



## Leddata server center process of sending program and call API

### 1. Bing callback function and then settle DLL return message data in this function.

CallBack(ProcessCallBackMessage);

int WINAPI ProcessCallBackMessage (int Param1, char \* Param2);

parameter: Param1 is the type of message

Param2 is the pointer of message data buffer structure

```
#define MAX_CALLBACK_SERIAL_PACKET_LENGTH 800

typedef struct _STRUCT_GPRS_CALLBAK_DATA_PACKET
{
    char cDtuImei[15+1];
    unsigned char SyncHead[3];
    unsigned char ucCommType; //command type;
    // char ucCardId[SERIAL_NUMBER_SIZE]; //controller serial number
    unsigned char ucDest;
    unsigned char ucSrc;
    unsigned char ucCtrl1;
    unsigned char ucCtrl2;
    // UINT uiImei;
    unsigned int iLength; //length of main body, 0 means no main body, to omit pDataBuffer pointer
    unsigned char pDataBuffer[MAX_CALLBACK_SERIAL_PACKET_LENGTH]; //main body pointer, it should be
    assigned by malloc and released by sending thread
    //unsigned char pDataBuffer[MAX_SERIAL_PORT_PACKET_LENGTH]; //main body pointer, it should be
    assigned by malloc and released by sending thread
}STRUCT_GPRS_CALLBAK_DATA_PACKET,* P_STRUCT_GPRS_CALLBAK_DATA_PACKET;
```

### 2. Call initializing function

DLL\_Net\_Init();

DLL\_GPRS\_Init();

### 3. Call DLL\_GPRS\_Link\_Server, connecting leddata server center

extern "C" \_\_declspec(dllexport) int \_\_stdcall DLL\_GPRS\_Link\_Server(const char \*dc\_host,unsigned short dc\_port,int

block,int iDtuType,char \*cLocalIp);

parameter: iDtuType: must be 2, block: must be 0;

return value: >0: means success; ≤ 0: means connection failed

### 4. Call DLL\_GPRS\_Auth to do user authentication

extern "C" \_\_declspec(dllexport) int \_\_stdcall DLL\_GPRS\_Auth( char \*cUserName, char \*cPassword,int iDtuType);

parameter : iDtuType: must be 2

return value: -1, //user socket error

0, //authentication failed

2, //user not exist

3, //authentication success

4, //password error

7, //login success (repeat logins)

### 5. Call DLL\_Xixun\_Get\_All\_Dtu\_Iimei\_Status to get the DTU list information that



belongs to the user.

### Return DTU information to main software by callback function;

Param1: ENUM\_COMMAND\_GET\_XIXUN\_DTU\_STATUS\_OFFLINE://online DTU

ENUM\_COMMAND\_GET\_XIXUN\_DTU\_STATUS\_ONLINE://offline DTU

Param2(refer to structure STRUCT\_GPRS\_CALLBACK\_DATA\_PACKET): cDtuImei is DTU's IMEI

## 6. Call DLL\_GPRS\_Get\_Imei\_Status to produce an instance for each DTU

```
extern "C" __declspec(dllexport) int __stdcall DLL_GPRS_Get_Imei_Status(char *cModImei,int iDtuType,int  
iCheckInType,int iUdpHoleFlag,int iUdpTcpType);
```

parameter : iDtuType: must be 2, iCheckInType: must be 0, iUdpHoleFlag: must be 0, iUdpTcpType: must be 0

return : Instance handle

```
HANDLE iHandle=DLL_GPRS_Get_Imei_Status(cDtuImei,2,0,0,0);
```

After producing instances, User could get the instance handle corresponding to the IMEI number by DLL\_GPRS\_GetClassHandle at any time.

```
extern "C" __declspec(dllexport) int __stdcall DLL_GPRS_GetClassHandle(char *cImei,int iDtuType);
```

parameter: iDtuType: Must be 2

return value: Instance handle

## 7. call DLL\_Make\_Contab\_File to produce negotiation table file

```
extern "C" __declspec(dllexport) int __stdcall DLL_Make_Contab_File(char *cPanthName,char *cFileName)
```

parameter:cPanthName: storage directory for program material ,

for example,if project name is "d:\\project\\test.xml", then directory of program material willbe "d:\\project\\test"

cFileName: should be "contab.txt";

