

PERCOBAAN 14

REAL TIME CLOCK (RTC) 12C887

TUJUAN

1. Memahami tentang real time clock
2. Memahami rangkaian interface RTC dengan mikrokontroller
3. Memahami pembuatan program pengambilan data RTC

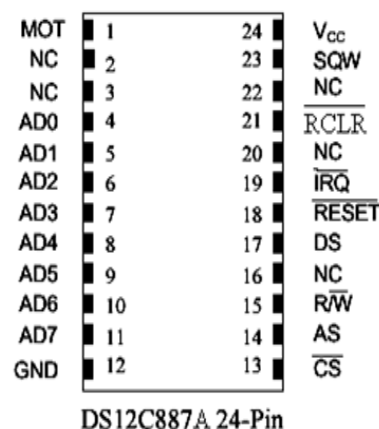
TEORI DASAR

Real Time Clock merupakan suatu chip (IC) yang memiliki fungsi sebagai penunjuk waktu. RTC DS 12C887 memiliki register yg dapat menunjukkan detik, menit, jam, tanggal, bulan dan tahun. RTC ini didesain memiliki 128 lokasi RAM yang terdiri dari 15 byte untuk data waktu serta kontrol, dan 113 byte sebagai RAM yang dapat digunakan sebagai RAM pada umumnya.

RTC DS 12C887 menggunakan bus yang termultipleks untuk menghemat pin. Timing yang digunakan untuk mengakses RTC dapat menggunakan intel timing atau motorola timing. RTC ini juga dilengkapi dengan pin IRQ untuk kemudahan dalam proses.

Operasi dari RTC

Berikut ini gambar pin-pin dari RTC DS 12C887, jumlah total pin-nya sebanyak 13 buah :



Gambar 14.1 Pin RTC DS 12C887

Berikut ini akan ditampilkan blok diagram dari RTC DALLAS 12C887. Terlihat dari blok diagram tersebut bahwa RTC terbagi menjadi beberapa bagian utama dengan kontrol maupun I/O untuk operasinya.

Tabel 14.1. berikut ini menjelaskan fungsi masing-masing pin yang ada :

Nama Pin	Keterangan
MOT	Motorola / Intel timing
AD0-AD7	Bus DATA / ADDRESS yang termultipleks
CS	Chip Select
AS	Address Strobe
R/W	Write Strobe
DS	Data Strobe
RESET	Inputan Reset
IRQ	Interrupt Request Output
SQW	Output Sinyal Kotak
VCC	Catu tegangan +5 volt
Gnd	Ground
RCLR	Hapus RAM

Penjelasan dari masing-masing pin RTC DS12C887 dijelaskan berikut :

1. GND, VCC

Merupakan pin untuk power supply RTC. VCC diberikan input +5volt. Ketika tegangan 5 volt diberikan dalam batas yang normal, RTC bebas diakses dan data dapat ditulis atau dibaca. Sedangkan ketika Vcc dibawah 4.25 volt, maka proses baca dan tulis tidak dapat dilakukan. Akan tetapi, proses pewaktu (timekeeping) tetap dilakukan oleh RTC tanpa dipengaruhi oleh tegangan input yang rendah. Saat VCC turun dibawah 3 volt, catu daya untuk RAM dan pewaktu dipindahkan ke internal lithium battery yang terdapat pada RTC. Fungsi pewaktu dari RTC memiliki akurasi ± 1 menit per bulan pada suhu 250 C walaupun tanpa adanya catu pada VCC.

2. MOT (Mode Select)

Pin MOT memberikan kebebasan untuk memilih dua tipe bus. Ketika pin ini dihubungkan dengan VCC, maka motorolla bus timing yang dipilih, sedangkan ketika dihubungkan ke

ground atau dibiarkan (tak dihubungkan) maka intel bus timing yang dipilih. Pin ini memiliki pull down internal sebesar sekitar 20 k Ω .

3. SQW (Square Wave Output)

Pin SQW dapat mengeluarkan sinyal kotak, dari satu sampai 13 pembagian yang disediakan oleh 15 buah tingkat pembagi (divider) internal dari RTC. Frekuensi dari SQW dapat dipilih dengan memprogram register A seperti pada tabel. Sinyal SQW dapat dihidupkan dan dimatikan dengan mengatur bit SQWE pada register B. Sinyal SQW tidak hidup ketika VCC dibawah 4,25 volt.

