Manuale di Istruzione

Inserto programmabile del sistema di amplificatori MGC*plus*

ML70B

A2000-1.0 it

HBM

Contenuto **Pagina**

Note sulla sicurezza

Uso appropriato

L'inserto programmabile ML70B può essere usato esclusivamente per compiti di misurazione e per quelli di controllo ad essi associati. Qualsiasi altro impiego non verrà considerato appropriato.

Per garantire il funzionamento in sicurezza, lo strumento deve essere usato secondo le specifiche descritte nel manuale di istruzione. Inoltre, è essenziale attenersi alle disposizioni di sicurezza ed ai regolamenti concernenti l'applicazione specifica. Quanto detto vale anche per l'impiego degli eventuali accessori.

Rischi generici non applicando le note sulla sicurezza

L'inserto ML70B corrisponde allo stato attuale della tecnica ed è di funzionamento sicuro. Tuttavia, l'inadeguata installazione e manovra da parte di personale non addestrato può comportare rischi residui.

Chiunque sia incaricate dell'installazione, messa in funzione, manutenzione o riparazione dello strumento, deve aver letto e compreso il manuale di istruzione, specialmente per la parte concernente le note sulla sicurezza.

Rischi residui

Le caratteristtiche ed il corredo di fornitura dell'ML70B coprono solo una parte del campo della tecnologia di misura. I progettisti, gli installatori ed i conduttori degli impianti devono inoltre progettare, realizzare e rispondere delle considerazioni ingegneristiche della tecnica di misura, al fine di minimizzare i rischi residui. Si deve sempre adempiere ai regolamenti preesistenti. I rischi residui concernenti la tecnologia di misurazione devono essere notificati.
Se operando con l'ML70B dovessero sussistere rischi residui, essi sono evidenziati in questo manuale dai seguenti simboli:

Simbolo:  **AVVERTIMENTO**

Significato: **Possibile situazione di pericolo**

Segnala una **possibile** situazione di pericolo, che – se non vengono rispettate le disposizioni di sicurezza – **può avere** come conseguenza gravi ingiurie corporali o la morte.

Operare con cognizione della sicurezza

I messaggi di errore possono essere annullati solo se la loro causa è stata rimossa e non sussiste più alcun pericolo.
Lo strumento adempie ai requisiti sulla sicurezza della DIN EN 61010-Parte1 (VDE 0411-Parte1); Classe di protezione I.

Per garantire la sufficiente immunità ai disturbi, usare esclusivamente il metodo di schermatura *Greenline* (vedere la pubblicazione HBM ”Concetto di schermatura *Greenline*”, cavi di misura idonei EMC, G36.35.0).

Modifiche e variazioni

Senza il nostro espilicito consenso l'inserto ML70B non può essere modificato ne strutturalmente che nella tecnologia di sicurezza. Qualsiasi modifica fa decadere la nostra responsabilità per gli eventuali danni che ne derivano.

In particolare è proibita qualsiasi riparazione e lavoro di saldatura sulla scheda madre. Per sostituire moduli completi si deve usare solo marteriale originale della HBM.

Personale qualificato

Questo strumento può essere installato ed usato solo da personale qualificato e che si attenga scrupolosamente ai dati tecnici ed ai regolamenti e requisiti di sicurezza sotto elencati.
Per il suo uso bisogna inoltre osservare le direttive legali e quelle sulla sicurezza concernenti l'applicazione da effettuare. Per gli eventuali accessori vale quanto sopra affermato.

Per personale qualificato si intendono le persone che abbiano esperienza con l'installazione, montaggio, messa in funzione e conduzione del prodotto e che per questa attività abbiano conseguito la corrispondente qualifica.

I lavori di manutenzione e riparazione su strumenti aperti e sotto tensione possono essere effettuati solo da personale addestrato, il quale sia consapevole dei rischi a cui è soggetto.

1 Introduzione

L'inserto programmabile ML70B è un modulo largo 4 unità atto ad occupare uno dei posti del sistema di strumenti MGC*plus*. L'inserto è liberamente programmabile col sistema di programmazione “CoDeSys”, secondo lo standard internazionale IEC61131−3 per PLC.

Cosa può fare l'ML70B?

- Gestire i valori misurati di qualsiasi altro inserto dell'MGC*plus* con cadenza di 2400 Hz, compresi i dati provenienti dal CANbus o Profibus, che possono essere letti dal sistema.
- Inviare comandi agli altri amplificatori.
- Trasmettere valori calcolati come fossero valori di misura dell'MGC*plus*, che possono essere mostrati dall'Indicatore/Terminale AB22A e che possono essere visualizzati ed elaborati dall'MGCplus-Assistant.
- Controllare fino a due pannelli di collegamento del tipo AP71, AP72, AP75, AP78 (per le combinazioni possibili vedere pagina 9).

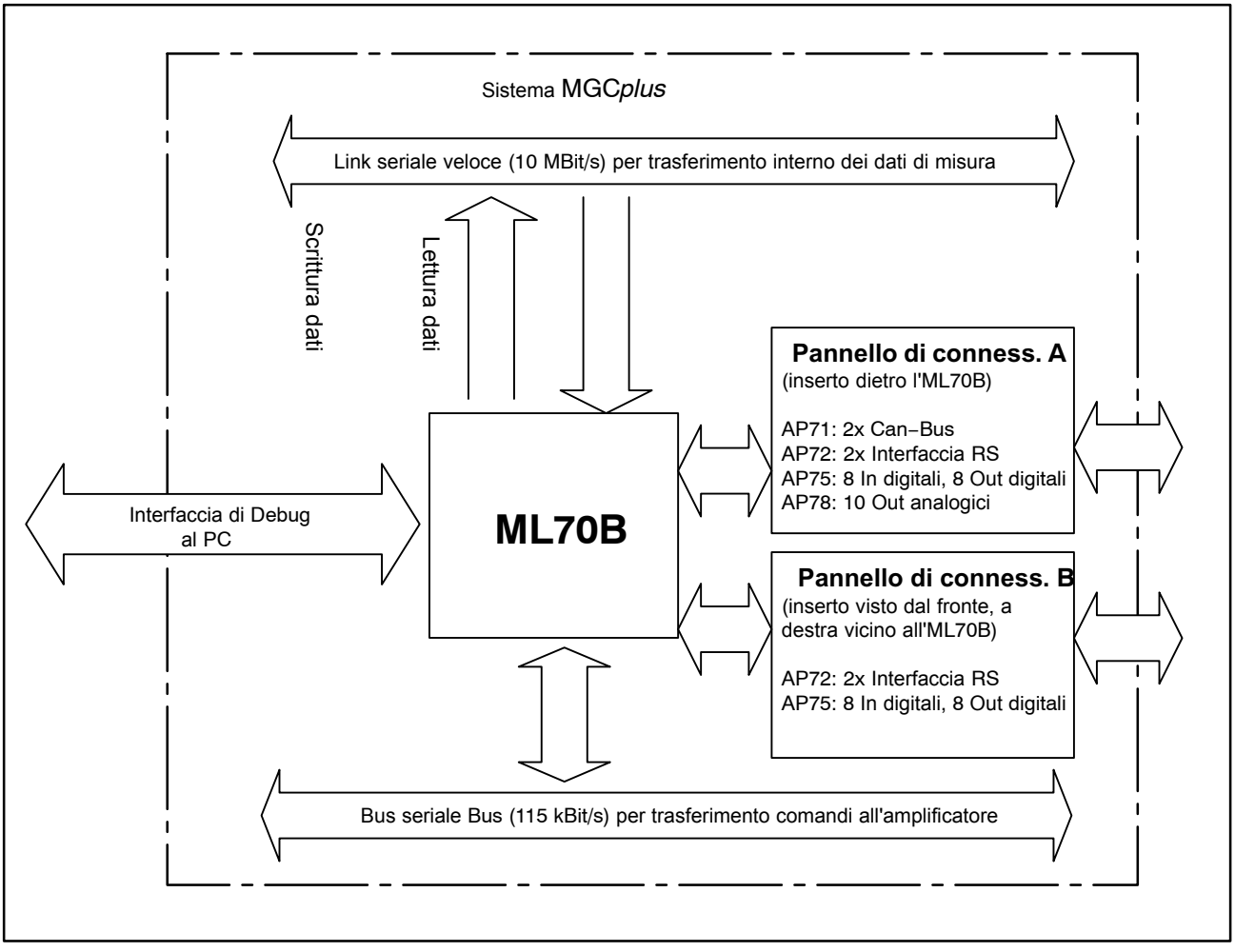
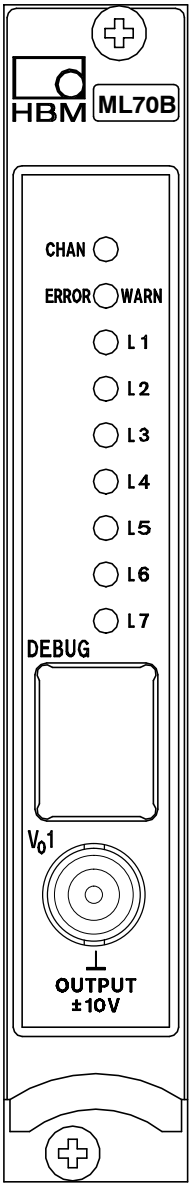


Fig.1.1: ML70B nel sistema MGC*plus*

2 Pannello frontale



Iscrizione	Colore	Significato
CHAN.	giallo	Selezione canale
ERROR/WARN	rosso	Errore
L1	rosso, giallo	secondo programmazione
L2	rosso, giallo	secondo programmazione
L3	rosso, giallo	secondo programmazione
L4	rosso, giallo	secondo programmazione
L5	rosso, giallo	secondo programmazione
L6	rosso, giallo	secondo programmazione
L7	rosso, giallo	secondo programmazione

I LED L1... L7 sono programmabili (vedere pagina 30).

3 Pannelli di connessione

L'ML70B può controllare fino a due dei seguenti pannelli di connessione:

AP71 (2x CAN–Bus)

AP72 (2x RS-232/RS-485/RS-422; selezionabili singolarmente via software)

AP75 (8x In digitali, 8x Out digitali)

AP78 (8x Out analogici)

Ciascuno dei pannelli sopra elencati può essere controllato singolarmente. Sono possibili le seguenti combinazioni:

AP71, AP72
AP71, AP75
AP72, AP72
AP72, AP75
AP75, AP72
AP75, AP75
AP78, AP72
AP78, AP75

I pannelli di connessione possono essere innestati direttamente dietro all'inserto amplificatore, oppure di fianco ad esso (vista da sopra, figura 3.1).

NOTA

Solo il pannello innestato dietro all'inserto ML70B dispone delle tensioni di uscita analogiche V_{O1} e V_{O2}!

Le uscite del pannello di connessione AP78 sono contrassegnate con **AO** (Analogue Output = Uscita Analogica) nel menu di impostazione.

Se il pannello di connessione AP75 è inserito direttamente dietro l'inserto, gli ingressi/uscite sono contrassegnati con "**A.**". Se invece esso è inserito davanti a destra dell'inserto, gli ingressi/uscite sono contrassegnati con "**B.**".

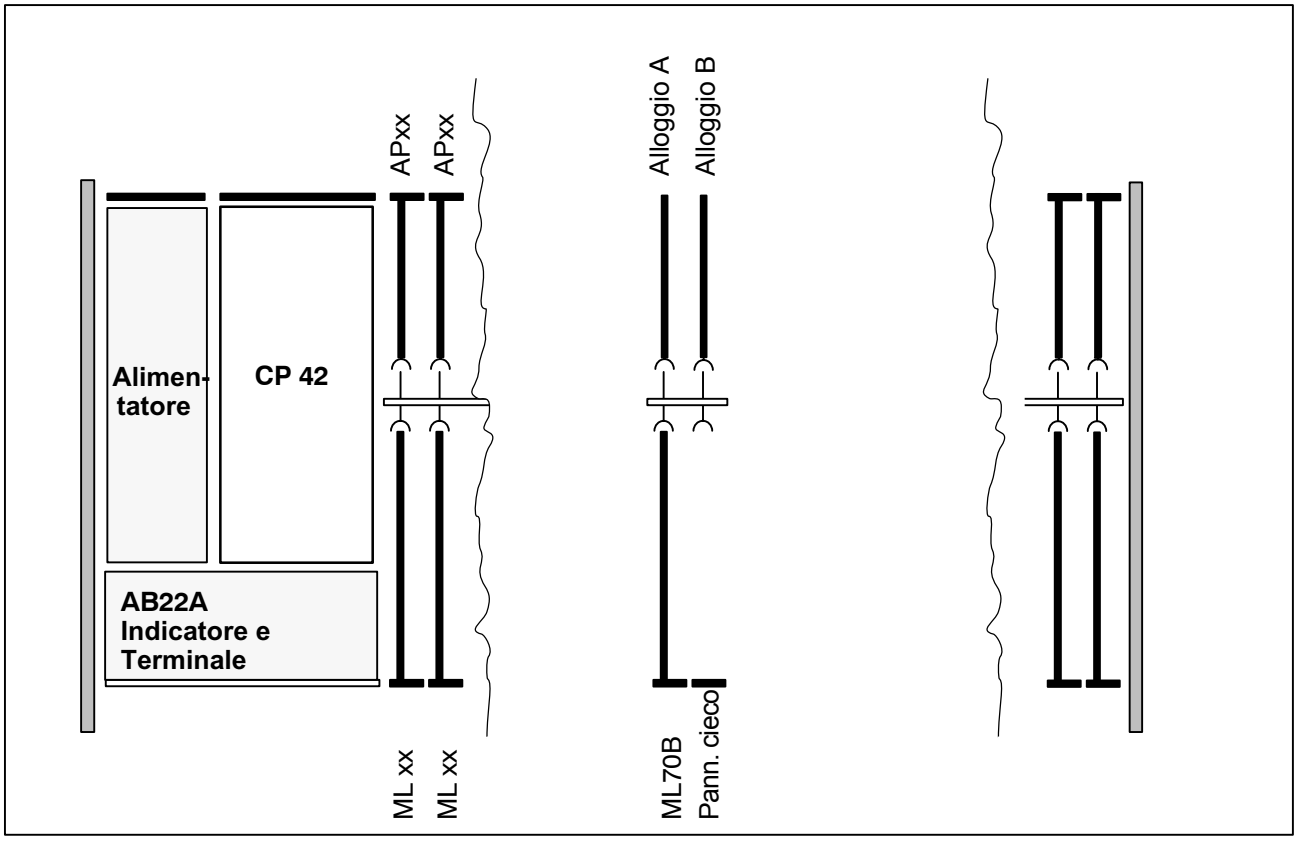
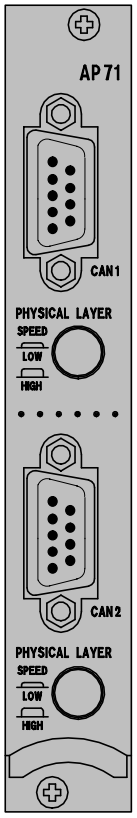
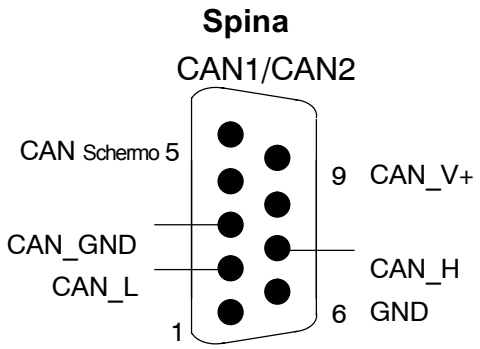


Fig. 3.1: Combinazione dei pannelli di connessione nel sistema strumenti (vista da sopra)

3.1 Pannello di connessione AP71





Physical Layer:

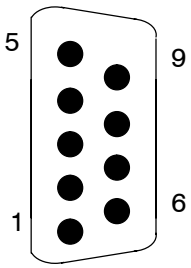
High: Standard CAN - Velocità fino a 1 MBaud secondo ISO11898 (rigidità dielettrica 24V).

