

# GSM-1 User Manual

rev. 1.09

applies to GSM-1 v3.0, v3.1 and v3.2  
use CyPro 2.6.0 or later



# Index

Index.....	1
Overview .....	2
Quick setup .....	3
1. Insert SIM.....	3
2. Edit configuration file.....	3
3. Send configuration file .....	3
4. Check operation.....	4
Connection options.....	6
Configuration .....	7
Description .....	7
Configuration file .....	7
Input/output variables .....	9
Input/output .....	9
Status bits .....	9
Status words .....	9
Module info.....	9
Incoming voice call and SMS.....	9
Outgoing voice call and SMS.....	9
Push message .....	9
Basic functions .....	10
SMS read/write.....	10
Alarm function .....	10
Incoming voice call.....	11
Outgoing voice call.....	11
Working with plc program.....	12
Incoming SMS.....	12
Outgoing SMS.....	12
Incoming voice calls.....	12
Outgoing voice calls.....	12
VoiceAndSMSAPI variables.....	12
Wiring schematics .....	13
Digital inputs, 24V output .....	13
Output using 230V contactor .....	13
Output using 230V bistable relay.....	13
Technical specifications .....	14
Mounting .....	14
Class B operation.....	14
LED signalization .....	14
Versions and upgrade .....	15

# Overview

GSM-1 is GSM/GPRS module for CyBro-2. Basic functions are:

## SMS

- read and write CyBro variables
- send SMS when a variable is set

## Voice

- call module to perform an action
- perform call when condition is satisfied

## GPRS

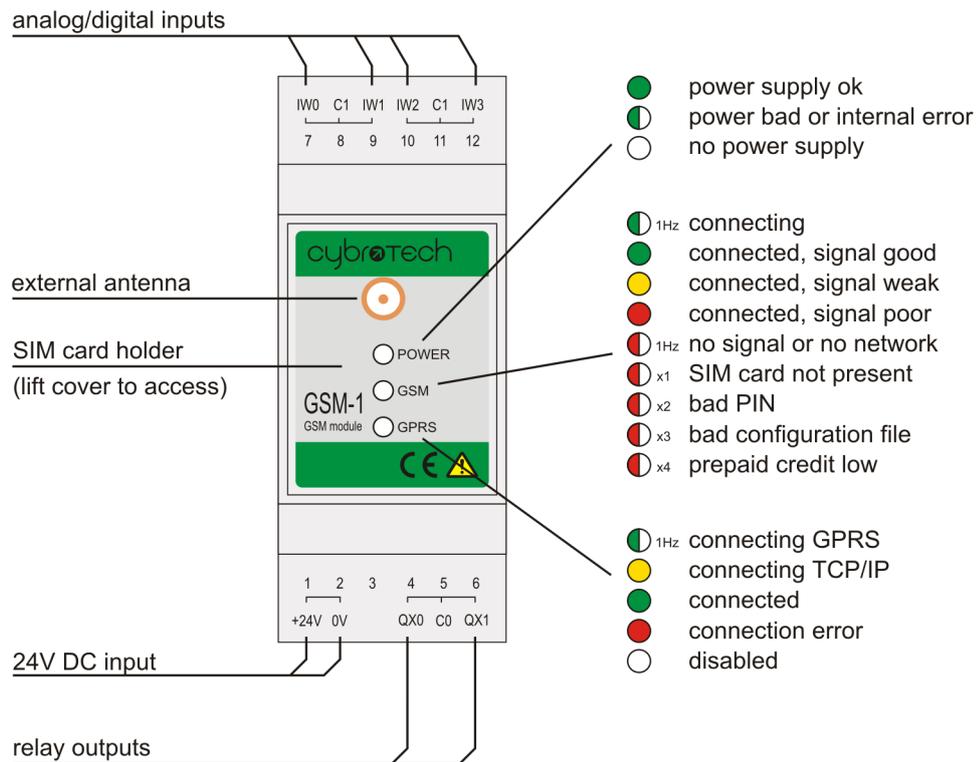
- remote programming and monitoring with CyPro
- remote data transfer with CyBro OPC server

GSM-1 connects to IEX bus, together with other i/o modules. CyBro serial port is not used. To use GSM-1, no programming is needed. Set a few configuration options, and module is ready.

## Hardware:

- three-band 900/1800/1900MHz
- internal GSM antenna
- external antenna connector (SMA type), automatic switch
- 4 analog/digital inputs
- 2 relay outputs

Unlike other IEX modules, only a single GSM-1 can be connected to CyBro.

