

# **ΘΕΜΑΤΙΚΗ ΕΝΟΤΗΤΑ 7**

***Προγραμματιζόμενη Λογική***

## Εκπαιδευτική Δραστηριότητα Δ2: «Καταιγισμού Ιδεών»

---

**Τίτλος:** Τι εννοούμε σαν Προγραμματιζόμενη Λογική

# Περιεχόμενα

## Θεματικής Ενότητας 7: Προγραμματιζόμενη Λογική

### ΣΥΝΤΗΡΗΣΗ ΕΓΚΑΤΑΣΤΑΣΕΩΝ ΑΥΤΟΜΑΤΙΣΜΟΥ & ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ

#### Θεματική Ενότητα 7 Προγραμματιζόμενη Λογική

##### 7.1 Εισαγωγή :Πλεονεκτήματα , βασικά χαρακτηριστικά

##### 7.2 Κύκλος λειτουργίας ενός PLC

##### 7.3 Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας

##### 7.4 Διευθυνσηοδότηση στην Step 7 και Ονοματολογία

##### 7.5 Μπλοκ προγραμματισμού στην Step 7

##### 7.6 Βασικές εντολές προγραμματισμού

##### 7.7 Βασικά Εργαλεία στην Step 7.

##### 7.8 Εντολές load και transfer

##### 7.9 Χρονικά

##### 7.10 Απαριθμητές

##### 7.11 Εντολές Ελέγχου της ροής του Προγράμματος

##### 7.12 Data block

##### 7.13 Παραμετροποιημένα Block

##### 7.14 Επεξεργασία Αναλογικών σημάτων

### ΠΡΟΣΔΟΚΩΜΕΝΑ ΜΑΘΗΣΙΑΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ (επιθυμητές Γνώσεις – Δεξιότητες – Ικανότητες)

- Να κατανοεί τον κύκλο λειτουργίας μιας C.P.U
- Να γνωρίζει τα βασικά λειτουργικά χαρακτηριστικά των δομικών στοιχείων ενός σταθμού της σειράς S – 7 300 .
- Να εντοπίζει κατηγορία βλάβης χρησιμοποιώντας τις ενδείξεις των LED μιας CPU .
- Να εντοπίζει την λογική διεύθυνση των δομικών στοιχείων ενός σταθμού της σειράς S – 7 300
- Να εντοπίζει την λογική διεύθυνση των σημάτων εισόδου – εξόδου ενός σταθμού της σειράς S – 7 300
- Να συναρμολογεί έναν σταθμό της σειράς S – 7 300
- Να μεταφέρει ένα project από την συσκευή προγραμματισμού σ έναν σταθμό της σειράς S – 7 300 ( Down load )
- Να μεταφέρει ένα project από έναν σταθμό της σειράς S – 7 300 στην συσκευή προγραμματισμού ( Up load ) .
- Να κατανοεί και να ερμηνεύει ένα απλό πρόγραμμα γραμμένο στην STL ή LAD γλώσσα προγραμματισμού .
- Να παρακολουθεί την εκτέλεση ενός προγράμματος μέσω της λειτουργίας Monitor στην STL και LAD γλώσσα προγραμματισμού.
- Να συνδέει On Line την συσκευή προγραμματισμού με έναν σταθμό της σειράς S – 7 300 .

# Περιεχόμενα

## Θεματικής Ενότητας 7: Προγραμματιζόμενη Λογική

### ΣΥΝΤΗΡΗΣΗ ΕΓΚΑΤΑΣΤΑΣΕΩΝ ΑΥΤΟΜΑΤΙΣΜΟΥ & ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ

#### Θεματική Ενότητα 7 Προγραμματιζόμενη Λογική

7.1 Εισαγωγή :Πλεονεκτήματα , βασικά χαρακτηριστικά

7.2 Κύκλος λειτουργίας ενός PLC

7.3 Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας

7.4 Διευθυνσηοδότηση στην Step 7 και Ονοματολογία

7.5 Μπλοκ προγραμματισμού στην Step 7

7.6 Βασικές εντολές προγραμματισμού

7.7 Βασικά Εργαλεία στην Step 7.

7.8 Εντολές load και transfer

7.9 Χρονικά

7.10 Απαραίτητες

7.11 Εντολές Ελέγχου της ροής του Προγράμματος

7.12 Data block

7.13 Παραμετροποιημένα Block

7.14 Επεξεργασία Αναλογικών σημάτων

### ΠΡΟΣΔΟΚΩΜΕΝΑ ΜΑΘΗΣΙΑΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ (επιθυμητές Γνώσεις – Δεξιότητες – Ικανότητες)

- Να κατανοεί την συμβολική διευθυνσηοδότηση και να αντιστοιχεί γενικά ( Global ) σύμβολα με διευθύνσεις .
- Να γνωρίζει τα Format στα οποία πρέπει να γράφονται οι βασικές μεταβλητές ενός προγράμματος στην Step 7 .
- Να γνωρίζει την χρήση των Data Block και την διευθυνσηοδότηση των μεταβλητών τους.
- Να αντιλαμβάνεται την έννοια των τοπικών μεταβλητών και να γνωρίζει τον τρόπο χρησιμοποίησής των .
- Να ερμηνεύει και να κατανοεί ένα παραμετροποιημένο μπλοκ .
- Να παρακολουθεί την εκτέλεση ενός παραμετροποιημένου μπλοκ σε συγκεκριμένη κλήση .
- Να αναγνωρίζει τις διευθύνσεις των αναλογικών σημάτων και να εντοπίζει πάνω στον σταθμό ένα αναλογικό σήμα .
- Να γνωρίζει την χρησιμοποιούμενη κλίμακα ενός αναλογικού σήματος μέσα σε ένα πρόγραμμα .

## **ΥΠΟΕΝΟΤΗΤΑ 7.1**

***Εισαγωγή: Πλεονεκτήματα, Βασικά χαρακτηριστικά***

## Εισαγωγή: Πλεονεκτήματα, Βασικά χαρακτηριστικά

---

Είναι ηλεκτρονικές συσκευές οι οποίες είναι σε θέση να πραγματοποιούν όλες τις λογικές συναρτήσεις, οι οποίες απαιτούνται για τον αυτοματισμό μίας μηχανής ή μίας βιομηχανικής διαδικασίας, ακολουθώντας ένα πρόγραμμα το οποίο αποθηκεύεται μέσα στην ίδια την συσκευή.

Η σημαντικότερη διαφορά στην περίπτωση χρησιμοποίησης των PLC είναι ότι τα κυκλώματα αυτοματισμού δεν πραγματοποιούνται με την λεγόμενη «**Συρματωμένη Λογική**» αλλά με το πρόγραμμα ή όπως αλλιώς λέγεται με την «**Προγραμματιζόμενη Λογική**».

# Τι είναι PLC (Programmable Logic Controllers)

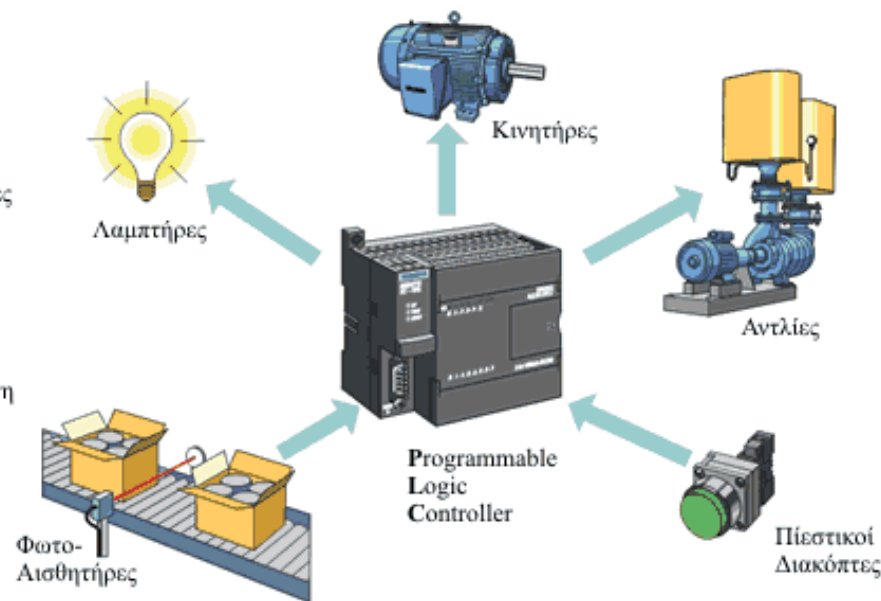
[Τα ΜΕΡΗ ΤΟΥ PLC](#)

[https://youtu.be/rFs0zwx\\_s9g](https://youtu.be/rFs0zwx_s9g)

1. Τα PLCs είναι **Προγραμματιζόμενοι Λογικοί Ελεγκτές** που ελέγχουν ένα οποιοδήποτε σύστημα το οποίο διαθέτει ηλεκτρικά σήματα ψηφιακού τύπου (ON / OFF) ή αναλογικού τύπου (π.χ. θερμοκρασία – στάθμη ,...).
2. Το PLC είναι σε θέση να αναγνωρίζει τις καταστάσεις που λαμβάνουν τα διάφορα αισθητήρια, να εκτελεί τις εντολές του προγράμματος και να μεταβιβάζει τα αποτελέσματα της επεξεργασίας εκεί όπου πρέπει να εκτελεστούν (π.χ μηχανές).

## PLCs (Programmable Logic Controllers)

Οι Προγραμματιζόμενοι Λογικοί Ελεγκτές, (PLC), ανήκουν στην οικογένεια των υπολογιστών. Χρησιμοποιούνται σε εμπορικές και βιομηχανικές εφαρμογές. Τα PLC's δρουν ως ελεγκτές μηχανών και διαδικασιών. Επιτηρούν τις εισόδους, λαμβάνουν αποφάσεις και ελέγχουν τις εξόδους τους με σκοπό την αυτοματοποίηση μηχανών και επεξεργασιών



<https://youtu.be/ReTtgzN-Dmc>

# Τι είναι PLC (Programmable Logic Controllers)

---

1. Τα PLCs είναι **Προγραμματιζόμενοι Λογικοί Ελεγκτές** που ελέγχουν ένα οποιοδήποτε σύστημα το οποίο διαθέτει ηλεκτρικά σήματα ψηφιακού τύπου (ON / OFF) ή αναλογικού τύπου (π.χ. θερμοκρασία – στάθμη ,..).

1. ΑΝΑΛΟΓΙΚΟΙ ΕΙΣΟΔΟΙ

<https://youtu.be/ZS3W7uBdQok>

1. ΨΗΦΙΑΚΟΙ ΕΙΣΟΔΟΙ

<https://youtu.be/qPEBjqW9cE8>



# Τι είναι PLC (Programmable Logic Controllers)

---

1. Τα πρώτα PLC εμφανίστηκαν στα τέλη της δεκαετίας του 60 στο χώρο της βιομηχανίας των αυτοκινήτων για να δώσουν λύση σε ένα μεγάλο πρόβλημα καθυστέρησης που είχαν μέχρι τότε. Το πρόβλημα ήταν ότι σε κάθε αλλαγή του παραγόμενου μοντέλου αυτοκινήτων όλοι οι ηλεκτρικοί πίνακες αυτοματισμού έπρεπε να ξηλωθούν και να ξαναγίνει η καλωδίωσή τους βάσει της νέας παραγωγικής διαδικασίας. Το μεγάλο μειονέκτημα των πρώτων PLC ήταν ότι απαιτούσαν προσωπικό με μεγάλη εξειδίκευση για να γίνονται κάθε φορά οι αλλαγές που ζητούσε η νέα παραγωγική διαδικασία. Με το πέρασμα των χρόνων τα PLC έγιναν πιο απλά, περισσότερο αξιόπιστα και πολύ παραγωγικά.
2. Από όσα αναφέραμε έως τώρα καταλαβαίνουμε ότι η σημαντικότερη διαφορά στην περίπτωση χρησιμοποίησης των PLC είναι **ότι τα κυκλώματα αυτοματισμού δεν πραγματοποιούνται με τη λεγόμενη «Συρματωμένη Λογική» αλλά με το πρόγραμμα ή όπως αλλιώς λέγεται με την «Προγραμματιζόμενη Λογική».**
3. Ιστορική αναδρομή <https://youtu.be/pPUnihpL6UI>

# Εισαγωγή: Πλεονεκτήματα, Βασικά χαρακτηριστικά

---

## **ΒΑΣΙΚΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ**

- **ΕΥΕΛΙΞΙΑ**
- **ΕΥΚΟΛΗ ΠΡΑΓΜΑΤΟΠΟΙΗΣΗ ΑΛΛΑΓΩΝ ΛΕΙΤΟΥΡΓΙΑΣ ΚΑΙ ΔΙΟΡΘΩΣΕΩΝ**
- **ΜΕΓΑΛΟΣ ΑΡΙΘΜΟΣ ΕΠΑΦΩΝ**
- **ΜΙΚΡΟ ΚΟΣΤΟΣ**
- **ΔΟΚΙΜΑΣΤΙΚΟΣ ΕΛΕΓΧΟΣ**
- **ΤΑΧΥΤΗΤΑ ΛΕΙΤΟΥΡΓΙΑΣ**
- **ΟΠΤΙΚΟΣ ΕΛΕΓΧΟΣ**
- **ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΚΥΚΛΩΜΑΤΩΝ**
- **ΕΓΚΑΤΑΣΤΑΣΗ ΣΕ ΠΕΔΙΑ ΙΣΧΥΟΣ**
- **ΑΠΟΘΗΚΕΥΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΤΟ PLC**

# Εισαγωγή: Πλεονεκτήματα, Βασικά χαρακτηριστικά

---

## **ΒΑΣΙΚΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ**

- ΔΥΝΑΤΟΤΗΤΑ ΣΥΝΔΕΣΗΣ ΜΕ ΟΘΟΝΕΣ , ΠΛΗΚΤΡΟΛΟΓΙΑ , ΕΚΤΥΠΩΤΕΣ
- ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ
- ΚΛΕΙΔΙ ΧΕΙΡΙΣΜΟΥ

# Εισαγωγή: Πλεονεκτήματα, Βασικά χαρακτηριστικά

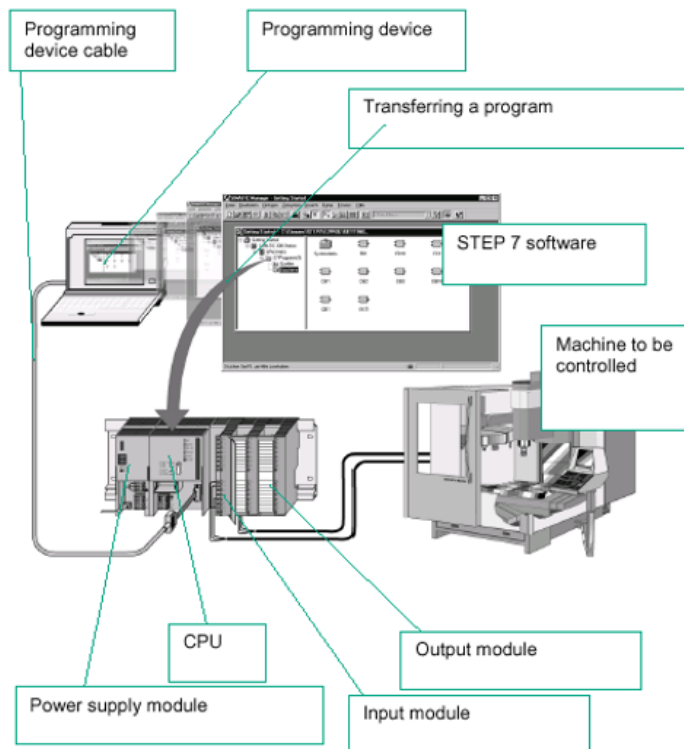
---

## ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΩΝ PLC

- *ΝΕΑ ΤΕΧΝΟΛΟΓΙΑ*
- *ΣΤΑΘΕΡΕΣ ΛΕΙΤΟΥΡΓΙΕΣ*
- *ΣΥΝΘΗΚΕΣ ΛΕΙΤΟΥΡΓΙΑΣ*

# Εισαγωγή: Πλεονεκτήματα, Βασικά χαρακτηριστικά

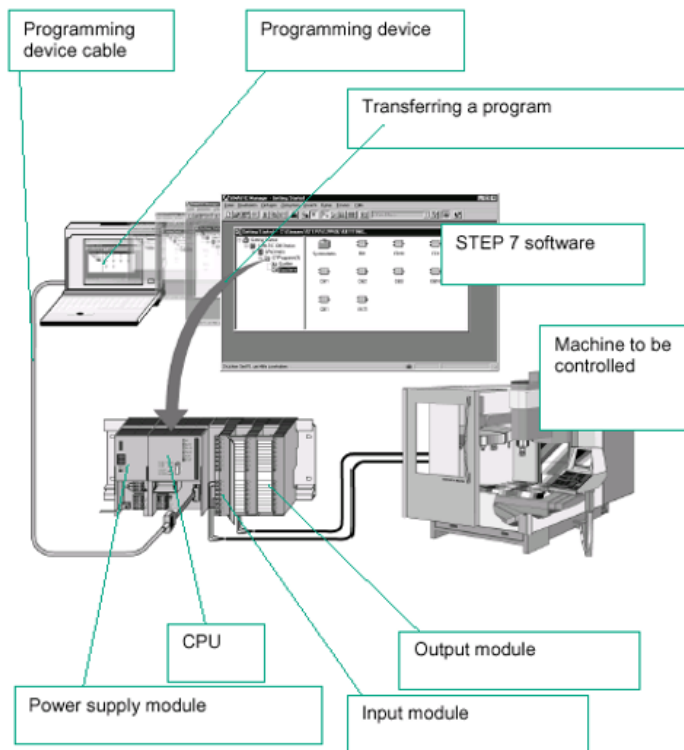
## ΑΠΑΡΑΙΤΗΤΟΣ ΕΞΟΠΛΙΣΜΟΣ



- ✓ *Συσκευή προγραμματισμού*
- ✓ *Πακέτο Software*
- ✓ *Τροφοδοτικό*
- ✓ *C.P.U : Μονάδα κεντρικής επεξεργασίας*
- ✓ *Κάρτες εισόδου*
- ✓ *Κάρτες εξόδου*

# Εισαγωγή: Πλεονεκτήματα, Βασικά χαρακτηριστικά

## ΑΠΑΡΑΙΤΗΤΟΣ ΕΞΟΠΛΙΣΜΟΣ



**ΣΥΣΚΕΥΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ:** Είναι το μέσον με το οποίο ο άνθρωπος επικοινωνεί με το PLC.

**ΠΑΚΕΤΟ SOFTWARE:** Το πακέτο software που περιέχει τα διάφορα «εργαλεία» όπως επίσης και τις διάφορες εντολές προγραμματισμού μπορούμε να το παρομοιάσουμε με την κοινή γλώσσα που χρησιμοποιούν ο άνθρωπος και η συσκευή προγραμματισμού ώστε να καταλαβαίνονται μεταξύ τους.

**ΤΡΟΦΟΔΟΤΙΚΟ:** Ο ρόλος του είναι να μετατρέπει την τάση τροφοδοσίας του δικτύου στην αναγκαία τάση τροφοδοσίας του PLC.

**CPU ( Central Processing Unit ):** Είναι ο εγκέφαλος του συστήματος, εδώ περιέχεται και εκτελείται τόσο το λειτουργικό πρόγραμμα του PLC όσο και το πρόγραμμα του χρήστη.

**ΚΑΡΤΕΣ ΕΙΣΟΔΟΥ ( DI / AI ):** Είτε ψηφιακές, είτε αναλογικές, αυτές έχουν τον ρόλο να μετατρέπουν τα σήματα της εγκατάστασης σε σήματα τα οποία μπορεί να επεξεργαστεί η CPU.

**ΚΑΡΤΕΣ ΕΞΟΔΟΥ ( DO / AO ):** Είτε ψηφιακές, είτε αναλογικές, αυτές έχουν τον ρόλο να μετατρέπουν τις αποφάσεις της CPU σε κατάλληλες τάσεις τις οποίες στέλνουν προς την εγκατάσταση.



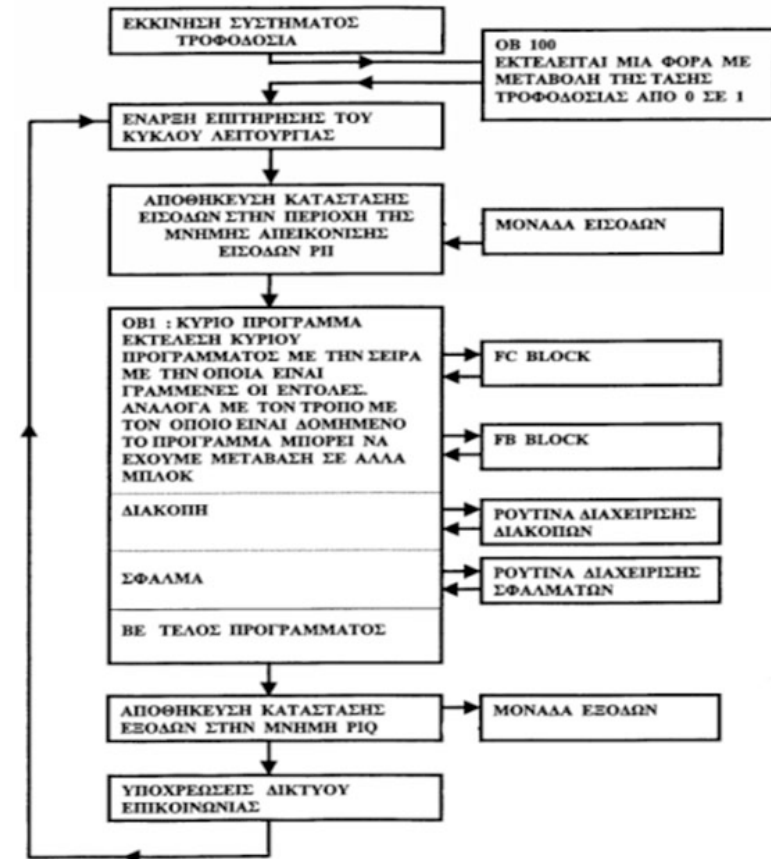
**ΥΠΟΕΝΟΤΗΤΑ 7.2**  
***Κύκλος Λειτουργίας ενός PLC***

# Εισαγωγή: Πλεονεκτήματα, Βασικά χαρακτηριστικά

## ΚΥΚΛΟΣ ΛΕΙΤΟΥΡΓΙΑΣ - SCAN TIME

Ένα πρόγραμμα το οποίο είναι αποθηκευμένο στην μνήμη ενός PLC εκτελείται με κυκλικό τρόπο. Αυτός ο κυκλικός τρόπος εκτέλεσης του προγράμματος ονομάζεται « Κύκλος Λειτουργίας » ή « Scan Time ».

