

Quando vi abbiamo presentato il nostro micro-computer vi abbiamo detto che sulla scheda CPU va inserita una EPROM 2708 da noi programmata senza la quale il micro-computer non è in grado di svolgere alcuna funzione.

All'interno di tale EPROM infatti sono contenute tutte le istruzioni necessarie alla CPU per poter eseguire i vari ordini che noi le forniamo tramite la tastiera esadecimale, per esempio quando pigiamo i due tasti

CONTROL - 1

la CPU leggendo nella EPROM capisce che deve farci vedere il contenuto di una cella di memoria; quando pigiamo i due tasti

CONTROL - 2

capisce che deve farci vedere il contenuto dei registri A-F-B-C-D-E-H-L-SP-PC; quando pigiamo i due tasti

un certo signor Deluca che esso non conosce e non sa dove abita.

Se noi gli diciamo:

«Prendi questo pacco e portalo al signor Deluca» immediatamente ci chiederà:

«Chi è il signor Deluca?»

«In quale via abita?»

«A che ora lo trovo in casa?»

«Debbo andarci in macchina o mi conviene prendere l'autobus?»

«Dopo aver consegnato il pacco ritorno indietro subito oppure devo aspettare una risposta?».

In altre parole per consentire a Marco di consegnare il pacco al signor Deluca noi dovremmo prima risolvere tutti questi interrogativi fornendogli precise istruzioni in proposito, cioè dovremmo dirgli:

1) Prendi questo pacco.

ALLA SCOPERTA del

CONTROL - 4

capisce che deve eseguire il programma da noi scritto in precedenza sulla memoria RAM; infine quando pigiamo

CONTROL - 5

capisce che deve trasferire il contenuto della memoria RAM su nastro magnetico.

Queste istruzioni contenute nella EPROM realizzano nel loro insieme un programma che viene chiamato genericamente MONITOR e che occupa le locazioni di memoria dalla **8000** alla **83FF**.

Se noi andiamo ad analizzare il MONITOR passo per passo ci accorgeremo che esso risulta costituito da un **programma principale** generico più tanti **programmini secondari specializzati** ciascuno per compiere una ben determinata funzione, i quali vengono di volta in volta richiamati dal programma principale e fatti eseguire a seconda degli ordini da noi impartiti tramite la tastiera esadecimale.

Questi programmini secondari si chiamano «subroutine» e la loro particolarità peculiare è che possono essere richiamati e fatti eseguire anche tramite un programma scritto dall'operatore sulla memoria RAM permettendo così di semplificare notevolmente la programmazione.

Per chi non sapesse che cosa è una «subroutine» e quali vantaggi si possono ottenere dal suo impiego faremo ora un esempio molto semplice che però, almeno nel nostro intento, dovrebbe servire a chiarire le idee anche ai più inesperti.

Supponiamo di avere un fattorino di nome Marco e di doverlo mandare a consegnare un pacco ad

2) Vai alla fermata dell'autobus n. 14.

3) Sali sull'autobus e scendi alla quinta fermata.

4) Vai avanti 100 metri a piedi poi svolta sulla destra in via Marconi e davanti al terzo portone fermati.

5) Suona dal portiere per farti aprire poi sali al secondo piano, prima porta sulla destra.

6) Bussa alla porta e chiedi del signor Deluca.

7) Se non è in casa puoi lasciare il pacco alla moglie oppure alla cameriera.

