MODBUS-RTU for Vision Touch WELLNESS

MODBUS-RTU protocol specifications for LAN control of Vision Touch WELLNESS series devices

Document name: **MODBUS-RTU_VT_WEL_01-23_ENG**
Installed software: VT_WEL_5_0_0_0**.pego**

READ AND KEEP

INDEX

GENERAL DESCRIPTION

1

DESCRIPTION OF CONTROLS

2

REGISTERS AND ADDRESSES DESCRIPTION

3

GLOSSARY

4

1: GENERAL DESCRIPTION

1.1 MODBUS PROTOCOL

The data communication system based on Modbus protocol allows to connect up to 247 devices in a common RS485 line with standard format and communication mode.
Communication takes place in half duplex by frame (transmitted continuously); only master (PC , PLC ...) can start polling with slaves as question/answer (only one slave addressed) and the polled slave answers. The slave answers after a minimum pause of 3,5 characters between received frame and the one to be transmitted.
Also broadcast communication mode exists where the master send a request to all the slaves simultaneously, and they give no answer back; this mode it's not available with this controller.
The data serial transmission mode implemented on the controller is RTU type (Remote Terminal Unit), where data are exchanged in binary format (8 bit characters).

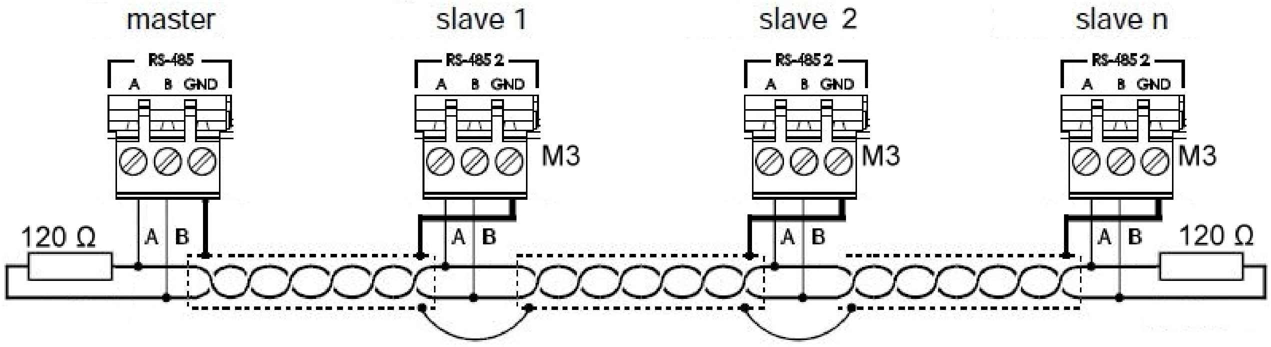
1.2 SERIAL CONFIGURATION

Serial line:	***RS485***
Baud rate:	***300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400***
Data length:	***8 bit***
Parity:	***none, left or right***

Serial transmission of characters in RTU format

Start	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Parità (optional)	Stop 1

Is recommended to connect a resistance of 120Ω tra A e B between A and B at the beginning and at the end of the line in case of communication problems.
For proper operation, the Master must have a polarized RS485.



1.3 MESSAGE FORMAT (FRAME)

Each message (Frame) is made, based on MODBUS-RTU standard, by the following parts:

Start	Device address	Function code	Data	CRC16		Stop
silence of (3,5 x byte time) msec	Byte	Byte	n x Byte	LSByte	MSByte	silence of (3,5 x byte time) msec

- **Start / Stop :**
  Message starts with a silence equal to 3.5 times of one byte transmission time. See chap. 4.1 for further clarifications.

- **Device address**:
  Device address with whom the master established the communication. It's a value between 1 and 247. Address 0 is reserved to the broadcast, message sent to all slave devices (not active on this controller). RS485 line allows to connect together up to 32 devices (1 Master + 31 slaves), but with appropriate "bridges" or relay devices it is possible to use the whole logical addressing field.

