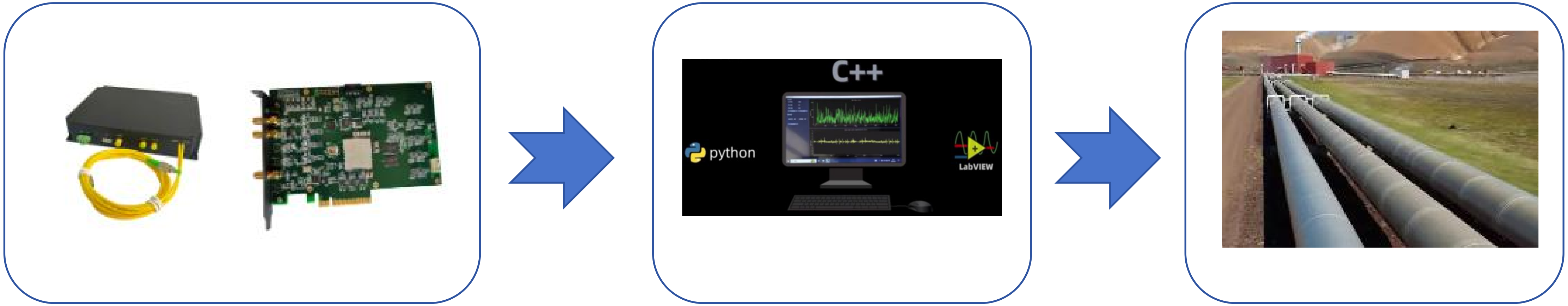
# DVS/DAS Product Introduction

- Our Business Model
- Hardware products for DVS/DAS
- About DEMO Software
- How to develop your own software
- DEMO software

# Our Business Model

Our company is the upstream manufacturer of Distributed Fiber Optic Vibration DVS/DAS system, we provide the hardware of DVS/DAS system (DVS/DAS integrated module + DAQ card), our DAQ card also provides API interface, customers need to develop their own application software program based on our API interface.



Hardware provided by YB Photonics

Customers develop their own software

Project Applications

# Hardware products for DVS/DAS

The hardware product of DVS/DAS is 3 parts, 1. DVS/DAS integrated module, 2. special DAQ card, 3. computer mainframe.

DVS/DAS integrated module products have integrated all the devices needed for detection (including lasers, AOM, EDFA, BPD and the necessary optical devices, output electrical signals); DAQ is customized for DVS/DAS applications, not a general-purpose acquisition card, built-in algorithms needed; computer mainframe is used to read the data captured by the DAQ, and subsequent information processing and software presentation. We can also help customers to integrate all these parts, that delivery is our hardware all-in-one products.

YB Photonics Offers Integrated DVS/DAS Module

YB Photonics offers specialized DAQ boards.

Customers can configure their own computer mainframe (motherboard, CPU, hard disk, power supply, etc.)

# About DEMO Software

In order to facilitate customers to view the effect of DVS/DAS system in a timely manner, we have developed a DEMO software (not open source), the software can be a simple demonstration of the function and export vibration data. If the customer wants to apply the DEMO software directly to the project, the software function is slightly sketchy, we do not recommend the practice, we recommend that customers according to the DAQ API, combined with the actual needs of their own projects to develop their own software.

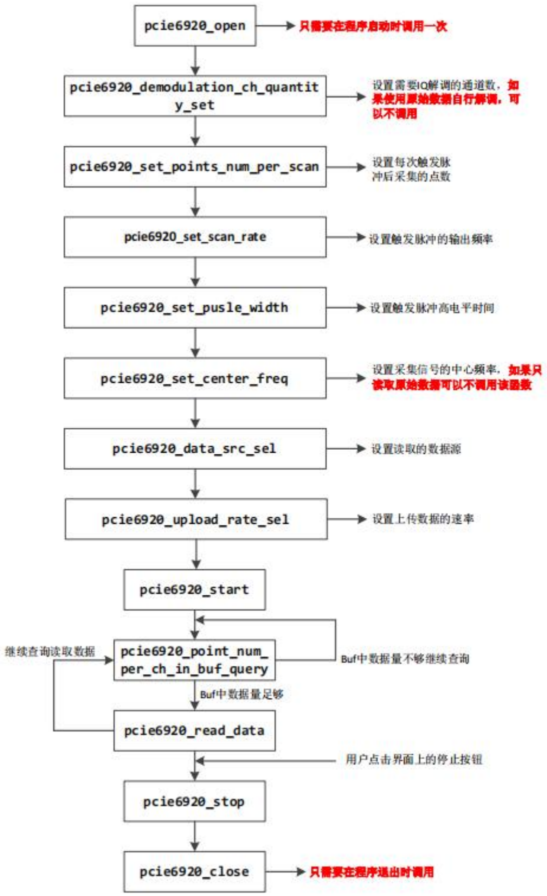
YB Photonics Provides DEMO Software

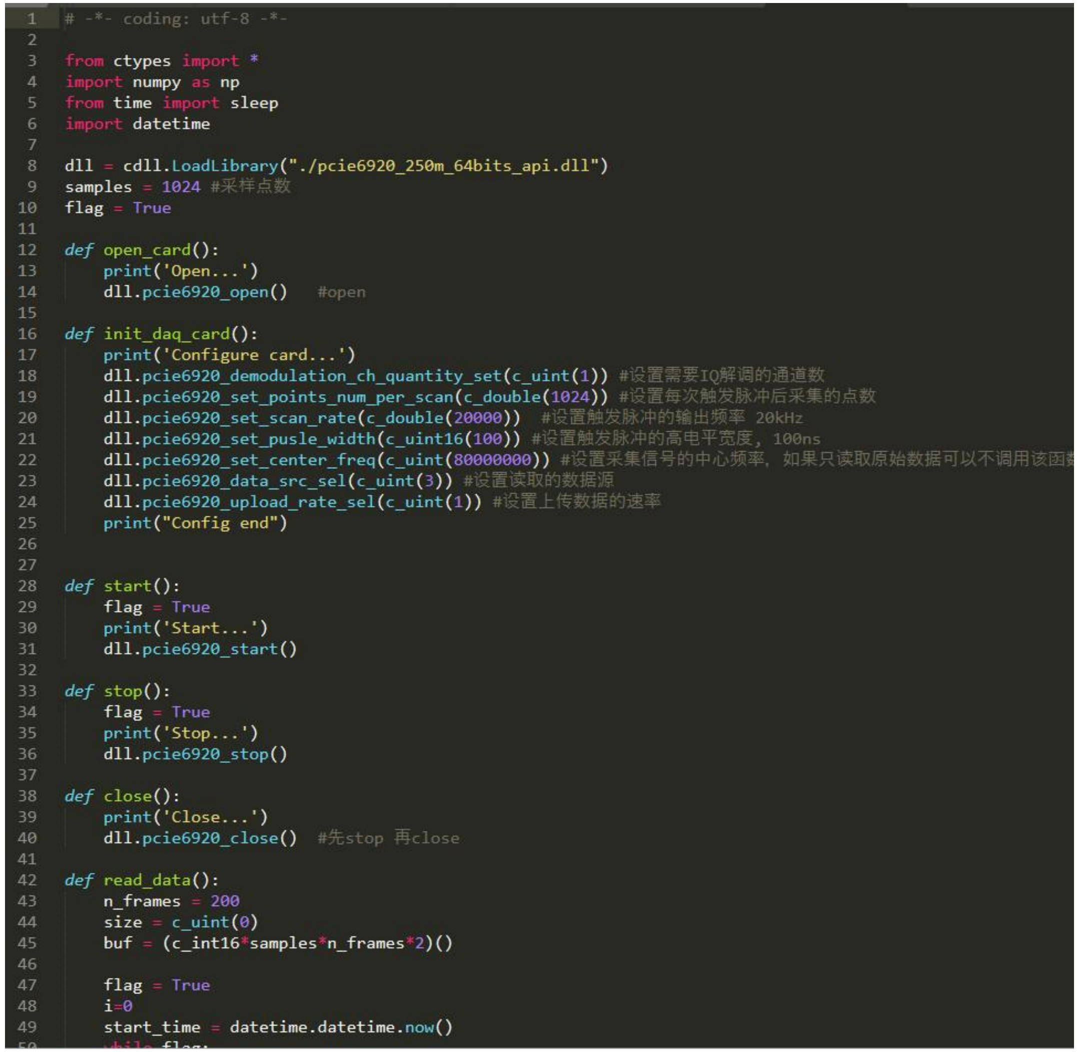
C, C++, Python, Labview etc.

Users develop software according to the DAQ API. The DAQ interface is in the form of a DLL dynamic link library, which can be used by customers in C, C++, Python, Labview and so on.

# How to develop your own software 1

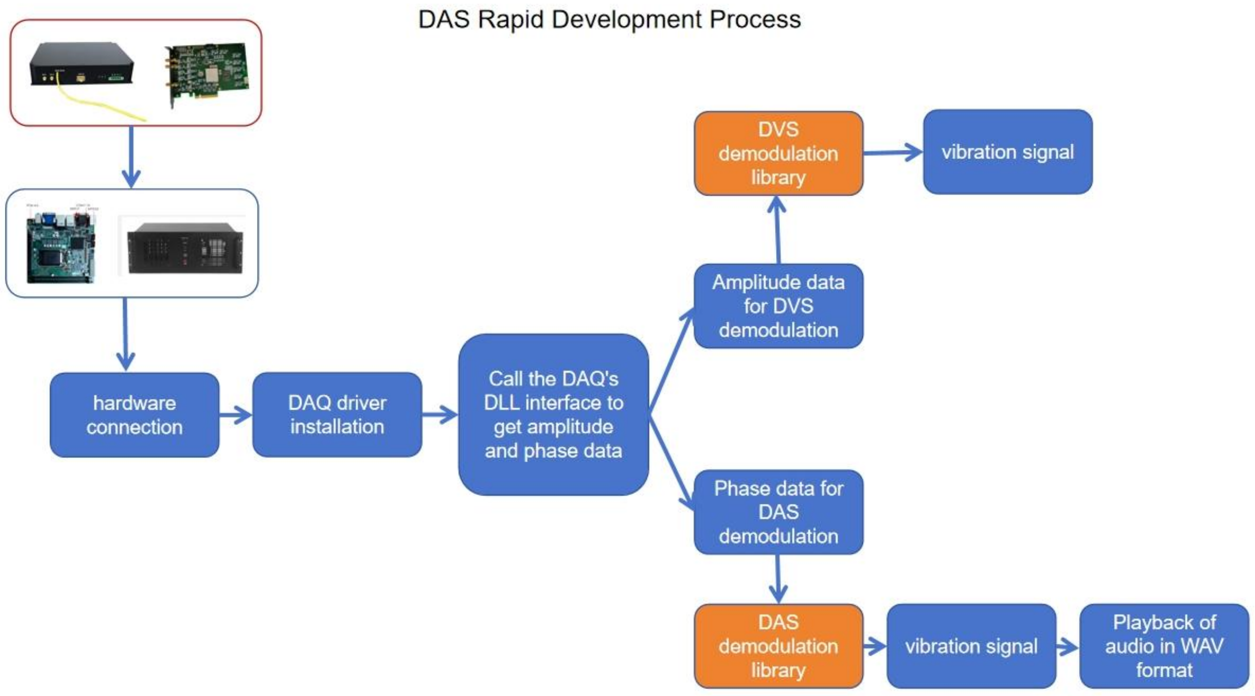
Using your favorite programming language, call the DAQ's DLL dynamic link library, and then make interface calls and read data according to the DAQ's interface documentation and data acquisition call flow.



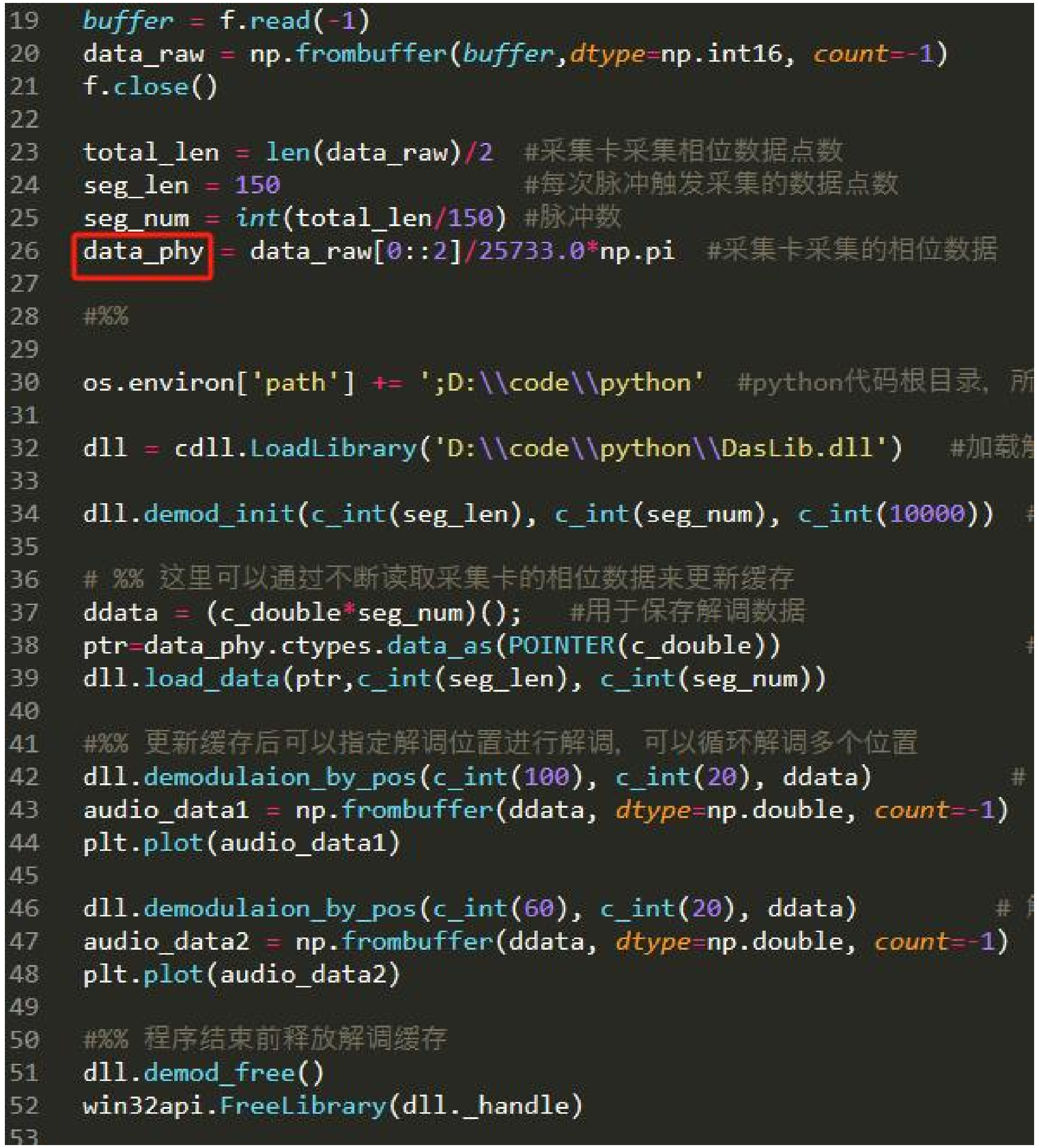

```python
# -*- coding: utf-8 -*-

from ctypes import *
import numpy as np
from time import sleep
import datetime

dll = cdll.LoadLibrary("./pcie6920_250m_64bits_api.dll")
samples = 1024 #采样点数
flag = True

def open_card():
    print('Open...')
    dll.pcie6920_open()    #open

def init_daq_card():
    print('Configure card...')
    dll.pcie6920_demodulation_ch_quantity_set(c_uint(1)) #设置需要IQ解调的通道数
    dll.pcie6920_set_points_num_per_scan(c_double(1024)) #设置每次触发脉冲后采集的点数
    dll.pcie6920_set_scan_rate(c_double(20000))  #设置触发脉冲的输出频率 20kHz
    dll.pcie6920_set_pusle_width(c_uint16(100)) #设置触发脉冲的高电平宽度, 100ns
    dll.pcie6920_set_center_freq(c_uint(80000000)) #设置采集信号的中心频率, 如果只读取原始数据可以不调用该函
    dll.pcie6920_data_src_sel(c_uint(3)) #设置读取的数据源
    dll.pcie6920_upload_rate_sel(c_uint(1)) #设置上传数据的速率
    print("Config end")


def start():
    flag = True
    print('Start...')
    dll.pcie6920_start()

def stop():
    flag = True
    print('Stop...')
    dll.pcie6920_stop()

def close():
    print('Close...')
    dll.pcie6920_close()  #先stop 再close

def read_data():
    n_frames = 200
    size = c_uint(0)
    buf = (c_int16*samples*n_frames*2)()

    flag = True
    i=0
    start_time = datetime.datetime.now()
    while flag:
```

# How to develop your own software 2

If you do DVS, you can get the vibration amplitude data directly from DAQ. If you do DAS, through DAQ you can get the amplitude and phase data after IQ demodulation, but this is not the final vibration data (can't be played as sound yet), the final audio data still need to be demodulated to get it. You can choose to demodulate by yourself, you can also choose to demodulate through our paid demodulation library (also a DLL dynamic link library).

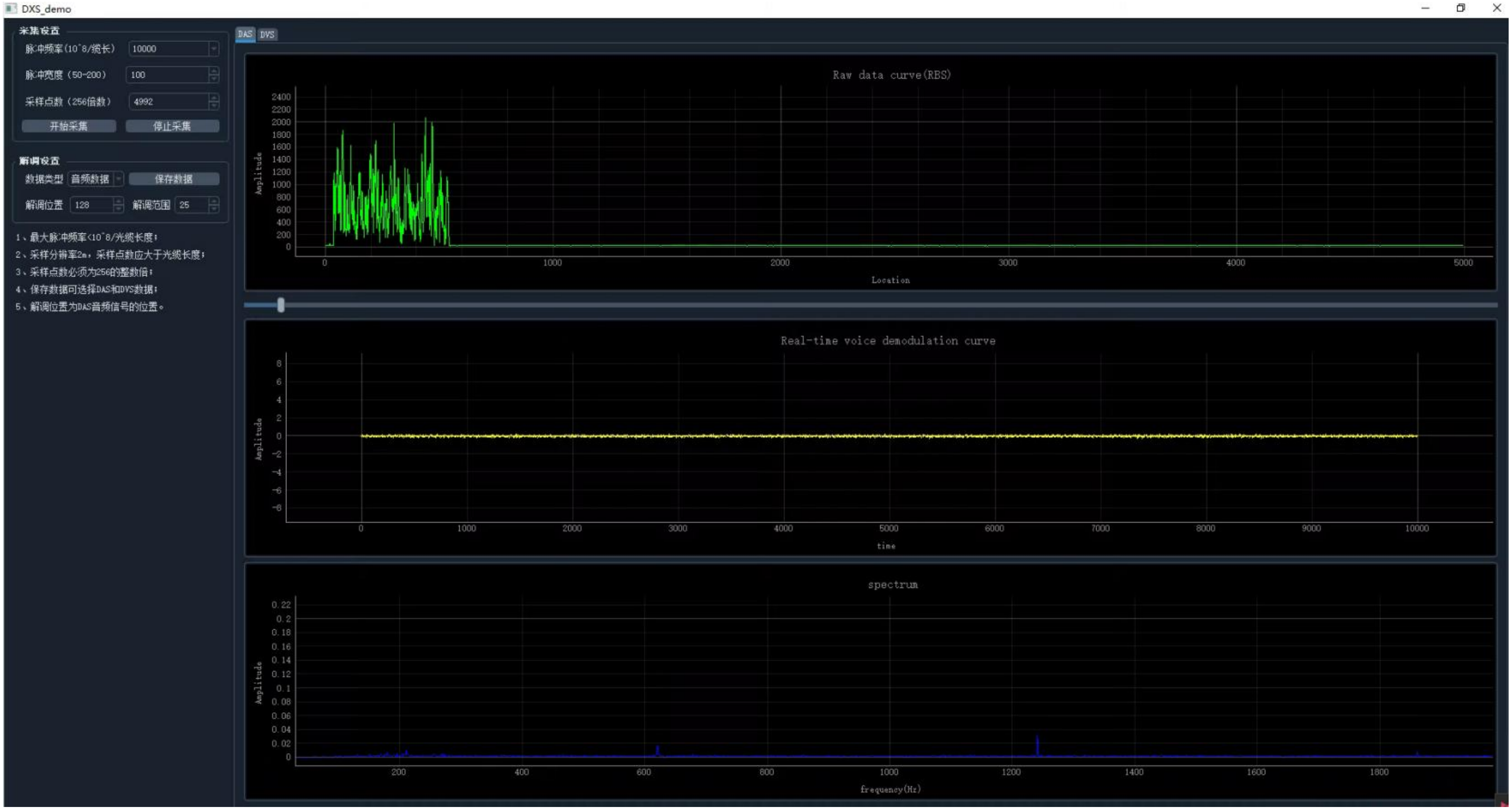

DAS Rapid Development Process

hardware connection
DAQ driver installation
Call the DAQ's DLL interface to get amplitude and phase data
Amplitude data for DVS demodulation
DVS demodulation library
vibration signal
Phase data for DAS demodulation
DAS demodulation library
vibration signal
Playback of audio in WAV format



```
buffer = f.read(-1)
data_raw = np.frombuffer(buffer,dtype=np.int16, count=-1)
f.close()

