



TSD-5A(RS232 Type)



1. 개요

TSD-5A(Twin Segment Display)모델은 RS232C 시리얼 통신을 통해 FND모듈 10개를 하나의 칩에서 다이내믹 디스플레이 가능하도록 개발된 DLDC-7S/R10칩을 사용하여 설계된 제품으로서 특징으로서 Blink, 도토 표기, BCD변환, Hex표기 등의 다양한 기능을 내장하고 있으며 사용자는 각 자리에 Ascii코드를 전송함으로써 해당문자를 쉽게 표시할 수 있는 장점을 지니고 있습니다. 또한 확장 명령어를 제공, 각 세그먼트를 개별제어가 가능하기에 사용자 정의 캐릭터를 표시할 수 있는 자유로움을 제공하고 있습니다. TSD-5A는 컴파일명령어와 호환성모드와 원칩솔루션스 명령어모드 두 가지를 간편하게 설정할 수 있도록 DIP 스위치를 제공하며 원칩솔루션스 명령어의 특징은 모든 명령어가 4byte기준으로 설계되었으며 에러검출기능을 갖고 있어 심한 노이즈 환경하에서도 신뢰성있는 디스플레이를 보장합니다.

2. 일반사항

- 시리얼 입력만으로 5자리*2라인(5 By 2)의 7세그먼트에 영, 숫자 표시 가능
- 프로토콜 RS232C (5V, 19200/9600, 8, N, 1)
- DIP 스위치 조정으로 8개의 ID 부여 가능(어드레스 설정페이지 참조)
- BCD, HEX 숫자, 영문자, 특수기호, DOT 표시 가능 (아래의 테이블 참조)
- 특정 위치의 FND를 FLASHING 할 수 있음
- 특정 위치의 SEGMENT를 개별 제어 할 수 있음(New)
- 캐소드(cathod) 커먼 타입의 FND사용(*)
- 컴파일 호환 명령 및 원칩솔루션스 명령지원

3. TSD의 기본 문자 코드 테이블(ASCII 코드와 호환)

		상위 니블					
Hex	2	3	4	5	6	7	
0	Blank	0		P		P	
1	]	1	A	q	A	q	
2	_	2	b	r	b	r	
3	=	3	C	S	c	S	
4	≡	4	d	T	d	t	
5	—	5	E	U	e	u	
6	┌	6	F		F		
7	└	7	g		g		
8	┐	8	H		h		
9	┘	9	I	y	I	y	
A	U	.	J		j		
B	n	—		[			
C	└┐		L	└	l		
D	—		ñ	]	ñ		
E	☐		n		n		
F	□		O	_	o		

- 회색 칸의 문자는 ASCII코드와 차이가 있는 부분입니다.
- FND 표현 가능여부에 따라 대소문자가 혼용되었습니다.
- 특정 기호들은 해당기호 문자가 표시 불가능하기 때문에 최대한 비슷한 기호로 대체하였습니다.

4. DIP_SW 설명

TSD 뒷면에는 6개의 기능 조절용 DIP스위치가 존재하며, 각 Sw별 기능은 아래 도표를 참조 바랍니다.



	On	Off	설명
Sw1	8(4)	0	ID설정용
Sw2	4(2)	0	Target ID=0xE0+ SW값의 합 ※괄호() 는 원칩 솔루션스
Sw3	2(1)	0	Command Set일 경우의 값
Sw4	컴파일 호환	원칩 솔루션스	컴파일 호환 / 원칩 솔루션스 Command Set 선택
Sw5	On	Off	Check Sum 기능On/Off ※원칩솔루션스 Command Set에서 지원
Sw6	19200	9600	Baud Rate

※ID설정에 대해 부가 설명을 하자면 Dip SW 1~3은 2진법을 사용하여 설정하는데 컴파일 호환 Command Set일 경우 하나의 TSD-5A에 2개의 ID가 할당이 됩니다. 라인 1의 ID가 0xE0라면 라인 2의 ID는 +1을 하여



0xE10이 되는 것입니다. 이것은 컴파일 테크놀러지의 SGN과 Tiny PLC와의 호환성을 고려하였기에 불가피한 것이었습니다.

단, 원칩솔루션스 Command Set의 경우 TSD-5A 하나에 SGN처럼 하나의 ID가 할당이 되면 이때 자리수의 경우 라인 1이 1,2,3,4,5 라인 2가 6,7,8,9,10로 뒷줄부터 아랫줄까지 일련 되게 자리수가 매겨집니다.

