

ALT-TDIVIDER DLL Manual

Version : 2019-10-02

Version : 2020-03-31

Version : 2020-09-29

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

목 차

1. ALT_TDIV_Create
2. ALT_TDIV_Close
3. ALT_TDIV_Lan_Start
4. ALT_TDIV_Lan_WStart
5. ALT_TDIV_Uart_Check
6. ALT_TDIV_Uart_WCheck
7. ALT_TDIV_Uart_Start
8. ALT_TDIV_Uart_WStart
9. ALT_TDIV_Setup
10. ALT_TDIV_GetLastErrorCode
11. ALT_TDIV_IsTcpConnect
12. ALT_TDIV_IsUdpConnect
13. ALT_TDIV_IsUartConnect
14. ALT_TDIV_Protocol_Load
15. ALT_TDIV_Protocol_LoadW
16. ALT_TDIV_SendIPAddress
17. ALT_TDIV_ReadIPAddress
18. ALT_TDIV_SendMACAddress
19. ALT_TDIV_ReadMACAddress
20. ALT_TDIV_VersionInfoRead
21. ALT_TDIV_ConfigEnableSend
22. ALT_TDIV_ConfigEnableRead
23. ALT_TDIV_ConfigDataSend
24. ALT_TDIV_ConfigDataRead
25. ALT_TDIV_EncOutDataSend
26. ALT_TDIV_EncOutDataRead
27. ALT_TDIV_PhotoOutDataSend
28. ALT_TDIV_PhotoOutDataRead
29. ALT_TDIV_DiffOutDataSend
30. ALT_TDIV_DiffOutDataRead
31. ALT_TDIV_SaveSend
32. ALT_TDIV_LoadSend
33. ALT_TDIV_EncClearSend
34. ALT_TDIV_EncCountRead
35. ALT_TDIV_SoftTriggerSend

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

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

3. ALT_TDIV_Lan_Start	
함수 원형	BOOL ALT_TDIV_Lan_Start (int Index, BOOL TCPIP, char *lpServer, int port);
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
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_TDIV_Close 함수를 호출하면 OPEN 된 통신 포트도 CLOSE 하고 Interface 객체도 제거합니다.

4. ALT_TDIV_Lan_WStart	
함수 원형	BOOL ALT_TDIV_Lan_WStart (int Index, BOOL TCPIP, wchar_t *lpServer, int port) ;
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
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_TDIV_Close 함수를 호출하면 OPEN 된 통신 포트도 CLOSE 하 고 Interface 객체도 제거합니다.

5. ALT_TDIV_Uart_Check	
함수 원형	BOOL ALT_TDIV_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 사용중임을 나타냅니다.
부가 설명	사용가능한 Uart Port 리스트를 만들 때 사용할 수 있습니다. ALT_TDIV_Close 함수를 호출하면 OPEN 된 통신 포트도 CLOSE 하고 Interface 객체도 제거합니다.

6. ALT_TDIV_Uart_WCheck	
함수 원형	BOOL ALT_TDIV_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_TDIV_Close 함수를 호출하면 OPEN 된 통신 포트도 CLOSE 하고 Interface 객체도 제거합니다.

7. ALT_TDIV_Uart_Start	
함수 원형	BOOL ALT_TDIV_Uart_Start (int Index, char *PortName, int Borate) ;
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.
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_TDIV_Close 함수를 호출하면 OPEN 된 통신 포트도 CLOSE 하고 Interface 객체도 제거합니다.

8. ALT_TDIV_Uart_WStart	
함수 원형	BOOL ALT_TDIV_Uart_WStart (int Index, wchar_t *PortName, int Borate);
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.
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_TDIV_Close 함수를 호출하면 OPEN 된 통신 포트도 CLOSE 하고 Interface 객체도 제거합니다.

