



Athena S.r.l.
Via del Progresso 36/38
36100 Vicenza - Italy
Tel: +39 0444 654127
Fax: +39 0444 562553
www.rationalseed.com
PI e CF: 03439740246
Numero REA: VI-325619
Capitale Sociale: 10000€ i.v.

ERMES Family Product User Manual

CMN00201800403

Document title	ERMES Family Product User Manual
Document code	CMN00201800403
Date	16/01/2012
Authors	Mirko Maistrello
Revision	Massimo Legnani
Approval	Mirko Maistrello; Massimo Legnani
Total pages	Front page + 25
State	Rev 3
Distribution	Internal; selected customers

Version history			
Code	Date	Authors	Note
CMN00201800401	10/28/2010	MM	
CMN00201800402	05/04/2011	MM	Introduction of: broadcast services; infrastructure and leaf mode; commands for setting timeout discovery; New ERMES-E and ERMES-X added.
CMN00201800403	27/12/2011	MM	Complete document review; New ModBus configuration interface description.

1	Introduction	3
2	Device Overview	3
2.1	Typical working sequence	4
2.2	Visualization Codes during Normal Activity	5
2.3	About Radio Communication over Mesh Networks.....	5
2.3.1	Dynamic Timing Adaptation to Network Structure	6
2.3.2	Smart Skip of Offline Slaves	6
2.4	Data Encryption	6
2.5	Application and Radio Stack Watchdogs.....	7
3	Installation and Commissioning.....	8
3.1	Electrical Connections	8
3.2	Sharing Network ID	9
3.3	Installation Quality Checking.....	10
3.3.1	Procedure Activation	11
3.3.2	Test End.....	11
3.4	ERMES Application Default Configuration	12
3.5	ERMES Radio Stack Default Configuration	13
3.6	Standard Network Usage	13
3.7	Types of ERMES nodes on a network.....	14
4	ModBus Configuration Interface	15
4.1	Radio Communication Stack Parameters.....	15
4.2	Application Parameters.....	16
4.3	Unique Serial Identifier	17
4.4	Network Identifier	18
4.5	RF Calibration Parameters.....	18
4.6	Firmware Version	18
4.7	Current Discovery Timeout	18
4.8	Quality Test.....	18
4.9	Quality Test Report.....	19
4.10	Network Structure Handler	19
5	Configuration via ASCII Terminal	21
5.1	Description of the necessary equipment	21
5.2	Ascii Interface Activation.....	21

5.3	Commands Description	22
5.3.1	Commands Syntax.....	22
5.3.2	Setting Command: SET	23
5.3.3	Reading Commands: GET	23
5.3.4	Report frames: REPORT	23
6	Acronyms	25

1 Introduction

The document contains the instructions for using the ERMES family devices and in particular:

- Device overview
- Installation and commissioning
- Configuration via ASCII terminal
- Configuration via ModBus commands

2 Device Overview

The ERMES family device is designed for transferring in a transparent manner data packets compliant with the ModBus RTU standard over wireless mesh network. Figure 2-1 shows a typical application of the ERMES device within a ModBus RTU network.

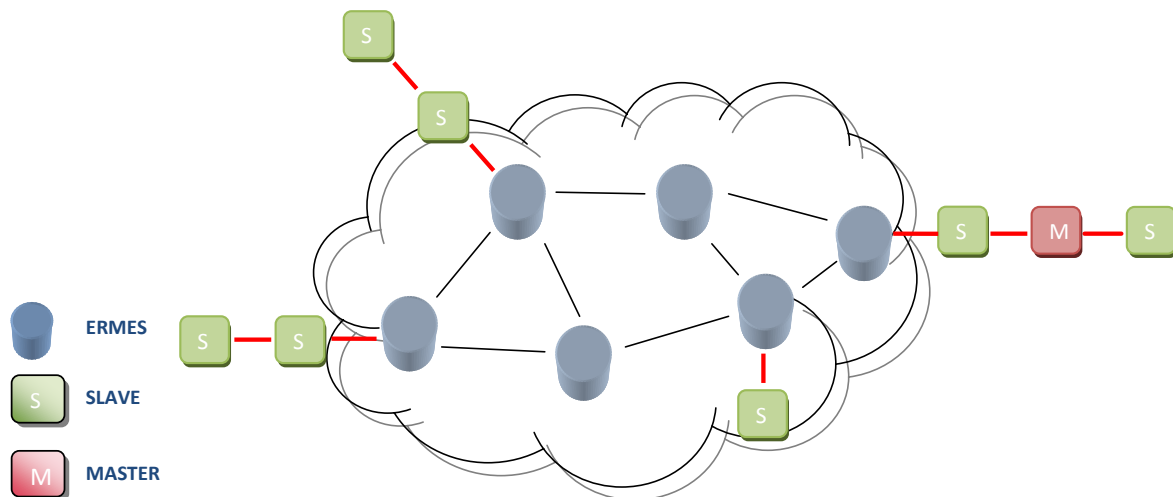


Figure 2-1

2.1 Typical working sequence

The typical working sequence of the system is the following:

