

# LA CONFIGURAZIONE E PROGRAMMAZIONE DELLA COMUNICAZIONE PROFIBUS-DP IN AMBIENTE SIEMENS STEP 7

Prof.Ing.Salvatore Cavaliere  
Università di Catania, Dipartimento di Ingegneria Elettrica, Elettronica e Informatica (DIEEI)  
Viale A.Doria 6  
95125 Catania (Italy)  
Tel: +39 095 738 2362, Fax: +39 095 738 2397  
E-mail: [salvatore.cavaliere@dieei.unict.it](mailto:salvatore.cavaliere@dieei.unict.it)

# Indice

1	Introduzione .....	3
2	Una breve panoramica sullo Standard Profibus DP .....	4
2.1	Il Protocollo Profibus DP .....	4
2.1.1	Physical Layer .....	4
2.1.2	Data Link Layer .....	4
2.2	I Servizi di Comunicazione del Profibus DP .....	7
2.2.1	Struttura del Telegramma nel Profibus DP Data Link Layer .....	7
2.2.2	Servizi SRD per lo Scambio di Dati .....	8
2.2.3	Servizi di Controllo Globale SDN .....	9
2.3	I File GSD .....	9
3	Configurazione di una Rete Profibus DP: Impostazione dei parametri .....	12
3.1	Definizione dei Parametri Profibus DP .....	12
3.1.1	Syn Time, $T_{SYN}$ .....	13
3.1.2	Station Delay of Responders, $T_{SDR}$ .....	13
3.1.3	Quiet Time, $T_{QUI}$ .....	14
3.1.4	Ready Time, $T_{RDY}$ .....	15
3.1.5	Setup Time, $T_{SET}$ .....	15
3.1.6	Safety Margin, $T_{SM}$ .....	15
3.1.7	Idle Time, $T_{ID}$ .....	15
3.1.8	Tempo di ritardo di trasmissione, $T_{TD}$ .....	17
3.1.9	Slot Time, $T_{SL}$ .....	17
3.1.10	Tempo di aggiornamento del GAP, $T_{GUD}$ .....	19
3.1.11	Target Rotation Time (Token Rotation Time) $T_{TR}$ .....	19
3.2	Impostazione dei Parametri della rete Profibus DP .....	21
3.2.1	Impostazione Standard .....	23
3.2.2	Impostazione Personalizzata .....	23
3.2.3	Calcolo del $T_{TR}$ .....	24
4	Configurazione Hardware del Profibus DP in Simatic Step 7 .....	28
4.1	Creazione di un Nuovo Progetto Step7 .....	28
4.2	Inserimento dell'Oggetto PLC che si sta programmando e dell'Oggetto Profibus .....	28
4.3	Configurazione dell'Oggetto PLC che si sta programmando .....	28
4.4	Configurazione della rete Profibus DP .....	30
4.5	Configurazione dei dispositivi afferenti la rete Profibus DP .....	34
4.5.1	Dispositivo Slave DP Generico .....	35
4.5.2	Dispositivo I/O Siemens .....	38
4.5.3	PLC I-Slave Siemens .....	38
5	Primitive di servizio disponibili in ambiente Simatic per la comunicazione .....	40
5.1	Gestione delle immagini di processo degli ingressi e delle uscite .....	40
5.2	Scambio dati basato su accesso diretto I/O .....	41
5.3	Scambio dati consistente .....	42
5.3.1	SFC14 DPRD_DAT .....	44
5.3.2	SFC15 DPWR_DAT .....	44
5.3.3	Definizione dei moduli di I/O nel Master e nello I-Slave DP .....	45
6	Scrittura di un programma basato sulla comunicazione Profibus DP .....	51
6.1	Scambio dati con accesso diretto I/O .....	51
6.2	Scambio dati consistenti .....	52
6.2.1	Programma utente per il DP Slave S7-300 con CPU315-2DP .....	54
6.2.2	Programma utente per il DP Master S7-300 con CPU416-2DP .....	56
6.3	Scambio dati diretto (DX) .....	59

# 1 Introduzione

Il Profibus DP è un sistema di comunicazione seriale e digitale, ideato per la comunicazione ad alta velocità tra controllori centrali (PLC/PC) e dispositivi remoti (I/O distribuiti, azionamenti, valvole, etc). La maggior parte delle comunicazioni con i dispositivi è ciclica, ossia avviene ad intervalli temporali che si mantengono costanti nel tempo a meno di un ritardo od anticipo non deterministico (noto come jitter); si noti, a tal proposito, che il Profibus DP non garantisce comunicazioni periodiche, ossia che avvengono ad intervalli temporali costanti nel tempo, anche se le oscillazioni di tali intervalli attorno al valore medio è sempre molto piccola (il jitter si mantiene molto vicino allo zero). Il Profibus DP prevede anche comunicazioni non cicliche (ad esempio una tantum) per la configurazione, la diagnostica e la gestione degli allarmi sui dispositivi intelligenti.

Il Profibus DP nasce come Standard Tedesco (norma DIN 19245), diviene successivamente uno standard Europeo (CENELEC EN 50170.2), e poi uno standard Internazionale (IEC 61158). La norma Profibus si compone in realtà di tre standard: FMS (FieldBus Message Specification), DP (Decentralised Peripherals) e PA (Process Automation). La Figura 1 mostra i livelli architetturali dei tre standard. Come si vede il Profibus FMS e DP condividono lo stesso livello fisico (Physical Layer) e logico (Data Link Layer), mentre il PA utilizza un sistema fisico particolare (norma IEC 61158-2), ideato per operare in ambienti esplosivi. L'unico standard che possiede un livello applicativo (Application Layer) è il Profibus FMS.

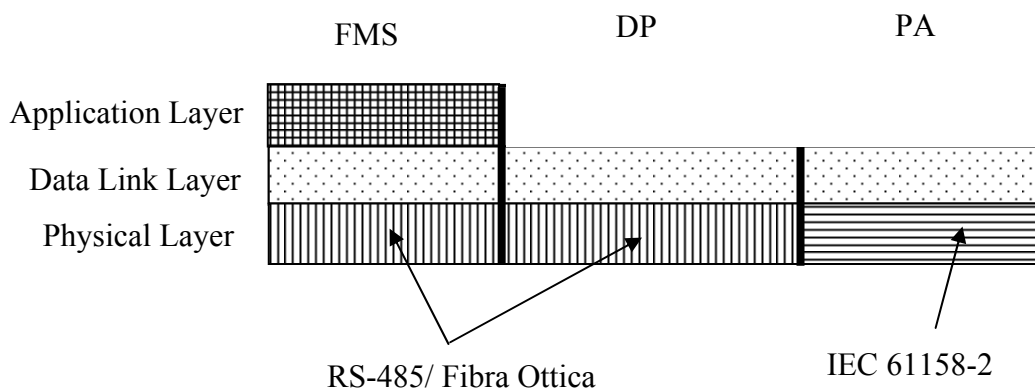


Figura 1 – Architettura degli standard Profibus

Nel seguito la descrizione verrà limitata al Profibus DP, e verranno illustrate le caratteristiche principali relative al protocollo (ovvero ai meccanismi utilizzati per la realizzazione della comunicazione) e ai servizi offerti per la comunicazione.

La conoscenza sia del protocollo sia dei servizi è propedeutica all'apprendimento delle modalità di configurazione e programmazione Profibus DP in ambiente Simatic Step 7.

## 2 Una breve panoramica sullo Standard Profibus DP

Come detto nell'introduzione, la descrizione dello standard Profibus DP verrà fatta illustrando separatamente gli aspetti del protocollo e dei servizi di comunicazione. Alla fine verrà illustrata una caratteristica molto importante dello standard Profibus DP relativa all'esistenza di particolari file (di estensione GSD), definiti per configurare in modo "standard" un qualunque dispositivo di comunicazione compatibile con lo standard.

### 2.1 Il Protocollo Profibus DP

La descrizione del protocollo verrà fatta separatamente per i due livelli che compongono il Profibus DP, ossia il livello fisico e il livello logico.

#### 2.1.1 Physical Layer

Il protocollo Profibus DP, a livello fisico, prevede i seguenti bit/rate: 9.6 Kb/s, 19.2 kb/s, 93.75 kb/s, 187.5 kb/s, 500 kb/s, 1.5 Mb/s, 12Mb/s.

E' possibile connettere fino a 126 nodi di comunicazione. Nel caso in cui la comunicazione è realizzata con RS-485, il sistema è suddiviso in singoli segmenti collegati tra loro da ripetitori; ogni segmento contiene al massimo 32 nodi di comunicazione.

Il Profibus DP utilizza, a livello fisico, una codifica dati NRZ (non return to zero), caratterizzata dal fatto che la codifica dei due valori del bit (1 e 0) e del segnale di inattività avviene tramite i valori di segnale elettrico illustrati dalla seguente tabella.

Simbolo	Codifica del Livello di Tensione/Corrente
1	High
0	Low
Inattività	High

Al fine di introdurre meccanismi che migliorano la qualità della trasmissione, ovvero abbassano la probabilità di errore nella trasmissione, ogni 8 bit di dati ne vengono trasmessi 11, in quanto vengono aggiunti:

- 1 bit di start, per sincronizzare l'inizio della trasmissione del byte di dati
- 1 bit di parità pari, che assume il valore 1 se il numero di bit nel byte dati è dispari, altrimenti assume il valore 0
- 1 bit di stop, per sincronizzare la fine della trasmissione del byte di dati

La Figura 2 mostra gli 11 bit trasmessi per ogni byte dati.

1 Bit di Start	8 bit di dati	1 bit di parità pari	1 Bit di Stop
----------------	---------------	----------------------	---------------

Figura 2 – Bit trasmessi per ogni byte di dati

#### 2.1.2 Data Link Layer

Il protocollo Profibus DP al livello Data Link prevede due tipologie di nodi di comunicazione: Master (classe 1 & 2) e Slave. Lo Slave è un dispositivo che non può prendere l'iniziativa a trasmettere in modo autonomo, ma deve essere sollecitato (su esplicito comando di lettura o di scrittura) da parte di un Master. I Master sono le uniche stazioni che possono prendere l'iniziativa

alla trasmissione. Il Master di Classe 2 è opzionale, e ce ne può essere solo uno; esso serve solo per motivi di supervisione e configurazione. I Master di Classe 1, in numero variabile (ma deve essere presente almeno un Master di Classe 1), sono quelli che scambiano informazioni con gli Slave. Tipicamente ciascun Master di classe 1 controlla (ossia legge e/o scrive) un gruppo di Slave. In accordo al protocollo Profibus DP gli Slave controllati da un Master non possono essere condivisi con altri Slave, anche se, in taluni casi, sono ammesse delle eccezioni.

La contesa sull'accesso al mezzo fisico è gestita unicamente dai Master, tramite un meccanismo di passaggio di token. Il token è un particolare messaggio (opportunamente codificato e riconoscibile da tutti i nodi di comunicazione afferenti al Profibus DP) che passa da un Master ad un altro, e viene tenuto in possesso da ciascun Master per un determinato periodo di tempo; allo scadere di tale tempo ciascun Master rilascia il token in possesso e lo invia ad un altro Master (il suo successore). Tale meccanismo garantisce ovviamente che istante per istante un solo Master trasmette evitando dunque collisioni sull'accesso del mezzo di comunicazione seriale.

Il passaggio del token da un Master all'altro è possibile grazie al fatto che ogni Master conosce l'indirizzo del successivo Master, al quale inviare il token. Periodicamente ogni Master controlla se nuovi master sono stati aggiunti/eliminati nell'intervallo di indirizzi compreso tra il proprio e quello successivo. Tale intervallo è detto GAP.

Durante il possesso del token, ciascun Master può leggere e/o scrivere tutti o un sottoinsieme degli Slave a lui assegnati; il numero di Slave letti/scritti dipende dal tempo di possesso del token da parte del Master, che è regolato dal meccanismo illustrato nel seguito.

In fase di configurazione viene stabilito il Target-Token-Rotation-Time ( $T_{TR}$ ), che determina il più lungo periodo di rotazione del token. Quando un Master rilascia il token fa partire un timer che viene fermato quando il Master riacquista il token. L'intervallo misurato è il Token Rotation Time ( $T_{RR}$ ). Il  $T_{RR}$  deve essere inferiore del Target Token Rotation Time ( $T_{TR}$ ). Quando un Master acquisisce il token, confronta il  $T_{TR}$  con il token rotation time ( $T_{RR}$ ). Vi possono essere due casi:

1.  $T_{RR} < T_{TR}$ . Il Master esegue tutte le trasmissioni previste fino a quando il valore di  $T_{TR}$  è raggiunto o tutte le trasmissioni previste sono concluse.
2.  $T_{RR} \geq T_{TR}$ . Il Master esegue solo una trasmissione e rilascia il token. Ciò viene consentito per permettere al master di trasmettere almeno un messaggio considerato più urgente degli altri.

A livello data link è possibile individuare due configurazioni: mono e multi-Master.

#### 2.1.2.1 Configurazione Mono-Master

Tale configurazione (vedi Figura 3) comprende:

- Un Master di Classe 1
- Un Master di Classe 2 (Opzionale, solo per configurazione, con indirizzo 126)
- Fino ad un Massimo di 125 Slaves (indirizzi da 0 a 125)

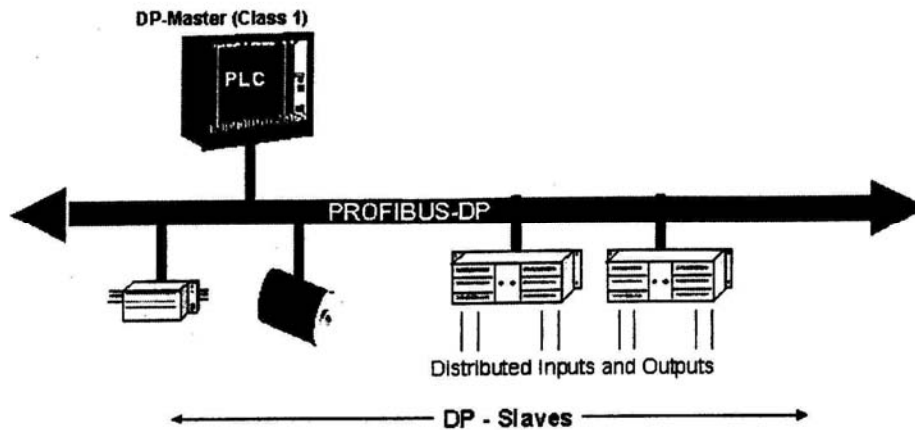


Figura 3 – Configurazione Mono-Master

Il vantaggio principale della configurazione mono-master risiede nella velocità di trasmissione elevata, dovuta alla continua scansione degli Slaves da parte del Master: Giusto per rendere l'idea è possibile, in questa configurazione, trasmettere 1Kbyte di dati in meno di 2ms.

#### 2.1.2.2 Configurazione Multi-Master

Tale configurazione (Figura 4) comprende:

- Uno o più Master
- Un massimo di 124 Slaves
- Al massimo 126 dispositivi sullo stesso bus

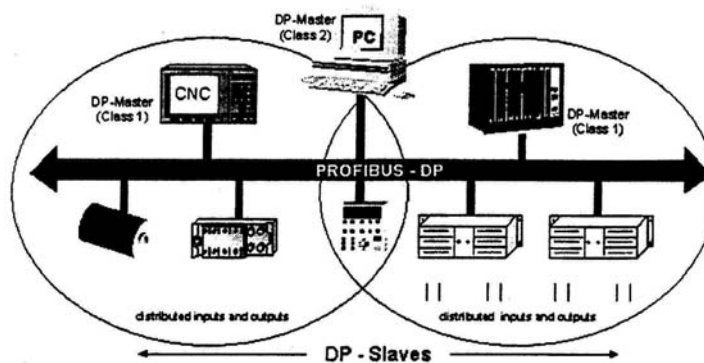


Figura 4 – Configurazione Multi-Master

Il principale vantaggio è la condivisione del mezzo fisico tra più unità di controllo (PLC), ciascuna delle quali coordina il proprio gruppo di Slaves. In particolari condizioni, è poi possibile condividere (in sola lettura) uno o più Slave (solo quelli intelligenti, ossia dotati di CPU, ad esempio PLC; sono generalmente chiamati I-Slave) tra più Master. Il principale limite della configurazione Multi-Master è la riduzione delle prestazioni rispetto la configurazione mono-master. Nella configurazione Multi-Master, il protocollo Profibus DP non prevede la comunicazione tra Master di Classe 1, ma solo tra ciascun Master di Classe 1 con il Master, se presente, di Classe 2.

## 2.2 I Servizi di Comunicazione del Profibus DP

I servizi di comunicazione offerti dal livello data link sono quelli che l'utente può utilizzare, ad esempio, per sviluppare un programma di controllo che necessita dello scambio dati tra un PLC e più dispositivi remoti, tramite la rete di comunicazione Profibus DP.

I servizi offerti dal Profibus DP si basano sulle seguenti caratteristiche:

- La modalità operativa dello scambio informativo è di tipo connectionless mode, ossia lo scambio dati tra master e slave non necessita di una preliminare apertura della comunicazione tramite un flusso extra di scambio dati.
- Oltre allo scambio dati Master-Slave (uno a uno), è possibile avere trasmissioni multi-peer (broadcast, multi-cast), ossia dal Master verso più Slave contemporaneamente.

Per il Profibus DP sono previste 2 tipologie di servizi di comunicazione utilizzabili ai livelli superiori:

- SRD, Invio di Dati e Richiesta di Dati (uno-a-uno)
- SDN, Invio di Dati senza Acknowledge (broadcast, multicast)

Prima di illustrare tali servizi verrà brevemente introdotta la struttura dei messaggi (chiamati anche telegrammi) inviati dal livello data link.

### 2.2.1 Struttura del Telegramma nel Profibus DP Data Link Layer

Ciascun messaggio informativo (ossia dati) inviati da un Master verso uno Slave, o da un Master di Classe 1 verso un Master di Classe 2 (o viceversa), viene confezionato al livello Data Link come sequenza dei campi:

- SD=Delimitatore Iniziale (presente al fine di permettere al dispositivo che riceve il messaggio di poter comprendere la tipologia del messaggio ricevuto, ovvero il tipo di servizio di comunicazione richiesto)
- FC=Codice Funzione (per distinguere se il telegramma si riferisce ad una richiesta, conferma o risposta)
- LE=Lunghezza dati (lunghezza DA+SA+FC+DSAP+SSAP+DU)
- LER=Ripetizione Lunghezza dati, dovuta a motivi di sicurezza, ossia per individuare meglio eventuali errori di trasmissione/ricezione
- DA=Indirizzo di destinazione (ossia del nodo di comunicazione al quale è rivolto il messaggio)
- SA=Indirizzo di sorgente (di chi trasmette)
- DSAP=Destination Service Access Point. Il DSAP è un codice che serve al nodo che riceve per capire quale servizio viene richiesto e che deve essere eseguito
- SSAP=Source Service Access Point. Il SSAP è un codice che serve al nodo che riceve il telegramma per capire quale è il servizio di comunicazione invocato dal nodo sorgente. Tale codice dovrà essere menzionato dal nodo che riceve nel momento in cui voglia rispondere al messaggio appena ricevuto; in tal modo il nodo che riceve può contattare direttamente l'entità responsabile del messaggio appena ricevuto
- DU=Data Unit (dati utente, da 1 a 244 bytes). Sono i dati che vengono inviati.
- FCS=Frame Checking Sequence. E' un campo inserito in base a particolari algoritmi che hanno lo scopo di verificare eventuali errori di trasmissione o di ricezione
- ED=End Delimiter, che individua la fine del messaggio.

## 2.2.2 Servizi SRD per lo Scambio di Dati

La modalità standard per lo scambio dati è basata sul servizio Data Exchange. Esistono poi due servizi (Read Input e Read Output) che non sono previsti per il Master di tipo 1, ma solo per quello di Classe 2.

### 2.2.2.1 Data Exchange

Il Master trasmette i dati di output allo Slave (ossia scrive le sue uscite) e lo Slave, in risposta, invia i propri ingressi. Se lo Slave è semplicemente un dispositivo di uscita, esso risponde con "E5H" che serve come conferma al Master per sapere che lo Slave ha ricevuto le uscite.

La Figura 5 mostra lo scambio informativo tra Master e Slave. Come si vede nei messaggi inviati non sono presenti i campi SSAP e DSAP, perché il servizio Data Exchange, essendo fondamentale, prevede dei valori di default per tali campi e non è dunque necessario inserirli.

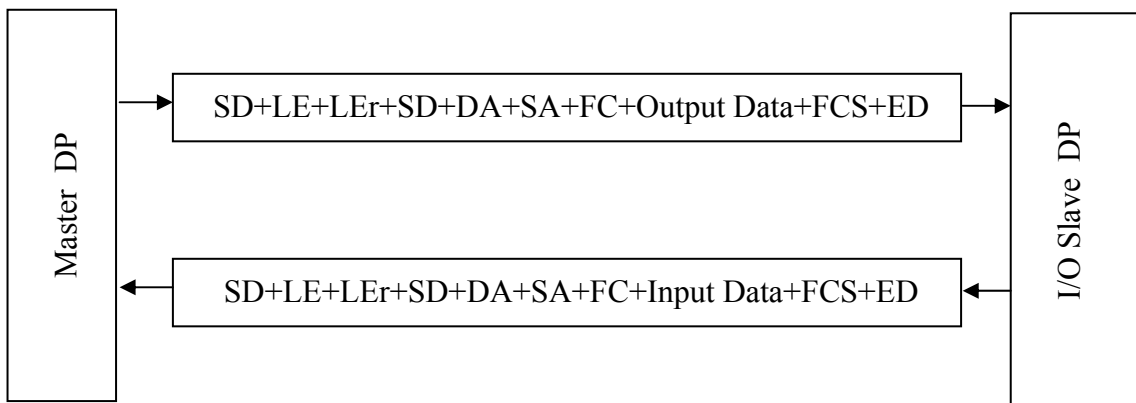


Figura 5 – Scambio Informativo Data Exchange

Le principali caratteristiche di tale scambio informativo sono:

- Il Master non specifica la coppia DSAP e SSAP, perché il servizio Data Exchange è quello di default, come già detto.
- La risposta dello Slave è immediata.
- Il numero massimo di dati che è possibile inviare è di 244 bytes.
- Nel caso di mancata risposta da parte dello Slave, il Master può ripetere la trasmissione più volte, fino ad un massimo numero di volte, definito in fase di configurazione.

### 2.2.2.2 Read Inputs

E' un servizio opzionale per il Master di Classe 2, mentre non è previsto per i Master di Classe 1. Il Master richiede ad uno Slave di Ingresso l'invio dei propri ingressi (vedi Figura 6).

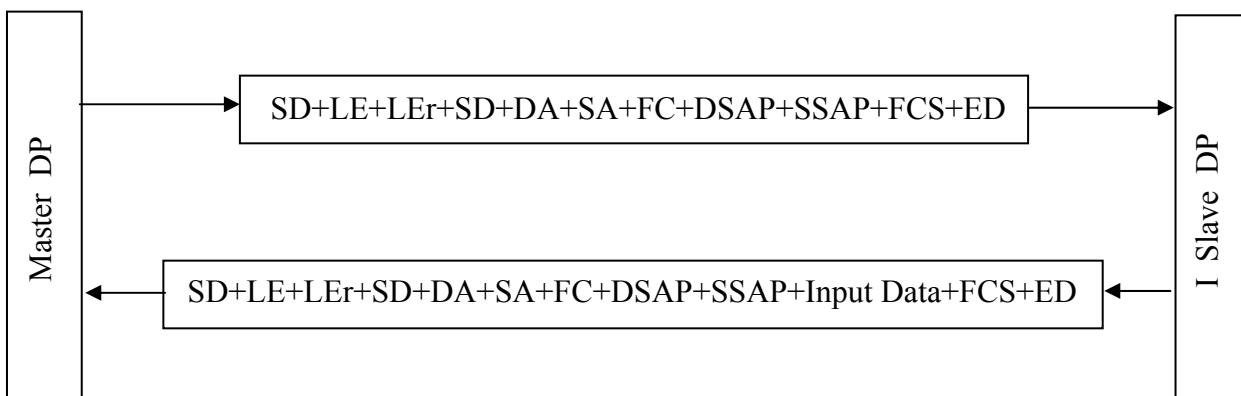


Figura 6 – Scambio Informativo Read Input



In questo scambio informativo il Master specifica come SSAP=56, che è la codifica del servizio di lettura degli ingressi dello Slave.

### 2.2.2.3 Read Outputs

E' un servizio opzionale per il Master di Classe 2, mentre non è previsto per i Master di Classe 1. Il Master richiede ad uno Slave di Uscita l'invia delle proprie uscite.

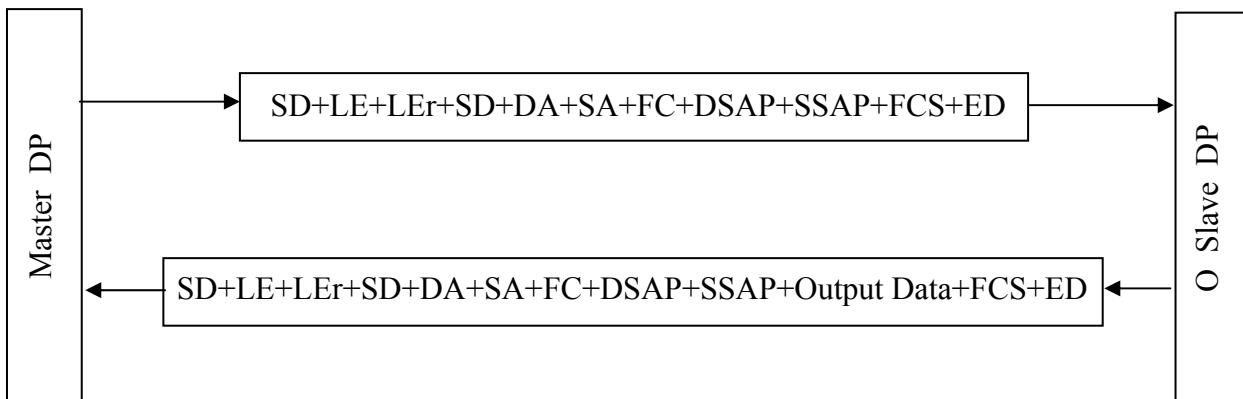


Figura 7 – Scambio Informativo Read Output

Il Master specifica come SSAP=57, che è la codifica del servizio di lettura delle uscite dello Slave.

