

## Semi FND DP485



### 1.개요

Semi FND DP485(OCS-045) 모델은 RS485 시리얼 통신을 통해 4 자리 FND 모듈에 숫자 및 문자 표시를 할 뿐 아니라, 1 개의 릴레이 출력(NO/NC), 및 스위치 입력(버튼 3 개, 센서 1 개)을 호스트로 전송하는 기능으로 간단한 리모트 I/O 기능을 수행할 수 있는 제품입니다. Display 는 각 자리에 Ascii 코드를 전송함으로써 해당 문자를 쉽게 표시할 수 있는 장점이 있고, Blink, Dot 표기, Hex 표기 등 다양한 기능을 내장하고 있습니다.

또한 통신 프로토콜은 LS 산전의 SV-iG5 인버터에 호환되도록 설계되어 있어 Master-k 120, 200 PLC 의 485 Master 모드에 직접 붙여 편리하게 사용할 수 있습니다.

### 2.일반사항 및 지원기능

**전원** : 220V(AC)

**Baudrate** : (9600/ 19200/ 38400)변경 가능

(8, N, 1)

**ID설정** : 255개의 ID 부여 가능

**통신** : RS485 통신

**디스플레이** : Ascii, HEX 숫자, 영문자, 특수기호, DOT 표기 사능

특정 위치의 FND FLASHING 가능

특정 위치의 DOT 표시 가능

특정 위치의 DOT FLASHING 가능

입력 : 4개 (버튼 입력3, NPN 근접센서 입력 1)

그외 : 호스트의 쓰기명령 응답(ACK)은 선택 가능

출력 : 릴레이 1개 (NO/NC릴레이)

프로토콜 : LS 산전 Master-k 485 Master 호환 프  
로토콜(인버터 제어용)

사이즈 : 81 mm(가로) x 74 mm(세로) x 33 mm(높이)

### 3.외형 및 치수





#### 4. DP485 설정 방법

\*SET 버튼을 누르고 전원 투입 시에 아래의 설정 모드로 전환되어집니다.



값 저장 및 다음 화면으로 이동

\*누른 상태에서 전원투입시  
설정 모드로 전환



##### 1. ID 설정 화면

0~ 255의  
ID 부여 가능



##### 2. Baud rate 설정 화면

1 - 9600  
2 - 19200  
3 - 38400



##### 3. 쓰기 ACK 설정 화면

O - 쓰기 활성화  
F - 쓰기 비활성

\*마지막으로 한번 더 누르면 설정  
완료 및 메인 화면으로 돌아옵니다.

## 5. 쓰기 명령 형식

쓰기 명령 형식 (국번4번으로 가정)

ENQ	국번	명령	주소				갯수	데이터				SUM		EOT
0	4	W	0	0	0	1	1	0	0	F	5	8	8	EOT
	국번4번	쓰기	주소 1 번지에					데이터값 Hexa 00F5				*note1		

DP 485 --> PLC	ACK	국번	명령	데이터값				SUM		EOT
	ACK	0	4	W	0	0	X	X	X	X
		국번4번	읽기	데이터값 000X				SUM XX		

명령 형식		국번	명령	주소				갯수	데이터				SUM			
릴레이 출력	ENQ	0	4	W	0	0	0	1	1	0	0	0	X	X	X	EOT
디스플레이 형식	ENQ	0	4	W	0	0	0	2	1	B	X	X	X	X	X	EOT
데이터1 HEX,DEC,ASCII-1	ENQ	0	4	W	0	0	0	3	1	X	X	X	X	X	X	EOT
데이터2 ASCII-2	ENQ	0	4	W	0	0	0	4	1	X	X	X	X	X	X	EOT
데이터3 ASCII-3	ENQ	0	4	W	0	0	0	5	1	X	X	X	X	X	X	EOT
데이터4 ASCII-4	ENQ	0	4	W	0	0	0	6	1	X	X	X	X	X	X	EOT

\*note  
SUM 값은 제어신호를 제외한 데이터를 모두 더한값의 하위 바이트를 ASCII 코드로 보냅니다

DP485 기본 문자코드 테이블(ASCII 코드와 호환)

		상위 nibble						
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	r	6	F		F			
7	L	7	g		g			
8	r	8	H		h			
9	J	9	I	y	I	y		
A	U	.	J		j			
B	n	—	[					
C	┌		L	l	l			
D	—		n	]	n			
E	ㄷ		n		n			
F	□		O	_	o			

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.

