metrilog

Remote Terminal Unit T707

User Guide

Table Of Contents

Introduction

The Remote Terminal Unit (RTU) model T707 is a low power GSM/GPRS based communication device, that includes flexible bus/serial interfaces and a data-logging unit. The T707 RTU is offered in several variants, depending on the protocol and the supported Input/Output (I/O) interface:

- SDI-12 (native or over RS-485);

- Davis Instruments' Vantage Pro (RS-232);

- Thies TDL14 data-logger (RS-232 or RS-422).

If not otherwise specified, this manual refers to the basic SDI-12 version.

The T707 RTU supports a maximum of 30 SDI-12 sensors with a total of 50 sampled values (tags). The SDI-12 version also supports output tags by using specially designed, custom "X"

commands (e.g. for switching valves). The Davis and Thies versions support one sensor (i.e. the data-logger) with a total of 50 sampled values (tags). The internal data logging memory (FIFO—First In, First Out) can store up to 50'000 values, the older values being overwritten when the memory fills up.

The SDI-12 variant conforms to the SDI-12 specification version 1.3. For additional information on the SDI-12 bus, please consult the following document: "SDI-12, A Serial-Digital Interface Standard for Microprocessor-Based Sensors, Version 1.3". The document can be found on the SDI-12 Support Group's web site at *http://www.sdi-12.org*.

While the physical connexion to the bus can be configured either for native SDI-12 signaling or RS-485, the protocol employed is always SDI-12.

This manual addresses the installation, configuration and maintenance of the T707 RTU.

Installation

The T707 RTU has been designed to operate in conjunction with the Metrilog M2M Gateway and Metrilog M2M Services. Before installing a T707 RTU in the field, it must be entered on the Metrilog M2M Gateway. Proceed as follows (you need administrator rights to add an RTU):

- Log-in on the Metrilog M2M Gateway;
- If you have sub-realms or sub-areas, navigate to the realm or area where you want to enter the new RTU;
- Select the realm or area in the left pane of the Web interface and select "New Remote Unit" on the right pane;
- Fill in the relevant information; you will find the RTU's "Serial Number" on the metallic label affixed on one side of the T707 RTU;
- Choose a suitable name for the "RTU Name", usually a name representative for the location the RTU is installed at;
- The "Login Name" and "Login Password" are used later by the RTU to authenticate to the M2M Gateway; you can enter whatever name and password you deem appropriate, however it is recommended to use a name similar to the RTU name (max. 10 characters);
- Select the appropriate RTU template (e.g. T707-SDI);
- Click OK to finish.

Although it is not required for the proper operation of the RTU, you will need to attach sensors and tags to the RTU, depending on what sensors you plan to install in the field.

T707 for SDI-12 Sensors

The T707 RTU uses a four-wire cable carrying the SDI-12 bus signals and the bus power supply (typically 12 volt). Alternatively the bus can be switched to RS-485 levels, but this is

allowed only if all devices on the bus support it. You cannot mix SDI-12 and RS-485 sensors on the same bus.

To switch the unit from SDI-12 (factory default) to RS-485 you must open the lid and change the on-board jumper from SDI to 485 (see also the picture below). For SDI-12 sensors, only the pins 1, 2 and 4 of the connector should be used (the brown, white and black wires of the cable).
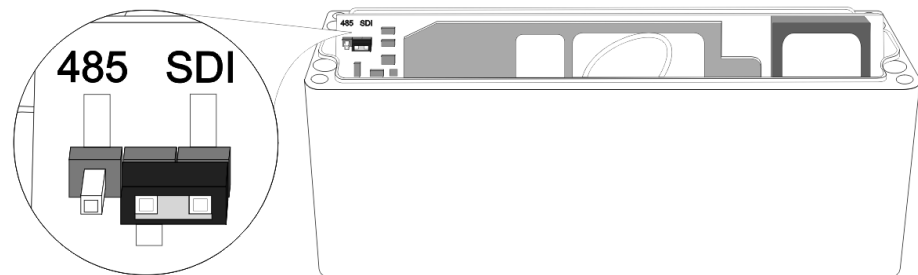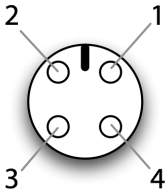
There are several possible bus configurations, see the table below for details.

	SDI-12 (default)
	RS-485, terminating unit; that means this device is situated at one end of the RS-485 bus and the internal terminating resistor (110 ohm) will be enabled.
	RS-485, on the bus but not at one of its ends; in this position, no terminating resistor is enabled.

Note: after opening the unit, make sure that the O-ring is properly seated before fastening the lid.

M12 SDI-12/RS-485 Plug
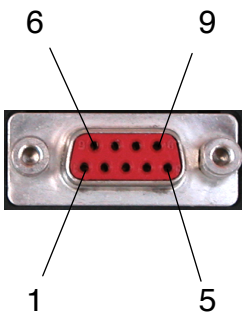
The pin-out of the SDI-12/RS-485 plug is shown below.

1	+Vbatt (6.5 to 20 Volt)	brown
2	-Vbatt, GND	white
3	RS-485 A (data)	blue
4	SDI-12, RS-485 B (data)	black

Notes: The colors in the last column are valid for the standard M12 cable delivered with the device. Pin 3 is not used in native SDI-12 mode.

RS-232 Service Connector

The RS-232 connector is used for maintenance and service. You should not have to deal with this connector unless indicated so by Metrilog support personnel. For additional information on the use of this connector, see the "Commands" section of this manual. The pin-out of the input connector is given below (as seen from the outside).

1	DCD - Data Carrier Detect
2	RD - Receiver Data
3	TD - Transmit Data
4	DTR - Data Terminal Ready
5	GND - Ground, chassis
6	DSR - Data Set Ready
7	RTS - Request To Send
8	CTS - Clear To Send
9	RI - Ring Indicator

T707 for Davis Vantage Pro

The following instructions are valid for the model running the T707-DAV firmware, to be installed in conjunction with a Vantage Pro weather station (manufactured by Davis Instruments). The T707 RTU can be installed both indoors and outdoors. **The M905 Connexion Box, as well as the mains adapter cannot be installed outdoors!**

In order to connect a T707 RTU to the Vantage Pro Console, a WeatherLink data cable from Davis Instruments is required. Note that for this configuration the data collected by the Davis Console is already logged into the WeatherLink data-logger, thus two levels of storage are used: one offered by the Davis System and a second by the T707 RTU.

The following diagram depicts the connections between the T707 RTU, the M905 Connexion Box and the mains adapter:
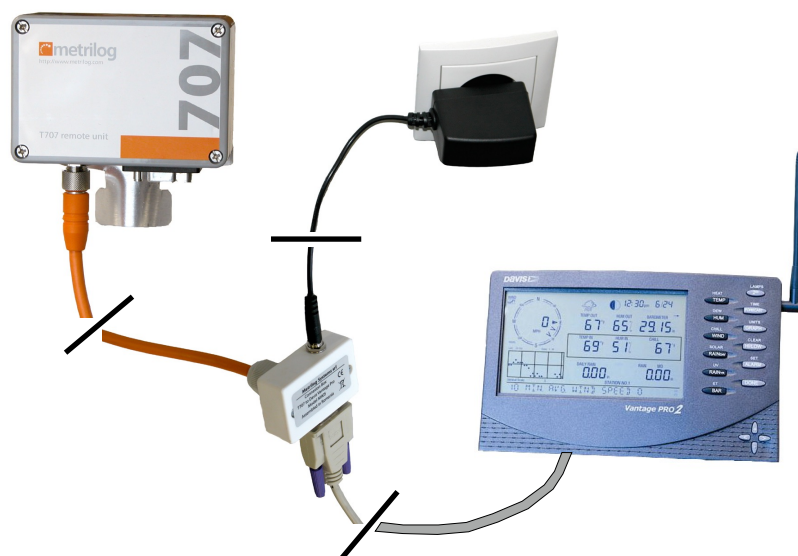
The service connector has the same significance as for the SDI-12 Variant.

T707 for Thies TDL14 Data-logger

The following instructions are valid for the model running the T707-THI software, to be installed in conjunction with a Thies model TDL14 data logger.