### 2.2.3 Servizi di Controllo Globale SDN

Vengono utilizzati per inviare messaggi multicast/broadcast, ossia dal Master ad un gruppo di Slave o a tutti. La gestione dell'invio dati broadcast o multicast avviene utilizzando l'indirizzo 127. Nel caso di invio dati multicast viene specificato il gruppo di nodi ai quali si inviano i dati (il gruppo è definito in fase di configurazione).

Esistono due tipologie di servizi di comunicazione SDN:

- Servizi FREEZE/UNFREEZE
  1. FREEZE. Permette al Master di imporre ad uno Slave o ad un gruppo di Slave di congelare i loro ingressi nello stato corrente. Il Master leggerà tali ingressi con il Data Exchange o con Read Inputs.
  2. UNFREEZE. Il sistema ritorna nel modo normale di scambio dati
- Servizi SYNC/UNSYNC
  1. SYNC. Permette al Master di imporre ad uno Slave o ad un gruppo di Slave che le uscite che il Master stesso invierà siano congelate (ossia non divengano effettive). Il Master invierà tali uscite con il Data Exchange.
  2. UNSYNC. Le uscite precedentemente inviate divengono effettive.

## 2.3 I File GSD

I dispositivi Profibus DP hanno ovviamente caratteristiche diverse fra loro in termini di funzionalità (numero di I/O, diagnostica) e di parametri di comunicazione (ad esempio velocità, tempi di risposta). Tali parametri variano per tipo di dispositivo e venditore e sono normalmente specificati nel manuale tecnico.

Per ottenere una configurazione Plug & Play di una rete Profibus le caratteristiche di ogni dispositivo sono memorizzate in un data sheet elettronico detto **Device Data Base**, memorizzato tramite un file di testo (ASCII) di estensione GSD (Generic Station Description).

I file Device Data Base descrivono il dispositivo Profibus DP in modo chiaro, non ambiguo e soprattutto in modo Standard. I file GSD sono forniti dal costruttore del dispositivo.

Il Device data Base è suddiviso in differenti porzioni, di cui almeno tre sono quelle obbligatorie:

- **Informazioni Generali.** Questa sezione contiene il nome del dispositivo e del suo produttore, il numero identificativo del dispositivo, ed altre informazioni di carattere generale.
- **Dati Specifici dello Slave.** Essenzialmente vengono riassunte le caratteristiche dello Slave in termini di velocità supportate e servizi supportati.
- **Specifiche riguardanti i moduli di I/O.** Questa sezione contiene tutte le specifiche relative agli Slaves come il numero ed il tipo di canali di I/O.

Tutte queste sezioni sono visibili in Figura 8, che mostra un file GSD reale (la marca e il modello sono stati ovviamente nascosti).

```

; General device information
GSD_Revision      = 1
Vendor_Name       = "xxxxxx"
Model_Name        = "xxxxxx"
Revision          = "V.x.x"
Ident_Number      = 0x1yyy
Protocol_Ident    = 0
Station_Type      = 0
FMS_supp         = 0
Hardware_Release  = "0"
Software_Release  = "0"
Bitmap_Device     = "xxxxxx"
;Bitmap_Diag
;Bitmap_SF

; Supported baudrates
9.6_supp          = 1
19.2_supp         = 1
93.75_supp        = 1
187.5_supp        = 1
500_supp          = 1
1.5M_supp         = 1
3M_supp           = 1
6M_supp           = 1
12M_supp          = 1

; MaxTcdr default values for supported baudrates
MaxTcdr_9.6       = 60
MaxTcdr_19.2      = 60
MaxTcdr_93.75     = 60
MaxTcdr_187.5     = 60
MaxTcdr_500       = 100
MaxTcdr_1.5M      = 150
MaxTcdr_3M        = 250
MaxTcdr_6M        = 450
MaxTcdr_12M       = 800

; General supported features
Redundancy        = 0
Repeater_Ctrl_Sig = 2
24V_Pins          = 0
Implementation_Type = "LSPM2"

; DP Slave related information
Freeze_Mode_supp = 1
Sync_Mode_supp   = 1
Auto_Baud_supp   = 1
Set_Slave_Add_supp = 0
User_Prm_Data_Len = 05
User_Prm_Data    = 0x00,0x00,0x00,0x00,0x00
Min_Slave_Intervall = 2
Fail_Safe        = 0
Max_Diag_Data_Len = 16
Slave_Family     = 1@SPS@PCD0

```

```
; Modules information
Modular_Station = 0
Max_Module      = 1
Max_Input_Len   = 2
Max_Output_Len  = 2
Max_Data_Len    = 4
Modul_Offset    = 0

Module = "0 Byte out/ 2 Byte in" 0x00,0x11
EndModule
```

Figura 8 – Struttura di un File GSD

### 3 Configurazione di una Rete Profibus DP: Impostazione dei parametri

Lo scopo del capitolo è stato quello di illustrare le relazioni utili a effettuare il settaggio dei parametri Profibus in fase di configurazione. In genere è possibile seguire due criteri per settare tali parametri:

- Standard
- Personalizzato

Il primo si basa sui valori di *default* contenuti nelle tabelle fornite dal *Profibus Specification* (documento contenente le specifiche del Profibus), mentre il secondo tiene conto dei particolari valori forniti dal costruttore di un determinato dispositivo.

Nel capitolo verranno mostrate delle relazioni che, a partire dal valore di alcuni parametri recuperabili dal *data sheet* dei dispositivi Profibus DP e dai relativi file *GSD*, è possibile ottenere dei valori minimi che devono essere rispettati.

Prima di introdurre le relazioni, verranno definiti i parametri che entrano in gioco nella configurazione di una rete Profibus DP.

#### 3.1 Definizione dei Parametri Profibus DP

Molti dei parametri che determinano la configurazione di una rete Profibus DP, sono di tipo temporale. In genere essi sono espressi in bit time ( $t_{bit}$ ). Il bit time è il tempo necessario per la trasmissione di un bit. Esso equivale al reciproco del transmission rate (velocità di trasmissione):

$$t_{bit} = 1 / \text{Transmission Rate (bit/s)}$$

La tabella 1 fornisce i valori di bit time in dipendenza della frequenza di trasmissione.

Tabella 1 – Valori del Bit time

Trasmission Rate bit/s	Bit time
9,6 K	104,2 $\mu$ sec
19,2 K	52,1 $\mu$ s
93,75 K	10,7 $\mu$ s
187,5 K	5,3 $\mu$ s
500 K	2 $\mu$ s
1,5 M	666,7 nsec
3 M	333,3 nsec
6 M	166,7 nsec
12 M	83,3 nsec

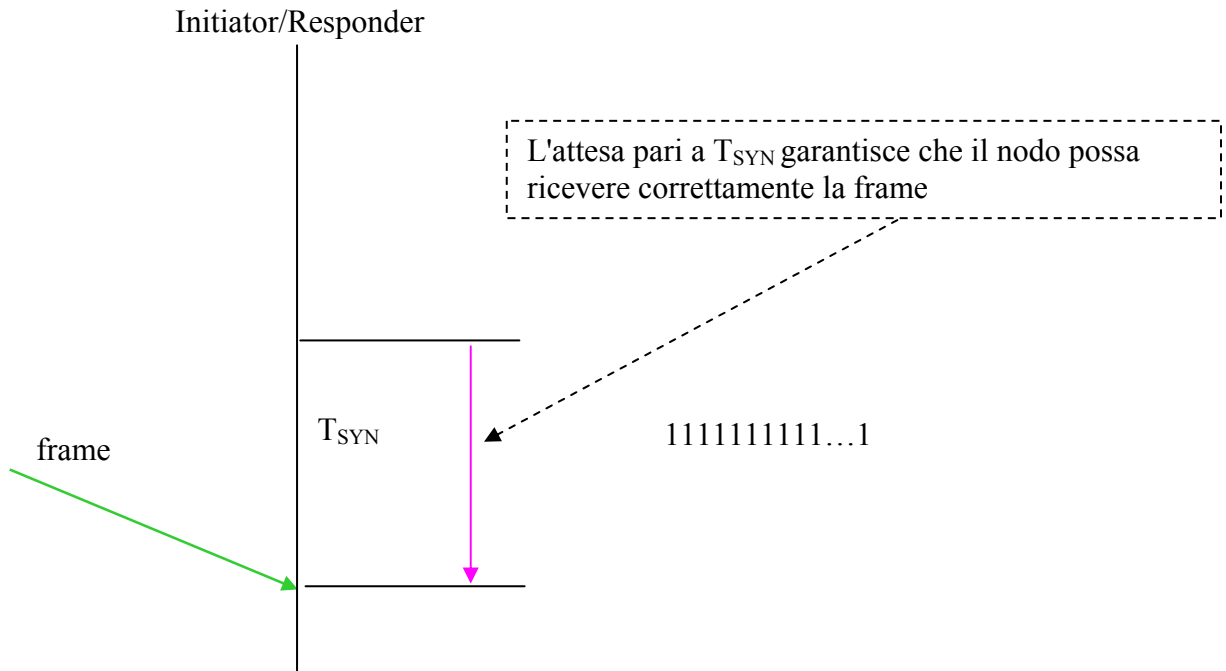
Nel seguito, tutti i parametri del profibus saranno espressi anche in termini di bit time.

I parametri verranno illustrati con riferimento ad un modello di scambio dati tra un generico Initiator e un Responder. In Profibus DP un Initiator è per definizione qualunque stazione che ha l'autorità di intraprendere ("iniziare") un processo di comunicazione; tipicamente è un Master di Classe 1. Un Responder è invece il partner della comunicazione intrapresa dall'Initiator; ad esempio potrebbe essere uno Slave DP.

### 3.1.1 Syn Time, $T_{SYN}$

Il tempo di sincronizzazione  $T_{SYN}$  è il minimo intervallo di tempo durante il quale ogni stazione generica (Initiator o Responder) dovrà ricevere lo stato di Idle (si ricorda che lo stato di Idle, di inattività, corrisponde alla trasmissione del livello di tensione alto, ossia al bit "1") dal mezzo di trasmissione prima che esso possa accettare l'inizio di una action frame (una action frame è un qualunque messaggio di richiesta di servizio, ad esempio un DataExchange) o una token frame.  $T_{SYN}$  è generalmente posto a:

$$T_{SYN} = 33 t_{bit}$$



### 3.1.2 Station Delay of Responders, $T_{SDR}$

Lo station Delay of Responders  $T_{SDR}$  è il periodo di tempo di cui necessita una stazione per rispondere. E' il suo tempo di reazione all'invio di una risposta. Vengono definiti i seguenti due ritardi della stazione :

- Minimum Station Delay of Responders (stazione che invia l'ack o risponde) :  $\min T_{SDR}$
- Maximum Station Delay of Responders :  $\max T_{SDR}$

La Figura 9 mostra il  $T_{SDR}$ ,  $\max T_{SDR}$  e  $\min T_{SDR}$ . Come si vede il Responder (ad esempio uno Slave) può rispondere ad una richiesta dell'Initiator (Master) dopo il  $\min T_{SDR}$  e entro il  $\max T_{SDR}$ . Il  $\min T_{SDR}$  è per default posto a  $11 t_{bit}$ . Il  $\max T_{SDR}$  dipende dal bit rate della rete ed è specificato per ogni dispositivo nel relativo file GSD.

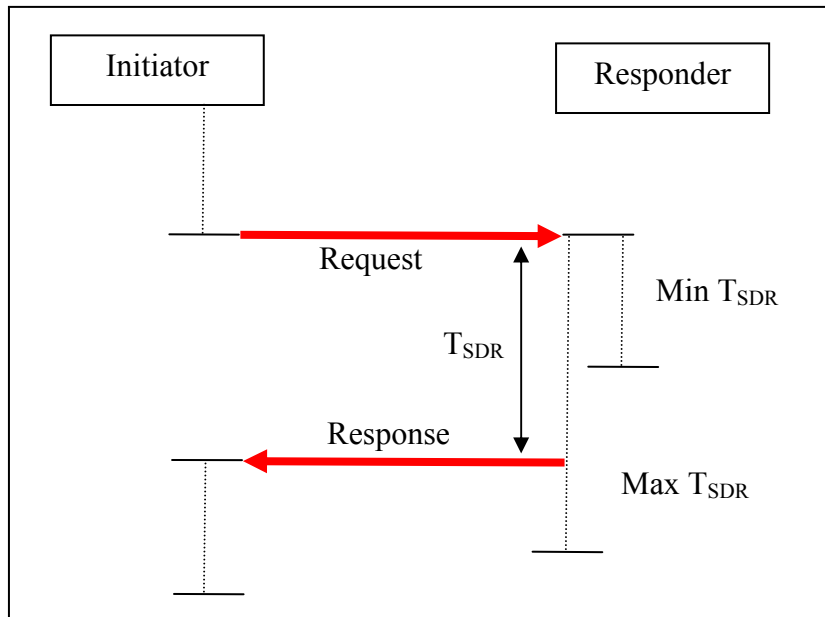


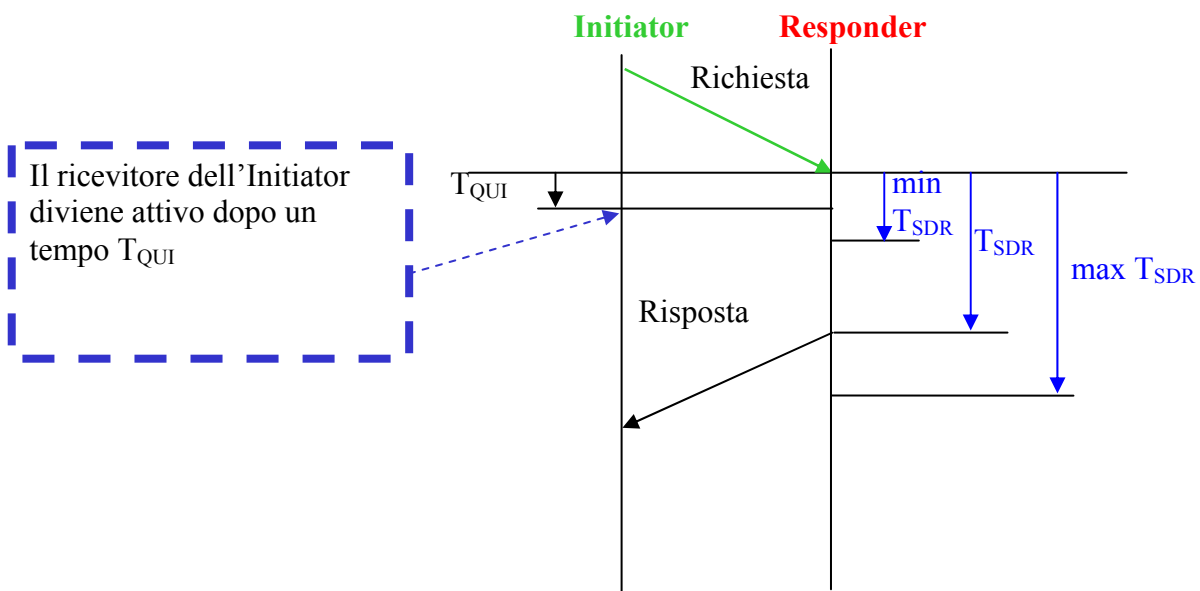
Figura 9 –  $T_{SYN}$ , max e min  $T_{SDR}$

### 3.1.3 Quiet Time, $T_{QUI}$

Il  $T_{QUI}$ , rappresenta il tempo che una stazione che trasmette deve attendere, dopo la fine di una frame, prima di abilitare il suo ricevitore. Sostanzialmente, è il tempo che trascorre affinché una stazione commuti da trasmettitore a ricevitore.

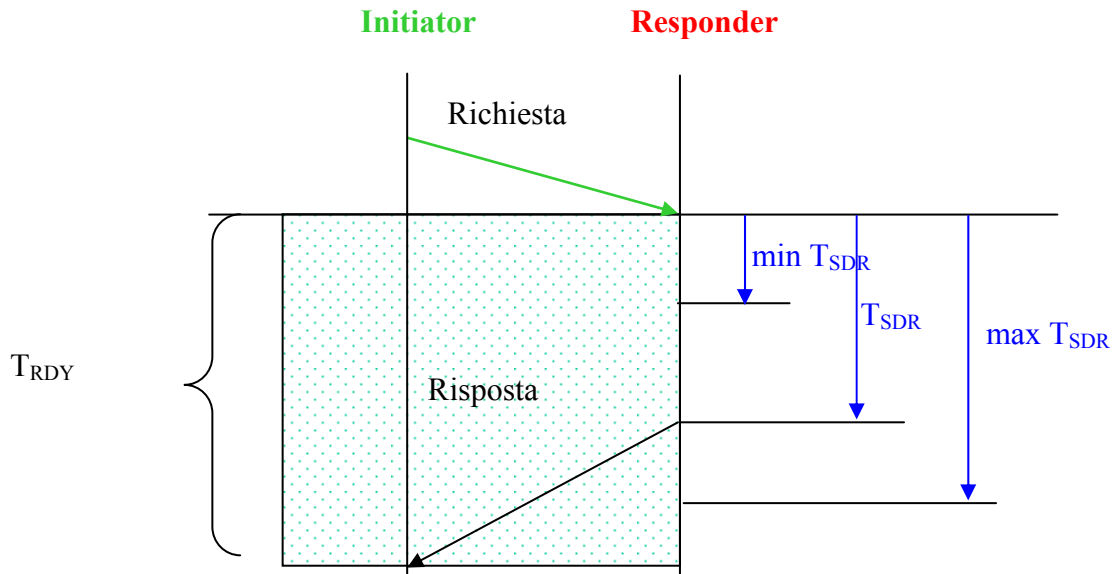
Durante questo **Quiet Time ( $T_{QUI}$ )** la trasmissione e ricezione della frame dovrebbe essere disabilitata. Per il  $T_{QUI}$  vale un'importante relazione:

$$T_{QUI} < \min T_{SDR}$$



### 3.1.4 Ready Time, $T_{RDY}$

Il Ready Time è il tempo all'interno del quale un Initiator (ad esempio un Master) dovrebbe essere pronto a ricevere un ack o una risposta dopo la trasmissione di una richiesta. Il Ready Time  $T_{RDY}$  viene generalmente posto uguale a  $\min T_{SDR}$ ,  $T_{RDY} = \min T_{SDR}$



### 3.1.5 Setup Time, $T_{SET}$

Il  $T_{SET}$  è il tempo che trascorre, in un dispositivo Profibus DP, dall'occorrenza di un evento fino all'attivazione della gestione della relativa reazione:

- Il valore viene espresso in bit time
- Dipende dalla componentistica elettronica (hardware)
- Viene a volte specificato nei manuali
- Nel caso non sia specificato si assume pari al valore massimo (255)

### 3.1.6 Safety Margin, $T_{SM}$

Il Safety Margin,  $T_{SM}$ , viene utilizzato in alcune relazioni come margine di sicurezza. Esso è così definito:

$$T_{SM} = 2 \text{ bit} + 2 * T_{SET} + T_{QUI}$$

### 3.1.7 Idle Time, $T_{ID}$

Vengono definiti due parametri identificati come Idle Time:

- $T_{ID1}$ . Esso è definito come il tempo che deve attendere l'initiator (ossia chi attiva una trasmissione) dopo la ricezione dell'ultimo bit di una frame di ack (conferma), di una frame di una risposta o di una token frame, prima di poter trasmettere una nuova action frame o una nuova token frame. La Figura 10 illustra quanto detto. Il  $T_{ID1}$  è definito come:

$$T_{ID1} = \max (T_{SYN} + T_{SM}, \min T_{SDR})$$

A causa dei valori molto piccoli generalmente assunti da  $\min T_{SDR}$  nella pratica, per il  $T_{ID1}$ , possiamo considerarlo pari alla somma tra  $T_{SYN}$  e  $T_{SM}$ , ossia  $T_{ID1} = T_{SYN} + T_{SM}$ .

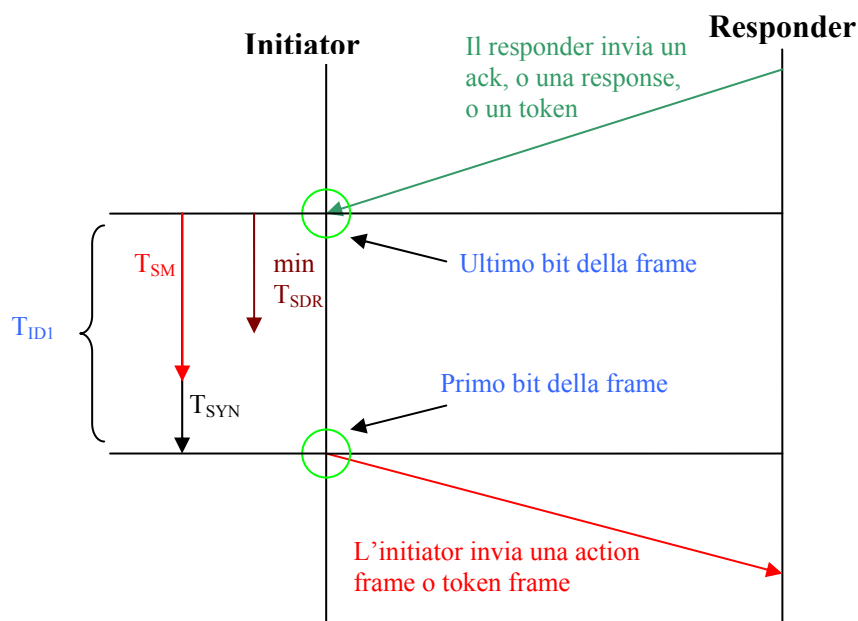


Figura 10 –  $T_{ID1}$

- $T_{ID2}$ . Esso è definito come il tempo che deve attendere l'initiator dopo l'invio di una action frame che **non deve essere confermata da un ack** (sono quelle ad esempio relative ai servizi SDN inviati a molti o tutti, per i quali non è previsto una conferma), prima di poter trasmettere una nuova action frame o una nuova token frame. La Figura 11 illustra quanto detto. Il  $T_{ID2}$  è definito come:

$$T_{ID2} = \max (T_{SYN} + T_{SM}, \max T_{SDR})$$

Essendo il  $\max T_{SDR}$  maggiore di  $T_{SYN} + T_{SM}$ , si può generalmente assumere che  $T_{ID2} = \max T_{SDR}$ .



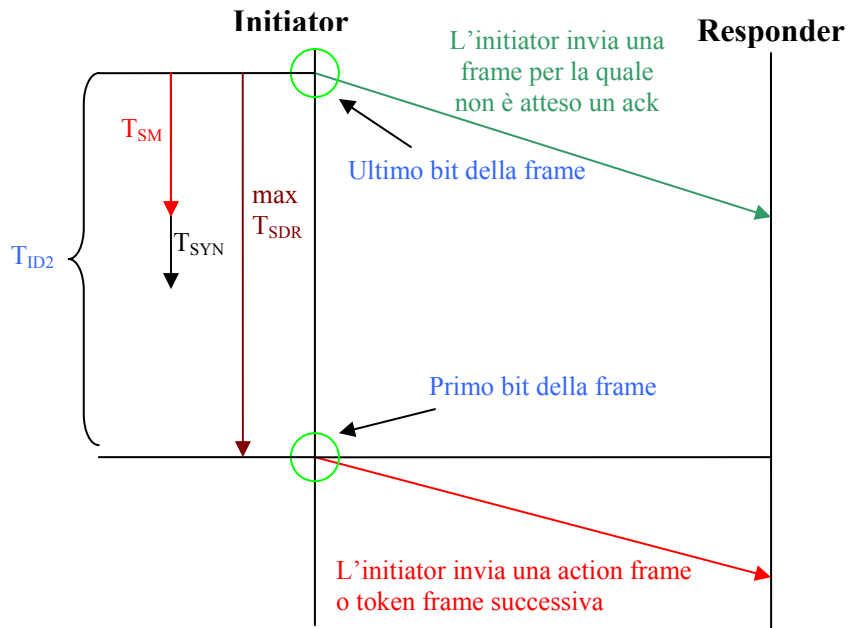


Figura 11 –  $T_{ID2}$

### 3.1.8 Tempo di ritardo di trasmissione, $T_{TD}$

Il Tempo di ritardo di trasmissione  $T_{TD}$  è il tempo massimo che trascorre sul mezzo di trasmissione tra il trasmettitore e il ricevitore quando una frame (cioè un messaggio) viene trasmessa. Il tempo di ritardo dei ripetitori dovrebbe essere considerato, se necessario.

