

Modbus Client

Manuale Modbus Client v2.38

7 febbraio 2024



Federico Turco
federico.turco97@gmail.com



Indice

1	Homepage	1
1.1	Modbus Secure	2
2	Finestra principale	3
2.1	FC01 - Coils	3
2.2	FC02 - Discrete inputs	5
2.3	FC03 - Holding registers	6
2.4	FC04 - Input registers	8
2.5	Diagnostica	9
3	Log pacchetti	10
4	Impostazioni	11
5	Template personalizzati	13
5.1	Definizione gruppi	15
5.2	Datatypes	16
5.3	Modificatori del formato	17
6	Tools holding registers	19
6.1	Tool comandi bit	19
6.2	Tool comandi byte	21
6.3	Tool comandi word	22
7	Gestione database	23
7.1	Salvataggio configurazione	23
7.2	Percorso configurazione	23
7.3	Caricamento configurazione	24
8	Menu a tendina	25
8.1	Menu File	25
8.2	Menu View	25
8.3	Menu Database	26
8.4	Menu Tools	26
8.5	Menu Import	26
8.6	Menu Export	27
8.7	Menu Info	27
9	Tasti di scelta rapida	28
10	Modbus Secure	29
10.1	Specifiche normativa	30
10.2	Estensioni X509v3	30
10.3	Generazione dei certificati	31

11 Altre note	32
----------------------	-----------

1 | Homepage

Con questo programma è possibile leggere e interrogare dispositivi compatibili con il protocollo Modbus sia RTU che TCP. Avviando il programma si apre la tab principale di connessione a uno slave. Selezionare TCP o RTU a seconda della tipologia di slave a cui ci si sta collegando. Per leggere slave RTU è necessario un convertitore USB-485. Nel caso di slave TCP a partire dalla versione v2.38 è possibile utilizzare una comunicazione criptata nella versione Modbus Secure.

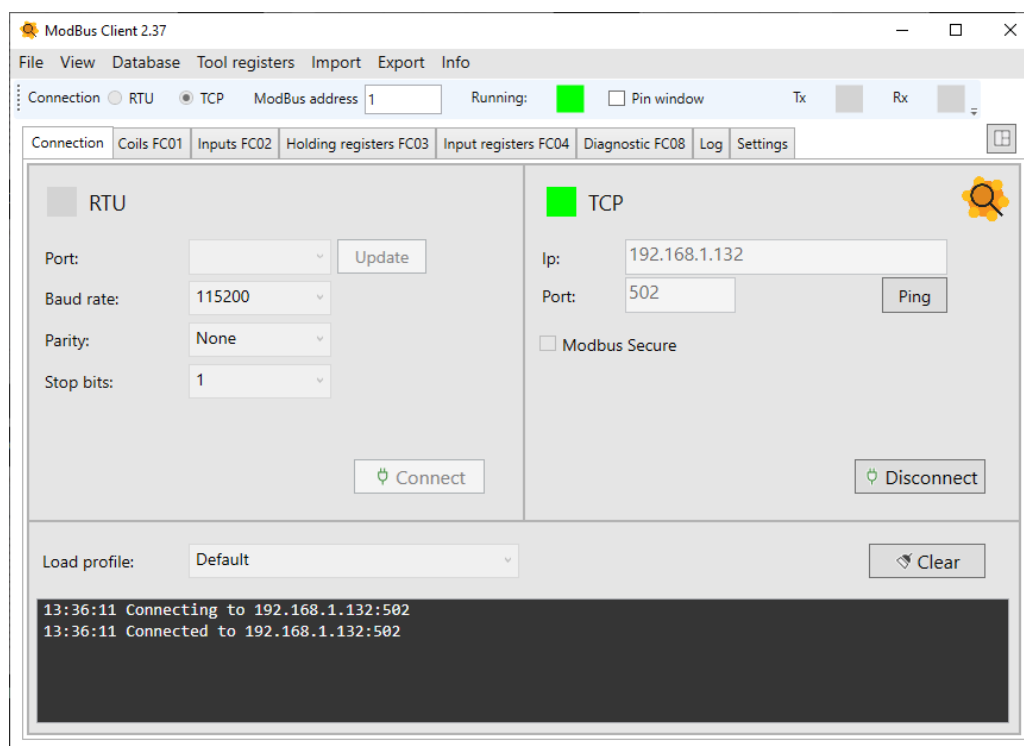
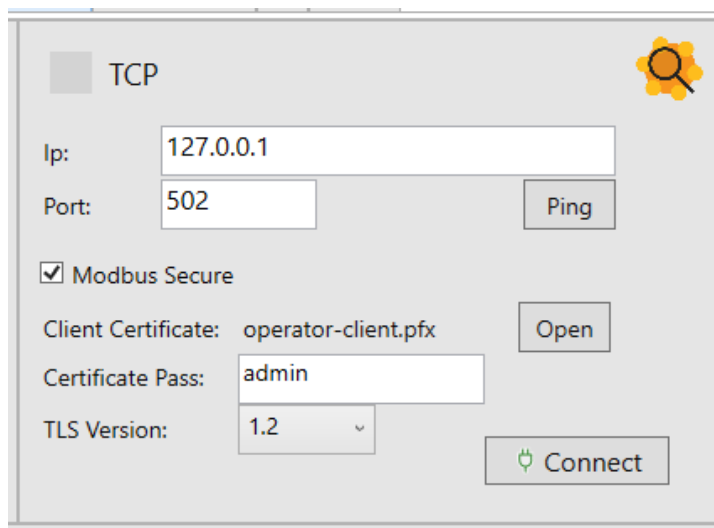


Figura 1.1: Homepage

Per testare la connessione a un indirizzo IP utilizzare il pulsante ping, se il ping ha successo il pulsante si colora di verde altrimenti di rosso. Quando la connessione a uno slave va a buon fine il cubetto "Running" si colora di verde (nel caso di connessioni RTU si considera a buon fine se riesce aprire correttamente la porta COM selezionata). Nella parte seguente del documento verranno descritte le varie tab e funzioni associate.

1.1 Modbus Secure

A partire dalla versione 2.38 è stato introdotto il supporto al protocollo ModBus Secure, secondo la specifica della normativa definita nel documento MB-TCP-Security-v21_2018-07-24. Flagghando la spunta "Modbus Secure" si abilita la configurazione "Secure" come riportato nell'immagine seguente:



The image shows a configuration window for a Modbus Client. At the top left, there is a tab labeled 'TCP'. Below it, the 'Ip:' field is set to '127.0.0.1' and the 'Port:' field is set to '502'. To the right of the port field is a 'Ping' button. Below these fields, the 'Modbus Secure' checkbox is checked. Underneath, the 'Client Certificate:' field is set to 'operator-client.pfx' with an 'Open' button to its right. The 'Certificate Pass:' field is set to 'admin'. The 'TLS Version:' dropdown menu is set to '1.2'. At the bottom right, there is a green 'Connect' button with a plug icon.

Figura 1.2: Modbus Secure

Il protocollo Modbus Secure verrà approfondito a seguire nella sezione 10, a livello di configurazione lavora via TCP criptando la comunicazione attraverso un layer SSL/TLS. Il protocollo Modbus Secure richiede (oltre a IP e porta) di caricare nel client un certificato protetto da password (da inserire nell'apposito campo). Il certificato va caricato in formato .pfx, questo formato contiene, assieme al certificato, anche la chiave privata da utilizzare per criptare la comunicazione. Il client supporta TLS v1.2 o 1.3 (la normativa richiede v1.2 o superiore).

2 | Finestra principale

Sulla destra della tab principale sono presenti 4 tab per leggere le varie risorse Modbus "FC 01 Coils", "FC02 Discrete Inputs", "FC 03 Holding Registers" e "FC04 Input Registers".

2.1 FC01 - Coils

La scheda Coils permette di leggere e scrivere uscite digitali con le funzioni FC01/FC05/FC15. I pulsanti Read/Loop/Write vengono sbloccati solo se la connessione a un dispositivo è andata a buon fine.

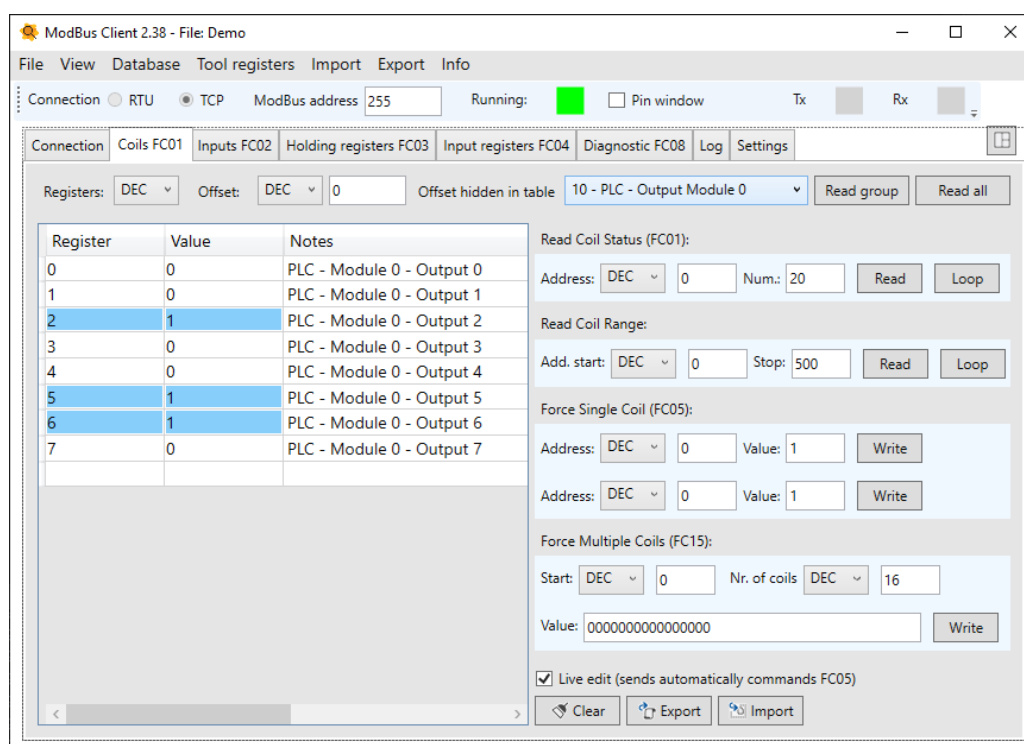


Figura 2.1: FC01 - Coils

La funzione loop permette di leggere i registri indicati in polling, l'intervallo di interrogazione si configura nella scheda impostazioni ("Settings"). Di default le celle lette vengono colorate di azzurro se il contenuto è popolato (inteso come > 0). In alto a destra sono presenti i comandi per leggere le risorse di un gruppo o tutte le risorse di tipo coils configurate nel profilo corrente. La configurazione delle risorse così come la creazione e associazione dei gruppi viene descritto nel capitolo 5.

