SMIC

Connect your devices to TCP/IP network

NetEyeS3400S

（RS485 to Analog Output 4ch）

Analog Output Product

SM Information and Communication

Homepage : http://www.smic21.com
Support : +82-2-6292-2100 khg@smic.co.kr, smic21@smic21.com
Address :A-705 Sktwintech Bld. Gassandong, GeumcheonGu, Seoul Korea Zip 153-023

SMIC

Data Acquisition Device Analog Output

Device Feature

Output channel number----------4
Output Range----------0~10V, 0~5V, 0-20mA, 4~20mA
Output signal----------Current/Voltage
Resolution----------12-bit
Accuracy:
Current Output----------±0.1% of FSR
Voltage Output----------±0.2% of FSR
Voltage Read Back Accuracy----------±0.05% of FSR)
Zero Drift-
Current Output----------±0.2uA/℃
Voltage Output----------±30uV/℃
Output Slope Rate:
Current Output----------0.125 – 128 mA/sec
Voltage Output----------0.0625 – 64 V/sec
Current Load Resistor----------0 to 300
Current Load Resistor at External Power----------0 to 600
External Power Range----------6-24VDC
Output BUS---------- RS485 with Standard Modbus protocol
Power input----------15~24V(AC/DC)
Power consumption----------<0.6W
Ambient temperature:
Operation----------0~70℃(32~158°F)
Storage---------- -20~85℃(-4~185°F)
Ambient humidity----------10%~90%RH
Material,enclosure----------Flame proof plastic
Colour----------Ice Blue
Size----------120*69*25 mm

SMIC

Data Acquisition Device Analog Output

Application:

- Remote data acquisition
- Process monitoring
- Industrial process control
- Energy management
- Supervisory control
- Security systems
- Laboratory automation
- Building automation
- Product testing
- Direct digital control

Wiring diagram and description:

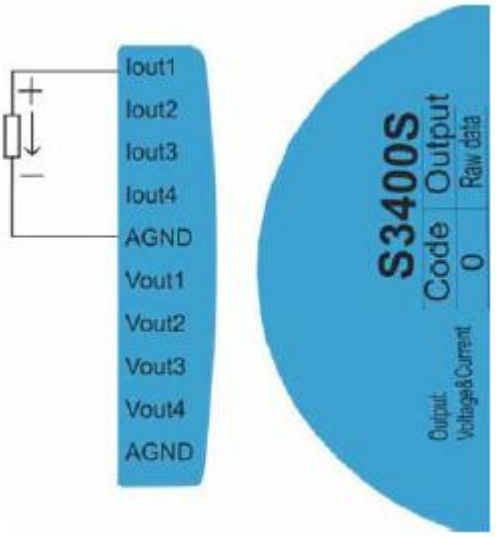


Figure 1 wiring for current output

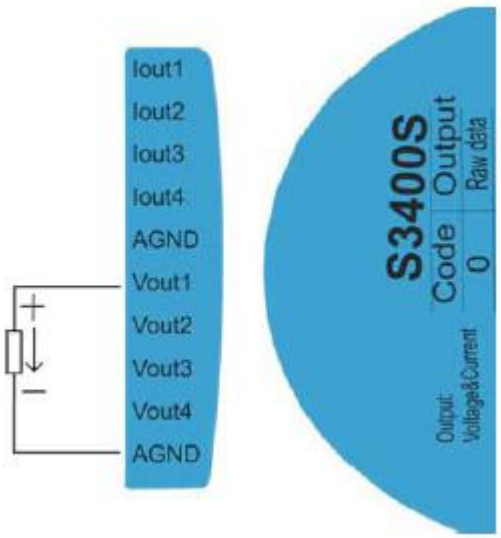


Figure 2 wiring for voltage output

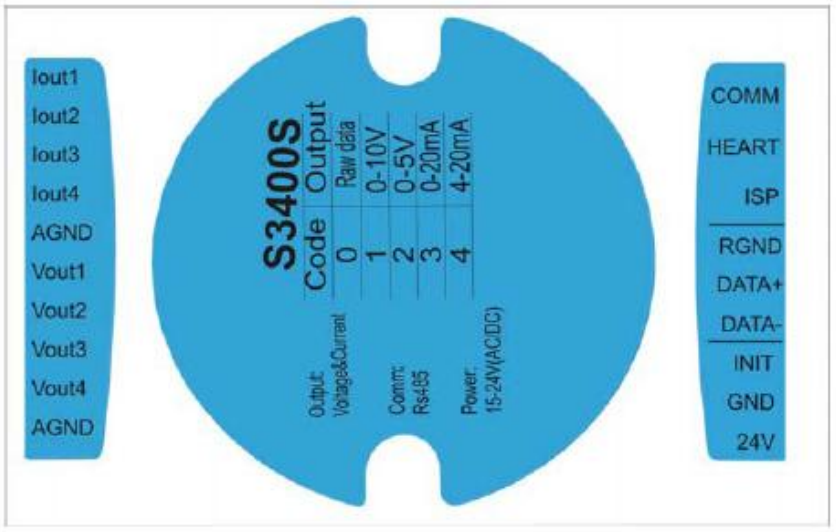
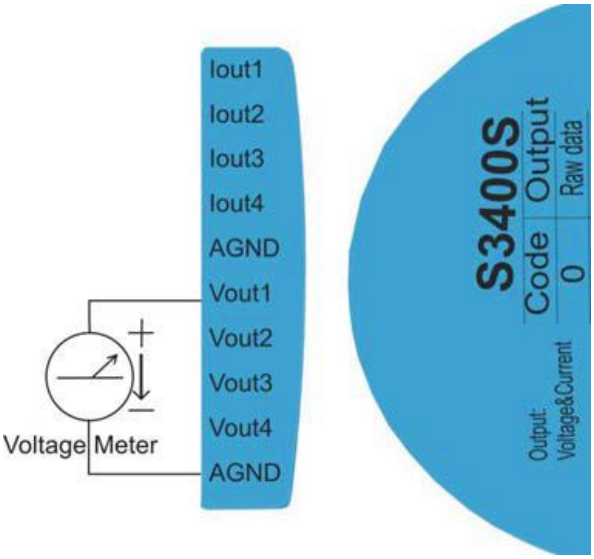


Figure 3 Terminal Definition

Calibration

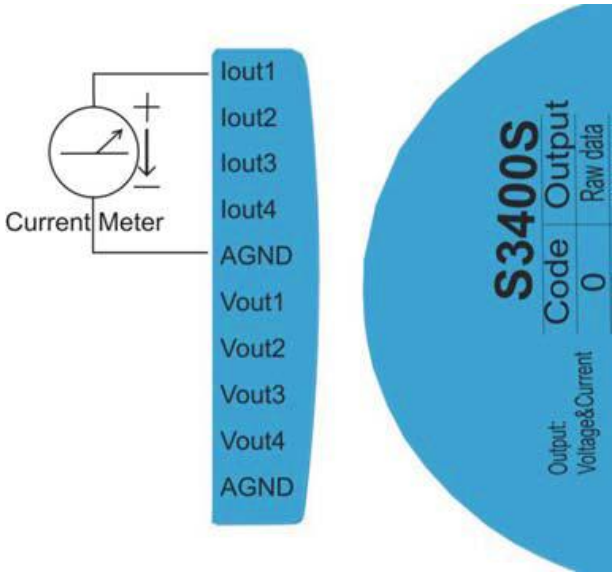
Voltage Calibration



1. Attach a voltage meter refer to the following picture
2. Power up the module and wait 30 minutes.
3. Write 1 to the corresponding range setting register
4. Write 0 to output register
5. Get the reading from multi-meter and write down to paper, RD1
6. Write 10000 to output register
7. Get the reading from multi-meter and write down to paper, RD2
8. Write RD1*1000 to the corresponding minimum calibration register
9. Write RD2*1000 to the corresponding maximum calibration register.

Calibration

Current Calibration



1. Attach a current meter refer to the following picture.
2. Power up the module and wait 30 minutes.
3. Write 3 to the corresponding range setting register
4. Write 0 to output register
5. Get the reading from multi-meter and write down to paper, RD1
6. Write 20000 to output register
7. Get the reading from multi-meter and write down to paper, RD2
8. Write RD1*1000 to the corresponding minimum calibration register
9. Write RD2*1000 to the corresponding maximum calibration register

Overview

Modbus protocol is a widely used and well-documented communications method. It provides a simple and effective means of programming our various products. The device communications follow a routine of queries issued by the master (your product) and our devices (the slaves). Modbus has various command types, we use only the **READ** , **WRITE** commands, type 3 and type 5 commands in the Modbus parlance. The master will prepare a packet comprised of the target address, command type (read or write), the starting address and number of bytes to be accessed, and finally a CRC for error detection. All devices are set to communicate over RS485 network using 1.2k through 115.2 k baud, N81 byte structure. Normal RS485 distance, termination and cabling rules apply. Physical layer is standard twisted pair + ground cabling. A shield is optional.

