

Serial to Ethernet User Guide

Version 1.00



©2013 WIZnet Co., Ltd. All Rights Reserved.

For more information, visit our website at <http://www.wiznet.co.kr>

Document Revision History

Date	Revision	Changes
2013-03-22	V1.00	Official Release

WIZnet's Online Technical Support

If you have something to ask about WIZnet products, write down your question on Q&A Board in WIZnet website (www.wiznet.co.kr). WIZnet will give an answer as soon as possible.

COPYRIGHT NOTICE

Copyright 2013 WIZnet Co., Ltd. All Rights Reserved.

Technical Support: support@wiznet.co.kr

Sales & Distribution: sales@wiznet.co.kr

For more information, visit our website at <http://www.wiznet.co.kr>

Table of Contents

1. Introduction	1
2. Input & Response Rule	2
3. Ethernet Command	7
4. Network Command	8
5. Management Command	12
6. Application Functions	15
7. Mode Configuration	16
8. Usage Examples	17

1. Introduction

■ 소개

AT Command 란 모뎀에서 사용하던 Command Set 으로 터미널의 입력 창을 통해서 명령을 입력하여 기능을 제어하는 역할을 하던 명령어이다. 현재 모뎀 이외의 많은 장치에서 장비를 제어하는 용도로 사용하고 있으며 기존 명령으로는 해결되지 않는 부분은 확장 명령을 통하여 자체적으로 추가하여 사용된다. 그리고 사용 방식이 AT Command 의 규칙을 따르지 않는 독자적인 Command 입력 방식에 대해서도 관습적으로 AT Command 라고 부른다.

WIZnet 에서도 제품 별 필요에 따라 자체적으로 변형한 AT Command 를 모듈에 적용시켜 사용해 왔으나, 이전의 제품에서는 Command 의 형태와 내용이 제품 별로 서로 상이하여 여러 제품을 동시에 사용하는 경우 제품 별로 Command 를 따로 기억하거나 프로그래밍을 해야 하므로 사용에 어려움이 있었다.

그러한 이유로, 본 Command Set 에서는 WIZnet 전 제품에서 공통적으로 사용이 가능하도록 Command Set 을 구성하여 한번 사용 법을 숙지한 사용자가 이 후 접하는 제품에 대해 쉽게 적응할 수 있도록, 그리고 한번 작성된 Interface Code 가 쉽게 재사용 될 수 있도록 하였고, 기존 명령에 비해 더 간편하고 합리적이며 확장 성을 갖추도록 하는 것을 목표로 하여 개발하게 되었다.

■ 특징

- WIZnet 디바이스의 특성 상 TCP/IP 통신에 특화된 명령으로 구성
- 명령의 분류를 기능별 특성과 Network 계층으로 구분
- Manual 입력 시 사용자가 편리하게 사용할 수 있도록 고려
- 하나의 명령이 중복된 기능을 수행하지 않으면서 명령의 개수가 적게 유지되도록 설계
- 지연되는 기능들은 Async 방식으로 실행되어 Blocking 없이 추가 동작이 가능
- 제어하는 Host 입장에서 자신에게 맞는 Polling 모드를 선택하여 사용이 가능
- 앞으로 개발될 WIZnet 제품에 공통적으로 적용될 AT Command

2. Input & Response Rule

1. Command General

■ Input Format

> Command 와 Parameter 를 입력하는 공통 형식

> 형식: **(ATCmd)(Operator)(Parameters)↓(Data)**

- (ATCmd): 명령 표에서 정의된 AT Command 이며, 구현에 따라서는 AT+의 입력을 생략하거나 대,소문자의 구분을 무시할 수도 있다.
- (Operator): 명령 표의 Property 에서 정의된 =, -, ? 부호를 나타내며, Parameters 와 연관되어 사용된다. Operator 없이 AT Command 만으로 사용할 수도 있다.
- (Parameters): 인자의 입력이 필요한 경우 Operator 와 연관되어 사용된다.
- ↓: 구분 자 역할을 하는 Enter 키를 나타낸다.
- (Data): 명령 입력 후에 데이터의 입력이 필요한 경우 (AT+NSEND), 이 위치에서 미리 지정된 크기만큼의 데이터가 입력되어야 한다.

