

DG21E5A

✓ KC인증

R-R-Diu-DGE5A

상호명: 일품 주식회사, 제조자: 일품 주식회사, 제조국: 한국

모델명: DG21E5A

✓ 사용 환경

정상 동작 온도 범위 = -25 ~ 70 [° C]

이슬이 맺히지 않을 것, 먼지가 없을 것.

✓ 전원

정격 전압 = DC 24 [V] (동작 가능 범위 20 ~ 27 [V])

최대 소모 전류 = 300 [mA]

✓ 통신

물리 규격: TIA/EIA-485A (RS485)

선로상 최대 장치 수 = 64 node

ESD 보호 = 15 [kV]까지

데이터 프로토콜: MODBUS RTU protocol

✓ DI 단자의 정격

외부 전원 없이 접점만 연결 가능

외부 DO와 직결 (DI 감지용 전원을 장치 내부에 포함)

✓ DI 감지 표시

ON 감지시 LED 점등

✓ DO 단자의 정격

Transistor 출력: Sink type

전류: 0~0.5 [A], 전압: 0~50 [V]

✓ DO 작동 표시

설정 OFF: LED OFF, 접점 개방

설정 ON: LED ON, 접점 단락

✓ 격리(Isolation)

포토커플러. 최대 격리 전압 = 3.75 [kV rms]

격리군1: RS485 단자

격리군2: 전원 단자, 모든 입출력(DI, DO) 단자

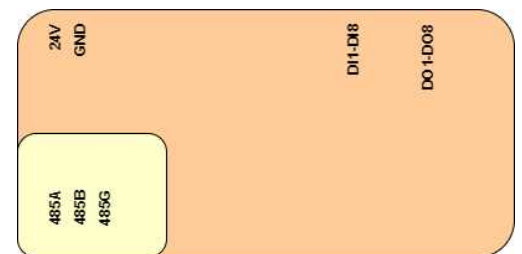
✓ 외형 치수

가로 145 [mm], 세로 90 [mm], 높이 41 [mm]

✓ 고정 방식

DIN rail에 장착 가능

나사 4개로 고정 가능 (가로 135 [mm], 세로 70 [mm])



[그림 1] E5A의 내부 격리

1. 개요

DG21E5A는 DiBA PLC(Programmable Logic Controller)의 CPU(Central Processing Unit) 모듈입니다. 모델명 DG21E5A의 제품명은 CPU type 2입니다. 모델명은 일품표식(DG)과 출시연도, 대표모델명(E5A)으로 구성됩니다.

사용자는 DST(Diurnal Standard Text) 파일을 이용하여 E5A가 처리할 작업을 지시할 수 있습니다. DST 파일은 PLC의 프로그래밍 언어 표준(IEC 61131-3) 중 ST(Standard Text)를 기반으로 설계된 일품의 PLC 프로그래밍 언어로 작성합니다. DST 파일의 형식은 Windows™ text 문서이므로 “메모장” 등의 기본 문서편집기로 생성 및 수정할 수 있습니다. DST 프로그래밍 방법은 DST 문법 설명서를 참고하십시오.

E5aLoader는 작성이 완료된 DST 파일을 읽어 들어서 E5A에 저장(Run)하거나, E5A에서 시험(Debug)할 수 있습니다. E5A에 저장된 작업은 전원이 꺼져도 유지되고, 다시 전원이 공급되면 바로 실행됩니다. 하지만 E5A가 작업 실행 도중 복구할 수 없는 문제를 만나면 30초 후 재기동하며, 시스템의 오작동 방지를 위해 저장된 작업을 사용 중지합니다. E5A를 시험하는 경우, 작업 내용은 E5A의 전원이 꺼지기 전까지만 유효합니다.

E5A는 PLC의 일반적인 기능을 수행할 수 있으며, RS485(2 wire) 1개와 WiFi(802.11 b/g/n) 1개를 가지고 있습니다. RS485로는 MODBUS RTU master/slave, user defined 통신을 수행할 수 있습니다. WiFi는 AP(Access Point, 802.11 b/g) 혹은 station으로 역할을 설정할 수 있고, 다양한 보안 프로토콜이 적용되어 안전하게 운용될 수 있습니다. WiFi로는 TCP/IP 상에서 MODBUS TCP server/client, user defined server/client 통신을 수행할 수 있습니다.

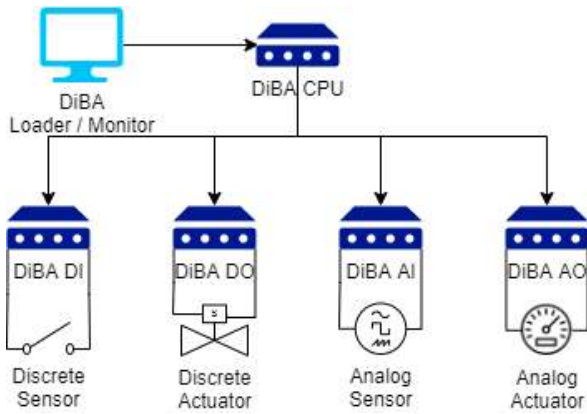
E5A는 8개의 DI(Digital Input, 측정용 전원 자체 공급)와 8개의 DO(Digital Output, 트랜지스터 sink형 ~0.5A)를 가지고 있습니다. 별도의 I/O 모듈 없이도 소수의 장치를 제어할 수 있습니다.

(주의) E5A는 시스템 및 네트워크를 운영하고 관리하는 역할을 수행할 수 있습니다. 자동제어 시스템의 안전하고 안정적인 운용을 위해 사용자는 E5A를 실제 자동제어 시스템에 적용하기 전부터 안전에 대한 고려를 반영해야 하고, 충분한 시험을 수행해야 합니다.

2. DiBA 시스템

2.1. DiBA 시스템의 기본 구성

DiBA(다이바)는 일품의 PLC 제품군에 대한 브랜드명입니다. PLC가 적용된 자동제어 시스템의 형태는 각기 다르지만 시스템을 구축하기 위한 기본 구조는 거의 비슷합니다.








[그림2] 자동제어시스템의 구성

[그림2]는 사용자가 일상에서 마주치는 DI(Digital Input: 만수 센서, 문 열림 센서 등과 같이 특정 조건에 대해 일치/불일치만을 감지하여 전기적 신호로 변환하는 장치)나 DO(Digital Output: 환풍기, 조명 등과 같이 전기적 신호로 ON/OFF하여 제어하는 장치), AI(Analog Input: 온도계, 영도계 등과 같이 조건 범위에 따라 전압/전류/저항의 크기로 변환하는 장치), AO(Analog Output: 유량 조절 밸브, DC 모터 등과 같이 전압/전류의 크기로 제어하는 장치) 등을 간략하게 표현합니다. 또한, DiBA CPU 모듈과 DiBA I/O(Input/Output) 모듈 사이에 서로 정보를 주고받을 수 있다는 것을 표현합니다.

컴퓨터에는 E5aLoader, SCADA(Supervisory Control And Data Acquisition) 등의 application을 설치하여 사용합니다.

일품의 자동제어 제품군(DiBA-21)은 다음의 제품들로 구성됩니다.

형태	모델명	제품명	설명
	DG21E5A	CPU type 2	DiBA CPU. WiFi 1개, RS485 1개, DI 8개, DO 8개 저장된 작업 내역에 따라 시스템을 운용
	E5aLoader	CPU loader	Windows™ PC용 application CPU 모듈에게 작업을 할당
	DG21E4H	MODBUS RTU DO	RS485 1개, DO 40개 CPU 혹은 SCADA가 설치된 PC의 제어를 받음.
	DG21E4I	MODBUS RTU DI	RS485 1개, DI 40개 CPU 혹은 SCADA가 설치된 PC의 제어를 받음.
	DG21E4J	MODBUS RTU IO	RS485 1개, DI 8개, DO 8개, AI 8개, AO 8개, UI 8개 CPU 혹은 SCADA가 설치된 PC의 제어를 받음.

Hardware의 형태를 갖는 제품의 모델명은 [일품표식] + [출시연도] + [대표모델명]의 조합입니다. 그러므로 CPU type 2를 일반적으로 지칭하는 경우에는 E5A라고 표기합니다. Hardware 형태의 제품은 모두 동일한 케이스를 사용하며, 치수는 145 x 90 x 41[mm](가로 x 세로 x 높이)입니다.

2.2. E5A의 확장성

[그림2]와 같은 피라미드 구조는 간단한 체계를 갖는 시스템에 적용하기에 편리하고 효율적입니다. 또한, SCADA를 이용하여 시스템을 운용하는 경우, 복수의 CPU를 SCADA의 아래에 두면 조금 더 복잡한 시스템에도 효율적으로 적용할 수 있습니다. 하지만 위험 상황에 대비한 안정성과 불규칙한 방향의 확장성을 모두 고려하면 분산 시스템을 구성하여 제어권을 적절하게 배분하는 것이 좋습니다.

위험 상황은 정전, PC의 수명, 시스템 자원의 병목 등에 의해 발생할 수 있습니다. PC는 정전으로 재기동하면, 정상상태로 돌아와서 시스템에 적절한 제어를 하기까지 소요되는 시간이 PLC CPU에 비해 많습니다. 이 지연은 시스템 관리의 공백 기간이 됩니다. 그리고 PC는 PLC CPU에 비해 많은 자원이 필요하므로 일부 자원의 수명이 PC 전체의 가동 시간을 제한하며, 문제가 발생한 자원을 수리할 때까지 PC를 사용하지 못합니다. 또한 피라미드 구조는 시스템의 최상단으로 data가 몰리는 특성이 있습니다. 시스템의 크기에 따라 data 수집 계획을 최적화하지 못하면, 통신망의 병목 현상으로 시스템 운용이 지장을 받을 수 있습니다.

시스템의 확장 방향은 미리 예측하기 어렵습니다. 상황의 변화에 따라 유연하고 빠르게 대응할 수 있도록 준비하는 것이 확장계획을 탄탄하게 세우는 것보다 좋은 선택입니다. 이를 위해 PLC 시스템의 확장성이 얼마나 뛰어난지를 검토하여 제품을 선정해야 합니다.

E5A는 일반적인 PLC CPU와 마찬가지로 전원 및 수명 위험에 대한 안정성이 PC에 비해 월등하게 높습니다. 위험 상황을 대비하려면 시스템 최상위 제어기는 PLC CPU가 되어야 합니다. 또한 간편하게 분산 시스템을 구성할 수 있도록 E5A는 지원하는 모든 통신 프로토콜에 대해 제어자(CPU)와 피제어자(I/O)의 역할을 모두 수행할 수 있습니다. RS485 interface를 통해, E5A는 하위의 I/O 모듈들을 제어할 수도 있고 상위 CPU 혹은 SCADA의 제어를 받을 수도 있습니다. WiFi로는 server interface와 client interface를 동시에 운용할 수 있습니다. 즉, server interface를 통해 다른 CPU 혹은 SCADA의 제어를 받을 수 있고, client interface를 통해 하위의 I/O 모듈들을 제어할 수도 있습니다. 그러므로 Interface의 조합을 이용하여 E5A 상호간 다양한 형태의 연동이 가능하며, 분산 시스템의 범위를 무한하게 확장할 수 있습니다.