Attaching the Connexion Cable to the Thies Data-logger

If you are performing the operations below while the Thies data-logger is powered, make sure the other end of the cable is NOT connected to the T707! Follow the steps shown in the pictures below:

1. Remove the top of a free cable gland
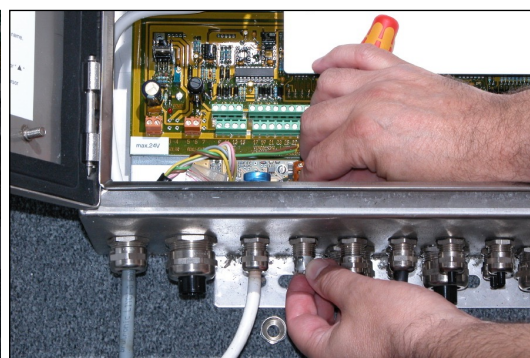
2. With a screwdriver, remove the metal shield
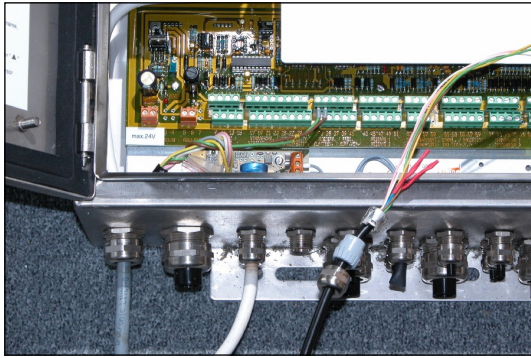
3. Mount gland parts on the cable as shown
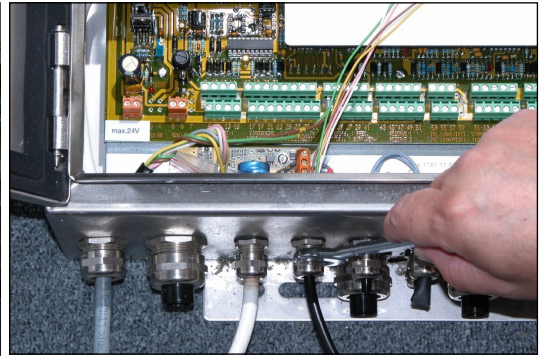
4. Pull the cables and fasten the gland
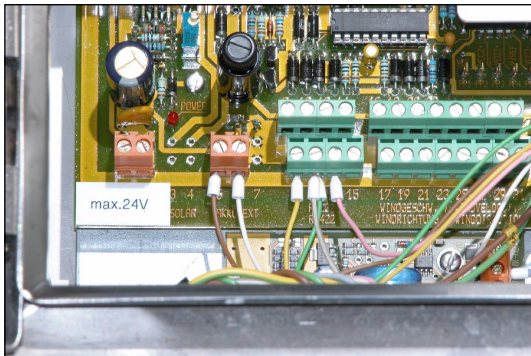
5. Attach the wires as shown (RS-232 version)
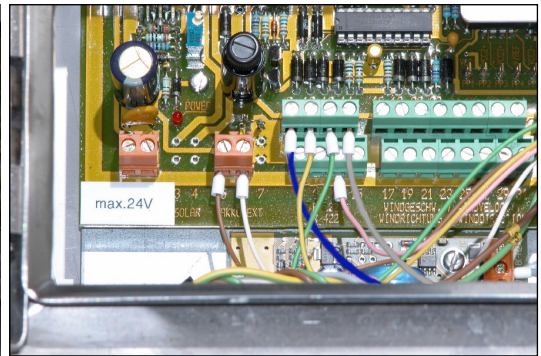
6. Attach the wires as shown (RS-422 version)

After all the wires have been properly connected, you can now attach the other end of the cable to the T707 unit.

Notes:

- Before connecting the T707 cable wires to the screw posts, make sure there are no other wires already connected! The T707 can not be operated in parallel to another device/ modem.

- If the cable gland does not provide a ground connection, a separate wire should be used to connect the cable shield to the ground (i.e. to the white wire).

- The switch between RS232 and RS422 is done in software through the Metrilog M2M Portal; it can be also done using a command issued through the service port, as illustrated below:

```
: attr rs422 true
```

 For more details see the also the section "Commands", as well as the section "RTU Attributes".

The table below shows the connections between the T707 cable and the TDL14 data-logger in both RS-232 and RS-422 modes.

T707 Cable	Thies TDL14	
	RS-232	RS-422
Brown	5 (+12V)	5 (+12V)
White	6 (GND)	6 (GND)
Yellow	9 (RxD)	10 (RxD-)
Green	11 (TxD)	12 (TxD+)
Pink	13 GND	13 GND
Gray	11 (TxD)	14 (TxD-)
Blue	N.C. (insulated!)	8 (RxD+)

Mechanical Installation

This section is valid for all variants.

The T707 unit should be preferably mounted outdoors on a 4 cm diameter mast or support by means of the hose clamp supplied. In this case fasten first the optional metallic mounting fixture to the T707 enclosure using the two M6 screws and the included spacers (see figure below). If the unit is not installed outdoors, rather left on a table or a cabinet, the metallic fixture is not required.
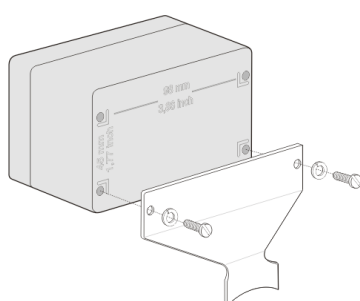
Fig. 1: Metallic fixture mount   Fig. 2: Recommended   Fig. 3: Acceptable
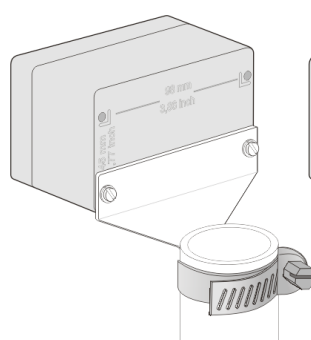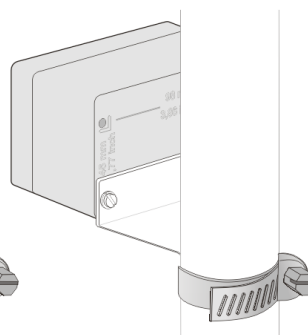
It is recommended to place the top of the unit (where the built-in antenna resides) not too close to metallic objects, especially to the mast itself. Ideally the unit should be mounted at the top of the mast (see also figures 2 and 3). Use the supplied tie wraps to fasten the cables to the mast after the mechanical installation is completed.

How it Works

After being powered up, a not-initialized RTU will first check the SIM card and extract the provider specific information (APN, Name, Password). This information, together with the initial address of the Metrilog M2M Gateway is stored in a table in the RTU's firmware.

Note: *The RTU must be already entered on the Metrilog M2M Gateway as described at the beginning of this section.*

At this point, the RTU is able to connect via GPRS and the Internet to the M2M Gateway. If the connection succeeds, the RTU retrieves its complete configuration from the Gateway, including its account data and the attached sensors. The account data is based on the credentials offered by information contained in the SIM card, the IMEI, and/or the serial number of the device. If the SIM card is not known to the M2M Gateway, then the RTU will not be able to log in and retrieve configuration and account data. The error messages in the log file document such situations, as well as other similar potential issues. You can read more about this in the "Error, Warning and Informational Messages" section.

Configuration

The T707 RTU can be configured automatically by the Metrilog M2M Gateway (preferred) or manually through the Service Port using a serial cable and a PC.

The Metrilog M2M Gateway

An important service offered by the M2M Gateway is the configuration of the T707 RTU. The whole behavior of an RTU is defined on the Gateway. The RTU parameters can be defined and/or modified at any time on the M2M Gateway after which they are sent to the RTU as soon as a connection is made.

Transferring data by means of GPRS can only be done by "pushing" it to a server. A GPRS connection can only be initiated by the RTU and not the other way round. That means that "polling" the RTU by the Gateway is not possible. In order to overcome this issue, the T707 RTU implements a flexible scheduler that specifies when and how often the RTU should initiate a connection to the M2M Gateway. Similar to many other parameters on the RTU, the scheduler is remote configurable through the M2M Gateway.

By changing a parameter on the Gateway, a "task" will be generated and added to a queue for that particular RTU. When the RTU connects, the task will be passed along and the RTU will execute and acknowledge it. If a task must be immediately executed, i.e. the time to the next scheduled connection is too long, the RTU can be "forced" to connect by sending it a specially constructed wake-up SMS (see also the "SMS Commands" section).

The functionality of an RTU is defined by a collection of attributes. A list with the description of the T707 RTU's attributes is given in the "Attributes" section of this document. The Web based User Interface of the M2M Gateway offers an easy and intuitive method to add/ modify/delete attributes. However, standard devices (e.g. the T707 RTU family, as well as a large collection of sensors), are already predefined by means of templates, so you should not need to add/change attributes.

Note: *Improper manipulation of the attributes can lead to malfunction of the RTUs. You must first understand what the function of a certain attribute is before attempting to change its value.*

The Service Port

The use of the Service Port is recommended only in special cases during an installation operation where there is no Internet access, or the RTU does not manage to contact the M2M Gateway.

