A821 PGH/L

Software Manual

[For Windows 95/98/NT]

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, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright 2000 by ICP DAS. All rights are reserved.

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. INTRODUCTION 4
1.1 REFERENCES 5
1.2 RANGE CONFIGURATION 6
2. DECLARATION & DEMO 7
2.1 USING VC++ & BC++Builder 9
2.1.1 VC++ Demo Result 9
2.1.2 BC++Builder Demo Result : 10
2.1.3 A821.H 11
2.2 USING VISUAL BASIC 16
2.2.1 VB Demo Result: 16
2.2.2 A821.BAS 17
2.3 USING DELPHI 22
2.3.1 Delphi Demo Result : 22
2.3.2 A821.PAS 23
3. FUNCTION DESCRIPTION 30
3.1 ERROR CODE 32
3.2 DRIVER FUNCTIONS 34
3.2.1 A821_DriverInit 34
3.2.2 A821_DriverClose 34
3.2.3 A821_DELAY 35
3.2.4 A821_Check_Address 35
3.3 TEST FUNCTIONS 36
3.3.1 A821_SHORT_SUB_2 36
3.3.2 A821_FLOAT_SUB_2 36
3.3.3 A821_Get_DLL_Version 37
3.3.4 A821_GetDriverVersion 37
3.4 COUNTER FUNCTION 38
3.4.1 A821_SetCounter 38
3.4.2 A821_ReadCounter 38
3.5 DI/DO FUNCTION 39
3.5.1 A821_DI 39
3.5.2 A821_DO 39
3.5.3 A821_OutputByte 40
3.5.4 A821_OutputWord 40
3.5.5 A821_InputByte 41
3.5.6 A821_InputWord 41
3.6 D/A FUNCTION 42
3.6.1 A821_DA 42
3.6.2 A821_Uni5_DA 43
3.6.3 A821_Uni10_DA 43
3.7 A/D FUNCTION 44
3.7.1 A821_AD_Float 44
3.7.2 A821_AD_Hex 45
3.7.3 A821_ADs_Hex 46
3.7.4 A821_ADs_Float 47
3.7.5 A821_Fast_SetChGain 48
3.7.6 A821_Fast_AD_Float 48

3.7.7 A821_Fast_AD_Hex 49
3.7.8 A821_Hex2Float 49
3.8 AD IINTERRUPT FUNCTION 50
3.8.1 A821_IntInstall 50
3.8.2 A821_IntStart 50
3.8.3 A821_IntGetCount 51
3.8.4 A821_IntGetHexBuf 51
3.8.5 A821_IntGetFloatBuf 52
3.8.6 A821_IntStop 53
3.8.7 A821_IntRemove 53
3.8.8 Interrupt Architecture 54
3.9 AD WITH CHANNEL SCAN 55
3.9.1 Introduction 55
3.9.2 A821_ChScan_Clear 56
3.9.3 A821_ChScan_Add 56
3.9.4 A821_ChScan_Set 57
3.9.5 A821_ChScan_PollingHex 58
3.9.6 A821_ChScan_PollingFloat 59
3.10 AD INTERRUPT, CHANNEL SCAN FUNCTION 60
3.10.1 Introduction 60
3.10.2 A821_ChScan_IntInstall 62
3.10.3 A821_ChScan_IntStart 62
3.10.4 A821_ChScan_IntStop 63
3.10.5 A821_ChScan_IntRemove 63
3.10.6 A821_ChScan_IntGetCount 63
3.10.7 A821_ChScan_IntGetHexBuf 64
3.10.8 A821_ChScan_IntGetFloatBuf 64
4. PROGRAM ARCHITECTURE 65
5. PROBLEMS REPORT 66

1. INTRODUCTION

The **A821** has a collection of DLLs for Windows 95/98 and NT 4.0 application. These DLLs are 32 bits and can be called by Visual C/C++, Borland C++, Visual BASIC, Delphi, and LabVIEW.

The A821 consists of these DLLs and device driver:

For Windows 95/98

- A821.dll, A821.lib →Libraries for A821 PGL/PGH card
- A821.Vxd → A821 Device driver for Windows 95/98

For Windows NT

- A821.dll, A821.lib →Libraries for A821 PGL/PGH card
- A821.sys, Napwnt.sys →A821 Device driver for Windows NT

These DLLs can perform a variety of data acquisition operations as follows:

- Get software version
- Initialization
- Digital Input/Output
- A/D , D/A conversion

1.1 REFERENCES

Please refer to the following user manuals:

- **Readme.txt:**
  Describes what files install into your system, and where can find it our.

- **Whatnew.txt:**
  Describes what is the difference in the versions.

- **SoftInst.pdf:**
  Describes how to install the software package under Windows 95/98/NT.

- **CallDll.pdf:**
  Describes how to call the DLL functions with VC++5, VB5, Delphi3 and Borland C++ Builder 3.

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

1.2 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 most high 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	+/- 0.625V	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. DECLARATION & DEMO

Please refer to "CallDLL.pdf".

for Windows 95/98 user:

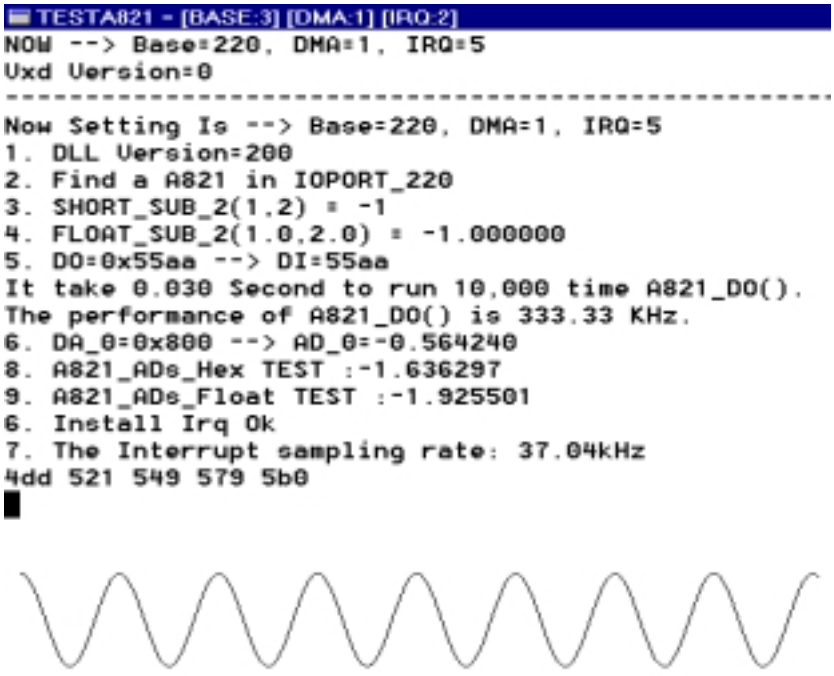
```
|--\Driver                <-- Driver for Windows 95/98 Only
|   |--\A821.DLL          <-- Dynamic Linking Library
|   |--\A821.Vxd          <-- Virtual Devie Driver
|   |
|   |--\BCB3              <-- for Borland C++ Builder 3
|   |   |--\A821.h        <-- Header file
|   |   |--\A821.Lib      <-- Import Library for BCB only
|   |   |--\A821u.cpp     <-- Some function for BCB
|   |
|   |--\Delphi3           <-- for Delphi 3
|   |   |--\A821.pas      <-- Declaration file
|   |   |--\A821u.pas     <-- Some function for Delphi
|   |
|   |--\VB5               <-- for Visual Basic 5
|   |   |--\A821.bas      <-- Declaration file
|   |   |--\A821u.bas     <-- Some function for VB
|   |
|   |--\VC5               <-- for Visual C++ 5
|       |--\A821.h        <-- Header file
|       |--\A821.Lib      <-- Import Library for VC only
```

for Windows NT user:

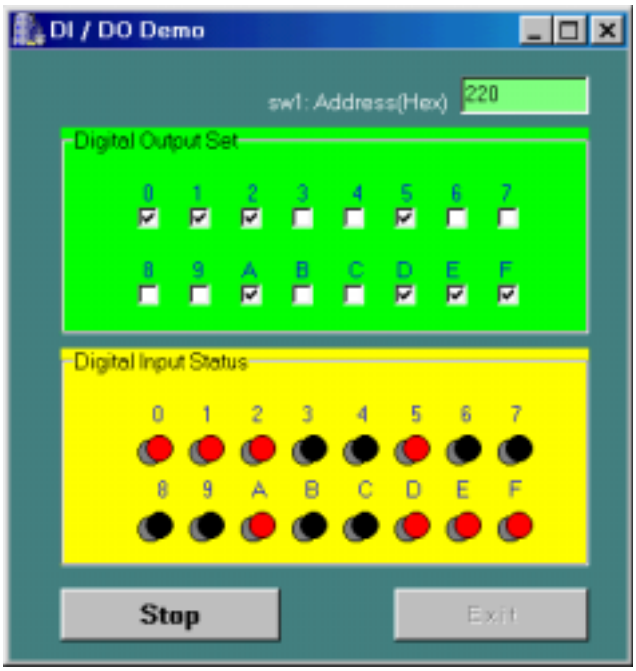
```
|--\Driver
  |   |--\A821.DLL        <-- Dynamic Linking Library
  |   |--\A821.sys        <-- device driver
  |   |--\Napwnt.sys      <-- device driver
  |   |
  |   |--\BCB3            <-- for Borland C++ Builder 3
  |   |   |--\A821.h      <-- Header file
  |   |   |--\A821.Lib    <-- Import Library for BCB only
  |   |   |--\A821u.cpp   <-- Some function for BCB
  |   |
  |   |--\Delphi3         <-- for Delphi 3
  |   |   |--\A821.pas    <-- Declaration file
  |   |   |--\A821u.pas   <-- Some function for Delphi
  |   |
  |   |--\VB5             <-- for Visual Basic 5
  |   |   |--\A821.bas    <-- Declaration file
  |   |   |--\A821u.bas   <-- Some function for VB
  |   |
  |   |--\VC5             <-- for Visual C++ 5
  |       |--\A821.h      <-- Header file
  |       |--\A821.Lib    <-- Import Library for VC only
```