- **Function code:**
  The code of the function to be executed or was executed; On device are active codes 0x03 (register reading), 0x06 (single register writing) and 0x2B/0x0E (identification data reading).

- **Data:**
  Data that must be exchanged.

- **CRC16:**
  Error checking field based on CRC16 algorithm. CRC16 is calculated on the whole message by the master device which is trasmitting and attached to the message itself. The slave, at the end of reception, calculates CRC16 on the message and compares it with the value learnt by the master; if the values do not match, the message will be considered not valid and will be discarded without sending any answer to the master.
  The following fragment of C code shows the CRC16 calculation mode:

```
unsigned int CRC16
void Modbus_CRC(unsigned char *Frame, unsigned char FrameLength)
{
unsigned char ByteCount;
unsigned char i;
unsigned char bit_lsb;
CRC16 = 0xFFFF;
for (ByteCount=0;ByteCount<FrameLength;ByteCount++)
  {
  CRC16^=Frame[ByteCount];
  for (i=0;i<8,i++)
    {
    bit_lsb = CRC16 & 0x0001;
    CRC16 = CRC16>>1;
    if (bit_lsb == 1)
      CRC16 ^= 0xA001;
    }
  }
}
```

1.4 MESSAGES SYNCHRONIZATION

Message synchronization between transmitter and receiver is made placing a pause on the messages at least 3.5 times the character transmission period. If the receiver does not receive any Byte for 3.5 times of one byte transmission period, the last message is considered completed and the next Byte received is set as the first one of a new message.

The slave, once received the complete message, decodes it and, if there are no errors, sends the answer message to the master. To send the answer, slave keeps RS485 line busy, wait a pause of 3.5 times the byte transmission period, send the complete message, wait 3.5 times the byte transmission period and then release the RS485 line.

The master unit will have to consider these periods to avoid risks of transmission overlap; in particular must be set a proper answer reception time-out before starting a new transmission (typical time-out value: 500msec or higher, for a baud rate = 9600).

1.5 ERROR MESSAGES (EXCEPTIONS)

If is not possible to complete the required operation, the device answers with an error message in the following format:

Device address	*Function Code*	*Exception Code*	*CRC16*	
Byte	Byte	Byte	LSByte	Byte

- **Device address**: Address of slave device answering
- **Function Code**: Function code MSb =1 (to show exception); i.e. 0x83 (for 0x03 reading) or 0x86 (for 0x06 writing)
- **Exception Code**: Exception codes handled by the device are the following:

Exception code	*Description*	*Exception cause*
0x01	Function not implemented	A function code not available was requested, different from 0x03, 0x06 and 0x2B/0x0E.
0x02	Address not valid	It's generated in several situations: - a not implemented register has been requested (or a not-existing area) - a reading of a number of registers that goes further on the implemented area has been requested (starting from requested address) - tried to write on a read-only area
0x03	Value not valid for datum	It's generated in several situations: - message 0x2B/0x0E DeviceIdCode is not correct - has been tried to write a parameter with an out of range value

- **CRC16:** Error control field based on the CRC16 algorithm.

Note:
In case the device identifies in the received message an error on format or in CRC16, the message is discarded (considered not valid) and no answer is sent.

2: DESCRIPTION OF CONTROLS

All the registers, to equalize the interpretation, are handled in a Word format (16 bit), even if an 8-bit parameter is contained.

2.1 REGISTER READING (0x03)

Format of the command sent by the Master:

Device address	Function Code	Register address		Number of registers		CRC16	
Byte	Byte	MSByte	LSByte	MSByte	Byte	Byte	MSByte

- **Device address**:
  The address of the slave device to be queried
- **Function Code**:
  Function code to be executed, in this case register reading (0x03)
- **Register address**:
  Starting register address for reading expressed with two Bytes; (MSByte) and (LSByte).
- **Number of registers**:
  indicates the number of Word required from the starting address. If a number of registers more than 1 is requested, the answer message will provide all the registers required with consecutive addresses starting from the address shown on the "register address" field.
  The number of registers to read is expressed on two Bytes, particularly for this controller (MSByte) must always be 0x00 and (LSByte) with range 1-10.
- **CRC16:**
  Error control field based on the CRC16 algorithm.