Note: *Improper commands issued through the Service Port may compromise the functionality of the RTU.*

To communicate with the RTU through the Service Port, a 9-pin modem serial cable and a PC with a terminal program (e.g. Hyperterminal, minicom, etc.) is required. The communication parameters are: 19200 Baud, 1 stop bit, no parity. The command line terminator is the line feed (hex 0A, or ctrl-J).

The command line interface (CLI) of the RTU can be also reached through GSM, if the network and the SIM card in the RTU support CSD calls. To reach the CLI in this way a modem must be used. Simply dial the RTU's phone number (specified by the attribute *gsmDataNumber*) to connect to the RTU.

Note: *Not all CLI commands are available through GSM calls, for more details see the "Commands" section.*

Some Useful Commands

Most of the time the RTU operates in sleep mode to conserve power; to wake it up, press a key (e.g. <enter>). Possibly you will be asked for a password—it is the same password that was defined on the M2M Gateway when the RTU was entered in the system. After entering the correct password a ":" (colon) prompt will be displayed on the terminal.

Note: *if you are not prompted for a password, that may be because the RTU has not been yet initialized, and therefore it is not configured. However, you can still operate the RTU over the CLI.*

A short list of useful commands during the installation is given below:

help
Displays the list of all accepted commands; for further information about a specific command, type **<cmd> -h**.

attr
Lists all RTU attributes and their values. By examining the attributes you can debug communication or sensor problems. In addition, you can alter certain attributes using the same command.

log
Lists the internal log messages. By examining the log messages you can debug various problems. To navigate in time through the log messages, use a date/time parameter, e.g. **log 12/03/2010 0:0:0**.

net modem
Enter a direct connection to the built-in GSM modem. At this point you can issue standard AT commands. For instance, the command AT+CSQ returns the current RSSI level and the Bit Error Rate (BER), while the command AT+CREG? returns the state of the GSM module, i.e. if it is registered on the GSM network or not (0,1 means registered, 0,0 is not registered). To leave the "modem" command, enter ctrl-X. The **net** command group is not available over GSM.

reboot
Performs a soft reboot of the RTU. Used with the **-cc** option, this command clears the configuration of the RTU.

For a complete list of commands and their description see the "Commands" section.

Data Acquisition Subsystem

Depending on the interface to the input/output devices (sensors), there are currently three different RTU versions:

- T707-SDI which supports sensors based on the SDI-12 protocol. T707 supports the SDI-12 protocol version 1.3. It has various operating modes and can be configured to accept a large variety of sensors. An internal jumper (see also the "Installation" section) allows switching between native SDI-12 hardware interface and RS-485 balanced interface; the later allows for longer cables, but unfortunately not many SDI-12 sensor manufacturers implement this mode.

- T707-DAV which supports the Davis Instruments, Inc. Vantage Pro weather stations; the WeatherLink interface (also from Davis Instruments) must be used for interfacing to the T707 RTU. The communication is done through an RS-232 link.

- T707-THI which supports a series of data-loggers manufactured by the company Adolf Thies GmbH & Co. KG. (e.g. model TDL14). Both the RS-232 and RS-422 interfaces are supported. The interface mode can be switched by means of the attribute *rs422*.

Specific functions of the Data Acquisition subsystem are configured by means of attributes. The subsystem is hierarchically structured, and it can have a number of sensors, each of them in turn having a number of tags. A tag in this sense represents a specific value that can be read from—e.g. a temperature, or written to—e.g. a coil or a valve. Thus there can be input or output tags.

Note: *Currently, only the SDI-12 RTU version supports output tags (by means of SDI-12 "X" commands); the Davis and Thies versions support only input tags.*

Although there are some common attributes, depending on each version specific attributes are used to define the functionality of the sensors and tags. For instance, *acquisitionMode* and *acquisitionSchedule* are common attributes for all kind of sensors; however, *sdiMethod* is specific to the SDI-12 version, while *archiveInterval* is specific to the Davis version. For more details on these attributes, see the "Attributes" section.

The Data Acquisition Subsystem includes a First-In, First-Out (FIFO) ring storage. The data provided by the sensors is stored in the FIFO until delivered by the Communication Subsystem via GPRS. The physical memory is backed-up by a small internal Lithium battery, thus keeping it alive even if the power is removed for extended periods of time.

Commands

The information in this chapter is intended for well trained technicians to pinpoint errors or other device malfunctions, as well as for developers that wish to implement special software in conjunction with the T707 RTU. Before using any command described in this chapter, make sure you have a good understanding on what the command does. Improper use may render an RTU inoperable.

The T707 RTU accepts three groups of commands:

- general: addresses internals of the RTU, or allow attribute manipulation;
- data acquisition (DACQ): operates on the data acquisition subsystem;
- communication (NET): view/modify network and GPRS communication parameters.

The Command Line Interface (CLI) can be reached either over the Service Port (the RS-232 connector) or over GSM by calling the RTU with a modem.

Note: Not all commands can be executed over GSM.

General Commands

The general commands are issued without a prefix, e.g.:

```
: ver
T707-SDI Remote Terminal Unit, ver 1.54 build 1096,
(c) 2004-2009 Metrilog Systems SRL
```

ver

Description   Returns the firmware version.

Example

```
: ver
T707-SDI Remote Terminal Unit, ver 1.54 build 1096,
(c) 2004-2009 Metrilog Systems SRL
```

attr

Description   Lists and sets various attributes. Issued without a parameter, only the RTU attributes and their value will be returned.

Usage   attr [-a | -s | -t | -h | -v]
attr [nodeid] attrib_name [attrib_value]

Parameters   **-a** – all attributes will be returned (RTU, sensors and tags)
-s – only sensors' attributes will be returned
-t – sensors' and tag's attributes will be returned

-v – all parameters exchange will be done using a CCITT-computed CRC; this option is used in conjunction with automated test equipment and software that communicates with the RTU during production or maintenance

nodeid – a specific node on the RTU; for the attributes on the RTU itself this parameter must be skipped

attrib_name – the name of a specific attribute

attrib_value – the value to be set to the respective attribute; if this parameter is not entered, the command will return the current value of the attribute

Remarks   If an attribute is changed on the RTU using the **attr** command, the change will be replicated to the M2M Gateway during the subsequent connection, when the time arrives as specified by the *connectSchedule* attribute. However, if for some reasons the first connection after the change does not succeed, the replication to the M2M Gateway will not be retried, thus leading to a possible de-synchronization. To correct such a situation, issue a "Sync Config" command on the Gateway (you must have "admin" rights to do this).

Examples

```
: attr
            connectMode: 2
        connectSchedule: */10
          connectSpread: 30
         timeZoneOffset: -3600
            smsSCANumber:
        smsNotifyNumber:
        smsAcceptNumber:
     smsAcceptNumberAlt:
              csdAccept: true
         csdAcceptNumber:
      csdAcceptNumberAlt:
              gsmNumber: +4369911191323
          gsmDataNumber: +4369951191323
                  tsURL: http://m2m.metrilog.net/rtu
               tsURLAlt:
                gprsAPN: wap.one.at
         gprsLoginName: wap
        gprsLoginPasswd: wap
           wapGatewayIP: 194.24.128.118
         wapGatewayPort: 9201
             gprsAPNAlt: web.one.at
       gprsLoginNameAlt: web
     gprsLoginPasswdAlt: web
        wapGatewayIPAlt: 81.223.143.133
      wapGatewayPortAlt: 9201
          gprsAPNUseAlt: true
     wapGatewayFallback: 2 8
                useHTTP: 2 1
              backupDNS: 195.34.133.133
```

```
              gsmAllowRoaming: false
                   gsmCarrier: 23205
          gsmCarrierSelection: auto
                     cliLogin: false
                     logLevel: 0
              protoDebugLevel: 0
                     sdiAdmin:
                  tsLoginName: Test_Metr
              tsLoginPassword: 2222
                 manufacturer: Metrilog Systems SRL
                         hwID: 1002
                       hwType: T707-SDI
                    swVersion: 1.54
                      swBuild: 1096 @ Mar 17 2010 17:17:31
             responseMaxBytes: 32000
              sensorArraySize: 50
               comboArraySize: 30
                      gsmIMEI: 353781013125024
                      gsmCCID: 983450406090026239
                   gsmVersion: 657d09gg.Q24PL001
                  currentDate: 26/8/2010 12:20:40 GMT+02.00
             nextConnnectDate: 26/8/2010 12:30:07 GMT+02.00
                 putdataDate: 6/8/2010 14:39:59 GMT+02.00
                 nextSyncDate: 27/8/2010 2:00:00 GMT+02.00
                         date: 26/8/2010 12:20:18 GMT+02.00
                       uptime: 1d, 0h, 53m
                  gsmCellInfo:
                    gsmMccMnc: 23205
                     gsmLacCi: 186,13007
                   gsmRssiBer: 20,0

  : attr -s
              *** sensor id: 33733
              acquisitionMode: 2
          acquisitionSchedule: */10
                   sdiAddress: W
                    sdiMethod: CC
                sdiPostmethod:
                      sdiInfo: Sensor unreachable
                     lastDate: 26/8/2010 12:20:00 GMT+02.00
                     nextDate: 26/8/2010 12:30:00 GMT+02.00
                  putdataDate: None

              *** sensor id: 11127
              acquisitionMode: 2
          acquisitionSchedule: */10
                   sdiAddress: A
                    sdiMethod: M
                sdiPostmethod:
                      sdiInfo: 13MetrilogM512rC1.4.1055#000000
```

```
                    lastDate: 26/8/2010 12:20:00 GMT+02.00
                    nextDate: 26/8/2010 12:30:00 GMT+02.00
                putdataDate: 26/8/2010 12:10:00 GMT+02.00

: attr 33733 acquisitionMode
2

: attr 33733 acquisitionMode 0

: attr 33733 acquisitionMode
0

: attr rs422 true

: attr rs422
true
```

