

*WINECAP*  
*PLC MODBUS*  
*Specifica Funzionale*

## Matrice delle revisioni

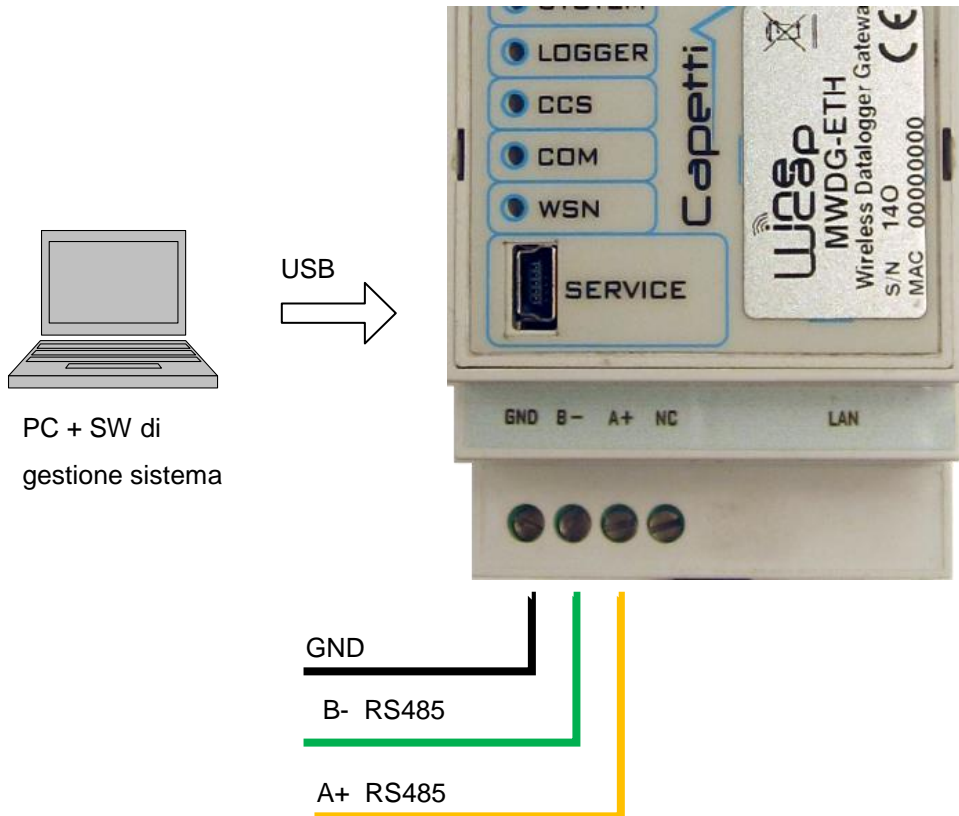
INDICE DI REVISIONE	DATA DI AGGIORNAMENTO	VISTO PER EMISSIONE	SEGNALAZIONE TIPO MODIFICA	DATA DECORRENZA	APPROVAZIONE
<b>1.0</b>	5/02/10		Prima emissione		

## Indice

Matrice delle revisioni .....	2
Indice .....	3
1. Connessione fisica RS485.....	4
1.1. Collegamento al connettore RS485 del dispositivo MWLI o MWDG.....	4
1.2. Collegamento al connettore RS485/232 del dispositivo.....	4
1.3. Parametri di comunicazione seriale canale PLC.....	5
2. Protocollo ModBus.....	6
2.1. Cenni sul protocollo ModBus.....	6
2.2. Modalità RTU.....	8
2.3. Frame RTU.....	8
2.4. Indirizzamento porta PLC del dispositivo.....	8
2.5. Funzioni ModBus abilitate.....	9
2.5.1. Read Holding Register (0x03).....	9
2.5.2. Preset Multiple Register (0x10).....	10
2.6. Gestione delle eccezioni.....	11
3. Registri ModBus.....	12
3.1. Registri di OUTPUT Programmabili.....	12
3.1.1. Tabella valori di misura.....	13
3.1.2. Tabella valori di stato delle uscite.....	14
3.1.3. Tabella contatori temporali delle uscite.....	15
3.1.4. Tabella valori di livello radio delle uscite.....	15
3.1.5. Tabella dei valori di livello batteria delle uscite.....	16
3.1.6. Tabella degli stati relay.....	16
3.1.7. Tabella valori di stato soglie e validità misure.....	16
3.2. Registri di CONFIGURAZIONE e STATO.....	18
3.2.1. Identificazione Datalogger.....	18
3.2.2. Accesso in lettura alle proprietà dei dataloggers.....	19
3.2.3. Accesso in scrittura alla configurazione dei dataloggers.....	19
3.2.4. Accesso in scrittura ai canali dei dataloggers.....	20
3.2.5. Tabella di configurazione gateway.....	21
3.2.6. Tabella di configurazione trasmettitori.....	21
3.2.7. Tabelle di configurazione dei canali di misura.....	22
3.2.8. Tabella Misure.....	23
4. Specifica sul download dei dati di un datalogger dalla basestation.....	25
Appendice A: Calcolo del CRC16 ModBus.....	27
Appendice B: Esempi.....	28

## 1. Connessione fisica RS485.

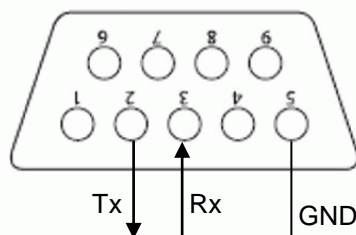
### 1.1. Collegamento al connettore RS485 del dispositivo MWLI o MWDG.



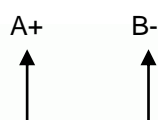
### 1.2. Collegamento al connettore RS485/232 del dispositivo.

I dispositivi **WDG** o **WLI-USB** permettono una connessione RS232 o RS485 con protocollo ModBus per interfacciarsi con un PLC esterno; le linee di comunicazione RS232/485 sono cablate in un connettore a vaschetta 9 poli secondo i due esempi riportati di seguito:

#### Collegamento PLC RS232



#### Collegamento PLC RS485



### 1.3. Parametri di comunicazione seriale canale PLC.

E' possibile impostare i parametri di comunicazione utilizzando il software **WineCapManager** (*vedi manuale*)

<b>Valori di Default</b>	
<b>Baud rate</b>	9600
<b>Bit Dato</b>	8
<b>Bit Stop</b>	1
<b>Parità</b>	PARI

## 2. Protocollo ModBus

### 2.1. Cenni sul protocollo ModBus.

Il protocollo ModBus definisce il formato e la modalità di comunicazione tra il PLC (*master*), che gestisce l'intero sistema di comunicazione, e una o più **basestations** (*slave*) che rispondono alle interrogazioni del master.

Questo protocollo definisce come tra il master e gli slave si stabiliscano e si interrompano le comunicazioni, come trasmettitore e ricevitore debbano essere identificati, come i messaggi debbano essere scambiati e come gli errori rilevati debbano essere interpretati.

Con l'interfaccia standard RS-485 si possono collegare sino a 31 slave ad un master nella stessa linea fisica (*BUS*).

Sostituendo l'ultimo elemento della linea con un apposito "bridge o ripetitore", si possono connettere altri 31 slave e così via sino al raggiungimento del numero massimo logico di indirizzi pari a 247 dispositivi massimi (*limite imposto dal protocollo ModBus*).

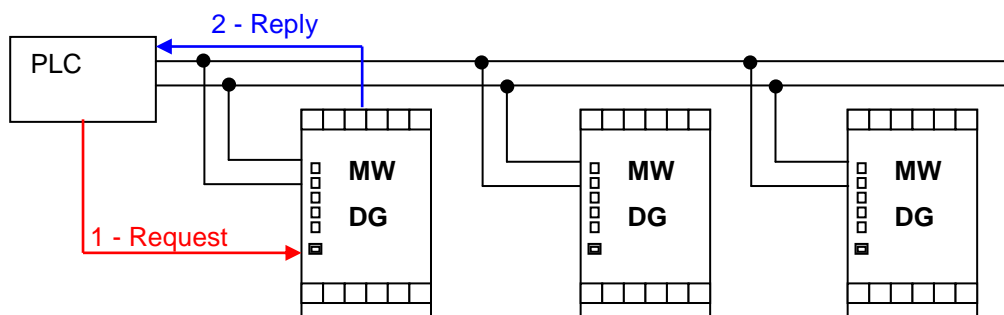
Di seguito una tabella riepilogativa delle caratteristiche di alcuni standard hardware solitamente utilizzati con il protocollo ModBus.