- *Διάβασμα κατάστασης των εισόδων*
- *Εκτελεί μία - μία τις εντολές του προγράμματος και ταυτόχρονα ενημερώνει τις περιοχές μνήμης στις οποίες αναφέρονται όλες οι εντολές γραφής ,που περιέχει του πρόγραμμα*
- *Από την περιοχή μνήμης των εξόδων ενημερώνει τις κάρτες εξόδου*
- *Εκτελεί τυχόν υποχρεώσεις από το πρωτόκολλο επικοινωνίας*





# Εισαγωγή: Πλεονεκτήματα, Βασικά χαρακτηριστικά

## ΚΥΚΛΟΣ ΛΕΙΤΟΥΡΓΙΑΣ - SCAN TIME

*Η πληροφορία για την κατάσταση των εισόδων αποκτάται μόνο στην αρχή του κύκλου και κατά την διάρκειά εκτέλεσης του η πληροφορία αυτή θεωρείται σταθερή*

*Η εκτέλεση μιας εντολής και η ενημέρωση της αντίστοιχης θέσης μνήμης γίνεται με την σειρά με την οποία είναι γραμμένη η εντολή στο πρόγραμμα*

Ένα πρόγραμμα το οποίο είναι αποθηκευμένο στην μνήμη ενός PLC εκτελείται με κυκλικό τρόπο. Αυτός ο κυκλικός τρόπος εκτέλεσης του προγράμματος ονομάζεται « Κύκλος Λειτουργίας » ή « Scan Time ».

Με το που ξεκινάει ένας κύκλος λειτουργίας αυτόματα ενεργοποιείται ένα κύκλωμα χρονοεπιτήρησης (Watch Dog ). Εάν ο κύκλος λειτουργίας εκτελεστεί στα όρια του μέγιστου χρόνου που εμείς αποδεχόμαστε ( ρυθμίζεται ) τότε ξεκινάει ο επόμενος κύκλος λειτουργίας, εάν όχι τότε η CPU μπαίνει σε κατάσταση STOP και έχουμε ένδειξη σφάλματος.

Ταυτόχρονα, με την εκτέλεση του προγράμματος ενημερώνονται οι περιοχές μνήμης των Merker ( M ) , Timer ( T ) , Counter ( C ) , Output ( Q ) ,.....

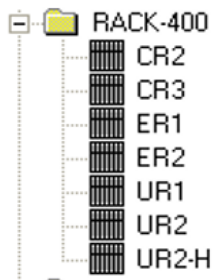
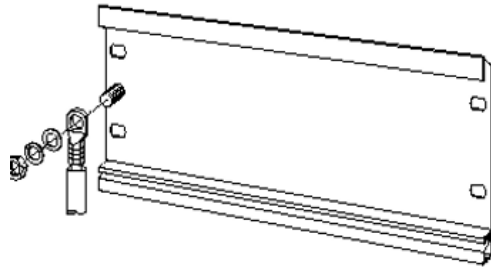
## **ΥΠΟΕΝΟΤΗΤΑ 7.3**

***Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας***

# Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας

## ΠΛΑΙΣΙΟ ΣΤΗΡΙΞΗΣ - RACK

Στην σειρά S7-300 των PLC ο ρόλος του πλαισίου στήριξης είναι απλά η στερέωση ( φυσική στήριξη ) των καρτών που αποτελούν τον σταθμό μας



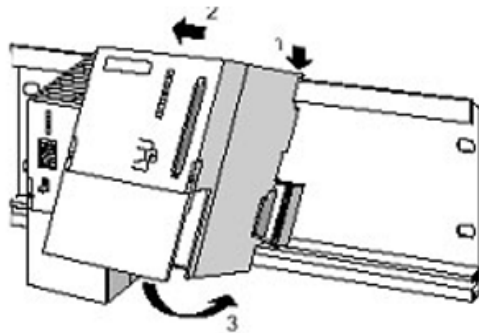
Για την σειρά S7-400 όπου ο ρόλος του πλαισίου στήριξης δεν περιορίζεται μόνο στην στερέωση των καρτών του σταθμού , αλλά περιέχει και τα bus επικοινωνίας

[ΑΝΑΛΥΤΙΚΑ ΤΑ ΜΕΡΗ](#)

<https://youtu.be/E2WNPXJf-Kw>

# Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας

## ΠΛΑΙΣΙΟ ΣΤΗΡΙΞΗΣ - RACK



- ✓ Πάνω στα Rack έχουμε τις λεγόμενες slot θέσεις (φυσικές θέσεις τοποθέτησης καρτών). Κάθε Rack έχει 11 slot θέσεις, από την slot θέση 1 έως και την slot θέση 11.
- ✓ Όταν στηρίζουμε τις διάφορες κάρτες πάνω σ ένα Rack πρέπει να τηρήσουμε τους εξής κανόνες.

- ✓ Slot θέση 1 : Στηρίζουμε πάντοτε το τροφοδοτικό
- ✓ Slot θέση 2 : Στηρίζουμε πάντοτε την CPU
- ✓ Slot θέση 3 : Στηρίζουμε πάντοτε την κάρτα διασύνδεσης από Rack σε Rack
- ✓ Slot θέση 4 : Από αυτήν την θέση και πέρα δομούμε τον σταθμό μας ανάλογα με τις προσωπικές μας ανάγκες

# Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας

---

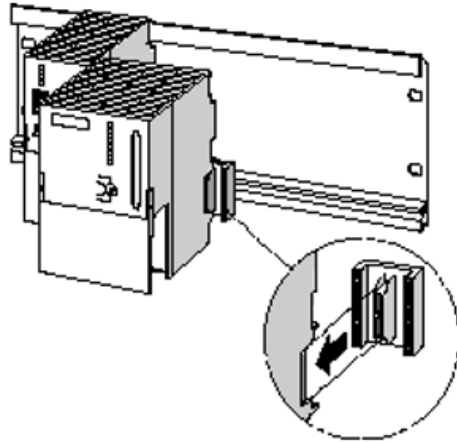
## ΚΑΡΤΑ ΔΙΑΣΥΝΔΕΣΗΣ

*Οι κάρτες διασύνδεσης, οι οποίες είναι οι λεγόμενες IM κάρτες χρησιμεύουν για την επικοινωνία της CPU με τις κάρτες του σταθμού οι οποίες βρίσκονται στα λεγόμενα Rack επέκτασης*

- ✓ **IM 365**
  - ✓ **IM 365 S**
  - ✓ **IM 365 R**
- ✓ **IM 360**
- ✓ **IM 361**

# Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας

## BUS ΕΠΙΚΟΙΝΩΝΙΑΣ



✓ **P – BUS : PERIPHERAL BUS :**

Στο P – Bus η πληροφορία μεταφέρεται  
σειριακά με ταχύτητα 1,5 MBPS

✓ **K – BUS : COMMUNICATION BUS :**

Και στο K – Bus η πληροφορία μεταφέρεται σειριακά με ταχύτητα  
187,5 KBPS .

Προσοχή στην σειρά S7-300 το K – Bus είναι προέκταση του MPI  
δικτύου επικοινωνίας .

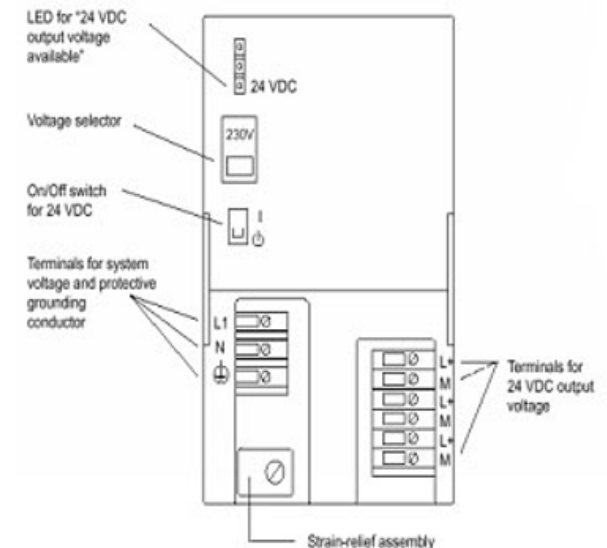
Όλες οι «έξυπνες» κάρτες παίρνουν MPI διεύθυνση

# Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας

## ΤΡΟΦΟΔΟΤΙΚΟ - PS : POWER SUPPLY

*Ο ρόλος του είναι να δημιουργήσει τις αναγκαίες τάσεις τροφοδοσίας που χρειάζεται ο σταθμός μας για την τροφοδοσία του.*

- *Κλέμες για την τάση τροφοδοσίας ( L1 , N ) και για τον αγωγό προστασίας*
- *Κλέμες για την DC τάση εξόδου στα 24 V ( L+ , M )*
- *Διακόπτη ON / OFF για τα 24 V DC*
- *Επιλογή τάσης τροφοδοσίας ( 230 VAC ή 120 VAC )*
- *Ενδεικτικό LED ύπαρξης τάσης εξόδου στα 24 VDC*



# Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας

---

## ΚΕΝΤΡΙΚΗ ΜΟΝΑΔΑ ΕΠΕΞΕΡΓΑΣΙΑΣ - CPU

- *Διάβασμα , Ερμηνεία και Εκτέλεση , με την σωστή διαδοχή των οδηγιών που περιέχονται στην μνήμη*
- *Έλεγχο του πρωτοκόλλου επικοινωνίας που έχουμε καθορίσει στο σύστημά μας*
- *Αποθήκευση των πληροφοριών που θέλουμε*
- *Εκτέλεση αριθμητικών πράξεων*

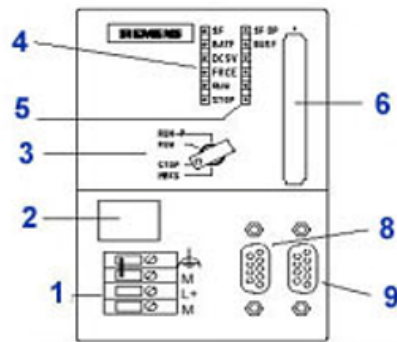
**ΤΟΝ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΗ**

**ΤΗΝ ΜΝΗΜΗ**



# Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας

## ΕΞΩΤΕΡΙΚΑ ΜΙΑ CPU ΠΑΡΟΥΣΙΑΖΕΙ



1: ΑΚΡΟΔΕΚΤΕΣ ΤΡΟΦΟΔΟΣΙΑΣ

2: ΘΕΣΗ ΓΙΑ ΜΠΑΤΑΡΙΑ

3: ΔΙΑΚΟΠΤΗ - ΚΛΕΙΔΙ

- ✓ ΘΕΣΗ RUN - P
- ✓ ΘΕΣΗ RUN
- ✓ ΘΕΣΗ STOP
- ✓ ΘΕΣΗ MRES

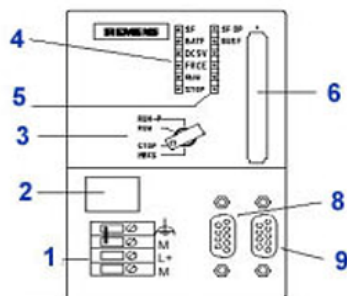


4: ΕΝΔΕΙΚΤΙΚΑ LED ΓΙΑ ΤΗΝ ΚΑΤΑΣΤΑΣΗ ΤΗΣ CPU

- ⊖ SF ... (red) ... hardware or software faults
- ⊖ BATF ... (red) ... battery fault
- ⊖ DC5V ... (green) ... 5 V DC supply for CPU and S7-300 bus is ok.
- ⊖ FRCE ... (yellow) ... force request is active
- ⊖ RUN ... (green) ... CPU in RUN; LED flashes at start-up with 1 Hz; in HALT mode with 0.5 Hz
- ⊖ STOP ... (yellow) ... CPU in STOP or HALT or start-up; LED flashes at memory reset request

# Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας

## ΕΞΩΤΕΡΙΚΑ ΜΙΑ CPU ΠΑΡΟΥΣΙΑΖΕΙ



### 3:ΔΙΑΚΟΠΤΗ - ΚΛΕΙΔΙ

- ✓ ΘΕΣΗ RUN-P
- ✓ ΘΕΣΗ RUN
- ✓ ΘΕΣΗ STOP
- ✓ ΘΕΣΗ MRES



### ΔΙΑΚΟΠΤΗΣ RUN – P / RUN / STOP / MRES

#### ΘΕΣΗ RUN – P

Σε αυτή την θέση του διακόπτη η CPU εκτελεί το πρόγραμμα που περιέχεται στην μνήμη της, έχουμε τη δυνατότητα πρόσβασης στο υπάρχον πρόγραμμα είτε για απλή παρακολούθηση είτε για να κάνουμε μεταβολές σ αυτό.

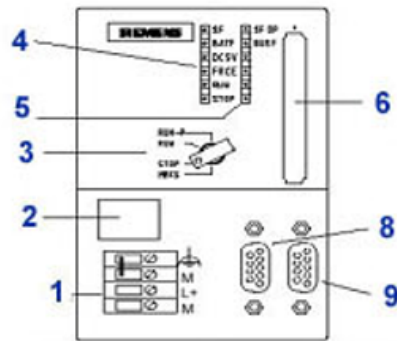
ΘΕΣΗ RUN: Σε αυτή τη θέση του διακόπτη η CPU εκτελεί το υπάρχον πρόγραμμα, δεν μπορούμε να κάνουμε αλλαγές σε αυτό.

ΘΕΣΗ STOP: Σε αυτή τη θέση η CPU δεν εκτελεί το υπάρχον πρόγραμμα, η πρόσβαση σε αυτό είτε για διάβασμα, είτε για μεταβολές εξαρτάται από την ύπαρξη ή μη του αντίστοιχου κλειδώματος μέσω του software.

ΘΕΣΗ MRES: Η θέση αυτή είναι με αυτόματη επαναφορά (λειτουργεί σαν μπουτόν) και χρησιμεύει για ολική διαγραφή της RAM μνήμης της CPU.

# Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας

## ΕΞΩΤΕΡΙΚΑ ΜΙΑ CPU ΠΑΡΟΥΣΙΑΖΕΙ



*5: ΕΝΔΕΙΚΤΙΚΑ LED ΓΙΑ ΤΗΝ ΚΑΤΑΣΤΑΣΗ ΤΟΥ PROFIBUS ΔΙΚΤΥΟΥ*

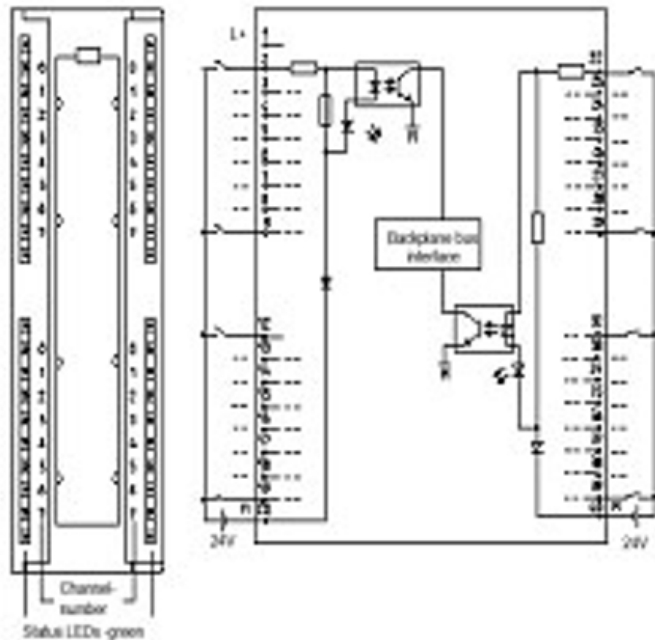
*6: ΘΕΣΗ ΓΙΑ ΤΟΠΟΘΕΤΗΣΗ ΕΞΩΤΕΡΙΚΗΣ ΜΝΗΜΗΣ*

*8: MPI INTERFACE*

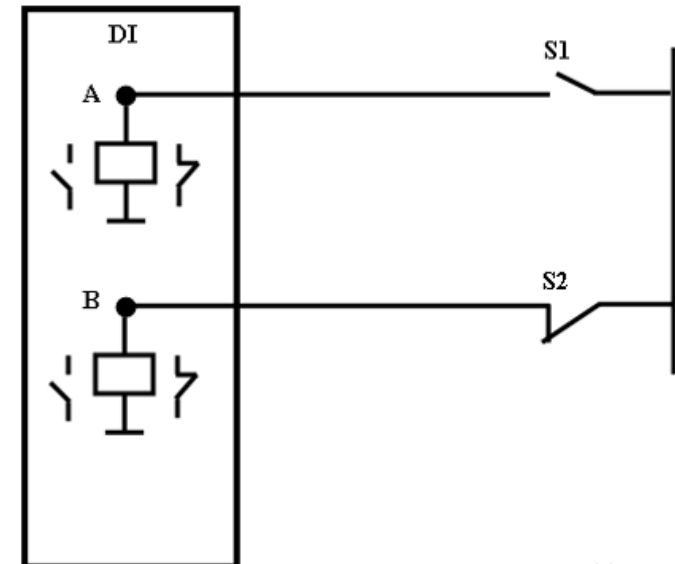
*9: PROFIBUS DP INTERFACE*

# Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας

## ΨΗΦΙΑΚΕΣ ΜΟΝΑΔΕΣ ΕΙΣΟΔΩΝ DI ( Digital Input )

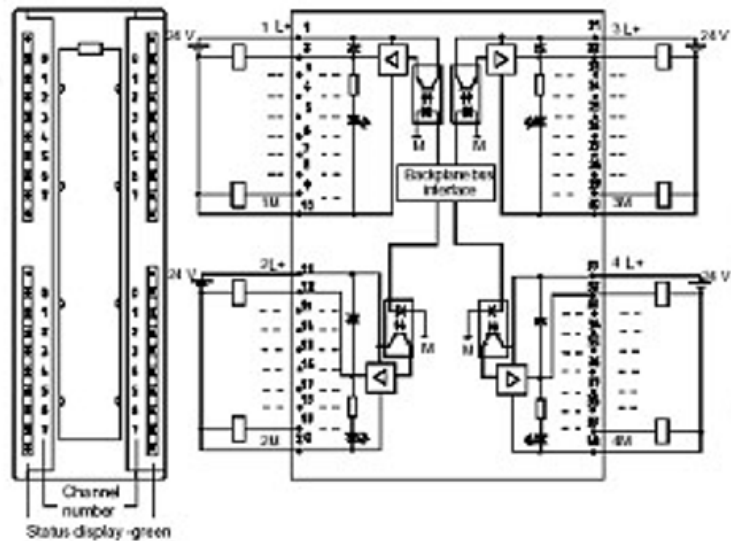


Η χρησιμοποίηση των ψηφιακών μονάδων εισόδου έχει σαν σκοπό να μεταφέρει στην CPU τις καταστάσεις των διαφόρων ψηφιακών αισθητηρίων ή διακοπών ελέγχου που διαθέτουμε στην εγκατάσταση



# Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας

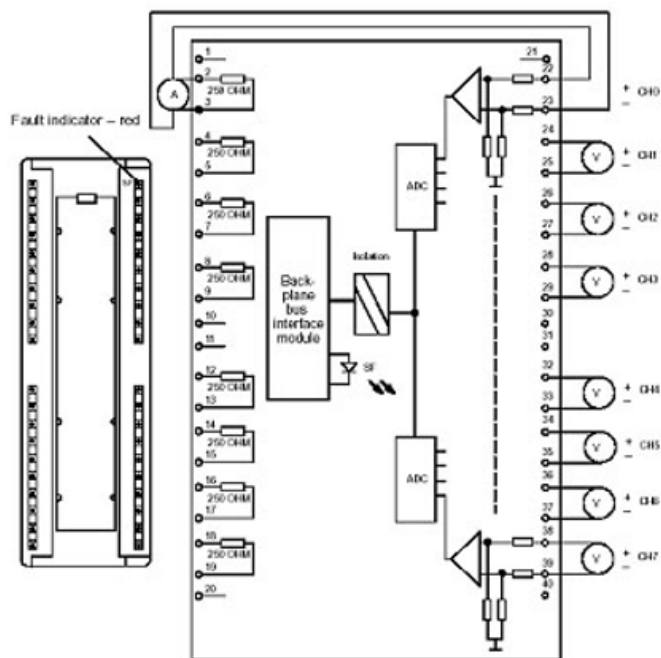
## ΨΗΦΙΑΚΕΣ ΜΟΝΑΔΕΣ ΕΞΟΔΩΝ DO (Digital Output )



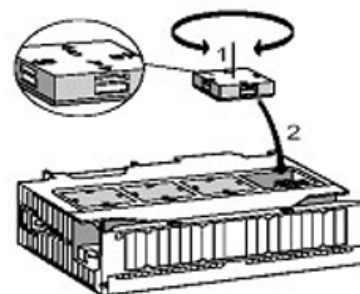
*Ο ρόλος τους είναι να μετατρέπουν τις αποφάσεις που παίρνει η CPU σε εντολές προς την εγκατάσταση*

# Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας

## ΑΝΑΛΟΓΙΚΕΣ ΜΟΝΑΔΕΣ ΕΙΣΟΔΩΝ AI ( Analog Input )



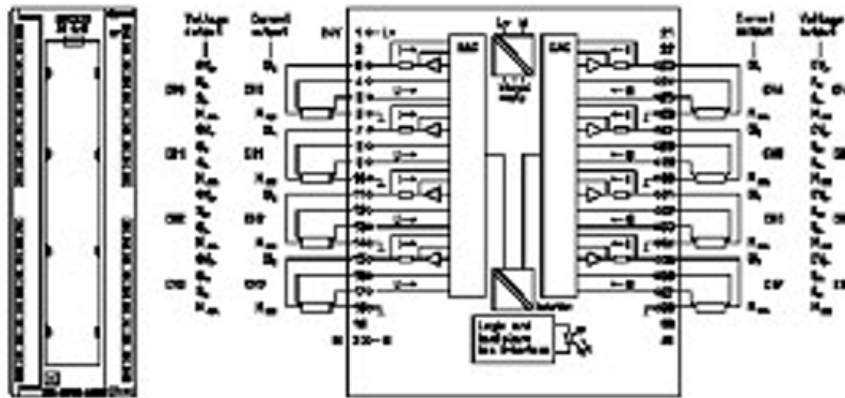
Οι μονάδες αναλογικών σημάτων έχουν τον ρόλο να διαβάζουν ένα αναλογικό μέγεθος να το μετατρέπουν σε έναν ψηφιακό αριθμό ( δυαδική αναπαράσταση ) και να το αποθηκεύουν σε μία δική τους περιοχή μνήμης . Από εκεί πλέον η CPU έχει την δυνατότητα να το διαβάσει και να το επεξεργάζεται



Θα πρέπει να έχουμε υπ όψη μας ότι η ρύθμιση που γίνεται με τους μικροδιακόπτες αφορά έναν συγκεκριμένο αριθμό καναλιών ( για κάθε έναν μικροδιακόπτη αντιστοιχούν δύο κανάλια ).

# Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας

## ΑΝΑΛΟΓΙΚΕΣ ΜΟΝΑΔΕΣ ΕΞΟΔΩΝ ΑΟ ( Analog Output )



Οι αναλογικές μονάδες εξόδου έχουν τον ρόλο να μετατρέπουν το αριθμητικό μέγεθος με το οποίο "σκέπτεται" η CPU στην κατάλληλη τιμή τάσης ή έντασης, ώστε να μπορεί να οδηγηθεί το ανάλογο εξάρτημα που ελέγχει το φυσικό μέγεθος της εγκατάστασής μας

# Δομή ενός σταθμού και βασικά χαρακτηριστικά λειτουργίας

---

## ΜΟΝΑΔΕΣ ΕΙΔΙΚΩΝ ΛΕΙΤΟΥΡΓΙΩΝ FM ( Function Module )

*Μία ειδική κάρτα , όπως δηλώνει και η ονομασία της , αναλαμβάνει μία συγκεκριμένη λειτουργία αυτοματισμού . Έτσι μια λειτουργία που είναι επαναλαμβανόμενη και συνηθισμένη ( επομένως τυποποιημένη ) μπορεί να υλοποιηθεί ( τόσο σε Hardware όσο και σε Software ) από μία ειδική κάρτα.*

Οι μονάδες ειδικών λειτουργιών έχουν τις δικές τους εισόδους – εξόδους ( συνήθως με ειδικά χαρακτηριστικά), τον δικό τους επεξεργαστή και συγκεκριμένη εφαρμογή.

