasi sinyal *Carrier* berdasarkan sinyal masukan dan kemudian mengirimkannya. Penerima berfungsi untuk memodulasi sinyal yang diterima sehingga diperoleh kembali sinyal informasi.

Radio pemancar penerima (RPP) yang banyak digunakan adalah RPP 144 MHz. Secara umum piranti ini memiliki karakteristik sebagai berikut:

- Metode komunikasi yang digunakan adalah *half-duplex*.
- Menggunakan *microphone* sebagai transduser sinyal audio ke tegangan dan menggunakan *speaker* sebagai transduser tegangan ke audio.
- Sinyal masukan yang diterima mempunyai frekuensi antara 300 Hz sampai dengan 3400 Hz. (Widiarti, 2004).

2.5 Liquid Crystal Display (LCD)

Liquid Crystal Display atau LCD merupakan sebuah indikator yang berupa tampilan layar. LCD terdiri atas tumpukan tipis atau sel dari dua lembar kaca dengan pinggiran tertutup rapat. Antara dua lembar tersebut di beri bahan kristal cair (*liquid crystal*) yang tembus cahaya. Permukaan masing-masing keping kaca mempunyai lapisan tembus cahaya seperti oksida timah (*tin oxide*) dan oksida indium (*indium oxide*). Sel mempunyai ketebalan 1×10^{-6} meter dan diisi dengan kristal cair.

M1632 merupakan modul LCD jenis dot matrik yang hanya memerlukan tegangan rendah untuk suplay tegangan, dan telah memiliki pengontrol (driver) CMOS di dalamnya. Pengontrol tersebut terdiri dari Character Generator (CG) ROM/ RAM, dan display data (DD) RAM. Semua fungsi dari display ini dikontrol oleh instruksi-instruksi dan modul ini dapat dengan mudah diinterfacekan dengan Micro Processor Unit (MPU).

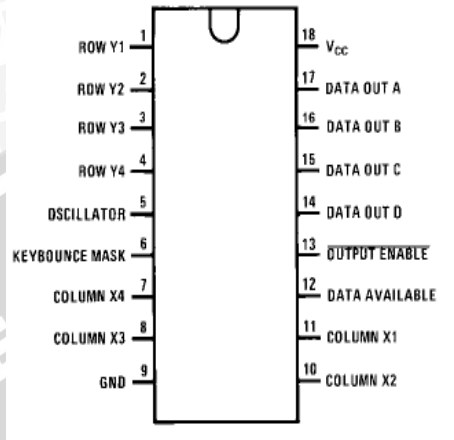
Adapun spesifikasi dari LCD ini adalah terdiri dari 16 karakter, CG ROM dengan 192 tipe karakter, CG RAM untuk 8 tipe karakter (program write), DD RAM berukuran 80x8 bit, rangkaian reset otomatis, dapat di interfacekan dengan MPU dalam 4 atau 8 bit, DD RAM dan CG RAM rangkaian osilator dalam pin yaitu 8 pin untuk data, 3 pin untuk control, 3 pin suplay tegangan dan 2 pin untuk lampu. (Anonim, 1987).



Gambar 2.8 Konfigurasi pin LCD

2.6 Keypad dan *Encoder* MM 74C922

Keypad merupakan papan tombol yang dapat dijadikan data masukan berupa angka-angka atau beberapa tanda tertentu. Masukan dari keypad akan dikodekan menjadi bilangan biner sehingga dibutuhkan encoder untuk mengkodekan semua nilai yang ada pada keypad. Encoder ini akan mengkonversikan sebuah switch tombol yang ditekan menjadi 4 bit data dalam bentuk biner



Gambar 2.9 Konfigurasi pin MM74C922

(Sumber: http://www.datasheetcatalog.com/datasheets_pdf/M/M/7/4/MM74C922.shtml)

Adapun fungsi dari pin-pin di atas adalah sebagai berikut:

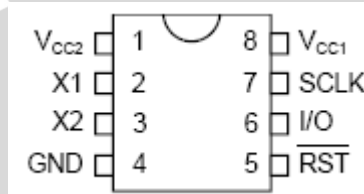
- Vcc merupakan masukan catu daya positif
- DA-DD merupakan keluaran data biner 4 bit
- OE merupakan keluaran 3 keadaan untuk kemudahan perluasan bus operasi
- DA merupakan pin yang berfungsi untuk mendeteksi penekanan tombol serta menunjukkan telah diterimanya data masukan.
- Row Y1 – Row Y4 merupakan jalur masukan baris dari keypad
- Col X1 – Col X4 merupakan jalur masukan kolom
- OSc merupakan masukan osilator internal
- *Keybounce Mask* (KBM) merupakan pin yang berfungsi untuk meredam *bouncing effect*.
- GND merupakan masukan *ground* (Anonim, 1999)

2.7 Real Time Clock (RTC) DS1302

DS1302 adalah *Serial RTC* dengan fitur-fitur sebagai berikut:

- Menghitung waktu mulai detik, menit, jam, tanggal, bulan, tahun, hari dalam minggu dengan kompensasi tahun kabisat sampai tahun 2100.
- Memory / RAM sebesar 31 byte
- Single byte atau burst akses
- Support battery Lithium atau Ni-Cd untuk backup supply
- Kemampuan Trickle Charge untuk pengisian battery jenis Ni-Cd.

Adapun konfigurasi pin dari RTC DS 1302 dapat dilihat pada gambar di bawah ini:



Gambar 2.10 Konfigurasi pin RTC DS1302

(Sumber: <http://pdfserv.maxim-ic.com/en/ds/DS1302.pdf>)

Adapun fungsi dari pin DS1302 adalah sebagai berikut:

- VCC2 merupakan masukan positif yang dihubungkan dengan catu daya utama.
- VCC1 merupakan masukan positif cadangan dari catu daya utama yang dihubungkan dengan baterai.
- GND merupakan pin yang dihubungkan ke *ground*
- X1 – X2 merupakan pin yang dihubungkan dengan kristal 32,768 KHz
- RST merupakan pin aktif rendah yang digunakan untuk mereset DS1302
- SCLK merupakan pin yang berfungsi untuk menyinkronkan data pada komunikasi serial
- I/O merupakan pin *bi-directional* yang digunakan untuk mengirim keluaran atau menerima masukan. (Anonim, 2005)

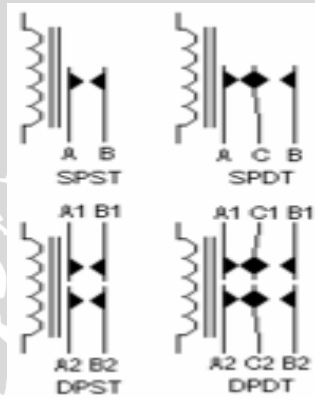
2.8 Relay

Relay berfungsi sebagai saklar otomatis. Aktif dan tidaknya saklar relay ditentukan oleh ada dan tidaknya arus yang mengalir didalam kumparan. Pada dasarnya relay terdiri atas kontak dan

kumparan dengan inti besi di dalamnya. Relay mempunyai dua keadaan yaitu normal tertutup (NC) dan normal terbuka (NO). Ada beberapa jenis *relay* antara lain:

- Relay SPST (*Single Pole Single Throw*)
- Relay SPDT (*Single Pole Double Throw*)
- Relay DPST (*Double Pole Single Throw*)
- Relay DPDT (*Double Pole Double Throw*)

Adapun simbol-simbol relay dapat dilihat pada gambar di bawah ini.



Gambar 2.11 Jenis-jenis Relay

(Sumber: http://en.wikipedia.org/wiki/Relay#Types_of_relay)

BAB III

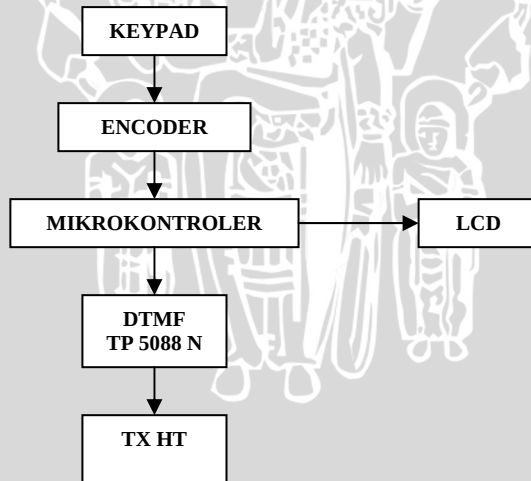
PERENCANAAN DAN PEMBUATAN ALAT

Perancangan dan pembuatan alat dilakukan di Laboratorium Elektronika Jurusan Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya yang dimulai pada bulan Juli 2004 sampai dengan Maret 2007.

Perancangan pada sistem ini terdiri dari dua bagian yaitu perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras meliputi perancangan rangkaian pengirim dan rangkaian penerima. Perancangan perangkat lunak meliputi perancangan pada mikrokontroler pengirim dan modul penerima.

3.1 Perencanaan Perangkat Keras

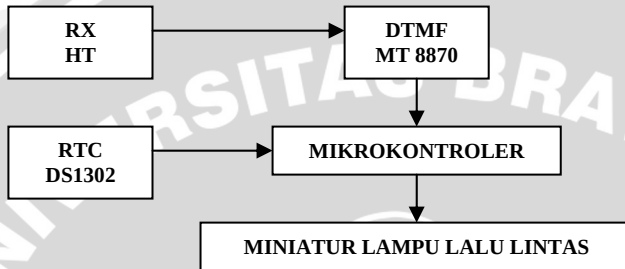
Prinsip kerja dari sistem pengirim adalah apabila operator ingin merubah waktu hijau pada salah satu jalur *traffic light* diatur dengan menginstruksikannya melalui *keypad*. Instruksi ini akan dipancarkan melalui *transmitter handy talkie* dalam bentuk sinyal analog dan setiap proses yang sedang terjadi ditampilkan di LCD.



Gambar 3.1. Diagram Blok Pengirim

Prinsip kerja dari sistem penerima adalah menerima sinyal analog dari sistem pengirim untuk kemudian dirubah menjadi 4 bit

bilangan biner. Di dalam mikrokontroler instruksi tersebut diolah dan digunakan untuk pengendalian waktu lampu hijau dari salah satu persimpangan yang dipilih oleh operator. Adapun gambar diagram blok penerima dapat dilihat pada gambar di bawah ini:



Gambar 3.2. Diagram Blok Penerima

3.1.1 Rangkaian Pengirim

Rangkaian pengirim terdiri dari DTMF TP 5088N, Keypad, Encoder MM 74C922, LCD Seiko M1362 dan Mikrokontroler AT89S52.

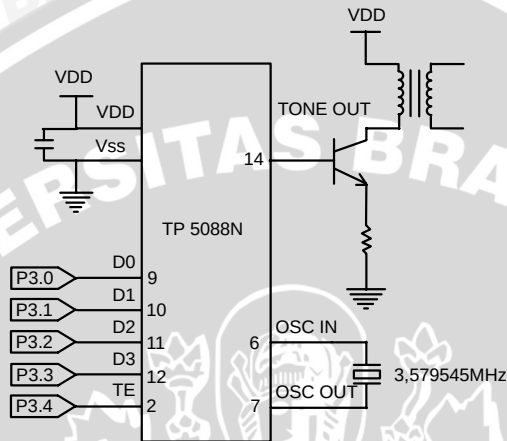
3.1.1.1 *Dual Tone Multiple Frequency* TP 5088N

Rangkaian *transmitter* berfungsi untuk memancarkan data ke sebuah *receiver*. Rangkaian ini menggunakan DTMF TP 5088N dengan beberapa rangkaian pelengkap yang berfungsi memancarkan dan mengubah sinyal digital menjadi sinyal analog.

TP 5088N menghasilkan satu sinyal *dual-tone* dengan cara mengirim logika tinggi ke pin *tone enable* disertai dengan empat bit yang mau dikirim ke D0-D3. TP 5088N akan secara otomatis merubah nilai D0-D3 menjadi sebuah sinyal dual tone berdasarkan nilai empat bit ini.

Agar TP 5088N ini dapat bekerja, maka dibutuhkan komponen tambahan. Komponen tambahan berupa sebuah kristal 3,579545 MHz yang dihubungkan ke OSC1 dan OSC 2. Transformer yang digunakan untuk output merupakan trafo yang digunakan untuk

coupling data DTMF ke *Handie Talky*. Adapun resistor yang digunakan merupakan pengatur besarnya amplitude. Keluaran dari transformer ini akan dihubungkan dengan bagian *microphone* dari RPP 144 MHz.



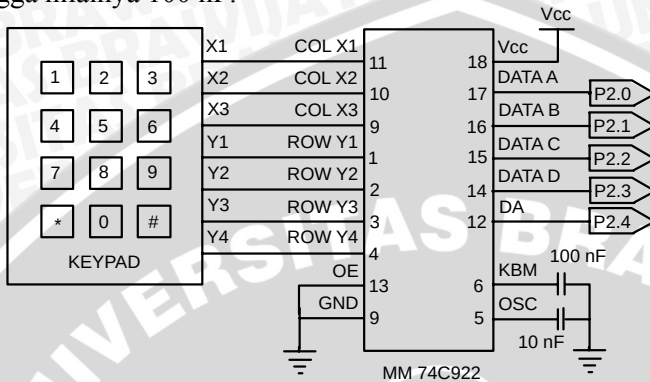
Gambar 3.3 Perencanaan rangkaian TP 5088N

3.1.1.2 Keypad dan *Encoder* MM 74C922

Dalam penelitian ini digunakan keypad 4 X 4. Untuk mengkodekan nilai yang ada pada keypad, dibutuhkan encoder MM 74C922 sehingga bisa menghemat port pada mikrokontroler. Encoder MM 74C922 ini akan mengkonversikan data dari keypad menjadi empat bit data dalam bentuk biner.

