MODULO
Modbus

Licenza d’uso manuale

Il manuale nasce semplicemente per rispondere alle esigenze di molte persone e in particolar modo quelle che si avvicinano per la prima volta al utilizzo dei BUS o Net.
Con un semplice percorso di avvicinamento il lettore sarà portato da iniziali aspetti storici alla filosofia, da note di colore e esempi pratici e funzionali.
Capirne gli aspetti base e poterne utilizzare tutte le potenzialità.

Ogni contributo allo sviluppo del manuale, suggerimento o segnalazione, sono ben graditi.

Ringraziamenti

Questo documento non sarebbe stato possibile realizzarlo senza la collaborazione di mia moglie Carolina e dei miei 2 figli Selene e Simone, per cui questo documento è dedicato a loro che mi anno sostenuto nella realizzazione.

1-Introduzione a Modbus

1.1- Origini

Il protocollo Modbus è stato concepito alla fine degli anni '70 dalla AEG Modicon, specificatamente per il collegamento dei propri PLC con unità esterne di programmazione e di acquisizione dati.
Per le sue caratteristiche di semplicità e affidabilità, pur nell'ambito di prestaazioni contenute, è stato poi ampiamente adottato nell'automazione industriale.
Esistono due varianti di Modbus: la prima, più propriamente denominata Modbus ASCII/RTU, che risale appunto agli anni 70, e la seconda, più performante, denominata Modbus Plus, che è stata introdotta nel 1980. Questa seconda variante, con caratteristiche più da rete locale, è stata utilizzata però solo in ambito PLC, e solo da Modicon, e non è oggetto della presente trattazione.

1.2-Campi di applicazione

I campi di applicazione possibili sono molto vari. Si va dall'Industria di processo, alla factory automation, al building automation

1.3- Principio di funzionamento

1.3.1- Il metodo Query/Response

La comunicazione prevede una tecnica Master/Slave, in cui un dispositivo Master inizia una transazione, denominata *query*, cui gli Slave rispondono mettendo a disposizione la *response*: si parla allora di metodo di accesso *Query/Response*, o "domanda/risposta".
Il codice funzione presente nella *query* del Master indica allo Slave, avente l'indirizzo presente nello specifico campo del messaggio, il tipo di azione da realizzare.
Gli ulteriori campi "data" del messaggio forniscono allo Slave tutte le informazioni addizionali necessarie alla funzione richiesta.
Schematizzando, se si considera la funzione lettura dati, il codice funzione 03 presente nella "domanda" del Master indica richiesta di lettura dati dai registri buffer dello Slave.
Il campo "data" del messaggio "domanda" da Master, contiene il range dei registri da leggere.
La risposta dello Slave prevede un messaggio in cui il campo funzione è una *semplice eco* del codice ricevuto dal Master, l'indirizzo è il suo proprio, mentre i dati rappresentano le informazioni effettivamente lette.
Nel caso di errore, il codice della risposta è modificato e il campo dati contiene a sua volta un codice che indica il tipo di errore.
Sono previste due modalità di trasmissione seriale: Modo ASCII e Modo RTU.
La desiderata modalità, unitamente ai parametri di comunicazione, quali baud rate, parità, etc., sono selezionati dall'utente all'atto delle configurazione del Master.

È prevista anche una modalità *broadcast*, in cui il Master trasmette un messaggio a tutti gli Slave, i quali eseguono quanto contenuto nella *query*, senza emissione di *response*.