<https://youtu.be/6HxtpwR9zCQ>



## **ΥΠΟΕΝΟΤΗΤΑ 7.4**

***Διευθυνσηοδότηση στην STEP 7 και Ονοματολογία***

# Διευθυνσιοδότηση στην STEP 7 και Ονοματολογία

## ΔΙΕΥΘΥΝΣΗ ΘΕΣΗΣ : SLOT – ADDRESS

Κάθε μονάδα η οποία ανήκει σ έναν σταθμό με PLC της σειράς S7-300 λαμβάνει μια διεύθυνση θέσης

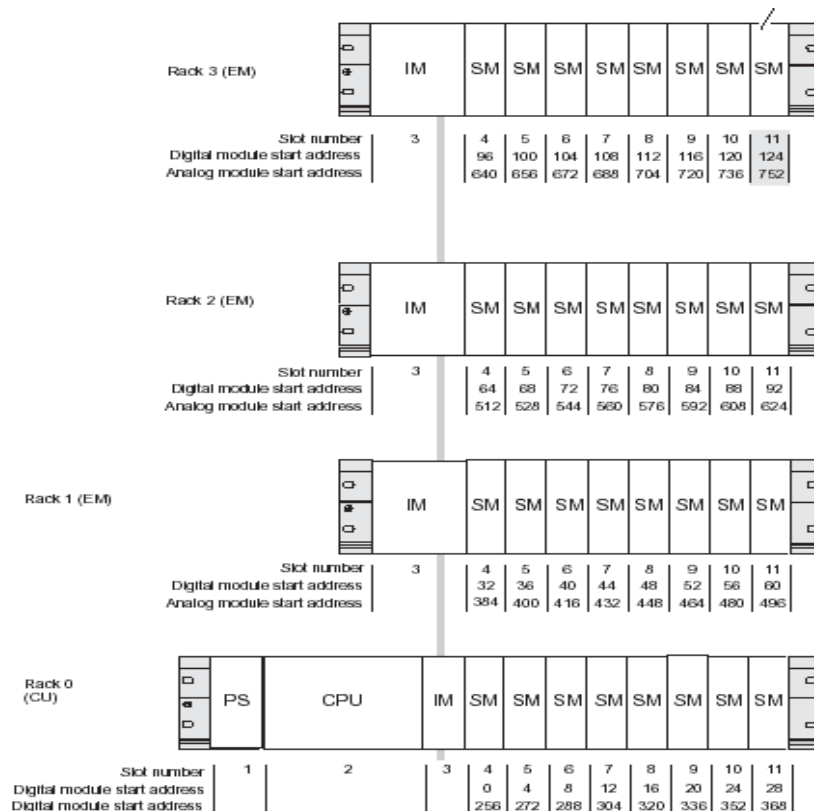
**CU : Central Unit . Το rack πάνω στο οποίο βρίσκεται η CPU το αναφέρουμε σαν κεντρική μονάδα .**  
**EM : Expansion Module .**  
 Χρησιμοποιούμε rack επέκτασης, εάν οι slot θέσεις του κεντρικού rack δεν επαρκούν για την εφαρμογή μας .



Κάθε μονάδα η οποία ανήκει σ έναν σταθμό με PLC της σειράς S7-300 λαμβάνει μια διεύθυνση θέσης. Αυτή αποτελείται από τον αριθμό του Rack που είναι τοποθετημένη η μονάδα και τον αύξων αριθμό θέσης της πάνω στο Rack . Κάθε Rack έχει 11 θέσεις, έτσι σε ένα πλήρες σύστημα (βασικό Rack + 3 Rack επέκτασης) με PLC της σειράς S7-300, οι διευθύνσεις που μπορούμε να έχουμε για τις διάφορες μονάδες που το απαρτίζουν δίνονται στην κάτω εικόνα.

# Διευθυνσιοδότηση στην STEP 7 και Ονοματολογία

## ΛΟΓΙΚΕΣ ΔΙΕΥΘΥΝΣΕΙΣ ΤΩΝ ΜΟΝΑΔΩΝ ΠΟΥ ΑΠΟΤΕΛΟΥΝ ΕΝΑΝ ΣΤΑΘΜΟ



- ✓ Αφήνουμε την CPU να δώσει τις λογικές διευθύνσεις ( Διευθυνσιοδότηση Συστήματος ) των μονάδων ( System Selection )
- ✓ Επιλέγουμε εμείς τις λογικές διευθύνσεις των μονάδων

<https://youtu.be/YVf8Z0kW48w>

# Διευθυνσιοδότηση στην STEP 7 και Ονοματολογία

## ΕΝΝΟΙΕΣ : BIT- BYTE – WORD - DWORD

➤ *BIT*      ■

➤ *BYTE*      ■■■■■■■■

➤ *WORD*      ■■■■■■■■ ■■■■■■■■

➤ *DWORD*      ■■■■■■■■ ■■■■■■■■ ■■■■■■■■ ■■■■■■■■

<https://youtu.be/3cpvOTultF4>

**BIT** : Το bit είναι η μικρότερη μονάδα αποθήκευσης της κατάστασης ενός ψηφιακού σήματος. Το bit είναι ο χώρος μιας κυψέλης μνήμης και μπορεί να πάρει δύο τιμές (κατάσταση `` 0 `` =0V και κατάσταση `` 1 ``).

**BYTE** : Μία ομάδα από 8 συνεχόμενα bit ορίζουν ένα byte. Οι διευθύνσεις των bit που αποτελούν ένα byte ξεκινώντας από δεξιά είναι από το «0» έως το «7»

**WORD** : Δύο συνεχόμενα byte ή 16 συνεχόμενα bit ορίζουν μία word. Οι διευθύνσεις των bit που αποτελούν μία word ξεκινώντας από δεξιά είναι από το «0» έως το «15».

**DWORD** : Δύο συνεχόμενες word ή 4 συνεχόμενα byte ή 32 συνεχόμενα bit ορίζουν μία Dword.

# Διευθυνσιοδότηση στην STEP 7 και Ονοματολογία

## ΟΡΓΑΝΩΣΗ ΠΕΡΙΟΧΗΣ ΜΝΗΜΗΣ

ΔΙΕΥΘΥΝΣΕΙΣ ΒΙΤ							
7	6	5	4	3	2	1	0

	A						
			B				
					C		

0	Δ Ι Ε Υ Θ Υ Ν Σ Ε Ι Σ  B Y T E
1	
2	
3	
4	
5	
6	
7	
..	
..	
N	

ΔΙΕΥΘΥΝΣΕΙΣ ΒΙΤ							
7	6	5	4	3	2	1	0


0	Δ Ι Ε Υ Θ Υ Ν Σ Ε Ι Σ  B Y T E
1	
2	
3	
4	
5	
6	
7	
8	
9	
..	
N	

## Διευθυνσιοδότηση στην STEP 7 και Ονοματολογία

### ΟΝΟΜΑΤΟΛΟΓΙΑ ΕΙΣΟΔΩΝ – ΕΞΟΔΩΝ ΑΠΌ ΡΙΙ ΚΑΙ ΡΙΩ

I x.y	Πληροφορία μεγέθους bit
IB x	Πληροφορία μεγέθους byte
IW x	Πληροφορία μεγέθους word
ID x	Πληροφορία μεγέθους Dword

Q x.y	Πληροφορία μεγέθους bit
QB x	Πληροφορία μεγέθους byte
QW x	Πληροφορία μεγέθους word
QD x	Πληροφορία μεγέθους Dword

### ΟΝΟΜΑΤΟΛΟΓΙΑ MERKER - LOCAL

<b>M x.y</b>
<b>MB x</b>
<b>MW x</b>
<b>MD x</b>

<b>L x.y</b>
<b>LB x</b>
<b>LW x</b>
<b>LD x</b>

## Διευθυνσιοδότηση στην STEP 7 και Ονοματολογία

### ΟΝΟΜΑΤΟΛΟΓΙΑ: ΧΡΟΝΙΚΩΝ – ΑΠΑΡΙΘΜΗΤΩΝ – DATA BLOCK

- ΧΡΟΝΙΚΑ : Tx
- ΑΠΑΡΙΘΜΗΤΕΣ : Cx
- ΔΕΔΟΜΕΝΑ ΑΠΟ DATA BLOCK

DBXy.z	DIYy.z	Για πληροφορία μεγέθους ενός bit
DBBy	DIBy	Για πληροφορία μεγέθους ενός byte
DBWy	DIWy	Για πληροφορία μεγέθους μιας word
DBDy	DIDy	Για πληροφορία μεγέθους μιας Dword

#### ΧΡΟΝΙΚΑ - TIMERS

Τα χρονικά χρησιμοποιούνται για να υλοποιήσουν αλγορίθμους που έχουν σχέση με χρόνο (Επιτήρηση, Αναμονή, Μέτρηση Χρονικού Διαστήματος, Δημιουργία Παλμών,..). Με τον όρο «χρονικό» εννοούμε μια πληροφορία η οποία **καταλαμβάνει χώρο μιας word** στην ξεχωριστή περιοχή μνήμης των χρονικών. Για να έχουμε πρόσβαση σε αυτήν την περιοχή μνήμης των χρονικών χρησιμοποιούμε το γράμμα T.

#### ΑΠΑΡΙΘΜΗΤΕΣ - COUNTERS

Ο απαριθμητής μας δίνει την δυνατότητα να εκτελούμε εργασίες απαρίθμησης και η πληροφορία να καταχωρείται σε μία μεταβλητή, η οποία έχει μέγεθος μιας word. Η περιοχή μνήμης των απαριθμητών είναι ξεχωριστή και για να έχουμε πρόσβαση σε αυτήν χρησιμοποιούμε το γράμμα C.

#### ΜΕΤΑΒΛΗΤΕΣ ΣΤΗΝ ΠΕΡΙΟΧΗ ΜΝΗΜΗΣ ΤΩΝ DATA BLOCK

Τα Data Block είναι περιοχές μνήμης στις οποίες μπορούμε να γράψουμε ή να διαβάσουμε διάφορες πληροφορίες.

## *Εκπαιδευτική Δραστηριότητα Δ3: «Άσκηση»*

---

***Τίτλος: Λογικές Διευθύνσεις Εισόδων - Εξόδων***



## **ΥΠΟΕΝΟΤΗΤΑ 7.5**

***Μπλοκ Προγραμματισμού στην STEP 7***

## Μπλοκ Προγραμματισμού στην STEP 7

---

### ΜΠΛΟΚ ΟΡΓΑΝΩΣΗΣ [ ORGANIZATION BLOCKS ] OB

*Τα μπλοκ οργάνωσης έχουν τον ρόλο του διαμεσολαβητή μεταξύ του λειτουργικού συστήματος και του προγράμματος εφαρμογής . Αυτά καλούνται μόνο από το λειτουργικό σύστημα και ελέγχουν :*

- *Την κυκλική επεξεργασία του προγράμματος εν σχέση με την παρουσία ή όχι κάποιου συμβάντος ( Interrupt )*
- *Την συμπεριφορά του αυτοματισμού του συστήματος κατά την εκκίνηση*
- *Την διαχείριση των παρουσιαζόμενων σφαλμάτων*

Μέσα σ ένα μπλοκ οργάνωσης γράφουμε κώδικα, ο οποίος εκτελείται όταν το συγκεκριμένο OB κληθεί από την CPU. Τα OB χωρίζονται σε κατηγορίες ανάλογα με την αιτία ενεργοποίησης των και έχουν διαφορετικούς βαθμούς προτεραιότητας (αναλυτικότερα για αυτό το θέμα θα μιλήσουμε σε αντίστοιχο κεφάλαιο).

## Μπλοκ Προγραμματισμού στην STEP 7

---

### ΣΥΝΑΡΤΗΣΕΙΣ [ FUNCTIONS ] FC

Οι Συναρτήσεις ( Function ) είναι μπλοκ μέσα στα οποία ο χρήστης γράφει κώδικα .

Τα FC χρησιμοποιούνται για :

- ✓ Υπολογισμός κάποιας συνάρτησης και απόδοση τιμής στο μπλοκ που το έχει καλέσει ( πχ υπολογισμός μαθηματικών συναρτήσεων )
- ✓ Έλεγχος μιας τεχνολογικής συνάρτησης ( πχ έλεγχος ανεξάρτητου τμήματος εγκατάστασης )

Βασικές ιδιότητες των FC είναι ότι «παραμετροποιούνται» και ότι «στερούνται μνήμης» .

Οι Μέσα σ ένα Function (FC) περιέχεται λοιπόν ένα κομμάτι προγράμματος το οποίο εκτελείται κάθε φορά που το συγκεκριμένο FC θα κληθεί από τον χρήστη (μέσω του κώδικα που έγραψε) μέσα από ένα άλλο μπλοκ προγράμματος (π.χ. OB , FC , FB).

## Μπλοκ Προγραμματισμού στην STEP 7

---

### ΜΠΛΟΚ ΣΥΝΑΡΤΗΣΕΩΝ [ FUNCTION BLOCK ] FB

*Τα Function Blocks ( FB ) είναι και αυτά μπλοκ μέσα στα οποία ο χρήστης γράφει κώδικα .*

*Οι βασικές ιδιότητες των FB είναι ότι « παραμετροποιούνται » και ότι έχουν στην διάθεσή τους μια « ειδική περιοχή μνήμης » ( Instance Data Block ) , μέσα στην οποία αποθηκεύονται αυτόματα οι τιμές που λαμβάνουν οι « Παράμετροι » και οι « Στατικές Μεταβλητές » του FB .*

## Μπλοκ Προγραμματισμού στην STEP 7

---

### ΜΠΛΟΚ ΔΕΔΟΜΕΝΩΝ [ DATA BLOCK ] DB

*Τα Μπλοκ Δεδομένων DB δεν περιέχουν κώδικα , αυτά τα χρησιμοποιούμε σαν περιοχές μνήμης για την αποθήκευση των πληροφοριών που εμείς θέλουμε και καταλαμβάνουν χώρο στην περιοχή της μνήμης εργασίας ( Work Memory ) των CPU .*

*Τα Data Block χωρίζονται στις εξής κατηγορίες :*

- **GLOBAL DATA BLOCK**
- **ΕΞΑΡΤΗΜΕΝΑ DATA BLOCK**
  - ✓ **INSTANCE DB**
  - ✓ **USER DEFINE TYPE DB**

# Μπλοκ Προγραμματισμού στην STEP 7

---

## ΜΠΛΟΚ ΣΥΣΤΗΜΑΤΟΣ & ΣΤΑΝΤΑΡ ΜΠΛΟΚ

### ΜΠΛΟΚ ΣΥΣΤΗΜΑΤΟΣ

- ✓ *SFC [ Function του συστήματος ]*
- ✓ *SFB [ Function Block του συστήματος ]*
- ✓ *SDB [ Data Block του συστήματος ]*

### ΣΤΑΝΤΑΡ ΜΠΛΟΚ

*Αυτά είναι έτοιμα μπλοκ που περιέχουν κώδικα για συνηθισμένες εφαρμογές αυτοματισμού , βρίσκονται στο πακέτο της βιβλιοθήκης "Standard Library" το οποίο συνοδεύει το Software Simatic Manager*

## **ΥΠΟΕΝΟΤΗΤΑ 7.6**

### ***Βασικές Εντολές Προγραμματισμού***

# Βασικές Εντολές Προγραμματισμού

## ΠΩΣ ΑΝΟΙΓΟΥΜΕ ΤΟ OB1

The image displays three screenshots from the Siemens Step 7 software interface, illustrating the process of creating and configuring an OB1 block.

**Top Screenshot:** Shows the 'Properties - Organization Block' dialog box. The 'Name' field is set to 'OB1', and the 'Symbolic Name' is 'BLOCK\_CALL'. The 'Created in Language' is set to 'STL'. The 'Code' field is set to 'Interface'. The 'Comment' field contains the text 'Main Program Sweep (Cycle)'.

**Bottom Left Screenshot:** Shows the 'Contents of Environment (Interface)' dialog box. The 'OB1' block is listed with the comment 'Main Program Sweep (Cycle)'. A 'Network 1' is also visible with the title 'Title:'.

**Bottom Right Screenshot:** Shows the 'Contents of Environment (Interface)' dialog box with the 'OB1' block selected. The 'Network 1' is highlighted, and the following text is displayed:

```
OB1 : ΜΠΛΟΚ ΔΙΑΧΕΙΡΙΣΗΣ  
Το OB1 θα το χρησιμοποιήσουμε μόνο σαν διαχειριστή  
του project μας  
Network 1: ΚΛΗΣΗ ΥΠΟ ΕΥΝΘΗΚΗ ΣΤΗΝ STL  
Καλούμε υπο συνθήκη το FC1 στο οποίο θα επεξεργαστούμε  
τις απλές εντολές της Άλγεβρας BOOL
```

A	I	0,0	// Συνθ. κλήσης
CC	FC	1	

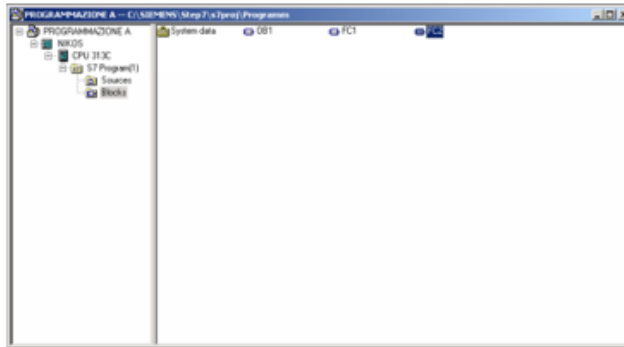


# Βασικές Εντολές Προγραμματισμού

- ΠΩΣ ΣΩΖΟΥΜΕ ΕΝΑ ΠΡΟΓΡΑΜΜΑ :FILE : SAVE



- ΠΩΣ ΔΗΜΙΟΥΡΓΟΥΜΕ ΕΝΑ FC:INSERT : S7 BLOKCS :FUNCTION



- ΠΩΣ ΔΗΜΙΟΥΡΓΟΥΜΕ ΕΝΑ ΝΕΟ NETWORK :INSERT : NETWORK



<https://youtu.be/BxNanbB8ne8?list=PLRtRKudOMmtFa9pcLrveRG5Cw4ZsvnYSo>

# Βασικές Εντολές Προγραμματισμού

## ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ BOOL ΣΤΗΝ STL

### ΚΛΗΣΗ ΕΝΟΣ FC ΥΠΟ ΣΥΝΘΗΚΗ : CC FC1

A // Συνδέει εν σειρά επαφή εργασίας  
AN // Συνδέει εν σειρά επαφή ηρεμίας  
O // Συνδέει παράλληλα επαφή εργασίας  
ON // Συνδέει παράλληλα επαφή ηρεμίας  
X // Λειτουργία αποκλειστικού ή  
XN // Λειτουργία αποκλειστικού ή με άρνηση  
A( // Συνδέει εν σειρά επαφή εργασ. σύνθ. κλάδου  
AN( // Συνδέει εν σειρά επαφή ηρεμί. σύνθ. κλάδου  
O( // Συνδέει παράλληλα επαφή εργ. σύνθ. κλάδου  
ON( // Συνδέει παράλληλα επαφή ηρεμ. σύνθ. κλάδου  
X( // Λειτουργία αποκλειστικού ή σε κλάδους  
XN( // Λειτουργία αποκλειστικού ή με άρνηση σε κλάδους  
) // Κλείσιμο παρένθεσης  
= // Δίνει αποτέλεσμα ( έξοδος )  
R // Απενεργοποιεί μία SR μνήμη  
S // Ενεργοποιεί μία SR μνήμη

# Βασικές Εντολές Προγραμματισμού

## ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ BOOL ΣΤΗΝ LAD

Network 1: Title:  
Comment:

---

---

➤ Σύμβολο επαφής εργασίας



➤ Σύμβολο επαφής ηρεμίας



➤ Σύμβολο πηνίου



➤ Σύμβολο για να ανοίξουμε νέο κλάδο



➤ Σύμβολο για να κλείσουμε κάποιο κλάδο



➤ Empty Box



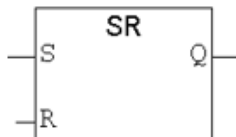
# Βασικές Εντολές Προγραμματισμού

## ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ BOOL ΣΤΗΝ LAD

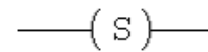
➤ Σύμβολο εντολής RS



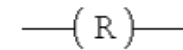
➤ Σύμβολο εντολής SR



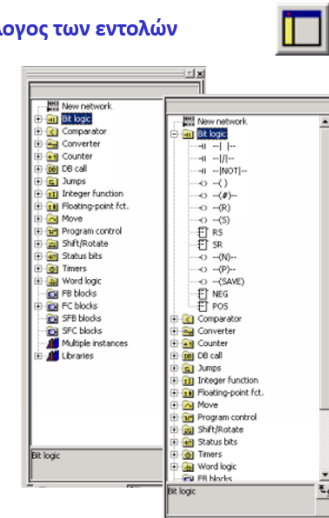
➤ Σύμβολο ενεργοποίησης ενός πηνίου μνήμης



➤ Σύμβολο απενεργοποίησης ενός πηνίου μνήμης



Ο κατάλογος των εντολών



# Βασικές Εντολές Προγραμματισμού

---

## **ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ**

➤ ΣΩΣΙΜΟ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΤΟ PG

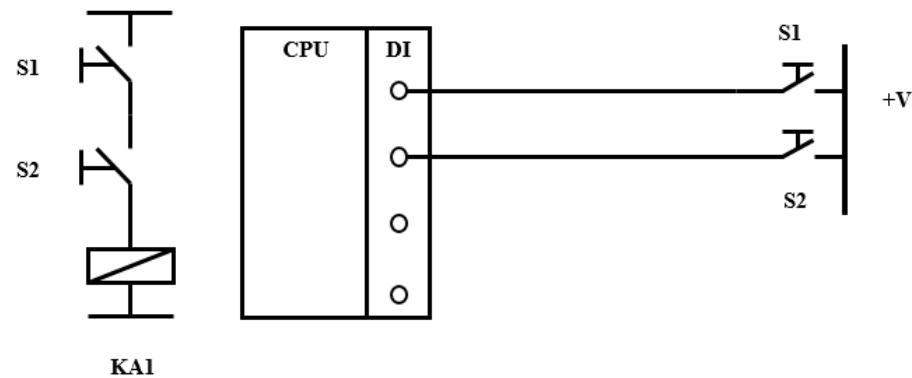


➤ ΔΟΣΙΜΟ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΤΟ PLC (DOWNLOAD)



# Βασικές Εντολές Προγραμματισμού

## ΠΑΡΑΔΕΙΓΜΑ Νο 1



FC1 : ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΝΤΟΛΩΝ BOOL ΣΕ STL

Comment:

Network 1: ΕΝ ΣΕΙΡΑ ΣΥΝΔΕΣΗ ΕΠΑΦΩΝ

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΔΥΟ ΕΠΑΦΩΝ ΕΡΓΑΣΙΑΣ ΕΝ ΣΕΙΡΑ ΣΥΝΔΕΔΕΜΕΝΕΣ

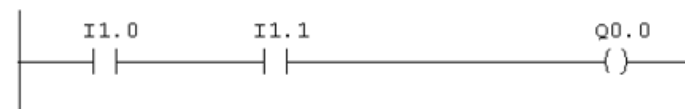
A	I	1.0
A	I	1.1
=	Q	0.0

FC2 : ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΝΤΟΛΩΝ BOOL ΣΕ LAD

Comment:

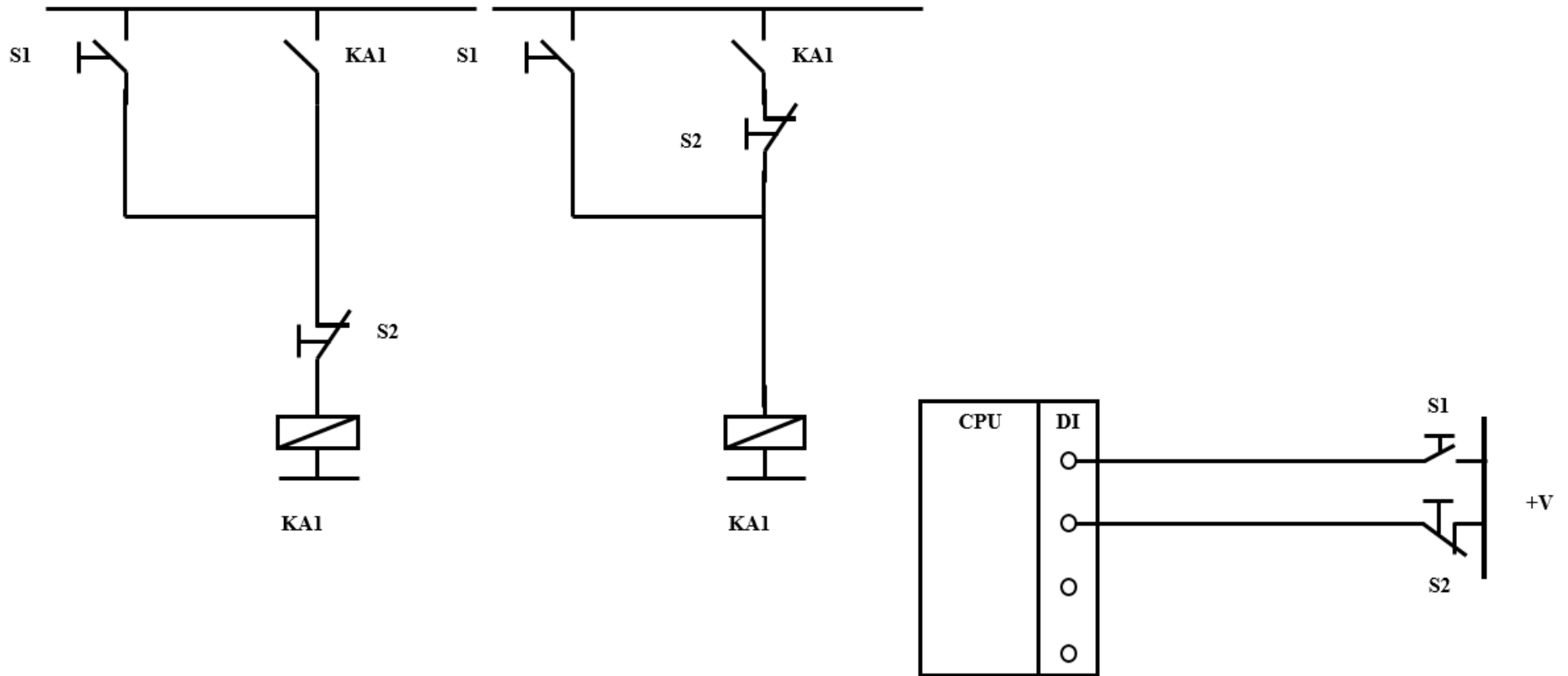
Network 1: ΕΝ ΣΕΙΡΑ ΣΥΝΔΕΣΗ ΕΠΑΦΩΝ

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΔΥΟ ΕΠΑΦΩΝ ΕΡΓΑΣΙΑΣ ΕΝ ΣΕΙΡΑ ΣΥΝΔΕΔΕΜΕΝΕΣ



# Βασικές Εντολές Προγραμματισμού

## ΠΑΡΑΔΕΙΓΜΑ Νο 2



# Βασικές Εντολές Προγραμματισμού

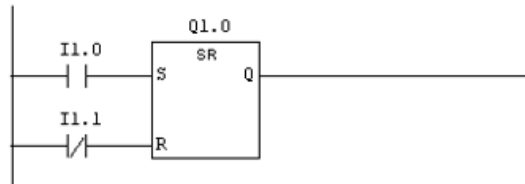
## ΠΑΡΑΔΕΙΓΜΑ Νο 2 Λύση στην LAD α:

FC10 : ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΝΗΜΗΣ

Comment:

**Network 1:** ΜΝΗΜΗ ΜΕ ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΣΤΟ R

Comment:

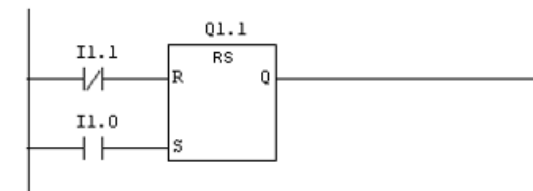


FC10 : ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΝΗΜΗΣ

Comment:

**Network 2:** ΜΝΗΜΗ ΜΕ ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΣΤΟ S

Comment:





# Βασικές Εντολές Προγραμματισμού

## ΠΑΡΑΔΕΙΓΜΑ Νο 2 Λύση στην LAD β:

FC10 : ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΝΗΜΗΣ

Comment:

**Network 3 :** ΜΝΗΜΗ ΜΕ ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΣΤΟ R

Comment:



**Network 4 :** Title:

Comment:

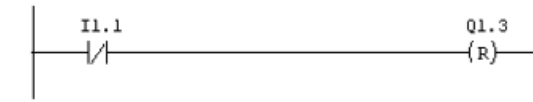


FC10 : ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΝΗΜΗΣ

Comment:

**Network 5 :** ΜΝΗΜΗ ΜΕ ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΣΤΟ S

Comment:



**Network 6 :** Title:

Comment:



# Βασικές Εντολές Προγραμματισμού

## ΠΑΡΑΔΕΙΓΜΑ Νο 2 Λύση στην STL α:

FC10 : ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΝΗΜΗΣ

Comment:

Network 1 : ΜΝΗΜΗ ΜΕ ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΣΤΟ R

Comment:

A	I	1.0
S	Q	1.0
AN	I	1.1
R	Q	1.0

FC10 : ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΝΗΜΗΣ

Comment:

Network 2 : ΜΝΗΜΗ ΜΕ ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΣΤΟ S

Comment:

AN	I	1.1
R	Q	1.1
A	I	1.0
S	Q	1.1

## ΠΑΡΑΔΕΙΓΜΑ Νο 2 Λύση στην STL b:

Network 3 : ΜΝΗΜΗ ΜΕ ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΣΤΟ R

Comment:

A	I	1.0
S	Q	1.2

Network 4 : Title:

Comment:

AN	I	1.1
R	Q	1.2

Network 5 : ΜΝΗΜΗ ΜΕ ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΣΤΟ S

Comment:

AN	I	1.1
R	Q	1.3

Network 6 : Title:

Comment:

A	I	1.0
S	Q	1.3

## **ΥΠΟΕΝΟΤΗΤΑ 7.7**

### ***Βασικά Εργαλεία στην Step 7***

## Βασικά Εργαλεία στην Step 7

---

### **ΕΛΕΓΧΟΣ ΛΕΙΤΟΥΡΓΙΑΣ ΠΡΟΓΡΑΜΜΑΤΟΣ**

Με το εργαλείο Monitor μπορούμε να ελέγξουμε το πρόγραμμα που περιέχεται σ' ένα μπλοκ.

- ✓ Να υπάρχει ON LINE σύνδεση μεταξύ της συσκευής προγραμματισμού και της CPU.
- ✓ Η CPU να βρίσκεται σε κατάσταση RUN ή RUN-P.
- ✓ Το πρόγραμμα που θέλουμε να ελέγξουμε πρέπει να έχει φορτωθεί στην CPU.

Επιλέγοντας το εργαλείο MONITOR η συσκευή προγραμματισμού συνδέεται ON LINE με την CPU και μας δείχνει την κατάσταση των μεταβλητών του προγράμματος σε Real Time.



## Βασικά Εργαλεία στην Step 7

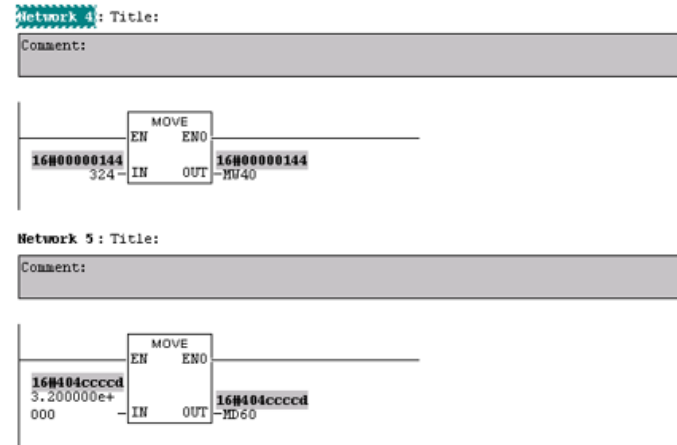
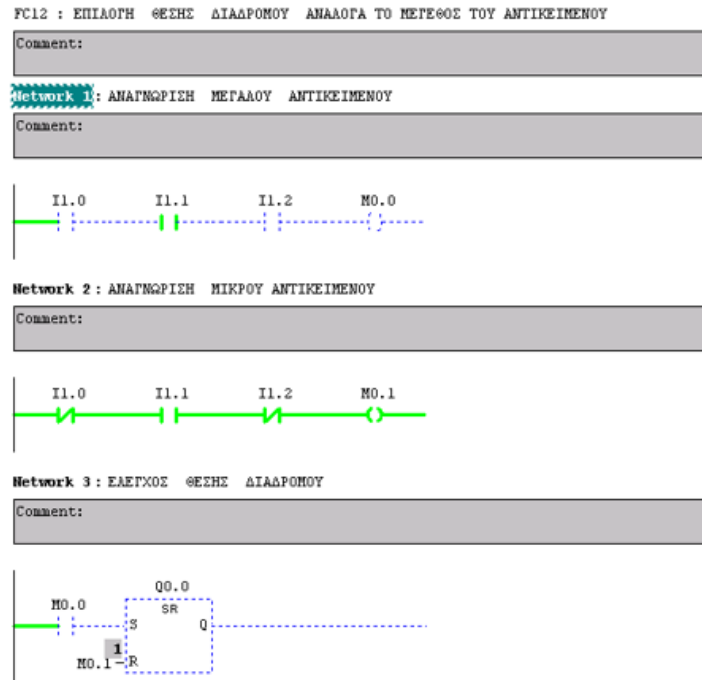
### Η λειτουργία Monitor στην STL

			▲	RLO	STA	STANDARD
AN	I	1.0		1	0	0
A	I	1.1		0	0	0
A	I	1.2		0	0	0
=	M	0.0		0	0	0

- ✓ **HIDE**
- ✓ **SHOW**
  - ✓ RLO
  - ✓ STA
  - ✓ DEFAULT STATUS
  - ✓ ADDRESS REGISTER 1/2
  - ✓ ACCU2
  - ✓ DB REGISTER 1/2
  - ✓ INDIRECT
  - ✓ STATUS WORD
- ✓ **REPRESENTATION**
  - ✓ DECIMAL
  - ✓ HEXADECIMAL
  - ✓ FLOATING – POINT
- ✓ **DIVIDING LINES**

# Βασικά Εργαλεία στην Step 7

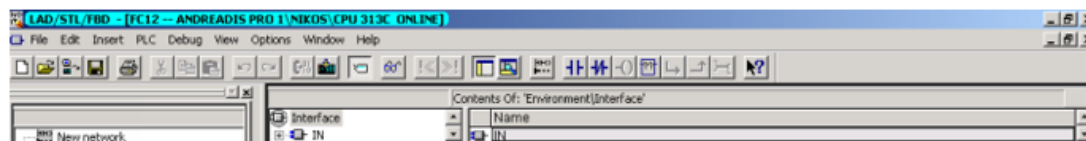
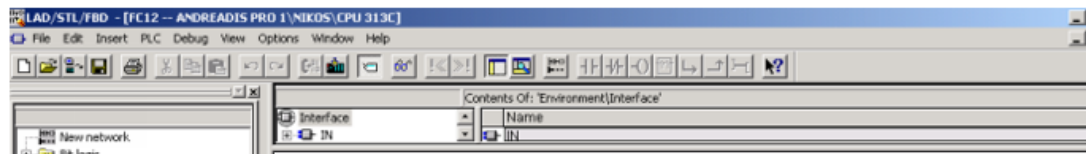
## Η λειτουργία Monitor στην LAD



## Βασικά Εργαλεία στην Step 7

### **ΕΡΓΑΛΕΙΟ OPEN OFFLINE / ONLINE PARTNER**

*Χρησιμεύει για να μπορέσουμε να βλέπουμε για ένα ανοιγμένο Μπλοκ στην συσκευή προγραμματισμού το αντίστοιχο που υπάρχει στην ON LINE (δηλαδή στην CPU) ή στην OFF LINE (δηλαδή στην συσκευή προγραμματισμού) κατάσταση.*



## Βασικά Εργαλεία στην Step 7

---

### ΠΙΝΑΚΑΣ ΣΥΜΒΟΛΩΝ

#### Άμεση Διευθυνσιοδότηση : *Direct Addressing*

##### ✓ *Απόλυτη διευθυνσιοδότηση : Absolute Address*

Ένα γράμμα για να μας καθορίσει το είδος της μεταβλητή  
π.χ.  $I$  = είσοδος ,  $Q_2$  = έξοδος ,  $M$  = market,  $T$  = χρονικό  
Ένα γράμμα και μία διεύθυνση ανάλογα το μέγεθος της  
μεταβλητής π.χ.  $I0.0$  ή  $QW2$ ,  $MD4$  ...

##### ✓ *Συμβολική διευθυνσιοδότηση : Symbolic Addresses*

Αντιστοιχούμε σε κάθε χρησιμοποιούμενη λογική μεταβλητή ή σε  
κάθε χρησιμοποιούμενο μπλοκ , ένα συμβολικό όνομα . Ο  
μέγιστος αριθμός των χαρακτήρων, που πρέπει να αποτελέσουν  
ένα σύμβολο, είναι 24. πχ : Motor – A



## Βασικά Εργαλεία στην Step 7

---

### ΠΙΝΑΚΑΣ ΣΥΜΒΟΛΩΝ

Τα χρησιμοποιούμενα σύμβολα μέσα σ' ένα πρόγραμμα χωρίζονται σε δύο κατηγορίες.

- ✓ Γενικά Σύμβολα-SHARED SYMBOLS: Έχουν ισχύ σε όλα το πρόγραμμα. : " "
- ✓ Τοπικά Σύμβολα -LOCAL SYMBOLS: Έχουν ισχύ μόνο στο μπλοκ του προγράμματος όπου ορίζονται. : #


## Βασικά Εργαλεία στην Step 7

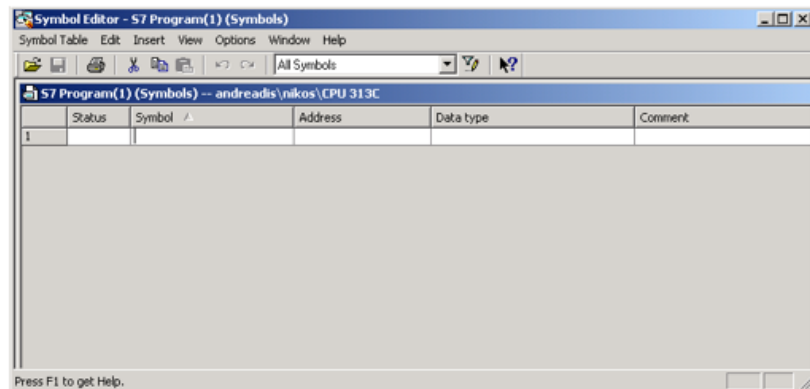
### ΠΙΝΑΚΑΣ ΣΥΜΒΟΛΩΝ

	<i>SHARED SYMBOLS</i>
ΙΣΧΥΣ	<ul style="list-style-type: none"><li>✓ ΙΣΧΥΕΙ ΣΕ ΟΛΟ ΤΟ ΠΡΟΓΡΑΜΜΑ ΤΟΥ ΧΡΗΣΤΗ</li><li>✓ ΜΠΟΡΕΙ ΝΑ ΧΡΗΣΙΜΟΠΟΙΗΘΕΙ ΑΠΟ ΟΛΑ ΤΑ ΜΠΛΟΚ</li><li>✓ ΕΧΕΙ ΤΗΝ ΙΔΙΑ ΕΝΝΟΙΑ ΣΕ ΟΛΑ ΤΑ ΜΠΛΟΚ</li><li>✓ ΠΡΕΠΕΙ ΝΑ ΕΙΝΑΙ ΜΟΝΑΔΙΚΟ ΣΕ ΟΛΟ ΤΟ ΠΡΟΓΡΑΜΜΑ ΤΟΥ ΧΡΗΣΤΗ</li></ul>
ΕΠΙΤΡΕΠΟΜΕΝΟΙ ΧΑΡΑΚΤΗΡΕΣ	✓ ΓΡΑΜΜΑΤΑ , ΑΡΙΘΜΟΙ , ΕΙΔΙΚΟΙ ΧΑΡΑΚΤΗΡΕΣ
ΧΡΗΣΗ	<p>ΜΠΟΡΟΥΜΕ ΝΑ ΟΡΙΣΟΥΜΕ ΣΥΜΒΟΛΑ ΓΙΑ</p> <ul style="list-style-type: none"><li>✓ I , IB , IW , ID , Q , QB , QW , QD</li><li>✓ PI., PQ..</li><li>✓ M , MB , MW , MD</li><li>✓ TIMERS , COUNTERS</li><li>✓ OB , FB , FC , SFB , SFC</li><li>✓ DB , UDT</li><li>✓ VAT</li></ul>
ΟΡΙΖΟΝΤΑΙ ..	ΠΙΝΑΚΑ ΣΥΜΒΟΛΩΝ

# Βασικά Εργαλεία στην Step 7

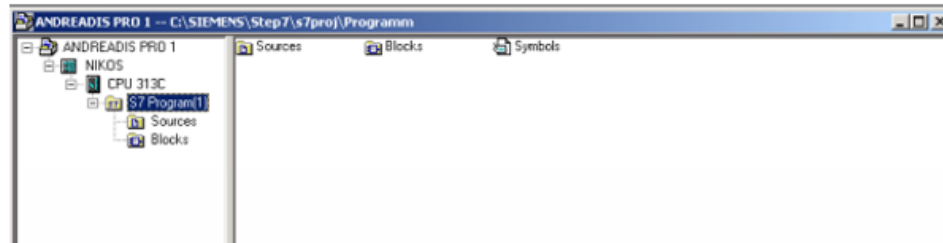
## ΠΙΝΑΚΑΣ ΣΥΜΒΟΛΩΝ

- ✓ ΑΥΞΩΝ ΑΡΙΘΜΟΣ
- ✓ STATUS :ΕΑΝ ΕΧΕΙ ΓΙΝΕΙ ΣΩΣΤΗ ΑΝΤΙΣΤΟΙΧΗΣΗ ΣΥΜΒΟΛΟΥ ΚΑΙ ΜΕΤΑΒΛΗΤΗΣ 
- ✓ SYMBOL:ΣΥΜΠΛΗΡΩΝΟΥΜΕ ΤΗΝ ΟΝΟΜΑΣΙΑ ΤΟΥ ΣΥΜΒΟΛΟΥ ( 24 ΧΑΡΑΚΤ.)
- ✓ ADDRESS :ΣΥΜΠΛΗΡΩΝΟΥΜΕ ΤΗΝ ΛΟΓΙΚΗ ΔΙΕΥΘΥΝΣΗ ΤΗΣ ΜΕΤΑΒΛΗΤΗΣ
- ✓ DATA TYPE :ΑΝΑΓΡΑΦΕΤΑΙ ΑΠΟ ΤΟΝ ΕΔΙΤΟΡ Ο ΤΥΠΟΣ ΔΕΔΟΜΕΝΩΝ ΤΗΣ ΜΕΤΑΒΛΗΤΗΣ
- ✓ COMMENT :ΓΡΑΦΟΥΜΕ ΤΥΧΟΝ ΣΧΟΛΙΑ ΠΟΥ ΘΕΛΟΥΜΕ ( 80 ΧΑΡΑΚΤ.)



# Βασικά Εργαλεία στην Step 7

## ΠΙΝΑΚΑΣ ΣΥΜΒΟΛΩΝ



The screenshot shows the Symbol Editor window for S7 Program(1) (Symbols). The window title is "Symbol Editor - S7 Program(1) (Symbols)". The menu bar includes Symbol Table, Edit, Insert, View, Options, Window, and Help. The toolbar contains icons for file operations and a search field labeled "All Symbols". The main area displays a table of symbols for the project "S7 Program(1) (Symbols) -- andreadis\nikos\CPU 313C".