Low: Bus Comfort per comunicazione nei veicoli, fino a 100 kBaud (CAN_V+ deve essere collegato ai 12V)

Fig. 3.2: Collegamenti alla spina dell'AP71

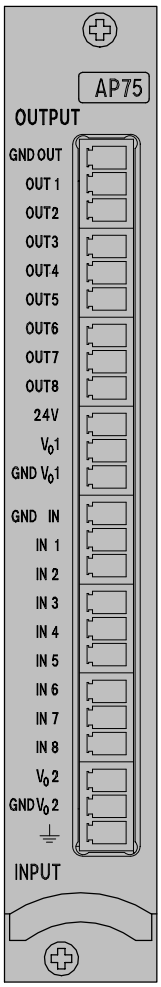
3.2 Pannello di connessione AP72

Spina



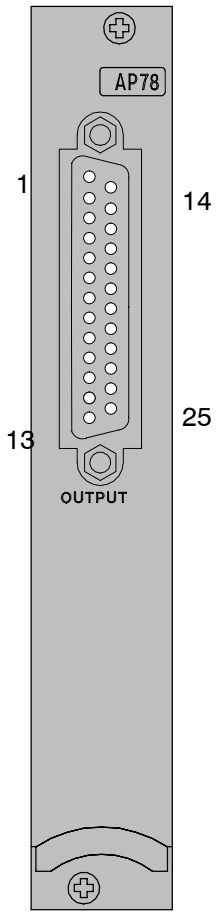
Pin	Funzione RS232	Funzione RS422	Funzione RS485
1	CD (non usato)	–	–
2	RXD – Receive Data	RXA – Receive Data A (non invertito)	–
3	TXD – Transmit Data	TXB – Transmit Data B (invertito)	RXB–Receive Data B (invertito)
4	DTR –12 V tramite resistori	–	–
5	GND	GND	GND
6	DSR (non usato)	–	–
7	RTS – Request To Send	TXA–Transmit Data A (non invertito)	RXA–Receive Data A (non invertito)
8	CTS – Clear To Send	RXB–Receive Data B (invertito)	–
9	RI (non usato)	–	–

3.3 Pannello di connessione AP75



Il pannello di connessione AP75 possiede otto ingressi digitali ed otto uscite digitali. Tutti gli ingressi e le uscite sono individualmente isolate galvanicamente ed hanno il proprio sistema di massa (GND OUT: massa per le uscite; GND IN: massa per gli ingressi).

3.4 Pannello di connessione AP78



Pin	Funzione
1	Uscita analogica del pannello di connessione AO3
2	Uscita analogica del pannello di connessione AO4
3	Uscita analogica del pannello di connessione AO5
4	Uscita analogica del pannello di connessione AO6
5	Uscita analogica del pannello di connessione AO7
6	Uscita analogica del pannello di connessione AO8
7	Uscita analogica del pannello di connessione AO9
8	Uscita analogica del pannello di connessione AO10
9	–
10	–
11	Uscita analogica dell'amplificatore VO1
12	Uscita analogica dell'amplificatore VO2
13	–
14	GND per AO3
15	GND per AO4
16	GND per AO5
17	GND per AO6
18	GND per AO7
19	GND per AO8
20	GND per AO9
21	GND per AO10
22	–
23	–
24	GND per VO1
25	GND per VO2

4 Modo operativo dell' ML70B

4.1 Risposta nel tempo

L'ML70B ha i seguenti compiti:

- lettura del valore di misura,
- esecuzione del programma caricato,
- controllo dei pannelli di connessione,
- comunicazione con i componenti dell'MGC*plus* (CP42, canali amplificatori, calcolatore remoto, ecc.)

Questi compiti vengono eseguiti ciclicamente (vedere figura 4.1).
La comunicazione si svolge continuamente sullo sfondo.

La lettura dei valori di misura è controllata da interrupt alla frquenza di 2400 Hz. Alla stessa cadenza i dati di misura entrano nell'MGC*plus*.
Dopo la lettura del valore di misura vengono controllati i pannelli di connessione e viene eseguito il programma utente.

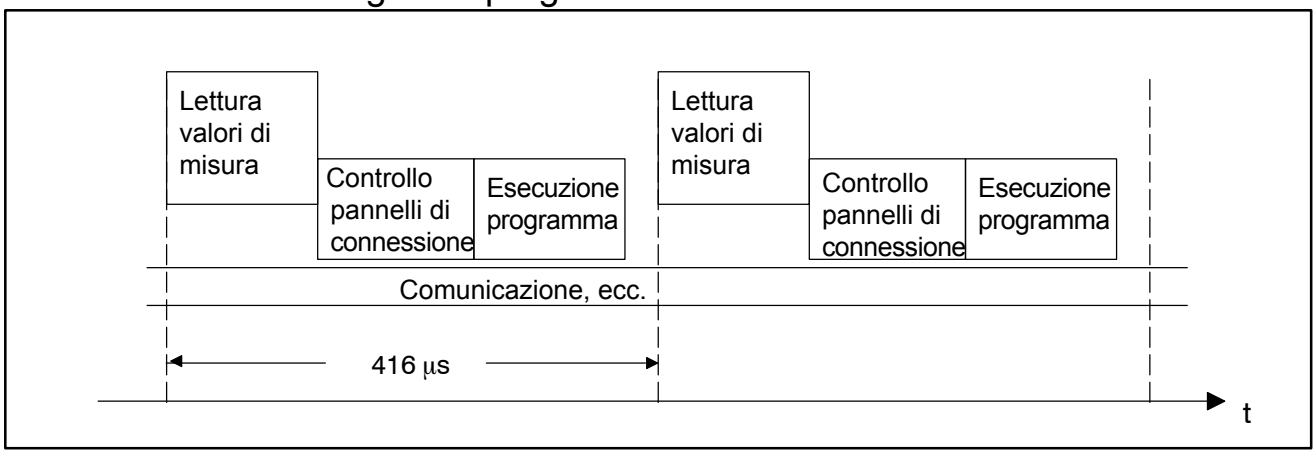


Fig. 4.1: Ciclo temporale dell'ML70B

Si raggiunge la massima frequenza di chiamata di 2400 Hz solo allorché il programma IEC viene svolto in meno di 400 µs (1/2400 Hz). Se dura un poco di più,la frequenza di chiamata si riduce di conseguenza (la successiva frequenza minore è 1200 Hz).

Di conseguenza, si dovrebbero formulare i programmi in modo tale che essi si chiudano il più rapidamente possibile in qualsiasi situazione. È questa la caratteristica che distingue i programmi per PLC dai programmi “normali”.

4.2 Trasmissione dei valori di misura

I moduli amplificatori (ML ...) ed il processore di comunicazione (CP ...) dell'MGC*plus* sono connessi con una veloce interfaccia dati sincrona (Link), con cui vengono scambiati i valori di misura nel sistema. Lo scambio dei dati avviene alla frequenza di 2400 Hz.

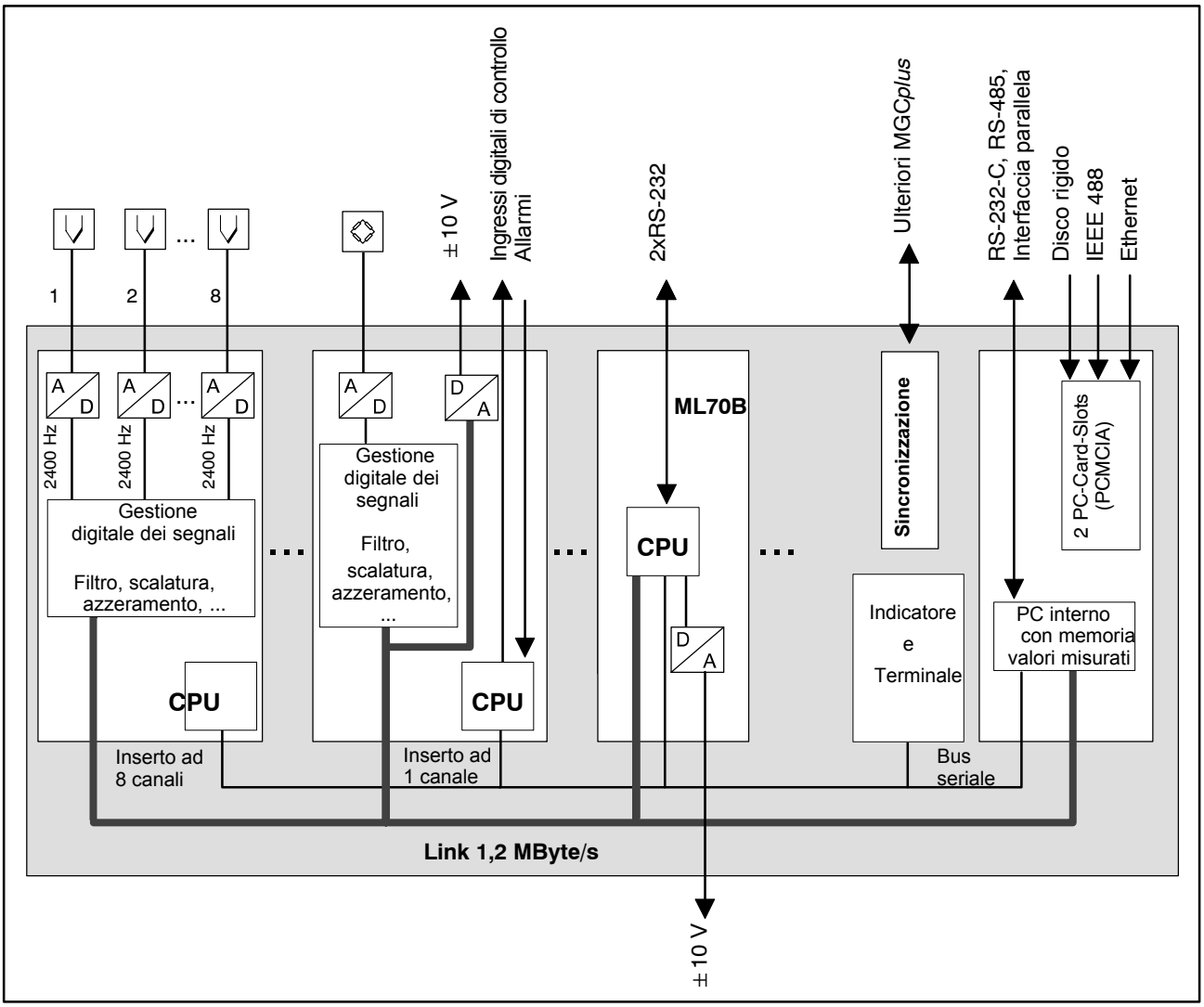


Fig. 4.2: Comunicazione all'interno dell'MGCplus

Ogni inserto può far uscire complessivamente 8 valori di misura con cadenza di trasferimento di 2400 Hz. Per un amplificatore monocanale ciò significa che possono venir trasferiti più segnali contemporaneamente (lordo, netto, valore di picco ...). Per un amplificatore pluricanale, può essere trasferito solo un segnale per ogni sottocanale, dato che complessivamente ne possono essere inoltrati otto. Per esempio, da un amplificatore pluricanale non possono essere richiesti contemporaneamente un valore lordo ed uno netto da un sottocanale.

Se i valori di misura vengono richiesti da più componenti dell'MGCplus, deve essere stabilita la sequenza di chiamata. Il processore di comunicazione CP42 gestisce il trasferimento dei dati ed è ad esso che si devono richiedere i valori di misura necessari. Ogni componente che necessiti dei valori di misura deve riferirsi a questo processore di comunicazione.

Si possono causare dei conflitti quando, per esempio, vengono richiesti due segnali contemporaneamente da un amplificatore ad 8 canali. In questo caso si deve annullare la seconda richiesta zittendo l'avviso di errore.

Fondamentalmente per la chiamata delle risorse Link vale:
fintanto che le risorse sono disponibili esse vengono allocate e, precisamente, allocate nella sequenza stessa delle richieste.

4.3 Struttura del programma

Per le ragioni precedentemente specificate, il programma ML70B è costituito dai seguenti stadi:

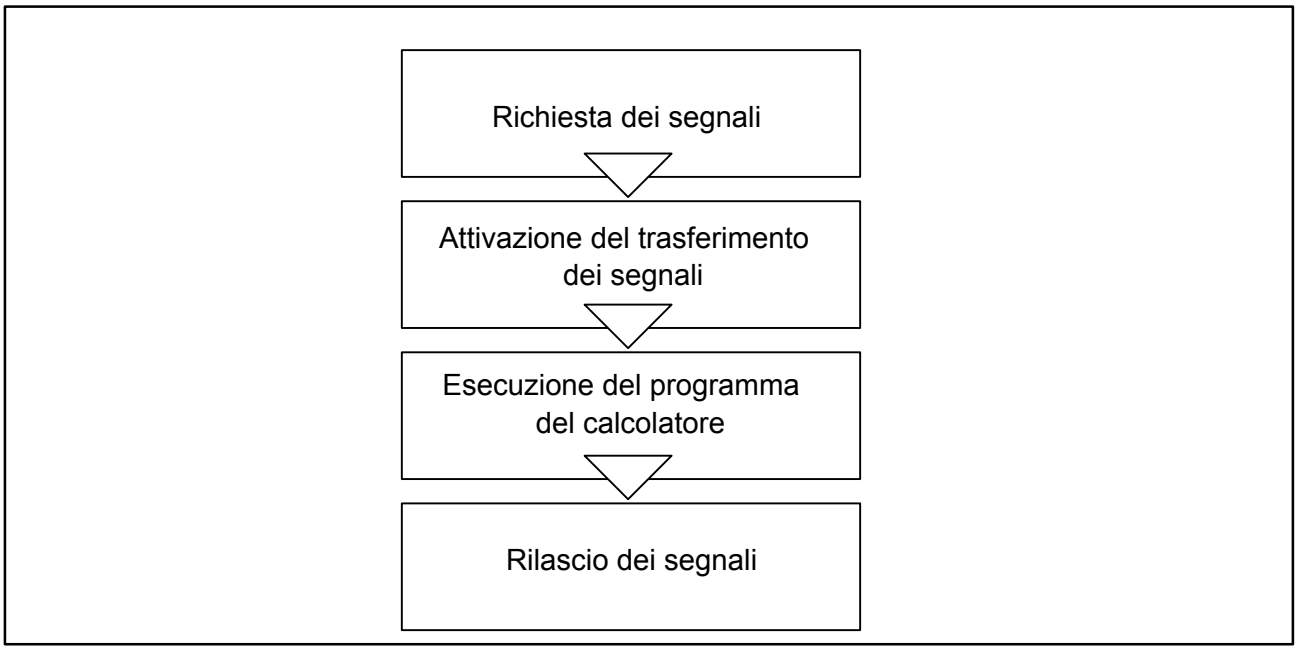


Fig. 4.3: Struttura del programma

Questa strutturazione del programma è la medesima per tutti i progetti in cui i valori misurati devono essere gestiti dall'MGC*plus*. Consigliamo di strutturare tutti i programmi di acquisizione come descritto nel paragrafo 4.5.

4.4 Salvataggio del programma

Il programma viene eseguito nella memoria volatile interna (RAM) dell'ML70B.

All'accensione dell'MGC*plus*, esso verifica se nella memoria volatile (FLASH) ci sia un programma. Se lo trova, il programma viene automaticamente copiato dalla memoria FLASH in quella volatile, indi eseguito.

Il sistema di programmazione può caricare un altro programma nell'ML70B e lo puo provare, senza modificare il programma nella memoria permanente. Fino a quando il contenuto della RAM non viene copiato nella FLASH,

spegnedo e riaccendendo l'MGC*plus*, l'ML70B esegue il vecchio programma della FLASH (vedere anche il capitolo 10).

Procedura per cancellare il contenuto della FLASH:

1. spegnere lo strumento MGCplus,
2. estrarre l'inserto ML70B dalla custodia. Nella parte sottostante dello inserto si trova un connettore a pattine a 10 poli,
3. inserire un cavallotto fra il Pin 9 ed il Pin10,
4. riaccendere lo strumento MGCplus, si cancella la memoria Flash,
5. rispegnere lo strumento MGCplus indi togliere il ponticello.

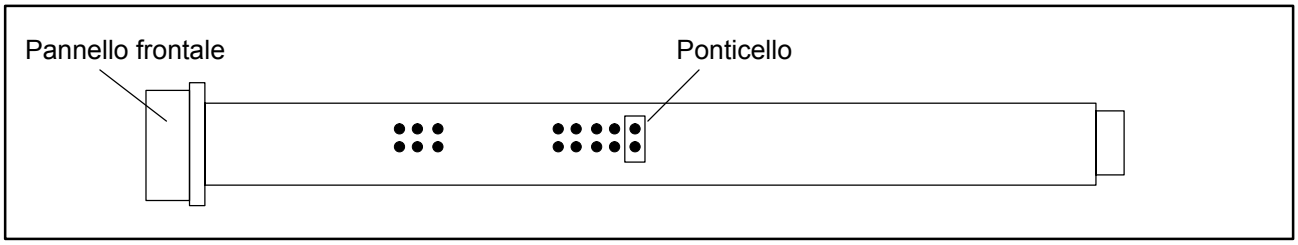


Fig. 4.4: Inserto- amplificatore ML70B; vista da sotto

Volendo salvare permanentemente un nuovo programma nell'ML70B, esso deve essere copiato dalla RAM nella FLASH-ROM (vedere CoDeSys “Boot project generation”).

