PET-7H16M
Quick Use Manual

Version 1.0.3, January 2023

Service and usage information for

PET-7H16M

Written by Sean

Edited by Anna Huang

Document Description

This document will guide users to quickly use the PET-7H16M module.

1. **Preceding procedure**：PET-7H16M network settings.
2. **Introduction to acquisition and trigger mode**: Brief description of PET-7H16M data acquisition and trigger mode.
3. **Simple to use**：Simple operation through the HSDAQ Utility, collecting PET-7H16M data, and displaying the data in the utility chart.
4. **Program development** :

    VC/.NET program: A brief description of the function calls process and program description according to different acquisition and trigger modes.
5. **LabVIEW development**: Simply use the process description.
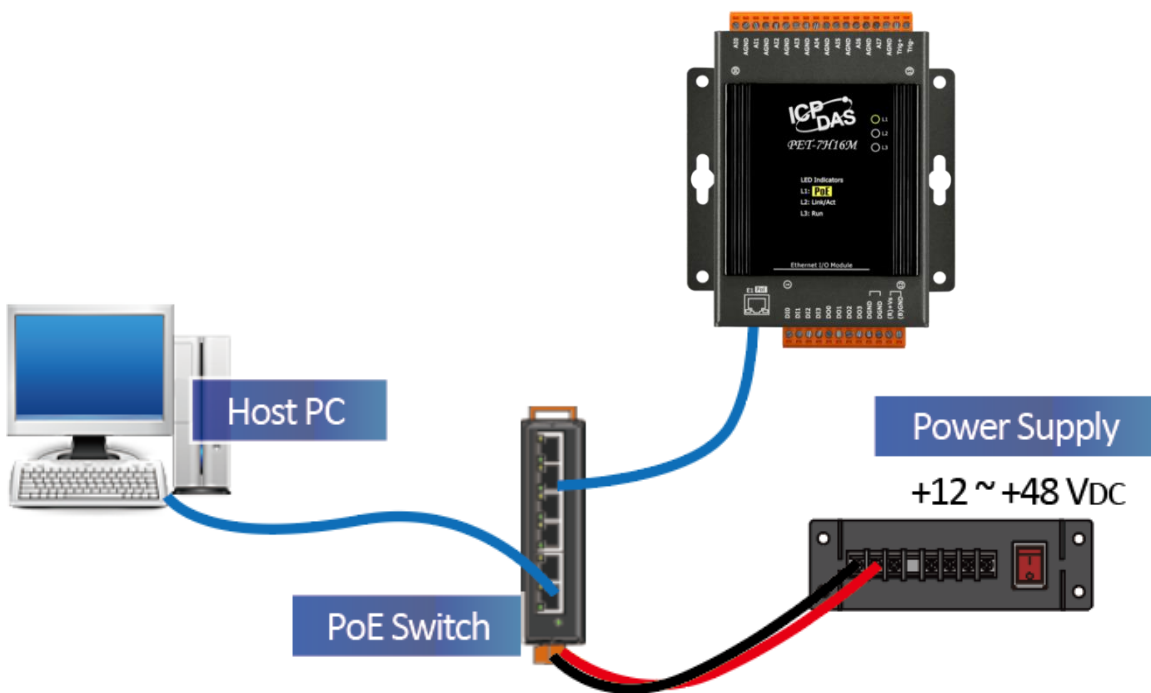
Table of Contents

Preceding Procedure

There are two ways for the PET-7H16M module getting the power. One is through Ethernet by a PoE switch; the other is as usual through wiring by an external power.
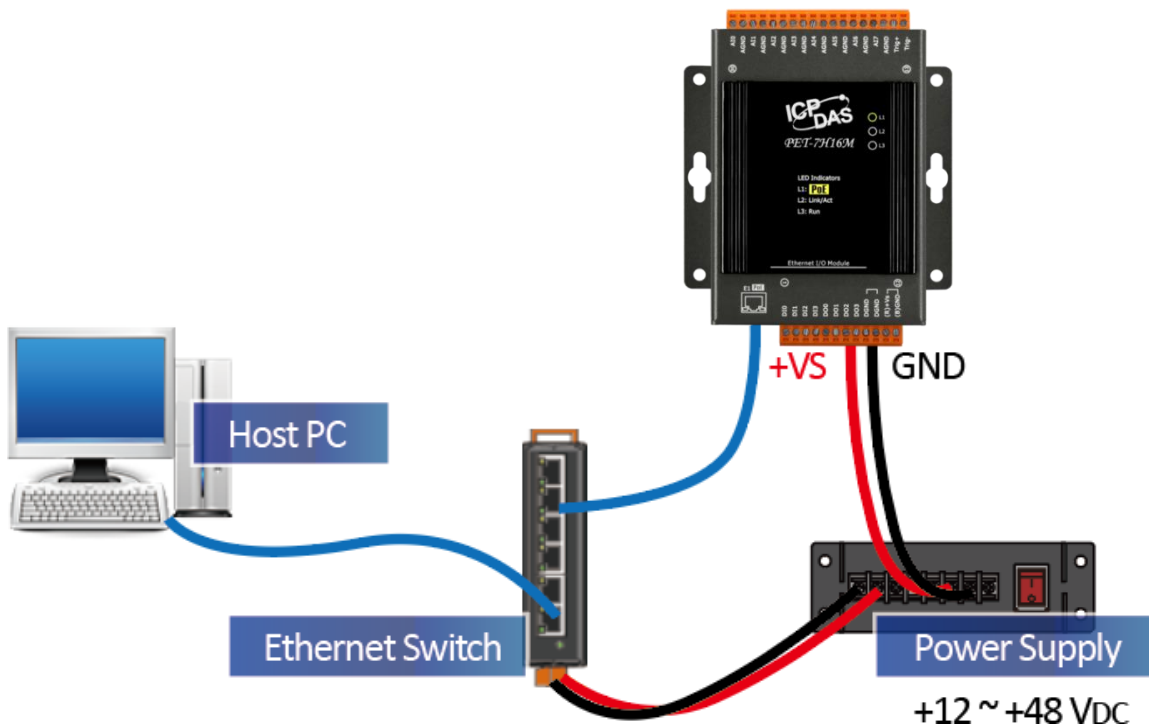
POE Power Supply

➢ Connect PC to the Ethernet port via the PoE switch.