## 릴레이 출력 예시 및 SUM 설명

사용 예시 (국번[ID] 4 번 가정)

데이터 4byte 중 **마지막 자리**의 byte 가 1 이면 ON, 0 이면 OFF 입니다.

### 릴레이 ON

05(ENQ) 30 34 57 30 30 30 31 31 30 30 30 31 36 45 04(EQT) <릴레이 ON>

### 릴레이 OFF

05(ENQ) 30 34 57 30 30 30 31 31 30 30 30 30 36 44 04(EQT) <릴레이 OFF>

## SUM 이란 무엇인가?

30 34 는 ID 04 를 나타내며, 36 45 는 SUM 을 말합니다.

SUM 은 ENQ 와 EQT 를 제외한 전송하는 모든 HEX 값을 더한후

$$30 + 34 + 57 + 30 + 30 + 30 + 31 + 31 + 30 + 30 + 30 + 31 = 26E$$

이 더하여 나온 값 26E 의 하위 두 바이트 26 E 를 각각 아스키표의 값대로

36 과 45 로 동일하게 HEX 의 형태로 뒤에 붙여 보내주면 됩니다.

## 디스플레이(HEX)

사용 예시 (국번[ID] 4 번 가정)

디스플레이에 관한 설명은 다소 복잡해질 수 있기 때문에, 구조에 대한 설명과 순차적인 적용방법으로 이해를 쉽게 하도록 유도해보겠습니다.

명령 형식		국번	명령	주소	갯수	데이터	SUM
릴레이 출력	ENQ	0	4	W	0 0 0 1	1 0 0 0 X	X X EOT
디스플레이 형식	ENQ	0	4	W	0 0 0 2	1 B X X X X	X X EOT
데이터1 HEX,DEC,ASCII-1	ENQ	0	4	W	0 0 0 3	1 X X X X	X X EOT
데이터2 ASCII-2	ENQ	0	4	W	0 0 0 4	1 X X X X	X X EOT
데이터3 ASCII-3	ENQ	0	4	W	0 0 0 5	1 X X X X	X X EOT
데이터4 ASCII-4	ENQ	0	4	W	0 0 0 6	1 X X X X	X X EOT



데이터				설명
B	A	X	X	십육진수 표기 (0~FFFF)
B	B	X	X	십진수 표기 (0~9999)
B	C	X	X	문자 표기 (ASCII 코드)
B	D	X	X	음수 십진수 표기 (0~-999)

HEX  
DEC  
ASCII

(앞서 나온 이미지를 확대 및 설명한 이미지입니다)

우선 주목해야하는 것은 **디스플레이 형식**입니다.

디스플레이 형식은 일단 **HEX, DEC, ASCII** 3 종류가 있습니다.

이는 받은 데이터를 어떻게 표시하는지에 대한 형식을 말하며, Default 값은 **DEC(10 진수)**입니다.

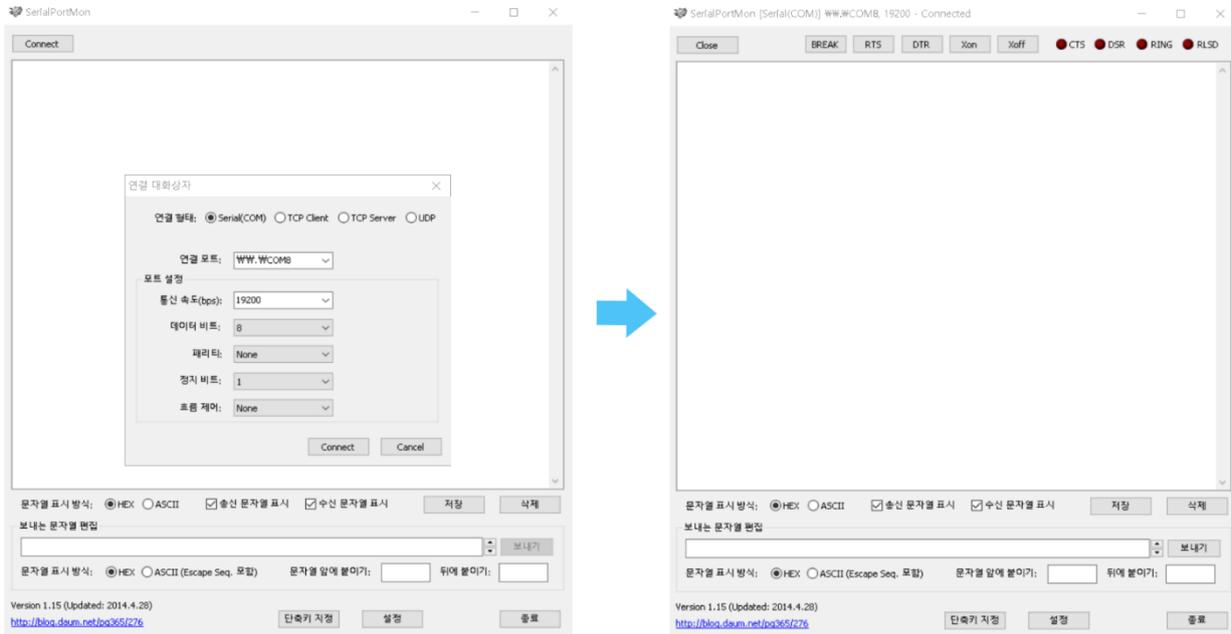
**HEX,DEC** 는 **데이터 1** 의 주소만 사용하며 (후에 나올 \*33)

