Expanding
Possibility

Function Manual for Hi6 Controller

Hi6 제어기 Modbus 기능 설명서

HD HYUNDAI ROBOTICS

HD
HYUNDAI
ROBOTICS

목차

1. 개요

1.1 사전 필요정보

이 설명서를 이해하기 위해서는 다음과 같은 사전 정보가 필요합니다.

1. Hi6 로봇 제어기 조작 지식
2. Modbus 프로토콜 지식

1.2 Modbus 기능

Hi6 로봇 제어기는 시리얼 통신과 이더넷 통신에 의한 Modbus 마스터, 슬레이브 기능을 모두 지원합니다.

1. MODBUS master 운용 예

   - 장비 제어 :

     MODBUS 를 지원하는 장비(ex, Gripper)를 제어할 수 있습니다.

2. MODBUS slave 운용 예

- 조작반 기능 :

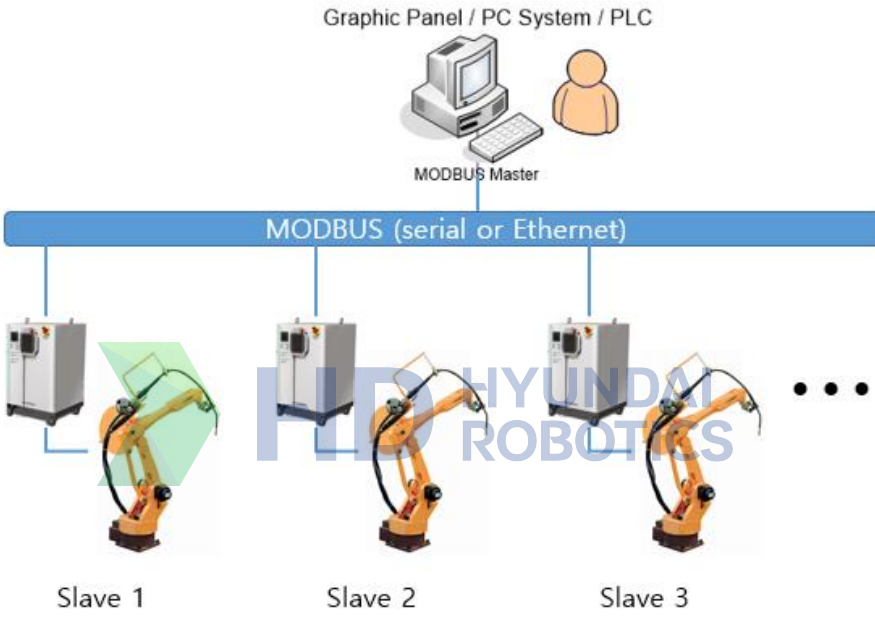
  MODBUS 를 지원하는 저렴한 GP(Graphic Panel)로 한대 혹은 여러 대의 로봇을 시리얼 또는 이더넷 통신으로 연결하여 사용할 수 있습니다.

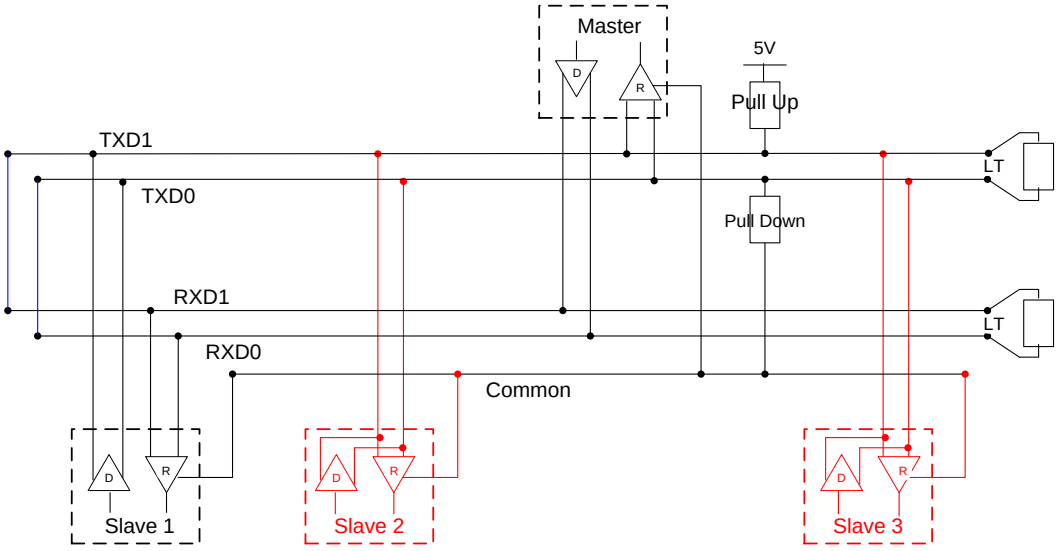
- PLC 통신 :

  MODBUS Master 기능을 갖는 PLC 들과의 통신을 저렴한 Solution 으로 제공합니다.

- PC 로봇운영 시스템 :

  PC 를 이용하여 로봇의 입출력 신호를 모니터링 하거나 제어하는 로봇 운영 시스템을 구축할 수 있습니다.





3. 지원 방식

	시리얼 통신	이더넷 통신
Master 운영	- 로봇언어 명령문	
Slave 운영	- 제어기 설정	- Ip : 제어기 설정 - Port : 502 (고정)

4. 전송 모드

	시리얼 통신	이더넷 통신
Master 운영	- binary 모드	
Slave 운영	- ASCII 모드 - RTU(binary) 모드	- binary 모드

5. 지원 펑션

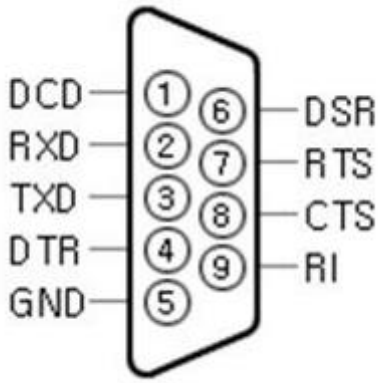
	시리얼, 이더넷 통신
Master 운영	- 03: read holding registers (multiple) - 16: write holding registers (multiple)
Slave 운영	- 01: read coils (bits) - 02: read discrete inputs (bits) - 03: read holding registers (multiple) - 04: read input registers (multiple) - 05: write single coil (bit) - 06: write single holding register - 15: write coils (multiple bits) - 16: write holding registers (multiple)

6. 슬레이브 주소

- 슬레이브 주소 : 1~247
- 명령어의 슬레이브 주소가 0 인 경우, 설정된 주소와 무관하게 모든 슬레이브가 동작하는 Broadcasting 기능을 지원합니다.

