

SPC

SMART PERIPHERAL CONTROLLER

SEVEN SEGMENT DISPLAY

Trademarks & Copyright

XT, AT, IBM, PC, and PC-DOS are trademarks of International Business Machines Corp.

MS-DOS is a registered trademark of Microsoft Corporation.

MCS-51 and Pentium are registered trademarks of Intel Corporation.

MetaLink ASM51 is copyright by MetaLink Corporation

Daftar Isi

1.	Pendahuluan	1
1.1	Spesifikasi Eksternal SPC SEVEN SEGMENT	1
1.2	Spesifikasi Internal 4 Bit Parallel SPC SEVEN SEGMENT	1
1.3	Spesifikasi Internal SPI SPC SEVEN SEGMENT	2
1.4	Spesifikasi Internal UART SPC SEVEN SEGMENT	2
1.5	Sistem yang Dianjurkan	3
2.	Perangkat Keras SPC SEVEN SEGMENT	4
2.1	Tata Letak Komponen SPC SEVEN SEGMENT	4
2.2	Setting Jumper Mode dan Antarmuka.....	4
2.3	Hubungan DT-51 Minimum System dengan SPC SEVEN SEGMENT	6
2.4	Modul Display Seven Segment.....	8
2.5	Ekspansi SPC SEVEN SEGMENT	10
2.6	Mencoba SPC SEVEN SEGMENT dengan ExCount.Hex.....	10
2.7	Mencoba SPC SEVEN SEGMENT dengan Ex4Bit.Hex	11
2.8	Mencoba SPC SEVEN SEGMENT dengan ExSPI.Hex.....	11
2.9	Mencoba SPC SEVEN SEGMENT dengan Ex232.Hex	11
2.10	Mencoba SPC SEVEN SEGMENT dengan Ex485.Hex	11
2.11	Menggunakan K11UARTEST11.EXE	12
2.12	Menggunakan IDSET.EXE.....	12
3.	Perangkat Lunak SPC SEVEN SEGMENT.....	13
3.1	Spesifikasi Mode Stand-Alone Counter	13
3.2	Spesifikasi Mode Display	14
3.2.1	Spesifikasi 4 Bit Parallel	14
3.2.1.1	Command Write Character.....	15
3.2.2	Spesifikasi SPI.....	16
3.2.3	Spesifikasi UART RS-232	18
3.2.4	Spesifikasi UART RS-485	20
3.3	Driver dan Rutin	22
3.4	Contoh Aplikasi dan Program	28
3.5	Kerangka Program	29
Lampiran		
A.	Skema Modul SPC SEVEN SEGMENT	32
B.	Skema Modul Display Seven Segment	33
C.	Daftar Command SPC SEVEN SEGMENT Mode Display (kecuali antarmuka 4 Bit Parallel).....	34
D.	Protokol SPC SEVEN SEGMENT Mode Display (kecuali antarmuka 4 Bit Parallel).....	35
E.	Tabel Karakter SPC SEVEN SEGMENT Mode Display (kecuali antarmuka 4 Bit Parallel).....	40

1. PENDAHULUAN

Smart Peripheral Controller / SPC SEVEN SEGMENT merupakan penampil 8 digit seven segment yang mendukung 4 macam antarmuka (interface) yaitu 4 bit Parallel, Serial Peripheral Interface (SPI), Universal Asynchronous Receiver/Transmitter (UART) RS-232 atau UART RS-485 sehingga mempermudah pengguna untuk memilih antarmuka yang diinginkan. Contoh aplikasi dari SPC SEVEN SEGMENT adalah untuk display karakter, penampil counter, penampil jam, tanggal, dan lain-lain.

1.1. SPESIFIKASI EKSTERNAL SPC SEVEN SEGMENT

Spesifikasi Eksternal SPC SEVEN SEGMENT sebagai berikut :

- Seven Segment 0,5 inci (merah).
- Kompatibel penuh dengan DT-51 Minimum System Ver 3.0, DT-51 Low Cost Micro System, dan DT-51 Low Cost Nano System.
- Hanya perlu 3 jalur kabel untuk interface dengan mikroprosesor / mikrokontroler lain.
- Dapat digunakan pada SPI, 4 bit Parallel, atau UART, namun tidak dapat dipakai bersama-sama.
- SPI dan 4 bit Parallel beroperasi pada taraf logic CMOS.
- UART beroperasi pada taraf logic RS-232 atau UART RS-485.
- Dapat di-ekspan sampai 256 board (khusus untuk antarmuka SPI dan UART RS-485).
- Memiliki kemampuan non-volatile counter dari -9.999.999 sampai dengan 9.999.999.
- Dapat berfungsi sebagai display karakter ataupun stand-alone display counter.
- Ukuran modul seven segment dapat diganti sesuai dengan keinginan pengguna.
- Built-in RTC (Real Time Clock).
- Memiliki EEPROM 100 byte yang dapat digunakan secara umum.
- Single supply 12 V DC – 30 V DC.
- Tersedia prosedur siap pakai dalam assembly MCS-51 untuk penggunaan SPC SEVEN SEGMENT.

1.2. SPESIFIKASI INTERNAL 4 BIT PARALLEL SPC SEVEN SEGMENT

Dalam penggunaan dari 4 bit Parallel SPC SEVEN SEGMENT akan dikenal adanya tiga layer (lapisan) penggunaan:

Pertama	:	4 bit Parallel Engine Layer
Kedua	:	4 bit Parallel Protocol Layer
Ketiga	:	4 bit Parallel Application Layer

4 Bit Parallel Engine Layer adalah lapisan yang mengurus semua kegiatan dari tiap bit yang akan diterima atau yang akan dikirim.

Bagian ini tidak perlu diubah kecuali untuk keperluan khusus. Bagi pengguna yang belum mahir dan berpengalaman tidak dianjurkan untuk mengubah bagian ini.

Bagian ini terdapat dalam *ENG_4BIT.INC*.

4 Bit Parallel Protocol Layer adalah lapisan yang terletak satu lapis lebih tinggi dari *4 Bit Parallel Engine Layer* dan dipergunakan untuk mengatur semua lalu lintas data dan sudah tersusun sesuai dengan kegunaan menjadi paket Sub-rutin.

Bagian ini tidak perlu diubah kecuali untuk keperluan khusus. Bagi pengguna yang belum mahir dan berpengalaman tidak dianjurkan untuk mengubah bagian ini.

Bagian ini terdapat dalam *7S_4BIT.INC*.

4 Bit Parallel Application Layer adalah lapisan terluar yang dipergunakan untuk berinteraksi secara langsung dengan pengguna.

Bagian ini tidak perlu diubah kecuali untuk keperluan khusus. Bagi pengguna yang belum mahir dan berpengalaman tidak dianjurkan untuk mengubah bagian ini.

Bagian ini terdapat dalam *7S_4BIT.INC*.

1.3. SPESIFIKASI INTERNAL SPI SPC SEVEN SEGMENT

Dalam penggunaan dari SPI SPC SEVEN SEGMENT akan dikenal adanya tiga layer (lapisan) penggunaan:

Pertama	:	SPI Engine Layer
Kedua	:	SPI Protocol Layer
Ketiga	:	SPI Application Layer

SPI Engine Layer adalah lapisan yang mengurus semua kegiatan dari tiap bit yang akan diterima atau yang akan dikirim.

Bagian ini tidak perlu diubah kecuali untuk keperluan khusus. Bagi pengguna yang belum mahir dan berpengalaman tidak dianjurkan untuk mengubah bagian ini.

Bagian ini terdapat dalam *ENG_SPI.INC*.

SPI Protocol Layer adalah lapisan yang terletak satu lapis lebih tinggi dari *SPI Engine Layer* dan dipergunakan untuk mengatur semua lalu lintas data dan sudah tersusun sesuai dengan kegunaan menjadi paket Sub-rutin.

Bagian ini tidak perlu diubah kecuali untuk keperluan khusus. Bagi pengguna yang belum mahir dan berpengalaman tidak dianjurkan untuk mengubah bagian ini.

Bagian ini terdapat dalam *7S_SPI.INC*.

I²C Application Layer adalah lapisan terluar yang dipergunakan untuk berinteraksi secara langsung dengan pengguna.

Bagian ini tidak perlu diubah kecuali untuk keperluan khusus. Bagi pengguna yang belum mahir dan berpengalaman tidak dianjurkan untuk mengubah bagian ini.

Bagian ini terdapat dalam *7S_SPI.INC*.

1.4. SPESIFIKASI INTERNAL UART SPC SEVEN SEGMENT

Dalam penggunaan dari UART SPC SEVEN SEGMENT akan dikenal adanya tiga layer (lapisan) penggunaan:

Pertama	:	UART Engine Layer
---------	---	-------------------

Kedua : UART Protocol Layer
Ketiga : UART Application Layer

UART Engine Layer adalah lapisan yang mengurus semua kegiatan dari tiap bit yang akan diterima atau yang akan dikirim.

Bagian ini tidak perlu diubah kecuali untuk keperluan khusus. Bagi pengguna yang belum mahir dan berpengalaman tidak dianjurkan untuk mengubah bagian ini.

Bagian ini terdapat dalam *ENG_232.INC* dan *ENG_485.INC*.

UART Protocol Layer adalah lapisan yang terletak satu lapis lebih tinggi dari *UART Engine Layer* dan dipergunakan untuk mengatur semua lalu lintas data dan sudah tersusun sesuai dengan kegunaan menjadi paket Sub-rutin.

Bagian ini tidak perlu diubah kecuali untuk keperluan khusus. Bagi pengguna yang belum mahir dan berpengalaman tidak dianjurkan untuk mengubah bagian ini.

Bagian ini terdapat dalam *7S_232.INC* dan *7S_485.INC*.

UART Application Layer adalah lapisan terluar yang dipergunakan untuk berinteraksi secara langsung dengan pengguna.

Bagian ini tidak perlu diubah kecuali untuk keperluan khusus. Bagi pengguna yang belum mahir dan berpengalaman tidak dianjurkan untuk mengubah bagian ini.

Bagian ini terdapat dalam *7S_232.INC* dan *7S_485.INC*.

1.5. SISTEM YANG DIANJURKAN

Perangkat keras :

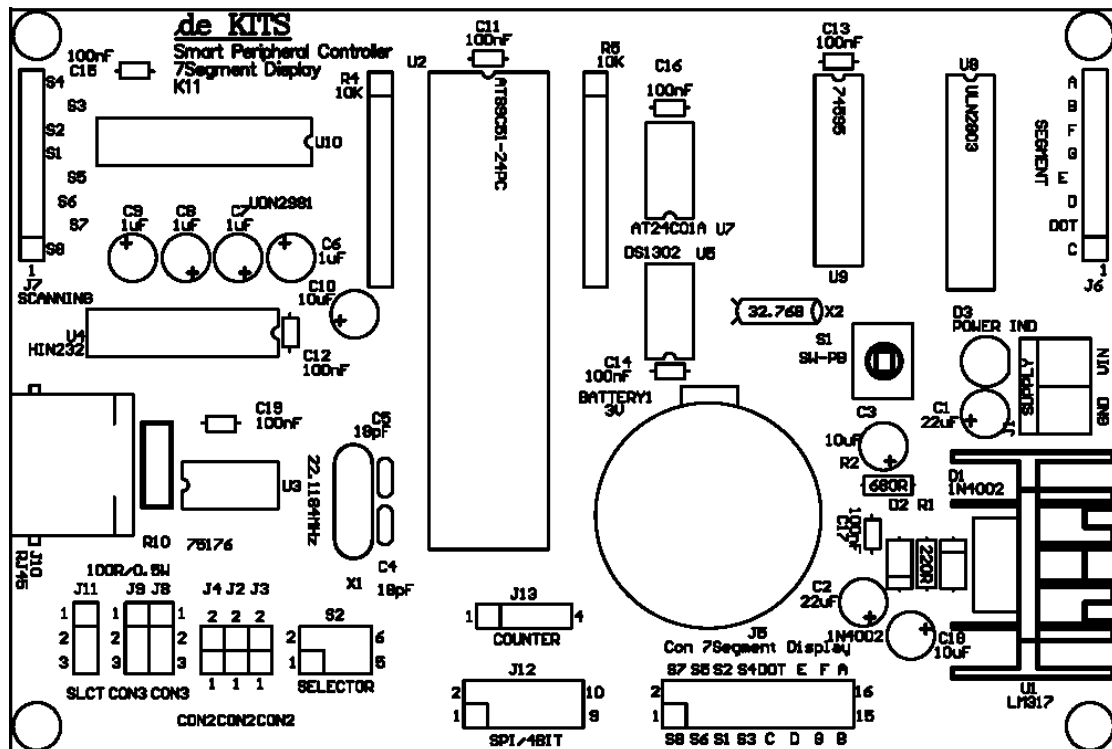
- PC XT™ / AT™ Pentium® IBM™ Compatible dengan port serial (COM1 / COM2).
- Board DT-51 Minimum System.
- Floppy Disk 3.5", kapasitas 1,44 Mbytes atau CD-ROM drive.
- Hard disk dengan kapasitas minimum 4 Mbytes.