➢ Connect the power supply to the PoE switch, which supplies power to the PET-7H16M.

External Power Supply

➢ Connect PC and PET-7H16M to the Ethernet Switch.

➢ Connect the power supply to the switch and PET-7H16M. (The external power supply should be in the range of +12 VDC to 48 VDC)

Network Settings

The factory default settings for the PET-7H16 network are as follows:

| Item	Default
IP Address	192.168.255.1
Subnet Mask	255.255.0.0
Gateway	192.168.0.1

The way to modify the PET-7H16M network configuration settings is through the HSDAQ Utility.

Steps

1. **Obtain and install the HSDAQ Utility in the following path**
    https://www.icpdas.com/tw/download/show.php?num=2327

2. **Run HSDAQ Utility**

3. **Press choose "Search" from the "Tools" menu**
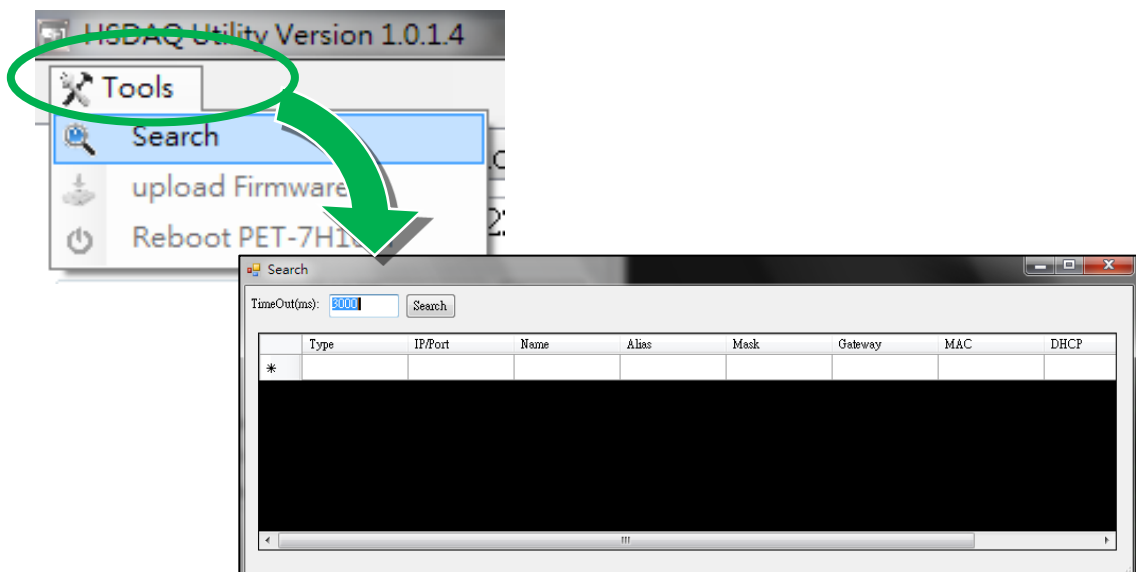
4. **The searched PET-7H16M module will be displayed on the screen of this program.**
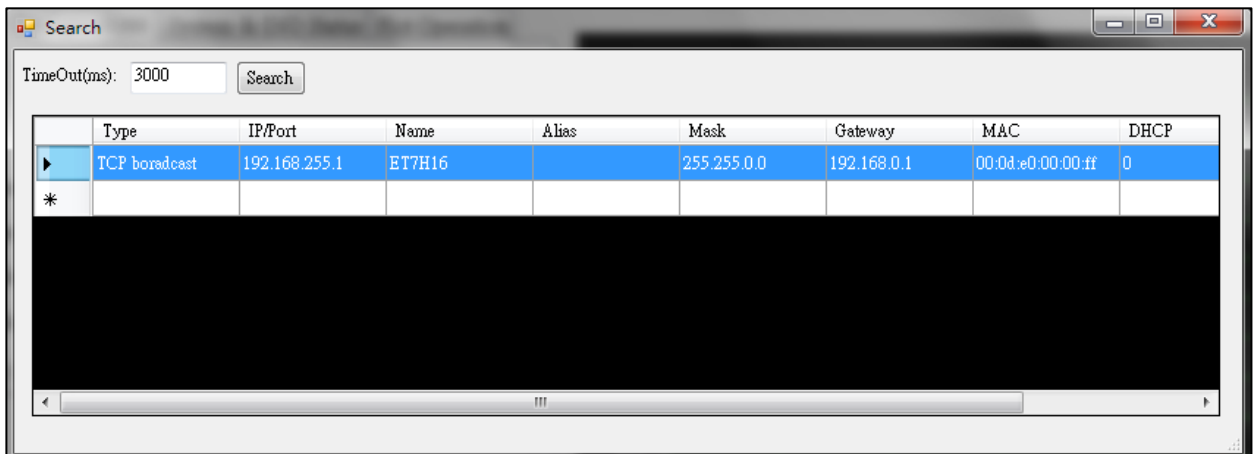
5. **Select the PET-7H16M to be set, and the following setting screen will appear.**

6. **Write the new setting value in the IP address/Mask/Gateway/Alias field and press the "OK" button**

7. **Wait for the PET-7H16M to reboot. (After reboot, the new setting value will take effect)**

Introduction to Acquisition and Trigger Mode

PET-7H16M Data Acquisition and Trigger Mode Description

Two data acquisition methods and multiple trigger modes provide high-speed acquisition of input values of the AI channel. The figure below shows the corresponding effective operating frequency range for the acquisition and trigger modes.

