MODBUS Organization

MODBUS/TCP Server

지원 버전 TOP Design Studio V1.0 이상



CONTENTS

Touch Operation Panel을 사용해주시는 고객님께 감사 드립니다.

- 1. 시스템 구성
 2 페이지

 연결 가능한 기기 및 네트워크 구성에 대해 설명합니다.
- 2. 외부 장치 선택
 3 페이지

 TOP의 기종과 외부 장치를 선택합니다.
- 3. TOP 통신 설정
 4 페이지

 TOP 통신 설정 방법에 대해서 설명합니다.

 4. 지원 어드레스
 10 페이지



1. 시스템 구성

본 드라이버는 TOP가 MODBUS/TCP 서버 기능을 추가하여 동작하도록 합니다

외부 장치	통신 방식	시스템 설정	케이블
MODBUS/TCP Client	Ethernet (TCP)	<u>3. TOP 통신 설정</u>	트위스트 페어 케이블 * <mark>주</mark> 1)

*주1) 트위스트 페어 케이블

- STP(실드 트위스트 페어 케이블) 혹은 UTP(비실드 트위스트 페어 케이블) 카테고리 3,4,5 를 의미 합니다.

- 네트 워크 구성에 따라 허브, 트랜시버 등의 구성기기에 접속 가능하며 이 경우 다이렉트 케이블을 사용 하십시오.

■ 연결 가능 구성

•N:N 연결





2. 외부 장치 선택

■ TOP 모델 및 포트 선택 후 외부 장치를 선택합니다.



설정	사항	내용		
ТОР	모델	TOP의 디스플레이와 프로세스를 확인하여 터치 모델을 선택합니다.		
	제조사	TOP와 연결할 외부 장치의 제조사를 선택합니다. "MODBUS Organization"를 선택 하십시오.		
		TOP와 연결할 외부 장치를 선택 합니다.		
외부 장치		모델	인터페이스	프로토콜
	PLC	MODBUS Slave	Ethernet	MODBUS TCP
		연결을 원하는 외부 장치가 시스 바랍니다.	스템 구성 가능한 기종인지 1장!	의 시스템 구성에서 확인 하시기



3. TOP 통신 설정

통신 설정은 TOP Design Studio 혹은 TOP 메인 메뉴에서 설정 가능 합니다. 통신 설정은 외부 장치와 동일하게 설정해야 합니다.

3.1 TOP Design Studio 에서 통신 설정

- (1) 통신 인터페이스 설정
 - [프로젝트] → [속성] → [TOP 설정] → [HMI 설정] → [HMI 설정 사용 체크] → [편집] → [이더넷]
 - TOP 통신 인터페이스를 TOP Design Studio에서 설정합니다.

프로젝트 옵션		X
HMI 변경	📁 PLC 추가 🎢 RLC 변경 🔀 PLC 삭제	
▼ ·□ TOP 설정 STS : RD1520X	날짜 / 시간 동기화 화면 옵션 단위 변환	
✓ - ○ 디바이스 설정 ○ - ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	프로젝트 옵션 회면 전환 HMI 설정 글로벌 잠금 옵션 프	또로젝트 스타일 스플래시 PLC 버퍼 동기화
COM2 (0)	· → HMI 설정 사용	
Ethernet (1)	HMI 설정 음션 Slave Project Setting	초기화 편집
USBDevice (0)	HMIDisable=0 HMIDisable=0 Project Name=New Project Start Mode=Menu Start Soreen No. = 1 Latch Use=0 Latch Set=0-0 Communication Error Message=0 USBErorMessage=0 DisplayPermissionLcon=0 DarkDraw=0 TOPID=1 AutoRun=0 Language=English Project Advanced Setting HMIDisable=0 RunSleep=50 CommunicationSleep=1 ProjectLoadType=1 IniDiaPrevScreen=1 ErrorLogWite=0 BlinkTime=0 WatchDog=0 SCoverOption=0 TaskMoritor=1 CommiSleepChgUse=0 CommiSleepChgUs	
		적용 닫기
거판		🕋 이더넷 🛛 🕹 🗡
🔯 시스템 🛛 🔤 경	치 📮 서비스 🕋 옵션	포트
		이더넷 포트 : ETH1 ▼ 0 ▼
		링크 속도 : 자동 ▼
PLC 보안	날짜/시간 디스플레이 터치 사운드	 맥 주소 : 00:00:00:00:00:00
		IP 주소 : 192.168.0.100
· • • • • • • • • • • • • • • • • • • •		서브넷 마스크 : 255.255.2
이더넷 시리얼	HDMI 초기화 센서 Wi-Fi	게이트웨이 : 192.168.0.1
		DNS (1) :
intil V		DNS (2) :
자가 진단 파일 관리자	· /Ping 상단 메뉴 핑 키페드 MRAM 분석 옵션	이더넷
관리자	옵션	기본 IP : 192.168.0.100 ▼
[System]		케이블 상태 :
	메뉴 활성화 HMI 설정 가져오기 확인 취소	브릿지 모드 : 🗌 브릿지 사용
		충돌 확인 적용 취소
		62 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7