1. The Modbus Master sends cyclically packets on the bus to the nodes belonging to its Modbus network
2. The ERMES device connected to the Bus Master, receives the packet and checks in memory whether there is already a valid path to the destination ModBus slave address written inside the packet
 - a. If the path exists then ERMES forwards the packet within the wireless network up to the radio node connected to the destination Modbus slave
 - b. If not, and if the recipient ModBus slave is not directly connected with the Bus Master, then ERMES activates a process for the route discovery. During this process any further request from the master device is discarded by ERMES.
 - i. If the discovery procedure succeeds, the packet is forwarded to radio node connected to the destination Modbus slave
 - ii. If the discovery procedure fails, the packet is discarded
 - c. If the recipient ModBus slave is directly connected with the Bus Master then the packet is discarded by ERMES
3. The ERMES device connected to the recipient Modbus slave, receives the packet from radio interface and extracts the ModBus packet forwarding it on its local bus. At this point it waits for the response.
 - a. If the response does not come from the slave (within the timeout period ModBus - AMT) ERMES device does not implement any further operation and returns to its normal activity.
 - b. In the case of a response from the Modbus slave, ERMES device encapsulates the received packet and forwards it through the radio network to the ERMES connected to the Modbus Master. So it returns to its normal activity.
4. If the ERMES device connected to the ModBus Master receives communication back from the device connected to the Modbus slave (within the timeout Radio - ART), it extracts the packet and sends it on its local bus. At this point the cycle is completed and the device returns to wait for further communications.
5. If the ERMES device connected to the ModBus Master does not receive communication back from the ERMES device connected to the Modbus slave (within the timeout Radio - ART), then it repeats the request for a number of times equal to the ARR application

parameter. If the ERMES device connected to the ModBus Master experiences a number of consecutive communication failures equal to AMCF parameter, the path is labeled as no longer valid.

2.2 Visualization Codes during Normal Activity

During normal operation, each ERMES device performs a periodic green LED (L1) flashing with a period of approximately 1s (see Figure 3-1). If there is neither traffic on air (that involves directly the ERMES node), nor on the bus, the LEDs L2 and L3 remain off.

L2 LED flashes after every packets transmission from the ERMES device to its local Bus.

L3 LED flashes after every packets transmission of the ERMES device over the air.

During normal network operation, the ERMES devices directly involved in the transmission of messages may flash both L2 and L3 or L3 only. In the first case the ERMES devices are terminal devices (such as those connected to ModBus master or ModBus slave directly addressed); in the second case the ERMES devices are intermediate node with the sole function of forwarding the radio packets.

2.3 About Radio Communication over Mesh Networks

Since the mesh networking is intended to replace the cable, it is inevitable to compare the performance of the system before and after the introduction of the radio network. This paragraph briefly describes the mechanisms in ERMES devices in charge of managing the radio network in order to make it reliable for the application layer.

From the physical layer point of view, in ModBus over RS485 all communications are point to point and the signal to noise ratio is normally enough to guarantee a very low PER. On the other side, radio networks are affected by a signal to noise ratio higher if compared with the cable, variable over time and unpredictable.

So, the radio channel compared to the cable is inherently less reliable. In order to improve the reliability in the point to point communication, specific mechanisms are implemented in order to give to the application level a communication channel with increased quality. These mechanisms are:

- Implementation of quality threshold for activating point to point links
- Implementation of algorithms for channel listening before transmit
- Implementation of algorithms for channel access delay with pseudo random windows
- Use of acknowledge in critical point to point communications
- Use of network acknowledge in critical communications
- Point to point packets retransmission
- End to end packets retransmission

All stated above, applies when paths exists and allow you to carry ModBus over the radio.

In a mesh network performance are further hampered because of:

- It is not possible to know in advance the network structure (number of devices, number of levels etc ...)
- Path are not known in advance
- Application environment is not known (interference, density equipment ... etc.)

The algorithms for discovering the paths are the most complex thing in the network and, as mentioned above, inevitably introduce latencies when some communication paths are not already present in the network. In this case, the network implements specific algorithms that introduce an additional layer of reliability in a so changeable network environment. These algorithms are:

1. Dynamic timing adaptation to network structure
2. Smart skip of offline slaves

2.3.1 Dynamic Timing Adaptation to Network Structure

The most critical thing of a mesh network is the routes discovery. This mechanism does not find its counterpart in wired Modbus network and introduces communication latency when paths does already exist.

A stable ERMES network is not affected of this latency since the paths are stored in distributed structures and used until they show repeated failures.

In order to reduce the time required to complete discovery algorithms it is also implemented a mechanism for determining the dynamic network structure. In fact, the discovery timeout is set in advance without knowing the specific network structure (number of nodes, number of levels, distribution of nodes over levels) and during normal network operations, the algorithm evaluates the network structure and adapts the discovery timeout to the specific condition in order to be more reactive to path changing.

2.3.2 Smart Skip of Offline Slaves

The smart skip of offline slaves algorithm allows to dynamically exclude from polling groups of ModBus devices which show repeated communication failures . The exclusion time of such devices (MCBT parameter) and the number of consecutive discovery failures (MCDF parameter) are two configurable parameters via the configuration interface. During the time of exclusion, all requests from the Modbus Master directed to the excluded slave are discarded.

2.4 Data Encryption

The encryption algorithm is designed to obfuscate the Modbus data transferred over the radio network so that any sniffer is not able to correctly decode the data. The algorithm has the characteristics:

1. Time-varying encryption: the same packet transmitted at two different times is different
2. Encryption depends on the node: the same package ModBus transmitted at the same time by two different nodes is different

Data encryption can be enabled or disabled via the configuration interface. Equipments with encryption enabled and equipments with encryption disabled cannot coexist in the same network.