hist

Description   Shows historians configuration. An RTU can manage a maximum of 50 historians. A list will be presented showing the local ID (on the RTU), the remote ID (on the Gateway), an index that associates a historian to a node, the oldest and the newest data timestamps for each historian. Note that the historians are dynamically allocated and when they are discarded on the Gateway, their data is not automatically deleted in the RTU, rather it will be overwritten. In the example below, only historians 19 to 24 are current, all others are older historians that are no more in use.

Example

```
: hist

Hist.   Node   Index      First                Last
  0    12237    -1    22/07/2010 20:10:00   26/07/2010 11:00:00
  1    12236    -1    22/07/2010 20:10:00   26/07/2010 11:00:00
  2    12239    -1    22/07/2010 20:10:00   26/07/2010 11:00:00
  3    12238    -1    22/07/2010 20:10:00   26/07/2010 11:00:00
  4    12837    -1    22/07/2010 20:10:00   26/07/2010 11:00:00
  5    11564    -1               None                  None
  6    11565    -1               None                  None
  7    11566    -1               None                  None
  8    11567    -1               None                  None
  9    11568    -1               None                  None
 10    11569    -1               None                  None
 11    11561    -1               None                  None
 12    11562    -1               None                  None
 13    11563    -1               None                  None
 14     2230    -1               None                  None
 15     2229    -1               None                  None
 16     2228    -1               None                  None
 17     2227    -1               None                  None
 18     2226    -1               None                  None
 19    11128     7    22/07/2010 20:10:00   26/08/2010 12:30:00
 20    11129     8    22/07/2010 20:10:00   26/08/2010 12:30:00
```

```
21      11130       9    22/07/2010 20:10:00    26/08/2010 12:30:00
22      11131      10    22/07/2010 20:10:00    26/08/2010 12:30:00
23      11132      11    22/07/2010 20:10:00    26/08/2010 12:30:00
24      11134      12    22/07/2010 20:10:00    26/08/2010 12:30:00
25      33658      -1                   None                   None
26      33657      -1                   None                   None
27      33738       0    22/07/2010 20:10:00    06/08/2010 14:39:59
28      33740       1    22/07/2010 20:10:00    06/08/2010 14:39:59
29      33739       2    22/07/2010 20:10:00    06/08/2010 14:39:59
30      33734       3    22/07/2010 20:10:00    06/08/2010 14:39:59
31      33741       4    22/07/2010 20:10:00    06/08/2010 14:39:59
32      33736       5    22/07/2010 20:10:00    06/08/2010 14:39:59
33      33735       6    22/07/2010 20:10:00    06/08/2010 14:39:59
```

timers

Description View next connect and acquisition times. "Connection" refers to the next time the RTU will connect to the M2M Gateway while "Acquisition" refers to the next time a sensor will be sampled.

Example

```
: timers
Next connect at:     26/8/2010 12:50:29 GMT+02.00
Next acquisition at: 26/8/2010 12:50:00 GMT+02.00
```

pin

Description Manipulates the PIN/PUK for the SIM card.

Usage pin [-v] [PIN[PUK]]
pin clear

Parameters **-v** – all parameters exchange will be done using a CCITT-computed CRC; this option is used in conjunction with automated test equipment and software that communicates with the RTU during production or maintenance
PIN – the PIN (a 4-digit number)
PUK – the PUK (usually an 8-digit number)
clear – clears the PIN and the PUK

Remarks The **pin** command does not operate on the SIM card itself; it only sets internal variables for PIN and PUK so that the RTU firmware knows how to activate the SIM card at start-up or during operation if the GSM module is reset. The PUK is used only in extreme situations, e.g. if the SIM card becomes PIN-locked. If a card with the PIN code disabled is used, then PIN and PUK codes are not required.

Examples

```
: pin 0123 45678901
```

```
: pin
PIN is set
PUK is set

: pin clear

: pin
PIN is not set
PUK is not set
```

hwid

Description   Sets/returns the serial ID number of the RTU.

Remarks   The serial ID number (also known as "Hardware ID") is unique and can be
set only once during the production.

Example

```
: hwid
hwID is 2234
```

nl

Description   This command allows setting the end of line character (CR or LF). Issued
without a parameter, it returns the current character expected as end of
line. Default is CR (0xD).

Examples

```
: nl
End of line is cr

: nl lf

: nl
End of line is lf
```

echo

Description   Enables/disables echo of the typed characters. Issued without a parameter,
it returns the current state. Default for echo is ON.

Examples

```
: echo
Echo is on

: echo off
```

fifo

Description   Returns the status of the internal FIFO (First In, First Out) memory. The
command can be also used to clear or read data out of the FIFO.

Usage   fifo {stat | clear | retr}

Parameters	**stat** – inquiry the status of the FIFO **clear** – clear the FIFO **retr** – retrieve a data slot out of the FIFO
Remarks	By issuing the *fifo clear* command, all data will be lost! This command should be used before an RTU removed from a site will be installed on a different site. For related information, see also the *reboot* command. The *fifo stat* command returns useful information in case of data storage problems.

Examples

```
: fifo stat
Data begins at 786432, ends at 1048576, data size 262144
Store pointer at 1045164, retrieve pointer at 1045128
Current FIFO pointer ff2ac, current block 63, oldest block 0
FIFO recent data on: 26/8/2010 14:50:00 GMT+02.00
FIFO oldest data on: 22/7/2010 20:10:00 GMT+02.00

: fifo retr 20/8/2010 0:0:0
Slot at 20/8/2010 0:10:00 GMT+02.00
  Hist 19, stat  OK, value      0.00
  Hist 20, stat  OK, value     42.69
  Hist 21, stat  OK, value     26.41
  Hist 22, stat  OK, value      0.00
  Hist 23, stat  OK, value      1.00
  Hist 24, stat  OK, value     11.51

: fifo clear
Are you sure? All FIFO data will be lost (y/n): y
```

log

Description	Displays log data. The command can be also used to clear the log memory or to inspect the status of the log memory. The log data is kept in a FIFO memory; the oldest data will be automatically deleted. Each log entry contains the date/time of the event, the event's code and the event's message.
Usage	log [date] log {clear | stat}
Parameters	**date** – date/time in standard format (see examples) from where the log data should be displayed; if this parameter is absent, data of the last log command issued will be used (in most cases, this is the oldest log data in the FIFO) **clear** – this parameter is used to clear the log buffer **stat** – the command will return the status of the log FIFO
Remarks	Note that no confirmation is requested when issuing the *log clear* command: the log data will be immediately erased. The *log stat* command returns useful information in case of log data storage problems. For a list explaining the log messages please see the "Error, Warning and Informational Messages" section.

