

ALT-ESTP DLL Manual

Version : 2019-10-01

Version : 2020-06-10

Version : 2020-09-29

Protocol Load 를 기본적으로 Enable 합니다. Disable 할 수는 없습니다.

목 차

1. ALT_ESTP_Create
2. ALT_ESTP_Close
3. ALT_ESTP_Lan_Start
4. ALT_ESTP_Lan_WStart
5. ALT_ESTP_Uart_Check
6. ALT_ESTP_Uart_WCheck
7. ALT_ESTP_Uart_Start
8. ALT_ESTP_Uart_WStart
9. ALT_ESTP_Setup
10. ALT_ESTP_GetLastErrorCode
11. ALT_ESTP_IsTcpConnect
12. ALT_ESTP_IsUdpConnect
13. ALT_ESTP_IsUartConnect
14. ALT_ESTP_Protocol_Load
15. ALT_ESTP_Protocol_LoadW
16. ALT_ESTP_SendIPAddress
17. ALT_ESTP_ReadIPAddress
18. ALT_ESTP_SendMACAddress
19. ALT_ESTP_ReadMACAddress
20. ALT_ESTP_SendMacroData
21. ALT_ESTP_ReadMacroData
22. ALT_ESTP_StrobeValueSend
23. ALT_ESTP_AllStrobeValueSend
24. ALT_ESTP_AllStrobeValueRead
25. ALT_ESTP_TriggerModeSend
26. ALT_ESTP_EncoderModeSend
27. ALT_ESTP_SaveSend
28. ALT_ESTP_LoadSend
29. ALT_ESTP_EncClearSend
30. ALT_ESTP_EncCountRead
31. ALT_ESTP_RunCodeSend
32. ALT_ESTP_PageIndexSend
33. ALT_ESTP_PageIndexRead
34. ALT_ESTP_SoftTriggerSend
35. ALT_ESTP_ConfigModeRead

36. ALT_ESTP_EncOffsetListSend

37. ALT_ESTP_EncOffsetListRead

1. ALT_ESTP_Create	
함수 원형	int ALT_ESTP_Create (void) ;
동작 설명	ALT-ESTP 제품을 위한 Interface 객체를 생성합니다. 제품을 사용하기 위해서 제일 먼저 호출해야 하는 함수입니다.
전달 인자	void
리턴 값	int : ALT_ESTP 제품에 대한 Interface 객체 번호를 Return 합니다. 객체 번호는 0 부터 시작하고 호출하는 모든 함수에서 객체 구분자로 사용됩니다.
부가 설명	ALT_ESTP_Create() 를 호출한 수만큼 ALT_ESTP_Close() 을 호출하여 생성된 객체를 제거해 주어야 합니다. void ALT_ESTP_Close (int Index) ;

2. ALT_ESTP_Close	
함수 원형	void ALT_ESTP_Close (int Index) ;
동작 설명	ALT_ESTP_Create 함수에서 생성된 Interface 객체를 제거합니다.
전달 인자	int Index : ALT_ESTP_Create 함수에서 생성된 객체 번호를 전달합니다.
리턴 값	void :
부가 설명	int ALT_ESTP_Create (void) 함수에서 생성된 Interface 객체를 제거합니다.

3. ALT_ESTP_Lan_Start	
함수 원형	BOOL ALT_ESTP_Lan_Start (int Index, BOOL TCPIP, char *lpServer, int port, char ChannelNum) ;
동작 설명	Interface 객체번호를 사용해서 LAN 통신 포트를 OPEN 합니다.
전달 인자	int Index : interface 객체 번호 BOOL TCPIP : Lan 통신 모드. TRUE:TCPIP, FALSE:UDP char *lpServer : Server IP Address. “192.168.10.10” 의 문자열 데이터 전달. int port : Lan Port 번호 char ChannelNum : ESTP 제품의 채널 수 (4, 8, 16, 32, 64)
리턴 값	BOOL : TRUE : Ethernet 통신이 연결되어 있습니다. UDP 통신의 경우에는 물리적인 연결을 확인하지 않습니다. FALSE : Ethernet 통신을 연결할 수 없습니다. Server IP Address 주소와 Port 번호를 찾지 못하면 시스템에 따라서 수십초 대기할 수 있습니다. UDP 통신에서는 Address 와 Port 존재 여부에 관계없이 TRUE 리턴합니다.
부가 설명	ALT_ESTP_Close 함수를 호출하면 OPEN 된 통신 포트도 CLOSE 하고 Interface 객체도 제거합니다.

