Chapter 17: SMS: Short Message Service

The I-8417/8817/8437/8837, I-7188EG, I-7188XG & Wincon-8xx7 controller can integrate with a GSM Modem to support SMS: Short Message Service. This allows user to request information or control something from his own cellular phone to the ISaGRAF controller. Beside, the controller can also send information and alarms to user’s cellular phone.

17.1: Hardware Installation

The I-8417/8817/8437/8837 supports SMS since its driver version of 2.24, while version 1.14 for I-7188EG, and version 1.12 for I-7188XG. If your driver is older one, please upgrade the hardware driver to the associate version or a higher version. The driver can be found from the below ICP DAS’s FTP site:


The I/O library should be re-installed if yours is older one. Please refer to section 1.2. Or you can refer to Appendix A.2 to simply install “C functions” with the below items.

   SMS_test, SMS_get, SMS_gets, SMS_send, SMS_sts
   and “I/O complex equipment” : SMS.

The GSM Modem GM29 (900/1800) is recommended for the ISaGRAF controller since its driver version of I-8xx7:2.47, I-7188EG:1.38, I-7188XG:1.35 & Wincon-8xx7:3.09. You may purchase them from ICP DAS or from your local agent. ICP DAS is not sure for other GSM modems working or not.

Note: Please REMOVE the password setting in SIM card, then plug it into GSM modem.
17.2: A SMS demo example

The demo project is located at demo_43, please refer to section 11.1 to install it to your ISaGRAF workbench. Or it can be downloaded at ICP DAS’s ftp site.

Variables:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>Boolean</td>
<td>Internal</td>
<td>Trigger to send an alarm message when K1 is pushed</td>
</tr>
<tr>
<td>M2</td>
<td>Boolean</td>
<td>Internal</td>
<td>Trigger to send a report message when a message is coming</td>
</tr>
<tr>
<td>K1</td>
<td>Boolean</td>
<td>Input</td>
<td>Pushbutton 1, connect to push4key</td>
</tr>
<tr>
<td>L1</td>
<td>Boolean</td>
<td>Output</td>
<td>Output 1, connect to show3led</td>
</tr>
<tr>
<td>L2</td>
<td>Boolean</td>
<td>Output</td>
<td>Output 2, connect to show3led</td>
</tr>
<tr>
<td>L3</td>
<td>Boolean</td>
<td>Output</td>
<td>Output 3, connect to show3led</td>
</tr>
<tr>
<td>Q1</td>
<td>Boolean</td>
<td>Internal</td>
<td>Test if message is coming</td>
</tr>
<tr>
<td>TMP</td>
<td>Boolean</td>
<td>Internal</td>
<td>Temporary usage</td>
</tr>
<tr>
<td>SMS_available</td>
<td>Boolean</td>
<td>Input</td>
<td>is SMS available? connect to SMS - status</td>
</tr>
<tr>
<td>T1</td>
<td>Timer</td>
<td>Internal</td>
<td>Blinking time of L1 to L3, init at T#500ms</td>
</tr>
<tr>
<td>data</td>
<td>Message</td>
<td>Internal</td>
<td>The coming Message</td>
</tr>
<tr>
<td>phone</td>
<td>Message</td>
<td>Internal</td>
<td>phone No. of sender</td>
</tr>
<tr>
<td>Date_time</td>
<td>Message</td>
<td>Internal</td>
<td>Message coming date &amp; time in string format</td>
</tr>
<tr>
<td>To_who</td>
<td>Message</td>
<td>Internal</td>
<td>phone No of receiver, please use your own No.</td>
</tr>
<tr>
<td>Msg_to_send</td>
<td>Message</td>
<td>Internal</td>
<td>Message to send out</td>
</tr>
<tr>
<td>Year1</td>
<td>Integer</td>
<td>Internal</td>
<td>Message coming year</td>
</tr>
<tr>
<td>Mon1</td>
<td>Integer</td>
<td>Internal</td>
<td>Message coming month</td>
</tr>
<tr>
<td>Day1</td>
<td>Integer</td>
<td>Internal</td>
<td>Message coming date</td>
</tr>
<tr>
<td>Wday1</td>
<td>Integer</td>
<td>Internal</td>
<td>Message coming week date</td>
</tr>
<tr>
<td>Hour1</td>
<td>Integer</td>
<td>Internal</td>
<td>Message coming hour</td>
</tr>
<tr>
<td>Min1</td>
<td>Integer</td>
<td>Internal</td>
<td>Message coming minute</td>
</tr>
<tr>
<td>Sec1</td>
<td>Integer</td>
<td>Internal</td>
<td>Message coming second</td>
</tr>
<tr>
<td>Q1_cnt</td>
<td>Integer</td>
<td>Internal</td>
<td>Message coming count, declared as retained variable</td>
</tr>
<tr>
<td>Msg_status</td>
<td>Integer</td>
<td>Internal</td>
<td>Message sending status</td>
</tr>
<tr>
<td>TMP_v</td>
<td>Integer</td>
<td>Internal</td>
<td>temporary usage</td>
</tr>
</tbody>
</table>

