A-821 PGL/H

Software Manual
[For Windows 2000]

Warranty

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for damage consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, ICP DAS assumes no responsibility for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

The names used for identification only maybe registered trademarks of their respective companies.

License

The user can use, modify and backup this software on a single machine**.** The user may not reproduce, transfer or distribute this software, or any copy, in whole or in part.

Table of Contents

1. DECLARATION FILES

Please refer to user manual "CallDLL.pdf".

For Windows 2000:

```
|--\Driver
    |--\A821.DLL       ← Dynamic Linking Library
    |--\A821.sys       ← device driver
    |--\Napwnt.sys     ← device driver
    |
    |--\BCB            ← For Borland C++ Builder
    |    |--\A821.H    ← Header file
    |    |--\A821.Lib  ← Import Library for BCB only
    |
    |--\Delphi         ← For Delphi
    |    |--\A821.pas  ← Declaration file
    |
    |--\VB             ← For Visual Basic
    |    |--\A821.bas  ← Declaration file
    |
    |--\VC             ← For Visual C++
         |--\A821.H    ← Header file
         |--\A821.Lib  ← Import Library for VC only
```

1.1 A821.H

```
#ifdef __cplusplus
    #define EXPORTS extern "C" __declspec (dllimport)
#else
    #define EXPORTS
#endif

/***************** DEFINE A821 RELATIVE ADDRESS *****************/
#define   A821_TIMER0          0x00
#define   A821_TIMER1          0x01
#define   A821_TIMER2          0x02
#define   A821_TIMER_MODE      0x03
#define   A821_AD_LO           0x04    /* Analog to Digital, Low Byte */
#define   A821_AD_HI           0x05    /* Analog to Digital, High Byte */
#define   A821_DA_CH0_LO       0x04    /* Digit to Analog, CH 0 */
#define   A821_DA_CH0_HI       0x05
#define   A821_DI_LO           0x06    /* Digit Input */
#define   A821_DO_LO           0x0D    /* Digit Output */

#define   A821_CLEAR_IRQ       0x08
#define   A821_SET_GAIN        0x09
#define   A821_SET_CH          0x0A
#define   A821_SET_MODE        0x0B
#define   A821_SOFT_TRIG       0x0C

#define   A821_POLLING_MODE         1

/*** define the gain mode ***/
#define   A821_BI_1            0
#define   A821_BI_10           1
#define   A821_BI_100          2
#define   A821_BI_1000         3
#define   A821_BI_2            1
#define   A821_BI_4            2
#define   A821_BI_8            3

#define   A821PGL              0
#define   A821PGH              1
```

```
#define   A821_NoError                     0
#define   A821_DriverOpenError             1
#define   A821_DriverNoOpen                2
#define   A821_GetDriverVersionError       3
#define   A821_InstallIrqError             4
#define   A821_ClearIntCountError          5
#define   A821_GetIntCountError            6
#define   A821_GetBufferError              7
#define   A821_InstallBufError             10
#define   A821_AllocateMemoryError         11
#define   A821_CardTypeError               12
#define   A821_TimeoutError                13
#define   A821_OtherError                  14
#define   A821_ExceedBoardNumber           15
#define   A821_CardNotFound                16
#define   A821_GetTotalBoardError          17
#define   A821_ChannelNoError              18
#define   A821_IntStopError                19
#define   A821_IntInstallEventError        20
#define   A821_GetConfigError              21
#define   A821_ActiveBoardError            22
#define   A821_ConfigCodeError             23
#define   A821_BufferFull                  24
#define   A821_NoChannelToScan             25
#define   A821_IntInstallChannelError      26
#define   A821_IntInstallConfigError       27


// Functions of Test
EXPORTS short CALLBACK A821_SHORT_SUB_2(short nA, short nB);
EXPORTS float CALLBACK A821_FLOAT_SUB_2(float fA, float fB);
EXPORTS WORD  CALLBACK A821_Get_DLL_Version(void);
EXPORTS WORD  CALLBACK A821_GetDriverVersion(WORD
*wDriverVersion);

// Functions of DI/DO
EXPORTS WORD  CALLBACK A821_DI(WORD *wInVal);
EXPORTS WORD  CALLBACK A821_DO(WORD wHexValue);
```

```
// Functions of AD/DA
EXPORTS WORD  CALLBACK A821_SetChGain
        (WORD wChannel, WORD wConfig, WORD wCardType);
EXPORTS WORD  CALLBACK A821_Fast_AD_Hex(WORD *wVal);
EXPORTS WORD  CALLBACK A821_Fast_AD_Float(float *fVal);
EXPORTS WORD  CALLBACK A821_AD_Hex
        (WORD wChannel, WORD wConfig, WORD wCardType, WORD *wVal);
EXPORTS WORD  CALLBACK A821_AD_Float
        (WORD wChannel, WORD wConfig, WORD wCardType, float *fVal);
EXPORTS WORD  CALLBACK A821_ADs_Hex( WORD wBuf[], WORD wCount
);
EXPORTS WORD  CALLBACK A821_ADs_Float( float fBuf[], WORD wCount );
EXPORTS WORD  CALLBACK A821_Hex2Float
        (WORD wConfig, WORD wCardType, WORD wHex, float *fVal);

EXPORTS WORD  CALLBACK A821_DA_Hex(WORD wHexValue);
EXPORTS WORD  CALLBACK A821_DA_Uni5(float fValue);
EXPORTS WORD  CALLBACK A821_DA_Uni10(float fValue);

// Functions of Driver
EXPORTS WORD  CALLBACK A821_DriverInit(WORD *wTotalBoards);
EXPORTS void  CALLBACK A821_DriverClose(void);
EXPORTS WORD  CALLBACK A821_DELAY(WORD wDownCount);
EXPORTS WORD  CALLBACK A821_Check_Address(void);
EXPORTS WORD  CALLBACK A821_GetConfigAddress
        (WORD *wAddrBase, WORD *wCurrentBoard);
EXPORTS WORD  CALLBACK A821_ActiveBoard( WORD wBoardNo );

EXPORTS void  CALLBACK A821_OutputByte
        (WORD wPortAddr, UCHAR bOutputVal);
EXPORTS void  CALLBACK A821_OutputWord
        (WORD wPortAddr, WORD wOutputVal);
EXPORTS WORD  CALLBACK A821_InputByte(WORD wPortAddr);
EXPORTS WORD  CALLBACK A821_InputWord(WORD wPortAddr);
```

```
// Functions of Interrupt
EXPORTS WORD  CALLBACK A821_Int_Install
        (HANDLE *hEvent, DWORD dwCount);
EXPORTS WORD  CALLBACK A821_Int_Start(WORD c1, WORD c2);
EXPORTS WORD  CALLBACK A821_Int_Stop(void);
EXPORTS WORD  CALLBACK A821_Int_Remove(void);
EXPORTS WORD  CALLBACK A821_Int_GetCount(DWORD *dwVal);
EXPORTS WORD  CALLBACK A821_Int_GetHexBuf
        (WORD wBuf[], DWORD dwNum );
EXPORTS WORD  CALLBACK A821_Int_GetFloatBuf
        (float fBuf[], DWORD dwNum );

// Functions of Channel Scan
EXPORTS void  CALLBACK A821_ChScan_Clear(void);
EXPORTS WORD  CALLBACK A821_ChScan_Add
        (WORD wChannel, WORD wConfig);
EXPORTS WORD  CALLBACK A821_ChScan_Set
        (WORD wChannel[], WORD wConfig[], WORD wChNum);
EXPORTS WORD  CALLBACK A821_ChScan_PollingHex
        (WORD wCardType, WORD wBuf[], WORD wNumPerCh);
EXPORTS WORD  CALLBACK A821_ChScan_PollingFloat
        (WORD wCardType, float fBuf[], WORD wNumPerCh);

// Functions of Channel Scan for Interrupt Only
EXPORTS WORD  CALLBACK A821_ChScan_IntInstall
        (HANDLE *hEvent, DWORD dwNumPerCh);
EXPORTS WORD  CALLBACK A821_ChScan_IntStart
        (WORD c1, WORD c2, WORD wCardType);
EXPORTS WORD  CALLBACK A821_ChScan_IntGetCount(DWORD *dwVal);
EXPORTS WORD  CALLBACK A821_ChScan_IntGetHexBuf(WORD wBuf[]);
EXPORTS WORD  CALLBACK A821_ChScan_IntGetFloatBuf(float fBuf[]);
EXPORTS WORD  CALLBACK A821_ChScan_IntStop(void);
EXPORTS WORD  CALLBACK A821_ChScan_IntRemove(void);
```