■ Response Format

> Command 입력 후 응답으로 반환되는 공통 형식

> 형식: **[(Type),(Id),(Param1),(Param2),(Param3),(Param4),(Param5),(Param6)]↓(Data)↓**

- (Type): 응답 종류를 나타내며, S, D, F, W, R, V 중에 하나가 사용된다.
- (Id): Socket 번호 등과 같이 동작하는 주체를 나타내며, Async 에서 사용된다.
- (Param1) ~ (Param6): 인자 출력 필요 시 사용되며 없는 경우 생략됨.
- ↓: 구분 자 역할을 하는 Enter 키를 나타내며, CRLF(0x0d, 0x0a)가 사용된다.
- (Data): 큰 데이터의 출력이 필요한 경우인 D 와 R 타입의 응답에서 사용된다.

■ Command Property Symbol on the table

- > '=' Parameter 입력 부호 사용 가능
- > '.' Individual 입력 부호 사용 가능
- > '?' Query 부호 사용 가능
- > '[' 부호 없이 사용 가능
- > '[x]' 부호 없이 사용 가능하며, x 부호로 대체됨
- > 'Savable'
 - 메모리에 저장되어 Profile 저장과 복구에 사용되는 명령 (일회성이 아님)
 - 구현에 따라서는 적용되지 않을 수도 있음. (Optional)
- > 'Async'
 - Delay 가 예상되는 명령으로 비 동기로 처리됨을 표시 (Blocking 안됨)
 - 구현에 따라서는 적용되지 않을 수도 있음. (Optional)

2. Parameter Rule

■ '=' 부호 (일반 인자 입력)

- > 명령 실행 시 인자가 입력되어야 하는 경우에 사용됨.
- > 형식: **(ATCmd)=(Param1),(Param2),(Param3),(Param4),(Param5),(Param6)↓**
- > 규칙: ■ 사용하지 않는 인자는 빈칸으로 둘 수 있다. Ex) AT+XXX=1,,3,,
 - 사용하지 않는 뒤 콤마는 생략 가능하다. Ex) AT+XXX=1,,3,, => AT+XXX=1,,3

■ '-' 부호 (개별 인자 입력) (Optional)

- > 형식: **(ATCmd)-(Index),(Param)↓**
- > 규칙: ■ Index 는 = 부호에서 사용되는 인자의 순서이다.
 - Ex) AT+XXX=1,2,3,4,5,6 의 일반 입력이 가능한 경우 3 번 인자만 입력하고 싶은 경우 => AT+XXX-3,(TheValueToInput)
 - 여러 인자의 입력이 가능하지만 하나의 인자만 필요한 경우에 사용.
 - 일반 인자 입력으로 대체가 가능하며, 1 개 이상의 인자를 입력하는 경우, 일반 인자 입력을 사용해야 한다.

■ '?' 부호 (응답 요청) (Index 방식은 Optional)

- > 형식: **(ATCmd)?↓** 또는 **(ATCmd)?(Index)↓**
- > 규칙: ■ Device 에 정보 출력을 요청하는 경우 사용되는 명령어이다.
 - Index 는 개별 정보 출력 요청 시 사용되며, Optional 구현 항목이다.
 - Index 가 가리키는 것은 S 응답의 경우 출력되는 항목의 번호이며, D 응답의 경우 출력되는 라인의 번호이다.

■ '[' 부호 (부호 없음)

- > 형식: **(ATCmd)↓**
- > 규칙: 인자의 입력이 필요 없는 명령을 사용하는 경우 적용된다.

■ '[x]' 부호 (부호 없음, x 부호로 대체됨) (Optional)

- > 형식: **(ATCmd)↓**
- > 규칙: 인자를 입력하지 않은 경우 자동으로 x 입력으로 치환된다.
 - Ex) AT+XXX => AT+XXX? 로 내부에서 자동으로 변환됨

- Enter 키(↓)는 Carriage Return (0x0d, \r)는 무시하고 Line Feed (0x0a, \n)만 인식한다.
그 이유는 0x0d 를 Enter 로 사용하고 데이터의 첫 바이트가 0x0a 인 경우와 Enter 로 0x0d, 0x0a 를 사용하는 경우를 구별할 수 없기 때문이다. 그러므로 프로그래밍 작성 시 LF 또는 CRLF 를 Enter 키로 사용할 수 있지만, CR 을 단독으로 사용하거나, LFCR 로 순서가 바뀌면 오류가 발생하므로 주의해야 한다.