자세한 내용은 원칩솔루션스 Command Set부분을 참조 바랍니다.

SW5 Check Sum기능의 경우 컴파일 혼란 Command set일 경우 Don't Care입니다.

TSD-5A의 기본 Baud는 컴파일의 SGN과 같이 기본 9600bps이나 Sw7의 조절로 19200까지 사용이 가능합니다.

5. 명령어 셋(Command Set)

TSD-5A에서는 널리 알려진 컴파일 테크놀러지사의 SGN Command Set외에도 원칩솔루션스 Command Set을 지원하며, 간단히 외부의 Dip SW조절을 통하여 Command Set을 변경할 수 있습니다.

먼저 컴파일(SGN) 호환 Command Set에 대해 설명한 다음 원칩솔루션스 Command Set에 대해 설명을 하겠습니다.

1)컴파일 호환 Command Set

■ 장치 ID

DIP-SW설정법에서 짧게 설명하였듯이 TSD-5A에는 컴파일 호환 Command Set일 경우 두개의 ID가 부여되며 라인2의 ID는 라인1의 ID에 +1을 한 것입니다.

빠른 이해를 위해 다음 예제 표를 참조바랍니다.

기본 Base ID는 0xE0입니다.

Sw 1	Sw 2	Sw 3	1번 라인 ID	2번 라인 ID
OFF(0)	OFF(0)	OFF(0)	$0xE0+0+0+0=0xE0$	$0xE0+1=0xE1$
ON(8)	OFF(0)	ON(2)	$0xE0+8+0+2=0xEA$	$0xEA+1=0xEB$
OFF(0)	ON(4)	OFF(0)	$0xE0+0+4+0=0xE4$	$0xE4+1=0xE5$

ID설정 예제

■ 각Digit별 위치



■ Digit 한계를 결정 명령

컴파일 SGN의 경우 모델에 따라 표시 자리 수 차이가 있었기에 지원되었던 명령으로 TSD-5A의 경우 5자리 고정 이기에 지원되지 않는 명령어로서 유저께서는 신경쓰지 않아도 되는 부분입니다

■ 기본 표시 명령

각자리의 데이터를 아스키 코드로 내보내는 명령으로서 각각의 자리를 개별적으로 제어하는 기능으로서 가장 기본적인 명령입니다.

ID	위치	데이터	설명
Target ID	1	ASCII	위치1의 자리에 ASCII값을 표시한다.
Target ID	2	ASCII	위치2의 자리에 ASCII값을 표시한다.
Target ID	3	ASCII	위치3의 자리에 ASCII값을 표시한다.
Target ID	4	ASCII	위치4의 자리에 ASCII값을 표시한다.
Target ID	5	ASCII	위치5의 자리에 ASCII값을 표시한다.

■ n진법 변환 표시 명령

어떤 2바이트 형식의 이진 데이터를 보내면 그 값을 10진 또는 16진 표시법으로 디스플레이 해주는 명령어로서 Host 장치에서 값을 변환하는 번거로움을 피할 수 있습니다.

ID	명령	Data High Byte	Data Low Byte	설명
Target ID	0xFA	데이터 상위바이트	데이터 하위 바이트	4자리(2,3,4,5위치) Hex값으로 표기
Target ID	0xFB	데이터 상위바이트	데이터 하위 바이트	5자리(위치1,2,3,4,5) BCD값으로 표기

명령 전송 포맷(4byte 길이)

■ Flash 명령

특정 위치를 Flash(깜박임) 표시하는 명령입니다. 위치와 명령어가 혼합된 형태입니다.

ID	위치	설명
Target ID	0xF0	전체 Flashing
Target ID	0xFF	Flash 정지
Target ID	0xF1	위치1의 자리 Flashing
Target ID	0xF2	위치2의 자리 Flashing
Target ID	0xF3	위치3의 자리 Flashing
Target ID	0xF4	위치4의 자리 Flashing
Target ID	0xF5	위치5의 자리 Flashing

■ Dot 제어 명령

특정 위치의 Dot를 On하거나 전체 OFF, 특정 도트를 Flashing 시키거나 전체 Flashing을 OFF하는 명령입니다. Dot가 Flashing하기 위해서는 Dot가 On이 되어있어야 합니다.

