

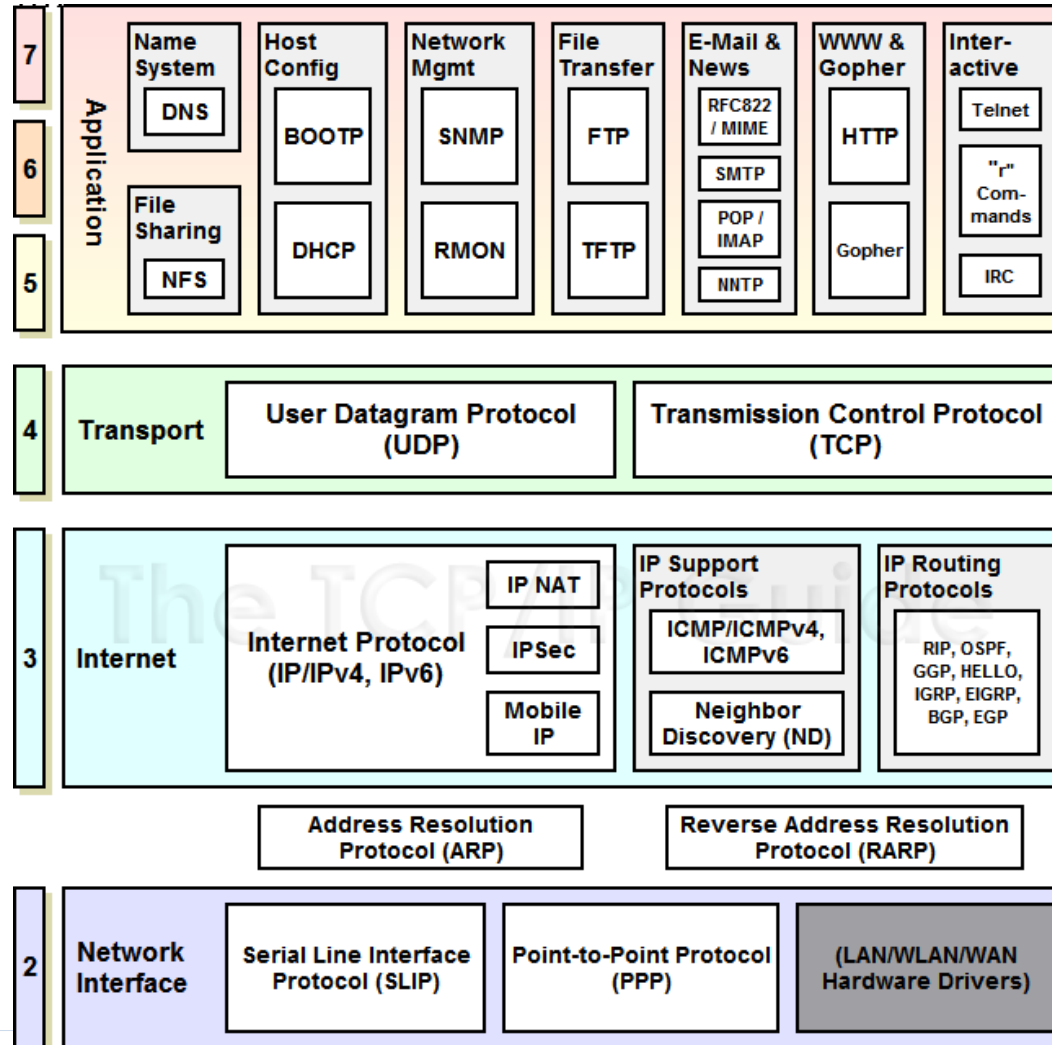
산업 기반 시설 보안 과정 2주차

보안프로젝트 조정원 대표
(ngnicky@naver.com)

