FF-AUTOMATION

AutoLog® GSM-PLC

Complete

SERVICE MANUAL

FOR ALL AutoLog GSM-PLCs

1 INTRODUCTION

1.1 AutoLog GSM-PLC

AutoLog GSM-PLC is like normal PLC (Programmable Logic Controller) which has I/O and programming capabilities etc, but it also has build-in GSM modem and easy to use GSM functionalities like sending measurements in SMS message or using GPRS, receiving commands via SMS or call, identifying incoming phone numbers etc.

AutoLog GSM-PLC is rock solid platform for any GSM monitoring or control purposes.

1.1.1 Inputs & Outputs (I/O)

In typical application, GSM-PLC is connected to target to measure different kinds of analog values, like temperatures, pressures, surface levels, voltages, currents etc. and digital ON/OFF status information. GSM-PLCs have also digital and analog outputs.

GSM-PLC has selectable analog input modules for all standard measurement ranges (0-20mA, 4-20mA, 0-5V, Pt100, RMS Voltage, and RMS Current etc.) so almost any sensor in the market can be connected to GSM-PLC. In addition GSM-PLC has free serial port which supports Modbus RTU protocol, so it can communicate directly with other PLCs, SCADA systems or intelligent meters / actuators etc.

1.1.2 Control program (Application program)

GSM-PLC is Programmable logic controller (PLC) which means that user can write control program which is capable of doing automation tasks like controlling actuators (e.g. pumps, motors) according to measurements (e.g. surface level, pressure, temperature, status), it can calculate operating times and counts and send measurement reports and alarm messages. It can also make RTC based controlling and PID controlling.

1.1.3 Data logging

One typical task for GSM-PLC in many applications is to log measurement data into its memory. GSM-PLC is capable of storing for example 16 measurements with time-stamp in one minute periods for 7 days. Stored log can be send to SCADA system using GPRS data transfer. This allows that GSM-PLC doesn't need to be on-line all the time in order to send measurement data to server. This is important especially in low power applications.

1.1.4 Alarm generation

GSM-PLC can generate alarms when a measurement goes bellow / above defined alarm limit or e.g. by combining many conditions (*diagnostic alarms*). Alarm is time critical information and

it can be send immediately to SCADA server, from where it can be forwarded as text GSM/SMS message or e-mail to defined (on-duty) service man.

1.1.5 Other Features

GSM-PLC has many other useful features like:

- remote programmability via GSM network
- RS232/RS485 Modbus master/slave connection to other systems
- I2C connection to AutoLog HMI front panel
- connection to AutoLog wireless sensor network (WSN) I/O modules etc,

1.1.6 SCADA connections

- AutoLog ControlMan Cloud SCADA Service
- Indusoft Web Studio SCADA software and AutoLog GSM driver
- Any SCADA with ODBC interface and AutoLog GSM driver

1.1.7 Typical applications

- pumping stations
- oil and gas pipelines, corrosion protection, valve controls, wellheads
- street light monitoring and energy saving (AutoLogStreet Light Control System)
- supervision and GSM alarm systems
- Cloud SCADA systems
- Special monitoring / controlling systems
- OEM monitoring / controlling applications
- Cold station monitoring and controlling
- Low power – solar panel / accumulator powered applications
- Any GSM monitoring / controlling system

More info can be found form AutoLog GSM-RTU pages!

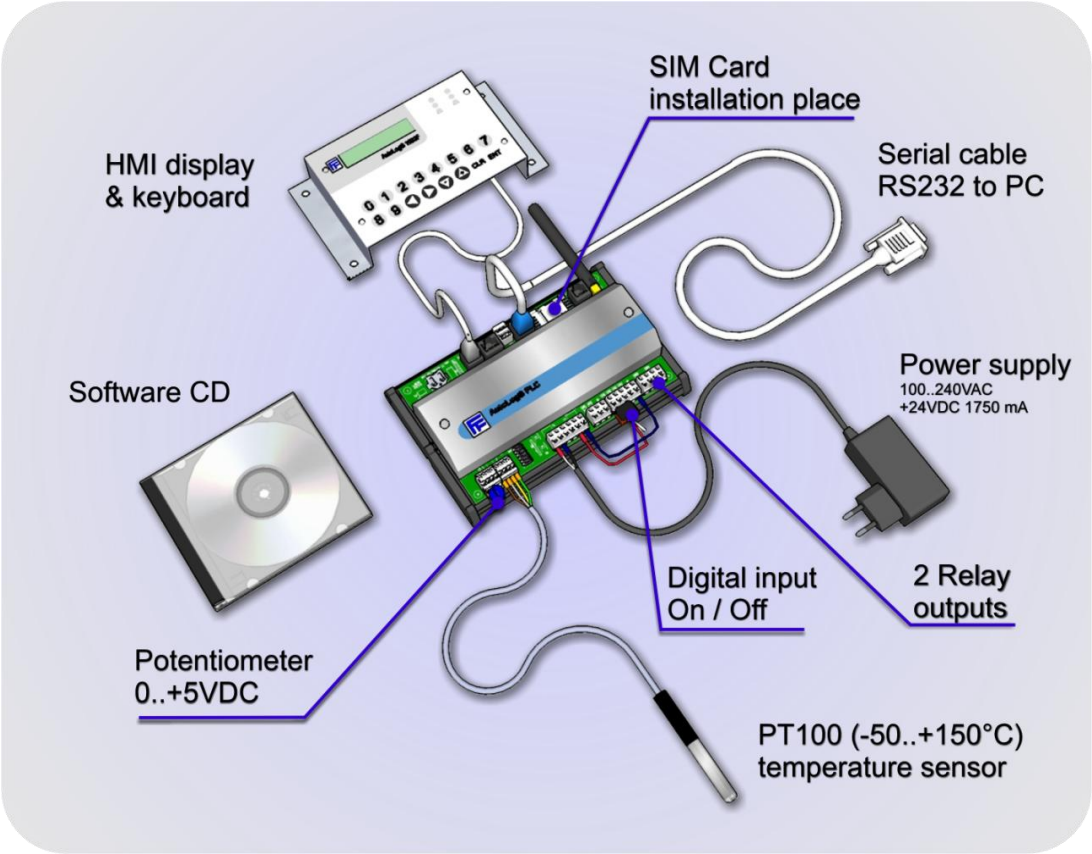
GETTING STARTED

2 Getting started with GSM-PLC hardware installation

2.1 Purpose of this chapter

This chapter is giving you quick instructions, how to set up GSM-PLC hardware ready for testing. The example pictures are from GSM-8 Demo Kit, but the same principles are working with other GMS models. Look the “GSM-PLC Hardware” part of this manual for more information of other GSM-PLC models.

2.2 System elements and connectors

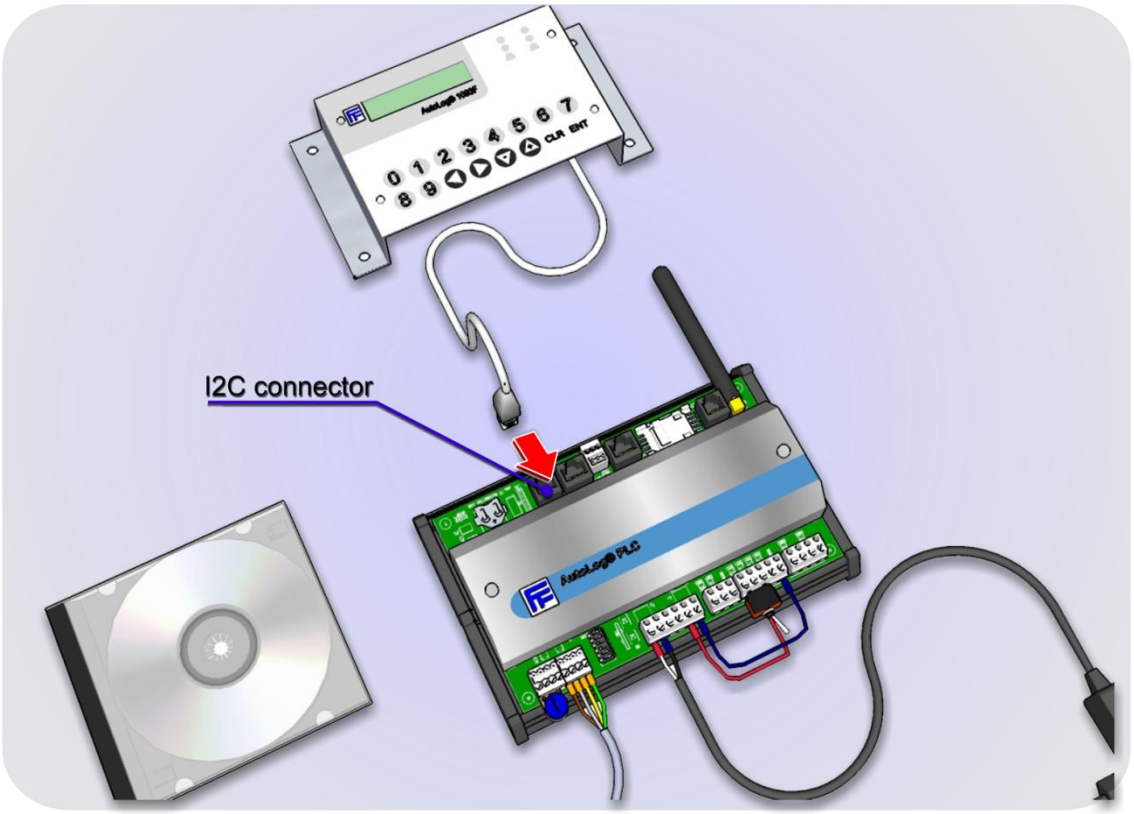


Pic 1: AutoLog® GSM-8 Demo kit.

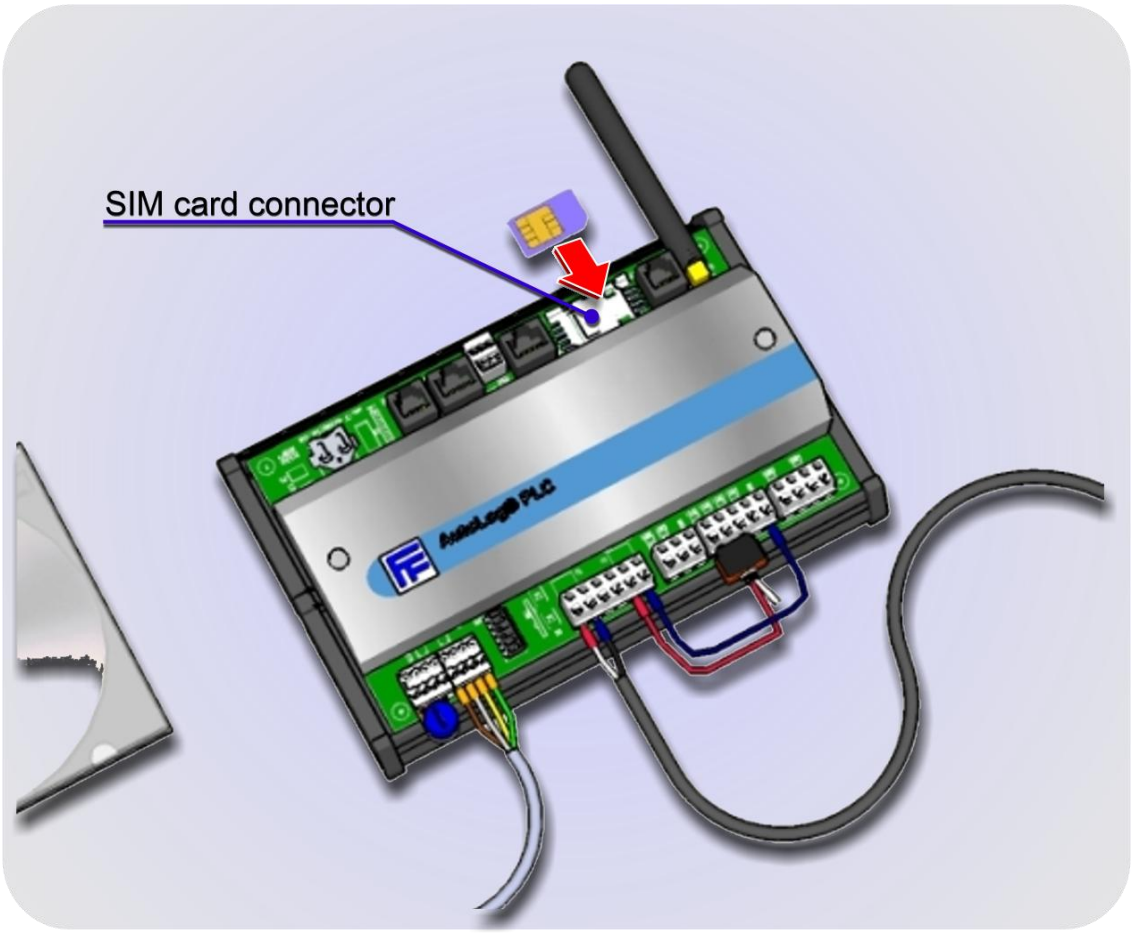
GSM-8 Demo Kit includes the above (Pic 1) components. Make sure you have all the components. Check that the cable connectors are not loose.

SIM card is not included to GSM-8 Demo Kit or any of our GSM-PLCs. You need to purchase it locally. Make sure it supports SMS messages and GPRS data.

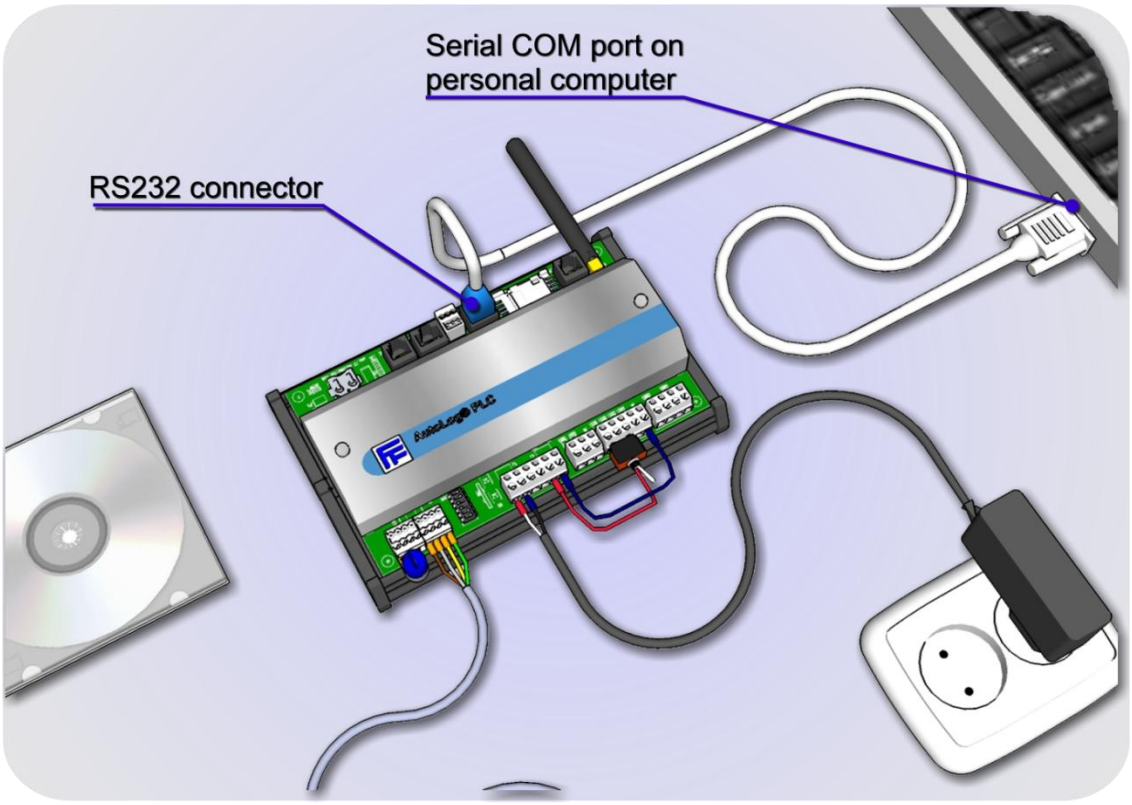
Power supply: GSM-8 power input is 24VDC (12-30VDC). In the picture the power supply is a transformer which transforms 100…240VAC to 24VDC. Do not connect 110VAC or 230VAC directly to PLC!!!



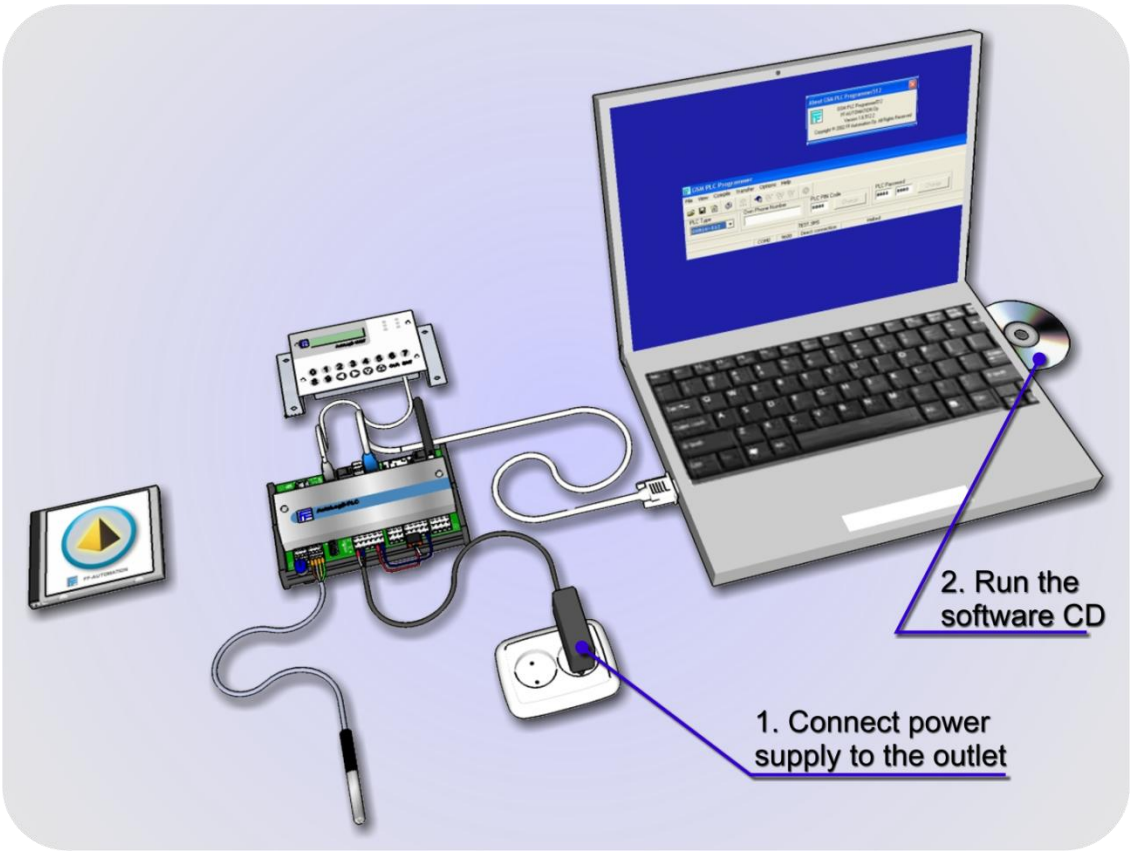
Pic 2: Connect HMI display to I²C connector.



Pic 3: Install SIM card into SIM connector. Close it firmly.



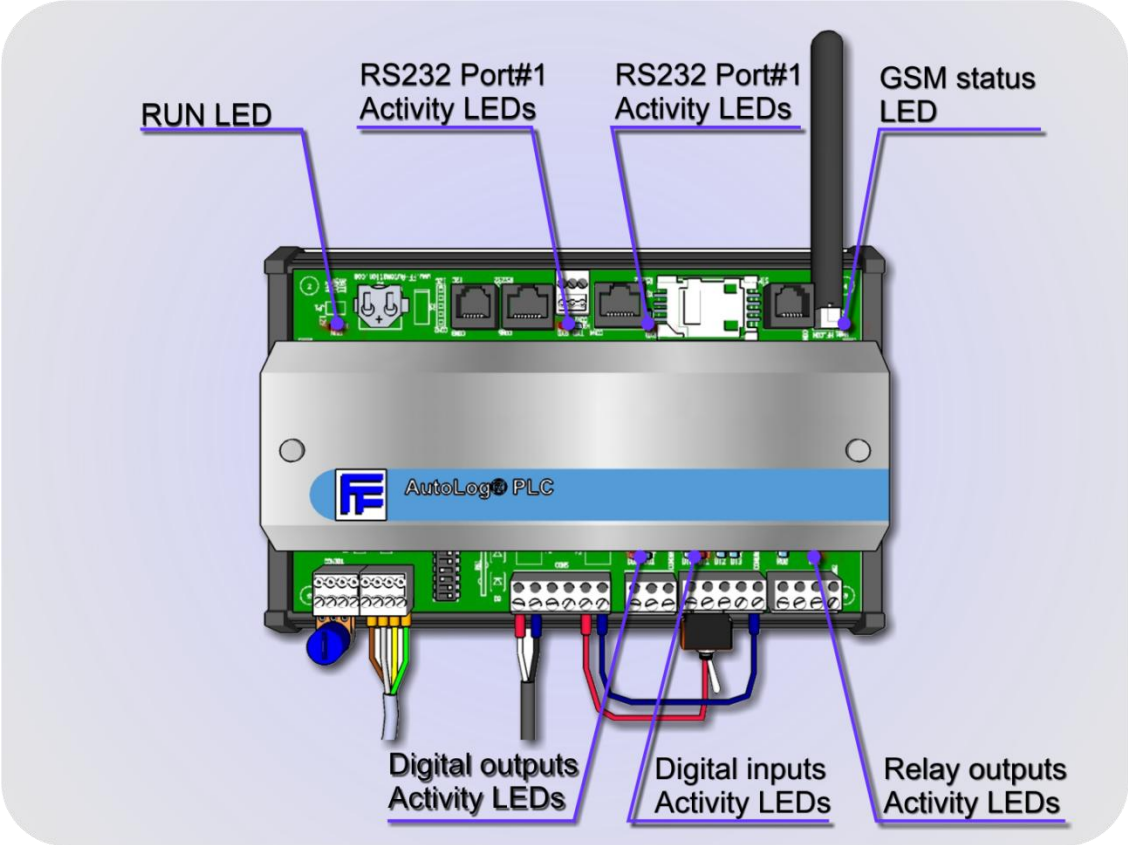
Pic 4: Connect communication cable to RS232 connector on PLC and to COM port on PC



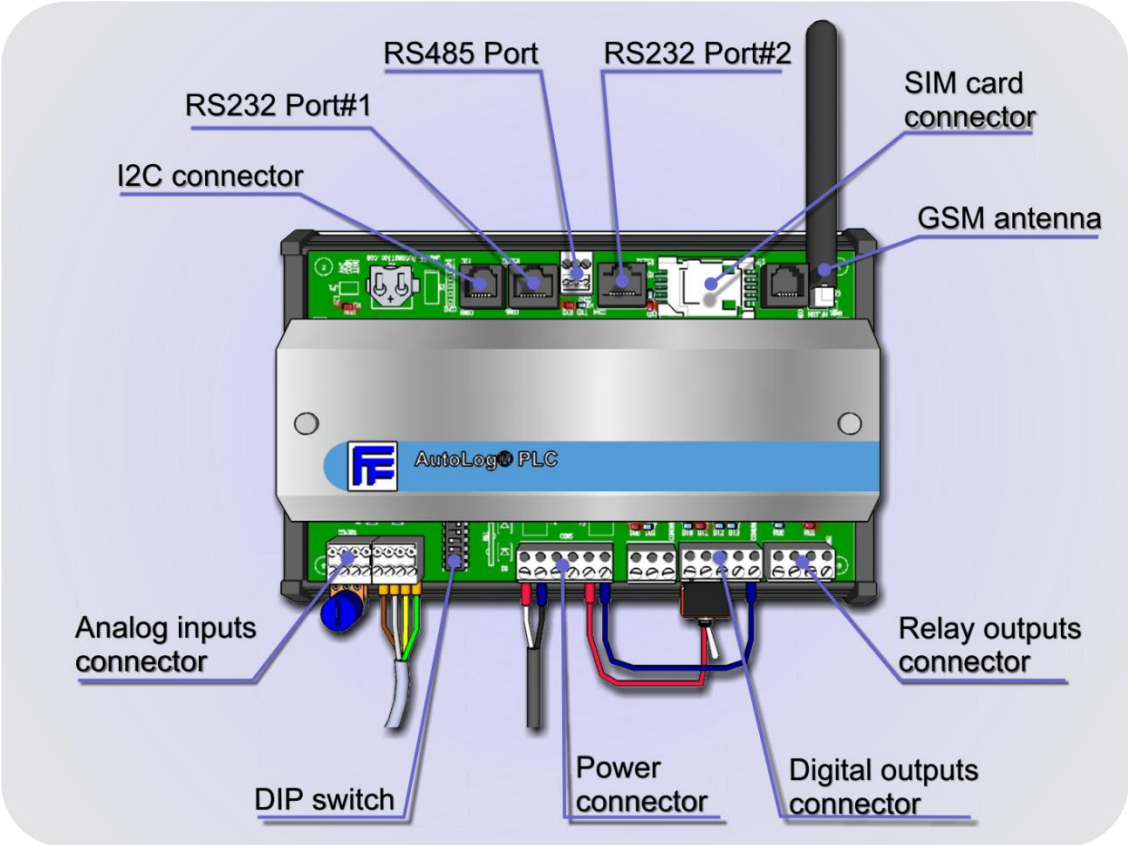
Pic 5: Connect power supply and run the CD software

More detailed instructions how to install the software is presented in next chapter.

2.3 GSM-8 PLC layout



Pic 6: AutoLog® GSM-8 PLC LEDs layout.



- **Pic 7: AutoLog® GSM-8 PLC connectors layout**
  RS 485 needs optional RS232 to RS485 conversion module (901157)

3 Getting started with GsmProgrammer software

3.1 Purpose of this chapter

This chapter gives instructions how to install the GsmProgrammer software and quick tutorial to make 1st test application program.

3.2 Installing GSM Programmer

3.2.1 Download latest version of GsmProgrammer

Latest version can be downloaded from the FF-Automation Web page's Distributor Area.

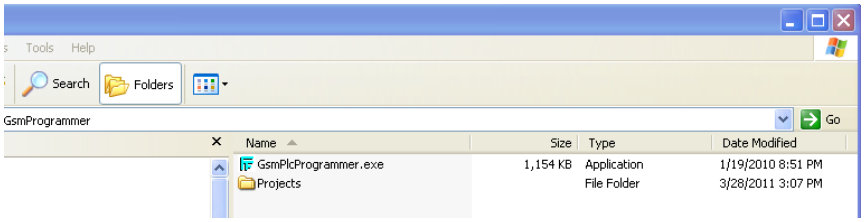
http://www.ff-automation.com/download/Distributor_signin.shtml

Please ask password from FF-Automation Oy!

E.g. to: antti.moijanen@ff-automation.com cc: info@ff-automation.com

Latest version of GsmProgrammer while writing this manual is v. 1.6.512.2

3.2.2 Installing the software



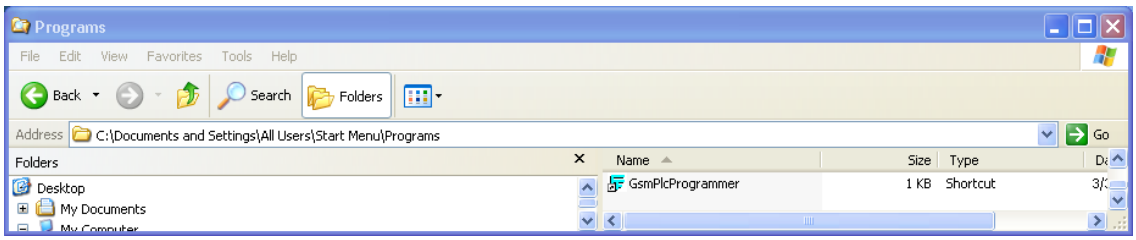
Create directory for GsmProgrammer (E.g. C:\GsmProgrammer).

Copy GsmProgrammer.exe into this directory.

Create new "Projects" folder under this directory.

3.2.3 Creating shortcut in Windows Start menu

Right click on “Programmer.exe” and select copy. (In Windows XP) Go to the folder C:\Documents and Settings\All Users\Start Menu\Programs and right click and select paste shortcut.



3.2.4 Creating shortcut in Windows desktop

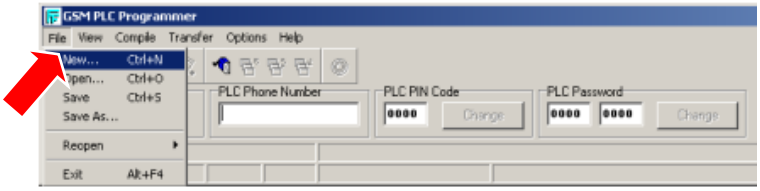
Right click on “Programmer.exe” and select copy. Go to desktop and right click and select paste shortcut.

Now you have installed GsmProgrammer!

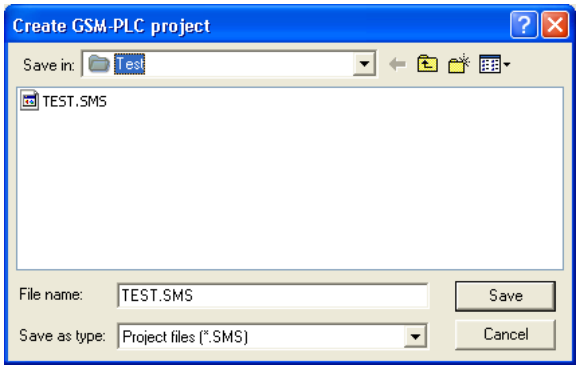
3.3 Creating new “Test” -Project

Double click “GsmProgrammer.exe” to start the program.

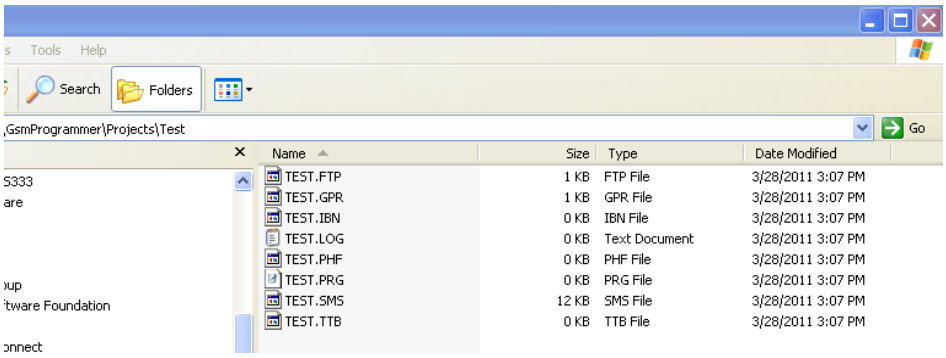
Create new project folder (e.g. Test) under C:\Program Files\GsmProgrammer\Projects\



Create new application program. File >> New:

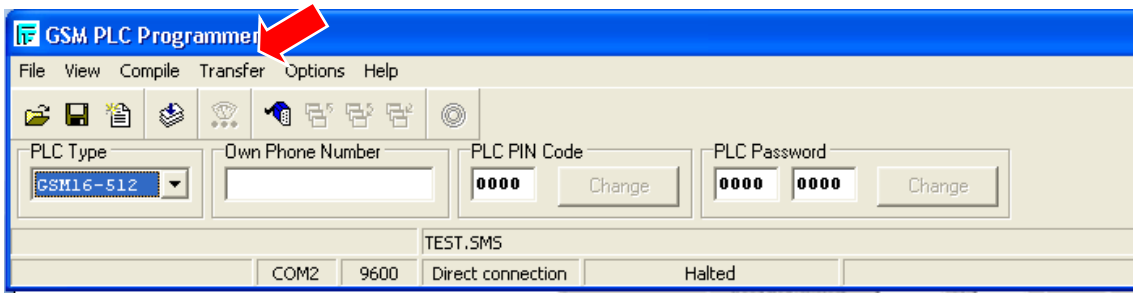


Write TEST.SMS and click Save.

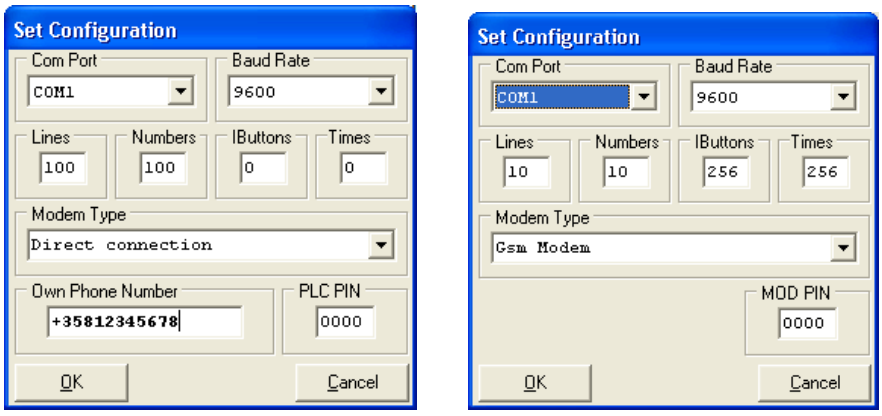


Now the Test folder should look like this.

3.3.1 Set the communication parameters



Define configuration settings: Transfer >> Config



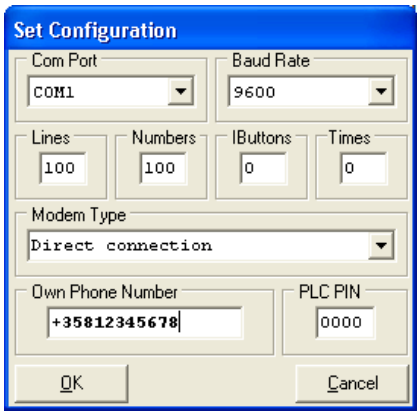
Connection Configurations for Direct connection and GSM modem connection.

- **Com Port:** PC's Com Port number (You can check PC's COM port numbers from Windows device manager. Start>>Run>>mmc devmgmt.msc).
- **Baud Rate:** (always 9600)
- **Lines:** Maximum lines is 512 for all new GSM-PLCs. This numbers defines how many programming lines you can have have in your project. Note! If you have GSM connection it is suggested to limit the programming lines to number of actually used (+ few extra) programming lines, because if you have here setting 512 and you request application program from GSM PLC, it will send all 512 lines although there are only e.g. 120 used programming lines. Normally you don't need to read project back from GSM-PLC through GSM network.
- **Numbers:** Maximum is 256 phone numbers. Look above note (in Lines row)! If you don't use phone numbers set this to zero (0).
- **iButtons:** Maximum is 256 iButton numbers. Look above note (in Lines row)! If you don't use iButtons set this to zero (0).
- **Times:** Maximum is 256 time tables. Look above note (in Lines row)! If you don't use time tables set this to zero (0).

- **Modem type**: Direct connection or GSM modem. GSM-PLC can be programmed using programming cable (Direct connection) or through GSM Network (Requires GSM modem type e.g. Wavecom).
- **Own Phone Number:** When using *Direct connection*, you should write some (any) number (e.g. +35812345678) here which simulates the GSM modem's phone number. If modem type is GSM modem, phone number is not needed, because GSM modem's phone number is used automatically.
- **PLC PIN: (SIM card is not needed for creating project)**
  Put SIM card to GSM phone and set SIM card's PIN code request OFF. (This depends on what GSM phone you are using. e.g. Search something like Settings->Security Settings->PIN code request. **It is important to set SIM card's PIN code request to OFF, otherwise GsmProgrammer might jam the SIM card to "PUK code mode".** When PIN code request is set OFF, you can leave PLC PIN to 0000 (or anything else)
- **If you wish to use PIN code then insert here the PIN code of the SIM card. If you enter wrong PIN code, GSM PLC tries two times to enter the PIN code. 3rd time will lock the SIM card and it can be restored only by feeding the PUK code.**
- **MOD PIN: If you have modem attached to your PC and you are using GsmProgrammer to monitor/program remote units, you need to insert here the PIN code of the SIM card that is in your modem.**

For testing select Baud rate: 9600, Lines e.g. 100, Modem:Direct connection, Own Phone number e.g. +35812345678.

3.3.2 Connect with the direct cable



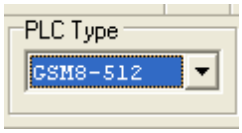
Change modem type to direct cable.

Check that DIP switch 4 = ON position. (Programming using direct connection through Ser1)

Note! If you change the DIP4 position you need to reboot the GSM-PLC in order to change the connection mode.

Note! It will take about 1-60 seconds for GsmProgrammer to connect with GSM-PLC.

3.3.3 Select PLC Type:



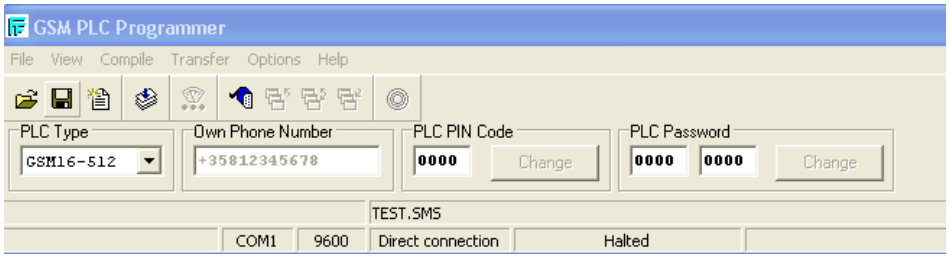
Select the type of PLC you are using. The code itself is compatible with all units but some features may change between PLC types, such as maximum program line amount & available phone numbers.

All new GSM-PLCs support 512 programming lines so you should select e.g. GSM8-512 if you are using new GSM-8 or GSM16-512 if you are using GSM-16 or GSM20 if you are using GSM-20.

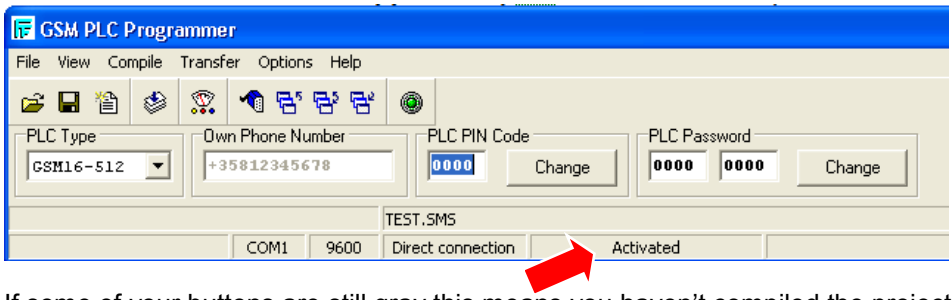
3.3.4 Checking the connection

Now connect the programming cable between PC's COM 1 (or any other serial port) and GSM-PLC's Serial port 1. GSM-PLC's Serial port 2 cannot be used for programming GSM-PLC!

GsmProgrammer establishes the connection quite slowly with GSM-PLC. You may need to wait up to 1 minute. When the connection is ready you can see the "Activated" text in the status bar. You can also see that the Rx and Tx LEDs are blinking faster (several times/ second) near the GSM-PLC's serial port 1.

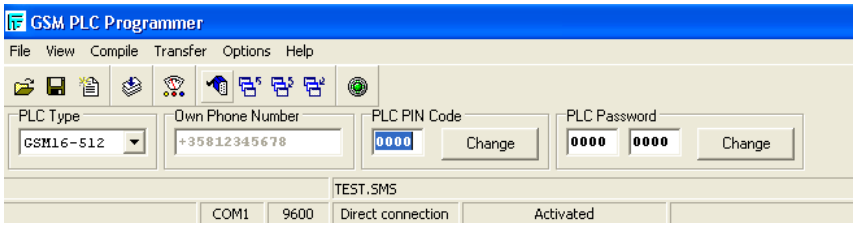


Connection to GSM-PLC is established when "Halted" status changes to "Activated"



If some of your buttons are still gray this means you haven't compiled the project yet. This doesn't matter, compilation can be done later.

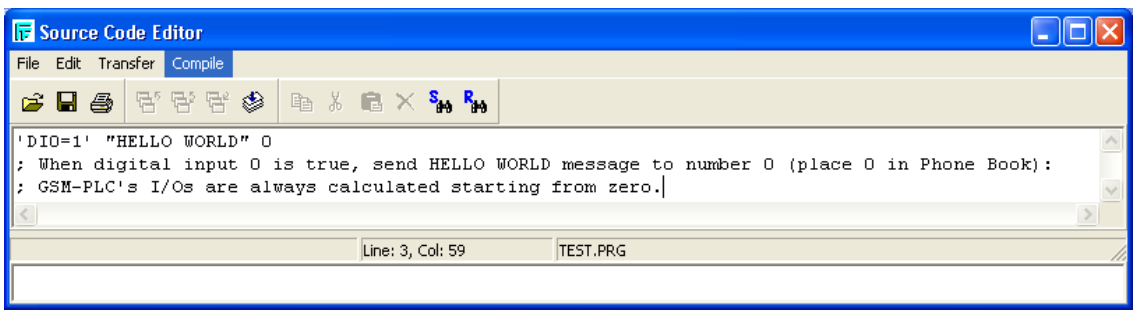
3.4 Creating first program code!



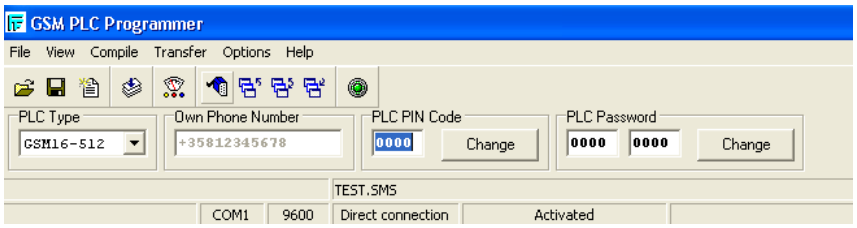
Open View >> Source Code

Program Editor opens. Now you can program your first application to this editor:

'DI0=1' "HELLO WORLD" 0
; When digital input 0 is true, program sends text message "HELLO WORLD" to number 0 (place 0 in Phone Book):



If you do not have phone number defined in Phone number list [0], message will not be sent.
You can add comments anywhere using the semicolon ";" character.

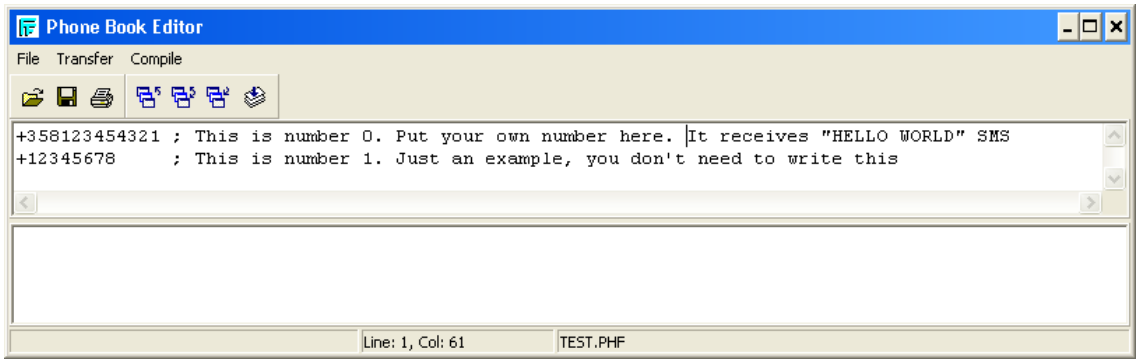


Select View >> Phone Book

Phone Book Editor opens:

- First row is telephone number 0 = place 0.
- Second row is telephone number 1 = place 1. Etc...

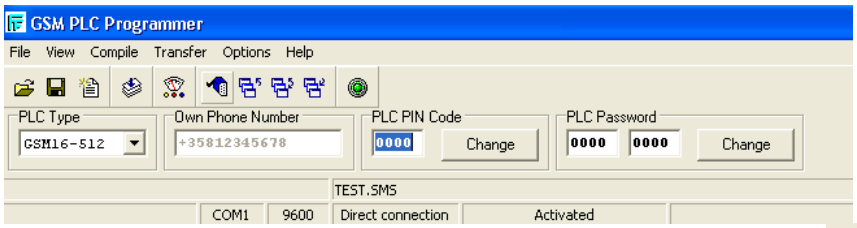
Put your mobile phone number to place 0.
You can put comments after ";" character in any line.



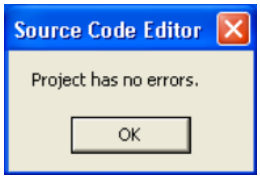
Use land code! e.g. +358 (=Finland). Do not use spaces between numbers!

3.5 Compiling the project

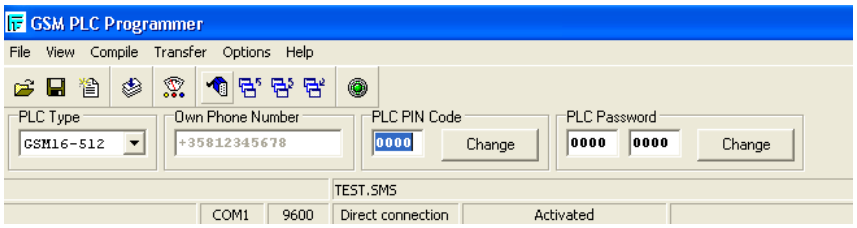
Save and close the source code and phone book windows.



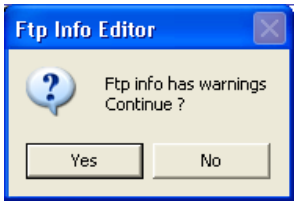
Select "Compile>>Compile" in main window or click Compile –button.



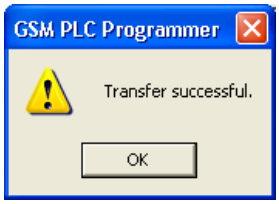
3.6 Transferring the project to GSM-PLC



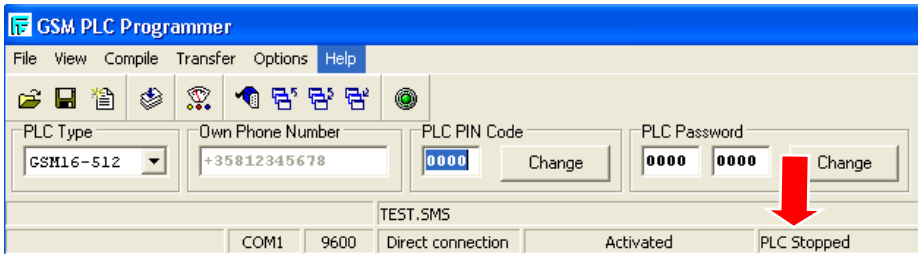
Transfer application program to GSM-PLC by selecting “Transfer >> Transfer Project” or by clicking Transfer –button. 



You will get the following warning message. You don’t need to care about this at this time. Click “OK”.



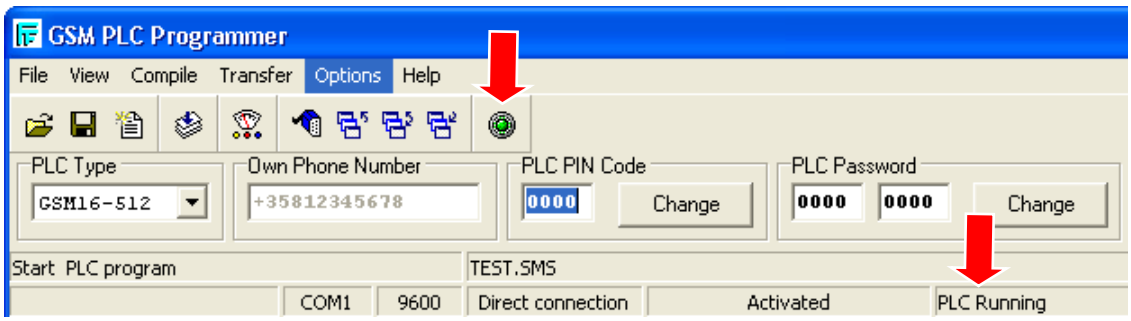
Press OK.



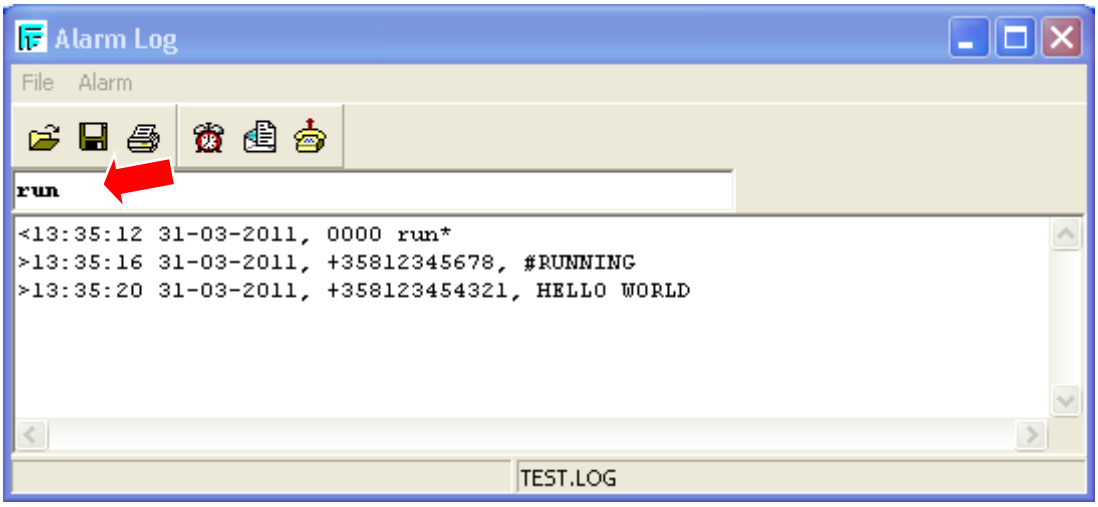
After transferring the project you can see text “PLC Stopped” in status line.

 **After transferring the project remember to start it!** (If you don’t click start button and you reboot the GSM-PLC, it will start the old program!)

You have two choices to start the project:



First: Press green “Start PLC” button. You will see “PLC Running” text in status bar.



Second: Open View >> Alarm Log >> write: RUN >> Press Enter.

Then you will receive the message: “+35812345678 #RUNNING” in Alarm Log window

Also “RUN” LED of GSM-PLC starts blinking.

You can test the “HELLO_WORLD” application with Alarm Log window:
Activate digital input 0, after that HELLO WORLD message should appear in Alarm Log window.

(AlarmLog Window can be used in many ways to test your application; these are describer later in this manual)

You can also use memory bits e.g. M0 instead of Digital input to trigger the SMS message.
‘M0=1’ “HELLO WORLD” 0

M0 can be set to 1 by writing “SET M0=1” in the Alarm log.

3.7 Testing project via GSM Network

You will need SIM card.

It is suggested to disable SIM card’s PIN code request. Put SIM card into your mobile phone and disable PIN code request. If you don’t disable PIN code request, the SIM card will go to PUK code state.

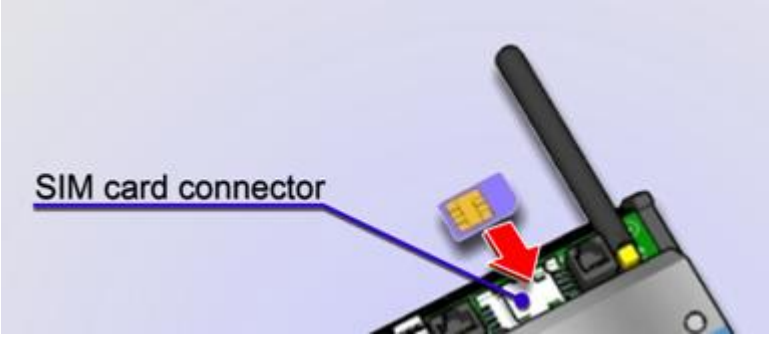
ALWAYS Test the application using cable connection and Alarm Log, before activating GSM modem.

Try to make SMS counter in your program which limits the maximum send SMS / day.

It is recommend to test application program using Alarm Log. Badly designed GSM-PLC program can start sending hundreds or even thousands of SMS messages and can be very expensive. Be sure to test the program before using GSM modem.

When DIP switch 4 = OFF, GSM modem is ON.

You will not receive any SMS if the GSM-PLC's DIP-4 switch is in "ON" position, which disables internal GSM modem. It is always suggested to test the application in this mode (DIP-4 = ON) first using AlarmLog window as user interface.



Switch power off and insert SIM card

DIP-4 should be in OFF position to activate internal GSM modem. Now you will lose the direct connection, because SER 1 is used by internal modem. There is no interface to the application other than your mobile now. (Except if you have HMI connected to I2C port or Modbus Master to SER2). Note! You can program GSM-PLC only through SER1.

Connect power on and check that GSM-PLC's SER1: "RX & TX" LEDs are blinking fast (~10Hz) and "RUN" led is blinking normally (~0.25Hz). It is normal that RUN led is ON about 2 seconds and then OFF about 2 seconds. In OFF position RUN led can have dimmed red light, which is normal. Check also that Modem "GSM" led is blinking normally about 0.5 ON and about 2 seconds OFF.

Now you can test the application by switching GSM-PLC's Digital Input 0 (DI0) to ON (=1) position. If you don't have digital input simulator, you can supply 24VDC between DO0 (IN0+) and COM "DO common ground". If you don't have power supply, you can connect Vout+ to IN0+ and Vout- to COM.

When you switch the DO0 in ON position Ser1's "Rx & Tx" led are turned OFF about 2 seconds, which means that SMS message is send to the GSM modem.

You should receive SMS message to your mobile phone, when the DI0 is turned to 1. It should take about 5-20 seconds.

EXPERTS ONLY:

 **If you decide to use PIN code, remember that "PLC PIN" set in GsmProgrammer must be the same as in SIM card's PIN. Otherwise SIM card will go to PUK mode when the GSM-PLC is powered and GsmProgrammer is connected. (Default PIN code setting in GsmProgrammer = 0000).**

3.8 I want to learn more! What to read next?

Congratulations! Now you have passed the "Getting started tutorial" part of this manual.

Next "Technical Reference" section is describing GsmProgrammer and GSM-PLC hardware in greater details. Start reading it! After that you can read the "Programming part". While you read, it is suggested to use GsmProgrammer to make own test applications.

In the end of this manual you can see information about the GSM-PLC models with detailed hardware pictures and key feature tables.

For SCADA interface user have two main options:

(1) AutoLog ControlMan

AutoLog ControlMan is Cloud-SCADA interface for AutoLog GSM-PLCs. It is used in hundreds of GSM-PLC projects all over the world. AutoLog ControlMan is designed by FF-Automation. It is suggested if you want to have cloud SCADA which can be used anywhere via Internet and you don't need own server PC. Server and application is maintained by FF-Automation.

(2) Indusoft Web Studio SCADA with AutoLog GSM-PLC driver

FF-Automation distributes Indusoft Web Studio SCADA and AutoLog GSM-PLC driver for Indusoft Web Studio SCADA. If you want to have own server PC, this is suggested solution.

Other options: AutoLog GSM-PLC driver can be used with any SCADA system which supports ODBC connection to PostGreSQL database.

TECHNICAL REFERENCE

4 AutoLog® GsmProgrammer description

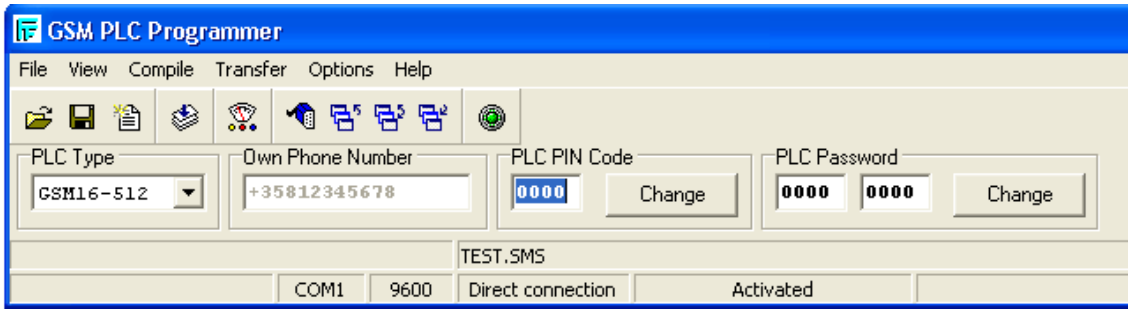
4.1 General description

AutoLog GsmProgrammer is Windows 95, 98, NT, 2000, XP, Vista and 7 compatible configuration program for AutoLog® GSM-PLCs. Configuration program is used to create, transfer and debug application programs for GSM-PLCs. GsmProgrammer allows all data in GSM-PLC to be monitored through the Alarm Log debug monitor. In the following sections we describe main features of GsmProgrammer.

4.2 Downloading and installing the software

Look the Getting Started part of this manual.

4.3 Main window



After GsmProgrammer is started, you will see the following view.

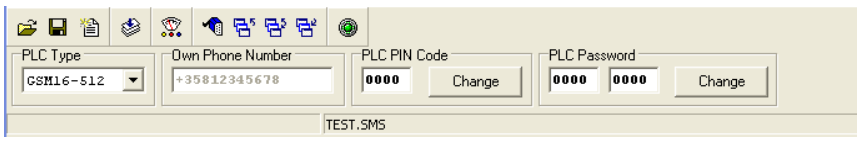
In case if most icons are inactive (gray), there is no connection to GSM-PLC and you will see “Halted” – text in the status line instead of “Activated”.

There are following main parts of GsmProgrammer’s “Main Window”

Menu Bar:

Toolbar:

Main section:



Status bar:

4.4 How to create new GSM-PLC project

To start new GSM-PLC project user can use command menu of GSM-PLC:

Select **File**, then **New**, or click New project –button 

After this a dialog box will open and user will be able to select the folder (e.g. C:\Program Files\GsmProgrammer\Projects\My_Project_Name), where the project data should be stored and which is the project name. After this the **Save**-button of the dialog box should be clicked.

Look Getting started part for more information how to create project!

4.5 Connecting GsmProgrammer with GSM-PLC

To establish a connection between PC and GSM PLC you can click **Config** button:

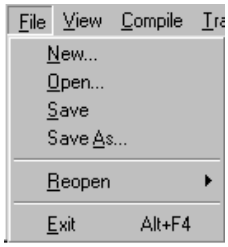
 Or in command menu select **Transfer**, then **Config,** to set correct configuration.

Look more information from the Getting Started part of this manual!

(Be sure that cable is connected to GSM PLC's Ser1 and DIP 4 is in ON position, connection establishment can take 1-2 minutes)

4.6 File

To access the "File" menu commands, click on "File" menu item.

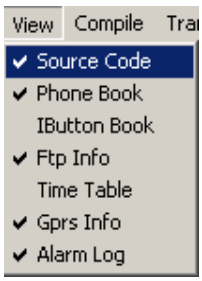


- "New" command is used to create the new GSM-PLC project.
- "Open" command is used to open the saved GSM-PLC project.
- "Save" command is used to save the GSM-PLC project.
- "Reopen" command is used to open one of the recently saved projects.

4.7 View

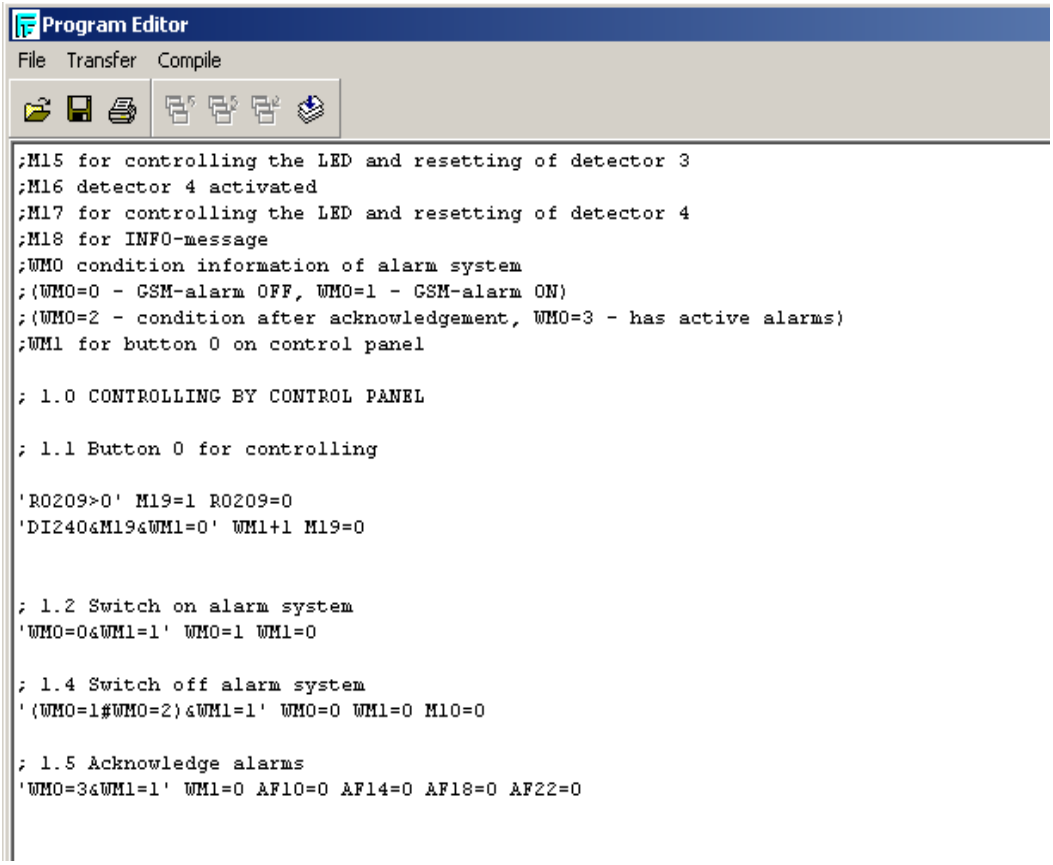
In the main window under the View menu you can open/activate editors.

- Selecting "Source Code" you can open/activate Program Editor window.
- Selecting "Alarm Log" you open/activate Alarm Log window (=Debugger).
- Phone book editor opens/activates by selecting "Phone Book".
- Also available: "IButton Book" Editor, "Ftp Info" Editor, "Time Table" Editor, "Gprs Info" Editor.

