

# **GSM .Net 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.

---

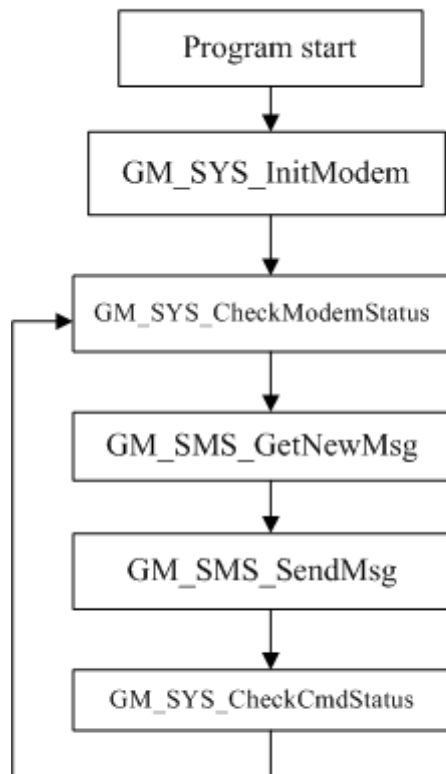
## Table of Contents

<b>CHAPTER 1 INTRODUCTION</b> .....	<b>3</b>
1.1 DESIGN FLOWCHART .....	3
<b>CHAPTER 2 GSM LIBRARY</b> .....	<b>4</b>
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:

```
//-- class for sending/reading SMS
public class Encode_Msg
{
    public string phoneNumber;    //phone number
    public string time;          //sms_time_stamp
    public string msg;           //message's content
    public int mode;             //encode style: 0=GSM_7BIT, 8=GSM_UCS2(uni-code)
}
```

## 2.2 Function

### --SYSTEM Function--

#### 2.2.1 GM\_SYS\_GetLibVersion

##### Prototype:

```
public static string GM_SYS_GetLibVersion()
```

##### Description:

Get Lib. version

##### Parameter:

No

##### Return:

Version

## 2.2.2 GM\_SYS\_GetLibDate

### Prototype:

```
public static string GM_SYS_GetLibDate()
```

### Description:

Get the library's date

### Parameter:

No

### Return:

The library's date, format="Jul 21 2010"

### 2.2.3 GM\_SYS\_InitModem

#### Prototype:

```
public static int GM_SYS_InitModem(string PinNum, int modem)
```

#### Description:

Initialize Modem

\*must use GM\_SYS\_CheckModemStatus() to check modem status later

#### Parameter:

PinNum: PIN code

modem: port number of the modem

#### Return:

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

## 2.2.4 GM\_SYS\_CloseModem

### Prototype:

```
public static 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

### Return:

GM\_NOERROR: no error  
GM\_CMDERROR: command error



## 2.2.5 GM\_SYS\_CheckModemStatus

### Prototype:

```
public static int GM_SYS_CheckModemStatus();
```

### Description:

Check the status of the modem; 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:

```
public static int GM_SYS_CheckCmdStatus();
```

### 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:

```
public static int GM_SYS_CheckSignal();
```

### 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:

```
public static int GM_SYS_CheckReg();
```

### 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:

```
public static int GM_SMS_SendMsg(Encode_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:

```
public static int GM_SMS_GetNewMsg(ref Encode_Msg strMsg)
```

### 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