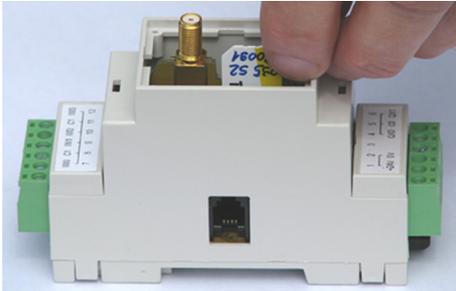


For a different product versions, please check "Versions and upgrade" chapter.

# Quick setup

## 1. Insert SIM

Open front cover and insert SIM in card holder.



## 2. Edit configuration file

Edit configuration file (CyPro/Examples/GsmIniDemo.ini) according to your data:

### Common

If your SIM card asks for a PIN, uncomment **SimPin** and enter correct PIN number.

```
[GSM]
SimPin=1234
```

### Voice

Enter your mobile phone number in **OutgoingPhones** group.

```
[OutgoingPhones]
+385919146308
```

### GPRS

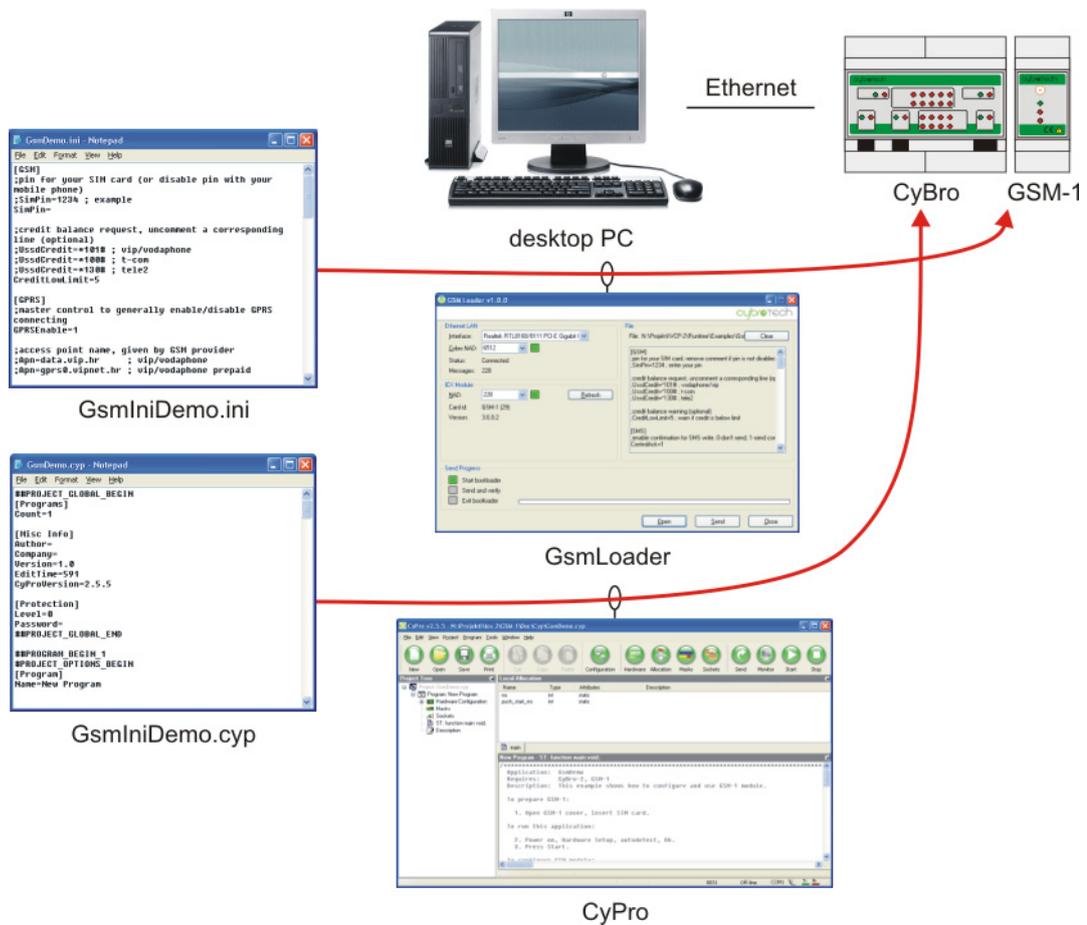
- Make sure that SIM card is data-enabled, ask your GSM provider for details.
- Set **Apn**, **Userld**, **Password** and **Dnslp** according to data provided by your GSM provider.
- Set **NAD** to network address of your CyBro. Use serial number, not alias.
- Set **Pushlp** to IP address (or domain name) where push message should be sent.

```
[GPRS]
Apn=data.vip.hr
Userld=38591
Password=38591
Dnslp=212.91.97.3

[Push]
NAD=8031
Pushlp=myhome.getmyip.com
```

## 3. Send configuration file

- Connect CyBro, GSM-1, power supply and ethernet.
- Start GsmLoader, select CyBro and GSM-1.
- Open and send GsmIniDemo.ini.
- Close GsmLoader (don't skip this).
- Start CyPro.
- Open GsmIniDemo.cyp, autodetect and start.