Encoder MM 74C922 dihubungkan dengan empat baris dari keypad (Y1-Y4) dan 4 kolom dari keypad (X1-X4). Agar Encoder MM 74C922 ini dapat bekerja maka dibutuhkan komponen tambahan yang lain. Komponen eksternal yang dibutuhkan pada Encoder MM 74C922 adalah dua buah kapasitor yang dihubungkan dengan pin 5 dan pin 6. Kapasitor pada pin lima diperlukan untuk melengkapi rangkaian *osilator internal* IC Encoder MM 74C922. Kapasitor pada pin 6 diperlukan untuk meredam getaran mekanik akibat penekanan tombol pada keypad yang dikenal sebagai gejala *bounce*. Nilai kapasitor yang dipasang pada pin lima disesuaikan dengan *datasheet* sebesar 10nF. Untuk kapasitor pin 6 (peredam

bouncing), nilainya sebesar 10 X C1 (kapasitor pada pin lima) sehingga nilainya 100 nF.



Gambar 3.4 Perencanaan rangkaian keypad dengan encoder MM 74C922

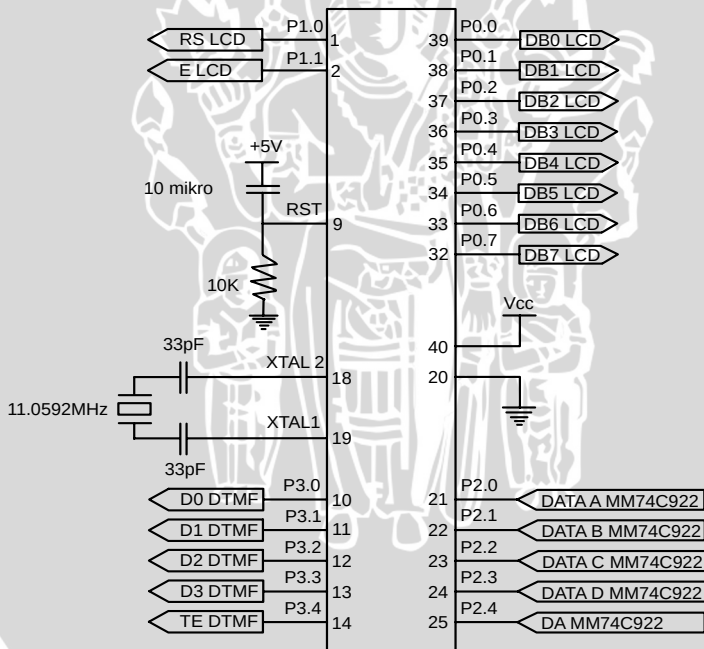
Pada saat Encoder MM 74C922 mendeteksi adanya penekanan tombol pada keypad, maka pin DA (*Data Available*) akan menjadi tinggi. Keluaran DA yang tinggi merupakan tanda bahwa Encoder MM 74C922 telah menerima data masukkan. Pin DA ini akan dihubungkan dengan port 2.4 pada mikrokontroler pemancar untuk memudahkan pengendalian perangkat lunaknya. Untuk bilangan biner hasil pengkodean yang dilakukan Encoder MM 74C922, dihubungkan dengan port 2.0 sampai dengan port 2.3.

3.1.1.3 Mikrokontroler AT89S52

Dalam perancangan alat ini, Mikrokontroler merupakan pengendali dari seluruh kerja sistem. Mikrokontroler dihubungkan dengan LCD, Encoder MM 74C922, dan DTMF TP 5088N. Adapun pemanfaatan port pada mikrokontroler ini adalah sebagai berikut:

- Port 0
Delapan port pada port 0 (p0.0-p0.7) digunakan sebagai jalur data delapan bit ke LCD. Jadi P0.0 dihubungkan dengan DB0 pada LCD dan seterusnya.
- Port 1
Port 1 digunakan untuk sinyal kendali pada LCD. Port 1.0 dihubungkan dengan *Register Select (RS)* pada LCD. Port 1.1 dihubungkan dengan pin *Enable (E)* pada LCD.

- Port 2
Port 2 digunakan untuk mengendalikan masukan dari *Encoder* MM 74C922. Secara detailnya, port 2.0-port 2.3 digunakan sebagai masukan data biner dari *Encoder* MM 74C922 ke mikrokontroler. Sedangkan port 2.4 dihubungkan dengan DA dari *Encoder* MM 74C922.
- Port 3
Secara umum, port 3 digunakan untuk mengendalikan sinyal yang dikirim dari mikrokontroler ke DTMF TP 5088N. Port 3.0-port 3.3 digunakan untuk membentuk bilangan biner yang akan dikirim ke DTMF. Sedangkan port 3.4 merupakan port yang digunakan untuk mengendalikan *tone enable* pada DTMF sehingga DTMF dapat menerima masukan dari mikrokontroler.



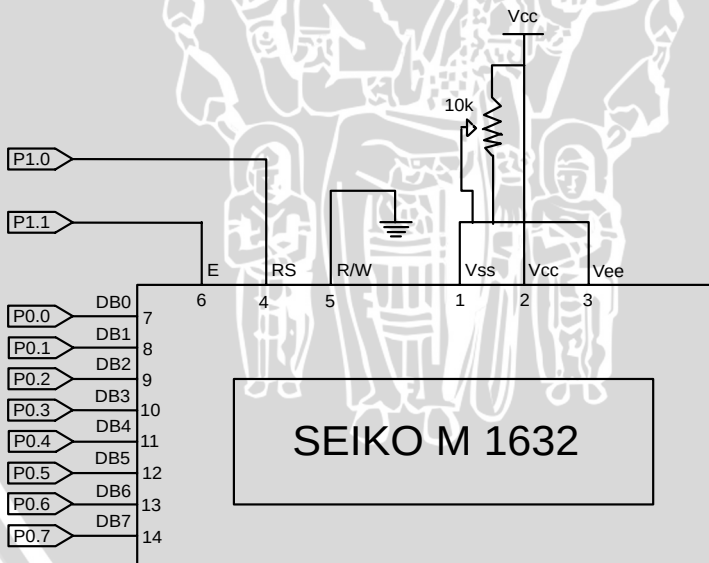
Gambar 3.5 Perencanaan penggunaan port pada AT89S52 sebagai modul pengirim

3.1.1.4 Liquid Crystal Display

LCD pada perancangan sistem ini digunakan untuk menampilkan kerja sistem. LCD merupakan petunjuk untuk menggunakan peralatan ini. LCD yang digunakan ialah tipe M 1632 buatan Seiko.

Adapun konfigurasi pin-pin pada LCD agar bisa bekerja adalah sebagai berikut:

- Pin 1 (Vss) dan pin 3 (Vee) dihubungkan dengan variable resistor 10 kilo ohm untuk mengatur kecerahan tampilan karakter.
- Pin 2 (Vcc) dihubungkan dengan catu daya.
- Pin 4 (RS) dihubungkan dengan P1.0 sebagai sinyal pemilih data berupa instruksi atau berupa data ASCII.
- Pin 5 (R/W) dihubungkan dengan ground karena dalam sistem ini, LCD hanya bekerja menerima data dari mikrokontroler.
- Pin 7-14 (D0-D7) dihubungkan dengan port 0 sebagai saluran data 8 bit.



Gambar 3.6 Perencanaan rangkaian LCD Seiko M 1632

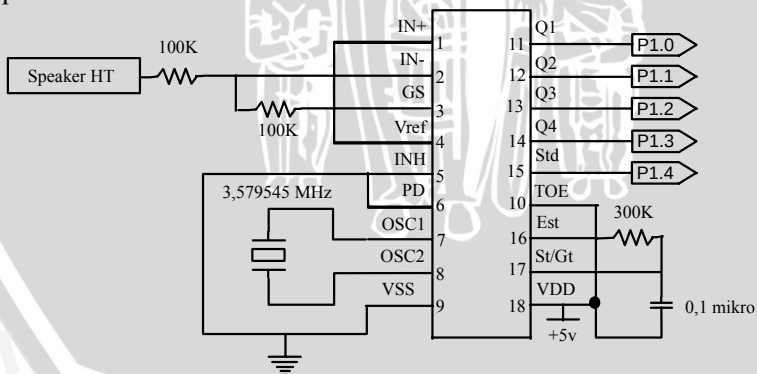
3.1.2 Rangkaian Penerima

Prinsip kerja dari rangkaian penerima adalah mengendalikan pewaktuan nyala lampu dan dapat menerima *interrupt* dari unit pemancar

3.1.2.1 Dual Tone Multiple Frequency MT 8870

DTMF MT 8870 berfungsi sebagai penerima sinyal analog dari unit pemancar untuk diubah menjadi bilangan digital. Sinyal *dual tone* yang masuk ke DTMF MT 8870 akan dirubah ke bilangan digital 4 bit sebelum di teruskan ke mikrokontroler pengendali.

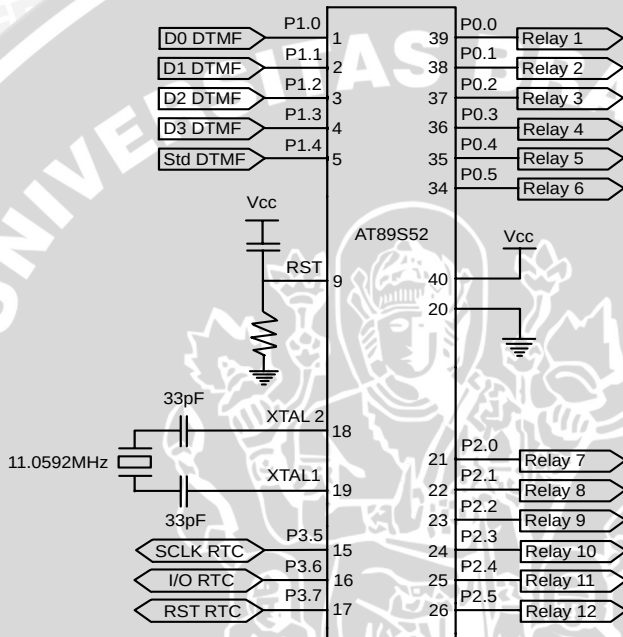
Pada rangkaian penerima, DTMF MT 8870 memiliki pin yang berfungsi sebagai pendeteksi adanya sinyal DTMF yang masuk. Pada saat ada sinyal DTMF yang masuk, DTMF MT 8870 akan mengeset Std pada logika tinggi. Keluaran dari Std inilah yang digunakan mikrokontroler sebagai pertanda adanya data yang dikirimkan transmitter. *Tristate Output Enable* (TOE) merupakan pin yang berfungsi sebagai pengatur data pada D0 sampai D3. Pada saat TOE berlogika '0', rangkaian output D0-D3 akan mengandung impedansi yang tinggi sehingga data tidak bisa diambil. Pada saat kaki TOE berlogika '1', Data pada output dapat diambil. Dalam perancangan, TOE dihubungkan dengan VDD sehingga data pada output dapat diambil terus tanpa perlu menunggu adanya set aktif pada TOE.



Gambar 3.7 Perencanaan rangkaian CM8870

3.1.2.2 Mikrokontroler AT89S52

Pada rangkaian penerima, Mikrokontroler dihubungkan dengan RTC DS1302, DTMF MT 8870 dan miniatur dari lampu lalu lintas. Fungsi utama dari mikrokontroler ini adalah mengatur mengendalikan miniatur lalu lintas berdasarkan instruksi perangkat lunak.



Gambar 3.8 Perencanaan penggunaan port pada AT89S52 sebagai modul penerima

Adapun pemanfaatan port pada mikrokontroler ini adalah sebagai berikut:

- Port 0
 - Port 0 digunakan untuk mengendalikan 6 buah lampu. Adapun port yang digunakan adalah sebagai berikut:
 - Port 0.0 digunakan untuk mengendalikan lampu merah pada jalan 1
 - Port 0.1 digunakan untuk mengendalikan lampu kuning pada jalan 1

- Port 0.2 digunakan untuk mengendalikan lampu hijau pada jalan 1
- Port 0.3 digunakan untuk mengendalikan lampu merah pada jalan 2
- Port 0.4 digunakan untuk mengendalikan lampu kuning pada jalan 2
- Port 0.5 digunakan untuk mengendalikan lampu hijau pada jalan 2
- Port 1

Port 1 dihubungkan dengan DTMF MT 8870. Adapun pemanfaatan port yang digunakan sebagai berikut:

 - Port 1.0-Port 1.3 digunakan untuk membaca data biner keluaran dari DTMF MT 8870
 - Port 1.4 dihubungkan dengan pin Std DTMF MT 8870, yang berfungsi sebagai pendeteksi adanya sinyal DTMF yang perlu untuk segera diproses melalui mekanisme interupsi.
- Port 2

Port 2 digunakan untuk mengendalikan 6 buah lampu. Adapun port yang digunakan adalah sebagai berikut:

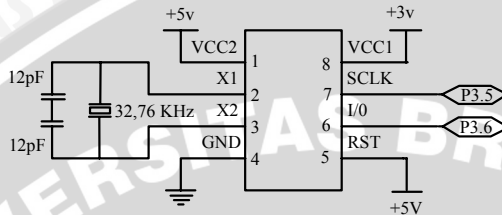
 - Port 2.0 digunakan untuk mengendalikan lampu merah pada jalan 3
 - Port 2.1 digunakan untuk mengendalikan lampu kuning pada jalan 3
 - Port 2.2 digunakan untuk mengendalikan lampu hijau pada jalan 3
 - Port 2.3 digunakan untuk mengendalikan lampu merah pada jalan 4
 - Port 2.4 digunakan untuk mengendalikan lampu kuning pada jalan 4
 - Port 2.5 digunakan untuk mengendalikan lampu hijau pada jalan 4
- Port 3

Port 3 dihubungkan dengan RTC DS 1302. Adapun port yang digunakan adalah sebagai berikut:

 - Port 3.5 dihubungkan dengan *Serial Clock* dari RTC DS1302
 - Port 3.6 dihubungkan dengan data dari RTC DS1302

3.1.2.3 Real Time Clock DS1302

RTC DS1302 berfungsi sebagai basis pewaktuan pada rangkaian penerima. Di mana fungsi paling penting dari RTC DS 1302 adalah mengatur lamanya waktu lampu menyala.



