



SDK-AcapLib2

MIL 접속 설명서

MIL 과 당사 영상입력 라이브러리「AcapLib2」에서 지원하고 있는 영상입력 보드의 접속 방법에 대해 설명합니다. 개발환경은「Visual C++ 6.0」 또는「Visual C++ 2005」입니다.

아래 프로그램에서는 에러처리는 생략합니다.

에러처리에 대해서는 부속된 당사 샘플 소프트를 참조해 주세요.

또한 당사 함수에 있어서 인수의 상세 내용에 대해서는

「SDK-AcapLib2 Library Manual(J20M08015*).pdf」를 참조해 주세요

```

/*--(입력조건)-----
   카메라 입력방식 종류 : Area sensor 카메라, Mono 8bit
   입력 사이즈, X 지연, Y지연 : 640×480, 0, 0
   Time out : 3000ms
   입력 모드 : Free run
   외부 트리거 : 사용하지 않음
   엔코더 : 사용하지 않음
-----*/

```

< 준비 >

사용하는 개발환경에 아래의 링크를 설정해 주세요

[AcapLib2]

AcapLib2.h

C:¥AVALDATA¥SDK-AcapLib2¥Include

AcapLib2.lib

C:¥AVALDATA¥SDK-AcapLib2¥Lib

[MIL]

mil.h

C:¥Program Files¥Matrox Imaging¥mil¥include

mil.lib

C:¥Program Files¥Matrox Imaging¥mil¥library¥winnt¥msc¥dll

```

#include <stdio.h>
#include <conio.h>
#include "AcapLib2.h"
#include "mil.h"

#pragma comment( lib, "AcapLib2.lib" )
#pragma comment( lib, "mil.lib" )

void GetErrorCode( char* cFunctionName );
BOOL ReadIniFile( void );

int     gnChannel=1;
char    gcIniFile[MAX_PATH] = "¥0";

////////////////////////////////////
// 이니셜 파일에서 읽어들임
BOOL ReadIniFile( void )
{
    OPENFILENAME    iniofn;
    iniofn.lStructSize    = sizeof(OPENFILENAME);
    iniofn.hwndOwner    = NULL;
    iniofn.lpstrFilter    = "Initialize File(*.ini)¥0*.ini¥0";
    iniofn.lpstrCustomFilter    = NULL;
    iniofn.nFilterIndex    = 1;
    iniofn.lpstrFile    = gcIniFile;
    iniofn.nMaxFile    = MAX_PATH;
    iniofn.lpstrFileTitle    = NULL;
    iniofn.nMaxFileTitle    = MAX_PATH;
    iniofn.lpstrInitialDir    = NULL;
    iniofn.lpstrTitle    = "The 'Initialize File' of the purpose read is chosen.";
    iniofn.Flags    = OFN_PATHMUSTEXIST | OFN_HIDEREADONLY;
    iniofn.lpstrDefExt    = ".ini";

    //파일 선택화면
    if( GetOpenFileName( &iniofn ) == FALSE )
        return FALSE;

    return( TRUE );
}

int main( int argc, char *argv[] )
{
    ACAPBOARDINFO abi;
    HANDLE hDev;
    DWORD st, et;
    PCHAR pUserBuffer = NULL;
    int nLoop = 1;
    int nData = 0;
    int nDataR = 0, nDataG = 0, nDataB = 0;
    int nXSize = 640, nYSize = 480, nBit = 8;

```

```

// <MIL> 변수 정의
MIL_ID MilApplication = M_NULL;           // Application identifier
MIL_ID MilSystem = M_NULL;               // System Identifier
MIL_ID MilDisplay = M_NULL;              // Display identifier
MIL_ID MilImage = M_NULL;                // Image buffer identifier.

// 채널 선택
if( MessageBox( GetFocus(), "Is Ch1 Used?", "Select Channel", MB_YESNO ) == IDYES )
    gnChannel = 1;
else if( MessageBox( GetFocus(), "Is Ch2 Used?", "Select Channel", MB_YESNO ) == IDYES )
    gnChannel = 2;
else
    return FALSE;

// ini설 파일의 선택
if( ReadIniFile() == FALSE ){
    MessageBox( GetFocus(), "ReadIniFile failed.", NULL, MB_OK );
    return FALSE;
}

// 보드 정보 취득
AcapGetBoardInfo( &abi );

// 보드 오픈
hDev = AcapOpen( abi.BORADINDEX[0].pBoardName, abi.BORADINDEX[0].nBoardID, gnChannel );
if( hDev == INVALID_HANDLE_VALUE ){
    GetErrorCode( "AcapOpen" );
    return 0;
}

// ini파일의 읽어들이기(ini파일명, 파일버스는 임의로 변경해 주시요)
if( AcapSelectFile( hDev, gnChannel, gcIniFile, 0 ) == 0 ){
    GetErrorCode( "AcapSelectFile" );
    return 0;
}

// 입력 사이즈 취득
if( AcapGetInfo( hDev, gnChannel, ACL_X_SIZE, 0, &nXSize ) == 0 ){
    GetErrorCode( "AcapGetInfo" );
    return 0;
}
if( AcapGetInfo( hDev, gnChannel, ACL_Y_SIZE, 0, &nYSize ) == 0 ){
    GetErrorCode( "AcapGetInfo" );
    return 0;
}
if( AcapGetInfo( hDev, gnChannel, ACL_CAM_BIT, 0, &nBit ) == 0 ){
    GetErrorCode( "AcapGetInfo" );
    return 0;
}
}