4. ALT_ESTP_Lan_WStart	
함수 원형	BOOL ALT_ESTP_Lan_WStart (int Index, BOOL TCPIP, wchar_t *lpServer, int port, char ChannelNum) ;
동작 설명	Interface 객체번호를 사용해서 LAN 통신 포트를 OPEN 합니다.
전달 인자	int Index : interface 객체 번호 BOOL TCPIP : Lan 통신 모드. TRUE:TCPIP, FALSE:UDP wchar_t *lpServer : Server IP Address. 유니코드 L“192.168.10.10” 의 문자열 데이터 전달. int port : Lan Port 번호 char ChannelNum : ESTP 제품의 채널 수 (4, 8, 16, 32, 64)
리턴 값	BOOL : TRUE : Ethernet 통신이 연결되어 있습니다. UDP 통신의 경우에는 물리적인 연결을 확인하지 않습니다. FALSE : Ethernet 통신을 연결할 수 없습니다. Server IP Address 주소와 Port 번호를 찾지 못하면 시스템에 따라서 수십초 대기할 수 있습니다. UDP 통신에서는 Address 와 Port 존재 여부에 관계없이 TRUE 리턴합니다.
부가 설명	ALT_ESTP_Close 함수를 호출하면 OPEN 된 통신 포트도 CLOSE 하고 Interface 객체도 제거합니다.

5. ALT_ESTP_Uart_Check	
함수 원형	BOOL ALT_ESTP_Uart_Check (int Index, char *PortName) ;
동작 설명	Interface 객체번호를 사용해서 Uart Port 가 유효한지 확인합니다.
전달 인자	int Index : interface 객체 번호 char *PortName : Uart Port 번호가 포함된 문자열 데이터를 전달합니다. 문자열은 “ WWW.WWCOM7 ” 형태로 전달합니다.
리턴 값	BOOL : TRUE : Uart Port 를 사용할 수 있습니다. FALSE : Uart Port 가 없는 번호이거나 이미 다른 프로그램에서 사용 중임을 나타냅니다.
부가 설명	사용가능한 Uart Port 리스트를 만들 때 사용할 수 있습니다. ALT_ESTP_Close 함수를 호출하면 OPEN 된 통신 포트도 CLOSE 하고 Interface 객체도 제거합니다.

6. ALT_ESTP_Uart_WCheck	
함수 원형	BOOL ALT_ESTP_Uart_WCheck (int Index, wchar_t *PortName) ;
동작 설명	Interface 객체번호를 사용해서 Uart Port 가 유효한지 확인합니다.
전달 인자	int Index : interface 객체 번호 wchar_t *PortName : Uart Port 번호가 포함된 문자열 데이터를 전달합니다. 문자열은 유니코드 L" www.wwcom7 " 형태로 전달합니다.
리턴 값	BOOL : TRUE : Uart Port 를 사용할 수 있습니다. FALSE : Uart Port 가 없는 번호이거나 이미 다른 프로그램에서 사용 중임을 나타냅니다.
부가 설명	사용가능한 Uart Port 리스트를 만들 때 사용할 수 있습니다. ALT_ESTP_Close 함수를 호출하면 OPEN 된 통신 포트도 CLOSE 하고 Interface 객체도 제거합니다.

7. ALT_ESTP_Uart_Start	
함수 원형	BOOL ALT_ESTP_Uart_Start (int Index, char *PortName, int Borate, char ChannelNum) ;
동작 설명	Interface 객체번호를 사용해서 Uart Port 를 OPEN 합니다.
전달 인자	int Index : interface 객체 번호 char *PortName : Uart Port 번호가 포함된 문자열 데이터를 전달합니다. 문자열은 " www.wocom7 " 형태로 전달합니다. int Borate : 통신 속도를 전달합니다. 9600, 19200, 38400, 57600, 115200 중에서 선택합니다. char ChannelNum : ESTP 제품의 채널 수 (4, 8, 16, 32, 64)
리턴 값	BOOL : TRUE : Uart Port 를 사용할 수 있습니다. FALSE : Uart Port 가 없는 번호이거나 이미 다른 프로그램에서 사용 중임을 나타냅니다.
부가 설명	ALT_ESTP_Close 함수를 호출하면 OPEN 된 통신 포트도 CLOSE 하고 Interface 객체도 제거합니다.