1.2 A821.BAS

```
Attribute VB_Name = "A821"
'*********************************************************************************
'     The Declare of A821.DLL for A821 DAQ Card
'*********************************************************************************

Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

'***************** DEFINE A821 RELATIVE ADDRESS *****************/
Global Const A821_TIMER0       = &H0
Global Const A821_Timer1       = &H1
Global Const A821_TIMER2       = &H2
Global Const A821_TIMER_MODE = &H3
Global Const A821_AD_LO        = &H4          '* Analog to Digital, Low Byte */
Global Const A821_AD_HI        = &H5          '* Analog to Digital, High Byte */
Global Const A821_DA_CH0_LO = &H4        '* Digit to Analog, CH 0 */
Global Const A821_DA_CH0_HI  = &H5
Global Const A821_DI_LO        = &H6          '* Digit Input */
Global Const A821_DO_LO        = &HD          '* Digit Output */

Global Const A821_CLEAR_IRQ = &H8
Global Const A821_SET_GAIN   = &H9
Global Const A821_SET_CH       = &HA
Global Const A821_SET_MODE  = &HB
Global Const A821_SOFT_TRIG  = &HC

Global Const A821_POLLING_MODE              = 1

'*** define the gain mode ***/
Global Const A821_BI_1          = 0
Global Const A821_BI_10         = 1
Global Const A821_BI_100        = 2
Global Const A821_BI_1000       = 3
Global Const A821_BI_2          = 1
Global Const A821_BI_4          = 2
Global Const A821_BI_8          = 3

Global Const A821PGL            = 0
Global Const A821PGH            = 1
Global Const A821_NoError                    = 0
Global Const A821_DriverOpenError            = 1
Global Const A821_DriverNoOpen               = 2
Global Const A821_GetDriverVersionError      = 3
Global Const A821_InstallIrqError            = 4
Global Const A821_ClearIntCountError         = 5
Global Const A821_GetIntCountError           = 6
Global Const A821_GetBufferError             = 7
Global Const A821_InstallBufError            = 10
Global Const A821_AllocateMemoryError        = 11
Global Const A821_CardTypeError              = 12
Global Const A821_TimeoutError               = 13
Global Const A821_OtherError                 = 14
Global Const A821_ExceedBoardNumber          = 15
Global Const A821_CardNotFound               = 16
Global Const A821_GetTotalBoardError         = 17
Global Const A821_ChannelNoError             = 18
Global Const A821_IntStopError               = 19
Global Const A821_IntInstallEventError       = 20
Global Const A821_GetConfigError             = 21
Global Const A821_ActiveBoardError           = 22
Global Const A821_ConfigCodeError            = 23
Global Const A821_BufferFull                 = 24
Global Const A821_NoChannelToScan            = 25
Global Const A821_IntInstallChannelError     = 26
Global Const A821_IntInstallConfigError      = 27

'******* Test Functions ***********
Declare Function A821_SHORT_SUB_2 Lib "A821.DLL" _
        (ByVal nA As Integer, ByVal nB As Integer) As Integer
Declare Function A821_FLOAT_SUB_2 Lib "A821.DLL" _
        (ByVal fA As Single, ByVal fB As Single) As Single
Declare Function A821_Get_DLL_Version Lib "A821.DLL" () As Integer
Declare Function A821_GetDriverVersion Lib "A821.DLL" _
        (wDriverVersion As Integer) As Integer

'******* DI/DO Functions ***********
Declare Function A821_DI Lib "A821.DLL" _
        (wInVal As Integer) As Integer
Declare Function A821_DO Lib "A821.DLL" _
        (ByVal wHexValue As Integer) As Integer
```

```
'******* AD/DA Functions ***********
Declare Function A821_SetChGain Lib "A821.DLL" _
        (ByVal wChannel As Integer, ByVal wConfig As Integer, _
        ByVal wCardType As Integer) As Integer
Declare Function A821_Fast_AD_Hex Lib "A821.DLL" _
        (wVal As Integer) As Integer
Declare Function A821_Fast_AD_Float Lib "A821.DLL" _
        (fVal As Single) As Integer
Declare Function A821_AD_Hex Lib "A821.DLL" _
        (ByVal wChannel As Integer, ByVal wConfig As Integer, _
        ByVal wCardType As Integer, wVal As Integer) As Integer
Declare Function A821_AD_Float Lib "A821.DLL" _
        (ByVal wChannel As Integer, ByVal wConfig As Integer, _
        ByVal wCardType As Integer, fVal As Single) As Integer
Declare Function A821_ADs_Hex Lib "A821.DLL" _
        (wBuf As Integer, ByVal wCount As Integer) As Integer
Declare Function A821_ADs_Float Lib "A821.DLL" _
        (fbuf As Single, ByVal wCount As Integer) As Integer
Declare Function A821_Hex2Float Lib "A821.DLL" _
        (ByVal wConfig As Integer, ByVal wCardType As Integer, _
        ByVal wVal As Integer, fVal As Single) As Integer

Declare Function A821_DA_Hex Lib "A821.DLL" _
        (ByVal wHexValue As Integer) As Integer
Declare Function A821_DA_Uni5 Lib "A821.DLL" _
        (ByVal fValue As Single) As Integer
Declare Function A821_DA_Uni10 Lib "A821.DLL" _
        (ByVal fValue As Single) As Integer


'******* Driver Functions ***********
Declare Function A821_DriverInit Lib "A821.DLL" _
        (wTotalBoards As Integer) As Integer
Declare Sub A821_DriverClose Lib "A821.DLL" ()
Declare Function A821_DELAY Lib "A821.DLL" _
        (ByVal wDownCount As Integer) As Integer
Declare Function A821_Check_Address Lib "A821.DLL" () As Integer
Declare Function A821_GetConfigAddress Lib "A821.DLL" _
        (wAddrBase As Integer, wCurrentBoard As Integer) As Integer
Declare Function A821_ActiveBoard Lib "A821.DLL" _
        (ByVal wBoardNo As Integer) As Integer
```

```
Declare Sub A821_OutputByte Lib "A821.DLL" _
        (ByVal wPortAddr As Integer, ByVal bOutputVal As Byte)
Declare Sub A821_OutputWord Lib "A821.DLL" _
        (ByVal wPortAddr As Integer, ByVal wOutputVal As Integer)
Declare Function A821_InputByte Lib "A821.DLL" _
        (ByVal wPortAddr As Integer) As Integer
Declare Function A821_InputWord Lib "A821.DLL" _
        (ByVal wPortAddr As Integer) As Integer


'******* IRQ Functions ***********
Declare Function A821_Int_Install Lib "A821.DLL" _
        (hEvent As Long, ByVal dwCount As Integer) As Integer
Declare Function A821_Int_Start Lib "A821.DLL" _
        (ByVal c1 As Integer, ByVal c2 As Integer) As Integer
Declare Function A821_Int_Stop Lib "A821.DLL" () As Integer
Declare Function A821_Int_Remove Lib "A821.DLL" () As Integer
Declare Function A821_Int_GetCount Lib "A821.DLL" _
        (dwVal As Long) As Integer
Declare Function A821_Int_GetHexBuf Lib "A821.DLL" _
        (wBuffer As Integer, ByVal dwNum As Long) As Integer
Declare Function A821_Int_GetFloatBuf Lib "A821.DLL" _
        (fbuffer As Single, ByVal dwNum As Integer) As Integer

' Functions of Channel Scan
Declare Sub A821_ChScan_Clear Lib "A821.DLL" ()
Declare Function A821_ChScan_Add Lib "A821.DLL" _
        (ByVal wChannel As Integer, ByVal wConfig As Integer) _
        As Integer
Declare Function A821_ChScan_Set Lib "A821.DLL" _
        (wChannel As Integer, wConfig As Integer, _
         ByVal wChNum As Integer) As Integer
Declare Function A821_ChScan_PollingHex Lib "A821.DLL" _
        (ByVal wCardType As Integer, wBuf as Integer, _
        ByVal wNumPerCh As Integer) As Integer
Declare Function A821_ChScan_PollingFloat Lib "A821.DLL" _
        (ByVal wCardType As Integer, fBuf as Single, _
        ByVal wNumPerCh As Integer) As Integer
```

```
' Functions of Channel Scan for Interrupt Only
Declare Function A821_ChScan_IntInstall Lib "A821.DLL" _
        (hEvent As Long, ByVal dwNumPerCh as Long) As Integer
Declare Function A821_ChScan_IntStart Lib "A821.DLL" _
        (ByVal c1 As Integer, ByVal c2 As Integer, _
        ByVal wCardType As Integer) As Integer
Declare Function A821_ChScan_IntGetCount Lib "A821.DLL" _
        (dwVal As Long) As Integer
Declare Function A821_ChScan_IntGetHexBuf Lib "A821.DLL" _
        (wBuf As Integer) As Integer
Declare Function A821_ChScan_IntGetFloatBuf Lib "A821.DLL" _
        (fBuf As Single) As Integer
Declare Function A821_ChScan_IntStop Lib "A821.DLL" () As Integer
Declare Function A821_ChScan_IntRemove Lib "A821.DLL" () As Integer
```