- AT+NSEND 에서는 명령과 인자를 입력한 후 Enter 키를 입력한 후, 데이터를 입력한다.

3. Response Rule

- ID Usage
Async 동작에서 사용되며 어떤 입력에 대한 응답인지를 구분하기 위하여 사용된다.
 - 0 시스템 ID
 - 1 ~ n 소켓 ID

- S 응답 (Success Response)
 - > 명령이 성공적으로 실행된 경우에 사용되며, 필요 시 인자와 함께 출력된다.
 - > 형식: **[S,(Id),(Param1),(Param2),(Param3),(Param4),(Param5),(Param6)]↓**
 - (Param1) ~ (Param6): 간략한 데이터의 출력이 필요한 경우 사용.

- D 응답 (Success Dump Response)
 - > 명령이 성공적으로 실행된 경우에 사용되며, Enter 가 포함된 큰 데이터를 출력한다.
 - > 형식: **[D,(Id),(Size)]↓(Data)↓**
 - (Size): 출력되는 Data 의 Byte Size

- F 응답 (Fail Response)
 - > 명령이 실패한 경우에 사용되며, 필요 시 인자와 함께 출력된다.
 - > 형식: **[F,(Id),(ErrorCode),(Param)]↓**
 - (ErrorCode): 실패한 이유에 대한 Error Code.
 - (Param): 해당 Error Code 에서 상세한 정보를 나타내는 경우 사용되는 값.

- W 응답 (Wait Response)
 - > 명령이 Async 로 동작하는 경우 사용되며, 해당 ID 로 명령이 시작되었음을 나타냄.
 - > 형식: **[W,(Id)]↓**

■ R 응답 (Data Receive Response)

> 수신된 Data 를 출력하는 경우 사용된다.

> 형식: **[R,(SockId),(Size),(SrcIP),(SrcPort)]↓(Data)↓**

- (SockId): 어떤 Socket 에서 수신한 데이터 인지를 표시함.
- (Size): 수신한 Data 의 Byte Size
- (SrcIP): 전송한 측의 IP 를 나타내며, UDP 에서 사용되며, TCP 시 생략됨.
- (SrcPort): 전송한 측의 Port 를 나타내며, UDP 에서 사용되며, TCP 시 생략됨.

■ V 응답 (Event Response)

> 사용자가 알아야 하는 Event 가 발생한 경우 사용됨.

> 형식: **[V,(Id),(EventCode)]↓**

- (Id): Event 가 발생한 주체.
- (EventCode): Event 항목을 나타내는 번호로 ID 의 종류에 따라 다르다.

4. Response Code

■ General Error Code

Num	Error Name	Descript
0	Unspecified Error	정의되지 않은 에러 (가능하면 사용하지 말 것)
1	Wrong Operator	잘못된 AT명령
2	Wrong Command Sign	잘못된 명령 부호
3	Wrong Arguments	잘못된 인자 (Param: 문제가 되는 인자 번호)
4	Out of Range	허용되지 않은 인자 크기 (Param: 문제가 되는 인자 번호)
5	Disabled	비활성화된 기능 (Param: 문제가 되는 인자 번호)
6	Not Allowed	허용되지 않는 동작 (Param: 문제가 되는 인자 번호)
7	Command Busy	해당 객체가 동작 중이므로 명령 입력 불가
8	Timeout	명령 처리 시간 초과

■ Socket Error Code

Num	Error Name	Descript
10	Socket Not Available	사용 가능한 소켓이 없음
11	Socket Closed	소켓이 Closed 상태
12	Port Not Available	해당 포트가 사용 중
13	Not Connected	소켓이 Connected 상태가 아님
14	Wrong Address	잘못된 주소 또는 포트
15	No Data Available	수신된 데이터 없음

■ Other Error Code

Num	Error Name	Descript
20	No Free Memory	여유 메모리 부족

■ Socket Event Code

Num	Event Name	Descript
0	Connected	TCP Server가 Listen상태에서 Connected 상태로 전환
1	Disconnected	TCP Connection이 Connected 에서 Disconnected 로 전환
2	Closed	TCP Socket이 Closed 상태로 전환
3	Data Received	해당 소켓에서 Data 를 수신