	Status	Symbol	Address	Data type	Comment
1		ΦΩΤΟΚΥΤΤΑΡΟ_1	I 5.0	BOOL	ΑΙΣΘΗΤΗΡΙΟ N 1
2		ΦΩΤΟΚΥΤΤΑΡΟ_2	I 5.1	BOOL	ΑΙΣΘΗΤΗΡΙΟ N 2
3		ΦΩΤΟΚΥΤΤΑΡΟ_3	I 5.2	BOOL	ΑΙΣΘΗΤΗΡΙΟ N 3
4		ΜΕΓΑΛΟ_TEM	M 10.0	BOOL	ΒΟΗΘΗΤΙΚΟ ΡΕΛΕ ΑΝΑΓΝΩΡΙΣΗΣ ΜΕΓΑΛΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ
5		ΜΙΚΡΟ_TEM	M 10.1	BOOL	ΒΟΗΘΗΤΙΚΟ ΡΕΛΕ ΑΝΑΓΝΩΡΙΣΗΣ ΜΙΚΡΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ
6		ΘΕΣΗ_A	Q 20.0	BOOL	ΘΕΣΗ ΔΙΑΔΡΟΜΟΥ ΓΙΑ ΜΕΓΑΛΑ ΑΝΤΙΚΕΙΜΕΝΑ
7		ΘΕΣΗ_B	Q 20.1	BOOL	ΘΕΣΗ ΔΙΑΔΡΟΜΟΥ ΓΙΑ ΜΙΚΡΑ ΑΝΤΙΚΕΙΜΕΝΑ
8		ΕΛΕΓΧΟΣ ΔΙΑΔΡΟΜΟΥ	FC 12	FC 12	ΜΠΛΟΚ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΔΙΑΔΡΟΜΟΥ
9					

Press F1 to get Help. CAPS

## Βασικά Εργαλεία στην Step 7

### Εμφάνιση προγράμματος στην STL με σύμβολα

FC12 : ΚΥΚΛΩΜΑ ΕΛΕΓΧΟΥ ΘΕΣΗΣ ΔΙΑΔΡΟΜΟΥ

**Network 1:** ΒΟΗΘΗΤΙΚΟ ΡΕΛΕ ΑΝΑΓΝΩΡΙΣΗΣ ΜΕΓΑΛΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

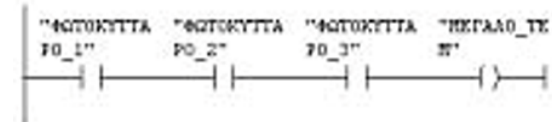
```
A      "ΦΩΤΟΚΥΤΤΑΡΟ_1"  
A      "ΦΩΤΟΚΥΤΤΑΡΟ_2"  
A      "ΦΩΤΟΚΥΤΤΑΡΟ_3"  
=      "ΜΕΓΑΛΟ_ΤΕΜ"
```

**Network 2:** ΒΟΗΘΗΤΙΚΟ ΡΕΛΕ ΑΝΑΓΝΩΡΙΣΗΣ ΜΙΚΡΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

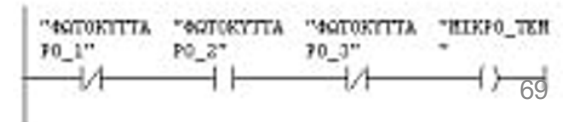
```
AN     "ΦΩΤΟΚΥΤΤΑΡΟ_1"  
A      "ΦΩΤΟΚΥΤΤΑΡΟ_2"  
AN     "ΦΩΤΟΚΥΤΤΑΡΟ_3"  
=      "ΜΙΚΡΟ_ΤΕΜ"
```

### Εμφάνιση προγράμματος στην LAD με σύμβολα

**Network 1:** ΒΟΗΘΗΤΙΚΟ ΡΕΛΕ ΑΝΑΓΝΩΡΙΣΗΣ ΜΕΓΑΛΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ



**Network 2:** ΒΟΗΘΗΤΙΚΟ ΡΕΛΕ ΑΝΑΓΝΩΡΙΣΗΣ ΜΙΚΡΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ



## Βασικά Εργαλεία στην Step 7

---

### **ΠΙΝΑΚΑΣ ΜΕΤΑΒΛΗΤΩΝ-VARIABLE TABLE**

Οι πίνακες μεταβλητών μας δίνουν το πλεονέκτημα της δυνατότητας παρακολούθησης διαφορετικών καταστάσεων ελέγχου και παρατηρήσεων ενός προγράμματος

Ο αριθμός του Variable Table που μπορούμε να δημιουργήσουμε σ' ένα project είναι 9.999.

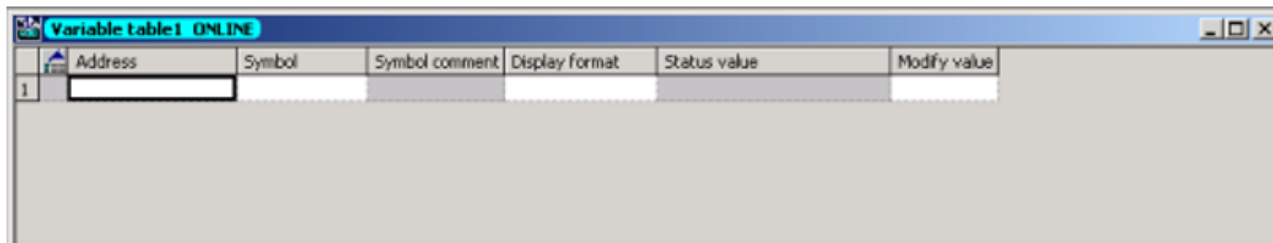
Μπορούμε να έχουμε σύνδεση με την CPU έχοντας ταυτόχρονα 6 Variable Table ανοιγμένα

Κάθε πίνακας μεταβλητών μπορεί να έχει ως 1024 γραμμές και κάθε γραμμή μπορεί να περιέχει το μέγιστο έως 255 χαρακτήρες.

## Βασικά Εργαλεία στην Step 7

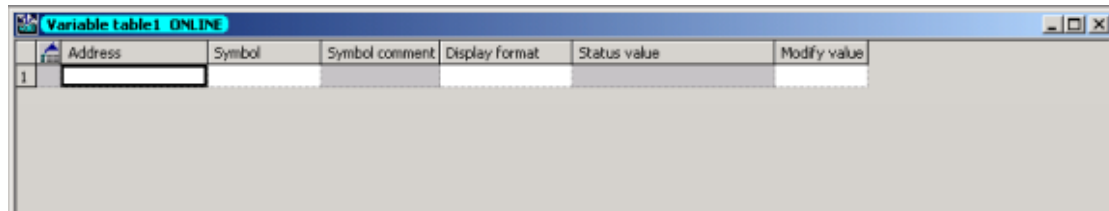
### *Πως ανοίγουμε έναν πίνακα μεταβλητών*

- ✓ Από το κεντρικό μενού του *SIMATIC Manager* και ανοιγμένο το *project* έως το επίπεδο *Block* επιλέγουμε *Insert, S7 Block, Variable Table*
- ✓ Έχοντας ανοιγμένο ένα μπλοκ προγραμματισμού *OB1, FC, FB* επιλέγουμε *PLC, Monitor/Modify Variables*.



## Βασικά Εργαλεία στην Step 7

### Οι στήλες του πίνακα μεταβλητών



	Address	Symbol	Symbol comment	Display format	Status value	Modify value
1						

- ✓ **ROW:** Αύξων αριθμός της γραμμής του πίνακα μεταβλητών
- ✓ **IDENTIFICATION:** Μας δείχνει εάν η μεταβλητή έχει υποβληθεί σε εξαναγκασμό ή όχι
- ✓ **ADDRESS:** Εδώ γράφουμε την λογική διεύθυνση της μεταβλητής
- ✓ **SYMBOL:** Εδώ γράφουμε την τυχόν χρησιμοποιούμενο σύμβολο μιας μεταβλητής.
- ✓ **SYMBOLIC COMMENT:** Εδώ ο editor του προγράμματος συμπληρώνει από μόνος του τυχόν σχόλια που συνοδεύουν το χρησιμοποιούμενο σύμβολο
- ✓ **DISPLAY FORMAT:** Εδώ ο editor συμπληρώνει από μόνος του το format με το οποίο θα εμφανίσει την τιμή της λογικής μεταβλητής
- ✓ **STATUS VALUE:** Απεικονίζει την τελευταία τιμή που έλαβε ο πίνακας μεταβλητών από το PLC για την αντίστοιχη μεταβλητή.
- ✓ **MODIFY VALUE:** Εδώ συμπληρώνουμε την τιμή που θέλουμε να επιβάλλουμε στην μεταβλητή μέσω της εντολής Modify



## Βασικά Εργαλεία στην Step 7

### ΠΙΝΑΚΑΣ ΜΕΤΑΒΛΗΤΩΝ-VARIABLE TABLE

	Address	Symbol	Symbol comment	Display format	Status value	Modify value
1	I 1.0	"ET"	Μπουτον Ετοιμότητας Λειτουργίας	BOOL	false	
2	I 1.1	"STOP"	Μπουτον Απενεργοποίησης Λειτουργίας	BOOL	true	true
3	I 1.2	"S1"	Μπουτον Χειρακίνητου Γεμίματος	BOOL	X	X
4	I 1.3	"F1"	Θερμικό Αντίλας	BOOL	true	
5	I 1.4	"X1"	Όριο Χαμηλής Στάθμης στην Δ1	BOOL	false	
6	I 1.5	"Y1"	Όριο Υψηλής Στάθμης στην Δ1	BOOL	true	
7	I 1.6	"X2"	Όριο Χαμηλής Στάθμης στην Δ2	BOOL	false	
8	I 1.7	"Y2"	Όριο Υψηλής Στάθμης στην Δ2	BOOL	true	
9	Q 0.0	"HA.BAΛΒΙΔΑ V1"	Ρελέ Ελέγχου Κατάστασης Ηλεκτροβαλβίδας V 1	BOOL	false	
10	Q 0.1	"HA.BAΛΒΙΔΑ V2"	Ρελέ Ελέγχου Κατάστασης Ηλεκτροβαλβίδας V 2	BOOL	false	
11	Q 0.2	"ANTΛΙΑ"	Ρελέ Αντλίας MA	BOOL	false	
12	M 20.0	"Ετοιμότητα Λειτουργίας"	Βοηθητικό Ρελέ Ετοιμότητας Λειτουργίας	BOOL	false	
13	M 20.1	"Χαμηλή Δ1"	Βοηθητικό Ρελέ Αναγνώρισης Χαμηλής Στάθμης στην Δ1	BOOL	false	
14	M 20.2	"Χαμηλή Δ2"	Βοηθητικό Ρελέ Αναγνώρισης Χαμηλής Στάθμης στην Δ2	BOOL	false	
15						

## **ΥΠΟΕΝΟΤΗΤΑ 7.8**

### ***Εντολές Load & Transfer***

## Εντολές Load & Transfer

---

- ✓ Η ροή πληροφορίας από μια περιοχή μνήμης στον ACCU 1 λέγεται φόρτωση (Loading) και επιτυγχάνεται χρησιμοποιώντας την εντολή 'L' (Load)
- ✓ Η ροή πληροφορίας από τον ACCU1 σε μια οποιαδήποτε άλλη περιοχή μνήμης λέγεται μεταφορά (Transferring) και επιτυγχάνεται χρησιμοποιώντας την εντολή 'T' (Transfer)

## Εντολές Load & Transfer

---

Όταν θέλουμε να διαβάσουμε ή να γράψουμε μια πληροφορία της οποίας το μέγεθος είναι ένα bit τότε, όπως είδαμε μέχρι τώρα, χρησιμοποιούμε τις λογικές εντολές (π.χ. για διάβασμα A I0.0 ή O I0.1. ενώ όταν θέλουμε να γράψουμε π.χ. = M0.0)

Εάν η πληροφορία την οποία θέλουμε να επεξεργαστούμε έχει μέγεθος **από byte και πάνω τότε, πρέπει να χρησιμοποιήσουμε τις εντολές Load (φόρτωση) και Transfer (Μεταφορά)**. Όταν δουλεύουμε με τις εντολές Load και Transfer υπεισέρχονται στην λειτουργία του PLC οι λεγόμενοι συσσωρευτές (Accumulator).

Στην σειρά S7-300 των PLC υπάρχουν δύο Accumulator ο ACCU1 και ο ACCU2.

Στην σειρά S7-400 του PLC υπάρχουν τέσσερις Accumulator ο ACCU1, ο ACCU2, ο ACCU3 και ο ACCU4.

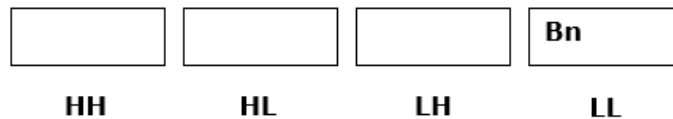
Οι Accumulators είναι θέσεις μνήμης μεγέθους 32 bit τους οποίους χρησιμοποιεί ο Editor του προγράμματος για να διαβάσει ή να μεταφέρει μια πληροφορία της οποίας το μέγεθος είναι από byte και πάνω.

## Εντολές Load & Transfer

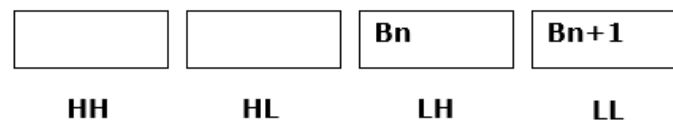
### ΣΥΝΑΡΤΗΣΗ ΦΟΡΤΩΣΗΣ: LOAD L

Η συνάρτηση φόρτωσης αποτελείται από τον κώδικα λειτουργίας ο οποίος είναι το γράμμα «L» και από το στοιχείο που θέλουμε να φορτώσουμε που μπορεί να είναι ένας σταθερός αριθμός ή μια αναγνωρίσιμη διεύθυνση

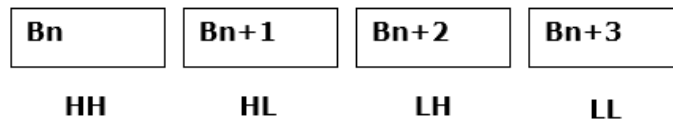
- Φόρτωση πληροφορίας μεγέθους Byte



- Φόρτωση πληροφορίας μεγέθους Word



- Φόρτωση πληροφορίας μεγέθους Double Word



## Εντολές Load & Transfer

---

### ΣΥΝΑΡΤΗΣΗ ΦΟΡΤΩΣΗΣ: LOAD L

➤ Φόρτωση εισόδων πάνω από τις κάρτες

- ✓ L PIBn
- ✓ L PIWn
- ✓ L PIDn

➤ Φόρτωση εξόδων από την περιοχή μνήμης PIQ

- ✓ L QBn
- ✓ L QWn
- ✓ L QDn

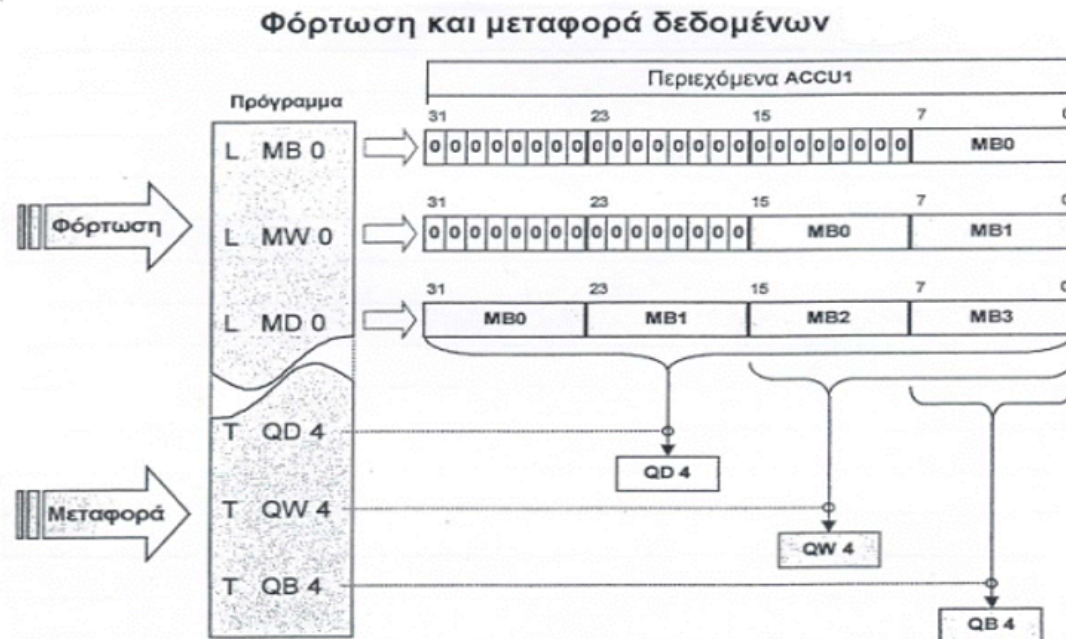
➤ Φόρτωση μεταβλητών από την περιοχή μνήμης του Merker

- ✓ L MBn
- ✓ L MWn
- ✓ L MDn

# Εντολές Load & Transfer

## ΣΥΝΑΡΤΗΣΗ ΜΕΤΑΦΟΡΑΣ: TRANSFER T

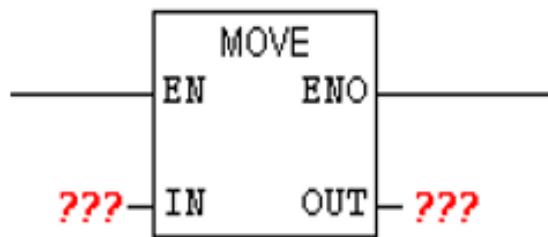
Η συνάρτηση μεταφοράς αποτελείται από τον κωδικό λειτουργίας «T» και την λογική διεύθυνση της περιοχής που θέλουμε να μεταφερθεί το περιεχόμενο του ACCU1



## Εντολές Load & Transfer

---

### Εντολές Load & Transfer στην LAD



- Είσοδος EN
- Είσοδος IN
- Έξοδος ENO
- Έξοδος OUT



## ΥΠΟΕΝΟΤΗΤΑ 7.9

*Χρονικά*

## Χρονικά

---

### ΧΡΟΝΙΚΑ T x

<b>T</b>	Προσδιορίζει την περιοχή μνήμης των χρονικών
<b>x</b>	Προσδιορίζει την διεύθυνση της word μέσα στην οποία αποθηκεύεται η τιμή του συγκεκριμένου χρονικού

**Μέγιστος χρόνος : 9990 sec ή 2H – 46M – 30s.**

# Χρονικά

## Πρόσδοση χρονικής καθυστέρησης

*W # 16# t xyz*

<i>t</i>	<i>Βάση χρόνου</i>
<i>xyz</i>	<i>Απόλυτη αριθμητική τιμή</i>

Time Base	Binary Code for the Time Base
10 ms	00
100 ms	01
1 s	10
10 s	11

Resolution	Range
0.01 second	10MS to 9S_990MS
0.1 second	100MS to 1M_39S_900MS
1 second	1S to 16M_39S
10 seconds	10S to 2H_46M_30S

✓ *L W # 16 # 2025*

✓ *L W # 16 # 1250*

}

*25 sec*

## Χρονικά

---

### Πρόσδοση χρονικής καθυστέρησης

*S5T # aH – bbM – ccS – ddMS*

<b><i>a</i></b>	<i>παρουσιάζει τον αριθμό των ωρών</i>
<b><i>bb</i></b>	<i>παρουσιάζει τον αριθμό των λεπτών</i>
<b><i>cc</i></b>	<i>παρουσιάζει τον αριθμό των δευτερολέπτων</i>
<b><i>dd</i></b>	<i>παρουσιάζει τον αριθμό των χιλιοστών των δευτερολέπτων</i>

**L S5T # 25 S**

## Χρονικά

---

*Τα είδη των χρονικών που διαθέτουμε στην STEP 7*

- ✓ *S – PULSE : Χρονικό παλμού*
- ✓ *S – PEXT : Χρονικό παλμού με αυτοσυγκράτηση*
- ✓ *S – ODT : Χρονικό καθυστέρησης έλξης*
- ✓ *S – ODTS : Χρονικό καθυστέρησης έλξης με αυτοσυγκράτηση*
- ✓ *S- OFFDT : Χρονικό καθυστέρησης πτώσης*

# Χρονικά

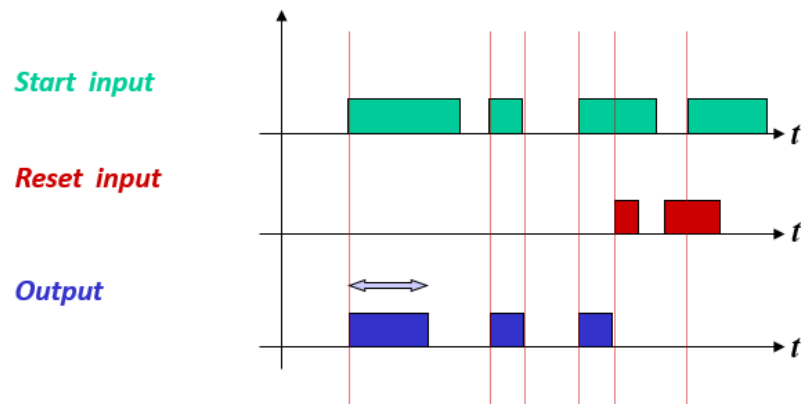
---

## ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΩΝ ΧΡΟΝΙΚΩΝ

- ✓ **Απαραίτητες εντολές για την λειτουργία των χρονικών**
  - Να δηλώσουμε την αιτία εκκίνησης του χρονικού**
  - Να δηλώσουμε την χρονική καθυστέρηση του χρονικού**
  - Να δηλώσουμε το είδος του χρονικού**
  - Να δηλώσουμε σε ποιο από όλα τα χρονικά αναφερόμαστε**
  
- ✓ **Προαιρετικές εντολές των χρονικών**
  - Η χρησιμοποίηση της εντολής *Reset***
  - Η φόρτωση του περιεχομένου της μνήμης του χρονικού στον *ACCU1***

# Χρονικά

## ΧΡΟΝΙΚΟ ΠΑΛΜΟΥ SP PULSE TIMER

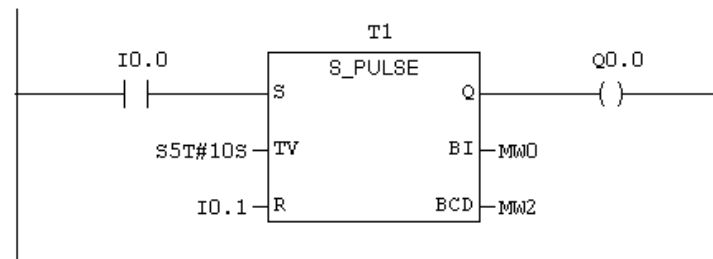


```

A      I      0.0          // ΣΥΝΘΗΚΗ ΕΚΚΙΝΗΣΗΣ           :ΑΠΑΡΑΙΤ. ΕΝΤΟΛΗ
L      S5T#10S          // ΔΙΑΡΚΕΙΑ ΧΡΟΝΙΚΗΣ ΚΑΘΥΣΤΕΡΗΣΗΣ :ΑΠΑΡΑΙΤ. ΕΝΤΟΛΗ
SP     T      1          // ΔΙΑΡΚΕΙΑ ΒΙΑΣ ΧΡΟΝΙΚΟΥ          :ΑΠΑΡΑΙΤ. ΕΝΤΟΛΗ
A      I      0.1          // ΣΥΝΘΗΚΗ ΜΗΔΕΝΙΣΜΟΥ             :ΠΡΟΑΙΡΕΤ. ΕΝΤΟΛΗ
R      T      1          // ΜΗΔΕΝΙΣΜΟΣ ΧΡΟΝΙΚΟΥ            :ΠΡΟΑΙΡΕΤ. ΕΝΤΟΛΗ
A      T      1          // ΧΡΗΣΙΜΟΠΟΙΗΣΗ ΕΠΑΦΗΣ ΧΡΟΝΙΚΟΥ  :ΠΡΟΑΙΡ. ΕΝΤ.
=      Q      0.0
L      T      1          // ΦΟΡΤΩΣΗ ΠΕΡΙΕΧ. ΤΗΣ ΜΝΗΜΗΣ ΣΤΟ ΔΥΑΔ.ΣΥΣ.ΑΡΙΘ. : ΠΡΟΑΙΡ.Ε
T      MW      0
LC     T      1          // ΦΟΡΤΩΣΗ ΠΕΡΙΕΧ. ΤΗΣ ΜΝΗΜΗΣ ΣΕ BCD ΚΩΔΙΚΑ : ΠΡΟΑΙΡ.Ε
T      MW      2
    
```