4.5 Il programma principale PLC_PRG

L'ML70B chiama il programma principale PLC_PRG sincronicamente con i valori di misura. Ogni programma dovrebbe essere strutturato col seguente schema in linguaggio di programmazione SFC (sequential function chart):

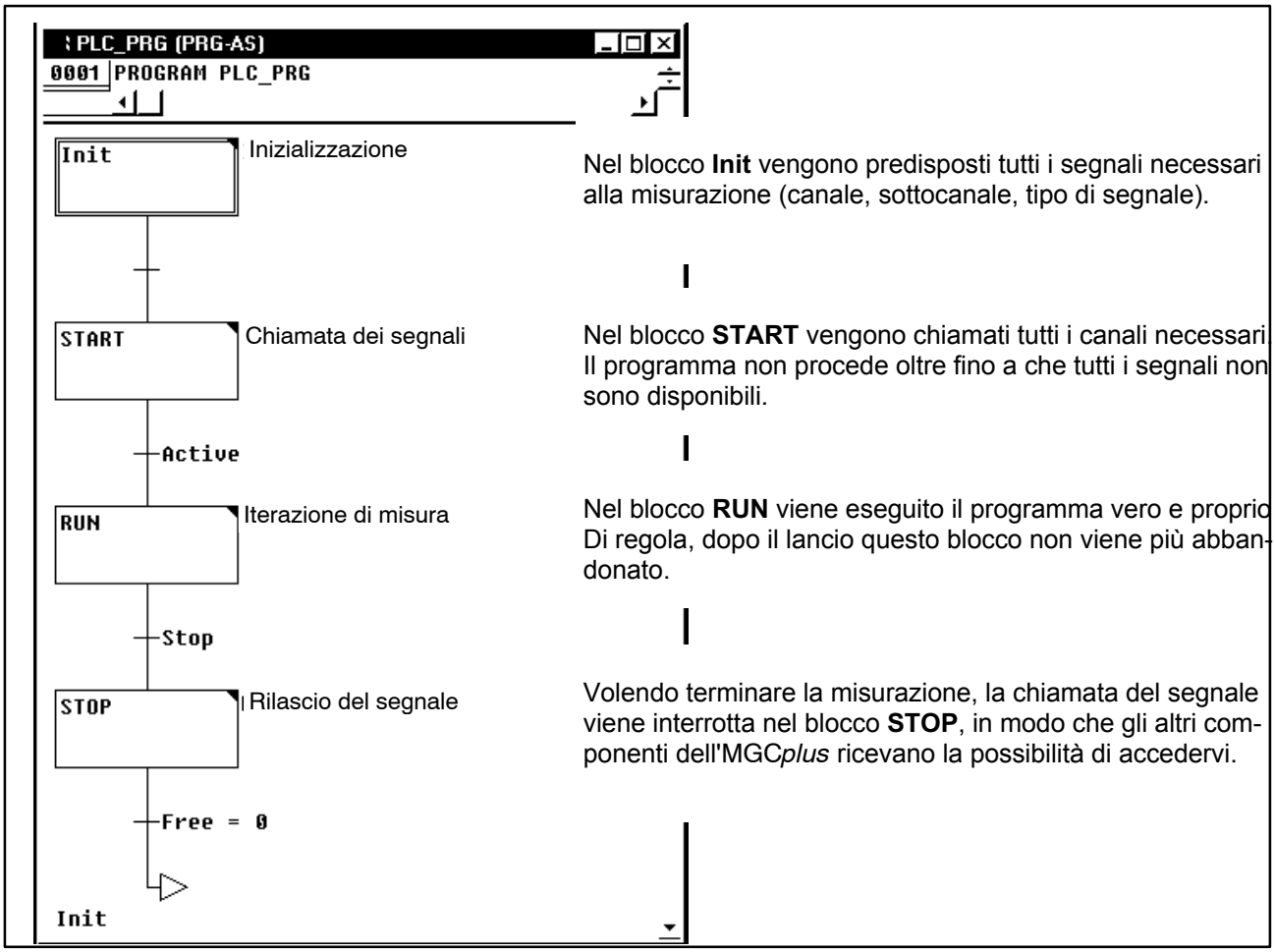


Fig. 4.5: Svolgimento programma nell'ML70B

Con questo schema si stabiliscono soltanto la richiesta ed il rilascio dei valori di misura. Le effettive funzioni del programma sono contenute nei singoli blocchi.

Il contenuto di ogni blocco può essere formulato in uno qualsiasi dei linguaggi IEC61131−3, e viene aperto con un doppio click sul blocco stesso.

NOTA

Il precedente schema non deve essere interpretato come un “normale” diagramma di flusso! Ad ogni chiamata del programma viene eseguito solo un singolo blocco e, al termine, si esce dal programma. Se la condizione di transizione fra i due blocchi risulta soddisfatta, alla successiva chiamata di programma viene eseguito il blocco seguente. Se la condizione non risulta soddisfatta, alla successiva chiamata viene ripetuto il medesimo blocco.

5 Introduzione alla programmazione

Per spiegare il sistema di programmazione CoDeSys e l'uso dell'ML70B, in questo capitolo viene impiegato un semplice esempio di programmazione. La descrizione dettagliata del sisstema di programmazione si trova sul CD Documentazione online CoDeSys.

Requisisti del sistema:

- Strumento MGC*plus* con:

  AB22A

  CP42 (opzionale)

  Canale1: ML55B, ML01B o simili (amplificatore monocanale)

  Canale2: ML55B, ML01B o simili (amplificatore monocanale)

  Canale3: ML70B con/senza pannello di connessione
- PC con sistema opeartivo Windows ed interfaccia seriale

5.1 Installazione del sistema di programmazione

1. Inserire il CD CoDeSys nel drive CD−ROM del calcolatore.
   Il Setup parte automaticamante. Se ciò non dovesse avvenire, lanciare manualmente il file **Setup.exe** del CD, con un doppio click.
2. Scegliere la lingua di Setup desiderata e cliccare su ***OK.***
3. Cliccare su ***Continue*** per chiudere la finestra ”Welcome...”.
4. Nella finestra di selezione del tipo di setup attivare ***Development system*** e cliccare su ***Continue.***
5. Nella finestra di selezione dei componenti dovrebbero essere attivi tutti i Checkbox (impostazione presunta). Mantenere le impostazioni delle cartelle di Target e cliccare su ***Continue.***

6. Scegliere la lingua dell'ambiente di programmazione e cliccare su ***Continue.***
7. Scegliere la cartella desiderata e cliccare su ***Continue.***

Dopo la vista del sommario segue l'installazione del sistema di sviluppo.
Il sistema di Target ML70B viene installato automaticamente con i dischi CD di corredo. Tuttavia si può installare il sistema Target anche in un secondo momento:

1. Usare ***Start*** → ***Programme*** → ***CoDeSys V2.2*** per lanciare il programma ***InstallTarget***.
2. Usare il bottone ***Open..*** per aprire il file <CD-Drive>/Hottinger/install/Hottinger.tnf.
3. Selezionare l'assegnazione “Hottinger Baldwin Messtechnik GmbH” nella finestra di sinistra e premere il bottone ***Install*****.** Rispondere con "Yes" alla domanda ”Installation directory does not exist...?”.

Nella finestra di destra **Installed target systems** dovrebbe ora apparire l'assegnazione “Hottinger Baldwin Messtechnik GmbH”.
Ciò conclude l'installazione.

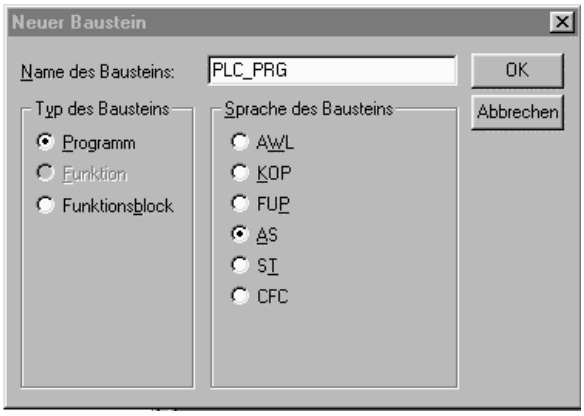
5.2 Esempio di programmazione

Per calcolare la potenza di un motore bisogna moltiplicare la coppia misurata con la velocità angolare misurata. Il risultato deve apparire sull'indicatore dello strumento MGC*plus* oppure in Assistant.
Formula per il calcolo della potenza:

$$Potenza\ [W] = Coppia\ [N \cdot m] \cdot Velocità \left[\frac{1}{\min}\right] \cdot \frac{2 \cdot \pi}{60}$$

1. Lanciare il sistema di programmazione CoDeSys mediante ***Start*** → ***Programs*** → ***CoDeSys V2.2.***
2. Usare ***File*** → ***New*** per generare un nuovo progetto.
3. Selezionare il sistema Target *HBM_ML70B* e confermare con ***OK*** (si apre automaticamente la finestra *NewPOU*).

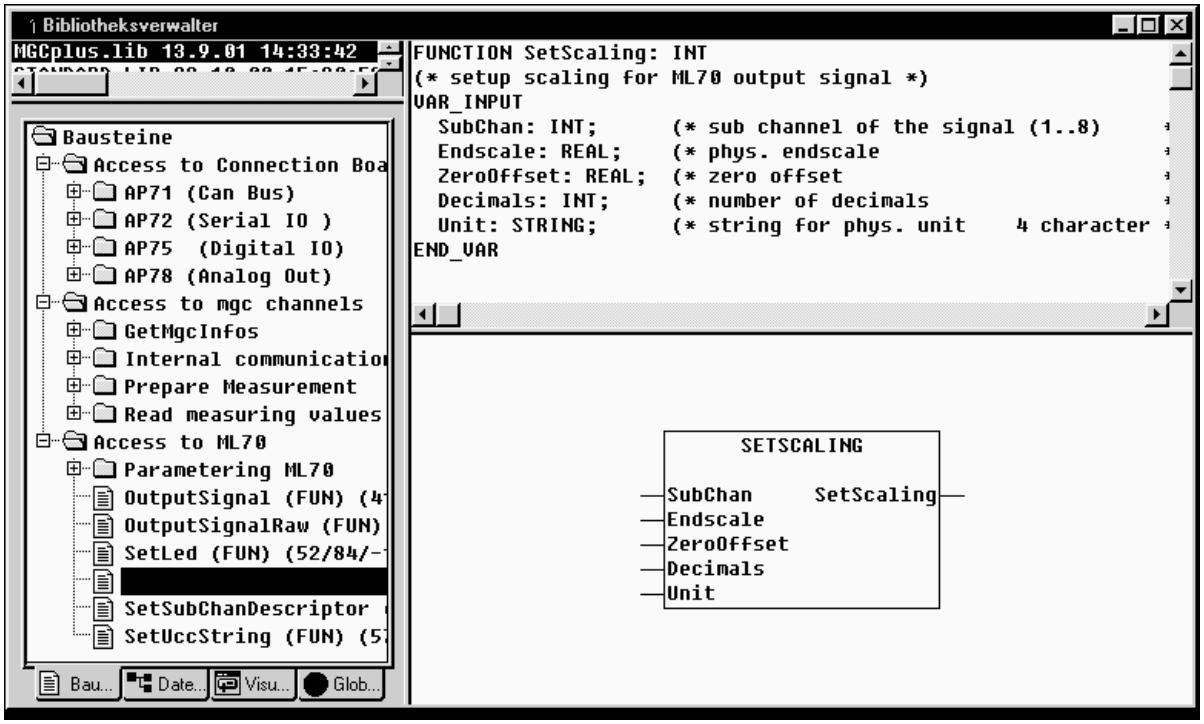
4. Selezionare l'oggetto *PLC_PRG* dal tipo *Program* del linguaggio SFC (carta delle funzioni sequenziali) e confermere con ***OK.***



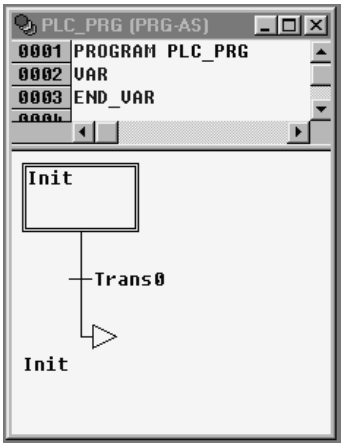
5. Per poter accedere ai componenti MGC*plus* dal programma, è necessario aver caricato la biblioteca MGCPLUS.LIB. A tal scopo aprire la finestra ***Window→Library management*** ed usare ***Insert→Additional library →MGCplus.lib*** per aggiungerla.

La biblioteca contiene tutte le funzioni per poter accedere ai componenti hardware dell'MGC*plus*. Ciascuna funzione disponibile si presenta come un POU con una breve descrizione del significato dei parametri.

Le regole speciali di calcolo, ecc. si trovano in biblioteche separate.



Dopo aver impostato il nuovo progetto, nell'area di lavoro appare la seguente finestra:



Creazione della struttura del programma principale

Come descritto nel paragrafo 4.5, il programma principale è costituito da quattro blocchi.

6. Cliccare sul trattino di separazione della transizione ***Trans0*** per far apparire una cornice tratteggiata attorno a Trans0 e, dal menu contestuale, scegliere (click destro del mouse) il comando ***Step transition (afterwards).*** Ripetere quest'azione due volte fino a raggiungere il Blocco 4.
7. Cliccando direttamente sul nome di una transizione o di un blocco, esso viene marcato in blu e può essere modificato. Ridenominare in questo modo i tre nuovi blocchi **START**, **RUN** e **STOP.**

Assegnazione dei commenti

8. Marcare un blocco cliccando appena fuori della sua cornice. Appare una ulteriore cornice punteggiata attorno al blocco. Cliccare col tasto destro del mouse sul blocco e scegliere ***Step attribute*** dal menu contestuale. Nella finestra che si apre possono essere scritti i propri commenti.
9. Per far apparire i commenti a destra, vicino al blocco, selezionare l'opzione ***Comments*** dal menu ***Tools –> Options...***

Programmazione del blocco “Init”

I segnali necessari nel sistema vengono chiamati dalla gestione della biblioteca usando la funzione *RequestSignal()*. Quali parametri di chiamata devono essere trasmessi il numero del canale, quello del sottocanale e quello del segnale. La codifica dei tipi di segnale è spiegata nella finestra *“Library Manager”*.
Il valore restituito dalla funzione è un identificatore (handle), con cui si può successivamente interrogare il valore di misura.

Nel nostro esempio, la coppia viene misurata come segnale lordo dal Canale 1, Sottocanale 1 e la velocità come segnale netto dal Canale 2, Sottocanale1.

10. Doppio click sul blocco “Init”. Viene richiesto il linguaggio desiderato. Scegliere quello **ST** (structured text) ed appare una nuova finestra per le azioni su questo blocco. Quali prime due linee di programma scrivere:

    HandleTorque := RequestSignal(1,1,0);

    HandleRpm := RequestSignal(2,1,1);

Le variabili “HandleTorque” ed “HandleRpm” devono essere dichiarate quali numeri interi (INT).

Si deve poi definire la scalatura per il risultato del calcolo:

per esempio, il massimo valore della coppia sia 500 Nm, e quello massimo della velocità sia 10000 min^{-1}.

Ne consegue che la massima potenza da misurare sarà:

$$10000 \cdot 100 \cdot 2 \cdot 3{,}14 / 60\ W = 104666\ W$$

11. Nella terza riga di programma assegnare:
    SetScaling(1,120000.0,0.0, 1,’W’);

Significato dei parametri:

1	Scalatura del Sottocanale 1 dell'ML70B
120000.0	Valore massimo che possa essere generato dal calcolo
0.0	Nessuna traslazione dello zero
1	Il risultato del calcolo deve essere mostrato con un decimale
’W’	L'unità di misura del valore è ’W’ (Watt)

Si conclude così la programmazione del blocco “Init” e può venir chiusa la finestra **Action Init.** Nell'angolo in alto a sinistra del blocco INIT appare ora un piccolo triangolo per segnalare che questo blocco è stato programmato.

Programmazione del blocco “START”

I segnali necessari nel sistema vengono chiamati dalla funzione *ActivateSignals()*. Il blocco START viene chiamato fino a che non avviene il rilascio del segnale di misura. Con ciò viene soddisfa la condizione per la transizione del blocco RUN.

Per il blocco START assegnare la seguente riga di programma

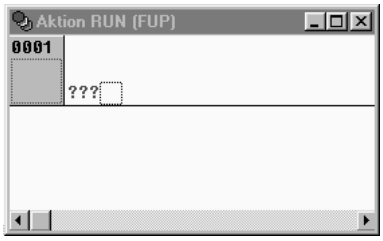
Ready := ActivateSignals(SIGNAL_REQUEST);

e dichiarare la variabile Ready quale numero intero (INT).

Programmazione del blocco “RUN”

Per il blocco RUN si usa il linguaggio di programmazione FBD (function block diagram). I calcoli effettivi vengono effettuati in questo blocco.
Nel nostro esempio di programma non si esce più da questo blocco.