Format of answer message from the slave:

Device address	Function Code	Bytes of datum No.	Datum 1		Datum 2		Datum n		CRC16	
Byte	Byte	Byte	MSByte	LSByte	MSByte	LSByte	Byte	Byte	Byte	MSByte

- **Device address**:
  The address of the slave device that responds
- **Function Code**:
  Function code to be answered to, in this case register reading (0x03)
- **Number of bytes of datum**:
  It contains the total number of bytes of datum.
  Consider that the number of bytes of datum is the double of the number of registers (because we talk about word). I.e. if in the message of request 2 registers are requested, the number of bytes of datum must be set as 4 in the answer message.
- **Datum n** :
  It contains the sequence of the data each expressed on two bytes; (MSByte) e (LSByte).
- **CRC16:**
  Error control field based on the CRC16 algorithm.

2.2 SINGLE REGISTER WRITING (0x06)

Format of the command sent by the Master:

Device address	Function Code	Register address		Datum		CRC16	
Byte	Byte	MSByte	LSByte	MSByte	Byte	Byte	MSByte

- **Device address**:
  The address of the slave device to be queried
- **Function Code**:
  Function code to be executed, in this case single register writing (0x06)
- **Register address**:
  Address of register to write expressed with two Bytes; (MSByte) and (LSByte).
- **Data**:
  Value to be assigned to the register expressed with two Bytes; (MSByte) and (LSByte).
- **CRC16:**
  Error control field based on the CRC16 algorithm.

Format of the answer message from the slave:

Device address	Function Code	Register address		Datum		CRC16	
Byte	Byte	MSByte	LSByte	MSByte	Byte	Byte	MSByte

The answer message is a simple echo of the message of request to confirm that the variable has been modified.

2.3 READING DEVICE IDENTIFICATION DATA (0x2B / 0x0E)

Format of the command sent by the Master:

Device address	Function Code	MEI type	Read Device Id Code	Object Id	CRC16	
Byte	Byte	Byte	Byte	Byte	LSByte	Byte

- **Device address**:
  The address of the slave device to be queried
- **Function Code**:
  Function code to be executed, in this case identification data reading (0x2B)
- **MEI type**:
  Modbus Encapsulated Interface type: it must be 0x0E.
- **Read Device Id Code**:
  It indicates the access type to data: it must be 0x01.
- **Object Id**:
  It indicates the starting object for data reading (range: 0x00 – 0x02).
- **CRC16:**
  Error control field based on the CRC16 algorithm.

Format of the answer message from the slave:

Device address	Functio n code	MEI Type	Read Device Id Code	Confor mity level	More Follows	Next Object Id	Number Of Object	Object Id (n)	Object Length (n)	Object Value (n)	CRC16	
Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte	ASCII String	LSByte	Byte

- **Device address**:
  The address of the slave device that responds
- **Function Code**:
  Function code to be executed, in this case identification data reading (0x2B)
- **MEI type**:
  Modbus Encapsulated Interface type: it must be 0x0E.
- **Read Device Id Code**:
  It indicates the access type to data: it must be 0x01.
- **Conformity level**:
  It indicates the slave conformity level: it is always 0x01.
- **More Follows**:
  It indicates the number of additional transactions requested: it is always 0x00.
- **Next Object Id**:
  It indicates the object that has to be requested in the eventual following transaction: it is always 0x00
- **Number Of Object**:
  The number of objects that follow (1, 2 o 3).

- **List of**:
  - **Object Id**:
    current object number.
  - **Object Length**:
    length of following string.
  - **Object Value**:
    ASCII string that contains the identification information.
- **CRC16:**
  Error control field based on the CRC16 algorithm.

