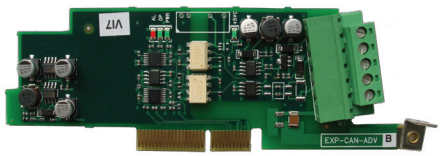
Manuale istruzione

EXP-CAN/DN-ADV

Scheda espansione interfaccia CANopen / DeviceNet



Sommario

Attenzione!

Utilizzare solo le viti in dotazione!

Isolamento rinforzato

PELV (Protective Extra Low Voltage) EN 61800-5-1.

Introduzione

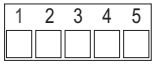
Il presente manuale descrive la scheda opzionale EXP-CAN/DN-ADV per il collegamento dei Drive serie ADV200 alle reti CANopen o DeviceNet
E' possibile montare una sola scheda bus di campo per Drive.
Il manuale si rivolge a progettisti e tecnici di manutenzione e messa in servizio di sistemi CANopen e DeviceNet. È necessaria dunque una conoscenza di base di CANopen e DeviceNet, vedere i seguenti manuali:

- CANopen CAL-Base COMMUNICATION PROFILE for Industrial Systems; CiA Draft Standard 301 Version 4.2 Date 13 February 2002 by CAN in Automation e. V.
- DeviceNet Specifications. Volume 1 - DeviceNet Communication Model and Protocol (Pubblicato da ODVA).
- DeviceNet Specifications. Volume 2 - DeviceNet Device Profiles and Object Library (Pubblicato da ODVA).

Fissaggio

Fare riferimento al capitolo "**Installazione schede opzionali**" del manuale ADV200 Guida rapida: **la scheda deve essere inserita nello slot 3**.

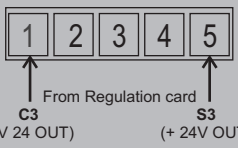
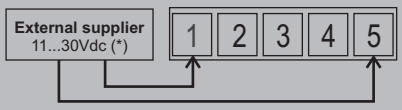
Connessione



Sezione dei cavi: 0.2 ... 2.5 mm^2 (AWG 24 ... 12)

Per il collegamento al Bus deve essere usato un doppino schermato (**del tipo indicato dalla specifica CANopen o DeviceNet**) che deve essere posato separato dai cavi di potenza, con una distanza minima di 20 cm. La schermatura del cavo deve essere continua e lo schermo connesso a massa in un solo punto.

Deve altresì essere assicurato il collegamento equipotenziale dei nodi CAN bus ADV200 tramite il morsetto 1 (V- / CAN_GND) della scheda.

PIN	DeviceNet	CANopen	Funzione	Max
			Morsetto BUS: consente di collegare la scheda alla rete CANopen o DeviceNet	
			INTERNAL SUPPLY (Connection not insulated)	
				
			EXTERNAL SUPPLY (Connection optocoupled)	
				
			(*) L'alimentatore deve essere dimensionato in accordo alle specifiche del bus utilizzato (CANopen o DeviceNet). La scheda assorbe 30 mA@24V.	
1	**V-**	**CAN_GND**	Terra / 0V / V-	**0V**
2	**CAN_L**	**CAN_L**	Linea bus CAN_L (dominante bassa)	**-**
3	**DRAIN**	**CAN_ SHLD**	Schermatura CAN -	**-**
4	**CAN_H**	**CAN_H**	Linea bus CAN_H (dominante alta)	**-**
5	**V+**	**CAN_V+**	Alimentazione positiva esterna CAN (dedicata per alimentare transceiver e optocoupler)	**11 ... 30V**

Importante! **Nota sulle terminazioni :**

Il primo e l'ultimo nodo della linea CAN devono avere una resistenza da 120 ohm tra i pin 2 e 4.

Attenzione!

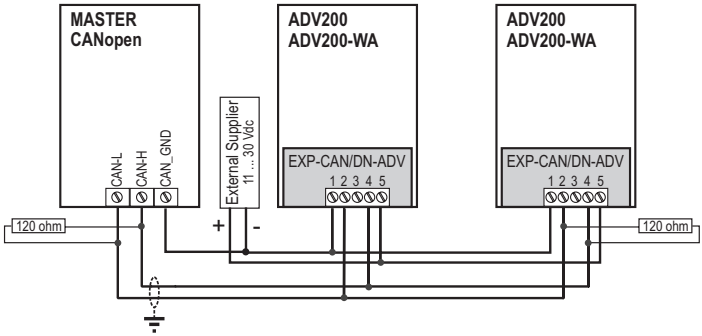
E' necessario alimentare i morsetti 1-5:

- se si utilizza un alimentatore esterno la rete risulta isolata galvanicamente,

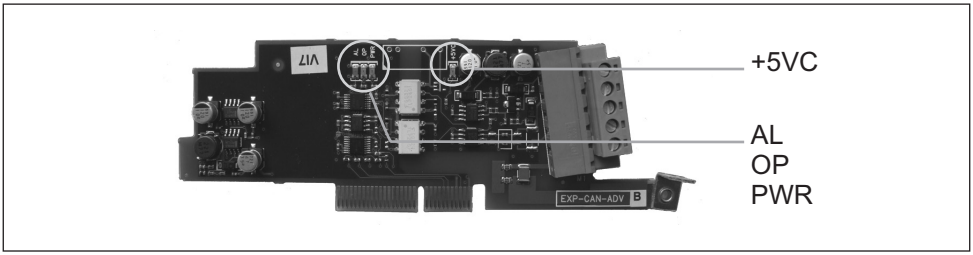
- se si deriva dalla scheda di regolazione (morsetti C3/S3, +24Vout / 0V24 out) la rete NON è isolata galvanicamente. Inoltre n**on deve essere superato** il valore massimo di corrente disponibile (totale max = 150 mA @ 24V), incluse eventuali altre schede di espansione.

- nelle reti CANopen in generale, il collegamento CAN_GND deve essere condiviso tra tutti i nodi partecipanti, a meno che la rete CANopen non sia completamente isolata galvanicamente. Qualora un nodo (master o slave) non disponga della connessione CAN_GND, oppure quest'ultima non venga usata, è compito dell'utilizzatore garantire la massima reiezione ai disturbi di modo comune per tutti i partecipanti alla rete.

La connessione tra le singole schede è realizzata con un cavo schermato, come illustrato nella figura seguente.



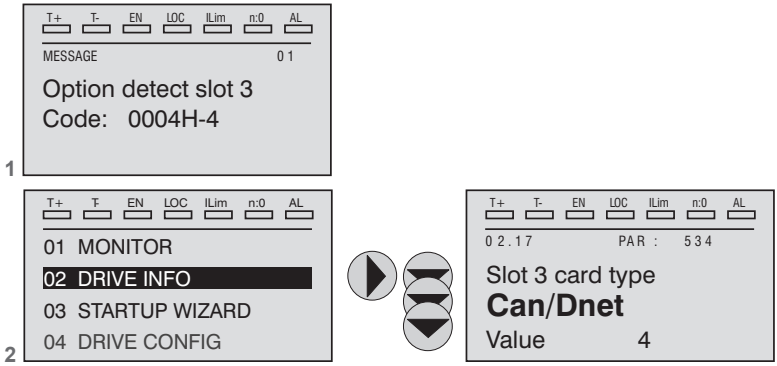
Leds



LEDs		CANopen	DeviceNet
AL	**(rosso)**	ON: l'interfaccia CANopen si trova in allarme	Indicatori dello stato della connessione DeviceNet, vedere tabella seguente.
OP	**(verde)**	ON: l'interfaccia CANopen si trova nello stato Operational	
PWR	**(verde)**	Indica che l'espansione è alimentatata ed attiva	
+5VC	**(verde)**	Indica che il nodo CAN optoisolato è alimentato correttamente.	

OP	AL	Significato
ON	ON	Power-up della scheda
BLINK	BLINK	Self test e Duplicate MAC-ID check in corso
BLINK	OFF	Attesa configurazione del Master e/o Polling I/O non attivo
ON	OFF	Polling I/O attivo, stato operativo
OFF	BLINK	Fault recuperabile (DUP MAC-ID fail, bus-off, bus-loss)
OFF	ON	Fault grave (errore di configurazione, errore interno)
OFF	OFF	DeviceNet non configurato

Riconoscimento della scheda opzionale



1 - All'accensione il drive riconosce la presenza della scheda opzionale nello slot di espansione 3, sul display per alcuni secondi viene visualizzato questo messaggio.

2- Nel menu 02 INFORMAZIONI DRIVE, delezionare il parametro PAR 534 **Tipo scheda slot 3** per leggere il tipo di scheda riconosciuta.

Valore	Descrizione	Tipo scheda
0	Nessuna	-
4	Can/Dnet	EXP-CAN/DN-ADV
255	Sconosciuta	-

1.0 Interfaccia CANopen

CANopen è un profilo di comunicazione per sistemi industriali basato su CAL.
Il documento di riferimento è la specifica CANopen CAL-Base COMMUNICATION PROFILE for Industrial Systems; CiA Draft Standard 301 Version 4.2 Date 13 February 2002 by CAN in Automation e. V.
Il drive implementa anche il profilo DS402 secondo la specifica CANopen Device Profile Drives and Motion Control v4.02
Come protocollo CAN (ISO 11898) viene usato CAN2.0A con l'identificatore a 11 bit.
L'interfaccia integrata CANopen è sviluppata come Minimum Capabilty Device.
Lo scambio di dati avviene ciclicamente; l'unità Master legge i dati messi a disposizione dagli Slaves e scrive i dati di riferimento agli Slaves.

1.1 Funzionalità CANopen

In questo capitolo vengono descritte le funzionalità del profilo di comunicazione CANopen.

Caratteristiche principali:

1) Viene gestito il Minimum Boot-up; il Extended Boot-up (CAL) non è supportato.
2) E' implementata la funzionalità SYNC.
3) L'assegnamento dinamico dei PDO e RTR.
4) Viene gestito il Node Guarding e il protocollo di HeartBeat.
5) Viene gestito il messaggio d'emergenza (EMERGENCY).
6) La funzione Dynamic ID distribution (DBT slave) non è gestita.
7) E' implementata una Pre-Defined Master/Slave connection per semplificare il lavoro svolto dal Master durante la fase d'inizializzazione. Inhibit-Times (espresso in unità di 100 uS) è modificabile fino ad un valore di 1 min.
8) La sincronizzazione ad alta risoluzione non è supportata.
9) Non viene gestito TIME STAMP.
10) Nell'accesso ai parametri strutturati il sottoindice opzionale 0FFhex (accesso a tutto l'oggetto) non è gestito.
11) Per ragioni di efficienza viene gestito solo il trasferimento dati Expedited (max. 4 Byte) dei servizi SDO.

