

MANUALE DI PROGRAMMAZIONE S2000

Questo manuale spiega come usare il software di programmazione S2000C, per ottenere dallo strumento le funzioni di calcolo disponibili.

Se non riuscite a ottenere un particolare risultato, non esitate per contattarci e chiederci chiarimenti in merito.

1. INTRODUZIONE

L' S2000 si compone di un microprocessore dotato di porta seriale, di alcuni ingressi e uscite analogiche e digitali . (vedere più avanti per le caratteristiche generali).

E' dotato inoltre di un software residente (cioè non modificabile se non sostituendo la EPROM) che può eseguire un certo numero di operazioni memorizzate in una EEPROM (memoria riscrivibile ma non cancellabile).

Il tipo di operazioni eseguibili e' fisso (il capitolo 3 le descrive), mentre il loro numero e' legato all' occupazione di EEPROM. In genere una istruzione occupa un byte mentre un numero ne occupa 4; il massimo numero di bytes e' 240.

Le operazioni da eseguire vengono per prima cosa scritte con un PC su un file, mediante l'editor di testo "EDIT" standard dell' MS-DOS, dove si possono modificare , memorizzare e richiamare come un normale file di testo.

Vi rimandiamo al manuale DOS per l'uso dell' editor citato.

Si può usare qualunque altro editor di testo a condizione che non introduca nessun tipo di caratteri di controllo nel file. Non si possono usare quindi editor come WORD o WORD STAR in modalità documento.

In seguito si esegue la compilazione e il trasferimento delle istruzioni usando il compilatore S2000C che viene fornito a richiesta completo di manuale aggiornato e di cavetto di collegamento.

Il compilatore controlla che tutte le istruzioni siano correttamente scritte , le codifica in codice comprensibile all' S2000 e le trasferisce al medesimo via porta seriale, verificando che siano state ricevute senza errori.

Il compilatore, elenca tutte le istruzioni compilate; in caso di errore di battitura delle istruzioni, il compilatore scriverà sul video l' errore con indicazione della istruzione errata.

Il compilatore è in grado di controllare se le istruzioni sono scritte in modo ortograficamente corretto ma non può verificare se la loro sequenza sia logica.

Durante il trasferimento all' S2000 possono verificarsi degli errori che vengono anche questi visualizzati su schermo. In questo caso bisognerà ripetere tutta la procedura.

Dopo il trasferimento del programma, l' S2000 inizia l' esecuzione dello stesso e se incontra problemi (di esecuzione) li invia al PC perché possano essere visualizzati.

Perciò mentre si effettua il collaudo del programma inserito nell' S2000, provando tutte le possibili combinazioni di valori di ingresso, bisogna lasciare il PC collegato all' S2000 e il programma S2000C in funzione.

Anche in questo caso, l' S2000 non e' in grado di verificare la logicità delle istruzioni; può solamente indicare che ha incontrato una divisione per zero o altro errore simile.

2. PROGRAMMAZIONE

La programmazione (la sequenza di istruzioni) è simile a quella della calcolatrici HP, cioè lavora con una pila di dati che vengono manipolati dalle istruzioni impartite.

La pila di dati è un casellario dove sono inseriti o tolti i dati a seconda della istruzione da eseguire; si immagina un tavolo dove si appoggiano -in pila- dei fogli dove sono scritti dei numeri.

Il foglio (o i fogli) appoggiato per ultimo e' quello che viene usato per primo nella operazione da eseguire. Una volta eseguita l' operazione, sulla pila troviamo il risultato (o i risultati) di quest' ultima.

Vediamo un esempio: vogliamo eseguire un somma di due numeri.

La sequenza di istruzioni e' la seguente :

1.2647	(riga 1)
3.6744	(riga 2)
+	(riga 3)
STOP	(riga 4)

N.B. Le istruzioni possono anche essere scritte di seguito, introducendo uno spazio tra due istruzioni.

In dettaglio vediamo i vari passaggi e la posizione dei dati sulla pila; sulla sinistra sono incasellate le istruzioni via via eseguite e sulla destra l' effetto delle operazioni sulla pila. Questi schemi sono qui stampati per pura didattica e non appariranno a video.