9. ALT_TDIV_Setup	
함수 원형	BOOL ALT_TDIV_Setup (int Index, char EncInNum, char PhotoInNum, char DiffInNum, char EncOutNum, char PhotoOutNum, char DiffOutNum) ;
동작 설명	Interface 객체번호를 사용해서 각 Input/Output Channel 의 수를 설정합니다.
전달 인자	int Index : interface 객체 번호 char EncInNum : Encoder 입력 채널의 수. Default=1 char PhotoInNum : Photo 입력 채널의 수. Default=8 char DiffInNum : Differential 입력 채널의 수. Default=8 char EncOutNum : Encoder 출력 채널의 수. Default=4 char PhotoOutNum : Photo 출력 채널의 수. Default=8 char DiffOutNum : Differential 출력 채널의 수. Default=16
리턴 값	BOOL : TRUE : 채널수가 변경되었습니다. FALSE : 잘못된 객체번호 혹은 채널 수가 전달되어 변경이 실패하였습니다.
부가 설명	

10. ALT_TDIV_GetLastErrorCode	
함수 원형	unsigned long ALT_TDIV_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 TDIV_PARAMETER_VALUE_ERROR (DWORD)0x00000100 #define TDIV_MEM_ALLOCATION_ERROR (DWORD)0x00000200 #define TDIV_SYNC_OBJECT_CREATE_ERROR (DWORD)0x00000300 #define TDIV_RECEIVE_THREAD_CREATE_ERROR (DWORD)0x00000400 #define TDIV_RECEIVE_PROTOCOL_TIMEOUT (DWORD)0x00000500 #define TDIV_RECEIVE_NOT_EXIST_COMMAND_ERROR (DWORD)0x00000600 #define TDIV_PARAMETER_EXCEPTION_ERROR (DWORD)0x00000700 </pre>
부가 설명	

11. ALT_TDIV_IsTcpConnect	
함수 원형	BOOL ALT_TDIV_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_TDIV_IsUdpConnect	
함수 원형	BOOL ALT_TDIV_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_TDIV_IsUartConnect	
함수 원형	BOOL ALT_TDIV_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_TDIV_Protocol_Load	
함수 원형	BOOL ALT_TDIV_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_TDIV_Protocol_LoadW	
함수 원형	BOOL ALT_TDIV_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_TDIV_SendIPAddress	
함수 원형	BOOL ALT_TDIV_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_TDIV_ReadIPAddress	
함수 원형	BOOL ALT_TDIV_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_TDIV_SendMACAddress	
함수 원형	BOOL ALT_TDIV_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_TDIV_ReadMACAddress	
함수 원형	BOOL ALT_TDIV_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_TDIV_VersionInfoRead	
함수 원형	BOOL ALT_TDIV_VersionInfoRead (int Index, unsigned long *FVersion, unsigned long *LVersion) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 설정된 Version 정보를 읽습니다.
전달 인자	int Index : interface 객체 번호 unsigned long *FVersion : HEX 값으로 이루어진 Firmware 버전정보 unsigned long *LVersion : HEX 값으로 이루어진 LOGIC 버전정보
리턴 값	BOOL : TRUE : 정상적으로 Version 정보를 Controller 에서 읽었습니다. FALSE : 정상적인 객체 번호가 아니거나 데이터 포인터가 잘못되었습니다.
부가 설명	

21. ALT_TDIV_ConfigEnableSend	
함수 원형	BOOL ALT_TDIV_ConfigEnableSend (int Index, BOOL Reserved, BOOL EncDivEnable, BOOL TimerEnable, BOOL *EncEnable, BOOL *PhotoEnable, BOOL *DiffEnable) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 각 기능 활성화를 설정합니다.
전달 인자	int Index : interface 객체 번호 BOOL Reserved : Reserved. Must be FALSE. BOOL EncDivEnable : Encoder 출력에 분주 기능을 사용할 지를 선택합니다. TRUE : 입력된 분주비에 의한 분주 기능을 선택합니다. FALSE : Encoder 입력대비 출력을 1:1 로 출력시킵니다. BOOL TimerEnable : Encoder 를 사용하지 않고 내부 Timer 에 의한 Simulation 동작을 선택합니다. TRUE : 10nsec 단위의 내부 Timer 를 동작시킵니다. FALSE : 내부 Timer 를 사용하지 않고 Encoder 를 동기신호로 사용합니다. BOOL *EncEnable : Encoder 출력을 활성화 시킵니다. BOOL [4] 의 데이터 형식의 포인터를 전달합니다. TRUE : 해당 채널의 Encoder 를 출력 시킵니다. FALSE : 해당 채널의 Encoder 를 출력하지 않습니다. BOOL *PhotoEnable : Photo 출력을 활성화 시킵니다. BOOL [8] 의 데이터 형식의 포인터를 전달합니다. TRUE : 해당 채널의 Output 을 출력 시킵니다. FALSE : 해당 채널의 Output 을 출력하지 않습니다. BOOL *DiffEnable : Diff 출력을 활성화 시킵니다. BOOL [16] 의 데이터 형식의 포인터를 전달합니다. TRUE : 해당 채널의 Output 을 출력 시킵니다. FALSE : 해당 채널의 Output 을 출력하지 않습니다.
리턴 값	BOOL : TRUE : 정상적으로 해당 기능의 활성/비활성을 설정했습니다. FALSE : 정상적인 객체 번호가 아니거나 전달 인자가 잘못되었습니다.
부가 설명	각 Output 동작에 대한 상태를 초기화 하려면 해당 출력을 FALSE -> TRUE 시키면 됩니다.