After transfer is complete, module will reset automatically. Minute or two later, it will connect to network, and both GSM and GPRS LEDs will be green. If configuration file has errors, GSM LED will blink.

## 4. Check operation

### SMS

Send SMS "cybro\_qx02=1" to GSM-1.

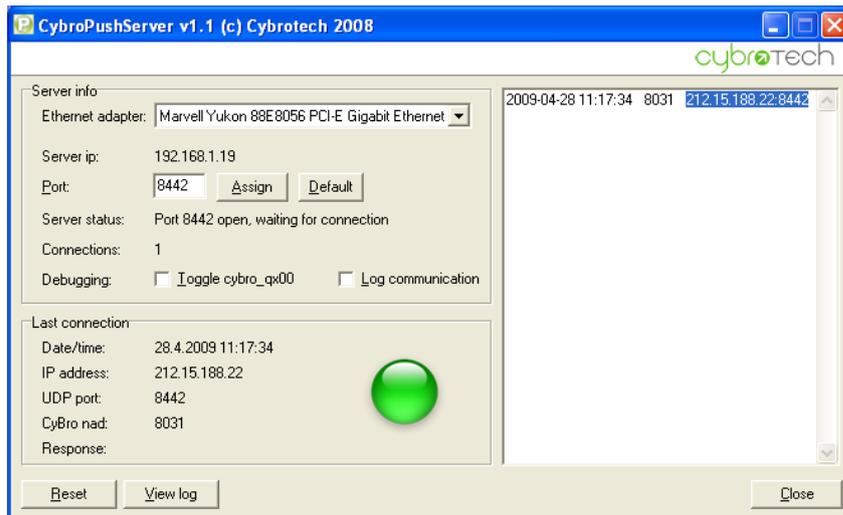
In a few seconds, relay will turn on. Confirmation message (if configured) will be received soon.

### Voice

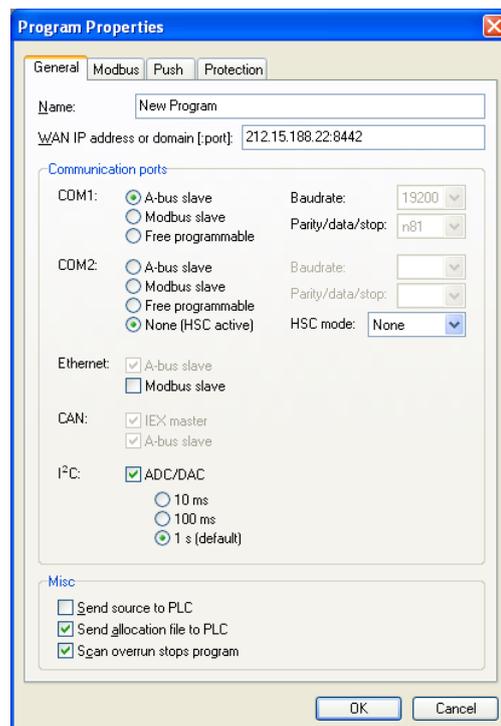
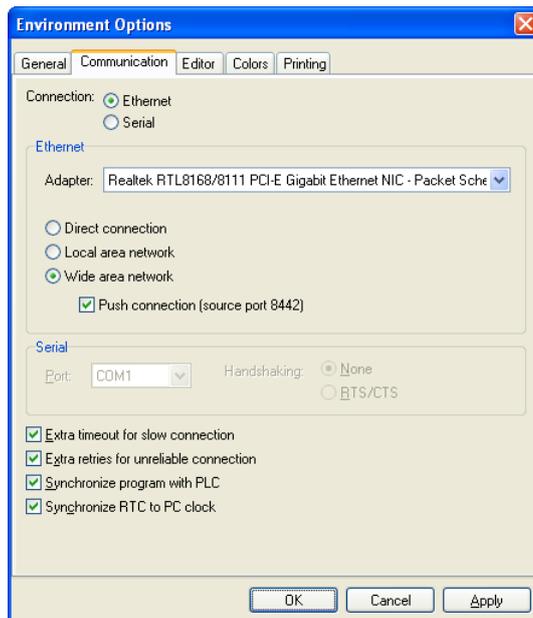
- Open variable monitor.
- Call GSM-1 from your mobile phone.
- Variable `gsm_incoming_call` becomes 1, and `gsm_incoming_rings` increments on each ring. Calling number will be displayed in array of variables `incoming_number`.
- Set `gsm_outgoing_index` to 0 and `gsm_outgoing_call_req` to 1 (will reset automatically). Your phone will ring.

### GPRS

- Make sure GSM module is configured to your WAN address (check <http://www.whatismyip.com>).
- Configure router to forward UDP port 8442 to your PC.
- Start CyBroPushServer, wait until push message appears.



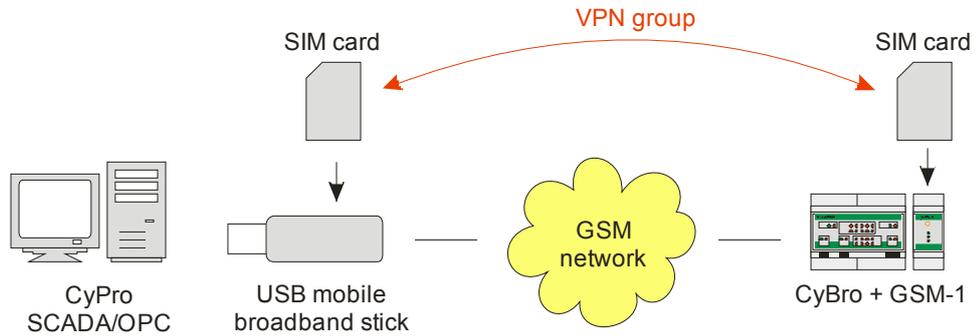
- Remember CyBro IP address and port number, and close CyBroPushServer (don't skip this).
- Start CyPro, open GsmInDemo.cyp.
- Configure communication for WAN, using push connection. Turn on **Extra retries for unreliable connection**. You may also need **Extra timeout for slow connection**.
- Open Configuration and enter IP address and port number from push server.



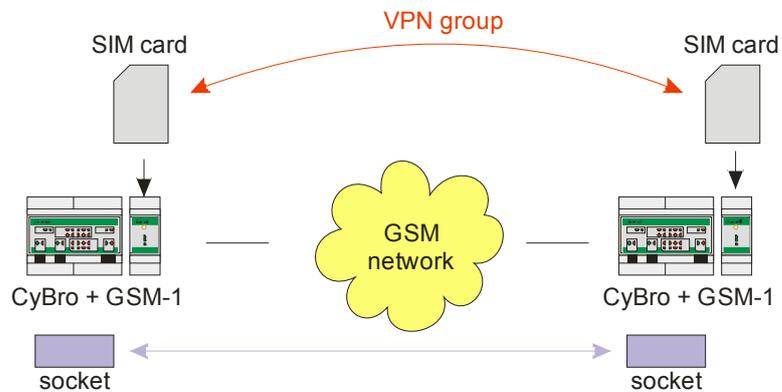
Status bar will show when communication is established.