1.1.1 Pre-defined Master/Slave Connection

Il Pre-defined Master/Slave connection consente una comunicazione peer-to-peer tra un Master e 127 Slave; l'indirizzo di Broadcast è zero.

1.1.2 NMT Servizi (Network Management)

I servizi NMT mandatorysono:

- Enter_Pre-Operational_State CS = 128
- Reset_Node CS = 129
- Reset_Communication CS = 130

Essendo usato il Minimum Boot-up sono gestiti anche i seguenti servizi NMT:

- Start_Remote_Mode CS = 1
- Stop_Remote_Mode CS = 2

Il COB-ID di un servizio NMT utilizzato per l'inizializzazione è sempre 0; CS è il Command Specifier che definisce il servizio NMT.

1.1.3 Inizializzazione

Il drive ADV supporta il meccanismo di Node Guarding e HeartBeat. La configurazione del Node Guarding è effettuabile dal master mediante gli elementi dell'Object Dictionary previsti dallo standard (1006h, 100Ch, 100Dh) e gli oggetti 1016h e 1017h per HeartBeat.
Il drive verifica il funzionamento del master mediante il Life Guarding. Qualora il controllo fallisca, il drive attiva l'allarme di Bus Loss. La soglia di Life Guarding è così calcolata:

Valore	Condizione
60ms **SYNC_PERIOD (*)**	Default. Nessuna parametrizzazione del Node Guarding.
LIFE_TIME_FACTOR	Uso della modalità sincrona. Se non indicato dal master, il LIFE_TIME_FACTOR è pari a 3 per default.
NODE_GUARDING_PERIOD (*)	impostato dal master
LIFE_TIME_FACTOR	è pari a 3 se non diversamente indicato.

1.1.4 Oggetti di Comunicazione

In questo capitolo vengono descritti gli oggetti di comunicazione del protocollo CANopen gestiti dalla scheda d'interfaccia.Gli oggetti di comunicazione gestiti sono i seguenti:

1) 1 SDO Server in ricezione.
2) 1 SDO Server in trasmissione.
3) 4 PDO in ricezione.
4) 4 PDO in trasmissione.
5) 1 Emergency Object.
6) 1 Node Guarding - Life Guarding.
7) 1 SYNC object.

Nella seguente tabella sono illustrati gli oggetti di comunicazione in ordine di priorità crescente verso il basso, e il Message Identifier; per ottenere il COB-ID risultante, a tale numero va aggiunto il Node-ID (indirizzo scheda).

OBJECT	**MESSAGE ID**
NODE GUARDING & HB	1792 700h+NodeId
1st SDO rx	1536 600h+NodeId
1st SDO tx	1408 580h+NodeId
1st PDO tx	384 180h+NodeId
1st PDO rx	512 200h+NodeId
2nd PDO tx	640 280h+NodeId
2nd PDO rx	768 300h+NodeId
3st PDO tx	384 380h+NodeId
3st PDO rx	512 400h+NodeId
4th PDO tx	640 480h+NodeId
4th PDO rx	768 500h+NodeId
EMERGENCY	220 80h+NodeId
SYNC	128 80h
NMT	Network Management

Tabella 1.4.1: Oggetti di Comunicazione

1.1.5 Elementi Object Dictionary

Il dizionario oggetti (Object Dictionary) è accessibile mediante un master CANopen. Nella seguente tabella sono riportati gli oggetti di comunicazione impiegati e l'accessibilità con master CANopen.

Index (hex)	Name
1000	Device Type
1001	Error Register
1002	Manufacturer status register
1005	COB-ID SYNC Message
1006	Communication cycle period
1008	Manufacterer Device Name
1010	Store parameter
1009	Manufacturer Hardware Version
100A	Manufacturer Software Version
100C	Guard Time
100D	Life Time Factor
1014	COB-ID Emergency
1016	HeartBeat time consumer
1017	HeartBeat time producer
1018	Identity object
1400	1st Receive PDO
1401	2nd Receive PDO
1402	3rd Receive PDO
1403	4th Receive PDO
1600	Receive PDO1 mapping parameter
1601	Receive PDO2 mapping parameter
1602	Receive PDO3 mapping parameter
1603	Receive PDO4 mapping parameter
1A00	Transmit PDO1 mapping parameter
1A01	Transmit PDO2 mapping parameter
1A02	Transmit PDO3 mapping parameter
1A03	Transmit PDO4 mapping parameter
1800	1st Transmit PDO
1801	2nd Transmit PDO
1802	3rd Transmit PDO
1803	4th Transmit PDO

Tabella 1.5.1: Oggetti profilo di comunicazione CANopen impiegati

Gli oggetti indicati in grassetto nella tabella permettono di scrivere i parametri assegnati con lo scambio dati nel PDO.
Il criterio di assegnazione è variabile e dipende dalla dimensione (in byte) del parametro scambiato.

1.1.6 RX PDO Entries

Per quanto riguarda il PDO Communication Parameter (index 1400h, 1401h) esso è così strutturato:

1) Subindex 0 (Number of supported entries) = 2.
2) Subindex 1 (COB-ID usato dal PDO) è strutturato come segue:
- Bit 31 (PDO valido/non valido) è impostabile via SDO.
- Bit 30 (RTR Remote Transmission Request) = 0 poiché questa funzionalità non è supportata.
- Bit 29 = 0 poiché usato 11-bit ID (CAN 2.0A).
- Bit 11-28 non usati.
- Bit 0-10 COB-ID (vedi tabella 1.4.1).

3) Cyclic-synchronous Subindex 2 (Transmission Type), oppure sincrona secondo come impostato dal master (1 se previsto SYNC, 254...255 se asincrono). Se non indicato è attiva la modalità sincrona.

1.1.7 TX PDO Entries

Per quanto riguarda il PDO Communication Parameter (index 1800h, 1801h) esso è così strutturato:
1) Subindex 0 (Number of supported entries) = 3.
2) Subindex 1 (COB-ID usato dal PDO) è strutturato come segue:
- Bit 31 (PDO valido/non valido) è impostabile via SDO.
- Bit 30 (RTR Remote Transmission Request) = 0 poiché questa funzionalità non è supportata.
- Bit 29 = 0 poiché usato 11-bit ID (CAN 2.0A).
- Bit 11-28 non usati.
- Bit 0-10 COB-ID (vedi tabella 1.4.1).

3) Cyclic-synchronous Subindex 2 (Transmission Type), oppure sincrona secondo come impostato dal master (1 se previsto SYNC, 254...255 se asincrono). Se non indicato è attiva la modalità sincrona.
4) Inhibit time

1.1.8 SDO Entries

Viene utilizzata solo la modalità di trasferimento dati Expedited (max. 4 Byte).
Per quanto riguarda il SDO Communication Parameter esso è così strutturato:
1) Subindex 0 (Number of supported entries) = 3 poiché il dispositivo è un Server del servizio SDO.
2) Subindex 1 e 2 (COB-ID usato dal SDO) è strutturato come segue:
- Bit 31 (SDO valido/non valido); essendo usati solo gli SDO di Default, è = 1.
- Bit 30 riservato = 0.
- Bit 29 = 0 poiché usato 11-bit ID (CAN 2.0A).
- Bit 11-28 non usati.
- Bit 0-10 COB-ID (vedi tabella 1.4.1).

L'elemento node ID of SDO's client resp. server non è supportato poiché vengono usati solo gli SDO di Default.

1.1.9 COB-ID SYNC Entries

Per quanto riguarda il parametro di comunicazione COB-ID SYNC, i 32 bit sono così strutturati:
- Bit 31 = 1 essendo la scheda d'interfaccia CANopen consumer di messaggi SYNC.
- Bit 30 = 0 poiché la scheda d'interfaccia non genera messaggi SYNC.
- Bit 29 = 0 poiché usato 11-bit ID (CAN 2.0A).
- Bit 11-28 non usati.
- Bit 0-10 COB-ID (vedi tabella 1.4.1).

1.1.10 COB-ID Emergency

Per quanto riguarda il parametro di comunicazione COB-ID Emergency Message, i 32 bit sono così strutturati:

- Bit 31 = 0 non essendo la scheda d'interfaccia CANopen consumer di messaggi Emergency.
- Bit 30 = 0 poiché la scheda d'interfaccia genera messaggi Emergency.
- Bit 29 = 0 poiché usato 11-bit ID (CAN 2.0A).
- Bit 11-28 non usati.
- Bit 0-10 COB-ID (vedi tabella 1.4.1).

1.2 Gestione CANopen

L'interfaccia utente del protocollo CANopen avviene tramite parametri drive.
I parametri sono gestiti con menu gerarchici. Tutti i parametri di scrittura relativi ai bus di campo sono attivi solo dopo il reset del drive. Di seguito sono riportati i parametri drive per la gestione del protocollo CANopen.
Per attivare la scheda EXP-CAN-ADV impostare il parametro PAR 4000 **Fieldbus type** come CANopen oppure DS402.
I seguenti parametri sono disponibili nel menu COMMUNICATION->FIELDBUS CONFIG :

PAR	Nome Par	tipo	Valore default	Attr
4004	Fieldbus baudrate	Enum	None	Scrittura
4006	Fieldbus address	2 byte senza segno	0	Scrittura
4010	Fieldbus M->S enable	Enum	On	Scrittura
4012	Fieldbus alarm mode	2 byte senza segno	0	Scrittura
4014	Fieldbus state	Enum	Stop	Sola lettura

- Fieldbus baudrate = Imposta il baudrate della rete. Valori disponibili per CANopen : 125k , 250k , 500k ,1M
- Fieldbus address = indirizzo di questo nodo slave nella rete , valori ammessi da 1 a 127
- Fieldbus M->S enable = se messo a Off i dati negli RPDO non vengono processati dal drive
- Fieldbus alarm mode = se messo a 1 il drive genera errori Opt Bus Fault relativi alla perdita di comunicazione (Bus Loss) anche quando il drive non è abilitato.
- Fieldbus state = stato della comunicazione per questo nodo su rete CANopen: Stop, Pre-Operational, Operational.