Gambar 3.9 Perencanaan rangkaian DS1302

Pin input dan output pada DS1302 merupakan satu pin dua arah. Sclk merupakan pin yang berfungsi untuk sinkronisasi komunikasi data serial. Pada Vcc 1 merupakan sumber tegangan utama (seperti sumber tegangan pada mikrokontroler), sedangkan pada Vcc 2 merupakan sumber tegangan cadangan agar waktu terus berjalan pada saat tegangan utama mati sehingga data di RTC tetap valid. Pada saat Vcc2 lebih kecil dari Vcc1, maka Vcc1 yang akan menjadi sumber tegangan.

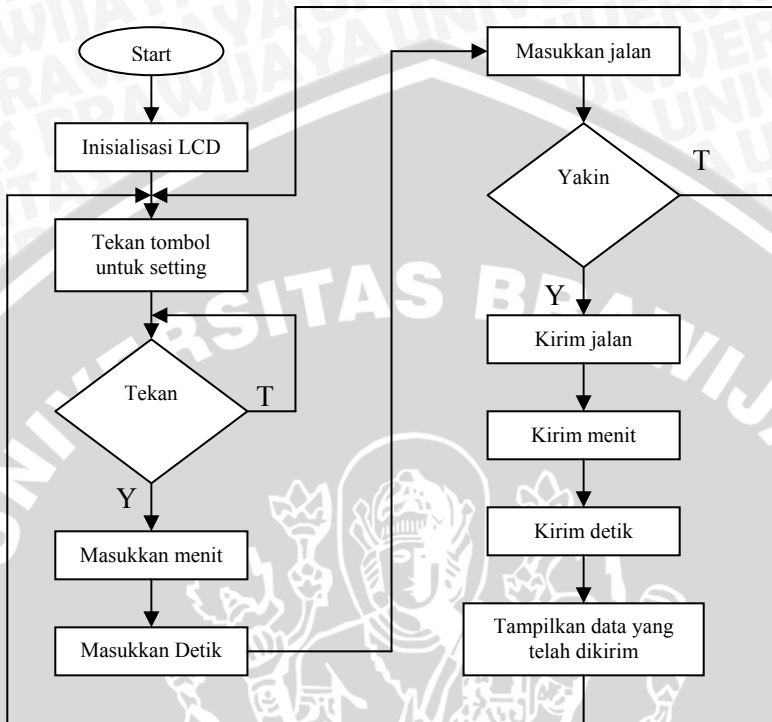
3.2 Perencanaan Perangkat Lunak

Pada mikrokontroler pengirim, perangkat lunak direncanakan untuk dapat mengendalikan pengiriman data pada DTMF dengan dibantu oleh LCD sebagai *user interface* dan keypad sebagai masukan data dari pemakai.

Pada mikrokontroler penerima, perangkat lunak diharapkan dapat mengendalikan urutan penyalan lampu, waktu menyalnya lampu dan menerima interupsi dari rangkaian pengirim.

3.2.1 Perangkat Lunak Mikrokontroler Pengirim

Sebagai pengirim data, mikrokontroler pengirim diharapkan dapat menampilkan proses yang terjadi di dalam mikrokontroler melalui LCD, menerima masukan dari keypad dan mengendalikan pengiriman data melalui DTMF TP5088N.



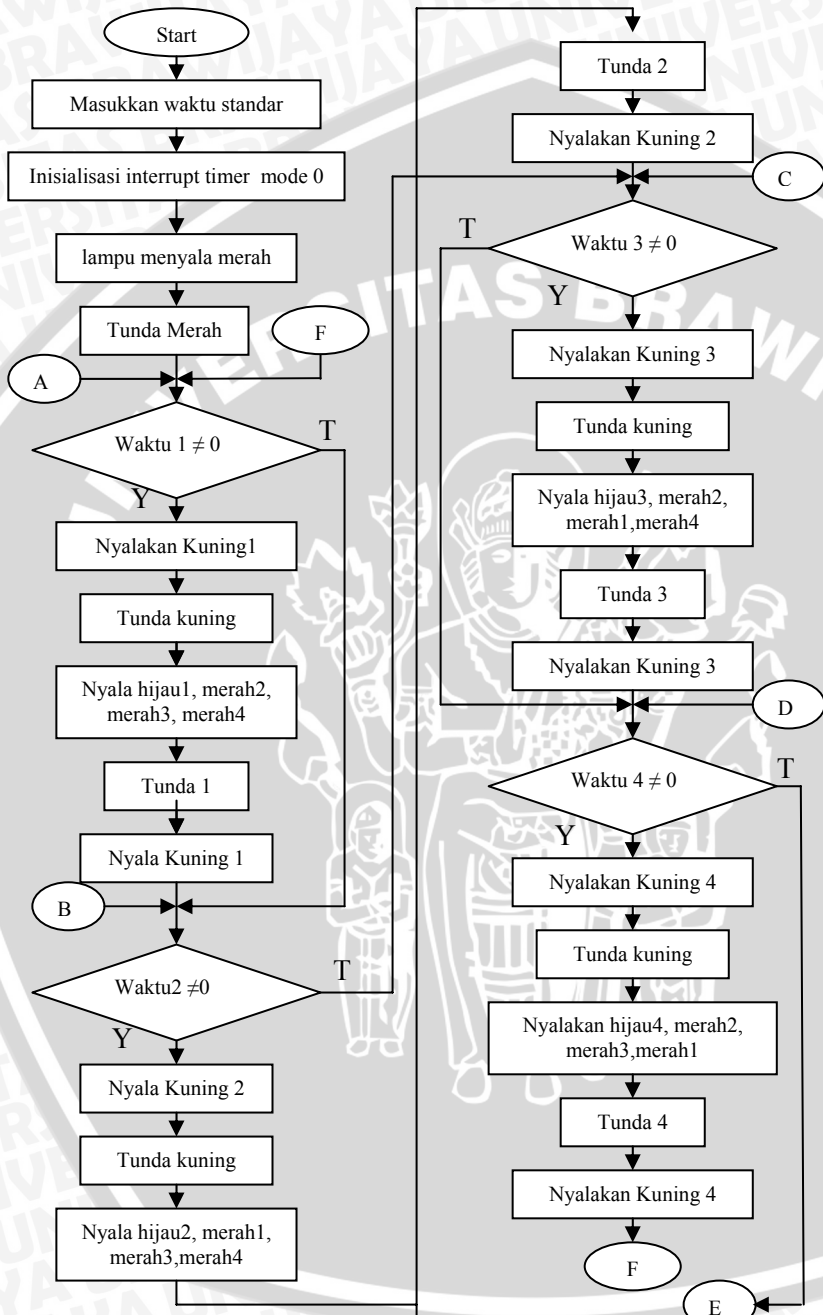
Gambar 3.10 Diagram alir prosedur pada mikrokontroler pengirim

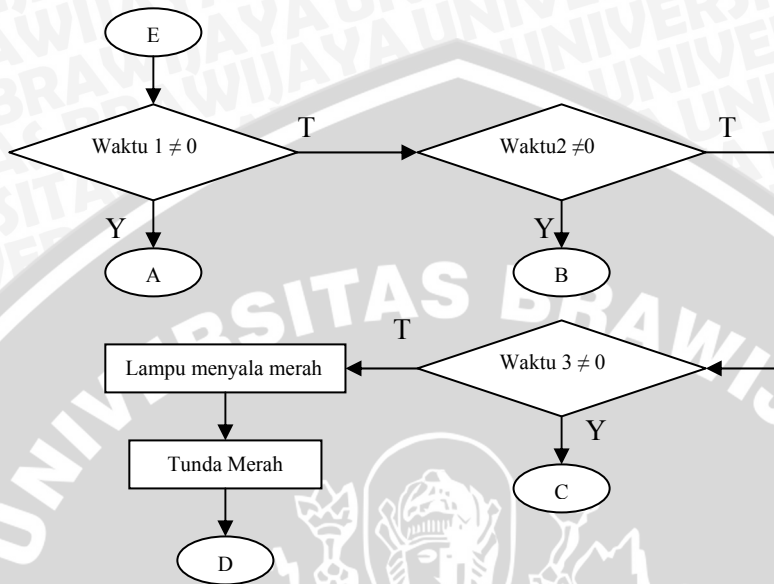
Pada saat pertama dinyalakan, Mikrokontroler akan menginisialisasi LCD. Setelah itu mikrokontroler akan menunggu masukan dari keypad. Setelah ada masukan, maka mikrokontroler akan memprosesnya dengan mengirimkan data tersebut. Proses pengiriman data dapat dilihat pada LCD. Setelah proses pengiriman selesai, mikrokontroler akan kembali menunggu masukan dari keypad. Proses ini akan berlangsung secara terus menerus.

3.2.2 Perangkat Lunak Mikrokontroler Penerima

3.2.2.1 Jalankan Miniatur

Perangkat lunak pada mikrokontroler penerima berfungsi untuk mengatur siklus penyalan lampu lalu lintas. Siklus penyalan lampu termasuk dengan penentuan waktu tunda untuk setiap lampu hijau yang dapat dibuat berbeda-beda.

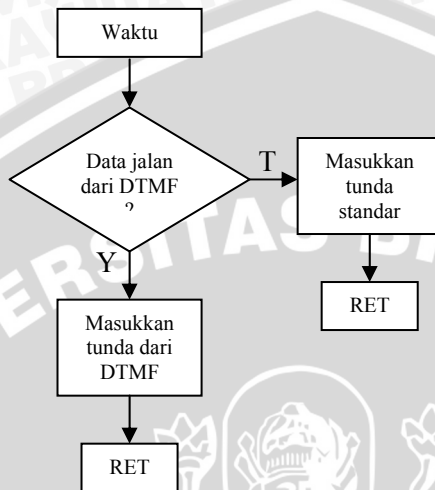




Gambar 3.11 Diagram alir jalankan miniatur

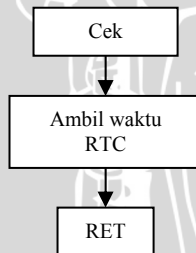
Proses yang terjadi pada minatur lampu lalu lintas adalah mengatur penyalan lampu hijau dari satu lampu ke lampu hijau yang lainnya. Proses ini berlangsung secara berurutan dari lampu hijau pertama ke lampu hijau kedua, dari lampu hijau kedua ke lampu hijau ketiga, dari lampu hijau ketiga ke lampu hijau keempat dan dari lampu hijau keempat ke lampu hijau pertama. Lamanya penyalan lampu hijau bergantung pada waktu tunda tiap lampu hijau yang dapat dibuat berbeda-beda.

Lamanya penyalan untuk lampu hijau diatur oleh sebuah sub rutin yang akan mengecek lamanya waktu tunda dari setiap lampu hijau. Yang dilakukan pertama kali oleh sub rutin ini adalah mengecek apakah lampu hijau pada jalan ini dirubah waktu tundanya oleh modul pengirim. Jika lampu hijau pada jalan yang sedang dicek tidak diubah waktu tundanya, maka digunakan waktu tunda standar. Pada saat lampu hijau yang dicek telah mengalami perubahan waktu tundanya, maka waktu tunda yang digunakan adalah waktu tunda dari modul pengirim yang terakhir untuk lampu tersebut. Adapun diagram alir dari sub rutin penentuan waktu tunda dapat dilihat pada gambar 3.12.



Gambar 3.12 Diagram alir penentuan waktu tunda

Untuk mengetahui jam pada RTC, digunakan sub rutin yang berfungsi untuk mengambil data dari RTC. Adapun sub rutin untuk mengambil data dari RTC adalah sebagai berikut:



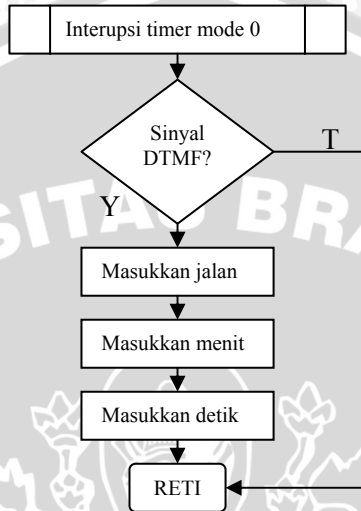
Gambar 3.13 Diagram alir rutin cek waktu

Adapun data yang diambil dari RTC adalah jam, menit dan detik.

3.2.2.2 Pembacaan Mode Masukan

Setelah *tone* diterima oleh DTMF penerima, maka data akan diproses untuk mengambil data lamanya waktu tunda dan jalan yang

diganti oleh operator. Dan mode ini berjalan pada vektor interupsi *timer mode 0* dari mikrokontroler.



Gambar 3.14 Diagram alir prosedur interupsi

Tanpa adanya sinyal DTMF, sebenarnya mikrokontroler akan tetap mengalami interupsi pada saat *timer 0 overflow*, tetapi karena sinyal DTMF yang dicek tidak ada, maka interupsi ini akan segera berakhir. Pada saat terjadi interupsi *timer mode 0* dan sinyal DTMF ada, maka interupsi ini akan memproses data-data dari modul penerima. Adapun data-data yang diproses secara berurutan adalah jalan yang diganti waktu tundanya, angka puluhan dari menit, angka satuan dari menit, angka puluhan dari detik dan yang terakhir angka satuan dari detik.