1.3 A821.PAS

```
unit A821;

interface

type PSingle=^Single;
     PWord=^Word;
     PInteger=^Integer;

Const
//***************** DEFINE A821 RELATIVE ADDRESS *****************/
 A821_TIMER0              =  $00;
 A821_TIMER1              =  $01;
 A821_TIMER2              =  $02;
 A821_TIMER_MODE          =  $03;
 A821_AD_LO               =  $04;    //* Analog to Digital, Low Byte */
 A821_AD_HI               =  $05;    //* Analog to Digital, High Byte */
 A821_DA_CH0_LO           =  $04;    //* Digit to Analog, CH 0 */
 A821_DA_CH0_HI           =  $05;
 A821_DI_LO               =  $06;    //* Digit Input */
 A821_DO_LO               =  $0D;    //* Digit Output */

 A821_CLEAR_IRQ           =  $08;
 A821_SET_GAIN            =  $09;
 A821_SET_CH              =  $0A;
 A821_SET_MODE            =  $0B;
 A821_SOFT_TRIG           =  $0C;

 A821_POLLING_MODE              =  1;

//*** define the gain mode ***/
 A821_BI_1           =  0;
 A821_BI_10          =  1;
 A821_BI_100         =  2;
 A821_BI_1000        =  3;
 A821_BI_2           =  1;
 A821_BI_4           =  2;
 A821_BI_8           =  3;

 A821PGL             =  0;
 A821PGH             =  1;
 A821_NoError                    = 0;
 A821_DriverOpenError            = 1;
 A821_DriverNoOpen               = 2;
 A821_GetDriverVersionError      = 3;
 A821_InstallIrqError            = 4;
 A821_ClearIntCountError         = 5;
 A821_GetIntCountError           = 6;
 A821_GetBufferError             = 7;
 A821_InstallBufError            = 10;
 A821_AllocateMemoryError        = 11;
 A821_CardTypeError              = 12;
 A821_TimeoutError               = 13;
 A821_OtherError                 = 14;
 A821_ExceedBoardNumber          = 15;
 A821_CardNotFound               = 16;
 A821_GetTotalBoardError         = 17;
 A821_ChannelNoError             = 18;
 A821_IntStopError               = 19;
 A821_IntInstallEventError       = 20;
 A821_GetConfigError             = 21;
 A821_ActiveBoardError           = 22;
 A821_ConfigCodeError            = 23;
 A821_BufferFull                 = 24;
 A821_NoChannelToScan            = 25;
 A821_IntInstallChannelError     = 26;
 A821_IntInstallConfigError      = 27;

// Function of Test
Function  A821_SHORT_SUB_2(nA, nB : SmallInt):SmallInt; StdCall;
Function  A821_FLOAT_SUB_2(fA, fB : Single):Single; StdCall;
Function  A821_Get_DLL_Version:WORD; StdCall;
Function  A821_GetDriverVersion(var wDriverVersion:WORD):Word; StdCall;

// Function of DI/DO
Function  A821_DO(wHexValue:Word):Word; StdCall;
Function  A821_DI(var wInVal:Word):Word; StdCall;
```

```
// Function of AD/DA
Function  A821_SetChGain(wChannel,wConfig,wCardType:WORD):Word;
StdCall;
Function  A821_Fast_AD_Hex(var wVal:WORD):Word; StdCall;
Function  A821_Fast_AD_Float(var fVal:Single):Word; StdCall;
Function  A821_AD_Hex
        (wChannel,wConfig,wCardType:WORD; var wVal:Word):Word; StdCall;
Function  A821_AD_Float
        (wChannel,wConfig,wCardType:WORD; var fVal:Single):Word; StdCall;
Function  A821_ADs_Hex( wBuf:PWord; wCount:WORD):WORD; StdCall;
Function  A821_ADs_Float(fBuf:PSingle; wCount:WORD):WORD; StdCall;
Function  A821_Hex2Float
        ( wConfig, wCardType:Word; wVal:Word; var fVal:Single ):WORD; StdCall;

Function  A821_DA_Hex(wHexValue:WORD):WORD; StdCall;
Function  A821_DA_Uni5(fValue:Single):WORD; StdCall;
Function  A821_DA_Uni10(Word;fValue:Single):WORD; StdCall;

// Function of Driver
Function  A821_DriverInit(var wTotalBoards:WORD):WORD; StdCall;
Procedure A821_DriverClose; StdCall;
Function  A821_DELAY(wDownCount:WORD):WORD; StdCall;
Function  A821_Check_Address:WORD; StdCall;
Function  A821_GetConfigAddress
        (var wAddrBase:WORD; var wCurrentBoard:WORD):WORD; StdCall;
Function  A821_ActiveBoard(wBoardNo:WORD):WORD; StdCall;

Procedure A821_OutputByte(wPortAddr:WORD; bOutputVal:Byte); StdCall;
Procedure A821_OutputWord(wPortAddr:WORD; wOutputVal:WORD); StdCall;
Function  A821_InputByte(wPortAddr:WORD):WORD; StdCall;
Function  A821_InputWord(wPortAddr:WORD):WORD; StdCall;

// Function of Interrupt
Function  A821_Int_Install(var hEvent:LongInt; dwCount:LongInt):WORD;
StdCall;
Function  A821_Int_Start(c1,c2:WORD):WORD; StdCall;
Function  A821_Int_Stop:WORD; StdCall;
Function  A821_Int_Remove:WORD; StdCall;
Function  A821_Int_GetCount(var dwVal:LongInt):WORD; StdCall;
Function  A821_Int_GetHexBuf(wBuf:PWORD; dwNum:LongInt):WORD; StdCall;
Function  A821_Int_GetFloatBuf(fBuf:PSingle; dwNum:LongInt):WORD; StdCall;
```

```
// Functions of Channel Scan
Procedure A821_ChScan_Clear; StdCall;
Function A821_ChScan_Add
        (wChannel:WORD; wConfig:WORD):WORD; StdCall;
function A821_ChScan_Set(wChannel:PWORD; wConfig:PWORD;
               wChNum:WORD):WORD; StdCall;
Function A821_ChScan_PollingHex
        (wCardType:WORD; wBuf:PWORD; wNumPerCh:WORD):WORD; StdCall;
Function A821_ChScan_PollingFloat
        (wCardType:WORD; fBuf:PSingle; wNumPerCh:WORD):WORD; StdCall;

// Functions of Channel Scan for Interrupt Only
Function A821_ChScan_IntInstall
        (var hEvent:LongInt; dwNumPerCh:LongInt):WORD; StdCall;
Function A821_ChScan_IntStart
        (c1:WORD; c2:WORD; wCardType:WORD):WORD; StdCall;
Function A821_ChScan_IntGetCount(var dwVal:LongInt):WORD; StdCall;
Function A821_ChScan_IntGetHexBuf(wBuf:PWORD):WORD; StdCall;
Function A821_ChScan_IntGetFloatBuf(fBuf:PSingle):WORD; StdCall;
Function A821_ChScan_IntStop:WORD; StdCall;
Function A821_ChScan_IntRemove:WORD; StdCall;

implementation

Function  A821_SHORT_SUB_2;                external 'A821.DLL' name
           'A821_SHORT_SUB_2';
Function  A821_FLOAT_SUB_2;                external 'A821.DLL' name
           'A821_FLOAT_SUB_2';
Function  A821_Get_DLL_Version;            external 'A821.DLL' name
           'A821_Get_DLL_Version';
Function  A821_GetDriverVersion;           external 'A821.DLL' name
           'A821_GetDriverVersion';

Function  A821_DO;                         external 'A821.DLL' name
           'A821_DO';
Function  A821_DI;                         external 'A821.DLL' name
           'A821_DI';
```

```
Function  A821_SetChGain;               external 'A821.DLL' name
        'A821_SetChGain';
Function  A821_Fast_AD_Hex;             external 'A821.DLL' name
        'A821_Fast_AD_Hex';
Function  A821_Fast_AD_Float;           external 'A821.DLL' name
        'A821_Fast_AD_Float';
Function  A821_AD_Hex;                  external 'A821.DLL' name
        'A821_AD_Hex';
Function  A821_AD_Float;                external 'A821.DLL' name
        'A821_AD_Float';
Function  A821_ADs_Hex;                 external 'A821.DLL' name
        'A821_ADs_Hex';
Function  A821_ADs_Float;               external 'A821.DLL' name
        'A821_ADs_Float';
Function  A821_Hex2Float;               external 'A821.DLL' name
        'A821_Hex2Float';

Function  A821_DA_Hex;                  external 'A821.DLL' name
        'A821_DA_Hex';
Function  A821_DA_Uni5;                 external 'A821.DLL' name
        'A821_DA_Uni5';
Function  A821_DA_Uni10;                external 'A821.DLL' name
        'A821_DA_Uni10';

Function  A821_DriverInit;              external 'A821.DLL' name
        'A821_DriverInit';
Procedure A821_DriverClose;             external 'A821.DLL' name
        'A821_DriverClose';
Function  A821_DELAY;                   external 'A821.DLL' name
        'A821_DELAY';
Function  A821_Check_Address;           external 'A821.DLL' name
        'A821_Check_Address';
Function  A821_GetConfigAddress;        external 'A821.DLL' name
        'A821_GetConfigAddress';
Function  A821_ActiveBoard;             external 'A821.DLL' name
        'A821_ActiveBoard';
```

```
Procedure A821_OutputByte;                    external 'A821.DLL' name
        'A821_OutputByte';
Procedure A821_OutputWord;                    external 'A821.DLL' name
        'A821_OutputWord';
Function  A821_InputByte;                     external 'A821.DLL' name
        'A821_InputByte';
Function  A821_InputWord;                     external 'A821.DLL' name
        'A821_InputWord';

Function  A821_Int_Install;                   external 'A821.DLL' name
        'A821_Int_Install';
Function  A821_Int_Start;                     external 'A821.DLL' name
        'A821_Int_Start';
Function  A821_Int_Stop;                      external 'A821.DLL' name
        'A821_Int_Stop';
Function  A821_Int_Remove;                    external 'A821.DLL' name
        'A821_Int_Remove';
Function  A821_Int_GetCount;                  external 'A821.DLL' name
        'A821_Int_GetCount';
Function  A821_Int_GetHexBuf;                 external 'A821.DLL' name
        'A821_Int_GetHexBuf';
Function  A821_Int_GetFloatBuf;               external 'A821.DLL' name
        'A821_Int_GetFloatBuf';

// Functions of Channel Scan
Procedure A821_ChScan_Clear;                  external 'A821.DLL' name
        'A821_ChScan_Clear';
Function  A821_ChScan_Add;                    external 'A821.DLL' name
        'A821_ChScan_Add';
function  A821_ChScan_Set;                    external 'A821.DLL' name
        'A821_ChScan_Set';
Function  A821_ChScan_PollingHex;             external 'A821.DLL' name
        'A821_ChScan_PollingHex';
Function  A821_ChScan_PollingFloat;           external 'A821.DLL' name
        'A821_ChScan_PollingFloat';
```

```
// Functions of Channel Scan for Interrupt Only
Function A821_ChScan_IntInstall;          external 'A821.DLL' name
        'A821_ChScan_IntInstall';
Function A821_ChScan_IntStart;            external 'A821.DLL' name
        'A821_ChScan_IntStart';
Function A821_ChScan_IntGetCount;         external 'A821.DLL' name
        'A821_ChScan_IntGetCount';
Function A821_ChScan_IntGetHexBuf;        external 'A821.DLL' name
        'A821_ChScan_IntGetHexBuf';
Function A821_ChScan_IntGetFloatBuf;      external 'A821.DLL' name
        'A821_ChScan_IntGetFloatBuf';
Function A821_ChScan_IntStop;             external 'A821.DLL' name
        'A821_ChScan_IntStop';
Function A821_ChScan_IntRemove;           external 'A821.DLL' name
        'A821_ChScan_IntRemove';

end.
```

