INDUSTRIAL PROGRAMMABLE INTERFACE

MODELS iPi **UC, UET, AST**

* **Modularity : 1 UC Module can drive up to 8 inputs modules and 8 outputs modules**
* **Compacity : 11000 serie block 22,5 mm pitch**
* **Din Rail 1/3 mounting**
* **Easy to use and to program on IBM PC* with the MS-DOS* *IPILOG* software.**
* **Provide a solution for all application process**
  * **Combinatory logic treatment**
  * **Sequential logic treatment (3 x 32 steps Grafcet sequencers)**
* **24VDC power supply**
* **Software selectable Alarm output and RUN/STOP input.**
* **Power up Self-test and running watching**
* **Hardware Watch Dog**
* **Slave Modbus*/Jbus* communication, baud rate, parity and slave Number software selectable**

The industrial Programmable interface, **iPi**, is an interface designed to command or control systems using simple combinatory logic or sequential logic functions. It can be easily programmed , on **IBM PC*** with a **MS-DOS*** software, called ***iPiLOG.***

The application debug can be made step by step with the **iPi** connected. The ***iPiLOG*** software allows to show and modify the internal variables.

In "*RUN*" mode, the application software automatically starts at **iPi** power-up, without using the PC*.

The **iPi** can be connected to a Modbus*/Jbus* master equipment. That connexion allows to take information of the remote process running into the **iPi** to centralize to a master (PLC or PC supervisor). By the addition of a serial interface (**ILPH SERIES**) to transform the RS-232 of the programming port into RS-485 or current loop, the **iPi** can be put on a Modbus*/Jbus* network with other slave equipments.

iPi APPLICATION RANGE

The **iPi** can be used, for example, in such a process :

- machine control,
- process control,
- decentralized process,
- remote I/O
- alarm - security,
- direct replacement of relay logic,
- etc. ...

V. Modbus*/Jbus* FUNCTIONNEMENT

V.1. GENERALITIES

The (RS-232C) iPi programming port allows the iPi to be accessed with the **Modbus*/Jbus*** communication protocol. This important feature makes the iPi able to communicate with others devices on a network, it also allows the iPi to collect some data for a Modbus*/Jbus* master. The iPi is only a slave device.

The iPi can be placed on a multi-point network but then needs the addition of a serial interface (ILPH) which translates the signal from *RS-232C* to *RS-485* (1 or 2 pairs) or in *Current loop*..The ENTRELEC interface range can be used to transform the RS-232C into other physical format:

- *RS-232C ->Current Loop* (Réf. : **84202-23**)
- *RS-232C -> RS-485 isolated or not* (Réf. : **84231-17** ou **84233-11**).

For a Modbus*/Jbus* master, the iPi is like a dual port RAM reflecting values of the different variables of the process (inputs, outputs, timer, counter, etc ...). These pseudo DPTR can be accessed in write or read mode. The iPi refresh variables that the Modbus*/Jbus* master can then use to remote or survey the process.

The different communication parameters (baud rate, parity or slave number, etc ...) can be set by using the *iPILOG* software.

The Modbus*/Jbus* protocol allows to a master to communicate with 255 slaves (logical addresses) on a same network. Only the master initiates the transaction. Transactions are either a query/response type (only a single slave is addressed) or a broadcast/no response type (all slaves are addressed). A broadcast message is indicated by a null address. A transaction comprises a single query and single response frame or a single broadcast frame.

To be more reliable, the communication protocol includes a security device in the frame. A control code **CRC 16 (Cyclical Redundancy Check)** is added in the end of the frame and permits to detect an error of transmission.

When a slave receives a message, it then calculates the CRC 16 of the received message and compares to the one contained in the transmitted message. In case of difference, the slave device simply discards the frame, sends no response and wait for a new query of the master. The master which does not receive a response, considers that the previous query frame was bad and sends again the query. If the comparison between the two CRC 16 are good but a data error occurs or an illegal register address, the slave indicates this in its response by a) setting the high bit of the function code to 1, b) and writing the appropriate error code in the next byte. The slave discards the message, and does not act on it in any way.

Error codes are :

1 - **Function code Unknown** - sent whenever the requesting message contains a function code not supported by the iPi, for example a function code different from 3, 4, 6 or 16 (10h), etc ... (see the concerned chapter).