1. Doppio click sul blocco RUN e selezionare il linguaggio FBD.
   Appare la seguente finestra:

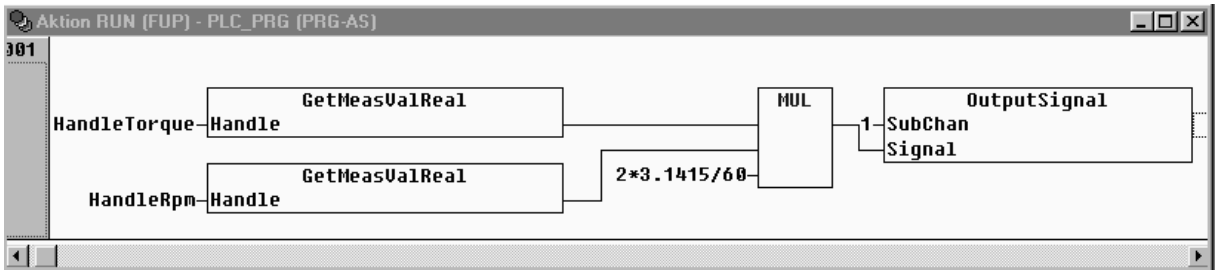


2. Puntare il mouse sul quadratino vicino ai tre punti interrogativi e, usando il menu contestuale (click del tasto destro), eseguire il comando ***POU.*** Viene attivato l'oggetto ***AND*** (presunto) e risulta marcato il nome.
3. Premere il tasto F2 (aiuto assegnazione). Si apre un dialogo per la scelta degli oggetti POU disponibili. Per il nostro esempio, scegliere innanzi tutto la categoria ***Standard functions*** nella finestra di sinistra. Dalla finestra di destra, biblioteca **Targets\Hottinger\...\MGCplus.lib**, cartella **Access to ML70B**, selezionare la funzione **OutputSignal**.
4. Indi cliccare sul gruppo dei tre punti interrogativi vicino alla variabile di ingresso **SubChan** ed assegnare il valore ‚1‘ .
5. Cliccare sul trattino di congiunzione vicino alla variabile d'ingresso **Signal**, facendo così apparire una cornice punteggiata e, nel menu contestuale, eseguire il comando **POU.**
6. Sostituire il testo marcato ‚AND‘ assegnando ‚MUL‘ (moltiplicazione) al suo posto.
7. Cliccare nuovamente sul trattino di congiunzione vicino alla variabile di ingresso superiore di ‚MUL‘ e, nel menu contestuale, eseguire il comando **POU.**
8. Premere il tasto F2 ed aggiungere la funzione **Standard-Functions → MGCplus.lib → Read measuring values → GetMeasValReal**.
9. Ripetere la procedura per la seconda variabile di ingresso di ‚MUL‘. Assegnare le variabili HandleTorque ed HandleRpm quali variabili di

ingresso GetMeasValReal(). Ora resta solo da inserire il fattore di correzione per la moltiplicazione.

10. Per farlo, cliccare sull'oggetto **MUL** e, nel menu contestuale, eseguire il comando **Input.** Appare un'altra assegnazione con ??? quale grandezza d'ingresso. Cliccare sul gruppo dei tre punti interrogativi e sostituirli con l'espressione **2 * 3.1415/60**

Ciò conclude la programmazione del blocco RUN. Lo schema risultante dovrebbe ora apparire come il sottostante:



Definizione delle transizioni

Le transizioni sono le condizioni di passaggio da un blocco al successivo. Sovrascrivere la prima transizione dopo Init con il valore TRUE. Con ciò il blocco Init viene eseguito esattamente una sola volta. La seconda transizione dopo START riceve la condizione Ready = 0 (la variabile assegnata nel blocco START). Quando Ready = TRUE, pertanto è terminata la chiamata del segnale, inizia il blocco successivo.
La terza e quarta transizione ricevono il valore FALSE e, di conseguenza, il blocco RUN non viene più lasciato.

La programmazione è ora terminata. Il programma che ne risulta dovrebbe essere simile al seguente:

5.3 Traduzione del progetto

Tradurre il progetto mediante il comando del menu ***Project → Translate all*** oppure premendo il tasto funzione **<F11>.** Dopo la traduzione, nella finestra dei messaggi in basso a destra, dovrebbe apparire **"0 errors"**. Se ciò non accade, verificare la correttezza di tutte le assegnazioni.
Fare anche attenzione a tutti i messaggi di errore.

Premendo il tasto funzione **<F4>** si giunge direttamente agli errori del programma.

5.4 Lancio del sistema target e caricamento programma

Il sistema target per questo esempio è un MGC*plus* composto come segue:

Indicatore/terminale AB22A
Processore di comunicazione CP42 (opzionale)
Canale 1: ML55B, ML01B o simili (amplificatore monocanale)
Canale 2: ML55B, ML01B o simili (amplificatore monocanale)
Canale 3: ML70B

Il PC e l'ML70B comunicano tramite un collegamento seriale. Connettere la presa DEBUG del pannello frontale dell'ML70B all'interfaccia seriale del PC con il cavo in dotazione. Per configurare l'interfaccia procedere come segue:

- Eseguire il comando di menu **Online → Communications parameter.** Nel dialogo che appare attivare il bottone ***New***, per configurare la connessione al sistema target.
- Nel nuovo dialogo assegnare un nome idoneo alla connessione e selezionare ***Serial RS232***. Cliccando ripetutamente su interfaccia COM1 si commuta su COM1, COM3, ecc. Chiudere ed abbandonare il dialogo con **OK**.
- Eseguire il comando di menu ***Online → Login*** del sistema di programmazione **CoDeSys** per stabilire il collegamento al sistema target e, con ***Online → Start*****,** lanciare il programma sul sistema target.

Selezionando l'ML70B con CHANNEL + / -. Sull'AB22A deve ora apparire il risultato della moltiplicazione. Sullo schermo del PC si possono ora osservare le finestre individuali delle variabili.
Volendo che il programma sia permanentemente alimentato dai dati dello ML70B, si deve chiamare il menu ***Online → Generate boot***. Ciò trasferisce il programma nella memoria non volatiile FLASH. Alla successiva accensione dell'MGC*plus*, questo programma verrà lanciato automaticamente.

6 Comunicazione con altri canali-amplificatori

6.1 Lettura dei valori misurati

Per poter leggere i valori di misura degli altri canali amplificatori dell'MGCplus è prima necessario richiederli ed attivarli (vedere paragrafo 4.5).

I valori di misura vengono letti tramite le funzioni `GetMeasValReal()` e `GetMeasValRaw()`.

Esempio:

Valore di misura: REAL

Handle: INT

ValMis := GetMeasValReal(Handle)

La variabile “Handle” deve essere stata precedentemente assegnata nel blocco START (vedere paragrafi 4.5 e 5.2).

Ai valori di misura si può poi accedere solo tramite questo Handle.
La funzione `GetMeasValReal()` restituisce sempre il valore corrente dello Handle specificato, sotto forma di grandezza fisica scalata.

GetMeasValRaw() restituisce il valore grezzo non scalato quale numero intero da 32 Bit, in formato DINT. Questo formato è quello da preferire quando, per ragioni di tempo, si deve rinunciare ai calcoli in formato REAL.
I valori grezzi si possono convertire in valori di misura fisici con la seguente formula:

$$\textit{Val. mis.} = \frac{\textit{V. grezzo}}{7680000} \cdot \textit{Fondo scala} - \textit{Trasl. zero}$$

Il valore di fondo scala e quello di traslazione dello zero sono definiti nella scalatura dell'amplificatore (vedere anche paragrafo 7.3) e possono essere richiesti all'amplificatore con la funzione `GetChannelInfo()`.

6.2 Invio dei comandi

L'ML70B comunica con gli altri inserti-amplificatori mediante le funzioni `SendMgcCommand()`, `WaitMgcAnswer()` e `GetMgcAnswer()`.

Nel primo blocco, la stringa di comando viene inviata al canale o sottocanale desiderato con la funzione `SendMgcCommand()`.

Dopo viene continuamente interrogata la funzione `WaitMgcAnswer()` fino a che essa restituisce il risultato 0 (OK) oppure <0 (Errore). Infine si ottiene la stringa di risposta con la funzione `GetMgcAnswer()`.

7 Configurazione dei componenti hardware

7.1 Uscite analogiche

Le uscite analogiche dell'ML70B vengono attivate dalle funzioni `SetAnalogOutputReal()` e `SetAnalogOutputInt()`.

Il parametro 1 corrisponde al numero dell'uscita analogica (1 =V_{O1} sulla presa BNC o dell'AP75/AP78; 2 = V_{O2} dell'AP75/AP78).

Col secondo parametro della funzione `SetAnalogOutputReal()` si può assegnare direttamente la tensione di uscita desiderata in Volt (–10....+10.) quale variabile REAL.

Se il calcolo in REAL prende troppo tempo, si può usare la funzione INT `SetAnalogOutputInt()`. Quale secondo parametro assegnare un numero intero (INT) compreso nel campo –30000 ... 30000. –30000 corrisponde a –10 V e 30000 corrisponde a +10 V.

7.2 LED del pannello frontale

I LED del pannello frontale dell'ML70B si possono attivare con la funzione `SetLED()`.
Il parametro 1 specifica il numero del LED (1...7) così come indicato sul pannello frontale. Il parametro 2 specifica lo stato desiderato del LED:
0 = spento; 1 = rosso; 2 = giallo.

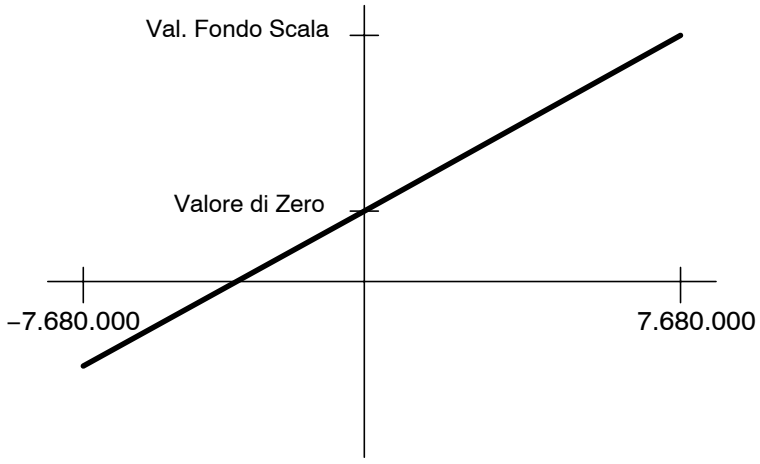
7.3 Uscita dei valori calcolati

La funzione `OutputSignal()` convoglia i valori calcolati sul Bus dati interno. Il parametro 1 specifica il numero del sottocanale desiderato a cui l'ML70B deve inviare il segnale. Il secondo parametro è il valore REAL da trasmettere.

NOTA

Per raffigurare correttamente i dati su catman® o sull'AB22A, prima dell'uscita del primo valore si deve definire la scalatura.

Nell'MGCplus tutti i valori di misura vengono trasferiti quali numeri interi (Integer) a 24 Bit. Il campo è di –7 680 000 ... +7 680 000. Tutti i numeri REAL vengono trasferiti dopo la conversione in Integer a 24 Bit secondo il sottostante schema:



La ricevente riconverte tutti i numeri Integer in numeri Real.

Per la conversione, devono essere assegnati all'ML70B i valori di fondo scala e di traslazione con la funzione `SetScling()`:

$$Val.misura = \frac{Val.Link}{7.680000} \bullet F.S. - Zero$$

8 Controllo dei pannelli di connessione

8.1 AP71/CAN

Il funzionamento dell'interfaccia CAN è spiegato dagli esempi di programmazione MGC_CanDemo, MGC_CanOpenDemo ed MGC_SDOTerm qui inclusi. Per operare il CAN_Hardware si usano le funzioni per l'AP71 della biblioteca MGCplus.lib. Per semplificare la trasmissione e la ricezione dei messaggi CAN, è disponibile l'ulteriore biblioteca MGCcan.lib che contiene tutte le funzioni necessarie.

8.1.1 Funzione base del driver CAN

Ambedue le interfaccia CAN del pannello di connessione AP71 sono controllate da interrupt del firmware. Il programma IEC61131–3 accede alle interfaccia CAN esclusivamente tramite la struttura dati sotto mostrata.
I messaggi CAN vengono trasmessi e ricevuti mediante una memoria buffer circolare. Comparando gli indici di lettura e scrittura, il programma IEC è in grado di stabilire se sono stati ricevuti nuovi messaggi.

```
0001 TYPE CAN_Interface :
0002 STRUCT
0003   pRx_Buffer    : POINTER TO CAN_Message;
0004   pTx_Buffer    : POINTER TO CAN_Message;
0005   pCallBacks    : POINTER TO CAN_CallBack;
0006
0007   CANMAX_RX_BUFFER: INT; (* Größe des Empfangsbuffers  n := (n + 1) MOD CANMAX_RX_BUFFER *)
0008   CANMAX_TX_BUFFER: INT; (* Größe des Sendebuffers  n := (n + 1) MOD CANMAX_TX_BUFFER *)
0009   CANMAX_CALLBACKS: INT; (* Größe des Callbackbuffers  n := (n + 1) MOD CANMAX_CALLBACKS *)
0010
0011   wBaudrate     : WORD; (* Beim Init-Aufruf einzustellende Baudrate *)
0012
0013   nRx_Index_DRV : INT := -1; (* Index des DRV's, für die zuletzt eingetragenen MSG in den Empfangsbuffer *)
0014   nTx_Index_DRV : INT := -1; (* Index des DRV's, für die als nächstes zu versendende  MSG aus dem Sendebuffer *)
0015   nWrite_DRV    : INT;       (* Index des DRV's, für die zuletzt eingetragenen MSG in den Empfangsbuffer
0016                               (für DRV's, die Eintragungen im Empfangsbuffer überschreiben können) *)
0017   nRx_Index_IEC : INT := -1; (* Index des IEC-Programms, für die als nächstes zu lesene MSG aus dem Empfangsbuffer *)
0018   nTx_Index_IEC : INT := -1; (* Index des IEC-Programms, für die zuletzt eingetragenen MSG in den Sendebuffer *)
0019   nWrite_IEC    : INT;  (* Index des IEC-Programms, für die zur Zeit beschriebene MSG in den Sendebuffer *)
0020   bOverWrite    : BOOL; (* hier kann der DRV anzeigen, daß ein Überlauf im Empfangsbuffer aufgetreten ist *)
0021   dwErrorCode   : DWORD; (* hier kann der DRV Errors eintragen, die dann im IEC-Programm ausgewertet werden können *)
0022   pArray        : ARRAY[0..32] OF WORD; (* not used *)
0023 END_STRUCT
0024 END_TYPE
0025
0026
```

La biblioteca MGCcan.lib contiene tutte le funzioni necessarie alla comoda lettura dei messaggi CAN dallla CAN_Interface.

Esempio 1:

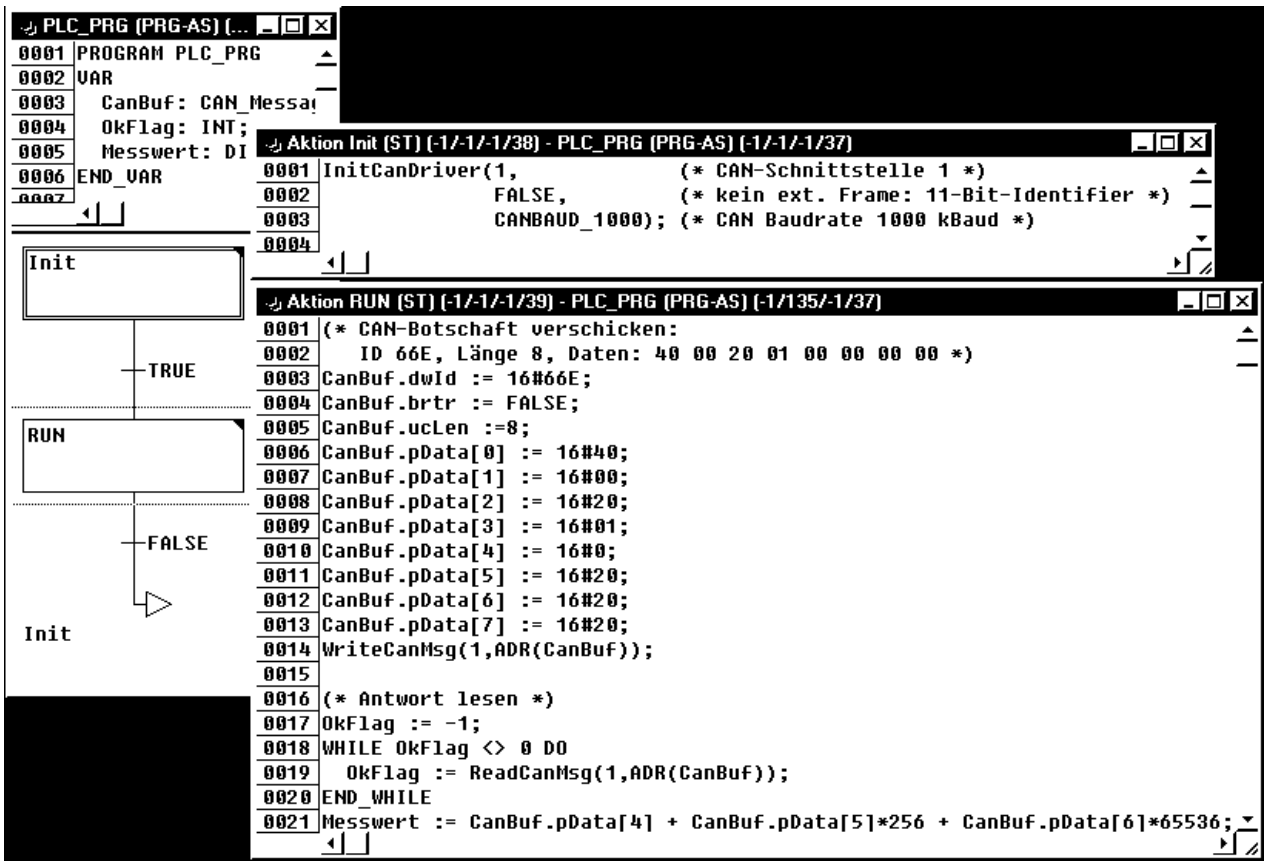
Comunicazione con un Nodo CAN sul piano Low-Level

Si vogliono leggere di contino i valori di misura di un Nodo CAN.

