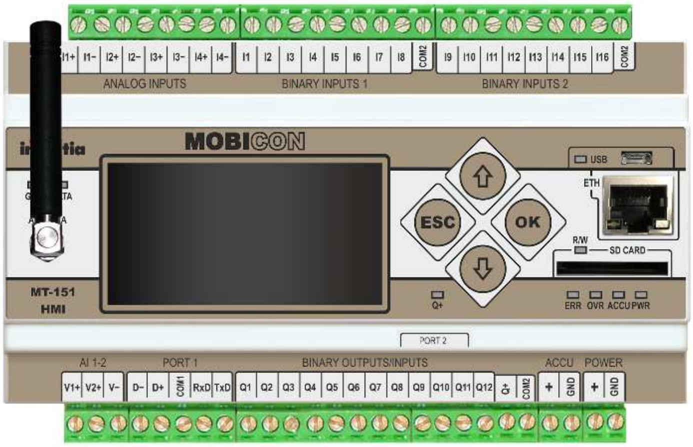
Professional Telemetry Module MOBICON MT-151 HMI version 2

User Manual



GSM/GPRS Telemetry Module for monitoring and control
Class 1 Telecommunications Terminal Equipment for
GSM 800/900/1800/1900 and UMTS 800/850/900/1900/2100

CE

INDEX

X Problem solving 108

1 Module destination

Telemetry Module **MT-151 HMI** is a device which incorporates functions of industrial PLC with integral graphical display, logger and protocol converter which enriching it with robust wireless GSM/GPRS connectivity. Thanks to access to two independent GSM/GPRS networks Dual-SIM technology used in this device ensures continuity of data flow not achieved in other solutions. Ethernet and two serial ports are powerful tools, allows communicating with other devices thus further expanding resources available to be used by user. With compact, robust design, integral GSM modem, attractive technical features and easy to use configuration tools the **MT-151 HMI** controller is an optimal solution for demanding wireless telemetry, control, diagnostic, surveillance and alarm systems.

General attributes of **MT-151 HMI**:

- Dual-SIM technology (possibility of using 2 SIM cards)
- Integral, quad-band 850/900/1800/1900 GSM modem
- 16 optoisolated binary inputs
- 12 optoisolated binary outputs with possibility of operation as binary inputs (selected independently for each output)
- 4 optoisolated 4 - 20mA analog inputs
- 2 optoisolated 0 -10V analog inputs
- Ethernet 10Base-T/100Base-TX port
- Optoisolated RS-232/485 serial port for communication with external devices
- Optoisolated RS-232 serial port with 5V power output dedicated for external control panels
- Graphical and textual OLED display, resolution 128x64 pixels, 6 lines 21 characters
- Diagnostic LEDs
- Connector for backup power (built-in battery charger)
- PLC functionality
- Standard industrial transmission protocols (Modbus ASCII/RTU, Modbus TCP, SNMP, Transparent, M-Bus, IEC,) support and routing
- 3-years warranty

2 GSM requirements

To proper operation, the module needs a SIM card supplied by a GSM operator providing GPRS and/or SMS services.

It is advised to use GPRS enabled SIM card with static IP addressing. The unique IP address of the SIM card is an identification for the module within the APN. This enables module-to-module and server-to-module communication within the APN structure. If SIM cards with dynamic addressing are used, only module-to-server communication is possible.

A good and strong GSM signal in the place where the module's antenna is located is necessary for the proper function of the module. Using the module in places where the signal is weak may lead to interruptions in transmission and possible loss of transmitted data along with increased costs generated by transmission retries.

3 How to use the manual

The manual was written for beginners as well as for advanced telemetry users. Each user will find useful information about:

Module design - this chapter presents the basic information about Module resources and design elements. Her is the information about how does the module work and how and where it may be employed. Chapter contains signaling description of LED indicators which is necessary knowledge during module installation.

Module connection diagrams - contains diagrams and procedures for connecting **MT-151** with devices and external elements like sensors, antennas or the SIM card.

First start of the module - contains recommended first start procedure.

Configuration - this chapter presents information about all available configuration parameters. All parameters concern firmware version compliant with documentation version.

Problem solving - all procedures for diagnostic operations.

Technical parameters - a revue of technical parameters and technical drawings.

Safety informations - information concerning conditions of secure use of the module.

Appendices - contain a register of changes in consecutive firmware versions, syntax of SMS messages and the memory map of the module, necessary for proper configuration of MTDataProvider and data collecting equipment.

4 Required programs

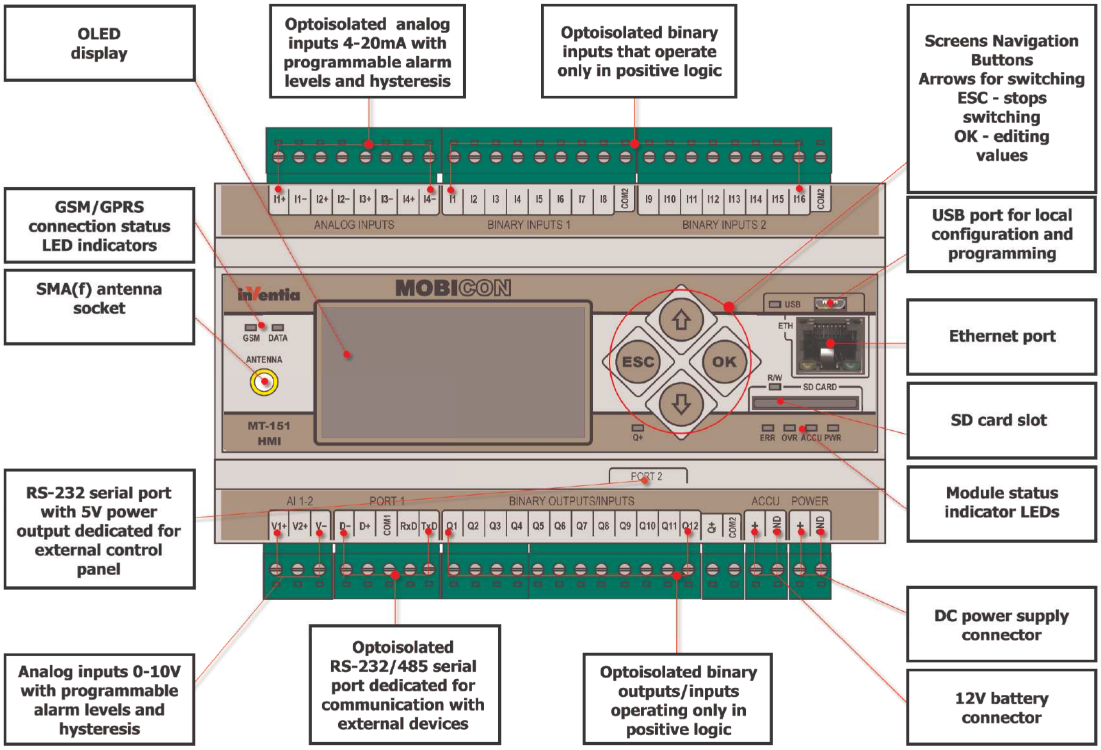
For proper start of working with the module has to be install an additional software for configuration and communication with **MT-151**. In this manual we are often use a software names that is cooperating with module. Below is available the short description of the applications with information what is necessary to install on the computer on every stage work with a device.

MTManager (MTM) (absolutely required for module setup and diagnostic) - main application for MT\ML module management. Program allows a local and remote configuration of the modules, programing of control algorithms, firmwares upgrade and resources monitor. Application is available on MT-DISC, the DVD that is delivered with **MT-151** or can be downloaded after login on www.inventia.pl web site.

MTData Provider (MTDP) (required for data transmission setup and diagnostic) - application for measurement data transmission in both ways, from and to the MT\ML modules. MT-Data Provider receive data from modules and share it in CSV files and write it to relational data base. The application supports OPC server functionality in two standards: DA\UA.

MTSpooler (MTS) (not required for module setup or diagnostic) - service that is designed for mass management of the MT modules configuration, especially for battery modules that working in sleep mode by most time is naturally behavior (the GSM modem is not logged to network). **MTSpooler** listens of module notification and after its receiving achieves tasks that was planned before.

5 Module design


OLED display
Optoisolated analog inputs 4-20mA with programmable alarm levels and hysteresis
Optoisolated binary inputs that operate only in positive logic
Screens Navigation Buttons Arrows for switching ESC - stops switching OK - editing values
GSM/GPRS connection status LED indicators
SMA(f) antenna socket
USB port for local configuration and programming
Ethernet port
SD card slot
Module status indicator LEDs
RS-232 serial port with 5V power output dedicated for external control panel
DC power supply connector
12V battery connector
Analog inputs 0-10V with programmable alarm levels and hysteresis
Optoisolated RS-232/485 serial port dedicated for communication with external devices
Optoisolated binary outputs/inputs operating only in positive logic
I1+
I1-
I2+
I2-
I3+
I3-
I4+
I4-
I1
I2
I3
I4
I5
I6
I7
I8
COM2
I9
I10
I11
I12
I13
I14
I15
I16
COM2
ANALOG INPUTS
BINARY INPUTS 1
BINARY INPUTS 2
inVentia
MOBICON
GSM
DATA
ANTENNA
MT-151 HMI
ESC
OK
USB
ETH
R/W
SD CARD
Q+
ERR
OVR
ACCU
PWR
PORT 2
AI 1-2
PORT 1
BINARY OUTPUTS/INPUTS
ACCU
POWER
V1+
V2+
V-
D-
D+
COM1
RxD
TxD
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
Q9
Q10
Q11
Q12
Q+
COM2
+
GND
+
GND

5.1 Hardware resources

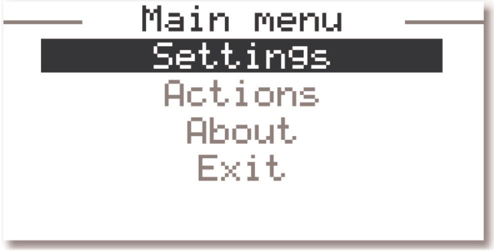
I1 - I16	binary inputs
Q1 -Q12	binary outputs that can operate as binary inputs
AI1 - AI4	4-20mA current analog inputs
AV1 - AV2	0-10V voltage analog inputs
PORT 1	RS-232/485 serial port
PORT 2	RS-232 serial port with 5V power output
Ethernet port	Ethernet 10Base-T/100Base-TX port
USB port	USB port used for configuration, programming and diagnostics
SD card reader	Allows to install SD memory card (max. 32GB FAT32)

5.1.1 Graphical display

Graphical OLED display and four navigation buttons placed on front panel of **MT-151 HMI** allow obtaining simple data operation panel. Diagnostic information and User defined screens are presented on the module display panel, up to 16 textual screens and 4 charts with auto data calibration. User can use the buttons to navigate between screens, stop switching the screens or allow editing values of the variables. Graphical display has got own menu for a basic functions management.

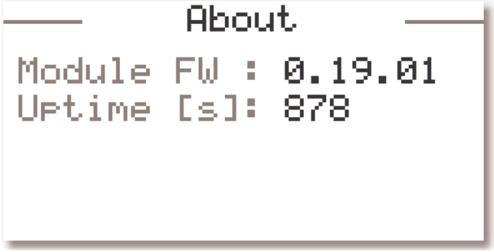
5.1.1.1 Display menu

Display menu is shown after pressing **OK** button for **more than 3 seconds**. Arrow keys allows to navigate between options, OK button selects next level, ESC backs to previous level. Exit is realized after select the proper option or automatically after 60 seconds without any action.



- Main menu options

Menu contains options:
Settings - in current version there is no option in these submenu
Actions - available options
Remove SD card - safety removing memory card
Passcode - entering the password for editing an internal registers from screen level. Security activation and enetering the individual pass code is managed on module configuration using **MTManager**.
About - shown current firmware version and uptime from power on.



Module FW : 0.19.01
Uptime [s]: 878

- Basic information about device

Exit - close menu

Detailed configuration of the display with all parameters description is available on chapter Display

5.1.2 Binary inputs

MT-151 HMI telemetry module is equipped with 16 optoisolated binary inputs marked as **I1 - I16** which operate in both, positive and negative logic. Binary inputs are divided into two groups (I1 - I8 and I9 - I16) with separate ground connector for each group. Additionally inputs marked as **I1 - I4** can operate in pulse mode with flow scaling function. The maximal frequency that can be measure is 2kHz.
The binary outputs **Q1 - Q12** can be individually configured to operate as binary inputs, however they support only more common positive logic.

5.1.3 Binary outputs

MT-151 HMI telemetry module is equipped with 12 optoisolated binary outputs marked as **Q1 - Q12**. Outputs state can be controlled locally by user written program or remotelly via GPRS, SMS or using one of available communication ports. The binary outputs **Q1 - Q12** can be individually configured to operate as binary inputs supporting positive logic.

5.1.4 Analog inputs 4-20mA

MT-151 HMI telemetry module is equipped with 4 optoisolated analog inputs operating in 4-20mA range and marked as **AI1 - AI4**. Analog inputs measurement can be scaled in engineering units (e.g. in meters or percents). For each analog input user can define 4 alarm levels (2 high and 2 low) with hysteresis. It is also possible to define deadband parameter - each time measured value changes by value higher than defined by this parameter deadband bit assigned to this analog inputs rises for one program cycle. These bits can be used for tracking analog input value.

5.1.5 Analog inputs 0-10V

MT-151 HMI telemetry module is equipped with 2 analog inputs operating in 0-10V range and marked as **AV1** and **AV2**. Analog inputs measurement can be scaled in engineering units (e.g. in meters or percents). For each analog input user can define 4 alarm levels (2 high and 2 low) with hysteresis. It is also possible to define deadband parameter - each time measured value changes by value higher than defined by this parameter deadband bit assigned to this analog inputs rises for one program cycle. These bits can be used for tracking analog input value.

5.1.6 Serial ports

MT-151 HMI telemetry module is equipped with 2 serial ports:

- **PORT 1** - optoisolated **RS-232/485** port designed for communication with external devices. Interface type and operating parameters are configurable by MTManager - program suite delivered for free with modules.
- **PORT 2** - **RS-232** port with 5V power output dedicated for communication with control panels and other devices.

5.1.7 Ethernet port

MT-151 HMI telemetry module is equipped with Ethernet 10Base-T/100Base-TX port used for communication with external devices.

5.1.8 USB port

MT-151 HMI telemetry module is equipped with USB (micro USB B) port which is used for device configuration (MTManager is required). This Port is visible in system (driver only for Windows) as COM port. Device communicates at 115200 bps with 8 data bits, no parity bit and 1 stop bit. **Port is not optoisolated!**

5.1.9 SD card reader

MT-151 HMI telemetry module is equipped with SD card reader supporting up to 32GB SD cards. Card should be formatted using FAT32 file system. User can store files with data from internal data logger on the memory card.

5.1.10 Real time clock

MT-151 HMI module is equipped with Real Time Clock (**RTC**). This clock is a source for time measurement for the module timers and time stamping measurements stored in the logger and sent via GPRS.
It is recommended to manually synchronize modules real time clock during the first configuration performed using the **MTManager** program.

NOTICE!
The RTC clock module does not automatically adjust to Daylight Saving Time.
It is recommended to use UTC time to avoid loss of data during manual time adjustments.

NOTICE!
The RTC clock is powered from an internal battery and as long as it is operated, there is no need to set the time again after power-off.
Since the clock precision is not absolute, periodical time adjustment may be necessary.

5.2 Internal resources

5.2.1 Logger

MT-151 HMI telemetry module is equipped with Logger capable of storing up to 12000 records, which allows to store measurements done every 10s for 30 hours or 40 days if measurement was saved every 5 minutes. Records are saved asynchronously, meaning that the record writing is triggered by an event (defined by user in the Event table). The event may be e.g.: counting the time by the timer, GPRS logon, crossing one of defined alarm thresholds and other.
The logger records consists of up to 4 data blocks, 28 16-bit registers each. Each record in the logger has a time stamp of the module internal Real Time Clock (RTC).
The data written in the logger is transmitted to IP address assigned during configuration. Sending of the logger content is triggered by user defined events. Confirmation of reception marks records as sent. In case of overflowing logger the oldest records are overwritten. Data from the logger could be stored on the installed memory card SD as separated comma files and transmitted into secondary IP address assigned during configuration.

5.2.2 Registers

MT-151 HMI module provides access to measurements, and other data via 16-bit registers divided into two groups according to Modbus device model - Input Registers (read only) and Holding Registers (read/write access). Remote access to registers is possible via SMS, GPRS (Modbus TCP and Standard Inventia protocol), serial ports (Modbus RTU) and Ethernet port (Modbus TCP).
Internal registers start from address 1189 and Input register is zeroed after module restart (e.g. power off, module update).

Holding registers to address 1188 are nonvolatile. Registers from 1189 to 8191 addresses are reset at startup of the module.
Input registers are reset at startup.

There is possibility to access single bits of Input and Holding Registers - or address them as Binary Inputs for Input Registers and Binary Outputs for Holding Registers. Detailed description of addressing method is described in Memory map chapter in Appendices.
Full list of registers is available in Memory map chapter in Appendices.

5.2.3 Counters

MT-151 HMI is equipped with 16 general purpose, internal counters.Their purpose is to count pulses understood as binary state changes of any bit available in the memory map. Each counter has one incrementing and one decrementing input and assigned 32-

bit register holding the difference of counted pulses. Initial state of the counters may be defined by user activating MTManager menu item **Initial settings** (more info in **MTManager** manual).

5.2.4 Timers

MT-151 HMI module is equipped with 16 general purpose, programmable synchronous timers CT1 - CT16 synchronized with module RTC. Their function is counting constant user defined time intervals in range of 1 minute to 24 hours. The User may appoint months, days of month and days of week in which the timer is active. In addition, there are available 16 general purpose programmable asynchronous timers CK1 - CK16. These timers start counting when module is powered or reset and they are not synchronized with RTC clock.

5.2.5 MT2MT buffer

MT2MT buffer enables creation of system where MT modules may exchange information (Holding Registers) with each other without any relaying instance. Data transmission from one module to the other is carried out by sending from one device group of Holding Registers with data to second device which has turned on MT2MT buffer functionality and defined MT2MT buffer which includes register addresses send from sending device. Data send by sender is saved to registers within buffer with the same address. Each time when new data arrives MT2MT_x bit is set, where x is position of sender IP address on receiver authorized IP list.

5.2.6 Constant parameters

In **MT-151 HMI** module configuration user can define up to 128 constant parameters - 16-bit values in range from -32768 to 32767 that may be further used for control program parameterization. Values of constant parameters are nonvolatile.

5.2.7 System flags

MT-151 HMI provides system of various internal flags that inform about module status. Most important bits are:

- FS1_acu - when set to 1, then battery is connected to module
- FS1_q+ - when set to 1, then there is no power provided for binary outputs
- FS1_gprs - when set to 1, then module is successfully logged into GPRS network
- FS1_gsm - when set to 1, then module is successfully logged into GSM network
- FS1_OVERRUN - when set to 1, an exceed of the runtime the internal program

Full list of System flags is available in Memory map chapter in Appendices.

5.2.8 Control program

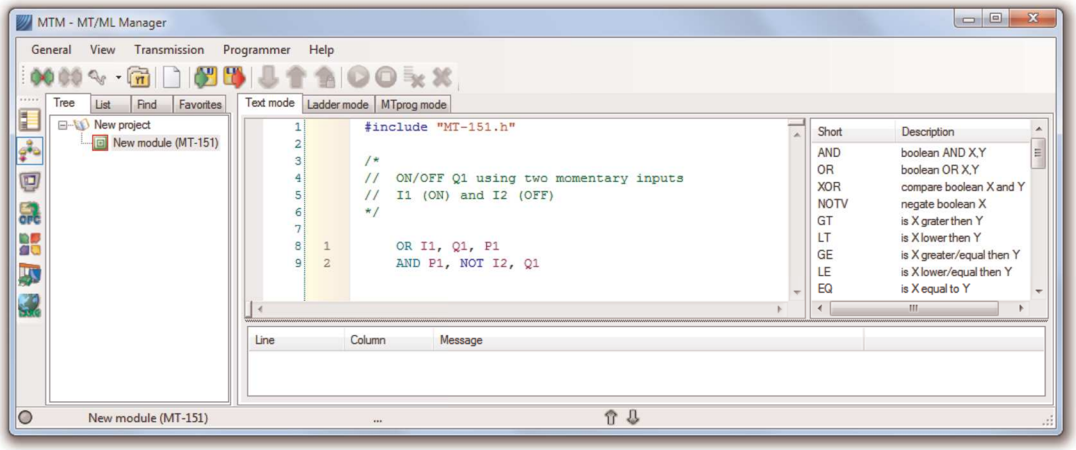
PLC functionality of **MT-151 HMI** module allows User to define algorithm of control and data processing in form of control program. Program can be written in three languages available in **MTManager**. The application is available with the modules and is free of charge.

Basic information about internal program running:

- The execution of program cycle starts every **100 ms**. Duration between cycles is count from beginning of first cycle to beginning of second. Run time length of the program not affect for next cycle until it exceeds 90ms. If program loop executes longer than **90 ms** (in case of write/erase flash memory of logger, large CPU processing load like lot calls of float function), next cycle will start immediately after ~10ms. At the same time flag FS1_OVERRUN (3 bit in register IREG2) will be set means time synchronization is lost.

- Maximum program length is 5000 instructions.
- Maximum instructions number executed in one cycle of the program is terminated to 10000.

Below a sample program is presented which executes start/stop algorithm with two momentary inputs (I1 - start, I2 - stop) and one output (Q1).

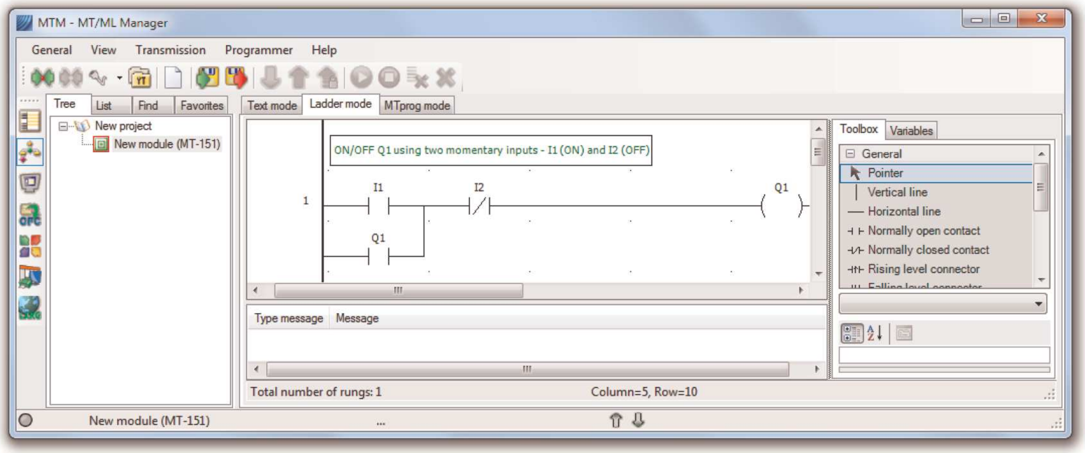


```
#include "MT-151.h"

/*
//  ON/OFF Q1 using two momentary inputs
//  I1 (ON) and I2 (OFF)
*/

    OR I1, Q1, P1
    AND P1, NOT I2, Q1
```

Short	Description
AND	boolean AND X,Y
OR	boolean OR X,Y
XOR	compare boolean X and Y
NOTV	negate boolean X
GT	is X grater then Y
LT	is X lower then Y
GE	is X greater/equal then Y
LE	is X lower/equal then Y
EQ	is X equal to Y

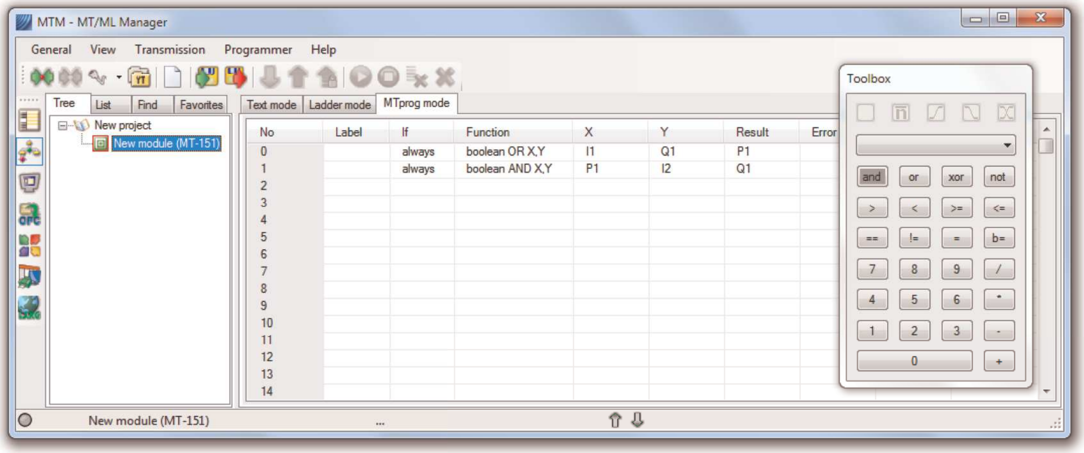
Line	Column	Message

MTManager - sample program in text mode



Total number of rungs: 1 Column=5, Row=10

MTManager - sample program in ladder



No	Label	If	Function	X	Y	Result	Error
0		always	boolean OR X,Y	I1	Q1	P1	
1		always	boolean AND X,Y	P1	I2	Q1	
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							

MTManager - sample program in MTprog mode

5.3 SIM cards slots

MT-151 HMI module is equipped with two SIM card slots that allow installing two miniature SIM cards (not micro!). For GPRS transmission it is advised to use static IP addressed SIM cards as it allows communication between devices and not only server and device. Module supports a low voltage 3.3V SIM cards.

5.4 Antenna

Connecting of the antenna is necessary for reliable data transmission from **MT-151 HMI** module. **SMA female** type antenna socket is placed on module front panel. Proper antenna placement is important during the module installation. In case of low GSM signal level using the directional antenna with high gain may be necessary.

5.5 Power supply

MT-151 HMI module can be powered from DC power supply providing voltage in range from **10.8 to 30 VDC**. In addition module supports using 12V SLA (Sealed Lead-Acid) battery as a backup power supply which provides power in case of loss the main one. Module has built in battery charger capable of servicing batteries with capacity up to 7Ah. Module starts automatic charging the battery if not fully charged. Module can be started only if main power supply is present.

NOTICE!
Exceeding the range of power supply may cause faulty operation or can damage the module!

5.6 Enclosure

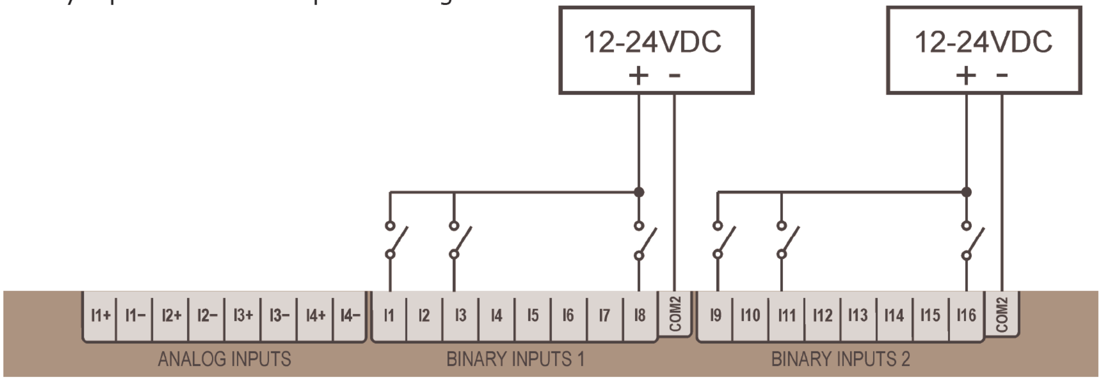
MT-151 HMI module is encapsulated in standard IP40 housing made of plastic compliant with safety requirements and protecting the module in standard operating environment.
The applied solution complies with standard industrial requirements for DIN rail mounting.

6 Connection diagrams

This chapter presents recommended wiring of external signals and installation procedur of the components.

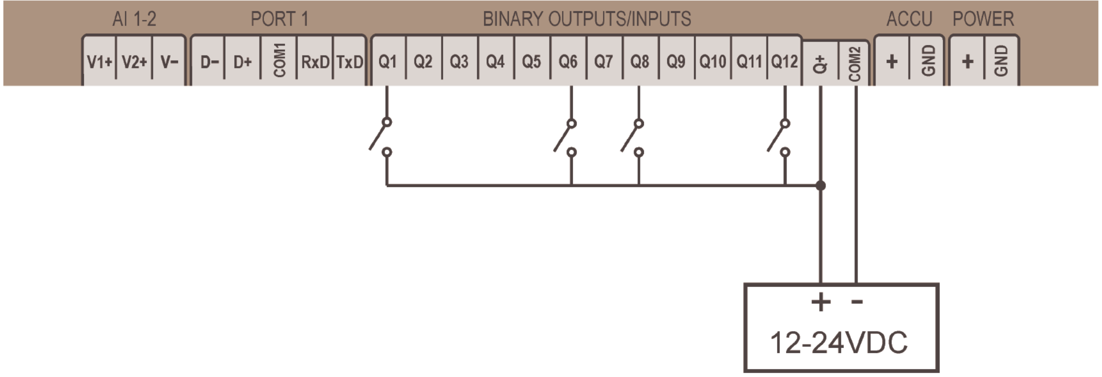
6.1 Binary inputs

Binary inputs **I1 - I16** in positive logic:



Binary inputs I1-I16 for proper operation, require connection the power to Q+ and COM2 terminal

Binary inputs **Q1 - Q12** in positive logic:



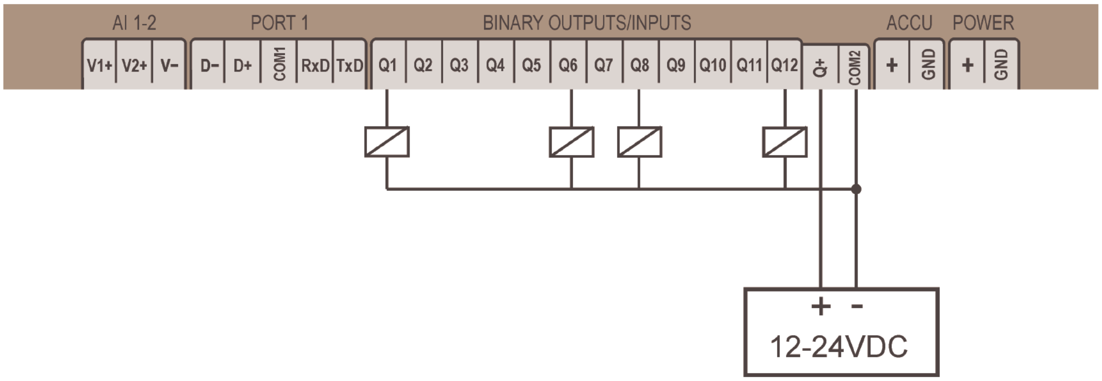
All binary inputs and outputs (**I1-I16**, **Q1-Q12**) from one group, galvanically isolated from rest of module and share common ground (COM2)

Attention!

- Power cables length should be < 10m
- Signal cables length should be < 30m
- For longer cables it is advised to use external overvoltage protection

6.2 Binary outputs

Binary outputs **Q1 - Q12** in positive logic:

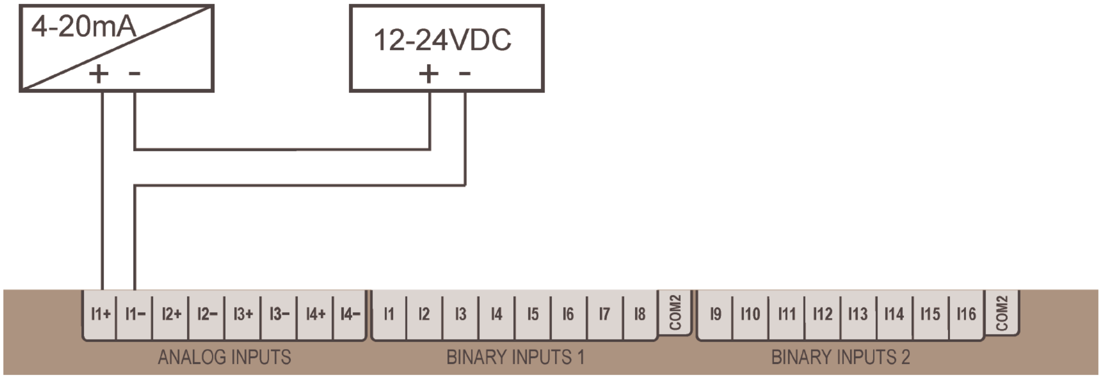


Attention!

