RIGOL

Programming Guide

IQ Modulation

Oct. 2013
RIGOL Technologies, Inc.

Guaranty and Declaration

Copyright

Trademark Information

RIGOL is a registered trademark of RIGOL Technologies, Inc.

Publication Number

PGG03103-1110

Notices

- **RIGOL** products are protected by patent law in and outside of P.R.C.
- **RIGOL** reserves the right to modify or change parts of or all the specifications and pricing policies at company's sole decision.
- Information in this publication replaces all previously corresponding material.
- **RIGOL** shall not be liable for losses caused by either incidental or consequential in connection with the furnishing, use or performance of this manual as well as any information contained.
- Any part of this document is forbidden to be copied or photocopied or rearranged without prior written approval of **RIGOL**.

Product Certification

RIGOL guarantees this product conforms to the national and industrial standards in China as well as the ISO9001:2008 standard and the ISO14001:2004 standard. Other international standard conformance certification is in progress.

Contact Us

If you have any problem or requirement when using our products or this manual, please contact **RIGOL**.
E-mail: service@rigol.com
Websites: www.rigol.com

Document Overview

This manual introduces how to program and download waveform data into the RF signal generator on the basis of the IQ dynamic chained library.

Main contents in this manual:

1. **IQ Data Generation and Download**
   Introduces the two methods for generating and downloading waveform table data into DSG3000 on the basis of the IQ dynamic chained library.

2. **Programming Functions**
   Introduces the syntax, functions, parameters and using instruction of each function in the IQ dynamic chained library.

3. **Programming Examples**
   Introduces how to use Microsoft Visual C++ to program and download waveform data.

Content conventions in this manual:

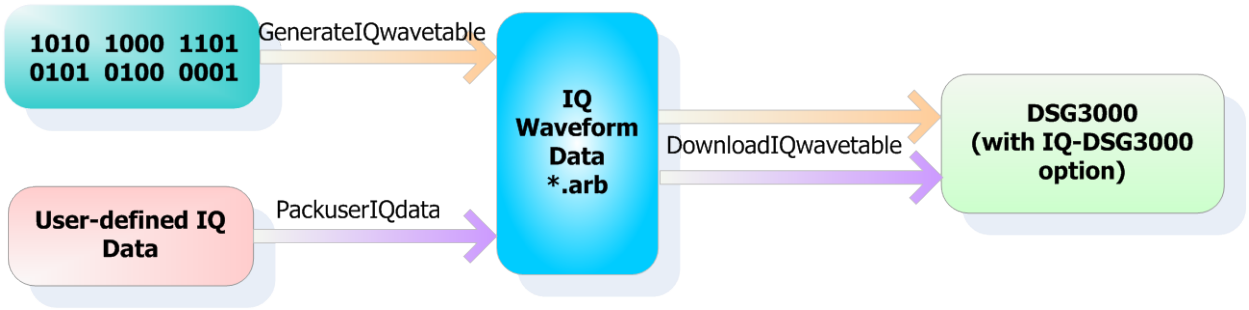
1. DSG3000 can communicate with the PC via USB, LAN or GPIB interface. Unless otherwise noted, the USB interface is used as an example in this manual to introduce how to program and download waveform data using the functions.

2. The DSG3000 mentioned in this manual refers to RF signal generator installed with the IQ-DSG3000 option.

Contents

IQ Data Generation and Download

There are two methods to generate waveform table data and download it into DSG3000 on the basis of the IQ dynamic chained library, as shown in the figure below.



Method 1: generate a new waveform table file by recalling the GenerateIQwavetable function. Then, download the waveform table file into DSG3000 by recalling the DownloadIQwavetable function.
Method 2: for user-defined IQ data stored in the PC, save the configuration information, pack it and generate a downloadable waveform table file by recalling the PackuserIQdata function. Then, download the waveform table file into DSG3000 by recalling the DownloadIQwavetable function.

Tip:

The **RIGOL** IQ dynamic chained library file (IQ.dll) includes the GenerateIQwavetable, DownloadIQwavetable and PackuserIQdata functions. To acquire this file, please download it from www.rigol.com.

Programming Functions

Function List:

Int32_t GenerateIQwavetable

Syntax int32_t GenerateIQwavetable(uint16_t UserPattern, int8_t *UserData, int32_t UserDlen, uint16_t PRBSType, int32_t SymLength, int32_t Symrate, int32_t CodMod, uint16_t Modtype, int32_t FskDev, uint16_t Filtype, double Filtercoeffcient, int32_t Implength, int32_t Oversap, char *Filepath)

Description Configure the data source, modulation type as well as the related parameters of the pulse-shaping filter, generate IQ waveform data and store it in the specified directory.

Parameter

Name	Type	Explanation
UserPattern	Unsigned 16-digit integer	Select the data source type: 0—user-defined; 1—PRBS code
UserData	8-digit integer array	User-defined input data
UserDlen	32-digit integer	The length of the data source when the UserPattern is 0
PRBSType	Unsigned 16-digit integer	Select the PRBS type: 0—PRBS7; 1—PRBS9; 2—PRBS11; 3—PRBS15; 4—PRBS16; 5—PRBS20; 6—PRBS21
SymLength	32-digit integer	The length of the sequence: ● When the UserPattern is 0, it is the length of the user-defined input data; ● When the UserPattern is 1, it is the length of the PRBS code. ≤ 2M symbols and fulfill the following relation with the number of over samples: SymLength * OverSap ≤ 4M
Symrate	32-digit integer	Code rate: 100syms/s ≤ Symrate ≤ 10M syms/s; Fulfill the following relation with the number of over samples: 1kHz ≤ Symrate * OverSap ≤ 50MHz
CodMod	32-digit integer	Code mode of the data source: 0—OFF; 1—Gray
Modtype	Unsigned 16-digit integer	Select the digital modulation type: 0—16QAM; 1—32QAM; 2—64QAM; 3—128QAM; 4—256QAM; 5—2ASK; 6—4ASK; 7—8ASK; 8—16ASK; 9—32ASK; 10—BPSK; 11—QPSK; 12—Pi/4QPSK; 13—Pi/4DQPSK; 14—8PSK; 15—MSK; 16—2FSK; 17—4FSK
FskDev	32-digit integer	FSK frequency offset: 1Hz ≤ FskDev ≤ 10MHz
Filtype	Unsigned 16-digit integer	The pulse-shaping filter type: 0—none; 1—Raised cosine; 2—Root raised cosine; 3—Gauss Filter
Filtercoeffcient	Double precision	The pulse-shaping filter coefficient: When the FilterType is 1 or 2, 0.05 ≤ Filtcoeff ≤ 1.00;

		When the FilterType is 3, 0.15 ≤ Filtcoeff ≤ 1000.00
Implength	32-digit integer	The filter length: the unit is symbol, 1 ≤ Implength ≤ 128
Oversap	32-digit integer	The number of over samples: 2 ≤ OverSap ≤ 32
Filepath	Character array	The storage directory of the waveform table file

Return Format

0—succeed;
0x0001—the symbol rate exceeds range;
0x0002—code mode error;
0x0003—Userpattern definition error;
0x0004—UserData definition error;
0x0005—PRBS type definition error;
0x0006—number of over samples definition error;
0x0007—modulation type definition error;
0x0008—filter type definition error;
0x0009—filter coefficient definition error;
0x0010—filter length definition error;
0x0011—file directory error;
0x0012—FSKDev definition error.

Int32_t DownloadIQwavetable

Syntax

int32_t DownloadIQwavetable(char *Addr, char * DstFlieName, char * SrcFileName, uint32_t Output)

Description

Download the IQ waveform table file into the instrument.

Parameter

Name	Type	Explanation
Addr	Character array	The VISA address of the instrument currently connected
DstFlieName	Character array	The name of the waveform table file downloaded into the instrument
SrcFileName	Character array	The name of the downloadable waveform table file (include the directory)
Output	Unsigned 32-digit integer	0—download the file into the instrument and output waveform 1—download the file to the instrument but do not output waveform

Return Format

0—succeed;
0x0011—source file directory error;
0x0013—destination filename error;
0x0014—VISA opening error.

Int32_t PackuserIQdata

Syntax int32_t PackuserdIQdata(char *SrcFileName, char *DstFlieName, uint32_t Datasource, int32_t SymLength, int32_t Symrate, uint32_t Modtype, uint32_t Filtype, double Filtcoeff, int32_t Implength, int32_t Oversap)

Description Save the IQ data configuration information, pack the data and generate downloadable waveform table file with the specified filename.

Parameter

Name	Type	Explanation
SrcFileName	Character Array	User-defined IQ data filename (include the directory)
DstFlieName	Character Array	The name of the waveform table file generated after packing the IQ data (include the directory)
Datasource	Unsigned 32-digit integer	Save the data source type information: 0—PRBS7; 1—PRBS9; 2—PRBS11; 3—PRBS15; 4—PRBS16; 5—PRBS20; 6—PRBS21; 7—UserData
SymLength	32-digit integer	Save the sequence length: the unit is symbol
Symrate	32-digit integer	Save the code rate: the unit is symbol/s
Modtype	Unsigned 32-digit integer	Save the digital modulation type information, the types available include: 0—16QAM; 1—32QAM; 2—64QAM; 3—128QAM; 4—256QAM; 5—2ASK; 6—4ASK; 7—8ASK; 8—16ASK; 9—32ASK; 10—BPSK; 11—QPSK; 12—Pi/4QPSK; 13—Pi/4DQPSK; 14—8PSK; 15—MSK; 16—2FSK; 17—4FSK
Filtype	Unsigned 32-digit integer	Save the pulse-shaping filter type: 0—none; 1—Raised cosine; 2—Root raised cosine; 3—Gauss Filter
Filtcoeff	Double precision	Save the filter coefficient
Implength	32-digit integer	Save the filter length, the unit is symbol
OverSap	32-digit integer	Save the number of over samples

Return Format 0—succeed;
0x0011—source file directory error;
0x0013—destination filename error.

Programming Examples

This section provides the example of generating waveform table data and downloading it into DSG3000 by programming on the basis of the IQ dynamic chained library in Microsoft Visual C++ 2010 environment.

Programming Preparations

You need to make the following preparations before programming.

1. Install Ultra Sigma. Please download the software from www.rigol.com and then follow the instructions to install the software.

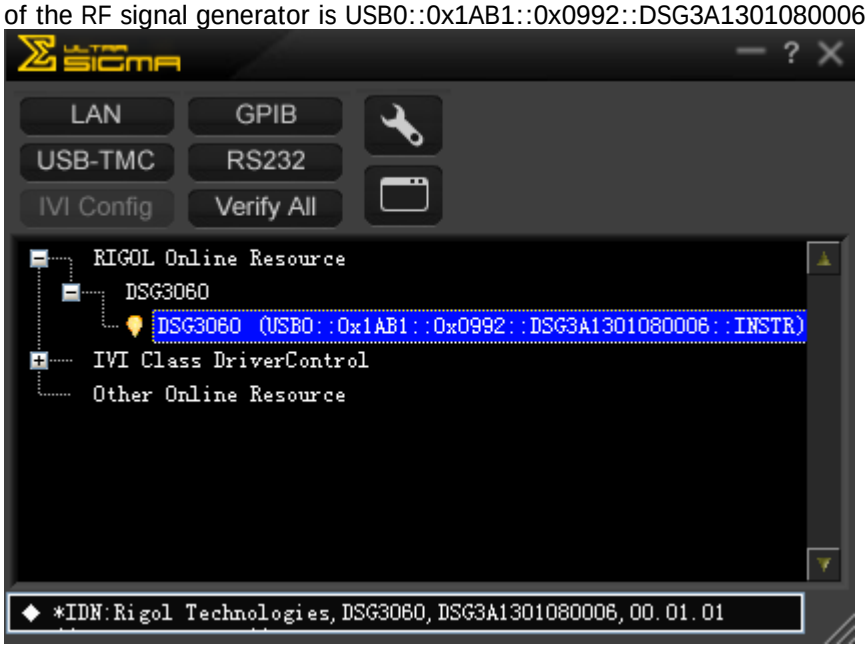
2. Download the Ultra IQ Station installation package from www.rigol.com and install it. At this point, the default installation directory of the IQ dynamic chained library is C:\Program Files\RIGOL Technologies, Inc\Ultra Sigma\Instrument Tools\RIGOL_DSG_Tools_Ultra IQ Station\IQ DLL.

3. Here, the USB interface of the RF signal generator is used to communicate with the PC. Please connect the USB DEVICE interface at the rear panel of the RF signal generator with the PC using USB cable.

4. After connecting the RF signal generator and PC, power on and start the RF signal generator.

5. At this point, the "**Found New Hardware Wizard**" dialog box is displayed on the PC and please follow the instructions to install the "USB Test and Measurement Device".

6. Acquire the USB VISA descriptor of the RF signal generator: run Ultra Sigma and search for the RF signal generator resource currently connected to the PC. The resource found will be displayed under the "RIGOL Online Resource" directory, including the model and USB interface information (namely the VISA descriptor) of the instrument, as shown in the figure below. In this example, the VISA descriptor of the RF signal generator is USB0::0x1AB1::0x0992::DSG3A1301080006::INSTR.



By now, the programming preparations are finished.

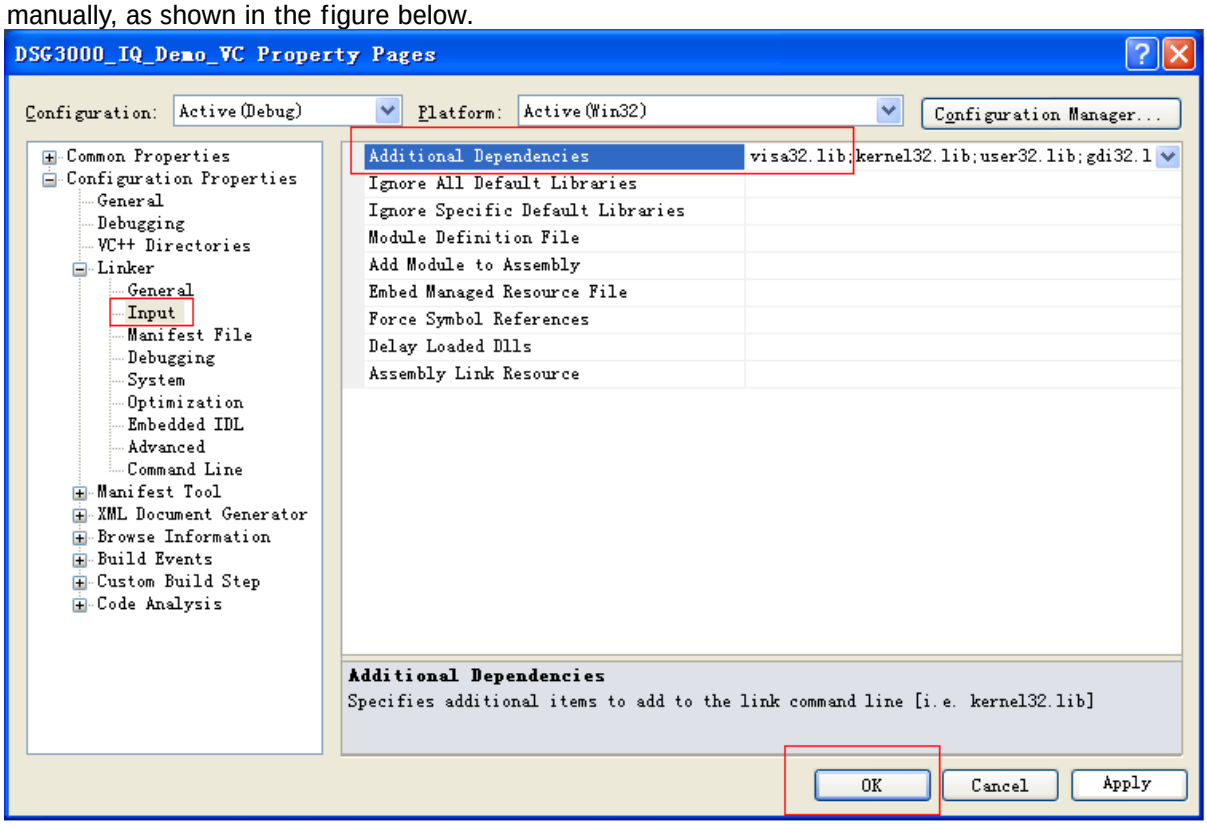
Programming Procedures

Program used in this example: Microsoft Visual C++ 2010

Functions realized in this example: generate downloadable IQ waveform table using the function interface provided by the dynamic library; encapsulate the present IQ data and generate downloadable IQ waveform table using the function interface provided by the dynamic library; search for instrument address and download the present IQ waveform table into DSG3000 using the function interface provided by the dynamic library.

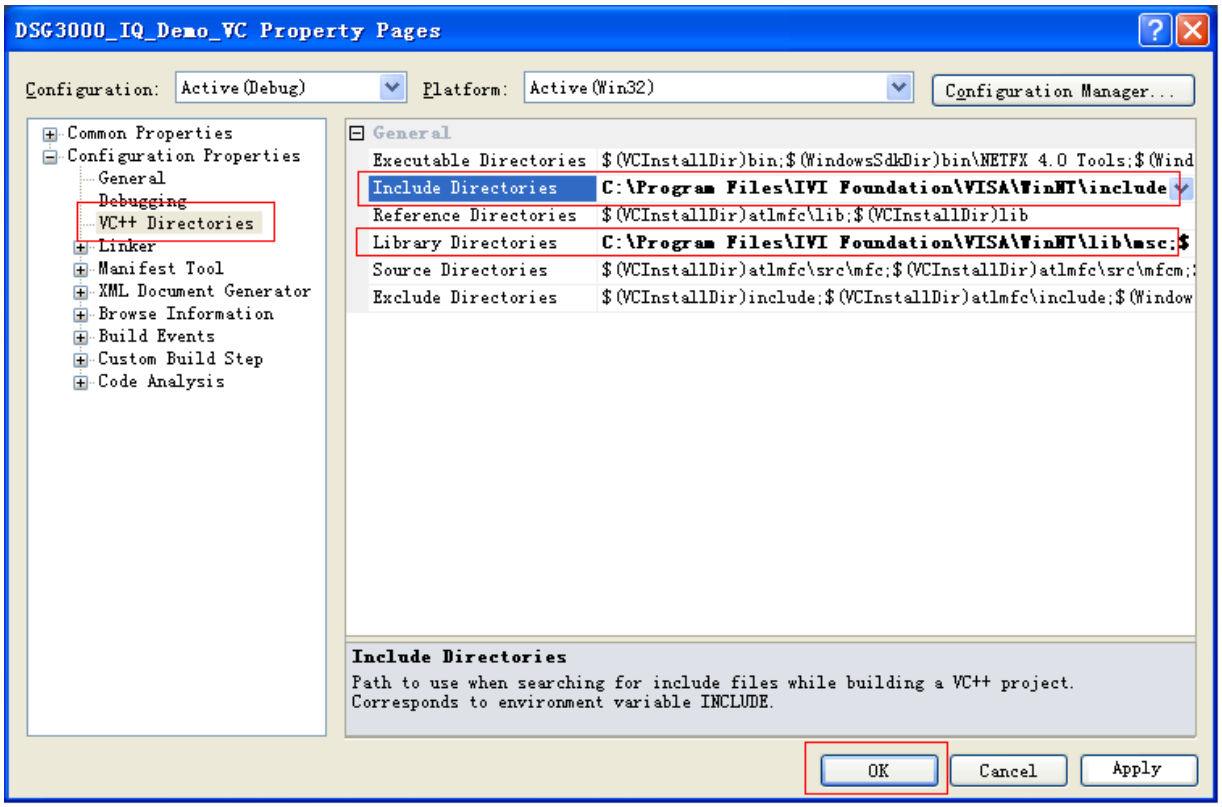
1. Run Microsoft Visual C++ 2010, create a program based on console and name it as DSG3000_IQ_Demo_VC.

2. Click **Project→ Properties** and add **visa32.lib** at the specified position in the pop-up interface manually, as shown in the figure below.



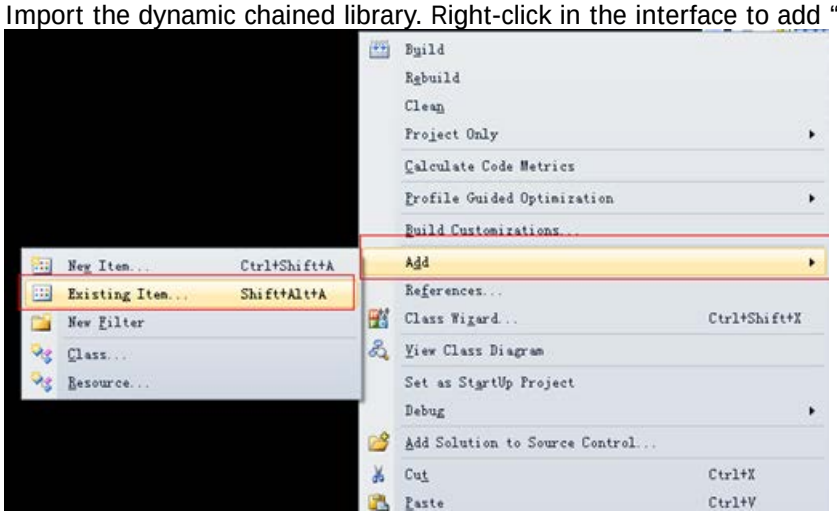
3. Click **Project→Properties** and add the **Include** and **Lib** directories in the **VC++ Directories** list in the pop-up interface.

   Note:
   The two directories added here are related to the installation directory of NI-VISA on your PC. Here,the default installation directory of NI-VISA is C:\Program Files\IVI Foundation\VISA.

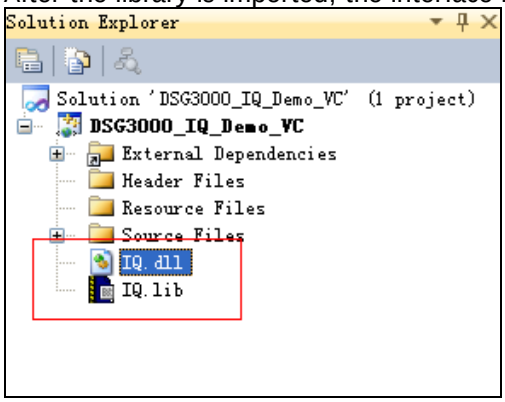


4. Place the **RIGOL** IQ dynamic library and the related header files under the program directory. Note that the header files required when programming include IQ.h, visa.h and visatype.h.

5. Import the dynamic chained library. Right-click in the interface to add “IQ.dll” and “IQ.lib”.



After the library is imported, the interface is as shown in the figure below.



6. Function interface 1: configure parameters and generate downloadable IQ waveform table.

```
void test_GenerateIQwavetable(void)
{
    //Define the value of each parameter according to the range in the manual
    int s32UserPattern = 0;
    char as8UserData[200] = {
    0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0,
    0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0,
    0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0,
    0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0,
    0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0,
    0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0,
    0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0,
    0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0,
    0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0,
    0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0
    };
    int s32UserLen = 200;

    int u32PRBSType = 1;
    int s32SymLength = 32;
    int s32Symrate = 500000;
    int s32CodMod = 1;
    int s32Modtype = 0;
    int s32FskDev = 1000000;
    int s32Filtype = 0;
    double f32Filtcoeff = 0.22;
    int s32Implength = 32;
    int s32Oversap = 4;
    char *ps8FilePath = "D:\\IQ_ARB\\TestIQ.arb";

    //Recall the library function interface to generate waveform table and return 0 when the operation
    succeeds
    int s32Res = 0;
    s32Res = GenerateIQwavetable(s32UserPattern, as8UserData, s32UserLen,
        u32PRBSType, s32SymLength, s32Symrate, s32CodMod, s32Modtype, s32FskDev, s32Filtype,
        f32Filtcoeff, s32Implength, s32Oversap, ps8FilePath);
}
```

7. Function interface 2: add parameters and generate downloadable IQ waveform table according to the present IQ data.
Prepare file: "D:\\IQ_ARB\\InPut.arb" (IQ data file)

```
void test_PackuserIQdata(void)
{
    // Define the value of each parameter according to the range in the manual
    char *ps8SrcFileName = "D:\\IQ_ARB\\InPut.arb";
    char *ps8DstFileName = "D:\\IQ_ARB\\OutPut.arb";

    unsigned int u32DataSour = 7;
    int s32SymLength = 32;
    int s32Symrate = 500000;
    unsigned int u32Modtype = 3;
    unsigned int   u32Filtype = 2;
    double f32Filtcoeff = 0.22;
    int s32Implength = 32;
    int s32Oversap = 4;

    //Recall the library function interface to acquire the corresponding output file and return 0 when
```

```
        the operation succeeds
        s32 s32Res = 0;
        s32Res = PackuserIQdata(ps8SrcFileName, ps8DstFileName, u32DataSour, s32SymLength,
        s32Symrate, u32Modtype, u32Filtype, f32Filtcoeff, s32Implength,   s32Oversap);
    }
```

8. Function interface 3: download the present waveform table in the PC into the device.

```
    unsigned short u16VisaNum = 0;
    char as8ViUnitStr[30][100] = {0};
    ViSession VdefaultRM;
    short visaInitFun(void)
    {
        ViStatus VStatus;

        VStatus = viOpenDefaultRM(&VdefaultRM);
        if (VStatus < VI_SUCCESS)
        {
            printf("vi open error!");
            return -1;
        }

        // Search for device
        ViFindList VFindList;
        ViUInt32 VUintNum;
        ViChar VUintStr[200];
        VStatus = viFindRsrc(VdefaultRM, "?*INSTR", &VFindList, &VUintNum, VUintStr);
        if (VStatus < VI_SUCCESS)
        {
            viClose(VdefaultRM);
            printf("no Find Vi Uint!");
            return -1;
        }

        // Search for the actual device
        u16VisaNum = VUintNum;
        ViUInt32 VNumTemp = 0;
        for (VNumTemp = 0; VNumTemp < VUintNum; VNumTemp++)
        {
            strcpy(as8ViUnitStr[VNumTemp], VUintStr);
            viFindNext(VFindList, VUintStr);
        }

        return 0;
    }

    // Download the waveform table into the device
    // The address of the destination device: as8ViUnitStr[0] :
    "USB0::0x1AB1::0x0992::DSG3A1301080006::INSTR"
    void test_DownloadIQwavetable(void)
    {
        char *ps8DstFileName = "DownLoad.arb";
        char *ps8SrcFileName = "D:\\IQ_ARB\\TestIQ.arb";

        int s32Res = -1;
        s32Res = DownloadIQwavetable(as8ViUnitStr[0], ps8DstFileName, ps8SrcFileName, 1);
    }
```

9. Running results
   1) The output file acquired after function interface 1 runs successfully: “D:\\IQ_ARB\\TestIQ.arb”;
   2) The output file acquired after function interface 2 runs successfully: “D:\\IQ_ARB\\OutPut.arb”;
   3) The file acquired in the local D: disk of DSG3000 after function interface 3 runs successfully: “DownLoad.arb”.