

# ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΩΝ ΚΕΝΤΡΙΚΟΥ ΕΛΕΓΧΟΥ ΕΓΚΑΤΑΣΤΑΣΕΩΝ (ΤΗΛΕΕΛΕΓΧΟΣ-ΤΗΛΕΧΕΙΡΙΣΜΟΣ)

Τα βήματα για να γίνει μια ανάπτυξη ενός συστήματος κεντρικού ελέγχου είναι τα παρακάτω :

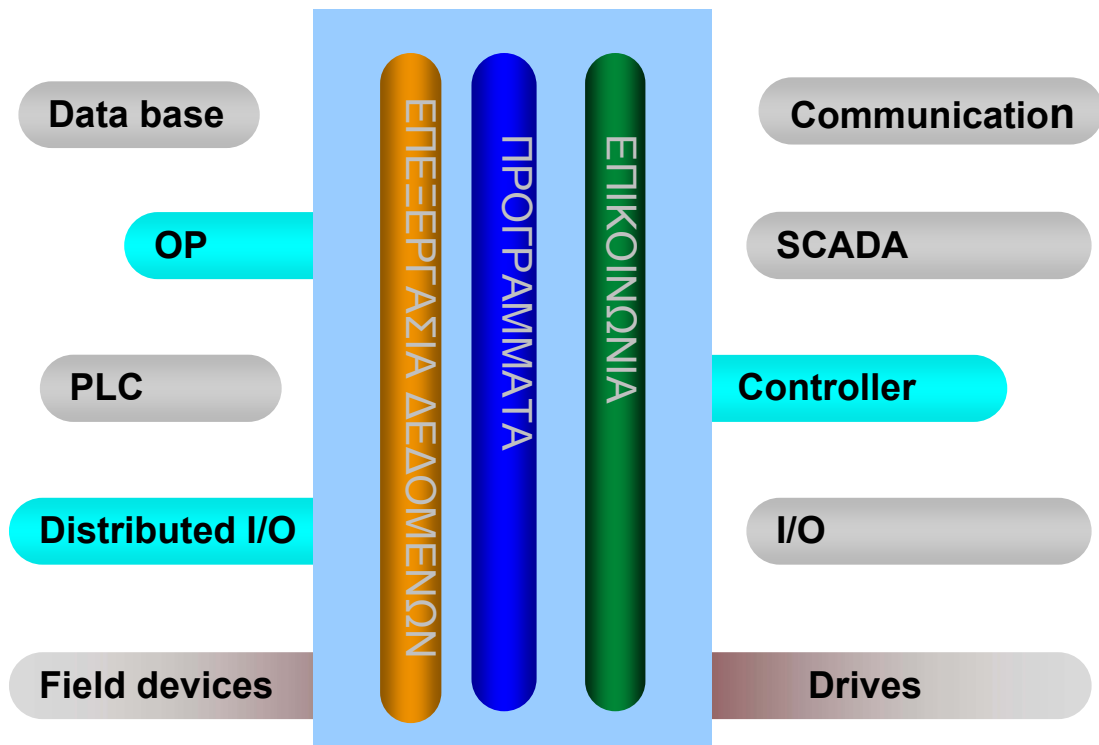
## Περιγραφή λειτουργίας της εγκατάστασης

Κατανοούμε την λειτουργία του συστήματος έτσι ώστε να συμπεράνουμε όλες τις διαδικασίες, και τις απαιτήσεις που έχει για να εξασφαλίσουμε την καλύτερη επιλογή συνθηκών λειτουργίας της εγκατάστασης.

## Επιλογή των στοιχείων της εγκατάστασης

Από την περιγραφή της λειτουργίας επιλέγουμε τα στοιχεία που θα χρησιμοποιήσουμε για να λειτουργήσουν οι διάφορες διαδικασίες (κινητήρες, βάνες, έμβολα κλπ). Στην συνέχεια επιλέγουμε τους πίνακες με τα απαραίτητα υλικά που θα μας οδηγούν τα παραπάνω στοιχεία.

## Σχεδίαση της αρχιτεκτονικής δομής της εγκατάστασης



Το επόμενο βήμα είναι να σχεδιάσουμε την αρχιτεκτονική δομή της εγκατάστασης των οποίων τα βήματα είναι τα εξής:

- Επιλέγουμε τα υλικά οδήγησης και σχεδιάζουμε τις απαιτούμενες καλωδιώσεις για την λειτουργία των στοιχείων.
- Από τα στοιχεία υπολογίζουμε τι πληροφορίες μπορούμε να έχουμε από αυτά, καθώς και τι εντολές ελέγχου χρειάζονται για να έχουμε τους επιθυμητούς αυτοματισμούς. Πολύ σημαντικό είναι να γνωρίζουμε εάν τα στοιχεία βρίσκονται σε κοντινές αποστάσεις μεταξύ τους ή όχι, έτσι ώστε να μπορέσουμε να κάνουμε την σωστότερη οικονομικοτεχνική επιλογή των PLC.
- Επιλέγουμε PLC με τα απαιτούμενα τεχνικά χαρακτηριστικά της εγκατάστασης δηλαδή κατάλληλο πλαίσιο στο οποίο τοποθετούνται το τροφοδοτικό , η CPU, κάρτες I/O, κάρτες επικοινωνίας και ότι άλλη κάρτα απαιτεί η εγκατάστασή μας.
- Επιλέγουμε το είδος της επικοινωνίας που θέλουμε να έχουμε μεταξύ της εγκατάστασης και των χειριστών ή υπευθύνων (εργοδηγών, προϊστάμενων, διεύθυνσης κλπ.) Στην επιλογή αυτή μπορούμε να έχουμε από ένα απλό χειριστήριο με μπουτόν και λυχνίες μέχρι υπολογιστικά συστήματα με βάσεις δεδομένων και τηλεέλεγχο –τηλεχειρισμό. Για να έχουμε σε ένα σύστημα

απαιτήσεις τηλεελέγχου – τηλεχειρισμού τότε θα πρέπει να υπάρχει σε δικτυακά περιβάλλοντα έναν υπολογιστή που να κρατάει τα δεδομένα πραγματικού χρόνου στην εσωτερική του Βάση Δεδομένων Πραγματικού Χρόνου (RDBMS). Αυτά τα δεδομένα να είναι δυνατόν να εμφανιστούν στην οθόνη των υπολογιστών από έναν ή περισσότερους clients ταυτόχρονα που συνδέονται με ίδιο υπολογιστή που φιλοξενεί τον server ή σε οποιοδήποτε άλλο υπολογιστή που συνδέεται με τον server με ένα δίκτυο NetBeUI ή TCP/IP.