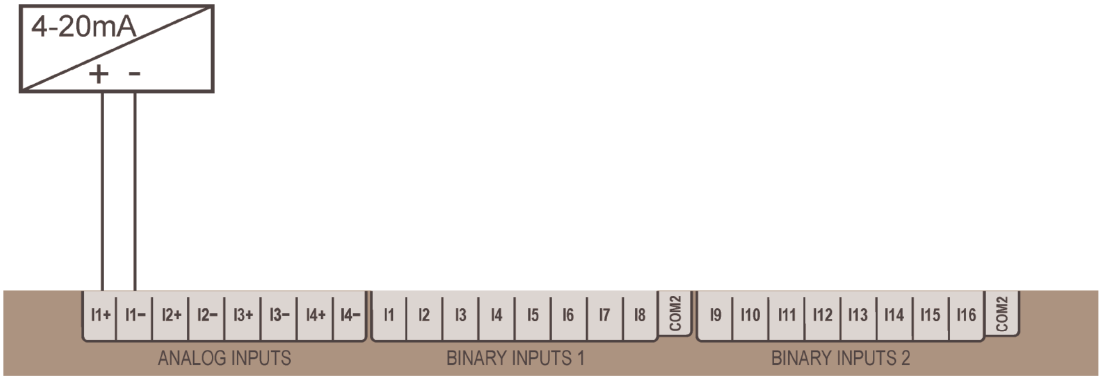
- Power cables length should be < 10m
- Signal cables length should be < 30m
- For longer cables it is advised to use external overvoltage protection

6.3 Analog inputs 4-20mA

Analog input **AI1** - connection with active sensor:



Analog input **AI1** - connection with passive sensor:

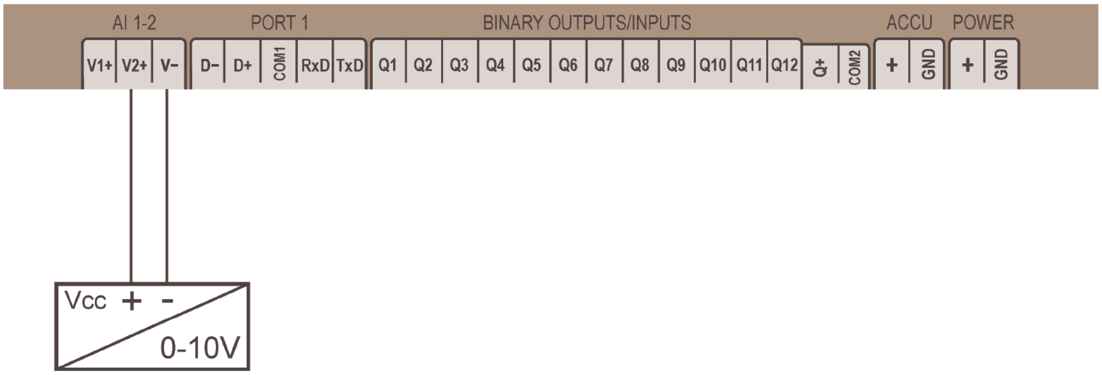


Attention!

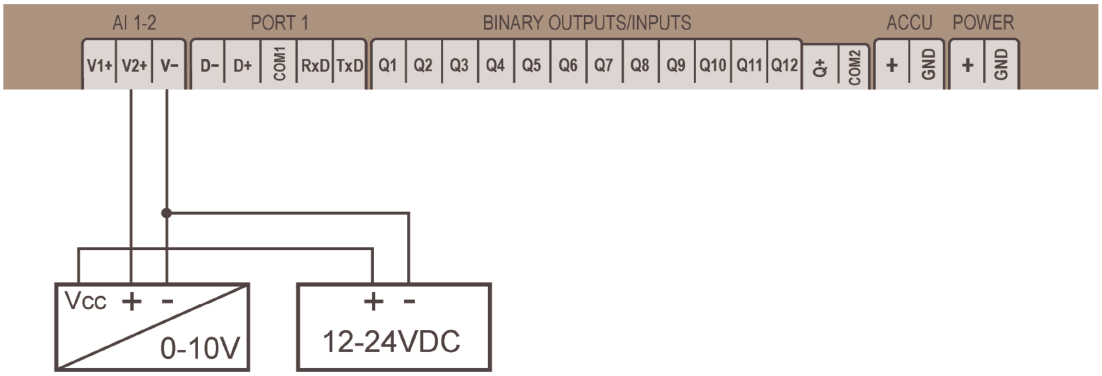
- Power cables length should be < 10m
- Signal cables length should be < 30m
- For longer cables it is advised to use external overvoltage protection

6.4 Analog inputs 0-10V

Analog input **AV1** - connection with active sensor:



Analog input **AI1** - connection with passive sensor:

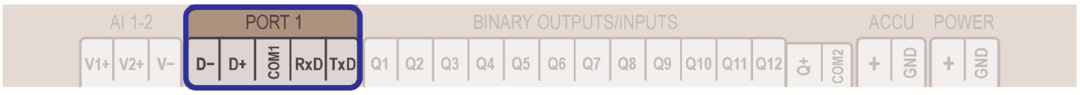


Attention!

- Power cables length should be < 10m
- Signal cables length should be < 30m
- For longer cables it is advised to use external overvoltage protection

6.5 Communication ports

Port 1 - optoisolated RS-232/485 serial communication port. Port connector is located on terminal block as pictured below.

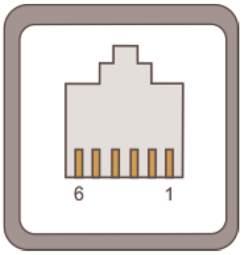


Connector name	Description
TXD	RS-232 - transceiver output
RXD	RS-232 - receiver input
COM1	Signal ground level for both modes
D+	RS-485 - transceiver output

D-	RS-485 - receiver input

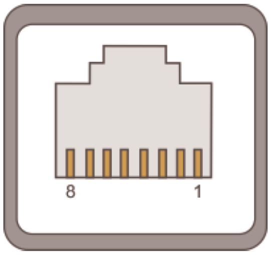
For operation in RS-232 mode cable length should not exceed 15m.

Port 2 - RS-232 serial communication port with 5V/500mA power output. No optoisolation. Port connector (RJ-12) is located on the front panel.



Pin	Description
1 - Vcc	+5V/500mA power output
2 – GND	Signal ground level
3 – TXD	Transceiver output
4 - RXD	Receiver input
5 – RTS	Handshake output (Ready To Send)
6 - CTS	Handshake input (Clear To Send)

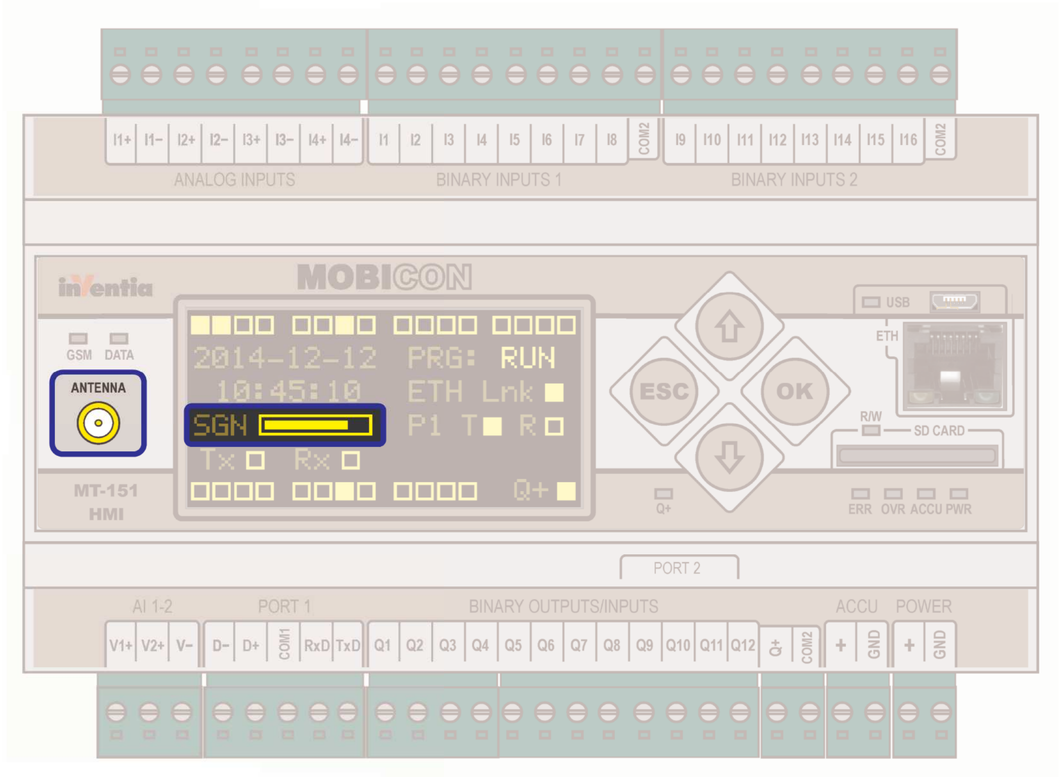
Ethernet port - optoisolated 10Base-T/100Base-TX communication port. Port connector (RJ-45) is located on the front panel.



Pin	Description
1 - TX+	Transceiver +
2 - TX-	Transceiver -
3 - RX+	Receiver +
4 - NC	Not connected
5 - NC	Not connected
6 - RX-	Receiver -
7 - NC	Not connected
8 - NC	Not connected

6.6 GSM antenna

Connecting the antenna is necessary for reliable data transmission from **MT-151 HMI** module. **SMA female** type antenna socket is placed on the module front panel.

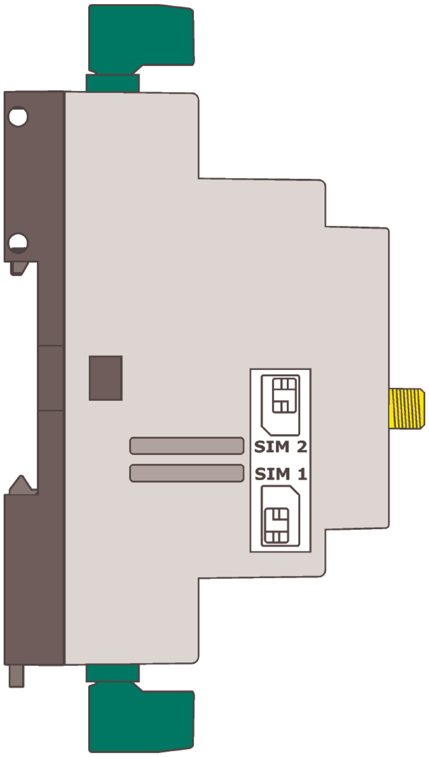


GSM signal strength LEds and antenna socket

Proper antenna placement is important during the module installation. In case of low GSM signal level using the directional antenna with high gain may be necessary.

6.7 SIM card installation

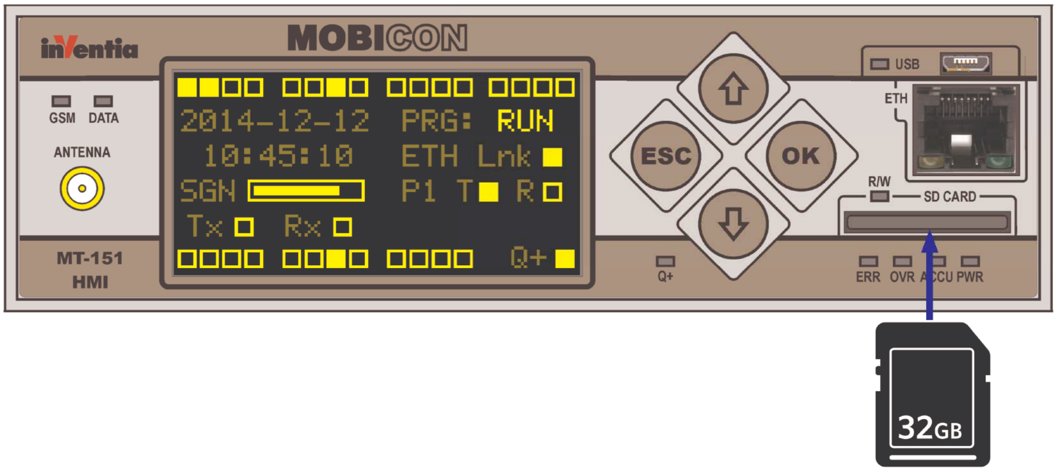
MT-151 HMI module is equipped with two SIM card slots that allow installing two miniature SIM cards (not micro!). For GPRS transmission it is advised to use static IP addressed SIM cards as it allows communication between devices and not only server and device. Module supports a low voltage 3.3V SIM cards.
It is advised to install SIM card after configuration of module to prevent module from writing wrong PIN to SIM.



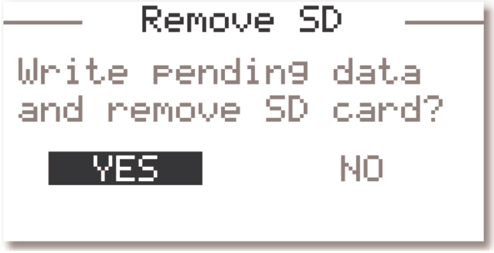
SIM Cards slots are on the left side of module's enclosure

6.8 SD memory card installation

SD memory card has to be slip in the gap narrow side facing toward the memory socket until a click. Correct installation and format of the memory card is signaling by **R/W** indicator. Light up of the diode can be delayed up to 30 second. Memory must be formatted in external reader.



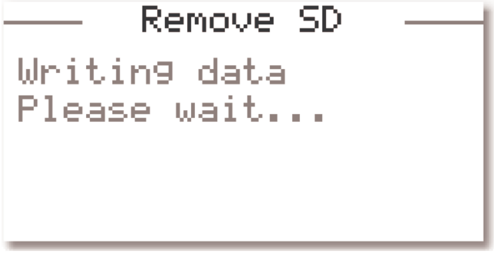
To extract memory card a narrow flattened tool like small screwdriver is required. Press OK button for 3 second, using arrow key, go to option **Actions** and select *Remove SD card* option and press OK, select Yes below question *Write pending data and remove SD card?* and confirm selection.



Remove SD

Write pending data
and remove SD card?

YES NO



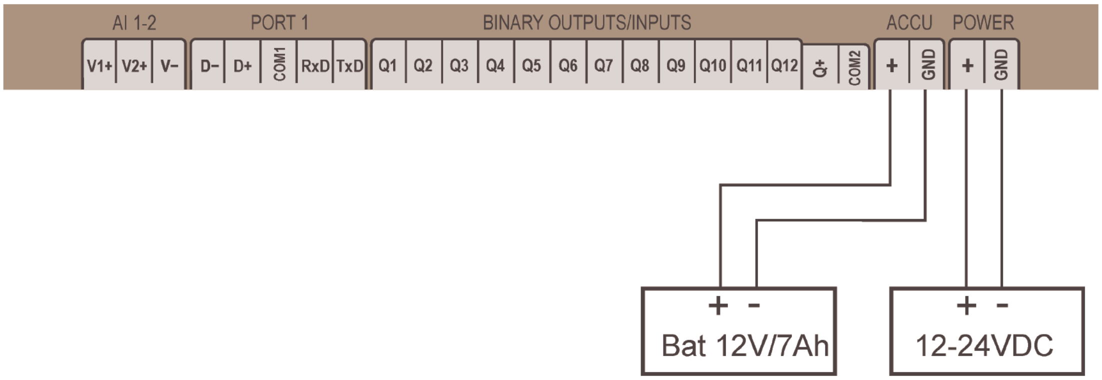
Remove SD

Writing data
Please wait...

If data has been written, **R/W** indicator is switching off for 60 second. Using a tool press gently the hidden card in socket until a click heard and release pressure. The card will eject and it's ready to intercept by fingers. If card won't be removed during 60 seconds will be switch on again.

6.9 Power supply

Power supply and 12V/7Ah battery (backup power)



Pin	Group	Description
+	POWER	Positive pole of mains power supply connector
GND		Negative pole of mains power supply connector
+	ACCU	Positive pole of backup battery connector
GND		Negative pole of backup battery connector

Module is equipped with charger to charging the external battery. Maximal current is limited to 400mA.

Attention!

- Power cables length should be < 10m
- Signal cables length should be < 30m
- For longer cables it is advised to use external overvoltage protection

NOTICE!
Due to high peak current of MT-151 HMI power supply should be able to deliver current >= 2A.
Improper power supply may results in faulty operation and can damage the module!

7 Starting the module

First start of the **MT-151 HMI** module requires a few simple activities. Please follow steps listed below:

1. **Connect signal wires and GSM antenna**

   Recommended connections diagrams for signal wires and the antenna are in **Connection diagrams** chapter**.**

2. **First configuration of the module**

   The scope of first configuration of **MT-151 HMI** is to enter parameters enabling login to GSM network and optionally GPRS network. A USB connection to the computer running **MTManager** program suite has to be established.
   Detailed information on how to install and use the **MTManager** program is attached in the installation DVD plate.

   In order to login to GSM/GPRS network the basic information about the SIM card and APN have to be provided to the module:

   In **GSM** group:

   Use of GPRS
   Yes - if using GPRS packet transmission is intended
   No - if the module is not going to use GPRS packet transmission
   Use of SMS
   Yes - if using SMS messaging is intended
   No - if the module is not going to use SMS messaging

   In **GSM/SIM1** group:

   SIM card name PIN number
   provide PIN code for SIM card that is going to be placed in the module (unless the card is set in pin-less mode).
   APN name
   provide APN name for GPRS transmission. Visible when *Use of GPRS* parameter is set to ***Yes***.
   APN user name
   provide user name (if required by the operator). Visible when *Use of GPRS* parameter is set to ***Yes***.
   APN password
   provide the password (if required by the operator). Visible when *Use of GPRS* parameter is set to ***Yes***.

   These parameters are the only parameters required to login to GSM/GPRS network. Bear in mind that the module with only the basic configuration does not have ability to send data. After checking the ability to login the full configuration of parameters has to be performed in order to use of intended extent.

3. **Inserting the SIM card**

   After downloading the first configuration disconnect the USB and power cables, insert the SIM card to SIM1 slot according to the instructions from

previous chapter and reconnect power cable. The module should login to the GSM/GPRS network.
The status of the module may be verified on main status screen at OLED display. More information in subchapter in Problem solving chapter.

Login sequence:

1. Module start
2. Verification of SIM card PIN code
3. Registration of modem in GSM network
4. Login to selected APN in GPRS network

Verify the configuration if any errors are indicated.

4. Setting the module time

The last, but very important element of modules startup is synchronizing the Real Time Clock of the module with the computer clock. It is crucial since lack of synchronization may result with faulty time stamping of the data in logger and may lead to data loss. More information about time synchronization is in MTManager user manual.

8 Interfaces and communication methods

The next chapters describes an each modes of operation the **MT-151 HMI** module that are available in case of communication port selection. Please read carefully description of every mode, it is necessary for proper choosing the right mode for best use of **MT-151 HMI** Next chapters describe a local and remote communication for data sharing between other devices on long distance.

8.1 Serial ports

PORT 1 is an optoisolated **RS-232/485** port designed for communication with external devices. Interface type and operating parameters are configurable using MTManager. Available modes:

- Transparent
- Modbus Master
- Modbus Slave
- Flex Serial
- M-Bus

PORT 2 is a **RS-232** serial port with 5V power output dedicated for communication with control panels and other devices. It is not optoisolated. This port allows to power on the external devices, available voltage 5VDC/500mA.
Available modes:

- Transparent
- Modbus Master
- Modbus Slave
- Flex Serial

8.1.1 Transparent mode

In this mode **MT-151 HMI** communication from serial port Port 1 is channeled to other communication port or GPRS network according to rules defined in Transparent routing table. This allows communicating two or more devices with the protocol that is not implemented in MT-151 HMI. Data received on serial port are transmitted to all targets that are set in routing table. Foreign Devices with specified addresses can be connected to other ports, Ethernet or 3G network.

Active transparent mode available receiving data in Modbus RTU type of commands. Module verified syntax and CRC of all frames and in case of detection Modbus command matching ID number with own ID. Equal ID execute the command and send an answer. The respond to that command allowing user to get data directly from device.

8.1.2 Modbus Master mode

In this mode **MT-151 HMI** can poll for data from and write data to external Slave devices connected to that port using Data blocks. Each Data block defines number of addresses in Slave device and matching registers in Holding Registers address space in module. Module polls for data with given interval and saves them in module. If new data is saved to those registers by SMS, GPRS transmission, control program or any other method but from Port 1, this new data will be written to Slave device. In this mode **MT-151 HMI** is using Modbus RTU functions 1, 2, 3 and 4 for polls and 5, 6 for writes and 16 for multiply write. For each data block there is **SLx_ok** bit, where x is

following number of data block, which informs about proper communication within this data block.
Also polls and writes from external devices communicating with MT-151 HMI can be routed to Port 1 according to rules defined in Modbus routing table. All those commands are automatically translated to Modbus RTU protocol.

8.1.3 Modbus Slave mode

In this mode **MT-151 HMI** module acts on this port as Modbus RTU Slave device and waits for incoming Modus RTU commands. It will react on command that is send to ID matching ID of **MT-151 HMI** for that port.
Incoming Modbus RTU commands can be routed to other ports or GPRS according to routing rules defined in Modbus routing table thus allowing to communicate devices connected to different ports. If command is routed to Ethernet port it is automatically translated into Modbus TCP protocol. If it is routed to GPRS then it is automatically translated to chosen in GPRS parameter section protocol.

8.1.4 Flex Serial

Flex Serial mode allow to create communication between external device that is connected to serial port directly form internal user program which runs on module. Access for internal resources is still available using other interfaces but on Port where Flex Serial is selected there is no possibility for sending and receiving data from other interfaces. Received data are identify and analyzed only in internal user program. Internal program is responsible for preparing question frames which will be send to external unit. Transmission is based on registers in internal space that are separated for two buffer for store the data. Sending and receiving data is independent. **Flex Serial** mode has got two dedicated configuration parameters: Max. data packet size i Data frame delimiter, describe maximal data bytes receiving and time of "silence" between successive packets, after which the data will be written to the receiving buffer. This parameters define rules of changing data flow receiving on port into data available on internal program.

Buffer structure
Buffer length is 256 registers 16 bits each but only little endian is used for communication. One 16 bits register store one byte. The big endian part is not in use.

Receiving data
Register PXRCV_NO controls data receiving. Register PXRCV_B1 … PXRCV_B512 collect raw data. These registers together creates Receiving buffer.
At first the received data are stored in internal buffer, size 6kB and then insert into Receiving buffer only when register PXRCV_NO value is 0. Along with data insert, module write into register PXRCV_NO length of package value. Internal user program have to process the data form receiving buffer when detect nonzero value on PXRCV_NO and when finish processing, write 0 value in PXRCV_NO and thus allow to enter another data packet. Received data are always inserted at the beginning of the buffer, first byte of packet goes into register PXRCV_B1. Additionally in register HR5257 at first bit module placed 1 when detect problem with receiving data like overload the buffer, like wrong parity or stop bits and other. PXRCV_ERR bit is not automatically reset. Internal user program have to manage of resetting the bit for proper communication diagnostics. State of this bit does not affect on operation of receiving data algorithm and inform user only about the problems with correct data receiving.

Data Receiving buffer resources

Name/Register address		Description
PORT 1	PORT 2	

P1RCV_NO	HR5000	P2RCV_NO	HR6000	Counter receiving data
P1RCV_B1... P1RCV_B256	HR5001- HR5256	P2RCV_B1... P2RCV_B256	HR6001- HR6256	Received data registers (little endian in use)
P1RCV_ERR	HR5257	P2RCV_ERR	HR6257	Receiver status register (b0 - data receiving error)

Sending data

Register PXSND_NO control sending operation. Sending buffer is composed from registers PXSND_B1 ... PXSND_B512. Writing non zero value into register PXSND_NO sends a package of the data from Sending buffer at length equal the value that was enter to PXSND_NO. Data is always send from the beginning of the buffer, first byte is taken from register PXSND_B1. Register PXSND_NO is automatically reset to zero value after reading data from Sending buffer. This means that next package to send can be prepared. Registers PXSND_BX can be modified only when PXSND_NO register is set to 0 value. Additional in holding registers space, mapped as binary output space is placed P2SND_ERR bit which is set to 1 when error is detected while sending data (e.g into PXSND_NO was enter value grates than 256). P2SND_ERR bit is not automatically reset. Internal user program have to manage of resetting the bit for proper communication diagnostics.

Data Sending buffer resources

Name/Register address				**Description**
PORT 1		**PORT 2**		
P1SND_NO	HR5500	P2SND_NO	HR6500	Counter sending data
P1SND_B1... P1SND_B256	HR5501- HR5756	P2SND_B1... P2SND_B256	HR6501- HR6756	Sending data registers (little endian in use)

8.1.5 M-Bus

M-Bus mode is an enlargement of **Modbus Master** mode with functionality called data mirroring that read data from external Slave device and store it in the internal register space of the module. The protocol is available only on port RS-232. This feature is developed in the module for directly operations with devices that support the M-Bus protocol which are used mainly in energy consumption measurements. Direct communication between **MT-151 HMI** module and **M-Bus** devices require use an external electrical M-Bus converter that have to be connected to PORT 1. Converter name is RM-120 and is available in additional accessories.

Implementation of the protocol in the module allows to choose how to address a device that are connected to **M-Bus** line. There are two options:

1. **Broadcast** - configuration of only one connected device without knowing its address. Module sends broadcast frames with 254 address.
2. **Unicast** - individual addressing each devices with selection that using real device address (range 1 ... 250) or logical address based on sent a addressing frame that contains serial number in address place. Module sends unicast frames with 253 address.

M-Bus mode in **MT-151 HMI** supports reading basic quantities like:

- Energy

- Volume
- Mass
- Flow
- Temperature
- Pressure
- Power
- Time
- some other values with special identification number that can be added manually to configuration of variables prepared for reading

Data reading

Data is reading according to configuration of each data blocks up to maximum 16. Each block can mapped up to 16 independent configuration variables. The reading of one block cannot be executed more often every 60 seconds. Processing of an information read from external devices, its interpretation or scaling is made according to units selected in configuration. The configuration of variables allow to choose number of final register and data format to store the information after reading. Each variable has got additional parameters as **Logical Unit**, **Tariff**, **Storage** and **Type**. This parameters allow to read specific measurements from connected device and records the data to module into appropriate registers. Variables assignment to register require the same values for all parameters **Value**, **Tariff**, **Storage** and **Type**. It is possibility for reading the same quantity for different Tariff and save this measurements in different registers. It is useful when device connected to the module sent different type of data that depends from working mode, example LQM-3 heat meter usually sends data as "Momentary" type but in case when pulses are not detect the type is change to "Error".

Communication diagnostic

M-Bus mode, like other protocols in module, has got flags and bits for communication diagnostic prepared in input registers space. State of these bits carry on information about proper communication with every connected device and all variables from each block. Flags are refreshed with every pool of the data, this is helpful to monitor state of communication with device. Example: two variables are define for the same quantity, one as "Momentary" type and second as "Error" type, emergency situation will be when flag for "Momentary" type goes to value 0 and in the same time flags 'Error" goes to 1. Device communication bits are located in register IREG350 (1 on bit X means correct data receiving from device X). Variables communication bits of each variable for all devices are located on registers from IREG351(DEVICE1) to IREG366(DEVICE16) (1 on bit X means correct receiving data from variable X) Register IREG370 stored real address device that is answered for Broadcast type of request.

This version of protocol implementation is prepared specially for proper works with heat meters, if need it can be enlargement, please contact us to discus details.

All parameters and its description required to run communication in **M-Bus** mode are explain in chapter Configuration.

8.2 Ethernet port

MT-151 HMI telemetry module is equipped with Ethernet 10Base-T/100Base-TX port used for communication with external devices. MT-151 HMI operates simultaneously as Master and Slave on this port.

Communication via this port is possible only with devices added to Authorized IP addresses list.

8.2.1 Modbus TCP Client

Modbus TCP Client functionality allows polling for data from and write data to external Modbus TCP Server devices connected to that port using Data blocks. Each Data block defines number of addresses in Server device and matches registers in Holding Registers address space in the module. Module polls for data with given interval and saves it in module. If new data is saved to those registers by SMS, GPRS transmission, control program or any other method, this new data will be written to Server.
In this mode **MT-151 HMI** is using Modbus TCP functions 1, 2, 3 and 4 for polls and 5, 6 for writes.
For each data block there is **TSLx_ok** bit, where x is following number of data block, which informs about proper communication within this data block.
Also polls and writes from external devices communicating with **MT-151 HMI** can be routed to Ethernet device according to rules defined in Modbus routing table. All those commands are automatically translated to Modbus RTU protocol.
Modbus TCP Client connects to servers using port 502.

8.2.2 Modbus TCP Server

MT-151 HMI operates as is server listening on port 502 and waits for Modbus TCP frames. It will react on command that are send to ID matching ID of **MT-151 HMI** for that port. Incoming Modbus TCP commands can be routed to other ports or GPRS according to routing rules defined in Modbus routing table thus allowing to communicate devices connected to different ports. If command is routed to serial port it is automatically translated into Modbus RTU protocol. If it is routed to GPRS then it is automatically translated to chosen parameter section protocol in GPRS.

8.3 Remote communication

MT-151 HMI module is equipped with build-in GSM/GPRS modem, which allows device to send and receive SMS messages and transmit data using GPRS network. In sent SMS messages special mnemonics may be used, which are dynamically changed according to value changes in the module memory. Same mnemonics in received messages can be used for polling and/or writing data to module. More details about SMS commands can be found in SNCS Simple Name Commands syntax chapter in Appendices.
GPRS data transmission allows communicating device with remote server or other device accessible from APN assigned with used SIM card.

MT-151 HMI is using two protocols for GPRS communication:

- *MT Standard* - module communicates using the protocol and transmission protection created by Inventia. This data frame is supported by all software tools provided with module.
- *UDP Standard* - data is send in form of Modbus RTU command encapsulated in standard UDP data frame. Data reception control is not available when using that data frame format. Detailed description of UDP Standard communication is available upon request from Inventia technical support team.

8.3.1 Dual-SIM

A slot for Two SIM cards allow installing in **MT-151 HMI** cards from different providers. Second transmission channel is used only when primary fails. One GSM connection is supported by the modem in one time. Dual SIM function activate automatically when a both SIM cards are selected in configuration.

Dual SIM - logon sequence

1. SIM card in SIM1 slot is used to login attempt.
2. Transmission retries are repeated with transmission timeout.

3. When all retries failed, module is waiting according to wait time after disconnection
4. SIM card in SIM2 slot is used to login attempt.
5. Transmission retries are repeated with transmission timeout.
6. When all retries failed, module is waiting according to wait time after disconnection
7. Cycle is looped until correct login

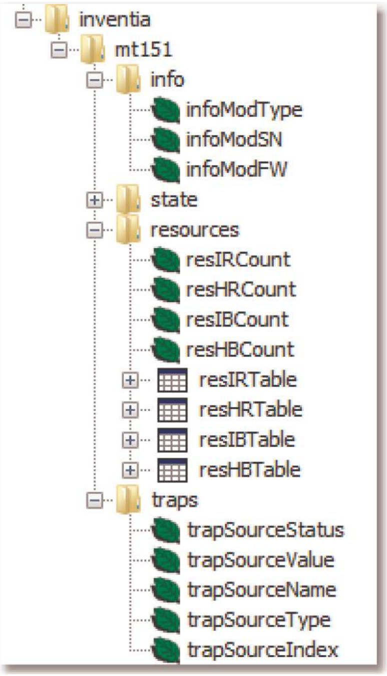
Successful login resets the module counter failed login attempts.

8.4 SNMP protocol

MT-151 HMI supports SNMP protocol version 1. Module operates as a SNMP agent – device which can be polled by server and can send unsolicited information (traps) to server. Data transmission is realized by exchange values of device variables (numbers, text). Variables are organized in form of tree. Each variable has unique OID (Object ID) identification number which plays a role of variable address. Next to standard variables referring to device interface description and network status producer can add branch with own parameters. Such branch should have unique ID number assigned to producer by IANA organization.

Inventia is using ID **42317**.

Variable tree structure is defined as a MIB data base and can be saved in text files using ASN1 (Abstract Syntax Notation One) notation. Variable tree MT-151 HMI has structure as below:



Sending unsolicited data (Traps)

Trap is a data packet send from device containing device ID, device IP address, timestamp and Specific ID (trap ID). To basic data described above device can add additional data from variable tree. Module adds to trap following variables in order as follows: **trapSourceStatus**, **trapSourceValue**, **trapSourceName**, **trapSourceType**, **trapSourceIndex**. Content of these variables can be set up in module configuration or can be copied from defined registers, from holding registers address space, which allows to dynamically assigning data to trap using user program.

While configuring a trap user defines triggering source of trap, Specific ID and values of variables added to trap. Server receiving trap after analyzing variables values (especially Specific ID) can connect trap with its description and present data in propels form. Since basic trap types as well as meaning and ranges of variables are predefined in MIB file it is essential that user when configuring traps used proper variable values as makes server configuration much easier.

Generating queries (Requests)

Request is a data packet sent to network device with query of the specific OID variable. **MT-151 HMI** has got possibility to generate 32 variable requests for 16 variable receivers. Control of Request sending is managed from internal program. Request response has to be in numeric format.

All SNMP configuration parameters are grouped in SNMP subgroups within Communication group in **MT-151 HMI** configuration.

8.5 IEC 60870-5-104 protocol

MT-151 HMI module is able to act as **IEC 60870-5-104 (IEC)** Slave device (Server). One TCP client can be service at one connection time. Module is listening on **2404** port. Available functions are shown in Interoperability Table. **IEC** protocol is adapted to be used in device which contain standard **Modbus** register spaces. Available data types are constrained to single point (bit value), 16 bits registers with sign (short scaled) and values with floating comma(short float).

More than 8000 registers in module is mapped on four variables blocks formatted just like **IEC** tables with **IEC** variables. Module provide access to blocks of variables, numbered subsequently and give access to subsequent registers or bits. This method allow user to access any of **MT-151 HMI** registers by executing function "read variable" (C_RD_NA_1) with correct **IOA** address. Module is responding with value fetched from resulting Modbus register/bit.

Internal mapping variable addresses between IEC and Modbus space

IEC variable (IOA)	**IEC typ danych**	**Modbus space**	**Description**
5001 **...** 9801	M_SP_NA_1	IB0 **...** IB5799	Discrete inputs (IREG bits)
10001 **...** 10300	M_ME_NB_1	IR0 **...** IR299	Input registers
20001 **...** 36384	M_SP_NA_1	HB0 **...** HB16383	Coils (HREG bits) from first 1024 registers
40001 **...** 48192	M_ME_NB_1	HR0 **...** HR8191	Holding registers

Additionally Module's configuration allows user to explicitly up to 200 variables on **IOA** range from 1 to 1000 address. For each variable user can set type and choose address or bit from internal memory map and allow access to them from **IEC** side. Each variable can own unique name, marked with a time stamp and signed as variable that is delivered on global request (interrogation). Global request is performed by command (C_IC_NA_1) with additional parameter that determine distinction for general request or group request. Eight group of parameters is ready to assignment the variables. Group request required enter an identify number for specific group 1 … 8.

Configuration of **IEC** protocol in module allows to send spontaneous sending of data after making a definition in Events table. There is maximum 32 events, each of them has got own trigger to set and data to send. Events can sent single data and whole groups. Triggering of the events can be controlled from internal user program using flags **P1** - **P256** as a trigger source. **IEC** events are not depending from standard GPRS events that are distributed by module.