2.5 Application and Radio Stack Watchdogs

Each ERMES device integrates two safety watchdogs that restarts the unit in case of an unexpected malfunction which precludes communication functionality.

The radio watchdog monitors the presence of radio communications that reach the application level and in case they are not present for specific time, it forces a reset of the device.

The application watchdog monitors the communication on RS485 bus and in the case they are not present for a specific time, it forces a reset of the device. The application watchdog is enabled only on the ERMES device connected to the Modbus Master as it is the only device on which there is continuous communication flow over the RS485 bus.

The radio watchdog and the application watchdog are configurable via the configuration interface.

3 Installation and Commissioning

3.1 Electrical Connections

Figure 3-1 shows how to connect an ERMES device to the BUS and to the power supply.

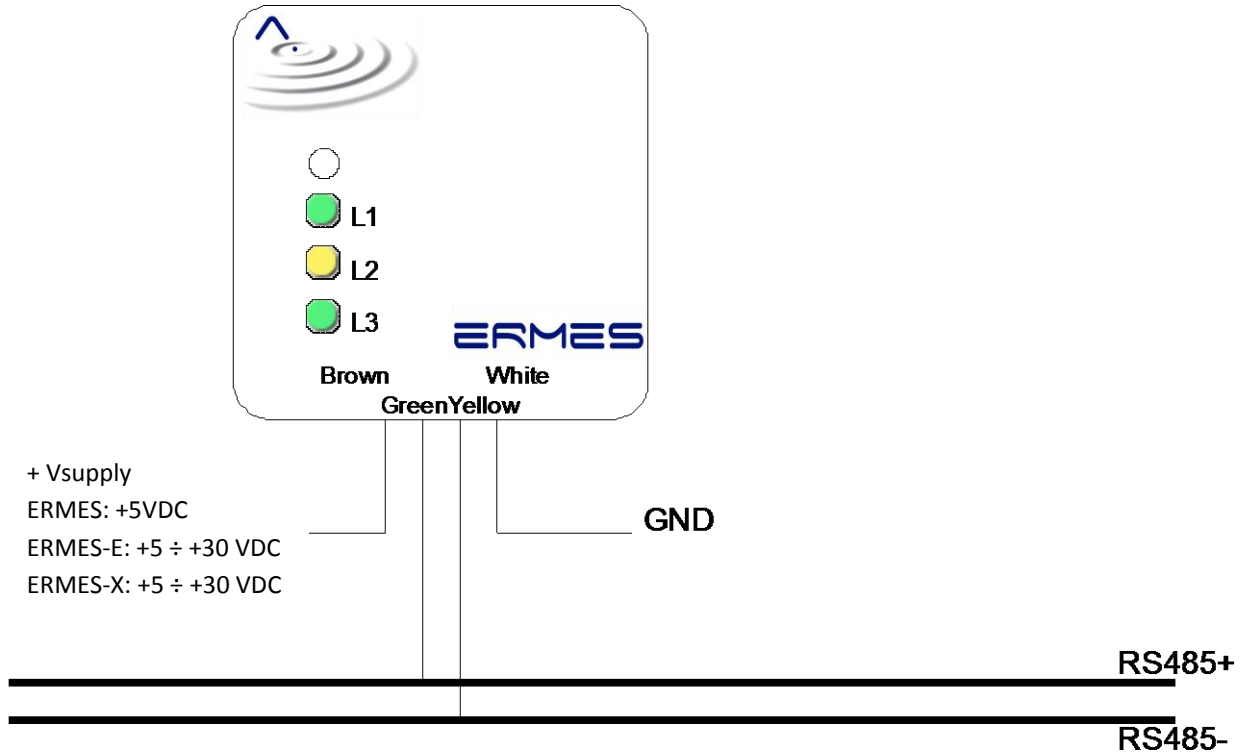


Figure 3-1

Similar connections apply to the device ERMES-X (same color of the cables). In this case the access to LEDs visualization and to the push-button is possible when the enclosure is open. The LEDs and button position on the electronic board is the same in ERMES and ERMES-E device.

Warning: ERMES does not have an isolated power supply stage nor an isolated RS485 interface. It is strongly recommended to provide an isolated power supply to ERMES in order to prevent damage to devices connected to the same plant. For details on the allowed supply voltages please refer to the specific installation manuals.

Upon connecting the device to power supply, a startup LEDs flashing sequence starts. The startup sequence includes:

1. Turning on all the LEDs for about 1 second
2. Switching off all the LEDs for about 2 s
3. Turning on all the LEDs for about 50 ms
4. Switching off all the LEDs for about 2 s
5. Starting standard view

3.2 Sharing Network ID

During installation, the only thing required by the installer is setting the same network identifier for all ERMES devices belonging to the same system. In fact, each ERMES device discards communications that do not belong to its network in order to allow the proper functioning of partial or total overlapping radio networks. The only exception to this behavior is given for communication with broadcast network ID in close proximity. Every ERMES device is manufactured with its own unique 32-bit identifier (serial number) and a unique network identifier. By default, the network identifier is equal to the serial number and, while the serial identifier cannot be changed, the network ID can be individually configured using one of the methods described in the manual.

The following describes the mechanism for exchanging the network identifier without involving any external configuration equipment. This procedure must be repeated on all devices belonging to the same system minus one. This device (hereinafter referred to as passive) can be chosen at random and it will be the network identifier dispatcher.

