Celciux Serisi EJ1N ile NJ PLC Haberleşmesi

İÇİNDEKİLER

Açıklama :

Bu dökümanda EJ1N serisi Celciux ısı kontrol cihazlarının Omron NJ serisi PLC ile haberleşmesi anlatılacaktır. Örnek uygulama, modbus üzerinden gerçek sıcaklık değeri(PV) okunacaktır, ardından ulaşılması istenen sıcaklık değeri (SP) olarak 96 derece ayarlanacaktır. EJ1N serisi tarafında yapılacak ayarlar(termokupl seçimi, kontrol tipi, fiziksel bağlantılar) için aşağıdaki dökümanı referans alabilirsiniz:

https://destek.omron.com.tr/wp-content/uploads/2022/03/Celciux-Serisi-Urunlerimizle-Sicaklik-Kontrolu.pdf

Bu döküman hazırlanırken aşağıdaki ürünler kullanılmıştır:

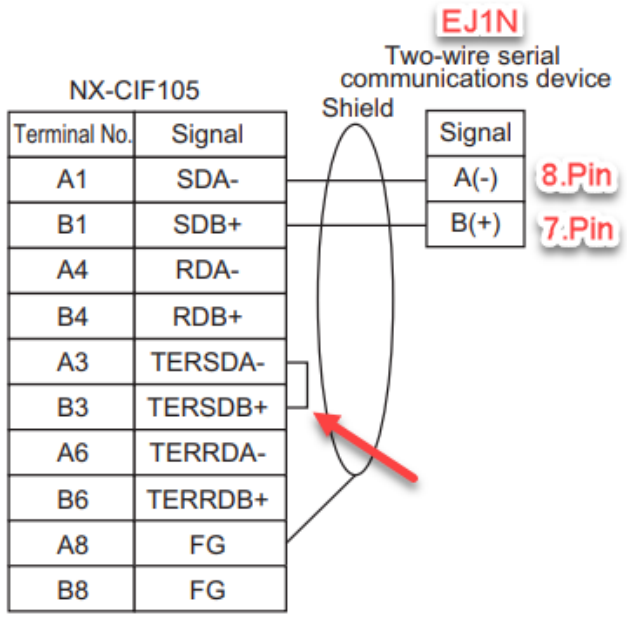
- 1 adet EJ1N-TC2A-QNHB....Ana Ünite
- 1 adet EJ1C-EDUA-NFLK....Sonlandırma Ünitesi
- 1 adet E58-CIFQ1.....Programlama Kablosu
- 1 adet NJ501-1500 serisi PLC
- 1 adet NX-ECC203 Ethercat Coupler
- 1 adet NX-CIF105 Modbus Haberleşme Kartı
- CX-Thermo programı
- Sysmac Studio
- 1 adet SSR Röle
- K Tipi Termokupl(Sıcaklık Sensörü)




Fiziksel Bağlantılar :

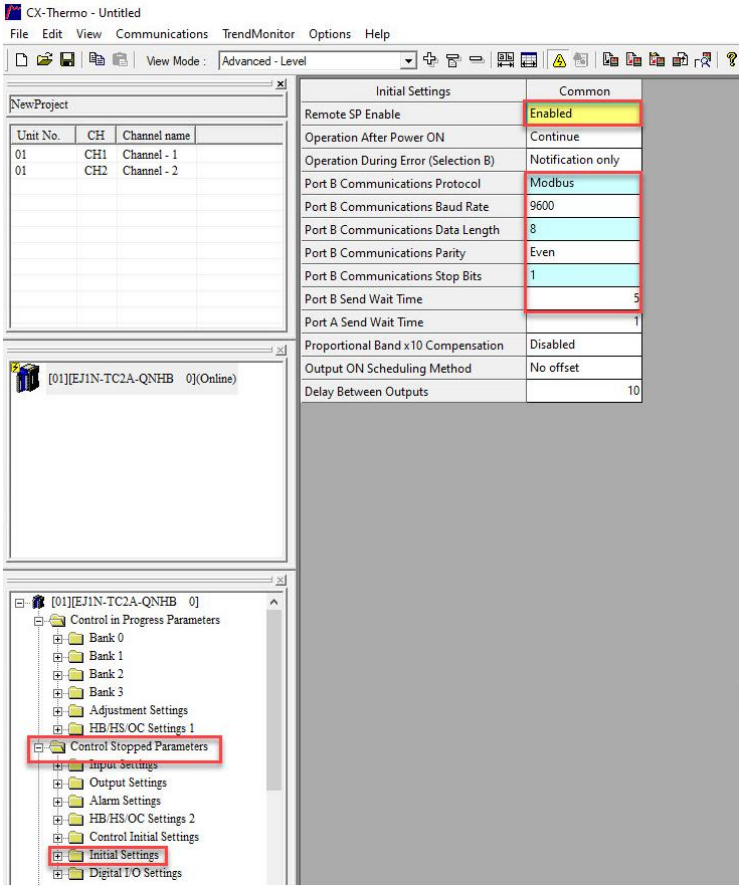
NJ PLC'de modbus haberleşme için NX-CIF105 modülü tercih edilmiştir. Haberleşme parametreleri, modüllerle alakalı menülerden ayarlanır, program içerisinde ise modbus ile alakalı okuma yazma blokları hazır bulunmaktadır.