1.3 Controllo del Process Data Channel

Questa funzione permette l'assegnazione di parametri o variabili applicative dell'azionamento ai dati del Process Data Channel.
Nel caso di protocollo CANopen, il PDC è realizzato mediante i messagi PDO (Process Data Object).
Il protocollo CANopen utilizza un numero impostabile di word per il Process Data Channel (abbr. PDC).
La configurazione del Process Data Channel per i bus di campo è la seguente:

Data 0 Data... Data n

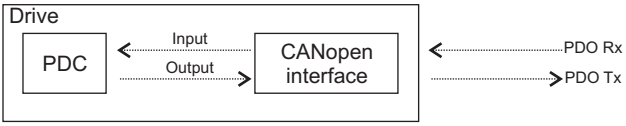
Il drive è in grado sia di leggere che di scrivere i dati del Process Data Channel. Un dato può essere sia di 2 che di 4 byte. Con il termine dati si intende un numero qualsiasi di parametri da 0 a 16, purchè il numero di byte richiesti da questi parametri non sia superiore a 32 byte.

Esempio:
Si possono avere:

- da 0 a 16 dati da 2 byte
- 1 dato a 4 byte + da 0 a 14 dati da 2 byte
- 2 dati da 4 byte + da 0 a 12 dati da 2 byte
...
- 8 dati da 4 byte

I dati scambiati mediante il PDC possono essere di due tipi:
- parametri dell'azionamento
- variabili di un'applicazione MDPlc. L'utilizzo di variabili MDPlc è descritto nei paragrafi 1.3.1 e 1.3.2.

Il master scrive i dati definiti come PDC input e legge i dati definiti come PDC output.



1.3.1 Configurazione PDC Input (FB XXX MS Parameter)

La configurazione dei dati scambiati negli RPDO avviene attraverso i parametri del menu COMMUNICATION->FIELDBUS M2S (vedere il manuale del drive).
Il mapping dei dati nei PDO viene effettuato in base al formato del dato impostato in Fieldbus M->Sn sys in base alle seguenti regole :

– I PDO vengono riempiti a partire dall'RPDO1
– Al raggiungimento della dimensione di 4 word il PDO è completo e si passa quindi a riempire l'RPDO successivo fino al limite di 4 PDO
– I dati a 32 bit (long o float) non possono essere divisi tra PDO , devono essere posizionati all'interno del PDO (viene generato un allarme)
– E' possibile creare PDO di dimensioni inferiori a 4 word , utilizzando **Fieldbus M->Sn ipa** = 0 ma assegnati (**Fieldbus M->Sn sys** diverso da Not Assigned ,Fill16 o Fill32) dopo un dato assegnato.
(N.B.: se l'assegnamento è Fill16 o Fill32 il dato viene comunque inserito nel PDO)
– Al primo parametro **Fieldbus M->Sn sys** = Not Assigned il riempimento dei PDO è completo. La dimensione dell'ultimo PDO dipende quindi dai dati che sono stati assegnati.
– ***Esempio : RPDO1 di 2 word e RPDO2 di 2 word:***
Fieldbus M->S1 ipa = 610 (Ramp ref 1 src)
Fieldbus M->S1 sys = Eu
Fieldbus M->S2 ipa = 4452 (Word decomp src)
Fieldbus M->S2 sys = Count 16
Fieldbus M->S3 ipa = 0

Fieldbus M->S3 sys = Fill 32
Fieldbus M->S4 ipa = 3660 (Compare input 1 src)
Fieldbus M->S4 sys = Count32
Fieldbus M->S5 sys = Not Assigned

1.3.2 Configurazione PDC output (FB XXX SM Parameter)

La configurazione dei dati scambiati negli RPDO avviene attraverso i parametri del menu COMMUNICATION->FIELDBUS S2M (vedere il manuale del drive).
Il mapping dei dati nei PDO viene effettuato in base al formato del dato impostato in **Fieldbus M->Sn sys** in base alle seguenti regole :

– I PDO vengono riempiti a partire dal TPDO1
– Al raggiungimento della dimensione di 4 word il PDO è completo e si passa quindi a riempire il TPDO successivo fino al limite di 4 PDO
– I dati a 32 bit (long o float) non possono essere divisi tra PDO, devono essere posizionati all'interno del PDO (viene generato un allarme)
– E' possibile creare PDO di dimensioni inferiori a 4 word, utilizzando **Fieldbus S->Mn ipa** = 0 ma assegnati (**Fieldbus M->Sn sys** diverso da Not Assigned) dopo un dato assegnato.
– Al primo parametro **Fieldbus S->Mn sys** = Not Assigned il riempimento dei PDO è completo. La dimensione dell'ultimo PDO dipende quindi dai dati che sono stati assegnati.

1.3.3 Uso del PDC in Applicazioni MDPlc

E' possibile configurare sia i dati PDC in input che quelli in output per consentire l'accesso diretto dei dati medesimi mediante il codice applicativo MDPlc.
Per i dati in lettura è sufficiente impostare **Fieldbus M->Sn sys** su Mdplc16 o Mdplc32, lasciando **Fieldbus M->Sn ipa** = 0.
L'applicazione Mdplc può ora leggere il dato in arrivo direttamente dal parametro **Fieldbus M->Sn m**on.
I dati in scrittura vengono configurati impostando **Fiedlbus S->Mn ipa** = [4184 +(n-1)*10] **(Dig Fieldbus S->Mn**, 1 ≤ n ≤ 16).
Fiedlbus S->Mn sys viene automaticamente settato a Mdplc. E' compito dell'applicazione scrivere il dato nel parametro **Dig Fieldbus S->Mn** per inviarlo sul bus.

1.4 Gestione SDO

Il servizio SDO è sempre disponibile.
L'accesso ai parametri drive avviene attraverso la Manufacturer Specific Profile Area (2000hex< index <5FFFhex).
L'indice da indicare nel comando SDO per accedere ad un parametro del drive si ottiene mediante le seguenti regole:
SDO index = PAR + 2000h
SDO subindex = 1

Il campo Data deve contenere il valore del parametro drive.

Esempio:
Scrittura del valore 1000 nel parametro PAR 600 **Dig Ramp ref 1** (258hex).

Le informazioni necessarie sono le seguenti:
1) L'indice SDO ottenuto con la formula è
   2000hex + 258hex = 2258h
2) Valore da scrivere 1000, corrispondente a 03E8 hex.

Index		Subindex				
22h	**58h**	**01h**	**E8h**	**03h**	**00h**	**00h**
Drive parameter index		Subindex	Drive parameter value to be assigned to SDO			

In caso vi sia un errore nell'impostazione o nella lettura del parametro, l'interfaccia CANopen invia un messaggio di Abort domain transfer; il valore di Application-error-codes assume i seguenti significati:

Error class	Error code	Additional code (hex)	Significato
6	0	0	Parameter doesn't exist
8	0	22	Acces failed because of present device state
6	1	2	Read/Write only error
8	0	0	Generic error
6	9	32	Minimum value
6	9	31	Maximum value
5	4	0	SDO time_out
5	4	1	Invalid command
3	9	30	Invalid value

1.5 Allarmi

Allarmi Fieldbus

I malfunzionamenti del bus sono segnalati mediante l'allarme Opt Bus Fault. Nel caso di CANopen, le cause possibili di anomalie sono:

- stato Bus-off della lina CAN;
- abilitazione del drive in uno stato diverso da Operational;
- superamento della soglia di monitoraggio (heartbeat, node guard, communication cycle period).
- errore di configurazione del drive

Questo allarme si attiva solo quando il drive è abilitato.
Il parametro PAR 4014 **Fieldbus alarm mode**, se posto a ON, abilita la generazione dell'allarme Field bus failure anche quando il drive è disabilitato.

Codice	Cfg	Descrizione	Azioni
0		Bus Loss	Check line for noise, terminations , problems with cabling
FF01	*	Fieldbus type does not match expansion card	Verify if EXP-CAN-ADV card is properly installed
FF02	*	Wrong baudrate selected	Check "Fieldbus baudrate" is one of 125k, 250k, 500k, 1M
FF03	*	Invalid address for node	Check "Fieldbus address"
FF04	*	Error initializing CAN interface	Internal error, contact manufacturer
FF14..FF23	*	Wrong object selected for mapping in channel M2S n	Check "Fieldbus M->Sn Dest

Codice	Cfg	Descrizione	Azioni
FF24..FF33	*	More than 1 Src pointing to M2S Channel n	Check for multiple destinations on “Fieldbus M->Sn Dest”
FF34..FF43	*	M2S Channel n , data size is wrong (16 bits on 32 bits or 32 bits on 16 bits parameter)	Check “Fieldbus M->Sn sys”
FF44..FF53	*	Invalid parameter in channel S2M n	Check “Fieldbus S->Mn src”
FF54..FF63	*	S2M Channel n , data size is wrong (16 bits on 32 bits or 32 bits on 16 bits parameter)	Check “Fieldbus S->Mn sys”
FF64..FF73	*	Wrong object selected for mapping in channel S2M n	Check “Fieldbus S->Mn src”
FF74..FF83	*	M2S Channel n : too many words in PDC	“Fieldbus M-Sn dest” & “Fieldubs M->Sn sys” address more than 16 words in PDC
FF84..FF93	*	S2M Channel n : too many words in PDC	“Fieldbus S->Mn src” & “Fieldubs S->Mn sys” address more than 16 words in PDC
FFB4..FFC3	*	Internal database error on channel n	Internal error, contact manufacturer
8110		CAN msg overflow	Too many packets for selected baudrate
8130		LifeGuard/HeartBeat error	Software timeout from master
FFC5		Wrong NMT message length	Check NMT packets
FFC6		Invalid NMT command	Check NMT packets
FFC7		CAN bus off	Check line state for problems
8100		CAN bus off	Check line state for hardware problems

Segnalazione allarmi drive

La gestione degli allarmi del drive viene effettuata tramite messaggio di Emergency, che contiene il codice di errore relativo all’allarme generato, secondo la seguente tabella :

Selezione	Codice
No alarm	0x0000
Overvoltage	0x3210
Undervoltage	0x3220
Ground fault	0x2110
Overcurrent	0x2310
Desaturation	0x2130
MultiUndervolt	0xFF06
MultiOvercurr	0xFF07
MultiDesat	0xFF08
Heatsink OT	0x4210
HeatsinkS OTUT	0x4310
Intakeair OT	0x4130
Motor OT	0xFF0C
Drive overload	0x8311

Selezione	Codice
Motor overload	0x7121
Bres overload	0x7112
Phaseloss	0xFF10
Opt Bus fault	0xFF11
Opt 1 IO fault	0xFF12
Opt Enc fault	0x3130
External fault	0x9000
Speed fbk loss	0x7310
Overspeed	0x8400
Plc1 fault	23
Plc2 fault	24
Plc3 fault	25
Plc4 fault	26
Plc5 fault	27
Plc6 fault	28
Plc7 fault	29
Plc8 fault	30
Emg stop alarm	31
Watchdog	32
Trap error	33
System error	34
User error	35
Power down	36
Speed ref loss	37
Not Used1	38
Opt 2 IO fault	39
Not Used2	40
Not Used3	41
Not Used4	42
Not Used5	43
Not Used6	44
Param error	45

1.6 Esempio di programmazione

Questo capitolo contiene un esempio di come programmare i parametri del drive ADV200 per poter leggere e scrivere i parametri da master CANopen tramite i canali di Processo (PDO). Per i canali di configurazione (SDO) vedere il capitolo 1.4. Il paragrafo 1.6.1 contiene le informazioni che servono su un master CANopen che governa una macchina. Nel paragrafo 1.6.2 vi sono le informazioni base per la programmazione del drive ADV200 partendo dalle condizioni di fabbrica.

