ST00615

6 channels

USER MANUAL

Internal version
rev. 0.2

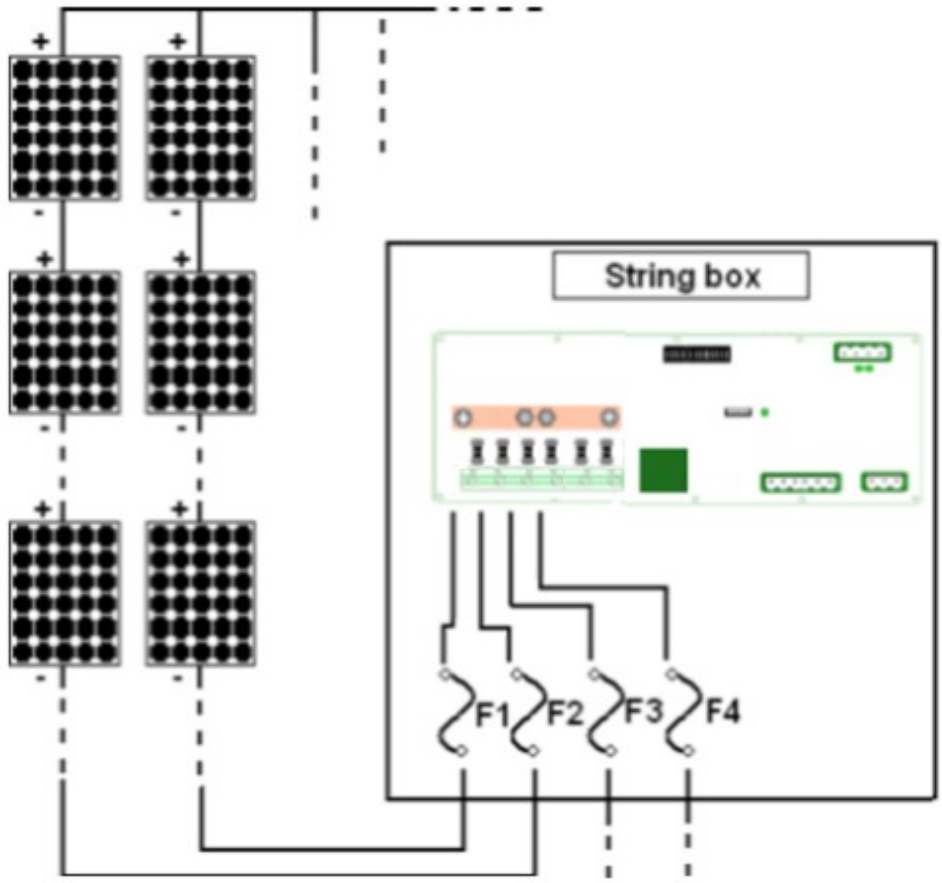
August 2017

Index

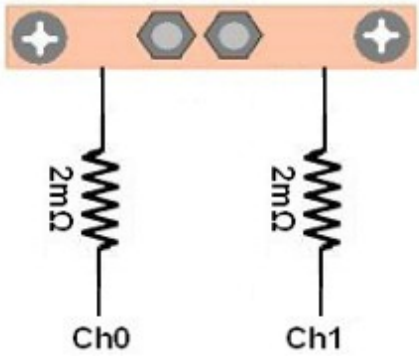
1 GENERAL NOTES

1.1 Introduction

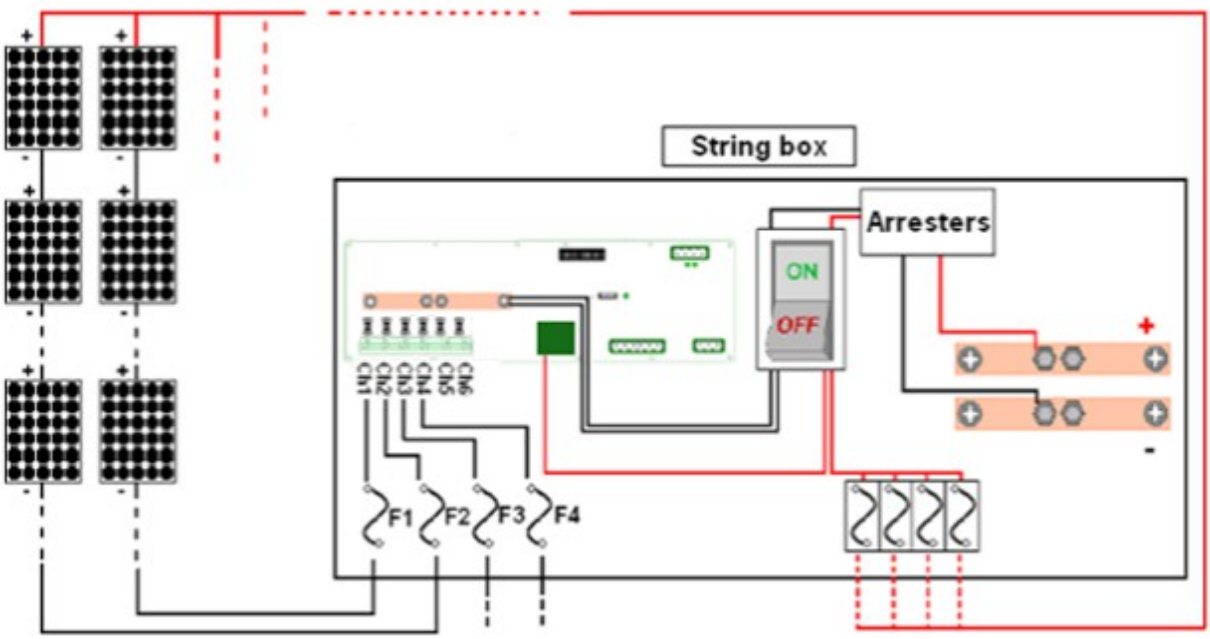
The ST0 module to string control, allow to monitoring current and voltage generated by photovoltaic panels strings. For example is possible connect each other 32 PV panels at 36V to each string channel, with positive pole connected each other. The negative pole of each string is brought to the dedicated input, like in the following picture:



After the strings input connector, on the ST0 board there is a resistor, it is necessary to detect the current follow:



and finally a copper bar connect all the negative poles, thus creating a common 0V.