Reading example of all controllers identification information with software VT_WEL rev. 0 (address 1)

Request message: (01 2B 0E 01 00 70 77)

- **Device address**: 0x01
- **Function code**: 0x2B
- **MEI type**: 0x0E
- **Read DeviceIdCode**: 0x01
- **ObjectId**: 0x00
- **CRC16:** to be calculated on the previous values

Answer message: (01 2B 0E 01 01 00 00 03 00 04 50 45 47 4F 01 08 56 54 5F 5F 5F 57 45 4C 02 03 30 30 30 2A CE)

- **Device address**: 0x01
- **Function code**: 0x2B
- **MEI type**: 0x0E
- **Read DeviceIdCode**: 0x01
- **Conformity level**: 0x01
- **More Follows**: 0x00
- **Next ObjectId**: 0x00
- **Number Of Object**: 0x03
- **ObjectId**: 0x00
- **Object Length**: 0x04
- **Object Value**: ‘PEGO’ (Vendor Name field in ASCII)
- **ObjectId**: 0x01
- **Object Length**: 0x08
- **Object** Value: ‘VT___WEL’ (Product Code field in ASCII)
- **ObjectId**: 0x02
- **Object Length**: 0x03
- **Object Value**: ‘000’ (Revision field in ASCII)
- **CRC16:** to be calculated on the previous values
-

3: REGISTERS AND ADDRESSES DESCRIPTION

Each register has a 16 bit dimension. It has been formed some blocks of variables (each with a different MSByte address) basing on the type of these variables. In the followings paragraphs are described in the detail all the available blocks and, for each block, the implemented variables.
At the beginning of each table it has been indicated in the first row if its data could be only read (READ-ONLY) or written and read (READ/WRITE).

TABLE COLUMNS DESCRIPTION:

- **Register** :
  It indicates the register address to be used in the Modbus command structure for reading or writing data to the instrument. It is expressed on two Bytes; (MSByte) and (LSByte).
- **Description** :
  Description of the register and possible corresponding programming variable of the device.
- **Meaning and Bytes range:**
  Dimension (MSByte and LSByte), allowed range and notes about the register.
- **U.M. :**
  Unit of measure of datum contained in the register.
- **Conv. :**
  Values contained in the registers that represent signed variables require a conversion and they are marked from **X** sign in the following column.
  Conversion procedure:
  - If the value contained in the register is included between 0 and 32767, it represents a positive or null number (the results is the value itself)
  - If the value contained in the register is included between 32768 and 65535, it represents a negative number (the results is the register value - 65536)
- **Molt :**
  It indicates the multiplication factor to be applied to the data of the register and that in combination with U.m and Conv allow accurate interpretation of the value contained in it.
  Example:
  A datum **(0x0012)** = 18 con Molt =**0,1** / U.m= **°C** / Conv=**C** corresponds to a temperature of (18x0,1)= **1,8 °C**
  A datum **(0xFFF0)** = 65520 con Molt =**0,1** / U.m= **°C** / Conv=**C** corresponds to a temperature [(65520 – 65536) x0,1] = **-1,6 °C**
  A datum **(0x0078)** = 120 con Molt =**1** / U.m= **min** / Conv=**C** corresponds to a time of (120x1)= **120 minutes**
  A datum **(0x0014)** = 20 con Molt =**0,1** / U.m= **°C** / Conv=**C** corresponds to a temperature of (20x0,1)= **2,0 °C**

3.1 ANALOG INPUTS

READ-ONLY						
Register	**Description**	**Bytes meaning and range**		**U.M.**	**Con v**	**Molt**
256	Ambient temperature	MSByte LSByte	Resolution 0,1°C range: -45°C .. +99°C Value = 999,9°C indicates faulty probe	°C	X	0,1
257	Ambient humidity	MSByte LSByte	Resolution 1% range: 0 .. 99% Value = 999% indicates faulty probe	%		1