2 - **Invalid Register Address** - sent whenever the requesting message contains a register address unknown by the iPi.

3 - **Data incorrect** - sent whenever the iPi is unable to process the message due to an error in the data field. For example : writing a timer value greater than 255.

The **Modbus*** protocol **RTU mode** (binary) from **GOULD MODICON** is a master/slave protocol (only 1 master on the network in multi-point mode).

The **Jbus*** protocol from **APRIL** is compatible, for the iPi, with the **Modbus*** protocol : the frame structure is the same , the function codes are identical (Read N words = 3, Write N words = 10, Write 1 word = 6).

NOTA : Addresses used by **Modbus*** coupler are incremented by 1 regarding the one actually transmitted on the serial link and understood by the **IPi**.

Ex : address used in a program if **Modbus*** coupler : 0002
address actually transmitted into the frame on the line : 0001
address used in a program if **Jbus*** coupler : 0001
address actually transmitted into the frame on the line : 0001

The general frame Format is :

Address	Function code	Data	CRC 16
8-bits	8-bits	N x 8-bits	16-bits

Frame synchronization can be maintened in RTU transmission mode only by simulating a synchronous message. The receiving device monitors the elapsed time between receipt of characters. If three and a half characters time elapse without a new character or completion of the frame, then the device flushes the frame and assumes that the next byte received will be an address.

T1 T2 T3	Address	Function Code	Data	CRC 16	T1 T2 T3
	8-bits	8-bits	N x 8-bits	16-bits	

Note : All the numbers are in hexadecimal mode.

Address : The address field immediately follows the beginning of frame and consists of 8-bits. These bits indicate the user assigned address of the slave device that is to receive the message sent by the attached master. Each slave must be assigned an unique address and only the addressed slave will respond to a query that contains its address. When the slave sends a response, the slave address informs the master which slave is communicating. In a broadcast message, an address of 0 is used. All slaves interpret this as an instruction to read and take action on the message, but not to issue a response message. This address can be set by the ***iPiLOG*** .software.

Function code : The function code field tells the addressed slaves what function to perform. The high order bit in this field is set ON by the slave device to indicate that other than a normal response (that is an exception response) is being transmitted to the master. This bits is 0 for a normal query or response message.

Data : The data field contains information needed by the slave to perform the specific function or it contains data collected by the slave in response to a query.

CRC 16 : This field allows the master and slave devices to check messages for errors in transmission.

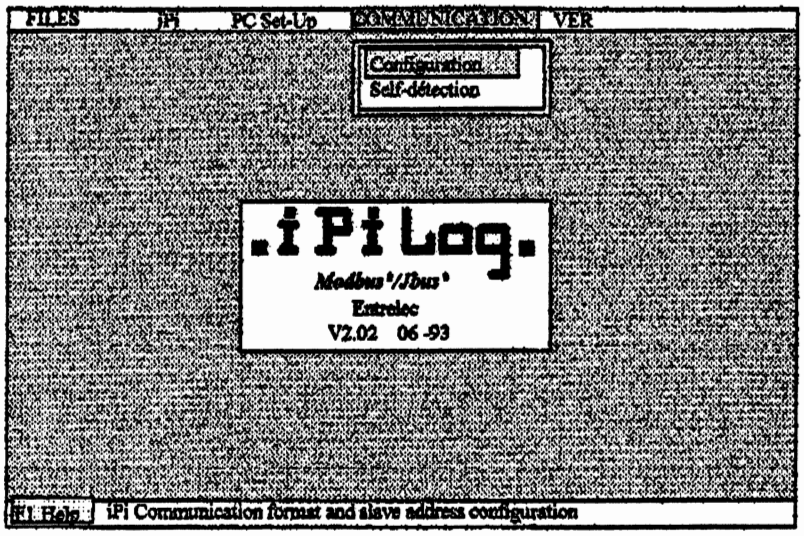
software.

iPILOG detects the communication parameters in that manner :

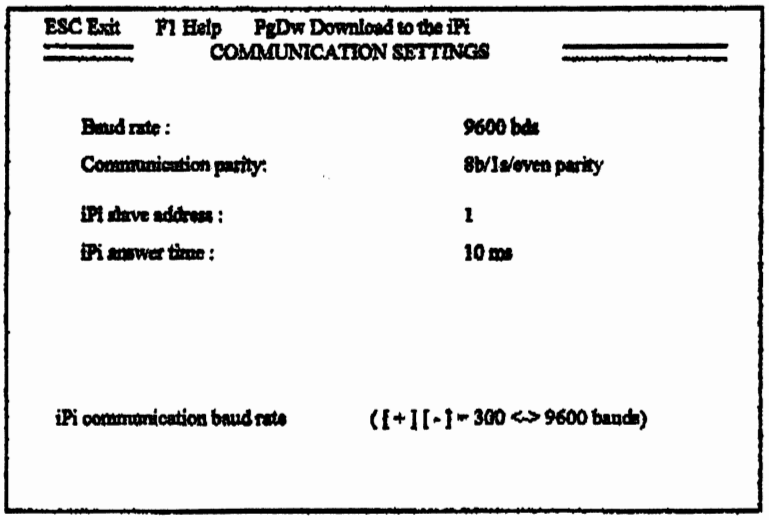
it sends a specific identification request and tests the iPi response with all the supported baud rate or transmission format. When the iPi answers, *iPILOG* can upload the transmission format and the address of the attached iPi. The setting of the parameters or the programmation can then be made.

How can we configure the iPi communication parameters ?

In the main menu, move with the arrows to the **COMMUNICATION** menu, then validate with the *ENTER* key. A new window appears which allows to configure or to detect the iPi parameters of the attached device.



To configure the iPi, move to the **Configuration** menu and validate with *ENTER*. *iPILOG* displays a new screen to set the parameters .



Each parameter can be easily modified by moving on the desired field with the arrows keys and using the + and - keys. By pressing these keys some choices are available :

Baud rate : from 300 to 9600 bauds

Transmission format : 1 Start bit, 8 data bits and 1 Stop bit not modifiable.

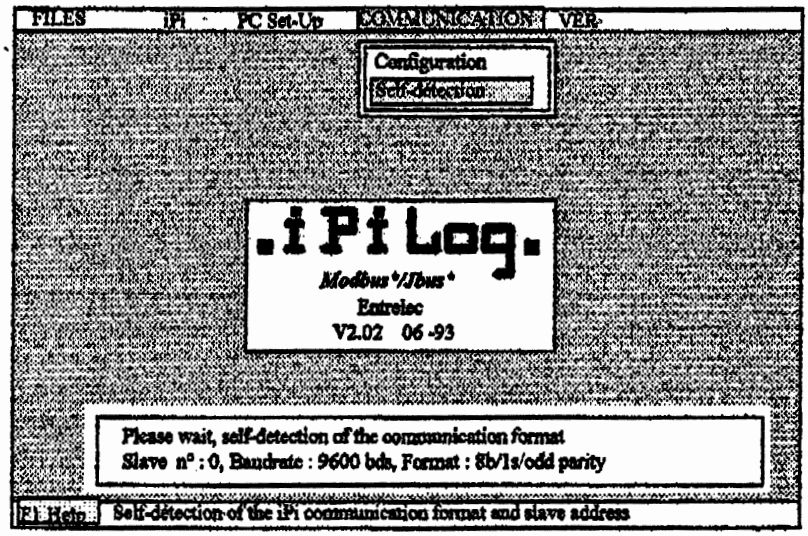
The parity can be set to : no parity, even or odd parity.

The iPi slave number : from 1 to 255

The iPi answer time is the delay between the receipt of a Modbus*/Jbus frame, its process and the answer sent to the master. This delay time can be set from 0 to 155 ms.

When the communication parameters have been set, pressing on the ***Page Down*** key download them to the iPi. These parameters will be effective at the next iPi power-up.

By choosing the **Self-détection** menu, ***iPiLOG*** tests all the baud rate and transmission format available whenever it can communicate with the iPi. It then opens a window which indicates the founded parameters. When it arrives to communicate, it uploads the baud rate, transmission format and the slave number of the attached iPi.



V.3. Modbus*/Jbus* FUNCTIONS SUPPORTED

The iPi accepts the followings Modbus*/Jbus* functions :

Function 1 and 2 :	Read N bits and N Inputs bits
Function 3 and4 :	Read N words or read N inputs words
Function 15 :	Write n bits
Function 16 :	Write n words
Function 8 :	Event counter read

entrelec

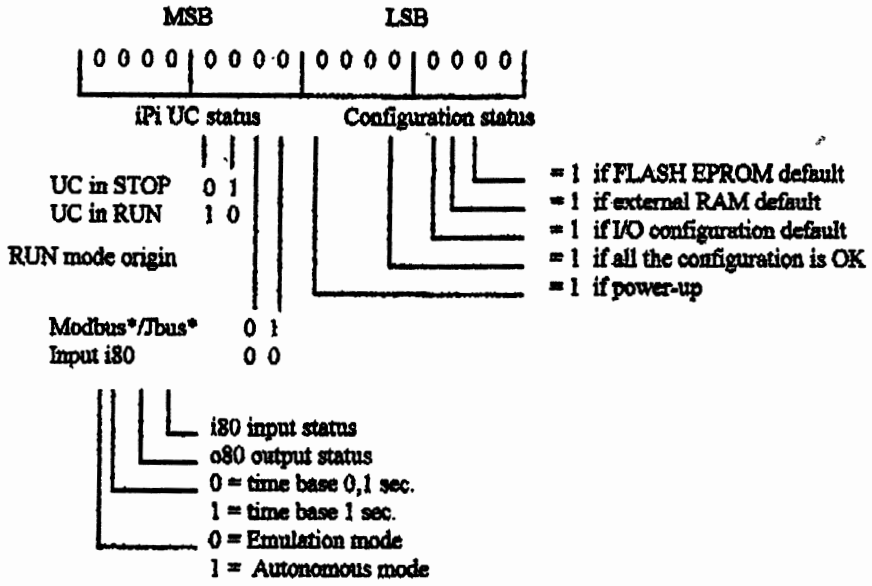
MARKETING Department

V.4. iPi MEMORY MAPPING

The iPi memory mapping is a following of logical addresses. This memory mapping is composed by 16 bits memory words and can be accessed either in bit or words mode.

Status Registre :

Access		Logical address		Comments
Read	Write			
X	X	0000h	00000d	iPi Status : reflecting the iPi environnement status.



iPi Inputs/Outputs status word

The memory inputs words can only be accessed in read mode. The memory Outputs words can be accessed in read or write mode. Only the LSB bytes of the memory inputs or outputs words are significant. An Output write with a value greater than FFh (255d) will be denied by the iPi which will return a Modbus*/Jbus* error code "Data incorrect".

All the memory mapping can be accessed either in word or bit mode.

Exemple :

i80 UC input access : @ bit = 16
i05 input of the module 0 : @ bit = 37

Access		Logical address		Comments
Read	Write			
X	-	0001h	00001d	Inputs of the iPi-UC module: i80 à i85
X	-	0002h	00002d	Inputs of the *module 0* iPi-8ET: i00 à i07
...	...	...	...	...
X	-	0009h	00009d	Inputs of the *module 7* iPi-8ET : i70 à i77
X	X	000Ah	00010d	Outputs of the module iPi-UC : o80 à o83
X	X	000Bh	00011d	Outputs of the *module 0* iPi-8ST : o00 à o07
...	...	...	...	...
X	X	0012h	00018d	Outputs of the *module 7* iPi-8ST : o70 à o77

WARNING : An attempt to read a memory word reflecting an input or output extension module not present will be rejected by the iPi which will send an error code "address incorrecte".

Version 1.1 07.92

SERIAL LINK INTERFACE

USER GUIDE

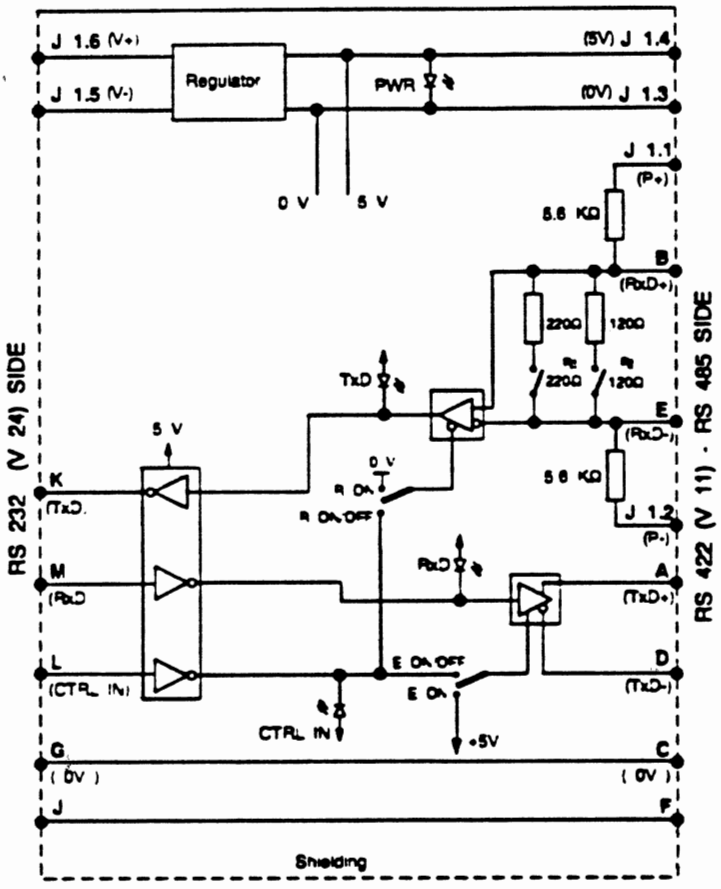
MODEL ILPH RS 232 / RS 422 - RS 485 NON ISOLATED
P/N 84231-17

Réf. 100.208.610

1. GENERAL

Interface between an RS 232 serial link and an RS 422 ou RS 485 serial link. Extend transmission beyond the 15 m limit of the RS 232 serial link, to cross "noisy" environments, to perform multipointing (network) etc.

2. SCHEMATIC DIAGRAM



3. TECHNICAL SPECIFICATIONS

3.1 POWER SUPPLY

- Power supply voltage : 8,5 V to 28 VDC
- Protection : polarity inversion
- Power requirement : less than 100 mA
- 1 yellow "power on" Led.
- Screw-type plug-in connector.

3.2 RS 232 LINK

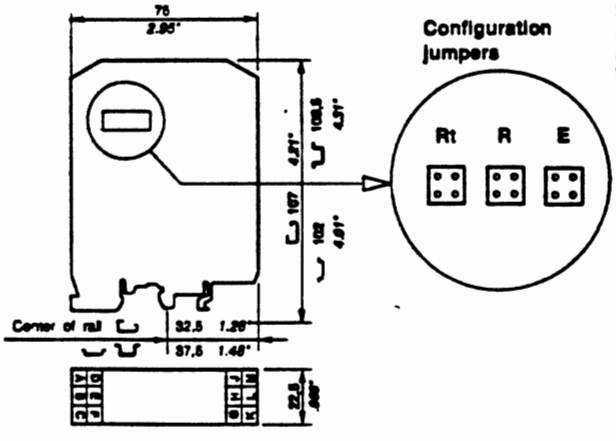
- EIA RS 232 C / CCITT V24 / V28
- I / O protection : over voltage
- Speed / Distance : 38400 Bauds / 15 m
- 3 green Leds (RxD, TxD, CTRL IN)
- Clamp connection.

3.3 RS 422 - RS 485 LINK

- EIA RS 485 and compatible EIA RS 422 / CCITT V11
- I / O protection : over voltage
- Speed / Distance : 38400 Bauds / 1200m
- Transmitter can communicate with up to 32 receivers simultaneously.
- Clamp connection.

3.4 PHYSICAL CHARACTERISTICS

- Black box series 11000 ENTRELEC, snaps onto DIN rail
- Temperature :
  - operation : 0 to 50° C
  - storage : - 20 to 70° C



entrelec

ENTRELEC · 184-186, Rue Léon Blum - 69627 VILLEURBANNE Cedex - France - Tél: 72 35 35 35 - Fax 72 33 79 54

4. CONFIGURATION

4.1 LINE AMPLIFIER CONFIGURATION

Configuration of amplifiers of the RS 422 - RS 485 (Receiver, Transmitter) line provides greater flexibility of use. The various configurations can be selected using the 2 jumpers (R, E) located inside the box.

4.1.1 RS 485 LINK ON ONE PAIR

R  R ON/OFF Jumper R in position R ON/OFF

E  E ON/OFF Jumper E in position E ON/OFF

The Receiver and the Transmitter are activated alternately (never at the same time) depending on the status of the CTRL IN signal.

CTRL IN STATUS	ACTION ON RS 485
0 logic (+3V ≤ U ≤ +25V)	Transmitter active / Receiver inactive
1 logic (-25V ≤ U ≤ -3V)	Transmitter inactive / Receiver active
High impedance	Transmitter inactive / Receiver active

NOTE : For RS 232 products runnig the RTS (REQUEST TO SEND) signal, connect RTS to CTRL IN.
Otherwise, connect M (RxD ILPH) to L (CTRL IN).

4.1.2 RS 485 LINK ON 2 PAIRS

R  R ON Jumper R in position R ON

E  E ON/OFF Jumper E in position E ON/OFF

Receiver permanently active
Transmitter controlled by the signal CTRL IN (see table 4.1.1 for Transmitter operation as a function of CTRL IN)

4.1.3 RS 422 LINK ON TWO PAIRS

R  R ON Jumper R in position R ON

E  E ON Jumper E in position E ON

The Transmitter and Receiver are both permanently active

4.2 POLARIZATION OF THE RS 422 - RS 485 LINE

The line must always be polarized. The ILPH is used to polarize the reception channel

Connection by 1 wire P+ (J1.1) with 5V (J1.4)
Connection by 1 wire P- (J1.2) with 0V (J1.3)

4.3 ADAPTING THE RS 422 - RS 485 LINE

The line must always be adapted to the level of the reception channel of each subscriber forming the end of the bus

The ILPH is used to adapt the reception channel by setting the jumper Rt correctly

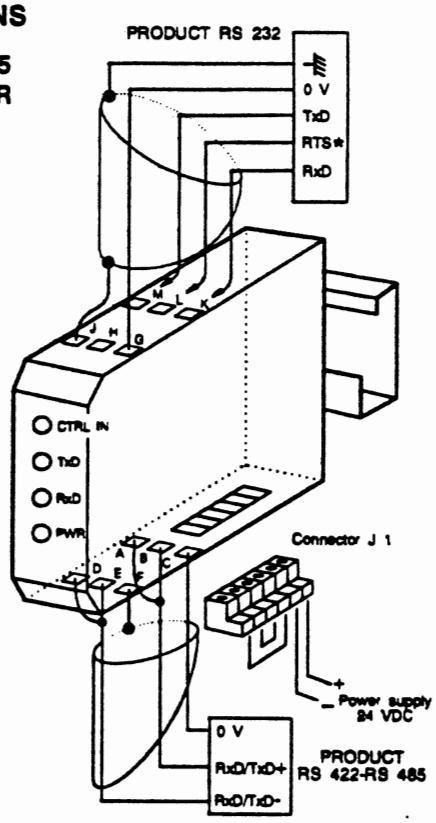
Rt  * Line adaptation Rt = 120 Ω (general case)

Rt  * Line adaptation, Rt = 220 Ω

Rt  * No line adaptation Rt = ∞

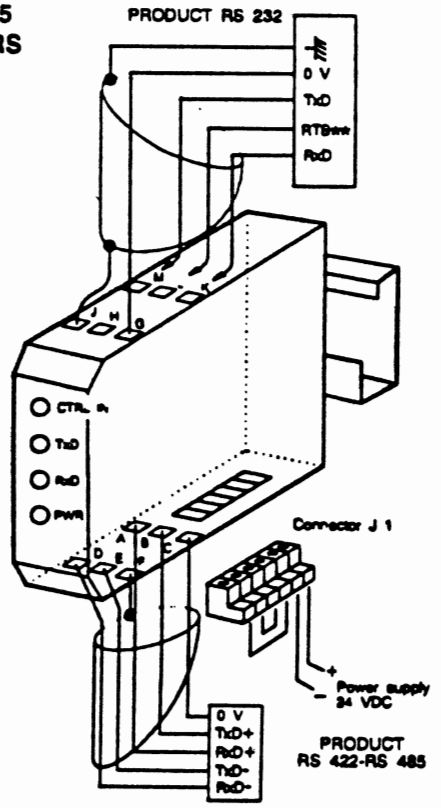
5. CONNECTIONS

RS 422 - RS 485 LINK ON 1 PAIR



* NOTE : If the RTS signal is not generated, connect M (RxD ILPH) to L (CTRL IN).

RS 422 - RS 485 LINK ON 2 PAIRS



** NOTE : Only to be connected for RS 485 two pairs (of no use for RS 422 two pairs). If the RTS signal is not generated, connect M (RxD ILPH) to L (CTRL IN).

entrelec