2.1 USING VC++ & BC++Builder

2.1.1 VC++ Demo Result



2.1.2 BC++Builder Demo Result :



2.1.3 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 A821_INTERRUPT_MODE        6

// ** 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 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_AdError1               100
#define A821_AdError2               -200.0
#define A821_InstallBufError        10
#define A821_AllocateMemoryError    11
#define A821_CardTypeError          12
#define A821_TimeoutError           13
#define A821_OtherError             14

#define A821_IntStopError           19
#define A821_IntInstallEventError   20
#define A821_ConfigCodeError        23
#define A821_BufferFull             24
#define A821_NoChannelToScan        25
#define A821_IntInstallChannelError 26
#define A821_IntInstallConfigError  27

// Function of Driver
EXPORTS WORD    CALLBACK A821_DriverInit(void);
EXPORTS void    CALLBACK A821_DriverClose(void);
EXPORTS WORD    CALLBACK A821_DELAY(WORD wBase, WORD
wDownCount);
EXPORTS WORD    CALLBACK A821_Check_Address(WORD wBase);

// Function 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);

// Function of Counter
EXPORTS void    CALLBACK A821_SetCounter
        (DWORD dwBase, WORD wCounterNo,
        WORD bCounterMode, DWORD wCounterValue);
EXPORTS DWORD   CALLBACK A821_ReadCounter
        (DWORD dwBase, WORD wCounterNo, WORD bCounterMode);
```

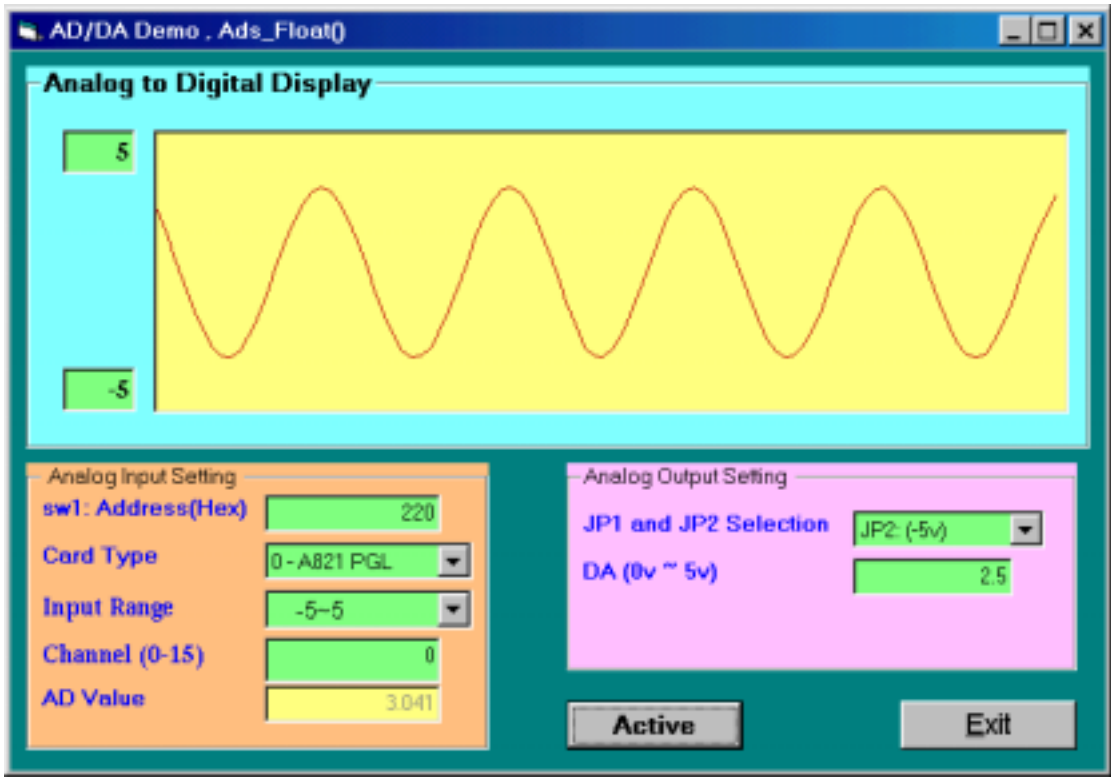
```
// Function of DI/DO
EXPORTS WORD      CALLBACK A821_DI(WORD wBase);
EXPORTS void CALLBACK A821_DO(WORD wBase, WORD wHexValue);

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);

// Function of AD
// Please uses A821_AD_Float(), A821_AD_Hex(),
//           A821_Fast_AD_Float(), A821_Fast_AD_Hex() functions
EXPORTS float CALLBACK A821_AD(WORD wBase,
           WORD wChannel, WORD wConfig, WORD wType);
EXPORTS float CALLBACK A821_Fast_AD(WORD wBase);

// Function of AD
EXPORTS WORD      CALLBACK A821_AD_Float
           (WORD wBase, WORD wChannel,
           WORD wConfig, WORD wType, float *fVal);
EXPORTS WORD      CALLBACK A821_AD_Hex
           (WORD wBase, WORD wChannel,
           WORD wConfig, WORD wType, WORD *wVal);
EXPORTS WORD      CALLBACK A821_ADs_Hex
           (WORD wBase, WORD wChannel, WORD wConfig,
           WORD wType, WORD wBuf[], WORD wCount);
EXPORTS WORD      CALLBACK A821_ADs_Float
           (WORD wBase, WORD wChannel, WORD wConfig,
           WORD wType, float fBuf[], WORD wCount);

EXPORTS void         CALLBACK A821_Fast_SetChGain
           (WORD wBase, WORD wChannel,
           WORD wConfig, WORD wCardType);
EXPORTS WORD      CALLBACK A821_Fast_AD_Float
           (WORD wBase, float *fVal);
EXPORTS WORD      CALLBACK A821_Fast_AD_Hex
           (WORD wBase, WORD *wVal);

EXPORTS WORD  CALLBACK A821_Hex2Float(WORD wHex,
           WORD wCardType, WORD wConfig, float *fVal);
```

```
// Function of DA
EXPORTS void  CALLBACK A821_DA(WORD wBase, WORD wHexValue);
EXPORTS void  CALLBACK A821_Uni5_DA(WORD wBase, float fValue);
EXPORTS void  CALLBACK A821_Uni10_DA(WORD wBase, float fValue);

// Function of Interrupt
// Please uses the newest function set A821_IntXXXXX series functions
// ****************************************************************
EXPORTS WORD  CALLBACK A821_InstallIrq
          (WORD wBase, WORD wIrq, HANDLE *hEvent, DWORD dwCount);
EXPORTS WORD  CALLBACK A821_AD_INT_Start
          (WORD wCardType, WORD Ch, WORD Gain,
          WORD c1, WORD c2);
EXPORTS WORD  CALLBACK A821_AD_INT_Stop(void);
EXPORTS WORD  CALLBACK A821_GetIntCount(DWORD *dwVal);
EXPORTS WORD  CALLBACK A821_GetBuffer
          (DWORD dwNum, WORD wBuffer[]);
EXPORTS WORD  CALLBACK A821_GetFloatBuffer
          (DWORD dwNum, float fBuffer[]);

// Function of Interrupt
EXPORTS WORD      CALLBACK A821_IntInstall(WORD wBase,
          WORD wIrq, HANDLE *hEvent, DWORD dwCount);
EXPORTS WORD      CALLBACK A821_IntStart(WORD wCardType,
          WORD wChannel, WORD wConfig, WORD c1, WORD c2);
EXPORTS WORD      CALLBACK A821_IntGetCount(DWORD *dwVal);
EXPORTS WORD      CALLBACK A821_IntGetHexBuf
          (DWORD dwNum, WORD wBuf[]);
EXPORTS WORD      CALLBACK A821_IntGetFloatBuf
          (DWORD dwNum, float fBuf[]);
EXPORTS WORD      CALLBACK A821_IntStop(void);
EXPORTS WORD      CALLBACK A821_IntRemove(void);