3.2 PARAMETERS

READ / WRITE

Register	Description	Bytes meaning and range		U.M.	Conv	Molt
768	Temperature Setpoint	MSByte	Step of 0.1 °C, with sign range: 0.0..HSt	°C		0,1
		LSByte				
769	Humidity setpoint	MSByte	Step of 1 % range: 0..100	%		1
		LSByte				
770	**dtC** Hot temperature differential	MSByte	Step of 0.1 °C range: +0.2..10.0 °C	°C		0,1
		LSByte				
771	**dUU** Differential of humidification	MSByte	Step of 1 % range: 1%..10%	%		1
		LSByte				
772	**drA** Air change duration	MSByte	Step of 1 minute range: 0..120 minuti	min		1
		LSByte				
773	**SrA** Air change temperature setpoint	MSByte	Step of 0.1 °C, with sign range: 30.0..99.0°C	°C		0,1
		LSByte				
774	**tF** Operating time	MSByte	Step of 1 minute range: 0..tFm minuti	min		1
		LSByte				
775	**tFm** Maximum operating time that can be set	MSByte	Step of 1 minute range: 0..720 minuti	min		1
		LSByte				
776	**oAp** Scheduled activation time	MSByte	Step of 1 minute range: 0...1439 minuti	min		1
		LSByte				
777	**Enable oAp** Enabling scheduled activation time	MSByte	range: 0..1 (0=disabled)	num		1
		LSByte				
778	**HSt** Maximum value of the temperature setpoint	MSByte	Step of 0.1 °C, with sign range: 20.0..99.0°C	°C		0,1
		LSByte				
779	**Ldt** Speed of the colour sequence during the automatic cycle	MSByte	Step of 1 second range: 1...120 seconds	sec		1
		LSByte				
780	**Lcy** Type of colour cycle	MSByte	Step of 1 range: 1...4 1=120 colours 2=4 colours (red, yellow, green, blue) 3=cold cycle 4=hot cycle	num		1
		LSByte				

3.2a REAL-TIME CLOCK PARAMETERS

READ/WRITE						
Register	**Description**	**Bytes meaning and range**		**U.M.**	**Conv**	**Molt**
1024	Minutes of the Clock	MSByte	Range: 0..59	Min.		1
		LSByte				
1025	Hour of the Clock	MSByte	Range: 0..23	Hour		1
		LSByte				
1026	Year	MSByte	Range: 2000..4000	num		1
		LSByte				
1027	Month	MSByte	Range: 1..12	num		1
		LSByte				
1028	Day	MSByte	Range: 1..28, 1..29, 1..30, 1..31 (depending on month and year)	num		1
		LSByte				
1029	Minutes of the Clock	MSByte	Range: 0..6 0=Monday … 6=Sunday	num		1
		LSByte				

N.B.
- When you change a parameter of the real-time clock, the seconds of the clock are forced to zero.
- Real-time clock parameters that cannot be modified with regulation in progress.

3.2b PARAMETERS

READ-ONLY

Register	Description	Bytes meaning and range		U.M.	Conv	Molt
512	**Cat** Ambient probe calibration	MSByte LSByte	Steps of 0.1 °C range: -10°C..+10°C	°C	x	0.1
513	**CaU** Humidity probe calibration	MSByte LSByte	Steps of 1 % range: -20%..+20%	%		1
514	**mOd** Functioning mode	MSByte LSByte	0 = sauna 1 = wet sauna 2 = turkish bath	num		1
515	**EnA** Air change enabling	MSByte LSByte	range: 0..1 (0=disabled)	num		1
516	**EnC** Enabling the programmable thermostat	MSByte LSByte	range: 0..1 (0= disabled)	num		1
517	**EUm** Enabling Easysteam humidifier	MSByte LSByte	range: 0..1 (0= disabled)	num		1