- Επιλέγουμε ένα SCADA που να περιλαμβάνει πληθώρα έτοιμων λειτουργιών καθώς και προσκολλημένες λειτουργίες, να είναι ταυτόχρονα και τρομερά εύκολο στο χρήστη να το επεκτείνει ή να το τροποποιήσει σε πάρα πολύ μικρό χρονικό διάστημα, χωρίς να διακόπτει την συλλογή και καταγραφή των δεδομένων (Δεν χρειάζεται compilation μετά από οποιαδήποτε αλλαγή στο εργαλείο ανάπτυξης, παρά μόνο ένα απλό save). Με τη δυνατότητα αυτή δεν υπάρχει απώλεια στην αποθήκευση των δεδομένων της εγκατάστασης για όσο διάστημα γίνεται μετατροπή σε κάποια π.χ. εικόνα. Ο χειριστής να μαθαίνει το πρόγραμμα εφαρμογής σε ελάχιστο διάστημα στην πιο μικρή του λεπτομέρεια με τη βοήθεια των εργαλείων Tutorial και πιο ειδικά με την εφαρμογή CBT (Computer Based Training) και του μικρού βιβλίου "Getting Started". Με τη βοήθεια εργαλείων και με τη χρήση του ποντικιού να επιλέγει μέσα από ειδικές μάσκες το είδος της εργασίας που θέλει να κάνει (π.χ. δημιουργία ολόκληρου project με εικόνες, καμπύλες, μπάρες, πρωτόκολλα και μηνύματα). Από εδώ και πέρα, να αρχίζει μία διαλογική συζήτηση του χειριστή με το σύστημα, όπου αυτός απαντά σε ερωτήσεις που του τίθενται και έτσι κρυφά από αυτόν δημιουργείται αυτόματα η εφαρμογή. Τελειώνοντας την ανάπτυξη και με την βοήθεια του εργαλείου Simulator να μπορεί να δοκιμάσει την εφαρμογή του για την καλή της λειτουργία χωρίς να είναι υποχρεωτικά συνδεδεμένος με τους τοπικούς σταθμούς. Τα προγράμματα εφαρμογής, επιτελούν τη λειτουργία τηλεελέγχου/τηλεχειρισμού των Εγκαταστάσεων με τρόπο κατανοητό και χωρίς περιττές ενέργειες από το χρήστη που στη συγκεκριμένη περίπτωση δεν είναι απαραίτητο να είναι ειδικευμένος ή εκπαιδευμένος στη πληροφορική. Όλη η σύνθετη λειτουργία γίνεται μακριά απ' αυτόν, πίσω από τις οθόνες που αυτός βλέπει και χειρίζεται. Αυτός, με τη βοήθεια του ποντικιού του και μέσα από τις ιεραρχικά δομημένες οθόνες του υπολογιστή του μπορεί να επιλέγει τρόπο λειτουργίας (τοπικά ή κεντρικά), να ενεργοποιεί ή ν' απενεργοποιεί αντλίες, να ζητά εκτύπωση καμπυλών ή σφαλμάτων που καταγράφηκαν κατά τη λειτουργία της εγκατάστασης κ.α. Σε καμία περίπτωση δεν είναι απαραίτητη η απομνημόνευση κωδικών προγραμμάτων ή εντολών λειτουργικού συστήματος για να επιτελεσθεί κάποια εργασία στο σύστημα, πέραν από αυτές που έχουν να κάνουν με την ασφάλεια του συστήματος (κωδικοί πρόσβασης). Οι βασικότερες λειτουργίες που πρέπει να έχουμε με ένα SCADA είναι να αποθηκεύει τα παραγωγικά στοιχεία σε σύστημα βάσης δεδομένων όπου να γίνεται προσπέλαση από οποιαδήποτε front end εργαλείο που υποστηρίζει ODBC/SQL. Αρχικά οι μεταβλητές (Tags) να συλλέγονται με κύκλο < 500 ms μέσω του πακέτου Tag Logging, να αρχειοθετούνται κυκλικά, επιλεγμένα κυκλικά, μη κυκλικά, να συμπιέζονται σε αθροίσματα, μέσες τιμές, Min και Max τιμές, να παρουσιάζονται σε καμπύλες και πίνακες και αποθηκεύονται στην SQL βάση δεδομένων.

Στην **κυκλική αρχειοθέτηση** η συλλογή δεδομένων να αρχίζει με την εκκίνηση του συστήματος και να εκτελείται σε σταθερούς κύκλους χρόνου μέχρι την

παύση του συστήματος. Βασικά να μπορεί σε κάθε αρχείο να ρυθμισθεί ο κύκλος συλλογής και αρχειοθέτησης ανεξάρτητα ο ένας από τον άλλο. (Η ρύθμιση μπορεί να είναι από 1 sec έως 1 χρόνο. Οι κύκλοι χρόνου τοποθετούνται σταθερά για κάθε μεταβλητή ή για κάθε ομάδα μεταβλητών και οι τελευταίες στατικά και σταθερά αποθηκεύονται μέχρι την λήξη του χρόνου.)

Στην **επιλεγμένα κυκλικά αρχειοθέτηση** η αρχειοθέτηση δεδομένων να αρχίζει με την εμφάνιση κάποιου γεγονότος (σήμα εκκίνησης) και εκτελείται σε σταθερούς κύκλους χρόνου μέχρι την εμφάνιση δεύτερου γεγονότος (σήμα παύσης). Στο stop σήμα επιπρόσθετα να αποθηκεύεται και η τελευταία συλλεγόμενη μεταβλητή.

Γεγονότα Start-Stop μπορούν να είναι:

- Η αλλαγή κατάστασης μίας λογικής μεταβλητής/τιμής
- Η εμφάνιση μίας οριακής τιμής της μεταβλητής/τιμής
- Η Ώρα και η ημερομηνία
- Ο χειρισμός του πληκτρολογίου και του Mouse.
- Εντολή από άλλη εφαρμογή εκτός του Simatic WinCC
- Συνδυασμός όλων των παραπάνω δυνατοτήτων.

Στην **μη κυκλική αρχειοθέτηση** μία δυαδική η αναλογική μεταβλητή/τιμή αποθηκεύεται μόνο μία φορά με την εμφάνιση κάποιου γεγονότος.

Start ή Stop γεγονότα είναι:

- Η αλλαγή κατάστασης μίας δυαδικής μεταβλητής/τιμής (impulse διέγερση)
- Η εμφάνιση μίας οριακής τιμής μίας αναλογικής μεταβλητής/τιμής
- Συνδυασμός των παραπάνω δυνατοτήτων.
- Γεγονός που έχει παραχθεί σε σχέση με την ώρα και ημερομηνία.
- Τηλεγράφημα.

Τα γεγονότα φτιάχνονται με την βοήθεια του προγραμματισμού Action.

Στην μη κυκλική αρχειοθέτηση αποθηκεύεται πάντα η στιγμιαία τιμή.

Το SCADA πρέπει να προσφέρει authorization level έτσι ώστε να κλειδώνεται έναντι μη εξουσιοδοτημένης πρόσβασης.

Με την βοήθεια της User εξουσιοδότησης από τον User Administrator πού περιλαμβάνει το πακέτο να μπορούν να προστατευθούν από μη εξουσιοδοτημένη προσπέλαση π.χ.

α) αλλαγές και εισαγωγές setpoints , επιλογές εικόνων και μιμικών διαγραμμάτων, αλλαγή παραμέτρων συναγερμών και ελέγχων , βρόγχων ελέγχων και

β) τα αρχεία από εγγραφή ( όπως διαμόρφωση /μετατροπή κενών στοιχείων φύλλων εργασίας και σύνταξη αναφορών μέσω του Excell ή παρομοίου προγράμματος).

# Ελεγκτές Προγραμματιζόμενης Λογικής (PLC)

## Εισαγωγή

Η τεχνική του αυτοματισμού μέχρι πριν τρις δεκαετίες περίπου, βασιζόταν στο μεγαλύτερο μέρος της στα συστήματα συρματωμένης λογικής. Δηλαδή, η λειτουργία του αυτοματισμού στηριζόταν στην συρμάτωση ανάμεσα στα διάφορα στοιχεία (π.χ. επαφές, πηνία, ρελέ, χρονικά κ.λπ., για αυτοματισμούς με ρελέ, πύλες AND, πύλες OR κ.λπ., για αυτοματισμούς με ηλεκτρικές πλακέτες).