```

```

// <MIL> MIL 의 초기화
MappAllocDefault( M_SETUP, &MilApplication, &MilSystem, &MilDisplay, M_NULL, M_NULL );
// <MIL> MIL 의 메모리 확보
if( nBit == 8 )
    MbufAlloc2d( MilSystem, nXSize, nYSize, 8+M_UNSIGNED, M_IMAGE+M_DISP, &MilImage);
//8bit mono
else if( nBit == 24 )
    MbufAllocColor( MilSystem, 3, nXSize, nYSize, 8+M_UNSIGNED,
        M_IMAGE+M_DISP+M_PACKED+M_BGR24, &MilImage );           //24bit color
else if( nBit == 32 )
    MbufAllocColor( MilSystem, 3, nXSize, nYSize, 8+M_UNSIGNED,
        M_IMAGE+M_DISP+M_PACKED+M_BGR32, &MilImage );           //32bit color
else
    return 0;
// <MIL> 버퍼를 클리어 함
MbufClear( MilImage, 0 );
// <MIL> 선두 어드레스 취득(MIL)
MbufInquire( MilImage, M_HOST_ADDRESS, &pUserBuffer );
// MIL에서 확보된 버퍼를AcapLib2에 등록
if( AcapSetBufferAddress( hDev, gnChannel, ACL_IMAGE_PTR, 1, pUserBuffer ) == 0 ){
    GetErrorCode( "AcapSetBufferAddress" );
    return 0;
}

// <MIL> MIL 윈도우 표시(MIL)
MdispSelect( MilDisplay, MilImage );

printf( "입력을 중지하기 위해서는 아무키나 입력해 주세요\n" );
do{
    st = GetTickCount();
    // 한개 영상촬영 개시
    if( AcapGrabStart( hDev, gnChannel, 1 ) == 0 ){
        GetErrorCode( "AcapGrabStart" );
        goto _ERROR;
    }
    // 한개 화면 촬영 중지
    if( AcapGrabStop( hDev, gnChannel ) == 0 ){
        GetErrorCode( "AcapGrabStop" );
        goto _ERROR;
    }
}
et = GetTickCount();

// <MIL> MIL 윈도우 표시
if( nBit == 8 )
    MbufPut( MilImage, pUserBuffer );
else if( nBit == 24 )
    MbufPutColor( MilImage, M_PACKED+M_BGR24, M_ALL_BANDS, pUserBuffer );
else if( nBit == 32 )
    MbufPutColor( MilImage, M_PACKED+M_BGR32, M_ALL_BANDS, pUserBuffer );

```

```

// (x, y) = (100, 100)의 데이터를 Dump
if( nBit == 8 ){
    nData = pUserBuffer[100 + 100 * nXSize];
    printf( "영상취득 매수: %4d 데이터(100, 100) : %3d, %dms\n",
            nLoop++, nData, et - st );
}
else if( nBit == 24 ){
    nDataR = pUserBuffer[100 * 3 + 2 + 100 * nXSize * 3];
    nDataG = pUserBuffer[100 * 3 + 1 + 100 * nXSize * 3];
    nDataB = pUserBuffer[100 * 3 + 0 + 100 * nXSize * 3];
    printf( "영상취득 매수: %4d 데이터(100, 100) : (R, G, B) = ( %3d, %3d, %3d), %dms\n",
            nLoop++, nDataR, nDataG, nDataB, et - st );
}
else if( nBit == 32 ){
    nDataR = pUserBuffer[100 * 4 + 2 + 100 * nXSize * 4];
    nDataG = pUserBuffer[100 * 4 + 1 + 100 * nXSize * 4];
    nDataB = pUserBuffer[100 * 4 + 0 + 100 * nXSize * 4];
    printf( "영상취득 매수: %4d 데이터(100, 100) : (R, G, B) = ( %3d, %3d, %3d), %dms\n",
            nLoop++, nDataR, nDataG, nDataB, et - st );
}
}while( !_kbhit() );

printf( "입력을 중지합니다. \n" );

_ERROR:
// 보드 Close
AcapClose( hDev, gnChannel );
// <MIL> 어플리케이션 개방
MappFreeDefault( MilApplication, MilSystem, MilDisplay, M_NULL, MilImage );

return 0;
}

void GetErrorCode( char* cFunctionName )
{
    ACAPERRORINFO aei;
    char errmsg[64] = "¥0";

    AcapGetLastErrorCode( &aei, FALSE );
    sprintf( errmsg, "%s\nCommon = 0x%04X\nBoard = 0x%04X\nExtend = 0x%04X\n",
            cFunctionName, aei.dwCommonErrorCode, aei.dwBoardErrorCode, aei.dwExtendErrorCode );
    MessageBox( NULL, errmsg, "Error", MB_OK | MB_ICONSTOP );
}

```

이상