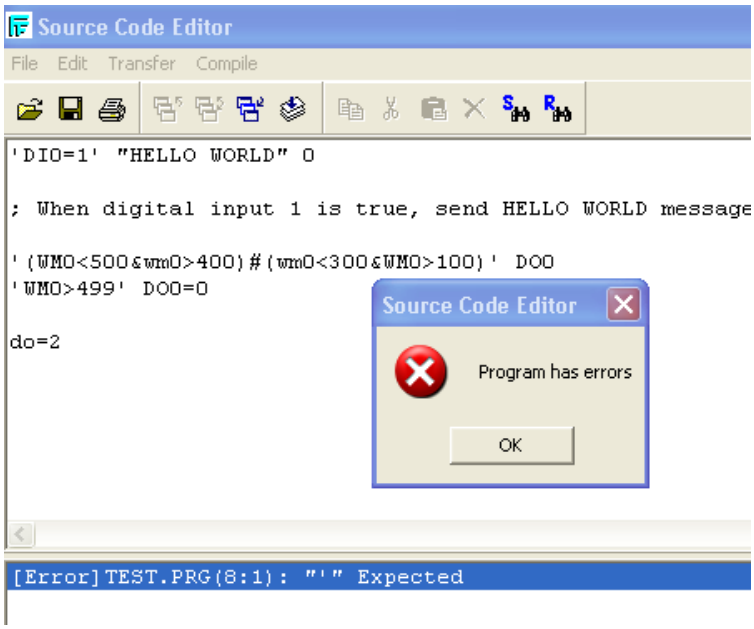


4.8 Program Editor

Program Editor window is used to write application programs. Maximum program size is 512 lines. You can write remarks with ";" character. These lines are not counted.



 Compile button. User can check if there are any errors in the source code by pressing the compile button.



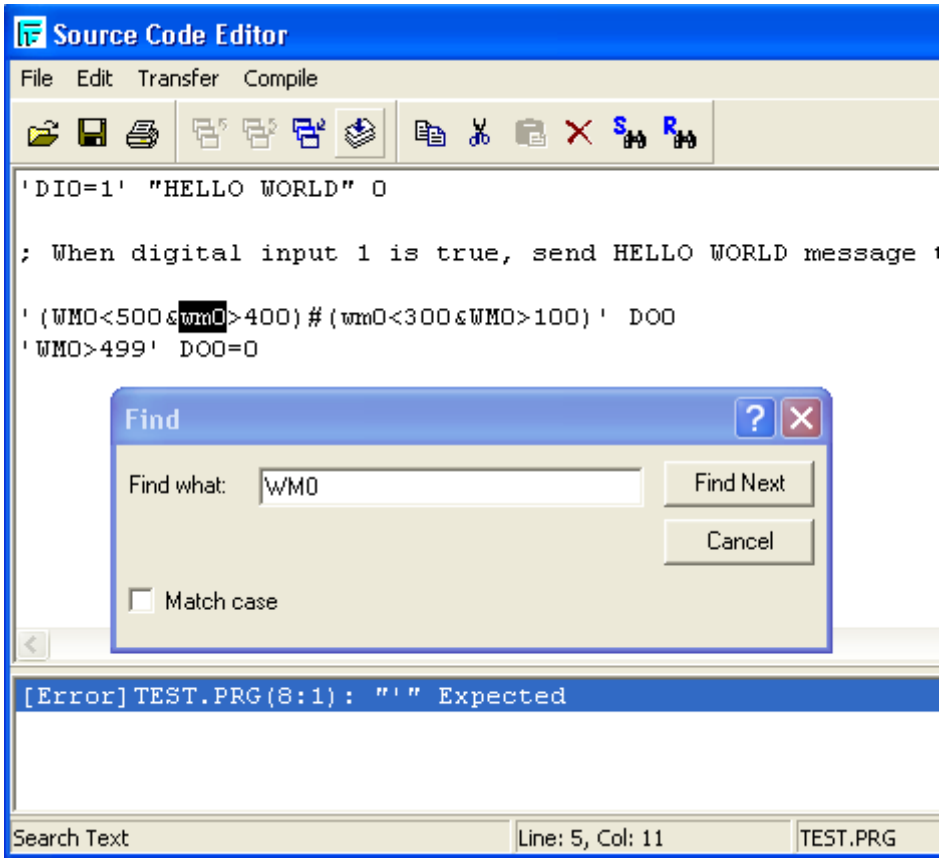
When there's a error, you will see [Error] message which will also inform you the programming line (8:1). (=Line 8). If you double click the error message, it will automatically jump to the appropriate error line.

 User can use these buttons for transfer, verify and read back program file (*.PRG).

Transfer Project downloads the application program to GSM-PLC.

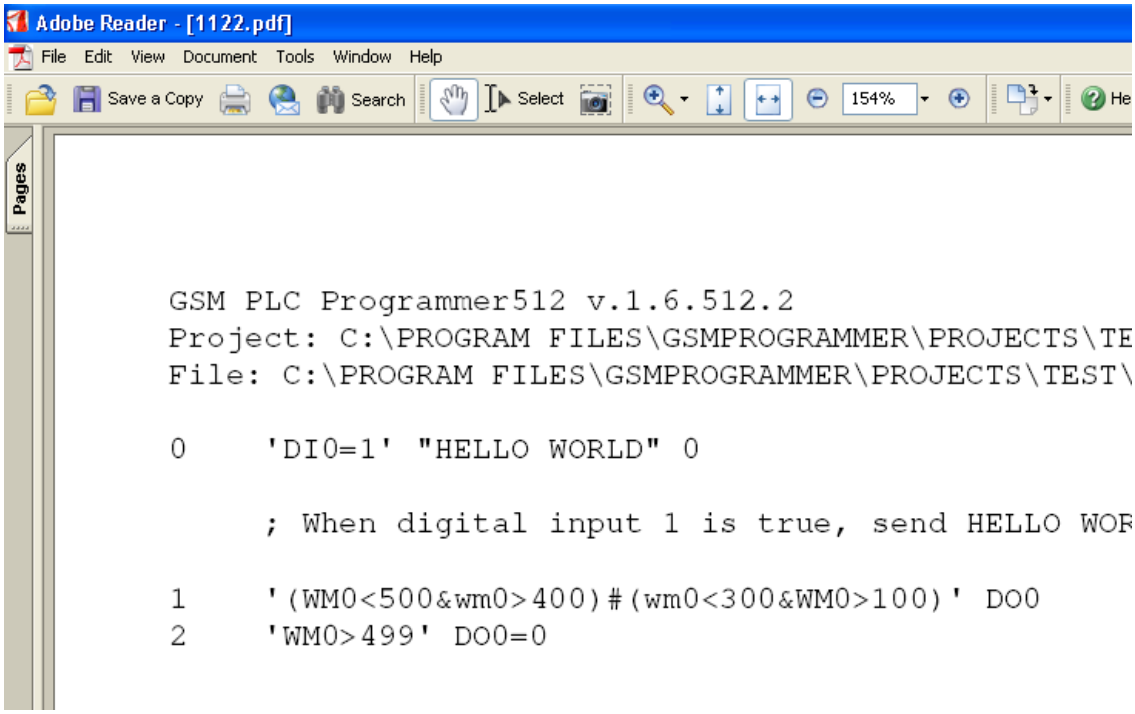
Verify Project uploads the application program from GSM-PLC and compares project to one that is active in the programmer. If it differs, you will get error message.

Read Back is used to upload application program from GSM-PLC to GsmProgrammer.
Note that comment lines are not stored in GSM-PLC, so you will lose the comment lines.



4.8.1 Limitations of program editor

Note! Only program lines in source code editor 0-255 can contain SMS message sending! Lines 256-511 cannot control SMS sending! (When you print the source code to e.g. using Adobe Acrobat PDF, you can see the programming line numbers.



When printing the source code to e.g. PDF file, you can see the programming line numbers.

You can download free PDF writer (e.g. CutePDF) http://www.cutepdf.com/

4.9 Alarm Log

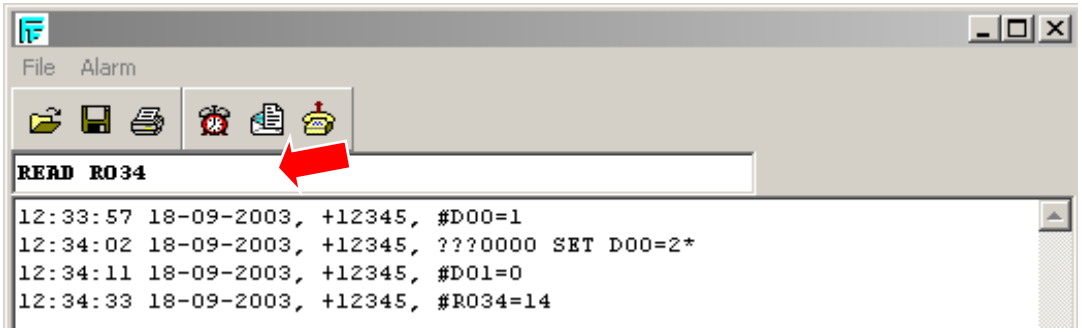
Alarm Log window is very important toot for testing and debugging the application program and setting program parameters etc.

Alarm Log window can be used e.g. to check SMS-messages that GSM-PLC sends out. In case there is a notation error in application program, line with errors can be seen in this window. Also debugging the program is easy to do using Alarm Log.

In Alarm Log there's an "input field" where you can simulate SMS messages or calls to GSM-PLC. While testing the application program you can e.g. set / read variable values in GSM-PLC through this window.

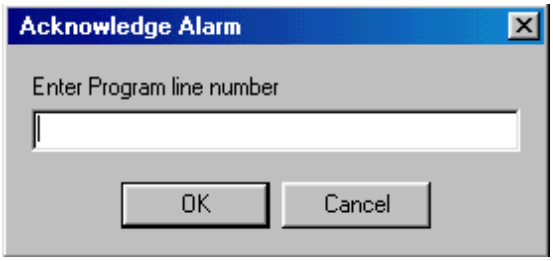
It is also possible to program the GSM-PLC using Alarm Log. E.g. if there is one program line you wish to add or change, you can send that line using Alarm Log instead of transferring the whole program. This feature is important if you download the program remotely through GSM network.

Note! Editing program stops the program execution. Remember to start new program by sending RUN command or pressing green RUN button in the main window.



If user wants to send command to GSM-PLC, user must manually enter message and press "Enter" or click on a Send **-**button.

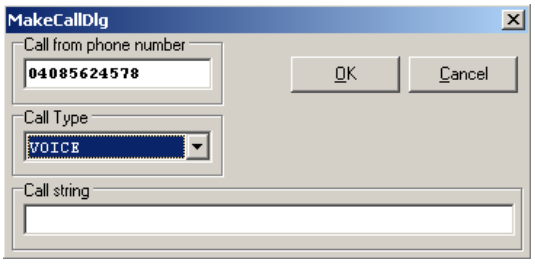
If user wants to send ACK command to GSM-PLC, user must click on Ack -button. Message box will be displayed:



User can add the program line, which has to be acknowledged and then press OK button.
(Each program line holds an Alarm Flag (AF), which can be set when triggering SMS out. This alarm flag can be used to send 2nd message, if no one reacts to 1st message. Acknowledging this alarm flag will close the resend sequence) This is special feature which is normally not needed.

User can open or save Alarm file (*.LOG) using these buttons.

User can simulate calls (voice, data, and fax) to GSM-PLC using this button. Call string field: you can ignore at his point



4.9.1 Examples of using Alarm Log:

Note! Alarm Log is very important for debugging the GSM-PLC program! Read this carefully!

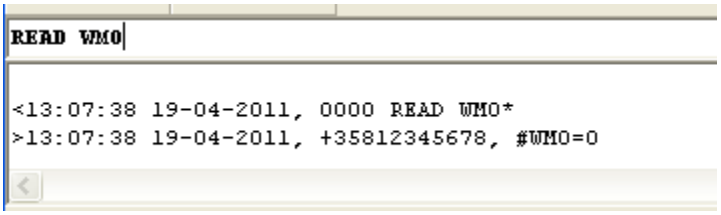
Note! All these commands can be used also with GSM phone, but the syntax is little bit different, you need to add password (normally 0000) in the beginning of the message and the asterisk (*) in the end.

Note! You can combine several commands in same command line in alarm log or SMS e.g.

READ WM4*0000 READ WM5*0000 READ WM6

SMS: 0000 READ WM4*0000 READ WM5*0000 READ WM6*

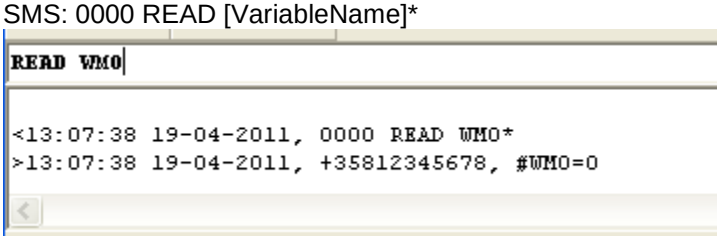
Note! Alarm log command is the same as SMS command, it just automatically adds the password in the beginning and the asterisk (*) in the end of the command line.



Direction of the message can be seen from the < and > -marks.
< = to GSM-PLC
> = from GSM-PLC

READ [VariableName] ;Reads variable value

SMS: 0000 READ [VariableName]*



In Alarm log window you will see the commands and the responses in SMS format.

SET [VariableName]=[Value] ;Sets variable value

SMS: 0000 SET [VariableName]=[Value]*

```
SET WM0=100

<13:07:38 19-04-2011, 0000 READ WM0*
>13:07:38 19-04-2011, +35812345678, #WM0=0
<13:08:30 19-04-2011, 0000 SET WM0=100*
>13:08:30 19-04-2011, +35812345678, #WM0=100
```

VIEW ALL? **;Shows all programming lines in GSM-PLC**

SMS: 0000 VIEW ALL?*

```
0000 VIEW ALL?*
+35812345678, #LINE 00: 'DI0=1' "HELLO WORLD" 0#LINE 01:
'(WM0<500&wm0>400)#(wm0<300&WM0>100)' DO0#LINE 02: 'WM0>499'
DO0=0#LINE 05: 'DI0' DO2
```

VIEW [line number] **;Shows only the defined line**.

SMS: 0000 VIEW [line number]*

This command is useful for verifying that the programming line before you replace the program line to GSM-PLC remotely through GSM network.

NUM ALL? **;Returns all phone book numbers in GSM-PLC**

SMS: 0000 NUM ALL?*

```
0000 NUM ALL?*
+35812345678, #NUM 0=+358123454321 #NUM 1=+12345
```

NUM [line_number_in_phone_book]=[Phone_number] ; Sets the phone number in GSM-PLC's phone book

SMS: 0000 NUM [line_number_in_phone_book]=[Phone_number]*

Note that this will stop the PLC program so you need to click RUN –button or send RUN command to start the PLC program.

Note also that this will not change the phone number in phone book editor, if you send it by SMS.

```
e.g. NUM 1=+1234567
```

RUN **;Starts GSM-PLC program**

SMS: 0000 RUN*

```
RUN

<09:52:32 17-04-2012, 0000 RUN*
>09:52:33 17-04-2012, 1234, #RUNNING
```

INIT [Programming line number] [New programming line] ;Changes program line

`e.g. INIT 5 'DI2' M2` ;Changes (overwrites) programming line number 5 with programming line `'DI2' M2`

SMS: 0000 INIT [Programming line number] [New programming line]*

This command is useful if you want to replace one program line to GSM-PLC remotely through GSM network.

Use this carefully, because it will not change the source code.

Note this command stops the GSM-PLC program. You will need to start the application program after this command.

You can use this command combined with RUN command:

e.g. `INIT 5 'DI2' M2* 0000 RUN`

or with SMS: `0000 INIT 5 'DI2' M2* 0000 RUN*`

READ AF[Line_number] ; Reads Alarm flag state in programming line (Experts only)

SMS: 0000 READ AF[Line_number]*

You can combine several commands in Alarm log and SMS e.g. Commands are executed from the left to right.

read af0*0000 read af1*0000 read af2 ;Reads alarm flag 1 ,2 & 3

SMS: 0000 read af0*0000 read af1*0000 read af2*

```
0000 read af0*0000 read af1*0000 read af2*
+35812345678, #AF0=0#AF1=0#AF2=1
```

ACK [programming_line_number] ;Acknowledges alarm flag (Experts only)

SMS: 0000 ACK [programming_line_number]*

e.g. ACK 2 ;Acknowledge alarm flag 2 (AF2)

SMS: 0000 ACK 2*

(Note! If there are unacknowledged alarms in GSM-PLC, the RUN –led will blink 4 times fast and then off for about 2 seconds)

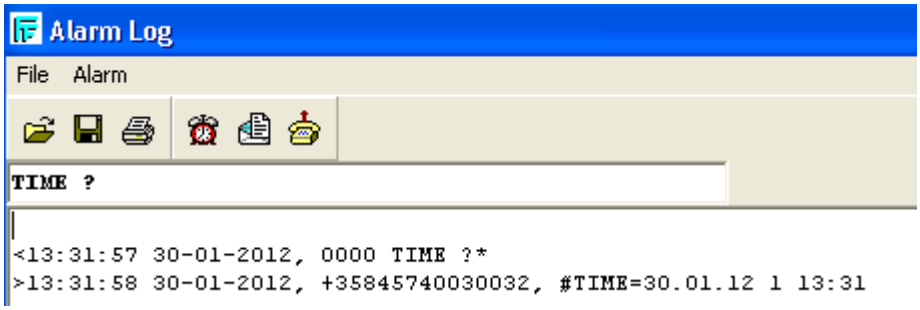
Every programming line has Alarm flag, AF2 means alarm flag in programming line 2.

Alarm flags are quite difficult to use and these are for expert purposes only. Normally these are not needed.

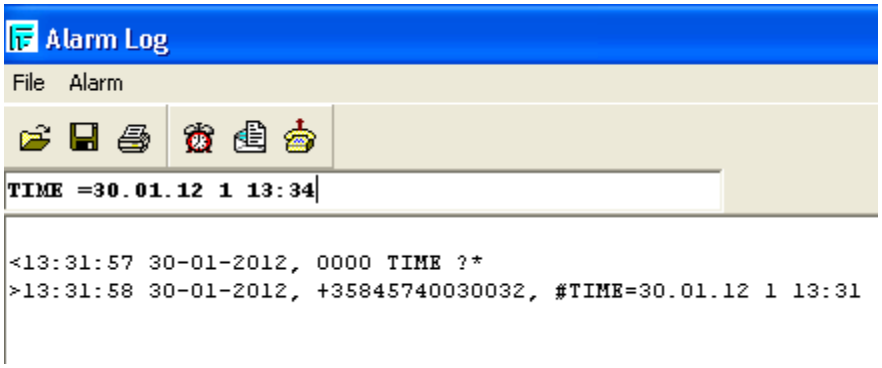
ACK ALL ;Acknowledges all alarm flags
(Experts only)

SMS: 0000 ACK ALL*

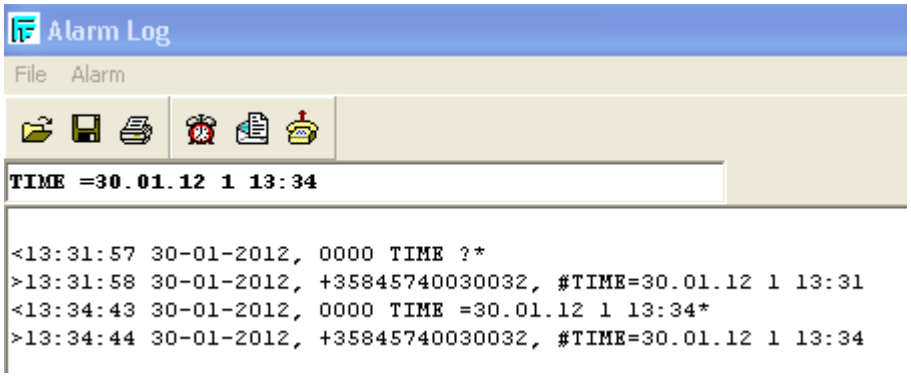
4.9.2 Setting the PLC clock with Alarm Log



TIME ? ;returns the date and time, you can copy the time format for next command.



Paste the time format and add one space between TIME and "=", and change the time.



4.9.3 Viewing SMS messages in the Alarm Log

From the Alarm Log you can see all the messages, which are send from GSM-PLC (if you have programming cable and connection between GsmProgrammer and GSM-PLC).

The messages send from GSM-PLC can be seen not only in the direct connection mode (DIP4=ON) but even if the GSM-PLC is in the GSM modem mode (DIP4=OFF).

```
>09:54:33 17-04-2012, +35850123452, D9,,1,1,1,20,9,0,0,0,1236,0,17
```

SMS messages format:

time stamp, destination phone number, message

4.9.4 FTP files in Alarm Log

If you are directly connected to GSM-PLC and PLC sends FTP files, these are store in your current project directory. Notification of receiving file is added to Alarm Log in following format:

21:15:35 17-07-2003, Received file, FFFFFFFF10307141804, 542 bytes

---

FTP ? ;Returns FTP configurations

SMS: 0000 FTP ?*

```
0000 FTP ?*
+35812345678, #FTP =1,222.118.20.68,/,FFFFFFFF,demopw
```

FILE [FTP_file_number]? ;Returns FTP file, max. size, current size and stored variables.

SMS: 0000 FILE [FTP_file_number]?*

e.g. FILE 0? ;Returns FTP file 0, max. size, current size and stored variables.

SMS: 0000 FILE 0?*

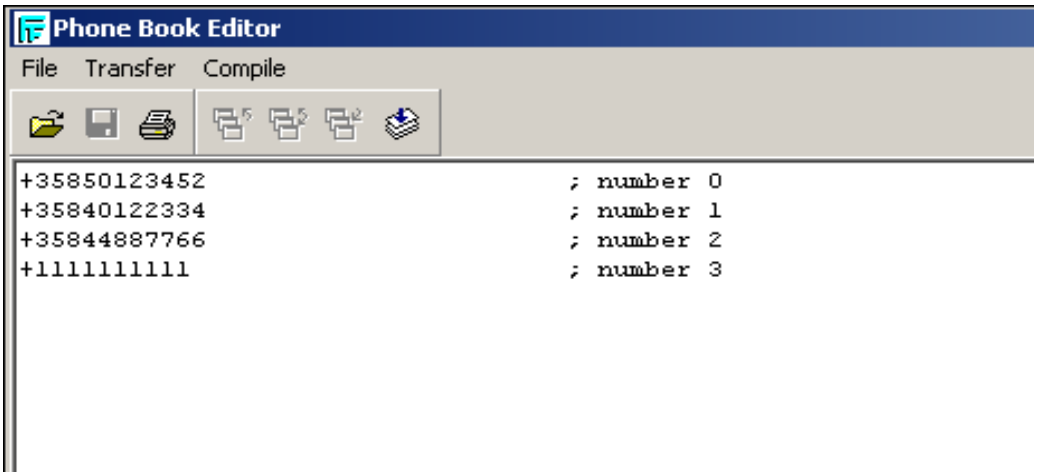
```
0000 file 0?*
+35812345678, #FILE 0=262144 80 CN0 AI0 AI1 AI2 AI3 DI0
```

File 0= ;This deletes the FTP file 0

SMS: 0000 FILE 0=*

4.10 Phone Book Editor

Phone Book Editor is used to define phone numbers to where GSM-PLC will send messages (Alarms, reports) or calls. There can be max. 245 phone numbers. In application program these phone numbers are **referred using line number.**



It is suggested to use country codes in the phone number (e.g. +358 in Finland)

Do not use spaces!

 User can use these buttons for transfer, verify and read back phone book file (*.PHF).

4.10.1 PROGRAMMING EXAMPLE

['Condition'] [Commands] ["Message"] [Phone_number_id]

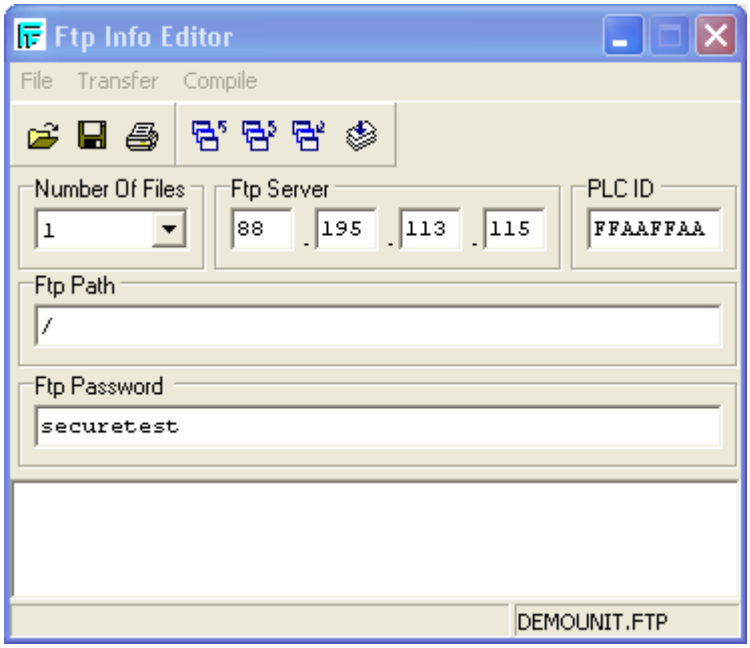
```
'M110=1&WM60<10' WM60+1 "D9,,1,%WM99,%WM11,%WM12,%WM13,%WM24,%WM22" 0

; When condition becomes true the message is send to phone number 0 =
+35850123452
```

Note! It is suggested to make some max limitation for sending SMS messages. e.g. in the example line the limitation is made by using WM60<10 in the condition and WM60+1 in the command.

Note! Always test the program in the alarm log before using it in the GSM modem mode.

4.11 Ftp Info Editor

(if you are using SMS messages only, you don't need this)



FTP Configuration settings:

Number Of Files None/1/2/4/8 (None disables FTP supporting in PLC)

Ftp Server Server IP address in dot form (setup your own FTP server or consult FF-Automation to setup FTP server (e.g. ControlMan has FTP server)

If you want to use your own FTP server, you can use e.g. FileZilla (free FTP server), Note that in order to use your own FTP server, you need to have static IP address, which is either public or you have routed FTP port 21 (default) to your FTP server PC

PLC ID (=login name) 8 HEX symbols (Hex sysmbols are 0,1,2,3…A,B,C,D,F

Ftp Path Path on server to store files (128 characters max)
Root directory is defined in FTP server, so "\" will place files in root dir.

Ftp Password Password for Ftp login (98 characters max)
If password doesn't exists "anonymous" login will be used

If you are directly connected to GSM-PLC, when PLC sends FTP files, these are store in your current project directory. Notification of receiving file is added to Alarm Log in following format:

21:15:35 17-07-2003, Received file, FFFFFFFF10307141804, 542 bytes

FFFFFFFF10307141804 File name
542 bytes File length

You can test FTP file sending by putting the following line to your Test program.

```
'DI0=1' "%CN0,%AI0@3,%DI0" 245
```

;This line will add values of Counter0, AI0,AI1,AI2,AI3 and DI0 to FTP file0 when DI0 changes to 1.

Compile the application and Transfer and Click RUN button. Generate some savings by changing the DI0 few times.

You can check the ftp 0 file using Alarm log

`file 0?`

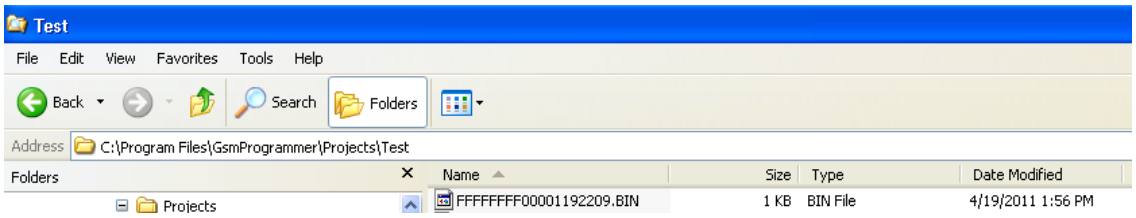
```
+35812345678, #FILE 0=262144 80 CN0 AI0 AI1 AI2 AI3 DI0
```

Set the RO38=0, FTP file0, do not delete after sending, Use Alarm Log

Set RO37=1, sends the FTP file.

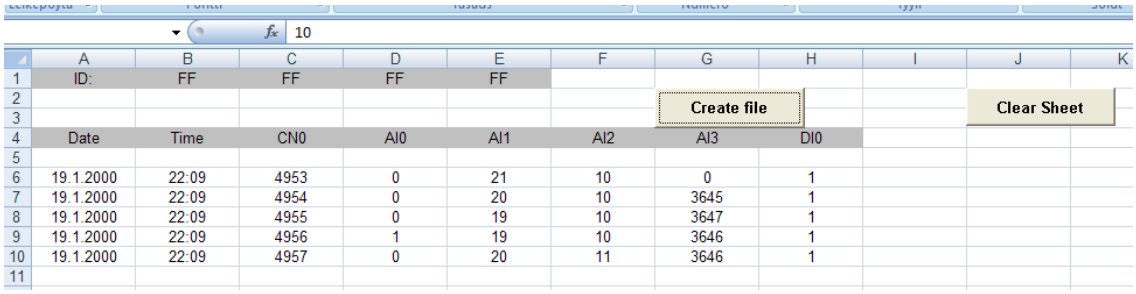
```
+35812345678, #RO37=1
Received file, FFFFFFFF00001192209.BIN, 98 bytes
```

Browse the FTP file from your project directory.



Ftp files can be opened with FileModifcation_FTP.xls. Please ask this application if it was not delivered to you with the GsmProgrammer.

Enable macros. Clear sheets and select the ftp0 file.



	A	B	C	D	E	F	G	H	I	J	K
1	ID:	FF	FF	FF	FF						
2											
3											
4	Date	Time	CN0	AI0	AI1	AI2	AI3	DI0			
5											
6	19.1.2000	22:09	4953	0	21	10	0	1			
7	19.1.2000	22:09	4954	0	20	10	3645	1			
8	19.1.2000	22:09	4955	0	19	10	3647	1			
9	19.1.2000	22:09	4956	1	19	10	3646	1			
10	19.1.2000	22:09	4957	0	20	11	3646	1			
11											

Picture from FileModification_FTP.xls program.

4.12 IButton Book Editor

(Experts only)

iButton gives a method of authentication for GSM-PLC. It is used only in special applications where authentication is needed e.g to open door or other access control.

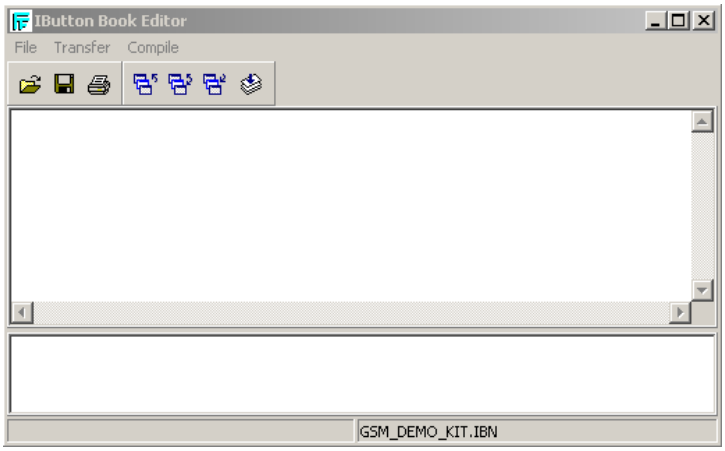
Amount of IButton Codes can be maximum 256.

IButton presentation format:

14 HEX symbols (MUST BE IN UPPER CASE!)

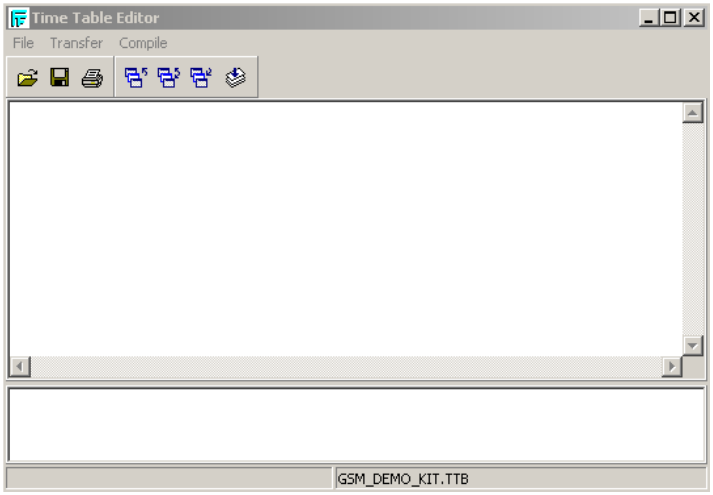
SSSSSSSSSSSFF

S -- Serial Number
F – Family Code
IButton can contain wildcards (“?”).



4.13 Time Table Editor

(Experts only)

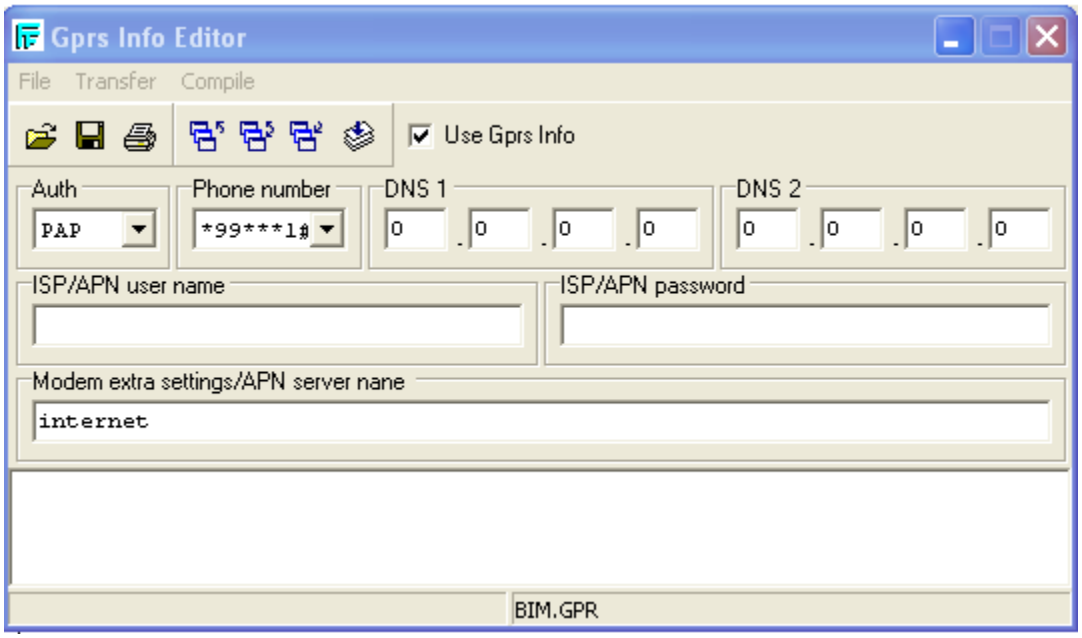


Using Time table editor is not suggested. Preferred method is to make “time table” in programming lines.

4.14 Gprs Info Editor

(if you are using SMS messages only, you don't need this)

Needed settings to enable GPRS connection:



Authentication method is PAP / CHAP. Default is PAP

Phone number "prefix" is *99***1# / *99# (Default is *99***1#)

APN username, APN password and APN username depends on your local GSM/GPRS operator:

DNS 1 & DNS 2 are normally left empty (These come automatically from your SIM card.)

Modem extra setting / APN server name

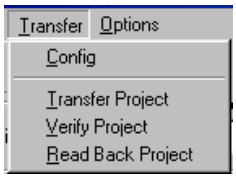
Examples:

Finland, Sonera/Elisa/DNA:	internet
Finland, Saunalahti:	saunalahti.internet
Sweden, Telia:	online.telia.se

Request APN settings from your local GSM operator or look it up from internet

4.15 Transfer

In the main window under Transfer menu there are three file transfer options. These are Transfer Project, Verify Project and Read Back Project. Also “Program Editor” - and “Phone Book Editor” windows have own Transfer menus but it transfers only either part of the project.

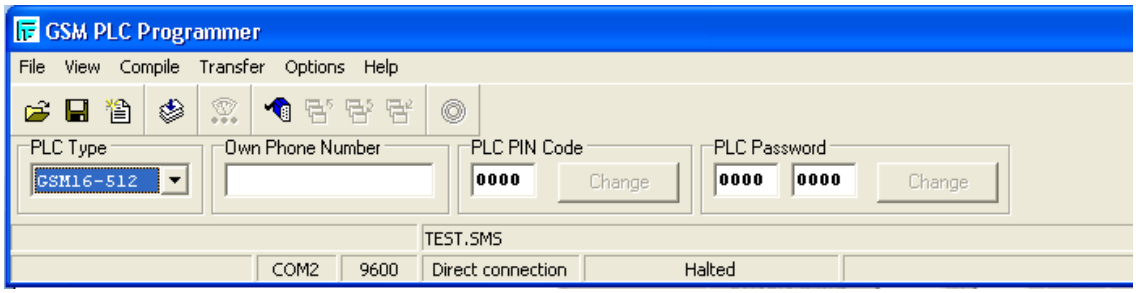


Transfer Project downloads the whole project (application program and phone book etc.) to GSM-PLC.

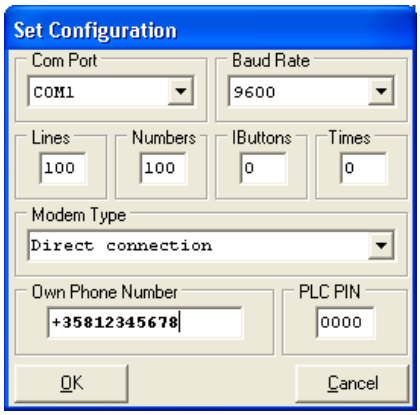
Verify Project uploads the project from GSM-PLC and compares project to one that is active in the programmer.

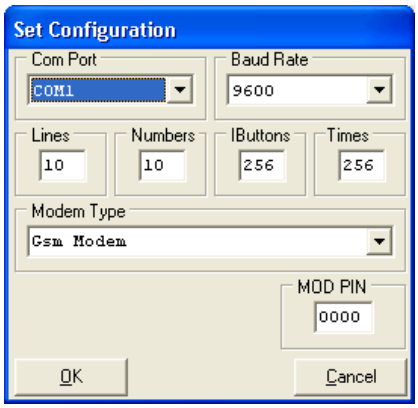
Read Back is used to upload project from GSM-PLC to GsmProgrammer.
Program Editor- and Phone Book Editor window’s ”transfer” options can be used when you need to transfer only either part of the project.

4.16 Configuration window



Define configuration settings: Transfer >> Config





Connection Configurations for Direct connection and GSM modem connection.

- **Com Port:** PC’s Com Port number (You can check PC’s COM port numbers from Windows device manager. Start>>Run>>mmc devmgmt.msc).
- **Baud Rate:** (always 9600)
- **Lines:** Maximum lines is 512 for all new GSM-PLCs. This numbers defines how many programming lines you have in your project. Note! If you have GSM connection it is suggested to limit the programming lines to number of actually used programming lines, because if you have here setting 512 and you request application program from GSM PLC, it will send all 512 lines although there are only e.g. 120 used programming lines.

- **Numbers:** Maximum is 256 phone numbers. Look above note (in Lines row)! If you don't use phone numbers set this to zero (0).
- **iButtons:** Maximum is 256 iButton numbers. Look above note (in Lines row)! If you don't use iButtons set this to zero (0).
- **Times:** Maximum is 256 time tables. Look above note (in Lines row)! If you don't use time tables set this to zero (0).
- **Modem type**: Direct connection or GSM modem. GSM-PLC can be programmed using programming cable (Direct connection) or through GSM Network (Requires GSM modem type WaveCom).
- **Own Phone Number:** When using *Direct connection*, you should write some (any) number (e.g. +35812345678) here which simulates the GSM modem's phone number. If modem type is GSM modem, phone number is not needed, because GSM modem's phone number is used automatically.
- **PLC PIN: (SIM card is not needed for creating project)**
  Put SIM card to GSM phone and set SIM card's PIN code request OFF. (This depends on what GSM phone you are using. e.g. Search something like Settings->Security Settings->PIN code request. **It is important to set SIM card's PIN code request to OFF, otherwise GsmProgrammer might jam the SIM card to "PUK code mode".** When PIN code request is set OFF, you can leave PLC PIN to 0000 (or anything else)
- **If you wish to use PIN code then insert here the PIN code of the SIM card. If you enter wrong PIN code, GSM PLC tries two times to enter the PIN code. 3rd time will lock the SIM card and it can be restored only by feeding the PUK code.**
- **MOD PIN: If you have modem attached to your PC and you are using GsmProgrammer to monitor/program remote units, you need to insert here the PIN code of the SIM card that is in your modem.**

4.17 Others


Button to Open GSM-PLC Project.


Button for Save GSM-PLC Project.


Button for Create New GSM-PLC Project.


Button to Compile GSM-PLC Project.


Button to Set Configurations.


Transfer Project, Verify Project, Read Back Project


RUN button to start GSM-PLC application.

4.18 PLC Type support:

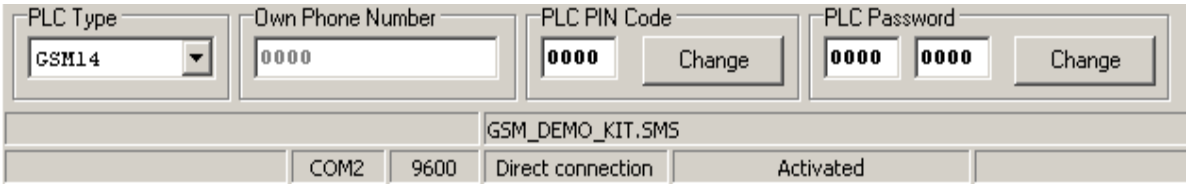
Select the type of PLC you are using. The code it self is compatible with all units but some features may change between PLC types, such as maximum program line amount & available phone numbers.

All new GSM-PLCs support 512 programming lines so you shuld select e.g. GSM8-512 if you are using new GSM-8 or GSM16-512 if you are using GSM-16

4.19 Own phone number

OWN PHONE NUMBER informs user about the current phone number. You can define here freely just some phone number, e.g. +12345678, it doesn't need to be the same as SIM card's phone number. It is just for testing.

4.20 PIN-code and password

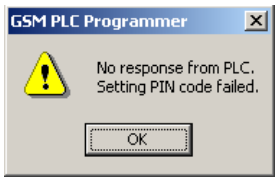


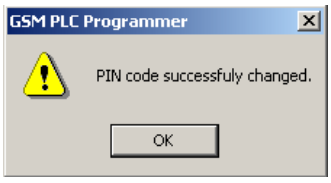
GSM-PLC PIN CODE allows user to edit PIN Code and change it in GSM-PLC after "Change" button is pressed.

It is suggested to turn off the PIN code request from the SIM card.

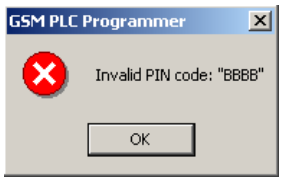
- Put SIM card to GSM phone and set SIM card's PIN code request OFF. (This depends on what GSM phone you are using. e.g. Search something like Settings->Security Settings->PIN code request. **It is important to set SIM card's PIN code request to OFF, otherwise GsmProgrammer might jam the SIM card to "PUK code mode".** When PIN code request is set OFF, you can leave PLC PIN to 0000 (or anything else)
- **If you wish to use PIN code then insert here the same PIN code which is configured in the SIM card. If you enter wrong PIN code, GSM PLC tries two times to enter the PIN code. 3rd time will lock the SIM card and it can be restored only by feeding the PUK code.**

There are 2 message boxes which may be displayed after GSM-PLC PIN CODE is changed:





If PIN Code you enter is not valid, next message box will be displayed:



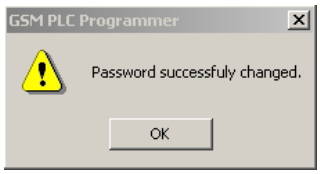
Note! PLC PIN code doesn't change the SMI card's PIN code. It can be used for extra security, but in most cases it is not needed.

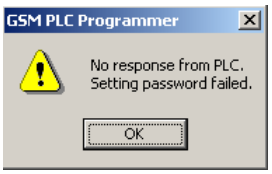
GSM-PLC PASSWORD allows user to edit Password code and change it in GSM-PLC after "Change"-button is pressed. Default PSW code is 0000.

To change GSM-PLC Password user must change digits only in right field.

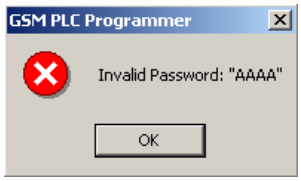
If you decide to use other than default "0000" password, remember to document this number!!! It cannot be recovered.

There are 2 message boxes may be displayed after changing the password:

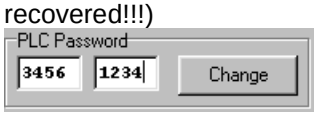




If Password code you enter is not valid, next message box will be displayed:



Left field contains the current PLC Password: (Password is stored in project file ,but cannot be recovered!!!)



After “Change”-button is pressed and when GSM-PLC successfully changed password, will be displayed next:

If you close the project and open it again. You need to remember the password and insert it in the left field to be able to modify the project.

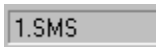
COM STATUS INFORMATION informs user about COM number, baud rate and state of communication.

If GSM-PLC or modem is not connected to PC, next status will be displayed (halted):

In the process of initialization, the status will be displayed as:

After completion of initialization the status will be displayed as:

FILE NAME informs user about the name of a current project file.



5 General Features of AutoLog GSM-PLC

5.1 GSM-PLC models and accessories:

Look GSM-PLC hardware –chapter

5.2 List of Main Features:

5.2.1 I/O:

- Analog inputs. Quantity depends on model
- Analog Outputs. Quantity depends on model
- Digital inputs. Quantity depends on model
- Digital outputs. Quantity depends on model
- Serial port 1. RS-232. Used for programming (DIP 4= ON) or by GSM modem (DIP 4=OFF).
- Serial port 2. RS232 / Optionally RS-485 / Optionally Modbus TCP Ethernet
- I2C port. For AutoLog HMI or iButton
- 2nd I2C port. Only in GSM-20. For connecting external AutoLog I/O cards.
- AU. GSM modem's Audio In/Out. (This is new feature which is not tested yet.)
- Power input (DC) for 12-30VDC operating voltage
- Power input (AC) for 12-24VAC operating voltage
- Power Output (DC) for supplying external devices or loading e.g. backup batteries (UPS). Level is same as supply voltage.
- GSM Antenna. Selectable models.

5.2.2 Other Hardware related:

- Backup lithium battery for retaining values in RAM (EEPROM) memory. Type: CR2032. 3 Volts.
    - Operating time about 5-10 years when GSM-PLC is powered.
    - Operating time drops to 6 months if PLC power is turned off.
- Real time clock and calendar. YY/MM/DD hh:mm:ss.
- place for GSM SIM card
- 6 DIP switches (DIP1-6)
- 3 Jumpers (JP1:1-2, JP1:3-4,JP1:5-6)
- Size and weight, look appendix!
- Enclosure: (Default) DIN rail mountable module (IP21), Optionally: ABS / polycarbonate plastic enclosures (IP65-67) or Stainless steel / aluminium enclosures.
- Optionally FF-Automation can supply enclosure with relays, wirings, sensors, fuses, switches and wirings installed and ready-to-use.
- As a hardware manufacturer, FF-Automation can design special features for special applications like integrated OEM (Original Equipment Manufacturer) solutions.
- AutoLog HMIs. Many models. Look Appendix!
- FF-Automation can supply also energy meters, which can be connected to Ser 2 and communicate using Modbus protocol.

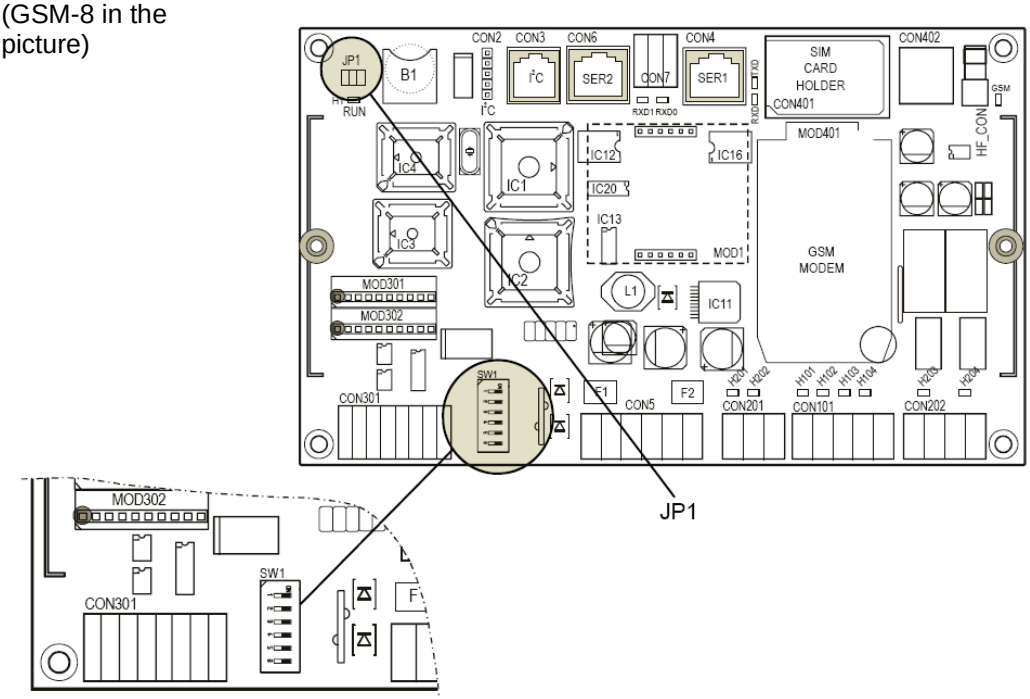
5.2.3 Software related features:

- **Wireless Communication:** (SER1) Bi-directional GSM/SMS/Call, one-way GPRS/FTP
- **Wired communication:** (SER2) Modbus RTU master/slave, 300-19200bit/. Optionally: Modbus TCP Master/slave over Ethernet.
- **Programming device:** AutoLog® GsmProgrammer or mobile phone (SMS)
- **Programming language:** Text type instruction list
- **Programming instructions:** Logical operations, arithmetic, compare, timer, counter, free define control instructions, security, reporting SMS/FTP, incoming phone number identification, time-table controls, data logging etc.
- **Variables:** Binary memories, 8-bit registers, 16-bit words, counters, retained variables, text strings memories for phone numbers, pulse variables, etc.

- **Application:** Max. 512 programming lines. Max. 240 phone numbers, max. 256 time tables, max. 256 iButtons, 32 PID controllers, max. 8 FTP files for data logging and reports, controls (SMS/Call), alarms (SMS/Call) etc.
- **Wireless M2M communication:** AutoLog ControlMan to GSM-RTU, Indusoft Web Studio SCADA to GSM-RTU, PostGre SQL Database to GSM-RTU, Mobile phone to GSM-RTU, GSM-RTU to GSM-RTU, Database to RTU, Mobile to RTU etc.
- **Analogue input signal selection with module:** Pt100 -50..+150 ºC, KTY10, 0..5 V, 0..10 V, 0..20 mA, 4..20 mA, RMS, special modules etc. Look Appendix!
- **Measurements:** temperature, current, voltage, kWh, pH, CO2, pressure, counters etc. (Needs external sensor, Optionally FF-Automation can supply also sensors)
- **Control room softwares:** AutoLog® ControlMan (Internet), Indusoft Web Studio SCADA, Any SCADA which supports ODBC connection to PostgreSQL Database. GSM-PLC Demo applications for Indusoft Web Studio.
- **Control application features:**
    - 512 programming lines
    - 240 phone numbers
    - 256 lines for time-table controls
    - 256 iButton
    - Full capability for making PLC (Programmable Logic Controller) type application programs
    - Easy to use GSM communication interface
    - Max. 32 PID controllers (depends on I/O quantity how many can be used)
    - Retained battery backup memories for variables and clock calendar
    - Password, incoming phone number detection and secret phone number security
    - Full programming and parameter settings with cable or through GSM network
    - Capability for free call controls
    - MODBUS master/slave for SER2
    - Analog input calibration
    - Alarm SMS to mobile(s), circulation alarm system support
    - Alarm acknowledgement with GSM or from ControlMan
    - Measurement report query from mobile phone
    - SMS controls from mobile phone or from ControlMan
    - Free define control commands etc
- **GPRS/FTP data logging, memory capability: 256 KB:**
    - 1 measurement = 43000 events = 30 days with 1 minute logging interval
    - 4 measurements = 21000 events = 15 days with 1 minute logging interval
    - 8 measurements = 13000 events = 9 days with 1 minute logging interval
    - Files can be send to FTP server through GPRS
    - max 8 FTP files
- **Full Applications** (FF-Automation can sell the product with ready-to-use application program)
    - AutoLog SaveLight for street light controlling and energy saving
    - AutoLog Wastewater pump station control
    - AutoLog GSM Demo Kit for System integrators for testing and demonstration.

5.3 DIP switches and Jumpers

(GSM-8 in the picture)



5.4 DIP-switches

DIP	ON	OFF
1	EEPROM not write protected.	EEPROM write protected. programming is not allowed
2	Calibration is enabled.	Calibration is disabled.
3	SER2: Force Serial port 2 mode to Modbus RTU, 9600 bit/s, parity = None, slave address = 1	SER2 mode is defined by RO215.
4*	SER1: Build-in GSM modem is disabled. Programming using direct connection through Ser1 (RS-232) is enabled. *DIP 4 is not used in GSM-20	SER1: Build-in GSM modem activated. (Reboot PLC) Programming using direct connection through Ser1 (RS-232) is disabled. *DIP 4 is not used in GSM-20
5	Data memory cleared after power reset	Data memory retained during power failure
6	US band selected (850/1900MHz)	EU band selected (900/1800MHz)

5.5 Jumpers in GSM-GW, GSM-4, GSM-8, GSM-16

JP1	ON	OFF
1-2	Backup battery connected	Backup battery not connected. Use this setting for long time storage.
3-4	Watchdog active	Watchdog not active
5-6	Power supply alarm limit 9 VDC (for 12VDC power supply)	Power supply alarm limit 17 VDC (for 24VDC power supply)

JP1-2: ON: When Backup battery is connected, the lithium battery is used to retain the variables and update the clock & calendar. In this position and if not powered, the battery can last about 6 months-2 years.

JP1-2 OFF: If the PLC is not powered for long time (e.g. in storage), the battery can be disconnected using this jumper.

JP 3-4 ON: (Default position). Watchdog is used to detect failures in PLC's system program and automatically resets the PLC if failure happens.

JP3-4 OFF: Watchdog is not active.

JP5-6 ON: PLC program is not executed if the power goes bellow the low alarm limit 9VDC. Use this setting if you are using 12VDC power supply.

JP5-6 OFF: PLC program is not executed if the power goes bellow the low alarm limit 17VDC. Use this setting if you are using 24VDC power supply.

5.6 Jumpers in GSM-20

JP1	ON	OFF
3-4	Watchdog active	Watchdog not active
1-2	Power supply alarm limit 9 VDC (for 12VDC power supply)	Power supply alarm limit 17 VDC (for 24VDC power supply)
JP2	**ON**	**OFF**
3-4	Enable PLC's internal 12VDC power supply for EXT I/O channel. 12V LED is ON.	Disable PLC's internal 12VDC power supply for EXT I/O channel. 12V LED is OFF.
1-2	Disabled	Disabled

Look circuit board! Number 1 shows the pin 1. Pin positions are as bellow:

JP1	**JP2**
4	**4**
3	**3**
2	**2 (Disabled)**
1	**1 (Disabled)**

JP1: 3-4 ON: (Default position). Watchdog is used to detect failures in PLC's system program and automatically resets the PLC if failure happens.

JP1:3-4 OFF: Watchdog is not active.

JP1:1-2 ON: PLC program is not executed if the power goes bellow the low alarm limit 9VDC. Use this setting if you are using 12VDC power supply.

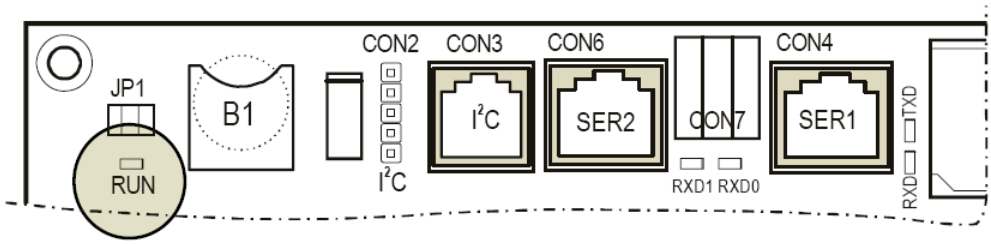
JP1:1-2 OFF: PLC program is not executed if the power goes bellow the low alarm limit 17VDC. Use this setting **only** if you are using 24VDC power supply.

JP2:3-4 ON: Enable PLC's internal 12VDC power supply for EXT I/O channel. 12V LED is ON.

JP2:3-4 OFF: Disable PLC's internal 12VDC power supply for EXT I/O channel. 12V LED is OFF.

JP2:1-2 ON: Disabled. (Pin is physically cut off.) If the PIN is not cut off, do not connect these with jumper!
JP2:1-2 OFF: Disabled

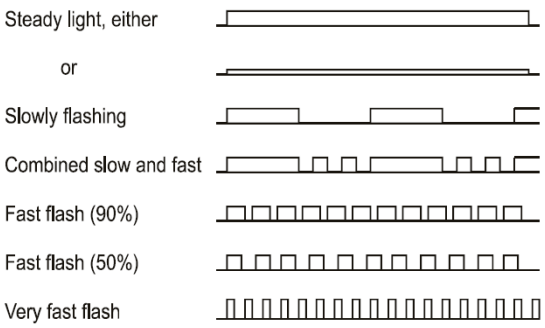
5.7 RUN - LED



Red RUN –LED indicates the functional state of the controller as follows:

LIGHT	PROGRAM STATE	CAUSED BY
Steady light	Initializing GSM-PLC	After power ON.
Slow blink (period 2 sec)	Initializing modem	
Very slow blink (period 4 sec)	Normal operation. PLC application program is running.	
Fast blink (period 0.5 sec)	PLC does not execute application program, but can receive and response to messages (SET, READ)	
Very fast blink (period 0.1 sec)	Power fail state	All outputs are cleared
Slow/fast blink	Active alarm that needs acknowledging. 4 fast blink and 2 seconds off.	Alarm flags are acknowledged automatically if timer is set. Manually using e.g. ACK ALL command in Alarm Log.

RUN –LED can be found on every GSM-PLC and the functionality is the same.



Trouble shooting

GSM-PLC run led blinks fast
a) Download program again
b) Send command DEL ALL from Alarm Log and download the whole project (program & phone book)

RUN Led is blinking fast and outputs are always OFF
a) GSM-PLC is in power failure state. Check supply voltage.