Επομένως, ένας πίνακας αυτοματισμού μπορούσε να κατασκευασθεί, μόνον όταν είχε τελειώσει οριστικά η μελέτη, δηλ. το συνδεσμολογικό σχέδιο. Επιπλέον, κάθε μετέπειτα αλλαγή είχε σαν αποτέλεσμα αντίστοιχη αλλαγή στην συρμάτωση (τροποποίηση του πίνακα), με τα γνωστά προβλήματα, π.χ. αν φτάνουν ή όχι οι ελεύθερες επαφές των ρελέ, αν υπάρχει χώρος για προσθήκη νέου υλικού κ.λπ.

Σήμερα, τα συστήματα **προγραμματιζόμενης λογικής**, με κύριους εκπρόσωπους στη βιομηχανία τους ελεγκτές προγραμματιζόμενης λογικής, έχουν πια εκτοπίσει τελείως τα προηγούμενα συστήματα αυτοματισμού.

Οι ελεγκτές προγραμματιζόμενης λογικής έκαναν την εμφάνισή τους στην αγορά στα τέλη του 1960, επιβαλλόμενοι τόσο από τις παραγωγικές διαδικασίες που απαιτούσαν όλο και πιο πολύπλοκη λογική στα συστήματα ελέγχου, όσο και από την πρόοδο στην βιομηχανία της ηλεκτρονικής που μπορούσε να προμηθεύσει τους κατασκευαστές με τα απαραίτητα στοιχεία (μνήμες, επεξεργαστές, A/D μετατροπείς, κτλ.).

Τα πρώτα PLC είχαν την δυνατότητα επεξεργασίας μόνο ψηφιακών σημάτων και βέβαια ο προγραμματισμός τους δεν ήταν, όπως συνηθίστε να λέμε, φιλικός στον χρήστη. Για παράδειγμα, η αποθήκευση του προγράμματος γινόταν σε κασέτες. Η δεκαετία του 70 ήταν η εποχή όπου τα PLC άρχιζαν να εγκαθίστανται δειλά - δειλά στην βιομηχανία, ενώ καθιέρωσή τους ήρθε την επόμενη δεκαετία (1980). Η παρουσία του προσωπικού υπολογιστή από το 1982 και μετά, από την μία μεριά προσέφερε απεριόριστες δυνατότητες και ευκολίες στην δημιουργία και τον έλεγχο προγραμμάτων, ενώ από την άλλη απείλησε τα PLC που κινδύνευαν προσωρινά από συστήματα ελέγχου με προσωπικούς υπολογιστές (PC based automation). Όμως το τοπίο ξεκαθάρισε οριστικά στα τέλη της δεκαετίας όπου τόσο τα PLC, όσο

και οι Η/Υ απέκτησαν την θέση τους στις παραγωγικές διαδικασίες με σαφές πλεονέκτημα υπέρ των PLC.

Στα συστήματα προγραμματιζόμενης λογικής, η κατασκευή και συρμάτωση του πίνακα είναι ανεξάρτητη από τη λειτουργία που πρόκειται να εκτελέσει ο αυτοματισμός, με άλλα λόγια η μελέτη δεν αποτελεί προϋπόθεση. Πάνω στις κλέμες του ελεγκτή συνδέονται όλα τα στοιχεία, που δίνουν εντολές (τερματικοί διακόπτες, μπουτόν κ.λπ.), καθώς και όλα τα στοιχεία που δέχονται εντολές (πηνία, ρελέ ισχύος κινητήρων, βαλβίδες, λυχνίες κ.λπ.).

Η λειτουργία του αυτοματισμού προγραμματίζεται στην μνήμη του ελεγκτή, ακόμα και την τελευταία στιγμή, πριν από τη θέση σε λειτουργία. Επομένως η μελέτη (πρόγραμμα) μπορεί να γίνεται παράλληλα με την επιλογή του υλικού και την κατασκευή του πίνακα.

Αν στην συνέχεια χρειαστεί να γίνουν αλλαγές στη λειτουργία, γεγονός σύνηθες στον αυτοματισμό, τότε αυτές γίνονται πολύ απλά «διορθώνοντας» το πρόγραμμα, χωρίς να χρειαστεί να επέμβουμε τη συρμάτωση του πίνακα.

Αυτή η ευελιξία στις μετατροπές και οι πολλές τους δυνατότητες αποτελούν το σπουδαιότερο πλεονέκτημα των ελεγκτών προγραμματιζόμενης λογικής στην τεχνική του αυτοματισμού.

Τέλος, από τις αρχές του 1990 έως σήμερα η ανάπτυξη των ελεγκτών προγραμματιζόμενης λογικής είναι αλματώδης και χαρακτηρίζεται από δίκτυα υψηλών ταχυτήτων, ταχύτατους και πανίσχυρους κεντρικούς επεξεργαστές.

## **1. Γενικά για τους ελεγκτές προγραμματιζόμενης λογικής**

### **1.1. Πλεονεκτήματα ελεγκτών προγραμματιζόμενης λογικής**

- Μεγιστοποιούν την ταχύτητα της διαδικασίας παραγωγής και κατά συνέπεια μειώνουν πολύ γρήγορα το χρόνο απόσβεσης της εγκατάστασης.
- Στο στάδιο της μελέτης δεν υπάρχει πρόβλημα αν επαρκούν ή όχι οι επαφές των ρελέ, των χρονικών ή των εξωτερικών τερματικών.
- Η λειτουργία του αυτοματισμού μπορεί να αλλάξει πολύ εύκολα σε οποιοδήποτε στάδιο (μελέτη, κατασκευή, θέση σε λειτουργία ή αργότερα).
- Ο εντοπισμός βλαβών διευκολύνεται γιατί για κάθε εξωτερική εντολή υπάρχει αντίστοιχο LED. Επίσης, η ροή του αυτοματισμού μπορεί να παρακολουθείται άνετα, με τη βοήθεια μιας συσκευής προγραμματισμού.
- Έχουν τη δυνατότητα επέκτασης. Δηλαδή αν προστεθεί κάποια καινούρια διαδικασία ή χρειάζεται κάποιος επιπλέον έλεγχος, μπορεί με την προσθήκη των κατάλληλων καρτών εισόδων / εξόδων (I/O), το ίδιο το PLC να την πραγματοποιήσει.
- Ο αυτοματισμός παραδίνεται συντομότερα σε λειτουργία, επειδή η μελέτη μπορεί να γίνεται παράλληλα με την τοποθέτηση και συρμάτωση του ελεγκτή.

- Δεν υπάρχει το γνωστό πρόβλημα των «μη ενημερωμένων» σχεδίων του πίνακα αυτοματισμού μετά από λίγο καιρό. Ο ελεγκτής έχει πάντα κρατημένο «μέσα του» το τελευταίο πρόγραμμα, το οποίο μπορεί να διαβαστεί με μία συσκευή προγραμματισμού ή να εκτυπωθεί σε χαρτί.
- Υπάρχει σημαντική οικονομία στο χώρο, τη συντήρηση (δεν υπάρχουν μηχανικές επαφές) και την κατανάλωση ενέργειας.
- Η τοποθέτηση μπορεί να γίνει χωρίς κίνδυνο και μέσα σε πεδία ισχύος.
- Ένας ελεγκτής μπορεί να συνδεθεί με περιφερειακές μονάδες για επιτήρηση - έλεγχο της εγκατάστασης (οθόνη, εκτυπωτής, πληκτρολόγιο), καταργώντας το κλασικό μιμητικό διάγραμμα και τον πίνακα χειρισμών. Επίσης, μπορεί να συνδεθεί με ηλεκτρονικό υπολογιστή για ανταλλαγή στοιχείων.
- Η γλώσσα προγραμματισμού είναι προσαρμοσμένη στο βιομηχανικό αυτοματισμό και άρα είναι προσιτή στο προσωπικό που μέχρι σήμερα συντηρούσε τους κλασσικούς πίνακες αυτοματισμού.
- Όλες οι κάρτες είναι τοποθετημένες στον ίδιο δίαυλο επικοινωνίας (bus), πράγμα που επιταχύνει την διαδικασία ανταλλαγής πληροφοριών.
- Τα PLC έχουν σχεδόν απεριόριστη διάρκεια ζωής λόγω των ασθενών ρευμάτων που χρησιμοποιούν. Ταυτόχρονα συμβάλουν στην οικονομικότερη, από πλευράς κατανάλωσης ηλεκτρικής ενέργειας, λειτουργία της εγκατάστασης.