Trigger \ Acquisition	Continuous	N Sample
Software AD	1 ~ 30 KHz	1 Hz ~ 200 KHz
External CLK AD	1 ~ 30 KHz	-
Post-Trigger	-	1 Hz ~ 200 KHz
Pre-Trigger	-	1 Hz ~ 200 KHz

Data Acquisition and Transmission Mode

1. Continuous transmission
   (Maximum sampling rate of 30 kHz per channel)

2. After collecting N data samples, the data is transferred to the Host PC
   (Maximum sampling rate of 200 kHz per channel)
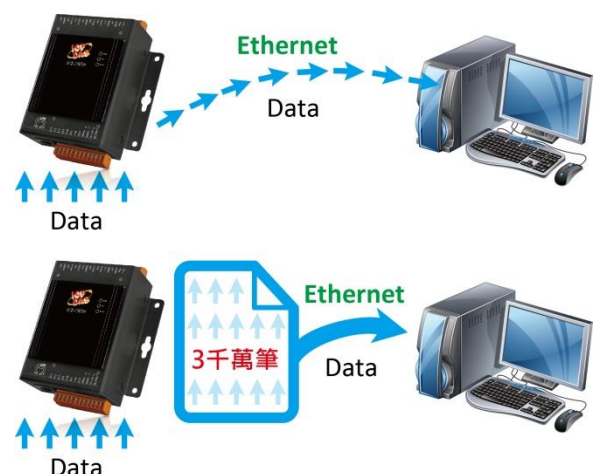
A/D Trigger Mode

1. **Software A/D Data Acquisition Mode**

   The A/D acquisition parameters are configured via a command from the Host PC. The continuous A/D acquisition or the acquisition of N data samples begins after the command is triggered.

2. **External Clock Signal Synchronization A/D Conversion Data Acquisition mode**

   The speed of the A/D acquisition and the amount of data acquired are controlled by external electrical signals. A falling edge for each output waveform triggers an AD conversion
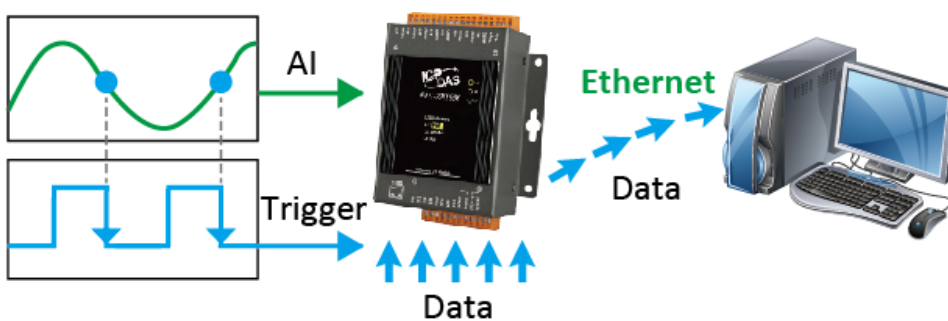
External Clock Signal Synchronization A/D Acquisition Mode

3. **External Digital Signal Event Trigger Mode**

   i. Pre-Trigger (acquisition of N data samples)

   The A/D data is continually collected and is temporarily stored in the memory on the PET-7H16M until the trigger signal is received. Once the trigger signal is received, the collected N data samples are then transferred to the Host PC.
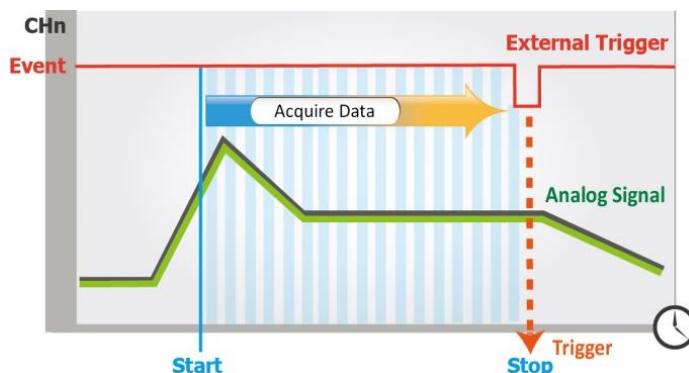
ii. Post-Trigger (acquisition of N data samples)

In this mode, the A/D acquisition of the N data samples is started once the trigger signal is received.
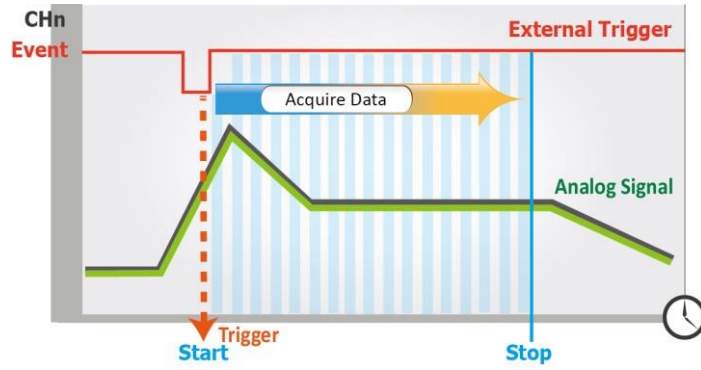
Simple to Use

The HSDAQ Utility can be used to perform simple AI high-speed acquisition operations. The collected data is automatically stored on the PC disk and the data is displayed on the chart. This tool can be used as an entry test and a simple diagnostic.

