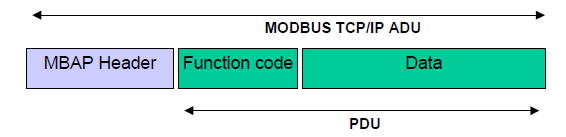
컴퓨터 네트워크(01)

Socket Program HW3 19.12.1

21300691 정원식

실험 결과

1. Modbus-TCP의 서버 포트 번호는 502 이다.
2. Modbus-TCP의 PDU 포맷은 아래와 같다.

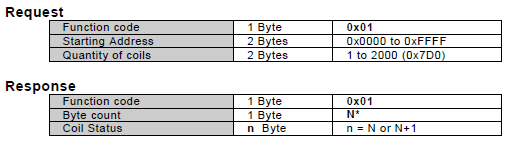


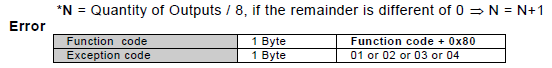
MBAP 헤더의 경우 아래와 같이 총 7바이트로 구성이 된다.

|  |  |  |
| --- | --- | --- |
| 필드 | 길이 | 설명 |
| Transaction Id | 2 Bytes | 트랜잭션을 구별하기 위해 사용된다. 즉 클라이언트가 보낸 request의 TxId는 서버에 의해 response 될때, 같은 TxId를 사용하여 클라이언트가 자기가 보낸 요청의 응답임을 확인하게 한다. |
| Protocol Id | 2 Bytes | 0이 기본 값이다 |
| Length | 2 Bytes | UId, Function Code, Data 세가지 필드 길이의 합이다. |
| Unit Identifier | 1 Bytes | 클라이언트를 구별하기 위해 사용된다. 이 값은 Client에 의해 초기화 되며 서버의 요청은 같은 UId 값을 보낸다. |

* Function Code: Fucntion Code는 요청하는 기능이 무엇인지 구별하기 위해 사용되며 1바이트의 크기를 가진다.
* Data: 기능별로 요청할 때 필요한 데이터이다. Read Coils Function을 예로 들면 Data로 2Bytes의 시작 주소와 2Bytes의 요청 양(총 4바이트) 을 가진다.

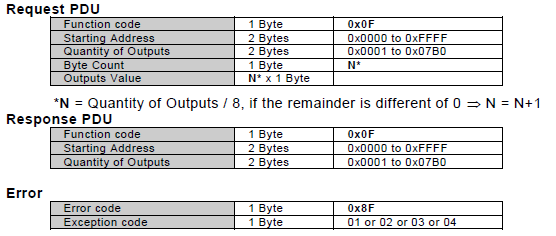
1. Read coils 함수는 특정 주소의 coil의 On/Off(True/False) 상태를 요청한다. Request, response, error 포맷은 아래와 같다. response 값은 비트 단위로 반환되며 빅엔디안이다. 즉 0번부터 15번 주소까지의 coil의 상태를 요청하면 반환되는 값은 아래와 같다.



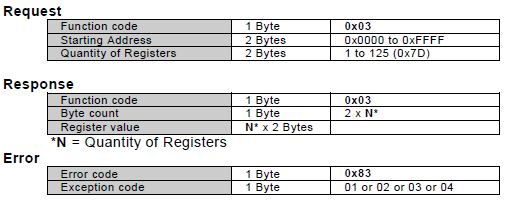


|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 첫번째 바이트 | | | | | | | | 두번째 바이트 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |

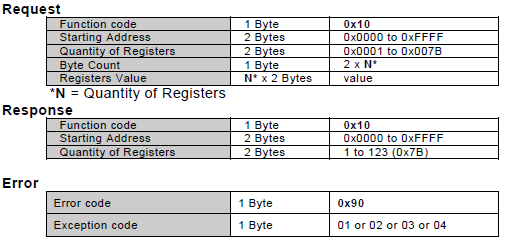
1. Write multiple coils 함수는 특정 주소의 coil의 On/Off(True/False) 상태를 바꾸도록 요청한다. Request, response, error 포맷은 아래와 같다.



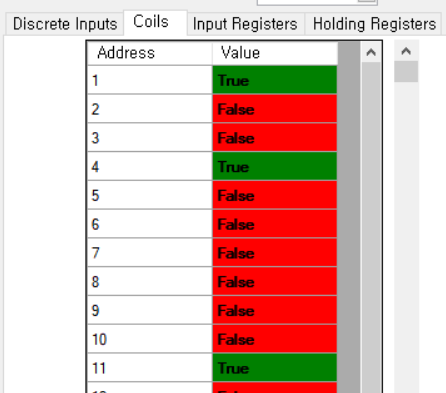
1. Read Holding Registers 함수는 특정 주소 holding register들의 Value 값을 요청한다. Request, response, error 포맷은 아래와 같다. response 값은 바이트 단위로 반환되며 빅엔디안이다.