## 1.2. Στάδια εργασίας στον προγραμματιζόμενο αυτοματισμό

Στην επίλυση ενός προβλήματος αυτοματισμού με ένα ελεγκτή συναντάμε τρία στάδια. Παρακάτω περιγράφονται τα στάδια αυτά, ενώ παράλληλα παρουσιάζεται η αντιστοιχία που έχουν με σύστημα συρματωμένης λογικής (π.χ. αυτοματισμός με ρελέ).

| Προγραμματιζόμενη λογική  | Συρματωμένη λογική  |
|---|---|
| 1. Επιλογή ελεγκτή προγραμματιζόμενης λογικής.  | 1. Επιλογή βοηθητικού ρελέ, χρονικών, κ.λπ. που χρειάζονται για τον αυτοματισμό, βάσει του συνδεσμολογικού σχεδίου. |
| 2. Τοποθέτηση ελεγκτή σε πίνακα.  | 2. Απλή τοποθέτηση του υλικού (βοηθητικά ρελέ, χρονικά) μέσα στον πίνακα.   |
| Συρμάτωση εξωτερικών στοιχείων της εγκατάστασης (τερματικοί, μπουτόν, διακόπτες, ενδ. λυχνίες, ρελέ ισχύος, βαλβίδες κ.λπ.), πάνω στις κλέμες του | Συρμάτωση των εξωτερικών στοιχείων της εγκατάστασης   |



ελεγκτή.

(τερματικοί, μπουτόν, διακόπτες, ενδ. λυχνίες, ρελέ ισχύος, βαλβίδες κ.λπ.) πάνω στην κλεμοσειρά του πίνακα.

3. Προγραμματισμός.

3. Συρμάτωση του υλικού αυτοματισμού (βοηθητικά ρελέ, χρονικά) μέσα στον πίνακα μεταξύ τους, σύμφωνα με το συνδεσμολογικό σχέδιο.

### **1.3. Δομή ελεγκτή προγραμματιζόμενης λογικής**

Η επιλογή ενός ελεγκτή προγραμματιζόμενης λογικής (τύπος - μέγεθος - κόστος) εξαρτάται κυρίως από το πλήθος των στοιχείων που δίνουν εντολή σ' αυτόν (είσοδοι) και το πλήθος των στοιχείων που δέχονται εντολή απ' αυτόν (έξοδοι), καθώς και από το πλήθος των λειτουργιών που απαιτείται να κάνει ο αυτοματισμός (μέγεθος προγράμματος, απαιτούμενη μνήμη και δυνατότητες της κεντρικής μονάδας).

Ανεξάρτητα, όμως, από τον τύπο και το μέγεθος, σε κάθε ελεγκτή συναντάμε τα εξής απαραίτητα στοιχεία:

- Πλαίσιο για τοποθέτηση των μονάδων
- Μονάδα τροφοδοσίας
- Κεντρική μονάδα (CPU) με τον μικροεπεξεργαστή και την μνήμη για το πρόγραμμα
- Μονάδες εισόδων
- Μονάδες εξόδων

#### **1.3.1 Πλαίσιο τοποθέτησης μονάδων**

Όλες οι μονάδες, από τις οποίες αποτελείται ένας ελεγκτής, πρέπει να τοποθετηθούν σε κάποιο πλαίσιο. Σ' αυτό είναι ενσωματωμένο το **σύστημα ζυγών**, δηλ. το σύστημα αγωγών, μέσω των οποίων επικοινωνούν οι διάφορες μονάδες μεταξύ τους για την ανταλλαγή πληροφοριών και για την τροφοδοσία τους.

##### Κεντρικό πλαίσιο (μόνο ένα)

Χαρακτηριστικό του είναι ότι μόνο σ' αυτό τοποθετείται η κεντρική μονάδα (CPU) και μάλιστα σε καθορισμένη θέση. Επίσης, εδώ τοποθετείται και η μονάδα τροφοδοσίας για ορισμένους τύπους ελεγκτών (π.χ. SIMATIC S7). Στις υπόλοιπες θέσεις του πλαισίου τοποθετούνται μονάδες εισόδων - εξόδων ή άλλες ειδικές μονάδες.

Σε μεγαλύτερα συστήματα ελεγκτών η μονάδα τροφοδοσίας δεν είναι σε μορφή κάρτας αλλά ερμαρίου και αποτελεί βασικό στοιχείο του κεντρικού πλαισίου.

### Πλαίσιο επέκτασης (ένα ή περισσότερα)

Σ' αυτό τοποθετούνται μόνο μονάδες εισόδων - εξόδων. Χρησιμοποιείται αν δεν φτάνουν οι θέσεις του κεντρικού πλαισίου, για να χωρέσουν όλες οι απαιτούμενες μονάδες.

Συνδέεται με το κεντρικό (ή με άλλα πλαίσια επέκτασης), μέσω ειδικής μονάδας διασύνδεσης και καλωδίου.

Συγκεκριμένα το πλαίσιο δημιουργείται τοποθετώντας όσες «μονάδες πλαισίου» απαιτούνται τη μία πλάι στην άλλη και συνδέοντάς τις. Κάθε τέτοια μονάδα πλαισίου έχει συγκεκριμένες θέσεις για τοποθέτηση μονάδων εισόδων - εξόδων.

### Χωροταξική διάταξη πλαισίων ελεγκτή

Όπως είπαμε προηγουμένως, ένας ελεγκτής αποτελείται από ένα κεντρικό πλαίσιο και τα απαιτούμενα πλαίσια επέκτασης.

Ανάλογα με την απόσταση από το κεντρικό πλαίσιο, στην οποία τοποθετούνται τα πλαίσια επέκτασης, τα συστήματα διακρίνονται σε δύο περιπτώσεις:

- Κεντρικό σύστημα
- Αποκεντρωμένο σύστημα

### Κεντρικό σύστημα

Το κεντρικό πλαίσιο και όλα τα πλαίσια επέκτασης βρίσκονται χωροταξικά σε μια θέση (π.χ. τοποθετημένα μέσα σ' έναν πίνακα). Η απόσταση του πιο μακρινού πλαισίου επέκτασης από το κεντρικό δεν επιτρέπεται να είναι μεγαλύτερη από π.χ. 2 - 2,5m.

Εκτός από τις καθορισμένες θέσεις των μονάδων τροφοδοσίας, CPU και διασύνδεσης, σε όλες τις άλλες θέσεις μπορούν να τοποθετηθούν μονάδες εισόδων - εξόδων.

### Αποκεντρωμένο σύστημα

Στο αποκεντρωμένο σύστημα ορισμένα πλαίσια επέκτασης βρίσκονται σε μεγάλη απόσταση από το κεντρικό.