산업 제어 시스템 프로토콜 이해 및 분석

산업 제어 시스템 프로토콜 이해 및 분석

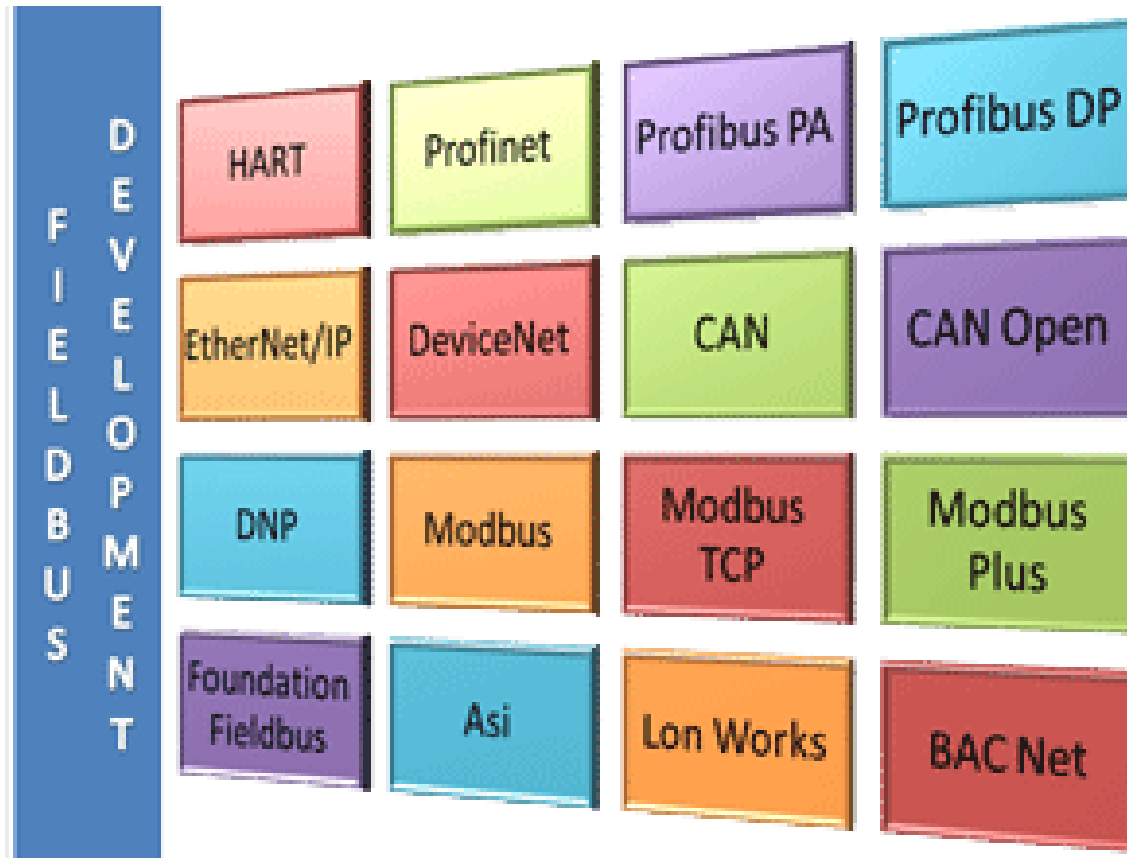
▶ 일반 IT 프로토콜 이해



산업 제어 시스템 프로토콜 이해 및 분석

▶ 일반 IT 프로토콜 이해

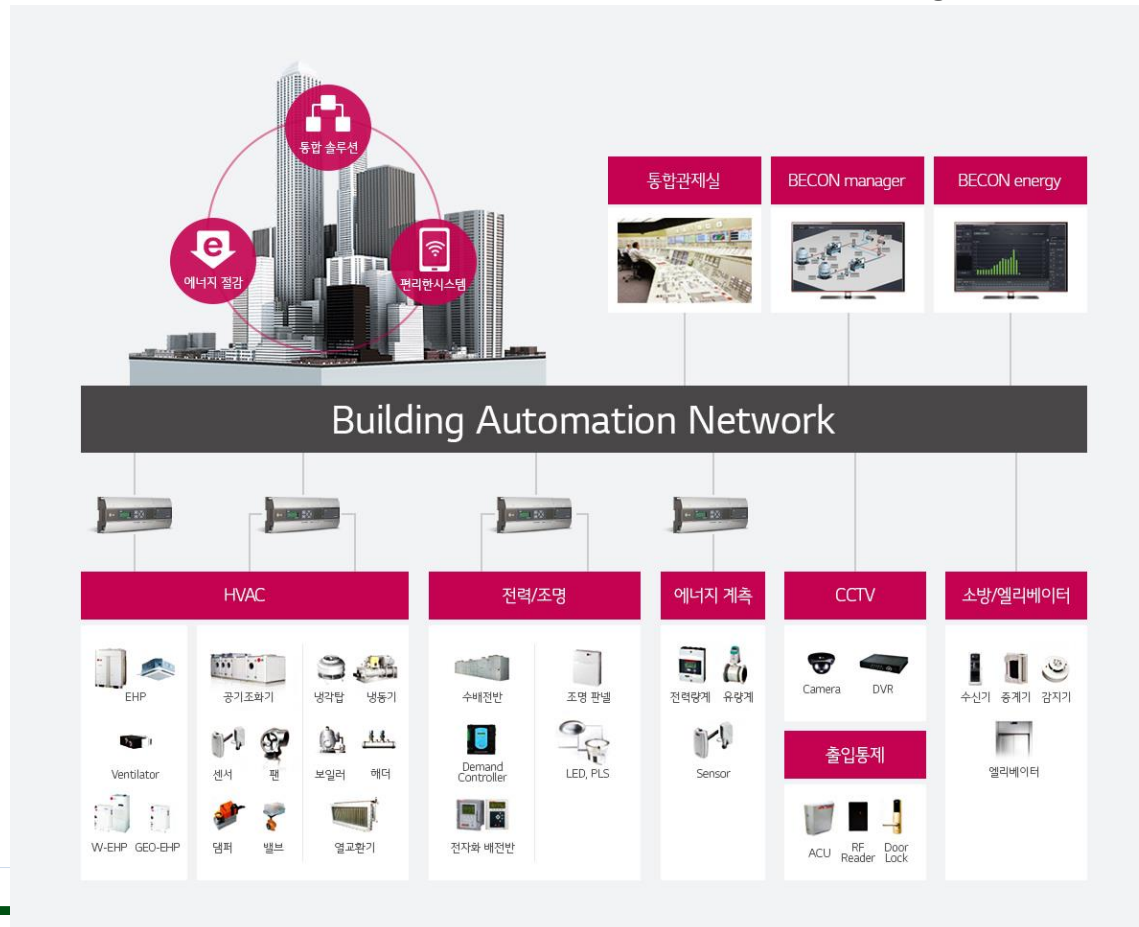
- 프로세스 자동화 프로토콜에는 PROFIBUS, DeviceNet, ControlNet, Modbus 및 CIP가 포함
- 이러한 프로토콜은 대부분 Purdue 모델의 레벨 3 이하에서 진행



산업 제어 시스템 프로토콜 이해 및 분석

▶ 일반 IT 프로토콜 이해

- 빌딩 자동화 프로토콜을 사용하면 난방, 환기 및 에어컨과 같은 응용 프로그램을 실행하는 제어 시스템의 부분들 간의 통신이 가능
- 이 범주 에서 사용되는 프로토콜에는 BACnet, C-Bus, Modbus, ZigBee 및 Z-Wave 등이 포함



산업 제어 시스템 프로토콜 이해 및 분석

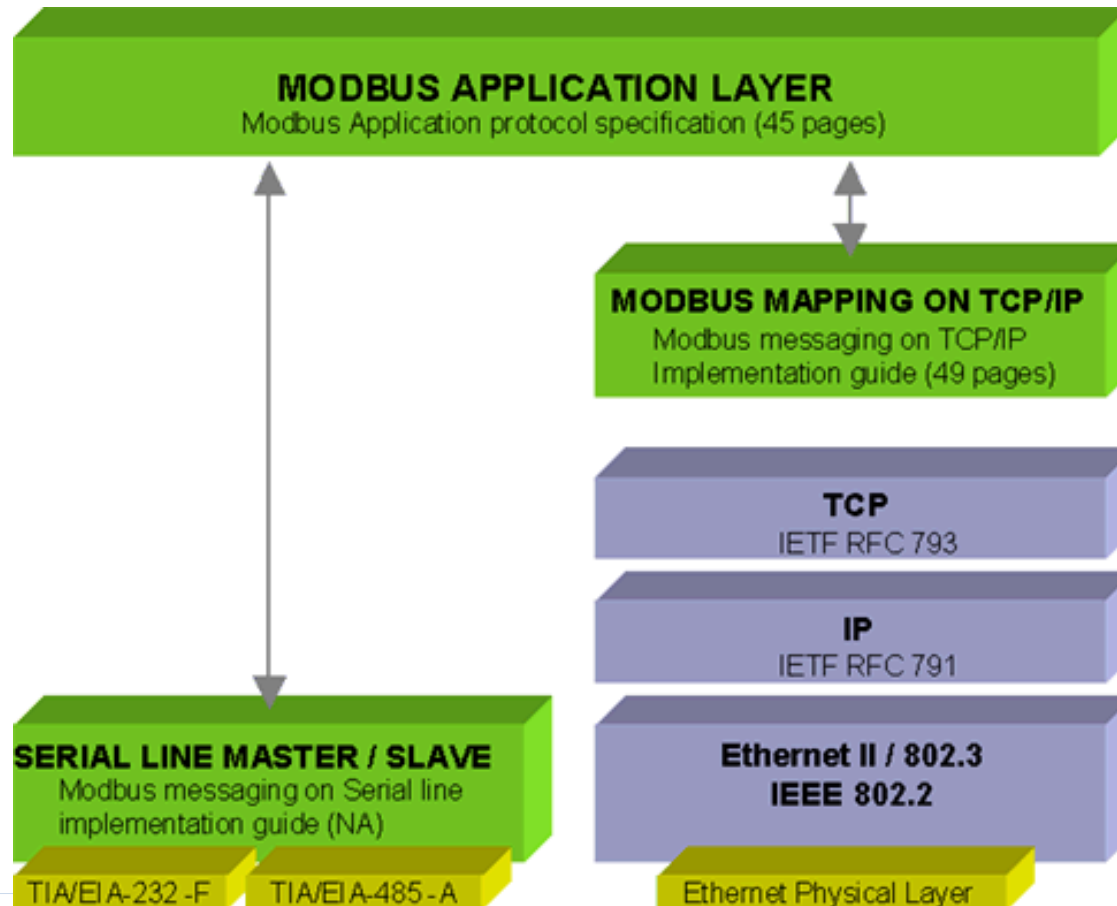
▶▶ Modbus 프로토콜 이해

- 1979 년에 도입 된 Modbus는 그 이후로 사실상의 표준처럼 사용되고 있고, 가장 널리 사용되는 ICS 프로토콜
- Modbus는 응용 프로그램 계층 메시징 프로토콜
- OSI 모델의 레벨 7에 배치되어, 다양한 유형의 통신 버스 또는 통신 매체를 통해 연결된 장치간에 클라이언트 / 서버 통신을 제공