4. AD0 – AD7 (Address/Data bus yang ter-multipleks dan bidirect)

Bus yang ter-multiplex menghemat penggunaan pin karena informasi address dan data waktu menggunakan jalur sinyal yang sama. Alamat muncul pada bagian pertama dari bus cycle dan dengan pin yang sama pula digunakan untuk data pada bagian kedua dari bus cycle. Multipleksing antara data dan address tidak memperlambat waktu akses dari RTC karena perubahan dari address menuju data dilakukan pada saat akses RAM internal pada RTC. Address harus valid saat terjadi falling edge pada AS/ALE, yang kemudian RTC akan me-latch address dari AD0 sampai AD6. Data valid yang akan dituliskan ke RTC harus valid dan dijaga stabil pada saat pin DS dan WR diberi pulsa. Pada proses pembacaan, RTC menghasilkan output 8 bit data pada saat pin DS dan RD diberi pulsa. Bus akan menjadi high impedance saat pulsa low diberikan pada pin DS (motorola) atau pulsa high diberikan pada pin RD (intel).

5. AS (Address Strobe Input)

Perubahan pulsa positif pada address strobe berguna untuk melakukan multipleks pada bus. Perubahan pulsa negatif pada AS akan menyebabkan address ter-latch. Perubahan positif selanjutnya akan menghapus address walaupun CS tidak diaktifkan. Perintah access seharusnya dikirimkan secara bersamaan.

6. DS (Data Strobe or Read Input)

Pin DS/RD mempunyai 2 mode operasi tergantung pada tingkat yang diberikan kepada MOT. Ketika MOT dihubungkan ke VCC, motorola bus timing dipilih dan pada mode ini DS merupakan pulsa positif saat bagian lanjut dari bus cycle dan disebut sebagai data strobe. Saat proses pembacaan, DS menunjukkan bahwa RTC memiliki bus yang

bidirectional. Pada proses penulisan, pulsa positif pada DS akan me-latch data yang ditulis. Ketika MOT dihubungkan ke ground, maka intel bus timing yang dipilih dan pin DS disebut sebagai read (RD). RD menjelaskan periode ketika RTC memperlakukan bus sebagai pembaca data. Sinyal RD mempunyai pengertian yang sama seperti sinyal Output Enable (OE) pada memori yang umum.

7. R/W (Read/Write Input)

Pin R/W juga memiliki 2 mode operasi. Ketika pin MOT dihubungkan ke VCC untuk motorola timing, R/W berada pada level yang menunjukkan apakah berada pada proses baca atau tulis. Proses baca ditunjukkan dengan level high pada R/W sementara DS juga high. Proses tulis ditunjukkan saat R/W low dan DS high. Ketika pin MOT dihubungkan ke ground untuk intel timing, sinyal R/W adalah sinyal active low dinamakan WR. Pada mode ini pin R/W mempunyai makna yang sama dengan sinyal Write Enable (WE) seperti pada RAM umum

8. CS (Chip Select Input)

Sinyal chip select harus di hubungkan low untuk dapat mengakses bus cycle. CS harus dijaga pada posisi aktif saat DS dan AS pada motorola timing atau RD dan WR untuk intel timing. Bus cycle yang terjadi tanpa mengaktifkan CS akan me-latch address tapi tidak ada akses yang bisa dilakukan. Saat VCC dibawah 4,25 volt, RTC secara internal menghalangi akses dengan cara secara internal tidak mengaktifkan input CS. Proses ini akan melindungi baik data RTC maupun data pada RAM saat tidak ada catu daya.

9. IRQ (Interrupt Request Output)

Pin IRQ adalah keluaran active low dari RTC 12C887 yang digunakan sebagai masukan interupsi pada prosessor. Keluaran IRQ tetap low selama bit status menyebabkan interupsi terjadi dan bit yang berhubungan yaitu interrupt-enable di-set. Untuk menghapus interupsi biasanya dilakukan dengan membaca register C. Pin RESET juga menghapus interupsi yang dilakukan atau yang belum terjadi (pending). Ketika tidak terjadi interrupt maka level IRQ berada pada kondisi high impedance. Beberapa peralatan interupsi dapat dihubungkan dengan IRQ bus. Bus IRQ adalah keluaran open drain dan memerlukan tambahan resistor pull-up.