8. ALT_ESTP_Uart_WStart	
함수 원형	BOOL ALT_ESTP_Uart_WStart (int Index, wchar_t *PortName, int Borate, char ChannelNum) ;
동작 설명	Interface 객체번호를 사용해서 Uart Port 를 OPEN 합니다.
전달 인자	int Index : interface 객체 번호 wchar_t *PortName : Uart Port 번호가 포함된 문자열 데이터를 전달합니다. 문자열은 유니코드 L" www.wwcom7 " 형태로 전달합니다. int Borate : 통신 속도를 전달합니다. 9600, 19200, 38400, 57600, 115200 중에서 선택합니다. char ChannelNum : ESTP 제품의 채널 수 (4, 8, 16, 32, 64)
리턴 값	BOOL : TRUE : Uart Port 를 사용할 수 있습니다. FALSE : Uart Port 가 없는 번호이거나 이미 다른 프로그램에서 사용 중임을 나타냅니다.
부가 설명	ALT_ESTP_Close 함수를 호출하면 OPEN 된 통신 포트도 CLOSE 하고 Interface 객체도 제거합니다.

9. ALT_ESTP_Setup	
함수 원형	BOOL ALT_ESTP_Setup (int Index, char ChannelNum) ;
동작 설명	Interface 객체번호를 사용해서 해당 객체의 채널 개수를 변경합니다.
전달 인자	int Index : interface 객체 번호 char ChannelNum : ESTP 제품의 채널 수 (4, 8, 16, 32, 64)
리턴 값	BOOL : TRUE : 채널수가 변경되었습니다. FALSE : 잘못된 객체번호 혹은 채널 수가 전달되어 변경이 실패하였습니다.
부가 설명	

10. ALT_ESTP_GetLastErrorCode	
함수 원형	unsigned long ALT_ESTP_GetLastErrorCode (int Index) ;
동작 설명	Interface 객체번호를 사용해서 마지막으로 실패한 기능의 Error Code 를 전달받습니다.
전달 인자	int Index : interface 객체 번호
리턴 값	unsigned long : 마지막으로 수행한 실패한 기능의 Error Code 를 전달받습니다. Error Code 의 내용은 다음과 같습니다. <pre> #define ESTP_PARAMETER_VALUE_ERROR (DWORD)0x00000100 #define ESTP_MEM_ALLOCATION_ERROR (DWORD)0x00000200 #define ESTP_SYNC_OBJECT_CREATE_ERROR (DWORD)0x00000300 #define ESTP_RECEIVE_THREAD_CREATE_ERROR (DWORD)0x00000400 #define ESTP_RECEIVE_PROTOCOL_TIMEOUT (DWORD)0x00000500 #define ESTP_RECEIVE_NOT_EXIST_COMMAND_ERROR (DWORD)0x00000600 #define ESTP_PARAMETER_EXCEPTION_ERROR (DWORD)0x00000700 </pre>
부가 설명	

11. ALT_ESTP_IsTcpConnect	
함수 원형	BOOL ALT_ESTP_IsTcpConnect (int Index) ;
동작 설명	Interface 객체번호를 사용해서 해당 객체의 통신이 TCPIP 모드 인지를 확인합니다. 통신 동작중에 TCPIP 통신의 끊김도 확인할 수 있습니다.
전달 인자	int Index : interface 객체 번호
리턴 값	BOOL : TRUE : TCPIP 통신이 연결되어 있습니다. FALSE : TCPIP 통신이 연결되어 있지 않습니다.
부가 설명	

12. ALT_ESTP_IsUdpConnect	
함수 원형	BOOL ALT_ESTP_IsUdpConnect (int Index) ;
동작 설명	Interface 객체번호를 사용해서 해당 객체의 통신이 UDP 모드 인지를 확인합니다. UDP 통신 모드에서는 통신 끊김을 확인할 수 없습니다.
전달 인자	int Index : interface 객체 번호
리턴 값	BOOL : TRUE : UDP 통신이 연결되어 있습니다. FALSE : UDP 통신이 연결되어 있지 않습니다.
부가 설명	

13. ALT_ESTP_IsUartConnect	
함수 원형	BOOL ALT_ESTP_IsUartConnect (int Index) ;
동작 설명	Interface 객체번호를 사용해서 해당 객체의 통신이 UART 모드 인지를 확인합니다. UART 통신 모드에서는 통신 끊김을 확인할 수 없습니다.
전달 인자	int Index : interface 객체 번호
리턴 값	BOOL : TRUE : UART 통신이 연결되어 있습니다. FALSE : UART 통신이 연결되어 있지 않습니다.
부가 설명	

