

Real Time Clock Menggunakan I2C Bus pada Modul DST-52

Jika pada umumnya IC Real Time Clock menggunakan jalur data paralel maka pada aplikasi ini akan dicontohkan penggunaan IC Real Time Clock menggunakan jalur data serial I2C bus. Kelebihan dari IC RTC yang menggunakan jalur data paralel adalah sistem I2C bus membutuhkan jalur data jauh lebih sedikit dari pada sistem paralel bus dengan kemampuan yang relatif sama dengan yang menggunakan jalur data paralel.

DS1307 adalah IC serial Real Time Clock (RTC) dimana alamat dan data ditransmisikan secara serial melalui sebuah jalur data dua arah I2C. Karena menggunakan Jalur data I2C maka hanya memerlukan dua buah pin saja untuk komunikasi. Yaitu pin untuk data dan pin untuk sinyal clock. Sistem jalur data I2C adalah suatu standar protokol sistem komunikasi data serial yang dikembangkan oleh Philips dan cukup populer karena penggunaannya cukup mudah.

Pada dasarnya, pada sistem I2C terbagi atas dua bagian, yaitu suatu device yang bertindak sebagai pengontrol atau *Master* dan suatu device yang dikontrol atau *Slave*. Master dan Slave saling berkomunikasi melalui jalur data bus I2C. Alat yang mengendalikan komunikasi data disebut Master dan alat yang dikendalikan oleh Master dikenal sebagai Slave. Pada satu jalur data I2C yang sama dapat terdapat slave lebih dari satu oleh karena itu I2C Bus harus dikendalikan Master yang dapat membangkitkan serial clock (SCL), mengontrol sistem komunikasi data (SDA), dan dapat menghasilkan kondisi-kondisi "START" dan "STOP". Pada hal ini DS1307 beroperasi sebagai slave pada I2C bus. Contoh bagaimana data ditransfer pada jalur data I2C adalah seperti pada gambar 1. Pada jalur data bus I2C hanya terdapat 2 buah jalur yang digunakan yaitu Clock (SCL) dan Data (SDA)

Terdapat beberapa macam jenis kondisi pada jalur data I2C, jenis kondisi tersebut adalah:

1. **Bus not busy:** Jalur data (SDA) dan clock (SCL) berlogika high
2. **Start data transfer:** Suatu perubahan kondisi pada jalur data, dari logika high ke logika low, ketika jalur data sedang berlogika high, menandakan kondisi START
3. **Stop data transfer:** Suatu perubahan kondisi pada jalur data, dari logika low ke logika high, ketika jalur data sedang berlogika high, menandakan kondisi STOP

4. **Data valid:** Suatu kondisi ketika jalur data menandakan data valid, yaitu ketika setelah kondisi START, jalur data tetap stabil selama periode high sinyal clock. Data pada jalur data harus berubah selama periode LOW dari sinyal clock. Terdapat satu pulsa clock untuk setiap bit data. Setiap proses pengiriman data dimulai dengan kondisi START dan diakhiri dengan kondisi STOP. Banyaknya jumlah byte data yang ditransfer diantara kondisi START dan STOP tersebut tidak terbatas, dan diatur oleh Master.

5. **Acknowledge:** Setiap device yang dituju telah menerima data dengan benar akan membangkitkan kondisi Acknowledge setiap menerima byte data. Device yang membangkitkan Acknowledge harus membangkitkan logika low pada jalur SDA selama sebuah pulsa clock. Untuk mengakhiri suatu proses pengiriman data Master harus memberikan suatu tanda dengan tidak memberikan tanda acknowledge melainkan memberikan tanda STOP pada slave.

Pada sistem jalur data I2C, terdapat dua tipe arah proses pengiriman data yaitu:

1. **Data transfer dari master menuju slave.** Byte pertama yang dikirimkan oleh master menuju slave adalah alamat slave. Lalu selanjutnya adalah byte-byte data.

Slave membalas dengan bit acknowledge setiap berhasil menerima 1 byte data.

2. **Data transfer dari slave menuju master.** Byte pertama (alamat slave) dikirimkan oleh master. Kemudian slave yang mempunyai alamat yang dituju oleh master membalas dengan bit acknowledge. Lalu diikuti dengan proses pengiriman byte-byte data dari slave menuju master. Master membalas dengan mengirimkan bit acknowledge setiap berhasil menerima 1 byte. Untuk mengakhiri proses pengiriman data master membalas dengan mengirimkan bit “not acknowledge” kepada slave.

