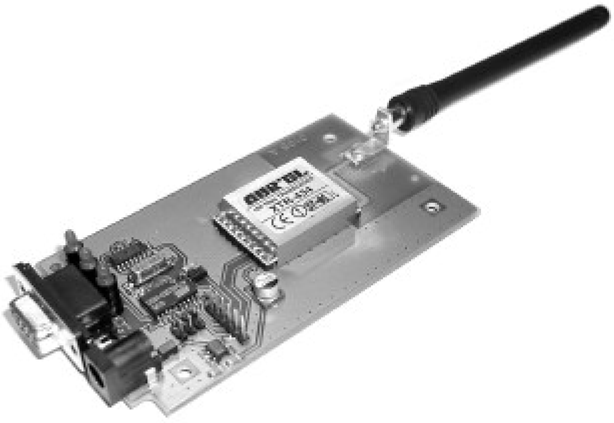
Single channel data transceiver module WIZ2-434

Available models:

- **WIZ2-434-RS**: data input by RS232 (±12V) logic, 9-15V supply
- **WIZ2-434-RSB**: same as above , but in a plastic shell.



The WIZ2-434-x modules are intelligent transceivers that support a digital data packet transfer of any length (1-96 bytes).

The data transferred to the remote unit are error-free since, at the arrival, are verified and confirmed.

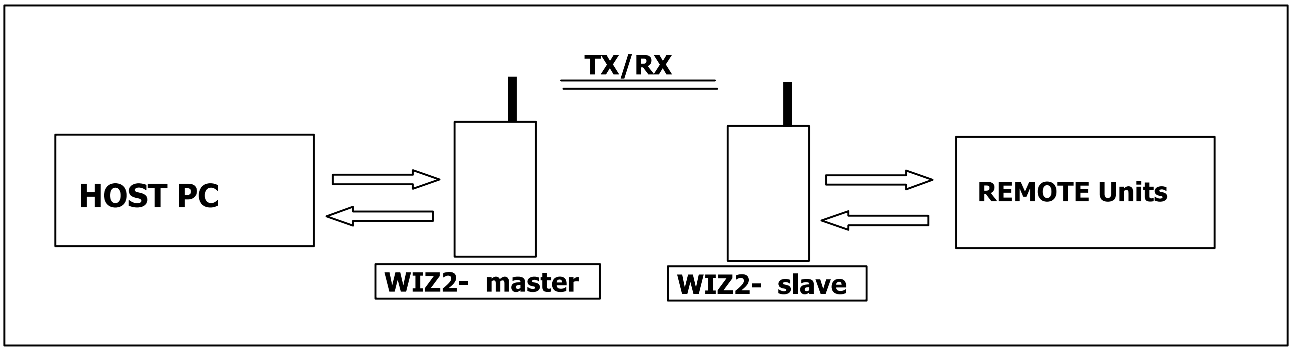
- Half-duplex serial connection with RS232 protocol, with a speed range programmable (9600, 19200, 57600, 115200 b/s)
- Data s packet length: 1-96 bytes
- Packet validity control at the reception
- RF connection through a 433.92 MHz band
- 2FSK modulation
- Status LED (Power ON, TX, RX)
- It complies with ETSI 300-220 regulations
- 9-15 V power supply

TABLE OF CONTENTS

Technical features are subject to change without notice. AUR°EL S.p.A does not assume responsibilities for any damage caused by the device s misuse.

Introduction

WIZ2-434-x modules allow to connect, via radio, two or more units; they supply a complete radio frequency system, by integrating receiver, transmitter and antenna and by indipendently administrating a low level packaging of the data sent from a control unit (for instance a Personal Computer), and the managing functions of the radio link interconnection.
They are supplied by a voltage ranging from 9 to 15V.



A 96 bytes maximum length data packet can be sent to the transmitter module by a personal computer; the data to be transmitted are memorized in an input buffer and forwarded to the remote connected WIZ2 module, to which are addressed.

The packet received by the WIZ2 module will be decoded, stored in the reception buffer and, in case, confirmed. It shall be forwarded, then, to the downstream connected unit, through a serial port.

