

# CX\_VAR\_ANALOG\_CTRL

Header File 경로

"C:\Module\_Linker\EQP\_Link\_Res\Apps\_Header\EQP\INTERFACE\Object\_Interface.h"

class CII\_VAR\_ANALOG\_CTRL

Example)

```
CX_VAR_ANALOG_CTRL aCH_NAME_XX;
```

```
CString var_name;
```

```
var_name = "Variable.Name";
```

```
STD_ADD_ANALOG( var_name, "%", 1, 0.0, 100.0 );
```

```
LINK_VAR_ANALOG_CTRL( aCH_NAME_XX, var_name );
```

## Get\_CHANNEL\_NAME 함수

class에 연결된 Channel 이름 (Object\_Name+Variable\_Name) 을 전달 받는다.

### Syntax

```
CString Get_CHANNEL_NAME( );
```

### Parameter

없음

### Return Value

Channel 이름을 CString 으로 전달 받음.

## Set\_DATA 함수

String 를 Analog으로 변환하여 데이터를 저장한다.

### Syntax

```
int Set_DATA( const CString& data );
```

### Parameter

data : 설정 할 값 (String)

### Return Value

> 0 : 성공 했을 때  
< 0 : 실패 했을 때 (IO Channel인 경우 Driver에서 Error 처리하거나. Setpoint Interlock에서 Error 처리할 때 발생)

## Get\_DATA 함수

Analog 값을 String 으로 변환하여 데이터를 가져온다.

### Syntax

```
int Get_DATA( CString& data );
```

### Parameter

data : 현재 값을 string 타입으로 가져온다.

### Return Value

> 0 : 성공 했을 때  
< 0 : 실패 했을 때

## Check\_DATA 함수

Parameter 값과 현재 Channel 값이 일치하는지 확인한다.

### Syntax

```
int Check_DATA( const CString& data );
```

### Parameter

data : 현재 Channel 값과 비교한다.

### Return Value

> 0 : 일치 했을 때  
< 0 : 일치하지 않을 때

## Get\_VARIABLE\_NAME 함수

class에 연결된 Variable 이름을 전달 받는다.

### Syntax

```
CString Get_VARIABLE_NAME( );
```

### Parameter

없음

### Return Value

Class 연결된 Variable 값

## Set\_VALUE 함수

Channel 값을 설정한다.

### Syntax

```
int Set_VALUE( const double value );
```

### Parameter

value : 설정 값

### Return Value

- > 0 : 설정이 성공 했을 때
- < 0 : 설정이 실패 했을 때  
(IO Channel인 경우 Driver에서 Error 처리하거나. Setpoint Interlock에서 Error 처리할 때 발생)

## Inc\_VALUE 함수

현재 Channel 값을 Parameter 값 만큼 증가시킨다.

### Syntax

```
int Inc_VALUE( const double value = 1.0 );
```

### Parameter

value : 현재 값을 기준으로 증가시킬 설정 값

### Return Value

- > 0 : 설정이 성공 했을 때
- < 0 : 설정이 실패 했을 때  
(IO Channel인 경우 Driver에서 Error 처리하거나. Setpoint Interlock에서 Error 처리할 때 발생)



## When\_LESS\_THAN 함수

현재 Channel 값이 Ref. 값 보다 작아질 때까지 기다린다. ( < )

### Syntax

```
int When_LESS_THAN( const double ref_value, const double sec = -1 );
```

### Parameter

ref\_value : 현재 Channel 값과 비교할 기준 값

sec : timeout 값

( 0 보다 작은 값일 때는 무한정 기다림)

### Return Value

> 0 : 조건이 만족될 때

== 0 : 연결된 상위 Object이 Abort 될 때

```
int Link_UPPER_OBJECT_ABORT(const CString& obj_name);
```

```
int Enable_UPPER_OBJECT_ABORT_CHECK();
```

< 0 : Timeout 발생시

## When\_MORE\_THAN 함수

현재 Channel 값이 Ref. 값 보다 커질 때까지 기다린다. ( > )

### Syntax

```
int When_MORE_THAN( const double ref_value, const double sec = -1 );
```

### Parameter

ref\_value : 현재 Channel 값과 비교할 기준 값

sec : timeout 값

( 0 보다 작은 값일 때는 무한정 기다림)

### Return Value

> 0 : 조건이 만족될 때

== 0 : 연결된 상위 Object이 Abort 될 때

```
int Link_UPPER_OBJECT_ABORT(const CString& obj_name);
```

```
int Enable_UPPER_OBJECT_ABORT_CHECK();
```

< 0 : Timeout 발생시

## Link\_UPPER\_OBJECT\_ABORT 함수

지연 함수 수행 시 Abort 처리할 Object 이름을 연결한다.

### Syntax

```
int Link_UPPER_OBJECT_ABORT( const CString& obj_name );
```

### Parameter

object : Abort 처리할 상위 Object 이름

\* 관련 함수

When\_\_LESS\_THAN( ... )

When\_\_MORE\_THAN( ... )

### Return Value

> 0 : Object 연결이 정상적으로 이루어질 때

< 0 : Object 연결이 실패 했을 때

# Enable\_\_UPPER\_OBJECT\_ABORT\_CHECK 함수

연결된 Object이 Abort 될 때 연동 되도록 활성화 된다.

## Syntax

```
int Enable__UPPER_OBJECT_ABORT_CHECK( );
```

## Parameter

아래의 함수가 수행 중일 때 Abort 처리 되도록 연결된 상위 Object 의 CTRL Channel을 Monitoring한다.