A typical Modbus packet looks like this:

Byte1 Device ID, the destination address for a particular message
Byte2 Function
Byte3 Starting address of the particular storage register(s) to be read or written, hi byte,
Byte4 Starting address low byte
Byte5 No. of registers to read/write (hi byte)
Byte6 No. of registers to read/write (low byte)
Byte7 CRC hi byte
Byte8 CRC low byte

Modbus Examples

READ Command (0x03):

This function is used to read the contents of multiple memory registers. The master to the Modbus must specify, the device ID, it's starting register and quantity of register desired. By convention if a data were to contain 2 byte, we would first send the Hi byte and then the Lo byte.

The master to the Modbus network will issue a read command: - Device ID=18
- Read 6 bytes of data
- Starting at register number 100 (64h)

Byte # Field Name (Hex) Data Description
Byte1 Slave Address 12 Modules with ID18 will be read
Byte2 Function 03 Read operation
Byte3 Starting Address Hi 00
Byte4 Starting Address Lo 64 Reading starting from register #100
Byte5 No. of Register to read Hi 00
Byte6 No. of Register to read Lo 03 Read a total of 3 registers
Byte7 Error Check (CRC) HI byte 46 The CRC is calculated using the CRC
Byte8 Error Check (CRC) LO byte B7 routine described below

The slave device with ID=11 will answer the master within a few milliseconds with the following response.
Byte # Field Name (Hex) Data Description

Byte1 Slave Address 12 Slave with ID18 is responding
Byte2 Function 03 we're responding to a read command
Byte3 Byte Count 06 6 bytes are coming
Byte4 Data1 Hi 00 byte1 of the data
Byte5 Data1 Lo 00 byte2 of the data
Byte6 Data2 Hi 00 byte3 of the data
Byte7 Data2 Lo 00 byte4 of the data
Byte8 Data3 Hi 00 byte5 of the data
Byte9 Data3 Lo 00 byte6 of the data
Byte10 Error Check (CRC) HI byte XX The CRC is calculated using the CRC
Byte11 Error Check (CRC) LO byte XX routine described below

WRITE command (0x06):
This function is used to write to a single memory register. The master of the Modbus must specify the device ID, its register address to be written and the data desired.
The master to the Modbus network will issue a write command: - Device ID=18
- Write to address 18
- Enter data 3 (03h)

Byte # Field Name (Hex) Data Description

Byte1 Slave Address 12 destination address
Byte2 Function 06 this is a write command
Byte3 Register Address Hi 00 address which will be written to, hi byte
Byte4 Register Address Lo 01 address which will be written to, low byte
Byte5 Data Hi 00 data that we are writing, hi byte
Byte6 Data Lo 03 data we are writing, low byte
Byte7 Error Check (CRC) HI byte XX The CRC is calculated using the CRC
Byte8 Error Check (CRC) LO byte XX routine described below

The slave device with ID=11 will answer the master within a few milliseconds with the following response.

Byte # Field Name (Hex) Data Description

Byte1 Slave Address 12 destination address
Byte2 Function 06 this is a write command
Byte3 Register Address Hi 00 address which will be written to, hi byte
Byte4 Register Address Lo 01 address which will be written to, low byte
Byte5 Data Hi 00 data that we are writing, hi byte
Byte6 Data Lo 03 data we are writing, low byte
Byte7 Error Check (CRC) HI byte XX The CRC is calculated using the CRC
Byte8 Error Check (CRC) LO byte XX routine described below

Table 1 : Slew rate table

Code	V/sec	mA/sec	Code	V/sec	mA/sec
0	Change instantly		6	2.0	4.0
1	0.0625	0.125	7	4.0	8.0
2	0.125	0.25	8	8.0	16.0
3	0.25	0.5	9	16.0	32.0
4	0.5	1.0	10	32.0	64.0
5	1.0	2.0	11	64.0	128.0