산업 제어 시스템 프로토콜 이해 및 분석

Modbus 프로토콜 이해

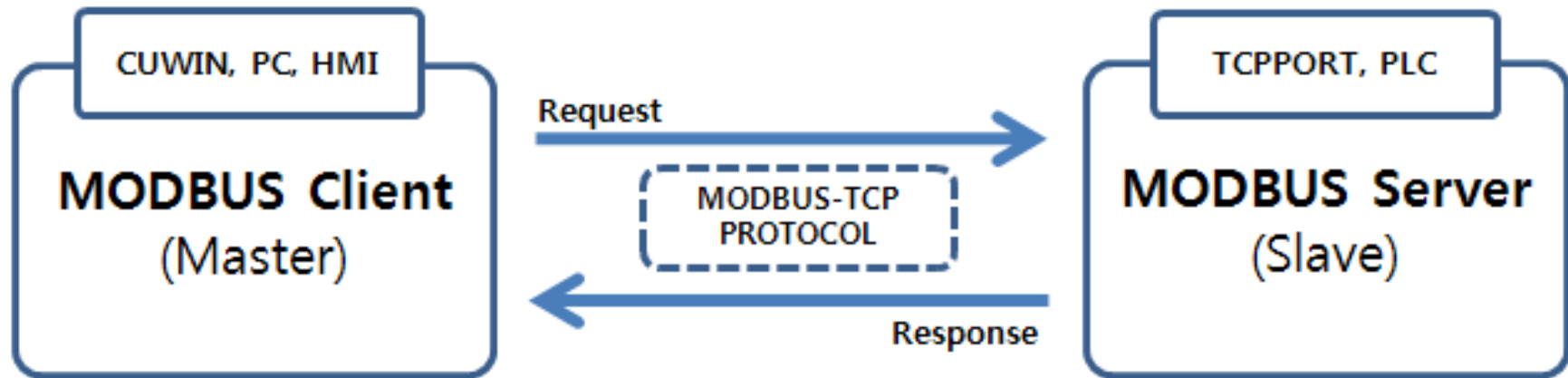
- 왼쪽 그림 - 시리얼 (RS-232 또는 RS-485)을 통한 Modbus 통신
- 오른쪽 그림 - 이더넷을 통한 통신에 동일한 응용 프로그램 계층 프로토콜이 사용



산업 제어 시스템 프로토콜 이해 및 분석

▶ Modbus 프로토콜 이해

- MODBUS-TCP 통신규격에는 마스터(Client)와 슬레이브(Server)의 역할이 나누어져 있음
- 슬레이브(Server)는 마스터(Client)가 요청하는 데이터에 대해 응답을 해줌
- 마스터(Client)에는 산업용터치 HMI 기기, 또는 PC 와 같은 상위 기기가 위치
- 슬레이브(Server)에는 TCPPORT 나 PLC 등이 위치
- 슬레이브(Server)는 상위기기에서 요청하는 동작만을 하는 수동적인 위치에 있음
- 반면 마스터(Client)쪽에서는 원하는 데이터를 읽어오거나, 원하는 데이터를 기입하는 등 적극적으로 슬레이브(Server) 기기를 다루어 주어야 합니다.

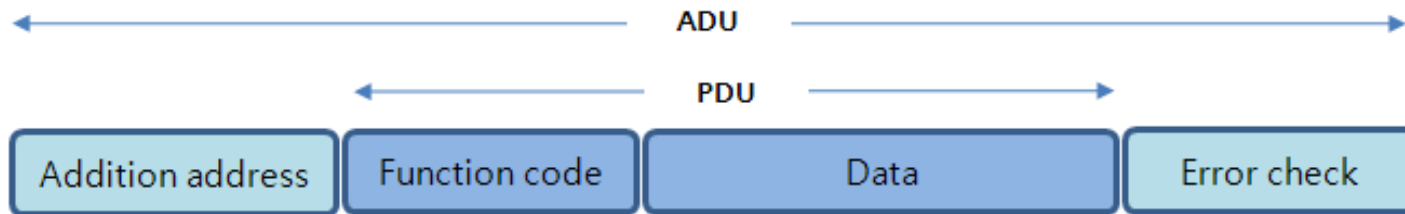


http://comfilewiki.co.kr/ko/doku.php?id=tcpport:modbus-tcp_%ED%94%84%EB%A1%9C%ED%86%A0%EC%BD%9C%EC%9D%B4%EB%9E%80:index

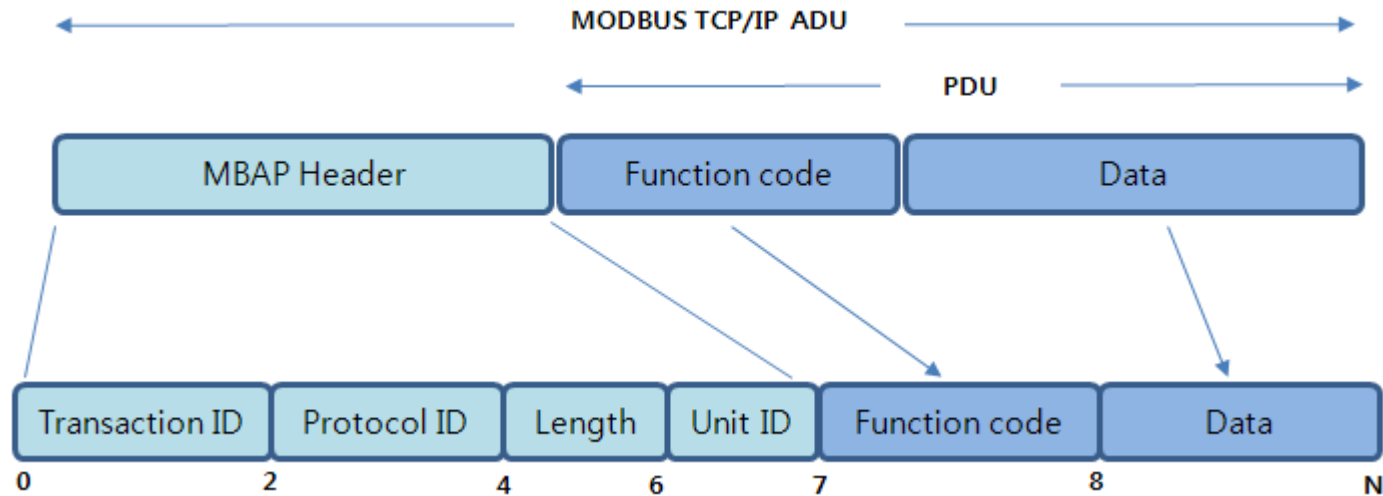
산업 제어 시스템 프로토콜 이해 및 분석

Modbus 프로토콜 이해

- 범용 MODBUS(RTU, ASCII) 프레임 구조



- MODBUS-TCP 프레임 구조



http://comfilewiki.co.kr/ko/doku.php?id=tcpport:modbus-tcp_%ED%94%84%EB%A1%9C%ED%86%A0%EC%BD%9C%EC%9D%B4%EB%9E%80:index

산업 제어 시스템 프로토콜 이해 및 분석

▶ Modbus 프로토콜 이해 - MBAP Header 부분

- MODBUS-TCP는 MBAP(Modbus Application Protocol)를 선두로 Function code, Data로 순으로 이루어져 있음. MBAP는 총 7 Byte이고 아래와 같은 내용의 Byte값을 나타냄
- **Transaction ID [2Bytes]**: 마스터(Client)가 최초 0x0000값 부터 통신시작 시 1씩 증가시키며 슬레이브(Server)는 그 값을 그대로 복사해서 사용. 쿼리 및 응답에 대해 한쌍으로 작업이 이루어졌는지를 확인하는 부분임
- **Protocol ID [2Bytes]**: 프로토콜의 ID를 나타내며 MODBUS-TCP는 0x0000의 고정값을 사용.
- **Length [2Bytes]**: Length 필드위치에서 프레임 마지막까지의 길이를 나타냄. 즉 Unit ID ~ Data끝까지의 Byte의 수를 나타냄
- **Unit ID [1 Byte]**: TPC/IP가 아닌 다른 통신선로의 연결되어있는 Slave를 구분하는 정보. Tcpport는 0x01로 고정

http://comfilewiki.co.kr/ko/doku.php?id=tcpport:modbus-tcp_%ED%94%84%EB%A1%9C%ED%86%A0%EC%BD%9C%EC%9D%B4%EB%9E%80:index