total_len = len(data_raw)/2  #采集卡采集相位数据点数
seg_len = 150                #每次脉冲触发采集的数据点数
seg_num = int(total_len/150) #脉冲数
data_phy = data_raw[0::2]/25733.0*np.pi  #采集卡采集的相位数据

#%%

os.environ['path'] += ';D:\\code\\python'  #python代码根目录, 所

dll = cdll.LoadLibrary('D:\\code\\python\\DasLib.dll')   #加载

dll.demod_init(c_int(seg_len), c_int(seg_num), c_int(10000))  #

# %% 这里可以通过不断读取采集卡的相位数据来更新缓存
ddata = (c_double*seg_num)();   #用于保存解调数据
ptr=data_phy.ctypes.data_as(POINTER(c_double))                   #
dll.load_data(ptr,c_int(seg_len), c_int(seg_num))

#%% 更新缓存后可以指定解调位置进行解调, 可以循环解调多个位置
dll.demodulaion_by_pos(c_int(100), c_int(20), ddata)             #
audio_data1 = np.frombuffer(ddata, dtype=np.double, count=-1)
plt.plot(audio_data1)

dll.demodulaion_by_pos(c_int(60), c_int(20), ddata)              #
audio_data2 = np.frombuffer(ddata, dtype=np.double, count=-1)
plt.plot(audio_data2)

#%% 程序结束前释放解调缓存
dll.demod_free()
win32api.FreeLibrary(dll._handle)
```

