

SCX_SERIAL_JG

Header File 경로

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

Class SCI_SERIAL_JG

Example)

```
SCX_SERIAL_JG x_serial;
```

```
int com_port = 5;
```

```
int rate = 9600;
```

```
int byte = 8;
```

```
int stop = 1;
```

```
int parity = 0;    // 0 : none
```

```
x_serial->INIT_COMPORT( com_port, rate, byte, stop, parity );
```

```
x_serial->OPEN_COMPORT();
```

INIT_COMPORT 함수

Serial 통신을 위해 port, rate, byte, stop, parity를 초기화 및 설정해주는 함수.

Syntax

```
int INIT_COMPORT(const int port, const DWORD rate, const BYTE byte, const BYTE stop, const BYTE parity);
```

Parameter

port	:	com port id	
rate	:	baud rate	// 1200, 1800, 2400, 4800, 7200, 9600, 19200, 38400 사용
byte	:	data bit	// 5, 6, 7, 8 사용
stop	:	stop bit	// 1, 2 사용
parity	:	parity 설정	// None, Even, Odd, Mark, Space 사용

Return Value

> 0	:	Initialize 성공 했을 때
< 0	:	Initialize 실패 했을 때

Get_COMPORT_HANDLE 함수

현재 Open 되어 있는 Com-Port의 Handle을 전달 받는다.

Syntax

```
HANDLE Get_COMPORT_HANDLE( );
```

Parameter

없음

Return Value

현재 Open된 com port handle 값
Com_Port가 open 되지 않은 상태이면, INVALID_HANDLE_VALUE 값을 리턴한다.

CLEAR_BUFFER 함수

com_port 수신 buffer에 있는 데이터를 받아오고, buffer를 Clear 한다.

Syntax

```
int CLEAR_BUFFER( Cstring* r_msg);
```

Parameter

*r_msg : 수신 buffer에 있는 데이터를 받아온다.

Return Value

> 0 : 성공 했을 때
< 0 : 실패 했을 때

OPEN_COMPORT 함수

INIT_COMPORT에 설정된 정보에 맞게 serial com port 를 open 합니다.

Syntax

```
int OPEN_COMPORT( );
```

Parameter

없음

Return Value

> 0 : 성공 했을 때
< 0 : 실패 했을 때

CLOSE_COMPORT 함수

open된 com port 를 close 합니다.

Syntax

```
int CLOSE_COMPORT( );
```

Parameter

없음

Return Value

> 0 : 성공 했을 때
< 0 : 실패 했을 때

DATA_SEND 함수

data 를 보냅니다.

Syntax

```
int DATA_SEND( const char* data );  
int DATA_SEND( const CString& data);
```

Parameter

data : 전송할 data (char type, 또는 CString type)

Return Value

> 0 : 성공 했을 때
< 0 : 실패 했을 때

CHAR_SEND 함수

hexa data 를 보냅니다.

Syntax

```
int CHAR_SEND( unsigned char* s_data, const int s_len );
```

Parameter

s_data : 전송할 data 값
s_len : 전송할 data 수

Return Value

> 0 : 성공 했을 때
< 0 : 실패 했을 때

DATA_RECV 함수

data 전송 후 수신 data 중에 종결 문자를 받으면 종결 문자를 제외한 수신 data를 전달한다.

Syntax

```
int DATA_RECV( const char *end_str, char * data, const int msec );  
int DATA_RECV( const char *wait_ch, char * data, const char* end_str, const int msec );  
int DATA_RECV( const char *wait_str, char * data, const char* end_str, const int msec );  
int DATA_RECV( const char *w_str1, const char *w_str2, char * data, const char* end_str, const int msec );  
int DATA_RECV( const CString& end_str, CString* data, const int msec );  
int DATA_RECV( const char end_char, char * data, const int msec );  
int DATA_RECV( const char end_char, CString* data, const int msec );
```

Parameter

end_str : 수신 data의 마지막 문자열
end_char : 수신 data의 마지막 문자
wait_ch : data를 수신할 때 시작 부분의 문자가 수신될 때까지 기다린다.
wait_str : data를 수신할 때 시작 부분의 문자열이 수신될 때까지 기다린다.
data : 수신된 data
msec : timeout (단위 : millisecond)
w_str1 : data 수신시 체크할 첫번째 문자열
w_str2 : data 수신시 체크할 두번째 문자열

Return Value

> 0 : 성공 했을 때
< 0 : 실패 했을 때 (Timeout 발생 또는 data 수신시 종결 문자를 받지 못한 경우)

CHAR_RECV 함수

data의 값이 null 아닌 길이가 0 보다 크면, 데이터를 전송 후 데이터를 수신하는 함수
수신 data의 개수를 확인하거나 수신 data 중에 종결 문자를 제외한 수신 data를 전달한다.

Syntax

```
int CHAR_RECV( unsigned char* data, int* r_ret, const int msec );  
int CHAR_RECV( const unsigned char wait_ch, unsigned char* data, int* r_ret, const int msec );  
int CHAR_RECV( unsigned char* data, const unsigned char end_ch, int* r_ret, const int msec );
```

Parameter

data : 수신 data
wait_ch : data를 수신할 때 시작 부분의 문자가 수신될 때까지 기다린다.
msec : timeout (단위 : millisecond)
end_ch : 수신 data의 마지막 문자
r_cnt : 수신된 data의 길이

Return Value

> 0 : 성공 했을 때
< 0 : 실패 했을 때 (Timeout 발생 또는 data 수신시 종결 문자를 받지 못한 경우)

RTS_ENABLE 함수

H/W RTS를 작동시키는 함수

Syntax

```
int RTS_ENABLE( );
```

Parameter

없음

Return Value

> 0 : 성공 했을 때
< 0 : 실패 했을 때

RTS_DISABLE 함수

H/W RTS를 중지시키는 함수

Syntax

```
int RTS_DISABLE( );
```

Parameter

없음

Return Value

> 0 : 성공 했을 때
< 0 : 실패 했을 때

DTR_ENABLE 함수

H/W DTR을 작동시키는 함수

Syntax

```
int DTR_ENABLE( );
```

Parameter

없음

Return Value

> 0 : 성공 했을 때
< 0 : 실패 했을 때

DTR_DISABLE 함수

H/W DTR를 중지시키는 함수

Syntax

```
int DTR_DISABLE( );
```

Parameter

없음

Return Value

> 0 : 성공 했을 때
< 0 : 실패 했을 때

INIT_COMPORT_WITHOUT_POPUP 함수

위의 INIT_COMPORT 함수와 기능은 동일하나 Initial 실패시 Error Message 창이 나타나지 않는다.

Syntax

```
int INIT_COMPORT(const int port, const DWORD rate, const BYTE byte, const BYTE stop, const BYTE parity);
```

Parameter

port : com port id
rate : baud rate // 1200, 1800, 2400, 4800, 7200, 9600, 19200, 38400 사용
byte : data bit // 5, 6, 7, 8 사용
stop : stop bit // 1, 2 사용
parity : parity 설정 // None, Even, Odd, Mark, Space 사용

Return Value

> 0 : Initialize 성공 했을 때
< 0 : Initialize 실패 했을 때

OnlyRecv_DATA 함수

오직 data 수신만 담당하는 함수 입니다.

Syntax

```
int OnlyRecv_DATA( const char* end_str, char* data, const int msec );  
int OnlyRecv_DATA( const char wait_ch, char* data, const char* end_str, const int msec );  
int OnlyRecv_DATA( const char* wait_str, char* data, const char* end_str, const int msec );  
int OnlyRecv_DATA( const CString& end_str, CString* data, const int msec );  
int OnlyRecv_DATA( unsigned char* data, int* r_cnt, const int msec );
```

Parameter

data : 수신 data (type : char*, unsigned char*, CString*)
wait_ch : data를 수신할 때 시작 부분의 문자가 수신될 때까지 기다린다.
wait_str : data를 수신할 때 시작 부분의 문자열이 수신될 때까지 기다린다.
msec : timeout (단위 : millisecond)
end_str : 수신 data의 마지막 문자열
r_cnt : 수신된 data의 길이

Return Value

> 0 : Initialize 성공 했을 때
< 0 : Initialize 실패 했을 때