4 Nozioni per lo sviluppo di strutture di programma

4.1 Programmi in una CPU

In una CPU vengono eseguiti due programmi distinti:

- il sistema operativo e
- il programma utente.

Sistema operativo

Il sistema operativo è contenuto in ogni CPU e organizza tutte le funzioni e le procedure della CPU che non sono legate a un compito di controllo specifico. I compiti del sistema operativo comprendono:

- gestione del nuovo avviamento (avviamento a caldo) e del riavviamento
- aggiornamento dell'immagine di processo degli ingressi ed emissione dell'immagine di processo delle uscite
- richiamo del programma utente
- rilevamento di allarmi e richiamo degli OB di allarme
- riconoscimento e gestione degli errori
- gestione delle aree di memoria
- comunicazione con dispositivi di programmazione e altri nodi di comunicazione.

Modificando i parametri del sistema operativo (preimpostazione del sistema operativo) è possibile influenzare il comportamento della CPU in determinate aree.

Programma utente

Il programma utente deve essere creato e caricato nella CPU. Il programma contiene tutte le funzioni necessarie per l'elaborazione dei vari compiti di automazione. I compiti del programma utente comprendono:

- la definizione delle condizioni per il nuovo avviamento (avviamento a caldo) e il riavviamento della CPU (p. es. la predefinizione dei segnali con un determinato valore)
- elaborazione dei dati di processo (p. es., combinazione di segnali binari, lettura e analisi di valori analogici; determinazione di segnali binari di uscita, emissione di valori analogici)
- reazione agli allarmi
- gestione di anomalie intervenute durante la normale esecuzione del programma.

4.2 Blocchi nel programma utente

4.2.1 Blocchi nel programma utente

Il software di programmazione STEP 7 offre la possibilità di strutturare il programma utente, ovvero di suddividerlo in singole sezioni indipendenti, ottenendo i seguenti vantaggi:

- i programmi di grandi dimensioni possono essere programmati in modo chiaro
- le singole parti del programma possono essere standardizzate
- l'organizzazione del programma viene semplificata
- le modifiche del programma si possono eseguire più facilmente
- il test del programma viene semplificato, poiché può essere eseguito per sezioni
- la messa in servizio viene facilitata.

Nell'esempio del processo di miscelazione industriale si è potuto vedere che un processo di automazione può essere suddiviso razionalmente in compiti distinti. Le sezioni di un programma utente strutturato corrispondondenti ai singoli compiti vengono definite blocchi di programma.

Tipi di blocchi

Vi sono diversi tipi di blocchi che possono essere utilizzati all'interno di programmi utenti S7:

Blocco	Descrizione della funzione	Vedere anche
Blocchi organizzativi (OB)	Gli OB determinano la struttura del programma utente:	"Blocchi organizzativi e struttura del programma"
Blocchi funzionali di sistema (SFB) e funzioni di sistema (SFC)	Gli SFB e le SFCs sono integrati nella CPU S7, e rendono accessibili alcune importanti funzioni di sistema.	"Blocchi funzionali di sistema (SFB) e funzioni di sistema (SFC)"
Blocchi funzionali (FB)	Gli FB sono blocchi con "memoria", programmabili dall'utente.	"Blocchi funzionali (FB)"
Funzioni (FC)	Le FC contengono routine di programma per le funzioni più utilizzate.	"Funzioni (FC)"
Blocchi dati di istanza (DB di istanza)	I blocchi dati di istanza vengono asseganti al blocco quando viene richiamato un FB/SFB. Essi vengono generati automaticamente nella compilazione.	"Blocchi dati di istanza"
Blocchi dati (DB)	I DB sono aree di dati per la memorizzazione dei dati utente. Oltre ai dati rispettivamente assegnati a un blocco funzionale, possono essere definiti dati globali utilizzabili da blocchi qualsiasi.	"Blocchi dati globali (DB)"

OB, FB, SFB, FC e SFC contengono parti del programma, e vengono pertanto definiti come blocchi di codice. Dipendono dalla CPU il numero ammesso di blocchi per ogni tipo di blocco e la loro lunghezza.

4.2.2 Blocchi organizzativi e struttura di programma

I blocchi organizzativi (OB) rappresentano l'interfaccia tra il sistema operativo e il programma utente. Essi vengono richiamati dal sistema operativo e comandano l'elaborazione ciclica del programma su interrupt, il comportamento di avvio del sistema di automazione e la gestione degli errori. Programmando i blocchi organizzativi è possibile determinare il comportamento della CPU.

Priorità di blocchi organizzativi

I blocchi organizzativi determinano la sequenza (eventi di avvio) in cui verranno elaborate le singole parti del programma. L'elaborazione di un OB può essere interrotta dal richiamo di un altro OB. È la priorità a stabilire quale OB può essere interrotto da un altro OB. Gli OB con priorità più alta interrompono quelli con priorità più bassa. L'OB 90 ha la priorità più bassa.

Tipi di allarmi e classi di priorità

Gli eventi di avvio che provocano il richiamo di un determinato OB vengono definiti anche allarmi o interrupt. La seguente tabella riporta i tipi di allarmi di STEP 7 e la priorità del blocco organizzativo assegnato. I blocchi organizzativi indicati e le loro classi di priorità non sono necessariamente presenti in tutte le CPU di S7 (vedere il manuale "Sistema di automazione S7-300, Configurazione e dati della CPU" e il manuale di riferimento "Sistema di automazione S7-400, M7-400, Caratteristiche delle unità modulari").

Tipo di allarme	**Blocchi organizzativi**	**Classe di priorità (predefinita)**	**Vedere anche**
Ciclo libero	OB 1	1	"Blocco organizzativo per l'elaborazione ciclica del programma (OB 1)"
Allarmi dall'orologio	da OB 10 a OB 17	2	"Blocchi organizzativi di allarme dall'orologio (da OB 10 a OB 17)"
Allarme di ritardo	OB 20 OB 21 OB 22 OB 23	3 4 5 6	"Blocchi organizzativi per l'allarme di ritardo (da OB 20 a OB 23)"
Schedulazione orologio	OB 30 OB 31 OB 32 OB 33 OB 34 OB 35 OB 36 OB 37 OB 38	7 8 9 10 11 12 13 14 15	"Blocchi organizzativi di schedulazione orologio (da OB 30 a OB 38)"
Interrupt di processo	OB 40 OB 41 OB 42 OB 43 OB 44 OB 45 OB 46 OB 47	16 17 18 19 20 21 22 23	"Blocchi organizzativi per interrupt di processo (da OB 40 a OB 47)"
Allarmi DPV1	OB 55 OB 56 OB 57	2 2 2	Programmazione di apparecchiature DPV1
Allarme di multicomputing	OB 60 Multicomputing	25	Funzionamento sincrono in multicomputing di diverse CPU

Tipo di allarme	Blocchi organizzativi	Classe di priorità (predefinita)	Vedere anche
Allarmi di sincronismo di clock	OB 61 OB 62 OB 63 OB 64	25	Progettazione di tempi di reazione del processo brevi e della stessa durata nel PROFIBUS DP
Errori di ridondanza	OB 70 Errore di periferia ridondata (solo nei sistemi H) OB 72 Errore di ridondanza CPU (solo nei sistemi H)	25 28	"Blocchi organizzativi per l'elaborazione degli errori (da OB 70 a OB 87 / da OB 121 a OB 122)"
Errori di asincronismo	OB 80 Errore di tempo OB 81 Errore di alimentazione OB 82 Allarme di diagnostica OB 83 Allarme di inserimento/estrazione OB 84 Errore di guasto hardware della CPU OB 85 Errore di esecuzione programma OB 86 Guasto del telaio di montaggio OB 87 Errore di comunicazione	25 (o 28, quando l'OB di errore di asincronismo si verifica nel programma di avviamento)	"Blocchi organizzativi per l'elaborazione degli errori (da OB 70 a OB 87 / da OB 121 a OB 122)"
Ciclo con priorità bassa	OB 90	29 [1)]	"Blocco organizzativo di priorità bassa (OB 90)"
Avviamento	OB 100 Nuovo avviamento (avviamento a caldo) OB 101 Riavviamento OB 102 Avviamento a freddo	27 27 27	"Blocchi organizzativi per l'avviamento (OB 100/OB 101/OB 102)"
Errori sincroni	OB 121 Errore di programmazione OB 122 Errore di accesso	Priorità dell'OB che causa l'errore	"Blocchi organizzativi per l'elaborazione degli errori (da OB 70 a OB 87 / da OB 121 a OB 122)"
1) Alla classe di priorità 29 corrisponde la priorità 0.29. L'OB 90 ha quindi priorità inferiore a quella del ciclo libero.			

Modifica della priorità

Gli allarmi e gli interrupt sono parametrizzabili con STEP 7. La parametrizzazione permette p. es. di deselezionare OB di allarme o classi di priorità nei blocchi di parametri: allarme dall'orologio, allarme di ritardo, schedulazione orologio e interrupt di processo.

Nelle CPU S7-300 la priorità dei blocchi organizzativi è assegnata in modo fisso.

Nelle CPU S7-400 (e nella CPU 318) l'utente può modificare con STEP 7 la priorità dei seguenti blocchi organizzativi.

- da OB 10 a OB 47
- da OB 70 a OB 72 (solo CPU dei sistemi H) e da OB 81 a OB 87 nello stato di funzionamento RUN.

Sono ammesse:

- da OB 10 a OB 47 le classi di priorità da 2 a 23,
- da OB 70 a OB 72 le classi di priorità da 2 a 28,
- da OB 81 a OB 87 le classi di priorità da 24 a 26; le CPU costruite a partire da metà 2001 (versione di firmware 3.0) offrono possibilità più ampie: per gli OB da 81 a 84 nonché per gli OB 86 e 87, si possono impostare le classi di priorità da 2 a 26.

A OB differenti può essere assegnata la stessa priorità. In questo caso, gli OB vengono elaborati nella sequenza in cui si presentano i relativi eventi di avvio.

Gli OB di errore, avviati in caso di errori di sincronismo, vengono elaborati con la medesima classe di priorità del blocco la cui elaborazione era in corso quando è stato riconosciuto l'errore.

Dati locali

Con la creazione dei blocchi di codice (OB, FC, FB) è possibile determinare una serie di dati locali temporanei. L'area dei dati locali a disposizione nella CPU è suddivisa tra le classi di priorità.

Nelle CPU S7-400 è possibile modificare con STEP 7 nel blocco parametri "Classi di priorità" il numero dei dati locali per ogni classe di priorità.

Informazione di avvio di un OB

Ogni blocco organizzativo dispone di un’informazione di avvio di 20 byte di dati locali, che il sistema operativo trasferisce all'avvio dell'OB. L'informazione di avvio contiene informazioni su evento di avvio dell'OB, data e ora di avvio, errori verificatisi ed eventi di diagnostica.

Un OB 40 (interrupt di processo) contiene, p. es. nell'informazione di avvio, l'indirizzo dell'unità che genera l'interrupt.