ID	위치	설명
Target ID	0xD0	전체 Dot Off
Target ID	0xD1	위치1의 Dot On
Target ID	0xD2	위치2의 Dot On
Target ID	0xD3	위치3의 Dot On
Target ID	0xD4	위치4의 Dot On
Target ID	0xD5	위치5의 Dot On
Target ID	0xD6	위치1의 Dot Flashing
Target ID	0xD7	위치2의 Dot Flashing
Target ID	0xD8	위치3의 Dot Flashing
Target ID	0xD9	위치4의 Dot Flashing
Target ID	0xDA	위치5의 Dot Flashing
Target ID	0xDF	전체 Dot Flashing OFF



■ 사용 예제

너무 널리 알려진 프로토콜이라 간단히 한 가지씩만 들어 설명하겠습니다. 좀 더 자세한 내용은 컴파일사의 SGN데이터 시트를 참조 바랍니다.

1) PicBasic의 경우

각 자리별로 ASCII코드를 통해 제어하는 예

```
SEROUT 8,30,0,0,[&HE0,1,"A"] '위치1에 "A"표시
SEROUT 8,30,0,0,[&HE0,2,"B"] '위치2에 "B"표시
SEROUT 8,30,0,0,[&HE0,3,"C"] '위치3에 "C"표시
SEROUT 8,30,0,0,[&HE0,4,"D"] '위치4에 "D"표시
SEROUT 8,30,0,0,[&HE0,4,"E"] '위치5에 "E"표시
```

2) CuBLOC의 경우

각 자리별로 ASCII코드를 통해 제어하는 예

Opencom 1, 9600, 3, 30, 20

Putstr 1,&hE0,1,&h33 'Rs232 채널1을 통하여 0xE0번지의
위치1에 3을 표시

※CuBLOC의 경우 원칩솔루션스 홈페이지 자료실의 기술
자료 게시판에 어플리케이션 노트로 TSD라이브러리를 제
공합니다.

3) PIC Micom의 경우

각 자리별로 ASCII코드를 통해 제어하는 예

```
printf("%c%c%c",0xe0,1,'0'); //위치1에 "0"표시
printf("%c%c%c",0xe0,2,'1'); //위치2에 "1"표시
```

2)원칩솔루션스 Command Set 및 장치 ID

원칩솔루션스 Command Set은 기존 Command Set의 문
제점인 도트명령중 개별 On기능은 있으나 개별 Off기능의
단점과 데이터 전송의 신뢰성 개선을 위해 만들어졌습니
다. n진법변환 표기(5byte) 명령어만 제외하면 4byte 포맷
(종료문자 0x00포함)으로 만들어졌으며

※Check Sum 기능에 대하여

Check Sum기능을 On했을 경우 앞의 명령과 데이터 바이
트를 XOR한 값을 종료문자의 0x00대신 보내면 됩니다.
TSD-5A는 Host로부터 온 명령을 분석하고 Check Sum
에 오류가 있을 경우 해당 명령을 무시해버리고 기존 문자
를 표기를 유지합니다. Feed Back은 TSD-5A에서 지원
하지 않습니다.(3선 방식과의 호환성)

이 Check Sum기능의 장점을 Host측에서 바라볼 때
RS-232프로토콜을 소프트웨어로 구성해서 프로그램을 작
성한다면 Host 마이컴의 경우 인터럽트 사용에 규제를 받
게 됩니다. 비동기 통신인 RS-232데이터를 보내는 도중에
인터럽트를 처리하다보면 해당 데이터는 깨지게 됩니다.
이때 Check Sum이 없는 디스플레이의 경우 엉뚱한 문자
나 공백 문자를 표기하는 경우가 대부분입니다. 이 경우
사람의 눈에도 이런 현상이 자주 보이게 되며 유저로부터
제품의 신뢰성을 떨어트리는 계기를 마련하게 됩니다. 허
나 Check Sum기능을 활용하다보면 깨진 데이터는 무시해
버리고 기존문자를 유지하기에 Host 마이컴 프로그램 시
행될 더 자유롭게 프로그램이 가능하며 무시된 데이터는
Host 측에서 주기적으로 디스플레이를 갱신시켜줌으로서
안정적인 표기를 할 수 있는 것입니다.

■ 장치 ID

원칩솔루션스 Command Set으로 설정했을 경우 TSD-5A
는 DIP-SW에 의해 하나의 ID를 부여받습니다.
빠른 이해를 돕기 위해 다음 예제 표를 참조바랍니다.

기본 Base ID는 0xE0입니다.