14. ALT_ESTP_Protocol_Load	
함수 원형	BOOL ALT_ESTP_Protocol_Load (int Index, unsigned long *Time, char *DataMem, int *DataSize, BOOL *SendMode) ;
동작 설명	Interface 객체번호를 사용해서 장치와 주고 받는 Protocol 데이터를 읽습니다.
전달 인자	int Index : interface 객체 번호 unsigned long *Time : Protocol 데이터를 주고 받은 시간. mmsec 단위의 카운트. char *DataMem : Protocol 데이터를 저장할 Byte 단위 메모리 포인터. int *DataSize : 위의 DataMem 메모리의 Byte 크기를 전달하고, 포인터에 저장되는 실제 Byte 개수를 Return 받습니다. BOOL *SendMode : Protocol 데이터가 송신/수신을 판단하는 Flag. TRUE : 송신 Protocol 데이터 FALSE : 수신 Protocol 데이터
리턴 값	BOOL : TRUE : Protocol 데이터를 정상적으로 읽었습니다. FALSE : 정상적인 객체 번호가 아니거나 데이터를 읽을 메모리 포인터가 정상적이지 않습니다.
부가 설명	DataMem 의 메모리 Byte 크기를 DataSize 에 전달해서 만일 Protocol 데이터보다 작으면 DataSize 크기만큼만 Protocol 데이터를 읽습니다.

15. ALT_ESTP_Protocol_LoadW	
함수 원형	BOOL ALT_ESTP_Protocol_LoadW (int Index, unsigned long *Time, wchar_t *Data, int *BDataLen, BOOL *SendMode) ;
동작 설명	Interface 객체번호를 사용해서 장치와 주고 받는 Protocol 데이터를 읽습니다.
전달 인자	int Index : interface 객체 번호 unsigned long *Time : Protocol 데이터를 주고 받은 시간. mmsec 단위의 카운트. wchar_t *Data : Protocol 데이터를 저장할 유니코드 단위 메모리 포인터. int *BDataLen : 위의 유니코드 Data 메모리를 Byte 크기로 전달하고, 포인터에 저장되는 실제 Byte 개수를 Return 받습니다. BOOL *SendMode : Protocol 데이터가 송신/수신을 판단하는 Flag. TRUE : 송신 Protocol 데이터 FALSE : 수신 Protocol 데이터
리턴 값	BOOL : TRUE : Protocol 데이터를 정상적으로 읽었습니다. FALSE : 정상적인 객체 번호가 아니거나 데이터를 읽을 메모리 포인터가 정상적이지 않습니다.
부가 설명	유니코드 Data 의 메모리를 Byte 크기로 BDataLen 에 전달해서 만일 Protocol 데이터보다 작으면 BDataLen 크기만큼만 Protocol 데이터를 읽습니다.

16. ALT_ESTP_SendIPAddress	
함수 원형	BOOL ALT_ESTP_SendIPAddress (int Index, char *IP, char *NetMask, char *GateWay, int PortNum) ;
동작 설명	Interface 객체번호를 사용해서 Lan 통신에 필요한 Network 정보를 설정합니다. Network 정보가 수정되면 Controller 전원을 Off-On 합니다.
전달 인자	int Index : interface 객체 번호 char *IP : char IP[4] 형식의 데이터를 전달하며 “192.168.10.10”의 경우에 IP[0]=192;IP[1]=168;IP[2]=10;IP[3]=10; 으로 전달합니다. char *NetMask : “255.255.255.1”의 데이터를 위의 IP 와 동일하게 전달합니다. char *GateWay : “192.168.10.1”의 데이터를 위의 IP 와 동일하게 전달합니다. int PortNum : LAN 통신 포트 번호를 전달합니다.
리턴 값	BOOL : TRUE : 정상적으로 Network 정보를 Controller 에 설정하였습니다. FALSE : 정상적인 객체 번호가 아니거나 데이터 포인터가 잘못되었습니다.
부가 설명	

17. ALT_ESTP_ReadIPAddress	
함수 원형	BOOL ALT_ESTP_ReadIPAddress (int Index, char *IP, char *NetMask, char *GateWay, int *PortNum) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 설정된 Network 정보를 읽습니다.
전달 인자	int Index : interface 객체 번호 char *IP : char IP[4] 형식의 메모리 포인터를 전달하며 “192.168.10.10” 의 경우에 IP[0]=192;IP[1]=168;IP[2]=10;IP[3]=10;의 형식으로 읽습니다. char *NetMask : “255.255.255.1” 의 데이터를 위의 IP 데이터와 동일한 형식으로 읽습니다. char *GateWay : “192.168.10.1” 의 데이터를 위의 IP 데이터와 동일한 형식으로 읽습니다. int *PortNum : LAN 통신 포트 번호를 읽습니다.
리턴 값	BOOL : TRUE : 정상적으로 Network 정보를 Controller 에서 읽었습니다. FALSE : 정상적인 객체 번호가 아니거나 데이터 포인터가 잘못되었습니다.
부가 설명	