산업 제어 시스템 프로토콜 이해 및 분석

▶ Modbus 프로토콜 이해 - Function Code 부분

- Function Code는 Modbus 프로토콜에서 제공하는 명령어 집합코드
- Function Code를 이용하여 슬레이브 Memory(Coil, Register)에 값을 읽어오거나 쓸수 있는 서비스
- Function Code코드 1~127사이의 값을 사용하지만, TCP/PORT에서는 1, 2, 4, 5, 6, 15, 16 값을 지원 -> 실질적으로 사용하는 서비스이기 때문

기능	기술
FC=01	코일 상태 읽기
FC=02	입력 상태 읽기
FC=03	다중 홀딩 레지스터 읽기
FC=04	입력 레지스터 읽기
FC=05	단일 코일 쓰기
FC=06	단일 홀딩 레지스터 작성
FC=07	예외 상태 읽기
FC=08	진단
FC=11	통신 이벤트 카운터 받기 (직렬 회선 전용)

기능	기술
FC=12	통신 이벤트 로그 가져 오기 (직렬 회선 전용)
FC=14	장치 식별 읽기
FC=15	여러 코일 쓰기
FC=16	다중 홀딩 레지스터 작성
FC=17	슬레이브 ID 신고
FC=20	파일 레코드 읽기
FC=21	파일 기록 쓰기
FC=22	마스크 쓰기 레지스터
FC=23	다중 레지스터 읽기 / 쓰기
FC=24	FIFO 대기열 읽기
FC=43	장치 식별 읽기

산업 제어 시스템 프로토콜 이해 및 분석

▶ TCPPORT의 MODBUS 데이터 메모리 구조

- MODBUS 데이터 모델은 입력과 출력 그리고 비트 단위 접근과 워드 단위 접근을 기준으로 총 4가지 형태로 나뉘어짐
- 데이터 종류별로 데이터 블록을 각각 지정하여 4개의 데이터 블록으로 사용하거나 비트영역, 워드영역을 나누어 두개의 데이터 블록을 사용할 수 있음

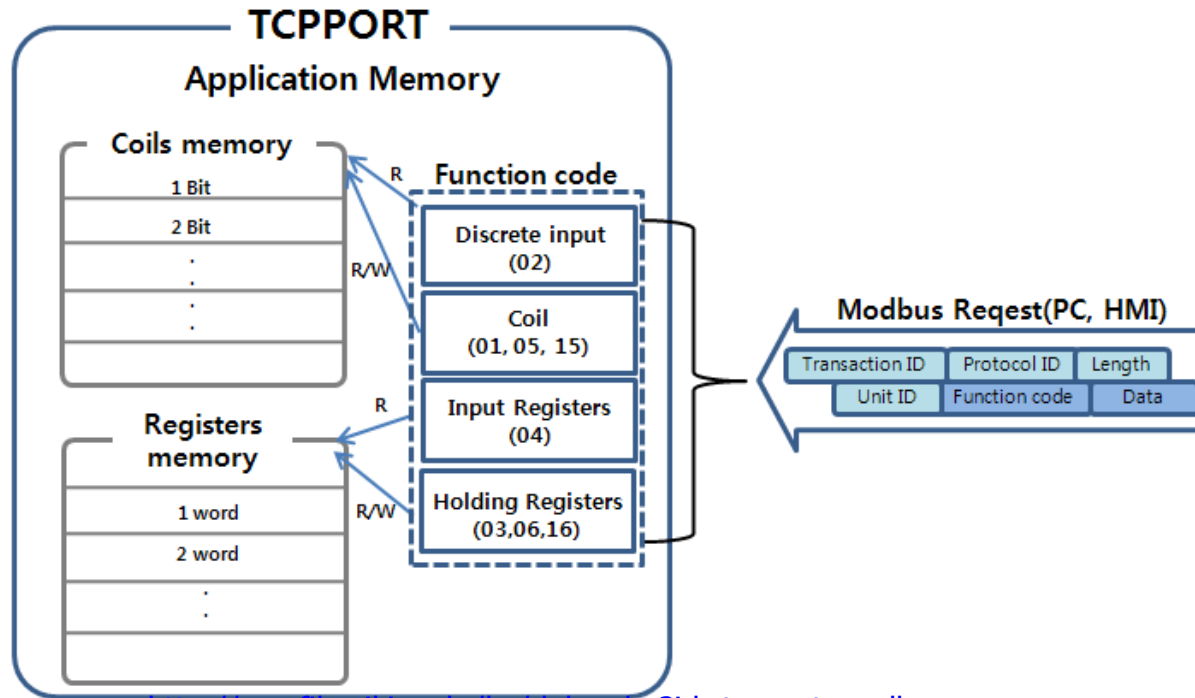
메모리	데이터모델	접근형태	읽기/쓰기	설명
Coil	Discrete Input	비트	읽기	상위장치에서 메모리 읽기 가능
Coil	Coils	비트	읽기/쓰기	상위장치에서 메모리 읽고, 쓰기 가능
Register	Input Registers	16비트 워드	읽기	상위장치에서 메모리 읽기 가능
Register	Holding Resisters	16비트 워드	읽기/쓰기	상위장치에서 메모리 읽고, 쓰기 가능

http://comfilewiki.co.kr/ko/doku.php?id=tcpport:modbus-tcp_%ED%94%84%EB%A1%9C%ED%86%A0%EC%BD%9C%EC%9D%B4%EB%9E%80:index

산업 제어 시스템 프로토콜 이해 및 분석

TCPPORT의 MODBUS 데이터 메모리 구조

- TCPPORT는 16비트 워드영역(Resisters)과 비트영역(Coils)으로 두개의 데이터 메모리로 나누어짐
- 메모리는 Slave(server)장비의 메모리를 말하며 Master(Client)는 위의 평선코드를 이용하여 Slave(server)장비의 메모리를 읽거나 원하는 값으로 변경할 수 있음
- 해당 Function Code에 따라 어떤 메모리를 접근할 것인지, 어떤 작업수행(Read, Write)을 행할 것인지가 나누어져 있음



http://comfilewiki.co.kr/ko/doku.php?id=tcpport:modbus-tcp_%ED%94%84%EB%A1%9C%ED%86%A0%EC%BD%9C%EC%9D%B4%EB%9E%80:index

산업 제어 시스템 프로토콜 이해 및 분석

▶▶ Modbus 프로토콜 이해 - Data 부분


- Data는 Function Code에 따라 그 구조가 조금씩 달라짐. Data는 기본적으로 Start Address, Length, Byte Count, Data의 형태를 가지고 있음
- **Start Address [2Bytes]**: 접근하려는 메모리의 시작번지. 2Byte로 표현되면 상위 Byte 우선순위. (예 0x4001번지 접근 시 0x40 0x01)
- **Length [2Bytes]**: 시작번지부터 값을 읽거나 쓸 길이를 나타냄
- **Byte Count [1Bytes]**: Request, Response에따른 메모리 Data의 byte 수를 나타냄. 즉, 읽어나거나 쓰려는 메모리 데이터의 Byte의 개수를 말함
- **DATA [N Byte]**: Request, Response에 따른 메모리 Data의 값 나타냄. 즉, 읽어나거나 쓰려는 메모리 값
- Funtion Code별 DATA 구조 및 디바이스 제어 상세 참고

http://comfilewiki.co.kr/ko/doku.php?id=tcpport:modbus-tcp_%ED%94%84%EB%A1%9C%ED%86%A0%EC%BD%9C%EC%9D%B4%EB%9E%80:index


산업 제어 시스템 프로토콜 이해 및 분석

Modbus 서버 설치 - 우분투










- Modbus slave 설치 - modbuspal.jar 파일
- <https://sourceforge.net/projects/modbuspal/files/modbuspal/RC%20version%201.6b/>