3.3 Pengujian Alat

Adapun untuk pengujian alat, tidak dilakukan pengujian bagian dari perangkat keras satu persatu. Pengujian dilakukan secara menyeluruh pada alat yang sudah ada apakah masih berfungsi atau tidak. Sedangkan untuk RTC dan Relay, dilakukan pengujian sendiri karena merupakan perangkat keras baru yang dipasangkan terhadap alat ini.

Dari pengujian alat yang sudah ada, maka didapatkan bahwa perangkat keras yang dipasangkan pada alat itu sudah bekerja dengan

baik. Jadi tidak dilakukan analisa per bagian karena alat ini terbukti masih berjalan sehingga membuktikan tidak ada kerusakan pada alat yang pernah ada. Adapun dari pengujian ini didapatkan bahwa waktu standar dari alat ini adalah 1 detik dengan alur nyala bergeser dari tiang 1, ke tiang 2, ke tiang 3, ke tiang 4, kembali ke tiang 1 dan seterusnya.

Untuk modul pengirim, juga didapatkan perangkat keras yang masih berfungsi dengan baik. Ini dibuktikan dengan dapat dikirimkannya data dari perangkat ini ke modul penerima. Adapun range pengiriman dari modul pengirim adalah dari 1 sampai 99. dan karena waktu standar pada modul penerima dalam satuan detik, alat ini hanya mampu merubah waktu tunda penyalan lampu dari 1 sampai 99 detik.

3.3.1 Pengujian Real Time Clock DS1302

Untuk pengujian Real Time Clock DS1302, maka digunakan program di mana RTC digunakan sebagai waktu tunda penyalan LED. Pada percobaan ini, Led dikedip-kedipkan selama 5 detik. Adapun program yang dimasukan dapat dilihat pada halaman lampiran.

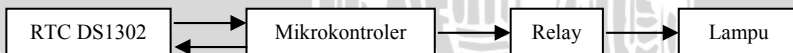
Adapun skema dari pengujian RTC DS1302 adalah sebagai berikut:



Gambar 3.15 Diagram blok pengujian RTC

3.3.2. Pengujian Relay

Untuk pengujian Relay, alat disusun sebagai berikut:



Gambar 3.16 Diagram blok pengujian Relay

Prinsip kerja dari skema di atas adalah mikrokontroler memberi logika tinggi dan logika rendah ke relay untuk mematikan dan menghidupkan lampu. Isi program dari mikrokontroler sama dengan isi program pada RTC DS1302 tetapi dengan tampilan keluaran berupa lampu yang digerakkan oleh relay yang diatur oleh mikrokontroler.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian

Pengujian dilakukan setelah semua bagian digabung menjadi satu. Tujuan pengujian adalah untuk memeriksa alat ini apakah telah bekerja sesuai dengan yang diharapkan. Pengujian dilakukan dengan mengganti waktu standar dengan menggunakan alat yang dibuat.

Untuk prosedur pengujiannya adalah dengan mengamati waktu nyala lampu pada tiap jalan. Lama nyala lampu dibandingkan dengan data yang dikirim oleh mikrokontroler pengirim. Pengoperasian alat secara terperinci adalah sebagai berikut :

1. Catu daya pemancar dan penerima dihidupkan terlebih dahulu. *Extra mic* pada pengirim dipasang pada mic handy talkie. *Extra mic* pada penerima dipasang pada speaker handy talkie. *Handy talkie* diaktifkan dengan mengubah posisi dari OFF ke ON. Pada pengujian digunakan frekuensi 144,44 MHz pada masing-masing transceiver *handy talkie*.
2. Tampilan LCD pada saat catu daya dinyalakan muncul perintah untuk menekan keypad jika ingin mengirim perintah.

```
*****
* Tekan Keypad utk *
* setting waktu    *
*****
```

3. Tampilan selanjutnya adalah sebagai berikut:

```
*****
* Menit:          *
* Detik:          *
*****
```

Data isian dimasukkan dengan penekanan keypad. Tampilan karakter waktu dalam dua digit. Data ini digunakan untuk mengganti waktu hijau pada jalan yang dipilih pengguna. Adapun tampilan saat memasukkan pemilihan jalan :

```
*****
* Jalan:          *
* Yakin [1-Y,2-T] *
*****
```


4. Operator diberi pilihan berupa penekanan tombol 1 bila yakin dan penekanan tombol 2 bila tidak yakin.
5. Operator menekan 1 maka tampilannya sebagai berikut :

```
*****
* kirim jalan:      *
*   mnt   dtk      *
*****
```

Gambar ini menunjukkan bahwa data sedang dikirim. Penekanan tombol 2 membuat tampilannya sebagai berikut:

```
*****
* Tekan Keypad utk *
* setting waktu     *
*****
```

6. Proses pengiriman data yang selesai membuat tampilan seperti gambar di bawah ini.

```
*****
* kirim jalan: 1    *
*   00 mnt 30 dtk  *
*****
```

Pada pengambilan data, antara modul pengirim dengan modul penerima diletakkan berjauhan kurang lebih 20 meter. Data yang ingin dirubah adalah lampu 1 dengan waktu tunda satu menit, lampu 2 dengan waktu tunda satu menit 30 detik, lampu 3 dengan waktu tunda 1 menit 30 detik dan lampu 4 dengan waktu tunda 2 menit. Data masuk ke dalam modul penerima sebagai interupsi. Sehingga saat merubah lampu 1, maka tidak langsung lampu 1 akan menyala tapi menunggu giliran setelah lampu 4.

Pada pengambilan data selanjutnya, Lampu 3 diisi dengan waktu tunda 0 menit 0 detik. Lampu 3 akan selalu merah karena tidak memiliki kesempatan lampu hijau untuk menyala karena tidak memiliki waktu tunda. Sehingga setelah lampu 2 menyala hijau, lampu 4 yang akan menyala hijau dan lampu 3 terus menyala merah.

4.2 Pembahasan

Setelah melakukan tahap perancangan dan pembuatan beserta pengujiannya, tahap selanjutnya adalah membahas dari semua hasil yang telah diperoleh. Penjelasan dari keseluruhan sistem yang ditulis terdiri dari tiga bagian yaitu mengenai keseluruhan sistem, perangkat keras dan perangkat lunak.

4.2.1 Keseluruhan Sistem

Hasil percobaan dengan mengikuti standar pengoperasian mampu membuktikan alat ini telah berjalan sesuai yang diharapkan. Alat ini hanya mampu mengirimkan data masukan untuk mengubah waktu lampu hijau di salah satu persimpangan. Lamanya waktu lampu hijau akan berubah setelah data sukses dikirimkan. Data terakhir yang dikirim tidak tersimpan. Lampu lalu lintas akan kembali ke waktu standar setelah catu daya dimatikan atau reset.

Pada saat catu daya rangkaian dinyalakan akan menyebabkan lampu hijau 1 menyala. Pada alat ini dibuat hanya satu arah saja yang bisa berjalan sehingga lampu di tiang yang lain akan menyala merah. Kesempatan bergerak seterusnya adalah lampu hijau di tiang kedua. Sebelum lampu hijau ini menyala, lampu kuning di tiang pertama dan tiang kedua akan menyala selama 2 detik. Lampu kuning di tiang pertama menandakan bahwa kesempatan bergerak kendaraan di tiang ini akan segera berakhir. Sedangkan lampu kuning di tiang kedua menandakan bahwa kesempatan bergerak untuk jalur ini akan dimulai. Setelah lampu kuning mati, maka lampu hijau pada jalan dua akan menyala, sedangkan lampu pada jalan yang lain menyala merah semua. Proses ini akan berlangsung secara terus menerus dengan pergeseran dari tiang satu ketiang dua, ke tiang ketiga dan ketiang keempat.

Pada jalan yang waktu tundanya di set 0, maka jalan ini akan menyala lampu merah terus menerus. Jika pada jalan 3 diset waktu tundanya nol, maka setelah lampu hijau 2 berakhir, maka bukannya jalan 3 yang lampunya menyala hijau tapi langsung melompat ke jalan 4.

4.2.2 Perangkat Keras

Seluruh operasi kerja sistem ini dikendalikan oleh mikrokontroler AT89S52. Tiap – tiap blok sistem dari mikrokontroler tidak diuji secara khusus. Pengujian mikrokontroler digabung dengan pengujian komponen-komponen lain yang dihubungkan ke dalam mikrokontroler tersebut. Dari pengujian, dapat diketahui port dalam mikrokontroler itu dapat bekerja dengan baik sebagai jalur sinyal kontrol.

Keypad dibentuk dalam susunan 4x3 yang menunjukkan 4 baris dan 3 kolom. Encoder digunakan untuk memudahkan

pembacaan data pada keypad oleh mikrokontroler. Oleh encoder, kombinasi 3 kolom dan 4 baris dari keypad dirubah menjadi kombinasi biner 4 bit. Pada saat ada penekanan keypad, maka akan menyebabkan pin DA berlogika tinggi. Agar penekanan keypad dapat terdeteksi, digunakan kapasitor 10nF untuk melengkapi rangkaian osilator. Pada saat pin DA berlogika tinggi, Data A – Data D membentuk kombinasi biner 4 bit yang merupakan hasil pembacaan baris dan kolom pada keypad. Pada saat penekanan tombol, terdapat gejala *bounce* yang disebabkan getaran mekanis sesaat setelah tombol dilepas atau ditekan sehingga sistem seolah-olah merasakan adanya penekanan tombol yang berulang-ulang atau lebih dari sekali. Untuk menghilangkan gejala ini, pada encoder MM79C22 dilengkapi pin *Key Bounce Mask*. Pada alat, pin ini dihubungkan dengan kapasitor 100 nF yang digroundkan. Fungsinya adalah memberikan waktu tunda pada saat keypad ditekan. Pada saat keypad ditekan, maka pin DA tidak langsung set logika tinggi tetapi menunggu waktu tunda penekanan keypad selesai, setelah waktu tunda selesai dan keypad masih ditekan, maka data telah siap dan DA di set ke logika tinggi.

LCD modul M1632 digunakan untuk menampilkan karakter-karakter berupa instruksi mikrokontroler. LCD modul M1632 telah bekerja efektif pada perancangan sistem ini. Untuk mendukung fungsi kerja LCD, digunakan resistor variabel 10 k Ω . Komponen ini berfungsi sebagai pengatur kecerahan karakter. Data yang ditampilkan di LCD merupakan data yang berasal dari masukan keypad yang telah diolah program maupun dari sumber program. Data-data yang ditampilkan merupakan kode ASCII yang sesuai dengan kombinasi biner 8 bit yang terlampir pada lembar data LCD Modul M1632.

IC TP5088N merupakan *tone encoder* yang dirancang untuk menghasilkan nada seperti yang digunakan dalam sistem telepon DTMF. TP5088N tidak memerlukan pengaturan frekuensi. Frekuensi keluaran akan otomatis dikeluarkan sesuai dengan kombinasi 4 bit masukan data ke TP5088N. Frekuensi yang dikeluarkan merupakan kombinasi dari sinyal *dual tone*. Prinsip kerja dari TP5088N adalah menunggu *tone enable* untuk di set logika tinggi. Pada saat *tone enable* di set logika rendah, maka osilator akan dimatikan sehingga *tone generator* dan keluaran transistor akan mati. Pada saat *tone enable* set ke logika tinggi, maka data pada D0-D4 akan dibaca untuk

kemudian *tone generator* mengkonversikan ke frekuensi pada *tone out* terus menerus sampai *tone enable* diset ke logika rendah.

IC MT 8870 digunakan sebagai masukan bagi mikrokontroler. Fungsi dari IC ini adalah menerjemahkan nada-nada DTMF menjadi kombinasi bilangan biner 4 bit. Prinsip kerja dari MT 8870 adalah pada saat ada sinyal masuk mengandung data DTMF dan melebihi konstanta waktu yang telah ditentukan, maka pin Std akan set ke logika tinggi. Konstanta waktu untuk pengenalan DTMF ditentukan oleh Resistor dan Kapasitor yang terhubung ke kaki St/Gt. Pada alat ini, pin Std dihubungkan dengan port 1.4 pada AT89S52. Pada saat pin Std berlogika nol, maka tidak ada nada DTMF yang masuk ke dalam MT 8870.

Agar data dari IC MT 8870 dapat diambil oleh mikrokontroler, maka pin TOE (*Tristate Output Enable*) harus diset logika tinggi. Pada saat TOE berlogika rendah, data berada pada posisi *high impedansi state* sehingga tidak dapat diambil oleh mikrokontroler. Pada alat ini, TOE di set logika tinggi terus (dihubungkan dengan VDD) sehingga setiap pin Std telah berlogika tinggi, data dapat langsung diambil oleh mikrokontroler.

RTC DS1302 digunakan sebagai referensi pewaktuan waktu tunda. Karena hanya sebagai referensi waktu tunda, maka tidak ada pengesetan waktu sesuai dengan keadaan saat ini. Pin dari RTC DS1302 yang terhubung dengan mikrokontroler hanyalah pin SCLK dan pin I/O data.

4.2.3 Perangkat Lunak

Pengendalian sistem yang berpusat pada mikrokontroler diatur oleh perangkat lunak. Pada perangkat lunak terdapat subprogram yang mengendalikan beberapa peripheral yang mendukung operasi kerja mikrokontroler dalam sistem ini. Inisialisasi harus dipenuhi lebih dahulu agar peripheral bersangkutan dapat berkerja pada proses selanjutnya.