18. ALT_ESTP_SendMACAddress	
함수 원형	BOOL ALT_ESTP_SendMACAddress (int Index, char *MAC) ;
동작 설명	Interface 객체번호를 사용해서 Lan 통신에 필요한 Network MAC 정보를 설정합니다. MAC 정보가 수정되면 Controller 전원을 Off-On 합니다.
전달 인자	int Index : interface 객체 번호 char *MAC : char MAC[6] 형식의 데이터를 전달하며 “00:11:22:33:44:55” 의 경우에 MAC[0]=0x00;MAC[1]=0x11;MAC[2]=0x22; MAC[3]=0x33;MAC[4]=0x44;MAC[5]=0x55; 으로 전달합니다.
리턴 값	BOOL : TRUE : 정상적으로 Network MAC 정보를 Controller 에 설정하였습니다. FALSE : 정상적인 객체 번호가 아니거나 데이터 포인터가 잘못되었습니다.
부가 설명	

19. ALT_ESTP_ReadMACAddress	
함수 원형	BOOL ALT_ESTP_ReadMACAddress (int Index, char *MAC);
동작 설명	Interface 객체번호를 사용해서 Controller 에 설정된 Network MAC 정보를 읽습니다.
전달 인자	int Index : interface 객체 번호 char *MAC : char MAC[6] 형식의 메모리 포인터를 전달하며 “00:11:22:33:44:55” 의 경우에 MAC[0]=0x00;MAC[1]=0x11;MAC[2]=0x22; MAC[3]=0x33;MAC[4]=0x44;MAC[5]=0x55; 으로 읽습니다.
리턴 값	BOOL : TRUE : 정상적으로 Network MAC 정보를 Controller 에서 읽었습니다. FALSE : 정상적인 객체 번호가 아니거나 데이터 포인터가 잘못되었습니다.
부가 설명	

20. ALT_ESTP_SendMacroData	
함수 원형	BOOL ALT_ESTP_SendMacroData (int Index, unsigned char MacroIndex, unsigned char MacroSize, unsigned long *MacroData) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 MACRO 프로그램을 다운로드 합니다.
전달 인자	Controller 동작에 영향을 미칠 수 있으므로 세부적인 내용은 제조사로 연락바랍니다.
리턴 값	BOOL : TRUE : 정상적으로 MACRO 프로그램이 다운로드 되었습니다. FALSE : 정상적인 객체 번호가 아니거나 전달인자가 잘못되었습니다.
부가 설명	

21. ALT_ESTP_ReadMacroData	
함수 원형	BOOL ALT_ESTP_ReadMacroData (int Index, unsigned char MacroIndex, unsigned char *MacroSize, unsigned long *MacroData) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 저장되어 있는 MACRO 프로그램을 업로드 합니다.
전달 인자	Controller 동작에 영향을 미칠 수 있으므로 세부적인 내용은 제조사로 연락바랍니다.
리턴 값	BOOL : TRUE : 정상적으로 MACRO 프로그램이 업로드 되었습니다. FALSE : 정상적인 객체 번호가 아니거나 전달인자가 잘못되었습니다.
부가 설명	

22. ALT_ESTP_StrobeValueSend	
함수 원형	BOOL ALT_ESTP_StrobeValueSend (int Index, char PageIndex, char ChannelIndex, unsigned short StrobeValue) ;
동작 설명	Interface 객체번호를 사용해서 Controller 의 원하는 Page 에 원하는 Channel 에 Strobe OnTime 값을 설정합니다.
전달 인자	int Index : interface 객체 번호 char PageIndex : Page Index. 0 ~ 255. char ChannelIndex : Strobe Channel Index. 0 ~ 3,7,15,31,63. unsigned short StrobeValue : Strobe OnTime Value. 0 ~ 4000
리턴 값	BOOL : TRUE : 정상적으로 Strobe OnTime 이 설정되었습니다. FALSE : 정상적인 객체 번호가 아니거나 전달인자가 잘못되었습니다.
부가 설명	Strobe OnTime 값은 기본 동작에서는 usec 단위입니다. 하지만 MACRO 동작에서는 시간 분해능 설정에 따라서 다양한 값으로 의미가 다를 수 있습니다.