IEC protocol provides time synchronization between Client and Server. This function is disabled in default. Changing the setting of parameter in configuration allows to turn on the IEC time synchronization if necessary.

First connection of the client to **MT-151 HMI** after reset makes a reply by sending the "end of initialization" (M_EI_NA_1) message.

Module configuration include some parameters responsible for communication efficiency. There are timeouts parameter called just like **IEC** specification: **T1**, **T2**, **T3** and parameters **K** and **W.** Description is available below in Configuration chapter.

Communication handling (events and data sending) consist on Current sending buffer with capacity of 250 records. Data in buffer are stored until the confirmation from Client comes. This means that global requests with more than 250 variables in seconds time periods cannot be executed correct because could exceeding a capacity. For increase speed of sending confirmations by Clients and release this way a occupied buffer record, we recommend to set on Client site timeout **T2** between **3 ... 5** seconds and **W** parameter to **4** or **5** value. Events prepared in configuration use Current sending buffer but they are stored in non-volatile memory and can be sends with delay in case of temporary exceeding a capacity.

9 Configuration

MT-151 HMI just like other MT devices is configured by using **MTManager** (MTM) which is provided for free with all MT equipment.

MTManager is an unified program environment that allows setting up and maintaining whole telemetric system or systems regardless of its scale. Possibility of arranging devices in groups of projects or putting them in folders makes effective managing of telemetry system easy.

All described on next pages parameters are available in MTManager configuration module after adding **MT-151 HMI** to project. Detailed description of MTManager functionality is provided in MTManager User Manual.

NOTICE!
Availability of different functions and parameters depends on module firmware version and the settings of parameters they may be dependent on.

For clarity and ease of use module configuration parameters of **MT-151 HMI** were divided into logically or functionally connected groups in the following order:

- Header group - contains unchanged parameters describing the module, its firmware and configuration.
- General group - contains basic configuration parameters.
- GSM group - contains parameters responsible for GSM/GPRS networking.
- Resources group - defines parameters of hardware and software resources related with measurements.
- Communication ports group - contains parameters controlling both local and remote communication using serial and Ethernet as well as GPRS transactions. It is possible to set up routing rules for each port allowing to automatically passing data between communication ports.
- Communication group - contains lists of transmission tasks to be carried out upon occurrence of activating criteria.

There is also possibility to set up initial values for some of module resources (like counters) using MTManager Presets tool.

9.1 Header

The **header** contains basic information describing the module, along configuration with version number and version of configuration file stored by the program. Information displayed is for information and verification purposes only and thus not available for user configuration.

9.1.1 Module name

Function	-	Displays name assigned to module during configuration
Data type	-	Text
Range	-	N/A, read-only parameter
Comments	-	N/A

9.1.2 Module type

Function - Displays the type of configured telemetry module

Data type - Text

Range - N/A, read-only parameter

Comments - N/A

9.1.3 Module serial number

Function - Displays serial number of telemetry module

Data type - Text

Range - N/A, Read-only parameter

Comments - This field displays serial number assigned to module during manufacturing. This number is static and unique identifier of the unit.

9.1.4 Modem firmware version

Function - Displays modem firmware version

Data type - Text

Range - N/A, read-only parameter

Comments - N/A

9.1.5 IMEI number

Function - Displays GSM modem IMEI number

Data type - Text

Range - N/A, read-only parameter

Comments - N/A

9.1.6 Firmware version

Function - Displays module firmware version

Data type - Text

Range - N/A, read-only parameter

Comments - N/A

9.1.7 Configuration file version

Function - Displays version identification of configuration file used for actual configuration

Data type - Text

Range - N/A, read-only parameter

Comments - Value depends on module firmware version. Auxiliary extension character defines the sub-version

9.1.8 Configuration identifier

Function - Displays identification number of current configuration

Data type - Hexadecimal number

Range - N/A, read-only parameter

Comments - The value of this parameter increases automatically by 1 after each successfully written configuration.

9.1.9 Last configuration date

Function - Displays date and time of last successful configuration change

Data type - Text

Range - N/A, read-only parameter

Comments - The value changes automatically after each successful configuration change. It is useful for tracing unauthorized configuration changes.

9.1.10 Last reading time

Function - Displays internal module time recorded during last configuration reading or during last time setting

Data type - Text

Range - N/A, read-only parameter

Comments - This field is useful in verifying last access time and checking internal module clock (RTC) settings

9.2 General

General group contains basic configuration and configuration protection parameters.

9.2.1 Device identifier

Function - Selects device identifier used which is added to data frames sent by device and then to identify sender by server software (e.g. MTDataProvider)

Data type - Selection list

Range - ***IP address***

IP address assigned to device by GSM provider is used as identifier. Advantage of the solution is possibility of changing device on site to other of same type without need to reconfigure server. SIM card used with device should have static IP address.

Serial number

Serial number of device is used as identifier. Advantage of this solution is a possibility of operation in APN with dynamic IP addressing.

Default value - ***IP address***

Comments - N/A

9.2.2 Module IP

Function - Displays IP address assigned to module by GSM provider during last communication with module. It is used for remote configuration via GPRS.

Data type - IP address

Range - ***0.0.0.0 - 255.255.255.255***

Default value - ***0.0.0.0***

Comments - When this field is left at default value 0.0.0.0 remote communication with the module is impossible. IP address can be inserted manually to allow access to remote module via GPRS. If you use feature of dual SIM card you should to be sure which SIM card has been used to communicate just right now.

9.2.3 Configuration password

Function - Defines the password protecting access to configuration of the module. The password will be required for both local and remote access, thus protecting against unauthorized configuration alterations.

Data type - Text

Range - Letters and numbers, max. 32 characters

Default value - N/A

Comments - Since the only way of unlocking the module is resetting it to factory settings, it is vital that the password is stored in a safe way and available when needed.

9.2.4 Configuration read disable

Function - Blocks reading of module configuration even while using valid password

Data type - Selection list

Range - ***Yes***

Reading of configuration from the module is impossible.

No

Module is not protected against reading of configuration.

Default value - ***No***

Comments - This parameter has no influence on uploading a new full configuration but prevents writing changes if configuration identifier in the module and in MTManager do not match

9.2.5 Error display time

Function - Defines (in seconds) time of displaying error code on

Data type - Number

Range - ***1 - 250 [s]***

Default value - ***30 [s]***

Comments - setting of too small value makes error code identification difficult while too long value extends the time span before module attempt to fix the problem.

9.2.6 UDP data frame format

Function - This parameter selects data frame type used by module for GPRS communication

Data type - Selection list

Range - ***MT Standard***

Module communicates using the protocol and transmission protection created by Inventia. This data frame is supported by all software tools provided with module.

UDP Standard

Data is sent in form of Modbus RTU command encapsulated in standard UDP data frame. Data reception control is not available while using that data frame format.

Default value - ***MT Standard***

Comments - Detailed description of UDP Standard communication is available upon request from Inventia technical support team.

9.2.7 GPRS transmission retries number

Function - Defines number of attempts to send data through GPRS network if the reply to original transmission does not arrive in a timely manner specified by Transmission timeout parameter.

Data type - Number

Range - ***0 - 9***

Default value - ***2***

Comments - Setting the value to 0 results in sending data without waiting for reception confirmation.
In normal conditions the value should not exceed 3. This prevents loss of transmitted data without blocking of subsequent rules processing. Bear in mind that subsequent data will be sent after reception of confirmation for reception of previous frame.

9.2.8 Transmission timeout

Function - Defines the wait time for reception confirmation of sent data frame.

Data type - Number

Range - ***1 - 60 [s]***

Default value - ***8 [s]***

Comments - The value of this parameter along with GPRS transmission retries number influence on maximum time of data frame sending. For default values the time is (2 + 1) * 8 = 24s. After that time module drops data frame from queue.

9.3 GSM

GSM group contains parameters responsible for GSM/GPRS networking. Proper configuration of those parameters is essential for successful GSM and GPRS communication.

9.3.1 Number of SIM cards

Function - Defines number of SIM cards used by device. There are two slots for SIM cards - SIM1 (upper slot) and SIM2 (lower slot)

Data type - Selection list

Range - ***1***

Only SIM1 slot is used by device

2

Both slots are used by device, Dual SIM feature is active.

Default value - ***1***

Comments - N/A

9.3.2 Use of GPRS

Function - Enables GPRS communication

Data type - Selection list

Range - ***Yes***

GPRS communication is allowed

No

GPRS communication is disabled

Default value - ***Yes***

Comments - If set to ***Yes*** allows user to configure parameters essential for setting up GPRS communication. When set to ***No*** module will make no attempt to log into GPRS network.
If both GPRS and SMS are not used module disables all modem functionality.

9.3.3 Use of SMS

Function - Enables SMS communication

Data type - Selection list

Range - ***Yes***

SMS communication is allowed

No

SMS communication is disabled

Default value - ***Yes***

Comments - If set to ***Yes*** allows module to both receive and send SMS to Authorized phone numbers. When set to ***No*** module will not send not service received SMS messages. All received SMS will be deleted.
If both GPRS and SMS are not used module disables all modem functionality.

9.3.4 SIM1

SIM1 group contains parameters responsible for establishing GSM/GPRS communication using SIM card inserted into SIM1 slot (the upper one).

9.3.4.1 Address IP

Function - Displays IP address assigned to SIM card placed in SIM holder slot number 1 using if the communication with module has been established earlier on that slot. It can be used for remote configuration via GPRS.

Data type - IP address

Range - ***0.0.0.0 - 255.255.255.255***

Default value - ***0.0.0.0***

Comments - When this field is left at default value 0.0.0.0 remote communication with the module is possible using other IP addresses. Obviously IP address can be inserted manually to allow access to remote module via that SIM card if is logged.

9.3.4.2 SIM card PIN number

Function - Defines PIN access code for SIM module delivered by GSM operator. For SIM modules not protected by PIN code, the value is insignificant.

Data type - Text

Range - Numerals, max 8 characters

Default value - N/A

Comments - Wrong PIN can cause SIM card lock

NOTICE!
Caution is vital, when setting the PIN code value. Entering incorrect PIN code may cause modules start-up impossible and lock SIM card. For security reasons module makes attempt to enter PIN code twice.

To unlock SIM card please follow procedure described in Problem solving chapter.

9.3.4.3 APN name

Function - Defines APN name which is used by module to carry out GPRS transmission using that SIM

Data type - Text

Range - Letters, numerals and special characters - max. 32 characters

Default value - N/A

Comments - Absence of APN name disables login into GPRS network

9.3.4.4 Authorization

Function - Allow to choose authentication method of PPP protocol.

Data type - Selection list

Range - ***None***

None authentication method chosen

PAP

PAP authentication method chosen

CHAP

CHAP authentication method chosen

Default value - ***None***

Comments - N/A

9.3.4.5 APN user name

Function - Defines APN user name, which will be used to log into APN

Data type - Text

Range - Letters, numerals and special characters - max. 32 characters

Default value - N/A

Comments - Optional parameter used only if required by GSM network operator

9.3.4.6 APN password

Function - Defines password, which will be used to log into APN

Data type - Text

Range - Letters, numerals and special characters - max. 32 characters

Default value - N/A

Comments - Optional parameter used only if required by GSM network operator

9.3.4.7 GPRS testing interval (ping)

Function - Defines in minutes interval of testing GPRS connection

Data type - Number

Range - ***0 - 250 [min.]***

Default value - ***40 [min.]***

Comments - Testing is performed by sending data frames to defined by the parameter GPRS testing address. Test frames are sent when the module is logged into APN and no communication is performed the period defined by this parameter. If the test fails, the module does not receive confirmation within 12 seconds and after 3 retries - the connection to the APN is reset.

9.3.4.8 GPRS testing address (ping)

Function - Defines IP address used for sending GPRS transmission test frames.

Data type - IP address

Range - ***0.0.0.0 - 255.255.255.255***

Default value - ***0.0.0.0***

Comments - When this field is left at default value 0.0.0.0 test frames are sent to IP chosen by module from Authorized IP list. It is advised to set this parameter to IP address of device collecting data or other IP address always connected to APN.

9.3.4.9 Roaming

Function - Defines whether operation in foreign GSM network is allowed

Data type - Selection list

Range - ***On***

In case of absence of no network, the module will attempt to login to other available network

Off

Login into foreign networks is not allowed

Default value - ***Off***

Comments - This parameter decides whether module will try to login to available foreign networks during the absence in the absence of home network. This is possible only when the SIM card in module has the roaming service enabled.

9.3.5 SIM2

SIM2 group contains parameters responsible for establishing the GSM/GPRS communication by using SIM card inserted into SIM2 slot (the lower one).

9.3.5.1 Address IP

Function - Displays IP address assigned to SIM card placed in SIM holder slot number 2 using if the communication with module has been established earlier on that slot. It can be used for remote configuration via GPRS.

Data type - IP address

Range - ***0.0.0.0 - 255.255.255.255***

Default value - ***0.0.0.0***

Comments - When this field is left at default value 0.0.0.0 remote communication with the module is possible using other IP addresses. Obviously IP address can be inserted manually to allow access to remote module via that SIM card if is logged.

9.3.5.2 SIM card PIN number

Function - Defines PIN access code for SIM module delivered by GSM operator. For SIM modules not protected by PIN code, the value is insignificant.

Data type - Text

Range - Numerals, max 8 characters

Default value - N/A

Comments - Wrong PIN can cause SIM card lock

NOTICE!
Caution is vital, when setting the PIN code value. Entering incorrect PIN code may cause module start-up impossible and lock SIM card. For security reasons module makes attempt to enter PIN code twice.

To unlock SIM card please follow procedure described in Problem solving chapter.

9.3.5.3 APN name

Function - Defines APN name which is used by module to carry out GPRS transmission using that SIM card

Data type - Text

Range - Letters, numerals and special characters - max. 32 characters

Default value - N/A

Comments - Absence of APN name disables login into GPRS network

9.3.5.4 Authorization

Function - Allows to choose authentication method of PPP protocol.

Data type - Selection list

Range - ***None***
None authentication method chosen
PAP
PAP authentication method chosen
CHAP
CHAP authentication method chosen

Default value - ***None***

Comments - N/A

9.3.5.5 APN user name

Function - Defines APN user name, which will be used to log into APN

Data type - Text

Range - Letters, numerals and special characters - max. 32 characters

Default value - N/A

Comments - Optional parameter used only if required by GSM network operator

9.3.5.6 APN password

Function - Defines password, which will be used to log into APN

Data type - Text

Range - Letters, numerals and special characters - max. 32 characters

Default value - N/A

Comments - Optional parameter used only if required by GSM network operator

9.3.5.7 GPRS testing interval (ping)

Function - Defines in minutes interval of testing GPRS connection

Data type - Number

Range - ***0 - 250 [min.]***

Default value - ***40 [min.]***

Comments - Testing is performed by sending data frames to defined by the parameter GPRS testing address. Test frames are sent when the module is logged into APN and no communication is performed during the period defined by this parameter. If the test fails, that is the module does not receive confirmation within 12 seconds and after 3 retries - the connection to the APN is reset.

9.3.5.8 GPRS testing address (ping)

Function - Defines IP address used for sending GPRS transmission test frames.

Data type - IP address

Range - ***0.0.0.0 - 255.255.255.255***

Default value - ***0.0.0.0***

Comments - When this field is left at default value 0.0.0.0 test frames are sent to IP chosen by module from Authorized IP list. It is advised to set this parameter to IP address of device collecting data or other IP address always connected to APN.

9.3.5.9 Roaming

Function - Defines whether operation in foreign GSM network is allowed

Data type - Selection list

Range - ***On***

In case of absence of home network, the module will attempt to login to other available network

Off

Login into foreign networks is not allowed

Default value - ***Off***

Comments - This parameter decides whether module will try to login to available foreign networks in the absence of home network. This is possible only when the SIM card in module has the roaming service enabled.

9.3.6 GPRS

GPRS contains parameters applying to GPRS communication handling valid for both SIM cards.

9.3.6.1 Sender IP address control

Function - Switches the control of sender IP address on/off

Data type - Selection list

Range - ***Yes***

The module exchanges information only with IP addresses present on the Authorized IP list.

No

The module exchanges information (configuration, responses for queries) with any IP address sending qualified query or command. In this case the identification of the sender goes by its current identifier.

Default value - ***Yes***

Comments - Switching the control off enables verification of the sender on the base of its currently assigned identifier other than IP address (e.g. serial number or virtual IP for MT-1XX series). This allows communication among units with dynamically assigned IP addresses (within same APN). Sender's identifier must reside on Authorized IP list in order to establish the communication.

NOTE!!!

The configuration frames are always checked automatically by module and this verification not depends from the parameter settings. This verification is not perform in two cases:

1. factory configuration is in the module
2. Authorized IP numbers table is empty

9.3.6.2 Wait time after disconnection

Function - Defines interval between GPRS connection attempts

Data type - Number

Range - ***0.01 - 655.350 [s]***

Default value - ***5.00 [s]***

Comments - N/A

9.3.7 SMS

SMS contains parameters related to sending and receiving of text messages by **MT-151 HMI** module.

9.3.7.1 Daily SMS limit

Function - Defines maximum quantity of SMS, the module may send during one day. The parameter protects against uncontrolled sending of SMS messages and consequent high running expenses.

Data type - Number

Range - ***0 - 65535***

Default value - ***0***

Comments - Setting this parameter to ***0*** removes daily SMS limit

NOTICE!
Reaching set by the parameter limit results with unconditional stop of SMS sending. One has to bear in mind that until 00:00 o'clock no messages will be sent even in alarm situations! Unsent due to limitation SMS messages are queued (the queue holds up to 16 messages) and will be sent when it is possible (after midnight). If the number of queued messages is higher than the limit set by user, there is a risk of immediate consuming of the next day limit.

9.3.7.2 Number of SMS sending retries

Function - Defines maximum quantity of retries of unsuccessful SMS transmission

Data type - Number

Range - ***0 - 16***

Default value - ***3***

Comments - After reaching the defined value the SMS is deleted from sending queue.

9.3.7.3 SMS limit exceed information

Function - Contains text of the SMS message sent upon reaching Daily SMS limit.

Data type - Text

Range - Letters, numerals and special characters - max. 160 characters

Default value - N/A

Comments - This information is sent beyond standard messages queue and only **once a day**. This message does not increment SMS messages sending counter.

9.3.7.4 Recipient of SMS limit exceed information

Function - Selects the SMS limit alert recipient

Data type - Selection list

Range - ***None*** and numbers defined in GSM -> Authorized numbers -> Phone list for SMS transmission

Default value - ***None***

Comments - N/A

9.3.7.5 Answer for blank SMS

Function - Defines the text of reply for empty SMS to the sender.

Data type - Text

Range - Letters, numerals and special characters - max. 160 characters

Default value - ***Hello, here MT-151***

Comments - In replay message text may be used symbolic names and macros following syntax rules defined in Appendices in the SNCS commands syntax chapter.

9.3.7.6 Incoming SMS handling

Function - allows to choose how will be handled SMS messages,

Data type - selection list

Range - ***By system***

Module received SMS messages, system program process the content according to SNCS syntax, settings of macros and symbolic names created by user. System automatically check authorized phone numbers and realize commands and orders included in the message. At the finish module sends back the confirmation (if not disabled) and the message with response to sender.

By user program

Module received SMS messages and load the content into local buffer (registers in HREG sets in next parameters)

Receiving messages

System automatically check authorization for sender phone numbers and enter the number of signs into first register that is typed in Buffer address holding incoming SMS. It is a information for internal user program about new message appears in local buffer. Next registers are filled by content of the message, each sign is stored to little-endian part of the register. Register Buffer address holding sender phone number is filled with first digit of the phone number. Internal user program must reset the value in first register after the message processing is finished. This action is a signal for an system program to load another message into the local buffer.
Internal system queue of unprocessed messages counts maximum 30 SMS messages.

Sending messages

Sending SMS messages using internal user program also requires a local buffer starting from address that is typed in Buffer address holding outgoing SMS parameter. First register is a number of signs in sending message, next registers

are for content (sign per register). Internal user program have to load first the message content, phone number to registers typed in Buffer address holding receiver phone number and set the first register with a count values of the signs in message. Loading the message content is permitted only when value in first register is zero. The sending is generated when into first register is enter non zero value - module's system add the message into internal queue and set into first register value 32767. When the message have been sent (or deleted from queue because an error occurred), module set into first register value "0". First register of the buffer can be reset by internal user program to allow sending next message one after another without waiting for 32767 value.
System will copied the message content from registers into internal system queue until the zero value on next registers or reach limit value set in first register (number of signs).

Default value - ***By system***

Comments - module counts numbers of send messages and stops sending if limit is reached.
Multipart SMS messages are not supported. When module receive mutlipart SMS send error message to sender. Selection of **By user program** option still allow to generate messages by Sending rules.

9.3.7.7 Buffer address holding incoming SMS (HREG)

Function - address [dec] in holding register space determine first register where the number of signs from received message will be stored. Next registers will store next signs of the message content.

Data type - Number

Range - ***0 - 8000***

Default value - ***1100***

Comments - Maximum size of received message:
161 registers for standard message
71 registers for messaged that include one special sign. Each special sign fill two registers of buffer. The first register store number of signs in both cases.

9.3.7.8 Buffer address holding sender phone number (HREG)

Function - address [dec] in holding register space determine first register where the first digit of the sender's phone number will be stored. Next registers will store next digits of the phone number string.

Data type - Number

Range - ***0 - 8000***

Default value - ***1300***

Comments - phone number can begin with "+" sign or ASCII number or country direction number in international format. Phone number must be finished with register that store value "0".

9.3.7.9 Buffer address holding outgoing SMS (HREG)

Function - address [dec] in holding register space determine first register where the number of signs for outgoing message will be stored. Next registers will store next signs of the message content.

Data type - Number

Range - ***0 - 8000***

Default value - ***1400***

Comments - N/A

9.3.7.10 Buffer address holding receiver phone number (HREG)

Function - address [dec] in holding register space determine first register where the first digits of the receiver's phone number will be stored. Next registers will store next digits of the phone number string. Last register must have zero value.

Data type - Number

Range - ***0 - 8000***

Default value - ***1600***

Comments - phone number can begin with "+" sign or ASCII number or country direction number in international format. Phone number must be finished with register that store value "0".

9.3.7.11 Formats

Formats subgroup contains parameters allowing user to define formats of date and time presented in SMS messages.

9.3.7.11.1 Date format

Function - Defines date format used by #date predefined symbolic name

Data type - Text

Range - Letters, numerals and special characters - max. 31 characters

Default value - ***YYYY-DD-MM***

Comments - In the text user can put any sign combination but predefined with special meaning listed below:

YYYY - if placed in this format text automatically changed for year in four digit notation (eg. 2013),

YY - if placed in this format text automatically changed for year in two digit notation (eg. 13),

MM - if placed in this format text automatically changed for month (eg. 07 for January),
DD - if placed in this format text automatically changed for day of month (eg. 26).

Example:

Parameter is set to:
Date of measurement: YYYY-MM-DD
Macro result is (providing today is 26th of July 2013):
Date of measurement: 2013-07-26

9.3.7.11.2 Time format

Function	-	Defines date format used by *#time* predefined symbolic name
Data type	-	Text
Range	-	Letters, numerals and special characters - max. 31 characters
Default value	-	***HH:MN:SS***
Comments	-	In the text user can put any sign combination but predefined with special meaning listed below: *HH* - if placed in this format text automatically changed for current hour in 24h format (eg. 01), *MN* - if placed in this format text automatically changed for current minutes (eg. 23), *SS* - if placed in this format text automatically changed for current seconds (eg. 45). Example: Parameter is set to: *Time of measurement: HH:MN:SS* Macro result is (providing the time is 01:23:45): *Time of measurement: 01:23:45*

9.3.7.12 Symbolic names

Symbolic names group contains names assigned by the user referring to bits or registers. There can be defined up to 32 symbolic names. In order to use a symbolic name in SMS put place there a name preceded by '#' sign in SMS text and send it from module - it will be automatically changed to value of corresponding register or bit. Symbolic names can be used in macros and to poll module for data using SMS.
More about SMS messaging can be found in SNCS commands syntax chapter in Appendices.

9.3.7.12.1 Number of symbolic names

Function	-	Defines number of user defined symbolic names.
Data type	-	Number
Range	-	***1 - 32***
Default value	-	***1***
Comments	-	N/A

9.3.7.12.2 Symbolic name table

Idx. - Index number

Symbolic name - Friendly name facilitating identification of module resource.
Letters, numerals and special characters - max. 50 characters.
Default value is ***IREG0***.

Address space - ***Binary Inputs***
Binary inputs (address 1XXX), read only
Binary Outputs
Binary outputs (address 0XXX), read/write
Input Registers
Input registers (address 3XXX) also known as analog inputs address space, read only
Holding Registers
Holding registers (address 4XXX) also known as internal registers and analog outputs address space, read/write

Register/bit address - Address of bit or register to which symbolic name is assigned.
0 - 65535
Default value is ***0***.

9.3.7.13 Macros

Macros group contains up to 16 user-defined macros. Macro may contain ASCII signs, symbolic names, SMS commands and other macros that will be put in SMS text. In order to use a macro in SMS put place there a name preceded by '*' sign in SMS text send from mobile phone to module or in SMS text sent from module or other macro. Using macros makes composing complex SMS texts and queries much more convenient and user friendly.
More about SMS messaging can be found in SNCS commands syntax chapter in Appendices.

9.3.7.13.1 Number of macros

Function - Defines number of user defined macros.

Data type - Number

Range - ***1 - 16***

Default value - ***1***

Comments - N/A

9.3.7.13.2 Macro table

Idx. - Index number

Macro name - Friendly name facilitating identification of macro.
Letters, numerals and special characters - max. 20 characters.
Default value is ***M0***.

Macro content - Text to which macro is decoded. May use other macros with lower index, symbolic names and SMS commands as described in SNCS commands syntax chapter in Appendices.

Letters, numerals, special characters - max. 160 characters
Default value is *#date #time*.

9.3.8 Authorized numbers

Authorized numbers comprises lists of phone numbers and IP addresses the module can communicate with. The list of IP addresses serves to granting access to configuration and data reception privileges. Numbers and addresses saved in this group are then used as receivers in Rules.

9.3.8.1 Number of phone numbers

Function	-	Defines the length of phone numbers list authorized to exchange SMS messages.
Data type	-	Number
Range	-	***0 - 32***
Default value	-	***0***
Comments	-	The value of this parameter may vary as the result of adding/deleting when using the context menu operating directly on Phone list.

9.3.8.2 Number of IP addresses

Function	-	Defines the length of the IP addresses list
Data type	-	Number
Range	-	***0 - 32***
Default value	-	***0***
Comments	-	The value of this parameter may vary as the result of adding/deleting when using the context menu operating directly on IP list.

9.3.8.3 Numbers from SIM phone-book always allowed

Function	-	allows to authorize all phone numbers from SIM card to accept voice calls
Data type	-	selection list
Range	-	***Yes*** Phones from SIM are authorized ***No*** Authorized phone numbers only on phone list
Default value	-	***No***
Comments	-	If **Yes** is selected module allows incoming voice calls from SIM card number even authorization list is blank. This feature expand list of authorization numbers for more than 32 position in configuration. Phone numbers from SIM cards without names are not supported.

9.3.8.4 Phone

Idx.	-	Index number

Name - Friendly name facilitating identification of the receiver while defining Rules. Max. length is 16 characters.

Number - Phone number assigned to list index. Max. 23 characters

Receiving - The module receives and analyzes SMS messages depending on selected setting. When receiving is not allowed, all SMS messages will be deleted
Default value: ✗ (not allowed)

9.3.8.5 IP

Idx. - Index number

Name - Friendly name facilitating identification of the receiver while defining Rules. Max. length is 16 characters.

SIM1 address - IP address assigned to list index used when SIM card installed in SIM1 slot is used

SIM2 address - IP address assigned to list index used when SIM card installed in SIM2 slot is used. Parameter is available only when two SIM cards are used.

Protocol ***UDP***
Communication is carried out using UDP protocol

Configuration - Value of this parameter determines whether remote configuration data arriving from selected IP will be ignored or accepted
Default value: ✓ (allowed)

Receiving - Value of this parameter determines whether data arriving from selected IP will be accepted or ignored
Default value: ✓ (allowed)

SNMP Query - Value of this parameter determines whether SNMP request arriving from selected IP will be accepted or ignored
Default value: ✗ (not allowed)

9.4 Resources

Group **Resources** encompasses a list of hardware and software resources available to users.

9.4.1 Binary inputs (I1 - I16)

All parameters listed in this group are set individually for each binary input. Binary inputs operate in both positive and negative logic at the same time.

9.4.1.1 Name

Function - Friendly name facilitating identification of the binary input task

Data type - Text

Range - Letters and numerals - max. 31 characters

Default value - Respectively from ***I1*** to ***I16***

Comments - N/A

9.4.1.2 Input type

Function - Defines binary input operating mode

Data type - Selection list

Range - ***Binary input***
Selected terminal operates as binary input
Counting input
Selected terminal operates in impulse detection mode.

Default value - ***Binary input***

Comments - According to selected mode MTManager displays additional configuration parameters for inputs I1 ... I4

9.4.1.3 Filtering

Function - Defines (in seconds) minimum duration of electrical state on the input to be considered stable, thereby defining maximum time duration of electrical signal is considered as noise

Data type - Number

Range - ***0.01 - 600.00 [s]***

Default value - ***0.10 [s]***

Comments - Increasing the value increases noise immunity but delays change detection.

9.4.1.4 Flow calculation trigger

Function - Selects marker or any bit from module's address space. Change of bits state to high initiates flow calculation process.

Data type - Selection list

Range - Name from bits' list (see in Appendices) or ***1min***. or ***1hour*** predefined marker

Default value - ***1min.***

Comments - Available for Counting input as selection type of Input for I1 - I4.

9.4.1.5 Flow scaling

Function - Selects time reference units for flow scaling

Data type - Selection list

Range - ***None***
Defines value increase between next initiations period of flow calculation
Minute (eng. units/min)
Defines value increase per minute
Hour (eng. units/h)
Defines value increase per hour

Default value - ***None***

Comments - Available for Counting input as selection type of Input for I1 - I4.

9.4.1.6 Impulse weight - multiplier

Function	-	Allows for result correction of the flow using multiplication function
Data type	-	Number
Range	-	***1 - 1000***
Default value	-	***1***
Comments	-	The calculated value of the flow is outcome a mathematical operation expressed by the formula: y = a * x / b - c where y - flow value a - Impulse weight - Multiplier (eng. units) b - Impulse weight - Divider (eng. units) c - Offset (eng. units) Available for Counting input as selection type of Input for I1 - I4.

9.4.1.7 Impulse weight - divider

Function	-	Allows for result correction of the flow using division function
Data type	-	Number
Range	-	***1 - 1000***
Default value	-	***1***
Comments	-	The calculated value of the flow is outcome a mathematical operation expressed by the formula: y = a * x / b - c where y - flow value a - Impulse weight - Multiplier (eng. units) b - Impulse weight - Divider (eng. units) c - Offset (eng. units) Available for Counting input as selection type of Input for I1 - I4.

9.4.1.8 Offset - engineering units

Function	-	Allows for result correction of the flow by subtracting constant value
Data type	-	Number
Range	-	***0 - 1000***
Default value	-	***0***
Comments	-	The calculated value of the flow is outcome a mathematical operation expressed by the formula: y = a * x / b - c where y - flow value a - Impulse weight - Multiplier (eng. units) b - Impulse weight - Divider (eng. units)

c - Offset (eng. units)

Available for Counting input as selection type of Input for I1 - I4.

9.4.1.9 Hi alarm - engineering units

Function	-	Defines **Hi** alarm level for flow calculation value in engineering units.
Data type	-	Number
Range	-	***-32768 - 32767***
Default value	-	***32767***
Comments	-	If value of flow calculation value is higher than value of this parameter, then the **HiHi** alarm flag is raised. The resetting level of this flag depends on Alarm hysteresis - engineering units setting. Available for Counting input as selection type of Input for I1 - I4.

9.4.1.10 Lo alarm - engineering units

Function	-	Defines **Lo** alarm level for flow calculation value in engineering units.
Data type	-	Number
Range	-	***-32768 - 32767***
Default value	-	***32767***
Comments	-	If value of flow calculation value is higher than value of this parameter, then the **Lo** alarm flag is raised. The resetting level of this flag depends on Alarm hysteresis - engineering units setting. Available for Counting input as selection type of Input for I1 - I4.

9.4.1.11 Alarm hysteresis - engineering units

Function	-	Defines the hysteresis value for flow alarm threshold. The value is set in engineering units.
Data type	-	Number
Range	-	***0 - 32767***
Default value	-	***100***
Comments	-	Setting hysteresis relevant for signal fluctuations prevents excessive activation of alarm flags. Available for Counting input as selection type of Input for I1 - I4.

9.4.2 Binary outputs (Q1 - Q12)

All parameters listed in this group are set individually for each binary output.

9.4.2.1 Name

Function	-	Friendly name facilitating identification of the binary output task

Data type - Text

Range - Letters and numerals - max. 31 characters

Default value - Respectively from ***Q1*** to ***Q12***

Comments - N/A

9.4.2.2 Input type

Function - Defines binary output operating mode

Data type - Selection list

Range - ***Binary input***

Selected terminal operates as binary input

Binary output

Selected terminal operates as binary output

Default value - ***Binary output***

Comments - N/A

9.4.2.3 Filtering

Function - Defines (in seconds) minimum duration of electrical state on the input to be considered stable, thereby defining maximum time duration of electrical signal is considered as noise

Data type - Number

Range - ***0.01 - 600.00 [s]***

Default value - ***0.10 [s]***

Comments - Increasing the value increases noise immunity but delays change detection.
This parameter is available in binary input mode only.

9.4.3 Analog inputs 4-20mA (AI1 - AI4)

MT-151 HMI is equipped with four current analog inputs operating in 4-20mA range. All parameters but sampling frequency are set individually for each input.