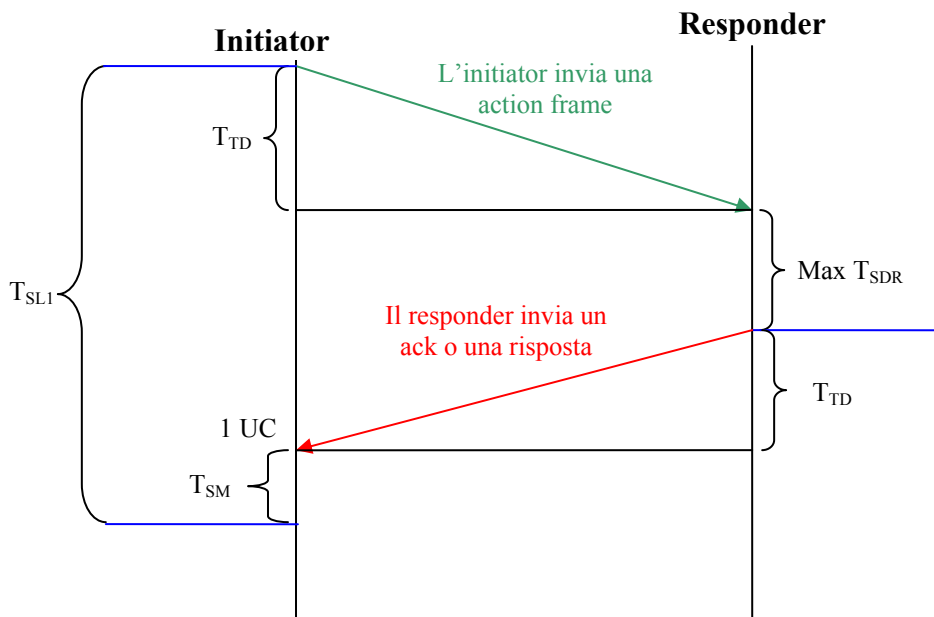
### 3.1.9 Slot Time, $T_{SL}$

Vengono considerati due Slot Time:

- $T_{SL1}$ . Il primo è definito come il tempo massimo che l'initiator aspetta per la ricezione completa del primo carattere della frame di ack o di risposta, dopo la trasmissione dell'ultimo bit di una action frame. Questo parametro, è dato dalla relazione seguente:

$$T_{SL1} = 2 * T_{TD} + \max T_{SDR} + 11 \text{ bit} + T_{SM}$$

che viene illustrata dalla Figura 12.



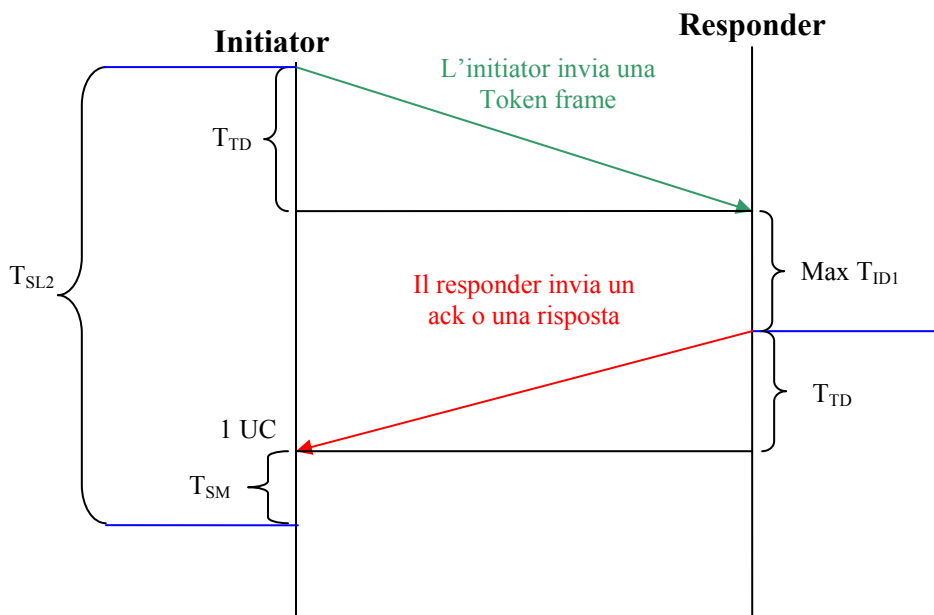
**N.B.** 1 UC indica che l'Initiator ha ricevuto il primo carattere della frame di ack o di risposta

Figura 12 –  $T_{SL1}$

- $T_{SL2}$ . Questo Slot Time è definito come il tempo massimo che l'initiator attende per la ricezione del primo carattere dell'ack o risposta dopo la trasmissione del token frame. Questo parametro è calcolato nel modo seguente:

$$T_{SL2} = 2 * T_{TD} + \max T_{ID1} + 11 \text{ bit} + T_{SM}$$

come si comprende esaminando la Figura 13.



**N.B.** 1 UC indica che l'Initiator ha ricevuto il primo carattere della frame di ack

Figura 13 –  $T_{SL2}$

Per semplificare la relazione, nel sistema viene utilizzato solamente lo Slot Time più lungo, ossia:

$$T_{SL} = \max (T_{SL1} , T_{SL2})$$

### 3.1.10 Tempo di aggiornamento del GAP, $T_{GUD}$

Il tempo di aggiornamento del GAP serve alle stazioni Master per inizializzare la manutenzione del GAP. Dopo la prima generazione della GAPL (GAP List), l'aggiornamento dell'immagine del GAP viene inizializzata ciclicamente dopo ogni intervallo  $T_{GUD}$ . Questa inizializzazione ha luogo:

- alla ricezione del successivo Token possibile, nel caso in cui sia ancora disponibile il Token Holding Time, dopo un ciclo di messaggio regolare
- durante la fase di trattenimento del Token successivo.

Il GAP Update Time è un multiplo del Target Rotation Time ed è definito come segue:

$$T_{GUD} = G * T_{TR} \quad 1 \leq G \leq 100$$

### 3.1.11 Target Rotation Time (Token Rotation Time) $T_{TR}$

Il Token Rotation Time è un parametro assai importante in quanto determina le prestazioni del sistema in termini di capacità di smaltire efficientemente il flusso informativo scambiato tra i Master e i loro Slave. Un valore troppo basso non permette l'interrogazione completa da parte di ciascun Master verso i propri Slave, portando a perdite di informazioni; di contro, un valore troppo alto crea ritardo sul tempo di interarrivo del token ad ogni Master, che introduce ritardi sulla lettura/scrittura dai/verso i propri Slave.

Il minimo *Token Rotation Time* di un sistema, dipende da:

1. numero di stazioni master
2. dai cicli di token. Un ciclo di Token, indicato con il parametro  $T_{TC}$ , è il carico del bus dovuto alla token frame inviata da un Master al suo successore Master. Il carico base totale per tutti i Master è dato da *na* cicli di token, dove *na* indica il numero di stazioni master. Con riferimento alla Figura 14, il parametro  $T_{TC}$  è dato da:

$$T_{TC} = T_{TF} + T_{TD} + T_{ID1}$$

dove:

- Token Frame Time  $T_{TF}$ , che è determinato dal numero di bit della token frame. Un carattere della frame è sempre costituito da 11 bit e perciò la Token Frame, essendo costituita da tre caratteri, ha una lunghezza di 33 bit.
  - Trasmission Delay Time  $T_{TD}$ , che dipende dalla lunghezza della rete (circa 5 ns/m senza ripetitori) ed è sostanzialmente molto più piccolo di tutti gli altri tempi e pertanto ai fini del calcolo può essere trascurato.
  - Idle Time  $T_{ID1}$
3. dalla durata dei cicli di messaggio scambiati tra tutti i Master e i relativi Slave. Un ciclo di messaggio è costituito da una action frame e dalla frame di risposta. Il tempo di ciclo del messaggio è costituito dai tempi di trasmissione della frame, dai tempi di ritardo della trasmissione e dai tempi di ritardo della stazione.

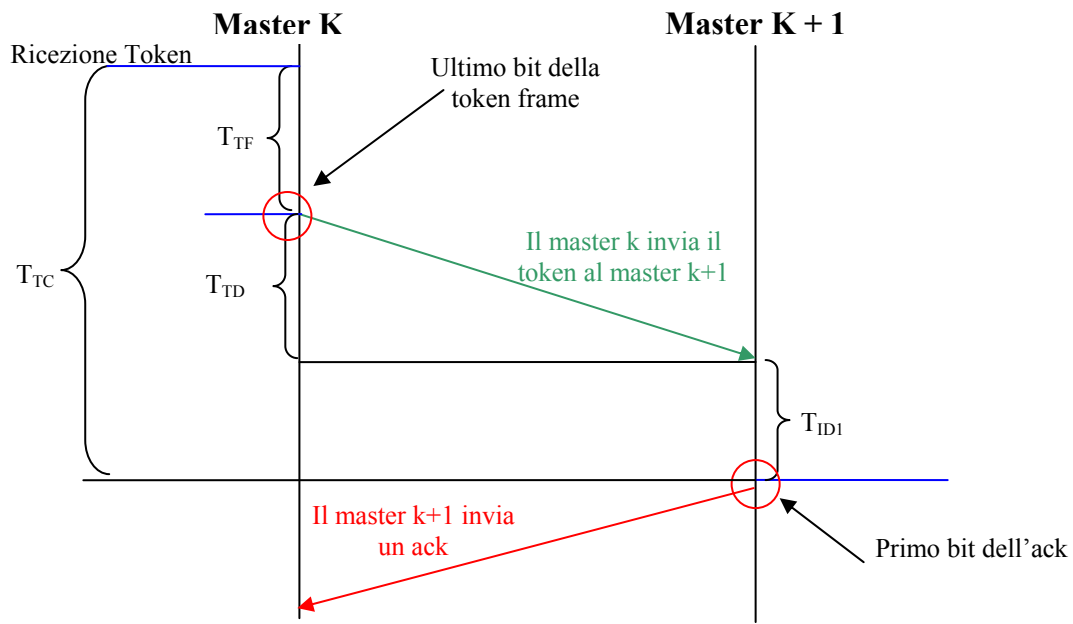


Figura 14-  $T_{TC}$

## **3.2 Impostazione dei Parametri della rete Profibus DP**

L'impostazione dei valori da assegnare ai parametri che regolano il funzionamento di una rete Profibus DP può essere fatta in due modalità. La prima (standard) prevede l'adozione di valori di default, che per molti scenari vanno bene. Nel caso in cui non soddisfino le aspettative in termini di prestazioni del sistema di comunicazione, è necessario adottare valori calcolati in base a formule più complesse.

In ogni caso, la Figura 15 può aiutare ricordare i principali parametri già illustrati che dovranno essere configurati.

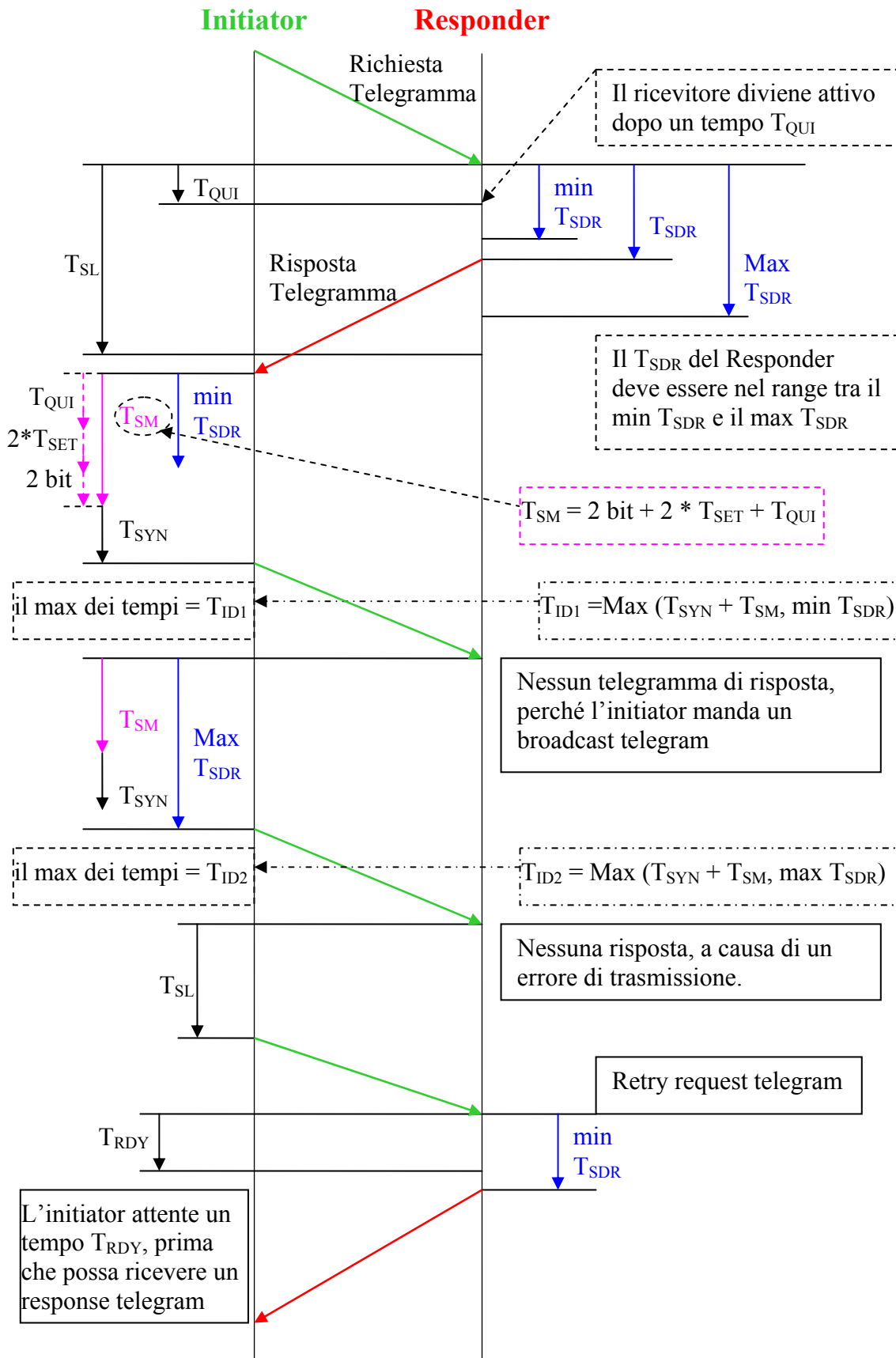


Figura 15

### 3.2.1 Impostazione Standard

In fase di configurazione, secondo questo criterio, i valori dei parametri da considerare sono dati dalle seguenti tabelle.

<b>Parameter</b>	<b>Baund rate</b>	12,0 MB	6,0 MB	3,0 MB	1,5 MB	500 Kb	187,5 Kb	93,75 Kb	19,2 Kb	9,6 Kb
<b>min T<sub>SDR</sub></b>		11	11	11	11	11	11	11	11	11
<b>max T<sub>SDR</sub></b>		800	450	250	150	100	60	60	60	60
<b>Slot Time T<sub>SL</sub></b>		1000	600	400	300	200	100	100	100	100
<b>Quiet Time T<sub>QUI</sub></b>		9	6	3	0	0	0	0	0	0
<b>Setup Time T<sub>SET</sub></b>		16	8	4	1	1	1	1	1	1
<b>Target Rotation Time T<sub>TR</sub></b>		8192	8192	8192	6656	6656	6656	6656	6656	6656
<b>Gap Factor G</b>		10	10	10	10	10	10	10	10	10

Velocità Kb/s	Tslot	Tset
9,6	100	10
19,2	120	15
45,75	640	95
93,75	124	45
187,5	400	80
500	250	80
1500	300	240
3000	400	4
6000	600	8
12000	1000	16

Inoltre è necessario considerare le seguenti relazioni.

$$T_{ID1} \geq T_{Syn} + T_{SM} \quad \text{dove:} \quad T_{SM} = 2bit + 2 * T_{SET} + T_{QUI}$$

$$T_{ID2} = \max (T_{Syn} + T_{SM}, \text{Max } T_{SDR})$$

$$T_{RDY} \leq \text{Min } T_{SDR}$$

**N.B.** In fase di configurazione, per la scelta dei valori dei parametri nelle tabelle precedenti, come transmission rate deve essere considerato quello del dispositivo più lento.

### 3.2.2 Impostazione Personalizzata

In questo caso, in fase di configurazione, possiamo considerare due casi:

- Presenza, nel data sheet di ciascun dispositivo che compone la rete Profibus DP da configurare, del parametro T<sub>SET</sub>

- Assenza di informazioni che permettono di individuare il valore esatto del parametro  $T_{SET}$  per ciascun dispositivo. In tal caso, si considera per il  $T_{SET}$  il valore massimo (pari a 255 bit). L'inconveniente nell'utilizzare questo valore per il  $T_{SET}$  è che non si ottiene una rete ottimizzata. Una possibile soluzione a questo problema, ai fini di ottenere una rete ottimizzata, è quella di effettuare dei test di laboratorio, partendo dal caso peggiore (valore massimo del  $T_{SET}$ ) e decrementando progressivamente questo valore fino a quando si osserva un corretto funzionamento della rete.

In entrambi i due casi appena illustrati, è possibile utilizzare le seguenti relazioni per determinare i rimanenti parametri della rete:

$$2 * T_{TD} + \max T_{SDR} + 11 \text{bit} + T_{SM} \leq T_{SL} \leq (2^{16} - 1) \text{ bit}$$

$$T_{SYN} + T_{SM} \leq \max T_{SDR} \leq (2^{16} - 1) \text{ bit}$$

**min  $T_{SDR}$**  = Dovendo prendere il più piccolo fra tutti i dispositivi possiamo già impostarlo pari a 11 bit

$$1 \text{ bit} \leq T_{SET} \leq (2^8 - 1) = (255) \text{ bit}$$

**$T_{QUI}$**  = Essendo trascurabile rispetto agli altri istanti di tempo, viene impostato a zero  $T_{BIT}$

$$1 \leq G \leq 100$$

$$T_{ID1} \geq T_{Syn} + T_{SM}$$

$$T_{ID2} = \max (T_{Syn} + T_{SM}, \text{Max } T_{SDR})$$

$$T_{RDY} \leq \text{Min } T_{SDR}$$

### 3.2.3 Calcolo del $T_{TR}$

In entrambi i criteri di impostazione (Standard o Personalizzata) dei parametri, il  $T_{TR}$  (per ogni singola stazione master) viene calcolato con la relazione seguente:

$$T_{TR}^{(DP)} = T_{DE} + T_{TR}^{(FMS)} + T_{ADD}$$

dove :

- $T_{ADD}$ 
  - Tiene conto del fatto che per ogni token, il Master dovrà inviare servizi di varia natura oltre che il polling ciclico degli Slave.  $T_{ADD}$  considera il contributo di uno solo di tali servizi. Facendo un calcolo per eccesso, si suppone che il  $T_{ADD}$  si riferisce ad un servizio di tipo confermato generico, come mostrato in figura 16.1. Ovviamente in questo modo si considera il caso peggiore in quanto un servizio non confermato richiede meno tempo rispetto ad un generico servizio confermato. Facendo riferimento alla Figura 16.1, il calcolo è il seguente:
  - $T_{ADD} = (T_{ID1} + T_{SL} + 2 * \text{Max\_Data\_Lenght} * 11) * (\text{Retry Counter} + 1)$
  - $T_{ID1}$  = Idle Time del Master
  - $T_{SL}$  = Slot Time
  - Ovviamente, in base al programma eseguito dai/dal Master, il calcolo del parametro  $T_{ADD}$  dovrà tenere conto quanti di questi servizi confermati vengono



inviati; nel caso vengano inviati anche servizi non confermati, allora essi possono essere inglobati nel calcolo precedente, che diventa per eccesso.

- $T_{DE} = [T_{SYN} + T_{IDI} + T_{SDR} + 2*(Header + Trailer)] * n^{\circ} \text{ slave} + (n^{\circ} \text{ Byte Input} * 11) + (n^{\circ} \text{ Byte Output} * 11)$ 
  - Tiene conto del tempo complessivo per il polling ciclico degli Slave. Si veda Figura 16.b
  - $T_{SYN}$  = Tempo di sincronizzazione
  - $T_{SDR}$  = Ritardo alla risposta di uno slave
  - Header + Trailer =  $11*18$  byte (9 per il DataExchange in richiesta, 9 per la risposta) = 198 bit
  - $n^{\circ} \text{ byte In}$  = Numero totale di byte di ingresso (tutti gli slave relativi al Master)
  - $n^{\circ} \text{ byte Out}$  = Numero totale di byte di uscita (tutti gli slave relativi al Master)
  - $n^{\circ} \text{ Slave}$  = Numero di slave relativi al Master
  
- $T_{TR}^{(FMS)}$  è legato al traffico relativo al Protocollo FMS, nel caso si supponga la presenza di dispositivi FMS nella rete. Se la rete è omogenea composta solo da dispositivi Profibus DP, allora  $T_{TR}^{(FMS)} = 0$

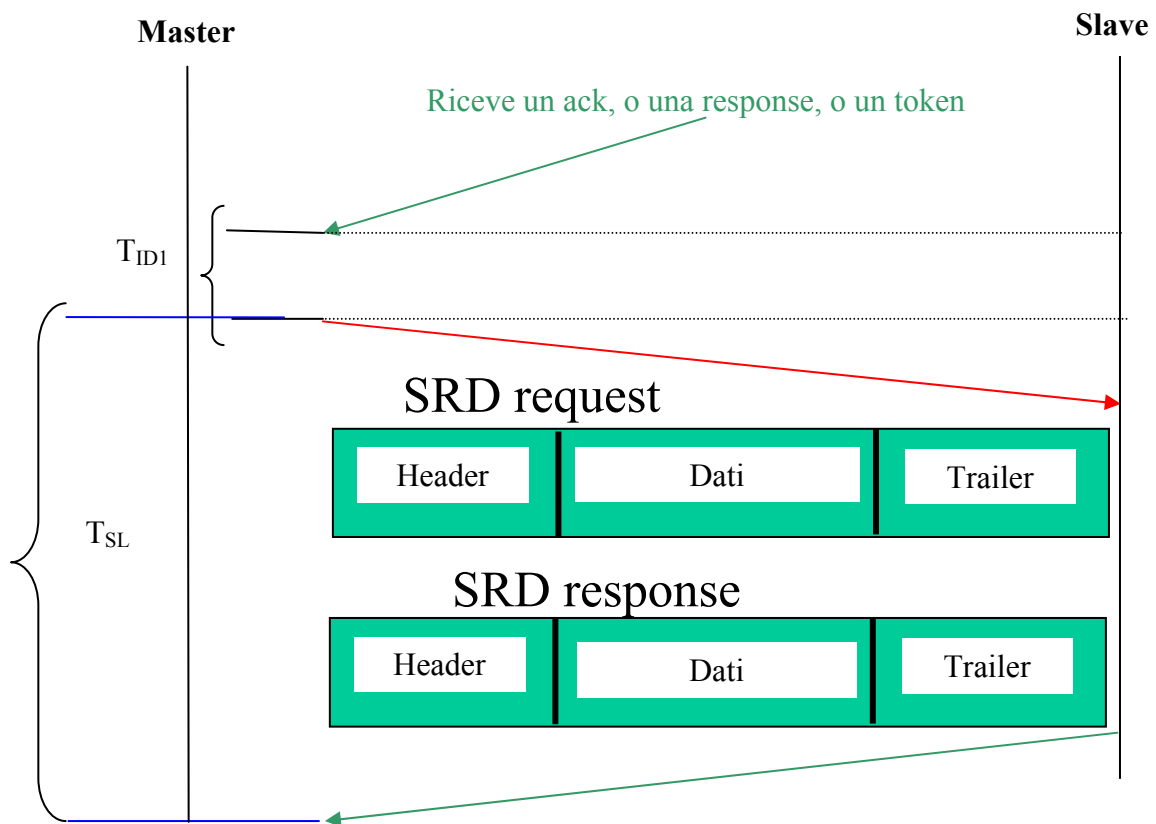


Figura 16.a –  $T_{ADD}$

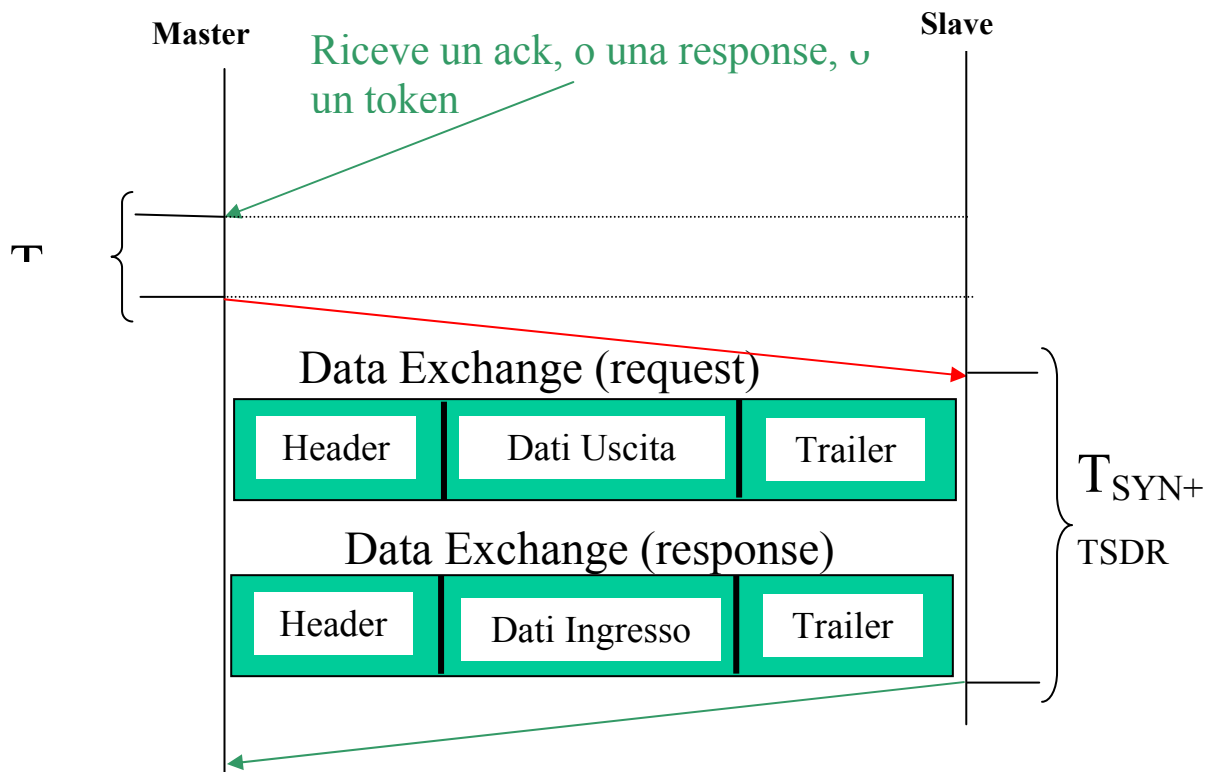


Figura 16.b –  $T_{DE}$

Nel caso di sistema **multimaster**, il  $T_{TR}^{(tot)}$  è dato dalla somma dei  $T_{TR}$  relativi ad ogni master.

$$T_{TR}^{(TOT)} = \sum_{i=1}^{N_a} T_{TR}(i) \quad \text{dove } N_a = n^\circ \text{ Staz. Master}$$

## 4 Configurazione Hardware del Profibus DP in Simatic Step 7

Questo capitolo illustrerà le procedure per la configurazione di una rete Profibus DP nell'ambiente di programmazione Simatic Step 7, fornendone esempi. I paragrafi seguenti illustreranno passo-passo le operazioni da compiere.