OB di allarme annullati

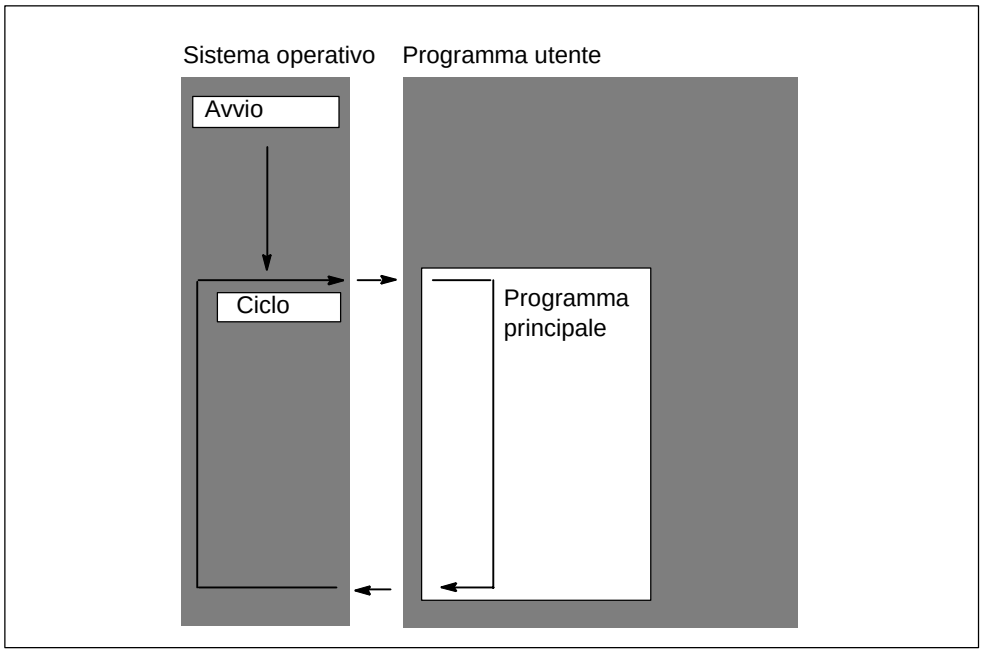
Se si imposta la classe di priorità 0 o si assegnano a una classe di priorità meno di 20 byte di dati locali, il relativo OB di allarme viene annullato. Gli OB di allarme annullati:

- non possono essere copiati nello stato di funzionamento RUN, o inseriti nel programma utente
- possono essere copiati nello stato di funzionamento STOP o inseriti nel programma utente, ma causano, al nuovo avviamento (avviamento a caldo) della CPU, un'interruzione del processo di avviamento e danno luogo ad una registrazione nel buffer di diagnostica.

Annullando OB di allarme non necessari, si aumenta l'area dei dati locali disponibile, che può quindi essere utilizzata per memorizzare dati temporanei in altre classi di priorità.

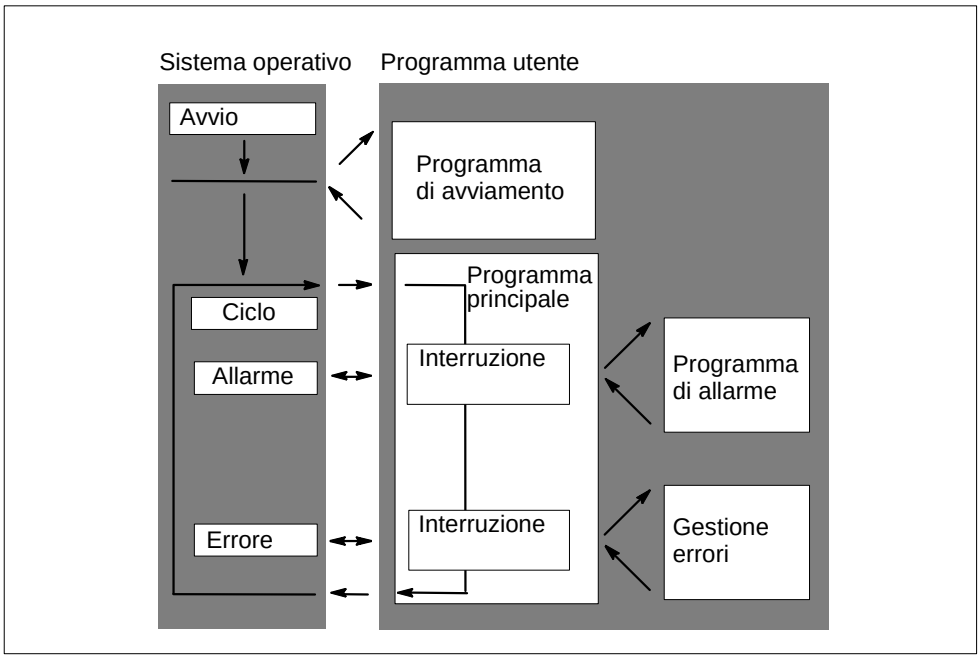
Elaborazione ciclica del programma

Nei controllori programmabili l'elaborazione ciclica del programma rappresenta l'elaborazione prevalente: il sistema operativo viene eseguito in un loop di programma (detto ciclo), e richiama quindi in ogni ciclo per una volta il blocco organizzativo OB 1 nel programma principale. Anche il programma utente nell'OB 1 viene elaborato in modo ciclico.



Elaborazione del programma comandata da evento

L'elaborazione ciclica del programma può essere interrotta mediante determinati eventi di avvio (interrupt). Se sopraggiunge uno di tali eventi, il blocco appena elaborato viene interrotto tra un comando e l'altro, e viene elaborato un altro blocco organizzativo assegnato all'evento di avvio. In seguito, l'elaborazione ciclica del programma riprende dal punto in cui è avvenuta l'interruzione.

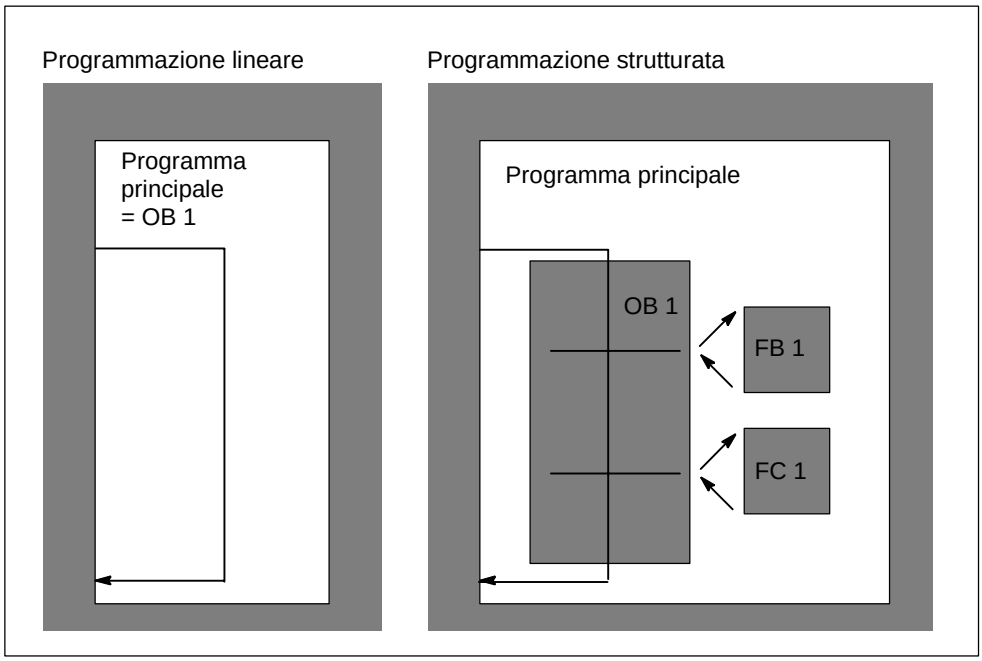


STEP 7 offre la possibilità di elaborare solo in caso di necessità quelle parti del programma utente che non devono essere elaborate ciclicamente. Il programma utente può essere scomposto in segmenti, e suddiviso in diversi blocchi organizzativi. Se il programma utente deve reagire a un segnale importante che capita relativamente di rado, (p. es. se un trasduttore di valore limite per la misurazione del grado di riempimento di un serbatoio segnala che il serbatoio è pieno), il segmento di programma da eseguire all'emissione del segnale potrà trovarsi in un OB che viene eseguito in modo comandato dall'evento.

Programmazione lineare o strutturata

È possibile scrivere l'intero programma utente nell'OB 1 (programmazione lineare). Questa operazione è consigliabile solo per programmi semplici che girano sulle CPU S7-300 occupando poca memoria.

I compiti di automazione complessi possono essere elaborati meglio se vengono suddivisi in compiti parziali più piccoli, che corrispondono alle funzioni tecnologiche del processo di automazione o che devono essere utilizzati più volte. Nel programma utente, i compiti parziali sono rappresentati dai correspondenti segmenti di programma, ossia dai blocchi (programmazione strutturata).



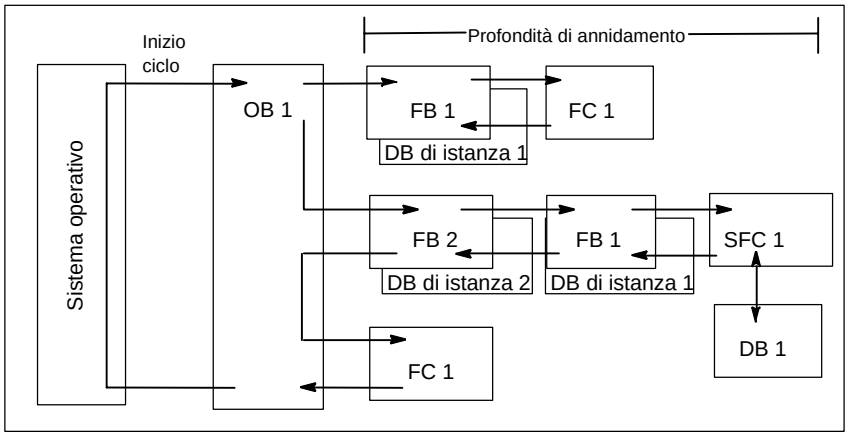
4.2.3 Gerarchia di richiamo nel programma utente

Affinché il programma utente funzioni, occorre richiamare i blocchi dei quali esso è composto. Ciò avviene mediante delle speciali operazioni di STEP 7, i richiami dei blocchi, che possono essere programmate e avviate solo all'interno di blocchi di codice.

Sequenza e profondità di annidamento

La sequenza e l'annidamento dei richiami dei blocchi costituiscono la gerarchia di richiamo. La profondità di annidamento ammessa dipende dalla CPU.

La seguente figura mostra in base a un esempio la sequenza e l'annidamento dei richiami di blocchi entro un ciclo di esecuzione.



Regole per la sequenza di creazione dei blocchi

- I blocchi vengono creati dall'alto in basso; si comincia quindi con la serie di blocchi superiore.
- Ogni blocco che viene richiamato deve essere essere già esistente; all'interno di una serie di blocchi la direzione in cui vengono creati va quindi da destra a sinistra.
- Viene creato da ultimo l'OB 1.

Una volta eseguite tali regole, esse comportano questa sequenza di creazione per l'esempio della figura:

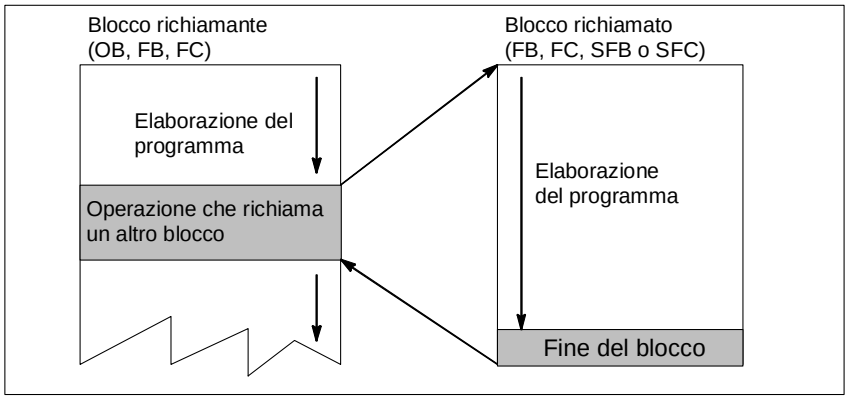
FC 1 > FB 1 + DB di istanza 1 > DB 1 > SFC 1 > FB 2 + DB di istanza 2 > OB 1

Avvertenza

Superata una determinata profondità di annidamento, lo stack di dati locali può essere eccedente (vedere anche Stack di dati locali).