Pada mikrokontroler pengirim, perangkat lunak telah dapat membaca masukan dari keypad untuk kemudian ditampilkan ke LCD dan mengolah data tersebut untuk kemudian dikirimkan melalui DTMF. Data yang dikirim DTMF merupakan data biner 4 bit. Dalam sekali pengiriman data ke mikrokontroler penerima, data yang dikirimkan sebanyak 5 kali secara berurutan. Data yang pertama

dikirim adalah data jalan, kemudian data menit puluhan, menit satuan, detik puluhan dan detik satuan. Antara pengiriman data 1 ke data yang lainnya terdapat jeda dimana DTMF tidak mengirimkan apa-apa. Ini digunakan untuk kemudahan pembacaan nada DTMF pada mikrokontroler penerima. Untuk setiap proses yang sedang terjadi di dalam mikrokontroler, LCD akan menampilkannya.

Pada mikrokontroler penerima, perangkat lunak telah dapat mengatur siklus lalu lintas dengan pewaktuan yang dinamis. Pewaktuan dapat diubah dari jarak jauh menggunakan modul pengirim. Untuk setiap penerimaan data, maka hanya 1 jalan yang sedang diganti pewaktuannya. Untuk referensi pewaktuan, mikrokontroler membaca data yang ada pada RTC DS1302. Proses pewaktuan dapat diamati pada waktu tunda dari menyalanya lampu hijau. Sedangkan untuk penyalan lampu kuning, waktu tundanya tidak bisa diubah, tetap selama 2 detik.

Adapun yang dapat menyebabkan mikrokontroler tidak dapat bekerja dengan baik (*hang*) adalah saat ada ketidakstabilan tegangan masukan pada mikrokontroler. Pada saat pemasangan alat ini, sebaiknya sumber tegangan untuk lampu di masukkan dulu sebelum memasukkan sumber tegangan untuk adaptor mikrokontroler. Ketidakstabilan tegangan ini juga kadang mereset data waktu yang ada di dalam RTC DS1302 (jam, detik dan menit bernilai 0). Namun, karena fungsi RTC hanya sebagai referensi waktu tunda, maka tidak mengganggu kerja alat ini secara keseluruhan.

Adapun penyempurnaan yang dapat dilakukan pada alat agar dapat mengendalikan persimpangan lebih baik lagi jika dipasang pada persimpangan yang nyata adalah dengan penambahan sebagai berikut :

1. Untuk persimpangan tertentu, dimungkinkan dibuatnya kesempatan bergerak dua fase.
2. Digunakan EEPROM untuk menyimpan data-data yang telah diterima sehingga pada saat mikrokontroler reset, waktu tunda yang digunakan waktu tunda terakhir yang diterima.
3. Perbaiki perangkat lunak pada modul pengirim untuk mengirim data tidak hanya untuk satu ruas jalan saja. Jadi pada saat proses merubah waktu tunda, bisa dilakukan lebih dari satu ruas jalan yang dirubah waktu tundanya dengan waktu tunda yang berbeda-beda.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil perancangan, pengujian dan analisa dapat ditarik kesimpulan sebagai berikut :

1. Perencanaan alat pengontrol jarak jauh lampu lalu lintas berbasis mikrokontroler AT 89S52 dapat diwujudkan.
2. Dengan menggunakan frekuensi 144,44 MHz alat ini telah mampu mengubah waktu hijau jalan yang dipilih.
3. RTC DS1302 dapat digunakan sebagai pereferensian waktu yang baik.

5.2 Saran

Ditambahkannya EEPROM pada modul penerima sehingga pada saat mikrokontroler terkena keadaan reset maka waktu tunda yang digunakan adalah waktu tunda yang dipakai pada saat mikrokontroler sebelum reset, bukan waktu tunda standar dari modul penerima.

UNIVERSITAS BRAWIJAYA



DAFTAR PUSTAKA

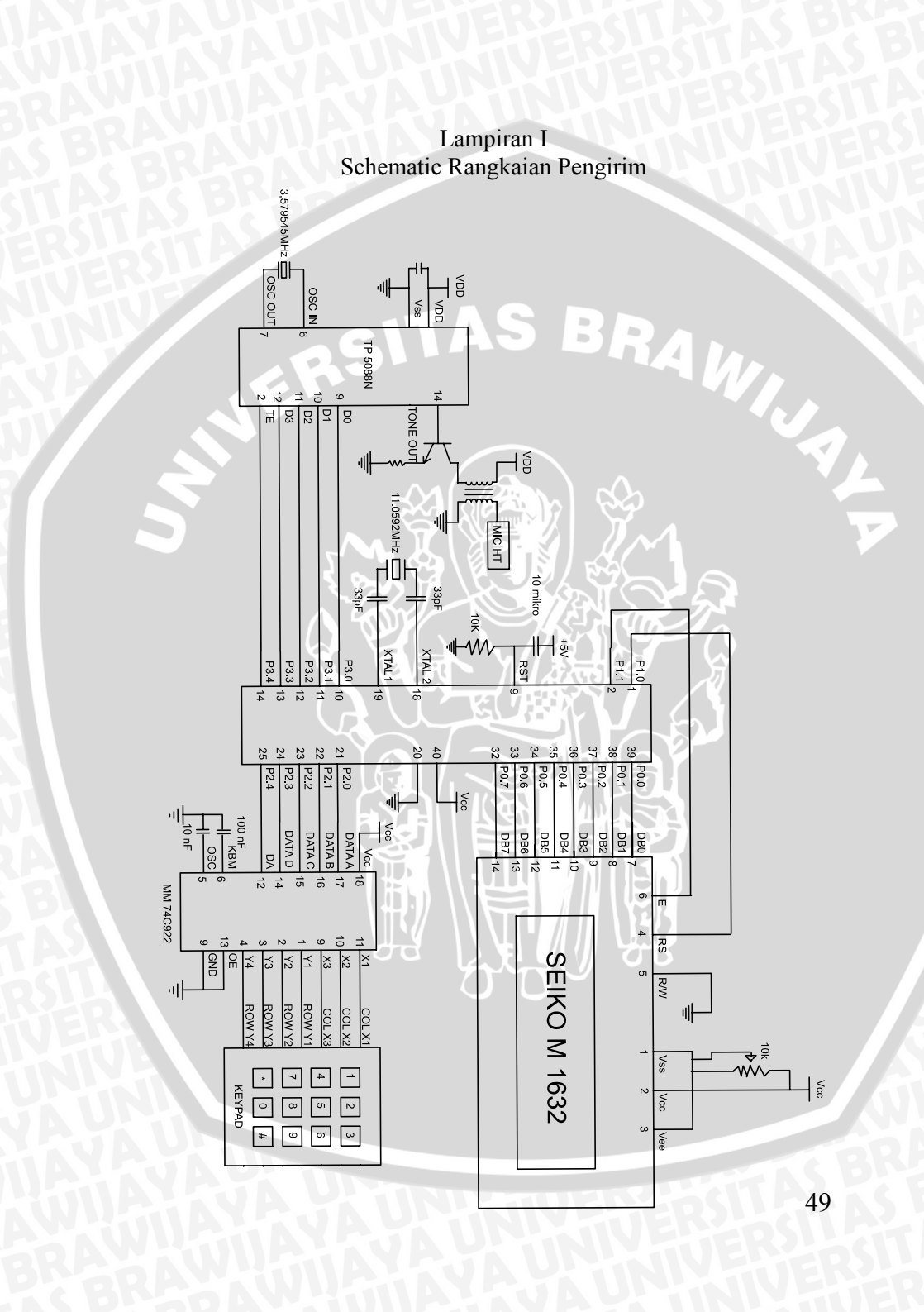
- Anonim, 1987, **Liquid Crystal Module M1632 User Manual**.
Seiko Instrument Inc. Tokyo.
- Anonim, 1991, **TP5088 DTMF Generator for Binary Data**.
National Semiconductor Corporation. California.
<http://www.national.com/> di akses 2 September 2006
- Anonim, 1997, **8870/70C CMOS Integrated DTMF Receiver**.
Mitel. Ottawa. <http://www.mitel.com/> di akses 2 September 2006
- Anonim, 2001, **MM74C922 16-Key Encoder**. Fairchild
Semiconductor Corporation. California.
<http://www.fairchildsemi.com/> di akses 2 September 2006
- Anonim, 2003, **8-bit Microcontroler with 8 K Bytes In-System
Programmable Flash AT89S52**. Atmel Corporation.
California. <http://www.atmel.com> di akses 2 September 2006
- Anonim, 2005, **DS1302 Trickle-Charge Timekeeping Chip**.
Maxim. California. <http://www.maxim-ic.com/> di akses 2
September 2006
- Roody, D., 1996. **Komunikasi Elektronika**. Erlangga. Jakarta.
- Widiarti, N, T, 2004, **Perancangan dan Pembuatan Alat
Pengontrol Jarak Jauh Lampu Lalu Lintas Berbasis
Mikrokontroler AT89C55**. Jurusan Fisika FMIPA
Unibraw. Malang.
http://www.datasheetcatalog.com/datasheets_pdf/M/M/7/4/MM74C922.shtml di akses 2 September 2006
http://www.datasheetcatalog.com/datasheets_pdf/C/M/8/8/CM8870.shtml di akses 2 September 2006
<http://www.innovativeelectronics.com> di akses 8 Juli 2006
http://www.keil.com/dd/docs/datashts/atmel/at89s52_ds.pdf di akses 2 September 2006
<http://pdfserv.maxim-ic.com/en/ds/DS1302.pdf> di akses 2 September 2006
http://www.tranzistoare.ro/datasheets2/13/135988_1.pdf di akses 2 September 2006
http://en.wikipedia.org/wiki/Relay#Types_of_relay di akses 2 September 2006

UNIVERSITAS BRAWIJAYA



Lampiran I
Schematic Rangkaian Pengirim

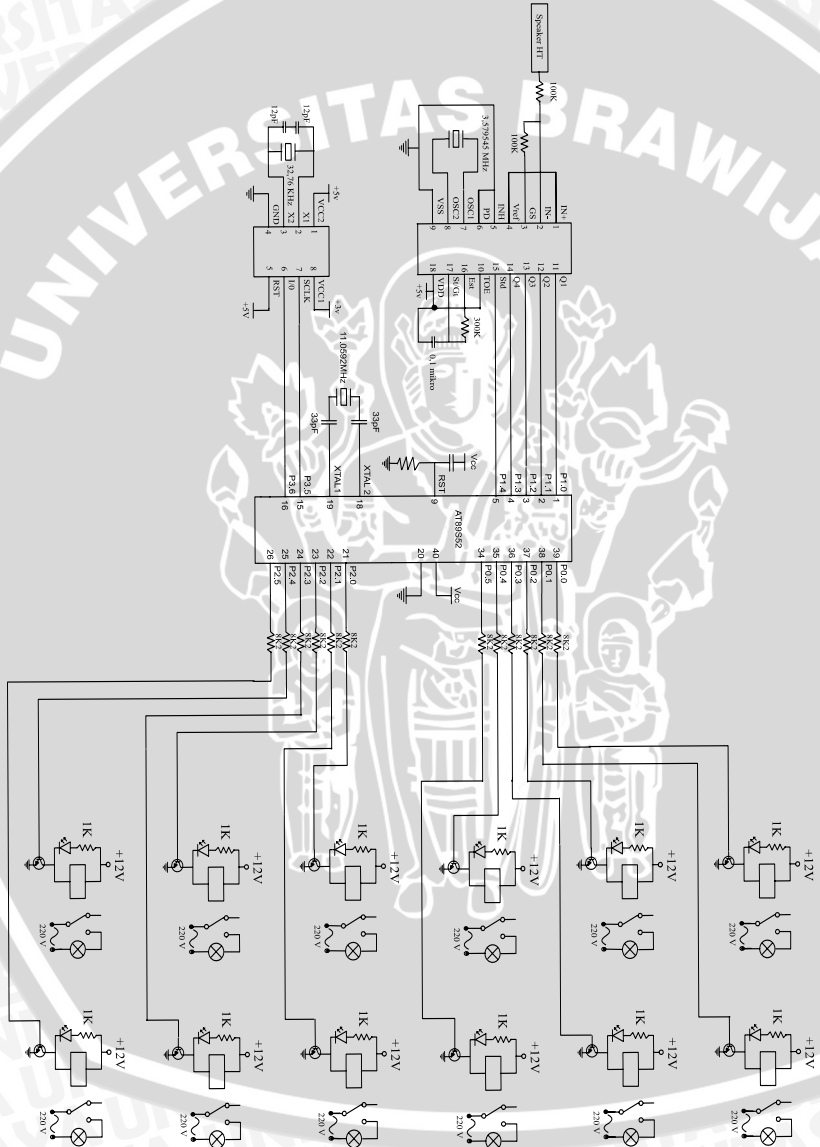
The schematic diagram illustrates the transmitter circuit. It features a TP5088N IC, a SEIKO M 1632 IC, a MM74C922 IC, and a keypad. The TP5088N IC is connected to a 3.579545MHz oscillator, a 11.0592MHz crystal, and a 33pF capacitor. The SEIKO M 1632 IC is connected to a 10k resistor and a 100nF capacitor. The MM74C922 IC is connected to a 100nF capacitor and a 10k resistor. The keypad is connected to the MM74C922 IC. The circuit is powered by VDD and VSS.