 **Download Latest Version**
ModbusPal.jar (1.1 MB)

Get Updates



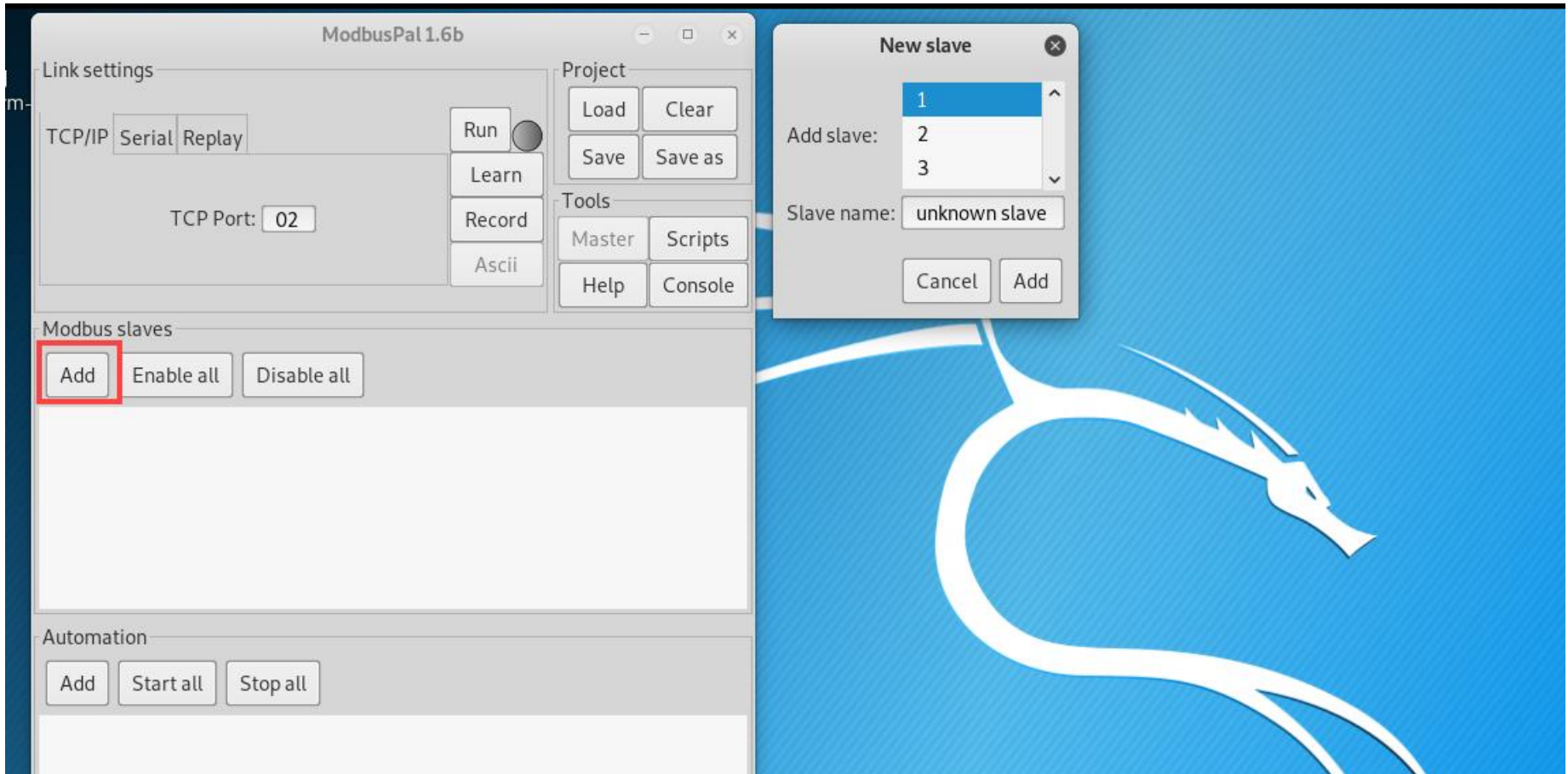
[Home](#) / [modbuspal](#) / RC version 1.6b

Name ▾	Modified ▾	Size ▾	Downloads / Week ▾
 Parent folder			
README.TXT	2011-07-25	364 Bytes	2  
modbuspal-help.zip	2011-07-25	1.5 MB	6  
modbuspal-javadoc.zip	2011-07-25	701.2 kB	3  
ModbusPal.jar	2011-07-25	1.1 MB	71  
Totals: 4 Items		3.3 MB	82

산업 제어 시스템 프로토콜 이해 및 분석

▶ Modbus Slave 동작 및 설정

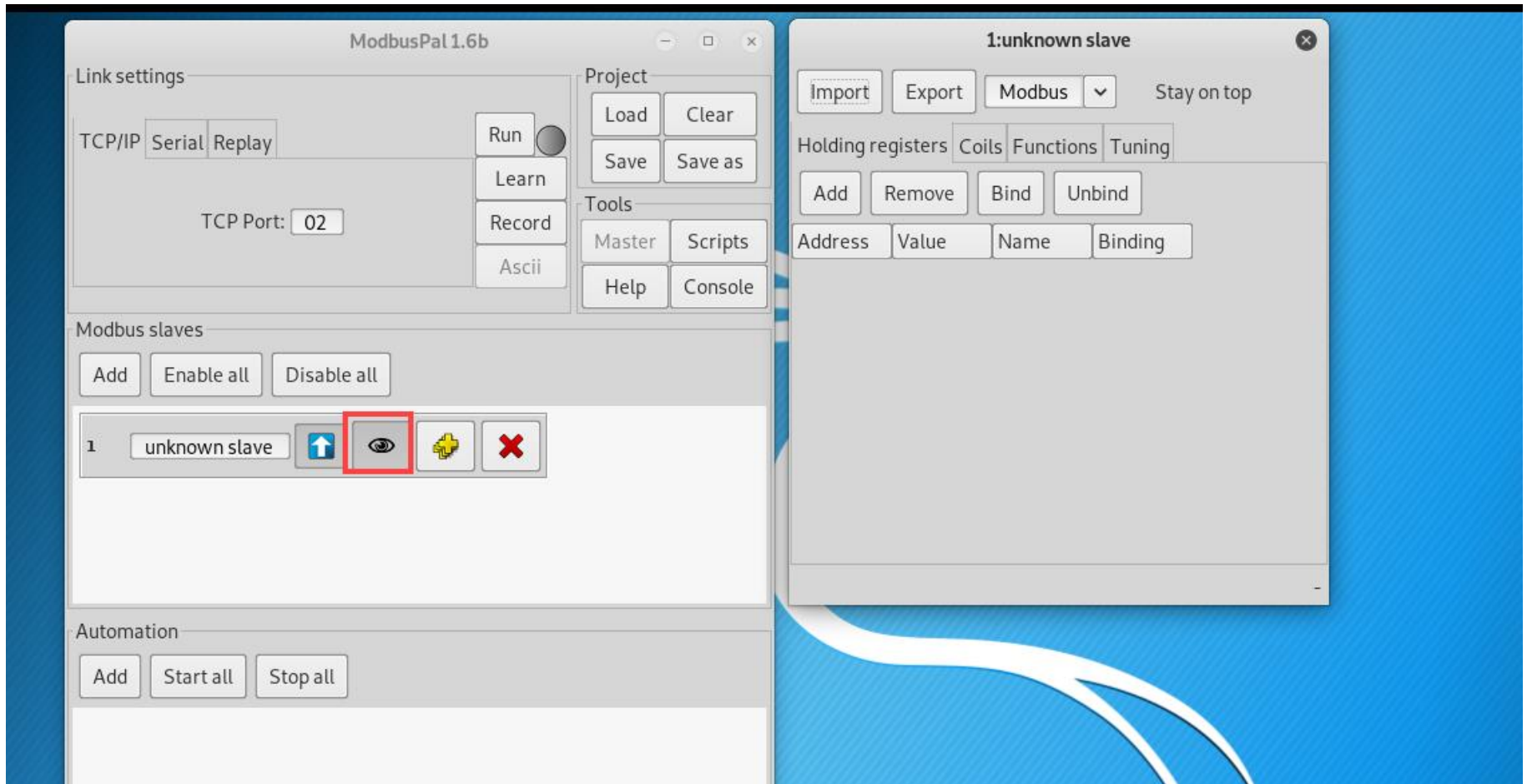
- java -jar ModbusPal.jar 실행 한 후에, slava 추가