3.3 INPUTS / OUTPUTS / ALARMS STATUS

READ-ONLY

Register	Description	Bytes meaning and range			U.M.	Conv	Molt
1280	outputs 1 status	MSByte	bit 7 (MSb)	Not used	num		1
			bit 6	Not used			
			bit 5	Not used			
			bit 4	Not used			
			bit 3	Not used			
			bit 2	allarme			
			bit 1	Not used			
			bit 0 (LSb)	Heated air change			
		LSByte	bit 7 (MSb)	Not used			
			bit 6	RGB light			
			bit 5	Light			
			bit 4	Humidification call			
			bit 3	Air change			
			bit 2	Hot call 3			
			bit 1	Hot call 2			
			bit 0 (LSb)	Hot call 1			
1281	outputs 2 status	MSByte	bit 7 (MSb)	Not used	num		1
			bit 6				
			bit 5				
			bit 4				
			bit 3				
			bit 2				
			bit 1				
			bit 0 (LSb)				
		LSByte	bit 7 (MSb)	Not used			
			bit 6				
			bit 5				
			bit 4				
			bit 3				
			bit 2				
			bit 1				
			bit 0 (LSb)				

Important

- The status of the outputs is that of the WEL functions and not of the Test Center.

READ-ONLY							
Register	**Description**	**Bytes meaning and range**			**U.M.**	**Conv**	**Molt**
1282	Inputs 1 status	MSByte	bit 7 (MSb)	Not used	num		1
			bit 6	Not used			
			bit 5	Not used			
			bit 4	Not used			
			bit 3	Not used			
			bit 2	Generic warning 3			
			bit 1	Generic warning 2			
			bit 0 (LSb)	Generic warning 1			
		LSByte	bit 7 (MSb)	Fans protections			
			bit 6	Humidifier alarm			
			bit 5	Not used			
			bit 4	Generic alarm			
			bit 3	Not used			
			bit 2	Disable humidity			
			bit 1	Disable hot			
			bit 0 (LSb)	Stand-by			
1283	Inputs 2 status	MSByte	bit 7 (MSb)	Not used	num		1
			bit 6				
			bit 5				
			bit 4				
			bit 3				
			bit 2				
			bit 1				
			bit 0 (LSb)				
		LSByte	bit 7 (MSb)	Not used			
			bit 6				
			bit 5				
			bit 4				
			bit 3				
			bit 2				
			bit 1				
			bit 0 (LSb)				

READ-ONLY

Register	Description	Bytes meaning and range			U.M.	Conv	Molt
1284	Alarms status 1	MSByte	bit 7 (MSb)	Configuration error 1 (Ec1)	num		1
			bit 6	EEPROM 100Master alarm (E0m)			
			bit 5	Not used			
			bit 4	Initialization error of Master (EnI)			
			bit 3	Not used			
			bit 2	Not used			
			bit 1	Fans protections (EF)			
			bit 0 (LSb)	Humidifier alarm (EU)			
		LSByte	bit 7 (MSb)	General alarm (Eg)			
			bit 6	Probe 5 anomaly (E5)			
			bit 5	Probe 4 anomaly (E4)			
			bit 4	Probe 3 anomaly (E3)			
			bit 3	Probe 2 anomaly (E2)			
			bit 2	Probe 1 anomaly (E1)			
			bit 1	EEPROM Vision Touch alarm (E0)			
			bit 0 (LSb)	Lack of communication (En)			
1285	Alarms status 2	MSByte	bit 7 (MSb)	Generic warning 2	num		1
			bit 6	Generic warning 1			
			bit 5	Serious alarm input dig. humid. (E9U)			
			bit 4	Digital input alarm humid.(E8U)			
			bit 3	Humid water drain alarm. (E6U)			
			bit 2	Humid. water discharge pre-alarm (E5U)			
			bit 1	Humid. water load alarm (E3U)			
			bit 0 (LSb)	Humidity channel probe fault. (E2U)			
		LSByte	bit 7 (MSb)	Hum current fault (E1U)			
			bit 6	Humid probe anomaly (E0U)			
			bit 5	Level alarm (ElU)			
			bit 4	EEPROM error (EEU)			
			bit 3	Humidity communication alarm (EnU)			
			bit 2	Not used			
			bit 1	Configuration Error 3 (Ec3)			
			bit 0 (LSb)	Configuration error 2 (Ec2)			
1286	Alarm status 3	MSByte	bit 7 (MSb)	Not used	num		1
			bit 6				
			bit 5				
			bit 4				
			bit 3				
			bit 2				
			bit 1				
			bit 0 (LSb)				
		LSByte	bit 7 (MSb)	Not used			
			bit 6	Not used			
			bit 5	Configuration error. analog output			
			bit 4	Configuration error. analog input			
			bit 3	Configuration error. digital output			
			bit 2	Configuration error digital input			
			bit 1	Not used			
			bit 0 (LSb)	Generic warning 3			