Modbus haberleşmesi için RS485 standardı üzerinden iki yollu bağlantı yapılır. İki yollu bağlantılarda haberleşme kartı üzerindeki RDA-/RDB+ veya SDA-/SDB+ çiftleri kullanılabilir. TERSDA- ve TERSDB+ arası kısa devre yapılmalıdır. EJ1N terminalleri üzerindeki 8(-) ve 7(+) pinleri veri alışverişi için kullanılmaktadır. Bağlantılar için aşağıdaki şema örnek alınabilir.



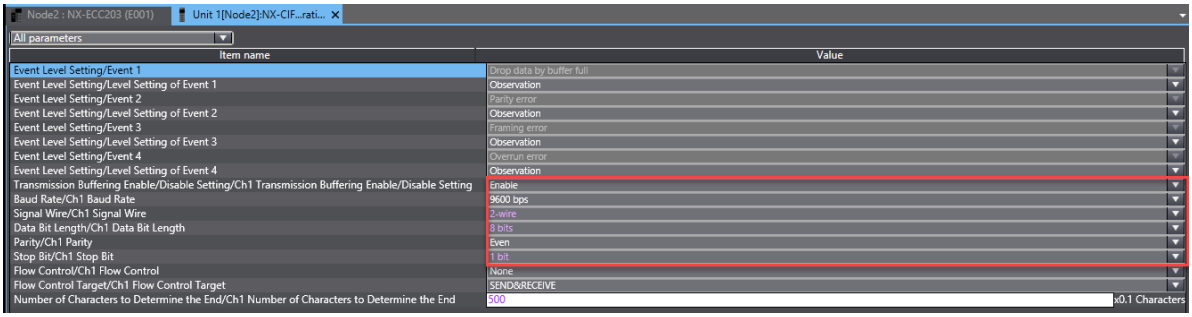
CX-Thermo Programında Haberleşme Ayarları :

CX-Thermo programı üzerinden öncelikle modbus ayarları yapılır. Bunun için Control Stopped Parameters bölümünden Initial Settings bölümüne girilir. Sırasıyla **Remote SP Enable Enabled** yapılır. Ardından **Port B Communication Protocol** parametresinden haberleşme tipi **Modbus** seçilir. **Baud Rate 9600, Data Length-Parity-Stop Bit** sırasıyla **8 Even 1** seçilir. Aşağıdaki resimde ilgili bölümler gösterilmiştir:



NX-CIF105 Tanıtılması ve Haberleşme Ayarları :

Sysmac Studio'da oluşturulan NJ projesinde yer alan ethercat konfigurasyonuna NX-CIF105 kartı eklenir. Kart üzerine çift tıklandığında haberleşme ayarlarının yapıldığı sayfa açılır. Buradaki ayarlar EJ1N ürününde yapılan haberleşme ayarları ile aynı olmalıdır. NX-CIF105 ürünün haberleşme ayarları aşağıdaki resimde gösterilmiştir:



NX_ModbusRtuRead & NX_ModbusRtuWrite :

Modbus üzerinden değişken okuma veya yazma blokları aşağıdaki gibidir:

NX_ModbusRtuWrite

The NX_ModbusRtuWrite instruction sends write commands from a serial port on an NX-series Communications Interface Unit or Option Board to Modbus-RTU slaves using Modbus-RTU protocol.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
NX_ModbusRtuWrite	Send Modbus RTU Write Command	FB	NX_ModbusRtuWrite_instance NX_ModbusRtuWrite Execute / Done DevicePort / Busy SlaveAdr / CommandAborted WriteCmd / Error WriteDat / ErrorID Option / ErrorIDEx Abort	NX_ModbusRtuWrite_instance(Execute, DevicePort, SlaveAdr, WriteCmd, WriteDat, Option, Abort, Done, Busy, CommandAborted, Error, ErrorID, ErrorIDEx);

NX_ModbusRtuRead

The NX_ModbusRtuRead instruction sends read commands from a serial port on an NX-series Communications Interface Unit or Option Board to Modbus-RTU slaves using Modbus-RTU protocol.

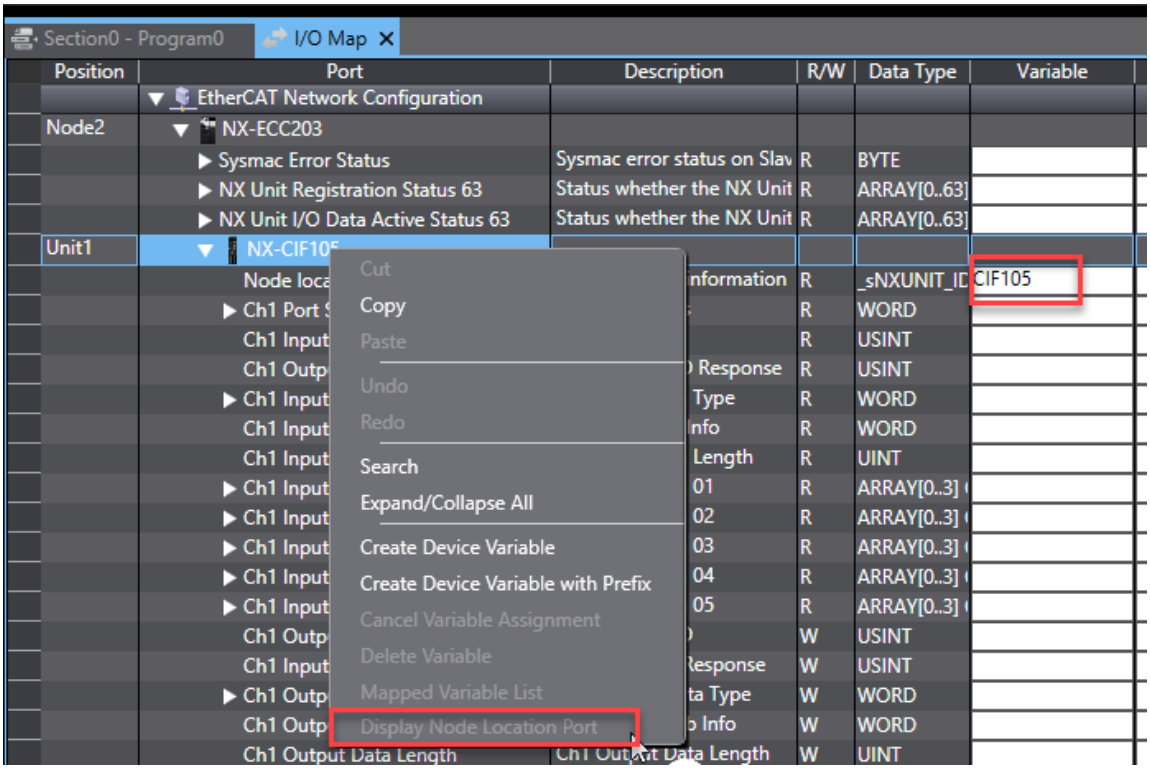
Instruction	Name	FB/ FUN	Graphic expression	ST expression
NX_ModbusRtuRead	Send Modbus RTU Read Command	FB	NX_ModbusRtuRead_instance NX_ModbusRtuRead Execute / Done DevicePort / Busy SlaveAdr / CommandAborted ReadCmd / Error ReadDat / ReadDat Option / ErrorID Abort / ErrorIDEx ReadSize	NX_ModbusRtuRead_instance(Execute, DevicePort, SlaveAdr, ReadCmd, ReadDat, Option, Abort, Done, Busy, CommandAborted, Error, ErrorID, ErrorIDEx, ReadSize);