7. 시리얼 통신 연결

- 커넥터 (DSUB – 9pin : female)



DB-9 RS232 코넥터

- 핀맵

Suggested DB9 Connector Pinout

DB9 Pin	RS-232	RS-485/RS-422 Full Duplex	RS-485 Half Duplex
1	DCD	TX-	Data-
2	RXD	TX+	Data+
3	TXD	RX+	
4	DTR	RX-	
5	Ground		
6	DSR		
7	RTS		
8	CTS		
9	RI		

8. Address 맵

Modbus Data Model	Relay name	Relay mapping				Function
		1bit		16bit		
		Register	Logical Addr.	Register	Logical Addr	
Input Discrete Add: 0x0000~0xffff Quantity: 1~2039(bit) **Input Registers** Add: 0x0000~0xffff Quantity: 1~127	MW					Read Function 02: read discrete Inputs (bits) 04: read input registers (multiple)
	DO					
	DI	di0~959 … fb9.di0~959	12000~13999	diw0~118 … fb9.diw0~118	12000~13999	
	SO					
	SI	si0~959	15000~16999	siw0~118	15000~16999	
	SW					
Coils Add: 0x0000~0xffff Quantity: 1~2039(bit) **Holding Registers** Add: 0x0000~0xffff Quantity: 1~127	MW			_mw0~9999	0~9999	Read Function 01: read coils (bits) 03: read holding registers (multiple) Write Function 05: write single coil (bit) 15: write coils (multiple bits) 06: write single holding register 16: write holding registers (multiple)
	DO	do0~959 … fb9.do0~959	10000~11999	dow0~118 … fb9.dow0~118	10000~11999	
	DI	di0~959 … fb9.di0~959	12000~13999	diw0~118 … fb9.diw0~118	12000~13999	
	SO	so0~959	14000~14999	sow0~118	14000~14999	
	SI	si0~959	15000~16999	siw0~118	15000~16999	
	SW			_sw0~9999	16000~65535	

- 상기 표의 기울임 꼴 큰 숫자는 Modbus 에서 사용하는 relay 그룹임
- MW(data memory for user), DO(digital output), DI(digital input), SO(system output), SI(system input), SW(system memory)
- Data 형식: Float 형식은 IEEE single-precision 32bit float point 을 사용하고, 8bit/16bit/32bit 들은 전부 signed 정수를 사용함.
- Relay 의 Endian 은 Little Endian 을 사용함;

  예, Float 형식인 dof0=6.515625(0x40D08000)인 경우,

  dol0=0x40D08000 → dow0=0x8000, dow2=0x40D0 → dob0=0x00, dob1=0x80, dob2=0xD0, dob3=0x40

  참고) Modbus 전송은 16 bit align 의 Big Endian 임. 즉, 상기 전송은 0x80, 0x00, 0x40, 0xD0 의 순서로 전송됨

9. SW 메모리 맵 : 추가된 맵을 사용하기 위해서 당사에 요청하십시오.

릴레이	모드버스 주소 (0 based, decimal)	ProConOs shared memory		설 명	비 고
		data type	address		
PLC 관련					
SW0	16000	INT	%MW3.32000	PLC 실행 모드(0=On, 1=Holding, 2=Starting, 3=Running, 4= Haltrequested, 5=Halt, 6=Stopping, 7=Stop, 8=Resetting, Others=Unknown)	
소프트웨어 버전					
SW5	16005	INT	%MW3.32010	Main SW Version의 1st	60.01-02 -> 60
SW6	16006	INT	%MW3.32012	Main SW Version의 2nd	60.01-02 -> 01
SW7	16007	INT	%MW3.32014	Main SW Version의 3rd	60.01-02 -> 02
프로그램 카운터					
SW101	16101	INT	%MW3.32202	제어기의 현재 프로그램 번호	
SW102	16102	INT	%MW3.32204	제어기의 현재 스텝 번호	
SW103	16103	INT	%MW3.32206	제어기의 현재 평션 번호	
SW104	16104	INT	%MW3.32208	제어기의 메인 프로그램 번호	
통산 가동시간					
SW199	16199	INT	%MW3.32398	선택모드 0=무효 1=통산(초기화후) 2=통산(전원투입후) 3=마지막사이클 4=현재사이클	
SL200	16200	DINT	%MD3.32400	모터ON 일	
SL202	16202	DINT	%MD3.32404	모터ON 시간(ms 단위)	
SL204	16204	DINT	%MD3.32408	가동 일	
SL206	16206	DINT	%MD3.32412	가동 시간 (ms 단위)	
SL208	16208	DINT	%MD3.32416	이동 일	
SL210	16210	DINT	%MD3.32420	이동 시간 (ms 단위)	
SL212	16212	DINT	%MD3.32424	사이클 회수	
SL214	16214	DINT	%MD3.32428	Wait, DI대기 일	

SL216	16216	DINT	%MD3.32432	Wait, DI대기 시간 (ms 단위)	
SL218	16218	DINT	%MD3.32436	타이머 대기 일	
SL220	16220	DINT	%MD3.32440	타이머 대기 시간 (ms 단위)	
로봇 위치					
SF300	16240	REAL	%MD3.32600	베이스좌표값 X(단위: mm)	
SF302	16242	REAL	%MD3.32604	베이스좌표값 Y(단위: mm)	
SF304	16244	REAL	%MD3.32608	베이스좌표값 Z(단위: mm)	
SF306	16246	REAL	%MD3.32612	베이스좌표값 RX(단위: deg)	
SF308	16248	REAL	%MD3.32616	베이스좌표값 RY(단위: deg)	
SF310	16250	REAL	%MD3.32620	베이스좌표값 RZ(단위: deg)	
SF312	16252	REAL	%MD3.32624	1축 위치(단위: mm or deg)	
SF314	16254	REAL	%MD3.32628	2축 위치(단위: mm or deg)	
SF316	16256	REAL	%MD3.32632	3축 위치(단위: mm or deg)	
SF318	16258	REAL	%MD3.32636	4축 위치(단위: mm or deg)	
SF320	16260	REAL	%MD3.32640	5축 위치(단위: mm or deg)	
SF322	16262	REAL	%MD3.32644	6축 위치(단위: mm or deg)	
SF324	16264	REAL	%MD3.32648	7축 위치(단위: mm or deg)	
SF326	16266	REAL	%MD3.32652	8축 위치(단위: mm or deg)	
SF328	16268	REAL	%MD3.32656	9축 위치(단위: mm or deg)	
SF330	16270	REAL	%MD3.32660	10축 위치(단위: mm or deg)	
SF332	16272	REAL	%MD3.32664	11축 위치(단위: mm or deg)	
SF334	16274	REAL	%MD3.32668	12축 위치(단위: mm or deg)	
SF336	16276	REAL	%MD3.32672	13축 위치(단위: mm or deg)	
SF338	16278	REAL	%MD3.32676	14축 위치(단위: mm or deg)	
SF340	16280	REAL	%MD3.32680	15축 위치(단위: mm or deg)	
SF342	16282	REAL	%MD3.32684	16축 위치(단위: mm or deg)	
로봇 속도					
SW349	16349	INT	%MW3.32698	선택모드 0=무효 1=축 속도(단위:mm/s or deg/s) 2=모터속도(단위:rpm)	
SF350	16350	REAL	%MD3.32700	1축 속도(단위: mm/s or deg/s)	
SF352	16352	REAL	%MD3.32704	2축 속도(단위: mm/s or deg/s)	
SF354	16354	REAL	%MD3.32708	3축 속도(단위: mm/s or deg/s)	
SF356	16356	REAL	%MD3.32712	4축 속도(단위: mm/s or deg/s)	
SF358	16358	REAL	%MD3.32716	5축 속도(단위: mm/s or deg/s)	
SF360	16360	REAL	%MD3.32720	6축 속도(단위: mm/s or deg/s)	
SF362	16362	REAL	%MD3.32724	7축 속도(단위: mm/s or deg/s)	