Per prima cosa si deve inizializzare il blocco **Init** dell'interfaccia CAN desiderata, in questo esempio l'interfaccia No. 1 dell'AP71. Si inizializza sia la struttura dati CAN_Interface che l'hardware dell'interfaccia CAN con la funzione `InitCanDriver()`.

Nel blocco **Run** si riempie la struttura dati CanBuf con i valori desiderati ed, infine, si trasmette il messaggio CAN con la funzione `WriteCanMsg()`.

La risposta viene letta con la funzione `ReadCanMsg()`. Nel caso di arrivo di un nuovo messaggio CAN, la funzione risponde con il valore 0.

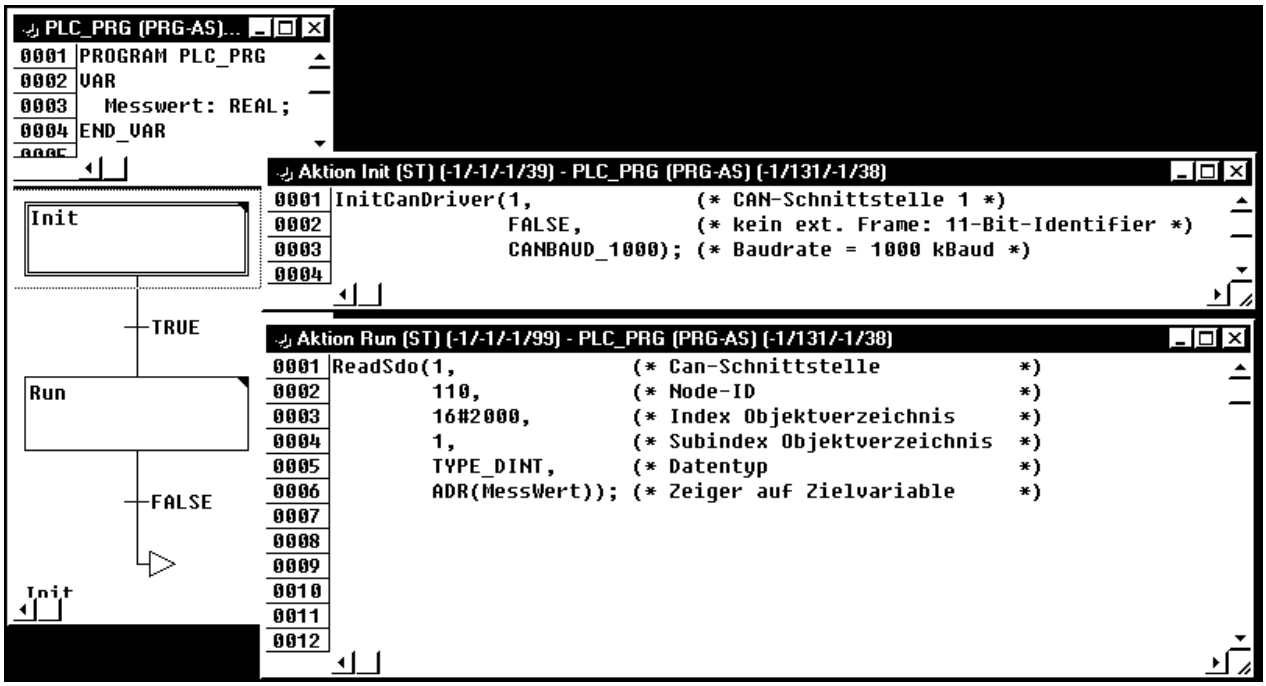


Esempio 2:

Controllo di un Nodo CAN mediante il protocollo CANopen

Si vuole leggere di continuo un valore di misura usando CANopen−SDO (Service Data Object).
Nella biblioteca MGCcan sono disponibili le funzioni ReadSdo() e WriteSdo() per la lettura e scrittura su oggetti del catalogo oggetti del CANopen−Slave.

Nel nostro esempio, il CANopen−Slave possiede il Nodo ID 110 ed il valore di misura risiede nel catalogo oggetti sotto l'indice 16#2000 ed il sottoindice 1, in formato DINT.



Nel blocco **Init** viene inizializzata (una sola volta) l'interfaccia CAN.
Nel blocco **Run** viene letto il valore di misura di CANopen-Slave con la funzione `ReadSdo()`.

8.2 Comunicazione seriale (AP72)

Con l'ML70B si possono controllare fino a quattro interfaccia seriali.
Le interfaccia No. 1 e 2 sono poste nella sede AP-A, quelle No. 3 e 4 sono poste nella sede AP-B. Si può scegliere fra tre tipi:

- RS-232
- RS422 (collegamento a pieno duplex a 4 fili con segnali di retroazione)
- RS485 (collegamento a mezzo duplex a 2 fili con segnali di retroazione)

Per trasmettere e ricevere i dati si devono approntare i corrispondenti buffer nel programma IEC. Le effettive operazioni di trasmissione e ricezione

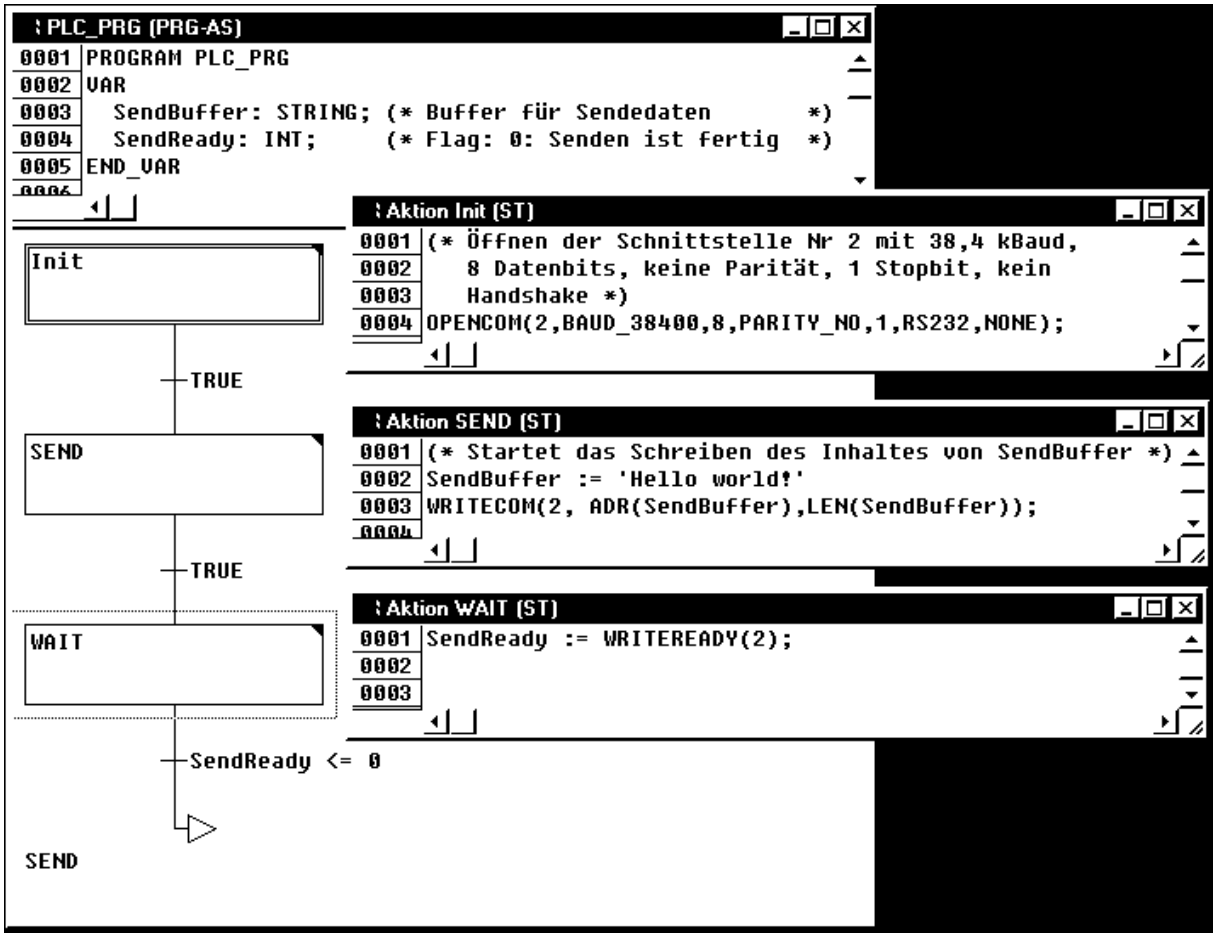
(controllate da interrupt) si svolgono sullo sfondo. I tipi di dati dei buffer dipende dalla specifica applicazione. Usualmente si tratta di STRING o di ARRAY OF BYTE.

8.2.1 Trasmissione dei dati

Per trasmettere i dati sono necessarie le funzioni `OpenCom()`, `WriteCom()` e `WriteReady()`.

I dati vengono trasmessi in tre passi:

1. apertura dell'interfaccia (vedere lo stato “Init” sottostante),
2. preparazione dei dati e inizio processo di trasmissione (vedere lo stato “SEND” sottostante),
3. attesa della fine del processo di trasmissione (vedere lo stato “WAIT” sott.).



L'esempio di programma trasmette senza fine la stringa “Hello world!” alla interfaccia No. 2 . Le costanti necessarie per l'impostazione dell'interfaccia (parametri della funzione OPENCOM()), si trovano nella gestione della biblioteca sotto il registro “Datatypes”.

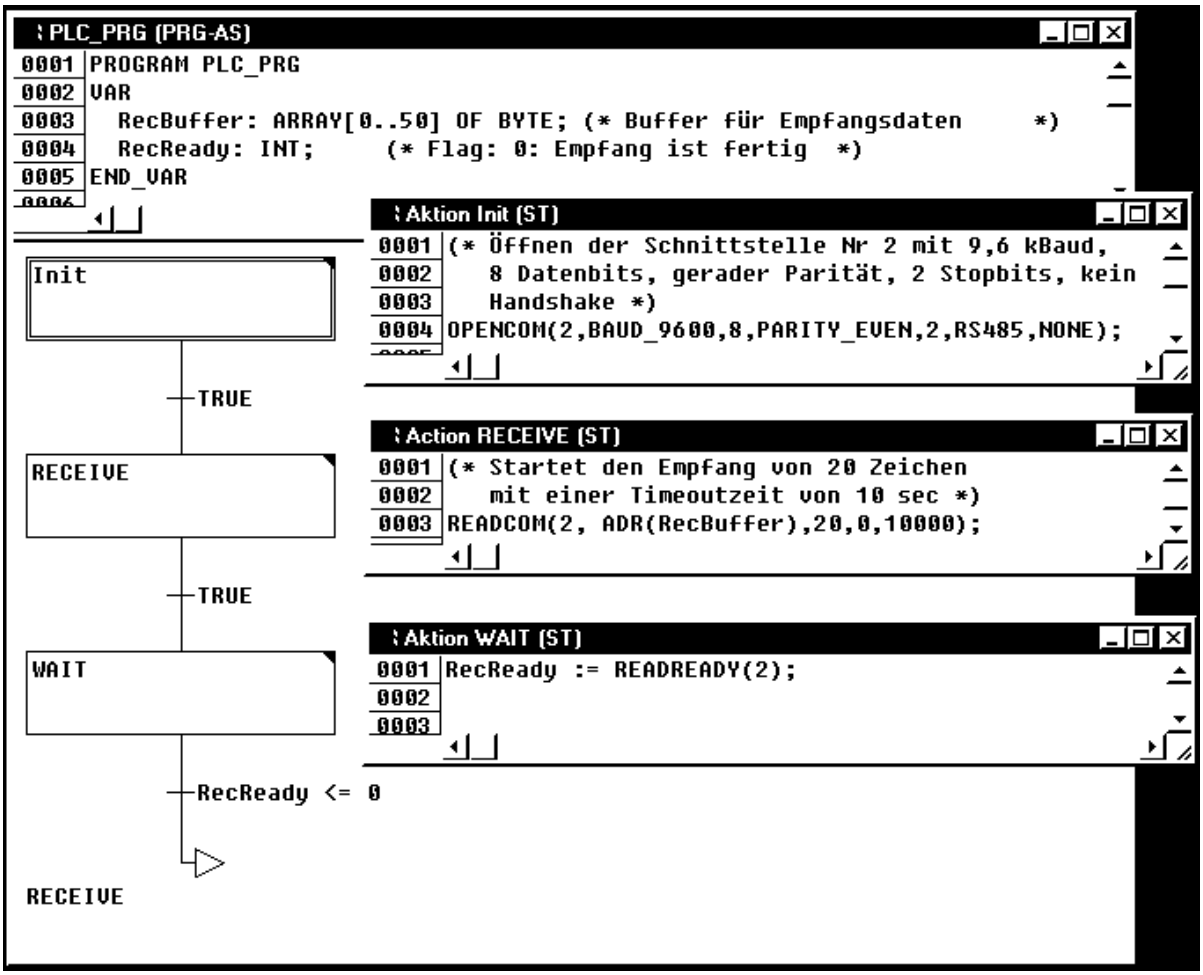
8.2.2 Ricezione dei dati

Per ricevere i dati sono necessarie le funzioni `OpenCom()`, `ReadCom()` e `ReadReady()`.

I dati vengono ricevuti in tre passi:

1. apertura dell'interfaccia (vedere lo stato “Init” sottostante),
2. inizio della ricezione (vedere lo stato “RECEIVE” sottostante),
3. attesa fino alla ricezione completa dei dati attesi (vedere lo stato “WAIT” sottostante).

Esempio 1:

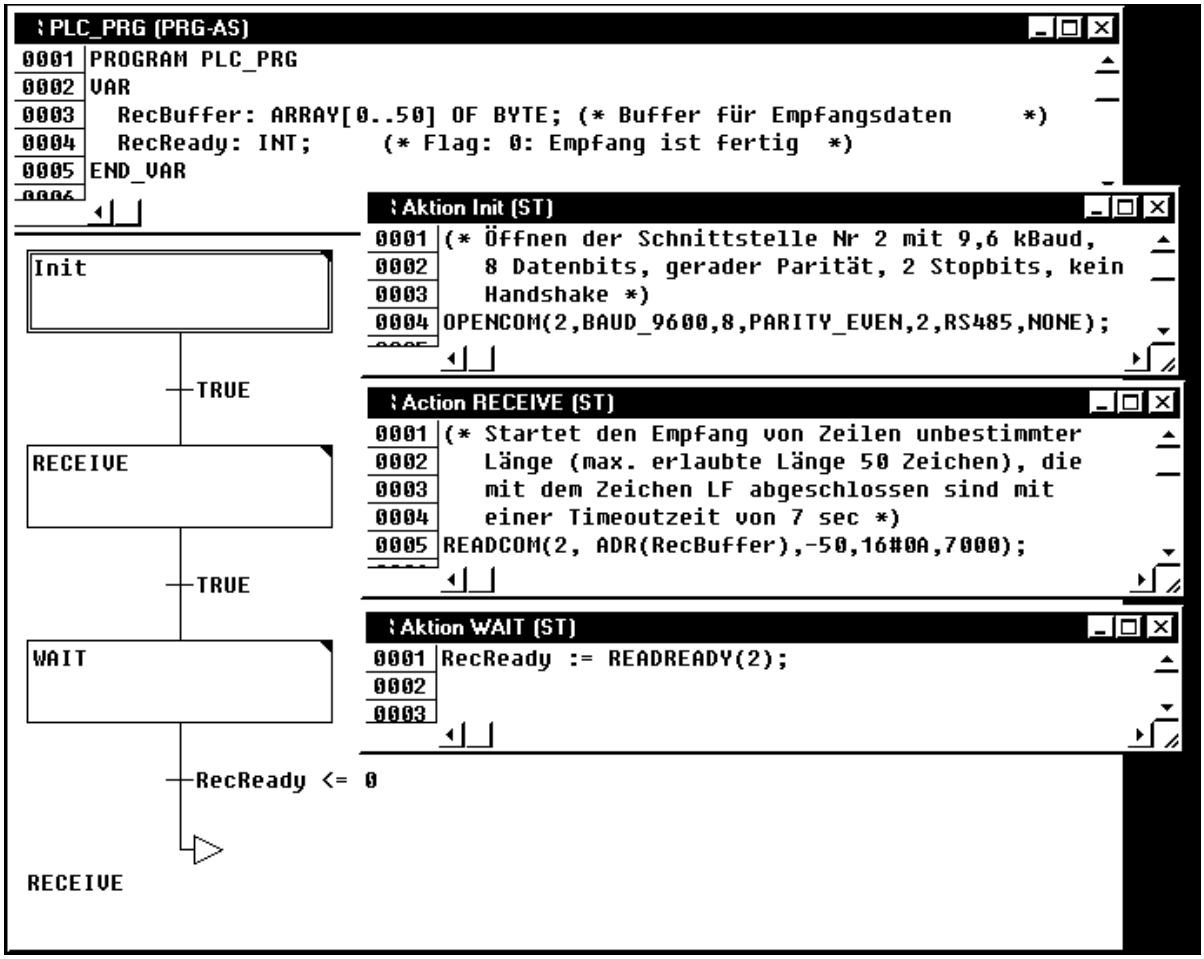


Il programma riceve continuamente blocchi di 20 caratteri ciascuno e li scrive nel buffer “RecBuffer”.