Perangkat lunak :

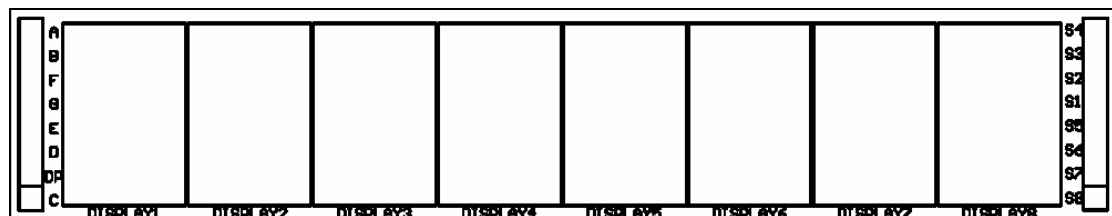
- Sistem operasi MS-DOS® atau PC-DOS™.
- Assembler ASM51©.
- File-file yang ada pada pada disket/CD program:
 - EXCOUNT.ASM, EXCOUNT.HEX, EX4BIT.ASM, EX4BIT.HEX, EXSPI.ASM, EXSPI.HEX, EX232.ASM, EX232.HEX, EX485.ASM, EX485.HEX.
 - 7S_4BIT.INC, 7S_SPI.INC, 7S_232.INC, 7S_485.INC, ENG_4BIT.INC, ENG_SPI.INC, ENG_232.INC, ENG_485.INC.
 - TMPT4BIT.ASM, TMPTSPI.ASM, TMPT232.ASM, TMPT485.ASM.
 - K11UARTEST.EXE, IDSET.EXE.
 - MANUAL SPC SEVEN SEGMENT.PDF, QUICK START SPC SEVEN SEGMENT.PDF.

2. PERANGKAT KERAS SPC SEVEN SEGMENT

2.1. TATA LETAK KOMPONEN SPC SEVEN SEGMENT



Modul SPC Seven Segment



Modul Display Seven Segment

2.2. SETTING JUMPER MODE DAN ANTARMUKA

SPC SEVEN SEGMENT dapat dioperasikan dalam 2 mode, mode display dan mode stand-alone counter. Pemilihan mode ini diatur dengan cara mengganti setting jumper S2.

SPC SEVEN SEGMENT S2	Mode
Don't care 	Stand-alone Counter
Pilih Antarmuka 	Display

Keterangan : ■■ => jumper tersambung (ON)
 □□ => jumper terlepas (OFF)
 Arah posisi jumper sesuai dengan posisi pada gambar **bagian 2.1**

Pada mode stand-alone counter, SPC SEVEN SEGMENT hanya berfungsi sebagai counter yang dikendalikan melalui header J13.

SPC SEVEN SEGMENT J13	Fungsi
Pin 1 (GND)	Titik referensi
Pin 2 (CTRCLR)	Pin clear untuk counter
Pin 3 (CTRDIR)	Pin count up/down untuk counter
Pin 4 (CTRCLK)	Pin clock untuk counter

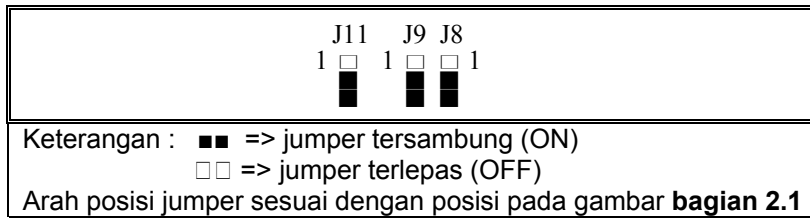
Penjelasan lebih lanjut mengenai mode ini terdapat pada **bagian 3.1**.

Pada mode display, SPC SEVEN SEGMENT dikendalikan oleh mikrokontroler atau komputer melalui antarmuka yang dikehendaki. Penggunaan antarmuka tidak dapat dilakukan bersamaan. Pemilihan antara antarmuka diatur dengan cara mengganti setting jumper S2.

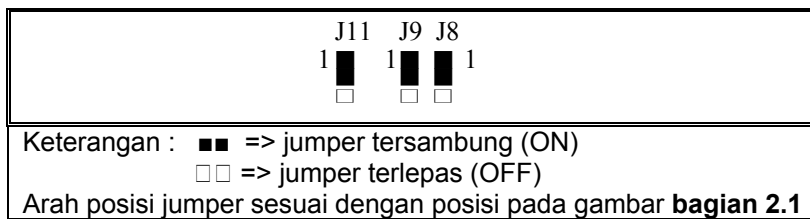
SPC SEVEN SEGMENT S2	Antarmuka
	SPI
	4 Bit Parallel
	UART RS-232
	UART RS-485

Keterangan : ■■ => jumper tersambung (ON)
 □□ => jumper terlepas (OFF)
 Arah posisi jumper sesuai dengan posisi pada gambar **bagian 2.1**

Untuk mengaktifkan mode UART RS-232, selain menentukan posisi jumper pada S2, jumper J8, J9, dan J11 harus diposisikan sebagai berikut :

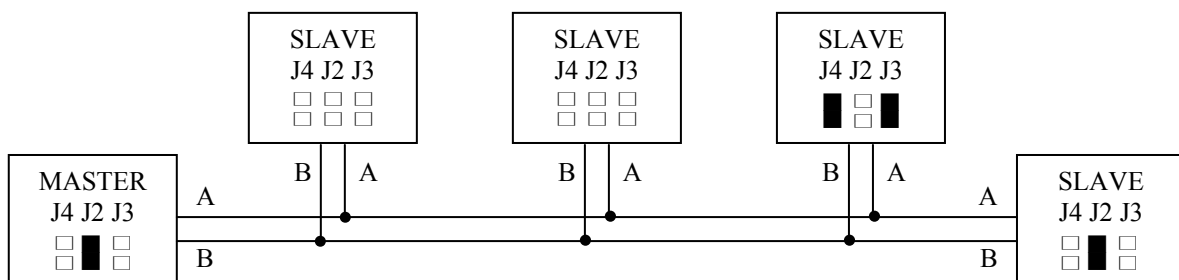


Untuk mengaktifkan mode UART RS-485, selain menentukan posisi jumper pada S2, jumper J8, J9, dan J11 harus diposisikan sebagai berikut :



Jika jumper dilepas, kedua pin di tengah dapat berfungsi sebagai jalur komunikasi UART dengan level tegangan CMOS. Pin 2 J8 berfungsi sebagai jalur receive untuk SPC SEVEN SEGMENT, sedangkan pin 2 J9 berfungsi sebagai jalur transmit dari SPC SEVEN SEGMENT.

Jumper J4, J2, dan J3 masing-masing adalah jumper bias +, termination, dan bias - . Dalam satu jaringan, jumper bias + dan jumper bias - hanya ada 1 pasang saja yang diaktifkan, sedangkan jumper termination hanya terpasang pada node-node yang paling ujung pada network tersebut. Contoh pemasangan jumper-jumper tersebut dapat dilihat pada gambar berikut ini.



Keterangan : ■■ => jumper tersambung (ON)
□□ => jumper terlepas (OFF)

Arah posisi jumper sesuai dengan posisi pada gambar **bagian 2.1**

2.3. HUBUNGAN DT-51 MINIMUM SYSTEM DENGAN SPC SEVEN SEGMENT

SPC SEVEN SEGMENT merupakan suatu sistem yang ‘Smart’. Selain dapat dihubungkan dengan DT-51 Minimum System atau dengan sistem mikroprosesor / mikrokontroler yang lain, SPC SEVEN SEGMENT dapat juga dihubungkan dengan komputer. Apabila Anda ingin menghubungkan SPC SEVEN SEGMENT dengan sistem yang lain kami sarankan untuk mempelajari skema SPC SEVEN SEGMENT (lihat **lampiran A**).

Hubungan secara SPI bus ditunjukkan pada tabel berikut:

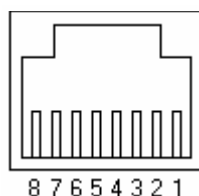
SPI Bus	DT-51 Minimum System PORT C & PORT 1	SPC SEVEN SEGMENT J12
SCK	Pin 13 (Port 1.4)	Pin 7 (CLK/SCK)
MOSI	Pin 14 (Port 1.5)	Pin 8 (DOT/MOSI)
CLR	Pin 15 (Port 1.6)	Pin 9 (CLR)
MISO	Pin 16 (Port 1.7)	Pin 10 (MISO)

Hubungan secara 4 bit Parallel ditunjukkan pada tabel berikut:

4 Bit Parallel	DT-51 Minimum System PORT C & PORT 1	SPC SEVEN SEGMENT J12
Data bit 0	Pin 9 (Port 1.0)	Pin 3 (D0)
Data bit 1	Pin 10 (Port 1.1)	Pin 4 (D1)
Data bit 2	Pin 11 (Port 1.2)	Pin 5 (D2)
Data bit 3	Pin 12 (Port 1.3)	Pin 6 (D3)
Clock	Pin 13 (Port 1.4)	Pin 7 (CLK/SCK)
DOT	Pin 14 (Port 1.5)	Pin 8 (DOT/MOSI)
Clear	Pin 15 (Port 1.6)	Pin 9 (CLR)

Hubungan secara UART RS-232 dan RS-485 dilakukan melalui konektor RJ45 pada J10 dengan urutan pin sebagai berikut:

J10 Tampak Depan/Sisi Lubang



No	Nama	Keterangan
1	B485RJ	RS485 B+
2	A485RJ	RS485 A-
3	NC	Tidak dipakai
4	SGND	Titik referensi sinyal
5	RX232RJ	RS232 receiver pin
6	TX232RJ	RS232 transmitter pin
7	SGND	Titik referensi sinyal
8	NC	Tidak dipakai

Untuk komunikasi UART RS-232 dan RS-485 terdapat beberapa jumper yang dipergunakan yaitu J2, J3, J4 (untuk RS-485) dan J8, J9 (untuk RS-232/RS-485). Konfigurasi jumper secara detail terdapat pada **bagian 2.2**.

Hubungan secara UART RS-232 ditunjukkan pada tabel berikut:

DT-51 Minimum System / Komputer SERIAL PORT DB 9	SPC SEVEN SEGMENT J10
Pin 5 (GND)	Pin 7 (SGND)
Pin 3 (TX)	Pin 6 (TX232RJ)
Pin 2 (RX)	Pin 5 (RX232RJ)

Hubungan secara UART RS-485 ditunjukkan pada tabel-tabel berikut:

DT-51 Minimum System / Komputer SERIAL PORT DB 9	RS232 - RS485 Level Converter
Pin 5 (GND)	GND
Pin 3 (TX)	TX
Pin 2 (RX)	RX

RS232 - RS485 Level Converter	SPC SEVEN SEGMENT J10
GND	Pin 4 (SGND)
B+	Pin 1 (B485RJ)
A-	Pin 2 (A485RJ)

Penting !

