GSM Library

for G-4510 series

User's Manual V1.2.0



High Quality, Industrial Data Acquisition, and Control Products

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Warranty

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright 2018 by ICP DAS CO., LTD. All rights reserved worldwide.

Trademark

The names used for identification only may be registered trademarks of their respective companies.

Contact us

If you have any problem, please feel free to contact us.
You can count on us for quick response.

Email：service@icpdas.com

Table of Contents

1. Introduction 1
1.1 4G and PAC Embedded Controller 1
1.2 Design Flowchart 2
2. GSM Library 3
2.1 Data Structure Define 3
2.2 SYSTEM Function 5
2.2.1 GM_SYS_GetLibVersion 6
2.2.2 GM_SYS_GetLibDate 7
2.2.3 GM_SYS_InitModem 8
2.2.4 GM_SYS_CloseModem 9
2.2.5 GM_SYS_CheckModemStatus 10
2.2.6 GM_SYS_CheckCmdStatus 11
2.2.7 GM_SYS_CheckSignal 12
2.2.8 GM_SYS_CheckReg 13
2.2.9 GM_SYS_EnableNITZ 14
2.2.10 GM_SYS_NITZUpdateRTC 15
2.2.11 GM_SYS_CheckNITZ 16
2.2.12 GM_SYS_SetCFUN 17
2.2.13 GM_SYS_GetQCSQ 18
2.2.14 GM_SYS_SetSimType 19
2.2.15 GM_SYS_GetInfo 20
2.3 SMS Function 21
2.3.1 GM_SMS_SendMsg 22
2.3.2 GM_SMS_GetNewMsg 23
2.4 3G / GPRS Data Transmission Function 24
2.4.1 GM_NET_SetNet 25
2.4.2 GM_NET_InstallLink 26
2.4.3 GM_NET_CloseNet 27
2.4.4 GM_NET_GetIP 28
2.4.5 GM_NET_CloseLink 29
2.4.6 GM_NET_GetLinkStatus 30
2.4.7 GM_NET_Send 31
2.4.8 GM_NET_GetNewPacket 32
3. Revision History 33

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1. Introduction

1.1 4G and PAC Embedded Controller

4G is a service that allows information to be sent and received across a mobile telephone network. It supports CSD (Circuit Switched Data), SMS (Short Message Service) and 4G LTE (Long Term Evolution). LTE is NOT related to GPS (the Global Positioning System), a similar acronym that is often used in mobile contexts. 4G offers instant connections whereby information can be sent or received immediately as the need arises, subject to radio coverage. This is why 4G users are sometimes referred to be as being "always connected". Immediacy is one of the advantages of 4G (and SMS) when compared to Circuit Switched Data. High immediacy is a very important feature for time critical applications.

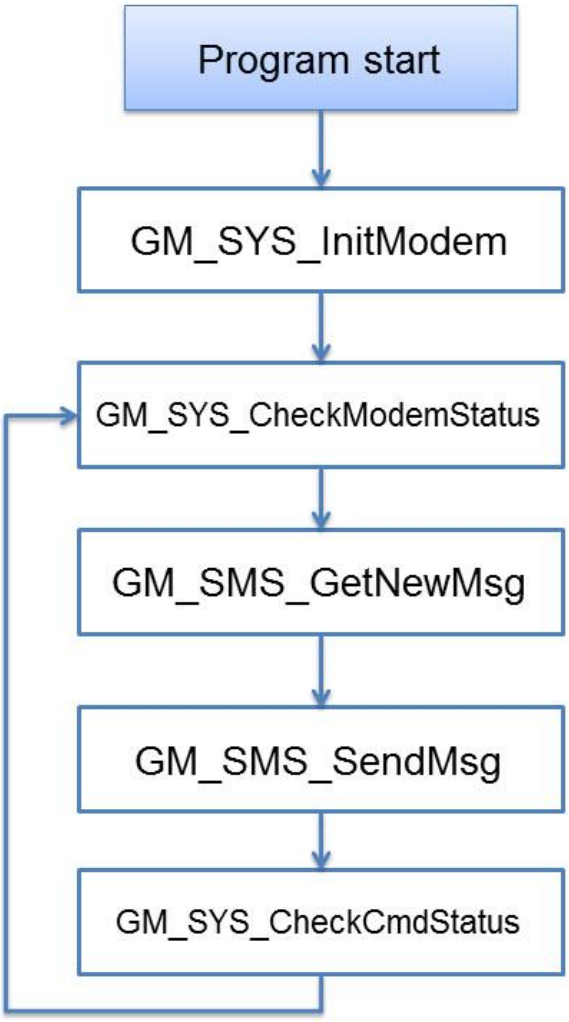
ICP DAS provides the 4G library for PAC embedded controller. The library is an easy way to applying the 4G service in the embedded controller. Otherwise, ICP DAS supports many IO modules and GPS modules for users. Therefore, there are many application architectures to apply in the system. Or users can integrate other controller system with 4G library. The follows is standard application architecture.



