

## 최대의 이익을 위한 최선의 선택!

LS 메카피온에서는 저희 제품을 선택하시는 분들께 최대의 이익을 드리기 위하여 항상 최선의 노력을 다하고 있습니다.

# LCOM Library 설명서

Ver1.2



### 안전에 관한 주의사항

- 사용전에 안전을 위한 주의사항을 반드시 읽고 정확하게 사용하여 주십시오.
- 사용설명서를 읽고 난 뒤에는 제품을 사용하는 사람이 항상 볼 수 있는 곳에 잘 보관하십시오.

**LS** 메카피온



# 목차

<b>1.</b>	<b>소개 .....</b>	<b>1-1</b>
<b>2.</b>	<b>User API.....</b>	<b>2-2</b>
2.1	목록.....	2-2
2.2	세부사항.....	2-5
2.2.1	일반 함수.....	2-5
2.2.2	통신 설정 함수.....	2-7
2.2.3	파라미터 R/W 설정 함수.....	2-13
2.2.4	제어 파라미터 설정 함수.....	2-18
2.2.5	모니터링 함수.....	2-23
2.2.6	알람 상태 및 처리 함수.....	2-28
2.2.7	입출력 포트 처리 함수 .....	2-30
2.2.8	모션 동작 함수.....	2-35
<b>3.</b>	<b>Preliminary.....</b>	<b>3-54</b>
3.1	라이브러리 설정 .....	3-54
3.2	통신 설정 .....	3-54



# 1. 소개

LCOM Library 는 L7 Series Drive 제품을 제어하기 위한 라이브러리 입니다. 사용자의 편의와 접근성을 높이기 위해 개발되었습니다. 기존에 Drive\_CM 을 사용하는 함수 형태의 API(Application Program Interface)를 모두 지원합니다. 라이브러리는 Visual C++ 개발환경을 지원합니다. 아래의 파일들로 구성됩니다.

- LCOM.dll
- LCOM.h
- LOCM.lib
- API\_EherCAT.dll
- API\_EtherCAT.lib
- RMC\_EthercatAPI.h
- EoESetup.xml

## 2. User API

### 2.1 목록

API's Name	Description
LCOM_GetLcomDllVersion	라이브러리 버전 표시
LCOM_GetEoESetupFile	EoE연결 파일 반환
LCOM_SetEoESetupFile	EoE연결 파일 설정
Communication	
LCOM_SDRVOpen	RS-485 통신채널 할당, Open작업
LCOM_EDRVOpen	EoE 통신채널 할당, Open작업
LCOM_TCPOpen	Ethernet TCP 통신채널 할당, Open작업
LCOM_DriveClose	Resource보호 및 통신 Closing 작업
LCOM_GetTimeout	드라이브 통신 송/수신 타임아웃 반환
LCOM_SetTimeout	드라이브 통신 송/수신 타임아웃 설정
LCOM_GetOnlineMode	Ethernet 네트워크 상태를 반환
LCOM_IsSlaveOnline	개별 Slave의 네트워크 상태를 반환
LCOM_GetSlaveCount	현재 연결된 Slave의 개수를 반환
LCOM_GetErrorCode	통신 에러 코드 반환
LCOM_GetErrorString	통신 에러 내용 반환
LCOM_GetTxPackets	통신 송신 패킷 반환
LCOM_GetRxPackets	통신 수신 패킷 반환
Parameter R/W	
LCOM_ReadByte	[USINT] 파라미터 읽기
LCOM_ReadByteS	[SINT] 파라미터 읽기
LCOM_ReadWord	[UINT] 파라미터 읽기
LCOM_ReadWordS	[INT] 파라미터 읽기
LCOM_ReadDWord	[UDINT] 파라미터 읽기
LCOM_ReadDWordS	[DINT] 파라미터 읽기
LCOM_ReadString	[STRING] 파라미터 읽기
LCOM_WriteByte	[USINT] 파라미터 쓰기
LCOM_WriteByteS	[SINT] 파라미터 쓰기
LCOM_WriteWord	[UINT] 파라미터 쓰기
LCOM_WriteWordS	[INT] 파라미터 쓰기
LCOM_WriteDWord	[UDINT] 파라미터 쓰기

LCOM_WriteDWordS	[DINT] 파라미터 쓰기
LCOM_WriteString	[STRING] 파라미터 쓰기
LCOM_DriveSoftReset	서보드라이브 S/W리셋
<b>Control Parameter</b>	
LCOM_SetControlMode	제어모드 설정
LCOM_SetElecGearRatio	기어비 설정
LCOM_SetCurGain	전류게인 설정
LCOM_SetInertiaRatio	관성비 설정
LCOM_SetVelGainKp	속도비례 게인 설정
LCOM_SetVelGainKiT	속도적분 시정수 설정
LCOM_SetPosGainKp	위치비례 게인 설정
LCOM_SetPosGainKvff	위치제어 속도 피드포워드 게인 설정
LCOM_SetPosGainKvffflt	위치제어 속도 디드포워드 게인 필터 시정수 설정
LCOM_GetCurGain	전류게인 반환
LCOM_GetInertiaRatio	관성비 반환
LCOM_GetVelGainKp	속도비례 게인 반환
LCOM_GetVelGainKiT	속도적분 시정수 반환
LCOM_GetPosGainKp	위치비례 게인 반환
LCOM_GetPosGainKvff	위치제어 속도 피드포워드 게인 반환
LCOM_GetPosGainKvffflt	위치제어 속도 디드포워드 게인 필터 시정수 반환
<b>Monitor</b>	
LCOM_FeedbackSpeed	피드백 속도 반환
LCOM_ActualPosition	실제 위치값 반환
LCOM_ActualVelocity	실제 속도값 반환
LCOM_ActualTorque	실제 토크값 반환
LCOM_SingleTurnData	1회전 내 데이터 반환
LCOM_MultiTurnData	다회전 데이터 반환
LCOM_WarningCode	경고 코드 반환
LCOM_DCLinkVoltage	DC-Link 전압 반환
LCOM_AccumOverload	누적 운전 과부하율 반환
LCOM_DriveTemp	드라이브 내부 온도 반환
<b>Alarm Handling</b>	
LCOM_IsAlarm	현재 알람 상태 반환
LCOM_ResetAlarm	현재 알람 상태 리셋
LCOM_GetLatestAlarmString	가장 최근 발생된 알람코드 반환
LCOM_GetAlarmHistory	알람 히스토리 반환
LCOM_ClearAlarmHistory	알람 히스토리 삭제

IO Handling	
LCOM_GetDigitalInputs	디지털 입력 신호 설정 값 반환
LCOM_IsNegativeLimit	역방향 리미트 스위치 반환
LCOM_IsPositiveLimit	정방향 리미트 스위치 반환
LCOM_IsHomeSensor	원점 센서 입력 반환
LCOM_GetDigitalOutputs	디지털 출력 신호 반환
LCOM_SetDigitalOutputsWithMask	디지털 출력 신호 설정
LCOM_GetDriveControlInput1	드라이브 제어 입력1 반환
LCOM_GetDriveControlInput2	드라이브 제어 입력2 반환
LCOM_SetDriveControlInput1	드라이브 제어 입력1 설정
LCOM_SetDriveControlInput2	드라이브 제어 입력2 설정
LCOM_GetDriveStateOutput	드라이브 출력 신호 반환
LCOM_TouchProbeFunction	터치 프로브 기능 설정
LCOM_TouchProbeStatus	터치 프로브 상태 반환
LCOM_TouchProbe1PositiveEdgeValue	터치 프로브 1 상승에지 위치값
LCOM_TouchProbe1NegativeEdgeValue	터치 프로브 1 하강에지 위치값
LCOM_TouchProbe2PositiveEdgeValue	터치 프로브 2 상승에지 위치값
LCOM_TouchProbe2NegativeEdgeValue	터치 프로브 2 하강에지 위치값
Motion Function	
LCOM_SetOperationMode	운전모드 설정
LCOM_Homing	서보축의 원점복귀 동작 명령
LCOM_IsHome	원점복귀 완료 신호
LCOM_HomeTorqueLimit	Stopper 이용 원점 복귀 시 토크 제한 값 설정
LCOM_Enable	Servo On상태 제어
LCOM_Disable	Servo Off상태 제어
LCOM_IsEnabled	Servo On/Off상태 확인
LCOM_EmergencyStop	비상정지 신호 설정
LCOM_Stop	운전 정지 신호 설정
LCOM_Start	운전 시작 신호 설정
LCOM_SetCurrentPosition	현재 위치를 원점으로 설정
LCOM_MoveAbsolute	절대치 이동 설정
LCOM_MoveAbs	절대치 이동 시 목표 위치만 입력
LCOM_MoveRelative	상대치 이동 설정
LCOM_MoveRel	상대치 이동 시 목표위치만 입력
LCOM_MoveIndex	해당 인덱스 No로 운전 실시
LCOM_ReadIndexParameter	인덱스 파라미터 읽기
LCOM_WriteIndexParameter	인덱스 파라미터 쓰기

LCOM_IsEndIndex	인덱스 종료 후 EOS신호
LCOM_SetInPosition	위치 도달범위 및 도달시간 설정
LCOM_IsInPosition	목표위치 도달 신호 출력
LCOM_TargetVelocity	목표 속도 설정
LCOM_ProfileVelocity	프로파일 속도 설정
LCOM_ProfileAccel	프로파일 가속도 설정
LCOM_ProfileDecel	프로파일 감속도 설정
LCOM_StopDecel	Quick Stop 감속도 설정
LCOM_SetInVelocity	목표 속도 도달 범위 설정
LCOM_IsInVelocity	목표 속도 도달 완료 신호
LCOM_TargetTorque	목표 토크 설정
LCOM_TorqueVelLimit	토크제어시 속도 제한
LCOM_TorqueSlope	토크 기울기 설정
LCOM_IsTorqueLimit	토크출력 제한 신호
LCOM_SelVelocityLimit	속도 제한 기능 설정
LCOM_SetVelocityLimit	토크 제어 시 제한 속도