### 4.1 Creazione di un Nuovo Progetto Step7

Aprire Simatic Manager e selezionare **File**→**Nuovo** (oppure CTRL+N) per creare un nuovo progetto. Selezionare (nel campo **Tipo**) l'opzione progetto, specificare la directory (nel campo **Destinazione**) in cui si intende salvare e il nome del progetto (nel campo **Nome**). Si inserisca ad esempio il nome *DuePLC*. Confermare con OK. Si è tornati così all'ambiente Simatic Manager e vedrete che è stato creato automaticamente anche l'oggetto MPI (Multi Point Interface) che gestisce la comunicazione tra PC e PLC.

### 4.2 Inserimento dell'Oggetto PLC che si sta programmando e dell'Oggetto Profibus

Nella parte sinistra del Simatic Manager, selezionare il nome del progetto appena creato e fare click con il pulsante destro del mouse; compare un menu dal quale dovrete scegliere l'opzione **Inserisci**→**Nuovo Oggetto**. Comparirà una lista di oggetti da inserire. Bisognerà scegliere quello che corrisponde al dispositivo che si sta programmando. Ad esempio se si intende programmare un PLC SIMATIC serie 400, dovrà essere scelto l'oggetto SIMATIC 400. Si inserisca, ad esempio, tale oggetto.

Nello stesso modo dovrà essere aggiunto un altro oggetto chiamato PROFIBUS.

Se si desidera si può modificare i nomi dei due oggetti appena creati (i due oggetti sono visualizzati nella finestra a destra del Simatic Manager); come esempio si inseriscano i nomi *Simatic 400* e *Profibus*.

### 4.3 Configurazione dell'Oggetto PLC che si sta programmando

Adesso, sempre nell'ambito del tool **Simatic Manager**, selezionare l'oggetto *Simatic 400* appena creato e fare doppio click. Verrà visualizzato nella finestra a destra, l'icona Hardware; selezionare tale oggetto e fare click due volte. Si aprirà il tool **Configurazione HW**, che permetterà di configurare le caratteristiche del PLC che si sta programmando.

Al fine della configurazione, è necessario inserire almeno un rack, una tipologia di alimentatore e una tipologia di CPU.

Per scegliere il rack, navigare sull'albero che compare sulla vostra destra e selezionare la directory relativa alla tipologia di SIMATIC che si sta programmando (nel nostro esempio SIMATIC 400). Vedrete una sotto-directory chiamata RACK 400 (nel caso si fosse scelta la tipologia SIMATIC 300, avreste visto la directory RACK -300). All'interno della directory RACK 400 esistono differenti tipologie di rack. Ad esempio scegliete la tipologia UR 2, ossia un rack universale a 9 slots. Trascinare il rack scelto a sinistra nella finestra in basso. Vedrete delle righe che corrispondono a dei connettori, nei quali installare l'hardware che si intende utilizzare (per lo meno bisogna inserire un alimentatore e una CPU).

Nel connettore 1 deve essere trascinato l'oggetto che descrive l'alimentatore. Tale oggetto si trascina sempre dalla directory che vedete sulla vostra destra; sia nella directory SIMATIC 300 sia

in quella SIMATIC 400 esiste una directory chiamata PS-300 e PS-400 rispettivamente. Ciascuna delle due directory contiene l'elenco degli alimentatori disponibili per le due CPU. Ad esempio scegliamo l'alimentatore PS407 10A. A partire dal connettore di posizione 2 (o 3 in dipendenza dello spazio occupato dall'alimentatore) dovrà essere posizionata l'esatta CPU che si sta programmando. Bisogna fare attenzione che la tipologia di CPU deve essere esattamente quella che si sta programmando, inclusa la versione di firmware, altrimenti vi saranno problemi in fase di caricamento del programma nel PLC. L'oggetto corrispondente esattamente alla vostra CPU dovrà essere scelto a destra e trascinato su un qualunque connettore (ad esempio nel connettore 3, visto che il 2 è occupato dall'alimentatore). Come esempio si scelga la CPU 416-2DP (dotata di interfaccia integrata Profibus DP). All'interno della directory corrispondente, troverete diversi oggetti che corrispondono a differenti versioni di firmware della stessa CPU; le differenze tra una versione e l'altro corrispondono a piccole differenze della configurazione hardware della CPU 416-2DP. Scegliete come esempio la *CPU 416-2DP* firmware *6ES7 416-2XK00-0AB0* e inseritela nel connettore 3 (sempre trascinando l'oggetto sul connettore).

Appena inserirete l'oggetto CPU, il tool Configurazione HW vi chiederà se la CPU dotata di interfaccia interna Profibus DP, dovrà essere connessa alla rete Profibus precedentemente creata (oggetto *Profibus* inserito al passo precedente). Inoltre nella stessa finestra è possibile impostare un indirizzo per il PLC appena inserito. Limitarsi ad impostare l'indirizzo (si scelga 2 come indirizzo di esempio) e scegliere di connettere l'interfaccia Profibus DP interna alla rete Profibus appena creata. Vedrete una immagine simile alla Figura 17.

A questo punto terminare il programma Configurazione HW scegliendo di salvare le modifiche appena apportate. Sarete tornati al tool Simatic Manager.

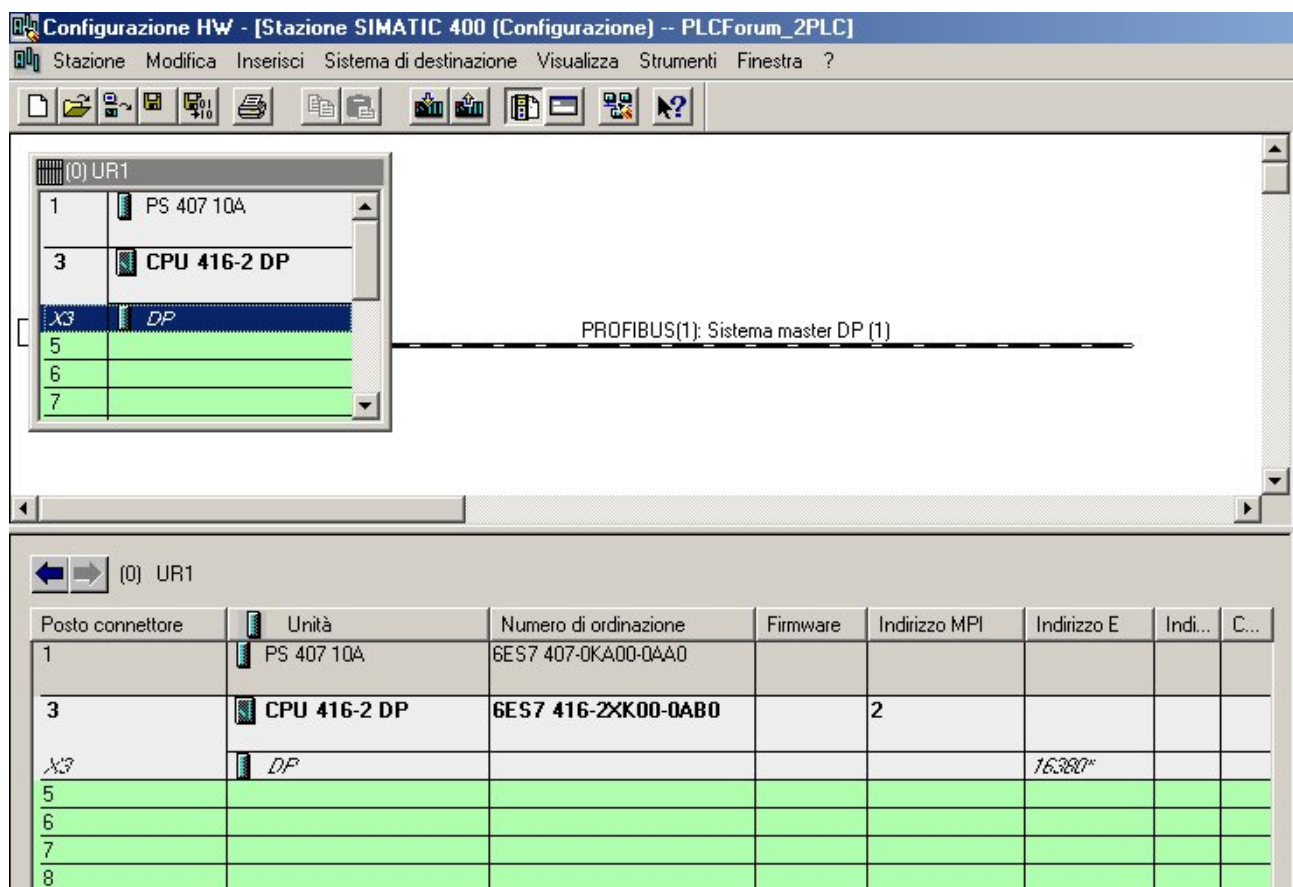


Figura 17

## 4.4 Configurazione della rete Profibus DP

Nel tool **Simatic manager**, Selezionare l'oggetto **Profibus** appena inserito e fare doppio click con il pulsante di sinistra; si attiverà il tool **NetPro** per la configurazione della rete Profibus DP. Tale tool visualizza gli oggetti appena creati, ossia **MPI**, **Profibus** e **Simatic 400**. Si noti che l'unità PLC inserita risulta connessa graficamente al Profibus, in base alla procedura illustrata precedentemente. La Figura 18 illustra ciò che appare nel tool NetPro.

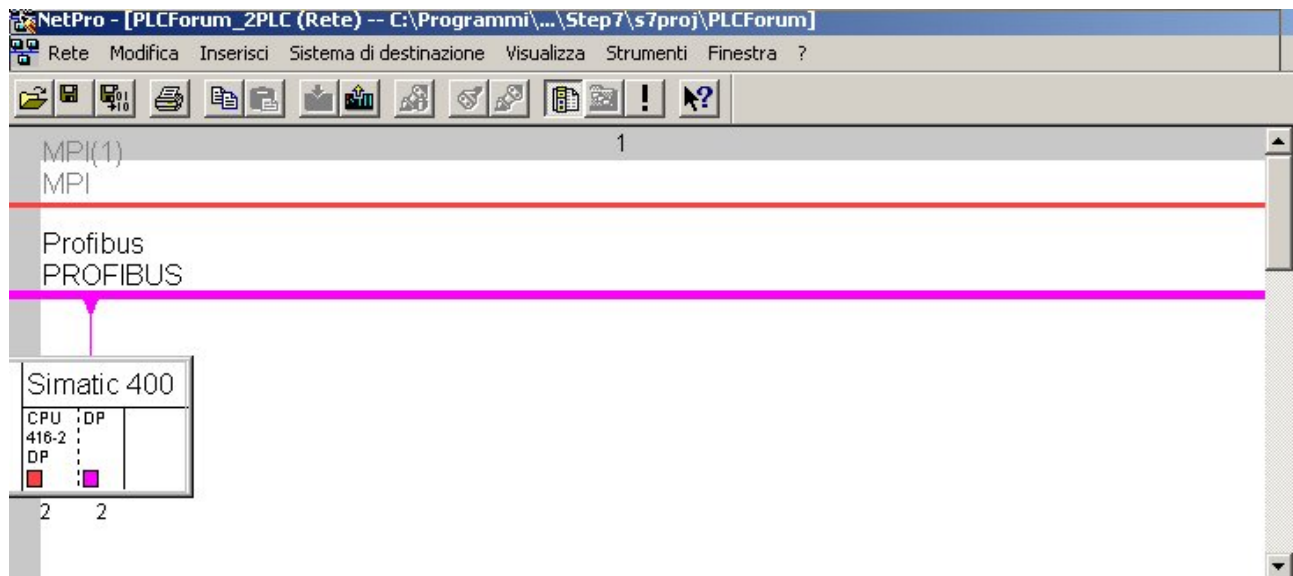


Figura 18

Selezionare la linea relativa all'oggetto Profibus e con il pulsante destro del mouse fare comparire il menu in cui deve essere selezionato **Proprietà dell'Oggetto**.

A questo punto apparirà una finestra contenente due tabelle (**Generale** e **Impostazioni di Rete**). Al fine di configurare la rete Profibus, si dovrà scegliere la tabella **Impostazioni di Rete** (mostrata dalla Figura 19).

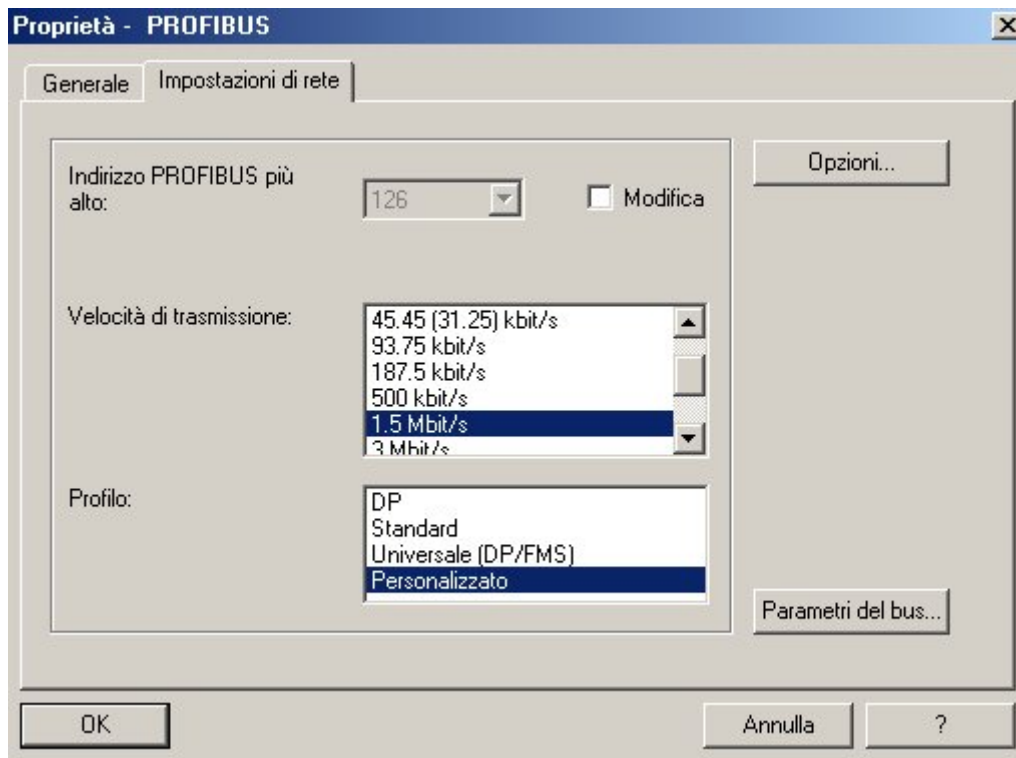


Figura 19

La tabella **Impostazioni di Rete** contiene tutti i campi necessari per la configurazione della rete. Essi sono:

- **Indirizzo PROFIBUS più alto.** Esso è indicato con HSA (Highest Station Address) nello standard Profibus. La sua presenza è dovuta al fatto che nella rete Profibus i dispositivi (sia Master sia Slave) hanno indirizzi crescenti da 1 fino a 126. Visto che il token viene passato solo tra i dispositivi Master, è importante specificare quale sia l'ultimo Master con indirizzo più alto, in modo da limitare il giro del token solo nell'intervallo (1,...,HSA). Molto spesso in fase di configurazione si decide di fissare un valore HSA superiore al numero di Master presenti attualmente (ad esempio ho 4 Master e fisso un HSA pari a 8), in modo da permettere l'estendibilità della rete in futuro. Tale parametro deve essere specificato principalmente nella configurazione multi-master della rete. Nella configurazione mono-master può anche essere lasciato al default 126.
- **Velocità di Trasmissione.** E' la frequenza di trasmissione alla quale dovrà trasmettere il dispositivo che si sta programmando (o meglio l'interfaccia Profibus DP del dispositivo che sto programmando) e tutti gli altri dispositivi (sia Master sia Slave). Per tale motivo tale valore dovrà essere posto pari al minimo tra tutte le velocità di trasmissione dei dispositivi presenti sulla rete. Ad esempio se ho 2 Master, uno che supporta la velocità 12 Mbps - milioni di bit per secondo - e un altro di 500kbps, si dovrà settare la velocità di trasmissione di tutti i dispositivi Master presenti sulla rete a 500kbps. Analogamente se ho una rete con 2 Master entrambi a 12 Mbps e diversi Slave tutti a 12 Mbps e un solo Slave a 500 Kbps, di nuovo la velocità di trasmissione di tutti i Master dovrà essere impostata a 500 kbps. Per una configurazione di default, lasciare il bit rate a 1,5 Mbps.
- **Profilo.** E' un parametro importantissimo nella configurazione della rete. Il motivo della sua esistenza risiede nel fatto che la configurazione della rete può richiedere molta esperienza e capacità tecnica, come si vedrà nel seguito. Per alleggerire il carico di lavoro nella configurazione della rete, esistono dei profili già definiti in cui molti parametri di configurazione sono già settati e dunque non richiedono l'intervento del configuratore. Tali profili sono: **DP**, **Standard** e **Universale**. Esiste un profilo chiamato **Personalizzato** che

permette al configuratore esperto la personalizzazione della configurazione. Tale profilo deve essere usato ad esempio quando la rete Profibus è eterogenea, ossia è composta da dispositivi non tutti SIMATIC Siemens. In tal caso l'uso degli altri profili è impossibile. Per il significato degli altri profili si rimanda al manuale tecnico. In questo capitolo verrà illustrata la configurazione più difficile, ossia quella personalizzata. Per una configurazione di default, lasciare settato il Profilo DP.

- **Parametri del bus.** Facendo click su questo pulsante si aprirà la finestra che permette l'assegnazione dei valori ai principali parametri che regolano il funzionamento di un Profibus DP e che sono stati illustrati nel Capitolo 3 (si veda la Figura 20).

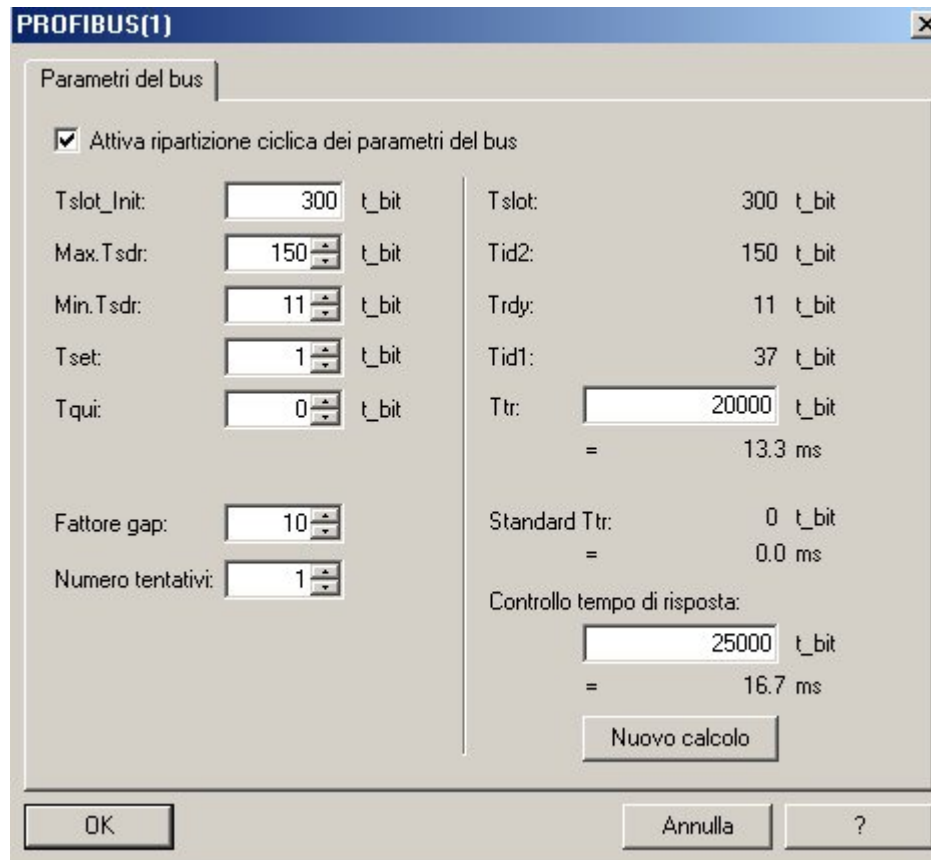


Figura 20

La possibilità di modificare tali parametri è consentita solo se l'utente ha selezionato il profilo Personalizzato, mentre nel caso in cui l'utente abbia scelto gli altri profili, i valori assegnati per default ai parametri Profibus non possono essere modificati. Ciò perché i profili preconfigurati sono stati definiti per specifiche configurazioni mono o multi-master, generalmente caratterizzate dalla presenza solo di dispositivi Siemens. Il profilo Personalizzato deve essere usato per reti generiche e dunque richiede la configurazione manuale di tutti i parametri. Nel seguito verranno specificati i campi della finestra che si attiva cliccando sul Parametri del bus. La tabella **Parametri del bus** presenta:

- **Attiva ripartizione ciclica dei parametri del bus.** Questa opzione è presente in tutti i profili ed è l'unica che può essere settata in tutti i profili. Quando questa opzione viene abilitata, l'insieme dei parametri definiti per l'oggetto Profibus in corso di definizione, sono trasmessi ciclicamente da tutti i Master DP che sono attivi nella rete Profibus DP. I dati vengono trasmessi tramite un telegramma multicast utilizzando il DSAP 63. Questa opzione è consigliata nel caso in cui si voglia connettere una unità di programmazione ad una rete Profibus in esecuzione, e non si conoscono gli esatti parametri del bus. Tramite l'invio ciclico dei parametri, l'unità



di programmazione può acquisire tali informazioni. Questa opzione non deve essere scelta nel caso si sia selezionata l'opzione ciclo di bus costante, descritta in seguito.

- **Tslot\_Init, Max.Tsdr, Min.Tsdr, Tset, Tqui, Tslot, Tid2, Trdy, Tid1, Ttr.** Sono stati già descritti nel Capitolo 3. Si ricordi che l'unità di misura è il  $t_{bit}$ , che dipende dal bit rate del mezzo di trasmissione, come definito dalla tabella 1. Si noti che il parametro  $T_{slot\_init}$  è il valore di inizializzazione del parametro  $T_{slot}$ ; si noti poi che i parametri  $Tid1$ ,  $Tid2$ ,  $Trdy$  vengono impostati automaticamente dal sistema.
- **Fattore gap, G**, già definito nel capitolo 3.
- **Opzioni.** Nella finestra Impostazioni di rete si trova un pulsante chiamato Opzioni. Facendo click su questo pulsante si aprirà la finestra che permette l'assegnazione di valori opzionali al funzionamento del sistema. Non sono generalmente indispensabili e servono solo a migliorare le prestazioni della rete Profibus. La finestra che si apre cliccando su Opzioni presenta tre tabelle: **Equidistanza**, **Nodi di rete** e **Cavi**. Le opzioni presenti in tali sottofinestre saranno descritte in seguito.
  - **Tabella Equidistanza.** Tale tabella permette di selezionare l'opzione di ciclo equidistante. Un ciclo di bus equidistante significa che l'intervallo di tempo tra due successivi possessi del token (ossia diritto alla trasmissione) da parte del Master DP, è costante. Ovviamente ciò può essere realizzato in reti con un solo Master DP di classe 1, che deve interrogare ciclicamente gli slave e ha la necessità che il ciclo di scansione (lettura/scrittura) di tali slave sia deterministico, ossia avvenga con ciclo di bus costante. Nel caso in cui si scegli l'equidistanza del ciclo di bus, verrà abilitato un pulsante che permette il calcolo automatico del ciclo più idoneo (ossia compatibile) con la configurazione Profibus definita. Visto che lo scambio dati nel Profibus può non solo interessare il Master DP e i suoi Slave, ma anche altri dispositivi connessi al Profibus (ad esempio una unità di programmazione e di monitoraggio), l'utente può specificare quanti di questi dispositivi sono presenti. In tal caso, il calcolo del ciclo viene fatto dal sistema in modo da garantire scansioni costanti con gli slave e scambi non ciclici con gli altri dispositivi presenti nella rete. Si faccia la prova a settare il numero di altri dispositivi collegati alla rete Profibus a 0 e ad aumentare tale numero. Si vedrà che il tempo di ciclo aumenta, per dare la possibilità di trasmettere anche il traffico scambiato con tali dispositivi aggiuntivi. Ovviamente l'utente può scegliere un qualunque valore del tempo di ciclo; l'aumento è sempre possibile, mentre la riduzione del valore consigliato può essere pericoloso, in quanto il tempo di ciclo può non mantenersi costante. Come si vede esiste un pulsante **Dettagli** che permette la visualizzazione delle porzioni del tempo di ciclo assegnate al master per la trasmissione ciclica (ossia regolare nel tempo) aciclica (ossia ad intervalli random) con gli Slave e per la trasmissione con altri dispositivi connessi al Profibus. La porzione ciclica non può essere cambiata, mentre le altre due porzioni possono essere modificate dall'utente.
  - **Tabella Nodi di Rete.** Il sistema di comunicazione Profibus DP che si sta configurando potrebbe contenere dispositivi che non possono essere inclusi nel progetto creato in ambito Step7; ciò accade ad esempio quando la rete Profibus DP contiene dispositivi Master non Siemens che non possono essere riconosciuti nell'ambiente Step7. A tal fine è possibile impostare manualmente il numero di dispositivi realmente presenti, tramite la tabella Nodi di rete. In questa tabella si vede che è possibile impostare il numero complessivo di nodi Master, il numero di nodi Slave, il numero di nodi che trasmettono traffico aggiuntivo (identificato con FDL/FMS/S7) e il carico di comunicazione relativo a tale traffico. E' importante mettere in evidenza che una rete Profibus DP può contenere dispositivi che scambiano informazioni utilizzando altri protocolli differenti dal Profibus DP,

sempre compatibili con esso a livello fisico e data link; essi sono i protocolli Profibus FMS, FDL e S7. Come già accennato nei capitoli precedenti a proposito del Profibus FMS, sono protocolli del tutto identici al protocollo Profibus DP a livello di trasmissione fisica e gestione logica dei messaggi (ossia hanno lo stesso livello fisico e data link) ma si differenziano dal protocollo Profibus DP per il fatto di possedere anche un terzo livello applicativo. Lo scambio informativo prodotto da due dispositivi Profibus FMS, ad esempio, può essere convogliato su una rete Profibus DP, ma tale scambio viene del tutto ignorato dai dispositivi Profibus DP, che sono incapaci di decifrare i messaggi Profibus FMS. Lo stesso per i protocolli FDL e S7. Per tale motivo, è possibile manualmente indicare la presenza di tali dispositivi e il relativo carico generato. Tale opzione è utile soprattutto quando si utilizzando gli strumenti automatici di dimensionamento e configurazione della rete Profibus. In tal caso le informazioni specificate in questa finestra permettono un adeguato dimensionamento. Nella configurazione personalizzata, tale finestra può non essere selezionata. Questa opzione non è disponibile per il profilo DP.