Con il box "Read Coil Range" è possibile leggere un range di uscite digitali definito dall'utente, sarà poi il programma eventualmente a dividere il comando in più richieste FC01 ciascuna di n coils indicati sopra (nell'esempio seguente pari a 20). E' possibile inoltre forzare coils multiple (FC15) utilizzando il form in basso ("Force Multiple Coils") o importando un file csv precedentemente esportato.

Le coils settate a 1 vengono colorate di verde se la scrittura va a buon fine:

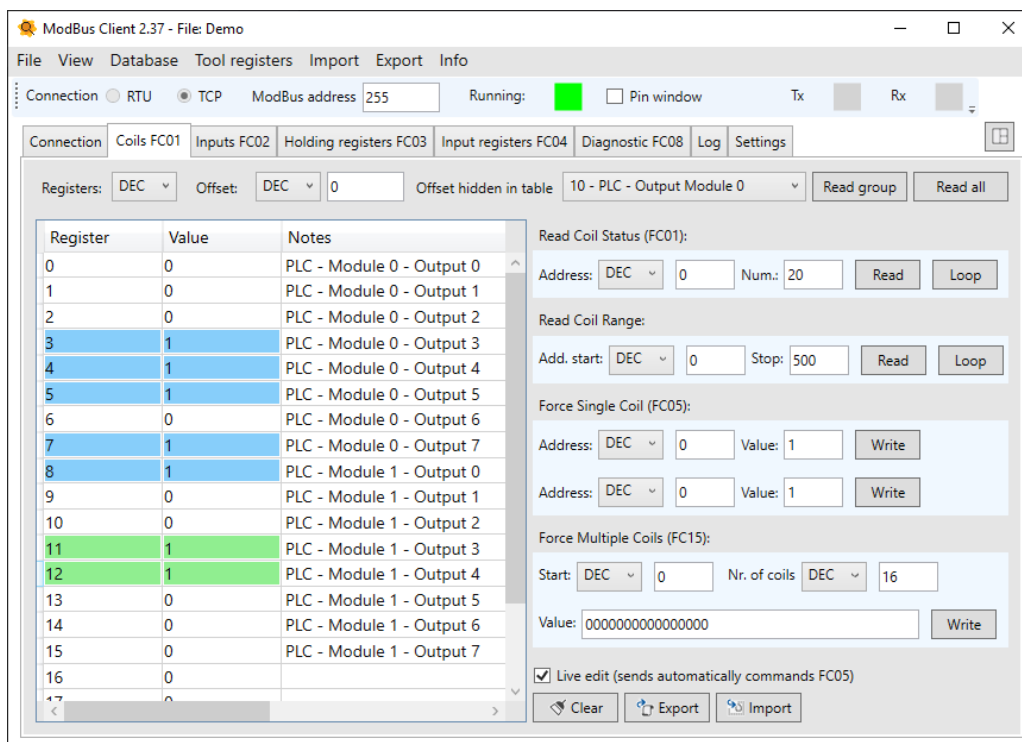


Figura 2.2: FC05 - Write Coils

Abilitando la modalità "Live Edit" è possibile modificare direttamente le coils editando la colonna "Value", in questo caso quando si modifica la riga il programma invia automaticamente il comando FC05 per impostare la coil al valore inserito.

2.2 FC02 - Discrete inputs

La scheda Inputs permette di leggere ingressi digitali con le funzioni FC02. I pulsanti Read/Loop come per la tab Coils vengono sbloccati solo se la connessione a un dispositivo è andata a buon fine. I pulsanti "Read group" e "Read all" permettono di leggere registri precedentemente configurati nel template personalizzato (i registri vengono letti singolarmente sia che siano o meno consecutivi tra di loro).

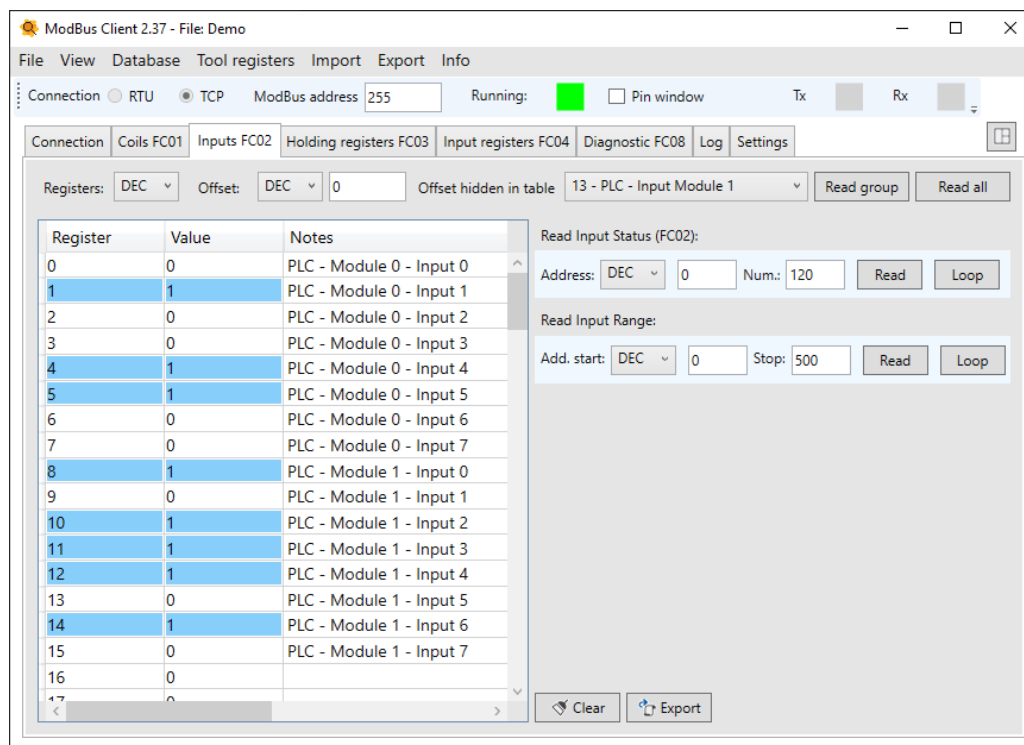


Figura 2.3: FC02 - Discrete Inputs

Con il box "Read Input Range" è possibile leggere un range di ingressi digitali definito dall'utente, sarà poi il programma eventualmente a dividere il comando in più richieste FC02 ciascuna di n ingressi specificati nel box delle letture singole (nell'immagine sopra pari a 120).

2.3 FC03 - Holding registers

La scheda Holding Registers permette di leggere e scrivere registri digitali a 16 bit con le funzioni FC03/FC06/FC16.

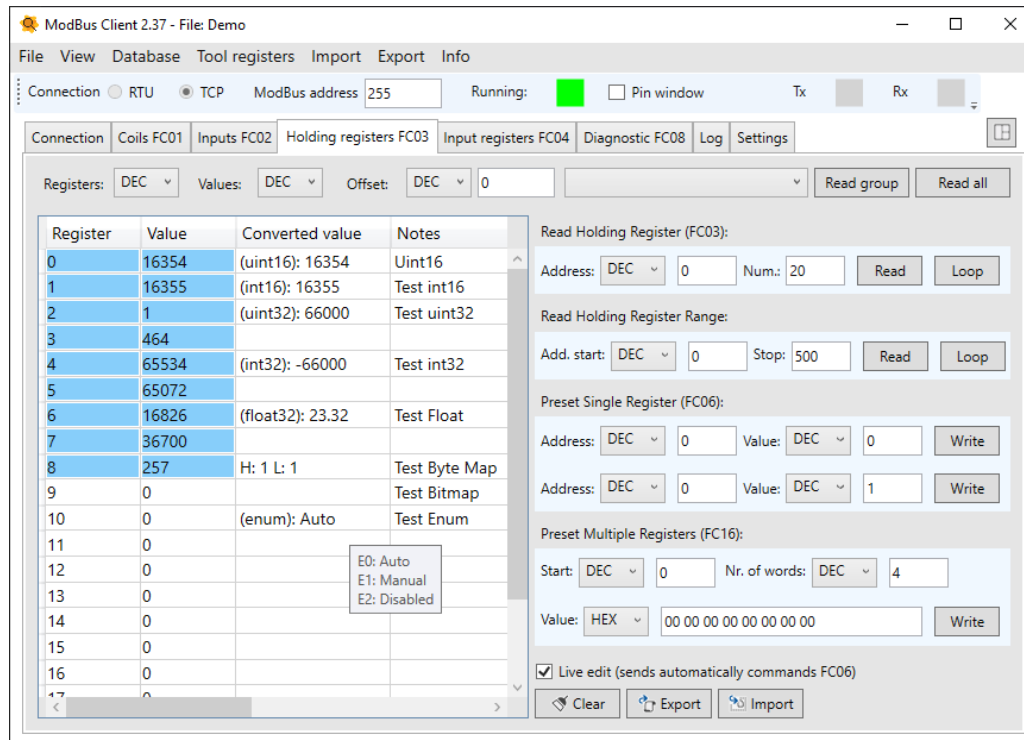


Figura 2.4: FC03 - Holding Registers

Il valore del campo "Value" può essere visualizzato sia in decimale (DEC) che in esadecimale (HEX). Nella tabella è possibile abilitare anche la visualizzazione del corrispondente valore in binario. Ampliando la finestra è possibile visualizzare informazioni aggiuntive (che vengono configurate nella finestra template), ad un registro holding infatti è possibile associare un'etichetta per identificarne il contenuto o visualizzare il valore convertito in integer/ float/string/etc.. Utilizzare il menu a tendina view per configurare le colonne che si desidera visualizzare nella tabella come mostrato nell'immagine seguente.

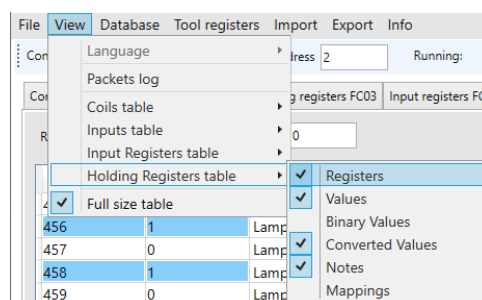


Figura 2.5: FC03 - View Tab Holding Registers

Con il box "Read Holding Register Range" è possibile leggere un range di registri definito dall'utente (anche > 123), sarà poi il programma eventualmente a dividere il comando in più richieste FC03 di n registri specificati nel box sopra (nell'immagine 120) e a popolare la tabella con tutti i registri richiesti.

Spuntando il flag "Live Edit" è possibile inviare la scrittura dei registri automaticamente modificando una cella delle colonne "Value", "Binary Value" o "Converted Value". La funzione live risulta molto utile per modificare in diretta i registri visualizzati.

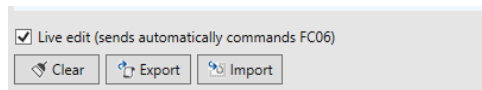


Figura 2.6: FC03 - Live edit

Le modifiche alle celle della colonna "Value" vengono inviate con la funzione FC06 (write single register), le modifiche alla colonna "Binary Value" vengono convertite da binario e inviate sempre come FC06, mentre le modifiche alla colonna "Converted Value" vengono inviate come FC06 o FC16 a seconda della tipologia del dato configurato (la funzione FC16 viene utilizzata per variabili integer o float a 32 o 64 bit). Nella colonna value è possibile inserire valori in esadecimale, se la visualizzazione è già in esadecimale il valore inserito sarà considerato sempre esadecimale, mentre nella visualizzazione decimale anteporre un "0x"/"x" o postporre un "h" per indicare che il valore che si vuole inviare è scritto in esadecimale. La tabella visualizzata può essere esportata in formato .csv o .json con il pulsante "Export" in basso a destra. Tabelle esportate in csv possono essere importate e inviate allo slave con il pulsante "Import". Questa funzione risulta comoda quando si vuole copiare una mappa di memoria da un PLC ad un altro o semplicemente esportare un backup ripristinabile in qualsiasi momento.

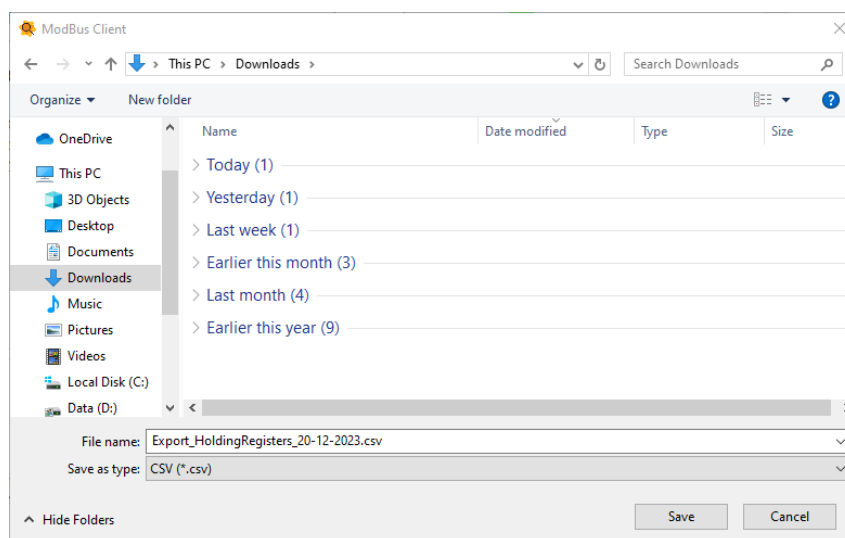


Figura 2.7: FC03 - Form import csv

2.4 FC04 - Input registers

La scheda Input Registers permette di leggere registri a 16 bit con le funzioni FC04. I pulsanti Read/Loop come per le altre tab vengono sbloccati solo se la connessione a un dispositivo è andata a buon fine.

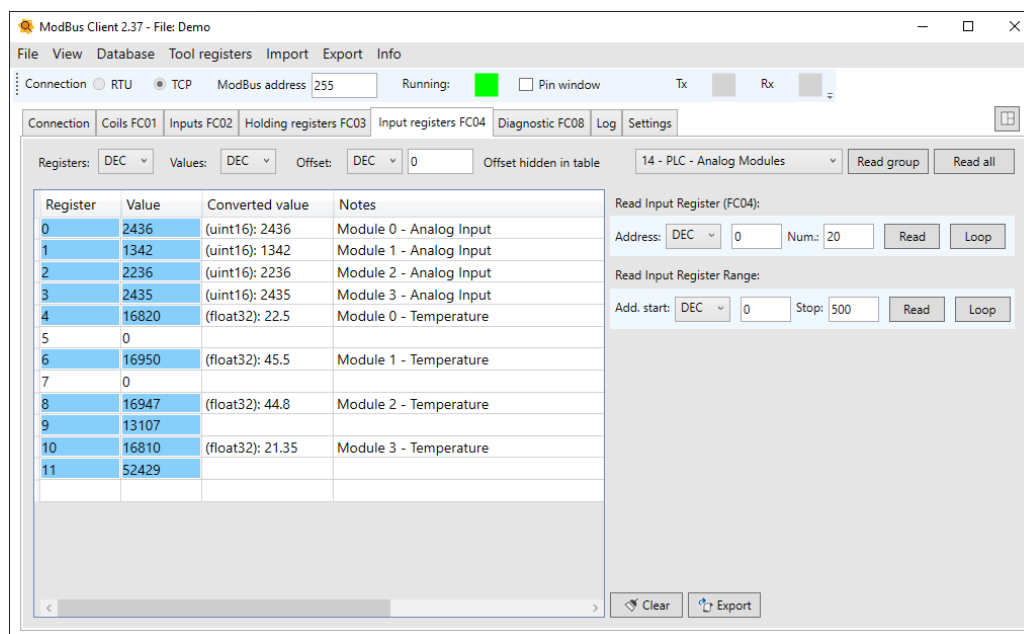


Figura 2.8: FC04 - Input Registers

Il valore del campo "Value" può essere visualizzato sia in decimale (DEC) che in esadecimale (HEX). A fianco viene mostrato anche il valore in binario. Ampliando la finestra è possibile visualizzare informazioni aggiuntive (che vengono configurate nella finestra template). Ad un registro infatti è possibile associare un'etichetta per identificarne il contenuto o visualizzare il valore convertito in integer/float/string/etc.. Per meglio chiarire questa parte si veda la sezione Template. Con il box "Read Input Register Range" è possibile leggere un range di registri definito dall'utente (anche > 123), sarà poi il programma eventualmente a dividere il comando in più richieste FC04 e a popolare la tabella con tutti i registri richiesti.

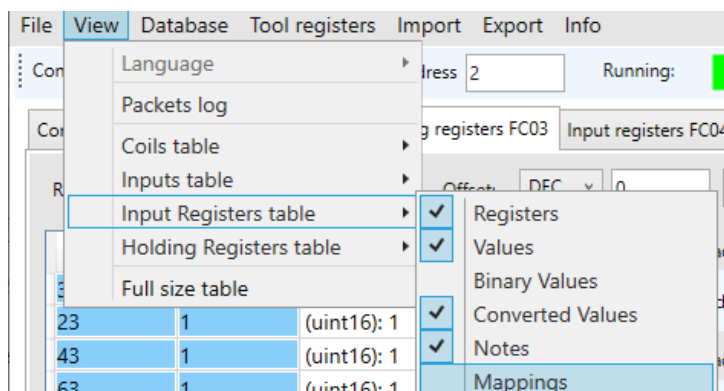


Figura 2.9: FC04 - View TabInput Registers

Come per gli holding register è possibile selezionare dal menu view quali colonne mostrare nella tabella.

2.5 Diagnostica

Nella tab diagnostica è possibile inviare comandi di diagnostica al dispositivo interrogato, si presti attenzione che non tutti i dispositivi rispondono alla funzione FC08.

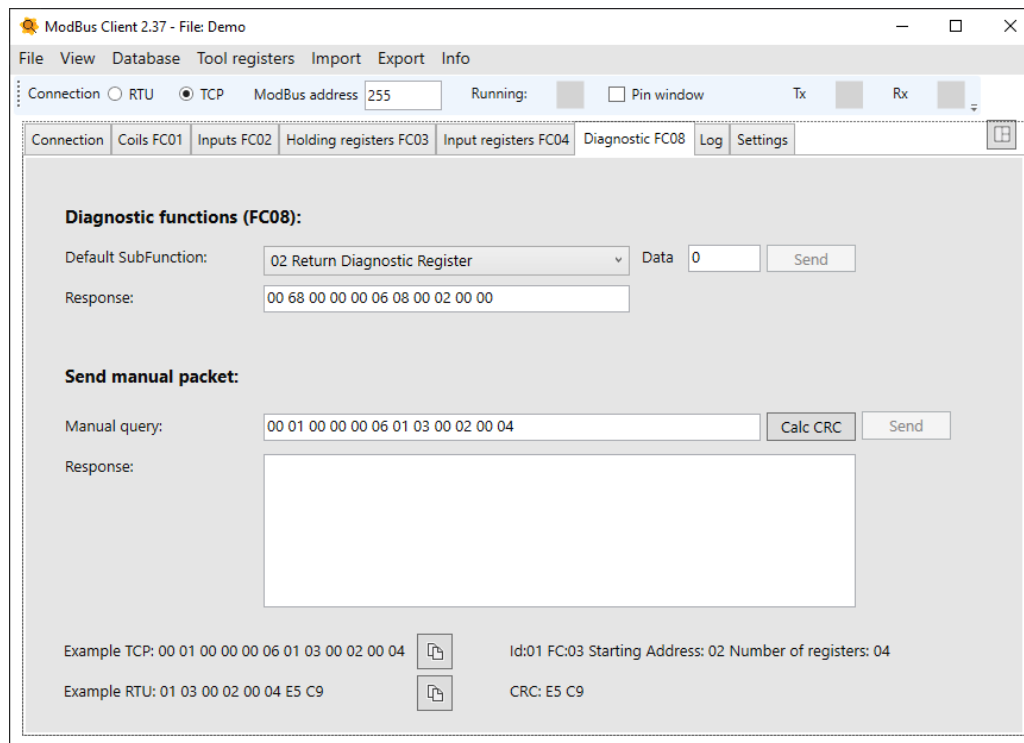


Figura 2.10: FC08 - Diagnostic

Funzioni FC08 supportate:

- 00 Return Query Data
- 01 Restart Communications Option
- 02 Return Diagnostic Register
- 03 Change ASCII Input Delimeter
- 04 Force Listen Only Mode
- 10 Clear Counters and Diagnostic Register
- 11 Return Bus Message Count
- 12 Return Bus Communication Error Count
- 13 Return Bus Exception Error Count
- 14 Return Slave Message Count
- 15 Return Slave No Response Count
- 16 Return Slave NAK Count
- 17 Return Slave Busy Count
- 20 Clear Overrun Counter and Flag

Nella tab diagnostica inoltre è possibile comporre trame ModBus manualmente, nel caso di trame RTU utilizzare il pulsante "Calc CRC" per calcolare e aggiungere in coda al pacchetto il CRC 16 ModBus. A livello di protocollo, nella versione RTU vengono aggiunti due byte di CRC utilizzati dallo slave per verificare l'integrità del pacchetto. In basso vengono mostrati due esempi di trame ModBus, utilizzare i due pulsanti per copiare gli esempi nel box della trama.

3 | Log pacchetti

La finestra di Log mostra i bytes raw inviati e ricevuti.

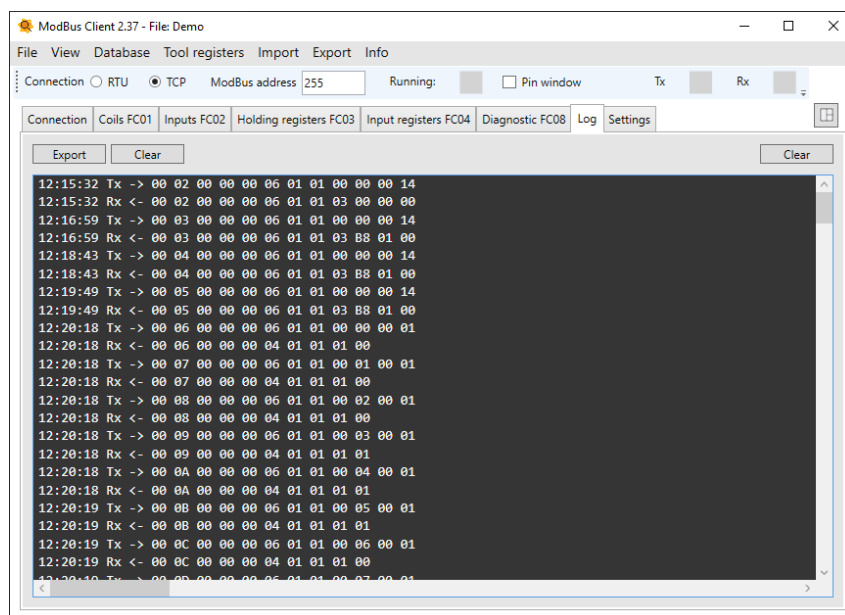


Figura 3.1: Finestra di log

La finestra di log può anche essere aperta in una finestra separata dal menu View -> Packet Log: (o con i tasti di scelta rapida Ctrl + L).

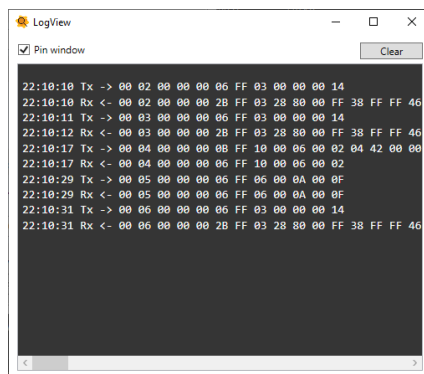


Figura 3.2: Finestra di log ridotta

Flaggando la spunta "Pin window" la finestra rimane sempre in primo piano rispetto alle altre.

4 | Impostazioni

La tab impostazioni contiene i parametri di funzionamento, attenzione che le impostazioni sono legate al profilo, cambiando profilo vengono caricate le rispettive impostazioni.

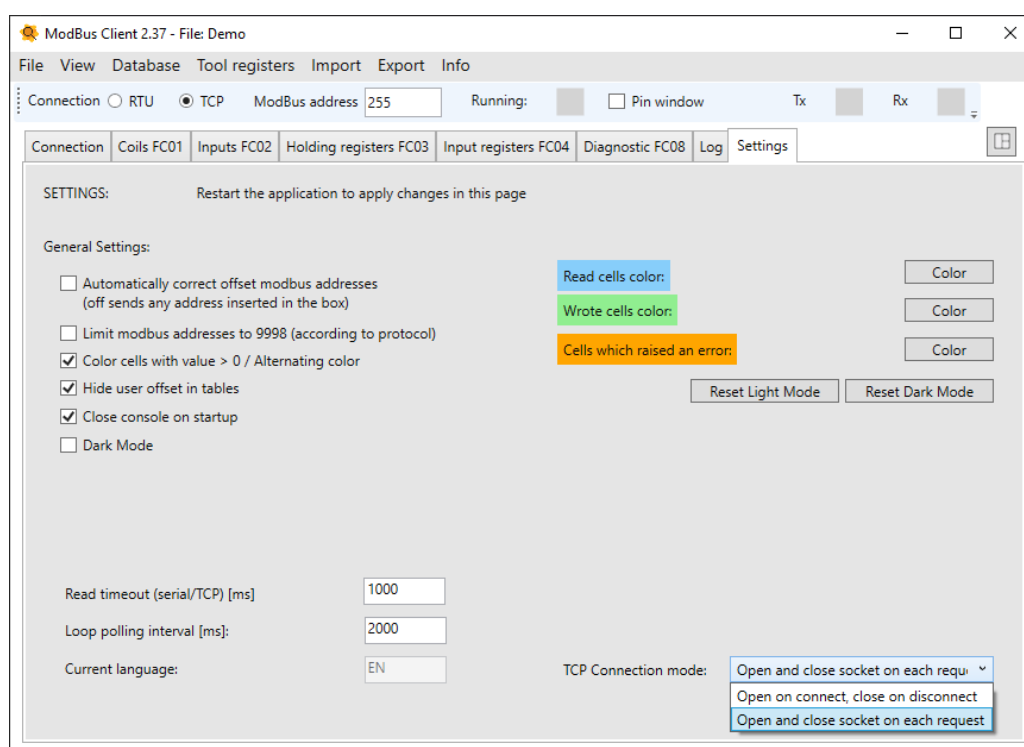


Figura 4.1: Tab impostazioni

- **CORREGGI AUTOMATICAMENTE OFFSET INDIRIZZI MODBUS TEXTBOX:** Se spuntata quando si inserisce un indirizzo nel formato originale del protocollo ad esempio holding register 40002 viene automaticamente inviato nella richiesta modbus come indirizzo 00001 eliminando quindi l'offset previsto per gli holding registers. In caso contrario viene richiesto l'indirizzo 40002. Stessa cosa per gli input registers (30001), discrete inputs (10001) o coils (1).
- **LIMITA GLI INDIRIZZI MODBUS A 9998:** Se spuntata vengono inviate solo le richieste di indirizzi tra 0 e 9998 (o corrispondente indirizzo Modbus es: 10001-19999/30001-39999/40001-49999). In caso contrario viene inviata la richiesta modbus qualsiasi sia l'indirizzo inserito. Ad esempio alcune CPU Beckhoff mettono a disposizione le word MW0,MW1,... partendo dall'holding register 0x3000 (dec 12288). Non spuntando la casella si possono quindi inviare richieste anche a indirizzi oltre il valore di 9998, fino a 65535.
- **COLORE CELLE > 0 / COLORE CELLE ALTERNATE:** Se spuntata vengono colorate solo le righe delle tabelle con un valore > 0, altrimenti le righe vengono colorate in maniera alternata.

- **NASCONDI OFFSET UTENTE NELLE TABELLE:** Se spuntata l'offset generale non viene visualizzato nelle tabelle ma viene contato nei comandi inviati via ModBus. (Per offset utente si intende l'offset inserito nel box presente in ogni tab).
- **CHIUDI CONSOLE ALL'AVVIO:** Chiude automaticamente la console quando si avvia l'applicazione.
- **DARK MODE:** Abilita la modalità scura (sfondi neri, testo bianco).

Oltre alle spunte sono presenti alcuni parametri quali:

- **READ TIMEOUT:** Timeout in ms di risposta a un comando.
- **LOOP POLLING INTERVAL:** Intervallo in ms tra una lettura e la successiva per i comandi di loop read.

Solo per connessioni TCP non secure è possibile scegliere tra due modalità di gestione delle socket:

- **TCP CONNECTION MODE:** Questo parametro permette di cambiare la gestione della socket TCP solo per la modalità non secure. A livello di socket TCP normalmente la connessione viene aperta, mantenuta aperta e chiusa al termine del suo utilizzo. Su connessioni instabili però risulta più comodo aprire e chiudere la socket ad ogni richiesta. La modalità corretta del protocollo sarebbe la prima, ma succede spesso su connessioni instabili (ad esempio slave connessi via modem) che si inizi la connessione, dopo qualche lettura cade la connessione, va in errore la socket e bisogna connettersi nuovamente con la socket che va in timeout non essendo più aperta sullo slave quando ritorna la connessione. Con la seconda modalità questo non accade perché la socket viene riaperta per ogni lettura e chiusa subito dopo per cui a livello pratico questa modalità può tornare molto utile.

5 | Template personalizzati

Nel menu "Database" -> "Open template editor" è possibile configurare etichette personalizzate, bit mappings ed eventuali conversioni in integer/float/string da associare ai vari registri.