Memory Map of NetEyeS4300S

Modbus register list: **Note: * means default value**

<table>
<tr><th rowspan="2">Address</th><th rowspan="2">Bytes</th><th colspan="2">Value range</th><th rowspan="2">Description</th><th rowspan="2">Property</th></tr>
<tr><th>Min</th><th>Max</th></tr>
<tr><td>0-3</td><td>4</td><td>1</td><td>4294967295</td><td>Serial number,unique for each product</td><td>R</td></tr>
<tr><td>4-5</td><td>2</td><td>100</td><td>65535</td><td>Firmware version number</td><td>R</td></tr>
<tr><td>6</td><td>1</td><td>1</td><td>254</td><td>Device address</td><td>R/W</td></tr>
<tr><td>7</td><td>2</td><td>3499</td><td>3499</td><td>Product model</td><td>R</td></tr>
<tr><td>8</td><td>1</td><td>1</td><td>255</td><td>Hardware version</td><td>R</td></tr>
<tr><td>9</td><td>2</td><td>12</td><td>1152</td><td>Baudrate setting
<table><tr><th>Value</th><th>Buadrate</th></tr><tr><td>12</td><td>1200</td></tr><tr><td>24</td><td>2400</td></tr><tr><td>48</td><td>4800</td></tr><tr><td>96</td><td>9600</td></tr><tr><td>192*</td><td>19200</td></tr><tr><td>384</td><td>38400</td></tr><tr><td>576</td><td>57600</td></tr><tr><td>1152</td><td>115200</td></tr></table>For example:write 96 to register 9 to set the baudrate 9600.</td><td>R/W</td></tr>
<tr><td>10</td><td>1</td><td>4</td><td>4</td><td>Gain range setting.</td><td>R</td></tr>
<tr><td>11-99</td><td>-</td><td>-</td><td>-</td><td>Reserved</td><td>-</td></tr>
</table>

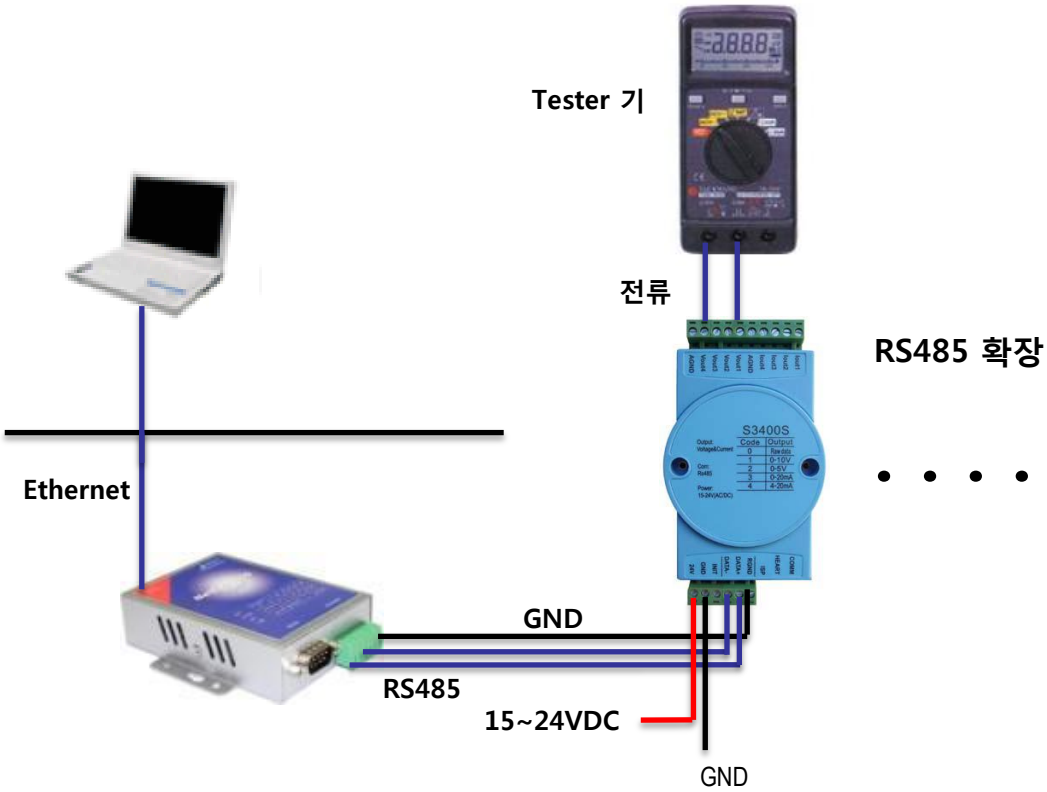
100-103	2	0	20000	Analog output, dependent on the corresponding range setting.Value for raw data is 0 through 4095,value for engineering units should mutiple 1000.For example,you want output 0.05V,should write 50 to the corresponding output register.	R/W
104-107	1	0	4	Output range setting.0 = raw data,1 = 0-10V,2 = 0 – 5V,3 = 0 – 20mA,4 = 4 – 20mA.	R/W
108-111	1	0	11	Slew rate setting.See table 1.	R/W
111-115	2	0	20000	Power up analog output,dependent on the corresponding range setting.	R/W
116	2	0	25000	The minimum reading from multimeter when calibrate analog output channel 1	R/W
117	2	0	25000	The maximum reading from multimeter when calibrate analog output channel 1	R/W
118	2	0	25000	The minimum reading from multimeter when calibrate analog output channel 2	R/W
119	2	0	25000	The minimum reading from multimeter when calibrate analog output channel 2	R/W
120	2	0	25000	The maximum reading from multimeter when calibrate analog output channel 3	R/W
121	2	0	25000	The maximum reading from multimeter when calibrate analog output channel 3	R/W
122	2	0	25000	The minimum reading from multimeter when calibrate analog output channel 4	R/W
123	2	0	25000	The maximum reading from multimeter when calibrate analog output channel 4	R/W
124	1	1	100	Respond delay for serial communication, the units is ms and default is 10ms	R/W