- \* When\_\_LESS\_THAN( ... )
- \* When\_\_MORE\_THAN( ... )

## Return Value

- > 0 : Object 연결이 정상적으로 이루어질 때
- < 0 : Object 연결이 실패 했을 때

# Disable\_\_UPPER\_OBJECT\_ABORT\_CHECK 함수

연결된 Object이 Abort 될 때 연동 되지 않도록 설정 된다.

## Syntax

```
int Disable__UPPER_OBJECT_ABORT_CHECK( );
```

## Parameter

아래의 함수가 수행 중일 때 Abort 처리 되지 않도록 연결된 상위 Object 의 CTRL Channel을 Monitoring 하지 않는다.

- \* When\_\_LESS\_THAN( ... )
- \* When\_\_MORE\_THAN( ... )

## Return Value

- > 0 : Object 연결이 정상적으로 이루어질 때
- < 0 : Object 연결이 실패 했을 때

## Get\_MIN\_MAX\_DEC 함수

연결된 Channel의 Min 값, Max 값 그리고 소수점 몇째 자리까지 표기하는지 정보를 가져온다.

### Syntax

```
int Get_MIN_MAX_DEC( double& min_value, double& max_value, int& i_dec );
```

### Parameter

min\_value : 연결된 Chanel의 Min 값  
max\_value : 연결된 Chanel의 Max 값  
i\_dec : 소수점 자리 표기

### Return Value

- > 0 : Chanel 정보를 정상적으로 가져올 때
- < 0 : Chanel 정보를 정상적으로 가져오지 못할 때

## Get\_MIN\_MAX\_INFO 함수

연결된 Channel의 Min 값과 Max 값을 String 타입의 정보로 가져온다.

### Syntax

```
int Get_MIN_MAX_INFO( CString& str_min, CString& str_max );
```

### Parameter

str\_min : 연결된 Chanel의 Min 값 (String Type)

str\_max : 연결된 Chanel의 Max 값 (String Type)

### Return Value

- > 0 : Chanel 정보를 정상적으로 가져올 때
- < 0 : Chanel 정보를 정상적으로 가져오지 못할 때

# Change\_MIN\_MAX\_DEC 함수

연결된 Channel의 Min 값, Max 값 그리고 소수점 자리 표기를 변경한다.

## Syntax

```
int Change_MIN_MAX_DEC( const double min_value, const double max_value, const int i_dec );
```

## Parameter

min\_value : 연결된 Chanel의 Min 값을 설정된 Parameter 값으로 변경  
max\_value : 연결된 Chanel의 Max 값을 설정된 Parameter 값으로 변경  
i\_dec : 소수점 자리 표기를 설정된 Parameter 값으로 변경

## Return Value

- > 0 : Chanel 정보가 정상적으로 변경될 때
- < 0 : Chanel 정보를 정상적으로 변경되지 않을 때



## Check\_CHANNEL\_NAME 함수

연결된 Channel의 이름이 일치하는지 확인하다.

### Syntax

```
int Check_CHANGE_NAME( const CString& ch_name );
```

### Parameter

ch\_name : 연결된 Chanel의 이름과 비교할 이름 값

### Return Value

- > 0 : Chanel 이름과 설정된 Parameter 값이 일치할 때
- < 0 : Chanel 이름과 설정된 Parameter 값이 일치하지 않을 때

## Disable\_\_AUTO\_POLLING 함수

연결된 Channel이 IO Reading Type인 경우 Auto Polling 기능을 중지시킨다.

### Syntax

```
int Disable__AUTO_POLLING( );
```

### Parameter

없음

### Return Value

> 0 : 성공 했을 때  
< 0 : 실패 했을 때

## Enable\_\_AUTO\_POLLING 함수

연결된 Channel이 IO Reading Type인 경우 설정된 sec로 Auto Polling 주기를 재설정시킨다.

### Syntax

```
int Enable__AUTO_POLLING( const double poll_sec );
```

### Parameter

poll\_sec : 설정된 값으로 Auto polling 주기를 재설정한다.

### Return Value

> 0 : 성공 했을 때  
< 0 : 실패 했을 때

# Update\_CHANGE\_DATA 함수

Data 관련 Change Count 값을 증가시켜 Module\_Link를 통해 정보를 강제 전송할 때 사용됨

## Syntax

```
int Update_CHANGE_DATA( );
```

## Parameter

없음

## Return Value

> 0 : 성공 했을 때  
< 0 : 실패 했을 때

## Check\_LESS\_THAN 함수

연결된 Channel의 값이 설정 값 보다 작은지 ( < ) 비교한다.

### Syntax

```
int Check_LESS_THAN( const double check_value );
```

### Parameter

check\_value : 비교 값

### Return Value

- > 0 : 현재 Channel 값이 비교 값 보다 작을 때
- < 0 : 현재 Channel 값이 비교 값 보다 클 때

## Check\_MORE\_THAN 함수

연결된 Channel의 값이 설정 값 보다 큰지 ( > ) 비교한다.

### Syntax

```
int Check_MORE_THAN( const double check_value );
```

### Parameter

check\_value : 비교 값

### Return Value

- > 0 : 현재 Channel 값이 비교 값 보다 클 때
- < 0 : 현재 Channel 값이 비교 값 보다 작을 때

## Check\_VARIABLE\_NAME 함수

연결된 Channel의 Variable 이름이 설정 값과 같은지 비교한다.

### Syntax

```
int Check_VARIABLE_NAME( const CString& var_name );
```

### Parameter

var\_name : Channel의 Variable 이름을 확인하기 위한 변수

### Return Value

- > 0 : 연결된 Channel의 Variable 이름과 일치 할 때
- < 0 : 연결된 Channel의 Variable 이름과 일치 하지 않을 때