22. ALT_TDIV_ConfigEnableRead	
함수 원형	BOOL ALT_TDIV_ConfigEnableRead (int Index, BOOL *Reserved, BOOL *EncDivEnable, BOOL *TimerEnable, BOOL *EncEnable, BOOL *PhotoEnable, BOOL *DiffEnable) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 각 기능의 활성화 상태를 읽습니다.
전달 인자	int Index : interface 객체 번호 BOOL *Reserved : Reserved. 데이터 포인터 전달. BOOL *EncDivEnable : Encoder 출력에 분주 기능 상태. BOOL *TimerEnable : 내부 Timer 동작 상태. BOOL *EncEnable : Encoder 출력 상태. BOOL [4] 의 데이터 포인터 전달합니다. BOOL *PhotoEnable : Photo 출력 상태. BOOL [8] 의 데이터 포인터 전달합니다. BOOL *DiffEnable : Diff 출력 상태. BOOL [16] 의 데이터 포인터 전달합니다.
리턴 값	BOOL : TRUE : 정상적으로 해당 기능의 활성화/비활성 상태를 읽었습니다. FALSE : 정상적인 객체 번호가 아니거나 전달 인자가 잘못되었습니다.
부가 설명	

23. ALT_TDIV_ConfigDataSend	
함수 원형	BOOL ALT_TDIV_ConfigDataSend (int Index, char EncSelect, char EncMulti, BOOL PEGPole, BOOL PInputPole, BOOL DInputPole, unsigned long TimeDiv, unsigned short EncOutLength, unsigned char EncDivRate, unsigned char EncInDivRate, unsigned char EncZPhaseSyncSelect, unsigned char EncZPhaseTriggerNumber) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 각 기능 동작을 위한 환경 데이터를 설정합니다.
전달 인자	<p>int Index : interface 객체 번호</p> <p>char EncSelect : Controller 동작을 위한 동기 입력 신호를 선택합니다. 0 : Encoder 신호 입력 선택. 1 : PEG 위치 결정 신호 입력 선택. 2 : Timer 내부 동기 신호 입력 선택.</p> <p>char EncMulti : 동기 신호로 Encoder 가 선택되었을 때 적용됩니다. 1,2,4 채배를 선택할 수 있습니다.</p> <p>BOOL PEGPole : PEG 위치 결정신호의 입력 극성을 선택합니다. TRUE : Rising-Edge FALSE : Falling-Edge</p> <p>BOOL PInputPole : Photo 입력 신호의 입력 극성을 선택합니다. TRUE : Rising-Edge FALSE : Falling-Edge</p> <p>BOOL DInputPole : Diff 입력 신호의 입력 극성을 선택합니다. TRUE : Rising-Edge FALSE : Falling-Edge</p> <p>unsigned long TimeDiv : 동기 신호로 Timer 가 선택되었을 때 적용됩니다. 10nsec 단위의 분해능을 갖는 Count 값을 입력합니다. 24비트 값입니다.</p> <p>unsigned short EncOutLength : Encoder 출력을 위한 Pulse Width 를 입력합니다. 10nsec 의 분해능을 갖는 Count 값을 입력합니다. EncMulti 값이 1,2 일 경우에만 적용됩니다.</p> <p>unsigned char EncDivRate : Encoder 출력을 위한 분주비를 입력합니다. Encoder 분주비가 적용되었을 때 이 값이 0이면 Encoder 는 출력되지 않습니다. 1 ~ 255 의 값 범위를 갖습니다.</p> <p>unsigned char EncInDivRate : Encoder 입력을 위한 Time-Filter 주파수를 분주하는 시정수 입니다. Clock Source 는 100MHz 를 사용하고 1값은 100MHz 를 직접사용. 2이상의 값은 짝수만 유효합니다.</p>