Sw 1	Sw 2	Sw 3	TSD-5A ID
OFF(0)	OFF(0)	OFF(0)	0xE0+0+0+0=0xE0
ON(4)	OFF(0)	ON(1)	0xE0+4+0+1=0xE5
OFF(0)	ON(2)	OFF(0)	0xE0+0+2+0=0xE2

ID설정 예제



■ 각Digit별 위치



※ID가 하나이고 위치가 1부터 10(0x0A)까지 있다는 것에 유의

■ 기본 표시 명령어

기본 표시 명령어는 SGN에서 각 자리를 ASCII코드에 의해 개별 제어가 기본이었던 것에 반에 각 자리의 세그먼트 제어 명령이 기본 표시 명령어입니다. 이는 I2C이용한 TSD-5B와의 일관성과 컴파일사의 CSGN과의 호환성을 고려한 것입니다. 다시 한 번 강조 하자면 원칩솔루션스 Command Set의 기본명령은 사용자정의 문자 표현이 가능한 세그먼트제어 명령입니다.

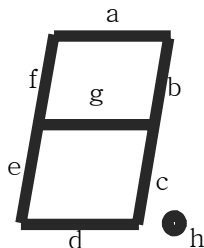
명령 순서는 ID+ 위치+ Data+ 종료코드(0x00)입니다. Check Sum기능이 On 되어있을 경우 종료코드(0x00)대신 Check Sum을 전송하시면 됩니다.

ID	위치	데이터	종료 Code / Check Sum	설명
Target ID	0x01	1Byte조합	0x00	위치1의 세그먼트를 제어.
Target ID	0x02	1Byte조합	0x00	위치2의 세그먼트를 제어.
Target ID	0x03	1Byte조합	0x00	위치3의 세그먼트를 제어.
Target ID	0x04	1Byte조합	0x00	위치4의 세그먼트를 제어.
Target ID	0x05	1Byte조합	0x00	위치5의 세그먼트를 제어.
Target ID	0x06	1Byte조합	0x00	위치6의 세그먼트를 제어.
Target ID	0x07	1Byte조합	0x00	위치7의 세그먼트를 제어.
Target ID	0x08	1Byte조합	0x00	위치8의 세그먼트를 제어.
Target ID	0x09	1Byte조합	0x00	위치9의 세그먼트를 제어.
Target ID	0x0A	1Byte조합	0x00	위치10의 세그먼트를 제어.

명령 전송 포맷(4byte 길이)

FND세그먼트와 1Byte조합의 Bit Map

Bit	7	6	5	4	3	2	1	0
세그먼트	h	g	f	e	d	c	b	a



■ ASCII 표시 명령

각 자리의 데이터를 ASCII코드로 보내는 명령으로서 각 자리의 자리를 개별적으로 제어하는 기능입니다.

ID	위치	ASCII	종료 Code / Check Sm	설명
Target ID	0xA1	code	0x00	위치1의 자리에 ASCII값을 표시
Target ID	0xA2	code	0x00	위치2의 자리에 ASCII값을 표시
Target ID	0xA3	code	0x00	위치3의 자리에 ASCII값을 표시
Target ID	0xA4	code	0x00	위치4의 자리에 ASCII값을 표시
Target ID	0xA5	code	0x00	위치5의 자리에 ASCII값을 표시
Target ID	0xA6	code	0x00	위치6의 자리에 ASCII값을 표시
Target ID	0xA7	code	0x00	위치7의 자리에 ASCII값을 표시
Target ID	0xA8	code	0x00	위치8의 자리에 ASCII값을 표시
Target ID	0xA9	code	0x00	위치9의 자리에 ASCII값을 표시
Target ID	0xAA	code	0x00	위치10의 자리에 ASCII값을 표시

■ n진법 변환 표시 명령

원칩솔루션스 명령으로서 유일하게 5byte 포맷이며 Check Sum기능이 지원되지 않는 명령으로서 어떤 2바이트 형식의 이진 데이터를 보내면 그 값을 10진 또는 16진 표시법으로 디스플레이 해주는 명령어로서 Master Device에서 값을 변환하는 번거로움을 피할 수 있습니다. TSD-5A는 DLDC-7S/R10칩을 사용하였기에 최대 10자리 표시능력을 지니고 있습니다. 따라서 2byte 데이터를 변환해서 표시하려면 10자리 중 5자리(BCD)또는 4자리(Hex)값의 첫 자리(높은 자리) 표시 위치를 정해야 합니다. 예를 들어 hex데이터 A3C2를 표기한다면 첫 자리는 A에 해당합니다. 따라서 A가 표시되기 시작할 위치를 말하는 것입니다.