- **Tabella Cavi.** Questa è l'ultima delle opzioni che un utente può scegliere. Permette di specificare la tipologia di cavi utilizzati (RS 485 o fibra ottica), la relativa lunghezza e l'eventuale presenza di ripetitori (nel caso di RS 485) o di Optical Link Modules (OLM), nel caso di fibra ottica. Tale opzione serve al tool Step 7 qualora vengono richieste dall'utente delle configurazioni dirette automatiche, al fine di scegliere la più adeguata configurazione.

## 4.5 Configurazione dei dispositivi afferenti la rete Profibus DP

L'ultimo passo nella configurazione consiste nella definizione dell'hardware presente nel sistema Profibus DP. Visto che fino ad ora si è proceduto solo a definire le caratteristiche del dispositivo Master DP che si sta programmando (incluso il rack e l'alimentazione), è necessario specificare quali sono i DP Slaves che sono di interesse al Master. Si presti attenzione al fatto che in una rete multi-master, solo alcuni Slaves sono di interesse per ogni Master; i altri termini gli Slaves vengono suddivisi logicamente in tanti gruppi; tutti gli Slaves di un gruppo vengono letti o scritti da un solo master DP. In tal caso si dovranno configurare solo gli Slaves che sono di interesse per il Master DP che si sta programmando, tralasciando gli altri. L'informazione sul numero totale di Master DP e Slaves viene, se si vuole, specificata nelle finestre opzionali viste precedentemente.

Si termini il programma **NetPro** in esecuzione e si ritorni alla finestra Simatic Manager. Si selezioni l'oggetto Profibus e si faccia un doppio click. Si aprirà il tool Configurazione HW. Supponendo di aver precedentemente configurato un Master DP **Simatic 400**, la finestra che dovrebbe apparire è quella mostrata in Figura 21.

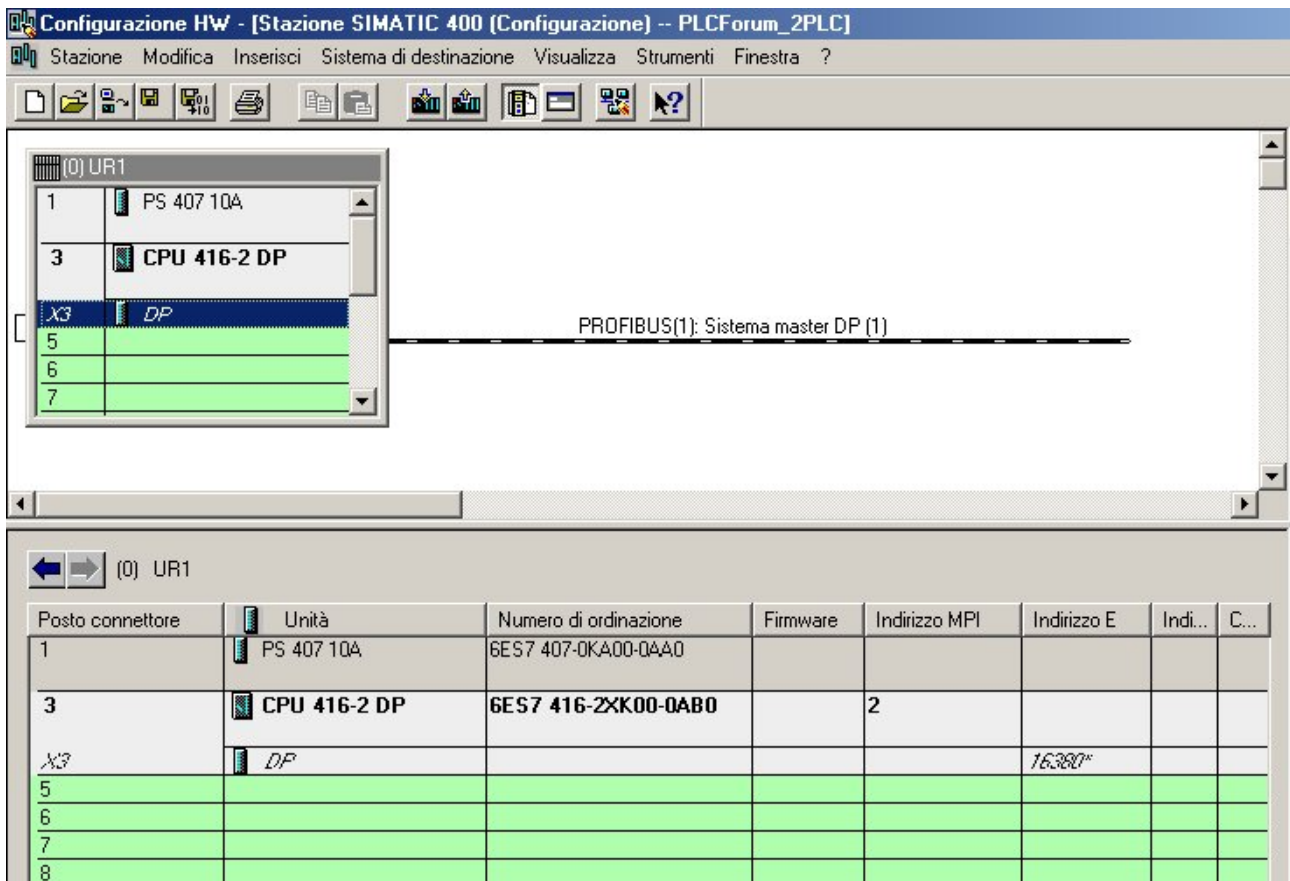


Figura 21

A questo punto si inizia la definizione dei dispositivi DP Slaves. Vi possono essere tre casi nella definizione di un DP Slave:

- DP Slave generico (ossia modulo I/O remoto o PLC) di marca non Siemens,
- un modulo di I/O remoto di marca Siemens (ad esempio ET 200B o ET 200M)
- una CPU Siemens (ad esempio serie 300 o 400) usata come Slave DP intelligente (I-Slave).

#### 4.5.1 Dispositivo Slave DP Generico

La configurazione di uno Slave DP avviene automaticamente grazie all'esistenza dei file GSD, già illustrati precedentemente, che descrivono tutte le caratteristiche di uno Slave DP, tra cui, ad esempio, i bit/rate supportati, i servizi supportati, le caratteristiche fisiche espresse sia in termini temporali sia in termini di numero e tipologia di ingressi/uscite.

Allo scopo di configurare uno Slave DP è necessario possedere il relativo file GSD (ogni ditta produttrice mantiene un sito da cui è possibile scaricare gratuitamente i file GSD aggiornati per ogni dispositivo Slave DP prodotto).

La configurazione di uno Slave DP avviene nell'ambito del programma Configurazione HW, già citato in precedenza.

Si supponga di considerare il progetto definito ai punti precedenti, caratterizzato dalla presenza di un PLC **Simatic 400**, CPU 416-2DP e di una rete Profibus già configurata.

Si selezioni l'oggetto **Simatic 400** e si faccia doppio click. Si attiverà il tool Configurazione HW. Nella porzione destra sono presenti gli oggetti che descrivono l'hardware disponibile (ad esempio rack, alimentatori, CPU), come già detto. Tra questi oggetti sono presenti anche i file GSD che descrivono gli Slave DP. Se lo slave non è di marca Siemens è probabile che il suo file GSD non sia disponibile. E' necessario, dunque caricarlo nel tool Configurazione HW, prima di poter procedere alla configurazione di uno Slave DP. Si supponga di voler configurare due Slave DP costituiti da un

modulo di Ingressi remoto a 16 bit prodotto da SAIA-BURGESS, denominato RIO 16I, e un modulo di 16 uscite digitali sempre della stessa ditta, denominato RIO 16O. Nel tool Configurazione HW, si vada su **Strumenti**→**Installa nuovo file GSD**. Viene aperta una finestra tramite la quale è possibile specificare la cartella in cui il file da aggiungere è memorizzato. Una volta selezionato il file esso verrà aggiunto nella struttura ad albero visibile nella porzione destra del tool Configurazione HW.

A questo punto è possibile selezionare i file GSD aggiunti e trascinarli nella finestra in alto del tool Configurazione HW posizionando il mouse nelle vicinanze del Profibus. Si aprirà una finestra tramite la quale è possibile configurare l'indirizzo assegnato allo Slave DP che si sta inserendo. Si supponga di aggiungere il file GSD del modulo di 16 Ingressi. Selezionando tale modulo, nella porzione in basso a sinistra del tool Configurazione HW vengono riassunte le caratteristiche dello Slave in termini di ingressi e uscite e vengono visualizzati gli indirizzi logici, appartenenti all'area di immagine di processo, che il sistema ha assegnato ai moduli di I/O appena installati. Ad esempio la Figura 22 mostra che 2 byte di ingresso del modulo RIO 16I hanno indirizzi IB0 (ossia I0.0, I0.1, ....., I0.7) e IB1 (ossia I1.0, I1.1, ....., I1.7).

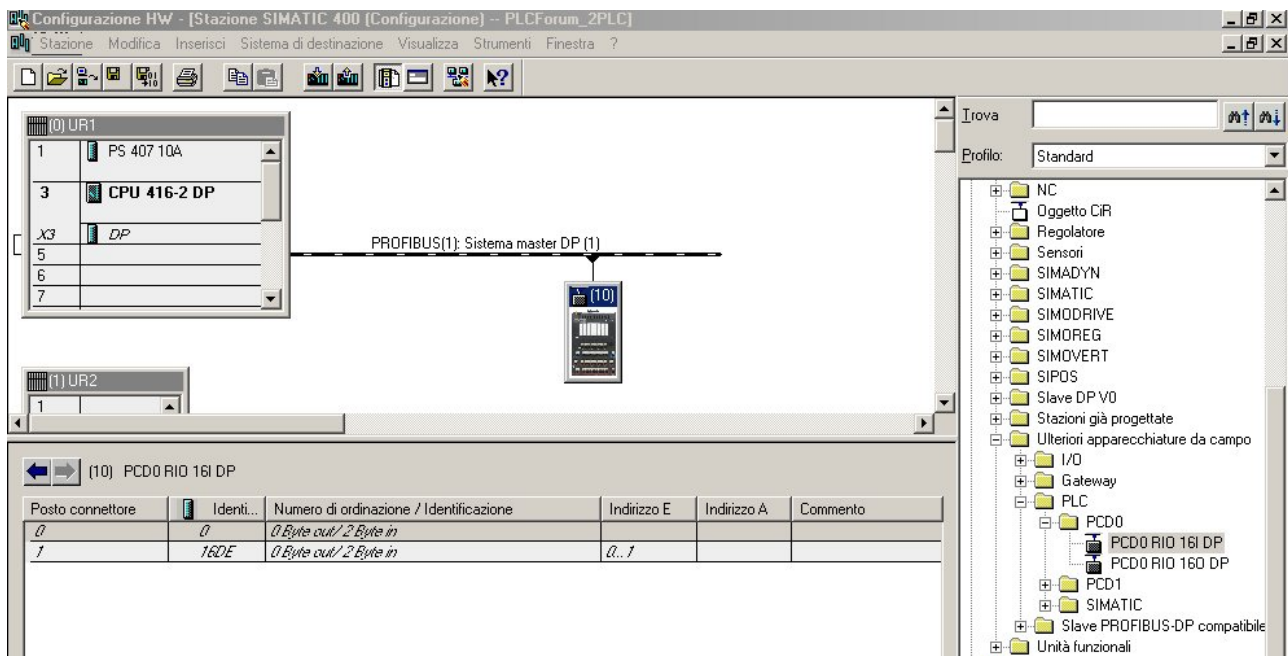


Figura 22

Si installi adesso lo slave di 16 uscite (RIO 16O) e si selezioni tale oggetto. Apparirà la finestra mostrata in Figura 23, in cui si vede che il sistema ha assegnato gli indirizzi QB0 e QB1 ai 2 byte di uscita del modulo remoto.

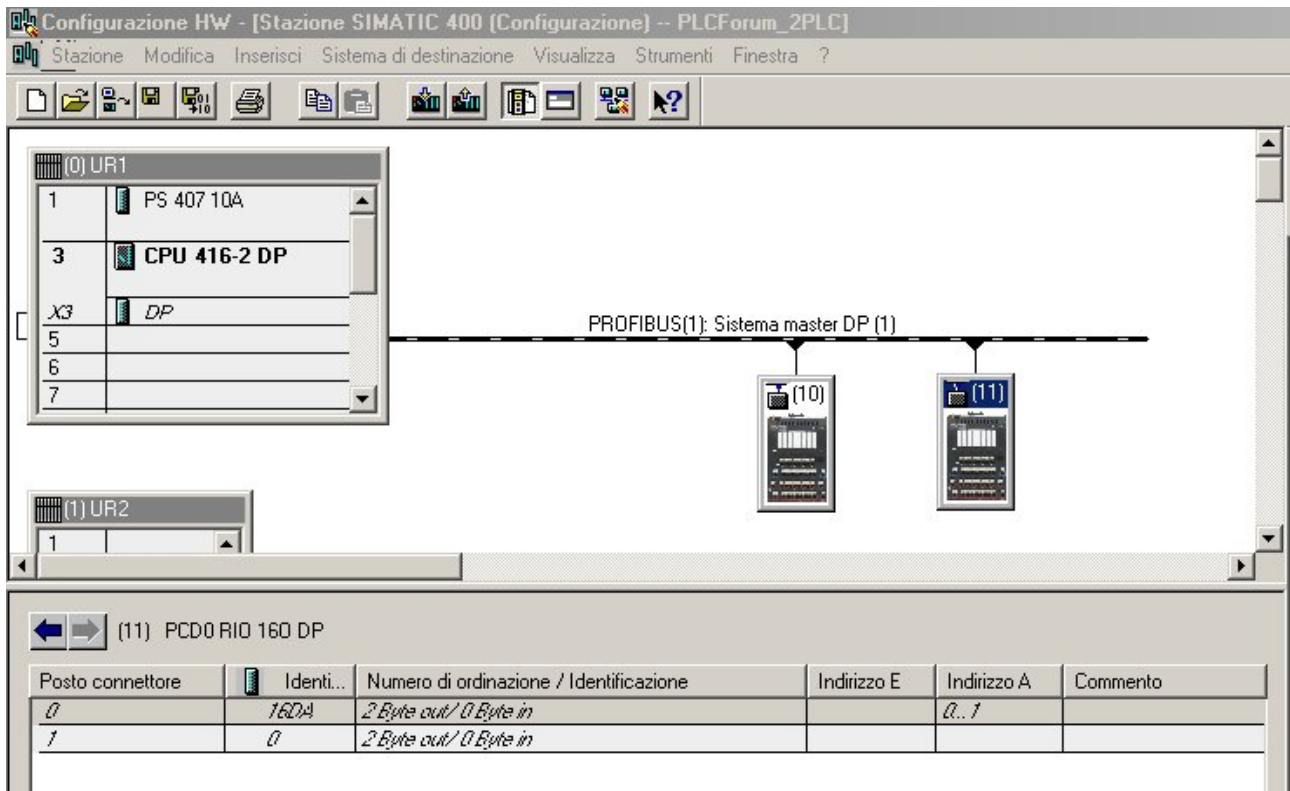


Figura 23

Ovviamente gli indirizzi possono essere cambiati, selezionando la riga corrispondente all'indirizzo assegnato dal sistema e facendo doppio click. Apparirà una finestra mostrata in Figura 24 tramite la quale è possibile modificare gli indirizzi.

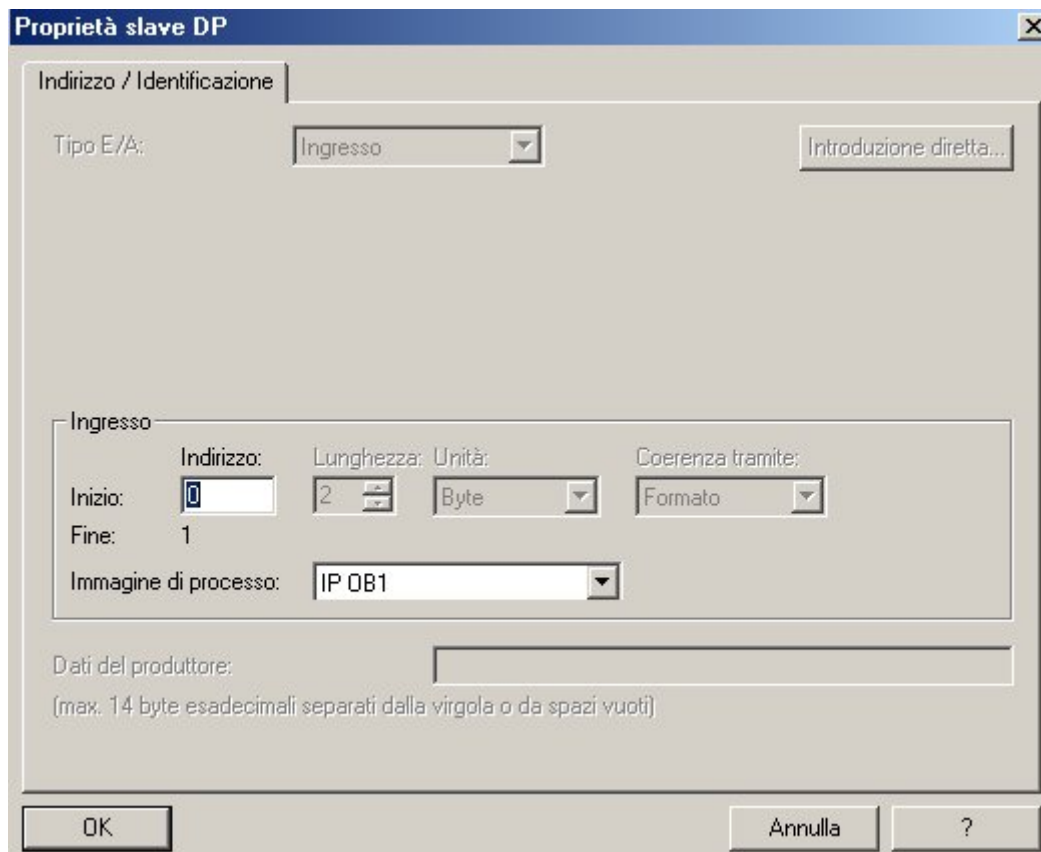


Figura 24

## 4.5.2 Dispositivo I/O Siemens

La procedura da seguire per la configurazione di uno Slave DP costituito da un dispositivo Siemens (ad esempio il modulo di I/O remoto ET200B) è identico a quello descritto per un dispositivo generico, tranne per il fatto che non è generalmente richiesto il caricamento del file GSD del dispositivo, in quanto la libreria presente nel Simatic comprende già la maggior parte dei dispositivi Slave DP Siemens. La Figura 25 mostra ad esempio l'inserimento del dispositivo Siemens ET200B (2 byte di ingresso/2 byte di uscita). Come si vede nella porzione destra compare il file GSD all'interno della cartella ET 200B.

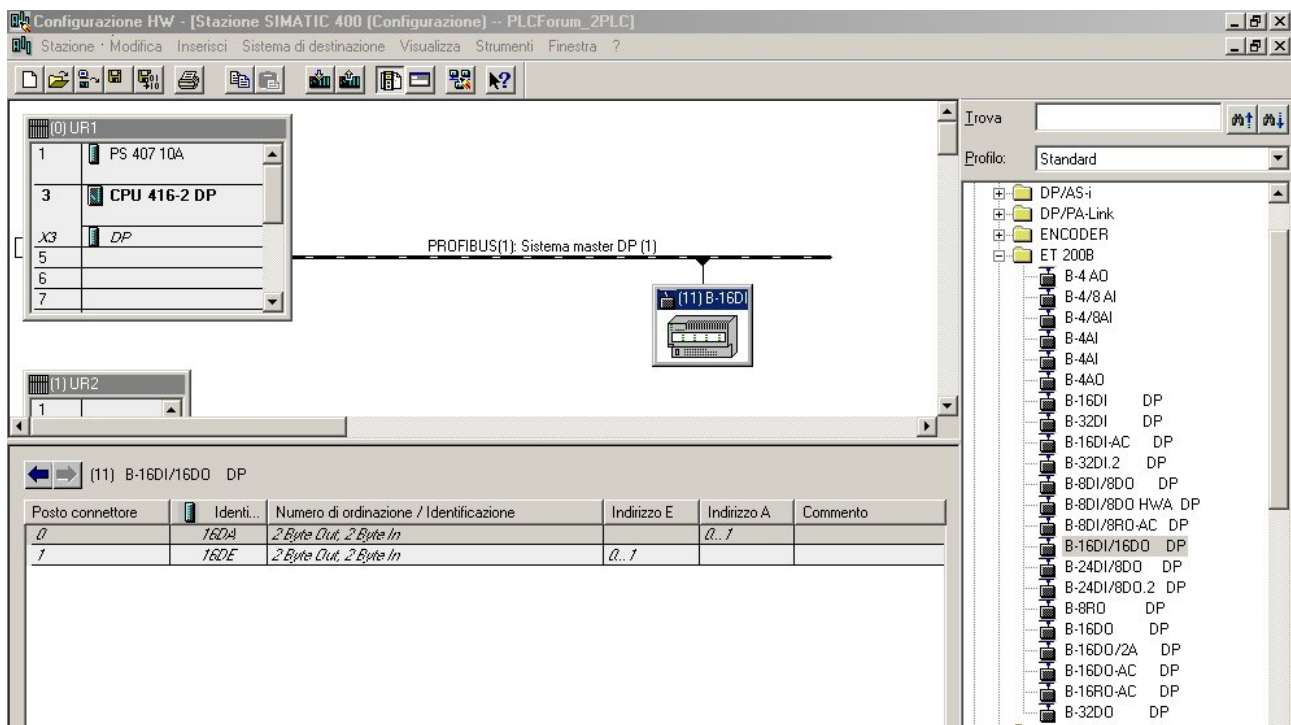


Figura 25

## 4.5.3 PLC I-Slave Siemens

In questo paragrafo verrà illustrato come configurare un PLC Siemens (Simatic 300 o 400) come I Slave DP, ossia come slave intelligente (capace di elaborare le informazioni ricevute). In verità, la procedura di configurazione verrà descritta fino ad un determinato passo, per poi essere ripresa nel capitolo successiva, una volta introdotte delle nozioni propedeutiche concernenti lo scambio Master-Slave tramite le primitive di lettura e scrittura, DPRD\_DAT e DPWR\_DAT.

Si supponrà, solo per esempio, di voler definire un PLC Simatic 300 (CPU 315-2DP) come I-Slave DP. Il primo passo è quello di definire l'oggetto **Simatic 300** nel Simatic Manager, come fatto per il Simatic 400. Si seguano tutte le procedure illustrate precedentemente per inserire la stazione Simatic 300. Si inserisca, come esempio, prima il RACK 300 (in questo caso nella directory relativa a RACK-300, troverete solo un oggetto: la Guida profilata), poi l'alimentatore PS-300 (si scelga ad esempio il modello PS307 2A) ed infine la CPU315-2DP (scegliendo ad esempio il firmware 6ES7 315-2AF01-0AB0).

Appena si inserisce la CPU, si aprirà la tabella **Proprietà-Interfaccia Profibus DP**, tramite la quale è possibile inserire l'indirizzo assegnato allo Slave DP e collegare l'interfaccia alla rete **Profibus** precedentemente definita.

E' necessario adesso modificare la modalità di funzionamento standard dell'interfaccia DP, che ha come valore di default **Master DP**; ovviamente dovrà essere specificato che l'interfaccia Profibus DP della CPU 315-2DP dovrà svolgere le funzioni di Slave DP. Per far ciò, si selezioni l'interfaccia DP (ad esempio facendo click sulla riga in cui è presente la sigla DP nella tabella in basso a sinistra del tool Configurazione HW) e si faccia doppio click. Comparirà la finestra **Proprietà-DP**. Si vada nella tabella **Modo di Funzionamento** e si scelga l'opzione Slave DP. Si preme OK per tornare al tool Configurazione HW.

Lo scambio dati tra Master e I-Slave è generalmente caratterizzato dall'uso di particolari funzioni di comunicazioni (DPRD\_DAT e DPWR\_DAT), descritte nel capitolo successivo, che utilizzano particolari aree di memoria per lo scambio informativo; tali aree sono dette aree consistenti.

