BY11800-COUNT

2-channel pulse counter with RS485 Modbus interface

(BY11800_M2_V1)

Via Como, 55 - 21050 Cairate (VA) - Tel. 0331/312523 - Fax. 0331/313077 - WEB: www.bytronic.it - e-mail: bytronic@bytronic.it

SUMMARY

Via Como, 55 - 21050 Cairate (VA) - Tel. 0331/312523 - Fax. 0331/313077 - WEB: www.bytronic.it - e-mail: bytronic@bytronic.it

1 REFERENCES

This manual is in the following updated state:

- File name:…………………………… UG_B_11800COUNT_E.docx
- Revision:………………………………… 00
- Date:………………………………………… 16.01.2019

The content relates to firmware version 02.01.00 and later.

THE IMAGES CONTAINED IN THIS DOCUMENTATION ARE FOR INDICATIVE PURPOSE AND COULD BE DIFFERENT FROM THE REAL ONES.

2 General

The BY11800-COUNT is a module for the metering of impulses coming from meters (electric, gas, water, piece counters, etc.) equipped with pulse repetition outputs with potential-free contacts (mechanical or electronic).

The pulses are counted on two independent inputs (C1 and C2) to which as many counting registers, both direct (TOTALIZERS) and 'weighed' (ACCUMULATORS) with a sophisticated programmable digit calculation system. Each direct count value can reach one billion billion pulses (1018) and then reset to zero and start over from scratch.

The count values are saved in permanent memory in real time, so that they are not lost in the event of a power failure.

On the front of the module there are 4 signaling LEDs, the meaning of which will be explained later.

The BY11800-COUNT can be interrogated and programmed via the optoisolated RS485 serial port at 3kV, with Modbus RTU or ASCII Modbus protocol.

There are various parameters and programmable commands to set correct operation.

It has the possibility of being powered with both ac and dc voltage, with a voltage between 80 and 265 Vac (100 - 380 Vdc).

A network of 32 instruments connected to each other and interrogated by a Master system (SCADA, PC etc…) can be created.

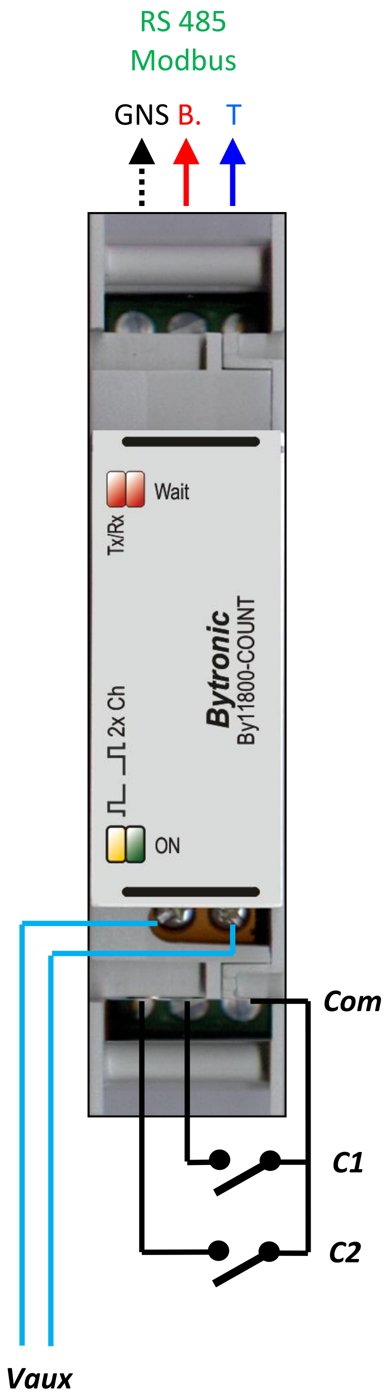
The setting of the node number and the speed of the serial port is done via Modbus, starting from the node number 1 and the speed of 115.200bps which are the factory settings.

The BY11800-COUNT is built in a container for DIN 46277 (EN 50022) 1 module bar.

More details in the following sections.

3 Connections

3.1 General diagram

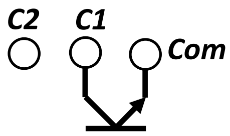


3.2 Checks and notes

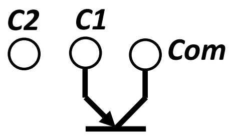
The terminal blocks accept cables with a maximum section of 1.5mmq. The maximum stripping length of the cable is 5mm and the maximum tightening torque is 0.5Nm

3.2.1 Connections to inputs C1 and C2

The contacts to be connected to the BY11800-COUNT, whether electronic or mechanical, **they must be potential-free**.
In the case of voltage or current signals, they must be adequately separated (relays, optocouplers, reed contacts, etc.).
The inputs C1 and C2 have internal pull-up resistances at + 5V with respect to the common pole Com.
The NPN contact connection (example C1) must be performed as follows:



The PNP contact connection (example C1) must be performed as follows:



The contacts can be mixed, i.e. an electronic NPN and / or PNP and / or a mechanical one.