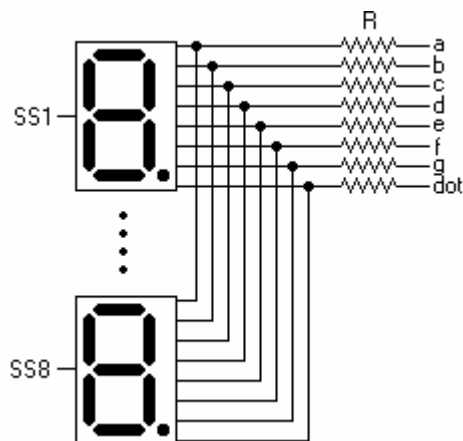
Catu daya 12 V DC dihubungkan dengan konektor J1 (Power). Perhatikan polaritasnya jangan sampai terbalik, karena dapat mengakibatkan kerusakan. Perhatikan hubungan referensi ground (GND) antara modul SPC SEVEN SEGMENT dengan DT-51 Minimum System.

2.4. MODUL DISPLAY SEVEN SEGMENT

Modul display seven segment dapat diganti sendiri sesuai dengan keinginan user. Untuk itu harap diperhatikan hal-hal sebagai berikut :

- Tegangan seven segment tiap segmennya
- Arus seven segment tiap segmennya
- Seven segment yang dipasang harus bertipe common anoda

Rangkaian yang dianjurkan :



Rumus menghitung nilai R :

$$R = \frac{V_{\text{supply}} - V_{\text{ss}} - 1.6}{I_{\text{ss}}}$$

R = nilai resistor R (ohm)
 V_{supply} = tegangan supply DC unregulated (volt) → 7 – 30 V DC
 V_{ss} = tegangan seven segment per segmentnya (volt)
 I_{ss} = arus seven segment per segmentnya (ampere)

Rumus menghitung daya R :

$$P = (V_{\text{supply}} - V_{\text{ss}} - 1.6) \times I_{\text{ss}}$$

P = daya resistor R (watt)
 V_{supply} = tegangan supply DC unregulated (volt) → 7 – 30 V DC
 V_{ss} = tegangan seven segment per segmentnya (volt)
 I_{ss} = arus seven segment per segmentnya (ampere)

Contoh :

Seven segment yang digunakan memiliki spesifikasi :

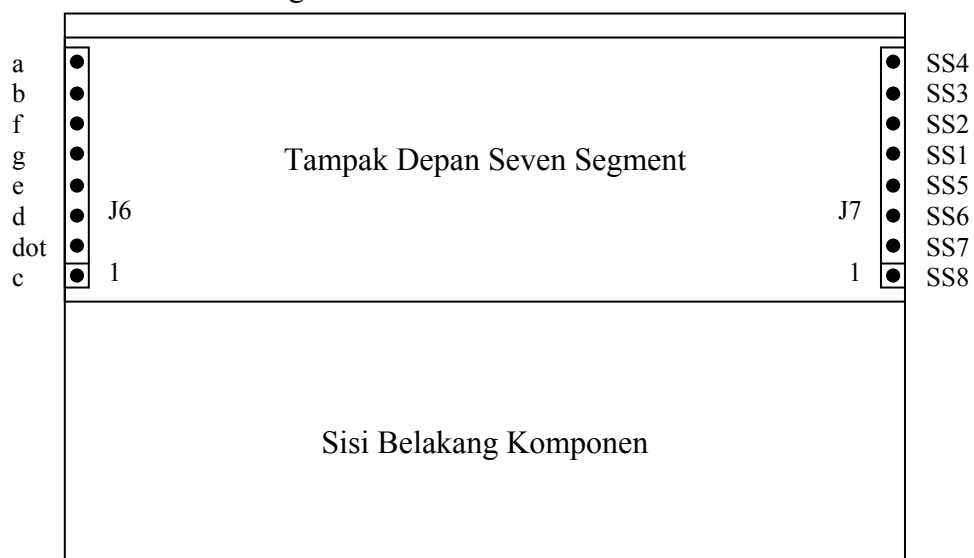
- $V_{\text{ss}} = 14 \text{ V DC}$
- $I_{\text{ss}} = 40 \text{ mA}$

Supply yang disediakan (terhubung ke J1) adalah 20 V DC.

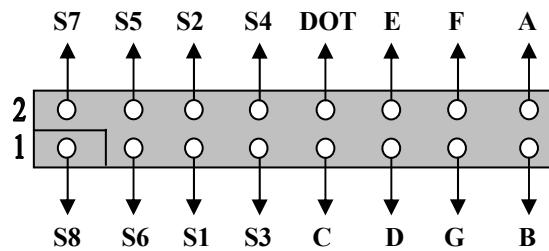
Nilai $R = (20 - 14 - 1.6) / 0.04 = 4.4 / 0.04 = 110 \Omega$, diambil nilai yang mendekati yaitu 120Ω .

Daya $R = (20 - 14 - 1.6) \times 0.04 = 4.4 \times 0.04 = 0.176 \text{ W} = 176 \text{ mW}$, dipilih R dengan daya $\frac{1}{4} \text{ W}$.

SS1 hingga SS8 dihubungkan pada common masing-masing seven segment. Konfigurasi pin-pin SS1 hingga SS8 dan a hingga dot pada SPC SEVEN SEGMENT adalah sebagai berikut :



Selain J6 dan J7, konektor untuk display seven segment juga disediakan pada J5 dengan konfigurasi sebagai berikut :



2.5. EKSPANSI SPC SEVEN SEGMENT

SPC SEVEN SEGMENT dapat di-ekspan sampai 256 board (hanya untuk antarmuka SPI dan UART RS-485). Beberapa hal yang perlu diperhatikan apabila menggunakan lebih dari satu board SPC SEVEN SEGMENT :

- Setiap board harus mempunyai alamat terprogram yang berbeda.
- Perhatikan konfigurasi jumper J2, J3, dan J4 pada jaringan (jika menggunakan UART RS-485).

Alamat awal bernilai 255. Tombol tactile switch (S1) yang ada berfungsi untuk menampilkan alamat dari board SPC SEVEN SEGMENT. Untuk kembali ke tampilan sebelumnya, tekan tombol sekali lagi (tombol bersifat toggle).

2.6. MENCOBA SPC SEVEN SEGMENT DENGAN EXCOUNT.HEX

Setting Hardware

- ◆ Atur setting jumper sehingga SPC SEVEN SEGMENT beroperasi pada mode Stand-Alone Counter (lihat **bagian 2.2**).
- ◆ Hubungkan DT-51 Minimum System dengan SPC SEVEN SEGMENT dengan hubungan sebagai berikut:

SPC SEVEN SEGMENT J13	DT-51 Minimum System
Pin 1 (GND)	PORT CONTROL Pin 2 (GND)
Pin 2 (CTRCLR)	PORT C & PORT 1 Pin 10 (Port 1.1)
Pin 3 (CTRDIR)	PORT C & PORT 1 Pin 12 (Port 1.3)
Pin 4 (CTRCLK)	PORT C & PORT 1 Pin 14 (Port 1.5)

- ◆ Hubungkan port serial DT-51 MinSys dengan COM1/COM2 dari komputer dengan menggunakan kabel serial.
- ◆ Download EXCOUNT.HEX yang terdapat pada disket/CD.

Proses Program EXCOUNT

- ◆ Setelah program selesai di-download maka Counter akan dimulai dari 0000000.
- ◆ Program akan count up sebanyak 6 kali dan count down sebanyak 8 kali secara berulang-ulang.

2.7. MENCoba SPC SEVEN SEGMENT DENGAN EX4BIT.HEX

Setting Hardware

- ◆ Atur setting jumper sehingga antarmuka yang digunakan adalah 4 bit parallel (lihat **bagian 2.2**).
- ◆ Hubungkan DT-51 Minimum System dengan SPC SEVEN SEGMENT (lihat **bagian 2.3**) melalui 4 bit parallel. Hubungkan juga referensi ground kedua board tersebut.
- ◆ Hubungkan port serial DT-51 MinSys dengan COM1/COM2 dari komputer dengan menggunakan kabel serial.
- ◆ Download EX4BIT.HEX yang terdapat pada disket/CD.

Proses Program EX4BIT

- ◆ Setelah program selesai di-download maka akan tampak urutan angka “12345678”.

2.8. MENCoba SPC SEVEN SEGMENT DENGAN EXSPI.HEX

Setting Hardware

- ◆ Atur setting jumper sehingga antarmuka yang digunakan adalah SPI (lihat **bagian 2.2**).
- ◆ Hubungkan DT-51 Minimum System dengan SPC SEVEN SEGMENT (lihat **bagian 2.3**) melalui SPI. Hubungkan juga referensi ground kedua board tersebut.
- ◆ Hubungkan port serial DT-51 MinSys dengan COM1/COM2 dari komputer dengan menggunakan kabel serial.
- ◆ Download EXSPI.HEX yang terdapat pada disket/CD.

Proses Program EXSPI

- ◆ Setelah program selesai di-download maka akan tampak tulisan “SPC 7SEG” berkedip.

2.9. MENCoba SPC SEVEN SEGMENT DENGAN EX232.HEX

Setting Hardware

- ◆ Atur setting jumper sehingga antarmuka yang digunakan adalah UART RS-232 (lihat **bagian 2.2**).
- ◆ Hubungkan DT-51 Minimum System dengan SPC SEVEN SEGMENT (lihat **bagian 2.3**) melalui UART RS-232. Hubungkan juga referensi ground kedua board tersebut.
- ◆ Hubungkan port serial DT-51 MinSys dengan COM1/COM2 dari komputer dengan menggunakan kabel serial.
- ◆ Download EX232.HEX yang terdapat pada disket/CD.

Proses Program EX232

- ◆ Setelah program selesai di-download maka akan tampak tulisan “SPC 7SEG” berkedip.

2.10. MENCoba SPC SEVEN SEGMENT DENGAN EX485.HEX

Setting Hardware

- ◆ Atur setting jumper sehingga antarmuka yang digunakan adalah UART RS-485 (lihat **bagian 2.2**).

- ◆ Hubungkan DT-51 Minimum System dengan SPC SEVEN SEGMENT (lihat **bagian 2.3**) melalui UART RS-485. Perhatikan hubungan referensi ground.
- ◆ Hubungkan port serial DT-51 MinSys dengan COM1/COM2 dari komputer dengan menggunakan kabel serial.
- ◆ Download EX485.HEX yang terdapat pada disket/CD.

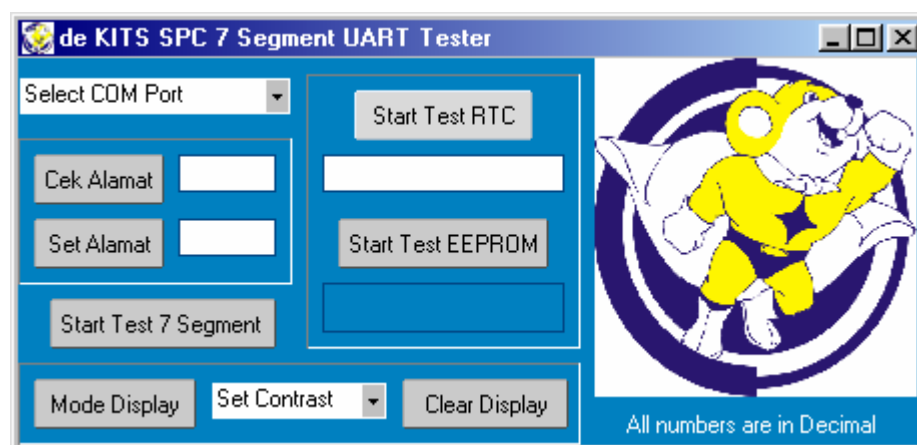
Proses Program EX485

- ◆ Setelah program selesai di-download maka akan tampak tulisan “SPC 7SEG” berkedip.

2.11. MENGGUNAKAN K11UARTEST.EXE

Setting Hardware

- ◆ Atur setting jumper sehingga antarmuka yang digunakan adalah UART RS-232 (lihat **bagian 2.2**).
- ◆ Hubungkan Komputer dengan SPC SEVEN SEGMENT (lihat **bagian 2.3**) melalui UART RS-232.
- ◆ Eksekusi K11UARTEST.EXE yang terdapat dalam disket/CD.