10. RESET (Reset Input)

Pin RESET tidak berpengaruh terhadap jam, kalender atau RAM. Pada saat penyalaan RTC, pin RESET dapat dijaga LOW untuk beberapa saat untuk menstabilkan catu daya. Lamanya waktu pin RESET diberikan low tergantung dari aplikasi yang digunakan. Akan tetapi jika reset digunakan saat power up (penyalaan), waktu RESET dijaga low harus melebihi 200 ms untuk meyakinkan bahwa timer internal dari RTC untuk penyalaan telah cukup. Ketika RESET low dan VCC diatas 4,25 volt, maka hal-hal yang terjadi adalah sebagai berikut :

- a. Periodic Interrupt Enable (PEI) dihapus menjadi nol.
- b. Alarm Interrupt Enable (AIE) dihapus menjadi nol.
- c. Update Ended Interrupt Flag (UF) dihapus menjadi nol.
- d. Interrupt Request Status Flag (IRQF) dihapus menjadi nol.
- e. Periodic Interrupt Flag (PF) dihapus menjadi nol.
- f. RTC tidak dapat diakses sampai RESET menjadi high.
- g. Alarm Interrupt Flag (AF) dihapus menjadi nol.
- h. Pin IRQ dalam kondisi high impedance.
- i. Square Wave Output Enable (SQWE) dihapus menjadi nol.
- j. Update Ended Interrupt Enable (UIE) dihapus menjadi nol.

Pada aplikasi umum, RESET dapat dihubungkan langsung VCC. Koneksi ini akan menjadikan RTC 12C887 dapat berfungsi saat terjadi kerusakan catu daya tanpa mempengaruhi register kontrol RTC.

Address Map dari RTC

RTC mempunyai RAM internal didalamnya, dalam memori tersebut terbagi menjadi beberapa register-register yang mempunyai fungsi dalam pengontrolan maupun pengambilan data RTC. Agar tidak terjadi kesalahan dalam pengontrolan maupun pengambilan data, peta alamat dari masing-masing register harus jelas

Address map dari RTC terdiri dari 113 byte RAM yg dapat digunakan oleh pengguna RTC, 11 byte RAM yang berisikan data RTC berupa waktu, kalender dan alarm, dan 4 byte yang digunakan untuk kontrol dan status. Keseluruhan 128 byte ini dapat ditulis atau dibaca secara langsung kecuali pada keadaan berikut:

1. Register C dan D adalah read-only.
2. Bit ke-7 dari register A adalah read-only.
3. Bit atas dari byte detik adalah read-only.

DEC	HEX			
0	00		SECONDS	0
			SECONDS ALARM	1
			MINUTES	2
13	0D		MINUTES ALARM	3
14	0E		HOURS	4
			HOURS ALARM	5
			DAY OF THE WEEK	6
49	31		DAY OF THE MONTH	7
50	32		MONTH	8
51	33		YEAR	9
			REGISTER A	10
			REGISTER B	11
			REGISTER C	12
			REGISTER D	13
127	7F		CENTURY	50

Gambar 14.2. Peta Alamat RTC

Waktu, Kalender dan Alarm

Informasi waktu dan kalender diperoleh dengan membaca byte memori yang sesuai. Waktu, kalender dan alarm di-set atau diinisialisasi dengan menulis ke byte RAM yang sesuai. Isi dari kesepuluh data kalender, waktu dan alarm dapat digunakan dalam mode biner atau BCD (Binary-Coded Decimal). Sebelum mengakses register kalender, waktu dan alarm, SET bit pada register B harus ditulis dengan level logika yang sesuai agar tidak mengganggu proses update yang dilakukan oleh RTC. Kesemua data kalender, waktu dan alarm harus menggunakan mode data yang sama (biner atau BCD), yaitu dengan cara menulis pada register B pada bit DM(Data Mode). Sekali di inisialisasi maka RTC akan terus menggunakan mode data yang dipilih sebelumnya. Tabel berikut menunjukkan mode data dari kalender, waktu dan alarm baik dalam mode biner atau BCD.

Tabel 14.2. Tabel data mode waktu, kalender dan alarm

TIME, CALENDAR AND ALARM DATA MODES

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

Register Kontrol RTC

RTC 12C887A mempunyai 4 buah register kontrol yang dapat diakses setiap saat, walaupun pada saat RTC melakukan update cycle.

