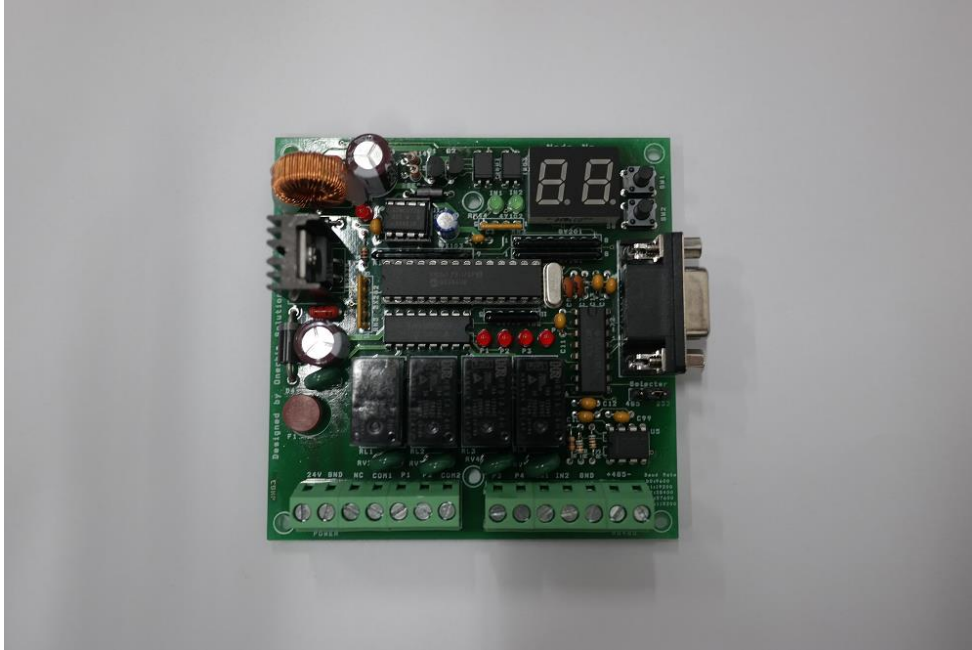


## Semi IO Lite



### 1.개요

Semi IO Lite 는 PLC 나 PC 등 호스트장치와 RS485(RS232)통신으로 연결하여 릴레이 4 개의 출력을 제어하거나 입력 2 점을 확인하는 리모트 I/O 보드입니다. Semi IO Lite 는 특히 LS 산전의 SV-iG5 인버터 통신 프로토콜에 호환되도록 설계되었으며 PLC 에 내장된 인버터 제어함수를 사용하여 제어가 간단합니다. 특히 Semi IO Lite 는 최소한의 입출력만으로 강력한 기능을 제공하므로 간편하고 경제적인 시스템을 구현할 수 있습니다.

### 2.일반사항 및 지원기능

전원 : DC24V

소비전류 : 500mA

Baudrate : (9600, 19200, 38400, 57600) 선택 N,8,1

ID설정 : (0~20까지 21가지)

출력 : SPST릴레이 4점 (250V 5A, 28V 10A)

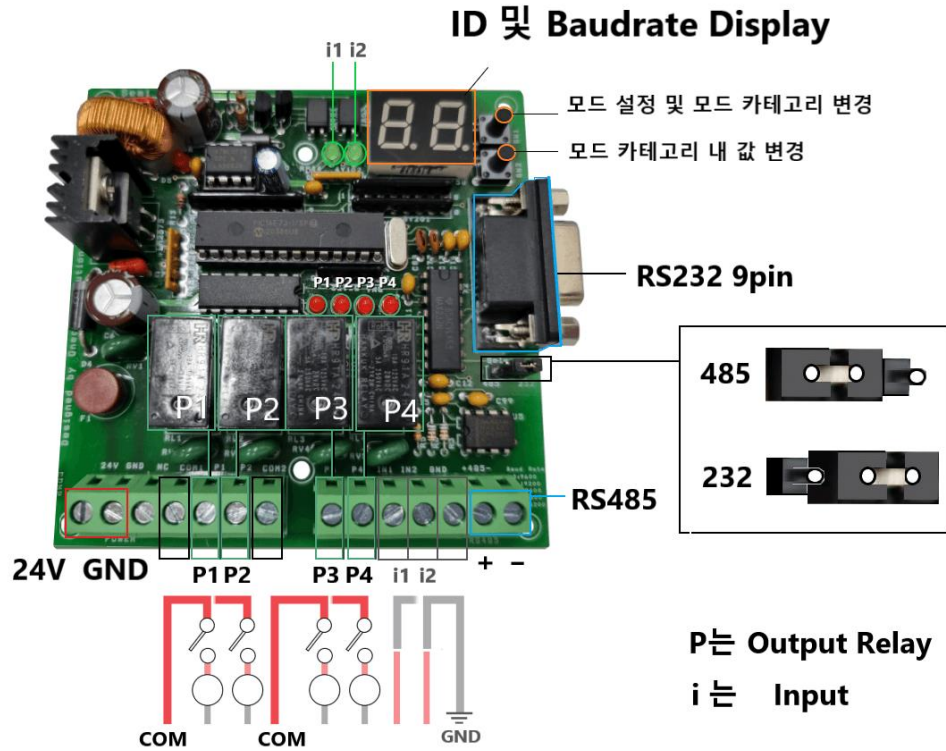
입력 : i1,i2 접점 입력 2점

통신 : RS232통신, RS485통신

전송형태 : LG BUS 호환 멀티드롭 링크 시스템

사이즈 : 91 mm x 91 mm,

### 3.외형



P는 Output Relay  
i 는 Input

### 4.설정

#### 국번 설정 및 통신속도 설정방법

- 1 디스플레이 오른쪽 스위치 SW1을 한번 누른다
- 2 XX 점멸됨 (국번 ID 설정모드)
- 3 스위치 SW2를 눌러 국번 ID 설정 (0 - 20)
- 4 국번이 설정되었으면 스위치 SW1을 눌러 보레이트 설정 모드로 간다
- 5 bX 점멸됨 (보레이트 설정모드)
- 6 스위치 SW2를 눌러 통신속도 (보레이트) 설정  
(b0 : 9600bps, b1 : 19200 bps, b2 : 38400 bps, b3 : 57600)
- 7 통신속도가 설정되었으면 스위치 SW1을 눌러 저장하고 빠져나온다

위는 국번과 baudrate 를 설정하는 방법입니다.



### 5. 명령어 형태

Semi\_IO\_Lite 통신 쓰기 명령 5가지 (국번 04로 가정)

명령어 형식	ENQ	국번	명령	주소	갯수	데이터	SUM	EOT
릴레이 출력	ENQ	0 4	W	0 0 0 1	1	0 0 X X	X X	EOT
타임아웃 리셋	ENQ	0 4	W	0 0 0 6	1	X X X X	X X	EOT
입력 유지 명령	ENQ	0 4	W	0 0 0 A	1	X X X X	X X	EOT

국번은 01 또는 20로 한정 / ASCII 코드로 전송      쓰기에 ACK 및 응답은 타임아웃리셋에서 설정

Semi\_IO\_Lite 통신 읽기 명령 2가지 (국번 04로 가정)

응답지연시간 값 x 10ms      ex) 'A'일경우 100ms

명령어 형식	ENQ	국번	명령	주소	갯수	SUM	EOT
DC24V 입력 읽기	ENQ	0 4	R	X X 0 7	1	X X	EOT
입력카운터값 읽기	ENQ	0 4	R	X X 0 B	1	X X	EOT

Semi\_IO ---> ACK + (국번 : 2) + R + (데이터 : 4) + (SUM : 2) + EOT

\*위 명령어는 아스키 문자를 기준으로 한 것입니다. HEX 전송시에 아스키 문자의 HEX값에 맞게 보내셔야 합니다.

ex) 위에 나온 입력 카운터 값 읽기를 HEX로 하면,

0x06 0x30 0x34 0x52 0x30 0x30 0x30 0x41 0x31 0x43 0x41 0x04

### 6. 명령어 예시

1) 릴레이 출력 방법 (어드레스: 0001)

릴레이는 4개가 한바이트를 형성: P1--> 비트0번, P4--> 비트3번에 각각 할당

릴레이 1,3 번을 ON : 2,4 번을 OFF 시킨다면

----> 해당하는 값을 2진수로 표시하면 : 00000101

----> hex값으로 표시하면 : 05

ENQ	국번	명령	주소	갯수	데이터	SUM	EOT
ENQ	0 4	W	0 0 0 1	1	0 0 0 5	7 2	EOT

SUM(ASCII) : 0+4+W+0+0+0+1+1+0+0+0+5 --> SUM(hex값) : 30+34+57+30+30+30+31+31+30+30+30+35 = 272

2) 타임아웃리셋(데이터 하위2바이트) 및 쓰기응답ON(데이터 상위2바이트) 설정하기

/ 호스트장치의 오류등으로 통신이 일정시간 이루어지지 않았을때 출력을 강제리셋합니다 \*릴레이 동작 끄는것을 의미

/ 초기 파워온시 아무런 명령이 없으면 타임아웃리셋 오프상태입니다

/ 리셋시간은 100ms 단위로 최소 100ms에서 최대 1초까지 설정가능하고 그 이상은 1초로간주됩니다

/ 리셋은 명령중 호스트 명령이 없어도 출력을 계속 유지하고 싶을때 0을 써주면 리셋오프됩니다

/ 쓰기응답을 활성화 할려면 데이터 상위 바이트에 1을, 불활성화는 0을 써주면 됩니다

/ 초기 파워온시 쓰기응답은 불활성화 되어있습니다

\*쓰기응답이란?