항 목	ТОР	외부 장치	비고
IP 주소	192.168.0.100	192.168.0.50	
서브넷 마스크	255.255.255.0	255.255.255.0	
게이트 웨이	192.168.0.1	192.168.0.1	

※ 위의 설정 내용은 본 사에서 권장하는 <u>예제</u>입니다.

항 목	설명
IP 주소	TOP의 IP 주소를 설정합니다.
서브넷 마스크	네트워크의 서브넷 마스크를 입력합니다.
게이트 웨이	네트워크의 게이트 웨이를 입력합니다.



(2) 통신 옵션 설정

■ [프로젝트] → [프로젝트 속성] → [PLC 설정 > Ethernet > PLC1 : MODBUS Slave]
 – MODBUS Slave 통신 드라이버의 옵션을 TOP Design Studio에서 설정합니다.

그 SYS: RD 1520X 부단 2 3 (NODBUS State) 응선 모들 성정 별칭: [PLC1 바인드 IP: Auto × RFID (0) 인터페이스: [thernet × 다바이스 성정 프로토콜: MODBUS TCP × COM1 (0) 프로토콜: MODBUS TCP × COM2 (0) 프로토퍼 100	젝트 옵션	×
□ SYS: RD 1520X ● FLC 2 = (MODBUS Slave] ● Friedbus (0) ● FID (0) □ Firledbus (0) ● Ethernet ○ COM1 (0) ● COM2 (0) ○ COM2 (0) ● OSA 48 ● PICC1: MODBUS Slave ● OSA 48 ● USBDevice (0) ● OSA 48 ● Primary Option Ethernet Protocol Ethernet Protocol TCP ∨ HMI Port 502 € Address Mode ● Base ∨	HMI 변경 PLC 추가 해외 PLC 변경 🔀 PLC 삭제	
	● 2년 모를 설정 ● PLC1 ● 바인드 IP : Auto ▼ ● 2년 모를 설정 ● PLC1 ● 바인드 IP : Auto ▼ ● COM3 (0) ● Etherost (1) ● OSBUS Slave ● Wreless (0) ● USBDevice (0) ● Primary Option ● Ethernet Protocol ● Primary Option ● Ethernet Protocol ● ITOP ▼ ● HIL Prot S02 ● Ethernet Protocol ■ 조건 ● Primary Option ● Ethernet Protocol ● Ethernet Protocol ■ CP ▼ ● HIL Port S02 ● ● Address Mode ● Base ▼	통신 매뉴얼

항 목	설 정	비고
인터페이스	"Ethernet"을 선택합니다.	<u>"2. 외부 장치</u>
프로토콜	TOP — 외부 장치 간 통신 프로토콜을 선택합니다.	<u> 선택" 참고</u>
Ethernet Protocol	외부 장치의 IP 주소를 입력 합니다.	
HMI Port	TOP의 모드버스 통신 포트 번호를 설정합니다.	
Address Mode	모드버스 PDU Address의 -1 여부를 설정합니다.	*주1)

*주1) 클라이언트의 사양에 맞게 설정하십시오.

TOP의 SYS0 데이터를 읽기 위해 주소 0을 요청하면 0-Base 선택. TOP의 SYS0 데이터를 읽기 위해 주소 1을 요청하면 1-Base 선택.