The acquisition and memorization of the identifier always involves a pair of ERMES apparatus, an "active" and a "passive". The "active" device requests the network identifier to the "passive" and once received and stored the answer in nonvolatile memory, it automatically asserts a reset and restart with the new network identifier. The "passive" ERMES device is only in charge of dispatching its network ID.

The procedure should be performed placing the "active" and the "passive" devices in close proximity (less than 3 cm, as shown in Figure 3-2) and activating the following procedure in "active" device:

1. Press the button and release quickly
2. Then press the button and maintain it pressed

The sequence must be performed without breaks in between. The correct sequence is shown by a simultaneous flash of all the LED in the "active" device. The flash corresponds also to the network identifier radio request transmission to the "passive" device.

The correct procedure completion (new network identifier saved in nonvolatile memory) is displayed by the startup visualization sequence immediately after the confirmation of successful request activation.

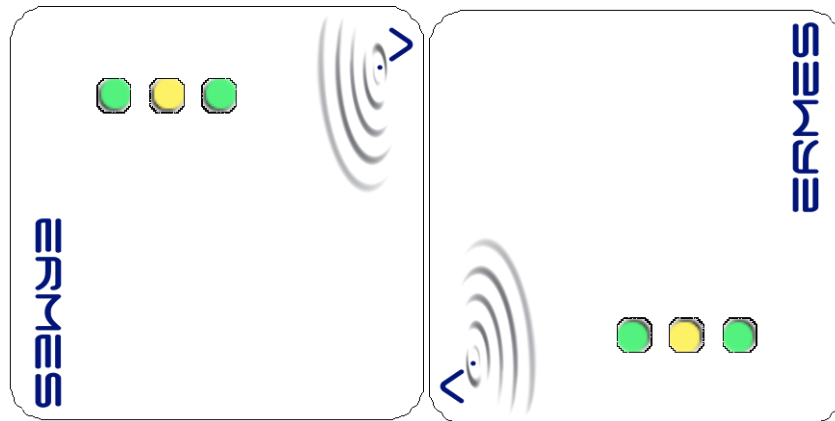


Figure 3-2

3.3 Installation Quality Checking

The evaluation of the quality of the installation is an optional task that the installer can make on all ERMES except the one connected to the Modbus Master. The procedure for assessing the goodness of the installation lasts about 2 minutes and must be activated on a device at a time. During this period of time the device repeatedly tries to communicate with ERMES device connected to the Modbus Master acquiring information about the robustness of the communication and the quality of network deployment. At the same time of the radio test, the device also discovers all the devices connected to its bus pinging all the addresses from 1 to 247. The Modbus slave addresses discovered, are sent using the radio test packets to the ERMES node connected to the ModBus Master.

In order to enable this assessment, a jumper should be mounted on the device connected to the Modbus Master. By default every ERMES device ignores its connection to the Master or to the Slave but in this case the information is necessary.

The installation of the jumpers is shown in Figure 3-3.

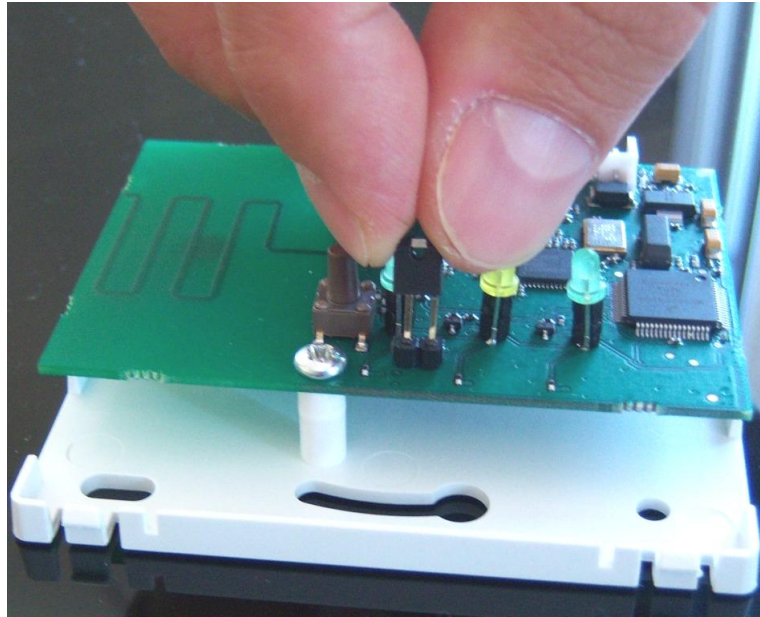


Figure 3-3

3.3.1 Procedure Activation

The activation of the mechanism for evaluating the quality of the connection is performed by pressing the button on the ERMES device of interest. In particular, the sequence is:

1. press the button and hold it for at least half a second but less than 3 seconds
2. release the button

The correct activation of the procedure is shown by repeated and simultaneous flashing of all LEDs with a period of about 1s. Such sequence of flashes is maintained for all the duration of the test.

3.3.2 Test End

The test is closed automatically by the device after about 2 minutes. At the end of the test, the device displays the test result. If the green LED (L3) is lit, it means that the link is stable and the mounting location is "good". If the yellow LED (L2) is lit, it means that the test has failed: the mounting position of the apparatus is not "good".