## 2.2 세부사항

### 2.2.1 일반 함수

<b>DLLFUNC int __stdcall LCOM_GetLcomDllVersion(int* pVersion)</b>		
라이브러리 버전 확인을 위해 사용합니다.		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
Argument		
pVersion		라이브러리에 등록된 버전 정보
Example		
<pre> /*현재 적용 중이 DLL 버전을 정보를 얻어 옵니다*/ // 버전 정보는 32bit로 관리됨. CString str; int ret, pVersion; ret = LCOM_GetLcomDllVersion(&amp;pVersion); str.Format("DLL Version = %d.%d.%d.%d",pVersion &gt;&gt; 24, (pVersion &amp; 0x00FF0000) &gt;&gt; 16 , (pVersion &amp; 0x0000FF00) &gt;&gt; 8, pVersion &amp; 0x000000FF); m_Msg_Version.SetWindowText(str); </pre>		

DLLFUNC int __stdcall LCOM_GetEoESetupFile(char* pFileName)		
현재 사용 하고 있는 EoESetup 설정 파일 경로 위치 확인		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
Argument		
pFileName		EOESetup파일 경로
Example		
<pre> /*EoESetup파일 경로 확인*/ char pFileName[256]; int ret; ret = LCOM_GetEoESetupFile(&amp;pFileName);                     </pre>		

DLLFUNC int __stdcall LCOM_SetEoESetupFile(char* pFileName)		
EoESetup 설정 파일 적용		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
Argument		
pFileName		EOESetup파일 경로
Example		
<pre> /*EoESetup파일 설정*/ int ret; char cEoECfgPath[256] = { 0 }; CString EoESetupFileName; CFileDialog mFileDlg(TRUE, "*.xml", "*.xml", OFN_LONGNAMES   OFN_PATHMUSTEXIST   OFN_HIDEREADONLY,                     "Xml file(*.xml) *.xml ",NULL); if (IDOK != mFileDlg.DoModal()) return;  m_EoESetupFileName = mFileDlg.GetPathName(); //EoESetup파일 경로 저장 wprintf(cEoECfgPath, "%s", m_EoESetupFileName); ret = LCOM_SetEoESetupFile(cEoECfgPath); //지정한 경로의 EoESetup파일을 적용.                     </pre>		

## 2.2.2 통신 설정 함수

DLLFUNC int __stdcall LCOM_SDRVOpen(int DriveId, char *pComPort, int dwBaud)		
RS-485통신을 연결 하기 위해 사용됩니다.		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
LCOM_MEI_SERIAL_OPEN_ERR	12	시리얼 통신 연결 에러
Argument		
DriveId	1~31	Node ID
pComPort	0~100	시리얼포트
dwBaud	9600~57600	Baudrate
Example		
<pre> /*시리얼 통신 연결(2축 기준)*/ int Baudrate,m_DriveID[32]; char cPortName[32]; CString str; //사용자가 설정한 축별 NodeId정보를 읽어옴 m_edit_driveId.GetWindowText(str); m_DriveId[0] = atoi(str); m_edit_driveId2.GetWindowText(str); m_DriveId[1] = atoi(str); //사용자가 설정한 port정보를 읽어옴 m_cb_comport.GetLBText(m_cb_comport.GetCurSel(),str); wsprintf(cPortName,_T("%s"), str); //사용자가 설정한 Baudrate정보를 읽어옴 m_cb_baud.GetLBText(m_cb_baud.GetCurSel(), s); Baudrate = atoi(s); //사용자의 입력 정보를 바탕으로 시리얼통신을 연결함. ret = LCOM_SDRVOpen(m_DriveId[0],cPortName, Baudrate); ret = LCOM_SDRVOpen(m_DriveId[1],cPortName, Baudrate); </pre>		

DLLFUNC int __stdcall LCOM_EDRVOpen(int DriveId, char *IpAddress, int PortNum)		
EtherCAT Of EtherNet 통신을 연결 하기 위해 사용됩니다.		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
LCOM_MEI_ETHERNET_OPEN_ERR	24	시리얼 통신 연결 에러

Argument		
DriveId	1~31	라이브러리에 등록된 버전 정보
IpAddress	0.0.0.0~255.255.255.255	슬레이브의 IP 어드레스
PortNum	1~55	이더넷포트
Example		
<pre> /*시리얼 통신 연결(2축 기준)*/ int Port,m_DriveID[32]; char cPortName[32],cIp[32], cIp2[32]; CString str; //Ethernet통신을 연결 하기 전 EoESetup파일을 적용 한 뒤에 할 것. //어드레스 정보를 가지고 있기 때문 //---Set EoE configuration path char cEoECfgPath[256] = { 0 }; wsprintf(cEoECfgPath, "%s", m_EoESetupFileName); ret = LCOM_SetEoESetupFile(cEoECfgPath);  //사용자가 설정한 축별 NodeId정보를 읽어옴 m_edit_driveId.GetWindowText(str); m_DriveId[0] = atoi(str); m_edit_driveId2.GetWindowText(str); m_DriveId[1] = atoi(str); //Taret IP어드레스 정보를 가져옴 m_edit_targetIp.GetWindowText(s); wsprintf(cIp, "%s", s); m_edit_targetIp2.GetWindowText(s); wsprintf(cIp2, "%s", s);  //Taret IP어드레스 정보를 가져옴 m_edit_targetPort.GetWindowText(s); Port = atoi(s);  ret = LCOM_EDRVOpen(m_DriveId[0], cIp, Port); ret = LCOM_EDRVOpen(m_DriveId[1], cIp2, Port);                     </pre>		

<b>DLLFUNC int __stdcall LCOM_TCPOpen(int DriveId, char *IpAddress, int PortNum)</b>		
Ethernet TCP 통신을 연결 하기 위해 사용합니다.		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
LCOM_MEI_ETHERNET_OPEN_ERR	24	시리얼 통신 연결 에러
Argument		
DriveId	1~31	각 슬레이브의 고유번호 지정(Drive Nodeid와 다름)
IpAddress	0.0.0.0~255.255.255.255	슬레이브의 IP 어드레스
PortNum		이더넷포트
Example		
<pre> /*시리얼 통신 연결(2축 기준)*/ int Port,m_DriveID[32]; char cPortName[32],clp[32], clp2[32]; CString str;  //사용자가 지정한 Id정보를 읽어옴 m_edit_driveId.GetWindowText(str); m_DriveId[0] = atoi(str); m_edit_driveId2.GetWindowText(str); m_DriveId[1] = atoi(str);  //Taret IP어드레스 정보를 가져옴 m_edit_targetIp.GetWindowText(s); wsprintf(clp, "%s", s); m_edit_targetIp2.GetWindowText(s); wsprintf(clp2, "%s", s);  //Taret IP어드레스 정보를 가져옴 m_edit_targetPort.GetWindowText(s); Port = atoi(s);  ret = LCOM_TCPOpen(m_DriveId[0], clp, Port); ret = LCOM_TCPOpen(m_DriveId[1], clp2, Port);  //주의 사항 : 연결하는 2축이상 연결 시 슬레이브의 MAC Address가 동일하면 안됨. </pre>		

<b>DLLFUNC int __stdcall LCOM_DriveClose(int DriveId)</b>		
드라이브의 통신을 연결 끊기 위해 사용합니다.		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
Argument		
DriveId	1~31	Node ID
Example		
<pre> /*통신 연결 끊기(2축 기준) int ret, State; // m_DriveId[0] slave의 네트워크 상태를 반환 ret = LCOM_IsSlaveOnline(m_DriveId[0], &amp;State); //네트워크 연결 되어 있으면 m_DriveId[0] slave의 네트워크 종료 if (State)     ret = LCOM_DriveClose(m_DriveId[0]); // m_DriveId[1] slave의 네트워크 상태를 반환 ret = LCOM_IsSlaveOnline(m_DriveId[1], &amp;State); //네트워크 연결 되어 있으면 m_DriveId[1] slave의 네트워크 종료 if (State)     ret = LCOM_DriveClose(m_DriveId[1]);         </pre>		

<b>DLLFUNC int __stdcall LCOM_SetTimeout(int pRxTimeout, int pConnectionTimeout)</b>		
드라이브 통신 송/수신 타임아웃 설정		
<b>DLLFUNC int __stdcall LCOM_GetTimeout(int* pRxTimeout, int* pConnectionTimeout)</b>		
드라이브 통신 송/수신 타임아웃 반환		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
Argument		
pRxTimeout	0~100	소켓 데이터 타임아웃 시간_초기값 3 [Seconds]
pConnectionTimeout	0~100	소켓 연결 타임아웃 시간_초기값 1 [Seconds]
Example		
<pre> /*통신 송/수신 타임아웃 설정*/ int ret; int pRxTimeout, pConnectionTimeout;         </pre>		

```
//통신 송수신 타임아웃 설정
ret = LCOM_GetTimeout(4, 2);
//통신 송/수신 타임아웃 설정 값 반환
ret = LCOM_GetTimeout(&pRxTimeout, &pConnectionTimeout);
```

DLLFUNC int __stdcall LCOM_IsSlaveOnline(int DriveId, int* State)		
개별 Slave의 네트워크 상태를 반환		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
LCOM_CONNECTION_SLALVE_ONLINE_ERR	39	슬레이브가 연결이 되어 있지 않음
Argument		
DriveId	1~31	Node ID
State	0~1	1: 연결된 상태 2: 연결이 되지 않은 상태
<pre>/* 각 slave의 네트워크 상태를 반환*/ int State; //m_DriveId[0] slave측의 연결 상태 확인 ret = LCOM_IsSlaveOnline(m_DriveId[0], &amp;State); if (State) m_MSG_AXIS1.SetWindowText("Connected"); else m_MSG_AXIS1.SetWindowText("Disconnect"); //m_DriveId[1] slave측의 연결 상태 확인 ret = LCOM_IsSlaveOnline(m_DriveId[1], &amp;State); if (State) m_MSG_AXIS2.SetWindowText("Connected"); else m_MSG_AXIS2.SetWindowText("Disconnect");</pre>		