3.2 TOP에서 통신 설정

※ "3.1 TOP Design Studio 에서 통신 설정" 항목의 "HMI 설정 사용"을 체크 하지 않은 경우의 설정 방법입니다.

■ TOP 화면 상단을 터치하여 아래로 <u>드래그</u> 합니다. 팝업 창의 "EXIT"를 터치하여 메인 화면으로 이동합니다.



- (1) 통신 인터페이스 설정
 - [제어판] → [이더넷]

	🆚 제어판	🛆 이더넷	× ×	
실행 VNC 뷰 DI	() 시스템 () () <th>Port Ethernet Port : ETH1 ▼ 0 Link Speed : Auto MAC Address : 90:9F:33:0A:EC:F9 IP Address : 192.168.0.100 Subnet Mask : 255.255.255.0 Gateway : 192.168.0.1 DNS (1) : DNS (2) :</th> <th></th> <th></th>	Port Ethernet Port : ETH1 ▼ 0 Link Speed : Auto MAC Address : 90:9F:33:0A:EC:F9 IP Address : 192.168.0.100 Subnet Mask : 255.255.255.0 Gateway : 192.168.0.1 DNS (1) : DNS (2) :		
스크린샷	자가 진단 파일 [System]	이더넷 Primary IP : 192.168.0.100 Cable Status : Bridge Mode : □Use Bridge Check duplicate 적용 취소		

항 목	ТОР	외부 장치	비고
IP 주소	192.168.0.100	192.168.0.50	
서브넷 마스크	255.255.255.0	255.255.255.0	
게이트 웨이	192.168.0.1	192.168.0.1	

[※] 위의 설정 내용은 본 사에서 권장하는 <u>예제</u>입니다.

항 목	설 명
IP 주소	TOP의 IP 주소를 설정합니다.
서브넷 마스크	네트워크의 서브넷 마스크를 입력합니다.
게이트 웨이	네트워크의 게이트 웨이를 입력합니다.



(2) 통신 옵션 설정

■ [제어판] → [PLC]

🔹 제어판	Drvier(ETH) PLC1(MODBUS Slave) -	
	Protocol MODBUS TCP Bind IP Auto	
PLC	Ethernet Proto, TCP - HMI Port 502 -	
<u></u>	Address Mode O-Base -	
이더넷		
HTT-7		
자가 진단		
[System	통신 진단	
	취소 적용	
_		

※ 위의 설정 내용은 본 사에서 권장하는 <u>예제</u>입니다.

항 목	설 정	비고
인터페이스	"Ethernet"을 선택합니다.	"2. 외부 장치
프로토콜	TOP — 외부 장치 간 통신 프로토콜을 선택합니다.	<u> 선택" 참고</u>
Ethernet Protocol	외부 장치의 IP 주소를 입력 합니다.	
HMI Port	HMI의 모드버스 통신 포트 번호를 설정합니다.	
Address Mode	모드버스 PDU Address의 -1 여부를 설정합니다.	*주1)

*주1) 클라이언트의 사양에 맞게 설정하십시오.

TOP의 SYS0 데이터를 읽기 위해 주소 0을 요청하면 0-Base 선택. TOP의 SYS0 데이터를 읽기 위해 주소 1을 요청하면 1-Base 선택.



3.3 통신 진단

본 드라이버는 통신 진단을 지원하지 않습니다.

클라이언트에서 접속과 데이터 읽기 요청을 시도하여 통신 연결을 확인하십시오. 주의) TOP가 실행(Run) 중이어야 합니다.



4. 지원 어드레스

TOP에서 지원하는 데이터에 대해 설명합니다.

주소	비트	워드	비고
SYS	0.0 ~ 10239.15	0 ~ 10239	*주1)

*<mark>주1)</mark> TOP-VIEW는 0~65535를 지원합니다.

※ TOP 내부 메모리 → 모드버스 데이터 모델링

TOP 내부 메모리를 모드버스 데이터로 표현하면 Holding Register에 해당합니다. 명령어 0x03으로 읽을 수 있으며 명령어 0x06, 0x10으로 값을 변경 할 수 있습니다. Coil, Discrete Input, Input Register를 접근하는 명령어도 지원하지만 명령어가 다르더라도 결국 같은 메모리 영역(TOP 내부 메모리)을 접근합니다.