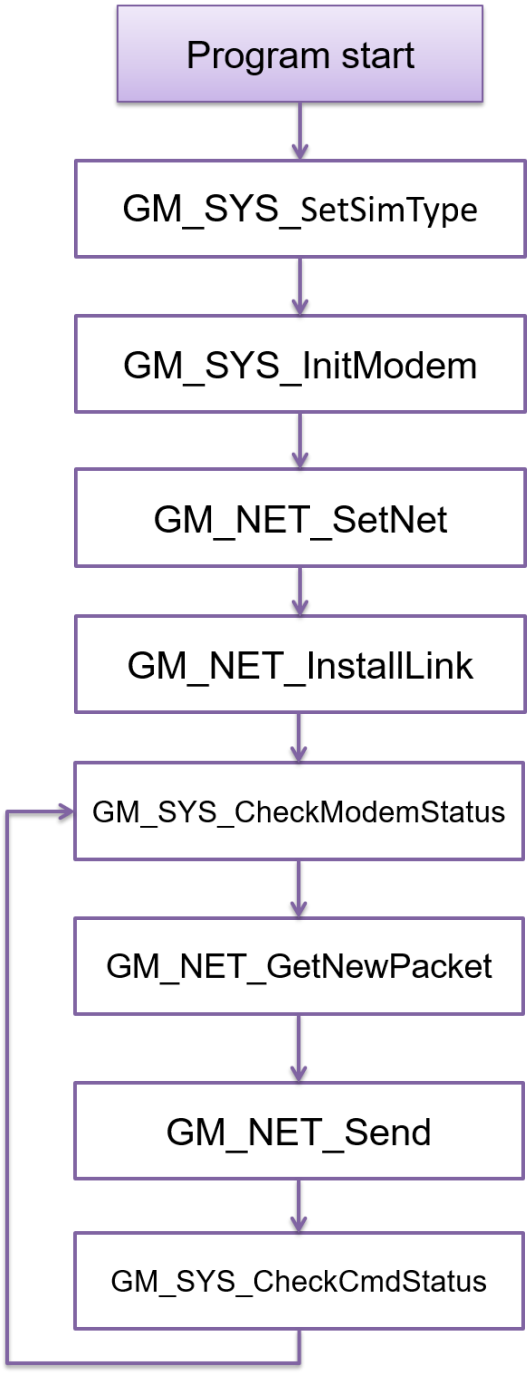
Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1.2 Design Flowchart

SMS Design Flowchart



GPRS Design Flowchart



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2. GSM Library

2.1 Data Structure Define

There are some data structure that is useful when you program with GSM library.

SMS：

```
//-- structure for sending/reading SMS
typedef struct STRENCODE_MSG{
    char phoneNumber[30];     //phone number
    char time[20];            //sms_time_stamp
    char msg[161];            //message's content
    unsigned char dataLen;    //Message's length
                              //Max length: 7-bit=160 words, UCS2=70 words(140 bytes)
    char mode;                //encode style: 0=GSM_7BIT, 8=GSM_UCS2(uni-code)
} strEncode_Msg;
```

GPRS：

```
//-- structure for reading GPRS sockets
typedef struct GPRSDATA{
    char data[1500];        //data
    int dataLen;            //data length
    char fromIP[16];        //IP of the server，000.000.000.000 and '0x00', total 16 byte
    unsigned int port;      //TCP/UDP port of the server
    int link;               //data of link[n]
} GPRSData;

//-- structure for setting network
typedef struct NET_PROFILE
{
    char APN[60];      //APN for network provided by your cellular provider
    char user[32];     //username for network provided by your cellular provider
    char pw[32];       //password for network provided by your cellular provider
    char DnsServerIP[16];   //The most basic task of DNS is to translate hostnames
                            //such as www.icpdas.com to IP address such as 96.9.41.131.
} NetProfile;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

SYSTEM：