UNIVERSITAS BRAWIJAYA



Lampiran II Schematic Rangkaian Penerima



UNIVERSITAS BRAWIJAYA



Lampiran III
Source Code Rangkaian Pengirim

```

$mod51
dtmf      equ    p3
displcr   equ    00000001b
funcset   equ    00111000b
entrmod   equ    00000110b
dispon    equ    00001100b
LCD_RS    bit    p1.0
lcd_CS    bit    p1.1
detik1    equ    32h
detik2    equ    33h
menit1    equ    30h
menit2    equ    31h
jalan     equ    34h

org 0
start: mov sp,#80h

ulang: mov p3,#0
      call initlcd
      call dispawal1
      call keypad
      call dispwaktu
      mov a,#086h
      call controlout
      call keypad
      call dataout
      mov menit1,a
      mov a,#087h
      call controlout
      call keypad
      call dataout
      mov menit2,a
      mov a,#0c6h
      call controlout

      call keypad
      call dataout
      mov detik1,a
      mov r0,#0ah
      call delayx
      call dispjalan
      mov a,#086h
      call controlout
      call keypad
      call dataout
      mov 50h,a
      mov r0,#02h
      call delayx

      pilih1: call keypad
              cjne a,#'1',pilih2
              jmp kirimdata
      pilih2: cjne a,#'2',pilih1
              jmp ulang
      kirimdata:
              call dispjalanout
              call dispdatakirim
              mov r0,#0AH
              call delayx
              jmp ulang

      dispawal1:
              acall initlcd
              mov dptr,#txtawal1
              acall printstring1
              mov dptr,#txtawal2
              acall printstring2

```

```

ret
dispwaktu:
acall initlcd
mov  dptr,#msknmenit
acall printstring1
mov  dptr,#mskndetik
acall printstring2
ret

```

```

dispjalan:
acall initlcd
mov  dptr,#msknjalan
acall printstring1
mov  dptr,#msknyakin
acall printstring2
ret

```

```

dispjalanout:
acall initlcd
mov  dptr,#txtjalan
acall printstring1
mov  dptr,#txtwaktu
acall printstring2
mov  a,#08ch
acall controlout
mov  a,50h
acall dataout
mov  a,50h
anl  a,#0fh
call burst
mov  a,#0c0h
acall controlout
mov  a,menit1
acall dataout
mov  a,menit1
anl  a,#0fh
call burst
mov  a,#0c1h
acall controlout
mov  a,menit2
acall dataout

```

```

mov  a,menit2
anl  a,#0fh
call burst
mov  a,#0c7h
acall controlout
mov  a,detik1
acall dataout
mov  a,detik1
anl  a,#0fh
call burst
mov  a,#0c8h
acall controlout
mov  a,detik2
acall dataout
mov  a,detik2
anl  a,#0fh
call burst
ret

```

```

dispdatakirim:
acall initlcd
mov  dptr,#txtdatakirim1
acall printstring1
mov  dptr,#txtdatakirim2
acall printstring2
ret

```

```

printstring1:
acall posisi1_1
clr  a
movc a,@a+dptr
jnz  printstringloop1

```

```

printstringloop1:
lcall dataout
acall delay_initlcd
inc  dptr
clr  a
movc a,@a+dptr
jnz  printstringloop1
ret

```



```

printstring2:
    acall posisi2_1
    clr a
    movc a,@a+dptr
    jnz printstringloop2
printstringloop2:
    acall dataout
    acall delay_initlcd
    inc dptr
    clr a
    movc a,@a+dptr
    jnz printstringloop2
    ret

posisi1_1:
    mov a,#1
posisi1:
    add a,#10000000b
    dec a
    acall controlout
    acall delay_initlcd
    ret
posisi2_1:
    mov a,#1
posisi2:
    add a,#11000000b
    dec a
    acall controlout
    acall delay_initlcd
    ret

burst: ORL A,#10H
        MOV dtmf,A
        MOV R0,#0AH
        ACALL delayx
        MOV dtmf,#00H
        MOV R0,#05H
        ACALL delayx

delay_initlcd:
    mov r6,#20h
delay_initlcd2:
    mov r7,#10h
    djnz r7,$
    djnz r6,delay_initlcd2
    ret

delay: mov r5,#0
u1: mov r6,#0
u2: mov r7,#0
u3: inc r7
    cjne r7,#5,u3
    inc r6
    cjne r6,#5,u2
    inc r5
    cjne r5,#5,u1
    ret
delayx: PUSH 0
        PUSH 1
        PUSH 2
delayx1: MOV r1,#0FAH
delayx2: MOV r2,#0FAH
        DJNZ r2,$
        DJNZ r1,delayx2
        DJNZ r0,delayx1
        POP 2
        POP 1
        POP 0
        RET

controlout:
    clr lcd_rs
    acall delay
    clr lcd_cs
    acall delay
    setb lcd_cs

```

```

acall delay
mov p0,a
clr lcd_cs
acall delay
ret
dataout:
setb lcd_rs
acall delay
clr lcd_cs
acall delay
setb lcd_cs
acall delay
mov p0,a ;data
clr lcd_cs
acall delay
ret
initlcd:
mov a,#funcset
acall controlout
acall delay_initlcd

mov a,#dispon
acall controlout
acall delay_initlcd

mov a,#dispcr
acall controlout
acall delay_initlcd

mov a,#entrmod
acall controlout
acall delay_initlcd
ret
keypad:
MOV A,#30H
LJMP tutup
bintang: CJNE A,#08H,pagar
MOV A,#2AH
LJMP tutup
JNB P2.4,$
MOV A,P2
ANL A,#0FH
CJNE A,#0BH,kedua
MOV A,#31H
LJMP tutup
kedua: CJNE A,#07H,ketiga
MOV A,#32H
LJMP tutup
ketiga: CJNE a,#03H,keempat
MOV A,#33H
LJMP tutup
keempat: CJNE A,#0AH,kelima
MOV A,#34H
LJMP tutup
kelima: CJNE A,#06H,keenam
MOV A,#35H
LJMP tutup
keenam: CJNE A,#02H,ketujuh
MOV A,#36H
LJMP tutup
ketujuh: CJNE A,#09H,kelapan
MOV A,#37H
LJMP tutup
kelapan:
CJNE A,#05H,kesembilan
MOV A,#38H
LJMP tutup
kesembilan:
CJNE A,#01H,kosong
MOV A,#39H
LJMP tutup
kosong: CJNE A,#04H,bintang

```

pagar: CJNE A,#00H,tutup
MOV A,#23H
tutup: JB P2.4,\$
MOV B,A
RET

txtawal1:

db 'tekan keypad utk',0

txtawal2:

db 'setting waktu',0

msknmenit:

db 'menit:',0

mskndetik:

db 'detik:',0

msknjalan:

db 'jalan:',0

msknyakin:

db 'yakin (1-Y,2-T)',0

txtjalan:

db 'kirim jalan',0

txtwaktu:

db ' mnt dtk',0

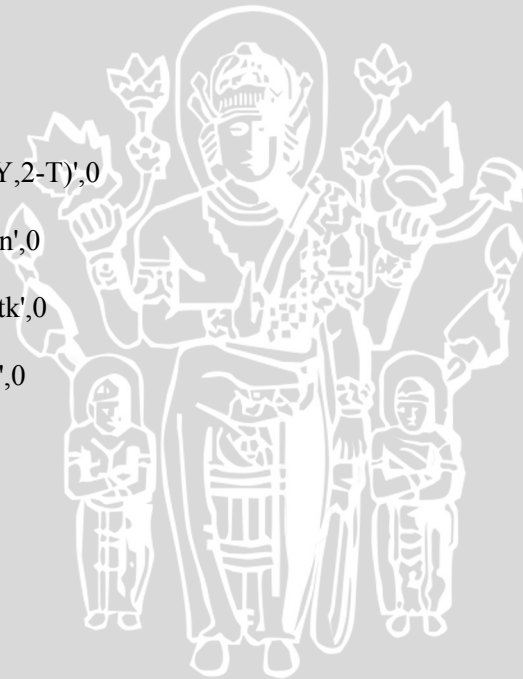
txtdatakirim1:

db 'data telah',0

txtdatakirim2:

db 'dikirim',0

end



UNIVERSITAS BRAWIJAYA



Lampiran IV Source Code Rangkaian Penerima

```

$MOD51                                jmp    start
                                       org    000bh
                                       jmp    intert0
                                       org    0100H

DTMF      equ    p1
Lampuh1   bit    p0.0
LampuK1   bit    p0.1
Lampum1   bit    p0.2
Lampuh2   bit    p0.3
LampuK2   bit    p0.4
Lampum2   bit    p0.5
LampuM4   bit    p2.0
LampuK4   bit    p2.1
LampuH4   bit    p2.2
LampuM3   bit    p2.3
LampuK3   bit    p2.4
LampuH3   bit    p2.5

seca      equ    24H
mina      equ    25H
hra       equ    26H
secb      equ    27H
minb      equ    28H
hrb       equ    29H
smenit1   equ    2ah
smenit2   equ    2bh
sm3        equ    2ch
sm4        equ    2dh
SD1        equ    33H
SD2        equ    34H
SD3        equ    35h
SD4        equ    36H
menitp     equ    37h
menits     equ    38h
detikp     equ    39h
detiks     equ    3ah

                                       $INCLUDE(DRIVER.ASM)
                                       ;=====
                                       ; Rutin Untuk Mengambil Waktu
                                       ;Pada RTC
                                       ;=====
cek_a:    mov    B,#sec
           lcall byterdclreg
           mov    seca,a
           mov    B,#min
           lcall byterdclreg
           mov    mina,a
           mov    B,#hr
           lcall byterdclreg
           mov    hra,a
           ret

                                       ;=====
                                       ; Rutin Untuk Waktu
                                       ;Pembanding
                                       ;=====
cek_b:    mov    B,#sec
           lcall byterdclreg
           mov    secb,a
           mov    B,#min
           lcall byterdclreg
           mov    minB,a
           mov    B,#hr
           lcall byterdclreg
           mov    hrb,a

org    0000h

```



```

ret
; =====
; Rutin Set Waktu Default
; setiap lampu menyala 1 s
; =====
SETDEF:    mov    Sd1,#01h
           mov    sd2,#01h
           mov    sd3,#01h
           mov    sd4,#01h
           mov    smenit1,#0
           mov    smenit2,#0
           mov    sm3,#0
           mov    sm4,#0
           ret
; =====
; Program Utama
; =====
org 400h
START:     mov    SP,#40H
           mov    R0,#00H
           mov    R1,#00H
           mov    R2,#00H
           mov    R3,#00H
           mov    R4,#00H
           mov    R5,#00H
           mov    R6,#00H
           mov    R7,#00H
           mov    P0,#0
           mov    P2,#0
           lcall  setdef
           mov    TMOD,#001H
           mov    TL0,#03DH
           mov    TH0,#0B0H
           clr    TF0
           setb   TR0
           mov    ie,#82h
           lcall  tampilmerah
HIJAU1:    lcall  preh1
           lcall  tampil1hijau
           lcall  end_a
           lcall  preh2
           lcall  tampil2hijau
           lcall  end_b
           lcall  preh3
           lcall  tampil3hijau
           lcall  end_c
           lcall  preh4
           lcall  tampil4hijau
           lcall  end_d
           sjmp   hijau1
           end_e:  mov    a,sd1
                   add    a,smenit1
                   cjne   a,#0,hijau1
                   add    a,sd2
                   add    a,smenit2
                   cjne   a,#0,hijau2
                   add    a,sd3
                   add    a,sm3
                   cjne   a,#0,hijau3
                   add    a,sd4
                   add    a,sm4
                   cjne   a,#0,hijau4
                   call    tampilmerah
                   jmp     end_e
; =====
; Tampilan LED
; =====
TAMPILM_ALL:  setb   lampum1
               setb   lampum2
               setb   lampum3
               setb   lampum4
               clr    lampuh1
               clr    lampuh2
               clr    lampuh3

```

```

clr    lampuh4
clr    lampuk1
clr    lampuk2
clr    lampuk3
clr    lampuk4
ret

```

TAMPILHO_A:

```

setb   LampuH1
setb   Lampuk1
ret

```

TAMPILHO_B:

```

setb   LampuH2
setb   Lampuk2
ret

```

TAMPILHO_C:

```

setb   LampuH3
setb   Lampuk3
ret

```

TAMPILHO_D:

```

setb   LampuH4
setb   Lampuk4
ret

```

TAMPILH_A:

```

clr    lampuk1
setb   lampuh1
clr    lampum1
clr    lampuh2
clr    lampuh3
clr    lampuh4
clr    lampuk1
clr    lampuk2
clr    lampuk3
clr    lampuk4
setb   lampum2
setb   lampum3
setb   lampum4
ret

```

TAMPILH_B: clr lampuk2

```

setb   lampuh2

```

```

clr    lampum2
clr    lampuh1
clr    lampuh3
clr    lampuh4
clr    lampuk1
clr    lampuk2
clr    lampuk3
clr    lampuk4
setb   lampum1
setb   lampum3
setb   lampum4
ret

```

TAMPILH_C: clr lampuk3

```

setb   lampuh3
clr    lampum3
clr    lampuh2
clr    lampuh1
clr    lampuh4
clr    lampuk1
clr    lampuk2
clr    lampuk3
clr    lampuk4
setb   lampum2
setb   lampum1
setb   lampum4
ret

```

TAMPILH_D: clr lampuk4

```

setb   lampuh4
clr    lampum4
clr    lampuh2
clr    lampuh3
clr    lampuh1
clr    lampuk1
clr    lampuk2
clr    lampuk3
clr    lampuk4
setb   lampum2
setb   lampum3
setb   lampum1

```

```

ret
tampilmo_A: setb lampum1
setb lampuk1
ret
tampilmo_b: setb lampum2
setb lampuk2
ret
tampilmo_c: setb lampum3
setb lampuk3
ret
tampilmo_d: setb lampum4
setb lampuk4
ret