// Function of Channel-Scan with Polling
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 wBase,
          WORD wCardType, WORD wBuf[], WORD wNumPerCh);
EXPORTS WORD      CALLBACK A821_ChScan_PollingFloat
          (WORD wBase, WORD wCardType,
          float fBuf[], WORD wNumPerCh);
```

```
// Function of Channel-Scan with Interrupt
EXPORTS WORD CALLBACK A821_ChScan_IntInstall(WORD wBase,
        WORD wIrq, 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);
```

2.2 USING VISUAL BASIC

2.2.1 VB Demo Result:



2.2.2 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
Global Const A821_INTERRUPT_MODE    = 6

'*** 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 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_AdError1               = 100
Global Const A821_AdError2               = -200.0
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_IntStopError           = 19
Global Const A821_IntInstallEventError   = 20
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

' Function of Driver
Declare Function A821_DriverInit Lib "A821.DLL" () As Integer
Declare Sub A821_DriverClose Lib "A821.DLL" ()
Declare Function A821_DELAY Lib "A821.DLL" _
      (ByVal wBase As Integer, ByVal wDownCount As Integer) As Integer
Declare Function A821_Check_Address Lib "A821.DLL" _
      (ByVal wBase As Integer) As Integer

' Function of Test
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

' Function of Counter
Declare Sub A821_SetCounter Lib "A821.DLL" _
      (ByVal dwBase As Long, ByVal wCounterNo As Integer, _
      ByVal bCounterMode As Integer, ByVal wCounterValue As Long)
Declare Function A821_ReadCounter Lib "A821.DLL" _
      (ByVal dwBase As Long, ByVal wCounterNo As Integer, _
       ByVal bCounterMode As Integer) As Long
```

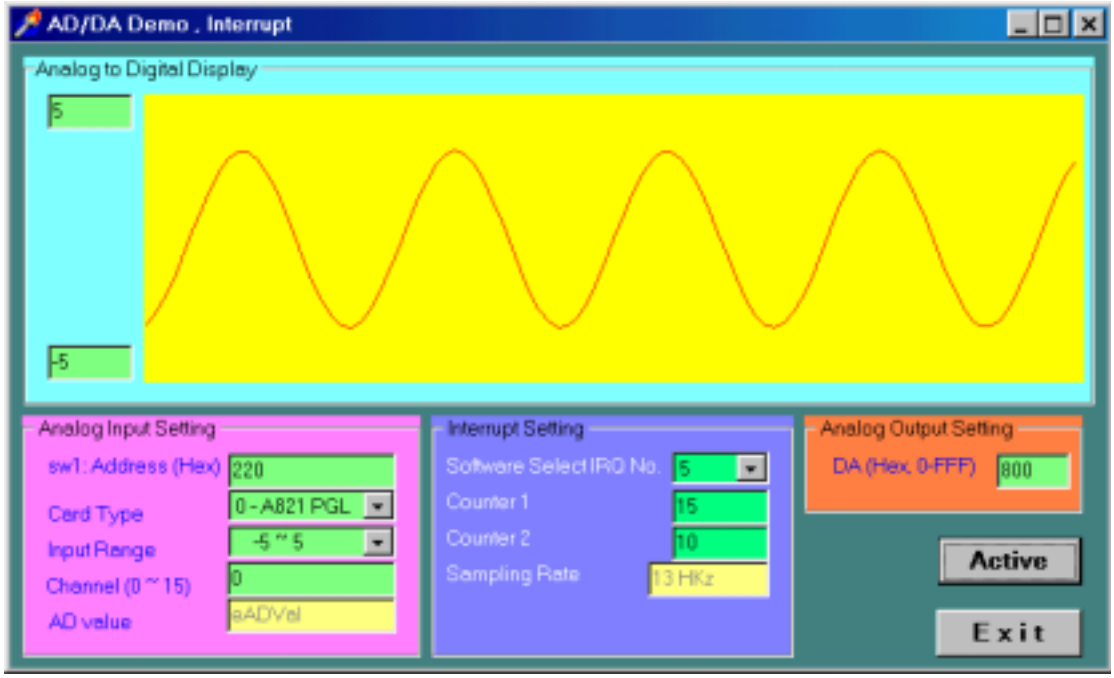
```
' Function of DI/DO
Declare Function A821_DI Lib "A821.DLL" _
      (ByVal wBase As Integer) As Integer
Declare Sub A821_DO Lib "A821.DLL" _
      (ByVal wBase As Integer, ByVal wHexValue As Integer)

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

' Function of AD
' Please uses A821_AD_Float(), A821_AD_Hex(),
'             A821_Fast_AD_Float(), A821_Fast_AD_Hex() function
Declare Function A821_AD Lib "A821.DLL" _
      (ByVal wBase As Integer, ByVal wChannel As Integer, _
       ByVal wConfig As Integer, ByVal wType As Integer) As Single
Declare function A821_Fast_AD Lib "A821.DLL" _
      (ByVal wBase As Integer) as Single

' Function of AD
Declare Function A821_AD_Float Lib "A821.DLL" _
      (ByVal wBase As Integer, ByVal wChannel As Integer, _
       ByVal wConfig As Integer, ByVal wType As Integer, _
       fVal As Single ) As Integer
Declare Function A821_AD_Hex Lib "A821.DLL" _
      (ByVal wBase As Integer, ByVal wChannel As Integer, _
       ByVal wConfig As Integer, ByVal wType As Integer, _
       wVal As Integer ) As Integer
Declare Function A821_ADs_Hex Lib "A821.DLL" _
      (ByVal wBase As Integer, ByVal wChannel As Integer, _
       ByVal wConfig As Integer, ByVal wType As Integer, _
       wBuf As Integer, ByVal wCount As Integer) As Integer
Declare Function A821_ADs_Float Lib "A821.DLL" _
      (ByVal wBase As Integer, ByVal wChannel As Integer, _
       ByVal wConfig As Integer, ByVal wType As Integer, _
       fBuf As Single, ByVal wCount As Integer) As Integer
```

```
Declare Sub A821_Fast_SetChGain Lib "A821.DLL" _
      (ByVal wBase As Integer, ByVal wChannel As Integer, _
       ByVal wConfig As Integer, ByVal wCardType As Integer)
Declare function A821_Fast_AD_Float Lib "A821.DLL" _
      (ByVal wBase As Integer, fVal As Single) as Integer
Declare function A821_Fast_AD_Hex Lib "A821.DLL" _
      (ByVal wBase As Integer, wVal As Single) as Integer

Declare function A821_Hex2Float Lib "A821.DLL" _
      (ByVal wHex As Integer, ByVal wCardType As Integer, _
       ByVal wConfig As Integer, fVal As Single) As Integer

' Function of DA
Declare Sub A821_DA Lib "A821.DLL" _
      (ByVal wBase As Integer, ByVal wHexValue As Integer)
Declare Sub A821_Uni5_DA Lib "A821.DLL" _
      (ByVal wBase As Integer, ByVal fValue As Single)
Declare Sub A821_Uni10_DA Lib "A821.DLL" _
      (ByVal wBase As Integer, ByVal fValue As Single)

' Function of Interrupt
' Please uses the newest function set A821_IntXXXXX series functions
' ****************************************************************
Declare Function A821_InstallIrq Lib "A821.DLL" _
      (ByVal wBase As Integer, ByVal wIrq As Integer, _
       hEvent As Long, ByVal dwCount As Long) As Integer
Declare Function A821_AD_INT_Start Lib "A821.DLL" _
      (ByVal wType As Integer, ByVal Ch As Integer, _
       ByVal Gain As Integer,  ByVal c1 As Integer, _
       ByVal c2 As Integer) As Integer
Declare Function A821_AD_INT_Stop Lib "A821.DLL" () As Integer
Declare Function A821_GetIntCount Lib "A821.DLL" _
      (dwVal As Long) As Integer
Declare Function A821_GetBuffer Lib "A821.DLL" _
      (ByVal dwNum As Integer, wBuffer As Integer) As Integer
Declare Function A821_GetFloatBuffer Lib "A821.DLL" _
      (ByVal dwNum As Integer, fBuffer As Single) As Integer
```

```
' Function of Interrupt
Declare Function A821_IntInstall Lib "A821.DLL" _
        (ByVal wBase As Integer, ByVal wIrq As Integer, _
         hEvent As Long, ByVal dwCount As Long) As Integer
Declare Function A821_IntStart Lib "A821.DLL" _
        (ByVal wType As Integer, ByVal Ch As Integer, _
         ByVal Gain As Integer,  ByVal c1 As Integer, _
         ByVal c2 As Integer) As Integer
Declare Function A821_IntGetCount Lib "A821.DLL" _
        (dwVal As Long) As Integer
Declare Function A821_IntGetHexBuf Lib "A821.DLL" _
        (ByVal dwNum As Integer, wBuffer As Integer) As Integer
Declare Function A821_IntGetFloatBuf Lib "A821.DLL" _
        (ByVal dwNum As Integer, fBuffer As Single) As Integer
Declare Function A821_IntStop Lib "A821.DLL" () As Integer
Declare Function A821_IntRemove Lib "A821.DLL" () As Integer

' Function of Channel-Scan with Polling
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 wBase As Integer, ByVal wCardType As Integer, _
         wBuf As Integer, ByVal wNumPerCh As Integer) As Integer
Declare Function A821_ChScan_PollingFloat Lib "A821.DLL" _
        (ByVal wBase As Integer, ByVal wCardType As Integer, _
         fBuf As Single, ByVal wNumPerCh As Integer) As Integer

' Function of Channel-Scan with Interrupt
Declare Function A821_ChScan_IntInstall Lib "A821.DLL" _
        (ByVal wBase As Integer, ByVal wIrq As Integer, _
         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
```