Steps

1. **Execute the HSDAQ Utility. Enter the IP address of the PET-7H16M in the following screen and click the "Connect" button.**
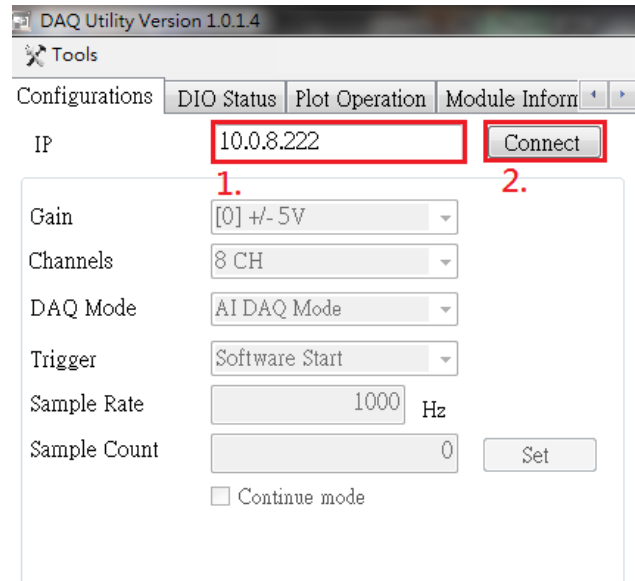
2. **Set the parameter value of AI high speed acquisition**

    2.1. Select Gain, which is +/-5V or +/-10V.

    2.2. Select the number of AI channels scanned

    2.3. Select trigger mode
        0: Software AD Trigger
        1: External CLK AD Trigger
        2: Pre-Trigger
        3: Post-Trigger

    2.4. Input Sample Rate (1 ~ 200 KHz)

2.5. Set the number of data acquisition (Sample Count)

The setting value is 0. It indicates Continuous Acquisition mode. Continue to store data continuously. Press the "Stop" button to stop the acquisition.

The setting value is greater than 0. It indicates N Sample Acquisition mode. When the number of acquisition data reaches the target count, the acquisition will stop automatically.

2.6. After the above steps are completed, press the "Set" button to write the setting value to the PET-7H16M module.
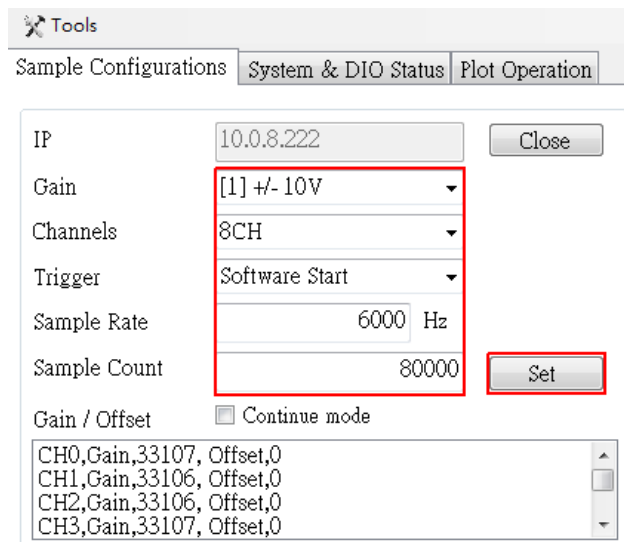
3. Start AI high speed data acquisition

3.1. Press the Start button. If the setting value of Sample Count is greater than 0, the acquisition will be stopped automatically when it reaches the set value. If the setting value of Sample Count is 0, you need to manually click Stop button to stop.

3.2. Select the folder where the data to be displayed is located, and the folder is layered to 1.IP address 2. Date.

3.3. Select which data file to display

3.4. The top of the following UI is the waveform diagram drawn by the data. Below is the data of each channel. Each page displays 1000 data.
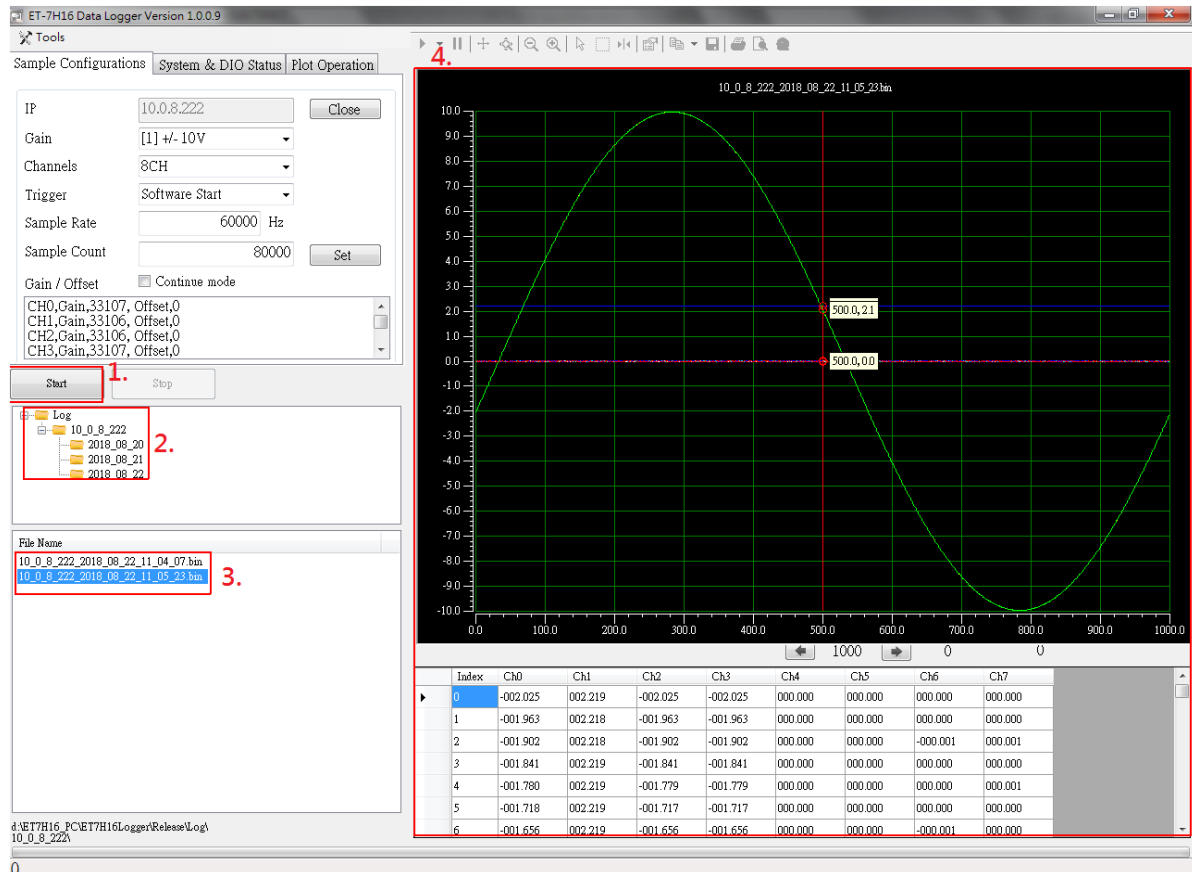
Program Development