Register A

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

UIP – Update in progress adalah status bit yang dapat dimonitor, merupakan bit read only dan tidak dipengaruhi oleh reset. Ketika bit UIP 1, proses update akan segera berlangsung. Ketika UIP 0, proses update tidak akan terjadi sekurangnya 244?s. Informasi kalender, waktu dan alarm yang ada pada RAM akan dapat diakses secara penuh ketika bit UIP nol. DV2, DV1, DV0 – Ketiga bit ini digunakan untuk menghidupkan atau mematikan osilator dan me-reset rantai perhitungan. Pola 010 adalah satu-satunya kombinasi yang akan menghidupkan osilator dan menyebabkan RTC dapat menyimpan waktu. Pola 11x akan menghidupkan osilator tapi menahan rantai perhitungan dalam kondisi reset.

RS2, RS2, RS1, RS0 – Keempat bit ini akan memilih frekuensi atau mematikan pulsa kotak pada pin SQW. Pin SQW dapat berfungsi juga sebagai periodic interrupt. Keempat bit ini dapat dibaca/tulis dan tidak dipengaruhi oleh RESET. Untuk mengaktifkannya, dapat dilakukan sebagai berikut :

1. Mengaktifkan interrupt dengan bit PIE.
2. Mengaktifkan keluaran SQW dengan bit SQWE.
3. Mengaktifkan keduanya dengan waktu dan frekuensi yang sama.
4. Tidak mengaktifkan keduanya.

Register B

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

SET – Ketika bit SET adalah 0, proses update berfungsi normal dengan menambahkan perhitungan sekali dalam satu detik. Ketika SET ditulis menjadi 1, semua proses update dihentikan sehingga program dapat menginisialisasi byte waktu dan kalender tanpa adanya proses update yang terjadi ditengah proses inisialisasi.

PIE – Periodic Interrupt Enable merupakan bit yang dapat dibaca/tulis yang akan menyebabkan bit Periodic Interrupt Flag (PF) pada register C mengatur pin IRQ menjadi low. Ketika bit PIE diset 1, interrupt secara periodik dihasilkan dengan mengatur pin IRQ menjadi low dengan periode tertentu sesuai dengan bit RS3-RS0 (register A). Apabila PIE 0, ini akan menghalangi pengaturan pin IRQ oleh interrupt periodik, tetapi bit PF masih diset dengan periode tertentu. PIE tidak dapat dirubah secara internal oleh RTC, tetapi akan dihapus menjadi 0 pada saat RESET.

UIE – Update Ended Interrupt merupakan bit yang dapat dibaca/tulis yang akan meng-enable Update ended Flag (UF) pada register C untuk memasukkan IRQ. Pin RESET yang di “low” atau bit SET yang di “high” akan menghapus bit UIE.

SQWE – Ketika bit Square Wave Enable di-set menjadi 1, gelombang kotak dengan periode yang sudah ditentukan oleh bit RS3-RS0 akan dikeluarkan pada pin SQW. Ketika SQWE di-set menjadi 0, maka pin SQW tetap low. Bit SQWE dapat dibaca/tulis dan dapat dihapus oleh RESET. SQWE di-set menjadi 1 ketika Vcc dihidupkan.

DM – Bit Data Mode (DM) mengindikasikan apakah informasi waktu dan kalender mempunyai format biner atau BCD. Bit DM di-set oleh program dengan memberikan

format yang sesuai dan dapat dibaca sesuai keperluan. Bit ini tidak dapat dirubah secara internal atau oleh RESET. ‘1’ pada DM menyatakan data biner sedangkan ‘0’ menyatakan DM mempunyai format BCD (Binary Coded Decimal).

24/12 – Bit ini digunakan untuk memilih mode jam, 24 atau 12. ‘1’ menunjukkan mode 24-jam dan ‘0’ menunjukkan mode 12-jam. Bit ini dapat dibaca/tulis dan tidak dipengaruhi secara internal atau RESET.

DSE – Merupakan bit khusus yang digunakan untuk merubah waktu ketika terjadi perubahan musim (biasanya terjadi pada negara dengan 4 musim). Ketika di-set 1, pada hari minggu pertama pada bulan april terjadi perubahan jam dari 1:59:59 AM menjadi 3:00:00 AM. Pada hari minggu terakhir bulan oktober saat waktu menunjukkan 1:59:59 AM berubah menjadi 1:00:00 AM. Hal ini tidak terjadi ketika DSE di-set 0. Bit ini tidak dipengaruhi secara internal atau RESET.

Register C

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

IRQF – Interrupt Request Flag (IRQF) bit diset menjadi 1 ketika hal berikut terjadi :

PF = PIE = 1

AF = AIE = 1

UF = UIE = 1

Contoh: $IRQF = (PF * PIE) + (AF * AIE) + (UF * UIE)$

Saat ketika bit IRQF adalah 1, pin IRQ menjadi “low”. Bit flag PF, AF, dan UF di hapus setelah register C dibaca oleh program atau ketika pin RESET “low”.