# Connection options

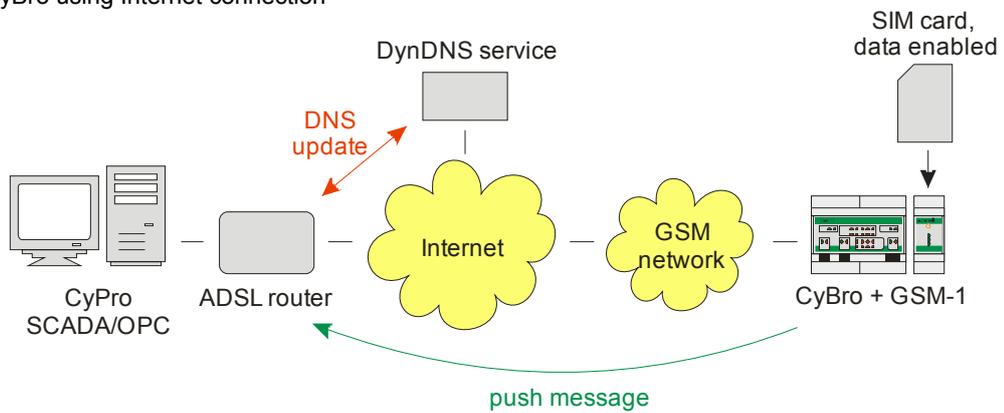
## 1. PC-CyBro using VPN SIM cards



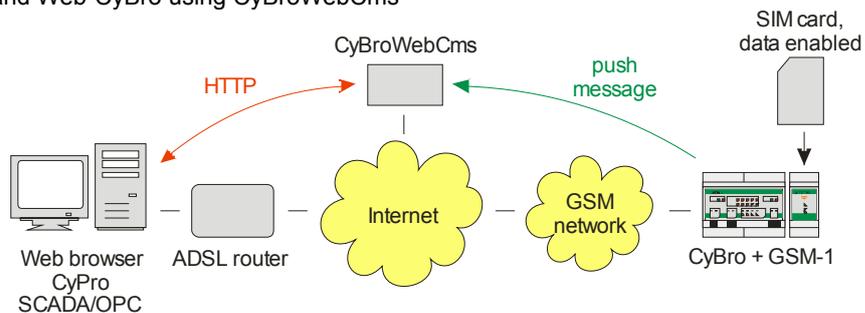
## 2. CyBro-CyBro using VPN SIM cards (UDP broadcast support is required)



## 3. PC-CyBro using Internet connection



## 4. PC-CyBro and Web-CyBro using CyBroWebCms



# Configuration

## Description

GSM-1 is configured by sending configuration ini file. Use text editor to edit, and GsmLoader to send file. Configuration depends on plc program. When allocation is modified, configuration file should be updated.

Parameters are divided by groups. The group name appears in square brackets ([ ]). Semicolon (;) is used to comment a line. Comment continue to the end of the line, everything between semicolon and end of line is ignored.

Maximum size of configuration file is 14Kb. Special or national characters are not allowed, and may cause parsing error.

## Configuration file

### Common

```
[GSM]
SimPin=1234
UssdCredit= *101#
CreditLowLimit=5
```

**SimPin** is needed if PIN of your SIM card is not disabled.

**UssdCredit** is string for checking credit limit, defined by gsm operator. Below **CreditLowLimit** GSM led will begin blinking. Credit is asked once per hour. If **UssdCredit** doesn't exist, checking is not performed.

Note: Be aware that prepaid SIM cards have limited period of validity, expiring of which module can't detect. When in doubt, check SIM in your mobile phone.

```
[IncomingPhones]
+38591123456
+385981234567
```

**IncomingPhones** contains list of phone numbers (international format, beginning with '+'), allowed to call or send SMS to module. If group doesn't exist, everybody have access. If group exist, but it is empty, no one will be able to access.

```
[OutgoingPhones]
+38591123456
+385981234567
```

**OutgoingPhones** contains list of phone numbers (international format, beginning with '+'), which will be called (or sent SMS) in case of an alarm.

### SMS read/write

```
[SMS]
;enable confirmation for SMS write, 0-don't send, 1-send confirmation
ControlAck=1
```

**ControlAck** tag defines weather SMS write confirmation is sent or not.

```
[CybroVars]
cybro_qx00=1024,1
cybro_iw00=2048,2
my_counter=6144,3
```

List of accessible CyBro variables. Variables are used for SMS read/write, or as a content of an alarm message. Use address from allocation file (.alc), generated by CyPro. Alc file contains hexadecimal address, so it should be converted to decimal. Variable type is 1-bit, 2-int, 3-long and 4-real.

## Alarm function

```
[SMSAlarmTrigger]
1024
1025
```

```
[SMSAlarmText]
Alarm: fire in cellar! CO=%co_measured:1%
Alarm: valve %valve is broken! Pressure is %pressure:2
```

Alarm is sent when trigger variable is set to 1. Module will automatically reset variable to 0.

## Voice function

```
[Voice]
;0-leave incoming call ringing, 1-hangup after first ring
RxHangup=0
```

**RxHangup** tag defines weather the incoming call is leave ringing or hanged up.

## GPRS function

```
[GPRS]
GPRSEnable=1 ; Global control to enable or disable GPRS.
Apn=gprs0.vipnet.hr
Userld=38591
Password=38591
DnsIp=212.91.97.3
```

**Apn**, **Userld**, **Password** and **DnsIp** are provided by GSM operator.

```
[Push]
NAD=8031 ; CyBro NAD (serial number, not alias).
PushIp=89.1.3.96 ; IP address or domain name to which push is sent.
PushPort=8442 ; Port to which push is addressed, use 8442.
PushPeriod=60 ; Period of the push repetition in seconds.
```

Push group is used for push message. Purpose of push message is to get IP address of CyBro.