9.4.3.1 Sampling frequency

Function - Defines analog input sampling frequency and measurement resolution

Data type - Selection list

Range - ***1Hz***

New measurement is available every second. Measurement is slower but more precise - resolution is nearly 20000 units (above 14 bits). This setting is advised for low-dynamics signals.

10Hz

New measurement is available every 100 milliseconds. Measurement is faster but less accurate - resolution is above 2000 units (11 bits). This setting is advised for low-dynamics signals.

Default value - ***1Hz***

Comments - N/A

9.4.3.2 Name

Function - Friendly name facilitating identification of the analog input task

Data type - Text

Range - Letters and numerals - max. 31 characters

Default value - Respectively from ***AI1*** to ***AI4***

Comments - N/A

9.4.3.3 Engineering units

Function - Allows user to enter unit name for information purpose

Data type - Text

Range - Letters and numerals - max. 15 characters

Default value - ***μA***

Comments - N/A

9.4.3.4 Low reference - internal units

Function - Defines number of μA corresponding to number of engineering units defined by Low reference - engineering units parameter

Data type - Number

Range - ***4000 - 20000 [μA]***

Default value - ***4000 [μA]***

Comments - Used along with other reference parameters for rescaling input signal to engineering units.

9.4.3.5 Low reference - engineering units

Function - Defines number of engineering units corresponding to number of μA defined by Low reference - internal units parameter

Data type - Number

Range - ***-32768 - 32767***

Default value - ***4000***

Comments - Used along with other reference parameters for rescaling input signal to engineering units.

9.4.3.6 High reference - internal units

Function - Defines number of μA corresponding to number of engineering units defined by High reference - engineering units parameter

Data type - Number

Range - ***4000 - 20000 [μA]***

Default value - ***20000 [μA]***

Comments - Used along with other reference parameters for rescaling input signal to engineering units.

9.4.3.7 High reference - engineering units

Function - Defines number of engineering units corresponding to number of µA defined by High reference - internal units parameter

Data type - Number

Range - ***-32768 - 32767***

Default value - ***20000***

Comments - Used along with other reference parameters for rescaling input signal to engineering units.

9.4.3.8 HiHi alarm - engineering units

Function - Defines **HiHi** alarm level for analog signal value in engineering units.

Data type - Number

Range - ***-32768 - 32767***

Default value - ***32767***

Comments - If value of analog signal is higher than value of this parameter, then the **HiHi** alarm flag is raised. The resetting level of this flag depends on Alarm hysteresis - engineering units setting.

9.4.3.9 Hi alarm - engineering units

Function - Defines **Hi** alarm level for analog signal value in engineering units.

Data type - Number

Range - ***-32768 - 32767***

Default value - ***32767***

Comments - If value of analog signal is higher than value of this parameter, then the **Hi** alarm flag is raised. The resetting level of this flag depends on Alarm hysteresis - engineering units setting.

9.4.3.10 Lo alarm - engineering units

Function - Defines **Lo** alarm level for analog signal value in engineering units.

Data type - Number

Range - ***-32768 - 32767***

Default value - ***-32768***

Comments - If value of analog signal is lower than value of this parameter, then the **Lo** alarm flag is raised. The resetting level of this flag depends on Alarm hysteresis - engineering units setting.

9.4.3.11 LoLo alarm - engineering units

Function - Defines **LoLo** alarm level for analog signal value in engineering units.

Data type - Number

Range - ***-32768 - 32767***

Default value - ***-32768***

Comments - If value of analog signal is lower than value of this parameter, then the **LoLo** alarm flag is raised. The resetting level of this flag depends on Alarm hysteresis - engineering units setting.

9.4.3.12 Alarm hysteresis - engineering units

Function - Defines in engineering units hysteresis for analog inputs alarms.

Data type - Number

Range - ***0 - 65535***

Default value - ***100***

Comments - Setting proper value prevents from turning on and off alarms too often, when measured value is oscillating around alarm value.

9.4.3.13 Deadband - engineering units

Function - Defines a minimum change of registered analog signal which should set to high state deadband flag corresponding to analog input where the change was detected (AI1_DB - AI4_DB). This flag is reset to 0 after one program cycle.

Data type - Number

Range - ***0 - 65535***

Default value - ***100***

Comments - Deadband is very useful for tracking analog signal on server - data is send only when analog input changes.

9.4.4 Analog inputs 0-10V (AV1 - AV2)

MT-151 HMI is equipped with two voltage analog inputs operating in 0-10V range. All parameters are set individually for each input.

9.4.4.1 Name

Function - Friendly name facilitating identification of the analog input task

Data type - Text

Range - Letters and numerals - max. 31 characters

Default value - Respectively ***AV1*** and ***AV2***

Comments - N/A

9.4.4.2 Engineering units

Function - Allows user to enter unit name for information purpose

Data type - Text

Range - Letters and numerals - max. 15 characters

Default value - ***mV***

Comments - N/A

9.4.4.3 Low reference - internal units

Function - Defines number of mV corresponding to number of engineering units defined by Low reference - engineering units parameter

Data type - Number

Range - ***0 - 10000 [mV]***

Default value - ***0 [mV]***

Comments - Used along with other reference parameters for rescaling input signal to engineering units.

9.4.4.4 Low reference - engineering units

Function - Defines number of engineering units corresponding to number of mV defined by Low reference - internal units parameter

Data type - Number

Range - ***-32768 - 32767***

Default value - ***0***

Comments - Used along with other reference parameters for rescaling input signal to engineering units.

9.4.4.5 High reference - internal units

Function - Defines number of mV corresponding to number of engineering units defined by High reference - engineering units parameter

Data type - Number

Range - ***0 - 10000 [mV]***

Default value - ***10000 [mV]***

Comments - Used along with other reference parameters for rescaling input signal to engineering units.

9.4.4.6 High reference - engineering units

Function - Defines number of engineering units corresponding to number of mV defined by High reference - internal units parameter

Data type - Number

Range - ***-32768 - 32767***

Default value - ***10000***

Comments - Used along with other reference parameters for rescaling input signal to engineering units.

9.4.4.7 HiHi alarm - engineering units

Function - Defines **HiHi** alarm level for analog signal value in engineering units.

Data type - Number

Range - ***-32768 - 32767***

Default value - ***32767***

Comments - If value of analog signal is higher than value of this parameter, then the **HiHi** alarm flag is raised.

The resetting level of this flag depends on Alarm hysteresis - engineering units setting.

9.4.4.8 Hi alarm - engineering units

Function - Defines **Hi** alarm level for analog signal value in engineering units.

Data type - Number

Range - ***-32768 - 32767***

Default value - ***32767***

Comments - If value of analog signal is higher than value of this parameter, then the **Hi** alarm flag is raised. The resetting level of this flag depends on Alarm hysteresis - engineering units setting.

9.4.4.9 Lo alarm - engineering units

Function - Defines **Lo** alarm level for analog signal value in engineering units.

Data type - Number

Range - ***-32768 - 32767***

Default value - ***-32768***

Comments - If value of analog signal is lower than value of this parameter, then the **Lo** alarm flag is raised. The resetting level of this flag depends on Alarm hysteresis - engineering units setting.

9.4.4.10 LoLo alarm - engineering units

Function - Defines **LoLo** alarm level for analog signal value in engineering units.

Data type - Number

Range - ***-32768 - 32767***

Default value - ***-32768***

Comments - If value of analog signal is lower than value of this parameter, then the **LoLo** alarm flag is raised. The resetting level of this flag depends on Alarm hysteresis - engineering units setting.

9.4.4.11 Alarm hysteresis - engineering units

Function - Defines in engineering units hysteresis for analog inputs alarms.

Data type - Number

Range - ***0 - 65535***

Default value - ***100***

Comments - Setting proper value prevents from too often turning on and off alarms when measured value is oscillating around alarm value.

9.4.4.12 Deadband - engineering units

Function - Defines a minimum change of registered analog signal which should set to high state deadband flag corresponding to analog input where the change was detected (AV1_DB and AV2_DB). This flag is reset to 0 after one program cycle.

Data type - Number

Range - ***0 - 65535***

Default value - ***100***

Comments - Deadband is very useful for tracking analog signal on server - data is send only when analog input changes.

9.4.5 Counters (CNT1 - CNT16)

Counters may be used to count any pulses (interpreted as bit or binary input state changes). Counters are equipped with two inputs each - one incrementing and one decrementing counter register value.

9.4.5.1 Incrementing input

Function - Defines the bit which state change increments counter value by 1

Data type - Number or Selection list

Range - ***0 - 65535*** or name from bit list (see bit list in Appendices)

Default value - N/A

Comments - Bit addresses ***0 - 9999*** point to analog inputs/binary inputs address space while addresses ***10000 - 65535*** point to Internal registers/binary outputs address space.
More information on calculating bit addresses can be found in Memory map chapter in Appendices.

9.4.5.2 Active edge of incrementing input

Function - Defines edge of incrementing bit which increments counter value by 1

Data type - Selection list

Range - ***0->1***
logical state change from 0 to 1
1->0
logical state change from 1 to 0

Default value - ***0->1***

Comments - N/A

9.4.5.3 Decrementing input

Function - Defines the bit which state change decrements counter value by 1

Data type - Number

Range - ***0 - 65535*** or name from bit list (see bit list in Appendices)

Default value - N/A

Comments - Bit addresses ***0 - 9999*** point to analog inputs/binary inputs address space while addresses ***10000 - 65535*** point to Internal registers/binary outputs address space.

More information on calculating bit addresses can be found in Memory map chapter in Appendices.

9.4.5.4 Active edge of decrementing input

Function - Defines edge of decrementing bit which decrements counter value by 1

Data type - Selection list

Range - ***0->1***
logical state change from 0 to 1
1->0
logical state change from 1 to 0

Default value - ***0->1***

Comments - N/A

9.4.5.5 Counting range (32 bits)

Function - Defines the bit which state change increments counter value by 1

Data type - Number

Range - ***0 - 2147483647***

Default value - ***0***

Comments - When counting up the counter is zeroed by next appearing pulse upon reaching declared value. When counting down, next pulse writes declared value into the counter upon reaching 0. Setting this parameter to *0* turns off counter.

9.4.6 Timers

Timers group contains configuration parameters of module timers.

9.4.6.1 Synchronous timers (CT1 - CT16)

Synchronous timers measure cyclically defined time intervals. They are synchronized with module real time clock (RTC). The CT flag corresponding with timer is set to high state in one program cycle while the setting value of period has been counted.

9.4.6.1.1 Start [HH:MM]

Function - Defines the synchronization point of timer with RTC

Data type - Time

Range - ***00:00 - 23:59***

Default value - ***00:00***

Comments - At the time defined by this parameter the module will always set timer flag to high state.

9.4.6.1.2 Period

Function - Defines time period counted by timer

Data type - Selection list

Range - ***None, 1 min., 2 min., 3 min., 5 min., 10 min., 15 min., 30 min., 1 hour, 2 hours, 3 hours, 4 hours, 6 hours, 8 hours, 12 hours, 24 hours***

Default value - ***None***

Comments - Choosing ***None*** disables the timer.

9.4.6.1.3 Days of week

Function - Defines days of week when timer is active

Data type - Multiple choice field

Range - ***Mo., Tu., We., Th., Fr., St., Su.***

Default value - ***Mo., Tu., We., Th., Fr., St., Su.*** (all week days are selected)

Comments - Timer is active when date and time of module RTC matching following equation:
(X **OR** Y)**AND** Z = 1,
where X=1, when current RTC day of week is selected on Days of week parameter; if it is not then X = 0,
Y=1, when current RTC day of month is selected on Days of month parameter; if it is not then Y = 0,
Z=1, when current RTC month is selected on Months parameter; if it is not then Z = 0.
E.g. if selected day of week is Friday, day of month is 13 and all months are selected timer will operate on all Fridays and on 13th day of each month.

9.4.6.1.4 Days of month

Function - Defines days of month when timer is active

Data type - Multiple choice field

Range - ***1 - 31, Last***

Default value - ***No day selected*** (no month day is selected)

Comments - Timer is active when date and time of module RTC matching following equation:
(X **OR** Y)**AND** Z = 1,
where X=1, when current RTC day of week is selected on Days of week parameter; if it is not then X = 0,
Y=1, when current RTC day of month is selected on Days of month parameter; if it is not then Y = 0,
Z=1, when current RTC month is selected on Months parameter; if it is not then Z = 0.
E.g. if selected day of week is Friday, day of month is 13 and all months are selected timer will operate on all Fridays and on 13th day of each month.

9.4.6.1.5 Months

Function - Defines months when timer is active

Data type - Multiple choice field

Range - ***Jan., Feb., Mar., Apr., May, Jun., Jul., Aug., Sep., Oct., Nov., Dec.***

Default value - ***Jan., Feb., Mar., Apr., May, Jun., Jul., Aug., Sep., Oct., Nov., Dec.*** (all months are selected)

Comments - Timer is active when date and time of module RTC matching following equation:
(X **OR** Y)**AND** Z = 1,
where X=1, when current RTC day of week is selected on Days of week parameter; if it is not then X = 0,
Y=1, when current RTC day of month is selected on Days of month parameter; if it is not then Y = 0,
Z=1, when current RTC month is selected on Months parameter; if it is not then Z = 0.
E.g. if selected day of week is Friday, day of month is 13 and all months are selected timer will operate on all Fridays and on 13th day of each month.

9.4.6.2 Asynchronous timers (CK1 - CK16)

Synchronous timers measure cyclically defined time intervals. They are not synchronized with module real time clock (RTC) - they start counting time when module is powered on or reset. Each time is counted CK flag corresponding to timer is set to high level for one program cycle.

9.4.6.2.1 Activating input

Function - Defines the bit which state turns on (bit set to logical 1) or off (bit set to logical 0) timer

Data type - Number or Selection list

Range - ***0 - 65535*** or name from bit list (see bit list in Appendices)

Default value - ***None***

Comments - Bit addresses ***0 - 9999*** point to analog inputs/binary inputs address space while addresses ***10000 - 65535*** point to Internal registers/binary outputs address space.
More information on calculating bit addresses can be found in Memory map chapter in Appendices.

9.4.6.2.2 Reset input

Function - Defines the bit which state resets timer. When bit is set to logical 1 - timer it stopped and zeroed. When bit is set to logical 0 - timer is counting.

Data type - Number or Selection list

Range - ***0 - 65535*** or name from bit list (see bit list in Appendices)

Default value - ***None***

Comments - Bit addresses ***0 - 9999*** point to analog inputs/binary inputs address space while addresses ***10000 - 65535*** point to Internal registers/binary outputs address space.
More information on calculating bit addresses can be found in Memory map chapter in Appendices.

9.4.6.2.3 Timer time unit

Function	-	Defines timer time unit and therefore precision
Data type	-	Selection list
Range	-	***1s, 0.01s***
Default value	-	***1s***
Comments	-	N/A

9.4.6.2.4 Counting range in timer units

Function	-	Defines timer counting range
Data type	-	Number
Range	-	***0 - 2147483647***
Default value	-	***0***
Comments	-	N/A

9.4.7 Constant parameters

Constant parameters are the constant values entered in configuration which can be used within **MT-151 HMI** program what allows to parameterize universal program for application needs.

9.4.7.1 Number of constant parameters

Function	-	Defines number of constant parameters on list
Data type	-	Number
Range	-	***0 - 128***
Default value	-	***0***
Comments	-	N/A

9.4.7.2 Number of constant parameters (textual)

Function	-	Defines number of constant textual parameters on list
Data type	-	Number
Range	-	***0 - 72***
Default value	-	***0***
Comments	-	N/A

9.4.7.3 Parameter 1 - 128

Function	-	Defines value of constant parameter
Data type	-	Number
Range	-	***-32768 - 32767***
Default value	-	***0***
Comments	-	N/A

9.4.7.4 Parameter 1....72 (textual)

Textual parameters in text format. Max. 31 characters.

9.4.8 SD card

Micro SD card can be installed in the module and is designated for store data from internal data logger. It is additional copy of the internal logger data. Information is stored in CSV files in the similar format applied in MT-Data Provider. File creation frequency and managing of free memory space is configurable. CSV file creation on the memory card is completely independent from internal logger feature and its data distribution that is configurable by events definition of record and rules sending of stored data blocks.

9.4.8.1 Use of card

Function	-	Turns on/off copying logger data to memory card function.
Data type	-	Selection list
Range	-	***Yes*** Copying is enabled ***No*** Copying is disabled
Default value	-	***No***
Comments	-	N/A

9.4.8.2 Start

Function	-	Defines the synchronization point of timer with RTC
Data type	-	Time
Range	-	***00:00 - 23:59***
Default value	-	***00:00***
Comments	-	Each time defined by this parameter the module will always create CSV file with logger data. User can define the solid cycle of backup file creations if period parameter will be other than option *None*.

9.4.8.3 Period

Function	-	Defines time period counted by timer
Data type	-	Selection list
Range	-	***None, 5 min., 10 min., 15 min., 30 min., 1 hour, 2 hours, 3 hours, 4 hours, 6 hours, 8 hours, 12 hours, 24 hours***
Default value	-	***None***
Comments	-	Choosing ***None*** disables the data copying function.

9.4.8.4 Delete data older than

Function	-	Erases files from memory card older than number of selected days
Data type	-	Selection list
Range	-	***0 - 365***
Default value	-	***0***
Comments	-	Value ***0*** turn off erasing an old files function.

9.4.8.5 Delete data when low on memory

Function	-	Erases oldest files from memory card when run out of on the card.
Data type	-	Selection list
Range	-	***Yes***

Old data files will be erased when the memory card is full.

No Old data files will not be erased, new files will not be created.

Default value - ***No***

Comments - N/A

9.4.9 Display

MT-151 HMI is equipped with 128x64 graphical panel used for presenting device status and user information in both textual and graphical form. User has possibility to add his own textual welcome screen, up to eight textual data screens and up to four charts. Presented data and text can change dynamically according to changes of register values or bit states. All screens change automatically in cycle. Status screens change automatically every 10s while display time of the user defined screens is configurable. User can also navigate through screens using arrow buttons located on right from screen. Pressing ESC button stops automatic screen change mechanism for 5 minutes. Below is a description of each screen.

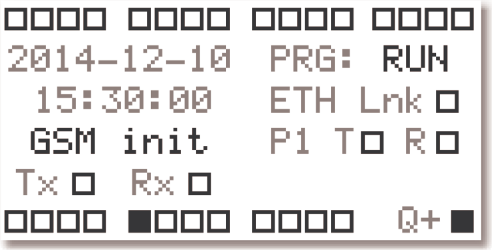
Start screen

After power up the module or upload a new configuration on display is shown startup screen with manufacturer logo and actual firmware version. Start screen cannot turned off.

Status screens

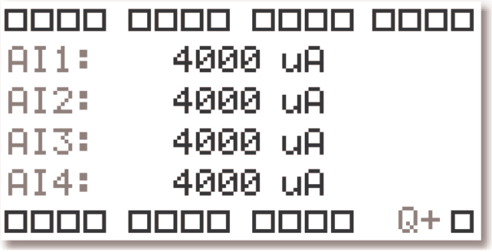
In default configuration module presents three status screens:

- **Module status screen** - presents digital I/Os state, module date and time, GSM modem state/signal, GSM modem activity (Tx and Rx), program status (RUN, STOP, WAIT, NONE), Ethernet link status ETH Lnk(if Ethernet is turned on) and serial ports activity (Tx and Rx for active ports, only P1 is shown on the picture below).



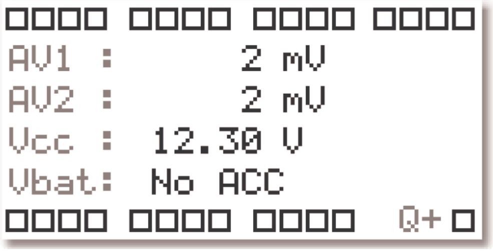
```
□□□□ □□□□ □□□□ □□□□
2014-12-10  PRG: RUN
 15:30:00   ETH Lnk □
 GSM init   P1 T□ R□
 Tx □  Rx □
□□□□ ■□□□ □□□□  Q+ ■
```

- **Current analog inputs status screen** - presents in engineering units with engineering unit name values measured on analog inputs AI1 - AI4 and digital I/Os state.



```
□□□□ □□□□ □□□□ □□□□
AI1:    4000 uA
AI2:    4000 uA
AI3:    4000 uA
AI4:    4000 uA
□□□□ □□□□ □□□□  Q+ □
```

- **Power and voltage analog inputs status screen** - presents in engineering units with engineering unit name values measured on analog inputs AV1 – AV2, mains power supply voltage, battery voltage (if available) and digital I/Os state.



```
□□□□ □□□□ □□□□ □□□□
AV1 :      2 mV
AV2 :      2 mV
Vcc :  12.30 V
Vbat:  No ACC
□□□□ □□□□ □□□□  Q+□
```

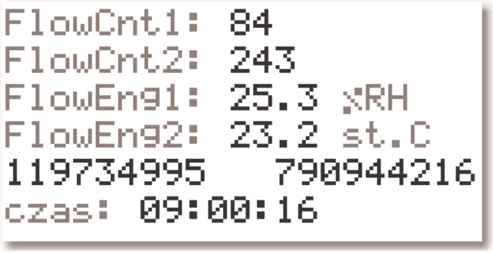
Last two status screens can be turned off in configuration while first one is always available as it provides vital information about module status and/or errors.

User screens

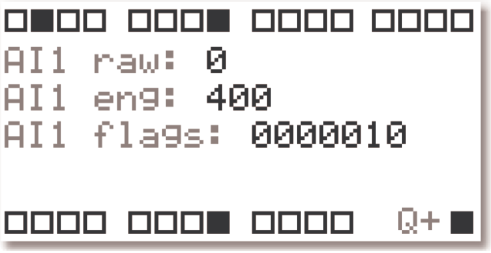
User can define three types of screen:

- **Welcome screen** - optional screen that is displayed during module startup for user defined time. It consists of 6 lines of static text 21 signs each. It's purpose to provide information about integrator and/or technical support contact. This screen can present only static text.

- **User screens** - up to 8 optional screens displayed during normal work by module. Screen consists with 4 to 6 lines of text depending whether user configured screen to present digital inputs and digital outputs/inputs state. It is possible to have one value per line that can be changed using keyboard located left of screen. To do so user needs to push OK button. First editable value is now highlighted and user can switch between available editable variables using arrow keys. Pressing OK chooses variable for editing (highlight starts blinking). Then user can choose value from available range by pressing OK. Pressing ESC at any step exits data entry mode.

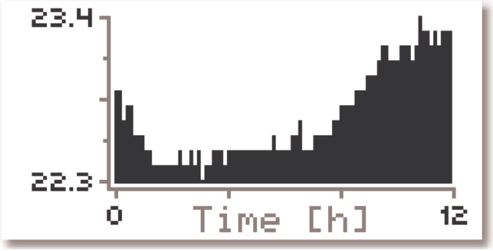


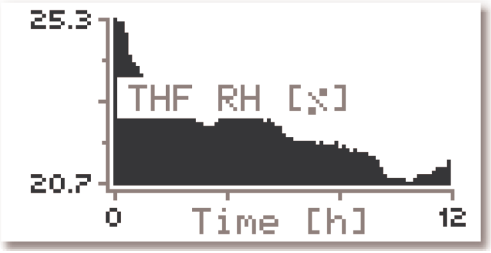
```
FlowCnt1: 84
FlowCnt2: 243
FlowEng1: 25.3 %RH
FlowEng2: 23.2 st.C
119734995   790944216
czas: 09:00:16
```



```
□■□□ □□□■ □□□□ □□□□
AI1 raw: 0
AI1 eng: 400
AI1 flags: 0000010

□□□□ □□□■ □□□□  Q+■
```

- **Chart screens** - up to 4 optional screens with charts presenting data change over time. Each chart allows presenting up to 90 samples. Each chart is using 100 registers from holding registers address space for data acquisition. Charts are using registers:
  W1 7000 - 7099
  W2 7100 - 7199
  W3 7200 - 7299
  W4 7300 - 7399





Registers are either filled automatically (parameter Data acquisition is set to *Automatic*) or by user (parameter Data acquisition is set to *User*) manually, by external device or by user program.
Length of units value on Y axis is limited to 4 symbols and one sign – values exceeding this limit are rounded down and presented in engineering notation if needed, e.g. -32100 is presented as -32K1, -31.99 is presented as -31.9

For details refer to Chart acquisition description located in Appendices.

9.4.9.1 Show status screens

Function	-	Turns on/off showing of **Current analog inputs status screen** and **Power and voltage analog inputs status screen**.
Data type	-	Selection list
Range	-	***Yes*** All three status screens are shown ***No*** Only main **Module status screen** is shown
Default value	-	***Yes***
Comments	-	N/A

9.4.9.2 Show welcome screen

Function	-	Turns on/off showing of user defined welcome screen during the Module startup.
Data type	-	Selection list
Range	-	***Yes*** Welcome screen is shown ***No*** Welcome screen is not shown
Default value	-	***Yes***
Comments	-	N/A

9.4.9.3 User screen count

Function	-	Sets number of User screens visible on device display.
Data type	-	Number
Range	-	***0 - 16***
Default value	-	***0***
Comments	-	N/A

9.4.9.4 Chart count

Function - Sets number of Charts screens visible on device display.

Data type - Number

Range - ***0 - 4***

Default value - ***0***

Comments - N/A

9.4.9.5 Password protected data entry

Function - Allows to activate a password protection for unauthorized data entry from display level.

Data type - Selection list

Range - ***Yes***

Security is active. Before enter value from display level user must enter access code in menu Passcode in display settings. Access to menu is granted when OK button is press and hold by 3 seconds.

No

Security is off. Data entry possible without enter the password.

Default value - ***No***

Comments - N/A

9.4.9.6 Access code

Function - access code which allow to enter data or change state from display level.

Data type - Number

Range - ***0 - 999999***

Default value - ***123456***

Comments - N/A

9.4.9.7 Data entry time interval [min.]

Function - define in minutes time limit for possibility of data modification from display level.

Data type - Number

Range - ***0 - 60***

Default value - ***30***

Comments - N/A

9.4.9.8 Welcome screen

Welcome screen is designed to show statical text information e.g. phone and address of an integrator. Welcome screen is presented just after Start screen. Display time is configurable. After Welcome screen module shows Module status screen.

9.4.9.8.1 Display time

Function - Sets screen displaying duration in range between 1 to 60 seconds.

Data type - Number

Range - ***1 - 60***

Default value - ***1***

Comments - N/A

9.4.9.8.2 Line 1 ... 6

Function - Allows to enter static text shown on display during module is startup.

Data type - Text

Range - Letters and numbers, maximum 35 characters

Default value - ***none***

Comments - A displayed text is brighter if is preceded with (!) exclamation mark. Display shows only 21 characters.

9.4.9.9 User screens SCR1 ... 16

Each from screens consists from max. 6 lines to present information like static text or dynamics values from allocated registers. Screens are changed in order from SCR1 to SCR16 after showing the Module status screen. Switching cycle is looped. Continuously displayed screen breaks the loop. Arrow keys can switch screens at any time in given sequence.

9.4.9.9.1 Display time

Function - Sets User screen displaying duration in range between 1 to 254 seconds with additional option doesn't show or continuously display.

Data type - Number or List

Range - ***Off, 1 - 254, Continuous***

Default value - ***Off***

Comments - Time settings not affect the buttons control.

9.4.9.9.2 Show inputs

Function - Enables/disables showing of binary inputs I1 ... I16 logical states on this screen. If states are presented they consume one of available text lines.

Data type - Selection list

Range - ***Yes***

Binary inputs state is presented (upper terminal lath)

No

Binary input state is not presented. Additional line is available for edition.

Default value - ***Yes***

Comments - N/A

9.4.9.9.3 Show outputs

Function - Enables/disables showing of binary inputs/outputs Q1 ... Q12 logical states on this screen. If states are presented they consume one of available text lines.

Data type - Selection list

Range - ***Yes***

Binary inputs/outputs state is presented (lower terminal lath)

No

Binary inputs/outputs state is not presented. Additional line is available for edition.

Default value - ***Yes***

Comments - N/A

9.4.9.9.4 Line 1 ... 6

Function - Allows to enter static text and dynamics data links displayed on display in up to 6 lines each 21 characters.

Data type - Text

Range - Letters and numbers, SNCS syntax, maximum 35 characters

Default value - ***none***

Comments - Access to line 5 and 6 is available after switching off preview of inputs/outputs.
A displayed text is brighter if is preceded with (!) exclamation mark.
Display shows only 21 characters.
More information in User screens programming chapter.

9.4.9.10 Charts W1 ... 4

Four screens for chart presentation of registers value in time function. Screens are changed in order from W1 to W4 after showing the User screens SCR1 ... 8. Switching cycle is looped. Continuously displayed screen breaks the loop. Arrow keys can switch screens at any time in given sequence.

9.4.9.10.1 Chart name

Function - Chart name visible for 3 seconds when entering chart screen.

Data type - Text

Range - Letters and numbers, maximum 15 characters

Default value - ***none***

Comments - Name can be shown again after pressing OK button.

9.4.9.10.2 Display time

Function - Sets Chart screen displaying duration in range between 1 to 254 seconds with additional option doesn't show or continuously display.

Data type - Number or List

Range - ***Off, 1 - 254, Continuous***

Default value - ***Off***

Comments - Time settings not affect the buttons control.

9.4.9.10.3 Data acquisition

Function - Allows choosing method of providing data.

Data type - Number or List

Range - ***Automatic***

Stores data from pointed register from selected space with fixed interval.

User

All sample values and timestamps are stored in Holding registers address space. Data to this registers can be entered manually, by external device or by user's program.

Default value - ***Automatic***

Comments - For details refer to Chart acquisition description located in Appendices

9.4.9.10.4 Sample interval

Function - Sets sampling interval for data points displayed on chart.

Data type - Selection list

Range - ***1 sec., 5 sec., 10 sec., 30 sec., 1 min., 4 min., 8 min., 16 min., 32 min., 64min.***

Default value - ***1 min.***

Comments - Parameter visible only when Data acquisition parameter is set to *Automatic*.

9.4.9.10.5 Register space

Function - Sets registers address space for chart data source register.

Data type - Selection list

Range - ***IREG***

Input (analog) registers space.

HREG

Holding registers space.

Default value - ***IREG***

Comments - Parameter visible only when Data acquisition parameter is set to *Automatic*.

9.4.9.10.6 Register address

Function - Sets registers address (dec) for chart data source

Data type - Selection list

Range - ***0 - 255*** *for* ***IR*** *space source*
0 - 8191 *for* ***HR*** *space source*

Default value - ***1***

Comments - Parameter visible only when Data acquisition parameter is set to *Automatic*.

9.4.9.10.7 Data scaling - multiplier

Function	-	Allows to set multiplying factor for data source register.
Data type	-	Number
Range	-	***1 - 1000***
Default value	-	***1***
Comments	-	Parameter visible only when Data acquisition parameter is set to *Automatic*.

9.4.9.10.8 Data scaling - divider

Function	-	Allows to set dividing factor for data source register.
Data type	-	Number
Range	-	***1 - 1000***
Default value	-	***1***
Comments	-	Quotient will be rounded down to integer value. Parameter visible only when Data acquisition parameter is set to *Automatic*.

9.4.9.10.9 Data format

Function	-	Allows setting a final view of the value in decimal fraction.
Data type	-	Selection list
Range	-	***Integer*** Acquired data will be presented as is. ***1 decimal place*** Acquired data will be presented as floating point value with one decimal place (e.g. 1001 as 100.1) ***2 decimal place*** Acquired data will be presented as floating point value with two decimal place (e.g. 1001 as 10.01)
Default value	-	***Integer***
Comments	-	Parameter visible only when Data acquisition parameter is set to *Automatic*.

9.4.9.10.10 Y axis scaling

Function	-	Allows choosing Y axis method of scaling.
Data type	-	Number or List
Range	-	***Automatic*** Maximum and minimum value of Y axis is set automatically according to values displayed on chart for best data presentation. ***User*** Maximum and minimum value of Y axis is set by user defined parameters
Default value	-	***Automatic***
Comments	-	N/A

9.4.9.10.11 Minimum value

Function	-	Sets minimum value of Y axis.
Data type	-	Number
Range	-	***-320.00 - 320.00***
Default value	-	***-320.00***
Comments	-	Parameter visible only when Y axis scaling parameter is set to *User*.

9.4.9.10.12 Maximum value

Function	-	Sets maximum value of Y axis.
Data type	-	Number
Range	-	***-320.00 - 320.00***
Default value	-	***-320.00***
Comments	-	Parameter visible only when Y axis scaling parameter is set to *User*.

9.5 Communication ports

Communication ports group contain parameters configuring operation of serial ports, Ethernet and routing data between communication interfaces of device.

9.5.1 Modbus ID - Port 1

Function	-	Defines Modbus ID for internal resources of device on Port 1 (Modbus RTU)
Data type	-	Number
Range	-	***0 - 255***
Default value	-	***1***
Comments	-	setting this value to ***0*** disables access to device resources from serial Port 1

9.5.2 Modbus ID - Port 2

Function	-	Defines Modbus ID for internal resources of device on Port 2 (Modbus RTU)
Data type	-	Number
Range	-	***0 - 255***
Default value	-	***1***
Comments	-	setting this value to ***0*** disables access to device resources from serial Port 2

9.5.3 Modbus ID - Ethernet

Function	-	Defines Modbus ID for internal resources of device on Ethernet port (Modbus TCP)
Data type	-	Number
Range	-	***0 - 255***
Default value	-	***1***
Comments	-	setting this value to ***0*** disables access to device resources from Ethernet port

9.5.4 Modbus ID - GPRS

Function	-	Defines Modbus ID for internal resources of device for polls incoming via GPRS network
Data type	-	Number
Range	-	***0 - 255***
Default value	-	***1***
Comments	-	setting this value to ***0*** disables access to device resources from GPRS network

9.5.5 Port 1/Port 2

Subgroup **Port 1** and **Port 2** contains configuration parameters of RS-232/485 serial Port 1

9.5.5.1 Operating mode

Function - Defines operating mode of serial port Port 1

Data type - Selection list

Range -

Inactive

Serial port is disabled

Transparent

Serial port communication is channeled to other communication port or GPRS network according to rules defined in Transparent routing table. Additional configuration parameters are available in Transparent mode group.