3.2.2 RS485 serial port connections

The serial interface is totally isolated (at 3kV) from the inputs and from the auxiliary power supply.
Connections A and B must be connected respectively to the corresponding ones of the pre-existing network or to the control Master device.

The 'GNS' terminal is to be used with shielded cable, but must not be connected to earth or other voltage references. The screens of several modules can be interconnected.

For significant distances (a few hundred meters), depending on the level of disturbance in the environment, the connection must be made with quality twisted and / or shielded cables.
Remember that at the last instrument of the RS485 network, the one furthest from the Master, a termination resistance must be applied between A and B (only one for the whole network), with a typical value of 120 Ohm.

If the network is made up of BY11800-COUNT instruments only, up to 32 can be connected. There are also solutions for larger networks. If so, contact Bytronic in advance.

Before connecting the instrument to the network, configure its address locally and always check that each instrument on the network has a unique address. If there are several instruments with the same address, they will respond simultaneously corrupting the data in transit.

After connecting A and B of a module to the network or to the Master with the network powered, if the 'Wait' LED lights up the connection is correct, while if the “TxRx” LED lights up, the connections must be exchanged.
The LEDs light up EVEN IF THE BY11800 MODULE IS OFF.

A well realized network allows to maximize the communication baudrate which, in the case of the BY11800-COUNT is 230400 bps.

However, it is advisable to always refer to the “MODBUS over serial line specification and implementation guide”, available on the website www.Modbus.org, which contains in detail the recommendations that must always be observed, especially as regards the 2-wire RS485 connection.

4 Operation

When the device is powered, the ON LED is on.

The arrival of an impulse, regardless of whether from C1 or C1, causes the yellow LED (2xCh) to light up for a minimum duration of 10 msec. If the speed of the impulses is high, the LED appears constantly on. The use of this LED is simply to indicate the presence of pulses and cannot be used for further counting indications.

The BY11800 realizes a SLAVE Modbus interface (RTU and ASCII Modbus) that can be interrogated through the high speed optoisolated RS485 serial port. Protocol recognition is automatic based on the request format.

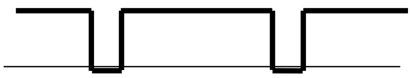
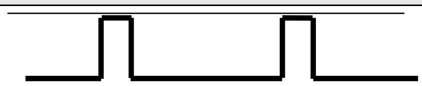
Before being put into service, the BY11800-COUNT needs to be configured.
In this regard, there are some Modbus registers accessible in reading and writing which are permanently stored.
The instrument must be configured individually, isolated from the Modbus network in which it will be inserted.
By default, the communication parameters are as follows:

- Modbus address (register 257) = 1
- Serial port baudrate (register 258) = 4 (115,200 bps)
- No parity (uneditable)
- 8 data bits (uneditable)
- 1 stop bit (uneditable)
- No handshaking (uneditable)

4.1 Configuration of inputs C1 and C2

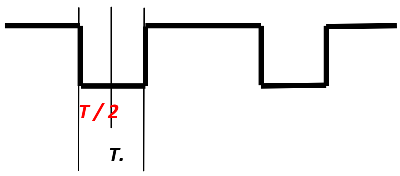
4.1.1 Pulse polarity (registers 259 and 278)

The inputs C1 and C2 are kept high (+ 5V) through internal resistances, with respect to the reference terminal Com.
The change of state of each input can occur when the external contact in correspondence of the impulse closes the input towards Com (condition <NO>, polarity "0") or when the external contact in correspondence of the impulse separates Com from the input (<NC> condition, polarity "1"):

Input for "NO" contact	Input for "NC" contact
	
If on C1: Set 259 = 0 (factory default) If on C2: Set 260 = 0 (factory default)	If on C1: Set 259 = 1 If on C2: Set 260 = 1

4.1.2 Input Filters (Registers 260 and 279)

The impulses coming from mechanical contacts (relays, reed contacts, switches, ...) are ALWAYS subject to 'bounces' to a greater or lesser extent depending on various factors (type of command, contact wear, etc.).
In use **with mechanical contacts, it is ALWAYS COMPULSORY to set an adequate filter time**, under penalty of incorrect counting of impulses.
This time can be optimally chosen by setting a value equal to half the valid pulse duration.
For example, if a meter outputs pulses of 100 ms duration (T), a filter time of 50 ms (T / 2) should be chosen.