	<p>다. 10의 값은 주파수를 10MHz 로 만듭니다. Time-Filter 값은 주파수*5 의 값을 가집니다. 1 ~ 255 의 값을 사용할 수 있습니다.</p> <p>EncZPhaseSyncSelect : Encoder 입력 동작에서 Z 상 을 사용할 때 실제 Encoder 의 Z 상을 사용할 수도 있고, Diff/Photo Input Port 를 Z 상 입력으로 사용할 수도 있습니다.</p> <p>EncZPhaseTriggerNumber : Encoder Z 상을 대체할 Diff/Photo Input 의 포트 번호</p>
리턴 값	<p>BOOL :</p> <p>TRUE : 정상적으로 환경 데이터를 설정했습니다.</p> <p>FALSE : 정상적인 객체 번호가 아니거나 전달 인자가 잘못되었습니다.</p>
부가 설명	

24. ALT_TDIV_ConfigDataRead	
함수 원형	BOOL ALT_TDIV_ConfigDataRead (int Index, char *EncSelect, char *EncMulti, BOOL *PEGPole, BOOL *PInputPole, BOOL *DInputPole, unsigned long *TimeDiv, unsigned short *EncOutLength, unsigned char *EncDivRate, unsigned char *EncInDivRate, unsigned char *EncZPhaseSyncSelect, unsigned char *EncZPhaseTriggerNumber) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 적용된 환경 데이터를 읽습니다.
전달 인자	int Index : interface 객체 번호 char *EncSelect : char 데이터 메모리 포인터. char *EncMulti : char 데이터 메모리 포인터. BOOL *PEGPole : BOOL 데이터 메모리 포인터. BOOL *PInputPole : BOOL 데이터 메모리 포인터. BOOL *DInputPole : BOOL 데이터 메모리 포인터. unsigned long *TimeDiv : unsigned long 데이터 메모리 포인터. 24비트 값입니다. unsigned short *EncOutLength : unsigned short 데이터 메모리 포인터. unsigned char *EncDivRate : unsigned char 데이터 메모리 포인터 unsigned char *EncInDivRate : unsigned char 데이터 메모리 포인터 unsigned char * EncZPhaseSyncSelect : unsigned char 데이터 메모리 포인터 unsigned char * EncZPhaseTriggerNumber : unsigned char 데이터 메모리 포인터
리턴 값	BOOL : TRUE : 정상적으로 환경 데이터를 읽었습니다. FALSE : 정상적인 객체 번호가 아니거나 전달 인자가 잘못되었습니다.
부가 설명	