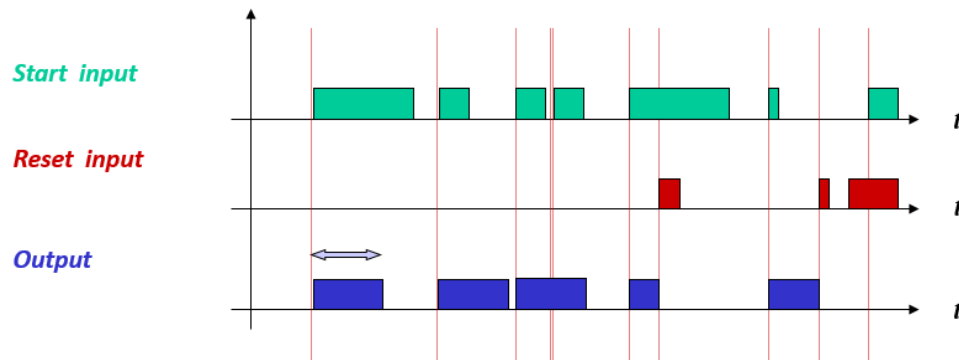
**Network 1:** ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΧΡΟΝΙΚΟΥ ΠΑΛΜΟΥ

Comment:



# Χρονικά

## ΧΡΟΝΙΚΟ ΠΑΛΜΟΥ ΜΕ ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ SE



Network 1: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΧΡΟΝΙΚΟΥ ΠΑΛΜΟΥ ΜΕ ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ

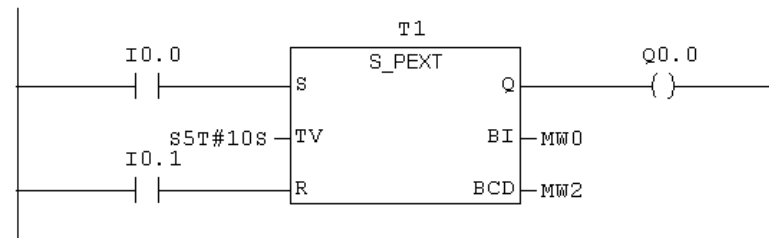
Comment:

```

A      I      0.0
L      S5T#10S
SE     T      1
A      I      0.1
R      T      1
A      T      1
=      Q      0.0
L      T      1
T      MW     0
LC     T      1
T      MW     2
    
```

Network 1: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΧΡΟΝΙΚΟΥ ΠΑΛΜΟΥ ΜΕ ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ

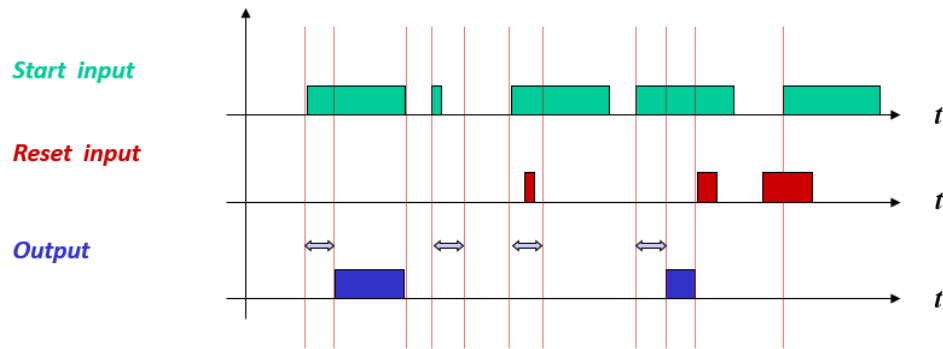
Comment:





# Χρονικά

## ΧΡΟΝΙΚΟ ΚΑΘΥΣΤΕΡΗΣΗΣ ΕΛΞΗΣ SD



Network 1: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΧΡΟΝΙΚΟΥ ΚΑΘΥΣΤΕΡΗΣΗΣ ΕΛΞΗΣ

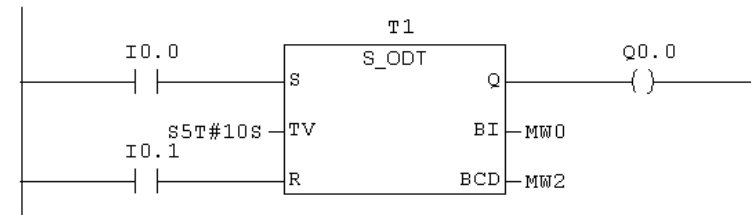
Comment:

```

A   I   0.0
L   S5T#10S
SD  T   1
A   I   0.1
R   T   1
A   T   1
=   Q   0.0
L   T   1
T   MW  0
LC  T   1
T   MW  2
    
```

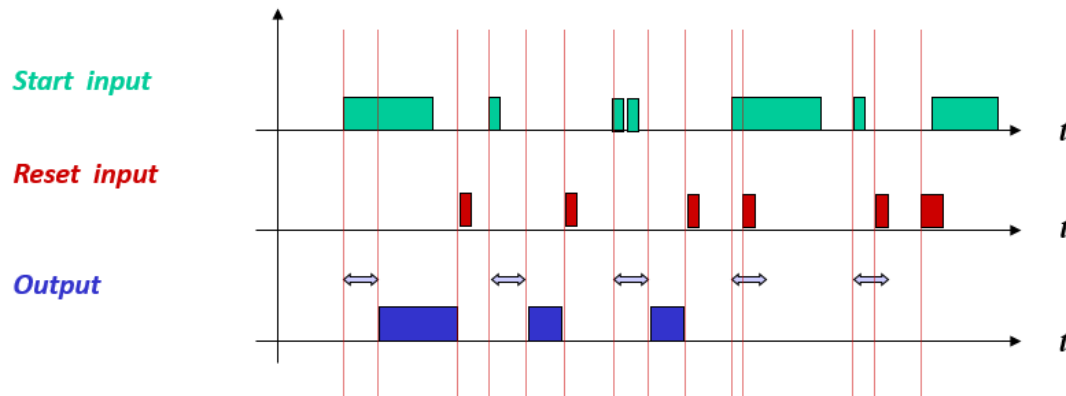
Network 1: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΧΡΟΝΙΚΟΥ ΚΑΘΥΣΤΕΡΗΣΗΣ ΕΛΞΗΣ

Comment:



# Χρονικά

## ΧΡΟΝΙΚΟ ΚΑΘΥΣΤΕΡΗΣΗΣ ΕΛΕΞΗΣ ΜΕ ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ SS



Network 1: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΧΡΟΝΙΚΟΥ ΚΑΘΥΣΤΕΡΗΣΗΣ ΕΛΕΞΗΣ ΜΕ ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ

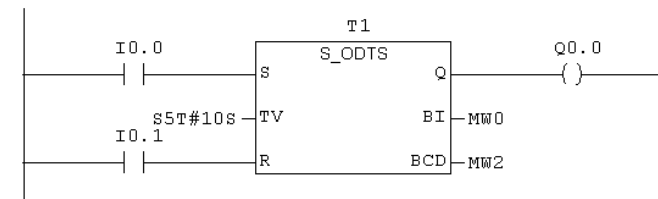
Comment:

```

A    I    0.0
L    S5T#10S
SS   T    1
A    I    0.1
R    T    1
A    T    1
=    Q    0.0
L    T    1
T    MW   0
LC   T    1
T    MW   2
    
```

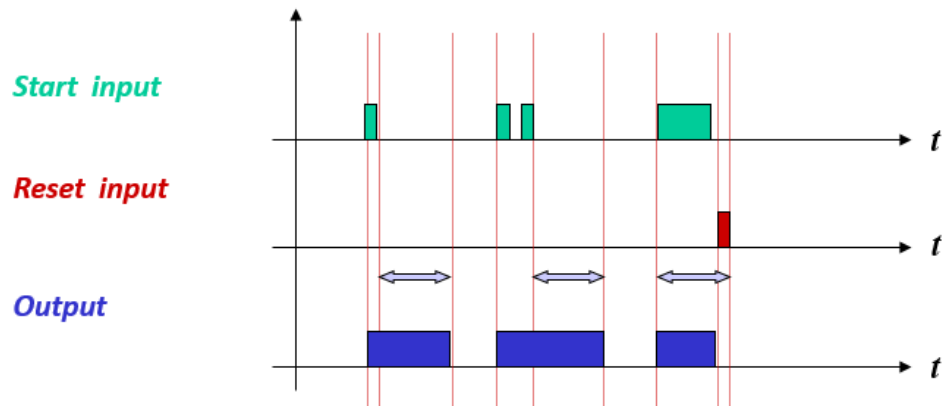
Network 1: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΧΡΟΝΙΚΟΥ ΚΑΘΥΣΤΕΡΗΣΗΣ ΕΛΕΞΗΣ ΜΕ ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ

Comment:



# Χρονικά

## ΧΡΟΝΙΚΟ ΚΑΘΥΣΤΕΡΗΣΗΣ ΠΤΩΣΗΣ SF OFF



Network 1: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΧΡΟΝΙΚΟΥ ΚΑΘΥΣΤΕΡΗΣΗΣ ΠΤΩΣΗΣ

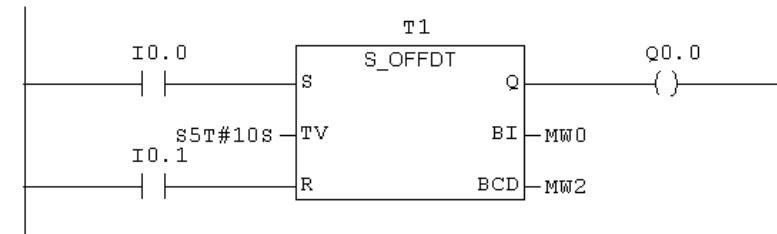
Comment:

```

A    I    0.0
L    S5T#10S
SF   T    1
A    I    0.1
R    T    1
A    T    1
=    Q    0.0
L    T    1
T    MW   0
LC   T    1
T    MW   2
    
```

Network 1: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΧΡΟΝΙΚΟΥ ΚΑΘΥΣΤΕΡΗΣΗΣ ΠΤΩΣΗΣ

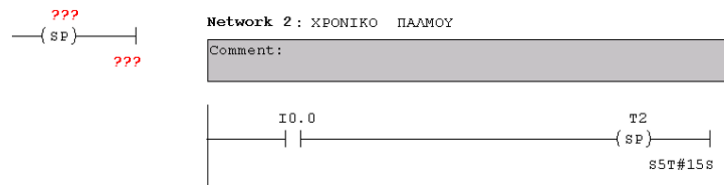
Comment:



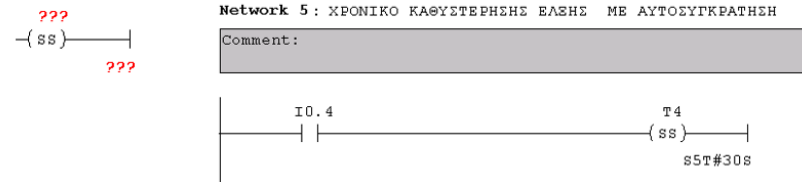
# Χρονικά

## Απλουστευμένες εντολές προγραμματισμού στην LAD

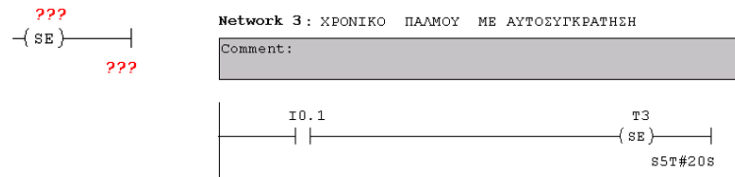
### ✓ ΧΡΟΝΙΚΟ ΠΑΛΜΟΥ



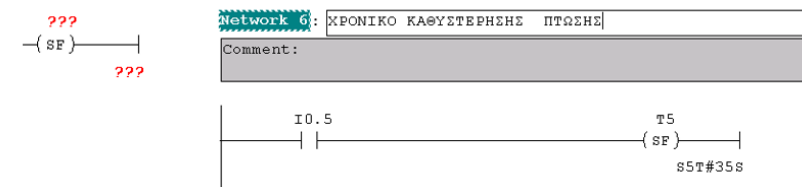
### ✓ ΧΡΟΝΙΚΟ ΚΑΘΥΣΤΕΡΗΣΗΣ ΕΛΞΗΣ ΜΕ ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ



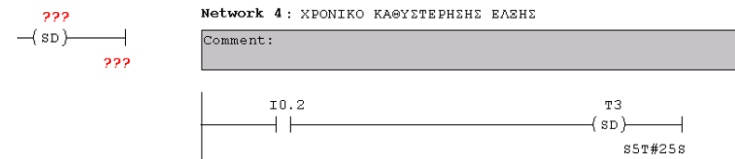
### ✓ ΧΡΟΝΙΚΟ ΠΑΛΜΟΥ ΜΕ ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ



### ✓ ΧΡΟΝΙΚΟ ΚΑΘΥΣΤΕΡΗΣΗΣ ΠΤΩΣΗΣ



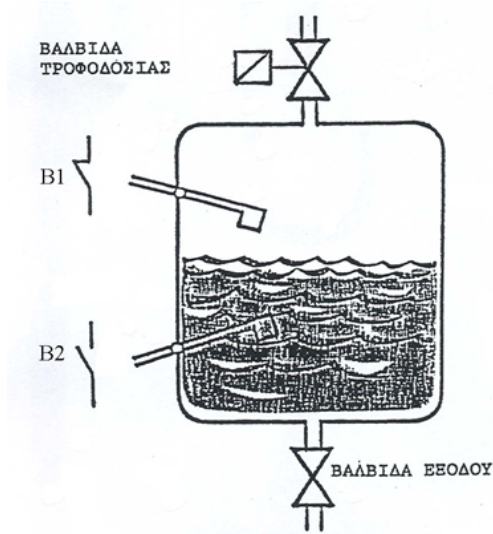
### ✓ ΧΡΟΝΙΚΟ ΚΑΘΥΣΤΕΡΗΣΗΣ ΕΛΞΗΣ



# Χρονικά

## Παράδειγμα

Ένα δοχείο τροφοδοτείται με υγρό από μία βαλβίδα . Η διαδικασία γεμίσματος πρέπει να αρχίσει όταν η στάθμη κατέβει κάτω από τον φλοτεροδιακόπτη B2 και να σταματάει όταν φτάσει το ύψος του φλοτεροδιακόπτη B1 . Το γέμισμα δεν πρέπει να σταματάει πρόωρα από τυχόν κυματισμούς ή πιτσιλίσματα . Πρέπει επίσης να υπάρχει δυνατότητα χειροκίνητης λειτουργίας . Η κατάσταση των επαφών ηρεμίας από τους φλοτεροδιακόπτες , ανάλογα την στάθμη στην δεξαμενή , φαίνεται στην δίπλα εικόνα .



# Χρονικά

## Παράδειγμα – Λύση στην STL

FC5 : ΤΡΟΦΟΔΟΣΙΑ ΔΟΧΕΙΟΥ ΜΕ ΥΓΡΟ

Comment:

**Network 1** : ΕΛΕΓΧΟΣ ΒΑΑΒΙΑΔΕΣ ΤΡΟΦΟΔΟΣΙΑΣ

Comment:

```
A(
O  "ΧΕΙΡΟΚ. START"          I0.0
O  "ΣΤΑΘΜΗ B2"              I0.2
)
S  "ΒΑΑΒΙΑΔΑ ΤΡΟΦΟΔΟΣΙΑΣ"  Q0.0
A(
ON "ΧΕΙΡΟΚ. STOP"           I0.1
O  "STOP ΣΤΑΘΜΗ B1"         M0.1
)
R  "ΒΑΑΒΙΑΔΑ ΤΡΟΦΟΔΟΣΙΑΣ"  Q0.0
NOP 0
```

**Network 2** : ΑΝΑΕΤΡΟΦΗ ΚΑΤΑΣΤΑΣΗΣ ΦΑΟΤΕΡΟΔΙΑΚΟΗΤΗ B1

Comment:

```
AN "ΣΤΑΘΜΗ B1"              I0.3
=  M 0.0
```

**Network 3** : ΕΝΕΡΓΟΠΟΙΗΣΗ ΧΡΟΝΙΚΟΥ ΓΙΑ ΜΕΤΡΗΣΗ ΧΡΟΝ.ΚΑΘΥΣΤΕΡΗΣΗΣ

Comment:

```
A  M 0.0
L  S5T#3S
SD "ΧΡΟΝΟΚΑΘΥΣΤΕΡΗΣΗ"      T1
```

**Network 4** : Title:

Comment:

```
A  "ΧΡΟΝΟΚΑΘΥΣΤΕΡΗΣΗ"      T1
=  "STOP ΣΤΑΘΜΗ B1"        M0.1
```

# Χρονικά

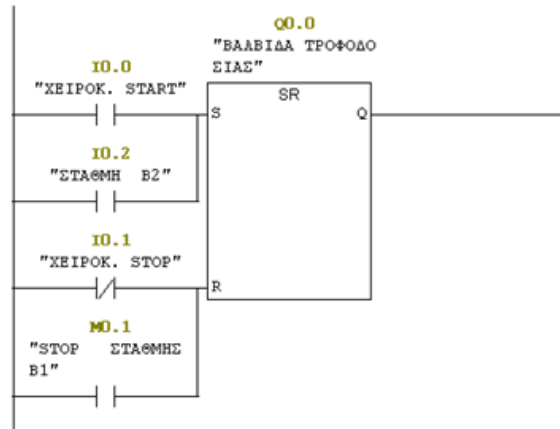
## Παράδειγμα – Λύση στην LAD

FC5 : ΤΡΟΦΟΔΟΣΙΑ ΔΟΧΕΙΟΥ ΜΕ ΥΓΡΟ

Comment:

**Network 1:** ΕΛΕΓΧΟΣ ΒΑΒΙΔΑΣ ΤΡΟΦΟΔΟΣΙΑΣ

Comment:



**Network 2:** ΑΝΑΣΤΡΟΦΗ ΚΑΤΑΣΤΑΣΗΣ ΦΑΩΤΕΡΟΔΙΑΚΟΠΤΗ B1

Comment:



**Network 3:** ΕΝΕΡΓΟΠΟΙΗΣΗ ΧΡΟΝΙΚΟΥ ΓΙΑ ΜΕΤΡΗΣΗ ΧΡΟΝ.ΚΑΘΥΣΤΕΡΗΣΗΣ

Comment:



**Network 4:** Title:

Comment:



**ΥΠΟΕΝΟΤΗΤΑ 7.10**  
***Απαριθμητές***



# Απαριθμητές

---

## Απαριθμητές Cx

<b>C</b>	Προσδιορίζει την περιοχή μνήμης των απαριθμητών
<b>x</b>	Προσδιορίζει την διεύθυνση της Word μέσα στην οποία αποθηκεύεται η τιμή ενός συγκεκριμένου απαριθμητή

Έχουμε την δυνατότητα να μετράμε κατ' αύξοντα ή κατά φθίνοντα τρόπο έχοντας τα εξής όρια για κάθε απαριθμητή.

<b>Επάνω όριο</b>	<b>999</b>
<b>Κάτω όριο</b>	<b>0</b>

## Απαριθμητές

---

**Τιμή προτοποθέτησης**

**W # 16 # wxyz**

**C # xyz**

<b>W</b>	= 0 σταθερά
<b>xyz</b>	= η τιμή προτοποθέτησης. Οι μηδενικές τιμές των ψηφίων xyz δεν παραλείπονται

**Χρήση του format: W # 16 # wxyz**

**L W # 16 # 0024**

**Χρήση του format: C # xyz**

**L C # 24**

## Απαριθμητές

---

### *Εντολές προγραμματισμού των απαριθμητών*

- *ΕΝΤΟΛΗ R : RESET COUNTER*
- *ΕΝΤΟΛΗ L: LOAD CURRENT COUNTER VALUE INTO ACCU1*
- *Εντολή LC: LOAD CURRENT COUNTER VALUE INTO ACCU 1 AS BCD*
- *Εντολή CU: COUNTER UP*
- *Εντολή CD: COUNTER DOWN*

# Απαριθμητές

## Παράδειγμα στην STL

```
FC11 : Title:
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΤΗΝ STL ΕΝΟΣ ΑΠΑΡΙΘΜΗΤΗ ΜΕ ΔΥΝΑΤΟΤΗΤΑ ΜΕΤΡΗΣΗΣ ΠΡΟΣ ΤΑ ΕΠΑΝΩ
ΚΑΙ ΠΡΟΣ ΤΑ ΚΑΤΩ

Network 1: Title:
ΜΕΤΑΒΟΛΗ ΤΗΣ ΚΑΤΑΣΤΑΣΗΣ ΤΟΥ I0.0 ΑΠΟ "0" ΣΕ "1" ΚΑΙ ΕΦ' ΟΣΟΝ ΕΙΝΑΙ ΣΕ
ΚΑΤΑΣΤΑΣΗ "1" Η ΤΟ I0.1 Η ΤΟ I0.2 ΤΟΤΕ ΑΛΛΑΖΕΙ ΤΟ ΠΕΡΙΕΧΟΜΕΝΟ ΤΟΥ ΑΠΑΡΙΘΜΗΤΗ
ΠΡΟΣ ΤΑ ΕΠΑΝΩ Η ΠΡΟΣ ΤΑ ΚΑΤΩ ΑΝΤΙΣΤΟΙΧΑ

      A   I   0.0
FR    C    1

Network 2: Title:
ΜΕΤΑΒΟΛΗ ΤΗΣ ΚΑΤΑΣΤΑΣΗΣ ΤΟΥ I0.1 ΑΠΟ "0" ΣΕ "1" ΠΡΟΚΑΛΕΙ ΤΗΝ ΑΥΞΗΣΗ
ΚΑΤΑ ΜΙΑ ΜΟΝΑΔΑ ΤΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ ΤΟΥ ΑΠΑΡΙΘΜΗΤΗ

      A   I   0.1
CU    C    1

Network 3: Title:
ΜΕΤΑΒΟΛΗ ΤΗΣ ΚΑΤΑΣΤΑΣΗΣ ΤΟΥ I0.2 ΑΠΟ "0" ΣΕ "1" ΠΡΟΚΑΛΕΙ ΤΗΝ ΜΕΙΩΣΗ
ΚΑΤΑ ΜΙΑ ΜΟΝΑΔΑ ΤΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ ΤΟΥ ΑΠΑΡΙΘΜΗΤΗ

      A   I   0.2
CD    C    1

Network 4: Title:
ΜΕΤΑΒΟΛΗ ΤΗΣ ΚΑΤΑΣΤΑΣΗΣ ΤΟΥ I0.3 ΑΠΟ "0" ΣΕ "1" ΠΡΟΚΑΛΕΙ ΤΗΝ ΦΟΡΤΩΣΗ
ΤΗΣ ΤΙΜΗΣ ΠΡΟΤΟΠΟΘΕΤΗΣ ΣΤΟΝ ΑΠΑΡΙΘΜΗΤΗ

      A   I   0.3
L     C#10
S     C    1
```

```
Network 5: Title:
Η ΚΑΤΑΣΤΑΣΗΣ "1" ΤΟΥ I0.4 ΠΡΟΚΑΛΕΙ ΤΟΝ ΜΗΔΕΝΙΣΜΟ ΤΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ ΤΟΥ
ΑΠΑΡΙΘΜΗΤΗ

      A   I   0.4
R     C    1

Network 6: Title:
ΦΟΡΤΩΝΟΥΜΕ ΣΤΟΝ ACCU-1 ΣΕ ΔΥΑΔΙΚΗ ΜΟΡΦΗ ΤΟ ΠΕΡΙΕΧΟΜΕΝΟ ΤΟΥ ΑΠΑΡΙΘΜΗΤΗ ΚΑΙ
ΤΟ ΜΕΤΑΦΕΡΟΥΜΕ ΣΤΗΝ MW80

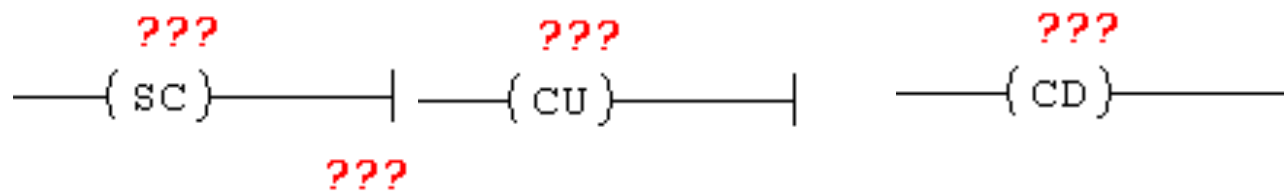
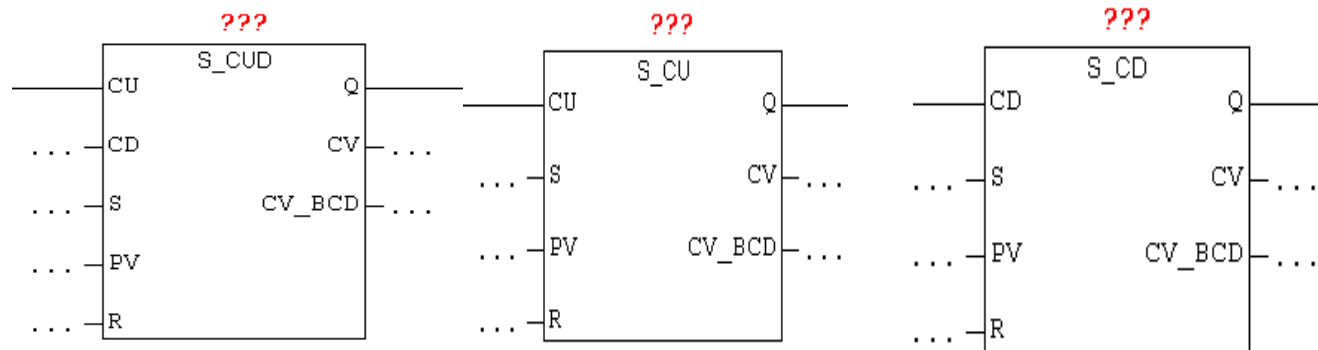
      L   C    1
T     MW   80

Network 7: Title:
ΦΟΡΤΩΝΟΥΜΕ ΣΤΟΝ ACCU-1 ΣΕ BCD ΚΩΔΙΚΑ ΤΟ ΠΕΡΙΕΧΟΜΕΝΟ ΤΟΥ ΑΠΑΡΙΘΜΗΤΗ ΚΑΙ
ΤΟ ΜΕΤΑΦΕΡΟΥΜΕ ΣΤΗΝ MW82

      LC  C    1
T     MW   82
```