3. E5A 사용 준비

3.1. 전원 및 신호선 연결

E5A의 모든 기능이 정상 동작하는 전원 전압의 범위는 20 ~ 29[Vdc]입니다. 전원에 30[Vdc] 이상의 전압을 인가하면, 부품이 파손될 수 있습니다. 24[Vdc] SMPS(Switching Mode Power Supply) 사용을 권장합니다.

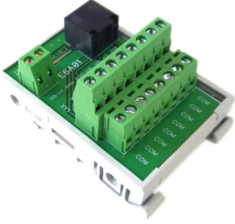

대부분의 자동제어 시스템은 전원을 차단하는 스위치를 구성에 포함합니다. 이는 전선 작업 과정에서 감전이나 시스템의 오작동에 의한 피해를 막기 위해 필요합니다. 작업자는 전선을 연결/분리하는 동안 시스템의 전원을 차단해야 합니다.

RS485는 3개의 전선으로 연결합니다. 485A와 485B는 장치에 따라 +와 -로 표기되기도 합니다. RS485는 multi-drop 방식(하나의 선로에 여러 장치가 연속해서 연결되는 통신 선로 연결 방식)을 사용하므로 장치들은 선로를 따라 일렬로 배치되어야 합니다. 선로에 가지를 쳐서 여러 방향으로 뻗어 나가면 통신 오류가 발생하게 됩니다.

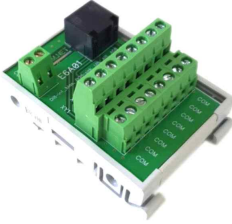

E5A는 터미널보드 E6A01(직결형 8 단자), E6B01(Relay형 8 단자, 최대 10A, 최대 30Vdc or 250Vac), E6E01(개별 격리형 8단자, 외부전원전압 10~30Vdc, 극성무관)과 연결할 수 있습니다.

<p>[그림3] 구성 및 연결 방법</p>	<p>[그림4] 연결 예시 사진</p>
-------------------------	-----------------------

DI는 일반적인 DO와 짝을 이뤄서 ON/OFF 상태를 읽습니다. 선로를 운영하는 전원은 DI가 공급하므로 외부 전원은 공급되지 않아야 합니다. DI 단자가 RJ45를 사용하므로 일반적인 외부 장치와는 터미널보드(RJ45는 direct cable 사용)를 통해 연결합니다. 터미널보드를 분리함으로써 단자의 수를 늘리고, DI 단자의 다양한 출력 특성에 맞는 터미널보드를 선택할 수 있습니다. DI를 E4H(디지털 출력 모듈)의 DO 단자나 E4J(통합 입출력 모듈)의 DO 단자와 바로 연결한다면, 터미널보드 없이 RJ45 direct cable만으로 연결하여 사용할 수 있습니다. 이 인터페이스는 특허 제10-2214702호로 보호받고 있습니다. DI에 적용할 수 있는 터미널보드는 E6A01, E6E01이 있습니다.

터미널보드	E6A01	E6E01
사진		
특성	직결형 8단자	개별 격리형 8단자, 외부전원전압 10~30Vdc, 극성무관

DO는 ON/OFF를 출력합니다. DO1이 OFF일 때 DO1과 GND 사이의 회로가 개방되고, ON일 때 DO1과 GND 사이의 회로가 단락됩니다. 최대 전류는 0.5[A]입니다. 전압은 DC만 가능하고, 최대 50[V]입니다. DO 단자가 RJ45를 사용하므로 일반적인 외부 장치와는 터미널보드를 통해 연결합니다. 터미널보드를 분리함으로써 단자의 수를 늘리고, DO 단자의 다양한 출력 특성에 맞는 터미널보드를 선택할 수 있습니다. 이 인터페이스는 특허 제10-2214702호로 보호받고 있습니다. DO를 E4I(디지털 입력 모듈)의 DI 단자나 E4J(통합 입출력 모듈)의 DI 혹은 UI 단자와 바로 연결한다면, 터미널보드 없이 RJ45 direct cable만으로 연결하여 사용할 수 있습니다. DO에 적용할 수 있는 터미널보드는 E6A01, E6B01이 있습니다.

터미널보드	E6A01	E6B01
사진		
특성	직결형 8단자	Relay형 8단자, 최대 10A, 최대 30Vdc or 250Vac

E5A의 상태표시 LED는 2개의 그룹 LED와 8개의 단자 LED로 구성되어, 단자들의 상태를 RJ45 모듈러잭 단위로 묶어서 표시합니다. +0~+7의 8개의 단자 LED는 그룹을 나타내는 2개의 LED 중 점등되는 그룹의 단자 상태를 나타냅니다. 즉, DI1 LED가 켜지면 +0~+7의 LED는 DI1~DI8 단자의 상태를 나타내고, DO1 LED가 켜지면 +0~+7의 LED는 DO1~DO8 단자의 상태를 나타냅니다. 그룹 LED가 2초씩 순서대로 켜져서 RJ45 모듈러잭 단위로 소속된 단자들(8개)의 상태를 각각 2초간 표시하여, 16개의 모든 단자의 상태를 4초간 돌아가며 보입니다.

단자 LED / 그룹 LED	+0	+1	+2	+3	+4	+5	+6	+7
DI1	DI1	DI2	DI3	DI4	DI5	DI6	DI7	DI8
DO1	DO1	DO2	DO3	DO4	DO5	DO6	DO7	DO8

3.2. 설정 초기화

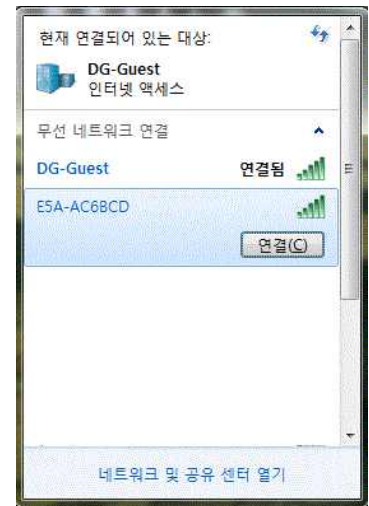
E5A는 암호를 알아야 WiFi 관련 설정을 변경할 수 있고 DST loading도 가능합니다. 사용자가 암호를 잊거나 실수로 잘못 설정한 경우, E5A의 INIT 버튼을 3초 동안 누르면 E5A는 저장된 작업을 지우고 WiFi 설정 및 암호를 초기값으로 변경합니다. 다음은 E5A의 설정 초기값입니다.

WiFi의 모드	AP
AP SSID	E5A-XXXXXX (X부분은 E5A마다 다름)
AP Password	ilpum.net
Domain명, IP Address (PC의 web browser에서 E5A로 접근하기 위한 주소)	e5a.net, 10.123.45.1
사용자 이름 (Login Name)	Admin
사용자 암호 (Login Password)	ilpum.net
등록된 network 정보	-
저장된 작업 (DST)	-

3.3. E5aLoader 연결

PC에서 E5aLoader를 실행하여 새로운 작업을 하기 위해서는 PC에서 Ethernet을 통해 E5A로 연결이 가능해야 합니다. E5A가 AP 모드로 설정되어 있다면, PC를 E5A의 무선 네트워크에 참여하도록 네트워크를 지정하여 연결합니다.

[그림5]는 Windows™에서 네트워크를 연결하는 모습입니다. “E5A-AC6BCD”라는 SSID를 가진 E5A로 연결합니다. 처음 연결할 때는 암호를 입력해야 합니다. “ilpum.net”을 입력합니다.



[그림5] AP 모드인 E5A에 연결

```

CONFIGURATION nameOfConf
TYPE T_UDF: (* 사용자 정의 변수형 선언 *)
STRUCT (* 구조체로 지정 *)
    length : BYTE;
    buffer : STRING(100);
END_STRUCT
END_TYPE
VAR_GLOBAL
    
```

```

gUsrSend : T_UDF;
gCount : SINT;
END_VAR

RESOURCE extLine1 ON SER_1 (* SER_1은 RS485의 interface명 *)
VAR_GLOBAL
  gConf : CONF_SER1;
  gCommUsr : COMM_SERUSR;
END_VAR

(* bootup시 최초 1회만 실행됨. priority는 1이 가장 높음. *)
TASK taskInit (SINGLE := TRUE, PRIORITY := 1);
(* 매 1초마다 실행 *)
TASK taskSync (INTERVAL := t#1s, PRIORITY := 2);
PROGRAM pgm1Init WITH taskInit : Prog1Init();
PROGRAM WITH taskSync : Prog1Sync();
END_RESOURCE
END_CONFIGURATION

PROGRAM Prog1Init
  (* 통신 설정: SET, user defined mode, 9600/N/8/1 *)
  extLine1.gConf.DATABITS := 8;
  extLine1.gConf.STOPBIT := 1;
  extLine1.gConf(GET_SET := 1, MODE := 3, RATE := 9600, PARITY := 0);
  extLine1.gCommUsr.EC := 0; (* 통신 결과 초기화 *)

  (* 작업 초기화 *)
  gCount := 0; (* 0:IP, 1:SSID, 2:mode, 3:domain_name *)
END_PROGRAM

PROGRAM Prog1Sync
  (* 1초마다 네트워크의 정보를 하나씩 ser1/user-defined로 출력. *)
  (* 통신 요청의 처리가 끝날 때까지 기다림 *)
  IF extLine1.gCommUsr.EC = 1 THEN
    RETURN;
  END_IF;

  CASE gCount OF
    0: gUsrSend.buffer := CONCAT(SEE_NW_IP(), ', ');
    1: gUsrSend.buffer := CONCAT(SEE_NW_SSID(), ', ');
    2: gUsrSend.buffer := CONCAT(SEE_NW_MODE(), ', ');
    3: gUsrSend.buffer := CONCAT(SEE_NW_DOMAIN(), '$1$r');
  ELSE gUsrSend.buffer := 'Undefined$1$r';

```



```

END_CASE;

(* 주기적 작업을 위해 카운터 사용 *)
gCount := gCount + 1;

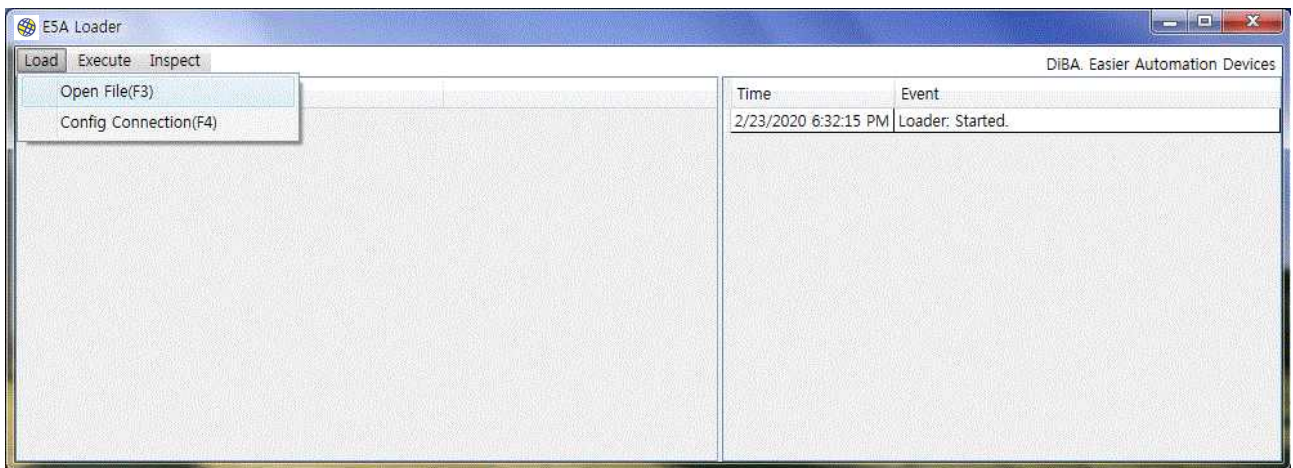
(* 4개마다 순환 *)
IF gCount >= 4 THEN
  gCount := 0;
END_IF;

(* 통신 요청: SEND *)
extLine1.gCommUsr.SEND_LEN := LEN(gUsrSend.buffer);
extLine1.gCommUsr(RECV_SEND := 1, SEND_ADDR := ADDR_OF(gUsrSend));
END_PROGRAM

```

[DST 1] see_nw.dst

[DST 1]은 E5A가 현재 사용하는 IP address, SSID, 모드, domain명을 RS485 interface를 통해 차례로 출력하는 작업을 작성한 것입니다. 이 내용을 그대로 복사하여 일반 text 편집기로 붙여넣고 이름을 지정하여 “dst”라는 확장자로 저장합니다. (편의를 위해 “see_nw.dst”로 부릅니다.)