쓰기 명령어 수행시 ACK(0x06)

/ 타임아웃 리셋값을 500 ms 로 설정하고 쓰기응답은 오프로 설정

으로 시작하는 보낸 명령의 값을 뺀것

ENQ	국번	명령	주소	갯수	데이터	SUM	EOT
ENQ	0 4	W	0 0 0 6	1	0 0 0 5	7 2	EOT

SUM(ASCII) : 0+4+W+0+0+0+1+1+0+0+F+5 --> SUM(hex값) : 30+34+57+30+30+30+31+31+30+30+30+35 = 272

쓰기응답(상위)

타임아웃(하위)

\*SUM도 아스키 값이므로, HEX 값으로 전송시 뒤 두자리를 각각 해당하는 아스키문자의 HEX값으로 바꿔 보내야 합니다.

### 3) 입력유지명령 설정하기

- / 호스트장치의 스캔타임이 늦어서 펄스폭이짧은 입력을 읽지 못하는 경우 신호를 일정시간 메모리상 유지시켜 주는 기능입니다
- / 설정가능시간은 4ms 부터 1초 까지 가능하며 단위는 4ms 입니다
- / 설정시간보다 긴펄스가 들어오면 유지기능이 무시되고 짧은펄스가 들어올경우에만 작동합니다
- / 입력유지 명령을 실행하려면 데이터 상위 바이트에, 유지시간을 하위 바이트에 해당핀을 셋트하면 됩니다
- / : IN2(비트1)를 입력유지 20ms 하고 IN1(비트0) 는 유지하지 않는다면 상위바이트는 5 (4ms X 5 = 20ms)를 하위는 0000010 (2)를 써준다

ENQ	국번		명령	주소				갯수	데이터				SUM		EOT
ENQ	0	4	W	0	0	0	A	1	0	5	0	2	8	4	EOT

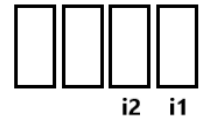
SUM(ASCII) : 0+4+W+0+0+0+A+1+0+5+0+2 --> SUM(hex사값) : 30+34+57+30+30+30+41+31+30+35+30+32 = 284



### 4) DC 24V 입력 읽기

- / 읽기 명령은 쓰기명령과 달리 명령을 주고 나서 Semi\_IO 에서 반응되는 응답을 읽어야 합니다
- / 따라서 명령을 내리는 PLC 나 CuBloc 의 처리능력에따라 응답지연시간이 필요합니다
- / DC 24V 입력 읽기명령을 받고 10ms 후에 응답하라는 명령은 다음과 같습니다
- / 이때 입력값을 0000 0000 0000 0011 이라하면 \*0011이라는 것은 i1과 i2 둘다 HIGH 상태인것

데이터값 끝자리 비트 4자리



ENQ	국번		명령	주소				갯수	SUM		EOT	PLC--> Semi_IO	
ENQ	0	4	R	0	A	0	7	1	B	F	EOT		
			읽기	10 ms --> hex사로 0A									
SUM(ASCII) : 0+4+R+0+A+0+7+1 --> hex사 : 30+34+52+30+41+30+37+31 = 1BF													
<b>Semi IO Lite -&gt; PLC</b> <b>뱉어내는 입력 상태 값</b>				ACK	국번		명령	데이터값 아스키				SUM	EOT
				ACK	0	4	R	0	0	0	3	A	2
						읽기	0011						

\*i1과 i2 두 입력이 어떤 상태인지 읽어오는 것

### 5) 입력카운터값 읽기

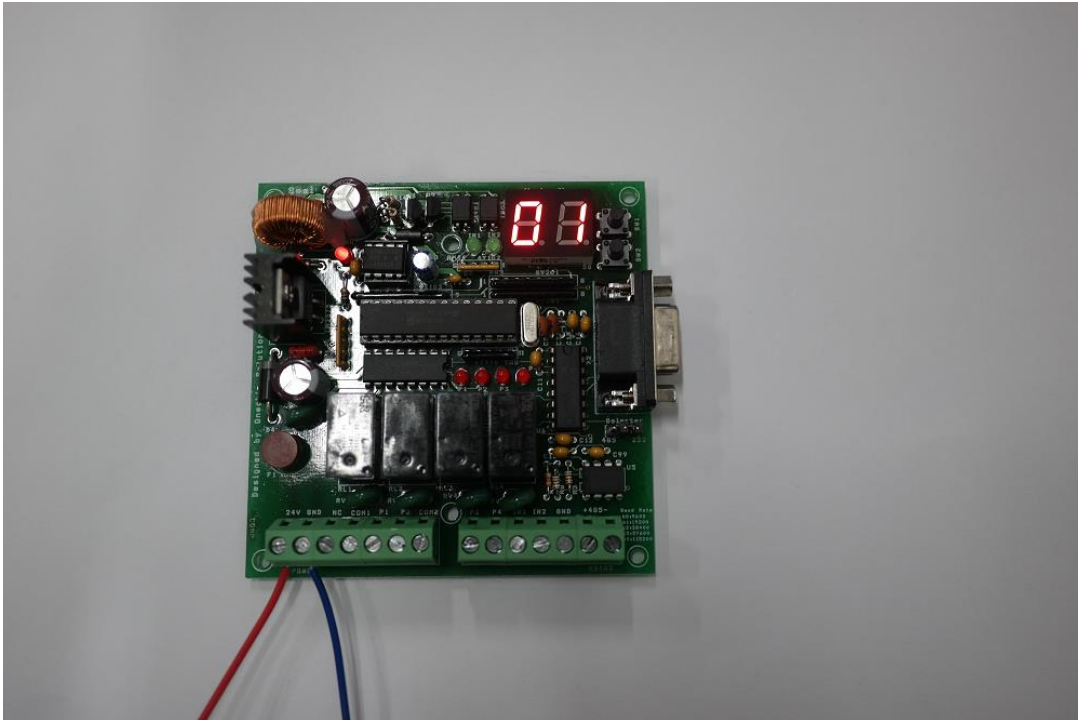
- / 입력핀 IN1 은 카운터로도 사용할수있습니다
- / 카운터 읽기 명령을 내리면 마지막 읽기 명령이후로 IN1 핀에 들어온 카운터값이 읽혀지고 카운터는 리셋됩니다
- / 카운터값은 설정여부와 관계없이 자동으로 처리되므로 메모리에서 읽기만 하면 됩니다 (반드시 입력유지명령은 OFF 여야합니다)
- / DC24V 입력과 마찬가지로 응답딜레이시간을 설정할수 있습니다
- / 이때 카운터값이 1010 0000 0101 1111 (십진수 41,055 ) 이라하면

ENQ	국번		명령	주소				갯수	SUM		EOT	PLC--> Semi_IO	
ENQ	0	4	R	0	A	0	B	1	C	A	EOT		
			읽기	10 ms --> hex사로 0A									
SUM(ASCII) : 0+4+R+0+A+0+B+1 --> hex사 : 30+34+52+30+41+30+42+31 = 1CA													
<b>Semi_IO --&gt; PLC</b> <b>뱉어내는 값</b>				ACK	국번		명령	데이터값				SUM	EOT
				ACK	0	4	R	A	0	5	F	A	2
						읽기	1010 0000 0101 1111						