1.6.1 Master CANopen

Questo paragrafo contiene un esempio di scambio dati visto dal master. Sono le informazioni normalmente dettate dalla specifica di macchina nel caso di applicazione governata da un master CANopen.

1.6.1.1 Descrizione Comunicazione PDO Master -> Slave

I due parametri da scrivere tramite i canali di processo sono il primo una parole di comandi (che chiameremo control word) in cui i singoli bit contengono alcuni comandi (es. enable, start ..). Il secondo canale di processo contiene il riferimento di rampa 1 (RampRef1) in rpm.

PDO CANopen: Master -> Drive (max 16 word)

Posizione	Descrizione	Formato	Unità Misura
Word1 M -> S	Control word	16 bit Word	...
Word2 M -> S	Ramp Ref 1	Int 16 bit	rpm
Word3 M -> S			
Word4 M -> S			
Word5 M -> S			
Word6 M -> S			
Word7 M -> S			
...			
...			
Word16 M > S			

CONTROL WORD

Bit	Descrizione	Note
0	EnableCmd	Comando di enable da master CANopen
1	StartCmd	Comando di start da master CANopen
2	Free	
3	Free	
4	Free	
5	Free	
6	Free	
7	Free	
8	Digital Out3	Uscita digitale 3 comandata da master CANopen
9	Digital Out4	Uscita digitale 4 comandata da master CANopen
10	Free	
11	Free	
12	Free	
13	Free	
14	Free	
15	Free	

1.6.1.2 Descrizione Comunicazione PDO Slave -> Master

Il master can legge tre parametri dal drive il primo contiene una parola di stato (*Status Word*) i cui singoli bit contengono informazione di stato del drive (es. DriveOk..). Il secondo parametro è la velocità attuale in rpm. Il terzo parametro contiene il valore dell'ingresso analogico 2.

PDO CANopen Slave > Master (max 16 Word)

Posizione	Descrizione	Formato	Unità Misura
Word1 S -> M	Status Word	16 bit Word	BitWide
Word2 S -> M	Actual Speed	Int 16 bit	rpm
Word3 S -> M	Analog Input 2	Int 16 bit	
Word4 S -> M			
Word5 S -> M			
Word6 S -> M			
Word7 S -> M			
...			
...			
Word16 S -> M			

STATUS WORD

Bit	Descrizione	Note
0	EnableState	Drive abilitato
1	Drive Ok	Drive Ok
2	Speed is zero	Soglia di velocità zero
3	Free	
4	Free	
5	Free	
6	Free	
7	Free	
8	Digital Input 4	Stato ingresso digitale 4 ADV200
9	Digital Input 5	Stato ingresso digitale 5 ADV200
10	Free	
11	Free	
12	Free	
13	Free	
14	Free	
15	Free	

1.6.2 Programmazione ADV200

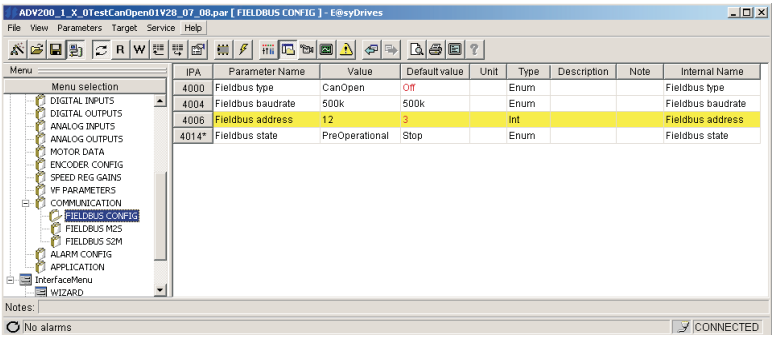
Nell'esempio riportato in questo paragrafo la prima ipotesi è che i parametri del drive ADV200 siano quelli di fabbrica (comando di **Default parameter**).

1.6.2.1 FIELDBUS CONFIG

Nell'esempio di suppone che il drive sia il nodo 12 e la comunicazione CANopen avvenga a una baudrate di 500k.

Nota: Tutte le impostazioni e la configurazioni fieldbus hanno effetto solo al successivo reset del drive.

Programmare i parametri del menù fieldbus come nella seguante figura:



Lo stato pre-operational dei led dell'espansione CANopen è indicato nella relativa colonna :

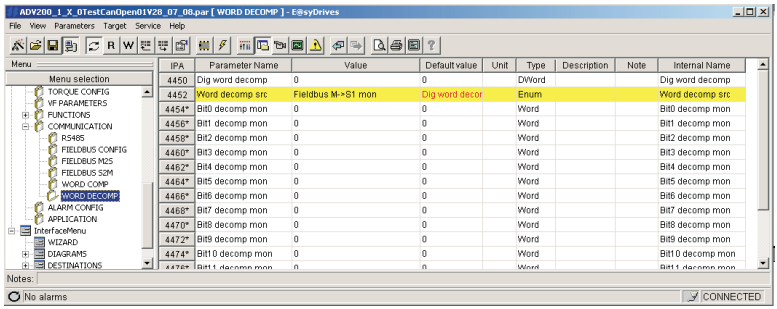
Led	Stato = Pre-operational		Stato = Operational
AL (Rosso)	ON	condizione di allarme	OFF
OP (Verde)	OFF	condizione di Pre-Operational	ON
PWR (Verde)	ON	espansione alimentata e attiva	ON
+5VC (Verde)	ON	alimentazione optoisolata CAN	ON

In queste condizioni la comunicazione dei canali di processo non è attiva.
Al termine dalla programmazione del drive (vedi paragrafi successivi) è possibile attivare la comunicazione dal master tramite il comando NMT "start node".

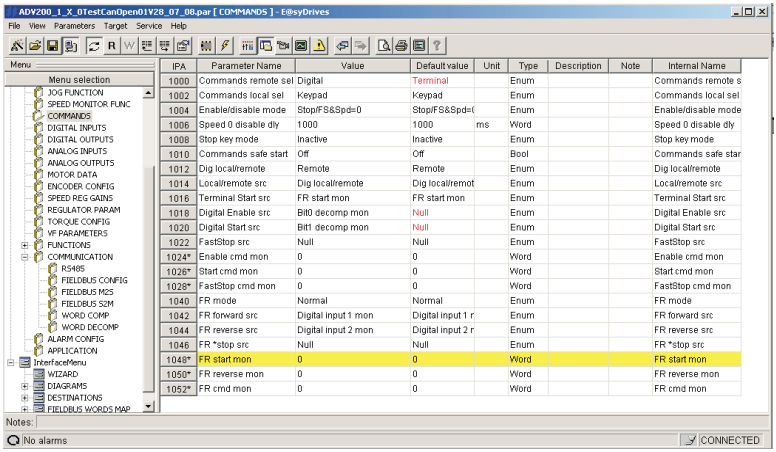
Alla ricezione di questo comando il parametro FieldBus State si porta nella condizione di Operational e il led verde OP diventa "ON". Solo ora i canali di processo sono attivi.

1.6.2.2 Configurazione canali MASTER -> SLAVE

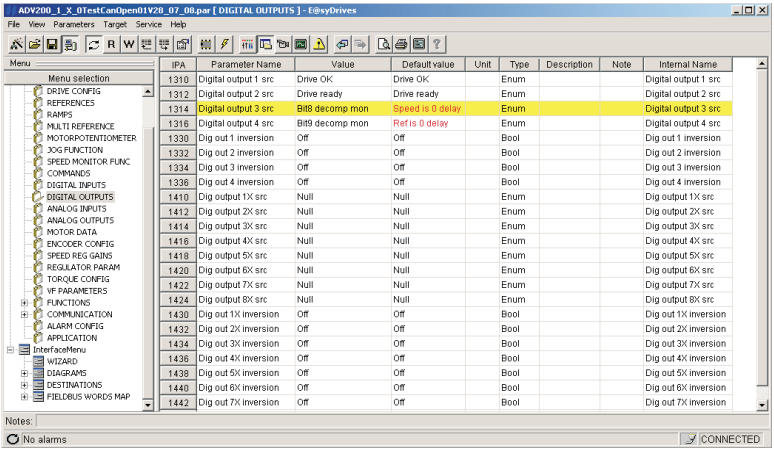
La **configurazione della control word** avviene utilizzando Wdecomp. La seguente figura mostra la programmazione di Wdecomp sulla prima word M > S (modalità "Export"):



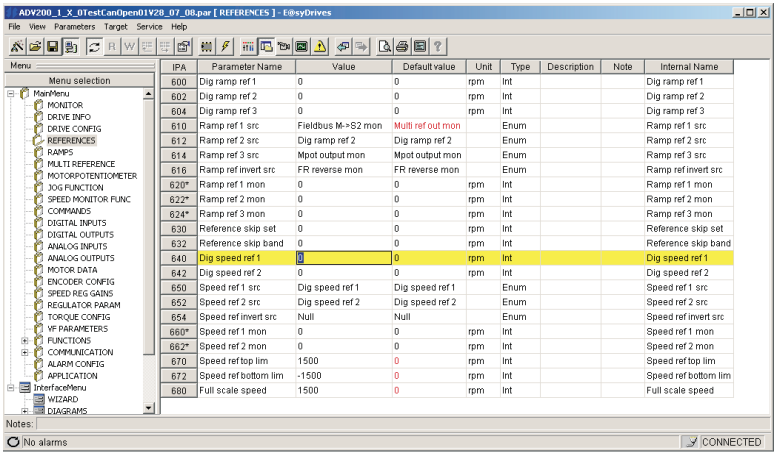
Ora basta connettere i singoli bit di wdecomp. Per i Comandi si ricorda che come indicato sul manuale ADV200, il drive deve essere in modalità “**Remote**” e “**Digital**”.
La programmazione dei primi due bit avviene nel menù command come mostrato in figura:



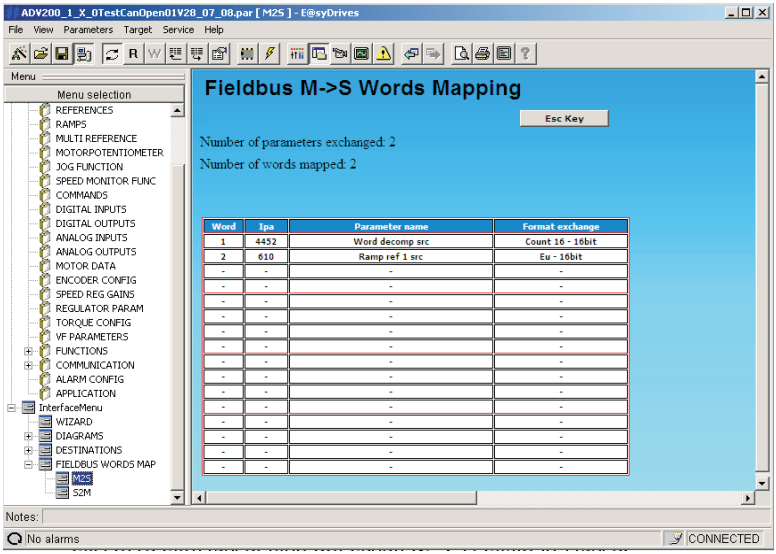
La programmazione dei bit 8 e 9 dalla “Command word” avviene come mostrato in figura (menù Digital Outputs):



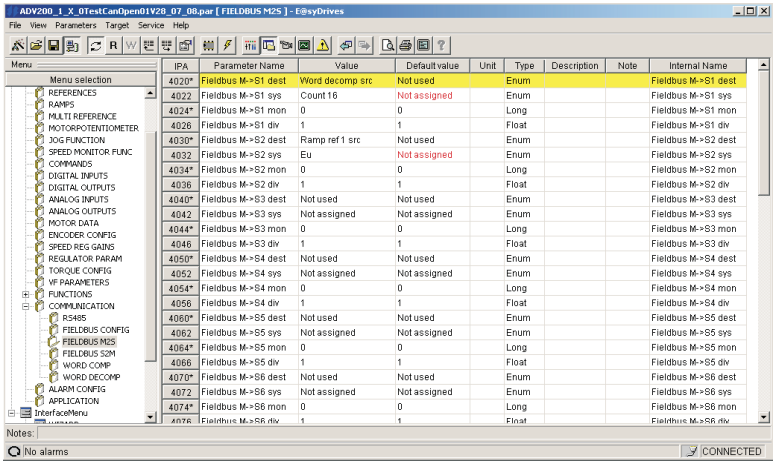
La configurazione della seconda word avviene nel menù “References”:



Dopo aver eseguito un comando di save e riacceso il drive è possibile verificare la corretta configurazione dei canali M -> S come in figura (Pagina Html):

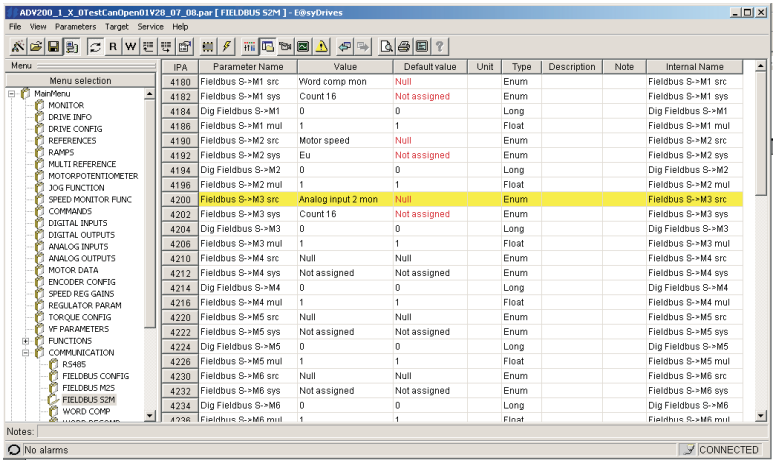


Le medesime informazioni sono disponibili anche nel menù Fieldbus M2S:

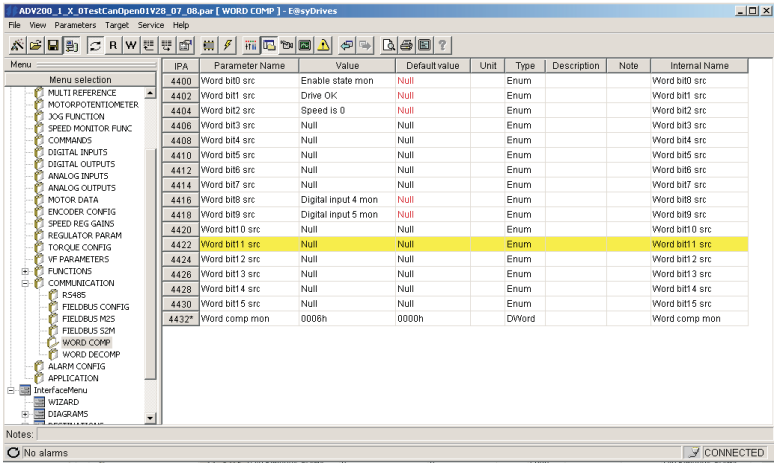


1.6.2.3 Configurazione canali SLAVE -> MASTER

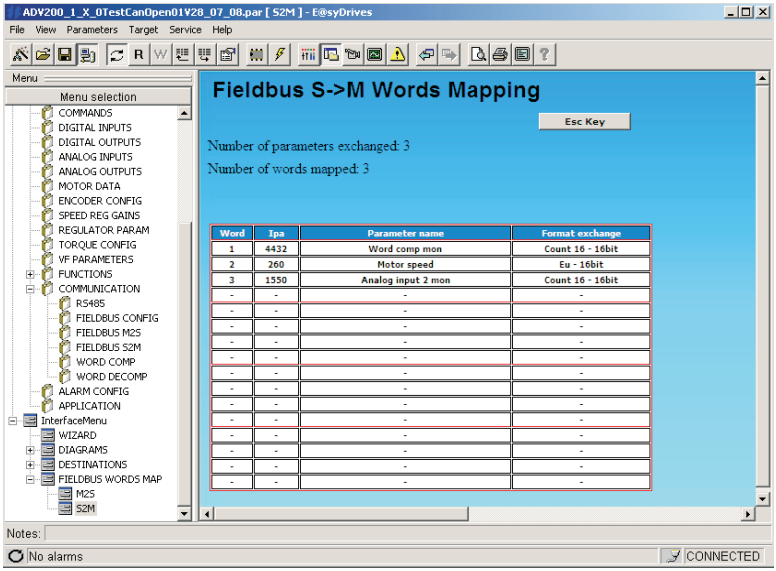
La configurazione di questi canali avviene nel menù Fieldbus S2M. Per la programmazione del primo canale viene usata la Wcomp.
La seguente figura mostra la programmazione S2M:



La seguente figura mostra la programmazione di Wcomp:



In modo analogo dopo un salvataggio e riaccensione del drive, è possibile verificare la corretta programmazione dei canali Slave -> Master:



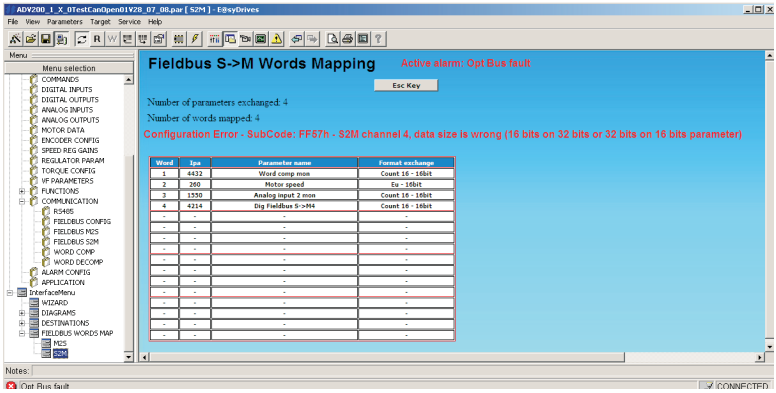
1.6.2.4 Verifica comunicazione

Alcune note /suggerimenti per la verifica della comunicazione.

- La comunicazione PDO è attiva solo in “Operational”. Verificare lo stato tramite Easy Drive o led scheda espansione.
- Per la comunicazione Master -> Slave nel menù FIELDBUS M2S è possibile verificare il valore ricevuto dal canale di comunicazione (Es per il primo canale è il parametro Fieldbus M->S1 Mon).
- Nel caso di comunicazione in EU (unità ingegneristice) ricordarsi che il valore letto sui parametri Mon di FIELDBUS M2S è in unità interne (vedere tabelle di conversione nel capitolo 5.0 SYSTEM INTERNAL VARIABLES, manuale “ADV200, Write the applications with the MDPlc” disponibile sul sito www.weg.net).

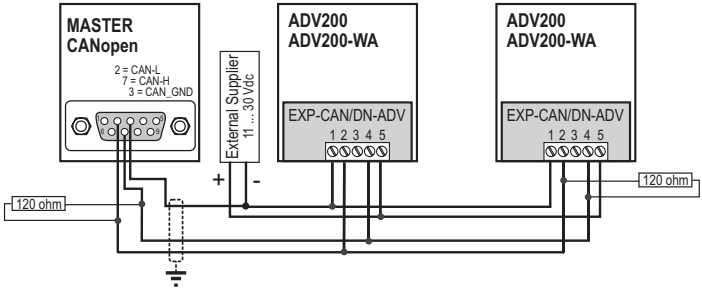
1.6.2.5 Errori di configurazione

Nel caso di errore di configurazione di un canale. All'accensione il drive và in allarme
"Option bus fault" e fornisce un codice di errore che indica il canale che ha generato l'allarme. Sul manuale dell'espansione vi è la tabella che riporta tutti i codici di errore.
Con il software di configurazione WEG_eXpress è sufficiente connettersi alla relativa pagina HTML come mostrato nella seguente figura:



1.7 Esempio di collegamento pannello di controllo master a nodi ADV200

Nella figura seguente viene illustrato il collegamento tra pannello di controllo master Canopen ed i nodi slave ADV200.