Characteristics of RS232, RS422, RS423 and RS485

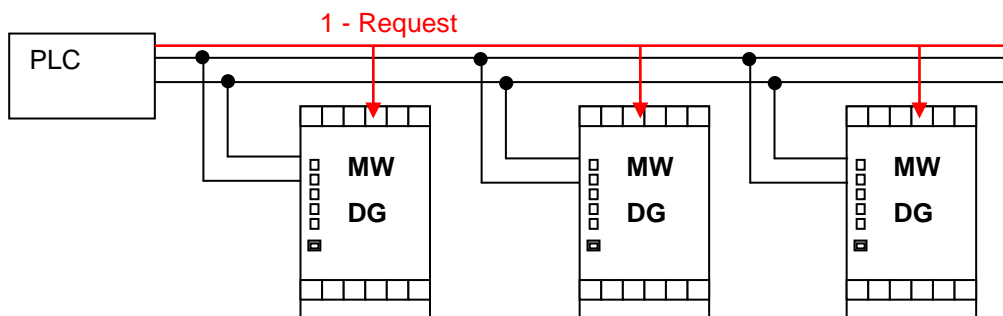
	RS232	RS423	RS422	RS485
Differential	no	no	yes	yes
Max number of drivers	1	1	1	32
Max number of receivers	1	10	10	32
Modes of operation	half duplex full duplex	half duplex	half duplex	half duplex
Network topology	point-to-point	multidrop	multidrop	multipoint
Max distance (acc. standard)	15 m	1200 m	1200 m	1200 m
Max speed at 12 m	20 kbs	100 kbs	10 Mbs	35 Mbs
Max speed at 1200 m	(1 kbs)	1 kbs	100 kbs	100 kbs
Max slew rate	30 V/ $\mu$ s	adjustable	n/a	n/a
Receiver input resistance	3..7 k $\Omega$	$\geq$ 4 k $\Omega$	$\geq$ 4 k $\Omega$	$\geq$ 12 k $\Omega$
Driver load impedance	3..7 k $\Omega$	$\geq$ 450 $\Omega$	100 $\Omega$	54 $\Omega$
Receiver input sensitivity	$\pm$ 3 V	$\pm$ 200 mV	$\pm$ 200 mV	$\pm$ 200 mV
Receiver input range	$\pm$ 15 V	$\pm$ 12 V	$\pm$ 10 V	-7..12 V
Max driver output voltage	$\pm$ 25 V	$\pm$ 6 V	$\pm$ 6 V	-7..12 V
Min driver output voltage (with load)	$\pm$ 5 V	$\pm$ 3.6 V	$\pm$ 2.0 V	$\pm$ 1.5 V

Solo il master può iniziare una transazione; quest'ultima può essere sia di tipo unicast, dove si ha l'invio di un comando da parte del master e si attende una risposta da parte dello slave, sia di tipo broadcast in cui il messaggio viene spedito dal master a tutti i dispositivi slave sulla linea i quali non danno alcuna risposta.

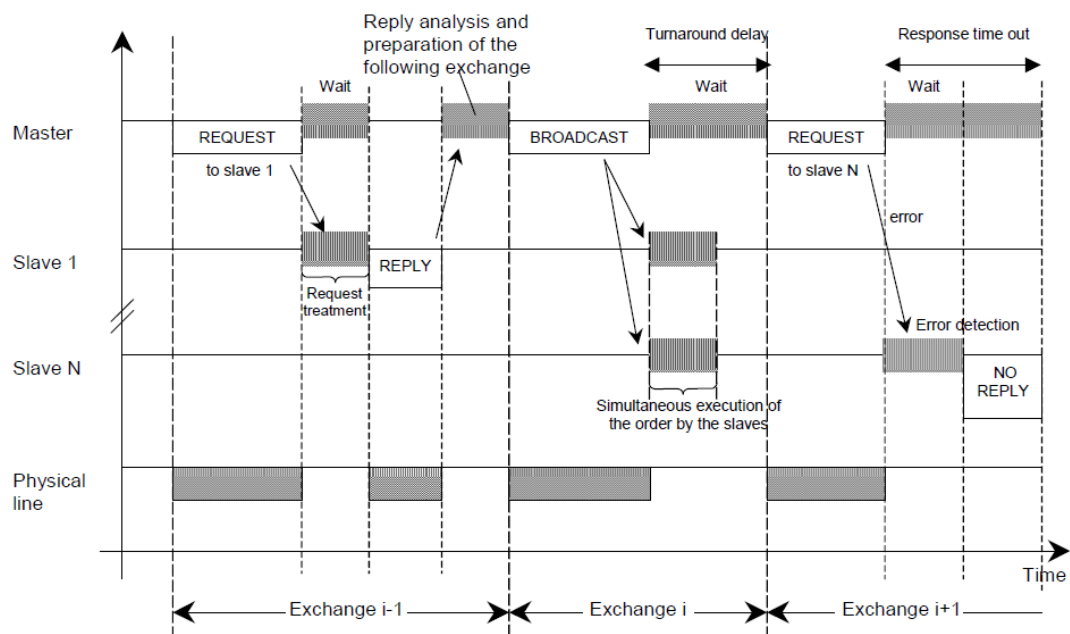
#### Esempio di comunicazione unicast.



**Esempio di comunicazione broadcast, in questo caso l'indirizzo Modbus 0 serve agli slave per identificare un messaggio di tipo broadcast.**



**Esempio di diagramma temporale di trasmissione di tipo unicast e broadcast**



**Tempi di risposta:**

- Request treatment circa 10 ms.
- Con un baudrate di 9600, si ha un tempo di trasmissione di circa 1 ms per byte trasmesso, quindi la durata del tempo di trasmissione dei pacchetti di REQUEST, REPLY e BROADCAST dipende dal numero di registri che si vogliono leggere o scrivere in quanto modificano la lunghezza del frame trasmesso.
- Nonostante i tempi di risposta siano nell'intorno di qualche decina di millisecondi si raccomanda l'utilizzo di tempi di timeout nell'ordine di qualche secondo.

## 2.2. Modalità RTU.

La modalità di comunicazione scelta per comunicare con la **basestation** è esclusivamente di tipo RTU, in quanto, rispetto alla modalità ASCII, risulta più efficiente in termini di rapporto quantità di informazione trasferita per byte.

Questa modalità utilizza un sistema per il controllo dell'errore di tipo CRC16 ANSI (*Cyclical Redundancy Check*) definito in Wikipedia al seguente URL:

[http://en.wikipedia.org/wiki/Cyclic\\_redundancy\\_check](http://en.wikipedia.org/wiki/Cyclic_redundancy_check).

Nell' "Appendice A" di questo documento vi è l'algoritmo scritto in ANSI C della procedura di verifica ed inserzione del CRC.

## 2.3. Frame RTU.

Nella modalità RTU i messaggi (*frames*) sono delimitati da un intervallo di silenzio della durata del tempo di trasmissione di un carattere circa moltiplicato per 3,5; naturalmente questo intervallo dipende dal baudrate utilizzato.

Il primo campo trasmesso riguarda l'indirizzo ModBus del dispositivo con cui si vuole parlare.

I caratteri trasmessi successivamente, per tutto il resto del frame, devono assolutamente essere inviati consecutivamente per poi concludersi con l'intervallo di silenzio per decretare la fine del messaggio.

Un generico frame RTU è costituito nel modo seguente:

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1-T2-T3-T4	8 BITS	8 BITS	$n \times 8$ BITS	16 BITS	T1-T2-T3-T4

Dove:

- START e END sono gli intervalli di silenzio (*tempo di trasmissione di 4 caratteri*)
- ADDRESS è l'indirizzo del dispositivo di destinazione
- FUNCTION è il comando
- DATA è il payload del comando o dati di risposta
- CRC16 è il sistema del controllo dell'errore.
- T1-T2-T3-T4 è il tempo di silenzio di interframe (*equivalente al tempo di trasmissione di circa 4 byte = 3.5 ms con baudrate di 9600*), questi tempi di attesa sono necessari per identificare un frame dal precedente.

## 2.4. Indirizzamento porta PLC del dispositivo.

L'indirizzo di default è il valore 17 (*0x11 in esadecimale*), il quale può essere modificato dal software **WineCapManager** fornito in dotazione (*vedi WineCapManager Manuale Utente*).

Gli indirizzi validi hanno un range di valori che va dal valore 1 al valore 247, mentre il valore 0 è riservato alle sole trasmissioni broadcast e utilizzato solo da parte del master.



## 2.5. Funzioni ModBus abilitate.

Esistono due funzioni in grado di leggere e scrivere uno o più registri:

- In lettura, Function = 0x03 (**Read Holding Register**)
- In scrittura, Function = 0x10 (**Preset Multiple Register**)

Entrambe le funzioni si riferiscono a dati di tipo Holding Register (registri = 4xxxx) dalla dimensione fissa a 16 bit.

La conversione da numero di registro a indirizzo su 16bit avviene sottraendo 40001 (*offset è posto a 0*), per esempio il registro 40108 equivale all'indirizzo 107 della classe holding register (4xxxx).

### 2.5.1. Read Holding Register (0x03).

Comando di lettura di uno o più registri consecutivi.

**Frame del comando di lettura dei registri (lato PLC):**

Slave Address	Function = 0x03	Start Address HIGH	Start Address LOW	Numb Register HIGH	Numb Register LOW	CRC16 LOW	CRC16 HIGH
---------------	-----------------	--------------------	-------------------	--------------------	-------------------	-----------	------------

**Esempio: lettura dei registri da 45001 a 45002 (2 registri consecutivi).**

Frame di comando in HEX = **11 03 13 88 00 02 42 35**

Dove:

- 0x11 = 17<sub>DEC</sub> indirizzo Modbus
  - 0x03 = 3<sub>DEC</sub> funzione Read Holding Register (4xxxx)
  - 0x1388 = 5000<sub>DEC</sub> indirizzo di partenza (5000)
  - 0x02 = 2<sub>DEC</sub> numero registri consecutivi da leggere
  - 0x4235 = CRC16 (0x42 parte LSB, 0x35 parte MSB)
- } Registro MB 45001

**Frame di risposta lettura dei registri (lato basestation):**

Slave Address	Function = 0x03	Byte Count	Data 1 HIGH	Data 1 LOW	.....	Data n HIGH	Data n LOW	CRC16 LOW	CRC16 HIGH
---------------	-----------------	------------	-------------	------------	-------	-------------	------------	-----------	------------

**Esempio: risposta delle letture dei registri da 45001 a 45002 (risposta esempio precedente)**

Frame di risposta in HEX = **11 03 04 08 B5 08 D8 FF EE**

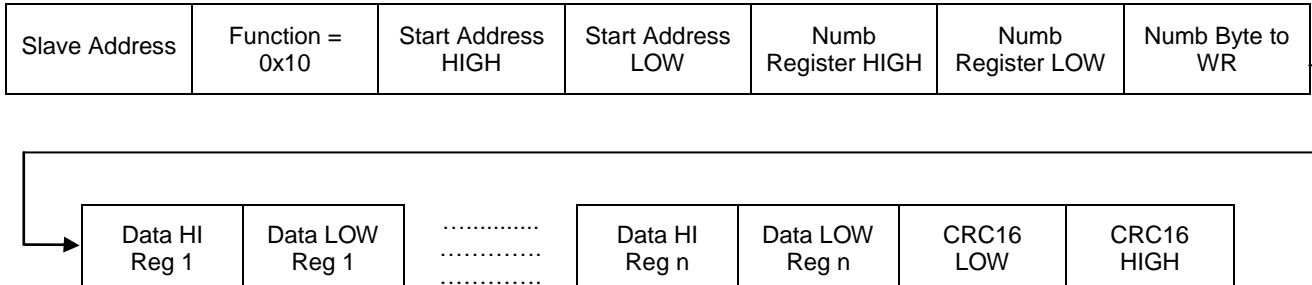
Dove:

- 0x11 = 17<sub>DEC</sub> indirizzo Modbus
- 0x03 = 3<sub>DEC</sub> funzione Read Holding Register (4xxxx)
- 0x04 = 4<sub>DEC</sub> numero di byte del payload di risposta, esso è sempre il doppio del numero dei registri poiché un registro ha dimensione 16 bit.
- 0x08B5 = 2229<sub>DEC</sub> valore del primo registro (08 parte MSB, B5 parte LSB).
- 0x08D8 = 2264<sub>DEC</sub> valore del secondo registro (08 parte MSB, D8 parte LSB).
- 0xFFEE = CRC16 (0xFF parte LSB, 0xEE parte MSB)

## 2.5.2. Preset Multiple Register (0x10).

Comando di scrittura di uno o più registri consecutivi.

**Frame del comando di scrittura dei registri (lato PLC):**



**Esempio: scrittura sui registri da 45101 a 45102 (2 registri consecutivi).**

Frame di comando in HEX = **11 10 13 EC 00 02 04 00 00 00 00 70 42**

Dove:

- 0x11 = 17<sub>DEC</sub> indirizzo ModBus
- 0x10 = 16<sub>DEC</sub> funzione Preset Multiple Register (4xxxx)
- 0x13EC = 5100<sub>DEC</sub> indirizzo del registro di partenza (PS: 5100 equivale al registro 45101 in quanto l'offset del primo registro parte dal valore 0 e non 1).
- 0x0002 = 2<sub>DEC</sub> numero di indirizzi consecutivi da scrivere.
- 0x02 = Totale byte da scrivere (= 2 x numero di registri).
- 0x0000 = valore scrittura del registro 1
- 0x0000 = valore scrittura del registro 2
- 0x7042 = CRC16 (0x70 parte LSB, 0x42 parte MSB)

**Frame di risposta lettura dei registri (lato WLI - M):**

Slave Address	Function= 0x10	Start Address HIGH	Start Address LOW	Numb Register HIGH	Numb Register LOW	CRC16 LOW	CRC16 HIGH
------------------	-------------------	--------------------------	-------------------------	--------------------------	-------------------------	--------------	---------------

**Esempio: risposta delle letture dei registri da 45101 a 45102 (risposta esempio precedente)**

Frame di risposta in HEX = **11 10 13 EC 00 02 86 29**

Dove:

- 0x11 = 17<sub>DEC</sub> indirizzo Modbus
  - 0x10 = 16<sub>DEC</sub> funzione Preset Multiple Register (4xxxx)
  - 0x13EC = 5100<sub>DEC</sub> indirizzo di partenza.
  - 0x02 = 2<sub>DEC</sub> numero registri consecutivi scritti.
  - 0x8629 = CRC16 (0x86 parte LSB, 0x29 parte MSB)
- } Registro MB 45101

## 2.6. Gestione delle eccezioni.

Nel caso in cui si chiedesse di scrivere su un registro di sola lettura oppure leggere un registro non esistente il dispositivo risponderà con un frame di eccezione, dalla seguente struttura:

Slave Address	Request Function + 0x80	Type Exception	CRC16 LOW	CRC16 HIGH
---------------	----------------------------	----------------	--------------	---------------

### **Esempio: Scrittura dei registri 41201 e 41202**

Frame di comando inviato dal master: **11 10 04 B0 00 02 04 00 00 00 00 9E DB**

Frame ricevuto da parte dello slave: **11 90 03 0D C4**

Dove:

- 0x11 = 17<sub>DEC</sub> indirizzo ModBus dello slave.
- 0x90 = 0x80 + 0x10 = eccezione riguardante il comando di scrittura (*0x10 Preset Multiple Register*).
- 0x03 = Tipo di eccezione.
- 0x0DC4 = CRC16 (*0x0D parte LSB, 0xC4 parte MSB*)

Il campo tipo di eccezione può contenere i seguenti valori:

Valore	Nome eccezione	Implementazione	Significato
0x01	ILLEGAL FUNCTION	Si	Funzione illegale, il dispositivo è in grado di rispondere solo alle funzione 0x03 e 0x10, le altre produrranno questa eccezione.
0x02	ILLEGAL DATA ADDRESS	Si	Indirizzo di registro non valido.
0x03	ILLEGAL DATA VALUE	Si	Valore del dato illegale, solitamente quando si tenta di scrivere su un registro un dato fuori dal range di esistenza.

### 3. Registri ModBus.

Il protocollo **ModBus** implementato all'interno delle **basestation** definisce due gruppi di registri che permettono di leggere le informazioni provenienti dai **dataloggers** wireless e gestire l'impianto.

Il primo gruppo è composto da registri di **OUTPUT Programmabili**.

Utile per esportare i dati in modo semplice e nell'ordine desiderato, impostabile all'interno della **basestation**.

Il secondo gruppo è un set di registri di **CONFIGURAZIONE e STATO**.

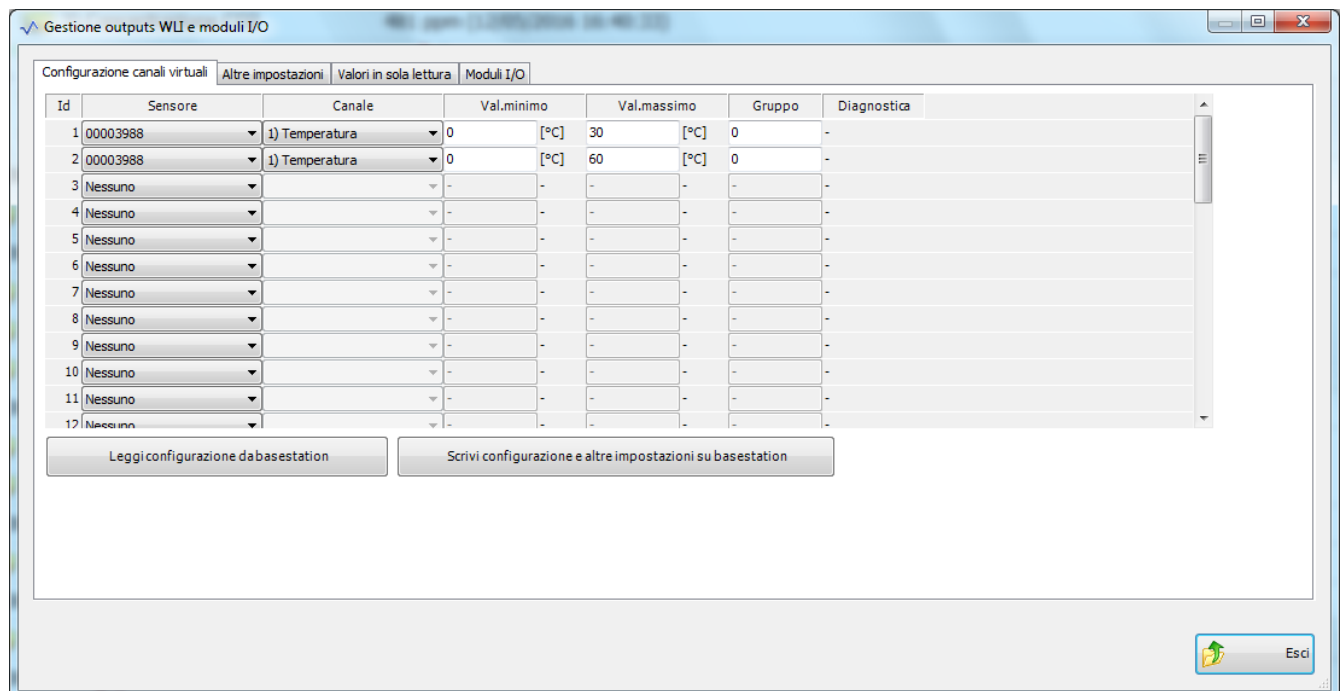
Questo set permette di accedere a informazioni più dettagliate, eseguire modifiche alla configurazione della **basestation** e scaricare anche i dati relativi alle misure pregresse, registrate all'interno della memoria della stessa.

#### 3.1. Registri di OUTPUT Programmabili

Sono registri in sola lettura, raggruppati in tabelle da 40 elementi, in cui sono pubblicate le grandezze e i valori di stato relativi ai **dataloggers**, nell'ordine definito e programmato all'interno della **basestation**.

Attraverso il software di gestione **WineCapManager** è possibile definire quali **dataloggers** e quali canali pubblicare sui vari registri.

La schermata di riferimento è raggiungibile al menù "BaseStation / Gestione Outputs WLI e Moduli I/O" (*fare riferimento al manuale WineCapManager*).



Le varie tabelle, in corrispondenza di una data posizione, esportano informazioni diverse del **datalogger** che è stato programmato per quella posizione.

Configurando sulle uscite i canali di misura in modo sequenziale, è possibile eseguire una lettura multipla all'interno della tabella per acquisire valori provenienti da più **dataloggers** in una sola transazione ModBus.

### 3.1.1. Tabella valori di misura.

Reg	Nome	Tipo	Dimensione (byte)	Significato
45001÷45040	Misura_Ch1 ÷ Misura_Ch40	Short	2	Valore misura uscita 1 ÷ Valore misura uscita 40

#### 3.1.1.1. Interpretazione dei valori restituiti:

- I valori di tali registri sono in formato Short ( 16 bit con segno ed in complemento a 2).
- La rappresentazione del valore dipende dal tipo di canale: nel caso della temperatura ambiente esso è rappresentato in centesimi di grado (il valore 2504 equivale a 25,04 °C).

Per l'umidità è in centesimi di percentuale relativa. Per conoscere gli altri casi riferirsi alla seguente tabella, tenendo conto della seguente formula:

$$\text{Valore} = (\text{Valore letto} + \text{offset}) \times \text{Scala};$$

Tipo Misura	Unità di misura
Temperatura	°C Range: -327,68 +327,67 (Scala = 0.01; offset = 0)
Umidità	%Rel Range: +0,00 +100,00 (Scala = 0.01; offset = 0)
Luce	Lux Range: +0 +10000 (Scala = 0.1; offset = +32768)
Deformazione	mm. Range: +0 + 25.000 (Scala = 0.001; offset = 0)
Concentrazione CO2	Ppm Range +0 +2000 (Scala =1; offset=0)
Energia Elettrica	KWh Range +0 +32768 (Scala =1; offset=0)
Energia Apparente	KVAh Range +0 +32768 (Scala =1; offset=0)
Volume fluido	L Range +0 +327680 (Scala =10; offset=0)
Energia Termica	KWh Range +0 +32768 (Scala =1; offset=0)
Inclinazione	° (sessagesimali) Range -1000 +1000 (Scala =100; offset=0)

### 3.1.2. Tabella valori di stato delle uscite.

Reg	Nome	Tipo	Dimensione (byte)	Significato
45301	Stato_Ch1	UShort	2	Valore dello stato uscita 1
÷	÷			÷
45340	Stato_Ch40			Valore dello stato uscita 40

#### 3.1.2.1. Interpretazione dei valori restituiti:

- I valori di tali registri sono in formato Short (16 bit con segno ed in complemento a 2).
- I valori hanno il seguente significato:
  - 0 = canale non configurato.
  - 1 = canale in stato di inizializzazione; tale stato permane sino al raggiungimento del primo valore di misura o al verificarsi di eventi di timeout che ne modificano lo stato quali lo stato di “missing” o di “fail”.
  - 2 = canale in stato di “online”, il canale è in funzionamento.
  - 3 = canale in stato di “missing”.
  - 4 = canale in stato di “fail”.

#### 3.1.2.2. Modalità Gestione “missing” e “fail”.

- In attesa della ricezione di un nuovo dato valido, la **basestation** replica sull'uscita l'ultimo dato valido pervenuto; trascorso un tempo TMAX1<sup>1</sup> senza dati validi pervenuti, **basestation** denomina il **datalogger** come “missing” e genera sull'uscita corrispondente l'ultimo valore letto correttamente;

se il **datalogger** “missing” trasmette un dato valido entro il tempo TMAX2<sup>1</sup>, dall'ultimo dato valido pervenuto, la **basestation** azzerà i timer e considera il **datalogger** in funzionamento normale;

se il **datalogger** “missing” non trasmette un dato valido entro un tempo TMAX2 dall'ultimo dato valido pervenuto, la **basestation** considera il **datalogger** come “fail” e porta tutte le uscite analogiche corrispondenti al quest'ultimo a 0V/0mA<sup>2</sup> mentre, via ModBus, mantiene l'ultimo valore letto correttamente;

la **basestation** fa uscire il **datalogger** dalla condizione “fail” al primo dato valido pervenuto dal quest'ultimo.

- la **basestation** rimane in condizione di normale funzionamento fino a quando il numero di **dataloggers** in condizione di “fail” contemporaneamente supera una soglia prefissata NKO<sup>1</sup>;

al superamento della soglia NKO la **basestation** entra a sua volta in modalità “fail”;

durante la condizione di allarme la **basestation** genera sulle uscite analogiche corrispondenti ai **dataloggers** in condizione di “fail” il dato 0V/0mA mentre pubblica su ModBus l'ultimo valore letto correttamente;

le uscite corrispondenti a **dataloggers** in normale funzionamento continuano a generare il valore corretto.

<sup>1</sup> Impostabile dal software “WineCapManager”.

<sup>2</sup> PWM = 0

### 3.1.3. Tabella contatori temporali delle uscite.

Reg	Nome	Tipo	Dimens (byte)	Significato
45501	Timer_Ch1	UShort	2	Valore temporale uscita 1
÷	÷			÷
45540	Timer_Ch40			Valore temporale uscita 40

#### 3.1.3.1. Interpretazione dei valori restituiti:

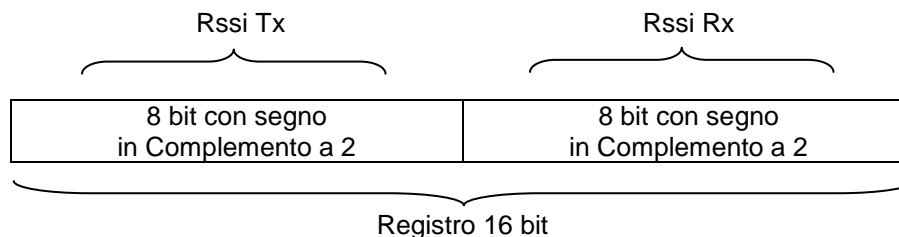
Contatore dei secondi passati dall'ultima misura ricevuta, tale valore si blocca sull'ultimo valore contabilizzabile (registro 16 bit senza segno, valore 65535) e mantiene tale valore sino a quando non arriva nuovamente una nuova misura (azzeramento).

### 3.1.4. Tabella valori di livello radio delle uscite.

Reg	Nome	Tipo	Dimens (byte)	Significato
45801	RF_Lev_Ch1	UShort	2	Valore livello radio uscita 1
÷	÷			÷
45840	RF_Lev_Ch40			Valore livello radio uscita 40

#### 3.1.4.1. Interpretazione dei valori restituiti:

Valore di entrambe le misure di Rssi (received signal strength indicator) in dBm, il registro ha il seguente formato:



dove per RSSI TX si intende il valore della potenza del segnale trasmesso, espresso in dBm, nella direzione **datalogger** → **basestation**,

mentre per Rssi Rx si intende il valore della potenza del segnale trasmesso, espresso in dBm, nella direzione **basestation** → **datalogger**.

Di seguito, tabella dei valori di RSSI con riferimenti soggettivi:

Giudizio	Visualizzazione Tacche	ReceivedSignal StrengthIndicator [dBm]	ReceivedSignal StrengthIndicator [%]
Ottimo	5 verdi	0 to -50	80-99
Buono	4 verdi	-51 to -67	60-79
Sufficiente	3 verdi	-68 to -85	40-59
Insufficiente	2 ambra	-86 to -91	20-39
Scarso	1 rossa	-92 to -120	0-19

### 3.1.5. Tabella dei valori di livello batteria delle uscite.

Reg	Nome	Tipo	Dimens (byte)	Significato
45401	Batteria_Ch1	UShort	2	Valore del livello batteria uscita 1
÷	÷			÷
45440	Batteria_Ch40			Valore del livello batteria uscita 40

#### 3.1.5.1. Interpretazione dei valori restituiti

Valori da 0 (batteria scarica) a 7 (batteria completamente carica).

### 3.1.6. Tabella degli stati relay.

Reg	Nome	Tipo	Dimens (byte)	Significato
45601	Relay_Ch1	UShort	2	Valore del Relay canale 1
÷	÷			÷
45614	Relay_Ch14			Valore del Relay canale 14

#### 3.1.6.1. Interpretazione dei valori restituiti:

Valori dei relay : 0 aperto, 1 chiuso.

in particolare:

- Relay CH1: (*normalmente chiuso*) si attiva quando un numero definito di **dataloggers** sono in "Fail".
- Relay CH2: (*normalmente chiuso*) si attiva quando almeno un valore batteria dei **dataloggers** registrati è al di sotto di un valore definito.

### 3.1.7. Tabella valori di stato soglie e validità misure.

Reg	Nome	Tipo	Dimens (byte)	Significato
45901	Allarmi_Ch1	UShort	2	Stato allarmi e invalidità uscita 1
÷	÷			÷
45940	Allarmi_Ch40			Stato allarmi e invalidità uscita 40

#### 3.1.7.1. Interpretazione dei valori restituiti:

Registro composto da bit di stato relativi al superamento delle soglie relative alla misura e lo stato di eventuale invalidità della stessa

Valore corrispondente alla somma dello stato dei singoli BIT all'interno. Fare riferimento alla seguente tabella:

#### Allarmi\_Chx:

CH Status (8 bit)							LVL Status (8 bit)								
x	x	x	x	x	Lvl Meas	CH Meas	Invalid	x	x	x	x	State Alarm	State Alarm	Max/Min	Event



- **Lvl Meas:** significa che il dato in memoria è un'acquisizione di verifica soglia eseguita dal **datalogger** all'intervallo di soglia prestabilito.
- **CH Meas:** significa che il dato in memoria è un'acquisizione di misura eseguita dal **datalogger** all'intervallo di campionamento prestabilito.
- **Invalid:** sensore guasto. Questo flag definisce la validità del campionamento. Il **datalogger** è in grado di invalidare la misura se rileva degli errori durante la fase di acquisizione.
- **State Alarm:** coppia di BIT che definisce, insieme a Max/Min, la condizione in cui si trova la grandezza.

State Alarm	
00	soglia rispettata
01	superamento intermedio ( <i>in caso di doppia soglia nella stessa direzione</i> )
10	superamento critico ( <i>negli altri casi</i> )
11	invalido ( <i>non si verifica mai</i> )

- **Max/Min:** definisce se l'eventuale superamento riguarda la soglia di massimo o di minimo.
- **Event:** indica che l'ultima informazione ricevuta riguardava una transizione di stato degli allarmi. In questo modo si possono discriminare le trasmissioni dovute al normale periodo di campionamento da quelle di ingresso in allarme o di allarme rientrato (*in quest'ultimo caso Event=1 e State Alarm=00*).

## 3.2. Registri di CONFIGURAZIONE e STATO

Attraverso questo set di registri è possibile accedere alle informazioni presenti all'interno della memoria della **basestation** e gestire l'impianto WSN.

I parametri relativi ai **dataloggers** configurati in memoria sono raggruppati in tabelle contenenti informazioni diverse.

Se una posizione in memoria non è occupata da nessun **datalogger** le tabelle conterranno valori invalidi in quella posizione.

### 3.2.1. Identificazione Datalogger

All'interno della mappa ModBus i singoli **dataloggers** sono identificati attraverso un indice (da 0 a 39) che rappresenta l'offset rispetto al registro base della singola proprietà.

*Esempio: il registro 900 rappresenta il livello batteria del **datalogger** con indice 0, il registro 939 rappresenta il livello batteria del **datalogger** con indice 1*

L'associazione tra l'indice del **datalogger** e il **datalogger** fisico può essere gestita indifferentemente in 2 modi:

- mediante ricerca automatica sul mac address  
Il mac address delle sonde (*rappresentato su di una doppia word*) viene pubblicato su ModBus attraverso una lista di 40 doppi registri. Scorrendo la lista è possibile determinare l'associazione tra indice e mac address
- mediante visualizzazione manuale dell'indice ModBus del **datalogger** attraverso **WineCapManager** (*menù proprietà del **datalogger***)

In generale, comunque, l'indice viene assegnato ai **dataloggers** nell'ordine con cui essi sono associati al sistema.

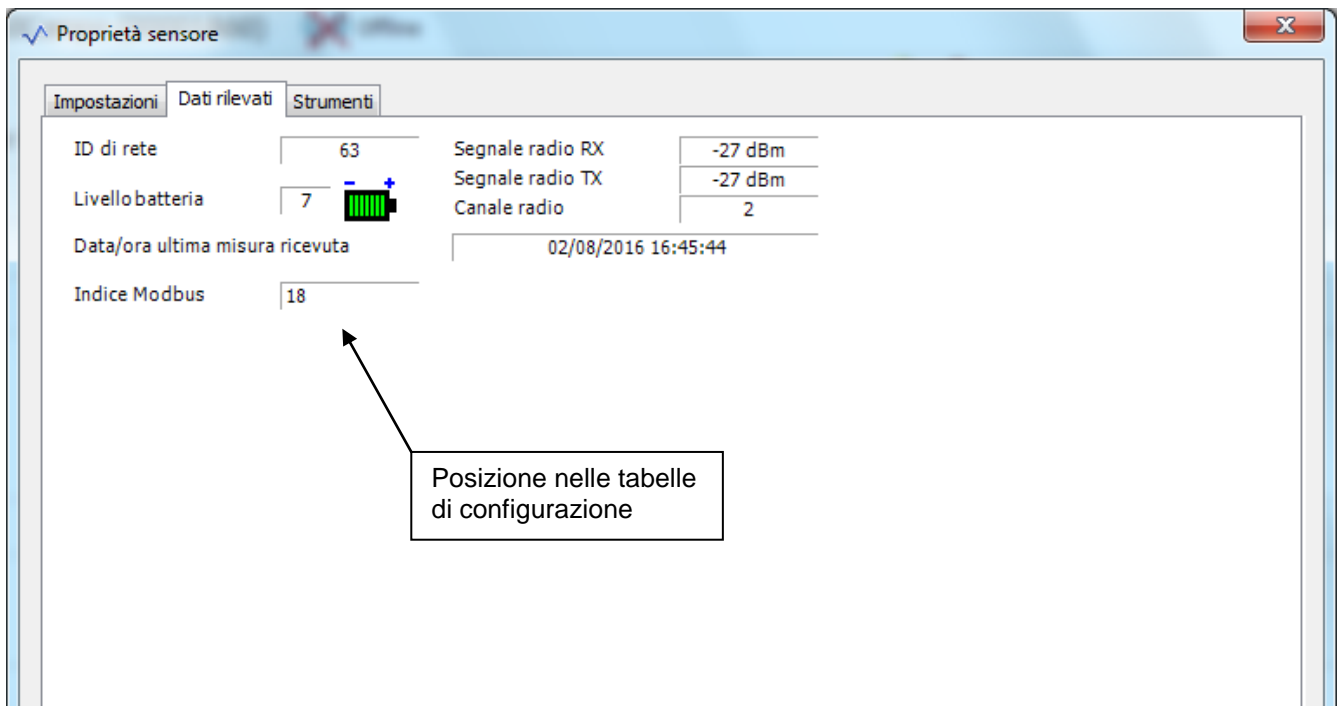
In caso di eliminazione di un **datalogger** dal sistema, gli indici dei **dataloggers** rimanenti non vengono modificati.

L'indice del **datalogger** eliminato verrà eventualmente assegnato al prossimo **datalogger** che verrà associato al sistema

Per sapere in che posizione sono salvati i **dataloggers** è possibile eseguire una scansione della tabella di elenco dei MAC address e avere la mappatura di tutti i **dataloggers** presenti.

In alternativa, volendo agire solo su un particolare **datalogger**, è possibile conoscerne la posizione leggendo l'indice ModBus presente nelle proprietà dello stesso.

Questo indice va utilizzato come offset da sommare al registro di inizio tabella. (*il primo valore è pari a zero*).



Alcuni registri permettono la scrittura per cui è possibile leggere e modificare le configurazioni dei singoli **dataloggers** attraverso di essi.

In questo caso, la scrittura del registro viene convertita dalla **basestation** in una trasmissione di nuova configurazione verso il **datalogger** relativo.

E' possibile anche tenere sotto controllo lo stato di allineamento tra la configurazione salvata e la conferma di ricevuta da parte del **datalogger**.

In ogni caso, l'allineamento dei parametri verso i **dataloggers** in campo è gestito dalla **basestation**.

In questo set di registri è possibile anche gestire e verificare i canali in scrittura dei **dataloggers** che presentano tale caratteristica.

Per esempio i canali relay dei **dataloggers** attuatore possono essere modificati e azionati attraverso la scrittura del valore del canale.

Anche in questo caso, come per le configurazioni, la propagazione del comando di attuazione viene eseguito in automatico dalla **basestation** e mantenuto allineato attraverso opportuni flag.

### 3.2.2. Accesso in lettura alle proprietà dei dataloggers

L'accesso in lettura a qualsiasi proprietà avviene con la semplice lettura del registro corrispondente.

Alcune proprietà di tipo "unsigned long" (es: *UTC ultima misura* o *MAC address del datalogger*) sono rappresentate attraverso una coppia di registri ModBus.

In questo caso il registro più piccolo rappresenta la word più significativa.

### 3.2.3. Accesso in scrittura alla configurazione dei dataloggers

Alcuni registri ModBus sono di tipo read/write per permettere all'utente di impostare la proprietà all'interno del **datalogger**. La sequenza con cui avviene la scrittura delle proprietà del **datalogger** è la seguente:

- l'utente imposta attraverso una serie di scritture ModBus tutta o parte della configurazione desiderata (es: *tempo di campionamento, valore di soglia di un canale, etc*);
- quando l'utente ha terminato di configurare un datalogger invia, attraverso ModBus, il comando di "commit" della configurazione (*scrittura del bit 7 del registro SensorFlag a valore 1*);

- da quel momento la nuova configurazione è pronta per essere trasmessa dalla **basestation** al **datalogger**. La trasmissione avverrà automaticamente non appena si verifica una delle seguenti condizioni:
  - o il **datalogger** invia una misura;
  - o il **datalogger** invia un messaggio di “hertbeat” periodico (*ogni 10 minuti*);
- nel caso in cui il **datalogger** non dia conferma di aver ricevuto la nuova configurazione questa verrà riproposta automaticamente dalla **basestation** stessa al verificarsi delle condizioni indicate al punto precedente;
- non appena il datalogger conferma di aver ricevuto la nuova configurazione, la **basestation** azzerava automaticamente il bit di commit e sospende i tentativi di invio della configurazione al **datalogger**;

### 3.2.4. Accesso in scrittura ai canali dei dataloggers

I **dataloggers** che hanno dei canali con funzione di “attuatore” al posto di “sensore”, permettono la scrittura del valore del canale (*Measure*) da parte della porta ModBus.

In questo caso, la sequenza di aggiornamento del valore di un canale da parte della porta ModBus è la seguente:

- l'utente imposta sul registro corrispondente al canale da scrivere il valore desiderato;
- la **basestation** setta a 1 il bit “Write pending” sul registro “Flag” del canale;
- la **basestation** trasmette immediatamente il comando al **datalogger**;
- nel caso in cui il **datalogger** non dia conferma di aver ricevuto ed eseguito il comando, la **basestation** ritrasmette automaticamente il comando al verificarsi di una delle seguenti condizioni:
  - o il **datalogger** invia una misura;
  - o il **datalogger** invia un messaggio di “heartbeat” periodico (*ogni 10 minuti*);
- non appena il **datalogger** conferma di aver ricevuto il comando, la **basestation** azzerava automaticamente il bit di “Write pending” sul registro Flag del canale e sospende i tentativi di invio del nuovo valore del canale.

### 3.2.5. Tabella di configurazione gateway

In questa tabella sono riportati alcuni parametri di configurazione del gateway utili per la gestione dell'impianto:

Indirizzo iniziale	Numero Registri	Nome	Read/Write	Nota
1200	10	BRIDGE_LABEL	R/W	Etichetta Nome del Gateway (20 caratteri)
4503	2	UTC_Time	R/W	Orologio Gateway in formato EPOCH time.