The value for C1, with 100 us resolution (±100 us), must be set in register 260 (factory setting 0.0)
The value for C2, with 100 us resolution (±100 us), must be set in register 279 (factory setting 0.0)
Incorrect settings of the mechanical contact filter time (too low or too high) cause counting errors.

4.2 Principle of accounting and accumulation

The 2 inputs are TOTALLY independent from each other with respect to the counting, processing and accumulation of the pulses.
The resources dedicated to each 'measurement channel' are symmetrical.

Each channel has:

- A pre-settable and resettable Pulse Totalizer, with a capacity of 1018 pulses, which contains ALL valid pulses (i.e. those that have passed their input filter), at a maximum speed of 9000 pulses per second (filter = 0). The Totalizer is saved in real time in the permanent memory every time its content changes. The Totalizer, for convenience, has been divided into two parts: the Units part, up to one billion (999.999.999) and the Billion part, also up to one billion (999.999.999). The Totalizer of each channel can be read respectively at addresses 522-524 for channel 1 and 528-530 for channel 2, according to the following table:

RO register	Description
522	Totalizer 1: Billion Billion (top)
523	Totalizer 1: Billion Billion (bottom)
524	Totalizer 1: Units up to one billion (top)
525	Totalizer 1: Units up to one billion (bottom)
528	Totalizer 2: Billion Billion (top)
529	Totalizer 2: Billion Billion (bottom)
530	Totalizer 2: Units up to one billion (top)
531	Totalizer 2: Units up to one billion (bottom)

  The addresses of the Totalizers are read only.
  To be able to pre-set a count value, or to reset a totalizer, you need to:
  - Pre-load the corresponding Totalizer SETTING register (261, 262, 263 and 264 for totalizer 1 and 280, 281, 282 and 283 for 2) with the desired values of units and billions. These registers have the same structure as the corresponding totalizers described in the table.
  - Set the totalizer setting command to 1 (256.2 for totalizer 1 and 256.3 for 2). The value is transferred from the setting registers to the totalization ones. All set registers that have been transferred are cleared, and the set command also returns to zero.
  - The reset occurs by leaving zero in the SETTING register and sending the required command.

- An Engineering Accumulator that can be activated / deactivated / configurable / pre-settable / resettable, which when active is able to relate to the totalizer in a dynamic way (modifying it according to its settings and content) or passively, converting the contents of the totalizer into a decomposed value partly integer and partly fractional according to the parameters. The purpose is to reproduce a remote numeral value in the same style as the possible numerator on the instrument from which the impulses come. The available parameters allow this accumulator to progress in the same way as the physical numerator of the instrument, as long as the number of incoming impulses allows the same resolution.

4.2.1 Use of engineering accumulators

When there is a counting device that has its own numerator, it is possible to program a number of starting pulses that correspond to the 'weighed' number represented by the device's numerator. The numerators of the instruments that count flows (liquids, gases, energy, etc.), are characterized by the presence of an integer part (liters, cubic meters, kWh ...) almost always followed by a fractional part (a 1, 2, 3 decimal places and rarely even more).
For this purpose there are 2 parameters for each accumulator, which allow to define:

- **Number of integers**, 1 to 9 (266 and 285) e
- **Number of fractional digits**, 0 to 9 (267 and 286).

IMPORTANT: before modifying the parameters relating to an engineering accumulator, disable it by setting the appropriate register (265 or 284) to 0. This is because otherwise there would be interaction with the Totalizer and some values could be destroyed, modified in an involuntary way or difficult to set due to the redundancy of partial calculations performed in real time.

That said, let's see how to proceed with the settings.
Theoretically we have a 'virtual' numerator made up of at most 9 whole decimal digits and as many fractional digits:

INTEGER_ACCUM									FRACTIONAL_ACCUM								
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

By modifying the Number of integer digits and Number of fractional digits parameters, it is possible to configure at least a single digit integer decimal numerator.