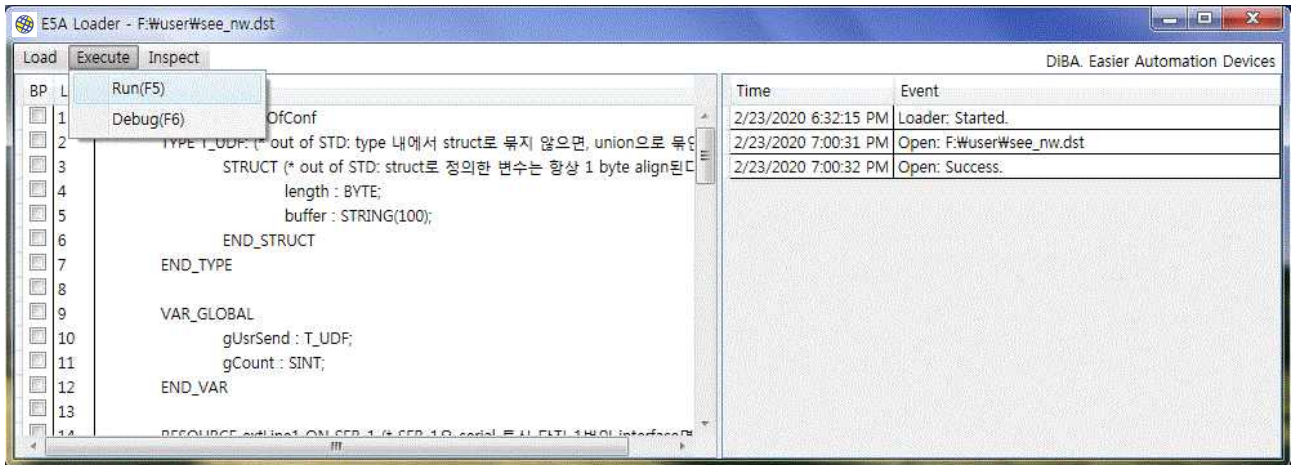
[그림6] E5aLoader의 Load 메뉴

[그림6]을 참고하여 E5aLoader를 실행하고, “Open File(F3)”로 “see_nw.dst” 파일을 엽니다.



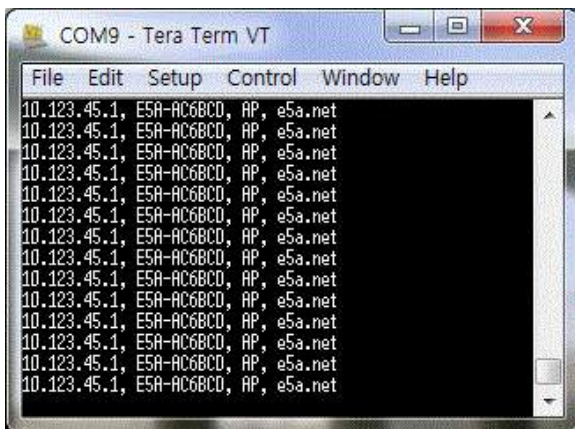
[그림7] E5A의 연결 정보 입력

E5A가 AP 모드일 때는 “e5a.net”이라는 domain명으로 연결할 수 있습니다. 필요한 곳을 모두 수정하고, Enter를 입력합니다. (PW에서 “Enter” key를 누르면 변경 내용이 저장되면서 다이얼로그창이 닫힙니다.)



[그림8] 작업의 저장 및 실행

“Run(F5)”을 눌러서 E5A에 작업을 저장하여 실행시킵니다.

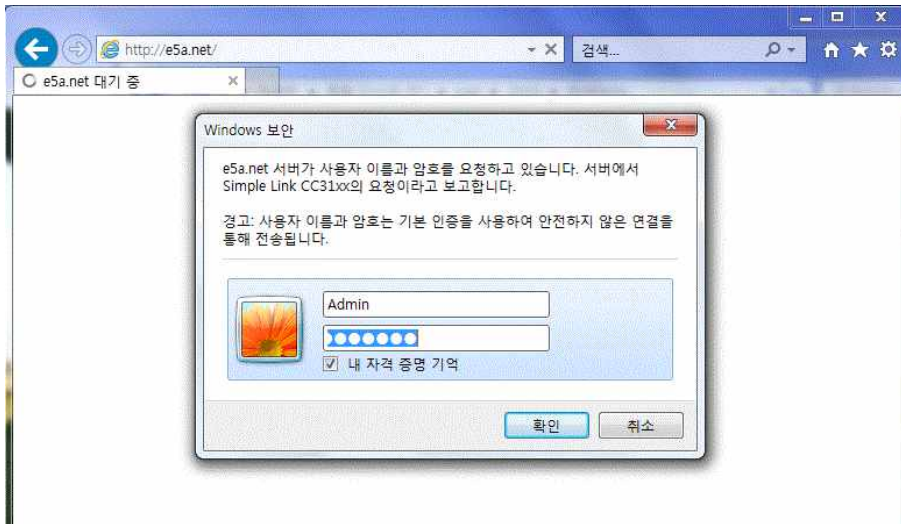


[그림9] RS485 출력 내용

E5A의 RS485 단자와 PC의 RS485 interface를 연결하고, Tera Term™을 실행합니다. PC에 RS485 interface가 기본으로 장착된 제품은 거의 없습니다. RS485 interface 모듈은 별도로 구매해야 합니다. Tera Term™은 Open Source Project로 만들어진 무료 터미널 프로그램입니다. Windows™용으로 download하여 사용하면 됩니다. Tera Term™의 “Setup” 중 “Serial port...”를 선택하여 통신 설정을 맞춰야 합니다. [그림9]에서는 Port: COM9, Baud rate: 9600, Data: 8 bit, Parity: none, Stop: 1 bit, Flow control: none, Transmit delay: 0 & 0으로 설정했습니다.

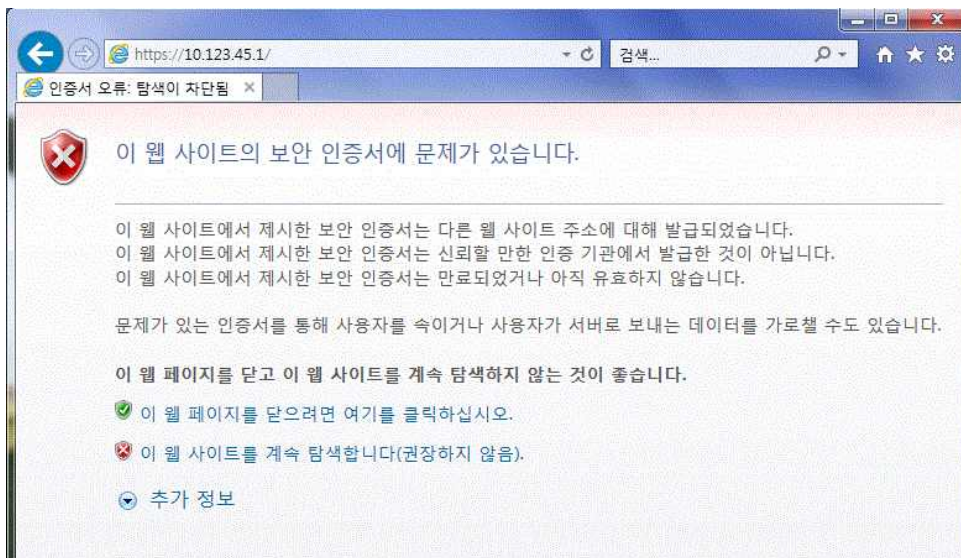
3.4. WiFi 설정 변경

E5A는 기존에 구축된 network에 참여하여 다른 장치들과 연동할 수 있습니다. 이를 위해 WiFi 모드를 station으로 변경합니다.



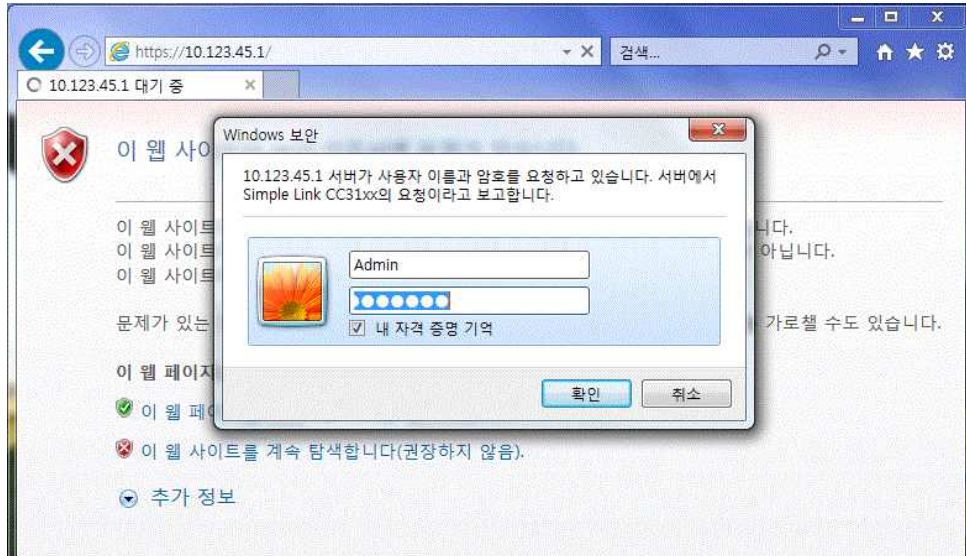
[그림10] Web Browser로 E5A에 접속

주소창에 도메인 이름을 “e5a.net”라고 입력합니다. 사용자 이름과 암호를 묻습니다. “Admin”과 “ilpum.net”을 입력합니다.



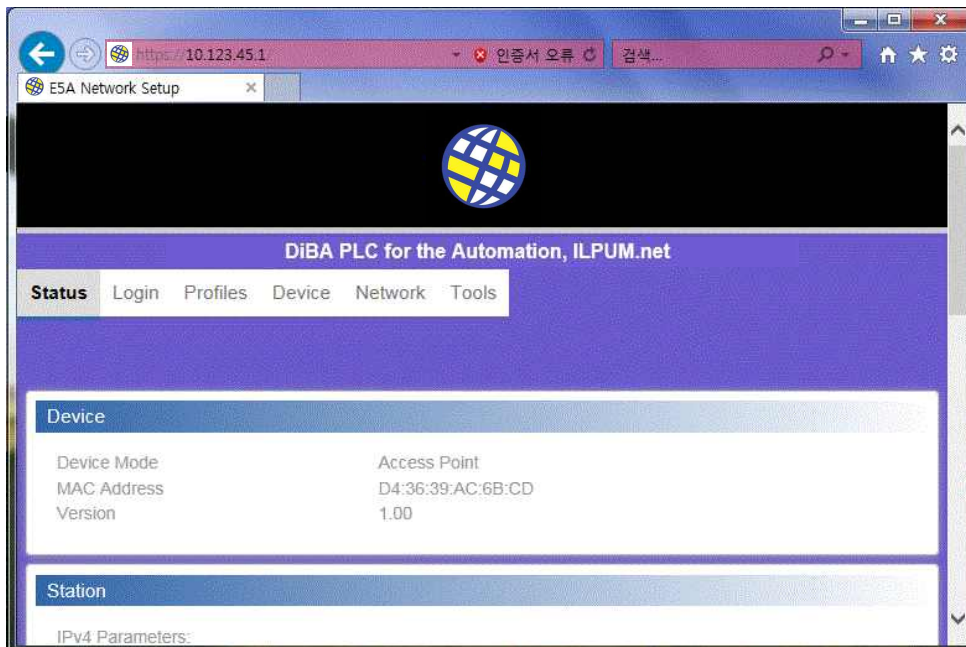
[그림11] 인증서 문제 알림

Web browser가 인증서에 문제가 있다고 알려줍니다. E5A가 사용하는 인증서는 인증기관을 통해 인증을 받은 것이 아니기 때문에 정상적인 결과입니다. 이 사실 인증서는 E5A의 네트워크 보안을 유지하기 위해 반드시 필요합니다. 보고를 무시하고 계속 진행합니다. “이 웹 사이트를 계속 탐색합니다(권장하지 않음).”을 누릅니다. 인증서 오류를 보고하는 창과 내용은 web browser에 따라 다를 수 있습니다.



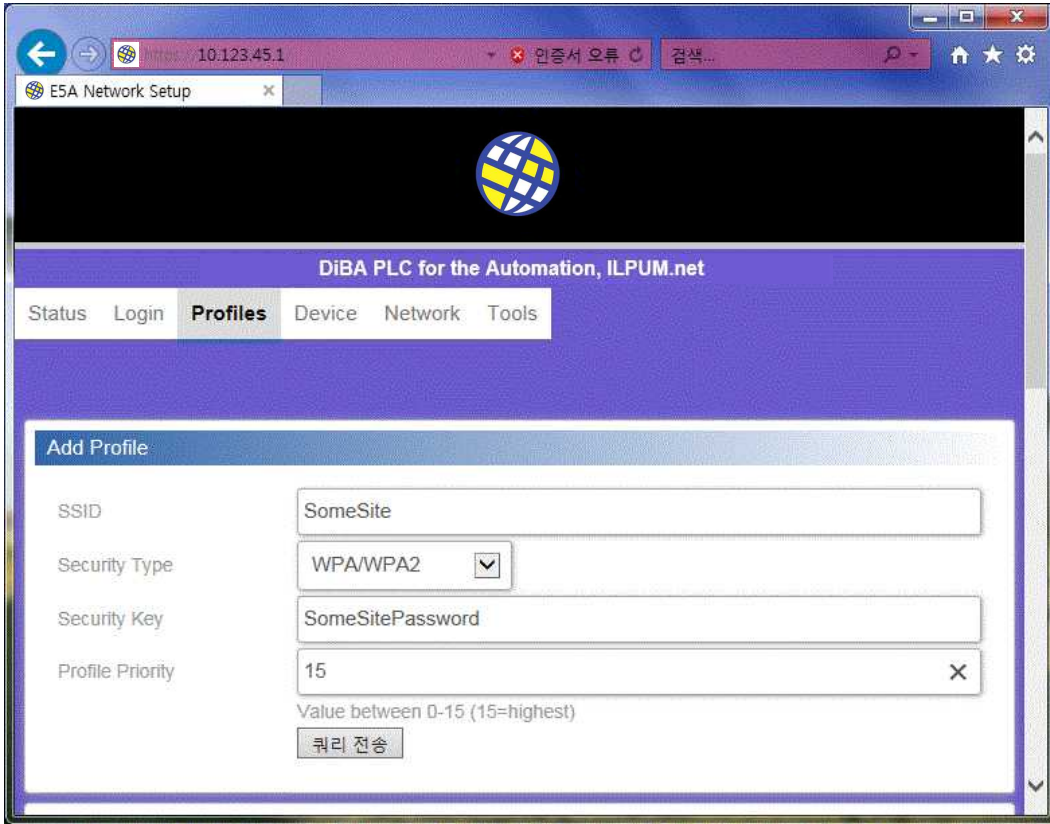
[그림12] 인증서 문제를 무시하고 진행

다시 사용자 이름과 암호를 묻습니다. 앞과 동일하게 입력합니다.



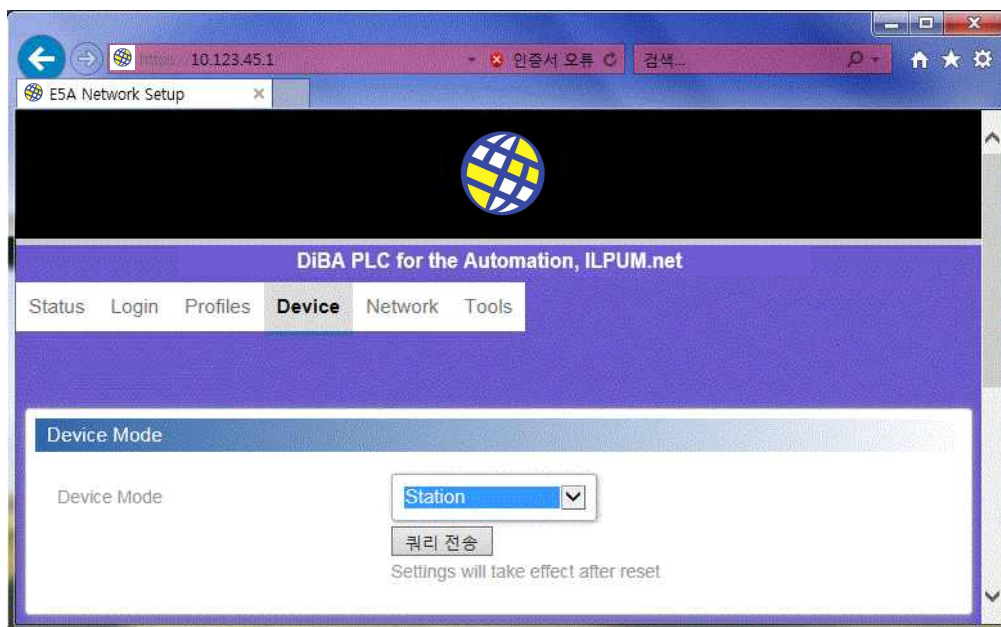
[그림13] E5A 네트워크 설정 첫 화면

첫 화면에 들어왔습니다. AP 모드로 설정된 것을 볼 수 있습니다.



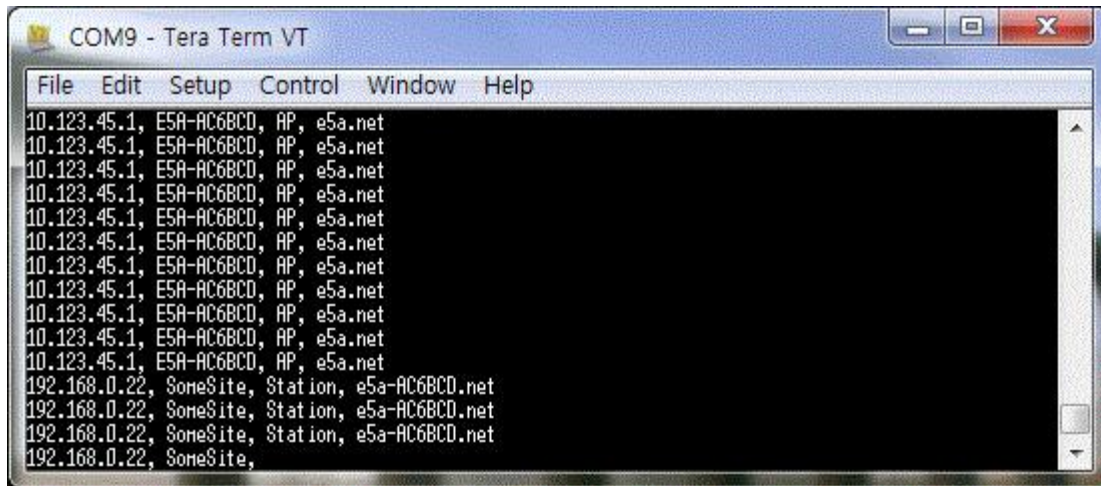
[그림14] Profiles 설정

Profiles 탭을 선택합니다. 기존에 구축된 네트워크 정보를 입력합니다. [그림 14]에서는 “SomeSite”라는 AP로 연결하는 정보를 보여줍니다. SSID, Security Type, Security Key는 연결한 AP의 정보를 입력하고, Profile Priority는 등록된 profile 중 연결 우선순위를 입력합니다. E5A는 우선순위에 따라 순차적으로 연결을 시도합니다. 모든 값을 입력한 후, 버튼을 누릅니다. (버튼에 “쿼리 전송”이라고 나오는데 Internet Explorer를 사용한 경우에 이렇고, Chrome에서는 “Add”로 표시됩니다.)



[그림15] Device 설정

Device 탭을 선택합니다. Device Mode를 Station으로 선택하고, 버튼을 누릅니다. 전원을 끄고 잠시 후 다시 켜면 E5A가 SomeSite에 연결됩니다. (E5A의 전원을 끄고, 내부에 축적된 전류를 모두 소진할 때까지 기다려야 합니다. 약 1초 정도 기다립니다.)



[그림16] Station으로 변경된 후 RS485 출력 내용

E5A가 “SomeSite” 네트워크에 연결되었습니다. 이제 “SomeSite” 네트워크에 속한 어떤 PC에서든 쉽게 E5A로 접근하여 작업을 변경하고, 상태를 감시하고, 제어를 할 수 있습니다.

4. 사용자의 감시 및 제어

E5A를 포함한 대부분의 PLC CPU는 HMI(Human Machine Interface)가 장치와 분리되어 제공됩니다. 사용자가 원할 때마다 시스템의 현재 상태를 보고, 제어 과정에 개입하며, 중요한 제어행위나 상태의 변화를 기록하기 위해 일반적으로 SCADA를 설치합니다.

SCADA는 다양한 형태로 구축될 수 있습니다. 가장 일반적인 형태는 PC용 application입니다. PC에 application 형태의 SCADA를 설치하고, 이를 실행하여 PLC CPU(E5A 등등)와 통신 interface를 통해 연결하고, 시스템에 대한 제어와 감시 및 기록(logging) 등을 수행합니다.

E5A는 SCADA를 위해 RS485와 WiFi의 2가지 경로를 제공할 수 있습니다. 사용자는 2가지 경로를 동시에 사용할 수도 있지만, 제어가 중복되어 예상과 다른 동작을 할 수 있으므로 한 가지만 선택하는 것을 권장합니다. RS485 interface를 선택하면 RS485에 연결된 다른 장치들도 SCADA에서 직접 제어 및 감시를 해야 합니다. 이 경우 SCADA는 RS485 선로에서 MODBUS RTU master 역할을 맡게 되며, RS485 선로 상의 나머지 모든 장치들은 MODBUS RTU slave가 되기 때문입니다. WiFi server interface를 선택하면, SCADA는 MODBUS TCP client 역할을 합니다. 이 경우, E5A는 RS485 선로상의 다른 장치들에 대해 MODBUS RTU master 역할을 수행할 수도 있습니다.

4.1. 간단하게 감시만 가능한 방법

히터와 쿨러를 이용하여 0~10도 사이의 온도로 유지하는 작업을 설계합니다. 실제 쿨러와 히터를 준비할 필요는 없고, 960 ~ 1070[Ω]의 구간을 조절할 수 있는 가변 저항만 있으면 실험할 수 있습니다. DO1에는 히터가 연결되고, DO2에는 쿨러가 연결되는 것으로 가정합니다. E5A가 DO1을 제어하여 ON시키면, DO1 LED가 ON되므로 이것을 보고 히터가 ON된 것으로 판단합니다.

```

CONFIGURATION nameOfConf
TYPE T_UDF:
  STRUCT
    length : BYTE;
    buffer : STRING;
  END_STRUCT
END_TYPE

VAR_GLOBAL
  gUsrSend0, gUsrSend1 : T_UDF;
  gTemp1 AT %IW16 : INT;
  gTemp2 AT %IW17 : INT;
  gTemp3 AT %IW18 : INT;
  gTemp4 AT %IW19 : INT;
  gOhm1 AT %IW32 : WORD;
  gOhm2 AT %IW33 : WORD;
  gOhm3 AT %IW34 : WORD;
  gOhm4 AT %IW35 : WORD;
  gDscGet AT %IW47 : WORD;

```

```

gDsc1 AT %IW48 : WORD;
gDsc2 AT %IW49 : WORD;
gDsc3 AT %IW50 : WORD;
gDsc4 AT %IW51 : WORD;
gDscSet AT %QW47 : WORD;
END_VAR

RESOURCE extLine1 ON ETH_1
  VAR_GLOBAL
    gConf : CONF_ETH1;
    gCommUsr : COMM_ETHUSR;
  END_VAR

  TASK taskInit (SINGLE := TRUE, PRIORITY := 1);
  TASK taskSync (INTERVAL := t#3s, PRIORITY := 2);

  PROGRAM pgm1Init WITH taskInit : Prog1Init();
  PROGRAM WITH taskSync : Prog1Sync();
END_RESOURCE
END_CONFIGURATION

PROGRAM Prog1Init
  (* 통신 설정: SET, user defined server mode, IPv4/23 *)
  extLine1.gConf(GET_SET := 1, MODE := 2, IPV4_6 := 0, PORT := 23);
  extLine1.gCommUsr.EC := 0; (* 통신 결과 초기화 *)
END_PROGRAM

PROGRAM Prog1Sync
  VAR
    vIdx, vDsc : INT := 0;
    vTemp, vOhm : REAL;
    vStr : STRING;
  END_VAR

  (* 통신 요청의 처리가 끝날 때까지 기다림 *)
  IF extLine1.gCommUsr.EC = 1 THEN RETURN; END_IF;

  (* 온도 확인 *)
  IF %IX0 THEN
    vIdx := 1;
    vTemp := gTemp1 / 10;
    vDsc := gDsc1;

```



```

    vOhm := gOhm1 / 10;
ELSIF %IX1 THEN
    vIdx := 2;
    vTemp := gTemp2 / 10;
    vDsc := gDsc2;
    vOhm := gOhm2 / 10;
ELSIF %IX2 THEN
    vIdx := 3;
    vTemp := gTemp3 / 10;
    vDsc := gDsc3;
    vOhm := gOhm3 / 10;
ELSIF %IX3 THEN
    vIdx := 4;
    vTemp := gTemp4 / 10;
    vDsc := gDsc4;
    vOhm := gOhm4 / 10;
END_IF;

(* 히터/쿨러 제어 *)
IF vIdx <> 0 THEN
    vStr := CONCAT('Temp', STR_FROM_INT(vIdx), ' = ', STR_FROM_REAL(vTemp), '
[degC], Heater = ');
    IF vTemp < 0 THEN
        %QX0 := 1;
        vStr := CONCAT(vStr, 'ON');
    ELSE
        %QX0 := 0;
        vStr := CONCAT(vStr, 'OFF');
    END_IF;
    vStr := CONCAT(vStr, ', Cooler = ');
    IF vTemp > 10 THEN
        %QX1 := 1;
        vStr := CONCAT(vStr, 'ON');
    ELSE
        %QX1 := 0;
        vStr := CONCAT(vStr, 'OFF');
    END_IF;
    vStr := CONCAT(vStr, '$!$r');
ELSE
    %QX0 := 0;
    %QX1 := 0;
    vStr := 'Heater = OFF, Cooler = OFF$!$r';

```

```

END_IF;

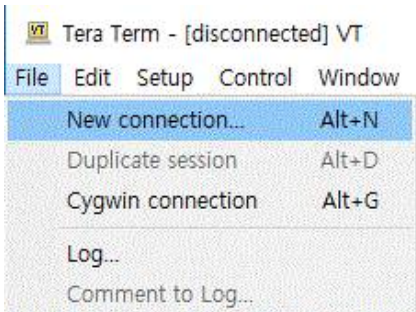
(* 통신 요청: SEND *)
gUsrSend0.buffer := vStr;
extLine1.gCommUsr(RECV_SEND := 1, SEND_ADDR := ADDR_OF(gUsrSend0), SEND_LEN
:= LEN(gUsrSend0.buffer));
IF vIdx = 0 THEN RETURN; END_IF;

vStr := CONCAT('DSC=', STR_FROM_INT(vDsc), ', set=', STR_FROM_INT(gDscSet), ',
now=', STR_FROM_INT(gDscGet));
vStr := CONCAT(vStr, ', R=', STR_FROM_REAL(vOhm), ' [Ohm]$I$r');
gUsrSend1.buffer := vStr;
extLine1.gCommUsr(RECV_SEND := 1, SEND_ADDR := ADDR_OF(gUsrSend1), SEND_LEN
:= LEN(gUsrSend1.buffer));
END_PROGRAM

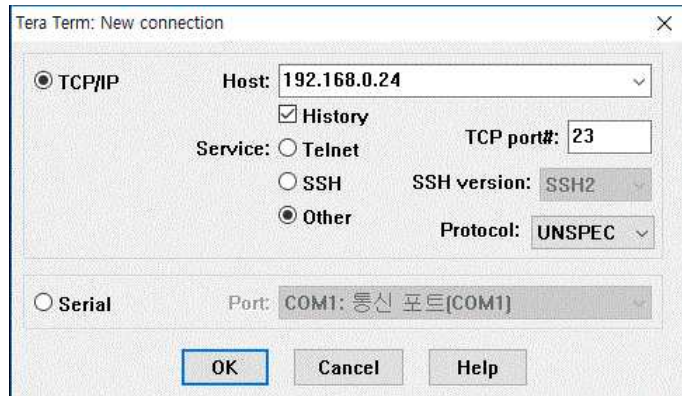
```

[DST 2] text_scada0.dst

위 작업을 실행시키고, Tera Term™을 이용해서 E5A에 접속합니다. E5A의 IP를 미리 확인해서 알고 있어야 합니다.



[그림17] 새 연결 열기



[그림18] 연결 설정

작업에서 설정한 대로 port는 23을 입력합니다. Tera Term™이 연결에 성공하면, UI 단자에 연결된 저항값에 따라 출력되는 내용을 볼 수 있습니다.

다음에 보이는 예는 무부하(저항을 연결하지 않은 경우), 단락(선으로 직접 연결한 경우, 저항값은 0), 저온 (0℃ 미만, 저항값으로는 1000Ω 미만, 히터 ON 조건), 정온(0~10℃ 사이, 저항값으로는 1000~1039Ω 사이, 히터 및 쿨러 OFF 조건), 고온(10℃ 초과, 저항값으로는 1039Ω 초과, 쿨러 ON 조건)의 각각에 대한 출력 내용입니다.

```

VT 192.168.0.24 - Tera Term VT
File Edit Setup Control Window KanjiCode Help
Heater = OFF, Cooler = OFF
Heater = OFF, Cooler = OFF
Heater = OFF, Cooler = OFF
Heater = OFF, Cooler = OFF

```

[그림19] 무부하

```

VT 192.168.0.24 - Tera Term VT
File Edit Setup Control Window KanjiCode Help
Temp1 = -200.000000 [degC], Heater = ON, Cooler = OFF
DSC=392, set=0, now=350, R=2.800000 [Ohm]
Temp1 = -200.000000 [degC], Heater = ON, Cooler = OFF
DSC=388, set=0, now=350, R=2.500000 [Ohm]
Temp1 = -200.000000 [degC], Heater = ON, Cooler = OFF
DSC=388, set=0, now=397, R=0.000000 [Ohm]
Temp1 = -200.000000 [degC], Heater = ON, Cooler = OFF
DSC=383, set=0, now=397, R=0.000000 [Ohm]

```

[그림20] 단락

```

VT 192.168.0.24 - Tera Term VT
File Edit Setup Control Window KanjiCode Help
DSC=0, set=0, now=382, R=950.200012 [Ohm]
Temp1 = -12.800000 [degC], Heater = ON, Cooler = OFF
DSC=0, set=0, now=382, R=949.799988 [Ohm]
Temp1 = -12.800000 [degC], Heater = ON, Cooler = OFF
DSC=0, set=0, now=382, R=949.599976 [Ohm]
Temp1 = -12.900000 [degC], Heater = ON, Cooler = OFF
DSC=0, set=0, now=382, R=949.400024 [Ohm]
Temp1 = -12.900000 [degC], Heater = ON, Cooler = OFF
DSC=0, set=0, now=382, R=949.299988 [Ohm]
Temp1 = -12.900000 [degC], Heater = ON, Cooler = OFF
DSC=0, set=0, now=382, R=949.400024 [Ohm]

```

[그림21] 저온

```

VT 192.168.0.24 - Tera Term VT
File Edit Setup Control Window KanjiCode Help
DSC=0, set=0, now=382, R=1032.199951 [Ohm]
Temp1 = 8.100000 [degC], Heater = OFF, Cooler = OFF
DSC=0, set=0, now=382, R=1031.900024 [Ohm]
Temp1 = 8.200000 [degC], Heater = OFF, Cooler = OFF
DSC=0, set=0, now=382, R=1032.000000 [Ohm]
Temp1 = 8.200000 [degC], Heater = OFF, Cooler = OFF
DSC=0, set=0, now=382, R=1032.000000 [Ohm]
Temp1 = 8.100000 [degC], Heater = OFF, Cooler = OFF
DSC=0, set=0, now=382, R=1031.800049 [Ohm]
Temp1 = 8.200000 [degC], Heater = OFF, Cooler = OFF
DSC=0, set=0, now=382, R=1032.000000 [Ohm]

```

[그림22] 정온

```

VT 192.168.0.24 - Tera Term VT
File Edit Setup Control Window KanjiCode Help
DSC=0, set=0, now=382, R=1062.900024 [Ohm]
Temp1 = 16.100000 [degC], Heater = OFF, Cooler = ON
DSC=0, set=0, now=382, R=1062.699951 [Ohm]
Temp1 = 16.000000 [degC], Heater = OFF, Cooler = ON
DSC=0, set=0, now=382, R=1062.599976 [Ohm]
Temp1 = 16.000000 [degC], Heater = OFF, Cooler = ON
DSC=0, set=0, now=382, R=1062.500000 [Ohm]
Temp1 = 16.000000 [degC], Heater = OFF, Cooler = ON
DSC=0, set=0, now=382, R=1062.500000 [Ohm]
Temp1 = 16.000000 [degC], Heater = OFF, Cooler = ON
DSC=0, set=0, now=382, R=1062.500000 [Ohm]
Temp1 = 16.000000 [degC], Heater = OFF, Cooler = ON
DSC=0, set=0, now=382, R=1062.500000 [Ohm]
Temp1 = 16.000000 [degC], Heater = OFF, Cooler = ON

```

[그림23] 고온

4.2. 간단한 제어와 감시가 가능한 방법

```

CONFIGURATION nameOfConf
TYPE T_UDF:
STRUCT
length : BYTE;
buffer : STRING;
END_STRUCT
END_TYPE

VAR_GLOBAL
gUsrSend, gUsrRecv : T_UDF;
gTemp1 AT %IW16 : INT;
gTemp2 AT %IW17 : INT;
gTemp3 AT %IW18 : INT;
gTemp4 AT %IW19 : INT;
END_VAR

RESOURCE extLine1 ON ETH_1
VAR_GLOBAL
gConf : CONF_ETH1;
gCommUsr : COMM_ETHUSR;
END_VAR

TASK taskInit (SINGLE := TRUE, PRIORITY := 1);
TASK taskSync (INTERVAL := t#1s, PRIORITY := 2);

PROGRAM pgm1Init WITH taskInit : Prog1Init();
PROGRAM WITH taskSync : Prog1Sync();
END_RESOURCE

```

```
END_CONFIGURATION
```

```
PROGRAM Prog1Init
```

```
(* 통신 설정: SET, user defined server mode, IPv4/23 *)
extLine1.gConf(GET_SET := 1, MODE := 2, IPV4_6 := 0, PORT := 23);
extLine1.gCommUsr.RECV_ADDR := ADDR_OF(gUsrRecv);
extLine1.gCommUsr.RECV_LEN := SIZEOF(gUsrRecv.buffer) / 8;
extLine1.gCommUsr(RECV_SEND := 0);
```

```
END_PROGRAM
```

```
PROGRAM Prog1Sync
```

```
VAR
```

```
  vLen, vPosL, vPosR : INT;
  vTemp : REAL;
  vStr0, vStr1 : STRING;
```

```
END_VAR
```

```
(* 명령 확인 *)
```

```
vLen := gUsrRecv.length;
IF vLen < 3 THEN RETURN; END_IF;
vStr0 := gUsrRecv.buffer;
vPosL := FIND(vStr0, '$l');
vPosR := FIND(vStr0, '$r');
IF vLen < 10 THEN
  IF (vPosL < 0) & (vPosR < 0) THEN RETURN; END_IF;
END_IF;
```

```
vStr1 := vStr0;
vStr0 := LEFT(vStr0, 1);
vStr1 := DELETE(vStr1, 0, 1);
vPosL := STR_TO_INT(vStr1);
IF vStr0 = '?' THEN
```

```
  CASE vPosL OF
```

```
    1: (* UI1 상태 출력 *)
```

```
      vStr1 := CONCAT('Temp1 = ', STR_FROM_REAL(gTemp1 / 10), ' [degC]');
```

```
    2: (* UI2 상태 출력 *)
```

```
      vStr1 := CONCAT('Temp2 = ', STR_FROM_REAL(gTemp2 / 10), ' [degC]');
```

```
    3: (* UI3 상태 출력 *)
```

```
      vStr1 := CONCAT('Temp3 = ', STR_FROM_REAL(gTemp3 / 10), ' [degC]');
```

```
    4: (* UI4 상태 출력 *)
```

```
      vStr1 := CONCAT('Temp4 = ', STR_FROM_REAL(gTemp4 / 10), ' [degC]');
```

```
  ELSE
```

```

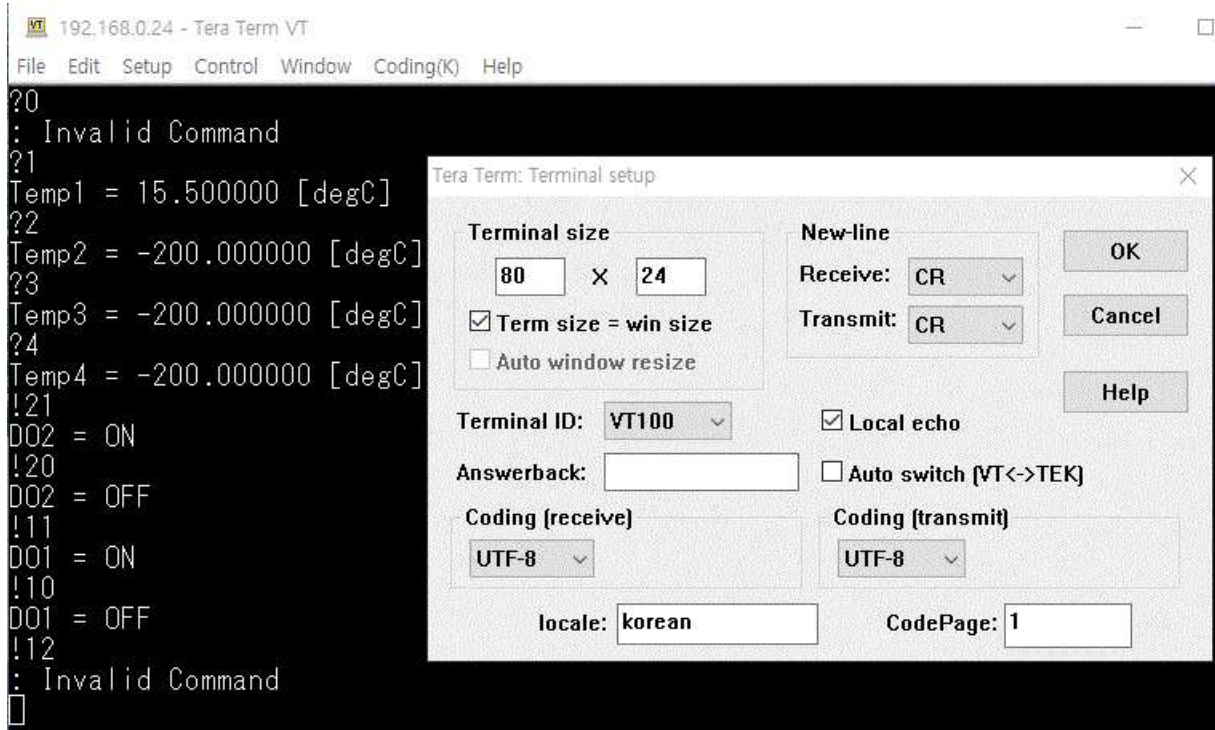
        vStr1 := ': Invalid Command';
    END_CASE;
ELSIF vStr0 = '!' THEN
    CASE vPosL OF
        10: (* DO1 OFF *)
            %QX0 := 0;
            vStr1 := 'DO1 = OFF';
        11: (* DO1 ON *)
            %QX0 := 1;
            vStr1 := 'DO1 = ON';
        20: (* DO2 OFF *)
            %QX1 := 0;
            vStr1 := 'DO2 = OFF';
        21: (* DO2 ON *)
            %QX1 := 1;
            vStr1 := 'DO2 = ON';
    ELSE
        vStr1 := ': Invalid Command';
    END_CASE;
ELSE
    vStr1 := ': Invalid Command';
END_IF;

(* 통신 요청: SEND & RECV *)
gUstrSend.buffer := CONCAT('$!$r', vStr1, '$!$r');
extLine1.gCommUstr.SEND_ADDR := ADDR_OF(gUstrSend);
extLine1.gCommUstr.SEND_LEN := LEN(gUstrSend.buffer);
extLine1.gCommUstr(RECV_SEND := 2);
END_PROGRAM

```

[DST 3] text_scada1.dst

Tera Term™으로 E5A에 접속하여 명령을 입력합니다. [그림24]와 같이 Terminal setup에서 Local echo를 ON으로 두어야 입력하는 명령이 화면에 보입니다.



[그림24] text_scada1 접속 화면과 terminal setup

5. 사양

5.1. 전원

E5A의 전원 회로는 SMPS 사용을 기준으로 설계되었습니다. 출력 24[Vdc], 20[W] 사용을 권장합니다.

(1) 한계치

전원 전압은 30[Vdc] 미만이어야 합니다.

(2) 정격

전원 전압은 20 ~ 27 [Vdc]에서 정상 동작합니다.

전류량은 순간 최대 0.3[A]입니다. 평균으로는 최대 0.2[A]입니다.

(3) Isolation

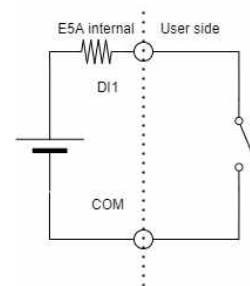
격리(isolation)는 회로 요소 사이의 전기적인 분리를 의미합니다. 시스템에 속한 장치들은 도체인 전선으로 연결되어 있으므로 버락이나 전원 충격파가 전선에 전달되면, 그 전선에 연결된 격리군(회로를 통해 직접 전류를 주고받을 수 있는 회로 무리)은 충격파가 소진될 때까지 영향을 받습니다. 보통은 회로 중 충격파와 가장 가깝고 저항이 낮은 곳이 먼저 파괴됩니다. 그러므로 자동제어 시스템을 설계하면서 적절한 격리군을 고려하지 않으면 충격파가 사용자에게도 전달될 수 있습니다. E5A는 사용자와 시스템의 안전을 위해 내부에 2개의 격리군이 있습니다.

격리군 1	전원, DI, DO
격리군 2	RS485

5.2. 입출력 및 통신

5.2.1. DI

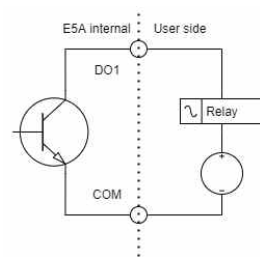
E5A는 8개의 DI를 가지고 있습니다. [그림25]는 그 중 DI1의 내부 회로를 이해하기 쉽게 표현한 것입니다. 나머지 DI도 동일한 형태를 가집니다. DI는 외부 DO의 상태를 측정하기 위한 전원을 공급합니다. [그림25]는 외부DO로써 스위치를 연결한 예입니다. [그림25]의 현재 상태는 스위치가 떨어진(개방된) 상태이고, DI1은 OFF로 인식합니다. 스위치를 눌러서 회로가 연결(단락)되면 DI1은 ON으로 인식합니다.



[그림25] DI 내부회로모델과 사용 예

5.2.2. DO

E5A는 8개의 DO를 가지고 있습니다. [그림26]은 그 중 DO1의 내부 회로를 이해하기 쉽게 표현한 것입니다. 나머지 DO도 동일한 형태를 가집니다. [그림26]에서 DO1과 COM 단자는 설정값에 따라 개방(open, 끊음)되거나 단락(short, 연결)됩니다. E5A의 전원이 공급된 초기에 갖는 DO의 상태는 OFF이고, DO1과 COM 단자는 개방(open)되어 있습니다(power-on default).