# Input/output variables

## Input/output

`gsm_ix00..gsm_ix03` - digital inputs, 0-off, 1-on  
`gsm_iw00..gsm_iw03` - analog inputs, voltage 0..10V, 0..1023  
`gsm_qx00..gsm_qx01` - relay outputs, 0-off, 1-on

note: digital and analog inputs share same terminals

## Status bits

`gsm_general_error` - one or more system errors occurred (timeout, program or bus error)  
`gsm_timeout_error` - communication failed, no messages are coming from the module  
`gsm_program_error` - internal or configuration error detected, module is not functional  
`gsm_bus_error` - module detected a number of communication errors, but it is still working  
`gsm_connected` - module connected to GSM network (domestic or roaming)  
`gsm_roaming` - domestic network not found, roaming connected  
`gsm_gprs_connected` - module connected to GPRS network

## Status words

`gsm_error_status` - module status: b7:voice\_tx\_error, b6:SMS\_tx\_error, b5:udp\_tx\_error, b4:no\_credit, b3:config\_error, b2:PIN\_error, b1:sim\_error, b0:network\_denied  
`gsm_gprs_status` - status of GPRS network registration (0..7-connecting, 8-connected, 9..11-error)

## Module info

`gsm_signal_quality` - GSM signal quality, range 0..31 (0..7 poor, 8..14 acceptable, 14..31 good)  
`gsm_credit_balance` - credit balance in local currency for prepaid SIM card (e.g. 24 means 24EUR), checked every 60min, always zero for postpaid.

## Incoming voice call and SMS

`gsm_incoming_call` - incoming call active, 1-ringing, 0-hangup  
`gsm_incoming_rings` - number of incoming rings, reset when receiving a call, counting up with each ring  
`gsm_incoming_digit_req` - set to read a single digit of incoming phone number, will reset automatically  
`gsm_incoming_digit_data` - single number of incoming phone number  
`gsm_message_received` - incoming message received (1-message received)  
`gsm_incoming_letter_req` - set to read a single letter of incoming sms message, will reset automatically.  
`gsm_incoming_letter_data` - single letter of incoming sms message

## Outgoing voice call and SMS

`gsm_outgoing_digit_data` - single digit of phone number to write to module  
`gsm_outgoing_digit_req` - set to write a single digit of outgoing phone to module buffer, will reset automatically  
`gsm_outgoing_digit_ack` - acknowledge on outgoing number digit  
`gsm_outgoing_index` - index of phone number to be called; -1 for number from cybro-defined, 0 for first number in the [OutgoingPhones] list, 1 for 2nd, 2 for 3rd...  
`gsm_outgoing_call_req` - set to make an outgoing voice call; phone will ring until call is denied, or for maximum of 60 seconds, will reset automatically  
`gsm_outgoing_letter_data` - single letter of sms message to write to module  
`gsm_outgoing_letter_req` - set to write a single character of outgoing sms message to module buffer, will reset automatically  
`gsm_outgoing_letter_ack` - acknowledge on outgoing message letter  
`gsm_outgoing_sms_req` - set to send custom defined sms message, will reset automatically

## Push message

`gsm_push_message_req` - request to send push message

# Basic functions

## SMS read/write

SMS function provides reading and writing CyBro variables.

Read commands are given as: <variable>=?

Write commands are given as: <variable>=<value>

SMS may contain multiple commands, separated by spaces. Mix of reading and writing is supported.

If reading succeeds, read value is returned. If writing succeeds, read value and 'ok' is returned (only when ControlAck is 1). If reading/writing fails (e.g. wrong name), err is returned.

Module first perform writes, then reads. Result is always an actual CyBro value.

Read variable:

```
send: a=?
```

```
reply: a=5
```

```
error: a=err
```

Write variable:

```
send: a=5
```

```
reply: a=5 ok
```

```
error: a=err
```

Multiple reads and writes:

```
send: a=? b=22 c=? d=35
```

```
reply: a=11 b=22 ok c=err d=30 ok
```

In this example, variable c doesn't exist and variable d is limited by CyBro program. If messaging is not working, it may be because SIM card is data-only.

## Alarm function

When trigger variable is set to 1, related SMS will be sent. After getting request, module will automatically write 0 to trigger variable.

Triggers, messages and phone numbers are located in separate groups:

```
[SMSAlarmTrigger]
```

```
1024
```

```
1025
```

```
[SMSAlarmText]
```

```
The quick brown fox jumps over lazy dog.
```

```
The slow brown dog jumps over lazy fox. Total jumps: %jump_count
```

```
[OutgoingPhones]
```

```
+491112345678
```

```
+491112345678
```

First trigger is related to first message/number, second trigger to second message/number and so on.