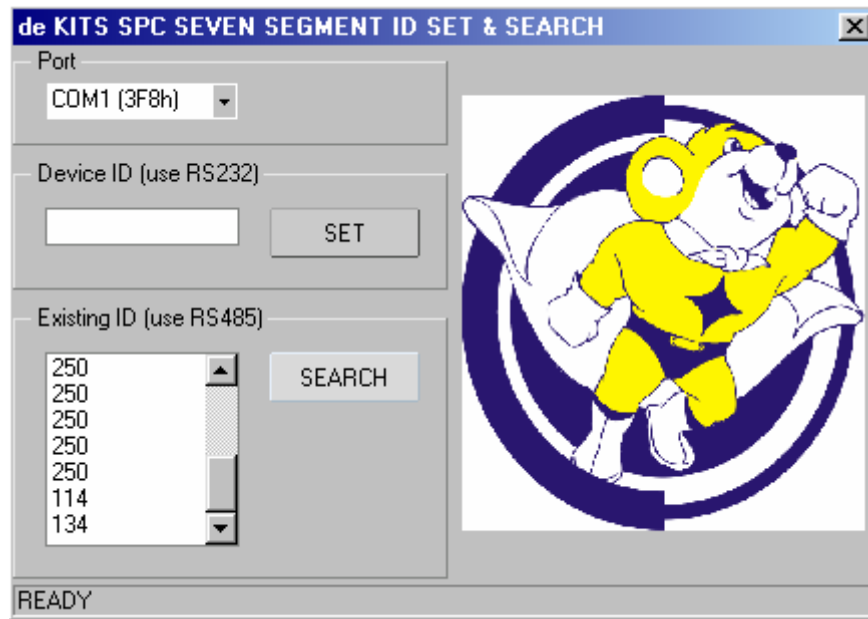
Proses Program K11UARTEST

- ◆ Pilih COM Port yang digunakan.
- ◆ K11UARTEST dapat digunakan untuk menguji fitur-fitur seperti tampilan mode tanggal, tampilan mode jam, proses tulis dan baca EEPROM, mode display karakter, contrast level, dan pemeriksaan atau pengisian nilai alamat.

2.12. MENGGUNAKAN IDSET.EXE

Setting Hardware

- ◆ Atur setting jumper sehingga antarmuka yang digunakan adalah UART RS-232 atau UART RS-485 (lihat **bagian 2.2**).
- ◆ Hubungkan Komputer dengan SPC SEVEN SEGMENT (lihat **bagian 2.3**) melalui UART RS-232 atau UART RS-485. Perhatikan hubungan referensi ground.
- ◆ Eksekusi IDSET.EXE yang terdapat dalam disket/CD.



Proses Program IDSET

- ◆ Pilih COM Port yang digunakan.
- ◆ Dengan menggunakan antarmuka UART RS-232, IDSET.EXE dapat digunakan untuk mengisi nilai alamat SPC SEVEN SEGMENT. Isikan nilai alamat pada kolom yang tersedia lalu tekan SET.
- ◆ Dengan menggunakan antarmuka UART RS-485, IDSET.EXE dapat digunakan untuk mencari nilai alamat SPC SEVEN SEGMENT yang terhubung dalam satu jaringan. Tekan SEARCH dan kolom yang tersedia akan menampilkan semua alamat SPC SEVEN SEGMENT yang terhubung.

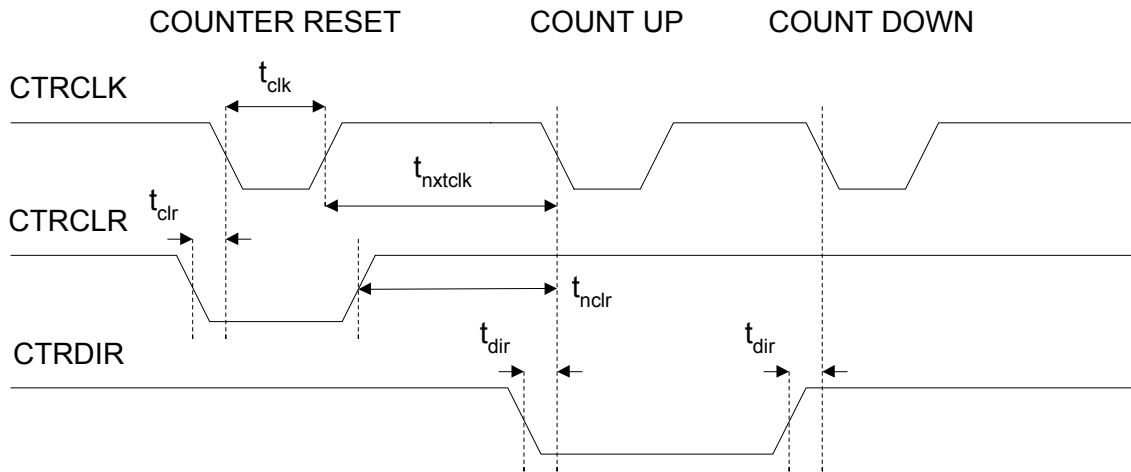
3. PERANGKAT LUNAK SPC SEVEN SEGMENT

Waktu yang dibutuhkan SPC SEVEN SEGMENT mulai menyala hingga siap dioperasikan (Start-up Time) = 150 ms.

3.1. SPESIFIKASI MODE STAND-ALONE COUNTER

Pada mode Stand-alone Counter, tampilan SPC SEVEN SEGMENT otomatis akan menunjukkan nilai counter terakhir yang disimpan pada memory SPC SEVEN SEGMENT. Counter dari SPC SEVEN SEGMENT mampu menampung bilangan mulai -9.999.999 hingga 9.999.999. Terdapat 3 pin yang mengontrol counter yaitu pin CTRLCK, CTRDIR, dan CTRLCLR.

SPC SEVEN SEGMENT J13	Fungsi
Pin 1 (GND)	Titik referensi
Pin 2 (CTRLCLR)	Jika berlogika '0' pada saat Pulsa pencacahan aktif, maka nilai counter akan dikembalikan ke "000000"
Pin 3 (CTRDIR)	Jika berlogika '0' maka sistem akan Count Up Jika berlogika '1' maka sistem akan Count Down
Pin 4 (CTRLCK)	Pulsa pencacahan, aktif pada falling edge (transisi dari logika '1' ke '0')



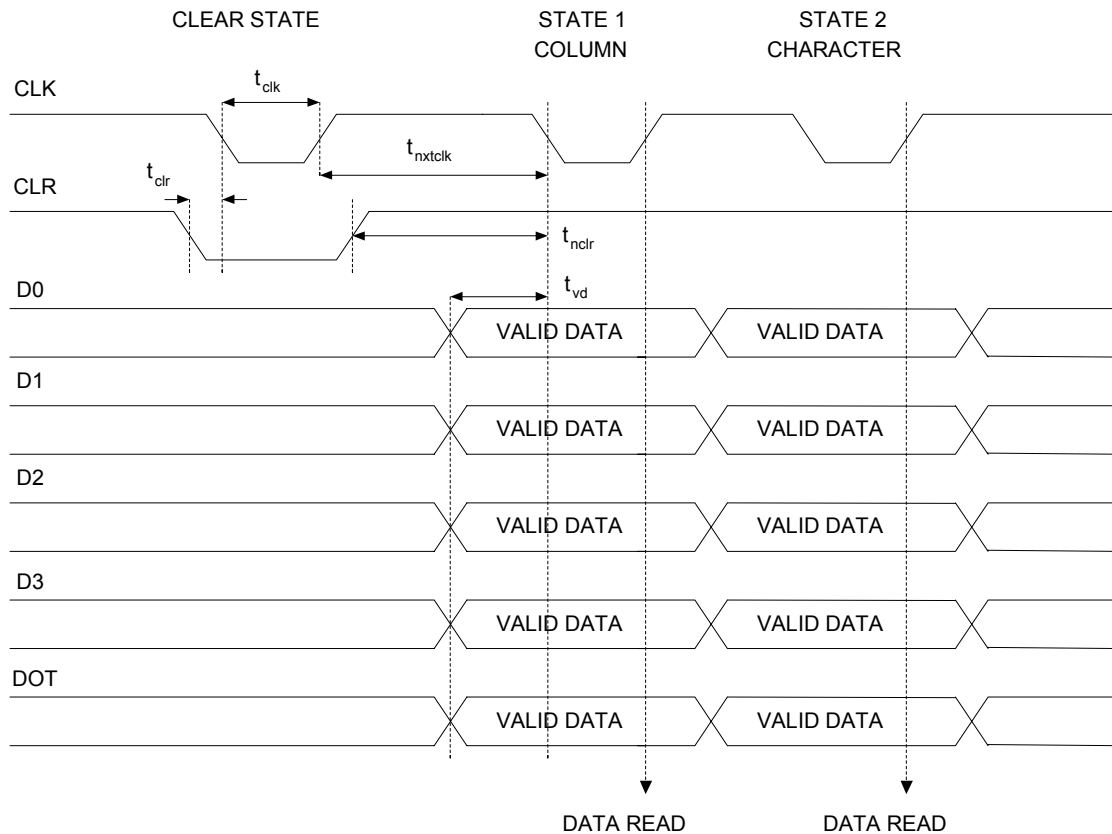
Simbol	Keterangan	Nilai		Satuan
		Min	Max	
t_{clk}	Lebar clock	150	–	μS
t_{clr}	Jarak pulsa clear \rightarrow clock	25	–	μS
t_{nxtclk}	Jarak clock \rightarrow clock berikutnya	150	–	μS
t_{nclr}	Jarak end clear \rightarrow clock berikutnya	25	–	μS
t_{dir}	Jarak pulsa dir \rightarrow clock	25	–	μS

3.2. SPESIFIKASI MODE DISPLAY

Pada mode Display, tampilan SPC SEVEN SEGMENT tergantung dari fungsi yang tersedia pada masing-masing antarmuka.

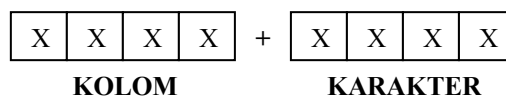
3.2.1. Spesifikasi 4 Bit Parallel

Pada antarmuka 4 bit parallel, SPC SEVEN SEGMENT hanya berfungsi sebagai display numerik 8 digit dan **tidak dapat** di-cascade. Kondisi tampilan (contrast level dan blink) hanya dapat diubah melalui antarmuka selain 4 bit parallel. Protokol yang digunakan sangat sederhana, yaitu mengirimkan 2 x 4 bit secara parallel. Timing diagram terdapat pada gambar berikut.



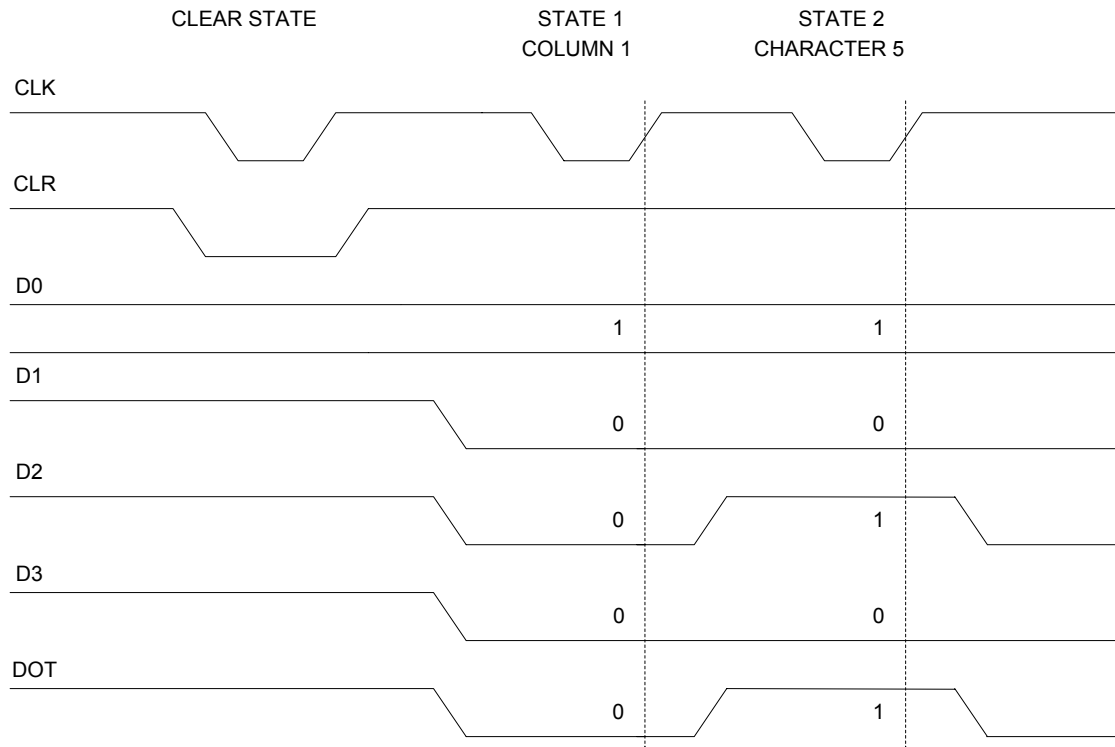
Simbol	Keterangan	Nilai		Satuan
		Min	Max	
t_{clk}	Lebar clock	150	–	μ S
t_{clr}	Jarak pulsa clear \rightarrow clock	25	–	μ S
t_{nxtclk}	Jarak clock \rightarrow clock berikutnya	150	–	μ S
t_{nclr}	Jarak end clear \rightarrow clock berikutnya	25	–	μ S
t_{vd}	Jarak perubahan data \rightarrow clock	25	–	μ S