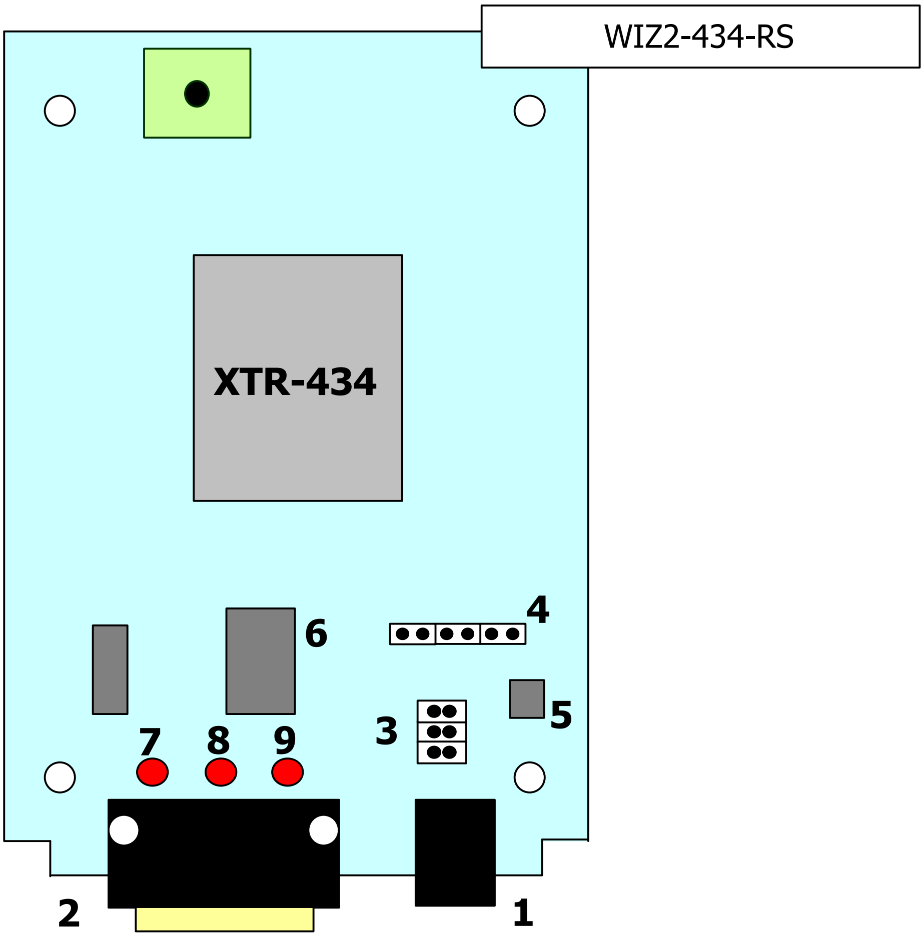
The WIZ2 modules bear a *polling* system where a master unit can address up to 255 slave units.

The host personal computer dispatches, to the WIZ2 module, the data preceded by an address byte of the remote module; the data exchanging administration unit adds the preamble and error control bytes.

Technical features are subject to change without notice. AUR°EL S.p.A does not assume responsibilities for any damage caused by the device s misuse.

1. Lay-out

The WIZ2-434-RS model foresees a 9-15V voltage range, supplied to the DC (1) connector and signals in RS232 logic on the female, 9 pins D (2) connector.



1	9-15V DC supply connector (internal Vcc, external GND)	6	Microcontroller
2	RS232, female, 9 pins, D type, connector	7	POWER signalling LED
3	Programming jumper	8	TX signalling LED
4	ICP jumper *(In Circuit Programming)*	9	RX signalling LED
5	5V voltage stabilizer		

Technical features are subject to change without notice. AUR°EL S.p.A does not assume responsibilities for any damage caused by the device s misuse.

