Modbus



Il protocollo Modbus permette lo scambio di dati tra dispositivi di automazione utilizzando una comunicazione seriale oppure Ethernet. Attraverso gli oggetti **Gateway Modbus** e **Gateway Modbus Slave**, il server Thinknx può leggere e scrivere valori Modbus, renderli disponibili alla supervisione e scambiarli con il sistema KNX.

Questa guida descrive:

- i concetti fondamentali necessari per interpretare la documentazione di un dispositivo Modbus;
- la configurazione del server Thinknx come Modbus Master;
- la creazione e l'utilizzo dei Datapoint;
- l'integrazione bidirezionale tra Modbus e KNX;
- l'utilizzo del server Thinknx come Modbus Slave;
- le procedure di verifica e troubleshooting.

Concetti fondamentali

Master e Slave

In una comunicazione Modbus, il **Master** avvia le richieste, mentre lo **Slave** risponde alle operazioni di lettura o scrittura ricevute. Uno Slave non avvia autonomamente una comunicazione verso il Master o verso altri Slave.

Quando viene utilizzato l'oggetto **Gateway Modbus**, il server Thinknx opera come Master e interroga uno o più dispositivi Slave. Può quindi leggere ciclicamente i loro dati oppure inviare comandi di scrittura.

Quando viene utilizzato l'oggetto **Gateway Modbus Slave**, il server Thinknx espone i propri Datapoint a un Master esterno, che potrà leggerli o scriverli secondo la configurazione effettuata.

Nelle implementazioni Modbus TCP vengono utilizzati anche i termini **Client** e **Server**. Dal punto di vista della comunicazione, il Client invia la richiesta e il Server risponde. Nella terminologia utilizzata dal Configurator, il Gateway Modbus Master svolge normalmente il ruolo di Client, mentre il dispositivo Slave svolge il ruolo di Server.

Aree dati Modbus

I dati Modbus sono organizzati in quattro aree principali.

Area dati	Dimensione	Accesso standard	Funzioni di lettura	Funzioni di scrittura
Coil	1 bit	Lettura e scrittura	1	5 oppure 15
Discrete Input	1 bit	Sola lettura	2	—
Input Register	16 bit	Sola lettura	4	—
Holding Register	16 bit	Lettura e scrittura	3	6, 16 oppure 22

Il tipo di area non viene determinato solamente dal numero visualizzato nel manuale del dispositivo. È la Function Modbus utilizzata a indicare se il server deve accedere, ad esempio, a una Coil oppure a un Holding Register.

Funzioni Modbus

Le principali funzioni utilizzabili nella configurazione sono riepilogate nella tabella seguente.

Function	Nome	Utilizzo
1	Read Coils	Legge lo stato di una o più Coil.
2	Read Discrete Inputs	Legge uno o più ingressi digitali.
3	Read Holding Registers	Legge uno o più Holding Register.
4	Read Input Registers	Legge uno o più Input Register.
5	Write Single Coil	Scrive lo stato di una singola Coil.
6	Write Single Holding Register	Scrive un singolo Holding Register.
15	Write Multiple Coils	Scrive più Coil consecutive.
16	Write Multiple Holding Registers	Scrive più Holding Register consecutivi.
22	Mask Write Register	Modifica bit selezionati all'interno di un Holding Register.

La funzione deve essere supportata dal dispositivo e deve corrispondere all'area dati indicata nella relativa documentazione Modbus.

Indirizzamento Modbus e offset Thinknx

I manuali dei dispositivi non utilizzano sempre lo stesso sistema di indirizzamento. Alcuni mostrano i registri partendo da **1**, mentre altri riportano direttamente l'indirizzo di protocollo partendo da **0**.

Il server Thinknx applica automaticamente un offset di **-1** all'indirizzo inserito nel Configurator.

```
Indirizzo trasmesso = Indirizzo inserito nel Configurator - 1
```

Indirizzo inserito nel Configurator	Indirizzo trasmesso dal server
1	0
10	9
100	99

Se il manuale utilizza una numerazione dei registri che parte da 1, normalmente il relativo numero deve essere inserito direttamente nel Configurator. Il server applicherà poi automaticamente l'offset.

Se il manuale riporta già l'indirizzo di protocollo partendo da 0, nel Configurator deve essere inserito il valore aumentato di 1. Ad esempio, per trasmettere l'indirizzo di protocollo 0 deve essere configurato l'indirizzo 1.

Le notazioni tradizionali **00001**, **10001**, **30001** e **40001** identificano anche l'area dati:

Notazione tradizionale	Area dati	Primo indirizzo di protocollo
00001	Coil	0
10001	Discrete Input	0
30001	Input Register	0
40001	Holding Register	0

Ad esempio, la notazione 40001 indica normalmente il primo Holding Register. Nel Datapoint Thinknx deve quindi essere selezionata la Function relativa agli Holding Register e deve essere configurato l'indirizzo numerico richiesto dal Configurator, senza utilizzare automaticamente l'intero valore 40001.

In questo caso, una configurazione tipica è:

Indicazione nel manuale	Indirizzo del dato nel Configurator	Indirizzo trasmesso
Holding Register 40001	1	0

Attenzione: prima di configurare tutti i Datapoint, verificare sempre la convenzione utilizzata dal produttore provando a leggere un singolo valore noto. Applicare manualmente un ulteriore -1 quando il server lo applica già porta alla lettura del registro precedente.

Conversione degli indirizzi esadecimali

Alcuni manuali Modbus riportano l'indirizzo dei Datapoint in formato esadecimale, normalmente identificato dal prefisso **0x**. Prima di inserirlo nel Configurator, l'indirizzo deve essere convertito nel corrispondente valore decimale.

La conversione può essere eseguita, ad esempio, tramite la **Calcolatrice di Windows** in modalità **Programmatore**:

- aprire la Calcolatrice e selezionare la modalità **Programmatore**;
- selezionare **HEX**;
- inserire l'indirizzo esadecimale senza il prefisso `0x`;
- leggere il valore corrispondente indicato alla voce **DEC**.

Ad esempio:

```
0x0010 = 16
```

La conversione da esadecimale a decimale non modifica la regola dell'offset Thinknx. Se **0x0010** rappresenta già l'indirizzo di protocollo, il valore decimale è **16** e nel Configurator deve essere inserito **17**, in modo che il server trasmetta l'indirizzo 16.

Indicazione nel manuale	Valore decimale	Indirizzo del dato nel Configurator	Indirizzo trasmesso dal server
0x0010	16	17	16 (0x0010)

Attenzione: la notazione esadecimale indica solamente la base numerica. Non permette di stabilire se il produttore utilizza una numerazione da 0 oppure da 1. Questa informazione deve essere verificata nel manuale del dispositivo o tramite una lettura di prova.