[그림26] DO 내부회로모델과 사용 예

5.2.3. RS485

E5A의 RS485는 최대 64개의 노드를 연결할 수 있습니다. Baudrate은 300 ~ 3,000,000[bps]로 설정할 수 있습니다.

사용자는 다음의 설정 중 하나만 선택하거나 아무 선택도 없이 E5A를 운용할 수 있습니다.

(1) MODBUS RTU master/slave 연결

E5A가 MODBUS RTU master로 설정되면, 다른 E5A나 MODBUS RTU slave를 운용하는 장치의 값을 읽거나 쓸 수 있습니다.

E5A가 MODBUS RTU slave로 설정되면, master 역할을 하는 SCADA나 다른 E5A가 E5A의 내부 메모리에서 값을 읽거나 쓸 수 있습니다. 이때 외부의 제어 범위를 제한하기 위해 E5A는 내부 메모리에 대한 쓰기 방지 기능을 가지고 있습니다. 사용자는 작업 내용으로 쓰기 방지 영역을 제어할 수 있습니다.

(2) 사용자 정의 프로토콜 serial 연결

E5A의 소개를 위해 [DST 2]에서 사용한 것이 사용자 정의 프로토콜 serial입니다. 이때 상대방 사용자 정의 프로토콜 serial의 역할을 한 것은 Tera Term™입니다.

5.2.4. WiFi

E5A의 WiFi는 802.11b/g/n 표준의 Station으로 운용이 가능하고, 802.11b/g 표준의 AP(Access Point)로 운용도 가능합니다. E5A의 초기 상태는 AP 모드로 운용됩니다. 초기 상태의 SSID는 E5A-XXXXXX 형식을 가지며, X부분은 E5A의 MAC address를 기반으로 구성됩니다. 초기 상태의 password는 “ilpum.net”이고, login ID는 “Admin”입니다. 비인가자의 접근을 막기 위해 SSL(Secure Socket Layer)을 사용합니다. 이를 위해 사용하는 보안 인증서가 인증기관에 등록되지 않은 자체제작 인증서이기 때문에 일반적인 web browser에서는 인증서 오류를 발생시킵니다. 사설 인증서의 사용은 E5A가 가진 특수성 때문에 선택된 것이므로 사용자는 web browser의 인증서 오류를 무시하고 사용해야 합니다.

(참고1) E5A는 IPv4 또는 IPv6 환경에서 사용할 수 있습니다. (단, 대부분의 시험이 IPv4에서만 진행되었음)

(참고2) 초기화 버튼(INIT)은 사용자가 과거에 설정한 보안 설정을 잊었을 때, 이를 복구하기 위해 존재합니다. 사용자가 초기화 버튼을 3초 이상 누르면, E5A는 초기 상태로 모든 설정이 변경되며 재기동합니다.

(주의1) web browser와 E5aLoader는 동시에 동일한 E5A로 연결할 수 없습니다. Web browser도 2개 이상을 동시에 동일한 E5A로 연결할 수 없습니다. E5A가 오작동할 수 있습니다.

(주의2) E5A는 server를 1개만 운용할 수 있습니다. 사용자는 MODBUS TCP server와 사용자 정의 TCP server 중 하나만 운용해야 합니다. 이 특성은 client에도 동일하게 적용됩니다.

(1) Web browser 연결 (network 설정)

AP로 운용되는 경우, E5A는 DNS를 운용하며 사용자는 “e5a.net”으로 E5A의 IP를 얻을 수 있습니다. 보안 관련 정보는 web browser로 E5A에 접속하여 변경할 수 있습니다. (Login ID는 변경할 수 없습니다.)

Station으로 운용하기 위해 접속할 AP에 대한 정보를 web browser로 접속하여 등록/삭제할 수 있습니다. 운용 모드를 AP나 Station으로 선택할 수 있으며, 이들 설정 변경은 E5A의 전원을 끄고 다시 켜는 과정을 통해 실제 운용에 반영됩니다.

(2) E5aLoader 연결

PC에서 실행되는 E5aLoader는 E5A에게 새로운 작업 내용을 시험/저장하기 위해 사용합니다. E5A와 E5aLoader는 E5A의 WiFi를 통해 연결됩니다. E5aLoader에서 E5A로 연결을 요청하므로 연결 전에 E5A의 운용 모드(AP 혹은 Station), IP(또는 hostname), password 등을 확인해두어야 합니다.

(3) MODBUS TCP server/client 연결

E5A가 MODBUS TCP server로 설정되면, SCADA나 다른 E5A는 client로 접속하여 E5A의 내부 메모리에서 값을 읽거나 쓸 수 있습니다. 단, client로 접속할 수 있는 장치의 수는 최대 1개입니다. 새로운 장치가 연결되면 이전의 연결이 끊어집니다.

E5A가 MODBUS TCP client로 설정되면 E5A는 다른 E5A나 MODBUS TCP server를 운용하는 장치로 접속하여 값을 읽거나 쓸 수 있습니다. E5A가 연결할 수 있는 server의 수는 최대 1개입니다. 다른 server로 접속하려면 기존의 연결을 끊고 다시 연결해야 합니다.

MODBUS TCP server/client 연결에 대해서는 보안 프로토콜이 적용되지 않습니다. 즉, 전송되는 내용이 암호화되지 않으며 password 등으로 접근 권한을 확인하는 과정도 없습니다. 그래서 E5A는 보안을 위해 내부 메모리에 대한 쓰기 방지 기능을 가지고 있으며, 사용자가 저장한 작업 내용에 따라 쓰기 방지 영역을 제어할 수 있습니다.

(4) 사용자 정의 프로토콜 TCP server/client 연결

E5A의 소개를 위해 [DST 3]에서 사용한 것이 사용자 정의 프로토콜 TCP server입니다. 이때 사용자 정의 프로토콜 TCP client의 역할을 한 것은 Tera Term™입니다. 이와 비슷하게 E5A는 사용자 정의 프로토콜 TCP client로도 설정될 수 있습니다. E5A가 server이든 client이든 최대 연결 수는 1개입니다.

5.3. 메모리

E5A는 3개의 메모리 영역을 가지고 있습니다. 이들은 입력 단자(UI)들이 mapping된 입력 메모리 영역, 출력 단자(AO, DO)들이 mapping된 출력 메모리 영역, 일반적인 변수들이 존재하는 내부 메모리 영역입니다. 입력 메모리 영역은 MODBUS의 input status와 input register 영역에 mapping되고, 출력 메모리 영역은 MODBUS의 coil status와 holding register 영역에 mapping되고, 내부 메모리 영역은 MODBUS의 general register 영역에 mapping됩니다.

메모리 영역 내에서 주소는 0부터 시작되며, 변수의 크기에 따라 실제 위치가 정해집니다. 변수의 형이 BOOL인 경우, 크기가 1bit이므로 연이은 변수의 간격은 1bit입니다. 변수의 형이 DWORD인 경우, 크기가 32bit(또는 4byte)이므로 연이은 변수의 간격은 4byte입니다.

E5A가 지원하는 기본 변수형은 13가지입니다.

변수형	Byte 크기	Bit 크기	최솟값	최댓값
BOOL	-	1	0	1
BYTE USINT	1	8	0	255
SINT	1	8	-128	127
WORD UINT	2	16	0	65,535
INT	2	16	-32,768	32,767

변수형	Byte 크기	Bit 크기	최솟값	최댓값
DWORD UDINT	4	32	0	4,294,967,295
DINT	4	32	-2,147,483,648	2,147,483,647
REAL	4	32	-3.4e38	3.4e38
TIME	8	64	-	-
TOD	8	64	tod#0:0:0.0	tod#23:59:59.999
DATE	8	64	d#1900-1-1	-
DT	8	64	dt#1900-1-1-0:0:0.0	-
STRING	64(지정 가능)	-	-	-

주소표현은 %IX0처럼 쓰는데, %는 주소를 나타내는 지시자이고, I는 입력 메모리 영역을 뜻합니다. 출력 메모리 영역은 Q를 사용하고, 내부 메모리 영역은 M을 사용합니다. X는 변수형으로 BOOL을 뜻합니다. 주소표현에 사용 가능한 변수형의 지시자는 B(BYTE, 1 byte 크기), W(WORD, 2 byte 크기), D(DWORD, 4 byte 크기), L(LWORD, 8 byte 크기)도 있습니다.

이해를 돕기 위해 동일한 주소에 각 주소 표현법을 적용한 예를 보입니다.

주소 값 = 64	ADDROF() 적용	SIZEOF() 적용
%MX64	64 ← ADDRUF(%MX64)	1 ← SIZEOF(%MX64)
%MB8	64 ← ADDRUF(%MB8)	8 ← SIZEOF(%MB8)
%MW4	64 ← ADDRUF(%MW4)	16 ← SIZEOF(%MW4)
%MD2	64 ← ADDRUF(%MD2)	32 ← SIZEOF(%MD2)
%ML1	64 ← ADDRUF(%ML1)	64 ← SIZEOF(%ML1)

내부 메모리 영역은 전체가 일반 영역(0 ~ 8191 [byte])입니다. 사용자가 E5A에 선언하는 변수는 위치를 특정하지 않는 경우에 내부 메모리에 배치됩니다. E5A의 함수 중에는 메모리의 주소를 지정하는 것들이 있는데, 그 함수에게 전달되는 주소는 반드시 함수에서 지정한 메모리 영역에 속해 있어야 합니다.

입력 메모리 영역은 일반 영역(0 ~ 2047 [byte]), 제한 영역(2048 ~ 8191 [byte]), 시스템 정보 영역(18000 ~ 19999 [byte])으로 구성됩니다. 일반 영역과 제한 영역은 사용자의 작업에서 읽고 쓰기가 가능합니다. 시스템 정보 영역은 사용자의 작업에서 읽기만 가능합니다. 제한 영역은 E5A의 쓰기 방지 기능을 위해 할당된 영역이므로 사용자가 값을 바꾸면, 특정 주소에 대해 쓰기 방지 기능이 변경될 수 있습니다.

주소표현	Byte 주소	Bit 주소	변수형	입력 메모리 영역 설명
%IX0	-	0	BOOL	DI1의 값. 0 = OFF, 1 = ON
%IX1	-	1	BOOL	DI2의 값. 0 = OFF, 1 = ON
%IX2	-	2	BOOL	DI3의 값. 0 = OFF, 1 = ON
%IX3	-	3	BOOL	DI4의 값. 0 = OFF, 1 = ON
%IX4	-	4	BOOL	DI5의 값. 0 = OFF, 1 = ON
%IX5	-	5	BOOL	DI6의 값. 0 = OFF, 1 = ON
%IX6	-	6	BOOL	DI7의 값. 0 = OFF, 1 = ON

주소표현	Byte 주소	Bit 주소	변수형	입력 메모리 영역 설명
%IX7	-	7	BOOL	DI8의 값. 0 = OFF, 1 = ON
%IW9900	19800	-	UINT	Design Year
%IW9901	19802	-	UINT	Family Number
%IW9902	19804	-	UINT	Product Number
%IW9903	19806	-	UINT	Compatibility Number
%IW9910	19820	-	WORD	MAC 1, 2
%IW9911	19822	-	WORD	MAC 3, 4
%IW9912	19824	-	WORD	MAC 5, 6
%IW9990	19980	-	UINT	Version
%IW9991	19982	-	UINT	Lot

출력 메모리 영역은 일반 영역(0 ~ 2047 [byte]), 개방 영역(2048 ~ 4095 [byte]), 제한 영역(4096 ~ 8191 [byte])으로 구성됩니다. 모든 영역은 사용자의 작업에서 읽고 쓰기가 가능합니다. 개방 영역은 MODBUS를 통한 외부 사용자의 접근에서도 쓰기가 항상 가능합니다. 개방 영역은 쓰기 방지 기능이 없는 곳이므로 쓰기 방지 기능에 대한 제어를 고려하지 않고도 외부로부터 명령을 전달받기 위해 이용할 수 있습니다. 제한 영역은 E5A의 쓰기 방지 기능을 위해 할당된 영역이므로 사용자가 값을 바꾸면 특정 주소에 대해 쓰기 방지 기능이 변경될 수 있습니다.