3.2.1.1. Command Write Character



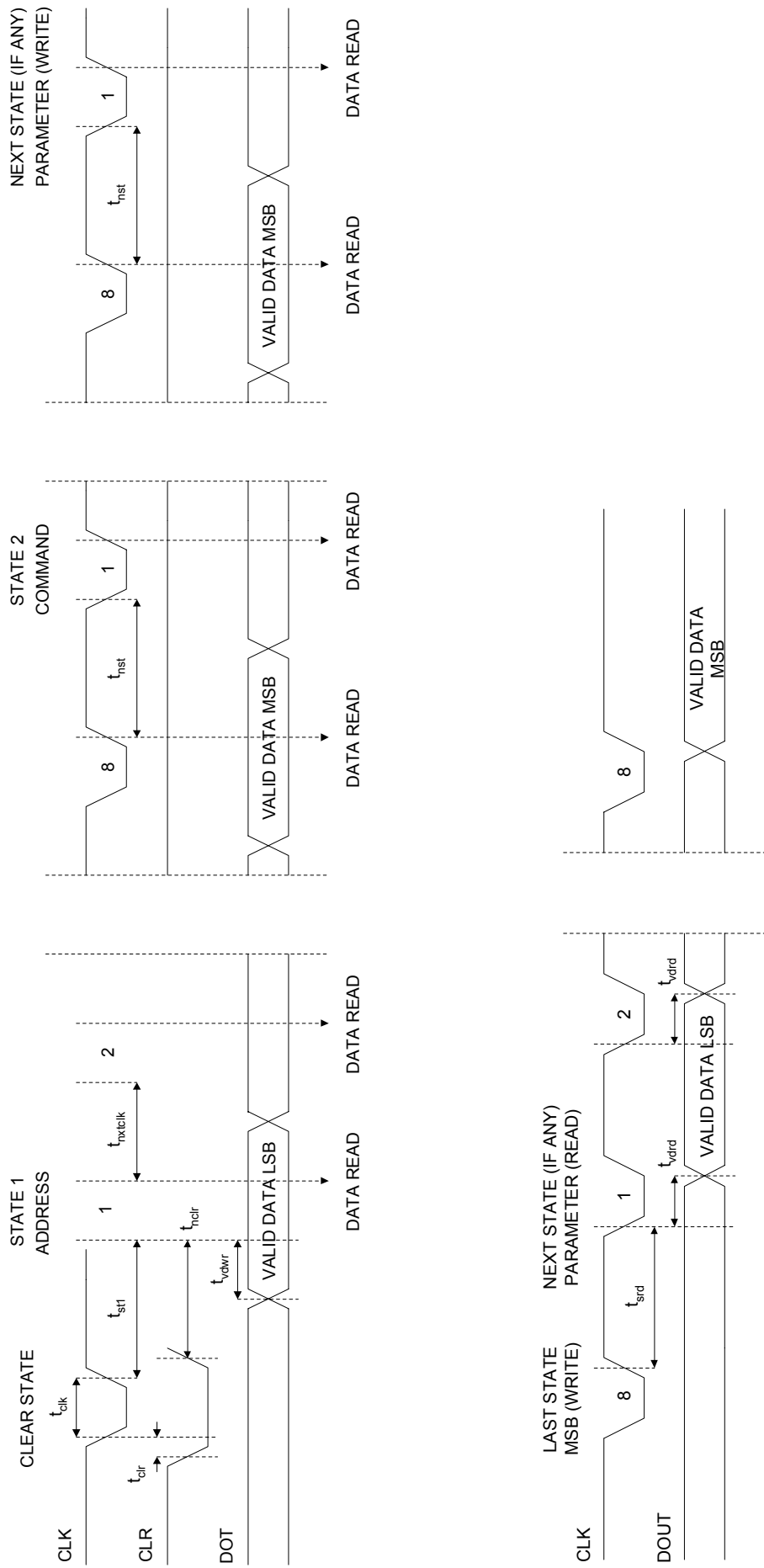
Hanya ada satu command pada antarmuka 4 bit Paralel, yaitu Write Character.

Data strobe yang pertama (state 1) adalah nomor kolom (1 – 8) secara biner (0001 – 1000). Apabila diisi dengan bilangan di luar range tersebut, SPC SEVEN SEGMENT tidak akan merespon command ini. Data strobe yang kedua (state 2) adalah karakter angka yang akan ditampilkan (0 – 9, –, kosong) dalam binary (0000 – 1011). Apabila diisi dengan bilangan di luar range tersebut, SPC SEVEN SEGMENT tidak akan merespon. Titik (dot) dapat dinyalakan dengan memberikan logic 1 pada pin DOT. Pin CLR berfungsi untuk mereset state SPC SEVEN SEGMENT kembali ke state 1. Di bawah ini diberikan contoh timing diagram untuk menampilkan angka 5 di kolom 1 dengan dot dinyalakan.



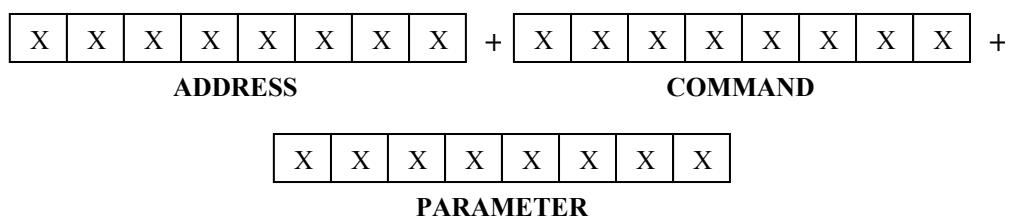
3.2.2. Spesifikasi SPI

Pada antarmuka SPI, SPC SEVEN SEGMENT dapat difungsikan sebagai display karakter, display counter, display jam, dan display tanggal. Tabel karakter yang dapat ditampilkan oleh SPC SEVEN SEGMENT dapat dilihat pada **lampiran E**. Pada antarmuka SPI, SPC SEVEN SEGMENT **dapat** di-cascade hingga 256 buah device. Timing diagram untuk pengiriman dan pembacaan data secara SPI dapat dilihat pada gambar berikut.



Simbol	Keterangan	Nilai		Satuan
		Min	Max	
t_{clk}	Lebar clock	150	–	μ S
t_{clr}	Jarak pulsa clear \rightarrow clock	25	–	μ S
t_{nxtclk}	Jarak clock \rightarrow clock berikutnya	150	–	μ S
t_{nclr}	Jarak end clear \rightarrow clock berikutnya	50	–	μ S
t_{st1}	Jarak clear \rightarrow state 1	50	–	μ S
t_{nst}	Jarak state \rightarrow state berikutnya	100	–	μ S
t_{vdwr}	Jarak perubahan data \rightarrow clock	25	–	μ S
t_{srd}	Jarak state write \rightarrow read	5	–	mS
t_{vdrd}	Jarak valid data state read	100	–	μ S

Protokol SPC SEVEN SEGMENT terdapat pada gambar berikut.



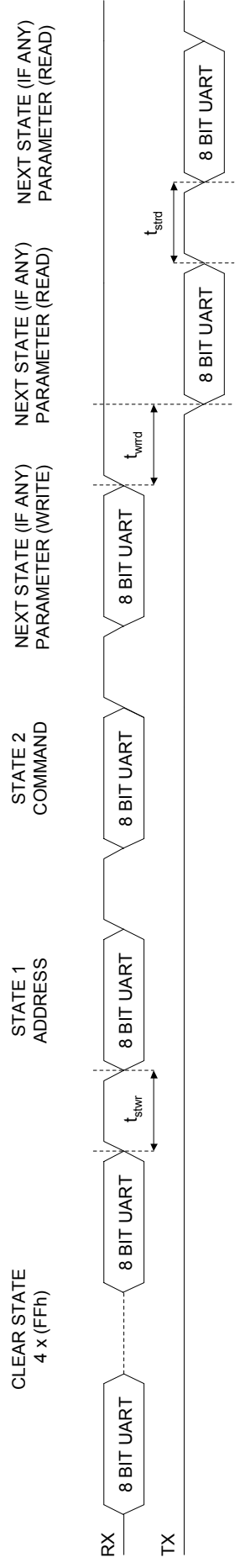
Address, command dan parameter-parameter IN dikirim secara serial dengan clock dimulai dari LSB. Parameter OUT dihasilkan oleh SPC SEVEN SEGMENT **setelah semua parameter IN dikirimkan**. Parameter OUT dibaca dengan membangkitkan clock sejumlah 8 x parameter OUT.

Daftar command dan protokol lengkap terdapat pada **lampiran C dan D**.

3.2.3. Spesifikasi UART RS-232

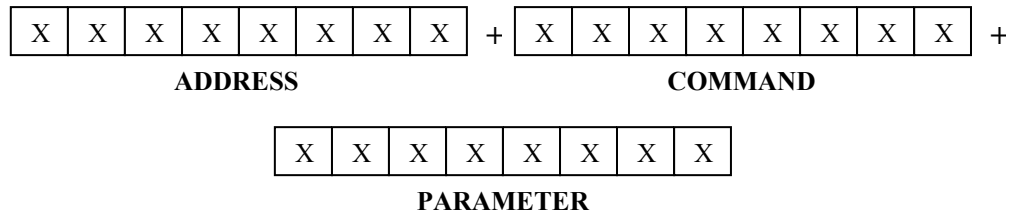
Konfigurasi komunikasi UART RS-232 adalah: Baud rate 9600 bps, 8 data bit, 1 stop bit, tanpa parity bit, dan tanpa flow control.

Pada antarmuka RS-232, SPC SEVEN SEGMENT dapat difungsikan sebagai display karakter, display counter, display jam, dan display tanggal. Tabel karakter yang dapat ditampilkan oleh SPC SEVEN SEGMENT dapat dilihat pada **lampiran E**. Pada antarmuka UART RS-232, SPC SEVEN SEGMENT **tidak dapat** di-cascade. Timing diagram untuk pengiriman dan pembacaan data RS-232 dapat dilihat pada gambar berikut.



SIMBOL	KETERANGAN	NILAI		SATUAN
		MIN	MAX	
t _{stwr}	Jarak antar write state	100	–	μS
t _{wrrd}	Jarak antar write → read state	5	–	mS
t _{strd}	Jarak antar read state	100	–	μS

Protokol SPC SEVEN SEGMENT terdapat pada gambar berikut.



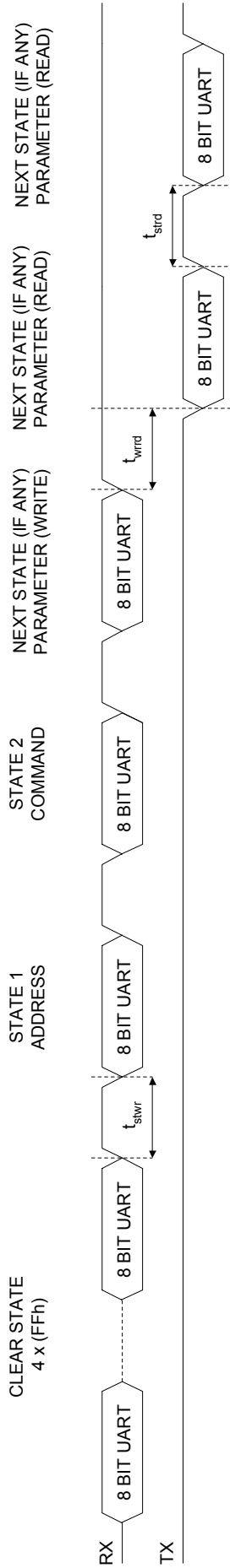
Address dapat diisi dengan sembarang byte.