23. ALT_ESTP_AllStrobeValueSend	
함수 원형	BOOL ALT_ESTP_AllStrobeValueSend (int Index, char PageIndex, unsigned short *StrobeValue) ;
동작 설명	Interface 객체번호를 사용해서 Controller 의 원하는 Page 에 모든 Channel 에 Strobe OnTime 값을 설정합니다.
전달 인자	int Index : interface 객체 번호 char PageIndex : Page Index. 0 ~ 255. unsigned short *StrobeValue : Strobe OnTime Value 의 배열 메모리 포인터. 배열의 수는 Channel 의 개수와 같아야 합니다. 0 ~ 4000
리턴 값	BOOL : TRUE : 정상적으로 Strobe OnTime 이 설정되었습니다. FALSE : 정상적인 객체 번호가 아니거나 전달인자가 잘못되었습니다.
부가 설명	Strobe OnTime 값은 기본 동작에서는 usec 단위입니다. 하지만 MACRO 동작에서는 시간 분해능 설정에 따라서 다양한 값으로 의미가 다를 수 있습니다.

24. ALT_ESTP_AllStrobeValueRead	
함수 원형	BOOL ALT_ESTP_AllStrobeValueRead (int Index, char PageIndex, unsigned short *StrobeValue) ;
동작 설명	Interface 객체번호를 사용해서 Controller 의 원하는 Page 의 모든 Channel 의 Strobe OnTime 값을 읽습니다.
전달 인자	int Index : interface 객체 번호 char PageIndex : Page Index. 0 ~ 255. unsigned short *StrobeValue : Strobe OnTime Value 을 저장할 배열 메모리 포인터. 배열의 수는 Channel 의 개수와 같아야 합니다.
리턴 값	BOOL : TRUE : 정상적으로 Strobe OnTime 이 읽어졌습니다. FALSE : 정상적인 객체 번호가 아니거나 전달인자가 잘못되었습니다.
부가 설명	Strobe OnTime 값은 기본 동작에서는 usec 단위입니다. 하지만 MACRO 동작에서는 시간 분해능 설정에 따라서 다양한 값으로 의미가 다를 수 있습니다.

25. ALT_ESTP_TriggerModeSend	
함수 원형	BOOL ALT_ESTP_TriggerModeSend (int Index, BOOL TriggerMode) ;
동작 설명	Interface 객체번호를 사용해서 Controller 의 동작을 일으키는 Trigger Edge 의 방향을 선택합니다.
전달 인자	int Index : interface 객체 번호 BOOL TriggerMode : Trigger Edge 의 방향을 선택합니다. TRUE : Rising-Edge FALSE : Falling-Edge
리턴 값	BOOL : TRUE : 정상적으로 Trigger Edge Mode 가 설정되었습니다. FALSE : 정상적인 객체 번호가 아닙니다.
부가 설명	

26. ALT_ESTP_EncoderModeSend	
함수 원형	BOOL ALT_ESTP_EncoderModeSend (int Index, char EncoderMode) ;
동작 설명	Interface 객체번호를 사용해서 Controller 의 동작을 일으키는 Encoder Count 의 입력 방식을 선택합니다.
전달 인자	int Index : interface 객체 번호 char EncoderMode : Encoder Count 의 입력 방법을 선택합니다. 0 : Pos. Direction/1-Multi. 1 : Pos. Direction/2-Multi. 2 : Pos. Direction/4-Multi. 3 : Pos./Neg. Direction/1-Multi. 4 : Pos./Neg. Direction/2-Multi. 5 : Pos./Neg. Direction/4-Multi.
리턴 값	BOOL : TRUE : 정상적으로 Encoder Mode 가 설정되었습니다. FALSE : 정상적인 객체 번호가 아니거나 전달인자가 잘못되었습니다.
부가 설명	Encoder 관련 동작은 MACRO 프로그램을 통해서만 사용할 수 있습니다.

27. ALT_ESTP_SaveSend	
함수 원형	BOOL ALT_ERS_SaveSend (int Index) ;
동작 설명	Interface 객체번호를 사용해서 Controller 의 환경 설정을 저장합니다.
전달 인자	int Index : interface 객체 번호
리턴 값	BOOL : TRUE : 정상적으로 환경 설정 데이터가 저장되었습니다. FALSE : 정상적인 객체 번호가 아닙니다.
부가 설명	환경 설정 데이터는 Network 정보와 MACRO Index, TriggerMode, EncoderMode 입니다. MACRO Index 가 저장되면 다음 전원 On 시에 해당 MACRO 가 자동 실행됩니다.