산업 제어 시스템 프로토콜 이해 및 분석

Modbus Slave 동작 및 설정

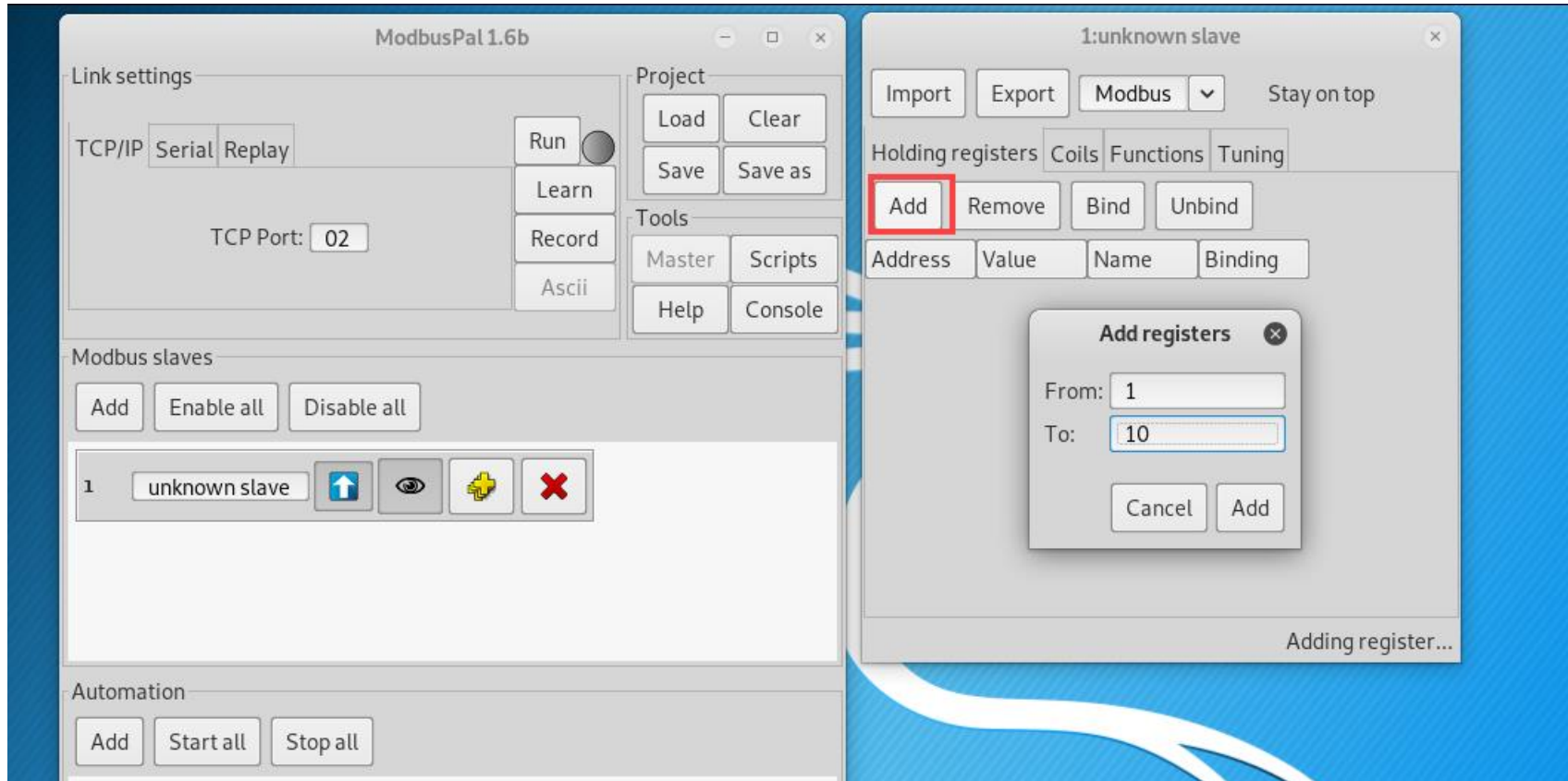
- 생성된 slave에 registers와 Coils 추가



산업 제어 시스템 프로토콜 이해 및 분석

▶ Modbus Slave 동작 및 설정

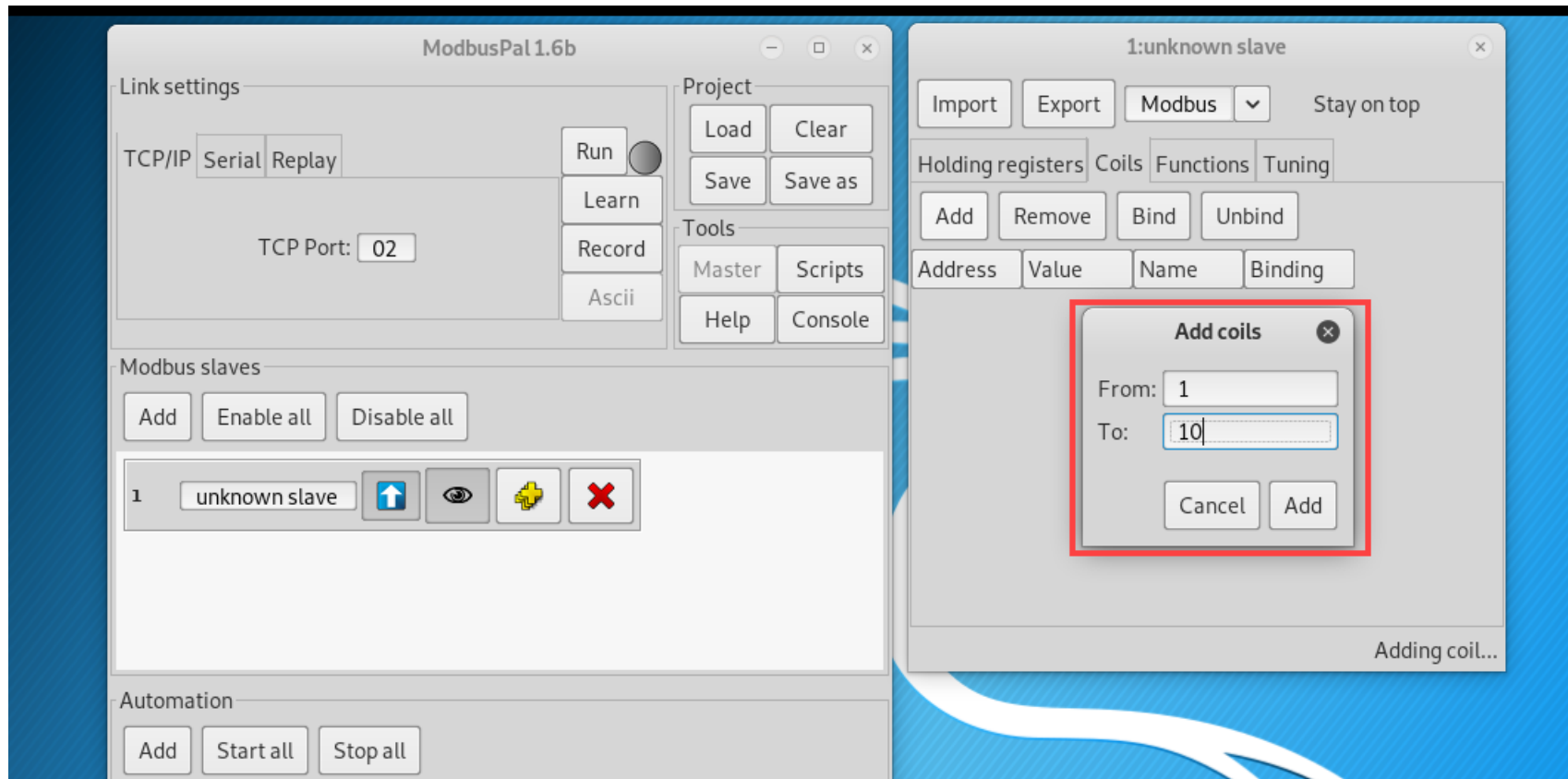
- 생성된 slave에 registers와 Coils 추가



산업 제어 시스템 프로토콜 이해 및 분석

▶ Modbus Slave 동작 및 설정

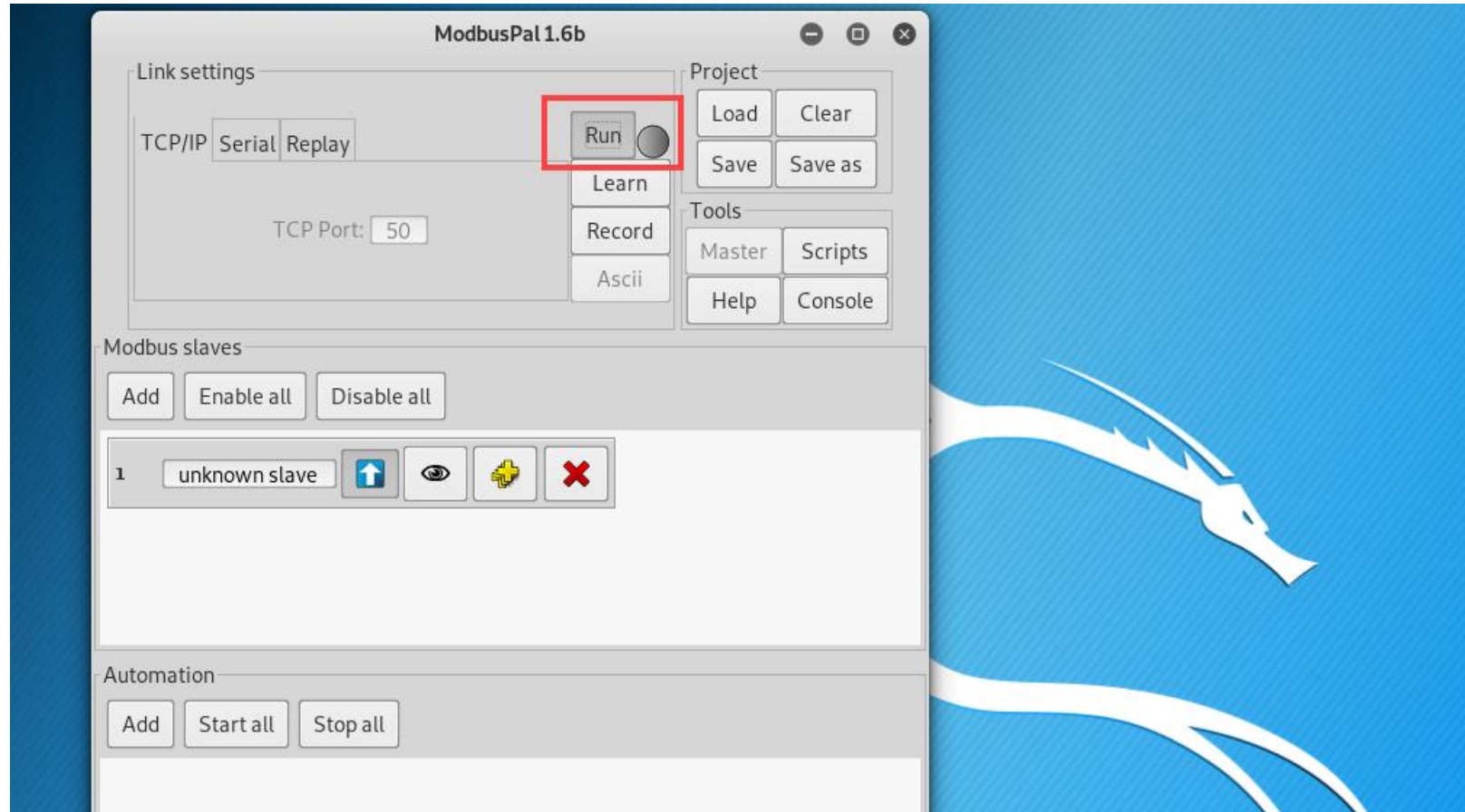
- 생성된 slave에 registers와 Coils 추가



산업 제어 시스템 프로토콜 이해 및 분석

▶ Modbus Slave 동작 및 설정

- 모두 추가되었으면 Slave 실행



산업 제어 시스템 프로토콜 이해 및 분석

Modbus Master 실행 및 설정

- qModMaster를 이용하여 생성된 Slave와 연결 시도

The image shows a Windows file explorer window on the left and the QModMaster application window on the right.

File Explorer:

- 이름 (Name)
- 수정한 날짜 (Modified Date)
- iconengines
- imageformats
- ManModbus
- platforms
- translations
- libEGL.dll
- libgcc_s_dw2-1.dll
- libGLSV2.dll
- libstdc++-6.dll
- libwinpthread-1.dll
- qModMaster.exe** (checked)
- qModmaster.exe.manifest
- qModMaster.ini
- QModMaster.log
- Qt5Core.dll
- Qt5Gui.dll
- Qt5Svg.dll
- Qt5Widgets.dll
- README.txt