Daftar command dan protokol lengkap terdapat pada **lampiran C dan D**.

3.2.4. Spesifikasi UART RS-485

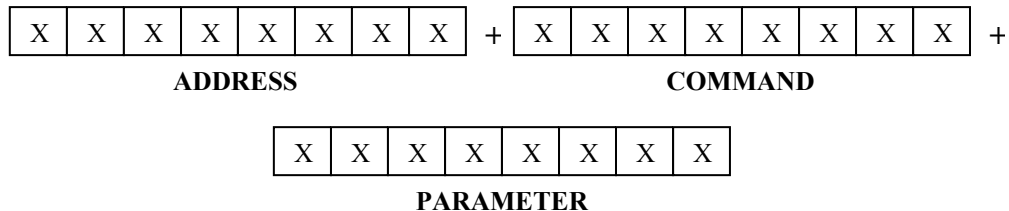
Konfigurasi komunikasi UART RS-485 adalah: Baud rate 9600 bps, 8 data bit, 1 stop bit, tanpa parity bit, dan tanpa flow control.

Pada antarmuka RS-485, SPC SEVEN SEGMENT dapat difungsikan sebagai display karakter, display counter, display jam, dan display tanggal. Tabel karakter yang dapat ditampilkan oleh SPC SEVEN SEGMENT dapat dilihat pada **lampiran E**. Pada antarmuka UART RS-485, SPC SEVEN SEGMENT dapat di-cascade hingga 256 buah device. Timing diagram untuk pengiriman dan pembacaan data RS-485 dapat dilihat pada gambar berikut.



SIMBOL	KETERANGAN	NILAI		SATUAN
		MIN	MAX	
t _{stwr}	Jarak antar write state	100	–	μS
t _{wrrd}	Jarak antar write → read state	5	–	mS
t _{strd}	Jarak antar read state	100	–	μS

Protokol SPC SEVEN SEGMENT terdapat pada gambar berikut.



Daftar command dan protokol lengkap terdapat pada **lampiran C dan D**.

3.3. DRIVER DAN RUTIN

SPC SEVEN SEGMENT dilengkapi dengan driver **7S_4BIT.INC**, **7S_SPL.INC**, **7S_232.INC**, dan **7S_485.INC** yang akan mempermudah user dalam pemrograman. Keempat driver menggunakan resource dari mikrokontroler 89C51 sebagai berikut :

- Internal RAM alamat 21h bit 0 dan 1 (kecuali 7S_4BIT.INC)
- Internal RAM dengan alamat 30h – 31h (7S_4BIT.INC), 30h – 46h (7S_232.INC), atau 30h – 47h (7S_SPL.INC dan 7S_485.INC)
- P1.0 – P1.6 (7S_4BIT.INC), P1.4 – P1.7 (7S_SPL.INC), atau P3.0 – P3.1 (7S_232.INC dan 7S_485.INC)

Sehingga tidak boleh dipakai oleh user untuk keperluan lain, kecuali user mampu melakukan modifikasi pengaturan memori dengan benar.

Keempat driver ini menggunakan beberapa register dan flag yang terdiri dari:

SSCOLOUMN	SSHOUR	SSROMADDR
SSCHARACTER	SSMINUTE	SSROMDATA
SSBUFFER	SSSECOND	SSADDRESS
SSCONTRAST	SSDATE	DEVADDRESS
SSDISPLAY	SSMONTH	MODE12
SSRTCSTAT	SSYEAR	PM
SSBLINK		

Register dan flag tersebut digunakan dalam rutin-rutin sebagai berikut (urutan pengiriman/penerimaan data sesuai dengan urutan tulisan) :

S7_INITIALIZE232

Fungsi : Melakukan inisialisasi baud rate pada antarmuka UART RS-232.

Input : -

Output : -

Keterangan :

- ❖ Rutin ini adalah hanya terdapat pada ENG_232.INC.

Metode : Jika antarmuka yang digunakan adalah UART RS-232, panggil rutin S7_INITIALIZE232 sebelum menggunakan rutin-rutin lain.

S7_INITIALIZE485

Fungsi : Melakukan inisialisasi baud rate pada antarmuka UART RS-485.

Input : -

Output : -

Keterangan :

- ❖ Rutin ini adalah hanya terdapat pada ENG_485.INC.

Metode : Jika antarmuka yang digunakan adalah UART RS-485, panggil rutin S7_INITIALIZE485 sebelum menggunakan rutin-rutin lain.

S7_WRITECHARACTER

Fungsi : Menulis karakter pada kolom N.

Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)

SSCOLUMN → kolom N (1 – 8)

SSCHARACTER → kode karakter (0H – 32H)

Output : -

Keterangan :

- ❖ Rutin ini adalah satu-satunya rutin untuk antarmuka 4 bit Paralel.

- ❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 dan 4 Bit Paralel tidak menggunakan SSADDRESS.

Metode : Isi parameter input dan panggil rutin S7_WRITECHARACTER.

S7_READCHARACTER

Fungsi : Membaca karakter pada kolom N.

Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)

SSCOLUMN → kolom N (1 – 8)

Output : SSCHARACTER → kode karakter (0H – 32H)

Keterangan :

- ❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 tidak menggunakan SSADDRESS.

Metode : Isi parameter input dan panggil rutin S7_READCHARACTER. Hasil akan terdapat pada parameter output.

S7_SETDISPLAY

Fungsi : Mengatur mode display, contrast level, RTC on-off, dan blinking.

Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)

SSDISPLAY → mode display (0 – 3)

SSCONTRAST → contrast level (1 – 5)
SSRTCSTAT → RTC off (0), RTC on (1)
SSBLINK → blinking off (0), blinking on (1)

Output : -
Keterangan :
❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 tidak menggunakan SSADDRESS.
❖ Mode display 0 = display karakter.
❖ Mode display 1 = display counter.
❖ Mode display 2 = display jam.
❖ Mode display 3 = display tanggal.
Metode : Isi parameter input dan panggil rutin S7_SETDISPLAY.

S7_RESETCOUNTER

Fungsi : Mengisi nilai counter menjadi 0000000.
Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)
Output : -
Keterangan :
❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 tidak menggunakan SSADDRESS.
Metode : Isi parameter input dan panggil rutin S7_RESETCOUNTER.

S7_PRESETCOUNTER

Fungsi : Memberi nilai awal pada counter.
Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)
SSBUFFER → counter byte 0 (0 – 9)
...
SSBUFFER + 6 → counter byte 6 (0 – 9)
SSBUFFER + 7 → positive sign (0), negative sign (1)
Output : -
Keterangan :
❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 tidak menggunakan SSADDRESS.
❖ SSBUFFER menunjukkan digit satuan (Least Significant Digit/LSD).
❖ SSBUFFER + 6 menunjukkan digit jutaan (Most Significant Digit/MSD).
Metode : Isi parameter input dan panggil rutin S7_PRESETCOUNTER.

S7_SETRTC

Fungsi : Mengisi nilai jam, menit, detik RTC, periode, serta mode jam.
Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)

SSHOUR → jam (0 – 12) atau (0 – 23)
MODE12 → mode 24 jam (0) atau 12 jam (1)
PM → AM (0) atau PM (1) dalam mode 12 jam
SSMINUTE → menit (0 – 59)
SSSECOND → detik (0 – 59)

Output : -

Keterangan :

- ❖ SSHOUR bernilai antara 0 – 12 untuk mode 12 jam dan 0 – 23 untuk mode 24 jam.
- ❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 tidak menggunakan SSADDRESS.

Metode : Isi parameter input dan panggil rutin S7_SETRTCTIME.

S7_SETRTCDATE

Fungsi : Mengisi nilai tanggal, bulan, dan tahun RTC.

Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)
SSDATE → tanggal (0 – 31)
SSMONTH → bulan (0 – 12)
SSYEAR → tahun (0 – 99)

Output : -

Keterangan :

- ❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 tidak menggunakan SSADDRESS.

Metode : Isi parameter input dan panggil rutin S7_SETRTCDATE.

S7_READRTCTIME

Fungsi : Membaca jam, menit, detik RTC, periode, serta mode jam.

Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)
Output : SSHOUR → jam (0 – 12) atau (0 – 23)
MODE12 → mode 24 jam (0) atau 12 jam (1)
PM → AM (0) atau PM (1) dalam mode 12 jam
SSMINUTE → menit (0 – 59)
SSSECOND → detik (0 – 59)

Keterangan :

- ❖ SSHOUR bernilai antara 0 – 12 untuk mode 12 jam dan 0 – 23 untuk mode 24 jam.
- ❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 tidak menggunakan SSADDRESS.

Metode : Isi parameter input dan panggil rutin S7_READRTCTIME.
Hasil akan terdapat pada parameter output.

S7_READRTCDATE

Fungsi : Membaca tanggal, bulan, dan tahun RTC.

Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)

- Output : SSDATE → tanggal (0 – 31)
SSMONTH → bulan (0 – 12)
SSYEAR → tahun (0 – 99)
- Keterangan : ❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 tidak menggunakan SSADDRESS.
- Metode : Isi parameter input dan panggil rutin S7_READRTCDATE. Hasil akan terdapat pada parameter output.

S7_READCOUNTER

- Fungsi : Membaca nilai counter yang tersimpan dalam memori.
- Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)
- Output : SSBUFFER → counter byte 0 (0 – 9)
...
SSBUFFER + 6 → counter byte 6 (0 – 9)
SSBUFFER + 7 → positive sign (0), negative sign (1)
- Keterangan : ❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 tidak menggunakan SSADDRESS.
❖ SSBUFFER menunjukkan digit satuan (Least Significant Digit/LSD).
❖ SSBUFFER + 6 menunjukkan digit jutaan (Most Significant Digit/MSD).
- Metode : Isi parameter input dan panggil rutin S7_READCOUNTER. Hasil akan terdapat pada parameter output.

S7_WRITEEEPROM

- Fungsi : Menulis data di alamat tertentu pada EEPROM SPC SEVEN SEGMENT.
- Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)
SSROMADDR → alamat EEPROM (0 – 99)
SSROMDATA → data EEPROM (0 – 255)
- Output : -
- Keterangan : ❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 tidak menggunakan SSADDRESS.
- Metode : Isi parameter input dan panggil rutin S7_WRITEEEPROM.

S7_READEEPROM

- Fungsi : Membaca data di alamat tertentu pada EEPROM SPC SEVEN SEGMENT.
- Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)
SSROMADDR → alamat EEPROM (0 – 99)
- Output : SSROMDATA → data EEPROM (0 – 255)

- Keterangan : ❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 tidak menggunakan SSADDRESS.
- Metode : Isi parameter input dan panggil rutin S7_READEEPROM. Hasil akan terdapat pada parameter output.

S7 COUNTUP

- Fungsi : Menambah nilai counter dengan nilai 1.
- Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)
- Output : -
- Keterangan : ❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 tidak menggunakan SSADDRESS.
- Metode : Isi parameter input dan panggil rutin S7_COUNTUP.

S7 COUNTDOWN

- Fungsi : Mengurangi nilai counter dengan nilai 1.
- Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)
- Output : -
- Keterangan : ❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 tidak menggunakan SSADDRESS.
- Metode : Isi parameter input dan panggil rutin S7_COUNTDOWN.

S7 SETADDRESS

- Fungsi : Memberi nilai alamat baru pada SPC SEVEN SEGMENT.
- Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)
DEVADDRESS → alamat baru SPC SEVEN SEGMENT (0 – 255)
- Output : -
- Keterangan : ❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 tidak menggunakan SSADDRESS.
❖ Dengan antarmuka UART RS-232, rutin ini dapat digunakan untuk memberi nilai alamat awal untuk SPC SEVEN SEGMENT.
- Metode : Isi parameter input dan panggil rutin S7_SETADDRESS.