SF364	16364	REAL	%MD3.32728	8축 속도(단위: mm/s or deg/s)	
SF366	16366	REAL	%MD3.32732	9축 속도(단위: mm/s or deg/s)	
SF368	16368	REAL	%MD3.32736	10축 속도(단위: mm/s or deg/s)	
SF370	16370	REAL	%MD3.32740	11축 속도(단위: mm/s or deg/s)	
SF372	16372	REAL	%MD3.32744	12축 속도(단위: mm/s or deg/s)	
SF374	16374	REAL	%MD3.32748	13축 속도(단위: mm/s or deg/s)	
SF376	16376	REAL	%MD3.32752	14축 속도(단위: mm/s or deg/s)	
SF378	16378	REAL	%MD3.32756	15축 속도(단위: mm/s or deg/s)	
SF380	16380	REAL	%MD3.32760	16축 속도(단위: mm/s or deg/s)	

로봇 부하율

SW399	16399	INT	%MW3.32798	선택 모드 0=무효, 1=I/Ir, 2=I/Ip, 3=연속	
SF400	16400	REAL	%MD3.32800	1축 부하율	
SF402	16402	REAL	%MD3.32804	2축 부하율	
SF404	16404	REAL	%MD3.32808	3축 부하율	
SF406	16406	REAL	%MD3.32812	4축 부하율	
SF408	16408	REAL	%MD3.32816	5축 부하율	
SF410	16410	REAL	%MD3.32820	6축 부하율	
SF412	16412	REAL	%MD3.32824	7축 부하율	
SF414	16414	REAL	%MD3.32828	8축 부하율	
SF416	16416	REAL	%MD3.32832	9축 부하율	
SF418	16418	REAL	%MD3.32836	10축 부하율	
SF420	16420	REAL	%MD3.32840	11축 부하율	
SF422	16422	REAL	%MD3.32844	12축 부하율	
SF424	16424	REAL	%MD3.32848	13축 부하율	
SF426	16426	REAL	%MD3.32852	14축 부하율	
SF428	16428	REAL	%MD3.32856	15축 부하율	
SF430	16430	REAL	%MD3.32860	16축 부하율	

컨베이어 동기

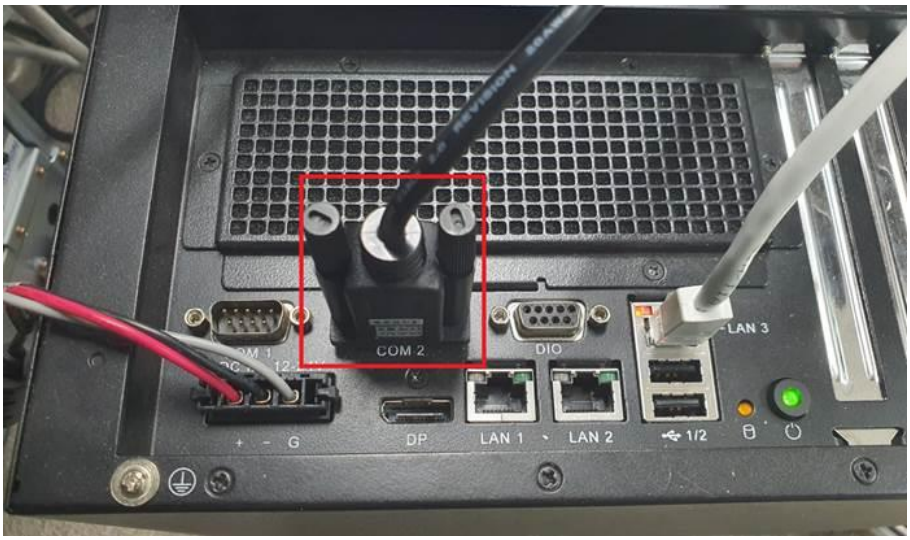
SW2200	18200	INT	%MW3.36400	펄스데이터 (채널1)	
SW2201	18201	INT	%MW3.36402	작업물 위치 (채널1)	
SW2202	18202	INT	%MW3.36404	이동속도 (채널1)	
SW2203	18203	INT	%MW3.36406	작업물 진입 수 (채널1)	
SW2204	18204	INT	%MW3.36408	리밋스위치 입력 (채널1)	
SW2205	18205	INT	%MW3.36410	raw 펄스값 (채널1)	
SW2210	18210	INT	%MW3.36420	펄스데이터 (채널2)	
SW2211	18211	INT	%MW3.36422	작업물 위치 (채널2)	

SW2212	18212	INT	%MW3.36424	이동속도 (채널2)	
SW2213	18213	INT	%MW3.36426	작업물 진입 수 (채널2)	
SW2214	18214	INT	%MW3.36428	리밋스위치 입력 (채널2)	
SW2215	18215	INT	%MW3.36430	raw 펄스값 (채널2)	

2. 시리얼 통신 설정

2.1 시리얼 케이블 연결

시리얼 케이블은 하기의 그림과 같이 COM2 포트에 직접 연결합니다.