Ένα αποκεντρωμένο σύστημα έχει νόημα, αν η προς αυτοματισμό εγκατάσταση έχει τα στοιχεία της (π.χ. τερματικοί, μπουτόν, βαλβίδες, λυχνίες, ρελέ ισχύος, κ.λπ.) «μοιρασμένα» σε περισσότερα «κέντρα βάρους».

Σε μια τέτοια περίπτωση «αντιστοιχούμε» ένα ή περισσότερα πλαίσια επέκτασης σε κάθε ένα τέτοιο βάρος. Αποτέλεσμα είναι η μεγάλη οικονομία καλωδίωσης από τα στοιχεία της εγκατάστασης προς τις μονάδες εισόδων - εξόδων.

Η διασύνδεση δύο μακρινών μεταξύ τους πλαισίων επιτυγχάνεται μέσω ειδικών μονάδων διασύνδεσης σε κάθε πλαίσιο και ειδικού καλωδίου. Στα πλαίσια αυτά επίσης είναι απαραίτητο να υπάρχει και η μονάδα τροφοδοσίας.

Πάνω σε κάθε κλάδο της «αλυσίδας» επιτρέπεται να υπάρχουν μέχρι 4 «σημεία», στα οποία θα τοποθετηθούν τα πλαίσια επέκτασης. Στα 4 αυτά «σημεία»

μπορούν να τοποθετηθούν περισσότερα από ένα πλαίσια με τη μέθοδο του κεντρικού συστήματος.

### **Σημείωση 1**

Αν και οι διάφορες είσοδοι - έξοδοι δεν είναι χωροταξικά στην ίδια θέση, αυτό δεν ενοχλεί πουθενά το χρήστη, όταν φτιάχνει το πρόγραμμα. Προγραμματιστικά δηλαδή δεν γίνεται καμία διάκριση σε «κοντινές» και «μακρινές» I/Q.

### **Σημείωση 2**

Εκτός από τη μέθοδο αποκεντρωμένου συστήματος με τη μορφή της «αλυσίδας» υπάρχει και η λεγόμενη «ακτινική» μορφή. Σ' αυτήν μπορούν να συνδεθούν ακτινικά με το κεντρικό πλαίσιο μέχρι 3 αποκεντρωμένες θέσεις - κάθε μια σε απόσταση 1Km. Σε κάθε αποκεντρωμένη θέση μπορεί να τοποθετηθούν περισσότερα από ένα πλαίσια με τη μέθοδο του κεντρικού συστήματος.

### Σύγκριση κεντρικού - αποκεντρωμένου συστήματος

Σε μια εγκατάσταση με χωροταξικά «μαζεμένα» όλα τα στοιχεία της, χρησιμοποιούμε πάντα το κεντρικό σύστημα.

Αντίθετα, σε μια εγκατάσταση με κατανεμημένα χωροταξικά τα στοιχεία της σε περισσότερα «κέντρα βάρους», πρέπει να εξεταστεί οικονομοτεχνικά, αν συμφέρει το αποκεντρωμένο σύστημα.

### **1.3.2. Μονάδα τροφοδοσίας**

Η μονάδα τροφοδοσίας χρησιμεύει για να δημιουργηθούν από την τάση του δικτύου οι απαραίτητες εσωτερικές τάσεις για την τροφοδοσία αποκλειστικά των ηλεκτρικών στοιχείων, που υπάρχουν μέσα στον ελεγκτή (τρανζίστορς, ολοκληρωμένα κ.λπ.). Επίσης για να διατηρηθεί το περιεχόμενο της μνήμης RAM σε μια διακοπή τάσης με τη βοήθεια μπαταρίας, που ενσωματώνεται σ' αυτή.

Τα σπουδαιότερα τεχνικά χαρακτηριστικά μιας μονάδας τροφοδοσίας είναι τα εξής:

#### Είσοδος

Ονομαστική τάση, ανοχές τάσης, συχνότητα, απορροφούμενο ρεύμα, προστασία.

#### Έξοδος

Ονομαστική τάση, ονομαστικό ρεύμα, προστασία βραχυκυκλώματος.

#### Διάφορα

Μπαταρία για διατήρηση μνήμης RAM.

Αν το πρόγραμμα ενός ελεγκτή πρόκειται να αποθηκευτεί σε μνήμη RAM, τότε απαραίτητα πρέπει να υπάρχει στο σύστημα και μια μπαταρία για τη διατήρηση του περιεχομένου της μνήμης σε μια διακοπή τάσης του δικτύου. Αυτή ή μπαταρία, που είναι συνήθως λιθίου τοποθετείται στη μονάδα τροφοδοσίας και μπορεί να κρατήσει το πρόγραμμα της μνήμης RAM για πολύ μεγάλο χρονικό διάστημα.

Πρέπει, όμως, να προσεχτούν τα εξής σημεία:

- α) Η μπαταρία θα πρέπει να αλλάζεται με την συχνότητα που ορίζει ο κατασκευαστής.
- β) Το SOFTWARE του ελεγκτή πρέπει να παρέχει στο χρήστη τη δυνατότητα να αξιολογήσει το γεγονός, ότι η μπαταρία έχει πέσει κάτω από το όριο ασφαλείας.
- γ) Η αντικατάσταση της μπαταρίας πρέπει πάντοτε να γίνεται με τον ελεγκτή υπό τάση για να μην χαθεί το πρόγραμμα.

### 1.3.3. Κεντρική μονάδα επεξεργασίας (CPU)

Η CPU έχει τα εξής χαρακτηριστικά:

- Ενσωματωμένη RAM εργασίας (Working Memory)
- Ενσωματωμένη RAM φορτώματος (Load memory)
- Εξωτερική Flash EPROM φορτώματος (Load memory) που επεκτείνει την ενσωματωμένη.

Η Load μνήμη περιλαμβάνει όλα τα Block Λογικής (συμπεριλαμβανομένων και Block που δεν απαιτούνται για την εκτέλεση του προγράμματος πχ. Block Header) ,μπλοκ Δεδομένων και Δεδομένων παραμετροποίησης( 4 Kbytes) που δεν χάνονται ούτε με το Reset της μνήμης ούτε με την απώλεια μπαταρίας του τροφοδοτικού. Με την Μεταγωγή της CPU από κατάσταση Stop - κατάσταση εκτέλεσης του προγράμματος μεταφέρονται από την Load μνήμη στην Working μνήμη μόνο τα κομμάτια των μπλοκ λογικής και δεδομένων που είναι απαραίτητα για την εκτέλεση του προγράμματος. Η working μνήμη είναι γρηγορότερη από την Load μνήμη και σβήνει με το μπουτόν Reset memory της CPU ή αν πέσει η μπαταρία του τροφοδοτικού.

Η CPU εμπεριέχει Status Leds και Leds σφαλμάτων ενώ ο τρόπος λειτουργίας επιλέγεται με κλειδί (KEY). Όταν το κλειδί μετακινηθεί ο τρόπος λειτουργίας της CPU δεν μπορεί να αλλάξει. Αυτή η δυνατότητα προστατεύει το πρόγραμμα της εφαρμογής από μη εξουσιοδοτημένη αλλαγή ή διαγραφή του.

Η CPU περιλαμβάνει διαγνωστική μνήμη μήκους 100-120 μηνυμάτων που δεν σβήνεται ούτε με την πτώση τάσης ούτε με το Reset της μνήμης και καταγράφονται με ώρα και ημερομηνία γεγονότα που συνδέονται με:

- Σφάλματα της CPU.
- Σφάλματα συστήματος της CPU.
- Σφάλματα περιφερειακών modules.
- Μεταγωγή από κατάσταση Stop-Εκτέλεση προγράμματος (RUN) -Stop.
- Προγραμματιστικά λάθη στο πρόγραμμα εφαρμογής.