**ASCII** 는 각 자리마다 **데이터 1,2,3,4** 총 4 개의 주소(후에 나올 \*33,34,35,36)를 사용합니다.

(\*32 는 디스플레이 형식을 변경할 때 씁니다.)

설명으로는 이해하는것에 어려움이 있을것으로 예상되어 예제를 통해 알아보겠습니다.

### 전원투입 및 통신 연결



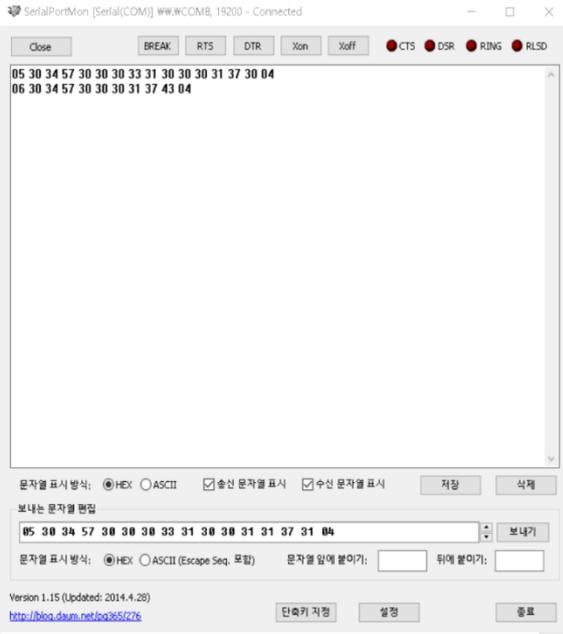
위는 전원을 넣었을 때, 기본 디스플레이 되어지는 이미지입니다.

### DEC 로 디스플레이 출력

디스플레이 형식을 건드리지 않았을 때 DISPLAY 되는 문자들은 Default 로 DEC 입니다. DEC 에서는 HEX 의 형태로 데이터를 기입시, 데이터의 값을 십진수로 DISPLAY 합니다. HEX 값으로 아래와 같이 데이터 3 주소에 0001 을 넣어주면



05(ENQ) 30 34 57 30 30 30 \*33 31 \*30 30 30 31\* 37 30 04(EQT) <디스플레이: 0001>  
 \*30 30 30 31\*은 DEC 에서 HEX 0001 을 의미, HEX 0001 은 DEC 로 1



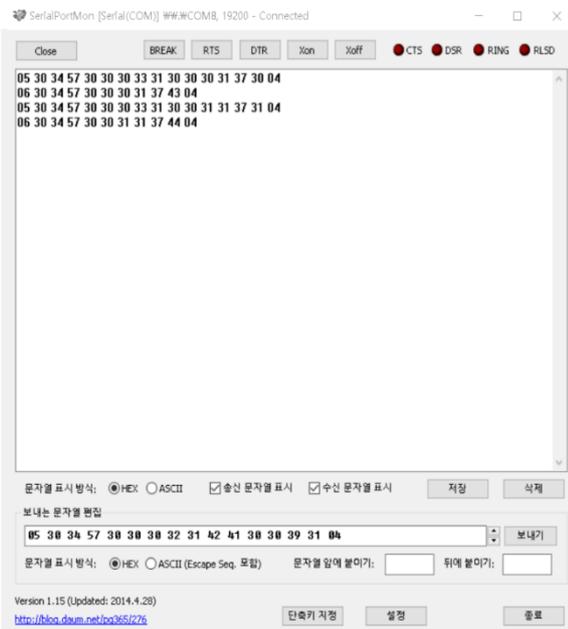
이번에는 HEX 값으로 아래와 같이 데이터 3 주소에 0011 을 넣어주면  
 05(ENQ) 30 34 57 30 30 30 \*33 31 30 30 31 31 37 31 04(EQT) <디스플레이: 0017>  
 \*30 30 31 31\*은 DEC 에서 HEX 0011 을 의미, HEX 0011 은 DEC 로 17  
 이처럼 HEX 의 데이터가 DEC 의 형태로 DISPLAY 되어지는것을 알 수 있습니다.