PF – Periodic Interrupt Flag adalah bit read-only yang diset menjadi 1 ketika edge dideteksi pada tap yang dipilih pada rantai pembagian. Bit RS3 – RS0 menghasilkan frekuensi periodik. PF diset menjadi 1 tanpa dipengaruhi oleh bit PIE. Ketika PF dan PIE ‘1’ , sinyal IRQ aktif dan akan mengeset bit IRQF. Bit PF akan dihapus secara pembacaan software pada register C atau RESET.

AF – ‘1’ pada bit Alarm Interrupt Flag (AF) mengindikasikan waktu yang dicapai sesuai dengan waktu alarm. Jika bit AIE juga 1, pin IRQ akan menjadi low dan ‘1’ akan terjadi pada bit IRQF. RESET atau pembacaan register C akan menghapus AF.

UF – Bit Update Ended Interrupt Flag akan diset setelah tiap proses update. Ketika bit UIE diset menjadi 1, ‘1’ pada UF akan menyebabkan bit IRQF menjadi 1 dan akan mengaktifkan pin IRQ. UF dihapus dengan membaca register C atau RESET.

BIT 3 sampai BIT 0 – Bit ini tidak dipakai dan selalu bernilai 0.

Register D

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

VRT – Valid RAM Time mengindikasikan kondisi dari baterai yang dihubungkan ke pin VBAT. Bit ini tidak dapat ditulis dan selalu menjadi 1 ketika dibaca. ‘0’ akan terjadi ketika baterai internal telah rusak. Bit ini tidak dipengaruhi oleh RESET.

BIT 6 sampai BIT 0 - Bit ini tidak dipakai dan selalu bernilai 0.

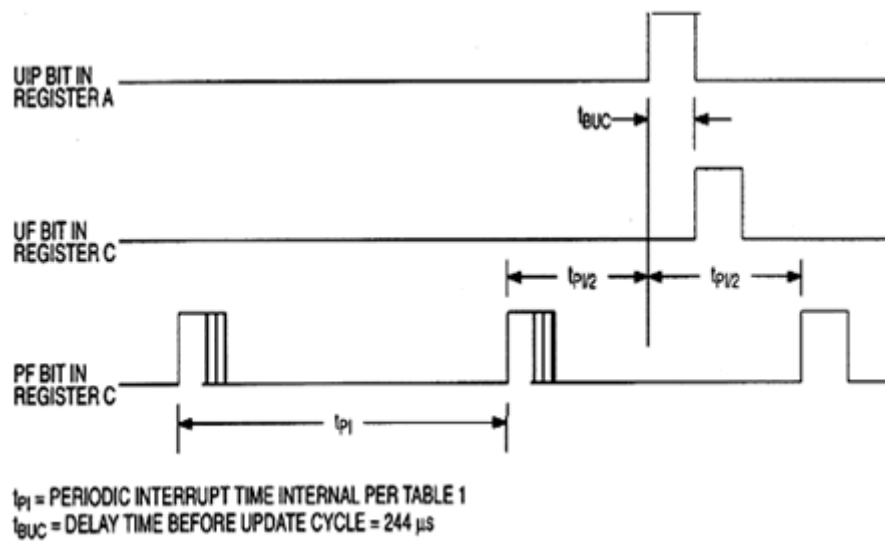
Proses Update

RTC 12C887 melaksanakan proses update sekali dalam satu detik walau tanpa memperhatikan bit SET pada register B. Ketika bit SET pada register B di-set menjadi 1, maka informasi waktu, kalender, dan alarm akan berhenti dan tidak akan berubah pada saat terjadi penambahan waktu, tetapi rantai perhitungan waktu akan tetap terjadi dan disimpan pada buffer internal. Hal ini akan menjaga akurasi waktu dari RTC saat terjadi proses pembacaan dan penulisan. Proses update ini juga akan membandingkan byte alarm dengan waktu yang terjadi saat itu.

Ada tiga metode untuk mengakses data pada RTC yang akan menghindari akses informasi waktu yang tidak konsisten. Metode pertama adalah menggunakan update ended interrupt. Jika di enable, interrupt akan terjadi tiap proses update selesai, dimana terdapat waktu 999 ms untuk membaca data waktu yang valid. Jika interupsi ini digunakan, bit IRQF pada register C harus dihapus sebelum meninggalkan proses interupsi.