Η διαγνωστική μνήμη μπορεί να διαβασθεί ON-LINE τοπικά με τον φορητό ηλεκτρονικό υπολογιστή.

Επίσης η CPU περιλαμβάνει Διαγνωστικό Alarm μπλοκ στο οποίο προγραμματίζοντας την Διεύθυνση μιας οποιασδήποτε κάρτας εισόδου / εξόδου λαμβάνονται διαγνωστικά bit για την κάρτα όπως:

- Βλάβη κάρτας
- Εσωτερικό εξωτερικό σφάλμα
- Πρόβλημα σε κάποιο κανάλι της κάρτας
- Έλλειψη εξωτερικής τάσης
- Έλλειψη φίσσας καλωδίων , Bit που μπορούν συνολικά να ενημερώσουν τοπική λυχνία ή να μεταφερθούν μέσω του δικτύου των οπτικών ινών στους ΚΣ. Ειδικά στις κάρτες αναλογικών εισόδων αν στο στάδιο αρχικής παραμετροποίησης της κάρτας ενεργοποιηθεί ο χρήστης την ανίχνευση κομμένου καλωδίου τότε είτε με την ενεργοποίηση του διαγνωστικού Alarm μπλοκ είτε με την μη ενεργοποίηση του αλλά οπτικά σε εξωτερικό LED της κάρτας (System Fault) ειδοποιείται τοπικά ή remote το σύστημα για το κομμένο καλώδιο οποιοδήποτε αναλογικού οργάνου(4....20mA)
- Ο μέσος κύκλος εκτέλεσης για 1000 εντολές είναι 0.3 m sec- 0.6 msec
- Τα εσωτερικά βοηθητικά ρελαί (Flags) είναι 2048 από τα οποία όλα μπορούν να είναι μόνιμα (διατήρηση περιεχομένου τους σε περίπτωση διακοπής τάσης ή μεταγωγής της CPU από RUN-Stop- RUN .
- Τα χρονικά και οι απαριθμητές που είναι ενσωματωμένα στην CPU είναι 128 και 64 αντίστοιχα εκ των οποίων όλα μπορούν να είναι μόνιμα.
- Ο μέγιστος αριθμός ψηφιακών εισόδων εξόδων είναι 512
- Ο μέγιστος αριθμός αναλογικών εισόδων και εξόδων είναι 64
- Υπάρχει ενσωματωμένο ρολόι πραγματικού χρόνου
- Υποστηρίζονται Γλώσσες προγραμματισμού όπως LAD (LADDER) FBD (Πύλες) STL ( λίστα εντολών) σύμφωνα με τα διεθνή Standards IEC 1131-3 Part 3 αλλά και επιπλέον γλώσσες προγραμματισμού με την χρήση Optional Software πακέτων όπως
- Όλες οι επιπλέον γλώσσες προγραμματισμού με μικρό ποσό μετάφρασης (Compilation) μεταφράζονται στις γλώσσες LAD, FBD, STL.
- Υποστήριζεται δομημένος προγραμματισμός με την ύπαρξη ειδικών μπλοκ οργάνωσης (OB) Block δεδομένων (DB, Block λειτουργία (FC, FB), Block Λειτουργιών συστήματος (SFC, SFB) και Block δεδομένων συστήματος (SDB).
- Υποστηρίζονται οι παρακάτω εντολές
- Λογικής bit BOOLEAN (AND, OR)
- Λογικής Word boolean (AND, OR) με 16 bit-Σταθερές.
- Λογικής Double Boolean (AND,OR) με 32 bit- Σταθερές
- Εντολές παλμού.
- Set / Reset bit (πχ. Inputs, Outputs, Memorys)
- Εντολές ολίσθησης Δεξιά, αριστερά και κυκλικής ολίσθησης.
- Set /Reset bit (π.χ. Inputs, Outputs, Memorys)

- Εντολές ολίσθησης δεξιά, αριστερά και κυκλικής ολίσθησης
- Εντολές χρονικών και απαριθμητών
- Αποθήκευσης και μεταφοράς τιμών από και προς καταχωρητές byte, Word, Doubleword.
- Εντολές σύγκρισης (16bit, 32 bit ακέραιων αριθμών, 32 bit δεκαδικών αριθμών).
- Αριθμητικές πράξεις όπως α) Πρόσθεση/πολλαπλασιασμό 16bit ακέραια  
β) Πρόσθεση/πολλαπλασιασμό 32 bit ακέραια  
γ) Πρόσθεση/πολλαπλασιασμό 32 bit δεκαδικών
- Εύρεση τετραγωνικής ρίζας, Λογαριθμικές πράξεις, τριγωνομετρικές λειτουργίες.
- Εντολές αλλαγής ελέγχου του προγράμματος από μπλόκ σε μπλοκ και από εντολή σε εντολή μέσα στο ίδιο μπλοκ .
- Εντολές μετατροπής κώδικα (πχ BCD σε 16 bit Ακέραια)
- Διάφοροι τρόποι εκτέλεσης του προγράμματος όπως κυκλικός, ελεγχόμενος από γεγονός ή από χρόνο
- Ένδειξη μεγίστου - ελαχίστου- μέσου κύκλου εκτέλεσης προγράμματος
- Υποστήριξη αναλογικό - ολοκληρωτικό- διαφορικού ελεγκτή κλειστού βρόχου (PID Controller) με την βοήθεια επιπλέον πακέτου παραμετροποίησης και πακέτου Block Λειτουργίας.

Οι παρακάτω PID Controllers είναι διαθέσιμοι

- α) Συνεχείς PID Controllers
- β) Controllers παλμού
- γ) Βηματικοί Controllers

Το πακέτο παραμετροποίησης υποστηρίζει Test λειτουργία και λειτουργία βελτιστοποίησης του κλειστού βρόχου. Τα Block Λειτουργίας καταλαμβάνουν στην μνήμη της CPU περίπου 6 KBYTE ενώ κάθε βρόγχος καταλαμβάνει περίπου 0.4 KBYTE.

Η επικοινωνία της CPU με τον φορητό προγραμματιστή (PG) και του τοπικού πληκτρολογίου και οθόνης (OPERATION PANEL) γίνεται RS485 πρωτόκολλο και ταχύτητες μετάδοσης μέχρι 187.500 bps.

Με το Interface και φίσσα που έχει 2 Interfaces και που τοποθετείται πάνω στην CPU, μπορεί ο χειριστής ταυτόχρονα να συνδέσει τον φορητό προγραμματιστή για λειτουργίες ελέγχου και εκσφαλμάτωση του προγράμματος της CPU και το OPERATION PANEL για την εμφάνιση στην οθόνη του των μιμικών διαγραμμάτων , της δυνατότητας αλλαγής των παραμέτρων λειτουργίας, των ενδείξεων λειτουργίας κινητήρων και την δυνατότητα χειροκίνητης λειτουργίας.

Στο Interface της CPU μπορούν να συνδεθούν ταυτόχρονα μέχρι 16 Stations (όπως CPUS, PGs, OPs) με μέγιστη απόσταση μεταξύ 2 γειτονικών σταθμών 50m χωρίς repeaters  
1100m με 2 repeaters, 11000 m με 10 repeaters σε σειρά και 93 km μέσω οπτικών ινών (Fiber optics)

Με το Interface μπορούν να ανταλλαχθούν μικρά Set δεδομένων (64 bytes κυκλικά) μεταξύ διαφόρων CPU's με απλό τρόπο.