Esempio 2 :

Nel caso in cui la lunghezza dei blocchi di dati sia ignota o variabile, si può scegliere il modo nel quale la ricezione cessa, quando arriva il contrassegno di terminazione.

Per la lunghezza dati viene trasmesso un numero negativo. Il valore del numero specifica la massima lunghezza del buffer disponibile. Infine si deve trasmettere il terminatore desiderato.



8.3 Ingressi ed uscite digitali (AP75)

Le uscite digitali dell'AP75 si attivano con la funzione `SetOutputAP75()` (biblioteca MGCplus.lib) (attenzione: è necessaria l'alimentazione esterna).

Il primo parametro indica il numero dell'uscita. Le uscite 1...8 si trovano nello AP75, sede A (sede direttamente dietro l'ML70B). Le uscite 9...16 si trovano nell'AP75, sede B.

Il secondo parametro specifica il livello: TRUE = 24 V, FALSE = 0 V.

Gli ingressi vengono letti con la funzione `GetInputAp75()`. La numerazione è analoga a quella delle uscite digitali. La risposta TRUE corrisponde

al livello > 10 V in ingresso, la risposta FALSE corrisponde al livello 0 V.

Esempio:

Si vuole attivare il livello HIGH dell'uscita No. 3 dell'AP75 della sede A:
SetOutputAp75(3,TRUE);

8.4 Uscite analogiche AP78

Le funzioni `SetAnalogOutputReal()` e `SetAnalogOutputInt()` vengono usate per controllare le uscite analogiche sull'AP78.

Il parametro 1 specifica il numero dell'uscita analogica (3...10). I numeri 1 e 2 sono riservati per le uscite analogiche dell'ML70B. .

Il secondo parametro della funzione `SetAnalogOutputReal()` specifica la tensione di uscita direttamente in Volt (–10....+10), come variabile REAL.

Se il calcolo in REAL impiega troppo tempo, si può usare la funzione `SetAnalogOutputInt()`. Il secondo parametro è un valore INT nel campo –30000 ... 30000. –30000 corrisponde a –10 V, 30000 corrisponde a +10 V.

9 Generazione del dialogo

9.1 Struttura del dialogo

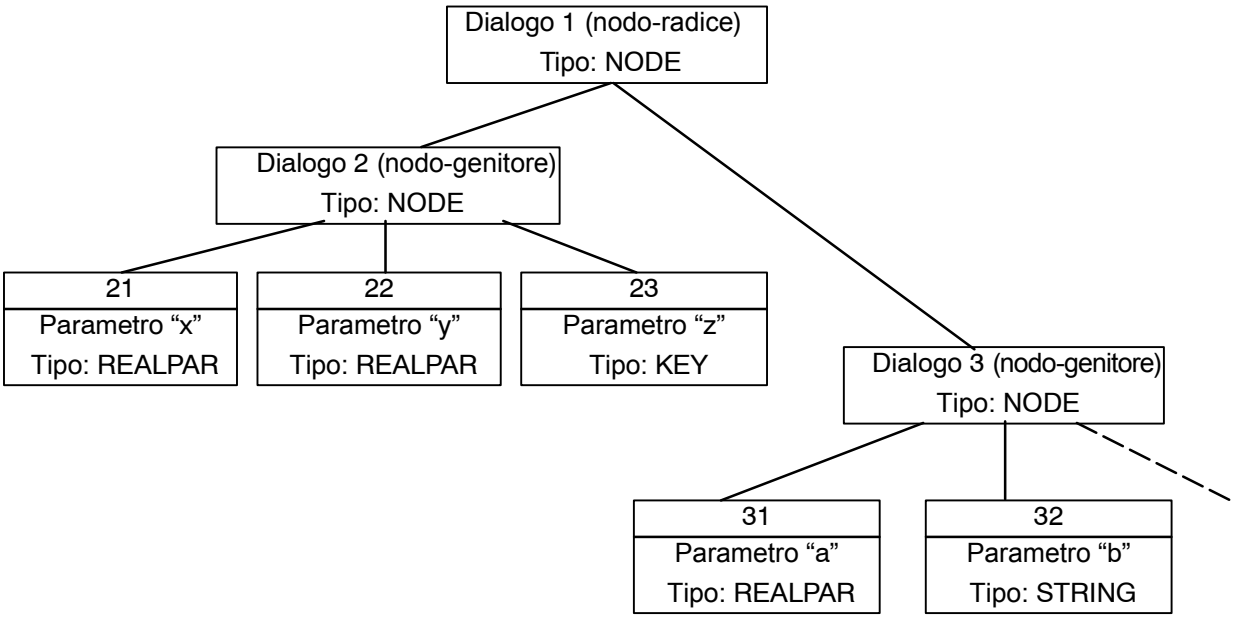
L'inserto di calcolo ML70B offre la possibilità di realizzare delle finestre di Dialogo, le quali possono essere chiamate dall'MGCplus-Assistant o dallo Indicatore/Terminale ABxx. Nel modo impostazione, si possono poi allocare i Dialoghi desiderati ai tasti funzione F3 ed F4 dell'ABxx. Si possono poi definire delle Azioni da richiamare coi tasti funzione durante il modo misura.

I dialoghi sono costituiti da diversi tipi di parametri. I parametri di tipo ”Nodo” formano nuove finestre di dialogo, mentre tutti gli altri tipi di parametro costituiscono il loro contenuto (menu, bottoni, liste di selezione, ecc.).

I parametri del dialogo formano una struttura ad albero. All'apice si trova il nodo della radice che realizza la visualizzazione di uscita. Tramite ulteriori “Nodi genitori” si può realizzare una diramazione “Sottodialogo” di un livello più in basso. I parametri subordinati direttamente ad un nodo formano un dialogo autocontenuto.

Ciascun nodo e ciascun parametro sono identificati da un numero univoco. Il nodo-radice ha sempre il numero 1. Sotto al nodo-radice si possono generare massimo 19 dialoghi assegnando loro i numeri 2 ... 999 (escluso il numero 20 che è già occupato).

I parametri dal numero 20 000 al 20 499 sono riservati per l'assegnazione ai tasti funzione dell'ABxx. Il nodo-radice ha il numero 10 000. Le funzioni desiderate si possono assegnare ai tasti F1 ... F4 durante la configurazione, con la finestra “Display”→ “F-keys”.



Prima di generare un dialogo, per ciascun parametro si devono dichiarare due variabili nel programma IEC:

1. La variabile stessa
2. Un blocco funzione con le proprietà specifiche ed una funzione per le assegnazioni nella lista di parametri

Tipo parametro	Tipo dati per il parametro	Tipo dati per le proprietà parametri	Note
NODE	INT	CreateNode	Nodo
REALPAR	REAL	CreateRealPar	Parametro del tipo REAL
INTPAR	INT	CreateIntPar	Parametro del tipo INT
DINTPAR	DINT	CreateDintPar	Parametro del tipo DINT
KEY	INT	CreateKeyPar	Bottone (come un bottone Windows)
MENUE	INT	CreateMenuePar	Menu di selezione (come un ComboBox Windows)
TEXT	STRING	CreateTextPar	Parametro del tipo STRING

Tab. 9.1: Tipi di parametri

9.2 Proprietà dei parametri

Ogni parametro è descritto da più proprietà e viene assegnato alla lista parametri da una chiamata di funzione.

Ad ogni proprietà è preassegnato un valore iniziale. Si devono pertanto assenare da programma solo le proprietà diverse da quelle preassegnate.

9.2.1 Proprietà del tipo di parametro NODE

I nodi (NODE) riuniscono i parametri in gruppi.

pValue: POINTER TO INT	Puntatore alla variabile (che deve essere definita anche nel programma IEC) contenete lo stato del nodo.
ParId: INT	Numero dei nodi (1...19) Nel modo SET dell, ABxx, i nodi 2..9 sono raffigutati in F3 ed i nodi 11..19 sono raffigurati in F4.
Name: STRING	Nome del nodo: questo testo è raffigurato nel Dialogo dell'ABxx quando, nel modo SET, vengono premuti i tasti corrispondenti (F3 od F4).
Root: INT	Viene specificato a quale nodo-radice appartiene il nodo (qui non effettuare alcuna assegnazione per il nodo 1)

Per generare i nodi usare la funzione `CreateNode()`;

NOTA

Il nodo col numero 1 (nodo radice) risulta già assegnato nell'ML70B.

9.2.2 Proprietà del tipo di parametro DINTPAR, INTPAR, REALPAR

Questi tipi di parametri sono usati per raffigurare i parametri numerici di DINT, INT e REAL.

Di seguito, <tipo> raffigura rispettivamente i tipi DINT, INT e REAL.

pValue: POINTER TO <tipo>	Puntatore alla variabile (anch'essa da definire nel programma IEC), da modificare con il dialogo parametri.
ParId: INT	Numero del parametro (2...999).
Name: STRING	Nome del parametro: questo valore viene raffigurato nel dialogo dell'ABxx, insieme al valore del parametro.
Root: INT	Qui viene definito a quale nodo-radice appartiene il nodo.
MinVal: <tipo>	Valore minimo sotto cui non si può impostare il parametro.
MaxVal: <tipo>	Valore massimo sopra cui non si può impostare il parametro.
EditWidth: INT	Larghezza di modifica
Decimals: INT	Numero di posizioni decimali per la raffigurazione dei parametri.
ScalFact: REAL	Fattore di scalatura per raffigurazione sull'ABxx.
Offset: REAL	Offset per raffigurazione sull'ABxx.
Unit: STRING	Unità fisica quale stinga da 4 caratteri che, nel dialogo parametri, appare insieme al parametro.
Flags: PARFLAGS	Proprietà speciale (finora non supportata).

Si possono usare i parametri ScalFact e Offset per raffigurare i valori dei parametri con un'altra scalatura. Desiderando raffigurare i parametri in modo non scalato (caso normale), impostare Scalfact = 1.0 ed Offset = 0.0

Valore indicato = Parametro * ScalFact – Offset

I parametri si generano con i blocchi funzione `CreateRealPar()`, `CreateIntPar()` e `CreateDintPar()`.

9.2.3 Proprietà del tipo di parametro KEY

Con il parametro KEY si generano i bottoni di commutazione nel Dialogo.
I bottoni permettono di lanciare azioni o di aprire Sottodialoghi.

Un bottone si genera col blocco funzione `CreateKey()`.

pValue: POINTER TO INT	Puntatore alla variabile (anch'essa da definire nel programma IEC), per raffigurare lo stato del bottone nel Dialogo.
ParId: INT	Numero del parametro (2...999)
Name: STRING	Nome del bottone: testo dell'etichetta del bottone nel Dialogo dell'ABxx e dell'Assistant.
Root: INT	Numero del nodo-radice a cui appartiene il parametro.
Flags: PARFLAGS	Proprietà speciale (finora non supportata).

9.2.4 Proprietà del tipo di parametro TEXT

Questo tipo di parametro è usato per creare testi nel Dialogo.
Un testo si genera col blocco funzione `CreateTextPar()`.

pValue: POINTER TO STRING	Puntatore alla variabile (anch'essa da definire nel programma IEC), da modificare col Dialogo.
ParId: INT	Numero del parametro (2...999)
Name: STRING	Nome del parametro: testo dell'etichetta del valore del parametro che appare sull'ABxx.
Root: INT	Numero del nodo-radice a cui appartiene il parametro.
Flags: PARFLAGS	Proprietà speciale (finora non supportata).

9.2.5 Proprietà del tipo di parametro MENUE

Usare questo tipo di parametro per creare campi di selezione nella finestra di dialogo. Il campo di selezione è raffigurato da una variabile INT.
Si può effettuare la scelta fra fino a 20 valori diversi, ciascuno dei quali ha un testo per etichetta.

Un campo di selezione si genera col blocco funzione `CreateMenuePar()`.

pValue: POINTER TO INT	Puntatore alla variabile (anch'essa da definire nel programma IEC), da modificare col menu.
ParId: INT	Numero del parametro (2...999)
Name: STRING	Nome del parametro: testo dell'etichetta del valore del parametro nel dialogo dell'ABxx.
Root: INT	Numero del nodo-radice a cui appartiene il parametro.
Flags: PARFLAGS	Proprietà speciale (finora non supportata).
Items: ARRAY[1..20] OF INT	Valori numerici da cui effettuare la selezione.
ItemTexts: ARRAY[1..20] OF STRING	Testi per i valori numerici.

Esempio 1:

Si vogliono moltiplicare i valori di misura di due canali. Il risultato deve essere trasferito all'uscita analogica e deve essere verificato che esso non sia sotto o sopra i valori Min e Max.

Impostazioni da effettuare:

P_{min}: soglia inferiore

P_{max}: soglia superiore

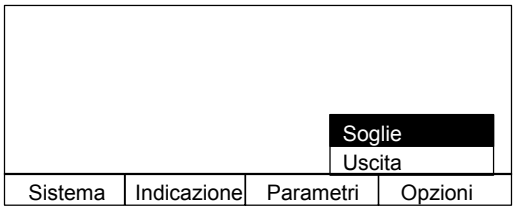
Delete: tasto di annullamento

Rate: cadenza di trasferimento dell'uscita analogica

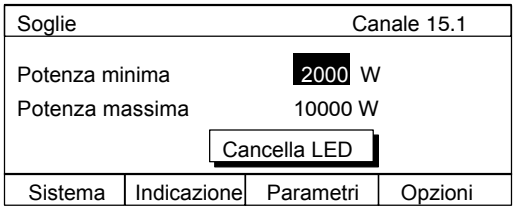
AnalogP1: punto 1 della caratteristica dell'uscita analogica

AnalogP2: punto 2 della caratteristica dell'uscita analogica

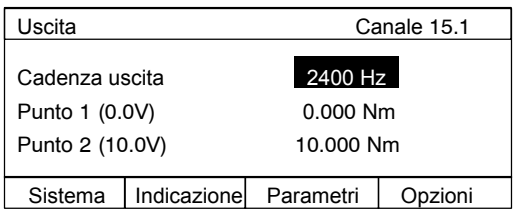
Nell'indicazione i parametri devono essere suddivisi nei seguenti gruppi:



Dialogo di uscita:
Campo di selezione per “Soglie” od “Uscite”

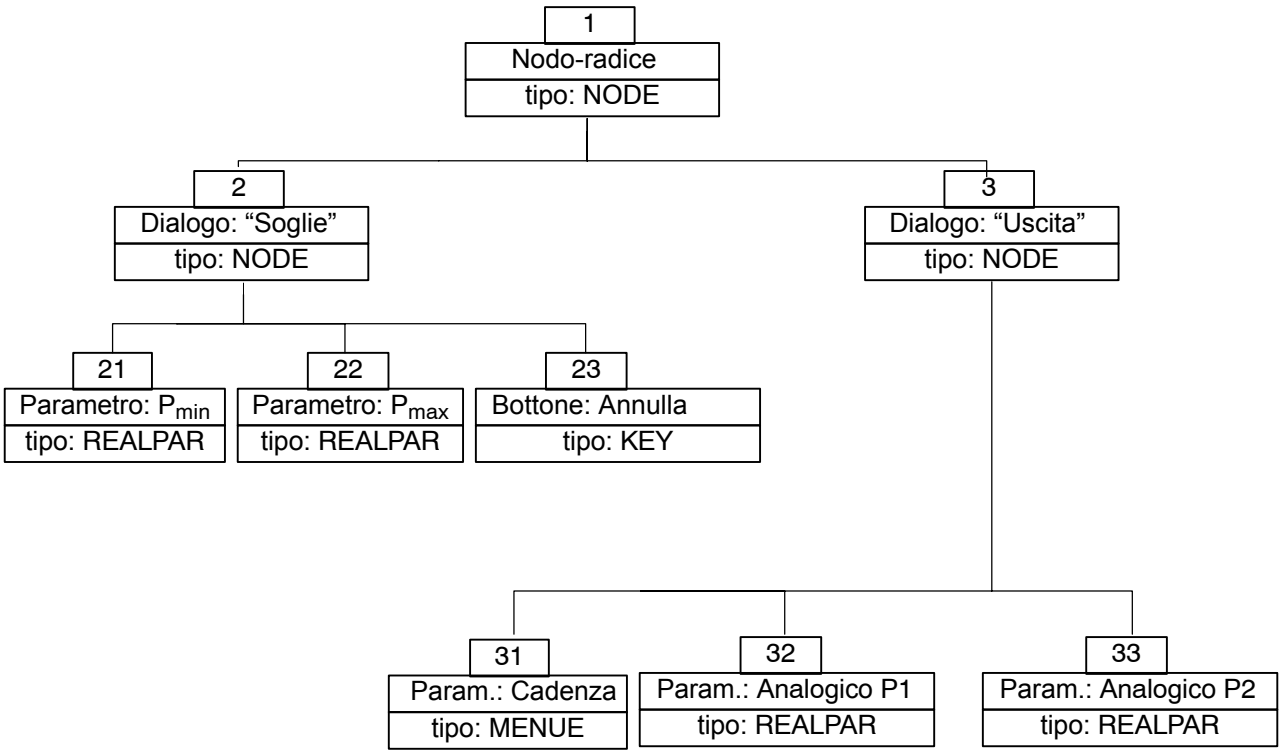


Nella finestra “Soglie” vengono indicati i parametri P_{min} , P_{max} ed il tasto Annulla



Nella finestra “Uscita” vengono indicati i parametri la cadenza di uscita ed i punti 1 e 2

I dialoghi formano la seguente struttura ad albero:



Per definire questo tipo di Dialogo, si devono codificare le proprietà dei parametri nel programma IEC−61131−3. Nella biblioteca MGCplus.lib si trovano i Sottogruppi prototipi “Parametering ML70B” degli oggetti dei processi funzionali, per descrivere i differenti tipi di parametri.