25. ALT_TDIV_EncOutDataSend	
함수 원형	BOOL ALT_TDIV_EncOutDataSend (int Index, char EncIndex, char EncOutSource, BOOL EncWindowMode, char EncOutType, char EncOutReverse, char EncOutTriggerNum, unsigned long EncWindowHigh, unsigned long EncWindowLow) ;
동작 설명	Interface 객체번호를 사용해서 Controller 의 Encoder 출력을 위한 조건 데이터를 설정합니다.
전달 인자	<p>int Index : interface 객체 번호</p> <p>char EncIndex : Encoder Out Index 를 선택합니다. 0 ~ 3.</p> <p>char EncOutSource : Encoder Out 동기 신호를 선택합니다.</p> <p>0 : Encoder 신호 선택. Encoder 범위 출력인 WindowMode 를 적용할 수 있습니다.</p> <p>1 : Diff 신호 선택. 해당 신호를 만나면 Encoder 신호가 출력됩니다.</p> <p>2 : Photo 신호 선택. 해당 신호를 만나면 Encoder 신호가 출력됩니다.</p> <p>BOOL EncWindowMode : Encoder 신호의 범위 동작을 선택합니다.</p> <p>TRUE : Encoder Out 범위 동작을 활성화 합니다.</p> <p>FALSE : Encoder Out 범위 동작을 비활성화 합니다.</p> <p>char EncOutType : Encoder Out 신호 형태를 선택합니다.</p> <p>0 : A/B Pulse 출력을 선택합니다.</p> <p>1 : Pulse/Dir 형식의 출력을 선택합니다.</p> <p>2 : CW/CCW 형식의 출력을 선택합니다.</p> <p>char EncOutReverse : Encoder Out 방향을 선택합니다.</p> <p>0 : Only Positive Direction 을 선택합니다.</p> <p>1 : Only Negative Direction 을 선택합니다.</p> <p>2 : Pos./Neg. Direction 을 선택합니다.</p> <p>char EncOutTriggerNum : Encoder Out 동기 신호를 Diff/Photo 를 선택했을 때 적용됩니다. IO Input Index 를 선택합니다. 0 ~ 7.</p> <p>unsigned long EncWindowHigh : Encoder Out 범위 동작 선택시에 적용됩니다. 입력 Encoder 값의 상위 제한값을 입력합니다. 24비트 값입니다.</p> <p>unsigned long EncWindowLow : Encoder Out 범위 동작 선택시에 적용됩니다. 입력 Encoder 값의 하위 제한값을 입력합니다. 24비트 값입니다.</p>
리턴 값	<p>BOOL :</p> <p>TRUE : 정상적으로 Encoder Out 조건 데이터를 설정했습니다.</p> <p>FALSE : 정상적인 객체 번호가 아니거나 전달 인자가 잘못되었습니다.</p>

부가 설명	
-------	--

26. ALT_TDIV_EncOutDataRead	
함수 원형	BOOL ALT_TDIV_EncOutDataRead (int Index, char EncIndex, char *EncOutSource, BOOL *EncWindowMode, char *EncOutType, char *EncOutReverse, char *EncOutTriggerNum, unsigned long *EncWindowHigh, unsigned long *EncWindowLow) ;
동작 설명	Interface 객체번호를 사용해서 Controller 의 Encoder 출력을 위한 조건 데이터를 읽습니다.
전달 인자	int Index : interface 객체 번호 char EncIndex : Encoder Out Index 를 선택합니다. 0 ~ 3. char *EncOutSource : char 형식의 데이터 포인터. BOOL *EncWindowMode : BOOL 형식의 데이터 포인터. char *EncOutType : char 형식의 데이터 포인터. char *EncOutReverse : char 형식의 데이터 포인터. char *EncOutTriggerNum : char 형식의 데이터 포인터. unsigned long *EncWindowHigh : unsigned long 형식의 데이터 포인터. 24비트 값입니다. unsigned long *EncWindowLow : unsigned long 형식의 데이터 포인터. 24비트 값입니다.
리턴 값	BOOL : TRUE : 정상적으로 Encoder Out 조건 데이터를 설정했습니다. FALSE : 정상적인 객체 번호가 아니거나 전달 인자가 잘못되었습니다.
부가 설명	