Informazioni necessarie

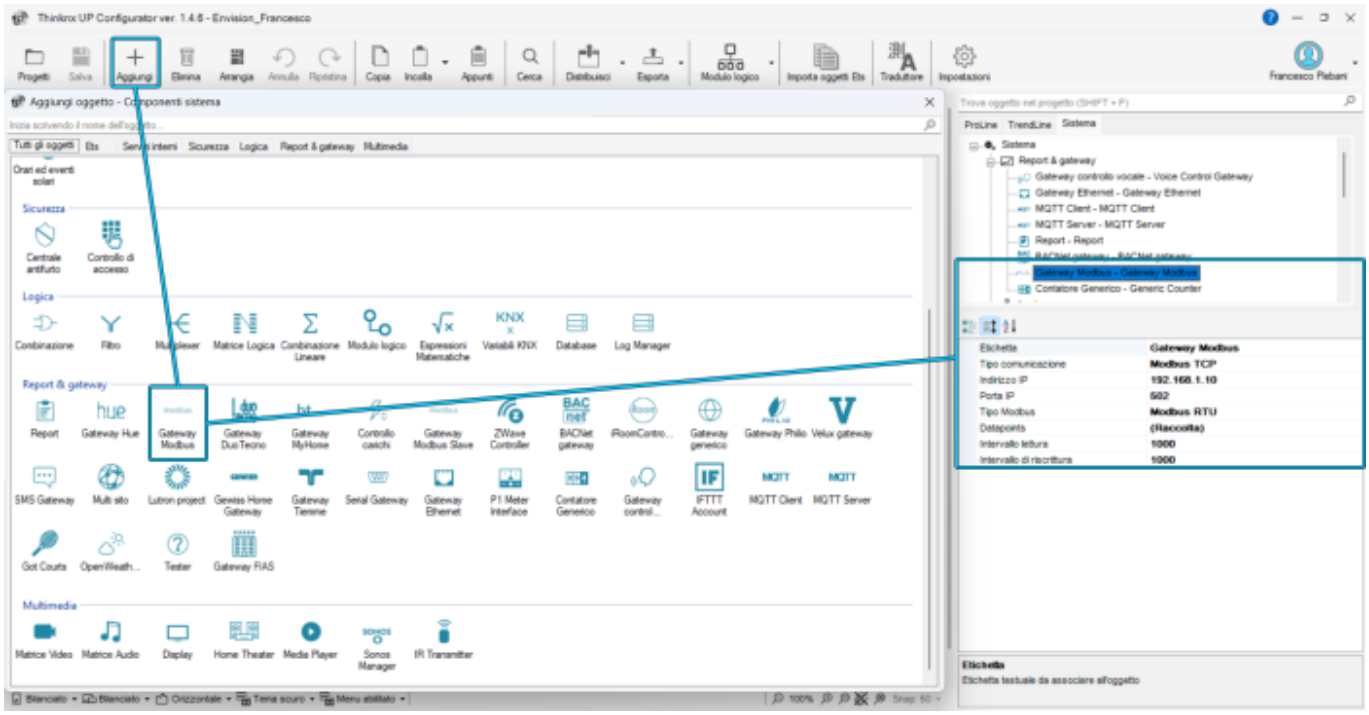
Prima di configurare il Gateway Modbus, recuperare il manuale o la tabella Modbus del dispositivo e identificare:

- ruolo del dispositivo;
- modalità di comunicazione: Modbus Seriale, Modbus Seriale su TCP oppure Modbus TCP;
- Slave Address o Unit ID;
- Baud Rate, Data Bits, Parità e Stop Bits per la comunicazione seriale;
- indirizzo IP e porta per la comunicazione Ethernet;
- indirizzo di ciascun dato;
- area dati e Function supportata;
- tipo e dimensione del valore;
- eventuale ordine dei registri o dei byte;
- eventuale fattore di scala;
- intervallo dei valori accettati in scrittura.

Verificare inoltre che sul server siano disponibili la porta di comunicazione necessaria e, quando prevista per il modello utilizzato, la relativa licenza Modbus.

È consigliabile verificare questi dati con un programma Modbus esterno prima di configurare l'intera integrazione in Thinknx.

Gateway Modbus – server Thinknx come Master



L'oggetto **Gateway Modbus** consente al server Thinknx di operare come Modbus Master. Il server interroga ciclicamente i dispositivi configurati e può scrivere i valori mediante i Datapoint, il gateway KNX oppure i comandi dell'oggetto.

Per aggiungere il gateway, accedere alla sezione **Sistema** del progetto, fare clic con il tasto destro sul nodo principale e aggiungere l'oggetto **Gateway Modbus** dalla categoria dedicata ai gateway.

Tipi di comunicazione

La proprietà **Tipo comunicazione** definisce il metodo utilizzato dal server.

Modbus Seriale

Questa modalità viene utilizzata per collegare direttamente il server a un bus seriale, normalmente tramite RS485.

- **Numero porta:** porta seriale del server utilizzata per la comunicazione. Il numero dipende dal modello di server.
- **Tipo Modbus:** variante Modbus utilizzata, ad esempio Modbus RTU.
- **Baud Rate:** velocità di comunicazione, che deve corrispondere a quella configurata sul dispositivo.
- **Data Bits:** numero di bit utilizzati per i dati.
- **Parità:** modalità di controllo della parità.
- **Stop Bits:** numero di bit di stop.
- **Timing comandi:** temporizzazioni utilizzate per ottimizzare lo scambio dati sul bus seriale. Possono dipendere dall'adattatore e dai tempi di risposta degli Slave.

Tutti i dispositivi presenti sullo stesso bus seriale devono utilizzare parametri di comunicazione compatibili e devono avere un indirizzo Slave univoco.

Modbus Seriale su TCP

Questa modalità viene utilizzata quando il bus seriale viene raggiunto attraverso un convertitore Ethernet/seriale. Il convertitore trasporta sulla rete la comunicazione seriale senza trasformarla in comunicazione Modbus TCP nativa.

- **Indirizzo IP:** indirizzo IP del convertitore seriale.
- **Porta IP:** porta TCP sulla quale il convertitore accetta la connessione.

La porta utilizzata deve corrispondere a quella configurata nel convertitore. Non deve essere necessariamente la porta standard 502.

Modbus TCP

Questa modalità viene utilizzata per comunicare direttamente con un dispositivo Modbus TCP.

- **Indirizzo IP:** indirizzo IP del dispositivo Modbus TCP.
- **Porta IP:** porta utilizzata dal dispositivo. La porta standard Modbus TCP è 502, ma il dispositivo può essere configurato diversamente.

Modbus Seriale su TCP e Modbus TCP non sono equivalenti. La modalità deve essere scelta in base al protocollo realmente supportato dal dispositivo o dal convertitore.

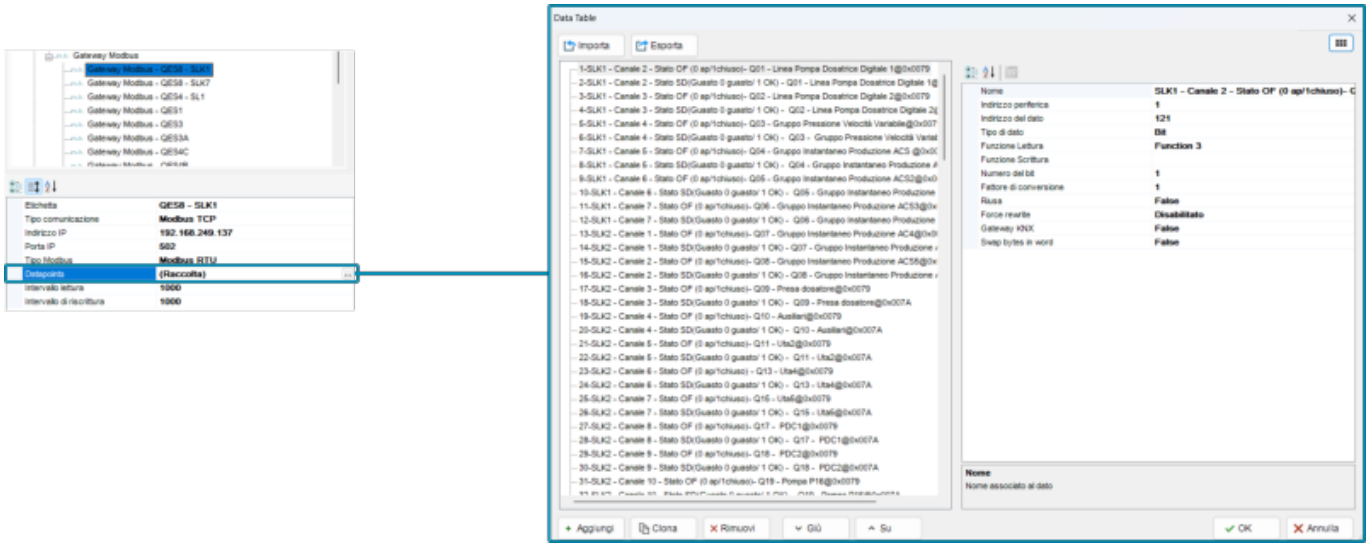
Parametri generali

- **Datapoints:** raccolta dei punti dati letti o scritti dal gateway.
- **Intervallo lettura:** pausa, espressa in millisecondi, tra due sequenze di polling consecutive.
- **Intervallo di riscrittura:** intervallo tra due sequenze di riscrittura. Viene utilizzato solamente dai Datapoint sui quali è abilitata la proprietà **Force rewrite**.