Bloklarda yer alan değişkenlerin açıklamaları şu şekildedir. Execute; fonksiyon bloğunun çalışması için yükselen kenar olarak tetiklenen bittir. DevicePort; kullanılan haberleşme portunun tanımlandığı bir değişkendir. SlaveAdr; haberleşmenin yapıldığı slave cihazın istasyon numarasıdır. DevicePort struct yapısındaki değişkenleri aşağıdaki gibidir:

Name	Meaning	Description	Data type	Valid range
DevicePort	Device port	Object that represents a device port	_sDEVICE_PORT	---
DeviceType	Device type	Type of the device to specify	_eDEVICE_TYPE	_DeviceNXUnit _DeviceEcatSlave _DeviceOptionBoard
NxUnit	Specified Unit	NX Unit to control	_sNXUNIT_ID	---
EcatSlave	Specified slave	EtherCAT slave to control	_sECAT_ID	---
OptBoard	Specified Option Board	Option Board to control	_sOPTBOARD_ID	---
Reserved	Reserved	Reserved	Reserved	---
PortNo	Port number	Port number 1: Port 1 2: Port 2	USINT	Depends on data type.

DeviceType; kullanılan haberleşme kartının tanımlanması için kullanılır. Kullanım şekline göre 3 farklı şekilde tanımlanabilir. Bu konfigürasyonda NX-CIF105 kartı kullanıldığı için **_DeviceNXUnit** olarak tanımlanır. **NXUnit**; tanımlanan NX kartıdır. **PortNo;** NXUnit için birinci veya ikinci port seçilir.

DevicePort değişkenini oluşturmak için, Sysmac Studio'da IO Map kısmına gelinir. NX1-CIF105'in üzerine gelinerek sağ tuş ile tıklanır ve 'Display Node Location Information' seçilir, kullanıcı variable kısmına değişken ismini belirler.



Program kısmında DeviceType'a bağlı değişkenlerin tanımlaması aşağıdaki gibi yapılır.

```
Port Tanimlamalari
1  //CIF105 ÜZERINDEN HABERLEŞME
2  myDevicePort.DeviceType:=_eDEVICE_TYPE#_DeviceNXUnit;
3  myDevicePort.NxUnit:=CIF105;
4  myDevicePort.PortNo:=USINT#1;
```

NX_ModbusRead bloğunda kullanılan ReadCmd komutu struct yapıda bir değişkendir.

Fun; kullanılacak okuma fonksyionu tanımlanır. Coil bilgisi okumak için _MDB_ READ_COILS; registerları okumak için _MDB_ READ_HOLDING_REGISTERS kullanılır.

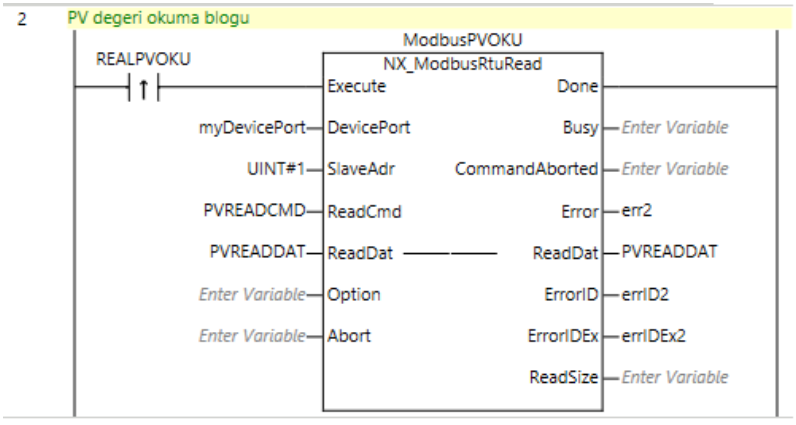
ReadAdr; veri okunacak adres yazılır.