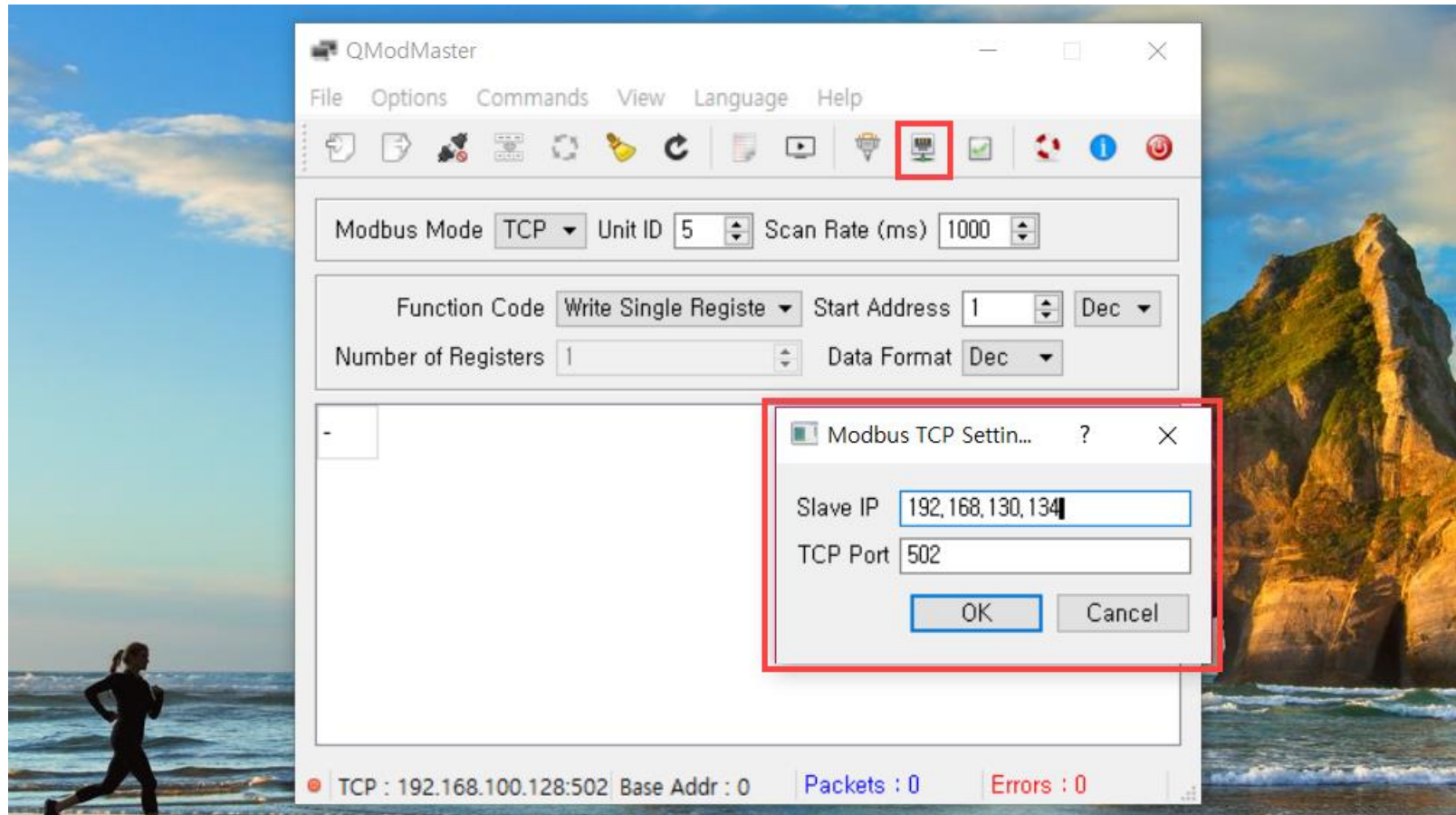
QModMaster Application:

- Modbus Mode: TCP
- Unit ID: 5
- Scan Rate (ms): 1000
- Function Code: Write Single Register
- Start Address: 1
- Dec (Data Format)
- Number of Registers: 1
- Data Format: Dec
- Status bar: TCP : 192.168.100.128:502 Base Addr : 0 Packets : 0 Errors : 0

산업 제어 시스템 프로토콜 이해 및 분석

▶▶ Modbus Master 실행 및 설정

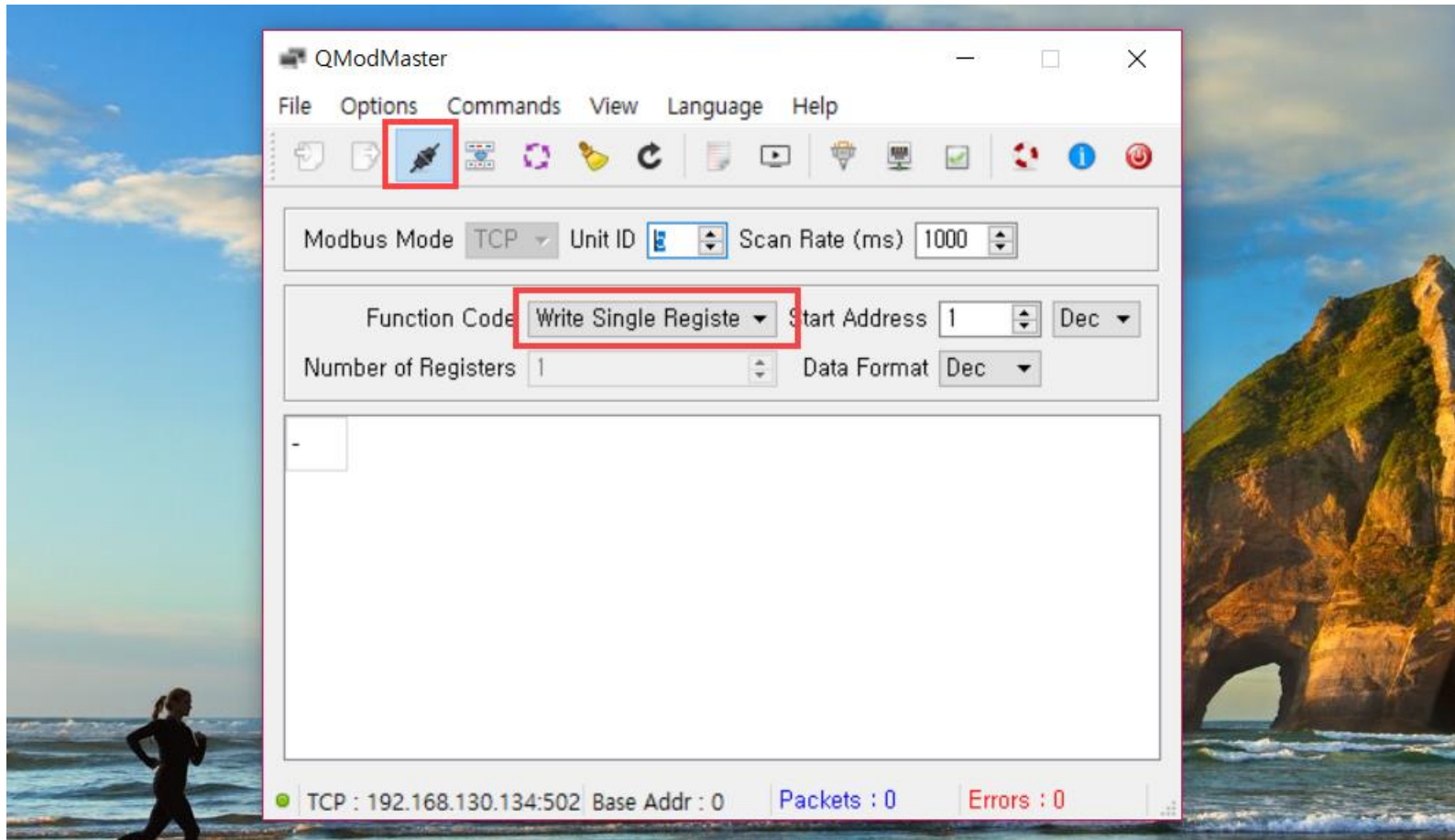
- Slave 주소와 TCP 포트(502/TCP) 설정



산업 제어 시스템 프로토콜 이해 및 분석

▶ Modbus Master 실행 및 설정

- 연결(Connect) 한 후에, Function Code를 설정하여 데이터 보냄



산업 제어 시스템 프로토콜 이해 및 분석

Modbus Master 실행 및 설정

- 원격 Slave 의 변경 정보를 확인

The image displays the ModbusPal 1.6b software interface. The main window is titled 'ModbusPal 1.6b' and contains several sections:

- Link settings:** Includes tabs for TCP/IP, Serial, and Replay. The TCP Port is set to 50. There are buttons for Run, Learn, Record, and Ascii.
- Project:** Includes buttons for Load, Clear, Save, and Save as.
- Tools:** Includes buttons for Master, Scripts, Help, and Console.
- Modbus slaves:** Includes buttons for Add, Enable all, and Disable all. A list of slaves is shown with one slave named 'unknown slave'.
- Automation:** Includes buttons for Add, Start all, and Stop all.

A secondary window titled '1:unknown slave' is open, showing a table of holding registers. The table has columns for Address, Value, Name, and Binding. The values are as follows:

Address	Value	Name	Binding
10			
2	65535		
30			
40			
50			
60			
70			
80			
90			
100			

At the bottom of the '1:unknown slave' window, a status message reads 'Adding coils completed.'