Alarm message may show values, in format %<variable name>[:<d>]. Variables should be defined in [CybroVars] group.

If <variable name> is defined, value will be sent, otherwise percent sign and following characters will be sent as text.

The number of decimal places can be specified with :<d>, where d is a single digit number. Integer variables are divided by 10 for each decimal place. Real variables are printed as-is. If the number of decimal places is not specified, floating point numbers will default to 6 decimal digits.

Note: if trigger variable is permanently forced to 1 by CyBro program, GSM-1 will send SMS messages repeatedly.

## Incoming voice call

When incoming call is received, variable `gsm_incoming_call` is set. When call is ended, variable will reset automatically. Variable `gsm_incoming_rings` is incremented on each ring. It will retain value until next call, then it will restart with first ring. In consequence, it will never again reach zero.

If `RxHangup` in `Voice` group is set, call is rejected after the first ring.

If voice call is not working, it may be because SIM card is data-only.

## Outgoing voice call

Outgoing voice call on numbers defined in [OutgoingNumbers] can be activated by CyBro program as follows:

- write phone number index to `gsm_outgoing_index`
- set `gsm_outgoing_call_req` to 1

Phone will ring until call is denied, or for maximum of 60 seconds. Will reset automatically.

If voice call is not working, it's possible that your SIM card is data-only, with no voice capabilities.

# Working with plc program

Note: this section applies to GSM-1 v3.1 only.

CyBro program has complete control on voice calls and SMS, both received and sent.

An example for such operation is GsmPlcDemo.cyp. Interface functionality is given in [VoiceAndSMSAPI](#). To use in your application, copy/paste all functions and variables.

## Incoming SMS

Negative flank of variable [incoming\\_message\\_reading](#) indicates that SMS is received. Message content and sender number are available in [incoming\\_message](#) and [incoming\\_number](#). CyBro will receive all messages, including those already handled by module.

## Outgoing SMS

To send a SMS, write text to [outgoing\\_message](#) and phone number to [outgoing\\_number](#), then set [send\\_message](#) to 1 (will reset automatically).

## Incoming voice calls

On incoming call variable [gsm\\_incoming\\_call](#) becomes 1, and [gsm\\_incoming\\_rings](#) increments on each ring. Negative flank of variable [incoming\\_number\\_reading](#) indicates that number is available in [incoming\\_number](#).

## Outgoing voice calls

To make a voice call, write phone number to [outgoing\\_number](#) and set [outgoing\\_call](#) to 1 (will reset automatically). Call will last until denied on receiving phone, or for a maximum of 60 seconds.

## VoiceAndSMSAPI variables

### Interface

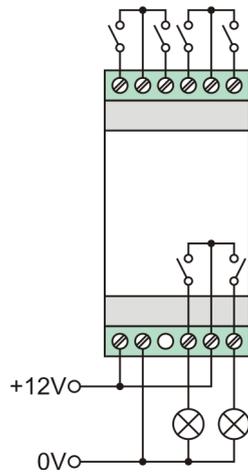
[incoming\\_number\[20\]](#) – buffer for incoming phone number, ASCII, zero terminated  
[incoming\\_number\\_reading](#) – flag indicating reading of incoming number  
[incoming\\_message\[160\]](#) – buffer for incoming message content, ASCII, zero terminated  
[incoming\\_message\\_reading](#) – flag indicating reading of incoming message  
[outgoing\\_number\[20\]](#) – buffer for outgoing phone number, ASCII, zero terminated  
[outgoing\\_call](#) – set to make a call  
[outgoing\\_message\[160\]](#) – buffer for outgoing message content, ASCII, zero terminated  
[send\\_message](#) – set to send an sms

### Local

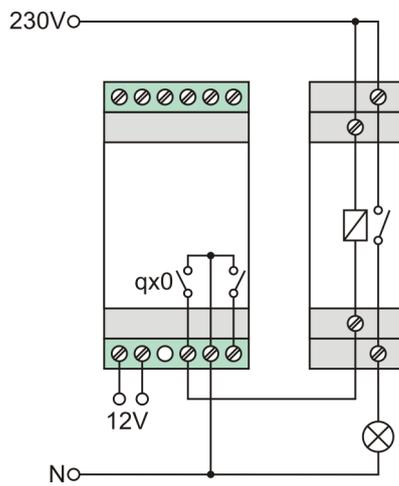
[incoming\\_number\\_size](#) – number of incoming phone digits  
[incoming\\_message\\_size](#) – number of letters in incoming message  
[number\\_write\\_index](#) – index of currently writing outgoing number  
[outgoing\\_number\\_writing](#) – flag indicating writing of outgoing number  
[sms\\_write\\_index](#) – index of currently writing outgoing sms letter  
[outgoing\\_message\\_writing](#) – flag indicating writing of outgoing sms message