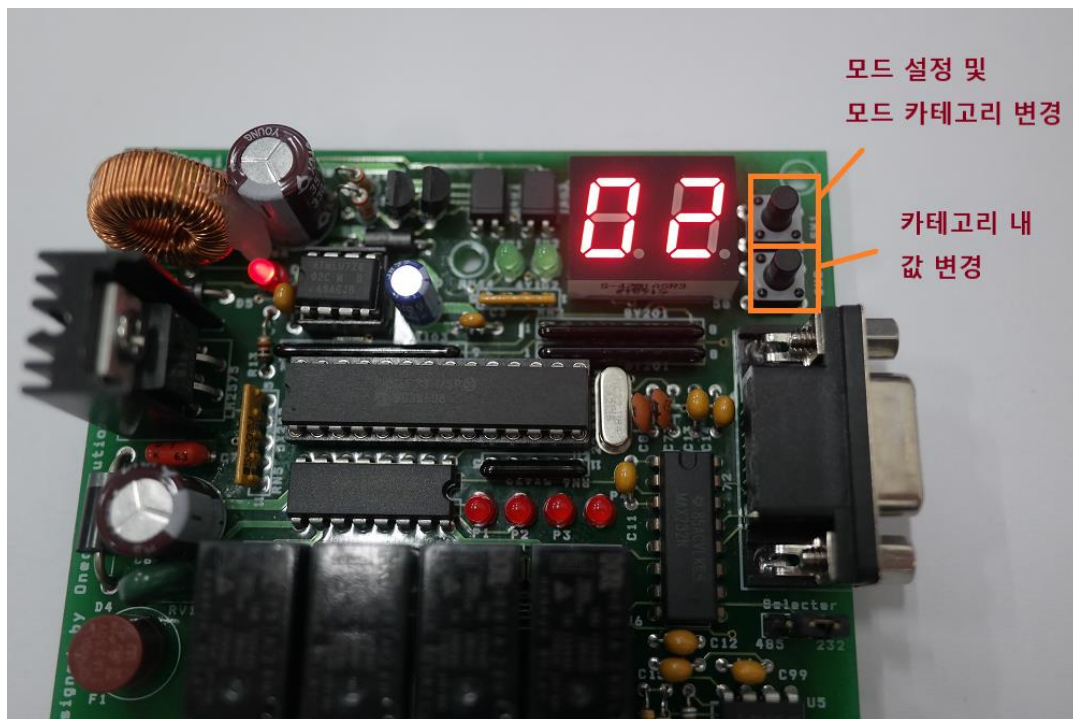
즉 i1이 카운터값 읽기 명령 전까지 41,055번의 입력이 있었던 것

- \*입력카운터값 읽기는 i1 입력만 사용하실 수 있습니다. (i1의 입력수만 카운트 합니다.)
- \*입력카운터값 읽기는 명령하면, i1의 이전 카운트값은 초기화 되고 i1의 입력을 다시 카운트 하게 됩니다.

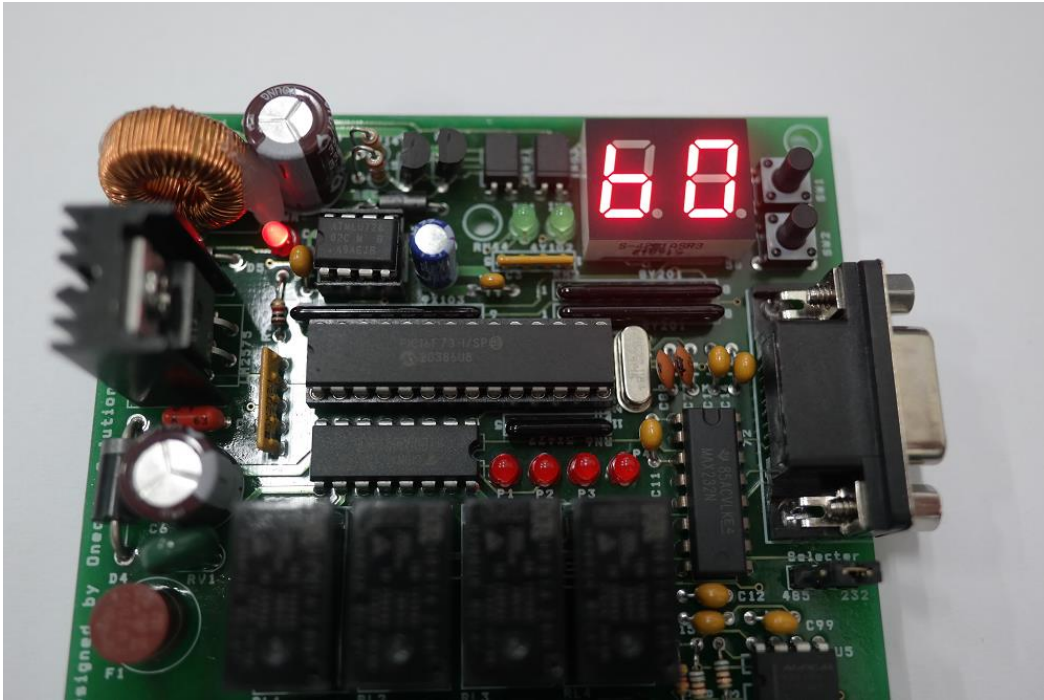
### 7.작동사진



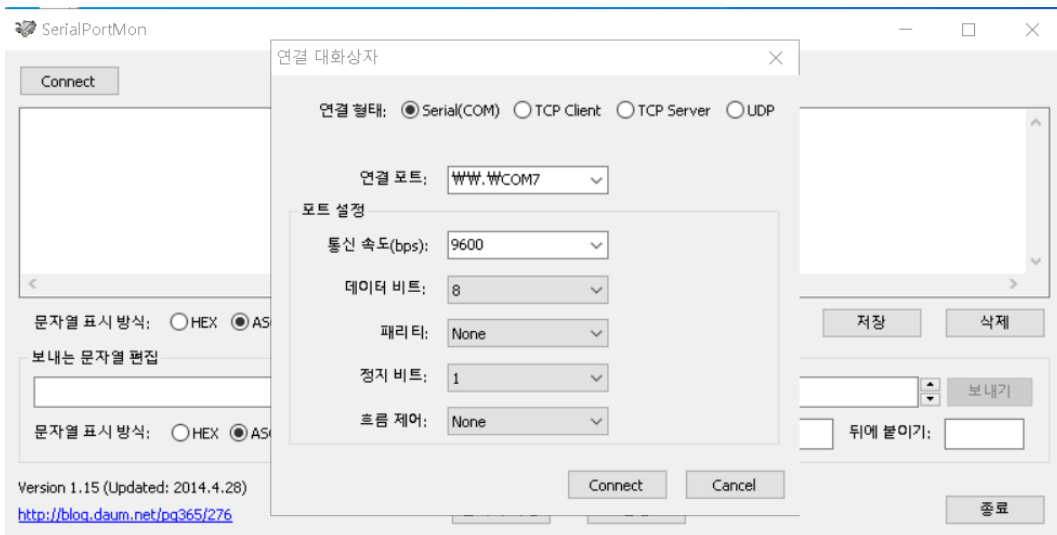
Semi IO Lite 에 24V 전원을 공급한 이미지입니다.



ID(국번)와 baudrate 를 설정해줍니다. \*여기서 모드란, ID 와 baudrate 변경을 얘기합니다.



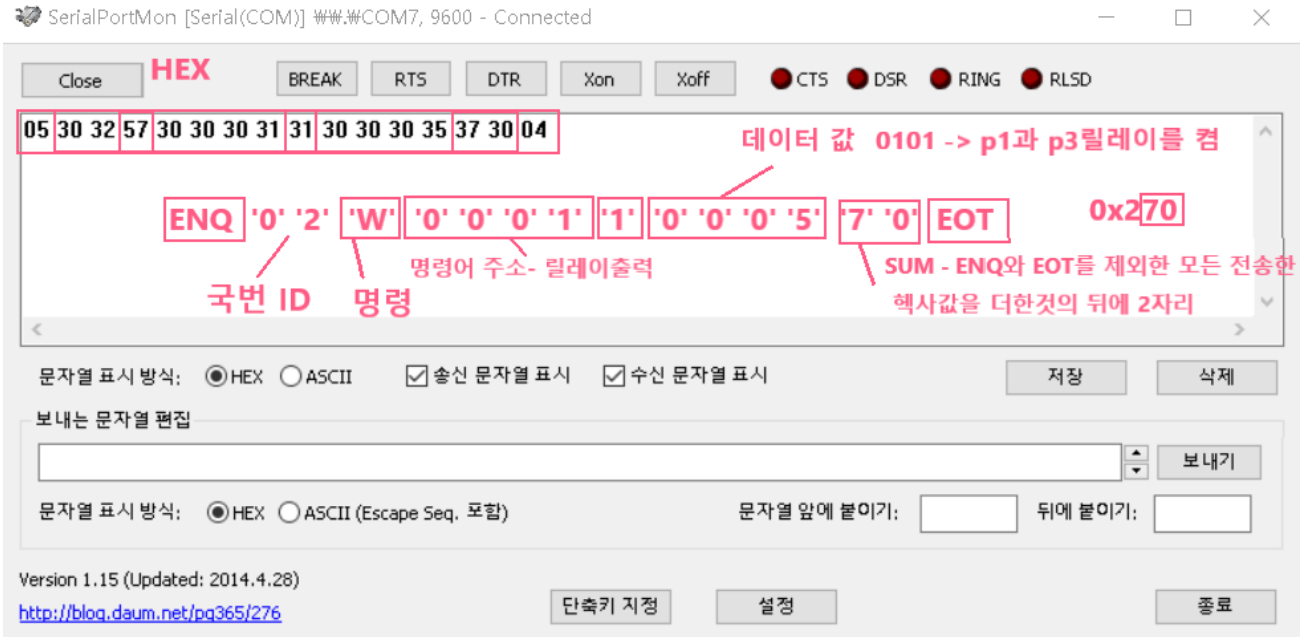
ID(국번)은 02 로 baudrate 는 b0(9600)으로 맞춰줍니다. 이제 오른쪽의 DSUB 9pin 단자에 DSUB to USB 케이블을 사용하여 PC 와 시리얼 통신을 할 것입니다.