Richiami dei blocchi

La figura seguente mostra la procedura di richiamo di un blocco all'interno del programma utente: il programma richiama il secondo blocco, le cui operazioni vengono completamente elaborate. Al termine dell'elaborazione del blocco richiamato, l'elaborazione del blocco richiamante viene ripresa a partire dall'operazione successiva al richiamo del blocco.



Prima di programmare un blocco, è necessario stabilire con quali dati deve avvenire l'elaborazione del programma; ciò significa che occorre dichiarare le variabili del blocco.

Avvertenza

- I parametri OUT devono essere descritti per ogni richiamo di blocco.
- Il sistema operativo resetta le istanze dell'SFB 3 "TP" in caso di avviamento a freddo. Se le istanze di tale SFB devono essere inizializzate dopo il nuovo avviamento (avviamento a caldo) occorre che l'utente richiami le istanze da inizializzare dell'SFB con PT = 0 ms. Ciò è realizzabile p. es. mediante una routine di inizializzazione nei blocchi che contengono le istanze di tale SFB.

4.2.4 Tipi di blocchi

4.2.4.1 Blocco organizzativo per l'elaborazione ciclica del programma (OB 1)

L'elaborazione ciclica del programma è l'elaborazione "normale" nei controllori programmabili. Il sistema operativo richiama ciclicamente l'OB 1 e con esso avvia l'elaborazione ciclica del programma utente.

Esecuzione dell'elaborazione ciclica del processo

La seguente tabella riporta le fasi dell'elaborazione ciclica del programma.

Fase	Esecuzione nelle CPU fino a 10/98	Esecuzione nelle CPU a partire da 10/98
1	Il sistema operativo avvia il tempo di controllo del ciclo.	Il sistema operativo avvia il tempo di controllo del ciclo.
2	La CPU legge lo stato degli ingressi nelle unità di ingresso, e aggiorna l'immagine di processo degli ingressi.	La CPU scrive i valori dall'immagine di processo delle uscite nelle unità di uscita.
3	La CPU elabora il programma utente ed esegue le operazioni specificate nel programma.	La CPU legge lo stato degli ingressi nelle unità d'ingresso, e aggiorna l'immagine di processo degli ingressi.
4	La CPU scrive nelle unità di uscita i valori dell'immagine di processo delle uscite.	La CPU elabora il programma utente ed esegue le operazioni specificate nel programma.
5	Alla fine di un ciclo il sistema operativo esegue i compiti previsti, p. es. caricamento e cancellazione di blocchi, ricezione e invio di dati globali.	Alla fine di un ciclo il sistema operativo esegue i compiti previsti, p. es. caricamento e cancellazione di blocchi, ricezione e invio di dati globali.
6	A questo punto, la CPU ritorna all'inizio del ciclo e avvia nuovamente il tempo di controllo del ciclo.	A questo punto, la CPU ritorna all'inizio del ciclo e avvia nuovamente il tempo di controllo del ciclo.

Immagini di processo

Affinché per l'intera durata dell'elaborazione ciclica del programma la CPU abbia a disposizione un'immagine coerente dei segnali di processo, la CPU, per interpellare le aree degli operandi ingresso (E) e uscita (A), non accede direttamente alle unità di ingresso/uscita, bensì ad un'area di memoria interna in cui si trova un'immagine degli ingressi e delle uscite.

Programmazione dell'elaborazione ciclica del programma

La programmazione dell'elaborazione ciclica del programma avviene durante la scrittura del programma utente con STEP 7 nell'OB 1 e nei blocchi richiamati.

L'elaborazione ciclica del programma inizia non appena il programma utente si è concluso senza errori.

Possibilità di interruzione

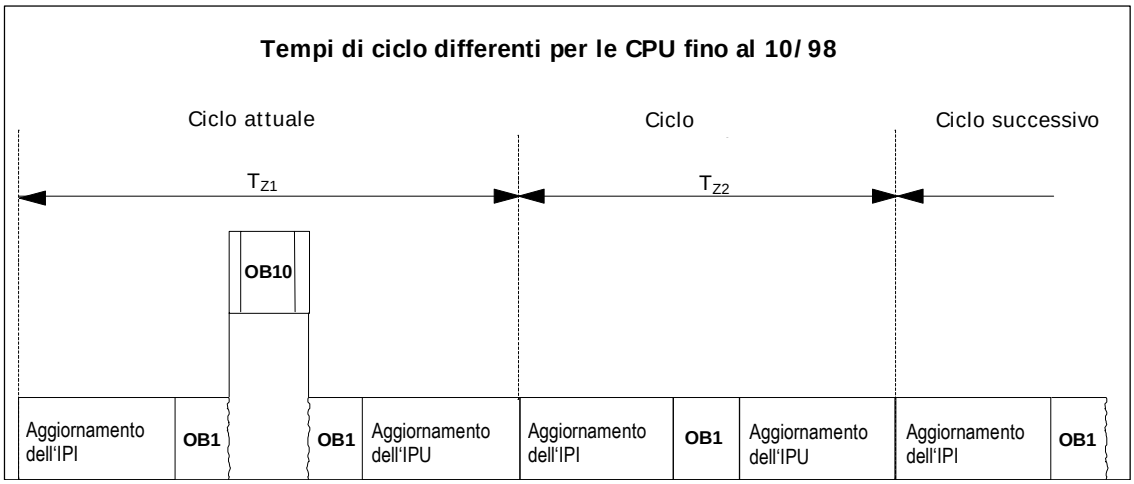
L'elaborazione ciclica del programma può essere interrotta mediante:

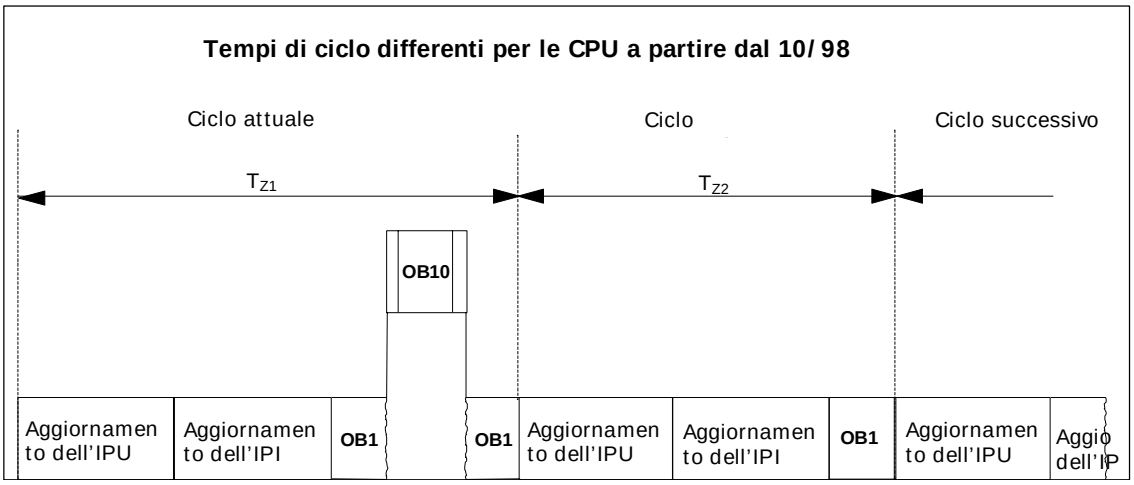
- un interrupt o allarme
- un comando di STOP (selettore dei modi operativi, comando di menu dal PG, SFC46 STP, SFB 46 20 STOP)
- la mancanza di tensione di rete
- un errore di un dispositivo o del programma.

Tempo di ciclo

Il tempo di ciclo è il tempo che richiede il sistema operativo per l'elaborazione del programma ciclico nonché di tutti le parti di programma che interrompono tale ciclo (p. es. elaborazione di altri blocchi organizzativi) e per le attività di sistema (p. es. aggiornamento dell'immagine di processo). Questo tempo viene controllato.

Il tempo di ciclo (TZ) non è uguale per ogni ciclo. Le figure seguenti riportano tempi di ciclo differenti (TZ1 ≠ TZ2) per le CPU fino al 10/98 e per quelle a partire dal 10/98.





Nel ciclo attuale, l'OB 1 viene interrotto da un allarme dall'orologio.

Tempo di controllo del ciclo

Con STEP 7 è possibile modificare il tempo di controllo di ciclo preimpostato. Allo scadere di questo tempo, la CPU va in STOP oppure viene richiamato l'OB 80, nel quale si può stabilire la modalità di reazione della CPU all'errore di tempo.

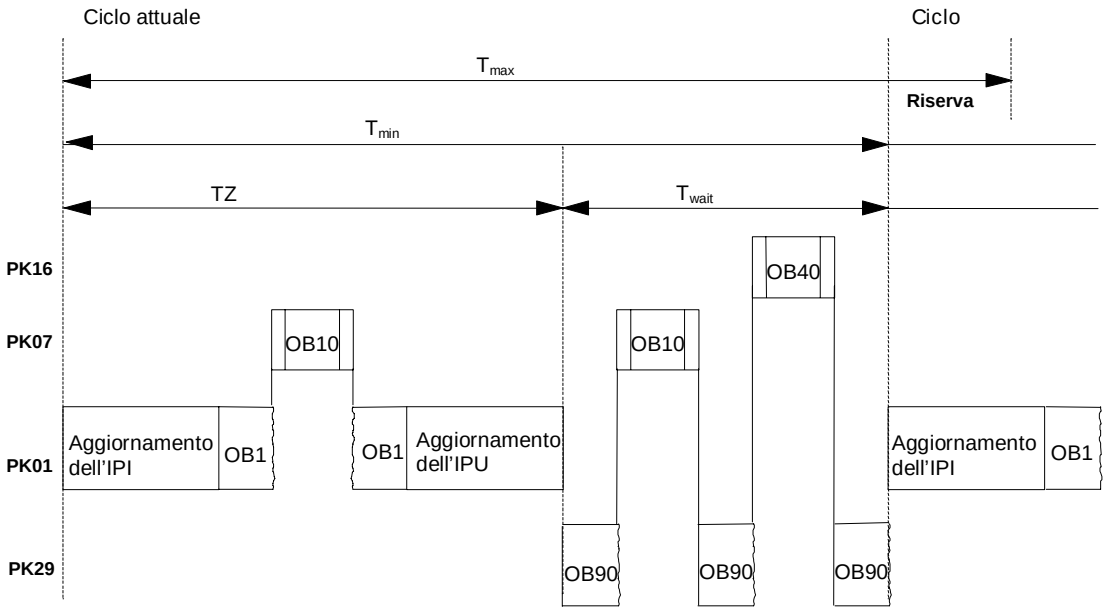
Tempo di ciclo minimo

Nelle CPU S7-400 e nella CPU 318 è possibile impostare con STEP 7 un tempo di ciclo minimo. Ciò può servire se:

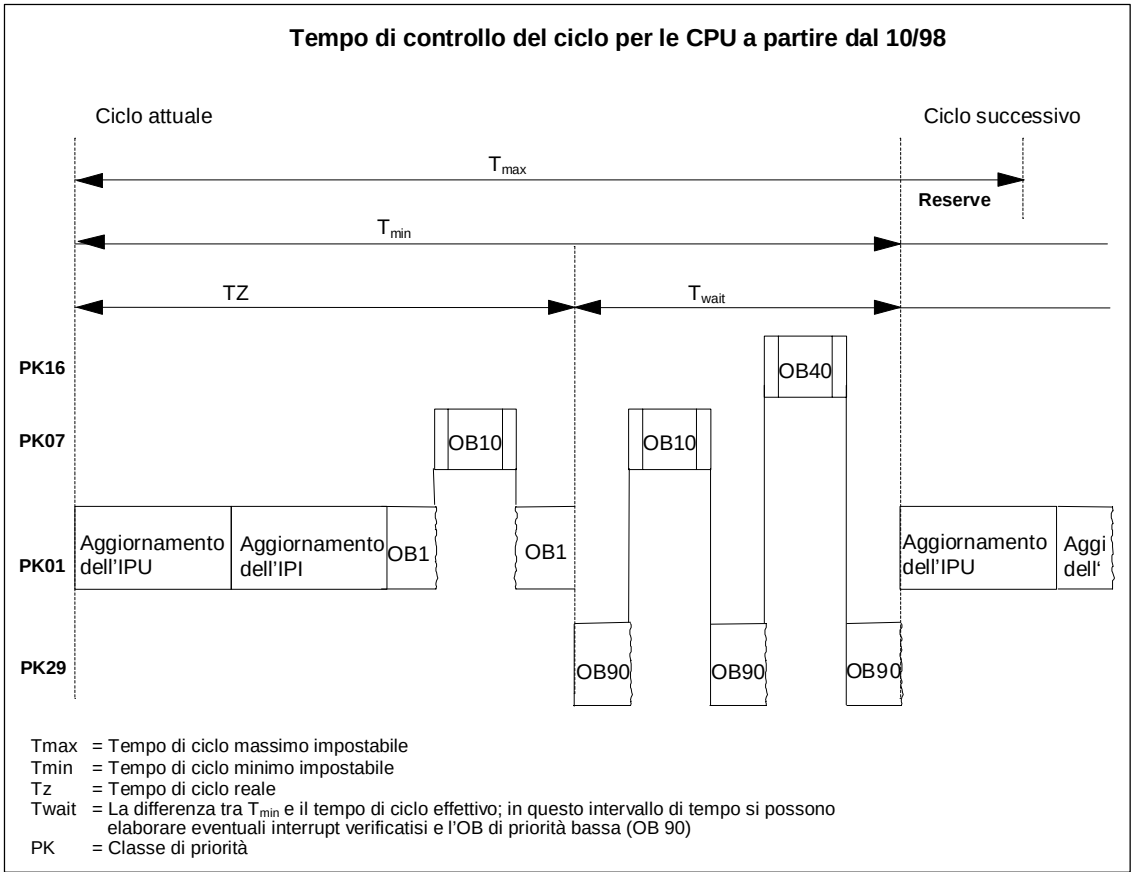
- sono previsti intervalli di tempo tra gli avvii dell'elaborazione del programma dell'OB 1 (ciclo libero) di uguale durata, oppure
- a fronte di un tempo di ciclo troppo breve, l'aggiornamento delle immagini di processo si verificherebbe troppo spesso.

Le figure seguenti riportano la funzione del tempo di controllo del ciclo nell'esecuzione del programma per le CPU fino al 10/98 e per quelle a partire dal 10/98.

Tempo di controllo del ciclo per le CPU fino al 10/98



Tmax = Tempo di ciclo massimo impostabile
Tmin = Tempo di ciclo minimo impostabile
Tz = Tempo di ciclo reale
Twait = La differenza tra T_{min} e il tempo di ciclo effettivo; in questo intervallo di tempo si possono elaborare eventuali interrupt verificatisi e l'OB di priorità bassa (OB 90)
PK = Classe di priorità



Aggiornamento dell'immagine di processo

Nell'elaborazione ciclica del programma della CPU l'immagine di processo si aggiorna automaticamente. Nelle CPU S7-400 e nella CPU 318 è possibile disattivare l'aggiornamento dell'immagine di processo se

- si intende invece accedere direttamente alla periferia, oppure se
- si vogliono aggiornare una o più immagini di processo degli ingressi e delle uscite in un altro momento utilizzando le funzioni di sistema SFC 26 UPDAT_PI e SFC 27 UPDAT_PO.

Carico di comunicazione

Con il parametro CPU "Carico del ciclo a causa della comunicazione" è possibile comandare entro certi limiti la durata dei processi di comunicazione che allungano sempre anche il tempo di ciclo. I processi di comunicazione possono essere p.es.: trasferimento di dati a un'altra CPU mediante MPI oppure caricamento di blocchi che è stato attivato mediante PG.

Le funzioni di test con il PG vengono appena influenzate da questo parametro, possono tuttavia allungare notevolmente il tempo di ciclo. Nel funzionamento di processo è possibile limitare il tempo messo a disposizione per le funzioni di test (solo S7-300).

Principio di funzionamento del parametro

Il sistema operativo della CPU mette costantemente a disposizione della comunicazione la percentuale progettata della capacità di elaborazione complessiva della CPU (funzionamento con suddivisione dei tempi). Se questa capacità di elaborazione non è necessaria per la comunicazione, può essere allora utilizzata per l'elaborazione restante.

Influenza sul tempo di ciclo effettivo

Senza eventi di asincronismo addizionali il tempo di ciclo di OB 1 si allunga di un fattore calcolabile secondo la formula seguente:

$$\frac{100}{100 - \text{"Carico del ciclo a causa della comunicazione (\%)"}}$$

Esempio 1 (nessun evento di asincronismo addizionale):

Impostando un carico del ciclo a causa della comunicazione del 50%, si può verificare un raddoppiamento del tempo di ciclo di OB1.

Allo stesso tempo il tempo di ciclo di OB 1 viene influenzato ancora da eventi di asincronismo (p.es. interrupt di processo oppure schedulazioni orologio) . Con l'allungamento del tempo di ciclo a causa della componente di comunicazione si verificano da un punto di vista statistico un numero ancora maggiore di eventi di asincronismo all'interno di un ciclo OB 1. Ciò allunga ulteriormente il ciclo di OB 1. Questo allungamento dipende da quanti eventi per ciclo di OB 1 compaiono e dalla durata dell'elaborazione dell'evento.

Esempio 2 (vengono considerati eventi di asincronismo addizionali):

Con un vero tempo di esecuzione di OB 1 di 500 ms si può verificare un tempo di ciclo effettivo di fino a 1000 ms a causa di un carico di comunicazione di 50 % (purché la CPU abbia sempre sufficienti job di comunicazione da elaborare). Se parallelamente viene eseguito ogni 100 ms una schedulazione orologio con tempo di elaborazione di 20 ms, allora questo avrebbe come effetto un allungamento complessivo del ciclo pari a 5*20 ms = 100 ms senza carico di comunicazione, ovvero il tempo di ciclo effettivo sarebbe pari a 600 ms. Poiché una schedulazione orologio interrompe anche la comunicazione, con un carico di comunicazione di 50% la schedulazione orologio influisce sul tempo ciclo per un valore pari a 10 * 20 ms, vale a dire in questo caso il tempo di ciclo effettivo non ammonta a 1000 ms ma a 1200 ms.

Avvertenze

- Verificare gli effetti di una modifica del valore del parametro "Carico del ciclo a causa della comunicazione" nel funzionamento dell'impianto.
- Occorre tener conto del carico di comunicazione quando si imposta il tempo di ciclo minimo; possono altrimenti verificarsi degli errori di tempo.

Suggerimenti

- Acquisire se possibile il valore preimpostato
- Aumentare il valore solo se la CPU viene impiegata principalmente per finalità di comunicazione e il programma utente presenta acriticità temporale !
- In tutti gli altri casi ridurre solo il valore !
- Impostare il funzionamento di processo (solo con S7-300) e limitare il tempo lì necessario per le funzioni di test!

4.2.4.2 Funzioni (FC)

Le funzioni fanno parte dei blocchi programmati dall'utente. La funzione è un blocco di codice "privo di memoria". Le variabili temporanee dell'FC vengono memorizzate nello stack dei dati locali. Dopo l'elaborazione dell'FC, questi dati vanno perduti. Per la memorizzazione dei dati, le funzioni possono utilizzare blocchi dati globali.

Poiché a una FC non è abbinata alcuna memoria, è necessario indicarne sempre i parametri attuali. Ai dati locali di una FC non può essere assegnato alcun valore iniziale.

Campo di applicazione

Una FC contiene un programma che viene eseguito ogni qualvolta che l'FC viene richiamata da un altro blocco di codice. Le funzioni possono essere utilizzate per

- restituire un valore di funzione al blocco richiamante (per esempio funzioni matematiche).
- eseguire una funzione tecnologica (per esempio controllo singolo con combinazione binaria).

Assegnazione di parametri attuali a parametri formali

Il parametro formale è il segnaposto del parametro "effettivo", il parametro attuale. I parametri attuali sostituiscono i parametri formali al richiamo dell'FC. Ai parametri formali di una FC devono quindi essere sempre asseganti parametri attuali (p.es. al parametro formale "Start" il parametro attuale "E3.6"). I parametri di ingressi, uscite e ingressi/uscite utilizzati dall'FC sono memorizzati come puntatori ai parametri attuali del blocco di codice che ha richiamato l'FC.

Differenze importanti tra i parametri di uscita di FC e FB

Per quanto riguarda i blocchi funzionali (FB), in caso di accesso ai parametri viene utilizzata la copia del parametro attuale nel DB di istanza. Se nel richiamare un FB non viene assegnato un parametro di ingresso o se non viene scritto un parametro di uscita nel blocco, vengono ancora utilizzati i valori meno recenti presenti nel DB di istanza (DB di istanza = memoria dell'FB).

Le funzioni (FC) non hanno una memoria. L'assegnazione dei parametri formali perciò, a differenza degli FB, non è opzionale ma obbligatoria. L'accesso a parametri di FC si esegue tramite indirizzi (puntatori multiarea). Utilizzando come parametro attuale un operando dell'area dati (blocco dati) o una variabile locale del blocco richiamante, per l'assegnazione dei parametri viene memorizzata temporaneamente una copia del parametro attuale nei dati locali del blocco richiamante.

Attenzione

Se in un caso simile non viene scritto un parametro OUTPUT in una FC, i valori emessi potrebbero essere casuali.

L'area resa disponibile per la copia nei dati locali del blocco richiamante non viene scritta per mancanza di assegnazione del parametro OUTPUT e resta quindi invariata. Di conseguenza, il valore presente in questa area viene emesso casualmente poiché i dati locali non sono preimpostati, p. es. con 0.

Osservare quanto segue:

- Inizializzare se possibile i parametri OUTPUT.
- I comandi di impostazione e resettaggio dipendono dal risultato logico combinatorio. Se con questi comandi si determina il valore di un parametro OUTPUT, con RLC = 0 non viene calcolato alcun valore.
- Assicurarsi che, al di là di tutti i possibili percorsi di programma all'interno del blocco, i parametri OUTPUT vengano scritti in ogni caso. Osservare in particolar modo i comandi di salto così come l'uscita ENO in KOP e FUP, nonché il SeS e gli effetti dei comandi MCR.

Avvertenza

Per quanto riguarda i parametri OUTPUT di un FB o i parametri INOUT di una FC e di un FB, i valori emessi non possono essere casuali poiché in questo caso il valore di uscita o quello di ingresso precedenti vengono mantenuti senza sovrascrivere i parametro. Tuttavia, anche in questo caso, è necessario osservare le indicazioni di cui sopra per non elaborare involontariamente valori "vecchi".