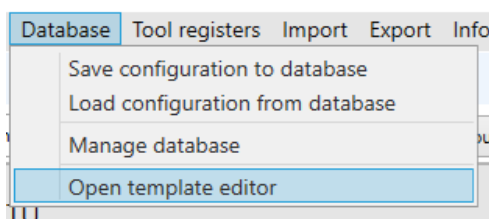


Figura 5.1: Menu Database

La finestra per inserire template personalizzati associati ai vari registri è la seguente:

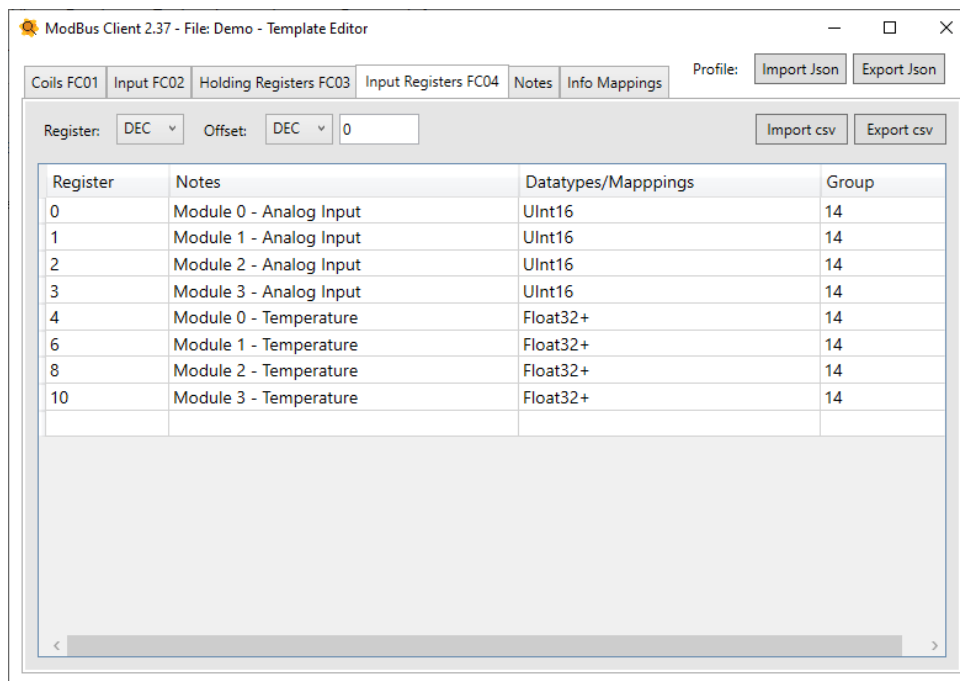


Figura 5.2: Template editor

Per ogni registro nella finestra template si può associare un eventuale etichetta, il datatype corrispondente, e se si desidera uno o più gruppi di risorse. Nella stessa finestra è possibile specificare se i registri inseriti sono da considerare in decimale o esadecimale ed un eventuale offset (positivo o negativo) da aggiungere ai valori inseriti. Ad esempio un offset di 100 sposta l'etichetta del registro 10 alla posizione 110. Questo

risulta comodo se, ad esempio, un PLC ha la %MW0 che inizia all'offset 0x4000. In questo caso si compila la tabella partendo da 0 e poi è sufficiente impostare come offset il valore HEX 4000. Nel caso in cui modelli diversi di PLC abbiano una posizione diversa delle %MW, per passare da un modello all'altro è sufficiente cambiare l'offset senza dover andare a modificare i registri uno per uno (si possono applicare anche offset negativi qualora fosse necessario). Si riportano a seguire gli screenshots di alcuni esempi basati sul template dell'immagine precedente:

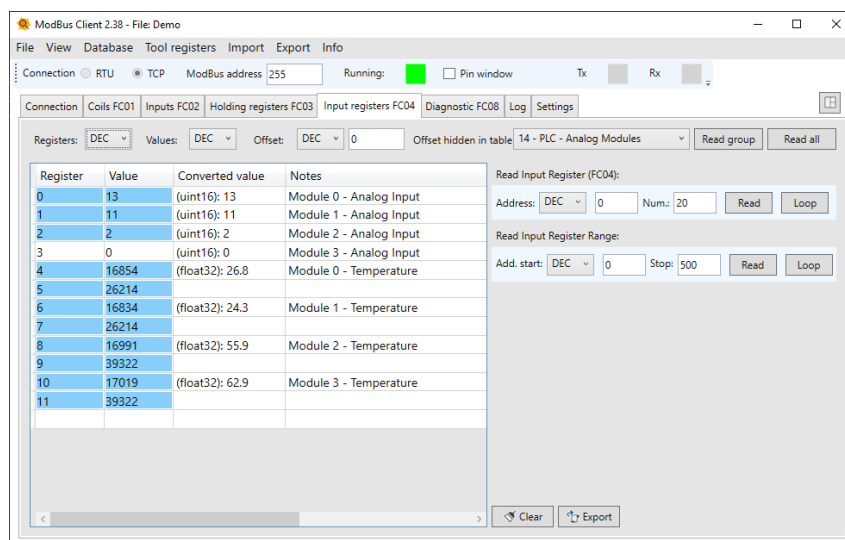


Figura 5.3: Template editor

Nella colonna "Notes" vengono visualizzate le descrizioni dei vari registri mentre nella colonna "Converted Value" viene visualizzato il contenuto di ciascun registro convertito nel datatype assegnatogli. In questo modo è molto semplice convertire valori a 32 o 64 bit, float o stringhe nel valore reale. Selezionando un gruppo dal menu a tendina inoltre il client va a leggere i registri del gruppo selezionato in ordine crescente.

Nella finestra template è possibile definire più gruppi di risorse e associarli ai vari registri, in questo modo nella finestra principale si possono leggere registri diversi (anche non consecutivi) richiamando il gruppo di risorse corrispondente.

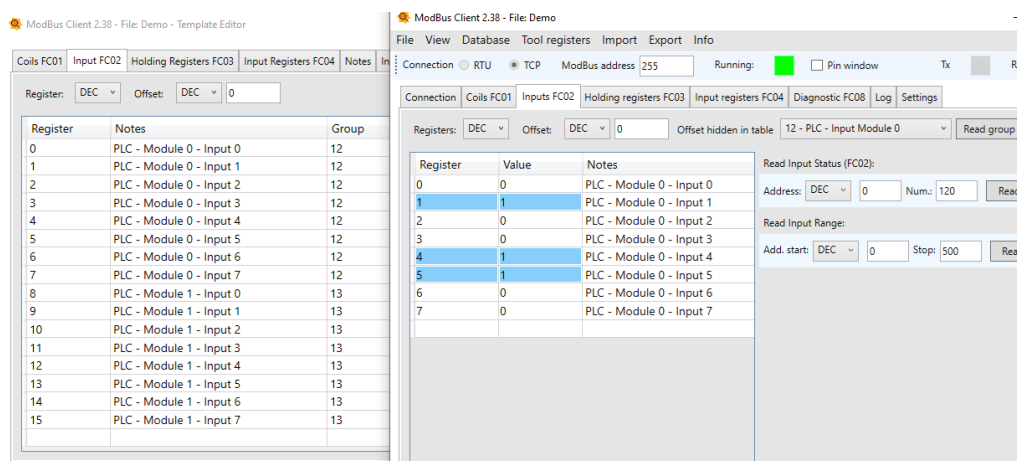


Figura 5.4: Gruppi di risorse

5.1 Definizione gruppi

Nella tab "Notes" della finestra template è possibile definire i gruppi di risorse da richiamare poi nella finestra principale. Non è necessario inserire i gruppi in ordine.

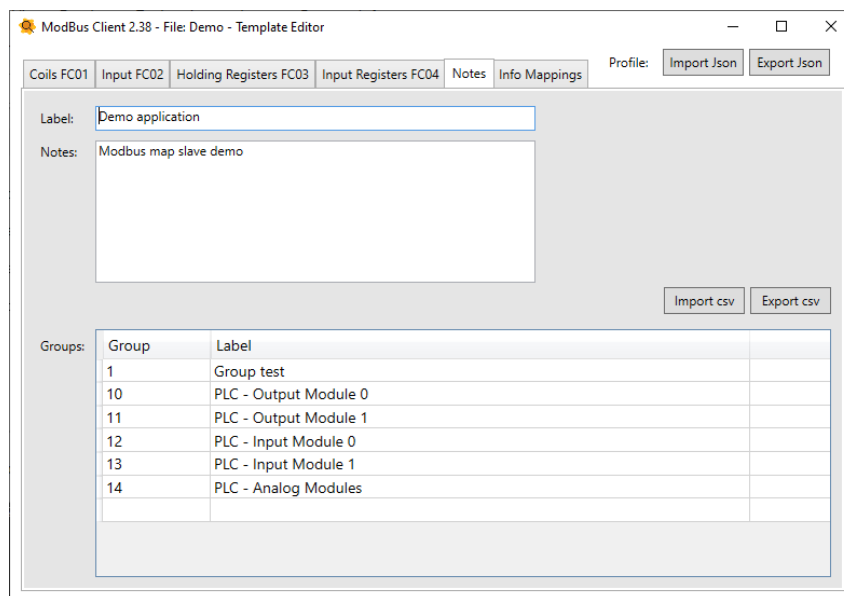


Figura 5.5: Definizione gruppi

L'ultima tab "Info Mappings" contiene un riepilogo dei datatypes implementati e gestiti dal client.

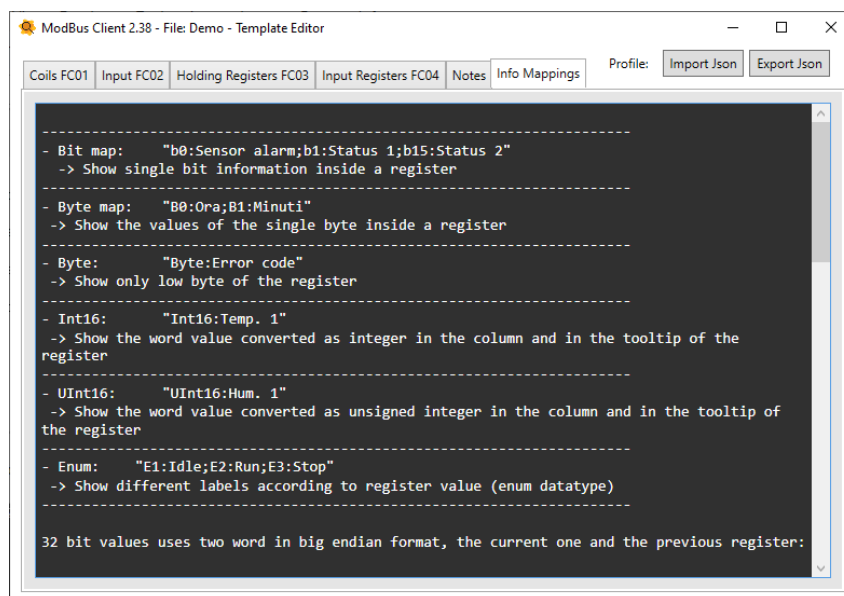


Figura 5.6: Info Mappings

5.2 Datatypes

Si riportano di seguito i vari datatypes supportati dal programma e configurabili nel template di un profilo.

Bit map

Mostra i singoli bit della word nella tooltip della riga con a fianco l'etichetta della risorsa.

```
b0:Presenza tensione;b1:Stato 1;b15:Stato
```

Byte map

Mostra i due byte della word nella tooltip della riga divisi per risorsa.

```
B0:Ora;B1:Minuti
```

Int16 / UInt16

Mostra la word con segno e visualizza il dato in int 16 nella tooltip della riga.

```
Int16:Temp. 1  
UInt16:Counter
```

Le variabili seguenti a 32 bit (due word) utilizzano la word del registro precedente (High Word) e corrente (Low Word) a cui fa riferimento nel formato Big Endian.

