

Acute Data Generator Software Development Kit (SDK) Programming Guide

For Data Generator 3000 and TravelData 3000

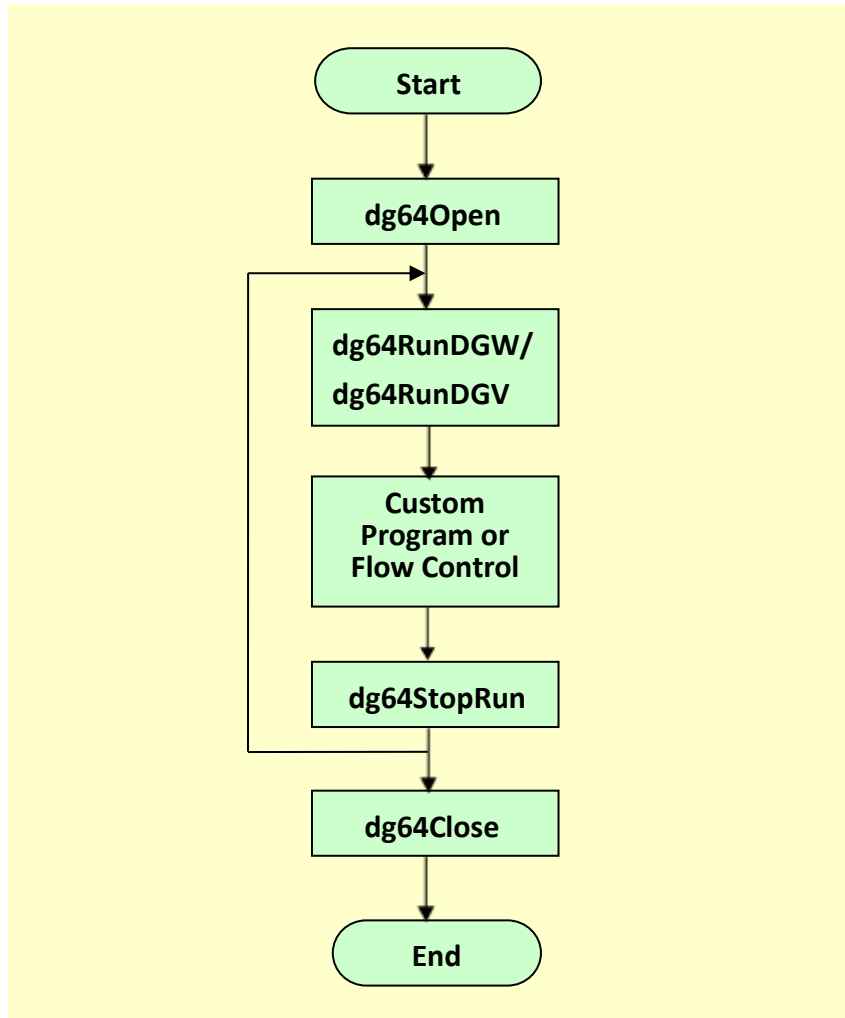
Version: 1.0

Publish: 2020/4/14

内容

SDK 架构介绍	3
HWINFO Structure 介绍	4
SDK 函式说明	4
bool dg64Open().....	4
bool dg64HwInfo(HWINFO *hi).....	4
bool dg64Close().....	5
bool dg64SetPodVolt(int iPod, int iTip, int iVolt).....	5
int dg64RunDGW(char* szFileName).....	5
int dg64RunDGV(char* szFileName).....	5
bool dg64Status(DWORD *pdwStatus).....	6
bool dg64LastError(int iBufSize, char* szBuf).....	6

SDK 架构介绍



此 **SDK** 提供一个开发接口让用户可以产生 DGW/TDW/DGV 等波形档, 并透过数据产生器来发送这些波形档。

HWINFO Structure 介绍

```
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 函式说明

bool dg64Open()

功能

寻找并启动目前连接此计算机上的数据产生器。

回传值

如果回传值为 **True**，代表模式设置成功。如果回传 **False** 值则代表设置失败。

bool dg64HwInfo(HWINFO *hi)

功能

取得数据产生器硬件信息，此信息会被填充在 **HWINFO** 结构中。

回传值

如果回传值为 **True**，代表模式设置成功。如果回传 **False** 值则代表设置失败。

备注

HWINFO cHwInfo = {0};

dgHwInfo(&cHwInfo); // cHwInfo.dwModel : 数据产生器机种名称

// cHwInfo.iUsbSpeed: 连接的USB接口 e.g. USB2.0/USB3.0

bool dg64Close()

功能

关闭此程序对数据产生器的控制，使 SDK 得以释放资源。

回传值

如果回传值为 True，代表模式设置成功。如果回传 False 值则代表设置失败。

int dg64RunDGW(char* szFileName)

int dg64RunDGV(char* szFileName)

功能

载入DGW/TDW/DGV 等波形档并发送。

参数

szFileName[in]:

Type: **char***

档案路径。

回传值

如果回传值为 True，代表模式设置成功。如果回传 False 值则代表设置失败。

// 错误码

```
#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)

功能

取得资料产生器目前状态。

参数

pdwStatus[in]:

Type: **DWORD***

状态:

DG_WAITING_EXT_EVENT = 0x40000000 // Wait for external event

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

回传值

如果回传值为 True，代表模式设置成功。如果回传 False 值则代表设置失败。

bool dg64LastError(int iBufSize, char* szBuf)

功能

取得数据产生器错误讯息。

参数

iBufSize[in]:

Type: **int**

传入缓冲区大小。

szBuf[in]:

Type: **char***

传入的缓冲区。

回传值

如果回传值为 True，代表模式设置成功。如果回传 False 值则代表设置失败。