Test Result	Description
Green Led stably on	PER with ERMES Master Radio < 20%
Yellow Led stably on	PER with ERMES Master Radio ≥ 20%

Table 3-1

After the end of the test, the device keeps displaying the results for about half an hour and then resumes the normal sequence of flashing. In any case the apparatus, after the end of the test is immediately able to operate properly. If you want to return to the normal visualization, simply press and quickly release the button on the device.

If the installer is interested in finding out more details about the test result, after the end of the test (with LED green or yellow LED stably on), he can repeat the same procedure used for the test's activation. In this case the device starts a complex view in which:

1. The L1 LED repeats periodically with period 6s, 5 flashes separated by a pause of 1 s - LED L1 simply gives the timing.
2. The LED L2 shows, periodically synchronized with L1, a number of flashes from 1 to 5 corresponding to the PER calculated during testing. The Table 3-2 specifies the values of PER related to the number of blinks.
3. LED L3 shows, periodically synchronized with L1, a number of flashes from 0 to 5 corresponding to the score of the deployment quality from the perspective of the node from which the test is activated.

In particular, L3 LED shows a score of connectivity. Every single point is given in reports on network conditions detected during the test:

- The average length of the network path to the Master less than 2 jumps: 1 point
- Loss of the route during the test for less than 3 times: 1 point
- Existence of alternative paths to the Master or Master in direct coverage: 1 point
- Number of neighbors is greater than 2 and less than 5: 1 point
- Average RSSI of neighboring nodes > -80 dBm: 1 point

L2 LED number of blinks	PER Range [%]
1	> 60 %
2	40% ÷ 60%
3	20% ÷ 40%
4	5% ÷ 15%
5	< 5%

Table 3-2

As the simple visualization, also this detailed view is maintained for about half an hour. If you want to return to the normal situation, simply press and quickly release the button on the device.

More details about the test result can be obtained by querying the device using one of the configuration methods described in Chapters 4 and 5.

3.4 ERMES Application Default Configuration

Each ERMES device comes with a default application configuration shown in Table 3-3. Any change to the default settings can be obtained through one of the configuration methods described in chapters 4 and 5.

Parameter	Description	Firmware setting	Value
AMT	Application ModBus Timeout	0x04	125 ms
ART	Application RF Timeout	0x10	500 ms

ARR	Application Radio Retransmission	0x03	3
AMCF	Application ModBus Consecutive Failure	0x04	4
AWT	Application Watchdog Timeout	0x04	20 min
MCDF	Max Consecutive Discovery Failure	0x03	3
MCBT	Max Communication Block Time	0x08	40 min
MSC	ModBus Serial Configuration	0x0C	9600 8 – n – 1
ENC	Encryption Enable	0x00	FALSE

Table 3-3

3.5 ERMES Radio Stack Default Configuration

Each ERMES device comes with a default radio stack configuration shown in Table 3-4. Any change to the default settings can be obtained through one of the configuration methods described in chapters 4 and 5.

Parameter	Description	Firmware setting	Value
RFPW	RF Output Power	0x3F	13 dBm
LNKTH	Link Activation Threshold	0x08	8 dBm offset over the sensitivity threshold
CCATH	Clear Channel Threshold	40	-80 dBm
DISCT	Maximun discovery timeout	4000	3,9 s ¹
RWT	Radio Watchdog Timeout	0x04	20 min
NT	Node Type (Infrastructure or Leaf)	0x00	Infrastructure

Table 3-4

3.6 Standard Network Usage

For the best use of Ermes devices to transport ModBus protocol it is necessary to be aware of:

- 1. A necessary condition for the operation of the network is the affixing of the jumpers as shown in Figure 3-3 to the node connected to the master device.**
- All sections of RS485 bus connected to an ERMES device must be correctly terminated and biased.
- The transport of ModBus Brradcast packets (ModBus address 0x00), while being supported by the network, it is strongly not recommended insofar as, on mesh multihop topology networks, the transport of broadcast messages involves long communications times and a low reliability in reaching all the networks elements. It is therefore recommended to limit the use of broadcast communications only to the phases supervised

¹ Every failed discovery will be repeated twice in order to get rid of asymmetric links

by an operator (for instance during installation) or for the transport of sporadic messages the reliability of which is not required.

4. Over networks with large numbers of devices, prevent the densification of the infrastructure elements within the same range. For more details, see section 3.7.

3.7 Types of ERMES nodes on a network

Within a ModBus transport network, ERMES devices, while behaving in an identical manner from the application point of view, can take two different behaviours in the routing framework:

- Infrastructure Node
- Leaf Node

An infrastructure node acts as intermediary in communication between a sender (the device connected to the Modbus Master) and a recipient (the unit is connected to the slave destination of the message). In the event that there is no direct visibility between the sender and the recipient, the infrastructure nodes combine the transmitter and receiver in a logical link.

A leaf node is a terminal element of the communication and is not able to act as intermediary in communication between a sender (the device connected to the Modbus Master) and a recipient (the unit is connected to the slave destination of the message).

By default all ERMES devices are infrastructure nodes.

The possibility of downgrading an infrastructure node in a leaf node is made available solely in order to optimize the operation of the network and is applicable only for installations that involve a large number of devices. The typical example is in fact given by a network in which, to satisfy application requirements (high number of slave ModBus), there is a large number of ERMES devices in the same range. Since by default all the elements are infrastructure node, during the paths discovery operations all are involved in the procedure critically slowing down the network. Instead, in the specific example, such procedures could be made much simpler by downgrading some of the equipment at leaf nodes. The best way would be to limit the number of infrastructure node in order to:

1. Ensure the radio coverage of the application environment.
2. Ensure the redundancy of the paths to have a robust network in the presence of change of environmental framework.
3. Avoid network congestion due to a too high concentration of infrastructure elements.