■ Other Event Code

Num	Event Name	Descript
-	-	-

3. Ethernet Command

Command	Prop.	Input Parameter	Input Resp.	Query Response
AT+ESET	= ? [?]	T/H	[S]	[S,,T/H]
AT+ESTAT	? [?]			[S,, T/H,P/U]

■ AT+ESET (Ethernet configuration)

- > Ethernet Mode 를 선택한다. Query 응답은 현 상태가 아닌 내가 설정한 변수를 반환한다.
- > Parameter ■ T: 10Mbps, H: 100Mbps

■ AT+ESTAT (View Ethernet Status)

- > 현재 Ethernet 의 상태를 반환한다.
- > Parameter ■ T: 10Mbps, H: 100Mbps
■ P: Plugged, U: Unplugged

4. Network Command

Command	Prop.	Input Parameter	Input Resp.	Query Response
AT+NSET	= - ? [?] Savable	S,(IP),(SN),(GW),(DNS),(DNS2) D	[S]	[S,,S,(IP),(SN),(GW),(DNS1),(DNS2)] [S,,D]
AT+NSTAT	? [?]			[S,,S/D,(IP),(SN),(GW),(DNS1),(DNS2)]
AT+NMAC	= ? [?] Savable	(MAC)	[S]	[S,,(MAC)]
AT+NOPEN	= Async	S/C/U,(SrcPort),(DstIP),(DstPort)	[S,,(SockId)] or [W,(SockId)] [S,(SockId)]	
AT+NAUTO	= ? [?] Savable	A,S/C/U,(SrcPort),(DstIP),(DstPort) R,(Idx)	[S,,(Idx)]	[D,,(size)]↓(Data)
AT+NCLS	= Async	(SockId)	[S,,(SockId)] or [W,(SockId)] [S,(SockId)]	
AT+NSEND	= Async	(SockId),(size),(DstIP),(DstPort)	[S,,(SockId)] or [W,(SockId)] [S,(SockId)]	
AT+NRECV	= [=] Async	(SockId),(BufLen)		[R,(SockId),(Size),(SrcIP),(SrcPort)]↓(Data)
AT+NSOCK	= ? [?]	(SockId)		[S,,S/C/U,(SrcPort),(DstIP),(DstPort)] [D,,(Size)]↓(Data)
AT+NOPT	= - ? [?] Savable	미정	미정	미정

■ AT+NSET (Network Configuration)

> Network 주소 설정을 수행한다. 주소 인자는 **Static** 설정 시 사용되는 값으로 비워 두면 Default 값 혹은 이전에 **Static** 으로 설정된 값이 사용되며, **DHCP** 동작 시에는 설정한 주소 값이 실제 값과 다를 수 있다.

> Parameter ■ S: Static, D: DHCP

■ IP: My IP Address, SN: Subnet Mask, GW: Gateway Address

■ DNS: Domain Name Server Address

■ AT+NSTAT (View Network Status)

> 현재 Network 의 상태를 확인한다. 설정 값이 아닌 사용되고 있는 값을 읽어서 반환하기 때문에 내가 설정했던 값과 다를 수 있다.

> Parameter ■ S: Static, D: DHCP, DNS: Domain Name Server Address
■ IP: My IP Address, SN: Subnet Mask, GW: Gateway Address

■ AT+NMAC (Manage MAC Address)

> MAC Address 의 값을 설정하거나 확인한다.

> Parameter ■ MAC: MAC Address in the form of 'xx:xx:xx:xx:xx:xx'

■ AT+NOPEN (Socket Open)

> Socket 을 Open 한다. TCP Client 의 경우 Connection 까지 수행된다. Destination IP 와 Port 인자는 TCP Server 시에는 사용되지 않으며, Client 의 경우에는 접속될 곳의 주소를 나타낸다. UDP 의 경우에는 입력하지 않아도 무방하나, 입력한 경우에는 Default Destination 주소 값으로 사용되어 AT+NSEND 시 주소 값을 입력하지 않은 경우에 사용된다. (계속 같은 곳으로 전송하는 경우에 편리)

> Parameter ■ S: TCP Server, C: TCP Client, U: UDP
■ SrcPort: Source(My) Port Number
■ DstIP: Destination IP Address, DstPort: Destination Port Number
■ SockId: Assigned Socket Number

■ AT+NAUTO (Socket Open Auto - when system boot)

> 본 명령을 사용하여 Socket Auto Open 설정을 만들고 AT+MPROF 를 사용하여 Flash Memory 에 저장해 두면, Booting 시 (Physical Layer 가 연결되어 있으면) 저장된 Socket Open 동작이 사용자의 개입 없이 자동으로 수행된다.

