6 Protocollo modbus

Il modbus è un protocollo di comunicazione seriale diventato uno standard de facto nella comunicazione di tipo industriale, ed è ora il protocollo di connessione più diffuso fra i dispositivi elettronici industriali. E' un protocollo di tipo richiesta/risposta ed offre dei servizi specificati da function codes.

SlimLine supporta il protocollo modbus Rtu sulle porte seriali, e modbus Over IP su connessione ethernet su porta ***502***. Il protocollo modbus Rtu sulla porta seriale ha come parametri di comunicazione di default ***115200, 8, e***, e l'indirizzo di nodo sia su porta seriale che su TCP/IP è ***1***.

6.1 Accesso variabili da modbus

Le funzioni del protocollo accedono tutte alla memoria utente ***MX100***, le funzioni supportate sono:

Code	Function	Tipo oggetto	Tipo accesso	Range indirizzo
01h	Read coil status	Bit singolo	Read	40000-44095 ***(Note 1)***
02h	Read input status	Bit singolo	Read	40000-44095 ***(Note 1)***
03h	Read holding registers	Word (16 Bit)	Read	40000-42047 ***(Note 2)***
04h	Read input registers	Word (16 Bit)	Read	40000-42047 ***(Note 2)***
05h	Force single coil	Bit singolo	Write	40000-44095 ***(Note 1)***
06h	Preset single register	Word (16 Bit)	Write	40000-42047 ***(Note 2)***
10h	Preset multiple registers	Word (16 Bit)	Write	40000-42047 ***(Note 2)***

Note 1) Nelle funzioni che accedono al bit singolo (In realtà ogni bit equivale ad un byte di memoria) si utilizza nel comando l'indirizzo della variabile, quindi dovendo accedere alla locazione ***MX100.50*** utilizzeremo come indirizzo il valore ***40050***.
Note 2) Nelle funzioni che accedono ai registri (16 Bits) occorre considerare l'indirizzo della variabile diviso per 2, quindi dovendo raggiungere da modbus la locazione ***MX100.50*** utilizzeremo come indirizzo il valore ***40025***.

6.1.1 Lettura variabili da modbus

Per la lettura delle variabili si utilizza il comando ***Read holding registers*** (Codice ***0x03***). Ipotizzando di dover accedere in lettura ad una variabile ***DWORD*** allocata in memoria all'indirizzo ***MX100.64*** calcoleremo l'indirizzo di lettura nel modo:

((Indirizzo variabile/2)+Offset)-1 → ((64/2)+40000)-1=40031 → 0x9C5F

Essendo una variabile ***DWORD*** dovremo leggere 2 registri consecutivi a partire dal suo indirizzo di allocazione, ipotizzando che il valore della variabile sia ***0x12345678*** avremo.

Frames modbus RTU

Stringa di comando: ***01 03 9C 5F 00 02 DA 49***
Stringa di risposta: ***01 03 04 56 78 12 34 66 D5***

Frames modbus TCP/IP

Stringa di comando: ***00 00 00 00 00 06 01 03 9C 5F 00 02***
Stringa di risposta: ***00 00 00 00 00 07 01 03 04 56 78 12 34***

La rappresentazione dei dati in SlimLine è nel formato ***Little-Endian***, la numerazione inizia dal byte meno significativo per finire col più significativo, quindi come si nota dalla stringa di risposta il valore della variabile a 32 bits ***0x12345678*** viene ritornato suddiviso in due registri a 16 bits con i valori ***0x5678***, ***0x1234***.

6.1.2 Scrittura variabili da modbus

Per la scrittura delle variabili si utilizza il comando ***Preset multiple registers*** (Codice ***0x10***). Ipotizzando di dover accedere in scrittura ad una variabile ***DWORD*** allocata in memoria all'indirizzo ***MX100.64*** calcoleremo l'indirizzo di scrittura nel modo:

((Indirizzo variabile/2)+Offset)-1 → ((64/2)+40000)-1=40031 → 0x9C5F

Essendo una variabile ***DWORD*** dovremo scrivere 2 registri consecutivi a partire dal suo indirizzo di allocazione, ipotizzando di dover scrivere nella variabile il valore ***0x12345678*** avremo.

Frames modbus RTU

Stringa di comando: ***01 10 9C 5F 00 02 04 56 78 12 34 D3 33***
Stringa di risposta: ***01 10 9C 5F 00 02 5F 8A***

Frames modbus TCP/IP

Stringa di comando: ***00 00 00 00 00 0B 01 10 9C 5F 00 02 04 56 78 12 34***
Stringa di risposta: ***00 00 00 00 00 06 01 10 9C 5F 00 02***

La rappresentazione dei dati in SlimLine è nel formato ***Little-Endian***, la numerazione inizia dal byte meno significativo per finire col più significativo, quindi come si nota dalla stringa di comando il valore da scrivere a 32 bits ***0x12345678*** viene definito suddiviso in due registri a 16 bits con i valori ***0x5678***, ***0x1234***.