■ 지원 명령어

Code (hex)	Descriptions
01	Read Coils
02	Read Discrete Inputs
03	Read Holding Registers
04	Read Input Registers
05	Write Single Coil
06	Write Single Register
OF	Write Multiple Coils
10	Write Multiple Registers

WHAT IS MODBUS?

The MODBUS protocol was developed in 1979 by Modicon, Incorporated, for industrial automation systems and Modicon programmable controllers. It has since become an industry standard method for the transfer of discrete/analog I/O information and register data between industrial control and monitoring devices. MODBUS is now a widely-accepted, open, public-domain protocol that requires a license, but does not require royalty payment to its owner.

MODBUS devices communicate using a master-slave (client-server) technique in which only one device (the Client(Master)) can initiate

transactions (called queries). The other devices (slaves/servers) respond by supplying the requested data to the master, or by taking the action requested in the query. A slave is any peripheral device (I/O transducer, valve, network drive, or other measuring device) which processes information and sends its output to the master using MODBUS. The Acromag I/O Modules form slave/server devices, while a typical master device is a host computer running appropriate application software. Other devices may function as both clients (masters) and servers (slaves).

Masters can address individual slaves, or can initiate a broadcast message to all slaves. Slaves return a response to all queries addressed to them individually, but do not respond to broadcast queries. Slaves do not initiate messages on their own, they only respond to queries from the master.

A master's query will consist of a slave address (or broadcast address), a function code defining the requested action, any required data, and an error checking field. A slave's response consists of fields confirming the action taken, any data to be returned, and an error checking field. Note that the query and response both include a device address, a function code, plus applicable data, and an error checking field. If no error occurs, the slave's response contains the data as requested. If an error occurs in the query received, or if the slave is unable to perform the action requested, the slave will return an exception message as its response (see MODBUS Exceptions). The error check field of the slave's message frame allows the master to confirm that the contents of the message are valid. Traditional MODBUS messages are transmitted serially and parity checking is also applied to each transmitted character in its data frame.

At this point, It's important to make the distinction that MODBUS itself is an application protocol, as it defines rules for organizing and interpreting data, but remains simply a messaging structure, independent of the underlying physical layer. As it happens to be easy to understand, freely available, and accessible to anyone, it is thus widely supported by many manufacturers.



WHAT IS MODBUS TCP/IP?

MODBUS TCP/IP (also MODBUS-TCP) is simply the MODBUS RTU protocol with a TCP interface that runs on Ethernet.

The MODBUS messaging structure is the application protocol that defines the rules for organizing and interpreting the data independent of the data transmission medium.

TCP/IP refers to the Transmission Control Protocol and Internet Protocol, which provides the transmission medium for MODBUS TCP/IP messaging.

Simply stated, TCP/IP allows blocks of binary data to be exchanged between computers. It is also a world-wide standard that serves as the foundation for the World Wide Web. The primary function of TCP is to ensure that all packets of data are received correctly, while IP makes sure that messages are correctly addressed and routed. Note that the TCP/IP combination is merely a transport protocol, and does not define what the data means or how the data is to be interpreted (this is the job of the application protocol, MODBUS in this case).

So in summary, MODBUS TCP/IP uses TCP/IP and Ethernet to carry the data of the MODBUS message structure between compatible devices. That is, MODBUS TCP/IP combines a physical network (Ethernet), with a networking standard (TCP/IP), and a standard method of representing data (MODBUS as the application protocol). Essentially, the MODBUS TCP/IP message is simply a MODBUS communication encapsulated in an Ethernet TCP/IP wrapper.

In practice, MODBUS TCP embeds a standard MODBUS data frame into a TCP frame, without the MODBUS checksum, as shown in the following diagram.



Modbus TCP/IP ADU

(This information is embedded into the data portion of the TCP frame)

The MODBUS commands and user data are themselves encapsulated into the data container of a TCP/IP telegram without being modified in any way. However, the MODBUS error checking field (checksum) is not used, as the standard Ethernet TCP/IP link layer checksum methods are instead used to guaranty data integrity. Further, the MODBUS frame address field is supplanted by the unit identifier in MODBUS TCP/IP, and becomes part of the MODBUS Application Protocol (MBAP) header (more on this later).