Modbus Master

MT-151 HMI operates as Modbus RTU Master on Port 1 serial port. It can poll for data from and write data to external Slave devices connected to that port using Data blocks. Also polls and writes from external devices communicating with MT-151 HMI can be routed to Port 1 according to rules defined in Modbus routing table. Additional configuration parameters are available in Modbus RTU Master mode group.

Modbus Slave

MT-151 HMI operates as Modbus RTU Slave on Port 1 serial port. External Master device can poll for data from and write data to module.

Flex Serial

Serial port is set to work with devices that have implemented own serial protocols that are not standard in this version of module. Transmission control and data transfer is controlled by internal user program.

M-Bus (PORT 1 only)

Serial port 1 is set to work in M-Bus protocol which allows to read data form heat meters. To proper connection an external converter RM-120 is required.

Default value - ***Inactive***

Comments - N/A

9.5.5.2 Interface type

Function - Defines electrical serial port standard used for communication

Data type - Selection list

Range - ***RS-232***

Half-duplex, 3-wire, ±12 VDC voltage interface. Only one device can be connected to port in this mode.

RS-485

Half-duplex, 2-wire differential interface. Many devices can be connected to port in this mode.

Default value - ***RS-232***

Comments - Available only for operating modes: Transparent, Modbus Master, Modbus Slave, Flex Serial only for **PORT 1**

9.5.5.3 Transmission speed

Function - Defines transmission speed in bits per second

Data type - Selection list

Range - ***1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 [bps]***

Default value - ***9600 [bps]***

Comments - N/A

9.5.5.4 Number of data bits

Function - Defines number of data bits for Flex Serial mode

Data type - Selection list

Range - ***7, 8***

available options list

Default value - ***8***

Comments - 7-bits mode unavailable choose of stop bits.
Parity **"Even"** or **"Odd"** require **1 bit** stop.
Parity **None** require **2 bits** stop. This limits are result of hardware capabilities and setting are selected automatically by module.
Data that are reading/writing are masked up to 7 bits. The oldest is cutting.

9.5.5.5 Stop bits

Function - Defines number of stop bits used during communication

Data type - Selection list

Range - ***1, 2***

Default value - ***1***

Comments - When one of Modbus operating modes is selected this parameter value does not influence communication - number of stop bits is automatically chosen according to Parity setting.

9.5.5.6 Parity

Function	-	Defines parity control of transmitted byte
Data type	-	Selection list
Range	-	***None, Even, Odd***
Default value	-	***None***
Comments	-	When one of Modbus operating modes is selected this parameter overrides Stop bits parameter setting as follows: ***None*** 1 stop bit ***Even*** or ***Odd*** 2 stop bits

9.5.5.7 Modbus

Function	-	Defines selection of Modbus protocol type to use
Data type	-	Selection list
Range	-	***RTU*** ***ASCII*** ***ASCII 8bit*** available option list
Default value	-	***RTU***
Comments	-	Available only for operating modes: Modbus Master, Modbus Slave

9.5.5.8 Transparent mode

In this mode communication on serial Port 1 is channeled to other communication port or GPRS network according to rules defined in Transparent routing table. These group contains additional communication parameters for this mode.

9.5.5.8.1 Max. data packet size

Function	-	Defines maximum size of data packet in bytes
Data type	-	Number
Range	-	***1 - 1408***
Default value	-	***256***
Comments	-	If number of data bytes in receiving buffer reaches declared value, data packet is sent according to rules defined in Transparent routing table.

9.5.5.8.2 Data frame delimiter

Function	-	Defines in seconds minimum interval between receiving data packets
Data type	-	Number
Range	-	***0.00 - 655.35 [s]***
Default value	-	***1.00 [s]***

Comments	-	If no new data arrives to receiving buffer within declared time, data already saved in that buffer is sent according to rules defined in Transparent routing table.

9.5.5.8.3 Channel reservation time

Function	-	Defines in seconds maintain time the transmission channel with external device transmitter.
Data type	-	Number
Range	-	***0.00 - 655.35 [s]***
Default value	-	***0.00 [s]***
Comments	-	N/A

9.5.5.9 Modbus Master mode

In this mode **MT-151 HMI** can poll for data from and write data to external Slave devices connected to that port using Data blocks. Also polls and writes from external devices communicating with **MT-151 HMI** can be routed to PORT 1 according to rules defined in Modbus routing table. This group provides additional configuration parameters for this mode.

9.5.5.9.1 Delay after error in communication w ith Slave

Function	-	Defines in seconds delay between error in communication and next communication for current Data block
Data type	-	Number
Range	-	***0 - 65535 [s]***
Default value	-	***15 [s]***
Comments	-	This time is measured separately for each Data block - error in communication on one block does not influence communication carried out using other Data blocks.

9.5.5.9.2 Number of read/w rite data blocks

Function	-	Defines number of data blocks to define
Data type	-	Number
Range	-	***0 - 16***
Default value	-	***0***
Comments	-	N/A

9.5.5.9.3 Response timeout [s]

Function	-	Defines in seconds maximum waiting answer time of SLAVE device.
Data type	-	Number
Range	-	***1 - 30***
Default value	-	***1***
Comments	-	N/A

9.5.5.9.4 Data blocks (read/w rite)

Data blocks defined in this group allow reading from and writing data to external Modbus RTU Slave devices. Each block is matching group of addresses from one slave device with group of registers in Internal registers address space in device. Data from external devices is polled from external device and written into those registers. Saving new data to those registers either by program or from remote via GPRS or other communication port or method automatically saves this data into Modbus RTU Slave device.

9.5.5.9.4.1 Modbus Slave ID

Function	-	Defines Modbus ID of Slave, which should be polled under this data block
Data type	-	Number
Range	-	***0 - 255***
Default value	-	***1***
Comments	-	setting this value to ***0*** disables data block

9.5.5.9.4.2 Address space in Slave

Function	-	Defines address space of Slave device where from data will be polled
Data type	-	Selection list
Range	-	***Binary Inputs*** Binary inputs (address 1XXX), read only ***Binary Outputs*** Binary outputs (address 0XXX), read/write ***Input Registers*** Input registers (address 3XXX) also known as analog inputs address space, read only ***Holding Registers*** Holding registers (address 4XXX) also known as internal registers and analog outputs address space, read/write
Default value	-	***Binary Inputs***
Comments	-	N/A

9.5.5.9.4.3 Mapped space address - Slave

Function	-	Defines address of first resource (bit or register depending on address space) of data block mapped from Slave to module
Data type	-	Number
Range	-	***0 - 65535***
Default value	-	***0***
Comments	-	setting this value to ***0*** disables data block

9.5.5.9.4.4 Mapped space size

Function	-	Defines number of Slave device addresses (bit or register depending on address space) to be mapped to registers of module
Data type	-	Number
Range	-	***1 - 2040***
Default value	-	***1***
Comments	-	N/A

9.5.5.9.4.5 Mapped space address - Module

Function - Defines address of register in Internal registers address space of module which is mapped to Slave resources defined in data block. If data does not fit within one register (e.g. 17 bits or 2 registers), next register is used as well.

Data type - Number

Range - ***0 - 8191***

Default value - ***1160***

Comments - N/A

9.5.5.9.4.6 Mapped space refresh interval

Function - Defines in seconds interval between polls of Slave resources within data block. Data writes are also executed with this interval

Data type - Number

Range - ***0 - 65535 [s]***

Default value - ***1***

Comments - Entering ***0*** forces communication with maximum possible speed. This speed depends on port communication speed and number of data blocks

9.5.5.10 Flex Serial mode

MT-151 module with active **Flex Serial** mode selected in configuration and external device connected to serial port allows communication in protocol different than Modbus. Whole communication in **Flex Serial** mode is controlled in internal user program. Data transfer between devices is using internal registers that are indicated in memory and operate as two data buffers, read and write. **Flex serial** registers are described also with functional description in chapter Communication Interfaces/Serial ports/Flex Serial

9.5.5.10.1 Max. data packet size [byte]

Function - Defines (in bytes) maximal number of data in package. When sending buffer have declared data number the package is sending.

Data type - Number

Range - ***1 - 256***

Default value - ***128***

Comments - N/A

9.5.5.10.2 Data frame delimiter [s]

Function - define the time (in seconds) between received signs. Exceeding causes sending of received data.

Data type - Number

Range - ***0,00 - 655,35 [s]***

Default value - ***1,00 [s]***

Comments - N/A

9.5.5.11 M-Bus mode

M-Bus mode allows reading data from 16 external devices connected to PORT 1 in this protocol. Read data are stored in module internal memory, ready to analyze or transmit according to settings that are configured by user. Proper operation in **M-Bus** protocol require electricity converter **RM-120** available in additional telemetry module accessories.

9.5.5.11.1 Transmission speed [bod]

Function	-	sets transmission speed for M-Bus mode on Port 1
Data type	-	Number
Range	-	***300, 600, 1200, 2400, 4800, 9600 [bps]*** supported speed values list
Default value	-	***9600***
Comments	-	N/A

9.5.5.11.2 Number of device

Function	-	defines number of external devices for pooling
Data type	-	Number
Range	-	***1 - 16***
Default value	-	***1***
Comments	-	N/A

9.5.5.11.3 Addressing

Function	-	selection of an addressing type for devices connected in M-bus.
Data type	-	Number
Range	-	***Unicast*** addressing based on real address of the device or logical addressing based on serial number of the device ***Broadcast*** addressing without knowledge of real address of the device. Only one device can be operated. MT module use global address (254) in Broadcast frame.
Default value	-	***Unicast***
Comments	-	N/A

9.5.5.11.4 M-BUS device (DEVx)

Group of parameters described request rules of devices that is read by **MT-151** with **M-Bus** protocol. Each device can have separately configuration of addresses, resources and variables for reading. In case of Broadcast addressing, only one position is on the list. **Unicast** addressing allows connection up to 16 external devices.

9.5.5.11.4.1 Name

Function	-	friendly name of connected **M-Bus** device
Data type	-	text
Range	-	letters and number, Max. 31 characters
Default value	-	***DEVX***

Comments - N/A

9.5.5.11.4.2 Addressing method

Function - allows to decide about devices addressing type by directly changing content of field A in M-Bus communication frame.

Data type - selection list

Range - ***Primary***

Address entered in **Address** parameter is used for pooling

Secondary

Serial number entered in Serial number parameter is used for pooling. Communication frames are sends for address 253.

Default value - ***Primary***

Comments - Lo-Hi byte order is used for setting **Secondary**

9.5.5.11.4.3 Address

Function - physical address that identify M-Bus device connected to module.

Data type - Number

Range - ***1 - 253***

Default value - ***1***

Comments - N/A

9.5.5.11.4.4 Serial number

Function - serial number that identify M-Bus device connected to module.

Data type - Number

Range - ***1 - 99999999***

Default value - ***1***

Comments - N/A

9.5.5.11.4.5 Restart communication before reading

Function - allows to sends SND_NKE telegram before start reading the data after communication restart

Data type - selection list

Range - ***Yes***

Telegram is sent before read

No

Telegram isn't sent before read

Default value - ***Tak***

Comments - this parameter is required in some devices to stop sending archive data records (Storage>0)

9.5.5.11.4.6 Bytes order

Function - selection of byte order for **Secondary** addressing

Data type - selection list

Range - ***Mode 1 (Lo-Hi)***

Mode 2 (Hi-Lo)

Default value - ***Mode 1***

Comments - N/A

9.5.5.11.4.7 Pooling interval [min]

Function - time interval between next queries in minutes

Data type - number

Range - ***1 - 60***

Default value - ***1***

Comments - N/A

9.5.5.11.4.8 Number of variables

Function - defines number of variable that will be read from connected device

Data type - number

Range - ***1 - 16***

Default value - ***1***

Comments - N/A

9.5.5.11.4.9 Variable table

Idx. - Index number

Value - selection of quantities which can be sent form device

Range:

Energy,Volume, Mass, Volume flow, Mass flow, Flow temperature, Return temperature, Temperature difference, External temperature, Pressure, Power, Operating time, On time, Extended(VIFE)

Unit/VIFE - selection of unit for quantity that will be sent (value is scaled automatically)

Range:

physical quantities units

selection **Extended(VIFE)** allows to enter HEX value with 4 - 16 number of signs for identify of quantity which will be comparison with VIFE field from information data blocks that comes from M-Bus device. Equality of both values is a condition to fill register selected from HREG space. VIFE field can store more than 1 byte, and besides values from standard specification can store special defined values by manufacturer of device. MT module can mapped any 2bytes/4signs extended value. Other parameter as Tariff and Storage are still used in mapping. Value captured by VIFE is not scaled so proper interpretation is required by user itself.

Range:

0000 - FFFFFFFFFFFFFFFF

Format - format type for stored data in registers
Range:
Float (32 bits)**, Int32**

Register address - address from holding register space HREG for data storage sent from device
Range:
0 - 8000

Logical unit - defines value for parameter Logical
Range:
0 - 255

Tariff - defines value for parameter Tariff
Range:
0 - 16

Storage - defines value for parameter Storage
Range:
0 - 255

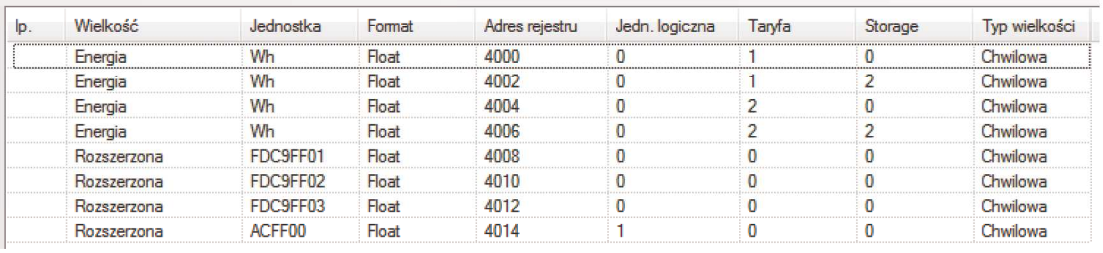
Type - selection of data kind
Range:
Instantaneous - actual read without error
Min, Max
During error - read when error occurred

Example frame from energy meter:

```
mbus: Data: Energy  3.710  kWh, unit=0, stor=0, tar=1,
func=0 -> dev var 1
mbus: Data: Energy  3.710  kWh, unit=0, stor=2, tar=1,
func=0 -> dev var 2
mbus: Data: Energy  0.770  kWh, unit=0, stor=0, tar=2,
func=0 -> dev var 3
mbus: Data: Energy  0.770  kWh, unit=0, stor=2, tar=2,
func=0 -> dev var 4
mbus: Data: ExtVIF [FDC9FF01]  223.000  , unit=0,
stor=0, tar=0, func=0 -> dev var 5
mbus: Data: ExtVIF [FDDBFF01]  0.000  , unit=0, stor=0,
tar=0, func=0
mbus: Data: ExtVIF [ACFF01]  0.000  , unit=0, stor=0,
tar=0, func=0
mbus: Data: ExtVIF [ACFF01]  0.000  , unit=1, stor=0,
tar=0, func=0
mbus: Data: ExtVIF [FDC9FF02]  224.000  , unit=0,
stor=0, tar=0, func=0 -> dev var 6
mbus: Data: ExtVIF [FDDBFF02]  0.000  , unit=0, stor=0,
tar=0, func=0
mbus: Data: ExtVIF [ACFF02]  0.000  , unit=0, stor=0,
tar=0, func=0
mbus: Data: ExtVIF [ACFF02]  0.000  , unit=1, stor=0,
tar=0, func=0
mbus: Data: ExtVIF [FDC9FF03]  223.000  , unit=0,
stor=0, tar=0, func=0 -> dev var 7
mbus: Data: ExtVIF [FDDBFF03]  0.000  , unit=0, stor=0,
tar=0, func=0
```

```
mbus: Data: ExtVIF [ACFF03]  0.000  , unit=0, stor=0,
tar=0, func=0
mbus: Data: ExtVIF [ACFF03]  0.000  , unit=1, stor=0,
tar=0, func=0
mbus: Data: ExtVIF [FF68]  0.000  , unit=0, stor=0,
tar=0, func=0
mbus: Data: ExtVIF [ACFF00]  0.000  , unit=0, stor=0,
tar=0, func=0
mbus: Data: ExtVIF [ACFF00]  0.000  , unit=1, stor=0,
tar=0, func=0 -> dev var 8
mbus: Data: ExtVIF [FF13]  0.000  , unit=0, stor=0,
tar=0, func=0
```

and MTM configuration that mapped M-Bus values to registers:



lp.	Wielkość	Jednostka	Format	Adres rejestru	Jedn. logiczna	Taryfa	Storage	Typ wielkości
	Energia	Wh	Float	4000	0	1	0	Chwilowa
	Energia	Wh	Float	4002	0	1	2	Chwilowa
	Energia	Wh	Float	4004	0	2	0	Chwilowa
	Energia	Wh	Float	4006	0	2	2	Chwilowa
	Rozszerzona	FDC9FF01	Float	4008	0	0	0	Chwilowa
	Rozszerzona	FDC9FF02	Float	4010	0	0	0	Chwilowa
	Rozszerzona	FDC9FF03	Float	4012	0	0	0	Chwilowa
	Rozszerzona	ACFF00	Float	4014	1	0	0	Chwilowa

In this frame some values are repeated each time for VIFE and for typical quantities (energy and VIFE: ACFF00) - selection must be specified using additional parameters like **Logical unit** and **Tariff**.

9.5.6 Ethernet

Subgroup **Ethernet** contains parameters configuring operation of Ethernet port.

9.5.6.1 Use of Ethernet

Function - Enables communication via Ethernet port

Data type - Selection list

Range - ***No***

Ethernet port is disabled

Yes

Ethernet port is enabled.

Default value - ***No***

Comments - **MT-151 HMI** operates on Ethernet port as Server - it allows remote connection from clients which then can poll for data or write to device. When needed module can connects to server as an client and trying to get the data according to Modbus TCP Client data blocks or can transmitting incoming data according to routing tables.

9.5.6.2 Ethernet port speed

Function - Enables impose concrete speed on Ethernet port.

Data type - Selection list

Range - ***Auto***

Port speed is negotiated automatically

10 Mb/s

Port speed is 10 Mb/s

		100 Mb/s Port speed is 100 Mb/s
Default value	-	***Auto***
Comments	-	N/A

9.5.6.3 Sender IP address control

Function	-	Switches the control of sender's IP address on/off
Data type	-	Selection list
Range	-	***Yes*** The module exchanges information only with IP address present on the Authorized IP addresses list. ***No*** The module exchanges information (configuration, responses for queries) with any IP address sending qualified query or command. In this case the identification of the sender goes by its current identifier.
Default value	-	***Yes***
Comments	-	Switching the control off enables verification of the sender in the base of its currently assigned identifier other than IP address (e.g. serial number). This allows communication among units with dynamically assigned IP addresses (within same APN). Sender's identifier must reside on Authorized IP addresses list in order to establish the communication.

9.5.6.4 IP address

Function	-	Enables configuration of IP address of module used on Ethernet
Data type	-	IP address
Range	-	***0.0.0.0 - 255.255.255.255***
Default value	-	***0.0.0.0***
Comments	-	N/A

9.5.6.5 Subnet mask

Function	-	Allows to enter IP mask defining subnet used by module
Data type	-	IP mask
Range	-	***0.0.0.0 - 255.255.255.255***
Default value	-	***0.0.0.0***
Comments	-	N/A

9.5.6.6 Default gateway

Function	-	Enables configuration of IP address of default Ethernet gateway
Data type	-	IP address

Range - ***0.0.0.0 - 255.255.255.255***

Default value - ***0.0.0.0***

Comments - N/A

9.5.6.7 IP routing table entry count

Function - Sets quantity of numbers that are allowed in Routing table

Data type - Number

Range - ***0 - 8***

Default value - ***0***

Comments - N/A

9.5.6.8 Routing IP

Idx. - Index number

Subnet - Defines subnet addresses included in one network area.

Mask - Defines range of authorized IP addresses.

Gateway - Defines IP gateway number for entered **Subnet**

9.5.6.9 Authorized IP addresses

Authorized numbers comprises lists of Ethernet IP addresses the module can communicate with.

9.5.6.9.1 Number of IP addresses

Function - Defines the length of the IP addresses' list allowed to communicate with device via Ethernet

Data type - Number

Range - ***0 - 16***

Default value - ***0***

Comments - N/A

9.5.6.9.2 IP

Idx. - Index number

Name - Friendly name facilitating identification of device. Max. length is 16 characters.

IP address - IP address assigned to Ethernet Device

Protocol ***UDP***

Communication is carried out using UDP protocol

TCP

Communication is carried out using TCP protocol

Configuration Value of this parameter determines whether remote configuration data arriving from selected IP will be ignored or accepted
Default value: ✓ (allowed)

Receiving Value of this parameter determines whether data arriving from selected IP will be accepted or ignored
Default value: ✓ (allowed)

SNMP Query	Value of this parameter determines whether SNMP request arriving from selected IP will be accepted or ignored **Default value:** ✓ (allowed)

9.5.6.10 Modbus TCP Client

In this group **MT-151 HMI** can poll for data from and write data to external Modbus TCP Slave devices connected to Ethernet port using Data blocks. Polls and writes from external devices communicating with MT-151 HMI can also be routed to Ethernet according to rules defined in Modbus routing table.

9.5.6.10.1 Delay after error in communication with Server

Function - Defines in seconds delay between error in communication and next communication for current Data block

Data type - Number

Range - ***0 - 65535 [s]***

Default value - ***15 [s]***

Comments - This time is measured separately for each Data block - error in communication on one block does not influence communication carried out using other Data blocks.

0.25 firmware version and higher has got an algorithm for skip request not active devices and don't block the reading of active units. Two types are distinguish:

1. device doesn't answer for requests but answer for PING - Another connection attempts are executed with higher duration of time, in sequence 1x, 2x, 3x, 8x, 16x more than entered value.
2. device doesn't answer for request and for PING - Another connection attempts are executed with 10x more than entered value.

Additionally duration of connection attempt is limited to 30 seconds, after this, next device is requested. Connection error with active device causes one request loss.

9.5.6.10.2 Number of read/write data blocks

Function - Defines number of data blocks to define

Data type - Number

Range - ***0 - 16***

Default value - ***0***

Comments - N/A

9.5.6.10.3 Response timeout

Function - Defines in seconds maximum waiting answer time of TCP server device.

Data type - Number

Range - ***1 - 30***

Default value - ***1***

Comments - N/A

9.5.6.10.4 Ethernet IP

Function - Allows to choose IP address of Modbus TCP Server device

Data type - Selection list

Range - ***None*** or one of Names defined on Authorized IP list for Ethernet

Default value - ***None***

Comments - N/A

9.5.6.10.5 Server Modbus ID

Function - Defines Modbus ID of TCP server polling using prepared data block

Data type - Number

Range - ***0 - 250***

Default value - ***0***

Comments - N/A

9.5.6.10.6 Address space in Server

Function - Defines address space of Modbus TCP Slave device where from data will be polled

Data type - Selection list

Range - ***Binary Inputs***

Binary inputs (address 1XXX), read only

Binary Outputs

Binary outputs (address 0XXX), read/write

Input Registers

Input registers (address 3XXX) also known as analog inputs address space, read only

Holding Registers

Holding registers (address 4XXX) also known as internal registers and analog outputs address space, read/write

Default value - ***Binary Inputs***

Comments - N/A

9.5.6.10.7 Mapped space address - Server

Function - Defines address of first resource (bit or register depending on address space) of data block mapped from Modbus TCP Server to module

Data type - Number

Range - ***0 - 65535***

Default value - ***0***

Comments - setting this value to ***0*** disables data block

9.5.6.10.8 Mapped space size

Function - Defines number of Modbus TCP Server device addresses (bit or register depending on address space) to be mapped to registers of module

Data type - Number

Range - ***1 - 2040***

Default value - ***1***

Comments - One register count 16-bits:
Example of mapping bits:
Mapped space address - Module: *116*
Mapped space address - Server TCP: *0*
Mapped space size: *10*
16 bits form Server goes to register 116

9.5.6.10.9 Mapped space address - Module

Function - Defines address of register in Internal registers address space of module which is mapped to Modbus TCP Server resources defined in data block. If data does not fit within one register (e.g. 17 bits or 2 registers), next register is used as well.

Data type - Number

Range - ***0 - 8191***

Default value - ***116***

Comments - N/A

9.5.6.10.10 Mapped space refresh interval [s]

Function - Defines in seconds interval between polls of Server resources within data block. Data writes are also executed with this interval

Data type - Number

Range - ***0 - 65535 [s]***

Default value - ***10***

Comments - Entering ***0*** forces communication with maximum possible speed.

9.5.7 Routing tables

Routing tables group consists of tables defining data routing rules between Port 1, Port 2, Ethernet and GPRS. There are different tables for different protocols.

9.5.7.1 Number of Modbus routing table rules

Function - Defines the length of the Modbus routing table

Data type - Number

Range - ***0 - 16***

Default value - ***0***

Comments - N/A

9.5.7.2 Number of Transparent routing table rules

Function - Defines the length of the Transparent routing table

Data type - Number

Range - ***0 - 4***

Default value - ***0***

Comments - N/A

9.5.7.3 Modbus routing table

Idx. - Index number

Name - Friendly name facilitating identification of routing rule purpose. Max. length is 31 characters.

Interface - ***None***

Routing rule is disabled.

Port 1

Modbus RTU Slave device is connected to Port 1. Option available only when Port 1 operates in Modbus RTU Master mode.

Port 2

Modbus RTU Slave device is connected to Port 2. Option available only when Port 2 operates in Modbus RTU Master mode.

Ethernet

Modbus TCP Server device is connected to Ethernet port. Option available only when Ethernet is turned on.

GPRS

Modbus TCP Server/RTU Slave device is connected to MT telemetry module logged into GPRS. Option available only when GPRS is turned on.

Port1 ID - ID of Modbus RTU Slave device as seen from Port 1
1 - 255

Port2 ID - ID of Modbus RTU Slave device as seen from Port 2
1 - 255

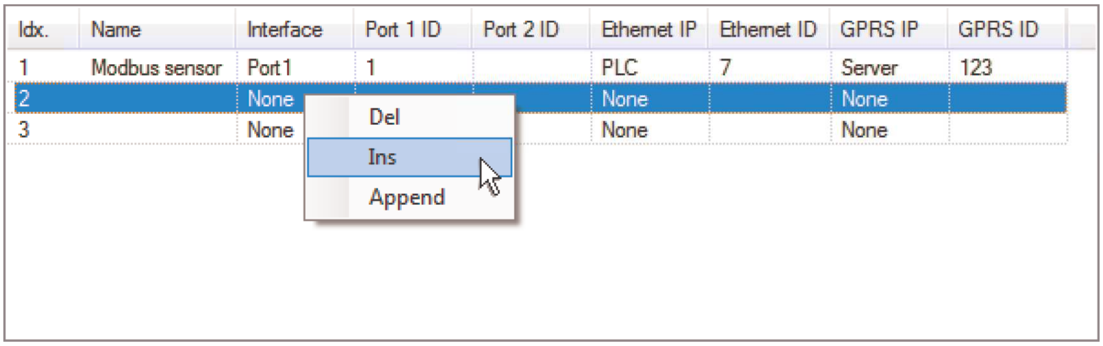
Ethernet IP - IP of Modbus TCP Server device as seen from Ethernet
0.0.0.0 - 255.255.255.255

Ethernet ID ID of Modbus TCP Server device as seen from Ethernet
1 - 255

GPRS IP IP of MT telemetry module as seen from GPRS
0.0.0.0 - 255.255.255.255

GPRS ID ID of MT telemetry module or device connected to it as seen from GPRS
1 - 255

Entries on this list may be easily added and deleted by using context menu activated by right mouse button click on any position of the list in parameters window.



Idx.	Name	Interface	Port 1 ID	Port 2 ID	Ethernet IP	Ethernet ID	GPRS IP	GPRS ID
1	Modbus sensor	Port1	1		PLC	7	Server	123
2		None			None		None	
3		None			None		None	

Modbus sensor from picture above is connected to Port 1 serial port and its Modbus ID is 1. It can be accessed from Ethernet by PLC - it should poll MT-151 HMI using Modbus TCP protocol for ID 7. This poll will be automatically translated to Modbus RTU and send to Port 1 with ID 1. This sensor can also be accessed from GPRS by Server. It should poll for ID 123 to get access to Modbus sensor.

9.5.7.4 Transparent routing table

Idx. - Index number

Name - Friendly name facilitating identification of routing rule purpose. Max. length is 31 characters.

Interface A - ***None***

Routing rule is disabled.

Port 1

All communication from Port 1 is routed to Interface B. Option visible only when Port 1 operating mode is set to Transparent.

Port 2

All communication from Port 2 is routed to Interface B. Option visible only when Port 2 operating mode is set to Transparent.

Ethernet

All communication from Ethernet IP given in next column is routed to Interface B.

GPRS

All communication from GPRS IP given in next column is routed to Interface B.

IP address A - IP address for Interface A. Parameter valid only for GPRS and Ethernet interfaces.

Interface B - ***None***

Routing rule is disabled.

Port 1

All communication from Port 1 is routed to Interface A. Option visible only when Port 1 operating mode is set to Transparent.

Port 2

All communication from Port 2 is routed to Interface A. Option visible only when Port 2 operating mode is set to Transparent.

Ethernet
All communication from Ethernet IP given in next column is routed to Interface A.
GPRS
All communication from GPRS IP given in next column is routed to Interface A.

IP address B - IP address for Interface B. Parameter valid only for GPRS and Ethernet interfaces.

9.6 Communication

Communication group contains parameters managing SMS and spontaneous data transmission. Here you can configure when transmission will be triggered, what data or message it will send and where it shall be send.

9.6.1 MT2MT buffer

MT2MT buffer enables creation of system where MT modules may exchange information (Holding Registers) with each other without any relaying instance. Data transmission from one module to other is carried out by sending from one device group of Holding Registers with data to other device which has turned on MT2MT buffer functionality and defined MT2MT buffer which includes register addresses send from sending device. Data send by sender is saved to registers within buffer with same address. Each time new data arrives MT2MT_x a Bit is set, where x is position of sender IP address on receiver authorized IP list.

9.6.1.1 Active

Function - Enables receiving GPRS frames to MT2MT buffer

Data type - Selection list

Range - ***No***
MT2MT buffer functionality is disabled
Yes
MT2MT buffer functionality is enabled

Default value - ***No***

Comments - When set to ***No*** module cannot receive GPRS frames to buffer, however it still can send data to other buffers.
GPRS is required for MT2MT communication.

9.6.1.2 Buffer address

Function - Defines address of register from Holding Registers address space where buffer begins

Data type - Number

Range - ***0 - 8191***

Default value - ***116***

Comments - Received data which does not fit within defined buffer is not saved in module.

9.6.1.3 Buffer size

Function - Defines number of registers from Holding Registers from which MT2MT buffer consist

Data type - Number

Range - ***1 - 700***

Default value - ***16***

Comments - Received data which does not fit within defined buffer is not saved in module.

9.6.2 Logger

Events subgroup contains parameters controlling logger functionality.

9.6.2.1 Primary transmission channel

Function - Defines primary transmission channel for logger data.

Data type - Selection list

Range - ***GPRS***

Logger records are sent using GPRS packet transmission interface.

Ethernet

Logger records are sent using Ethernet interface.

Default value - ***GPRS***

Comments - N/A

9.6.2.2 Primary Recipient

Function - Defines IP address which shall receive logger data frames

Data type - Selection list

Range - ***None*** and addresses defined in GSM -> Authorized numbers -> IP list for GPRS transmission

Default value - ***None***

Comments - N/A

9.6.2.3 Alternative transmission channel

Function - Defines alternative transmission channel for logger data.

Data type - Selection list

Range - ***GPRS***

Logger records are sent using GPRS packet transmission interface.

Ethernet

Logger records are sent using Ethernet interface.

Default value - ***GPRS***

Comments - N/A

9.6.2.4 Alternative Recipient

Function - Defines IP address which shall receive logger data frames when Primary Recipient is unavailable

Data type - Selection list

Range - ***None*** and addresses defined in GSM -> Authorized numbers -> IP list for GPRS transmission

Default value	-	***None***
Comments	-	N/A

9.6.2.5 Recipient UDP port

Function	-	Defines UDP port to which the logger shall be sent
Data type	-	Number
Range	-	***1024 - 65535***
Default value	-	***7110***
Comments	-	N/A

9.6.2.6 Number of logger data blocks

Function	-	Defines the length of the Logger data blocks table
Data type	-	Number
Range	-	***0 - 4***
Default value	-	***0***
Comments	-	N/A

9.6.2.7 Logger data block table

Idx. - Index number

Name - Friendly name facilitating identification of data blocks purpose. Max. length is 16 characters.

Address space - Defines address space of data block

Input Registers

Input registers (address 3XXX) also known as analog inputs address space, read only

Holding Registers

Holding registers (address 4XXX) also known as internal registers and analog outputs address space, read/write

Data block address - Defines address of register from which data block begins

0 - 8191

Data block size Defines number of registers which are in data block

1 - 28

9.6.3 Events

Events subgroup contains definitions of changes of binary resources states (flags, inputs, outputs, bits) which then can be used for triggering SMS and data sending in Rules and also trigger record saving and logger transmission.

9.6.3.1 Number of events

Function	-	Defines the length of the Event table
Data type	-	Number
Range	-	***0 - 32***
Default value	-	***0***
Comments	-	N/A

9.6.3.2 Event table

Idx. - Index number

Name - Friendly name facilitating identification of event purpose. Max. length is 16 characters.

Triggering bit - Defines the bit which state change will trigger event
0 - 65535 or name from bit list (see bit list in Appendices)
For binary output space is required to enter prefix 10xxx before address, for flag P1 value of triggered bit is 11600 (P1 bit is address 1600[dec])

Triggering edge - ***0->1***
Trigger event on rising edge.
1->0
Trigger event on falling edge.
0<->1
Trigger event on any edge.

Write data blocks to logger - Toggles saving logger data blocks to logger as new record on/off on occurring event.
Default value is ✗ (off).

Trigger logger sending - Toggles sending the logger content on/off on occurring event.
Default value is ✗ (off).

9.6.4 Data blocks

Data blocks subgroup contains definitions of data which then can be used send using Rules.