Test Procedures of NetEyeS4300S

1. NetEye1500의 IP address와 Port Number, Serial Parameter들을 설정한다.

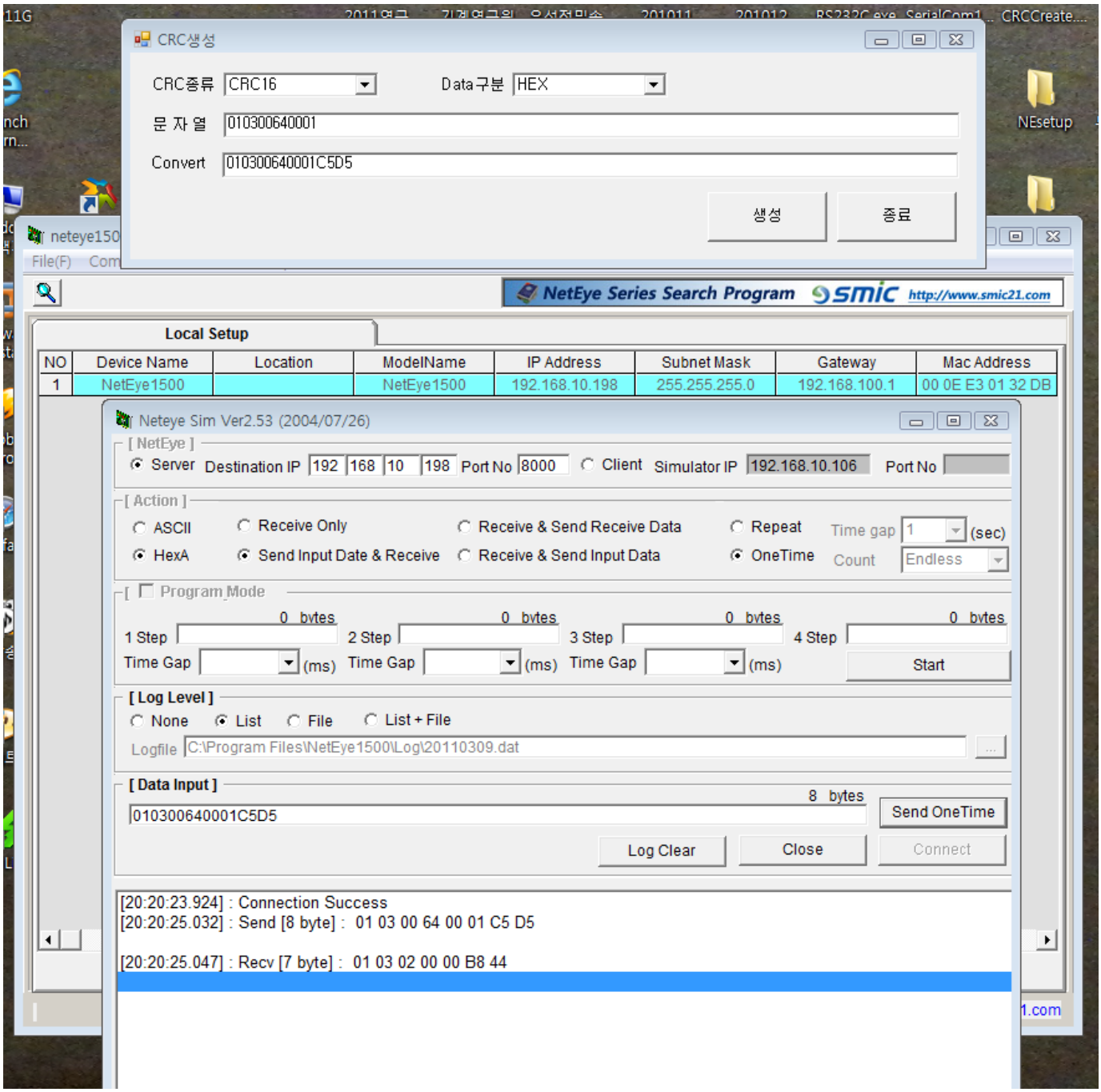
2. 현재 설정 값 : IP address – 192.168.10.198, Port Number – 8000
   Serial Parameters - RS485, 19200, N, 8, 1