The ST0 board also provides two digital inputs and an on-board sensor which allow to measure the temperature.
The digital inputs allows to detect the arrester state and the power disconnector switch state.
Is possible communicate with the ST0 board through an RS485 serial port. Using Modbus RTU protocol, or with Kernel Sistemi protocol, is possible monitoring all the physical quantities measured (temperature, currents, voltage). Moreover is possible keep monitored the fuses status on the string box, through the reading of an internal register on memory map (DATA30034).



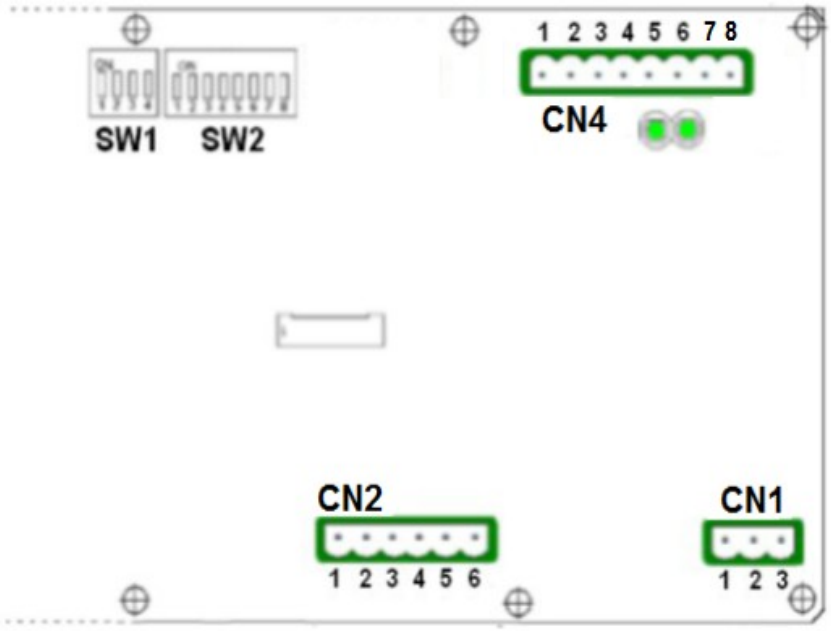
In the following image there are the “ST0 string controller” with all the wiring. Obviously isn't necessary connect all the specified devices, they are indicated to give a connection general idea.