# Απαριθμητές

## Απαριθμητές στην LAD



## ΥΠΟΕΝΟΤΗΤΑ 7.11

*Εντολές Ελέγχου της ροής του Προγράμματος*

## Εντολές Ελέγχου της ροής του Προγράμματος

---

- **Εντολές διακλάδωσης:** Ελέγχουν την ροή του προγράμματος μέσα σ' ένα μπλοκ.
- **Εντολές κλήσεις των μπλοκ:** Ελέγχουν την ροή του προγράμματος κάνοντας τα περάσματα από μπλοκ σε μπλοκ
- **Εντολές τέλους προγράμματος:** Για τον τερματισμό ενός μπλοκ προγραμματισμού

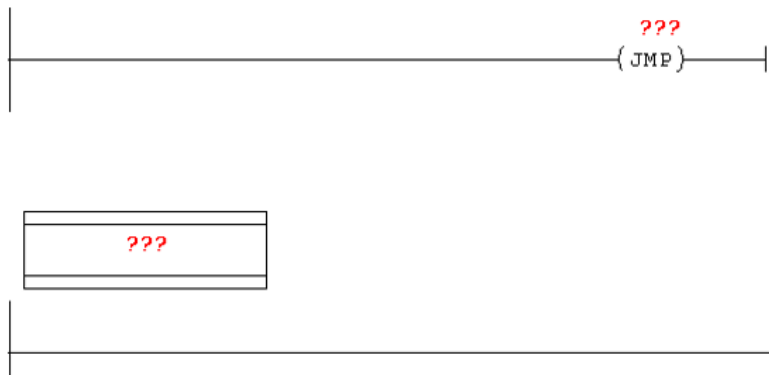
# Εντολές Ελέγχου της ροής του Προγράμματος

## Εντολές Διακλάδωσης

Διακλάδωση χωρίς συνθήκη: *JU Jump Unconditional*

- *JU <jump label>*

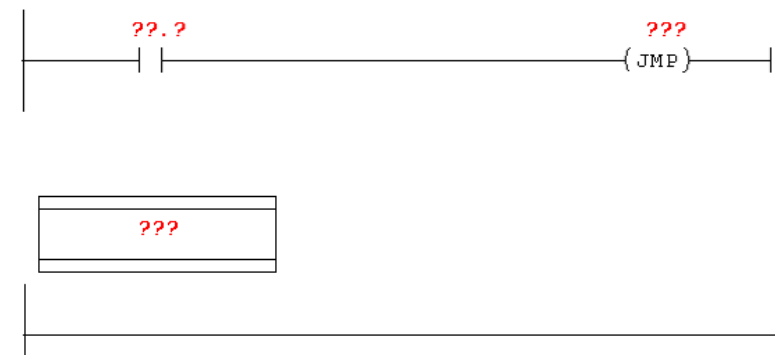
```
      L   15
      T   MW   60
      JU   FOR2
FOR1: L   35
      T   MW   60
FOR2: A   I    1.3
```



Διακλάδωση με συνθήκη (RLO = 1): *JC*

- *JC <Label>*

```
      A   I    1.1
      A   I    1.2
      JC  JOVR
      L   IW   8
      T   MW  22
JOVR: A   I    2.1
```





# Εντολές Ελέγχου της ροής του Προγράμματος

## Εντολές τερματισμού ενός Block

➤ **Εντολή τερματισμού BE:**

➤ **Εντολή τερματισμού BEU:**

➤ **Εντολή τερματισμού BEC :**

Με τις εντολές τερματισμού προγράμματος μας επιτρέπεται να τερματίσουμε το πρόγραμμα μέσα στο μπλοκ στο οποίο γράφουμε την εντολή. Το πρόγραμμα επιστρέφει στο μπλοκ εκείνο από το οποίο έγινε η κλήση. Σε περίπτωση χρήσης αυτών των εντολών πρέπει να ξέρουμε ότι:

Οι τοπικές μεταβλητές του μπλοκ μέσα στο οποίο γράφουμε την εντολή τερματισμού παύουν να ισχύουν και ισχύουν οι τοπικές μεταβλητές του μπλοκ στο οποίο επέστρεψε το πρόγραμμα μας.

Τα Data Block τα οποία ήταν ανοιχτά στο μπλοκ στο οποίο επιστέφουμε εξακολουθούν να ισχύουν.

Εντολή τερματισμού BE: το μπλοκ στο οποίο είναι γραμμένη τερματίζεται άσχετα από την κατάσταση του RLO.

Εντολή τερματισμού BEU: το μπλοκ στο οποίο είναι γραμμένη τερματίζεται άσχετα από την κατάσταση του RLO.

Στην Step 7 δεν υπάρχει καμία διαφορά μεταξύ των εντολών BE και BEU

Εντολή τερματισμού BEC :τερματίζουμε ένα μπλοκ εφόσον το RLO έχει κατάσταση 1. Εάν το RLO έχει την κατάσταση «0» η εντολή BEC δεν εκτελείται. Προσοχή όμως σ' αυτήν την περίπτωση, αμέσως μετά την εντολή BEC το RLO λαμβάνει κατάσταση «1».

## Εντολές Ελέγχου της ροής του Προγράμματος

---

### Εντολές κλήσης των Block

- ✓ UC : Κλήση ενός μπλοκ χωρίς συνθήκη
- ✓ CC : Κλήση ενός μπλοκ με συνθήκη
- ✓ CALL: Κλήση ενός μπλοκ χωρίς συνθήκη και απαραίτητη εντολή για παραμετροποιημένα μπλοκ



**ΥΠΟΕΝΟΤΗΤΑ 7.12**  
***Data Block***

## **Data Block**

---

*Τα Μπλοκ Δεδομένων δεν περιέχουν κώδικα και χρησιμοποιούνται από τα Μπλοκ Λογικής για την διαχείριση των πληροφοριών .*

*Όταν ένα μπλοκ Δεδομένων μεταφερθεί στην CPU τότε καταλαμβάνει χώρο στην μνήμη εργασίας της CPU .*

*Στις πληροφορίες ενός Μπλοκ Δεδομένων μπορούμε να έχουμε προσπέλαση με εντολές Bit , Byte , Word , Dword .Επίσης μπορούμε να χρησιμοποιήσουμε απόλυτη ή συμβολική διευθυνσιοδότηση .*

### **GLOBAL DATA BLOCK**

#### **ΕΞΑΡΤΗΜΕΝΑ DATA BLOCK**

1. **INSTANCE DATA BLOCK**
2. **UDT [ User Define Data Type ] DATA BLOCK**

## **Data Block**

---

*Ο αριθμός των Data Block που μπορούμε να δημιουργήσουμε σε ένα project και το μέγεθος της μνήμης του κάθε Data Block εξαρτάται από τα χαρακτηριστικά της CPU.*

*Για παράδειγμα η CPU 314 C-2 DP έχει τις εξής δυνατότητες:*

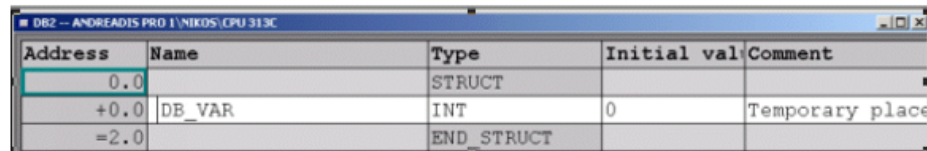
- Μέγιστος αριθμός : 127*
- Μέγιστη χωρητικότητα μνήμης : 16 kB για κάθε Data Block*

*Φυσικά θα πρέπει να ελέγξουμε και την χωρητικότητα της work memory που διαθέτει η CPU .*

# Data Block

Η διαδρομή για να δημιουργήσουμε ένα Data Block είναι από το κεντρικό menu του Simatic Manager :

1. **INSERT**
2. **S7 BLOCKS**
3. **DATA BLOCKS**

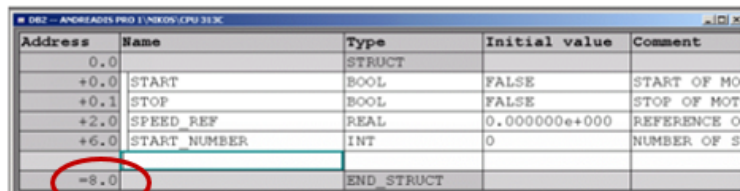


Address	Name	Type	Initial val	Comment
0.0		STRUCT		
+0.0	DB_VAR	INT	0	Temporary place
=2.0		END_STRUCT		

- **DATA VIEW**
- **DECLARATION VIEW**

# Data Block

- **ADDRESS**
- **NAME**
- **TYPE**
  - **ELEMENTARY TYPES**
  - **COMPLEX TYPES**
- **INITIAL VALUE**
  - *Κατάλληλο Format*
  - *Αυτόματη συμπλήρωση τιμής 0 από τον editor*
  - *Οι αρχικές τιμές δεν μεταφέρονται αυτόματα στην CPU*
- **COMMENT**



Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	START	BOOL	FALSE	START OF MO
+0.1	STOP	BOOL	FALSE	STOP OF MOT
+2.0	SPEED_REF	REAL	0.000000e+000	REFERENCE O
+6.0	START_NUMBER	INT	0	NUMBER OF S
+8.0		END_STRUCT		

Για να δούμε τις τρέχουσες τιμές :

- Από το View επιλέγουμε *Data View*
- Δημιουργείτε η στήλη *Actual Value*
- Ενεργοποιούμε το εργαλείο *Monitor*

## Data Block

---

**Ονοματολογία Μεταβλητών :** Η ονοματολογία των μεταβλητών που ορίζονται μέσα σε ένα *Data Block* εξαρτάται από :

*Το μέγεθος της μεταβλητής*

*Τον χρησιμοποιούμενο Register για να ανοίξουμε ένα Data Block*

ΜΕΤΑΒΛΗΤΗ ΜΕΓΕΘΟΥΣ	ΧΡΗΣΗ ΤΟΥ REGISTER B	ΧΡΗΣΗ ΤΟΥ REGISTER I
1 BIT	DBX <sub>a.b</sub>	DIX <sub>a.b</sub>
1 BYTE	DBB <sub>a</sub>	DIB <sub>a</sub>
1 WORD	DBW <sub>a</sub>	DIW <sub>a</sub>
1 DWORD	DBD <sub>a</sub>	DID <sub>a</sub>



## ***Data Block***

---

***Εντολές πρόσβασης στην STL :***

<b>ΜΕΤΑΒΛΗΤΗ ΜΕΓΕΘΟΥΣ</b>	<b>ΧΡΗΣΗ ΤΟΥ REGISTER B</b>	<b>ΧΡΗΣΗ ΤΟΥ REGISTER I</b>
<b>1 BIT</b>	<b>A DBXa.b</b>	<b>A DIXa.b</b>
<b>1 BYTE</b>	<b>L DBBa</b>	<b>L DIBa</b>
<b>1 WORD</b>	<b>L DBWa</b>	<b>L DIWa</b>
<b>1 DWORD</b>	<b>L DBDa</b>	<b>L DIDa</b>

<b>ΜΕΤΑΒΛΗΤΗ ΜΕΓΕΘΟΥΣ</b>	<b>ΧΡΗΣΗ ΤΟΥ REGISTER B</b>	<b>ΧΡΗΣΗ ΤΟΥ REGISTER I</b>
<b>1 BIT</b>	<b>= DBXa.b</b>	<b>= DIXa.b</b>
<b>1 BYTE</b>	<b>T DBBa</b>	<b>T DIBa</b>
<b>1 WORD</b>	<b>T DBWa</b>	<b>T DIWa</b>
<b>1 DWORD</b>	<b>T DBDa</b>	<b>T DIDa</b>

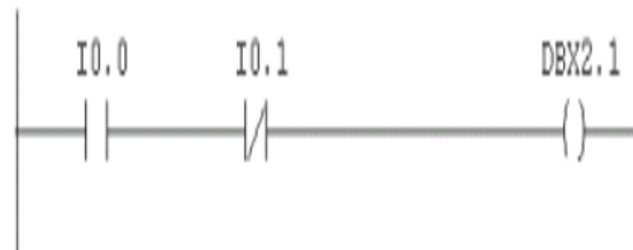
# Data Block

## Εντολές πρόσβασης στην LAD :

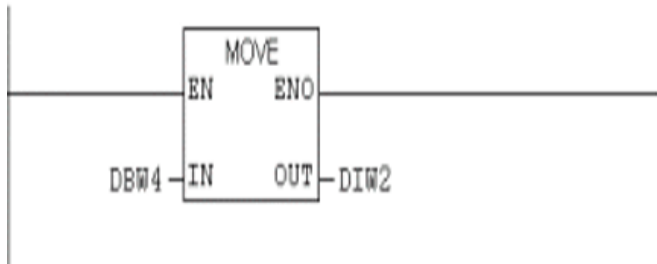


*Διάβασμα μεταβλητής μεγέθους Bit & χρησιμοποίηση του Register B*

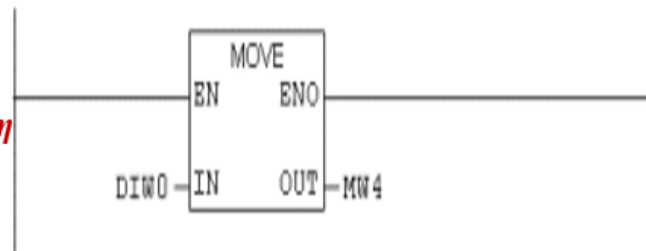
*Γράψιμο μεταβλητής μεγέθους Bit & χρησιμοποίηση του Register B*



*Διάβασμα μεταβλητής μεγέθους Word & χρησιμοποίηση του Register B . Γράψιμο μεταβλητής μεγέθους Word & χρησιμοποίηση του Register I.*



*Διάβασμα μεταβλητής μεγέθους Word & χρησιμοποίηση του Register I . Μεταφορά στην περιοχή των Merker .*



## Data Block

### Σύνθετες εντολές πρόσβασης στην STL :

Διάβασμα μεταβλητής μεγέθους Bit :

*OPN*    *DB*    *1*    } *A*    *DB1.DBX 0.0*  
*A*        *DBX* *0.0*

Γράψιμο μεταβλητής μεγέθους Bit :

*OPN*    *DB*    *1*    } =    *DB1.DBX 0.1*  
=        *DBX* *0.1*

Διάβασμα μεταβλητής μεγέθους Word:

*OPN*    *DB*    *1*    } *L*    *DB1.DBW 4*  
*L*        *DBW* *4*

Γράψιμο μεταβλητής μεγέθους Word :

*OPN*    *DB*    *5*    } *T*    *DB5.DBW 4*  
*T*        *DBW* *4*

# Data Block

## Σύνθετες εντολές πρόσβασης στην LAD

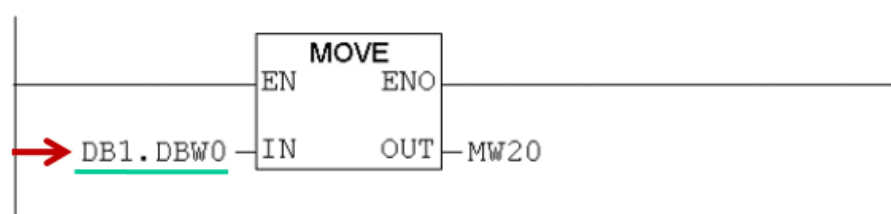
Διάβασμα μεταβλητής μεγέθους Bit :



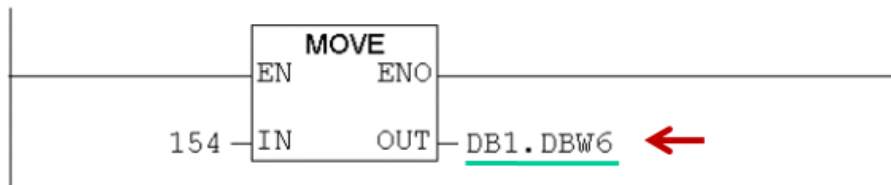
Γράψιμο μεταβλητής μεγέθους Bit :



Διάβασμα μεταβλητής μεγέθους Word:



Γράψιμο μεταβλητής μεγέθους Word :



## Data Block

---

### *Ισχύς ενός ανοικτού Data Block*

- Ένα Data Block παραμένει ανοικτό έως ότου δεν ανοίξει κάποιο άλλο Data Block με τον ίδιο register.
- Όταν από ένα OB ή από ένα FC καλέσουμε κάποιο άλλο FC το Data Block το οποίο ήταν ανοιγμένο στο μπλοκ κλήσης παραμένει ανοιγμένο και στο νέο μπλοκ το οποίο κλήθηκε . Και όχι μόνο αλλά και επιστρέφοντας στο αρχικό μπλοκ αυτό ισχύει ακόμη
- Ο προηγούμενος κανόνας δεν ισχύει εάν το μπλοκ που καλείται είναι FB
- Προσοχή όταν μέσα σε ένα παραμετροποιημένο FB κάνουμε χρήση του register I .

## Data Block

### Τύποι Δεδομένων μεταβλητών ενός Data Block

ΛΕΞΗ-ΚΛΕΙΔΙ	ΜΗΚΟΣ ΣΕ BIT	ΠΑΡΑΔΕΙΓΜΑ
BOOL	1	1 n 0 / TRU n FALSE
BYTE	8	B#16#A9
WORD	16	W#16#12AF
DWORD	32	DW#16#ADAC1EF5
CHAR	8	'W'
S5 TIME	16	S5T#5s_200ms n 5s200ms
INT	16	123
DINT	32	65539
REAL	32	1.2 n 1.200000e+000
TIME	32	T#2D_1H_3M_45S_12MS
DATE	16	D#1993-01-20
TIME_OF_DAY	32	TOD#12:23:45:12

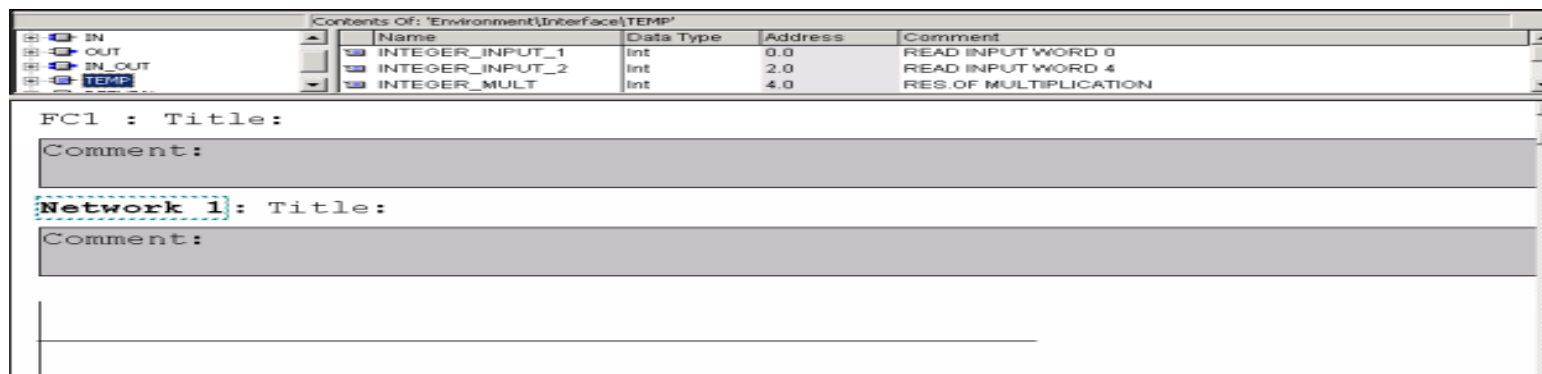


**ΥΠΟΕΝΟΤΗΤΑ 7.13**  
***Παραμετροποιημένα Block***

## Παραμετροποιημένα Block

### Τοπικές Μεταβλητές «L»

- Τις προσωρινές μεταβλητές τις χρησιμοποιούμε για να αποθηκεύσουμε προσωρινά πληροφορίες που χρειαζόμαστε σ' ένα μπλοκ προγραμματισμού.
- Οι προσωρινές μεταβλητές είναι μεταβλητές οι οποίες αποθηκεύονται στην περιοχή μνήμης L – Stack και μπορούν να χρησιμοποιηθούν σε όλα τα ήδη των μπλοκ (OB, FC, FB) που περιέχουν λογικές εντολές.
- Οι προσωρινές μεταβλητές που χρησιμοποιούνται μέσα σ' ένα μπλοκ πρέπει να οριστούν στον πίνακα δηλώσεων αυτού του μπλοκ στην γραμμή «temp».