9.6.4.1 Number of data blocks

Function - Defines the length of the Data block table

Data type - Number

Range - ***0 - 32***

Default value - ***0***

Comments - N/A

9.6.4.2 Data block table

Idx. - Index number

Name - Friendly name facilitating identification of data blocks purpose. Max. length is 16 characters.

Address space - Defines address space of data block
Input Registers
Input registers (address 3XXX) also known as analog inputs address space, read only
Holding Registers
Holding registers (address 4XXX) also known as internal registers and analog outputs address space, read/write

Data block address - Defines address of register from which data block begins
0 - 8191

Data block size	Defines number of registers which are in data block ***1 - 256***

9.6.5 Rules

Rules subgroup consists of list of communication rules allowing to send SMS messages or send selected Data blocks via GPRS when selected event occurs.

9.6.5.1 Number of rules

Function	-	Defines number of Rules to define
Data type	-	Number
Range	-	***0 - 32***
Default value	-	***0***
Comments	-	N/A

9.6.5.2 Rule

9.6.5.2.1 Name

Function	-	Friendly name facilitating identification of the rule
Data type	-	Text
Range	-	Letters and numerals - max. 31 characters
Default value	-	Respectively from ***RULE1*** to ***RULE32***
Comments	-	N/A

9.6.5.2.2 Triggering event

Function	-	Defines event which triggers transmission
Data type	-	Selection list
Range	-	***None*** and events defined in Event table
Default value	-	***None***
Comments	-	N/A

9.6.5.2.3 Transmission type

Function	-	Defines transmission type
Data type	-	Selection list
Range	-	***None*** Rule is disabled ***SMS*** Rule triggers sending SMS message ***GPRS*** Rule triggers sending data using GPRS ***Ethernet*** Rule triggers sending data using Ethernet interface
Default value	-	***None***
Comments	-	SMS, Ethernet and GPRS options are visible only when those methods of communication are enabled

9.6.5.2.4 Receiver

Function	-	Defines receiver of SMS or data package (depends on Transmission type setting)
Data type	-	Selection list
Range	-	***None*** and numbers defined in GSM -> Authorized numbers -> Phone list for SMS transmission ***None*** and addresses defined in GSM -> Authorized numbers -> IP list for GPRS transmission
Default value	-	***None***
Comments	-	N/A

9.6.5.2.5 SMS text

Function	-	Allows to enter text which will be send as SMS
Data type	-	Text
Range	-	Letters, numerals and special characters - max. 160 characters
Default value	-	N/A
Comments	-	It is possible to add to SMS text macros, symbolic names and special mnemonics dynamically changed for values from module, e.g. analog input value or binary input state. Description of this mnemonics can be found in SMS commands syntax chapter in Appendices. Parameter is visible only when Transmission type parameter is set to SMS.

9.6.5.2.6 Data block

Function	-	Defines data block which is sent via GPRS by rule
Data type	-	Selection list
Range	-	***None*** and events defined in Data block table
Default value	-	***None***
Comments	-	Parameter is visible only when Transmission type parameter is set to GPRS.

9.6.6 SNMP

Subgroup **SNMP** contains a list of transmission tasks performed by the module using SNMP (Supports Traps and Requests).

9.6.6.1 Community string - read

Function	-	Password required to access module resources. Read-only access
Data type	-	Text
Range	-	Letters and numbers, max. 31 characters
Default value	-	***public***
Comments	-	N/A

9.6.6.2 Community string - read/write

Function	-	Password required to access module resources. Read/write access

Data type - Text
Range - Letters and numbers, max. 31 characters
Default value - ***private***
Comments - N/A

9.6.6.3 Trap handling

Function - Enables or disables traps sending feature.
Data type - Selection list
Range - ***No***
Traps handling disabled
Yes
Traps handling enabled
Default value - ***No***
Comments - N/A

9.6.6.4 Request handling

Function - Enables or disables requests sending feature.
Data type - Selection list
Range - ***No***
Requests handling disabled
Yes
Requests handling enabled
Default value - ***No***
Comments - N/A

9.6.6.5 Traps

9.6.6.5.1 Number of trap receivers

Function - Defines number of trap receivers (max. 4) IP addresses of receivers can be added in Trap receivers list available when this parameter is > 0.
Data type - Number
Range - ***0 - 4***
Default value - ***0***
Comments - N/A

9.6.6.5.2 Number of traps

Function - Defines number of trap visible to define in Trap table.
Data type - Number
Range - ***0 - 32***
Default value - ***0***
Comments - N/A

9.6.6.5.3 Trap data source

Function - Allows to choose whether data added to traps is defined in configuration or loaded from registers
Data type - Selection list
Range - ***Registers***

Add Holding registers as data source. (HR1024 ... HR1027)

Configuration

Data source is defined in Trap table configuration. Registers (HR1024 ... HR1027) are still allocated to SNMP feature.

Default value - ***Registers***

Comments - N/A

9.6.6.5.4 Trap receivers

Parameter - Friendly name facilitating identification of IP receivers in SNMP traps definitions. Max. length is 20 characters.

Value - IP number

9.6.6.5.5 Trap table

Idx. - Index number

Specific ID - Provides to server information about type of trap event. Basic information for trap meaning distinction by SNMP server. User can set any value from range: ***0 - 65535***

MIB file provided by Inventia lists several values of parameter:

10 analog input alarm activated

11 new analog input measurement

20 synchronous/asynchronous timer reached its threshold

30 binary input state change

31 binary input state readout

40 counter overflow

Triggering bit - Allow selecting a marker or a predefined bit, will be send after the trap changed.

Triggering edge - Allow selecting an edge of triggers data trap transmission (0->1, 1->0, 0<->1)

Trap name - Defines text which will be sent in trap as **trapSourceName** variable. It should provide information about trap source, e.g. binary input name. Entered text is parsed as SNCS command syntax (excluding macros). This includes using #TXT(x) mnemonic, which inserts constant text string from x position in Text list. X can be addressed indirectly using register value e.g. #TXT(HR116) will point to 1 position on Text list providing that register located on address 116 in holding registers address space is 1.

Status - Defines value which is sent as **trapSourceStatus** variable. User can set any value from range: ***0 - 65535***

MIB file provided by Inventia lists several values of parameter:

1	unknown (noStatus)
2	normal
3	alarm Hi – (highWarning)
4	alarm HiHi – (highCritical)
5	alarm Lo – (lowWarning)
6	alarm LoLo – (lowCritical)
7	timer reached its threshold (timeExpired)
8	ON (turnON)
9	OFF (turnOFF)
10	counter overflow (countOverflow)

If Trap data source parameter is set to *Registers* then Status column is not visible and **trapSourceStatus** variable value is copied from register HR1030+5*(trap_index-1).

Value - Defines value which is sent as **trapSourceValue** variable. Value can be enters directly or can be loaded from internal registers. User can set any value from range: ***0 - 65535***

Possible register syntax:

IRxxxx	value of Input Register address xxxx
HRxxx	value of Holding Register address xxx
IBxxx	value of Binary Input address xxx
HBxxx	value of Binary Output address xxx

If Trap data source parameter is set to *Registers* then Value column is not visible and **trapSourceValue** variable value is copied from register HR1031+5*(trap_index-1).

Type - Defines value which is sent as **trapSourceType** variable. User can set any value from range: ***0 - 65535***

MIB file provided by Inventia lists several values of parameter:

1	keep alive (keepAlive)
2	local input (localInput)
3	external input (extInput)
4	voltage analog input (analogVoltage)
5	current analog input (analogCurrent)
6	synchronous timer (timerSync)
7	asynchronous timer (timerAsunc)
8	counter (counter)
9	powering voltage (supplyVoltage)

If Trap data source parameter is set to *Registers* then Type column is not visible and **trapSourceType** variable value is copied from register HR1032+5*(trap_index-1).

Index - Defines value which is sent as **trapSourceIndex** variable. This value should provide information about input channel (e.g. 2 for voltage analog input AV2) or binary input number thus allowing to strictly determining source of trap. User can set any value from range: ***0 - 65535***

If Trap data source parameter is set to *Registers* then Index column is not visible **trapSourceIndex** variable value is copied from register HR1033+5*(trap_index-1).

9.6.6.6 Requests

9.6.6.6.1 Number of request receivers

Function - Defines number of trap receivers (max. 16) IP addresses of receivers can be added in Request receivers list available when this parameter is > 0.

Data type - Number

Range - ***0 - 16***

Default value - ***0***

Comments - N/A

9.6.6.6.2 Request count

Function - Defines number of request visible to define in Request table.

Data type - Number

Range - ***0 - 32***

Default value - ***0***

Comments - N/A

9.6.6.6.3 Request receivers

Idx. - Index number

Name - Friendly name facilitating identification of IP receivers in SNMP Request definitions. Max. length is 20 characters.

Address IP - IP number

9.6.6.6.4 Request table

9.6.6.6.4.1 Triggering bit

Function - Defines marker or bit which triggers transmission request

Data type - Selection list

Range - ***None*** *or* bit number ***0 - 65535*** or name from *bit list*
Declaring bit from Binary input space require add prefix 10xxx to address value of the digital bit (e.g. flag P1 address is 1600[dec] means triggering bit 11600 address)

Default value - ***None***

Comments - N/A

9.6.6.6.4.2 Triggering slope

Function - Defines edge of incrementing bit triggering transmission of the request

Data type - Selection list

Range - ***0->1***
logical state change from 0 to 1
1->0
logical state change from 1 to 0
0<->1
both direction changes

Default value - ***0->1***

Comments - N/A

9.6.6.6.4.3 Receiver address

Function - Allows recipient selection from list of Request receivers

Data type - Selection list

Range - ***None*** or numbers defined in Request receivers

Default value - ***None***

Comments - N/A

9.6.6.6.4.4 OID

Function - Allows entering variable name Object ID for reading in prepared request. OID is entry in ASN.1 notation. OID should be available in MIB files that can be distributed with SNMP devices Module MT supports only numerical 32-bits variables in answers (Integer, TimeTicks, Gauge, Counter).

Data type - Text

Range - ***Numbers*** and ***dots***, max. 64 characters, max. 15 levels

Default value - ***None***

Comments - N/A

9.6.6.6.4.5 Destination register address

Function - Defines first register address in holding space that are stored low 2bytes value of read variable. High 2 bytes are stored in next one register (In default HR1025)

Data type - Number

Range - ***1024 - 8192***

Default value - ***1024***

Comments - N/A

9.6.6.6.4.6 Read flag

Function - Allows to choose, from defined list, the marker that will be set after receiving an correct answer

		and saving the result of request in destination register
Data type	-	Selection list
Range	-	***None*** None selected flag ***P1 - P256*** Available marker, can be use for programming
Default value	-	***None***
Comments	-	Used marker is not automatically reset and requires programming reset. One cycle of the internal program is recommended to delay a reset function of the marker after it was set. Reset at the same cycle of the program isn't able to trigger a request.

9.6.6.6.4.7 Error flag

Function	-	Allows to choose, from defined list, the marker that will be set after receive an error answer or error code answer (Non-existent variable) No answer is not signaled.
Data type	-	Selection list
Range	-	***None*** None selected flag ***P1 - P256*** Available marker, can be use for programming
Default value	-	***None***
Comments	-	Used marker is not automatically reset and requires programming reset. One cycle of the internal program is recommended to delay a reset function of the marker after it was set. Reset at the same cycle of the program could not able to trigger a request.

9.6.7 Spooler

Spooler is a service that can be installed during MTManager installation. Spooler is designed mostly to remote management of the battery modules that have sleep functions activated all the time. Using Spooler user can change the configuration settings of the module including the new firmware version uploading. Devices that are logged into GSM provider all the time can be managed too. The most important features are module's time synchronization and possibility for group management of the modules in the same time.

Spooler requires IIS (Internet Information Services) installed in the Windows OS system.

9.6.7.1 Triggering event

Function	-	Defines event which triggers transmission a notification to **Spooler** service.
Data type	-	Selection list
Range	-	***None*** and events defined in Event table

Default value - ***None***

Comments - N/A

9.6.7.2 Transmission channel

Function - Defines transmission channel for spooler request.

Data type - Selection list

Range - ***GPRS***

Spooler request is sent via GPRS packet transmission interface.

Ethernet

Spooler request is sent via Ethernet interface.

Default value - ***GPRS***

Comments - N/A

9.6.7.3 Address

Function - Defines the IP address of the computer running MTSpooler service.

Data type - List of choices

Range - List of authorized IP addresses

Default value - ***None***

Comments - N/A

9.6.8 IEC 60870-5-104

Subgroup **IEC 60870-5-104** contents configuration parameters responsible for communication in protocol as a name of subgroup. IEC works on GPRS and Ethernet interfaces. Module has a Slave role and listening on port **2404.** Only one Client can be connected at the same time. Full description of using is available in chapter Interfaces and communication methods. Parameters described below are necessary for proper operation with external devices on IEC protocol. Some of them has got names taken straight from the technical specification of the IEC. If description that is written in manual is not enough, please learn more in source documentation of the IEC.

9.6.8.1 Common address

Function - IEC server address, module identify number is IEC network

Data type - Number

Range - ***1 - 254***

Default value - ***1***

Comments - N/A

9.6.8.2 T1 [s]

Function - T1 timeout - time limit for test execution or data sending

Data type - Number

Range - ***1 - 254***

Default value - ***15***

Comments - N/A

9.6.8.3 T2 [s]

Function	-	T2 timeout - time limit for confirmation in case no data
Data type	-	Number
Range	-	***1 - 30***
Default value	-	***10***
Comments	-	Recommended values T2 < T1

9.6.8.4 T3 [s]

Function	-	T3 timeout - time limit for test data frame sending in case long idle time
Data type	-	Number
Range	-	***1 - 300***
Default value	-	***0***
Comments	-	N/A

9.6.8.5 K

Function	-	Maximal number of frames that module is sending and waits for confirmation of delivery
Data type	-	Number
Range	-	***1 - 8***
Default value	-	***8***
Comments	-	N/A

9.6.8.6 W

Function	-	Maximal number of received inquiries before sends a confirmation of receiving
Data type	-	Number
Range	-	***0 - 8***
Default value	-	***4***
Comments	-	N/A

9.6.8.7 Number of variables

Function	-	number of variables in table
Data type	-	Number
Range	-	***0 - 4***
Default value	-	***0***
Comments	-	N/A

9.6.8.8 Event count

Function	-	number of events in variable's table
Data type	-	Number
Range	-	***0 - 32***
Default value	-	***0***
Comments	-	N/A

9.6.8.9 Time synchronization

Function - allows activation of time synchronization in module (Server) from external device (Client)

Data type - Selection list

Range - ***Yes***

Server time will be synchronized according to settings from Client.

No

Synchronization of the time is switch off.

Default value - ***No***

Comments - N/A

9.6.8.10 Variable table

Idx. - Index number

Name - Friendly name

IOA - quantity address from IEC protocol side
Range:
1 - 1000

Type - variable type format
Range:
short
float
single point

Space - selection of register space in MT module
Selection list
Input registers
Przestrzeń rejestrów wejściowych (wartość domyślna)
Holding registers
Przestrzeń rejestrów wewnętrznych

Address - register address in module MT
Range:
0 - 8191

Time Tag - allows to choose that time stamp will be add or not to requested variable
Default value: ✗ without time stamp

Interrogation - allows to choose that variable will be send in answer for global request (general interrogation)
Default value: ✗ not sent

9.6.8.11 Group table

Idx. - Index number

Name - Variable name entry in Variable table

Group 1 - 8 - add variable to selected Group, value of the variable will be send after interrogation of the Group.
Default value: ✗ not added

9.6.8.12 Event table

Idx. - Index number

Triggering bit - Defines the bit which state change will trigger event
0 - 65535 or name from bit list (see bit list in Appendices)
For binary output space is required to enter prefix 10xxx before address, for flag P1 value of triggered bit is 11600 (P1 bit is address 1600[dec])

Triggering slope - Event triggering edge
Selection list
0->1
rising edge (default value)
1->0
falling edge
0<->1
any edge

Data sent - Event content
Range:
Variable - single variable
Group - variable group

Data selection - allows to choose specific **Variable** or **Group**. Selection depends from settings of parameter ***Przesyłane dane***

9.7 Presets

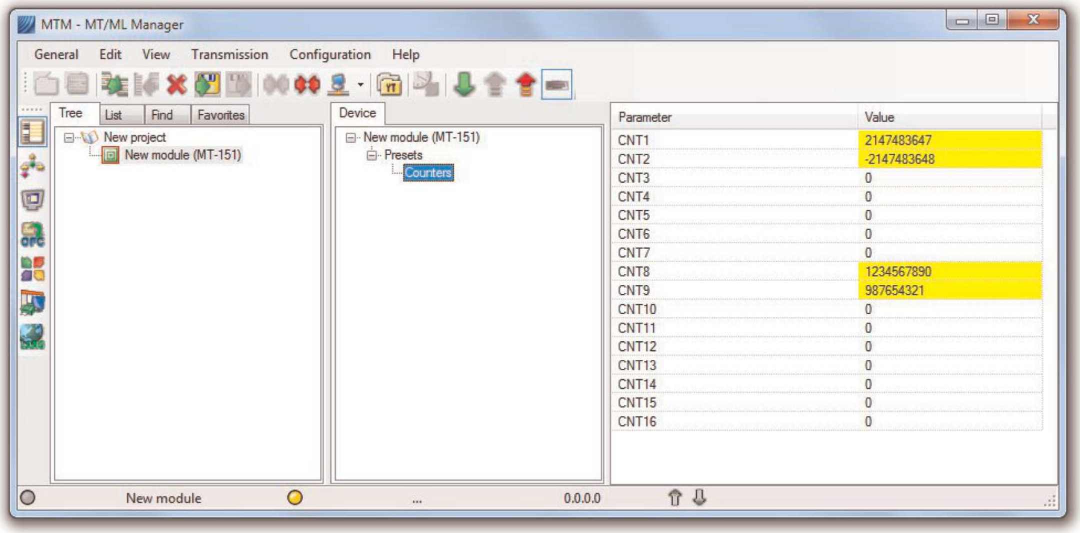
In order to expand module application area it is furbished with initial settings for some resources. It is necessary when the module is operating as a pulse counter for measuring devices (e.g. water consumption meter with pulse output) having initial count other than zero. Due to **Presets** the actual value of counter register may be equalized with mechanical counter of the device thus not disturbing the functionality of the system.
In order to set **Presets** go to Configuration menu and select the Initial settings option or click following icon on the toolbar.

The **Presets** icon is active only when the module is connected and selected transmission channel is not set to Spooler. Sending data in **Presets** mode is possible only as sending changes. Bear in mind that sending configuration changes result in immediate and irrevocable updating of the resource.
When **Presets** mode is selected all configuration groups disappear from the panel and only parameters that may have initial value set are displayed. For MT-151 HMI module those parameters are **counters CNT1 - CNT16**.

9.7.1 Counters (CNT1 - CNT16)

Name of resource - counters CNT1 - CNT16
Data type - number
Range - ***-2147483648 - 2147483647***
Default value - ***0***
Comments - N/A

After entering new value counter field turns yellow. It means that value visible in MTManager is not written to device. To send new values to device press **Write changes** button (second from the right on toolbar).

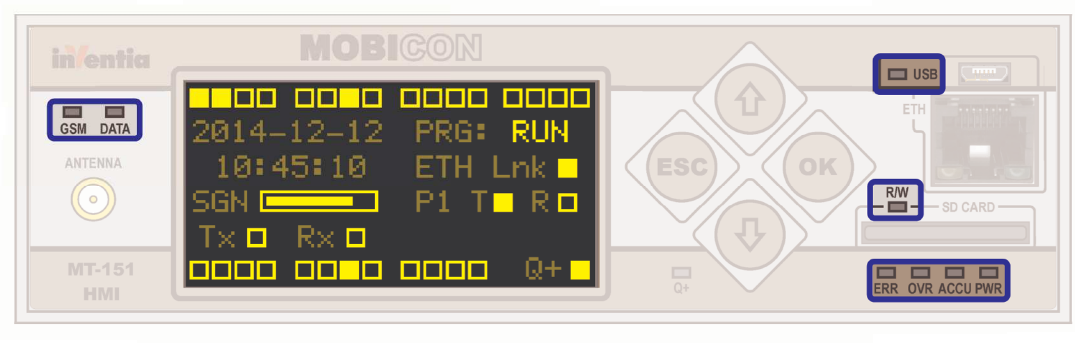


Parameter	Value
CNT1	2147483647
CNT2	-2147483648
CNT3	0
CNT4	0
CNT5	0
CNT6	0
CNT7	0
CNT8	1234567890
CNT9	987654321
CNT10	0
CNT11	0
CNT12	0
CNT13	0
CNT14	0
CNT15	0
CNT16	0

10 Problem solving

10.1 Module Status Screen and LEDs

Information displayed on LED indicators and OLED Display placed on MT-151 HMI front panel are a great help during module startup and troubleshooting. Signalized states are displayed on the Main Module status screen and additional screens as text or graphic information.

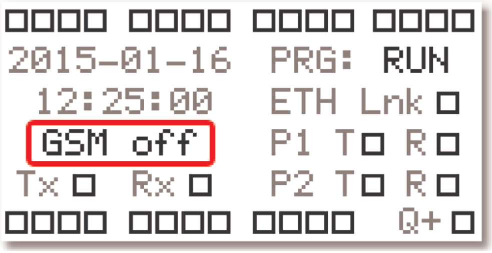


LED indicators meaning:

- **GSM** LED light on green color - module logged into GSM 2G network, **DATA** LED on - module is transmitted in GPRS\EDGE
- **GSM** LED light on blue color - module logged into 3G network, **DATA** LED on - module is transmitted in UMTS\HSPA+.
- **GSM** LED light on red color - communication error that description is coded on **DATA** and **SGN-LEVEL** LEDs.
- **DATA** indicator is lighted only when module is correctly logged to GPRS service
- **USB** indicator is lighted when USB connection is correct, blinking when transmission
- **R/W** indicator is lighted when MicroSD card is formatted and correctly installed, blinking when memory card is read or written
- **PWR** indicator is lighted when main power is connected, turn off after module restarting for a 5 seconds
- **ACCU** indicator is lighted when additional battery is connected, blinking when battery voltage is low (below 11,5V)
- **OVR** indicator is lighted when control program is stopped. It is also lit for few seconds after writing new firmware or configuration to module indicating that module should not be powered off.
- **ERR** indicator is lighted when an error forcing automatic reboot. The reason may be lack of GPRS/3G communication. Triple flash of ERR LED indicates that current firmware does not support the function used in the program. In this situation please update module firmware to most recent.

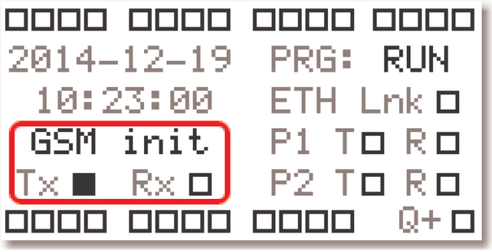
10.1.1 GSM status

GSM status on Module Status Screen provides information about GSM login initiations, short blinks **Tx** (indicates sending the data and SMS), **Rx** (indicates reception of the data and SMS) when transmitted data and signal strength after login procedure is finished (GSM LED is lights on). Signal level is signaled on progress bar which a full fill means a maximum strength.



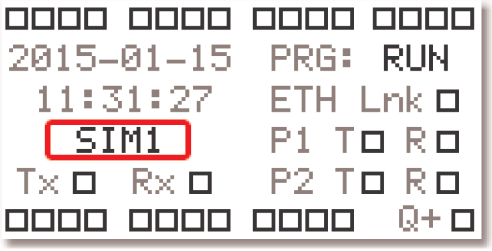
```
□□□□ □□□□ □□□□ □□□□
2015-01-16  PRG: RUN
 12:25:00   ETH Lnk □
 GSM off    P1 T□ R□
Tx □  Rx □  P2 T□ R□
□□□□ □□□□ □□□□  Q+ □
```

- Start or restart the GSM modem



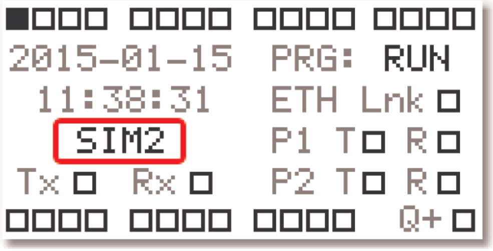
```
□□□□ □□□□ □□□□ □□□□
2014-12-19  PRG: RUN
 10:23:00   ETH Lnk □
 GSM init   P1 T□ R□
Tx ■  Rx □  P2 T□ R□
□□□□ □□□□ □□□□  Q+ □
```

*- GSM connection is initialized, **Tx** and **Rx** blinking few times*



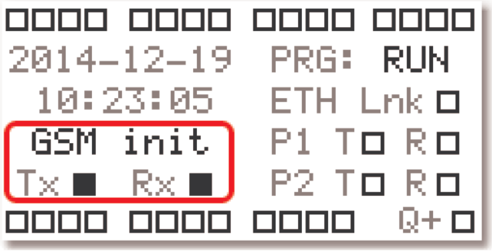
```
□□□□ □□□□ □□□□ □□□□
2015-01-15  PRG: RUN
 11:31:27   ETH Lnk □
   SIM1     P1 T□ R□
Tx □  Rx □  P2 T□ R□
□□□□ □□□□ □□□□  Q+ □
```

or



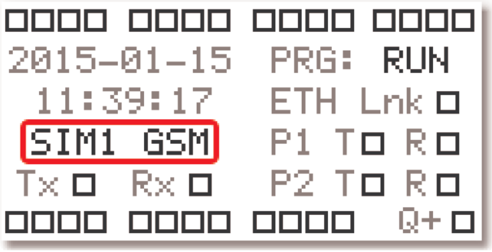
```
■□□□ □□□□ □□□□ □□□□
2015-01-15  PRG: RUN
 11:38:31   ETH Lnk □
   SIM2     P1 T□ R□
Tx □  Rx □  P2 T□ R□
□□□□ □□□□ □□□□  Q+ □
```

- Currently used SIM card



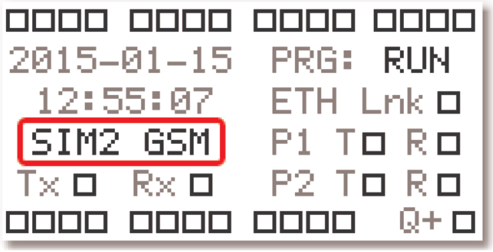
```
□□□□ □□□□ □□□□ □□□□
2014-12-19  PRG: RUN
 10:23:05   ETH Lnk □
 GSM init   P1 T□ R□
Tx ■  Rx ■  P2 T□ R□
□□□□ □□□□ □□□□  Q+ □
```

*- GSM connection is still initialized, **Tx**/**Rx** blinking several times*



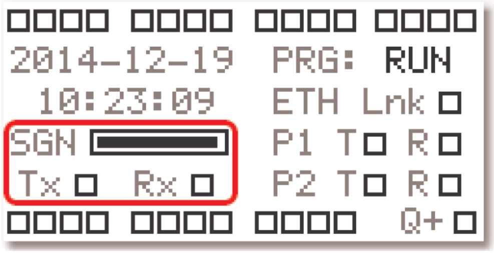
```
□□□□ □□□□ □□□□ □□□□
2015-01-15  PRG: RUN
 11:39:17   ETH Lnk □
SIM1 GSM    P1 T□ R□
Tx □  Rx □  P2 T□ R□
□□□□ □□□□ □□□□  Q+ □
```

or



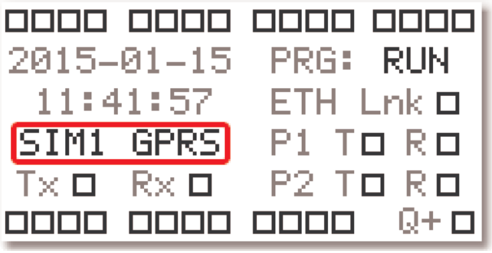
```
□□□□ □□□□ □□□□ □□□□
2015-01-15  PRG: RUN
 12:55:07   ETH Lnk □
SIM2 GSM    P1 T□ R□
Tx □  Rx □  P2 T□ R□
□□□□ □□□□ □□□□  Q+ □
```

- Device is logged in GSM service on current SIM card



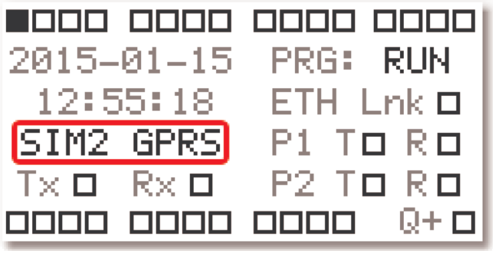
```
□□□□ □□□□ □□□□ □□□□
2014-12-19  PRG: RUN
 10:23:09   ETH Lnk □
SGN ▬▬▬▬▬   P1 T□ R□
 Tx □ Rx □  P2 T□ R□
□□□□ □□□□ □□□□  Q+ □
```

*- Signal level indicator is shown just after correct GSM login (**GSM LED** light on), **SGN** presents high signal strength*



```
□□□□ □□□□ □□□□ □□□□
2015-01-15  PRG: RUN
 11:41:57   ETH Lnk □
 SIM1 GPRS  P1 T□ R□
 Tx □ Rx □  P2 T□ R□
□□□□ □□□□ □□□□  Q+ □
```

or



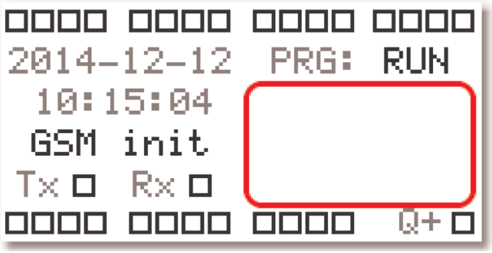
```
■□□□ □□□□ □□□□ □□□□
2015-01-15  PRG: RUN
 12:55:18   ETH Lnk □
 SIM2 GPRS  P1 T□ R□
 Tx □ Rx □  P2 T□ R□
□□□□ □□□□ □□□□  Q+ □
```

*- Correct login into GPRS Service (**GPRS LED** light on) on current SIM card*

Pressing OK Button, while signal strength bar is visible, allow to check which SIM card is in use.

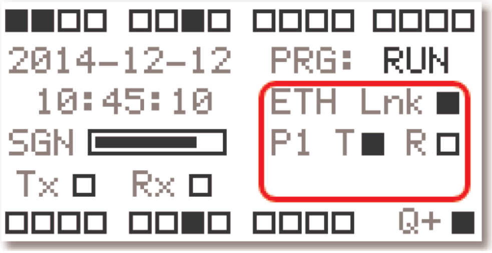
10.1.2 Interfaces activity

Activity on internal interfaces is shown on Main status screen when specific port is enabled in configuration. Short blinks of **Tx** indicate data packet sending while **Rx** LED blink indicates reception of data packet.



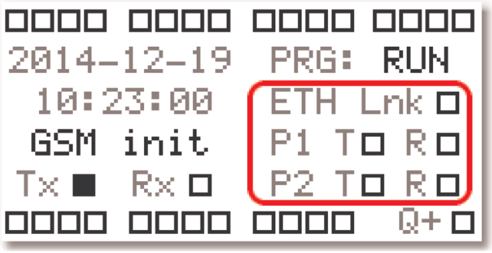
```
□□□□ □□□□ □□□□ □□□□
2014-12-12  PRG: RUN
 10:15:04
 GSM init
 Tx □ Rx □
□□□□ □□□□ □□□□  Q+ □
```

- All wired interfaces are disabled in configuration, GSM modem is initialized.



```
■■□□ □□■□ □□□□ □□□□
2014-12-12  PRG: RUN
 10:45:10   ETH Lnk ■
SGN ▬▬▬▬    P1 T■ R□
 Tx □ Rx □
□□□□ □□■□ □□□□  Q+ ■
```

- Ethernet port is enabled and cable is correctly connected, Port 1 is enabled and transmits a data now, Port 2 is disabled in configuration, Signal level is quite well. Modem is logged in GSM.

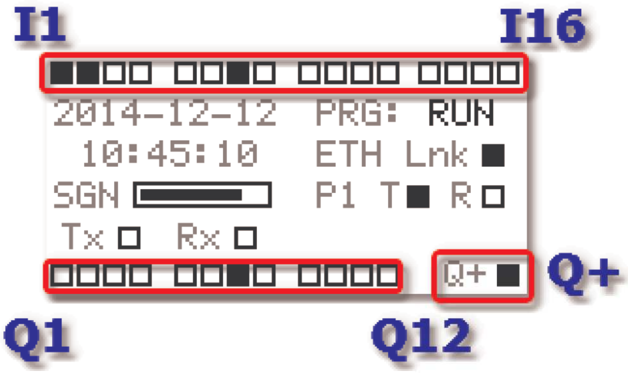


```
□□□□ □□□□ □□□□ □□□□
2014-12-19  PRG: RUN
 10:23:00   ETH Lnk □
 GSM init   P1 T□ R□
 Tx ■  Rx □ P2 T□ R□
□□□□ □□□□ □□□□  Q+ □
```

- Ethernet port is enabled but cable is not connected, Port 1 and 2 is enabled in configuration but not transmit a data, GSM modem is initialized.

10.1.3 Binary inputs/outputs

Indicators of binary inputs and outputs are signaling logical state of all pins (on - high state, off - low state).



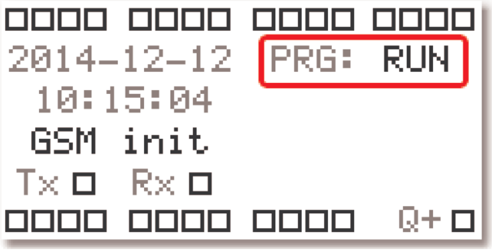
```
■■□□ □□■□ □□□□ □□□□
2014-12-12  PRG: RUN
 10:45:10   ETH Lnk ■
SGN ▬▬▬▬▬   P1 T■ R□
 Tx □  Rx □
□□□□ □□■□ □□□□  Q+ ■
```

- on pins I1, I2, I7, Q7 are high state

If any pointer from group Q1-Q12 is blinking with 2Hz frequency, then module detected difference between state of corresponding output bit and actual state of pin. It may happen when outputs are not powered (Q+ pointer is off) or are connected directly to ground.