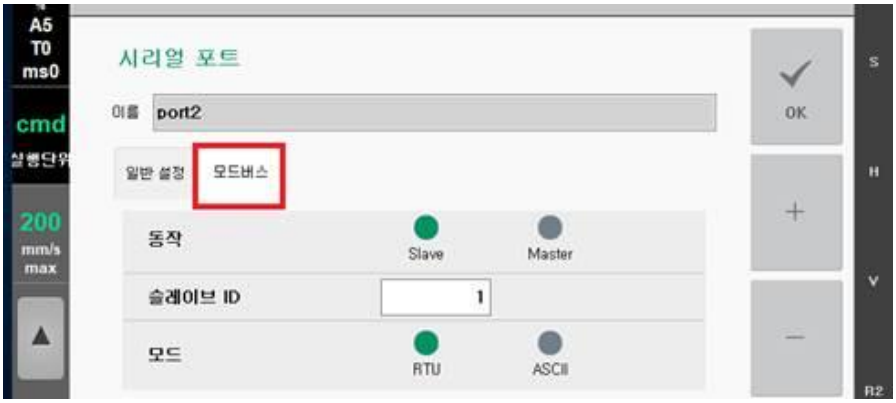


2.2 시리얼 포트 용도 설정

시리얼 포트의 용도를 MODBUS 로 설정하는 것은 『설정 → 2: 제어 파라미터 → 3: 시리얼 포트』 화면의 『일반 설정』 탭에서 다음과 같이 설정할 수 있습니다.

2.3 모드버스 환경 설정

모드버스의 세부사항은 『모드버스』 탭에서 다음과 같이 설정할 수 있습니다.



- 동작 : Master 로 운영할지, Slave 로 운영할지 선택합니다. Master 로 운영시 로봇언어 명령어에 의해 수행되기 때문에 Slave id 와 Mode 는 사용하지 않습니다
- 슬레이브 ID : 모드버스 시리얼의 slave 로 통신하기 위한 ID 를 설정합니다.
- 모드 : 모드버스 시리얼의 slave 로 통신하기 mode 를 설정합니다.

3. 마스터 운영

3.1 로봇 언어

로봇언어 명령문을 사용하여 모드버스 마스터 쿼리를 구성하고 이를 슬래이브에 전송할 수 있습니다. 해당 명령문을 실행할 때 데이터 송수신이 이루어지며 만약 주기적으로 데이터를 송수신하는 것이 필요하다면 이를 위한 작업 프로그램을 구성한 후 멀티 태스크로 구동할 수도 있겠지만 이 경우는 내장 PLC 를 통하여 통신하는 것을 권장합니다.

3.1.1 명령어 (Modbus)

설명	모드버스 마스트 통신을 위한 명령문입니다.		
문법	modbus enet2,sid=65,fc=16,addr=0,len=3,wait=3,var=arr		
파라미터	enet2	통신 객체 (이더넷 또는 시리얼)	
	sid	슬래이브 id	1 ~ 255
	fc	펑션 코드 03 = read holding register(multiple) 16 = write multiple register	03, 16
	addr	시작 주소	0 ~ 65534
	len	데이터 길이	1 ~ 127
	wait	통신 대기시간	
	var	데이터 송수신을 위한 배열 변수 (미지정시 자체 모드버스 맵이 사용됨)	
세부 설명	• 로봇 언어로 모드버스 마스트 통신을 하는 명령문 입니다. • 모드버스 통신에 대한 이해를 위해서 별도로 학습하시기 바랍니다.		

3.1.2 샘플 프로그램

3.1.2.1 이더넷 통신

하기는 onRobot gripper를 제어하기 위한 샘플 프로그램으로 Hi6 제어기와 onRobot gripper는 Modbus tcp로 통신합니다. 여기서 hi6 제어기는 master로 운영되며 gripper가 slave로 운영됩니다.

0060.job

```
Hyundai Robot Job File; { version: 1.6, mech_type: "780(YL012-0D)", total_axis: 6, aux_axis: 0 }
call 61,1 # onRobot module open
call 61,2,0 # onRobot gripper hold
delay(3)
call 61,2,300 # onRobot gripper release
call 61,0 # onRobot module close
delay(3)
end
```

0061.job

```
Hyundai Robot Job File; { version: 1.6, mech_type: "780(YL012-0D)", total_axis: 6, aux_axis: 0 }
param mode,grip
if (mode == 1) # enet module open
  import enet
  global enet2,arr
  if (arr==0)
    arr=Array(5)
  endif
  # onRobot gripper enet connect
  enet2=enet.ENet("tcp") #udp,tcp
  enet2.ip_addr="192.168.1.111" #OnRonot IP
  enet2.lport=502
  enet2.rport=502
  if (enet2.state() < 1)
    enet2.open
    enet2.connect #tcp인 경우
  else
    stop
  endif
  print "enet2.state", enet2.state()
elseif (mode == 0) # enet module close
  enet2.close
else # onRobot gripper operate
```

```
  arr[0] = 300 # force (0~400)
  arr[1] = grip # width (0~1100)
  arr[2] = 1 # control (1:grip, 8=stop, 16=offset grip)
  modbus enet2,sid=65,fc=16,addr=0,len=3,wait=3,var=arr
endif
end
```

3.1.2.2 시리얼 통신

하기는 onRobot gripper를 시리얼 통신으로 제어한다고 가정하였을 때 샘플 프로그램입니다. 먼저 시리얼 통신 설정에서 포트 용도를 〈MODBUS〉로 모드버스의Operation은 〈master〉로 설정되어야 합니다.

0060.job

```
Hyundai Robot Job File; { version: 1.6, mech_type: "780(YL012-0D)", total_axis: 6, aux_axis: 0 }
call 61,1 # onRobot module open
call 61,2,0 # onRobot gripper hold
delay(3)
call 61,2,300 # onRobot gripper release
call 61,0 # onRobot module close
delay(3)
end
```


0061.job

```
Hyundai Robot Job File; { version: 1.6, mech_type: "780(YL012-0D)", total_axis: 6, aux_axis: 0 }
param mode,grip
if (mode == 1) # enet module open
  global sci2,arr
  if (arr==0)
    arr=Array(5)
  endif
  # onRobot gripper enet connect
  sci2=com.Sci(2) # serial port 2 object
elseif (mode == 0) # enet module close
  print “sci close”
else # onRobot gripper operate
  arr[0] = 300 # force (0~400)
  arr[1] = grip # width (0~1100)
  arr[2] = 1 # control (1:grip, 8=stop, 16=offset grip)
  modbus sci2,sid=65,fc=16,addr=0,len=3,wait=3,var=arr
endif
```

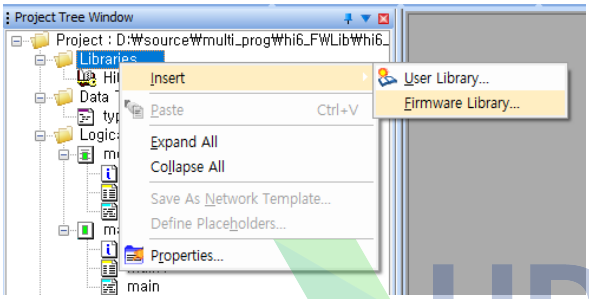
```
end
```

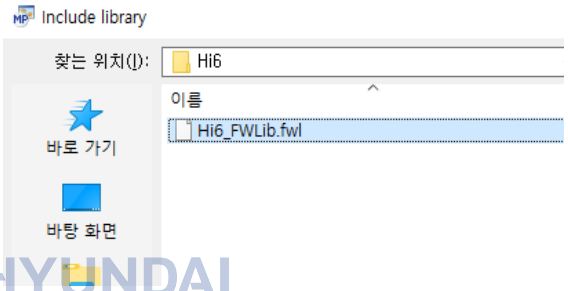
3.2 내장 PLC

내장 PLC 래더 로직에 의해 모드버스 마스터 쿼리를 구성하고 이를 슬래이브에 전송할 수 있습니다. 이를 사용하기 위해서는 내장 PLC 에 대한 지식이 필요하며 관련 설명서를 참고하십시오.

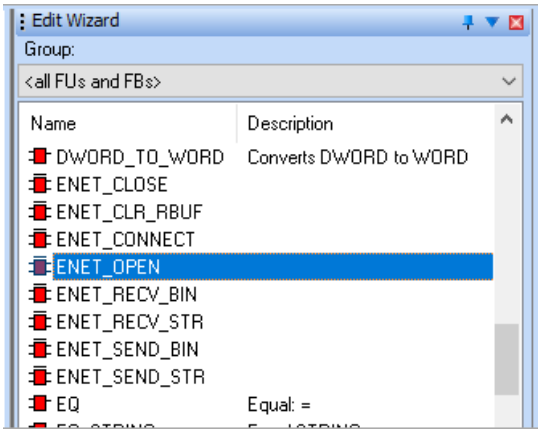
3.2.1 Native firmware library 추가

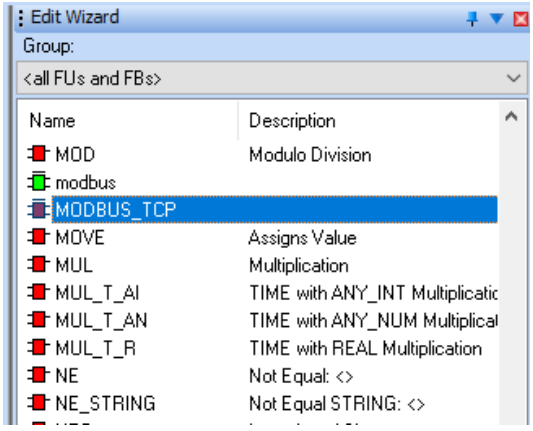
하기의 그림과 같이 Libraries 〉 Insert 〉 Firmware Library… 에서 "Hi6_FWLib.fwl" 파일을 선택하여 추가합니다.





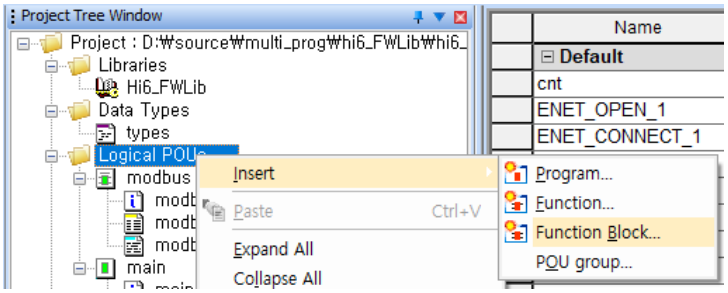
Edit Wizard에서 ENET 통신과 관련된 펑션 블록들과 MODBUS_TCP와 관련된 펑션 블록이 추가된 것을 확인할 수 있습니다.



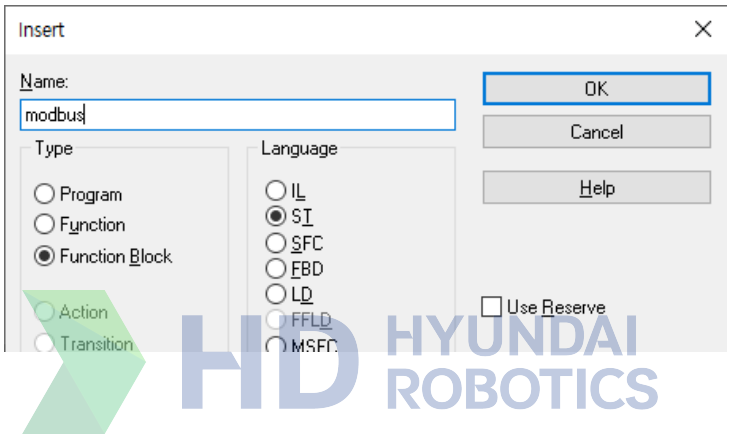


3.2.2 평션 블록 추가

하기의 그림과 같이 Logical POUs 〉 Insert 〉 Function Block... 에서 Function Block을 추가합니다.

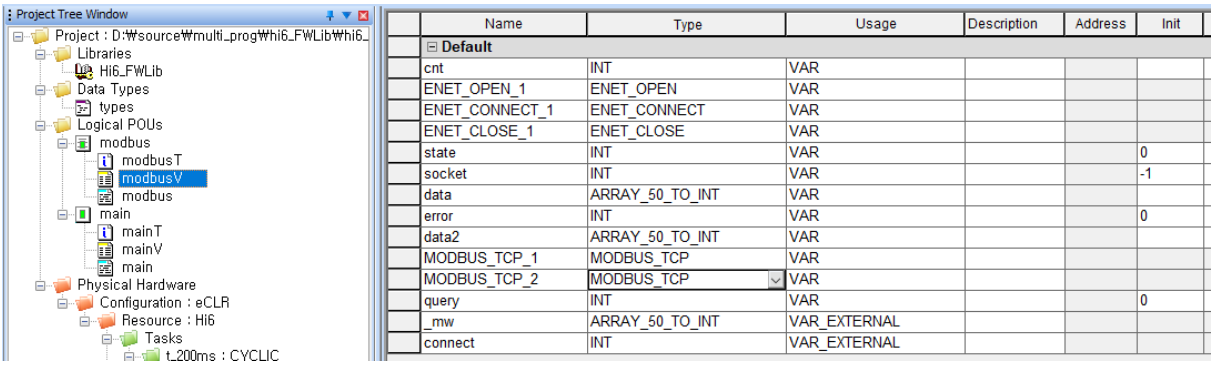


본 예제에서는 modbus라는 이름으로 ST언어를 사용하겠다고 선택하였습니다.