6.2 Accesso Real time clock da modbus

E' possibile accedere al real time clock utilizzando i comandi modbus di accesso ai registri le funzioni supportate sono:

Code	Function	Tipo oggetto	Tipo accesso	Range indirizzo
03h	Read holding registers	Word (16 Bit)	Read	100-105 (150 Epoch time)
04h	Read input registers	Word (16 Bit)	Read	100-105 (150 Epoch time)
06h	Preset single register	Word (16 Bit)	Write	100-105 (150 Epoch time)
10h	Preset multiple registers	Word (16 Bit)	Write	100-105 (150 Epoch time)

I registri (16 Bits) del real time clock sono allocati in locazioni consecutive a partire dall'indirizzo modbus 100. I registri contengono il valore attuale del real time clock e scrivendo un nuovo valore il real time clock verrà automaticamente aggiornato.

Address	Register	Note
100	Second	Valore secondi (Range da 0 a 59)
101	Minute	Valore minuti (Range da 0 a 59)
102	Hour	Valore ora (Range da 0 a 23)
103	Day	Valore giorno (Range da 1 a 31)
104	Month	Valore mese (Range da 1 a 12)
105	Year	Valore anno (Range da 1900 a 2037)

6.2.1 Lettura RTC da modbus

Per la lettura dei registri del real time clock si utilizza il comando ***Read holding registers*** (Codice ***0x03***). Dovremo leggere 6 registri consecutivi a partire dall'indirizzo di allocazione, l'indirizzamento di modbus prevede un offset di 1, quindi ***99*** (***0x0063***).

Frames modbus RTU

Stringa di comando: ***01 03 00 63 00 06 35 D6***
Stringa di risposta: ***01 03 0C 00 1E 00 30 00 0B 00 1D 00 09 07 DA A2 32***

Frames modbus TCP/IP

Stringa di comando: ***00 00 00 00 00 06 01 03 00 63 00 06***
Stringa di risposta: ***00 00 00 00 00 0F 01 03 0C 00 1E 00 30 00 0B 00 1D 00 09 07 DA***

Come si vede dalla risposta il valore è:

Secondi: 30 (***0x001E***)
Minuti: 48 (***0x0030***)
Ora: 11 (***0x000B***)
Giorno: 29 (***0x001D***)
Mese: 9 (***0x0009***)
Anno: 2010 (**0x07DA**)

6.2.2 Scrittura RTC da modbus

Per la scrittura dei registri del real time clock si utilizza il comando ***Preset multiple registers*** (Codice ***0x10***). Dovremo scrivere 6 registri consecutivi a partire dall'indirizzo di allocazione, l'indirizzamento di modbus prevede un offset di 1, quindi ***99*** (***0x0063***). Ipotizziamo di dover impostare nel real time clock i valori:

Secondi: 30 (***0x001E***)
Minuti: 48 (***0x0030***)
Ora: 11 (***0x000B***)
Giorno: 29 (***0x001D***)
Mese: 9 (***0x0009***)
Anno: 2010 (**0x07DA**)

Frames modbus RTU

Stringa di comando: ***01 10 00 63 00 06 08 00 1E 00 30 00 0B 00 1D 00 09 07 DA 5D C8***
Stringa di risposta: ***01 10 00 63 00 06 B0 15***

Frames modbus TCP/IP

Stringa di comando: ***00 00 00 00 00 13 01 10 00 63 00 06 08 00 1E 00 30 00 0B 00 1D 00 09 07 DA***
Stringa di risposta: ***00 00 00 00 00 06 01 10 00 63 00 06***

6.3 Accesso Epoch time da modbus

E' allocato anche un registro a 32 bits per il valore di data/ora in Epoch time, l'acceso a questo registro in lettura e/o scrittura và sempre effettuato usando due registri a 16 bits.

Address	Register	Note
150	Epoch time	Epoch time

6.3.1 Lettura Epoch time da modbus

Per la lettura dell'epoch time si utilizza il comando ***Read holding registers*** (Codice ***0x03***). Dovremo leggere 2 registri consecutivi a partire dall'indirizzo di allocazione, l'indirizzamento di modbus prevede un offset di 1, quindi ***149*** (***0x0095***).

Frames modbus RTU

Stringa di comando: ***01 03 00 95 00 02 D4 27***
Stringa di risposta: ***01 03 04 30 B5 4C A3 90 6C***

Frames modbus TCP/IP

Stringa di comando: ***00 00 00 00 00 06 01 03 00 95 00 02***
Stringa di risposta: ***00 00 00 00 00 07 01 03 04 30 B5 4C A3***

Come si vede dalla risposta il valore è: ***0x4CA330B5 → 1285763253 → GMT: Wed, 29 Sep 2010 12:27:33 UTC***.

6.3.2 Scrittura Epoch time da modbus

Per la scrittura dell'epoch time si utilizza il comando ***Preset multiple registers*** (Codice ***0x10***). Dovremo scrivere 2 registri consecutivi a partire dall'indirizzo di allocazione, l'indirizzamento di modbus prevede un offset di 1, quindi ***149*** (***0x0095***). Ipotizziamo di dover impostare il valore:

GMT: Wed, 29 Sep 2010 12:27:33 UTC → 1285763253 → 0x4CA330B5

Frames modbus RTU

Stringa di comando: ***01 10 00 95 00 02 04 30 B5 4C A3 50 A3***
Stringa di risposta: ***01 10 00 95 00 02 51 E4***

Frames modbus TCP/IP

Stringa di comando: ***00 00 00 00 00 0B 01 10 00 95 00 02 04 30 B5 4C A3***
Stringa di risposta: ***00 00 00 00 00 06 01 10 00 95 00 02***