```

```

;=====
;Rutin Untuk Mengeluarkan preh1:
;Tampilan
;
;=====

```

```

tampilmerah: call cek_a      preh1: call tampilmo_a
mov A,seca
add A,#1
mov seca,A
CLR C
subb A,#5AH
jc TAMPILAN_0
Tmin_0: mov seca,A
mov A,mina
add A,#01H
mov mina,A
CLR C
subb A,#5AH
;Lebih Dari 60 Menit= Jam +1
jc TAMPILAN_0
TJAM_0: mov mina,A
mov A,hra
add A,#01H
mov hra,A

```

```

TAMPILAN_0: lcall preh_1A: lcall cek_b
TAMPILM_ALL mov A,hrb

```

```

lcall cek_b
mov A,hrb
CLR C
subb A,hra
jc TAMPILAN_0
mov A,minB
CLR C
subb A,mina
jc TAMPILAN_0
mov A,secb
CLR C
subb A,seca
jc TAMPILAN_0
ret

```

```

mov a,smenit1
add a,sd1
cjne a,#0,pretmp1
jmp hijau2

```

```

call tampilmo_a
call cek_a
mov A,seca
add A,#2
mov seca,A
CLR C
subb A,#5AH
jc preh_1A
mov seca,A
mov A,mina
add A,#01H
mov mina,A
CLR C
subb A,#5AH
jc preh_1A

```

```

mov mina,A
mov A,hra
add A,#01H
mov hra,A

```

```

CLR    C
subb   A,hra
jc     preh_1A
mov    A,minB
CLR    C
subb   A,mina
jc     preh_1A
mov    A,secb
CLR    C
subb   A,seca
jc     preh_1A
ret
tampil1hijau: call cek_a
mov    A,mina
add    A,smenit1
mov    minA,A
mov    A,seca
add    A,sd1
mov    seca,A
CLR    C
subb   A,#5AH
jc     Tmin1X
Tmin_1: mov    seca,A
mov    A,mina
add    A,#01H
mov    minA,A
Tmin1X: mov    a,mina
CLR    C
subb   A,#5AH
jc     TAMPILAN_1
TJAM_1: mov    minA,A
mov    A,hra
add    A,#01H
mov    hra,A
TAMPILAN_1: lcall
TAMPILH_A
lcall  cek_b
mov    A,hrb
CLR    C
subb   A,hra
jc     TAMPILAN_1
mov    A,minB
CLR    C
subb   A,mina
jc     TAMPILAN_1
mov    A,secb
CLR    C
subb   A,seca
jc     TAMPILAN_1
ret
End_A: lcall tampilho_a
ret
preh2: mov    a,smenit2
add    a,sd2
cjne   a,#0,pretmp2
jmp    hijau3
pretmp2: call tampilmo_b
lcall  cek_a
mov    A,seca
add    A,#2
mov    seca,A
CLR    C
subb   A,#5AH
jc     preh_2A
mov    seca,A
mov    A,mina
add    A,#01H
mov    minA,A
CLR    C
subb   A,#5AH
jc     preh_2A
mov    minA,A
mov    A,hra
add    A,#01H
mov    hra,A
preh_2A: lcall cek_b
mov    A,hrb
CLR    C

```

```

subb A,hra
jc preh_2A
mov A,minB
CLR C
subb A,mina
jc preh_2A
mov A,secb
CLR C
subb A,seca
jc preh_2A
ret
End_b:      lcall tampilho_b
ret
preh3:      mov a,sm3
add a,sd3
cjne a,#0,pretmp3
jmp hijau4
pretmp3:    lcall tampilmo_c
lcall cek_a
mov A,seca
add A,#2
mov seca,A
CLR C
subb A,#5AH
jc Tmin2X
Tmin_2:    mov seca,A
mov A,mina
add A,#01H
mov mina,A
Tmin2X:    mov a,mina
CLR C
subb A,#5AH
jc TAMPILAN_2
TJAM_2:    mov mina,A
mov A,hra
add A,#01H
mov hra,A
TAMPILAN_2:      lcall
TAMPILH_B
lcall cek_b
mov A,hrb
CLR C
subb A,hra
jc TAMPILAN_2
mov A,minB
CLR C
subb A,mina
jc TAMPILAN_2
mov A,secb
CLR C
subb A,seca
jc TAMPILAN_2
ret

```



```

jc    preh_3A
mov    A,minB
CLR    C
subb   A,mina
jc    preh_3A
mov    A,secb
CLR    C
subb   A,seca
jc    preh_3A
ret
tampil3hijau: lcall cek_a
mov    A,mina
add    A,Sm3
mov    mina,A
mov    A,seca
add    A,sd3
mov    seca,A
CLR    C
subb   A,#5AH
jc    Tmin3X
Tmin_3: mov    seca,A
mov    A,mina
add    A,#01H
mov    mina,A
Tmin3X: mov    a,mina
CLR    C
subb   A,#5AH
jc    TAMPILAN_3
TJAM_3: mov    mina,A
mov    A,hra
add    A,#01H
mov    hra,A
TAMPILAN_3: lcall
TAMPILH_C
lcall  cek_b
mov    A,hrb
CLR    C
subb   A,hra
jc    TAMPILAN_3
mov    A,minB
CLR    C
subb   A,mina
jc    TAMPILAN_3
mov    A,secb
CLR    C
subb   A,seca
jc    TAMPILAN_3
ret
end_c: call  tampilho_c
ret
preh4: mov    a,sm4
add    a,sd4
cjne   a,#0,pretmp4
jmp    end_e
pretmp4: call  tampilmo_d
lcall  cek_a
mov    A,seca
add    A,#2
mov    seca,A
CLR    C
subb   A,#5AH
jc    preh_4A
mov    seca,A
mov    A,mina
add    A,#01H
mov    mina,A
CLR    C
subb   A,#5AH
jc    preh_4A
mov    mina,A
mov    A,hra
add    A,#01H
mov    hra,A
preh_4A: lcall  cek_b
mov    A,hrb
CLR    C
subb   A,hra
jc    preh_4A

```

```

mov    A,minB
CLR    C
subb   A,mina
jc     preh_4A
mov    A,secb
CLR    C
subb   A,seca
jc     preh_4A

ret
tampil4hijau: lcall cek_a
mov    A,mina
add    A,SM4
mov    mina,A
mov    A,seca
add    A,sd4
mov    seca,A
CLR    C
subb   A,#5AH
jc     Tmin4X
Tmin_4: mov    seca,A
mov    A,mina
add    A,#01H
mov    mina,A
Tmin4X: mov    a,mina
CLR    C
subb   A,#5AH
jc     TAMPILAN_4
TJAM_4: mov    mina,A
mov    A,hra
add    A,#01H
mov    hra,A
TAMPILAN_4: lcall
TAMPILH_D
lcall  cek_b
mov    A,hrb
CLR    C
subb   A,hra
jc     TAMPILAN_4

mov    A,minB
CLR    C
subb   A,mina
jc     TAMPILAN_4
mov    A,secb
CLR    C
subb   A,seca
jc     TAMPILAN_4

ret
end_d: call  tampilho_d
ret

org    08c0h
intert0: JNB   P1.4,jakhiri
JB      P1.4,$
CALL    delay
mov     A,#0
JMP     hijau1t
Jakhiri: JMP    akhiri
jhijau2t: jmp    hijau2t
hijau1t: JB      P1.4,$
mov     a,DTMF
ANL     A,#0FH
cjne    A,#1,jhijau2t
acall   delay
jnb     p1.4,$
acall   delay
mov     a,dtmf
jb      p1.4,$
anl     a,#0fh
swap    a
mov     menitp,a
jnb     p1.4,$
acall   delay
mov     a,dtmf
jb      p1.4,$
anl     a,#0fh
mov     menitp,a
acall   delay

```

```

jnb    p1.4,$
acall  delay
mov    a,dtmf
jb     p1.4,$
acall  delay
anl    a,#0fh
swap   a
mov    detikp,a
jnb    p1.4,$
acall  delay
jb     p1.4,$
acall  delay
mov    a,dtmf
anl    a,#0fh
mov    detiks,a
acall  delay
mov    a,menitp
orl    a,menits
acall  delay
mov    smenit1,a
acall  delay
mov    a,detikp
orl    a,detiks
acall  delay
mov    sd1,a
acall  delay
LJMP   akhiri
jhijau3t: jmp hijau3t

```

HIJAU2t:

```

cjne   A,#2,jHIJAU3t
acall  delay
jnb    p1.4,$
acall  delay
mov    a,dtmf
jb     p1.4,$
anl    a,#0fh
swap   a
mov    menitp,a

```

```

jnb    p1.4,$
acall  delay
mov    a,dtmf
jb     p1.4,$
anl    a,#0fh
mov    menits,a
acall  delay
jnb    p1.4,$
acall  delay
mov    a,dtmf
jb     p1.4,$
acall  delay
anl    a,#0fh
swap   a
mov    detikp,a
jnb    p1.4,$
acall  delay
jb     p1.4,$
acall  delay
mov    a,dtmf
anl    a,#0fh
mov    detiks,a
acall  delay
mov    a,menitp
orl    a,menits
acall  delay
mov    smenit2,a
acall  delay
mov    a,detikp
orl    a,detiks
acall  delay
mov    sd2,a
acall  delay
LJMP   akhiri

```

HIJAU3t:

```

A,#3,HIJAU4t
acall  delay
jnb    p1.4,$
acall  delay

```

cjne

mov a,dtmf	HIJAU4t:
jb p1.4,\$	cjne A,#0FH,trimahijau4
anl a,#0fh	LJMP akhiri
swap a	trimahijau4: acall delay
mov menitp,a	jnb p1.4,\$
jnb p1.4,\$	acall delay
acall delay	mov a,dtmf
mov a,dtmf	jb p1.4,\$
jb p1.4,\$	anl a,#0fh
anl a,#0fh	swap a
mov menitp,a	mov menitp,a
acall delay	jnb p1.4,\$
jnb p1.4,\$	acall delay
acall delay	mov a,dtmf
mov a,dtmf	jb p1.4,\$
jb p1.4,\$	anl a,#0fh
acall delay	mov menitp,a
anl a,#0fh	acall delay
swap a	jnb p1.4,\$
mov detikp,a	acall delay
jnb p1.4,\$	mov a,dtmf
acall delay	jb p1.4,\$
jb p1.4,\$	acall delay
acall delay	anl a,#0fh
mov a,dtmf	swap a
anl a,#0fh	mov detikp,a
mov detiks,a	jnb p1.4,\$
acall delay	acall delay
mov a,menitp	jb p1.4,\$
orl a,menits	acall delay
acall delay	mov a,dtmf
mov sm3,a	anl a,#0fh
acall delay	mov detiks,a
mov a,detikp	acall delay
orl a,detiks	mov a,menitp
acall delay	orl a,menits
mov sd3,a	acall delay
acall delay	mov sm4,a
LJMP akhiri	acall delay

```
mov    a,detikp
orl     a,detiks
acall   delay
mov     sd4,a
acall   delay
```

```
akhiri: clr    TF0
        setb   TR0
        reti
delay:  push   5
        push   6
        push   7
        mov    r5,#02H
delay1: mov    r6,#0FAH
delay2: mov    r7,#0FAH
        djnz   R7,$
        djnz   r6,delay2
        djnz   r5,delay1
        pop    7
        pop    6
        pop    5
        ret
END
```



UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



Lampiran V
Source Code Pengujian RTC DS1302 dan Relay

\$MOD51

seca equ 24H

org 0100H

\$INCLUDE(DRIVER.ASM)

org 400h

start: mov p0,#0ffh

call delay

mov p0,#0

call delay

jmp start

delay: mov b,#sec

lcall byterdclreg

add a,#5

mov seca,a

clr c

subb a,#5AH

jc delay1

mov seca,a

delay1: mov b,#sec

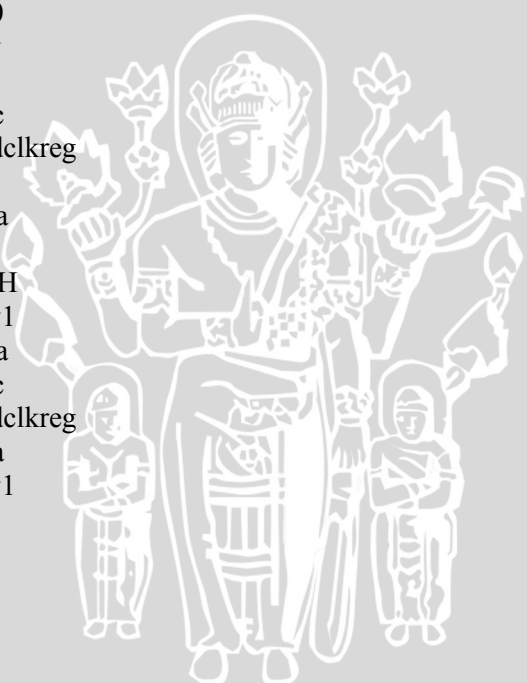
lcall byterdclreg

subb a,seca

jc delay1

ret

end



Lampiran VI
Source Code Driver RTC DS1302

```

;-----
;RTC          SEREE          DT51
;-----
;SCLK(1)      SK(1)          P1.0(1)
;I/O(1)       DI/DO(1)      P1.1(1)
;RST\0)       CS(1)         P1.3(1)
;-----
;RTC Address I/O
;-----
SEC           EQU    00H
MIN           EQU    02H
HR            EQU    04H
DATE          EQU    06H
MONTH         EQU    08H
DAY           EQU    0AH
YEAR          EQU    0CH
CTR           EQU    0EH      ;CONTROL REGISTER
TCR           EQU    10H      ;TRICKLE CHARGE REGISTER
Flag          DATA  0020H
Fail          BIT    Flag.0

CSEG

;-----
;RESET SERIAL EEPROM & RTC
;-----
RESET:        CLR     P1.0      ;SK=0
               SETB    P1.3      ;CS=1
               CLR     P1.3      ;CS=0
               RET

;-----
;WRITE BIT TO SERIAL EEPROM
;IN : A=DATA TO WRITE
;-----
WRSEREE:      PUSH    ACC
               PUSH    07H
               MOV     R7,#8