## HEX 로 디스플레이 출력

이번에는 디스플레이 형식을 DEC 에서 HEX 로 바꿔보겠습니다.

HEX 로 바꾸기 위해서 주소는 2 번 주소(32)로 바꾸고 데이터값은 BA□□ 을 넣어줘야 합니다. 다시 default 였던 DEC 로 돌아가려면 BB□□ 을 넣어주면 됩니다. (□□의 자리에는 무엇을 넣어주어도 상관없으나, SUM 값은 맞춰줘야합니다.)

기존의 DEC 로 디스플레이 했던 HEX \*30 30 31 31\*은 그대로 DP485 에 남아있습니다. 이 상태에서 디스플레이 형식만 변경해주면, DEC 로 디스플레이 해주던 그 데이터가 HEX 로 바뀌어 디스플레이 되어질 것 입니다.





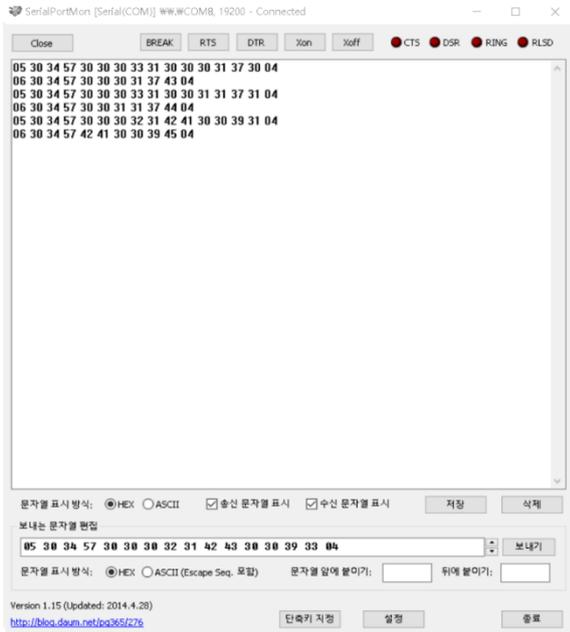
05(ENQ) 30 34 57 30 30 30 \*32 31 \*42 41\* 30 30 39 31 04(EQT) <DEC 에서 HEX 로>  
 \*42 41\* -> BA(디스플레이 형식 HEX 변경 명령), 디스플레이 형식을 바꿀때는 \*32 로

0011 = <디스플레이: 0011>

이렇게 디스플레이 형식만 DEC 에서 HEX 로 바꾸어주었는데, DISPLAY 되었던 문자가  
 0017 에서 0011 로 바뀌었습니다. 0017(DEC) -> 0011(HEX)

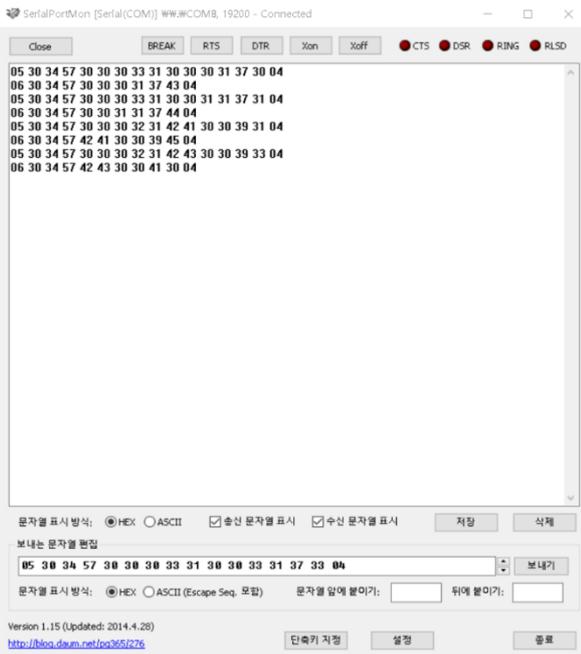
### ASCII 로 디스플레이 출력

### 아스키로 바꿔주는 명령 및 아스키로 디스플레이 출력



05(ENQ) 30 34 57 30 30 30 \*32 31 42 43 30 30 39 33 04(EQT) (ASCII 로변경)

