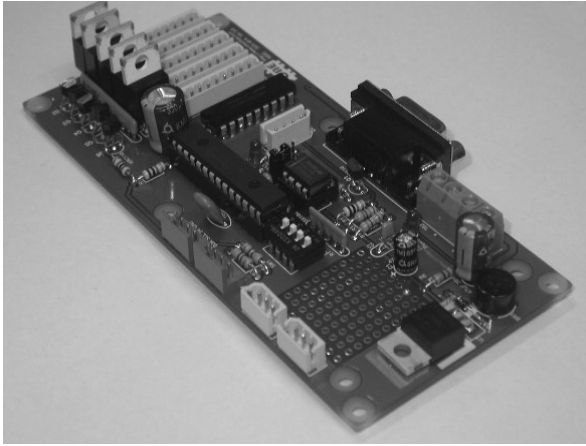


BFD-Con 2.0 (RS-232,485,I2C Type)



1.개요

BFD-Con은 RS232, RS485,I²C 시리얼 통신을 통해 FND 모듈 5개를 하나의 칩에서 다이내믹 디스플레이 가능하도록 개발된 설계된 제품으로서 특징으로서 Blink, 도토 표기, BCD변환, Hex표기 등의 다양한 기능을 내장하고 있으며 사용자는 각 자리에 ASCII코드를 전송함으로써 해당 문자를 쉽게 표시할 수 있는 장점을 지니고 있습니다. 또한 각 세그먼트를 개별제어가 가능하기에 사용자가 정의 캐릭터를 표시할 수 있는 자유로움을 제공하고 있습니다. BFD-Con는 기존 CSG, SGN명령어와 호환성을 고려하여 만들어 졌으며 원칩솔루션스 확장명령어를 지원하여 안정성을 증대시켰습니다.

2.일반사항

- 24V(AC,DC) 전원사용
- 시리얼통신으로 Big FND 에 7 세그먼트에 영, 숫자 표시 가능
- RS232,RS485,I²C 지원
- 프로토콜 - 컴파일호환Set, 원칩솔루션스 Set(OCS-Set)호환
- DIP 스위치 조정으로 8개의 ID 부여 가능
(그 이상의 ID는 문의바람)
- BCD, HEX 숫자, 영문자, 특수기호, DOT 표시 가능 (아래의 테이블 참조) , Flip기능,모아쓰기 기능
- 특정 위치의 FND를 FLASHING 할 수 있음
- 특정 위치의 FND를 Flip할 수 있음
- 특정 위치의 SEGMENT를 개별 제어 할 수 있음(New)
- 애노드(Anode) 커먼 타입의 FND사용(*)
- 컴파일 CuBLOC의 CSG라이브러리 호환

기존의 BFD-Con의 경우 RS232와 RS485통합모델과 I2C 모델이 하나의 보드에 서로 다른 펌웨어로 존재하였으나 2.0버전에서는 두 펌웨어가 하나의 칩에 혼합되어 동시에 3가지 통신 방식을 지원할 수 있게 되었음을 알려드립니다.