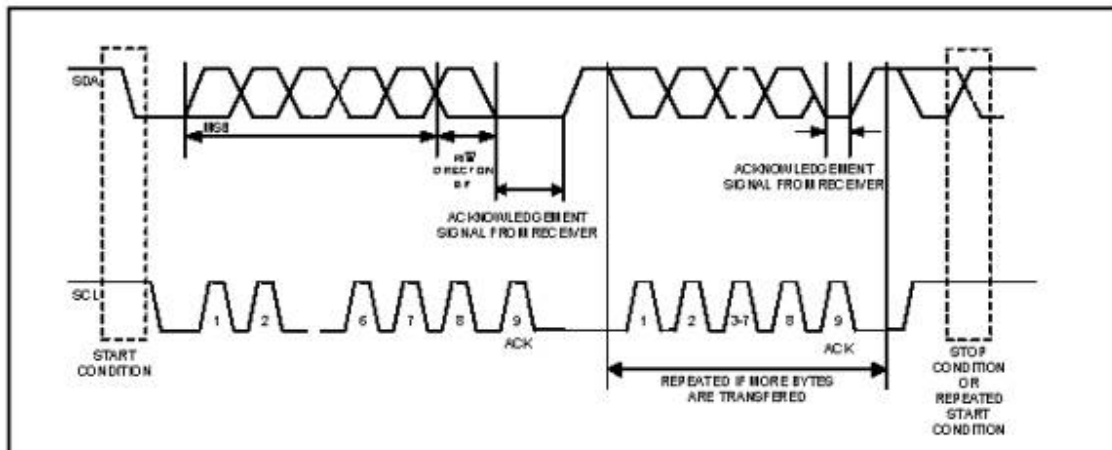
Pada aplikasi ini DS1307 bekerja dengan dua mode, yaitu:

1. **Mode Slave Penerima (Master Menulis Pada Slave):** Data serial dan clock diterima melalui SDA dan SCL. Setiap menerima byte data DS1307 akan merespon dengan membangkitkan bit acknowledge. Untuk mengawali proses pengiriman data dari master menuju slave diawali dengan kondisi START dan diakhiri dengan kondisi STOP. Setiap slave akan membaca alamat yang dituju oleh master dan memeriksa apakah alamat tersebut sama dengan alamat Slave tersebut, seperti yang terdapat pada gambar 2. Byte alamat slave adalah byte pertama yang diterima slave setelah master membangkitkan kondisi START. Byte alamat terdiri dari 7 bit data, untuk DS1307 byte alamat tersebut adalah 1101000b, dan diikuti oleh bit arah (R/W), yang mana untuk penulisan data ke slave adalah 0. Setelah menerima dan menganalisa byte alamat, DS1307 membangkitkan tanda acknowledge pada jalur SDA. Kemudian master akan mengirimkan sebuah data word alamat pada DS1307 untuk mengeset register pointer pada DS1307.

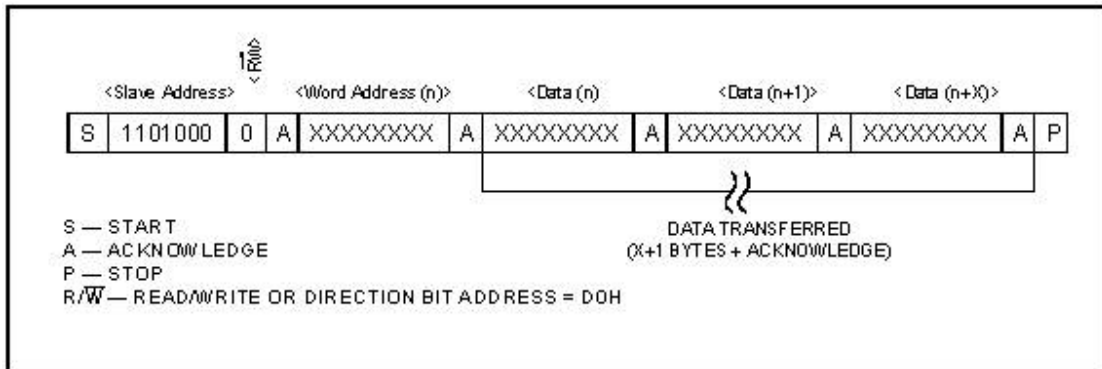
Setelah itu Master dapat mengakhiri proses pengiriman data ataupun melanjutkannya dengan mengirimkan byte data pada DS1307. Register pointer akan bertambah nilainya secara otomatis setiap terjadi proses penulisan data. Untuk mengakhiri proses pengiriman data master membangkitkan kondisi STOP.