DLLFUNC int __stdcall LCOM_GetSlaveCount(int* Count)		
현재 연결된 Slave의 갯수를 반환		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
Argument		
Count	1~31	연결된 slave의 개수
Example		
/*연결된 Slave의 갯수 반환*/		

```
int ret,cnt;
CString str;
//현재 연결된 slave의 갯수를 반환 하는 함수
ret = LCOM_GetSlaveCount(&cnt);
str.Format("%d", cnt);
m_Axis_cnt.SetWindowText(str);
```

- 라이브러리에 실제로 사용하는 예러만 적음

<b>DLLFUNC int __stdcall LCOM_GetErrorCode(char* cErrorString)</b>	
통신 에러 코드 및 내용 반환	
Return Code	
에러 코드표 참조	
Argument	
cErrorString	에러 코드표 참조
Example	
/*통신 중 발생하는 에러 코드 및 내용 반환*/	
int code=0;	
char Ebuf[256] = { 0 };	
CString str;	
//에러 코드는 Return변수 이며 에러 내용을 반환 합니다.	
code = LCOM_GetErrorCode(Ebuf);	
str.Format("ErrCode=%d(0x%X), %s", code, code, Ebuf);	
m_static_errMsg.SetWindowText(str);	

<b>DLLFUNC int __stdcall LCOM_GetTxPackets(BYTE DriveID, unsigned cahr* pBuffer, unsigned short* pTxSize)</b>		
통신 송신 패킷 반환		
<b>DLLFUNC int __stdcall LCOM_GetRxPackets(BYTE DriveID, unsigned cahr* pBuffer, unsigned short* pRxSize)</b>		
통신 수신 패킷 반환		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
LCOM_PARAMETER_INVALID_ER R	3	송/수신된 파라미터의 데이터가 없을 경우
LCOM_NO_EMPTY_CONNECTION	40	Node I ID를 찾을 수 없을 경우

_ERR		
Argument		
DriveID	1~31	Node ID
pBuffer	0~65	송/수신 버퍼 데이터 최대 65byte
pTxSize/pRxSize	0~65	송/수신 버퍼 길이
Example		
<pre> /*PrintPackets*/ int ret; BYTE sendData[65]={0x00}; BYTE recvData[65]={0x00}; WORD txSize=0,rxSize=0; CString s,sDummy; //송신 데이터 packet 표시 ret = LCOM_GetTxPackets(m_Dest, sendData, &amp;txSize); s.Empty(); sDummy.Empty(); for(int i=0; i&lt;txSize; i++) {     sDummy.Format("%02X", sendData[i]);     s+= sDummy; } m_static_tx.SetWindowText(s); //수신 데이터 packet 표시 ret = LCOM_GetRxPackets(m_Dest, recvData, &amp;rxSize); s.Empty(); sDummy.Empty(); for(int i=0; i&lt;rxSize; i++) {     sDummy.Format("%02X", recvData[i]);     s+= sDummy; } m_static_rx.SetWindowText(s); </pre>		

### 2.2.3 파라미터 R/W 설정 함수

<b>DLLFUNC int __stdcall LCOM_ReadByte(unsigned cahr DriveID, unsigned short Addr, unsigned char SubIndex, unsigned char* pResult)</b>		
[USINT] 파라미터 읽기		
<b>DLLFUNC int __stdcall LCOM_ReadByteS(unsigned cahr DriveID, unsigned short Addr, unsigned char SubIndex, char* pResult)</b>		
[SINT] 파라미터 읽기		
<b>DLLFUNC int __stdcall LCOM_ReadWord(unsigned cahr DriveID, unsigned short Addr, unsigned char SubIndex, unsigned short* pResult)</b>		
[UINT] 파라미터 읽기		
<b>DLLFUNC int __stdcall LCOM_ReadWordS(unsigned cahr DriveID, unsigned short Addr, unsigned char SubIndex, short* pResult)</b>		
[INT] 파라미터 읽기		
<b>DLLFUNC int __stdcall LCOM_ReadDWord(unsigned cahr DriveID, unsigned short Addr, unsigned char SubIndex, unsigned long* pResult)</b>		
[UDINT] 파라미터 읽기		
<b>DLLFUNC int __stdcall LCOM_ReadDWordS(unsigned cahr DriveID, unsigned short Addr, unsigned char SubIndex, signed long* pResult)</b>		
[DINT] 파라미터 읽기		
<b>DLLFUNC int __stdcall LCOM_ReadStirng(unsigned cahr DriveID, unsigned short Addr, unsigned char SubIndex, char* pResult)</b>		
[STRING] 파라미터 읽기		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
LCOM_MEL_ETHERNET_NORESPONSE_ERR	26	수신된 데이터가 6byte보다 작을 경우
LCOM_MEL_ETHERNET_WRITE_ERROR	27	송신 데이터 Write 실패 했을 경우
LCOM_MEL_ETHERNET_REPLYCMD_ERR	29	송/수신 패킷이 일치 하지 않을 경우
LCOM_MEL_ETHERNET_RXPACKET_MISSING_ERR	32	수신 패킷이 양이 많을 경우
LCOM_MEL_ETHERNET_SOCKET_NULL_ERR	34	이더넷 연결이 되어 있지 않을 경우
LCOM_MEL_ETHERNET_DRIVEID_NOTACTIVE_ERR	36	Node IID를 찾을 수 없을 경우
LCOM_CONNECTION_EMPTY_ERROR	38	송/수신된 파라미터의 데이터가 없을 경우

Argument		
DriveID	1~31	Node ID
Addr	0x0000~ 0xFFFF	Index 주소(2byte)
SubIndex	0x00~0xFF	Sub-Index 주소(1byte)
pResult		수신 버퍼 데이터 최대 65byte
Example		
<pre> /*파라미터 읽기 type별 예제*/ int ret; //[USINT]파라미터 read unsigned char pResult; ret = LCOM_ReadByte(m_DriveId[0], 0x6099, 0x00, &amp;pResult); //[SINT]파라미터 read char pResult; ret = LCOM_ReadBytes(m_DriveId[0], 0x6098, 0x00, &amp;pResult); //[UINT]파라미터 read unsigned short pResult; ret = LCOM_ReadWord(m_DriveId[0], 0x60B9, 0x00, &amp;pResult); //[INT]파라미터 read short pResult; ret = LCOM_ReadWords(m_DriveId[0], 0x6077, 0x00, &amp;pResult); //[UDINT]파라미터 read unsigned long pResult; ret = LCOM_ReadDWord(m_DriveId[0], 0x6079, 0x00, &amp;pResult); //[DINT]파라미터 read signed long pResult; ret = LCOM_ReadDWords(m_DriveId[0], 0x607C, 0x00, &amp;pResult); //[STRING]파라미터 read char pResult; ret = LCOM_ReadString(m_DriveId[0], 0x240D, 0x00, &amp;pResult);  //SubIndex파라미터 read 시 //예를 들어 기어비(0x6091)의 Motor회전수[UDINT]SubIndex1을 읽을 경우 unsigned long pResult; ret = LCOM_ReadDWord(m_DriveId[0], 0x6091, 0x01, &amp;pResult); </pre>		

<b>DLLFUNC int __stdcall LCOM_WriteByte(unsigned cahr DriveID, unsigned short Addr, unsigned char SubIndex, unsigned char Data)</b>		
[USINT] 파라미터 쓰기		
<b>DLLFUNC int __stdcall LCOM_WriteByteS(unsigned cahr DriveID, unsigned short Addr, unsigned char SubIndex, char Data)</b>		
[SINT] 파라미터 쓰기		
<b>DLLFUNC int __stdcall LCOM_WriteWord(unsigned cahr DriveID, unsigned short Addr, unsigned char SubIndex, unsigned short Data)</b>		
[UINT] 파라미터 쓰기		
<b>DLLFUNC int __stdcall LCOM_WriteWordS(unsigned cahr DriveID, unsigned short Addr, unsigned char SubIndex, short Data)</b>		
[INT] 파라미터 쓰기		
<b>DLLFUNC int __stdcall LCOM_WriteDWord(unsigned cahr DriveID, unsigned short Addr, unsigned char SubIndex, unsigned long Data)</b>		
[UDINT] 파라미터 쓰기		
<b>DLLFUNC int __stdcall LCOM_WriteDWordS(unsigned cahr DriveID, unsigned short Addr, unsigned char SubIndex, signed long Data)</b>		
[DINT] 파라미터 쓰기		
<b>DLLFUNC int __stdcall LCOM_WriteString(unsigned cahr DriveID, unsigned short Addr, unsigned char SubIndex, char* Data)</b>		
[STRING] 파라미터 쓰기		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
LCOM_MEL_ETHERNET_NORESPONSE_ERR	26	수신된 데이터가 6byte보다 작을 경우
LCOM_MEL_ETHERNET_WRITE_ERROR	27	송신 데이터 Write 실패 했을 경우
LCOM_MEL_ETHERNET_REPLYCMD_ERR	29	송/수신 패킷이 일치 하지 않을 경우
LCOM_MEL_ETHERNET_RXPACKET_MISSING_ERR	32	수신 패킷이 양이 많을 경우
LCOM_MEL_ETHERNET_REQUEST_BYTESIZE_TOOBIG_ERR	33	송신 패킷의 양이 많을 경우
LCOM_MEL_ETHERNET_SOCKET_NULL_ERR	34	이더넷 연결이 되어 있지 않을 경우
LCOM_MEL_ETHERNET_DRIVEID_NOTACTIVE_ERR	36	Node IID를 찾을 수 없을 경우