3.2.3 변수 추가

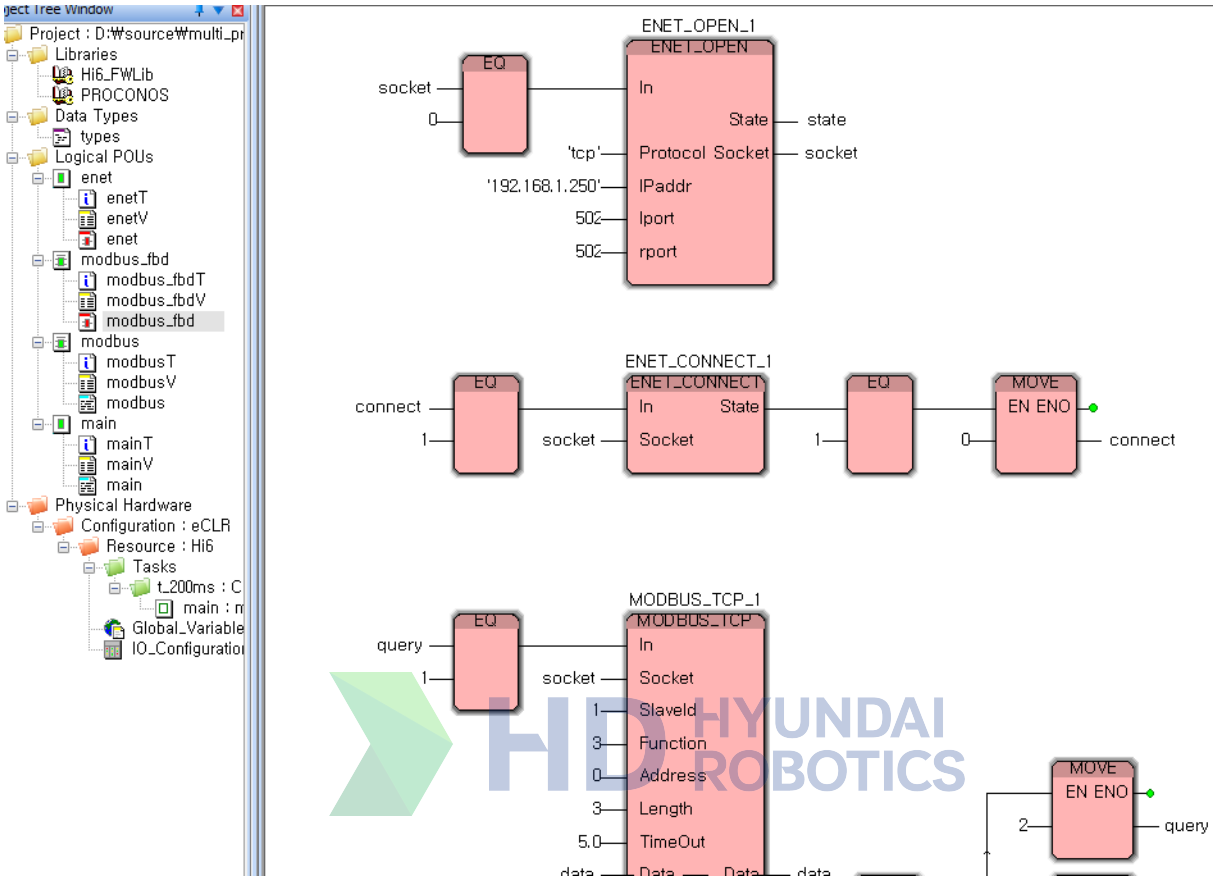
하기의 그림과 같이 변수 관리 화면에서 변수를 등록합니다.



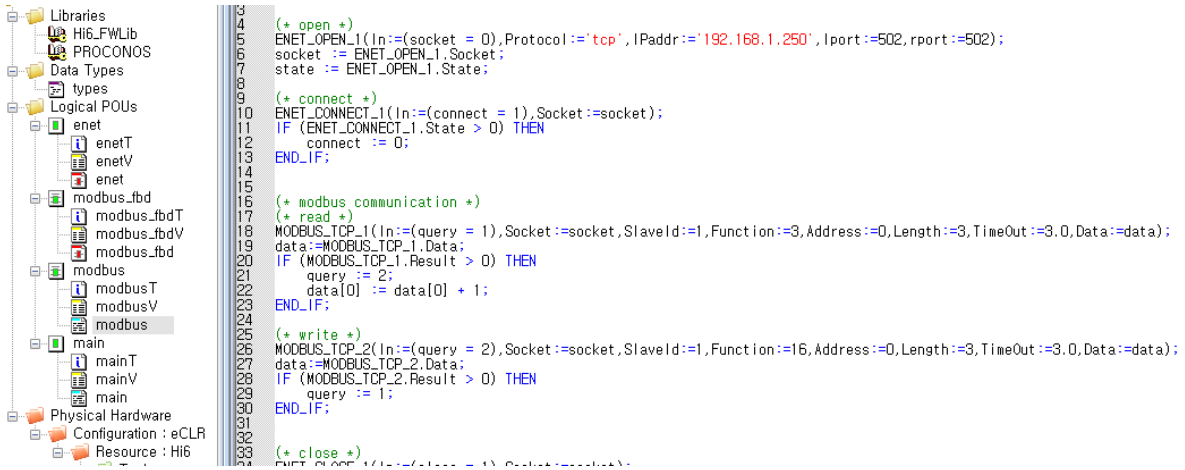
3.2.4 평션 블록 작성

하기의 그림과 같이 프로그램 작성 화면에서 사용자는 프로그램을 작성합니다.

(FBD 언어)