2. REFERENCE

2.1 RANGE CONFIGURATION

The AD converter of A821PGH/L is 12 bits under all configuration code. If the analog input range is configured to +/- 5V range, the resolution of one bit is equal to 2.44 mV. If the analog input range is configured to +/- 2.5V range, the resolution will be 1.22 mV. If the analog input signal is about 1 V, use configuration 0/1/2 (for A821PGL) will get nearly the same result **except resolution. So choose the correct configuration code can achieve the highest precision measurement.**

A-821PGL Input Signal Range Configuration Code Table

Bipolar/Unipolar	Input Signal Range	Configuration Code
Bipolar	**+/- 5V**	**0**
Bipolar	**+/- 2.5V**	**1**
Bipolar	**+/- 1.25V**	**2**
Bipolar	**+/- 0.0625V**	**3**

A-821PGH Input Signal Range Configuration Code Table

Bipolar/Unipolar	Input Signal Range	Configuration Code
Bipolar	**+/- 5V**	**0**
Bipolar	**+/- 0.5V**	**1**
Bipolar	**+/- 0.05V**	**2**
Bipolar	**+/- 0.005V**	**3**

2.2 ERROR CODE

Error Code	Description
A821_NoError	OK
A821_DriverOpenError	Please check if the driver installed correctly.
A821_DriverNoOpen	Please call the driver initial function to open the driver firstly.
A821_GetDriverVersionError	Can't call the device driver function. Please call the driver initial function to open the driver firstly.
A821_InstallIrqError	1. Please call the driver initial function to open the driver firstly. 2. Please check the resource that does not conflicted with other device.
A821_ClearIntCountError	Can't call the device driver function. Please call the driver initial function to open the driver firstly.
A821_GetIntCountError	Can't call the device driver function. Please call the driver initial function to open the driver firstly.
A821_GetBufferError	Can't call the device driver function. Please call the driver initial function to open the driver firstly.
A821_AllocateMemoryError	Can't allocate the memory for buffer. Please check your system's resource.
A821_CardTypeError	Valid range is : 0 to 1 (A-821L or A-821H)
A821_TimeoutError	ADC timeout error
A821_OtherError	Unknown error
A821_ExceedBoardNumber	The driver initial function will returns number of total boards. The board number to be active must less then this number. For example: if the driver initial function returns 3, and the valid range is 0 to 2.

A821_CardNotFound	The card not found by the device driver. Please check your hardware and driver settings.
A821_ChannelNoError	The valid AD channel no range is 0 to 15. The valid DA channel no range is 0 to 1.
A821_IntStopError	Can’t call the device driver function. Please call the driver initial function to open the driver firstly.
A821_IntInstallEventError	Can’t call the device driver function. Please call the driver initial function to open the driver firstly.
A821_GetConfigError	1. Can’t call the device driver function. Please call the driver initial function to open the driver firstly. 2. Call the “ActiveBoard” function firstly.
A821_ActiveBoardError	1. Please call the driver initial function to open the driver firstly. 2. The driver initial function will returns number of total boards. The board number to be active must less then this number.
A821_ConfigCodeError	Please refer to “Section 2.1 Range Configuration Code” for the valid configuration code.
A821_BufferFull	The buffer for the Channel Scan had full. The max number of buffer is 100 channels to scan. Please use the A821_ChScan_Clear() function to clear the buffer.
A821_NoChannelToScan	The user has to setting the channels to scan. Please use the A821_ChScan_Add() function to add channel and configuration-code to buffer for the Channel Scan.
A821_IntInstallChannelError	Fail to install the list for the Channel Scan into the interrupt service routine. Please check your system’s resource.
A821_IntInstallConfigError	Fail to install the list for the Channel Scan into the interrupt service routine. Please check your system’s resource.

2.3 OTHER MANUALS

Please refer to the following user manuals:

- **SoftInst.pdf:**
  Install the software package under Windows 95/98/NT/2000.

- **CallDll.pdf:**
  Include the declaration files and call the DLL functions with VC++5, VB5, Delphi3 and Borland C++ Builder 3.

- **ResCheck.pdf:**
  Check the resources I/O Port address, IRQ number and DMA number for add-on cards under Windows 95/98/NT/2000.

- **PnPInstall.pdf:**
  Install the Plug and Play information file (*.inf) under Windows 95/98/2000.

3. FUNCTION DESCRIPTION

These function in DLL are divided into several groups as following:

1. The test functions
2. The DI/O functions
3. The AD/DA fixed-mode functions
4. The Driver functions
5. The AD Interrupt Mode functions
6. The AD, Channel Scan functions
7. The AD Interrupt, Channel Scan functions

The functions of test listing as follows:

1. A821_SHORT_SUB_2
2. A821_FLOAT_SUB_2
3. A821_Get_DLL_Version
4. A821_GetDriverVersion

The functions of DI/O listing as follows:

1. A821_DI
2. A821_DO
3. A821_InputByte
4. A821_InputWord
5. A821_OutputByte
6. A821_OutputWord

The functions of AD/DA listing as follows:

1. A821_SetChGain
2. A821_Fast_AD_Hex
3. A821_Fast_AD_Float
4. A821_AD_Hex
5. A821_AD_Float
6. A821_ADs_Hex
7. A821_ADs_Float
8. A821_Hex2Float
9. A821_DA_Hex
10. A821 _DA_Uni5
11. A821 _DA_Uni10

The functions of Driver listing as follows:

1. A821_DriverInit
2. A821_DriverClose
3. A821_DELAY
4. A821_Check_Address
5. A821_GetConfigAddress
6. A821_ActiveBoard

The functions of AD Interrupt listing as follows:

1. A821_Int_Install
2. A821_Int_Start
3. A821_Int_Stop
4. A821_Int_Remove
5. A821_Int_GetCount
6. A821_Int_GetHexBuf
7. A821_Int_GetFloatBuf

The functions of AD, Channel Scan listing as follows:

1. A821_ChScan_Clear
2. A821_ChScan_Add
3. A821_ChScan_Set
4. A821_ChScan_PollingHex
5. A821_ChScan_PollingFlaot

The functions of AD Interrupt, Channel Scan listing as follows:

1. A821_ChScan_IntInstall
2. A821_ChScan_IntStart
3. A821_ChScan_IntStop
4. A821_ChScan_IntRemove
5. A821_ChScan_IntGetCount
6. A821_ChScan_IntGetHexBuf
7. A821_ChScan_IntGetFloatBuf

In this chapter, we use some keywords to indicate the attribute of Parameters.

Keyword	Setting parameter by user before calling this function ?	Get the data/value from this parameter after calling this function ?
[In]	Yes	No
[Out]	No	Yes
[In, Out]	Yes	Yes

Note: All of the parameters need to be allocated spaces by the user.

3.1 TEST FUNCTION

3.1.1 A821_SHORT_SUB_2

- **Description:**
  Compute C=A-B in **short** formats, **short=16 bits sign integer.** This function is provided for testing purpose.

- **Syntax:**
  short A821_SHORT_SUB_2(short nA, short nB);

- **Parameter:**
  nA : [In] short integer
  nB : [In] short integer

- **Return:**
  return=nA-nB → short integer

3.1.2 A821_FLOAT_SUB_2

- **Description:**
  Compute A-B in **float** format, **float=32 bits floating pointer number.** This function is provided for testing purpose.

- **Syntax:**
  float A821_FLOAT_SUB_2(float fA, float fB);

- **Parameter:**
  fA : [In] floating point value
  fB : [In] floating point value

- **Return:**
  return=fA-fB → floating point value

3.1.3 A821_Get_DLL_Version

- **Description:**
  Read the software version of the A821.DLL.

- **Syntax:**
  WORD A821_Get_DLL_Version(void) ;

- **Parameter:**
  void

- **Return:**
  return=0x200 → Version 2.00 **(WORD=16 bits unsigned integer)**

3.1.4 A821_GetDriverVersion

- **Description:**
  This subroutine will get the version number about the device driver.

- **Syntax:**
  WORD A821_GetDriverVersion(WORD *wDriverVersion);

- **Parameter:**
  wDriverVersion :[Out] the address of wDriverVersion.
  when wDriverVerion=0x210 → version 2.10

- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.2 DI/DO FUNCTION

3.2.1 A821_DI

- **Description:**
  This subroutine will read the 16 bits data from the digital input port.

- **Syntax:**
  WORD A821_DI(WORD *wInVal);

- **Parameter:**
  wInVal : [In] 16 bits Digital-Input value.

- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.2.2 A821_DO

- **Description:**
  This subroutine will send the 16 bits data to digital output port.

- **Syntax:**
  WORD A821_DO(WORD wHexValue);

- **Parameter:**
  wHexValue : [In] 16 bit data send to digital output port

- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.2.3 A821_OutputByte

- **Description:**
  This subroutine will send the 8 bits data to the desired I/O port.