\*42 43\* -> BC(디스플레이 형식 ASCII 변경 명령), 디스플레이 형식을 바꿀때는 \*32 로

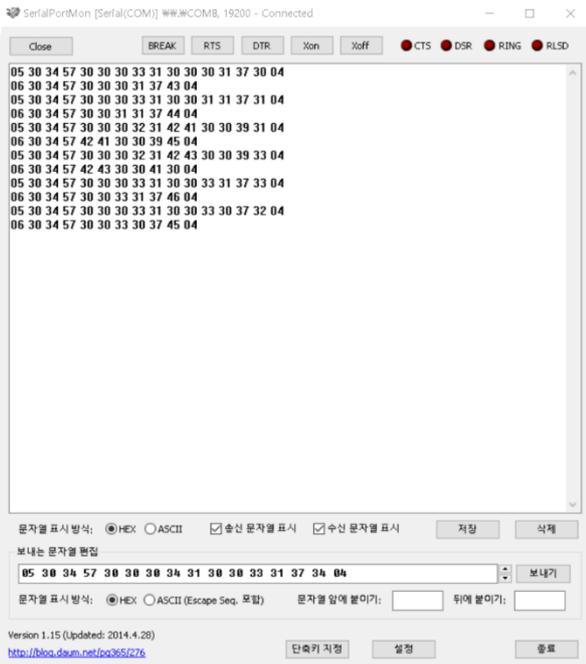


05(ENQ) 30 34 57 30 30 30 \*33 31 30 30 \*33 31\* 37 33 04(EQT) (1\_ \_ \_ 출력)  
 \*33 31\*은 ASCII 에서 '1'을 나타내는 0x31 을 의미, \*33 은 ASCII 1 번째 자리



05(ENQ) 30 34 57 30 30 30 \*33 31 30 30 \*33 30\* 37 32 04(EQT) (0\_ \_ \_ 출력)

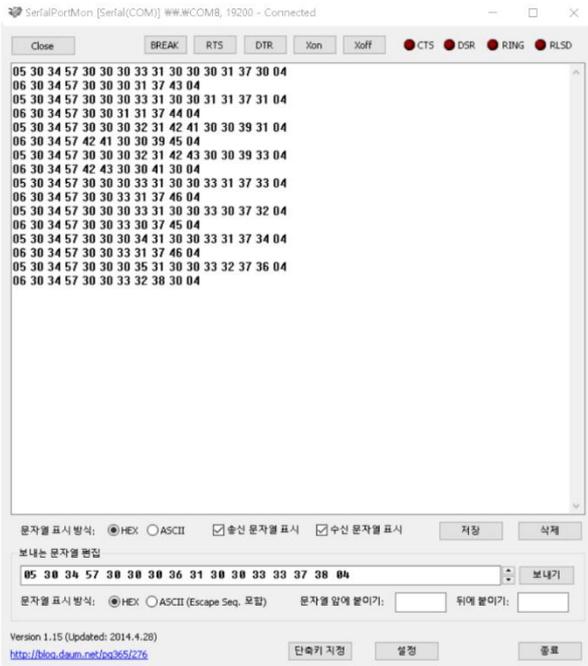
\*33 30\*은 ASCII 에서 '0'을 나타내는 0x30 을 의미, \*33 은 ASCII 1 번째 자리



05(ENQ) 30 34 57 30 30 30 \*34 31 30 30 \*33 31\* 37 34 04(EQT) (0 1 \_\_ 출력)  
\*33 31\*은 ASCII 에서 '1'을 나타내는 0x31 을 의미, \*34 는 ASCII 2 번째 자리



05(ENQ) 30 34 57 30 30 30 \*35 31 30 30 \*33 32\* 37 36 04(EQT) (0 1 2 \_ 출력)  
 \*33 32\*은 ASCII 에서 '2'을 나타내는 0x32 을 의미, \*35 는 ASCII 3 번째 자리



05(ENQ) 30 34 57 30 30 30 \*36 31 30 30 \*33 33\* 37 38 04(EQT) (0 1 2 3 출력)  
 \*33 33\*은 ASCII 에서 '3'을 나타내는 0x33 을 의미, \*36 는 ASCII 4 번째 자리