Development Preparation

Get the library used for program development

PET-7H16M SDK library supports 32/64 bit Windows 7/8/10.

The files in PET-7H16M DAQ SDK Library

Files	Description
HSDAQ.dll	
HSDAQ.lib	Used for VC programs
HSDAQ.h	
HSDAQNet.dll	Used for .Net programs

User can get the SDK library files from web site, the file path is as below

https://www.icpdas.com/en/download/show.php?num=2326

Before using HSDAQ.dll & HSDAQNet.dll, you must install HSDAQ_SDK_package.exe to use the dll correctly. The program and SDK get the same path. Need to restart the computer after installation.
For installation steps, please refer to "PET-7H16M Standard API User Manual".

Prepare Development Tools

Install Microsoft Visual Studio 2008 or later on PC

How to use the SDK development program in the Visual Studio development environment

Refer to the HSDAQ API Reference Manual in the following path.

https://www.icpdas.com/en/download/show.php?num=2599

In Section 1.4.1, learn more about developing C# programs using SDK in Visual Studio.

In Section 1.4.2, learn more about developing VB.NET programs using SDK in Visual Studio.

In Section 1.4.3, learn more about developing VC programs using SDK in Visual Studio.

Demo program download path:

https://www.icpdas.com/en/download/show.php?num=2328

Example Description

Use the program to develop high-speed AI acquisition functions to provide various sample programs according to different trigger modes and data acquisition/transmission methods. For high-speed AI acquisition related API function description and function call flow, please refer to the the HSDAQ API Reference manual in the following path, Section 2.4 High Speed IO API.

https://www.icpdas.com/en/download/show.php?num=2599

The following two modes of high-speed acquisition examples and a sample data logger example.

1. Continuous mode acquisition and software AD trigger of high-speed acquisition
2. N sample mode acquisition and software AD trigger of high-speed acquisition
3. N sample mode acquisition and software AD trigger of data logger example.