ReadSize; okunan verinin büyüklüğüdür. Word veya word türünde dizi tanımlanabilir.

ReadCmd değişkenleri aşağıdaki gibi tanımlanabilir. İlk önce fonksyion sonrasında okunacak adres ve verinin büyüklüğü tanımlanmıştır.

```
12  //GERCEK SICAKLIK OKUMA
13
14  PVREADCMD.Fun:=_MDB_READ_HOLDING_REGISTERS;
15  PVREADCMD.ReadAdr:=16#0200;
16  PVREADCMD.ReadSize:=16#0001;
```

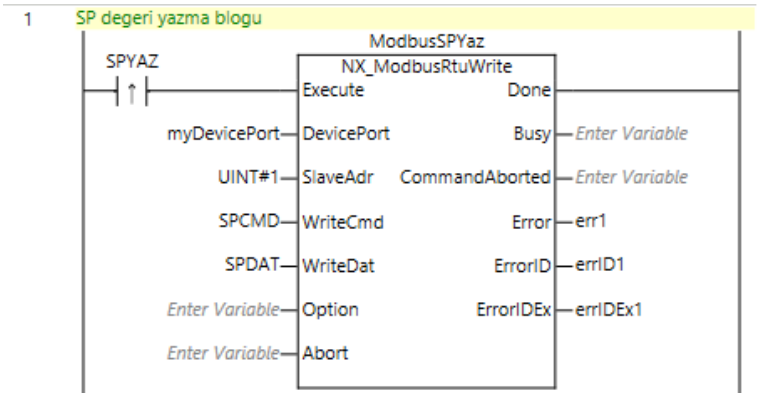
Değişkenler tanımlandıktan sonra ModbusRead komutları aşağıdaki gibi kullanılabilir:



NX_ModbusWrite bloğunda kullanılan WriteCmd komutu, struct yapıda bir değişkendir.

Fun; yazma fonksiyonu tanımlanır. Coil veya registerların okunması için _Mdb_WriteSingleCoil/ _Mdb_writeSingleRegister şeklinde tanımlanabilir. WriteAdr;

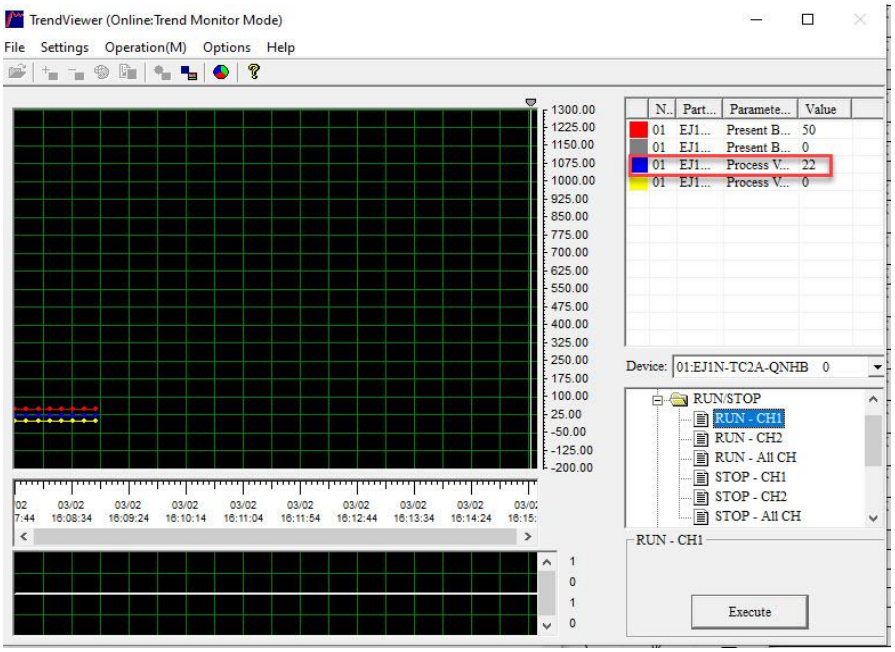
verinin yazılacağı adres girilir. WriteSize; yazılacak verinin büyüklüğü belirtilir. Kullanılan fonksyiona göre değişir, bir word veya word türünde bir dizi olarak tanımlanabilir.