## 8. Call DLL\_GPRS\_GetSystemStatus controller enters the preparing status of receiving programs

```
extern "C" __declspec(dllexport) int __stdcall DLL_GPRS_GetSystemStatus(void * iHandle/*serial port handle*/,unsigned char  
ucDest,unsigned char ucSrc,unsigned char ucCtrl1,unsigned char ucCtrl2,unsigned char *pPassword);
```

parameter : iHandle: instance handle: ucCtrl1: must be 0x80;ucCtrl2: must be 1; ucDest: 0; ucSrc:0; pPassword:NULL;

return value: 0;

callback function gets Param1=2051, means controller enters preparing status of receiving programs then could do following operations—step 9

## 9. Call DLL\_GPRS\_ReadyToSendProgram controller enters status of receiving programs

```
extern "C" __declspec(dllexport) void __stdcall DLL_GPRS_ReadyToSendProgram(void * iHandle/*serial port handle*/,char  
*cPathName/* directory and name of program */,int iMainVer/*version number*/,int iChildVer/*date version */,unsigned char  
ucDest,unsigned char ucSrc,unsigned char ucCtrl1,unsigned char ucCtrl2);
```

parameter: iHandle: instance handle; cPathName: path file name of the project file; iMainVer: 9; iChildVer: 0x20100915;

ucDest: 0; ucSrc:0; ucCtrl1:0; ucCtrl2:0;

callback function gets Param1=2017, means controller enters the receiving status then could do step 10 operations

## 10. Call DLL\_GPRS\_SendProgramToPanel controller starts to receiving programs

```
extern "C" __declspec(dllexport) void __stdcall DLL_GPRS_SendProgramToPanel(void * iHandle/*serial port*/,char  
*cPanthName/* directory and name of program */,unsigned char ucDest,unsigned char ucSrc,unsigned char ucCtrl1,unsigned char  
ucCtrl2);
```



parameter : iHandle: instance handle; cPathName: path file name of the project file; iMainVer: 9; iChildVer: 0x20100915;  
ucDest: 0; ucSrc:0; ucCtrl1:0; ucCtrl2:0;

callback function gets Param1=2012, means the receiving process

Param2 is structure pointer STRUCT\_GPRS\_CALLBACK\_DATA\_PACKET

STRUCT\_GPRS\_CALLBACK\_DATA\_PACKET\*pSt=(STRUCT\_GPRS\_CALLBACK\_DATA\_PACKET \*)Param2;

int iPrecentValue=((int \*)&pSt->pDataBuffer[0]); //process value, 0~100

callback function gets Param1=2020, means receiving programs finished

NOTE: advices: start thread to send program, send message to the main software interface in callback function; after main software gets the message will draw the process; this is to avoid the block of main software interface.

#### Annotations:

1. Ask controller to download specified program in FTP, set FTP parameters in Hardware setup.

```
extern "C" __declspec(dllexport) void __stdcall Dll_DownloadProgramFromFtp(void * iHandle, char *cImei, char  
*cProgramName, unsigned int uiProgramSize, unsigned int uiMainVer, unsigned int uiChildVer, unsigned int uiWidth, unsigned int  
uiHeight);
```

2. Web server ask controller to download specified programs in FTP, set FTP parameters in Hardware setup

In this demo, Dll\_WebServerDownloadProgramFromFtp parameters cWebServerFunction is only the instance for customer, no actual meanings; customer could refer to passage 5.6 "method webpage" in the document of " Access interface of Web server terminal side 1.2 "

```
extern "C" __declspec(dllexport) void __stdcall Dll_WebServerDownloadProgramFromFtp(void * iHandle, char *cImei, char  
*cProgramName, unsigned int uiProgramSize, unsigned int uiMainVer, unsigned int uiChildVer, unsigned int uiWidth, unsigned int  
uiHeight, char *cServerAdd, unsigned int uiPort, char * cWebServerFunction);
```