Continuous mode acquisition and software AD trigger of high-speed acquisition

API function call process chart

HS_Device_Create	Create a connection to the device and initialize the device
↓	
HS_SetAIScanParam	Set the AI scan parameter • Sampling rate <=30 KHz • Trigger mode = 0 (0: Software trigger) • TargetCnt = 0 (0: Continuous mode)
↓	
HS_StartAIScan	Start data acquisition of scanning AI channels
↓	
HS_GetAIBufferStatus (No / Yes)	Determine how much data is in the buffer
↓ Yes	
HS_GetAIBufferHex	Read the data from the buffer
↓	
Stop (No / Yes)	Whether stop data acquisition manually
↓ Yes	
HS_StopAIScan	Stop data acquisition
↓	
HS_Device_Release	Release the device from system
↓	
Exit	Exit
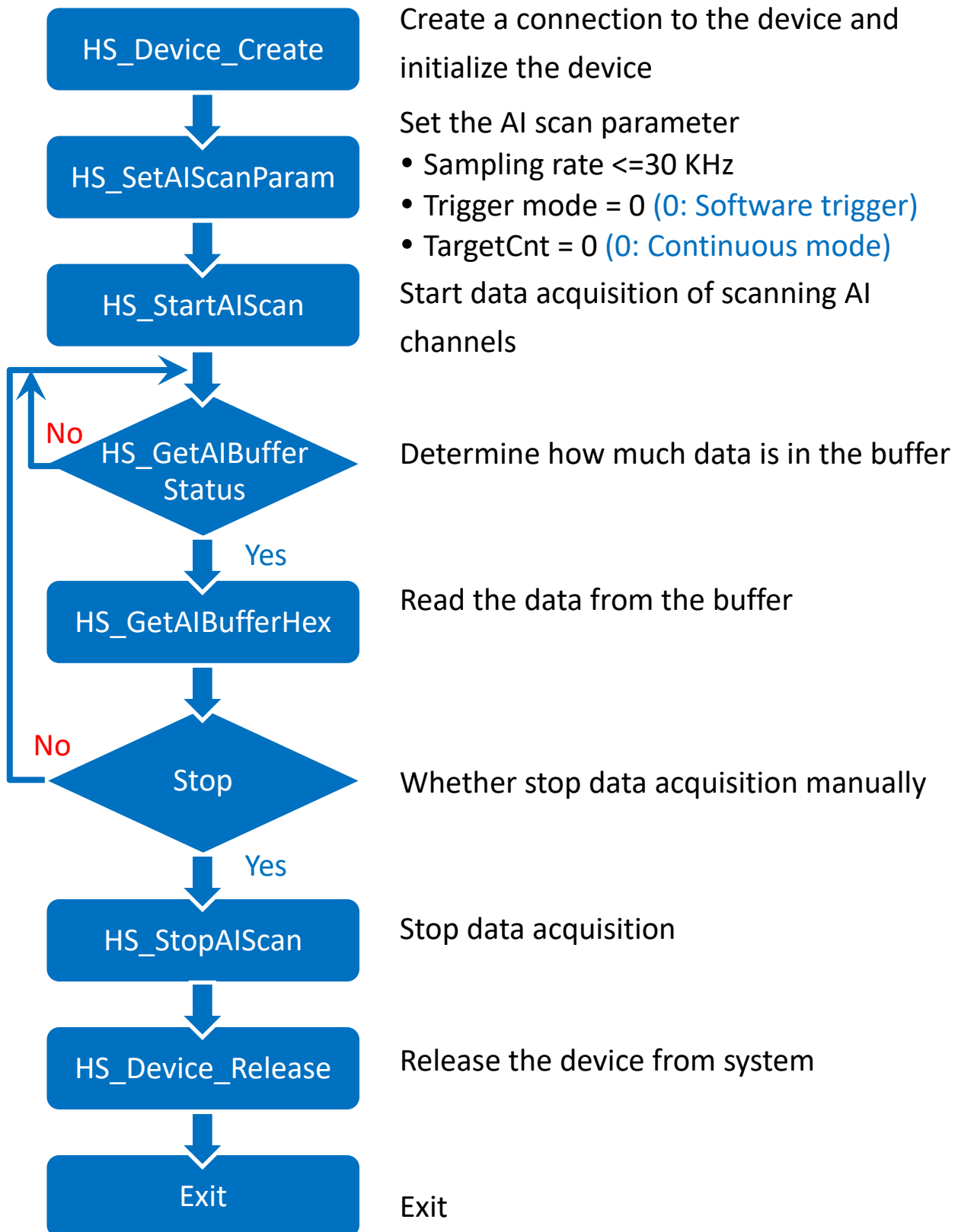
[C program Snippet code]

```c
HANDLE hHS;
WORD BufferStatus=0;
float fdataBuffer[10000];
unsigned long ulleng=0;

hHS = HS_Device_Create("192.168.1.1");
// Create a connection to the device and initialize the device

HS_SetAIScanParam(hHS, 8, 0, 0, 20000, 0, 0,0);
/*Set the AI scan parameter
•    Sampling rate =20KHz
•    Trigger mode = 0 (0: Software trigger)
•    TargetCnt =0    (0: Continuous mode)
*/
HS_StartAIScan (hHS);
// Start data acquisition of scanning AI channel

ret=HS_GetAIBufferStatus(hHS,&BufferStatus,&ulleng);
// Determine how much data is in the buffer

if(ret==false){
   printf("Error code 0x%x\r\n",HS_GetLastError());
}
else
{
   if(ulleng)
   {
   readsize=HS_GetAIBuffer(hHS,fdataBuffer, ulleng); // Read the
data from the buffer

       …
```

```
    }
}
HS_StopAIScan (hHS); // Stop data acquisition
HS_Device_Release (hHS); // Release the device from system
```

