

EZGemPlus DII User Manual

USER'S MANUAL

(Ver 2.7.3)

2022.03.31

www.nviasoft.co.kr

대표전화) 031-776-2250

팩스(Fax) 031-776-2252

Email) all@nviasoft.co.kr

주소 : 경기 성남시 중원구 사기막골로 124 SKn테크노파크 메가동 813호



목차

1. Edit History	4
2. EZGEM이란	5
3. GEM Dll Libray 등록	6
4. 환경설정(START 전).....	7
(1) OnEZGemEvent / OnEZGemMsg 함수 추가.....	7
(2) MDLN/SOFTREV 값 세팅.....	11
(3) SVID관련 등록 및 값 세팅.....	12
(3) RemoteCommand(RCMD) 등록.....	17
(4) Spool 관련 항목 세팅.....	17
(5) 사용자 정의 메시지(User Message define) 정의.....	18
(6) 사용자 정의 메시지(User Message define) 처리.....	19
(6) LOG파일 관련 함수.....	20
void EnableLog() / void DisableLog().....	20
void SetLogFile(string strFileNam)	20
void SetLogRetention(short nDays).....	20
int GetLogRetention()	20
void SkipLog(short nStream , short Function).....	20
void SkipLogReset(short nStream , short Function).....	20
4. GEM Driver 구동 및 정지	21
5. GEM Driver Event 발생에 따른 처리.....	23
6. 각 기능별 함수 및 이벤트 정리.....	31

[1] GEM Driver 구동 및 정지	31
[2] SVID 관련	31
[3] 알람(ALID) 관련	31
[4] 이벤트(CEID) 관련	32
[5] ECID 관련	32
[6] Control State 관련	33
[7] Communication 관련	33
[8] SPOOLING 관련	33
[9] Remote Command 관련	34
[10] 직접 Message 구성 관련 함수	35
[11] Terminal Message 관련	35
[12] LOG 파일 관련	35
[13] EZGEM.RPT 파일 관련	36
[14] Transaction Msg 관련	36
[15] GEM에서 제공하는 EVENT (Event ID의 역할)	37

1. Edit History

버전	날짜	변경 내용	작성자
1.0	2018 . 03 . 22	※Initial Release	csseo
1.1	2019 . 01 . 07	※Update Release	skpark
1.1.2	2019 . 02 . 22	※fix Dll	skpark
1.1.5	2019 . 05 . 23	※update Dll(1.6)	skpark
2.2.0	2019. 12. 12	※svid array지원	skpark
2.5.5	2020.07.01	※Big File 디렉토리 생성/삭제 ※GetSVIDValue 추가	skpark
2.6.1	2021.01.18	※GetECValue추가	skpark
	2021.03.04	※SendEventReportEx,SendMsgEx ※GetSysByteEx 함수 추가(36p)	skpark
2.7.2	2021.07.14	※함수 추가 및 안정화 추가	skpark

2. EZGEM이란

ezGem Driver은 라이브리를 통해서 간단하게 반도체 표준 통신 프로토콜인 SECS(SEMI Equipment Communications Standard)를 사용할 수 있게 도와주는 통신 Driver입니다.

SECS를 기반으로 하여 GEM의 기본적인 사양을 자동 처리합니다.

자동 처리할 수 있는 부분은 모듈화하여 간단한 코딩으로도 GEM을 구현 할 수 있습니다.

ezGem의 경우 ActiveX Control(ocx) 의 경우 대부분의 32bit windows개발 환경에서 사용가능합니다.

ezGem Plus의 경우 ActiveX Control을 이용하여 32/64bit 기반의 windows에서 사용가능합니다.

다만, 64bit의 경우 ezGem Dll 버전을 사용하여야 합니다.

Dll을 사용하기 위해서는 dll에 필요한 windows dll이 필요합니다.

해당 파일을 엔비아소프트 홈페이지 자료실에서 다운 받아서 설치하실 수 있습니다.

<http://nviasoft.co.kr/board/view.php?page=1&no=151&id=file>

Microsoft Visual C++ 2010 재배포 가능 패키지(x64) Download url :

https://download.microsoft.com/download/3/2/2/3224B87F-CFA0-4E70-BDA3-3DE650EFEB5/vcredist_x64.exe

Microsoft Visual C++ 2010 재배포 가능 패키지(x86) Download url :

https://download.microsoft.com/download/5/B/C/5BC5DBB3-652D-4DCE-B14A-475AB85EEF6E/vcredist_x86.exe

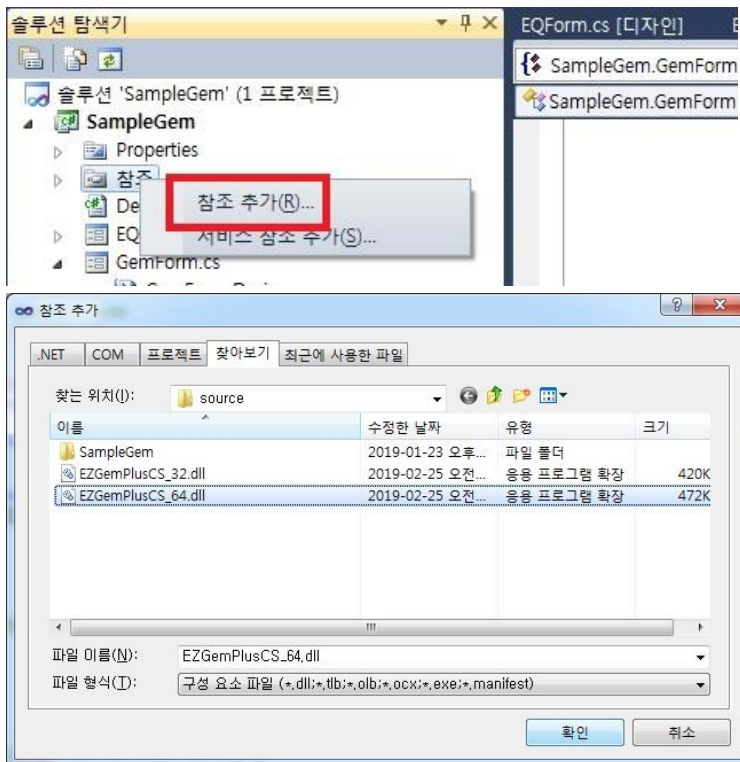
3. GEM Dll Library 등록

1. Dll Library 참조하기

- 1) VS “**솔루션 탐색기**”의 “**참조**” 항목에서 마우스 **우클릭** 후, ‘**참조추가**’클릭
- 2) 도구 상자 항목 선택에서 ‘**찾아보기**’ 클릭
- 3) 설치되어 있는 **EZGemPlusCS_64.dll** 선택하여 이동
- 4) 64비트의 경우 EZGemPlusCS_64.dll 항목 선택 후 ‘**확인**’버튼을 눌러 참조 추가
- 5) 32비트의 경우 EZGemPlusCS_32.dll 항목 선택 후 ‘**확인**’버튼을 눌러 참조 추가
- 6) 소스상에 추가

using EZGemPlusCS;

```
CEZGemPlusLib m_gem = new CEZGemPlusLib();
```



4. 환경설정(START 전)

(1) OnEZGemEvent / OnEZGemMsg 함수 추가

```
CEZGemPlusLib m_gem = new CEZGemPlusLib(); // dll 참조
m_gem.OnEZGemEvent += new ON_EZGEM_EVENT(OnEventReceived);
m_gem.OnEZGemMsg += new ON_EZGEM_MSG(OnMsgReceived);
```

dll 참조 후 참조된 dll의 내장 함수를 이용하여 메시지와 내부 함수를 연결함.

OnEventReceived(Event ID 의 역할)

Event ID	내용	설명
1	Connect	통신이 연결됨.
2	Disconnect	통신 연결이 끊어짐.
401	Msg In	메시지가 들어옴(Stream , Function)
402	Msg Out	메시지를 전송함(Stream , Function)
501	S2F37 IN	메시지 S2F37 응답을 한 경우 발생.
1001	offline	Offline으로 Control이 변경됨.
1002	local	Online Local로 변경됨.
1003	Remote	Online Remote로 변경됨.
1010	Communicating	Communication상태가 됨.
1015	Host ECID	Host ECID(S2F15)가 들어옴.
1030	RemoteCommand	RemoteCommand(S2F41)이 들어옴.
1050	TerminalMessage	Terminal Message(S10F3 / S10F5) 가 들어옴.

```

private void OnEventReceived(IntPtr lpParam, short nEventId, int IParam)
{
    switch (nEventId)
    {
        case 1: // tcp connect
            OnConnected();
            break;
        case 2: // tcp disconnect
            OnDisconnected();
            break;
        case 10: // tcp disconnect
            AddLog( string.format("EZGEM LICENSE IS EVALUATION"));
            break;
        case 11: // tcp disconnect
            AddLog( string.format("EZGEM LICENSE IS PURCHASE"));
            break;
        case 401 : // Msg In
            OnMsgIn(IParam);
            break;
        case 402 : // Msg Out
            OnMsgOut(IParam);
            break;
        case 501 : // S2F37
            OnEventEnableDisable(IParam);
            break;
        case 1001:
            OnOffline();
            break;
        case 1002:
            OnOnlineLocal();
            break;
        case 1003:
            OnOnlineRemote();
            break;
        case 1010:
            OnCummunicating();
    }
}

```

```
        break;
    case 1030 : // S2F41w
        OnRemoteCommand( IParam );
        break;
    case 1015 : // S2F15w
        OnNewHOST_ECID( IParam );
        break;
    case 1050 : // S10F3, S10F5
        OnTerminalMsg( IParam );
        break;
    default :
        break;
}
}
```

```

private void OnMsgReceived( IntPtr IpParam, int IMsgId)
{
    //DisableAutoReply로 설정한 Stream,Function 메시지가 들어오는 곳
    ///////////////////////////////////////////////////
    short nStream=0, nFunction = 0, nWbit = 0;
    int nLength = 0;
    m_gem.GetMsgInfo( IMsgId, ref nStream, ref nFunction, ref nWbit,
ref nLength);
    //들어온 메시지의 Stream, Function , Wbit ,Length 값을 받아서
처리함.
    if( m_nControlState == ControlValue.CONTROL_OFFLINE ) {
        m_gem.AbortMsg( IMsgId);
        // OFFLINE인 경우 Sx,F0로 Abort
    }
    else
    {
        If( nStream == 7 && nFunction == 1 )
            OnS7F1( IMsgId ); // 해당 Stream,Function에서 처리.
        ...
    }
}

```

(2) MDLN/SOFTREV 값 세팅

void SetModelName(string strModelName)

장비의 MDLN을 세팅함.

void SetSoftRev(string strSoftRev)

장비의 SoftwareRevision값을 세팅함.

Ex)

```
m_gem.SetModelName("NVIASOFT");  
m_gem.SetSoftRev("1.1.10");
```

S1F13W

<L[2/1]

<A[8/1] 'NVIASOFT'> /* MDLN */

<A[6/1] '1.1.10'> /* SOFTREV */

>

short GetRuntimeState()

드라이버 구동시 라이선스 유무를 반환함.

라이선스 유무는 함수의 리턴값이 1인 경우 인식, 0인 경우 미인식

다른 라이선스 확인 방법은 로그상에 PRODUCT TYPE을 통해서 판단.

2021-07-14 14:53:30 , [EZNET:EVENT] PRODUCT TYPE : PURCHASED <<라이선스 동글 인식

2021-07-14 14:55:18 , [EZNET:EVENT] PRODUCT TYPE : EVALUATION <<라이선스 동글 미인식

(3) SVID관련 등록 및 값 세팅

※모든 ID는 외부파일이나 DB로 관리하여 이것을 한꺼번에 읽어들이는 방법을 권장합니다.

< 1 > SVID (Status Variable ID) 연결

int AddSVID(int SVID, string strName, string strFormat, string strUnit);

해당 SVID를 메모리에 등록

```
ex) m_gem.AddSVID(2001, "ControlState", "U1", "");
m_gem.AddSVID(2005, "CarrierID", "A", "");
m_gem.AddSVID(2010, "Yield", "F4", "%");
```

DATA 형식에 맞게 FORMAT을 지정합니다.

일반적인 경우 : "A","B","BOOL","I1","I2","I4","I8","U1","U2","U4","U8","F4","F8"
(SEMI표준의 FORMAT CODE를 문자열로 입력)

ARRAY인 경우 : "B","BOOL","I1","I2","I4","I8","U1","U2","U4","U8","F4","F8"

의 경우 ARRAY를 붙여서 ARRAY형으로 SVID를 구현 할 수 있다.

```
ex) m_gem.AddSVID(3001, "TEST_I2_ARRAY", "I2_ARRAY", "");
m_gem.SetSVIDValue( 3001 , "1,2,3,4,5");//자동으로 I2의 값을 세팅함.
```

```
<S1F3W
<L[1/1]
<U4[1/1] 3001> /* SVID { TEST_I2_ARRAY } */ /* SVID */
>
>
<S1F4
<L[1/1]
<I2[5/1] 1,2,3,4,5> /* SV 총 5개의 아이템이 추가됨*/
>
>
```

Predefined List와 사용자 직접 구성방식이 있습니다.

“FILE” 의 경우 File을 읽어서 파일의 내용을 SVID Binary[n]의 형태로 저장.

```
m_gem.SetSVIDValue(SVID_FILE_BODY, strFile); // FILE 명이 들어간다.
```

< 1 > Special Case : PREDEFIEND LIST FORMAT (“L”)

알람과 이벤트의 경우 호스트에서 Enable/Disable처리를 하면 그 상태를 GEM Driver가 간직하고 있고 그 Status를 SVID의 LIST로 표현할 수 있습니다.

```
m_gem.AddSVID(2100,"Alarm Enabled","L","");
m_gem.AddSVID(2101,"Alarm Disabled","L","");
m_gem.AddSVID(2102,"Alarm Not Cleared","L","");
m_gem.AddSVID(2103,"Event Enabled","L","");
m_gem.AddSVID(2104,"Event Disabled","L","");
```

SVNAME에 "ALARM" + "ENABLE","DISABLE","NOT CLEAR" 등의 문자열을 조합하여 입력하면 그 입력값(이름)에 따라 GEM Driver가 자동적으로 리스트 처리를 합니다.

이벤트 (CEID)도 마찬가지로 "EVENT" + "ENABLE","DISABLE"등의 문자열을 조합하면 됩니다.

이 경우 Message 내용은 아래처럼 구성이 됩니다.

```
<L
      <U4 4001>      /* ALID */
      :
      <U4 4010>      /* ALID */
>
<L
      <U4 1001>      /* CEID */
      :
      <U4 1008>      /* CEID */
>
```

< 2 > Special Case : USER DEFINE MESSAGE FORMAT ("OBJECT")

이는 EZGEM Driver의 막강한 기능으로 형식이나 구조에 얽매이지 않고 사용자

가

원하는 형식이나 구조로 표현 할 수 있는 기능입니다.

한 개의 SVID항목을 SECS Driver의 SECS Message처럼 처리합니다.

⇒ 최초선언시

```
m_gem.AddSVID(2105, "LoadPortStatus", "OBJECT"); //Message 직접
구성
```

⇒ 내용구성(Update)시

```
// SECS Message구성방법은 SECS Driver를 참고하시기 바랍니다.
```

```
// 모든 Message ID 대신 해당 SVID를 입력합니다.
```

```
m_gem.CloseMsg(2105); // 메시지종료 → 기존 데이터를 삭제
```

```
// (주의)새로 update시 반드시 CloseMsg처리해
```

야함.

```
m_gem.OpenListItem(2105); // <L[3]
    m_gem.AddAsciiItem(2105, strLoadPortID, 1);
    m_gem.AddAsciiItem(2105, strCarrierID, 20);
m_gem.OpenListItem(2105); // <L[25]
    for (int i=0;i<25;i++) {
        m_gem.AddU1Item(2105, m_nSlotMap[i], 1);
    }
m_gem.CloseListItem(2105);
m_gem.CloseListItem(2105);
```

이 경우 실제 Message교환시 아래처럼 나타납니다.

```
S1F3W // HOST→EQ
<L
    <U4 2105> /* SVID */
>
S1F4 // HOST←EQ
<L[1]
    <L[3]
        <A[1] "1"> /* LOAD PORT ID */
        <A[20] "CID_XXX "> /* CARRIERID */
        <L[25]
            <U1 3> /* SLOTMAP [1] */
            <U1 1> /* SLOTMAP [2] */
            :
            <U1 3> /* SLOTMAP [25] */
        >
    >
>
```

< 3 > Special Case : Array형을 사용하는 경우

SVID에 Array형을 추가하는 할 수 있는 부분입니다.

기본형 BOOL,B,U1,U2,U4,U8,I1,I2,I4,I8,F4,F8 가 적용 가능합니다.

FORMAT의 "기본형 ARRAY", "기본형_ARRAY", "기본형-ARRAY"로 선언하여

사용가능.

최초선언시

```
m_gem.AddSVID(3001, "U2 ARRAY", "U2 ARRAY" , "");
//Message 직접구성
```

⇒ 내용구성(Update)시

```
m_gem. SetSVIDValue (3001, ""1,2,3,4,5"" );  
S1F3W // HOST→EQ  
<L  
    <U4 3001> /* SVID */  
>  
S1F4 // HOST←EQ  
<L[1]  
    <U2[5] 1 2 3 4 5> /* U2 ARRAY */  
>
```

[2] CEID (Collective Event ID) 연결

```
int AddCEID(int ID,string NAME,string COMMENT)
```

```
ex) m_gem.AddCEID(1001, "OnlineRemote","");
```

```
ex) m_gem.AddCEID(1002, "CarrierArrived","");
```

[3] ECID (Equipment Constant ID) 연결

```
int AddECID(int ID,string NAME,string UNIT,string FORMAT)
```

```
ex) m_gem.AddECID(3001, "DefaultControlState","","U1");
```

```
ex) m_gem.AddECID(3101, "Feeding Speed","mm/sec", "F4");
```

[4] ALID (Alarm ID) 연결

```
int AddALID(int ID,string ALTX,string ALCD)
```

```
ex) m_gem.AddAlarmID(4001, "EMO","");
```

(3) RemoteCommand(RCMD) 등록

int AddRemoteCommand(string RCMD)

→ Host Command(S2F41)에서 사용할 Command를 등록하는 함수입니다. 미리 선언해놓지 않으면 GEM Driver가 HOST로부터 명령을 받았을 때 NAK 처리합니다.

```
ex) m_gem.AddRemoteCommand("PPSELECT");  
     m_gem.AddRemoteCommand("START");
```

(4) Spool 관련 항목 세팅

void EnableSpooling()

→ 스펴기능을 사용할 때 사용하는 함수. GEM Driver는 Default로 Enable상태입니다.

void DisableSpooling()

→ 스펴기능을 사용하지 않을 때 사용하는 함수입니다. 호스트에서 S2F43을 통해 Spool Stream/Function을 지정해도 SpoolingDisabled상태라면 스펴처리되지 않습니다.

void SetMaxSpoolCount(short nMaxSpoolCount)

→ 스펴의 최대 크기를 지정합니다. 따로 지정하지 않으면 GEM Driver에서 지정된 최대 크기값을 이용합니다.

void SetMaxSpoolTransmit(short nMaxTransmit)

→ 스펴된 메시지를 호스트로 전송할 때 한꺼번에 전송할 스펴의 개수입니다. Default로는 0(Zero)으로 지정이 되어 스펴된 모든 메시지를 보냅니다.

void EnableSpoolOverwrite()

→ 스펴된 메시지가 최대 크기를 초과한 경우 새로 스펴된 메시지를 OverWrite하고자 하는 경우에 사용하는 함수입니다.

void DisablePoolOverwrite()

→ 스펴된 메시지가 최대 크기를 초과한 경우 새로 스펴된 메시지를 OverWrite 하지않고

곧바로 삭제하고자 하는 경우 사용하는 함수입니다.

(5) 사용자 정의 메시지(User Message define) 정의

int DisableAutoReply(short STREAM, short FUNCTION)

→ GEM Driver가 SECS Message를 수신했을 때 그것을 Driver가 처리하지 않고

사용자(개발자)가 직접처리하고자 하는 경우, 이 함수를 사용하여 GEM Driver에게

미리 통보를 해 놓습니다.

그러면 GEM Driver는 해당 Stream,Function의 메시지를 수신하면 아무것도 처리하지

않고 **OnMsgRequested**(long msgid) 이벤트를 통해 사용자에게 넘깁니다.

또한 GEM Driver가 인식하지 못하는 새로운 Message (ex, S3F17, S2F103...)도

송수신하고자 할 때 이 함수를 사용합니다.

만약 S3F17을 이 함수를 이용하여 선언해 놓지 않은 상황에서 GEM Driver가 S3F17을 받게 되면 S9F3 (Unrecognized Stream) 혹은 S9F5 (Unrecognized Function)을 내보내게 됩니다.

SVID,CEID,ECID,ALID,TRACEID,RPTID,DATAID등의 FormatCode를 (필요시) 변경합니다.

EZGEM Driver는 위의 모든 Format을 U4로 정의되어 있고

수신의 경우 DATA TYPE에 상관없이 수신이 가능합니다.

(FORMAT CHECK기능은 별도)

송신의 경우 U4로 전송되게끔 되어 있으나 사용자가 해당 스펴에 맞게 조정할 수 있습니다. "I1","I2","I4","U1","U2","U4","B","BOOL"

SetFormatCode("SVID" , "U2");

```

SetFormatCode( "ECID" , "U2" );
SetFormatCode( "ALID" , "U2" );
SetFormatCode( "CEID" , "U2" );
SetFormatCode( "TRID" , "U2" );
SetFormatCode( "RPTID" , "U2" );
SetFormatCode( "DATAID" , "U2" );

```

(6) 사용자 정의 메시지(User Message define) 처리

int CreateMsg(short STREAM, short FUNCTION)

→ GEM Driver가 전송할 SECS Message를 생성.

메모리에 해당 Stream,Function, Wbit를 가지는 메시지를 생성합니다.

```

int IMsgId = m_gem.CreateMsg(14, 1, 1); // 전송할 해당 메시지를 생성
m_gem.OpenListItem(IMsgId);
m_gem.CloseListItem(IMsgId);
m_gem.SendMsg(IMsgId);                // 해당 메시지를 호스트로 전송

```

int SendMsg(int IMsgId)

→ GEM Driver가 SECS Message를 전송.

```

int IMsgId = m_gem.CreateMsg(14, 1, 1); // 전송할 해당 메시지를 생성
m_gem.OpenListItem(IMsgId);
m_gem.CloseListItem(IMsgId);
m_gem.SendMsg(IMsgId);                // 해당 메시지를 호스트로 전송

```

(6) LOG파일 관련 함수

void EnableLog() / void DisableLog()	
Function	<p>로그를 적용하거나 생략합니다.</p> <p>환경파일(Configuration File)의 LOG항목과 동일하기 때문에, 환경파일을 적용하면 이 Method는 불필요합니다.</p> <p>초기값은 LOG=TRUE이기 때문에 로그를 적용하게끔 되어 있습니다.</p>
void SetLogFile(string strFileNam)	
Function	<p>환경파일을 사용하지 않고 직접적으로 로그파일을 설정할 때 사용합니다.</p> <p>로그파일을 설정하지 않으면 default log파일(ezGem.log)에 기록됩니다.</p> <p>참고로, 로그는 날짜별로 Backup됩니다.</p> <p>※파일명에 “_”를 넣지 않습니다. 넣을 경우 백업에 문제가 생길 수 있습니다.</p>
Parameters	<p>string strFileName :</p> <p>→ 로그파일 경로 및 파일명, 파일명 상위 폴더에 현재 날짜로 폴더생성. (파일이 존재하면 이어서 작성을 하고 없으면 생성합니다)</p>
Example	<p>m_secs.SetLogFile("LOG\\GEMLOG.log", TRUE);</p> <p>일 경우 LOG\\20171012\\ GEMLOG_20171012_[0001].LOG로 생성</p> <p>20171012는 로그 생성일(시스템의 today)</p>
void SetLogRetention(short nDays)	
Function	날짜별로 백업된 로그파일을 몇일간 보유할 것인지를 설정합니다.
Parameters	<p>Short nDays :</p> <p>→ 로그파일 보유일</p>
int GetLogRetention()	
Function	로그파일을 보유기한 값을 읽어옵니다.
Return Value	로그파일 보유일
void SkipLog(short nStream , short Function) void SkipLogReset(short nStream , short Function)	
Function	<p>특정 Stream,Function 메시지를 로그파일을 바이너리로만 남기며, 텍스트 형식으로는 남기지 않습니다. 따라서, 로그의 파일 크기를 크게 줄일 수 있습니다.</p>

4. GEM Driver 구동 및 정지

int Start(),int Stop()

- 성공시 0 (Zero)을 반환
- 실패시 음수(Error Code)를 반환

```
ex) if (m_gem.Start() < 0) {  
        AfxMessageBox("구동실패");  
        return;  
    }
```

Short GetRuntimeState()

- 최근의 라이선스 체크시 라이선스가 체크되었는지를 반환.
라이선스 체크는 Start함수 호출시, 일정시간이 경과 후 지속적으로 확인.
- 0 : 라이선스가 확인되지 않았음.
- >1 : 라이선스가 확인되었음.

```
ex) short nState = m_gem.GetRuntimeState();
```

void GoOnlineRemote() / OnEZGemEvent(nEventId = 1003)

- Control State를 ONLINE_REMOTE로 변경합니다.
GEM 드라이버가 구동만 되고 아직 HOST Connection이 이루어지지 않은상태에서 사용을 해도, 나중에 Connection→EstablishComm이후에 자동으로 Remote 상태로 전환이 되며 이때 사용자에게 이벤트(OnOnlineRemote())를 발생시킵니다.

이 함수를 호출하지 않으면, GEM Driver는 OFFLINE상태로 있으며, OFFLINE상태에서 HOST가 ONLINE_REQUEST(S1F17)을 보내면 ONLINE_REMOTE로 변경이 됩니다.

- Control State를 ONLINE_LOCAL로 변경합니다.
- Control State를 ONLINE LOCAL로 변경합니다.

void GoOnlineRemote() / OnEZGemEvent(nEventId = 1003)

void GoOffline()/ OnEZGemEvent(nEventId = 1001)

→ Control State를 OFFLINE으로 변경합니다.

5. GEM Driver Event 발생에 따른 처리

EVENT : OnConnected/ OnEZGemEvent(nEventId = 1)
EVENT : OnDisconnected/ OnEZGemEvent(nEventId = 2)

→EZNET SECS Driver와 마찬가지로 HOST가 연결되었을 때 혹은 끊어졌을 때 발생하는 이벤트입니다.

EVENT : OnCommEstablished/ OnEZGemEvent(nEventId = 1010)

→GEM Driver가 구동이 되면 HOST가 접속하게 됩니다.

HOST가 접속후 S1F13(CommRequest)를 보내오면 그 응답을 내보낸후 이 이벤트가 발생합니다.

HOST가 S1F13을 보내오지 않으면 EQ(GEM Driver)가 일정시간 (CR Timer : 기본 5 초) 기다린후

EQ에서 HOST로 S1F13을 보내고 그 응답을 받으면 이 이벤트가 발생합니다.

이 이벤트가 발생되면 CommState는 "COMMUNICATING" 상태로 됩니다.

EVENT : OnOnlineRemote/ OnEZGemEvent(nEventId = 1030)

→EQ의 ControlState가 ONLINE_REMOTE로 변경되었을 때 발생하는 이벤트입니다.

GEM Driver에서 GoOnlineRemote()함수를 이용하여 설비측에서 OnlineRemote로 변경하거나

HOST에서 S1F17(OnlineRequest)요청하여 그것을 Accept한 경우 이 이벤트가 발생합니다.

이 때 필요한 SVID, CEID 등을 처리해야 합니다.

int SetSVIDValue(int SVID, string strNewValue)

→SVID의 값을 지정할 때 사용하는 함수입니다. 값(Value)은 문자열로 지정합니다.

CTestDlg::OnOnlineRemote()

{

 m_strControlState = "ONLINE_REMOTE";

 m_gem. SetSVIDValue (2001, "3");

 m_gem.SendEventReport(1003); // 해당 Remote Event 보고

}

int GetSVIDValue(long SVID, ref string strNewValue)

→SVID의 값을 가져올 때 사용하는 함수입니다.

```
m_gem.SetSVIDValue(SVID.LOT_ID, "LOT_01");  
string strLogId = "";  
m_gem.GetSVIDValue(SVID.LOT_ID, ref strLogId);  
MessageBox.Show(strLogId); // strLotID = "LOT_01";
```

int SetECValue(long SVID, ref string strNewValue)

→ECID의 값을 지정할 때 사용하는 함수입니다. 값(Value)은 문자열로 지정합니다.

```
m_gem.SetECIDValue(SVID.EC_10, "30");  
string strTimer = "";  
m_gem.GetECValue(SVID.EC_10, ref strTimer);  
MessageBox.Show(strTimer); // strTimer = "30";
```

int SetECRange(long ECID, string strMinValue, string strMaxValue)

→ECID의 값을 최소와 최대값을 지정할 때 사용하는 함수입니다

```
m_gem.SetECRange(SVID.EC_10, "1" , "30");  
인 경우, 1~30사이의 값만을 저장할 수 있음.
```

int SetECValue(long SVID, ref string strNewValue)

→ECID의 값을 지정할 때 사용하는 함수입니다. 값(Value)은 문자열로 지정합니다.

```
m_gem.SetECIDValue(SVID.EC_10, "30");  
string strTimer = "";  
m_gem.GetECValue(SVID.EC_10, ref strTimer);  
MessageBox.Show(strTimer); // strTimer = "30";
```

int SendEventReport(CEID)

이때 REPORT가 연결되어 있으면 자동으로 연결되어 전송이 됩니다.

또한 HOST에서 S2F37로 해당 이벤트를 Disable상태로 세팅해두었다면 이 함수를 호

출해도

전송이 되지 않습니다.

```
CTestDlg::OnOnlineRemote()  
{  
    m_strControlState = "ONLINE_REMOTE";  
  
    m_gem.SetSVIDValue(2001, "3");  
    m_gem.SendEventReport(1003);  
}
```

- 주의

SendEventReport 함수를 이용하기 전에 **SetSVIDValue** 함수를 먼저 호출해야 합니다.

왜냐하면 **SendEventReport(S6F11)** 메시지가 전송될 때 **REPORT**가 연결되어 있고

그 **REPORT**내의 **SVID**값이 미리 반영되어 있어야 하기 때문입니다.

EVENT : OnOnlineLocal/ OnEZGemEvent(nEventId = 1002)

→EQ의 ControlState가 ONLINE_REMOTE로 변경되었을 때 발생하는 이벤트입니다.

```
CTestDlg::OnOnlineLocal()
{
    m_strControlState = "ONLINE_LOCAL";
    m_gem.SetSVIDValue(2001, "2");
    m_gem.SendEventReport(1002);
}
```

**EVENT : OnNewECValue (int MSGID)
/ OnEZGemEvent(nEventId = 1015)**

```
int GetHostSetECID(int MSGID, ref int nECID,ref string strECValue)
int ReplyHostSetECID(int MSGID,short nReturnCode)
```

→호스트로부터 S2F15 ECV 변경 요청을 받은경우 발생하는 이벤트입니다.

이때 한 개 이상의 ECV변경 요청이 있을 수 있으므로 그 Parameter로 그 개수가 넘어옵니다.

OnNewECValue(int IMsgId)

```
{
    int nReturnCode = 0x00;
    short nECCount = 0; int nECID = 0; string strNewValue = "";
    While( nECCount > -1 )
    {
        nECCount = m_gem.GetHostSetECID(IMsgId, ref nECID , ref strNewValue );
        //HOST에서 전송한 ECID, ECValue를 변수에 저장함.
        m_gem.SetECValue( nECID , strNewValue );
        //변수에 저장된 값을 ECID와 strNewValue에 세팅.
    }

    // HOST가 보내온 ECID와 그 값에 문제가 있는지 확인합니다.
    // 위의 값들은 응답메세지를 내보낸 이후에는 소실되므로 필요시
    // 별도의 변수에다 저장을 해두어야 합니다.
    // 이상 유무에 따라 nReturnCode값을 다르게 하여 응답 메시지를 내보낸다.
    m_gem.ReplyHostSetECID(IMsgId,nReturnCode);
}
```

**EVENT : OnRemoteCommand(int MSGID)
/ OnEZGemEvent(nEventId = 1030)**

**int GetRemoteCommand(int MSGID , ref string strCommand);
String GetRemoteCommandParam
(int MSGID,int INDEX,ref string CPNAME, ref string CPVALUE)
Short ReplyRemoteCommand(int MSGID, string RCMD)**

→HOST로부터 S2F41을 받은 경우 발생하는 이벤트입니다.

Command수신시 RCMD(Command Name), Command Parameter Name(CPNAME),
Command Parameter Value(CPVAL)등을 여러 개의 조합으로 받을 수 있으므로
그 개수만큼 각 Parameter Name,Value를 확인하여 조치를 취하면 되며 사용법은 아
래와
같습니다.

OnRemoteCommand(int IMsgId)

```
{  
    // CPNAME, CPVAL을 조회, 확인합니다.  
    String strCommand = "";  
    Short nParamCount = m_gem.GetRemoteCommand( IMsgId , ref strCommand );  
    // RCMD의 NAME과 CPNAME, CPVALUE의 Count가 반환됨.  
    for (int i=0;i<nParamCount;i++) {  
        CString strParmName = "", strParmValue = "";  
        m_gem.GetRemoteCommandParam(IMsgId, i , ref strCPName , ref strCPValue );  
        // 해당 위치의 CPName과 CPValue를 가져옴.  
    }  
  
    // 파라미터 VALIDATION해서 그에 따른 응답을 전송합니다.  
    m_gem.ReplyRemoteCommand( IMsgId , strCommand , NACK );  
    // HBACK의 값에 따른 의미.  
    // 0 : Acknowledge, command has been performed. ( ACK )  
    // 1 : Command does not exist ( AddRemoteCommand 에 추가 안 된 경우 자동 전송 )  
    // 2 : Cannot perform now (NACK )  
    // 3 : At least one parameter is invalid ( NACK )  
    // 4 : Acknowledge, command will be performed with completion signaled later by an  
event( ACK )  
    // 5 : Rejected, Already in Desired condition ( NACK )
```

}

EVENT : OnTermMsg(MSGID)

/ OnEZGemEvent(nEventId = 1050)

⇒ int GetTerminalMsg(int MSGID,ref int nTid,ref string strMessage);

→HOST로부터 터미널메세지 (S10F3,S10F5)을 받은 경우 발생하는 이벤트입니다.

Void **OnTermMsg**(int IMsgId)

```
{
    // 필요시 GUI에 Display합니다.
    INT nReturnCode = 0x00;
    Int nCount = 0;
    While( nCount > -1 )
    {
        nCount = m_gem.GetTerminalMsg( IMsgId , ref nTid , ref strMessage );
        //받을 메시지가 없는 경우 -1로 리턴함.
        m_listbox.AddString(strMsg);
        if( nCount < 0 ) break;
    } // 자동으로 응답을 처리함.
}
```

EVENT : OnSecsMsgIn(int IParam)

/ OnEZGemEvent(nEventId = 401)

EVENT : OnSecsMsgOut(int IParam)

/ OnEZGemEvent(nEventId = 402)

→모든 SECS Message 송수신에 대해 발생하는 이벤트입니다.

이는 GUI상의 간단한 로그를 표시하기에 큰 도움이 됩니다.

Void OnMsgIn(int IParam)

```
{
    int nStream = 0, int nFunction = 0;
    nStream = (int)(IParam / 1000 );
    nFunction = IParam % 1000;
    AddLog( string.Format("(H->E) S{0},F{1}", nStream, nFunction ));
    // (H->E) S7,F19
}
```

6. 각 기능별 함수 및 이벤트 정리

[1] GEM Driver 구동 및 정지

- * int **Start()**
→ EZGEM Driver 구동
- * short **GetRuntimeState ()**
→ EZGEM Driver의 라이선스가 확인 되었는지 반환
- *int **Stop()**
→ EZGEM Driver 정지

[2] SVID관련

- * int **AddSVID**(int SVID, string strName, string strDataType, string strUnit)
→ SVID등록
- * int **SetFormatCode**("SVID" , string strFormatCode)
→포맷설정(SECS Message SVID 항목)
- * int **SetSVIDValue**(int SVID, string strValue)
→ SVID 값 변경
- *int **GetSVIDValue**(int SVID, ref string strValue)
→ SVID 값 가져옴.

[3] 알람(ALID) 관련

- * int **AddALID**(int ALID, string strName, string strALCD)
→ALID등록
- *int **SetFormatCode**("ALID" , string strFormatCode)
→포맷설정(SECS Message SVID항목)
- * int **SendAlarmReport**(int ALID, short ALCD)
→알람리포트(S5F1)전송

[4] 이벤트(CEID) 관련

- * int **AddCEID**(int CEID, string strName, string strComment)
→CEID등록
- * short **SetFormatCode**("CEID" , string strFormatCode)
→포맷설정(SECS Message CEID항목)
- * short **SendEventReport**(int CEID)
→이벤트보고(S6F11)메시지 전송

[5] ECID 관련

- * int **AddECID**(int ECID, string strName, string strUnit, string strDataFormat)
→ECID등록
- * short **SetFormatCode**("CEID" , string strFormatCode)
→포맷설정(SECS Message ECID항목)
- * short **SetECRange**(int ECID, string strMinValue, string strMaxValue)
→구간 (Range) 설정
- * short **SetECValue**(int ECID, string strCurrentValue)
→현재 값 설정
- * short **GetECValue**(int ECID, string strCurrentValue)
→현재 값 설정
- * EVENT OnEventReceived(**IntPtr** lpParam, **short** nEventId, **int** lParam)
→ nEventId = 1015 호스트로부터 S2F15를 받음

[6] Control State관련

- * void **GoOnlineRemote()**
→Control State를 Online_Remote로 전환
- * void **GoOnlineLocal()**
→Control State를 Online_Local로 전환
- * void **GoOffline()**
→Control State를 Offline으로 전환
- * EVENT OnEventReceived(IntPtr lpParam, short nEventId, int lParam)
→ nEventId = 1001, ControlState가 Offline 로 변경됨
- * EVENT OnEventReceived(IntPtr lpParam, short nEventId, int lParam)
→ nEventId = 1002, ControlState가 Online_Local로 변경됨
- * EVENT OnEventReceived(IntPtr lpParam, short nEventId, int lParam)
→ nEventId = 1003, ControlState가 Online_Remote로 변경됨

[7] Communication관련

- * short **CommmRequest**
→Connection후 S1F13(EstablishCommunication)진행까지의 대기시간 설정
m_gem.CommRequest = (short)10;
// Connection후 10초 후에 Establish가 안 된 경우, 장비에서 S1F13을 보냄.

[8] SPOOLING관련

- * void **EnableSpooling()**
→Spooling가능하게끔 설정
- * void **DisableSpooling()**
→Spooling불가능하게끔 설정
- * CString **GetSpoolTime**(ref string strStartTime , ref string strFullTime)
→Spooling시작시간과 풀타임을 변수로 전달해줌.
- * void **SetMaxSpoolCount**(uint nMaxSpoolCount)
→스풀의 최대 크기 설정
- * int **GetMaxSpoolCount**()
→스풀의 최대 크기 확인

- * void **EnableSpoolOverWrite()**
→스풀덮어쓰기 설정
- * void **DisableSpoolOverWrite()**
→스풀덮어쓰기 불가능하게 설정
- * short **SetMaxSpoolTransmitCount()**
→스풀전송크기 설정
- * short **GetMaxSpoolTransmitCount()**
→스풀전송크기 확인

[9] Remote Command관련

- * int **AddRemoteCommand**(string strRemoteCommand)
→RemoteCommand등록
- * EVENT OnEventReceived(**IntPtr** lpParam, **short** nEventId, **int** lParam)
→ nEventId = 1030, S2F41이 들어옴.
- * short **GetRemoteCommand**(int lMsgId, ref string strRCMD)
→S2F41에 들어온 RCMD를 리턴함. "START", "PPSELECT" ...
- * short **GetRemoteCommandParam**(int lMsgId, ref string strCPName,
ref string CPValue , ref short ValueType)
→S2F41에 들어온 CPName, CPValue , CPValue's Type을 리턴함.
- * CString **GetRemoteCommandParamValue**(long lMsgId, LPCTSTR strCommand, short
nIndex)
→CPVAL(Command Parameter Value) 확인 조회
- * short **ReplyRemoteCommand**(long lMsgId, LPCTSTR strCommand, short ACK)
→RemoteCommand의 응답 메시지(S2F42)를 내보냄

[10] 직접 Message구성 관련 함수

* `int CEZGemPlusLib.AddArrayItem(int lMsgId, string strDataType, int nDataCount, string strData)`

→생성한 메시지에 아이템을 추가함.

[11] Terminal Message관련

* EVENT **OnTermMsg**(long lMsgId, LPCTSTR strMsg, short TID)

→HOST로부터 S10F3(터미널메세지)를 받음

* short **ReplyTerminalMessage**(long lMsgId,INT nReturnCode)

→수신메세지를 허용하는 응답을 내보냄

* EVENT **OnTermMsgMulti**(long lMsgId, short TID, short nCount)

→HOST로부터 S10F5(터미널메세지)를 받음

* CString **GetTermMsgMulti**(long lMsgId, short nIndex)

→수신한 터미널메세지를 확인 및 조회

[12] LOG파일 관련

* void **EnableLog()/DisableLog()**

→선언한 로그파일에 메시지 로그를 남길지/남기지 않을 지를 선택

* void **SetLogFile**(LPCTSTR strFileName , short bLogSecsII);

→남겨질 로그 파일의 이름선언 및 로그를 SECS-II 메시지도 남길지를 선언.

Ex) **SetLogFile("LOG\\GEM.LOG" , true);** 인 경우

LOG\\20171012\\GEM_20171012_[0001].LOG 파일에 로그를 남긴다.

* void **SetLogRetention**(short nDays)

→날짜별로 백업된 로그파일을 몇일간 보유할 것인지를 설정함

* void **GetLogRetention**(short nDays)

→로그파일의 보유기한 값을 읽어옴

* void **SkipLog**(short nStream, short nFunction)

→특정 메시지에 대하여 SECS-II메시지를 로그에 남기지 않음.

[13] EZGEM.RPT파일 관련

* void SetReportFilePath(LPCTSTR strPath)

→S2F33, S2F35를 통해 정의된 DYNAMIC LINK DEFINE을 저장하는

REPORT파일을 저장하는 파일을 설정한다.

프로그램 시작시 세팅된 Path의 파일을 읽어서 이전의 DYNAMIC LINK DEFINE을 불러 온다.

[14] Transaction Msg관련

* int SendEventReportEx(int nCeid , int nTransactionID);

→S6F11W Event를 발생시 해당 SystemByte를 TransactionID에 연결한다.

→추후 int GetSysByteEx(int nTransactionID);를 통해서 리턴값으로 보낸 S6F11메시지의

SystemByte를 받을 수 있다.

* int SendMsgEx(int lMsgId , int nTransactionID);

→해당 메시지를 전송시 SystemByte를 TransactionID에 연결한다.

→추후 int GetSysByteEx(int nTransactionID);를 통해서 리턴값으로 보낸 메시지의

SystemByte를 받을 수 있다.

* int GetSysByteEd(int nTransactionID);

→SendEventReportEx, SendMsgEx를 통해서 연결해 둔 TransactionID의 SystemByte를

받아온다.

[15] GEM에서 제공하는 EVENT (Event ID의 역할)

Page 6의 전체사용 가능 내부 이벤트 목록

Event ID	내용	설명
1	Connect	통신이 연결됨.
2	Disconnect	통신 연결이 끊어짐.
10	Evaluation mode start	개발 모드로 시작함 (Lock Key 없음)
11	Runtime mode start	정식 모드로 시작함 (Lock Key 있음)
101	Linktest request in	Linktest요청 수신
102	Linktest request out	Linktest요청 송신
103	Linktest response in	Linktest응답 수신
104	Linktest response out	Linktest응답 송신
202	Transaction timeout	S9F9 발생
203	Unrecognized DeviceID	S9F1 발생
204	Unrecognized Stream	S9F3 발생
205	Unrecognized Function	S9F5 발생
206	Invalid Data	S9F7 발생
207	Discard Msg	S9F9 발생. 처리되지 않은 메시지가 폐기 처리됨
208	Abort Msg In	수신한 메시지가 Abort 처리됨
301	Timeout T1	T1 발생
302	Timeout T2	T2 발생
303	Timeout T3	T3 발생
304	Timeout T4	T4 발생
305	Timeout T5	T5 발생
306	Timeout T6	T6 발생
307	Timeout T7	T7 발생
308	Timeout T8	T8 발생
309	Retry Limit	Retry Limit 발생
401	Msg In	메시지 수신 (Stream , Function)
402	Msg Out	메시지 송신 (Stream , Function)
1001	Offline	Offline으로 Control이 변경됨.
1002	Local	Online Local로 변경됨.
1003	Remote	Online Remote로 변경됨.
1010	Communicating	Communication상태가 됨.
1015	Host ECID	Host ECID(S2F15)가 들어옴.

1020	Trace Time Set	호스트가 Trace 를 설정 함
1030	RemoteCommand	RemoteCommand(S2F41)이 들어옴.
1040	Spool Updated	Spool 업데이트 됨
1041	Spool Full	Spool 이 가득 참
1050	TerminalMessage	Terminal Message(S10F3 / S10F5) 가 들어옴.