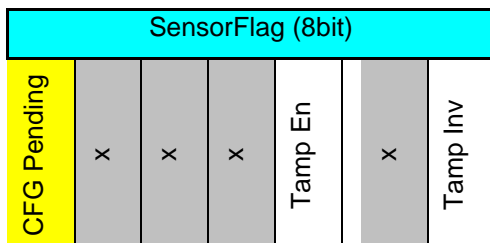
- **BRIDGE\_LABEL:** Stringa ordinata dal registro minore al registro maggiore, all'interno del registro il MostSignificantByte corrisponde al primo carattere e il LessSignificantByte al secondo carattere.
- **UTC Time:** scritto su due registri contigui in ordine: MostSignificantWord seguito da LessSignificantWord. Può essere scritto per regolare l'orologio dell'impianto.

### 3.2.6. Tabella di configurazione trasmettitori

In questa tabella sono riportati i parametri di configurazione dei trasmettitori presenti nell'impianto:

Indirizzo iniziale	Lunghezza elenco	Nome	Read/Write	Nota
0	40	Nch	R	Numero canali presenti su ogni trasmettitore
100	80	MAC	R	MAC address del trasmettitore
300	40	TimeMeas	R/W	Tempo di campionamento
400	40	TimeThreshold	R/W	Tempo di soglia allarme
500	40	SensorFlag	R/W	Flag inizializzazione invio configurazione al sensore corrispondente ( <i>vedi dettaglio</i> )
600	40	NetStatus	R	Variabile di stato della rete WSN
700	40	RSSI_tx	R	Forza segnale ricevuto
800	40	RSSI_rx	R	Segnale lato trasmettitore
900	40	BattLevel	R	Livello batteria (0-7)
1300	80	UTC_Meas	R	Ora del timer ultima misura ricevuta ( <i>vedi dettaglio</i> )

- **MAC:** MAC address del trasmettitore associato a quella posizione (valido per tutte le tabelle successive) scritto su due registri contigui in ordine: MostSignificantWord seguito da LessSignificantWord.
- **TimeMeas e TimeThreshold:** tempi di campionamento misure e di controllo soglie. Espressi in secondi, max 9 ore
- **SensorFlag**



- CFG Pending: Bit di segnalazione configurazione modificata non ancora aggiornata verso la sonda trasmettitore. Porre a 1 quando si vuole inviare la configurazione alla sonda e attendere l'azzeramento per conferma
- Tamp en/Tamp inv: gestione di sensore anti tamper

- **NetStatus:**

0	Canale non configurato. Nessun trasmettitore associato.
1	Canale nello stato di inizializzazione, tale stato permane sino al raggiungimento del primo valore di misura o al verificarsi di eventi di timeout che ne modificano lo stato quali lo stato di "warning" o di "fail".
2	Canale in stato di "online", il canale è in funzionamento.
3	Canale in stato di "warning".
4	Canale in stato di "fail"

- **UTC\_Meas:** E' in formato EPOCH time, scritto su due registri contigui (*MostSignificantWord* seguito da *LessSignificantWord*).

### 3.2.7. Tabelle di configurazione dei canali di misura

In questa tabella sono riportati i parametri di configurazione dei canali presenti a bordo di ogni trasmettitore.

Ogni trasmettitore può avere fino a 4 canali, quindi le tabelle sono uguali e ripetute per ogni canale. Gli indirizzi iniziali di ogni tabella sono i seguenti:

Canale 0	indirizzo 2100
Canale 1	indirizzo 3100
Canale 2	indirizzo 7100
Canale 3	indirizzo 8100

#### 3.2.7.1. Tabella configurazione canale

Offset di indirizzo	Lunghezza elenco	Nome	Read/Write	Nota
+0	40	ChType	R	Tipo canali
+100	40	ChOffset	R	Offset di canale
+200	40	ChGainBase	R	Guadagno fondo scala
+300	40	ChGainExp	R	Esponente guadagno
+400	40	ChThresConf	R/W	Flags di configurazione canale e settaggio soglie
+500	40	S1	R/W	Soglia1
+600	40	S2	R/W	Soglia2
+700	40	F1	R/W	Following1
+800	40	F2	R/W	Following2

- **ChType:** tipo di misura di canale

• **ChThresConf:**

CHx ThresConf (8 bit)							
Follow Available	Meas Enable	Follow Enable	Send All	En S2	type S2	En S1	Type S1

- **Follow Available:** impostazione per eseguire l'inseguimento misura disponibile
- **Follow Enable:** porre a 1 per eseguire l'inseguimento misura
- **Meas Enable:** abilitazione del canale di misura
- **Send All:** invio di tutte le misure quando in stato di allarme
- **En S1 e S2:** abilitazione delle Soglie 1 e 2
- **Type S1 e S2:** Tipo di soglia → 1=superiore, 2= inferiore
- **S1 e S2:** Valori di soglia. Su 16bit, valore al netto di ChGain e ChOffset
- **F1 e F2:** Valori di inseguimento rispetto con vecchio valore.  
F1=delta di confronto positivo e F2 negativo. Valore al netto di ChGain e ChOffset

### 3.2.8. Tabella Misure

Per quanto riguarda i valori di misura dei canali essi possono essere in lettura o in scrittura in base allo stato del flag leggibile sul registro ChThresConf

Indirizzo	Lunghezza elenco	Nome	Read/Write	Nota
6200	40	CH0 Meas	R/W	Valore di misura
6300	40	CH0 MeasFlag	R	Flag di soglia superata, test in corso, etc
6400	40	CH1 Meas	R/W	Valore di misura
6500	40	CH1 MeasFlag	R	Flag di soglia superata, test in corso, etc
6600	40	CH2 Meas	R/W	Valore di misura
6700	40	CH2 MeasFlag	R	Flag di soglia superata, test in corso, etc
6800	40	CH3 Meas	R/W	Valore di misura
6900	40	CH3 MeasFlag	R	Flag di soglia superata, test in corso, etc

- **CHx Meas:** Misura letta dalla sonda, su 16bit al netto della correzione  
Misura vera = (CHx\_Meas + ChOffset) \* ChGainBase \* 10 ^ ChGainExp
- **CHx MeasFlag:**

CH Status (8 bit)							LVL Status (8 bit)								
x	x	x	x	x	Lvl Meas	CH Meas	Invalid	WRT Pending	Writeable	x	x	State Alarm	State Alarm	Max/Min	Event

- **Lvl Meas:** significa che il dato in memoria è un'acquisizione di verifica soglia eseguita dalla sonda all'intervallo di soglia prestabilito.
- **CH Meas:** significa che il dato in memoria è un'acquisizione di misura eseguita dalla sonda all'intervallo di campionamento prestabilito.
- **Invalid:** Sensore guasto. Questo flag definisce la validità del campionamento. La sonda è in grado di invalidare la misura se rileva degli errori durante la fase di acquisizione.
- **WRT Pendig:** Bit di segnalazione di scrittura del canale modificata non ancora aggiornata verso la sonda trasmettitore. Viene posta a 1 quando si scrive sul canale CHx Meas un valore, e si azzerà in automatico quando si conclude l'allineamento con la sonda.
- **Writeable:** se il BIT=1 allora il canale è scrivibile. Questo significa che è possibile modificarne il valore scrivendo i registri del valore canale
- **State Alarm:** coppia di BIT che definisce, insieme a Max/Min, la condizione in cui si trova la grandezza.

Stati	
00	soglia rispettata
01	superamento intermedio (in caso di doppia soglia nella stessa direzione)
10	superamento critico (negli altri casi)
11	invalido (non si verifica mai)

- **Max/Min:** definisce se l'eventuale superamento riguarda la soglia di massimo o di minimo.
- **Event:** indica che l'ultima informazione ricevuta riguardava una transizione di stato degli allarmi. In questo modo si possono discriminare le trasmissioni dovute al normale periodo di campionamento da quelle di ingresso in allarme o di allarme rientrato (in quest'ultimo caso Event=1 e State Alarm=00).



#### 4. Specifica sul download dei dati di un datalogger dalla basestation

- 1) Registri modbus interessati alla procedura di download sono descritti nella seguente tabella (*sono tutti holding register*).