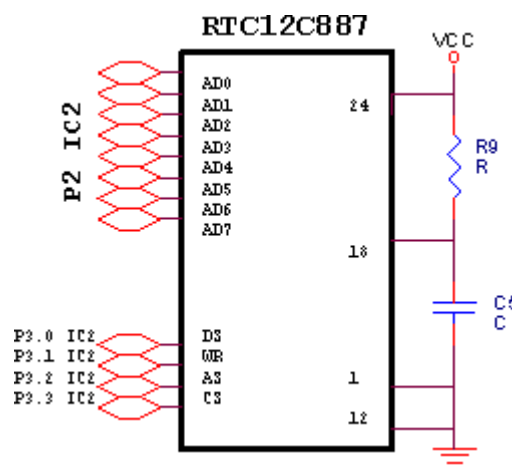
Metode yang kedua adalah dengan menggunakan bit update in progress (UIP) yang terdapat pada register A yang menyatakan apakah proses update sedang berlangsung. Bit UIP akan mengeluarkan pulsa tiap satu detik. Setelah bit UIP menjadi high, maka proses update akan terjadi 244 μ s kemudian dan jika low terbaca pada bit UIP, maka pengguna mempunyai waktu 244 μ s sebelum data waktu/kalender akan berubah. Sehingga,

pengguna harus menghindari proses interupsi yang menyebabkan waktu pembacaan data melebihi 244 μ s.



Gambar 14.3 Hubungan antara Update-Ended dan Periodic Interrupt

Metode yang ketiga adalah menggunakan periodic interrupt untuk menunjukkan apakah proses update sedang berlangsung. Bit UIP pada register A akan di-set high diantara pengaktifan dari bit PF pada register C. Periode interrupt yang melebihi waktu t_{BUC} akan menyebabkan data yang valid dapat dicapai pada setiap terjadinya interupsi. Pembacaan harus telah selesai dalam waktu $(t_{PI}/2 + t_{BUC})$ untuk memastikan data tidak dibaca saat proses update.

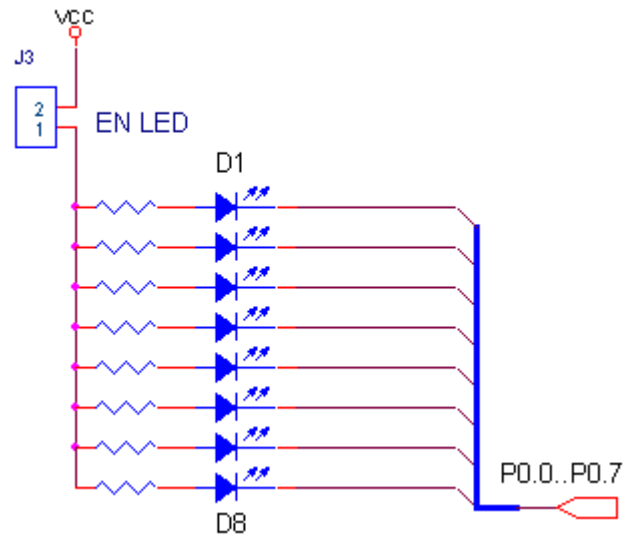


Gambar 14.6 Rangkaian interface RTC12c887

Percobaan 14.1.

Pengambilan data detik pada RTC DS12887 dengan display LED

Pada percobaan ini, data detik RTC diambil dan dikeluarkan ke LED.