* 각 Parameters들은 NetEye1500 Manual에 의하여 새로 설정 할 수 있다.

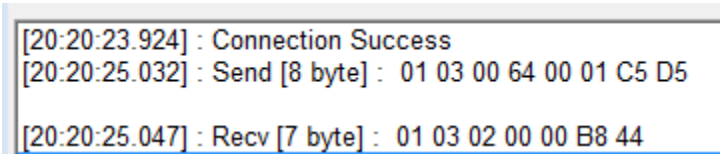


Configuration

3. CRCcreat.exe Program과 NetEye Setup Program의 Tool에 있는 NetEye Simulator를 실행 시킨다.



4. **CRCcreat.exe Program: CRC16 check sum data를 2 bytes 생성 시킨다.**
5. **NetEye Simulator: NetEye 1500을 통하여 NetEyeS4300S와 통신하기 위하여 modubus protocol에 의한 request data를 보낸다.**



6. 상기의 통신내용은 Device ID 01의 NetEyeS4300S device에서 Hexa 0x64 즉 100번째 register에서 Register 1개를 읽는 내용이다.
7. Device 01로 부터 Register 1개 2bytes를 00 00를 읽은 결과 이다.
8. 일반적으로 아래와 같은 통신으로 Device의 각종 Meory Map의 Parameter들을 변경, 조회 할 수 있다.

Device ID 확인 방법

1) PC-> NE4300s

ADDR	내용	Value(HEX)	비고
0	고정	ff	
1	Read/Write	03	Read
2	ADDR-H	00	NES4300S의 ADDR의 Register
3	ADDR-L	06	
4	Data Count -H	00	Reading Register 의 갯수
5	Data Count -L	01	
6	CRC-1	71	계산식
7	CRC-2	D5	계산식

2) PC <- NE4300s 응답

ADDR	내용	Value(HEX)	비고
0	고정	ff	
1	Read/Write	03	Read
2	Count	02	Data Count (ADDR9~10)
3	고정	00	
4	Device Address	초기치 : FE(254)	NES4300S의 ADDR
5	CRC-1	0	계산식
6	CRC-2	0	계산식

현재는 01로 설정되어 있음

Device ID 변경 방법

1) PC-> NES4300S

ADDR	내용	Value(HEX)	비고
0	ID ADDR	FE	현재 NES4300S의 ID ADDR
1	Read/Write	06	Write
2	ADDR-H	00	NE3302의 ADDR 의 Register
3	ADDR-L	06	
4	변경 ADDR -H	00	
5	변경 ADDR -L	01~FE	NES4300S의 변경할 ID ADDR
6	CRC-1	BC	계산식
7	CRC-2	04	계산식

현재는 01로 설정되어 있음

현재는 01로 설정되어 있음

```
[20:42:22.748] : Send [8 byte] :  01 06 00 06 00 02 E8 A

[20:42:22.763] : Recv [8 byte] :  01 06 00 06 00 02 E8 A
```