Un **Intervallo lettura** molto breve aumenta il traffico e può provocare timeout o instabilità, soprattutto su bus seriali con molti dispositivi. Iniziare con pochi Datapoint e con un intervallo prudente, quindi ridurlo solamente dopo avere verificato la stabilità della comunicazione.

Configurazione dei Datapoint Master

Aprire la raccolta **Datapoints**, aggiungere un nuovo elemento e configurare i parametri richiesti.



Identificazione e indirizzo

- **Nome:** nome utilizzato per identificare il Datapoint nel progetto.
- **Indirizzo periferica:** indirizzo del dispositivo Slave nel quale è disponibile il dato.
- **Indirizzo del dato:** indirizzo del dato. Il server trasmette il valore configurato applicando automaticamente l'offset -1.

Assegnare nomi descrittivi, ad esempio `Temperatura mandata`, `Allarme pompa` oppure `Setpoint zona 1`, così da riconoscere facilmente i Datapoint quando vengono selezionati da altri oggetti.

Tipo e funzioni

- **Tipo di dato:** tipo di dato da leggere o scrivere. Deve corrispondere alla dimensione e alla rappresentazione indicata dal produttore.
- **Funzione Lettura:** Function Modbus utilizzata per leggere il valore.
- **Funzione Scrittura:** Function Modbus utilizzata per scrivere il valore.

Un Input Register, ad esempio, viene normalmente letto con Function 4 e non può essere scritto. Un Holding Register viene normalmente letto con Function 3 e può essere scritto con una funzione supportata dal dispositivo, come Function 6 o Function 16.

Estrazione e conversione

- **Numero del bit:** identifica il bit del dato utilizzato dal Datapoint quando il valore richiesto è contenuto all'interno di un registro.
- **Fattore di conversione:** fattore per il quale viene moltiplicato il valore grezzo letto.

Ad esempio, se il dispositivo restituisce il valore 235 per rappresentare 23,5 °C, deve essere utilizzato un fattore di conversione pari a 0,1.

Prima di applicare il fattore di conversione, verificare che il valore grezzo e il segno siano interpretati correttamente. Un tipo di dato errato può produrre valori apparentemente casuali o molto lontani da

quelli attesi.

Riusa

La proprietà **Riusa** permette di riutilizzare una lettura già eseguita quando più Datapoint estraggono bit differenti dallo stesso indirizzo.

Ad esempio, se un Holding Register contiene sedici stati digitali, è possibile creare più Datapoint con gli stessi valori di **Indirizzo periferica** e **Indirizzo del dato**, modificando solamente **Numero del bit**:

- il primo Datapoint esegue la lettura con **Riusa** disabilitato;
- i Datapoint successivi utilizzano **Riusa** abilitato.

Il driver legge il registro una sola volta e utilizza il valore ricevuto per estrarre i diversi bit. Questo riduce il numero di richieste inviate allo Slave e migliora la velocità del polling.

Force rewrite

Quando **Force rewrite** è abilitato, il server riscrive ciclicamente l'ultimo valore inviato al Datapoint. La frequenza è definita dalla proprietà generale **Intervallo di riscrittura** del Gateway Modbus.

Questa funzione deve essere utilizzata solamente quando il dispositivo richiede un aggiornamento periodico del comando. Non abilitarla indiscriminatamente, perché aumenta il traffico e il numero di scritture inviate.

Gateway KNX

Abilitando **Gateway KNX**, il Datapoint può scambiare il valore con il bus KNX.

- **Tipo di dato KNX:** tipo di dato KNX associato al valore Modbus.
- **Gruppo KNX:** indirizzo di gruppo sul quale viene inviato il valore letto da Modbus.
- **Invia su cambio val.:** invia il valore sul bus KNX solamente quando il valore Modbus cambia.
- **Enable KNX to Modbus:** abilita la direzione di comunicazione da KNX verso Modbus.
- **Different KNX group:** indirizzo di gruppo dal quale il server riceve il valore da scrivere su Modbus.

Il flusso standard è il seguente:

```
Dispositivo Modbus → Gateway Modbus → Gruppo KNX di feedback
```

Abilitando la comunicazione bidirezionale viene aggiunto il flusso opposto:

```
Gruppo KNX di comando → Gateway Modbus → Dispositivo Modbus
```

Verificare che il **Tipo di dato KNX** sia compatibile con il valore Modbus e con l'eventuale **Fattore di conversione**.

Comandi del Gateway Modbus

I comandi dell'oggetto consentono di eseguire scritture Modbus richiamabili da scene, logiche, pulsanti o altri oggetti Thinknx.

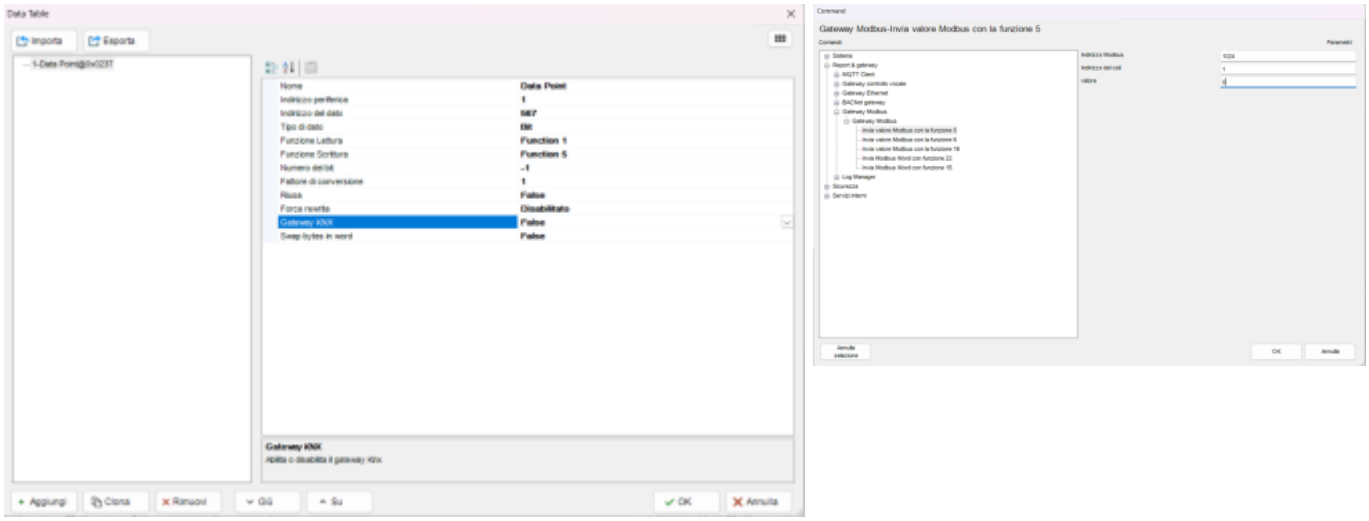
Function 5 – Write Single Coil

Scrive lo stato di una singola Coil.

Nel Configurator il comando è denominato **Invia valore Modbus con la funzione 5** e utilizza i seguenti parametri:

- **Indirizzo Modbus:** indirizzo dello Slave.
- **Indirizzo del coil:** indirizzo della Coil.
- **valore:** valore da scrivere, 0 = OFF oppure 1 = ON.

Il dispositivo deve supportare la Function 5 sull'indirizzo selezionato.