4.2.4.3 Blocchi funzionali (FB)

I blocchi funzionali fanno parte dei blocchi programmati dall'utente. Un blocco funzionale è un blocco "con memoria". Esso dispone di un blocco dati correlato come memoria (blocco dati di istanza). Sia i parametri che vengono trasmessi all'FB, sia le variabili statiche vengono memorizzati nel blocco dati di istanza. Le variabili temporanee vengono memorizzate nello stack dei dati locali.

Al termine dell'elaborazione dell'FB, i dati memorizzati nel DB di istanza non vanno perduti, come invece accade a quelli memorizzati nello stack dei dati locali.

Avvertenza

Per evitare errori durante l'uso degli FB leggere nell’appendice il paragrafo Tipi di dati ammessi nel trasferimento dei parametri.

Campo di applicazione

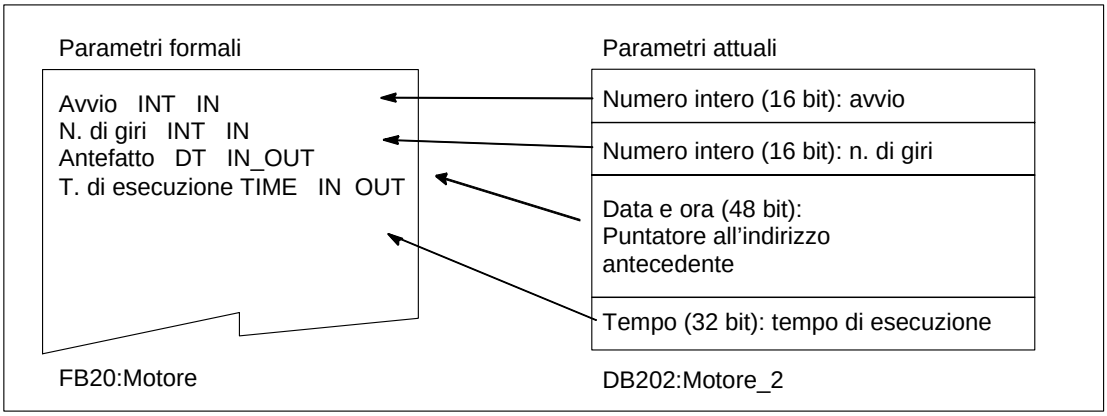
Un FB contiene un programma che viene eseguito ogni qualvolta che l'FB viene richiamato da un altro blocco di codice. I blocchi funzionali facilitano la programmazione delle funzioni frequentemente ricorrenti e complesse.

FB e DB di istanza

A ogni richiamo di un blocco funzionale che trasferisce parametri viene assegnato un blocco dati di istanza.

Con il richiamo di diverse istanze di un FB si possono comandare più apparecchiature con un solo FB. Per esempio, usando dati di istanza differenti per i vari motori, l'FB per un solo tipo di motore può comandare più motori. I dati per ogni singolo motore (per esempio numero di giri, tempo di rampa, ore di funzionamento accumulate, ecc.) possono essere memorizzati in uno o più blocchi dati di istanza.

La figura seguente mostra i parametri formali di un FB che sostituiscono i parametri attuali, memorizzati nel blocco dati di istanza.



Variabili del tipo di dati FB

Se il programma utente è strutturato in modo tale che in un FB vengano richiamati altri blocchi funzionali già esistenti, con l'FB del tipo di dati sarà possibile registrare gli FB da richiamare, come variabili statiche, nella tabella di dichiarazione delle variabili dell'FB richiamante. In questo modo, si ottiene un annidamento di variabili e la concentrazione dei dati di istanza in un blocco dati di istanza (multistanza).

Assegnazione di parametri attuali a parametri formali

In genere in STEP 7 non è necessario assegnare i parametri attuali ai parametri formali di un FB. Esistono tuttavia delle eccezioni. I parametri attuali devono essere assegnati:

- a un parametro ingressi/uscite (di transito) di un tipo di dati composto (p. es. STRING, ARRAY oppure DATE_AND_TIME)
- a tutti i tipi di parametri (p. es. TIMER, COUNTER o POINTER)

STEP 7 assegna i parametri attuali ai parametri formali di un FB:

- *quando nell'istruzione di richiamo vengono definiti dei parametri attuali:* le operazioni dell'FB usano i parametri attuali a disposizione.
- *quando nell'istruzione di richiamo non viene definito alcun parametro attuale:* le operazioni dell'FB usano i valori memorizzati nel DB di istanza.

La tabella seguente mostra a quali variabili dell'FB devono essere assegnati i parametri attuali.

Variabili		**Tipo di dati**	
	Tipo di dati semplice	**Tipo di dati composti**	**Tipo di parametro**
Ingresso	Parametro non necessario	Parametro non necessario	Parametro attuale necessario
Uscita	Parametro non necessario	Parametro non necessario	Parametro attuale necessario
Ingr./uscite	Parametro non necessario	Parametro attuale necessario	–

Assegnazione di valori iniziali a parametri formali

E possibile assegnare valori iniziali ai parametri formali nella parte di dichiarazione dell'FB. Tali valori vengono trasferiti nel DB di istanza assegnato all'FB.

Se nell'istruzione di richiamo ai parametri formali non viene assegnato alcun parametro attuale, STEP 7 usa i valori memorizzati nel DB di istanza. Tali dati possono essere i valori iniziali specificati nella tabella di dichiarazione delle variabili dell'FB.

La tabella seguente mostra quali variabili si possono assegnare a un valore iniziale. Siccome dopo l'elaborazione del blocco i dati temporanei non vengono memorizzati, ad essi non può essere assegnato alcun valore.

	Tipo di dati		
Variabili	**Tipo di dati semplici**	**Tipo di dati composti**	**Tipo di parametro**
Ingresso	Valore iniziale ammesso	Valore iniziale ammesso	–
Uscita	Valore iniziale ammesso	Valore iniziale ammesso	–
Ingr./uscite	Valore iniziale ammesso	–	–
Statiche	Valore iniziale ammesso	Valore iniziale ammesso	–
Temporan.	–	–	–

4.2.4.4 Blocchi dati di istanza

A ogni richiamo di un blocco funzionale che trasferisce parametri viene assegnato un blocco dati di istanza. Nel DB di istanza vengono memorizzati i parametri attuali e i dati statici dell'FB. Le variabili dichiarate nell'FB determinano la struttura del blocco dati di istanza. Con il termine istanza si definisce il richiamo di un blocco funzionale. Per esempio, se un blocco funzionale viene richiamato cinque volte nel programma utente S7, di tale blocco esisteranno cinque istanze.

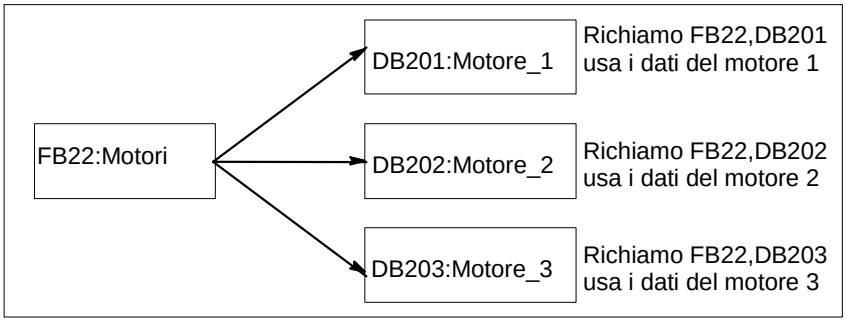
Creazione di un DB di istanza

Per poter creare un blocco dati di istanza, deve già esistere l'FB a cui il blocco deve essere assegnato. Il numero dell'FB viene stabilito al momento della creazione del blocco dati di istanza.

Un DB di istanza per ogni istanza

Se a un blocco funzionale (FB) che controlla un motore vengono assegnati più blocchi dati di istanza, l'FB può essere usato per il controllo di diversi motori.

I dati dei singoli motori (p. es. numero di giri, tempo di rampa, ore di esercizio totali) vengono memorizzati nei diversi blocchi dati. A seconda del DB assegnato all'FB richiamato, è possibile controllare un altro motore. In questo modo, un solo blocco funzionale è sufficiente per più motori (vedere la figura seguente).

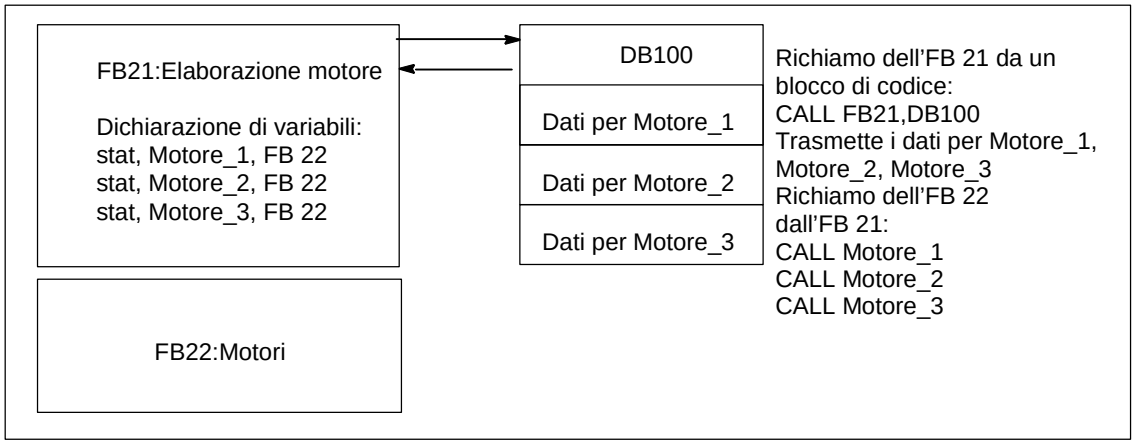


Un solo DB di istanza per più istanze di un FB

A un FB possono essere trasmessi dati di istanza di diversi motori contenuti in un DB di istanza. Per farlo, occorre richiamare i controlli del motore in un altro FB. Inoltre è necessario dichiarare nella parte di dichiarazione dell'FB richiamante le variabili statiche per le singole istanze (multistanze) con il tipo di dati dell'FB.

Usando un solo DB di istanza per più istanze di un FB, si risparmia memoria e si ottimizza l'uso dei blocchi dati.

Nella figura seguente, per esempio, l'FB chiamante è l'FB 21 "Elaborazione del motore", le variabili sono del tipo di dati FB 22 e le istanze vengono definite Motore_1, Motore_2 e Motore_3.



