

RIGOL

编程手册

IQ 调制

2013 年 10 月
RIGOL Technologies, Inc.

保证和声明

版权

© 2013 北京普源精电科技有限公司版权所有。

商标信息

RIGOL 是北京普源精电科技有限公司的注册商标。

文档编号

PGG03003-1110

声明

- 本公司产品受已获准及尚在审批的中华人民共和国专利的保护。
- 本公司保留改变规格及价格的权利。
- 本手册提供的信息取代以往出版的所有资料。
- 对于本手册可能包含的错误，或因手册所提供的信息及演绎的功能，以及因使用本手册而导致的任何偶然或继发的损失，**RIGOL** 概不负责。
- 未经 **RIGOL** 事先书面许可不得影印复制或改编本手册的任何部分。

产品认证

RIGOL 认证本产品符合中国国家产品标准和行业产品标准及 ISO9001:2008 标准和 ISO14001:2004 标准，并进一步认证本产品符合其它国际标准组织成员的相关标准。

联系我们

如您在使用此产品或本手册的过程中有任何问题或需求，可与 **RIGOL** 联系：

电子邮箱：service@rigol.com

网址：www.rigol.com

文档概述

本手册详细介绍基于 IQ 动态链接库编程并下载波形数据至射频信号源的操作方法。

文档的主要内容:

1. IQ 数据生成和下载

本节概述了基于 IQ 动态链接库生成波表数据并下载至 DSG3000 的两种方法。

2. 编程函数

本节逐条介绍 IQ 动态链接库中各函数的格式、功能、参数以及使用说明等。

3. 编程实例

本节给出如何使用 Microsoft Visual C++ 开发工具编程并加载波形数据。

文档的内容约定:

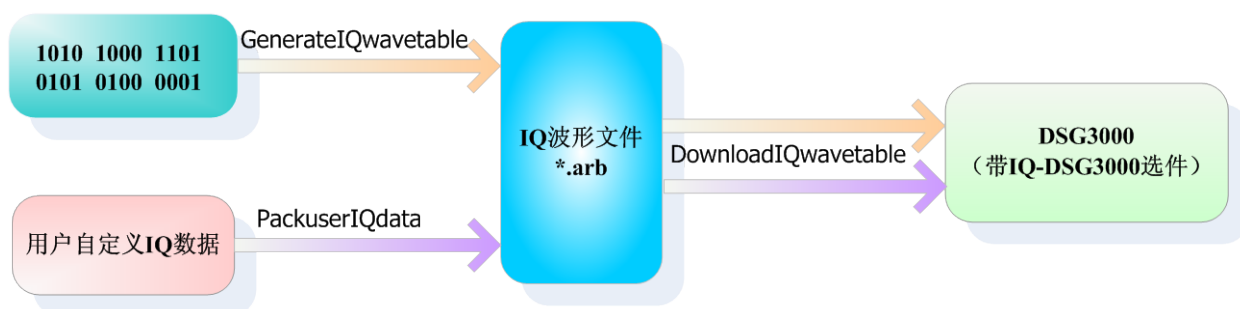
1. DSG3000 与计算机之间可以通过 USB、LAN 或 GPIB 接口建立远程通信。如无特殊说明，本手册以 USB 接口通信为例介绍通过调用函数编程和下载波形数据。
2. 本手册中的 DSG3000 均指带 IQ-DSG3000 选件的射频信号源。

目录

保证和声明.....	I
文档概述.....	II
IQ 数据生成和下载	1
编程函数	2
Int32_t GenerateIQwavetable.....	2
Int32_t DownloadIQwavetable	3
Int32_t PackuserIQdata	4
编程实例	5
编程准备	5
编程步骤.....	6

IQ 数据生成和下载

基于 IQ 动态链接库生成波表数据并下载至 DSG3000 的方法有两种，如下图所示。



方法一：通过调用 GenerateIQwavetable 函数生成一个新的波表文件。然后，调用 DownloadIQwavetable 函数将该波表文件下载至 DSG3000。

方法二：对于 PC 已存储的用户自定义 IQ 数据，则通过调用 PackuserIQdata 函数保存配置信息并将其打包生成可加载的波表文件，之后调用 DownloadIQwavetable 函数下载该波表文件至 DSG3000。