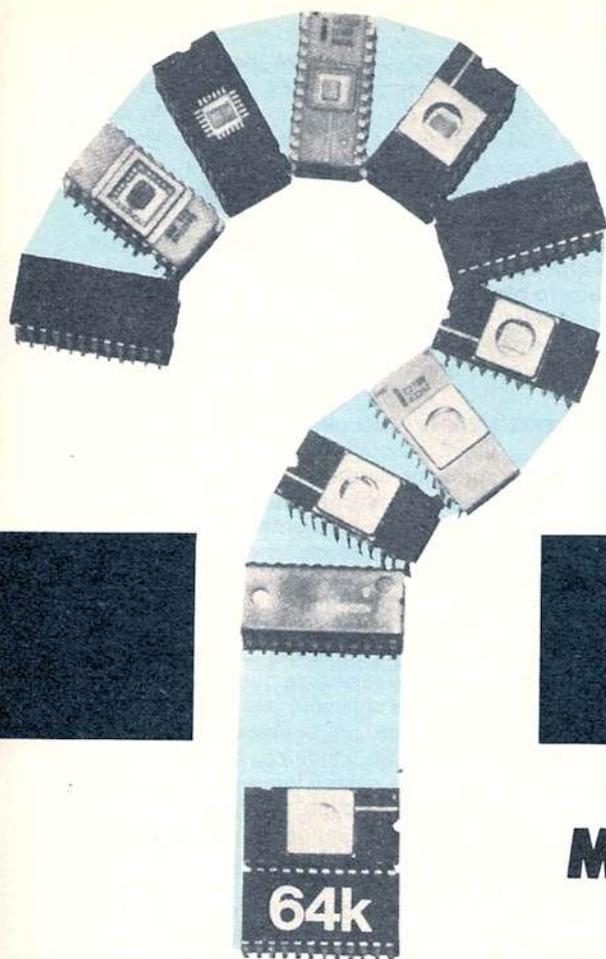
8) Fatto questo ritorna alla fermata dell'autobus a cui eri sceso.

9) Per tornare devi prendere l'autobus n. 27 ed ancora devi scendere alla quinta fermata che è quella qui all'angolo.

Come vedete per far compiere a Marco un'operazione molto semplice come lo è appunto consegnare un pacco ad una persona noi dovremmo prima fornirgli una serie piuttosto lunga di istruzioni, cioè fornirgli un vero e proprio programma di lavoro che esso avrà poi premura di eseguire.

Una cosa più o meno analoga avviene anche all'interno del nostro micro-computer, basti pensare che solamente per far comparire un certo numero su uno degli 8 display disponibili è necessario fornire alla CPU scrivendole sulla EPROM oppure sulla memoria RAM non meno di 10-15 istruzioni.

Se dopo qualche giorno dalla prima consegna noi abbiamo necessità di recapitare un secondo pacco al signor Deluca però questa volta vogliamo mandarci un altro fattorino che ha nome Andrea, per raggiungere lo scopo possiamo seguire due strade diverse e cioè:



EPROM abbiamo scritto tutta la serie di istruzioni necessarie per far comparire un numero sul display è inutile che poi torniamo a scrivere tutte queste istruzioni ogni volta che in un programma ci occorre di visualizzare sul display un determinato numero: meglio dire alla CPU «vai da Marco e fatti spiegare» oppure «vai da Andrea e fatti spiegare» e raggiungere così lo scopo con una sola istruzione, invece di doverne scrivere 10-15.

In realtà alla CPU noi non diremo «vai da Marco» o «vai da Andrea», bensì gli diremo per esempio:

CALL 80AA (esegui la subroutine 80AA) cioè gli forniremo ogni volta l'indirizzo della cella di memoria EPROM in cui è contenuta la prima istruzione della «subroutine» che vogliamo sfruttare ed essa automaticamente eseguirà tutto questo sottoprogramma, dopodiché tornerà al programma principale ed eseguirà l'istruzione immediatamente

MONITOR

MICROCOMPUTER Z80

Il programma MONITOR memorizzato nell'interno della EPROM che trova alloggio sulla scheda CPU del nostro microcomputer si compone di tanti sottoprogrammi o «subroutine» di uso comune che noi possiamo in qualsiasi istante richiamare e far eseguire semplificando così notevolmente la programmazione.

1) Ripetere ad Andrea tutte le istruzioni che già avevamo fornito a Marco la prima volta perdendo così un bel po' di tempo in spiegazioni.