Il capitolo successivo illustrerà in dettaglio la sintassi e la semantica delle precedenti funzioni di comunicazione e il significato delle aree di memoria consistenti. Lo stesso capitolo descriverà successivamente come definire tali aree dati sia nel Master DP sia nello I-Slave. Il Capitolo 6, illustrerà, invece, come scrivere un programma sia per Master sia per Slave DP che utilizzi tale aree.

Per il momento si sospenda la creazione del progetto, per poi riprenderla al Capitolo 5, durante la spiegazione della creazione di aree dati consistenti.

## 5 Primitive di servizio disponibili in ambiente Simatic per la comunicazione

Come detto nel Capitolo 2, il Protocollo Profibus DP offre due differenti categorie di servizi:

- Servizi per lo scambio Dati. Sono usati per scrivere e leggere dati dei dispositivi di I/O distribuito. Sono servizi uno-a-uno e sono confermati.
- Servizi di Controllo. Permettono di inviare comandi di controllo a dispositivi Profibus DP. Sono servizi broadcast/multicast e non sono confermati.

In questa dispensa verranno approfonditi solo i servizi per lo scambio dati, che sono ovviamente da considerarsi di importanza primaria, in quanto permettono l'esecuzione dell'algoritmo o degli algoritmi di controllo distribuiti sui dispositivi Profibus DP.

Le principali caratteristiche dei servizi disponibili per lo scambio dati si possono riassumere nei seguenti punti:

- lo scambio dati può avvenire tra qualunque dispositivo prodotto da qualunque ditta, purché sia compatibile con lo standard Profibus DP
- lo scambio dati è confermato, ossia per ogni dato inviato è fornita una conferma sulla sua ricezione.
- L'entità dei dati scambiati per ogni transazione è minore o uguale a 244 bytes

Il Capitolo 2 ha messo in luce che lo scambio dati viene fondamentalmente realizzato dal servizio Profibus DP Data Exchange.

Scopo di questo capitolo è capire come l'ambiente Simatic realizzi la comunicazione basata sui servizi Profibus DP tra un Master e gli Slave (sia I/O distribuiti sia I Slave). Prima di illustrare ciò verranno forniti alcuni concetti di base sulla gestione degli Ingressi e Uscite in un PLC. Tale argomento è propedeutico alla comprensione del contenuto del Capitolo.

### 5.1 Gestione delle immagini di processo degli ingressi e delle uscite

La gestione degli ingressi e delle uscite in un PLC è gestita in base alle aree immagini di processo degli ingressi e delle uscite, che sono due particolari aree di lavoro in cui vengono mantenute le immagini appunto (ossia le copie) degli ingressi e delle uscite reali.

In particolare, in un qualunque PLC, tutti i valori assunti dagli ingressi vengono copiati nell'area di immagine di processo degli ingressi all'inizio di ogni ciclo di scansione del programma (ad esempio l'OBI in ambiente Simatic). Nonostante gli ingressi reali cambino, la CPU vedrà sempre l'immagine di processo degli ingressi statica per tutto il corso del programma (ossia per tutta la scansione del programma).

Analogamente le uscite vengono elaborate durante il corso del programma e solo alla fine di ogni ciclo di scansione le uscite vengono aggiornate, ossia l'area immagine di processo delle uscite viene ricopiata nelle uscite reali. La Figura 26 mostra quanto detto.

Nei PLC Siemens e in base al linguaggio IEC61131-3, le aree di ingresso/uscita vengono identificate con le sigle I e Q.



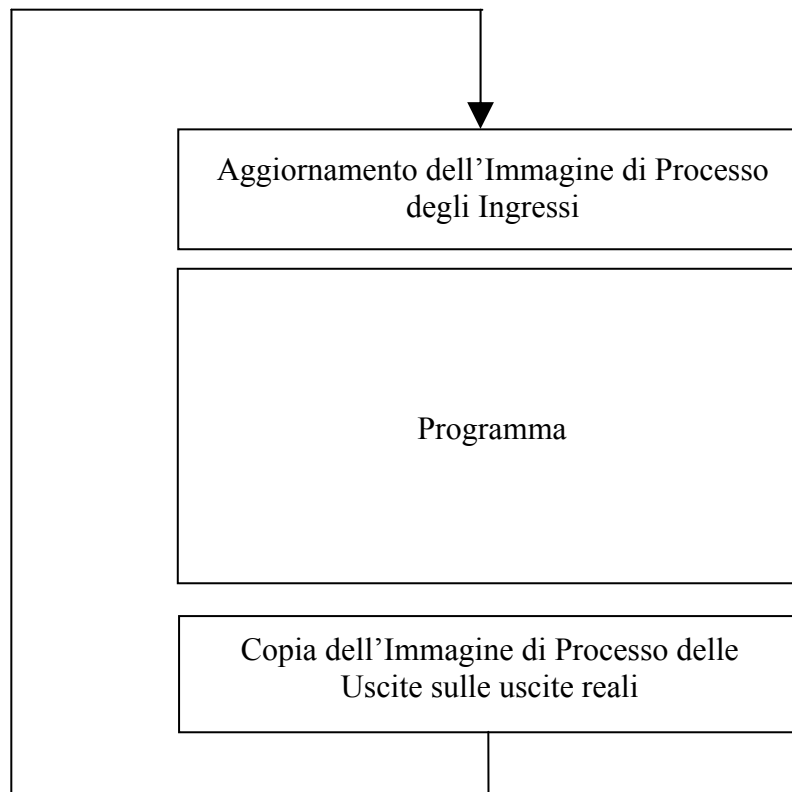


Figura 26 – Aggiornamento delle Aree Immagine di Processo I/O

## 5.2 Scambio dati basato su accesso diretto I/O

Il meccanismo più semplice per lo scambio dati tra Master DP e Slave DP disponibile in ambiente Simatic Step7, si basa sulla corrispondenza dei meccanismi di accesso diretto agli ingressi e alle uscite con il servizio Data Exchange presente nel Profibus DP. E' importante evidenziare che tale scambio dati è possibile solo se si accede ai bit, ad un Byte, a due byte o a quattro byte. Non è possibile cioè accedere a tre byte di uno Slave DP o ad un numero superiore a 4 byte, tramite i meccanismi di accesso diretto I/O.

L'accesso diretto agli ingressi e alle uscite fa uso della notazione standard IEC 61131-3 e ai comandi di accesso diretto agli ingressi e alle uscite. In particolare, la notazione standard relativa agli ingressi è **Izy.y**, dove z specifica se l'accesso è relativo ad un bit (in tal caso secondo lo standard IEC 1131-3 permette di omettere il simbolo del bit), oppure ad un byte, o a multipli di esso; invece y.y identifica l'indirizzo dell'ingresso. Per le uscite si utilizza l'analoga notazione **Qzy.y**. I comandi di accesso diretto agli ingressi e alle uscite dipendono dal linguaggio di programmazione; in linguaggio Ladder, ad esempio, il contatto permette di leggere un ingresso binario, mentre un coil permette di scrivere una uscita binaria. In genere per leggere e scrivere un byte può essere utilizzata la funzione MOVE disponibile in ambiente Simatic.

In base a quanto detto, l'accesso agli ingressi e alle uscite di uno Slave DP da parte di un Master DP, si realizza tramite i comandi di accesso diretto I/O e alla notazione standard IEC 1131-3. La corrispondenza tra ingressi e le uscite reali di uno Slave DP e gli ingressi e le uscite indicate tramite la notazione standard avviene grazie al fatto che, in fase di configurazione degli Slave DP, a ciascuno Slave vengono fatti corrispondere opportuni intervalli di indirizzi diretti (di ingresso o di uscita); si vedano, a tal proposito, le Figure 22, 23, 24 o 25.

Lo scambio dati Master-Slave DP basato su accesso diretto I/O si realizza grazie al fatto che tutti i dati letti e scritti dal Master DP da e verso i suoi Slave DP (tramite ad esempio il servizio

DataExchange) vengono memorizzati automaticamente nell'area delle immagini di processo degli ingressi e delle uscite, mostrata in Figura 26. In particolare, in presenza di accesso diretto agli ingressi e alle uscite di uno Slave DP tramite il meccanismo di accesso diretto prima illustrato, viene realizzato l'aggiornamento delle aree immagine di processo degli ingressi e delle uscite; come mostrato in Figura 27.

Come è stato detto in precedenza questo meccanismo è limitato al byte, coppia di byte o doppia coppia di byte letti/scritti da/su uno Slave DP da parte di un Master DP. La Figura 27 mette in evidenza che ogni byte letto dal Master DP viene aggiornato nell'area degli ingressi e ogni byte presente nell'area delle uscite viene scritto nello Slave DP.

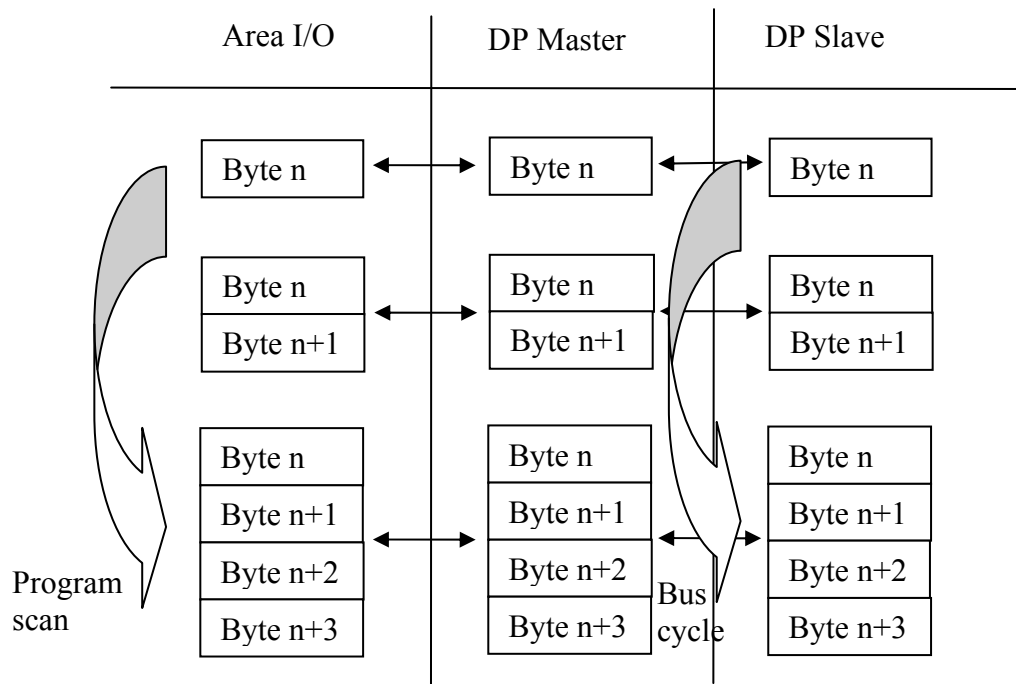


Figura 27 – Interazione tra Scambio Profibus DP e Aree Immagini di Processo

Si consideri ad esempio la Figura 22, in cui è stato configurato uno Slave di 16 bit di ingresso. In tal caso erano state assegnati gli indirizzi IB0 e IB1 ai due byte dello Slave DP. Dunque se volessi leggere il terzo bit dello Slave DP, basterebbe utilizzare un comando di accesso diretto (ad esempio un contatto in linguaggio Ladder) associando l'indirizzo IO.2, che corrisponde al terzo bit dello Slave DP configurato in Figura 22.

### 5.3 Scambio dati consistente

Lo scambio dati consistente non avviene tramite accesso diretto ma in base all'utilizzo di particolari funzioni di sistema di comunicazione. Lo scambio dati consistente può essere applicato solo per gli I-Slave.

Le funzioni utilizzate dipendono dalla tipologia di interfaccia Profibus DP del dispositivo Master DP. Esistono, infatti, due differenti interfacce:

- Modulo di interfaccia esterno alla CPU (ad esempio S7-300 CP 342-5)
- Interfaccia Integrata nella CPU

Negli esempi di configurazione precedenti abbiamo sempre considerato l'interfaccia integrata e lo continueremo a fare anche nel seguito. Dunque verranno illustrati le modalità di comunicazione esistenti in ambiente Simatic per interfacce Profibus DP integrate.

Come detto in precedenza, lo scambio dati basato sui meccanismi di accesso diretto agli ingressi e alle uscite avviene grazie al fatto che tutti i dati letti o scritti dal Master DP (da/verso gli Slave DP) transitano attraverso le aree immagine di processo. Come detto nei paragrafi 5.1 e 5.2, l'aggiornamento di tali aree è legato alla scansione ciclica del programma, nel senso che gli ingressi vengono aggiornati all'inizio del ciclo e le uscite alla fine (oppure, in alcuni modelli di CPU Siemens sia gli ingressi sia le uscite vengono aggiornate alla fine della scansione). D'altra parte, l'aggiornamento dei dati di I/O da e verso dispositivi Slave DP è determinato esclusivamente dallo scambio ciclico del DP Master con i DP Slaves (che dipende dalla rotazione del token) ed è del tutto indipendente dalla scansione ciclica del programma. Ciò significa che di fatto c'è un conflitto sull'accesso all'area immagine dei processi I/O e dunque per ogni accesso all'area immagine dei processi sia da parte della CPU sia da parte dell'interfaccia Profibus, è necessario garantire la consistenza dell'aggiornamento dei dati ricevuti e mandati da e verso la rete Profibus DP. La consistenza per le strutture dati composte da 1, 2 e 4 byte è garantita nei sistemi PLC Siemens; mentre non lo è per strutture dati composte da 3 byte o più grandi di 4 bytes.

In tal caso, sono necessarie particolari strutture dati e particolari funzioni che garantiscano la lettura e la scrittura dell'area immagine di processo I/O in modo consistente.

Le strutture dati che vengono utilizzate non hanno niente a che fare con l'area immagine di processo I/O. Sono dei particolari moduli di ingresso e uscita che vengono definiti in fase di configurazione, sia nel Master DP sia nello I-Slave DP. I dati letti e scritti tramite il Profibus DP da e verso gli I-Slave DP transitano in questi moduli. Ad un modulo di uscita del Master deve corrispondere un modulo di ingresso dello I-Slave, per realizzare lo scambio dati dal Master verso Slave. Ad un modulo di uscita dello I-Slave deve corrispondere un modulo di ingresso del Master per realizzare lo scambio in senso inverso. Ovviamente lo scambio dati è sempre realizzato dalla funzione DataExchange che aggiorna i precedenti moduli.

La Figura 28, mostra un esempio di tali aree.

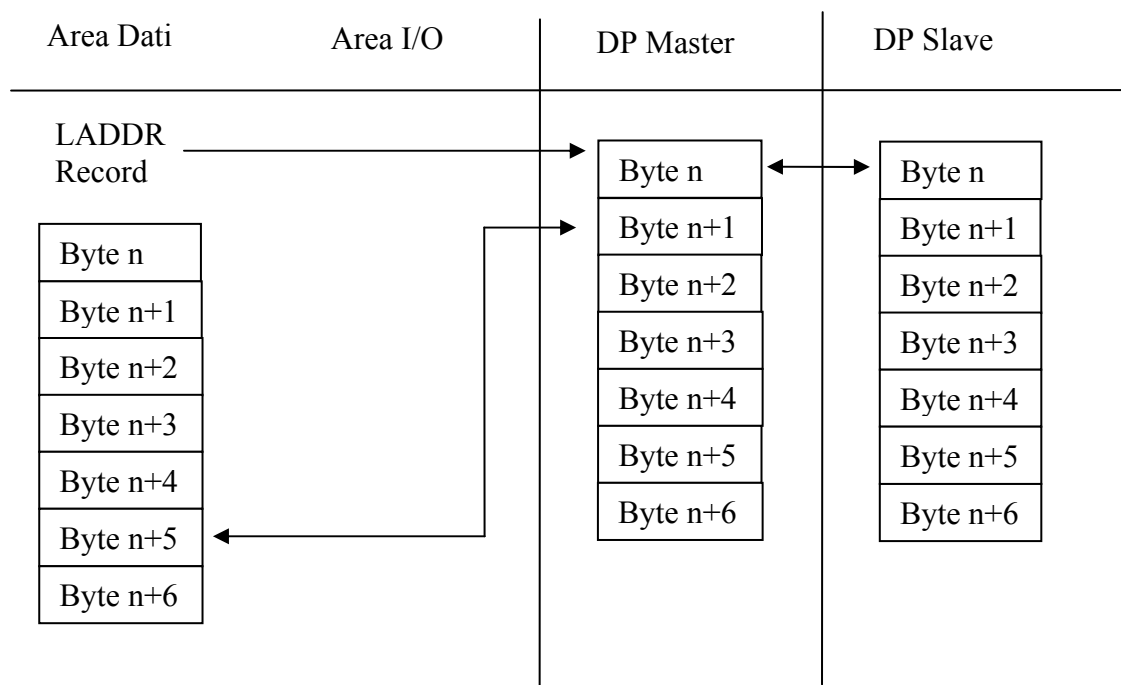


Figura 28 - Accesso ad un blocco di dati consistente > 4 byte

La figura mostra in corrispondenza di Master DP e Slave DP due moduli di I/O (potrebbero essere uno di uscita e l'altro di ingresso o viceversa). Ogni modulo è contraddistinto da un parametro LADDR mostrato solo per il Master DP in Figura 28 (ma esiste anche per lo I-Slave), che è il puntatore al modulo area dati di ingresso da cui il DP Master o l'I-Slave deve leggere o il puntatore al modulo area dati di uscita su cui il DP Master o lo I-Slave deve scrivere.

In figura viene mostrato che, noto LADDR e nota la lunghezza del modulo, è possibile trasferire il contenuto di un modulo del Master Dp (o nello I-Slave) in una zona di memoria interna e viceversa. Ciò avviene grazie a particolari funzioni che leggono e scrivono in questi moduli in modo consistente (ad esempio se sto leggendo 20 bytes dal modulo di I/O non voglio che nel frattempo il Master DP aggiorni in questo modulo i dati letti dallo Slave DP).

Lo scambio dati consistente è basato sulle funzioni DPRD\_DAT e DPWR\_DAT e si basa, appunto, sui particolari moduli di memoria definiti sia nel Master sia nello Slave, chiamate aree dati consistenti, appena descritti. Esse sono strutture dati abbastanza complesse, la cui area ha una lunghezza di 3 bytes o di un numero superiore a 4.

### **5.3.1 SFC14 DPRD\_DAT**

La funzione di sistema SF14 viene utilizzata per leggere dati, in modo consistente, da un modulo di ingresso (modulo che raccoglie degli ingressi provenienti dal Profibus DP) e per trasferire tali dati in una qualunque area di memoria del PLC (ad esempio area I o area DB). Ogni accesso di lettura si riferisce ad un modulo di ingresso. Se sono presenti differenti moduli di dati consistenti di ingresso, è necessario utilizzare una chiamata alla funzione SFC14 per ogni modulo che si vuole leggere. I parametri che devono essere definiti sono:

- LADDR. Serve per specificare in esadecimale l'indirizzo di partenza del modulo di ingresso, configurato nel tool Configurazione HW come sarà mostrato in seguito
- RET\_VAL. Permette di specificare un eventuale codice di errore nell'esecuzione della funzione di lettura.
- RECORD. Area di destinazione sulla CPU dei dati da leggere. La lunghezza specificata deve corrispondere a quella specificata nel tool Configurazione HW del modulo DP Slave.

### **5.3.2 SFC15 DPWR\_DAT**

La funzione di sistema SF15 viene utilizzata per trasferire dati consistenti da una qualunque area di memoria della CPU S7 (ad esempio area Q o area DB) ad un modulo di uscita, precedentemente configurato. Ogni accesso di scrittura si riferisce ad un modulo di uscita. Se sono presenti differenti moduli di dati consistenti di uscita, è necessario utilizzare una chiamata alla funzione SFC15 per ogni modulo che si vuole scrivere. I parametri che devono essere definiti sono:

- LADDR. Serve per specificare in esadecimale l'indirizzo di partenza del modulo di uscita configurato nel tool Configurazione HW
- RECORD. Area sorgente sulla CPU dei dati da leggere (ad esempio area di tipo DB o area di tipo Q). La lunghezza specificata deve corrispondere a quella specificata nel tool Configurazione HW del modulo di uscita.
- RET\_VAL. Permette di specificare un eventuale codice di errore nell'esecuzione della funzione di scrittura.

Come già detto, il parametro LADDR di entrambe le funzioni DPRD\_DAT e DPWR\_DAT è il puntatore al modulo di ingresso da cui è necessario leggere informazioni da trasferire in altre aree di memoria o il puntatore al modulo di uscita in cui vengono scritti i dati provenienti da altre aree interne di memoria. Il parametro LADDR viene definito insieme ad ogni modulo nel programma Configurazione HW; l'unica avvertenza è convertire tali indirizzi in formato esadecimale.

Nel seguito verrà illustrata la procedura per la definizione dei moduli di I/O nello I-Slave DP e nel Master DP.

### 5.3.3 Definizione dei moduli di I/O nel Master e nello I-Slave DP

Scopo di questa sezione è descrivere come si definiscono i moduli di I/O necessari per lo scambio dati tra Master DP e I-Slave DP. A tale scopo si riprenda il progetto creato al Capitolo 4, nello stesso punto in cui era stato interrotto. Come si ricorderà erano stati definiti: un oggetto **Simatic 400** (Master DP), l'oggetto **Profibus** e l'oggetto **Simatic 300** (già configurato come Slave DP).

I moduli di memoria I/O dovranno essere create prima nello Slave DP e poi nel Master.

In base a quanto fatto nel Capitolo 4, ci dovremmo trovare nel tool Configurazione HW relativamente all'oggetto **Simatic 300**. Se non si è più in questo tool, si parta dal Simatic Manager, si selezioni l'oggetto **Simatic 300** e si faccia doppio click sull'oggetto Hardware che compare a destra. Sarete tornati nel tool Configurazione HW dell'oggetto **Simatic 300**. Dovreste vedere qualcosa di simile a quanto raffigurato in Figura 29.

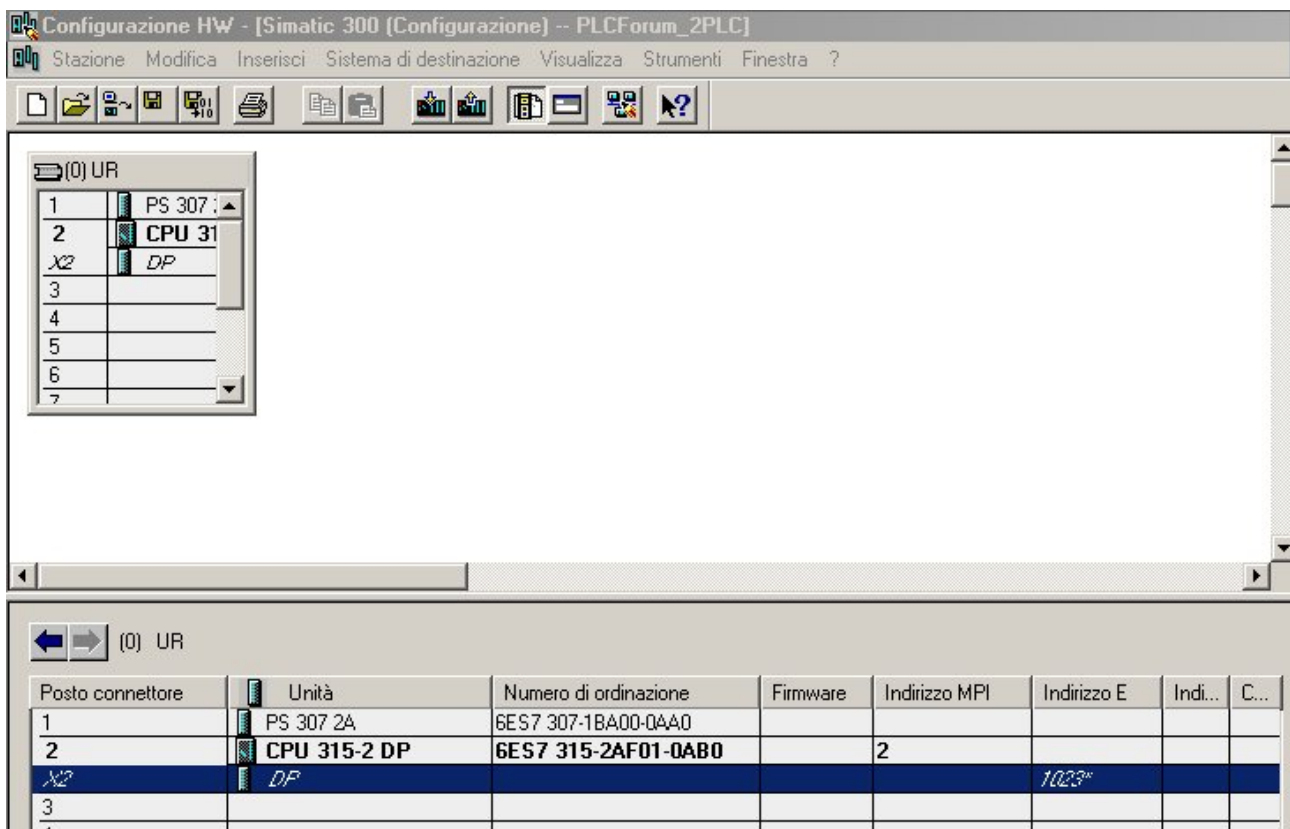


Figura 29

Si definiranno adesso due moduli di 10 byte, uno configurato come ingresso e l'altro come uscita. Si supponrà che entrambi i moduli abbiano, nello I-Slave, indirizzo 1000 e siano interamente consistenti (è possibile infatti definire anche aree parzialmente consistenti).

Si apra la finestra **Proprietà-DP** (facendo doppio click sulla riga relativa all'interfaccia DP) si selezioni la tabella **Configurazione** e si prema il tasto **Nuova**. Apparirà una tabella che potrà essere configurata solo nella porzione Slave (a destra).