To understand how to correctly configure an accumulator, we will make the practical example of an electricity meter that measures kWh with 6 integers and 2 decimal digits, with 2 pulse outputs: an electronic for calibration, which provides 10000 pulses / kWh and a mechanic (relay), able to supply 1 pulse every kWh. The counter display numerator indicates "058372.27" kWh.
The intention is to configure one of the accumulators so that the reading of the integer part register (273-274 or 292-293) remotely starts from 58372 and that the fractional part (275-276 or 294-295) starts from 27 and that from that moment both the advancement of the counter numerator and the content of the corresponding accumulator are identical.
Before knowing how many digits to set, we need to decide which counter output to use.
The electronic output has a large number of pulses BUT these would ALL be memorized 'consuming' the totalisation capacity faster. The mechanical output allows to 'save' on the totalizer capacity, but by emitting a pulse at each kWh it does not allow to appreciate the two decimals, which even if we decided to use they would remain at zero. In this case, we should only use the integer part accumulator.
For example, we decide to use the electronic output, after having checked that at maximum power the pulse emission frequency is less than 9000 per second, and then to configure 6 integer digits and 2 decimals.
With the accumulator DISABLED (265 = 0 or 284 = 0) we therefore configure:

- **Number of integers = 6** (266 or 285) and
- **Number of fractional digits = 2** (267 or 286).

We pre-set the numerator values in the accumulator, respectively in the integer part (273-274 or 292-293) and in the fractional part (275-276 or 294-295):

273-274 292-293						275-276 294-295	
	5	8	3	7	2	2	7

Note: the counter may trigger the last digit even after a single pulse. It is not possible to predict to keep the trigger aligned because the quantity of pulses that the counter has already recorded is not known.
To make sure that the advancement of the accumulator digits is the same as the counter numerator (less than one unit on the least significant digit), we have the following parameters:

- **Divisor**, from 1 to 1000 (268 or 287). It is used to define EVERY HOW MANY PULSES the accumulator must be updated.
- **Integer quantity to add**, zero to 10 Integer number - 1 (269-270 or 288-289). This quantity is added to the integer part of the accumulator each time the accumulator needs to be updated. The rest is overlooked.
- **Fractional amount to add**, from zero to 10 Number of fractional digits - 1 (271-272 or 290-291). This amount is added to the decimal part of the accumulator each time the accumulator needs to be updated. The rest is added to the integer part.

In our case, the right-hand decimal advances every 10 Wh. If we want to see it progress correctly, if we have 10,000 imp / kWh available, that is 10 imp / Wh, we must set the Divisor = 100, so that the accumulator update takes place every 10 Wh. The Totalizer still keeps and stores ALL the impulses.
Each time the accumulator has to be updated, the quantity to be added will be:

- **Integer quantity to add = 0,**
- **Fractional amount to add = 1** (because compared to 10 Number of fractional digits, 100 in our case)

In this way, every 10Wh the value 0.01 will be added to the accumulator.
This explains why the accumulator has been defined as such and not as a counter, since it allows each update to 'accumulate' a value made up of an integer and a decimal part, which does not necessarily have to be 1.
When the counter limit is reached / exceeded, i.e. 10 Number of integers, the accumulator will show the same values on the counter's numerator, which in the meantime has restarted from zero, BUT will continue to keep ALL the pulses in the totalizer, which will return to zero only after reaching 1018 pulses, unless manually changed or reset first.

Only after completing the configuration described above can the accumulator be activated.

IMPORTANT:

When the accumulator is enabled (265 = from 0 to 1 or 284 = from 0 to 1):

- if the pre-set value in the accumulator itself is zero, if there are pulses in the totalizer, the accumulator updates itself based on the number of pulses present, i.e. it assumes the equivalent value based on the parameters set, otherwise
- if the pre-set value in the accumulator is NOT zero, the number of equivalent pulses will be calculated based on the parameters set and the corresponding totalizer will be overwritten with this value.

In our case, having pre-set the value 58372.27 with Divisor = 100 and quantity to be added = 0.01, the value of pulses that will be entered in the totalizer (as if they had been counted) will be:

PULSES = (((**INTEGER_ACCUM** * **10**$^{P\_NDecimalDigits}$) + **FRACTIONAL_ACCUM**) * **Dividsor** / ((**ADDED_AMOUNT_INTEGER_PART** * **10**$^{P\_NDecimalDigits}$) + **ADDED_AMOUNT_FRACTIONAL_PART**)
= (((**58372** * **100**) + **27**) * **100**) / ((**0** * **100**) + **1**)
= **5837227**

This value will overwrite the corresponding totalizer and will be stored permanently.
The corresponding value of the accumulator with respect to the number of pulses in the totalizer is given by:

ACCUM = (**PULSES** * ((**ADDED_AMOUNT_INTEGER_PART** * **10**$^{P\_NDecimalDigits}$) + **ADDED_AMOUNT_FRACTIONAL_PART**)) / **Divisor**

The integer part of the division represents **INTEGER_ACCUM** (273-274 or 292-293), while the rest represents **FRACTIONAL_ACCUM** (275-276 or 294-295).

IMPORTANT:

When an accumulator is enabled:

- any direct modification to its value causes the overwriting of the corresponding Totalizer e
- any direct modification to the value of a Totalizer is immediately reflected on the corresponding accumulator.

As already mentioned, if possible, avoid changes to the configuration of an accumulator with the accumulator enabled.

By default, the accumulators are disabled (265 = 0 and 284 = 0).
The rest of the accumulator parameters are configured as follows:

- **Number of integer digits = 9** (266 and 285) and
- **Number of fractional digits = 9** (267 and 286).
- **Divisor** = 1 (268 and 287).
- **Integer quantity to add = 0** (269-270 and 288-289)
- **Fractional amount to add = 1** (271-272 or 290-291)
- **ACCUMULATOR integer part = 0** (273-274 and 292-293)
- **ACCUMULATOR fractional part = 0** (275-276 and 294-295)

4.3 Serial port configuration, address and baudrate (registers 257 and 258)

Be very careful when sending serial port configuration commands. It is an operation to be carried out on ONLY ONE connected instrument at a time. The entered parameters become IMMEDIATELY operative and if they are different from the previous ones they cause the immediate interruption of the communication.

The risks involved in modifying the parameters in an existing network of instruments are:

- Changing the communication speed requires that in order to communicate again with the instrument, the baudrate of the entire network must be adjusted during the interrogation of the instrument itself.
- The change of address, if it is mistakenly the same as that of another, would cause communication irregularities due to the simultaneous attempt to answer the 2 instruments when interrogated. In this case, one of the two instruments should be isolated from the network and reconfigured locally.

Always promptly note the value of the modified parameter (Address and / or communication speed).
The factory default for baudrate is 115200bps and the node number (address) is **1**.
Valid address numbers (register 257) are 1 to 255.
The communication baudrate (register 258) is coded as follows:
0 = 9600, 1 = 19200, 2 = 38400, 3 = 57600, 4 = 115200 and 5 = 230400bps.
If you modify these parameters and are unable to trace the set values, the way to find out is:

- separate the instrument from the mains
- query the scanning instrument (for each node number and / or for each baudrate), pointing one of the two configuration registers until an answer is obtained. The minimum waiting time between one interrogation and another must be 0.5 sec.
- Interact with the instrument using the identified address and baudrate.

4.4 Limitations

The maximum pulse acquisition frequency is 9000 pulses / second, with pulse time = pause time, when the relative filter time register is = 0.0.

5 MODBUS communications

5.1 General

The system communicates using the MODBUS protocol managed in RTU (and JBUS) or ASCII MODBUS mode.
The recognition of the ASCII or RTU protocol is automatic: the instrument replies with the same protocol as the question.

Only 3 Function codes are implemented:
03 (Read Holding Registers)
04 (Read Input Registers)
06 (Write Single Registers)

The 2 Function Codes 03 and 04 can be perfectly superimposed, i.e. they act in the same way on all the registers accessible for reading.
All implemented Function Codes are fully supported by the related Error Codes and Exception Codes.
The addresses of the described registers refer to the MODBUS RTU standard. They remain valid also for JBUS and ASCII MODBUS.
Refer to the MODBUS specifications for further details.

5.2 Communication parameters

Parameter	Setting
Baud rate	9600 - 19200 - 38400 - 57600 - 115200 - 230400
Parity	None (N)
Data bits	8
Stop bit	1
Flow Control	Nobody

To avoid communication errors, consider the instrument response finished ONLY after a pause time of AT LEAST 100mSec from which the response begins, considering the frame valid with all the characters received within the 100mSec distance between one character and another.

5.3 Function Codes

Function	Command
READING	**0x03** (Read Holding Registers)
	0x04 (Read Input Registers)
WRITING	**0x06** (Write Single Register)

5.4 Basic structure of the registers

The architecture and identification of the registers is very different from the classic "device oriented" one that the MODBUS standard provides. The reasons for this choice can be summarized in a better and more streamlined management of communication, which given the relative simplicity of the system is suitable here.

The registers were identified by type:

Register type	Description
RW (Read / Write)	Registers defined RW can be both read and written [03] = [04] / [06].
RO (Read Only)	Registers defined RO can only be read [03] = [04].

5.5 Reading the registers

Reading is allowed on the RW and RO registers indifferently both as Holding (03) and as Input (04) registers. The answer will be ONLY the instrument whose node number corresponds to that of the request, WHICH MUST BE UNIQUE on the network.

5.5.1 Reading of registers in binary mode (RTU)

The conversation takes place in binary Bytes.

