```
PPPPPP                                                                  |
PP   PP                  t             t                                |
PP   PP                  ttt           ttt                  i           |
PP  pp                   t             t                               |
PPPPP   u    u  n nnn    t      aaa    t      oooo   r rr  i            |
PP      u    u  nn   n   t         a   t     o    o  rr    i            |
PP      u    u  n    n   t     aaaaa   t     o    o  r     i            |
PP      u   uu  n    n   t     a   a   t     o    o  r     i            |
PP       uuu u  n    n    ttt   aaa a   ttt   oooo   r      ii           |
```

Brevissima, non completa e tanto meno esaustiva descrizione sull'utilizzo, notazione e calcolo dei puntatori per i PLC della famiglia S7 300.

. Abstract

In S7 300 esiste una famiglia di operandi e dati che prendono il nome di puntatori. Questi hanno un ruolo importantissimo sulla ricerca e indicizzazone dei dati. Essitono varie possibilità sull'utilizzo dei puntatori, vari metodi e notazioni. Qui si prenderà in considerazione solo una piccola parte.

Vista la pericolosità del software quando si utilizzano i puntatori l'autore non è in alcun modo responsabile di danni a persone, animali e cose provenienti dalla lettura di questo documento, tutti i nomi citati come S7 300 sono protetti dai marchi di fabbrica dei rispettivi fornitori/produttori.
Queste note sono a carattere personale, qualunque altro scopo e utilizzo deve prima essere vagliato dall'autore.
In queste note spesso si useranno termini non propriamente tecnici o corretti, non viene trattato l' utilizzo di ARR e compagnia ... proprio per dimostrare la potenzialità dei puntatori ;-).

. Cenni sui puntatori

S7 300 dispone di 3 tipi dipuntatori:

.1 Puntatore a 16 Bit
.2 Puntatore a 32 Bit
.3 Puntatore parametro ANY

Qui trateremo solo i primi due anche se il parametro ANY è a volte un chiccheria ;-).
Bisogna stare attenti a non confondere i primi due parametri, perchè servono a cose diverse.

. Puntatore a 16 Bit

I puntatori a 16 Bit vengono spesso usati assieme a variabili temporane di tipo intero. Il loro unico scopo di esistenza è per aprire i DB. L'esempio più banale infatti è proprio questo:

```
AUF DB20          // Apertura del DB 20
```

Se vogliamo indicizzare l'apertura del DB o comunque non essere soggetti a restrizioni possiamo, posto la variabile temporane N_DB un intero :

```
L   35            // Carica 35 in ACC1
T   #N_DB         // Trasferisci ACC1 in N_DB

AUF DB[#N_DB]     // Qui si apre la DB 35
```

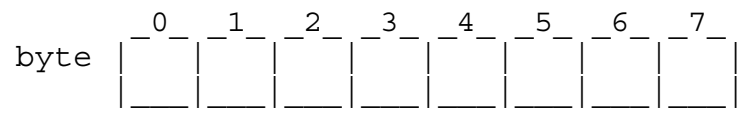
Lo scopo e l'utilizzo di questi puntatori è limitato solo per questo utilizzo. In definitiva non si tratta di un vero punata=tore come si utilizza in linguaggi tipo c/c++, pascal, ASSEBLY ..., ma di un operatore per aprire i DataBlock.

. Puntatore a 32 Bit

La definizione è errata in quato non utilizza lo spazio di indirizzamento totale di 32 Bit, ma solo di una parte. Non viene qui approfondito come e perchè, ma verranno date le basi solo per utilizzarli subito senza tante stramberie e perfezionamenti.

Vediamo prima di tutto come è composto un byte nel S7 300, in particolare si deve ricordare (a torto o a ragione che sia) che in S7 300 si lavora a byte, quindi tutti i nostri puntamenti su DB dovranno essere fatte a byte.

Un byte è fatto da 8 bit accatastati, per chi ha famigliarità con le liste diciamo che è la concatenazione di otto bit 8-).



Questa è larappresentazione grafica di un byte, ovvimente questo byte per essere reale farà parte di una memoria, quindi occuperà un indirizzo ben preciso. Astrattamente possiamo pensare che sia parte di un'area dimemoria, ad esempio un gruppo di registri, che in S7 300 potrebbero essere dei Merker Byte ...

Quindi se fosse così l'idirizzo di un byte in una macchina che lavora a byte dovrebbe essere rintracciato dalla formula:

Indirizzo = n_Byte, cioè l'idirizzo di un byte in una zona di memoria che inizia da 0 e ha M byte contigui è proprio dato dal posto in cui esso si trova.

Paradosalmente, a dimostrazione della validità, basti pensare a quando si dice quanto vale il MB44? ebbene sì, stiamo cercando il valore del byte che occuma la memoria di registr M nella posizione 44!

Per nostra fortuna, il puntatore in S7 300 è un po' più comples= so proprio per soperire ad alcune defficenze di questo modo di ragionare. Esempio, se volessimo puntare al bit 4 del byte come potremo fare?

Proprio per questo motivo S7 300 usa una notazione diversa per puntare ai dati.

