



TAC VISTA® IV M.S.A. IPCL EDITOR è il linguaggio per il controllo di processo sviluppato da TAC per la programmazione delle unità RPU (Remote Processing Unit). Consente di gestire, con ottime capacità e grande flessibilità, funzioni di monitoraggio e controllo dell'impianto, così come calcoli e regolazioni. TAC VISTA® IV M.S.A. IPCL EDITOR è un linguaggio di alto livello utilizzato per definire le relazioni all'interno di una RPU tra ingressi e uscite e tra la parte logica di controllo, controllori e timer. Si tratta di un linguaggio semplice da programmare, leggere, stampare e testare. Gli errori sono facili da trovare e correggere.

Linguaggio di programmazione

Il linguaggio è composto da una serie di istruzioni e regole di impiego. La programmazione è in inglese. È possibile archiviare come codice di controllo riservato di una certa RPU o decidere di avere un codice standard (di *default*) per un tipo specifico di RPU.

Editor

Il programma viene creato come file di testo con l'editor IPCL. TAC VISTA® IV M.S.A. IPCL EDITOR funziona come un processore di testo con funzioni aggiuntive per la conversione ed il trasferimento del codice programma. Utilizzando la tecnica a finestre di dialogo è possibile avere diversi file e programmi aperti contemporaneamente. Testi ed elementi di testo possono essere copiati da altri file o programmi utilizzando *editing* convenzionali. Questo consente di riutilizzare il codice programma di altri impianti e di utilizzare anche un editor di testo qualsiasi, quale ad esempio Microsoft® Word.

L'editor IPCL offre le seguenti funzioni:

- La funzione di *editing* del file di testo è effettuata utilizzando, ad esempio, le funzioni trova/sostituisci e taglia/incolla. Vengono utilizzati anche i menu a finestre presenti in Microsoft® Windows.
- Conversione di file di testo in codice programma eseguibile per singola unità RPU.
- Trasferimento programma; ovvero a conversione eseguita, trasferimento del codice programma alla RPU configurata come periferica di ricevimento.

dati tecnici

Sistema operativo	Microsoft® Windows NT/2000	Versioni Microsoft® Windows	
		Windows NT	versione 4.0 con Service Pack 6
		Windows 2000	con Service Pack 2
		Windows XP	Professional

identificativo prodotto

Prodotto	Descrizione	Codice
TAC VISTA® IV M.S.A. IPCL EDITOR	Linguaggio di programmazione per sistemi TAC Zone II e System 7	0101-03-05

funzioni

Riportiamo alcuni esempi di istruzioni di programma che variano a seconda del tipo di RPU.

Istruzioni logiche

AND	AND logico
OR	OR logico
NOT	NOT logico
NE	NOT EQUAL (XOR) logico
SET	Assegnazione logica
NET	Assegnazione invertita
TVT	Timer variabile tempo
TVA	Timer alternativo
S1	Set a 1 incondiz.
S0	Set a 0 incondiz.

Variabili logiche

IN	Ingressi digitali
LL	Allarme basso
HL	Allarme alto
OUT	Uscite digitali
FI	Variabili fitt.
ILV	Variabili logiche interne
TVS	Stato variabile tempo
TVAS	Stato controllo altern.

Funzioni logiche speciali

TIME	Assegnazione tempo
PWRDWN	Set impostato al ripristino della corrente in seguito ad un'interruzione
OFFLINE	Controllo comunicazione
CLARM	Set impostato se è presente uno o più allarmi nell'unità centrale
LLARM	Set impostato se sono presenti degli allarmi nella lista di allarmi

Forzature

F1	Forzatura incondiz. ON
F0	Forzatura incondiz. OFF
AU	Ritorno allo stato normale, non-forzato, AUTO
C1	Forzatura condiz. ON se LAC*=1, altrimenti AUTO
C0	Forzatura condiz. OFF se LAC*=1, altrimenti AUTO

Funzioni aritmetiche

RT	Tempo in corso
CNT	Contatore
MV	Tabella valori misurati
SV	Set tabella valore
RV	Variabile controllore
IFV	Tabella variabili interne a virgola mobile
AOUT	Tabella uscite analogiche
SQRT	Radice quadrata; salvata in FLAC**
+	Addizione
-	Sottrazione
/	Divisione
*	Moltiplicazione
>>	Max
<<	Min
>	Comparazione aritmetica con risultato logico
>=	Comparazione aritmetica con risultato logico
<	Comparazione aritmetica con risultato logico
<=	Comparazione aritmetica con risultato logico
=	Uguale
CONST	Costante: numero intero o a virgola mobile

Condizioni e salti

J xxx	Salto incond. a xxx
CJ xxx	Salto incond. a xxx LAC*=1, altrimenti passaggio all'istruzione successiva
NJ xxx	Salto incond. a xxx se LAC*=0, altrimenti passaggio all'istruzione successiva
JSR xxx	Salta a subroutine con indirizzo xxx
RTN	Salto di ritorno da subroutine
IF	Avvia istruzione
THEN	Continua da istruzioni da eseguire se la condizione è vera (true)
ELSE	Continua da istruzioni da eseguire se la condizione è falsa (false)
ENDIF	Istruzione di fine

Altre istruzioni

AL	Condizione allarme
PU	Condizione impulso
R1 (n)	Attiva controllore (n)
R0 (n)	Disattiva controllore (n)
MI (n)	Sposta al min il segnale di uscita da controllore REG (n)
MA (n)	Sposta al max il segnale di uscita da controllore REG (n)
DON	Ritardo ON
DOFF	Ritardo OFF
DOFFON (n)	Ritardo ON/OFF
D v	Deviazione; aggiorna su deviazione di almeno v unità
T n	Aggiorna ogni n secondi
DL (n) y	Ritardo salto a y, tempo attesa per tabella n
RPU (n)	numero unità RPU n; n=0: globale per tutte le RPU
ZC (n)	Numero n Unità controllore di zona
ZG (n)	Numero n Gruppo controllore di zona
STIM hh:mm	Imposta orologio
YEAR	Anno rilevato da orologio RPU
MONTH	Mese rilevato da orologio RPU
DATE	Giorno rilevato da orologio RPU
HOUR	Ora rilevata da orologio RPU
MIN	Minuto rilevato da orologio RPU
SEC	Secondo rilevato da orologio RPU
WDAY	Giorno della settimana; 1=Lunedì... 7=Domenica
END	Fine programma
* LAC =	Accumulatore Logico, variabile ausiliaria che serve come locazione di salvataggio per gli ultimi stati logici letti o calcolati.
** FLAC =	Accumulatore a virgola mobile, variabile ausiliaria che serve come locazione di salvataggio per l'ultima variabile letta o l'ultimo numero a virgola mobile calcolato.