ID	명령	위치	Data High Byte	Data Low Byte	설명
Target ID	0xCA	첫 자리 표시 위치	데이터 상위 바이트	데이터 하위 바이트	지정된 위치에 4자리 Hex값으로 표기
Target ID	0xCB	첫 자리 표시 위치	데이터 상위 바이트	데이터 하위 바이트	지정된 위치에 5자리 BCD값으로 표기

명령 전송 포맷(5byte 길이)

■ Flash 명령

특정 위치를 Flash(깜박임)을 On/Off하는 명령입니다. 기존 명령은 개별적으로 On하는 기능은 있었으나 개별적으로 Off하는 기능이 없었기에 명령 바로 뒤에 1(On)과 0(Off)를 보내어 해당 기능을 동작시킬지 해제시킬지 결정하도록 명령을 확장 시켰습니다.

ID	위치	On/Off	종료 Code / Check Sm	설명
Target ID	0xF1	1 / 0	0x00	위치1의 자리 Flash On/Off
Target ID	0xF2	1 / 0	0x00	위치2의 자리 Flash On/Off
Target ID	0xF3	1 / 0	0x00	위치3의 자리 Flash On/Off
Target ID	0xF4	1 / 0	0x00	위치4의 자리 Flash On/Off
Target ID	0xF5	1 / 0	0x00	위치5의 자리 Flash On/Off
Target ID	0xF6	1 / 0	0x00	위치6의 자리 Flash On/Off
Target ID	0xF7	1 / 0	0x00	위치7의 자리 Flash On/Off
Target ID	0xF8	1 / 0	0x00	위치8의 자리 Flash On/Off
Target ID	0xF9	1 / 0	0x00	위치9의 자리 Flash On/Off
Target ID	0xFA	1 / 0	0x00	위치10의 자리 Flash On/Off
Target ID	0xFF	1 / 0	0x00	전체 Flash On/Off



■ Dot 제어 명령

◇ Dot ON/OFF명령

전체 또는 특정 위치의 Dot를 On/Off하는 명령입니다.

ID	위치	On/Off	종료 Code / Check Sm	설명
Target ID	0xD1	1 / 0	0x00	위치1의 자리 Dot On/Off
Target ID	0xD2	1 / 0	0x00	위치2의 자리 Dot On/Off
Target ID	0xD3	1 / 0	0x00	위치3의 자리 Dot On/Off
Target ID	0xD4	1 / 0	0x00	위치4의 자리 Dot On/Off
Target ID	0xD5	1 / 0	0x00	위치5의 자리 Dot On/Off
Target ID	0xD6	1 / 0	0x00	위치6의 자리 Dot On/Off
Target ID	0xD7	1 / 0	0x00	위치7의 자리 Dot On/Off
Target ID	0xD8	1 / 0	0x00	위치8의 자리 Dot On/Off
Target ID	0xD9	1 / 0	0x00	위치9의 자리 Dot On/Off
Target ID	0xDA	1 / 0	0x00	위치10의 자리 Dot On/Off
Target ID	0xDF	1 / 0	0x00	전체 Dot On/Off

◇ Dot Flash On/Off 명령

전체 또는 특정 위치의 Dot를 Flash 시키거나 해제하는 명령으로서 Flash명령 전에 Dot가 켜져 있어야 Flash가 가능합니다.

ID	위치	On/Off	종료 Code / Check Sm	설명
Target ID	0xE1	1 / 0	0x00	위치1의 자리 Flash On/Off
Target ID	0xE2	1 / 0	0x00	위치2의 자리 Flash On/Off
Target ID	0xE3	1 / 0	0x00	위치3의 자리 Flash On/Off
Target ID	0xE4	1 / 0	0x00	위치4의 자리 Flash On/Off
Target ID	0xE5	1 / 0	0x00	위치5의 자리 Flash On/Off
Target ID	0xE6	1 / 0	0x00	위치6의 자리 Flash On/Off
Target ID	0xE7	1 / 0	0x00	위치7의 자리 Flash On/Off
Target ID	0xE8	1 / 0	0x00	위치8의 자리 Flash On/Off
Target ID	0xE9	1 / 0	0x00	위치9의 자리 Flash On/Off
Target ID	0xEA	1 / 0	0x00	위치10의 자리 Flash On/Off
Target ID	0xEF	1 / 0	0x00	전체 Flash On/Off

■ Refresh Rate

FND 갱신 주기를 변경하는 명령입니다. Default로 200Hz로 설정되어있습니다. 이를 통하여 Flash기능의 주기를 조절이 가능합니다.