2.3 USING DELPHI

2.3.1 Delphi Demo Result :



2.3.2 A821.PAS

```
unit A821;

interface

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

const

 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;
 A821_INTERRUPT_MODE     = 6;

//*** 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;
```

```
 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_AdError1               = 100;
 A821_AdError2               = -200.0;
 A821_InstallBufError        = 10;
 A821_AllocateMemoryError    = 11;
 A821_CardTypeError          = 12;
 A821_TimeoutError           = 13;
 A821_OtherError             = 14;

 A821_IntStopError           =19;
 A821_IntInstallEventError   =20;
 A821_ConfigCodeError        =23;
 A821_BufferFull             =24;
 A821_NoChannelToScan        =25;
 A821_IntInstallChannelError =26;
 A821_IntInstallConfigError  =27;

// Function of Driver
function      A821_DriverInit:Word; StdCall;
procedure     A821_DriverClose; StdCall;
function      A821_DELAY(wBase,wDownCount:Word):Word; StdCall;
function      A821_Check_Address(wBase:Word):Word; StdCall;

// 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 Counter
procedure A821_SetCounter(dwBase:LongInt; wCounterNo:WORD;
          bCounterMode:WORD; wCounterValue:LongInt); StdCall;
function A821_ReadCounter(dwBase:LongInt; wCounterNo:WORD;
          bCounterMode:WORD):LongInt; StdCall;
```

```
// Function of DI/DO
function        A821_DI(wBase:Word):Word; StdCall;
procedure       A821_DO(wBase,wHexValue: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 AD
// Please uses A821_AD_Float(), A821_AD_Hex(),
//          A821_Fast_AD_Float(), A821_Fast_AD_Hex() function
function  A821_AD(wBase,wChannel,wConfig,wType:Word):Single; StdCall;
function  A821_Fast_AD(wBase:WORD):Single; StdCall;

// Function of AD
function  A821_AD_Float(wBase, wChannel, wConfig, wType:Word;
        var fVal:Single):WORD; StdCall;
function  A821_AD_Hex(wBase, wChannel, wConfig, wType:Word;
        var wVal:WORD):WORD; StdCall;
function  A821_ADs_Hex(wBase,wChannel,wConfig,wType:WORD;
        wBuf:PWord; wCount:WORD):Word; StdCall;
function  A821_ADs_Float(wBase,wChannel,wConfig,wType:Word;
        fBuf:PSingle; wCount:Word):Word; StdCall;

procedure       A821_Fast_SetChGain(wBase:WORD; wChannel:WORD;
                wConfig:WORD; wCardType:WORD); StdCall;
function        A821_Fast_AD_Float
                (wBase:WORD; var fVal:Single):WORD; StdCall;
function        A821_Fast_AD_Hex
                (wBase:WORD; var wVal:WORD):WORD; StdCall;

function        A821_Hex2Float(wHex:WORD; wCardType:WORD;
                wConfig:WORD; var fVal:Single):WORD; StdCall;

// Function of DA
procedure A821_DA(wBase,wHexValue:Word); StdCall;
procedure A821_Uni5_DA(wBase:Word; fValue:Single); StdCall;
procedure A821_Uni10_DA(wBase:Word;fValue:Single); StdCall;
```

```
// Function of Interrupt
// Please uses the newest function set A821_IntXXXXX series functions
// ***************************************************************
function  A821_InstallIrq(wBase,wIrq:Word; var hEvent:LongInt;
          dwCount:LongInt):Word; StdCall;
function  A821_AD_INT_Start
          (wCardType, Ch,Gain,c1,c2:Word):Word; StdCall;
function  A821_AD_INT_Stop:Word; StdCall;
function  A821_GetIntCount(var dwVal:LongInt):Word; StdCall;
function  A821_GetBuffer(dwNum:LongInt; wBuffer:PWord):Word; StdCall;
function  A821_GetFloatBuffer(dwNum:LongInt; fBuffer:PSingle):Word;
StdCall;

// Function of Interrupt
function  A821_IntInstall(wBase,wIrq:Word; var hEvent:LongInt;
          dwCount:LongInt):Word; StdCall;
function  A821_IntStart(wCardType, Ch,Gain,c1,c2:Word):Word; StdCall;
function  A821_IntGetCount(var dwVal:LongInt):Word; StdCall;
function  A821_IntGetHexBuf(dwNum:LongInt; wBuffer:PWord):Word; StdCall;
function  A821_IntGetFloatBuf
          (dwNum:LongInt; fBuffer:PSingle):Word; StdCall;
function  A821_IntStop:Word; StdCall;
function  A821_IntRemove:Word; StdCall;

// Function of Channel-Scan with Polling
procedure A821_ChScan_Clear;
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(wBase:WORD; wCardType:WORD;
          wBuf:PWORD; wNumPerCh:WORD):WORD; StdCall;
function A821_ChScan_PollingFloat(wBase:WORD; wCardType:WORD;
          fBuf:PSingle; wNumPerCh:WORD):WORD; StdCall;

// Function of Channel-Scan with Interrupt
function A821_ChScan_IntInstall(wBase:WORD; wIrq:WORD;
          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

uses math;

// Driver functions
function  A821_DELAY;                   external 'A821.DLL' name
          'A821_DELAY';
function  A821_Check_Address;           external 'A821.DLL' name
          'A821_Check_Address';
function  A821_DriverInit;              external 'A821.DLL' name
          'A821_DriverInit';
procedure A821_DriverClose;             external 'A821.DLL' name
          'A821_DriverClose';

// Test functions
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 of Counter
procedure A821_SetCounter;              external 'A821.DLL' name
          'A821_SetCounter';
function  A821_ReadCounter;             external 'A821.DLL' name
          'A821_ReadCounter';

// DI/DO functions
procedure     A821_DO;                  external 'A821.DLL' name 'A821_DO';
function      A821_DI;                  external 'A821.DLL' name 'A821_DI';

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';

// AD functions
// Please uses A821_AD_Float(), A821_AD_Hex(),
//          A821_Fast_AD_Float(), A821_Fast_AD_Hex() function
function  A821_AD;                      external 'A821.DLL' name 'A821_AD';
function  A821_Fast_AD;                 external 'A821.DLL' name
          'A821_Fast_AD';
```

```
// AD functions
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';

procedure A821_Fast_SetChGain;          external 'A821.DLL' name
          'A821_Fast_SetChGain';
function  A821_Fast_AD_Float;           external 'A821.DLL' name
          'A821_Fast_AD_Float';
function  A821_Fast_AD_Hex;             external 'A821.DLL' name
          'A821_Fast_AD_Hex';

function  A821_Hex2Float;               external 'A821.DLL' name
          'A821_Hex2Float';

// DA functions
procedure A821_DA;                      external 'A821.DLL' name 'A821_DA';
procedure A821_Uni5_DA;                 external 'A821.DLL' name
          'A821_Uni5_DA';
procedure A821_Uni10_DA;                external 'A821.DLL' name
          'A821_Uni10_DA';

// Interrupt functions
// Please uses the newest function set A821_IntXXXXX series functions
// ******************************************************************
function  A821_InstallIrq;              external 'A821.DLL' name
          'A821_InstallIrq';
function  A821_GetBuffer;               external 'A821.DLL' name
          'A821_GetBuffer';
function  A821_GetFloatBuffer;          external 'A821.DLL' name
          'A821_GetFloatBuffer';
function  A821_GetIntCount;             external 'A821.DLL' name
          'A821_GetIntCount';
function  A821_AD_INT_Start;            external 'A821.DLL' name
          'A821_AD_INT_Start';
function  A821_AD_INT_Stop;             external 'A821.DLL' name
          'A821_AD_INT_Stop';
```

```
// Interrupt functions
function  A821_IntInstall;                  external 'A821.DLL' name
          'A821_IntInstall';
function  A821_IntStart;                    external 'A821.DLL' name
          'A821_IntStart';
function  A821_IntGetCount;                 external 'A821.DLL' name
          'A821_IntGetCount';
function  A821_IntGetHexBuf;                external 'A821.DLL' name
          'A821_IntGetHexBuf';
function  A821_IntGetFloatBuf;              external 'A821.DLL' name
          'A821_IntGetFloatBuf';
function  A821_IntStop;                     external 'A821.DLL' name
          'A821_IntStop';
function  A821_IntRemove;                   external 'A821.DLL' name
          'A821_IntRemove';

// Function of Channel-Scan with Polling
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';

// Function of Channel-Scan with Interrupt
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.
```