Examples

```
: log
18/08/2010 13:46:07 - 03: INIT: system boot after hardware reset
18/08/2010 13:46:29 - 40: GSM: PIN not set
18/08/2010 13:46:31 - 05: INIT: system started
18/08/2010 13:52:25 - 30: COMM: timeout, exit
18/08/2010 22:10:50 - 21: RPC: transport error -13, state 11b (exit)
19/08/2010 02:20:51 - 21: RPC: transport error -46 (http 502), state 7  (exit)
19/08/2010 02:30:48 - 21: RPC: transport error -46 (http 502), state 7  (exit)
19/08/2010 02:40:32 - 21: RPC: transport error -46 (http 502), state 7  (exit)
19/08/2010 02:51:00 - 21: RPC: transport error -46 (http 502), state 7  (exit)
19/08/2010 03:00:38 - 21: RPC: transport error -46 (http 502), state 7  (exit)
19/08/2010 03:11:01 - 21: RPC: transport error -46 (http 502), state 7  (exit)
19/08/2010 03:31:04 - 21: RPC: transport error -46 (http 502), state 7  (exit)
19/08/2010 03:41:01 - 21: RPC: transport error -46 (http 502), state 7  (exit)
19/08/2010 13:10:48 - 33: COMM: TCP/IP failed
19/08/2010 17:00:52 - 33: COMM: TCP/IP failed
20/08/2010 19:21:44 - 21: RPC: transport error -13, state 11a (exit)
21/08/2010 16:01:02 - 33: COMM: TCP/IP failed
21/08/2010 16:10:50 - 33: COMM: TCP/IP failed
21/08/2010 19:51:29 - 21: RPC: transport error -13, state 7  (exit)
22/08/2010 08:30:44 - 37: COMM: connection to WAP GW failed

: log 25/8/10 0:0:0
25/08/2010 19:41:23 - 21: RPC: transport error -13, state 11c (exit)
25/08/2010 23:01:31 - 33: COMM: TCP/IP failed
25/08/2010 23:51:27 - 21: RPC: transport error -13, state 7  (exit)

: log stat
sor 608458, current 608432, csave 608432, cli 608432, eor 608432
```

exit

Description	Exits the CLI
Remarks	This command will exit the current level of the CLI; if already on the lowest level, then the device will enter sleep mode. To re-activate the CLI, simply enter a key.

Example

```
: exit
Exiting...
```

gsmres

Description	This command resets the built-in GSM Modem.
Remarks	If the command is issued through a modem call, the connection will break up.

Example

```
: gsmres
```

reboot

Description This command reboots the RTU. Using the appropriate parameters, the command may be also used to erase the internal configuration of the RTU.

Usage reboot [{-h | -c | -cc}]

Parameters **-c** – after reboot, erase the volatile configuration data
-cc – after reboot, erase both the volatile and persistent configuration data

Remarks If no parameter is supplied, the RTU will simply reboot. If the **-c** parameter is used, only the volatile configuration, i.e. the data retrieved from the M2M Gateway will be erased. The initial configuration data stored in flash will be kept and will be used to connect to the Gateway an retrieve the volatile configuration data. If the **-cc** parameter is supplied, then the whole configuration, including the data in the flash memory will be erased. The RTU reboots to factory default settings. Together with the **fifo clear** and **log clear** commands, the **reboot -cc** command must be issued to an RTU removed from a site that will be installed on a different site.

Examples

```
: reboot
Are you sure you want this? (N/y): y
Shut down time arrived!
26/08/2010 17:11:40 - INIT: system boot after software reset
26/08/2010 17:12:04 - INIT: system started

: reboot -cc
Persistent and volatile configurations will be cleared at reboot!
Are you sure you want this? (N/y):
Shut down time arrived!
26/08/2010 17:12:54 - INIT: system boot after software reset
26/08/2010 17:12:59 - INIT: invalid config, attempting flash restore
26/08/2010 17:12:59 - INIT: flash restore failed, configuration cleared
26/08/2010 16:13:11 - INIT: Operator data loaded
26/08/2010 16:13:11 - INIT: system started
```

help

Description Lists all available commands with a short explanation.

csd

Description Switches operation from GPRS to CSD.

Usage csd mode phone_number user password

Parameters **mode** – true/false; if true, connections to the Internet will be made through CSD
phone_number – the phone number of the PPP server to call
user – user name for the PPP account
password – password for the PPP account

Remarks　This mode of operation is not recommended, and should be used only if the GPRS service is unavailable for extended periods of time. The costs of traffic are reported by the minute and not by data units (e.g. KByte). In addition, a GSM operator must provide the necessary information for operating in this mode (phone number, user and password of the point of presence). Note that after a reboot, the RTU sets the CSD operation automatically to false.

Example

```
: csd true +43123456789 ppp ppp

: csd
true, +43123456789, ppp, ppp
```

Data Acquisition Commands

Data acquisition commands can be issued either by using the **dacq** prefix, or by entering the CLI level *dacq*, e.g.:

```
: dacq
DACQ:
```

In the above example, the prompt "DACQ:" will indicate that the CLI is now at the data acquisition command level. Obviously, only the *dacq* related commands can be issued at this level.

dacq t

Description　Issues a command in transparent mode to a sensor.

Usage　dacq t *<command>*

Remarks　For SDI-12 version: *<command>* is a standard SDI-12 command, including the exclamation point ("!"); for Davis version: *<command>* is a Davis command as described in the "Vantage Pro Serial Communication Reference Manual" (see Davis Instruments web site http://www.davisnet.com); for Thies version: *<command>* is a Thies TDL14 command as described in the Thies TDL14 User Guide.

Examples

```
: dacq t AI!
A13MetrilogM512rC1.4.1055#000000

DACQ: t AM!
A0048
DACQ: t AD0!
A+25.17+31.71+0+0+1+0+1+12.03
```

dacq info

Description　Get the INFO/VER string from all the attached sensors.

Usage	dacq info
Remarks	The result(s) of the command are not displayed on the console, rather all sensors are requested for the info data and the respective attributes are updated (*sdiInfo*, *thiesInfo* or *davisInfo*). Use the *attr -s* command to display these attributes.
Example	

```
: dacq info
```

dacq auto

Description	Enables/disables the automatic sampling of the sensors
Usage	dacq auto {on|off|stat}
Parameters	**on** – enable automatic sampling **off** – disable automatic sampling **stat** – return status of the automatic sampling
Examples	

```
: dacq auto stat
Automatic data acquisition enabled
: dacq auto off
: dacq auto stat
Automatic data acquisition disabled
```

dacq dmp

Description	Returns data from the data-logger
Usage	dacq dmp <*date*>
Parameters	**date** – date in the dd/mm/yy hh:mm:ss format
Remarks	This command applies only to the Davis and Thies versions.

dacq settime

Description	Sets the time on the data-logger
Usage	dacq settime
Remarks	This command applies only to the Davis and Thies versions. The system date/time will be sent to the data-logger, thus the command has no parameters.

dacq direct

Description	Transparently connects the CLI to the data-logger serial port.
Usage	dacq direct
Remarks	This command applies only to the Thies version. To exit this command use <ctrl X>. When sending commands to the Thies data-logger in this mode,

you must use the <ctrl B> and <ctrl C> characters to begin and end a command (see the Thies documentation).

dacq boot

Description	Re-flashes a SDI-12 sensor (remote firmware update).
Usage	dacq boot sdi_adr [url]
Parameters	**sdi_adr** – address of the SDI-12 sensor to be updated **url** – the URL of the firmware file to be downloaded
Remarks	This command applies only to the SDI-12 version; only SDI-12 sensors that support this functionality can be remote upgraded. The URL is optional if it was already invoked with a previous similar command. Generally this form of updating the sensors' firmware is not preferred, it is recommended to use the M2M Gateway remote firmware feature.

Examples

```
dacq boot 0 http://fw.metrilog.at/dw/m514.bin
```

dacq exit

Description	Exits the dacq level.
Usage	dacq exit

Examples

```
DACQ: exit
Exiting...
:
```

dacq help

Description	Lists all available dacq commands with a short explanation.

Network Commands

Network commands can be issued either by using the **net** prefix, or by entering the CLI level *net*, e.g.:

```
: net
NET:
```

In the above example, the prompt "NET:" will indicate that the CLI is now at the network commands level. Only *net* related commands can be issued at this level.

Note: *Most network commands cannot be issued over a remote modem connection, as the modem is busy and cannot open a TCP/IP connection.*

net modem

Description	Opens a direct connection to the GSM modem.

Usage	net modem [baud-rate]
Parameters	**baud-rate** – modem baud rate (optional); by default, the modem is set to 115200 bps
Remarks	Used to issue AT commands to the GSM modem when debugging SIM card problems or general GSM registration issues. The command can be used only over the service port. To exit the command, use <ctrl-X> or <ctrl-C>.
Example	

```
: net modem
at
OK
```

net stat

Description	Display the status of the communication task, i.e. if an IP connection is pending, active, or down.
Usage	net stat
Examples	

```
: net stat
Comm task is down
TCP/IP task is down

: net stat
Comm task is running
TCP/IP task is running
#  IP addr.       Mask            Up  Type  MTU  Flags  Peer/router
 0 10.101.149.48  255.255.255.255 yes ppp   1500 *      192.168.111.111
```

net cmd

Description	Performs various network related commands.
Usage	net cmd {gwup	gwdown	login	logout	getconfig	gettasks	getlogin	getdate	putdata	upgrade	down}
Parameters	**gwup** – connect to a WAP gateway; if networking is not up, it will be first started **gwdown** – disconnect from a WAP gateway; if the RTU is already logged into the M2M Gateway, a logout will be first issued **login** – log into the M2M Gateway; if the network is not up, or the connection to the WAP gateway is not done, the respective actions will be first initiated and only if successful a connection to the M2M Gateway will be attempted **logout** – log out of the M2M Gateway; connections to the WAP gateway (if any) and to the network will remain active **getconfig** – a getconfig command will be initiated; the current configuration of the RTU will be replaced (i.e. synchronized) with the one on the M2M Gateway **gettasks** – retrieve pending tasks (if any) on the M2M Gateway

getlogin – force a getlogin command to the M2M Gateway; by using the credentials (serial ID, SIM Card CCID or module IMEI), the RTU will retrieve its login data (name and password)

getdate – force a synchronization of the internal clock

putdata – force a data push to the M2M Gateway; any pending data will be pushed to the M2M Gateway

upgrade <URL> – force an upgrade; an URL must be specified, see examples

down – shut down the network connection; if the connection to the M2M Gateway or to the WAP gateway are active, they will be first closed

Remarks
The **net** command can be used only over the service port. These commands are not intended for normal operation, rather for debugging purposes. After starting-up the network, a 60 seconds timeout applies, i.e. the network will be brought down after 60 seconds of inactivity.

Examples

```
: net cmd login
```

```
: net cmd getdate
```

```
: net cmd upgrade http://server.provider.com/images/t707sdi_1_55.bin
```

```
: net cmd down
```

net exit

Description
Exits the net level.

Usage
net exit

Examples

```
NET: exit
Exiting...
:
```

net help

Description
Lists all available net commands with a short explanation.

SMS Commands

The T707 RTU is able to execute a small set of commands received through SMS. This is useful in cases a CSD connection is not possible. The commands are formatted as XML strings that are parsed by the RTU; note however that they are limited in functionality. In addition, the SMS commands are executed only if either the *smsAcceptNumber* or the *smsAcceptNumberAlt* attributes are set to the MSISDN of the SMS sender (see also the "Attributes" section). SMSes from other senders will not be executed.

Note: *If neither of the two attributes is set, the RTU will execute all the SMSes received, no matter the MSISDN of the originator. This is however an important security hole.*

Connection Parameters SMS

Description
Sets the initial operator parameters for an RTU to be able to connect through GPRS.

Format
<connection [apn="apn"] [name="name"] [passwd="password"]
        [gwip="ip_address"] [gwport="port"] [roaming="true | false"]
        [useHTTP="mode delay"]/>

Parameters
apn – APN (Access Point Name), to be obtained from the GSM operator
name – user name, to be obtained from the GSM operator
passwd – password, to be obtained from the GSM operator
gwip – WAP gateway IP address, to be obtained from the GSM operator
gwport – Wap gateway port, to be obtained from the GSM operator
roaming – set to true if roaming is accepted, and false otherwise
useHTTP – set the *useHTTP* attribute (see also the RTU attributes section)

Remarks
As with any XML document, all or none of the elements may be included in an SMS; several successive SMSes can also be used. Note however that after receiving such an SMS, the RTU will immediately try to connect to the M2M Gateway, if the "server" parameter has been also set (see also the next SMS command). This command is useful to start-up an RTU with a SIM card from a provider not yet in the internal list of the RTU.

Example

```
<connection apn="wap.connex.ro" name="wap" passwd="connex"
gwip="193.226.1.2" gwport="9201" roaming="true" useHTTP="1 8"/>
```

Server Parameters SMS

Description
Sets the M2M Gateway parameters.

Format
<server tsurl="M2MGW_URL" />

Parameters
tsurl – URL of the M2M Gateway the RTU will connect to

Remarks
This is normally the URL of Metrilog's M2M Gateway. Note that if the XML string does not reach the maximum of 160 characters per SMS, both the Connection and the Server parameters can be sent in a single message, e.g. <connection ... /> <server ... />.

Example

```
<server tsurl="https://www.metrilog.net/rtu"/>
```

Wake-up SMS

Description
Wakes-up the RTU to perform asynchronous operations.

Format
<wakeup attn="operation" />

Parameters
attn – specifies the operation to be executed. Following operations are currently defined:
        – **login**, to execute an out of schedule login to the server, e.g. to check for possible tasks on server

– **getlogin**, to force a getlogin (retrieve new name/password based on credentials)

– **reboot**, to force a reboot of the RTU; this operation accepts the optional parameters **-c** and **-cc** (see the "General Commands" section for additional information).

Examples

```
<wakeup attn="login"/>
```

```
<wakeup attn="reboot -cc"/>
```

Set Debug Level SMS

Description Sets the system logging level on the RTU and/or on the M2M Gateway.

Format <debug [loglevel="2"] [protodebuglevel="0"] />

Parameters **loglevel** – sets the level of the logging system on the RTU
protodebuglevel – sets the level of the logging for this particular RTU on the M2M Gateway

Remarks For additional details please consult the "RTU Attributes" section, in particular the *logLevel* and *protoDebugLevel* attributes.

Example

```
<debug protodebuglevel="0" loglevel="2"/>
```

Modem Commands SMS

Description Sends commands directly to the GSM modem.

Format <modem cmd="modem_command" />

Parameters **cmd** – a string representing a valid AT command

Remarks The AT command will be passed to the modem and the result will be piped to the logging subsystem. Only one AT command per SMS can be issued; however, certain commands can be included in a single string starting with AT and separated by semicolons (;). Use utmost care when issuing such commands, as the modem can be rendered completely useless if misconfigured!

Example

```
<modem cmd="ATZ"/>
```

More About Logging

By appropriately setting the *logLevel* attribute you can control the logging level of the T707 RTU. As described in the "Attributes" section, following levels are available:

• 0 — only errors messages, as well as some initialization messages at boot time are displayed;

• 1 — error messages (all of the above) plus informational messages are displayed;

- 2 — debug level, all messages are displayed;
- 255 — no message will be displayed.

At debug level (2) a large amount of messages will be displayed, i.e. stored in the internal logging facility. Thus, it is not recommended for normal use because it may substantially increase the amount of data traffic over GPRS.

However, there are other ways to selectively increase the logging level for displaying only messages from certain subsystems of the RTU. By setting the *logLevel* attribute to a message number, then only these messages will be displayed/stored (for all valid message numbers, see the "Error, Warning and Informational Messages" section of this manual). As an example, if the logLevel is set to 30, then all states of the Communication State Machine task will be displayed.

Sometimes you may need to filter all messages related to a specific task; this can be done by adding 100 to the number of the first message in a task class. As en example, if you need only messages for the Data Acquisition task, then set the *logLevel* attribute to 180 (first message number of the Data Acquisition task is 80, add 100 so the result is 180).

Attributes

This section explains the attributes used to define a T707 RTU. By changing the value of an attribute, the functionality of the RTU can be altered. You must first understand the function of a certain attribute before attempting to change its value, as improper manipulation of the attributes can lead to malfunction of the RTU. You can alter the value of an attribute either over the serial port using the command *attr*, or through the M2M Gateway User Interface.

There are three levels of attributes: RTU, sensor and tag. All known attributes to date are described on the following pages. Future firmware versions may add or deprecate attributes.

Note: *The 'type' in the table below refers to operations on attributes performed over the serial Service Port of the RTU.*

Note: *Certain attributes require multiple parameters; multi-parameter attribute can be issued on the serial Service Port by quoting it, as e.g.* **attr useHTTP "2 1"**.

RTU Attributes

Attribute	Type	Description
connectMode	r/w	Defines how the RTU connects to the M2M Gateway. Following modes are defined: 0 – no connection, stopped 1 – manual connection (must be initiated e.g. over the service port) 2 – automatic mode, defined by the *connectSchedule* attribute
connectSchedule	r/w	Defines the connection schedule of the RTU. It uses a cron syntax, ex: */10 means that the RTU should connect every ten minutes, starting with minute 0. A complete cron string may specify minutes, hours, days, months and the day of the week. The smallest unit that can be set is therefore one minute. However, for practical reasons related to GPRS communication specifics, the minimum recommended interval is five minutes. For additional details on configuring cron, consult the unix man pages for *crontab (5)*.

