

SCX_ASYNC_TIMER_CTRL

Header File 경로

"C:\Module_Linker\EQP_Link_Res\Apps_Header\EQP\UTILITY\EQP_BODY__UTILITY_LINK.h"

class SCI_ASYNC_TIMER_CTRL

Example)

```
SCX_ASYNC_TIMER_CTRL x_timer;
```

```
x_timer->START_COUNT_UP( 10.0 );           // 0.0부터 count 시작하고 곧바로 함수가 리턴 된다.
```

```
// any code ...
```

```
double cur_sec = x_timer->Get_COUNT_TIME(); // 위의 start_count_up( ) 진행 후 경과된 시간(sec)를 리턴합니다.
```

REGISTER_COUNT_CHANNEL 함수

counting 되는 time 값을 channel 값으로 연결하는 함수.

Syntax

```
int REGISTER_COUNT_CHANNEL( const CString& object_name, const CString& variable_name );
```

Parameter

object_name : 연결할 Channel의 Object Name을 의미합니다.

variable_name : 연결할 Channel의 Variable Name을 의미합니다.

Return Value

> 0 : Channel 연결이 성공 했을 때

< 0 : Channel 연결이 실패 했을 때

REGISTER_COUNT_CHANNEL 함수

counting 되는 time 값을 channel 값으로 연결하는 함수.

Syntax

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

Parameter

ch_name : 연결할 Channel의 Name을 의미합니다.

Return Value

- > 0 : Channel 연결이 성공 했을 때
- < 0 : Channel 연결이 실패 했을 때

START_COUNT_DOWN 함수

설정된 값부터 0.0 될 때까지 counting down를 시작한다.

Syntax

```
int START_COUNT_DOWN( const double sec );
```

Parameter

sec : 설정된 sec 값부터 counting down 한다. (단위 sec).

Return Value

> 0 : 성공 했을 때
< 0 : 실패 했을 때

START_COUNT_UP 함수

0.0 부터 설정된 값까지 counting up를 시작한다.

Syntax

```
int START_COUNT_UP( const double sec );
```

Parameter

sec : counting up 할 target 값 (단위 sec).

Return Value

> 0 : 성공 했을 때
< 0 : 실패 했을 때

Get_CURRENT_TIME 함수

timer start 이후 경과된 시간 (sec) 값을 리턴 한다.

Syntax

```
int Get_CURRENT_TIME( );
```

Parameter

없음

Return Value

- > 0 : Channel 연결이 성공 했을 때
- < 0 : Channel 연결이 실패 했을 때

STOP 함수

start된 timer을 정지시킨다.

Syntax

```
int STOP( );
```

Parameter

없음

Return Value

> 0 : Object 연결이 성공 했을 때
< 0 : Object 연결이 실패 했을 때

STOP_ZERO 함수

start된 timer을 정지시키고 값을 0.0으로 초기화한다.

Syntax

```
int STOP_ZERO( );
```

Parameter

없음

Return Value

의미 없음

PAUSE 함수

start된 timer을 일시 중지시킨다.

Syntax

```
int PAUSE( );
```

Parameter

없음

Return Value

> 0 : Object 연결이 성공 했을 때
< 0 : Object 연결이 실패 했을 때

RESUME 함수

일시 중지된 timer를 재실행 시킨다.

Syntax

```
int RESUME( );
```

Parameter

없음

Return Value

> 0 : Object 연결이 성공 했을 때
< 0 : Object 연결이 실패 했을 때

Get_TIMER_DB_MAX_SIZE 함수

timer DB의 Max 값을 리턴 한다.

Syntax

```
int Get_TIMER_DB_MAX_SIZE( );
```

Parameter

없음

Return Value

timer DB의 Max 값을 리턴 한다.

Get_TIMER_SIZE 함수

현재 활성화된 timer의 전체 수를 리턴한다.

Syntax

```
int Get_TIMER_SIZE( );
```

Parameter

없음

Return Value

현재 실행중인 전체 timer 수를 리턴 한다.

Get_TIMER_INDEX 함수

현재 실행된 timer의 ID 리턴한다.

Syntax

```
int Get_TIMER_INDEX( );
```

Parameter

없음

Return Value

현재 실행중인 timer의 ID를 리턴 한다.

INIT_STOPWATCH 함수

Stop-Watch 기능을 초기화 한다.

Syntax

```
int INIT_STOPWATCH( );
```

Parameter

없음

Return Value

Get_mSEC_STOPWATCH 함수

Stop-Watch 시작 후 경과된 시간(msec)을 리턴 한다.

Syntax

```
int Get_mSEC_STOPWATCH( );
```

Parameter

없음

Return Value

경과된 시간 (단위 : msec)

Get_SEC_STOPWATCH 함수

Stop-Watch 시작 후 경과된 시간(sec)을 리턴 한다.

Syntax

```
int Get_SEC_STOPWATCH( );
```

Parameter

없음

Return Value

경과된 시간 (단위 : sec)