S7 READADDRESS

- Fungsi : Membaca alamat SPC SEVEN SEGMENT.
- Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)

Output : DEVADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)

Keterangan :

- ❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 tidak menggunakan SSADDRESS.
- ❖ Dengan antarmuka SPI atau UART RS-485, rutin ini dapat digunakan untuk mencari alamat SPC SEVEN SEGMENT yang terhubung.

Metode : Isi parameter input dan panggil rutin S7_READADDRESS. Hasil akan terdapat pada parameter output.

S7_CLEARDISPLAY

Fungsi : Membersihkan tampilan (hanya untuk mode display karakter)

Input : SSADDRESS → alamat SPC SEVEN SEGMENT (0 – 255)

Output : -

Keterangan :

- ❖ Untuk antarmuka SPI dan UART RS-485, SSADDRESS harus diisi dengan alamat SPC SEVEN SEGMENT yang dikehendaki. Antarmuka UART RS-232 tidak menggunakan SSADDRESS.

Metode : Isi parameter input dan panggil rutin S7_CLEARDISPLAY.

S7_CLEARSTATE

Fungsi : Melakukan reset terhadap jalur komunikasi yang digunakan.

Input : -

Output : -

Keterangan :

- ❖ Untuk antarmuka UART RS-232 dan UART RS-485, S7_CLEARSTATE akan mengirimkan 4 byte FFH.
- ❖ Untuk antarmuka SPI, S7_CLEARSTATE akan mengirimkan pulsa Clear State (lihat di timing diagram bagian 3.2.2).

Metode : Jika proses pengiriman command sebelumnya terputus/tidak sempurna, panggil rutin S7_CLEARSTATE terlebih dahulu sebelum memulai pengiriman baru.

3.4. CONTOH APLIKASI DAN PROGRAM

Apabila ingin menampilkan tulisan “HELLO” dengan SPI-bus, UART RS-232, atau UART RS-485, maka register-register yang harus diisi dan listing programnya adalah:

Cuplikan Listing Program:

```
;UART RS-232 tidak membutuhkan SSADDRESS
MOV  SSADDRESS, #0           ;untuk modul spc alamat 000

MOV  SSDISPLAY, #0          ;mode display karakter
MOV  SSCONTRAST, #5         ;contrast level = 5
```

```

MOV    SSRTCSTAT, #1           ;rtc run
MOV    SSBLINK, #0            ;blinking off
ACALL  S7_SETDISPLAY

MOV    SSCOLOUMN, #1
MOV    SSCHARACTER, #1DH      ;H
ACALL  S7_WRITECHARACTER
ACALL  DELAY

MOV    SSCOLOUMN, #2
MOV    SSCHARACTER, #1AH      ;E
ACALL  S7_WRITECHARACTER
ACALL  DELAY

MOV    SSCOLOUMN, #3
MOV    SSCHARACTER, #21H      ;L
ACALL  S7_WRITECHARACTER
ACALL  DELAY

MOV    SSCOLOUMN, #4
MOV    SSCHARACTER, #21H      ;L
ACALL  S7_WRITECHARACTER
ACALL  DELAY

MOV    SSCOLOUMN, #5
MOV    SSCHARACTER, #23H      ;o
ACALL  S7_WRITECHARACTER
ACALL  DELAY

```

3.5. KERANGKA PROGRAM

Bagi user yang ingin membuat program aplikasi SPC SEVEN SEGMENT dengan menggunakan rutin yang sudah ada, maka ada beberapa driver berikut harus dimasukkan (include).

ENG_4BIT.INC dan *7S_4BIT.INC* merupakan driver yang akan selalu digunakan untuk setiap aplikasi SPC SEVEN SEGMENT yang menggunakan antarmuka 4 Bit Paralel.

ENG_SPI.INC dan *7S_SPI.INC* merupakan driver yang akan selalu digunakan untuk setiap aplikasi SPC SEVEN SEGMENT yang menggunakan antarmuka SPI.

ENG_232.INC dan *7S_232.INC* merupakan driver yang akan selalu digunakan untuk setiap aplikasi SPC SEVEN SEGMENT yang menggunakan antarmuka UART RS-232.

ENG_485.INC dan *7S_485.INC* merupakan driver yang akan selalu digunakan untuk setiap aplikasi SPC SEVEN SEGMENT yang menggunakan antarmuka UART RS-485.

Kerangka pemrograman SPC SEVEN SEGMENT menggunakan Assembler MetaLink ASM51[®] sebagai berikut :

1. TMPT4BIT.ASM:

```

;-----
;FILE TEMPLATE UNTUK SPC SEVEN SEGMENT
;DENGAN DT51-MINSYS DAN ANTARLUKA 4 BIT PARALLEL
;-----
$MOD51
    CSEG
    ORG    4000H

```

```

LJMP  START
ORG   4100H
$INCLUDE (ENG_4BIT.INC) ;DRIVER UNTUK ANTARMUKA
$INCLUDE (7S_4BIT.INC) ;4 BIT PARALLEL

```

```

START:  MOV   SP, #50H
        .
        .
        .
        END

```

2. TMPTSPI.ASM:

```

;-----
;FILE TEMPLATE UNTUK SPC SEVEN SEGMENT
;DENGAN DT51-MINSYS DAN ANTARMUKA SPI
;-----
$MOD51
CSEG
ORG   4000H
LJMP  START
ORG   4100H
$INCLUDE (ENG_SPI.INC) ;DRIVER UNTUK ANTARMUKA SPI
$INCLUDE (7S_SPI.INC) ;

START:  MOV   SP, #50H
        .
        .
        .
        END

```

3. TMPT232.ASM:

```

;-----
;FILE TEMPLATE UNTUK SPC SEVEN SEGMENT
;DENGAN DT51-MINSYS DAN ANTARMUKA UART RS-232
;-----
$MOD51
CSEG
ORG   4000H
LJMP  START
ORG   4100H
$INCLUDE (ENG_232.INC) ;DRIVER UNTUK ANTARMUKA
$INCLUDE (7S_232.INC) ;UART RS-232

START:  MOV   SP, #50H
        LCALL S7_INITIALIZE232
        .
        .
        .
        END

```

4. TMPT485.ASM:

```

;-----
;FILE TEMPLATE UNTUK SPC SEVEN SEGMENT
;DENGAN DT51-MINSYS DAN ANTARMUKA UART RS-485
;-----
$MOD51
CSEG
ORG   4000H
LJMP  START
ORG   4100H
$INCLUDE (ENG_485.INC) ;DRIVER UNTUK ANTARMUKA

```



```

        $INCLUDE (7S_485.INC)      ;UART RS-485

START:   MOV     SP, #50H
        LCALL  7S_INITIALIZE485
        .
        .
        .
        END

```

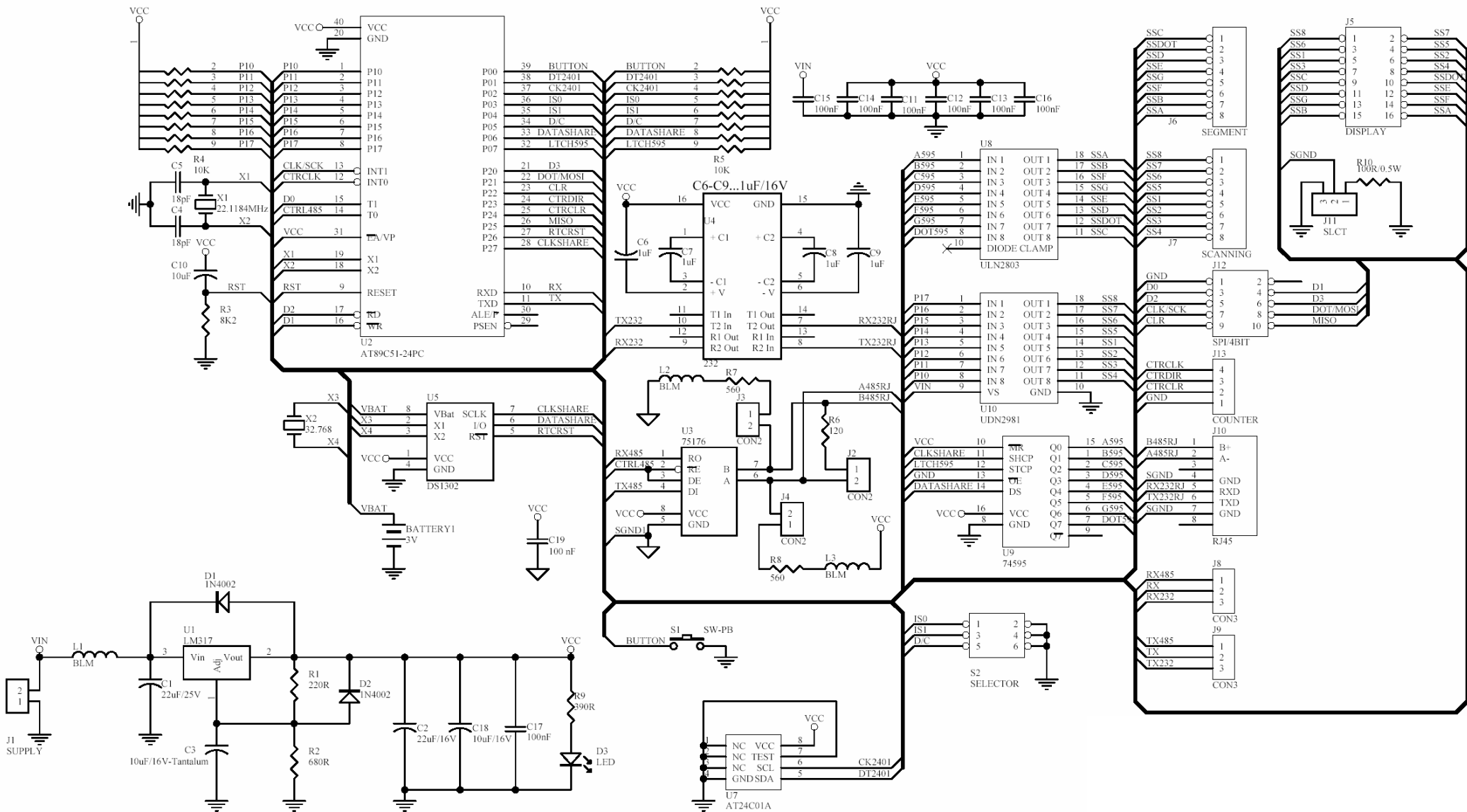
Hal Yang Perlu Diperhatikan

- Apabila batere lithium 3V DC pada SPC SEVEN SEGMENT dilepas, maka setting SPC SEVEN SEGMENT akan kembali pada nilai-nilai defaultnya yaitu :
 - RTC stop
 - Mode display : display karakter
 - Waktu : 00 : 00 : 00
 - Tanggal : 01/01/2001
 - Contrast level : 5
 - Blinking : off