Partiamo con un esempio:

```
U  M 5.4        // Voglimo interogare il bit 4 del byte 5.
```

S7 300 quindi usa un puntamento a bit e non a byte, cosa vuol dire questo? Semplice, se come abbiamo impotizzato prima, per calcolare l'indirizzo di un byte tramite una macchina che lavora a "bit" avremmo una fromuletta del tipo:

```
  Indirizzo = n_Byte * 8 + Offset_Bit
```

Ebbene si, con alcune riserve questa è proprio la formula da utilizzare per calcolare l'idirizzo di un bit in un byte.

Esempio:

```
   U  M 5.4         // Interrogo il bit 4 del byte 5 dell'area M

   // allo stesso modo

   L  5             //
   ITD              // da INT A DINT
   L  L#8           // carico una costante a 32 Bit
   *D               // Moltiplicazione a 32 Bit
   L L#4            //
   +D               //
   T  #INDIRIZZO    // variabile temorane DINT

   U  M[#INDIRIZZO] // stessa cosa di fare M 5.4
```

. Conclusione

Pur sapendo di non aver trattato l'argomento in modo esaustivo, ma di averlo sorvolato, ho mostrato con poche righe le potenzia= lità dei puntatori utilizzati in modo normalissimo, cioè come semplici formulette. In conclusione una dimostrazione dell' uso che si potrebbe fare:

```
 Condizioni al contorno : DB genrico, formattato come insieme di
                          byte da 0 a 100. Si vuole scandire
                          tutto il DB leggendo ogni singolo byte
                          e sostituendo il precedente con la
                          media di entrambi ;-) ...
                          In termini matematici :
                          f(n)   = (f(n+1)+f(n))/2
                          f(Max) = f(Max)


      L  #i_N_DB          // Variabile di ingresso dellafunzione
      T  #N_DB            // Carico su temporanea intera

      AUF DB[#N_DB]       // Apro il DB richiesto

      L  100              //
Loop: T  #CONT            // TEMPORANEA BYTE
      L  100              // 100 - CONT = 0 ... 100
      L  #CONT            //
      -I                  //
      ITD                 // trasformo in un DINT
      L  L#8              //
      *D                  // Ora ho un puntatore che punta alla
```

```
                        // (100-CONT)_esima locazione
      T  #PUNT          // Variabile temporanea a 32 Bit
      L  DW#16#8        //
      +D                //
      T  #PUNT_1        // Puntatora all'elemento successiovo

      L  DBB[#PUNT]     //  f(n)
      L  DBB[#PUNT_1]   //  f(n+1)
      +I                //  f(n+1) + f(n)
      L  2              //
      /I                //  (f(n+1) + f(n))/2
      T  DBB[PUNT]      //

      L #CONT           //
      LOOP Loop         //
```

Come si è visto è molto semplice utilizzare i puntatori, questo metodo ha ilvantaggio di essere semplice e con poche variabili supplementari andar a fare divertenti giochetti ;-), certo però che per una perfetta padronaza si devono utilizzare in congiunta i registri ARR ... Ma questa è tutta un'altra storia.

. Ultimi cenni sul calcolo

La forma numerica di un classico pntatore in S7 300 è dichiarata in questo modo secondo la classica notazione:

```
     L  P#5.6           // carichiamo bit 6 byte 5
     T  #Temp_32_bit    // usiamo una temporanea
     U  M[#Temp_32_bit] // interoghiamo il bit
```

Tutte queste stramberie nascono dalla semplice notazione che nel S7 300 possiamo rappresentare i puntatori come una sequenza di dove alcuni gruppi hanno particolare significato:

```
              Indirizzo byte        Indirizzo bit
              ______________________ ___
 Puntatore:   xxxxxxxxxxxxxxxxxxxxx.nnn

 quindi l'indirizzo per esempio : 4.2 in forma a bit viene
 rappresentato come             : 100.010
```

Altri esempi per comprendere meglio :	Puntatore	forma a bit
	7.5	111.101
	10.7	1010.111
	15.1	1111.001
	16.0	10000.000

Senza togliere suspance a nessuno ;-) la tabellina indica chiaramente la base di numerazione dei puntaori, infatti come già accennato prima la base è 8, infatti 8 sono i bit del byte e

quindi per logica anche la base deve essere otto.

In pratica lavorando in base 8 (Ottale per la cronaca ;-)) i calcoli deventono molto semplici, ma senza scomodare labase 8 e uttilizzando la più famigliare notazione hexadecimale, i calcoli vengono semplificati, infatti:

```
 Puntatore a bit  : N_Byte *  8_hex + Offset_Bit
 Puntatore a byte : N_Byte *  8_hex
 Puntatore a word : N_Byte * 10_hex   (10_hex = 16_dec = 8_dec*2)
```

Se l'esempio di prima ivece di byte fossere state 100 word, il calcolo del puntatore sarebbe stato modificato e convertito circa in questo modo:

```
 L  #CONT
 L  DW#16#10
 *D
```

Esistono molti altri giochetti che si possono fare ma questi ven= gono con la pratica.

. Cenni sugli ANY

Più che cenni solo 2 parole.
I parametri ANY sono pntatori di strutture, sono lunghi più di 4 Byte e la loro struttura può memorizzare che tipo di dato è,i che spiazzamento di memoria si rova ...
L'utilizzo di questi parametri è fatto spesso in congiunta con la funzione BLOK_MOVE del S7 300.