27. ALT_TDIV_PhotoOutDataSend	
함수 원형	BOOL ALT_TDIV_PhotoOutDataSend (int Index, char PhotoIndex, char PhotoOutSource, BOOL PhotoOutRepeat, BOOL PhotoOutPole, char PhotoOutReverse, unsigned long PhotoOutOffset, char PhotoOutTriggerNum, unsigned long PhotoOutDistance, unsigned short DelayTime, unsigned short ActiveTime, unsigned long PhotoOutCount, unsigned char PhotoZCheck) ;
동작 설명	Interface 객체번호를 사용해서 Controller 의 Photo Out 출력을 위한 조건 데이터를 설정합니다.
전달 인자	<p>int Index : interface 객체 번호</p> <p>char PhotoIndex : Photo Output Index. 0 ~ 7.</p> <p>char PhotoOutSource : Photo Output 동기 신호를 선택합니다. 0 : Encoder 신호 값에 의한 출력 선택. 1 : Diff Input 신호를 동기로 출력 선택. 2 : Photo Input 신호를 동기호 출력 선택.</p> <p>BOOL PhotoOutRepeat : Photo Output 동기 조건이 맞았을 때 입력된 OutCount 만큼 연속해서 출력되는 모드 선택합니다. TRUE : 연속 출력 모드 선택. FALSE : 출력 조건에 대해 1회 출력 모드 선택.</p> <p>BOOL PhotoOutPole : Photo Output 출력의 극성을 선택합니다. TRUE : Rising-Edge FALSE : Falling-Edge</p> <p>char PhotoOutReverse : Encoder 입력을 동기 신호로 선택했을 때 적용됩니다. 0 : Positive Direction 출력 선택. 1 : Negative Direction 출력 선택. 2 : Pos./Neg. Direction 출력 선택.</p> <p>unsigned long PhotoOutOffset : Encoder 입력을 동기 신호로 선택했을 때 적용됩니다. Encoder 신호의 시작점에서 동작점까지의 Encoder 개수를 입력합니다. 24비트 값입니다.</p> <p>char PhotoOutTriggerNum : Photo Output 출력을 위한 동기 신호로 Diff/Photo 입력을 선택했을 때 적용됩니다. 0 ~ 7.</p> <p>unsigned long PhotoOutDistance : Encoder 입력을 동기 신호로 선택했을 때 적용됩니다. 규칙적으로 반복되는 동작점의 거리를 Encoder 개수로 입력합니다. 24비트 값입니다.</p> <p>unsigned short DelayTime : Photo Out 출력을 위한 조건에 맞았을 때</p>

	<p>신호 출력 전에 대기시간을 갖기 위한 값입니다. usec 단위로 입력하고 0 ~ 6000 범위입니다.</p> <p>unsigned short ActiveTime : Photo Out 출력의 Pulse Width 을 위한 값입니다. usec 단위로 입력하고 1 ~ 4000 범위입니다.</p> <p>unsigned long PhotoOutCount : Photo Out 출력을 위한 조건이 맞았을 때 반복해서 출력시키는 개수입니다. 24비트 값입니다.</p> <p>unsigned char PhotoZCheck : Photo Out 동작의 기준을 Encoder 의 Z 상 입력으로 시작합니다.</p>
리턴 값	<p>BOOL :</p> <p>TRUE : 정상적으로 Photo Out 조건 데이터를 설정했습니다.</p> <p>FALSE : 정상적인 객체 번호가 아니거나 전달 인자가 잘못되었습니다.</p>
부가 설명	

28. ALT_TDIV_PhotoOutDataRead	
함수 원형	BOOL ALT_TDIV_PhotoOutDataRead (int Index, char PhotoIndex, char *PhotoOutSource, BOOL *PhotoOutRepeat, BOOL *PhotoOutPole, char *PhotoOutReverse, unsigned long *PhotoOutOffset, char *PhotoOutTriggerNum, unsigned long *PhotoOutDistance, unsigned short *DelayTime, unsigned short *ActiveTime, unsigned long *PhotoOutCount, unsigned char *PhotoZCheck) ;
동작 설명	Interface 객체번호를 사용해서 Controller 의 Photo Out 출력을 위한 조건 데이터를 읽습니다.
전달 인자	int Index : interface 객체 번호 char PhotoIndex : Photo Output Index. 0 ~ 7. char *PhotoOutSource : char 데이터 포인터. BOOL *PhotoOutRepeat : BOOL 데이터 포인터. BOOL *PhotoOutPole : BOOL 데이터 포인터. char *PhotoOutReverse : char 데이터 포인터. unsigned long *PhotoOutOffset : unsigned long 데이터 포인터. char *PhotoOutTriggerNum : char 데이터 포인터. unsigned long *PhotoOutDistance : unsigned long 데이터 포인터. unsigned short *DelayTime : unsigned short 데이터 포인터. unsigned short *ActiveTime : unsigned short 데이터 포인터. unsigned long *PhotoOutCount : unsigned long 데이터 포인터. unsigned char *PhotoZCheck : unsigned char 데이터 포인터.
리턴 값	BOOL : TRUE : 정상적으로 Photo Out 조건 데이터를 읽었습니다. FALSE : 정상적인 객체 번호가 아니거나 전달 인자가 잘못되었습니다.
부가 설명	