# DEMO software



**采集设置**

脉冲频率(10ˆ8/缆长) 10000

脉冲宽度（50-200） 100

采样点数（256倍数） 4992

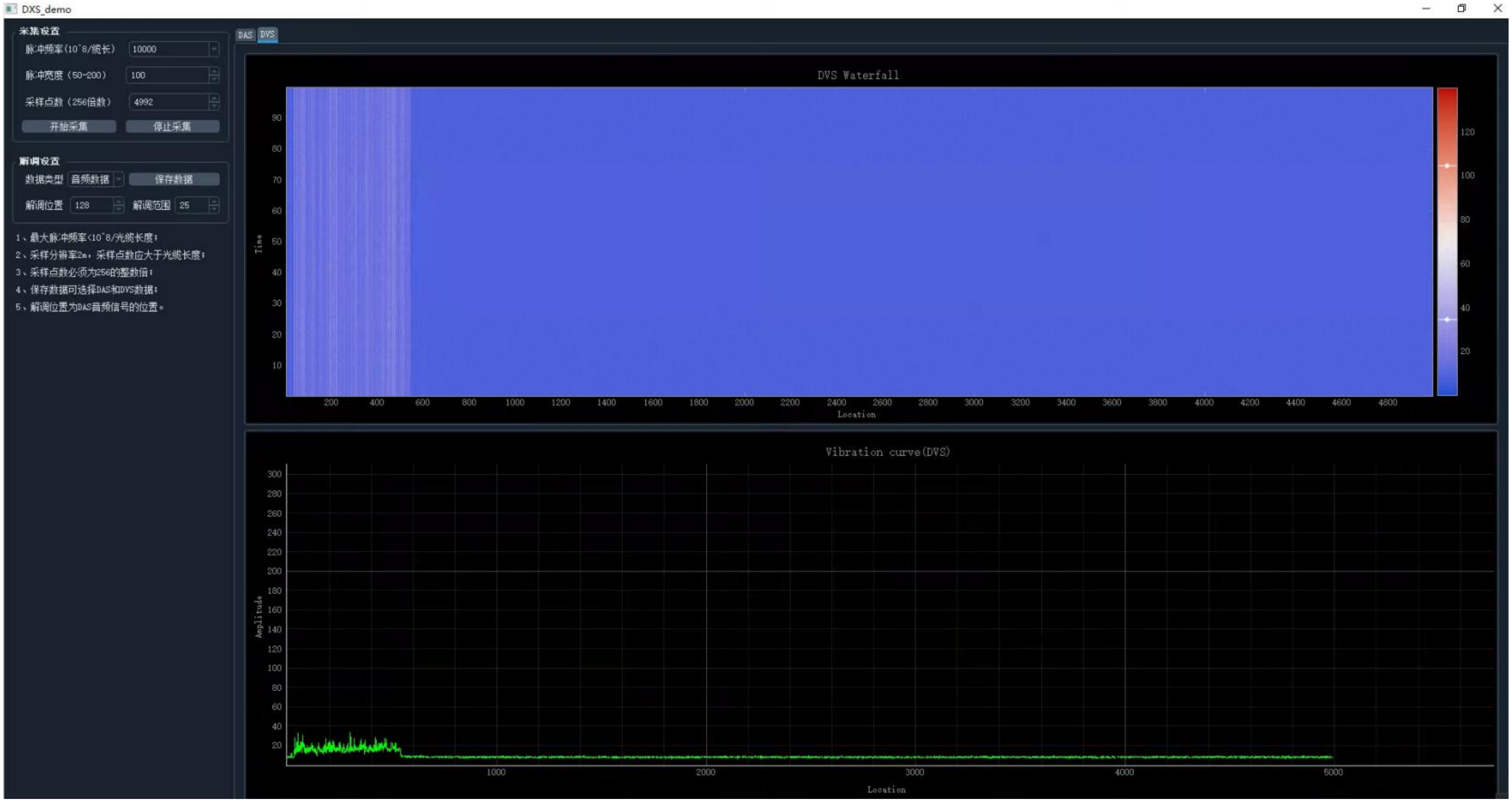
**解调设置**

数据类型 音频数据

解调位置 128　解调范围 25

1、最大脉冲频率<10ˆ8/光缆长度；
2、采样分辨率2m，采样点数应大于光缆长度；
3、采样点数必须为256的整数倍；
4、保存数据可选择DAS和DVS数据；
5、解调位置为DAS音频信号的位置。

# DEMO software



**采集设置**

脉冲频率(10ˆ8/缆长) 10000

脉冲宽度（50-200） 100

采样点数（256倍数） 4992

**解调设置**

数据类型 音频数据

解调位置 128　解调范围 25

1、最大脉冲频率<10ˆ8/光缆长度；
2、采样分辨率2m，采样点数应大于光缆长度；
3、采样点数必须为256的整数倍；
4、保存数据可选择DAS和DVS数据；
5、解调位置为DAS音频信号的位置。