Per ulteriori informazioni sul protocollo di comunicazione ModBus riferirsi al documento dedicato.

Nome Funzione	Registro iniziale	Num Reg	R/W	Valori validi
DL MAC	4000	4	W/R	8 byte
DL UTC start	4004	2	W/R	4 byte
DL Payload 1	4100 - 4220	121	R	
DL Payload 2	4300 - 4420	121	R	

- 2) Premesso che il **datalogger** deve essere regolarmente registrato su EEPROM per eseguirne il download, l'algoritmo deve prevedere i seguenti passi:

- Scrivere sui registri 4000-4003 il MAC del **datalogger** interessato al download.
- Scrivere sui registri 4004-4005 il timestamp (*UTC UNIX time*) di partenza.
- Leggere alternativamente i registri [4100-4220] e [4300-4420] partendo dal primo intervallo; la lettura alternata permette al motore di download di avanzare sino al termine del flusso.

I registri 4100 e 4300 contengono un sequence number che avanzerà di una unità in base a tale alternanza.

- Per ogni lettura decodificare il frame di dimensione fissa di 240 byte (*120 registri Modbus successivi a quello di sequence number*) secondo l'organizzazione dei dati proposta di seguito (riferimento Protocollo zCap):

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
									CH1			CH2			CH3			CH4							
CheckSum	UTC				DEV Status	RSSI Rx	RSSI Tx	CH Status	LVL Status	Value	CH Status	LVL Status	Value	CH Status	LVL Status	Value	CH Status	LVL Status	Value	CH Status	LVL Status	Value			
1	4				1	1	1	1	1	2	1	1	2	1	1	2	1	1	2	1	1	2			

DEV Status (8bit)						
Bat	Bat	Bat	x	x	Off Line	Test
						Tamper

CH Status (8 bit)						
CH	CH	CH	x	x	Lvl Meas	CH Meas
						Invalid

Lvl Status(8 bit)						
x	x	x	x	Stato	Stato	Dir[Max = 1]
						Event

- La dimensione del singolo pacchetto di misura dipende dal numero dei canali realmente posseduti dal **datalogger** (*e non solo da quelli abilitati*); in base a quest'ultimo parametro tale dimensione resterà sino al completamento del download.

La dimensione è facilmente calcolabile in quanto vi è una parte fissa (*comune a tutti i dataloggers*) di 8 byte più 4 byte (*dimensione parte di canale di singolo*) moltiplicata per il numero di canali totali del **datalogger**.

Quindi: **datalogger** 1 CH  $\rightarrow 8 + (4 \times 1) = 12$  byte, **datalogger** 2 CH  $\rightarrow 8 + (4 \times 2) = 16$  byte ,  
**datalogger** 3 CH  $\rightarrow 8 + (4 \times 3) = 20$  byte e **datalogger** 4 CH  $\rightarrow 8 + (4 \times 4) = 24$  byte.

- f) La dimensione di 240 byte permette di essere un multiplo di 12,16,20 e 24 byte;  
infatti per ogni frame:
- **datalogger** 1 CH, 12 byte  $\rightarrow$  1 frame contiene 20 pacchetti
  - **datalogger** 2 CH, 16 byte  $\rightarrow$  1 frame contiene 15 pacchetti
  - **datalogger** 3 CH, 20 byte  $\rightarrow$  1 frame contiene 12 pacchetti
  - **datalogger** 4 CH, 24 byte  $\rightarrow$  1 frame contiene 10 pacchetti

In tal modo i 120 registri conterranno perfettamente un numero esatto di frame senza dover gestire un eventuale resto (*pezzi di frame tra il buffer di registri e quello successivo*).

- g) i **dataloggers** che non hanno tutti canali abilitati, lasceranno nel frame dei buchi (*i 4 byte riservati al canale avrà valore pari a 0xFF*) quindi facilmente identificabili.
- h) Il motore del download avanzerà con la lettura alternata dei registri sino a quando non raggiungerà la fine; a tal punto saranno ripetuti gli ultimi due frame (*lo si capisce in base al valore del sequence number il quale non incrementerà il proprio valore*) anche nel caso in cui il software dovesse continuare a chiedere alternativamente la serie dei due registri.

Notare che è possibile individuare l'ultimo pacchetto valido anche decodificando il buffer con payload utile in quanto l'ultimo frame (*con il sequence number più elevato*) ha valore 0xFF per tutto il suo contenuto; si consiglia, comunque, di utilizzare i registri di sequence number per identificare il termine del download.

- 3) Imporre il valore 0 al registro dell'UTC time significa eseguire il download di tutti i dati presenti sulla **basestation** del **datalogger** interessato.
- 4) Nel caso si volessero gestire download ripetuti si consiglia di registrare come timestamp per il download successivo l'ultimo timestamp prelevato sul payload da una misura di tipo OnLine (*flag Offline = 0*); in tal modo sarà possibile recuperare eventuali pacchetti offline dovuti ad una perdita momentanea del sincronismo tra il **datalogger** e la **basestation**.

Per ricapitolare:

- a) Partire con UTC time a zero per prelevare dall'inizio i dati di un **datalogger** ed eseguire il primo download.
- b) Durante il download aggiornare la variabile UTC time solo se all'interno del frame vi sono pacchetti di tipo "online" (*flag Offline = 0*), sino al completamento dell'download.
- c) Al successivo download, utilizzare la variabile UTC time aggiornata. In tal modo si evita di rileggere ad ogni riavvio tutti i dati letti nel download precedente.

## Appendice A: Calcolo del CRC16 ModBus

```

// -----
// calcolo e verifica del CRC 16
// del modbus standard RTU
// -----
// input:  Buffer[] = buffer da de/codificare
//         dataLength = lunghezza del buffer da de/codificare
//         check = flag (0 = inserisci CRC16, 1= controlla CRC16)
//
// output True se check=0 oppure check=1 è in check risulta positivo
// -----
-----
unsigned char MB_CRC16(unsigned char Buffer[],unsigned char dataLength,
                       unsigned char check) {

    unsigned short CheckSum;
    unsigned short j;
    unsigned char lowCRC;
    unsigned char highCRC;
    unsigned short i;

    CheckSum = 0xffff;
    for (j=0; j<dataLength; j++){
        CheckSum = CheckSum^(unsigned short)Buffer[j];
        for(i=8;i>0;i--){
            if ((CheckSum)&0x0001)
                CheckSum = (CheckSum>>1)^0xa001;
            else
                CheckSum>>=1;
        }
        highCRC = CheckSum>>8;
        CheckSum<<=8;
        lowCRC = CheckSum>>8;

        if (check==1){ // controlla CRC -----//
            if ((Buffer[dataLength+1] == highCRC) &
                (Buffer[dataLength] == lowCRC ))
                return 1;
            else
                return 0;
        } else { // inserisci CRC -----//
            Buffer[dataLength] = lowCRC;
            Buffer[dataLength+1] = highCRC;
            return 1;
        }
    }
}
// -----
// End -----
// -----

```



**Example: Request from holding Register 45301 to 45340 (Read Output State Values)**

MB Address = 0x10 = 16<sub>10</sub> (SW "WineCapManager" set)  
 MB Command = 0x03 (Read multiple holding register 4xxxx)  
 Start Register = 0x14B4 = 5300<sub>10</sub> (offset 1 → 5001)  
 Numb Register = 0x0028 = 40<sub>10</sub>  
 CrcCalc = 0x0343

Frame to TX: "10 03 14 B4 00 28 03 43"

Response: "10 03 50 00 02 00 02 00 02 00 02 00 02 00 02 00 02 00 02 00 02 00 02 00 04 00  
 04 00 02 00 02 00  
 00 51 96"

MB Address = 0x10  
 MB Command = 0x03 (Read multiple holding register 4xxxx)  
 Numb Byte Read = 0x50 = 80<sub>10</sub> (40 register x 2 byte)  
 Payload:  
     CH01 = 0x0002 = 2<sub>10</sub> = Online  
     CH02 = 0x0002 = 2<sub>10</sub> = Online  
     .....  
     CH40 = 0x0000 = 0<sub>10</sub> = not configured Channel.  
 CrcCalc = 5196