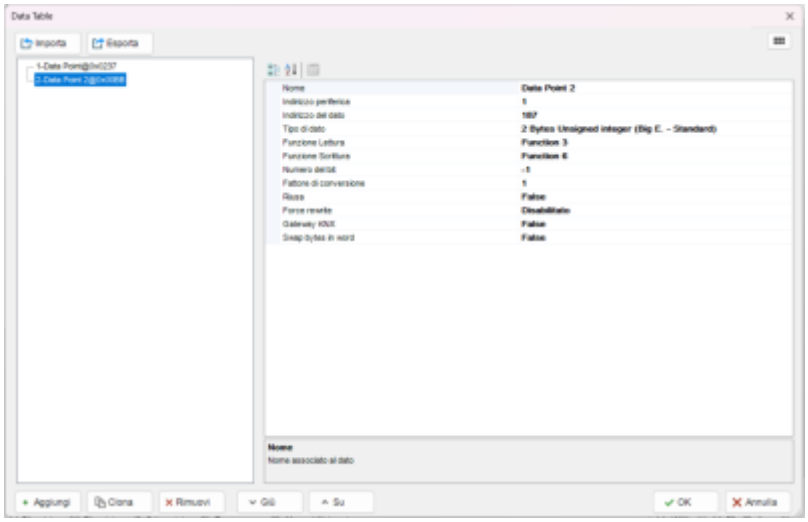
Function 6 – Write Single Holding Register

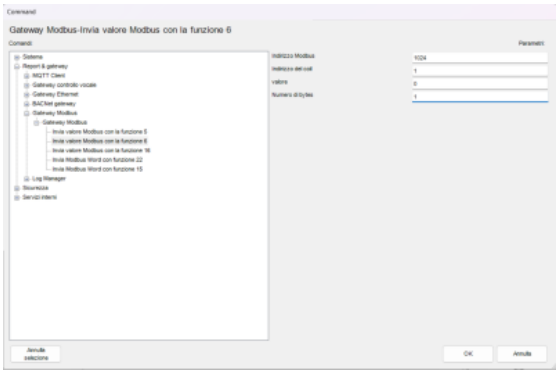
Scrive un singolo Holding Register.

Nel Configurator il comando è denominato **Invia valore Modbus con la funzione 6** e utilizza i seguenti parametri:

- **Indirizzo Modbus:** indirizzo dello Slave.
- **Indirizzo del coil:** indirizzo del registro da scrivere.
- **valore:** valore da scrivere.
- **Numero di bytes:** numero di byte configurato per il valore.

A livello Modbus, la Function 6 opera su un singolo registro da 16 bit. Nonostante il parametro sia denominato **Indirizzo del coil** nel Configurator, per questa funzione identifica l'indirizzo dell'Holding Register da scrivere. Verificare nel manuale del dispositivo il formato e l'intervallo del valore accettato.





Function 15 – Write Multiple Coils

La Function 15, conosciuta anche come **Write Multiple Coils**, permette di scrivere contemporaneamente lo stato di più Coil Modbus consecutive.

Attenzione: per richiamare la Function 15 dalla supervisione, utilizzare un **comando generico** associato al comando **Invia Modbus Word con funzione 15**. Non utilizzare un oggetto **Lampadina**, poiché la Function 15 deve ricevere tutti i parametri necessari per scrivere più Coil consecutive e non gestisce un semplice comando ON/OFF su una singola Coil.

Ogni Coil rappresenta un singolo bit:

- **0 = OFF**
- **1 = ON**

Le Coil vengono scritte a partire dall'indirizzo specificato in **Indirizzo del primo registro**. Il software utilizza un unico valore decimale per definire lo stato delle Coil e lo converte internamente in binario. Il bit meno significativo corrisponde alla prima Coil, il bit successivo alla seconda e così via.

Per impostare su ON le prime 10 Coil:

```
1111111111 = 1023
```

Per impostare su ON le prime 20 Coil:

```
11111111111111111111 = 1048575
```

In generale, per impostare su ON le prime N Coil consecutive:

```
Valore = 2^N - 1
```

Numero di Coil	Valore decimale
8	255
10	1023
16	65535

Numero di Coil	Valore decimale
20	1048575

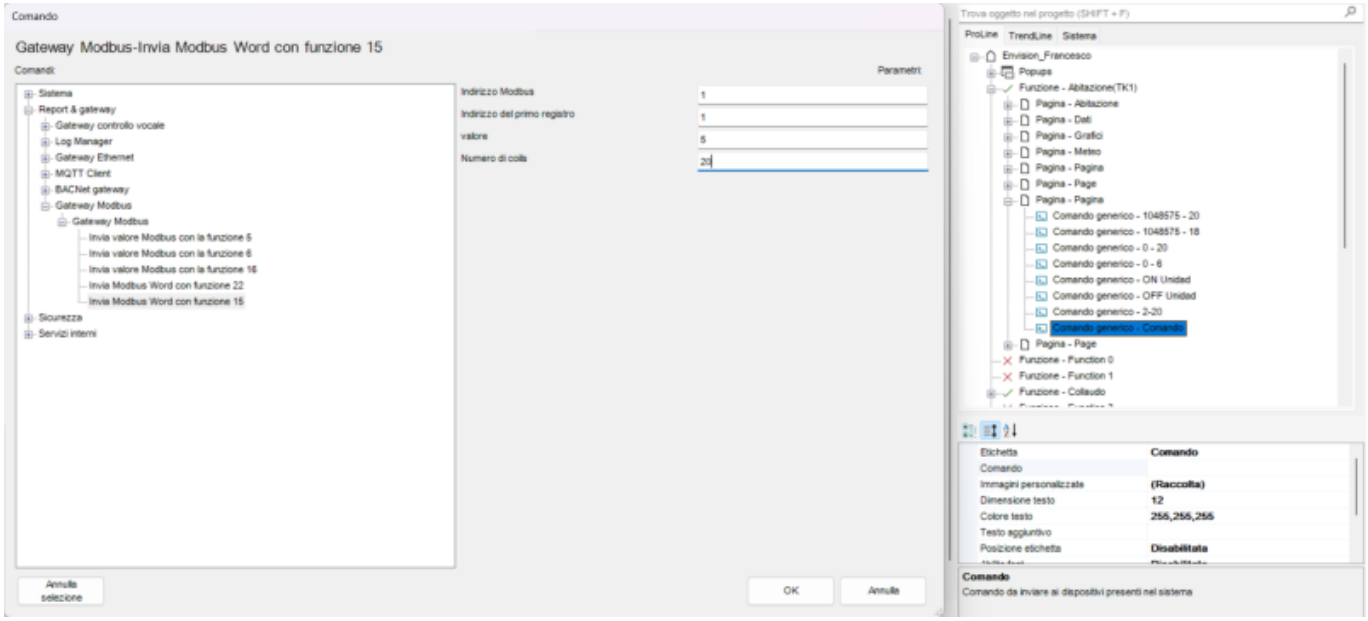
È possibile controllare anche Coil singole impostando solamente i bit corrispondenti. Ad esempio:

```
00000101 = 5
```

Il valore 5 imposta su ON la prima e la terza Coil, lasciando le altre su OFF.

Parametri del comando:

- **Indirizzo Modbus:** indirizzo dello Slave.
- **Indirizzo del primo registro:** indirizzo della prima Coil da scrivere.
- **valore:** valore decimale che rappresenta gli stati delle Coil.
- **Numero di coils:** numero di Coil consecutive da scrivere.



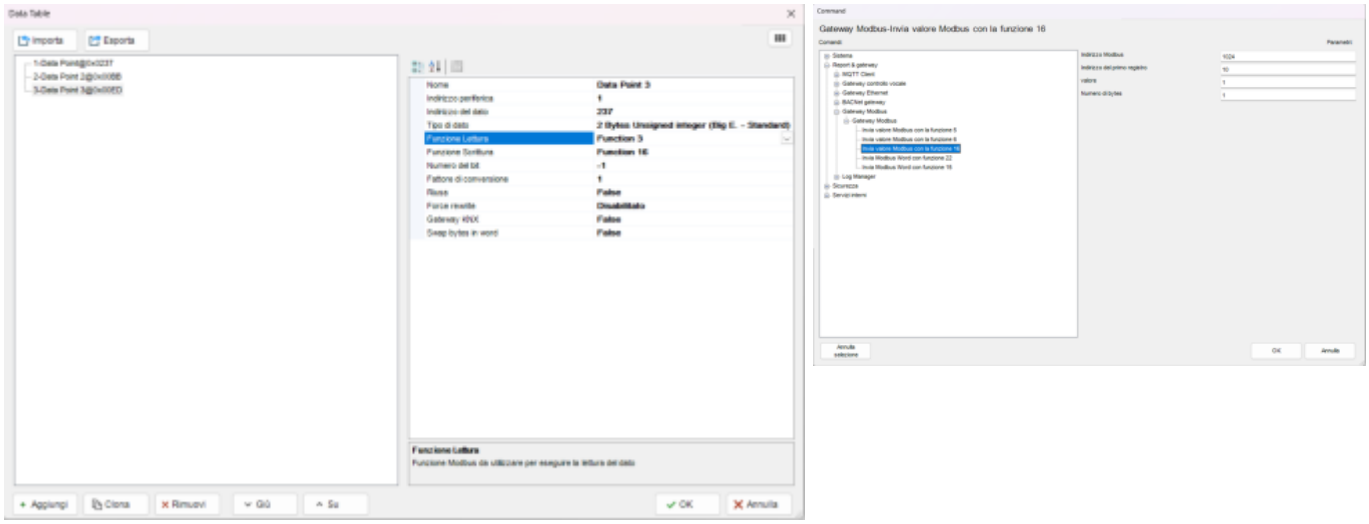
Function 16 – Write Multiple Holding Registers

Scrive uno o più Holding Register consecutivi.