Project architecture:

Operation actions:
1. If K1 is pushed, an Alarm message will be sent.
2. If the user sends a message in format, for example, T0200 or T1500 to the controller, the blinking period will change to 200ms and 1500ms. And then the controller will respond a report message back to the user.
I/O connection:

LD program: work

- Trigger to send an alarm message when K1 is pushed
- Get message Sending status every scan cycle
- Blink outputs
- Message coming count, Q1_cnt is declared as retained variable
ST program: rcv_msg

Q1 := SMS_test();

if Q1 then

Year1 := SMS_get(1);
Mon1 := SMS_get(2);
Day1 := SMS_get(3);
Wday1 := SMS_get(4);
Hour1 := SMS_get(5);
Min1 := SMS_get(6);
Sec1 := SMS_get(7);

phone := SMS_gets(2);
date_time := SMS_gets(3);
data := SMS_gets(1);

Check the coming message. For ex. T1500 will result T1=1500 ms, while T0300 result T1=300ms, however TAB10 will result T1=0 ms (not valid)

if mid(data,1,1) = 'T' then

TMP_v := ANA(mid(data,4,2));

(* valid format *)
if TMP_v >= 50 and TMP_v <= 9999 then
  T1 := TMR(TMP_v); (* convert to timer *)

Msg_to_send := 'Current T1 change to ' + Msg(TMP_v) + ' ms.';
M2 := TRUE;

else (* invalid format*)

Msg_to_send := '!!! Wrong command, Val should be between T0050 to T9999. Current T1 remains at ' + Msg(Ana(T1)) + ' ms.';
M2 := TRUE;

end_if;

end_if; (* if mid(data,1,1) = 'T' then *)

end_if; (* if Q1 then *)
ST program : snd_msg

if (Msg_status <> 1) and SMS_available then
  if M1 then (* alarm triggering *)
    TMP := SMS_send(to_who,'K1 is pushed!');
    M1 := FALSE;
  elsif M2 then (* Report triggering *)
    TMP := SMS_send(phone,Msg_to_send); (* report message back *)
    M2 := FALSE;
  end_if;
end_if;

More description of SMS_sts, SMS_send, SMS_test, SMS_get & SMS_gets, Please refer to ISaGRAF’s On-line Help. “Library” – “C functions” – “SMS_xxxx”

Message sending status:
0: waiting for a new sending request
1: busy. (message is processing now)
21: The message is sent successfully
-1: SMS system is not available
-2: Timeout, No response.

if message sending status is not 1:busy
Must disable it (set to FALSE) after SMS_send is called

Message sending status:
0: waiting for a new sending request
1: busy. (message is processing now)
21: The message is sent successfully
-1: SMS system is not available
-2: Timeout, No response.
**SMS_GET**

Description:
Function  Get message date and time from controller's date & time

Arguments:

<table>
<thead>
<tr>
<th>REF_</th>
<th>Integer</th>
<th>to get what? , 1 ~ 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: get year, (N_ = 2000 ~ 2099 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2: get month, (N_ = 1 ~ 12 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3: get date, (N_ = 1 ~ 31 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4: get week date,(N_ = 1 ~ 7, 7 means Sunday )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5: get hour, (N_ = 0 ~ 23 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6: get minute, (N_ = 0 ~ 59 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7: get second, (N_ = 0 ~ 59 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

others: return N_=-1 : error

return:

| N_ | Integer | Return associated with Ref_. If return -1, it may be "No message" or Ref_ out of range of 1 ~ 7 |

