

CX_VAR_STRING_CTRL

Header File 경로

"C:\Module_Linker\EQP_Link_Res\Apps_Header\EQP\INTERFACE\Object_Interface.h"

class CII_VAR_STRING_CTRL

Example)

```
CX_VAR_STRING_CTRL sCH_NAME_XX;
```

```
CString var_name;
```

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

```
STD_ADD_STRING( var_name );
```

```
LINK_VAR_STRING_CTRL( sCH_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

설정 값을 저장한다.

Syntax

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

Parameter

data : 설정 값

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

현재 Channel 값을 String 타입 데이터로 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

When_DATA 함수

현재 Channel 값이 Parameter 값과 일치할 때까지 기다린다.

Syntax

```
int When_DATA( const CString& data, const double sec = -1 );  
int When_DATA( const CString& data1, const CString& data2, const double sec = -1 );
```

Parameter

data : 현재 Channel 값과 일치하는지 확인 할 Parameter 값
* data1 과 data2 는 OR 조건임.
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_NOT_DATA 함수

현재 Channel 값이 Parameter 값과 일치하지 않을 때까지 기다린다.

Syntax

```
int When_NOT_DATA( const CString& data, const double sec = -1 );  
int When_NOT_DATA( const CString& data1, const CString& data2, const double sec = -1 );
```

Parameter

data : 현재 Channel 값과 다른지 확인 할 Parameter 값
* data1 과 data2 는 AND 조건임.
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

Get_VARIABLE_NAME 함수

class에 연결된 Variable 이름을 전달 받는다.

Syntax

```
CString Get_VARIABLE_NAME( );
```

Parameter

없음

Return Value

Class 연결된 Variable 값

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_DATA(...)

When_NOT_DATA(...)

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

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

When_DATA_LIST 함수

현재 Channel 값이 Parameter List 중에 하나라도 일치할 때까지 기다린다.

Syntax

```
int When_DATA_LIST( const CStringArray& l_data, const double sec = -1 );
```

Parameter

l_data : 현재 Channel 값과 일치하는지 확인 할 Parameter List
* list 내 설정 값들은 OR 조건임.

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

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

Get_STRING 함수

현재 Channel 값을 String 타입 데이터로 가져온다.

Syntax

```
int Get_DATA_STRING( );
```

Parameter

없음

Return Value

현재 Channel 값을 String 타입으로 가져온다.

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