Istruzione	contenuto pila
nessuna	<u>nessuno</u>
1.2647	<u>1.2647</u>
3.6744	3.6744 <u>1.2647</u>
+	4.9391

Notare:

- Anche un numero è un'istruzione.
- La base della pila è sottolineata.
- All' inizio la pila è vuota perciò ogni operazione di calcolo è inesequibile e darà luogo a un errore di esecuzione (rilevato dall' S2000, non dal compilatore).
- L' istruzione STOP non è eseguita dall' S2000 ma dal compilatore. Vedere l' elenco istruzioni per ulteriori chiarimenti.

E' evidente che l' uso della pila è molto comodo: può infatti sostituire le parentesi che si adoperano per la normale notazione scientifica delle formule. Vediamone subito un esempio: vogliamo fare il seguente calcolo : $1.25 * (100 - 20)$.

Come regola generale, dato che il nostro sistema non prevede le parentesi, dovremo essere noi a far eseguire per prime le operazioni all' interno delle suddette.

Istruzione	contenuto pila
100	100
20	20 <u>100</u>
-	<u>80</u>
1.25	1.25 <u>80</u>
*	100

Notare:

- Questo sistema di calcolo riduce la quantità di memoria spesa per memorizzare le istruzioni.

La pila può quindi contenere i risultati parziali dei calcoli, perciò un calcolo come il seguente risulta facile da eseguire : $2 * 7 - 5 / 2 - 15.52$

Istruzione	contenuto pila
2	<u>2</u>
7	7 <u>2</u>
*	<u>14</u>
5	5 <u>14</u>
2	2 5 <u>14</u>
/	2.5 <u>14</u>
-	<u>11.5</u>
15.52	15.52 <u>11.5</u>
-	<u>-4.02</u>

A questo punto bisogna poter inserire nel calcolo anche le variabili di ingresso e uscita e non solamente costanti, altrimenti lo strumento diventa inutile.

Le variabili in realtà sono nient'altro che operazioni come le classiche +, -, , solamente esse hanno altri simboli o nomi e usano, modificano o depositano valori sulla pila.

Ad esempio vediamo l'uso di un ingresso analogico e di una uscita analogica.

istruzione	contenuto pila
AI1	<u>0.953</u> (misura)
2	2 <u>0.953</u>
/	<u>0.4765</u>
AO1	n.nnnn

Come si vede l'istruzione AI1 lascia sulla pila il valore letto al relativo ingresso analogico, questo valore viene diviso per 2 e poi viene usata l'istruzione AO1 che manda il valore calcolato alla uscita analogica 1.

Questa istruzione toglie dalla pila il numero che usa, di conseguenza il numero sulla cima della pila è quello preesistente all'inizio dei nostri calcoli, perciò sconosciuto.

La pila può contenere al massimo 10 valori numerici, se si inseriscono più di dieci numeri, quelli che stanno alla base vengono perduti, mentre se una istruzione non ha sulla pila tutti i numeri di cui ha bisogno, prenderà numeri di casuali.

Esistono anche istruzioni che usano o determinano risultati "digitali".

Il valore digitale è un numero normale che viene interpretato FALSO quando vale 0 e VERO quando è diverso da zero. Può perciò essere usato anche nelle istruzioni che operano su numeri normali, e viceversa: le operazioni che si aspettano valori digitali trattano semplicemente il numero come digitale ovvero uguale o diverso da 0.

Le operazioni che restituiscono un valore digitale, danno 0 oppure 1 per comodità.

Vediamone una, per esempio la istruzione COMP, la quale dati due valori in ingresso restituisce 1 se il primo valore inserito è maggiore del secondo.

istruzione	contenuto pila
AI1	<u>0.725</u>
5e-1	0.500 <u>0.725</u>
COMP	<u>1.000</u>
AI2	<u>0.098</u>
0.100	0.100 <u>0.098</u>
COMP	<u>0.000</u>

Notare:

- I numeri possono essere introdotti anche in notazione scientifica nei formati :

+/-n.nnnnnne+ /- nn oppure +/-n.nnnnnn.