RTU reading function			
Binary request frame		**Binary response frame**	
Field	Range	Field	Description
Node	1 - 255	Node	The same as the request
Function	3 - 4	Function	The same as the request
Top address	1 - 65535 (0-0xFFFF)	Bytes amount	Length in BYTES of the returned data block. Double the required registers are worth.
Lower part address			
Upper part Reg. No. required	Always 0		
Lower part Reg. No. required	1 - 125 (1-0x7D)		
Lower part CRC	Calculated, between 0 and 65535 (0-0xFFFF)	Bytes Required (2 x Register)	
		Lower part CRC	Calculated, between 0 and 65535 (0-0xFFFF)
Top CRC		Top CRC	
TOTAL: 8 Bytes		TOTAL: 5 Bytes + N. Bytes Required	

The associated response in the event of an error is as follows:

Error framing RTU read function		
Field	Range	Description
Node	The same as the request	
Function	The same as the request + 128 (0x80)	If request = 3, function = 131 (83 Hex) otherwise if = 4, function = 132 (84 Hex)
Exception Code	1 - 4	1 = Function not supported 2 = Register address or range not valid 3 = Invalid quantity of required registers 4 = Function unavailable / busy
Lower part CRC	Calculated, between 0 and 65535 (0-0xFFFF)	
Top CRC		
TOTAL: 5 Bytes		

5.6 Reading of registers in ASCII mode

The conversation takes place in 7-bit ASCII characters, which in pairs represent the HEXADECIMAL value of the data to be sent or received. Both in transmission and reception, the telegrams always open with the ":" (colon) and always close with CR (Carriage Return), ie binary byte = 13 and LF (Line Feed) which is 10 in binary. RTU mode, the calculation of the CRC is replaced by that of the LRC (Longitudinal Rendoundancy Check).

ASCII reading function			
ASCII-HEX request frame		**ASCII-HEX response frame**	
Field	Range	Field	Description
Start Transmission	":"	Start Transmission	The same as the request
Node	01 - FF	Node	The same as the request
Function	03 - 04	Function	The same as the request
Top address Lower part address	0001 - FFFF	Bytes amount	Length in BYTES of the returned data block (02 - FA)
Upper part Reg. No. required	00 (Always)		
Lower part Reg. No. required	01 -7D	Bytes Required in ASCII-HEX (2 x Byte)	
LRC	Calculated, between 00 and FF	LRC	Calculated, between 00 and FF
Carriage Return	13 bin (1 character)	Carriage Return	13 bin (1 character)
Line Feed	10 bin (1 character)	Line Feed	10 bin (1 character)
TOTAL: 17 Characters		TOTAL: 11 Characters + 2 for each Byte requested	

The associated response in the event of an error is as follows:

Error framing ASCII read function		
Field	Range	Description
Start Transmission	":"	
Node	The same as the request	
Function	ASCII-HEX value of the requested function + 0x80	If request = 03, function = 83 otherwise if = 04, function = 84
Exception Code	01 - 04	01 = Function not supported 02 = Register address or range not valid 03 = Invalid quantity of required registers 04 = Function unavailable / busy
LRC	Calculated, between 00 and FF	
Carriage Return	13 bin (1 character)	
Line Feed	10 bin (1 character)	
TOTAL: 11 Characters		

5.7 Writing of registers

Writing is allowed only on RW registers.
Only the Single Register (06) writing function is implemented. ONLY the instrument whose node number corresponds to that of the request will react to the command, WHICH MUST BE UNIQUE on the network.

5.7.1 Writing of registers in binary mode (RTU)

The conversation takes place in binary Bytes.

RTU write function			
Binary request frame		**Binary response frame**	
Field	Range	Field	Description
Node	1 - 255	Node	Same as the request frame.
Function	6	Function	
Top address	1 - 65535 (1-0xFFFF)	Top address	
Lower part address		Lower part address	
Upper part of the data	0 - 65535 (0-0xFFFF)	Upper part of the data	
Lower part of the data		Lower part of the data	
Lower part CRC	Calculated, between 0 and 65535 (0-0xFFFF)	Lower part CRC	
Top CRC		Top CRC	
TOTAL: 8 Bytes		TOTAL: 8 Bytes	

The answer in case of an error is as follows:

Error framing RTU write function		
Field	Range	Description
Node	The same as the request	
Function	The same as the request + 128 (0x80)	Request = 6, function = 134 (86 Hex)
Exception Code	1 - 4	1 = Function not supported 2 = Register address not valid 3 = Invalid value 4 = Function unavailable / busy
Lower part CRC	Calculated, between 0 and 65535 (0-0xFFFF)	
Top CRC		
TOTAL: 5 Bytes		