Nel Configurator il comando è denominato **Invia valore Modbus con la funzione 16** e utilizza i seguenti parametri:

- **Indirizzo Modbus:** indirizzo dello Slave.
- **Indirizzo del primo registro:** indirizzo del primo registro.
- **valore:** valore da scrivere.
- **Numero di bytes:** numero di byte utilizzati dal comando.

La dimensione configurata deve essere coerente con il numero di registri e con il tipo di valore richiesto dal dispositivo.



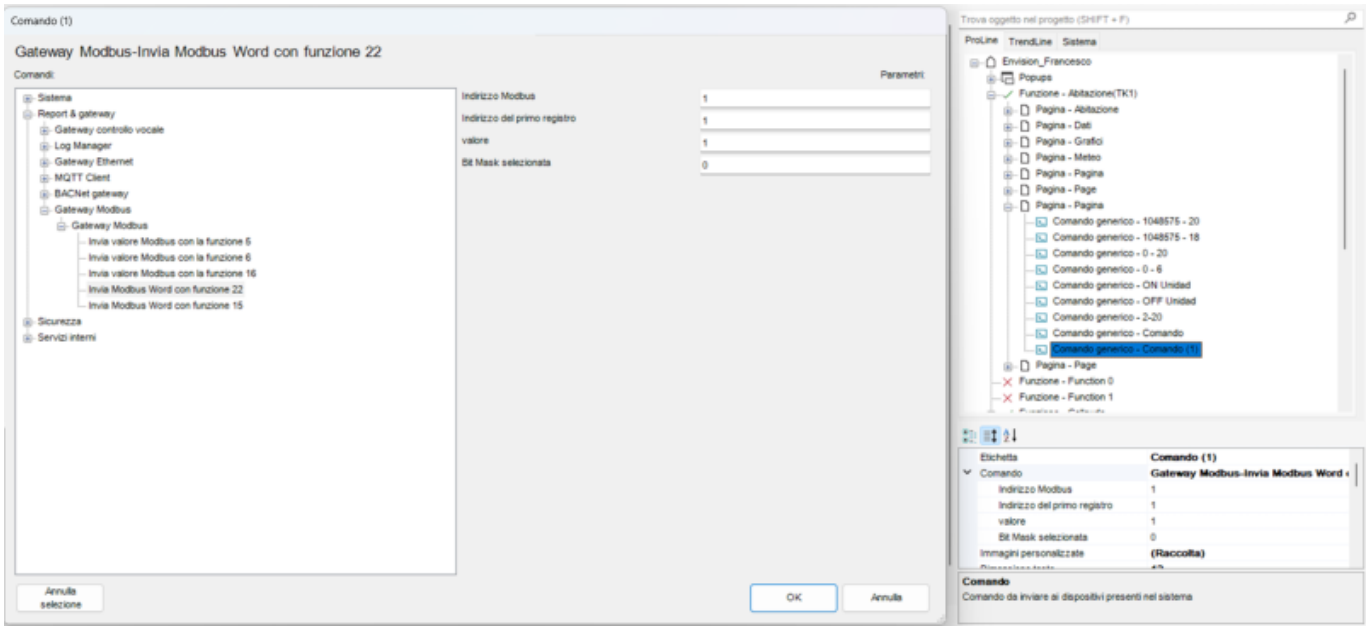
Function 22 – Mask Write Register

La Function 22 permette di modificare bit selezionati all'interno di un Holding Register senza gestire il registro come un semplice comando di scrittura completa.

Nel Configurator il comando è denominato **Invia Modbus Word con funzione 22** e utilizza i seguenti parametri:

- **Indirizzo Modbus:** indirizzo dello Slave.
- **Indirizzo del primo registro:** indirizzo del registro.
- **valore:** valore da applicare.
- **Bit Mask selezionata:** maschera che identifica i bit interessati dalla modifica.

Il significato della maschera e il risultato atteso devono essere verificati inizialmente su un registro di prova o tramite un simulatore Modbus.



Esempi di configurazione Master

Gli esempi seguenti utilizzano valori dimostrativi. Prima di applicarli a un dispositivo reale, sostituire tutti i parametri con quelli indicati nel relativo manuale Modbus.

Esempio 1: lettura di un Holding Register seriale

Si desidera leggere una temperatura da un dispositivo con le seguenti caratteristiche:

- Modbus RTU;
- Slave Address 1;
- comunicazione 9600 baud, 8 data bit, nessuna parità e 1 stop bit;
- temperatura disponibile nel primo Holding Register;
- Function 3;
- valore unsigned a 16 bit;
- valore grezzo 235 corrispondente a 23,5 °C.

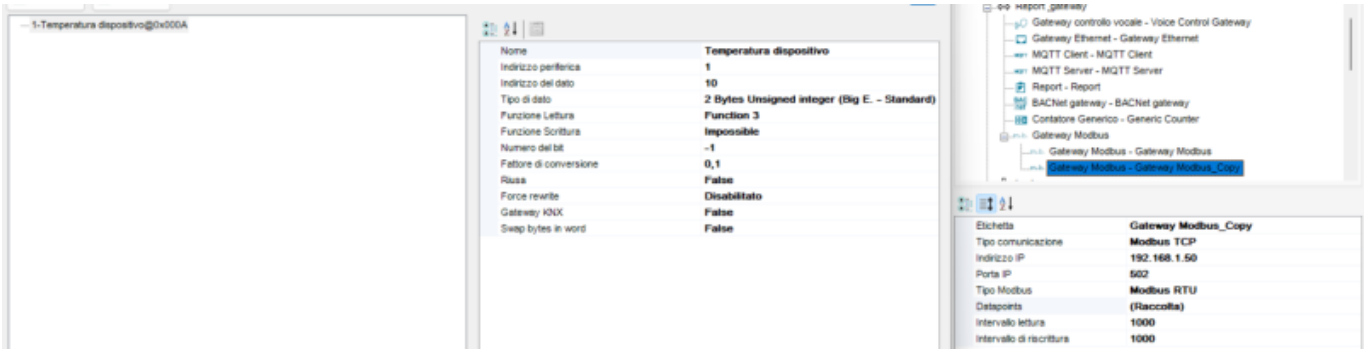
Configurazione del gateway:

Proprietà	**Valore**
Tipo comunicazione	Modbus Seriale
Numero porta	Porta seriale utilizzata dal server
Baud Rate	9600
Data Bits	8
Parità	None
Stop Bits	1

Configurazione del Datapoint:

Proprietà	**Valore**
Nome	Temperatura dispositivo
Indirizzo periferica	1
Indirizzo del dato	10
Indirizzo trasmesso dal server	0
Tipo di dato	Tipo unsigned a 16 bit corrispondente
Funzione Lettura	Function 3
Fattore di conversione	0.1

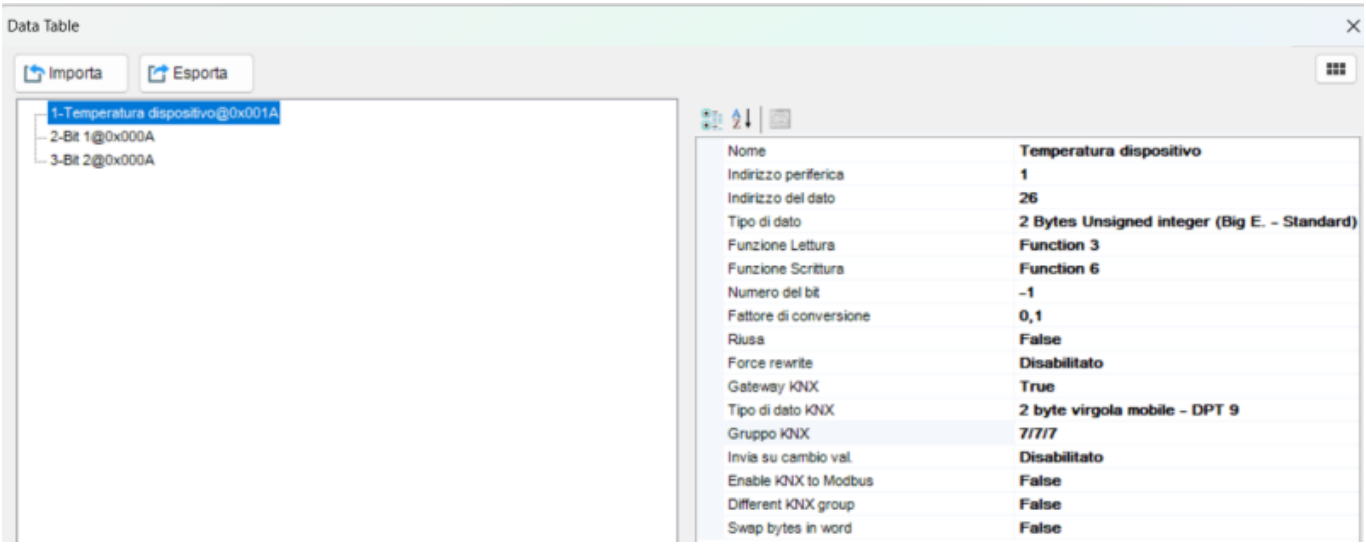
Il valore atteso all'interno di Thinknx è 23,5.



Esempio 2: manuale con indirizzo di protocollo da 0

Il manuale indica esplicitamente **Protocol Address 25**. Poiché si tratta già dell'indirizzo trasmesso sul protocollo, nel Configurator deve essere inserito 26.

Valore	Indirizzo
Indirizzo indicato dal manuale	25
Indirizzo del dato nel Configurator	26
Indirizzo trasmesso dal server	25

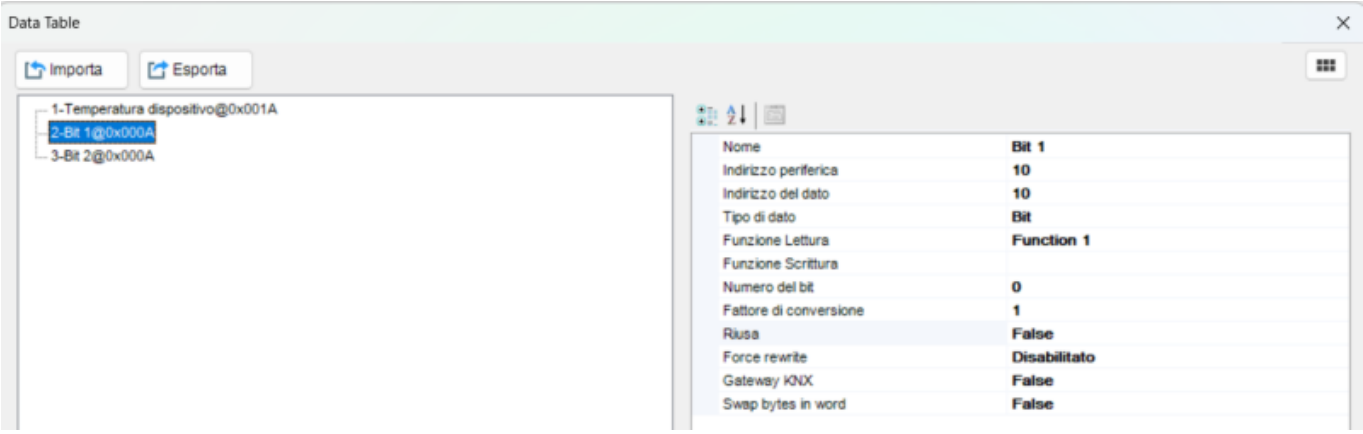


Esempio 3: lettura di più stati dallo stesso registro

Un Holding Register contiene diversi stati di allarme, uno per ciascun bit.

Creare più Datapoint con gli stessi valori di **Indirizzo periferica**, **Indirizzo del dato**, **Tipo di dato** e **Funzione Lettura**. Configurare un **Numero del bit** differente per ogni allarme.

Nel primo Datapoint lasciare **Riusa** disabilitato. Nei successivi abilitare **Riusa**, in modo che il server esegua una sola lettura del registro e riutilizzi il valore ricevuto.



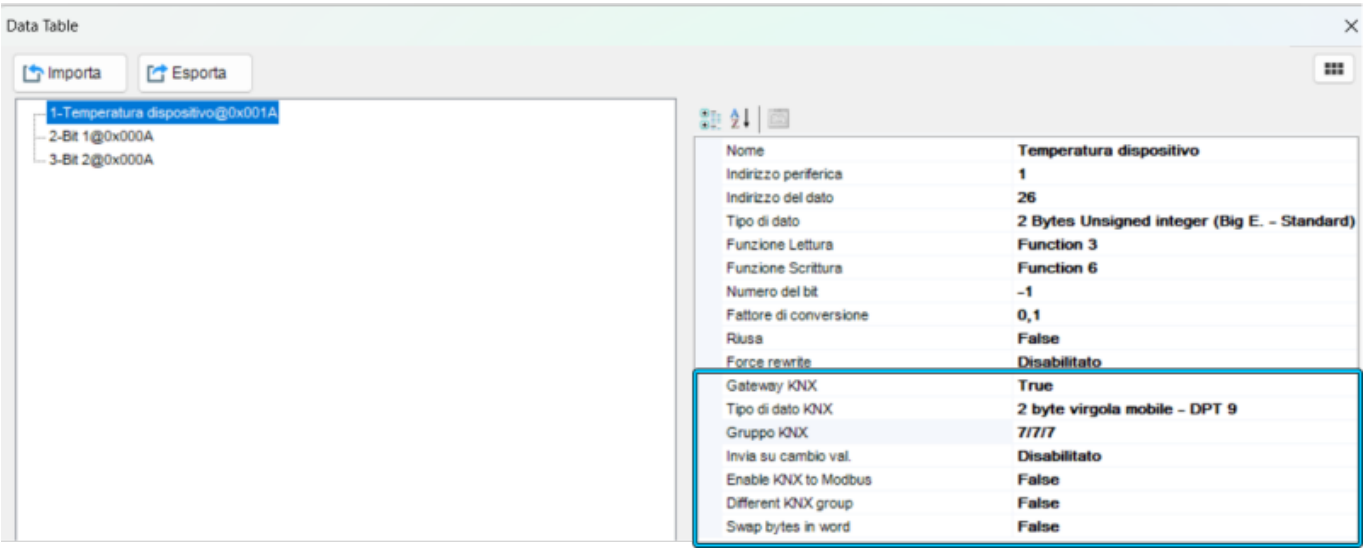
Esempio 4: integrazione Modbus–KNX

Per inviare una temperatura letta da Modbus sul bus KNX:

1. Configurare e verificare prima il Datapoint Modbus.
2. Abilitare **Gateway KNX**.
3. Selezionare il **Tipo di dato KNX** corrispondente alla temperatura.
4. Configurare **Gruppo KNX** con il gruppo di feedback.
5. Abilitare **Invia su cambio val.** se il valore deve essere inviato solamente quando cambia.

Per consentire anche la scrittura da KNX verso Modbus:

1. Verificare che il registro sia scrivibile e che la **Funzione Scrittura** sia corretta.
2. Abilitare **Enable KNX to Modbus**.
3. Configurare **Different KNX group** con il gruppo di comando.
4. Inviare un valore di prova e verificare la risposta del dispositivo.



Esempio 5: lettura tramite Modbus TCP

Si desidera leggere l'Holding Register 10 di un dispositivo Modbus TCP con indirizzo IP 192.168.1.50, porta 502 e Unit ID 1.