> Dump Format: (Idx),S/C/U,(SrcPort),(DstIP),(DstPort)↓

> Parameter ■ A: Add, R: Remove, Idx: Service Index (returned value of '?' operation)
■ S: TCP Server, C: TCP Client, U: UDP
■ SrcPort: Source(My) Port Number
■ DstIP: Destination IP Address, DstPort: Destination Port Number

■ AT+NCLS (Socket Close)

> Open 상태의 Socket 을 Close 한다. Connected 된 경우에는 자동으로 접속이 종료되며, 모든 설정 값이 사라지고, 변수가 초기화된다. Closed 상태의 Socket 을 Close 하는 경우, Async 로 구현된 경우라도 W 응답 없이 바로 S 가 회신된다.

> Parameter ■ SockId: Assigned Socket Number

■ AT+NSEND (Send Data)

> Connected 된 TCP Socket 또는 Open 된 UDP Socket 으로 입력된 크기만큼의 Data 를 전송한다. Sync 로 구현된 경우 명령이 입력되어 문제가 없으면 S 응답이 회신되고, S 를 확인한 후 전송할 Data 를 입력하면 된다. 지정된 길이만큼 입력하면 전송이 시작된다. Async 로 구현된 경우에는 명령이 입력되고 문제가 없으면 W 응답이 회신되고, W 를 확인한 후 전송할 Data 를 입력하면 된다. 지정된 길이만큼 입력이 완료되면 응답 없이 전송이 시작되며 전송 완료 시 S 응답이 회신된다.

> Parameter ■ SockId: Assigned Socket Number, Size: Data size to be sent.
 ■ DstIP: Destination IP Address, DstPort: Destination Port Number
 UDP 에서 필요 시 사용된다.

■ AT+NRECV (Receive Data)

> Connected 된 TCP Socket 또는 Open 된 UDP Socket 으로 입력된 크기만큼의 Data 를 수신한다. 회신할 최대 Size 값을 입력하지 않으면 Platform 이 허용하는 최대 값이 사용되며, SockId 를 입력하지 않으면 먼저 수신된 순서대로 Data 를 회신한다.

AT+MSET 에서 Poll Mode 가 Disabled 이면 본 명령의 입력이 없어도 Data 를 수신하는 즉시 회신되지만, Poll Mode 가 Semi 또는 Full Mode 인 경우 본 명령을 입력해야만 수신된 Data 를 회신한다. 수신된 데이터가 있는지 여부는 Event 를 통해서 알 수 있다.

> Parameter ■ SockId: Assigned Socket Number, 미 지정 시 수신 순서대로 회신됨.
 ■ BufLen: 내가 수신 가능한 최대 Buffer Size 로 입력이 없으면 Default 값이 사용되며, 그 값은 구현되는 Platform 에 따라서 다르다.
 ■ SrcIP: Source(Peer) IP Address, SrcPort: Source(Peer) Port Number
 UDP 에서 사용됨.

■ AT+NSOCK (Socket Status)

> 현재 Socket(들)의 상태를 회신한다. Query 의 경우 Closed 가 아닌 모든 Socket 에 대한 Status 를 출력한다.

> Dump Format: (SockId),S/C/U,(SrcPort),(DstIP),(DstPort)↓

> Parameter ■ SockId: Socket Number to display
 ■ S: TCP Server, C: TCP Client, U: UDP
 ■ SrcPort: Source(My) Port Number
 ■ DstIP: Destination IP Address, DstPort: Destination Port Number
 TCP 시 Connected 상태, UDP 시 Default 값이 있을 때 출력된다.

■ **AT+NOPT** (Manage Socket Option)

> 소켓의 Option 을 설정하거나 확인한다.