From the figure, we see that the function code and data fields are absorbed in their original form. Thus, a Modbus TCP/IP Application Data Unit (ADU) takes the form of a 7 byte header (transaction identifier + protocol identifier + length field + unit identifier), and the protocol data unit (function code + data). The MBAP header is 7 bytes long and includes the following fields:

• **Transaction/invocation Identifier (2 Bytes):** This identification field is used for transaction pairing when multiple messages are sent along the same TCP connection by a client without waiting for a prior response.

• Protocol Identifier (2 bytes): This field is always 0 for MODBUS services and other values are reserved for future extensions.

• Length (2 bytes): This field is a byte count of the remaining fields and includes the unit identifier byte, function code byte, and the data fields.

• Unit Identifier (1 byte): This field is used to identify a remote server located on a non TCP/IP network (for serial bridging). In a typical MODBUS TCP/IP server application, the unit ID is set to 00 or FF, ignored by the server, and simply echoed back in the response.

The complete MODBUS TCP/IP Application Data Unit is embedded into the data field of a standard TCP frame and sent via TCP to well-known system port 502, which is specifically reserved for MODBUS applications. MODBUS TCP/IP clients and servers listen and receive MODBUS data via port 502.

We can see that the operation of MODBUS over Ethernet is nearly transparent to the MODBUS register/command structure. Thus, if you are already familiar with the operation of traditional MODBUS, then you are already very with the operation of MODBUS TCP/IP.

(1) Read Single Coil: 01

MASTER 기기에서 Slave 기기 측(국번:17번)의 "000020-000056 Coil" 데이터를 읽어 오는 예제를 통해 "01"명령어 프레임을 설명 합니다.

RTU Mode



0: OFF /

High

 FF_H

00_H

Low

 00_{H}

00_H

(2) Force Single Coil: 05

MASTER 기기에서 Slave 기기 측의 Coil 000173 에 FORCE "ON" 하는 예제를 통해 "05"명령어 프레임을 설명 합니다.

RTU Mode (Master → Slave : 요청 프레임) ►∎ Force Data Length 명령어 Force data Comment Transaction Identifier Protocol Identifier Unit ID (Slave rfr五小 선두디바이스 Force ON ı Field Force OFF Н L н L Н L Н L Н L 00 00 00 00 06 05 00 AC FF 00 Hex 02 11 (Slave → Master : 응답 프레임) Length Unit ID (Slave rfr五) 명령 어 Comment Transaction Identifier Protocol Identifier Force data 선두디바이 Field 스 Н Н L Н L L L н Н L 02 00 00 00 06 11 05 00 AC FF 00 Hex 00



(1) Read Input Status : 02

MASTER 기기에서 Slave 기기 측(국번:17번)의 "**1**00197-**1**00218 Input" 데이터를 읽어 오는 예제를 통해 "02"명령어 프레임을 설명 합니다.

	(1	Master	r → SI	ave :	요청 3	뜨레임)															
Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave 叶玉)	80 80 80	선두디바이스		디바이 <u>스</u> 점 수											
	н	L	н	L	Н	L			н	L	н	L										
Hex	00	03	00	00	00	06	11	02	00	C4	00	16										
(Slave	→ Ma	ster :	응답	프레임	4)								_									
Cor	Tra		Pro		Len		Unit	명권	데	(H0	터(Inp	uts			;데이티	너 상태						
Comment	Transaction Identifier		Protocol Identifier		Length Field	-	Ð	명 평 어	이터		 	_		Coils	204	203	202	201	200	199	198	197
nt	lion		Ide		Fielc	! -	(Slave rfr五)		갯	102	10212	10218		on/off	1	0	1	0	1	1	0	0
	Iden		ntifi		_	•	/e 구		가 슈(byte)	10204~10197	212~	218~		Coils	212	211	210	209	208	207	206	205
	tifie		er.				번		te)	101	~10205	1021		on/off	1	1	0	1	1	0	1	1
	-									97	05	13		Coils	220	219	218	217	216	215	214	213
	н	L	н	L	н	L								on/off	-	-	1	1	0	1	0	1
Hex	00	03	00	00	00	06	11	02	03	AC	DB	35								(): OFF /	(1:ON



(1) Read Input Registers : 04

MASTER 기기에서 Slave 기기 측(국번:17번)의 "**3**00009 Register" 데이터를 읽어 오는 예제를 통해 "03"명령어 프레임을 설명 합니 다.