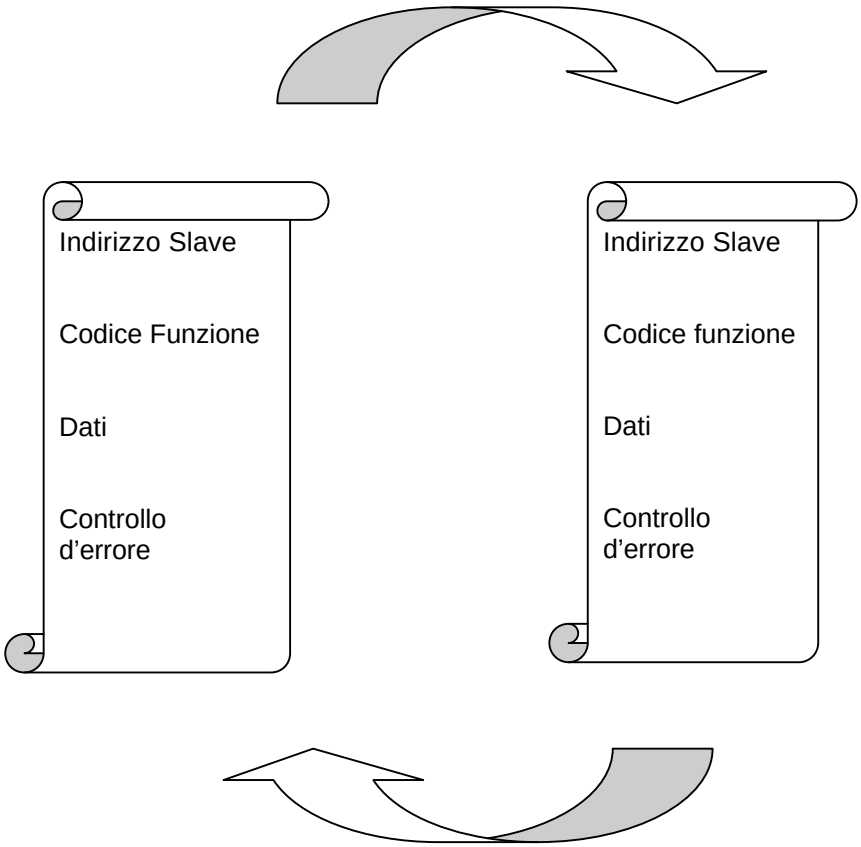

Figura 1: schematizzazione Query/Response

1.3.2- La modalità ASCII

In questa modalità, il messaggio è suddiviso in byte, e ogni byte di messaggio è inviato come carattere ASCII (nel codice ASCII, un carattere è codificato su 7 bit più un bit di parità).
Il vantaggio di questa modalità risiede nel fatto che la trasmissione è rigorosamente asincrona, con la possibilità di intervallare a piacere i singoli byte (l'intervallo di tempo massimo previsto è di 1 secondo).
Ovviamente, parlando di *vantaggio*, ci si riferisce a situazioni applicative in cui le caratteristiche funzionali dei device in rete sono di livello limitato, non in grado di sopportare una trasmissione sincrona.
Il formato dei byte è il seguente:
-1 bit di Start
-7 bit di dati, il bit più significativo inviato per primo
-1 bit di parità (con parità pari/even o dispari/odd a scelta)
-1 bit di Stop, oppure 2 nel caso in cui non si usi la parità.

Il controllo di errore è del tipo LRC, Longitudinal Redundancy Check, una variante più semplice del CRC, effettuato su tutti i caratteri ASCII costituenti il messaggio complessivo (in seguito, un maggior dettaglio su LRC).

1.3.3- La modalità RTU

In questa modalità, ciascun byte contiene due caratteri esadecimali da 4 bit ciascuno, da cui una maggiore densità di caratteri e un migliore *data throughput* a parità di data rate. Il format di ciascun byte di messaggio è:

-1 bit di Start
-8 bit di dati, il bit più significativo inviato per primo
-nessun bit di parità
-1 bit di Stop
Il controllo di errore è del tipo CRC, Cyclic Redundancy Check, che viene effettuato sull'insieme dei byte costituenti il messaggio.
Più precisamente, si tratta dello standard CRC16.
Nel modo RTU il messaggio completo deve essere inviato in modo continuo, quindi non, come nel caso ASCII, un carattere alla volta.
Questo spiega l'assenza del bit di parità (controllo a livello di singolo byte), è l'uso del CRC (controllo sull'intero messaggio).
Usualmente, quando si parla di Modbus, si intende la modalità RTU, nettamente più efficiente della modalità ASCII.

1.4- Architettura: topologia, configurazioni, numero di stazioni

Più che di bus di campo, si parla più propriamente di "Modbus serial connection", prevista per collegamenti Master/Slave punto a punto o multipunto, su lunghezze tipicamente di 350 metri, e data rate da 50 a 38.4 Kbps
In effetti, questi numeri sono da considerarsi solo come riferimento, in quanto alcune caratteristiche del protocollo non sono definite.
In particolare, ci si riferisce a standard di interfaccia, baud rate, numero dei bit di stop.
Sono presenti un’unità Master e più unità Slave, fino a un massimo di 247.
Questo numero è da intendersi come limite logico del protocollo.
È comunque possibile implementare configurazioni diverse.
L'indirizzo 0 è utilizzato per la funzione broadcast.
La topologia più diffusa è a bus, ma sono implementabili anche strutture a stella o ad albero.
Infine, non è definito un mezzo fisico specifico, anche se la maggior parte delle implementazioni prevede il doppino.
Sono utilizzati anche fibra ottica e cavo coassiale.

1.5- I componenti del sistema

Master possono essere un PC, un'interfaccia operatore, un terminale di programmazione, un PLC.
Per unità Slave si intende un qualsiasi device, o nodo della rete, in grado di interpretare il protocollo Modbus.

1.6- Caratteristiche del protocollo

1.6.1- Livelli di protocollo implementati

Il protocollo Modbus definisce il formato e le modalità di comunicazione tra un Master che gestisce il sistema e più Slave che rispondono alle interrogazioni del Master.
Definisce inoltre come Master e Slave stabiliscono e terminano la comunicazione, le modalità di scambio dati e di rilevamento errori.
Modbus implementa tre dei sette livelli OSI:
-Livello 1 (fisico): Interfaccia RS232, per collegamento con singolo RTU (Remote Terminal Unit), oppure interfaccia standard RS485 multidrop per il collegamento con più RTU
-Livello 2 (Data Link): Standard IEC 870-5
-Livello 7 (Applicazione): Standard EN 1434-3

Come precedentemente detto, per i messaggi sono da distinguersi due *Format* differenti: ASCII e RTU.
Per quanto l'interesse sia centrato sul messaggio RTU, si fornisce anche, per completezza di informazione, il formato ASCII.

