Il protocollo MODBUS

TABLE OF CONTENT

Sommario

IL PROTOCOLLO MODBUS

II protocollo MODBUS definisce il formato e la modalità di comunicazione tra un "master" che gestisce il sistema e uno o più "slave" che rispondono alle interrogazioni del master.
II protocollo definisce come il master e gli slave stabiliscono ed interrompono la comunicazione, come trasmettitore e ricevitore devono essere identificati, come i messaggi devono venire scambiati e come gli errori rilevati.
Si possono connettere un master e fino a 247 slave su una linea comune , occorre notare che questo è un limite logico del protocollo, l'interfaccia fisica può peraltro limitare ulteriormente il numero di dispositivi, per esempio l'interfaccia standard RS-485 prevede un massimo di 31 slave connessi alla linea.
Sostituendo l'ultimo elemento della linea con un apposito "bridge o ripetitore", si possono connettere altri 31 slave e cosi via sino al raggiungimento del numero massimo logico di dispositivi applicati.
Solo il master può iniziare una transazione.
Una transazione può avere il formato domanda/risposta diretta ad un singolo slave o broadcast in cui il messaggio viene inviato a tutti i dispositivi sulla linea che non dannorisposta.
Una transazione è composta da una struttura singola domanda/singola risposta o una struttura singolo messaggio broadcast/nessuna risposta.
Alcune caratteristiche del protocollo sono definite e sono :

- standard di interfaccia
- parità
- numero di stop bit
- ed il formato RTU (binario).

Esiste anche il protocollo MODBUS di tipo ASCII ma normalmente viene implementato il modo RTU in quanto più efficiente.
Il protocollo JBUS è funzionalmente identico al MODBUS e se ne differenzia per la diversa numerazione degli indirizzi:
nel MODBUS questi partono da zero (0000 = 1° indirizzo) mentre nel JBUS partono da uno (0001 = 1° indirizzo) mantenendo questo scostamento per tutta la numerazione.
Nel seguito, se non esplicitamente menzionato, pur facendo riferimento al MODBUS la descrizione si considera valida per entrambi i protocolli,

1. FORMATO DEI MESSAGGI

Per poter comunicare tra due dispositivi, il messaggio deve essere contenuto in un "involucro" L'involucro lascia il trasmettitore attraverso una "porta" ed è "portato" lungo la linea fino ad una analoga "porta" sul ricevitore.

MODBUS stabilisce il formato di questo involucro che, tanto per il master che per lo slave. comprende:

- L'indirizzo del dispositivo con cui il master ha stabilito la transazione (l'indirizzo O corrisponde ad un messaggio broadcast inviato a tutti i dispositivi slave).
- II codice della funzione che deve essere o è stata eseguita.
- I dati che devono essere scambiati.
- II controllo d'errore composto secondo l'algoritmo CRC16.

Se un dispositivo individua un errore nel messaggio ricevuto (di formato, di parità o nelCRC16) il messaggio viene considerato non valido e scartato, uno slave che rilevi un errorenel messaggio quindi non eseguirà l'azione e non risponderà alla domanda, cosi come se l'indirizzo non corrisponde ad un dispositivo in linea.

2.1 Formato dei caratteri

Normalmente i dispositivi che adottano il protocollo MODBUS utilizzano in formato 8, N, 1
Ovvero : 8 bit di dati, senza alcun controllo di parità e con 1 bit di stop.

2.2 L'indirizzo

Come sopra menzionato, le transazioni MODBUS coinvolgono sempre il master,che gestisce la linea, ed uno slave per volta (tranne nel caso di messaggi broadcast).
Per identificare il destinatario del messaggio viene trasmesso come primo carattere un byte che contiene l'indirizzo numerico del dispositivo slave selezionato.
Ciascuno degli slave quindi avrà assegnato un diverso numero di indirizzo che lo identifica univocamente.
Gli indirizzi ammissibili sono quelli da 1 a 247, mentre l'indirizzo 0, che non può essere assegnato ad uno slave, posto In testa al messaggio trasmesso dal master indica che questo è "broadcast", cioè diretto a tutti gli slave contemporaneamente.
Possono essere trasmessi come broadcast solo messaggi che non richiedano risposta per espletare la loro funzione, quindi solo le assegnazioni.

2.3 II codice funzione

II secondo carattere del messaggio identifica la funzione che deve essere eseguita nel messaggio trasmesso dal master, cui lo slave risponde a sua volta con lo stesso codice ad indicare che la funzione è stata eseguita.
Normalmente le MODBUS più utilizzate sono quelle riportate di seguito :

Funzione	Descrizione
01	Read Coil Status
02	Read Input Status
03	Read Holding Registers
04	Read Input registers
05	Force Single Coil
06	Prese! Single register
07	Read Status
15	Force multiple Coils
16	Preset Multiple Registers

2.4 Il CRC16

Gli ultimi due caratteri del messaggio contengono il codice di ridondanza ciclica (Cyclic Redundancy Check) calcolato secondo l'algoritmo CRC16.
Per il calcolo di questi due caratteri il messaggio (indirizzo, codice funzione e dati scartando i bit di start, stop e l'eventuale parità) viene considerato come un unico numero binario continuo di cui il bit più significativo (MSB) viene trasmesso prima.
Il messaggio viene innanzitutto moltiplicato per 216 (spostato a sinistra di 16 bit) e poi diviso per 216+215+22+1 espresso come numero binario (1100000000000101).
Il quoziente intero viene poi scartato e il resto a 16 bit (inizializzato a FFFFh all'inizio per evitare il caso di un messaggio di soli zeri) viene aggiunto di seguito al messaggio trasmesso.
Il messaggio risultante, quando diviso dal dispositivo ricevente per lo stesso polinomio (216+215+22+1) deve dare zero come resto se non sono Intervenuti errori (il dispositivo ricevente ricalcola il CRC).
Di fatto, dato che il dispositivo che serializza i dati da trasmettere (UART) trasmette prima il bit meno significativo (LSB) anziché il MSB come dovrebbe essere per il calcolo del CRC, questo viene effettuato invertendo il polinomio.
Inoltre, dato che il MSB del polinomio influenza solo il quoziente e non il resto, questo viene eliminato rendendolo quindi 1010000000000001.
La procedura passo-passo per il calcolo del CRC16 è la seguente:

1. Caricare un registro a 16 bit con FFFFh (tutti i bit a 1).
2. Fare l'OR esclusivo del primo carattere con il byte superiore del registro , porre il risultato nel registro.
3. Spostare il registro a destra di un bit.
4. Se il bit uscito a destra dal registro (flag) è un 1, fare l'OR esclusivo del polinomio generatore 1010000000000001 con il registro.
5. Ripetere per 8 volte i passi 3 e 4.
6. Fare l'OR esclusivo del carattere successivo con il byte superiore del registro, porre il risultato nel registro.
7. Ripetere i passi da 3 a 6 per tutti i caratteri del messaggio.
8. Il contenuto del registro a 16 bit è il codice di ridondanza CRC che deve essereaggiunto al messaggio

2.5 Sincronizzazione dei messaggi

La sincronizzazione del messaggio tra trasmettitore e ricevitore viene ottenuta interponendo una pausa tra i messaggi pari ad almeno 3.5 volte il tempo di un carattere.
Se il dispositivo ricevente non riceve per un tempo di 3,5 caratteri, ritiene completato il messaggio precedente e considera che il successivo byte ricevuto sarà il primo di un nuovo messaggio e quindi un indirizzo.

LE FUNZIONI MODBUS

Viene riportata di seguito la descrizione dettagliata delle funzioni MODBUS più utilizzate

3.1 Read Output Status (01)

Questa funzione permette di richiedere lo stato ON o OFF di variabili logiche binarie.
Il modo broadcast non è permesso.

Domanda

Oltre all'indirizzo dello slave e al codice funzione (01) il messaggio contiene l'indirizzo di partenza (starting Address) espresso su due byte e il numero di bit da leggere anch'esso su due byte. La numerazione degli indirizzi parte da zero (bit1 = 0) per il MODBUS, da uno (bit1 = 1) per il JBUS.

Esempio: Richiesta di lettura dallo slave 17 del bit dal 0004 al 0015.

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA bit # HI	DATA bit # LO	CRC HI	*CRC* LO
11	01	00	03	00	0C	CE	9F

Risposta

Oltre all'indirizzo dello slave e al codice funzione (01) il messaggio comprende un carattere che contiene il numero di byte di dati e i caratteri contenenti i dati.
I dati sono impaccati, così che un byte contiene lo stato di 8 bit, il bit meno significativo del primo byte contiene il bit corrispondente allo starting Address e così via.
Se il numero di bit da leggere non è multiplo di 8, l'ultimo carattere è completato con zeri nei bit più significativi.

Esempio: Risposta alla richiesta sopra riportata.

ADDR	FUNC	DATA byte count	DATA bit 04..11	DATA bit 12..15	CRC HI	*CRC* LO
11	01	02	CD	0B	6D	68

3.2 Read Input Status (02)

Questa funzione è operativamente identica alla precedente.

3.3 Read Output Registers (03)

Questa funzione permette di richiedere II valore di registri a 16 bit (word) contenenti variabili numeriche.
Il modo broadcast non è permesso.

Domanda

Oltre all'indirizzo dello slave e al codice funzione (03) il messaggio contiene l'indirizzo di partenza (starting Address) espresso su due byte e il numero di word da leggere anch'esso su due byte. Il numero massimo dì word che possono essere lette è 125. La numerazione degli indirizzi parte da zero (word1= 0) per il MODBUS, da uno (word1 =1)per il JBUS

Esempio: Richiesta di lettura dallo slave 25 dei registri da 069 a 0071.