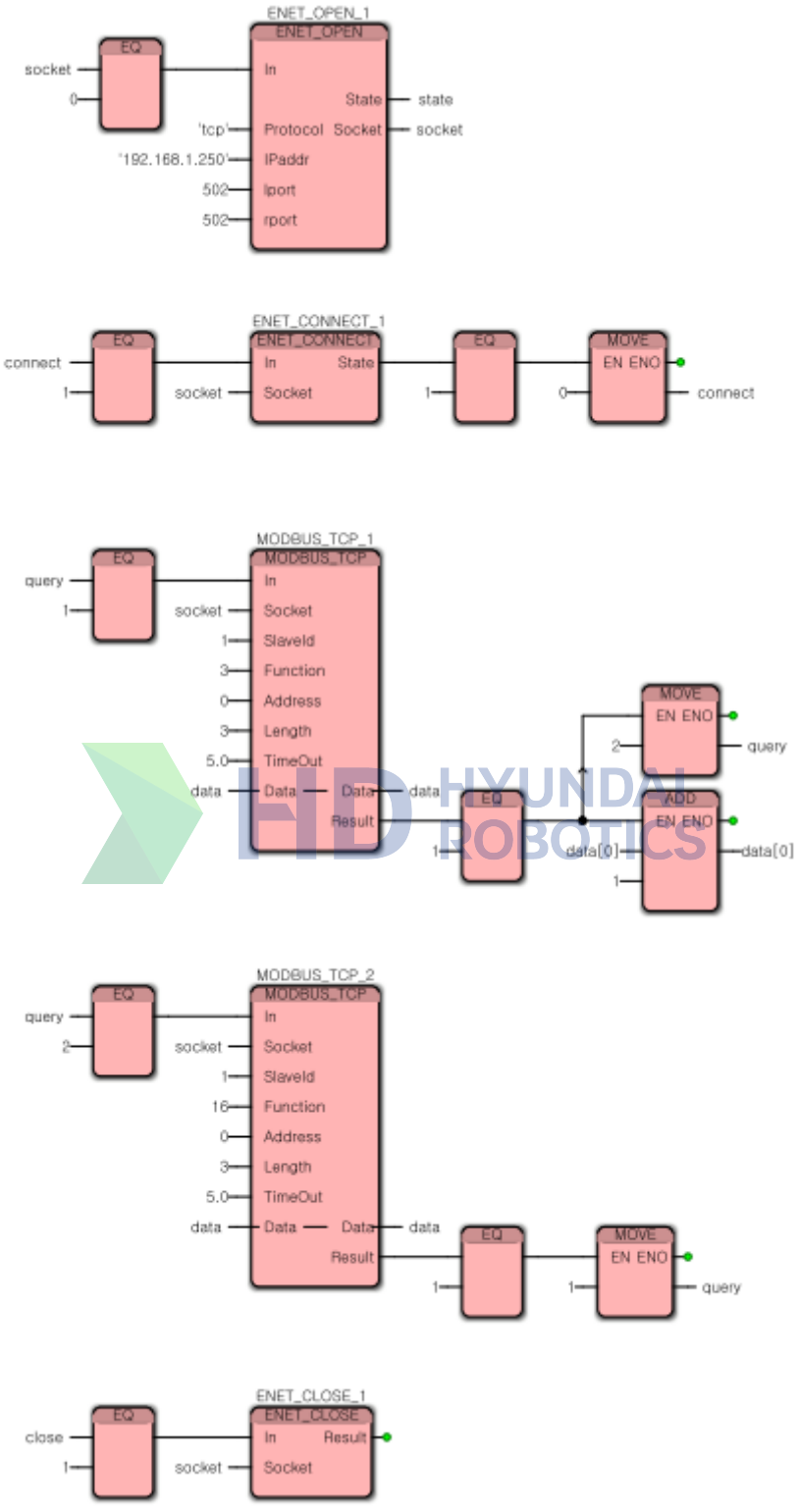


(ST언어)



3.2.5 샘플 펑션 블록

(FBD 언어)

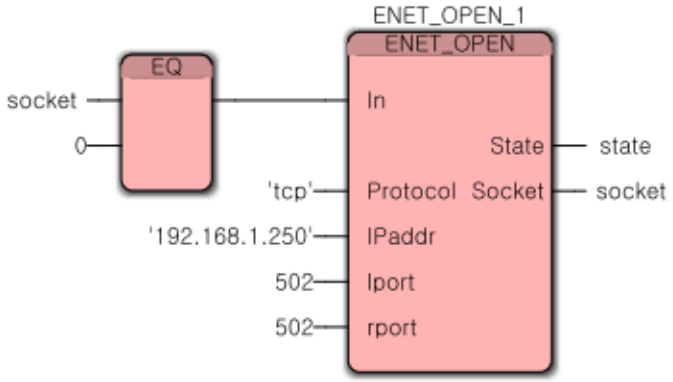


(ST 언어)

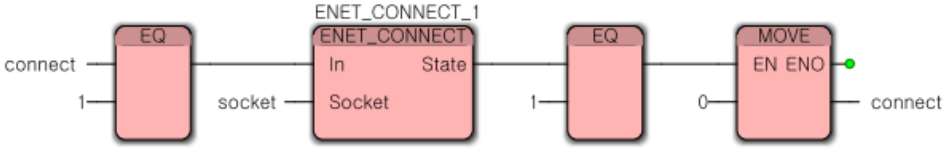
```
(* open *)
ENET_OPEN_1(In:=(socket = 0),Protocol:='tcp',IPaddr:='192.168.1.250',Iport:=502,rport:=502);
socket := ENET_OPEN_1.Socket;
state := ENET_OPEN_1.State;

(* connect *)
ENET_CONNECT_1(In:=(connect = 1),Socket:=socket);
IF (ENET_CONNECT_1.State > 0) THEN
    connect := 0;
END_IF;


(* modbus communication *)
(* read *)
MODBUS_TCP_1(In:=(query = 1),Socket:=socket,SlaveId:=1,Function:=3,Address:=0,Length:=3,TimeOut:=3,0
,Data:=data);
data:=MODBUS_TCP_1.Data;
IF (MODBUS_TCP_1.Result > 0) THEN
    query := 2;
    data[0] := data[0] + 1;
END_IF;

(* write *)
MODBUS_TCP_2(In:=(query = 2),Socket:=socket,SlaveId:=1,Function:=16,Address:=0,Length:=3,TimeOut:=3,
0,Data:=data);
data:=MODBUS_TCP_2.Data;
IF (MODBUS_TCP_2.Result > 0) THEN
    query := 1;
END_IF;


(* close *)
ENET_CLOSE_1(In:=(close = 1),Socket:=socket);
```


3.2.6 샘플 펑션 블록 설명

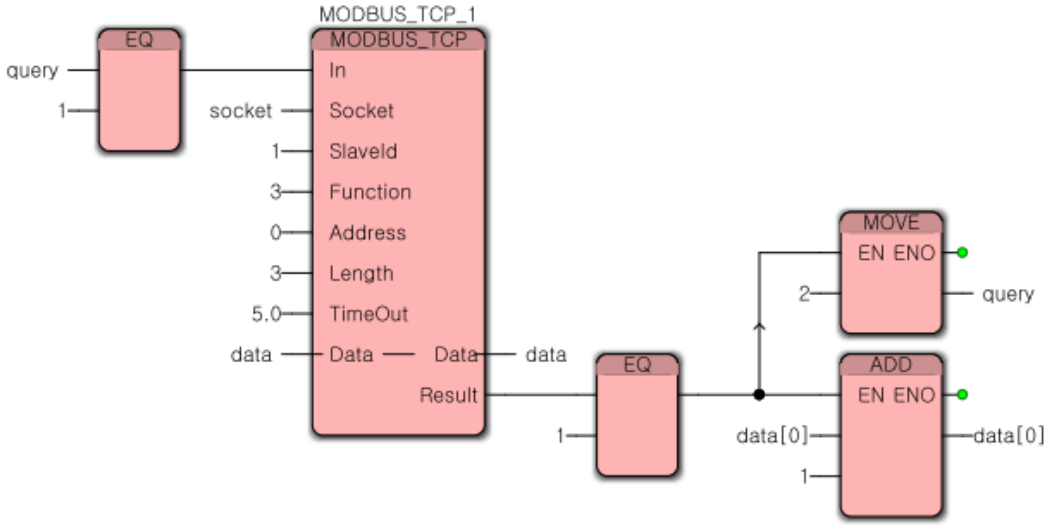
- PLC 를 RUN 하면 socket 변수가 0 으로 초기화 되어 있어 enet 통신을 위한 소켓을 자동으로 open 합니다.
- IPaddr 은 연결할 상대 장치의 ip address 를 지정합니다.