> Parameter ■ 미정

5. Management Command

Command	Prop.	Input Parameter	Input Resp.	Query Response
AT	? []		[S]	[D,,(Size)]↓(Data)
AT+MSET	= - ? [?] Savable	E/D,F/S/D,(Country)	[S]	[S,,E/D,F/S/D,(Country)]
AT+MSTAT	? [?]			[S,,(Version)]
AT+MEVT	= ? [=]	(Id)		[V,(Id),(EventIdx)] [D,,(Size)]↓(Data)
AT+MUSART	= - ? [?] Savable	(BR),(W),(P),(S),(F)	[S]	[S,,(BR),(W),(P),(S),(F)]
AT+MSPI	= - ? [?] Savable	미정	미정	미정
AT+MPROF	= ? [?]	S/V,(Idx) L,D/(Idx)	S, L 의 경우 [S]	[D,,(Size)]↓(Data)
AT+MRST	= []	F	[S]	

■ AT (Terminal Check)

> Terminal 의 동작을 확인하기 위해서 사용되며, 입력 후 [S] 가 회신되면 정상이다.

'?' 부호를 사용하여 Query 를 하면 직전에 입력한 명령을 회신하며, 이것은 입력 시 누락된 것이 없는 지를 확인하기 위해서 사용할 수 있다.

> Dump Format: 특정 형식 없이 직전 입력 값이 그대로 출력됨.

■ AT+MSET (System Configuration)

> System 의 환경 변수들을 설정한다. Echo 는 입력한 문자를 회신하는 것을 말하며 자신이 입력한 문자를 확인 가능하므로 디버깅 시 유용하다. F/S/D 로 설정이 가능한 Polling Mode 는 D 로 설정 시 Received Data 와 Event 가 발생 즉시 바로 회신되며, S 로 설정 시 Event 는 바로 출력되지만 Received Data 는 AT+NRECV 를 입력해야만 회신된다. 그리고 F 로 설정하면 Received Data 는 AT+NRECV 를 입력해야만 회신되고, Event 또한 AT+MEVT 를 입력해야 회신된다. Received Data 가 있는지 여부는 Event 를 통해서 알 수 있으며, Event 가 있는지의 여부는 지속적으로 AT+MEVT 로 확인을 해야만 알 수 있다. Country 설정은 ISO 3166-1 alpha-2 의 형식에 맞춰 입력해야 하며 내부적으로 국가 설정을 필요로 하는 곳에서 사용된다.

- > Parameter
 - E: Echo Enable, D: Echo Disable
 - F: Full Poll Mode, S: Semi Poll Mode, D: Poll Disable
 - Country: Country Code (ISO 3166-1 alpha-2)

■ AT+MSTAT (View System Status)

- > 시스템의 상태를 회신한다.
- > Parameter
 - Version: FW Version.

■ AT+MEVT (Event Polling)

- > Event Queue 에 저장된 Event 를 회신하며, AT+MSET 에서 Poll Mode 가 Full 로 설정된 경우에 사용된다. Id 가 지정되면 해당 Id 에서 발생한 Event 만 출력하며, 미 지정 시 발생 시간 순으로 출력된다. '?' 부호로 Query 를 하면 해당 Platform 에서 구현된 모든 ID 항목을 회신한다.
- > Dump Format: (Id),(Kind)↓
- > Parameter
 - Id: Item ID like SockId, EventIdx: 해당 ID 의 Event Index
 - Kind: ID 의 분류, Ex) System ID, Socket ID, WiFi ID, ...

■ AT+MUSART (USART Configuration)

- > USART 를 설정하거나 변수 값을 확인한다.
- > Parameter
 - BR: Baud rate, W: Word length
 - P: Parity bit, S: Stop bit, F: Flow Control

■ AT+MSPI (SPI Configuration)

- > SPI 를 설정하거나 변수 값을 확인한다.
- > Parameter
 - 미정

■ AT+MPROF (Profile Configuration)

- > Profile 을 저장하거나(S), 저장된 것을 확인할 수 있고(V), 기존에 저장된 Profile 을 불러올 수 있다(L). 저장 가능한 Profile 의 개수는 Platform 에 맞게 적절히 가능하며, Index 는 0 번부터 시작된다. (그러므로 0 번은 반드시 구현되어야 한다.) 'V' 인자나 Query 를 하여 Profile 을 확인하는 경우 'V'는 개별 Profile 을 확인하는 용도로 사용하며 Query 는 모든 Profile 을 확인하는 경우에 사용된다.
- > Dump Format: (ItemName),(Param1),...↓
시스템마다 환경 변수의 사용 여부가 달라 일률적으로 정할 수는 없지만, 기본적인 서식은 위와 같이 하도록 한다. Ex) MUSART,115200,8,0,1,0↓ NAUTO,0,S,5000↓
- > Parameter
 - V: View, S: Save, L: Load, D: Default, Idx: Profile Index

■ **AT+MRST** (System Reset)

> 시스템 Reset 을 실행한다. 'F' 인자를 사용하여 수행하면 Factory Reset 을 수행한다.

> Parameter ■ F: Factory Reset

6. Application Functions

Command	Prop.	Input Parameter	Input Resp.	Query Response
AT+FDHCPD	미정	미정	미정	미정
AT+FDNS	미정	미정	미정	미정
AT+FPING	미정	미정	미정	미정
AT+FGPIO	미정	미정	미정	미정

■ AT+FDHCPD (Run DHCP Server)

> WiFi 가 AP 로 동작하는 경우 DHCP Server 를 동작시킨다.

> Parameter ■ 미정

■ AT+FDNS (DNS Query)

> 설정된 DNS 주소로 Query 를 수행한다.

> Parameter ■ 미정

■ AT+FPING (Ping Test)

> 지정된 주소로 Ping Test 를 수행한다.

> Parameter ■ 미정

■ AT+FGPIO (GPIO Control)

> 지정된 GPIO 를 Set/Clear 한다.

> Parameter ■ 미정

7. Mode Configuration

■ Polling Mode

> 소켓에 수신된 Data 와 각각의 ID 에서 발생한 Event 의 출력을 어떻게 할 것인지에 대한 설정이며, 각각 동작은 Interrupt 와 Polling 으로 구분이 된다. 회신이 Interrupt 로 동작하게 되면 사용자의 개입 없이 즉각 회신되며, Polling 으로 동작하게 되면 발생 후 일단 Queue 에 보관된 다음, 사용자의 개입에 의하여 순차적으로 회신된다.

> 각 모드 별 Data 수신 시와 Event 발생 시의 동작은 다음과 같다.

- Full Mode: 사용자는 AT+MEVT 를 통하여 발생한 Event 가 없는 지 주기적으로 확인해야 한다. Socket 을 통하여 Data 를 수신한 경우에도 AT+MEVT 를 통하여 수신 여부를 알 수 있으며, 사용자는 Received Event 를 확인한 후 AT+NRECV 명령을 입력하여야만 데이터를 받을 수 있다. 이 모드는 Host 프로그램을 단순하게 구현할 때 적합하며 Interrupt 방식으로 인한 예상치 못한 회신을 고려하지 않아도 되지만, Received Data 와 Event 가 Overflow 될 가능성이 있으므로 주의해야 한다.
- Semi Mode: Event 가 발생하면 즉시 회신되어 사용자 Event 를 확인하기 위하여 AT_MEVT 를 입력할 필요가 없다. Socket 을 통해 Data 가 수신되면 즉시 Event 가 회신되지만 Data 를 받기 위해서는 AT+NRECV 를 입력해야 한다. Default 모드이며, 읽어 오는 Data 의 크기를 지정할 필요가 있을 때 적합하다.
- Disabled: Polling Mode 를 Disable 시켜 Received Data 와 Event 가 Interrupt 방식으로 발생 즉시 회신되므로 AT+NRECV 와 AT+MEVT 는 사용되지 않는다. Host 에서 이 방식에 맞추어 구현하려면 Data 수신을 위한 Buffer 를 최대 값에 맞추어 준비되어야 한다. 수신을 위해 별도의 명령이 필요하지 않으므로 사용자가 Manual 로 명령을 입력하는 경우나 테스트 시에 편리하다.

8. Usage Examples

■ Basic Usage

> Terminal 이 정상 동작하는 것을 확인하고 현재 Network 상태를 확인 후 방금 입력한 명령이 제대로 입력되었는지 확인

- AT↓ Terminal 확인 명령
- [S]↓ Terminal 이 정상적으로 동작하는 것을 확인
- AT+NSTAT?↓ 현재 Network 상태를 Query (이 명령에서는 '?' 가 생략이 가능함)
- [S,,D,192.168.0.100,255.255.255.0,192.168.0.1,8.8.8.8]↓
 현재 DHCP 로 설정되어 있으며, DHCP Server 에 의해 192.168.0.100 의
 주소를 할당 받은 상태인 것을 알 수 있다.
- AT?↓ 직전 입력한 명령을 확인
- [D,,9]↓AT+NSTAT?↓ 방금 입력했던 명령과 같은 지를 확인