5.8 SER 1 & 2 Communication & Modem - LEDs

Modem status LED (GSM)

LED off:
Modem switched off
(Not ready)

LED on:
Modem switched on
Connecting to network

LED flashing slowly:
Modem switched on
(Idle mode)

LED flashing rapidly:
Modem switched on
(Transmission mode)

Communication LEDs:

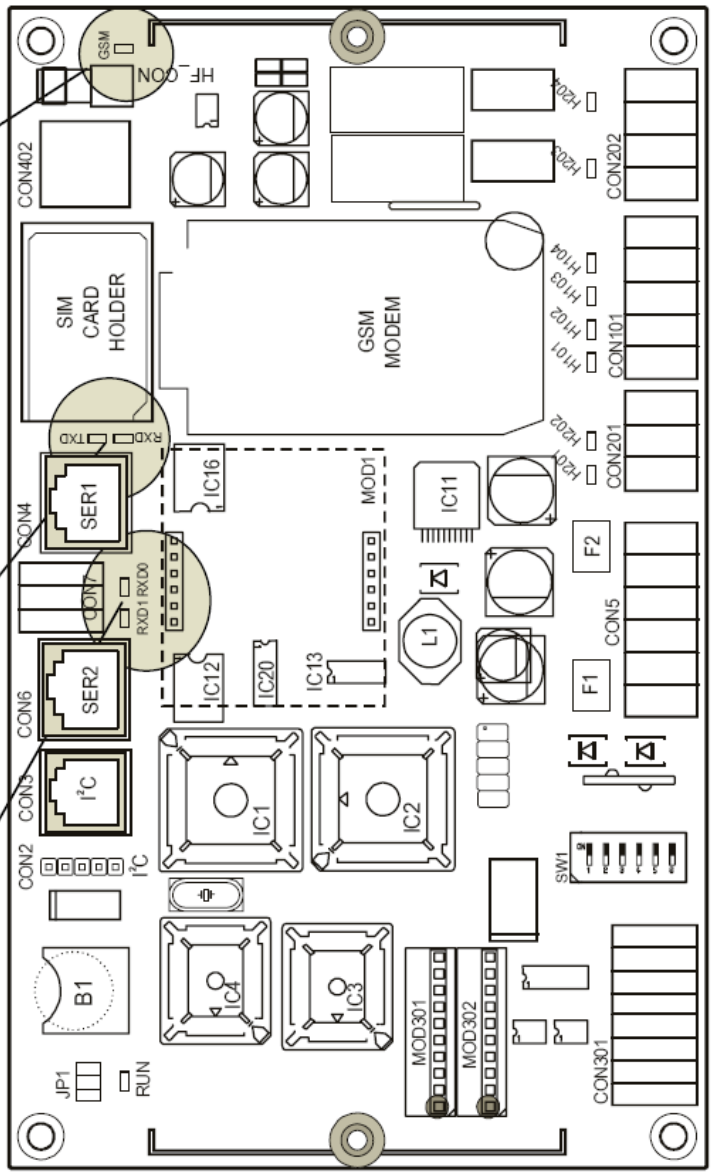
RXD LED on (blinking)
Receiving data on SER1

TXD LED on (blinking)
Sending data on SER1

Communication LEDs:
(Near RS-485 connector)

RXD LED on (blinking)
Receiving data on SER2

TXD LED on (blinking)
Sending data on SER2



Communication –LEDs can be found on every GSM-PLC and the functionality is the same. (In the picture is GSM-8)

Trouble shooting

Program sends messages to Alarm Log when PC is connected to GSM-PLC but when modem is connected, there is no SMS-message.

a) Check the red led located near GSM-modem. When led is blinking, modem has connection to network. Check DIP switch 4 position.
b) Check the phone number defined in phone list

If Modem's status led is not blinking:

a) Check PIN code setting from GSM-PLC. GSM-PLC tries to configure the modem twice, so if PIN code is wrong, you have one attempt left to set correct PIN code to SIM card.
b) Check that the SIM card supports 900MHz network

5.9 I/O - LEDs

Indication LEDs:
(Digital outputs)

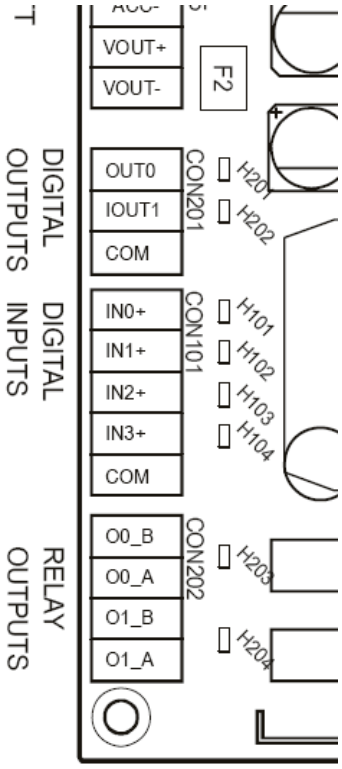
H201 Output 0
H202 Output 1

Indication LEDs:
(Digital inputs)

H101 Input 0
H102 Input 1
H103 Input 2
H104 Input 3

Indication LEDs:
(Relay outputs)

H203 Relay Output 0
H204 Relay Output 1

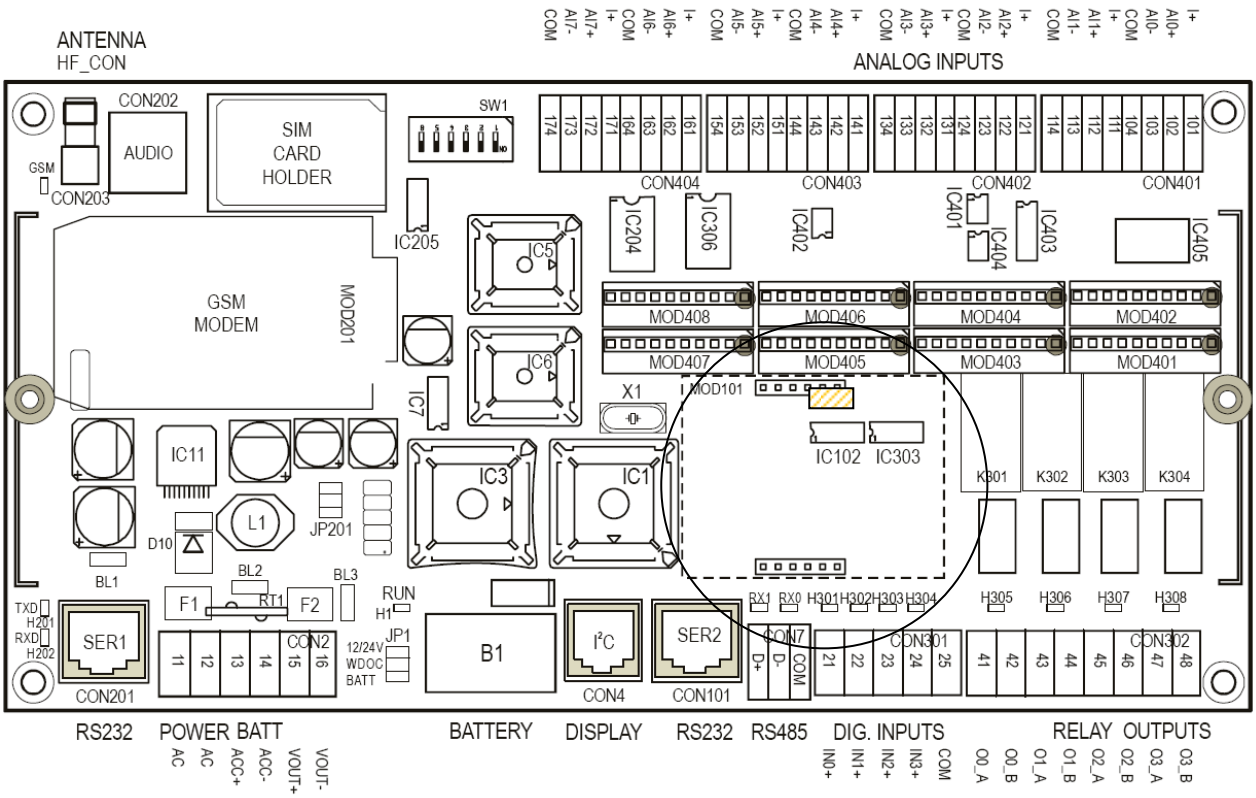


I/O –LEDs can be found on every GSM-PLC and the functionality is the same.
(In the picture is GSM-8)

Trouble shooting:

RUN Led is blinking fast and outputs are always OFF
a) GSM-PLC is in power failure state. Check supply voltage.

5.10 RS-232 Plug-in Module (PiM)



(The above picture is taken from GSM-16, but the principle is the same in every GSM-PLC)

RS-232 Plug-in modules are placed to RS-232 Plug-in module place so that the mounting hole (diameter about 3mm) in the modules circuit board is on the top of the mounting hole in the PLC's circuit board (diameter about 3mm). Plastic connector should be used to tighten the PiM to the PLC board.

Note! The PLC board's mounting hole may be covered with the sticker. In this case you should remove the sticker.

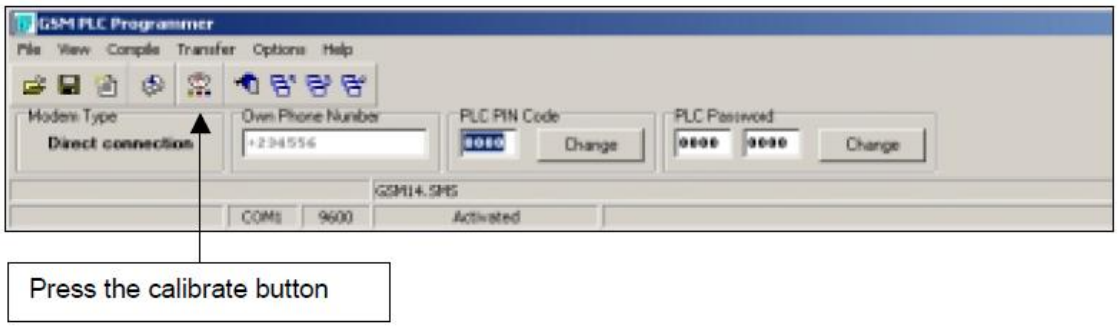
In the PLC board there is a jumper (marked as ▨) to connect the Rx signal off, if PiM is not used. Before inserting the PiM, power off the PLC and remove this jumper.

Take power off from the PLC, when you insert or remove the PiM!

5.11 Analog input calibration

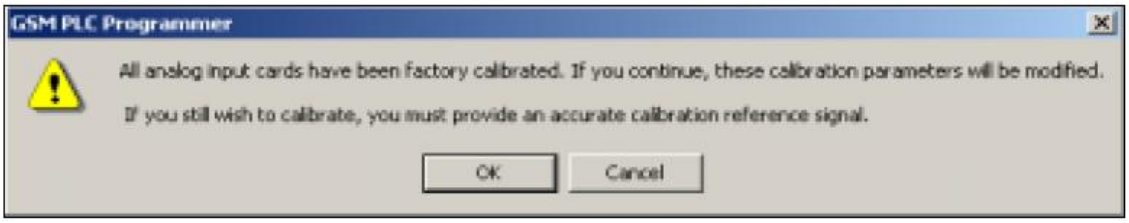
When you add new analog input modules to GSM-PLC board, you need to calibrate the modules to get the right values.

Note! Normally analog inputs are already calibrated by FF-Automation in the factory before delivery!!!



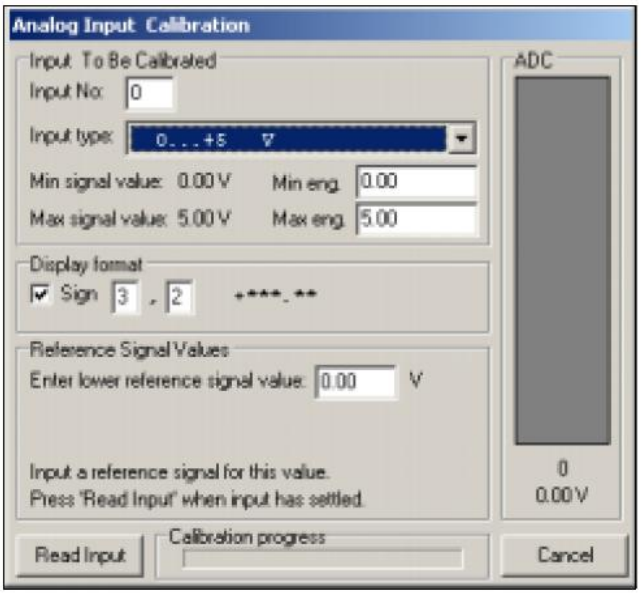
1. After pressing the calibrate button, user can see following message box

2. Press OK.



3. Choose Input Number, Input Type, Min eng. value, Max eng. value, Display format and enter lower reference signal value (voltage, current etc.) After that read the input by pressing the read input button.

4. Finally enter upper reference signal value and press the read input button. GsmProgrammer informs if calibration is successfully performed.



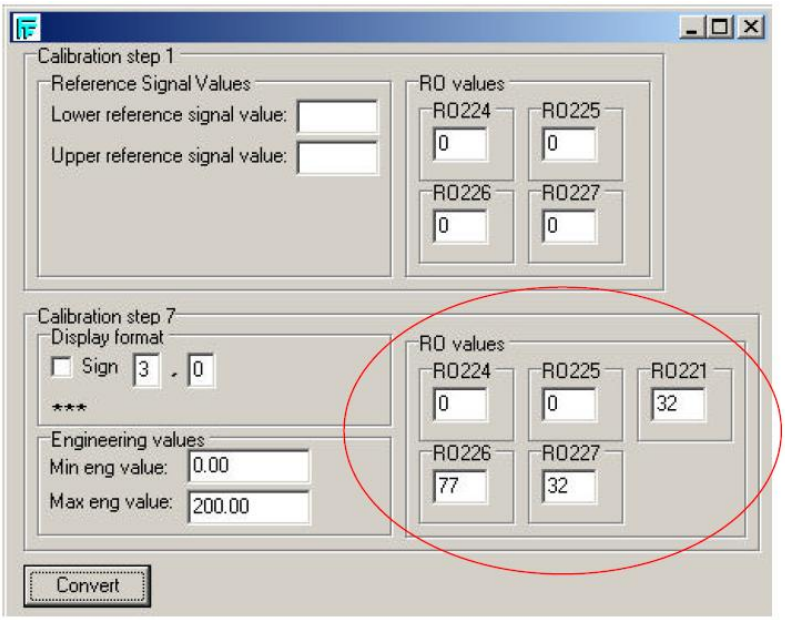
5.12 Analog input scaling

Analog input scaling can be done by FF-Automation in the Factory before delivery – if FF-Automation gets the scaling parameters from customer when products are ordered.

Sometimes customer doesn't know the scaling parameters during the time of order and so the scaling needs to be done afterwards.

Scalings can be done with the GSM programmer's calibration program, but it is strongly recommended that scaling are done using the help of cal.exe program!!! It the calibration program is used you will need to also calibrate the analog inputs (you cannot just give new scaling) and it is much bigger job for which you need calibrated signal generators etc.

1. Start Cal.exe (if you don't have this program ask it from ff-automation or download it from FF-Automation's distributor download area)
2. Set display format
3. Set Engineering values (min, max)
4. Press Convert
5. Take with RO values: RO221, RO224, RO225, RO226, RO227



6. Select the used analog input (0…7…) with RO222
7. Send these RO values and RO220=9 to GSM-PLC with text message.

Example text message: analog input 5 and 0000=PSW:
0000 SET RO222=5*0000 SET RO224=0*0000 SET RO225=0*0000 SET RO221=32*0000 SET RO226=77*0000 SET RO227=32****0000 SET RO220=9****

or with Alarm Log:

SET RO222=5*0000 SET RO224=0*0000 SET RO225=0*0000 SET RO221=32*0000 SET RO226=77*0000 SET RO227=32****0000 SET RO220=9***

Using cal.exe program you will not overwrite calibration parameters, only the scaling parameters.

6 AutoLog GSM-PLC mounting and physical installation

6.1 Installation

AutoLog GSM-PLC's can be installed on DIN rail. This is the default option. Optionally GSM-PLCs can have screw connectors. Stainless steel and polycarbonate enclosure models can be mounted also side of pole. In special cases it is suggested to contact FF-Automation for selecting right installation type!

6.2 Enclosures

By default AutoLog GSM-PLCs are delivered with IP20 metal cover. (case)

"GSM-PLC Unit" models are mounted inside IP65 ABS plastic enclosure. Optional HMI can be mounted on the cover of this plastic enclosure.

Optionally GSM-PLCs can be delivered inside IP65-IP67 polycarbonate plastic enclosure (more resistant to direct sunshine.)

Optionally GSM-PLCs can be delivered also inside metal enclosure (Aluminum or Stainless steel). This kind of enclosure needs external antenna if the signal strength is insufficient.

In special cases it is suggested to contact FF-Automation for selecting right enclosure!

6.3 Environmental requirements

The inputs and outputs of the GSM-PLCs are isolated in groups. The serial connectors RS232 (SER2), parallel with it RS485 and RS232 (SER1) are not isolated from PLC's internal voltage. To ensure proper function of GSM-PLC the installation circumstances must be carefully considered.

Operating temperature -20 to +60°C

Moisture, corrosive gases, liquids and conductive dust must not be present where PLC boards are installed.

The PLC boards don't resist heavy vibration very well.

The distance from electromagnetic field generation devices, such as electric motors, switch gear, thyristors, welding equipment, switched power supplies and power converters/inverters must be adequate.

PLC boards are quite immune against light sources.

If some item above or any other environmental feature may cause errors to PLC function it is advised to install the PLC in the steel plate enclosure. It is also advisable to install the PLC install possible auxiliary input/output relays, fuses and power units in their own enclosure near the PLC. All the contactors connected to PLC have to be equipped with RC protection devices and the 24 VDC control relays with extinguish diodes.

6.4 Field wiring

6.4.1 Earthing / grounding

The metal parts of the PLC enclosure must be connected to the plant's logic ground.

6.4.2 Power supply connections

Normally no functional grounding is necessary when 12/24 volt floating voltage system is used with the PLC's power supply connection. It is important that the 24 volt wiring is carefully kept isolated from ground level and 230 volt supply voltages.

The CPU board of GSM-PLC, receives the +12/24 VDC supply voltage from an external isolated power supply unit. The controller converts this to the voltage it requires.

6.4.3 Digital inputs (PNP)

12/24 VDC / max. 8 mA / PNP / opto-isolated

The digital inputs are isolated in group in GSM-PLC's CPU board. Externally powered potential free contacts or PNP type inductive/capacitive sensors may be connected to PLC. The supply voltage for the PNP sensors is normally taken from PLC's terminals.

It is recommended to use twisted pair shielded cables in input wiring. The cabling should be installed separately from the 230/400 VAC power cabling. The cable shields may be connected to logic ground only at one point, normally at the end on the PLC enclosure. If there is heavy electromagnetic disturbance, the PLC's unisolated inputs can be isolated from field equipment with relays or optoelectronic modules.

6.4.4 Digital outputs (NPN)

12/24VDC / max. 2A / group max. 4A / NPN / opto-isolated

6.4.5 Digital (relay) outputs (NPN)

12-24 VDC / max. 3A , 24-230 VAC

Digital outputs are isolated in GSM-PLC board (1A / 30VDC / 250 VAC). It is recommended to use twisted pair shielded cables in output wiring. The cabling should be installed separately from the 230/400 VAC power cabling. The cable shields may be connected to logic ground only at one point, normally at the end on the PLC enclosure. If there is heavy electromagnetic disturbances, the PLC's isolated outputs can be double isolated from field equipment with relays or optoelectronic modules.

6.4.6 Analog inputs

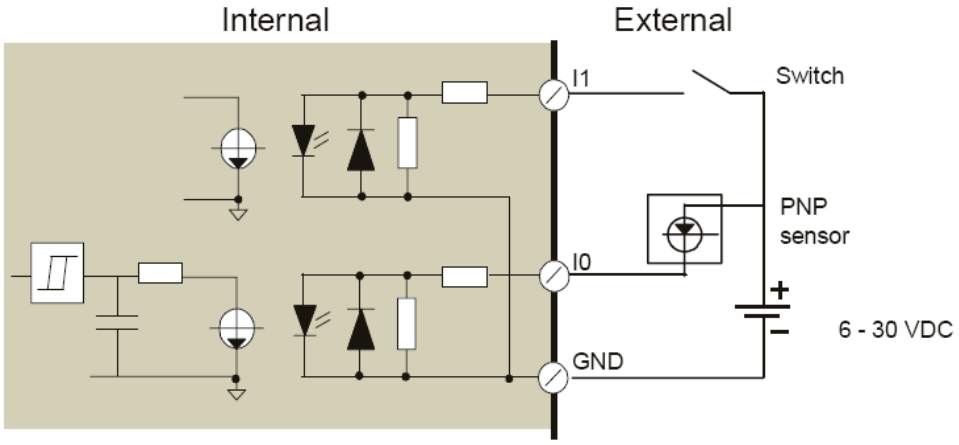
12 bit, pointwise plug-in input modules:
Pt100 (-50..150°C / 0..500°C / -250..750°C),
0..5mA,
0..20mA,
4..20mA Isolated,
KTY10 (-50..150°C),
NTC (-5..50°C), 0..2V, 0..5V, 0..10V, -10..10V,
RMS 40VAC/25VAC/ 0.25VAC etc.

The analog inputs are unisolated in GSM-PLC boards. If active transducers are used, it should be checked that the output circuits of the transducers are galvanically isolated from its supply voltages. If you can't be sure of isolation, it is advisable to furnish the transducer with an external galvanic isolator. The supply power for passive transducer is normally taken from PLC's power unit.

It is recommended to use twisted pair shielded cables in output wiring. The cabling should be installed separately from the 230/400 VAC power cabling. The cable shields may be connected to logic ground only at one point, normally at the end on the PLC enclosure. Connection examples for digital inputs / outputs

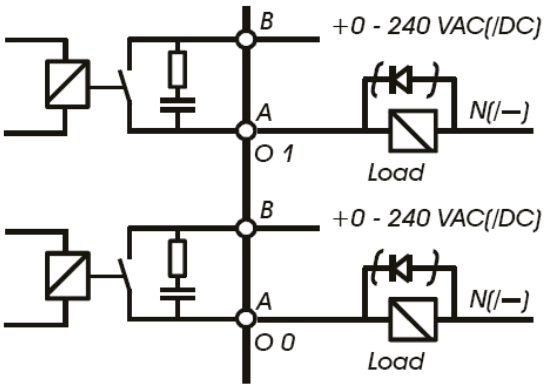
6.4.7 Connecting a switch or PNP sensor to digital input

Terminal strips for 1x1.5 mm^2 wire are provided.

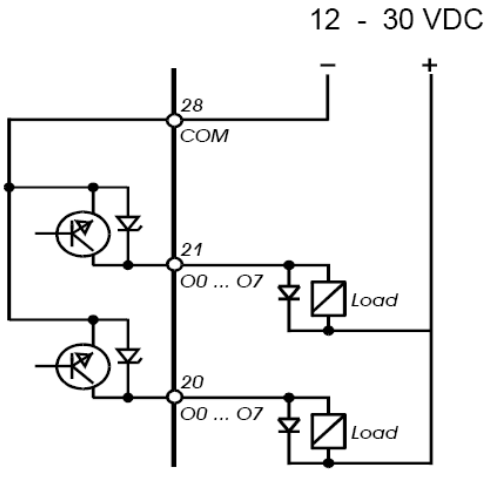


6.4.8 Connecting loads to relay outputs

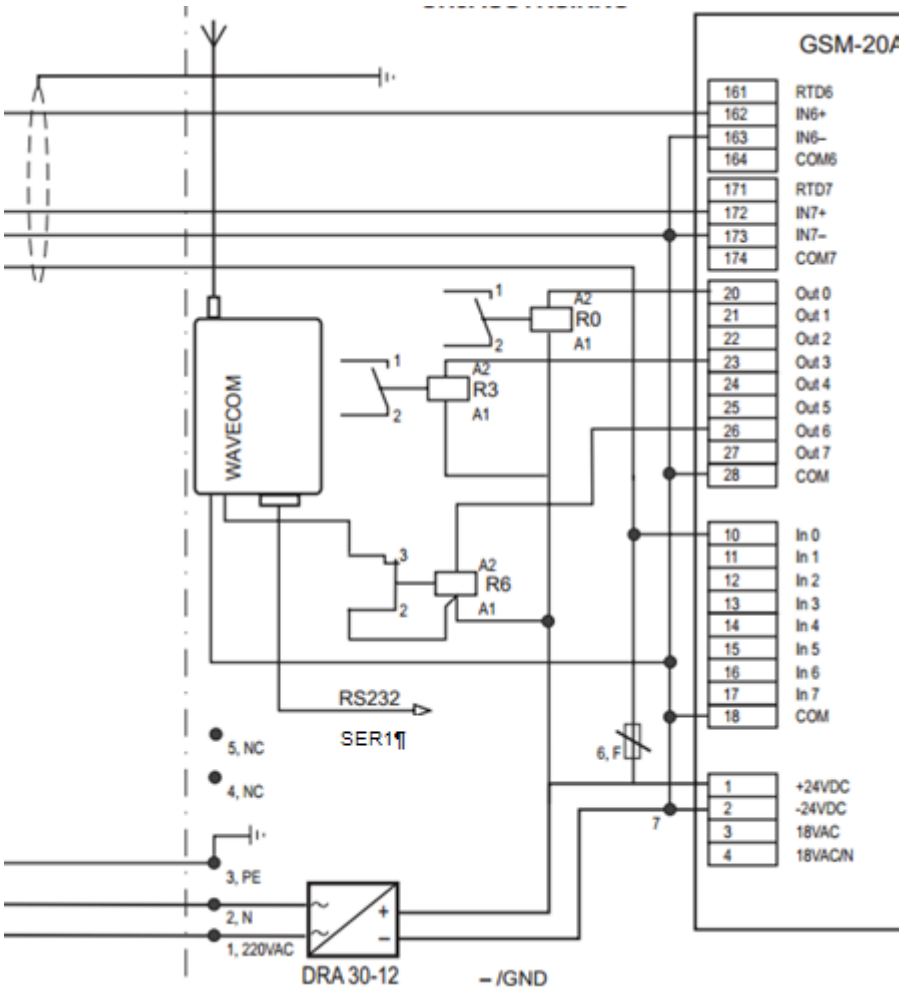
In GSM-PLC's there are transistor outputs. In GSM-8 and GSM-16 there are also relay outputs. Terminal strips for 1x1.5 mm^2 wire are provided.



Connection of load to relay output



Connection of load to group isolated transistor (FET) output



Example of connecting relay R6 to switch power off from the modem when DO6 =1 (NC). Digital outputs are NPN type.

Following code resets gsm modem at the midnight.

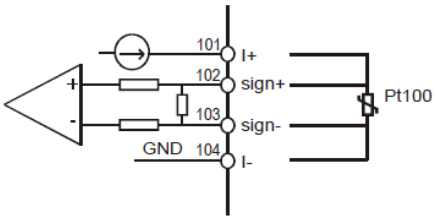
```
'CLK=2359' M90
'M90=1' DO6=1 ;resets gsm modem
'DO6S10' DO6=0
```

6.4.9 Connecting Analog inputs

There are 0 - 8 individually adaptable analog inputs which can be used for the measurement of temperature, current or voltage signals. The type of analog input is defined according to adapter module. Every measured signal / range needs its own individual adapter module also called as analog input module.

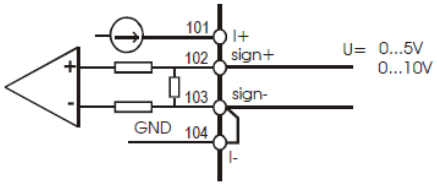
6.4.10 Connection examples

Because of the low signal levels to be measured, shielded twisted-pair cables should be used. The shield should be grounded at one end only in order to avoid degradation of accuracy caused by external disturbances.



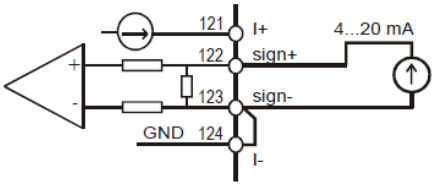
Temperature measurement with a PT100 sensor

The current (I+ - I-) flows through the PT100 sensor. The measurement is made by inputs sign+ and sign-



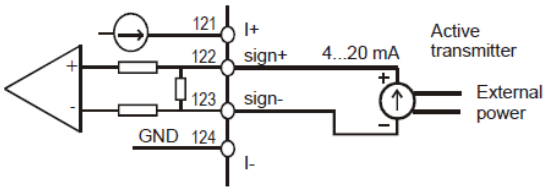
Voltage measurement

Two ranges are available: 0...5V or 0...10V.



Current measurement

Two ranges are available: 0...20mA or 4...20mA.



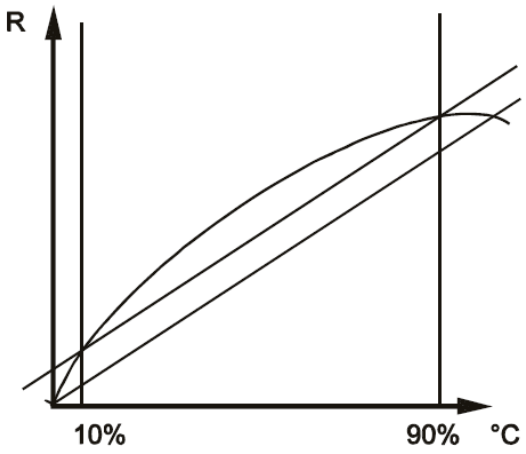
Current measurement, isolated input module

Available range is: 4...20mA.

6.4.11 Temperature measurement with a PT100 sensor

The Pt100 sensor is nonlinear. The following temperatures corresponds to the resistance values given in the table (DIN 43 760) below:

°C	W	Diff. W	°C	W	Diff. W	°C	W	Diff. W
-50	80.31	0.40	55	121.32	0.39	160	161.04	0.37
-45	82.29	0.40	60	123.24	0.38	165	162.90	0.37
-40	84.27	0.39	65	125.16	0.39	170	164.76	0.37
-35	86.25	0.39	70	127.07	0.38	175	166.61	0.37
-30	88.22	0.39	75	128.98	0.38	180	168.46	0.37
-25	90.19	0.39	80	130.89	0.38	185	170.31	0.37
-20	92.16	0.39	85	132.80	0.38	190	172.16	0.37
-15	94.12	0.39	90	134.79	0.38	195	174.00	0.37
-10	96.06	0.39	95	136.60	0.38	200	175.84	0.37
-5	98.04	0.39	100	138.50	0.38	205	177.68	0.37
0	100	0.39	105	140.39	0.37	210	179.51	0.37
5	101.95	0.39	110	142.29	0.39	215	181.34	0.37
10	103.90	0.39	115	144.17	0.37	220	183.17	0.37
15	105.85	0.39	120	146.06	0.38	225	184.99	0.36
20	107.79	0.39	125	147.94	0.37	230	186.82	0.37
25	109.73	0.38	130	149.82	0.37	235	188.63	0.36
30	111.67	0.39	135	151.70	0.37	240	190.45	0.36
35	113.61	0.39	140	153.58	0.38	245	192.26	0.36
40	115.54	0.39	145	155.45	0.38	250	194.07	0.36
45	117.47	0.39	150	157.31	0.37	255	195.88	0.36
50	119.40	0.39	155	159.04	0.37	260	197.69	0.36



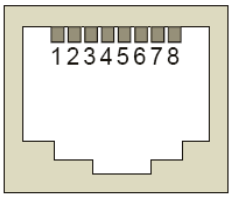
In order to make the adjustment of the Pt100 input as accurate as possible over the measurement range, different adjustment points have to be used. Instead of adjusting the OFFSET at the lowest value we use a 10% point as the adjustment point for the low end of the measurement range. In the same way we use the 90% point as the adjustment point for the high value. This increases the overall accuray.
The curve in picture is exaggerated.

6.5 Serial communication

The GSM8 PLC board has two RS232 serial connections by RJ 45 connectors anc with SER2 a parallel RS485 connector. The RS485 communication needs ar additional converter board installed on CPU card. The communication behaviour of serial ports are defined using register outputs.

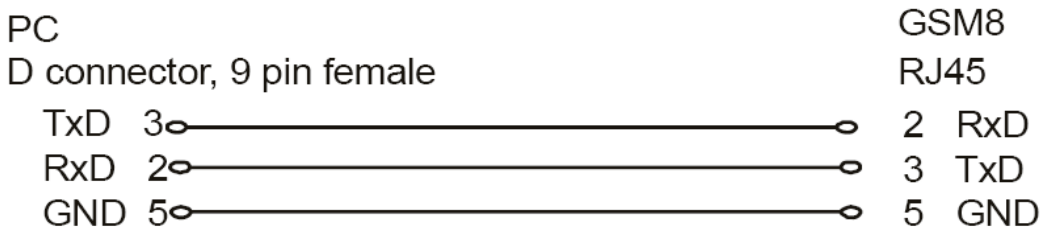
6.6.1 Connector signals

Pin	Signal	Pin	Signal
1	+5V (form PLC)	6	N/C
2	RXD (data to PLC)	7	RTS
3	TXD (data from PLC)	8	CTS
4	DTR (approx. 12V from PLC)		
5	GND		

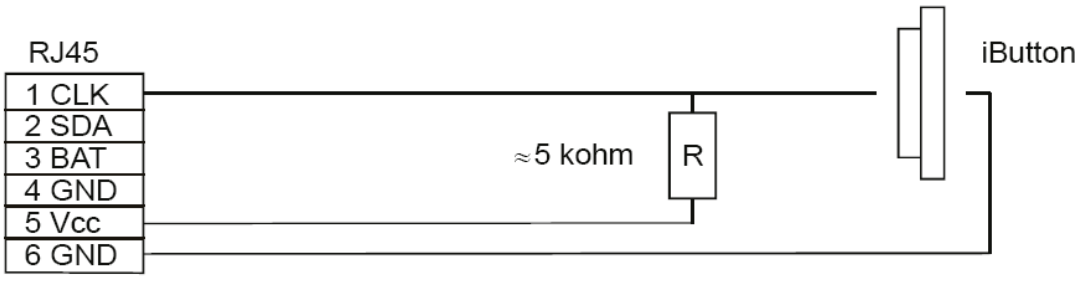


Pin numbers of RJ45 connector, seen from front of female connector

6.5.1 Interconnection cables



6.6 Ibutton cable connection



PROGRAMMING MANUAL

7 Variables

AutoLog GSM-PLC's are programmed using AutoLog GsmProgrammer software.

To open the application program editor click View->Source code

GSM-PLCs have many different variable types. You can use these variables in your application.

You don't need to declare variables. There are no symbolic names for variables so it is advisable to make good side notes! Look Pumping station demo application for example!"

Note that not all variables values are retained, after the GSM-PLC is rebooted.

You can force to clear variable values by setting the DIP switch 5 to ON position and rebooting the PLC, but note that you will lose all the serial port settings etc. if you have not defined these in your application program.

If the lithium battery goes empty, you will also lose all the variable values, if the PLC is rebooted.

7.1 Variable list

Syntax is: [Variable type][memory address number]

e.g. DI10

DI **Digital input (DI0-DI[n]), number (n) depends on HW model,**

DO **Digital output (DO0-DO[n]), number (n) depends on HW model,**
Unused DOs can be used as memories but with care! (DO[n+1]-DO240)

AI **Analog input (AI0-AI[n]), number (n) depends on HW model,**

AO **Analogue output (AO0-AO[n]), number (n) depends on HW model,**
Unused AOs can be used as word memories but with care! (AO[n+1]-AO255)

M **Binary Memory (M0-M255)**

WM **Word Memory (WM0-WM255) (16-bit, 0-65535)**

WP **Word Pointer to WM area**

CN **Counter, number is same as for Digital input (DI0-DI[n]),**
Number (n) depends on HW model

RO **Register Outputs, (RO0-255), special system program function registers**
Look list of RO addresses and their functions later in this manual. Do not use these for any other purpose as reading or setting system program parameters.

N **Phone book numbers (N0-N244)**
N245-252 Writes string to FTP files 0-8
N253 Incoming call number
N254 **SMS phone number (N254)**
Incoming SMS phone number
N255 Writes string to HMI display

RO94 Incoming Call type

DATE **Date**

CLK **CLK (Time)**

$[Mes] $ Self defined messages

T **Clip phone number (without land code)**

P **Pulse variable. Pulse n=1 = 1 sec, n=2 = 1 min.**
Pulse variable is active one program cycle every second/minute

AF **AF variable. Alarm Flag**
Every program [n] line has alarm flag AF[n]

CS **Call status**

TC **Time Control table**

7.2 Digital input

DI[n] Binary input, Boolean, quantity depends on hardware, n=[0-255]

DI240-DI255 Reserved for (HMI) KEYPAD,
0= DI240, 1=DI241, 2=DI242, 3=DI243, 4=DI244, 5=DI245, 6=DI246, 7=DI247, 8=DI248, 9=DI249, A=DI250, B=DI251, C=DI252, D=DI253, E=DI=254 F=DI255

e.g. DI0 ; gets physical digital input 0 value (0=0V, 1=24V) on GSM-PLC
In source code: 'DI0=1' "HELLO WORLD" 0 ; if digital input =1 then send "HELLO WORLD" text to phone number 0.

Digital inputs in I/O expansion cards, Only GSM-20 supports I/O expansion cards:

Digital input addresses in expansion card RIO8, Only in GSM-20
RIO8 (Card address 1. J12=On, J34=On, J56= On) DI32-39, (connected card status RO2.0)
RIO8 (Card address 2. J12=Off, J34=On, J56= On) DI40-47, (connected card status RO2.1)
RIO8 (Card address 3. J12=On, J34=Off, J56= On) DI48-55, (connected card status RO2.2)
RIO8 (Card address 4. J12=Off, J34=Off, J56= On) DI56-63, (connected card status RO2.3)
RIO8 (Card address 5. J12=On, J34=On, J56= Off) DI64-71, (connected card status RO2.4)
RIO8 (Card address 6. J12=Off, J34=On, J56= Off) DI72-79, (connected card status RO2.5)
RIO8 (Card address 7. J12=On, J34=Off, J56= Off) DI80-87, (connected card status RO2.6)
RIO8 (Card address 8. J12=Off, J34=Off, J56= Off) DI88-95, (connected card status RO2.7)

Digital inputs addresses in expansion card DI16, Only in GSM-20!
DI16 (Card address 1. J12=On, J34=On) DI32-47, (connected card status RO2.0 & RO2.1)
DI16 (Card address 2. J12=Off, J34=On) DI48-63, (connected card status RO2.2 & RO2.3)
DI16 (Card address 3. J12=On, J34=Off) DI64-79, (connected card status RO2.4 & RO2.5)
DI16 (Card address 4. J12=Off, J34=Off) DI80-95, (connected card status RO2.6 & RO2.7)

7.3 Digital output

DO[n] Binary output, Boolean, quantity depends on hardware, n=[0-255]

- DO246=1 Activate Transparent mode.
- DO247=1 Incoming number (Num253) recognized.
- DO248…DO255 Controls led of HMI control panel, 1…8

DO248 controls LED1
DO249 controls LED2
DO250 controls LED3
DO251 controls LED4

DO252 controls LED5
DO253 controls LED6
DO254 controls LED7
DO255 controls LED8

Digital outputs in I/O expansion cards, Only GSM-20 supports I/O expansion cards:

Digital output addresses in expansion card DO32, Only in GSM-20!
DO32 (Card address 1. J12=On, J34=Off) DO32-63, (connected card status RO3.0-RO3.3)
DO32 (Card address 2. J12=Off, J34=On) DO64-95, (connected card status RO3.4-RO3.7)

Digital output addresses in expansion card RIO8, Only in GSM-20!
RIO8 (Card address 1. J12=On, J34=On, J56= On) DO32-39, (connected card status RO3.0)
RIO8 (Card address 2. J12=Off, J34=On, J56= On) DO40-47, (connected card status RO3.1)
RIO8 (Card address 3. J12=On, J34=Off, J56= On) DO48-55, (connected card status RO3.2)
RIO8 (Card address 4. J12=Off, J34=Off, J56= On) DO56-63, (connected card status RO3.3)
RIO8 (Card address 5. J12=On, J34=On, J56= Off) DO64-71, (connected card status RO3.4)
RIO8 (Card address 6. J12=Off, J34=On, J56= Off) DO72-79, (connected card status RO3.5)
RIO8 (Card address 7. J12=On, J34=Off, J56= Off) DO80-87, (connected card status RO3.6)
RIO8 (Card address 8. J12=Off, J34=Off, J56= Off) DO88-95, (connected card status RO3.7)

D*igital output addresses in expansion card RO16, Only in GSM-20!*
RO16 (Card address 1. J12=On, J34=On) DO32-47, (connected card status RO3.0 & RO3.1)
RO16 (Card address 2. J12=Off, J34=On) DO48-63, (connected card status RO3.2 & RO3.3)
RO16 (Card address 3. J12=On, J34=Off) DO64-79, (connected card status RO3.4 & RO3.5)
RO16 (Card address 4. J12=Off, J34=Off) DO80-95, (connected card status RO3.6 & RO3.7)

Digital output values are not retained after booting the GSM-PLC.

7.4 Analog input

AI[n] Analog input, n=[0-255], quantity depends on hardware,
12-bit value calibrated to raw values 0-4000 (not 0-4096) for easier handling.

Analog inputs in I/O expansion cards, Only GSM-20 supports I/O expansion cards:

Analog input addresses in expansion card EXA8/4, Only in GSM-20!
EXA8/4 (Card address 1. J12=On, J34=On, J56= On) AI32-39, (connected card status RO4.0)
EXA8/4 (Card address 2. J12=Off, J34=On, J56= On) AI40-47, (connected card status RO4.1)
EXA8/4 (Card address 3. J12=On, J34=Off, J56= On) AI48-55, (connected card status RO4.2)
EXA8/4 (Card address 4. J12=Off, J34=Off, J56= On) AI56-63, (connected card status RO4.3)
EXA8/4 (Card address 5. J12=On, J34=On, J56= Off) AI64-71, (connected card status RO4.4)
EXA8/4 (Card address 6. J12=Off, J34=On, J56= Off) AI72-79, (connected card status RO4.5)
EXA8/4 (Card address 7. J12=On, J34=Off, J56= Off) AI80-87, (connected card status RO4.6)
EXA8/4 (Card address 8. J12=Off, J34=Off, J56= Off) AI88-95, (connected card status RO4.7)

7.5 Analog output

AO[n] Analog output, quantity depends on hardware, n=[0-255]
12-bit value calibrated to raw values 0-4000 (not 0-4096) for easier handling.

Analog outputs I/O expansion cards, Only GSM-20 supports I/O expansion cards:

A*nalog output addr. in expansion card EXA8/4, Only in GSM-20!*
EXA8/4 (Card address 1. J12=On, J34=On, J56= On) AO32-35, (connected card status RO4.0)
EXA8/4 (Card address 2. J12=Off, J34=On, J56= On) AO40-43, (connected card status RO4.1)
EXA8/4 (Card address 3. J12=On, J34=Off, J56= On) AO48-51, (connected card status RO4.2)
EXA8/4 (Card address 4. J12=Off, J34=Off, J56= On) AO56-59, (connected card status RO4.3)
EXA8/4 (Card address 5. J12=On, J34=On, J56= Off) AO64-67, (connected card status RO4.4)
EXA8/4 (Card address 6. J12=Off, J34=On, J56= Off) AO72-75, (connected card status RO4.5)
EXA8/4 (Card address 7. J12=On, J34=Off, J56= Off) AO80-83, (connected card status RO4.6)
EXA8/4 (Card address 8. J12=Off, J34=Off, J56= Off) AO88-91, (connected card status RO4.7)

Analog output values are not retained after booting the GSM-PLC

7.6 Binary memory

M[n] Binary memory, (0-1) [n]= 0..255

M 0-127 are reset in booting up the GSM-PLC.
M 128-255 are retained (if DIP5=OFF), lithium battery backup

7.7 Word memory

WM[n] 16-bit variable (0-65535), [n]= 0..511 (0-255 can be used directly in source code, 256-511 needs to be used with Word Pointers (WP).

WM 0-511 are retained (if DIP5=OFF), lithium battery backup

e.g.(1) 'WM10>500' DO2

7.8 Word pointer

WP[n] [n]=0..255 , 16-BIT VARIABL , 0-65535

Usage: e.g. WP4 points to value of WMx, WM4 defines the x.

Word pointers can be used e.g. to make 10 point average sample calculation of analog input with limited programming lines.

Use word pointers only when you master all the other variables.

7.9 Counter

CN[n] Counter, 16-bit value, connected to DI[n], byte, quantity depends on hardware.
DI's are read every 5ms, but updated to user between application program cycles
e.g. Counter CN0 is increased when detecting that DI0 has changed from 0 to 1.
Counter will go automatically to zero if it goes to "65536".
You don't need to define counters in application source code.

0-255 are retained (if DIP5=OFF), lithium battery backup

e.g. 'CN0=1000' "Digital input 0 changed to on 1000 times" 0

Note! You can also use WM as a counter:

```
'DI0' M0
'M0=1' WM10+1
```

7.10 Register output

RO[n] Special function registers , 8-bit (0-255) , n=[0-255]
RO0 Sleep control register

RO2 Connected DI expansion cards,
RO3 Connecter DO expansion cards,
RO4 Connected EXA8/4 expansion cards,

RO9 Sets analog outputs voltage level independently for every AO
Bit.0=AO0 Bit.1=AO1 (0=0..5V 1=0..10V)
e.g. (RO9=3 means AO0 & AO1 output voltage is set to 0…10V)
RO10 EXA8/4 Adr0; output type bits 0-3; I/O addr. 32 –> (0=5V,1=0-10V
RO11 EXA8/4 Adr1; output type bits 0-3; I/O addr. 40 –>

RO12 EXA8/4 Adr2; output type bits 0-3; I/O addr. 48 –>
RO13 EXA8/4 Adr3; output type bits 0-3; I/O addr. 56 –>
RO14 EXA8/4 Adr4; output type bits 0-3; I/O addr. 64 –>
RO15 EXA8/4 Adr5; output type bits 0-3; I/O addr. 72 –>
RO16 EXA8/4 Adr6; output type bits 0-3; I/O addr. 80 –>
RO17 EXA8/4 Adr7; output type bits 0-3; I/O addr. 88 –>

RO30 Power control register (reset once per day, or signal low etc)
bit 0 = 1; reset modem
bit 2 = 1; reset Com1 +5V (pin 1)
bit 3 = 1; reset Com2 +5V (pin 1)
;bit0 can be used e.g. to reset modem once per day or when signal level is too ;low

RO33 Keypad/display type
Unique code for every HMI, can be used also for diagnostics purposes
RO34 System program version (latest is 169 when writing this manual)
RO35 Status of Jumpers/DIP switches 1-6 (DIP1 = bit 0, DIP6=bit 5)
RO36 Power fail info (power fail is active=1) (SET with JP1:1-2=ON=9V /OFF = 17V)

RO37 FTP transfer control & status (1=send FTP (file number RO38), 2=send ok, >128 = error code; some error codes make automatic modem reset.

RO38 File number (0-7 send file, but keep in memory (file is filled as FIFO), 8-15 send and delete after successful status (This is normal method). There are max. 8 FTP files (0-7)

RO39 PLC ID (e.g. 80-89 means GSM-16, current version is 80)
RO40 Flash error (Can happen e.g. when programming PLC and starting up) When GSM-PLC is programmed, program goes to RAM, when started it is stored to Flash memory. Check DIP1 position!

RO45 Flash Manufacturer
RO46 Flash device code
RO54 Function error register
0 = no error
1 = Overflow
2 = Convert error
3 = invalid function number
RO55 Function register
FCN1 = Unsigned integer
FCN2 = Signed integer

RO56 Ser2 modbus timeout (1=10ms, suggested e.g. 50=500ms, for Modbus master)
RO58 Ser2 RTS/CTS control register
Bit0=0 auto RTS control, Bit0=1 No RTS control

RO59 Ser2 modbus error counter
RO60 Exception status

RO64 Wakeup time Hours
RO65 Wakeup time Minutes

RO80 Ser2 modbus Rejected messages counter (Timeouts or other errors)
RO81 Ser2 modbus Accepted messages counter (works also with slave ?)
RO82 Ser2 modbus master queue depth (status)

RO84 Ser2 Modbus conditional message address, 255 sends all (defined in modbus master configuration)

RO88 Ser2 modbus Error slave ID
RO89 Ser2 modbus error type

RO90 Modem state
0 = initializing modem after power on
1 = Modem OK
2 = Modem not in network
3 = Modem power off
4 = Modem not in use

RO92 Modem's driver status

RO93 Pick up time for incoming call
0 - Don't pick up (let it ring)
1 – hang up (can be used as acknowledgement)
2..255 - Hold line open for n seconds)

RO94 Incoming Call type; 1 = Voice, 2 = Data, 4 = Fax (SIM should have support, ask from operator)

RO95 Incoming phone number: last digit

RO96 Incoming phone number: 2nd last digit

RO97 Incoming phone number two last digits (RO97*10+RO96)

RO98 Modem Signal strength (0..31(best)), 0-10 insufficient, >15 sufficient for GPRS,
99 - communication error
100 State after modem initialization (update interval=96s)

RO99 1=iButton found (=iButton reader gets new serial code from iButton device)

RO100-RO107 iButton serial number (8 byte number)

RO128 PID group 1 (defines how many controllers are in this group and the update time. e.g 1 = 1 PID controller, update time =100ms, 5 = 5 PID controllers, update time =500ms, 10 = 8 PID controllers (=max), update time =1s)

RO129 PID group 2

RO130 PID group 3

RO131 PID group 4

RO132 PID pulse delay (for digital output 3-point controlling, 1=10ms)

RO136-RO143 PID (open and close bits for 3-point controllers, e.g. for PID0, open bit = RO136.0 close bit=140.0)

RO204 Controls for 8 LEDs on HMI display unit (same as DO248-DO255)

RO207 Bit information from keys 0-7. (same as DI240-DI247)

RO208 Bit information from keys 8-F (same as DI248-DI255)

RO209 Last character received from the keyboard, (ASCII code)

RO211 PLC status
bit0: 1 = Running, 0 = Stopped
bit1: 1 = Flash error, 0 = Flash OK
bit2: 1 = Flash write protected, 0 = Flash not protected

RO215 Ser2 Mode
0 = modbus slave
1 = modbus master
2 = UCP
3 = TAP

RO219 SER2 DATA FORMAT
0=default=8 bit, parity none
1= 7bit, parity even
2= 7bit, parity odd
3= 8bit, parity even
4= 8bit, parity odd

RO220 Analoque input calibration register (Controls the calibration)
Note! Calibation is normally done with calibration interface.

Error codes #128 - if (x1>x2)

#129 - if (y1>y2)
#130 - DIP switch disables calibration
#132 - gain is too big

RO221 Scaling format register
RO222 Analoque input calibration channel
RO223 Calibration info
RO224 Analoque input Low calibration point Hi
RO225 Analoque input Low calibration point Lo
RO226 Analoque input High calibration point Hi
RO227 Analoque input High calibration point Lo

RO229 Serial channel 2 speed
0 = 300 bit/s
1 = 1200 bit/s
2 = 2400 bit/s
3 = 4800 bit/s
4 = 9600 bit/s
5 = 19.2 kbit/s
6 = 28.8 kbit/s
7 = 38.4 kbit/s
8 = 57.6 kbit/s

RO240 Disable display update. While this register <>0, application program has no access to display and display can be controlled e.g. using SMS messages.

RO241 Power failure counter
RO242 Clock control register
0 = disable clock interrupt
1 = Disable clock driver for setting the time
2 = clock interrupt based on RO64 & RO65
5 = write new time & date to clock (only if RO242=1)
10 = enable clock interrupt

RO243 Modbus slave address (1-254)

RO245 Wildcard digits 1&2
RO246 Wildcard digits 3&4

RO247 Date and time information: month
RO248 Date and time information: day
RO249 Date and time information: day of the week (1-7) (Monday-Sunday)
RO250 Date and time information: hour
RO251 Date and time information: minute
RO252 Date and time information: second
RO255 Date and time information: year (0-255) (offset 2000 years)

RO 0-255 are retained (if DIP5=OFF), lithium battery backup

7.11 Phone Number

N[n] [n]=0…244, Phone numbers are defined in phone book editor, [n]=line number in phone number editor.

 Note! Only program lines in source code editor 0-255 can contain SMS message sending! Lines 256-511 cannot control SMS sending!

N0-244 are always retained, Stored to Flash

Reserved phone number addresses:

N245-N255 cannot be defined in phone book! These variable addresses are not used to store outgoing SMS phone numbers. These variables have special purpose:

N245	Writes the text string to	FTP file 0
N246	Writes the text string to	FTP file 1
N247	Writes the text string to	FTP file 2
N248	Writes the text string to	FTP file 3
N249	Writes the text string to	FTP file 4
N250	Writes the text string to	FTP file 5
N251	Writes the text string to	FTP file 6
N252	Writes the text string to	FTP file 7

N253	Incoming Call number (only available during one program cycle)
N254	Incoming SMS number (only available during one program cycle)

N255	Writes the text string to	HMI Display (I^2C)

Incoming phone number identification is done automatically and it is available during one program cycle.
e.g. ‘N1’ condition is true if incoming phone number (N254) is same as phone book number 1.

7.11.1 Advanced usage of phone numbers (only for experts):

RO95, RO96, RO97 are updated only if is used number comparison in condition field ('T0') and telephone number in phone book has wild cards symbols(?).

For example:

In phone book: N0=+781232292??

In program: 'T0' DO248 ; Led blink
'P1' "%RO95, %RO96, %RO97" 255

When user calls from phone +78123229252, user will see on display: 5,2,52

7.12 Incoming SMS phone number

Incoming phone number is stored in phone book place N254. (This number is only available during one program cycle).

This variable can be used to send SMS-messages by request.
E.g. '($WEATHER)' M0
‘M0=1’ "TEMP = %AI0 C, WIND %AI1 m/s, DIR %AI2 deg" 254

Everyone, who sends request WEATHER, gets current weather information to GSM phone. There are also possibilities to limit access only to those phone numbers that are defined in phone book.

'($WEATHER) & (N0 # N1)' ; $WEATHER message must come from phone numbers ; defined in phone book places 0 or 1.

7.13 Incoming Call type

SIM-card can hold three different phone numbers; Voice call, Data call and Fax number. In GSM-PLC it is possible to detect to what number has been called. Number info is located in RO94.

RO94 = 1 (VOICE)
RO94 = 2 (DATA)
RO94 = 4 (FAX).

This variable can be used to trigger event in GSM-PLC.

E.g. check the incoming phone number & the dialled phone number
#LINE 97: 'T0&RO94=4' "FAX" 255 ; phone number 0 is calling to FAX number
#LINE 98: 'T0&RO94=1' "VOICE" 255
#LINE 99: 'T0&RO94=2' "DATA" 255

Or not to check call type:
#LINE 96: 'T0' "Any call" 255

7.14 Date (DATE)

Date variable can be used in condition field only e.g. DATE=3112.
DATE is also included in RO 248 & RO249.

7.15 Time (CLK)

CLK variable can be used in condition field only e.g. CLK=1200.
CLOCK is also included in RO250 & RO251.

7.16 $ Self defined messages

User is able define own control words.
E.g. '($OPEN DOOR)' DO0=1

Note1: Message sent from GSM-phone must be format **$OPEN DOOR*** (Self defined messages are available only during one program cycle).

Note2: Self defined messages are case sensitive so **$Open Door*** will not work!

Note3: Format **0000 $OPEN DOOR*** also works, if the password is 0000 (=not used).

Note4: **$OPEN DOOR** doesn't work, you need to add asterisk (*) mark in the end of the SMS message.

There can be parameters after control word. Parameters are separated from control word with "=" character. Parameters are located starting from WM0. If parameter is invalid (e.g. $TEMP=3A*), GSM PLC will send SMS-message back with "???" characters.

E.g. SMS-message $TEMP=25* will change the WM0 value to 25.
After message $TEMPS=25=30=40=80* WM0=25, WM1=30, WM2=40 and WM3=80

You can test this with Alarm Log: e.g. (1) $t=123, (2) read WM0

Program example:

Setting room temperature:

'($TEMP)' M0=1 ; Command received.
'M0=1&WM0>10 & WM9<30' M1 WM1=WM0 ; If parameter is OK, set flag and ; save parameter.
'M0=1&!M1' M0=0 "Check parameter" 254 ; Command received but ; parameter is out of range.
'M1=1' M1=0 "TEMP is set to %WM1" 254 ; Send ACK back.

7.17 Pulse variable

Syntax: P1 or P2

P1= pulse every 1 second
P2= pulse every 1 minute

Pulse duration is one program cycle.

7.18 AF-variable (Alarm flag)

For each program line / SMS-message there is AF-bit (alarm flag).
This bit is set when SMS-message with alarm reset time>0 is generated.
This bit is cleared when ACK is received or alarm time has expired.

With this variable it is possible to create cycle alarms.

E.g. Line 0 'DI0S10' M0
Line 1 'M0=1' "PUMP 1 Relay alarm" 0 30
Line 2 'AF1M2' "PUMP 1 Relay alarm" 1 20
Line 3 'AF2M2' "Is anybody out there? PUMP 1 RELAY ALARM!!!" 2 10

When input 0 has been active for 10 seconds, GSM-PLC sends alarm message to phone number 0. If nobody acknowledges this message within 2 minutes, another message is sent to phone number 1. Again if there is no ACK for second alarm message, 3rd message is sent to phone number 2.
After 30 minutes from first alarm message, all alarm flags are cleared.
There will be no new alarm message unless DI0 has been inactive.

Acknowledgement can be done either by clearing corresponding alarm flag or by clearing all alarms. It is also possible to clear alarm flags from application program.

'AF0=0#AF1=0#AF2=0' ; if one of the alarm flags changes from 1 to 0…

Test in Alarm Log: (1) Read AF0

7.19 T[n] phone number for phone call

In condition field this allows triggering one operation per phone number.
GSM-PLC does not answer the call.

E.g. 'T0#T1' DOS5 ; If call comes from phone number 0 or 1, set 5 sec pulse to output.

Don't use any land code (etc. +358…).
In action field this triggers voice call to defined phone number.
See example in 6.17

7.20 CS[n] Call Status

CS0 to CS255 shows the status for outgoing call.

Possible values are:

0 idle (no active call)
1 request to call
2 call in progress
3 OK (destination phone pick up the tube)
4 NO ANSWER (if no hang-up is detected time-out 25 sec)
128 BUSY (if the called party is already in communication)
129 NO CARRIER (call setup failed or remote user release)
130 UNKNOWN

Conditions for CS:
CSxx>, CSxx<, CSxx= with and without hysteresis.

Action for CS:
CSxx=YY with one limitation: CS0 to CS99 can be set only to zero.

The number of the CS has to be similar with the program line, which call to the phone.

Examples:

```
#LINE 60: 'DI0' T10                    ; CALL TO PHONE 10
#LINE 97: 'CS60=1' DO248               ; CALL STATE - REQUEST
#LINE 98: 'CS60=2' DO249               ; CALL STATE - PROGRESS
#LINE 99: 'CS60=4' CS60=0 DO250S5  ; CALL STATE - NO ANSWER Set state to idle
```

7.21 TC[n] Timed control

See also system command TTBL

TC0 to TC255 Table for timed control.

Shows the status of the selected time table

E.g.

#LINE 10: 'TC0' DO1 ;If timetable0 active, set output active

#LINE 11: 'TC1&!DI1' DO1; If TC1 control is active and device is in auto mode, set output active

7.22 PID controllers (Notice! Use AlproWin.exe for PID simulation)

For advanced user only – contact FF-Automation for more information!

7.22.1 Register Variables of Controllers

The GSM-PLC system software includes 32 12-bit direct digital controllers (DDC) with PID characteristics. The controller parameters are held in 16-bit WGx memories and in 8-bit register outputs (RO).
The number of used controllers per group is given in the register outputs (RO128 - RO131). This number also determines the group update interval depending on the number of controllers used.
If one controller in a group is updated, the update time is 100 ms. Similarly, if two controllers in a group are updated, the update time interval is 200 ms. Maximum update interval is 25500 ms, or 25.5 seconds.
If in PLC's application program the update interval is not defined the default update interval of 500 ms is used (RO128 - RO131=5).

	Controller number								RO number
Group 1	0	1	2	3	4	5	6	7	RO128
Group 2	8	9	10	11	12	13	14	15	RO129
Group 3	16	17	18	19	20	21	22	23	RO130
Group 4	24	25	26	27	28	29	30	31	RO131

Example: If it is desired that three controllers update interval is100 ms and two controllers 500 ms, parameters should be set as follows

Group 1 - desired update interval is 100 ms, then RO128 = 1
Group 2 - desired update interval is 100 ms, then RO129 = 1
Group 3 - desired update interval is 100 ms, then RO130 = 1
Group 4 - desired update interval is 500 ms, then RO131 = 5

Controllers operating mode is given in 16-bit variables (WGx 0, 8, 16 etc.). Every controller has the following individual operating modes.

0 Controller not in use
1 Controller is in normal automatic operate mode
2 Controller is in inverted automatic operate mode

3 Controller is in manual operate mode

7.22.2 Three point controllers

There are 8 PID controllers available in GSM-PLC, and every one of these can operate also as a three points controller. The operating parameters are in 16-bit variables WGx 0-255.

Controllers	0 - 1	2 – 3	4 - 5	6 - 7
Control interval Pulse interval Valve closing bit Valve opening bit	RO 128 RO 132 RO 136 (0,1) RO 140 (0,1)	RO 129 RO 133 RO 136 (2,3) RO 140 (2,3)	RO 130 RO 134 RO 136 (4,5) RO 140 (4,5)	RO 131 RO 135 RO 136 (6,7) RO 140 (6,7)

The operating parameters are in 16-bit variables WGx 0-255.

Controller	0	1	2	3	4	5	6	7
Mode	0	8	16	24	32	40	48	56
Actual value	1	9	17	25	33	41	49	57
Set value	2	10	18	26	34	42	50	58
D time/100ms	3	11	19	27	35	43	51	59
I time/100ms	4	12	20	28	36	44	52	60
Gain term	5	13	21	29	37	45	53	61
Output	6	14	22	30	38	46	54	62
Aux. variable	7	15	23	31	39	47	55	63

7.22.3 Control Algorithm

The control algorithm is as follows:

```
DY=   (100/gain)*{e(ti)-(e(ti-1)) +          ; P term
      e(tI)/integration time constant+       ; I term
      Derivation time*(e(ti)-2e(ti-1)+e(ti-2)} ; D term
```

The D term controls how strongly the control process reacts to rapid changes in the input signal. The differentiation time constant is an 8-bit parameter and is given in hundreds of milliseconds (value 1 - 255 equals 0.1-25.5s). The larger time and the greater change to the output signal.
If the differentiation time constant is given as 0, the term is not used.

The I term controls how quickly the process reacts to stabilize offset error. The integration time constant is a 16-bit parameter and is given in hundreds of milliseconds (0.1-6553.5s). The larger integration time constant the slower process reacts to offset errors. If the integration time constant is given as 0, the term is not used.

The P term controls the gain such that: P=100/gain, where gain can be from 0.01 to 100.

Therefore	if gain = 0	P - TERM NOT USED
	if gain = 5	P = 20
	if gain = 0.5	P = 200
	if gain = 0.02	P = 5000
	if gain = 0.01	P = 10000

The following figures describe the effect of controllers differents parameters.

Effect of the gain P

P is a value 1 - 10000 corresponding to gains of 0.01 - 100.0 .
(200 correspods to a gain of 0,5.)

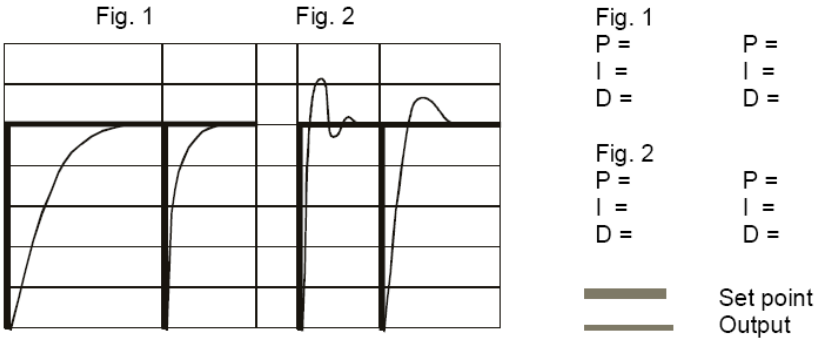
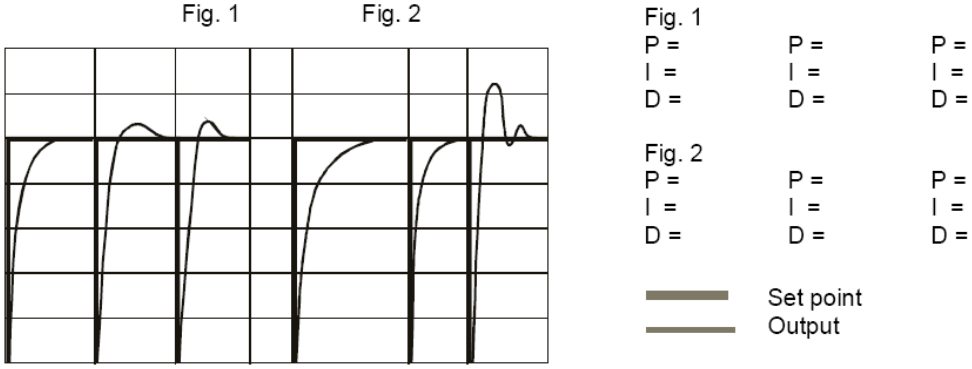


Figure 1 shows how increasing the gain yields faster control action.

Figure 2 shows how reducing the gain stabilises the oscillation. caused by a short integration time.