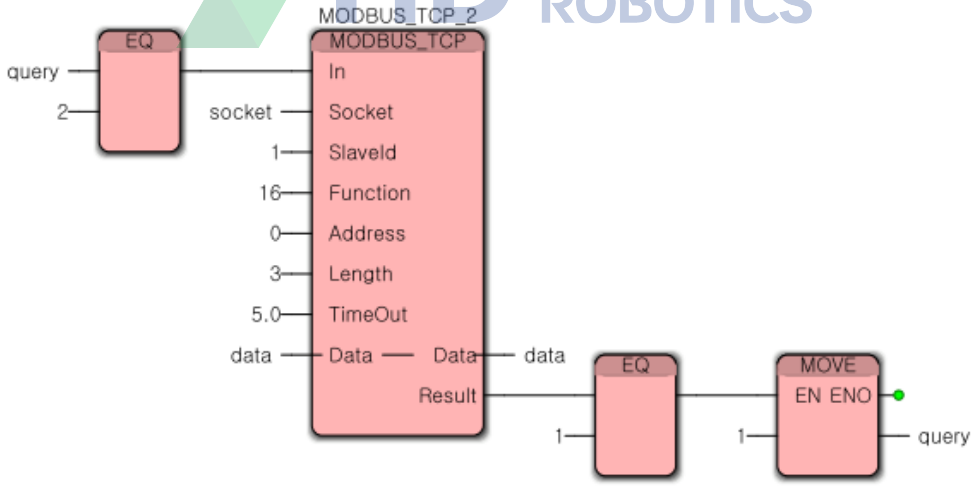
- connect 변수를 1 로 설정하면 슬레이브 장치와 연결 동작을 수행한 후 connect 변수를 0 으로 변경합니다.



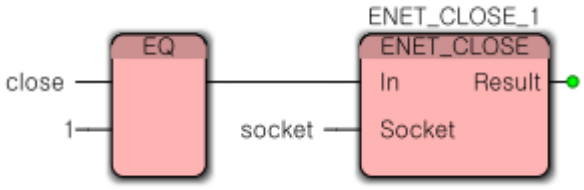
- query 변수를 1 로 설정하면 Function:=3,Address:=0,Length:=3 에 의해 슬래이브에서 0 번지부터 3 개의 데이터를 얻어와 data 의 배열 변수로 전달합니다. (read)
- Result 가 1 이면 query 를 2 로 설정하고 data[0]의 변수값을 1 증가합니다.



- query 변수가 2 이면 Function:=16,Address:=0,Length:=3 에 의해 data 배열 변수에 설정된 값을 슬래이브의 0 번지부터 3 개의 데이터를 설정합니다. (write)
- Result 가 1 이면 query 를 1 로 설정합니다.

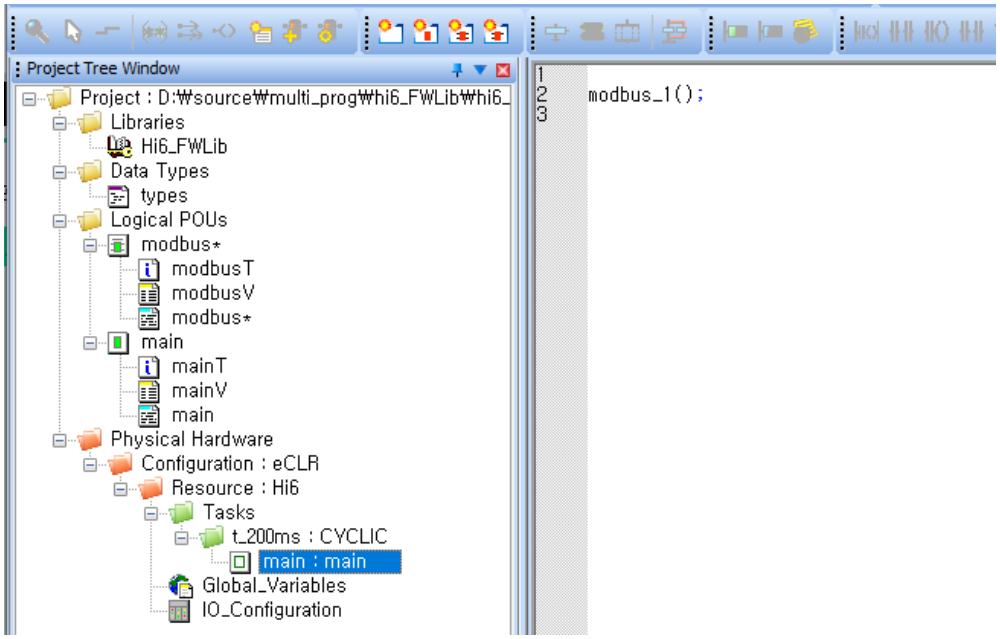


- close 변수를 1 로 설정하면 소켓을 닫게 됩니다.



3.2.7 샘플 펑션 블록 가동

- main 이란 Program POU 에서 샘플로 작성한 펑션 블록을 호출하여 실행합니다.
- main 이란 Program POU 는 CYCLIC task 에서 200ms 마다 실행합니다.



고객 지원

대표 번호: 1670-5041 | 이메일: robotics@hyundai-robotics.com

이용 시간: 평일(월 ~ 금) 09:00 ~ 18:00 | 주말 및 공휴일 휴무

제품이나 서비스에 대한 자세한 문의는 당사의 고객지원팀으로 연락하시기 바랍니다.

GRC: 경기도 성남시 분당구 분당수서로 477

대구: 대구광역시 달성군 유가읍 테크노순환로 3 길 50

울산: 울산광역시 북구 매곡산업로 21 자동차조선기술관 201-5 호

중부: 충남 아산시 염치읍 송곡길 161

광주: 광주광역시 광산구 평동산단로 170-3 B 동 101 호

ARS 1588-9997 | 1 로봇영업 2 서비스영업 3 구매상담 4 고객지원 5 투자문의 6 채용 및 일반 문의

www.hyundai-robotics.com