Target 주소	명령	On/Off	종료 Code / Check Sm	설명
Write Mode 주소	0xC0	1/0	0x00	200Hz(On),100Hz(OFF)

명령 전송 포맷(4Byte 길이)

■ 특수 명령

명령 다음에 반전 데이터를 보내는 것으로 이뤄집니다. 여기서 반전이라는 것은 1의 보수를 뜻합니다.

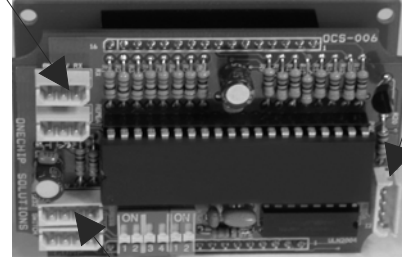
Target 주소	명령	반전	종료 Code / Check Sm	설명
Write Mode 주소	0xCC	0x33	0x00	전체 지우기 명령(Clear) -캐릭터만 지워짐 -기능명령 유효
Write Mode 주소	0xCE	0x31	0x00	리셋 명령(rEset)

명령 전송 포맷(4Byte 길이)

■ TSD-5시리즈 뒷면

PICBASIC이나 일반 마이크로 컴퓨터와의 접속을 위한 5V레벨의 RS232단자입니다.

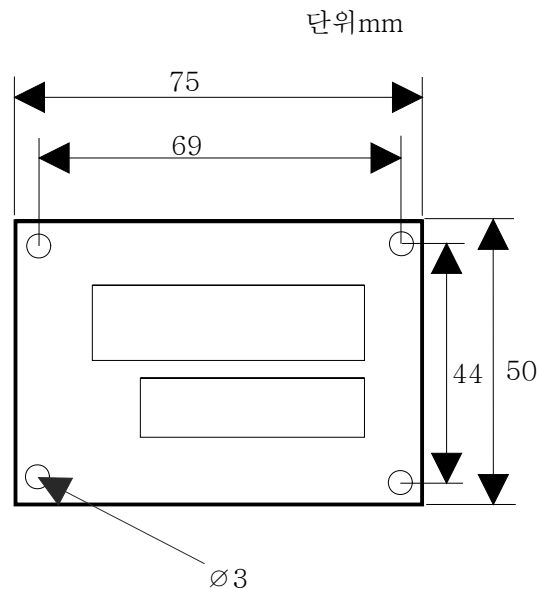
I²C연결단자
※I²C모델일 경우



PC와의 인터페이스를 위한 RS232포트입니다. PC의 D-Sub(3번(TX)핀을 EFX단자에 연결하시면 됩니다.

ID 및 기능설정을 위한 DIP스위치

■ 외형치수(단위:mm)



※제품에 전원투입 후 모듈이 명령을 받기위해 준비하는 대기 시간이 필요합니다. 전원투입 후 약 200ms후에 명령을 사용하시기 권장합니다. 모듈이 기동을 준비를 하는 동안에는 어떠한 명령에도 응답하지 않습니다. 이는 전원이 불안정한 환경 하에서 오동작을 방지하기위한 최소 권장 딜레이입니다.



■ 사용 예제

Check Sum기능Off상태와 On상태를 각각 예로 들어 설명하겠습니다.

◇Check Sum기능을 OFF했을 경우(DIP스위치5 OFF

1) PicBasic의 경우

각 자리수 별로 ASCII코드를 통해 제어하는 예

```
SEROUT 8,30,0,0,[&HE0,&HA1,"A",0] '위치1에 "A"표시
SEROUT 8,30,0,0,[&HE0,&HA2,"B",0] '위치2에 "B"표시
SEROUT 8,30,0,0,[&HE0,&HA3,"C",0] '위치3에 "C"표시
SEROUT 8,30,0,0,[&HE0,&HA4,"D",0] '위치4에 "D"표시
SEROUT 8,30,0,0,[&HE0,&HA4,&H30,0] '위치5에 "0"표시
```

INTEGER형 변수 I의 상위 바이트와 하위 바이트 값을
BCD 5자리로 위치6번부터 표시 예, 명령 &HCB

```
DIM I AS INTEGER
10 I = I + 1
SEROUT 8,30,0,0,[&HE0,6,&HCB,I.H,I.L]
DELAY 1
GOTO 10
```

INTEGER형 변수 I의 상위 바이트와 하위 바이트 값을
Hex 4자리로 위치7번부터 표시 예, 명령 &HCA

```
DIM I AS INTEGER
10 I = I + 1
SEROUT 8,30,0,0,[&HE0,7,&HCA,I.H,I.L]
DELAY 1
GOTO 10
```