Il collegamento è del tipo a tre fili, linea CAN e cavo equipotenziale tra morsetti 1 (V- / CAN_GND) delle schede EXP-CAN/DN-ADV, 0 V alimentatore e pin 3 (CAN_GND) connettore Can su terminale Vedo.
Deve essere garantita la continuità dello schermo lungo tutta la linea CAN.
Lo schermo viene messo a terra in un solo punto, in prossimità dell'alimentatore della linea CAN; questo alimentatore può anche essere utilizzato per l'alimentazione del terminale Vedo. E' sconsigliabile l'utilizzo dell'alimentatore suddetto per altre utenze, in particolare circuiti ausiliari con rele.
Lo schermo della linea Can può anche essere messo a terra in due o più punti, come ad esempio nel caso in cui i nodi della linea Can siano distribuiti all'interno di quadri elettrici separati, a patto che le connessioni di terra siano eseguite correttamente.

2.0 Funzionamento secondo profilo DS402

Se il parametro **Fieldbus type** è impostato su DS402 il drive lavora con il profilo standard per Drives & Motion Control Ver 2.0 e riporta Device Identity 192H (402) nell' oggetto 1000h.
Il drive ADV200 supporta il Velocity Mode.
Nella configurazione di default il drive viene automaticamente impostato per utilizzare i PDO no. 6 (DS402 par 7.2.1 & 7.2.2), mappati su RPDO1 e TPDO1 con COB-ID 200h & 180h +NodeId

PDO	Nr. Oggetto	Nome Oggetto	Descrizione
6	6040h	Controlword	controls the state machine and the nominal speed (vl)
	6042h	Target velocity (vl)	
6	6041h	Statusword	shows status and the current speed (vl)
	6044h	vl control effort	

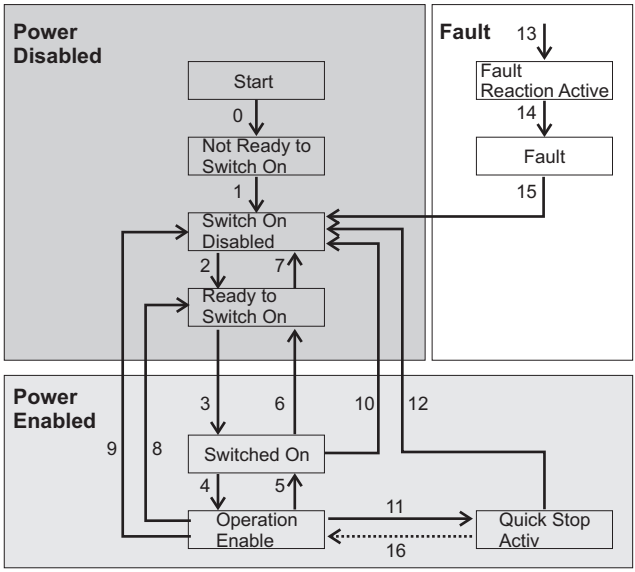
I rimanenti PDO possono essere impostati dall'utente.

Gli oggetti DS402 implementati sono i seguenti :

N. Oggetto		Descrizione	Tipo	Accesso	Mandatory
6040h	VAR	Controlword	UNSIGNED16	rw	M
6041	VAR	Statusword	UNSIGNED16	ro	M
6042h	VAR	vl target velocity (1)	INTEGER16	rw	M
6043h	VAR	vl velocity demand (1)	INTEGER16	ro	M
6044h	VAR	vl control effort(1)	INTEGER16	ro	M
6046h	ARRAY	vl velocity min max amount	UNSIGNED32	rw	M
6048h	RECORD	vl velocity acceleration	vl velocity acceleration	rw	M
6049	RECORD	vl velocity deceleration	vl velocity deceleration	rw	M
6060h	VAR	Modes of operation (2)	INTEGER8	rw	M
6061h	VAR	Modes of operation display	INTEGER8	ro	M

(1) L'unità di misura per gli oggetti 6042h , 6043h , 6044h è espressa in Rpm
(2) L'oggetto 6060h è disponibile solo in quanto obbligatorio. Dato che il drive supporta esclusivamente il Velocity Mode non è possibile modificare il valore di questo oggetto.

Il dispositivo opera secondo la macchina a stati DS402 (vedere CiA DSP 402 V 2.0 , par 10.1.1) :


Power
Disabled
Start
0
Not Ready to
Switch On
1
Switch On
Disabled
2
7
Ready to
Switch On
Fault
13
Fault
Reaction Active
14
Fault
15
Power
Enabled
3
6
10
12
9
8
Switched On
4
5
11
Operation
Enable
Quick Stop
Activ
16

3.0 Interfaccia DeviceNet

Il presente capitolo descrive il collegamento del drive ADV200 a reti DeviceNet. Questo capitolo si rivolge a progettisti e tecnici di manutenzione e messa in servizio di sistemi DeviceNet:
E' necessaria dunque una conoscenza di base di DeviceNet fornita dai seguenti manuali:

- DeviceNet Specifications. Volume 1 - DeviceNet Communication Model and Protocol (Issued by ODVA).
- DeviceNet Specifications. Volume 2 - DeviceNet Device Profiles and Object Library (Issued by ODVA)

3.1 Descrizione Generale di DeviceNet

DeviceNet è un profilo di comunicazione per sistemi industriali basati su CAN. Come protocollo CAN (ISO 11898) viene utilizzato CAN2.0A con l'identificatore a 11 bit.
Il drive ADV200 è sviluppato come Slave UCMM Capable Device per operare solo in Predefined Master/Slave Connection Set.
Lo scambio di dati avviene ciclicamente; l'unità Master legge i dati messi a disposizione dagli Slave e scrive i dati di riferimento degli Slave; i Baud Rate che la scheda SBI può supportare sono: 125 kbit, 250 kbit e 500 kbit.

Il supporto fisico è la linea seriale RS485; al Bus possono essere collegati un numero massimo di 64 Slave.

3.2 Funzionalità DeviceNet

In questo capitolo vengono descritte le funzionalità di DeviceNet gestite dal drive ADV200. Le caratteristiche salienti sono:

1. Il drive può operare solo come Slave in Predifined Master/Slave Connection Set.
2. All'interno del Predefined Master/Slave Connection Set il drive è un UCMM Capable Device.
3. Viene gestito l'Explicit Messaging.
4. Viene gestito il Polling per l'interscambio veloce ciclico Master/Slave.
5. Viene implementato il meccanismo di rilevazione del Duplicate MAC ID.

Per quanto concerne l'Explicit Messaging, viene gestita la frammentazione del pacchetto con un massimo di 32 byte totali.

Taglie di connessione

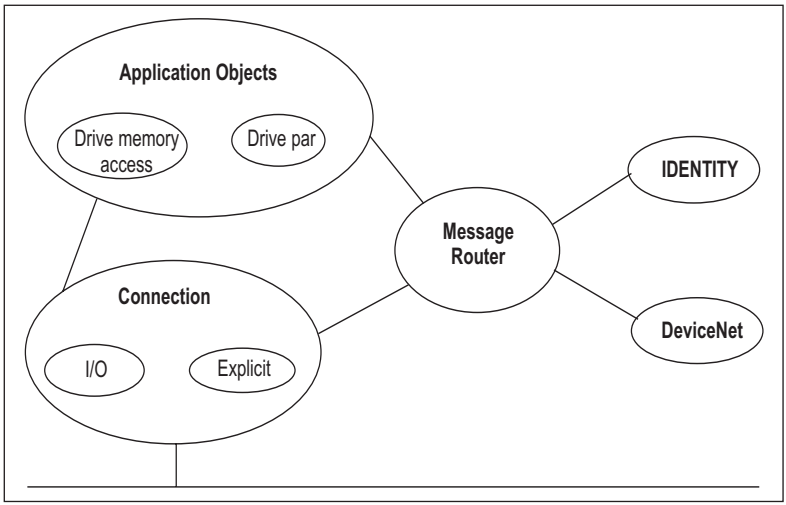
CONNECTION INSTANCE	PRODUCED	CONSUMED
Polled I/O	Depending on frame setting	
Explicit messaging	32	32

3.3 Descrizioni oggetti

Di seguito viene riportata la descrizione degli oggetti gestiti dal drive ADV200.

3.3.1 Object Model

La seguente figura rappresenta "Object Model" dell'ADV200.



La tabella seguente indica:
1. Le classi dell'oggetto presenti nella scheda EXP-CAN-ADV.
2. Se la classe è obbligatoria.
3. Il numero di istanze presenti in ogni classe.

Vedere DeviceNet Specifications per le classi Standard.

Object	Optional/Required	# of Instances
Identity	Required	1
Message Router	Required	1
DeviceNet	Required	1
Connection	Required	1 I/O, 3 Explicit
Parameter	Optional	many
Drive Parameter Access	Optional	many
Drive memory Access	Optional	many

3.3.2 How Objects Affect Behavior

L'Affect Behaviour degli oggetti è riportato nella seguente tabella:

Object	Effect on Behavior
Identity	Supports Reset Service
Message Router	No effect
DeviceNet	Port attributes configuration
Connection	Contains the number of logical ports
Parameter	Drive parameters read/write
Drive Parameter Access	Drive parameters read/write
Drive Memory Access	Drive parameters read/write

3.3.3 Defining Object Interface

L'interfaccia degli oggetti del drive ADV200 è la seguente:

Object	Interface
Identity	Message router
Message Router	Explicit Messaging Connection Instance
DeviceNet	Message router
Connection	Message router
Parameter	Message router
Drive Parameter Access	Message router
Drive memory Access	Message router

3.4 Trasferimento dati via Explicit Messaging

Il trasferimento dati via Explicit Messaging avviene tramite due nuovi oggetti: uno per l'accesso ai parametri Drive, l'altro per l'accesso diretto alla memoria del drive.

3.4.1 Parameter Access

Per la lettura/scrittura dei parametri Drive viene definito l'oggetto Drive Parameter Access con le seguenti caratteristiche:

- Class ID: Fh.
- Class Attribute: Revision
- Instance Attribute: Questa instanza non prevede attributi.

3.4.1.1 Class Code

Class code: F hex

3.4.1.2 Class attributes

Number	Need in implementation	Access Rule	Name	DeviceNet Data Type	Description of Attribute	Semantics of values
1	Optional	Get	Revision	UINT	Revision of this object	

3.4.1.3 Instance Attributes

Number	Need in implementation	Access Rule	Name	DeviceNet Data Type	Description of Attribute	Semantics of values
This instance provide attributes and correspond to the PAR						

3.4.1.4 Common Services

Questo oggetto non ha servizi comuni.

3.4.1.5 Object Specific Services

Service Code	Need in implementation		Service Name	Description of Service
	Class	Instance		
0hex	n/a	Required	Get_Attribute_Single	Read drive parameter value
10hex	n/a	Required	Set_Attribute_Single	Writes drive parameter value

3.4.1.6 Behavior

Questo oggetto è l'interfaccia tra la rete DeviceNet e tutti i parametri Drive.
L'accesso al parametro Drive avviene tramite l'indice del parametro stesso.
*Esempio lettura di un parametro (PAR 600 **Dig Ramp ref 1**):*

- Eseguire un Get_Attribute_Single della classe Fh
- istanza =600 (258 hex)
- attributo di classe 1
- il drive risponde con 4 byte (formato Dword).

*Esempio scrittura di un parametro (PAR 600 **Dig Ramp ref 1**):*

- Eseguire un Set_Attribute_Single della classe Fh
- istanza = 600 (258 hex)
- attributo di classe 1
- per impostare il valore 1000, selezionare Word 2 byte (il formato del parametro è INT, 16 bit)
- nel caso di errore il drive non risponde (timeout).

byte	**VALUE**	**XX**	Low byte - Low word drive parameter drive
			High byte - Low word drive parameter drive
			Low byte - High word drive parameter drive
			High byte - High word drive parameter drive

Il numero di byte del campo Value dipende dalla lunghezza del parametro del drive.

Esempio:
se il tipo di parametro del drive Integer la lunghezza di VALUE è 2 bytes.

3.4.2 Drive Parameter Access

Per la lettura/scrittura dei parametri Drive viene definito l'oggetto Drive Parameter Access con le seguenti caratteristiche:
- Class ID: 66h.
- Class Attribute: Revision
- Instance Attribute: Questa instanza non prevede attributi.

3.4.2.1 Class Code

Class code: 66 hex

3.4.2.2 Class attributes

Number	Need in implementation	Access Rule	Name	DeviceNet Data Type	Description of Attribute	Semantics of values
1	Optional	Get	Revision	UINT	Revision of this object	

3.4.2.3 Instance Attributes

Number	Need in implementation	Access Rule	Name	DeviceNet Data Type	Description of Attribute	Semantics of values
This instance does not provide attributes						

3.4.2.4 Common Services

Questo oggetto non ha servizi comuni.

3.4.2.5 Object Specific Services

Service Code	Need in implementation		Service Name	Description of Service
	Class	Instance		
32hex	n/a	Required	Get_Drive_Value	Get parameter from Drive
33hex	n/a	Required	Set_Drive_Value	Set parameter into Drive
34hex	n/a	Required	Get_Typed_Drive_ Value	Read drive parameter value indicating the data type
35hex	n/a	Required	Set_Typed_Drive_ Value	Writes drive parameter value indicating the data type

3.4.2.6 Behavior

Questo oggetto è l'interfaccia tra la rete DeviceNet e tutti i parametri Drive.
L'accesso al parametro Drive avviene tramite l'indice del parametro stesso; se il parametro non esiste o non è possibile accedervi per qualsiasi motivo (es. tentare di scrivere un parametro di sola lettura), viene ritornato un codice d'errore.
Non è possibile accedere ai parametri Drive in formato testo.
In seguito sono riportati schemi di come deve essere organizzato il pacchetto dati per leggere/scrivere parametri Drive.

A) Write Drive Parameter

In questo esempio viene riportata la scrittura di un parametro Drive; vengono distinti i casi di scrittura positiva ed errata.

A-1) Write Drive Parameter Request

Il pacchetto per la scrittura di un parametro Drive è composto come segue:

DATA TYPE	FIELD	VALUE	MEANING
Byte	Service Code	33hex	Set Drive Parameter - Object Specific Service
See Note 1)	Class ID	66hex	Drive Parameter Access Class Object
	Instance ID	XXXX	Drive Parameter Index in format Low byte-High byte
Byte 2)	VALUE	XX	Low byte-Low word drive parameter value
			High byte-Low word drive parameter value
			Low byte-High word drive parameter value
			High byte-High word drive parameter value

1) Byte o Word in base al tipo di allocazione effettuata dal Master.
2) Il numero di Byte del campo Value dipende dalla lunghezza del parametro Drive; es.: se il parametro Drive è di tipo Integer, la lunghezza di VALUE è 2 Byte.

A-2) Write drive parameter - Reply OK

Se il parametro Drive è stato scritto correttamente, il pacchetto di risposta è:

DATA TYPE	FIELD	VALUE	MEANING
Byte	Service Code	33hex OR 80hex	Set Drive Parameter Reply code- Object Specific Service.
Word	Result	0000	Result field equal to zero means writing correctly executed.

A-3) Write drive parameter - Reply Error

Se la scrittura del parametro Drive viene rifiutata, la risposta è la seguente:

DATA TYPE	FIELD	VALUE	MEANING
Byte	Service Code	33hex OR 80hex	Set Drive Parameter Reply code- Object Specific Service.
Word	Result	XXXX 1	Drive specific error code.

1) Per i codici d'errore vedere tabella 3.4.1.

B) Read Drive Parameter

In questo esempio viene riportata la lettura di un parametro Drive; vengono distinti i casi di lettura positiva ed errata.

B-1) Read Drive Parameter Request

Il pacchetto per la lettura di un parametro Drive è composto come segue:

DATA TYPE	FIELD	VALUE	MEANING
Byte	Service Code	32hex	Get Drive Parameter - Object Specific Service.
See Note 1)	Class ID	66hex	Drive Parameter Access Class Object.
See Note 1)	Instance ID	XXXX	Drive Parameter Index in format Lowbyte-High byte.

1) Byte o Word in base al tipo di allocazione effettuata dal Master.

B-2) Read drive parameter - Reply OK

Se il parametro Drive è stato letto correttamente, il pacchetto di risposta è:

DATA TYPE	FIELD	VALUE	MEANING
Byte	Service Code	32hex	Get Drive Parameter Reply code- Object Specific Service.
Word	Result	0	Result field equal to zero means reading correctly executed.
Byte 1)	VALUE	XX	Low byte-Low word drive parameter value.
			High byte-Low word drive parameter value.
			Low byte-High word drive parameter value.
			High byte-High word drive parameter value.

1) Il numero di Byte del campo Value dipende dalla lunghezza del parametro Drive; es.: se il parametro Drive è di tipo Integer, la lunghezza di VALUE è 2 Byte.

B-3) Read drive parameter - Reply Error

Nel caso in cui la lettura del parametro Drive sia stata rifiutata, la risposta è:

DATA TYPE	FIELD	VALUE	MEANING
Byte	Service Code	32hex	Get Drive Parameter Reply code- Object Specific Service.
Word	Result	XXXX 1	Drive specific error code.

1) Per i codici d'errore vedere tabella 3.4.1.

C) Write Typed Drive Parameter

In questo esempio viene riportata la scrittura di un parametro; vengono distinti i casi di scrittura positiva ed errata.

In questo caso, oltre all'indice ed il valore, viene indicato anche il tipo di dato utilizzato nella trasmissione. L' eventuale conversione di tipo è effettuata automaticamente dal firmware.

C-1) Write Drive Parameter Request

Il pacchetto per la scrittura di un parametro è composto come segue:

DATA TYPE	FIELD	VALUE	MEANING
Byte	Service Code	35hex	Set Drive Parameter - Object Specific Service
See Note 1)	Class ID	66hex	Drive Parameter Access Class Object
	Instance ID	XXXX	Drive Parameter Index in format Low byte-High byte
Byte 2)	DATA TYPE	XX	Value data type
Byte 3)	VALUE	XX	Low byte-Low word drive parameter value
			High byte-Low word drive parameter value
			Low byte-High word drive parameter value
			High byte-High word drive parameter value

1) Byte o Word in base al tipo di allocazione effettuata dal Master.
2) La codifica dei tipi di dato è riportata alla tabella 3.4.2.
3) Il numero di Byte del campo Value dipende dalla lunghezza del parametro Drive; es.: se il parametro Driveè di tipo Integer, la lunghezza di VALUE è 2 Byte.

C-2) Write drive parameter - Reply OK

Se il parametro Drive è stato scritto correttamente, il pacchetto di risposta è:

DATA TYPE	FIELD	VALUE	MEANING
Byte	Service Code	33hex	Set Drive Parameter Reply code- Object Specific Service.
Word	Result	0000	Result field equal to zero means writing correctly executed.

C-3) Write drive parameter - Reply Error

Se la scrittura del parametro Drive viene rifiutata, la risposta è la seguente:

DATA TYPE	FIELD	VALUE	MEANING
Byte	Service Code	33hex	Set Drive Parameter Reply code- Object Specific Service.
Word	Result	XXXX 1)	Drive specific error code.

1) Per i codici d'errore vedere tabella 3.4.1.

D) Read Drive Parameter

In questo esempio viene riportata la scrittura di un parametro; vengono distinti i casi di scrittura positiva ed errata.

In questo caso, oltre all'indice ed il valore, viene indicato anche il tipo di dato utilizzato nella trasmissione. L'eventuale conversione di tipo è effettuata automaticamente dal firmware.

D-1) Read Drive Parameter Request

Il pacchetto per la lettura di un parametro Drive è composto come segue:

DATA TYPE	FIELD	VALUE	MEANING
Byte	Service Code	36hex	Get Drive Parameter - Object Specific Service.
See Note 1)	Class ID	66hex	Drive Parameter Access Class Object.
	Instance ID	XXXX	Drive Parameter Index in format Lowbyte-High byte.
Byte 2)	DATA TYPE	XX	Value data type

1) Byte o Word in base al tipo di allocazione effettuata dal Master.
2) La codifica dei tipi di dato è riportata alla tabella 3.4.2.

D-2) Read drive parameter - Reply OK

Se il parametro Drive è stato letto correttamente, il pacchetto di risposta è:

DATA TYPE	FIELD	VALUE	MEANING
Byte	Service Code	32hex	Get Drive Parameter Reply code- Object Specific Service.
Word	Result	0	Result field equal to zero means reading correctly executed.
Byte 1)	VALUE	XX	Low byte-Low word drive parameter value.
			High byte-Low word drive parameter value.
			Low byte-High word drive parameter value.
			High byte-High word drive parameter value.

1) Il numero di Byte del campo Value dipende dalla lunghezza del parametro Drive; es.: se il parametro Drive è di tipo Integer, la lunghezza di VALUE è 2 Byte.