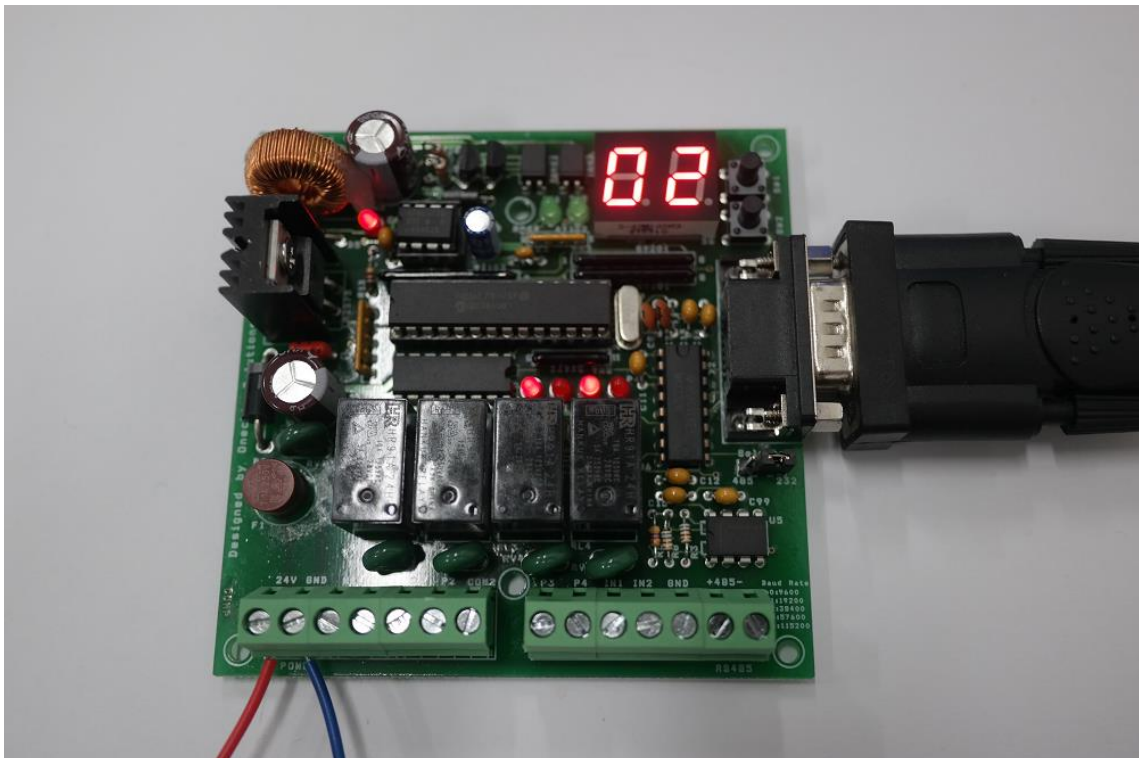
이제 PC 로 넘어가서 명령어들을 Semi IO Lite 에 보내줄 것입니다.

SerialPortMon 이라는 시리얼 전송 및 모니터링 응용프로그램을 사용하였습니다.

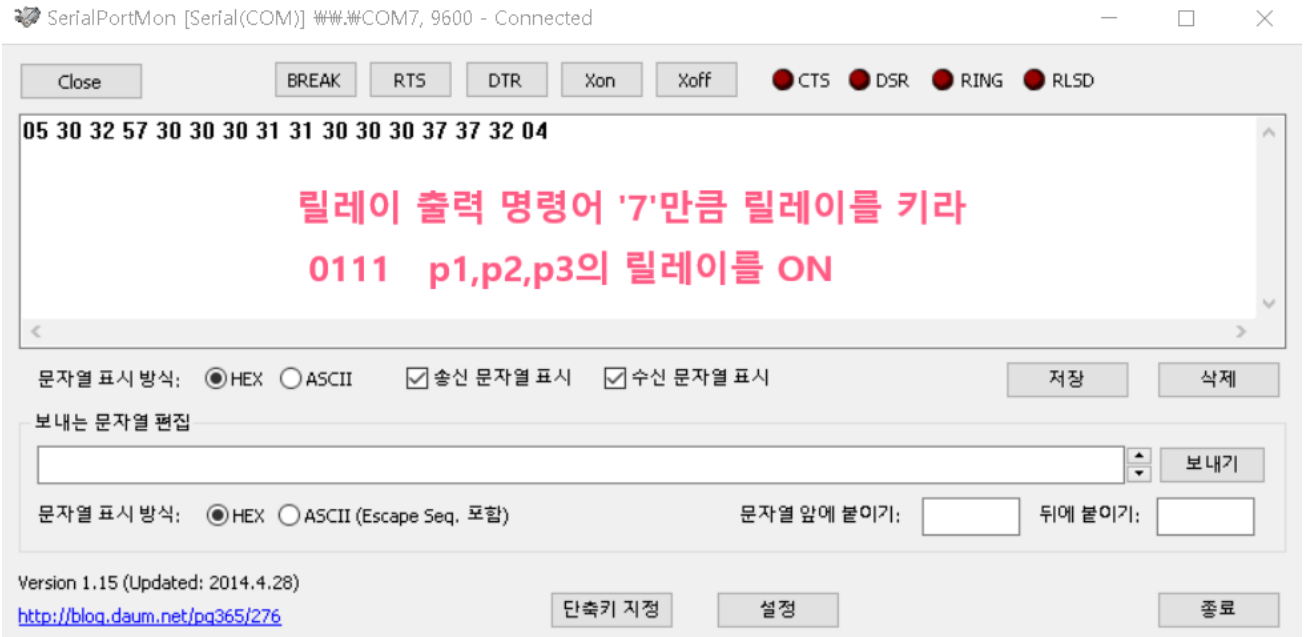
값은 전부 HEX 형태로 보내겠습니다.



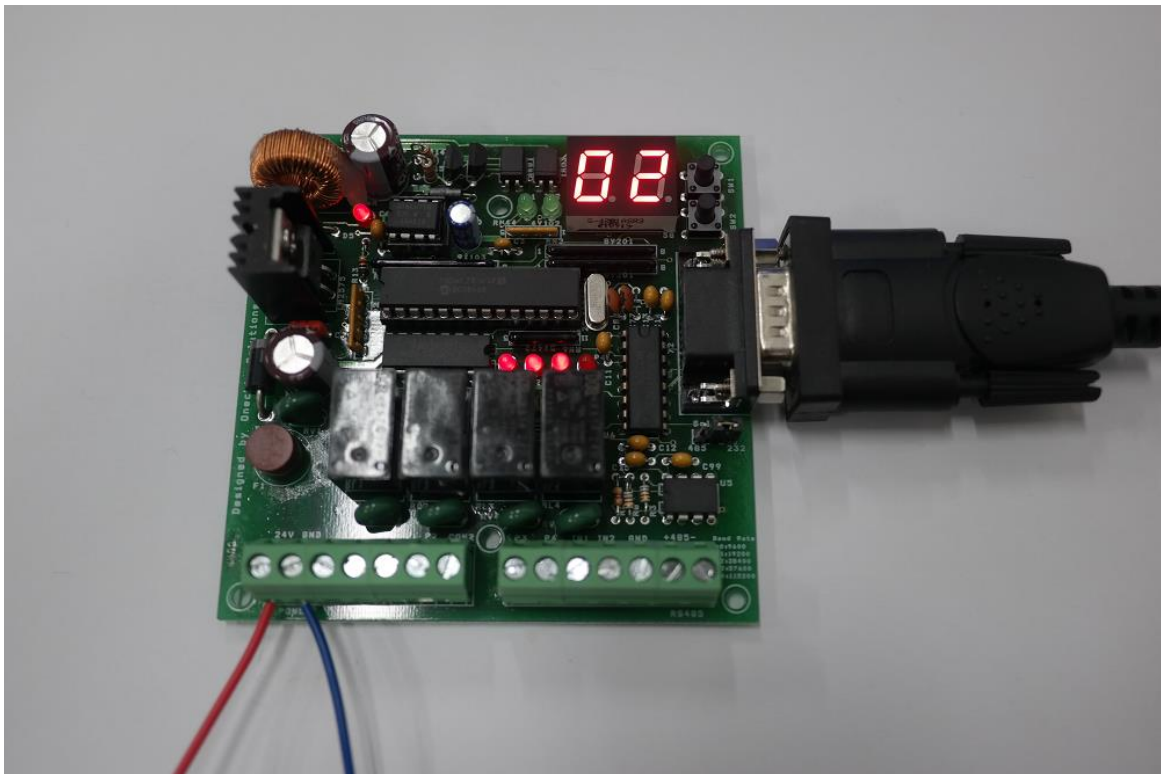
먼저 쓰기 명령어를 통해 p1 과 p3 릴레이를 키는 명령어를 보냈습니다.



위처럼 p1 과 p3 의 릴레이가 켜진 것을 LED 를 통해 확인할 수 있습니다.



이번에는 p1 p2 p3 릴레이 3 개를 키는 명령어를 보내보았습니다.