- **Syntax:**
  void A821_OutputByte(WORD wPortAddr, UCHAR bOutputVal);

- **Parameter:**

wPortAddr	: [In] I/O port address, for example, 0x220
bOutputVal	: [In] 8 bit data send to I/O port

- **Return:**
  void

3.2.4 A821_OutputWord

- **Description:**
  This subroutine will send the 16 bits data to the desired I/O port.

- **Syntax:**
  void A821_OutputByte(WORD wPortAddr, WORD wOutputVal);

- **Parameter:**

wPortAddr	: [In] I/O port address, for example, 0x220
wOutputVal	: [In] 16 bit data send to I/O port

- **Return:**
  void

3.2.5 A821_InputByte

- **Description:**
  This subroutine will input the 8 bit data from the desired I/O port.

- **Syntax:**
  WORD A821_InputByte(WORD wPortAddr);

- **Parameter:**
  wPortAddr : [In] I/O port address, for example, 0x220

- **Return:**
  16 bits data with the leading 8 bits are all 0

3.2.6 A821_InputWord

- **Description:**
  This subroutine will input the 16 bit data from the desired I/O port.

- **Syntax:**
  WORD DIO_InputWord(WORD wPortAddr);

- **Parameter:**
  wPortAddr : [In] I/O port address, for example, 0x220

- **Return:**
  16 bits data.

3.3 A/D , D/A FUNCTION

3.3.1 A821_SetChGain

- **Description:**

  The subroutine sets the channel number and configuration code for the ADC. And then delay for the settling time.

  The user have to call this function once before calling the "A821_Fast_AD_Hex()", "A821_Fast_AD_Float()", "A821_Int_Start()", "A821_ADs_Hex()" and "A821_ADs_Float()" functions.

- **Syntax:**

  WORD A821_SetChGain
  (WORD wChannel, WORD wConfig, WORD wCardType);

- **Parameter:**

  wChannel : [In] A/D channel number, 0 to 15.
  wConfig : [In] Configuration code, refer to "Section 2.1 Range Configuration " for detail information.
  wCardType : [In] 0 → A-821PGL, 1→ A-821PGH

- **Return:**

  Please refer to "Section 2.2 Error Code" for the detail information.

3.3.2 A821_Fast_AD_Hex

- **Description:**
  This subroutine will perform an A/D conversion by polling. The A/D converter is 12 bits for A821PGH/L. The user have to call the "A821_SetChGain()" function firstly.

- **Syntax:**
  WORD A821_Fast_AD_Hex(WORD *wVal);

- **Parameter:**
  wVal : [Out] 12 bits hex value of Analog-Input.

- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.3.3 A821_Fast_AD_Float

- **Description:**
  This subroutine will perform an A/D conversion by polling. The A/D converter is 12 bits for A821PGH/L. This subroutine will compute the result according to the **configuration code**. The user have to call the "A821_SetChGain()" function firstly.

- **Syntax:**
  WORD A821_Fast_AD_Float(float *fVal);

- **Parameter:**
  fVal : [Out] Floating point value of Analog-Input.

- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.3.4 A821_AD_Hex

- **Description:**
  This subroutine will perform an A/D conversion by polling. The A/D converter is 12 bits for A821PGH/L.

- **Syntax:**

```
WORD A821_AD_Hex
    (WORD wChannel, WORD wConfig,  WORD wCardType, WORD *wVal);
```

- **Parameter:**

wChannel	: [In] A/D channel number, 0 to 15.
wConfig	: [In] Configuration code, refer to "Section 2.1 Range Configuration Code" for detail information
wCardType	: [In] 0 → A-821PGL, 1→ A-821PGH
wVal	: [Out] 12 bits hex value of Analog-Input.

- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.3.5 A821_AD_Float

- **Description:**
  This subroutine will perform an A/D conversion by polling. The A/D converter is 12 bits for A821PGH/L. This subroutine will compute the result according to the **configuration code**.

- **Syntax:**

```
WORD A821_AD_Float
    (WORD wChannel, WORD wConfig,  WORD wCardType, float *fVal);
```

- **Parameter:**

wChannel	: [In] A/D channel number, 0 to 15.
wConfig	: [In] Configuration code, refer to "Section 2.1 Range Configuration Code" for detail information
wCardType	: [In] 0 → A-821PGL, 1→ A-821PGH
fVal	: [Out] Floating point value of Analog-Input.

- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.3.6 A821_ADs_Hex

- **Description:**

  This subroutine will perform a number of A/D conversions by polling. This subroutine is very similar to A821_AD_Hex except that this subroutine will perform wCount of conversions instead of just one conversion. The A/D conversing at the ISA bus's max. speed. After A/D conversing, the A/D data are stored in a buffer in Hex format. The **wBuf** is the starting address of this data buffer. The user have to call the "A821_SetChGain()" function firstly.

- **Syntax:**

  WORD A821_ADs_Hex(WORD wBuf[], WORD wCount);

- **Parameter:**

  wBuf : [Out] Starting address of the data buffer
  (In WORD format)

  The user must allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. The user can analyzes these data from the buffer after calling this function.

  wCount : [In] Number of A/D conversions will be performed

- **Return:**

  Please refer to "Section 2.2 Error Code" for the detail information.

3.3.7 A821_ADs_Float

- **Description:**

  This subroutine will perform a number of A/D conversions by polling. This subroutine is very similar to A821_AD except that this subroutine will perform wCount of conversions instead of just one conversion. The A/D conversing at the ISA bus's max. speed. Then the A/D data are stored in a data buffer in Float format. The **fBuf** is the starting address of this data buffer. The user have to call the "A821_SetChGain()" function firstly.

- **Syntax:**

  WORD A821_ADs_Float(float fBuf[], WORD wCount);

- **Parameter:**

fBuf	: [Out] Starting address of the data buffer (In float format) The user must allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. The user can analyzes these data from the buffer after calling this function.
wCount	: [In] Number of A/D conversions will be performed

- **Return:**

  Please refer to "Section 2.2 Error Code" for the detail information.

3.3.8 A821_DA_Hex

- **Description:**
  This subroutine will send the 12 bits data to D/A analog output. The output range of D/A maybe 0-5V or 0-10V **setting by hardware jumper, JP1.** The software **can not detect** the output range of D/A converter. **For examples, if hardware select -5V, the 0xfff will send out 5V. If hardware select -10V, the 0xfff will send out 10V**. **The factory setting select 0-5V D/A output range.**

- **Syntax:**
  WORD A821_DA_Hex(WORD wHexValue);

- **Parameter:**
  wHexValue : [In] 12 bit data send to D/A converter

- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.3.9 A821 _DA_Uni5

- **Description:**
  This subroutine will send the 12 bits data to D/A analog output. The output range of D/A dependent on **setting by hardware jumper, JP1 (-5v or –10v).** The software **can not detect** the output range of D/A converter. This subroutine can be used only when the jumper's settings are **: -5v .** The **output range is between 0.0v and 5.0v.** Please refer to hardware manual to setting jumpers.

- **Syntax:**
  void A821 _DA_Uni5 (float fValue);

- **Parameter:**
  fValue : [In] 12 bit data send to D/A converter

- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.3.10 A821 _DA_Uni10

- **Description:**

  This subroutine will send the 12 bits data to D/A analog output. The output range of D/A dependent on **setting by hardware jumper, JP1 (-5v or –10v).** The software **can not detect** the output range of D/A converter. This subroutine can be used only when the jumper's settings are **: -10v .** The **output range is between 0.0v and 10.0v.** Please refer to hardware manual to setting jumpers.

- **Syntax:**

  void A821 _DA_Uni10 (float fValue);

- **Parameter:**

  fValue : [In] 12 bit data send to D/A converter

- **Return:**

  Please refer to "Section 2.2 Error Code" for the detail information.

3.4 DRIVER FUNCTION

3.4.1 A821_DriverInit

- **Description:**

This subroutine will open the device driver. After calling the A821_DriverInit() function, the user still have to call the A821_ActiveBoard() function firstly before access the device.

- **Syntax:**

WORD A821_DriverInit(WORD *wTotalBoards);

- **Parameter:**

WTotalBoards : [Out] Returns an number of how many board that found by the driver.

- **Return:**

Please refer to "Section 2.2 Error Code" for the detail information.

3.4.2 A821_DriverClose

- **Description:**

This subroutine will close the device driver.

- **Syntax:**

void A821_DriverClose(void);

- **Parameter:**

void

- **Return:**

void

3.4.3 A821_DELAY

- **Description:**
  This subroutine will delay **wDownCount** mS(machine independent timer).
  This function uses the System-Clock to implement delay function.
  The unit of A821_DELAY() is 0.5uSeconds.

- **Syntax:**
  WORD A821_DELAY(WORD wDownCount);

- **Parameter:**
  wDownCount : [In] Number of 0.5uS will be delay

- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.4.4 A821_Check_Address

- **Description:**
  This subroutine will detect the A-821PGH/L in I/O base address. This subroutine will perform one A/D conversion, if success → find a A-821PGH/L. This function will always return 0 if the user set the trigger mode to external.

- **Syntax:**
  WORD A821_Check_Address(void);

- **Parameter:**
  None

- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.4.5 A821_GetConfigAddress

- **Description:**

  This subroutine returns the Base-Address and board-number of the current board.

  If the current board is invalid, the Base-Address will be 0.

- **Syntax:**

```
WORD A821_GetConfigAddress
            (WORD *wAddrBase, WORD *wCurrentBoard);
```

- **Parameter:**

  wAddrBase : [Out] Returns the Base-Address of the current board.
  wCurrentBoard : [Out] Returns the board-number of the current board.

- **Return:**

  Please refer to "Section 2.2 Error Code" for the detail information.