Dot On / Off ,Flashing

```
SEROUT 8,30,0,0,[&HE0,&HD4,1,0] '위치4 도트on
SEROUT 8,30,0,0,[&HE0,&HD5,0,0] '위치5 도트off
SEROUT 8,30,0,0,[&HE0,&HE4,1,0] '위치4 도트flashing
```

데이터 Flashing

```
SEROUT 8,30,0,0,[&HE0,&HF4,1,0] '위치4 flash on
SEROUT 8,30,0,0,[&HE0,&HF4,0,0] '위치4 flash off
```

2) CuBLOC의 경우

※CuBLOC의 경우 원칩솔루션스 홈페이지 자료실의
기술자료 게시판에 어플리케이션 노트로 TSD라이브러리를
제공합니다.

각 자리수 별로 ASCII코드를 통해 제어하는 예

```
Opencom 1, 9600, 3, 30, 20 'Rs232 채널1을

Putstr 1,&hE0,&ha1,&h33,0 '0xE0번지의 위치1에 3을 표시
Putstr 1,&hE0,&ha2,"A",0 '0xE0번지의 위치2에 A를 표시
```

INTEGER형 변수 I의 상위 바이트와 하위 바이트 값을
BCD 5자리로 위치6번부터 표시 예, 명령 &HCB

```
Dim i as integer
do
i=i+1
Putstr 1,&hE0,&hcb,6,i.byte1,i.byte0
delay 100
loop
```

INTEGER형 변수 I의 상위 바이트와 하위 바이트 값을
Hex 4자리로 위치7번부터 표시 예, 명령 &HCA

```
Dim i as integer
do
i=i+1
Putstr 1,&he0,&hca,7,i.byte1,i.byte0
delay 100
loop
```

Dot On / Off ,Flashing

```
Putstr 1,&he0,&hd4,1,0 '위치4 도트on
Putstr 1,&he0,&hd5,0,0 '위치5 도트off
Putstr 1,&he0,&he4,1,0 '위치4 도트flashing
```

데이터 Flashing

```
Putstr 1,&he0,&hf4,1,0 '위치4 문자 flash On
Putstr 1,&he0,&hf4,0,0 '위치4 문자 flash Off
```

3) PIC Micom의 경우(CCS-C)

각 자리수 별로 ASCII코드를 통해 제어하는 예

```
printf("%c%c%c%c",0xe0,0xa6,'0',0); //위치 6에 "0"표시
printf("%c%c%c%c",0xe0,0xa7,0x30,0); //위치 7에 "0"표시
printf("%c%c%c%c",0xe0,0xa8,'1',0); //위치 8에 "1"표시
printf("%c%c%c%c",0xe0,0xa9,'2',0); //위치 9에 "2"표시
printf("%c%c%c%c",0xe0,0xaa,'3',0); //위치10에 "3"표시
```

INTEGER형 변수 I의 상위 바이트와 하위 바이트 값을
BCD 5자리로 위치6번부터 표시 예, 명령 &HCB

```
long i;
do{
i++;
printf("%c%c%c%c",0xe0,0xcb,6,i>>8,i);
delay_ms(1);
}
```

INTEGER형 변수 I의 상위 바이트와 하위 바이트 값을
Hex 4자리로 위치7번부터 표시 예, 명령 &HCA

```
long i;
do{
i++;
printf("%c%c%c%c",0xe0,0xca,6,i>>8,i);
delay_ms(1);
}
```

Dot On / Off ,Flashing

```
printf("%c%c%c%c",0xe0,0xd4,1,0); //위치4 도트on
printf("%c%c%c%c",0xe0,0xd5,0,0); //위치5 도트off
printf("%c%c%c%c",0xe0,0xe4,1,0); //위치4 도트flashing
```

데이터 Flashing



```
printf("%c%c%c%c", 0xe0, 0xf4, 1, 0); //위치4 문자 flash On
printf("%c%c%c%c", 0xe0, 0xf4, 0, 0); //위치4 문자 flash Off
```

◇Check Sum기능을 ON했을 경우(DIP스위치5 ON)

1) PicBasic의 경우

각 자리 별로 ASCII코드를 통해 제어하는 예

```
DIM CK AS Byte
CK=&HA XOR "A" 'CK는 명령과 데이터를 XOR한 값
SEROUT 8,30,0,0,[&HE0,&HA1,"A",CK] '위치1에 "A"표시
CK=&HA XOR "B"
SEROUT 8,30,0,0,[&HE0,&HA2,"B",CK] '위치2에 "B"표시
CK=&HA XOR "C"
SEROUT 8,30,0,0,[&HE0,&HA3,"C",CK] '위치3에 "C"표시
CK=&HA XOR "D"
SEROUT 8,30,0,0,[&HE0,&HA4,"D",CK] '위치4에 "D"표시
CK=&HA XOR &H30
SEROUT 8,30,0,0,[&HE0,&HA4,&H30,CK] '위치5에 "0"표시

※중첩 연산이 지원되지 않기에 두줄로 나눠서 표현하였습니다.
```

Dot On / Off ,Flashing

```
DIM CK AS Byte
CK=&HD4 XOR 1
SEROUT 8,30,0,0,[&HE0,&HD4,1,CK] '위치4 도트on
CK=&HD5 XOR 0
SEROUT 8,30,0,0,[&HE0,&HD5,0,CK] '위치5 도트off
CK=&HE4 XOR 1
SEROUT 8,30,0,0,[&HE0,&HE4,1,CK] '위치4 도트flashing
```

데이터 Flashing

```
DIM CK AS Byte
CK=&HF4 XOR 1
SEROUT 8,30,0,0,[&HE0,&HF4,1,CK] '위치4 flash on
CK=&HF4 XOR 0
SEROUT 8,30,0,0,[&HE0,&HF4,0,CK] '위치4 flash off
```

2) CuBLOC의 경우

※CuBLOC의 경우 원칩솔루션스 홈페이지 자료실의 기술자료 게시판에 어플리케이션 노트로 TSD라이브러리를 제공합니다.

각 자리 별로 ASCII코드를 통해 제어하는 예

```
Opencom 1, 9600, 3, 30, 20 'Rs232 채널1을
Putstr 1,&hE0,&ha1,&h33,(&ha1 xor &h33)
'0xE0번지의 위치1에 3을 표시
Putstr 1,&hE0,&ha2,"A",(&ha2 xor "A")
'0xE0번지의 위치2에 A를 표시
```

Dot On / Off ,Flashing

```
Putstr 1,&hE0,&hd4,1,(&hd4 xor 1) '위치4 도트on
Putstr 1,&hE0,&hd5,0,(&hd5 xor 0) '위치5 도트off
Putstr 1,&hE0,&he4,1,(&he4 xor 1) '위치4 도트flashing
```

데이터 Flashing

```
Putstr 1,&hE0,&hf4,1,(&hf4 xor 1) '위치4 문자 flash On
Putstr 1,&hE0,&hf4,0,(&hf4 xor 0) '위치4 문자 flash Off
```

3) PIC Micom의 경우(CCS-C)

각 자리 별로 ASCII코드를 통해 제어하는 예

```
printf("%c%c%c%c", 0xe0, 0xa6, '0', 0xa6^'0');
//위치 6에 "0" 표시
printf("%c%c%c%c", 0xe0, 0xa7, 0x30, 0xa7^0x30);
//위치 7에 "0" 표시
printf("%c%c%c%c", 0xe0, 0xa8, '1', 0xa8^'1');
//위치 8에 "1" 표시
printf("%c%c%c%c", 0xe0, 0xa9, '2', 0xa9^'2');
//위치 9에 "2" 표시
printf("%c%c%c%c", 0xe0, 0xaa, '3', 0xaa^'3');
//위치10에 "3" 표시
```

Dot On / Off ,Flashing

```
printf("%c%c%c%c", 0xe0, 0xd4, 1, 0xd4^1); //위치4 도트on
printf("%c%c%c%c", 0xe0, 0xd5, 0, 0xd5^0); //위치5 도트off
printf("%c%c%c%c", 0xe0, 0xe4, 1, 0xe4^1); //위치4
도트flashing
```

데이터 Flashing

```
printf("%c%c%c%c", 0xe0, 0xf4, 1, 0xf4^1); //위치4 문자 flash
On
printf("%c%c%c%c", 0xe0, 0xf4, 0, 0xf4^0); //위치4 문자 flash
Off
```

※제품에 전원투입 후 모듈이 명령을 받기위해 준비하는 시간이 필요합니다. 전원투입 후 약 200ms후에 명령을 사용하시기 권장합니다. 모듈이 기동준비를 하는 동안에는 명령에 응답하지 않습니다. 이는 전원이 불안정한 환경 하에서 오동작을 방지하기위한 최소 권장 딜레이입니다.

TSD-5A 매뉴얼

Last Modified Date : 2006.03.31