Attribute	Type	Description
connectSpread	r/w	The value of this attribute is specified in seconds. A random value in the interval 0 ... *connectSpread* will be added to the computed *connectSchedule.* If 0, connection spreading will not be used.
timeZoneOffset	r/w	The value of this attribute is specified in seconds and represents the offset to the UTC time zone (which has the value 0). Positive values represent westwards time zones while negative values eastwards time zones.
smsSCANumber	r/w	SMS Service Center Address: required by certain providers in order to send SMSes (currently not used).
smsNotifyNumber	r/w	A phone number to use when sending notification SMSes (currently not used).
smsAcceptNumber	r/w	Number of the SMS sender from which the RTU will accept SMSes. If the attribute is empty, all SMSes will be accepted. If the attribute is set, only SMSes from this number will be accepted.
smsAcceptNumberAlt	r/w	Alternate number from where SMSes will be accepted, similar to *smsAcceptNumber*.
csdAccept	r/w	Boolean attribute: if true, CSD calls will be accepted.
csdAcceptNumber	r/w	The telephone number from where the RTU will accept CSD calls (through modem). If the caller ID doesn't match this number, the call will be rejected. If the attribute is empty, all calls will be accepted.
csdAcceptNumberAlt	r/w	Alternate number from where CSD calls will be accepted, similar to *csdAcceptNumber*.
gsmNumber	r/w	Own GSM number (MSISDN).
gsmDataNumber	r/w	Own GSM data number (Data MSISDN).
tsURL	r/w	M2M Gateway URL. Default is Metrilog's M2M Gateway http://m2m.metrilog.net/rtu
tsURLAlt	r/w	Alternate M2M Gateway URL (currently not used).
gprsAPN	r/w	GPRS Access Point Name. The value of this attribute is specific to each GSM operator.
gprsLoginName	r/w	Login name to use with the *gprsAPN*.
gprsLoginPasswd	r/w	Password to use with the *gprsAPN*. Both login name and login password are provided by the GSM operator.
wapGatewayIP	r/w	IP address of the WAP Gateway (provided by the GSM operator).
wapGatewayPort	r/w	Port of the WAP Gateway (provided by the GSM operator).

Attribute	Type	Description
gprsAPNAlt	r/w	Alternate GPRS Access Point Name. The alternate APN is generally used for connections over Internet, while the standard APN for connections over WAP. Some GSM Providers offer a single APN for both types of communication. The use of the alternate APN depends on the value of the *gprsAPNUseAlt* attribute.
gprsLoginNameAlt	r/w	Login name to use with the *gprsAPNAlt*.
gprsLoginPasswdAlt	r/w	Password to use with the *gprsAPNAlt*. Both login name and login password are provided by the GSM operator.
wapGatewayIPAlt	r/w	IP address of the alternate WAP Gateway (provided by the GSM operator).
wapGatewayPortAlt	r/w	IP port of the alternate WAP Gateway (provided by the GSM operator).
gprsAPNUseAlt	r/w	If true, use the alternate APN for Internet data transfers; the same applies if the alternate WAP Gateway must be used (see also the *wapGatewayFallback* attribute below).
wapGatewayFallback	r/w	Defines the operation through the alternate WAP Gateway. The attribute consists of a string of two numeric parameters: the first represents the number of unsuccessful retries before switching to the alternate WAP Gateway; the second represents the number of hours elapsed before switching back to the standard WAP Gateway.
useHTTP	r/w	Defines the operation via HTTP. The attribute consists of a string of two numeric parameters: the first defines the communication mode while the second defines the number of hours elapsed before switching to HTTP. Following communication modes are defined: — 0: WAP; — 1: HTTP; — 2: automatic, i.e. by default WAP, but after the number of hours (defined by the second parameter) of WAP communication failure, switch to HTTP. In automatic mode, after the first successful connection over HTTP, the system reverts to WAP.
backupDNS	r/w	IP address of a backup DNS server used in case those offered by the GSM operator are not operational.
gsmAllowRoaming	r/w	If true, roaming will be allowed. If the attribute is set to false, the RTU can still register to a roaming network but no GPRS connection will be initiated.

Attribute	Type	Description
gsmCarrier	r/w	The ID of the GSM operator's network the RTU should operate on.
gsmCarrierSelection	r/w	Used in conjunction with *gsmCarrier* to define the operating mode as follows: — auto: the GSM module will choose the network to register on; — man: the module will be forced on the network specified by the *gsmCarrier* attribute. Note that should the module be forced to operate in roaming, the *gsmAllowRoaming* attribute must be set true.
cliLogin	r/w	If true, authentication will be required to access the serial Service Port.
logLevel	r/w	Log level reporting: 0—error; 1—info; 2—debug. A value of 255 will totally inhibit reporting (no messages will be issued). For other specific reporting levels, see the "Error, Warning and Informational Messages" section.
protoDebugLevel	r/w	A debug level added to the HTTP headers (X-Debug) during data interchange with the server (the M2M Gateway); by default it is 0 and should be increased only when debugging the protocol implementation. Values higher than 0 determine additional logging on the M2M Gateway for this specific RTU.
rs422	r/w	If true, the Thies interface operates in RS-422 mode; if false, the Thies interface operates in RS-232 mode. This attribute is specific to the T707-THI variant.
sdiAdmin	r/w	Send SDI command(s) in pseudo-transparent mode to an SDI sensor. Issuing such an attribute on the Service Port does not make sense, as one can use the *dacq t* command interactively. The attribute is intended to be updated through the M2M Gateway. More than one command can be sent in one string; while spaces between commands are optional, they improve readability. If a delay is required between commands, it can be specified as "\n", where "n" specifies the delay in seconds. Example: "0M! \5 0D0!". The results returned by the sensor(s) are piped into the log subsystem. This attribute is specific to the T707-SDI variant.
tsLoginName	r	Login name to use when accessing the M2M Gateway.
tsLoginPassword	r	Password to use when accessing the M2M Gateway.
manufacturer	r	Manufacturer's name.

Attribute	Type	Description
hwID	r	Serial ID of the RTU.
hwType	r	RTU type.
swVersion	r	Software version.
swBuild	r	Additional information on the software version.
responseMaxBytes	r	The largest block of data that the RTU can process (receive/transmit) in one chunk. This information is used by the server (the M2M Gateway) during the communication to avoid overloading the RTU's resources.
sensorArraySize	r	The maximum number of tags that can be attached to the RTU.
comboArraySize	r	The maximum number of sensors that can be attached to the RTU.
gsmIMEI	r	IMEI of the GSM module.
gsmCCID	r	CCID of the SIM card.
gsmVersion	r	Firmware version running in the GSM module.
currentDate	r	Current date as returned by the internal Real Time Clock (RTC).
nextConnnectDate	r	Date of the next scheduled connection.
putdataDate	r	The date of the last data pushed on the server.
nextSyncDate	r	The next date when the RTU's internal RTC will be synchronized against the server.
date	r	Date of the last successful connection to the server.
uptime	r	Amount of time elapsed since the last reset.
gsmCellInfo	r	Cell info as transmitted by the GSM operator. Not all GSM operators transmit the cell info, so this attribute might be empty.
gsmMccMnc	r	Operator country (Mobile Country Code) and network (Mobile Network Code), as defined by the GSM Organization.
gsmLacCi	r	Local Area Code and Cell ID: this attribute is updated by the GSM modem before every attempt to connect over GPRS. The LAC and CI define a unique antenna tower (base station) of a GSM operator.
gsmRssiBer	r	Relative Signal Strength Indicator and Bit Error Rate. Both parameters are retrieved from the GSM modem before a connection attempt. RSSI ranges from 0 to 31 and BER from 0 to 7. For RSSI higher values means stronger signal, while BER should be 0. RSSI under 10 and/or BER higher than 3 usually signalize unstable communication and should be avoided when installing RTUs.

Sensor Attributes

Attribute	Type	Description
acquisitionMode	r/w	Defines how the sensor will be sampled. Following modes are defined: 0 – no sampling, sensor is disabled 1 – manual sampling (currently not implemented) 2 – automatic mode, defined by the *aquisitionSchedule* attribute.
acquisitionSchedule	r/w	Defines the sensor sampling schedule. It uses a cron syntax, ex: */10 means that the sensor should be sampled every ten minutes starting with minute 0. A complete cron string may specify minutes, hours, days, months and the day of the week. The smallest unit that can be set is therefore one minute. For additional details consult the unix man pages for *crontab (5)*.
sdiAddress	r/w	The SDI address of the sensor. This attribute is valid only for the T707-SDI variant.
sdiMethod	r/w	The acquisition method used; as per the SDI-12 specification, this can be M, C, R, MC, CC, RC and V. This attribute is valid only for the T707-SDI variant.
sdiPostmethod	r/w	An (optional) SDI-12 command sent after sampling the sensor. Usually this is an SDI-12 "X" command. This attribute is valid only for the T707-SDI variant.
thiesMethod	r/w	Defines the request sent to the Thies data-logger. It can currently be either "ds" or "dS". The second variant uses a checksum on the incoming data. This attribute is valid only for the T707-THI variant.
archiveInterval	r/w	Specifies the archiving interval on a Davis WeatherLink logger/interface. This attribute is valid only for the T707-DAV variant.
sdiInfo	r	Returns the result of the "I" command to an SDI-12 sensor. This attribute is valid only for the T707-SDI variant.
davisVersion	r	Returns the software version of the Davis WeatherLink logger/interface. This attribute is valid only for the T707-DAV variant.
thiesInfo	r	Returns the software version of the Thies data-logger. This attribute is valid only for the T707-THI variant.
lastDate	r	The last date the sensor was sampled.
nextDate	r	The next date when the sensor will be sampled.