01 device ID을 02 바꾼다.

```
[20:45:22.881] : Send [8 byte] :  FF 03 00 06 00 01 71 D5

[20:45:22.896] : Recv [7 byte] :  FF 03 02 00 02 10 51
```

device ID가 02로 바뀐 것을 확인한다..

```
[20:47:20.317] : Connection Success
[20:47:20.988] : Send [8 byte] :  02 03 00 64 00 01 C5 E6

[20:47:21.003] : Recv [7 byte] :  02 03 02 00 00 FC 44
```

device ID 02 의 100번째 Register를 읽는다.
Default 00 00

```
[20:49:52.823] : Connection Success
[20:49:53.930] : Send [8 byte] :  02 06 00 06 00 01 A8 38

[20:49:53.961] : Recv [8 byte] :  02 06 00 06 00 01 A8 38
```

device ID 02 를 다시 01로 바꾼다.

```
[20:53:52.837] : Connection Success
[20:53:53.680] : Send [8 byte] :  01 03 00 64 00 01 C5 D5

[20:53:53.695] : Recv [7 byte] :  01 03 02 00 00 B8 44
```

device ID 01의 100번 째 register를 다시 읽는다.

Lout1 1st channel의 Analog Output 4~20mA내에서 변경하는 법

1. Memory Map의 104~107 4개의 register (4개의 channel에 할당 됨)에 각각 channel에 대하여 0~10V (1), 0~5V(2), 0~20mA(3), 4~20mA(4) 로 설정한다.

[21:23:43.519] : Send [8 byte] : 01 03 00 68 00 04 C5 D5

[21:23:43.534] : Recv [13 byte] : 01 03 08 00 01 00 02 00 03 00 04 D 14

4개 channel의 설정 상황을 조회한다.
결과 ch1 0~10V, ch2 0~5V, ch3 0~20mA, ch4=4~20mA

[22:09:13.084] : Send [8 byte] : 01 06 00 68 00 04 09 D5

[22:09:13.099] : Recv [8 byte] : 01 06 00 68 00 04 09 D5

4개 channel중 ch1의 설정 4~20mA로 바꾼다.

[21:29:00.885] : Send [8 byte] : 01 03 00 68 00 04 C5 D5

[21:29:00.901] : Recv [13 byte] : 01 03 08 00 04 00 02 00 03 00 04 58 14

4개 channel중 ch1의 설정 4~20mA로 바뀐걸 확인한다..

[21:31:02.830] : Send [8 byte] : 01 06 00 69 00 04 58 15

[21:31:02.971] : Recv [8 byte] : 01 06 00 69 00 04 58 15

4개 channel중 ch2의 설정 4~20mA로 바꾼다..

[21:32:29.816] : Send [8 byte] : 01 03 00 68 00 04 C5 D5

[21:32:29.831] : Recv [13 byte] : 01 03 08 00 04 00 04 00 03 00 04 D0 14

4개 channel중 ch2의 설정 4~20mA로 바뀐걸 확인한다.

[21:33:42.652] : Send [8 byte] : 01 06 00 6A 00 04 A8 15

[21:33:42.730] : Recv [8 byte] : 01 06 00 6A 00 04 A8 15

4개 channel중 ch3 0x6A의 설정 4~20mA로 바꾼다.

[21:35:39.590] : Send [8 byte] : 01 03 00 68 00 04 C5 D5

[21:35:39.605] : Recv [13 byte] : 01 03 08 00 04 00 04 00 04 00 04 61 D5

4개 channel 모두가 4~20mA로 바뀐걸 확인한다.

2. **100~103 register 에 0~20000까지의 값을 설정하여 Analog Output의 value 를 정한다.**

[21:59:39.971] : Send [8 byte] : 01 06 00 64 27 10 D2 29

[21:59:40.002] : Recv [8 byte] : 01 06 00 64 27 10 D2 29

1번 channel 10mA test 기 바늘 확인

[22:01:48.312] : Send [8 byte] : 01 06 00 64 4E 20 FC 6D

[22:01:48.328] : Recv [8 byte] : 01 06 00 64 4E 20 FC 6D

1번 channel 20mA 로 변경되는 걸 test 기 바늘 확인

3. **5000= 0x1388, 10,000= 0x2710 20,000= 0x4E20, 25,000= 0x61A8, 20,950= 0x51D6**

100= 0x64, 101= 0x65, 102= 0x66, 103= 0x67,

104= 0x68, 105= 0x69, 106= 0x6A, 107= 0x6B