READ-ONLY						
Register	**Description**	**Bytes meaning and range**		**U.M.**	**Conv**	**Molt**
1287	RGB light code	MSByte	Step of 1	num		1
		LSByte	range: 0..119			

3.4 DEVICE STATUS

READ / WRITE							
Register	**Description**	**Bytes meaning and range**			**U.M.**	**Conv**	**Molt**
1536	Device status	MSByte	bit 7 (MSb)	Not used	num		1
			bit 6	Not used			
			bit 5	Not used			
			bit 4	Not used			
			bit 3	Enable modification air change status			
			bit 2	Enable modification RGB light status			
			bit 1	Enable modification light status			
			bit 0 (LSb)	Enable modification stand-by status			
		LSByte	bit 7 (MSb)	Not used			
			bit 6	Not used			
			bit 5	Not used			
			bit 4	Not used			
			bit 3	Air change status 1 = active 0 = not active			
			bit 2	RGB light status 1 = active 0 = not active			
			bit 1	Light status 1 = active 0 = not active			
			bit 0 (LSb)	Stand-by status 1 = stand-by 0 = ON			

To request a change in one of the bits of device status, the master must send in LSByte the required value for the bit and in MSByte the corresponding bit set to 1. Example: to force the state of standby, the master must send MSByte = 00000001 and LSByte = 00000001.

4: GLOSSARY

- **Binary Number:**
  It's used in computing for the internal representation of the numbers, thanks to the simplicity of physically realize an element with two states (0, 1) rather than a higher number, but also for the correspondence with the true and false logic values.

- **Decimal Number:**
  In the decimal system, all integers can be represented using the ten digits that indicate the first ten natural numbers, including zero. The value of each of these figures depends on the position it occupies within the number, and increases by power of 10 in power of 10, proceeding from right to left.

- **Hexadecimal Number:**
  It's part of a positional number system in base 16, that is, which uses 16 symbols instead of the traditional 10 decimal number system. For the hexadecimal are generally used symbols 0 to 9 and then the letters A to F, for a total of 16 symbols. By convention a number is expressed in hexadecimal preceded by 0x (example: 0x03) or H (example: H03).

- **bit:**
  A bit is a binary digit, that is, one of the two symbols of the binary number system, classically called zero (0) and one (1). It is the definition of a logic unit. It is also defined the elementary units of information processed by a computer.

- **Byte:**
  It's the necessary amount of bits to define an alphanumeric character; in particular, a Byte consists of a sequence of 8-bit (example: 10010110).

- **Word:**
  It's the measuring unit that fix the length of information to 16 bits which is also equivalent to 2 Bytes (example: 10010110 01101011).

- **LSb:**
  Least significant bit of a binary number (the first bit on the right of the number indicated)

- **MSb:**
  Most significant bit of a binary number (the first bit on the left of the number indicated)

- **LSByte:**
  Least significant bit of a Word (Byte on the right of the Word indicated)

- **MSByte:**
  Most significant bit of a Word (Byte on the left of the Word indicated)

PEGO S.r.l.
Via Piacentina, 6/b
45030 OCCHIOBELLO –ROVIGO-
Tel : 0425 762906
Fax: 0425 762905
www.pego.it
e-mail: info@pego.it

Distributor: