Manuale d'uso del SUPER-CHILDZ

(copia completa del documento originale)

PROGRAMMA MONITOR

Il programma inizia da ED3F e i comandi possibili sono i seguenti:

P:
Imposta il "POINTER" ad un indirizzo. I prossimi 4 numeri esadecimali sono l'indirizzo.

M:
Esegue il display della memoria Vengono esposte 24 locazioni di memoria a partire da dove punta il "PONTER". Il "POINTER" viene incrementato di 24.

T:
Serve a caricare dati in memoria a partire da dove punta il "POINTER". Se il comando immediatamente precedente è stato un "M", il "POINTER" vine portato all'inizio dell'ultimo campo esposto ed esso viene ricopiato per consentire eventuali modifiche.i successivi nuneri esadecimali sono quell da caricare. La pressione del tasto RETORN fa caricare tutti i caratteri precedenti al cursore. S sono un numero dispari l'ultimo carattere viene peduto.

<::
Serve per spostare il cursore a sinistra.

>:
Serve per spostare il cursore a destra.

K:
Serve per abblencare tutto lo schermo.

G:
Serve per andare alla locazione puntata dal "POINTER".

ESC:
Serve per ridare il controllo al monitor. I registri del programma in esecuzione vengono salvati ed esposti sul video nell'ordine che segue: PC, SP, AF, BC, DE, HL, IX, IY, AF', BC', DE', HL'.

H:
Serve per settare un HALT nella locazione puntata da "Pointer". Il prossimo carattere numerico indica il codice dell'HALT. Quando durante l'esecuzione del programma si incontra un HALT così preparato il programma si interrompe e si accende il LED di HALT sul ChildZ. Alla pressione di ESC il controllo torna al monitor e vengono evidenziate su video sia i registri del programma al momento dell'HALT sia il codice dell'HALT incontrato. È possibile a questo punto adoperare tutti i controlli del monitor E poi rientrare nel programma interrotto con l'automatico ricaricamento dei registri del programma al momento della sospensione.

X:
Serve riprendere un programma sospeso come un comando di ESC. I registri del programma vengono ripristinati.

V:
Consente di variare i registi del programma prima di rientrare con un comando X. Per variare I registri del programma prima di rientrarvi, battere gli indirizzi in esadecimale alle richieste del monitor, per non variare e passare ad altri registri battere SPACE, per terminare le variazioni battere RETUN.

NOTE:

- Qualunque carattere diverso da quelli consentite viene perduto.
- Il cursore può muoversi solo nell'ambito del rigo dei comandi.
- La pressione di uno dei tasti di comando, annulla altri comandi in digitazione.

FILES DEFINITI

1.
Un file di tipo “text editor” è un file costituito a blocchi the 832 byte che contengono campi ASCII di lunghezza variabile ciascuno dei quali è delimitato dal BIT 7 ON sul suo primo byte. Il file termina con un campo dello stesso tipo che contiene /* . . . .

2.
Si dice ”file definito" un file di tipo text editor costituito da una parte definizione e da una parte dati. La parte definizione, che deve essere registrata all’inizio del file, è costituita da un certo numero di campi descrittori (max 40) sono preceduti e seguiti da un campo contenente $.
La parte dati Deve seguire la parte definizione ed è costituita da records. Ciascuno record a sua volta è un insieme di campi variabili. Il numero di campi di cui è costituito un record è fisso nell’ambito del file ed è uguale al numero di descrittori presenti nella parte definizione. Ogni descrittore della parte definizione descrive il corrispondente campo del record. Il descrittore è in effetti il nome del campo corrispondente nel record, se questo nome comincia per BLANK si intende che il campo corrispondente del record e numerico, altrimenti è alfanumerico. Un campo nel record è un campo variabile ASCII costituito solo da numeri e se è negativo seguito dal segno meno.

SYSIO

- SYSIO è un modulo già assemblato che raccoglie un gruppo di routine utili per la lettura e ho la scrittura di un file definito.
- Ciascuna routine può essere richiamata con CALL all'entry point unico di SYSIO che si chiama anch'esso SYSIO ed è alla locazione 0200H. La scelta della rondine è fatta a mezzo di un codice che deve essere preparato nell'accumulatore.
- I registri IX e IY sono preservati.

Le routines disponibili in SYSIO sono:

1.
OPEN DI FILE INPUT (codice 01)
Viene aperto il file di input. Se il file non è definito ritorna con A = 0. Se il file è definito torno con A = n dove n è il numero di campi ed HL punta ad una tabella di puntatori a campi definiti ASCII contenenti i descrittori. La tabella finisce con FFFF.

2.
OPEN DI FILE OUTPUT (codice 02)
Viene aperto il file the output.

3.
SCRITTURA DI $
Viene scritto u record $.

4.
LETTURA DI UN CAMPO (codice 04)
Legge il prossimo campo presente sul file di input nel campo definito ASCII puntato da HL. Se EOF torna con A = 0FFH. E il record di EOF è un campo definito ASCI puntato da Hl.

5.
SCRITTURA DI UN CAMPO (codice 05)
Scrive sul file di output il campo definito ASCII puntato da HL.

6.
LETTURA DI UN RECORD (codice 06)
Legge il record successivo nei campi definiti ASCII indicati dalla tabella puntata da HL (se qualche elemento della tabella è a ZERO il corrispondente campo del record viene saltato. La tabella termina con FFFF.Se EOF, A = 0FFH ed il record di EOF è in un campo definito ASCII puntato da HL.

7.
SCRITTURA DI UN RECORD (codice 07)
Scrive il prossimo record con il contenuto dei campi definiti ASCII descritti dalla tabella puntata da HL (se qualche elemento della tabella è a ZERO il campo corrispondente viene saltato). L tabella termina con FFFF.

8.

CLOSE FILE INPUT (codice 08)
Il file di input viene chiuso.

CLOSE FILE OUTPUT (codice 09)
Scrive il record di EOF contenuto nel campo definito ASCII puntato da HL e chiude il file di output.

ROUTINES DI UTILITA' DA EPROM 1

PROMPT (call EC4B)
Abblenca l'ultima riga dell'area video.

SCROLL (call EC40)
Sposta in alto il video di un rigo e abblenca l'ultima riga (chiama PROMPT). Distrugge HL, DE e BC.

RDTAST (callEc98)
Al termine il carattere letto è disponibile in A e nella locazione 0098H. Il carattere ESC è riservato all'uso del monitor. Nessun registro è distrutto.

SPEGNI (call EF6C
Spegne i display e i LED del ChildZ.

INESA (callEC00)
Trasforma un campo con caratteri ASCII in un campo con caratteri BCD. Solo i caratteri da 0 ad F sono ammessi in input in HL indirizzo del campo di partenza, in DE indirizzo del campo di arrivo,. In B numero di bytes del campo di arrivo. Al termine HL = campo di partenza +1, DE = cambia di arrivo +1 B = 0, ,A e C distrutti.

ESA (callEC22)
Trasforma un campo con caratteri BCD in un campo con caratteri ASCII. Il campo risultante conterrà solo caratteri da 0 a F In DE. Indirizzo del campo di partenza, in HL indirizzo del campo di arrivo, in B numero dei bytes del campo di partenza. Al termine HL = campo di arrivo +1, B = 0, A e C distrutti.

ABBL (callEC8A)
Abblenca tutta l'area video. Tutti i registri sono distrutti.

SCAMB (call EC80)
Scambia il contenuto di due campi. In HL e DE indirizzo dei campi da scambiare. In B numero dei bytes da scambiare. Al termine B = 0, HL e DE puntano al prossimo byte successivo all'ultimo scambiato.

SCAVID (call EC6F)
Scambia l'area video visibile con il comodo video. Tutti i registri sono distrutti.

ROUTINES DI UTILITA' DA EPROM 2

ESPONI (call E0BD)
Espone un messaggio a partire da BUF + 5 e fermandosi quando incontra un carattere cl BIT 7 in ON.

ANCLDI (call E0C0)
Come ESPONI ma espone il messaggio puntato da HL nel punto puntato da DE.

WAIT (call E2F5)
Aspetta un certo tempo. In TEMPO + 1 numero di tempi di 250 microsecondi da attendere. In TEMPO un moltiplicatore , NAX = FEFE. Al termine tutti i registri distrutti, anche i registri alternativi.

INIOUT (call E1D1)
Avvia il tape cassette di scrittura, inizializza la porta PIO1B per output, scrive 80 bits di sincronismo, azera CKSUM. Tutti i registri dostrutti, anche i registri alternativi.

INIIN (call E18B)
Avvia il tape cassette di lettura, inizializza la porta PIO1B per input, legge i bits di sincronismo, azzera CKSUM. Tutti i registri distrutti, anche i registri alternativi.

LEGCAR (call E144)
Legge u carattere e lo depone in A. Tutti i registri distrutti.

LEGGI (call E172)
Legge un gruppo di caratteri (usa LEGCAR) il cui numero è impostato in BC e lo depone nel campo indivcato da HL, somma ogni carattere letto in CKSU. Distrugge i registri.

SCRV (call E23D)
Scrive tanti bits di sincronismo su tape output quanti sono indicati in B.

SCRCAR (call E270)
Scrive il carattere contenuto in A e lo somma in CKSUM. Tutti i egistri sono distrutti.

METTI (call E25F)
Scrive un gruppo di caratteri (isa SCRCAR) il cui numero è impostato in BC.. Tutti i egistri sono distrutti.

FERMA (call E1BF)
Ferma i motori dei tape cassette di input e di output.

WRITEA (call E29C)Scrive su tape di output un programma. Richiede il nome del programma. Prime di scrivere è necessario preparare i campi IDENT = A5, NFILE = numero del programma, LEN1 e LEN2 con le lunghezze dei due pezzi che possno costituire il programma, STADR1 e STADR2 con i due indirizzi degli eventuali due pezzi di programma, EPADR con l'eventuale indirizzo di START del programma (altrimenti a 0000). Se il programma è in due pezzi, al messaggio PROGRAM NAME

rispondere RETURN. Se il programma è in un solo pezzo, al PROGRAM NAME inserire il nome del programma. Tutti i registri sono distrutti.

SOM (call E312)
Somma due campi contenenti numeri positivi in BCD. In HL indirizzo delle unità del campo di partenza (campo da sommare) in DE indirizzo delle unità del campo di arrivo, in C lunghezza del campo di partenza, in B lunghezza del campo di arrivo, Al termine BC. HL ed A distrutti, De = ordine alto del capo di arrivo.

SOT (call E328)
Sottrae due campi contenenti numeri BCD positivi. In HL indirizzo delle unità del campo di partenza (campo da sottrarre), in DE indirizzo delle uinità del campo di arrivo, in C lunghezza del campo di partenza, in B lunghezza del campo di arrivo. Al termine HL, BC ed A distrutti, DE = ordine alto del campo di arrivo.

SOPP (call E33E)
Sopprime gli zeri non significativi. In DE ordine alto del campo, in B lunghezza del campo. Al termine DE punta alla prima cifra significativa del campo, B contiene il numero di cifre significative, A è distrutto.

COMPC (call E349)
Confronta due campi carattere. In DE indirizzo del primo campo, in HL indirizzo del secondo campo, in B lunghezza dei campi. Al termine si può eseguire JP Z per DE = HL, JP NZ per DE <> HL, JP P per DE > HL, JP N per DE < HL. Al termine DE e HL puntano al primo carattere diverso o al primo carattere successivo ai campi (se uguali), B contiene il numero di caratteri disuguali, A è distrutto.

RMESS (call E358)
RMESS1 (call E35B)
RMESS2 (call E35E)
RMESS3 (call E369)
Legge da tastiera carattere per carattere e porta da BUF + 4 a BUF + 60. Se si entra da RMESS1 porta da dove punta DE fino a BUF + 60. Se si entra da RMESS2 porta da dave punta DE a dove punta HL. Se si entra da RMESS3 i valori di inizio e di fine rimangono quelli precedentemente impostati (è possibile variare il valore di inizio mettendolo in COMHL e di fine mettendo in COMB la lunghezza richiesta). E' gestito il cursore. Il tasto RUBOUT cancella l'ultimo carattere introdotto.. CNTRL/Z cancella tutto il rigo e riporta il cursore all'inizio. Il tsto RETURN provoca l'abblencamento dal cursore in poi e il ritorno alla routine chiamante. In H viene caricato il numero di caratteri battuti. DE punta all'ultimo carattere battuto, La pressione di CNTRL/tasto alfanumerico (eccetto D) carica il valore del tasto nel campo COMAND e fa ritornare alla routine chiamante con le modalità del tasto RTURN (negli altri casi il campo COMAND viene messo a ZERO). Battendo ZERO caratteri la lunghezza del messaggio viene considerata 1 (in HL).

ROUTINES DI UTILITA' DA EPROM 3. CAMPI DEFINITI.

Un campo definito è quello preceduto da un byte di controllo il cui significato è:

Bit 7 = 0 Il cmpo che segue è ASCII
Bit 7 = 1 Il campo che segue è BCD

BIT 6 = 0 il campo che segue è CARATTERE
BIT 6 = 1 il campo che segue è NUMERICO

BIT 5 (ha valore solo se il campo è numerico)
BIT 5 = 0 il campo numerico che segue è POSITIVO
BIT 5 = 1 il campo che segue è NEGATIVO

BITS da 5 a 0 (se il campo è carattere) lunghezza – 1 del campo che segue
BITS da 4 a 0 (se il campo è numerico) lunghezza – 1 del campo che segue

Di conseguenza la massima lunghezza di un campo CARATTERE (ASCII o BCD) è di 64 bytes, mentre la massima lunghezza di un campo numerico (ASCII o BCD) è di 32 bytes, cioè 32 cifre con segno se ASCII o 64 cifre con segno se BCD.

Una serie di routines di utilità consentono di effettuare operazioni logiche e aritmetiche tra campi definiti. Sono previste molte routines per spostamenti e conversioni.

CLEAR (E477)
Azzera o abblenca il campo definito puntato da HL.Il campo viene abblencato se è ASCII CARATTERE, viene azzerato con zeri binari se è BCD e con zeri ASCII se è ASCII NUMERICO. Se il campo è NUMERICO il segno viene messo a+. A e B vengono distrutti.

MOVE (call E4AC)
Sposta il campo definito puntato da HL nel campo definito puntato da DE. Lo spostamento, quando necessario, con conversione e l'allineamento è a sinistra per i campi carattere a destra per quelli numerici. Le traduzioni impossibili si traducono in NOP. Tutti i registri vengono distrutti.

LOAD (callE4CF)
Sposta il campo definito puntato da HL nell'area puntata da DE. Tutto il campo, compreso il byte di definizione, viene spostato. Tutti i registri vengono distrutti.

COMP (call E4E5)
Confronta due campi definiti puntati da HL e DE, purchè dello stesso tipo, anche se di lunghezze diverse. Il confronto è fatto da sinistra a destra se campi carattere, è algebrico se i campi sono numerici. Se i campi non sono dello stesso tipo i risultati sono imprevedibili. Dopo la CALL è consentito di eseguire un JP Z, JP N, JP P, ecc. facendo riferimento al campo puntato da DE. Per testare >= o <= eseguire sempre prima il test per =. Tutti i registri vengono distrutti.

COMPAS (call E529)
Come la COMP solo che il confronto è fatto in assoluto purchè i campi siano numerici altrimenti si hanno risultati imprevedibili.

ADD (call E5A1)
Somma algebricamente il campo definito puntato da HL nel campo definito puntato da DE. I campi devono essere numerici BCD. I due campi possono avere lunghezze diverse. L'eventuale overflow viene perso. Tutti i registri vengono distrutti.

SOTT (call E597)
Sottrae algebricamente il campo definito puntato da HL nel campo definito puntato da DE.I campi devono essere numerici BCD altrimeni risultati sono imprevedibili. I due campi possono avere lunghezze diverse. Tutti i registro vengono distrutti.

BINBCD (call E6A3)
Trasforma il campo binario positivo di 2 bytes puntato da HL in un campo definito BCD puntato da DE. Il campo binario deve essere nella forma Z80 (cioè prima il byte di ordine basso e poi quello di ordine alto). Se il campo di arrivo non è BCD numerico, la BINBCD si traduce in un NOP. Tutti i registri vengono distrutti.

BINB (call E6B8)
Come la BINBCD eccetto che viene trasformato un solo byte binario puntato da DE.

BCDBIN (call E651)
Il campo definito puntato da HL viene trasformato in un campo binario di 2 bytes nella forma Z80. Se il campo di partenza non è BCD numerico positivo o se è maggiore di 65525, la BCDBIN si traduce in un CLEAR del campo di arrivo. Tutti i registri vengono distrutti.

EDITS (call E5F2)
Sposta il contenuto di un campo definito ASCII numerico puntato da HL in un'area il cui indirizzo di inizio è in DE. Gli zeri non significatvi vengono soppressi, se ilcampo di partennza è negativo, viene esposto il segno MENO dopo la cifra delle unità. Se il campo di partenza non è ASCII NUMERICO i risutati sono imprevedibili. Tutti i registri vengono distrutti.

EDITSND (call E5FD)
Come EDITS solo che il campo di partenza puntato da HL può essere non definito; iun questo paso però la lunghezza deve essere in B mentre C deve essere a 0 se il campo è positivo o avere il BIT 5 a 1 se il campo è negativo. Tutti i registri vengono distrutti.

EDITV (call E617)
Come EDITS eccetto che il campo viene spostato a partire dalla prima cifra significativa.

EDITVND (callE622)
Come EDITSND eccetto che il campo viene spostato a partire dalla prima ifra signiicativa

PUTTA (call E702)
Sposta in buffer di scrittura, se c'è posto, il record logico, altrimenti scrive su TAPE il buffer e poi sposta il record. Per le modalità precise leggere le informazioni sulla scrittura di records. Tutti i registri vengono distrutti.

GETTA (call E778)

Sposta nella workarea utente il prossimo record logico prelevandolo dal buffer. Se il buffer è vuoto viene letto un altro record fisico da nastro. Per le modalità pecise leggere le informazioni sulla lettura di recors.

CLOSEO (call E6D4)
Serve a chiudere un file di output. L'ultmo buffer rimasto in memoria viene scritto su nastro. Vedere le istruzioni più precise nella parte che descrive la scrittura di records su nastro. Tutti i registri vengono distrutti.

DISALT (call E7D6)
Il contenuto dell'area ADISPL viene esposta sui 6 display del ChildZ . I bytes a blank non vengono esposti. La routine è definitiva (rimane in loop su se stessa) si può uscire solo con ESC.

ABLALT (call E7F2)
L'area ADISPL viene abblencata, I registri HL, DE e BC vengono distrutti.

ABLGEN (call E7F8)
L'area puntata da HL viene abblencata per i bytes indicati da BC + 1 byte. I registri HL, DE e BC vengono distrutti.

ROUTINES DI UTILITA' DA EPROM 4

MULT (call E807)
Moltiplica algebricamente il campo definito puntato da HL per il campo definito puntato da DE. Il risultato è in (DE). I campi devono essere numerici e BCD altrimenti l'operazione non è eseguita. Tutti i registri vengono distrutti. Per migliorare la velocità mettere in DE il fattore con il maggior numero di cifre.

DIV (call E8B1)
Divide algebricamente il campo definito puntato da HL per il campo definito puntato da DE, Il quoto si avrà nel campo definito puntato da BC e il resto in (HL). Tutti i registri vengono distrutti. Per migliorare la velocità mettere il divisore in un campo definito che abbia tanti digits quante sono le cifre significative del dividendo.

PRENDI (call E993)
Il campo definito ASCII puntato da DE, viene riempito partendo da destra verso sinistra e estendendo con zeri a sinistra con i numeri ASCII che si trovano alla sinistra di (HL), eventuali blanks prima del numero vengono persi. Il primo blank a sinistra del numero prova la fine del trasferimento, se il numero ha più cifre del campo definito di arrivo viene troncato a sinistra. I registi vengono distrutti tranne HL che al termine conterrà l'inizio del campo definito.

PUTR (callE9BA)
La PUTR richiede che sia stato fatto un open della stampa (vedi stampa apposita).

ABBRIG (calll EA09)
Consente di abblencare il rigo di stampa patto che sia stato fatto l'OPEN della stampa. Distrugge HL, BC, DE, A.

LINFED (call EA21)
Consente di ottenere delle spaziature verticali sulla LA36. Occorre che sia stato fatto l'OPEN della stampa. In B deve essere messo il numero di spaziature richieste (è valido anche ZERO). Se le spaziature richieste fanno raggiungere la fine del modulo si ottiene un FORMFEED.

FORFED (call EA3A)
Consente di ottenere il salto al modulo successivo purché sia stato fatto l'open della stampa.

PRICAR (call EA58)
Consente di stampare il carattere contenuto in A sulla LA36 o sul video a seconda se il bit ZERO di FLARIG è 1 o 0.

DELAY (call EA8D)
Questa routine fa lampeggiare il cursore nella estremità destra in basso del video. Alla pressione di un tasto diverso da 1 a 9 la routine viene terminata. La pressione di un tasto da 1 a 9 fa si che la routine venga terminata dopo un tempo proporzionale al numero battuto.

DELAY1 (call EA8F)
Come DELAY solo che il tempo viene regolato dal valore impostato in A.

PRLA36 (call EADD)
Il carattere presente in A viene stampato sulla LA36.

EDITP (call EB04)
Il tempo definito ASCII viene spostato dove indica DE con soppressione di zeri non significativi e con punteggiatura. Se il campo è negativo viene esposto un MENO alla destra del numero.

HARDCP (call EB52)
L'ultimo rigo del video viene esposto su LA36, Segue uno SCROLL.

HARD1 (call EB55)
Come HARDCP eccetto che la stampa su LA36 parte da dove punta HL.

HARD2 (call EB58)
Come HARDCP eccetto che la stampa su LA36 parte da dove punta HL e termina dove punta DE.

FORM (call F041)
Riceve una tabella puntata da HL, formata da tante entrate create con la Macro FORMAT e terminata con FF. Espone un video le costanti, richiede campi di input numerici o alfanumerici, espone i defauld e richiama le routines di controllo utente.

RIESP (call E003)
Ingresso al control program.

RIESP1 (call F1D9)
Ingresso al control program con attesa. Battendo un tasto si prosegue e si va a RIESP.

ROUTINES DI UTILITA' DA EPROM 5. CAMPI BINARI

Un campo binario è un campo di 6 bytes che contiene un numero binario. Il byte di ordine più alto è quello con indirizzo più basso. Si il numero è negativo esso è rappresentato nella forma complemento a 2.

CVB (call F4EB)
Consente di trasformare un numero ASCII in un campo binario. Il numero può contenere la punteggiatura, se negativo deve avere un carattere MENO alla sua destra, non può contenere blanks, è delimitato alla sua sinistra da un blank e non può contenere più di 15 cifre significative. Prima di richiamare la CVB, DE deve contenere l'indirizzo del campo binario di arrivo e HL un indirizzo che sia a destra del campo SCII e che disti…….*qui purtroppo manca la continuazione*

CVD (call F57A)
Consente di trasformare un campo binario in un campo definito ASCII di 15 bytes. Il campo di arrivo può essere sono definito con DEFS 16 perché anche il byte di definizione viene prodotto dal CCVD. Ne risulterà quindi un campo definito con segno che conterrà un numero ASCII di 15 cifre al massimo. Prima di richiamare la CVD, HL deve contenere l'indirizzo del campo binario di partenza e DE l'indirizzo dello spazio di arrivo (di 16 bytes) per il campo definito ASCII risultante. Sono modificati i registri A e BC.

SOMB (call F%%C)
Consente di sommare algebricamente il campo binario puntato da HL nel campo binario puntato da DE. I registri A e BC vengono distrutti.

SOTB (call F56B)
Consente di sottrarre algebricamente il campo binario puntato da HL nel campo binario puntato da DE. I registri A e BC vengono distrutti

MULTB (call F5E0)
Consente di moltipiicare algebricamente il campo binario puntato da HL per il campo binario puntato da DE. I prodotto è memorizzatone nel campo binario puntato da DE. I registri A e BC vengono distrutti.

DIVB (call F6B6)
Consente di dividere algebricamente il campo binario puntato da DE per il campo binario puntato da HL. Il quoto è memorizzato nel campo binario puntato da DE e l sto nel campo puntato da HL. I registri A e BC vengono distrutti.

COMPB (call F7CC)
Consente di confrontare algebricamente il campo binario puntato da DE con il campo binario puntato da HL. Dopo la call è consentito utilizzare JP Z, o JP NZ, o JP M per (DE) minore, JP P per (DE) maggiore. I registri A e BC vengono distrutti.