LCOM_CONNECTION_EMPTY_ER R	38	송/수신된 파라미터의 데이터가 없을 경우
Argument		
DriveID	1~31	Node ID
Addr	0x0000~ 0xFFFF	Index 주소(2byte)
SubIndex	0x00~0xFF	Sub-Index 주소(1byte)
Data		Write할 데이터
Example		
<pre> /*파라미터 쓰기 type별 예제*/ int ret; //[USINT]파라미터 write unsigned char pData; ret = LCOM_WriteByte(m_DriveId[0], 0x6099, 0x00, pData); //[SINT]파라미터 write char pResult; ret = LCOM_WriteBytes(m_DriveId[0], 0x6098, 0x00, pData); //[UINT]파라미터 write unsigned short pResult; ret = LCOM_WriteWord(m_DriveId[0], 0x60B9, 0x00, pData); //[INT]파라미터 write short pResult; ret = LCOM_WriteWords(m_DriveId[0], 0x6077, 0x00, pData); //[UDINT]파라미터 write unsigned long pResult; ret = LCOM_WriteDWord(m_DriveId[0], 0x6079, 0x00, pData); //[DINT]파라미터 write signed long pResult; ret = LCOM_WriteDWords(m_DriveId[0], 0x607C, 0x00, pData); //[STRING]파라미터 write char pResult; ret = LCOM_WriteString(m_DriveId[0], 0x240D, 0x00, pData);  //SubIndex파라미터 write 시 //예를 들어 기어비(0x6091)의 Motor회전수[UDINT]SubIndex1을 쓰기 할 경우 unsigned long pResult; ret = LCOM_WriteDWord(m_DriveId[0], 0x6091, 0x01, pData); </pre>		

<b>DLLFUNC int __stdcall LCOM_DriveSoftReset(unsigned char DriveID)</b>		
서보드라이브 S/W 리셋 [0x2700:0x0009][0x2701:0x01]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Example		
<pre> /*소프트웨어 리셋*/ int ret; // m_DriveId[0] slave측의 프로시저 기능을 활성화 하여 S/W Reset 실행. ret = LCOM_DriveSoftReset(m_DriveId[0]);                     </pre>		

<b>DLLFUNC int __stdcall LCOM_StoreParameter (unsigned char DriveID)</b>		
파라미터 저장 [0x1010:01] [0x1010:05]		
L7P Series의 경우 인덱스 파라미터를 저장하기 위해서 SubIndex 5도 write해줘야함(매뉴얼참조)		
SubIndex1와 SubIndex5 저장 사이에 Delay 2초 적용.		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Example		
<pre> /*파라미터 저장*/ int ret; // m_DriveId[0] slave측의 데이터 저장 ret = LCOM_StoreParameter(m_DriveId[0]);                     </pre>		

### 2.2.4 제어 파라미터 설정 함수

<b>DLLFUNC int __stdcall LCOM_SetControlMode(unsigned char DriveID, unsigned short Mode)</b>		
L7P 제어 모드 설정 [UINT][0x3000]		
Return Code		

LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Mode	0~1	0: 인텍스 위치운전 모드 1: 펄스입력 위치운전 모드
Example		
<pre> /*제어 모드 설정*/ int ret; unsigned short Mode; Mode = 1; //펄스입력 위치 운전 모드 // m_DriveId[0] slave축의 제어 모드 1로 변경 ret = LCOM_SetControlMode(m_DriveId[0], Mode); </pre>		

<b>DLLFUNC int __stdcall LCOM_SetElecGearRatio(unsigned char DriveID, unsigned long MotorRev, unsigned long ShaftRev)</b>		
기어비 설정[0x6091]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
MotorRev	0~1073741824	Sub-Index1:[UDINT]모터 회전수
ShaftRev	0~1073741824	Sub-Index2:[UDINT]샤프트 회전수
Example		
<pre> /*기어비 설정*/ int ret; unsigned long MotorRev, ShaftRev; MotorRev = 1; //모터 회전수 ShaftRev = 1; //샤프트 회전수 // m_DriveId[0] slave축의 기어비 설정 ret = LCOM_SetElecGearRatio(m_DriveId[0], MotorRev, ShaftRev); </pre>		

<b>DLLFUNC int __stdcall LCOM_SetCurGain(unsigned char DriveID, unsigned short Gain)</b>		
--	--	--

<b>DLLFUNC int __stdcall LCOM_GetCurGain(unsigned char DriveID, unsigned short* Gain)</b>		
전류 게인 설정[0x2514]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Gain	1~150	[UINT] 전류 게인[%]
Example		
<pre> /*전류 게인 설정*/ int ret; unsigned short Gain; Gain = 10; //전류 게인 설정 값 // m_DriveId[0] slave측의 전류 게인 설정 ret = LCOM_SetCurGain(m_DriveId[0], Gain); // m_DriveId[0] slave측의 전류 게인값 반환 ret = LCOM_GetCurGain(m_DriveId[0], &amp;Gain);                     </pre>		

<b>DLLFUNC int __stdcall LCOM_SetInertiaRatio(unsigned char DriveID, unsigned short Ratio)</b>		
<b>DLLFUNC int __stdcall LCOM_GetInertiaRatio(unsigned char DriveID, unsigned short* Ratio)</b>		
관성비 설정[0x2100]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Ratio	1~3000	[UINT] 관성비[%]
Example		
<pre> /*기어비 설정*/ int ret; unsigned short Ratio; Ratio = 10; //관성비 설정 값 // m_DriveId[0] slave측의 관성비 설정 ret = LCOM_SetInertiaRatio(m_DriveId[0], Ratio); // m_DriveId[0] slave측의 관성비 반환 ret = LCOM_GetInertiaRatio(m_DriveId[0], &amp;Ratio);                     </pre>		

<b>DLLFUNC int __stdcall LCOM_SetVelGainKp(unsigned char DriveID, unsigned short Gain)</b> <b>DLLFUNC int __stdcall LCOM_GetVelGainKp(unsigned char DriveID, unsigned short* Gain)</b>		
속도 비례 게인 설정 [0x2102]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Gain	1~2000	[UINT] 속도비례게인[Hz]
Example		
<pre> /*속도 비례 게인 설정*/ int ret; unsigned short Gain; Gain = 10; //속도 비례 게인 설정 값 // m_DriveId[0] slave축의 속도 비례 게인 설정 ret = LCOM_SetVelGainKp(m_DriveId[0], Gain); // m_DriveId[0] slave축의 속도 비례 게인 반환 ret = LCOM_GetVelGainKp(m_DriveId[0], &amp;Gain); </pre>		

<b>DLLFUNC int __stdcall LCOM_SetVelGainKiT(unsigned char DriveID, unsigned short Gain)</b> <b>DLLFUNC int __stdcall LCOM_GetVelGainKiT(unsigned char DriveID, unsigned short* Gain)</b>		
속도 적분 시정수 설정 [0x2103]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Gain	1~1000	[UINT] 속도적분시정수[ms]
Example		
<pre> /*속도 적분 시정수 설정*/ int ret; unsigned short Gain; Gain = 10; //속도 적분 시정수 설정 값 // m_DriveId[0] slave축의 속도 적분 시정수 설정 ret = LCOM_SetVelGainKiT(m_DriveId[0], Gain); </pre>		

```
// m_DriveId[0] slave축의 속도 적분 시정수 반환
ret = LCOM_GetVelGainKiT(m_DriveId[0], &Gain);
```

**DLLFUNC int \_\_stdcall LCOM\_SetPosGainKp(unsigned char DriveID, unsigned short Gain)**  
**DLLFUNC int \_\_stdcall LCOM\_GetPosGainKp(unsigned char DriveID, unsigned short\* Gain)**

위치 비례 게인 설정[0x2101]

Return Code

LCOM_OK	0	에러 없이 통신 성공
---------	---	-------------

파라미터 쓰기와 동일 Return Code

Argument

DriveID	1~31	Node ID
Gain	1~500	[UINT] 위치비례게인 [/s]

Example

```
/*위치 비례 게인 설정*/
int ret;
unsigned short Gain;
Gain = 10; //위치 비례 게인 설정 값
// m_DriveId[0] slave축의 위치 비례 게인 설정
ret = LCOM_SetPosGainKp(m_DriveId[0], Gain);
// m_DriveId[0] slave축의 위치 비례 게인 반환
ret = LCOM_GetPosGainKp(m_DriveId[0], &Gain);
```

**DLLFUNC int \_\_stdcall LCOM\_SetPosGainKvff(unsigned char DriveID, unsigned short Gain)**  
**DLLFUNC int \_\_stdcall LCOM\_GetPosGainKvff(unsigned char DriveID, unsigned short\* Gain)**

위치제어 속도 피드포워드 게인 설정[0x210C]

Return Code

LCOM_OK	0	에러 없이 통신 성공
---------	---	-------------

파라미터 쓰기와 동일 Return Code

Argument

DriveID	1~31	Node ID
Gain	1~100	[UINT] 위치제어 속도 피드포워드 게인 [%]

Example

```
/*위치 비례 게인 설정*/
int ret;
unsigned short Gain;
```

```
Gain = 10; //위치 비례 게인 설정 값
// m_DriveId[0] slave측의 위치 비례 게인 설정
ret = LCOM_SetPosGainKp(m_DriveId[0], Gain);
// m_DriveId[0] slave측의 위치 비례 게인 반환
ret = LCOM_GetPosGainKp(m_DriveId[0], &Gain);
```

<b>DLLFUNC int __stdcall LCOM_SetPosGainKvff_ft(unsigned char DriveID, unsigned short Gain)</b> <b>DLLFUNC int __stdcall LCOM_GetPosGainKvff_ft(unsigned char DriveID, unsigned short* Gain)</b>		
위치제어 속도 피드포워드 게인 필터 시정수 설정[0x210D]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Gain	0~1000	[UINT] 위치제어 속도 피드포워드 게인 필터 시정수[0.1ms]
Example		
<pre>/*위치제어 속도 피드포워드 게인 필터 시정수 설정*/ int ret; unsigned short Gain; Gain = 10; //시정수 설정 값 // m_DriveId[0] slave측의 시정수 설정 ret = LCOM_SetPosGainKvff_ft(m_DriveId[0], Gain); // m_DriveId[0] slave측의 시정수 반환 ret = LCOM_GetPosGainKvff_ft(m_DriveId[0], &amp;Gain);</pre>		