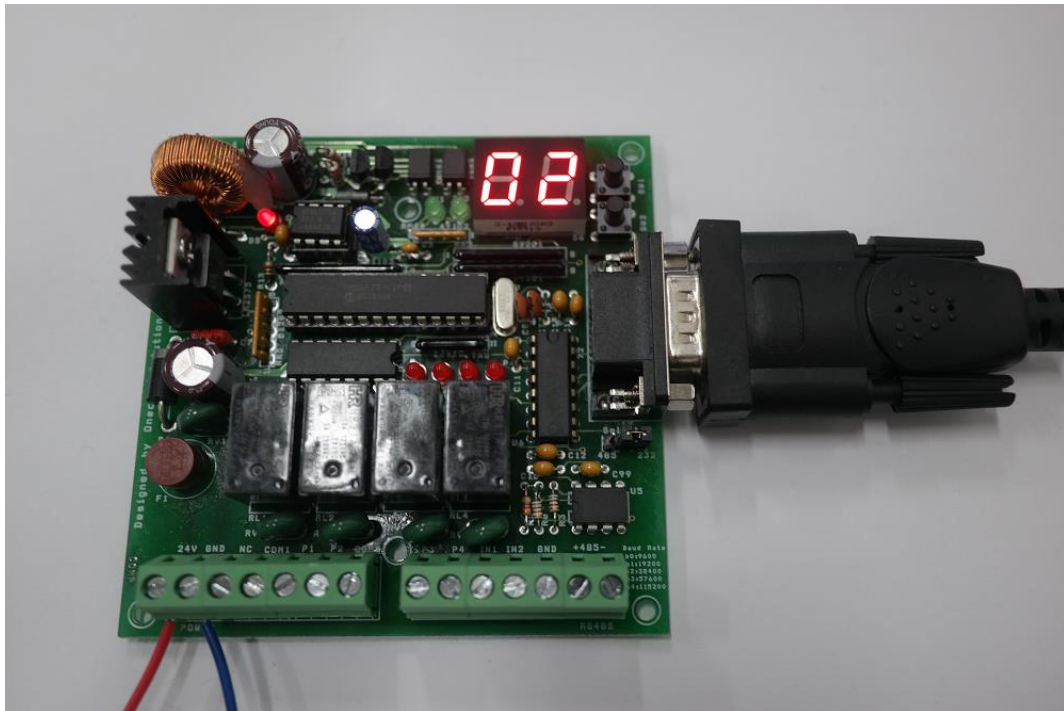


위처럼 p1 과 p2, p3 릴레이 3 개가 켜진 것을 LED 를 통해 확인할 수 있습니다.

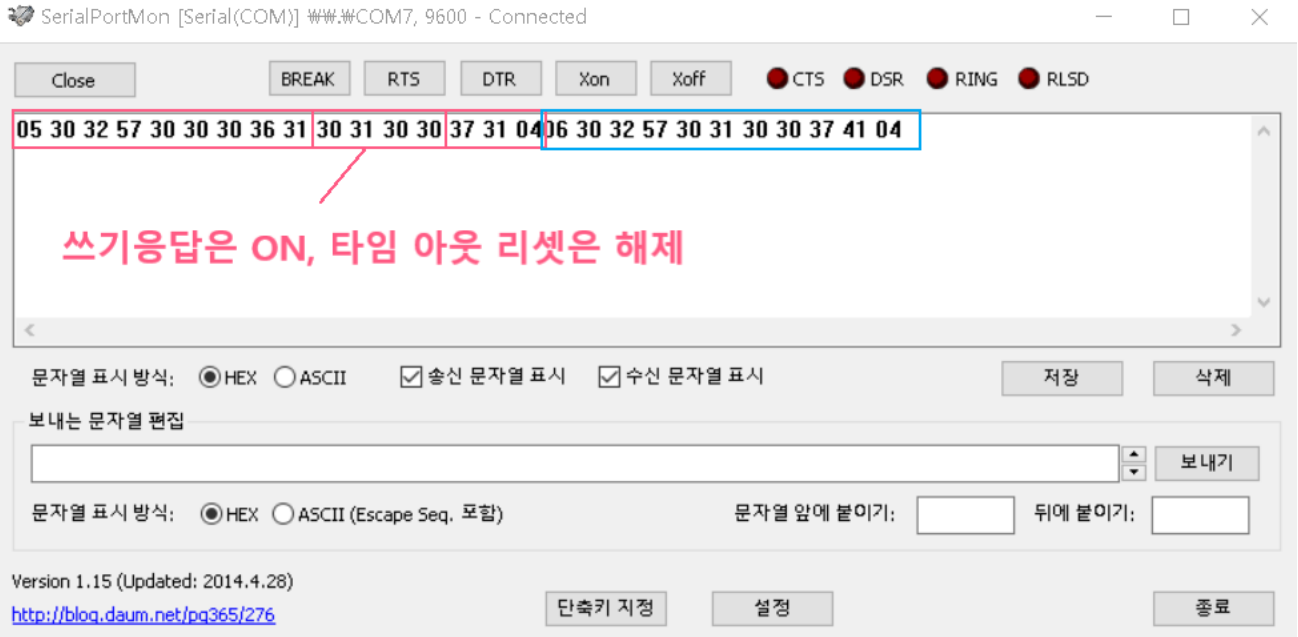




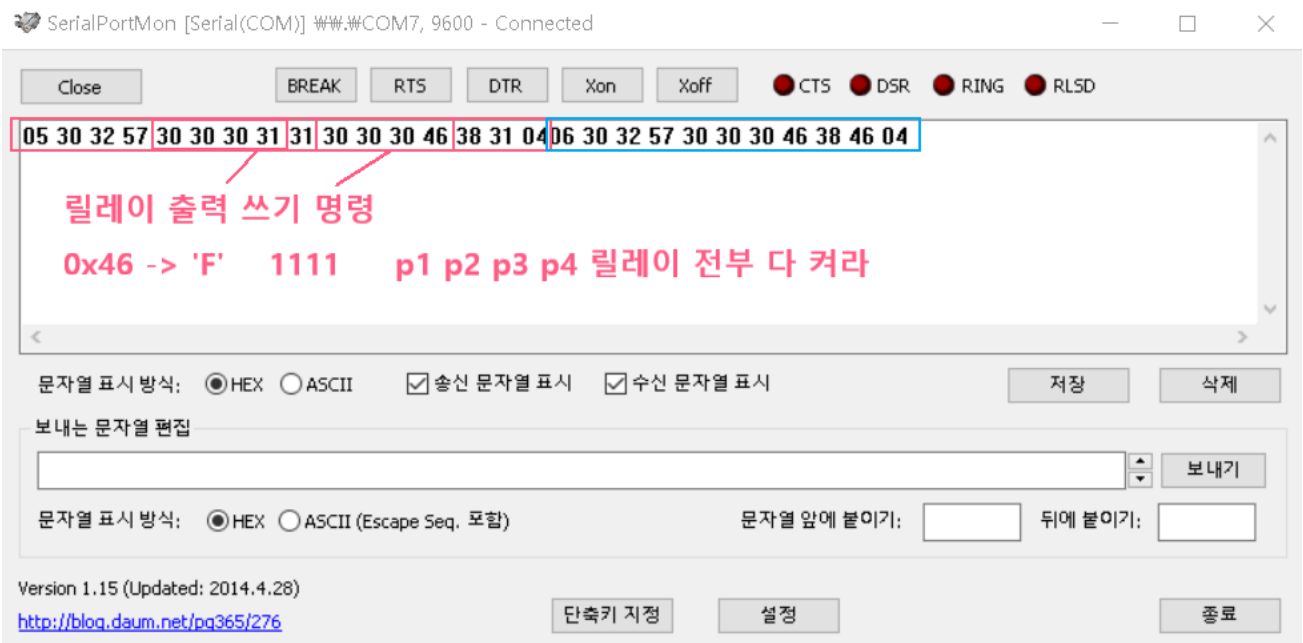
이번에는 쓰기응답과 타임아웃 리셋 명령어를 보내보았습니다. 쓰기응답은 ON 으로 타임 아웃 리셋은 500ms 로 설정하였습니다. 타임 아웃 리셋이란 일정시간 동안 호스트로부터 Semi IO Lite 가 명령어를 받지 못하면, 릴레이들의 출력을 꺼버리는 명령입니다. 명령어를 보내자마자 바로 옆에 ACK 로 시작하는 응답데이터가 날라온 것을 볼 수 있습니다.