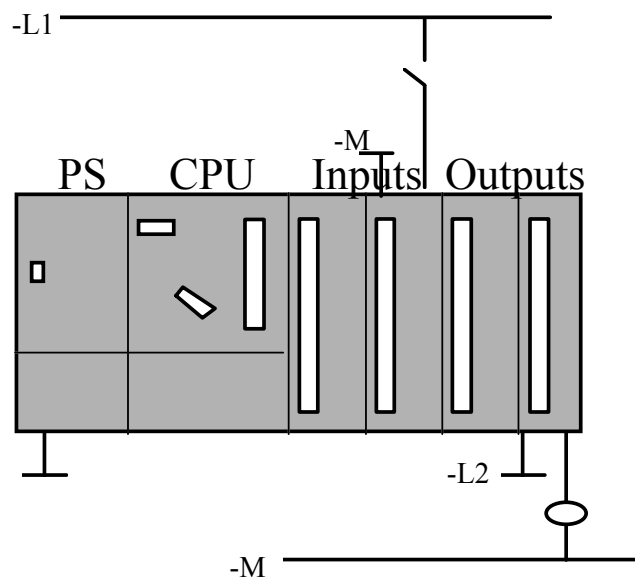
Στην κεντρική μονάδα (CPU) ο μικροεπεξεργαστής προσπελαύνει συνεχώς (κυκλικά) το πρόγραμμα, που είναι γραμμένο στην μνήμη. Ρωτάει συνεχώς, αν οι διάφορες είσοδοι έχουν ή δεν έχουν τάση (επαφές κλειστές ή ανοιχτές), επεξεργάζεται τις εντολές του προγράμματος και βάσει αυτών εξαναγκάζει τις εξόδους να διεγερθούν ή όχι (δηλ. να αποκτήσουν ή όχι τάση, οπότε διεγείρονται ή όχι τα συνδεδεμένα σ' αυτές ρελέ, βαλβίδες κ.λπ.).

Το πρόγραμμα γράφεται στη μνήμη με τη βοήθεια μιας συσκευής προγραμματισμού (προγραμματιστής). Αυτή συνδέεται στην κεντρική μονάδα μόνο όταν πρόκειται να γραφτεί ή να μεταφερθεί το πρόγραμμα στην μνήμη, ή αν πρόκειται να γίνουν αλλαγές. Επίσης, χρησιμοποιείται για τον έλεγχο των διαφόρων σημάτων κατά την εξέλιξη του προγράμματος και για την ανεύρεση σφαλμάτων.

#### 1.3.4. Μονάδες εισόδων / εξόδων (I/Q)

Τα καλώδια που έρχονται από τα αισθητήρια της παραγωγικής διαδικασίας (τερματικοί, μπουτόν, διακόπτες), συνδέονται στις κλέμες των μονάδων εισόδων (είσοδοι του ελεγκτή).

Αντίστοιχα, τα καλώδια που πηγαίνουν προς τα ρελέ ισχύος, βαλβίδες, λυχνίες κ.λπ. συνδέονται στις κλέμες των μονάδων εξόδου (έξοδοι του ελεγκτή).



**Σχ. 1.2.** Δομή προγραμματιζόμενου ελεγκτή. PS = τροφοδοτικό, CPU = κεντρική μονάδα επεξεργασίας, Inputs = είσοδοι, Outputs = έξοδοι., -M = ο ακροδέκτης M του τροφοδοτικού (αρνητικός πόλος)

Ο ελεγκτής αντιλαμβάνεται ότι ένα αισθητήριο είναι ανοιχτό ή κλειστό από το αν εμφανίζεται τάση L1 στην αντίστοιχη κλέμα εισόδου. Επίσης, αν από το πρόγραμμα δοθεί εντολή για διέγερση π.χ. μιας βαλβίδας, τότε εμφανίζεται η τάση L2 στην αντίστοιχη κλέμα εξόδου.

Οι τάσεις L1, L2 δεν παρέχονται από την μονάδα τροφοδοσίας του ελεγκτή και πρέπει να δημιουργηθούν από εμάς με κατάλληλο τροφοδοτικό (για DC) ή μετασχηματιστή τάσης χειρισμού (για AC).

### Μονάδες ψηφιακών εισόδων

Ένας ελεγκτής αντιλαμβάνεται ότι μια εξωτερική επαφή (π.χ. τερματικός) έκλεισε, όταν στην αντίστοιχη κλέμα εισόδου εμφανίζεται τάση. Η τάση αυτή ονομάζεται τάση εισόδων.

Η τάση για την τροφοδοσία των εισόδων δεν δημιουργείται από τη μονάδα τροφοδοσίας του ελεγκτή, αλλά πρέπει να τη δημιουργήσουμε εμείς με κατάλληλο τροφοδοτικό (για DC) ή μετασχηματιστή τάσης χειρισμού (για AC). Εξάιρεση αποτελούν συνήθως οι πολύ μικροί ελεγκτές, στους οποίους ο κατασκευαστής μπορεί να έχει ενσωματώσει ένα μικρό τροφοδοτικό.

### **Σημείωση**

Μια μονάδα εισόδων έχει συνήθως 4, 8, 16 ή 32 εισόδους, ανάλογα με τον τύπο του ελεγκτή και τη τάση. Υποχρεωτικό είναι να χρησιμοποιείται η ίδια τάση για όλες τις εισόδους μιας μονάδας εισόδων. Για μια άλλη όμως μονάδα εισόδων μπορεί να χρησιμοποιηθεί διαφορετική τάση, όπως:

### **Η κάρτα ψηφιακών εισόδων 32X24 VDC** ικανοποιεί τα χαρακτηριστικά

- Τάση εισόδου : Ονομαστική τιμή 24 VDC
- Επιτρεπτή περιοχή 20.4 - 28.8 VDC
- Γαλβανική απομόνωση
- Περιοχή τάσης για το σήμα "1" 15-30VDC , Περιοχή τάσης για το σήμα "0" -3 - 5V
- Ένδειξη της κατάστασης του σήματος της κάθε ψηφιακής εισόδου με LED.
- Επιπρόσθετη φίσσα καλωδίωσης που μετακινείται απλά και χωρίς κίνδυνο να τοποθετηθεί σε λάθος τύπο κάρτας (περιλαμβάνει Key πολικότητας)
- Μέγιστος χρόνος ανταπόκρισης στην ονομαστική τάση εισόδου :1.2 -4.8 ms
- Ρεύμα εισόδου για σήμα "1" μέγιστο 7.5 mA
- Δυνατότητα για σύνδεση με δυσύρματο BERO ( επιτρεπτό υπόλοιπο ρεύματος αισθητηρίου 1.5 mA)
- Δυνατότητα συλλογής ψηφιακής πληροφορίας μέχρι 1000m με μπλενταρισμένο καλώδιο 600 m χωρίς μπλενταρισμένο καλώδιο.

Τα σπουδαιότερα τεχνικά χαρακτηριστικά μίας μονάδας εισόδων είναι: το πλήθος εισόδων, η γαλβανική απομόνωση, η ονομαστική τάση, οι ανοχές τάσης για



σήμα «1», οι ανοχές τάσης για σήμα «0», η μέγιστη συνολική διαδρομή καλωδίων, το ρεύμα που απορροφά κάθε είσοδος σε σήμα «1», το ρεύμα που απορροφά η μονάδα συνολικά από τα εσωτερικά DC 5V, η απαιτούμενη πρίζα καλωδίων (τύπος).