### 2.2.5 모니터링 함수

<b>DLLFUNC int __stdcall LCOM_FeedbackSpeed(unsigned char DriveID, short* Vel)</b>		
피드백 속도 반환[0x2600]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		

DriveID	1~31	Node ID
Vel	-32768~32767	[INT] 속도피드백[rpm]
Example		
<pre> /*피드백 속도 반환*/ int ret; short Vel;  // m_DriveId[0] slave측의 피드백 속도 반환 ret = LCOM_FeedbackSpeed(m_DriveId[0], &amp;Vel);                     </pre>		

<b>DLLFUNC int __stdcall LCOM_ActualPosition(unsigned char DriveID, long* Pos)</b>		
실제 위치값 반환[0x6064]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Pos	-2147483647~ 2147483647	[DINT] 실제 위치값[UU]
Example		
<pre> /*실제 위치값 반환*/ int ret; long Pos;  // m_DriveId[0] slave측의 실제 위치값 반환 ret = LCOM_ActualPosition(m_DriveId[0], &amp;Pos);                     </pre>		

<b>DLLFUNC int __stdcall LCOM_ActualVelocity(unsigned char DriveID, long* Vel)</b>		
실제 속도값 반환[0x606C]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Vel	-2147483647~ 2147483647	[DINT] 실제 속도값[UU/s]
Example		

```

/*실제 속도값 반환*/
int ret;
long Vel;
// m_DriveId[0] slave축의 실제 속도값 반환
ret = LCOM_ActualVelocity(m_DriveId[0], &Vel);
    
```

DLLFUNC int __stdcall LCOM_ActualTorque(unsigned char DriveID, short* Torque)		
실제 토크값 반환[0x6077]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Torque	-32768~32768	[INT] 실제 토크값[0.1%]
Example		
<pre> /*실제 토크값 반환*/ int ret; short Torque; // m_DriveId[0] slave축의 실제 토크값 반환 ret = LCOM_ActualTorque(m_DriveId[0], &amp;Torque);                 </pre>		

DLLFUNC int __stdcall LCOM_SingleTurnData(unsigned char DriveID, unsigned long* Data)		
1회전 내 데이터 반환[0x2607]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Data	0~4294967295	[UDINT] 1회전 내 데이터[pulse]
Example		
/*1회전 내 데이터 반환*/		

```
int ret;
unsigned long Data;
// m_DriveId[0] slave축의 데이터 반환
ret = LCOM_SingleTurnData(m_DriveId[0], &Data);
```

DLLFUNC int __stdcall LCOM_MultiTurnData(unsigned char DriveID, unsigned long* Data)		
다회전 데이터 반환[0x260A]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Data	0~4294967295	[UDINT] 다회전 데이터[pulse]
Example		
/*다회전 데이터 반환*/		
<pre>int ret; unsigned long Data; // m_DriveId[0] slave축의 데이터 반환 ret = LCOM_MultiTurnData(m_DriveId[0], &amp;Data);</pre>		

DLLFUNC int __stdcall LCOM_WarningCode(unsigned char DriveID, unsigned short* Code)		
경고 코드 반환[0x2614]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Code	0~65535	[UINT] 경고 코드
Example		
/*경고 코드 반환*/		
<pre>int ret; unsigned short Code; // m_DriveId[0] slave축의 경고 코드 반환 ret = LCOM_WarningCode(m_DriveId[0], &amp;Code);</pre>		

<b>DLLFUNC int __stdcall LCOM_DCLinkVoltage(unsigned char DriveID, unsigned short* Volt)</b>		
DC-Link 전압 반환[0x2605]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Volt	0~65535	[UINT] DC-Link 전압[V]
Example		
<pre> /*DC-Link 전압 반환*/ int ret; unsigned short Volt; // m_DriveId[0] slave측의 경고 코드 반환 ret = LCOM_DCLinkVoltage(m_DriveId[0], &amp;Volt); </pre>		

<b>DLLFUNC int __stdcall LCOM_AccumOverload(unsigned char DriveID, short* Overload)</b>		
누적 운전 과부하율 반환[0x2603]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Overload	-32768~32768	[INT] 누적 운전 과부하율[0.1%]
Example		
<pre> /*누적 운전 과부하율 반환*/ int ret; short Overload; // m_DriveId[0] slave측의 과부하율 반환 ret = LCOM_AccumOverload(m_DriveId[0], &amp;Overload); </pre>		

<b>DLLFUNC int __stdcall LCOM_DriveTemp(unsigned char DriveID, short* Temp)</b>		
드라이브 내부 온도 반환[0x260B]		
Return Code		

LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Temp	-32768~32768	[INT] DC-Link 전압[degree]
Example		
<pre> /*드라이브 내부 온도 반환*/ int ret; short Temp;  // m_DriveId[0] slave측의 온도 반환 ret = LCOM_AccumOverload(m_DriveId[0], &amp;Temp);                     </pre>		

### 2.2.6 알람 상태 및 처리 함수

DLLFUNC int __stdcall LCOM_IsAlarm(unsigned char DriveID, int* State, unsigned short* Code)		
현재 알람 상태 및 코드 반환[0x2121:01][0x603F]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
State	0~1	현재 알람 상태 정보
Code	0~100	[UINT]알람 코드
Example		
<pre> /*현재 알람 상태 반환*/ int ret, AlarmState; unsigned short Alarmcode; CString str;  // m_DriveId[0] slave측의 알람 비트 확인 ret = LCOM_IsAlarm(m_DriveId[0], &amp;alarmState, &amp;AlarmCode); if(AlarmState) str.Format("AlarmCode= %X)", AlarmCode); // 알람코드는 Hex로 표시됨                     </pre>		

DLLFUNC int __stdcall LCOM_ResetAlarm(unsigned char DriveID)		
현재 알람 상태 리셋[0x211F:11]		

Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Example		
<pre> /*현재 알람 상태 리셋*/ int ret; // m_DriveId[0] slave측의 알람 리셋 ret = LCOM_ResetAlarm(m_DriveId[0]);         </pre>		

DLLFUNC int __stdcall LCOM_GetAlarmHistory(unsigned char DriveID, unsigned char SubIndex ,char* Buf)		
알람 히스토리 내용 반환[0x2702:01~10]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Buf		알람코드 내용
Example		
<pre> /*현재 알람 코드 및 내용 반환*/ int ret; unsigned short SubIndex; char AlarmBuf[256]={0}; CString str;  SubIndex = 1; //가장 최근의 알람 히스토리 데이터 // m_DriveId[0] slave측의 최근 발생된 알람 히스토리 데이터 반환 ret= LCOM_GetAlarmHistory(m_DriveId[0], SubIndex, AlarmBuf); str.Format("AlarmHistory= %s", Alarmbuf); m_static_AlarmMsg.SetWindowText(str);         </pre>		

DLLFUNC int __stdcall LCOM_ClearAlarmHistory(unsigned char DriveID)		
알람 히스토리 삭제[0x2700:0x0003][0x2701:0x01]		
Return Code		

LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Example		
<pre> /*알람 히스토리 삭제*/ int ret; // m_DriveId[0] slave측의 알람 히스토리 삭제 ret = LCOM_ClearAlarmHistory(m_DriveId[0]);                     </pre>		

### 2.2.7 입출력 포트 처리 함수

<b>DLLFUNC int __stdcall LCOM_SetDigitalInputSignals(unsigned char DriveID, unsigned char SignalNo, unsigned short Signal, unsigned short Level)</b>		
디지털 입력 신호 설정[0x2200~0x2207]		
<b>DLLFUNC int __stdcall LCOM_GetDigitalInputSignals(unsigned char DriveID, unsigned char SignalNo, unsigned short* Signal, unsigned short* Level)</b>		
디지털 입력 신호 반환[0x2200~0x2207]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
LCOM_PARAMETER_INVALID_ERR	3	어드레스가 일치 하지 않음
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
SignalNo	1~8	[0x2200~0x2208] 0x2200 : 디지털입력 신호 1
Signal	0~12	0: NONE 1:POT 2:NOT 3:HOME 4:STOP 5:PCON 6: GAIN2 7:P_CL 8:N_CL 9:PROBE1 10: PROBE2 11:EMG 12:A_RST
Level	0~1	0: A접점 1: B접점
Example		
<pre> /*디지털 입력 신호 설정*/ int ret; unsigned char SignalNo,Signal,Level;                     </pre>		

```
SignalNo = 1; //0x2200번지 디지털입력 신호 1
Signal = 3; //HOME 입력 신호로 설정
Level = 0; //A접점으로 사용
// m_DriveId[0] slave축의 디지털입력 신호 1설정
ret = LCOM_SetDigitalInputSignals(m_DriveId[0],SignalNo,Signal,Level);

ret = LCOM_GetDigitalInputSignals(m_DriveId[0],SignalNo,&Signal,&Level);
```

**DLLFUNC int \_\_stdcall LCOM\_SetDigitalOutputSignals(unsigned char DriveID, unsigned char SignalNo, unsigned short Signal, unsigned short Level)**

디지털 출력 신호 설정[0x2210~0x2213]

**DLLFUNC int \_\_stdcall LCOM\_GetDigitalOutputSignals(unsigned char DriveID, unsigned char SignalNo, unsigned short\* Signal, unsigned short\* Level)**

디지털 출력 신호 반환[0x2210~0x2213]