28. ALT_ESTP_LoadSend	
함수 원형	BOOL ALT_ESTP_LoadSend (int Index) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 저장된 환경 설정을 적용시킵니다.
전달 인자	int Index : interface 객체 번호
리턴 값	BOOL : TRUE : 정상적으로 저장된 환경 설정 데이터를 Controller 에 적용했습니다. FALSE : 정상적인 객체 번호가 아닙니다.
부가 설명	환경 설정 데이터는 Network 정보와 MACRO Index, TriggerMode, EncoderMode 입니다. 저장되어 있던 환경 설정 데이터가 Controller 에 적용되면 현재 사용중이던 환경 설정 데이터는 지워집니다.

29. ALT_ESTP_EncClearSend	
함수 원형	BOOL ALT_ESTP_EncClearSend (int Index) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 입력되는 Encoder Count 를 0 으로 초기화 합니다.
전달 인자	int Index : interface 객체 번호
리턴 값	BOOL : TRUE : 정상적으로 Encoder Count 를 초기화 했습니다. FALSE : 정상적인 객체 번호가 아닙니다.
부가 설명	

30. ALT_ESTP_EncCountRead	
함수 원형	BOOL ALT_ESTP_EncCountRead (int Index, unsigned long *EncCount) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 입력되는 Encoder Count 를 읽습니다.
전달 인자	int Index : interface 객체 번호 unsigned long *EncCount : Encoder Count 개수를 저장하는 메모리 포인터 입니다. EncoderMode 선택에 따라서 개수가 틀려질 수 있습니다.
리턴 값	BOOL : TRUE : 정상적으로 Encoder Count 를 읽었습니다. FALSE : 정상적인 객체 번호가 아니거나 전달 인자가 잘못되었습니다.
부가 설명	

31. ALT_ESTP_RunCodeSend	
함수 원형	BOOL ALT_ERS_RunCodeSend (int Index, char RunCode) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 저장된 MACRO 프로그램을 실행시킵니다.
전달 인자	int Index : interface 객체 번호 char RunCode : 실행을 원하는 MACRO Index. 1 ~ 10 RunCode 가 0이면 MACRO 실행을 정지하고 기본 동작을 수행합니다.
리턴 값	BOOL : TRUE : 정상적으로 새로운 MACRO RunCode 를 적용합니다. FALSE : 정상적인 객체 번호가 아니거나 RunCode 가 아닙니다.
부가 설명	RunCode 는 자동 저장되지 않습니다. ALT_ESTP_SaveSend 명령을 사용해서 환경 설정 값을 저장해야 합니다.

32. ALT_ESTP_PageIndexSend	
함수 원형	BOOL ALT_ESTP_PageIndexSend (int Index, char PageIndex) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 기본 작업 PageIndex 를 설정합니다.
전달 인자	int Index : interface 객체 번호 char PageIndex : 현재 기본 작업을 원하는 Page Index. 0 ~ 255.
리턴 값	BOOL : TRUE : 정상적으로 새로운 Page Index 가 설정되었습니다. FALSE : 정상적인 객체 번호가 아닙니다.
부가 설명	Page Index 는 저장되지 않습니다.

33. ALT_ESTP_PageIndexRead	
함수 원형	BOOL ALT_ESTP_PageIndexRead (int Index, char *PageIndex) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 기본 작업 PageIndex 를 읽습니다.
전달 인자	int Index : interface 객체 번호 char *PageIndex : 현재 기본 작업 Page Index 를 저장할 메모리 포인터.
리턴 값	BOOL : TRUE : 정상적으로 기본 Page Index 를 읽었습니다. FALSE : 정상적인 객체 번호가 아니거나 메모리 포인터가 정상이지 않습니다.
부가 설명	