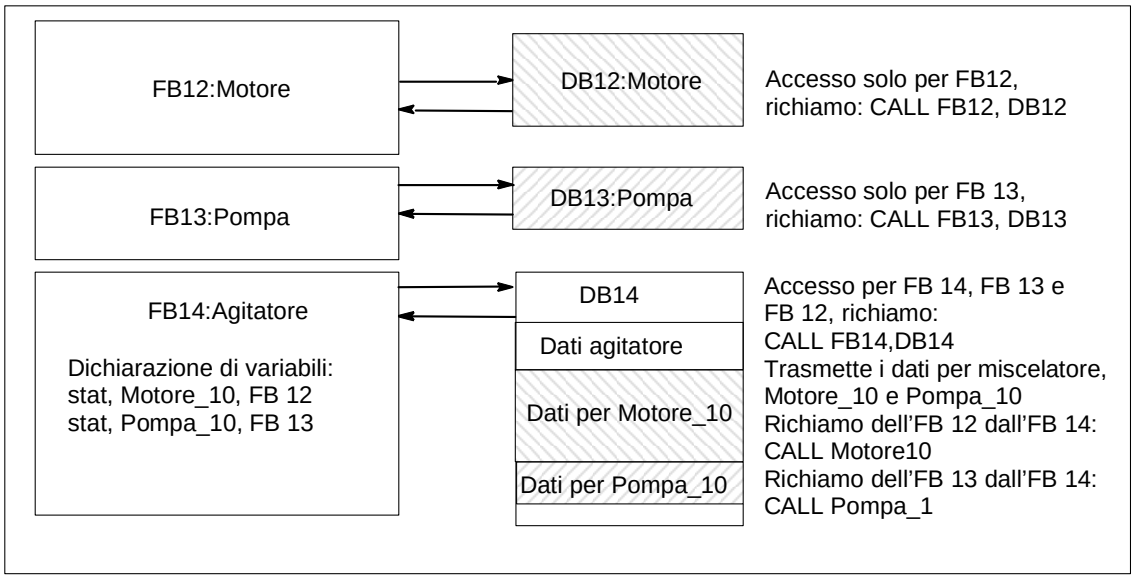
Nell'esempio, l'FB 22 non richiede alcun blocco dati di istanza, poiché i suoi dati di istanza sono memorizzati nel blocco dati di istanza dell'FB richiamante.

Un solo DB di istanza per più istanze di diversi FB (multiistanze)

In un blocco funzionale possono essere richiamate istanze di altri FB già creati. I dati di istanza necessari a tal fine possono essere assegnati al blocco dati di istanza dell'FB richiamante, ovvero in questo caso non si ha bisogno di blocchi dati supplementari per gli FB richiamati.

Per quanto riguarda tali multistanze contenute in un DB di istanza, è necessario dichiarare per ciascuna istanza, nella parte di dichiarazione dell'FB richiamante, variabili statiche con il tipo di dati dell'FB richiamato. Il richiamo all'interno dell'FB viene effettuato senza specificare un DB di istanza, ma semplicemente con il nome della variabile.

Nell'esempio della figura, i dati di istanza assegnati vengono memorizzati insieme in un blocco dati di istanza.



4.2.4.5 Blocchi dati globali (DB)

A differenza dei blocchi di codice, i blocchi dati non contengono istruzioni STEP 7, ma servono alla registrazione dei dati utente. Nei blocchi dati sono quindi compresi i dati con cui opera il programma utente. I blocchi dati globali servono alla registrazione di dati utente che possono essere usati da tutti gli altri blocchi.

La dimensione dei DB è variabile. Per la dimensione massima ammessa, vedere le descrizioni della CPU /70/ e /101/.

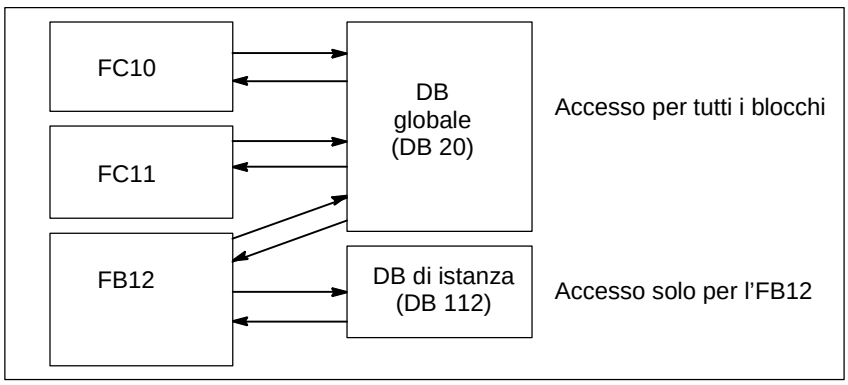
La struttura dei blocchi dati globali può essere stabilita liberamente.

Blocchi dati globali nel programma utente

Quando un blocco di codice (FC, FB, OB) viene richiamato, esso può temporaneamente occupare spazio in memoria nell'area dei dati locali (L-Stack). Un blocco di codice può anche aprire un'area di memoria sotto forma di un DB. Al contrario dei dati dell'area di dati locali, i dati contenuti in un DB non vengono cancellati quando si chiude il DB o al termine dell'elaborazione del relativo blocco di codice.

Ogni FB, FC o OB ha accesso di lettura/scrittura a un DB globale. I dati vengono conservati anche quando si chiude il DB.

È possibile aprire contemporaneamente un DB globale e un DB di istanza. La figura seguente mostra i diversi accessi ai blocchi dati.



4.2.4.6 Blocchi funzionali di sistema (SFB) e funzioni di sistema (SFC)

Blocchi predefiniti

Non è necessario programmare ogni funzione. Le CPU S7 forniscono blocchi predefiniti che possono essere richiamati dal programma utente.

Per maggiori informazioni consultare la guida di riferimento sui blocchi di sistema e funzioni di sistema (Rimandi alla descrizione dei linguaggi e Guida a blocchi e attributi di sistema).

Blocchi funzionali di sistema

Il blocco funzionale di sistema (SFB) è un blocco funzionale integrato nella CPU S7. Siccome fanno parte del sistema operativo, gli SFB non vengono caricati come parte del programma. Sia gli FB che gli SFB sono blocchi "con memoria". Anche per gli SFB è necessario creare blocchi dati di istanza e caricarli nella CPU come parte del programma.

Le CPU S7 forniscono SFB per:

- la comunicazione attraverso i collegamenti progettati
- per funzioni speciali integrate (p. es. SFB 29 "HS_COUNT" sulla CPU 312 IFM e sulla CPU 314 IFM)

Funzioni di sistema

La funzione di sistema è una funzione preprogrammata, ed è integrata nella CPU S7. Le SFC possono essere richiamate dal programma. Siccome fanno parte del sistema operativo, le SFC non vengono caricate come parti del programma. Sia le FC che le SFC sono blocchi "privi di memoria".

Le CPU S7 forniscono SFC per

- funzioni di copia e di blocco
- controllo del programma
- gestione dell'orologio e del contatore ore d'esercizio
- trasferimento di set di dati
- trasferimento degli eventi da una CPU a tutte le CPU collegate in modo di funzionamento multicomputing
- gestione degli allarmi dall'orologio e degli allarmi di ritardo
- gestione di eventi di errore di sincronismo, eventi di allarme ed eventi di errore di asincronismo
- informazioni sui dati di sistema statici e dinamici, p.es. diagnostica
- aggiornamento dell'immagine di processo ed elaborazione del campo di bit
- indirizzamento delle unità
- periferia decentrata
- comunicazione di dati globali
- comunicazione attraverso collegamenti non progettati
- creazione di messaggi riguardanti il blocco dati

Ulteriori informazioni

Per maggiori informazioni su SFB e SFC leggere il manuale di riferimento "Software di sistema per S7-300/400, Funzioni standard e di sistema". Per informazioni sulla disponibilità di SFB e SFC consultare il manuale "Sistema di automazione S7-300, Configurazione e dati della CPU" o il manuale di riferimento "Sistema di automazione S7-400, M7-400, Caratteristiche delle unità modulari.

4.2.5 Blocchi organizzativi per l'elaborazione del programma su interrupt

4.2.5.1 Blocchi organizzativi per l'elaborazione del programma su interrupt

Mediante la disponibilità degli OB di allarme, le CPU S7 forniscono la possibilità di

- elaborare segmenti di programma su interrupt periodico
- reagire in modo ottimale a segnali esterni del processo.

Il programma utente ciclico non deve chiedere continuamente se si sono verificati eventi di allarme. Al contrario, in caso di allarme, il sistema operativo fa sì che venga elaborata la parte del programma utente che si trova nell'OB di allarme, e stabilisce in che modo il controllore programmabile debba reagire a tale allarme.

Tipi di allarmi e applicazioni

La tabella seguente mostra in che modo possono essere utilizzati i diversi tipi di allarmi.

.

Tipo di allarme	OB di allarme	Esempi applicativi
Allarme dall'orologio	da OB 10 a OB 17	Calcolo della portata di un processo di miscelazione a fine turno
Allarme di ritardo	da OB 20 a OB 23	Comando del ventilatore che deve continuare a funzionare ancora per 20 secondi prima del disinserimento di un motore.
Allarme di schedulazione orologio	da OB 30 a OB 38	Rilevamento del livello di segnale per un impianto di regolazione
Interrupt di processo	da OB 40 a OB 47	Avviso che nel serbatoio è stato raggiunto il livello massimo.

4.2.5.2 Blocchi organizzativi di allarme dall'orologio (da OB 10 a OB 17)

Le CPU S7 mettono a disposizione OB di allarme dall'orologio, che possono essere elaborati ad una certa data o a determinati intervalli di tempo.

Gli allarmi dall'orologio possono essere avviati:

- una volta sola in un determinato momento (ora assoluta con data)
- periodicamente con l'indicazione del momento di inizio e della frequenza di ripetizione (per esempio, ogni minuto, ogni ora, ogni giorno).

Regole per gli allarmi dall'orologio

Gli allarmi dall'orologio possono essere gestiti solo se sono stati parametrizzati e se il blocco organizzativo corrispondente è contenuto nel programma utente. In caso contrario, viene immesso un messaggio di errore nel buffer di diagnostica, ed eseguita la gestione di errori di sincronismo (OB 80, vedere "Blocchi organizzativi per l'elaborazione degli errori (da OB 70 a OB 87 / da OB 121 a OB 122)").

Gli allarmi dall'orologio periodici devono corrispondere a una data reale. La ripetizione mensile dell'OB 10 con data di avvio 31.1 non è possibile. In questo caso, infatti, l'OB verrebbe avviato solo nei mesi che hanno effettivamente 31 giorni (dunque non in febbraio, aprile, giugno, etc).

Un allarme dall'orologio che viene attivato durante l'avvio (nuovo avviamento = avviamento a caldo, oppure riavviamento), viene elaborato solo dopo il completamento di tale operazione.

Gli OB di allarme dall'orologio che sono stati deselezionati tramite la parametrizzazione non possono essere avviati. La CPU rileva un errore di programmazione e va in STOP.

Dopo il nuovo avviamento (avviamento a caldo), gli allarmi dall'orologio impostati devono essere riattivati (ad esempio, con l'ausilio di SFC 30 ACT_TINT nel programma di avviamento).

Avviamento dell'allarme dall'orologio

Affinché la CPU possa avviare un allarme dall'orologio, quest'ultimo deve essere prima impostato e quindi attivato. Esistono tre possibilità di avviamento:

- avviamento automatico dell'allarme dall'orologio tramite parametrizzazione con STEP 7 (blocco parametri "Allarme dall'orologio")
- impostazione e attivazione dell'allarme dall'orologio tramite l'SFC 28 SET_TINT e l'SFC 30 ACT_TINT dal programma utente
- impostazione dell'allarme dall'orologio tramite parametrizzazione con STEP 7 e attivazione tramite l'SFC 30 ACT_TINT dal programma utente.

Interrogazione dell'allarme dall'orologio

Per chiedere al sistema se e quando sono stati impostati allarme dall'orologio, si può

- richiamare l'SFC 31 QRY_TINT o
- richiedere l'elenco parziale sullo stato degli allarmi della lista degli stati del sistema.

Disattivazione dell'allarme dall'orologio

Gli allarmi dall'orologio non ancora elaborati possono essere disattivati con l'SFC29 CAN_TINT. Gli allarmi dall'orologio disattivati possono essere nuovamente impostati con l'SFC 28 SET_TINT e attivati con l'SFC 30 ACT_TINT.

Priorità degli OB di allarme dall'orologio

Nella preimpostazione, tutti gli otto OB di allarme dall'orologio hanno la stessa classe di priorità (2), e vengono elaborati in base alla sequenza dei rispettivi eventi di avviamento. La classe di priorità può essere modificata mediante la parametrizzazione.