D-3) Read drive parameter - Reply Error

Nel caso in cui la lettura del parametro Drive sia stata rifiutata, la risposta è:

DATA TYPE	FIELD	VALUE	MEANING
Byte	Service Code	32hex	Get Drive Parameter Reply code- Object Specific Service.
Word	Result	XXXX 1	Drive specific error code.

1) Per i codici d'errore vedere tabella 3.4.1.

Tabella 3.4.1: Codici errore accesso parametri

Codice	Descrizione
1	Numero parametro errato
9	Valore massimo superato
10	Valore minimo superato
11	Valore non ammesso per il parametro
12,13	Parametro in sola lettura
16,31	Parametro non scrivibile con drive abilitato
20	Errore caricamento parametri
21	Errore salvataggio parametri
23	Timeout sul parametro
Altro	Errore generico, contattare l'assistenza

Tabella 3.4.2: Tipi formato parametro

FORMATO	VALORE	SIGNIFICATO
DB_T_VOID	0	Ritorno al valore nel formato originale
DB_T_INT	3	16 bit con segno
DB_T_WORD	6	16 bit senza segno
DB_T_LONG	4	32 bit con segno
DB_T_DWORD	7	32 bit senza segno
DB_T_FLOAT	8	Formato Float IEEE 754

3.5 Operativita Polling

Questo tipo di operatività DeviceNet viene utilizzata per l'interscambio veloce ciclico dei parametri Drive tra il Master e il drive ADV200.
Le caratteristiche dell'operatività Polling sono:

1. La lunghezza del pacchetto è configurabile tramite gli appositi parametri del drive (vedere il menu COMMUNICATION) e può variare da 1 a 16 word per

entrambe le direzioni (Slave->Master e Master->Slave).

2. La scheda, in qualità di Slave, consuma in Polling dati di Output e produce come risposta dati di Input.

La configurazione dei parametri Drive scambiati in Polling è impostabile utilizzando i parametri di configurazione disponibili nell'azionamento. (vedere il menu COMMUNICATIONS).

3.6 Configurazione dell'interfaccia Devicenet

La configurazione dell'interfaccia DeviceNet avviene tramite parametri drive. I parametri sono gestiti con menu gerarchici. Tutti i parametri di scrittura relativi all'interfaccia DeviceNet sono attivi solo dopo il reset del drive. Di seguito sono riportati i parametri drive per la gestione dell'interfaccia DeviceNet.

Per attivare la scheda EXP-CAN-ADV impostare il parametro PAR 4000 **Fieldbus type** come "DeviceNet".
I seguenti parametri sono disponibili nel menu COMMUNICATION->FIELDBUS CONFIG :

PAR	Nome Par	tipo	Valore default	Attr
4004	Fieldbus baudrate	Enum	None	Scrittura
4006	Fieldbus address	2 byte senza segno	0	Scrittura
4010	Fieldbus M->S enable	Enum	On	Scrittura
4012	Fieldbus alarm mode	2 byte senza segno	0	Scrittura
4014	Fieldbus state	Enum	Stop	Sola lettura

- Fieldbus baudrate = Imposta il baudrate della rete. Valori disponibili per DeviceNet : 125k , 250k , 500k
- Fieldbus address = indirizzo di questo nodo slave nella rete , valori ammessi da 1 a 63
- Fieldbus M->S enable = se messo a Off i dati Polling da master a slave non vengono gestiti
- Fieldbus alarm mode = se messo a 1 il drive genera errori Opt Bus Fault relativi alla perdita di comunicazione (Bus Loss) anche quando il drive non è abilitato.
- Fieldbus state = stato della comunicazione per questo nodo su rete DeviceNet
  - Stop : Nessuna comunicazione con il master
  - Pre-Operational :Il Riconoscimento da parte del master è in corso
  - Operational : Polling I/O attivo

3.7 Allarmi

3.7.1 Allarmi DeviceNet

I malfunzionamenti del bus sono segnalati mediante l'allarme **Opt Bus Fault** failure. Nel caso di DeviceNet, le cause possibili di anomalie sono:

- stato Bus-off della lina CAN;
- abilitazione del drive in uno stato diverso da Polling I/O
- superamento della soglia di timeout della connessione.

Questo allarme si attiva solo quando il drive è abilitato.
Il parametro PAR 4014 **Fieldbus alarm mode**, se posto a ON, abilita la generazione dell'allarme **Opt Bus Fault** anche quando il drive è disabilitato.

Codice	Cfg	Descrizione	Azioni
0		Bus Loss	Check line for noise, terminations, problems with cabling
FF01	*	Fieldbus type does not match expansion card	Verify if EXP-CAN-ADV card is properly installed
FF04	*	Error initializing CAN interface	Internal error, contact manufacturer
FF24..FF33	*	More than 1 Src pointing to M2S Channel n	Check for multiple destinations on “Fieldbus M->Sn Dest”
FF34..FF43	*	M2S Channel n , data size is wrong (16 bits on 32 bits or 32 bits on 16 bits parameter)	Check “Fieldbus M->Sn sys”
FF44..FF53	*	Invalid parameter in channel S2M n	Check “Fieldbus S->Mn src”
FF54..FF63	*	S2M Channel n, data size is wrong (16 bits on 32 bits or 32 bits on 16 bits parameter)	Check “Fieldbus S->Mn sys”
FF74..FF83	*	M2S Channel n : too many words in PDC	“Fieldbus M-Sn dest” & “Fieldubs M->Sn sys” address more than 16 words in PDC
FF84..FF93	*	S2M Channel n : too many words in PDC	“Fieldbus S->Mn src” & “Fieldubs S->Mn sys” address more than 16 words in PDC
FFB4..FFC3	*	Internal database error on channel n	Internal error, contact manufacturer
8110		CAN msg overflow	Too many packets for selected baudrate
FFC5		Wrong baudrate	Wrong baudrate
FFC6		Wrong MacID	Check “Fieldbus address” is 1 to 63
FFC7		CAN bus off	Check line state for problems
FFC8		System error on connection	Check master for proper connection
FFC9		Duplicate MacID Check failed	Check “Fieldbus address” is unique in the network

3.7.2 Gestione Allarmi Drive

Non è prevista la funzionalità di stato allarmi del drive.
Nessun trattamento speciale viene quindi fatto per lo stato degli allarmi del drive.
Il firmware ADV200, mette a disposizione una serie di parametri per rilevare lo stato del drive. Fare riferimento al manuale del drive per ulteriori informazioni.

3.8 Controllo del Process Data Channel

Questa funzione permette l’assegnazione di parametri o variabili applicative dell’azionamento ai dati del Process Data Channel.
L’interfaccia DeviceNet per ADV200 utilizza un numero impostabile di word per il Process Data Channel (abbr. PDC Process Data Channel).
La configurazione del Process Data Channel per l’interfaccia DeviceNet è la seguente:

DATO 0 DATO... DATOn

Lo Slave è in grado sia di leggere che di scrivere i dati del Process Data Channel.
I dati DeviceNet letti dallo Slave sono indicati come dati d’ingresso; i dati scritti in

DeviceNet dallo Slave sono indicati come dati d'uscita.
Un dato può essere sia di 2 che di 4 byte. Con il termine dati , si intende un numero qualsiasi di questi ultimi compreso tra 0 e 16, purchè il numero totale di byte richiesti non sia superiore a 32byte.

Esempio
Si possono avere :
- da 0 a 16 dati da 2 byte
- 1 dato a 4 byte + da 0 a 14 dati da 2 byte
- 2 dati da 4 byte + da 0 a 12 dati da 2 byte
...
- 8 dati da 4 byte

I dati scambiati mediante il PDC possono essere di due tipi:
-parametri dell'azionamento
-variabili di un'applicazione MDPlc

La composizione dei dati in ingresso e uscita del PDC è definita con opportuni parametri come descritto nei paragrafi 3.8.1 e 3.8.2.
Il master scrive ciclicamente i dati definiti come PDC input e legge ciclicamente i dati definiti come PDC output.

3.8.1 Configurazione PDC Input (Parametro SYS_FB_XXX_MS)
Vedere paragrafo 1.3.1.

3.8.2 Configurazione PDC Output (Parametro SYS_FB_XXX_SM)
Vedere paragrafo 1.3.2.

3.8.3 Configurazione I/O Virtuali Digitali
Il firmware ADV200, fornisce la funzionalità di Word Comp e Word Decomp Virtual Digital I/O, che consente lo scambio di segnali discreti tra master e slave e viceversa.
È possibile inviare comandi al drive attraverso le funzionalità del PAR 4452 **Word decomp**. Il significato dei singoli bit è programmabile. Può essere impostato su un canale "Fieldbus M->Sn" come Count 16.
Lo stato del drive viene letto nel PAR 4432 **Word Comp**, programmabile su qualsiasi canale "Fieldbus S->Mn" come Count 16. Il significato di ogni singolo bit può essere scelto dall'utente attraverso i PAR 4400 **Word Bit 0 src** ... PAR 4430 **Word Bit 15 src**.
Per una descrizione dettagliata di questi parametri, vedere il manuale del drive.

3.8.4 Uso del PDC in Applicazioni MDPlc
Vedere par. 1.3.3 Uso del PDC in Applicazioni MDPlc

3.9 Esempio di programmazione

Vedere capitolo 1.6.

Nel paragrafo 1.6.2.2 : P4000 **Fieldbus Type** = DeviceNet.

Dimensioni in ingresso e uscita dell'area di "Polling I/O" : con l'esempio contenuto nei par. 1.6.1.1 e 1.6.1.2 si ha una dimensione di 4 byte in scrittura e 6 in lettura.

Riferimenti

CiA :	CAN in Automation, gruppo internazionale utilizzatori.
CAN :	Controller Area Network.
PDO:	Process Data Object, messaggi di servizi senza conferma usati per trasferimento dati da/per dispositivo in tempo reale.
DBT:	Distributor. E' un elemento di servizio del CAN Application Layer nel Modello di Riferimento CAN; è compito del DBT distribuire COB-ID ai COB che sono usati dal CMS.
SDO:	Service Data Object, messaggi di servizi con conferma usati per trasferimento dati aciclico da/per dispositivo.
NMT:	Network Management. E' un elemento di servizio del CAN Application Layer nel Modello di Riferimento CAN; esegue inizializzazione, configurazione e gestione errori di una rete CAN.
CS:	Command Specifier; definisce il servizio NMT.
COB-ID	COB-Identifier. Identifica univocamente un COB all'interno della rete; esso determina anche la priorità del COB.