提示：

由 RIGOL 提供的 IQ 动态链接库文件（IQ.dll）包含 GenerateIQwavetable、DownloadIQwavetable 和 PackuserIQdata 三个函数。如需获取该文件，请登陆网站www.rigol.com下载。

编程函数

函数列表:

- ◆ [Int32_t GenerateIQwavetable](#)
- ◆ [Int32_t DownloadIQwavetable](#)
- ◆ [Int32_t PackuserIQdata](#)

Int32_t GenerateIQwavetable

函数格式 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 Filtercoefficient, int32_t Implength, int32_t Oversap, char *Filepath)

功能描述 配置数据源、调制类型和脉冲成型滤波器的相关参数生成 IQ 波形数据，并将其存储在指定路径中。

参数

名称	类型	说明
UserPattern	无符号 16 位整型	选择数据源类型： 0—用户自定义方式；1—PRBS 码
UserData	8 位整型数组	用户自定义输入数据
UserDlen	32 位整型	UserPattern 选择方式 0 时数据源的长度
PRBSType	无符号 16 位整型	选择 PRBS 类型： 0—PRBS7；1—PRBS9；2—PRBS11； 3—PRBS15；4—PRBS16；5—PRBS20； 6—PRBS21
SymLength	32 位整型	序列长度： ● UserPattern 选择 0 时，即用户自定义输入数据长度； ● UserPattern 选择 1 时，即 PRBS 码长度。 ≤ 2M symbols，并且与过采样个数满足如下关系： $SymLength * OverSap \leq 4M$
Symrate	32 位整型	码率： $100syms/s \leq Symrate \leq 10M syms/s$ ， 与过采样个数满足如下关系： $1kHz \leq Symrate * OverSap \leq 50MHz$
CodMod	32 位整型	数据源编码方式：0—OFF；1—Gray
Modtype	无符号 16 位整型	选择数字调制类型： 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 位整型	FSK 频率偏移： $1Hz \leq FskDev \leq 10MHz$
Filtype	无符号 16 位整型	脉冲成型滤波器类型 0—none；1—Raised cosine； 2—Root raised cosine；3—Gauss Filter
Filtercoefficient	双精度型	脉冲成型滤波器系数 FilterType 为 1 或 2 时， $0.05 \leq Filtcoeff \leq 1.00$ ； FilterType 为 3 时， $0.15 \leq Filtcoeff \leq 1000.00$
Implength	32 位整型	滤波器长度：单位为 symbol， $1 \leq Implength \leq 128$
Oversap	32 位整型	过采样个数： $2 \leq OverSap \leq 32$
Filepath	字符型数组	波表文件保存路径

返回格式 返回 0—成功；
 0x0001—码率超出范围；
 0x0002—编码方式错误；
 0x0003—Userpattern 定义错误；
 0x0004—UserData 定义错误；
 0x0005—PRBS 类型定义错误；
 0x0006—过采样个数定义错误；
 0x0007—调制方式定义错误；
 0x0008—滤波器类型定义错误；
 0x0009—滤波器系数定义错误；
 0x0010—滤波器长度定义错误；
 0x0011—文件路径错误；
 0x0012—FSKDev 定义错误。

Int32_t DownloadIQwavetable

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

功能描述 将 IQ 波表文件下载到仪器。

参数

名称	类型	说明
Addr	字符型数组	当前连接仪器的 VISA 地址
DstFlieName	字符型数组	波表文件下载到仪器中的文件名
SrcFileName	字符型数组	可加载波表文件名（包含路径）
Output	无符号 32 位整型	0—下载到仪器并输出波形 1—下载到仪器中不输出波形

返回格式 返回 0—成功；
 0x0011—源文件路径错误；
 0x0013—目标文件名错误；
 0x0014—VISA 打开错误。

Int32_t PackuserIQdata

函数格式 int32_t PackuserIQdata(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)