Attribute	Type	Description
putdataDate	r	The last date the sensor values were pushed unto the server (M2M Gateway). Normally the equivalent RTU attribute is used, unless a sensor has a different schedule.

Tag Attributes

Attribute	Type	Description
sdiIndex	r/w	Defines the position of the tag's value in the string returned by a "Dn" or "Rn" command. The *sdiIndex* starts from 0, for the first position in the string. This attribute is valid only for the T707-SDI variant.
commandMode	r/w	Defines how commands will be issued to output tags. Following modes are defined: 0 – no commands, tag is disabled 1 – commands will be sent manually (currently not implemented) 2 – commands will be sent in automatic mode, defined by the *commandSchedule* attribute. This attribute is valid only for the T707-SDI variant.
commandSchedule	r/w	Defines the schedule to send commands to an output tag. It uses a cron syntax, ex: "0 1-5" means that the command will be sent between one and 5 AM, every hour at minute 0. A complete cron string may specify minutes, hours, days, months and the day of the week. The smallest unit is therefore one minute. For additional details consult the unix man pages for *crontab (5)*. This attribute is valid only for the T707-SDI variant.
sdiCommand	r/w	The string representing the command to be sent to an output tag; usually it is an "X" command (or a manufacturer custom command). This attribute is valid only for the T707-SDI variant.
sdiMethod	r/w	The acquisition method used; as per the SDI-12 specification, this can be "M", "C", "R", "MC", "CC", "RC" and "V". This attribute is valid only for the T707-SDI variant.
sdiPostmethod	r/w	An (optional) SDI-12 command send after sampling the sensor. Usually this is an SDI-12 "X" command. This attribute is valid only for the T707-SDI variant.

Attribute	Type	Description
conversionParams	r/w	A string representing a first or second degree polynomial that is used by the RTU to convert the input value; the converted value will be stored in the internal memory. This feature can be used to convert e.g. Fahrenheit degrees to Celsius, or vice-versa. As an example, the formula Celsius = 0,5555*Fahrenheit - 17,7778 (generic: y = a*x + b) translates to the string 0.5555 -17.7777. For second degree polynomials, three parameters are required, and their order is always from left to right (ex.: a, or a b, or a b c). This attribute is valid only for the T707-DAV variant.
dashValue	r/w	Value returned by the Davis WeatherLink that represents an invalid tag value. This attribute is valid only for the T707-DAV variant.
davisIndex	r/w	Defines the position of the tag's value in the string returned by a "DMPAFT" command. The *davisIndex* starts with 0, for the first position in the string. This attribute is valid only for the T707-DAV variant.
davisSize	r/w	Defines the the value's length (in bytes) in the string returned by the DMPAFT command. This attribute is valid only for the T707-DAV variant.
thiesIndex	r/w	Defines the position of the tag's value in the string returned by a Thies data-logger "ds" command. The *thiesIndex* starts from 0, for the first position in the string. This attribute is valid only for the T707-THI variant.
lastValue	r	The value returned by the tag the last time when the sensor was sampled.
status	r	The status of the tag after it was sampled the last time. 00 means OK, other values signify errors (e.g. 01 "missing data").
putdataDate	r	The last date the tag values were pushed unto the server (M2M Gateway). Normally the equivalent RTU or sensor attribute is used, unless a tag has a different schedule, or if the data flow was interrupted.

Error, Warning and Informational Messages

This section explains the meaning of various messages that can be issued by the T707 RTU. The messages are issued over the service connector (if a serial connection is present), as well as sent to the internal logging facility. The messages stored in the logging facility can be visualized using the "log" command (see also the section "Commands"), or on the M2M Gateway if a connection over GPRS is still possible.

General Information Messages

Note: *Error messages numbers 0, 1 and 2 are not used*

REBOOT	3	device has been rebooted
CONFIG_DATA_OK	4	configuration data ok
SYSTEM_OK	5	system started
FW_DOWNLOAD_FAIL	6	remote firmware download failed
FW_DOWNLOAD_OK	7	remote firmware download succeeded
FW_UPGRADE_FAIL	8	remote firmware upgrade failed
POWER_FAIL	9	power supply to the device failed

Initialization Errors or Warnings

CANNOT_ALLOC_XMEM	10	possibly out of xmem space
CONFIG_RAM_DATA_INVALID	11	RAM configuration data invalid
FLASH_RESTORE_FAILED	12	failed to restore from flash, RAM data cleared
GETCONFIG_REQUIRED	13	need to call the GETCONFIG rpc
FIFO_INCONSISTENT	14	FIFO found inconsistent and was cleared
LOG_INCONSISTENT	16	inconsistent log data

Remote Procedure Call (RPC) Related Messages

RPC_PARSER	20	parser/serializer error
RPC_TRANSPORT	21	transport error
RPC_OTHER	22	other RPC errors
RPC_PUTDATA	23	putdata incomplete, still data to send
REQ_GETLOGIN	24	invalid password, try a getlogin()
GETLOGIN_DONE	25	getlogin already tried once unsuccessfully
RPC_SETATTRIB	26	setattrib errors
RPC_TASKS	27	task related messages

Communication Task Messages

COMM_STATE	30	state changes in the comm finite states machine
COMM_HANGS	31	the COMM task hangs

```
NET_START              32    starting up the networking
NET_FAIL               33    failed to start networking
NET_OK                 34    networking succeeded
NET_STOP               35    tear down the networking
NET_DOWN               36    networking went down unexpectedly
WAPGW_DOWN             37    WAP gateway is unreachable
WRONG_PARAMS           38    wrong task parameters (e.g. at upgrade)
WAPGW_ALT              39    switch to alternate wap gateway
```

GSM Modem Related Messages

```
GSM_SIM                40    SIM card management related messages
GSM_SMS                41    SMS management related messages
GSM_CSD_CALL           42    CSD call related messages
GSM_INIT               43    general GSM modem initialization
GSM_MAIN_CELL_INFO     44    GSM main cell info
GSM_OTHER_CELL_INFO    45    GSM other cell info
```

XML Parser/serializer Messages

```
SERIALIZER_BUFFER_FULL  50    serializer buffer full related messages
SERIALIZER_HEADER_FULL  51    wbxml serializer header full
NULL_ATTRIBUTE_VALUE    52    null attribute value
```

WTP/WSP/HTTP Layers Error/debug Messages

```
WTP_ENGINE             60    WTP layer error/messages
WSP_CONNECT            61    connect error/messages
WSP_DISCONNECT         62    disconnect error/messages
WSP_HEADER             63    WSP header messages
HTTP                   66    http messages
```

Storage (FIFO) Messages

```
FIFO_STORAGE           70    FIFO storage messages
```

Data Acquisition Task Messages

```
DACQ_SDI               80    SDI-12 messages
DACQ_SDI_CMD           81    commands sent via some SDI attribute
DACQ_HANGS             82    the DACQ task hangs
DACQ_SDI_BOOTLOAD      83    SDI-12 sensor bootload messages
DACQ_DAV               84    DAVIS messages
DACQ_THI               85    Thies messages
```

Technical Specifications

Parameter	Value
GSM/GPRS	850/900/1800/1900 MHz
Interfaces	RS-485, RS-232, SDI-12
Supported protocols	TCP/IP, WAP, XML, WBXML
Internal storage	256 Kbytes, non-volatile (50'000 data points)
Sampling interval	Programmable (minutes, hours, days, weeks)
Power supply	External, 5.5 to 20 Volt
Power consumption	Standby, receive: avg. 1mA (at 12 V) Transmit max 300 mA (at 12 V) Absolute maximum 600 mA (at 5.5 V)
Operating temperature	-20ºC to +60ºC
Dimensions	110/75/55 mm
Weight	400 g
Environmental protection class	IP66