Deneme :

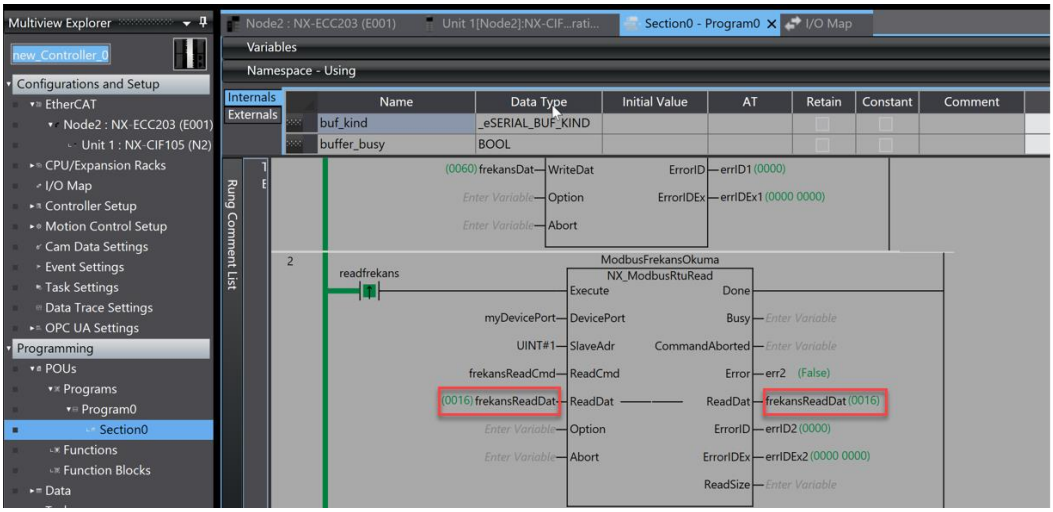
EJ1N de gerçek sıcaklık değeri(PV) okuma:

EJ1N ürününden değer okumak için ModbusRead bloğu kullanılmıştır. Blok içerisinde modbus adresi olarak 0200 değeri girilmiştir. Bu değer girildikten sonra, ortam sıcaklığı 22 derece olarak okunmuştur.



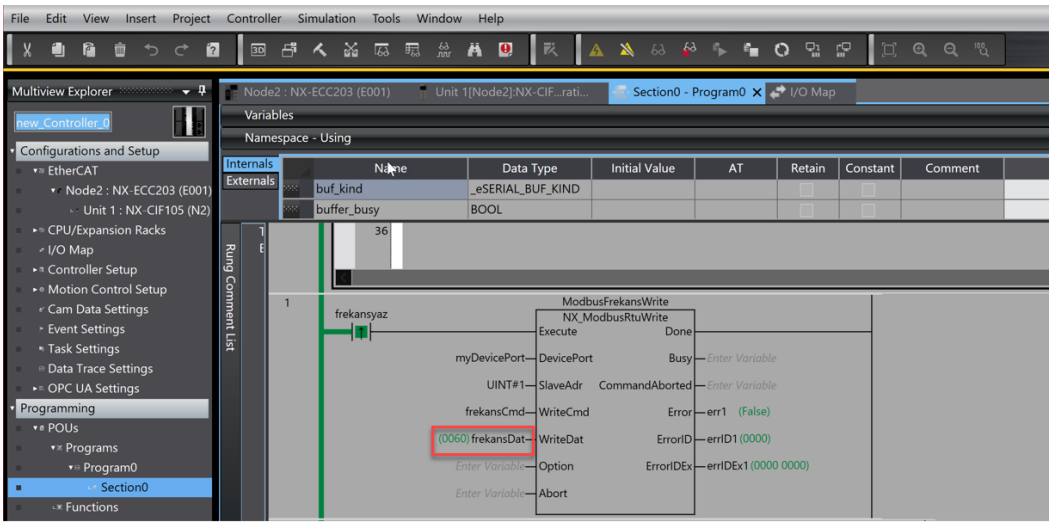
Sysmac Studio programında gerçek sıcaklık değeri(PV) okuma:

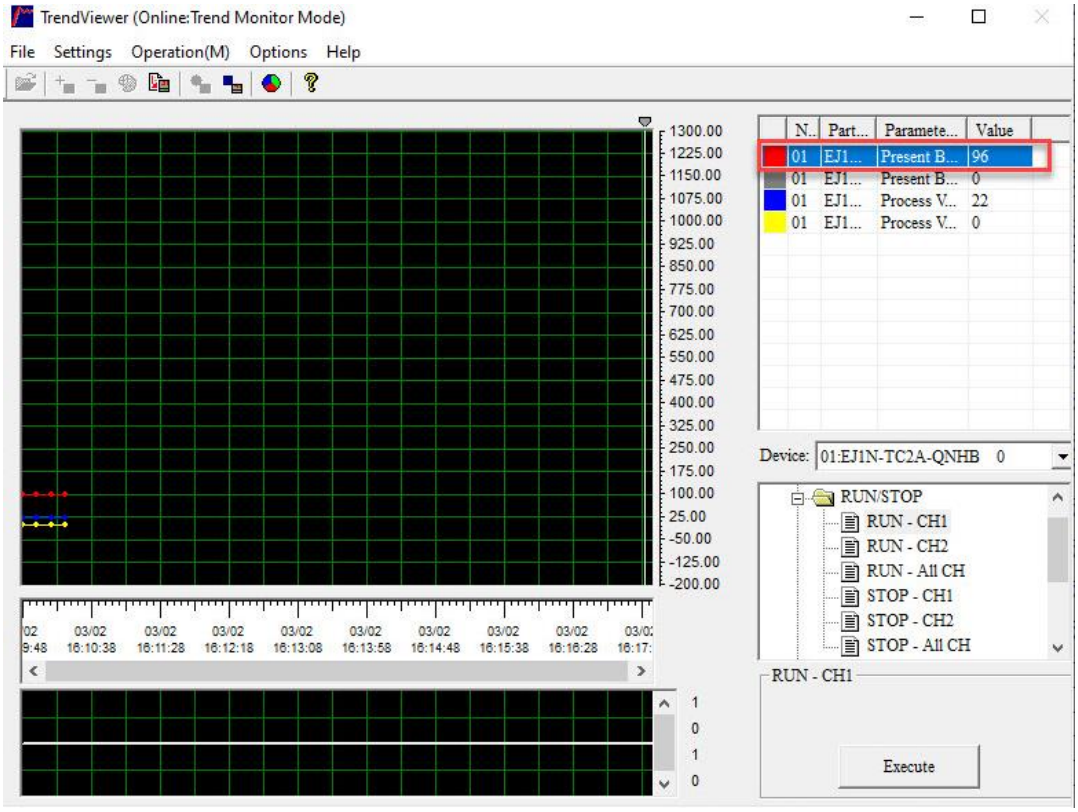
Sysmac Studio programında değer hexadecimal olarak gözükmektedir. Burada PV değeri 0016 olarak okunmuştur. Bu değer de decimal tipine çevrildiğinde 22 dereceye tekabül etmektedir:



Modbus üzerinden ulaşılması istenen sıcaklık değerini(SP) girme:

Modbus üzerinden ulaşılması istenen sıcaklık değerini(SP) girmek için, 0240 adresi kullanılır. Bu uygulamada 0240 adresine 0060 hexadecimal değeri yollanmıştır. Bu da decimal tipinde 96 dereceye tekabül eder. Aşağıdaki ekran görüntülerinde bu durum gösterilmiştir:




TrendViewer (Online:Trend Monitor Mode)
File Settings Operation(M) Options Help
1300.00
1225.00
1150.00
1075.00
1000.00
925.00
850.00
775.00
700.00
625.00
550.00
475.00
400.00
325.00
250.00
175.00
100.00
25.00
-50.00
-125.00
-200.00
N.. Part... Paramete... Value
01 EJ1... Present B... 96
01 EJ1... Present B... 0
01 EJ1... Process V... 22
01 EJ1... Process V... 0
Device: 01:EJ1N-TC2A-QNHB 0
RUN/STOP
RUN - CH1
RUN - CH2
RUN - All CH
STOP - CH1
STOP - CH2
STOP - All CH
RUN - CH1
Execute