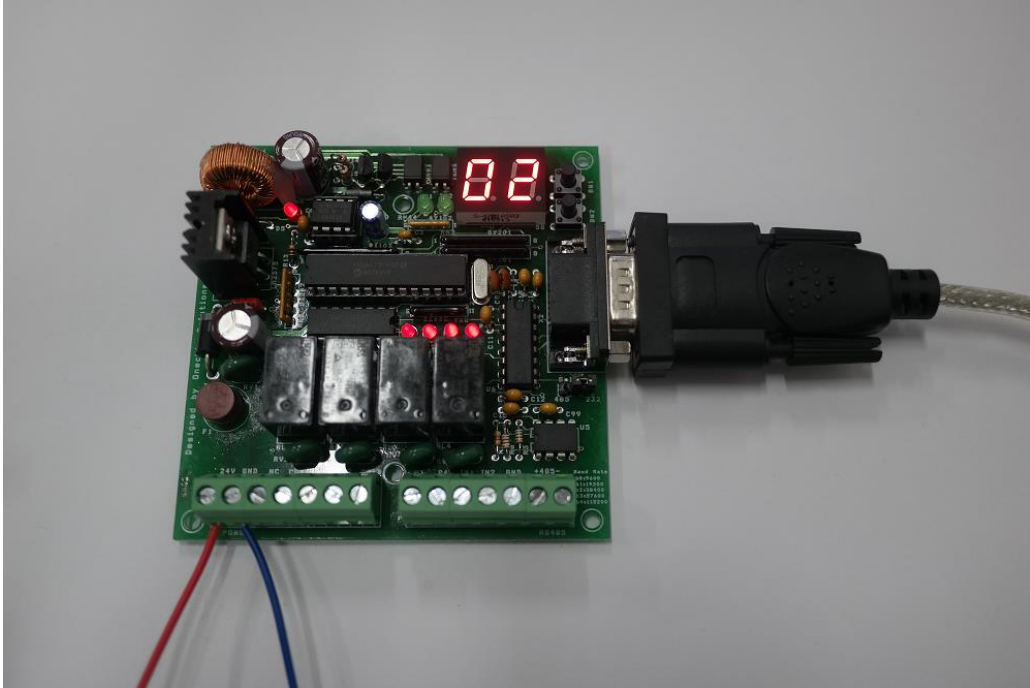
명령을 보내고 타임 아웃 리셋 설정으로 인하여, 500ms 이후 모든 릴레이 출력이 꺼진 것을 LED 를 통해서 볼 수 있습니다.



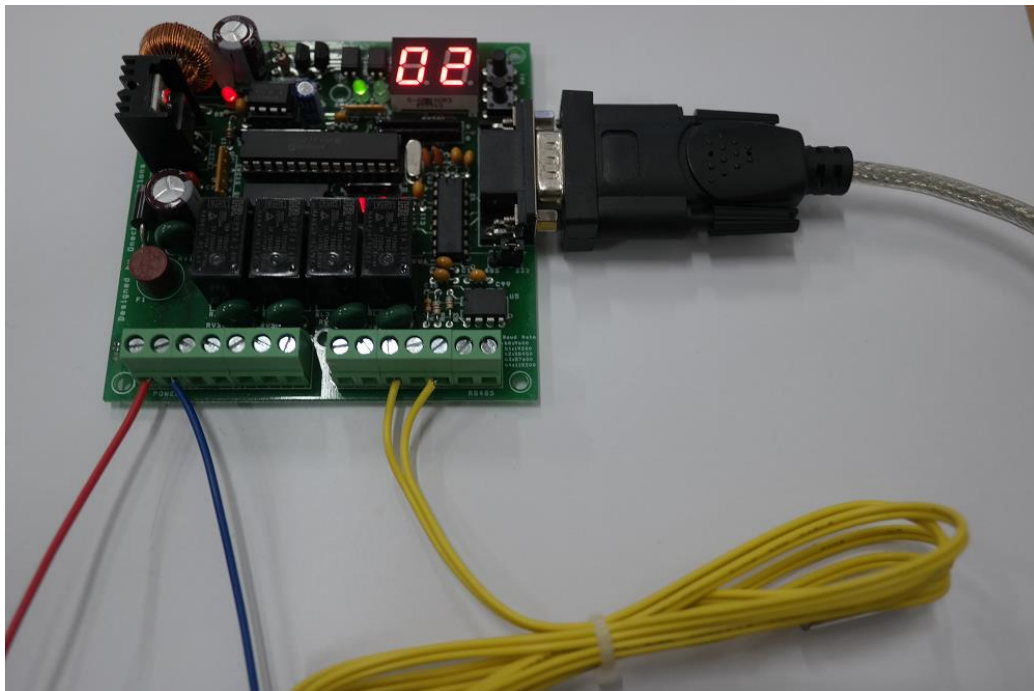
이번에는 릴레이 출력을 유지하기 위해서 쓰기 응답은 ON 하되, 타임 아웃 리셋은 해제하는 명령어를 다시 보내보았습니다.



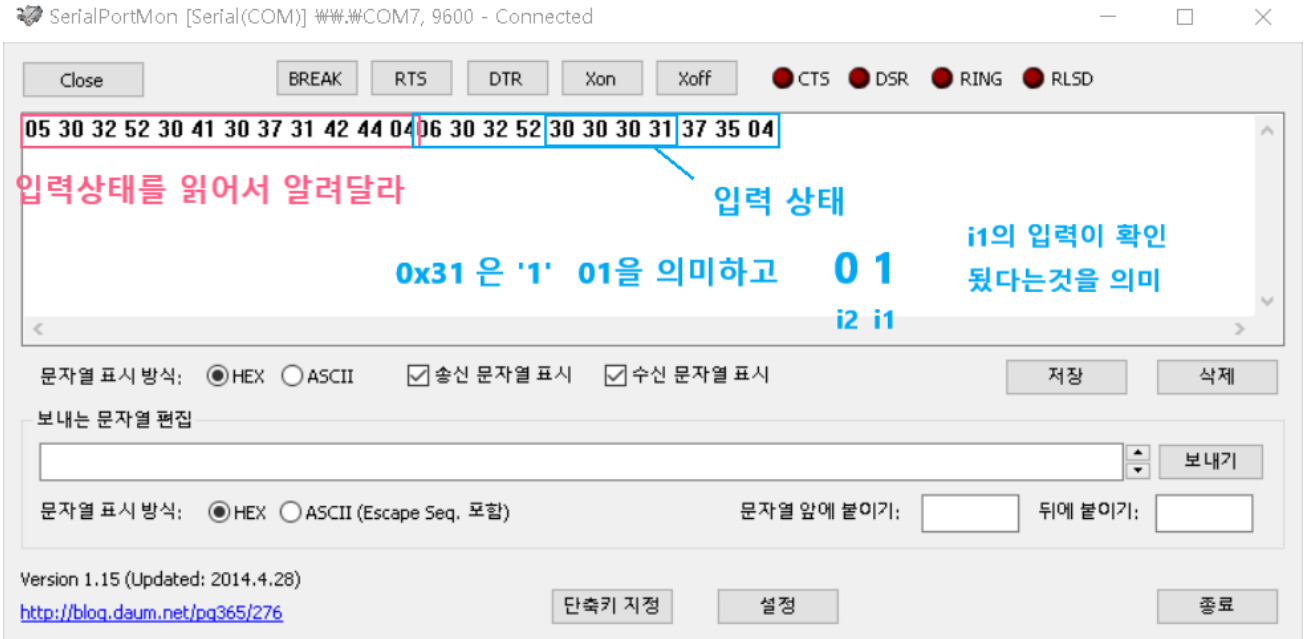
바로 이어서 p1, p2, p3, p4 모든 릴레이를 출력 ON 시키는 쓰기 명령을 보내었습니다. 쓰기 명령 이후에 ACK로 시작하는 응답 데이터가 온 것을 확인할 수 있습니다.



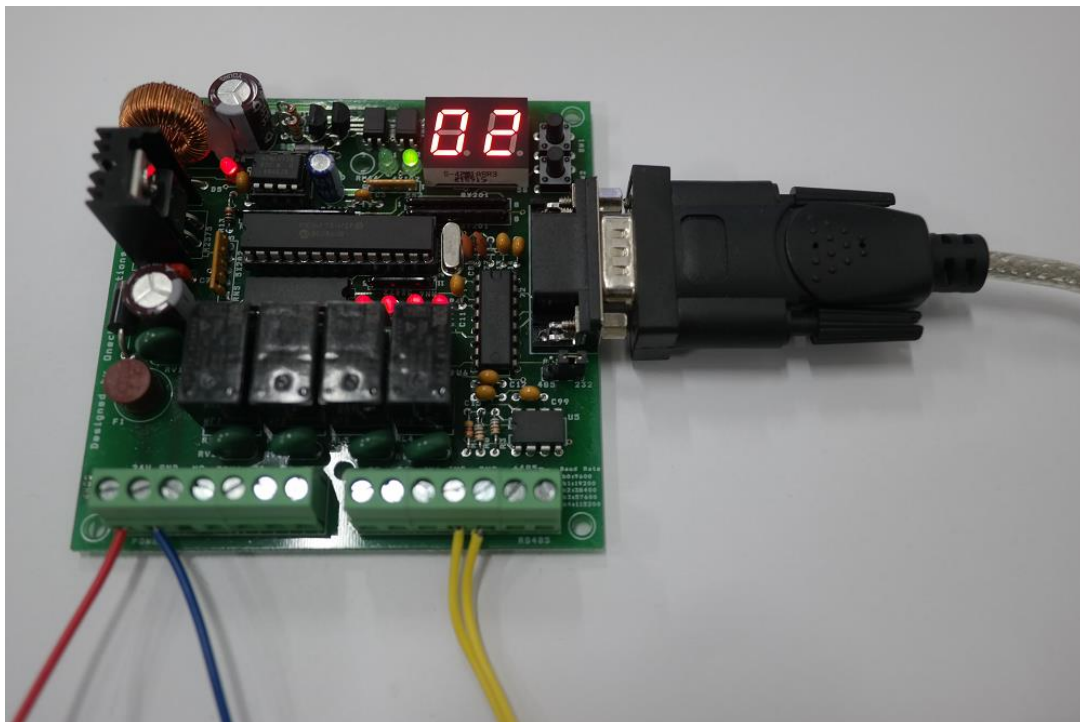
위처럼 p1, p2, p3, p4 모든 릴레이가 출력이 ON 된 것을 LED 를 통해 확인하실 수 있습니다. 쓰기응답은 ON 이 되었지만, 타임 아웃 리셋은 없으므로 명령 이후 릴레이 출력이 유지되는 것을 볼 수 있습니다.