2 HARDWARE CHARACTERISTICS

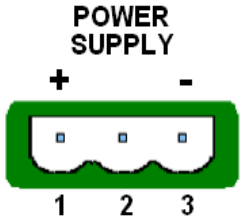
Microprocessor	STM32F303
Power supply	24Vdc
Power consumption (W)	< 3W
Maximum number of monitored strings	6
Maximum common voltage	1500V with precision better than 1,5%.
Range of measurement	0...90A
Current reading accuracy	Better than 0,15%
Current reading precision	Typical 0,5%
Communication	Modbus RS485 / RS487
Digital Inputs	Two digital inputs 24Vdc PNP
Analog inputs	one on board temperature sensor to know the temperature inside the string box panel
Working temperature's range	From -20 to +80 °C
Temperature's drift 0°C ÷ 70°C	Better than 50mA a 12,5A
Working atmosphere	Without corrosive gas
ID Address	Defined by dip-switches
Size	-

N°	Type of resources
1	Sensor on board to read the temperature (precision better than 1,5%).
1	RS485 serial port. This serial port is used to connect many “ST0 string controllers” into a network or to a PC. Is possible select the communication characteristics with a dip-switchs on board (node address, baud rate, parity, and communication protocol, that may be Modbus RTU or Kernel). This COM is divided in two connectors in order to facilitate the wiring.
1	PT100 inputs (from 0 to 300 °C) to temperature reading, with accuracy better than 1,5%.
2	PNP digital inputs 24Vdc, typically used to arrester connection, switches or other devices.
6	This board can manage the current reading of 6 strings until 15A with typical precision of 0,5%. and a temperature between -20 and +80 °C

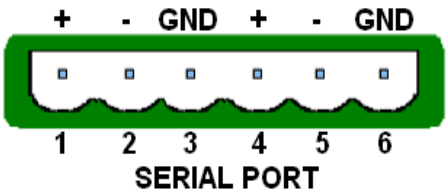
Cpu



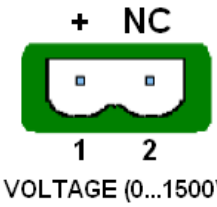
2.1 CN1



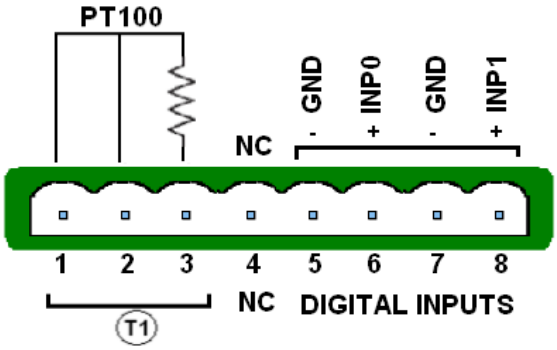
2.2 CN2



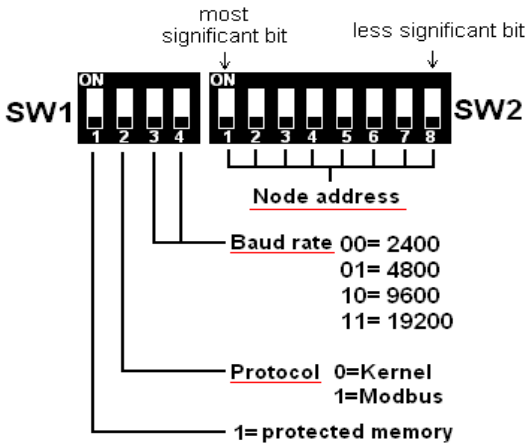
2.3 CN3



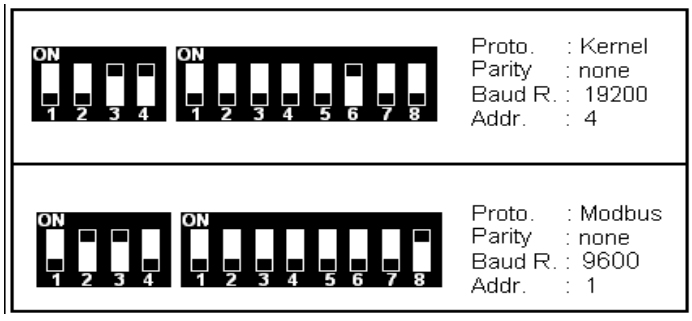
2.4 CN4



2.5 Dip-switches



Some dip-swiches examples:



2.6 Fixing system of the the naked board (without supporting box)

To fix the naked board (without case) is necessary use plastic spacers with dual clutch. The plastic spacers must be 4x20mm or 4x25mm (4mm is the hole diameter on the board). Look the below picture.