Modifica dell'ora impostata

Per modificare l'ora impostata, esistono le seguenti possibilità:

- un orologio master sincronizza l'ora per le unità master e slave
- nel programma utente l'ora viene impostata nuovamente tramite l'SFC 0 SET_CLK.

Comportamento dopo la modifica dell'ora

La seguente tabella mostra il comportamento degli allarmi dall'orologio dopo la modifica dell'ora.

Se...	**allora...**
mediante lo spostamento in avanti dell'ora vengono scavalcati uno o più allarmi dall'orologio,	viene avviato l'OB 80 e nelle informazioni di avviamento di quest'ultimo vengono registrati gli allarmi dall'orologio scavalcati.
nell'OB 80 gli allarmi dall'orologio scavalcati non sono stati disattivati dall'utente,	gli allarmi dall'orologio scavalcati non vengono recuperati.
nell'OB 80 gli allarmi dall'orologio scavalcati non sono stati disattivati dall'utente,	il primo allarmi dall'orologio scavalcato viene recuperato, mentre gli altri vengono ignorati.
mediante lo spostamento all'indietro dell'ora gli allarmi dall'orologio già elaborati risultano nuovamente in sospeso,	l'elaborazione di questi allarmi dall'orologio non viene ripetuta.

4.2.5.3 Blocchi organizzativi di allarme di ritardo (da OB 20 a OB 23)

Le CPU S7 mettono a disposizione OB di allarme di ritardo, mediante i quali è possibile programmare un'elaborazione ritardata di alcune parti del programma utente.

Regole per gli allarmi di ritardo

Gli allarmi di ritardo possono essere gestiti solo se nel programma della CPU si trova il blocco organizzativo corrispondente. In caso contrario, viene immesso un messaggio di errore nel buffer di diagnostica, ed eseguita la gestione di errori di asincronismo (OB 80, vedere "Blocchi organizzativi per l'elaborazione degli errori (da OB 70 a OB 87 / da OB 121 a OB 122)").

Gli OB di allarme di ritardo che sono stati deselezionati tramite la parametrizzazione non possono essere avviati. La CPU rileva un errore di programmazione e va in STOP.

Gli allarmi di ritardo vengono attivati allo scadere del tempo di ritardo indicato nell'SFC 32 SRT_DINT.

Avviamento di un allarme di ritardo

Per avviare un allarme di ritardo, è necessario impostare il tempo di ritardo nell'SFC 32, dopo il cui avvio deve essere richiamato l'OB corrispondente. La lunghezza massima del tempo di ritardo può essere verificata nel manuale "Sistema di automazione S7-300, Configurazione e dati della CPU" e nel manuale di riferimento "Sistemi di automazione S7-400, M7-400, Caratteristiche delle unità modulari".

Priorità degli OB di allarme di ritardo

Nella preimpostazione gli OB di allarme di ritardo hanno le classi di priorità da 3 a 6. Le classi di priorità possono essere modificate tramite parametrizzazione.

4.2.5.4 Blocchi organizzativi di schedulazione orologio (da OB 30 a OB 38)

Le CPU S7 mettono a disposizione OB di schedulazione orologio che interrompono l'elaborazione ciclica del programma in determinati intervalli di tempo.

La schedulazione orologio viene avviata a intervalli di tempo stabiliti. L'avviamento del clock avviene nel momento in cui lo stato di funzionamento passa da STOP a RUN.

Regole per schedulazione orologio

Durante l'impostazione dei clock, fare attenzione che tra gli eventi di avviamento delle singole schedulazioni orologio rimanga un tempo sufficiente per la gestione di queste ultime.

Gli OB di schedulazione orologio che sono stati deselezionati tramite la parametrizzazione non possono essere avviati. La CPU rileva un errore di programmazione e va in STOP.

Avviamento di una schedulazione orologio

Per avviare una schedulazione orologio, con STEP 7 è necessario preimpostare un clock nel blocco parametri "Schedulazione orologio". Il clock è sempre un multiplo intero del clock base di 1 millisecondo.

Clock = n x Clock base 1 ms

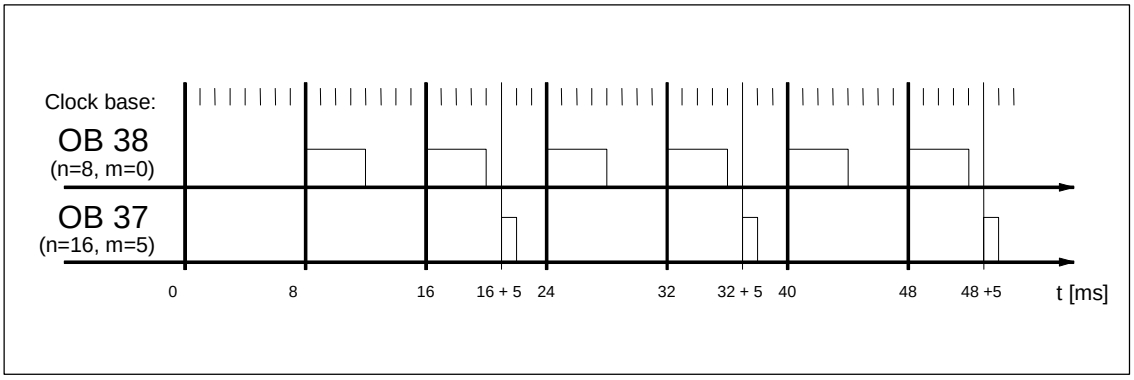
Nella preimpostazione, i nove OB di schedulazione orologio disponibili impostano i clock (vedere la seguente tabella). Il clock di default diventa attivo se l'OB di schedulazione orologio ad esso assegnato è stato caricato. Tuttavia, con la parametrizzazione, è possibile modificare i valori preimpostati. Per informazioni sul limite massimo si consiglia di consultare il manuale "Sistema di automazione S7-300, Configurazione e dati della CPU" e il manuale di riferimento "Sistemi di automazione S7-400, M7-400, Caratteristiche delle unità modulari.

Offset con schedulazioni orologio

Per evitare che le schedulazioni orologio di diversi OB di schedulazione orologio ricevano un comando di avviamento contemporaneamente, e che possa quindi verificarsi un errore di tempo (superamento del tempo di ciclo), esiste la possibilità di preimpostare un offset. Mediante questa operazione, l'elaborazione di una schedulazione orologio viene rimandata ad un momento successivo.

Offset = m x Clock base (con $0 \leq m < n$)

La seguente figura riporta l'elaborazione di un OB di schedulazione orologio con offset (OB 37), a differenza di una schedulazione orologio offset delle fasi (OB 38).



Priorità dell'OB di schedulazione orologio

La tabella seguente mostra i clock preimpostati e le classi di priorità degli OB di schedulazione orologio. Clock e classi di priorità possono essere modificati parametrizzandoli.

OB di schedulazione orologio-	Clock in ms	Classe di priorità
OB 30	5000	7
OB 31	2000	8
OB 32	1000	9
OB 33	500	10
OB 34	200	11
OB 35	100	12
OB 36	50	13
OB 37	20	14
OB 38	10	15

4.2.5.5 Blocchi organizzativi di interrupt di processo (da OB 40 a OB 47)

Le CPU S7 mettono a disposizione OB di interrupt di processo, che reagiscono a segnali dalle unità (p. es. unità di ingresso/uscita SM, processori di comunicazione CP, unità funzionali FM). Per le unità digitali e analogiche parametrizzabili, con STEP 7 è possibile impostare il segnale che deve avviare l'OB. Per far questo, con CP e FM utilizzare le relative maschere di parametrizzazione.

Gli interrupt di processo vengono avviati quando un'unità di ingresso/uscita, che supporta interrupt di processo, con l'abilitazione dell'interrupt di processo parametrizzabile, trasmette alla CPU un segnale di processo ricevuto, oppure quando un'unità funzionale segnala un interrupt alla CPU.

Regole per interrupt di processo

Gli interrupt di processo possono essere gestiti solo se nel programma della CPU si trova il blocco organizzativo corrispondente. In caso contrario, viene registrato un messaggio di errore nel buffer di diagnostica, ed eseguita la gestione di eventi di errore di asincronismo (vedere "Blocchi organizzativi per l'elaborazione degli errori (da OB 70 a OB 87 / da OB 121 a OB 122)").

Gli OB di interrupt di processo che sono stati deselezionati mediante la parametrizzazione non possono essere avviati. La CPU rileva un errore di programmazione e va in STOP.

Parametrizzazione di unità di ingresso/uscita che supportano interrupt di processo

Tutti i canali di un'unità di ingresso/uscita che supportano interrupt di processo possono avviare questi ultimi. A questo scopo, nei set di parametri di tali unità, utilizzando STEP7 è necessario stabilire quanto segue:

- in che modo un interrupt di processo deve essere avviato
- quale OB di interrupt di processo deve essere elaborato (la preimpostazione prevede l'OB 40 per l'elaborazione di tutti gli interrupt di processo).

Con STEP 7 si attiva la generazione degli interrupt di processo delle unità funzionali. Assegnare parametri aggiuntivi nelle maschere di parametrizzazione di tali unità funzionali.

Priorità degli OB di interrupt di processo

Nella preimpostazione gli OB di interrupt di processo hanno le classi di priorità da 16 a 23. Le classi di priorità possono essere modificate mediante parametrizzazione.

4.2.5.6 Blocchi organizzativi per l'avviamento (OB 100 / OB 101 / OB 102)

Tipi di avviamento

Si possono distinguere i seguenti tipi di avviamento

- Riavviamento (escluso S7-300 e S7-400H)
- Nuovo avviamento (avviamento a caldo)
- Avviamento a freddo

Nella seguente tabella si può consultare quale OB richiama il sistema operativo per ogni tipo di avviamento.

Tipo di avviamento	OB rispettivo
Riavviamento	OB 101
Nuovo avviamento (Avviamento a caldo)	OB 100
Avviamento a freddo	OB 102

Eventi di avvio per gli OB di avviamento

La CPU esegue un avviamento

- dopo RETE ON
- se si commuta l'interruttore di stati di funzionamento da STOP a "RUN"/"RUN-P"
- a seguito di richiesta da parte di una funzione di comunicazione
- dopo la sincronizzazione nel funzionamento multicomputing
- in un sistema H dopo l'accoppiamento (solo per CPU di riserva)

A seconda dell'evento di avvio, della CPU con cui su opera, e dei parametri impostati ion essa, viene richiamato il rispettivo OB di avviamento (OB 100, OB 101, OB 102).

Programma di avviamento

Per definire le condizioni marginali per il comportamento all'avviamento della CPU (valori di inizializzazione per RUN, valori di avviamento per le unità di periferia), occorre memorizzare il programma di avviamento nel blocco organizzativo OB 100 per il nuovo avviamento (avviamento a caldo), OB 101 per il riavviamento, OB 102 per l'avviamento a freddo.

Il programma di avviamento può avere una lunghezza qualsiasi; per la sua esecuzione non esiste alcun limite di tempo e il controllo del tempo di ciclo non è attivo. Nel programma di avviamento non è possibile l'elaborazione su interrupt. All'avvio lo stato di segnale di tutte le uscite digitali è 0.

Tipo di avviamento dopo l'avviamento manuale

Nelle CPU S7-300-è possibile soltanto il nuovo avviamento manuale (avviamento a caldo) o l'avviamento a freddo (solo CPU 318-2).