功能描述 保存 IQ 数据配置信息，将其打包并以指定文件名生成可加载的波表文件。

参数

名称	类型	说明
SrcFileName	字符型数组	用户自定义的 IQ 数据文件名（包含路径）
DstFlieName	字符型数组	打包 IQ 数据后生成的波表文件名（包含路径）
Datasource	无符号 32 位整型	保存数据源类型信息： 0—PRBS7； 1—PRBS9； 2—PRBS11； 3—PRBS15； 4—PRBS16； 5—PRBS20； 6—PRBS21； 7—UserData
SymLength	32 位整型	保存序列长度：单位为 symbol
Symrate	32 位整型	保存码率：单位为 symbol/s
Modtype	无符号 32 位整型	保存选择数字调制类型信息，包括以下类型： 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	无符号 32 位整型	保存脉冲成型滤波器类型 0—none； 1—Raised cosine； 2—Root raised cosine； 3—Gauss Filter
Filtcoeff	双精度型	保存滤波器系数
Implength	32 位整型	保存滤波器长度，单位为 symbol
OverSap	32 位整型	保存过采样个数

返回格式 返回 0—成功；
0x0011—源文件路径错误；
0x0013—目标文件名错误。

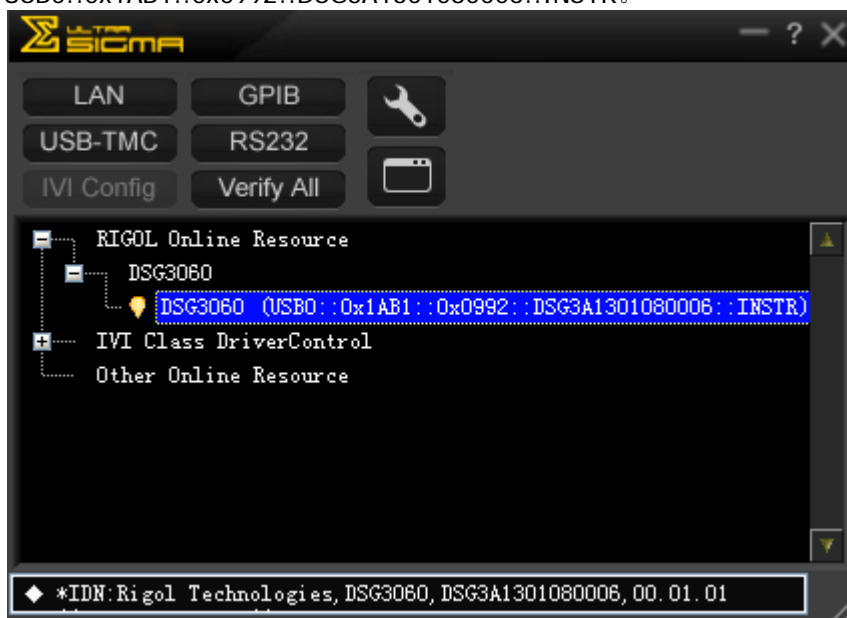
编程实例

本节给出在 Microsoft Visual C++ 2010 环境下，基于 IQ 动态链接库编程生成波表数据并下载至 DSG3000 的实例。

编程准备

编程之前，您需要做如下准备工作：

1. 安装 Ultra Sigma 通用 PC 软件。请登陆 www.rigol.com 下载该软件，然后按照指导进行安装。
2. 请到 www.rigol.com 下载 Ultra IQ Station 安装包并将其成功安装。此时，IQ 动态库的默认安装路径为 C:\Program Files\RIGOL Technologies, Inc\Ultra Sigma\Instrument Tools\RIGOL_DSG_Tools_Ultra IQ Station\IQ DLL。
3. 本文应用射频信号源的 USB 接口与计算机通信。请使用 USB 数据线将射频信号源后面板的 USB DEVICE 接口与计算机相连。
4. 射频信号源与计算机正确连接后，请给射频信号源上电并开机。
5. 此时，计算机上将弹出“硬件更新向导”对话框，请按照向导的提示安装“USB Test and Measurement Device”。
6. 获取射频信号源的 USB VISA 描述符：运行 Ultra Sigma，搜索当前连接到 PC 上的射频信号源资源。搜索到的资源显示在“RIGOL Online Resource”目录下，包含仪器的型号和 USB 接口信息（即 VISA 描述符），如下图所示。本实例使用的射频信号源的 VISA 描述符为 USB0::0x1AB1::0x0992::DSG3A1301080006::INSTR。



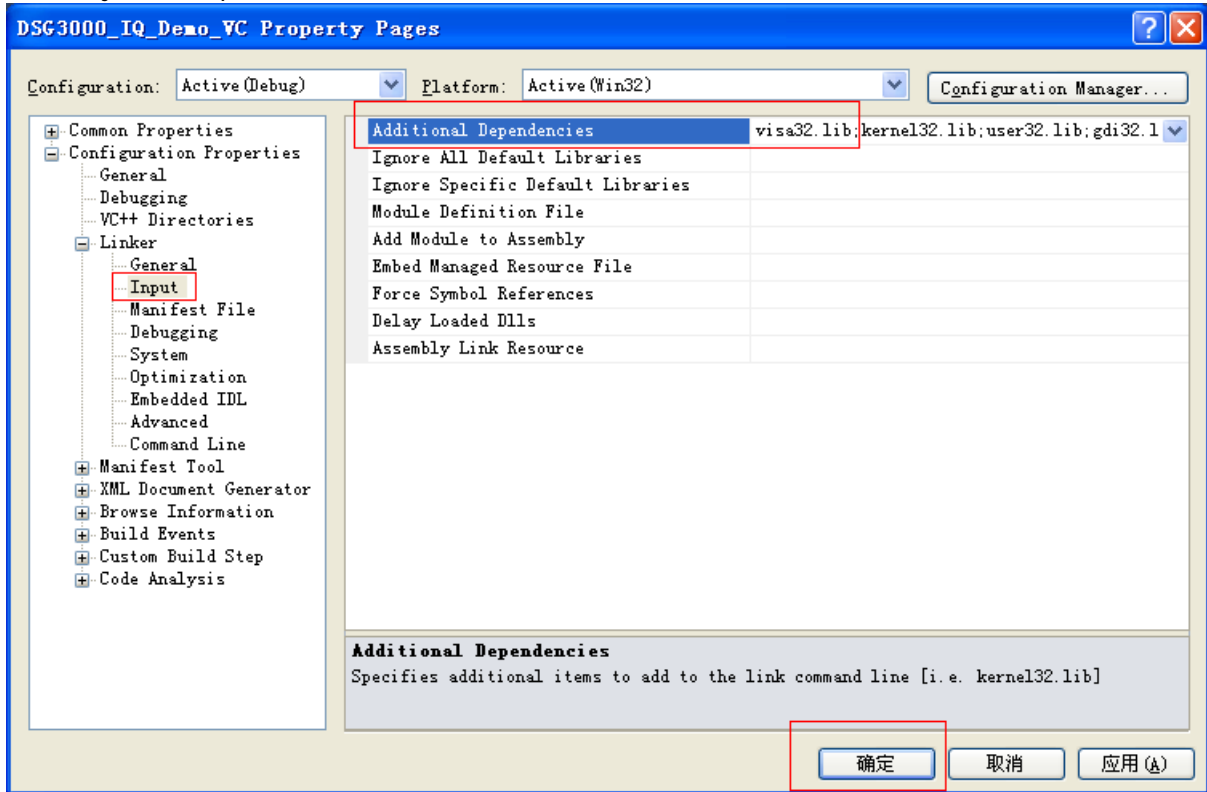
至此，编程准备工作结束。

编程步骤

本例使用的程序：Microsoft Visual C++ 2010

本例实现的功能：使用动态库中提供的函数接口生成可下载的 IQ 波表；使用动态库中提供的函数接口封装已存在的 IQ 数据并生成可下载的 IQ 波表；查找仪器地址，使用动态库中提供的函数接口下载已存在的 IQ 波表到 DSG3000 中。

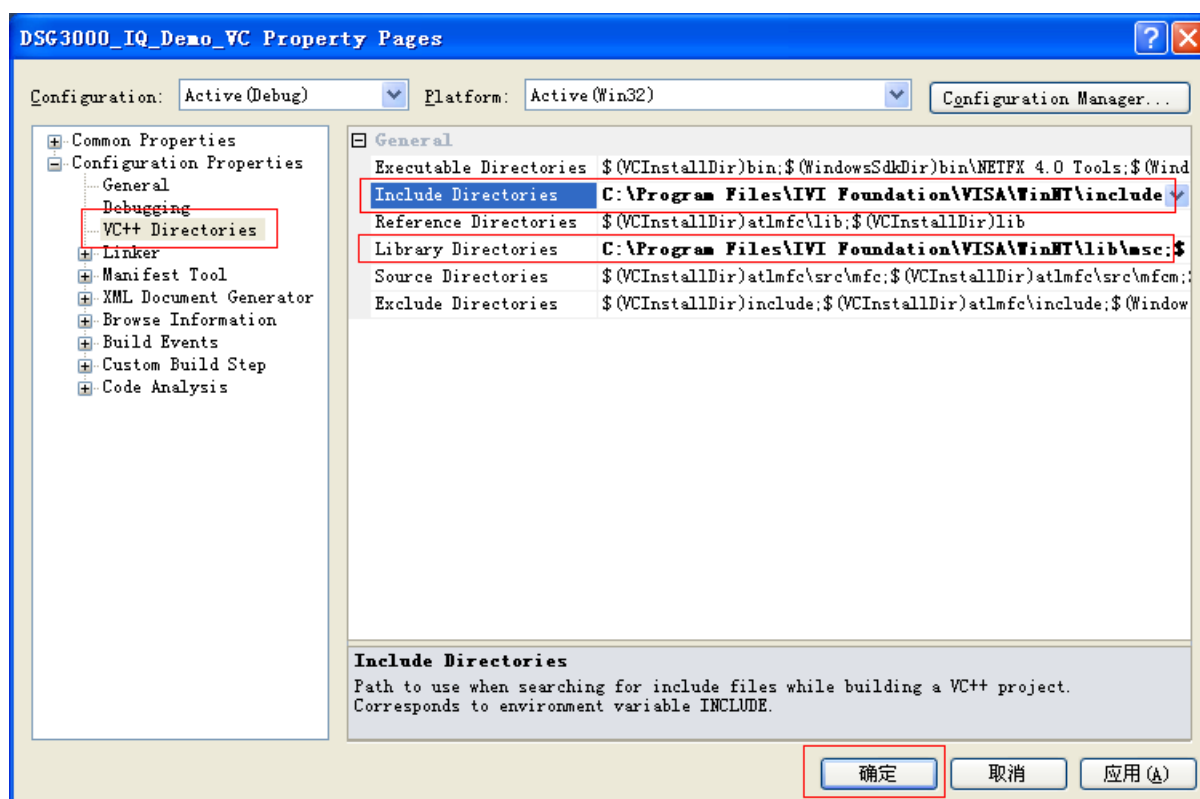
1. 运行 Microsoft Visual C++ 2010，新建一个基于控制台的程序，命名为 DSG3000_IQ_Demo_VC。
2. 点击 Project→ Properties，在弹出界面的指定位置手动添加 visa32.lib，如下图所示。



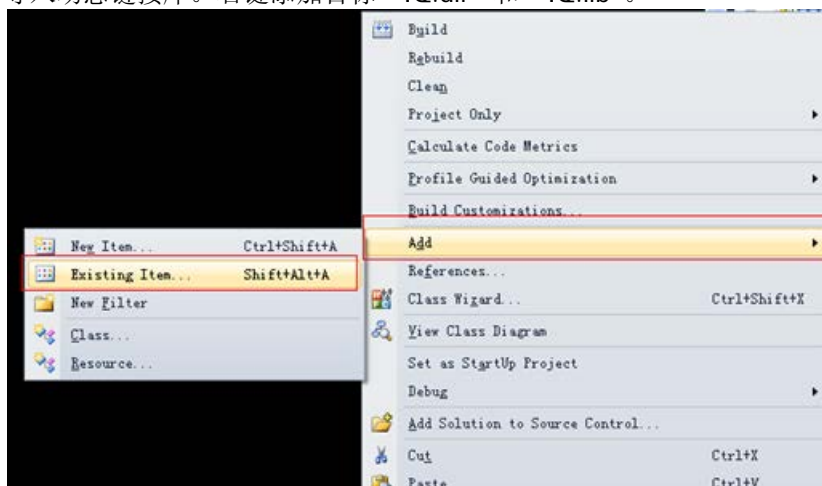
3. 点击 Project→Properties，在弹出界面的 VC++ Directories 列表中添加 Include 和 Lib 路径。