3. FUNCTION DESCRIPTION

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

1. The Driver functions
2. The Test functions
3. The Counter functions
4. The DI/DO functions
5. The D/A functions
6. The AD Polling functions
7. The AD Interrupt functions
8. The AD Channel-Scan Polling functions
9. The AD Channel-Scan Interrupt functions

The functions of driver listing as follows:

1. A821_DriverInit
2. A821_DriverClose
3. A821_Check_Address
4. A821_DELAY

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 test listing as follows:

1. A821_SetCounter
2. A821_ReadCounter

The functions of DI/DO listing as follows:

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

The functions of D/A Conversion listing as follows:

1. A821_DA
2. A821_Uni5_DA
3. A821_Uni10_DA

The functions of A/D Polling Conversion listing as follows:

1. A821_AD_Float
2. A821_AD_Hex
3. A821_ADs_Hex
4. A821_ADs_Float
5. A821_Fast_SetChGain
6. A821_Fast_AD_Float
7. A821_Fast_AD_Hex
8. A821_Hex2Float

The functions of A/D Interrupt Conversion listing as follows:

1. A821_IntInstall
2. A821_IntStart
3. A821_GetHexBuf
4. A821_GetFloatBuf
5. A821_IntGetCount
6. A821_IntStop
7. A821_IntRemove

The functions of A/D Polling Conversion with Channel-Scan :

1. A821_ChScan_Clear
2. A821_ChScan_Add
3. A821_ChScan_Set
4. A821_ChScan_PollingFloat
5. A821_ChScan_PollingHex

The functions of A/D Interrupt Conversion with Channel-Scan :

1. A821_ChScan_IntInstall
2. A821_ChScan_IntStart
3. A821_ChScan_GetHexBuf
4. A821_ChScan_GetFloatBuf
5. A821_ChScan_IntGetCount
6. A821_ChScan_IntStop
7. A821_ChScan_IntRemove

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 ERROR CODE

Error Code	Descriptions
A821_NoError	OK!
A821_DriverOpenError	Fail to open the device driver. Please check does the card have plug in your system and check does the device driver have installed. Or, please try to plug the card in the other slot and try to re-install the device driver.
A821_DriverNoOpen	Users have to call the A821_DriverInit() function before calling into other A821 functions.
A821_GetDriverVersionError	Fail to communication with device driver. Please check does the driver have installed? Or, try to re-install driver again.
A821_InstallIrqError	Fail to install the ISR with the specified IRQ/DMA number. Please check does the driver have installed? Check does the IRQ/IO address and DMA resources conflicts with other device? And check the system's resource and free some resource.
A821_ClearIntCountError	Fail to communication with device driver. Please check does the driver have installed? Or, try to re-install driver again.
A821_GetIntCountError	Fail to communication with device driver. Please check does the driver have installed? Or, try to re-install driver again.
A821_GetBufferError	Fail to communication with device driver. Please check does the driver have installed? Or, try to re-install driver again.
A821_AllocateMemoryError	Fail to allocate memory for data buffer. Please check your system's resource and free some memory.
A821_CardTypeError	The CardType should be 0: A821PGL or 1:A821PGH

A821_TimeoutError	For A/D (Analog Input) functions, the DLL functions waiting for A/D converter to complete the operation. The Max. time to waiting is 500ms. Please check your hardware settings of Base address. Try to plug the card in other slot.
A821_IntStopError	Fail to communication with driver, or fail to stop the interrupt. Please check does the driver have installed? Or, try to re-install driver again.
A821_IntInstallEventError	Fail to install the event-object into device driver. Please check does the driver have installed? And check the system's resource and free some resource. Or, try to re-install driver again.
A821_ConfigCodeError	For valid configuration code, please refer to Section "1.2 Range Configuration".
A821_BufferFull	The buffer size of the Channel-Scan List is 100. Program can't add channels more than 100.
A821_NoChannelToScan	Before calling the Channel-Scan Polling or Interrupt, users have to setting the channels into the Channel-Scan List by calling related functions. Please refer to A821_ChScan_Clear(), A821_ChScan_Add() and A821_ChScan_Set() functions.
A821_IntInstallChannelError	Fail to copy the channels of Channel-Scan List into device driver. Please check does the driver have installed? And check the system's resource and free some resource. Or, try to re-install driver again.
A821_IntInstallConfigError	Fail to copy the configuration-code of Channel-Scan List into device driver. Please check does the driver have installed? And check the system's resource and free some resource. Or, try to re-install driver again.

3.2 DRIVER FUNCTIONS

3.2.1 A821_DriverInit

- **Description :**
  This subroutine will open the device driver. Programs have to call this function once before calling into others functions.

- **Syntax :**
  WORD A821_DriverInit (void);

- **Parameter :**
  None

- **Return :**
  Please refer to "Section 3.1 Error Code".

3.2.2 A821_DriverClose

- **Description :**
  This subroutine will close the device driver.

- **Syntax :**
  void A821_DriverClose (void);

- **Parameter :**
  None

- **Return :**
  None

3.2.3 A821_DELAY

- **Description :**
  This subroutine will delay **wDownCount** (machine dependent timer).
  2000 count = 1ms, 1 count = 0.5us

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

- **Parameter :**
  wBase : [In] I/O port base address, for example, 0x220
  wDownCount : [In] Number to delay, 1 count = 0.5us

- **Return :**
  Please refer to "Section 3.1 Error Code".

3.2.4 A821_Check_Address

- **Description :**
  This subroutine will detect the A-821PGH/L in I/O base address = **wBase**. This subroutine will perform one A/D conversion, if success → find a A-821PGH/L.

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

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

- **Return :**
  Please refer to "Section 3.1 Error Code".

3.3 TEST FUNCTIONS

3.3.1 A821_SHORT_SUB_2

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

- **Syntax :**
  short SHORT_SUB_2(short nA, short nB);

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

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

3.3.2 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 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

3.3.3 A821_Get_DLL_Version

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

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

- **Parameter :**
  None

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

3.3.4 A821_GetDriverVersion

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

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

- **Parameter :**
  wDriverVersion : [Out] return the driver-version.
  When wDriverVerion=0x210 → version 2.10

- **Return :**
  Please refer to "Section 3.1 Error Code".

3.4 COUNTER FUNCTION

3.4.1 A821_SetCounter

- **Description:**
  This subroutine will set the 8254 counter mode and value.

- **Syntax:**
  void A821_SetCounter(WORD wBase, WORD wCounterNo,
  WORD bCounterMode, DWORD wCounterValue);

- **Parameter:**
  wBase : [In] I/O port base address, for example, 0x220
  wCounterNo : [In] Counter Number 1 to 2 for the 8254
  wCounterMode : [In] Counter Mode 0 to 5 for the 8254
  wCounterValue : [In] Counter Value 0 to 65535 for the 8254

- **Return:**
  None

3.4.2 A821_ReadCounter

- **Description:**
  This subroutine will read the 8254 counter value.

- **Syntax:**
  DWORD A821_ReadCounter(WORD wBase, WORD wCounterNo,
  WORD bCounterMode);

- **Parameter:**
  wBase : [In] I/O port base address, for example, 0x220
  wCounterNo : [In] Counter Number 1 to 2 for the 8254
  wCounterMode : [In] Counter Mode 0 to 5 for the 8254

- **Return:**
  Return the counter's value and only the lower WORD is valid.

3.5 DI/DO FUNCTION

3.5.1 A821_DI

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

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

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

- **Return :**
  16 bits data read from the digital input port

3.5.2 A821_DO

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

- **Syntax :**
  void A821_DO(WORD wBase, WORD wHexValue);

- **Parameter :**
  wBase : [In] I/O port base address, for example, 0x220
  wHexValue : [In] 16 bits data send to digital output port

- **Return :**
  None

3.5.3 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 bits data send to I/O port

- **Return :**
  None

3.5.4 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 bits data send to I/O port

- **Return :**
  None

3.5.5 A821_InputByte

- **Description :**
  This subroutine will input the 8 bits 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

3.5.6 A821_InputWord

- **Description :**
  This subroutine will input the 16 bits 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.

3.6 D/A FUNCTION

3.6.1 A821_DA

- **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 :**
  void A821_DA(WORD wBase, WORD wHexValue);

- **Parameter :**
  wBase : [In] I/O port base address, for example, 0x220
  wHexValue : [In] 12 bits data send to D/A converter

- **Return :**
  None

3.6.2 A821_Uni5_DA

- **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) , JP10/JP11 (Bipolar or Unipolar).** The software **can not detect** the output range of D/A converter. This subroutine can be used only when the jumpers settings are **: Unipolar , -5v .** The **output range is between 0.0v and 5.0v.** Please refer to hardware manual to setting jumpers.

- **Syntax :**

void A821_Uni5_DA(WORD wBase, WORD wChannel, float fValue);

- **Parameter :**

wBase	: [In] I/O port base address, for example, 0x220
wChannel	: [In] D/A channel number, validate for 0
fValue	: [In] 12 bits data send to D/A converter

- **Return :**

None

3.6.3 A821_Uni10_DA

- **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) , JP10/JP11 (Bipolar or Unipolar).** The software **can not detect** the output range of D/A converter. This subroutine can be used only when the jumpers settings are **: Unipolar , -10v .** The **output range is between 0.0v and 10.0v.** Please refer to hardware manual to setting jumpers.