Return Code

LCOM_OK	0	에러 없이 통신 성공
LCOM_PARAMETER_INVALID_ERR	3	어дрес가 일치 하지 않음

파라미터 쓰기와 동일 Return Code

Argument

DriveID	1~31	Node ID
SignalNo	1~8	[0x2210~0x2213] 0x2210 : 디지털출력 신호 1
Signal	0~12	0: NONE 1:BRAKE 2:ALRM 3:READY 4:ZSPD 5:INPOS1 6: TLMT 7:VLMT 8:INSPD 9:WARN 10: TGON 11:INPOS2
Level	0~1	0: A접점 1: B접점

Example

```
/*디지털 입력 신호 설정*/
int ret;
unsigned char SignalNo,Signal,Level;
SignalNo = 1; //0x2210번지 디지털출력 신호 1
Signal = 3; //READY 출력 신호로 설정
Level = 0; //A접점으로 사용
// m_DriveId[0] slave축의 디지털출력 신호 1설정
ret = LCOM_SetDigitalOutputSignals(m_DriveId[0],SignalNo,Signal,Level);
```

```
ret = LCOM_GetDigitalOutputSignals(m_DriveId[0],SignalNo,&Signal,&Level);
```

DLLFUNC int __stdcall LCOM_GetDigitalInputs(unsigned char DriveID, unsigned long* Val)		
디지털 입력 값 반환[0x60FD]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Val	0~ 4294967295	[UDINT] 각 bit신호는 매뉴얼 참조
Example		
<pre>/*디지털 입력 값 반환*/ int ret; unsigned long Val; // m_DriveId[0] slave측의 디지털 입력 상태 값 반환 //32bit로 되어 있으며 각 bit값은 Argument 참조 ret = LCOM_GetDigitalInput(m_DriveId[0],&amp;Val);</pre>		

DLLFUNC int __stdcall LCOM_IsNegativeLimit(unsigned char DriveID, int* State)		
역방향 리미트 스위치 반환[0x60FD:0]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
State	0~1	[UDINT] [0x60FD]의 0 bit값
Example		
<pre>/*NOT 입력 값 반환*/ int ret; int State; // m_DriveId[0] slave측의 디지털 입력 값 중 NOT(역방향 리미트 스위치)상태 반환 ret = LCOM_IsNegativeLimit(m_DriveId[0],&amp;State);</pre>		

--

DLLFUNC int __stdcall LCOM_IsPositiveLimit(unsigned char DriveID, int* State)		
정방향 리미트 스위치 반환[0x60FD:1]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
State	0~1	[UDINT] [0x60FD]의 1 bit값
Example		
<pre> /*POT 입력 값 반환*/ int ret; int State;  // m_DriveId[0] slave측의 디지털 입력 값 중 POT(정방향 리미트 스위치)상태 반환 ret = LCOM_IsPositiveLimit(m_DriveId[0],&amp;State);                     </pre>		

DLLFUNC int __stdcall LCOM_IsHomeSensor(unsigned char DriveID, int* State)		
원점 센서 입력 반환[0x60FD:2]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
State	0~1	[UDINT] [0x60FD]의 2 bit값
Example		
<pre> /*HOME 입력 값 반환*/ int ret; int State;  // m_DriveId[0] slave측의 디지털 입력 값 중 HOME(원점 센서 입력)상태 반환 ret = LCOM_IsHomeSensor(m_DriveId[0],&amp;State);                     </pre>		

DLLFUNC int __stdcall LCOM_GetDigitalOutputs (unsigned char DriveID, unsinged long* Val)		
디지털 출력 상태 신호 반환[0x60FE::1]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Val	0~ 4294967295	[UDINT] [0x60FE::1] 물리적 출력값 반환 각 bit 신호는 매뉴얼 참조
Example		
<pre> /*물리적 출력 값 반환*/ int ret; int State; // m_DriveId[0] slave측의 디지털 출력값 반환 ret = LCOM_GetDigitalOutputs(m_DriveId[0],&amp;Val);                     </pre>		

DLLFUNC int __stdcall LCOM_SetDigitalOutPutsWithMask (unsigned char DriveID, unsinged long Val, unsinged long Mask)		
디지털 출력 상태 신호 설정[0x60FE::1] [0x60FE::2]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Val	0~ 4294967295	[UDINT] [0x60FE::1] 물리적 출력 설정 각 bit 신호는 매뉴얼 참조
Mask	0~ 4294967295	[UDINT] [0x60FE::2] 비트 마스크 설정 각 bit 신호는 매뉴얼 참조
Example		
<pre> /*물리적 출력 값 반환*/ int ret;                     </pre>		

```

unsigned long Val,Mask;

Val = 0x01000000; //DO#1출력 On

Mask = 0x00010000; //DO#1 강제출력 Enable

// m_DriveId[0] slave축의 출력 상태 신호 설정

ret = LCOM_SetDigitalOutPutsWithMask(m_DriveId[0],Val,Mask);
    
```

### 2.2.8 모션 동작 함수

<b>DLLFUNC int __stdcall LCOM_SetOperationMode(unsigned char DriveID, unsigned char Mode)</b>		
운전 모드 설정[0x6060]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Mode	0~10	[SINT] 1: Profile Position 모드 3: Profile Velocity 모드 4: Profile Torque 모드 6: Homing 모드 8: Cyclic Synchronous Position 모드 9: Cyclic Synchronous Velocity 모드 10: Cyclic Synchronous Torque 모드
Example		
<pre> /*운전 모드 설정*/ int ret; unsigned char mode; mode = 6; // Homing 모드 선택 // m_DriveId[0] slave축의 운전 모드를 Homing모드로 선택 ret = LCOM_SetOperationMode(m_DriveId[0], mode);     </pre>		

```

DLLFUNC int __stdcall LCOM_Homing(unsigned char DriveID, char Method, long Speed, long FineSpeed
, unsigned long Acc, short Offset)
    
```

서보축의 원점 복귀 동작 명령		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Method	-4 ~ 35	[SINT] Homing 방법 매뉴얼 참조
Speed	0~ 1073741824	[UDINT] Switch탐색 속도
FineSpeed	0~ 1073741824	[UDINT] Zero 탐색 속도
Acc	0~ 1073741824	[UDINT] Homing 가속도
Offset	-536870922 ~536870922	[DINT] Home Offset
Example		
<pre> /*Homing 운전*/ int ret; char Method; long Speed,FineSpeed,HomeOffset; unsigned long Acc;  Method = 35; Speed = 500000; FineSpeed = 100000; Acc = 200000; HomeOffset = 500;  // m_DriveId[0] Homing 35방법으로 위의 설정으로 동작 ret = LCOM_Homing(m_DriveId[0], Method, Speed, FineSpeed, Acc, HomeOffset);         </pre>		

DLLFUNC int __stdcall LCOM_IsHome(unsigned char DriveID, int* State)		
원점복귀 완료 신호[0x2122::0]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공

파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Example		
<pre> /*Homing 완료 신호*/ int ret; int State; // m_DriveId[0] Homing 완료 신호 확인 ret = LCOM_IsHome(m_DriveId[0],&amp;State);         </pre>		

<b>DLLFUNC int __stdcall LCOM_HomeTorqueLimit(unsigned char DriveID, unsigned short Torque, unsigned short Time)</b>		
Stopper 이용 원점 복귀 시 토크 제한 값 설정		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Torque	0~2000	[UINT] Stopper 이용 원점 복귀 시 토크 제한값 설정
Time	0~1000	[UINT] Stopper 이용 원점 복귀 시 시간 설정
Example		
<pre> /*Stopper 관련 설정*/ int ret; unsigned short Torque,Time; // m_DriveId[0] Stopper 관련 파라미터 설정 ret = LCOM_IsHomeTorqueLimit(m_DriveId[0], Torque, Time);         </pre>		

<b>DLLFUNC int __stdcall LCOM_Enable(unsigned char DriveID)</b> <b>DLLFUNC int __stdcall LCOM_Disable(unsigned char DriveID)</b>		
Servo ON/OFF상태 제어 [0x211F:12]		
Return Code		

LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Example		
<pre> /*Servo on/off상태 제어*/ int ret; // m_DriveId[0] slave축 Servo on상태로 변경 ret = LCOM_Enable(m_DriveId[0]); // m_DriveId[0] slave축 Servo off상태로 변경 ret = LCOM_Disable(m_DriveId[0]);                     </pre>		

<b>DLLFUNC int __stdcall LCOM_IsEnabled(unsigned char DriveID, int* State)</b>		
Servo ON/OFF상태 확인[0x211F:12]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
State	0~1	0: Servo Off 1: Servo On
Example		
<pre> /*Servo on/off상태 확인*/ int ret,State; // m_DriveId[0] slave축 Servo on/off 상태 반환 ret = LCOM_IsEnabled(m_DriveId[0], &amp;State);                     </pre>		

<b>DLLFUNC int __stdcall LCOM_EmergencyStop(unsigned char DriveID)</b>		
비상정지 신호 설정[0x211F:10]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Example		
<pre> /*비상 정지 신호 설정*/                     </pre>		

```
int ret;
// m_DriveId[0] slave측 0x211F bit를 전부 clear 후 EMG 비트만 set
ret = LCOM_EmergencyStop(m_DriveId[0]);
```

DLLFUNC int __stdcall LCOM_Stop(unsigned char DriveID)		
운전 정지 신호 설정[0x211F:03]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Example		
/*운전 정지 신호 설정*/		
int ret;		
// m_DriveId[0] slave측 0x211F Stop 비트만 clear 후 set		
ret = LCOM_Stop(m_DriveId[0]);		

DLLFUNC int __stdcall LCOM_Start(unsigned char DriveID)		
운전 시작 신호 설정[0x2120:00]		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Example		
/*운전 시작 신호 설정*/		
int ret;		
// m_DriveId[0] slave측 0x2120 bit를 전부 clear 후 START 비트만 set		
ret = LCOM_Start(m_DriveId[0]);		

DLLFUNC int __stdcall LCOM_SetCurrentPosition(unsigned char DriveID, long HomeOffset)		
현재 위치를 원점으로 설정 (Servo On상태 일 시만 적용 됨)		
Return Code		

LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
HomeOffset	-536870911~ 536870911	[DINT]설정 위치값[UU]
Example		
<pre> /*Offset값에 따른 현재 위치 원점 설정*/ int ret; long HomeOffset; int State; // 원점 설정은 Servo On상태 일때만 적용이 되기 때문에 // Servo의 상태 확인. ret = LCOM_IsEnabled(m_DriveId[0], &amp;State); if (!State){     ret = LCOM_Enable(m_DriveId[0]); } HomeOffset = 0; //위치를 0으로 원점 설정 ret = LCOM_SetCurrentPosition(m_DriveId[0], HomeOffset);                     </pre>		

<b>DLLFUNC int __stdcall LCOM_MoveAbsolute(unsigned char DriveID, signed long Pos, unsigned long Vel, unsigned long Acc, unsigned long Dec )</b>		
절대치 이동 설정		
<b>DLLFUNC int __stdcall LCOM_MoveRelative(unsigned char DriveID, signed long Pos, unsigned long Vel, unsigned long Acc, unsigned long Dec )</b>		
상대치 이동 설정		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Pos	-2147483647~ 2147483647	[0x607A] [DINT]목표 위치[UU]
Vel	0~	[0x6081]

	2147483647	[UDINT]프로파일 속도[UU/s]
Vcc	0~ 2147483647	[0x6083] [UDINT]프로파일 가속도[UU/s^2]
Dec	0~ 2147483647	[0x6084] [UDINT]프로파일 감속도[UU/s^2]
Example		
<pre> /*절대치 상대치 이동 설정*/ int ret; unsigned char mode; signed long Pos; unsigned long Vel, Acc, Dec; mode = 1; // PP운전 모드 Pos = 5242880; // 목표 위치 Vel = 873800; // 속도 Acc = Dec= 8738000; //가감속 설정 // m_DriveId[0] slave축의 운전 모드를 PP모드로 선택 ret = LCOM_SetOperationMode(m_DriveId[0], mode); // m_DriveId[0] slave축 Servo On상태 ret = LCOM_Enable(m_DriveId[0]); //절대치 이동 명령 ret = LCOM_MoveAbsolute(m_DriveId[0], Pos, Vel, Acc, Dec); Pos = 52428800; // 목표 이동 거리 //상대치 이동 명령 ret = LCOM_MoveRelative(m_DriveId[0], Pos, Vel, Acc, Dec);         </pre>		

DLLFUNC int __stdcall LCOM_MoveAbs (unsigned char DriveID, signed long Pos)		
절대치 이동 목표 위치만 입력		
DLLFUNC int __stdcall LCOM_MoveRel (unsigned char DriveID, signed long Pos)		
상대치 이동 목표 위치만 입력		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Pos	-2147483647~ 2147483647	[0x607A] [DINT]목표 위치[UU]

```

Example
/*절대치 상대치 목표 위치 설정*/
int ret;
signed long Pos;
Pos = 5242880;          // 목표 위치
//절대치 이동 목표 위치 설정
ret = LCOM_MoveAbs(m_DriveId[0], Pos);
Pos = 5242880;          // 목표 이동 거리
//상대치 이동 목표 위치 설정
ret = LCOM_MoveRel(m_DriveId[0], Pos);
    
```

<b>DLLFUNC int __stdcall LCOM_MoveIndex (unsigned char DriveID, int IndexNo)</b>		
해당 인덱스 No로 운전 실시		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
State	0~1	[0x2122:01] [UINT]INPOS1
Example		
/*해당 인덱스 No로 운전*/		
int ret,IndexNo;		
IndexNo = 0; //0번 인덱스 설정		
// m_DriveId[0] slave축 0번 인덱스로 운전 시작		
ret = LCOM_MoveIndex(m_DriveId[0], 0);		

<b>DLLFUNC int __stdcall LCOM_ReadIndexParameter (unsigned char DriveID, int IndexNo, INDEXMOTIOM_PROFILE *pInfo)</b>		
인덱스 파라미터 읽기		
<b>DLLFUNC int __stdcall LCOM_WriteIndexParameter (unsigned char DriveID, int IndexNo, INDEXMOTIOM_PROFILE *pInfo)</b>		
인덱스 파라미터 쓰기		
Return Code		

LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
IndexNo	0~63	[0x3100+ indexNo] [USINT]인덱스번호
*pInfo		[SubIndex] [UINT]SubIndex 1 : 인덱스 타입 [DINT]SubIndex2 : 이동거리 [DINT]SubIndex3 : 속도(Velocity) [DINT]SubIndex4 : 가속도(Acceleration) [DINT]SubIndex5 : 감속도(Deceleration) [DINT]SubIndex6 : 레지스트레이션 이동거리 [DINT]SubIndex7 : 레지스트레이션 속도 [UINT]SubIndex8 : 반복 회수 [UINT]SubIndex9 : 대기시간 [UINT]SubIndex10 : 다음 인덱스 [UINT]SubIndex11 : 인덱스 동작 액션
Example		
<pre> /*인덱스 파라미터 읽기*/ /*LCOM.dll에 선언되어 있는 구조체 사용. typedef struct IndexMotionProfileSt {     Int IndexNo;     unsigned short  Type;         long  Dist;         long  Vel;         long  Acc;         long  Dec;         long  RegDist;         long  RegVel;         unsigned short  Repeat;         unsigned short  Dwell;         unsigned short  Next;         unsigned short  Action;     }INDEXMOTION_PROFILE, *PINDEXMOTION_PROFILE; */ </pre>		

```

INDEXMOTION_PROFILE pInfo;
CString s;
int ret,IndexNo;
UpdateData();
// 인덱스 번호 설정_설정에 따라 인덱스 파라미터 업로드.
IndexNo = m_cb_IndexNo.GetCurSel();
if (m_Slave_State[0]) {
    // 인덱스 번호를 기준으로 인덱스 파라미터 read
    ret = LCOM_ReadIndexParameter(m_DriveId[0], IndexNo, &pInfo);
    // 인덱스 파라미터 정보를 콤보컨트롤러 및 에디터컨트롤러에 표시
    if (!ret)
    {
        //Index type
        /*0: Absolute, 1: Relative, 2: Registration Absolute, 3: Registration Relative,
        4: Blending Absolute, 5: Blending Relative, 6: Rotary Absolute, 7: Rotary Relative,
        8: Rotary Shortest, 9: Rotary Positive, 10 :Rotary Negative*/
        m_cb_IndexType.SetCurSel(pInfo.Type);
        m_Index_Dist2 = pInfo.Dist;           //Distance
        m_Index_Vel2 = pInfo.Vel;           //Velocity
        m_Index_Acc2 = pInfo.Acc;           //Acceleration
        m_Index_Dec2 = pInfo.Dec;           //Deceleration
        m_Index_RegDist2 = pInfo.RegDist; //Registration Distance
        m_Index_RegVel2 = pInfo.RegVel; //Registration Velocity
        m_Index_Repeat2 = pInfo.Repeat; //Repeat Count
        m_Index_Dwell2 = pInfo.Dwell; //Dwell Time
        m_cb_Next.SetCurSel(pInfo.Next); //Next Index
        m_cb_Action.SetCurSel(pInfo.Action); //Action/*0: Stop, 1: Wait for Start, 2: Next Index*/
        UpdateData(false);
    }
    // 인덱스 번호를 기준으로 인덱스 파라미터 write
    pInfo.Type = m_cb_IndexType.GetCurSel(); //indexType
    pInfo.Dist = m_Index_Dist;           //Distance
    pInfo.Vel = m_Index_Vel;           //Velocity
    pInfo.Acc = m_Index_Acc;           //Acceleration
    pInfo.Dec = m_Index_Dec;           //Deceleration
    pInfo.RegDist = m_Index_RegDist; //Registration Distance
    pInfo.RegVel = m_Index_RegVel; //Registration Velocity
    pInfo.Repeat = m_Index_Repeat; //Repeat Count
}

```

```

pInfo.Dwell = m_Index_Dwell;           //Dwell Time
pInfo.Next = m_cb_Next.GetCurSel();    //Next Index
pInfo.Action = m_cb_Action.GetCurSel(); //Action/*0: Stop,1: Wait for Start, 2: Next Index*/

ret = LCOM_WriteIndexParameter(m_DriveId[0], IndexNo, &pInfo);

}
    
```

DLLFUNC int __stdcall LCOM_IsEndIndex (unsigned char DriveID, int* State)		
인덱스 action이 Stop혹은 Wait for Start일경우 인덱스 종료후 EOS신호		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 읽기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
State	0~1	[0x2122:01] [UINT]INPOS1
Example		
/*EOS 신호 출력*/		
int ret,State;		
// m_DriveId[0] slave측 EOS bit 상태 반환		
ret = LCOM_IsEndIndex(m_DriveId[0], &State);		

DLLFUNC int __stdcall LCOM_SetInPosition (unsigned char DriveID, unsigned long PosWindow, unsigned long TimeWindow)		
위치 도달범위 및 도달시간 설정		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
PosWindow	0~ 1073741823	[0x6067] [UDINT]위치 도달범위[UU]
TimeWindow	0~65535	[0x6068]

		[UINT]위치 도달시간[ms]
Example		
/*위치 도달범위 및 도달시간 설정*/		
int ret;		
unsigned long PosWindow, TimeWindow;		
PosWindow = 5242880; // 위치 도달 범위		
TimeWindow = 10; // 위치 도달 시간		
//위치 도달범위 및 도달 시간 설정		
ret = LCOM_SetInPosition(m_DriveId[0], PosWindow,TimeWindow);		

<b>DLLFUNC int __stdcall LCOM_IsInPosition (unsigned char DriveID, int* State)</b>		
목표위치 도달 신호 출력		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
State	0~1	[0x2121:04] [UINT]INPOS1
Example		
/*목표위치 도달 신호 출력*/		
int ret,State;		
// m_DriveId[0] slave측 INPOS1 bit 상태 반환		
ret = LCOM_IsInPosition(m_DriveId[0], &State);		