2.7 Status led

On the board there is a status led which with its blinking show the board status. There are two possible different blinking ways: blinking each 0,5 sec, or blinking faster. If the blinking is 0,5 sec ON and 0,5 sec OFF, it means the the board is ready to communicate with an external device, instead if the blinking is faster than 0,5 sec, it means that the board is in test mode with all the dip-switch OFF. In this way the board isn't ready to communicate with an external device.

2.8 RS485 communication cable

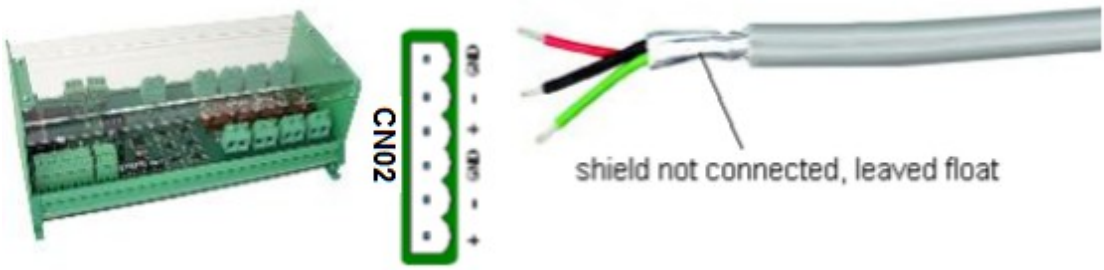
Everything about the RS485 connection, must meet certain features:

Maximum cable length
it must be no longer than 1,2Km (it means the entire line length, and not the conection between two nodes)

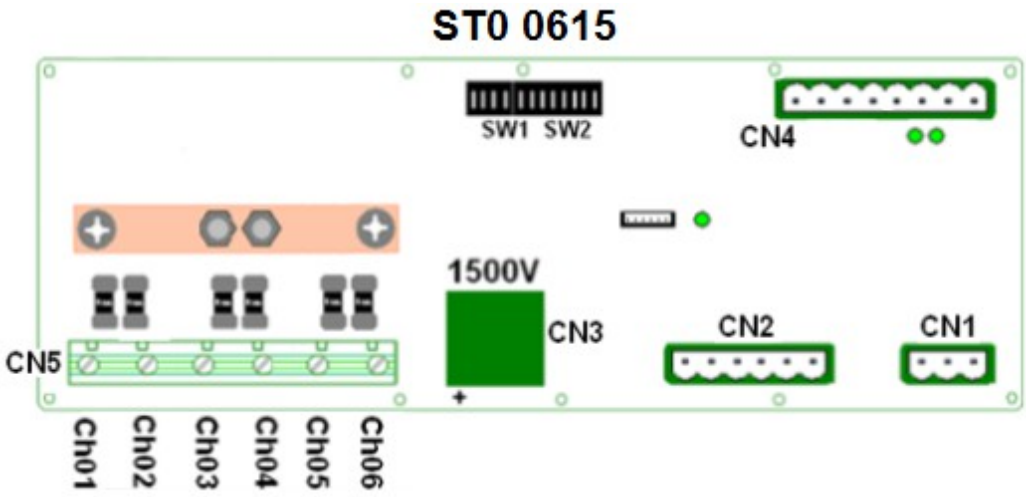
Maximum number of slaves
it's possible connect up to a maximum of one hundred slaves

Technical characteristics of the cable to use
It must be a three-wire cable 3 x 0.75mm

How to do the RS485 connection
The RS485 connection must be a three wires connection (TX+, TX- and GND) with a shielded cable. The cable shield must be leaved float, it means that the shield must be not connected neither one side nor the other one.



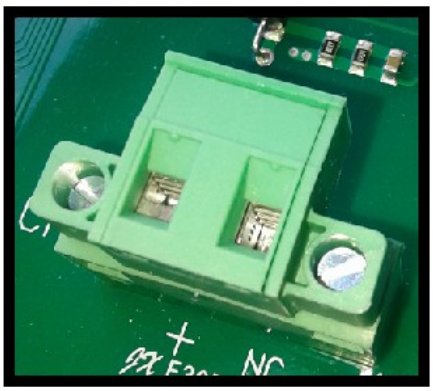
2.9 ST0 0615



2.10 Informations about wires and connectors

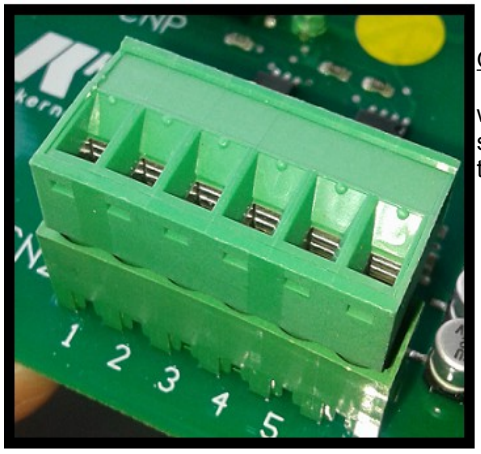
CHANNEL CONNECTOR

wire range:.....until 6 mm²
stripping lenght:.....6 – 7 mm
torque.....0,85 N/m



VOLTAGE CONNECTOR

wire range:.....AWG 12 – 14 ≈ 0,5 - 2 mm²
stripping lenght:.....7 – 8 mm
torque.....5 Lb - In



OTHER CONNECTORS (power supply, digital inputs etc...)

wire range:.....2,5 mm²/0,34 - 2,5 mm²
stripping lenght:.....7 – 8 mm
torque.....5 Lb - In

COPPER BAR BOLTS

torque.....2,2/2,5 N/m

3 MEMORY MAP

The ST1 has the following memory map, it's made of 16 bits locations (1word) called “DATA”. Because each DATA is composed by 16 bits, its maximum value will be 65535.

DATA.30001	Inputs
DATA.30002	Inst Curr Str_01 *(mA [0...30000])*
DATA.30003	Inst Curr Str_02 *(mA [0...30000])*
DATA.30004	Inst Curr Str_03 *(mA [0...30000])*
DATA.30005	Inst Curr Str_04 *(mA [0...30000])*
DATA.30006	Inst Curr Str_05 *(mA [0...30000])*
DATA.30007	Inst Curr Str_06 *(mA [0...30000])*

...

DATA.30034	Fuse status (Ch1...Ch6)

...

DATA.30040	Inst V_1 *(V [0...1500])*

...

DATA.30045	Inst T_2 *(°C [-20...+120]) – on board*

...

DATA.30047	Sum of all currents (A)
DATA.30048	Power (W) - MSW
DATA.30049	Power (W) - LSW

...

DATA.30052	RMS Curr Str_01 (average value on last 6 seconds)
DATA.30053	RMS Curr Str_02 (average value on last 6 seconds)
DATA.30054	RMS Curr Str_03 (average value on last 6 seconds)
DATA.30055	RMS Curr Str_04 (average value on last 6 seconds)
DATA.30056	RMS Curr Str_05 (average value on last 6 seconds)
DATA.30057	RMS Curr Str_06 (average value on last 6 seconds)

...

DATA.40001	Set up PARITY mode: 1: none 2: even 3: odd
DATA.40002	Offset Curr Str_01
DATA.40003	Offset Curr Str_02
DATA.40004	Offset Curr Str_03
DATA.40005	Offset Curr Str_04
DATA.40006	Offset Curr Str_05
DATA.40007	Offset Curr Str_06

...

DATA.40040	Offset V_1

...

DATA.40043	Offset Aux_2

...

DATA.40045	Offset T_on board

…

DATA.40052	Gain Curr Str_1
DATA.40053	Gain Curr Str_2
DATA.40054	Gain Curr Str_3
DATA.40055	Gain Curr Str_4
DATA.40056	Gain Curr Str_5
DATA.40057	Gain Curr Str_6

…

DATA.40090	Gain V_1

…

DATA.40093	Gain Aux_2

…

DATA.40095	Gain T_on board

Notes:
Each “offset DATA” has 0 as default value. Each “gain DATA” has 1000 as default value. *The value 1000 means **x1***, in this way, for example, is possible write 500 and make the value **x0,5.**

3.1 Memory map description

DATA.30001: the first two bits of these register are the mirror status of the four digital inputs on the board (INP0, INP1 on CN4). So if DATA.30001 = 0000000000000011 [bin] = 3 [dec], it means that all the two digital inputs are ON.

DATA.30002...DATA.30007: these registers contains the current value of the current reading on each channel. It is in mA

DATA.30034: the sixteen bits of DATA.30034 show if each channel current reading is under 200mA or not. This threshold represent the fuse status.

DATA.30040...DATA.30049: these registers show the reading temperature (T2), the voltage reading (on connector CN3) etc...

DATA.30052, DATA.30059: these registers contains the average value on last 6 seconds of the current reading. Obviously these values are more stable than the instantaneous values show in registers DATA.30002...DATA.30025

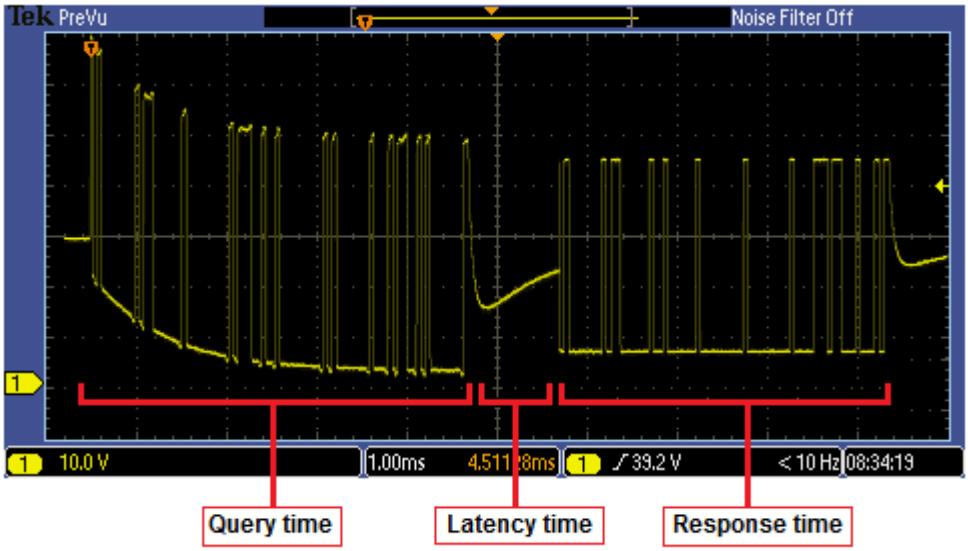
DATA.40001: through this register is possible set the communication parity. The default value is zero, so “no parity”

DATA.40002, DATA.4009: these are the offset registers. These registers (whose default value is 0) allow to add a constant value to the current reading. This allow to adjust a possible reading error. For example if DATA.30002 show 2300 (it means that channel CH1 read 2,3A), writing DATA.40002 = 200 the new value of the reading will be DATA.30002 = 2500 (it means that channel CH1 read 2,5A).

DATA.40052, DATA.40059: these are the gain registers. These registers (whose default value is 1000) allow to multiply a constant value to the current reading. This allow to adjust a possible reading error. For example if DATA.30002 show 2300 (it means that channel CH1 read 2,3A), writing DATA.40052 = 1500 the new value of the reading will be DATA.30002 = 3450 (it means that channel CH1 read 3,45A, 2300 x 1,5 =3450).

3.2 Reading speed

The analogic values of the currents, the voltage and the temperature are read simultaneously 10 times per second (100 msec scan time), then are inserted in it's own fifo (a fifo for each analogic value), 16 values deep. The value read from the board is the mobile mean of the fifo, so it is the mean of the last 16 raad values (1.6 sec), updated every 100 msec. This is done to make the analogic readout more stable and it is a good compromise between speed and readout stability.
The istant values of the analogic are temporary stored into a hidden memory area, not accesible to the COM port.
The update time depend on the speed polling time of the SCADA and the commnunication baud rate.
The total time requested to read the board via com port is splitted into three times: *the query time*, *the latency time* and *the response time*.
The query time is the time needed by the SCADA to send the MODBUS query packet and depends on the baud rate (about 4 msec at 19200 baud rate).
The latency time is the time need by the board to process the query and prepare the answer, it is between 1 and 2 msec and it is independent on the baud rate.
The response time is the time needed byte the boad to send the MODBUS answer packet, it's depend on the baud rate and on the number of registers read at a time, for a single register read at 19200 baud it is about 4 msec.
So at 19200 baud rate the total time needed to read a single register is about 10 msec., you have to add 1 msec every other register read, for example to read 16 registers with a single query will take 10 msec + 15 * 1 msec = 25 msec.



4 OREDER CODES

Here below the order codes:

CODE	DESCRIPTION	PICTURE
ST0 0615\NC	Device with support for din rail bar	
ST0 0615\NK	Device without support for din rail bar