N sample mode acquisition and software AD trigger of high-speed acquisition

API function call process chart

Flow	Description
HS_Device_Create	Create a connection to the device and initialize the device
↓	
HS_SetAIScanParam	Set the AI scan parameter • Sampling rate 1 ~ 200 KHz • Trigger mode = 0 (0: Software Trigger) • TargetCnt > 0 (> 0: N Sample Mode)
↓	
HS_StartAIScan	Start data acquisition of scanning AI channels
↓	
HS_GetAIBufferStatus (No → loop back; Yes ↓)	Determine total number of samples reaches the target count
↓ Yes	
HS_GetAIBufferHex	Read the data from the buffer
↓	
HS_StopAIScan	Stop data acquisition
↓	
HS_Device_Release	Release the device from system
↓	
Exit	Exit
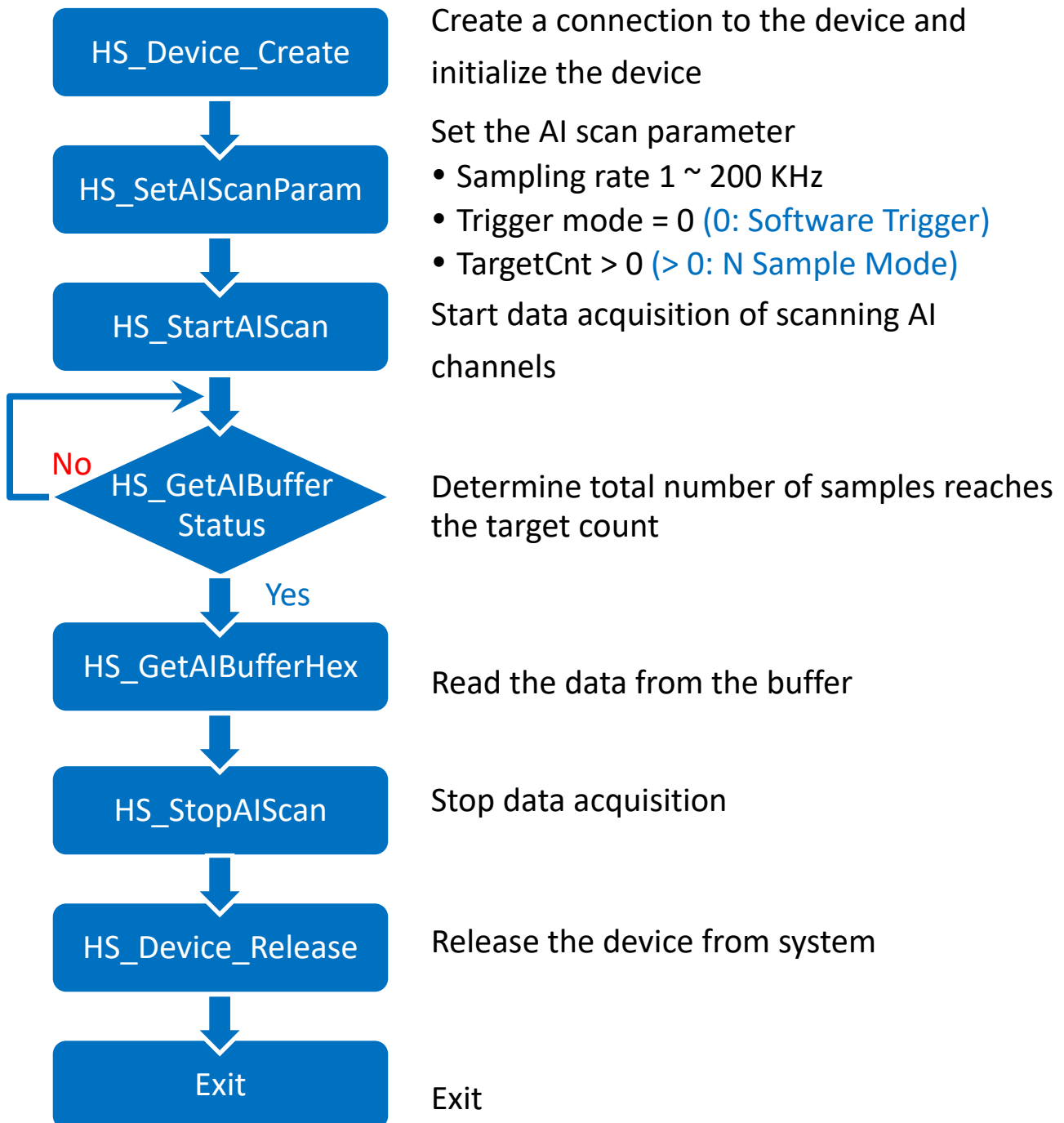
[C program Snippet code]

```c
HANDLE hHS;
WORD BufferStatus=0;
float fdataBuffer[10000];
unsigned long ulleng=0;
hHS = HS_Device_Create("192.168.1.1");
//Create a connection to the device and initialize the device

HS_SetAIScanParam(hHS, 8, 0, 0, 100000, 2000000, 0,0);
/*Set the AI scan parameter
•   Sampling rate =100KHz
•   Trigger mode = 0 (0: Software trigger)
•   TargetCnt =2000000 (2000000 samples for acquisition)
*/
HS_StartAIScan (hHS);
//Start data acquisition of scanning AI channel

ret=HS_GetAIBufferStatus(hHS,&BufferStatus,&ulleng);
// Determine how much data is in the buffer

if(ret==false){
   printf("Error code 0x%x\r\n",HS_GetLastError());
}
else
{
     if(BufferStatus>2) //AI buffer overflow
     {
      /* 2:   AD_BUF_OVERFLOW
          4:   AD_SCAN_STOP
          8:   AD_DATA_SAMPLING_TIMEOUT
      */
      break;
```
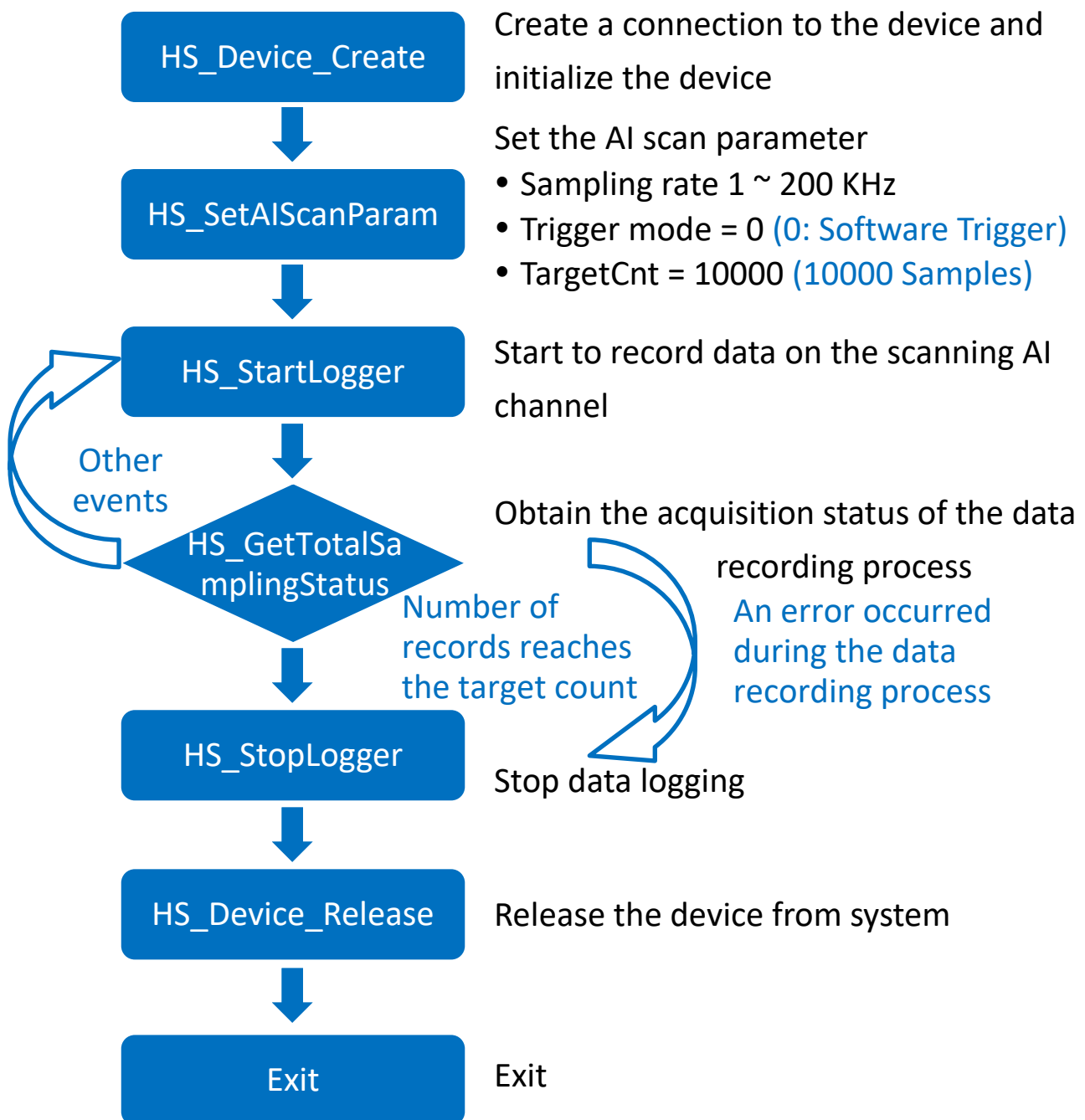
```
    }
    if(ulleng==targetCnt) //The total sample of data in the buffer
reaches the target count
    {
        unsigned long size=targetCnt;
        readsize=HS_GetAIBufferHex(hHS,(WORD *)dataBuffer,size);
        // Read the N sample of data from the buffer
    }
}
HS_StopAIScan (hHS); // Stop data acquisition
HS_Device_Release (hHS); // Release the device from system
```