<b>DLLFUNC int __stdcall LCOM_TargerVelocity(unsigned char DriveID, long Vel)</b>		
PV모드 및 CSV모드에서 목표 속도를 설정		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Vel	-2147483647~ 2147483647	[0x60FF] [DINT]목표 속도[U/s]
Example		

```

/*목표 속도 설정*/
int ret;
long vel;
vel = 873800; //목표 속도 설정
// m_DriveId[0] slave축 목표 속도 설정
ret =LCOM_TargerVelocity(m_DriveId[0], Vel);
    
```

DLLFUNC int __stdcall LCOM_ProfileVelocity(unsigned char DriveID, unsigned long Vel)		
프로파일 속도 설정		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Vel	0~ 2147483647	[0x6081] [UDINT]프로파일 속도[UU/s]
Example		
<pre> /*프로파일 속도 설정*/ int ret; unsigned long vel; vel = 873800; //프로파일 속도 설정 // m_DriveId[0] slave축 프로파일 속도설정 ret =LCOM_ProfileVelocity (m_DriveId[0], Vel);                 </pre>		

DLLFUNC int __stdcall LCOM_ProfileAccel(unsigned char DriveID, unsigned long Acc)		
프로파일 가속도 설정		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Acc	0~ 2147483647	[0x6083] [UDINT]프로파일 가속도[UU/s^2]
Example		
<pre> /*프로파일 가속도 설정*/                 </pre>		

```
int ret;
unsigned long Acc;
Acc = 8738000; //프로파일 가속도 설정
// m_DriveId[0] slave축 프로파일 가속도설정
ret =LCOM_ProfileAccel(m_DriveId[0], Acc);
```

DLLFUNC int __stdcall LCOM_ProfileDecel(unsigned char DriveID, unsigned long Dec)		
프로파일 감속도 설정		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Dec	0~ 2147483647	[0x6084] [UDINT]프로파일 감속도[UU/s^2]
Example		
/*프로파일 감속도 설정*/		
int ret;		
unsigned long Dec;		
Dec = 8738000; //프로파일 감속도 설정		
// m_DriveId[0] slave축 프로파일 감속도설정		
ret =LCOM_ProfileDecel(m_DriveId[0], Dec);		

DLLFUNC int __stdcall LCOM_StopDecel(unsigned char DriveID, unsigned long Dec)		
Quick Stop 감속도 설정		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Dec	0~ 2147483647	[0x6085] [UDINT]프로파일 감속도[UU/s^2]
Example		
/*QuickStop 감속도 설정*/		
int ret;		

```

unsigned long Dec;

Dec = 8738000; // 감속도 설정

// m_DriveId[0] slave축 감속도설정

ret =LCOM_StopDecel(m_DriveId[0], Dec);
    
```

**DLLFUNC int \_\_stdcall LCOM\_SetInVelocity (unsigned char DriveID, unsigned long VelWindow, unsigned long TimeWindow)**

속도 도달범위 및 도달시간 설정

Return Code

LCOM_OK	0	에러 없이 통신 성공
---------	---	-------------

파라미터 쓰기와 동일 Return Code

Argument7]

DriveID	1~31	Node ID
VelWindow	0~65535	[0x606D] [UINT]속도 도달범위 [UU/s]
TimeWindow	0~65535	[0x606E] [UINT]속도 도달시간 [ms]

Example

```

/*속도 도달범위 및 도달시간 설정*/
int ret;

unsigned long VelWindow, TimeWindow;

VelWindow = 873800; // 속도 도달 범위
TimeWindow = 10; // 속도 도달 시간

//속도 도달범위 및 도달 시간 설정

ret = LCOM_SetInPosition(m_DriveId[0], VelWindow,TimeWindow);
    
```

**DLLFUNC int \_\_stdcall LCOM\_IsInVelocity(unsigned char DriveID, int\* State)**

목표 속도 도달 완료 신호

Return Code

LCOM_OK	0	에러 없이 통신 성공
---------	---	-------------

파라미터 일기와 동일 Return Code

Argument

DriveID	1~31	Node ID
State	0~1	[0x2121:07] [UINT]INSPD

```

Example
/*속도 위치 도달 신호 출력*/
int ret,State;
// m_DriveId[0] slave축 INSPD bit 상태 반환
ret = LCOM_IsInVelocity(m_DriveId[0], &State);
    
```

**DLLFUNC int \_\_stdcall LCOM\_TargetTorque(unsigned char DriveID, long Torque)**

목표 토크 설정

Return Code

LCOM_OK	0	에러 없이 통신 성공
---------	---	-------------

파라미터 일기와 동일 Return Code

Argument

DriveID	1~31	Node ID
Torque	-5000~5000	[0x6071] [INT]목표 토크[0.1%]

Example

```

/*목표 토크 설정*/
int ret;
long Torque;
Torque = 100; //10%토크 설정
// m_DriveId[0] slave축 토크 설정
ret =LCOM_TargetTorque(m_DriveId[0], Torque);
    
```

**DLLFUNC int \_\_stdcall LCOM\_TorqueVelLimit(unsigned char DriveID, unsigned long TorqueVelLimit)**

토크제어시 속도 제한

Return Code

LCOM_OK	0	에러 없이 통신 성공
---------	---	-------------

파라미터 일기와 동일 Return Code

Argument

DriveID	1~31	Node ID
TorqueVelLimit	0~ 2147483647	[0x607F] [UDINT]속도 제한[UU/s]

Example

```

/*속도 제한 설정*/
int ret;
    
```

```

unsigned long TorqueVelLimit;

TorqueVelLimit = 8738000; //1000rpm 제한
// m_DriveId[0] slave축 속도 제한

ret =LCOM_TorqueVelLimit (m_DriveId[0], TorqueVelLimit);
    
```

DLLFUNC int __stdcall LCOM_TorqueSlope(unsigned char DriveID, unsigned long TorqueSlope)		
토크 기울기 설정		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 일기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
TorqueSlope	0~ 2147483647	[0x6087] [UDINT]토크 기울기[0.1%/s]
Example		
<pre> /*토크 기울기 설정*/ int ret; unsigned long TorqueSlope; TorqueSlope = 1000; //100% // m_DriveId[0] slave축 토크 기울기 설정  ret =LCOM_TorqueVelLimit (m_DriveId[0], TorqueSlope);                     </pre>		

DLLFUNC int __stdcall LCOM_TorqueLimit(unsigned char DriveID, int* State)		
토크 출력 제한 신호		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 일기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
State	0~1	[0x2121:05] [UINT]TLMT
Example		
<pre> /*토크 출력 제한 신호*/ int ret,State;  // m_DriveId[0] slave축 TLMIT bit 상태 반환                     </pre>		

```
ret = LCOM_TorqueLimit (m_DriveId[0], &State);
```

DLLFUNC int __stdcall LCOM_SelVelocityLimit(unsigned char DriveID, unsigned short Select)		
속도 제한 기능 설정		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
Select	0~3	[0x230D][UINT] 0: 제한 속도 값(0x230E)으로 제한 1:모터 최대 속도로 제한
Example		
<pre>/*속도 제한 기능 설정*/ int ret; unsigned short Select; Select = 1 //모터 최대 속도로 제한 // m_DriveId[0] slave축 속도 제한 기능 설정 ret = LCOM_SelVelocityLimit (m_DriveId[0], Select);</pre>		

DLLFUNC int __stdcall LCOM_SetVelocityLimit(unsigned char DriveID, unsigned short velocityLimit)		
토크 제어 시 제한 속도		
Return Code		
LCOM_OK	0	에러 없이 통신 성공
파라미터 쓰기와 동일 Return Code		
Argument		
DriveID	1~31	Node ID
velocityLimit	0~6000	[0x230E] [UINT]제한 속도[rpm]
Example		
<pre>/*토크 제어 시 제한 속도*/ int ret; unsigned short velocityLimit; velocityLimit = 1000 //1000rpm 제한 // m_DriveId[0] slave축 속도 제한 속도 설정</pre>		

```
ret = LCOM_SetVelocityLimit (m_DriveId[0], VelocityLimit);
```

## 3. Preliminary

### 3.1 라이브러리 설정

- Visual C++ 개발 환경에서 프로젝트 생성 후 라이브러리 지정은 LCOM.lib 만 명시적으로 지정
- 프로젝트 폴더 LCOM.lib, API\_EtherCAT.dll, API\_EtherCAT.lib, RMC\_EthercatAPI.h 필요.

#### Example

```
/*라이브러리 정의*/
#include "LCOM.h"
#pragma comment(lib, "LCOM.lib")
```

### 3.2 통신 설정

- 자사 Cofigration Tool 인 Drive\_CM 에서 EoESetup.xml 필요하며 초기 Drive\_CM 에서 Step->Configuration->Ethernet 설정 후 생성된 EoESetup 파일을 사용하여야 IP 가 정상적으로 할당됩니다.
- EhterCAT Of Ethernet(EoE)이 지원되는 Drive 의 경우 네트워크의 연결된 순서와는 별개로 Drive 에 설정되어 있는 NodeID 로 구분을 합니다. EOESetup.xml 에 의해 순차적으로 IP 가 설정되지만 실제적으로 통신을 할 경우는 NodeID 를 사용 합니다.
- iX7NH Drive 의 Ethernet TCP 2 축 이상 연결 시 IP Address 와 MAC Address 값을 다르게 설정 해야 합니다. 또한 Nodeid 스위치의 값이 99 로 설정되어 있어야 Ethernet 통신이 가능합니다.
- L7P Drive 는 RS-485 시리얼 통신만이 지원되며 4 종류의 Baudrate 를 지원[0x3002]
- 초기 설정 값은 57600[bps].
  - 9600[bps]
  - 19200[bps]
  - 38400[bps]

- 57600[bps]
- Drive 와 Application 에서 설정한 Baudrate 가 맞지 않으면 통신이 되지 않습니다.