1.6.2- Format del messaggio ASCII

I messaggi iniziano con il simbolo ASCII "due punti" ("**:**" è codificato come 3A esadecimale nella codifica ASCII) e terminano con CR+LF (vale a dire con la codifica ASCII di Carriage Retour e Line Feed, rispettivamente 0D e 0A).
Possono essere trasmessi solo i caratteri da 0 a 9, e da A ad F. Possono esservi intervalli anche dell'ordine del secondo tra i diversi caratteri costituenti il messaggio. Se il tempo è superiore, lo Slave considera errore di trasmissione. Complessivamente, una "trama" ASCII è cosi' strutturata:

START = 1 carattere (i "due punti") ASCII
ADDRESS = 2 caratteri ASCII
FUNCTION = 2 caratteri ASCII
DATA = 2 caratteri ASCII
LRC CHECK = 2 caratteri ASCII
END = 2 caratteri ASCII, CR+LF

START	ADDRESS	FUNCTION	DATA	LRC Check	END
1 Char	2 Chars	2 Chars	2 Chars	2 Chars	2 Chars CR+LF

Figura 2 :Esempio di trama ASCII

Il campo LRC è sostanzialmente un byte, che contiene un valore binario codificato su 8 bit. Il valore di LRC è calcolato dal Master, che lo pone nel messaggio. Lo Slave calcola a sua volta il valore di LRC all'atto della ricezione del messaggio, e lo confronta con il mvalore presente nel campo LRC. Se i due valori non sono uguali, è segnalato un errore.
LRC è calcolato sommando l'insieme dei byte del messaggio, scartando gli eventali Carry, e facendo poi il complemento a due del risultato.

1.6.3-Format del messaggio RTU

I messaggi iniziano con un intervallo di "silenzio" sulla linea, che deve durare almeno l'equivalente di 3,5 caratteri (equivalenti a 3,5 msec trasmettendo a 9600 bit/sec).
Successivamente viene inviato l'indirizzo dello Slave compreso in un range tra 1 e 247 (l'indirizzo 0 è per la funzione di broadcast e qualifica un messaggio diretto a tutti gli Slave contemporaneamente, senza necessità di risposta vda parteb degli Slave stessi).
Il campo funzione è su 8 bit e il protocollo Modbus prevede un certo numero di codici, ciascuno dedicato ad una funzione. Il campo dati prevede due digit esadecimali, da 00 ad FF, aventi lo scopo di fornire informazioni addizionali sulla funzione da svolgere.
Il messaggio è chiuso da un CRC , calcolato su tutto il messaggio, cui segue un periodo di silenzio della stessa durata dello Start.
Nel caso in cui sia individuato un errore alla ricezione del messaggio, oppure se lo Slave non è in grado di realizzare l'azione voluta, lo Slave stesso genera un messaggio di errore, che viene inviato al Master come ripsosta.

In dettaglio:

START =	periodo di silenzio equivalente a 3,5 caratteri
ADDRESS =	8 bit
FUNCTION =	8 bit
DATA =	Nx8 bit
CRC =	16 bit
END =	periodo di silenzio equivalente a 3,5 caratteri

START	ADDRESS	FUNCTION	DATA	LRC Check	END
3,5 Chara	8 Bits	8 Bits	N x 8bit	16 bits	3,5 Chara

Figura 3: Esempio di trama RTU

1.7-Funzioni Modbus

Funzioni Modbus Implementate = sottoinsieme di funzioni Modbus:

- 01 = read coil status
- 02 = read input status
- 03 = read holding register
- 04 = read input register
- 05 = force single coil
- 06 = preset single register
- 07 = read status
- 15 = force multiple coils
- 16 = preset multiple coils

CRC = CRC16 standard

1.8- Una nota sul protocollo JBUS

Con riferimento ad alcuni prodotti, spesso si parla di protocollo JBUS.
Tale protocollo è funzionalmente identico a Modbus, e l'unica differenza consiste nella numerazione degli indirizzi:

-Modbus = gli indirizzi partono da 0 (0000 = primo indirizzo)
In effetti, gli indirizzi "legali" vanno da 1 a 247, in quanto l'indirizzo 0 è per la funziona Broadcasr

-JBUS = gli indirizzi partono da 1 (0001 = primo indirizzo)

1.9- Enti promotori e sostenitori

Non vi sono enti promotori o sostenitori.
Si fa presente che la società che aveva inizialmente definito il protocoolo, la Modicon, fa attualmente parte del Groupe Schneider

Sito Web www.modbus.org