Esempi di numeri validi : 1.1278 , 0.00345, -4.28e-3, 10000, 1e4, 7848.33.

Vediamo il seguente esempio che permette di azzerare una misura (AI1) se essa è minore di un certo valore, in questo caso 0.200.

istruzioni	contenuto pila
AI1	<u>0.123</u>
DUP	0.123 <u>0.123</u>
0.200	0.200 0.123 <u>0.123</u>
COMP	0.000 <u>0.123</u>
*	<u>0.000</u>

In questo esempio è stata usata una nuova istruzione : DUP .
DUP esegue la duplicazione del valore in cima alla pila.
L' ultima operazione azzerava il valore di A11 oppure lo lascia intatto.

Con una opportuna combinazione delle operazioni base di seguito elencate è possibile eseguire una serie di calcoli e di scelte sui valori di ingresso e uscita analogici o digitali. Il programma viene eseguito dall' inizio alla fine senza interruzione eccetto quando esistono errori di esecuzione.

Alla fine della lista di istruzioni è necessario inserire l' istruzione STOP; si tratta di un comando per il compilatore, il quale a questo punto compila la lista e la trasmette all' S2000 se è stata data l'opzione di trasmissione.

Il tempo di ciclo totale dipende molto dal numero e dal tipo di istruzioni eseguite.

3. ELENCO ISTRUZIONI

Questo elenco descrive ciò che le operazioni fanno; al fianco del nome vengono indicati i valori che la operazione deve trovare sulla pila e quelli che lascia sulla pila stessa: ($n \leftarrow n_1, n_2, n_3, \text{ecc.}$); alla sinistra della freccia sono i numeri che rimangono sulla pila al termine dell' operazione, alla destra sono i numeri che sono necessari all' operazione per lavorare; n_1 è il numero introdotto per primo e così via perciò n_1 è alla base della pila.

I valori considerati digitali sono indicati con 'd'.

Sono indicati anche i valori non ammessi per certe operazioni: questi valori determinano un errore di esecuzione che interrompe lo svolgimento delle operazioni e comunica l'errore incontrato attraverso il PC. L' S2000 riprende poi dall' inizio l'esecuzione del programma.

+ (Somma) ($n \leftarrow n_1 + n_2$)
somma due numeri, sostituisce i due valori con il risultato.

- (Sottrazione) ($n \leftarrow n_1 - n_2$)
sottrae il numero in cima alla pila dal numero che sta sotto.

*** (Moltiplicazione)** ($n \leftarrow n_1 * n_2$)
moltiplica due numeri.

/ (Divisione) ($n < n_1 / n_2$)
divide due numeri. ERRORE per $n_2 = 0$.

NEG ($n \leftarrow -n$)
nega il numero sulla pila.

DUP ($n, n \leftarrow n$)
duplica il numero sulla pila.

DEL (scarta $\leftarrow n$)
elimina il numero sulla pila (non è sinonimo di azzeramento).

SWAP ($n \leftarrow n$)
scambia i due numeri in testa alla pila, il numero in cima diventa il secondo e viceversa.