Float

Raccoglie due word e visualizza il dato in float nella tooltip della riga.

```
Float:Temperatura locale 1
```

Int32 / UInt32

Raccoglie due word e visualizza il dato in int32 o uint32 nella tooltip della riga.

```
Int32:Temperatura locale 1  
UInt32:Temperatura locale 2
```

Le variabili seguenti a 64 bit (due word) utilizzano le tre word dei registri precedenti (High Word) e corrente (Low Word) a cui fa riferimento nel formato Big Endian. Con i modificatori di formato è possibile specificare se utilizzare il registro corrente e i successivi tre.

Int64 / UInt64

```
Int64:Timestamp start time  
UInt64:Timestamp start time
```

String(len[, offset])

Con oggetti stringa è possibile convertire il contenuto dei registri in stringa (NULL terminated string). Nell'esempio a seguire vengono convertiti 8 byte in 8 caratteri ASCII con un offset di -2 (la stringa inizia dal registro precedente).

```
String(8,-2):Modello
```

5.3 Modificatori del formato

Swap

Aggiungendo un segno "-" o la stringa "_swap" vengono utilizzate le due word invertite, in formato Little Endian.

Swap: "UInt32-" "UInt32_swap"

Word offset

Aggiungendo un segno "+" viene utilizzato il registro corrente come High Word e il successivo come Low Word (Big Endian).

Offset: "UInt32+"

Word offset + Swap

"UInt32+" oppure "UInt32_swap+"

Combina le due precedenti, usa il registro corrente e successivo nel formato Little Endian.

Nell'esempio seguente si vedono 4 diversi mapping applicati al registro 102, come si vede nel caso di conversioni di variabili a 32 bit si può scegliere quale word utilizzare per comporre il risultato finale.

Template:

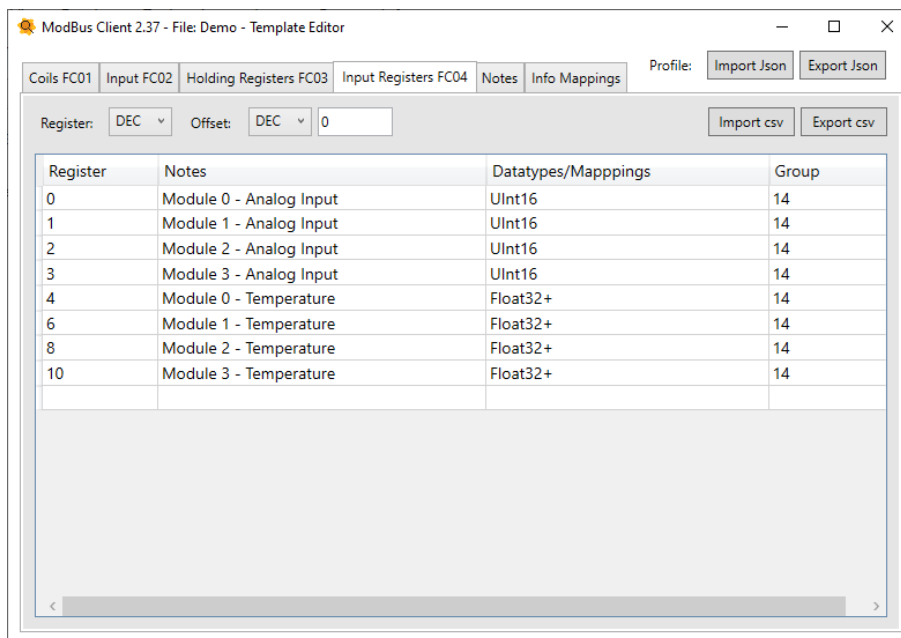


Figura 5.7:

Esempio conversione finale:

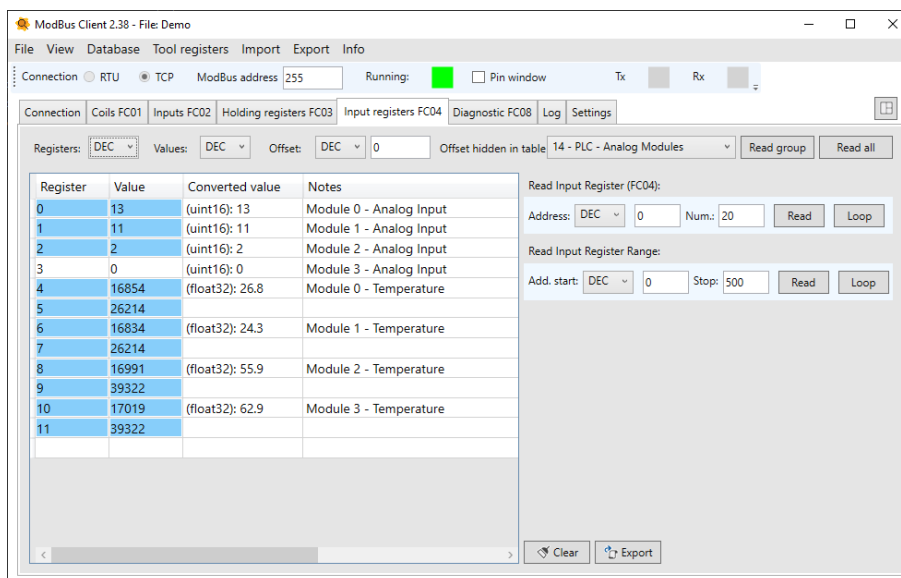


Figura 5.8:

6 | Tools holding registers

Per pilotare singoli bit o singole byte/word di holding registers sono presenti alcuni tools accessibili dal menu a tendina "Tool registers". I tool seguenti si applicano solo a holding register, sono utilizzati principalmente per pilotare su PLC oggetti di tipo %MX,%MB,%MW:

6.1 Tool comandi bit

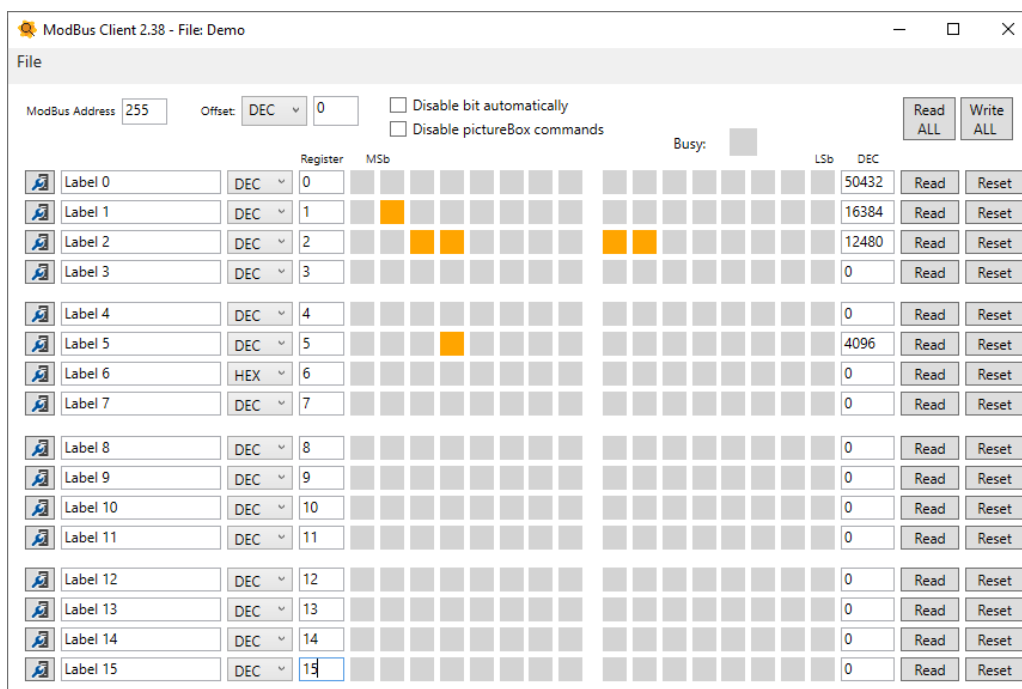


Figura 6.1: Tool comandi bit

Nella finestra Tool command bit è possibile leggere e visualizzare i registri nei singoli bit che li compongono. Premendo sul singolo bit è possibile invertirne lo stato (di default lavorano come un "passo passo" altrimenti se è flaggata l'opzione "disabilita bit automaticamente" al click del mouse il bit è forzato a 1 e successivamente a 0).

Premendo i pulsanti a fianco delle etichette si apre la finestra visualizzata a seguire con la quale è possibile dare un'etichetta specifica ai singoli bit all'interno di una word. Le etichette in questione fanno riferimento solo alla finestra della pagina precedente, non vengono visualizzate nella tab "Holding registers FC 03" della finestra principale.

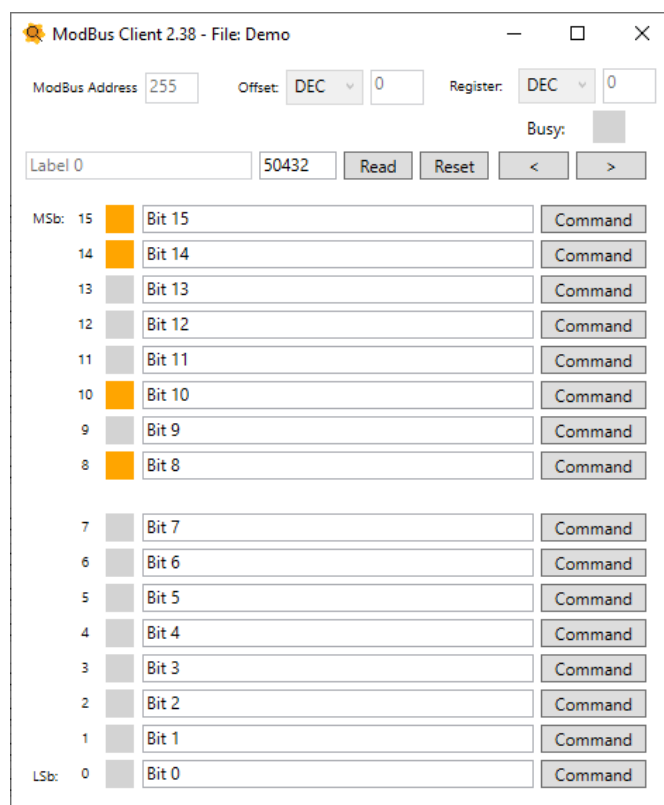


Figura 6.2: Tool comandi bit - Label Bits

Le etichette assegnate ai singoli bit diventano poi tooltip dei bit nella visualizzazione principale riportata nell'immagine 6.1. I pulsanti "<" e ">" in alto a destra permettono di scorrere fra le word della finestra principale.

