

GSM library for GTM-201

User Manual

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 2011 by ICP DAS Co., LTD. All rights reserved worldwide.

Trademark

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

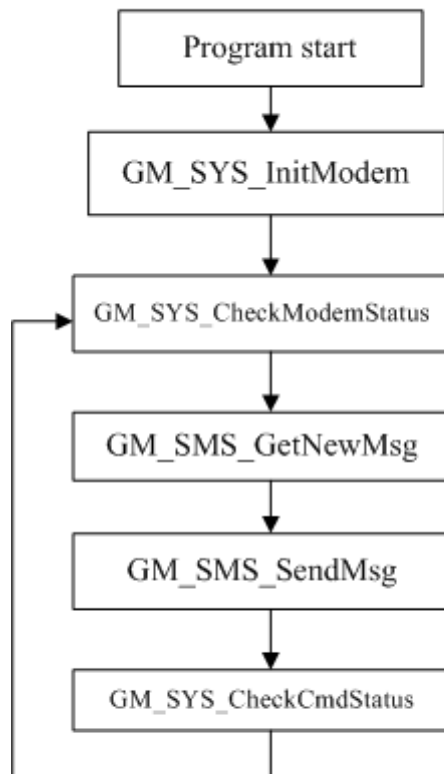
Table of Contents

| | |
|-------------------------------------|----------|
| CHAPTER 1 INTRODUCTION | 3 |
| 1.1 DESIGN FLOWCHART | 3 |
| CHAPTER 2 GSM LIBRARY | 4 |
| 2.1 DATA STRUCTURE DEFINE | 4 |
| 2.2 FUNCTION..... | 5 |
| --SYSTEM Function--..... | 5 |
| 2.2.1 GM_SYS_GetLibVersion | 5 |
| 2.2.2 GM_SYS_GetLibDate | 6 |
| 2.2.3 GM_SYS_InitModem | 7 |
| 2.2.4 GM_SYS_CloseModem | 8 |
| 2.2.5 GM_SYS_CheckModemStatus..... | 9 |
| 2.2.6 GM_SYS_CheckCmdStatus..... | 10 |
| 2.2.7 GM_SYS_CheckSignal | 11 |
| 2.2.8 GM_SYS_CheckReg | 12 |
| --SMS Function--..... | 13 |
| 2.2.9 GM_SMS_SendMsg | 13 |
| 2.2.10 GM_SMS_GetNewMsg..... | 14 |

Chapter 1 Introduction

1.1 Design Flowchart

SMS Design Flowchar



Chapter 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;
```

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: GTM-201
}SYSProfile;
```

2.2 Function

--SYSTEM Function--

2.2.1 GM_SYS_GetLibVersion

Prototype:

```
int GM_SYS_LibVersion(void);
```

Description:

Get Lib. version

Parameter:

no

Return:

Version format = A.BC

2.2.2 GM_SYS_GetLibDate

Prototype:

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

Description:

Get lib. date

Parameter:

date, format="Jul 21 2010"

Return:

no

2.2.3 GM_SYS_InitModem

Prototype:

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

Description:

Initialize Modem

*must use GM_SYS_CheckModemStatus() to check modem status later

Parameter:

sysProfile: set system profile

Return:

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

2.2.4 GM_SYS_CloseModem

Prototype:

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

Description:

Close the modem

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

Parameter:

mode: 0: close modem, but maintain it power on
1: close modem and set it power off (only for MiniOS7)

Return:

GM_NOERROR: no error
GM_CMDERROR: command error

2.2.5 GM_SYS_CheckModemStatus

Prototype:

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

Description:

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

Parameter:

no

Return:

GM_NOERROR: modem register success, can service

GM_NOREG: modem not registered, can't service

2.2.6 GM_SYS_CheckCmdStatus

Prototype:

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

Description:

Get the status of the command you sent

Parameter:

no

Return:

| | |
|---------------|--|
| 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 |

2.2.7 GM_SYS_CheckSignal

Prototype:

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

Description:

Check signal quality

Parameter:

no

Return:

| Signal: | signal quality |
|---------|--------------------|
| 0 | -113 dBm or less |
| 1 | -111 dBm |
| 2...30 | -109... -53 dBm |
| 31 | -51 dBm or greater |

2.2.8 GM_SYS_CheckReg

Prototype:

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

Description:

Check register

Parameter:

no

Return:

Register flag

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

--SMS Function--

2.2.9 GM_SMS_SendMsg

Prototype:

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

Description:

Send a message

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

Parameter:

strMsg: the message

Return: None

GM_NOERROR no error

GM_NOREG: not registered, or can't service

GM_BUSY: modem busy

2.2.10 GM_SMS_GetNewMsg

Prototype:

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

Description:

Get a new sms message

Parameter:

msg: new sms message

Return:

0: no new message

1: new message coming

Version Record

| Version | By | Date | Description |
|---------|------|------------|-------------|
| 1.0.0 | Malo | 2011/04/18 | release |
| | | | |
| | | | |
| | | | |