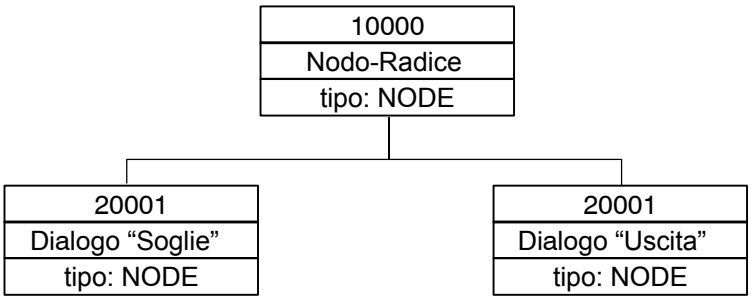
Per codificare l'albero di parametri precedente nel programma IEC, è necessario il seguente codice sorgente. Questo codice deve girare una sola volta. Consigliamo di integrare questo codice nel blocco INT del programma di misura (vedere il paragrafo 4.5).

Spiegazioni	IEC61131–3 – Testo sorgente
I singoli Dialoghi (gruppi di parametri) sono costituiti dai Nodi (qui i numeri 2 e 3) e dai singoli parametri. Ciascun parametro è collegato con il nodo tramite la proprietà “Root”.	(* ------ Menü ”Grenzen” ------------ *) AttrNode2.pValue := ADR(Node2); (* Nodo menu ”Soglie” *) AttrNode2.Name := ’Limits’; (* Testo mostrato *) AttrNode2.ParId := 2; (* Numero parametro *) AttrNode2.Root := 1; (* Nodo-radice *) AttrNode2(); (* Genera l'assegnazione nella lista param. *) AttrPmin.pValue := ADR(Pmin); (* Coppia minima *) AttrPmin.ParId := 21; (* Numero parametro *) AttrPmin.Name := ’Minimum power’; (* Testo mostrato *) AttrPmin.Root := 2; (* Nodo-radice *) AttrPmin.Decimals := 3; (* Numero di decimali per l'indicatore *) AttrPmin.Unit := ’Nm’; (* Unità fisica *) AttrPmin(); (* Genera l'assegnazione nella lista param. *) AttrPmax.pValue := ADR(Pmax); (* Coppia massima *) AttrPmax.ParId := 22; (* Numero parametro *) AttrPmax.Name := ’Maximum power’; (* Testo mostrato *) AttrPmax.Root := 2; (* Nodo-radice *) AttrPmax.Decimals := 3; (* Numero di decimali per l'indicatore *) AttrPmax.Unit := ’Nm’; (* Unità fisica *) AttrPmax(); (* Genera l'assegnazione nella lista param. *) AttrKey1.pValue := ADR(ClearLeds); (* Tasto ”Cancella LED" *) AttrKey1.ParId := 23; (* Numero parametro *) AttrKey1.Name := ’LEDs löschen’; (* Testo mostrato *) AttrKey1.Root := 2; (* Nodo-radice *) AttrKey1(); (* Genera l'assegnazione nella lista param. *) (* ------ Menu ”Uscita” ------------ *) AttrNode3.pValue := ADR(Node3); (* Menü ”Uscita” *) AttrNode3.Name := ’Output’; (* Testo mostrato *) AttrNode3.ParId := 3; (* Numero parametro *) AttrNode3.Root := 1; (* Nodo-radice *) AttrNode3(); (* Genera l'assegnazione nella lista param. *) AttrMenueRate.pValue := ADR(Rate); (* Menu di selezione ”Cadenza di uscita” *) AttrMenueRate.ParId := 31; (* Numero parametro *) AttrMenueRate.Name := ’Output rate’;(* Testo mostrato *) AttrMenueRate.Root := 3; (* Nodo-radice *) AttrMenueRate.Items[1] := 1; (* Valore numerico 1. Assegnazione di menu *) AttrMenueRate.ItemTexts[1] := ’2400 Hz’;(* Testo 1. Assegnazione di menu *) AttrMenueRate.Items[2] := 2; (* Valore numerico 2. Assegnazione di menu *) AttrMenueRate.ItemTexts[2] := ’1200 Hz’;(* Testo 2. Assegnazione di menu *) AttrMenueRate.Items[3] := 4; (* Valore numerico 3. Assegnazione di menu *) AttrMenueRate.ItemTexts[3] := ’600 Hz’;(* Testo 3. Assegnazione di menu *) AttrMenueRate.Items[4] := 8; (* Valore numerico 4. Assegnazione di menu *) AttrMenueRate.ItemTexts[4] := ’300 Hz’;(* Testo 4. Assegnazione di menu *) AttrMenueRate(); (* Genera l'assegnazione nella lista param. *) AttrP1.pValue := ADR(AnalogP1); (* Punto caratteristica 1 Uscita analogica *) AttrP1.ParId := 32; (* Numero parametro *) AttrP1.Name := ’Point 1 (0.0 V)’; (* Testo mostrato *) AttrP1.Root := 3; (* Nodo-radice *) AttrP1.Decimals := 3; (* Numero di decimali per l'indicatore *) AttrP1.Unit := ’Nm’; (* Unità fisica *) AttrP1(); (* Genera l'assegnazione nella lista param. *) AttrP2.pValue := ADR(AnalogP2); (* Punto caratteristica 2 Uscita analogica *) AttrP2.ParId := 33; (* Numero parametro *) AttrP2.Name := ’Point 2 (10.0 V)’; (* Testo mostrato *) AttrP2.Root := 3; (* Nodo-radice *) AttrP2.Decimals := 3; (* Numero di decimali per l'indicatore *) AttrP2.Unit := ’Nm’; (* Unità fisica *) AttrP2(); (* Genera l'assegnazione nella lista param. *)

Esempio 2:

In un programma utente si vogliono definire i tasti funzione “Start” e “Stop”.

Questo esempio ha la seguente struttura ad albero:



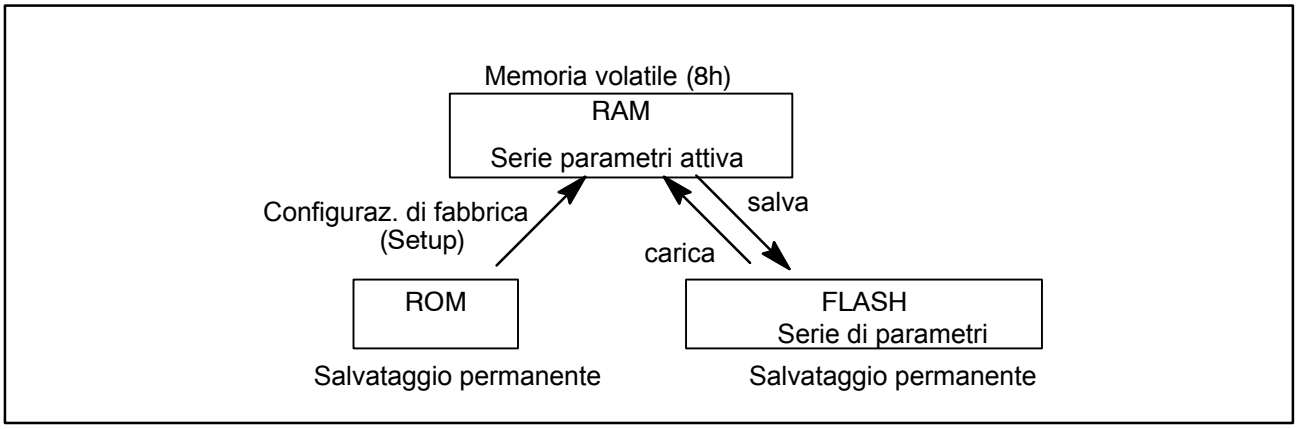
IEC–Testo sorgente

```
(* ––– Tasti funzione –––––– *)
AttrNodeF.pValue := ADR(NodeF);      (* Nodo tasti funzione *)
AttrNodeF.Name := 'F-keys';          (* Testo mostrato *)
AttrNodeF.ParId := 10000;            (* Numero parametro *)
AttrNodeF();                         (* Genera l'assegnazione nella lista param. *)
AttrKeyF1.pValue := ADR(FStart);     (* Tasto "Start" *)
AttrKeyF1.ParId := 20000;            (* Numero parametro *)
AttrKeyF1.Name := 'Start';           (* Testo mostrato *)
AttrKeyF1.Root := 10000;             (* Nodo-Radice *)
AttrKeyF1();                         (* Genera l'assegnazione nella lista param. *)
AttrKeyF2.pValue := ADR(FStop);      (* Tasto "Stop" *)
AttrKeyF2.ParId := 20001;            (* Numero parametro *)
AttrKeyF2.Name := 'Stop';            (* Testo mostrato *)
AttrKeyF2.Root := 10000;             (* Nodo-Radice *)
AttrKeyF2();                         (* Genera l'assegnazione nella lista param. *)
```

Le funzioni vengono attivate da ABxx, mediante incremento del valore delle variabili `Fstart` od `Fstop` ad ogni pressione del tasto. Perciò il programma deve interrogare ciclicamente il valore delle variabili, per poter eseguire la funzione desiderata.

10 Salvataggio dei parametri di impostazione

I parametri di Dialogo descritti nel capitolo 9 possono essere salvati.
Tutti gli inserti MGCplus effettuano il salvataggio col seguente principio:

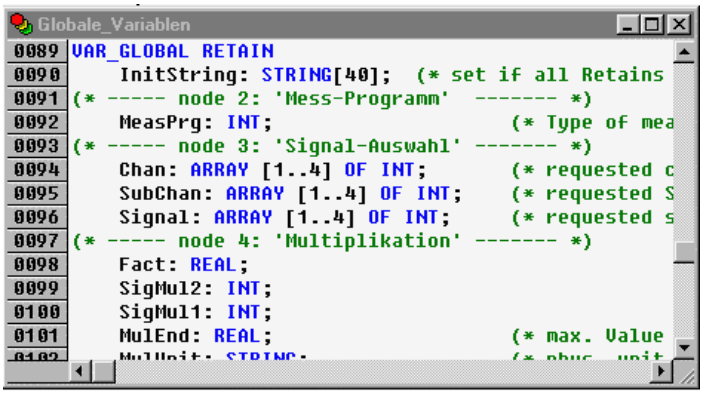


Le impostazioni di configurazione dell'inserto vengono sempre lette dai parametri residenti nella memoria di lavoro (RAM) la quale, in caso di mancanza di tensione, li conserva per almeno 8 ore.

Al momento del salvataggio della serie corrente di parametri, ne viene effettuata una copia nella memoria permanente (Flash). All'atto del caricamento, la memoria Flash viene copiata in quella di lavoro. Ristabilendo la configurazione di fabbrica, la memoria ROM viene copiata in quella di lavoro.

L'esempio che segue dovrebbe chiarire l'effetto di questo principio sul proprio programma utente. L'esempio utilizzato è l'applicazione standard disponibile nella directory `..\CodeSys..\Targets\HBM\ML70B_DemoPrj`.

Tutte le variabili che devono essere salvate permanentemente, devono essere poste nella “area RETAIN” quali variabili globali. Allo spegnimento/accensione dell'MGCplus, quest'area **non** viene inizializzata. In caso di mancanza di tensione, i dati qui posti vengo conservati per circa 8 ore.

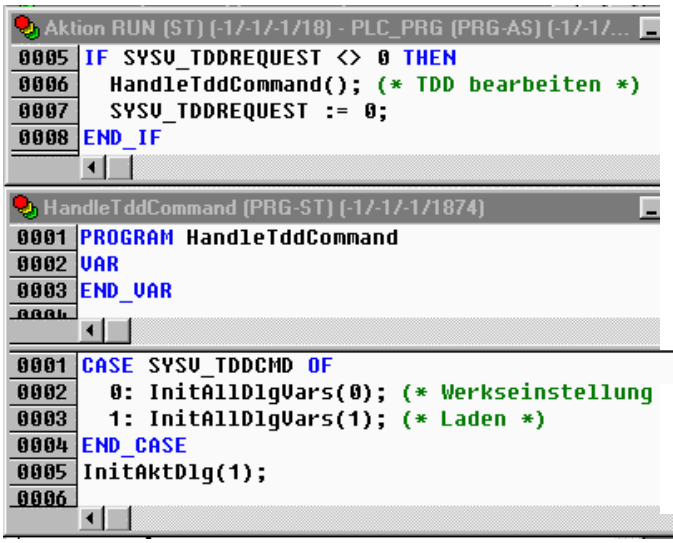


Quando l'indicatore/terminale oppure una interfaccia esterna (p.es. Assistant) tramette il comando **Save** all'ML70B, il sistema operativo dell'ML70B copia

l'area RETAIN nella memoria non volatile. Se invece all'ML70B viene trasmesso il comando **Load**, il sistema operativo copia la serie di parametri dalla memoria FLASH all'area RETAIN.

Dato che il sistema operativo non può conoscere la configurazione di fabbrica del programma utente, è l'utente stesso che deve fare eseguire la funzione **Load factory settings** nel proprio programma.

Tramite le variabili di sistema `SYSV_TDDREQUEST` e `SYSV_TDDCMD` si può chiedere al programma utente IEC se è stato ricevuto un comando TDD (Save/Load).



Quando viene ricevuto un comando TDD, si incrementa la variabile del sistema operativo SYSV_TDDREQUEST.
Qui viene interrogata ciclicamente la variabile di sistema. Alla ricezione del comando TDD viene chiamato il programma `HandleTddCommand`.
La variabile `SYSV_TDDCMD` mostra il tipo di comando ricevuto:
0: Carica impostazione di fabbrica
1: Carica
2: Salva

Qui il programma gestisce anche il comando “Load” dato che, al cambiamento di alcuni parametri, devono essere effettuate delle impostazioni addizionali interne di programma.

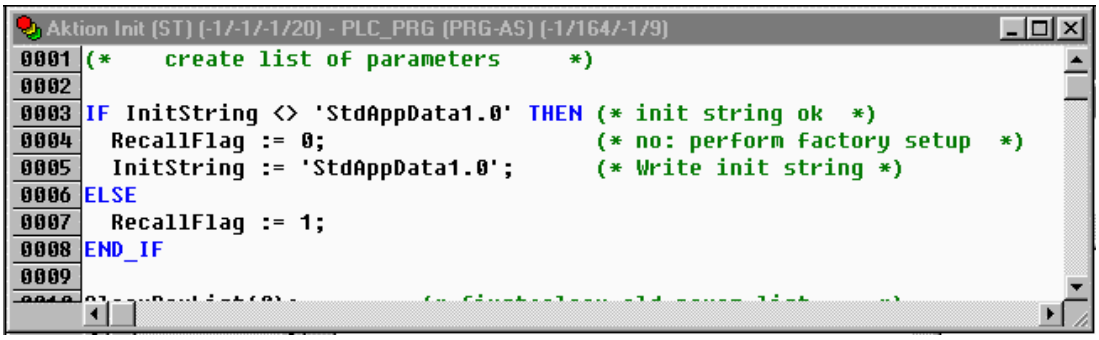
Quando viene salvata l'area RETAIN, il sistema operativo calcola una somma di verifica.