ADDR	**FUNC**	**DATA start Addr HI**	**DATA start Addr LO**	**DATA bit # HI**	**DATA bit # LO**	**CRC HI**	***CRC* LO**
19	03	00	44	00	03	46	06

Risposta

Oltre all'indirizzo dello slave e al codice funzione (03) il messaggio comprende un carattere che contiene il numero di byte di dati e i caratteri contenenti i dati.
I registri richiedono due byte ciascuno, il primo dei quali contiene la parte più significativa.

Esempio: Risposta alla richiesta sopra riportata.

ADDR	**FUNC**	**DATA byte count**	**DATA byte 69 HI**	**DATA byte 69 LO**	**DATA byte 70 HI**	**DATA byte 70 LO**	**DATA byte 71 HI**	**DATA byte 71 LO**	**CRC HI**	***CRC* LO**
19	03	06	02	2B	00	00	00	64	AF	7A

3.4 Read Input Registers (04)

Questa funzione è operativamente identica alla precedente.

3.4 Force Single Coil (05)

Questa funzione permette di forzare lo stato di una singola variabile binaria ON o OFF.
Il modo broadcast è permesso.

Domanda

Oltre all'indirizzo dello slave e al codice funzione (05) il messaggio contiene l'indirizzo della variabile da forzare su due byte e due caratteri di cui il primo è posto a FFh (255) per forzare lo stato ON e OOh per forzare OFF, il secondo è posto a zero in ogni caso.
La numerazione degli indirizzi parte da zero (bit1 = 0) per il MODBUS. da uno (bit1 = 1) per il JBUS.

Esempio: Richiesta di forzare ON sullo slave 47 il bit 4.

ADDR	FUNC	DATA bit HI	DATA bit LO	DATA ON / OFF	DATA (zero)	CRC HI	*CRC* LO
2F	05	00	03	FF	00	7A	74

Risposta

La risposta consiste nel ritrasmettere il messaggio ricevuto dopo che la variabile è stata modificata.

Esempio: Risposta alla richiesta sopra riportata.

ADDR	FUNC	DATA bit HI	DATA bit LO	DATA ON / OFF	DATA (zero)	CRC HI	*CRC* LO
2F	05	00	03	FF	00	7A	74

3.5 Preset Single Register (06)

Questa funzione permette di impostare il valore di un singolo registro a 16 bit.
Il modo broadcast è permesso.

Domanda

Oltre all'indirizzo dello slave e al codice funzione (06) il messaggio contiene l'indirizzo della variabile espresso su due byte e il valore che deve essere assegnato.
La numerazione degli indirizzi parte da zero (word1 = 0) per il MODBUS, da uno (word1 = 1) per il JBUS.

Esempio: Richiesta di forzare 928 sullo slave 35 all'indirizzo 26.

ADDR	FUNC	DATA bit # HI	DATA bit # LO	DATA Word HI	DATA Word LO	CRC HI	*CRC* LO
23	06	00	19	03	A0	5E	07

Risposta

La risposta consiste nel ritrasmettere il messaggio ricevuto dopo che la variabile è stata modificata.

Esempio: Risposta alla richiesta sopra riportata.

ADDR	FUNC	DATA bit # HI	DATA bit # LO	DATA Word HI	DATA Word LO	CRC HI	*CRC* LO
23	06	00	19	03	A0	5E	07

3.6 Read Status (07)

Questa funzione permette di leggere lo stato di otto bit predeterminati con un messaggio compatto.
Il modo broadcast non è permesso.

Domanda

II messaggio comprende solo l'indirizzo dello slave e il codice funzione (07).

Esempio: Richiesta dello stato dallo slave 25.

ADDR	FUNC	CRC HI	*CRC* LO
19	07	5E	07

Risposta

Oltre all'indirizzo dello slave e al codice funzione (07) il messaggio comprende un carattere che contiene i bit di stato.

Esempio: Risposta alla richiesta sopra riportata.

ADDR	FUNC	CRC HI	CRC HI	*CRC* LO
19	07	6D	63	DA

3.7 Force Multiple Coils (15)

Questa funzione permette di forzare lo stato di ciascuna variabile binaria in un blocco consecutivo.
Il modo broadcast è permesso.

Domanda

Oltre all'indirizzo dello slave e al codice funzione (15) il messaggio contiene l'indirizzo di partenza (starting Address) espresso su due byte, il numero di bit da scrivere, il numero di byte che contengono i dati e i caratteri di dati, i dati sono impaccati, cosi che un byte contiene lo stato di 8 bit, il bit meno significativo del primo byte deve contenere il bit corrispondenteallo starting Address e cosi via.
Se il numero di bit da scrivere non è multiplo di 8, l'ultimo carattere va completato con zeri nei bit più significativi.
La numerazione degli indirizzi parte da zero (bit1 = 0) per il MODBUS. da uno (bit1 = 1) per il JBUS.