2) Consegnare il pacco ad Andrea e dirgli semplicemente:

«Chiedi a Marco cosa devi fare per recapitare il pacco».

Come vedete la seconda soluzione ci permette di risparmiare una notevole quantità di tempo infatti con la sola istruzione:

«Vai da Marco e chiedi cosa devi fare»

possiamo risolvere lo stesso problema per cui la prima volta sono state necessarie ben 9 istruzioni.

Ebbene Marco in questo nostro esempio fa le veci di una delle tante «subroutine» contenute nella EPROM infatti una volta che noi su questa

successiva a quella che l'aveva mandata alla subroutine.

Con questo dovrete avere compreso l'enorme importanza che riveste il fatto di conoscere alla perfezione il MONITOR in tutte le sue «sfumature» ed è proprio per tale motivo, cioè per consentirvi di sfruttare al massimo tutte queste subroutine nei vostri programmi risparmiando così un notevole spazio sulla memoria RAM e semplificando notevolmente la programmazione, che noi oggi abbiamo deciso di elencarvele una per una in questo articolo.

Non solo ma parlando delle subroutine avremo modo anche di spiegarvi a grandi linee come vengono gestite le «periferiche» cioè le varie schede di interfaccia, nel nostro microcomputer e questo

vi servirà moltissimo per imparare ad usarlo meglio e a sfruttarlo nella pienezza delle sue possibilità.

SUBROUTINE 80AA (Inserimento dati da tastiera)

Questa subroutine ci permette di caricare nel registro A (cioè nell'accumulatore) quel numero che noi pigiamo sulla tastiera esadecimale.

In pratica quando noi facciamo eseguire alla CPU l'istruzione **CALL 80AA** (cioè vai alla subroutine 80AA), questa si mette automaticamente in attesa che un tasto venga pigiato e non appena noi pigiamo per esempio il tasto n. 5, sempre automaticamente carica il numero 05 nel registro A quindi passa ad eseguire l'istruzione immediatamente successiva nel programma originario.

SUBROUTINE 80EC (Uscita dati sui display)

Questa subroutine ci permette di far comparire sui display il numero contenuto nelle prime 8 locazioni della memoria RAM, cioè nelle locazioni dalla **0000** alla **0007**.

In pratica facendo eseguire l'istruzione **CALL 80EC** il numero o la lettera dell'alfabeto contenuta nella locazione 0000 viene visualizzata sul display 0, cioè il primo sulla destra, il numero contenuto nella locazione 0001 viene visualizzato sul display 1, il numero contenuto nella locazione 0002 viene visualizzato sul display 2 e così di seguito fino alla locazione 0007 il cui contenuto viene visualizzato sul display 7 cioè sull'ultimo display a sinistra.

Se in una di queste locazioni di memoria fosse contenuto un numero superiore a 0F, sul relativo display compariranno altre lettere dell'alfabeto (oltre a quelle della tastiera esadecimale) oppure dei simboli strani come per esempio un trattino orizzontale, un trattino e un punto ecc. ecc.

SUBROUTINE 8003 (Shift a sinistra dei display 0-1)

Questa subroutine trasferisce il contenuto della locazione di memoria 0000 nella locazione 0001 e subito dopo carica il contenuto dell'accumulatore (registro A) nella locazione 0000.

Il contenuto precedente della locazione 0001 va irrimediabilmente perduto.

Poiché le locazioni di memoria 0000 e 0001 sono in pratica lo specchio dei display 0 e 1 è ovvio che utilizzando tale subroutine insieme alla 80AA e alla 80EC è possibile fare in modo che il numero pigiato sulla tastiera esadecimale compaia automaticamente sul display 0 e nello stesso tempo il numero che prima era visualizzato sul display 0 si sposti di una posizione verso sinistra e vada a finire sul display 1 che gli sta accanto.

SUBROUTINE 800C (Shift a sinistra del display 6-5-4-3)

Questa subroutine trasferisce il contenuto della

locazione di memoria 0005 nella locazione 0006, il contenuto della locazione 0004 sulla locazione 0005, il contenuto della locazione 0003 sulla locazione 0004 e per ultimo carica il contenuto dell'accumulatore A nella locazione 0003.

In pratica vale il discorso fatto in precedenza cioè utilizzando tale subroutine in abbinamento con la 80AA e la 80EC, come vedremo negli esempi che seguono, noi possiamo fare in modo che il numero pigiato sulla tastiera esadecimale compaia sul display 3 ed automaticamente i numeri contenuti in precedenza nei display 3-4-5 si spostino tutti di una posizione verso sinistra.

SUBROUTINE 8015 (Shift generico verso sinistra)

Questa subroutine esegue una funzione analoga alle due precedenti, cioè permette sempre di ottenere uno spostamento dei contenuti di talune locazioni di memoria nella locazione immediatamente successiva e di caricare quindi il contenuto del registro A nella locazione più bassa, però rispetto alle precedenti è più «flessibile» in quanto ci permette di scegliere le locazioni di memoria che debbono essere interessate da questo «shift».

In pratica, prima di chiamare tale subroutine con l'istruzione **CALL 8015**, noi dobbiamo scrivere nella coppia di registri HL l'indirizzo dell'ultima cella di memoria che deve essere interessata dallo shift (indirizzo più alto) poi dobbiamo scrivere nel registro C quante celle di memoria vogliamo che siano interessate da questo spostamento.

Per esempio ponendo HL=0001 e C=02 si otterrà lo stesso effetto che si ottiene con la subroutine 8003; ponendo invece HL=0006 e C=04 si otterrà lo stesso effetto che si ottiene con la subroutine 800C.

Se invece volessimo spostare verso sinistra i numeri contenuti in tutti gli 8 display e caricare il contenuto dell'accumulatore A nel display 0 dovremmo porre HL=0007 e C=08.

SUBROUTINE 8024 (Carica in A il contenuto dei primi due display)

Questa subroutine ci permette di caricare nell'accumulatore A il numero che appare visualizzato sui primi due display a destra.

Per esempio se sul display 1 è presente il numero 8 e sul display 0 il numero 5, facendo eseguire alla CPU l'istruzione **CALL 8024** (cioè esegui la subroutine 8024), ad esecuzione avvenuta nel registro A ci ritroveremo il numero 85.

SUBROUTINE 802A (Carica in A il contenuto di due display adiacenti)

Questa subroutine è simile alla precedente tuttavia è più «flessibile» come uso in quanto ci permette di scegliere la coppia di display il cui contenuto deve essere caricato nell'accumulatore A.

Per esempio se volessimo caricare nell'accumulatore A il contenuto dei display 5 e 4 dovremmo prima scrivere nella coppia di registri HL l'indirizzo 0005 (00 in H e 05 in L), cioè l'indirizzo della locazione di memoria che è in pratica lo specchio del display 5, quindi chiamare la subroutine 802A ed automaticamente il contenuto di questi due display verrà «compattato» in un unico numero a due cifre nell'interno del registro A.

SUBROUTINE 8034 (Carica il contenuto di A nelle locazioni 0001 e 0000)

Questa subroutine effettua praticamente l'operazione inversa rispetto alla 8024 infatti ogni volta che viene chiamata trasferisce il contenuto dell'accumulatore sdoppiandolo nelle locazioni di memoria 0001 e 0000 dopodiché, chiamando la subroutine 80EC, è possibile visualizzare questo numero rispettivamente sui display 1 e 0 (cioè i primi due sulla destra).

SUBROUTINE 803C (Carica il contenuto di A in due locazioni adiacenti)

Questa subroutine è in pratica l'inverso della 802A infatti se noi poniamo in HL ad esempio l'indirizzo 0006 poi chiamiamo la 803C, automaticamente il contenuto di A (sdoppiato) verrà caricato nella locazione di memoria 0006 e in quella immediatamente precedente, cioè nella 0005.