3.4.6 A821_ActiveBoard

- **Description:**

  This subroutine active the specified board and then calling the A821_Check_Address() function to check this hardware automatically. If the function can not access this device, it also returns A821_CardNotFound error code. Please refer to the "A821_DriverInit()" function for the valid range of board number.

- **Syntax:**

```
    WORD A821_ActiveBoard( WORD wBoardNo );
```

- **Parameter:**

  wBoardNo : [In] The board number to be active.

- **Return:**

  Please refer to "Section 2.2 Error Code" for the detail information.

3.5 AD , INTERRUPT FUNCTION

3.5.1 A821_Int_Install

- **Description:**

  This subroutine will install interrupt handler and allocate buffer. For more detail information of using interrupt please refer to "Section 3.5.8 Architecture of Interrupt Mode".

- **Syntax:**

  WORD A821_Int_Install(HANDLE *hEvent, DWORD dwCount);

- **Parameter:**

hEvent	: [In] The Event handle that created by the user.
dwCount	: [In] The desired A/D entries count for interrupt transfer.

- **Return:**

  Please refer to "Section 2.2 Error Code" for the detail information.

3.5.2 A821_Int_Start

- **Description:**

  This subroutine will clear the interrupt-counter and start the interrupt transfer for a specific A/D channel and programming the gain code and sampling rate. The user have to call the "A821_SetChGain()" function once firstly.

- **Syntax:**

  WORD A821_Int_Start(WORD c1, Word c2);

- **Parameter:**

c1,c2	: [In] the sampling rate is 2M/(c1*c2) c1➔ Counter1, c2➔ Counter2

- **Return:**

  Please refer to "Section 2.2 Error Code" for the detail information.

3.5.3 A821_Int_Stop

- **Description:**
  This subroutine will stop the interrupt transfer.

- **Syntax:**
  WORD A821_Int_Stop(void);

- **Parameter:**
  void.

- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.5.4 A821_Int_Remove

- **Description:**
  This subroutine will remove the interrupt handler and free the buffer.

- **Syntax:**
  WORD A821_Int_Remove(void);

- **Parameter:**
  void.

- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.5.5 A821_Int_GetCount

- **Description:**
  This subroutine will read the transferred count of interrupt.

- **Syntax:**
  WORD A821_Int_GetCount(DWORD *dwVal)

- **Parameter:**
  dwVal : [Out] Returns the interrupt transferred count.

- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.5.6 A821_Int_GetHexBuf

- **Description:**

This subroutine will copy the transferred interrupted data into the user's buffer.

- **Syntax:**

WORD A821_Int_GetHexBuf(WORD wBuf[], DWORD dwNum)

- **Parameter:**

wBuf : [Out] The address of wBuffer(In WORD format).
The user must allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. The user can analyzes these data from the buffer after calling this function.

dwNum : [In] The number to transfer.

- **Return:**

Please refer to "Section 2.2 Error Code" for the detail information.

3.5.7 A821_Int_GetFloatBuf

- **Description:**

This subroutine will copy the transferred interrupted data into the user's buffer.

- **Syntax:**

WORD A821_Int_GetFloatBuf(float fBuf[],DWORD dwNum)

- **Parameter:**

fBuf : [Out] The address of fBuffer(In float format).
The user must allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. The user cans analyze these data from the buffer after calling this function.

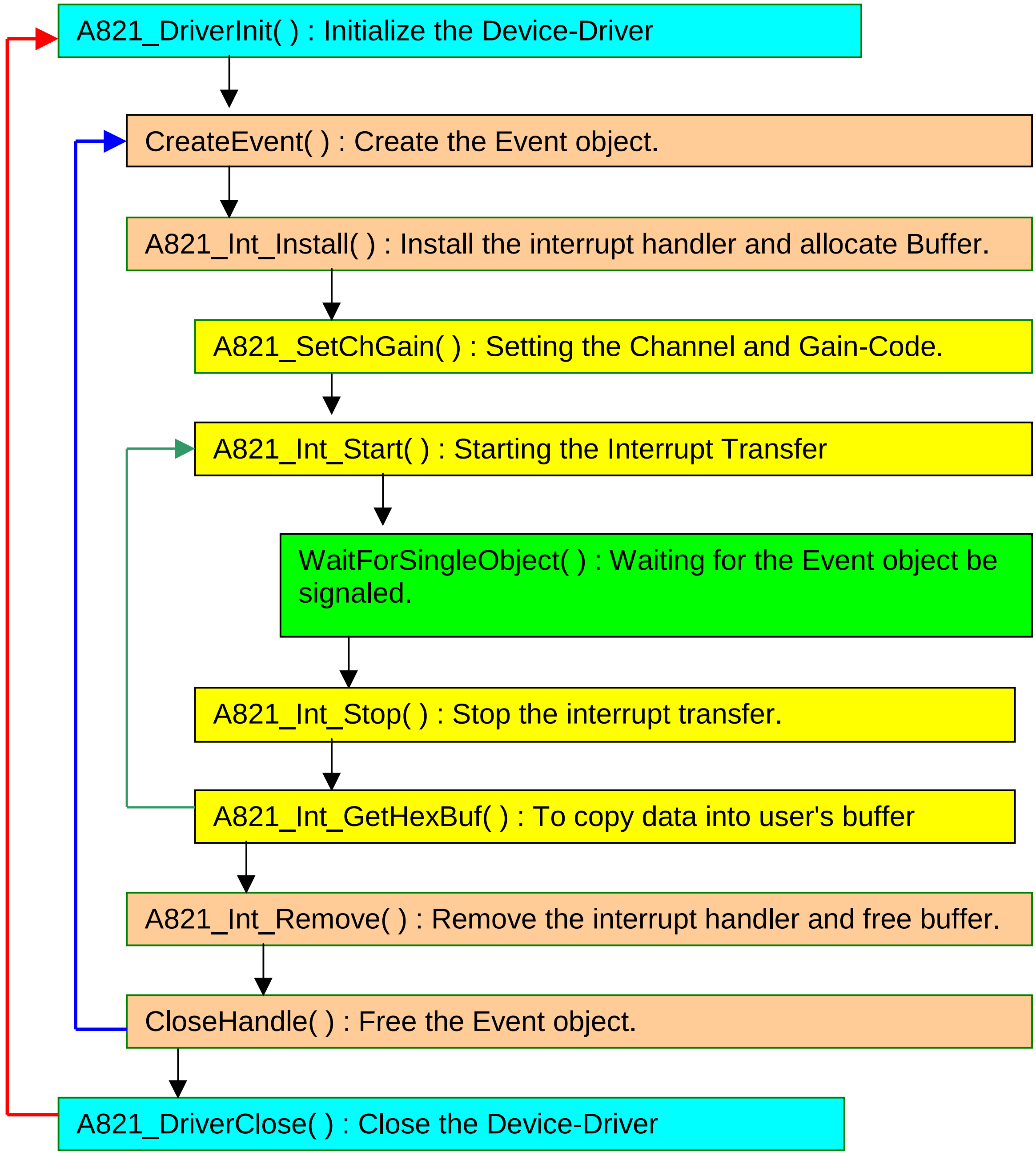
dwNum : [In] The number to transfer.

- **Return:**

Please refer to "Section 2.2 Error Code" for the detail information.

3.5.8 Architecture of Interrupt mode

The 3.5.1 to 3.5.7 are these functions to perform the A/D conversion with interrupt transfer. The flow chart to program these functions is given as follows:

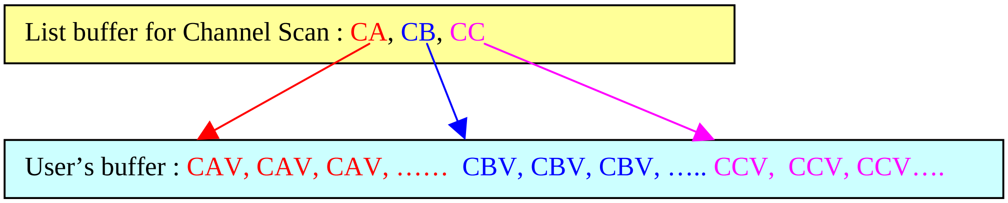

A821_DriverInit( ) : Initialize the Device-Driver
CreateEvent( ) : Create the Event object.
A821_Int_Install( ) : Install the interrupt handler and allocate Buffer.
A821_SetChGain( ) : Setting the Channel and Gain-Code.
A821_Int_Start( ) : Starting the Interrupt Transfer
WaitForSingleObject( ) : Waiting for the Event object be signaled.
A821_Int_Stop( ) : Stop the interrupt transfer.
A821_Int_GetHexBuf( ) : To copy data into user's buffer
A821_Int_Remove( ) : Remove the interrupt handler and free buffer.
CloseHandle( ) : Free the Event object.
A821_DriverClose( ) : Close the Device-Driver

3.6 AD, CHANNEL SCAN FUNCTION

3.6.1 Introduction

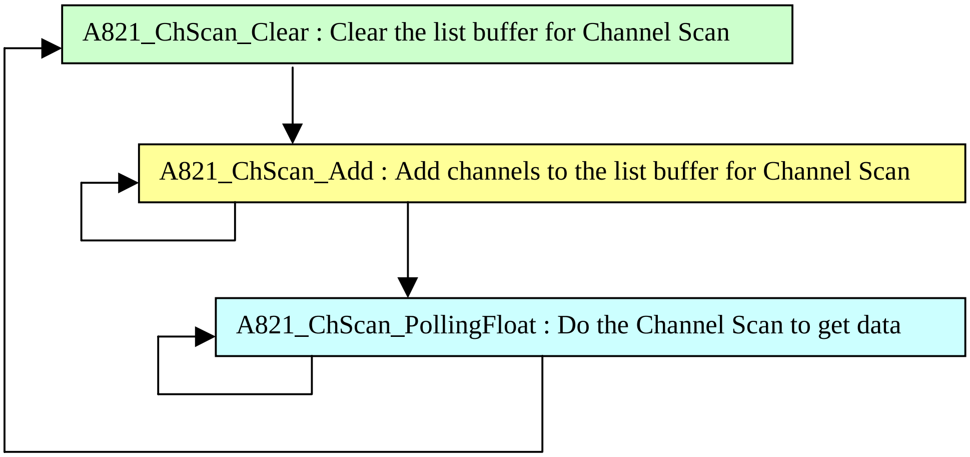
The user can specify channels into a list buffer. The other function will do the ADC to get the data. And then read the list buffer to change to next channel and set to specify configuration code.