	(Mastei	r → Sl	ave :	요청 표	뜨레임)						
Comment	Transaction Identifier		Protocol Identifier Transaction Identifier			Length Field		8080で Unit ID (Slave 作五)		선두디 바이 스		(Word Count)	다. 바이스템수
	н	L	Н	L	Н	L			н	L	н	L	
Hex	00	04	00	00	00	06	11	04	00	08	00	01	
	(9	Slave -	→ Mas	ster :	응답 Ξ	뜨레임)						
Comment			Protocol Identifier			Unit ID (Slave ரு. த.)	80 80 80	ㅠㅇㅍ ㅗ슈byte)	년 30009) 더 Register			
	н	L	н	L	н	L				Н	L		
Hex	00	04	00	00	00	05	11	04	02	00	0A		



(1) Read Holding Registers : 03

MASTER 기기에서 Slave 기기 측(국번:17)의 "400108 - 400110 Register" 데이터를 읽어 오는 예제를 통해 "03"명령어 프레임을 설 명 합니다.

	(Maste	r → Sl	ave :	요청 3	프레임)										
Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave rf 五)	평 평 정	선두 디 바 이 스		디 바이 스점 수						
	Н	L	Н	L	Н	L			н	L	H.	L					
Hex	00	05	00	00	00	06	11	03	00	6B	00	03					
(Slave	→ Ma	ster :	응답	프레임	님)												
Comment	Transact		Protoco	1	Length Field		Unit ID	명 평장	데 이 터			데이	티	1			
nt	Transaction Identifier	Protocol Identifier		-	Field		Unit ID (Slave rfr五)		개 代byte)	40108	Register	40109	Register	40110	Register		
	н	L	н	L	н	L				н	L	н	L	н	L		
Hex	00	05	00	00	00	09	11	03	06	02	2B	00	00	00	64		

(2) Preset Single Register : 06

Slave 기기 측의 400002 Register 에 00 03 (hex) 데이터를 입력 하는 예제를 통해 "06"명령어 프레임을 설명 합니다.

(Master → Slave : 요정 프레임)													
Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave rfr五)	ᄧᇒᅎ	선두디바이스		Preset data		
	Н	L	Н	L	Н	L			Н	L	н	L	
Hex	00	06	00	00	00	06	11	06	00	01	00	03	
	(9	Slave ·	→ Ma	ster :	응답	프레읻)						
Comment	Protocol Identifier Transaction Identifier			Length Field		Unit ID (Slave rfr五小	あると	선두디 바이 스		Preset data			
	н	L	н	L	н	L			Н	L	н	L	
Hex	00	06	00	00	00	06	11	06	00	01	00	03	

(Master → Slave : 요청 프레임)



(3) Preset Multiple Register : 10

Slave 기기 측의 **4**00002 Register 에 "00 0A (hex)", "01 02 (hex)" 연속한 두 개의 데이터를 입력 하는 예제를 통해 "10"명령어 프레 임을 설명 합니다. (Error Code : 90_H)

(Master \rightarrow	Slave :	요청	프레임)
-----------------------	---------	----	-----	---

					-	_ " _	. ,												
Comment	Trans	I	Proto	1	Leng		Unit	명찍오	선 두		(Wor	Quan	데 이 터		데이	터			
nent	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave rfr五)	어	선 두 다 바이 스		(Word Count)	Quantity of Register (Word Count)		ty of Register Count)		40002	Register	40003	Register
	н	L	н	L	н	L			н	L	н	L		н	L	н	L		
Hex	00	07	00	00	00	0B	11	10	00	01	00	02	04	00	0A	01	02		
	()	Slave	→ Ma	ster :	응답	프레읻])	_	_		_								
Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave 作五)	명 평 정	선 두 다 바이 스		(Word Count)	Quantity of Register							
	н	L	н	L	н	L			н	L	н	L							
Hex	00	07	00	00	00	06	11	10	00	01	00	02							