SUBROUTINE 8055 (Carica nei registri DE il numero contenuto nei display 6-5-4-3 dopo averlo compactato)

Questa subroutine ci permette di caricare nella coppia di registri DE il numero che appare visualizzato sui display 6-5-4-3.

Per esempio, ammesso che sui display risultino presenti i seguenti numeri:

display 6 = A
display 5 = F
display 4 = 8
display 3 = 1

facendo eseguire l'istruzione **CALL 8055**, cioè «vai alla subroutine 8055», automaticamente nella coppia di registri DE verrà caricato il numero AF81, cioè AF nel registro D e 81 nel registro E.

SUBROUTINE 8078 (Carica il contenuto di DE sdoppiato nelle locazioni 0006-0005-0004-0003)

Questa subroutine ci permette di caricare le quattro cifre che compongono il numero contenuto nella coppia di registri DE rispettivamente nelle locazioni di memoria 0006-0005-0004-0003 che come abbiamo detto in precedenza sono lo specchio dei display 6-5-4-3.



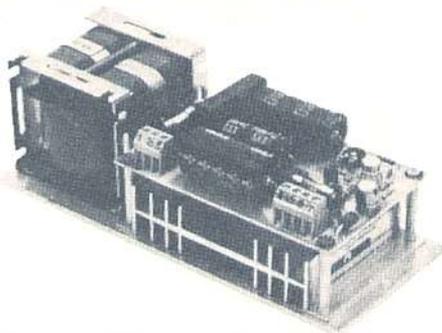
ITALSTRUMENTI s.r.l.

00147 Roma - Via del Caravaggio, 113
Tel. (06) 51.10.262 centralino

COMPONENTI ELETTRONICI

ALIMENTATORI STABILIZZATI

- protetti contro il corto circuito
- tensione di uscita regolabile dai 12V ai 14,5V
- corrente erogabile da 0,8 A/h a 15 A/h



ROBUSTI CONTENITORI METALLI PER LE PIU SVARIATE APPLICAZIONI

- alimentatori a giorno per usi generali
- centrali multimpiego (impianti dall'allarme, antincendio, ecc.)
- sirene (elettroniche e meccaniche)
- assemblaggi elettronici in generale da L. 10.000 a L. 40.000

CONVERTITORE c.c./c.a.

- trasforma la corrente continua delle batterie in tensione alternata 220 V. (INVERTER)
- possibilità di uso come carica batterie
- potenza nominale 500 W
- potenza di picco 850 W
- L. 270.000 disponibili altri modelli



LINEARI DI POTENZA

- Mod. 15 W - Mod. 50 W - Mod. 30 W
- Frequenza 26,8 + 27,5 MHz
- Impedenza d'uscita 50 Ohm
- da L. 29.000 a L. 75.000

ANTENNE OMNIDIREZIONALI PER RICETRASMISSIONE

- 27 MHz
- VHF 156 + 174 MHz
- UHF 430 + 470 MHz

EMETTITORI RICETRASMITTENTI PORTATILI

Ne consegue che se dopo la **8078** noi chiamiamo la subroutine **80EC**, automaticamente vedremo comparire il contenuto di **De** sui quattro display appena menzionati.

SUBROUTINE 8069 (Carica il contenuto del registro **A** nelle prime 8 locazioni della memoria **RAM**)

Questa subroutine ci permette di caricare il numero contenuto nel registro **A** in tutte le prime 8 locazioni della memoria **RAM**, dalla **0000** alla **0007** compresa.

Per esempio se il registro **A** contiene il numero **08**, facendo eseguire l'istruzione **CALL 8069**, automaticamente in tutte le locazioni dalla **0000** alla **0007** compresa ci ritroveremo il numero **08**, dopodiché chiamando la subroutine **80EC** potremo far comparire su tutti i display il numero **8**.