The data will be saved into the following style:


List buffer for Channel Scan : CA, CB, CC
User's buffer : CAV, CAV, CAV, …… CBV, CBV, CBV, ….. CCV, CCV, CCV….

Note:
CA= Channel A; CB= Channel B; CC= Channel C
CAV= Channel A's value; CBV= Channel B's value;
CCV= Channel C's value

The user program's architecture as following:


A821_ChScan_Clear : Clear the list buffer for Channel Scan
A821_ChScan_Add : Add channels to the list buffer for Channel Scan
A821_ChScan_PollingFloat : Do the Channel Scan to get data

3.6.2 A821_ChScan_Clear

- **Description:**
  This subroutine will clear the list buffer for the Channel Scan.

- **Syntax:**
  void A821_ChScan_Clear(void);

- **Parameter:**
  None

- **Return:**
  None

3.6.3 A821_ChScan_Add

- **Description:**
  This function will add the specified channel number and configuration-code into the list buffer for the Channel Scan. The max number of the list buffer for the Channel Scan is 100 channels.

- **Syntax:**
  WORD A821_ChScan_Add(WORD wChannel, WORD wConfig);

- **Parameter:**

wChannel	: [In] Which channel to be scanned.
WConfig	: [In] Specify the configuration-code for this channel. Please refer to "Section 2.1 Range Configuration " for detail information.

- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.6.4 A821_ChScan_Set

- **Description:**

  This function will clear the list buffer and then copy the specified list of channel(s) and configuration-code(s) into the list buffer for the Channel Scan. The max number of the list buffer for the Channel Scan is 100 channels.

- **Syntax:**

  ```
  WORD A821_ChScan_Set
      (WORD wChannel[], WORD wConfig[], WORD wChNum);
  ```

- **Parameter:**

wChannel	: [In] The list of channel(s) to be scanned.
WConfig	: [In] The list of configuration-code(s) for channel(s). Please refer to "Section 2.1 Range Configuration".
wChNum	: [In] Total channels pass into and to be scanned.

- **Return:**

  Please refer to “Section 2.2 Error Code” for the detail information.

3.6.5 A821_ChScan_PollingHex

- **Description:**

  This subroutine will perform a number of A/D conversions by polling. And after get the channel's data, it then read the list buffer for the Channel Scan to change to next channel and set to specified configuration code. The A/D conversing at the ISA bus's max. speed. After A/D conversing, the A/D data are stored in a buffer in Hex format.

  Before calling this function, the user have to call the A821_ChScan_Clear() and A821_ChScan_Add() functions to setup the list buffer for Channel Scan. Please refer to the "Section 3.6.1 Introduction" for more information.

- **Syntax:**

```
WORD A821_ChScan_PollingHex
            (WORD wCardType, WORD wBuf[], WORD wNumPerCh);
```

- **Parameter:**

  WCardType : [In] 0: A-821L 1: A-821H

  wBuf : [Out] Starting address of the data buffer (WORD format)

  The user must allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. The user can analyzes these data from the buffer after calling this function.

  The buffer size
  = Total-Channels * wNumPerCh * sizeof(WORD)

  wNumPerCh : [In] Number of A/D conversions will be performed for every channel.

- **Return:**

  Please refer to "Section 2.2 Error Code" for the detail information.

3.6.6 A821_ChScan_PollingFloat

- **Description:**

  This subroutine will perform a number of A/D conversions by polling. And after get the channel's data, it then read the list buffer for the Channel Scan to change to next channel and set to specified configuration code. The A/D conversing at the ISA bus's max. speed. After A/D conversing, the A/D data are stored in a buffer in floating format.

  Before calling this function, the user have to call the A821_ChScan_Clear() and A821_ChScan_Add() functions to setup the list buffer for Channel Scan. Please refer to the "Section 3.6.1 Introduction" for more information.

- **Syntax:**

  WORD A821_ChScan_PollingFloat
  (WORD wCardType, float fBuf[], WORD wNumPerCh);

- **Parameter:**

  WCardType : [In] 0: A-821L 1: A-821H

  fBuf : [Out] Starting address of the data buffer (floating format)
  The user must allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. The user can analyzes these data from the buffer after calling this function.

  The buffer size
  = Total-Channels * wNumPerCh * sizeof(float)

  wNumPerCh : [In] Number of A/D conversions will be performed for every channel.

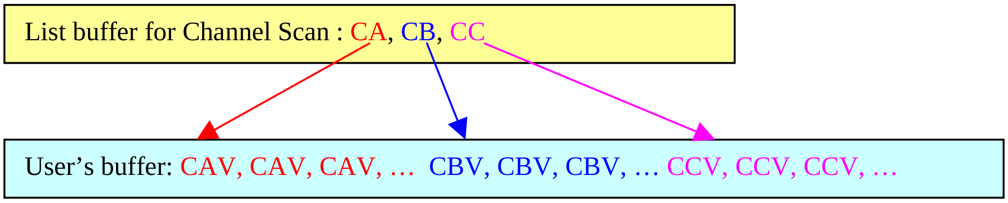
- **Return:**

  Please refer to "Section 2.2 Error Code" for the detail information.

3.7 AD INTERRUPT, CHANNEL SCAN FUNCTION

3.7.1 Introduction

The user can specify channels into a list buffer. The other function will do the ADC to get the data. And then read the list buffer to change to next channel and set to specify configuration code.



List buffer for Channel Scan : CA, CB, CC

User's buffer: CAV, CAV, CAV, … CBV, CBV, CBV, … CCV, CCV, CCV, …

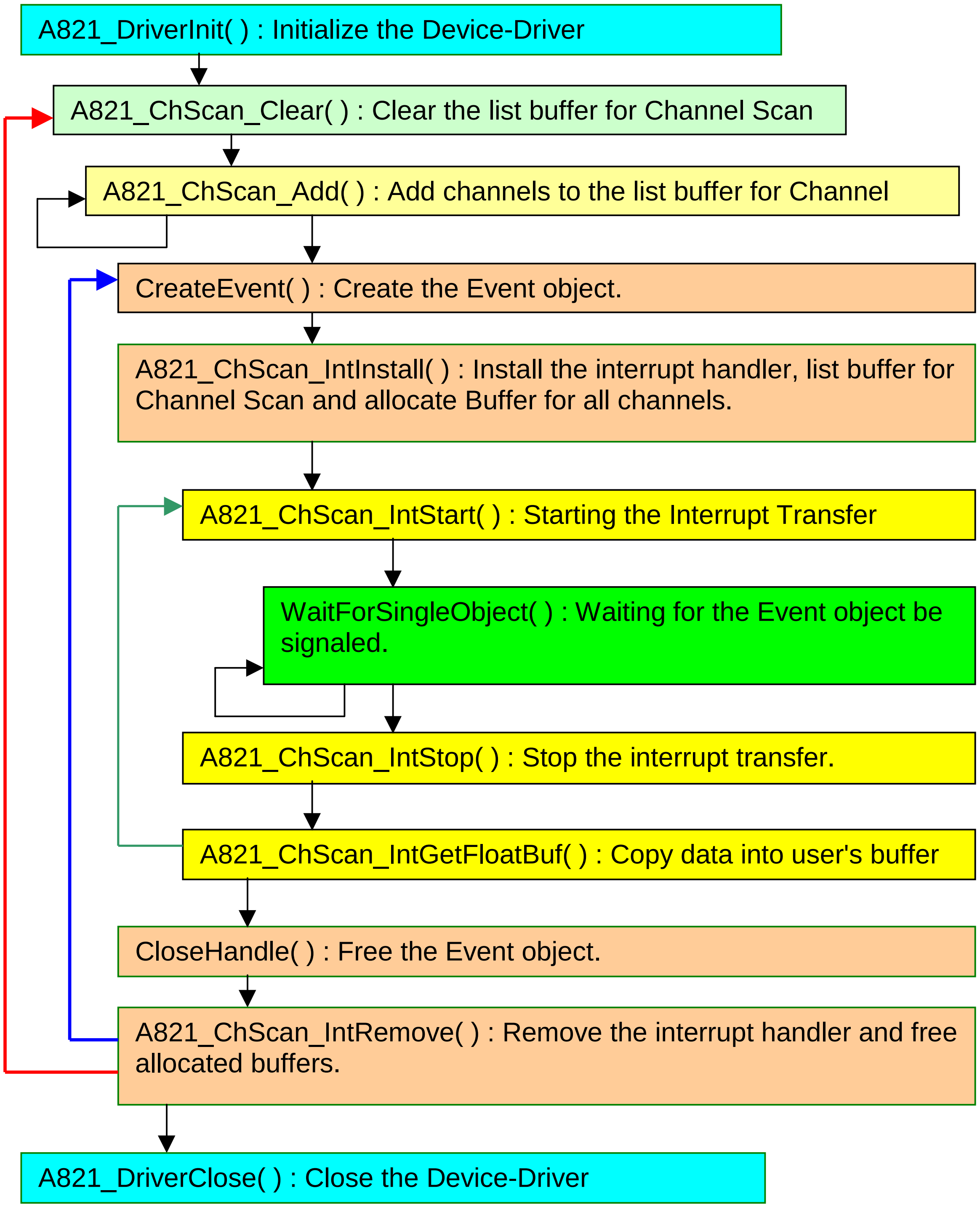
The data will be saved into the following style:

Note:
CA= Channel A; CB= Channel B; CC= Channel C
CAV= Channel A's value; CBV= Channel B's value;
CCV= Channel C's value

After setting to the next channel and specified configuration code, it have to delay for the settling time before next ADC. The interrupt service routinue doesn't delay for the settling time. Thus, to get the correct ADC data, the user have to slow-down the sampling-rate of interrupt.