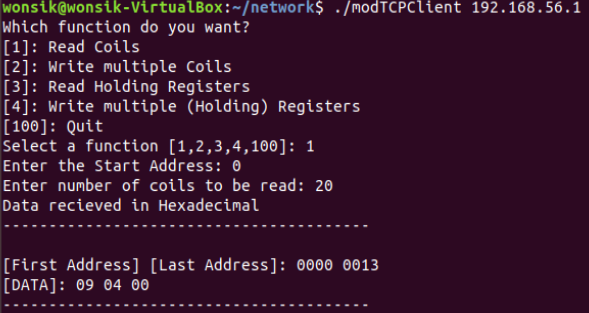


1. Write multiple Holding Registers 함수는 특정 주소 register의 value값을 바꾸도록 요청한다. Request, response, error 포맷은 아래와 같다.

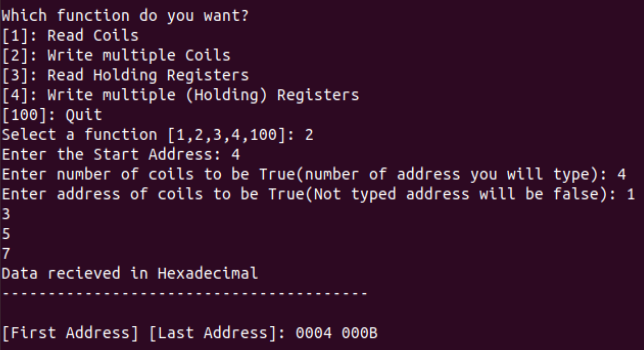


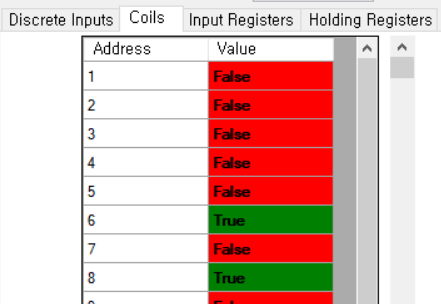
1. 아래와 같이 coil의 상태 되어 있다고 가정하자(1,4,11 On). 이때 read coils function을 실행하여 0번 주소부터 총 20개의 coil 값을 읽으면 아래와 같이 09 04 00을 응답으로 받는다. 즉 09(1번,4번) 04(11번) 00(X) 으로 정확히 켜져있는 값을 응답하는 것을 확인할 수 있다.



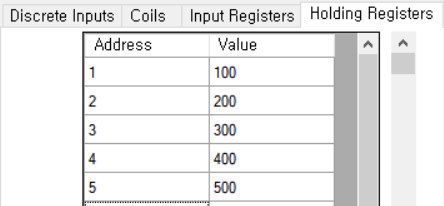


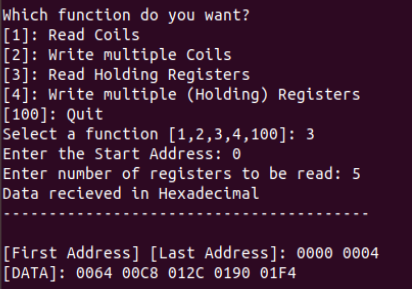
1. coil의 상태가 모두 꺼져 있다고 가정하자(all Off). 이때 write multiple coils function을 실행하여 4번 주소부터 총 4개의 주소를 On하고 나머지 주소는 Off할것을 요청한다. (1,3,5,7) 이 경우 1,3 번 주소는 4번 주소보다 아래이므로 값이 반영되지 않는다. 반면 5,7번 주소의 경우 4번주소보다 위이므로 값이 반영된다.





1. 아래와 같이 register의 상태 되어 있다고 가정하자(0: 100, 1: 200, 2: 300, 3: 400, 4: 500). 이때 read registers function을 실행하여 0번 주소부터 총 5개의 register 값을 읽으면 아래와 같이 0064(100) 00CB(200) 012C(300) 0190(400) 01F4(500)을 응답으로 받는다.





1. register의 상태가 모두 꺼져 있다고 가정하자(all 0). 이때 write multiple registers function을 실행하여 4번 주소부터 총 4개의 주소를 100,200,300,400으로 설정한다. 그 결과 5:100, 6:200, 7:300, 8:400 으로 register 값이 설정되었다.