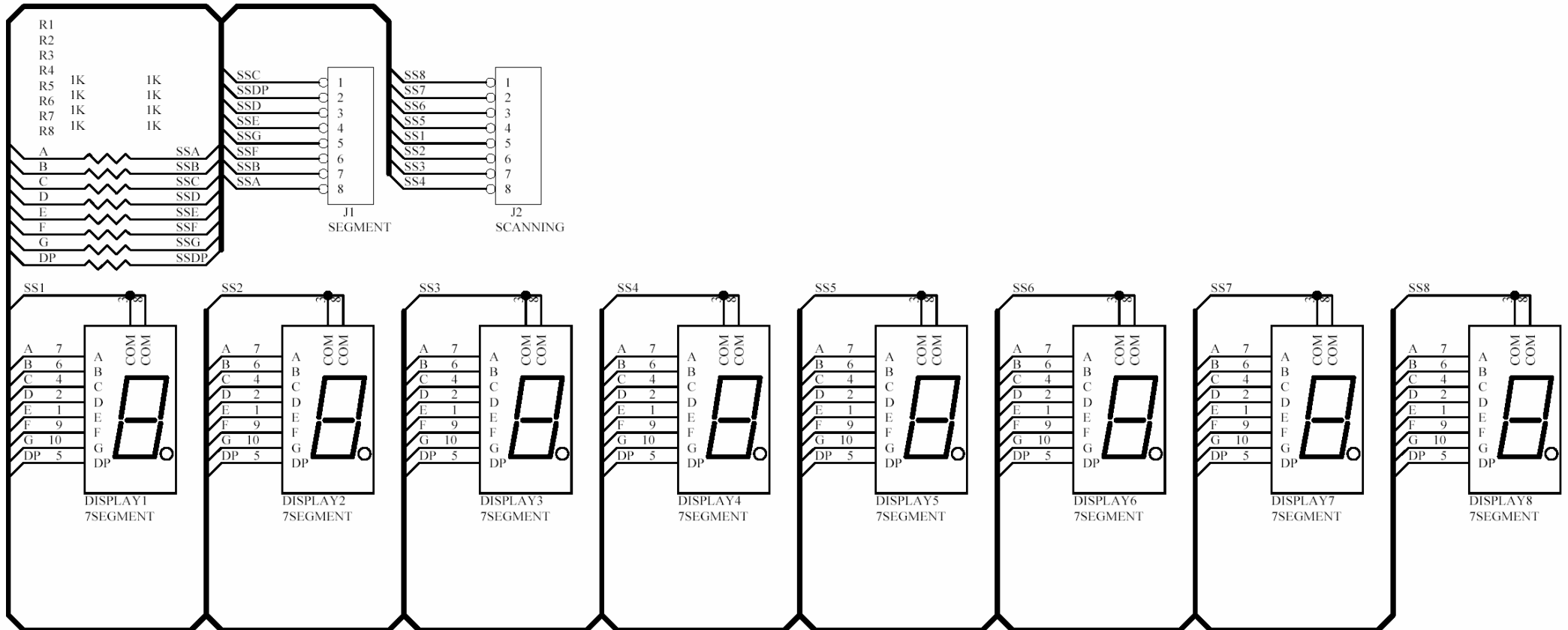
◆ *Terima Kasih atas kepercayaan Anda menggunakan produk kami, bila ada kesulitan, pertanyaan atau saran mengenai produk ini silahkan menghubungi technical support kami :*

support@innovativeelectronics.com

Lampiran A : Skema Modul SPC SEVEN SEGMENT



Lampiran B : Skema Modul Display Seven Segment



Lampiran C : Daftar Command SPC SEVEN SEGMENT Mode Display (kecuali antarmuka 4 Bit Paralel)

Command	Hex	Keterangan	Par. In	Par. Out
00010000B	10H	Write char to column N	2	–
00010001B	11H	Read char from column N	1	1
00010010B	12H	Set display	1	–
00010011B	13H	Counter reset	–	–
00010100B	14H	Counter preset	8	–
00010101B	15H	Set RTC time	3	–
00010110B	16H	Set RTC date	3	–
00010111B	17H	Read RTC time	–	3
00011000B	18H	Read RTC date	–	3
00011001B	19H	Read counter	–	8
00011010B	1AH	Write EEPROM	2	–
00011011B	1BH	Read EEPROM	1	1
00011100B	1CH	Count up	–	–
00011101B	1DH	Count down	–	–
00011110B	1EH	Set device address	1	–
00011111B	1FH	Read device address	–	1
00100000B	20H	Clear Display (Display Mode)	–	–
11111111B	FFH	Reset state	3	–

Lampiran D : Protokol SPC SEVEN SEGMENT Mode Display (kecuali antarmuka 4 Bit Parallel)

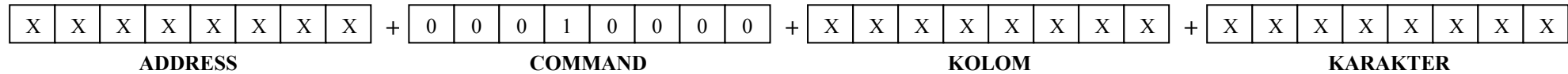
X	X	X	X	X	X	X	X
---	---	---	---	---	---	---	---

 : Data dari master controller
(komputer/minimum system)

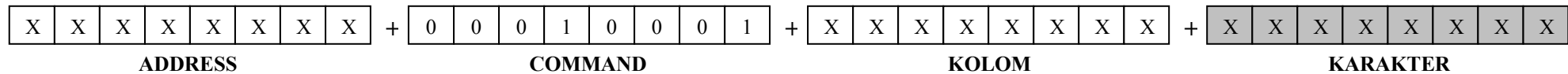
X	X	X	X	X	X	X	X
---	---	---	---	---	---	---	---

 : Data dari SPC Seven Segment

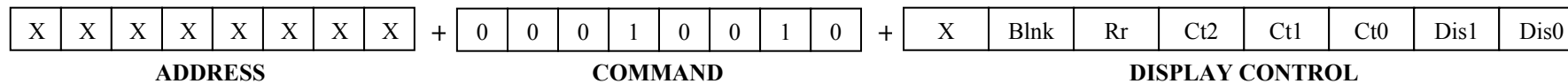
Protokol Write Character



Protokol Read Character



Protokol Set Display



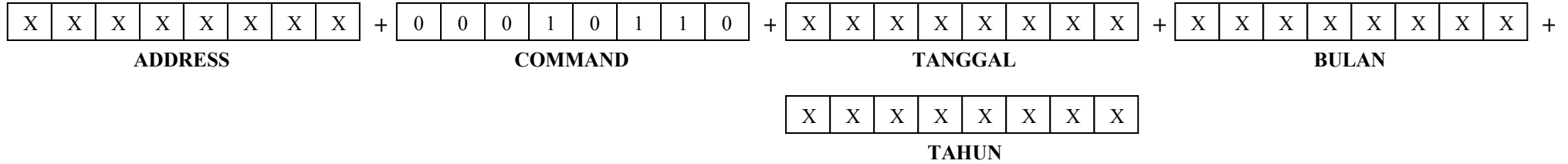
Dis1	Dis0	Mode display
0	0	Display Karakter
0	1	Display Counter
1	0	Display Jam
1	1	Display Tanggal

Ct2	Ct1	Ct0	Contrast
0	0	1	Level 1
0	1	0	Level 2
0	1	1	Level 3
1	0	0	Level 4
1	0	1	Level 5

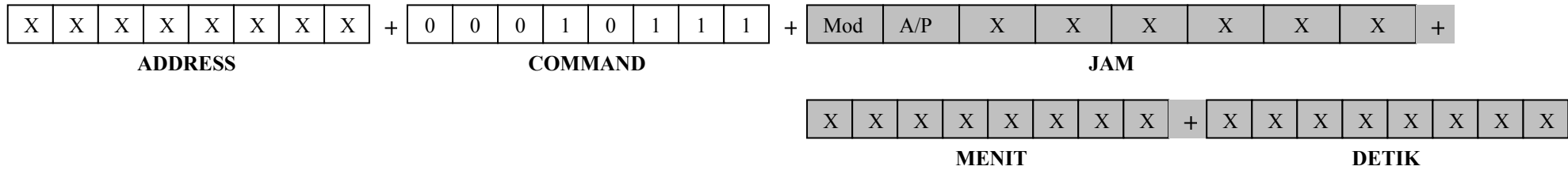
Rr	RTC
0	Off
1	On

A/P	Periode
0	AM
1	PM

Protokol Set RTC Date



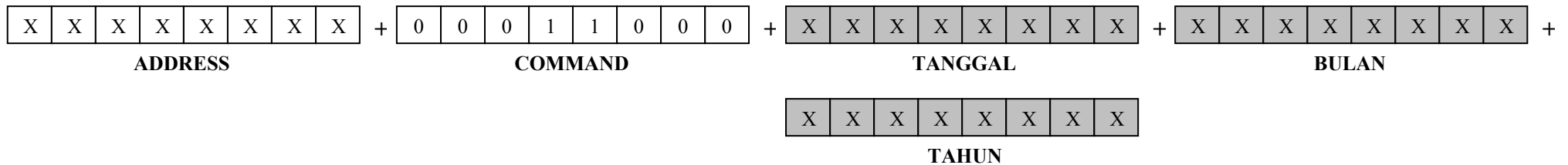
Protokol Read RTC Time



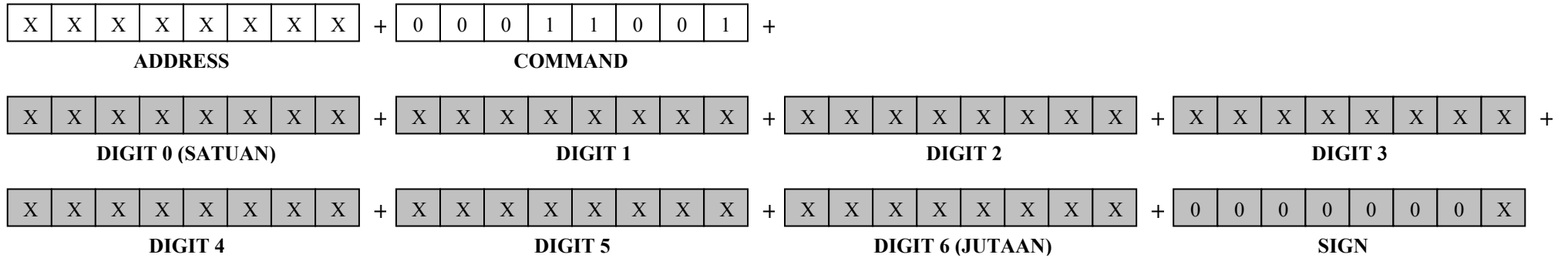
Mod	Mode Jam
0	24 Jam
1	12 Jam

A/P	Periode
0	AM
1	PM

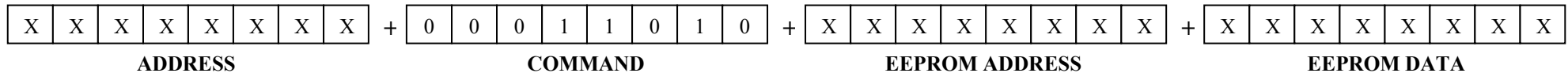
Protokol Read RTC Date



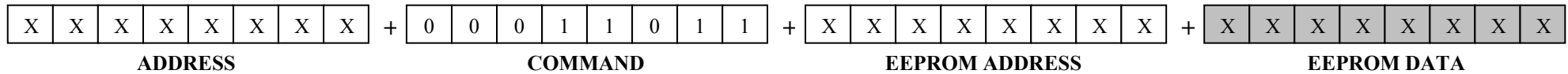
Protokol Read Counter



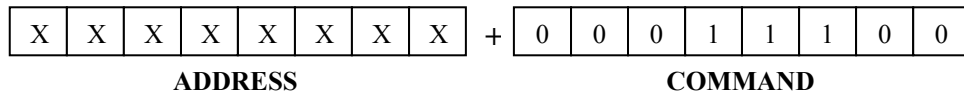
Protokol Write EEPROM



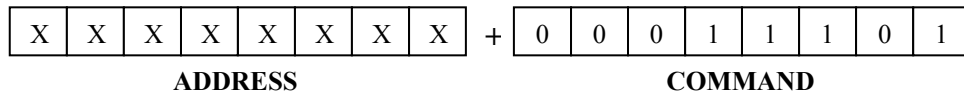
Protokol Read EEPROM



Protokol Count Up



Protokol Count Down



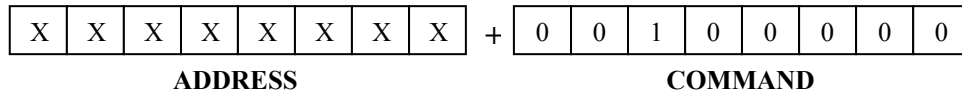
Protokol Set Device Address



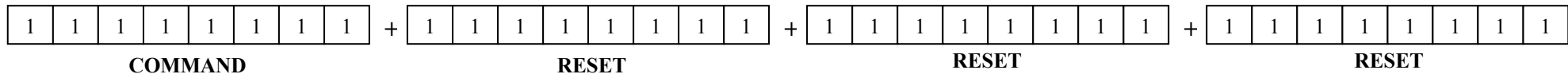
Protokol Read Device Address



Protokol Clear Display



Protokol Reset State (UART RS-232 dan UART RS-485)



Lampiran E : Tabel Karakter SPC SEVEN SEGMENT Mode Display (kecuali antarmuka 4 Bit Paralel)

Upper Lower	000	001	010	011	100	101	110
000							
001							
010							
011							
100							
101							
110							
111							