4 ModBus Configuration Interface

From the configuration point of view, each ERMES device behaves as a Modbus slave with address 254 (0xFE) and using that address, the user can read or write internal memory areas in order to:

1. Read and / or modify the default configuration
2. Activate test procedures
3. Read the state apparatus
4. Run an advanced diagnostic on the apparatus

All ModBus memory regions are modeled in terms of holding register. The allowed Modbus commands are therefore:

- Read Holding Register (0x03)
- Write Multiple Register (0x10)

Regarding register of non-volatile memory, it is not allowed to read and write arbitrary groups of that registers. These areas must be read or written by blocks as shown in Table 4-1.

Only certain registers are write enabled for the user. Table 4-1 shows the map of the Modbus registers of the ERMES device.

Reg address	Description	Size [word - 16 bit]	User Policy
0	Radio parameters	4	R/W ²
4	Application parameters	5	R/W ²
9	Unique serial number	2	R
11	Network identifier	2	R/W ²
13	RF calibration parameters	4	R
17	Firmware version	2	R
19	Current discovery timeout	1	R
20	Quality test	1	R/W
21	Quality test report	11	R
32	Network structure handler	2	R

Table 4-1

4.1 Radio Communication Stack Parameters

The memory area containing the parameters of the communication stack is divided as shown in Table 4-2:

Address	Description	Valid values
0 - Byte H	Transmit power	0x00 ÷ 0x3F; 0xFF 0x00 → RF OFF; 0x3F → Max Output Power + 13 dBm

² After that command, ERMES automatically restarts

		0xFF → Used with write command, it sets the default value (0x3F)
0 - Byte L	Link activation threshold	0x00 ÷ 0x5A; 0xFF ³ 0x00 → -120 dBm; 0x3F → -30 dBm; 0xFF → Used with write command, it sets the default value (0x08)
1 - Byte H	Clear channel threshold	0x00 ÷ 0x5A; 0xFF ³ 0x00 → -120 dBm; 0x3F → -30 dBm; 0xFF → Used with write command, it sets the default value (0x28)
1 - Byte L	Maximum discovery timeout	0x03E8 ÷ 0xFFFF; 0xFFFF 0x03E8 → 1s 0xFFFF → Used with write command, it sets the default value (0x0FA0)
2 - Byte H		
2 - Byte L	Stack Watchdog Timeout (step of 5 min)	0x00 ÷ 0xFE; 0xFF 0x00 → Radio Watchdog Disabled; 0xFF → Used with write command, it sets the default value (0x04)
3 - Byte H	Node type (leaf or Infrastructure)	0x00 ÷ 0xFE; 0xFF 0x00 → Infrastructure node 0x01 → leaf node 0x02 ÷ 0xFF → Used with write command, it sets the default value (0x00)
3 - Byte L	Not Care	

Table 4-2

4.2 Application Parameters

The memory area containing the application parameters is divided as shown in Table 4-3:

Address	Description	Valid values
4 - Byte H	Number of retransmissions, Master side in case of no response – (ARR);	0x01 ÷ 0xFE; 0xFF 0xFF → Used with write command, it sets the default value (0x03)
4 - Byte L	Number of consecutive failures before deleting a path (and issuing a new path discovery) – (AMCF);	0x01 ÷ 0xFE; 0xFF 0xFF → Used with write command, it sets the default value (0x04)
5 - Byte H	Application RF Reply timeout (step of 31,25ms) – (ART);	0x01 ÷ 0xFE; 0xFF ⁴ 0xFF → Used with write command, it sets the default value (0x10 → 500 ms)
5 - Byte L	Application ModBus Reply timeout (step of 31,25ms) – (AMT);	0x01 ÷ 0xFE; 0xFF ⁴ 0xFF → Used with write command, it sets the default value (0x04 → 125 ms)
6- Byte H	Application Watchdog Timeout (step of 5	0x00 ÷ 0xFE; 0xFF

³ Pay close attention to the change of threshold parameters. Too high activation thresholds or too low transmission thresholds may preclude proper operation of the device.

⁴ The ModBus and radio application timeouts are optimized for the defaults bit rate of 9600. In the case of changing the bit rate may also be appropriate to change also the timeouts.

	min)	0x00 → Application Watchdog disabled; 0xFF → Used with write command, it sets the default value (0x04 → 20 min)
6 - Byte L	Number of consecutive discovery failures before issuing the polling skip	0x01 ÷ 0x0F; 0xFF 0xFF → Used with write command, it sets the default value (0x03)
7 - Byte H	Polling skip period (step of 5 min)	0x00 ÷ 0xFE; 0xFF 0x00 → Smart skip disabled; 0xFF → Used with write command, it sets the default value (0x08 → 40 min)
7 - Byte L	MODBUS driver setting – (MSC)	0x00 ÷ 0x1F; 0xFF 00: 1200-N-8-1 01: 1200-E-8-1 02: 1200-O-8-1 03: 1200-N-8-2 04: 2400-N-8-1 05: 2400-E-8-1 06: 2400-O-8-1 07: 2400-N-8-2 08: 4800-N-8-1 09: 4800-E-8-1 0A: 4800-O-8-1 0B: 4800-N-8-2 0C: 9600-N-8-1 0D: 9600-E-8-1 0E: 9600-O-8-1 0F: 9600-N-8-2 10: 19200-N-8-1 11: 19200-E-8-1 12: 19200-O-8-1 13: 19200-N-8-2 14: 38400-N-8-1 15: 38400-E-8-1 16: 38400-O-8-1 17: 38400-N-8-2 18: 57600-N-8-1 19: 57600-E-8-1 1A: 57600-O-8-1 1B: 57600-N-8-2 1C: 115200-N-8-1 1D: 115200-E-8-1 1E: 115200-O-8-1 1F: 115200-N-8-2 0xFF → Used with write command, it sets the default value (0x0C)
8 - Byte H	Encryption	0x00 ÷ 0xFE; 0xFF 0x00 → Not encrypted transmission 0x01 → Encrypted transmission 0x02 ÷ 0xFF → Used with write command, it sets the default value (0x00)
8 - Byte L	Not Care	

Table 4-3

4.3 Unique Serial Identifier

The memory area starting from the Modbus register 9 contains a unique serial device's radio identifier. This value is available only in read mode and the MSB matches the high byte of the register 9.

The unique serial number of the device is a 32-bit integer.

4.4 Network Identifier

The memory area starting from Modbus register 11 contains the network identifier. This value is available to the user both in read and write mode and the MSB matches the high byte of the register 11.

The network identifier of the device is a 32-bit integer and must be shared by all devices that belong to the same network. In production the network identifier is set equal to the serial number.

0x00000000 The network identifier is not allowed.

4.5 RF Calibration Parameters

The memory area starting from Modbus register 13 contains the RF calibration parameters of the device. This value is available to the user only in read mode and occupies 4 Modbus registers.

4.6 Firmware Version

The memory area from Modbus register 17 contains the firmware version loaded on the device. This value is available to the user only in read mode and the MSB matches the high byte of the register 17.

The firmware version is a 32-bit integer.

4.7 Current Discovery Timeout

The memory area starting from Modbus register 19 contains the current discovery timeout, result of the dynamic network structure adaptation algorithm. On startup this value coincides with the maximum timeout discovery and then is reduced according to the real network structure. The dynamic timeout discovery is available for the user in read mode and takes only a single register.

4.8 Quality Test

The quality tests register is located at address 20 and contains a flag that indicates whether or not the connection test with the ERMES device connected to the Master is in progress. A value of 1 indicates that the test is in progress, the value 0 indicates that no test is active.

The register is available both in reading and writing mode. In writing mode it can be used as a strobe register to activate the test with a Modbus command (same function described in section 3.3.1). Whatever the value is written, if a test is not already active, it will be enabled.

4.9 Quality Test Report

The memory area starting from register 21 contains a series of Modbus registers that hold the communication test result. The area has a size of 11 records and it is available to the user only in read mode. Table 4-4 describes the meaning of the various registers.

Address	Description	Valid values
21	Number of transmission requests	0x0000 ÷ 0xFFFF
22	Number of successful transmissions (ack received)	0x0000 ÷ 0xFFFF
23	Number of failures because of channel busy	0x0000 ÷ 0xFFFF
24	Number of physical frame detection	0x0000 ÷ 0xFFFF
25	Number of valid application packets received	0x0000 ÷ 0xFFFF
26	Number of collected neighbours	0x0000 ÷ 0xFFFF
27	Number of alternative paths toward the Master	0x0000 ÷ 0xFFFF
28	Average hop to the destination	0x0000 ÷ 0xFFFF
29	Number of successful discoveries	0x0000 ÷ 0xFFFF
30	Number of failed discoveries	0x0000 ÷ 0xFFFF
31 - Byte H	Average neighbour quality	00 – 5A (00: -120 dBm 5A: -30 dBm)
31 - Byte L	Not Care	

Table 4-4

4.10 Network Structure Handler

The memory area starting from Modbus register 32 contains the link to the internal volatile device memory allowing to visit the internal network tables. The area has a size of 2 records and it is available to the user in read-only mode. Table 4-5 shows the meaning of the various registers.

Address	Description	Valid values
32 - Byte H	Path table size	0x0000 ÷ 0xFFFF
32 - Byte L	Path table address	0x0000 ÷ 0xFFFF
33 - Byte H		
33 - Byte L	Not Care	

Table 4-5

Starting from the memory address that contains the table, the user can visit the whole table structure stored on the device.

Specifically, the table is an array of paths of the following basic structures:

```
typedef struct{
    typeaddr addr;
    byte listsize;
    typemodbustableentry* listptr;
    byte nexthopid;
    byte pathlen;
    byte quality;
    byte failcount :4;
```

```

byte linktype      :2;
byte lnkvalcount  :2;
} typepathtableentry;

```

Where:

`addr` is the radio destination unique serial number

`listsize` is the number of ModBus slave connected to the radio destination

`listptr` is the ModBus Slave table pointer

`nexthopid` is the next hop index within the table in order to reach the radio destination node

`pathlen` is the path length

`quality` is the average RSSI with `addr` (only valid if `addr` is directly reachable)

`failcount` is the number of consecutive failures with the radio destination `addr`

`linktype` is the link type

`lnkvalcount` is a counter for asymmetric links

Each table row takes 11 bytes. In each row the user can also find the length and the pointer to the ModBus table of slaves connected to the `addr` device.