Si definisca prima il modulo di ingresso, settando il campo **Tipo di Indirizzo** a Ingresso, e il campo **Indirizzo** a 1000. In basso a sinistra si setti la **Lunghezza** a 10, l'unità a Byte e la Coerenza a Tutto. Dovrebbe comparire la finestra configurata come mostrato in Figura 30.

Proprietà - DP - (R0/S2.1) - Configurazione - Riga 1

Modo: MS (Configurazione master-slave)

Partner DP: master	Partner locale: slave
Indirizzo DP:	Indirizzo DP: 3
Nome:	Nome: DP
Tipo indirizzo:	Tipo indirizzo: Ingresso
Indirizzo:	Indirizzo: 1000
"Posto connettore":	"Posto connettore":
Immagine di processo:	Immagine di processo: IP 0B1
OB di allarme:	Indirizzo di diagnostica:

Lunghezza: 10	Commento:
Unità: Byte	
Coerenza: Tutto	

OK      Applica      Annulla      ?

Figura 30

Si confermano le modifiche appena fatte e si preme il tasto OK. Nella tabella configurazione apparirà il modulo di dati appena definito. Si ripeta la stessa procedura per la definizione del modulo di uscita. Alla fine i due moduli di dati verranno visualizzati come in Figura 31.

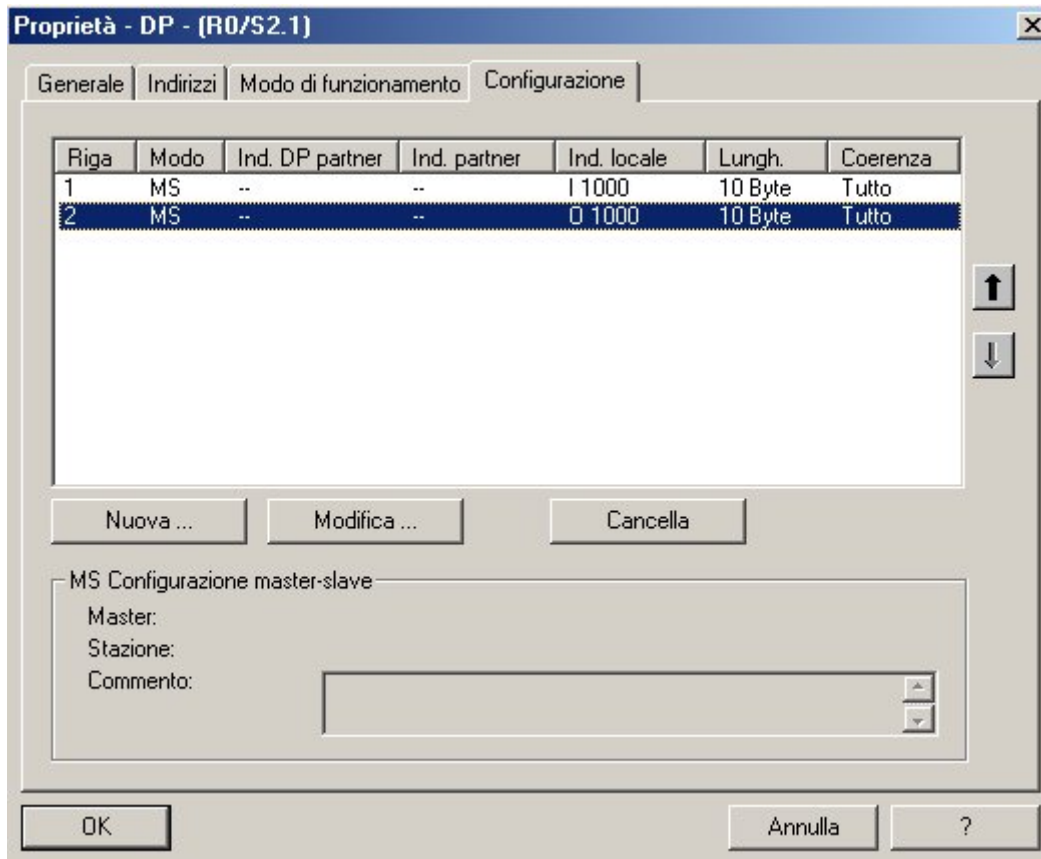


Figura 31

Si chiuda la finestra **Proprietà-DP** (premendo il tasto Ok) e si torni nel tool **Configurazione HW**. Si chiuda il tool **Configurazione HW** relativo allo Slave Simatic 300 e si salvino le modifiche appena introdotte. Siete ritornati nel SIMATIC Manager.

A questo punto è necessario creare un collegamento tra il Master DP e lo I-Slave appena definito, in quanto è necessario mettere in relazione le aree dati consistenti definite nello Slave DP, con quelle corrispondenti che si dovranno definire nel Master.

In SIMATIC Manager si selezioni l'oggetto **Simatic 400** (il Master DP) e si faccia doppio click sull'oggetto Hardware, in modo da attivare il programma Configurazione HW relativo al Master. Nella struttura ad albero visualizzata nella porzione destra, si individui la cartella "**Stazioni già progettate**" all'interno della cartella Profibus-DP. Visto che abbiamo precedentemente configurato una stazione 300, si selezioni il simbolo CPU 31x all'interno della cartella **Stazioni già progettate** e lo si trascini sul simbolo del Profibus DP in alto a sinistra (ossia sulla linea che rappresenta il sistema di comunicazione, ossia l'oggetto Profibus precedentemente creato). Comparirà la finestra mostrata in Figura 32.

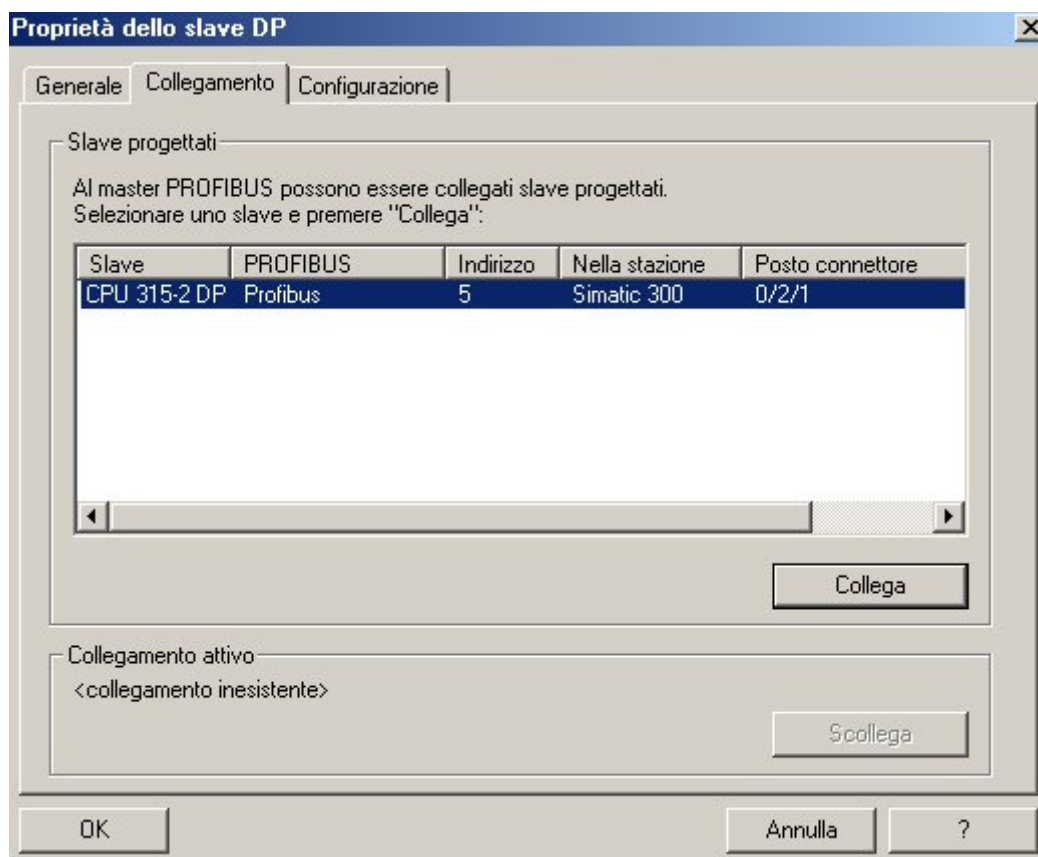


Figura 32

Si preme il tasto **Collega**, mostrato in Figura 32, per creare il collegamento logico tra Master e Slave, necessario per mettere in relazione le aree di memoria nei due dispositivi.

Appena premuto il tasto, nel campo Collegamento attivo (mostrato in Figura 32) verrà riassunto il collegamento appena effettuato. A questo punto devono essere messe in relazione le aree di memoria nei due dispositivi. A tal fine, nella finestra **Proprietà dello Slave DP**, si selezioni la tabella **Configurazione**. Come si vede compariranno i moduli di dati definiti per lo Slave. Si selezioni uno dei due moduli (ad esempio il modulo di ingresso per lo Slave) e si faccia doppio click (oppure si preme il tasto Modifica). Comparirà una finestra che permetta la definizione dell'area di memoria nel Master DP, corrispondente a ciascun modulo dati consistente definito nello I-Slave. In particolare al modulo di ingresso dello Slave dovrà corrispondere un modulo di uscita nel Master e viceversa. Dunque al modulo di ingresso nello Slave, dovrà corrispondere un modulo di uscita nel Master sempre di 10 byte e tutta consistente. Come indirizzo dell'area si setti anche in questo caso 1000 (i due indirizzi possono ovviamente anche non coincidere, visto che i due moduli dati corrispondenti possono occupare posizioni di memoria diverse nel Master e nello Slave).

Si dovrebbe ottenere quanto mostrato in Figura 33.



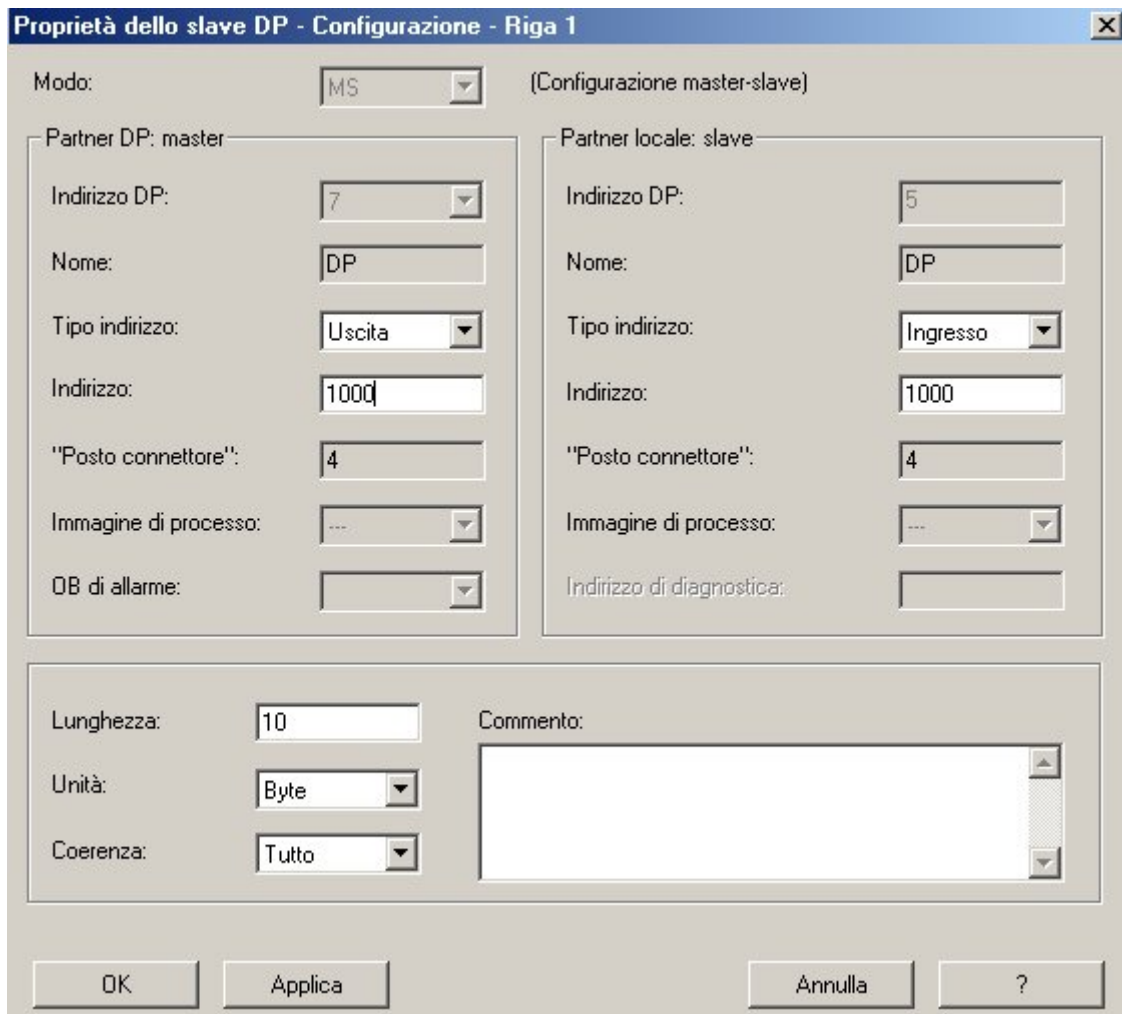


Figura 33

Si ripeta la stessa procedura per il modulo di uscita dello Slave al quale dovrà corrispondere il modulo di ingresso nel Master.

Una volta definite le due aree di memoria nel Master, dovrebbe comparire la finestra mostrata in Figura 34.

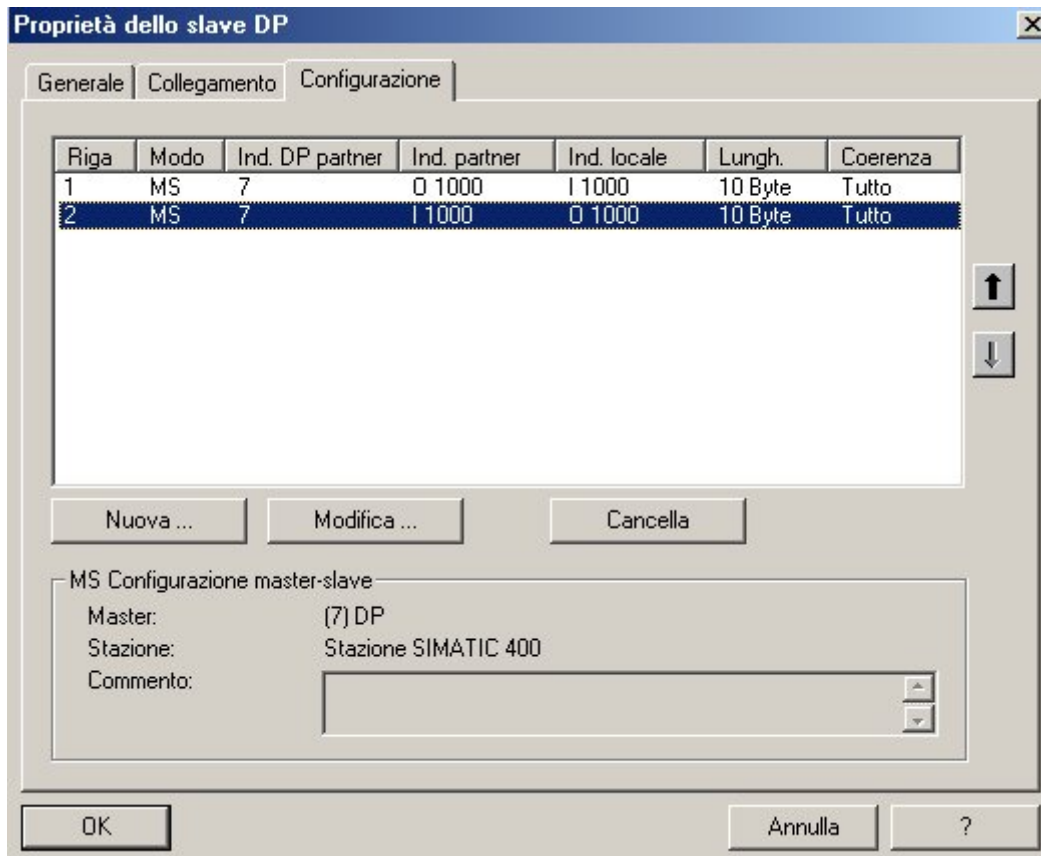


Figura 34

Come si vede il modulo di area di memoria di 10 byte di ingresso nello I-Slave (Ind.locale) viene fatta corrispondere al modulo di area di memoria di 10 byte di uscita nel Master (Ind.partner, di indirizzo DP 7, supposto nell'esempio); entrambe le aree sono consistenti. Analogamente l'area di memoria di 10 byte di uscita nello Slave (Ind.locale) viene fatta corrispondere all'area di memoria di 10 byte di ingresso nel Master.

Adesso si confermi con Ok e si torni al tool Configurazione HW.

A questo punto la configurazione del Master e dello Slave in termini di moduli dati per lo scambio informativo consistente è completata ed è possibile procedere alla scrittura di un programma (sia sul Master sia sullo Slave) per realizzare lo scambio dati; si veda il Capitolo 6 a tal proposito.

Ovviamente lo scambio dati tra i corrispondenti moduli del Master DP e dello I-Slave DP avverrà in automatico tramite il DataExchange del Profibus DP.

## 6 Scrittura di un programma basato sulla comunicazione Profibus DP

Scopo del capitolo è fornire alcuni esempi della definizione di un programma che utilizza il sistema di comunicazione Profibus DP per lo scambio dati. In particolare verrà prima fornito un esempio relativo allo scambio dati tra un Master DP e uno Slave DP costituito da un modulo di I/O remoto, basato sull'utilizzo di meccanismi di accesso diretto I/O (ossia basati sull'immagine di processo di I/O). Poi verrà fornito un esempio di programma che prevede lo scambio informativo tra un Master DP e uno I-Slave DP (PLC), che fa uso dello scambio dati consistente.

### 6.1 Scambio dati con accesso diretto I/O

La scrittura di un programma che fa uso della comunicazione Profibus DP basata su uno scambio dati con accesso all'immagine di processo di I/O, è assai semplice in quanto è identica a quella adottata per la gestione delle unità di ingresso/uscita locali. Tale tipo di scambio informativo è generalmente confinato agli Slave costituiti da moduli di I/O remoto ai quali, in fase di configurazione (come visto nei capitoli precedenti), vengono assegnati degli indirizzi appartenenti all'area di immagine di processo. Dunque la gestione degli Slave avviene semplicemente utilizzando gli indirizzi assegnati in fase di configurazione.

Si faccia riferimento all'esempio presentato nel paragrafo 4.5, in cui è stata configurata una rete profibus DP alla quale sono connessi un Master DP *Simatic 400* (CPU 414-2DP), uno Slave DP RIO 16I e uno Slave DP RIO 16O.

Adesso si supponga di voler generare una uscita che oscilla periodicamente assumendo i valori 0 e 1, realizzando in tal modo una forma d'onda quadra. Si supponga sia  $T$  il periodo in msec del segnale e siano  $T/2$  le durate assunte dai valori alto e basso. Si supponga che l'uscita che si desidera attivare sia il primo bit del modulo remoto RIO 16O, al quale è stato assegnato l'indirizzo di default Q0.0. Si supponga infine che la generazione del segnale sia controllata dal quinto bit del modulo remoto RIO 16I, nel senso che un valore alto di tale bit abilita la generazione del segnale di onda quadra.

Si vada nella cartella blocchi dell'oggetto *Simatic 400* e si faccia doppio click. Si selezioni l'oggetto OB1 sulla destra e si faccia doppio click aprendo l'editor. Si supponga di scrivere il programma in linguaggio ladder.

La Figura 35 mostra il programma realizzato. La variabile start è una variabile appartenente all'area di memoria (ad esempio M20.0), che serve per realizzare l'andamento periodico del segnale. La variabile start deve essere inizializzata a FALSA (0). I timer T\_ON (l'uscita del timer assume il valore dell'ingresso dopo il tempo di ritardo associato al timer) hanno una durata di  $T/2$ .

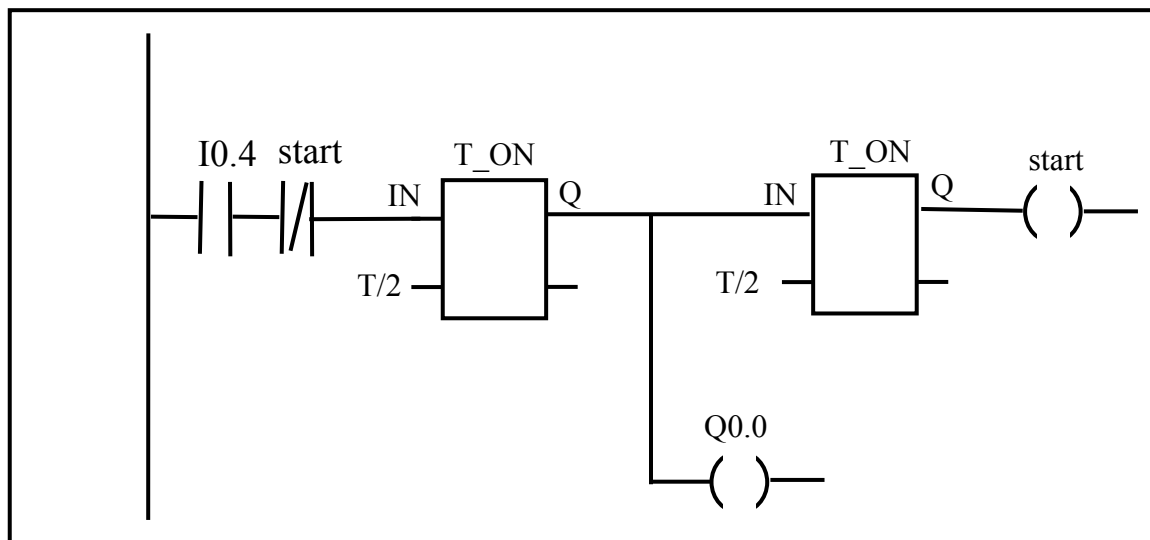


Figura 35

Si chiuda l'editor per la creazione del programma. Si selezioni la cartella blocchi e la si carichi nel PLC, che deve essere in STOP. Adesso si porti il PLC in Run e se i led SF DP e BUSF non sono accessi, il sistema è già funzionante, ossia il Master DP ha il controllo della rete Profibus. Appena l'ingresso remoto I0.4, corrispondente al quinto bit del primo byte del modulo RIO 16O viene attivato, l'uscita Q0.0 corrispondente al primo bit del modulo RIO 16O inizierà a lampeggiare con periodo T.

## 6.2 Scambio dati consistenti

Scopo di questa sezione è quello di mostrare un esempio dell'uso delle funzioni SFC14 e SFC15, già descritte in altre sezioni precedenti. Le funzioni verranno utilizzate per realizzare lo scambio informativo tra un DP Master e un DP I-Slave, realizzato da un PLC.

La Figura 36.a mette in evidenza che lo scambio dati tra DP Master e DP Slave (ossia dati che transitano dal Master verso lo Slave, nel senso che il Master vuole scrivere dati sullo Slave) deve avvenire utilizzando SFC15 da parte del Master per scrivere dati in uscita sul modulo di uscita e da parte del DP Slave utilizzando la SFC14 per leggere tali dati dal corrispondente modulo degli ingressi, dati che verranno trasferiti dalla SFC14 in un'area di memoria interna al I-Slave.

Ovviamente l'inverso capita nel caso di scambio informativo al contrario (ossia dal DP Slave verso il DP Master, nel senso che il Master intende leggere dati dallo Slave), come mostrato in Figura 36.a. Nel lato I-Slave, viene usata la funzione SFC15 per scrivere le uscite dello Slave da un'area di memoria interna al modulo delle uscite dello I-Slave. Lato Master-DP, tali uscite verranno ricevute nel modulo di ingressi corrispondente e verranno trasferite in un'area di memoria interna tramite la SFC14.

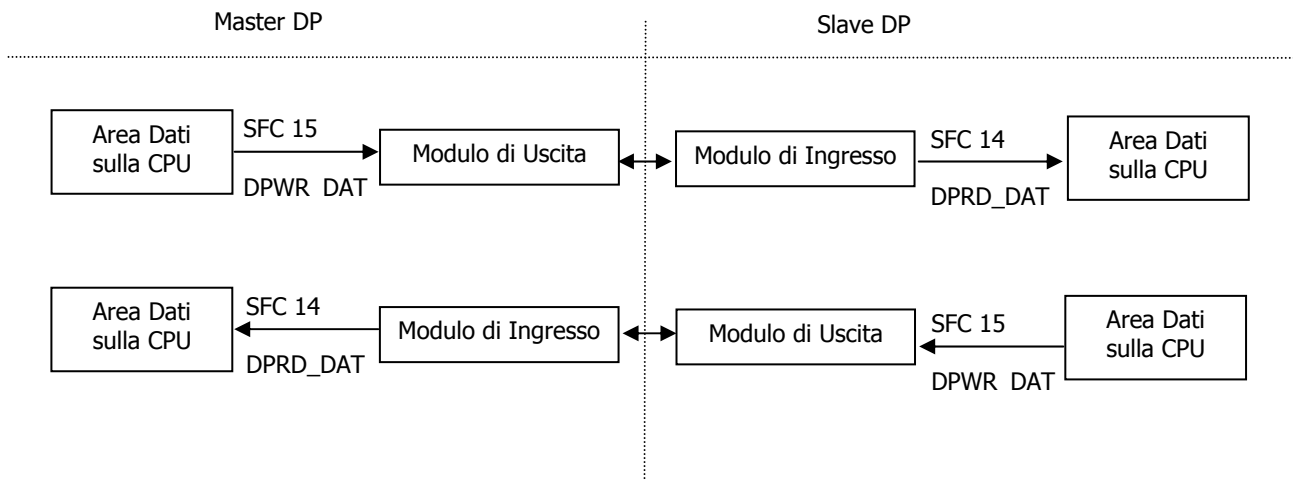


Figura 36.a

Allo scopo di fornire un esempio realistico ed interessante verrà supposto di adottare una configurazione hardware composta da un PLC S7-400 come DP Master (in particolare una CPU 416-2DP) e sempre da un PLC, ma della serie S7-300 (315-2DP), come DP I Slave.