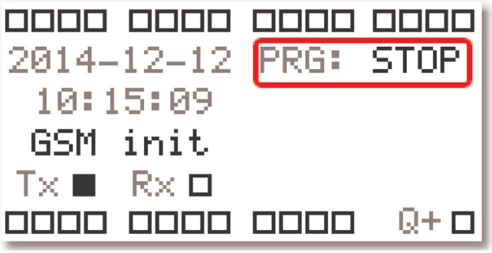
10.1.4 Internal program status

Indicator of internal program status are signaling tree states.



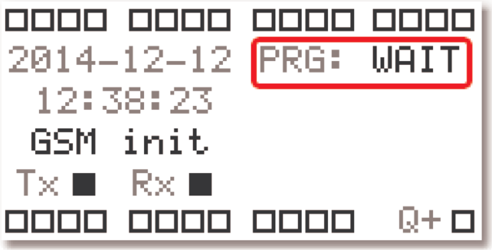
```
□□□□ □□□□ □□□□ □□□□
2014-12-12  PRG: RUN
 10:15:04
 GSM init
 Tx □  Rx □
□□□□ □□□□ □□□□  Q+ □
```

- Internal program is running now



```
□□□□ □□□□ □□□□ □□□□
2014-12-12 PRG: STOP
 10:15:09
 GSM init
 Tx ■  Rx □
□□□□ □□□□ □□□□  Q+ □
```

- Internal program is stopped now

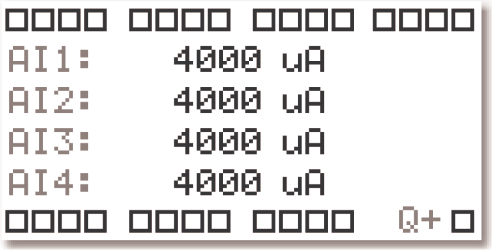


```
□□□□ □□□□ □□□□ □□□□
2014-12-12 PRG: WAIT
 12:38:23
 GSM init
 Tx ■  Rx ■
□□□□ □□□□ □□□□  Q+ □
```

- Internal program upload in progress now

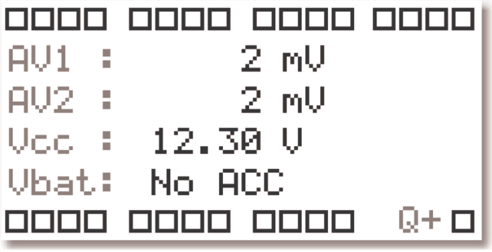
10.1.5 Additional status screens

Additional status screens can be switch off in configuration.



```
□□□□ □□□□ □□□□ □□□□
AI1:    4000 uA
AI2:    4000 uA
AI3:    4000 uA
AI4:    4000 uA
□□□□ □□□□ □□□□  Q+ □
```

- Measurement values in engineering units on analog inputs AI1 ... AI4 with the unit name of measure

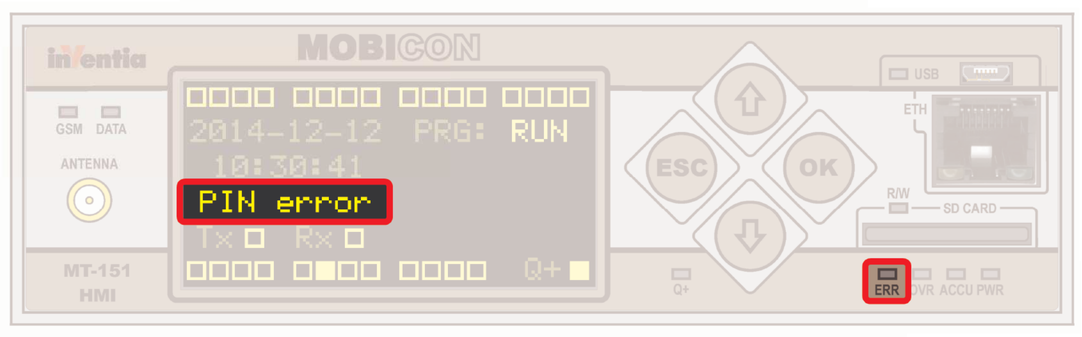


```
□□□□ □□□□ □□□□ □□□□
AV1 :        2 mV
AV2 :        2 mV
Vcc :  12.30 V
Vbat:  No ACC
□□□□ □□□□ □□□□  Q+ □
```

- Measurement values in engineering units on analog inputs AV1 ... AV2 with the unit name of measure, an actual power voltage and additional battery voltage if connected

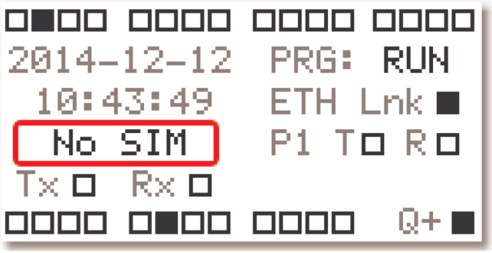
10.2 MT-151 HMI Error signaling

Despite the efforts of the software developers some operational errors of the module may occur. It is often imperative to diagnose and remove the cause of error. Error signaling is a tool for solving problems. Following the diagnostic information presented on Module Status Screen displayed information or error messages, the **ERR** LED indicator is lighted. Error messages are shown in place of signal level indicator.



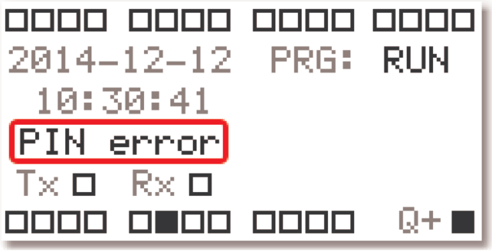
Error signaling in HMI version

10.2.1 Standard errors



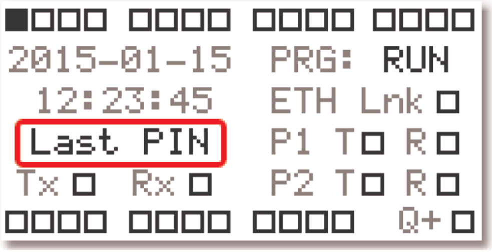
```
□■□□ □□□□ □□□□ □□□□
2014-12-12  PRG: RUN
 10:43:49   ETH Lnk ■
  No SIM    P1 T□ R□
 Tx □  Rx □
□□□□ □■□□ □□□□  Q+ ■
```

- No SIM card inserted



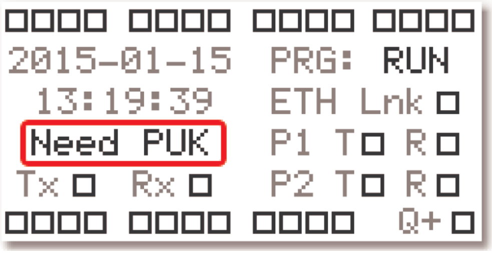
```
□□□□ □□□□ □□□□ □□□□
2014-12-12  PRG: RUN
 10:30:41
PIN error
 Tx □  Rx □
□□□□ □■□□ □□□□  Q+ ■
```

- Wrong PIN number to SIM card



```
■□□□ □□□□ □□□□ □□□□
2015-01-15  PRG: RUN
 12:23:45   ETH Lnk □
 Last PIN   P1 T□ R□
 Tx □  Rx □ P2 T□ R□
□□□□ □□□□ □□□□  Q+ □
```

- Second attempt to enter the PIN code was failure, Enter right PIN code using cellphone.



```
□□□□ □□□□ □□□□ □□□□
2015-01-15  PRG: RUN
 13:19:39   ETH Lnk □
 Need PUK   P1 T□ R□
 Tx □  Rx □ P2 T□ R□
□□□□ □□□□ □□□□  Q+ □
```

- SIM card is blocked, Enter right PUK code using cellphone.

10.3 Unlocking the SIM card

Triple insertion of wrong PIN code results in locking the SIM card. Locked card renders SMS and data transmission impossible. Locked SIM card is signaled on main status screen.
In order to unlock the SIM card do the following:

- Power off the module
- Remove the SIM card from device
- Insert the SIM card to a mobile phone that accepts the SIM issued by your operator
- Start the phone and insert the PUK code followed by PIN code
- Power the module on
- Insert proper PIN into configuration
- Power the module off
- Install the SIM card in the module
- Power the module on

Executing the procedure unlocks the SIM card and enables module's proper operation.

11 Technical parameters

11.1 General

Dimensions (height x width x depth)	157mm x 86mm x 58mm
Weight	400g
Mounting method	35mm DIN rail
Operating temperature	-20°C, +65°C
Ingress protection class	IP40
Humidity	up to 95%, non-condensing

11.2 GSM modem

Modem type	Cinterion EHS6
GSM band	Quad band 850/900/1800/1900Mhz
UMTS	800/850/900/1900/2100Mhz
Antenna	50Ω

11.3 Power supply

Direct current DC (12VDC, 24VDC)	**10.8 - 30V**
Input current for 12VDC	Idle 0.12A Active 0.40A Max 2.00A
Input current for 24VDC	Idle 0.06A Active 0.20A Max 1.00A
External battery nominal voltage	12V
External battery nominal capacity	7Ah
Maximum external battery charging current	400mA

NOTICE!
Due to high momentary current consumption the power supply must be capable of delivering > 2A of current.
Inappropriate power supply may result in faulty operation or can damage the module!

11.4 Binary inputs

For binary inputs I1-I16	
Input voltage range	0 - 30V
Input current limit	2,4mA
Input voltage for high state (1)	> 9,4V
Input voltage for low state (0)	< 8,4 V

For binary outputs Q1-Q12 operating in binary input mode	
Maximum input voltage	30V
Input current	10 mA / 24V
Input voltage for high state (1)	> 9,4V
Input voltage for low state (0)	< 8,4V

11.5 Binary outputs

Maximum output current	100mA
Voltage drop for 100mA	< 0.5V max.
Current in low state (0)	< 100μA max.

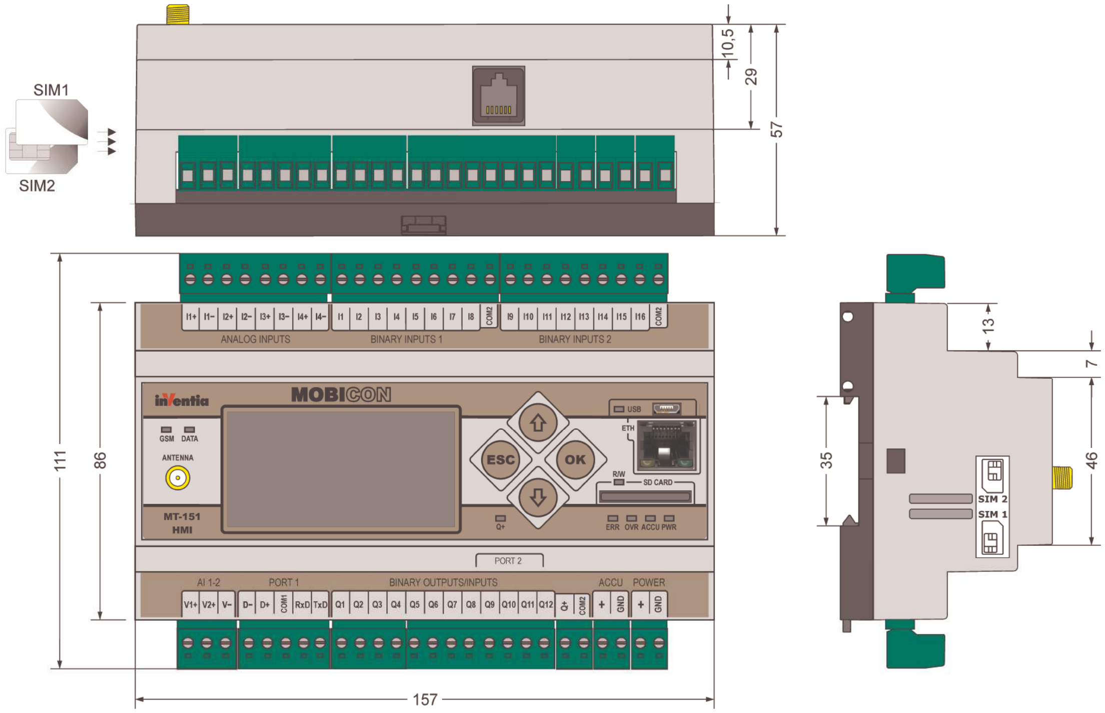
11.6 Analog inputs 4-20mA

Measuring range	4 - 20mA
Maximum input current	50mA
Dynamic impedance	typically 55 Ω
Voltage drop at 20mA	< 5V
A/D converter resolution	14 bits
Accuracy @ 25°C	±0.2%

11.7 Analog inputs 0-10V

Measuring range	0 - 10V
Maximum input voltage	20V
Dynamic impedance	typically 197kΩ
A/D converter resolution	12 bits
Accuracy @ 25°C	0.5%

11.8 Drawings and dimensions



NOTICE!
All dimensions in millimeters.

12 Safety information

12.1 Working environment

When deploying telemetry modules one has to observe and comply to local legislation and regulations. Using the telemetry module in places where it can cause radio noise or other disturbances is strictly prohibited.

12.2 Electronic equipment

Thou most of modern electrical equipment is well RF (Radio Frequency) shielded there is no certainty that radio waves emitted by the telemetry module's antenna may have negative influence on it's function.

12.2.1 Heart pacemakers

It is recommended that the distance between the antenna of telemetry module and the Heart Pacemaker is greater than 20 cm.
This distance is recommended by manufacturers of Pacemakers and in full harmony with results of studies conducted independently by Wireless Technology Research.

12.2.2 Hearing aids

In rare cases the signal emitted by the telemetry module's antenna may disturb hearing aids functions. Should that occur, one has to study detailed operating instructions and recommendations for that particular product.

12.2.3 Other medical equipment

Any radio device including the telemetry module may disturb the work of electronic medical equipment.
When there is a need of installing telemetry module in vicinity of medical equipment one has to contact the manufacturer of this equipment in order to make sure that the equipment is adequately protected against interference of radio frequency waves (RF).

12.2.4 RF Marked equipment

The restriction against installing telemetry modules in areas marked as radio frequency (RF) prohibition zones must be unconditionally observed.

12.3 Explosive environment

Installation of telemetry modules in the environment where explosion hazard is present is not permitted. Usually, but not always, these places are marked with warning signs. Where there is no marking do not install telemetry modules at liquid or gas fuels stores, inflammable materials stores, nor places contaminated with metal or wheat dust.

13 Appendices

13.1 Register of changes - documentation

Compatibility with firmware **- v2.00.04**

----

v2.00.04 - 2017-09-19

- Explication of parameter description: *Sender IP address control*
- Registry of changes was split to device and documentation
- Actualization to actual version of module firmware (MTC)

v2.00.02 - 2017-08-30

- Correct voltage value of supply range
- Minor text bug fixes

v2.00.01 - 2017-07-15

- First released of hardware version V2

13.2 Register of changes - device

Current firmware version **- v2.00.04**
Require MTManager - **v5.2.2.6** or higher

----

v2.00.04 - 2017-07-11

- critical fix in authorization of data transmission
- fix in SIM cards switching
- add an error signalization GSM diode - diode blinks red colour
- other minor bug fixes

v2.00.01 - 2017-06-23

- first released in hardware version V2

13.3 SNCS Simple Name Command Syntax

Description of SNCS commands

Internal application of a module is able to receive, process and send short text messages (SMS). There is a set of command which can be put in SMS and e-mail message, allowing the user to read from and write (SMS only) to internal registers placed in module's memory. Some commands can be used in device configuration parameters for managing of contents presentation on internal graphics panel.

Characters with special meaning:

Character	Description	SMS	Display
#	Starts a command ATTENTION! putting two hash signs one after another will prevent module from processing command following it. However after sending one of hash signs will be deleted - this allows to control resources of one module from another, e.g. set binary output Q1 to '1' (##HB128=1)	supported	supported
*****	starts a macro	supported	N/A
>	used as first character in SMS text inhibits parsing of SMS	supported	N/A
$	used as first character in SMS text inhibits answering to this SMS	supported	N/A
!	used as first character before (#) starts command character allow edit a values	N/A	supported
:	used after numerical address of a register allow to set the range of permissible values (ex. !#HR1024:10,100)	N/A	supported
|	line contents separator, right and left text align	N/A	supported

After reception of SMS message, internal application tries to parse SMS text and execute command enclosed in it. Parsing process generates new message text, which is send back to user (if module is allowed to, either by configuration or by presence/absence of '$' sign).

Commands are formatted as follows:

#[prefix.]symbol[=value]

where:

prefix defines data representation and register count

symbol defines register address and register space being accessed

value defines data to be written to register (s)

Prefix is optional; when not present, data is interpreted according to preset defaults.

Basic read commands:

#HR0

When module receives and parses the SMS message containing this command, command string will be replaced with value of register 0 read from holding registers space, noted in decimal format, and this value will be put in SMS sent back to user. Answer to this command sent back will be:

>10

where 10 is value read from holding register 0.

If received SMS contains any other characters than correctly formatted commands, these characters will be copied unaltered to message being sent back. This allows user to freely compose text of return message and include register values together with some informational text. For example, if user sends containing:

GSM signal level is #IR132%

then module will answer with:

>GSM signal level is 96%

where 96 is a value read from input register 132.

It should be noted that answer from module begins with '>' sign - it means that this SMS was generated by module. If module receives SMS beginning with '>', such message will be ignored (not parsed). This prevents endless "looping" of messages in case they are being exchanged between modules

Writing to register is archived by expanding basic command with '=' sign and value that should be written:

#HR20=2

User should be aware that writing is allowed only to holding register space.

When module receives SMS with write command, it executes the command and sends back value written. For example, sending to module SMS with text:

#HR1=1234

causes module to write value 1234 to holding register 10 and send back SMS with text:

>1234

Both read and write commands can be expanded by adding a prefix, which defines data format (notation). Prefix should be placed between '#' mark (command start) and register symbol, and should contain one (or more) characters ended with a dot. For example, to read an input register 4 in hexadecimal format, one should use a command:

#H.IR4

and module's answer will be:

>1FC8

Prefixes can also be used with write commands.

Command can operate on more than one register. Register count can be included in prefix, after character denoting data format (which is then mandatory). For example, command:

#D2.HR4=123456

causes write 123456 to two registers, HR4 and HR5 (32-bit variable).

Full list of available prefixes is enclosed below.

User can define in MTManager own symbolic names in module's configuration and assign them to registers. Then, such names can be used instead of register symbols. It allows user to define "friendly" names for registers and to erase access to bit values. For example, if user has defined symbolic name "output" and assigned it to bit 48 of internal registers space (which is equal to 0 bit of HR3 register), then sending a command:

#output=1

causes module to write 1 to bit 0 of HR3 register. There are several predefined (internal) symbolic names.

Apart from symbolic names, user can define macros. A macro is defined as a name and a text assigned to this name. Parsing of received message begins with macro expansion. Parser looks for words beginning with '*' sign and replaces such names with assigned strings. Once macro expansions ends, new message text is being interpreted and commands executed. It allows user to place both commands and symbolic names in macro text. Furthermore, macros can contain another macro names ("nested" macros), but only those defined higher in macro list. For example, if configuration contains following macros (in order shown):

No.	Macro name	Macro text
1	counter	*mttime: counter of I1: #D2.HR4
2	mttime	#date #time
3	state	*mttime: GSM - #IR132%

then macro *mttime used in macro number 3 (*state) will be correctly expanded and SMS text:
***state**

after macro expansion (before executing commands) will be changed to:
#date #time: GSM - #IR132%

but expansion of macro 1 will not contain text assigned to macro name *mttime, therefore text being executed after macro 1 was used will look like:
***mttime: counter of I1: #D2.HR4**

this in turn causes module to send back SMS containing:
>*mttime: input 0 counter: 123

Register spaces

Module's firmware distinguishes two register spaces: Input Registers and Holding Registers. Access to register space can be made by calls to 16-bit registers or by calls to individual bits.

Symbol	Description
HR{0-n}	Holding registers space. Read/write access. 16-bit registers.
IR{0-n}	Input registers space. Read only. 16 bit registers.
HB{0-16*n}	Bit access to holding registers space. One can access individual bits (or groups of bits). Read/Write. Bit mapping is as follows: bits 0-15 correspond to holding register 0, bits 16-31 - to holding register 1 and so on.
IB{0-16*n}	Bit access to input registers space. One can access individual bits (or groups of bits). Read only. Bit mapping is as follows: bits 0-15 correspond to input register 0, bits 16-31 - to input register 1 and so on.

Register symbols can be preceded by prefixes, which can define amount of data being processed and data format.

Available prefixes:

Register space HR, IR (16-bit registers)

Prefix	Description
B[1-4]	Binary format, 16 characters (bits) default, bits from most to least significant. Prefix can contain register count (1-4) being processed (register symbol defines lowest register) - in resulting string, rightmost character corresponds to bit with lowest number.
D[1-4]	Decimal format, 1-5 characters, unsigned. Prefix can contain register count (1-4) being processed (register symbol defines lowest register) - number returned is decimal notation of n*16 bit value where most significant bit is placed in register with lowest address (big-endian).
H[1-4]	Hexadecimal format, 4 characters. Prefix can contain register count (1-4) being processed - returned string contains n*4-character groups, leftmost group correspond to register with lowest address (big-endian).

S	Decimal format, 1-5 characters (with '-' sign when needed), signed. Access to single register treated as 16-bit signed value.
F[1-3]	Converts decimal value to floating point number. Number next to prefix defines number of digits after dot.
T	Textual representation of holding register value (only HR, low byte) - max. 63 characters. Require sign of the end: null (0x0000)
TXT(X[+Y])	Read Textual constant parameter value that is defines in configuration branch Resources\Constant parameters\Text. X means index from parameters list (value from 1 to 72). It is possible to dynamically substitution of index value taken from internal registers space or bits. Y value is optional and define constant offset for index table.

Bit access to register spaces - HB, IB

Prefix	Description
B[1-64]	Binary format. Amount of bits being displayed provided in prefix. Bits are presented in order from least to most significant (opposite to binary representation of whole register).
D[1-64]	Decimal format. Value presented is calculated from amount of bits provided in prefix, with bit with lowest address being least significant (*little endian*)
H[1-64]	Hexadecimal format. Value presented is calculated from amount of bits provided in prefix, with bit with lowest address being least significant (*little endian*)

Predefined symbolic names

Name	Description
TIME	Returns local time read from RTC registers
DATE	Returns local date read from RTC registers
NAME	Returns module name
SERIAL	Returns module serial number
IPADDR	Returns module current IP address
CR	Enters new line in SMS text

Other examples:

Read Input Registers address 23:
#IR23

Write value 1 to Holding Register 3:
#HR0=3

Binary representation of Input Register 17 (readout):
#B.IR17

Read flag (bit) 4:
#B.IB17

Write hexadecimal value **01AC** to holding register 4:
#H.HR4=01AC

Read 8 bits starting from address 16 (Input Registers 1):
#B8.IB16

Read decimal number consisting from 6 bits starting from address 64 (Input Register 4):
#D6.IB64

Write single bit to register (Bit 0 to Holding Register 3):
#HB48=1

Read signed number from register:
#S.IR18

Read Holding Register address 122 with two decimal places presentation:

#F2.HR122

Write texts from registers starts from HR7000 address to register 0:

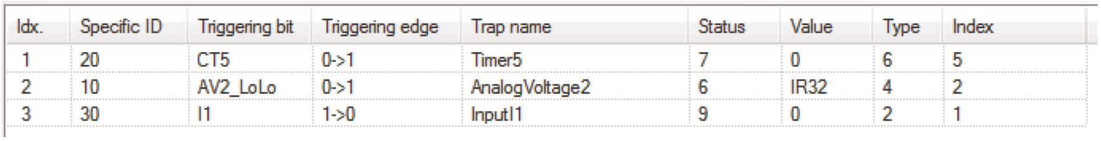
#T.HR7000

Read the constant textual parameter value depends from bit address number 272 (I1) from analog inputs space (include binary input space). If the input I1=0 then will be read a value of the text parameter under index 1 (0+1). If the Input I1=1 then will be read a value of the text parameter under index 2 (1+1):

#TXT(#IB272+1)

13.4 SNMP - trap configuration example

Below is presented sample configuration of three traps with data assigned to trap variables directly in configuration (Trap data source parameter is set to Configuration).



Idx.	Specific ID	Triggering bit	Triggering edge	Trap name	Status	Value	Type	Index
1	20	CT5	0->1	Timer5	7	0	6	5
2	10	AV2_LoLo	0->1	AnalogVoltage2	6	IR32	4	2
3	30	I1	1->0	InputI1	9	0	2	1

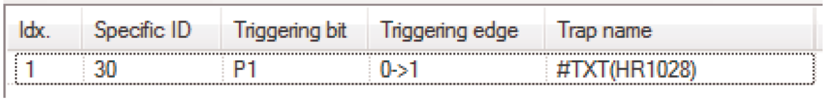
Trap number 1 has Specific ID set to 20, which according to MIB file means that synchronous/asynchronous timer reached its threshold. Triggering bit is set to CT5 and triggering edge to 0->1, which means that this trap will be triggered when timer CT5 will count up to its threshold and set its flag to 1. Timer5 is the name of a trap. Status is set to 7, which according to MIB file means that timer reached its threshold, while type set to 6 informs that this timer is synchronous. Index identifies which timer it is.

Trap number 2 informs about alarm on analog input (Specific ID=10 – alarm on analog input). Trap is triggered by activation of LoLo alarm on analog input AV2. Status set to 6 informs that this is LoLo alarm. Type set to 4 informs that alarm was triggered on voltage analog input, while index points analog input AV2. Value provides information about current value of analog input by addressing its register.

Trap number 3 is triggered by falling edge on binary input I1. Specific ID set to 30 informs that it is triggered by change of binary input logical state. Status set to 9 informs that this binary input was turned off. Type set to 2 informs that this binary input is internal module binary input while index points to first binary input.

13.4.1 Sending traps using internal program

Below is presented sample configuration of three traps with data assigned to trap variables indirectly via holding registers (Trap data source parameter is set to Registers).



Idx.	Specific ID	Triggering bit	Triggering edge	Trap name
1	30	P1	0->1	#TXT(HR1028)

Trap is defined by:

- Specific ID – it is 30 corresponding to binary input state change,
- Triggering bit and Triggering edge – they point to rising edge of program flag P1
- Trap name – it is set to Text constant parameter with index set up by value stored in register with address 1028 in holding registers address space.

Values of trap variables are copied from registers as follows:

- **trapSourceStatus** from register address 1030+5*(trap_index-1) in holding registers address space (HR1030)
- **trapSourceValue** from register address 1031+5*(trap_index-1) in holding registers address space (HR1031)
- **trapSourceType** from register address 1032+5*(trap_index-1) in holding registers address space (HR1032)
- **trapSourceIndex** from register address 1033+5*(trap_index-1) in holding registers address space (HR1033)

This configuration allows to control from program when trap is send and what information it carries. This allows sending data from external sources and using it to trigger traps. However it is possible to send only one trap per program cycle.

Below is sample program controlling trap sending. It sends data from external binary inputs (up to 16) which are mapped to REG3 register. It assumes that names are stored in text table from position 1 to 16.

```
#include "MT-151.h"
#define  INPUT_REG  REG3  //  register  containing  external  binary  inputs
state
#define LOCAL_REG REG4
#define MASK REG6
#define TRAP_STATUS hreg &1030 // SNMP variables registers
#define TRAP_VALUE hreg &1031
#define TRAP_TYPE hreg &1032
#define TRAP_INDEX hreg &1033
#define TRAP_NAME hreg &1034 // trap name index
IF NOT P1 JMP check // there was jump in previous cycle?
```

```
BCPY 0, P1 // if yes - reset flag
JMP end
check:
BXOR INPUT_REG, LOCAL_REG, REG5 // check if there was change in inputs
state
NE REG5, 0, P1 // set flag if there is a change
IF P1 JMP select // seek for changed bit
JMP end
select:
MOVE 0, REG255 // loop counter = bit index
MOVE 1, MASK // mask
loop:
BTST REG5, MASK, P10 // check if this bit was changed
IF P10 JMP change // if yes prepare trap data
ADD REG255, 1, REG255
LSL MASK, 1, MASK
NE REG255, 16, P255
IF P255 JMP loop // repeat for 16 bits
change:
MOVE 3, TRAP_TYPE // set trap type to 3 - external input
ADD REG255, 1, TRAP_INDEX // set index of input that triggered trap
ADD REG255, 1, TRAP_NAME // set index of name of input that triggered
trap
BTST INPUT_REG, MASK, P10 // check binary input change slope
IF P10 JMP one
zero:
BNOT MASK, MASK
BAND LOCAL_REG, MASK, LOCAL_REG // zero bit in local copy of inputs
status
MOVE 9, TRAP_STATUS // set trap status to 9 - off
MOVE 0, TRAP_VALUE // set trap value to 0
JMP end
one:
BOR LOCAL_REG, MASK, LOCAL_REG // set bit in local copy of inputs status
MOVE 8, TRAP_STATUS // set trap status to 8 - on
MOVE 1, TRAP_VALUE // set trap value to 1
end:
EXT
```

13.5 List of Bits

During its operation **MT-151 HMI** is setting a series of binary variables associated with the I/O and module diagnostics. **MTManager**, for user convince, has implemented list of predefined bits.

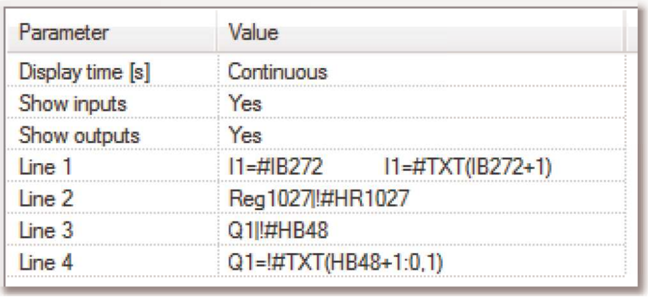
Name of bit	Description
I1 - I16	Binary inputs I1 - I16
IQ1 - IQ12	Binary inputs IQ1 - IQ12 (outputs Q1 -Q12 operating in binary input mode)
Q1 - Q12	Binary outputs Q1 -Q12
CT1 - CT16	Synchronous timer flags CT1 - CT16
CK1 - CK16	Asynchronous timer flags CK1 - CK16
AI1_LoLo - AI4_LoLo	Analog inputs 4-20mA alarm bits - LoLo alarm. Measured value lower than LoLo alarm threshold.
AI1_Lo -AI4_Lo	Analog inputs 4-20mA alarm bits - Lo alarm. Measured value lower than Lo alarm threshold.

AI1_Hi - AI4_Hi	Analog inputs 4-20mA alarm bits - Hi alarm. Measured value higher than Hi alarm threshold.
AI1_HiHi - AI4_HiHi	Analog inputs 4-20mA alarm bits - HiHi alarm. Measured value higher than HiHi alarm threshold.
AV1_LoLo, AV2_LoLo	Analog inputs 0-10V alarm bits - LoLo alarm. Measured value lower than LoLo alarm threshold.
AV1_Lo, AV2_Lo	Analog inputs 0-10V alarm bits - Lo alarm. Measured value lower than Lo alarm threshold.
AV1_Hi, AV2_Hi	Analog inputs 0-10V alarm bits - Hi alarm. Measured value higher than Hi alarm threshold.
AV1_HiHi, AV2_HiHi	Analog inputs 0-10V alarm bits - HiHi alarm. Measured value higher than HiHi alarm threshold.
AI1_ABOVE - AI4_ABOVE	Analog inputs 4-20mA alarm bits - out of measurement range. Measured value lower than 3.5mA.
AI1_BELOW - AI4_BELOW	Analog inputs 4-20mA alarm bits - out of measurement range. Measured value higher than 20.5mA.
AV1_ABOVE, AV2_ABOVE	Analog inputs 0-10V alarm bits - out of measurement range. Measured value lower than 0V.
AV1_BELOW, AV2_BELOW	Analog inputs 0-10V alarm bits - out of measurement range. Measured value higher than 10V.
P1 - P256	General purpose program flags P1 - P256

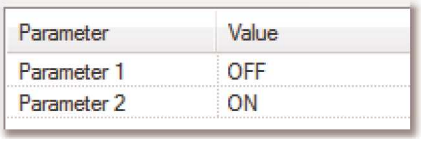
More information about available bits can be found in Memory map.

13.6 User screens programming

User screens can present static and dynamic texts, which are presented in brighter color.
Example of the MTManager configuration of User screen and Textual static parameters

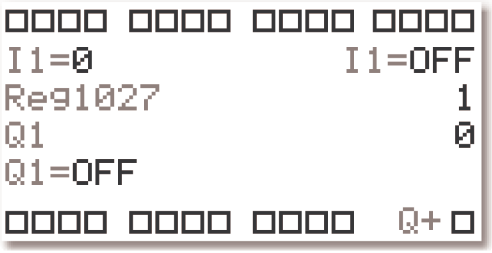


<table>
<tr><th>Parameter</th><th>Value</th></tr>
<tr><td>Display time [s]</td><td>Continuous</td></tr>
<tr><td>Show inputs</td><td>Yes</td></tr>
<tr><td>Show outputs</td><td>Yes</td></tr>
<tr><td>Line 1</td><td>I1=#IB272 I1=#TXT(IB272+1)</td></tr>
<tr><td>Line 2</td><td>Reg1027|!#HR1027</td></tr>
<tr><td>Line 3</td><td>Q1||!#HB48</td></tr>
<tr><td>Line 4</td><td>Q1=!#TXT(HB48+1:0,1)</td></tr>
</table>



Parameter	Value
Parameter 1	OFF
Parameter 2	ON

and result screen:

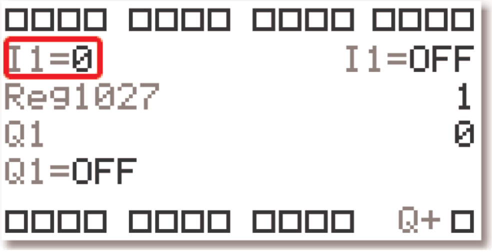


```
□□□□ □□□□ □□□□ □□□□
I1=0           I1=OFF
Reg1027             1
Q1                  0
Q1=OFF
□□□□ □□□□ □□□□  Q+ □
```

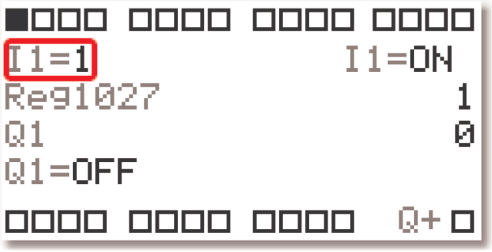
Dynamic texts are:

1. Mnemonics used in text messaging (SMS) which are described in SNCS commands syntax chapter of MT-151 HMI manual. Those commands allow to present values of single bits and registers in various forms (integer number, floating point number, hex, ASCII encoded text). Detailed description is provided in mentioned manual.