- **Syntax :**

void A821_Uni10_DA(WORD wBase, WORD wChannel, float fValue);

- **Parameter :**

wBase	: [In] I/O port base address, for example, 0x220
wChannel	: [In] D/A channel number, validate for 0
fValue	: [In] 12 bits data send to D/A converter

- **Return :**

None

3.7 A/D FUNCTION

3.7.1 A821_AD_Float

- **Description :**

This subroutine will perform a 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 wBase,   WORD wChannel,
                   WORD wConfig,  WORD wType, float *fVal);
```

- **Parameter :**

wBase	: [In] I/O port base address, for example, 0x220
wChannel	: [In] A/D channel number,
wConfig	: [In] Configuration code, refer to "Section 1.2 Range Configuration" for detail information
wType	: [In] 0 → A-821PGL, 1→ A-821PGH
fVal	: [Out] Return the AD value in floating format.

- **Return :**

Please refer to "Section 3.1 Error Code".

3.7.2 A821_AD_Hex

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

- **Syntax :**
  WORD A821_AD_Hex(WORD wBase, WORD wChannel,
  WORD wConfig, WORD wType, WORD *wVal);

- **Parameter :**
  wBase : [In] I/O port base address, for example, 0x220
  wChannel : [In] A/D channel number,
  wConfig : [In] Configuration code, refer to "Section 1.2 Range Configuration" for detail information
  wType : [In] 0 → A-821PGL, 1→ A-821PGH
  wVal : [Out] Return the AD value in floating format.

- **Return :**
  Please refer to "Section 3.1 Error Code".

3.7.3 A821_ADs_Hex

- **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. 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.

- **Syntax :**

```
WORD A821_ADs_Hex(WORD wBase, WORD wChannel,
                  WORD wConfig, WORD wType, WORD wBuf[],
                  WORD wCount);
```

- **Parameter :**

wBase	: [In] I/O port base address, for example, 0x220
wChannel	: [In] A/D channel number
wConfig	: [In] Configuration code, refer to "Section 1.2 Range Configuration" for detail information
wType	: [In] 0 → A-821PGL, 1→ A-821PGH
wBuf	: [Out] Starting address of the data buffer (in WORD format) Users have to allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. Users can analyze these data from the buffer after calling this function.
wCount	: [In] Number of A/D conversions will be performed

- **Return :**

Please refer to "Section 3.1 Error Code".

3.7.4 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.

- **Syntax :**