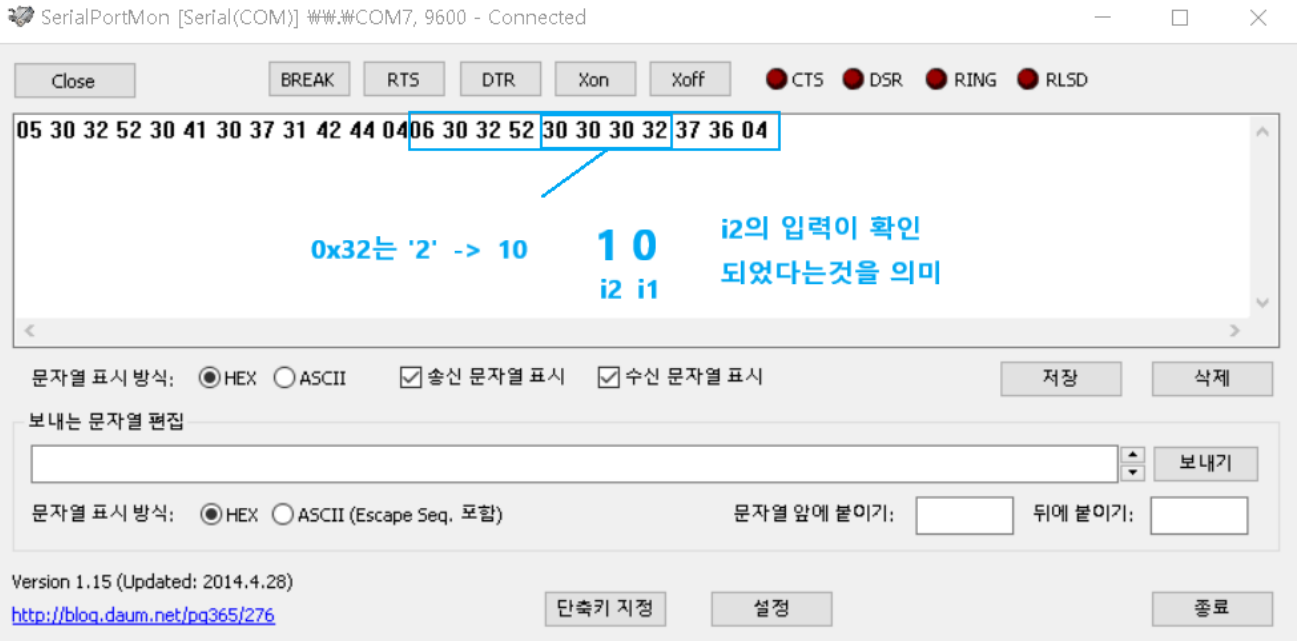
다음은 입력 명령을 보내기 위해 입력 단자에 입력을 넣어보았습니다. 저 도선은 온도센서인데, Semi IO Lite의 입력은 접점 입력(1,0)만 받기 때문에, 온도값이 날아오지 않습니다. 제 눈 앞에 보이는 도선이라 그냥 가져다 입력으로 사용한 점 양해 부탁드립니다. 보면 i1에 입력이 들어오자 i1의 녹색 LED가 켜진 것을 볼 수 있습니다.



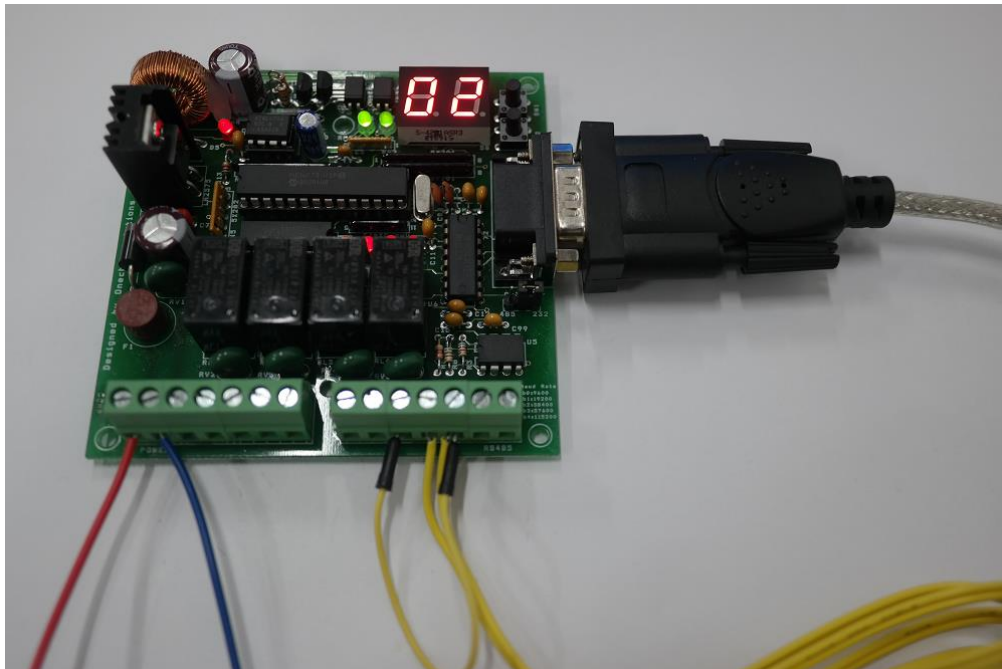
입력상태를 읽어서 알려달라는 명령어를 보내보았습니다. 그에 대한 응답 값을 바로 보내준 것을 확인할 수 있습니다. 값의 영역에 0x31 이 날라왔다는 것은 아스키로 '1'이라는 문자가 날라온 것인데 이것은 이진수로 01 을 나타내고 오른쪽 비트가 i1 입력, 왼쪽 비트가 i2 의 입력상태이기에 즉, i1 이 입력 HIGH 상태라는 것을 나타냅니다.



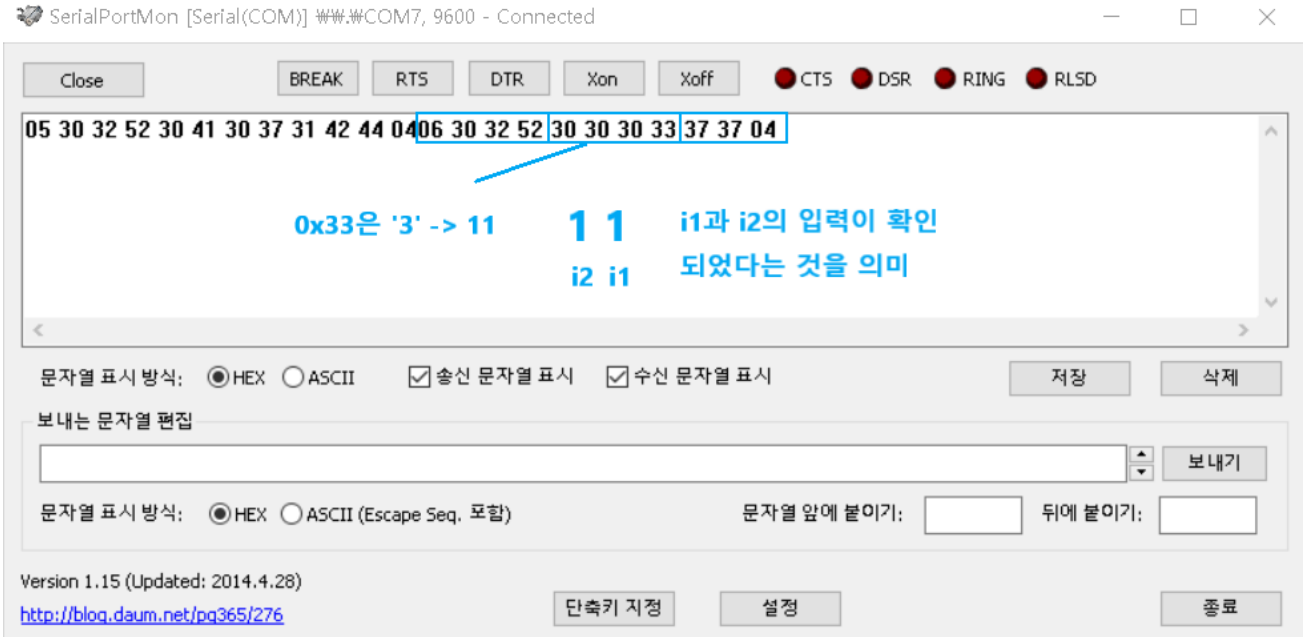
이번에는 i1 이 아니라 i2 에다가 도선을 연결하여 i2 의 입력을 HIGH 상태로 만들어보았습니다. i2 의 녹색 LED 에 불이 들어온 것으로 입력 여부를 확인할 수 있습니다.



