

Acute Data Generator Software Development Kit (SDK) Programming Guide

For Data Generator 3000 and TravelData 3000

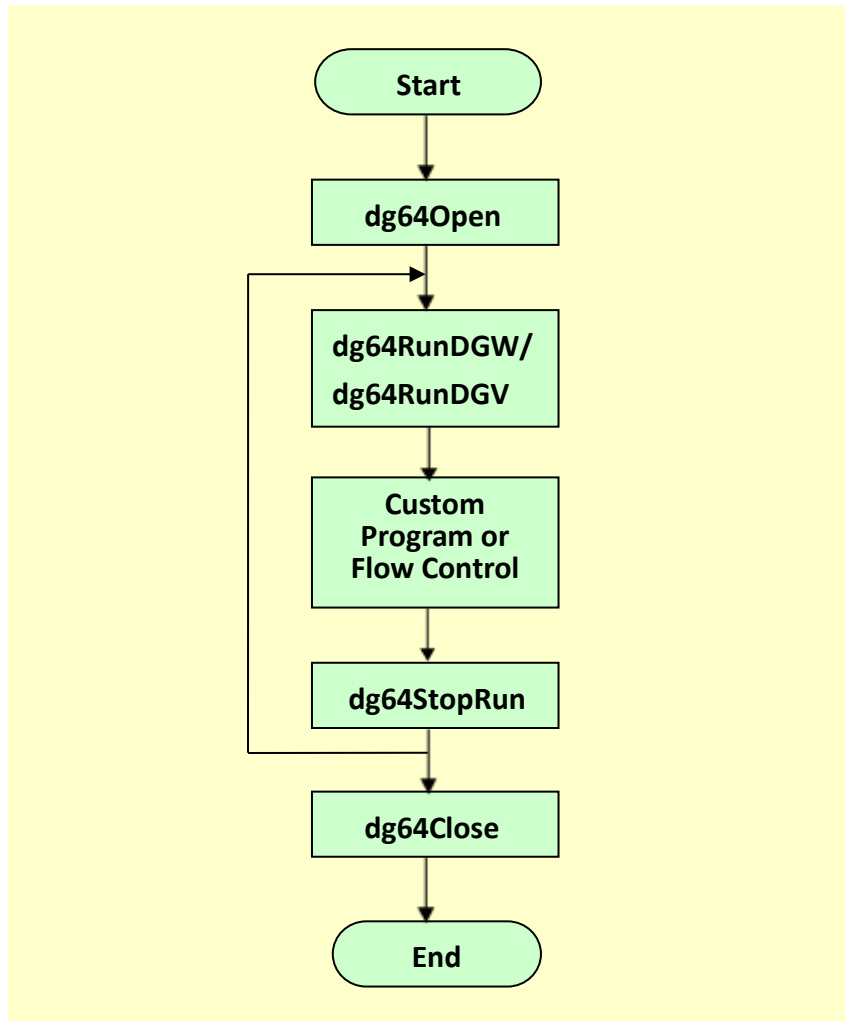
Version: 1.0

Publish: 2019/12/18

Contents

SDK Control Flow and simple introduction.....	3
HWINFO Structure Introduction.....	4
SDK Function Definitions.....	4
bool dg64Open().....	4
bool dg64HwInfo(HWINFO *hi).....	4
bool dg64Close().....	5
int dg64RunDGW(char* szFileName).....	5
int dg64RunDGV(char* szFileName).....	5
bool dg64Status(DWORD *pdwStatus).....	6
bool dg64LastError(int iBufSize, char* szBuf).....	6

SDK Control Flow and simple introduction



This SDK provides an open interface for users to generate the pattern by loading Data Generator waveform file. The filename extension is DGW/TDW/DGV.

HWINFO Structure Introduction

```
typedef struct _HWINFO
{
    DWORD    dwModel; // 0x33064 = DG3064B, 0x33096 = DG3096B, 0x33128 = DG3128B
                // 0x23008 = TD3008E, 0x23116 = TD3116B, 0x23216 = TD3216B
    char      szSerialNo[32]; // Serial number
    int       iUsbSpeed; // 0 = USB2.0 I/F, 1 = USB3.0 I/F
    int       iPods; // Available POD number
    int       iChannels; // Available channel number
    int       iMemDepth; // Maximum memory depth for each channel
    int       iMaxFreq; // Maximum frequency
    int       iMinFreq; // Maximum frequency
}HWINFO;
```

SDK Function Definitions

bool dg64Open()

Connect the data generator device with computer.

Return value

Return true if the function succeeded; false if the function failed.

bool dg64HwInfo(HWINFO *hi)

Get the data generator hardware information, this information will be filled in the HWINFO struct.

Return value

Return true if the function succeeded; false if the function failed.

Remarks

HWINFO cHwInfo = {0};

dgHwInfo(&cHwInfo); // cHwInfo.dwModel can get the data generator model name
// cHwInfo.iUsbSpeed can get the USB I/F

bool dg64Close()

Disconnect the data generator device with computer.

Return value

Return true if the function succeeded; false if the function failed.

int dg64RunDGW(char* szFileName)

int dg64RunDGV(char* szFileName)

Load the DGW/TDW/DGV File and run.

Parameters

szFileName[in]:

Type: **char***

The file path of data generator waveform file.

Return value

Return 0 if the function succeeded; not 0 (error code) if the function failed.

// Error code about the return of dg64RunDGW/dg64RunDGV function

```
#define ERR_DG_FILE_NOT_FOUND      0x0001
#define ERR_DG_FILE_OPEN           0x0002
#define ERR_DG_FILE_TYPE           0x0003
#define ERR_DG_FILE_FORMAT         0x0004
#define ERR_DG_FILE_VERSION        0x0005
#define ERR_DG_MEMORY_NOT_ENOUGH   0x0006
#define ERR_DG_SIGNAL_INPUT_NAME   0x0007
#define ERR_DG_CHANNEL_NUMBER      0x0008
#define ERR_DG_TIME_STAMP           0x0009
#define ERR_DG_FREQ_COLLISION      0x000A
#define ERR_DG_NO_INPUT_NAME       0x000B
#define ERR_DG_DATA_LENGTH         0x000C
#define ERR_DG_NO_TIME_MARK        0x000D
#define ERR_DG_NO_CHANNEL_NUMBER   0x000E
#define ERR_DG_MEMORY              0x000F
#define ERR_DG_FILE_READ           0x0010
#define ERR_DG_NO_HARDWARE         0x0063
#define ERR_DG_FREQ_NOT_MATCH      0x8001
#define WARN_DG_DEV_MEMORY_LIMIT   0x8100
#define WARN_DG_PC_MEMORY_LIMIT    0x8101
```

bool dg64Status(DWORD *pdwStatus)

Get the current status of data generator.

Parameters

pdwStatus[in]:

Type: **DWORD***

Status:

DG_WAITING_EXT_EVENT = 0x40000000 // Wait for external event

DG_WAVEFORM_SENDING = 0x80000000 // Sending the waveform (busy)

Return value

Return true if the function succeeded; false if the function failed.

bool dg64LastError(int iBufSize, char* szBuf)

Get the error code message.

Parameters

iBufSize[in]:

Type: **int**

The buffer size for error code message.

szBuf[in]:

Type: **char***

The buffer for error code message.

Return value

Return true if the function succeeded; false if the function failed.