Esempio: Richiesta di forzare, sullo slave 12, 4 bit a partire dall'indirizzo 1.
I bit 1 e 4 forzati a "1", gli altri a "O".

ADDR	FUNC	DATA start ADDR HI	DATA start ADDR LO	DATA bit # HI	DATA bit # LO	DATA byte Count	DATA bit 1..4	CRC HI	*CRC* LO
0C	0F	00	00	00	04	01	09	3F	09

Risposta

Oltre all'indirizzo dello slave e al codice funzione (15) il messaggio comprende l'indirizzo di partenza (starting Address) e il numero di bit scritti.

Esempio: Risposta alla richiesta sopra riportata.

ADDR	FUNC	DATA start ADDR HI	DATA start ADDR LO	DATA bit # HI	DATA bit # LO	CRC HI	*CRC* LO
0C	0F	00	00	00	04	55	15

3.8 Preset Multiple Registers (16)

Questa funzione permette di impostare il valore di un blocco consecutivo di registri a 16 bit. Il modo broadcast è permesso.

Domanda

Oltre all'indirizzo dello slave e al codice funzione (16) il messaggio contiene l'indirizzo di partenza (starting Address), il numero di word da scrivere, il numero di byte che contengono i dati e i caratteri di dati. La numerazione degli indirizzi parte da zero (word1 = 0} per il MODBUS, da uno (word1 = 1) per il JBUS.

Esempio: Richiesta di impostare, sullo slave 17, 1 word all'indirizzo 35. con valore 268.

ADDR	FUNC	DATA start ADDR HI	DATA start ADDR LO	DATA Word # HI	DATA Word # LO	DATA byte Count	DATA Word 35 HI	DATA Word 35 LO	CRC HI	*CRC* LO
11	10	00	22	00	01	02	01	0C	6C	87

Risposta

Oltre all'indirizzo dello slave e al codice funzione (16) il messaggio comprende l'indirizzo di partenza (starting Address) e il numero di word scritte.

Esempio: Risposta alla richiesta sopra riportata.

ADDR	FUNC	DATA start ADDR HI	DATA start ADDR LO	DATA bit # HI	DATA bit # LO	CRC HI	*CRC* LO
11	10	00	22	00	01	A3	53

LA GESTIONE DEGLI ERRORI

In MODBUS esistono due tipi di errori, gestiti in modo diverso: errori di trasmissione ederrori operativi. Gli errori di trasmissione sono errori che alterano il messaggio, nel suo formato, nella parità (se è usata), o nel CRC16.
Il dispositivo che rilevi errori di questo tipo nel messaggio lo considera non valido e non da risposta. Qualora invece il messaggio sia corretto nella sua forma ma la funzione richiesta, per qualsiasi motivo, non sia eseguibile, si ha un errore operativo. A questo errore il dispositivo slave risponde con un messaggio di eccezione.
Questo messaggio è composto dall'indirizzo, dal codice delta funzione richiesta, da un codice d'errore e dal CRC. Per indicare che la risposta è la notifica di un errore il codice funzione viene ritornato con il bit più significativo a"1".

Esempio: Richiesta di lettura dallo slave 10 del bit 1185.

ADDR	**FUNC**	**DATA start Addr HI**	**DATA start Addr LO**	**DATA bit # HI**	**DATA bit # LO**	**CRC HI**	***CRC* LO**
0A	01	04	A1	00	01	AC	63

Risposta

La richiesta chiede il contenuto del bit 1185. che non esiste nello slave.
Questi risponde con il codice d'errore "02" (ILLEGAL DATA ADDRESS) e ritorna il codicefunzione 81h (129).

Esempio: Eccezione alla richiesta sopra riportata.

ADDR	**FUNC**	**DATA exept. code**	**CRC HI**	***CRC* LO**
0A	81	02	B0	53

4.1 Codici d'eccezione

Sotto vengono riportati i codici di eccezione più utilizzati :

Codice	*Nome*	*Significato*
01	ILLEGAL FUNCTION	Il codice di funzione ricevuto non corrisponde ad una funzione permessa sullo slave indirizzato.
02	ILLEGAL DATA ADDRESS	L'indirizzo cui fa riferimento il campo dati non è un indirizzo permesso sullo slave indirizzato.
03	ILLEGAL DATA VALUE	Il valore da assegnare cui fa riferimento il campo dati non è permesso per questo indirizzo.
07	NAK-NEGATIVEACKNOWLEDGEMENT	La funzione non può essere eseguita nelle attuali condizioni operative o si è tentato di scrivere in un indirizzo a sola lettura.