The sampling-rate is for all channels.
For example:
The list buffer for the Channel Scan is setting to channel-2 and channel-0. The sampling-rate is setting to 10KHz. In actually, the cannel-2 has the sampling-rate 5KHz and the channel-0 also has the sampling-rate 5KHz.

The user program’s architecture as following:


A821_DriverInit( ) : Initialize the Device-Driver
A821_ChScan_Clear( ) : Clear the list buffer for Channel Scan
A821_ChScan_Add( ) : Add channels to the list buffer for Channel
CreateEvent( ) : Create the Event object.
A821_ChScan_IntInstall( ) : Install the interrupt handler, list buffer for Channel Scan and allocate Buffer for all channels.
A821_ChScan_IntStart( ) : Starting the Interrupt Transfer
WaitForSingleObject( ) : Waiting for the Event object be signaled.
A821_ChScan_IntStop( ) : Stop the interrupt transfer.
A821_ChScan_IntGetFloatBuf( ) : Copy data into user's buffer
CloseHandle( ) : Free the Event object.
A821_ChScan_IntRemove( ) : Remove the interrupt handler and free allocated buffers.
A821_DriverClose( ) : Close the Device-Driver

3.7.2 A821_ChScan_IntInstall

- **Description:**
  This subroutine will install interrupt handler, copy the list buffer for Channel Scan into kernel-mode driver and allocate buffers for every channels. Before install the interrupt, the user have to call the "A821_ChScan_Clear()" and "A821_ChScan_Add()" functions to setup the list buffer for Channel Scan firstly. For more detail information of using interrupt please refer to “Section 3.7.1 Introduction".

- **Syntax:**
  WORD A821_ChScan_IntInstall(HANDLE *hEvent, DWORD dwNumPerCh);

- **Parameter:**

hEvent	: [In] The Event handle that created by the user.
dwNumPerCh	: [In] The desired A/D count for every channels to transfer.

- **Return:**
  Please refer to “Section 2.2 Error Code” for the detail information.

3.7.3 A821_ChScan_IntStart

- **Description:**
  This subroutine will clear the interrupt-counter and start the interrupt transfer for the specific A/D channels and programming the gain code and sampling rate.

- **Syntax:**
  WORD A821_ChScan_IntStart(WORD c1, WORD c2, WORD wCardType);

- **Parameter:**
  c1,c2 : [In] the sampling rate is 2M/(c1*c2); c1=Counter1, c2=Counter2
  wCardType : [In] 0: A-821L 1: A-821H

- **Return:**
  Please refer to “Section 2.2 Error Code” for the detail information.

3.7.4 A821_ChScan_IntStop

- **Description:**
  This subroutine will stop the interrupt transfer.
- **Syntax:**
  WORD A821_ChScan_IntStop(void);
- **Parameter:**
  void.
- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.7.5 A821_ChScan_IntRemove

- **Description:**
  This subroutine will remove the interrupt handler and free the buffers.
- **Syntax:**
  WORD A821_ChScan_IntRemove(void);
- **Parameter:**
  void.
- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.7.6 A821_ChScan_IntGetCount

- **Description:**
  This subroutine will read the transferred count of interrupt.
- **Syntax:**
  WORD A821_Int_GetCount(DWORD *dwVal)
- **Parameter:**
  dwVal : [Out] Returns the interrupt transferred count.
- **Return:**
  Please refer to "Section 2.2 Error Code" for the detail information.

3.7.7 A821_ChScan_IntGetHexBuf

- **Description:**

  This subroutine will copy the transferred interrupted data into the user's buffer.

- **Syntax:**

  WORD A821_ChScan_IntGetHexBuf(WORD wBuf[])

- **Parameter:**

  wBuf : [Out] The address of wBuf(WORD format).

  The user must allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. The user can analyzes these data from the buffer after calling this function.

  Buffer size = Total-Channels * dwNumPerCh * sizeof(WORD)

- **Return:**

  Please refer to "Section 2.2 Error Code" for the detail information.

3.7.8 A821_ChScan_IntGetFloatBuf

- **Description:**

  This subroutine will copy the transferred interrupted data into the user's buffer.

- **Syntax:**

  WORD A821_ChScan_IntGetFloatBuf(float fBuf[])

- **Parameter:**

  fBuf : [Out] The address of fBuf(float format).

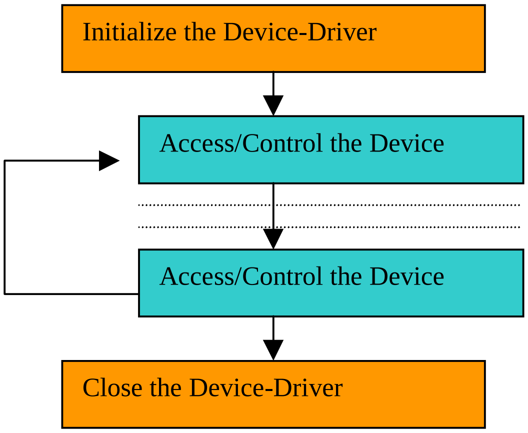
  The user must allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. The user cans analyze these data from the buffer after calling this function.

  Buffer size = Total-Channels * dwNumPerCh * sizeof(float)

- **Return:**

  Please refer to "Section 2.2 Error Code" for the detail information.

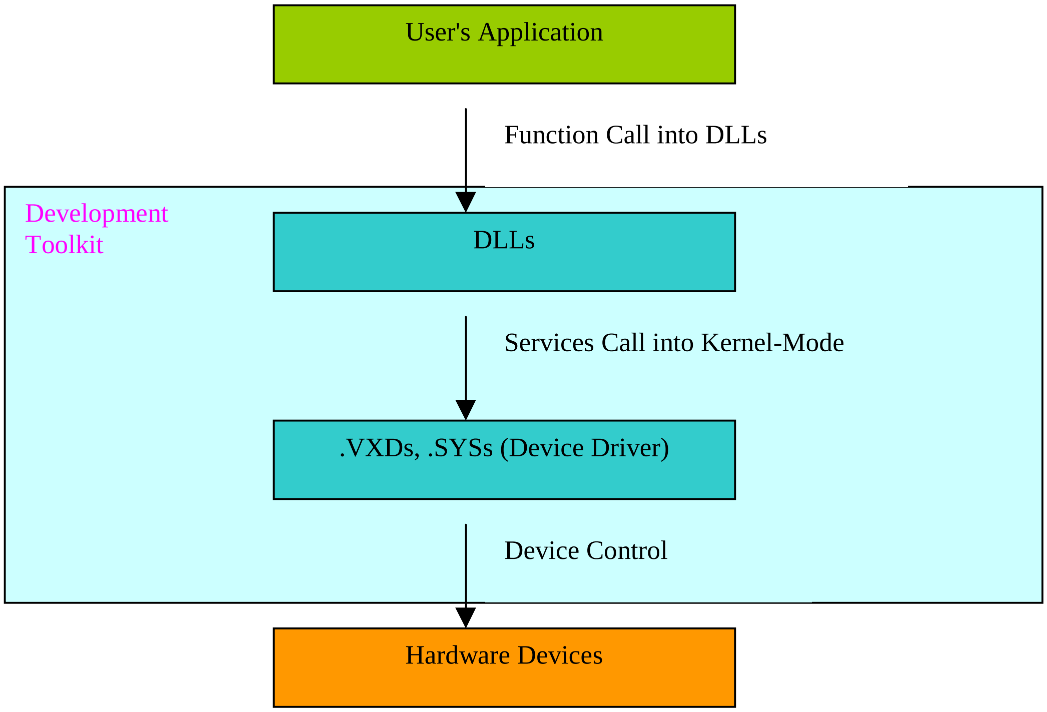
4. PROGRAM ARCHITECTURE


Initialize the Device-Driver
Access/Control the Device
Access/Control the Device
Close the Device-Driver

```
A821_DriverInit()

….
     A821_InputByte( … )
     ……..
     ……..
     A821_OutputByte(…)
…..

A821_DriverClose()
```



5. PROBLEMS REPORT

Technical support is available at no charge as described below. The best way to report problems is send electronic mail to

Service@icpdas.com

on the Internet.

When reporting problems, please include the following information:

1) Is the problem reproducible? If so, how?

2) What kind and version of Operation Systems that you running? For example, Windows 3.1, Windows for Workgroups, Windows NT 4.0, etc.

3) What kinds of our products that you using? Please see the product's manual.

4) If a dialog box with an error message was displayed, please include the full text of the dialog box, including the text in the title bar.

5) If the problem involves other programs or hardware devices, what devices or version of the failing programs that you using?

6) Other comments relative to this problem or any Suggestions will be welcomed.

After we received your comments, we will take about two business days to testing the problems that you said. And then reply as soon as possible to you. Please check that we have received your comments? And please keeping contact with us.

E-mail: Service@icpdas.com
Web-Site: http://www.icpdas.com