2. Mode Slave Pengirim (Master Membaca Dari Slave): Byte pertama diterima dan diolah oleh slave seperti pada mode penerima, tetapi bit arah bernilai 1. DS1307 mengirimkan data serial pada SDA ketika menerima sinyal clock pada SCL. Untuk memulai proses pengiriman data diawali dengan kondisi START dan diakhiri dengan kondisi STOP, seperti yang terdapat pada gambar 6. Byte yang berisi data alamat diterima setelah master membangkitkan kondisi START. Byte alamat DS1307 terdiri dari 7-bit alamat dan 1 bit arah. 7-bit alamat tersebut adalah 1101000 dan bit arah tersebut (R/W) adalah 0 untuk *read*. Setelah menerima dan mengolah data alamat, DS1307 akan membalas dengan membangkitkan bit acknowledge pada SDA. Lalu kemudian DS1307 mulai mengirimkan data dimulai dari data yang terdapat pada alamat yang ditunjuk oleh register pointer. Nilai register pointer secara otomatis akan bertambah setiap terjadi proses pembacaan 1 byte data. Untuk mengakhiri proses pengiriman data maka master harus mengirimkan tadan “not acknowledge” kepada slave.

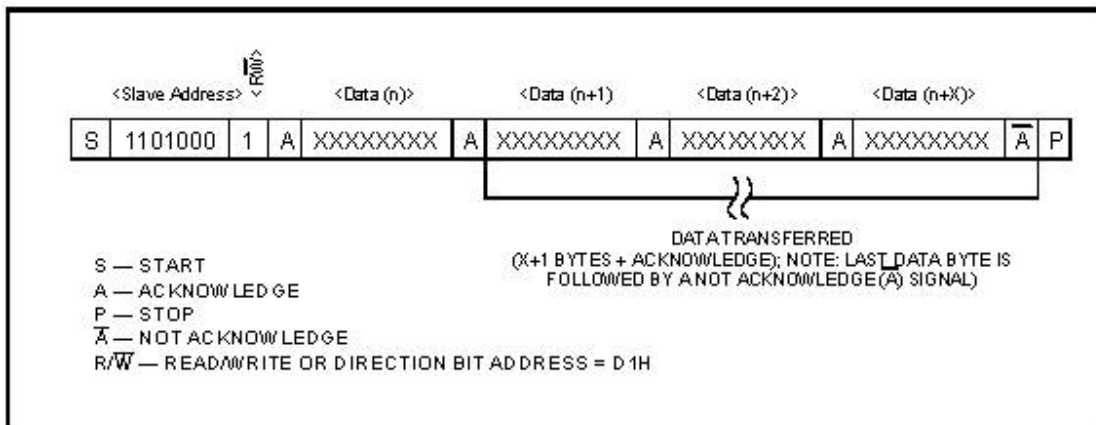
Untuk dapat mengambil nilai waktu dan tanggal maka master harus melakukan proses pembacaan data (Read) pada slave (DS1307), dengan alamat register sesuai dengan tabel 1. Setiap nilai-nilai waktu atau tanggal disimpan pada register yang mempunyai alamat yang berbeda-beda, misalnya untuk register detik yang menyimpan nilai detik, menempati alamat register 00h. **AsoB 081005, Delta Electronic.**



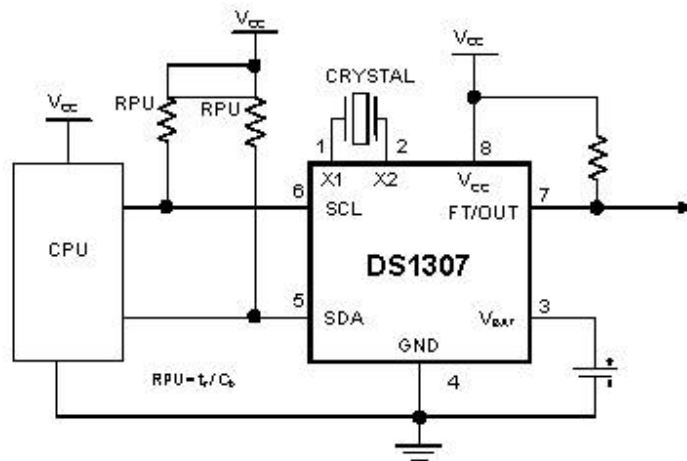
Gambar 1



Gambar 2



Gambar 3



Gambar 4

Tabel 1

ADDRESS	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	FUNCTION	RANGE
00H	CH	10 Seconds			Seconds			Seconds	00-59	
01H	0	10 Minutes			Minutes			Minutes	00-59	
02H	0	12	10 Hour	10 Hour	Hours			Hours	1-12 +AM/PM 00-23	
		24	PM/AM							
03H	0	0	0	0	0	DAY		Day	01-07	
04H	0	0	10 Date		Date			Date	01-31	
05H	0	0	0	10 Month	Month			Month	01-12	
06H	10 Year			Year			Year	00-99		
07H	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08H-3FH								RAM 56 x 8	00H-FFH	