Nelle CPU S7-400 è possibile effettuare un riavviamento manuale con il selettore dei modi operativi e il selettore di avviamento (CRST/WRST), se è stato così stabilito mediante la parametrizzazione con STEP 7. Il nuovo avviamento manuale (avviamento a caldo) è possibile anche in assenza di parametrizzazione.

Tipo di avviamento dopo avviamento automatico

Nelle CPU S7-300, dopo RETE ON è possibile solo un nuovo avviamento (avviamento a caldo).

Nelle CPU S7-400 l'utente può decidere se un avviamento automatico dopo RETE ON debba essere un nuovo avviamento (avviamento a caldo) oppure un riavviamento.

Cancellazione dell'immagine di processo

Nel riavviamento di una CPU S7-400, l'immagine di processo delle uscite viene cancellata per default dopo l'elaborazione del ciclo restante. Se il programma utente, dopo il riavviamento, deve continuare l'elaborazione con i valori che erano attuali prima di tale operazione, è possibile deselezionare la cancellazione dell'immagine di processo.

Controllo della configurazione prefissata-attuale delle unità

Mediante la parametrizzazione, è possibile stabilire che prima dell'avviamento venga verificato che tutte le unità elencate nella tabella di configurazione siano effettivamente collegate e siano del tipo corretto.

Se si attiva il controllo delle unità, l'avviamento non ha luogo qualora venga riscontrata una differenza tra la configurazione prefissata e quella attuale.

Tempi di controllo

Per garantire un avviamento senza errori del controllore programmabile si possono parametrizzare i seguenti tempi di controllo:

- tempo massimo consentito per il trasferimento dei parametri alle unità
- tempo massimo consentito per la segnalazione di pronto delle unità dopo RETE ON
- nelle CPU S7-400, il tempo massimo di interruzione nel quale è ancora consentito un riavviamento.

Scaduti i tempi di controllo, la CPU passa allo stato STOP, oppure si può effettuare solo un nuovo avviamento (avviamento a caldo).

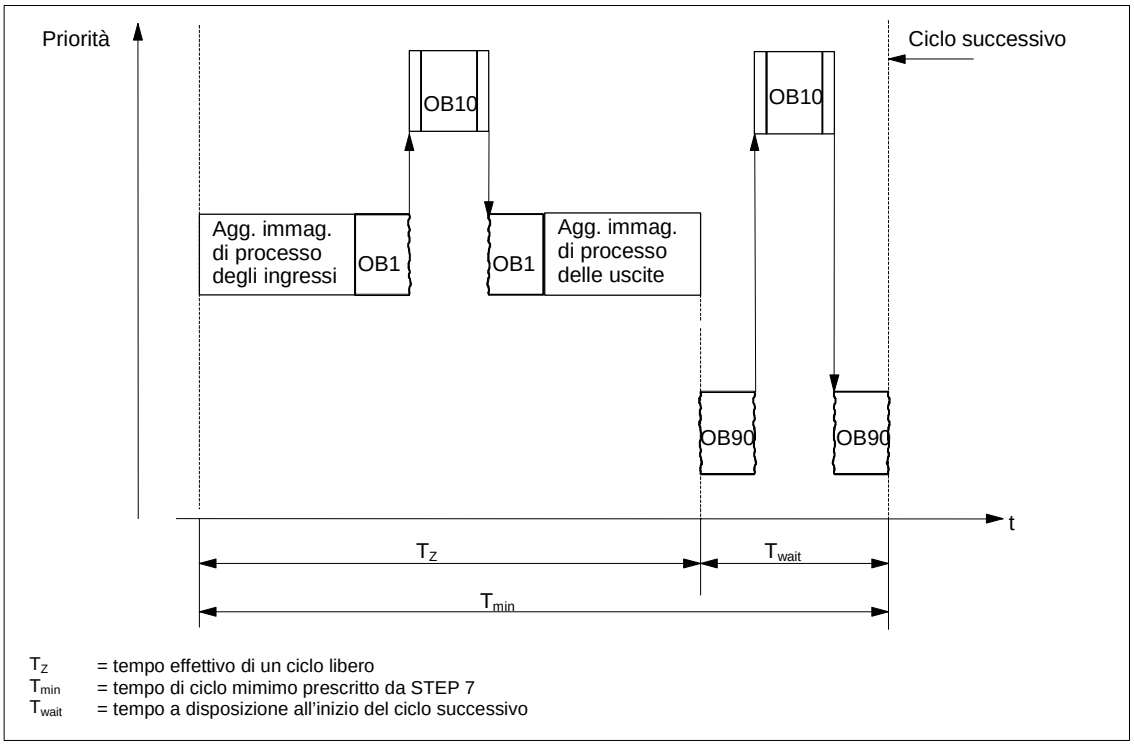
4.2.5.7 Blocco organizzativo di ciclo di priorità bassa (OB 90)

Se con STEP 7 l'utente ha impostato un tempo di ciclo minimo che risulta essere superiore al tempo di ciclo reale, alla fine del programma ciclico la CPU dispone ancora di tempo di elaborazione residuo, che viene utilizzato per elaborare l'OB di priorità bassa. Se l'OB 90 non è presente nella CPU, la CPU attende che sia trascorso il tempo di ciclo minimo preimpostato. Si può pertanto far svolgere mediante l'OB 90 processi senza criticità temporale, evitando tempi di attesa.

Priorità dell'OB di priorità bassa

L'OB di priorità bassa detiene la classe di priorità 29, che corrisponde alla priorità 0.29. Si tratta dunque dell'OB con la priorità più bassa. La classe di priorità non può essere modificata mediante parametrizzazione.

La figura seguente riporta un esempio di esecuzione di ciclo di priorità bassa, di ciclo libero e di OB 10 (con le CPU a partire da 10/98).



T_Z = tempo effettivo di un ciclo libero
T_{min} = tempo di ciclo mimimo prescritto da STEP 7
T_{wait} = tempo a disposizione all'inizio del ciclo successivo

Programmazione dell'OB 90

Il tempo di esecuzione dell'OB 90 non viene controllato dal sistema operativo della CPU; l'utente potrà pertanto programmare nell'OB 90 dei loop di qualsiasi lunghezza.
Osservare la coerenza dei dati utilizzati nel programma di priorità bassa tenendo conto in sede di programmazione di quanto segue:

- gli eventi di reset dell'OB 90 (vedere manuale di riferimento "Software di sistema per S7-300/400, Funzioni standard e di sistema"),
- l'aggiornamento asincrono dell'immagine di processo relativo all'OB 90.

4.2.5.8 Blocchi organizzativi per l'elaborazione degli errori (da OB 70 a OB 87 / da OB 121 a OB 122)

Tipi di errori

Gli errori che le CPU S7 riconoscono, e ai quali è possibile reagire tramite i blocchi organizzativi, sono suddivisi in due categorie:

- errori di sincronismo: questi errori possono essere assegnati a una determinata parte del programma utente. L'errore viene prodotto durante l'elaborazione di una determinata operazione. Se non è stato caricato l'OB di errore di sincronismo corrispondente, la CPU va in STOP quando si manifesta l'errore.
- errori di asincronismo: questi errori non possono essere correlati direttamente al programma utente elaborato. Si tratta di errori nella classe di priorità o di errori del controllore programmabile (per esempio, difetti delle unità). Se non è stato caricato l'OB di errore di asincronismo corrispondente, la CPU va in STOP quando si manifesta l'errore (ad eccezione dell'OB 70, OB 72, OB 81).

La tabella seguente riporta i tipi di errore che possono manifestarsi, suddivisi secondo la categoria degli OB di errore.

Errori di asincronismo e di ridondanza	**Errori di sincronismo**
OB 70 Errore di ridondanza della periferia (solo CPU H)	OB 121 Errore di programmazione (p.es. DB non caricato)
OB 72 Errore di ridondanza CPU (solo nelle CPU H, p.es. guasto a una CPU)	OB 122 Errore di accesso alla periferia (p.es. accesso a un'unità di ingresso/uscita inesistente)
OB 80 Errore temporale (p.es. superamento del tempo di ciclo)	
OB 81 Errore alimentatore (p. es. errore batteria)	
OB 82 Allarme di diagnostica (p.es. cortocircuito nell'unità di ingresso)	
OB 83 Allarme di estrazione/inserimento (p.es. estrazione di un'unità di ingresso)	
OB 84 Errore hardware CPU (errore dell'interfaccia della rete MPI)	
OB 85 Errore di esecuzione programma (p.es. OB non caricato)	
OB 86 Guasto al telaio di montaggio	
OB 87 Errore di comunicazione (p. es. ID telegramma errato nella comunicazione GD)	

Uso degli OB per errori di sincronismo

Gli errori di sincronismo vengono prodotti durante l'elaborazione di una determinata operazione. Quando si presentano questi errori, il sistema operativo crea una registrazione nell'area U-Stack e avvia l'OB per gli errori di sincronismo.

Gli OB di errore, richiamati dagli errori di sincronismo, vengono elaborati come parti del programma con la stessa classe di priorità del blocco che viene elaborato al rilevamento dell'errore. L'OB 121 e l'OB 122 possono anche accedere ai valori che, al momento dell'interruzione, erano memorizzati negli accumulatori e negli altri registri. I valori possono essere utilizzati per reagire alla condizione di errore e ritornare quindi all'esecuzione del programma (p. es. nel caso di errori di accesso a un'unità analogica nell'OB 122 con la SFC 44 RPL_VAL indicare un valore sostitutivo). In questo modo, i dati locali caricano non solo l'OB di errore, ma anche l'L-stack di questa classe di priorità.

Nelle CPU S7-400, da un OB di errore di sincronismo può essere avviato un altro OB di errore di sincronismo. Nelle CPU S7-300 questo non è possibile.

Uso degli OB per errori di asincronismo

Quando il sistema operativo della CPU rileva un errore di asincronismo, avvia l'OB di errore corrispondente (da OB 70 a OB 73 e da OB 80 a OB 87). Gli OB di errore di asincronismo hanno preimpostata la priorità più alta: non possono essere interrotti da altri OB se tutti gli OB di errore di asincronismo hanno la stessa priorità. Se compaiono contemporaneamente diversi OB di errori di asincronismo con la stessa priorità, vengono elaborati nella sequenza in cui si presentano.

Mascherare gli eventi di avviamento

Con le funzioni di sistema (SFC) è possibile mascherare, ovvero rinviare o inibire, gli eventi di avviamento di alcuni OB di errore. Per maggiori informazioni sull'argomento e sui singoli blocchi organizzativi consultare il manuale di riferimento "Software di sistema per S7-300/400, Funzioni standard e di sistema".

Tipo di OB di errore	SFC	Funzione dell'SFC
OB di errore di sincronismo-	SFC 36 MSK_FLT	Maschera gli eventi di errore di sincronismo. Gli eventi di errore mascherati non avviano alcun OB di errore, e non comportano nessuna reazione sostitutiva programmata.
	SFC 37 DMSK_FLT	Demaschera eventi di errore di sincronismo
OB di errore di asincronismo	SFC 39 DIS_IRT	Inibisce globalmente gli eventi di allarme e di errore di asincronismo . Gli eventi di errore inibiti non avviano OB di errore in alcun ciclo successivo della CPU, e non portano alla reazione sostitutiva programmata.
	SFC 40 EN_IRT	Abilita gli eventi di allarme e di errore di asincronismo
	SFC 41 DIS_AIRT	Ritarda gli eventi di allarme e di errore di asincronismo di alta priorità fino alla fine dell'OB
	SFC 42 EN_AIRT	Abilita gli eventi di allarme e di errore di asincronismo di alta priorità

Avvertenza

Per ignorare gli allarmi è più efficiente bloccarli nell'avviamento tramite SFC, invece di caricare un OB vuoto (con contenuto BE).