Gambar 14.7. Rangkaian interface ke LED

1. Hubungkan jumper LED_EN
2. Hubungkan kabel data dan kontrol pada board RTC
3. Hubungkan modul Microcontroller Trainer dengan power supply +5V
4. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
5. Buka Program M-IDE Studio for MCS-51, sebagai editor dan compiler program
6. Ketik program berikut ini:

```
CS bit P3.3
ALE bit P3.2
Write bit P3.1
Read bit P3.0
PortData equ P0
RegisterA equ 10
RegisterB equ 11
;
org 0h
clr CS ; Aktivkan RTC
call INIT_RTC
start: mov PortData,#00h ; 00h = alamat detik pada RTC
      clr ALE ; Penguncian alamat 0-1-0
      setb ALE
      clr ALE
      ;
```

```

        setb Write ; sinyal write = 1
        ;
        setb Read
        clr Read ; sinyal read = 0
        mov A,PortData ; Baca data dari alamat 00h = detik
        setb Read
        ;
        mov P0,A ; simulasi data ke LED = P0
        sjmp start
        ;
INIT_RTC:
        setb Read
        mov PortData,#RegisterA ;
        clr ALE
        setb ALE
        clr ALE
        ;
        setb WRite
        clr Write
        mov PortData,#20h
        setb WRite
        ;
        mov PortData,#RegisterB
        clr ALE
        setb ALE
        clr ALE
        ;
        setb WRite
        clr WRite
        mov PortData,#02h
        setb WRite
        ;
        RET
end

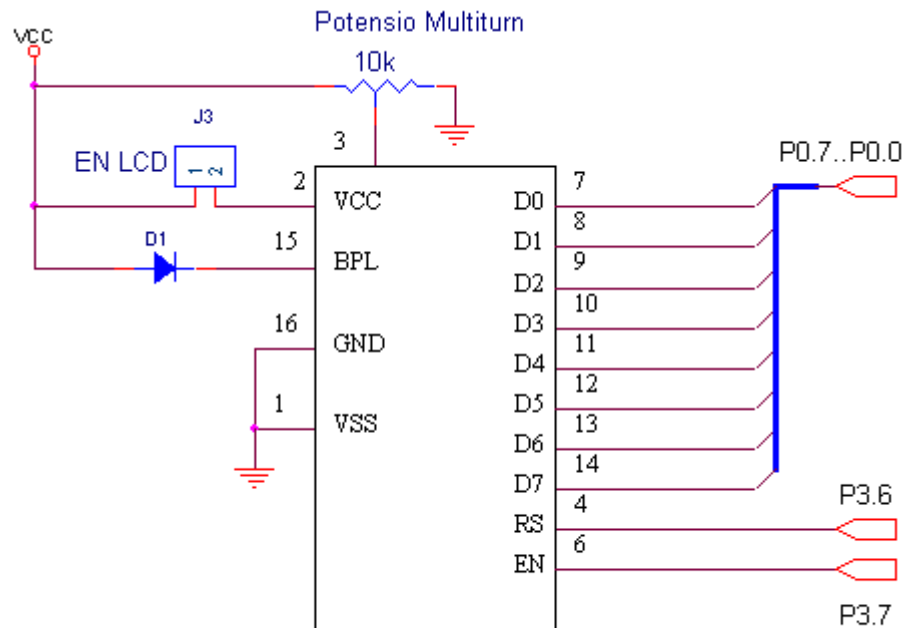
```

7. Simpanlah program yang anda ketik dan beri nama : prog141a.asm
8. Pada program MIDE tersebut pilih Build /F9 atau untuk melakukan kompilasi program dari *.asm ke *.hex.
9. Lakukan pemrograman mikrokontroller dengan menggunakan Program ISP Software (Lihat Petunjuk Penggunaan)

Percobaan 14.2.

Pengambilan data detik dan menit pada RTC12887 dengan display LCD Karakter 2x 16.

Pada percobaan ini, data detik dan menit diambil dan ditampilkan pada LCD Karakter 2x16.



LCD Character 2 X 16

Gambar 14.8 Interface LCD Carácter 2x16

1. Hubungkan jumper LCD_EN
2. Hubungkan kabel data dan kontrol pada board RTC ke rangkaian mikrokontroller
3. Hubungkan modul Microcontroller Trainer dengan power supply +5V
4. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
5. Buka Program M-IDE Studio for MCS-51, sebagai editor dan compiler program
6. Ketik program berikut ini:

```
CS bit P3.3
ALE bit P3.2
Write bit P3.1
Read bit P3.0
PortRTC equ P2
RegisterA equ 10
RegisterB equ 11
Puluhan equ 30h
Satuan equ 31h
detik equ 32h
Menit equ 33h
```

```

        org 0h
call init_LCD
call init_RTC
; call setDetik
; call setMenit
call write_char
start:
    call readDetik
    call bin2Dec
    call write2LCDDetik
    ;
    call readMenit
    call bin2Dec
    call write2LCDMenit
    ;
    sjmp start
    ;
write_char:
    mov dptr,#word1 ;DPTR = [ address word1 ]
    mov r3,#16 ;R3=16,number character to be display
    mov r1,#80h ;R1=80h,address DDRAM start position
    acall write_inst
    ;
write1:clr a ; A = 0
    movc a,@a+dptr ; A = [A+ DPTR]
    mov r1,A ; R1 = A
    inc dptr ; DPTR = DPTR +1
    acall write_data;
    djnz r3,write1 ; R3 = R3-1,
    ret
    ;
Init_lcd:
    mov r1,#00000001b ;Display clear
    call write_inst ;
    mov r1,#00111000b;Function set,Data 8 bit,2linefont 5x7
    call write_inst ;
    mov r1,#00001100b ;Display on,cursor off,cursorblink off
    call write_inst
    mov r1,#00000110b ;Entry mode, Set increment
    call write_inst
    ret
    ;
Write_inst:
    clr P3.6 ; P3.6 = RS =0
    mov P0,R1 ; P0 = D7 s/d D0 = R1
    setb P3.7 ; P3.7 =EN = 1
    call delay ; call delay time
    clr P3.7 ; P3.7 =EN = 0
    ret
    ;
Write_data:
    setb P3.6 ; P3.6 = RS =1
    mov P0,R1 ; P0 = D7 s/d D0 = R1
    setb P3.7 ; P3.7 =EN = 1
    call delay ; call delay time
    clr p3.7 ; P3.7 =EN = 0