# Wiring schematics

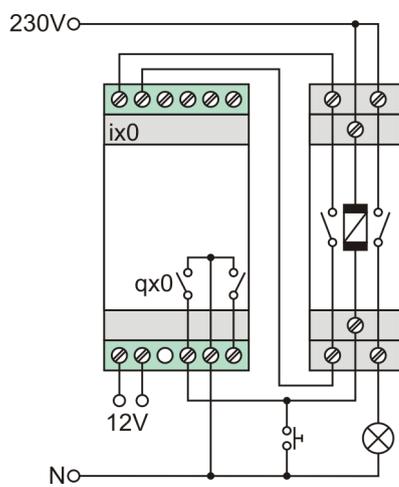
## Digital inputs, 24V output



## Output using 230V contactor



## Output using 230V bistable relay



## Technical specifications

GSM module	three-band 900/1800/1900MHz
GPRS class	mobile station class B
Antenna	internal or external
Connector	SMA with automatic switch
External antenna	50ohm nominal
Input type	digital / analog 0..10V
Internal pull-up	12V, 2mA
A/D converter	10-bit
Analog readout	0..1023
Accuracy	5% FSR at 25°C
Update time	5ms typ.
Output type	relay contact, normally open
Load	max. 3A/250V AC or 3A/30V DC, resistive
Update time	5ms typ.
Power supply	24V DC (18-26V)
Power consumption	75mA standby, 100mA active
Operating conditions	0..50°C, 0..85% rh non-condensing
Dimensions	36x99x70mm

For a stand-alone operation, connect power supply to +24V and GND terminals.

## Mounting

When operating on internal antenna, it is recommended to mount GSM-1 away from CyBro and other modules. Placing at top of cabinet will provide a better signal, resulting in more stable communication.

## Class B operation

Mobile station class B can connect to both GSM and GPRS, but only one connection may be used at a time. During GSM service (voice call or SMS), GPRS is suspended, then resumed automatically after the GSM is concluded. The consequences are:

- receiving SMS is postponed and voice calls are blocked while GPRS is active (outgoing SMS is not affected)
- remote GPRS connection is not possible while voice call is active

## LED signalization

### POWER

on - module is connected to 24V power supply  
blinking - unstable power supply or internal error  
off - no power supply

### GSM

green blinking, 1 per second - connecting  
green - registered to domestic GSM network, good signal quality  
yellow - registered to domestic GSM network, acceptable signal quality  
red - registered to domestic GSM network, poor signal quality  
Short off every 2 seconds (green, yellow or red) means roaming.  
red, 1 blink per second - not registered, no signal or no network  
red, 1 short blink in 2 seconds - SIM card not present  
red, 2 short blinks in 2 seconds - bad PIN  
red, 3 short blinks in 2 seconds - bad configuration file  
red, 4 short blinks in 2 seconds - credit low

### GPRS

green blinking, 1 per second - connecting  
yellow - GPRS connected, TCP/IP connecting  
green - GPRS and TCP/IP connected  
red - disconnected, will try to connect again  
off - no connection, GPRS disabled

## Versions and upgrade

Released versions:

hardware code	firmware version	cypro version	applicable demo
GSM-1 v1.1	v3.0	v2.5.6 and later	GsmIniDemo.cyp
GSM-1 v1.1	v3.1	v2.5.8 and later	GsmIniDemo.cyp GsmPlcDemo.cyp
GSM-1 v1.1	v3.2	v2.6.2 and later	GsmIniDemo.cyp GsmPlcDemo.cyp

As hardware for all versions is same, each module can be upgraded to latest firmware. Upgrade options are:

upgrade option	hardware needed	software needed	remark
upgrade over CAN bus	P-CAN adapter	IEX-loader v2.0.3	safe method
upgrade over Ethernet	CyBro v2.5.6 and later	IEX-loader v3.0.0b	experimental
send to service	-	-	small fee may be applied

Change history:

firmware version	deleted	added
v3.0	-	-
v3.1	incoming number visible directly	phone number and sms content accessible from plc program, check GsmPlcDemo
v3.2	-	option to reset module from plc program

Plc program written for v3.0 will work with v3.1 and v3.2, except if it use variables to directly display received phone number. In such case, program should be adjusted to method used in GsmPlcDemo.

Plc program written for v3.1 will work with v3.2 without any changes.

Backward compatibility is not retained.

We recommend using the latest software revision whenever possible.