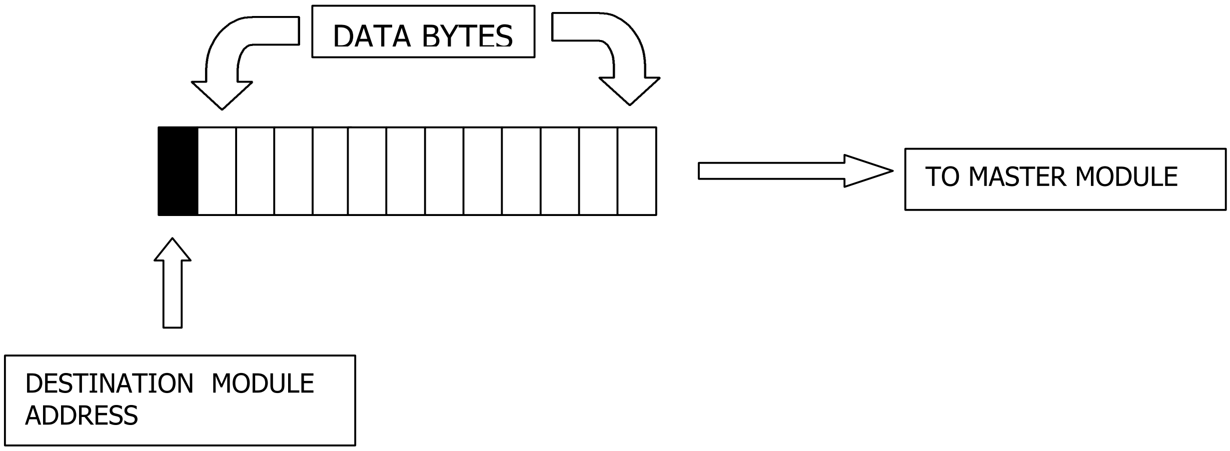
2. Addressing

The WIZ2 modules are conceived to be applied in a master slave system (255 maximum slave number). The distinction between master and slave units is of software type, according to the address contained in the ADDRESS register:

- ADDRESS = 0: system s MASTER module;
- ADDRESS = 1   255: SLAVE module.

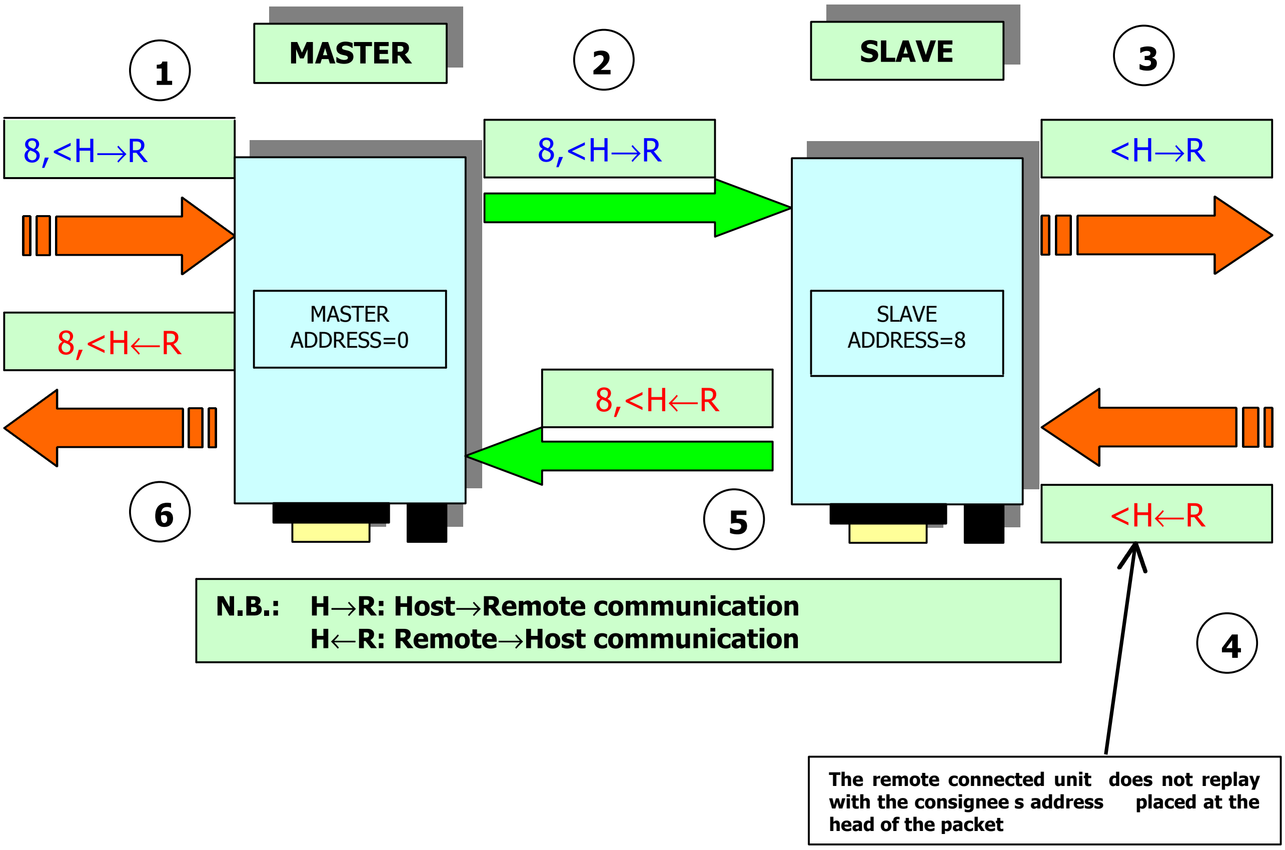
The system must always have a master module programmed with a 0 address.

The microcontroller fitted inside the WIZ2 modules, administrates the packing of the data entering from port RS232; the data package addressing to be transmitted is carried out by the external system requiring the connection with a remote unit. The addressing byte must be placed before the data bytes.



2.1 Conversation administration between units

Illustrated here below is the addresses management procedure in a master slave conversation. In this specific example reference is made to slave module with 8 address:


MASTER
SLAVE
1
2
3
8,<H→R
8,<H→R
<H→R
MASTER
ADDRESS=0
SLAVE
ADDRESS=8
8,<H←R
8,<H←R
<H←R
6
5
4
N.B.: H→R: Host→Remote communication
H←R: Remote→Host communication
The remote connected unit does not replay with the consignee s address placed at the head of the packet

1. The data s addressing administration is entrusted to the system placed upstream the WIZ2 master module: the communication administrator must place, before the data packet bytes, a byte containing the address of the WIZ2 module destination.

2. The master module transmits the redundancy + address + data packet, via radio channel.

3. The packet is received by the slave module and checked out: if considered acceptable by the WIZ2 remote module, it is then forwarded to the unit connected to it, reproducing the original data string without the front placed address.

4. If the data sent by the master were, for instance, a command foreseeing a reply from the remote unit, the returning data packet shall then be forwarded to the WIZ2 master module.

5. This unit (an industrial PC or a sensor, for instance) IS NOT BOUND to place, at the head of the data s packet, the address of the unit that originated the interrogation (the master): the slave module shall care to place its own address at the head of the data packet (*).

6. The master module, at the reception, of the data packet via radio, DOES NOT CANCEL the byte of the remote module s address, and sends downstream all the received data s bytes: this function appears to be particularly useful in polling systems ambient, where, a reply data packet, can reach the questioning unit in different delays.

This procedure is good for WIZ2 modules use in the industry, where PLCs can be instructed to operate on machines or questioned to monitor sizes. Being not equipped with the adequate logic, in order to handle the data string that composes the answer to a possible interrogation, the WIZ2 remote module places, therefore, at the head of the data from the PLC, its own address so to avoid that the so originated packet, is understood, from the other slave modules present in the system, as a command coming from the master module (the head placed address shall cause the reply rejection from all other users of the system, taking into consideration the unique addressing).

(*) this device will avoid that a reply packet coming from a slave module, is understood, by an other slave module as its own, in case the address starts with the same 1st byte of the data packet.

Note:

The destination module address is shown as a binary value.
For instance: the value 4 has to be understood as an integer number (binary configuration 00000100) and not as an ASCII type 4 which corresponds to a decimal value 52 whose binary configuration is 00110100.

It is suggest to pay attention to the conversion between decimal values and ASCII types.