```
WORD A821_ADs_Float(WORD wBase, WORD wChannel,
                    WORD wConfig, WORD wType, float fBuf[],
                    WORD wCount);
```

- **Parameter :**

wBase	: [In] I/O port base address, for example, 0x220
wChannel	: [In] A/D channel number
wConfig	: [In] Configuration code, refer to "Section 1.2 Range Configuration" for detail information
wType	: [In] 0 → A-821PGL, 1→ A-821PGH
fBuf	: [Out] Starting address of the data buffer **(in float format)** Users have to allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. Users can analyze these data from the buffer after calling this function.
wCount	: [In] Number of A/D conversions will be performed

- **Return :**

Please refer to "Section 3.1 Error Code".

3.7.5 A821_Fast_SetChGain

- **Description :**
  This subroutine will setting the Channel number and Gain-Code (Config-Code) for the function "A821_Fast_AD()".

  A821_AD_Float() = A821_Fast_SetChGain() + A821_Fast_AD_Float()

- **Syntax :**
  void A821_Fast_SetChGain(WORD wBase, WORD wChannel,
  WORD wConfig, WORD wCardType);

- **Parameter :**
  wBase : [In] I/O port base address, for example, 0x220
  wChannel : [In] A/D channel number,
  wConfig : [In] Configuration code, refer to "Section 1.2 Range Configuration" for detail information
  wCardType : [In] 0 → A-821PGL, 1→ A-821PGH

- **Return :**
  None

3.7.6 A821_Fast_AD_Float

- **Description :**
  This subroutine will perform a A/D conversion by polling. The A/D converter is 12 bits for A821PGH/L. The function A821_Fast_SetChGain() must be called once before calling this function.

  A821_AD_Float() = A821_Fast_SetChGain() + A821_Fast_AD_Float()

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

- **Parameter :**
  wBase : [In] I/O port base address, for example, 0x220
  fVal : [Out] Return the AD value in floating format.

- **Return :**
  Please refer to "Section 3.1 Error Code".

3.7.7 A821_Fast_AD_Hex

- **Description :**
  This subroutine will perform a A/D conversion by polling. The A/D converter is 12 bits for A821PGH/L. The function A821_Fast_SetChGain() must be called once before calling this function.

  A821_AD_Hex() = A821_Fast_SetChGain() + A821_Fast_AD_Hex()

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

- **Parameter :**
  wBase : [In] I/O port base address, for example, 0x220
  wVal : [Out] Return the AD value in WORD format.

- **Return :**
  Please refer to "Section 3.1 Error Code".

3.7.8 A821_Hex2Float

- **Description:**
  Compute the Hex(WORD) to floating value.

- **Syntax:**
  WORD A821_Hex2Float(WORD wHex, WORD wCardType,
  WORD wConfig, float *fVal);

- **Parameter:**
  wHex : [In] The Hex(WORD) value need to compute.
  wCardType : [In] 0 → A-821PGL, 1→ A-821PGH
  wConfig : [In] Refer to "Section 1.2 Range Configuration".
  fVal : [Out] Return the value in floating format.

- **Return:**
  Please refer to "Section 3.1 Error Code".

3.8 AD IINTERRUPT FUNCTION

3.8.1 A821_IntInstall

- **Description :**
  This subroutine will install ISR (interrupt Service Routine) for a specific IRQ number n and allocate the data buffer. For more detail information of using interrupt please refer to "Section 3.8.8 Interrupt Architecture".

- **Syntax :**
  WORD A821_IntInstall
  (WORD wBase, WORD wIrq, HANDLE *hEvent, DWORD dwCount);

- **Parameter :**
  wBase : [In] The I/O port base address for A821 card.
  wIrq : [In] The IRQ number .
  hEvent : [In] The handle of event object that created by user.
  dwCount : [In] The desired A/D entries count for interrupt transfer.

- **Return :**
  Please refer to "Section 3.1 Error Code".

3.8.2 A821_IntStart

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

- **Syntax :**
  WORD A821_IntStart(WORD wCardType, WORD wChannel,
  WORD wConfig, WORD c1, Word c2)

- **Parameter :**
  wCardType : [In] 0: for A821PGL, 1: for A821PGH
  wChannel : [In] the channel number for AD operation.
  wConfig : [In] the Config-Code, refer to "Section 1.2 Range Configuration".
  c1,c2 : [In] The value setting to 8254 counter1 and counter2, and the sampling rate is 2M/(c1*c2)

- **Return :**
  Please refer to "Section 3.1 Error Code".

3.8.3 A821_IntGetCount

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

- **Syntax :**
  WORD A821_IntGetCount(DWORD *dwVal)

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

- **Return :**
  Please refer to "Section 3.1 Error Code".

3.8.4 A821_IntGetHexBuf

- **Description :**
  This subroutine will copy the transferred interrupted data into the user's buffer.

- **Syntax :**
  WORD A821_IntGetHexBuf(DWORD dwNum, WORD wBuffer[])

- **Parameter :**
  dwNum : [In] The entry no to transfer.
  wBuffer : [Out] The address of wBuffer. (in WORD format)
  Users have to allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. Users can analyze these data from the buffer after calling this function.

- **Return :**
  Please refer to "Section 3.1 Error Code".

3.8.5 A821_IntGetFloatBuf

- **Description :**
  This subroutine will copy the transferred interrupted data into the user's buffer.

- **Syntax :**
  WORD A821_IntGetFloatBuf(DWORD dwNum, float fBuffer[])

- **Parameter :**
  dwNum : [In] the entry no to transfer.
  fBuffer : [Out] the address of fBuffer (in floating format)
  Users have to allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. Users can analyze these data from the buffer after calling this function.

- **Return :**
  Please refer to "Section 3.1 Error Code".

3.8.6 A821_IntStop