34. ALT_ESTP_SoftTriggerSend	
함수 원형	BOOL ALT_ESTP_SoftTriggerSend (int Index, unsigned long TriggerHigh, unsigned long TriggerLow) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 Software 통해서 Trigger 를 전달합니다.
전달 인자	int Index : interface 객체 번호 unsigned long TriggerHigh : Trigger 개수의 상위반을 담당합니다. unsigned long TriggerLow : Trigger 개수의 하위반을 담당합니다.
리턴 값	BOOL : TRUE : 정상적으로 Soft-Trigger 가 전달되었습니다. FALSE : 정상적인 객체 번호가 아닙니다.
부가 설명	Trigger 개수의 High/Low 는 다음과 같이 입력됩니다. 4-Channel Trigger 의 경우) 0 번 : TriggerHigh:0x00000000, TriggerLow:0x00000001 1 번 : TriggerHigh:0x00000000, TriggerLow:0x00000002 2 번 : TriggerHigh:0x00000001, TriggerLow:0x00000000 3 번 : TriggerHigh:0x00000002, TriggerLow:0x00000000 16-Channel Trigger 의 경우) 0 번 : TriggerHigh:0x00000000, TriggerLow:0x00000001 1 번 : TriggerHigh:0x00000000, TriggerLow:0x00000002 2 번 : TriggerHigh:0x00000000, TriggerLow:0x00000004 3 번 : TriggerHigh:0x00000000, TriggerLow:0x00000008 4 번 : TriggerHigh:0x00000000, TriggerLow:0x00000010 5 번 : TriggerHigh:0x00000000, TriggerLow:0x00000020 6 번 : TriggerHigh:0x00000000, TriggerLow:0x00000040 7 번 : TriggerHigh:0x00000000, TriggerLow:0x00000080 8 번 : TriggerHigh:0x00000001, TriggerLow:0x00000000 9 번 : TriggerHigh:0x00000002, TriggerLow:0x00000000 10 번 : TriggerHigh:0x00000004, TriggerLow:0x00000000 11 번 : TriggerHigh:0x00000008, TriggerLow:0x00000000 12 번 : TriggerHigh:0x00000010, TriggerLow:0x00000000 13 번 : TriggerHigh:0x00000020, TriggerLow:0x00000000 14 번 : TriggerHigh:0x00000040, TriggerLow:0x00000000 15 번 : TriggerHigh:0x00000080, TriggerLow:0x00000000

35. ALT_ESTP_ConfigModeRead	
함수 원형	BOOL ALT_ESTP_ConfigModeRead (int Index, char *ChannelNum, BOOL *TriggerMode, char *EncoderMode) ;
동작 설명	Interface 객체번호를 사용해서 Controller 의 Channel 개수, TriggerMode, EncoderMode 를 읽습니다.
전달 인자	int Index : interface 객체 번호 char *ChannelNum : Controller 에 적용되는 Channel 개수를 저장하는 메모리 포인터. BOOL *TriggerMode : Controller 에 적용되어 있는 TriggerMode 를 저장하는 메모리 포인터. BOOL *EncoderMode : Controller 에 적용되어 있는 EncoderMode 를 저장하는 메모리 포인터.
리턴 값	BOOL : TRUE : 정상적으로 Controller 의 동작 모드를 읽었습니다. FALSE : 정상적인 객체 번호가 아니거나 메모리 포인터가 정상이지 않습니다.
부가 설명	

36. ALT_ESTP_EncOffsetListSend	
함수 원형	BOOL ALT_ESTP_EncOffsetListSend (int Index, unsigned short EncCount, unsigned short *EncOffsetList) ;
동작 설명	Interface 객체번호를 사용해서 Encoder 변위 값의 List 데이터를 전달합니다. 변위 값은 16bit 로 구성되며 256개 까지 전달할 수 있습니다.
전달 인자	int Index : interface 객체 번호 unsigned short EncCount : 장치로 전달하려는 Encoder 변위 값의 개수. unsigned short *EncOffsetList : Controller 에 전달되는 Encoder 변위 데이터의 배열 메모리 포인터. 위의 변위 개수와 실제 데이터의 개수가 일치해야 합니다.
리턴 값	BOOL : TRUE : 정상적으로 데이터를 전달하였습니다. FALSE : 정상적인 객체 번호가 아니거나 메모리 포인터가 정상 아닙니다.
부가 설명	

37. ALT_ESTP_EncOffsetListRead	
함수 원형	BOOL ALT_ESTP_EncOffsetListRead (int Index, unsigned short *EncCount, unsigned short *EncOffsetList) ;
동작 설명	Interface 객체번호를 사용해서 Encoder 변위 값의 개수와 데이터 List 를 읽습니다. 변위 값은 16bit 로 구성되며 256개 까지 전달할 수 있습니다.
전달 인자	int Index : interface 객체 번호 unsigned short *EncCount : 장치에 저장되어 있는 Encoder 변위 값의 개수를 받기 위한 포인터. unsigned short *EncOffsetList : Controller 에 저장되어 있는 Encoder 변위 데이터 List 를 읽기위한 메모리 포인터. 최대 256개의 저장 공간 이 필요합니다.
리턴 값	BOOL : TRUE : 정상적으로 데이터를 읽었습니다. FALSE : 정상적인 객체 번호가 아니거나 메모리 포인터가 정상이 아닙니다.
부가 설명	