All'accensione dell'MGCplus, per prima cosa il sistema operativo controlla se è corretta la somma di verifica dei dati RETAIN. Se non lo è, viene copiata automaticamente la serie di parametri dalla menoria FLASH nella RAM.

NOTA

Tuttavia, questo procedimento non garantisce ancora che la serie di parametri sia idonea al programma in esecuzione. Infatti, per esempio, la serie di parametri potrebbe essere tuttora quella del programma caricato precedentemente.

Ne consegue che il programma dovrebbe iniziare con le seguenti verifiche:



Quando nella stringa `InitString` non vengono trovati valori idonei alla applicazione, viene automaticamente eseguita la impostazione di fabbrica e, conseguentemente, tutte le variabili della serie di parametri assumono valori significativi.

11 Modo multiprogramma

Si possono eseguire più programmi fra loro indipendenti e visibilmente “contemporanei” (multitasking). Nella pagina di registro “Ressourcen” è disponibile l'assegnazione “Task configuration”. Qui si possono collegare differenti programmi a diversi Task. Sono possibili fino a 32 diversi Task. Per ogni Task individuale si può assegnare la frequenza di chiamata in incrementi di ms e la priorità.

La descrizione dettagliata della configurazione del Task si trova nell'aiuto Online del CoDeSys sotto **Contents –> Ressourcen –> Task configuration**.

NOTA

La frequenza di chiamata del programma IEC dell'ML70B è di 2400 Hz (vedere il paragrafo 4.1).

La frequenza di chiamata del Task deve essere espressa in millisecondi. Si ottiene il trasferimento equidistante dei valori di misura solo quando il risultato della seguente formula *Divisore* è un numero intero:

$$Divisore = \frac{2400\ Hz}{1000} \bullet Intervallo\_\ chiamata\ [ms]$$

Ne consegue che, per modo operativo multitasking, la massima frequenza di uscita con valori di misura esattamente equidistanti è di 200 Hz.

12 Casi speciali

12.1 Uscita equidistante dei valori di misura

In alcune applicazioni può essere necessario che i valori di misura escano ad intervalli di tempo eguali (equidistanti). Con i seguenti passi si determina il massimo tempo di calcolo del programma, e poi si definisce questi tempo come cadenza di uscita:

1. Effettuare il Log in.
2. Nella finestra ”Globalvariables” cancellare la variabile di sistema SYSV_MAXEXECTIME.
3. Lanciare il programma.
4. Leggere la variabile globale SYSV_MAXEXECTIME.
5. All'inizio del programma IEC aggiungere la riga
   SYSV_REQEXECTIME := <valore>.
   Al posto di <valore> mettere il valore trovato.

Con ciò si rileva il massimo tempo di calcolo e si definisce la cadenza di uscita. Il tempo deve essere specificato in incrementi di 1/2400 Hz = 416,6 µs.

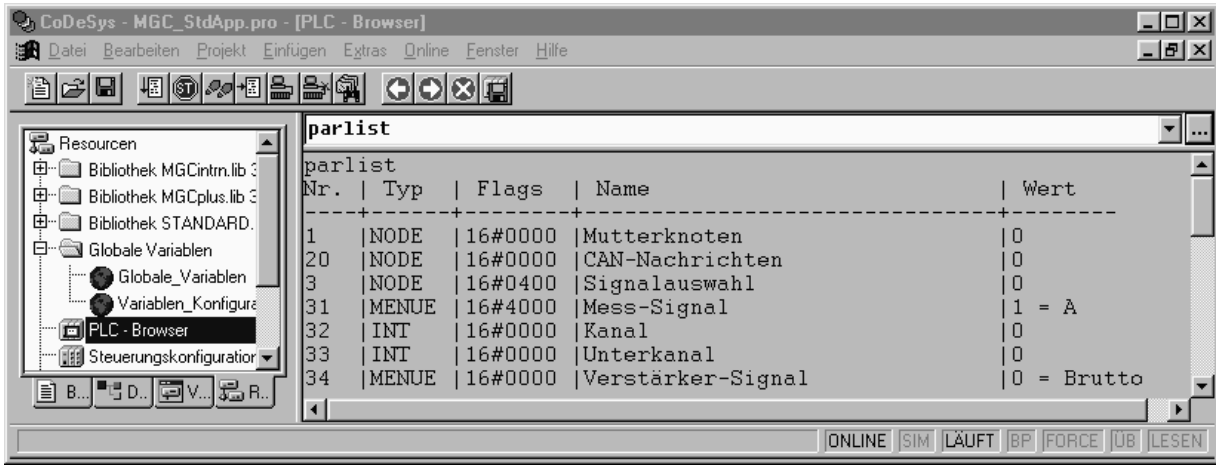
12.2 Variazione del numero di sottocanali

Quale numero di Sottocanali dell'ML70B se ne può scegliere da 1 a 128.

Mediante la interfaccia esterna trasmettere il comando **PAR9990, <numero di sottocanali>.** Dopo di ché, spegnere e riaccendere l'MGCplus.
Ora risulta impostato permanentemente il numero di Sottocanali desiderato.

13 Funzione di Debug (PLC–Browser)

Per facilitare la ricerca degli errori nel codice, il sistema di programmazione CoDeSys possiede alcune funzioni di Debug. Nel registro “Ressourcen” appare la voce PLC Browser. Il Browser è costituito da una riga di comando e da una finestra di uscita. Il comando inserito nella riga di comando trasmette il suo risultato nella finestra di uscita.



Il manuale di istruzione dettagliato del Browsers si trova nell'aiuto Online del CoDeSys sotto **Contents –> Ressourcen –> PLC browser.**
Ulteriori istruzioni per inserire i comandi si trovano nei bottoni etichettati “...”, alla destra della riga di assegnazione del PLC Browser.

Nel PLC Browser sono disponibili i seguenti comandi:

Comando	**Descrizione**
comp	Confronto memoria
compd	Confronto memoria con Memory Dump
mem	Hexdump di un area di memoria
memc	Hexdump relativo all'indirizzo di start del codice sul controllo
memd	Hexdump relativo all'indirizzo del data base sul controllo
memset	Assegna area memoria: <Indirizzo-Start>, <Fill–Byte>, <Lunghezza>
metrics	Visualizza PLC metrico
reflect	Riflessione della riga di comando corrente, per ragioni di prova
dpt	Lettura della tabella dei Puntatori–Dati
ppt	Lettura della Tabella-POU
pid	Lettura dell'ID-Progetto
pinf	Lettura delle Info-Progetto
?	Uscita elenco dei comandi disponibili
cycletimes	Visualizza i tempi di ciclo del programma sull'ML70B
parlist	Visualizza l'elenco dei parametri dell'ML70B
signals	Visualizza l'elenco di tutti i segnali richiesti dall'ML70B

14 Variabili di sistema

Nome variabile	Tipo	Significato
SYSV_ActualExecTime	WORD	Tempo di esecuzione dell'ultimo blocco, misurato in incrementi temporali di 1/2400 Hz = 416 µs
SYSV_MaxExecTime	WORD	Max. tempo di esecuzione di tutti gli ultimi blocchi, misurato in incrementi temporali di 1/2400 Hz = 416 µs
SYSV_ReqExecTime	WORD	Tempo di esecuzione dell'ultimo blocco, misurato in incrementi temporali di 1/2400 Hz = 416 µs
SYSV_ParChangeF-romML70B	WORD	Incrementato dall'ML70B ogni volta che un parametro viene modificato dall'esterno (interfaccia esterna CP od AB)
SYSV_ParChangeFromIEC	WORD	
SYSV_TddRequest	WORD	Incrementato ogni volta che viene ricevuto un comando TDD dall'esterno (interfaccia esterna CP od AB)
SYSV_TddCmd	WORD	Ultimo comando TDD ricevuto
SYSV_TddPar	WORD	Ultima serie di parametri TDD ricevuta

15 Messaggi di errore

Molte funzioni della biblioteca MGCPLUS.LIB restituiscono dei codici di errore, che permettono di cercarne la causa con maggior precisione.
I valori negativi sono veri errori, quelli positivi sono informazioni di stato.

CodiceErrore	Significato
0	Nessun errore
−100	Errore alla chiamata di RemovePar(): questo parametro non esiste
−101	Il Nodo-Radice dato non esiste
−102	Saturazione memoria alla definizione di un parametro
−103	Troppe voci nel nodo
−104	Elenco voci già vuoto
−105	La voce non può essere cancellata (1 e 20)
−106	Nodo-Radice 20 non permesso !
−107	I Nodi numero 1 e 20 sono riservati
−108	Numero parametro errato
200	ActivateSignals() ancora attivo
−210	OpenComPort(): PrtNr errata; questa interfaccia seriale non esiste
−211	OpenComPort(): Parity−Info errata
−212	OpenComPort(): Numero Stopbit errato
−213	OpenComPort(): Baudrate non esistente
−214	OpenComPort(): Numero dei bit di dato errato
−215	Com−Port non aperta
−216	ReadCom() Parametro BUFF: puntatore non valido
−217	Superamento Timeout durante la ricezione seriale
−218	Superata la lunghezza riga durante la ricezione seriale
−219	Errore di ricezione della interfaccia seriale (Parity, Framing o Overrun-Error)
−220	Hardware−Handshake possibile solo con RS232
−221	Software−Handshake non possibile con RS485
−230	Numero canale errato
−231	Numero sottocanale errato
−233	SetScaling(): fine scalatura 0 non permesso
−234	SetScaling(): sono possibili 0...5 cifre decimali
240	In attesa di risposta
−241	Protocollo comando già in corso
−242	Timeout della interfaccia interna MGC
−250	Richiesti troppi segnali con RequestSignal()
−251	ActivateSignals(): segnale non disponibile
−260	SetAnalogOutput(): l'uscita analogica non esiste

16 Dati tecnici dell'ML70B

Uscite analogiche		
Max. numero di uscite analogiche		2 (10 con AP78)
Cadenza aggiornamento uscite analogiche	Hz	2400
Tensione nominale	V	± 10 V asimmetrica
Resistenza di carico ammessa	kΩ	> 5
Resistenza interna	Ω	< 5
Residuo alternato (76,6 kHz)	mVpp	< 12
Deriva a lungo termine (oltre 48 h)	mV	< 3
Influenza della temperatura ambiente, ogni 10K, sulla Sensibilità di misura Punto zero	 % mV	 < 0,08 tipico 0,04 < 3 tipico 2
Programmazione		
Linguaggio di programmazione		IEC61131−3
Memoria Dati del programma (volatile)	kByte	224
Memoria Dati del programma (non volatile)	kByte	16
Memoria Codice del programma (volatile) (2x disponibili per Online−Change)	kByte	2 x 160
Memoria Codice del programma (non volatile)	kByte	160
Memoria per Project−Sourcen (non volatile)	kByte	192
Frequenza di chiamata del programma IEC	Hz	2400, sincronizzata con la gestione val. misura dell'MGCplus
Numero dei sottocanali		1...128 (impostabile dall'utente)
Potenza di calcolo utilizzabile		30.000 operazioni Float / s oppure 150.000 operazioni Integer / s
Pannelli di connessione supportati		
Numero di pannelli di conness. controllabili		0; 1 oppure 2
Tipi di pannelli di connessione supportati		AP71 (2 interfaccia CAN) AP72 (2 interfaccia seriali) AP75 (8 IN digitali, 8 OUT digitali, livello 24 V) AP78 (8 uscite analogiche)
Meccanica		
Campo nominale di temperatura	°C	−20 ... +60
Campo della temperatura di esercizio	°C	−20 ... +60
Campo della temperatura di magazzinaggio	°C	−25 ... +70
Tensioni di esercizio	V	+14,6 ... +17,0 (< 90 mA) −14,6 ... −17,0 (< 100 mA) −7 ... −9 (<10 mA)
Formato schede	mm	Europa 100 x 160
Larghezza	mm	20,3 (4 U)
Spina di collegamento		DIN 41612 indiretta

17 Dati tecnici dei pannelli di connessione

17.1 AP71

Interfaccia CAN		
Numero delle interfaccia CAN		2
Baudrate	Baud	10 k, 20 k, 50 k, 125 k, 250 k, 500 k, 1 M
Lunghezza linea	m	1000, 1000, 1000, 500, 250, 100, 25
Accoppiamento hardware del Bus, selezionab. individualmente per ciascuna interfaccia CAN		ISO 11898−24 V “Low−Speed”
Tecnica di connessione		2x SubD a 9 poli, ognuna separata galvanicamente dall'alimentazione e da massa di misura
Meccanica		
Campo nominale di temperatura	°C	−20 ... +60
Campo della temperatura di esercizio	°C	−20 ... +60
Campo temperatura di magazzinaggio	°C	−25 ... +70
Tensioni di esercizio	V	+14,6 ... +17,0 (<130 mA) −14,6 ... −17,0 (<140 mA) −7 ... −9 (<10 mA)
Formato schede	mm	Europa 100 x 160
Larghezza	mm	20,3 (4 U)
Spina di collegamento		DIN 41612 indiretta

17.2 AP72

Interfaccia		
Baudrate	kBaud	9,6; 19,2; 38,4; 57,6; 115,2
Separazione galvanica	V	tipico 500
Tecnica di connessione		Presa Sub-D a 9 poli
Meccanica		
Campo nominale di temperatura	°C	−20 ... +60
Campo della temperatura di esercizio	°C	−20 ... +60
Campo temperatura di magazzinaggio	°C	−25 ... +70
Tensione di esercizio	V	+5 ... (< 100 mA)
Formato schede	mm	102 x 112
Larghezza	mm	20,3 (4 U)

17.3 AP75, AP78

ML78 + Pannello connessione		AP78	AP75
Uscite analogiche			
Max. numero uscite analogiche		10 (2 uscite filtrabili, di cui 1 accessibile anche dal pannello frontale dell' ML78)	2 (ambedue filtrabili, di cui 1 accessibile anche dal pannello frontale dell'ML78)
Separazione galvanica	V	tipico 500[1)]	–
Risoluzione conversione D/A	bit	16	
Sistema di massa		2[2)]	1, separato dal sistema di massa digitale
Tensione nominale	V	±10 asimmetrica	
Resistenza di carico ammessa	kΩ	≥ 5	
Resistenza interna	Ω	< 5	
Residuo alternato (76,6 kHz)	mV_{pp}	< 12	
Deriva a lungo termine (> 48h)	mV	< 3	
Influenza della temperatura ambiente, ogni 10 K	% mV	< 0,08; tipico 0,04 < 3; tipico 2	
Ingressi digitali			
Max. numero ingressi analogici			8 (16)[3)]
Campo della tensione ingresso	V		0 ... 24
Separazione galvanica	V		tipico 500
Potenziale Low	V		< 5
Potenziale High	V		>10
Sistemi di massa			1, separato dalla uscita digitale
Uscite digitali			
Max. numero uscite digitali			8 (16)[3)]
Campo tensione di uscita	V		0 ... 24
Corrente di uscita	A		0,5
Corrente di cortocircuito	A		1,5
Separazione galvanica	V		tipico 500
Tempo di reazione	ms		< 4
Sistemi di massa			1, separato dall'ingresso digitale
Alimentazione	V		24 (esterna)
Meccanica			
Campo nominale di temperatura	°C	–20 ... +60	
Campo temperatura di esercizio	°C	–20 ... +60	
Campo temp. di magazzinaggio	°C	–25 ... + 70	
Tensioni di esercizio	V	+14,6 ... +17,0 (< 120 mA)/ –17,0 ... –14,6 (< 120 mA) / –9,0 ... –7,0 (< 10 mA)	
Formato schede / Larghezza	mm	Europa 160 x 100 / 20,3 (4 U)	
Tecnica di connessione		Sub-D a 25 poli	Morsetti a vite innesetabili

1) Le uscite filtrabili non sono separate galvanicamente !

2) 1 sistema di massa per 2 uscite analogiche filtrabili ed 1 sistema di massa per le restanti 8 uscite analogiche

3) Usando 2 pannelli di connessione AP75: 16 ingressi digitali e 16 uscite digitali

4) Sui ognuno dei 2 pannelli di connessione sono disponibili ambedue le uscite analogiche VO1 e VO2

Riserva di modifica.
Tutti i dati descrivono i nostri prodotti in forma generica.
Pertanto essi non costituiscono alcuna garanzia formale e
non possono essere la base di alcuna nostra responsabilità.

HBM Italia srl

Via Pordenone, 8 I 20132 Milano - MI
Tel.: +39 0245471616; Fax: +39 0245471672
E-Mail: info@it.hbm.com ; support@it.hbm.com
Internet: www.hbm.com

measurement with confidence

A2000-1.0 it