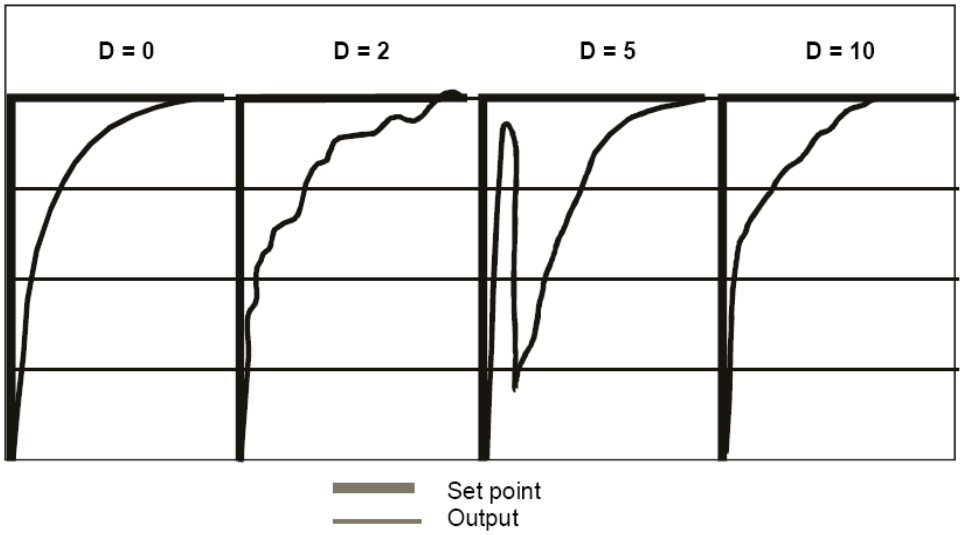
The sum of the proportional, derivative and integral terms is multiplied by the gain, P. Thus the gain, P, strengthens or weakens the effects of these coefficients.



The figures show the behaviour of the same process for two different values of gain, P and varying integration time, I. It can be seen that using high gain will result in overshoot even with longer integration times. If the integration time, I is too short, an oscillation of the type shown in the right-hand figure will occur.

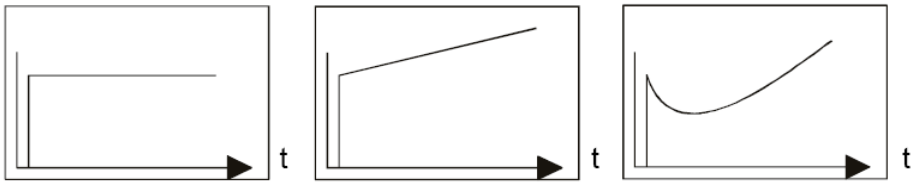
Effect of the derivative time constant, D

The effect of the derivative time is critical, as illustrated in the figures below. Derivative control is unsuitable for many types of systems, or is not necessary at all.



8.3 Controller types

The figure below shows the step response of the three basic controller types.



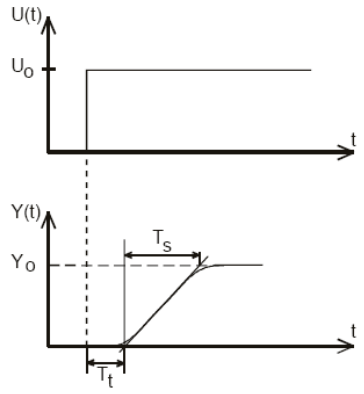
- The P controller has constant gain, and the result is unsatisfactory, especially if the gain is low.
- The PI controller changes the apparent gain and correscts the error in time.
- The PID controller exaggerates variations of error in order to abtain rapid error correction. The step responce becomes faster.

8.4 Controller tuning

In a control system suitable values must be found for the control parameters P,I and D. Suitable parameters can be determined using mathematics or a Bode diagram, however, these methods can be time consuming. Controllers are usually tuned on the basis of experimental data from the control system.
Two simple and effective methods for this are described below:

- Step response method:

The delay and rise time of the step response of the process are determined, and the controller settings are calculated on that basis.



P Controller: $U_P = \frac{T_t U_0}{T_s Y_0} = c$

PI Controller : $U_P = 1.25\ c$
$T_I = 3\ T_t$

PID Controller : $U_P = 1.25\ c$
$T_I = 3\ T_t$
$T_D = 0.42\ T_t$

- Oscillation method:

The gain and oscillation cycle time at the point of oscillation are determined, and the controller settings are calculated on that basis.

P Controller: $K_P = 0.5\ K_{Pcr}$

PI Controller: $K_P = 0.455\ K_{Pcr}$
$T_I = 0.85 T_{cr}$

PID Controller: $K_P = 0.6\ K_{Pcr}$
$T_I = 0.5 T_{cr}$
$T_D = 0.12 T_{cr}$

$K_P = \frac{1}{U_P} = \text{gain}$

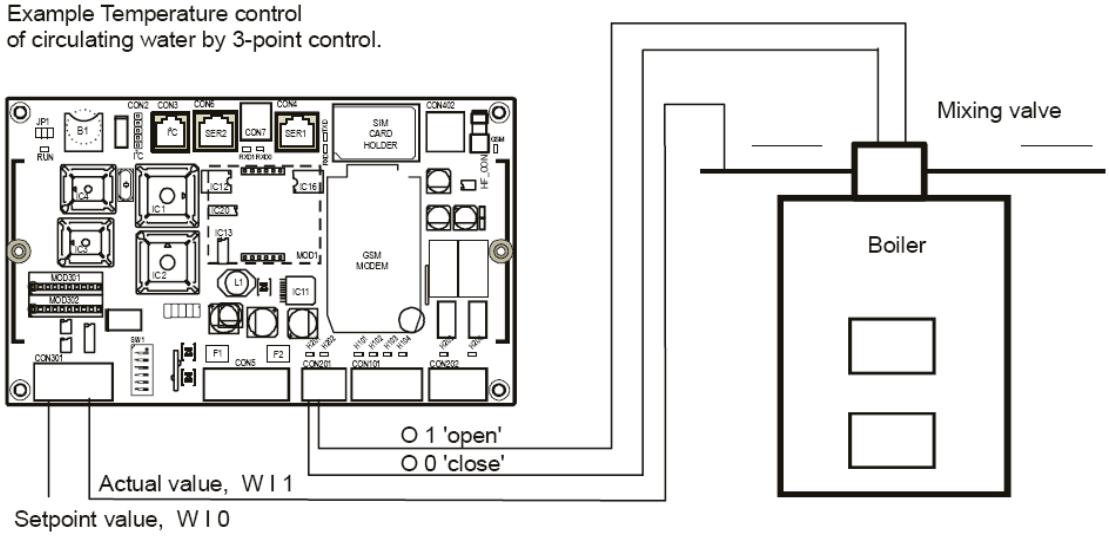
T_I = Integration time constant
T_D = Differentiation time constant
K_{Pcr} = Critical gain at which the process oscillates
T_{cr} = Cycle time of process oscillation

The above two methods yield reasonable starting points for the controller parameters which can be further refined during operation. The ALPro software can be used for monitoring controller variables, building trend data and producing producing documentation

A typical application, controlling the temperature of the circulating water in a central heating system, is shown below.

The mixing valve is controlled by a pulse driven actuator motor. The dead time in an application of this kind can be several seconds., as changes in the water temperature will not immediately be detected by the sensor. It is, therefore, futile to give further drive pulses to the valve before the true effect of the previous action is known. This dead time can be accounted for when tuning the controller using the step response method.

Example Temperature control
of circulating water by 3-point control.



Logic program for water heating:

```
STR   R    C    010    ; Control interval is defined to
EQ    R    O    128    ; 1 second (10*100 ms)
STR   W    C    050    ; Integration time (50 * 0.1s) ( 0 - 65535)
EQ    W    GM   012    ; greater value gives slower action
STR   W    C    002    ; Differentiation time (2 * 0.1s) ( 0 - 255)
EQ    W    GM   011    ; greater value gives bigger steps
STR   W    C    500    ; Gain P = (100/500)  P=0.2
EQ    W    GM   013    ; greater value gives slower action
STR   W    I    001    ; Set point, e.g. from potentiometer
EQ    W    GM   010
STR   W    I    000    ; Actual value from temperature sensor
EQ    W    GM   009
STR   W    C    001    ; Controller in automatic mode
EQ    W    GM   008
STR   R    C    010    ; Pulse interval 1s
EQ    R    O    132
STR   R    O    140    ; Read 'open' bits
BIT        M    020    ; Convert to bits (controllers 0 - 7)
STR        M    021    ; Use bit for controller 1
EQ         O    001    ; to open valve
STR   R    O    136    ; Read 'close' bits
BIT        M    020    ; Convert to bits
STR        M    021    ; Use bit for controller 1
EQ         O    002    ; to close valve
STOP
```

Above logic program is just an example from AlproWin programming software
(It is not for GSM-PLCs)

8 Programming Syntax:

AutoLog GSM-PLC's are programmed using AutoLog GsmProgrammer software.

To open the application program editor click View->Source code

Each programming line has the following possible syntaxes:

['Condition'] [Action(s)]

['Condition'] ["Message"] [Phone_number_id] [Optional: Acknowledge_time_minutes]

['Condition'] [Action(s)] ["Message"] [Phone_number_id] [Optional: Acknowledge_time_minutes]

GSM-PLC has max 512 programming lines.

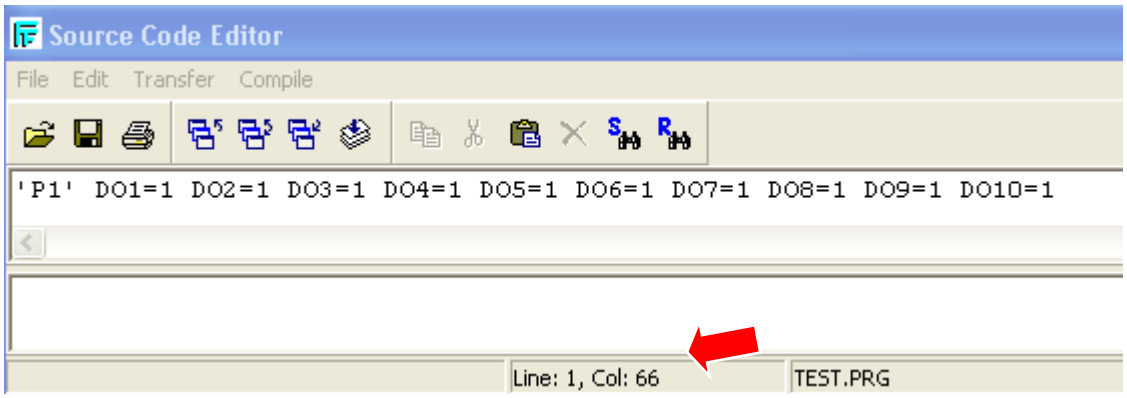
One programming line is max.140 characters long. (the length of SMS message).

There can be several commands in one SMS-message. Commands are executed from left to right. Phone number is max. 16 characters.

SMS-message must not end in the middle of the command.

Examples:

```
'P1' DO1=1 DO2=1 DO3=1 DO4=1 DO5=1 DO6=1 DO7=1 DO8=1 DO9=1 DO10=1
DO11=1 DO12=1 DO13=1 DO14=1 DO15=1 DO16=1 DO17=1 DO18=1 WM1=100
WM200=1000

;Above line has 140 characters including spaces

'P1'DO1=1DO2=1DO3=1DO4=1DO5=1DO6=1DO7=1DO8=1DO9=1DO10=1DO11=1DO12=1DO1
3=1DO14=1DO15=1DO16=1DO17=1DO18=1WM1=100WM200=1000WM201=1000WM202=100

;Above line has 139 characters. Program line works also without
spaces, but it will be hard to read. Anyway it gives you ability to
put some extra commands in the line.

;Use spaces always when it's possible to make code more readable!
```



Line length (Col:66) can be seen in the status bar.

8.1 'Condition' -field

8.1.1 Condition (IF sentence)

Format: ' condition '

Variables:

DIn ; input
DOn ; output
Mn : memory
AIn ; analogue input
CNn ; counter
WMn ; word memory
Aon ; analogue output
WPn ; word pointer to WM area
ROn ; special function registers
Nn : SMS phone number
Tn ; clip phone number (without land code)
Pn ; pulse n=0:1sec,n=2:1min.
; pulse variable is active one program cycle every second/minute
AFn ; alarm ACK info
CSn ; call status
TCn ;Time control table

Operands:

& AND
# OR
X XOR
! NOT
<, =, > smaller,equal,bigger
( ) brackets
$ compare incoming SMS-message

Each programming line contains IF… THEN structure, where IF part is defined between ' ' – characters. This is same as condition.

e.g. 'AI0>WM0' M10 "ALARM" 0, where IF sentence is 'AI0>WM0'

'M1' M1=0 M0=0 "YOU PRESS: %WM0@3 and the status of the digital input 0 is %DI0" 0

You can also add several conditions inside IF sentence

'DI0&DI1#DI2'
'DI0&(DI1#DI2)'
'(AI0>500)&(AI1>600)'

8.1.2 Variables and operands in 'Condition' field

	WM & AO	CN	AI	RO	M	WP	CS	AF	Constant
WM	=,<,>,&,#	=,<,>,&,#	=,<,>,&,#	=,<,>,&,#	=,<,>,&,#	=,<,>,&,#	&,#	&,#	=,<,>
CN	=,<,>,&,#	=,<,>,&,#	=,<,>,&,#	=,<,>,&,#	=,<,>,&,#	=,<,>,&,#	&,#	&,#	=,<,>
AI	=,<,>,&,#	=,<,>,&,#	=,<,>,&,#	=,<,>,&,#	=,<,>,&,#	=,<,>,&,#	&,#	&,#	=,<,>
RO	&,#	&,#	&,#	&,#	&,#	&,#	&,#	&,#	=,<,>
M	&,#	&,#	&,#	&,#	&,#	&,#	&,#	&,#	=
WP	=,<,>,&,#	=,<,>,&,#	=,<,>,&,#	=,<,>,&,#	=,<,>,&,#	=,<,>,&,#	&,#	&,#	=,<,>
CS	&,#	&,#	&,#	&,#	&,#	&,#	&,#	&,#	=,<,>
AF	&,#	&,#	&,#	&,#	&,#	&,#	&,#	&,#	=

8.1.3 Using Derivation in condition field

Derivative operation is done using "=" character in condition

Examples:

'DI1=1' ; positive derivation (change from 0 to 1)
'M9=0' ; negative derivation (change from 1 to 0)

```
'DI0' DO0=1          ;When DI0 changes to 1, DO0 changes to 1
                     ;When DI0 changes to 0, DO0 stays 1
                     ;Don't need derivation

'DI0=1' DO0=1        ;When DI0 changes to 1, DO0 changes to 1
                     ;When DI0 changes to 0, DO0 stays 1
                     ;Changes only when positive derivation in DI0

'DI0=1' DO0          ;When DI0=0, DO0 changes to 0
                     ;When DI0 changes to 1, DO0 changes to 1 for one
                     ;program cycle only.
                     ;This happens because the positive derivate is true
                     ;only for one program cycle

'DI0' DO0            ;When DI0=1 then DO0=1 and when DI0=0 then DO0=0
                     ;Don't need derivation

'!DI0' DO0           ;When DI0=0 then DO0=1 and when DI0=1 then DO0=0
                     ;Don't need derivation

You can also use derivation as the following:

'DI2S1' M2  ; S1 means DI needs to be 1 second ON, after M2 goes to 1.
'M2=1' WM30+1
```

8.1.4 Using delay in condition field

S	seconds	; S5 = delay of 4 to 5 seconds, Max 255
M	minutes	; M5 = delay of 4 to 5 min

Syntax: '[Variable][S/M][0-255]'

Examples:

```
'DI0S5' DO0          ;When DI0 has been 1 for 5 seconds, DO0 turns to 1
                     ;When DI0 turn to 0, DO0 turns to 0 immediately
                     ;Delay works only in ON direction

'(DI0)S5' DO0        ;Same effect as 'DI0S5' DO0

'!DI0S5' DO0         ;When DI0 has been 0 for 5 seconds, DO0 turns to 1
                     ;When DI0 turn to 0, DO0 turns to 0 immediately
                     ;This Delay works only in ON direction

'(DI0=1)S5' DO0      ;This syntax doesn't work, because derivative cannot
                     ;the active for 5 seconds
```

```
'(DI0&M0)S5' DO0  ;When DI0 & M0 has been 1 for 5 seconds  DO0 turns
                  ;to 1
                  ;When DI0 or M0 turns to 0, DO0 turns to 0
                  immediately
                  ;This Delay works only in ON direction

‘DI0’ DO0S5       ;This delay works only in OFF direction
                  ;When DI0 turns to 0, DO0 turns to 0 after 5 secs

‘DI0S5’ DO0S5     ;This is not allowed syntax

'DI0S5' DO0=1     ;To make two direction delay you need to write 2
'!DI0S5' DO0=0    ;lines

'(WM0>400)S5' DO0 ;When WM0 has been over 400 for 5 secs DO0 turns to
                  ;1

‘DI0s120’         ;true when DI0 has been 1 for 120 seconds

‘DI0M2’           ;true when DI0 has been 1 for 2 minutes
```

You can also use derivation and delay as the following:

e.g. Calculating operating times:

```
'DI2S1' M2  ; S1 means DI needs to be 1 second ON, after M2 goes to 1.
'M2=1' WM30+1
```

or calculating total and daily operating times

```
'DI2S1' M2
'M2=1' WM30+1 WM32+1
'WM32>9999' WM33+1 WM32=0

'CLK=0700' M80
'M80=1' WM31=WM30 WM30=0 ;resets daily counter and moves it to other
WM31 which can be send using sms.
```

sending sms

```
'M80=1 "%WM31,%WM32 %WM33” 0
; send sms to phone number index 0 (in phone phone book)
```

8.1.5 Other examples how to use condition -field:

8.1.5.1 Counter input:

```
‘DI0S1’ WM0+1     ;This syntax doesn’t work(!) because condition is
always true if DI0=1 and time >1 second and the counter WM0 will
increase every programming cycle.

‘DI0S1’ M0        ;This syntax works because it increases WM0 only
‘M0=1’ WM0+1      ;when M0 becomes true (positive edge).
```

8.1.5.2 Clock triggered event:

```
‘CLK=0800’ WM1=WM0 WM0=0 ;This syntax doesn’t work(!) because the
condition is true for 1 minute so both WM1 and WM0 becomes zero.
```

```
'CLK=0800' M20    ;This syntax works because it increases WM0 only
M20=1 WM1=WM0 WM0=0 ;when M20 becomes true (positive edge).
```

8.1.5.3 Operating time counter:

```
'DI3&P1' WM4+1          ;counting operating time (DI3=1) seconds
'WM4>59' WM5+1 WM4=0    ;counting operating time (DI3=1) minutes
'WM5>59' WM6+1 WM5=0    ;counting operating time (DI3=1) hours
```

8.1.5.4 Initializations:

```
'M100#!M100'      ;this condition is always true so you can set fixed
           values to variables after this condition

'M50=0'M50=1      ;this condition is true only for the first program
                  cycle after booting the GSM-PLC
```

```
'M50#!M50' WM50=AI0 WM51=AI1 WM52=AI2 WM53=AI3 WM54=AI4 ;moving the
raw values of analog inputs to WMs. In Report message you  cannot use
AI values if you want to send RAW values.
```

8.1.5.5 Modem reset:

```
;works only with build-in gsm modem
;-----------------------------------
'CLK=2359' M90
'M90=1' RO30=1 ;resets gsm modem, system program sets it back to 0
           automatically
```

```
; Resets external gsm modem through DO0 clock 23:59
;-------------------------------------------------------------
'CLK=2359' M90
'M90=1' DO0=1 ;resets gsm modem
'DO0S10' DO0=0

;normally modem reset is not needed, but if the gsm modem jams it can
solve problem.
```

8.1.5.6 Setting binary variables(!):

'M100#!M100' M0=M10 M1=M11   ;This syntax doesn't work correctly(!)

You need to use syntax:

'M10' M0
'M11' M1

also do not use syntax M7=DI1

'DI1' M7

for every variable you need to use own programming line(!)

8.1.5.7 Packing binary variables (M) to Word Memory (WM)

```
'M100#!M100' WM100=M100        ;Packs M100,M101,M102…M115 to WM100 so
that M100= bit 0, M101 = bit1 … M115=bit15

'M100#!M100' WM100=DI0         ;Packs DI0-DI15 to WM100 so that DI0=
bit 0, DI1 = bit1 … DI15=bit15
```

8.1.5.8 Conditions for sending SMS:

```
;If M0 or M1 or M21 changes its state, send message to phone number 0.
'M0=0 # M0=1 # M1=0 # M1=1 # M21=0 # M21=1' "D0,,%M0,%M1,%M21" 0

'CLK=0800' M80
'M80=1&WM60<20'WM60+1 "D9,,0,%WM99" 0
; sends sms when clock = 8:00
; safety limit 20 for max sms / day
```

8.1.5.9 Others

```
'DI0S5'            ;if DI0 has been true for 5 seconds

'N0#N1'            ;if incoming phone number is same as N0 or N1

'DI0=1'            ;if digital input 0 has changed to 1 (positive
                   ;derivation)

'$WEATHER'         ;if incoming SMS = $WEATHER then…

'RO247=7&CLK=1200';Every Sunday at 12 o'clock

'P1&DI0S2'         ;Condition for time counter. Counting activates
                   ;only if input has been active for two seconds.

'AI2S50>400'       ;S50 defines delay of fifty seconds

'WP0<AI1'          ;Value in address defined by WM0 is less than
                   ;analogue input 1.
```

```
'($REPORT1)' M65
'M65=1'
"KK1=%WM11,KK2=%WM31,KA1=%WM22h,KA2=%WM42h,PK=%AI0m,YR=%M8,LR1=%M3,LR2
=%M4,KH=%M11,LVP=%M9,TVP=%M10,LT=%AI7C,G=%RO98" 254

; if incoming sms is $REPORT1, then sends report sms as feedback
; RO98 is GSM signal strength

Look $ Self defined messages!
```

```
'($P1)' M61
'M61=1&WM0<10000' WM12=WM0 WM13=0 WM22=WM1  "Set: Pump1 operating
count = %WM12 Pump1 operating time = %WM22 hours" 254
'M61=1&WM0>9999' "SET failed!: Pump1=operating count maximum is 9999"
254

; if incoming sms is $P1=10=123 then sets pump operating count to 10
and operating time to 123, send feedback sms

Look $ Self defined messages!
```

8.2 Operation(s) -field

['Condition'] **[Action(s)]**

['Condition'] **["Message"] [Phone_number_id] [Optional: Acknowledge_time_minutes]**

['Condition'] **[Action(s)] ["Message"] [Phone_number_id] [Optional: Acknowledge_time_minutes]**

Operation-field contains variable control –Action(s) or SMS-message sending or both.

Note! Messages can be send only from programming lines 0-255. You cannot send message using programming line 256-511.

8.2.1 Action – field format:

Var
- DOn ; digital output
- CNn ; counter
- WMn ; word memory
- AOn ; analogue output, can be used as WM's
- ROn ; special function registers
- Mn ; memory
- AFn ; alarm flag
- WPx ; indexed memory
- @n ; Define array in SMS message.
- Tn ; voice call (see also variable CS)
- CSn ; call status

+ plus (with use of WM's and AO's)
- minus (with use of WM's and AO's)
. multiplication (with use of WM's and AO's) (. = dot)
/ divide (with use of WM's and AO's)

8.2.2 Variables and operands in "action" field

	WM	CN	AI	RO	M	WP	DI	CS	AF	Constant
WM	=,+,-,.,/	=,+,-,.,/	=,+,-,.,/	=,+,-,.,/	=,+,-,.,/	=,+,-,.,/	=			=,+,-,.,/
CN	=,+,-	=,+,-	=,+,-	=,+,-	=,+,-	=,+,-				=,+,-
AI										
RO	=	=	=	=	=	=				=
M	=	=	=	=	=	=	=			=
WP	=,+,-,.,/	=,+,-,.,/	=,+,-,.,/	=,+,-,.,/	=,+,-,.,/	=,+,-,.,/				=,+,-,.,/
CS										=
AF										=

8.2.2.1 Variable control examples:

```
DO1S30              ; 30 seconds pulse to output 1
CN0=0               ; Reset counter 0
AF0=0               ; Reset alarm flag 0 from program
```

There is no overflow information for plus and minus operations.
Word variables are 16 bit long so the maximum is 65535!

```
WM0+WM10            ; Add variable WM10 to variable WM0. Save result to
                    ; WM0.
WM0+CN3             ; Add counter 3 value to word memory 0.
WP4+WM10            ; Add WM10 to value in address defined by WM4.
```

8.2.3 Variables control examples for WM and RO:

RO (Register Outputs) are 8-bit variables. WM (Word Memories) are 16-bit variables. RO cannot be “copied” directly to WM. Look examples bellow:

Syntax: `WM[x]=RO[y]  ; WM[x]=RO[y]*256 + RO[y+1]`

e.g.

```
WM102=RO97          ; WM102 = RO97 * 256 + RO98          (in practise)
```

8.2.4 Variable control examples for WM and M:

M is 1-bit binary memory and WM is 16-bit word memory.

```
WM100=M0            ; WM100 bit0 = M0, WM100 bit1= M1 … WM100 bit 15=M15
```

8.2.5 Variable Multiply and division examples for WM:

If you are going to do multiply or division operations, you need to allocate two WMs.

Multiply: WMx . Var16 -> WM[x] bits 15..00, WM[x-1] bits 32..16

Division: WMx / Var16 -> WM[x] = integer, WM[x-1] = remainder

;Var16 means any 16-bit variable (e.g. WM, AI, AO, constant)

Note! Remainder = Modulo(WMx/Var16)
e.g. if WMx= 5, Var16=3, Result = 1, Remainder = 2
In other words you cannot put result as 1.2 if you want to have real result you need to multiply remainder e.g. by 1000 and the divide it again with Var16 (3) to get the decimal part e.g. to HMI display.

Normally it is suggested to make multiply and division operations in SCADA system and only use RAW integer values in GSM-PLC as much as possible.

Examples:

WM10 need to be multiplied by 40 once per second.

```
'P1'  WM255=WM10 WM255.40 ; WM254 is reserved for overflow
```

Note! For multiply operation you might need to use several WMs. Result is stored in WM255 (low word) and WM254 (high word), result is 65536*WM254 + WM255

```
'P1'  WM10.40 WM255=WM10 WM254=WM9

Above syntax doesn't work, because for every pulse it will multiply
the result.

'P1'  WM255=WM10 WM255/40 ; WM254 is reserved for remainder

Note! For division operation you might need to use several WMs. Result
is stored in WM252 (low word) and WM253 (high word), result can be
calculated in SCADA using formula 65536*WM255 +
((100000*WM254/40)/100000)
```

Note!

Analog inputs are automatically scaled to engineering units when those are send over SMS.

Message = "Analog input = %AI0" returns something like "Analog input = 1.23"

Also when analog inputs are send to display (I2C) it will automatically scale the value to engineering units.

Scalings for analog inputs are defined using special registers – Look analog input calibration chapter!)

8.2.6 Message –field format

['Condition'] **["Message"] [Phone_number_id] [Optional: Acknowledge_time_minutes]**

['Condition'] [Action(s)] **["Message"] [Phone_number_id] [Optional: Acknowledge_time_minutes]**

Note! Messages can be send only from programming lines 0-255. You cannot send message using programming line 256-511.

8.2.7 SMS message syntax

SMS message supports almost all 7-bit ASCII characters, but no control characters (000-031) or (127) or Semi-colon ; (059)

http://www.sciencebuddies.org/science-fair-projects/project_ideas/CompSci_ASCII_Table.shtml

Note that e.g. Å,Ä,Ö,å,ä,ö,€,£, ¤, §, ½ , ¨ are not included in 7-bit ASCII character set.

In such case GsmProgrammer gives information window: **Programs has errors**

Semi-colon ; mark is reserved for remarking the program line, so it cannot be used in text message!

```
'P1' "!#$%&'()*+,./0123456789:<=>?@ABCDEFGHIJKLMNOPQRTSUVW
XYZ[\]^_`abcxyz{|}~" 0
```

For example above characters can be used.

8.2.8 Inserting variable values into SMS message:

Syntax: %[Variable] (optional [@additional_arrays])

%-variable will insert defined variable value into SMS-message, e.g. %CN0 will insert value of CN0 into text.

examples of message -field:

“Report: %WM10 hours, %AI0 m/s, Status=%M10”

"Array: %WM4@7" 1 Send 8 variables starting from WM4. (WM4, WM5, WM6, WM7, WM8, WM9, WM10, WM11)

8.2.9 Phone_number_ID

[N0-N2444] Index number to GSM phone number table.

N245	Writes the text string to	FTP file 0
N246	Writes the text string to	FTP file 1
N247	Writes the text string to	FTP file 2
N248	Writes the text string to	FTP file 3
N249	Writes the text string to	FTP file 4
N250	Writes the text string to	FTP file 5
N251	Writes the text string to	FTP file 6
N252	Writes the text string to	FTP file 7

N253 Incoming Call number (only available during one program cycle)
N254 Incoming SMS number (only available during one program cycle)

N255 Writes the text string to HMI Display (I^2C)

8.2.10 Optional: Acknowledge_time_minutes

Alarm reset time. Any number other than zero defines that message requires acknowledgement.
If ACK is not received, alarm reset time defines the time in minutes when system resets alarm flag (AF).

If "alarm reset time" is not defined, no ACK is required.

Syntax : AF[n]

n = programming line number

8.2.11 SMS message examples:

”Tank level HIGH” 1 ; Send text ”Tank level HIGH” to phone number ; [1], doesn’t wait for acknowledgement.

”Burglar” 1 10 ; Send text ”burglar” to phone number [1] and wait for acknowledgement, new alarm can be triggered to next phone number e.g. after 2 minutes if alarm flag has not been cleared (‘AF[n]S120’ “2nd try burglar” 2)

”Customers so far %CN0” 1 ; Send counter value to phone number[1].

"Array: %WM4@7" 1 ; Send 8 variables starting from WM4 to phone number 1.

For each program line, there is a AF (alarm flag) bit. This bit is set when a SMS-message that requires an acknowledgement, is sent (alarm reset time>0). GSM-PLC will not send new alarm from this program line until corresponding AF bit is cleared, either by sending ACK to GSM-PLC or alarm reset time expires.

syntax: ACK [n] (n=programming line) acknowledges AF[n]

or

ACK ALL = acknowledges all alarm flags

See also command ACK.

9 SMS Programming Protocol for GSM-20/16/8/4/GW

This chapter describes commands which can be send using GSM phone / modem to GSM-PLC or using Alarm Log.

These commands cannot be used in program lines.

9.1 Specification

- One SMS-message is max.120 characters long. There can be several commands in one SMS-message. Phone number is max. 16 characters.
- SMS-message must not end in the middle of command.
- When a command line is sent to GSM-PLC, it checks the incoming message format. If GSM-PLC found an error from message, it will send back the whole message, marking the error with “???”
- GSM-PLC stops program execution after receiving program modify command (INIT, DEL and NUM=).
- GSM-PLC must be started with RUN instruction after program editing. Total amount of variables is defined by the hardware.
- Variable type ALL can be used, when user wants to point all inputs/outputs or symbolic links. e.g. NUM ALL? Sends all phone numbers
- Start of SMS-message to GSM-PLC starts always with password and ends to *-character.
- Program line can contain condition (IF-sentence)) and multiple actions (THEN) but only one SMS-message sending and one timer.
- There can be only one timer function / program line.
- Action can be either SMS-message, PLC-operation or both
- Field separation mark between IF & THEN sentences is a SPACE, multiple spaces or COMMA(‘,’). Text-field is separated with “ “
- Condition field is separated with ‘ ‘ (ASCII CODE??)
- Response to query made to GSM-PLC starts always with #-character.
- For alarm messages send from GSM PLC #-character before line number in the beginning of message means that ACKNOWLEDGEMENT is needed.
- GSM-PLC will automatically initialise GSM-modem after power-up.
- Length of password is always 4 digits.
- Commands may be as full word or first letter:
  NUM = N
  INIT = I
- PSW = PSW because we also have PIN instruction (so we can not use P for both).

9.2 CONFIGURATION COMMANDS

9.2.1 GSM number (NUM)

Define GSM-number **NOTICE! This command stops the application program execution! PLC must be started with RUN-command**
Delete GSM-number
Read GSM-number

Format:

```
[password] NUM [i] = [gsm_number]*    ; define GSM number
[password] NUM [i] =*                 ; delete GSM number
[password] NUM [i] =?*                ; read GSM number
```

NUM	FCN
i	ID number
zzzzzzz	GSM number
?	Request
*	End of message
ALL	All

Examples:

NOTE If you use GSM PROGRAMMER password and end of line char ('*') is not needed

```
Query:      1234 NUM 1=+358953063153*  ; define phone number[1]
Response:   #NUM 1=+358953063153

Query:      1234 NUM ALL=*                    ; remove all phone numbers
Response:   #ALL NUM DELETED

Query:      1234 NUM 0?*              ; read phone number[0]
Response:   #NUM 0=+358953063153

Query:      1234 NUM A?*              ; ask all phone numbers from PLC
Response:   #NUM A=empty              ; if no phone numbers or
            #NUM 0=+358053063153 NUM 1=+358953063154*
```

9.2.2 BTN

- Define Ibutton serial number
- Delete Ibutton serial number
- Read Ibutton serial number

Format:

```
[password]  BTN [i]=zzzzzzz*    ; Define iButton number
[password]  BTN [i]=*           ; Delete iButton number
[password]  BTN [i]?*           ; Read iButton number
```

Password	
BTN	FCN
i	ID number
zzzzzzz	Ibutton serial number
?	Request
*	End of message
ALL	All

Examples:

```
Query:      1234 BTN 1=0800446A *     ; Define Serial number[1]
Response:   #BTN1=0800446A

Query:      1234 BTN ALL=*                  ; Remove all Serial numbers
Response:   #ALL NUM DELETED
```

Query: 1234 BTN 0?* ; Read Serial number[0]
Response: #BTN0 =0800446A

Query: 1234 BTN ALL?* ; Ask all Serial numbers from PLC
Response: #NUM A=empty ; If no iButton defined or
#NUM 0=0800446A NUM 1=045D6A88*

9.2.3 Password (PSW)

- Define password

Format: [password] PSW MMMM MMMM*

PSW function
MMMM new password

Examples:

Query: 1234 PSW 4321 4321*
Response: # PSW OK ; password changed successfully
PSW BAD ; if password didn't change!

9.2.4 PIN-Code for GSM modem (PIN)

- Define PIN-code

Format: [password] PIN NNNN NNNN*

PIN function
NNNN new PIN-code

Examples:

Query: 4321 PIN 3322 3322*
Response: # PIN OK ; PIN-code changed successfully
PIN BAD ; mistake in changing PIN code!

9.2.5 GPRS settings (FTP)

- Defines the GPRS system provider settings to MODEM

Format:
GPRS=M,P,[aaa.aaa.aaa.aaa],[bbb.bbb.bbb.bbb],[username],[Password],
["at+cgdcont=1,"IP","internet"]

[ ] - field may be empty

M - PPP Authentication Method
1 Use PAP authentication.
2 Use CHAP authentication.

P – Phone number:
1 - "*99***1#"
2 - "*99#"

aaa.aaa.aaa.aaa – DNS1
bbb.bbb.bbb.bbb – DNS2

Examples:

SONERA & Elisa in Finland uses following GPRS settings
APN= “internet", no password/username required

So command in whole for Sonera SIM card:

GPRS =1,1,,,,, internet

9.2.6 FTP settings (FTP)

Note! For this feature you need WaveCom GPRS modem with TCP/IP stack support

- Define FTP settings

Format: [password] FTP=n,IP,Path,PLC ID,<Password>

n	number of files (1, 2, 4, 8 ;1x256k, 2x128k, 4x64k, 8x32k)
IP	Server IP address xxx.xxx.xxx.xxx
Path	Path to server to store files (128 char max.)
PLC ID	8 Hex symbols
Password	Password for login. If password doesn’t exist, “anonymous” login will be used (98 char max.)

Examples:

Set FTP parameters: 4321 FTP<sp>=8,123.123.123.123,\,00112233,gruyere*

Response: #FTP=8,123.123.123.123,\,00112233,gruyere

Read FTP parameters: 4321 FTP ?
Response: #FTP=8,123.123.123.123,\ffa,00112233,gruyere
or #FTP = EMPTY

Delete FTP parameters: 4321 FTP =
Response: #FTP=EMPTY

9.2.7 Launching file transfer over GPRS using FTP protocol

Triggering file transfer happens through special function registers located in Register output area
There are two RO’s used to control FTP file transfer

RO37 = FTP command / Status register
RO38 = File number

Command RO37 = 1; - Request to send file defined in RO38

RO37 status:

2	file succesfully sent
128	invalid file number
129	file not used in application program
130	file empty
131	iChip not connected
132	can’t open FTP session
133	wrong path
134	can’t store file
135	can’t delete File
136	can’t send File
137	can’t close File
139	modem doesn’t support FTP

RO38 – file number:

0…7 ;Keep file in memory after transmission
or
8…15 ;Delete file after successful transmission

9.2.8 Time table definition (TTBL)

(Using time tables is not suggested)

- Time table definition command

Format: [password] TTBL [n]=m,s,t,d

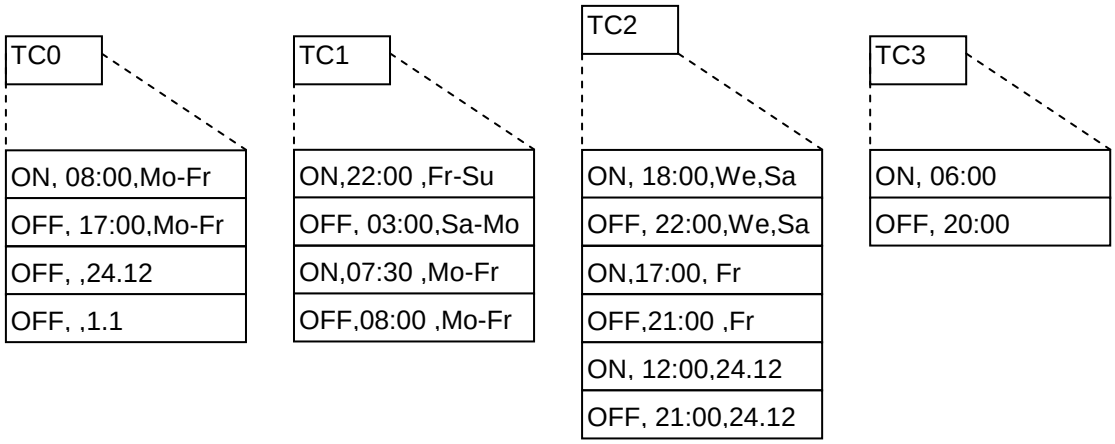
n	Line number 0..255
m	TC variable address (0..127)
s	Status 0-OFF, 1-ON
t	Time hh:mm 00:00 to 23:45 (Table is checked every 15 minutes)
d	Date/weekday dd.mm/1(Monday)-7(Saturday). Can be left empty. Conditions for weekday: 1-3 = from Monday to Wednesday 2- = Tuesday only 2,4 = Tuesday and Thursday A=B is invalid 1-7, 4-1 are invalid definitions

Either t, d or both must be defined in control line

There can be tot. 256 definition lines for ON & OFF controls for different TC variables. These lines can be freely used for as many as 256 TC variables.
The line order for TC is important. User should define first normal ON and OFF controls and after that the exceptions. The last active line defines the state of the TC.

E.g
TTBL 0= 5,1,08:30,1-5 ;TC5 is set to ON at 8:30 am from Monday to Friday
TTBL 1= 5,0,16:30,1-5 ;TC5 is set to OFF at 16:30 from Monday to Friday
TTBL 2= 5,0,00:00,24.12 ;TC5 is set to OFF at 24.12 (if this line is the 1st line of the TC5 and 24.12 is Wednesday, lines to follow would over write this control)

Time control function is executed every 15 minutes.
Note! If PLC is not running when TTBL line should trigger TC to be active, control will not take place after power returns.



Examples:

Query:	1234 TTBL 0=10,1,08:00,1-5*	; define ON control for unit TC10
Response:	#TTBL 0=10,1,08:00,1-5	; to line 0 (out of 256)
Query:	1234 TTBL 0=*	; remove control line from time table
Response:	#TTBL 0 DELETED	
Query:	1234 TTBL 0?*	; read line 0
Response:	#TTBL 0=10,1,08:00,1-5*	
Query:	1234 TTBL A?*	; ask all TC definitions from PLC
Response:	#TTBL A=empty	; if no TC's or
	#TTBL 0=10,1,08:00,1-5 TTBL 1=10,0,16:00,1-5 TTBL 2=10,0,,24.12*	

Notice! Time control tables can be at any order within time table, so if you need to add new exception line for time control n, you can insert this line at the first available location AFTER normal ON/OFF control.

Timetable row TTBL	Time control block	Action 0-OFF 1-ON	Time	Date/Weekday
0	5	1	06:00	1-5 (Mo-Fr)
1	5	0	18:00	1-5 (Mo-Fr)
20	5	0		1.6
100	5	0		24.12
150	5	0		31.12

E.g. Using TC for changing the summer / winter time

TTBL 0 = 0,1,03:00,14.04 ;use table 0; set TC0 active at spring
TTBL 0 = 0,0,04:00,25.09 ;reset TC0 at autumn

In application program set device to send time correction message to itself

Line 0: TC0=1 “0000 TIME=0400,%RO250,%RO249%RO248*”*
;when TC0 goes active send new time to PLC

Line 1: TC0=0 “0000 TIME=0300,%RO250,%RO249%RO248*”*
;when TC0 goes inactive send new time to PLC

Q: When we change time back one hour, doesn’t the unit send the same message after time gets four o’clock?
A: After one hour the message is not resent because the TC0 is already 0 so negative derivations won’t take place. The only place when TC0 is set active is one 15 minute window at spring.

9.3 I/O-CONTROL COMMAND

9.3.1 SET

- Define output status

Format: [password] SET [var] [n]=0/1*

SET	function	
var	DOn=x	digital output, where n defines output number
	CNn=xxx	counter n
	Mn=x	memory n
	WGn=xxxxx	PID controller variable n
	WMn=xxxxx	word memory n
	AOn=xxxxx	analogue output n
	ROn=xxx	configuration register n

Examples:

Query: 4321 SET DO1=0*
Response: #DO1=0

Query: 4321 SET WM100=1300* ; set counters alarm level to 1300
Response: #WM100=1300 ; program could be: 'CN0>WM100' DO0=1

9.3.2 READ

- Read I/O-status

Format: [password] READ [var][n]*

READ	function	
Var	DOn	Digital Output, where n defines output number
	DIn	Digital Input
	AIn	Analoque Input
	AOn	Analogue Output
	CNn	Counter n
	Mn	Memory n
	WGn	16-bit PID controller variable n
	WMn	16-bit variable n
	AOn	16-bit variable n
	ROn	Special function register
	AFn	Alarm Flag n
	CSn	Call Status n

Examples:

Query: 4321 READ DO2*
Response: #DO2=0

Query: 4321 READ RO100*
Response: #RO100=14

Query: 4321 READ AI0*
Response: #AI0=3000

9.4 PROGRAMMING COMMANDS

9.4.1 INIT

- Define control lines
- **NOTICE! This command stops the application program execution! PLC must be started with RUN-command**
- **INIT command is not used in source code editor**

Format: [password] INIT [line] '[Condition]' [Action] ["text"] [num] [alarm reset time]*

Examples:

; Send message "ALARM" to ph. number 1 when DI0 goes active
Query: 4321 INIT 3 'DI0=1' "ALARM" 1*
Response: #Line3:'DI0=1' "ALARM" 1

; Set output0 active when input 0 =1 and input 1 = 0
Query: 4321 INIT 7 'DI0&!DI1' DO0*
Response: #Line7: 'DI0&!DI1' DO0

; Turn output 0 off after it has been active for 5 seconds
Query: 4321 INIT 9 'DO1S5' DO1=0*
Response: #Line9: 'DO1S5' DO1=0

; If DI2 XOR DI3 = 1, set output 0; active for 3 seconds
Query: 4321 INIT 10 'DI2XDI3' DO0S3*
Response: #Line10: 'DI2XDI3' DO0S3

; When input 0 changes to 1, send counter value 2 to ph. number 1
Query: 4321 INIT 11 'DI0=1' "Counter:%CN2" 1*
Response: #Line11: 'DI0=1' "Counter %CN2"

; Display counter value to display once/minute
Query: 4321 INIT 'P2' "Counter: %CN0" 255*
Response: #Line12: 'P2' "Counter: %CN0"

; If incoming SMS message ="$STATUS*", set flag
Query: 4321 INIT '($STATUS)' M0*
Response: #Line12: '($STATUS)' M0

; If status request flag active, send input values to the last received ph. number
Query: 4321 INIT 'M0' "STATUS= %DI0,%DI1,%DI2,%DI3" 254*
Response: #Line12: 'M0' "STATUS=%DI0,%DI1,%DI2,%DI3" 254

9.4.2 VIEW

- Read control line(s) from device

Format: password VIEW line*

Each 'INIT, Condition, Control' forms a line

Examples:

Query: 4321 VIEW 07*
Response: #Line7: 'DI0&!DI1' DO0

Query: 4321 VIEW ALL*
Response: #Line1:'DO0S5' "ALARM" 1 0 #Line2:'DI2=0' DO2=1 #Line3….etc.
#Line8:'CN1=7' DO1=1 #Line9: 'DO1S8' DO1=0

9.4.3 DEL

- Delete program line
- **NOTICE! This command stops the application program execution ! PLC must be started with RUN-command**

Format: *Password* DEL line*

Examples:

Query: 4321 DEL 07*
Response: #LINE 07 DELETED ; line deleted
#LINE 07 EMPTY ; if there is nothing to delete

Query: 4321 DEL ALL*
Response: #ALL DELETED

9.4.4 RUN

Start application program execution

Format: *Password* RUN*

Examples:

Query: 4321 RUN*
Response: #RUNNING ; application program is running
#FLASH FAILED ; flash failure

Query: 4321 RUN ?*
Response: #RUNNING
#STOPPED
#FLASH FAILED

9.5 ALARM CONTROL

9.5.1 ACK

Acknowledge alarm

Message needs to be acknowledged if the "Alarm reset time" is defined after phone number in program line. For each alarm message there is corresponding alarm flag. This flag is set every time a SMS-message that requires ACK, is sent. Message sending is allowed only if corresponding Alarm flag =0. There are three different way to reset Alarm flag; By sending ACK command, Alarm reset time expires or application program resets the alarm flag.

Format: [password] ACK [ID_number]*

ID_number = Program line number that created alarm message.
"ALL" clears all active alarms

Examples:

Message in: #02 Door opened
Query: 4321 ACK 02* ; acknowledge alarm ID 02

Messages in: #00 Tank 1 high level
#01 Tank 2 high level
Query: 4322 ACK ALL ; acknowledges all alarms

Note!
Alarm message needs not to be acknowledged only if message starts with "#xx" characters (xx = line number).

9.5.2 PRT

Print Command to HMI

Format: *Password* PRT "TEXT TO HMI" n*

n	is time in minutes that all other print commands are disabled.
RO240	shows the remaining disable time
0	= enable print immediately
1..254	= disable time in minutes
255	= disable prints

Example:

Query: 4321 PRT "Code is 6934" 2* ; Lock message to display for 1-2 minute.

9.6 Real time clock

9.6.1 Setting the clock

Send message
Password TIME =dd.mm.yy wd hh:mm

Where dd =1 to 31
mm = 1 to 12
yy = 00..99
wd = 1..7
hh = 00..23
mm = 00..59

Example:
0000 TIME =30.07.01 1 10:56*

Notice! There must be space between command TIME and Equal "=" character. Also date, weekday and time must be separated with space.

Message PSW TIME ? Returns date and time from real time clock.

Using the clock: See commands Date and Time.

9.7 iButton

9.7.1 Reading the iButton

GSM-PLC checks once/second if iButton device is connected to system.
If it founds iButton device, it sets the RO99=1 and reads the serial code from iButton to RO100…RO107.

Notice!

In this version only the “Read Serial code” function is supported.

9.8 FTP file system (Datalogging)

There are altogether 256kbytes of memory that can be used for datalogging.
This memory space can be divided in 4 different ways
1 file size of 256k
2 files sizes of 128k
4 files sizes of 64k
8 files sizes of 32k

Files are transferred from PLC using FTP file transfer (See sec. 2.5 FTP)

Values are saved to file similar way than sending an SMS

For each file there is a phone number location
Phone number 245: File number 0
Phone number 246: File number 1
Phone number 247: File number 2
Phone number 248: File number 3
Phone number 249: File number 4
Phone number 250: File number 5
Phone number 251: File number 6
Phone number 252: File number 7

Example:
saving a record into file 0
‘P1&M8’ “%AI0%AI1” 245 ; save record (AI0 & AI1) to file 0 once/sec if M8=1

9.8.1 FTP file structure

File contains header block & n*records.

Each file contains the header information that describes the record structure

File header:

PLC ID - 4 bytes
Variable count (length of header block= variable count*2)
Var 1 type – 1 byte
Var 1 addr – 1 byte
Var 2 type – 1 byte
Var 2 addr – 1 byte
…
Var n type – 1 byte
Var n addr – 1 byte

Record format:

Time stamp – 4 bytes
Var 1 value – 2 bytes
Var 1 value – 2 bytes
…
Var n value – 2 bytes

Time stamp format:

Byte 0: YYYY YYYM YYYYYYY - Year 00..99
Byte 1: MMMD DDDD MMMM – Month 1..12
Byte 2: 000H HHHH DDDDD – Day 1..31
Byte 3: 00mm mmmm HHHHH – Hour 00..23

Mmmmmm – Minutes 00..59

Variable types:

DI = 0
DO = 1
M = 2
AI = 3
CN = 4
RO = 5
WM = 6
AO = 7
PhNum = 10

Example 1:

File: FADE12340002030003010027090506010 5FF00270906060305EF

Where
Header info:
FADE1234 = PLC ID
0002 = variable count
0300 = Var1 type & address (AI0)
0301 = Var2 type & address (AI1)
Record 1:
00270905 = timestamp Jan. 7 9:05
0601 = Var1 value
05FF = Var2 Value
Record 2:
00270906 = timestamp Jan. 7 9:06
0603 = Var1 value
05EF = Var2 value

Example 2:

PLC: CCBBCCBB, Number of records: 3

10-06-06 19:54,N253=+358407695459,AI0=0
10-06-06 19:54,N253=+79112354030,AI0=10
10-06-06 19:54,N253=+78129322281,AI0=3

Header info:
CC BB CC BB = PLCID
00 02 = Variable count
0A FD = Var 1 type & address (N253)
03 00 = Var 2 type & address (AI0)

Record 1:
0C CA 13 36 - Time stamp 10-06-06 19:54
0C 91 53 48 70 96 45 95 F9 48 - Phone number
(according GSM standard: 0c - length, 91 - international prefix, and number)
00 00 - Value ai0
Record 2:
0C CA 13 36
0B 91 97 11 32 45 30 F0 F9 48
00 0A
Record 3:
0C CA 13 36
0B 91 87 21 39 22 82 F1 F9 48
00 03

9.8.2 FTP file

Get file info

Format: Password FILE n?*

Response: #FILE n = <FileSize> <used> <record variables>

Examples:

Query: 4321 FILE 1?
Response: #FILE 1= 32768 180 AI0 AI1

Query: 4321 FILE 2?
Response: #FILE 1= 32768 16358 CN8 WM7 WM8 WM9

Query: 4321 FILE 0?
Response: #FILE 0= UNUSED

Clearing File:

Query: 4321 File 1=
Response:

Examples:

Query: 4321 FILE 1=
Response: #FILE 1= 32768 0 AI0 AI1
Or
Query: 4321 FILE 0=
Response: #FILE 0= UNUSED

9.9 Modbus Slave/Master (GSM Ver.1.38)

Added RO's (compatible with standard AL20 except RO215):

RO056	TIMEOUT MODBUS SER2
RO058	SER2 CTS/RTS CONTROL (ACC.0=1 NO RTS/CTS HANDSHAKE)
RO059	SER2 MODBUS ERRORS
RO060	EXCEPTION STATUS
RO080	SER2 MODBUS REJECTED MESSAGES
RO081	SER2 MODBUS RCEIVED MESSAGES
RO082	SER2 QUEUE DEPTH
RO084	Conditional message address
RO088	SER2 MODBUS NON ANSWERED ID
RO089	SER2 MODBUS MESSAGE ERROR TYPE
RO215	SER2 MODE (0=Modbus salve, 1=Modbus master)
RO219	SER2 DATA FORMAT (0=default=8 bit, parity none) (1= 7bit, parity even) (2= 7bit, parity odd) (3= 8bit, parity even) (4= 8bit, parity odd)
RO229	SER2 BAUD (0=300, 1=1200, 2=2400, 3= 4800, 4=9600, 5=19200)
RO243	MODBUS OWN SLAVE ID (1-254)

9.9.1 Modbus memory map

cmd 01/05,15	length, offset
Binary output (DO)	0100h 0000h
Binary memory (M)	0100h 0400h
Binary memory (AF)	0100h 0800h

cmd 02

Binary input (DI)	0100h 0000h	DI
Time control (TC)	0100h 0400h	TC

cmd 04

Register input (AI 8bit)	0100h 0000h	
Word input (AI 12bit)	0100h 0400h	AI

cmd 03/06,16

Register output (RO)	0100h 0000h
Register memory (CS)	0100h 0400h
Word output 0 ->255 (AO)	0100h 0800h
Word memory 0 ->255 (WM)	0100h 0C00h
Word general memory 0 -> 255 (WG)	0100h 1400h
Counter memory 0-> 1023 (CN)	0100h 6000h

9.9.2 RO215 - SER2 MODE

Modbus Slave mode
Modbus Master mode
UCP
TAP
Transparent mode

DIP SW3 Overrides SER2 modes

When ON:

SER2 Mode	Modbus Slave
Baud	9600
Own ID	01

9.9.3 Modbus memory map for commands 09, 10

Block numbers	Descriptions
0000-01ff	(*) Program lines, phone numbers, I-Buttons
0400-0401	(*) PIN, PSW
0800-0815	Modbus Master configuration
0C00-0C01	(*) FTP Info
ffff	Status Block

(*) – for future development

9.9.4 Programming Modbus Master configuration via ALPROWIN

Messages must be defined only for SER1 in ALPROWIN programming tool.
Maximum 255 messages
(Note! Although Modbus master configurations are defined in Alprowin for SER1 (COM1), those will be used for SER2 (COM2) in GSM-PLC.

3.10.5. Launching Modbus Master conditional messages

RO84 is used to define the number of conditional message you need to send to slave unit.

RO84=255 triggers all conditional messages.
After command execution RO84=0.

9.9.5 Example of configuring SER2 as Modbus master

You need AlproWin programming software, so you can configure Modbus master to AL GSM PLCs. If you don't have this software ask it from FF-Automation.
Note that you can use only COM 2 (serial port 2) for Modbus master configuration. COM1 is for GSM modem. (You need to make configurations in COM1 in Alprowin, although the configuration is used in SER2 in AL GSM PLC!)
Write the following parameters (code) using GSM Programmer (not Alprowin):
R O 215 = 5 (SER2 mode, 1 = modbus master)
R O 229 = 4 (SER2 baud rate, 4= 9600bps, 5=19200bps)

Modbus master configuration can be transferred to AL GSM PLC only if the serial port 2 is in modbus slave mode, speed 9600, address 1. (DIP3=ON)

You can force the serial port 2 to modbus slave mode if you use on-board DIP switches:
DIP 3 = ON (forces serial port 2 to modbus slave mode, speed 9600, address 1)

You need also to set the alprowin serial port to same setting (port 2 to modbus mode, speed 9600, address 1)

Before you can transfer the modbus configuration you need to set STOP command to AL GSM PLC (Red STOP icon in the alprowin, stops the run led on-board)

After the modbus configuration is transferred you need to press green start button.
Note also that if you configure AL GSM PLC as modbus slave you don't need AlproWin.
Write the following parameters (code) using GSM Programmer (not alprowin) if you are using modbus slave in SER2:

RO 243 = slave adddress e.g. 1
RO 215 = 0 (modbus slave)
RO 229 = 4 (4 =9600, 5=19200)
Note that if DIP 3 = ON, it will override these parameters.

Look AlproWin User Handbook for more information!
Note also that if you need RS485 you need hardware module for that.

9.10 Transparent mode

This mode allows to connect to 3rd party device through GSM PLC.
GSM PLC can be set to transparent mode when it detects incoming call
In transparent modem PLC echoes the data coming in from

10 GSM-PLC's Application Program –Principle of operation

10.1 Program cycles

Program cycle time is max. 100 ms, depending on the application program length. Normally its about 5-20 ms.

Program cycle includes the following steps:

1. Write outputs (Digital outputs, Analog outputs)
2. Read Inputs (Digital inputs, Analog inputs)
3. Execute Application program (line by line, from the beginning to the end, from left to right if there are many commands in one line)

   Counter inputs CN[n] are updated once per 5ms, giving max frequency of 100Hz

GSM driver runs as background task.

As background task also runs
- PID controllers
- Display driver
- Serial protocols

SMS sending cycle:

- Each programming line (0-255) has two field for SMS sending 1) flag (0/1), which defines if SMS needs to be send, 2) The SMS message which needs to be send
- Note that programming lines 256-511 cannot be used for SMS sending!
- Program checks all fields for each programming line (0-255) and if the flag is 1, it will send the SMS message.
- Normally user don't need to know the above, because SMS sending cycle is working automatically. User don't need to set flags.

Storing to FILE:
- Storing to File happens same way as triggering SMS message, only that phone number defines the destination File. PhNum 245 = File 0, PhNum 246 = File 1 etc.

10.2 Retaining / clearing variables

Some variables are retained when GSM-PLC is booted (power reset)

Retained variables are retained only if DIP5=0 and Backup battery has sufficient voltage. Backup battery need to be replaced with new one about once / 5-10 years in normal operation when GSM-PLC is powered. If the GSM-PLC is not powered back-up battery can go empty much faster (within some months).

Program code and phone book are stored to flash memory, so these are retained also if backup battery goes empty.

Also user can inialize variables in application program with default values, so if backup battery goes empty, these variables will be set automatically back to normal values.

10.3 Principle of operation

GSM-PLC can be used as simple PLC and it can also send predefined SMS-messages with parameters to desired GSM-numbers.

GSM-PLC can also make normal call to phone.

GSM-PLC can also receive SMS messages including parameters.

GSM-PLC can perform logical, Date/time based and SMS-message based operations. Operations can either be controlling outputs or sending SMS-messages (Alarms, Data logging messages, Info etc.).

If there is a display unit attached to PLC, display keys is seen in digital inputs 240..255.
Display LED's can be controlled through digital outputs 240..255.

GSM-PLC stops application program execution after it has received INIT or DEL command. GSM-PLC starts executing the application program after it receives RUN command.

11 Transferring program to GSM-PLC

11.1 Memory List for Programming GSM-PLC via cable:

Look also the Transferring the project to GSM-PLC- in Getting started chapter how to do this!

Memory list for Programming GSM-PLC

1. Connect Laptop's or PC's Serial port to GSM-PLC's SER1 using AutoLog Programming cable.
2. Change DIP switch 4 to ON -position (enables programming / disables GSM modem)
3. Reboot GSM-PLC (switch off/on the power).
4. Start GSMProgrammer, set Transfer->Config to [e.g. COM2], 9600, Direct connection
5. Wait until the the status becomes ACTIVATED and the LEDs start blinking (wait about 5 seconds-1 minute)
6. Create PLC program, Compile and transfer it to GSM-PLC.
7. Start GSM-PLC program (if you reboot PLC without starting it first, it will start old PLC program after reboot!)
8. In programming mode (DIP4=1) you can test the application with GSMProgrammer's Alarm Log window.

Memory List for Running GSM-PLC Program over GSM modem (experts only!)

1. Start GSM-PLC program (if you reboot PLC without starting it first, it will start old PLC program after reboot!)
2. Change DIP switch 4 to OFF -position (enables GSM modem / disables programming).
3. Reboot GSM-PLC (switch off/on the power).
4. Wait until GSM modem is initialized (Look GSM and ser port LEDs.).
5. Test the application.

Normally you make application program with GsmProgrammer and you compile and transfer the application program to GSM PLC via cable (AutoLog Programming cable).

11.2 Transferring program to GSM-PLC via GSM network

After you have commissioned the project, it almost always happens that customer wants some changes to the application program. It is not necessary to go to the site to make update. You can download the application program change via GSM network using GSM phone or GsmProgrammer with External GSM modem attached in PC's serial port.

Cautions! When you are updating application program via GSM network you need to be very careful, not to make any mistakes so double check everything before sending commands!

Do not add new programming line(s) in the middle! You can only replace existing programming lines or lines which are reserved = .empty.

11.2.1 Updating application program using GSM phone

Note! If you have only few lines which are ***replaced***, then its easy to upload the new program using GSM phone.

Note! If you are ***adding new*** line(s), then you need to program all the rest lines also, because line number changes for the rest of the program. New lines can be added in the end of the program or you can replace ".empty" lines with new programming line.

Here's an example how to replace programming line.

1. Make changes to the program and compile the program with GsmProgrammer.

2. Print the source code e.g. using PDF printer (CutePDF is free).
3. Printed source code has program line numbers (look example bellow)

Old programming lines:

```
40 'WM53<1060' M30
45 'WM54<1060' M31
```

New programming line:

```
40 'WM53<1040' M30
45 'WM54<1040' M31
```

4. Send following SMS to the GSM-PLC

0000 VIEW 40*0000 VIEW 45*

5. You will receive SMS from the GSM-PLC which confirms that these lines are correct.

#LINE 40: ‘WM53<1060’ M30#LINE 45: ‘WM54<1060’ M31

6. Edit the answer (or make new SMS message) to the following format:

0000 INIT 40 ‘WM53<1040’ M30*0000 INIT 45 ‘WM54<1040’ M31*0000 RUN*

Do not forget the RUN command in the end. Otherwise the program will not change and program is stopped.

7. You will receive the following SMS from the GSM-PLC:

#LINE 40: ‘WM53<1040’ M30#LINE 45: ‘WM54<1040’ M31#RUNNING

That’s it! If you need to upload the program to other similar GSM-PLCs, the repeat steps 4-7.

Check again that you have changed also the programming code in GsmProgrammer!

Remember also to update the line which includes version number of the application program if you are using it!

11.2.2 Programming GSM-PLC via GSM network using GSMProgrammer

- **Do not insert password and *-character (end of line) into program line.** PC-program adds those automatically when you transfer your application program from Program Editor or from Alarm Log control window to GSM-PLC.

- You can change GSM-PLC password from GSM-Programmer. In main window there is a password code box (default=0000).

11.2.3 Program Editor

- **Do not insert INIT-command.**

Structure: ’Condition’ Action
Or
’Condition’ ”SMS-message” phone_number_id
Or
’Condition’ Action ”SMS-message” phone_number_id

11.2.4 AlarmViewer

- When is changed the program from Alarm Log, you must insert INIT-command and line number before program line, but you don't need to put "*" in the end of line.

Structure: INIT line 'Condition' Action
Or
INIT line 'Condition' "text" phone_number_id
Or
INIT line 'Condition' Action "text" phone_number_id

11.3 Initialization other changes for GSM-PLC using GSM phone

Normally these are made with GsmProgrammer, but if you need to change these afterwards remotely through GSM network you can do it using the following commands:

Define phone numbers using GSM- phone:

1234 NUM0=+35850123456*
[Password] [Number 0] [Phone number]

GSM-PLC response: #NUM0=+35850123456*

Define password using GSM- phone

1234 PSW 4321 4321*
[Old Password] [Command] [New Password]

GSM-PLC response: #PSW OK
(normally password is not needed!)