```

```

        ret
    ;
delay: mov R0, #0
    delay1: mov R7, #0fh
        djnz R7, $
        djnz R0, delay1
        ret
    ;
ReadDetik:
    mov PortRTC, #00h ; 00h = alamat detik pada RTC
    clr ALE ; Penguncian alamat 0-1-0
    setb ALE
    clr ALE
    setb Write ; sinyal write = 1
    ;
    setb Read
    clr Read ; sinyal read = 0
    mov A, PortRTC ; Baca data dari alamat 00h = detik
    setb Read
    ret
    ;
ReadMenit:
    mov PortRTC, #02h ; 00h = alamat detik pada RTC
    clr ALE ; Penguncian alamat 0-1-0
    setb ALE
    clr ALE
    setb Write ; sinyal write = 1
    ;
    setb Read
    clr Read ; sinyal read = 0
    mov A, PortRTC ; Baca data dari alamat 00h = detik
    setb Read
    ret
    ;
init_RTC:
    clr CS
    setb Read
    mov PortRTC, #RegisterA ;
    clr ALE
    setb ALE
    clr ALE
    ;
    setb Write
    clr Write
    mov PortRTC, #20h
    setb Write
    ;
    mov PortRTC, #RegisterB
    clr ALE
    setb ALE
    clr ALE
    ;
    setb Write
    clr Write
    mov PortRTC, #06h
    setb Write

```

```

        RET
        ;
Bin2Dec:
        mov b,#10d
        div ab
        mov puluhan,a
        mov satuan,b
        ret
        ;
Write2LCDDetik:
        mov r1,#0c4h
        call write_inst
        mov a,Puluhan
        add a,#30h
        mov r1,a
        call write_data
        ;
        mov r1,#0c5h
        call write_inst
        mov a,Satuan
        add a,#30h
        mov r1,a
        call write_data
        ret
        ;
Write2LCDMenit:
        mov r1,#0c2h
        call write_inst
        mov a,Puluhan
        add a,#30h
        mov r1,a
        call write_data
        ;
        mov r1,#0c3h
        call write_inst
        mov a,Satuan
        add a,#30h
        mov r1,a
        call write_data
        ret
        ;
setMenit:
        setb Read
        mov PortRTC,#02h;
        clr ALE
        setb ALE
        clr ALE
        ;
        setb WRite
        clr Write
        mov PortRTC,#21d
        setb WRite
        ret
        ;
setDetik:
        setb Read

```

```

        mov PortRTC,#00h;
        clr ALE
        setb ALE
        clr ALE
        ;
        setb WRite
        clr Write
        mov PortRTC,#00h
        setb WRite
        ret
word1: DB ' My RTC Test '; here is the data to be look up
        end

```

7. Simpanlah program yang anda ketik dan beri nama : prog142a.asm
8. Pada program MIDE tersebut pilih Build /F9 atau untuk melakukan kompilasi program dari *.asm ke *.hex.
9. Lakukan pemrograman mikrokontroller dengan menggunakan Program ISP Software (Lihat Petunjuk Penggunaan)
10. Lakukan modifikasi pada program tersebut untuk memunculkan data jam, hari, bulan dan tahun pada LCD Karakter
11. Lakukan modifikasi pada program tersebut untuk menampilkan data detik dan menit pada rangkaian 7 segmen