The ModBus table is an array of the following basic structures:

```

typedef struct{
    typemodbusaddr addr;
    byte failcount   :4;
    byte consdiscfail :4;
    byte blockcount;
} typemodbustableentry;

```

Where:

`addr` is the ModBus slave address

`failcount` is the number of consecutive communication failures with the ModBus destination `addr`

`consdiscfail` is the number of consecutive discovery failures with the ModBus destination `addr`

`blockcount` is the polling skip counter

Each table row takes 3 bytes.

5 Configuration via ASCII Terminal

The same settings that can be activated via the standard Modbus interface are also available through ASCII interface.

5.1 Description of the necessary equipment

The equipment required for reading and setting the configuration parameters via ascii terminal are:

- Personal Computer with Hyperterminal or equivalent software and serial interface (native or USB - to - Serial converter).
Serial interface configuration:
115200baud 8-n-1 without flow control.
Since ERMES does not show echo of typed characters, you should activate the "Echo locally typed characters"
- Serial cable for connection
- RS232 - RS485

Figure 5-1 shows the recommended connection diagram for ERMES configuration.

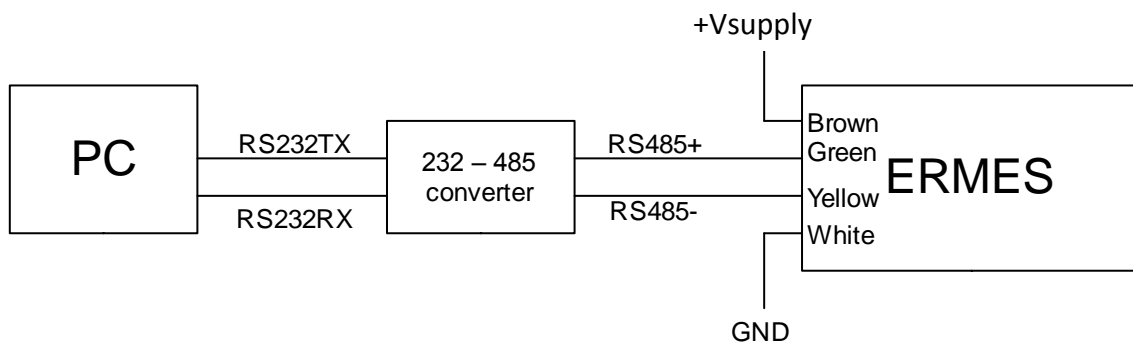


Figure 5-1

Warning: ERMES does not have an isolated power supply stage nor an isolated RS485 interface. It is strongly recommended to provide an isolated power supply to ERMES in order to prevent damage to devices connected to the same plant. For details on the allowed supply voltages please refer to the specific installation manuals.

5.2 Ascii Interface Activation

The Ascii configuration interface is activated by holding down the button during the power up. The success of the operation is given by a flash of yellow LED after the startup sequence. In addition to yellow LED flashing, on the terminal should appear:

ATHENA RSC - ERMES - CONFIG TOOL

At this point the button should be released and the device is ready to receive ASCII commands.

5.3 Commands Description

The commands accepted by ERMES are divided into two groups: GET (**G**) and SET (**S**). GET commands are used to read the value of some registers described in Table 4-1 while the SET commands are used to modify some configuration register (shown in Table 4-1 and marked as write enabled).

ERMES replies to every command received with a report message (**R**) reporting the required data (in case of GET) or confirming the settings (in case of SET).

Table 5-1 shows the matching between ascii commands and ModBus command.

Address	Description	ASCII code	User policy
0	Radio parameters	R	R/W ²
4	Application parameters	A	R/W ²
9	Unique serial number	S	R
11	Network identifier	I	R/W ²
13	RF calibration parameters	C	R
17	Firmware version	W	R
19	Current discovery timeout	T	R
20	Quality test	X	R/W
21	Quality test report	L	R
32	Network structure handler	Q	R

Table 5-1

The SET commands are enabled only on registers blocks marked as write enabled.

5.3.1 Commands Syntax

Commands are composed of two characters and one or more numeric parameters. Each command ends with a carriage return <CR><LF> (Ascii 13; Ascii 10).

The first character can be **S** in the case of SET command or **G** in the case of GET command.

The second character is the requested command as shown in Table 5-1, third column.

The command parameter consists of one or more hexadecimal numbers depending on the type of command.

Every GET or SET command provides a response (report) whose syntax is typically:

R:Xx...x<CR><LF>

iiii is the starting address of the written register block (in hex)

nnnn is the number of written registers (in hex)

And with GET type commands the device responds with a string formatted as follows:

R:nnxx...xx

where:

nn is the number of read bytes (in hex)

xx...xx is the content (in hex) of records read

In case of invalid command the generic reply is the following:

R:ER

6 Acronyms

Table 6-1 shows the list of acronyms.

Acronym	Meaning
AMCF	Application ModBus Consecutive Failure
AMT	Application ModBus Timeout
ARR	Application Radio Retransmission
ART	Application RF Timeout
AWT	Application Watchdog Timeout
ENC	Encryption Enable
MCBT	Max Communication Block Time
MCDF	Max Consecutive Discovery Failure
MSC	ModBus Serial Configuration
PER	Packet Error Rate
RSSI	Received Signal Strength Indication

Table 6-1