注意：

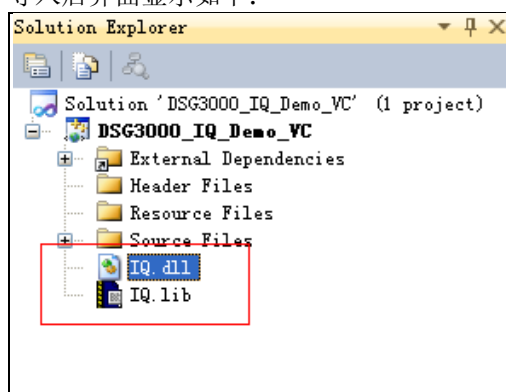
此处添加的两个路径与您计算机上 NI-VISA 的安装路径相关。此处默认为 NI-VISA 安装在 C:\Program Files\IVI Foundation\VISA 路径下。



4. 将 **RIGOL** 提供的 IQ 动态库及相关头文件放到工程所在的目录下。注意：编写程序时需要包含的头文件包括：IQ.h、visa.h 和 visatype.h。
5. 导入动态链接库。右键添加目标 “IQ.dll” 和 “IQ.lib”。



导入后界面显示如下：



6. 函数接口 1: 配置参数, 生成可下载的 IQ 波表。

```
void test_GenerateIQwavetable(void)
{
    //根据手册给定的范围定义各个参数的值
    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,
        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 s32Filttype = 0;
    double f32Filtcoeff = 0.22;
    int s32Implength = 32;
    int s32Oversap = 4;
    char *ps8FilePath = "D:\\IQ_ARB\\TestIQ.arb";

    //调用库函数接口, 生成波表, 执行成功时返回 0
    int s32Res = 0;
    s32Res = GenerateIQwavetable(s32UserPattern, as8UserData, s32UserLen,
        u32PRBSType, s32SymLength, s32Symrate, s32CodMod, s32Modtype, s32FskDev, s32Filttype,
        f32Filtcoeff, s32Implength, s32Oversap, ps8FilePath);
}
```

7. 函数接口 2: 添加参数, 根据已经存在的 IQ 数据生成可下载的 IQ 波表。

准备文件: "D:\\IQ_ARB\\InPut.arb" (IQ 数据文件)

```
void test_PackuserIQdata(void)
{
    //根据文档给定范围设置参数的值
    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 u32Filttype = 2;
    double f32Filtcoeff = 0.22;
    int s32Implength = 32;
    int s32Oversap = 4;

    //调用库函数接口得到相应的输出文件, 执行成功时返回 0
    s32 s32Res = 0;
```

```

    s32Res = PackuserIQdata(ps8SrcFileName, ps8DstFileName, u32DataSour, s32SymLength,
    s32Symrate, u32Modtype, u32Filttype, f32Filtcoeff, s32Implength, s32Oversap);
}

```

8. 函数接口 3: 下载 PC 中已存在的波表到设备中。

```

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;
    }

    // 搜索设备
    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;
    }

    // 搜索具体的仪器
    u16VisaNum = VUIntNum;
    ViUInt32 VNumTemp = 0;
    for (VNumTemp = 0; VNumTemp < VUIntNum; VNumTemp++)
    {
        strcpy(as8ViUnitStr[VNumTemp], VUIntStr);
        viFindNext(VFindList, VUIntStr);
    }

    return 0;
}

// 下载波表到设备中
// 目标设备地址: 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. 运行结果:

- 1) 函数接口 1 运行成功后得到输出文件: “D:\\IQ_ARB\\TestIQ.arb”;
- 2) 函数接口 2 运行成功后得到输出文件: “D:\\IQ_ARB\\OutPut.arb”;
- 3) 函数接口 3 运行成功后在 DSG3000 本地目录 D:盘下得到文件: “DownLoad.arb”。