6.2 Tool comandi byte

ModBus Client 2.38 - File: Demo

File

ModBus Address: 255 Offset: DEC 0 Busy: ☐

Register Value

Register	Value
MSB: Label 0 LSB: Label 1	DEC 0 DEC 0
MSB: Label 2 LSB: Label 3	DEC 1 DEC 0
MSB: Label 4 LSB: Label 5	DEC 2 DEC 0
MSB: Label 6 LSB: Label 7	DEC 3 DEC 0
MSB: Label 8 LSB: Label 9	DEC 4 DEC 0
MSB: Label 10 LSB: Label 11	DEC 5 DEC 0
MSB: Label 12 LSB: Label 13	DEC 6 DEC 0
MSB: Label 14 LSB: Label 15	DEC 7 DEC 0

Buttons: Read ALL, Write ALL, Read, Write

Figura 6.3: Tool comandi byte

Dalla finestra riportata sopra è possibile inviare comandi ai singoli byte che verranno poi scritti come singole word sul target. E' possibile creare finestre personalizzate con cui inviare comandi a registri riferiti anche a posizioni diverse. I pulsanti "<" e ">" in alto a destra permettono di scorrere fra 4 diversi profili della finestra. Le celle scritte vengono colorate di verde mentre quelle lette di blu.

6.3 Tool comandi word

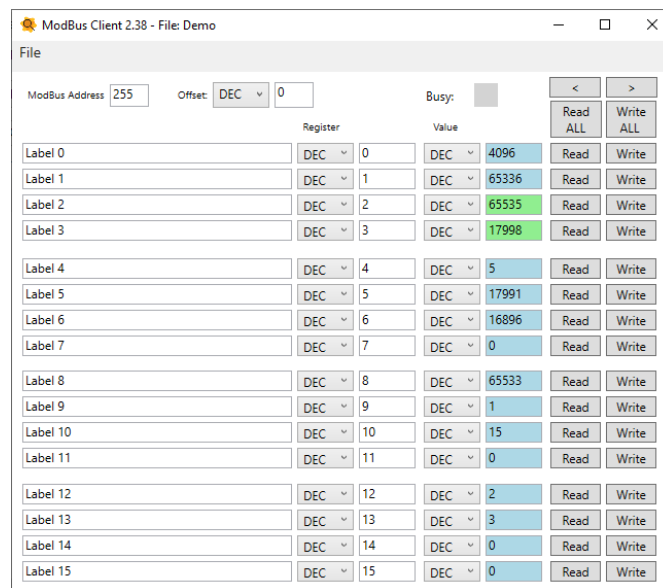


Figura 6.4: Tool comandi word

La finestra word contiene le stesse funzioni viste in precedenza per i singoli bytes, solo divise per word. Come la precedente con i pulsanti "<" e ">" in alto a destra si può scorrere fra 4 diversi profili della finestra. Le celle scritte vengono colorate di verde mentre quelle lette di blu.

7 | Gestione database

7.1 Salvataggio configurazione

Premendo "Salva configurazione nel Database" viene aperta una finestra di dialogo dove inserire il nome della nuova configurazione personalizzata:

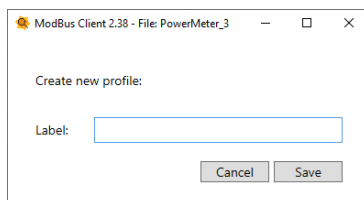


Figura 7.1: Salva nuovo profilo

Premendo "Salva" all'interno della cartella "Json" viene creata una cartella con il nome inserito nella finestra. NON sovrascrivere cartelle esistenti, per apportare modifiche ad una configurazione personalizzata è sufficiente aprirla, le modifiche verranno salvate automaticamente alla chiusura della finestra principale (eventualmente l'utente può salvare lo stato attuale dal menu File->Salva)

7.2 Percorso configurazione

Nella cartella "Json" sono presenti le cartelle contenenti le configurazioni personalizzate, una cartella per ogni profilo inserito. La cartella "Default" contiene la configurazione del programma quando non vengono utilizzate configurazioni personalizzate (se si utilizza il programma senza caricare un profilo, ogni modifica viene salvata in questa cartella). Le altre cartelle vengono generate quando si salva la configurazione attuale come un nuovo profilo.

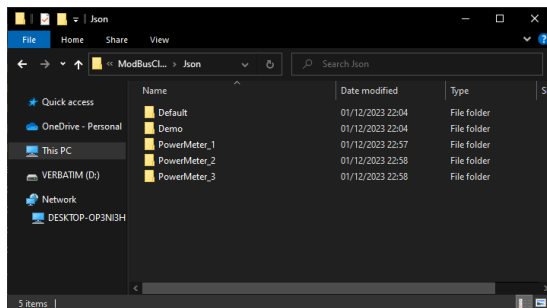


Figura 7.2: Directory profili database

L'utente nell'uso normale non ha bisogno di apportare modifiche direttamente in questa cartella, è sufficiente usare i form per salvare e caricare i profili direttamente dalla finestra principale.

7.3 Caricamento configurazione

Premendo "Carica configurazione dal Database" è possibile caricare una configurazione personalizzata precedentemente salvata:

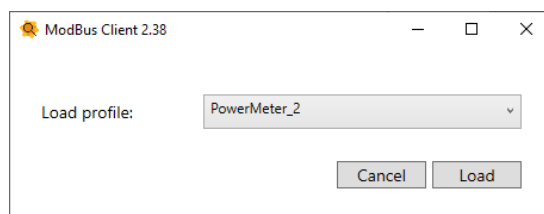


Figura 7.3: Carica profilo

Una volta caricata una configurazione personalizzata qualsiasi modifica inserita nel programma verrà salvata nella cartella personalizzata senza modificare la struttura della configurazione "Default". Per importare o esportare un Profilo personalizzato utilizzare il tool "Manage database", qui è possibile esportare un profilo come file .zip, importare un profilo esportato da un altro client o semplicemente selezionare il profilo da caricare. Quando si chiude la finestra viene caricato nel client il profilo attualmente selezionato.

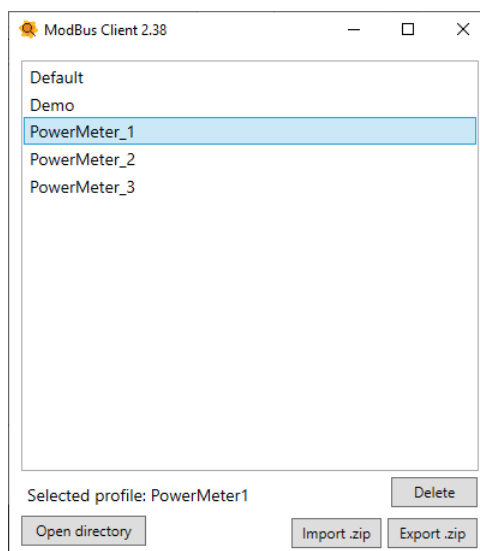


Figura 7.4: Database manager

A partire dalla versione v2.37 i profili possono essere caricati direttamente nella tab home tramite l'apposito menu a tendina:

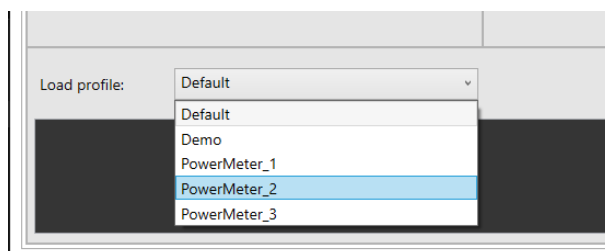


Figura 7.5: Selezione profilo homepage

8 | Menu a tendina

8.1 Menu File

Il menu file contiene i comandi per salvare la configurazione e aprire/chiusure la console.

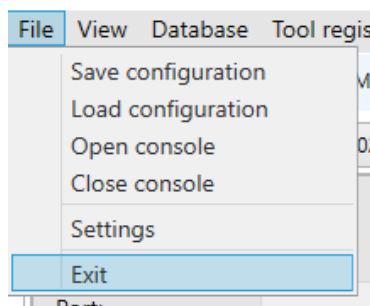


Figura 8.1: Menu File

8.2 Menu View

Nel menu view è possibile cambiare la lingua del programma così come visualizzare e nascondere le colonne dei registri delle varie tab come si vede nell'immagine seguente:

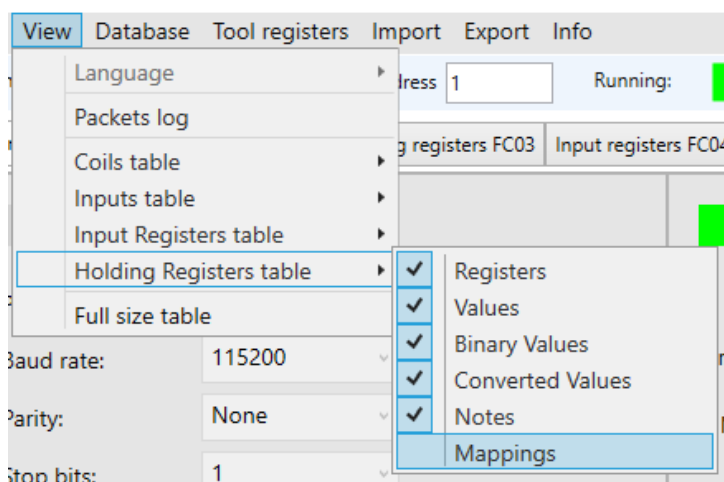


Figura 8.2: Menu View

8.3 Menu Database

Nel menu database è possibile creare/caricare un profilo e accedere al tool di import/export dei profili. L'ultima voce del menu "Open temprate editor" apre la finestra di modifica dei template personalizzati.

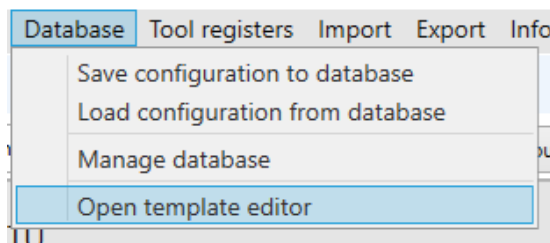


Figura 8.3: Menu Database

8.4 Menu Tools

Nel menu tools è possibile aprire le finestre di comando per pilotare singoli bit/byte/word.