29. ALT_TDIV_DiffOutDataSend	
함수 원형	BOOL ALT_TDIV_DiffOutDataSend (int Index, char DiffIndex, char DiffOutSource, BOOL DiffOutRepeat, BOOL DiffOutPole, char DiffOutReverse, unsigned long DiffOutOffset, char DiffOutTriggerNum, unsigned long DiffOutDistance, unsigned short DelayTime, unsigned short ActiveTime, unsigned long DiffOutCount, unsigned char DiffZCheck) ;
동작 설명	Interface 객체번호를 사용해서 Controller 의 Diff Out 출력을 위한 조건 데이터를 설정합니다.
전달 인자	<p>int Index : interface 객체 번호</p> <p>char DiffIndex : Diff Output Index. 0 ~ 15.</p> <p>char DiffOutSource : Diff Output 동기 신호를 선택합니다. 0 : Encoder 신호 값에 의한 출력 선택. 1 : Diff Input 신호를 동기로 출력 선택. 2 : Photo Input 신호를 동기호 출력 선택.</p> <p>BOOL DiffOutRepeat : Diff Output 동기 조건이 맞았을 때 입력된 OutCount 만큼 연속해서 출력되는 모드 선택합니다. TRUE : 연속 출력 모드 선택. FALSE : 출력 조건에 대해 1회 출력 모드 선택.</p> <p>BOOL DiffOutPole : Diff Output 출력의 극성을 선택합니다. TRUE : Rising-Edge FALSE : Falling-Edge</p> <p>char DiffOutReverse : Encoder 입력을 동기 신호로 선택했을 때 적용됩니다. 0 : Positive Direction 출력 선택. 1 : Negative Direction 출력 선택. 2 : Pos./Neg. Direction 출력 선택.</p> <p>unsigned long DiffOutOffset : Encoder 입력을 동기 신호로 선택했을 때 적용됩니다. Encoder 신호의 시작점에서 동작점까지의 Encoder 개수를 입력합니다. 24비트 값입니다.</p> <p>char DiffOutTriggerNum : Diff Output 출력을 위한 동기 신호로 Diff/Photo 입력을 선택했을 때 적용됩니다. 0 ~ 7.</p> <p>unsigned long DiffOutDistance : Encoder 입력을 동기 신호로 선택했을 때 적용됩니다. 규칙적으로 반복되는 동작점의 거리를 Encoder 개수로 입력합니다. 24비트 값입니다.</p> <p>unsigned short DelayTime : Photo Out 출력을 위한 조건에 맞았을 때</p>

	<p>신호 출력 전에 대기시간을 갖기 위한 값입니다. usec 단위로 입력하고 0 ~ 6000 범위입니다.</p> <p>unsigned short ActiveTime : Diff Out 출력의 Pulse Width 을 위한 값입니다. usec 단위로 입력하고 1 ~ 4000 범위입니다.</p> <p>unsigned long DiffOutCount : Diff Out 출력을 위한 조건이 맞았을 때 반복해서 출력시키는 개수입니다. 24비트 값입니다.</p> <p>unsigned char DiffZCheck : Diff Out 동작의 기준을 Encoder 의 Z 상 입력으로 시작합니다.</p>
리턴 값	<p>BOOL :</p> <p>TRUE : 정상적으로 Diff Out 조건 데이터를 설정했습니다.</p> <p>FALSE : 정상적인 객체 번호가 아니거나 전달 인자가 잘못되었습니다.</p>
부가 설명	