## Παραμετροποιημένα Block

---

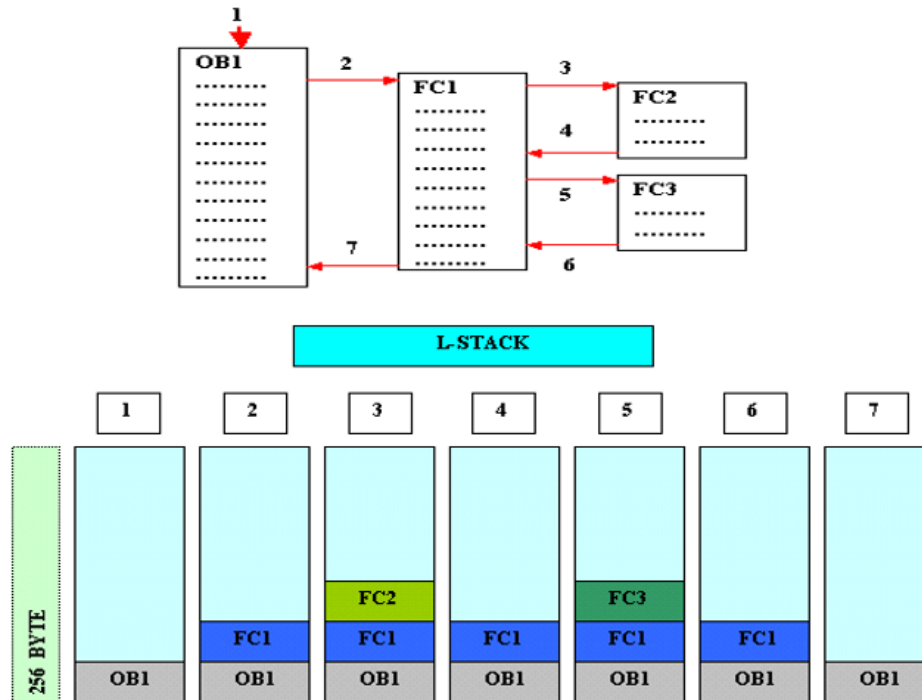
### Τοπικές Μεταβλητές «L»

**L –STACK** : Διαθέτουμε έξη διαφορετικές περιοχές μνήμης L – STACK , οι οποίες ορίζονται ανάλογα την κατηγορία του OB που είναι ενεργοποιημένο

- |                  |  |
|------------------|--|
| <b>1 ΠΕΡΙΟΧΗ</b> | <b>Αναφέρεται όταν είναι ανοιγμένο το OB1</b>  |
| <b>2 ΠΕΡΙΟΧΗ</b> | <b>Αναφέρεται όταν είναι ανοιγμένο κάποιο περιοδικό OB</b>   |
| <b>3 ΠΕΡΙΟΧΗ</b> | <b>Αναφέρεται όταν είναι ανοιγμένο κάποιο OB εκκίνησης</b>   |
| <b>4 ΠΕΡΙΟΧΗ</b> | <b>Αναφέρεται όταν είναι ανοιγμένο κάποιο OB που αναγνωρίζει σύγχρονα σφάλματα (Σφάλμα οφειλόμενο σε προγραμματισμό)</b> |
| <b>5 ΠΕΡΙΟΧΗ</b> | <b>Αναφέρεται όταν είναι ανοιγμένο κάποιο OB που αναγνωρίζει ασύγχρονο σφάλμα (Σφάλμα οφειλόμενο στο Hardware)</b>       |
| <b>6 ΠΕΡΙΟΧΗ</b> | <b>Αναφέρεται σε ότι αφορά τα OB επικοινωνίας</b>  |

# Παραμετροποιημένα Block

## Τοπικές Μεταβλητές «L»



Όταν σε κάποιο μπλοκ του project μας θέλουμε να χρησιμοποιήσουμε τοπικές μεταβλητές τότε στο πρόγραμμα του μπλοκ πρέπει πρώτα να ορίσουμε τις μεταβλητές αυτές και μετά να τις χρησιμοποιήσουμε

## Παραμετροποιημένα Block

---

### Παραμετροποιημένα Block

Παραμετροποιημένα μπλοκ χρησιμοποιούμε όταν η ίδια λειτουργία ενός συστήματος αυτοματισμού χρησιμοποιείται σε περισσότερα τμήματα της εγκατάστασης μας

Η χρησιμοποίηση παραμετροποιημένων μπλοκ μας δίνει τα εξής πλεονεκτήματα:

- - Το πρόγραμμα δημιουργείται μόνο μια φορά
- - Το παραμετροποιημένο μπλοκ αποθηκεύεται στην μνήμη μόνο μια φορά αλλά μπορούμε να το καλέσουμε και να τρέξουμε το πρόγραμμα που περιέχει για διαφορετικά τμήματα όσο συχνά θέλουμε.
- - Σ' ένα παραμετροποιημένο μπλοκ δίνουμε πραγματικές διευθύνσεις μόνο κατά την στιγμή κατά την οποία το καλούμε.

Η εντολή με την οποία καλούμε ένα παραμετροποιημένο μπλοκ είναι η εντολή : **CALL**

# Παραμετροποιημένα Block

## Παραμετροποιημένα FC

- ✓ ΠΑΡΑΜΕΤΡΟΥΣ
- ✓ TEMPORARY DATA
- ✓ RETURN

## ΠΑΡΑΜΕΤΡΟΥΣ

- ✓ ΠΑΡΑΜΕΤΡΟΥΣ **IN**
- ✓ ΠΑΡΑΜΕΤΡΟΥΣ **OUT**
- ✓ ΠΑΡΑΜΕΤΡΟΙ **IN/OUT**

Contents Of: 'Environment\Interface'	
	Name
[-] Interface	IN
[-] IN	OUT
[-] OUT	IN_OUT
[-] IN_OUT	TEMP
[-] TEMP	RETURN
[-] RETURN	

## Παραμετροποιημένα Block

---

### Παραμετροποιημένα FC - Άσκηση

Στην εγκατάσταση μας έχουμε μια σειρά από σφάλματα τα οποία όταν δημιουργούνται θέλουμε να έχουμε για κάθε σφάλμα μια οπτική προειδοποίηση.

Η οπτική προειδοποίηση θέλουμε να είναι ως εξής.

- ✓ Όταν έρθει ένα οποιοδήποτε σφάλμα θέλουμε το αντίστοιχο Led να αναβοσβήνει.
- ✓ Όταν ο χειριστής πατήσει το μπουτόν «Αναγνώριση Σφαλμάτων» που αντιστοιχεί στο συγκεκριμένο σφάλμα τότε :
  - ✓ - Εάν το σφάλμα παραμένει το αντίστοιχο LED να ανάβει μόνιμα έως ότου το σφάλμα απαλειφθεί, οπότε και θα σβήσει.
  - ✓ - Εάν το σφάλμα έχει ήδη απαλειφθεί το LED να σβήσει μόλις πατήσουμε το μπουτόν αναγνώριση σφάλματος.

# Παραμετροποιημένα Block

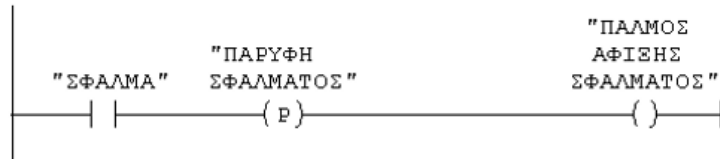
## Παραμετροποιημένα FC – Άσκηση: Το κύκλωμα

FC2 : ΠΑΡΑΔΕΙΓΜΑ ΣΗΜΑΤΟΔΟΤΗΣΗΣ ΣΦΑΛΜΑΤΩΝ

Comment:

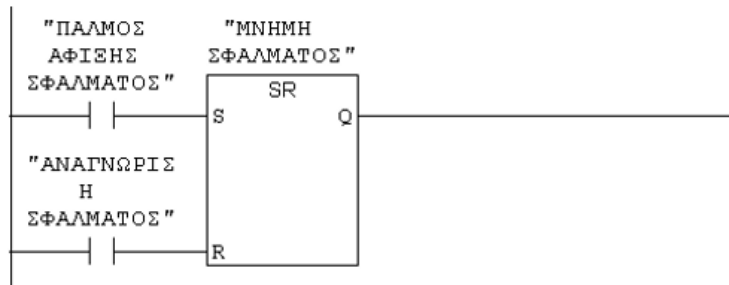
Network 1 : ΑΦΙΕΣΗ ΣΦΑΛΜΑΤΟΣ

Comment:



Network 2 : ΑΠΟΘΗΚΕΥΣΗ ΠΛΗΡΟΦΟΡΙΑΣ ΑΦΙΕΣΗΣ ΣΦΑΛΜΑΤΟΣ ΚΑΙ ΑΝΑΓΝΩΡΙΣΗ ΣΦΑΛΜΑΤΟΣ

Comment:

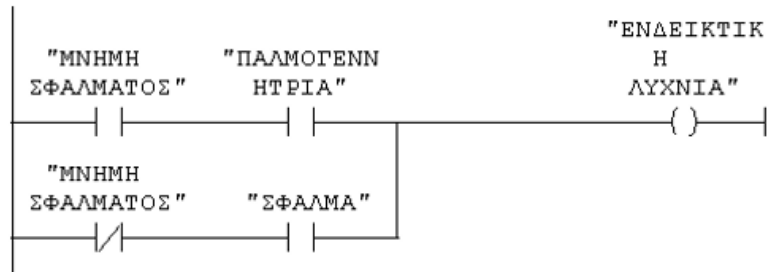


# Παραμετροποιημένα Block

## Παραμετροποιημένα FC – Άσκηση: Το κύκλωμα

Network 3 : ΣΗΜΑΤΟΔΟΤΗΣΗ ΣΦΑΛΜΑΤΟΣ

Comment:



# Παραμετροποιημένα Block

## Παραμετροποιημένα FC – Άσκηση : Ο πίνακας δηλώσεων

Contents Of: 'Environment\Interface\IN'

Name	Data Type	Comment
ERROR	Bool	ΣΦΑΛΜΑ
ACKN_ERROR	Bool	ΑΝΑΓΝΩΡΙΣΗ ΣΦΑΛΜΑΤΟΣ
FLASHER	Bool	ΠΑΛΜΟΓΕΝΝΗΤΡΙΑ

Contents Of: 'Environment\Interface\OUT'

Name	Data Type	Comment
DISPLAY	Bool	ΕΝΔΕΙΚΤΙΚΗ ΑΥΧΝΙΑ

Contents Of: 'Environment\Interface\IN\_OUT'

Name	Data Type	Comment
ERROR_MEMORY	Bool	ΜΝΗΜΗ ΣΦΑΛΜΑΤΟΣ
ADGE_ERROR	Bool	ΠΑΡΥΦΗ ΣΦΑΛΜΑΤΟΣ

Contents Of: 'Environment\Interface\TEMP'

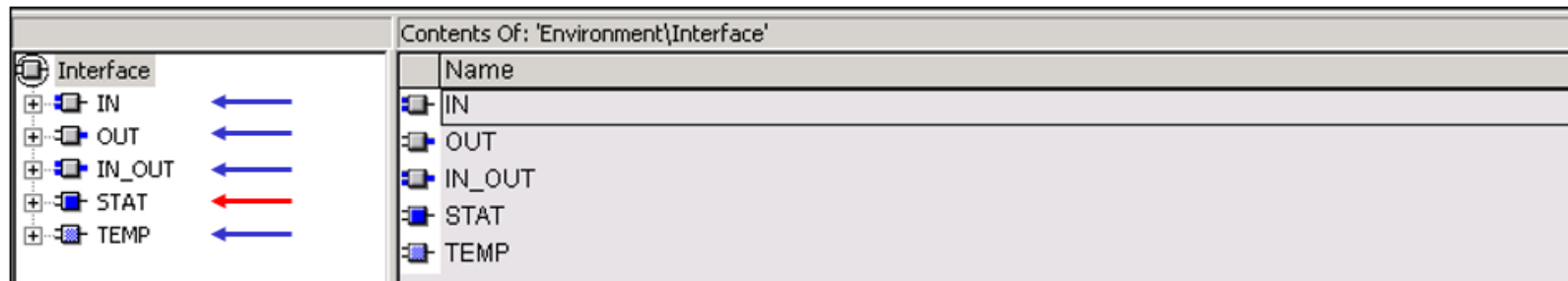
Name	Data Type	Address	Comment
PULSE	Bool	0.0	ΠΑΛΜΟΣ ΑΦΙΣΗΣ ΣΦΑΛΜΑΤΟΣ



## Παραμετροποιημένα Block

### Παραμετροποιημένα FB

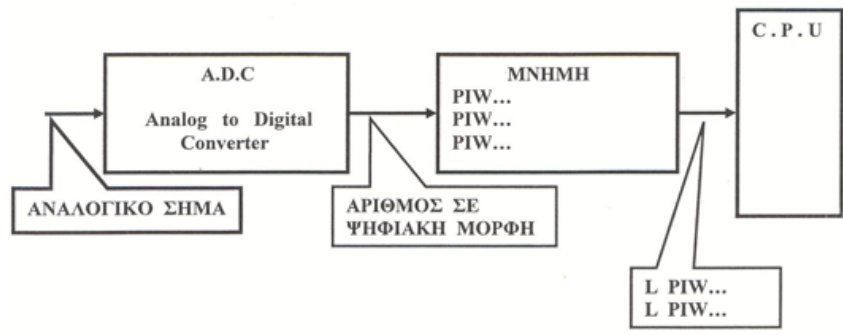
Η διαφορά μεταξύ ενός παραμετροποιημένου FC και ενός παραμετροποιημένου FB είναι ότι το παραμετροποιημένο FB διαθέτει μνήμη.



```
CALL FB1, DB2
```

**ΥΠΟΕΝΟΤΗΤΑ 7.14**  
***Επεξεργασία Αναλογικών σημάτων***

# Επεξεργασία Αναλογικών σημάτων

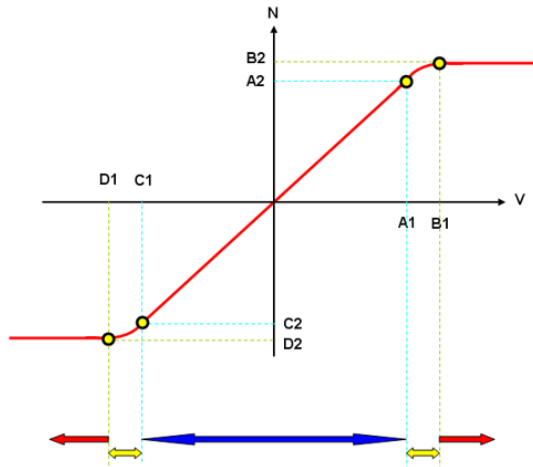


**L PIW...**



**T PQW...**

# Επεξεργασία Αναλογικών σημάτων



- **RATED RANGE**
- **OVER RANGE**
- **OVERFLOW**

ΠΕΡΙΟΧΗ	ΤΑΣΗ π.χ:		ΕΝΤΑΣΗ π.χ:	
	Περιοχή +/- 10V	Μονάδες	Περιοχή 4..20mA	Μονάδες
<b>OVERFLOW</b>	$\geq 11.759$	32767	$\geq 22.815$	32767
<b>OVER RANGE</b>	11.7589 . . 10.0004	32511 . . 27649	22.810 . . 20.0005	32511 . . 27649
<b>RATED RANGE</b>	10.00 7.5 . . - 7.5 -10.00	27648 20736 . . -20736 -27648	20.00 16.5 . . . 4.000	27648 20736 . . . 0.000
<b>UNDER RANGE</b>	-10.0004 . . -11.759	-27649 . . -32512	3.9995 . . 1.1852	-1 . . -4864
<b>UNDERFLOW</b>	$\leq -11.76$	-32768	$\leq 1.1845$	-32768

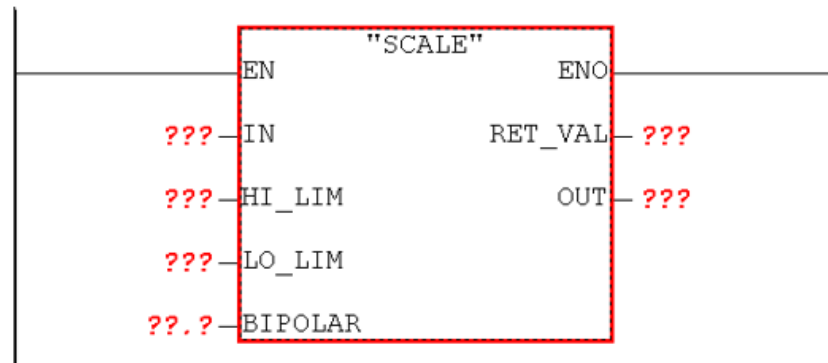
## Επεξεργασία Αναλογικών σημάτων

### Διευθύνσεις Αναλογικών σημάτων

Rack 3	Τροφοδοτικό	IM (Receive)	640	656	672	688	704	720	736	752	
			έως 654	έως 670	έως 686	έως 702	έως 718	έως 734	έως 750	έως 766	
Rack 2	Τροφοδοτικό	IM (Receive)	512	528	544	560	576	592	608	624	
			έως 526	έως 542	έως 558	έως 574	έως 590	έως 606	έως 622	έως 638	
Rack 1	Τροφοδοτικό	IM (Receive)	384	400	416	432	448	464	480	496	
			έως 398	έως 414	έως 430	έως 446	έως 462	έως 478	έως 494	έως 510	
R 0	Τροφοδοτικό	CPU	IM (Send)	256	272	288	304	320	336	352	368
				έως 270	έως 286	έως 302	έως 318	έως 334	έως 350	έως 366	έως 382
Θέση	2	3	4	5	6	7	8	9	10	11	

## Επεξεργασία Αναλογικών σημάτων

### Χρήση του FC 105

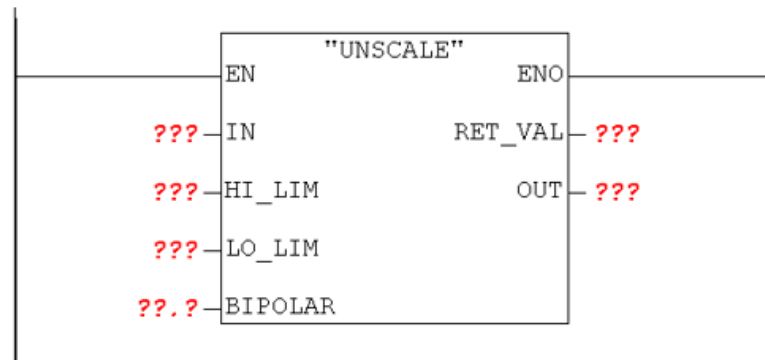


```
CALL "SCALE"
  IN      :=
  HI_LIM :=
  LO_LIM :=
  BIPOLAR:=
  RET_VAL:=
  OUT     :=
```

ΠΑΡΑΜΕΤΡΟΣ	ΤΙ ΣΗΜΑΙΝΕΙ
<b>EN</b>	Μόνο στην ladder εάν θέλουμε το FC 105 να εκτελείται υπό συνθήκη η όχι
<b>IN</b>	Γράφουμε την διεύθυνση της αναλογικής εισόδου που θέλουμε να κλιμακοποιήσουμε π.χ PIW 752
<b>HI-LIM</b>	Γράφουμε το πάνω όριο του φυσικού μεγέθους στο οποίο θέλουμε να αντιστοιχήσουμε τον ψηφιακό αριθμό +27648 της αναλογικής εισόδου σε real μορφή .
<b>LO-LIM</b>	Γράφουμε το κάτω όριο του φυσικού μεγέθους στο οποίο θέλουμε να αντιστοιχήσουμε τον ψηφιακό αριθμό 0 η -27648 της αναλογικής εισόδου σε real μορφή .
<b>BIPOLAR</b>	Η είσοδος bipolar καθορίζει εάν θα πρέπει να μετατρέπονται ακόμη και οι αρνητικοί αριθμοί η όχι. Στην ουσία εδώ καθορίζουμε εάν το σήμα μας είναι μονοπολικό η διπολικό . Εάν η κατάσταση του merker είναι «0» τότε το σήμα μας είναι μονοπολικό , εάν η κατάσταση του merker είναι «1» τότε το σήμα μας είναι διπολικό
<b>ENO</b>	Μας δίνει την κατάσταση της εισόδου EN
<b>RET_VAL</b>	Η έξοδος RET VAL έχει την τιμή «0» όταν η εκτέλεση του μπλοκ γίνεται χωρίς σφάλματα στην έξοδο αυτή δίνουμε την διεύθυνση μιας MW ...όπου αποθηκεύεται η τιμή RET VAL
<b>OUT</b>	Εδώ δίνουμε την διεύθυνση μιας DOUBLE WORD όπου θα αποθηκεύεται η κλιμακοποιημένη τιμή της αναλογικής εισόδου . Ο χώρος μνήμης πρέπει να είναι DOUBLE WORD διότι η νέα τιμή είναι σε real μορφή.

## Επεξεργασία Αναλογικών σημάτων

### Χρήση του FC 106



```
CALL "UNSCALE"
  IN      :=
  HI_LIM :=
  LO_LIM :=
  BIPOLAR :=
  RET_VAL :=
  OUT     :=
```

ΠΑΡΑΜΕΤΡΟΣ	ΤΙ ΣΗΜΑΙΝΕΙ
<b>EN</b>	Μόνο στην ladder εάν θέλουμε το FC 106 να εκτελείται υπό συνθήκη η όχι
<b>IN</b>	Δίνουμε τον real αριθμό που θέλουμε να μετατρέψουμε σε ψηφιακή μορφή . Εδώ δίνεται διεύθυνση DOUBLE WORD που περιέχει τον real αριθμό. Π.χ MD 40
<b>HI-LIM</b>	Γράφουμε το όριο του real αριθμού στο οποίο θέλουμε να αντιστοιχήσουμε τον ψηφιακό αριθμό +27648
<b>LO-LIM</b>	Γράφουμε το όριο του real αριθμού στο οποίο θέλουμε να αντιστοιχήσουμε τον ψηφιακό αριθμό 0 η -27648
<b>BIPOLAR</b>	Η είσοδος bipolar καθορίζει εάν θα πρέπει να μετατρέπονται ακόμη και αρνητικοί αριθμοί η όχι. Στην ουσία εδώ καθορίζουμε εάν το σήμα μας είναι μονοπολικό η διπολικό . Εάν η κατάσταση του merker είναι «0» τότε το σήμα μας είναι μονοπολικό , εάν η κατάσταση του merker είναι «1» τότε το σήμα μας είναι διπολικό
<b>ENO</b>	Μας δίνει την κατάσταση της εισόδου EN
<b>RET_VAL</b>	Η έξοδος RET VAL έχει την τιμή «0» όταν η εκτέλεση του μπλοκ γίνεται χωρίς σφάλματα στην έξοδο αυτή δίνουμε την διεύθυνση μιας MW ...όπου αποθηκεύεται η τιμή RET VAL
<b>OUT</b>	Εδώ δίνουμε την διεύθυνση μιας αναλογικής εξόδου η την διεύθυνση μιας word όπου θα αποθηκεύεται η αποκλιμακοποιημένη τιμή

## *Εκπαιδευτική Δραστηριότητα Δ3: «Άσκηση»*

---

**Τίτλος: Παράδειγμα προγραμματισμού κυκλώματος**



## Σύνοψη Θεματικής Ενότητας

Ερωτήσεις αυτοαξιολόγησης