N sample mode acquisition and software AD trigger of data logger example

PET-7H16M has a data logging function. The data collected by the module can be transferred to the Host PC and saved as a data log file (.bin, .txt and etcs)

API function call process chart

HS_Device_Create	Create a connection to the device and initialize the device
HS_SetAIScanParam	Set the AI scan parameter • Sampling rate 1 ~ 200 KHz • Trigger mode = 0 (0: Software Trigger) • TargetCnt = 10000 (10000 Samples)
HS_StartLogger	Start to record data on the scanning AI channel
HS_GetTotalSamplingStatus	Obtain the acquisition status of the data recording process
HS_StopLogger	Stop data logging
HS_Device_Release	Release the device from system
Exit	Exit

Other events

Number of records reaches the target count

An error occurred during the data recording process

[C program Snippet code]

```c
HANDLE hHS;
hHS = HS_Device_Create("192.168.1.1");
//Create a connection to the device and initialize the device

HS_SetAIScanParam(hHS, 8, 0, 0, 20000,2000000, 0,0);
/*Set the AI scan parameter
•    Sampling rate =20KHz
•    Trigger mode = 0 (0: Software trigger)
•    TargetCnt =2000000 (2000000 samples for acquisition)
*/
HS_StartLogger(hHS,NULL,2,0);
// Start to record data on the scanning AI

while(;;)
{
  ret=HS_GetTotalSamplingStatus(hHS,&ulleng,&triggerStatus);
  // Obtain the acquisition status of the data recording process

  if(ret==false)
  {
    printf("Error code 0x%x\r\n",HS_GetLastError());
  }
  else
  {
    if(CHECK_BIT(triggerStatus,7)==true && targetCnt>0)
  //Number of records reaches the target count
    {
       if(ulleng>=targetCnt)
       {
         ...
          break; //break the loop
```

```
        }
      }
      else
      {
        if(CHECK_BIT(triggerStatus,4) && HECK_BIT(triggerStatus,5) )
        // An error occurred during the data recording process
        {
          break; // break the loop
        }
        else
        {
          ....   //Continued in the loop }
        }
      }
    }
  }
}
HS_StopLogger(hHS); // Stop data logging
HS_Device_Release(hHS); // Release the device from system

/* Use the Data logger API function to read data log files */
HANDLE hlf;
TCHAR tcgetfulfilelPath[MAX_PATH]={0};
int ind=HS_GetAllLogFiles(NULL,1);

if(ind>0)
{
for(int i=0;i<ind;i++)
{
    hlf=HS_LogFile_Open_byIndex(i,tcgetfulfilelPath);
    DWORD samplecount;
    char startdate[32],starttime[32];
    HS_GetLogFile_AIScanSampleInfo(hlf,&samplecount,startdate,sta
```

```
rttime);
    float *fdatabuff=( float *)malloc(sizeof(float)*samplecount);
    HS_GetLogFile_AIData(hlf, 0, samplecount, fdatabuff);
    HS_LogFile_Close(hlf);
   }
}
```

LabVIEW Development

HSDAQ LabVIEW toolkit is developed for accessing the ICP DAS high-speed data acquisition devices. It provides an easy-to-use interface and simplifies the programming process. The HSDAQ LabVIEW toolkit provides an HSDAQ.llb with basic sub-VIs for controlling the HSDAQ devices, and several demo programs to demonstrate how to use the sub-Vis.

LabVIEW examples and related components can be obtained from the following path

https://www.icpdas.com/en/download/show.php?num=2327