The WIZ2 packing system does not utilize sequences of *escape* types, therefore it places, at the user disposal, the 256 combinations of the ASCII 8 bits coding.

Before sending a new data packet to the serial gate it is necessary to wait till the previous packet s RF transmission is over. All data entering the serial gate during the RF transmission forwarding phase, are bound to be irremediably lost.
It is therefore suggested to the User that, during the administration software writing, to pay very much attention to a correct timing of the data packets going out from the PC s serial gate, specially when the packets length is variable. Infact, if it is decided to forward a long packet followed by a short one, it is necessary to wait all the time required by the RF transmission of the long packet before sending, to the serial gate, the short one.

The packet's RF transmission time T is given by:

T = 3.6 mS + (NumByte+2) x 0.156 mS

Example #1 - Packet lenght: 1 Byte
Transmission time T= 4.1 milliseconds

Example #2 - Packet lenght: 32 Byte
Transmission time T= 8.9 milliseconds

Example #3 - Packet lenght: 96 Byte
Transmission time T= 18.9 milliseconds

It is suggested to add, precautionally, few mS to the RF transmission time between the dispatch of a packet and the subsequent one.

Technical features are subject to change without notice. AUR°EL S.p.A does not assume responsibilities for any damage caused by the device s misuse.

2.3 MASTER AND SLAVE MODULES: (SUMMARY)

	MASTER	**SLAVE**
It places the address at the head of the data packet → RF	No	Yes
It removes the address from the data packet head → RS232	No	Yes
It verifies the destination address	No	Yes
It verifies the checksum	Yes	Yes
Internal address	0	≠ 0

3. Programming

The WIZ2 modules may be programmed through the serial port, by closing the jumper 3 and supplying, subsequently, the module.

The programming mode foresees to enter a WRITE or READ type command.

Here below is reported the list of the registers, their addresses and functions.

REGISTER S NAME	**ADDRESS**	**FunCTion**	**DEFAULT**
ADDRESS	$00	WIZ2 module s address	$00
RS232	$03	RS232 protocol options: 00: no parity 01: odd parity 02: not in use 03: even parity	$00 (no parity)

3.1 WRITE

The WRITE command enters a value in a register of the microcontroller s internal EEPROM, contained in the WIZ2 module.

Format:

#_<address_register>_<value_register>

The programming string is composed by a 3 bytes sequence:

- # : ASCII type $23 which enables the microcontroller to receive a command of a register writing.
- <address_register>: is the register s address to be written
- <value_register>: is the register s new value

Note:
<address_register> and <value_register> have to be intended as binary values. For instance: the value 4 has to be understood as an integer number (binary configuration 00000100) and not as an ASCII type 4 which corresponds to a decimal value 52 whose binary configuration is 00110100.

It is suggest to pay attention to the conversion between decimal values and ASCII types.

The WIZ2 packing system does not utilize sequences of *escape* types, therefore it places, at the user disposal, the 256 combinations of the ASCII 8 bits coding.

3.2 READ

The READ command reads the register s value of the microcontroller s internal EEPROM, contained in the WIZ2 module.

Format:

$_<address_register>

The programming string is composed by a 2 bytes sequence:

- $: ASCII type $24 which enables the microcontroller to receive a command of a register reading
- <address_register>: is the address of the register to be read.

Note:
<address_register> has to be intended as a binary value. For instance: the value 4 has to be understood as an integer number (binary configuration 00000100) and not as an ASCII type 4 which corresponds to a decimal value 52 whose binary configuration is 00110100.

It is suggest to pay attention to the conversion between decimal values and ASCII types.

The WIZ2 packing system does not utilize sequences of *escape* types, therefore it places, at the user disposal, the 256 combinations of the ASCII 8 bits coding.

3.3 PROGRAMMING PROCEDURE

The module programming, takes place by closing the jumper 3 before supplying the module.

The data packet containing the programming word shall not be forwarded to the remote module.

The programming procedure takes place at the speed set by the jumpers 1 and 2, and at the correct parity. The WIZDEMOx software is pre-set at a 9600 b/s speed without parity bits.

3.4 REGISTERS

Here below is reported the list of the registers utilized for the WIZ2-434-x modules and their functions.