#IB272 will present 1 or 0 on screen depending on input I1 state



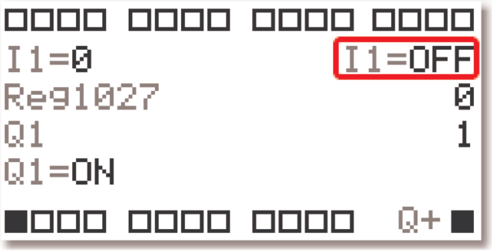
```
□□□□ □□□□ □□□□ □□□□
I1=0           I1=OFF
Reg1027             1
Q1                  0
Q1=OFF
□□□□ □□□□ □□□□  Q+ □
```



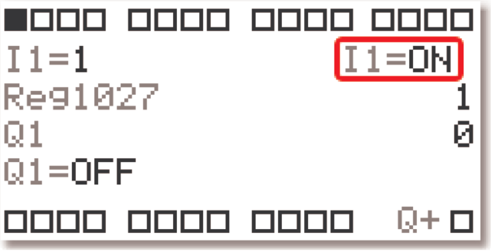
```
■□□□ □□□□ □□□□ □□□□
I1=1            I1=ON
Reg1027             1
Q1                  0
Q1=OFF
□□□□ □□□□ □□□□  Q+ □
```

2. #TXT(X[+Y]) mnemonics used for inserting texts from Resources->Constant parameters->Text list. X can be number or SMS-like command returning integer value and Y is optional offset provided as static number. Command returns text which index is equal to X (X+Y if Y is used) on the picture mentioned above.

#TXT(IB272+1) will return text index 1 if I1 is 0 (0+1=1) and text index 2 if I1 is 1 (1+1=2).



```
□□□□ □□□□ □□□□ □□□□
I1=0           I1=OFF
Reg1027             0
Q1                  1
Q1=ON
■□□□ □□□□ □□□□  Q+ ■
```

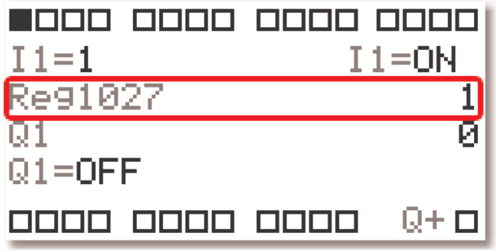


```
■□□□ □□□□ □□□□ □□□□
I1=1            I1=ON
Reg1027             1
Q1                  0
Q1=OFF
□□□□ □□□□ □□□□  Q+ □
```

3. On screen editable values – mnemonics used before preceded with exclamation mark (! sign) are editable. In addition you can limit changes range after a colon sign (: sign). By providing minimum and maximum value separated by semicolon. Range applies to actual value of register doesn't depend on presentation format (e.g. as floating point value).

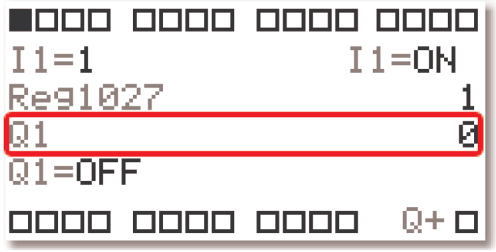
!#HR1027
presents and allows to change value in full range of register 1027 from Holding registers address space

!#HR1027:10,100
presents and allows to change value in range from 10 to 100 of register 1027 from Holding registers address space



```
■□□□ □□□□ □□□□ □□□□
I1=1            I1=ON
Reg1027             1
Q1                  0
Q1=OFF
□□□□ □□□□ □□□□  Q+□
```

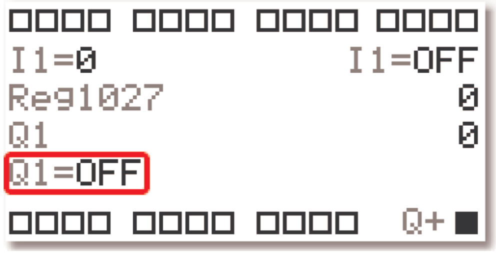
!#HB48
presents and allows changing sate of Q1 output



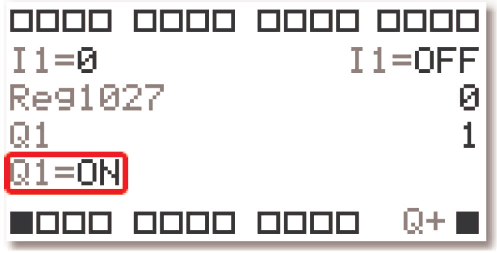
```
■□□□ □□□□ □□□□ □□□□
I1=1            I1=ON
Reg1027             1
Q1                  0
Q1=OFF
□□□□ □□□□ □□□□  Q+□
```

!#TXT(HB48+1)
presents and allows changing state of Q1 output. Output state is presented as text (index 1 for value 0 and index 2 for value 1)

!#TXT(HB48+1:0,1)
presents and allows changing state of Q1 output. Output state is presented as text (index 1 for value 0 and index 2 for value 1). In addition changes of variable value are limited to range from 0 to 1



```
□□□□ □□□□ □□□□ □□□□
I1=0           I1=OFF
Reg1027             0
Q1                  0
Q1=OFF
□□□□ □□□□ □□□□  Q+■
```

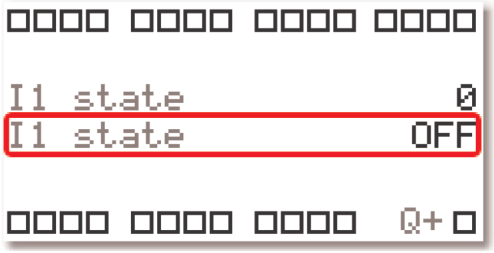


```
□□□□ □□□□ □□□□ □□□□
I1=0           I1=OFF
Reg1027             0
Q1                  1
Q1=ON
■□□□ □□□□ □□□□  Q+■
```

To make texts more clear it is possible to change text alignment from default left to right alignment. For this purpose is used vertical bar sign (| sign). Every text placed after that sign is right aligned. Entering in line text:

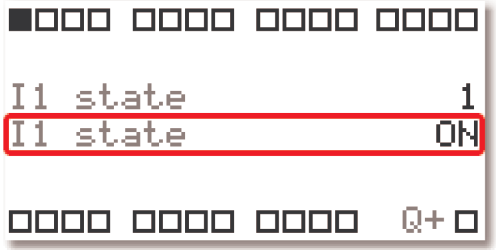
I1 state: | #IB272

when I1 is 0 will result on screen



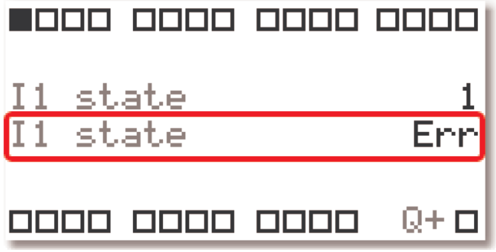
```
□□□□ □□□□ □□□□ □□□□

I1 state            0
I1 state          OFF

□□□□ □□□□ □□□□   Q+□
```

and when I1 is 1 it will give



```
■□□□ □□□□ □□□□ □□□□

I1 state            1
I1 state           ON

□□□□ □□□□ □□□□   Q+□
```

Syntax errors are signaled like on screen below



```
■□□□ □□□□ □□□□ □□□□

I1 state            1
I1 state          Err

□□□□ □□□□ □□□□   Q+□
```

13.6.1 Chart acquisition description

Each chart is using 100 registers from holding registers address space for data acquisition. Chart W1 is using registers 7000 to 7099, chart W2 registers 7100 to 7199, chart W3 registers 7200 to 7299 and chart W4 registers 7300 to 7399. Those registers are either filled automatically (parameter Data acquisition is set to *Automatic*) or by user (parameter Data acquisition is set to *User*) manually, by external device or by user program. Those registers correspond to various functions:

	Register offset	Function
Header	+0	Time in seconds before taking next data sample.
	+1	Number of samples presented on screen (max. 90)
	+2	Timestamp of last sample (LoHi) – format is exactly the same as used by RTC module +3 and available in Timestamp register
	+3	
	+4	First 14 bits are used for storing address of sampled register. 15 th bit is pointing out sampled register address space (0 – holding registers address space, 1 – analog inputs address space).
	+5	“Sample taken” flag
	+6	Sampling interval in seconds

	+7	Reserved for future functionality
	+8	Reserved for future functionality
	+9	Reserved for future functionality
Data	+10	Newest sample (left side of chart) – value between -32000 and 32000.
	...	...
	+99	Oldest sample (right side of chart) – value between -32000 and 32000.

When Data acquisition parameter is set to *User,* a number of samples are presented as units on axis X. In this mode, only data registers (+10...+99) and register responsible for the number of samples (+1) presented on screen need to be set.

When a chart screen is displayed user can press OK button to show the chart name that can be configured.
The module restarts may cause discontinuity of data acquisition.

13.7 Memory map

All accessible from remote and by program resources of MT-151 HMI module were collected in four address spaces: Binary Inputs, Input Register, Binary Outputs and Holding Registers. Spaces of Binary Inputs and Input Register and spaces of Binary Outputs and Holding Registers are connected in pairs and contain the same resources. The difference between spaces is in the way of accessing the resources - Binary Inputs and Outputs are used for accessing individual bits and groups of bits while Input and Holding Registers address spaces allow access to the full registers and register groups. This difference results in a different way of addressing. In the Input Registers and Holding Registers address spaces each address is assigned to the each register while the Binary Inputs and Outputs address spaces each address corresponds to individual bit. The memory map tables are arranged by their addresses for addressing registers. To calculate the addresses of the individual bits in the Binary Inputs, use the following equation:

register_address * 16 + bit_position = bit_address

To calculate the addresses of the individual bits in the Binary Outputs, use the following equation:

register_address * 16 + bit_position +10000 = bit_address

For example, in the PRG_FLG1 register from Input Registers address space (address 2) on position 7 is located FS1_gprs bit indicating GPRS logon. Using that formula, you can specify the address of FS1_gprs bit in Binary Inputs address space as follows:

2 * 16 +7 = 39.

13.7.1 Input registers/binary inputs address space

Input registers/binary inputs address space (read only), access using Modbus RTU and TCP functions 2 and 4																			
Address		Bit																Name	Description
Reg	Bit [0]	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0	0	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	Reserved
1	16	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	Reserved
2	32	---	---	FS1_3G	FS1_lowacu	FS1_pwr	FS1_acu	FS1_sd	FS1_usb	FS1_gprs	FS1_gsm	FS1_q+	FS1_prog	FS1_ovr	FS1_stop	FS1_new	FS1_fs	PRG_FLG1	Status flags: FS1_fs - first program scan FS1_new - informs about loading new program FS1_stop = 1 when program is stopped FS1_ovr = 1 delay of program running FS1_prog = 1 when error in user program FS1_q+ = 1 when binary outputs are not powered FS1_gsm = 1 when module is logged into GSM network FS1_gprs = 1 when module is logged into GPRS network FS1_usb = 1 when USB cable is connected FS1_sd = 1 when microSD card is installed in slot FS1_acu = 1 when battery is connected FS1_pwr = 1 - main power is connected FS1_lowacu = 1 low voltage battery FS1_3G =1 - 3G modem
3	48	---	SIM_USE	ROAMING	PIN_OK	PIN_ATTE	PIN_WRONG	PUK_REQ	NO_SIM	---	---	---	---	---	SD_write	FS2_rtc_min	FS2_rtc_sec	PRG_FLG2	System flags: FS2_rtc_sec - 1Hz impulsator (1 second) FS2_rtc_min - 1/60Hz impulsator (1 minute)

																			SD_write - writing on SD card in progress now NO_SIM - no SIM card detected PUK_REQ - PUK code required PIN_WRONG - wrong PIN code PIN_ATTE - Two attempts made PIN_OK - Pin code correct ROAMING - module in roaming SIM_USE which card is used = 0(SIM1), 1(SIM2)
4	64	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}	2^{-16}	RTC_FSEC	RTC - fraction of second
5	80	int16(LoHi)																RTC_Sec	RTC - second (0 - 59)
6	96	int16(LoHi)																RTC_Min	RTC - minute (0 - 59)
7	112	int16(LoHi)																RTC_Hour	RTC - hour (0 - 23)
8	128	int16(LoHi)																RTC_DofW	RTC - day of week (1 - Sunday, 7 - Saturday)
9	144	int16(LoHi)																RTC_Day	RTC - day of month (1-31)
10	160	int16(LoHi)																RTC_Mon	RTC - month (1-12)
11	176	int16(LoHi)																RTC_Year	RTC - year (2000-2099)
12	192	int32(LoHi)																RTC	Timestamp
13	208																		
14	224	int32(LoHi)																ON_TMR	Time in seconds since power on
15	240																		
16	256	CT16	CT15	CT14	CT13	CT12	CT11	CT10	CT9	CT8	CT7	CT6	CT5	CT4	CT3	CT2	CT1	CLOCK	Synchronous timers flags (set for 1 program cycle)
17	272	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	BIN	Binary inputs
18	288	---	---	---	---	IQ12	IQ11	IQ10	IQ9	IQ8	IQ7	IQ6	IQ5	IQ4	IQ3	IQ2	IQ1	BFB	Binary outputs/inputs pin state
19	304	int16(LoHi)																AI1_raw	Analog input AI1 measurement [mA]
20	320	int16(LoHi)																AI2_raw	Analog input AI2 measurement [mA]
21	336	int16(LoHi)																AI3_raw	Analog input AI3

																			measurement [mA]
22	352	int16(LoHi)																AI4_raw	Analog input AI4 measurement [mA]
23	368	int16(LoHi)																AI1	Analog input AI1 measurement [engineering units]
24	384	int16(LoHi)																AI2	Analog input AI2 measurement [engineering units]
25	400	int16(LoHi)																AI3	Analog input AI3 measurement [engineering units]
26	416	int16(LoHi)																AI4	Analog input AI4 measurement [engineering units]
27	432	---	AI2_ABOVE	AI2_BELOW	AI2_DBD	AI2_HiHi	AI2_Hi	AI2_LoLo	AI2_Lo	---	AI1_ABOVE	AI1_BELOW	AI1_DBD	AI1_HiHi	AI1_Hi	AI1_LoLo	AI1_Lo	ALM_I12	Alarm bits for AI1 - AI4 analog inputs: AIx_ABOVE - measurement above 20.5mA AIx_BELOW - measurement below 3.5mA AIx_DBD - measurement change higher than deadband AIx_LoLo - LoLo alarm flag AIx_Lo - Lo alarm flag AIx_Hi - Hi alarm flag AIx_HiHi - HiHi alarm flag
28	448	---	AI4_ABOVE	AI4_BELOW	AI4_DBD	AI4_HiHi	AI4_Hi	AI4_LoLo	AI4_Lo	---	AI3_ABOVE	AI3_BELOW	AI3_DBD	AI3_HiHi	AI3_Hi	AI3_LoLo	AI3_Lo	ALM_I34	
29	464	int16(LoHi)																AV1_raw	Analog input AV1 measurement [mV]
30	480	int16(LoHi)																AV2_raw	Analog input AV2 measurement [mV]
31	496	int16(LoHi)																AV1	Analog input AV1 measurement [engineering units]
32	512	int16(LoHi)																AV2	Analog input AV2 measurement [engineering units]

33	528	---	AV2_ABOVE	AV2_BELOW	AV2_DBD	AV2_HiHi	AV2_Hi	AV2_LoLo	AV2_Lo	---	AV1_ABOVE	AV1_BELOW	AV1_DBD	AV1_HiHi	AV1_Hi	AV1_LoLo	AV1_Lo	ALM_V12	Alarm bits for AV1 - AV2 analog inputs: AVx_ABOVE - measurement above 10V AVx_BELOW - measurement below 0V AVx_DBD - measurement change higher than deadband AVx_LoLo - LoLo alarm flag AVx_Lo - Lo alarm flag AVx_Hi - Hi alarm flag AVx_HiHi - HiHi alarm flag
34	544	int16(LoHi)																AVAKU	Battery voltage [mV]
35	560	int16(LoHi)																AVZ	Power supply voltage [mV]
36	576	CNT16_OVFL	CNT15_OVFL	CNT14_OVFL	CNT13_OVFL	CNT12_OVFL	CNT11_OVFL	CNT10_OVFL	CNT9_OVFL	CNT8_OVFL	CNT7_OVFL	CNT6_OVFL	CNT5_OVFL	CNT4_OVFL	CNT3_OVFL	CNT2_OVFL	CNT1_OVFL	CNT_OVFL	Counter overflow bits (set for 1 program cycle)
37	592	CK16	CK15	CK14	CK13	CK12	CK11	CK10	CK9	CK8	CK7	CK6	CK5	CK4	CK3	CK2	CK1	CKx	Asynchronous timers flags (set for 1 program cycle)
38	608	P1SL16_ok	P1SL15_ok	P1SL14_ok	P1SL13_ok	P1SL12_ok	P1SL11_ok	P1SL10_ok	P1SL9_ok	P1SL8_ok	P1SL7_ok	P1SL6_ok	P1SL5_ok	P1SL4_ok	P1SL3_ok	P1SL2_ok	P1SL1_ok	...	SLx_ok=1 when data block 1 - 16 communication on serial port number 1 is OK
39	624									C8	C7	C6	C5	C4	C3	C2	C1	...	Program counters Cx overflow flags
40	640									T8	T7	T6	T5	T4	T3	T2	T1	...	Program timers Tx flags
41	656	TSL16_ok	TSL15_ok	TSL14_ok	TSL13_ok	TSL12_ok	TSL11_ok	TSL10_ok	TSL9_ok	TSL8_ok	TSL7_ok	TSL6_ok	TSL5_ok	TSL4_ok	TSL3_ok	TSL2_ok	TSL1_ok	...	TSLx_ok=1 - when data block x communication on Ethernet port is OK
42	672	MT2MT_16	MT2MT_15	MT2MT_14	MT2MT_13	MT2MT_12	MT2MT_11	MT2MT_10	MT2MT_9	MT2MT_8	MT2MT_7	MT2MT_6	MT2MT_5	MT2MT_4	MT2MT_3	MT2MT_2	MT2MT_1	...	MTx bit informs about receiving data to MT2MT buffer from device, which IP number is saved on x position on

43	688	MT2MT_32	MT2MT_31	MT2MT_30	MT2MT_29	MT2MT_28	MT2MT_27	MT2MT_26	MT2MT_25	MT2MT_24	MT2MT_23	MT2MT_22	MT2MT_21	MT2MT_20	MT2MT_19	MT2MT_18	MT2MT_17	...	Authorized -> IP list
44	704	P2SL16_ok	P2SL15_ok	P2SL14_ok	P2SL13_ok	P2SL12_ok	P2SL11_ok	P2SL10_ok	P2SL9_ok	P2SL8_ok	P2SL7_ok	P2SL6_ok	P2SL5_ok	P2SL4_ok	P2SL3_ok	P2SL2_ok	P2SL1_ok	...	P2SLx_ok=1 when data block 1 -16 communication on serial port number 2 is OK
45	720	P1SL32_ok	P1SL31_ok	P1SL30_ok	P1SL29_ok	P1SL28_ok	P1SL27_ok	P1SL26_ok	P1SL25_ok	P1SL24_ok	P1SL23_ok	P1SL22_ok	P1SL21_ok	P1SL20_ok	P1SL19_ok	P1SL18_ok	P1SL17_ok	...	P1SLx_ok=1 when data block 17 - 32 communication on serial port number 1 is OK
46	736	P2SL32_ok	P2SL31_ok	P2SL30_ok	P2SL29_ok	P2SL28_ok	P2SL27_ok	P2SL26_ok	P2SL25_ok	P2SL24_ok	P2SL23_ok	P2SL22_ok	P2SL21_ok	P2SL20_ok	P2SL19_ok	P2SL18_ok	P2SL17_ok	...	P2SLx_ok=1 when data block 17 - 32 communication on serial port number 2 is OK
...	...	...																...	...
80	1280	int32(LoHi)																FL_I1	Current flow value I1
81	1296																		
82	1312	int32(LoHi)																FL_I2	Current flow value I2
83	1328																		
84	1344	int32(LoHi)																FL_I3	Current flow value I3
85	1360																		
86	1376	int32(LoHi)																FL_I4	Current flow value I4
87	1392																		
88	1408	int32(LoHi)																FL_ENG_I1	Calculated flow value I1 (engineering units)
89	1424																		
90	1440	int32(LoHi)																FL_ENG_I2	Calculated flow value I2 (engineering units)
91	1456																		
92	1472	int32(LoHi)																FL_ENG_I3	Calculated flow value I3 (engineering units)
93	1488																		
94	1504	int32(LoHi)																FL_ENG_I4	Calculated flow value I4 (engineering units)
95	1520																		
...	...	...																...	...
127	2032	---	---	---	---	---	---	---	---	Rst_	Rst_Fir	Rst_	---	Rst_Po	---	Rst_W	---		Last restart cause:

										OS	mware	Config		wer_on		atchdog			Rst_OS - restarted by system Rst_Firmware - restart after firmware update Rst_Config - restart after configuration update Rst_Power_on - restart after power on Rst_Watchdog - watchdog restart
...	...	...																...	...
130	2080	int16(LoHi)																SMS_CNT	SMS send since power-on
131	2096	int16(LoHi)																SMS_DAILY_CNT	Daily SMS counter
132	2112	int16(LoHi)																SYG_LEV	GSM signal level [%]
133	2128	int32(LoHi)																FIRMWARE_VER	Firmware version y.xx.zz (encoded in HEX)
134	2144																		
135	2160	int16(LoHi)																PRG_CLINE	Number of program lines executed in previous program cycle
136	2176	int16(LoHi)																PRG_CTIME	Time of execution of previous program cycle [ms]
...	...	...																...	...
138	2208	int16(LoHi)																PAR_1	Parameter 1
...	...	...																...	...
265	4240	int16(LoHi)																PAR_128	Parameter 128
...	...	...																...	...
350		MB16	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MBX	MBX=1 when module receive correct answer from M-BUS device X
351		PORT 1 - M-BUS -Device 1 correct variable reading																	1 on bit X means correct reading variable number X
...	...	...																...	...
366		PORT 1 - M-BUS -Device 16 correct variable reading																	1 on bit X means correct reading variable number X
370																			

13.7.2 Holding registers/binary outputs address space

Holding registers/binary outputs address space (read/write), access using Modbus RTU and TCP functions: read - 1, 3; write - 5, 6, 15, 16)

Address		Bit																Name	Description
Reg	Bit [0]	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0	0	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	Reserved
1	16	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	Reserved
2	32	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	PS1_STOP	SYS_FLG1	PS1_STOP - writing 1 stops program, 0 - starts program
3	48			---	---	Q12	Q11	Q10	Q9	Q8	Q7	Q6	Q5	Q4	Q3	Q2	Q1	BOUT	Bits controlling binary outputs 1 - output set to high level, 0 - low level
4	64	int32(LoHi)																CNT1	32-bit counter register
5	80																		
6	96	int32(LoHi)																CNT2	32-bit counter register
7	112																		
8	128	int32(LoHi)																CNT3	32-bit counter register
9	144																		
10	160	int32(LoHi)																CNT4	32-bit counter register
11	176																		
12	192	int32(LoHi)																CNT5	32-bit counter register
13	208																		
14	224	int32(LoHi)																CNT6	32-bit counter register
15	240																		
16	256	int32(LoHi)																CNT7	32-bit counter register
17	272																		
18	288	int32(LoHi)																CNT8	32-bit counter register
19	304																		
20	320	int32(LoHi)																CNT9	32-bit counter register
21	336																		
22	352	int32(LoHi)																CNT10	32-bit counter register
23	368																		
24	384	int32(LoHi)																CNT11	32-bit counter register
25	400																		
26	416	int32(LoHi)																CNT12	32-bit counter

27	**432**			register
28	**448**	int32(LoHi)	CNT13	32-bit counter register
29	**464**			
30	**480**	int32(LoHi)	CNT14	32-bit counter register
31	**496**			
32	**512**	int32(LoHi)	CNT15	32-bit counter register
33	**528**			
34	**544**	int32(LoHi)	CNT16	32-bit counter register
35	**560**			
36	**576**	int32(LoHi)	REG_CK1	CK1 asynchronous timer - current value
37	**592**			
38	**608**	int32(LoHi)	REG_CK2	CK2 asynchronous timer - current value
39	**624**			
40	**640**	int32(LoHi)	REG_CK3	CK3 asynchronous timer - current value
41	**656**			
42	**672**	int32(LoHi)	REG_CK4	CK4 asynchronous timer - current value
43	**688**			
44	**704**	int32(LoHi)	REG_CK5	CK5 asynchronous timer - current value
45	**720**			
46	**736**	int32(LoHi)	REG_CK6	CK6 asynchronous timer - current value
47	**752**			
48	**768**	int32(LoHi)	REG_CK7	CK7 asynchronous timer - current value
49	**784**			
50	**800**	int32(LoHi)	REG_CK8	CK8 asynchronous timer - current value
51	**816**			
52	**832**	int32(LoHi)	REG_CK9	CK9 asynchronous timer -

53	**848**																		current value
54	**864**	int32(LoHi)																REG_CK10	CK10 asynchronous timer - current value
55	**880**																		
56	**896**	int32(LoHi)																REG_CK11	CK11 asynchronous timer - current value
57	**912**																		
58	**928**	int32(LoHi)																REG_CK12	CK12 asynchronous timer - current value
59	**944**																		
60	**960**	int32(LoHi)																REG_CK13	CK13 asynchronous timer - current value
61	**976**																		
62	**992**	int32(LoHi)																REG_CK14	CK14 asynchronous timer - current value
63	**1008**																		
64	**1024**	int32(LoHi)																REG_CK15	CK15 asynchronous timer - current value
65	**1040**																		
66	**1056**	int32(LoHi)																REG_CK16	CK16 asynchronous timer - current value
67	**1072**																		
68	**1088**	int16(LoHi)																RESTART	Module restart counter
69	**1104**									CLK_C8	CLK_C7	CLK_C6	CLK_C5	CLK_C4	CLK_C3	CLK_C2	CLK_C1		C1 - C8 program counters counting inputs (active on rising edge)
70	**1120**									RST_C8	RST_C7	RST_C6	RST_C5	RST_C4	RST_C3	RST_C2	RST_C1		C1 - C8 program counters resetting inputs (active on 1)
71	**1136**									EN_T8	EN_T7	EN_T6	EN_T5	EN_T4	EN_T3	EN_T2	EN_T1		T1 - T8 program timers enable

																			bits (active on 1)
72	**1152**									RST_T8	RST_T7	RST_T6	RST_T5	RST_T4	RST_T3	RST_T2	RST_T1		T1 - T8 program timers resetting bits (active on 1)
...	...	...																...	...
100	**1600**	P16	P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	PFLG	General purpose program flags
...	...	...																	
115	**1840**	P256	P255	P254	P253	P252	P251	P250	P249	P248	P247	P246	P245	P244	P243	P242	P241		
116	**1856**	int16(LoHi)																REG1	General purpose 16-bit register
...	...	...																...	...
371	**5936**	int16(LoHi)																REG256	General purpose 16-bit register
372	**5952**	int32(LoHi)																DREG1	General purpose 32-bit register (signed value)
373	**5968**																		
...	**...**	...																...	...
626	**10016**	int32(LoHi)																DREG128	General purpose 32-bit register (signed value)
627	**10032**																		
...	...	...																...	...
630	**10080**	int16(LoHi)																PV_C1	C1 program counter threshold value
631	**10096**	int16(LoHi)																PV_C2	C2 program counter threshold value
632	**10112**	int16(LoHi)																PV_C3	C3 program counter threshold value
633	**10128**	int16(LoHi)																PV_C4	C4 program counter threshold value
634	**10144**	int16(LoHi)																PV_C5	C5 program counter

				threshold value
635	**10160**	int16(LoHi)	PV_C6	C6 program counter threshold value
636	**10176**	int16(LoHi)	PV_C7	C7 program counter threshold value
637	**10192**	int16(LoHi)	PV_C8	C8 program counter threshold value
638	**10208**	int16(LoHi)	PV_T1	T1 program timer threshold value
639	**10224**	int16(LoHi)	PV_T2	T2 program timer threshold value
640	**10240**	int16(LoHi)	PV_T3	T3 program timer threshold value
641	**10256**	int16(LoHi)	PV_T4	T4 program timer threshold value
642	**10272**	int16(LoHi)	PV_T5	T5 program timer threshold value
643	**10288**	int16(LoHi)	PV_T6	T6 program timer threshold value
644	**10304**	int16(LoHi)	PV_T7	T7 program timer threshold value
645	**10320**	int16(LoHi)	PV_T8	T8 program timer threshold value
646	**10336**	int16(LoHi)	REG_C1	C1 program counter current value
647	**10352**	int16(LoHi)	REG_C2	C2 program counter

				current value
648	**10368**	int16(LoHi)	REG_C3	C3 program counter current value
649	**10384**	int16(LoHi)	REG_C4	C4 program counter current value
650	**10400**	int16(LoHi)	REG_C5	C5 program counter current value
651	**10416**	int16(LoHi)	REG_C6	C6 program counter current value
652	**10432**	int16(LoHi)	REG_C7	C7 program counter current value
653	**10448**	int16(LoHi)	REG_C8	C8 program counter current value
654	**10464**	int16(LoHi)	REG_T1	T1 program timer current value
655	**10480**	int16(LoHi)	REG_T2	T2 program timer current value
656	**10496**	int16(LoHi)	REG_T3	T3 program timer current value
657	**10512**	int16(LoHi)	REG_T4	T4 program timer current value
658	**10528**	int16(LoHi)	REG_T5	T5 program timer current value
659	**10544**	int16(LoHi)	REG_T6	T6 program timer current value
660	**10560**	int16(LoHi)	REG_T7	T7 program timer

				current value
661	**10576**	int16(LoHi)	REG_T8	T8 program timer current value
...	...	...	...	...
1024	**16384**	int16(LoHi)(trapSourceStatus - SNMP variable)	HR1024	General purpose 16-bit register
1025	**16400**	int16(LoHi)(trapSourceValue - SNMP variable)	HR1025	General purpose 16-bit register
1026	**16416**	int16(LoHi)(trapSourceType - SNMP variable)	HR1026	General purpose 16-bit register
1027	**16432**	int16(LoHi)(trapSourceIndex - SNMP variable)	HR1027	General purpose 16-bit register
1028	**16448**	int16(LoHi)(trapSourceName - SNMP variable)	HR1028	General purpose 16-bit register
...	...	...	...	...
...	...	...	...	...
1185	**18986**	int16(LoHi)(trapSourceStatus - SNMP variable used when data source are Registers)	HR1185	General purpose 16-bit register
1186	**18992**	int16(LoHi)(trapSourceValue - SNMP variable used when data source are Registers)	HR1186	General purpose 16-bit register
1187	**19008**	int16(LoHi)(trapSourceType - SNMP variable used when data source are Registers)	HR1187	General purpose 16-bit register
1188	**19024**	int16(LoHi)(trapSourceIndex - SNMP variable used when data source are Registers)	HR1188	General purpose 16-bit register
1189	19040	int16(LoHi)	HR1189	General purpose 16-bit register zeroed at reset
...	...	...	...	...
4095	65520	Last register that is available as bitwise		
...	...	...	...	...
5000	-	PORT 1 - Flex Serial data receiving buffer	P1RCV_NO	Counter received data
5001	-	PORT 1 - Flex Serial data receiving buffer	P1RCV_B1	First register of

				receiving buffer
...	...	...	...	...
5256	-	PORT 1 - Flex Serial data receiving buffer	P1RCV_B256	Last register of receiving buffer
5257	-	PORT 1 - Flex Serial data receiving buffer	P1RCV_ERR	Receiving buffer status
...	...	...	...	...
5500	-	PORT 1 - Flex Serial data sending buffer	P1SND_NO	Counter sent data
5501	-	PORT 1 - Flex Serial data sending buffer	P1SND_B1	First register of sending buffer
...	...	...	...	...
5756	-	PORT 1 - Flex Serial data sending buffer	P1SND_B256	Last register of sending buffer
...	...	...	...	...
6000	-	PORT 2 - Flex Serial data receiving buffer	P2RCV_NO	Counter received data
6001	-	PORT 2 - Flex Serial data receiving buffer	P2RCV_B1	First register of receiving buffer
...	...	...	...	...
6256	-	PORT 2 - Flex Serial data receiving buffer	P2RCV_B256	Last register of receiving buffer
6257	-	PORT 2 - Flex Serial data receiving buffer	P2RCV_ERR	Receiving buffer status
...	...	...	...	...
6500	-	PORT 2 - Flex Serial data sending buffer	P2SND_NO	Counter sent data
6501	-	PORT 2 - Flex Serial data sending buffer	P2SND_B1	First register of sending buffer
...	...	...	...	...
6756	-	PORT 2 - Flex Serial data sending buffer	P2SND_B256	Last register of sending buffer
...	...	...	...	...
8191		int16(LoHi)	HR8191	General purpose 16-bit register

				zeroed at reset

The bold address numbers means those Registers are nonvolatile.

14 About User Manual

User Manual for
Telemetry Module MOBICON MT-151 HMI V2

Copyright © 2017 Inventia Sp. z o.o.
v2.00.04
September 5th, 2017

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.