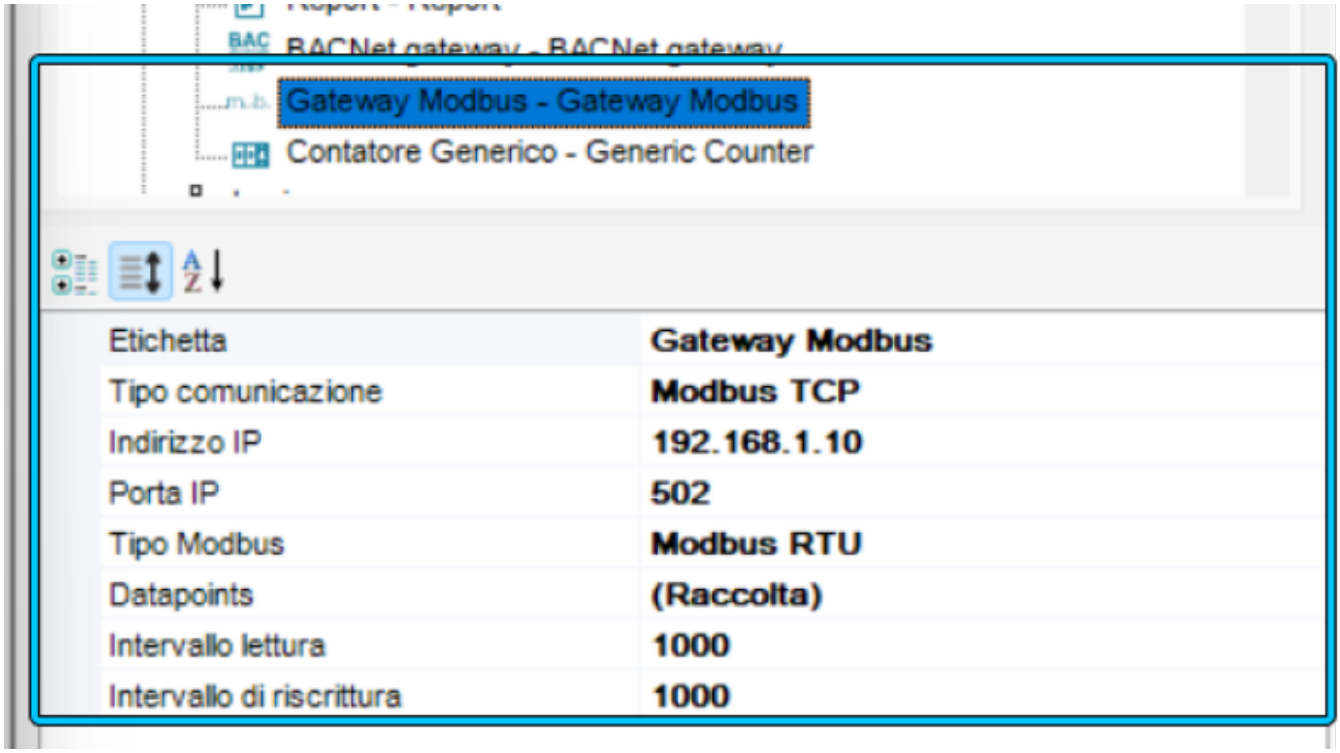
Configurazione del gateway:

Proprietà	**Valore**
Tipo comunicazione	Modbus TCP
Indirizzo IP	192.168.1.50
Porta IP	502

Configurazione del Datapoint:

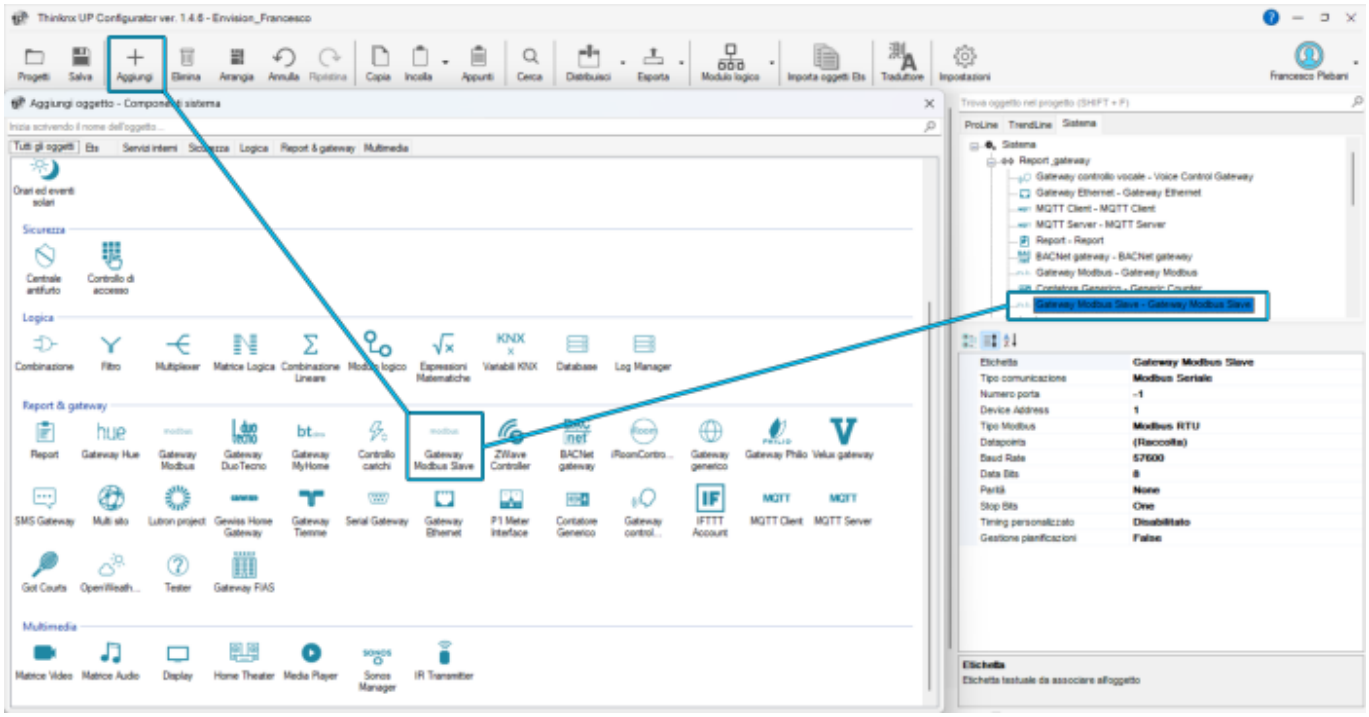
Proprietà	**Valore**
Indirizzo periferica	1
Indirizzo del dato	10
Indirizzo trasmesso dal server	9
Funzione Lettura	Function 3

Dopo il caricamento del progetto, verificare che il server apra la connessione TCP e riceva una risposta valida dal dispositivo prima di aggiungere altri Datapoint.



Gateway Modbus Slave

L'oggetto **Gateway Modbus Slave** permette al server Thinknx di esporre Datapoint a un Master esterno. Il Master può leggere o scrivere i valori disponibili secondo le funzioni e gli indirizzi configurati.



Parametri di comunicazione

La proprietà **Tipo comunicazione** definisce il metodo di comunicazione.

Per **Modbus Seriale** sono disponibili:

- **Numero porta**;
- **Baud Rate**;
- **Data Bits**;
- **Parità**;
- **Stop Bits**;
- **Timing personalizzato**, per modificare i tempi collegati alla gestione RTS e alla trasmissione seriale;
- **Gestione pianificazioni**, per abilitare o disabilitare la gestione delle pianificazioni.

Per **Modbus Seriale su TCP** oppure **Modbus TCP**, la proprietà **Porta IP** definisce la porta sulla quale il Gateway Modbus Slave rimane in ascolto. Non viene richiesto un indirizzo IP remoto, poiché il server Thinknx opera come Slave e attende la connessione del Master.

Ulteriori proprietà:

- **Device Address:** indirizzo con il quale il server viene identificato sul bus Modbus.
- **Tipo Modbus:** variante seriale utilizzata, selezionabile tra RTU e ASCII.
- **Datapoints:** raccolta dei valori esposti dal server.