Technical features are subject to change without notice. AUR°EL S.p.A does not assume responsibilities for any damage caused by the device s misuse.

3.4.1 ADDRESS ($00)

The ADDRESS module contains the WIZ2 module s address.
It corresponds to the location $00 of the EEPROM memory of the microcontroller.

The values contained in the register could be:

- 0: the unity is the MASTER of the system
- 1-255: the unity is a SLAVE of the system

The value of this register, at the reception of a data packet, via RF, is compared with the first entering byte: if the two values are the same, the packet is considered effectively addressed to this WIZ2 unit.

A WIZ2 slave module places its own address before the data packet that has to be transmitted via radio.

The default value of the ADDRESS register is $00.

The register s contents can be changed by entering the command :

```
# 0 <new_value>
```

To read the register s contents, the command sequence is:

```
$ 0
```

For the programming procedures refer to para.: 3.3 - 3.4

3.4.2 RS232 ($03)

The RS232 register defines the settings of the parity of the RS232 transmission.
The allowed values are:

- 00: no parity
- 01: odd parity
- 02: not in use
- 03: even parity

It corresponds to the location $03 of the microcontroller s EEPROM memory.

In detail, the setting values correspond to those taken from the first two bits of the register according to the logic:

- RS232(0)= PARITY (0: no parity; 1: parity)
- RS232(1)= ODD (0: odd parity; 1: even parity)

The odd parity, for example, takes up the value b11 in register $03.

The default value of the RS232 register is $00 (no parity).

Technical features are subject to change without notice. AUR°EL S.p.A does not assume responsibilities for any damage caused by the device s misuse.

The register s contents may be changed by entering the command:

```
# 3 <new_value>
```

To read the register s contents the command sequence is:

```
$ 3
```

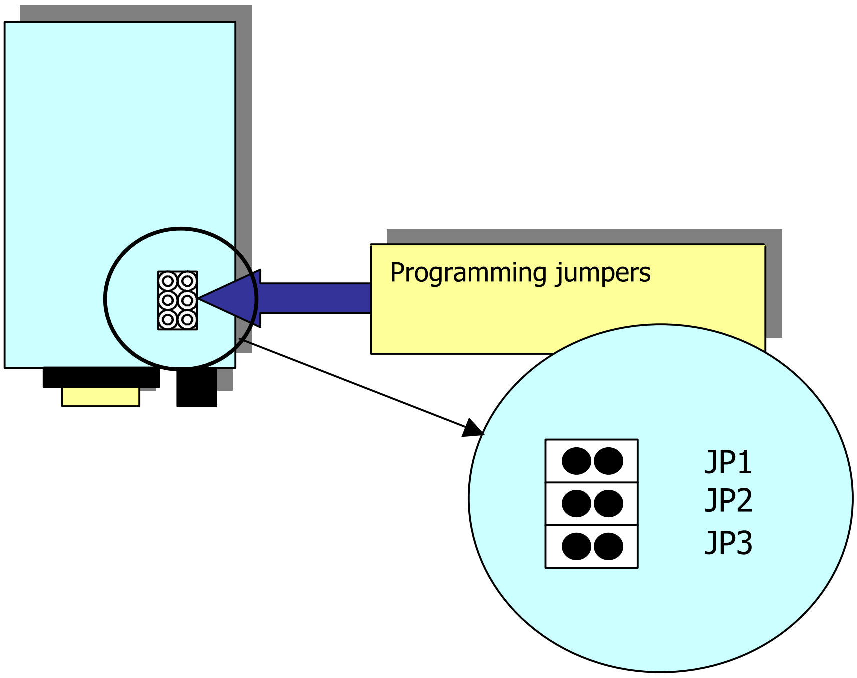
For the programming procedures please refer to para.: 3.3

Note:

It is suggested to verify, in advance, the RS232 characteristics of the unity connected to the WIZ2 module and to adequately programme such register.

4. Programming Jumpers

To reach the 3 programming jumpers carefully remove the front cover of the box (RSB model).



By closing JP3 from DATA MODE (data transreceiving) the system switches to PROGRAMMING MODE.