```
//-- structure for setting system parameters
typedef struct SYS_PROFILE
{
    char PINCode[5];        //The pin code of SIM card, ex: "0000"
    int modemPort;          //modem port number.
    int hardware;           //hardware type. 0: Other Hardware, 1: G-4514
}SYSProfile;
```

Tips & Warnings



1. The GSM library needs OS7_COM.lib. Please include it.
2. The speed of 4G LTE must not be less than 1 packet / 2 second.
3. The GSM library needs the Timer that installed by "InstallUserTimer()". Please don't collide with it.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2 SYSTEM Function

Function definition	Description
GM_SYS_GetLibVersion	Get Library version
GM_SYS_GetLibDate	Get Library date
GM_SYS_InitModem	Initialize Modem
GM_SYS_CloseModem	Close the modem
GM_SYS_CheckModemStatus	Check modem status, and suggest you check it in your loop every time
GM_SYS_CheckCmdStatus	Get the status of the command you sent
GM_SYS_CheckSignal	Check signal quality
GM_SYS_CheckReg	Check register
GM_SYS_EnableNITZ	Enable NITZ function
GM_SYS_NITZUpdateRTC	Update the RTC of the System by NITZ
GM_SYS_CheckNITZ	Check the status of NITZ
GM_SYS_SetCFUN	Set CFUN if not registered
GM_SYS_GetQCSQ	Get service type, RSRI, RSRP, SINR, RSRQ (For MDM-4400)
GM_SYS_SetSimType	Set service type to register the base station.
GM_SYS_GetInfo	Get service type, operator, band, channel

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2.1 GM_SYS_GetLibVersion

Get library version.

Syntax

```
int GM_SYS_GetLibVersion(void);
```

Parameters

None

Return values

Version format = A.BC

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2.2 GM_SYS_GetLibDate

Get library date.

Syntax

```
void GM_SYS_GetLibDate(
    char* libDate
);
```

Parameters

libDate
a string of lib. date, format="April 2 2018"

Return values

None

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2.3 GM_SYS_InitModem

Initialize Modem.

**must use GM_SYS_CheckModemStatus() to check modem status later

Syntax

```
int GM_SYS_InitModem(
    SYSProfile sysProfile
);
```

Parameters

sysProfile
set system profile

Return values

GM_NOERROR：success
GM_COMERROR：comport error
GM_INITERROR：init fail error

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2.4 GM_SYS_CloseModem

Close the modem.

**Please call GM_SYS_InitModem() to wake up modem after using GM_SYS_CloseModem(1) to shut down the modem.

Syntax

```
int GM_SYS_CloseModem(
    int mode
);
```

Parameters

mode

0：close modem, but maintain it power on

1：close modem and set it power off

Return values

GM_NOERROR：no error

GM_CMDERROR：command error

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2.5 GM_SYS_CheckModemStatus

Check modem status, and suggest you check it in your loop every time.

Syntax

```
int GM_SYS_CheckModemStatus(void);
```

Parameters

None

Return values

GM_NOERROR：modem register success, can service
GM_NOREG：modem not registered, can't service

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2.6 GM_SYS_CheckCmdStatus

Get the status of the command you sent.

Syntax

```
int GM_SYS_CheckCmdStatus(void);
```

Parameters

None

Return values

GM_BUSY：modem busy, you can't send other command
GM_NOERROR：success
GM_TIMEOUT：time out
GM_CMDERROR：command error
Other：please refer to error codes of GSM.h

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2.7 GM_SYS_CheckSignal

Check signal quality.

Syntax

```
int GM_SYS_CheckSignal(void);
```

Parameters

None

Return values

signal quality

0	-113 dBm or less
1	-111 dBm
2...30	-109... -53 dBm
31	-51 dBm or greater

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2.8 GM_SYS_CheckReg

Check register.

Syntax

```
int GM_SYS_CheckReg(void);
```

Parameters

None

Return values

Register flag

0：not registered
1：registered, home network
2：not registered, and searching...
3：registration denied
4：unknown
5：registered, roaming

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2.9 GM_SYS_EnableNITZ

Enable NITZ function.

**NITZ function can auto-adjust RTC of the system at the moment of the modem registering to GSM system.
**Please call "GM_SYS_NITZUpdateRTC" to update RTC after GM_SYS_EnableNITZ(1).

Syntax

```
void GM_SYS_EnableNITZ(
    int nitz
);
```

Parameters

nitz
0：disable
1：enable

Return values

None

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2.10 GM_SYS_NITZUpdateRTC

Update the RTC of the System by NITZ.

**Notice: this function will disable all 4G/3G/GSM function about 1~2 minutes.
**Please use this function after you stop all 4G/3G/GSM function

Syntax

```
void GM_SYS_NITZUpdateRTC(void);
```

Parameters

None

Return values

None

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2.11 GM_SYS_CheckNITZ

Check the status of NITZ.

Syntax

```
void GM_SYS_CheckNITZ(void);
```

Parameters

None

Return values

0：fail to update RTC
1：success
2：updating

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2.12 GM_SYS_SetCFUN

Set CFUN function start if not registered.

Syntax

```
int GM_SYS_SetCFUN(void);
```

Parameters

None

Return values

None

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2.13 GM_SYS_GetQCSQ

Get Service type, RSRI, RSRP, SINR, RSRQ.

※For MDM-4400 only.

Syntax

```
void GM_SYS_GetQCSQ(char *Qstr);
```

Parameters

Qstr

a string of QCSQ.

NB1,M1：

"Service type",LTE_RSRI,LTE_RSRP,LTE_SINR,LTE_RSRQ

GSM：

"Service type",GSM_RSRI

Return values

None

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2.14 GM_SYS_SetSimType

Set service type to register the base station.

Syntax