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

- **Syntax :**
  WORD A821_IntStop(void)

- **Parameter :**
  None

- **Return :**
  Please refer to "Section 3.1 Error Code".

3.8.7 A821_IntRemove

- **Description :**
  This subroutine will remove the ISR (Interrupt Service Routine) and free the data buffer. Users should get the data buffer before calling this function.

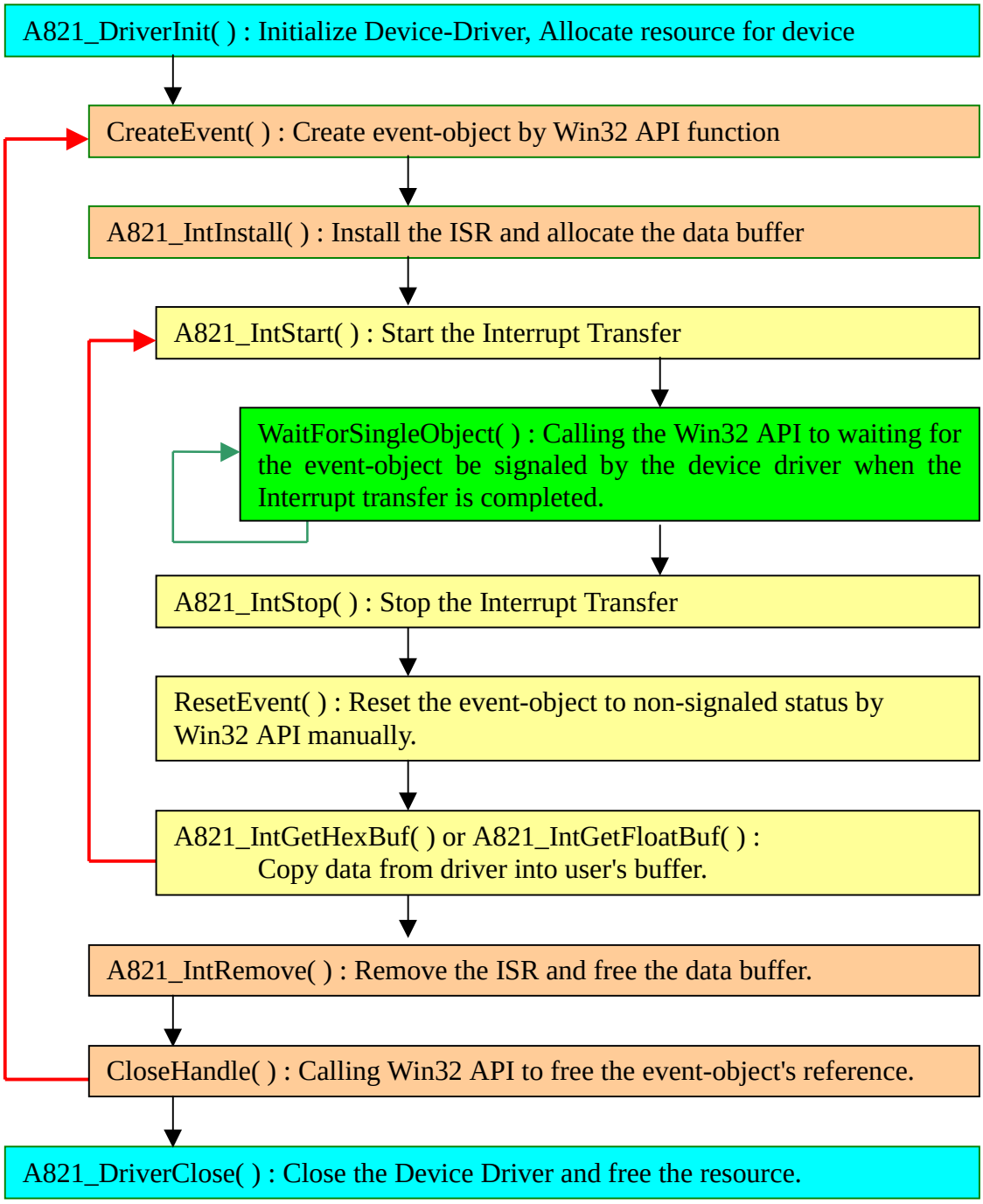
- **Syntax :**
  WORD A821_IntRemove(void)

- **Parameter :**
  None

- **Return :**
  Please refer to "Section 3.1 Error Code".

3.8.8 Interrupt Architecture

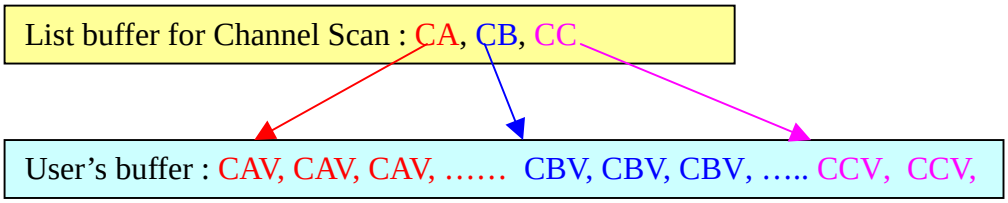
The 3.8.1 to 3.8.7 are these functions to perform the A/D conversion with interrupt transfer. The flow chart to program these functions is given as follows:



3.9 AD WITH CHANNEL SCAN

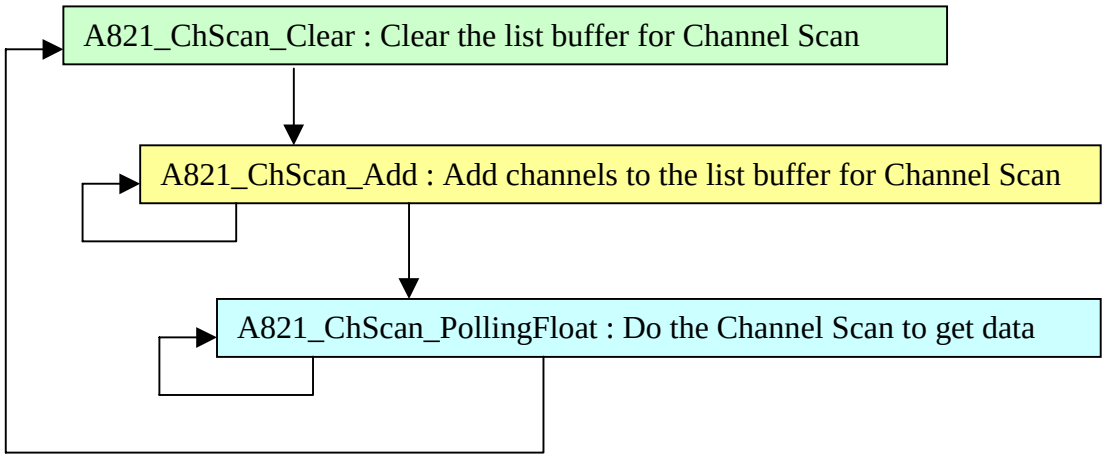
3.9.1 Introduction

The user can specify channels into a list buffer. Other functions 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:
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:



3.9.2 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

3.9.3 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 1.2 Range Configuration".

- **Return:**
  Refer to "Section 3.1 Error Code".

3.9.4 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 1.2 Range Configuration".
  wChNum : [In] Total channels pass into and to be scanned.

- **Return:**
  Refer to "Section 3.1 Error Code".

3.9.5 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() or A821_ChScan_Set() functions to setup the list buffer for Channel Scan. Please refer to the "Section 3.9.1 Introduction" for more information.

- **Syntax:**

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

- **Parameter:**

  wBase : [In] the I/O port base address for A821 card.
  WCardType : [In] 0: A-821L 1: A-821H
  wBuf : [Out] Starting address of the data buffer (WORD format)
  Users have to allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. Users 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:**

  Refer to "Section 3.1 Error Code".

3.9.6 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() or A821_ChScan_Set() functions to setup the list buffer for Channel Scan. Please refer to the "Section 3.9.1 Introduction" for more information.

- **Syntax:**

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

- **Parameter:**

  wBase : [In] the I/O port base address for A821 card.
  WCardType : [In] 0: A-821L 1: A-821H
  fBuf : [Out] Starting address of the data buffer (floating format)
  Users have to allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. Users 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.

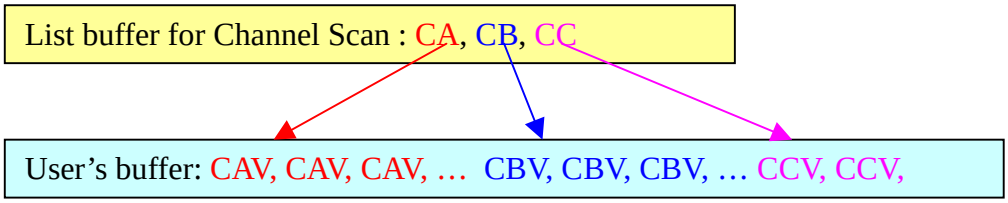
- **Return:**

  Refer to "Section 3.1 Error Code".

3.10 AD INTERRUPT, CHANNEL SCAN FUNCTION

3.10.1 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.



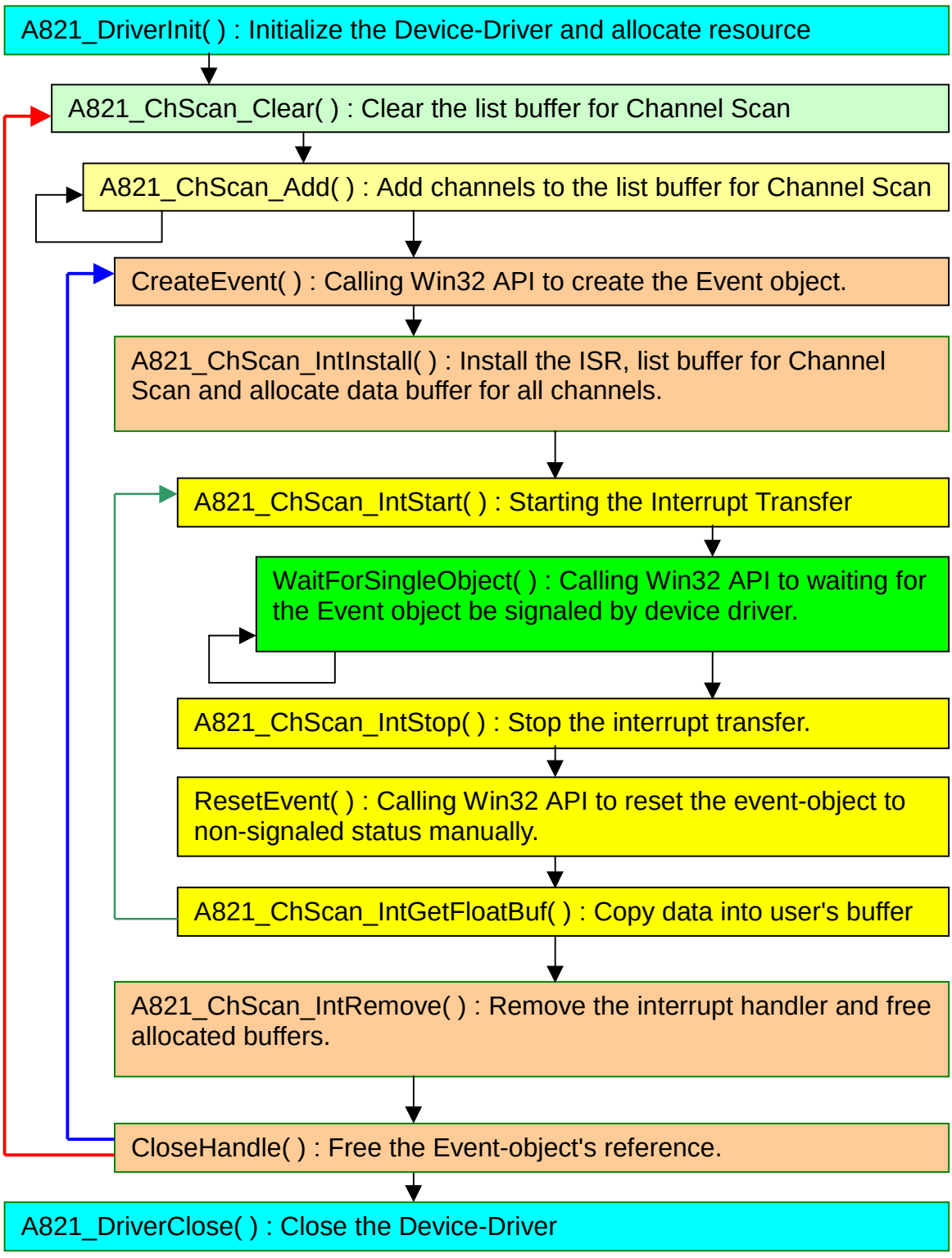
The data will be saved into the following style:

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:



3.10.2 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()" or "A821_ChScan_Set() functions to setup the list buffer for Channel Scan firstly. For more detail information of using interrupt please refer to "Section 3.10.1 Introduction".

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

- **Parameter:**
  wBase : [In] the I/O port base address for A821 card.
  wIrq : [In] the IRQ number n.
  hEvent : [In] The Event handle that created by the user.
  dwNumPerCh : [In] The desired A/D count for every channels to transfer.

- **Return:**
  Refer to "Section 3.1 Error Code".

3.10.3 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:**
  Refer to "Section 3.1 Error Code".

3.10.4 A821_ChScan_IntStop

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

- **Syntax:**
  WORD A821_ChScan_IntStop(void);

- **Parameter:**
  None

- **Return:**
  Refer to "Section 3.1 Error Code".

3.10.5 A821_ChScan_IntRemove

- **Description:**
  This subroutine will remove the ISR (Interrupt Service Routine) and free the buffers.

- **Syntax:**
  WORD A821_ChScan_IntRemove(void);

- **Parameter:**
  None

- **Return:**
  Refer to "Section 3.1 Error Code".

3.10.6 A821_ChScan_IntGetCount

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

- **Syntax:**
  WORD A821_ChScan_IntGetCount(DWORD *dwVal)

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

- **Return:**
  Refer to "Section 3.1 Error Code".

3.10.7 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).
  Users have to allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. Users can analyzes these data from the buffer after calling this function.
  Buffer size = Total-Channels * dwNumPerCh * sizeof(WORD)

- **Return:**
  Refer to "Section 3.1 Error Code".

3.10.8 A821_ChScan_IntGetFloatBuf

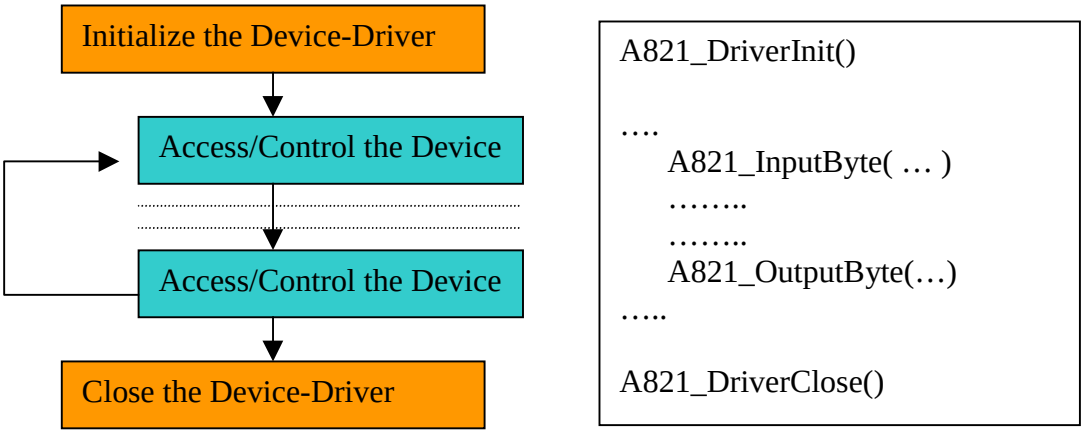
- **Description:**
  This subroutine will copy the transferred interrupted data into the user's buffer.

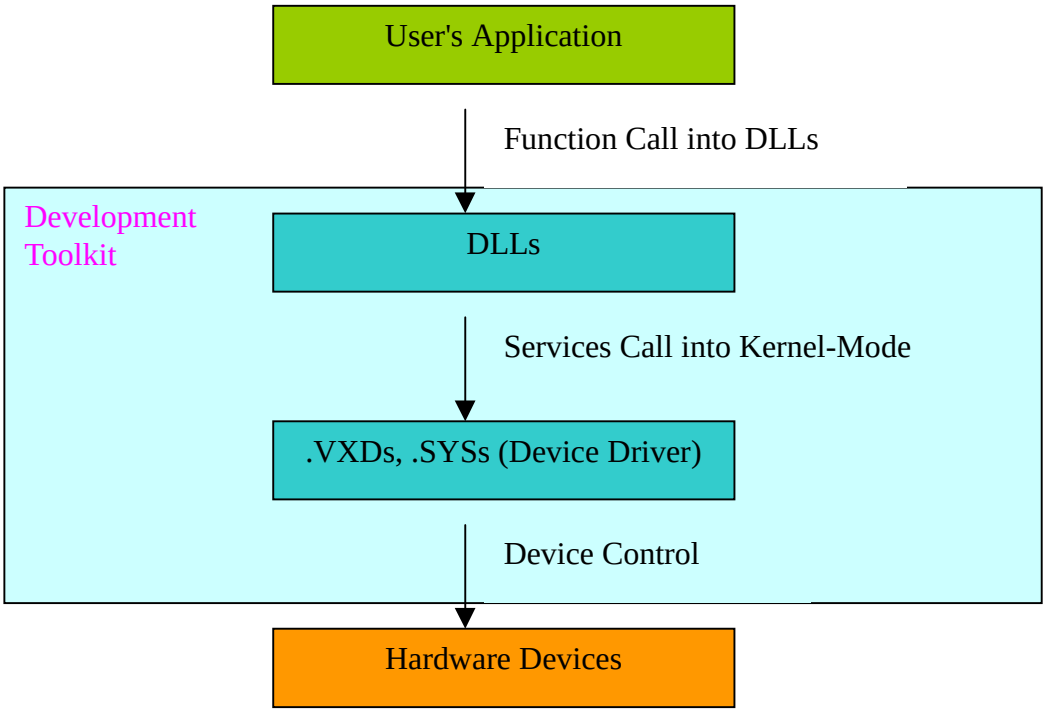
- **Syntax:**
  WORD A821_ChScan_IntGetFloatBuf(float fBuf[])

- **Parameter:**
  fBuf : [Out] The address of fBuf(float format).
  Users have to allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. Users can analyze these data from the buffer after calling this function.
  Buffer size = Total-Channels * dwNumPerCh * sizeof(float)

- **Return:**
  Refer to "Section 3.1 Error Code".

4. PROGRAM ARCHITECTURE





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