주소표현	Byte 주소	Bit 주소	변수형	출력 메모리 영역 설명
%QX0	-	0	BOOL	DO1의 값. 0 = OFF, 1 = ON
%QX1	-	1	BOOL	DO2의 값. 0 = OFF, 1 = ON
%QX2	-	2	BOOL	DO3의 값. 0 = OFF, 1 = ON
%QX3	-	3	BOOL	DO4의 값. 0 = OFF, 1 = ON
%QX4	-	4	BOOL	DO5의 값. 0 = OFF, 1 = ON
%QX5	-	5	BOOL	DO6의 값. 0 = OFF, 1 = ON
%QX6	-	6	BOOL	DO7의 값. 0 = OFF, 1 = ON
%QX7	-	7	BOOL	DO8의 값. 0 = OFF, 1 = ON

5.4. 처리 능력

작업 내용은 작업객체 단위로 나뉘어져 실행됩니다. 각 작업객체에 대해 실행 주기를 설정할 수 있습니다. 작업객체의 최소 실행 간격은 100ms입니다. 한 작업객체의 최대 실행 가능 시간은 1s입니다. 실행 시간을 초과하면 오류가 발생되고, 이후의 명령행은 실행되지 않습니다.

작업으로 등록할 수 있는 DST 명령행은 최대 약 1000줄입니다. 단, 사용자가 입력한 행의 수와 항상 일치하지는 않습니다.

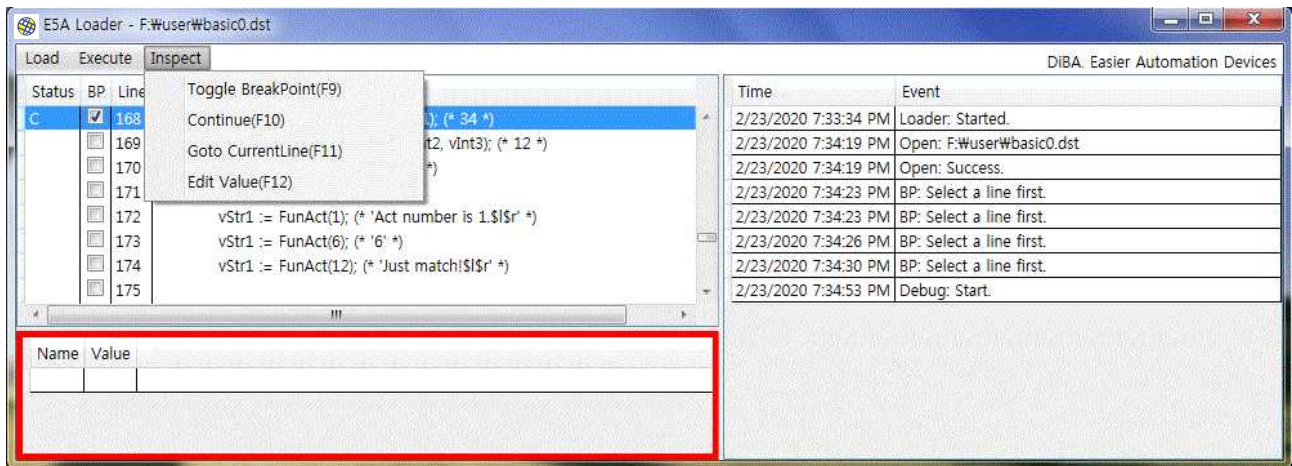
6. E5aLoader 운용

E5aLoader의 메뉴 구성은 다음과 같습니다. 단축키는 괄호 안에 표시합니다.

상위 메뉴 이름	하위 메뉴 이름	설명
Load	Open File(F3)	DST file을 읽어 들입니다.
	Config Connection(F4)	E5A 연결 정보를 입력합니다.
Execute	Run(F5)	DST file을 E5A에 저장하고, 실행합니다.
	Debug(F6)	DST file을 E5A에서 시험합니다.
Inspect	Toggle BreakPoint(F9)	E5A 작업 진행을 멈추는 BP를 설정 혹은 해제합니다.
	Continue(F10)	BP에 멈춘 작업을 다시 진행시킵니다.
	Goto CurrentLine(F11)	BP에 멈춘 경우, 그 위치로 화면을 이동합니다.
	Edit Value(F12)	E5A의 메모리 값을 변경합니다.

Load 메뉴와 Execute 메뉴는 3.3.에서 이미 설명했으므로 여기서는 Debug(F6)를 선택하여 E5A를 시험하는 방법에 대해 설명합니다.

6.1. Inspect 메뉴



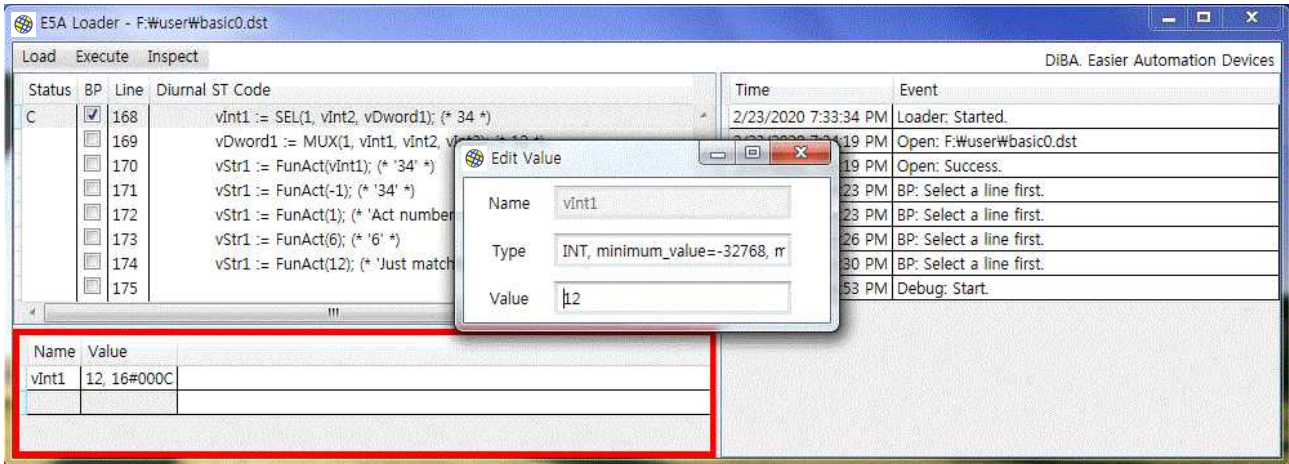
[그림27] Inspect 메뉴

시험(Debug) 중에는 Status창과 Value창이 나타납니다. Status창은 작업 진행이 멈춘 위치와 E5A에서 발생한 오류 정보를 표시합니다. Value창은 사용자가 원하는 위치의 값을 표시하고, 실행 중일 때(파란색)와 멈췄을 때(빨간색)의 테두리 색이 다릅니다.

사용자는 작업내용창에서 keyboard의 방향키와 마우스 등으로 작업행을 선택할 수 있습니다. 이렇게 선택된 것을 커서라고 하며, 커서가 있는 위치에 대해 BP를 설정하거나 해제할 수 있습니다. BP의 설정/해제는 작업행을 마우스로 더블클릭하거나, 메뉴에서 Toggle BreakPoint(F9)을 선택하거나, keyboard의 F9을 눌러서 지시할 수 있습니다.

작업이 BP에서 멈추면 Value창의 내용이 자동으로 갱신됩니다. 사용자는 Value창에 새로운 위치를 지정하거나 기존 위치를 제거할 수 있습니다. Value창에 표시할 내용이 있는 경우, 마우스로 더블클릭하거나 메뉴의

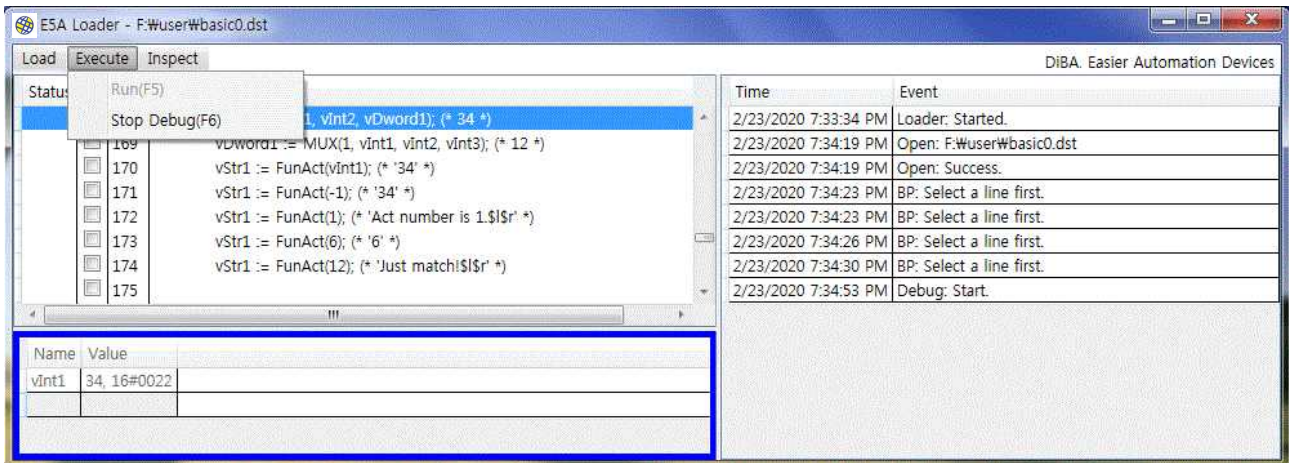
Edit Value(F12)를 선택하거나 keyboard의 F12를 눌러서 값편집창을 열 수 있습니다. 사용자는 값편집창(Edit Value)에서 새로운 값을 지정할 수 있습니다.



[그림28] Edit Value

BP에 멈춘 작업을 다시 진행시키려면 메뉴에서 Continue(F10)을 선택하거나 keyboard의 F10을 누르면 됩니다. E5A가 실행 중에도 BP를 설정할 수 있지만, Value창과 값편집창은 사용할 수 없습니다.

E5A에 대한 시험을 멈추려면, Stop Debug(F6) 메뉴를 선택합니다. [그림 30]에서는 실행 상태에 있을 때의 화면을 보이고 있습니다. Stop Debug(F6)는 E5A가 BP에 멈춘 상태에서도 사용 가능합니다.



[그림29] Stop Debug