SUBROUTINE 8067 (Carica un **11** in tutte le prime 8 locazioni della memoria **RAM**)

Questa subroutine ci permette di caricare il numero **11** in tutte le prime 8 locazioni della memoria **RAM**, dalla **0000** alla **0007** compresa, e poiché un **11** se viene visualizzato fa spegnere il display, è ovvio che chiamando dopo di questa la subroutine **80EC** vedremo tutti gli 8 display spegnersi.

Precisiamo che questa subroutine, a differenza di tutte le altre, qualora venga chiamata, modifica il contenuto del registro **A**, quindi se interessa che il contenuto precedente non vada perduto è necessario salvarlo nello **STACK** con l'istruzione **PUSH AF** prima di chiamare la subroutine e richiamarlo quindi al termine con l'istruzione **POP AF**.

SUBROUTINE 80D6 (Genera un ritardo di 1 millisecondo)

Questa subroutine non ha nessuna funzione particolare se non quella di impiegare all'incirca 1 millisecondo per essere eseguita: può quindi essere sfruttata come base dei tempi per generare dei ritardi nel corso di un programma qualora si voglia per esempio far comparire un numero sui display per un determinato periodo di tempo prima di proseguire con il resto dell'elaborazione.

SUBROUTINE 808E (Genera un ritardo di durata variabile)

Questa subroutine è simile alla precedente con l'unica differenza che il ritardo generato è pari ad 1 millisecondo moltiplicato per il numero esadecimale contenuto nelle locazioni di memoria **0020** e **0021**.

Per esempio se in queste locazioni fosse contenuto il numero **0005**, cioè **05** nella locazione **0020** e **00** nella locazione **0021**, facendo eseguire la subroutine **808E** si otterrebbe un ritardo di $5 \times 1 = 5$ millisecondi.

SUBROUTINE 8094 (Genera un ritardo di durata dipendente dal contenuto di **HL**)

A differenza della precedente questa subroutine genera un ritardo pari ad 1 millisecondo moltiplicando per il numero esadecimale contenuto nella coppia di registri **HL**, quindi se in **HL** è contenuto per esempio il numero **000A**, chiamando la **8094** si otterrà un ritardo di 10 millisecondi (infatti **A** in esadecimale è uguale a 10).

SUBROUTINE 80E3 (Genera un ritardo fisso di 5 secondi)

Questa subroutine, a differenza delle ultime due che abbiamo visto, genera un ritardo costante pari a 5 secondi nell'esecuzione del programma, ritardo che potremo sfruttare, come già anticipato in precedenza, ad esempio per far comparire sui display il risultato intermedio di una certa elaborazione in modo da avere il tempo di leggerlo prima che l'elaborazione stessa proceda e tale risultato venga modificato.

SUBROUTINE 813C (Carica il contenuto di **HL** nelle locazioni **0006-0005-0004-0003** dopo averlo sdoppiato)

Chiamando questa subroutine, simile nelle sue funzioni alla **8078**, noi possiamo caricare le quattro cifre che compongono il numero contenuto nella coppia di registri **HL** rispettivamente nelle locazioni di memoria **0006-0005-0004-0003** dopodiché, chiamando la subroutine **80EC**, potremo far comparire tale numero sui display **6-5-4-3**.

Queste sono in pratica le subroutine che noi possiamo sfruttare con i nostri programmi tuttavia, così come le abbiamo elencate, potrebbero anche risultare incomprensibili per i più inesperti ed è proprio per fugare ogni ombra di dubbio che in un articolo a parte, su questa stessa rivista, vi porteremo alcuni esempi illustrativi del loro impiego.

Prima di concludere vorremmo ricordare a tutti quanti ce lo hanno richiesto telefonicamente o per iscritto che gli indirizzi da utilizzare per accedere tramite programma ad un singolo display sono i seguenti:

F0 = display 0
F1 = display 1
F2 = display 2
F3 = display 3
F4 = display 4
F5 = display 5
F6 = display 6
F7 = display 7