Via Como, 55 - 21050 Cairate (VA) - Tel. 0331/312523 - Fax. 0331/313077 - WEB: www.bytronic.it - e-mail: bytronic@bytronic.it

5.8 List of available registers

5.8.1 Legend:

ADDRESS =	**Modbus register number [.Bit of register].** When the register contains a numeric value, its address is INTEGER. On the other hand, when it contains the Boolean value of a specific bit (flag), the bit must be identified by the point followed by its positional value 0-15 starting from the right. (e.g. 258.10 indicates the eleventh bit of register 258).
READ FUNCTION CODE = 3 or 4.	
	It is the number of the reading function (eg 3 or 4).
WRITE FUNCTION CODE = 6.	
	It is the number of the writing function (eg 6). In the case of register RO this value must be "0" or null.
TYPE = 1234 - 4321 - 3412 - 2134 - 12 - 21 - 10 – 01	
	It represents the position of the Bytes to define the register / half register. The Bytes arriving starting from the address indicated in the ADDRESS field are always in order from left to right. To compose the correct value, you must follow the numerical sequence indicated. Zero indicates byte suppression in order of arrival. The highest byte is indicated by the number "1" and the other bytes must be composed accordingly. With 1234, 4321, 3412 or 2134 we want to indicate a value to be obtained from 2 consecutive registers (32-bit value). The Register following the one indicated in the ADDRESS field MUST be omitted (skipped, not listed). Example: if address = 312, register 313 must not exist. With 12 or 21 we want to indicate a value to be obtained from the entire Register (16bit) whose the upper part is in position "1". With 10 we want to indicate an 8-bit value contained in the first Byte received of the Register, while with 01 contained in the second Byte received of the Register.
FORMAT= DEC - DECS - BIN - BOL- HEX - IEEE - BCD - IEEE.	
	The 32, 16 or 8 bit binary value must be converted to: DEC / DECS = Unsigned / signed decimal value BIN = Binary string of "0" and "1" BOL = True or False value of the bit specified in the argument address HEX = Hexadecimal representation BCD = A 0-9 character every 4 bits IEEE = Floating point IEEE 754, type 1234.
AUTHORIZATION = RO – RW remote permissions.	
	With RO it is not allowed to change the register value. With RW it is allowed to change the value of the register, using the write function provided.
	FACTOR = D - C - M –DM – N comma position. It only makes sense in Decimal formats (DEC or DECS). With N, the decimal place remains integer. With D, the value is to be considered multiplied by 0.1. With C, the value is to be considered multiplied by 0.01. With M, the value is to be considered multiplied by 0.001. With DM, the value is to be considered multiplied by 0.0001.

5.8.2 GROUP BY11800-COUNT CONFIGURATION REGISTERS

ADDRESS	READ	WRITE	TYPE	DESCRIPTION	FORMAT	UM	AUTH	FACTOR
256.0	3	6	12	SYS CMD: 1 = RESET	BOL		RW	No.
256.1	3	6	12	SYS CMD: 1 = Default parameters	BOL		RW	No.
256.2	3	6	12	SYS CMD: 1 = Set totalizer 1	BOL		RW	No.
256.3	3	6	12	SYS CMD: 1 = Set totalizer 2	BOL		RW	No.
256.4	3	6	12	SYS CMD: 1 = Save totalizers	BOL		RW	No.
257	3	6	01	Modbus address	DEC		RW	No.
258	3	6	01	bps (0 = 9600_1 = 19200_2 = 38400_3 = 57600_4 = 115200_5 = 230400)	DEC		RW	No.
259.0	3	6	01	Totalizer Input Polarity 1	BOL		RW	No.
260	3	6	12	Totalizer 1 input pulse filter time	DEC	ms	RW	D.
261	3	6	1 2 3 4	SETTING Totalizer 1 (BILLION)	DEC		RW	No.
263	3	6	1 2 3 4	SETTING Totalizer 1 (UNIT)	DEC		RW	No.
265.0	3	6	01	1 = Enable Engineering Accumulator 1	BOL		RW	No.
266	3	6	01	Number of integer digits Accumulator 1 (1-9)	DEC		RW	No.
267	3	6	01	Number of fractional digits Accumulator 1 (0-9)	DEC		RW	No.
268	3	6	12	Pulse divider for Accumulator 1	DEC		RW	No.
269	3	6	1 2 3 4	Integer quantity to add to Accumulator 1	DEC		RW	No.
271	3	6	1 2 3 4	Fractional quantity to add to Accumulator 1	DEC		RW	No.
273	3	6	1 2 3 4	ACCUMULATOR 1 - INTEGER PART	DEC		RW	No.
275	3	6	1 2 3 4	ACCUMULATOR 1 - FRACTIONAL PART	DEC		RW	No.
278.0	3	6	01	Totalizer Input Polarity 2	BOL		RW	No.
279	3	6	12	Totalizer 2 input pulse filter time	DEC	ms	RW	D.
280	3	6	1 2 3 4	SETTING Totalizer 2 (BILLION)	DEC		RW	No.
282	3	6	1 2 3 4	Totalizer 2 (UNIT) SETTING	DEC		RW	No.
284.0	3	6	01	1 = Enable Engineering Accumulator 2	BOL		RW	No.
285	3	6	01	Number of integer digits Accumulator 2 (1-9)	DEC		RW	No.
286	3	6	01	Number of fractional digits Accumulator 2 (0-9)	DEC		RW	No.
287	3	6	12	Pulse divider for Accumulator 2	DEC		RW	No.
288	3	6	1 2 3 4	Integer quantity to add to Accumulator 2	DEC		RW	No.
290	3	6	1 2 3 4	Fractional quantity to add to Accumulator 2	DEC		RW	No.
292	3	6	1 2 3 4	ACCUMULATOR 2 - INTEGER PART	DEC		RW	No.
294	3	6	1 2 3 4	ACCUMULATOR 2 - FRACTIONAL PART	DEC		RW	No.