Datapoint Slave

Per ciascun Datapoint sono disponibili:

- **Nome:** nome del Datapoint.
- **Indirizzo del dato (inizio):** primo indirizzo dell'intervallo nel quale viene esposto il dato.

- **Indirizzo del dato (fine):** ultimo indirizzo dell'intervallo nel quale viene esposto il dato.
- **Tipo di dato:** tipo di dato letto o scritto.
- **Funzione Lettura:** funzione utilizzata dal Master per leggere il valore.
- **Funzione Scrittura:** funzione utilizzata dal Master per scrivere il valore.
- **Numero del bit:** bit del valore associato al Datapoint.
- **Fattore di conversione:** fattore applicato al valore grezzo.
- **Gruppo KNX:** indirizzo di gruppo KNX associato.
- **Tipo di dato KNX:** tipo di dato KNX.
- **Invia su cambio val.:** invia il valore sul bus KNX quando il valore Modbus cambia.

Esempio di utilizzo

Per rendere disponibile a un sistema esterno una temperatura presente sul bus KNX:

1. Aggiungere il Gateway Modbus Slave.
2. Configurare la comunicazione e il **Device Address**.
3. Creare un Datapoint con l'indirizzo, la funzione e il tipo di dato richiesti.
4. Associare il **Gruppo KNX** che contiene la temperatura.
5. Caricare il progetto sul server.
6. Utilizzare un Master Modbus esterno per leggere l'indirizzo configurato.
7. Confrontare il valore ricevuto con quello presente sul bus KNX.

Verifica e messa in servizio

La verifica deve essere eseguita inizialmente con un solo dispositivo e un solo Datapoint. Dopo avere confermato la comunicazione, è possibile aggiungere gli altri valori.

Test del dispositivo con Modbus Poll

Modbus Poll può essere utilizzato come Master esterno per verificare un dispositivo Slave indipendentemente da Thinknx.

Procedura consigliata:

1. Collegare il PC al dispositivo tramite rete oppure adattatore seriale.
2. Configurare gli stessi parametri indicati nel manuale.
3. Impostare Slave Address, Function, indirizzo e quantità di registri.
4. Leggere un singolo valore noto.
5. Verificare sia l'indirizzo indicato dal manuale sia la sua convenzione zero-based o one-based.
6. Annotare la configurazione funzionante e riprodurla nel Gateway Modbus Thinknx.

Se Modbus Poll non comunica con il dispositivo, il problema deve essere risolto prima di procedere con la configurazione Thinknx.

Test del server con Modbus Slave

Modbus Slave può essere utilizzato per simulare il dispositivo e osservare le richieste inviate dal server Thinknx quando opera come Master.

Questa prova permette di controllare:

- connessione del server;
- Slave Address;
- Function utilizzata;
- indirizzo trasmesso dopo l'offset -1;
- numero di registri o Coil;
- valore inviato in scrittura.

Per una prova Modbus TCP, configurare nel Gateway Modbus l'indirizzo IP del PC sul quale è in esecuzione il simulatore. Verificare che firewall e porta TCP consentano la connessione.

Gli strumenti Modbus Poll e Modbus Slave sono disponibili sul sito Modbus Tools.

Analisi Modbus TCP con Wireshark

Wireshark permette di acquisire il traffico Modbus TCP e analizzare le richieste e le risposte scambiate sulla rete.

Filtri utili:

```
mbtcp
modbus
tcp.port == 502
ip.addr == 192.168.1.100
```

Sostituire l'indirizzo IP dell'esempio con quello del dispositivo. La cattura permette di controllare apertura della connessione TCP, Function Code, indirizzi, quantità richieste, dati ed eventuali risposte di errore.

Wireshark non consente di intercettare direttamente un bus Modbus RTU attraverso una normale scheda di rete. Per il traffico seriale sono necessari un'interfaccia e un metodo di acquisizione adatti al bus utilizzato.

Troubleshooting

Il dispositivo non risponde

Controllare:

- alimentazione e collegamenti fisici;
- polarità e cablaggio del bus RS485;
- porta seriale selezionata;
- indirizzo IP e porta TCP;
- Indirizzo periferica, Slave Address o Unit ID;

- Baud Rate, Data Bits, Parità e Stop Bits;
- modalità Modbus RTU, ASCII, Modbus Seriale su TCP oppure Modbus TCP;
- firewall e raggiungibilità di rete;
- tempi di risposta del dispositivo.

Prima di modificare il progetto Thinknx, verificare il dispositivo con Modbus Poll o uno strumento equivalente.

Viene letto il registro sbagliato

- Controllare se il manuale utilizza indirizzi da 0 oppure da 1.
- Ricordare che il server Thinknx trasmette l'indirizzo configurato meno 1.
- Verificare che non sia stato applicato manualmente un secondo offset.
- Non utilizzare automaticamente l'intera notazione 40001 come **Indirizzo del dato**.
- Controllare che la **Funzione Lettura** corrisponda all'area dati.

Il valore è errato

- Verificare **Tipo di dato** e dimensione del valore.
- Controllare se il dato è signed oppure unsigned.
- Verificare l'eventuale ordine dei registri o dei byte indicato dal produttore.
- Disabilitare temporaneamente il **Fattore di conversione** e controllare il valore grezzo.
- Controllare se il dato occupa più registri consecutivi.
- Verificare **Numero del bit** quando viene estratto un singolo bit.

La lettura funziona ma la scrittura no

- Verificare che il dato sia scrivibile.
- Controllare che la **Funzione Scrittura** sia supportata dal dispositivo.
- Verificare l'intervallo dei valori accettati.
- Controllare l'indirizzo e l'offset.
- Verificare che **Enable KNX to Modbus** sia abilitato quando il comando proviene da KNX.
- Provare la scrittura direttamente con Modbus Poll.
- Controllare l'eventuale risposta Modbus Exception.

La comunicazione è instabile

- Aumentare l'**Intervallo lettura**.
- Ridurre temporaneamente il numero di Datapoint.
- Utilizzare **Riusa** per più bit dello stesso registro.
- Controllare **Intervallo di riscrittura** e **Force rewrite**.
- Verificare cablaggio, terminazioni e qualità del convertitore seriale.
- Controllare se lo Slave risponde entro i tempi richiesti.

Il valore Modbus non viene inviato a KNX

- Verificare prima che il Datapoint Modbus venga letto correttamente.
- Controllare che Gateway KNX sia abilitato.
- Verificare **Tipo di dato KNX** e indirizzo di gruppo.
- Disabilitare temporaneamente **Invia su cambio val.** per escludere che il valore non venga inviato perché invariato.
- Controllare il monitor KNX e le flag degli oggetti coinvolti.

Modbus Exception

Le risposte di eccezione più comuni sono:

Codice	Significato	Verifica principale
01	Illegal Function	La funzione non è supportata dal dispositivo.
02	Illegal Data Address	L'indirizzo o l'intervallo richiesto non è disponibile.
03	Illegal Data Value	Il valore o la quantità richiesta non è valida.
04	Server Device Failure	Il dispositivo non è riuscito a completare l'operazione.

Consultare sempre il manuale del dispositivo, perché il produttore può prevedere condizioni aggiuntive.

Procedura rapida di diagnostica

1. Raccogliere tutti i parametri dal manuale Modbus.
2. Testare il dispositivo con Modbus Poll o uno strumento equivalente.
3. Leggere un solo valore noto.
4. Identificare la convenzione di indirizzamento.
5. Creare un solo Datapoint nel Gateway Modbus Thinknx.
6. Verificare l'indirizzo trasmesso considerando l'offset -1.
7. Controllare il valore grezzo senza conversioni.
8. Verificare successivamente **Tipo di dato** e **Fattore di conversione**.
9. Provare la scrittura, se richiesta.
10. Aggiungere il gateway KNX solamente dopo avere verificato Modbus.
11. Estendere gradualmente la configurazione agli altri Datapoint.

From:
http://www.thinknx.com/wiki/ - **Learning Thinknx**

Permanent link:
http://www.thinknx.com/wiki/doku.php?id=it:modbus

Last update: **2026/09/16 11:02**