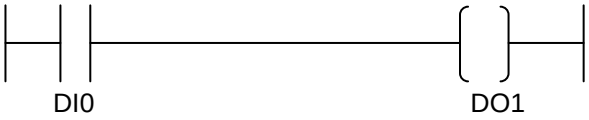
Define PIN-code using GSM- phone

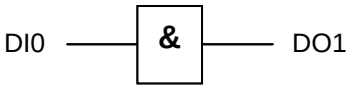
1234 PIN 4321 4321*
[Old Password] [Command] [New Password]

GSM-PLC response: #PIN OK
(normally PIN code is disabled!)

12 Basic programming examples

12.1 Connect input 0 to output 1.

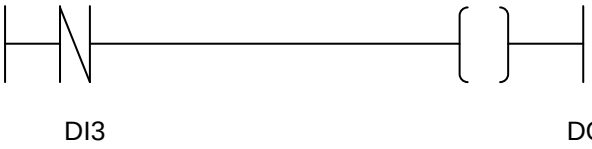


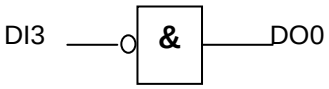


Program:

‘DI0’ DO1 ; If input 0 = true, control output 1 on.

12.2 Connect inverse input 3 to output 0

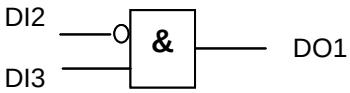




Program:

‘!DI3’ DO0 ; If input 3 = false, control output 0 on.

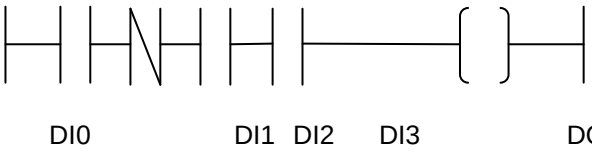
12.3 Connect inverse input 2 and input 3 to output 1.

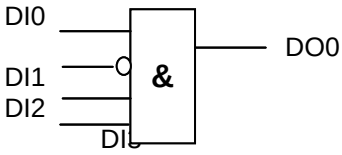


Program:

‘!DI2&DI3’ DO1 ; If input 2 = false and input 3=true, ; control output 1 on.

12.4 Several inputs to AND-block

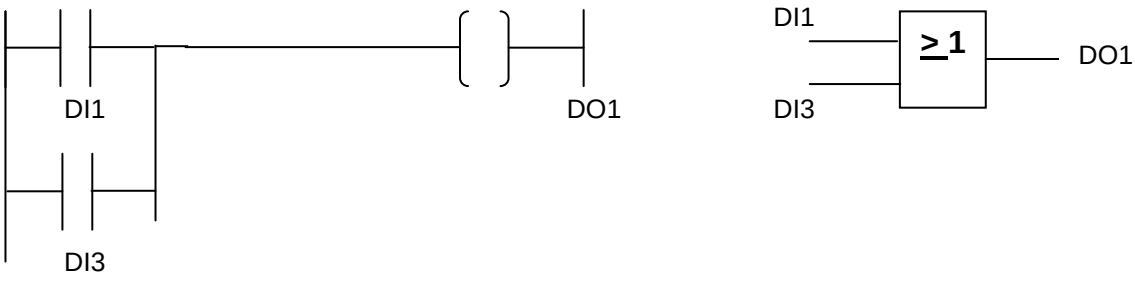




Program:

‘DI0&!DI1&DI2&DI3’ DO0 ; If input 0 = true and input 1=false and ; input 2=true and input 3=true, control ; output 0 on.

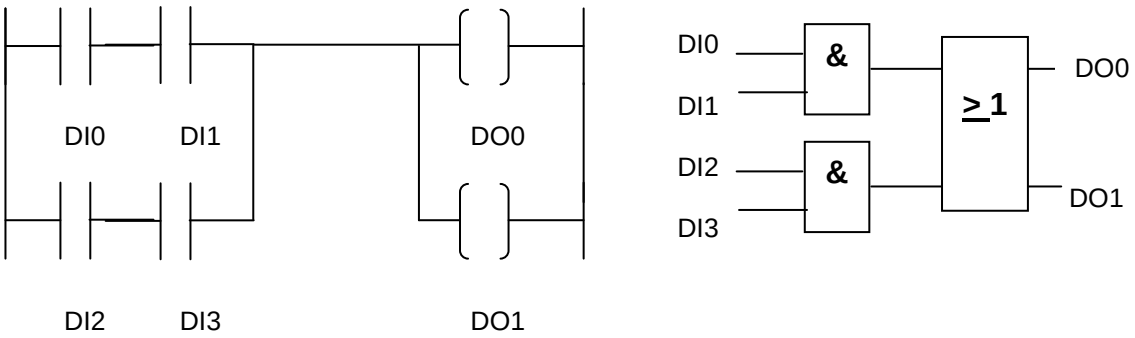
12.5 OR-block



Program:

'DI1#DI3' DO1 ; If input 1 = true or input 3=true control
; output 1 on.

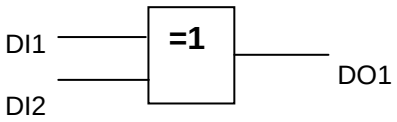
12.6 Combination of OR and AND blocks.



Program:

'(DI0&DI1)#(DI2&DI3)' DO0 DO1 ; If inputs 0 and 1 or 2 and 3 are
; true set output 0 and output 1 on.

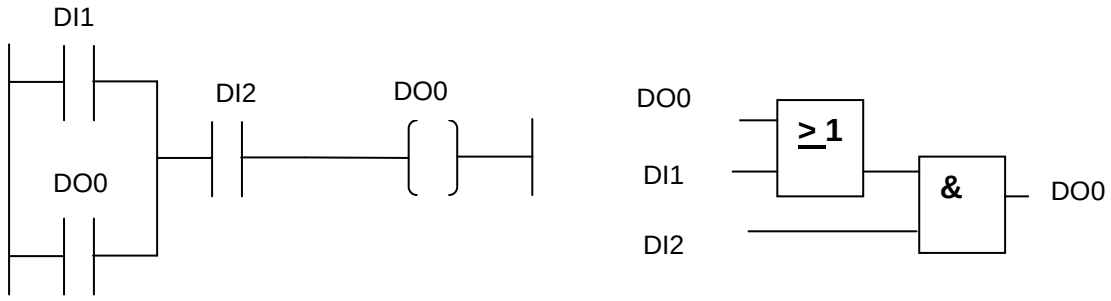
12.7 XOR-blocks



Program:

'DI1XDI2' DO1 ;If only input 1=true or only input 2=true,
; set output 1 on.

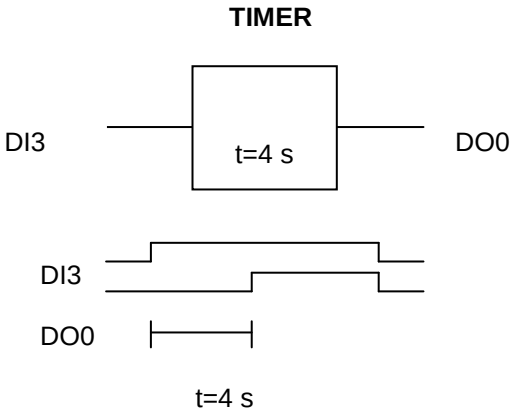
12.8 Set/Reset output



Program:

‘(DI1#DO0)&DI2’ DO0 ; If input 1=true or output 0=true and
; input 2=true, set output 0 on.

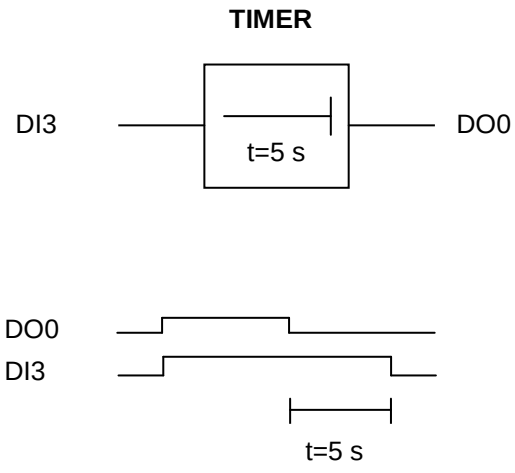
12.9 Going on time delay



Program:

’DI3S4’ DO0 ; If input 3 has been true 4 seconds, control
; output 0 on. (Timer is not exact 4 seconds)

12.10 Going off time delay

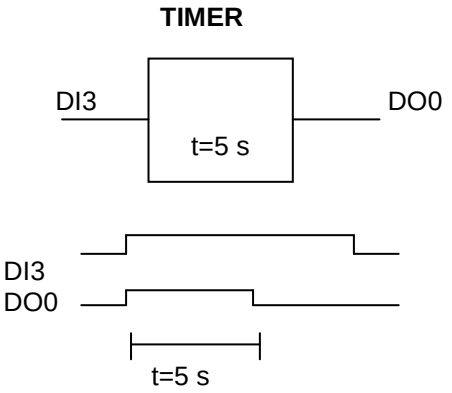


t=5 s

Program:

‘DI3’ DO0S5 ; If input 3 =true, output 0 is 5 seconds on after input 3 =false.

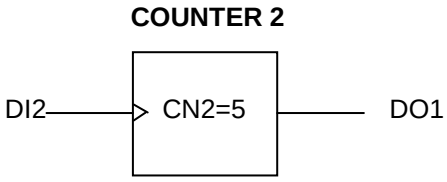
12.11 Pulse delay



Program:

'**DI3=1' DO0S5**	; If input 3=true, output 0 is 5 seconds on. ; (positive derivation)

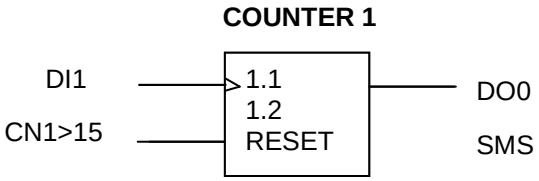
12.12 Counter



Program:

'CN2=5' DO1	; If the value of counter 2 is 5, control ; output 1 on. Counter 2 calculates from ; input 2.

12.13 Counter and message



Program:

'CN1>15' CN1=0	; If the value of counter 1 is more than ; 15, reset counter 1.
'**CN1>10' "In counter: %CN1" 1**	; If the value of counter is more than ; 10, send message (which contain the ; value of counter) to phone number1.

12.14 Clock and calendar printing to keypad/phone

- 255 is the ”phone number” of keypad (DI255)
- RO248: day of month
- RO247: month
- RO250: hours
- RO251: minutes
- RO252: seconds

Program:

```
'DI1' "%RO248.%RO247" 255          ; If input 1=true, print date.
'DI2' "%RO250:%RO251.%RO252" 255   ; If input 2=true, print time.
```

12.15 Controls from clock and calendar

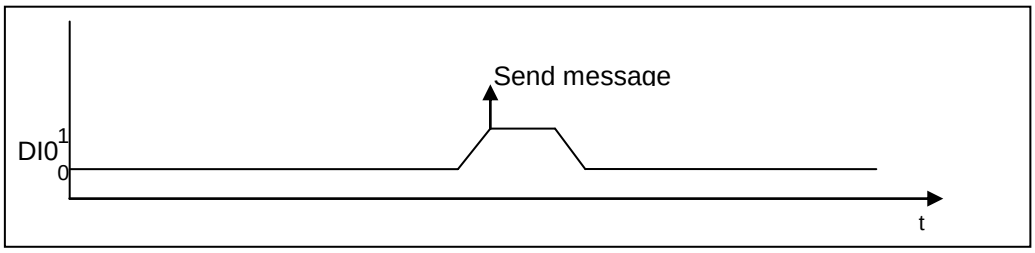
- RO249: day of the week
- RO250: hours
- RO251: minutes

Program:

```
'RO249=3&CLK=2330' DO1=1           ; When day of the week is 3
                                   ; and clock/time is 23:30, set
                                   ; output1 on.
'RO249=4&CLK=1210' DO1=0           ; When day of the week is 4
                                   ; and clock/time is 12:10, set
                                   ; output 1 off (reset).
```

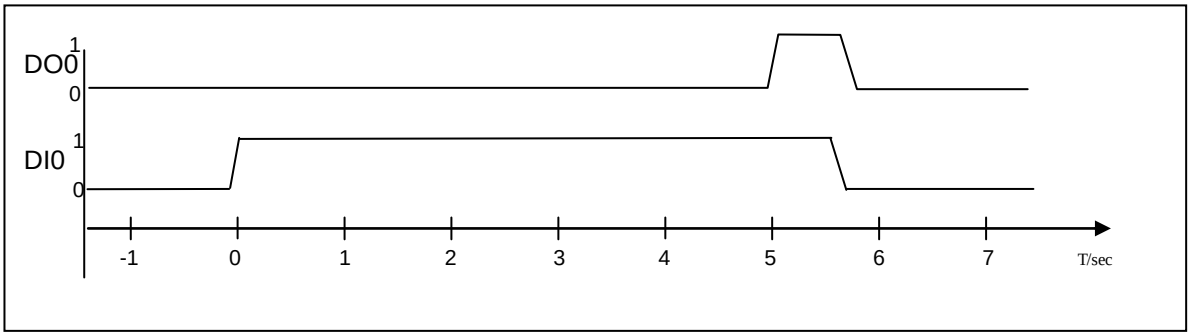
12.16 Send message when input 0 goes active.

```
'DI0=1'        "Tank level LOW"        0          0
 [Condition]    [Message]              [Phone #]  [optional: no ack.]
```



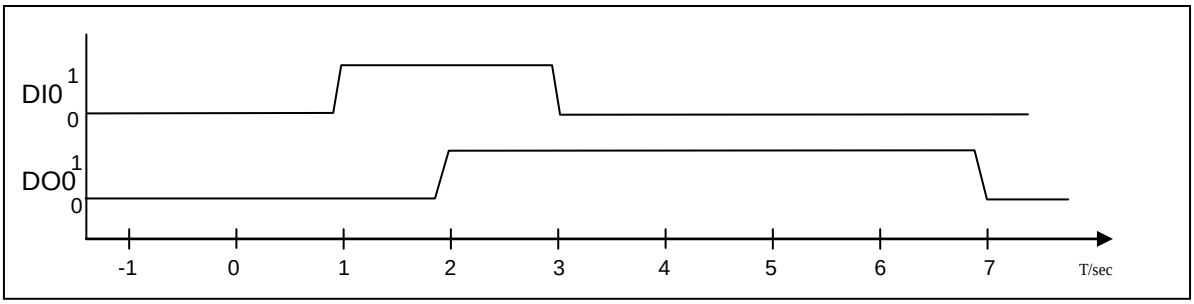
12.17 Set output 0 active if input has been on for 5 seconds

'DI0S5' **DO0**
[Condition] [Action]



12.18 Set output 0 active for 5 seconds if input 0 has been on for 1 second

'DI0S1' **DO0=1**
[Condition] [Action]
'DO0S5' **DO0=0**
[Condition] [Action]



12.19 Send message out if inputs 0 and 1 are active or input 3 is not active

'(DI0&DI1)#!DI3'	**"ALARM"**	**1**	**0**
[Condition]	[Message]	[Phone # id]	[no ack.]

12.20 Linking two PLC GSM4's together

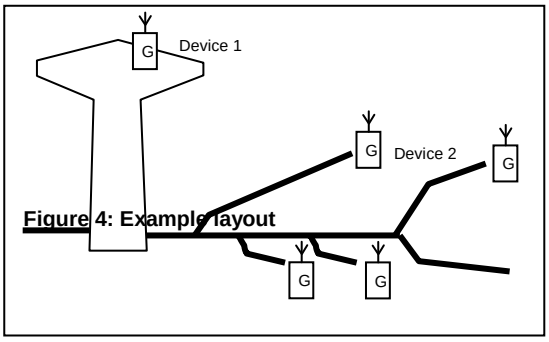


Figure 4: Example layout

1st GSM4 measures water level (Device 1)
2nd GSM4 controls pump (Device 2)

If water level drops below the set value, Device 1 sends a pumping request to device 2 at the pump station. After water level has increased enough, Device 1 send request to stop pumping.

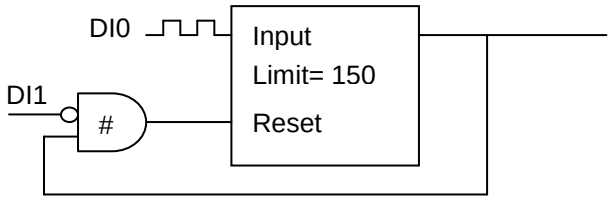
Program on device 1 "Water level control"

'(AI0<1000)'	**"1111 SET DO0=1*"**	**1**
[Condition]	[Message]	[Phone # id]
'(AI0>1500)'	**"1111 SET DO0=0*"**	**1**
[Condition]	[Message]	[Phone # id]

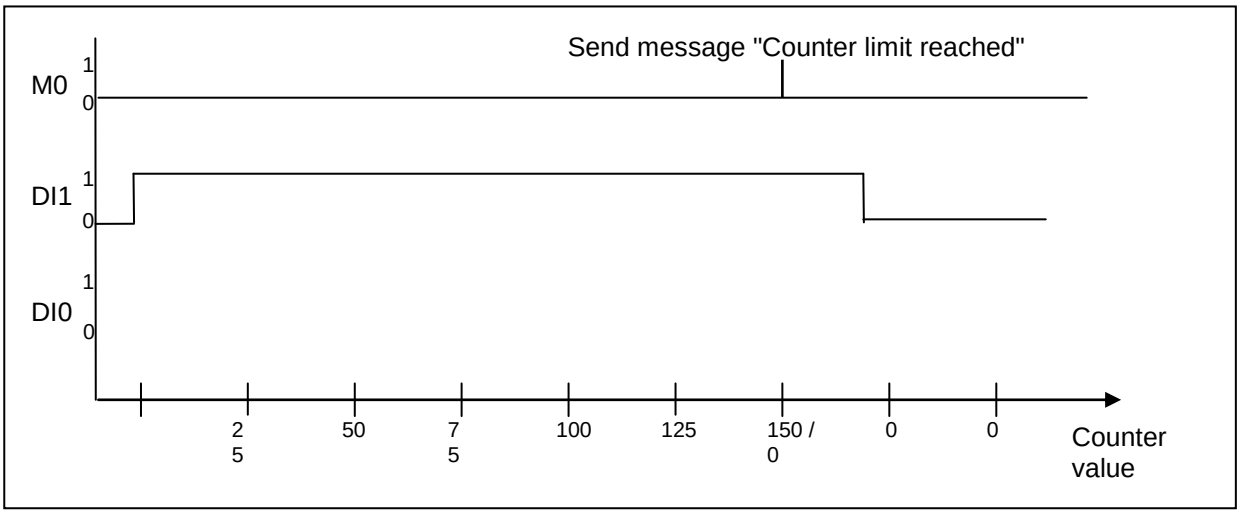
Password has to be set to 1111 on Device 2. There is no need for more programming on Device 2

12.21 Counter unit

Send message after counter is enabled (DI1=1) and it reaches 150 pulses. After this, reset counter.



'!DI1#M0'	**CN0=0**		
[Condition]	[Operation]		
'(CN0=150)'	**M0**		
[Condition]	[Operation]		
'M0'	**"Counter limit reached"**	**1**	**0**
[Condition]	[Alarm]	[Phone #]	[Ack]



12.22 Circulation alarm for three cellular phones

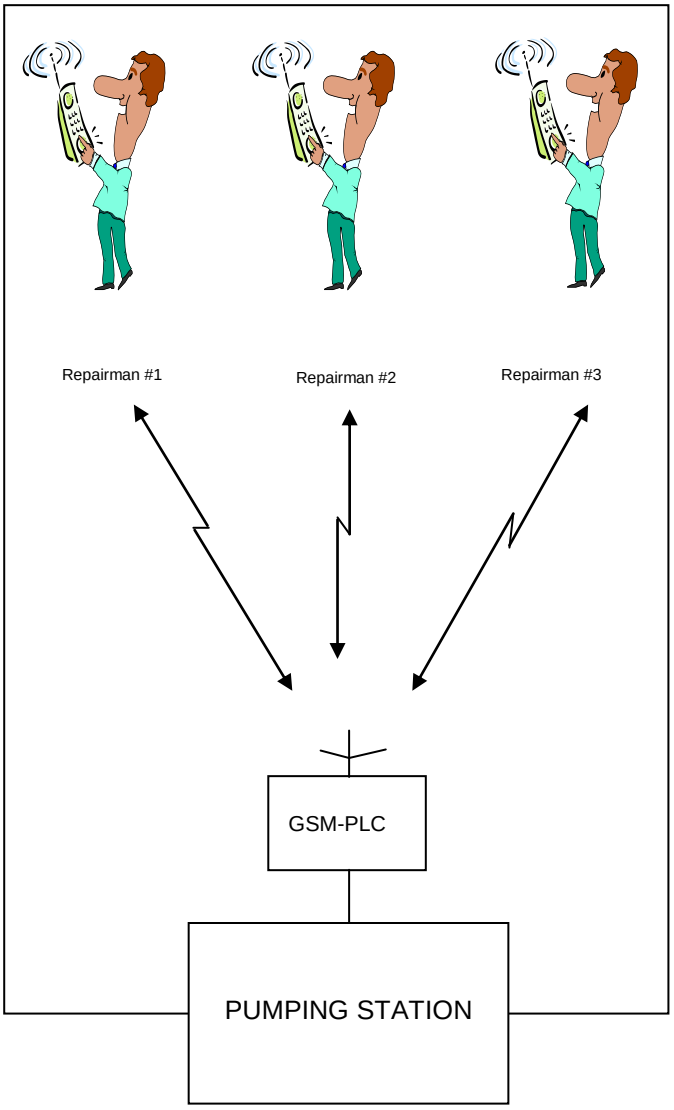
<u>The operating principle of circulation alarm program:</u>

When input 0 has been active for 30 seconds, GSM-PLC sends alarm message to repair man 1 (line 0 and 1). If nobody acknowledges this message with in 15 minutes, another message is sent to repair man 2 (line 2).

Again if there is no ACK from second repairman, 3rd message is sent to repair man 3 (line 3).

When any of AF-variables is acknowledged, new alarm sending is allowed but only after DI0 has been inactive (lines 4, 5 and 6).

If there has been no ACK from repair mans, after 60 minutes from the first alarm message, all alarm flags are cleared.



<u>Program:</u>

Line 0: 'DI0S30' M0=1
Line 1: 'M0=1' "ALARM 1" 0 60

Line 2: 'AF1M15&M0' "ALARM 2" 145
Line 3: 'AF2M15&M0' "ALARM 3" 2 30
Line 4: 'AF1=0#AF2=0#AF3=0' M1=0
Line 5: 'M0=1' M1=1
Line 6: '!DI0&!M1' M0=0

<u>Acknowledge alarm:</u>

Command: password ACK ALL*, clears all active alarms.

12.23 Time counter using WM-variables

Time counter is build using WM-variables and pulse variable (P1). For each parameter (Hour, min, sec) there is reserved one WM variable.

If the input 3 has been true for one second, add one to 'second' memory (=WM30) [line 0].

If the value of the 'second' memory is more than 59, add one to the 'minute' memory (=WM31) [line 1].

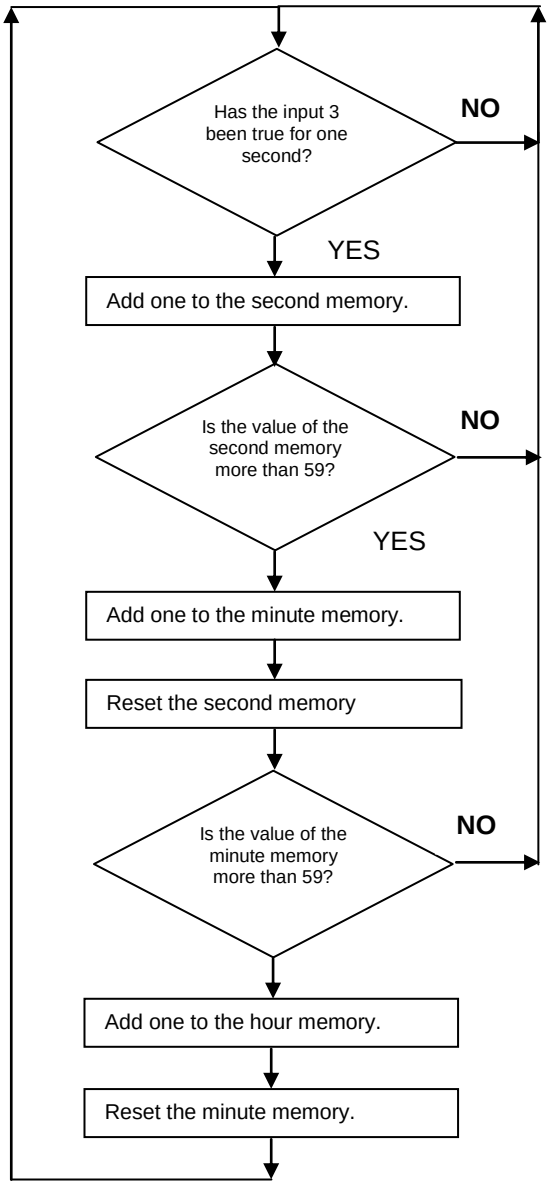
If the value of the WM31 is more than 59, add one to variable WM32 (=hour memory) [line 2].

If the value of the WM30 is more than 59, reset WM30 [line 3].

If the value of the WM31 is more than 59, reset WM31 [line 4].

When for example the machine has been used 1000 hours, GSM-PLC send "SERVICE SIGNAL" to repair man [line 5].

Line 0: 'DI3&P1' WM30+1
Line 1: 'WM30>59' WM31+1
Line 2: 'WM31>59' WM32+1
Line 3: 'WM30>59' WM30=0
Line 4: 'WM31>59' WM31=0
Line 5: 'WM32=1000' "SERVICE SIGNAL" 0



12.24 Password recognition using Word Pointer(GSM-PLC+control panel)

Password recognition uses WP (Word pointer). Password verification takes place, when all part of the password is inserted. Password can be changed afterwards, if needed.

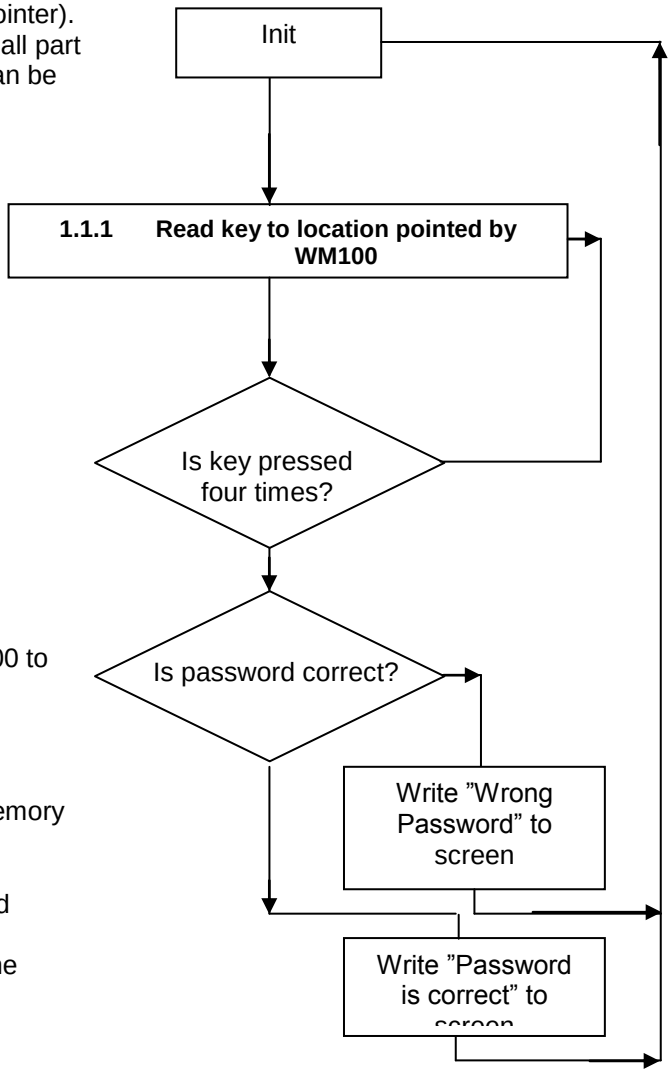


Table for password settings:

Character	Value of variable
0	12288 (3000H)
1	12544 (3100H)
2	12800 (3200H)
3	13056 (3300H)
4	13312 (3400H)
5	13568 (3500H)
6	13824 (3600H)
7	14080 (3700H)
8	14336 (3800H)
9	14592 (3900H)

Program:

Line 1:
Set password to 0123 and pointer WM100 to 150.
(RO207 – bit information from keys 0-7)

Line 2:
If key is pressed, write key number to memory location pointed
by WM100.
Increase key pressed counter by one and pointer WM100 by one.
(RO209 – last character received from the keyboard)

Line 3:
If first, second, 3rd and 4th number is like declared password, write “Wrong Password” to screen.

Line 4:
If password isn’t correct, send “Wrong password” to phone number 0 and initialise pointer.

Program:

Line 1: '!M0' WM50=12288 WM51=12544 WM52=12800 WM53=13056 WM100=150 M0=1 WM0=0 RO207=0

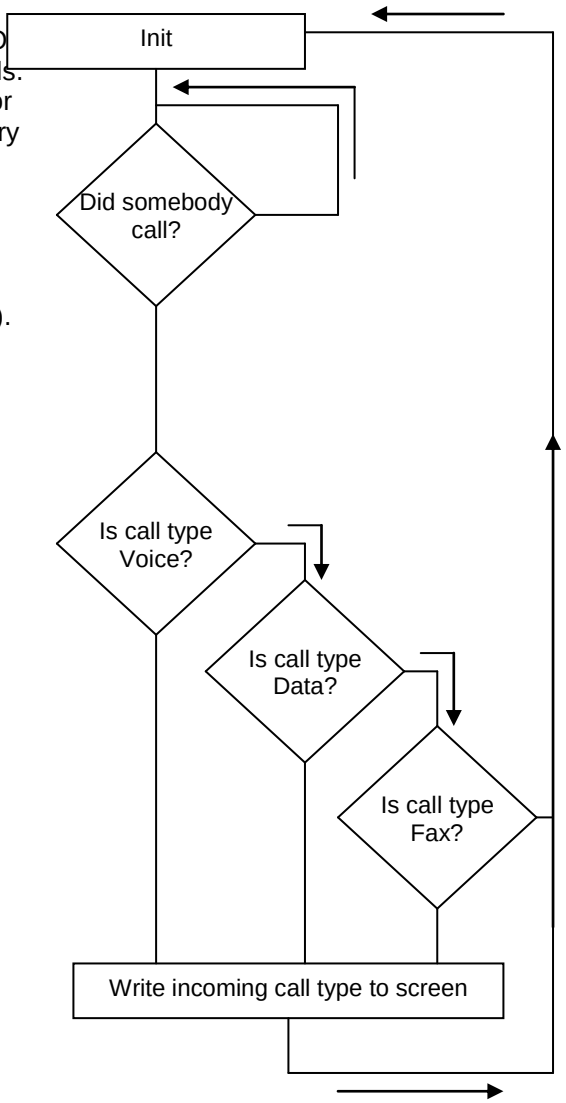
Line 2: 'RO209>0' WP100=RO209 RO209=0 WM0+1 WM100+1

Line 3: 'WM0=4&WM150=WM50&WM151=WM51&WM152=WM52&WM153=WM53' M0=0 WM0=0 "Password is correct" 255

Line 4: 'WM0=4' M0=0 "Wrong password!" 255

12.25 Voice/Fax/Data call recognising

To recognise incoming call type, this example uses RO
This register output recognises data, voice and fax calls.
If this register output value is 1, the call type is voice, or if registry entry value is 2, call type is data and if registry entry value is 4, call type is fax.



Program:

Line 1:
Initialise incoming call type register (register output 94).

Line 2:
If RO94 is 4, writes FAX to screen.

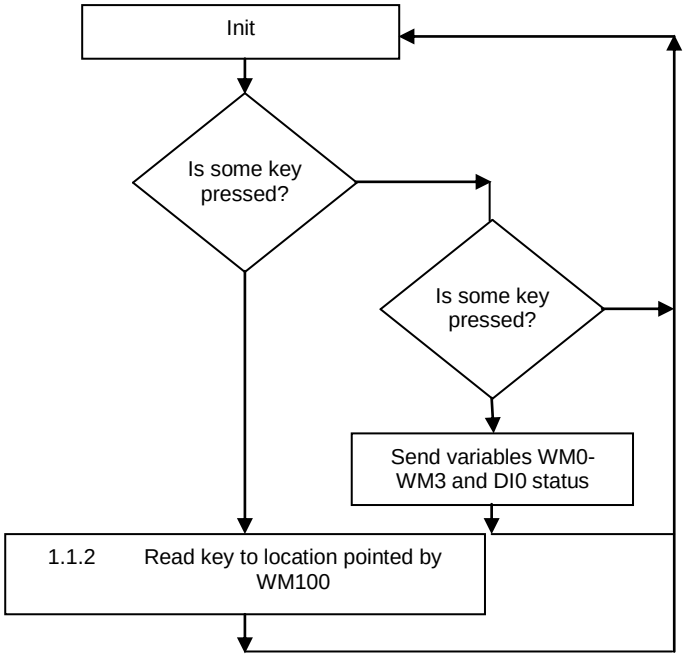
Line 3:
If RO94 is 1, writes VOICE to screen

Line 4:
If RO94 is 2, writes DATA to screen

```
Line 1:     '!M0' RO94=0 M0=1
Line 2:     'RO94=4' "FAX" 255
Line 3:     'RO94=1' "VOICE" 255
Line 4:     'RO94=2' "DATA" 255
```

12.26 Sending the values of the variables

This program uses normal send command, to send variable value to GSM-phone.To send variable by SMS, put %-character before variable in normal send command (example “%WM0”). This example uses WP (Word Pointer) to read keys to variables (Read “Password recognition using Word Pointer” to more information about WP).



Program:

Line 1:
Initialise variables

Line 2:
If key is pressed, Read key to location pointed by WM100

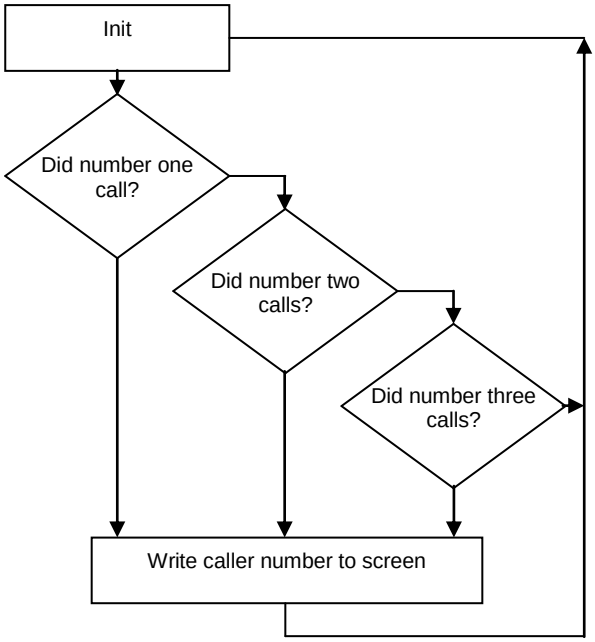
Line 3:
If GSM-PLC receive the SMS-message: $G, set memory 1.

Line 4:
If M1 is true send four first keys, that were pressed and the status of DI0.

Line 1: '!M0' M0=1 WM100=0 WM0=0 WM1=0 WM2=0 WM3=0
Line 2: 'RO209>0' WP100=RO209 RO209=0 WM100+1
Line 3: '($G)' M1=1
Line 4: 'M1' M1=0 M0=0 "YOU PRESS: %WM0@3 and the status of the digital input 0 is %DI0" 0

12.27 Detect the phone number of the incoming call

Program can detect only those phone numbers, which are defined in the phone book. In example, T0 is 1st number in phone book, T1 is 2nd and T2 is 3rd.



Program:

Line 1:
If number 1 calls, write it to screen.

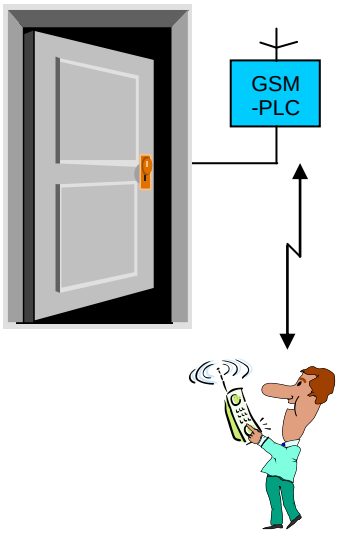
Line 2:
If number 2 calls, write it to screen.

Line 3:
If number 3 calls, write it to screen.

Line 1: 'T0' "Number 1 calling" 255
Line 2: 'T1' "Number 2 calling" 255
Line 3: 'T2' "Number 3 calling" 255

12.28 Opening the door by calling (the door with electric lock)

This example calls to the predefined phone number.



Program:

Line 1:
If the telephone number 0 call, set memory 1.

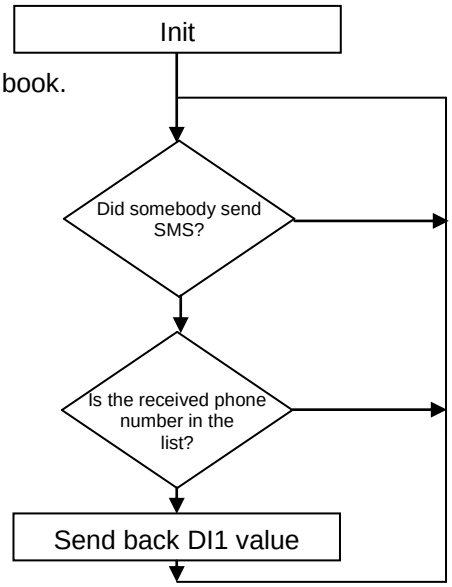
Line 2:
If the memory 1 is true, digital output 0 is true (=lock is open).

Line 3:
Reset memory 1 after 60 seconds (=lock is closed).

Line 1: 'T0' M1=1
Line 2: 'M1' DO0
Line 3: 'M1S60' M1=0

12.29 Recognising incoming SMS number

This program recognises four phone numbers from phone book.
This example sends back the value of the DI0-variable,
if the phone book includes incoming phone number.



Program:

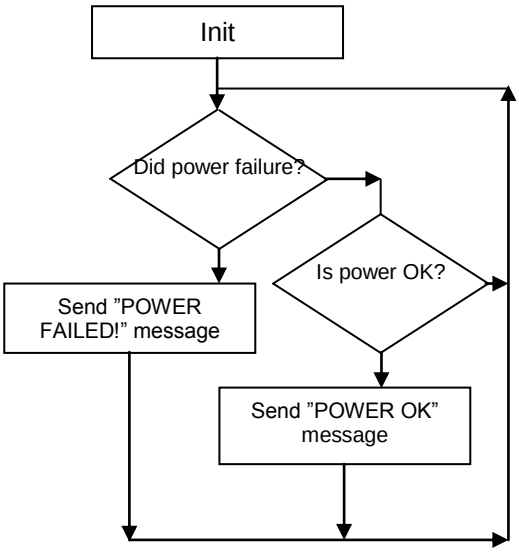
Line 1:
If PLC receives SMS "$INFO" from phone number 1-4…

Line 2:
… send back the value of the DI0-variable.

Line 1: '($INFO)&(N0#N1#N2#N3)' M1=1
Line 2: 'M1=1' M1=0 "DI0: %DI0" 254

12.30 Detect power failure

Example uses Register Output 36, which will change to 1, when power failure is detected. Device will shut down automatically; two minutes after power fail detection. When somebody put power to GSM-PLC, this example sends message: “POWER OK” (if GSM-PLC has not power troubles). If troubles occurred, program sends SMS-message:
“POWER FAILED!”.



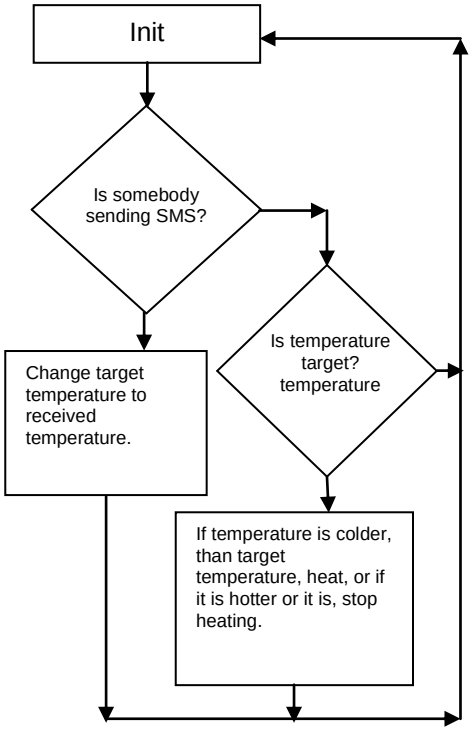
Program:

Line 1:
If power is OK or it is OK during the system start-up, send message to the phone number 0.

Line 2:
If power is failed or it fails during the system starts up, send SMS-message: "POWER FAIL!" to the phone number 0 .

Line 1: 'RO36=0' "POWER OK" 0
Line 2: 'RO36=1' "POWER FAIL!" 0

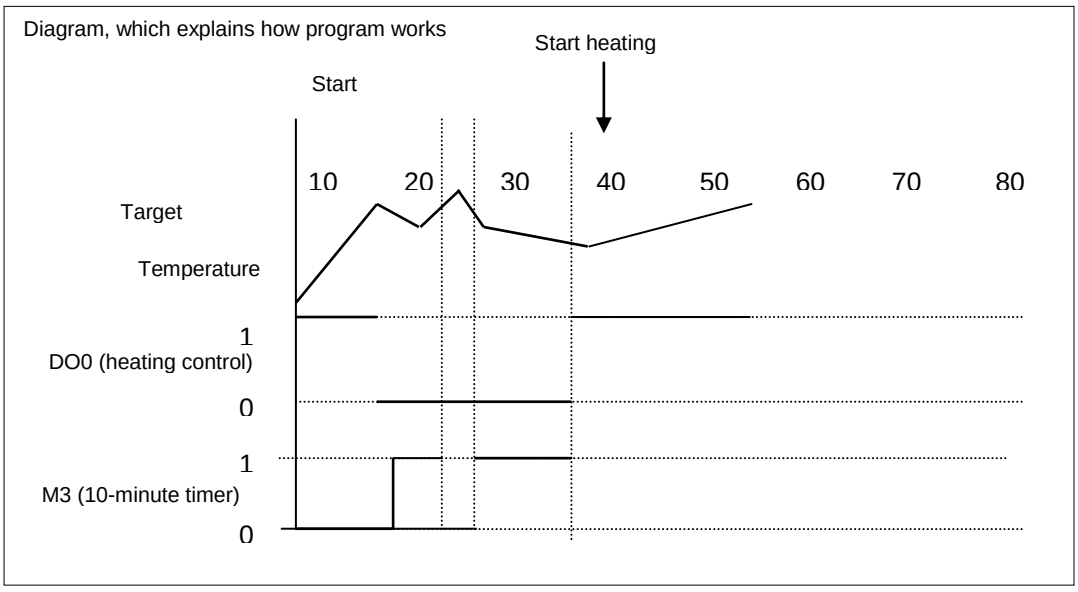
12.31 Thermostat



This example compares room temperature to set temperature. if the room temperature is lower than the set temperature, puts heating on. If the room temperature is higher or equal to the set temperature, program puts off heating. If the room temperature is 1C° colder than the set temperature, the program starts ten-minute delay. If during this ten-minute delay temperature rises above the set value, counter stops.
If after 10 minutes delay temperature is still below the set value, turn heating ON.

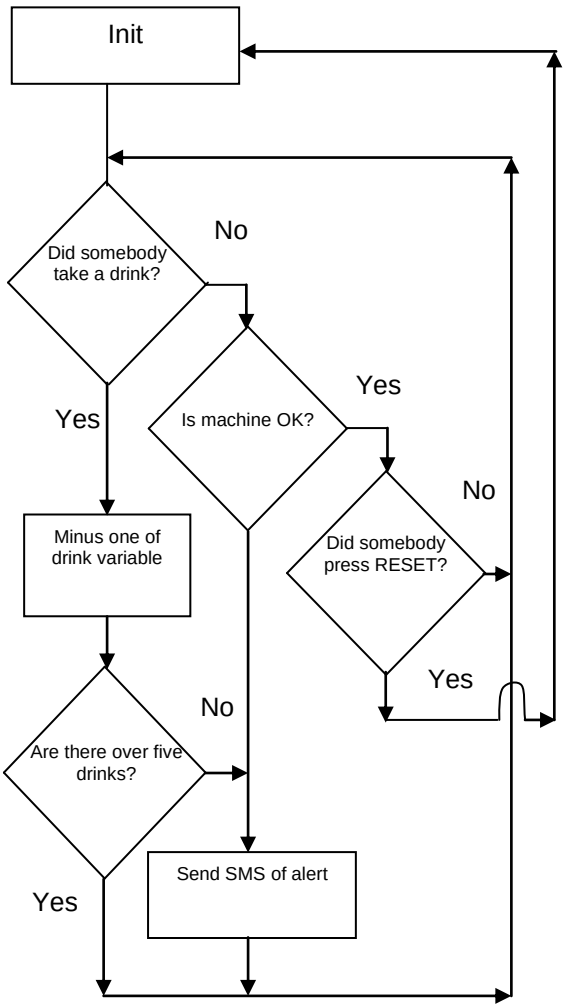
Line 1: When program starts, init variables.
Line2: If program gets message “$SET TEMP=xx”, reset delay counter.
Line 3: If temperature is higher or equal to target temperature, close heating.
Line 4: If temperature is higher or it is the target temperature, set temperature, when ten-minute delay starts.
Line 5: If temperature is temperature, when “start ten-minute delay”, starts delay.
Line 6: If ten minutes delay has started and program has got pulse, increment WM2.
Line 7: If program get minute pulse and temperature is higher than the setting ten minutes start delay, restart delay.
Line 8: If ten minutes is ago of temperature is lower or it is “start ten-minute delay”, start heating.
Line 9: If temperature is lower than the target temperature, start heating.
Line 10: If GSM-PLC received “$SET TEMP” message, scale temperature parameter
to AD converter value (0..4000).

```
Line 1:  '!M1' M1=1 WM0=20 WM0.20 WM0+1000 M2=0 WM2=0
Line 2:  '($SET TEMP)' M0 M2=0 WM2=0
Line 3:  '(AI1>WM0#AI1=WM0)' DO0=0
Line 4:  'AI1=WM0#AI1>WM0' M2=1 WM1=WM0 WM1-20
Line 5:  'M2&(AI1<WM1#AI1=WM1)' M3=1
Line 6:  'M3&P1' WM2+1
Line 7:  'M3&WM1<AI1&P2&!WM1=AI1' WM2=0 M3=0
Line 8:  'M3&WM2=600&(AI1=WM1#WM1>AI1)' M3=0 WM2=0 M2=0
Line 9:  'AI1<WM0&!M2' DO0=1
Line 10: 'M0=1' WM0.20 WM0+1000
```



12.32 Vending machine

This example sends SMS-message to declared number, if there is less than five bottles left or there is some trouble in the vending machine. If the door of the vending machine is open, system doesn't send any alerts. SMS-message "$INFO" returns the "drinks left" information.



Program:

Line1: Initialise variables
Line 2: If drink 1, 2 or 3 was selected, remove one of WM0
Line 3: If drink 4 was selected, remove one of WM1
Line 4: If drink 5 was selected, remove one of WM2
Line 5: If GSM-PLC received SMS: "$INFO"
Line 6: Send back status SMS
Line 7: If there are five bottles of drink 1, send the warning SMS
Line 8: If there are five bottles of drink 2, send SMS
Line 9: If there are five bottles of drink 3, send SMS
Line 10: If there are no bottles of drink 1, send SMS
Line 11: If there are no bottles of drink 2, send SMS
Line 12: If there are no bottles of drink 3, send SMS
Line 13: If there are some troubles in the drink machine, send SMS

Line 1: '!M0#DI7' M0=1 WM0=60 WM1=20 WM2=20
Line 2: '(DI0=1#DI1=1#DI2=1)&WM0>0' WM0-1
Line 3: 'DI3=1&WM3>0' WM1-1
Line 4: 'DI4=1&WM4>0' WM2-1
Line 5: '($INFO)' M1=1
Line 6: 'M1=1' M1=0 "Drink 1: %WM0 Drink 2: %WM1 Drink 3: %WM2" 254
Line 7: WM0=5&!DI5' "There are five bottles left of drink 1, 2, 3" 0
Line 8: 'WM1=5&!DI5' "There are five bottles left of drink 4" 0
Line 9: 'WM2=5&!DI5' "There are five bottles left of drink 5" 0
Line 10: 'WM0=0&!DI5' "There are no bottles left of drink 1, 2, 3" 0
Line 11: 'WM1=0&!DI5' "There are no bottles left of drink 4" 0
Line 12: 'WM2=0&!DI5' "There are no bottles left of drink 5" 0
Line 13: 'DI6=1&!DI5' "Error detected" 0

If you found any errors in these examples, please let us know!

If you have good examples, please suggest so we can update the manual.

12.33 Pump station monitoring system DEMO Application

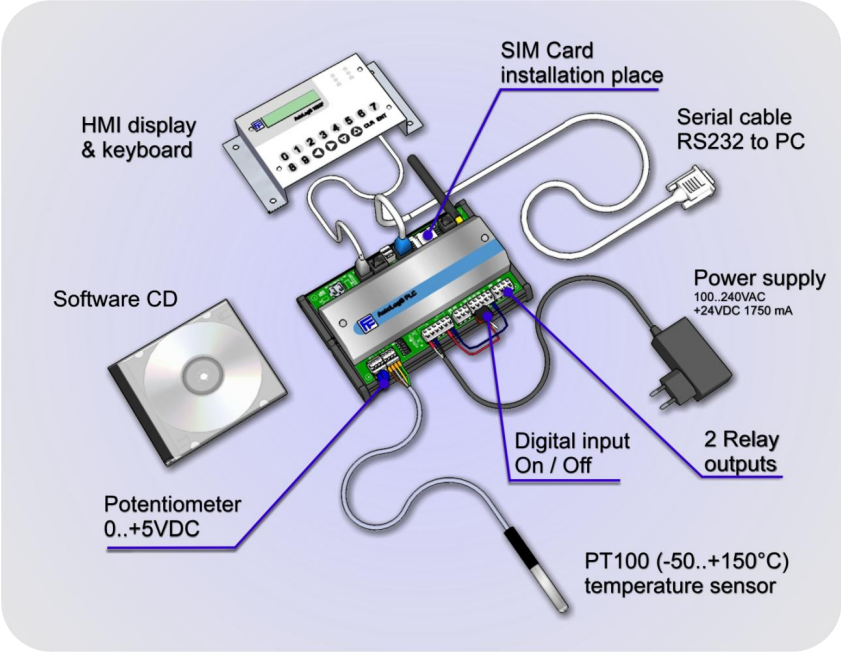
This demo application can be downloaded also from our distributor download site!

```
; GSM_PUMP_MONITORING DEMO PROGRAM
; VERSION 1.00 DATE 20th April 2012
; AUTHOR: AM / FF-Automation Oy
; Water pump station monitoring only, program doesn't control pumps, ask other demo if
; you need pump controlling!
; For DEVICE: AutoLog GSM-8 or AutoLog GSM-16
; AI0 =surface level (4-20mA->0-X meters), AI1= Pump1 current (4-20mA->0-Y A)
; Pump start count and operating times, surface level and digital input status and
; alarms
; Indusoft Web Studio SCADA with DB2SMS GSM driver using PostgreSql database interface


; Program Version
; ------------------------------

'M50#!M50' AO222=1 ; Version number can be placed e.g. on AO222

; variable, which is hardly used in other purposes.
; you can also use AO223 for program type and AO224 for subtype 1, etc.
; proper version handling is VERY important!

; Empty lines reserved for program updates (without changing the line ; numbers of other
; programming lines)
; These lines are useful if you need to download program changes
; remotely through GSM network later on.

.empty
.empty
.empty

; DI0 = High water level limit switch (NC= Normally Closed),
; 0 = alarm , 1 = normal state, DI0=!M0
; DI1 = Pump 1 heat relay alarm (NC), 0 = alarm, 1 = normal state ,
; DI1=!M1
; DI2 = Pump 1 control switch state, Manual (0) / Auto (1)
; DI3 = Pump 1 Operating state, 1 = Running 0 = Stopped , DI3=M3


; 0.0 INITIALISATIONS
;---------------------------------------------------------------------
; M2 = Pump 1 Manual (0) / Auto (1)= DI2

'DI2' M2
;Pump 1 Manual (0) / Auto (1) state to (binary memory number 2) M2

'M50#!M50' WM50=AI0 WM51=AI1 WM52=AI2 WM53=AI3 ; analog inputs raw
; values are copied to WM50-53

; 0.1 GSM modem reset at clock 2359
;-------------------------------------
'CLK=2359' M90
'M90=1' RO30=1 ;resets gsm modem, system program sets it back to 0 automatically, just
;in case the modem gets jammed.

; 1.0 HIGH LEVEL ALARM AND NORMALIZATION
;---------------------------------------------------------------------
; DI0 = High water level limit switch (NC= Normally Closed), 0 = alarm , 1 = normal
state, DI0=!M0
; M0 = High water level alarm bit, value: 1 = alarm (DI0=0) , 0 = normal state (DI0=1)

'!DI0S3' M0=1                ; ALARM state, 3 seconds delay for water waving, note: NC
                             ;input!
'DI0S120' M0=0               ; NORMALIZATION state, 120 seconds delay, for hysteresis
;(so that the alarm state doesn't repeat too often)
.empty                       ; Empty lines are reserved for program updates through
;sms.
.empty

; 2.0 HEAT RELAY ALARM (PUMP 1 FAULT) AND NORMALIZATION
;---------------------------------------------------------------------
; DI1 = Pump 1 heat relay alarm (NC), 0 = alarm, 1 = normal state , DI1=!M1
; M1 = Pump 1 heat relay alarm bit, value: 1 = alarm (DI1=0) , 0 = normal state (DI1=1)
```

```
'!DI1S3' M1=1                                    ; ALARM state 1 second delay "just in
case", note: NC tulo!
'DI1S120' M1=0                                   ; NORMALIZATION delay 120 seconds for
"hysteresis"
.empty

;3.0 PUMP 1 (P1) START COUNT AND OPERATING TIME
;-----------------------------------------------------------------

;3.1 P1 24H START COUNTER (24h) and P1 TOTAL START COUNTER
;-------------------------------------------------------------

; DI3 = Pump 1 Operating state, 1 = Running 0 = Stopped
; M3 = Pump 1 Operating state, 1 = Running 0 = Stopped
; WM10 = P1 START COUNTER (today)
; P1 TOTAL START COUNTER
; WM12 = P1 TOTAL START COUNTER low byte 0-9999
; WM13 = P1 TOTAL START COUNTER high byte 10000x
; P1 TOTAL START COUNTER = (10000 x WM13) + WM12 (Counted in SCADA!), Note: WM (Word
Memories) has 16 bits, so the maximum is 65535

'DI3S1' M3
'M3=1' WM10+1 WM12+1
'WM12>9999' WM13+1 WM12=0
.empty

;3.3 P1 24H OPERATING TIME and P1 TOTAL OPERATING TIME
;-----------------------------------------------------

; DI3 = Pump 1 Operating state, 1 = Running 0 = Stopped
; WM14 = P1 OPERATING TIME SECONDS (today)
; WM15 = P1 OPERATING TIME MINUTES (today)
; WM16 = P1 OPERATING TIME HOURS (today)
; WM20 = P1 OPERATING TIME SECONDS TOTAL
; WM23 = P1 OPERATING TIME IN SECONDS sssss today

'DI3&P1' WM14+1 WM20+1 WM23+1 ;Seconds (today), TOTAL and Seconds sssss today.
'WM14>59' WM15+1 WM14=0       ;minutes (today)
'WM15>59' WM16+1 WM15=0                ;hours (today)
.empty

; P1 TOTAL OPERATING TIME
; WM20 = P1 OPERATING TIME SECONDS TOTAL
; WM21 = P1 OPERATING TIME MINUTES TOTAL
; WM22 = P1 OPERATING TIME HOURS TOTAL HHHHH 0-65535 hours

'WM20>59' WM21+1 WM20=0       ;minutes TOTAL
'WM21>59' WM22+1 WM21=0                ;hours TOTAL
.empty

; 3.5 P1 RESETTING 24H COUNTERS
;------------------------------------------------------

; M80 = CLOCK TRIGGERED VARIABLE
; WM11 = P1 START COUNT LAST 24H HISTORY CLOCK 7:00 - klo 7:00
; WM60 maximum SMS count per day, safety limit

'CLK=0700' M80
'M80=1' WM11=WM10 WM10=0 WM60=0; P1 start count move and reset, commands are executed
from left to right order.
.empty

; P1 Operating time 24h reset

; WM14 = P1 OPERATING TIME SECONDS (today)
; WM15 = P1 OPERATING TIME MINUTES (today)
; WM16 = P1 OPERATING TIME HOURS (today)

; WM17 = P1 OPERATING TIME SECONDS (last 24h history)
; WM18 = P1 OPERATING TIME MINUTES (last 24h history)
; WM19 = P1 OPERATING TIME HOURS (last 24h history)
; WM23 = P1 OPERATING TIME IN SECONDS sssss today
; WM24 = P1 OPERATING TIME IN SECONDS sssss (last 24h history)

'M80=1' WM17=WM14 WM18=WM15 WM19=WM16 WM24=WM23 WM14=0 WM15=0 WM16=0 WM23=0; P1
operating time move and reset
.empty

; 4.0 START COUNT & OPERATING TIME ALARMS
;---------------------------------------------------------------
```

```
.empty
.empty   ; These are made in SCADA system
.empty

; 5.0 SMS SENDINGS
;---------------------------------------------------------------

; 5.11 PACKING BINARY STATES TO ONE 16-BIT WORD MEMORY AND SENDING

'M50#!M50' WM99=M0  ;SPECIAL COMMAND WHICH PACKS BINARY MEMORIES M0-M15 TO ONE 16-BIT
WORD MEMORY
.empty

; M0 = High water level alarm bit, value: 1 = alarm (DI0=0) , 0 = normal state (DI0=1)
; M1 = Pump 1 heat relay alarm bit, value: 1 = alarm (DI1=0) , 0 = normal state (DI1=1)
; M2 = Pump 1 Manual (0) / Auto (1)= DI2
; M3 = Pump 1 Operating state, 1 = Running 0 = Stopped

; WM50-WM53 analog input raw values (0-4000) Note! 12-bit Analog inputs are calibrated
to 0-4000 not 0-4096 for easier handling.

; WM60 maximum SMS count per day, safety limit
; WM11 = P1 START COUNT LAST 24H HISTORY CLOCK 7:00 - klo 7:00
; WM12 = P1 TOTAL START COUNTER low byte 0-9999
; WM13 = P1 TOTAL START COUNTER high byte 10000x

; WM24 = P1 OPERATING TIME IN SECONDS sssss (last 24h history)
; WM22 = P1 OPERATING TIME HOURS TOTAL HHHHH 0-65535 hours

; 5.1 ALARM MESSAGE
; ----------------------------

; Alarm message is send when any of the alarm bit changes its value to ; alarm state or
normal state.
; After the GSM-PLC is booted, only active alarm generates sms (M0-
; M127 are reseted when GSM-PLC boots up)
; Alarm activation delay should be the same for every alarm so after
; the boot up maximum 1 alarm sms is send.
; DB2SMS format: Dn,,Var(n),Var(n+1),Var(n+2),Var(n+3)
; DB2SMS is driver between SCADA and GSM-PLCs. Look more from DB2SMS
; documentation!

; ALARM is send when any of the alarm bit changes its value to alarm
; state or to normal state.

'(M0=0#M0=1#M1=0#M1=1)&WM60<10' WM60+1 M110
'M110=1&WM60<10' "D9,,1,%WM99,%WM50,%WM51,%WM52,%WM53,%WM11,%WM12,%WM13,%WM24,%WM22" 0

; D9,,1 one in here informas for SCADA system that this is alarm message so it is not
shown in daily report view
.empty

; 5.2 P1 REPORT MESSAGE (Send once every day at 07:00)
;-----------------------------------------------

; Same information is send as in Alarm message

'M80=1&WM60<10'WM60+1
"D9,,0,%WM99,%WM50,%WM51,%WM52,%WM53,%WM11,%WM12,%WM13,%WM24,%WM22"0 ; P1 tiedot
.empty
; ...."D9,,0 zero in here means that this is report message and it is shown in SCADA in
report view

; 6.0 SET PUMP 1 TOTAL START COUNT & TOTAL OPERATING TIME WITH SMS
;-------------------------------------------------------------------------
; WM12 = P1 TOTAL START COUNTER low byte 0-9999
; WM13 = P1 TOTAL START COUNTER high byte 10000x

'($P1)' M70
'M70=1&WM0<10000' WM12=WM0 WM13=0 WM22=WM1  "SET: P1 TOTAL START COUNT = %WM12, P1 TOTAL
OPERATING TIME = %WM22 HOURS" 254
'M70=1&WM0>9999' "SET FAILED!: P1 = START COUNT MAXIMUM is 9999" 254
.empty

; 7.0 REPORT SMS QUERY
;-------------------------------------------------------------------------
'($REPORT)' M72
'M72=1' "StartCount-24h=%WM11,-
Tot=%WM22h,Level=%AI0m,HLimAlarm=%M0,HeatRelay=%M1,Man/Aut=%M2,GsmSign(0-30)=%RO98" 254
```

13 ControlMan GSM-8 DEMO KIT APPLICATION -MANUAL



Order code: 900990 AutoLog ControlMan GSM-8 Demo Kit
(Ask also Indusoft Web Studio GSM-8 Demo Kit!)

13.1 Introduction

GSM-8 ControlMan demo kit is sold to customers who want to get familiar with the AutoLog GSM-PLC and ControlMan interface and maybe demonstrate the technology to their own customers. There is ready made application program which shows some features of GSM-PLC and AutoLog ControlMan. It needs to be noted that the GSM demo Kit's GSM application program is not written for beginners, because it shows the advanced features like HMI controls, time-tables and GPRS communication.

13.2 Features:

- Adjustable Low and High alarm levels for potentiometer input
- Adjustable Low and High alarm levels for temperature input
- Digital input alarms (DI0 and DI2) when input has been ON for five seconds
- Timetable control for Digital Outputs 0 and 1
- Manual output control for Digital output 0 (SMS)
- HMI & keypad
- Data Logging & File transfer to Controlman Server
- 2 Counters (each input acts also as counter input)
- Call in/Output on (Digital output 3)
- Data by request to cell phone
- Data transfer and Alarms to ControlMan service

13.3 HMI:

- 7 different views
- Password protected alarm level change
- Alarm status info in HMI leds
- Counter reset

13.4 Factory Settings

In order to use GSM/GPRS modem you need to purchase SIM card from local operator.

PLC is shipped from the factory with setting “Internal modem activated”. This feature is controlled by DipSw4 in PLC board. By setting DipSw4 to ON position and resetting PLC, internal modem is switched OFF and connection to PC can be made.

- Default PIN code in PLC is 0000.
  When you insert your SIM card to PLC either switch PIN Code request OFF from your SIM card OR adjust PIN code in SIM CARD or in PLC to correct one. (See operation manual for how to change PIN code using GSM Programmer)

13.5 Getting Started

In order to use GSM/GPRS modem you need to purchase SIM card from local operator.

After powering up the unit you are able to use the device through HMI.

13.5.1 Initializing phone numbers

To Enable the SMS/GPRS usage, you need to initialize your phone number & GPRS settings to GSM-PLC

Phone number
Phone number places 0 & 1 are reserved for your phone numbers. Some SMS commands in demo program do not have incoming number check, so these commands can be sent from any number.

Notice that Num 20 is used by application program. Do not change this number. Otherwise the SMS connection to ControlMan service is lost.

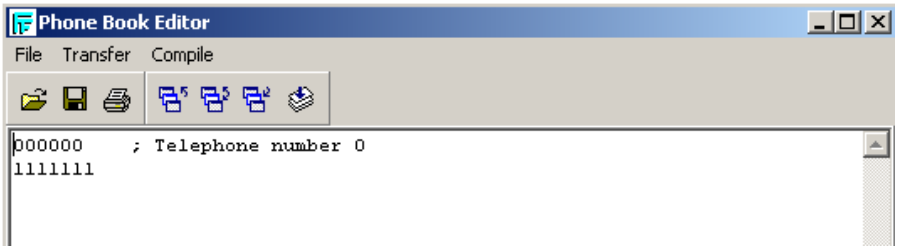
To change/initialize Phone numbers in Num 0 & Num 1, send following SMS to PLC
0000 NUM [n]=[your phone number]*0000 RUN*

Please use Country code with you phone number

OR

Use GSM Programmer to edit the phone numbers

- Enter your phone number to place 0 or 1 and transfer file to PLC



After transmission remember to start the PLC by pressing green start icon!

13.6 Configured SMS message controls

Configured "number free" SMS messages: These messages can be sent from any phone

- $REPORT* send status information back to the number where request came from.
  Response format
  "Temp: 25C, Pot: 300, Inputs: 0 1, outputs: 0 0 1 0, Counters: 0, 1"

- $RESET* Reset counter values. Acknowledgement sent to ControlMan & cell phone
  Response from PLC: "Counters Reset: CN0=0, CN2=0"

- $FTP=x=y* adjust data logging intervals
  x = log interval in minutes
  y = send interval in minutes
  Response from PLC: "Log interval= %WM35 min, Tx interval = %WM36 min"

- $POTENTIOMETER LEVELS=x=y=z* Set potentiometer levels in %.
  x = low level alarm
  y = high level alarm
  z = tank size (used to scale the percentage to AD-converter values) (Optional)
  Response from PLC: "new pot. alarm levels are: low xx, high yy"

- $MANUAL=x* Set DO0 mode to manual. X defines the output status
  x=0; output OFF
  x=1; output ON
  Response from PLC: "Mode changed to MANUAL. Output status = n"

  E.g. to set output ON send message **$MANUAL=1*** to GSM PLC

- $AUTO* return output 0 control back to automatic mode.
  Response from PLC: "Mode changed to Automatic"

Configured "number restricted" SMS messages: incoming phone number must be defined in Phnum0 or 1

- $CODE=x=y=z=n*; change security code for entering the alarm level pages. Can be changes only from phone numbers defined in Num 0 & Num 1. Response format :
  "NEW CODE is x y z n"

- $TIMETABLE (to be used only through ControlMan)

- $TT RESET* reset time tables, acknowledgement to ControlMan & cell phone. Reset command is possible to send only from phone numbers defined in Num 0 & 1
  Response from PLC: "Timetable Reset"

13.7 Configured Alarm messages:

- DIGITAL INPUT ALARM 0
  is sent to ControlMan after DI0 has been active for 5 sec

- DIGITAL INPUT ALARM 2
  is sent to ControlMan after DI2 has been active for 5 sec

- Potentiometer alarm LOW
  is sent to ControlMan after potentiometer value has dropped below alarm level for 3 sec.

- Potentiometer alarm HIGH
  is sent to ControlMan after potentiometer value has gone above high alarm level for 3

sec

- Temperature alarm. Temp now -20.5 C
  Combined High& Low alarm. This alarm has 10 sec filtering.

You can send these messages to your cell phone directly by changing the phone number place 20 in alarm lines in application program to one that has your phone number defined. However the alarm messages can also be forwarded to your cell phone & e-mail using ControlMan service.

13.8 Communication to ControlMan Service

PLC sends data to web Scada (ControlMan) using GPRS. The logging interval & send interval can be adjusted by sending command $FTP=x=y*, where x is logging interval in minutes and y is send interval.

Data can also be transferred to ControlMan using SMS, but as for big data quantities this is not cost effective way
SMS can be used as Backup way to get important data transferred.

$FTP=5=60* -> log data every 5 min, send once/hour

The default values are: log every 2 min, send every 10 minutes. Minimum logging & send interval is 1 min.

Data logging takes also place by event. Every time output changes status, data logging is triggered

One data logging adds 22 bytes to file, so using settings 2 minute logging/10 minute sending, one transferred file has
110 bytes of data+ header 30 bytes.

However notice that some GSM/GPRS operators may have minimum data block size of 1024kbytes, so even your data packed size is only 140 bytes, your data account may show 1024 bytes

Logged parameters are:
- Analogue inputs 0 & 1
- Counters 0 & 2
- Digital inputs 0 & 2
- Digital outputs 0..3
- Signal strength
- DO 0 control mode

Some data is sent to ControlMan using SMS messages. These are changes to

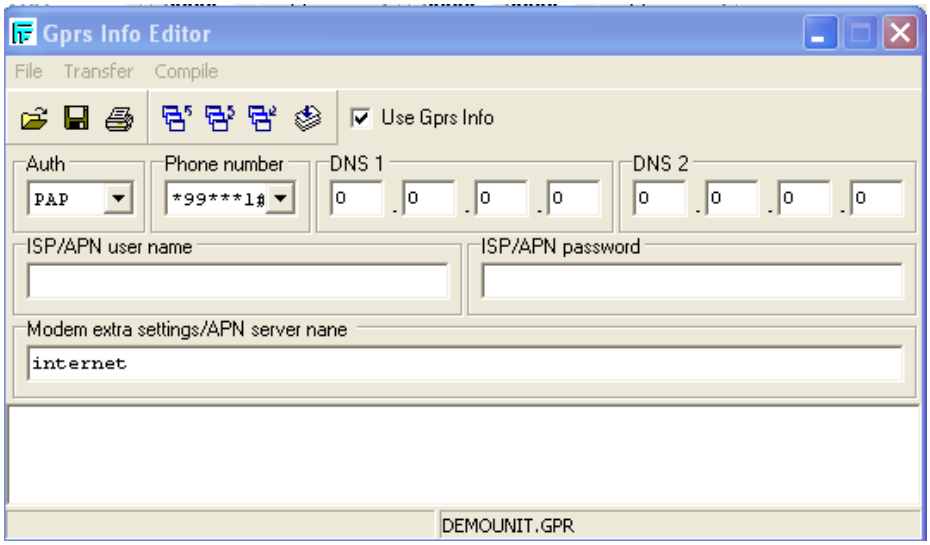
- Potentiometer alarm levels
- Temperature alarm levels
- Data logging settings
- Time table

Alarm messages are also sent over SMS. These alarms can be forwarded to email and/or SMS from ControlMan

13.9 GPRS settings

In order to get GPRS data working, you need to set correct parameters to GSM PLC.
These parameters depend on your GSM operator.
You need to find out the GSM Operator APN name (Access point name) and possible login username & password.

Easiest way to initialize these parameters to PLC is to use GSM Programmer. There's a GPRS editor where you need to fill this information and transfer it to PLC. Open GSM Programmer, connect PC to PLC, and feed in the parameters and send configuration to PLC.



Should you experience problems with GPRS (data cannot be seen in ControlMan, please check the RO37 status in GSM PLC (send SMS **0000 READ RO37***)

If value is other that 0 or 2, it is an error code.

From this error code we can see how far the establishing GPRS connection gets.

Typical error codes:

- 164 can not establish connection to GPRS network
  check APN, username & password
- 166 error in logging to FTP server
  contact FF-automation to enable FTP login

13.10 HMI

You can restore factory settings by pressing 0-key for 10 seconds. This adjusts the security code to 0000, potentiometer alarm levels to 20% and 80% and temperature alarm levels to 0 and 50°C

Use “C” and “D” key to scroll between screens

Long press “E” in “View /Change alarm levels” gets you to security code view. Enter code and press “F”
Wrong code will return you to Time&Date view. “B” key takes you back to code enter sequence
Correct code allows you to browse through alarm levels.

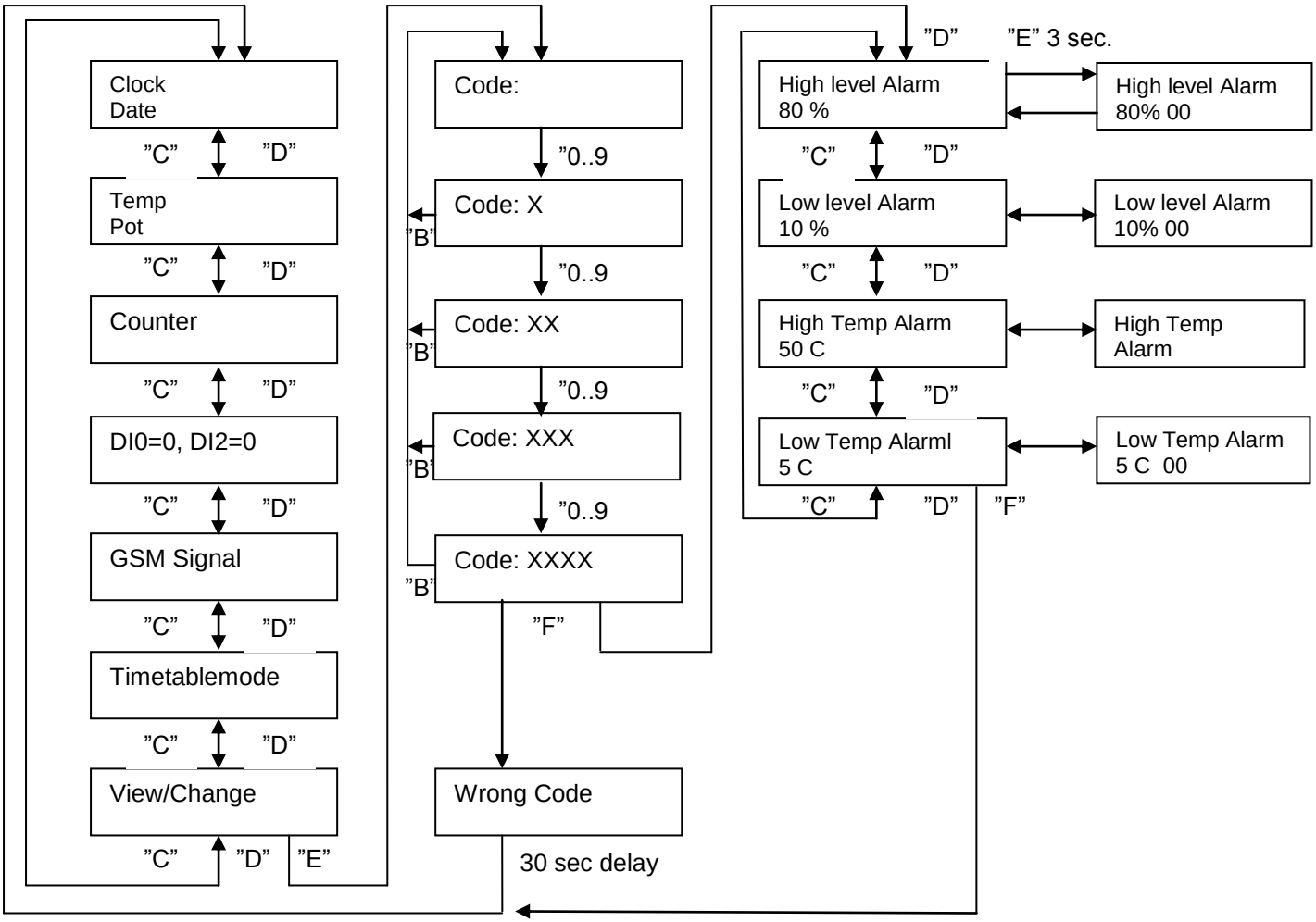
Long press “E will go to change parameter view. Enter new value and Press “F” to save the value
Pressing “E” will get you out without saving
To Exit from View /Change display, press long “F”

”B”= reset value

”C”= toggle sign

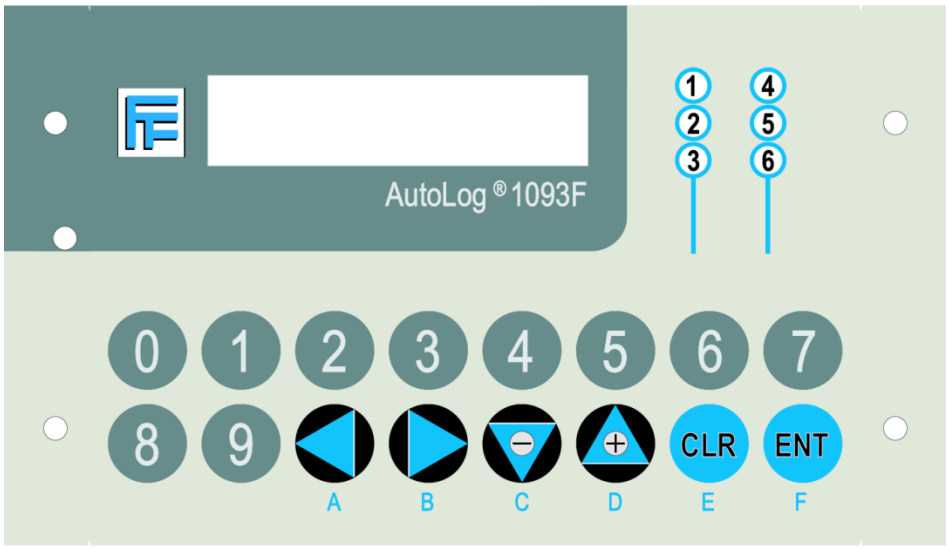
”E”= Exit without saving

”F”=Exit & save



13.10.1 HMI Leds

There are six leds in the HMI unit



Led 1 Digital input 0 alarm led
Led 2 Digital input 2 alarm led
Led 3 Temperature alarm led (high and low combined together)

Led 4 Potentiometer alarm high
Led 5 Potentiometer level between alarm levels (OK)
Led 6 Potentiometer low alarm

All but Potentiometer level OK led blinks after alarm has occurred.
Pressing “B” for 3 seconds acknowledges all alarms
-> If alarm condition is true, blink goes to steady ON
-> If alarm condition is not true anymore, led goes OFF

13.11 Trouble shooting

- You don't get response to preconfigured SMS messages

  a) Check the GSM PLC leds.
  - RUN led should blink on steady ½Hz frequency. If not, PLC is at STOP condition and should be started by sending **0000 RUN*** command

  - GSM-status led should blink (H1 or SERV). When led is blinking, modem has connection to GSM network.
  If modem status led is OFF, GSM-modem has no power
  -> Check DipSw4 and set it in OFF position and reset the power

  If led is steady ON, GSM modem has power, but has no connection to GSM network
  -> Check SIM card and PIN code setting
  -> Check connection from antenna to GSM-modem.
  -> Check the DipSw6 that selects the Modem band between EU/US band(EU bands: DipSw6=OFF)

  b) Check that number sending is active in your mobile phone.

- You don't get response to messages that checks the incoming Phone number
  a) Check that PLC is running (Run led blinks)
  b) Check the phone numbers defined in Phone Book places 0 & 1. Enter the phone number with the country code

- RUN led is blinking fast and outputs are always OFF:

  a) GSM-PLC is in power failure state. Check supply voltage.

- If Tx and Rx leds are not blinking:

  a) Check PIN code settings.

  b) Restart GSM modem or GSM-PLC.

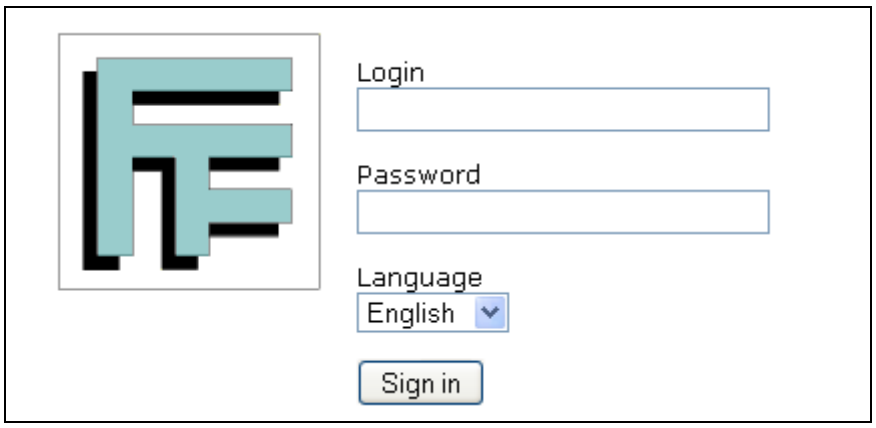
  c) Check connection between GSM modem and PLC (GsmProgrammer: PC and PLC).

- GsmProgrammer: Remember start changes with RUN command/button.

- GsmProgrammer: Check config settings (PLC type, COM port, PLC password, PLC PIN code, amount of program lines, amount of phone number etc.)

13.12 ControlMan Interface

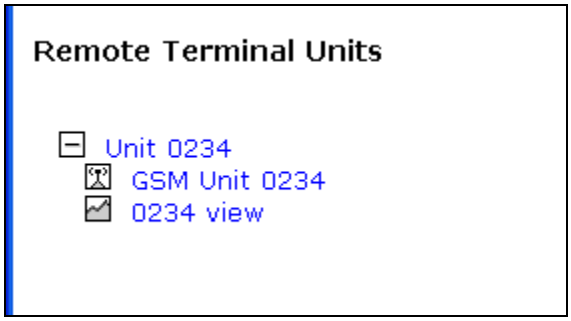
Login to page https:\\www.controlman.net



Enter login and password and press “Sing in” button

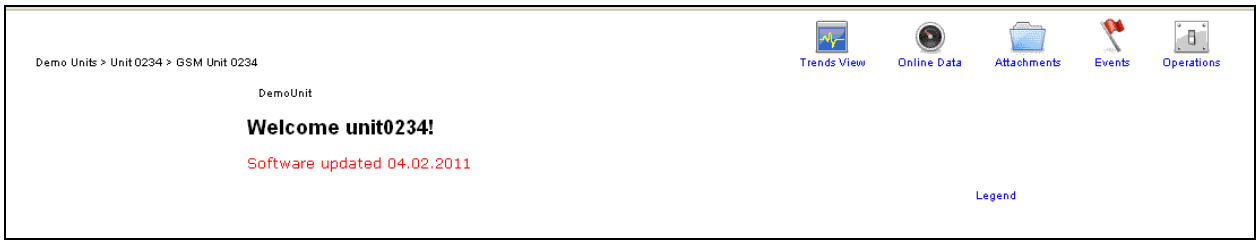
Following page will open

From the left navigation window you’re able to browse your units

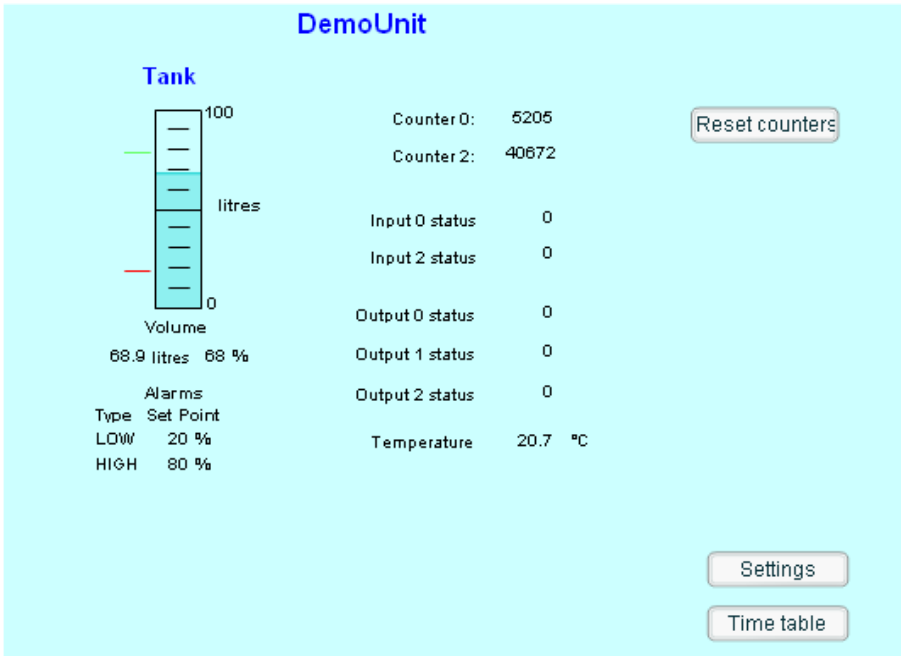


There are two views to choose from: Unit view & Flash view

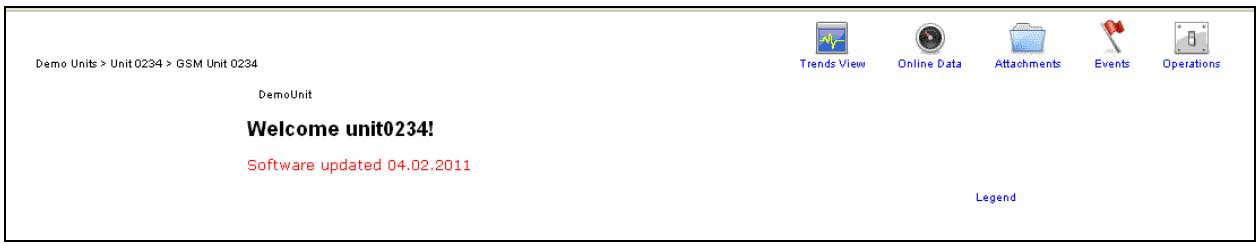
Unit view:



And Flash view:



13.13 Unit view:



In unit view there's 5 icons to choose from:

- Trends View
- Online Data
- Attachment
- Events
- Operations

13.13.1 Trend view:

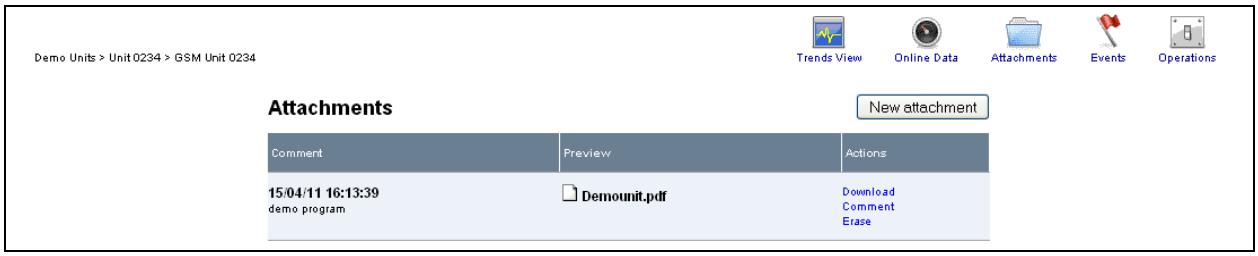
Trends View will display the tagged parameters:

13.13.2 Online Data -view

Online data View displays the latest data image from the unit

13.13.3 Attachment -view

Attachment View allows you upload & download documents and pictures to server

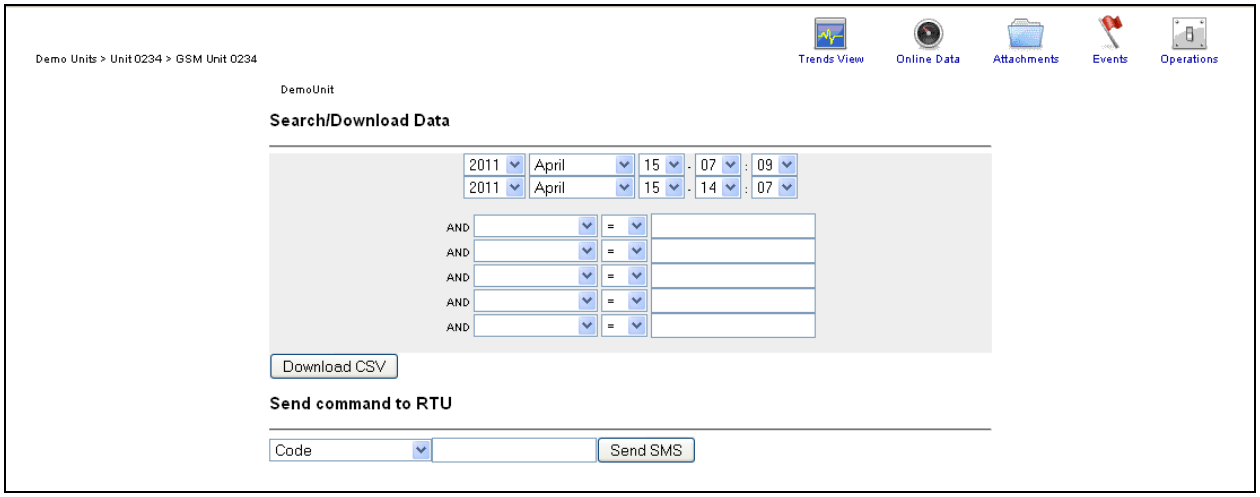


13.13.4 Events -view

Events view is disabled from end user. In this view you can browse the communication between Server & PLC.

13.13.5 Operation -view

Operation view allows you to download data to your local computer



The data comes in CSV format.

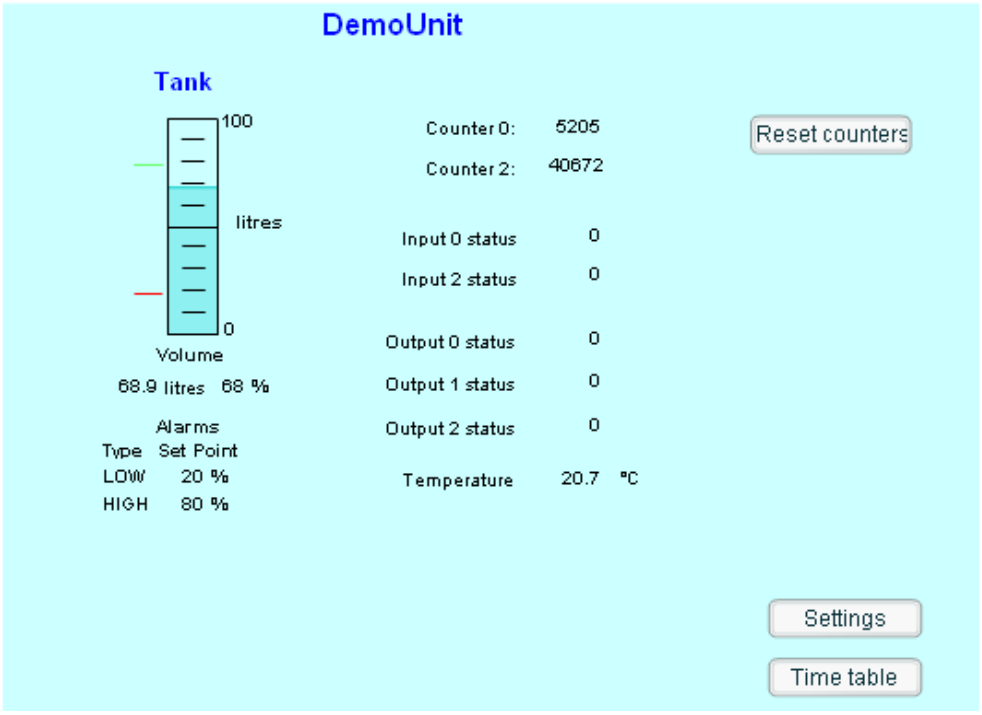
Also from this page, if you have high enough access level, you can send predefined messages to PLC.

13.14 Flash -view

The other Basic view available for this demo is Flash view
It has three screens that you can view:
-Main -view,
-Timetable -view and
-Settings -view

13.14.1 Main - view

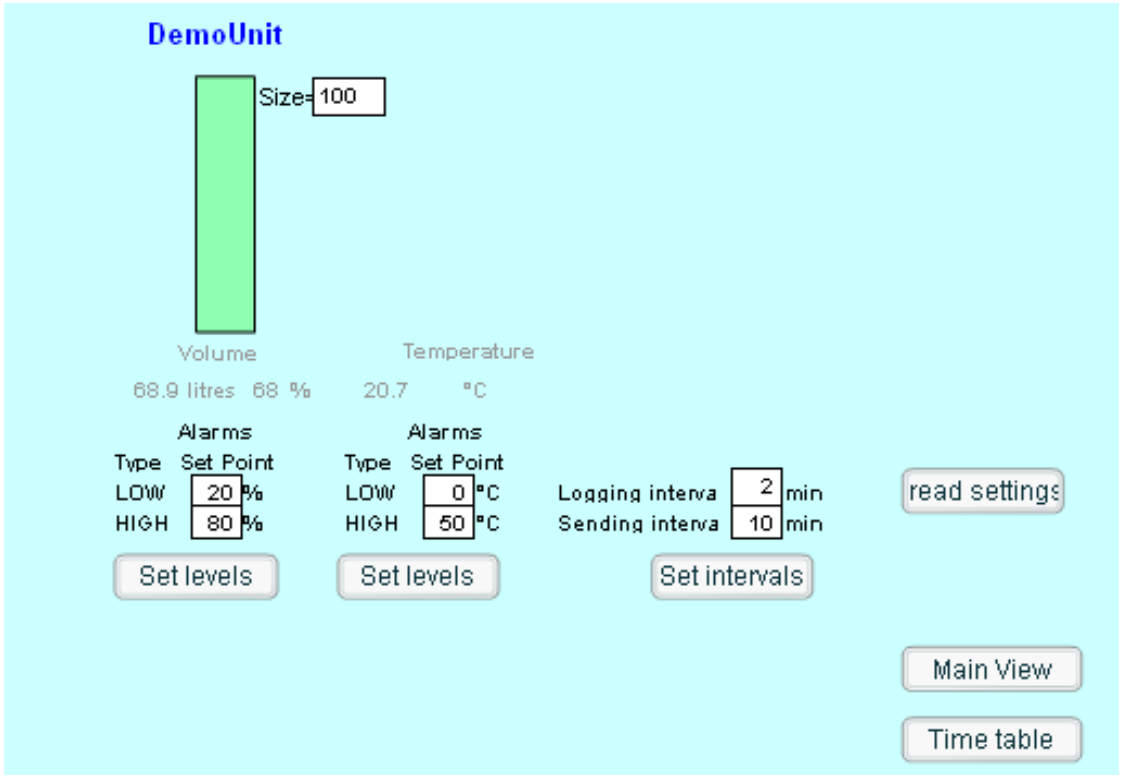
In main screen you see basic info from demo unit



Pressing “Reset counters” button will trigger an SMS to PLC that will reset the counters.
Buttons “Settings and “Time table” will take you different views

13.14.2 Settings –view

In settings view you can adjust the alarms levels and FTP parameters



13.14.3 Time Table -view

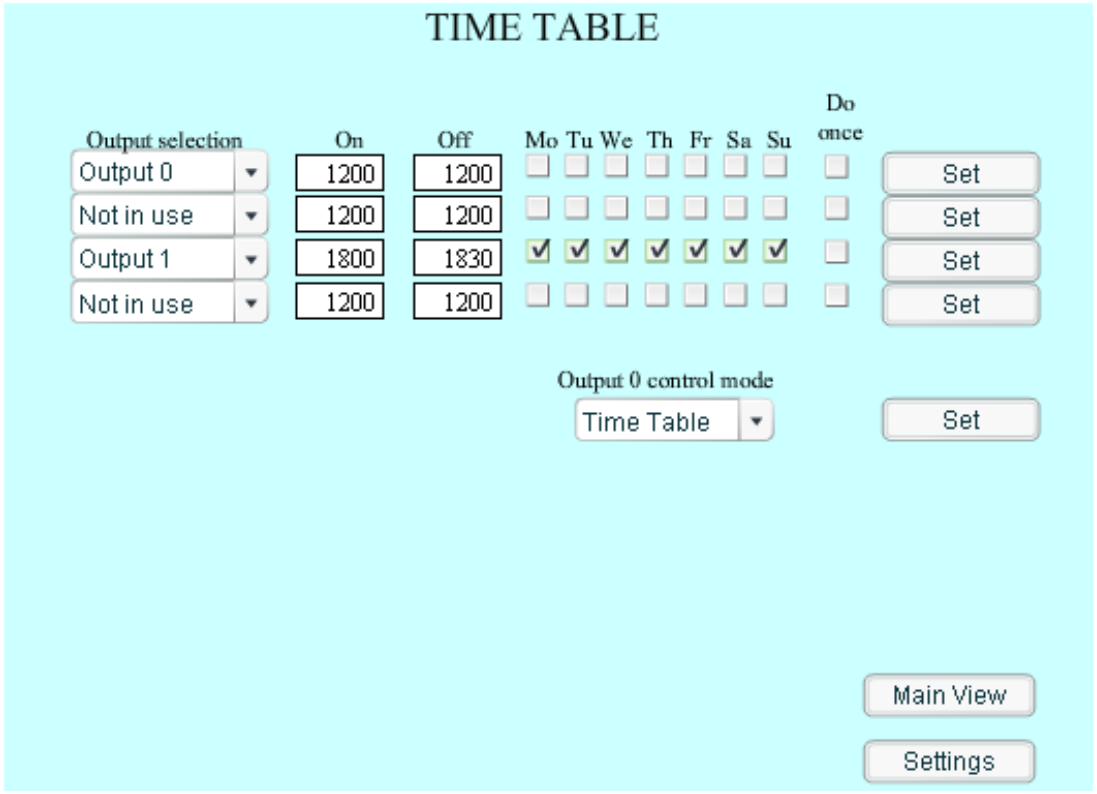
In time table view you have 4 time controls for Digital Outputs to choose from

In Output selection you select the output you wish control
On time defines when output goes active
Off time defines when output goes inactive
Weekday selection select week program

There's also Mode selection for output 0
Available modes are:

- Time Table
- Manual mode, output OFF
- Manual mode, output ON

This selection allows you to control output manually.
When you turn control mode back to Time Table, output starts to follow immediately
The Timetable definitions



13.15 APPENDIX A. PLC PROGRAM FOR DEMO-KIT

```
; Program: demo program
; Version: 1.0
; Date   : 12.4.2011
; Author : Jouni Paavonen
```
;; Additional Comments by Antti Moijanen

```
;Basic initializations in power up: display page 1
'!M127' M127=1 WM17=1


'N20' M29         ;if message in from ControlMan (num 20), set bit active
(M29)
```

;;Incoming phone number identification is done automatically and it is available during one ;;program cycle.
;;e.g. 'N20' condition is true if incoming phone number (N254) is same as phone book number 20.

```
;********** ANALOGUE INPUT CONVERSION ***********
;(potentiometer gives 4000 when turned left and 0 in right, so we converse the
value)
;Change AI from 4000..0 ->0..100            ; (4000-AI0)/40
'M1#!M1' WM20=4001 WM20-AI0 WM21=WM20 WM20/40         ;Do always
```

;;Syntax of the Division: WM[x] / Var16 -> WM[x] = integer, WM[x-1] = remainder
;; Commands are executed from left to right
;;WM20-AI0 , the result is put in WM20
;;WM20/40, the result is put in WM20. Remainder is put in WM19

```
;********* ANALOGUE INPUT ALARM LEVELS ***********
;Set alarms levels using SMS, trigger response to server (M20=1) and to Cell
phone (optional)
; WM120=low level
; WM121=high level
; WM124=Tank size Optional
;incoming message format: $POTENTIOMETER LEVELS=20=80=1300*

'($POTENTIOMETER LEVELS)' M0 WM120=WM0 WM121=WM1 WM112=WM2 M20=1
;3rd parameter is optional. If it's not included, keep the parameter unchanged
'WM112>0' WM124=WM112 WM2=0
;Send SMS only if message NOT from ControlMan service
'M0&!M29' "new pot. alarm levels are: low %WM120, high %WM121" 254

; WM122=low temperature level
; WM123= high temperature level
'($TEMP LEVELS)' M0 WM122=WM0 WM123=WM1 M20=1
;Send SMS only if message NOT from ControlMan service
'M0&!M29' "New temp. alarm levels are: low %WM122, high %WM123" 254
```

;;Look $ Self defined messages!

```
;Scale analogue alarm levels (engineering units to ad-converter units).
(P1=once/second)
;Ad converter values from 0..4095
;Potentiometer alarm levels 0=0 100=4000
;Convert engineering units to AD-units 0..100%->0..4000
'P1' WM255=WM120 WM255.40 WM130=WM255 WM255=WM121 WM255.40 WM131=WM255

;Temperature alarm levels -50=0 150=4000; modify values also for display
(WM132&WM133)
;modify set values for the display value 0..4000 -> -50..150
;if set value less than 0 C, set flag "negative" (M197, M198)
'P1&WM122<1000' WM255=1000 WM255-WM122 WM255/20 WM132=WM255 M197=1
'P1&WM122>999' WM255=WM122 WM255-1000 WM255/20 WM132=WM255 M197=0
'P1&WM123<1000' WM255=1000 WM255-WM123 WM255/20 WM133=WM255 M198=1
'P1&WM123>999' WM255=WM123 WM255-1000 WM255/20 WM133=WM255 M198=0
```

```
;*************** ANALOGUE INPUT ALARMS ***************
;compare potentiometer value to alarm levels
;there's 3 second hysteresis

'AI0<WM130' M70   ;Low alarm comparison

; time stamp:23 April 2011 9:08:20   "A3,230411090820,Potentiometer alarms
;LOW"

'M70s3' M71=1 "A3,%RO248%RO247%RO255%RO250%RO251%RO252,Potentiometer alarm
LOW" 20 ;

;HMI led control for Pot. alarm LOW
'M71&!M72&(P1XDO253)#M72' DO253     ;blink HMI led if alarm active, steady
light if acked
```

;;M72 = Pot. low level ack
;;DO248…DO255 Controls led of HMI control panel, 1…8
;;DO253 controls LED6

```
;************ HIGH LEVEL COMPARISON ****************
'AI0>WM131' M75   ;High alarm comparison
'M75s3' M76=1 "A4,%RO248%RO247%RO255%RO250%RO251%RO252,Potentiometer alarm
HIGH" 20

;HMI led control for Pot. alarm HIGH
'M76&!M77&(P1XDO251)#M77' DO251     ;blink HMI led if alarm active, steady
light if acked

'AI0>WM130&AI0<WM131' DO252             ;steady led in no alarms

;Temperature alarm
'AI1>WM133' DO3               ;Output 3 ON If temperature exceeds the high level
'AI1<WM132 # AI1>WM133' M85
'M85s10' M86=1 "A5,%RO248%RO247%RO255%RO250%RO251%RO252,Temperature alarm. Temp now %AI1
C" 20
'AI1>WM132 & AI1>WM133' M87
'M87S10' M86=0
```

;;ERROR , this line should be'AI1>WM132 & AI1<WM133' M87

```
;HMI led control for Temperature alarm HIGH & LOW
'M86&!M88&(P1XDO250)#M88' DO250 ;blink when active & not acked, steady light when active
& acked
```

;;

85	temp over alarm levels
86	temp alarm
87	temp between alarm levels
88	temp alarm acked

;;

```
;********* DIGITAL INPUT ALARMS ********
'DI0S5' M80=1 "A1,%RO248%RO247%RO255%RO250%RO251%RO252,DIGITAL INPUT ALARM 0" 20
'DI2S5' M81=1 "A2,%RO248%RO247%RO255%RO250%RO251%RO252,DIGITAL INPUT ALARM 2" 20

;HMI led control for Digital input alarms
'M80&!M82&(P1XDO248)#M82' DO248 ;if alarm is active, blink led, if acked: steady ON
'M81&!M83&(P1XDO249)#M83' DO249         ;if alarm is active, blink led

;Acknowledge alarms
'DI251S2&(M80#M81#M71#M76#M86)' M82=1 M83=1 M72=1 M77=1 M88=1 ;Ack all alarms (LONG
press "B")
```

;;DI240-DI255 Reserved for (HMI) KEYPAD, B=DI251

```
'!DI0&M82' M80=0 M82=0                          ;When normalized & acked, clear alarm
'!DI2&M83' M81=0 M83=0
'AI0>WM130&M72' M71=0 M72=0
'AI0<WM131&M77' M76=0 M77=0
```

```
;*************** TIME TABLE **************
;There are 4 time tables to choose from. Each has 4 parameters:
;destination = if time table is true, where it will affect.
; 0 = not in use
; 1 = output 0
; 2 = output 1
;Start time = 0 refers to 12:00, 1439 refers to 11:59
;Stop time = 0 refers to 12:00, 1439 refers to 11:59
;Weekday info = bit 0: monday, bit 1:Tuesday, bit 3:Wednesday, etc...
;Parameters for 1st time table are AO109..AO113
```

;;RO250 Date and time information: hour
;;RO251 Date and time information: minute

```
;make minutecounter out of hour & minute registers. AO200 changes between 0..1439, 720 =
24:00

'M1#!M1' WM254=RO250 WM254/256 WM253=0 WM254.60 WM252=RO251 WM252/256 WM254+WM252
WM254+720 WM254/1440 AO200=WM253
```

;;Syntax for copying RO to WM
;; WM[x]=RO[y] ; WM[x]=RO[y]*256 + RO[y+1]
;;WM254=RO250 ; WM254 = RO250 * 256 + RO251

;;Division: WM[x] / Var16 -> WM[x] = integer, WM[x-1] = remainder

```
;Check for weekday. Load Weekday info for 1st time table
;unpack weekday control to bits 181..187
'M1#!M1' M181=AO112
;compare bits to weekday (RO249).
'(RO249=1&M181) # (RO249=2&M182) # (RO249=3&M183) # (RO249=4&M184) # (RO249=5&M185) ' M0

'((RO249=6&M186) # (RO249=7&M187) # M0) & AO200<720' M204=1           ;result in M204

;Do same comparison for all 4 weekday fields
'M1#!M1' M181=AO116
'(RO249=1&M181) # (RO249=2&M182) # (RO249=3&M183) # (RO249=4&M184) # (RO249=5&M185) ' M0
'((RO249=6&M186) # (RO249=7&M187) # M0) & AO200<720' M205=1

'M1#!M1' M181=AO120
'(RO249=1&M181) # (RO249=2&M182) # (RO249=3&M183) # (RO249=4&M184) # (RO249=5&M185) ' M0
'((RO249=6&M186) # (RO249=7&M187) # M0) & AO200<720' M206=1

'M1#!M1' M181=AO124
'(RO249=1&M181) # (RO249=2&M182) # (RO249=3&M183) # (RO249=4&M184) # (RO249=5&M185) ' M0
'((RO249=6&M186) # (RO249=7&M187) # M0) & AO200<720' M207=1

;combine weekday and minute data
'M204 & (AO200>AO110 & AO200<AO111)' M200         ;1st time table
'M205 & (AO200>AO114 & AO200<AO115)' M201
'M206 & (AO200>AO118 & AO200<AO119)' M202
'M207 & (AO200>AO122 & AO200<AO123)' M203

;Combine weekday and minute data & control selection
;result of all 4 ttbl:s for output 1
'((AO109=1&M200) # (AO113=1&M201) # (AO117=1&M202) # (AO121=1&M203))' M140
;result of all 4 ttbl:s for output 2
'((AO109=2&M200) # (AO113=2&M201) # (AO117=2&M202) # (AO121=2&M203))' M142


;********* MANUAL MODE *********
;controls output 0
'($MANUAL)' M0 AO90=1 AO91=WM0 M62=1              ;Change mode and send info also to CM
'M0&!M29' "Mode changed to MANUAL. Output status = WM0" 254
;********* AUTO MODE *********
'($AUTO)' AO90=0 M62=1
'M0&!M29' "Mode changed to Automatic" 254


;************ SET TIME TABLE PARAMETERS *************
'(N0#N1#N20#N21) & ($TIMETABLE)' M0
;              Destination   Start  Stop     Weekday
          Destination
'M0 & WM0=0' AO109=WM1 AO110=WM2 AO111=WM3 AO112=WM4 M197=0 M79=1     ;         0 = not
in use
'M0 & WM0=1' AO113=WM1 AO114=WM2 AO115=WM3 AO116=WM4 M198=0 M79=1     ;         1 = R1
'M0 & WM0=2' AO117=WM1 AO118=WM2 AO119=WM3 AO120=WM4 M199=0 M79=1     ;         2 = R2
'M0 & WM0=3' AO121=WM1 AO122=WM2 AO123=WM3 AO124=WM4 M200=0 M79=1     ;
```

```
; If changed, Send values back to server
'M79' M79=0 "D19,%RO248%RO247%RO255%RO250%RO251%RO252,%AO109@15" 20


'(N0#N1#N20#N21)&($TT RESET)' M0 AO109=0 AO110=0 AO111=0 AO112=0 AO113=0 AO114=0 AO115=0
AO116=0 AO117=0 AO118=0
'M0' AO119=0 AO120=0 AO121=0 AO122=0 AO123=0 AO124=0 M79=1
'M0&!M29' "Timetable Reset" 254

;*********** OUTPUT CONTROL FOR DO0 & DO1 ********************
;  auto mode   #   manual mode & ON
'(M140&AO90=0) # (AO90=1&AO91=1)' DO0                                   ;DO0 control
'M142' DO1                                                              ;DO1 control

;*********** REPORT REQUEST IN ***********
'($REPORT)&!M29' M0           ;request from cell phone
'($REPORT)& M29' M20=1        ;request from server
'M0' "Temp: %AI1, Pot: %WM20 %, Inputs: %DI0 %DI2, outputs: %DO3 %DO2 %DO1 %DO0,
Counters: %CN0, %CN1" 254

;********* SET NEW SECURITY CODE ***************
'($CODE)&(N0#N1#N20#N21)' M0 M0=1 WM18=50     ;code can be changed only from phone
numbers 0 & 1
'M0&!M29' "NEW CODE is %WM0 %WM1 %WM2 %WM3" 254             ;send confirmation to cell
phone
'M0' "D37,,%WM0@3" 20                                       ;send data to server
;Save new code to WM50..WM53
'M0' WM0+48 WP18=WM0 WM18+1 WM1+48 WP18=WM1 WM18+1 WM2+48 WP18=WM2 WM18+1 WM3+48
WP18=WM3

;************ display control **************
;WM17 is used as display pointer


'WM17=1&P1' "Clock: %RO250:%RO251:%RO252 Date : %RO248.%RO247.%RO255" 255
'WM17=2&P1' "Temp = %AI1 C      Pot  = %WM20 %" 255
'WM17=3&P1' "Counter 0= %CN0                    " 255
'WM17=4&P1' "DI0=%DI0, DI2=%DI2                    " 255
'WM17=5&P1' "GSM signal level %RO98" 255
'WM17=6&P1& AO90=0' "Timetablecontrol   Auto         " 255
'WM17=6&P1& AO90=1' "Timetablecontrol   Manual out %AO91" 255
'WM17=7' "View/Change     Alarm levels" 255

'WM17=30' "Factory settings restored" 255

'WM17=16' "CODE:"255
'WM17=17' "CODE: x" 255
'WM17=18' "CODE: xx" 255
'WM17=19' "CODE: xxx" 255
'WM17=20' "CODE: xxxx" 255
'WM17=21' "WRONG CODE" 255

'WM17=100&P1' "Low level alarm  %WM120 %" 255
'WM17=101&P1' "High level alarm %WM121 %" 255
'WM17=102&!M197&P1' "Low temp alarm:   %WM132 C" 255
'WM17=102&M197&P1'  "Low temp alarm:  -%WM132 C" 255
'WM17=103&!M198&P1' "High temp alarm   %WM133 C" 255
'WM17=103&M198&P1'  "High temp alarm  -%WM133 C" 255

'WM17=200&P1' "Low level alarm:  %WM120 % %WM99" 255
'WM17=201&P1' "High level alarm %WM121 % %WM99" 255
'WM17=202&P1&!M197&!M199' "Low temp alarm:  %WM132 C  %WM99" 255
'WM17=202&P1&!M197&M199'  "Low temp alarm:  %WM132 C -%WM99" 255
'WM17=203&P1&!M198&!M199' "High temp alarm  %WM133 C  %WM99" 255
'WM17=203&P1&!M198&M199'  "High temp alarm  %WM133 C -%WM99" 255
'WM17=202&P1&M197&!M199'  "Low temp alarm: -%WM132 C  %WM99" 255
'WM17=202&P1&M197&M199'   "Low temp alarm: -%WM132 C -%WM99" 255
'WM17=203&P1&M198&!M199'  "High temp alarm -%WM133 C  %WM99" 255
'WM17=203&P1&M198&M199'   "High temp alarm -%WM133 C -%WM99" 255

;*********** HMI KEY CONTROL **************
'DI252=1&WM17<200' WM17-1     ;Display DOWN
'WM17=0' WM17=7               ;Check for underflow
'WM17=99' WM17=103            ;Adjust steps underflow

'DI253=1&WM17<200' WM17+1     ;Display UP
'WM17=8' WM17=1               ;Check for overflow
'WM17=104' WM17=100           ;Adjust steps overflow
```

```
;************ Enter Password  routine *************
;long Press "E" starts the Password entering routine
;M7= code entering mode, WM10=key press cntr, WM99=pointet to save data, WM7= timeout
'DI254s3 & WM17=7' WM17=16 M7=1 WM10=1 WM99=55 RO209=0 WM7=30
'DI251&M7' WM10=1 WM99=55 RO209=0 WM7=30                    ;return key "B", start from
beginning
; Read code from the keyboard (RO209 is the latest key data in ASCII format)
; WM10 is the char counter
'RO209>0&RO209<65&M7' RO208=0 WP99=RO208 RO209=0 WM17+1 WM10+1 WM99+1 RO210=0 WM7=30

'DI255&WM10=5' WM10=0 M7=0    ;"F"=Enter -> WM10+1 (make comparison) M7=0 (exit set
mode)

;Compare Codes
'M7=0&WM55=WM50&WM56=WM51&WM57=WM52&WM58=WM53' M8 ;If Codes are same M8 is true

'M7=0&M8' WM17=100                                  ;Code OK -> display mode = 100

'M7=0&!M8' WM17=21 M5=1                             ;Code NOT OK 10 sec display 21
'M5S10' M5=0 WM17=1 WM10=0                          ;then return to start screen

'P1&WM7>0' WM7-1                                    ;Decrement T_O timer once/sec
'M7&WM7=0' WM17=1 WM10=0 M7=0                       ;Time out

;********** Enter new levels *************
'DI254S2&WM17>99&WM17<200' WM17+100 M60             ;Long press goes to "change value
disp"
'M60=1' M57=1 RO209=0 WM99=0                        ;M57=set active, , WM99=new value

; number pressed, add key to WM99
'M57 & RO209>0 & RO209<65 & WM99<101' RO208=0 WM103=RO208 WM103-48 RO209=0 WM99.10
WM99+WM103 RO210=0

; Press "B" ->Start over, reset set value
'M57 & DI251' WM99=0

; Press "C" -> toggle sign
'M57 & (DI252=1XM199)' M199

; Press "F" -> save and return
'M57 & DI255=1&WM17<202' M57=0 WM17-100 WM102=WM17 WM102+20 WP102=WM99 WM111=0 M20=1      ; ok WM102=para
'M57 & DI255=1&M199&WM17>201' WM17-100 WM102=WM17 WM102+20 WM255=50 WM255-WM99 WM255.20
WP102=WM255 M20=1
'M57 & DI255=1&!M199&WM17>201' WM17-100 WM102=WM17 WM102+20 WM255=50 WM255+WM99 WM255.20
WP102=WM255 M20=1

; Press "E" -> return from set without saving
'M57 & DI254=1& WM17>199' M57=0 WM17-100 WM111=0; cancel

;Restore Factory settings LONG PRESS "0"
;log   & send   Code for change settings =0000  Pot alarm levels Temp alarms levels 0 &
50C
'DI240S10' WM35=2 WM36=10 WM50=48 WM51=48 WM52=48 WM53=48 WM120=20 WM121=80 WM122=1000
WM123=2000 WM17=30 M101=1
'M101S10' M101=0 WM17=1 ;after 10 sec display go back to basic view

;**** Exit from view/set ****
'DI255S2 & !M57 & WM17>99' WM17=1

;********* RESET COUNTERS ***********
;Either SMS $RESET or 2 sec press on key "A", send info to Server also (M62)
'($RESET)#DI250s5' M0 CN0=0 CN2=0 M62=1
'M0&!M29' "Counters Reset: CN0=%CN0, CN2=%CN2" 254          ;Send ack message back

;*********** CALL IN ****************
; Control output but calling to unit
'(T0#T1)' DO2s10                        ;10 sec pulse to output

;********* FTP LOGGING & Transfer ***********
'DO0=1#DO0=0#DO1=1#DO1=0#DO2=1#DO3=1#DO3=0' M62=1           ;log & send on event in
outputs
'($FTP)' M0 WM35=WM0 WM36=WM1 M20=1                         ;Get new FTP parameters
'M0&!M29' "Log interval= %WM35 min, Tx interval = %WM36 min" 254
'WM35=0' WM35=1                                             ;Log interval cannot be 0
min
'WM36=0' WM36=5                                             ;Tx interval limited to min.
5 min
```

```
'P2' WM30+1 WM31+1            ;increment logging timer and transfer timer once/minute
(P2)

'WM30=WM35#WM30>WM35' M60 WM30=0                              ;Timer baser logging (M60)
'M60=1#M62=1' "%WM21,%AI1,%CN0,%CN2,%DI0,%DI2,%AO90,%DO0@3,%RO98" 245

;Transfer
'WM31=WM36#WM35>WM36' M61 WM31=0                              ;Timer based Transmit (M61)
'M61=1# M62S30' M62=0 RO38=8 RO37=1 ;

;Parameter sending to SERVER *** SMS ***
'M20S30' M20=0 "D12,,%WM120@4,%WM35@1" 20 ;Send alarm levels WM120..WM123 + WM35&WM36 to
server
;backup sending if enabled
;'RO37>2&!(RO37=130)' "D0,,%WM21,%AI1,%CN0,%CN2,%DI0,%DI2,%AO90,%DO0@3,%RO98" 20
'!M128&RO98<30' M128=1 "SERIAL: demo0234" 21                  ;For identification purposes
```

13.16 MEMORY MAP for GSM- Demo Application Program

address	WM	Address	AO	Address	M
0	SMS Parameter 0	90	DO0 control	0	common memory
1	SMS Parameter 1	91	DO0 status in Manual mode	1	
2	SMS Parameter 2				
3	SMS Parameter 3	109	ttbl 1 destination	5	Code not OK flag
4	SMS Parameter 4	110	ttbl 1 start		
		111	ttbl 1 stop	7	"enter code"-mode
7	Timeout counter for entering code	112	ttbl 1 weekday	8	Codes match
		113	ttbl 2 destination		
10	counter for code characters	114	ttbl 2 start	20	send settings SMS to server
		115	ttbl 2 stop		
17	display selection	116	ttbl 2 weekday	29	message in from CM
18	pointer to code	117	ttbl 3 destination		
19	reserved	118	ttbl 3 start	57	"enter value"-mode
20	Potentiometer eng units	119	ttbl 3 stop		
21	potentiometer AD-units	120	ttbl 3 weekday	60	timer baser logging
		121	ttbl 4 destination	61	timer baser transmit
30	log Timer	122	ttbl 4 start	62	event based saving&transmit
31	Tx timer	123	ttbl 4 stop		
		124	ttbl 4 weekday	71	pot low alarm
35	log set value			72	Pot low alarm ack
36	Tx set value	200	Minute counter		
				76	pot high alarm
50	Code 0			77	pot high alarm ack
51	Code 1				
52	Code 2			79	Send Timetable to CM
53	Code 3			80	dig input 0 alarm
				81	Dig input 2 alarm
55	entered code 0			82	Dig input 0 alarm ack
56	entered code 1			83	dig input 2 alarm ack
57	entered code 2				
58	entered code 3			85	temp over alarm levels

99	Set value from HMI
102	pointer for parameter change
103	register for getting pressed key data
112	Tank size
120	Pot. Alarm level Low (Eng)
121	Pot. Alarm level high (Eng)
122	Temp. Alarm level low (AD)
123	Temp alarm level high (AD)
124	Tank size
130	Pot alarm level low (AD)
131	Pot alarm level high (AD)
132	temp alarm level low (Eng)
133	temp alarm level high (Eng)
252	common memory
253	common memory
254	common memory
255	common memory

86	temp alarm
87	temp between alarm levels
88	temp alarm acked
101	factory setting display
127	basic initializations
128	send identification info
140	ttbl status for output 0
142	ttbl status for output 1
181	weekday Mo from ttbl
182	weekday Tu from ttbl
183	weekday We from ttbl
184	weekday Th from ttbl
185	weekday Fr from ttbl
186	weekday Sa from ttbl
187	weekday Su from ttbl
197	negative value for "temp lo"
198	negative value for "temp hi"
199	"C" pressed- toggle sign
200	ttbl1 weekday&time
201	ttbl2 weekday&time
202	ttbl3 weekday&time
203	ttbl4 weekday&time
204	ttbl1 weekday check
205	ttbl2weekday check
206	ttbl3 weekday check
207	ttbl4 weekday check

Address	DI	Address	DO	
0	Pulse input 0	0	output 0	controlled by TTBL's OR manually
		1	output 1	Controlled by TTBLs only
2	Pulse input 2	2	output 2	On when temp above high limit
		3	output 3	10 sec. ON pulse when call recieved from PhNum 0 or PhNum 1
240	HMI button "0"			
241	HMI button "1"	248	HMI led 0	
242	HMI button "2"	249	HMI led 1	
243	HMI button "3"	250	HMI led 2	
244	HMI button "4"	251	HMI led 3	
245	HMI button "5"	252	HMI led 4	
246	HMI button "6"	253	HMI led 5	
247	HMI button "7"			
248	HMI button "8"			
249	HMI button "9"			
250	HMI button "A"			
251	HMI button "B"			
252	HMI button "C"			
253	HMI button "D"			
254	HMI button "E"			
255	HMI button "F"			

HARDWARE & ACCESSORIES

14 GSM-PLC HARDWARE

14.1 List of Main Features:

Look Main features -chapter

14.2 GSM-PLC hardware - Comparison chart:

Code	Designation	Analog Inputs*	Digital Inputs	Digital Outputs	Relay Outputs	Analog Outputs	I/O Expansion	Free SER PORTS	Opt. SER Plug in Module PiM	Other
900684	AL GSM-4	-	4	1	2	-	N	1**	Y	Integrated GSM modem
900693	AL GSM-4 UNIT	-	4	1	2	-	N	1**	Y	+Enclosure, +power
980693	AL GSM-4 UNIT+HMI		4	1	2	-	N	1**	Y	Above +HMI
900951	AL GSM-8	(2*)	4	2	2	-	N	1	Y	Integrated GSM modem
900960	AL GSM-8 UNIT	(2*)	4	2	2	-	N	1	Y	+Enclosure, +power
980960	AL GSM-8 UNIT+HMI	(2*)	4	2	2	-	N	1	Y	Above +HMI
900501	AL GSM-16AGR	(8*)	4	-	4	-	N	1	Y	Integrated GSM modem
900480	AL GSM-16AGR UNIT	(8*)	4	-	4	-	N	1	Y	+Enclosure, +power
980480	AL GSM-16AGR UNIT+HMI	(8*)	4	-	4	-	N	1	Y	Above +HMI
900755	AL GSM-20AN	(8*)	8*	8	-	2	Y	1	Y	Needs optional external GSM modem! *AI 0-8 or DI8-24, one AI can be changed to 2 DI
ask!**	AL GSM-20AN UNIT	(8*)	8*	8	-	2	Y	1	Y	Includes external GSM modem, enclosure and power
ask!**	AL GSM-20AN UNIT+HMI	(8*)	8*	8	-	2	Y	1	Y	Above +HMI
900666	AL GSM GW	-	-	-	-	-	N	1	Y	Integrated GSM modem. No I/O, Modbus to GSM gateway or Wireless Wi-Fi I/O to GSM.
900667	AL GSM GW UNIT	-	-	-	-	-	N	1	Y	+ Enclosure + power
980667	AL GSM GW UNIT+HMI	-	-	-	-	-	N	1	Y	Above + HMI

*Analog inputs need optional Analog input modules.
** GSM-20 Unit is sold as custom unit because it has so many order parameters like I/O expansion cards, enclosure etc.

14.3 Special type AutoLog GSM-PLCs (Comparison chart):

Code	Designation	Analog Inputs	Digital Inputs	Digital Outputs	Relay Outputs	Analog Outputs	I/O EXPansion	SER PORTS	SER Plug in Module (PiM)**	Other
900140	AL GSM-10SP PLC	5***	4	4	-	-	-	2	(Y)	Mainly designed for cathodic protection.

**Support for SER Plug in Module (PiM) is place for different kinds of AutoLog modules. For example to get Ethernet (Modbus TCP) or RS-485 (Modbus RTU). Look more from serial Plug-in Modules page (PiM) further in this price list!

*** AutoLog 10 SP PLC doesn't need analog input modules. Analog inputs can be tuned in PLC software. 16-bit resolution, Input ranges can be selected point-wise using software. Possible input ranges are: 0-800mVDC, 0-3.2VDC, 0-50VDC and 0-100VDC. Surge protected and opto –isolated inputs..