5.8.3 GROUP READ ONLY REGISTERS BY11800-COUNT

ADDRESS	READ	WRITE	TYPE	DESCRIPTION	FORMAT	UM	AUTH	FACTOR
512	3		12	Product code	DEC		RO	No.
513	3		12	Manufacturer Code	DEC		RO	No.
514	3		1 2 3 4	CPU Serial Number	DEC		RO	No.
516	3		10	Instrument Model	DEC		RO	No.
516	3		01	Instrument version	DEC		RO	No.
517	3		10	Revision	DEC		RO	No.
517	3		01	Day	DEC		RO	No.
518	3		10	Month	DEC		RO	No.
518	3		01	Year	DEC		RO	No.
519	3		12	Cpu ID	DEC		RO	No.
521	3		10	Bootloader Version (Major)	DEC		RO	No.
521	3		01	Bootloader Version (Minor)	DEC		RO	No.
522	3		1 2 3 4	TOTALIZER 1 (BILLION)	DEC		RO	No.
524	3		1 2 3 4	TOTALIZER 1 (UNIT)	DEC		RO	No.
526	3		1 2 3 4	Accumulator 1 reset value	DEC		RO	No.
528	3		1 2 3 4	TOTALIZER 2 (BILLION)	DEC		RO	No.
530	3		1 2 3 4	TOTALIZER 2 (UNIT)	DEC		RO	No.
532	3		1 2 3 4	Accumulator 2 reset value	DEC		RO	No.

6 Technical data

TECHNICAL FEATURES		
Power Supply		80-265Vac 50/60 Hz / 100-380 Vdc
Self-consumption		1 VA
Isolation		RS485 / power supply
Test voltage		3 kV
Serial interface		RS485 (isol.3kV)
Serial communication protocol		Slave Modbus RTU (JBUS) / ASCII Modbus
Serial communication baudrate		9600-19200-38400-57600-115200-230400 (programmable)
Serial communication parameters		8, N, 1 no handshaking
Modbus address		1 to 255 (programmable)
Number of Pulse Totalizers		2
Capacity of the pulse totalizers		1018 (1 Billion Billion)
MAX total pulse rate		9000 / sec (Ton = Toff, Tfilter = 0)
Totalizers saving method		Continuous
Time Filters mechanical contacts		0.0 - 5000.0 ms
Number of engineering accumulators		2
Capacity Engineering accumulators		999999999,999999999.
Dimensions / weight		1 DIN module / 0.10 Kg
Mechanical characteristics	Mounting Type:	DIN rail 50022
	Degree of protection:	Complete instrument: IP20 / front: IP30
Environmental conditions	room temperature:	0 ... + 45 ° C
	extreme range:	-5 ... + 55 ° C
	storage temperature:	-10 ... + 70 ° C
	relative humidity:	10 ... 95%
	atmospheric pressure:	70 ... 110 kPa
Reference standards	Safety:	EN 61010-1 CAT II
	Degree of protection of the enclosures (IP):	EN 60529
	Electromagnetic compatibility (immunity):	EN 61000-6-2
	Electromagnetic compatibility (emission):	EN 61000-6-4