- RADQ** (n \leftarrow n)
 esegue la radice quadrata del numero. ERRORE per n < 0.
- LOG** (n \leftarrow n)
 esegue il logaritmo base 10 del numero. ERRORE per n <= 0.
- LN** (n \leftarrow n)
 esegue il logaritmo base 'e' del numero. ERRORE per n <= 0.
- EXP10** (n \leftarrow n)
 esegue l' elevazione a potenza n di 10.
- EXPE** (n \leftarrow n)
 esegue l' elevazione a potenza n di 'e' .
- EXP** (n \leftarrow n1, n2)
 esegue l' elevazione a potenza n2 di n1. ERRORE se n1 < 0.
- COMP** (d \leftarrow n1, n2)
 confronta due numeri sulla pila, se n1 > n2: d = 1.
- SWITCH** (n \leftarrow n1, n2, d)
 seleziona due numeri in relazione allo stato di un valore digitale: se d = 0 allora n = n1; se d = 1 allora n = n2;
- NOT** (d \leftarrow d)
 nega d: se d = 0 allora d = 1; se d = 1 allora d = 0.
- AI1, AI2, AI3, AI4** (n \leftarrow analog input specificato)
 deposita in cima alla pila il valore letto dal relativo ingresso analogico con valori compresi tra -1.000 e +1.000 in corrispondenza di un segnale di ingresso -20..+20mA. La misura può arrivare fino a ± 1.0240 con una risoluzione di 2.5. Dovrà essere seguita da una istruzione ZS per poter leggere un segnale 4..20mA.
- AO1, AO2** (analog output specificato < n)
 copia il valore in cima alla pila sulla uscita analogica. Il numero deve essere compreso tra 0 e +1.000, che darà luogo a una corrente in uscita compresa tra 0 e 20mA. La conversione viene eseguita per fino a un massimo di 1.0240. Dovrà essere preceduta da una istruzione ZA per ottenere una uscita 4..20mA.
- ZA** (n \leftarrow n)
 sposta lo zero di un valore compreso tra 0 e 1 del 20%.
 E' necessario per fare uscire 4..20 mA dalle uscite analogiche quando il valore n è compreso tra 0.000 e 1.000.
- ZS** (n \leftarrow n)
 sottrae lo zero del 20% a un valore numerico compreso tra 0 e 1.
 Risulta indispensabile per leggere gli ingressi analogici in 4..20 mA.

- DI1, DI2** (d \leftarrow ingresso digitale specificato)
deposita sulla pila il valore digitale letto all' ingresso specificato: 0 se l' ingresso è aperto, 1 se l' ingresso è chiuso.
- DO1, DO2** (uscita digitale specificata \leftarrow d)
copia il valore in cima alla pila sull' uscita digitale specificata. Il valore 0 apre il circuito mentre il valore 1 lo chiude.
- STO0, STO1 ecc fino a STO4** (registro specificato, n \leftarrow n)
memorizza nel registro specificato il valore in cima alla pila senza rimuoverlo da quest' ultima. I registri da 0 a 4 sono disponibili per memorizzare variabili per qualsiasi uso.
- RCL0, RCL1 ecc fino a RCL4** (n \leftarrow registro specificato)
recupera dal registro specificato il valore che contiene e lo deposita in cima alla pila.
E' l' istruzione complementare a quella di STO0 ecc.
- %...%** non si tratta di una istruzione ma dei delimitatori di commento. Tra i caratteri % può essere introdotta una frase di lunghezza a piacere, che non verrà presa in considerazione dal compilatore.

4. ISTRUZIONI SPECIALI

Queste istruzioni svolgono compiti particolari che descriveremo tra breve.

INTEGRATORE CON INGRESSO ANALOGICO.

Le seguenti istruzioni sono relative alla messa in funzione di due integratori ad ingresso analogico, che faranno uscire degli impulsi di durata fissa a 100ms sulle uscite digitali.

La massima frequenza in uscita è di 5 Hz.

L' integratore 1 e 2 faranno uscire l' impulso rispettivamente sulla uscita 1 e 2.

Per attivare gli integratori basterà inserire nel registro valore di impulso (PVAL) un numero compreso tra 10000 e 4 miliardi (2e9) . E' poi necessario passare all' integratore la variabile di ingresso, che può essere il risultato di un calcolo e deve essere compresa tra 40000 e 4 miliardi (al fondo scala) per avere una precisione non inferiore a quella dichiarata.

Se PVAL1 = 40000 e IVAL1 = 40000 si avrà in uscita un impulso al secondo.

Esempio : si vuole totalizzare una energia (un impulso = 1 Mcal) misurando la differenza di temperatura tra mandata e ritorno e la portata di una condotta. AI1 e' la portata in mc/h, AI2 e' la temperatura alta, AI3 e' la temperatura bassa. I rispettivi fondo scala sono 2l/s, 0-50°C, 0-50°C.

La massima portata in Kcal/s è: $2 * 50 = 100$. Moltiplicheremo questa portata per 1000 per entrare nel range ammesso, e inseriremo questo valore in IVAL1.