In particolare verrà adottata la stessa configurazione mostrata nel paragrafo 4.5. Inoltre verrà utilizzata la stessa definizione delle aree consistenti illustrata nel capitolo 5; come si ricorderà sia nel Master sia nello Slave DP erano state definiti due moduli di aree consistenti, rispettivamente di Ingresso e di Uscita, di 10 bytes ciascuna, con indirizzi 1000 per entrambi (sia sul Master sia sullo Slave) e con parametro **Consistenza** settato a "Totale".

Si supponrà di voler sviluppare uno scambio dati tra Master e Slave mostrato in Figura 36.b. Il Master DP vuole scrivere sullo Slave DP 10 byte conservati nell'area DB20 (tali dati vengono visti come ARRAY[1..10] of BYTE); si supponga che tali bytes devono essere scritti nell'area I dell'I-Slave (I100-I109). Dunque è necessario che avvenga il seguente flusso:

- Il Master DP deve scrivere i 10 bytes nel suo modulo di uscite tramite la funzione SFC15
- L'I-Slave DP deve leggere i 10 byte presenti nel suo modulo di ingresso, tramite la SFC14 e trasferirli nell'area I100..I109

Viceversa si voglia che il Master DP legga le uscite dell'I-Slave poste in Q100..Q109 e li voglia memorizzare internamente nell'area DB10 (contenente un ARRAY[1..10] of BYTE). Dunque è necessario che avvenga il seguente flusso:

- L'I-Slave DP deve scrivere i dati Q100..Q109 nel suo modulo di uscita tramite la SFC15
- Il Master DP deve leggere i 10 bytes da suo modulo di ingresso corrispondente tramite la funzione SFC14, che trasferirà questi dati nell'area DB10

Si vuole altresì che se il Master invia al DP Slave una sequenza di 10 byte dove il primo byte ad esempio assume il valore esadecimale "11", il Master riceverà immediatamente un sequenza di 10 byte, dove il primo byte assumerà esattamente il valore esadecimale "11". Ossia si vuole che il primo byte I100 ricevuto dallo Slave venga copiato in Q100 e rimandato al Master.

L'esempio di programmazione verrà mostrato separando la descrizione per il DP Slave e il DP Master.

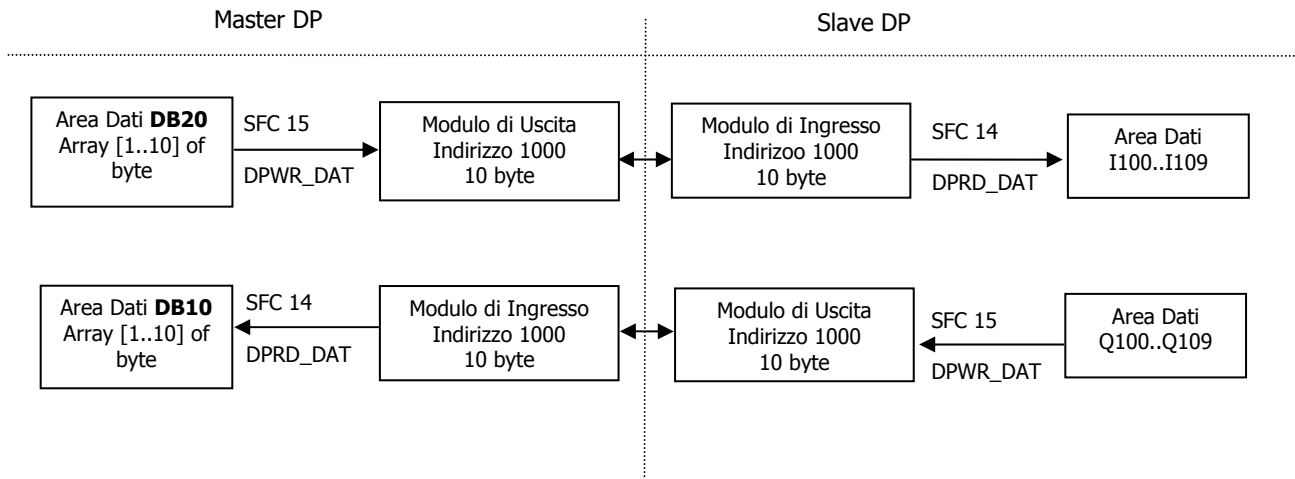


Figura 36.b

### 6.2.1 Programma utente per il DP Slave S7-300 con CPU315-2DP

Si supponerà di scrivere un programma in linguaggio Ladder sul DP Slave all'interno del OB1. Scopo del programma è fare in modo che il DP Slave legga i 10 byte di dati consistenti inviati dal DP Master, rinviandone altri 10, dove il primo byte è lo stesso primo byte ricevuto dal DP Master.

Verrà supposto di leggere i 10 bytes ricevuti dal Master tramite la funzione DPRD\_DAT, trasferendoli dal modulo di ingresso dell'I-Slave nell'area di immagine di processo degli ingressi a partire dall'indirizzo IB100; il DP Slave copia il primo byte IB100, nell'area di immagine di processo delle uscite all'indirizzo QB100. Infine il DP Slave scrive i 10 bytes consistenti a partire dal QB100 sul suo modulo delle uscite tramite la funzione DPWR\_DAT.

Per scrivere il programma si proceda come segue:

- Si apra il progetto contenente la configurazione hardware dei due PLC (Simatic 400, Master DP, e Simatic 300, I Slave DP) e la configurazione Profibus (il nome che avevamo assegnato era DuePLC)
- Si selezioni l'oggetto Simatic 300 e si faccia doppio click su esso
- Si scenda nel sottoalbero che si aprirà fino a giungere alla cartella Blocchi
- Si selezioni la directory Blocchi
- Nella porzione destra del Simatic Manager, comparirà il simbolo OB1
- Si faccia click 2 volte sull'oggetto OB1
- Si selezioni il linguaggio KOP
- Si scriva il codice mostrato in Figura 37. Al fine di inserire le chiamate alle funzioni DPRD\_DAT e DPWR\_DAT, si faccia attenzione a configurare i parametri formali (LADDR, RET\_VAL e RECORD) come mostrato in Figura 37. Come si vede il codice scritto in Ladder fa uso della funzione MOVE per copiare il byte IB100 nel byte QB100. Si noti che LADDR vale W#16#3E8, che corrisponde all'indirizzo del modulo di ingresso 1000 nella funzione DPRD\_DAT e all'indirizzo del modulo di uscita 1000 nella funzione DPWR\_DAT; ciò in accordo alla definizione delle aree consistenti fatta nel capitolo 5. Si suppone poi di utilizzare le aree MW200 e MW202 nel DP Slave per memorizzare eventuali valori di ritorno. Si noti infine che i valori dei campi RECORD sono 10 bytes a partire dall'indirizzo IB100 per la funzione DPRD\_DAT e 10 bytes a partire da QB100 per la funzione DPWR\_DAT.

- Si salvi il programma appena creato, si chiuda l'editor dell'OB1 e si torni in Simatic Manager. Come si vede, la directory blocchi dell'oggetto SIMATIC 300 contiene gli oggetti SFC14 e SFC15 oltre che OB1.
- E' necessario a questo punto creare due Blocchi Organizzativo (OB) allo scopo di evitare il blocco della CPU dello I Slave. Esistono particolari blocchi organizzativi che servono a gestire eventi straordinari, come errori derivanti da risultati di diagnostica e guasti. Nel caso in cui si verificano tali eventi il sistema operativo del PLC Simatic richiama opportuni blocchi organizzativi che vengono programmati per gestire l'evento occorso. In particolare, esiste il blocco OB 82 che serve per gestire errori derivanti da operazioni di diagnostica, e il OB 86 che gestisce eventuali errori di funzionamento del rack. Se occorrono tali eventi e se il sistema operativo non trova questi OB, la CPU passa da RUN a STOP. Per prevenire tale situazione, evitando che la CPU dello I Slave si blocchi, devono essere creati a questo punto del progetto i blocchi OB 82 e OB 86. Non importa se essi non contengano codice, l'importante è che ci siano. Tali blocco devono essere inseriti nella cartella Blocchi della stazione Simatic 300. Si selezioni a tal fine la cartella Blocchi e si prema il pulsante destro del mouse, facendo comparire il menu di opzioni. Si scelga l'inserimento di un nuovo oggetto e in particolare di un blocco organizzativo. Si inserisca OB82 nel nome dell'oggetto e si schiacci il pulsante OK, per la creazione del blocco. Analogamente si faccia per l'OB 86.
- Si scarichino tutti i blocchi della cartella blocchi della stazione Simatic 300 nella CPU 3152-DP, che deve trovarsi in STOP.
- Si porti in RUN la CPU. Si dovrebbe vedere il led SF DP accendersi e il led BUSF lampeggiare (dovuto al fatto che ancora il Master è in STOP).

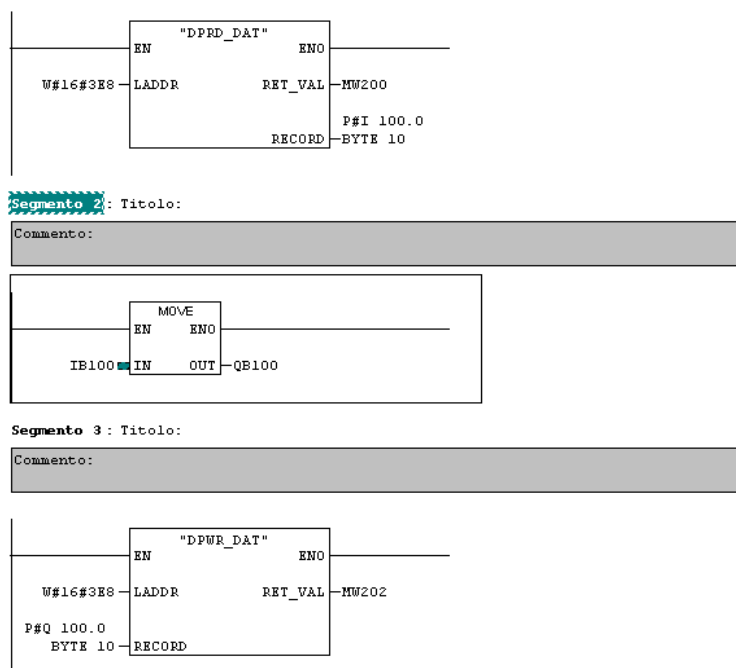


Figura 37

## 6.2.2 Programma utente per il DP Master S7-300 con CPU416-2DP

Il programma nel DP Master S7-400 è il duale di quello appena descritto. Ovviamente si dovranno specificare le due chiamate alle funzioni SFC15 e SFC14 nell'ordine, in quanto il DP Master invia i 10 bytes al DP Slave (tramite una DPWR\_DAT sul suo modulo di uscita) e riceve 10 bytes tramite una DPRD\_DAT dal modulo di ingresso. Come fatto per il caso precedente si sceglierà il linguaggio KOP.

In questo esempio, a differenza del programma sullo Slave, si suppone di utilizzare i blocchi dati contigui di lunghezza 10 byte ciascuno con indirizzi di partenza DB10 e DB20 per la memorizzazione per i dati di ingresso e di uscita dell'I Slave (si ricorda che nel caso dello Slave si erano utilizzate le aree di immagine di processo I e O, con indirizzi di partenza I100 e Q100). In particolare, verrà supposto che i 10 byte contigui a partire dal byte DB20 servono al Master per scrivere sul modulo di uscite del Master DP, mentre i 10 byte contigui a partire dal byte DB10 servono al Master per trasferire i dati dal suo modulo di ingressi.

E' chiaro che durante la scrittura del programma si dovrà procedere alla definizione delle due aree dati caratterizzate da indirizzi di partenza DB10 e DB20 e di lunghezza 10 byte ciascuno, come verrà illustrato nel seguito.

Per scrivere il programma si proceda come segue:

- Si apra il progetto DuePLC, se non era già aperto
- Si selezioni l'oggetto Simatic 400 e si faccia doppio click su esso
- Si percorra l'intero sottoalbero contenuto nella cartella Simatic 400 fino a giungere nella cartella Blocchi
- Si selezioni la directory Blocchi, in modo che nella porzione destra del Simatic Manager, compaia il simbolo OB1
- Si faccia 2 volte click sull'oggetto OB1
- Si selezioni il linguaggio KOP
- Si scriva il codice mostrato in Figura 38. Ovviamente sono presenti solo le due chiamate alle funzioni SFC15 e SFC14 nell'ordine, in quanto il DP Master invia i 10 bytes al DP Slave e riceve 10 bytes dal DP Slave. Come già detto precedentemente, ogni funzione SFC14 e



SFC15, ha dei parametri formali (LADDR, RET\_VAL e RECORD) che si dovranno configurare come mostrato in Figura 38.

- Si salvi il programma appena creato, si chiuda l'editor dell'OB1 e si torni in Simatic Manager.
- Come fatto per lo I-Slave, è necessario a questo punto creare i due Blocchi Organizzativi OB82 e OB86 allo scopo di evitare blocco della CPU del Master. Come già detto, il blocco organizzativo OB82 (interrupt diagnostico) e il blocco OB86 (rack failure) vengono richiamati dal sistema operativo del Master in corrispondenza di errori derivanti da diagnostica e da guasti. Nel caso in cui il blocco non sia stato definito, lo stato della CPU del Master DP passa da RUN a STOP. Allo scopo di prevenire tale situazione ed evitare che la CPU del Master si blocchi, devono essere creati i due blocchi sopracitati. Si proceda come fatto per lo I-Slave. Anche in questo caso non è importante specificare il codice all'interno dei due blocchi organizzativi OB82 e OB86.
- A questo punto del progetto è necessario creare le due aree di memoria (Data Block - DB, blocchi dati di memoria interna) in cui immagazzinare i dati letti e scritti dallo e sullo I-Slave. Si ricordi che si è supposto di memorizzare i dati verso e dallo Slave in memoria interna e non nell'area immagine di processo del Master, come fatto per lo Slave. Bisogna selezionare l'area dati interna, ossia definire due blocchi di 10 bytes ciascuno. Come detto si è scelto di fissare gli indirizzi di partenza a DB10 e DB20. Per far ciò si proceda così:
  - Si selezioni la cartella Simatic 400 e si selezioni la cartella blocchi in essa contenuta
  - Con il pulsante di destra si faccia comparire il menù e si scelga **Inserisci nuovo oggetto**, e si scelga l'opzione Blocco Dati
  - Si aprirà la finestra **Proprietà-Blocco Dati** composta da più tabelle. Nella tabella **Generale-Parte 1**, si specifichi DB10 nel campo **Nome**. Si faccia Ok.
  - Nella cartella blocchi comparirà anche l'oggetto DB10
  - Si selezioni l'oggetto DB10 e si faccia doppio click su di esso, in modo si possa aprire l'editor dei blocchi dati.
  - Nel campo **Nome** si metta un nome, ad esempio **InputData (che compare anche nella Figura 38)**
  - Nel campo **Tipo** si specifichi che si tratta di un Array (vettore) di 10 byte consecutivi. La sintassi standard per la definizione di un vettore di 10 elementi è ARRAY[1..10]
  - Si faccia click nella riga successiva (sotto quella dove abbiamo specificato il nome e ARRAY, in modo da aggiungere un'altra riga. Nel campo **Tipo** di questa riga, bisogna scrivere il tipo Byte, in modo da completare la definizione di un ARRAY di 10 elementi di Byte. Dovrà apparire come mostrato in Figura 39.
  - Si salvi e si chiuda l'editor di testo, ritornando al Simatic Manager
  - Si ripeta la stessa procedura per creare il blocco dati DB20 (si potrà ad esempio assegnare il nome **OutputData, come mostrato in Figura 38**). Si dovrà ottenere la stessa definizione mostrata in Figura 40.
- Si scarichino tutti i blocchi della cartella blocchi della stazione Simatic 400 nella CPU 416-2DP, che deve trovarsi in STOP.
- Si porti in RUN la CPU. I led SF DP e led BUSF dovrebbero spegnersi; se ciò accade la comunicazione tra Master e Slave avviene correttamente.

Per testare lo scambio informativo, una volta caricati i programmi nelle due CPU, si connetta il PC in cui gira il Programma PG/PC Siemens con il PLC S7-400 (Master DP).

Si scelga l'opzione **Visualizza→Online**. Si selezioni l'oggetto CPU 416-2 DP e si prema il tasto destro del mouse, in modo da far comparire il menù. Si scelga l'opzione **Sistema di destinazione→Controlla e comanda variabili**.

Si specifichino nel campo **Operando** le due variabili DB20.DBB0 (per visualizzare il primo byte ricevuto dal DP Slave) e DB10.DBB0 (per monitorare il primo byte inviato al DP Slave). Si dovrebbe ottenere quanto mostrato in Figura 41.

Si monitorino le due variabili tramite il comando **Variabile→Controlla**; le due variabili assumeranno il valore B#16#00 (valore 0). Adesso si setti il valore da inviare al DP Slave (valore della variabile DB20.DBB0); si scelga ad esempio il valore B#16#11 e si scelga l'opzione **Variabile→Attiva valori di comando**. Si noterà che entrambe le variabili cambieranno di valore, in quanto il DP Master invia il valore B#16#11 al DP Slave che lo rimanda indietro al DP Master. Ciò dimostrerà il corretto funzionamento dei programmi realizzati.

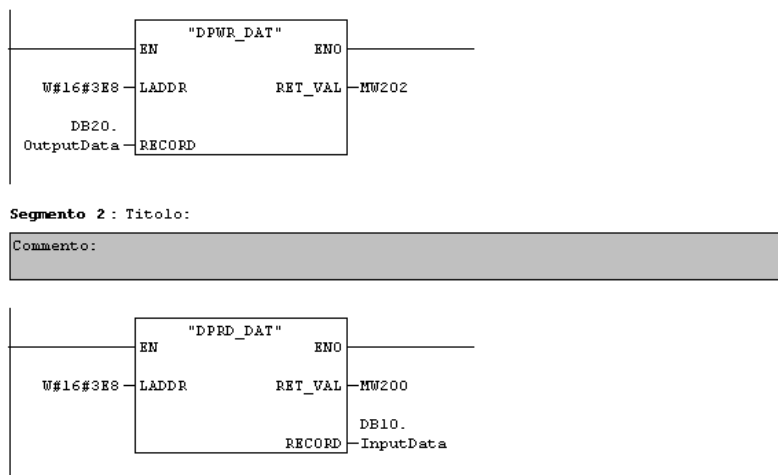


Figura 38

Indirizzo	Nome	Tipo	Valore iniziale	Commento
0.0		STRUCT		
+0.0	InputData	ARRAY[1..10]		
*1.0		BYTE		
=10.0		END_STRUCT		

Figura 39

Indirizzo	Nome	Tipo	Valore iniziale	Commento
0.0		STRUCT		
+0.0	OutputData	ARRAY[1..10]		
*1.0		BYTE		
=10.0		END_STRUCT		

Figura 40

Operando	Simbolo	Formato di visualizzazione	Valore di stato	Valore di comando
DB10.DBB 0		HEX		
DB10.DBB 9		HEX		
DB20.DBB 0		HEX		B#16#11
DB20.DBB 9		HEX		

Figura 41

## 6.3 Scambio dati diretto (DX)

Con scambio dati diretto si intende la comunicazione tra due stazioni CPU progettate come slave DP, che vengono gestite dallo stesso master DP. Nel progetto di esempio che segue è stata realizzata la seguente configurazione.

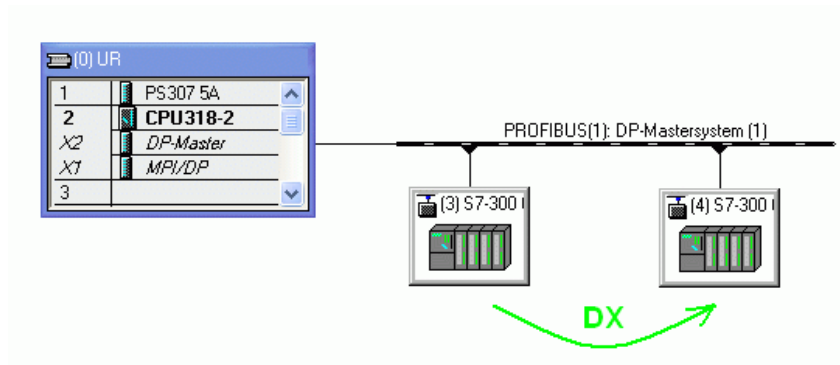


Figura 42 Configurazione del progetto di esempio

Dopo aver incorporato le tre stazioni CPU nel progetto STEP7 e aver accoppiato le due stazioni slave al master, procedere nel seguente modo per la progettazione della scambio dati diretto.

1. Passare nella HW Config del master DP.
2. Nella prima stazione slave progettare almeno una riga come scambio diretto per la comunicazione master-slave.

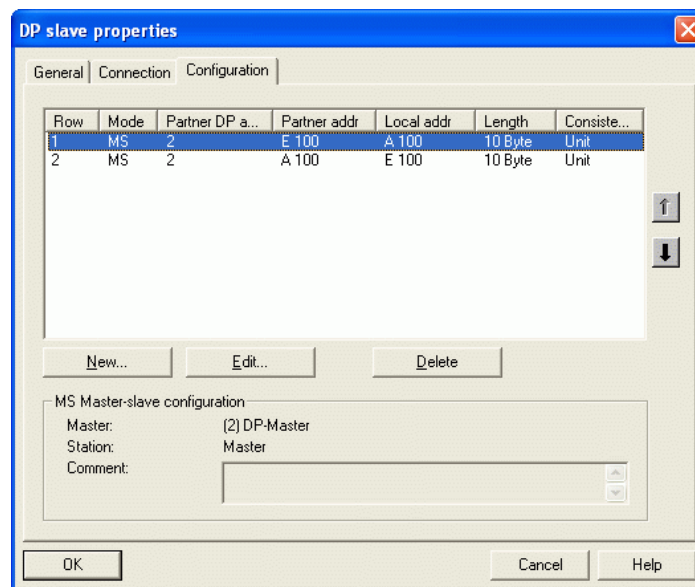


Figura 43 Configurazione della CPU 315-2DP con indirizzo DP 3

3. Almeno uno slot dovrebbe essere progettato con dati di ingresso sul master e dati di uscita sullo slave. Questi dati vengono poi "ascoltati in parallelo" dal secondo slave DP.

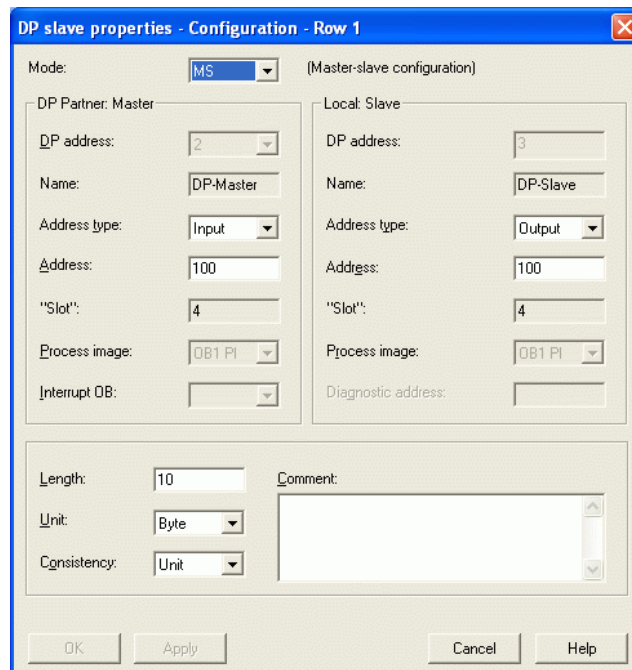


Figura 44 Configurazione master-slave

4. Passare nella configurazione del secondo slave DP nella HW Config del master DP.
5. Se qui si introduce una nuova riga, nella modalità della scambio dati vengono proposte due possibilità. Scegliere la modalità **DX** per lo scambio dati diretto ed indicare il parametro.

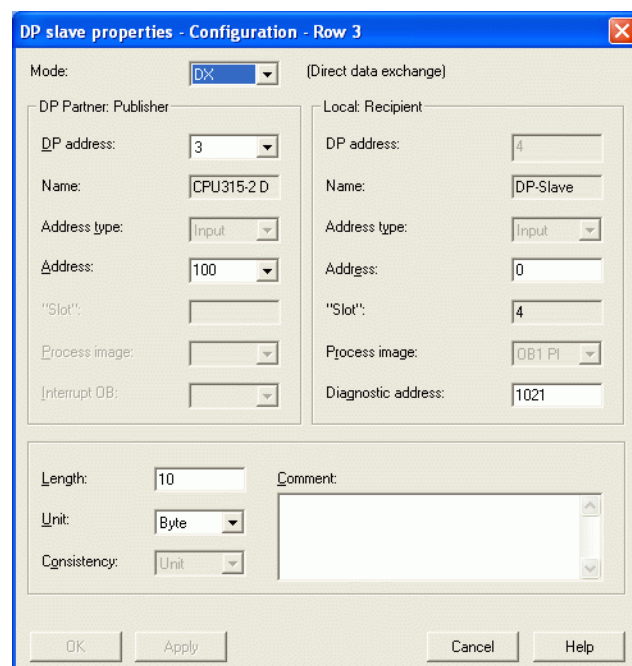


Figura 45 Configurazione della CPU 315-2DP con l'indirizzo DP 4

6. In questo modo il secondo slave DP può "ascoltare in parallelo" i dati che il primo slave DP invia al master DP.