Note:

1. SMS_gets & SMS_get can be called to get message
2. After SMS_gets(1) is called (get message data), the message buffer will reset to "No message". So if the orther information are need, please call SMS_get(1~7) & SMS_gets(2) & SMS_gets(3) before calling SMS_gets(1)

Example: demo_43
SMS.GETS

Description:
Function    Get message data and other information

Arguments:
REF_        Integer  to get what ?,  1 ~ 3

1: get message data
2: get phone No. of sender
3: get date & time in string format

others: return M_ = 'error'

return:

M_        Message    Return associated with Ref_. If return 'error', it may be "No message" or Ref_ out of range of 1 ~ 3

Note:
1. SMS_gets & SMS_get can be called to get message
2. After SMS_gets(1) is called (get message data), the message buffer will reset to "No message". So if the other information are need, please call SMS_get(1~7) & SMS_gets(2) & SMS_gets(3) before calling SMS_gets(1)

Example:  demo_43
**SMS_SEND**

**Description:**
Trigger the controller to send a new message

**Arguments:**

- **No_** message to which phone No., e.g. '+886920119135', max len is 31 digits
- **M_** message the message to send

**Return:**

- **Q_** Boolean True: ok., False: wrong phone No or "message sending status" is not 0 or 21

**Note:**

1. Please call SMS_sts to get the "Message Sending status" before calling SMS_send. SMS_send only works when status is not 1:busy

2. A successfully SMS_send request will reset the "Message sending status" to "1:busy", and after that, by the time, it will set to the associate status. For ex. 21:successfully sent

**Example:** demo_43
**SMS_STS**

**Description:**
Function: Get Message Sending status

**return:**

N_ Integer

0: waiting for a new sending request
1: busy. (One message is processing now)
21: The message is sent successfully

-1: SMS system is not available (Check GSM Modem & SIM card)
-2: Timeout, No response. (It May be no such a phone No.)

**Note:**

1. Please call SMS_sts to get the "Message Sending status" before calling SMS_send. SMS_send only works when status is not 1:busy

2. A successfully SMS_send request will reset the "Message sending status" to "1:busy", and after that, by the time, it will set to the associate status. For ex. 21:successfully sent

**Example:** demo_43
SMS_TEST

Description: Test if message coming or not

return:

\[
\text{Q} \quad \text{Boolean} \quad \text{TRUE: A message is coming, FALSE: No message}
\]

Note:

1. SMS_gets & SMS_get can be called to get message
2. After SMS_gets(1) is called (get message data), the message buffer will reset to "No message". So if the other information are need, please call SMS_get(1~7) & SMS_gets(2) & SMS_gets(3) before calling SMS_gets(1)

Example: demo_43
**I_RESET**

**Description:**

Function  Reset the controller

*** Target:  I-8417/8817/8437/8837

return:

Q  boolean   The return value has no meaning since the controller will reset

**Note:**

Please use this function very careful. If the controller is always reset, please refer to section 1.3.7 of the "User's Manual Of The I-8417/8817/8437/8837" to delete the project inside the controller.

**Example:**

(* OK1 is declared as boolean input, TMP as boolean internal *)

if  OK1=TRUE  then
   TMP := i_reset();
end_if;

(
TIME_STR

Description: Convert date & time to string format

Arguments:

YEAR_ integer year, 2000 ~
MON_ integer month, 1 ~ 12 (January ~ December)
MDAY_ integer mday, 1 ~ 31
WDAY_ integer wday, 1 ~ 7 (Monday ~ Sunday)
HOUR_ integer hour, 0 ~ 23
MIN_ integer minute, 0 ~ 59
SEC_ integer second, 0 ~ 59

If given wrong input parameters will return M_ = " (empty string). For. ex. give MON_=14

return:

M_ message length is 24 characters. For ex. 'Feb/18/2003,13:25:45,Tue'

Note: Please use sysdat_r & systim_r to get system date & time