Ora rimane da determinare il valore PVAL1: abbiamo la portata in Kcal , e vogliamo avere un impulso ogni 1000 Kcal. Dato che abbiamo moltiplicato per 1000 per entrare in range, faremo altrettanto con PVAL, quindi otteniamo 1000000.

istruzioni	commento
1000000	valore impulso
PVAL1	
AI2	
AI3	
-	differenza ingressi
50	fondo scala temperat.
*	
AI1	
*	
2	fondo scala portata
*	
1000	sistemazione range
*	
IVAL1	
STOP	
PVAL1, PVAL2	(registro valore di impulso \leftarrow n) memorizza nel registro il valore di impulso . ERRORE per n minore di 0.
IVAL1, IVAL2	(registro ingresso integratore \leftarrow n) trasferisce all' integratore il valore dalla variabile da integrare. ERRORE per n minore di 0.

ITYPE (registro tipo integratori \leftarrow d)
trasferisce al registro TIPO INTEGRATORI la variabile digitale d.
Quando la variabile digitale d è 0, il tipo di integratori è quello appena visto, cioè con ingresso analogico; quando d è diverso da 0, il tipo di integratori è quello che segue, cioè con ingresso digitale.
Non è possibile avere un integratore digitale e uno analogico contemporaneamente.

INTEGRATORE CON INGRESSO DIGITALE.

A differenza del precedente, questo integratore conteggia gli impulsi in ingresso (DI1 e DI2) e li moltiplica per una variabile analogica IVAL1, IVAL2.

Un impulso all'ingresso DI1 sarà moltiplicato per la variabile IVAL1 e sommato ai precedenti. Al raggiungimento del valore di impulso si avrà un impulso in uscita. Gli impulsi in ingresso possono avere una frequenza di 20Hz mentre quelli in uscita possono avere una frequenza massima di 5 Hz. Valgono le stesse indicazioni date per il tipo analogico per quanto riguarda i range ammissibili per PVAL e IVAL.

IVAL1, IVAL2 (registro ingresso integratore \leftarrow n)
trasferisce all' integratore il valore della variabile da moltiplicare agli impulsi in ingresso . ERRORE per n minore di 0.

5. USO DEL PROGRAMMA ' S2000C.EXE'

Come già detto il compilatore S2000C converte il file di testo in forma intelligibile dal S2000.

Per lanciare la sola compilazione del programma senza trasferirlo all' S2000 battere sulla tastiera : **S2000C** **nomefile.ext** **<invio>**.

La scrittura **<invio>** significa premere il tasto INVIO sulla tastiera, in altri casi contrassegnato con ENTER o RETURN, mentre il simbolo significa che bisogna battere uno spazio.

Verranno stampati gli errori se presenti, e i relativi numeri di linea.

Per inviare il file all' S2000, aggiungere il comando /t .

Esempio : **S2000C** **nomefile.ext** /t **<invio>**.

Il programma S2000C rimane attivo finché non si preme il tasto **ESC**, per permettere di visualizzare i messaggi di errore eventualmente ricevuti dal S2000.

E' possibile anche lanciare il programma per ottenere solo i messaggi di errore dall' S2000, senza compilare o trasferire alcun programma.

Per fare questo basta dare il comando **S2000C** **<invio>**, senza dare alcun nome di file.

Nel dischetto troverete un esempio di file di programma **pro.txt**. Provate a lanciare il programma come segue: **S2000C** **pro.txt** /t **<invio>** naturalmente dopo aver collegato lo strumento al PC (porta seriale COM1) e alla rete.

Dopo qualche secondo sul video apparirà la scritta : **trasmissione al S2000** seguita da: **trasmissione terminata** . Questo significa che l' operazione è stata eseguita correttamente.

Se non dovesse apparire la scritta : **trasmissione terminata**, premere il tasto **ESC** e ripetere l' operazione finché non si ottiene il risultato descritto.

6. CODICI DI ERRORE

Durante il caricamento del programma o l' esecuzione dello stesso, si possono verificare alcuni errori che vengono inviati allo schermo sottoforma di codici di errore es: E05, #E02. La lista che segue espone per esteso il significato di ciascun codice.