14.4 AutoLog GSM-GW (Gateway)

14.4.1 GSM-GW Main Features:

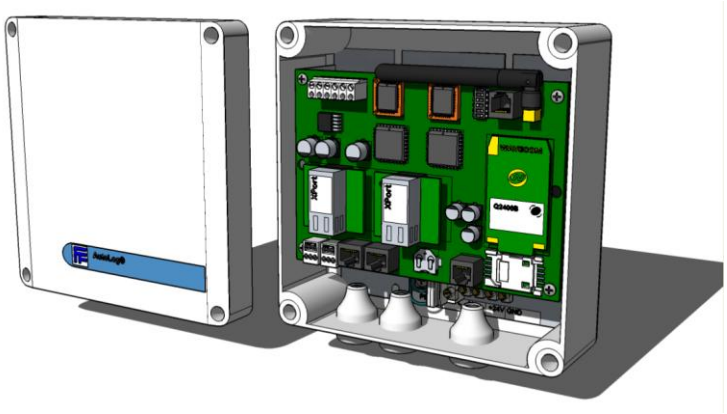
- **No I/O (can read data from serial port, or wireless I/O using Modbus)**
- Integrated GSM modem (for SMS, GPRS and call)
- 2 Serial ports
  - Ser 1 is reserved for GSM modem / programming
  - Ser 2 can be used as Modbus RTU M/S RS-232
  - Optionally Ser 2 can be used as RS-485 or Modbus TCP (Ethernet). Optional Wireless sensor master module can also be connected to this port.
- I2C interface for HMI (character based display & keypad)
- Stubby antenna (optionally stubby antenna can be changed to external antenna)
- SMA Antenna connector
- Operating power 10-30VDC or 12-24VAC
- Power output 24VDC
- Accumulator connection. Charge connection with deep discharge protection.
- 230VAC power supply is included only to GSM units!
- Real time clock & calendar
- 512 programming lines, 32 build-in PID controllers, time-tables
- data logging to FTP files (e.g. 1x 43000 or 4x 21000 or 8x 13000 Time-stamped records)
- incoming phone number identification, 254 phone numbers
- Operating temperature -30…+65°C, Storage temperature -40…+80°C
- Read more!

14.4.2 GSM-GW I/O:

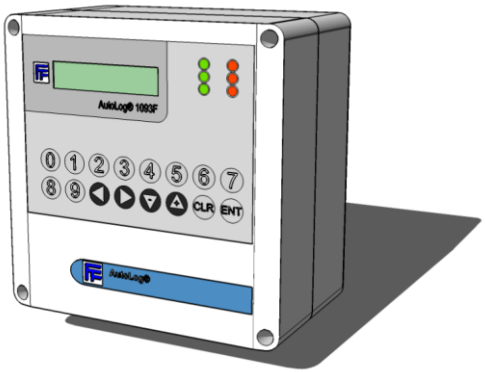
GSM-GW I/O	
I/O	There's no I/O in GSM-GW. It can be connected to Modbus (master/slave) device using Serial port 2.

14.4.3 GSM-GW Order codes:

900666 GSM-GW (case)



900667 GSM-GW Unit (Plug in Modules are optional)

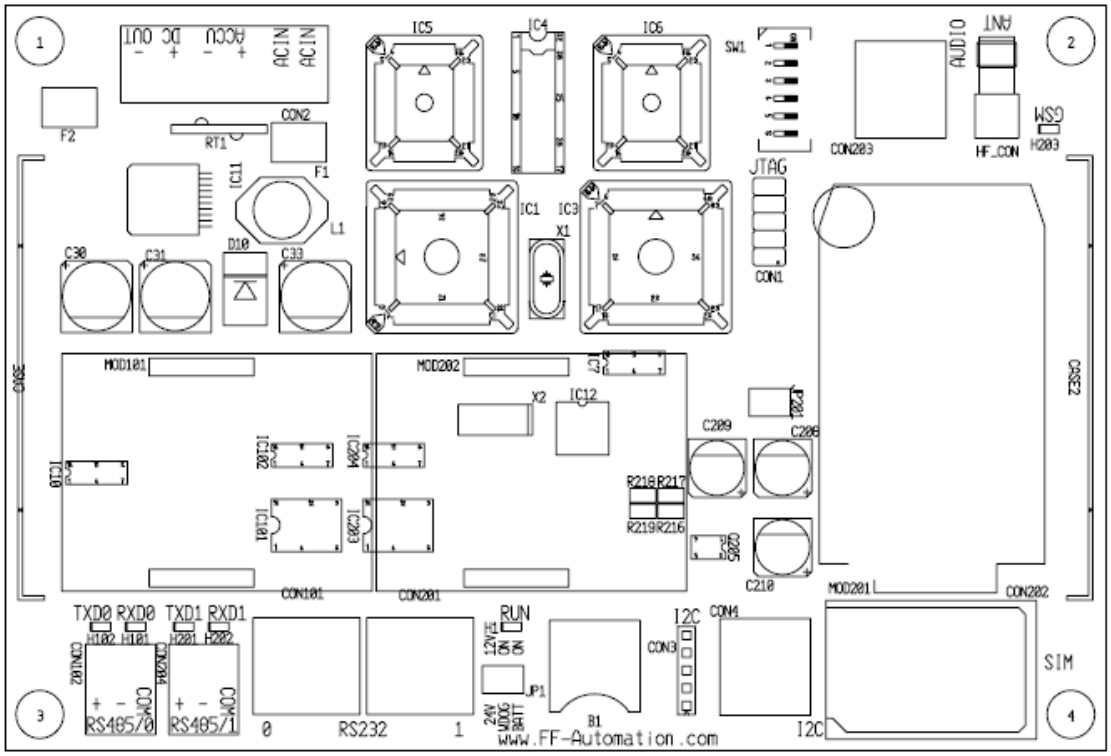


900667 GSM-GW Unit + HMI

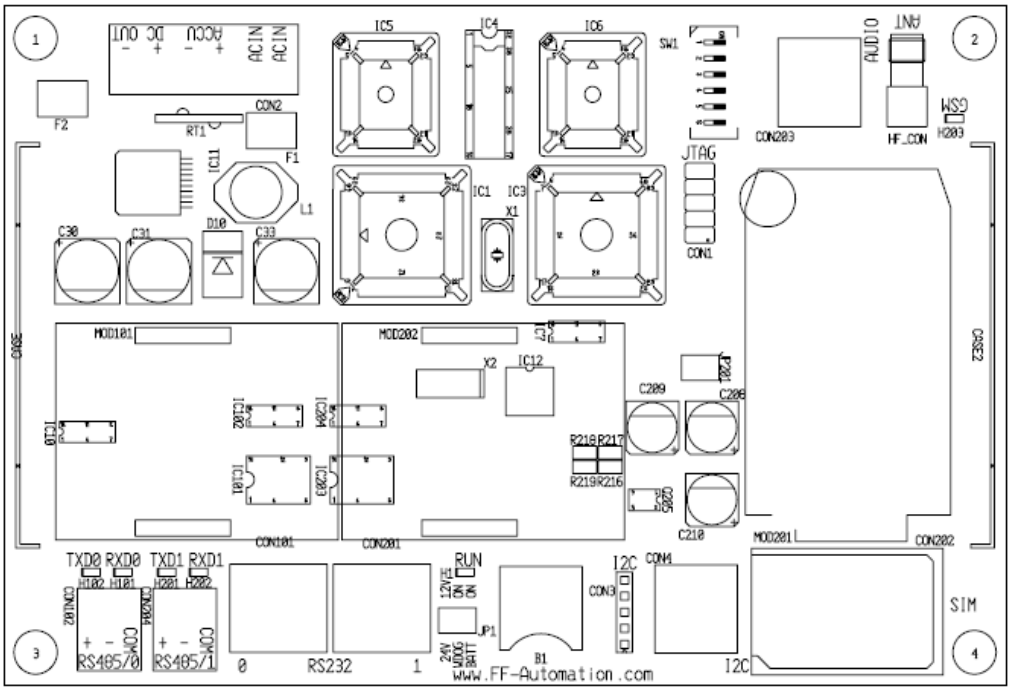
GSM-Gateway		
Code	**Designation**	**Description**
900666	GSM-GW (case)	IP20 metal cover, DIN rail mounting, stubby antenna
900667	GSM-GW Unit	ABS plastic box, IP65, power supply
980667	GSM-GW Unit + HMI	as above + HMI mounted on the cover, IP65
N/A	GSM-GW custom unit	Look customized GSM units!

- Power is only included to GSM UNITS! Please specify the power supply type!
- Stubby antenna is default, other antenna types can be ordered as options (Stubby antenna price will be deducted)!

14.4.4 GSM-GW circuit board connectors

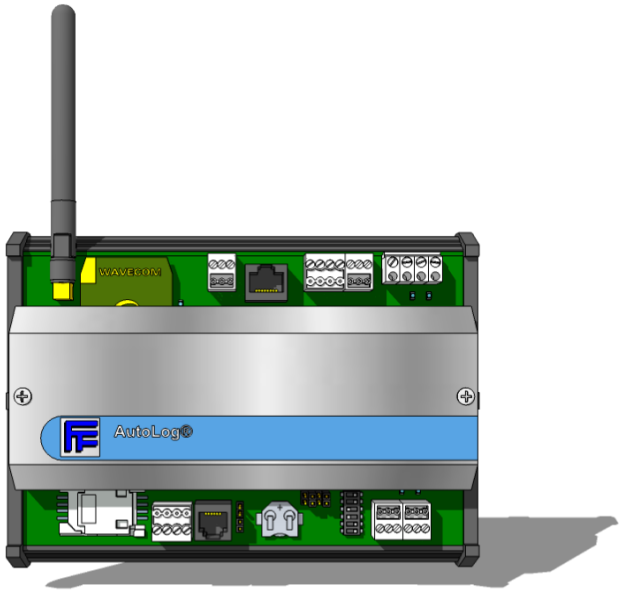


ID	Description
CON2	Power supply input, battery connection and Power output
CON3	I^2C port for display/keypad unit
CON201	RS-232 Serial connection, SER1
CON101	RS-232 Serial connection, SER2
CON102	RS-485 Serial connection, SER2, RS-485 needs optional RS-232 to RS-485 Plug-in Module (PiM)
CON203	Modem Audio
CON202	SIM card holder. Normally customer purchases SIM card locally.
MOD101	Plug-in Module (PiM) place for SER2
MOD102	Plug-in Module (PiM) place for SER1
HF_CON	GSM modem antenna connector
IC1	CPU processor
IC3	Address coder
IC5	Flash, system program
IC6	Flash, application program
B1	Backup lithium battery for retaining values in RAM (EEPROM) memory. Type: CR2032. 3 Volts.
SW1	DIP switches 1-6
JP1	Jumpers
JP1:1-2	Power fail info (ON= 9V, OFF=17V), When power fail is active RO36=1
JP1:3-4	Watch dog (ON= active, OFF= disabled)
JP1:5-6	Backup battery (ON=enabled, OFF=disabled)



14.4.5 AutoLog GSM-GW picture

14.5 AutoLog GSM-4



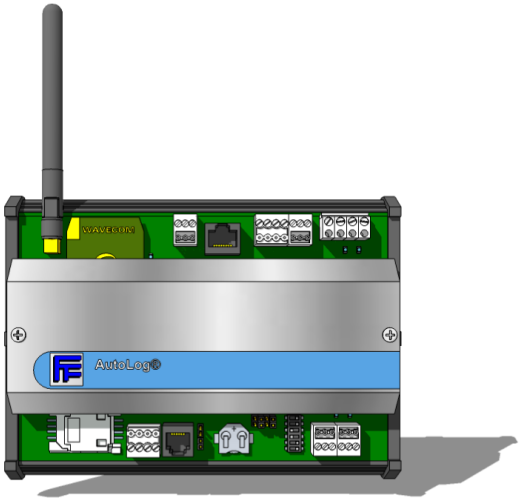
14.5.1 GSM-4 Main Features:

- **4 Digital Inputs, 1 Digital Output, 2 Relay Outputs**
- Integrated GSM modem (for SMS, GPRS and call)
- 1 Serial port
  - Ser1 port can be used for programming / Modbus / Modem
  - Mode is selected with DIP switch
- I2C interface for HMI (character based display & keypad)
- Stubby antenna (optionally stubby antenna can be changed to external antenna)
- SMA Antenna connector
- Operating power 10-30VDC or 12-24VAC
- Power output 24VDC
- Accumulator connection. Charge connection with deep discharge protection.
- 230VAC power supply is included only to GSM units!
- Real time clock & calendar
- 512 programming lines, 32 build-in PID controllers, time-tables
- data logging to FTP files (e.g. 1x 43000 or 4x 21000 or 8x 13000 Time-stamped records)
- incoming phone number identification, 254 phone numbers
- Operating temperature -30…+65°C, Storage temperature -40…+80°C
- Look more!

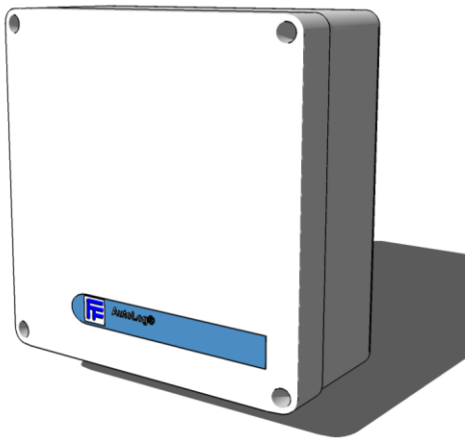
14.5.2 GSM-4 I/O:

GSM-4 I/O	
Digital Inputs	4 digital inputs, 24VDC, max 8mA, opto-isolated, PNP
Digital Output	1 digital output, 12-24VDC, 0.5A, NPN
Relay Outputs	2 relay outputs, 5A /250VAC 1A/30VDC

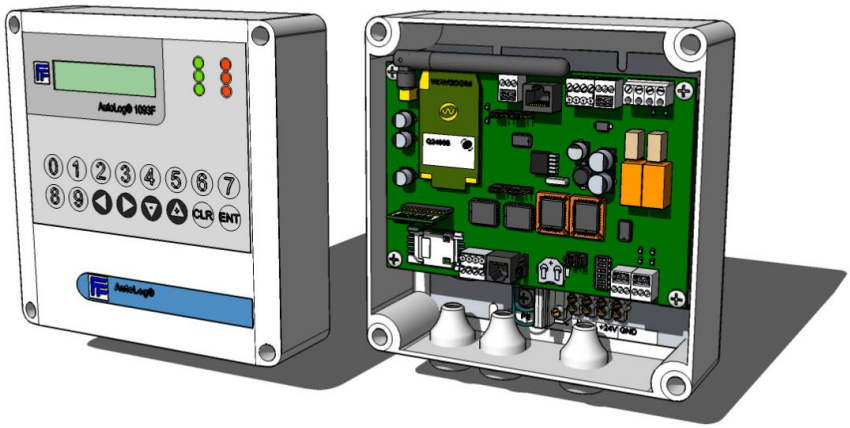
14.5.3 GSM-4 Order codes:



900684 GSM-4 (case)



900693 GSM-4 Unit

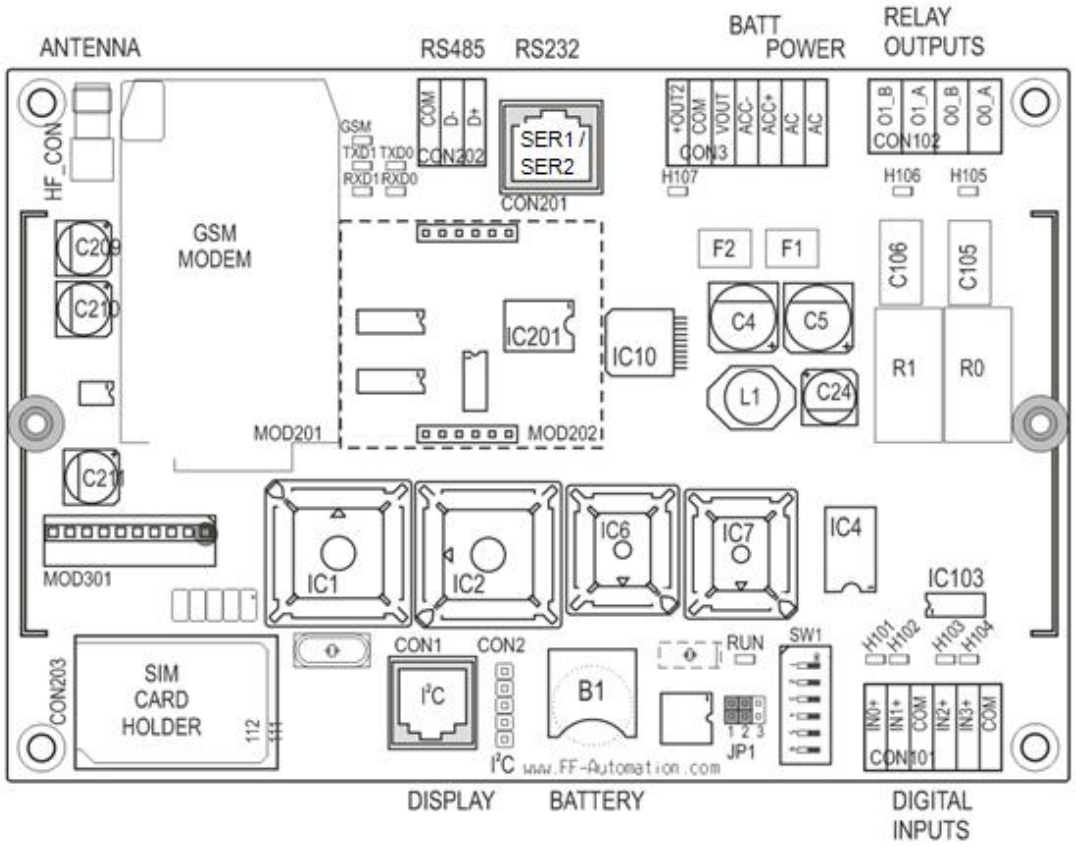


980693 GSM-4 Unit + HMI

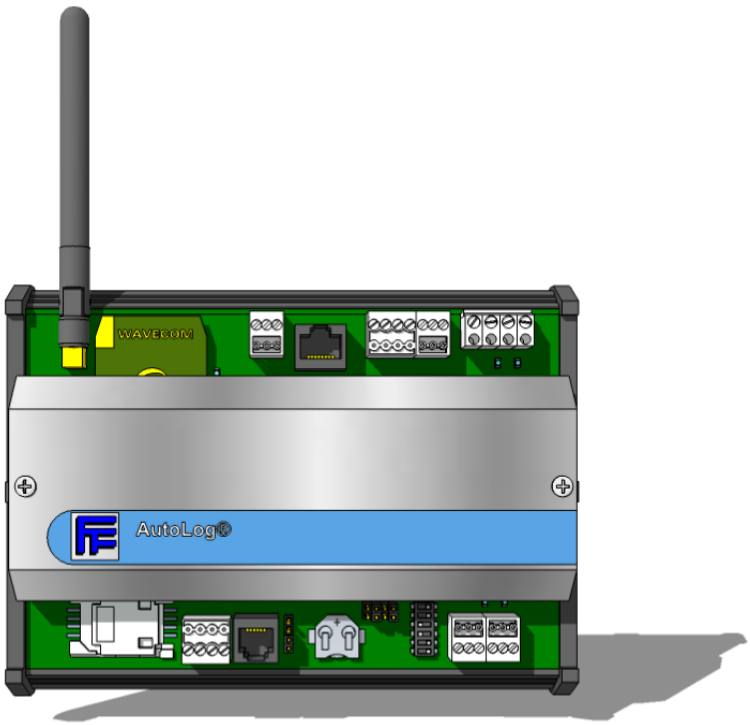
GSM-4 : ORDER CODES		
Code	**Designation**	**Description**
900684	GSM-4 (case)	IP20 metal cover, DIN rail mounting, stubby antenna
900693	GSM-4 Unit	ABS plastic box, IP65, power antenna
980693	GSM-4 Unit + HMI	as above + HMI mounted on the cover, IP65
N/A	GSM-4 custom unit	Look customized GSM units!

- Power is only included to GSM UNITS! Please specify the power supply type!
- Stubby antenna is default, other antenna types can be ordered as options (Stubby antenna price will be deducted)!

14.5.4 GSM-4 circuit board connectors



14.5.5 AutoLog GSM-4 picture



14.6 AutoLog GSM-8

14.6.1 GSM-8 Main Features:

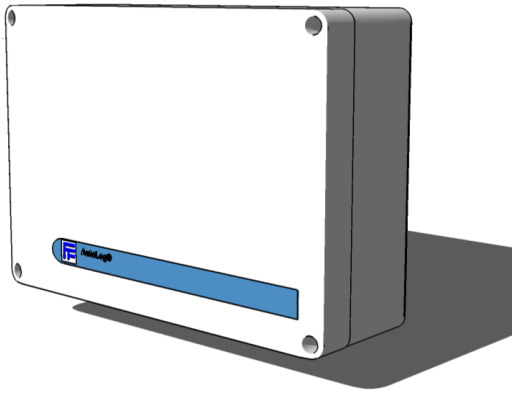
- **0-2 Analog Inputs(*), 4 Digital Inputs, 2 Digital Outputs, 2 Relay Outputs**
- Integrated GSM modem (for SMS, GPRS and call)
- 2 Serial ports
  - Ser 1 is reserved for GSM modem / programming
  - Ser 2 can be used as Modbus RTU M/S RS-232
  - Optionally Ser 2 can be used as RS-485 or Modbus TCP (Ethernet). Optional Wireless sensor master module can also be connected to this port.
- I2C interface for HMI (character based display & keypad)
- Stubby antenna (optionally stubby antenna can be changed to external antenna)
- SMA Antenna connector
- Operating power 10-30VDC or 12-24VAC
- Power output 24VDC
- Accumulator connection. Charge connection with deep discharge protection.
- 230VAC power supply is included only to GSM units!
- Real time clock & calendar
- 512 programming lines, 32 build-in PID controllers, time-tables
- data logging to FTP files (e.g. 1x 43000 or 4x 21000 or 8x 13000 Time-stamped records)
- incoming phone number identification, 254 phone numbers
- Operating temperature -30…+65°C, Storage temperature -40…+80°C
- Look more!

14.6.2 GSM-8 I/O:

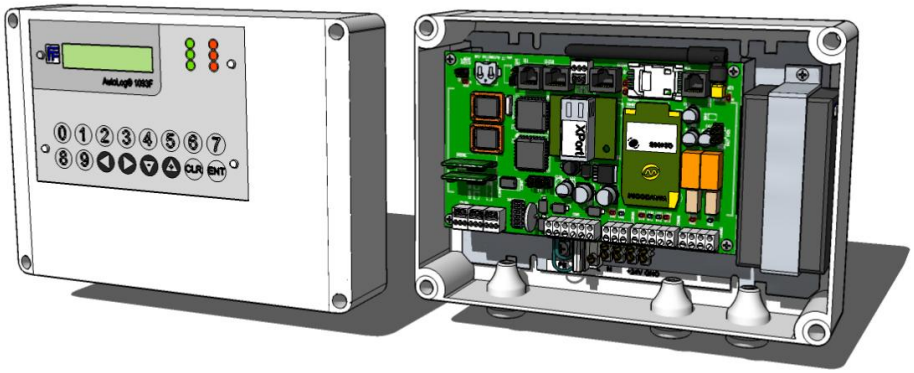
GSM-8 I/O	
Analog Inputs	0-2* analog inputs, 12 bit, specify analog input modules!
Digital Inputs	4 digital inputs, 24VDC, max 8mA, opto-isolated, PNP
Digital Output	2 digital output, 12-24VDC, 2A, NPN
Relay Outputs	2 relay outputs, 5A /250VAC 1A/30VDC

14.6.3 GSM-8 Order codes

900951 GSM-8 (case)



900960 GSM-8 Unit

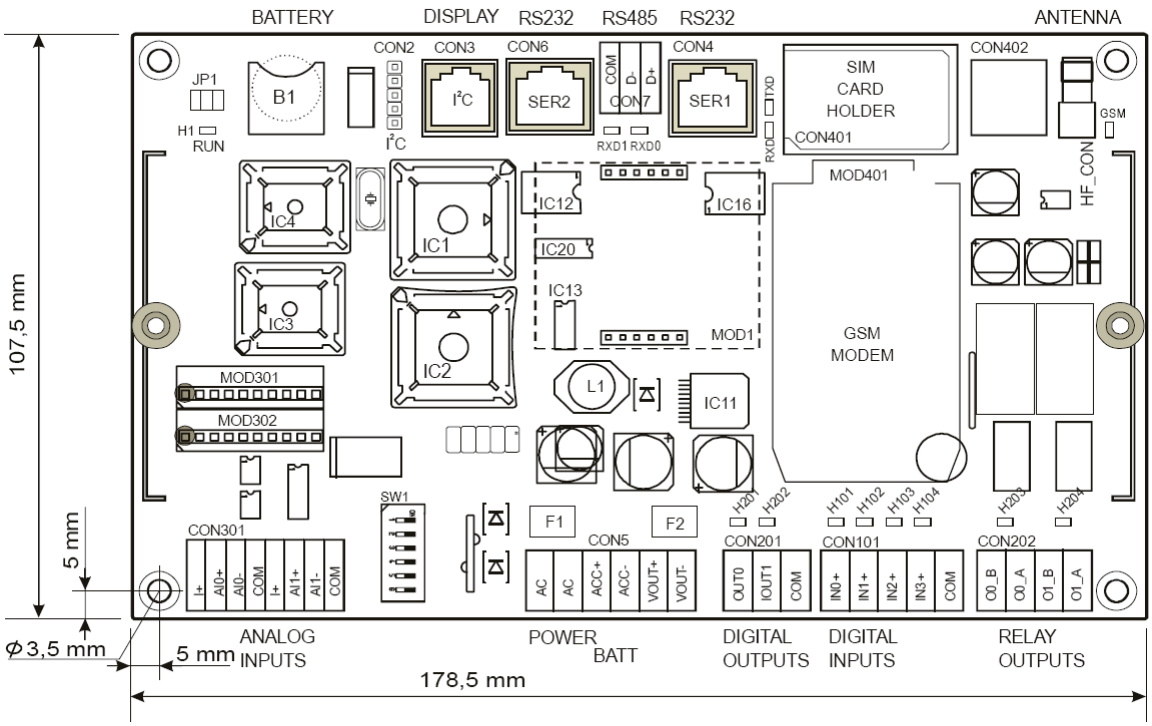


980960 GSM-8 Unit + HMI (Plug in Module is optional)

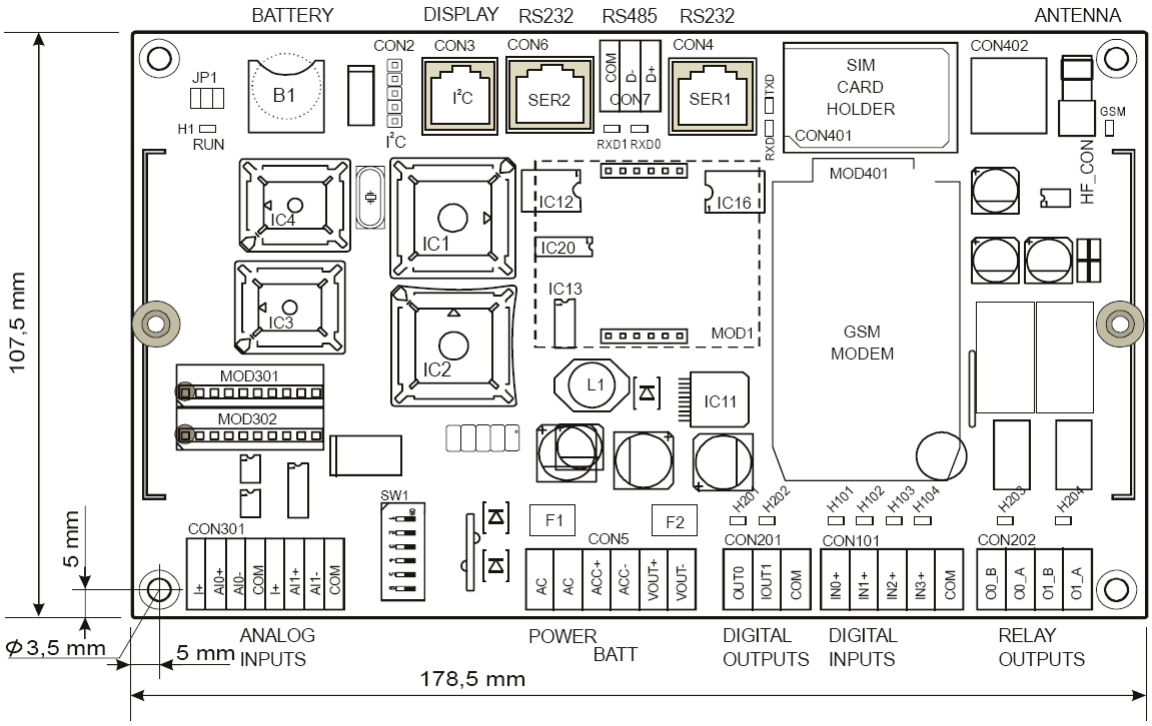
GSM-8 : ORDER CODES		
Code	**Designation**	**Description**
900951	GSM-8 (case)	IP20 metal cover, DIN rail mounting, stubby antenna
900960	GSM-8 unit	ABS plastic box, IP65, power supply
980960	GSM-8 unit + HMI	as above + HMI mounted on the cover, IP65
N/A	GSM-8 custom unit	Look customized GSM units!

- (*) = Analog input modules are NOT included. So by default all GSM products have no analog inputs. Analog input modules are small modules which are attached to GSM board with plug in connector. The reason for this is that user can select just the right analog input module combination that is needed. E.g. (0-2*) means that GSM board supports maximum of 2 analog input modules. Look the list of available analog inputs modules in the “Accessories”.
- Power is only included to GSM UNITS! Please specify the power supply type!
- Stubby antenna is default, other antenna types can be ordered as options (Stubby antenna price will be deducted)!

14.6.4 GSM-8 circuit board connectors



ID	Description
CON3	I²C port for display/keypad unit
CON4	RS-232 Serial connection, SER1
CON5	Power supply input, battery connection and Power output
CON6	RS-232 Serial connection, SER2
CON7	RS-485 Serial connection, SER2, RS-485 needs optional RS-232 to RS-485 Plug-in Module (PiM)
CON101	Digital inputs 0-3
CON201	Digital outputs 0-1
CON202	Relay outputs 0-1
CON301	Analog inputs 0-1
HF_CON	GSM modem antenna connector
MOD1	Mounting connectors for AutoLog Plug-in Module (PiM)
MOD30x	Analog input modules 0-1
MOD401	SIM card holder. Normally customer purchases SIM card locally.
IC1	CPU processor
IC2	Address coder
IC3	Flash, system program
IC4	Flash, application program
B1	Backup lithium battery for retaining values in RAM (EEPROM) memory. Type: CR2032. 3 Volts.
SW1	DIP switches 1-6
JP1	Jumpers
JP1:1-2	Power fail info (ON= 9V, OFF=17V), When power fail is active RO36=1
JP1:3-4	Watch dog (ON= active, OFF= disabled)
JP1:5-6	Backup battery (ON=enabled, OFF=disabled)



14.6.5 AutoLog GSM-8 picture

AutoLog GSM-16

14.6.6 GSM-16 Main Features:

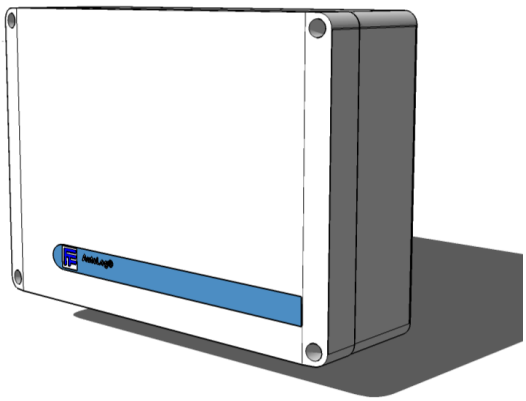
- **0-8 Analog Inputs(*), 4 Digital Inputs, 4 Relay Output**
- Integrated GSM modem (for SMS, GPRS and call)
- 2 Serial ports
  - Ser 1 is reserved for GSM modem / programming
  - Ser 2 can be used as Modbus RTU M/S RS-232
  - Optionally Ser 2 can be used as RS-485 or Modbus TCP (Ethernet). Optional Wireless sensor master module can also be connected to this port.
- I2C interface for HMI (character based display & keypad)
- Stubby antenna (optionally stubby antenna can be changed to external antenna)
- SMA Antenna connector
- Operating power 10-30VDC or 12-24VAC
- Power output 24VDC
- Accumulator connection. Charge connection with deep discharge protection.
- 230VAC power supply is included only to GSM units!
- Real time clock & calendar
- 512 programming lines, 32 build-in PID controllers, time-tables
- data logging to FTP files (e.g. 1x 43000 or 4x 21000 or 8x 13000 Time-stamped records)
- incoming phone number identification, 254 phone numbers
- Operating temperature -30…+65°C, Storage temperature -40…+80°C
- Look more!

14.6.7 GSM-16 I/O:

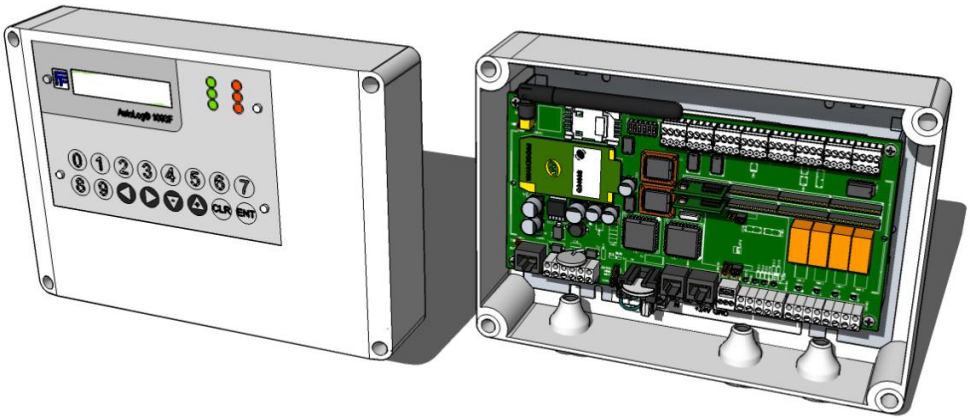
GSM-16 I/O	
Analog Inputs	0-8* analog inputs, 12 bit, specify analog input modules!
Digital Inputs	4 digital inputs, 12-24VDC, max 8mA, opto-isolated, PNP
Relay Outputs	4 relay outputs, 5A /250VAC 1A/30VDC

14.6.8 GSM-16 Order codes:

900501 GSM-16 (case)



900480 GSM-16 Unit

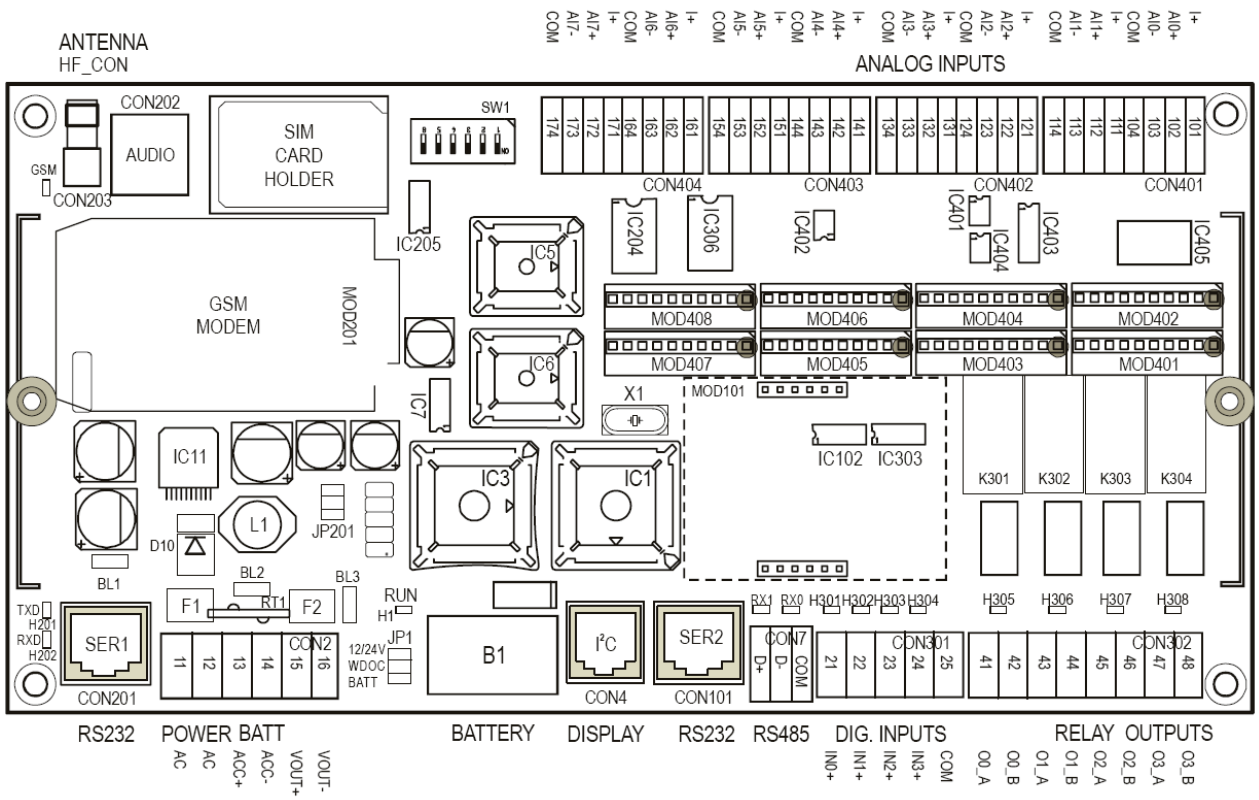


980480 GSM-16 Unit + HMI

GSM-16: ORDER CODES		
Code	**Designation**	**Description**
900501	GSM-16 (case)	IP20 metal cover, DIN rail mounting, stubby antenna
900480	GSM-16 unit	ABS plastic box, IP65, power supply
980480	GSM-16 unit + HMI	as above + HMI mounted on the cover, IP65
N/A	GSM-16 custom unit	Look customized GSM units!

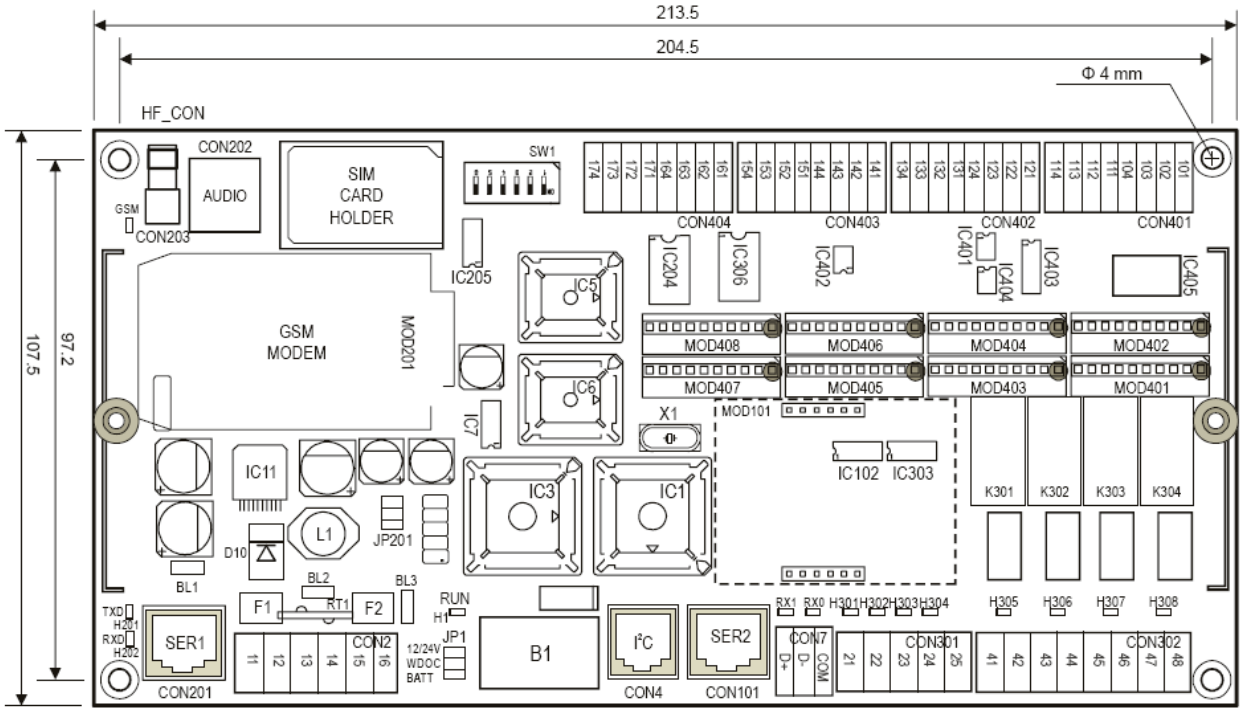
- (*) = Analog input modules are NOT included. So by default all GSM products have no analog inputs. Analog input modules are small modules which are attached to GSM board with plug in connector. The reason for this is that user can select just the right analog input module combination that is needed. E.g. (0-8*) means that GSM board supports maximum of 8 analog input modules. Look the list of available analog inputs modules in the “Accessories”.
- Power is only included to GSM UNITS! Please specify the power supply type!
- Stubby antenna is default, other antenna types can be ordered as options (Stubby antenna price will be deducted)!

14.6.9 GSM-16 circuit board connectors

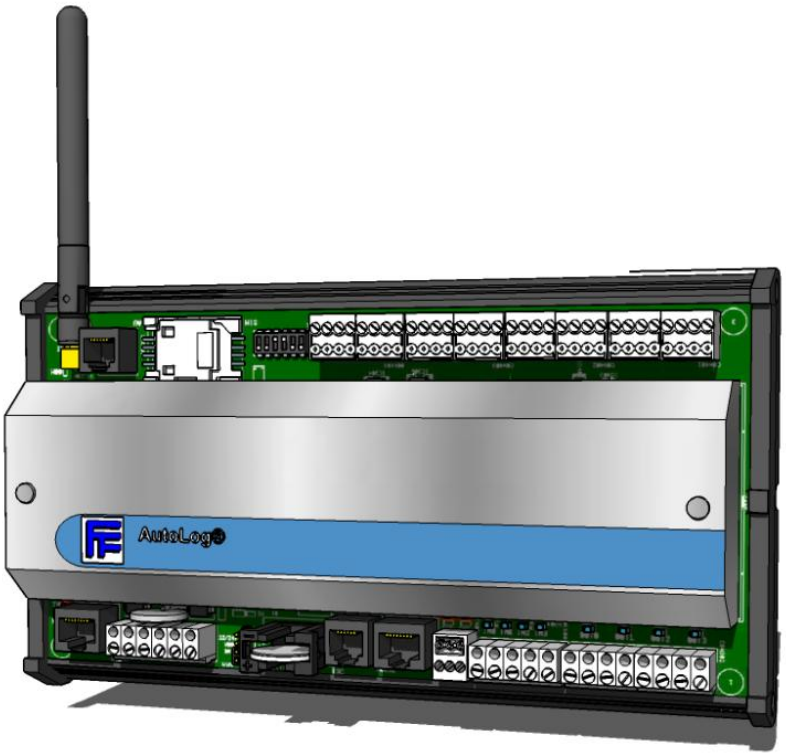


ID	Description
CON2	Power supply input, battery connection and Power output
CON4	I²C port for display/keypad unit
CON7	RS-485 Serial connection, SER2, RS-485 needs optional RS-232 to RS-485 Plug-in Module (PiM)
CON101	RS-232 Serial connection, SER2
CON201	RS-232 Serial connection, SER1
CON202	Audio connector
CON203	GSM modem antenna connector
CON301	Digital Inputs 0-3
CON302	Relay outputs 0-3
CON40x	Analog inputs 0-7
MOD101	Mounting connectors for AutoLog Plug-in Module (PiM)
MOD201	GSM modem connector
MOD40x	Analog input modules 0-1
SIM	SIM card holder
IC1	CPU processor
IC3	Address coder
IC5	Flash, application program
IC6	Flash, system program
B1	Backup lithium battery for retaining values in RAM (EEPROM) memory. Type: CR2032. 3 Volts.
SW1	DIP switches 1-6
JP1	Jumpers
JP1:1-2	Power fail info (ON= 9V, OFF=17V), When power fail is active RO36=1
JP1:3-4	Watch dog (ON= active, OFF= disabled)
JP1:5-6	Backup battery (ON=enabled, OFF=disabled)

14.6.10 AutoLog GSM-16 Dimensions



14.6.11 AutoLog GSM-16 picture



14.7 AutoLog GSM-20

14.7.1 GSM-20 Main Features:

- **0-8 Analog Inputs (*), 8 Digital Inputs, 8 Digital Outputs, 2 Analog outputs on-board (+supports I/O expansion cards)**
- **I2C interface for I/O expansion cards**
- **Needs external GSM modem (for SMS and GPRS)**
- 2 Serial ports
  - Ser 1 is reserved for GSM modem / programming (just switch the cable, DIP4 is not used)
  - Ser 2 can be used as Modbus RTU M/S RS-232
  - Optionally Ser 2 can be used as RS-485 or Modbus TCP (Ethernet). Optional Wireless sensor master module can also be connected to this port.
- I2C interface for HMI (character based display & keypad)
- Stubby antenna (optionally stubby antenna can be changed to external antenna)
- SMA Antenna connector
- Operating power 10-30VDC or 12-24VAC
- Power output 24VDC
- Accumulator connection. Charge connection with deep discharge protection.
- 230VAC power supply is included only to GSM units!
- Real time clock & calendar
- 512 programming lines, 32 build-in PID controllers, time-tables
- data logging to FTP files (e.g. 1x 43000 or 4x 21000 or 8x 13000 Time-stamped records)
- incoming phone number identification, 254 phone numbers
- Operating temperature -30…+65°C, Storage temperature -40…+80°C
- Look more!

14.7.2 GSM-20 I/O:

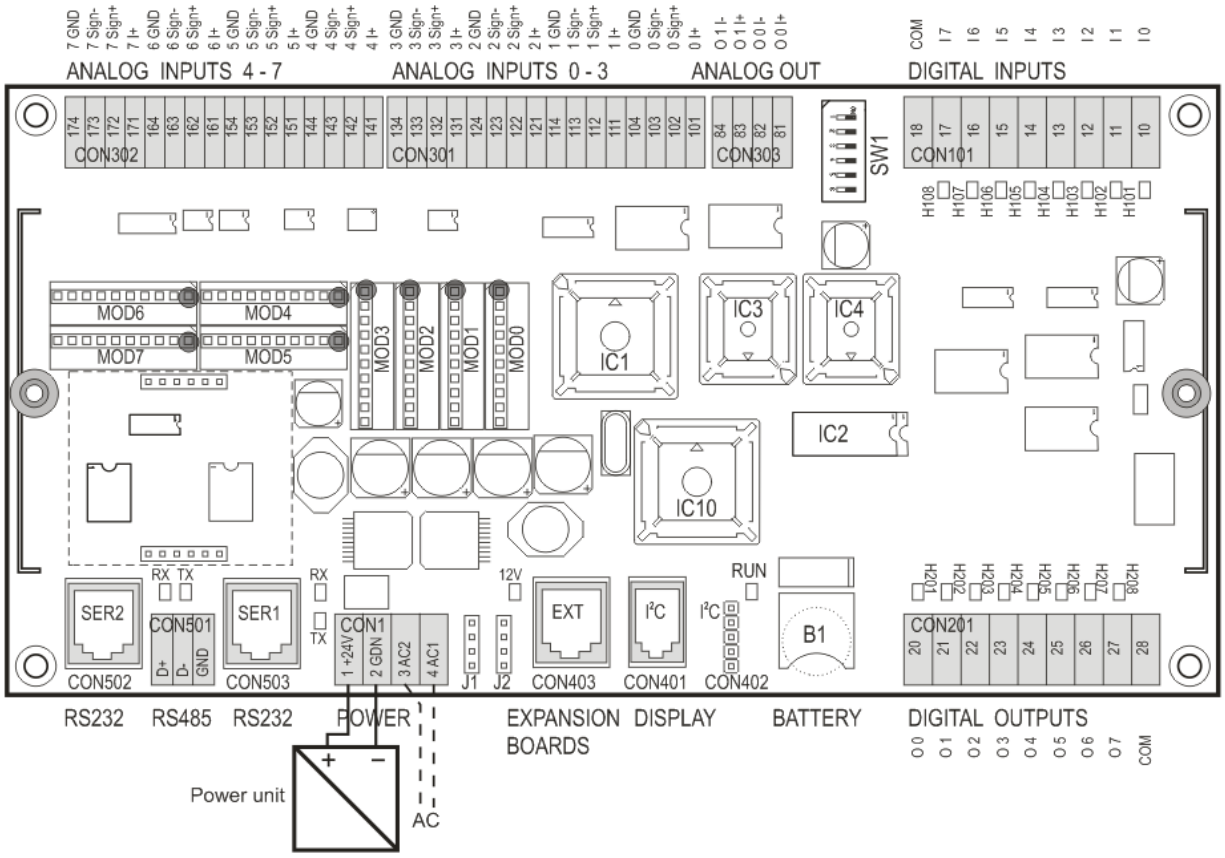
GSM-20 I/O (ON MAIN BOARD)	
Analog Inputs	0-8* analog inputs, 12 bit, specify analog input modules!
Analog Outputs	2 analog outputs, 12 bit, 0-5VDC / 0-10VDC
Digital Inputs	8 digital inputs, 24VDC, max 8mA, opto-isolated, PNP
Digital Outputs	8 digital outputs, 24VDC, max. 2A, max. 4A per group, NPN

14.7.3 GSM-20 Order codes:

GSM-20: ORDER CODES (MAIN BOARD)		
Code	**Designation**	**Description**
900755	GSM-20 (case)	IP20 metal cover, DIN rail mounting (No GSM modem, No antenna)
N/A	GSM-20 Unit	Look customized GSM Units!
N/A	GSM-20 unit + HMI	as above + HMI mounted on the cover, IP65

- (*) = Analog input modules are NOT included. So by default all GSM products have no analog inputs. Analog input modules are small modules which are attached to GSM board with plug in connector. The reason for this is that user can select just the right analog input module combination that is needed. E.g. (0-8*) means that GSM board supports maximum of 8 analog input modules. Look the list of available analog inputs modules in the "Accessories".
- Power is only included to GSM UNITS! Please specify the power supply type!
- Antenna is only included to GSM-20 UNIT! Stubby antenna is default, other antenna types can be ordered as options (Stubby antenna price will be deducted)!
- Note! GSM-20 (case) needs external GSM modem!

14.7.4 GSM-20 circuit board connectors



ID	Description
CON1	Power supply (Power input)
CON401	I²C port for display/keypad unit (RJ-11)
CON502	RS-232 Serial connection, SER2
CON402	I²C port for display/keypad unit
CON503	RS-232 Serial connection, SER1
CON501	RS-485 Serial connection, SER2, RS-485 needs optional RS-232 to RS-485 Plug-in Module (PiM)
CON101	Digital inputs
CON201	Digital outputs
CON202	Relay outputs
CON301	Analog inputs
HF_CON	GSM modem antenna connector
MOD1	Mounting connectors for AutoLog Plug-in Module (PiM)
MOD30x	Analog input modules 0-1
MOD401	SIM card holder
IC1	CPU processor
IC2	Address coder
IC3	Flash, system program
IC4	Flash, application program
B1	Backup lithium battery for retaining values in RAM (EEPROM) memory.
SW1	DIP switches 1-6
JP1	Jumpers
JP1:1-2	Power fail limit (ON= 9V, OFF=17V), if power input is less than limit stops PLC application program
JP1:3-4	Watch dog (ON= active, OFF= disabled)
JP2:3-4	12VDC Power to I2C EXT (ON=enabled, OFF=disabled)

In the table below are possible definitions of the input and output points of GSM20 PLC's CPU board.

Conn. num.	Digital output	Digital input
10		I 00
11		I 01
12		I 02
13		I 03
14		I 04
15		I 05
16		I 06
17		I 07
18		COM
20	O 00	
21	O 01	
22	O 02	
23	O 03	
24	O 04	
25	O 05	
26	O 06	
27	O 07	
28	COM	

Analog OUTPUTS

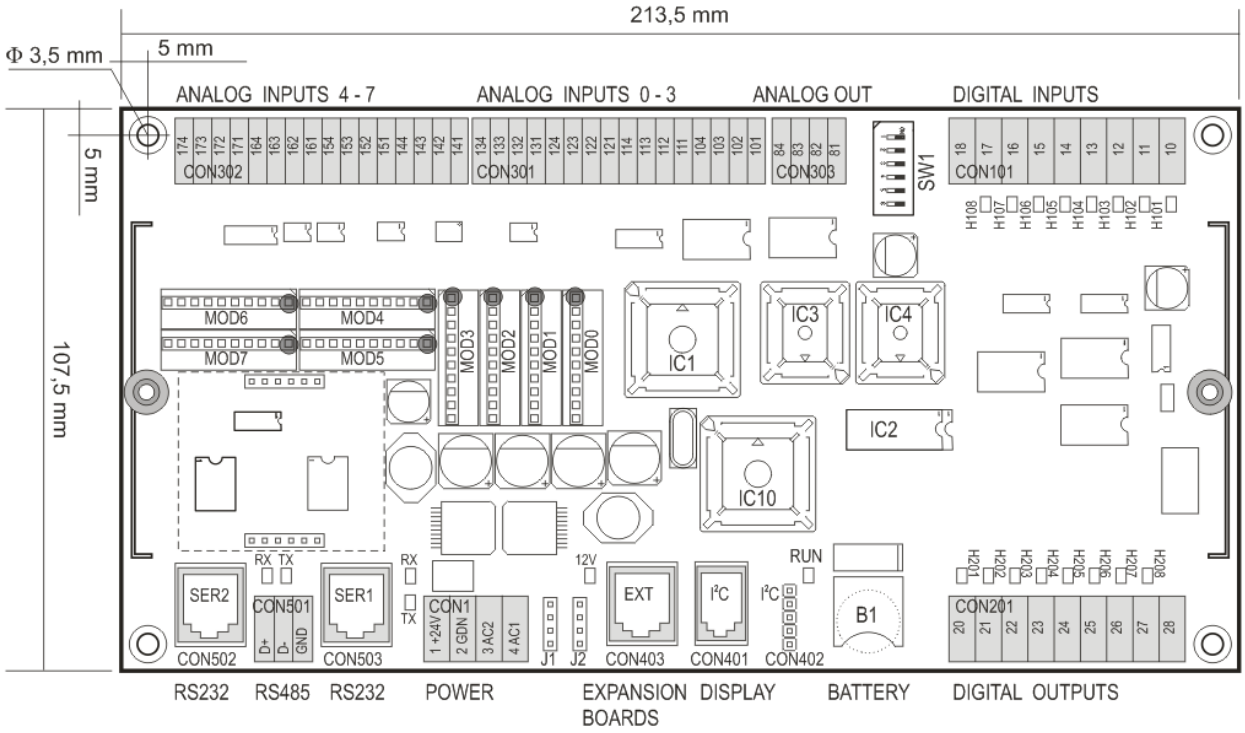
Conn num.	Analog output
81	AO 00
82	GND
83	AO 01
84	GND

Analog inputs / digital inputs

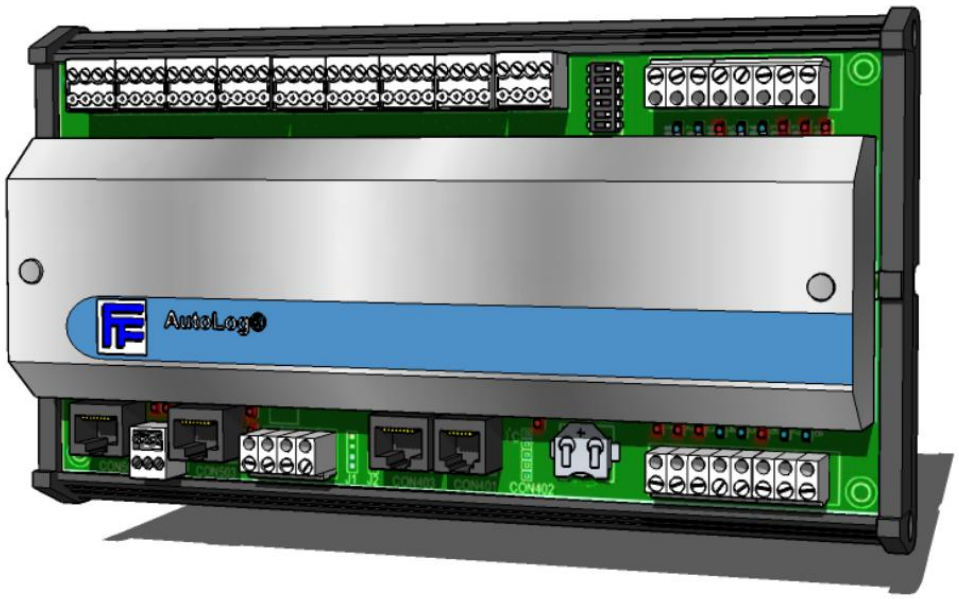
Conn. num.	Defining device	Analog input	Digital input
101	Module	I +	I 08
102	Module	AI 00 +	I 09
103	Module	AI 00 -	I 10
104		GND	GND
111	Module	I +	I 11
112	Module	AI 01 +	I 12
113	Module	AI 01 -	I 13
114		GND	GND
121	Module	I +	I 14
122	Module	AI 02 +	I 15
123	Module	AI 02 -	I 16
124		GND	GND
131	Module	I +	I 17
132	Module	AI 03 +	I 18
133	Module	AI 03 -	I 19
134		GND	GND
141	Module	I +	I 20
142	Module	AI 04 +	I 21
143	Module	AI04 -	I 22
144		GND	GND
151	Module	I +	I 23
152	Module	AI 05 +	I 24
153	Module	AI 05 -	I 25
154		GND	GND
161	Module	I +	I 26
162	Module	AI 06 +	I 27
163	Module	AI 06 -	I 28
164		GND	GND
171	Module	I +	I 29
172	Module	AI 07 +	I 30
173	Module	AI 07 -	I 31
174		GND	GND

For additional digital inputs I08-> , optional AI to DI convertor module(s) are needed.

14.7.5 AutoLog GSM-20 Dimensions



14.7.6 AutoLog GSM-20 picture



14.7.7 GSM-20 Expansion I/O cards:

Note! Only GSM-20 supports I/O expansion cards.