30. ALT_TDIV_DiffOutDataRead	
함수 원형	BOOL ALT_TDIV_DiffOutDataRead (int Index, char DiffIndex, char *DiffOutSource, BOOL *DiffOutRepeat, BOOL *DiffOutPole, char *DiffOutReverse, unsigned long *DiffOutOffset, char *DiffOutTriggerNum, unsigned long *DiffOutDistance, unsigned short *DelayTime, unsigned short *ActiveTime, unsigned long *DiffOutCount, unsigned char *DiffZCheck) ;
동작 설명	Interface 객체번호를 사용해서 Controller 의 Diff Out 출력을 위한 조건 데이터를 읽습니다.
전달 인자	int Index : interface 객체 번호 char DiffIndex : Diff Output Index. 0 ~ 15. char *DiffOutSource : char 데이터 포인터. BOOL *DiffOutRepeat : BOOL 데이터 포인터. BOOL *DiffOutPole : BOOL 데이터 포인터. char *DiffOutReverse : char 데이터 포인터. unsigned long *DiffOutOffset : unsigned long 데이터 포인터. char *DiffOutTriggerNum : char 데이터 포인터. unsigned long *DiffOutDistance : unsigned long 데이터 포인터. unsigned short *DelayTime : unsigned short 데이터 포인터. unsigned short *ActiveTime : unsigned short 데이터 포인터. unsigned long *DiffOutCount : unsigned long 데이터 포인터. unsigned char *DiffZCheck : unsigned char 데이터 포인터.
리턴 값	BOOL : TRUE : 정상적으로 Diff Out 조건 데이터를 읽었습니다. FALSE : 정상적인 객체 번호가 아니거나 전달 인자가 잘못되었습니다.
부가 설명	

31. ALT_TDIV_SaveSend	
함수 원형	BOOL ALT_TDIV_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 각 기능의 환경 및 동작 설정 데이터 입니다. 전원을 On 시에 저장된 데이터로 자동 동작합니다.

32. ALT_TDIV_LoadSend	
함수 원형	BOOL ALT_TDIV_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 각 기능의 환경 및 동작 설정 데이터 입니다. 저장되어 있던 환경 설정 데이터가 Controller 에 적용되면 현재 사용중이던 환경 설정 데이터는 지워집니다.

33. ALT_TDIV_EncClearSend	
함수 원형	BOOL ALT_TDIV_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	

34. ALT_TDIV_EncCountRead	
함수 원형	BOOL ALT_TDIV_EncCountRead (int Index, unsigned long *SEncCount, unsigned long *AEncCount, unsigned long *DEncCount, unsigned long *ZEncCount) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 입력되는 Encoder Count 를 읽습니다.
전달 인자	int Index : interface 객체 번호 unsigned long *SEncCount : Encoder 입력을 동기신호로 선택하면 표시되는 Count 값입니다. 내부 Timer 를 동기신호로 사용하면 표시되지 않습니다. unsigned long *AEncCount : Reserved. SEncCount 값과 동일하게 생각하면 됩니다. unsigned long *DEncCount : Encoder/Timer 의 동기 신호 입력 Count 값입니다. 각 Encoder Out, Diff Out, Photo Out 에 공급되는 신호의 Count 값입니다. unsigned long *ZEncCount : Encoder 의 Z 상의 입력 카운트 값을 읽습니다.
리턴 값	BOOL : TRUE : 정상적으로 Encoder Count 를 읽었습니다. FALSE : 정상적인 객체 번호가 아니거나 전달 인자가 잘못되었습니다.
부가 설명	

35. ALT_TDIV_SoftTriggerSend	
함수 원형	BOOL ALT_TDIV_SoftTriggerSend (int Index, unsigned char SoftTrigger) ;
동작 설명	Interface 객체번호를 사용해서 Controller 에 Software 통해서 Trigger 입력을 전달합니다.
전달 인자	int Index : interface 객체 번호 unsigned char SoftTrigger : Trigger 입력의 개수가 Default 로 8개 이므로 unsigned char 데이터로 전달합니다.
리턴 값	BOOL : TRUE : 정상적으로 Soft-Trigger 가 전달되었습니다. FALSE : 정상적인 객체 번호가 아닙니다.
부가 설명	SoftTrigger 는 다음과 같이 입력됩니다. 0 번 : SoftTrigger:0x01 1 번 : SoftTrigger:0x02 2 번 : SoftTrigger:0x04 3 번 : SoftTrigger:0x08 4 번 : SoftTrigger:0x10 5 번 : SoftTrigger:0x20 6 번 : SoftTrigger:0x40 7 번 : SoftTrigger:0x80