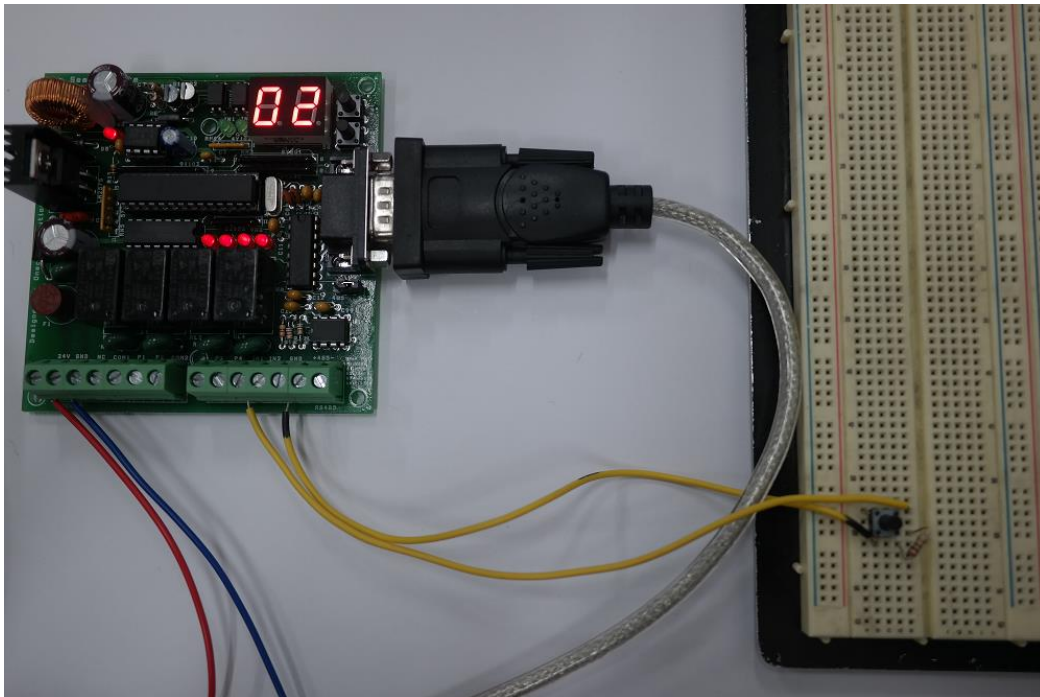
위처럼 아까와 동일하게, 명령어로 입력을 확인하는 명령어를 보내니, 응답 값을 보내준 것을 볼 수 있습니다. 이번에는 0x32 아스키 문자로 '2'가 날라왔는데 이를 이진수로는 10 이 되고 이 비트값을 봤을 때 i2 의 입력이 HIGH 상태라는 것을 알 수 있습니다.



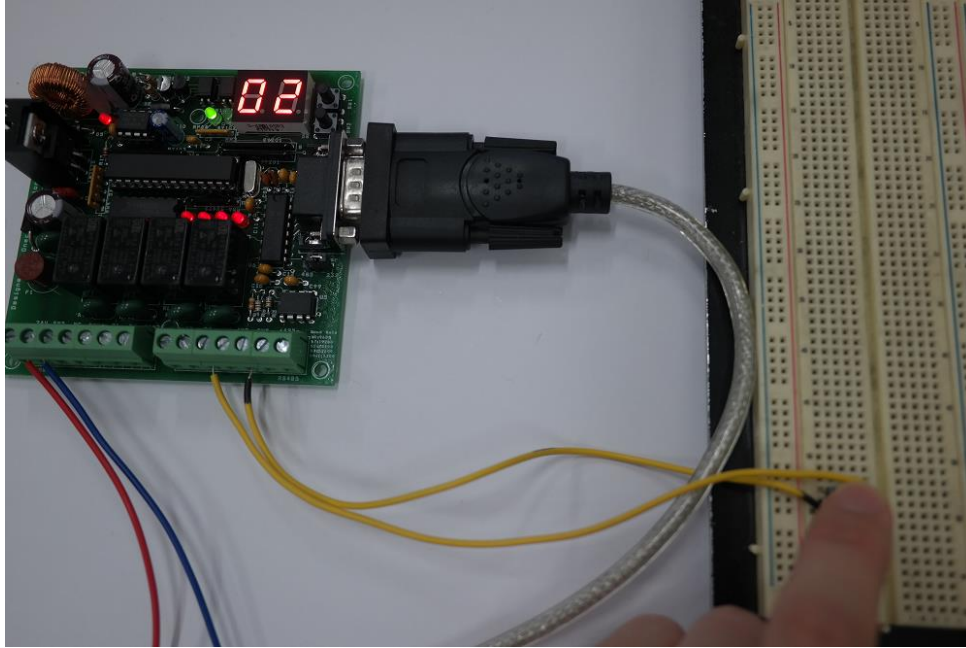
이번에는 i1 과 i2 두 입력상태를 HIGH 로 만들기 위해 둘 다 도선을 연결하였습니다. 위처럼 i1 과 i2 두 입력의 녹색 LED 가 불이 켜진 것을 확인할 수 있습니다.



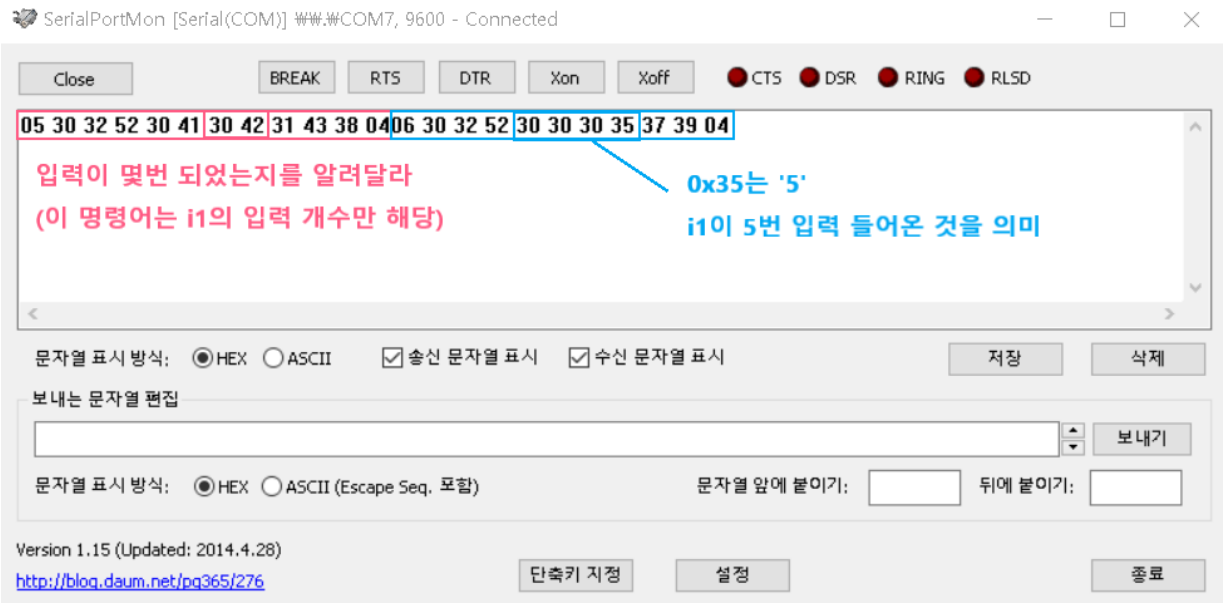
이번에는 같은 명령어를 보내니 응답값으로 0x33 이 온 것을 볼 수 있습니다. 0x33 은 아스키 문자로 '3'이며 3 을 이진수로 하면 11 이것을 비트로 보았을 때 i1 과 i2 두 입력이 HIGH 상태라는 것을 볼 수 있습니다.



이번에는 입력 카운터값을 얻는 명령어를 보내기 위해 i1 의 입력에 스위치를 달았습니다. 입력 카운터 값은 몇번 입력이 들어왔는지 수를 세는 것이며, 오직 i1 입력만 이 기능을 지원합니다.



이제 이러한 식으로 5 번을 눌러서, 카운팅이 5 번 되었는지를 확인할 것입니다. 누를 때 i1 의 녹색 LED 가 불이 들어오는 것을 확인할 수 있습니다.



입력 카운터값(i1 의)을 보내달라는 명령을 보내자 응답 값으로 0x35 가 날라온 것을 볼 수 있습니다. 0x35 는 아스키 문자로 '5' 5 번 눌렀다는 것을 확인할 수 있습니다. 날라오는 아스키문자는 HEX 를 디스플레이 하는것으로 10 번 누른 값이 날라왔다면 아스키문자 'A'가 날라올 것입니다. 입력 카운터값 읽기 명령어는 한번 실행하면 카운터 값을 보여주고, 그 이전의 카운터 값은 초기화하고 다시 카운팅 합니다. 이 글을 작성하면서 처음 명령어를 보냈을 때는, 이미 입력 읽기 명령어를 보내고 읽었던 상황이라 5 보다 큰 값이 나왔었습니다. 즉 초기화 시키고 다시 버튼 5 번 누른 후 받은 값이 저 값입니다.