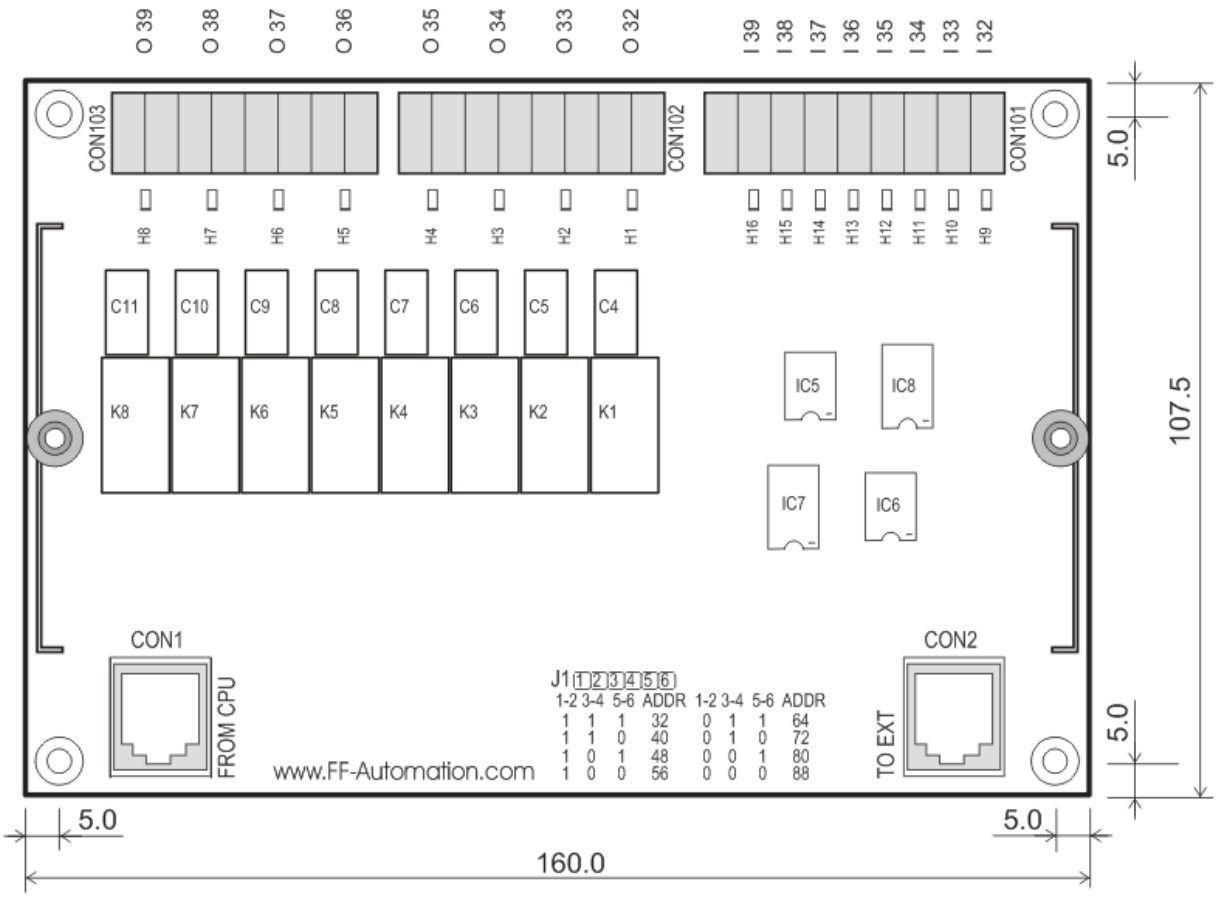
GSM-20 EXPANSION I/O (MAX. NUMBER OF I/O)	
Analog Inputs	Max. 8 AI on-board, Max. 8 x EXA 8/4 = 64 AI Total AI = 72
Analog Outputs	2 AO on-board, Max 8 x EXA8/4 = 32 AO Total AO = 34
Digital Inputs	8 DI on-board, Max. 4 x DI16 = 64 DI Total DI = 72
Digital Outputs	8 DO on-board, Max. 2 x DO32 = 64 DO Total DO = 72

GSM-20: ORDER CODES (I/O EXPANSION CARDS)		
Code	**Designation**	**Description**
900775	AL20 DI16	16 x Digital inputs, 24VDC, max. 8mA, potential free contact, PNP
900772	AL20 DO32	32 x Digital outputs, 24VDC, max. 1A, NPN
900765	AL20 RIO8	8 x Digital inputs, 24VDC, max. 8mA, potential free contact, PNP 8 x Relay outputs 24-230VAC, max. 3A
900770	AL20 RO16	16 x Relay outputs, 24-230VAC, max. 3A
900837	AL20 EXA 8/4*	8 x Analog inputs (plug in analog input modules are not included) 4 x Analog outputs

I/O Expansion board cable included (code 900750)

- (*) = Analog input modules are NOT included to EXA 8/4. So by default EXA 8/4 have no analog inputs. Analog input modules are small modules which are attached to EXA 8/4 board with plug in connector. The reason for this is that user can select just the right analog input module combination that is needed. Look the list of available analog inputs modules in the “Accessories”.
- I2C cable is included to I/O expansion card price

14.7.8 AutoLog GSM-20: RIO8 I/O Expansion card



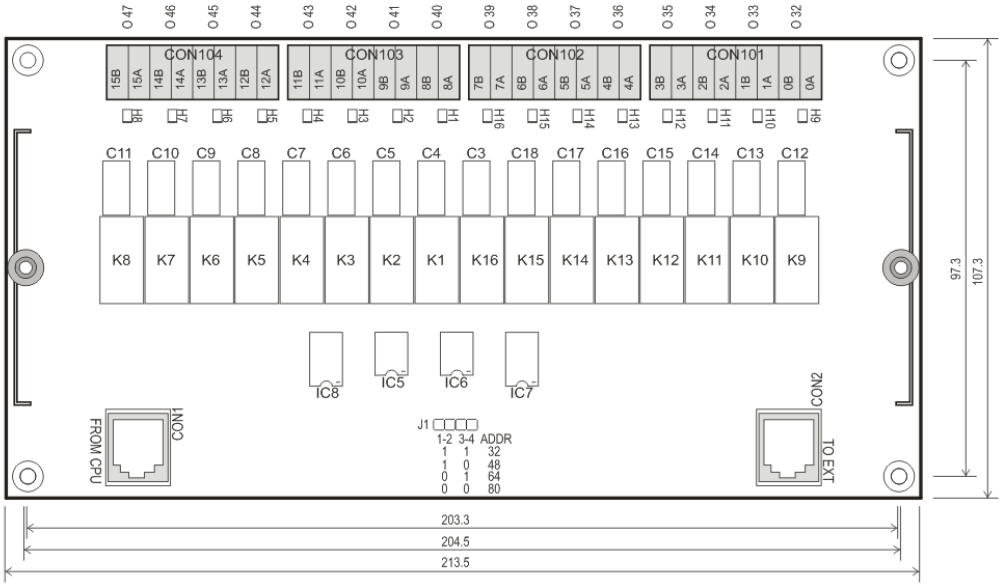
Digital input addresses in expansion card RIO8, Only in GSM-20

RIO8 (Card address 1. J12=On, J34=On, J56= On) DI32-39, (connected card status RO2.0)
RIO8 (Card address 2. J12=Off, J34=On, J56= On) DI40-47, (connected card status RO2.1)
RIO8 (Card address 3. J12=On, J34=Off, J56= On) DI48-55, (connected card status RO2.2)
RIO8 (Card address 4. J12=Off, J34=Off, J56= On) DI56-63, (connected card status RO2.3)
RIO8 (Card address 5. J12=On, J34=On, J56= Off) DI64-71, (connected card status RO2.4)
RIO8 (Card address 6. J12=Off, J34=On, J56= Off) DI72-79, (connected card status RO2.5)
RIO8 (Card address 7. J12=On, J34=Off, J56= Off) DI80-87, (connected card status RO2.6)
RIO8 (Card address 8. J12=Off, J34=Off, J56= Off) DI88-95, (connected card status RO2.7)

Digital output addresses in expansion card RIO8, Only in GSM-20

Digital output addresses in expansion card RIO8, Only in GSM-20!
RIO8 (Card address 1. J12=On, J34=On, J56= On) DO32-39, (connected card status RO3.0)
RIO8 (Card address 2. J12=Off, J34=On, J56= On) DO40-47, (connected card status RO3.1)
RIO8 (Card address 3. J12=On, J34=Off, J56= On) DO48-55, (connected card status RO3.2)
RIO8 (Card address 4. J12=Off, J34=Off, J56= On) DO56-63, (connected card status RO3.3)
RIO8 (Card address 5. J12=On, J34=On, J56= Off) DO64-71, (connected card status RO3.4)
RIO8 (Card address 6. J12=Off, J34=On, J56= Off) DO72-79, (connected card status RO3.5)
RIO8 (Card address 7. J12=On, J34=Off, J56= Off) DO80-87, (connected card status RO3.6)
RIO8 (Card address 8. J12=Off, J34=Off, J56= Off) DO88-95, (connected card status RO3.7)

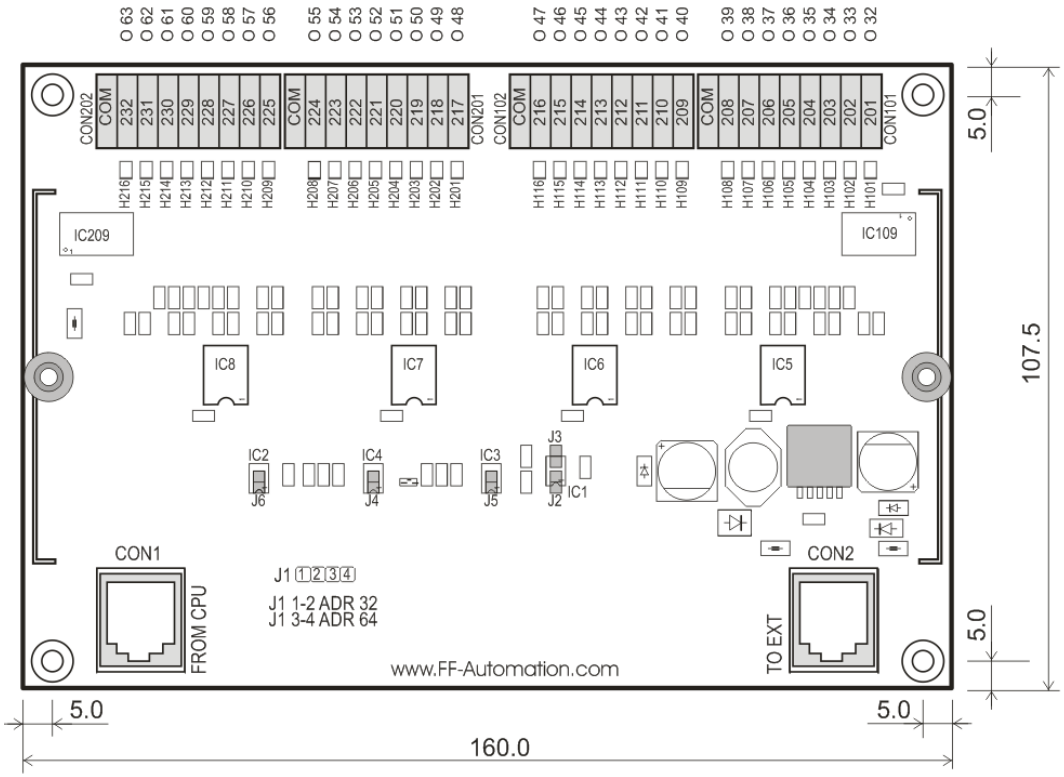
14.7.9 AutoLog GSM-20: RO16 I/O Expansion card



Digital output addresses in expansion card RO16, Only in GSM-20!

RO16 (Card address 1. J12=On, J34=On) DO32-47, (connected card status RO3.0 & RO3.1)
RO16 (Card address 2. J12=Off, J34=On) DO48-63, (connected card status RO3.2 & RO3.3)
RO16 (Card address 3. J12=On, J34=Off) DO64-79, (connected card status RO3.4 & RO3.5)
RO16 (Card address 4. J12=Off, J34=Off) DO80-95, (connected card status RO3.6 & RO3.7)

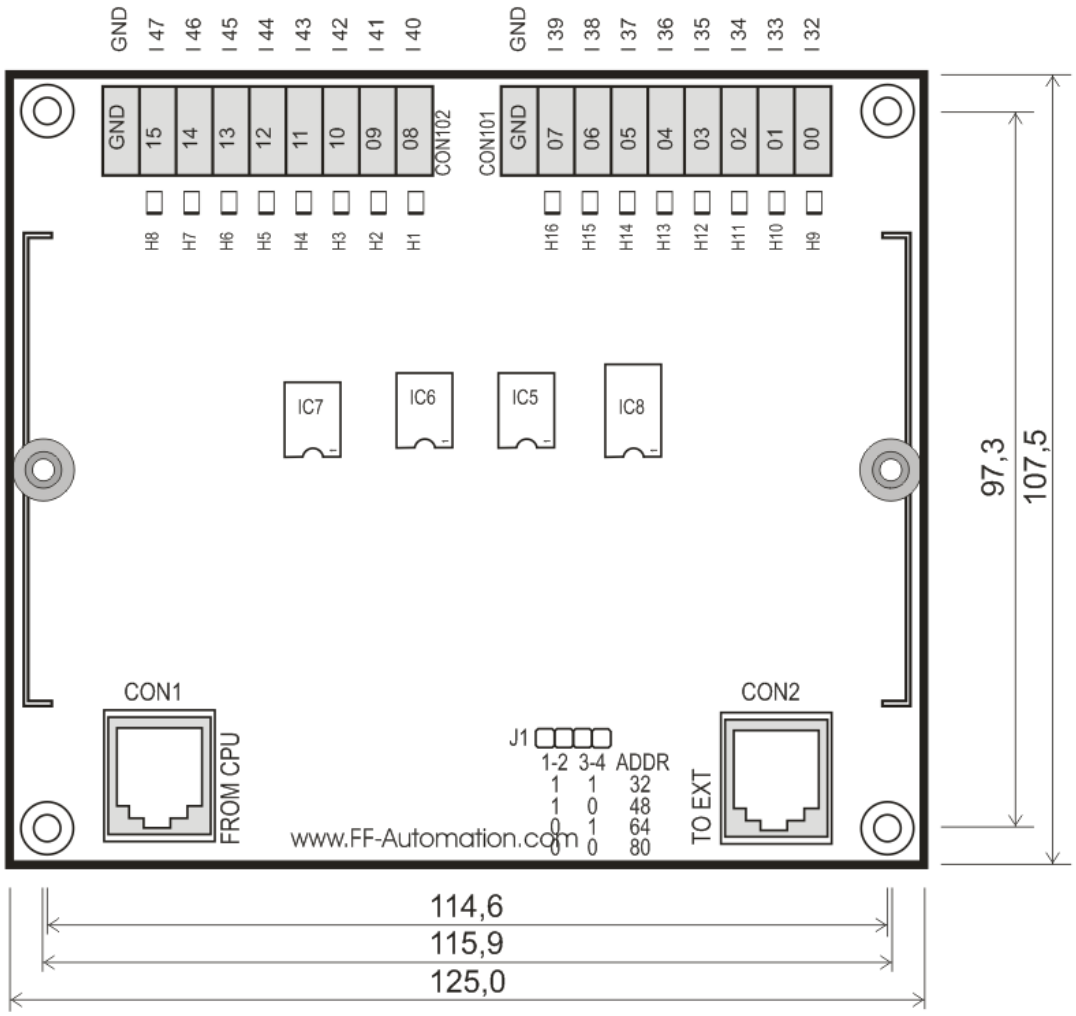
14.7.10 AutoLog GSM-20: DO32 I/O Expansion card



Digital output addresses in expansion card DO32, Only in GSM-20!

DO32 (Card address 1. J12=On, J34=Off) DO32-63, (connected card status RO3.0-RO3.3)
DO32 (Card address 2. J12=Off, J34=On) DO64-95, (connected card status RO3.4-RO3.7)

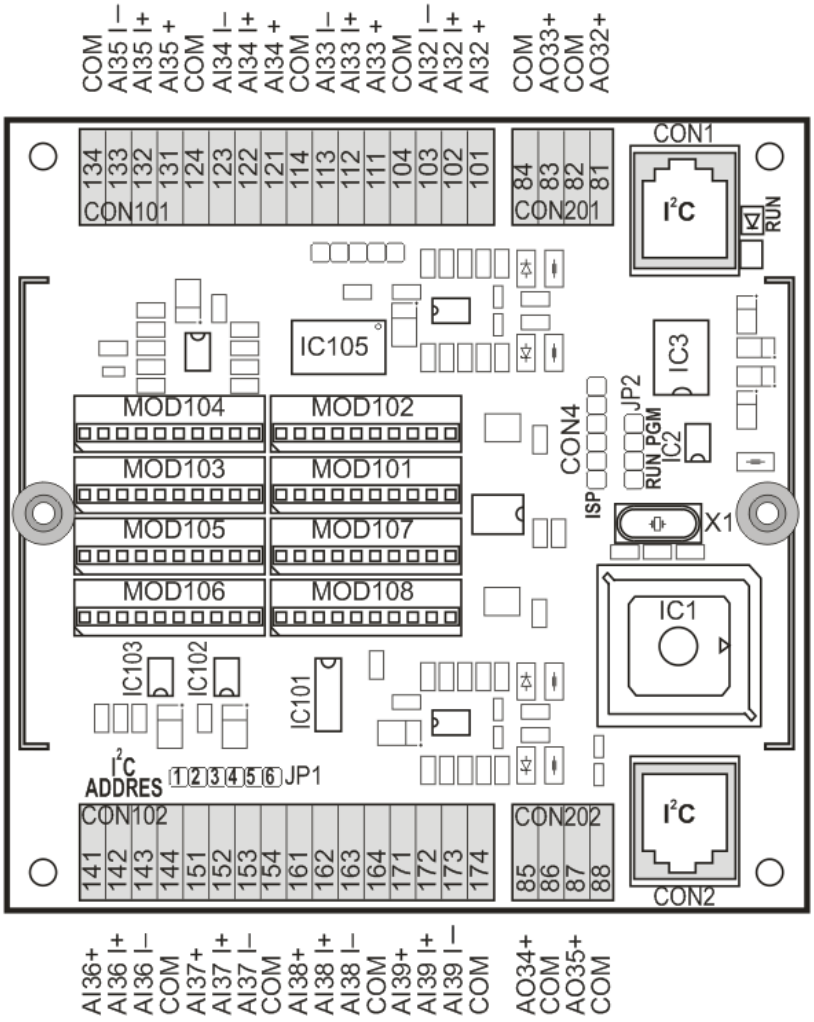
14.7.11 AutoLog GSM-20: DI16 I/O Expansion card



Digital inputs addresses in expansion card DI16, Only in GSM-20!

DI16 (Card address 1. J12=On, J34=On) DI32-47, (connected card status RO2.0 & RO2.1)
DI16 (Card address 2. J12=Off, J34=On) DI48-63, (connected card status RO2.2 & RO2.3)
DI16 (Card address 3. J12=On, J34=Off) DI64-79, (connected card status RO2.4 & RO2.5)
DI16 (Card address 4. J12=Off, J34=Off) DI80-95, (connected card status RO2.6 & RO2.7)

14.7.12 AutoLog GSM-20: EXA 8/4 I/O Expansion card



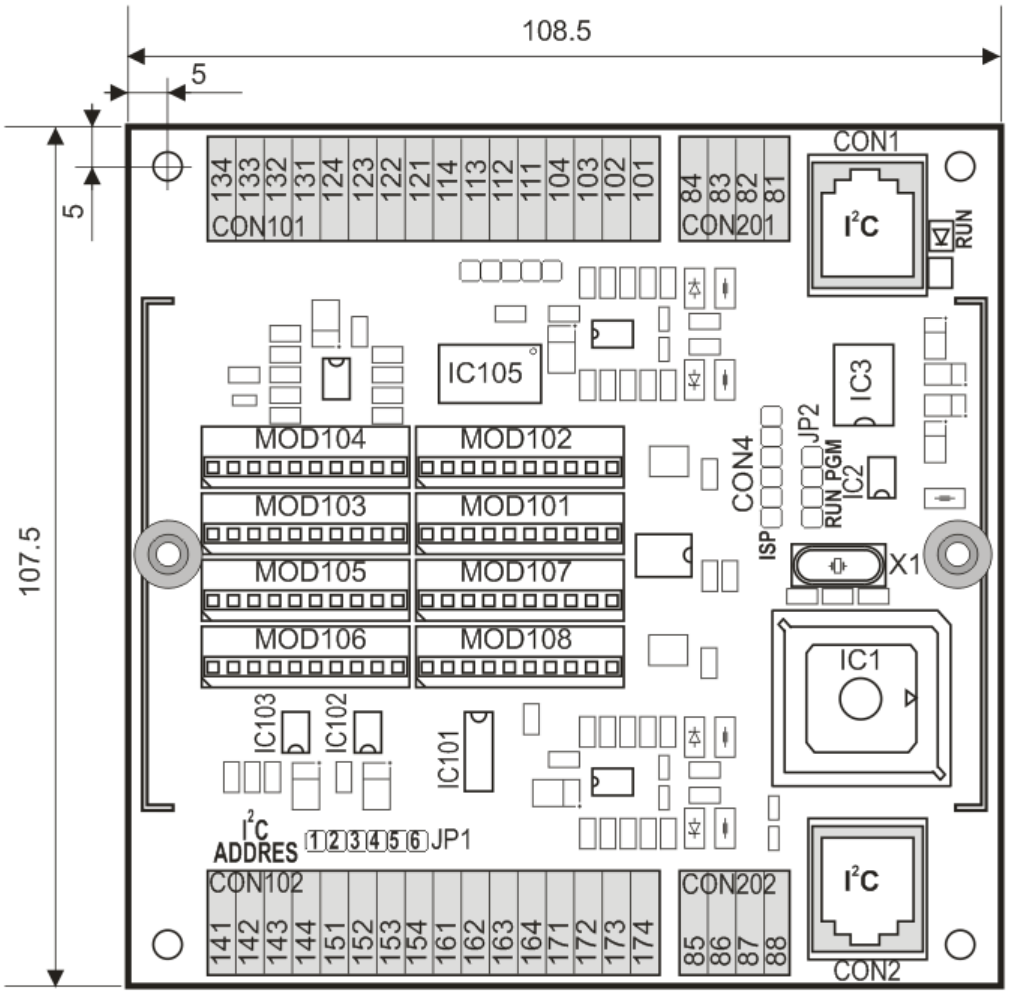
Analog input addresses in expansion card EXA8/4, Only in GSM-20!

EXA8/4 (Card address 1. J12=On, J34=On, J56= On) AI32-39, (connected card status RO4.0)
EXA8/4 (Card address 2. J12=Off, J34=On, J56= On) AI40-47, (connected card status RO4.1)
EXA8/4 (Card address 3. J12=On, J34=Off, J56= On) AI48-55, (connected card status RO4.2)
EXA8/4 (Card address 4. J12=Off, J34=Off, J56= On) AI56-63, (connected card status RO4.3)
EXA8/4 (Card address 5. J12=On, J34=On, J56= Off) AI64-71, (connected card status RO4.4)
EXA8/4 (Card address 6. J12=Off, J34=On, J56= Off) AI72-79, (connected card status RO4.5)
EXA8/4 (Card address 7. J12=On, J34=Off, J56= Off) AI80-87, (connected card status RO4.6)
EXA8/4 (Card address 8. J12=Off, J34=Off, J56= Off) AI88-95, (connected card status RO4.7)

Analog output addr. in expansion card EXA8/4, Only in GSM-20!

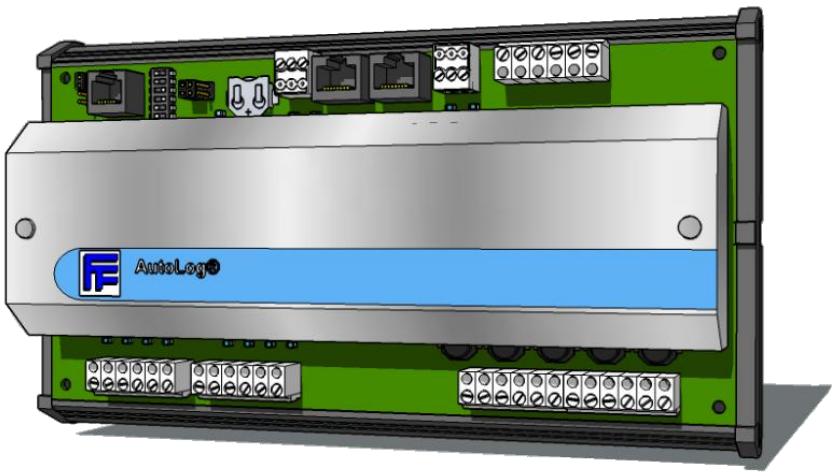
EXA8/4 (Card address 1. J12=On, J34=On, J56= On) AO32-35, (connected card status RO4.0)
EXA8/4 (Card address 2. J12=Off, J34=On, J56= On) AO40-43, (connected card status RO4.1)
EXA8/4 (Card address 3. J12=On, J34=Off, J56= On) AO48-51, (connected card status RO4.2)
EXA8/4 (Card address 4. J12=Off, J34=Off, J56= On) AO56-59, (connected card status RO4.3)
EXA8/4 (Card address 5. J12=On, J34=On, J56= Off) AO64-67, (connected card status RO4.4)
EXA8/4 (Card address 6. J12=Off, J34=On, J56= Off) AO72-75, (connected card status RO4.5)
EXA8/4 (Card address 7. J12=On, J34=Off, J56= Off) AO80-83, (connected card status RO4.6)
EXA8/4 (Card address 8. J12=Off, J34=Off, J56= Off) AO88-91, (connected card status RO4.7)

AutoLog GSM-20: EXA 8/4 I/O Expansion card's Dimensions



Depth 35 mm

14.8 AutoLog GSM-10SP



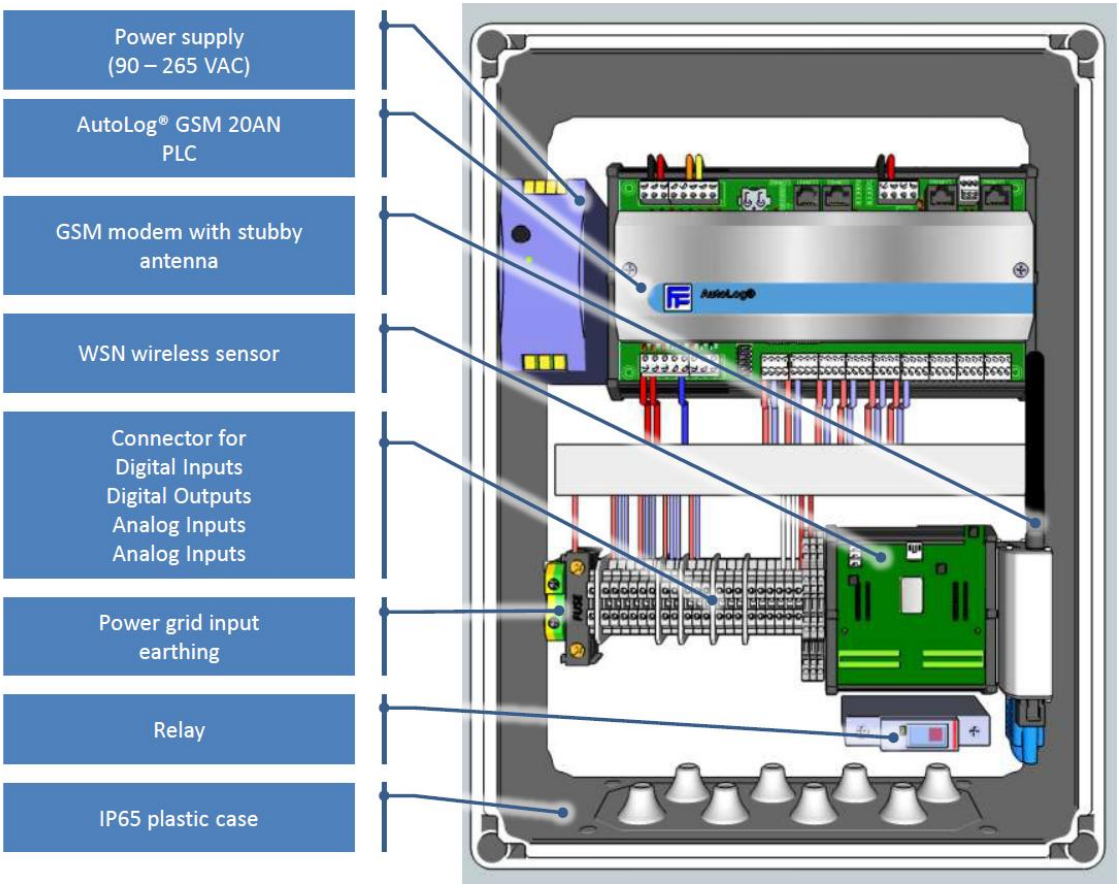
Especially designed for Cathodic Protection applications, but can be used in other applications also!

GSM-10SP I/O	
Analog Inputs	5 analog inputs, AutoLog 10 SP PLC doesn't need analog input modules. Analog inputs can be tuned in PLC software. 16-bit resolution. Input ranges can be selected point-wise using software. Possible input ranges are: 0-800mVDC, 0-3.2VDC, 0-50VDC and 0-100VDC. Surge protected and opto-isolated inputs..
Digital Inputs	4 digital inputs, 12-24VDC, max 8mA, opto-isolated, PNP
Digital Outputs	4 digital outputs,12/24VDC / max. 1A / NPN / opto-isolated

GSM-10SP: ORDER CODES		
Code	**Designation**	**Description**
900140	GSM-10SP (case)	IP20 metal cover, DIN rail mounting, (Doesn't include GSM modem)

- Power is not included!
- Stubby antenna is default, other antenna types can be ordered as options (Stubby antenna price will be deducted)!
- Note! GSM-10SP (case) needs external GSM modem!

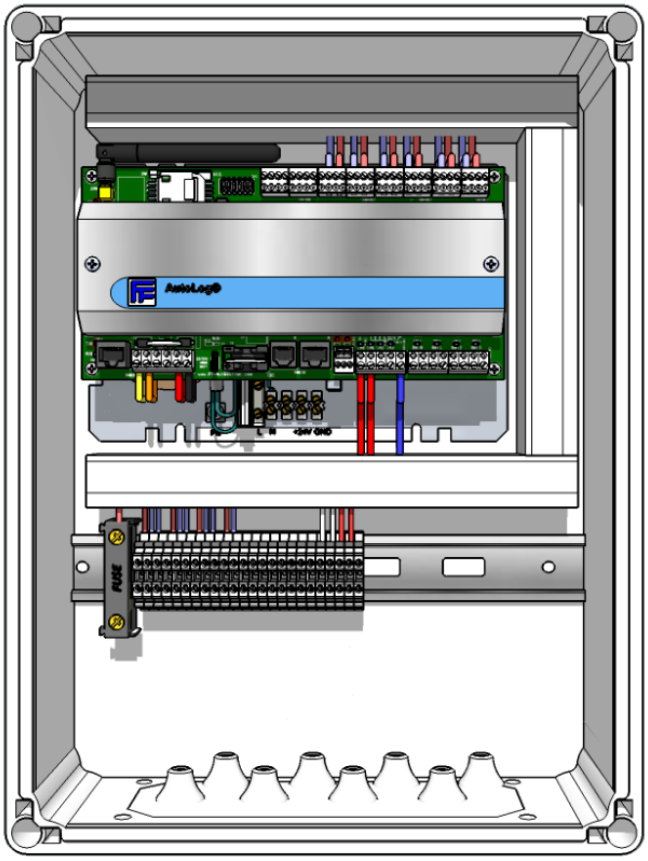
14.9 Customized GSM Units



AutoLog GSM-Units are flexible for different kinds of configurations.

FF-Automation can help customers for selecting optimal components and design

- **GSM-PLC type** (e.g. GSM-20 in the picture)
- **I/O Expansion cards** (only GSM-20 support I/O expansion cards)
- **Analog input modules** (e.g. 4 x 4..20mA Isolated, 4 x Pt100)
- **Terminal blocks (connection strips) for I/O and power** (e.g. Phoenix connectors)
- **I/O to terminal block wirings**
- **Fuse for main power**
- **Relay(s)** (for digital outputs)
- **Antenna type, Internal or external, cable length** (Internal stubby in the picture)
- **Enclosure type, material, size, IP class, mounting, inlet** (e.g. IP66 polycarbonate enclosure with FMC 35F multi-inlets, Size: 300 x 400 x 132 mm)
- **HMI (**type, mounting, HMI can be mounted e.g. to enclosure door)
- **Application program** (FF-Automation can do application programming according to specification or customer can do programming itself after education)
- **Wireless I/O** (FF-Automation's AutoLog WSN master module can be connected to Ser2 using Modbus RTU protocol. Master module communicates with sensor modules using 868MHz/915MHz frequency. Supports automatic routing. EU free channel. 300m distance.
- **ControlMan Setup** (GSM-PLC can be added to ControlMan page)

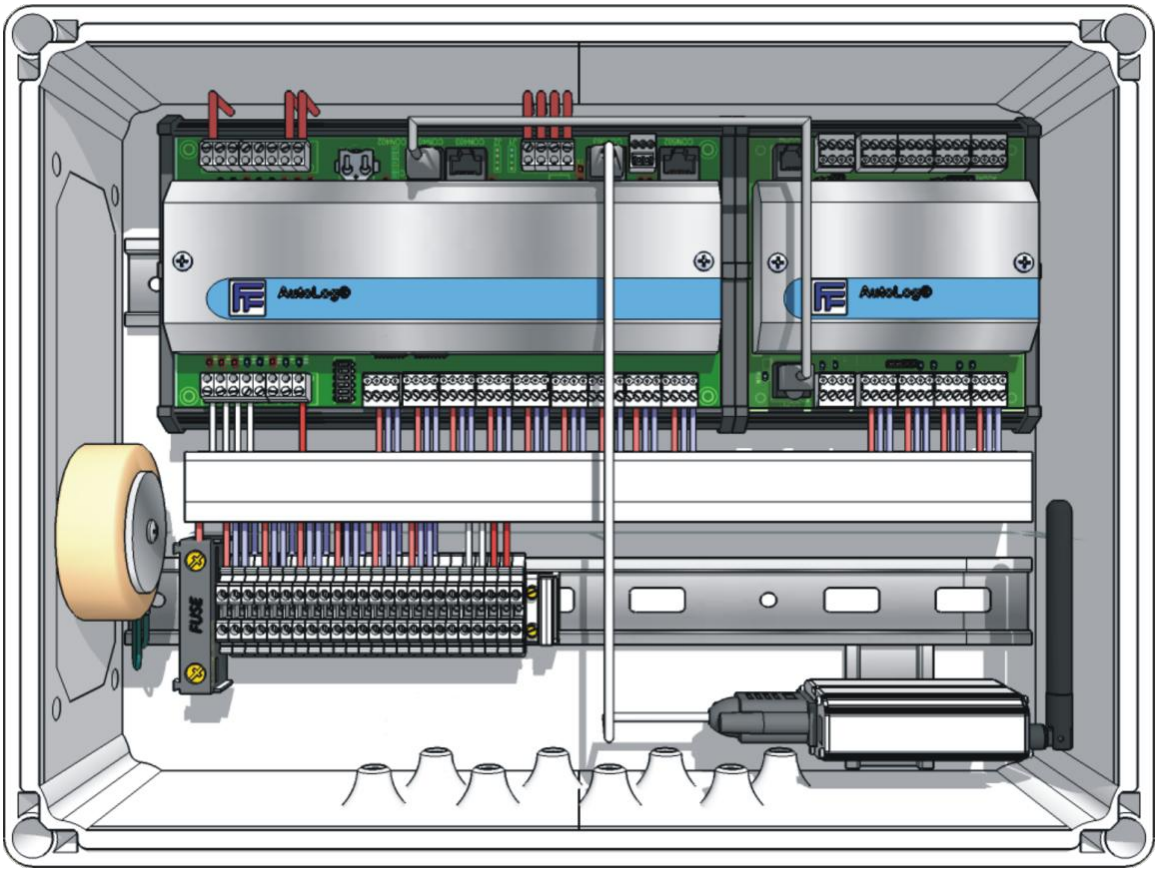


Example 1: GSM-16

14.9.1 Main Options:

- **GSM-PLC type** (e.g. GSM-16 in the picture)
- **Analog input modules** (e.g. 6 x 4..20mA Isolated)
- **Terminal blocks (connection strips) for I/O and power** (e.g. Phoenix connectors)
- **I/O to terminal block wirings**
- **Fuse for main power**
- **Antenna type, Internal or external, cable length** (Internal stubby in the picture)
- **Enclosure type, material, size, IP class, mounting, inlet** (e.g. IP66 polycarbonate enclosure with FMC 35F multi-inlets, Size: 300 x 400 x 132 mm)
- **HMI** (type, mounting, HMI can be mounted e.g. to enclosure door)
- **Application program** (FF-Automation can do application programming according to specification or customer can do programming itself after education)
- **ControlMan Setup** (GSM-PLC can be added to ControlMan page)

FF-Automation can help customers for selecting optimal components and design.



Example 2: GSM-20 + EXA8/4 Analog expansion card

14.9.2 Main Options:

- **GSM-PLC type** (e.g. GSM-20 in the picture)
- **I/O Expansion cards** (only GSM-20 support I/O expansion cards, EXA8/4 in the picture)
- **Analog input modules** (e.g. 12 x 4..20mA Isolated)
- **Terminal blocks (connection strips) for I/O and power** (e.g. Phoenix connectors)
- **I/O to terminal block wirings**
- **Fuse for main power**
- **Antenna type, Internal or external, cable length** (Internal stubby in the picture)
- **Enclosure type, material, size, IP class, mounting, inlet** (e.g. IP66 polycarbonate enclosure with FMC 35F multi-inlets, Size: 300 x 400 x 132 mm)
- **HMI (**type, mounting, HMI can be mounted e.g. to enclosure door)
- **Application program (**FF-Automation can do application programming according to specification or customer can do programming itself after education)
- **ControlMan Setup** (GSM-PLC can be added to ControlMan page)

FF-Automation can help customers for selecting optimal components and design

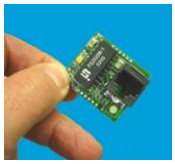
14.10 GSM-PLC Accessories

14.10.1 External GSM-Modem (for GSM-20 only)

Most of AutoLog® GSM-PLCs are using integrated onboard GSM modems. Only GSM-20AN needs external GSM modem. GSM-20 UNIT includes this external GSM modem.

EXTERNAL GSM MODEM (FOR GSM-20)	
900701	WAVECOM FASTRACK XTEND GSM/GPRS modem

14.10.2 Serial Plug-in Modules (PiM) for serial port 2



AutoLog® Serial Plug-in modules (PiM) can be used to convert the serial ports functionality to support special communication methods. AutoLog® Serial Plug-in modules can be connected to GSM-PLC's serial port plug-in connectors. All AutoLog GSM-PLCs (GSM-GW, GSM-4, GSM-8, GSM-16 and GSM-20) support AutoLog® Plug-in module (PiM). Reserves one serial port (Ser2).

Serial Plug-in Modules (FOR SERIAL PORT 2)		
901157	AL RS485 PiM ,ISOLATED	RS-232 to RS-485 conversion module, isolated. Reserves one serial port.
901165	AL RS485NI PiM, NOT ISO	RS-232 to RS-485 conversion module, not isolated. Reserves one serial port.
900065	AL MODBUS TCP PiM	RS-232 to Ethernet Converter, Modbus to Modbus TCP protocol conversion. Reserves one serial port.
900066	AL OEM TCP PiM	RS-232 to TCP/IP Converter, STANDARD, for OEM users for changing Modbus RTU to own protocol. Reserves one serial port.

14.10.3 GSM Antennas

AutoLog® GSM-PLCs needs GSM antenna. Left Up (Cone), Right Up (Miniw), Left down (Stubby), Right down (Whip). GSM-UNIT includes GSM antenna, please specify the antenna type when ordering!

GSM ANTENNAS	
900677	GSM ANTENNA CONE SMA 0.3m
900678	GSM ANTENNA CONE SMA 2.5m
900679	GSM ANTENNA MINIW SMA 0.3m
900703	GSM ANTENNA MINIW SMA 2.5m
900706	WHIP ANTENNA SMA 2.5m
900707	STUBBY GSM ANTENNA SMA91

14.10.4 Analog input modules

AutoLog GSM-PLCs needs to be equipped with analog input modules before analog inputs can be used. Analog input modules can be used in GSM-PLCs (GSM-8, GSM-16 and GSM-20).

ANALOG INPUT MODULES	
900781	AI Module, Pt100, -50…+150 °C
900782	AI Module, Pt100, 0…+500 °C
900784	AI Module, Pt100, 0…+800 °C
900789	AI Module, Pt100, 0…+250 °C
900805	AI Module, Pt1000, -50…+150 °C
900707	MOD.for TMP SENSOR
900798	AI Module KTY, -50 ... +150 °C
900797	AI Module NTC, -5 ... +50 °C
900796	AI Module, 0…5mA, not isolated
900795	AI Module, 0…20mA, not isolated
900799	AI Module, 4…20mA, not isolated
900800	AI Module, 4…20mA, isolated
900802	AL20 Special AI Module, Customer defined
900806	AI Module, 0…1V, not isolated
900783	AI Module, 0…2V, not isolated
900788	AI Module, 0…5V, not isolated
900785	AI Module, 0…10V, not isolated
900786	AI Module, -4…+4V, not isolated
900787	AI Module, 0…+20V, not isolated
900791	AI Module, -10…+10V, not isolated
900790	AI RMS Module, 0…+0.333VAC, not isolated
900776	AI RMS Module, 0…+0.25VAC, not isolated
900808	AI RMS Module, 0…+25VAC, not isolated
900807	AI RMS Module, 0…+40VAC, not isolated
900809	AI Module, LIGHT Sensor
900804	AI Module, Pt100, -50…+150 °C

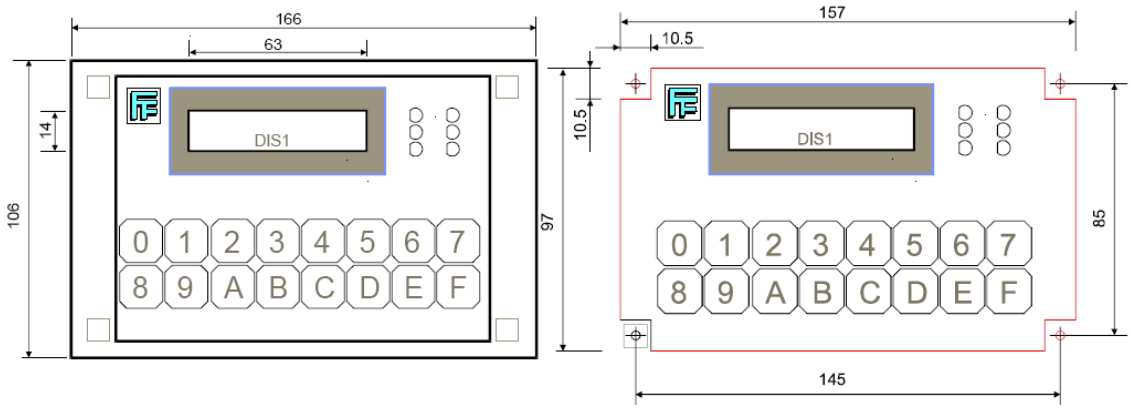
14.10.5 Analog to Digital Inputs conversion modules

Analog input can be converted to digital input using analog to digital conversion module. E.g. if its used in AL20AN which has 8 analog inputs, these inputs can be converted to additional 8 digital inputs. Conversion module works all AutoLog which have AI module places (AL4 - AL20) and all GSM-PLCs (GSM8, GSM-16, GSM20). These cannot be used in AL10SP.

ANALOG TO DIGITAL CONVERSION MODULES		
Code	**Designation**	**Description**
900792	AI to DI,Isolated	AI to DI conversion module, isolated

15 AutoLog GSM-PLC Accessories

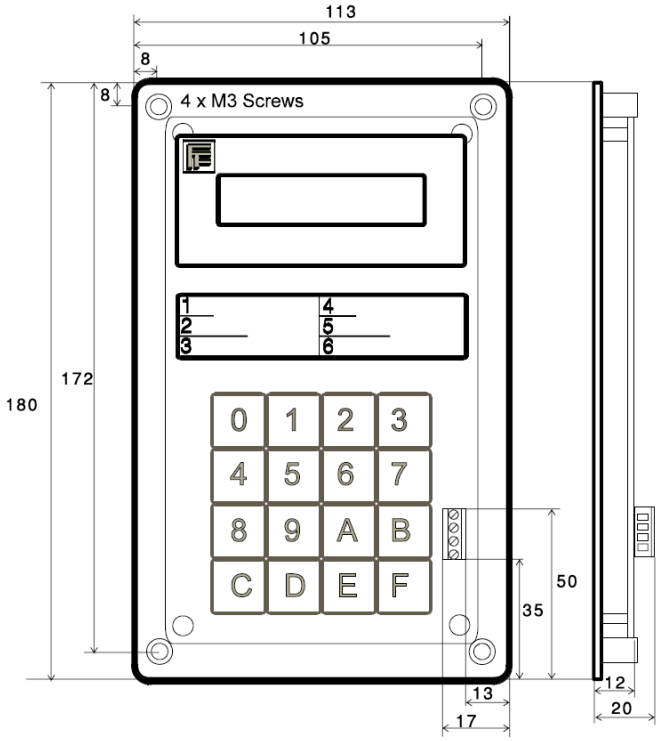
15.1 Display / keypad unit AL1093D, AL1093F



AL1093D dimensions

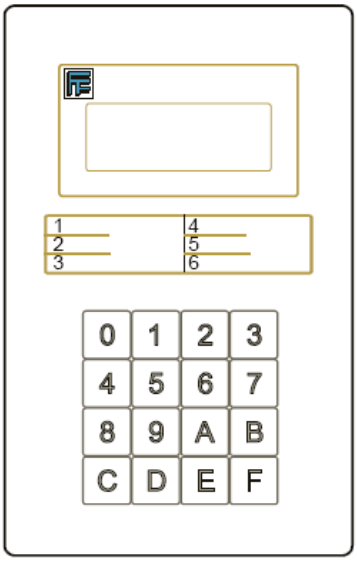
AL1093D/F:

Display	2x16 character alphanumeric LCD display, backlit
Function keys	16 keys (0 - F), each key can control one binary output in PLC (16 pcs)
Timing control	Contains clock and calendar, battery backup
Power supply	powered through I^2C connection cable (1m) from CPU board.
Protection	AL1093D is front panel mount model, rear construction is open and the unit can be fitted with a client-designed face plate. Model AL1093F is encapsulated, for front panel mounting, front panel seal IP54 and the display can be fitted with client designed face plate.



AL1093F & AL1420 Dimensions

15.2 Display / keypad unit AL1420

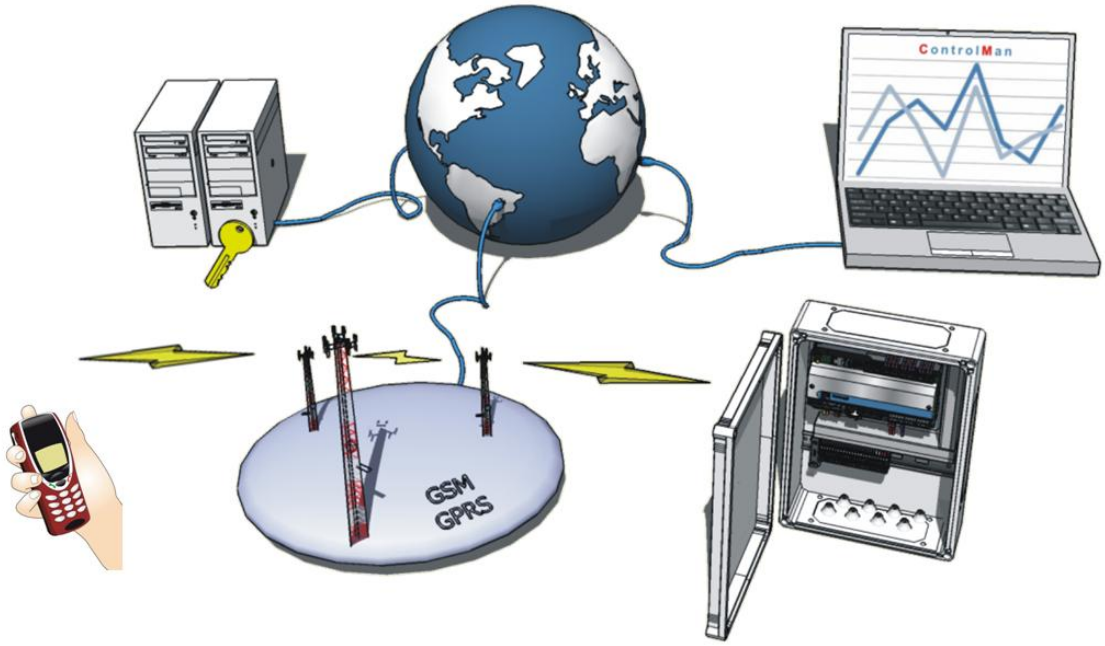


AL1420:

Display	4 x 20 character alphanumeric LCD display, backlit
Keypad	16 keys (0 - F) each key can control bit output
Indicator lights	6 LEDs, controlled by bit outputs, buzzer
Power supply	supply voltage through system cable (0.9m) from PLC's supply unit.
Protection	open structure, not encapsulated. The display unit can be fitted with a client-designed face plate.

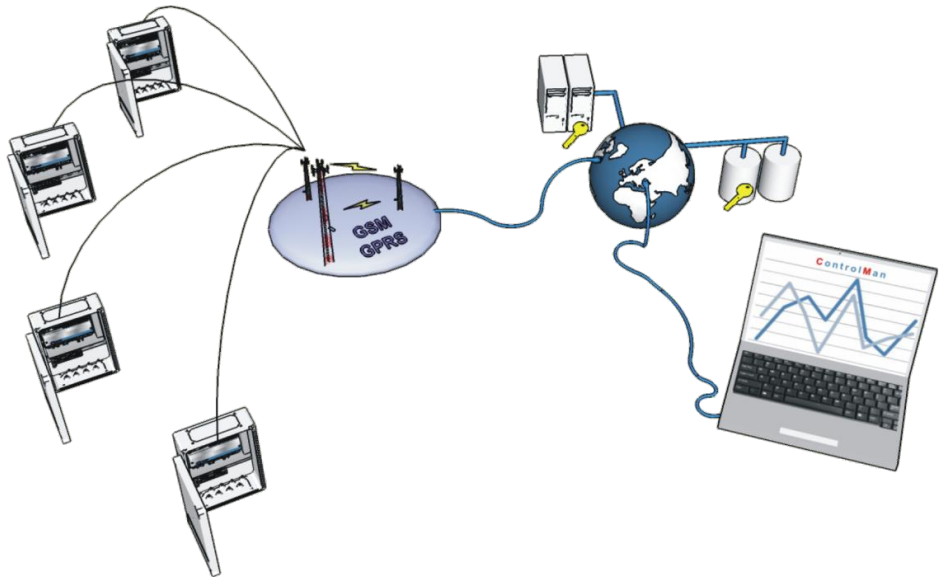
SCADA SYSTEMS & INTERFACES

16 AutoLog ControlMan - SCADA as Web Service (Cloud SCADA)



AutoLog ControlMan is Cloud SCADA Service for GSM control units. It is hosted internet application service. AutoLog ControlMan can be opened with normal web browser from anywhere, anytime if you have Internet access. System allows many simultaneous users.

AutoLog GSM-RTUs are installed along the remote targets to perform measurement and control tasks. Control units are communicating with AutoLog ControlMan server using GSM- and Internet networks.



System can be expanded.lmost unlimitedly. System allows creating many users and user groups with different user levels. System can be remotely maintained.
Read more!

16.1 Introduction

FF-Automation's AutoLog ControlMan is ultra modern and cost effective solution for remote supervision and control. It is suitable for remote monitoring and controlling almost anything: machines, devices, processes, pipelines, tanks, street lights, pumps, valves, real estates, unmanned stations, cold rooms, cargoes, environment, flood etc. If you have many targets which are hard to reach and you want to have a way to monitor and control them, then AutoLog -ControlMan is most likely what you are searching for.

16.1.1 Wireless supervision and control service via GSM and Internet

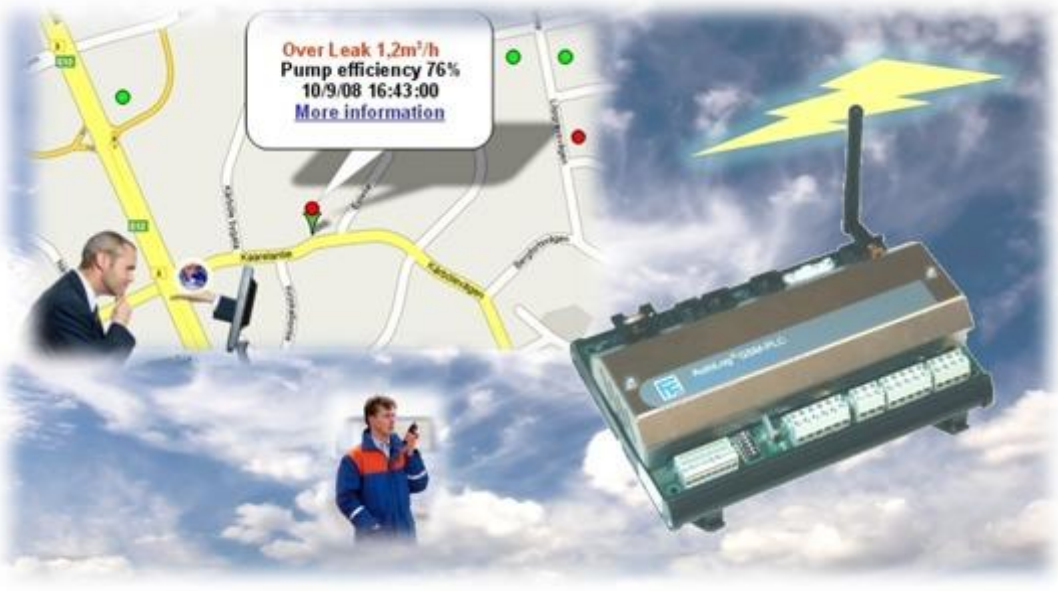
AutoLog ControlMan is complete solution which includes both AutoLog GSM-RTU units and hosted Internet application service. GSM-RTUs are installed along the remote targets to perform measurement and control tasks. GSM-RTUs are communicating wirelessly using GSM/GPRS network. ControlMan web service is used with normal web browser through Internet - anywhere, anytime!

16.1.2 Centralized architecture - global data collection & sharing

Today's trend for global companies is to use centralized Internet server databases and web based application interfaces. This way the information is shared to everybody regardless of the physical location.

AutoLog ControlMan integrates easily with other systems and databases. Open standard interfaces and mainstream technology guarantees long term and cost effective solution. New features are constantly included to the service to follow today's and future needs.

AutoLog ControlMan uses global GSM- and Internet networks for communication. Information can be collected and shared with all participants regardless of their physical location.



16.1.3 Global Networks - ready to use without investments

AutoLog ControlMan uses existing and global communication solutions. Measurement data is send wirelessly via existing and almost global GSM and Internet networks. Users log in to service through Internet network. No network investment costs!

System uses *local GSM operators* to minimize the communication costs. GSM communication uses Internet between local GSM operator and ControlMan server.

16.1.4 Hosted Service- Cost efficient and care-free solution !

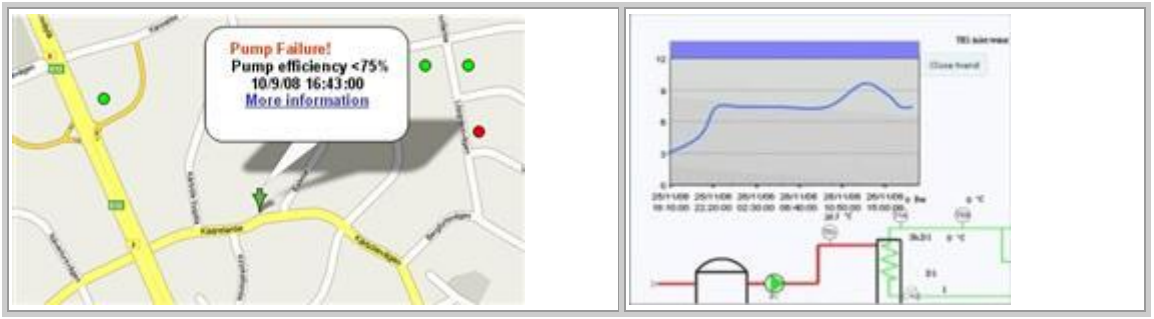
AutoLog ControlMan is hosted Internet "control room" service for remote target supervision and control. Server is hosted (24/7/365) by Verio(*), which is awarded as most reliable Internet hosting company in 2010. Customer doesn't need own server PC or maintenance personnel. Existing PCs can be used to connect to server application without any software installations. User just needs user name and password to log in to service through web browser.

(*) ControlMan can be hosted also by other hosting companies or it can run on customer's own server.

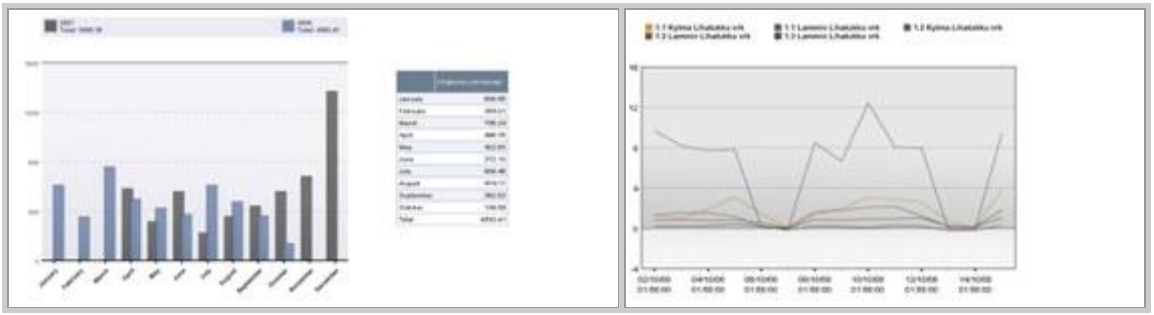
16.1.5 Graphical User Interface (GUI) views

System includes lots of technology, but normal user doesn't need to know or care about this. User just wants to see the status of their remote targets as easily and illustratively as possible. ControlMan gives versatile *Graphical User Interface* including dynamic maps, measurement trends, alarms views, animated process views, reports etc. Views can be designed according to application specifications.

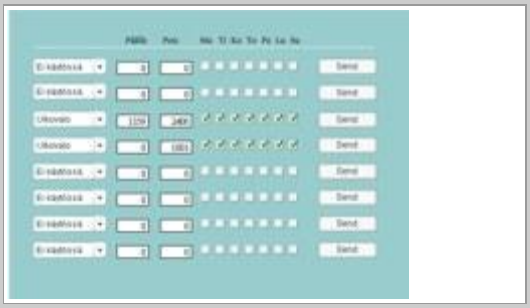
User can log in to the system from any PC with Internet access using web browser. User doesn't need to install any software for using ControlMan.



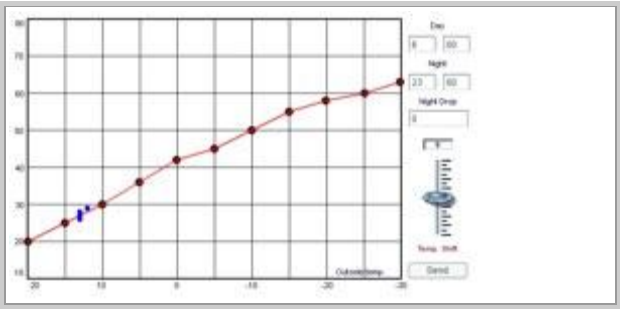
On *dynamic map view* user can see the target site locations and alarm information. Clicking the object on the map opens site views for further analysis. On the rights there is example of *dynamic process view* which gives immediate information of the process status. User can for example click the valve symbol to set new set point which is automatically sent to control unit.



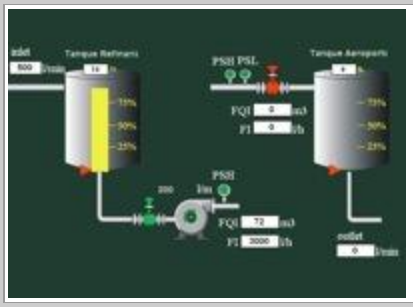
Measurements are stored to database. This data can be analyzed in many ways, e.g. using cumulative bar graph- and trends views. These views give the ability to see the process history and to detect e.g. the symptoms of some problems before it's too late.



From alarm view user can see active alarms. Alarms can be acknowledged. On the right there is example of a week control program. Control unit has own clock & calendar, so week programs can be loaded to control unit's memory. Targets can be controlled according to defined times and days according to this program.



An example of XY -control curve. E.g. building heaters / air conditioners can be controlled according to outside temperature. On the right there is an example of parameter setting view for alarm limits and PID control units.



FF-Automation can make views according to customer application specification.

Process view, from where user can control valves and see tanks surface levels and flow rates.

16.1.6 Database

ControlMan setup information, control unit definitions, users, user groups, measurement history etc. are stored in ControlMan server's SQL database. ControlMan server database can be connected to other systems using standard ODBC interface and SQL queries. If needed user can upload measurement history from server database to own PC for offline analysis.

16.1.7 Communication

GSM control units communicate wirelessly through GSM network to local GSM operator access point. GPRS communication from operator's access point to ControlMan server is using Internet IP network. Virtual Private Network (VPN) can be used if local GSM operator supports it, but in most cases it is not needed.

Major part of typical communication is logged measurement data which is transferred using GPRS packet data. GSM/SMS messages can be used for sending alarms, controls, report

queries etc. SMS message communication allows also direct communication between control units and between control unit and GSM phone. Serviceman can for example send direct control or setup command to control unit from GSM phone using SMS message.

GSM control unit can be controlled also using free call controls. Control unit can identify incoming phone number and make defined controls.

16.1.8 Expandability

The system uses global communication networks and it operates on hosted servers, so it can be expanded unlimitedly.

16.1.9 Remote Maintenance

Control units can be programmed remotely through GSM network. ControlMan application pages can be maintained remotely through Internet network.

16.1.10 Safety

- Operates only from defined phone numbers
- Password identification when programming control unit
- Secret phone numbers
- User name and password authentication for Web pages
- Possibility to use VPN if local GSM operator supports it. (Normally it's not needed).

16.1.11 Mobile Applications

	FF-Automation can make views according to customer application specification.

Normally graphical user interface views are browsed with desktop PC's Internet browser, but ControlMan pages can be designed to be used also with small resolution mobile phones. Mobile phone's Internet browser should have Flash support.
Older GSM phones can be used to receive SMS text messages like alarms, work orders (Trouble ticket system), measurement reports and also to send control commands, set point changes, acknowledgements etc.

16.2 Complete Solutions

16.2.1 AutoLog product family:

- from I/O level (also wireless I/O and sensors if needed)
- to Programmable Control unit (RTU= Remote Terminal Units)
- to Communication equipments (GSM, TCP/IP, Wi-Fi, Ethernet, etc.)
- to Server Solutions (ControlMan, Web Studio SCADA system)
- to Graphical User interfaces and Maintenance systems (Web based)

16.2.2 FF-Automation project services:

- Project planning
  - Hardware, software and communication architecture
  - Project planning documents
- Project design
  - Application programming
  - Communication design
  - Graphical user interface design
  - Project documentation
- Factory acceptance tests (FAT)
- Commissioning and Site acceptance tests (SAT)
- User and maintenance training
- 10 years spare part guarantee
- Support, maintenance, training
- Long term relationships, long life span systems, upgrades with new technology.

16.2.3 Distributors and System integrators

FF-Automation can make very beneficial agreements with distributors and system integrators. The level of technical support depends on need and capacity. FF-Automation can e.g. design the system, help commissioning and maintain the system etc. so in other words do all technical works. Step by step companies can start designing and maintaining ControlMan systems independently. FF-Automation provides training when needed.

For example Finnish company C2 Information Systems started by marketing and distributing ControlMan system to street light control and energy saving projects. Now they are commissioning and maintaining ControlMan projects independently. Over 20 cities are using ControlMan for controlling street lights.

If you have special knowledge from some business area we can give you exclusive user right for this business sector. You can use ControlMan also as general solution for remote control and monitoring projects.

16.3 Applications

Picture: Street Light control, Tank monitoring, Low power Pipeline monitoring.

- AutoLog ControlMan can be used in many different kinds of applications for example:
- Pipeline Cathodic Protection monitoring,
- Tank level monitoring,
- Street light controlling & dimming (energy saving system),
- Water pumping station monitoring and control,
- Real estate monitoring and control,
- Device and machine monitoring and control,
- Base station condition monitoring,
- Temperature monitoring and controlling in cold rooms etc,
- Environment monitoring applications, flood detection etc,
- Process monitoring
- Road applications, Road signs, traffic monitoring, light control, etc,
- Integrated OEM solutions,
- Low power applications (located out of power network),

- Wireless Wi-Fi sensor projects (GSM control unit works as access points)

And any other remote monitoring and controlling application! Ask more!

16.4 Benefits of ControlMan

- Cost effective and care-free solution. No need for expensive control room server PC, new operator workstations. No need for own maintenance personnel.
- Uses existing and global GSM and Internet communication solutions. No network investment costs.
- Maintenance changes and application programming can be done remotely. No need for expensive on-site maintenance visits e.g. when adding new devices.
- Long term solution - very flexible for future expansions and integrating with new technologies. Based on mainstream technologies.

- Easy to use interface to analyze and learn the behavior of remote targets and react to possible failure or abnormal situations before it's too late.
- Increase total income. Increased productivity, less failures, less on-site visits.
- Shared knowhow. Information can be shared between authorized user groups with different user levels. - Also management knows what is happening!
- Critical alarms are automatically detected and send immediately to repairman’s mobile phone or e-mail. Automatic work order generations.

17 Indusoft Web Studio SCADA interface

FF-Automation has developed GSM/GPRS driver for Indusoft Web Studio which allows:
1) Easy alarm and report forwarding to GSM phones
2) Bidirectional communication with AutoLog GSM-RTUs using SMS
3) Receiving measurement data from AutoLog GSM-PLC via GPRS/FTP

http://www.ff-automation.com/products/SCADA.shtml

OR

www.indusoft.com

Ask more info!

Ready-made:

Installation and user guide for AutoLog GSM driver - documentation!

Demo applications for AutoLog GSM-PLC using SMS and FTP communication and alarm sending!

18 Custom SCADA interface

FF-Automation has developed GSM/GPRS driver for Postgre Database which allows:
1) Easy alarm and report forwarding to GSM phones
2) Bidirectional communication with AutoLog GSM-RTUs using SMS
3) Receiving measurement data from AutoLog GSM-PLC via GPRS/FTP

This database can be used as interface between any SCADA software and AutoLog GSM-PLCs.

THANK YOU FOR READING THIS MANUAL!

If you found any errors in this manual or have any suggestions, please let us know!

Note that this manual is still under heavy construction and you will receive frequent updates for it. You can download the latest version of this manual from our web page.

http://www.ff-automation.com/download/Distributor_signin.shtml

Please ask password from FF-Automation Oy!

E.g. to: antti.moijanen@ff-automation.com cc: info@ff-automation.com