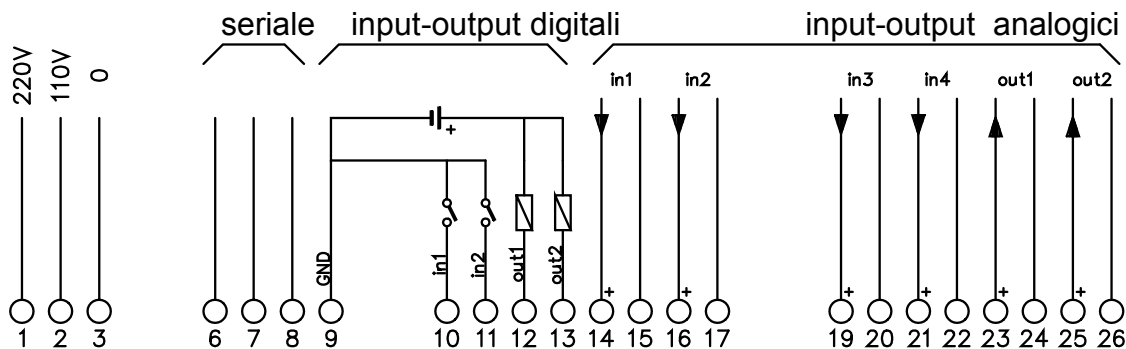
Codice Errore	Significato	Azione da intraprendere
#E01	Errore di trasmissione	Ritentare la trasmissione
#E02	Errore di protocollo	come sopra
#E03	Trasmissione interrotta	Controllare l' integrità del cavetto
E04	Riservato	
E05	Divisione per zero	Rivedere sequenza istruzioni
E06	Radicando negativo	c.s.
E07	Riservato	
E08	Logaritmo di numero negativo o nullo	c.s.
E09	Potenza con base negativa	c.s.
E10	Codici eeprom errati	Ricaricare programma; se l' errore permane probabile guasto EEPROM.
E11	Riservato	
E12	Valore impulso negativo	c.s.

N.B. : I primi tre errori contrassegnati con # possono accadere solo durante la trasmissione del programma al S2000.

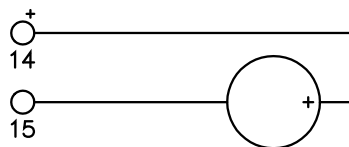
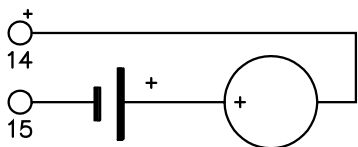
7. SPECIFICHE TECNICHE

- Ingressi digitali : tensione a contatto aperto 5V, corrente a contatto chiuso 1mA.
- Uscite digitali : max 300mA , 30Vcc.
- Ingressi analogici : 0..20mA o 4..20mA con resistenza di ingresso 100 ohm.
- Uscite analogiche : 0..20mA o 4..20mA attive, massima resistenza di carico 300 ohm.
- Alimentazione : 110 o 220 Vca +- 10% 50 o 60 Hz, 3.5 VA.
- Temperatura e umidità : 0..+50°C, 90% a 40°C non condensante
- Contenitore: per aggancio su profilato 35mm (DIN46277) o fissaggio a vite, in NORYL auto estinguente.
- Dimensioni e peso: 157.5 x 95 x 69 mm, 500 g
- conversione analogica a 12 bits sia in ingresso che in uscita (4096 punti di risoluzione)
- precisione totale migliore dello 0,3 %.
- velocità di conversione di 150 mS per ogni canale.
- ingressi e uscite digitali e analogiche non isolate (negativo in comune).
- LED verde per segnalazione presenza alimentazione.
- LED rosso per segnalazione mal funzionamento (lampeggiante).
- interfaccia RS232-C optoisolata per programmazione.

8. SCHEMA DI COLLEGAMENTO



Particolare di collegamento per gli input analogici:



Collegamento di un sensore passivo.

Collegamento di un sensore attivo.

L' S2000 dispone di interfaccia RS232-C ai morsetti 6,7,8.

Questi vanno collegati attraverso un cavetto appositamente cablato come appare nelle seguenti illustrazioni