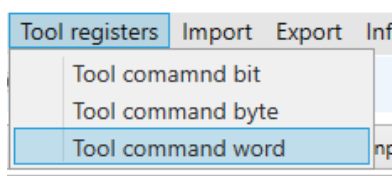


Figura 8.4: Menu Tools

8.5 Menu Import

Nel menu import è possibile inportare una tabella c oils o holding registers esportata precedentemente inviarla inviarla allo slave. Nel caso in cui i registri siano consecutivi è possibile scegliere se inviarli singolarmente (write single coil/write single register) o in blocco (write multiple coils/write multiple registers).

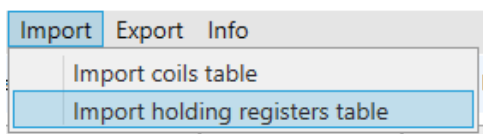


Figura 8.5: Menu Import

8.6 Menu Export

Nel menu export è possibile esportare le tabelle delle varie schede in formato csv.

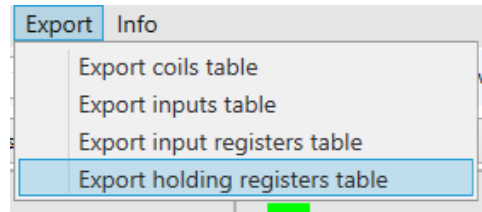


Figura 8.6: Menu Export

8.7 Menu Info

Dal menu info è possibile aprire questa guida, visualizzare la licenza del programma e data/numero di build.

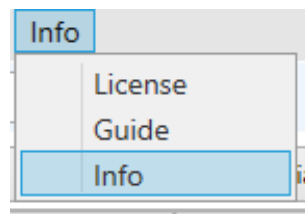


Figura 8.7: Menu Info

Nella finestra info è possibile leggere, oltre alla versione corrente, anche data e ora di build.

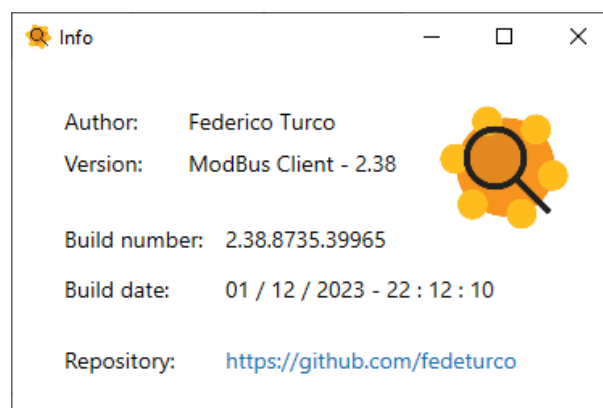


Figura 8.8: Finestra Info

9 | Tasti di scelta rapida

- **Ctrl + Num 1 - 9:** Seleziona la tab con l'indice selezionato dal numero premuto
- **Ctrl + Q/Ctrl + W:** Chiude la finestra
- **Ctrl + E:** Legge il range di registri della tab selezionata
- **Ctrl + R:** Legge i registri della tab selezionata (inputs/coils/input registers/holding registers)
- **Ctrl + T:** Apre la finestra di modifica template per il profilo attualmente selezionato
- **Ctrl + Y:** Apre finestra di import csv/json
- **Ctrl + U:** Apre finestra di export csv/json
- **Ctrl + I:** Apre finestra info versione software
- **Ctrl + O:** Apre il menu per caricare un profilo
- **Ctrl + P:** Avvia/Ferma il polling per la tab selezionata (pulsante loop)
- **Ctrl + S:** Salva eventuali modifiche al profilo attualmente selezionato
- **Ctrl + Shift + S:** Apre il menu per salvare il profilo corrente come nuovo profilo
- **Ctrl + D:** Apre la finestra di gestione del database
- **Ctrl + F:** Attiva/disattiva modalità schermo intero tabelle
- **Ctrl + G:** Legge il gruppo selezionato nel menu a tendina per la tab corrente
- **Ctrl + H:** Legge tutte le risorse inserite a template per la tab corrente
- **Ctrl + K:** Avvia/Ferma il polling per la tab selezionata sul range indicato (pulsante loop)
- **Ctrl + L:** Apre finestra di log
- **Ctrl + Shift + C:** Apre chiude console di debug client
- **Ctrl + B:** Connetti/Disconnetti
- **Ctrl + M:** Passa da TCP a RTU e viceversa
- **Del:** Cancella contenuto tabelle della tab corrente

10 | Modbus Secure

Il protocollo Modbus Secure utilizza gli stessi frame del protocollo TCP standard incapsulati tramite TLS. Il protocollo TLS fornisce un sistema di autenticazione tramite certificati x.509v3. La normativa richiede TLS con versione 1.2 o superiore (attualmente siamo alla versione 1.3), il client supporta entrambe le versioni (1.2 e 1.3). L'utilizzo dei certificati richiede la creazione di un certificato e di una chiave privata per il server, così come per ogni client che si collegherà al gateway. Si riporta di seguito una schema per chiarire i ruoli dei certificati sia lato server che client:

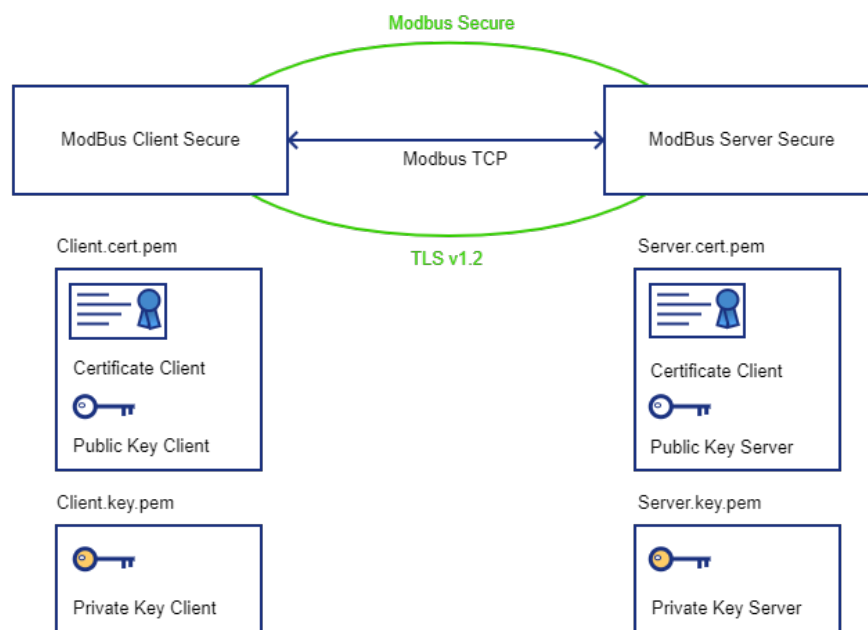


Figura 10.1: Schema certificati Modbus Secure

La normativa richiede di aggiungere al certificato x.509v3 alcune estensioni tra cui un OID (Object Identifier standardized by the International Telecommunications Union) per definire il ruolo del client al momento dell'autenticazione. La specifica prevede infatti che tutti i client possano utilizzare funzioni di lettura ma solo i client con il ruolo di "operator" possano utilizzare funzioni di scrittura (di coils o holding registers).

10.1 Specifiche normativa

Si riportano di seguito le specifiche principali della normativa (MB-TCP_Security-v21_2018-07-24):

- Il protocollo utilizzato deve essere \geq TLS 1.2
- Cipher suite: RSA per key exchange
- Cipher suite: AES128 per cifratura dei pacchetti
- Cipher suite: SHA256SUM per il controllo di integrità dei messaggi
- Default cipher: TLS_RSA_WITH_AES_128_CBC_SHA256

10.2 Estensioni X509v3

Si riporta di seguito il contenuto delle estensioni che richiede la normativa da aggiungere ad un certificato X509v3 standard:

X509v3 extensions:

X509v3 Subject Key Identifier:

38:A4:CC:19:6D:6D:E2:AA:7C:82:75:44:A0:59:39:81:47:D3:13:F0

X509v3 Authority Key Identifier:

keyid:38:A4:CC:19:6D:6D:E2:AA:7C:82:75:44:A0:59:39:81:47:D3:13:F0

X509v3 Basic Constraints:

CA:FALSE

X509v3 Key Usage: critical

Digital Signature, Non Repudiation, Key Encipherment

1.3.6.1.4.1.50316.802.1:

..Operator

X509v3 Subject Alternative Name:

IP Address:192.168.1.10

10.3 Generazione dei certificati

Gli step per generare i certificati con openssl sono i seguenti (sono gestite le righe multiple per cui si possono copiare e incollare i comandi direttamente sulla shell):

Generare certificato e chiave privata per il server (modificando l'ip con l'indirizzo corretto del server):

```
# Server
openssl req -x509 -newkey rsa:4096 -sha256 -days 360 \
-keyout server.key.pem -out server.cert.pem \
-nodes -subj "/C=IT/ST=Italy/L=Rovereto/O=ModBusServer/OU=ModBusServer/CN=ModbusSecurityServer" \
-addext "keyUsage=critical,digitalSignature,nonRepudiation,keyEncipherment" \
-addext "subjectAltName=IP:192.168.1.20"
```

Generare uno o più certificati/chiavi private per i client che vorranno collegarsi al server (si riportano tre esempi di seguito, anche in questo caso va aggiornato l'ip):

```
# Client 1:
openssl req -x509 -newkey rsa:4096 -sha256 -days 360 \
-keyout client1.key.pem -out client1.cert.pem -nodes \
-subj "/C=IT/ST=Italy/L=Rovereto/O=ModBusClient/OU=ModBusClient/CN=ModbusSecurityClient" \
-addext "keyUsage=critical,digitalSignature,nonRepudiation,keyEncipherment" \
-addext "1.3.6.1.4.1.50316.802.1=ASN1:UTF8String:Operator" \
-addext "subjectAltName=IP:192.168.1.10"
```

Su windows molto spesso l'abbinata certificato+chiave private viene unita in un unico file protetto da password (.pfx), Per unire certificato e chiave in un file pfx usare il comando seguente:

```
openssl pkcs12 -export -out client1.pfx -inkey client1.key.pem -in client1.cert.pem
```

Per aprire un certificato visualizzare il contenuto si può usare il comando:

```
openssl x509 -in client1.cert.pem -text
```

11 | Altre note

Il numero massimo di indirizzi che è possibile leggere con un unico comando è pari a 123 per richieste TCP o 125 per richieste RTU.

Se si inserisce un numero maggiore di registri viene visualizzato un messaggio di errore.