BFD-Con의 기본 문자 코드 테이블

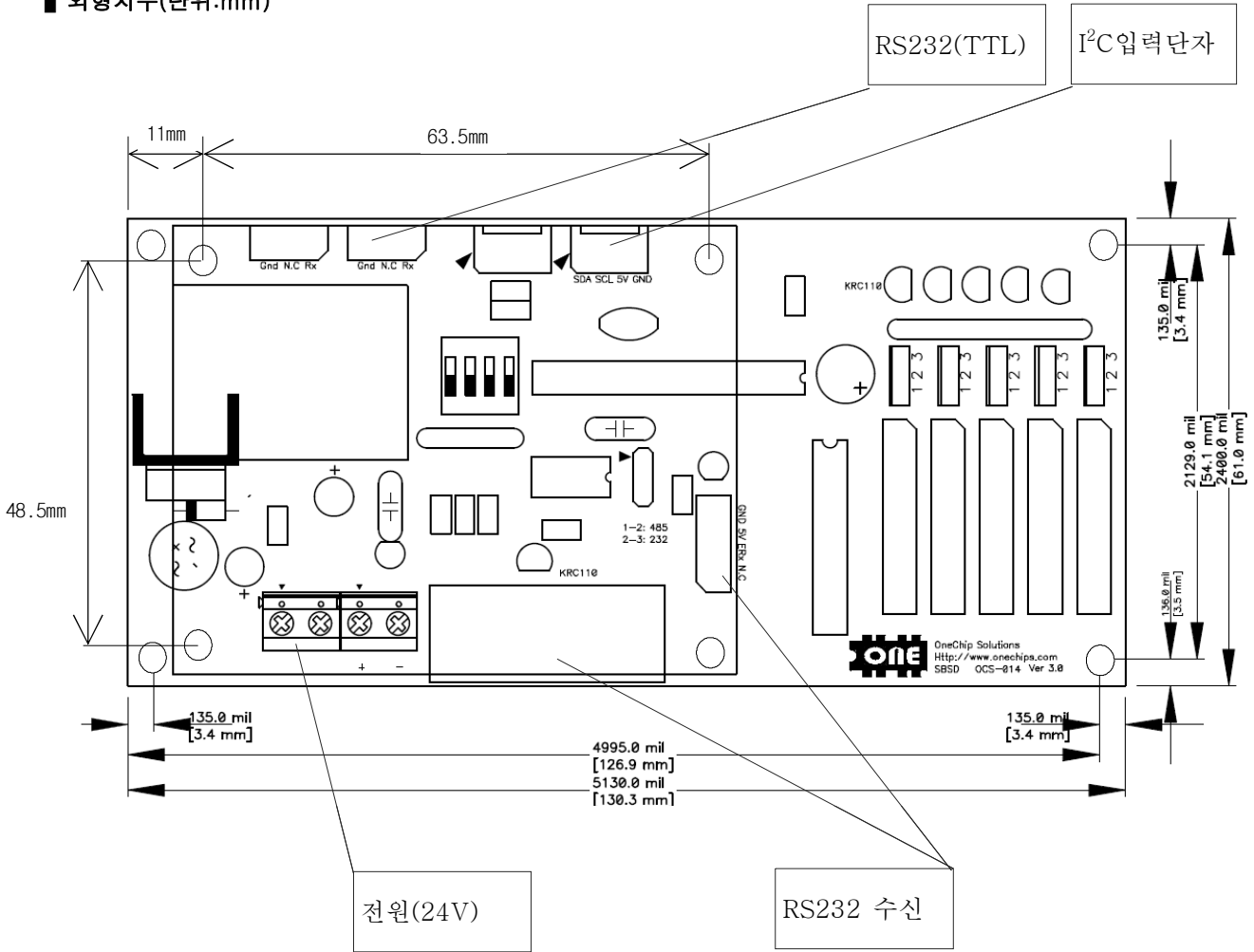
상위 니블

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	見	6	F		F	
7	ㄱ	7	g		g	
8	ㄴ	8	H		h	
9	ㄷ	9	l	y	l	y
A	U	.	J		j	
B	n	—		[
C	ㄹ	ㄷ	L	ㄱ	l	
D	—	ㄷ	n̄]	n̄	
E	ㄷ	ㄷ	n		n	
F	□	ㄷ	O	_	o	

(ASCII 코드와 호환)

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

■ 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 방지하기 위한 최소 권장 딜레이입니다.

RS232채널과 RS485채널은 동시 사용이 불가능합니다. I2C채널과 RS232또는 485채널은 동시 사용해도 가능하나 그런 중복된 사용은 권장하지 않습니다.

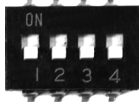
본제품은 종단 저항을 내장하고 있지 않습니다 RS485 채널 사용시 필요에 따라서 적절한 종단저항을 외부에 달아주셔야 합니다.

I2C통신(SLOW)은 비트 하나의 폭이 RS232통신(9600)의 1/10배나 될 정도로 빠른 통신입니다. 이러한 이유로 한 개의 명령 프레임 보낸 후 다음 프레임 보내기전에 처리가 완료할 정도의 딜레이를 요구합니다. 최대긴 명령의 연산시간은 250uS정도 이니 이것에 유념해주시기 바랍니다. 혹 이것을 오해하셔서 I2C_Write명령 또는 I2C_READ명령 사이사이마다 딜레이를 주는 경우가 있는데 이것이 아니라 뒤에 소개되는 명령과 명령 사이의 딜레이를 뜻하는 것으로 I2C_STOP과 다음I2C_START사이입니다.

RS232,485채널 사용편

4. 232,485채널 사용시 DIP_SW, Jumper 설명 및 ID

BFD-Con 뒷면에는 6개의 기능 조절용 DIP스위치가 존재하며, 각 SW별 기능은 아래 도표를 참조 바랍니다.



	On	Off	설명
Sw1	SGN-Set	OCS-Set	명령어 Set 선택
Sw2	4	0	ID설정용 Target ID=0xE0+ 설정된 SW값의 합
Sw3	2	0	
Sw4	1	0	

※ID설정에 대해 부가 설명을 하자면 Dip SW 2~4은 2진법을 사용하여 설정합니다.

I2C모드의 경우 어드레스 설정은 매뉴얼 뒷부분 I2C편을 참조 하세요

JP1 : RS232, 485선택용 점퍼
1-2 : RS232 선택
2-3 : RS485 선택
▲ 표시된 핀이 1번입니다.

5. 명령어 셋(Command Set)

BFD-Con은 원칩솔루션스 Command Set(이하OCS-Set)을 사용하며, 널리 알려진 컴파일 테크놀러지사의 SGN-Set 호환 모드를 외부의 DIP스위치를 통하여 변경할 수 있습니다.

먼저 OCS-Set에 대해 설명을 하고 뒷부분에서 SGN호환 Set에 대해 간단히 설명하겠습니다.

1)OCS Command Set 및 장치 ID

OCS-Set은 기존 Command Set의 문제점인 도트명령중 개별 On기능은 있으나 개별 Off기능의 단점과 데이터 전송의 신뢰성 개선을 위해 만들어졌습니다. n진법변환 표기(5byte) 명령어만 제외하면 4byte 포맷(종료문자 0x00포함)으로 만들어졌기에 안정적인 전송에 유리합니다.

※Check Sum 기능에 대하여

Check Sum기능은 종료코드 자리에 명령과 데이터 바이트를 XOR한 값을 내면 됩니다. 사용하지 않으시려면 체크섬 대신 종료코드 0x00을 보내시면 됩니다. BFD-Con는 Host로부터 온 명령을 분석하고 Check Sum에 오류가 있을 경우 해당 명령을 무시해버리고 기존 문자를 표기를 유지합니다. Feed Back (Host로의 송신) 은 BFD-Con에서

지원하지 않습니다(3선 방식(VCC, GND, SIG) 과의 호환성).

이 Check Sum기능의 장점은 외부 노이즈에 대처할 수 있다는 것과 Host측에서 바라볼 때 RS-232프로토콜을 소프트웨어로 구성해서 프로그램을 작성한다면 Host 마이컴의 경우 인터럽트 사용에 규제를 받게 됩니다. 비동기 통신인 RS-232데이터를 보내는 도중에 인터럽트를 처리하다보면 해당 데이터는 깨지게 됩니다. 이때 Check Sum이 없는 디스플레이의 경우 엉뚱한 문자나 공백 문자를 표기하는 경우가 대부분입니다. 이 경우 사람의 눈에도 이런 현상이 자주 보이게 되며 유저로부터 제품의 신뢰성을 떨어트리는 계기를 마련하게 됩니다. 허나 Check Sum기능을 활용하다보면 깨진 데이터는 무시해 버리고 기존문자를 유지하기에 Host 마이컴 프로그램 시 훨씬 더 자유롭게 프로그램이 가능하며 무시된 데이터는 Host 측에서 주기적으로 디스플레이를 갱신시켜줌으로서 안정적인 표기를 할 수 있는 것입니다.

기본적인 4byte명령의 경우 명령 포맷은 ID+ 명령+ 데이터 + CheckSum입니다.

여기서 CheckSum은 다음과 같이 산출됩니다.

$$\text{CheckSum} = \text{명령 XOR 데이터}$$

■ 각Digit별 위치



■ 기본 표시 명령어

기본 표시 명령은 SGN에서 각 자리를 ASCII코드에 의해 개별 제어라 기본이었던 것에 반에 각 자리의 세그먼트 제어 명령이 기본 표시 명령어입니다. 이는 I2C이용한 DLDC-7S/I5와의 일관성과 컴파일사의 SGN과의 호환성을 고려한 것입니다. 다시 한 번 강조 하자면 OCS-Set의 기본명령은 사용자정의 문자 표현이 가능한 세그먼트제어 명령입니다.

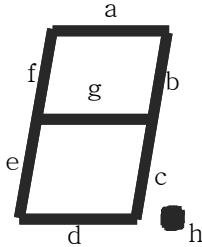
명령 순서는 ID + 위치 + Data + 종료코드(0x00)입니다. Check Sum기능을 사용할 경우 종료코드 대신 Check Sum을 전송하시면 됩니다.

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.

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 내보내는 명령으로서 각각의 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

뒤집힌 ASCII 표시 명령

각 자리의 데이터를 ASCII코드로 내보내는 명령으로서 위 ASCII표시 명령과 다른점은 문자가 위아래가 뒤집혀서 출력이 된다는 것입니다. FND를 뒤집어 시간과 분을 표기할 때 유용합니다.

ID	위치	ASCII	종료 Code / Check Sm	설명
Target ID	0xB1	code	0x00	위치1의 자리에 ASCII값을 표시
Target ID	0xB2	code	0x00	위치2의 자리에 ASCII값을 표시
Target ID	0xB3	code	0x00	위치3의 자리에 ASCII값을 표시
Target ID	0xB4	code	0x00	위치4의 자리에 ASCII값을 표시
Target ID	0xB5	code	0x00	위치5의 자리에 ASCII값을 표시

Digit 한계를 결정 명령(모델 지정명령)

이것은 FND가 고정적인 CSG시리즈나 SGN시리즈를 위한 것으로 FND를 사용자의 취향에 맞게 자리를 선정하여 장착할 수 있는 BFD-Con에는 사용자의 혼돈을 초래할 수 있어 공식적으로 지원하지 않습니다.

n진법 변환 표시 명령

OCS-Set으로서는 유일하게 5byte 포맷이며 Check Sum 기능이 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. BFD-Con는 최대 5자리 표시능력을 지니고 있습니다. 따라서 2byte 데이터를 변환해서 표시하려면 5자리 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.(5자리를 모두 사용하였을 때를 기준으로 설명하였습니다)

ID	명령	위치	Data High Byte	Data Low Byte	설명
Target ID	0xCA	0x01	데이터 상위 바이트	데이터 하위 바이트	지정된 위치에 4자리 Hex값으로 표기
Target ID	0xCB	0x01	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	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	0xDF	1 / 0	0x00	전체 Dot On/Off

◇ Dot Flash On/Off 명령

전체 또는 특정 위치의 Dot를 Flash 시키거나 해제하는 하는 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	0xEF	1 / 0	0x00	전체 Flash On/Off



Refresh Rate

FND 갱신 주기를 변경하는 명령입니다. Default로 200Hz로 설정되어 있습니다. 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)

사용 예제

Check Sum기능Off상태와 On상태를 각각 예로 들어 설명하겠습니다.

◇Check Sum기능을 사용하지 않을 경우(Default)

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 위치1번부터 표시 예, 명령 &HCB

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

INTEGER형 변수 I의 상위 바이트와 하위 바이트 값을 Hex 4자리로 위치2번부터 표시 예, 명령 &HCA

```
DIM I AS INTEGER
10 I = I + 1
SEROUT 8,30,0,0,[&HE0,2,&HCA,1,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 원칙솔루션스 홈페이지 자료실의 기술자료 게시판에 어플리케이션 노트로 SGN라이브러리를 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 위치2번부터 표시 예, 명령 &HCA

```
Dim i as integer
do
i=i+1
Putstr 1,&he0,&hca,2,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",0xe0,0xa1,'0',0); //위치 1에 "0"표시
printf("%c%c%c%c",0xe0,0xa2,0x30,0); //위치 2에 "0"표시
printf("%c%c%c%c",0xe0,0xa3,'1',0); //위치 3에 "1"표시
printf("%c%c%c%c",0xe0,0xa4,'2',0); //위치 4에 "2"표시
printf("%c%c%c%c",0xe0,0xa5,'3',0); //위치 5에 "3"표시
```



INTEGER형 변수 I의 상위 바이트와 하위 바이트 값을 BCD 5자리로 위치1번부터 표시 예, 명령 &HCB

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

INTEGER형 변수 I의 상위 바이트와 하위 바이트 값을 Hex 4자리로 위치2번부터 표시 예, 명령 &HCA

```
long i;
do{
    i++;
    printf("%c%c%c%c",0xe0,0xca,2,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 사용할 경우

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 SGN라이브러리를 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",0xe0,0xa1,'0',0xa1 ^ '0');
//위치 1에 "0"표시
printf("%c%c%c%c",0xe0,0xa2,0x30,0xa2 ^ 0x30);
//위치 2에 "0"표시
printf("%c%c%c%c",0xe0,0xa3,'1',0xa3 ^ '1');
//위치 3에 "1"표시
printf("%c%c%c%c",0xe0,0xa4,'2',0xa4 ^ '2');
//위치 4에 "2"표시
printf("%c%c%c%c",0xe0,0xa5,'3',0xa5 ^ '3');
//위치 5에 "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
```


1)SGN 호환 Command Set

간단하게 BFD-Con에서 지원하는 SGN-Set호환 명령어를 소개합니다. SGN호환 명령어는 컴파일사의 TinyPLC에서 지원하는 명령어 셋으로 TinyPLC를 사용할 경우 DIP-SW1을 On하여 사용하기 바랍니다.

아래에는 명시하지 않았지만 TinyPLC의 Dot표기 기능도 지원합니다.

■ 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 SGN라이브러리를 제공합니다.

3) PIC Micom의 경우

각 자리별로 ASCII코드를 통해 제어하는 예

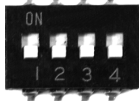
```
printf("%c%c%c",0xe0,1,'0'); //위치1에 "0"표시
printf("%c%c%c",0xe0,2,'1'); //위치2에 "1"표시
```

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

I²C채널 사용편

6. DIP_SW 설명 및 ID

I²C채널 사용시 3개의 기능 조절용 DIP스위치를 사용하여 주소를 설정할 수 있으며 아래 도표를 참조 바랍니다.



	On	Off	설명
Sw1			I2C채널에선 이용하지 않음
Sw2	8	0	ID설정용 Target ID=0x70+ 설정된 SW값의 합
Sw3	4	0	
Sw4	2	0	

※ID설정에 대해 부가 설명을 하자면 Dip SW 2~4은 2진법을 사용하여 설정합니다.

장치 ID

위에서 짧게 설명하였듯이 하나의ID를 부여받으며 7Bit어드레스 I²C를 사용하기 때문에 상위 니블 Base어드레스 4bit와 LSB 1bit를 제외한 3bit(ID)를 사용하여 동종 Device간 주소를 구별합니다. 이러한 통신방식의 특성으로 인하여 채널당 최대 8개의 Device사용할 수 있는 것입니다.

빠른 이해를 위해 다음 표를 참조바랍니다. 베이스 어드레스는 0x70(고정)입니다.

7 Bit Address 형식+R/_W비트						
7 (MSB)	6	5	4	3	2	1 (LSB)
Base 주소			Sw1	Sw2	Sw3	R/_W
0	1	1	1	ID설정 용		

Sw1	Sw2	Sw3	R/_W	Base Address : 0x70
OFF	OFF	OFF	1/0	0x70 + 0x00
OFF	OFF	ON	1/0	0x70 + 0x02
OFF	ON	OFF	1/0	0x70 + 0x04
OFF	ON	ON	1/0	0x70 + 0x06
ON	OFF	OFF	1/0	0x70 + 0x08
ON	OFF	ON	1/0	0x70 + 0x0A
ON	ON	OFF	1/0	0x70 + 0x0C
ON	ON	ON	1/0	0x70 + 0x0E

기본 주소 0x70에 위 DIP-SW를 2진수 SW1(가중치8), Sw2(가중치4), Sw3(가중치2)로 생각해서 ON은 1,OFF는 0으로 환산해서 더하시면 그것이 바로 모듈의 실제 주소가 됩니다.

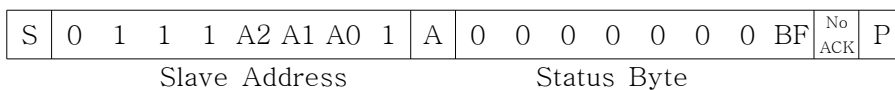
예를 들어 0x72의 Write모드 주소 설정 법은 Sw1(OFF), SW2(OFF), SW3(ON)로 설정하시면 됩니다.

7. 명령어

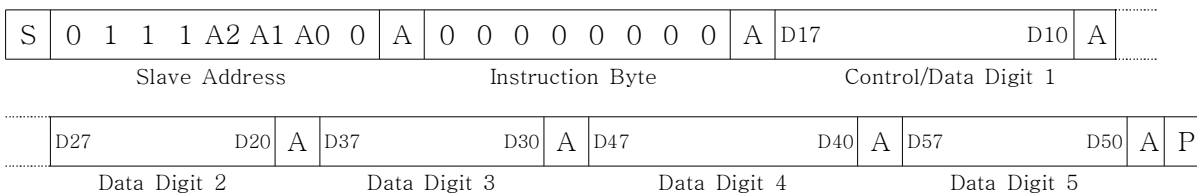
I²C통신의 대부분은 전송마다 Ack를 주고/받고 클럭 라인을 통해 동기를 맞추기에 꼭 Check Sum을 사용하지 않고서도 상당한 신뢰성을 보이는 통신 중에 하나입니다.

I²C 7Bit주소 체계에서는 8Bit의 LSB의 경우 R/_W비트로서 1의 경우 Read, 0의 경우 Write모드를 의미합니다.

하단의 DLDC-7S/I10 명령어 구조도 참조 바랍니다.



a. READ Mode



b. Write Mode

DLDC-7S/I10 명령어 구조도

*Read 모드의 경우

READ Address+Status Byte : 2byte 형식

Status Byte는 LSB 1bit만 유효하면 1bit는 Busy Flag(BF)로서 명령이 처리되는 동안 1을 유지합니다. 허나 BF플래그를 체크 할 때 제품에 통신 부하를 주기 때문에, 이것을 간단히 Delay 명령으로 대체할 것을 권장합니다. 처리시간이 가장 많이 걸리는 명령일 경우 250us정도 소요되는데, 명령과 명령사이에 약 250us의 Delay를 주면 구지 BF플래그를 체크하지 않아도 되는 이점이 있습니다.

*Write 모드의 경우 명령의 특성이라면 레지스터 포인터가 자동증가 한다는 것입니다. 따라서 처음 문자를 쓰기 시작할 위치를 정하고 데이터를 연속으로 보내게 되면 위치(레지스터 포인터)가 자동증가 되면서 한 번에 몰아쓰기가 가능합니다. 그러기에 명령 형식이 가변길이라 할 수 있습니다. 예를 들어 위치1부터 10까지 표기하려면 아래와 같다

Write Address + 명령 + 위치1지정 + 위치1데이터 + 위치2데이터 + 위치3데이터 + 위치4데이터 + 위치5데이터
 위치는 1부터 5까만 자동 증가합니다.

■ 각Digit별 위치



■ 기본 표시 명령어

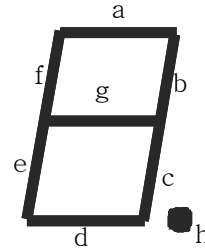
기본 표시 명령어는 SGN(컴파일러)에서 각 자리를 ASCII코드에 의해 개별 제어가 기본이었던 것에 반에 각 자리의 세그먼트 제어 명령어 기본 표시 명령어입니다. 컴파일사의 CSGN과의 호환성을 고려한 것입니다. 몰아 쓰기가 가능하여 어느 위치에서 시작하든지 쓰고 싶은 위치까지 문자를 연이어 쓰기가 가능합니다. 아래 도표에서는 특정위치에 특정 문자 하나를 기록하는 것입니다.

Target 주소	위치	데이터	설명
Write Mode 주소 0x01	1Byte조합	위치0의 세그먼트를 제어.	
Write Mode 주소 0x02	1Byte조합	위치1의 세그먼트를 제어.	
Write Mode 주소 0x03	1Byte조합	위치2의 세그먼트를 제어.	
Write Mode 주소 0x04	1Byte조합	위치3의 세그먼트를 제어.	
Write Mode 주소 0x05	1Byte조합	위치4의 세그먼트를 제어.	

몰아쓰기가 가능한 명령

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 데이터를 개별적으로 제어하는 기능입니다. 몰아쓰기가 가능한 명령입니다.

Target 주소	위치	ASCII	설명
Write Mode 주소 0xA1	code	위치1의 자리에 ASCII값을 표시	
Write Mode 주소 0xA2	code	위치2의 자리에 ASCII값을 표시	
Write Mode 주소 0xA3	code	위치3의 자리에 ASCII값을 표시	
Write Mode 주소 0xA4	code	위치4의 자리에 ASCII값을 표시	
Write Mode 주소 0xA5	code	위치5의 자리에 ASCII값을 표시	

몰아쓰기가 가능한 명령

■ 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. 2byte데이터를 변환해서 표시하려면 5자리 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. 이 명령은 성격상 몰아쓰기가 가능하지 않습니다.

Target 주소	명령	위치	Data High Byte	Data Low Byte	설명
Write Mode 주소 0xCA	첫 자리 표시 위치	데이터 상위 바이트	데이터 하위 바이트	지정한 위치에 4자리 Hex값으로 표기	
Write Mode 주소 0xCB	첫 자리 표시 위치	데이터 상위 바이트	데이터 하위 바이트	지정한 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. Flash 명령은 신뢰성을 위하여 몰아쓰기를 지원하지 않습니다.



Target 주소	위치	On/Off	설명
Write Mode 주소	0xF1	1 / 0	위치1의 자리 Flash On/Off
Write Mode 주소	0xF2	1 / 0	위치2의 자리 Flash On/Off
Write Mode 주소	0xF3	1 / 0	위치3의 자리 Flash On/Off
Write Mode 주소	0xF4	1 / 0	위치4의 자리 Flash On/Off
Write Mode 주소	0xF5	1 / 0	위치5의 자리 Flash On/Off
Write Mode 주소	0xFF	1 / 0	전체 Flash On/Off

명령 전송 포맷(3Byte 길이)

■ Dot 제어 명령

◇ Dot ON/OFF명령(몰아쓰기를 지원안함)
전체 또는 특정 위치의 Dot를 On/Off하는 명령입니다.

Target 주소	위치	On/Off	설명
Write Mode 주소	0xD1	1 / 0	위치1의 자리 Dot On/Off
Write Mode 주소	0xD2	1 / 0	위치2의 자리 Dot On/Off
Write Mode 주소	0xD3	1 / 0	위치3의 자리 Dot On/Off
Write Mode 주소	0xD4	1 / 0	위치4의 자리 Dot On/Off
Write Mode 주소	0xD5	1 / 0	위치5의 자리 Dot On/Off
Write Mode 주소	0xDF	1 / 0	전체 Dot On/Off

명령 전송 포맷(3Byte 길이)

◇ Dot Flash On/Off 명령(몰아쓰기를 지원안함)
전체 또는 특정 위치의 Dot를 Flash 시키거나 해제하는
하는 명령으로서 Flash명령 전에 Dot가 켜져 있어야
Flash가 가능합니다.

Target 주소	위치	On/Off	설명
Write Mode 주소	0xE1	1 / 0	위치1의 자리 Flash On/Off
Write Mode 주소	0xE2	1 / 0	위치2의 자리 Flash On/Off
Write Mode 주소	0xE3	1 / 0	위치3의 자리 Flash On/Off
Write Mode 주소	0xE4	1 / 0	위치4의 자리 Flash On/Off
Write Mode 주소	0xE5	1 / 0	위치5의 자리 Flash On/Off
Write Mode 주소	0xEF	1 / 0	전체 Flash On/Off

명령 전송 포맷(3Byte 길이)

■ 특수 명령

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

Target 주소	명령	반전	설명
Write Mode 주소	0xCC	0x33	전체 지우기 명령(Clear) -캐릭터만 지워짐, 기능명령 유효
Write Mode 주소	0xCE	0x31	리셋 명령(rEset)
Write Mode 주소	0xCF	0x30	펌웨어(Firmware) 버전을 6,7,8,9,10자리에 표기

명령 전송 포맷(3Byte 길이)

■ 사용 예제

CuBLOC에서 사용과 PIC마이컴에서 사용 예제 크게 두 가
지로 나누고, CuBLOC의 경우 I2C함수를 이용하는것과
CSG내장 라이브러리를 이용하는 법으로 나눠 설명 하겠
습니다.

1) CuBLOC의 경우

◇CuBLOC을 I2C 함수를 사용해서 제어하기

*CuBLOC의 경우 원칩솔루션스 홈페이지 자료실의 기술
자료 게시판에 어플리케이션 노트로 구동라이브러리를 제
공합니다.
CuBLOC에서는 Hex데이터 표기를 0xaa와 같은 형식과
&haa와 같은 형식 둘다 지원합니다. 착오 없으시길 바랍
니다. 에러 검출에 대한 예제는 원칩솔루션스 홈페이지의
어플리케이션 노트를 참고 바랍니다.

위치3에 문자'-'를 표기하는 예

```
Set I2c 9,8
```

```
Dim err As Byte
I2cstart
err=I2cwrite(&h70) '주소
err=I2cwrite(&h03) '위치 3
err=I2cwrite(0b01000110) '사용자 정의 문자'-'
I2cstop
```

위치2에 'A'를 표기하는 예

```
Set I2c 9,8
Dim err As Byte
I2cstart
err=I2cwrite(&h70) '주소
err=I2cwrite(&ha2) '아스키 코드 명령
err=I2cwrite(asc("A")) 'A의 아스키 코드값을 전송
I2cstop
```

위치2부터 위치 5까지 'ABC'를 표기하는 예

```
Set I2c 9,8
Dim err As Byte
I2cstart
err=I2cwrite(&h70) '주소
err=I2cwrite(&ha2) '아스키 코드 명령
err=I2cwrite(asc("A")) 'A의 아스키 코드값을 전송
err=I2cwrite(asc("B")) 'B의 아스키 코드값을 전송
err=I2cwrite(asc("C")) 'C의 아스키 코드값을 전송
I2cstop
```



INTEGER형 변수 I의 상위 바이트와 하위 바이트 값을 BCD 5자리로 위치1번부터 표시 예, 명령 0xCB

```

Set I2c 9,8
  Dim i as integer
  Dim err As Byte
do
  i=i+1
  I2cstart
  err=I2cwrite(&h70) '주소
  err=I2cwrite(&hcb) 'bcd변환표시 명령
  err=I2cwrite(1) '위치 1
  err=I2cwrite(i.byte1) '상위 바이트
  err=I2cwrite(i.byte0) '하위바이트
  I2cstop
  delay 100
loop

```

INTEGER형 변수 I의 상위 바이트와 하위 바이트 값을 Hex 4자리로 위치2번부터 표시 예, 명령 0xCA

```

Set I2c 9,8
  Dim i as integer
  Dim err As Byte
do
  i=i+1
  I2cstart
  err=I2cwrite(&h70) '주소
  err=I2cwrite(&hca) 'hex변환표시 명령
  err=I2cwrite(2) '위치 2
  err=I2cwrite(i.byte1) '상위 바이트
  err=I2cwrite(i.byte0) '하위바이트
  I2cstop
  delay 100
loop

```

Dot On / Off ,Flashing

```

Set I2c 9,8
  Dim err As Byte
I2cstart
err=I2cwrite(&h70) '주소
err=I2cwrite(&hd2) '위치 2
err=I2cwrite(0) '위치2 Dot Off
I2cstop
delay 1 '필요에 따라

```

```

I2cstart
err=I2cwrite(&h70) '주소
err=I2cwrite(&hd3) '위치 3
err=I2cwrite(1) 'Dot 3 On
I2cstop
delay 1 '필요에 따라

```

```

I2cstart
err=I2cwrite(&h70) '주소
err=I2cwrite(&he3) '위치 3
err=I2cwrite(1) '위치 3 Dot flash
I2cstop

```

INTEGER형 변수 I를 n진법 변환명령 없이 출력하기

```

Set I2c 9,8
  Dim i as integer
  Dim err As Byte
  Dim ST As String *3
do
  i=i+1
  ST = Dp(i,3,0) 'i를 3자리 문자열로 변환하여 ST에
  I2cstart
  err=I2cwrite(&h70)
  err=I2cwrite(&ha1) '1번자리부터
  err=I2cwrite(ST_a(0))'문자열 ST의 0번 문자
  err=I2cwrite(ST_a(1))'문자열 ST의 1번 문자
  err=I2cwrite(ST_a(2))'문자열 ST의 2번 문자
  I2cstop
  delay 100
loop

```

데이터 Flashing On/Off

```

Set I2c 9,8
  Dim err As Byte
I2cstart
err=I2cwrite(&h70) '주소
err=I2cwrite(&hf3) '위치 3
err=I2cwrite(1) '위치 3 flash On
I2cstop

```

```

delay 1 '필요에 따라

I2cstart
err=I2cwrite(&h70) '주소
err=I2cwrite(&hf4) '위치 4
err=I2cwrite(0) '위치 4 flash Off
I2cstop

```

Busy Flag(BF) 체크하는 예

```

Set I2c 9,8
  Dim err As Byte
  Dim status As Byte
I2cstart
err=I2cwrite(&h71) '//0x71 읽기 모드 주소=쓰기주소+1
status=I2cread() //0번 비트가 BF이다.
I2cstop
IF status.bit0 Then ..... //0번 비트가 BF
//BF가 셋되어 있으면?? ..

```

※BF체크보다는 적당한 Delay를 권장합니다.

◇CuBLOC을 CSG내장 라이브러리를 사용해서 제어하기

※본 제품은 컴파일사의 CSG메인 칩셋인 필립스SAA1064와 하위 호환성을 갖게 설계되어 위치 체계가 같습니다. 따라서 CSG-4S(컴파일사)와 같이 BFD-CON 또한 CuBLOC의 CSG내장 라이브러리를 이용할 때는 반듯이 숙지해야 할 내용이 있습니다. 바로 위치에 관한 것인데, CSG 라이브러리는 내부적으로 위치정보에 +1(더하기1)을 해서 출력을 한다는 것입니다. 따라서 CSG 내장 라이브러리를 사용하여 생각했던 위치에 출력을 하려면 위치 데이터에 -1하여야 합니다. 이는 컴파일 사의 CSG-4S가 SAA1064칩을 사용함으로써 CSG라이브러리를 사용시 유저의 실수로 위치 0에 데이터를 쓸 경우 컨트롤 비트를 건드리는 오류 범하게 되는데 이를 방지하고자 라이브러리 내부적으로 +1을 시키는 방식을 택한 것으로 사료됩니다. 하나 이로 인하여 CSG를 내장 라이브러리로 제어할 때와 I2C함수로 제어 할 때의 위치 값이 하나씩 차이나게 되는 문제점이 있습니다. 결론적으로 CuBLOC에서 CSG라이브러리를 사용해서 BFD_CON을 사용할 때는 위치 데이터에서 하나씩 감소시켜 사용하라고 알고 계시면 됩니다. 또한 CSG라이브러리에서는 주소가 아래와 같이 매칭 됩니다.

BFD-CON주소		CSG주소
0x70	→	0
0x72	→	1
0x74	→	2
0x76	→	3

CUBLOC CSG라이브러리에서 주소 매칭관계

Csghex를 사용하여 출력(BFD-CON 주소:0x70)

```
Set I2c 9,8
Csghex 0,&hffff '0x70주소 위치 1부터 FFFF를 표기
```

※csghex CuBLOC내장 라이브러리로서 항상 위치1번 자리 부터 표기를 합니다.

Csgdec를 사용하여 출력(BFD-CON 주소:0x70)

```
Set I2c 9,8
Csgbcd 0,&hffff '0x70주소 위치 1부터 65535중 5535를 표기
```

※csgdec CuBLOC내장 라이브러리로서 항상 위치1번 자리 부터 표기를 합니다.

Csgxput를 사용하여 출력(BFD-CON 주소:0x70)

```
Set I2c 9,8
Csgxput 0,0,0b01000110
'0x70주소 위치 1 에사용자 정의 문자'ㄱ'
```

'위 csgxput함수는 아래 와 같이 I2C함수로 등가 표현이 가능합니다.

```
Set I2c 9,8
Dim err As Byte
I2cstart
err=I2cwrite(&h70) '주소
err=I2cwrite(&h01) '위치 1
err=I2cwrite(0b01000110) '사용자 정의 문자'ㄱ'
```

Csgnput을 사용하여 위치1~9까지 0~9를 차례대로 표기하는 예 (BFD-CON 주소:0x70)

```
Set I2c 9,8
Csgnput 0,0,&h30 '위치 1(1-1)에 아스키코드0x30("0")표기
Csgnput 0,1,&h31 '위치 2(2-1)에 아스키코드0x31("1")표기
Csgnput 0,2,&h32 '위치 3(2-1)에 아스키코드0x32("2")표기
Csgnput 0,3,&h33 '위치 4(2-1)에 아스키코드0x33("3")표기
Csgnput 0,4,&h34 '위치 5(2-1)에 아스키코드0x34("4")표기
```

'위 csgnput 함수는 I2C함수로 아래와 같이 표현이 가능한데 아래와 같이 I2C함수로 사용할 경우 CuBLOC에서 시스템 부하가 줄어드는 장점이 있습니다.

```
Set I2c 9,8
Dim err As Byte
I2cstart
err=I2cwrite(&h70) '주소
err=I2cwrite(&ha1) '위치 1부터 아스키 코드 명령
err=I2cwrite(&h30) '0의 아스키 코드값을 전송(위치 1부터)
err=I2cwrite(&h31) '1의 아스키 코드값을 전송
err=I2cwrite(&h32) '2의 아스키 코드값을 전송
err=I2cwrite(&h33) '3의 아스키 코드값을 전송
err=I2cwrite(&h34) '4의 아스키 코드값을 전송(위치 5까지)
I2cstop
```

2) PIC Micom의 경우(CCS-C)

위치3에 문자'ㄱ'를 표기하는 예

```
i2c_start();
i2c_write(0x70); //주소
i2c_write(0x03); //위치 3
i2c_write(0b01000110); //사용자 정의 문자'ㄱ'
i2c_stop();
```

위치2에 'A'를 표기하는 예

```
i2c_start();
i2c_write(0x70); //주소
i2c_write(0xa2); //아스키 코드 명령
i2c_write('A'); //'A'의 아스키 코드값을 전송
i2c_stop();
```

위치2부터 위치 5까지 'ABC'를 표기하는 예

```
i2c_start();
i2c_write(0x70); //주소
i2c_write(0xa2); //아스키 코드 명령
i2c_write('A'); //'A'의 아스키 코드값을 전송
i2c_write('B'); //'B'의 아스키 코드값을 전송
i2c_write('C'); //'C'의 아스키 코드값을 전송
i2c_stop();
```



INTEGER형 변수 i의 상위 바이트와 하위 바이트 값을 Hex 4자리로 위치2번부터 표시 예, 명령 0xCA

```

long i;
do{
    i++;
    i2c_start();
    i2c_write(0x70);
    i2c_write(0xca); //4자리 Hex변환 명령
    i2c_write(2); //표시 시작 위치
    i2c_write(i>>8); //상위 바이트
    i2c_write(i); //하위바이트
    i2c_stop();
    delay_us(250);
}

```

INTEGER형 변수 i의 상위 바이트와 하위 바이트 값을 Bcd 5자리로 위치1번부터 표시 예, 명령 0xCB

```

long i;
do{
    i++;
    i2c_start();
    i2c_write(0x70);
    i2c_write(0xcb); //5자리 BCD변환 명령
    i2c_write(1); //표시 시작 위치
    i2c_write(i>>8); //상위 바이트
    i2c_write(i); //하위바이트
    i2c_stop();
    delay_us(250);
}

```

데이터 Flashing On/Off

```

i2c_start();
i2c_write(0x70);
i2c_write(0xf2); //위치 2 문자 Flash
i2c_write(1); //Flash ON
i2c_stop();
delay_us(100); //필요에 따라

i2c_start();
i2c_write(0x70);
i2c_write(0xf3); //위치 3 문자 Flash
i2c_write(0); //Flash OFF
i2c_stop();

```

Dot On / Off ,Flashing

```

i2c_start();
i2c_write(0x70);
i2c_write(0xd2); //위치 2
i2c_write(0); //Dot OFF
i2c_stop();
delay_us(100); //필요에 따라

```

```

i2c_start();
i2c_write(0x70);
i2c_write(0xd3); //위치 3
i2c_write(1); //Dot ON
i2c_stop();
delay_us(100); //필요에 따라

```

```

i2c_start();
i2c_write(0x70);
i2c_write(0xe3); //위치 3
i2c_write(1); //Dot Flash
i2c_stop();

```

Busy Flag(BF) 체크하는 예

```

int status;

i2c_start();
i2c_write(0x71); //읽기 모드 주소
status=i2c_read(0); //ACK를 보내지 않고 읽는다.
i2c_stop();
if (bit_test(status,0)) delay_us(250); //0번비트가 BF
//BF가 셋되어 있으면 ..

```

※BF체크보다는 적당한 Delay를 권장합니다.

BFD-Con v2.0 매뉴얼
Last Modified Date : 2007.12.11