```
int GM_SYS_SetSimType(int type);
```

Parameters

Mode

MDM-4400：

0：GSM

8：Cat-M1

9：Cat-NB1

MDM-4001-AU：

0：GSM

2：UTRAN

3：GSM W/EGPRS

4：UTRAN W/HSDPA

5：UTRAN W/HSUPA

6：UTRAN W/HSDPA and HSUPA

7：E-UTRAN

Return values

None

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2.15 GM_SYS_GetInfo

Get Service type, operator, band, channel.

Syntax

```
void GM_SYS_GetInfo(char *info);
```

Parameters

Info

The info is string.

"Service type","Operator","Band","Channel"

Return values

None

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.3 SMS Function

Function definition	Description
GM_SMS_SendMsg	Send a message
GM_SMS_GetNewMsg	Get a new sms message

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.3.1 GM_SMS_SendMsg

Send a message.

**must use "GM_SYS_CheckCmdStatus()" to check status later

Syntax

```
int GM_SMS_SendMsg(
    strEncode_Msg* strMsg
);
```

Parameters

strMsg
the message that will be sent.

Return values

GM_NOERROR：no error
GM_NOREG：not registered, or can't service
GM_BUSY：modem busy

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.3.2 GM_SMS_GetNewMsg

Get a new sms message.

Syntax

```
int GM_SMS_GetNewMsg(
    strEncode_Msg* msg
);
```

Parameters

msg

new sms message

Return values

0：no new message

1：new message coming

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.4 3G / GPRS Data Transmission Function

Function definition	Description
GM_NET_SetNet	Set Net profile data
GM_NET_InstallLink	Built TCP/UDP link
GM_NET_CloseNet	Close Network
GM_NET_GetIP	Get local IP
GM_NET_CloseLink	Close client link[n]
GM_NET_GetLinkStatus	Get status of Link[n]
GM_NET_Send	Send a packet
GM_NET_GetNewPacket	Get the new packet

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.4.1 GM_NET_SetNet

Set Net profile data.

Syntax

```
int GM_NET_SetNet(
    NetProfile netProfile
);
```

Parameters

netProfile
Net profile data

Return values

GM_NOERROR：no error
GM_CMDERROR：command error

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.4.2 GM_NET_InstallLink

Built TCP/UDP link.

Syntax

```
int GM_NET_InstallLink(
    int n,
    int tcp,
    char* serverIP,
    unsigned int serverPort
);
```

Parameters

n

link number (0~9)
4G (G-4514 series)：0~9

tcp

client type, tcp=1 for TCP client ; tcp=0 for UDP client

serverIP

IP or Domain name of the server, ex: "61.111.222.333", "test.com.tw"

serverPort

TCP/UDP Port of the server (1~65535), ex: 1234

Return

GM_NOERROR：correct parameter to install TCP/UDP link
GM_CMDERROR：command error

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.4.3 GM_NET_CloseNet

Close Network.

Syntax

```
int GM_NET_CloseNet(void);
```

Parameters

None

Return values

GM_NOERROR：no error
GM_CMDERROR：command error
GM_BUSY：modem busy

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.4.4 GM_NET_GetIP

Get local IP.

Syntax

```
void GM_NET_GetIP(
    char* ipaddr
);
```

Parameters

ipaddr

IP string, format: char ipaddr[16];

Return values

None

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.4.5 GM_NET_CloseLink

Close client link[n].

Syntax

```
int GM_NET_CloseLink(
    int n
);
```

Parameters

n

4G (G-4514 series)：0~9

Return values

GM_NOERROR：no error
GM_CMDERROR：command error
GM_BUSY：modem busy

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.4.6 GM_NET_GetLinkStatus

Get status of Link[n].

Syntax

```
int GM_NET_GetLinkStatus(
    int n
);
```

Parameters

n

4G (G-4514 series)：0~9

Return values

0：not link

1：linked

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.4.7 GM_NET_Send

Send a packet.
**must use "GM_SYS_CheckCmdStatus()" to check status later

Syntax

```
int GM_NET_Send(
    char link,
    char* data,
    int dataLen
);
```

Parameters

link
link number
4G (G-4514 series)：0~9

data
data that will be sent

dataLen
data length, Max.=1000

Return values

GM_NOERROR：no error
GM_CMDERROR：command error
GM_BUSY：modem busy

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.4.8 GM_NET_GetNewPacket

Get the new packet.

Syntax

```
int GM_NET_GetNewPacket(
    GPRSData* gprsData
);
```

Parameters

gprsData
new data packet

Return values

0：no new packet
1：new packet coming

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

3. Revision History

Revision	Date	Author	Description
1.2.0	2019/08/21	Amon	Release version

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000