```

```

NxtWRBit:   RLC    A
            JC     WRBitH
WRBitL:      CLR    P1.1
            AJMP   WRCIk
WRBitH:      SETB   P1.1
WRCIk:       CLR    P1.0
            SETB   P1.0
            DJNZ   R7,NxtWRBit
            POP     07H
            POP     ACC
            RET

```

```

;-----
;READ BIT FROM SERIAL EEPROM
;OUT : A=READ DATA
;-----

```

```

RDSEREE:     PUSH   07H
            CLR     A
            MOV     R7,#8
            SETB    P1.1
RDCIk:        CLR    P1.0
            SETB    P1.0
            JB      P1.1,RDBitH
RDBitL:       CLR    C
            AJMP   ShiftInBit
RDBitH:       SETB   C
ShiftInBit:   RLC     A
            DJNZ   R7,RDCIk
            POP     07H
            RET

```

```

;-----
;CHECK READY\BUSY SERIAL EEPROM
;-----

```

```

CHKCONSEREE: PUSH   07H
            PUSH   06H
            PUSH   05H
            MOV    R5,#2
            MOV    R7,#0FFH
            MOV    R6,#0FFH
            CLR     Fail

```



```

StrbCS:    SETB  P1.1
           CLR   P1.3                ;CS=0
           SETB  P1.3                ;CS=1
W4Con:     CLR   P1.0
           SETB  P1.0
           JB    P1.1,WaitTimer1
           MOV   R5,#2
           AJMP  Busy
WaitTimer1: DJNZ  R7,W4Con
           MOV   R7,#0FFH
           DJNZ  R5,W4Con
           SETB  Fail
           AJMP  Ready
Busy:      CLR   P1.0
           SETB  P1.0
           JNB   P1.1,WaitTimer2
           AJMP  Ready
WaitTimer2: DJNZ  R6,Busy
           MOV   R6,#0FFH
           DJNZ  R5,Busy
           SETB  Fail
Ready:     POP   05H
           POP   06H
           POP   07H
           RET

```

```

;-----
;READ DATA FROM SERIAL EEPROM
;IN : DPTR = ADDRESS
;OUT : A = READ DATA
;-----

```

READEE:

```

PUSH  07H
PUSH  06H
MOV   R6,#2
MOV   R7,#0FFH
CLR   Fail
SETB  P1.3                ;CS=1
MOV   A,DPH
ANL   A,#01H

```

```

ORL      A,#00001100B ;Bit3=SB Bit2,1=OPCODE
ACALL    WRSEREE      ;   1   1 0
MOV      A,DPL
ACALL    WRSEREE
SETB     P1.1
Dummy:   JB      P1.1,WaitTimer3
          AJMP    DataReady
WaitTimer3: DJNZ   R7,Dummy
            MOV    R7,#0FFH
            DJNZ   R6,Dummy
            SETB   Fail
            AJMP   EOREADEE
DataReady: ACALL    RDSEREE
EOREADEE: CLR     P1.3 ;CS=0
            POP     06H
            POP     07H
            RET

```

```

;-----
;ERASE AND WRITE TO SERIAL EEPROM ENABLE
;-----

```

```

EWEN:
        PUSH     ACC
        SETB     P1.3 ;CS=1
        MOV      A,#00001001B
        ACALL    WRSEREE ;   1   0 0 1
        MOV      A,#10000000B ;Bit7=OPCODE
        ACALL    WRSEREE
        CLR      P1.3 ;CS=0
        POP      ACC
        RET

```

```

;-----
;ERASE AND WRITE TO SERIAL EEPROM DISABLE
;-----

```

```

EWDS: PUSH     ACC
        SETB     P1.3 ;CS=1
        MOV      A,#00001000B ;Bit3=SBBit2,1,0=OPCODE
        ACALL    WRSEREE
        MOV      A,#00000000B ;Bit7=OPCODE
        ACALL    WRSEREE

```

CLR
POP
RET

P1.3
ACC

;CS=0

;WRITE DATA TO SERIAL EEPROM
;IN : DPTR = ADDRESS
; A = DATA TO WRITE

WRITEEE:

PUSH ACC
SETB P1.3 ;CS=1
MOV A,DPH
ANL A,#01H
ORL A,#00001010B ;Bit3=SB Bit2,1=OPCODE
ACALL WRSEREE ; 1 0 1
MOV A,DPL
ACALL WRSEREE
POP ACC
ACALL WRSEREE
ACALL CHKCONSEREE
CLR P1.3 ;CS=0
RET

;WRITE DATA AT ALL SERIAL EEPROM
;IN : A=DATA TO WRITE

WRAL:

PUSH ACC
SETB P1.3 ;CS=1
MOV A,#00001000B ;Bit3=SB
Bit2,1,0=OPCODE
ACALL WRSEREE ; 1 0 0 0
MOV A,#10000000B ;Bit7=OPCODE
ACALL WRSEREE ; 1
POP ACC
ACALL WRSEREE
ACALL CHKCONSEREE
CLR P1.3 ;CS=0
RET

```

;-----
;ERASE DATA AT SERIAL EEPROM
;IN : DPTR=ADDRESS TO ERASED
;-----
ERASEEE:  PUSH  ACC
          SETB  P1.3                      ;CS=1
          MOV   A,DPH
          ANL   A,#01H
          ORL   A,#00001110B             ;Bit3=SB
Bit2,1=OPCODE
          ACALL  WRSEREE  ; 1 1 1
          MOV   A,DPL
          ACALL  WRSEREE
          ACALL  CHKCONSEREE
          CLR   P1.3                      ;CS=0
          POP   ACC
          RET

;-----
;ERASE ALL DATA AT SERIAL EEPROM
;-----
ERAL:     PUSH  ACC
          SETB  P1.3                      ;CS=1
          MOV   A,#00001001B             ;Bit3=SB
Bit2,1,0=OPCODE
          ACALL  WRSEREE  ; 1 0 0 1
          MOV   A,#00000000B             ;Bit7=OPCODE
          ACALL  WRSEREE  ; 0
          ACALL  CHKCONSEREE
          CLR   P1.3                      ;CS=0
          POP   ACC
          RET

;      RTC DS1302 ROUTINE
;-----
;WRITE BIT TO RTC
;IN : A = DATA/ADDRESS TO BE WRITE
;-----
WRBITRTC: PUSH  07H
          MOV   R7,#8
ShfOutBit: RRC   A

```

	JNC	WrBitLow
WrBitHigh:	SETB	P1.1
	AJMP	ClkIn
WrBitLow:	CLR	P1.1
ClkIn:	CLR	P1.0
	SETB	P1.0
	DJNZ	R7,ShfOutBit
	RRC	A
	POP	07H
	RET	

```

;-----
;READ BIT FROM RTC
;OUT : A = READ DATA
;-----

```

RDBITRTC:	PUSH	07H	
	CLR	A	;Init ACC to zero
	SETB	P1.1	;P1.1 set as input
	MOV	R7,#8	
ClkOut:	SETB	P1.0	
	CLR	P1.0	
	JNB	P1.1,RdBitLow	
RdBitHigh:	SETB	C	
	AJMP	ShfInBit	
RdBitLow:	CLR	C	
ShfInBit:	RRC	A	
	DJNZ	R7,ClkOut	
	POP	07H	
	RET		

```

;-----
;WRITE BYTE TO CLOCK REGISTER
;IN : B = REGISTER ADDRESS
;   A = DATA TO WRITE
;-----

```

BYTEWRCLKREG:	PUSH	ACC	
	PUSH	B	
	ACALL	RESET	
	XCH	A,B	;A=Address
	ORL	A,#80H	;Bit7=1
	ANL	A,#0FEH	;Bit0=0=WR


```

ACALL    WRBITRTC    ;Write address
XCH      A,B         ;A=Data
ACALL    WRBITRTC    ;Write Data
ACALL    RESET
POP      B
POP      ACC
RET

```

```

;-----
;READ BYTE FROM CLOCK REGISTER

```

```

;IN  : B = Register Address

```

```

;OUT : A = Data read
;-----

```

```

BYTERDCLKREG:  PUSH  B
               ACALL  RESET
               XCH    A,B           ;A=Address
               ORL    A,#81H        ;Bit7,Bit0=1
               ACALL  WRBITRTC      ;Write Address
               ACALL  RDBITRTC      ;A=Read Data
               ACALL  RESET
               POP    B
               RET

```

```

;-----
;WRITE DISABLE ROUTINE
;-----

```

```

WRDIS:        PUSH  ACC
               PUSH  B
               MOV   A,#80H
               MOV   B,#CTR
               ACALL BYTEWRCLKREG
               POP   B
               POP   ACC
               RET

```

```

;-----
;WRITE ENABLE ROUTINE
;-----

```

```

WRENB:        PUSH  ACC
               PUSH  B
               MOV   A,#00H
               MOV   B,#CTR

```

```

ACALL    BYTEWRCLKREG
POP      B
POP      ACC
RET

;-----
;RUN CLOCK ROUTINE
;-----
RUNCLK:  PUSH  ACC
          PUSH  B
          MOV   B,#SEC
          ACALL BYTERDCLKREG
          ANL   A,#7FH                ;0XXXXXXXB
          ACALL BYTEWRCLKREG
          POP   B
          POP   ACC
          RET

;-----
;STOP CLOCK ROUTINE
;-----
STOPCLK: PUSH  ACC
          PUSH  B
          MOV   B,#SEC
          ACALL BYTERDCLKREG
          ORL   A,#80H                ;1XXXXXXXB
          ACALL BYTEWRCLKREG
          POP   B
          POP   ACC
          RET

;-----
;WRITE BYTE TO RAM REGISTER
;IN : B = RAM ADDRESS
;    A = DATA TO WRITE
;-----
BYTEWRRAM: PUSH  ACC
            PUSH  B
            ACALL RESET
            XCH   A,B                ;A=Address
            ORL   A,#0C0H            ;Bit7,Bit6=1
            ANL   A,#0FEH            ;Bit0=0=WR

```

```

ACALL      WRBITRTC      ;Write address
XCH  A,B      ;A=Data
ACALL      WRBITRTC      ;Write Data
ACALL      RESET
POP  B
POP  ACC
RET

```

```

;-----
;READ BYTE FROM RAM
;IN  : B = RAM Address
;OUT : A = Data read
;-----

```

```

BYTERDRAM:      PUSH  B
                ACALL  RESET
                XCH  A,B      ;A=Address
                ORL  A,#0C1H    ;Bit7,Bit6,Bit0=1
                ACALL  WRBITRTC ;Write address
                ACALL  RDBITRTC ;A=Data read
                ACALL  RESET
                POP  B
                RET

```

```

;-----
;WRITE BURST TO CLOCK REGISTER
;IN  : R0 = Point to first buffer addr
;Remark : - Prepare 8 clock register data
;          start @ addr point by R0
;          - Write start @bit 0 address 0
;-----

```

```

BURSTWRCLKREG:  PUSH  ACC
                PUSH  00H
                PUSH  07H
                ACALL  RESET
                MOV  A,#0BEH    ;Clock Burst Reg. (WR)
                ACALL  WRBITRTC
                MOV  R7,#8
NxtByteWR:      MOV  A,@R0
                ACALL  WRBITRTC
                INC  R0
                DJNZ  R7,NxtByteWR

```

```

ACALL      RESET
POP        07H
POP        00H
POP        ACC
RET

;-----
;READ BURST TO CLOCK REGISTER
;IN : R0 Point to first buffer addr
;OUT : 8 clock register data
;      start @ addr point by R0
;REMARK : Read start @bit 0 address 0
;-----
BURSTRDCLKREG:
    PUSH    ACC
    PUSH    00H
    PUSH    07H
    ACALL    RESET
    MOV      A,#0BFH      ;Clock Burst Reg. (RD)
    ACALL    WRBITRTC
    MOV      R7,#8
NxtByteRD: ACALL    RDBITRTC
    MOV      @R0,A
    INC      R0
    DJNZ     R7,NxtByteRD
    ACALL    RESET
    POP      07H
    POP      00H
    POP      ACC
    RET

;-----
;WRITE BURST TO RAM
;IN : R0 = point to first buffer addr
;      R7 = N data to be write
;Remark : - Prepare N RAM data
;          start @ addr point by R0
;          - Write start @bit 0 address 0
;-----
BURSTWRRAM:
    PUSH     ACC

```

```

PUSH      00H
PUSH      07H
ACALL     RESET
MOV       A,#0FEH      ;RAM Burst Reg. (WR)
ACALL     WRBITRTC
NxtRAMWR: MOV   A,@R0
          ACALL  WRBITRTC
          INC   R0
          DJNZ  R7,NxtRAMWR
          ACALL  RESET
          POP   07H
          POP   00H
          POP   ACC
          RET

```

```

;-----
;READ BURST FROM RAM
;IN : R0 = Point to first buffer addr
;    R7 = N data to be read
;OUT : N RAM data start @ addr point by R0
;REMARK : Read start @bit 0 address 0
;-----

```

```

BURSTRDRAM:  PUSH  ACC
              PUSH  00H
              PUSH  07H
              ACALL  RESET
              MOV   A,#0FFH
              ACALL  WRBITRTC
NxtRAMRD:    ACALL  RDBITRTC
              MOV   @R0,A
              INC   R0
              DJNZ  R7,NxtRAMRD
              ACALL  RESET
              POP   07H
              POP   00H
              POP   ACC
              RET

```

Sumber: <http://www.innovativeelectronics.com> (Application note 47)