Data speed	**JP1**	**JP2**
9600 b/s	O	O
19200 b/s	X	O
57600 b/s	O	X
115200 b/s	X	X

X= closed jumper
O= open jumper

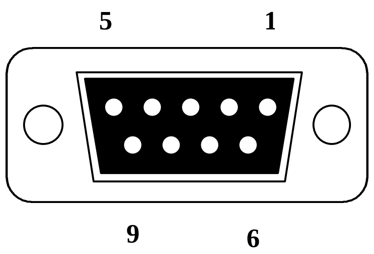
In the original configuration all the jumpers are open (DATA MODE, 9600 b/s).

APPENDIX A: CONNECTIONS

A.1 RS232 SERIAL CONNECTION

The WIZ2 modules converse with external devices through a 9 pins RS232 serial interface.
The handshaking signals between connected units, with an RS232 protocol, are not utilized by WIZ2 modules.

During the via cable programming phase, (para. 3.3) the RTS and CTS lines are used.



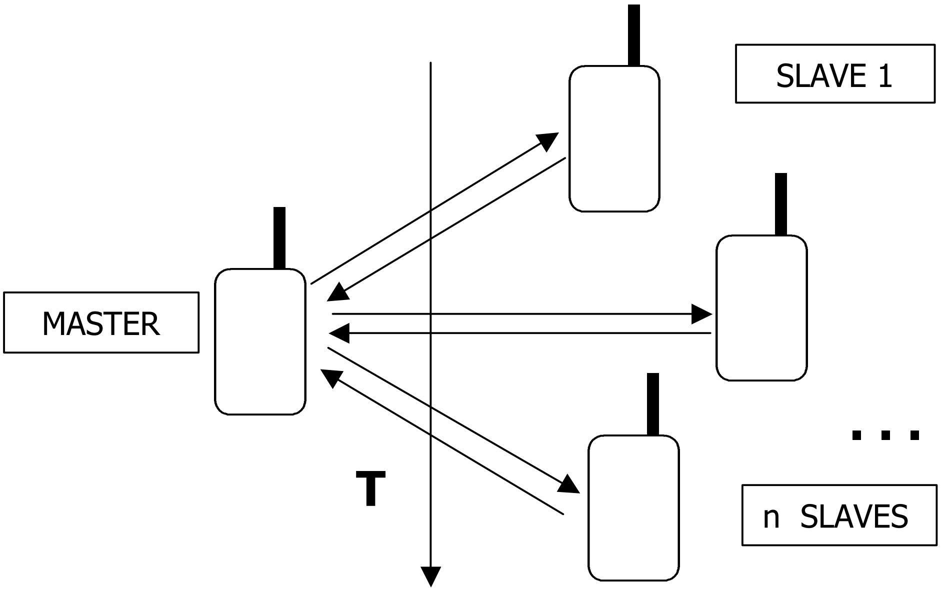
9 PINS CONNECTOR (female)

View of the connector fitted on the WIZ2 module

The connection cable with the master unit must be pin-to-pin, the one connected with the remote unit, could be null-modem (see A.2)

Pin	***Signal (PC)***		***Input/Output (WIZ2)***	***Usage in WIZ2***
1	DCD	Data carrier Detect		
2	RX	Receive Data	Output	Tx data to PC unit
3	TX	Transmit Data	Input	Rx data from PC unit
4	DTR	Data Terminal Ready		
5	GND	Ground	/	Ground
6	DSR	Data Set Ready		
7	RTS	Request To Send	Input	Not used
8	CTS	Clear To Send	Output	Not used
9	RI	Ring Indicator	Input	

APPENDIX B: MASTER AND SLAVE MODULES; THEIR USE IN A *POLLING* TYPE QUESTIONING SYSTEM



In a polling system the unity connected to the master module administrates the slaves questioning sequence, deciding the timing and the addresses of the remote units to be monitored. The master system must divide the time in slots assigned to each slave module, considering a maximum time within which, is assumed, the reply, from the remote unit, is back.

The WIZ2 modules are programmed with a software address which identifies them inside the system only.

Technical features are subject to change without notice. AUR°EL S.p.A does not assume responsibilities for any damage caused by the device s misuse.