■ Initialization

> 입력 Echo 를 Off 로, 국가 설정을 한국으로 설정하고, Network 주소를 DHCP 로 설정

- AT+MSET=D,,KR↓ Echo 를 Off 하고 국가를 한국으로 설정
- AT+NSET=D↓ DHCP 로 전환 설정

> Polling Mode 를 Full Mode 로 설정하고, Network 주소를 Static 으로 다음과 같이 설정
IP: 192.168.0.100, SN: Default 유지, GW: 192.168.0.1, DNS: Default 유지

- AT+MSET=,F↓ 또는 AT+MSET=2,F↓
- AT+NSET=S,192.168.0.100,,192.168.0.1↓

■ Socket Usage

> Socket 을 TCP Server 로 Open 하고 Connected 되면 Data "abc" 를 전송하는 방법
(5000 번 포트 사용)

- AT+NOPEN=S,5000↓ Socket 을 TCP Server 로 Port 5000 을 사용하여 Open
- Sync 시 [S,,1]↓로, Async 시 [W,1]↓ 이 후 [S,1]↓로 동작 성공을 확인 (1 번 소켓 할당)
- [V,1,0]↓ 외부에서 접속하여 1 번 Socket 이 Connected 된 것을 확인한다.
- AT+NSEND=1,3↓ 1 번 Socket 으로 3 Bytes 를 전송

- Sync 인 경우 즉시, Async 의 경우 [W,1]↓을 확인 후에 "abc"를 입력
- Sync 인 경우 [S]↓, Async 의 경우 [S,1]↓ 로 전송 성공을 확인

> Socket 을 TCP Client 로 Open 하고 192.168.0.5:5001 에 Connect 하여 "abc"를 전송하는 방법 (5000 번 포트 사용)

- AT+NOPEN=C,5000,192.168.0.5,5001↓
- Sync 시 [S,,1]↓로, Async 시 [W,1]↓ 이 후 [S,1]↓로 동작 성공을 확인 (1 번 소켓 할당)
- AT+NSEND=1,3↓ 1 번 Socket 으로 3 Bytes 를 전송
- Sync 인 경우 즉시, Async 의 경우 [W,1]↓을 확인 후에 "abc"를 입력
- Sync 인 경우 [S]↓, Async 의 경우 [S,1]↓ 로 전송 성공을 확인

> Socket 을 UDP 로 Open 하여 192.168.0.5:5001 로 Data "abc"를 전송하는 방법 (5000 번 포트 사용)

- AT+NOPEN=U,5000↓
- Sync 시 [S,,1]↓로, Async 시 [W,1]↓ 이 후 [S,1]↓로 동작 성공을 확인 (1 번 소켓 할당)
- AT+NSEND=1,3,192.168.0.5,5001↓ 1 번 Socket 으로 해당 주소에 3 Bytes 를 전송
- Sync 인 경우 즉시, Async 의 경우 [W,1]↓을 확인 후에 "abc"를 입력
- Sync 인 경우 [S]↓, Async 의 경우 [S,1]↓ 로 전송 성공을 확인

> Socket 을 UDP 로 Open 하고 Default 로 192.168.0.5:5001 를 설정한 다음 해당 주소로 "abc"를 전송한 다음 192.168.0.6:5001 로 "abc"를 전송하는 방법 (5000 번 포트 사용)

- AT+NOPEN=U,5000,192.168.0.5,5001↓
- Sync 시 [S,,1]↓로, Async 시 [W,1]↓ 이 후 [S,1]↓로 동작 성공을 확인 (1 번 소켓 할당)
- AT+NSEND=1,3↓ 1 번 Socket 으로 Default 주소에 3 Bytes 를 전송
- Sync 인 경우 즉시, Async 의 경우 [W,1]↓을 확인 후에 "abc"를 입력
- Sync 인 경우 [S]↓, Async 의 경우 [S,1]↓ 로 전송 성공을 확인
- AT+NSEND=1,3,192.168.0.6,5001↓ 1 번 Socket 으로 해당 주소에 3 Bytes 를 전송
- Sync 인 경우 즉시, Async 의 경우 [W,1]↓을 확인 후에 "abc"를 입력
- Sync 인 경우 [S]↓, Async 의 경우 [S,1]↓ 로 전송 성공을 확인