### Μονάδες ψηφιακών εξόδων

Οι μονάδες ψηφιακών εξόδων χρησιμεύουν για τη διέγερση των εξωτερικών στοιχείων της εγκατάστασης, όπως ρελέ κινητήρων, βαλβίδες, ενδεικτικές λυχνίες κλπ. Όταν από το πρόγραμμα δοθεί εντολή για τη διέγερση ενός π.χ. εξωτερικού ρελέ, τότε κλείνει ο αντίστοιχος «διακόπτης» της εξόδου. Η τάση εμφανίζεται στην κλέμα εξόδου και το ρελέ οπλίζει. Η τάση αυτή ονομάζεται τάση εξόδων. Ο «διακόπτης» εξόδου είναι συνήθως ηλεκτρονικός (τρανζίστορ, triac), αλλά μπορεί να είναι και μηχανική επαφή μικρορελέ.

Η τάση για την τροφοδοσία των μονάδων εξόδων δεν δημιουργείται από τη μονάδα τροφοδοσίας του ελεγκτή, αλλά πρέπει να τη δημιουργήσουμε εμείς με κατάλληλο τροφοδοτικό (για DC) ή μετασχηματιστή τάσης χειρισμού (για AC).

### **Σημείωση**

Τα κυκλώματα και οι τάσεις των εισόδων είναι τελείως ανεξάρτητα από τα κυκλώματα και τις τάσεις των εξόδων. Επομένως η τάση για τις εισόδους μπορεί να είναι διαφορετική από την τάση για τις εξόδους. Επιπλέον υπάρχει η δυνατότητα και ξεχωριστής τάσης ανά μονάδα εισόδων ή εξόδων. Συνήθως μια μονάδα εξόδων περιλαμβάνει 4 ή 8 ή 16 ή 32 εξόδους, ανάλογα με τον τύπο του ελεγκτή και την τάση.

Τα σπουδαιότερα τεχνικά χαρακτηριστικά μίας μονάδας εξόδων είναι: το πλήθος των εξόδων, η γαλβανική απομόνωση, η ονομαστική τάση, οι ανοχές τάσης, το ονομαστικό ρεύμα κάθε εξόδου, το ελάχιστο ρεύμα φορτίου, η ταυτόχρονη φόρτιση εξόδων μιας ομάδας, η προστασία εξόδων, η μέγιστη συνολική διαδρομή καλωδίων, η συχνότητα ζεύξεων, το ρεύμα που απορροφά η μονάδα συνολικά από τα εσωτερικά 5 V, η απαιτούμενη πρίζα καλωδίων (τύπος).

Αν γίνει κάποιο βραχυκύκλωμα στο εξωτερικό κύκλωμα μιας εξόδου, τότε χρειάζεται η «επέμβαση» της προστασίας που υπάρχει μέσα στη μονάδα. Στην απλούστερη περίπτωση, η προστασία αυτή είναι μια ασφάλεια υπερταχείας τήξης και μάλιστα μία ανά μονάδα εξόδων. Αντίθετα, στις μονάδες DC 24V, είναι συνηθισμένη η «ηλεκτρονική» προστασία. Σ' αυτή την περίπτωση, όταν συμβεί ένα εξωτερικό βραχυκύκλωμα, η προστασία μειώνει ή και μηδενίζει ακόμα την τάση εξόδου της μονάδας, ώστε το ρεύμα εξόδου να μην υπερβεί το μέγιστο επιτρεπόμενο.

Μία τυπική κάρτα ψηφιακών εξόδων είναι:

### **Η κάρτα ψηφιακών εξόδων 32X24VDC σε ομάδες των 8 εξόδων έχει**

- Τάση τροφοδοσίας 24VDC
- Γαλβανική απομόνωση
- Επιτρεπτή περιοχή τάσης 20.4...28.8 VDC
- Τάση εξόδου για "σήμα"1" 24VDC  $\pm 0.8V$
- Ρεύμα εξόδου για "1" σε 60° , 0.5A
- Ελάχιστο ρεύμα για "1" σε 60° 5mA

- Ρεύμα εξόδου για "0" σε 60° , 0.5A
- Συνολικό ρεύμα εξόδου ( ανά ομάδα εξόδων ) 2A
- Φορτίο Λαμπτήρα 5W
- Συχνότητα ζεύξεων επαφών  
ΩΜΙΚΩΝ 100HZ  
ΕΠΑΓΩΓΙΚΩΝ 0.5HZ  
Φορτία ενδείξεως 100HZ
- Ένδειξη κατάστασης του σήματος της κάθε ψηφιακής εξόδου με LED
- Επιπρόσθετη φίσσα καλωδίων
- Ηλεκτρονική προστασία από βραχυκύκλωμα

Δυνατότητα αποστολής εντολής μέχρι 600m χωρίς μπλενταρισμένο καλώδιο και 1000m με μπλενταρισμένο

#### Μονάδες αναλογικών εισόδων

**Η αναλογική κάρτα εισόδων** μπορεί να επεξεργασθεί αισθητήρια με δυνατότητα μετρήσεων βασικών περιοχών τάσης

± 1V /200 KΩ Αντίσταση εισόδου

1..5V/200 KΩ Αντίσταση εισόδου και περιοχών ρεύματος

4...20mA/80Ω Αντίσταση εισόδου

±20mA/ 80Ω Αντίσταση εισόδου

Θερμοστοιχεία N, E, J, K

PT100 Standard /10 Mohms / 4 κανάλια η 1 κανάλι

αλλάζοντας τον τύπο της μέτρησης με μηχανικά jumpers πάνω στην κάρτα και ρυθμίζοντας διάφορα μεγέθη (πχ πάνω -κάτω όριο) από το Software.

- Η ανάλυση του A/D μετατροπέα της κάρτας είναι 12 bits
- Ο κύκλος ολοκλήρωσης / μετατροπής για κάθε κανάλι 2.5/100 msec
- Το μήκος καλωδίου μέχρι το αισθητήριο θα είναι τουλάχιστον 200m με μπλενταρισμένο καλώδιο
- Έχει γαλβανική απομονωση
- Προστασία έναντι ανάστροφου πολικότητας
- Επιτρεπτή τάση εισόδου για κανάλι τάσης 20V
- Επιτρεπτό ρεύμα εισόδου για κανάλι ρεύματος 40mA
- Αντιστάθμιση Θερμοκρασίας : εσωτερική ή εξωτερικό με Module αντιστάθμισης.
- Όριο σφάλματος λειτουργίας ( πάνω από την περιοχή θερμοκρασίας που αναφέρεται στην περιοχή εισόδου ) max +-1%
- Όριο Βασικού σφάλματος (Όριο σφάλματος λειτουργίας στα 25° που αναφέρεται στην περιοχή εισόδου ) max +- 0.6 %
- Δυνατότητα διάγνωσης μέσω κόκκινου Led για σφάλματα καναλιών

- Φίσσα καλωδίων με στοιχείο κωδικοποίησης. Όταν η φίσσα τοποθετείται για πρώτη φορά στην κάρτα τότε το στοιχείο κωδικοποίησης επιδρά στο να μπορεί να τοποθετηθεί η φίσσα σε κάρτες της ίδιας περιοχής τάσης ή ρεύματος.

### Μονάδες αναλογικών εξόδων

#### **Η αναλογική κάρτα εξόδων έχουν:**

- Τάση τροφοδοσίας 24VDC
- Γαλβανικά απομονωμένη
- Περιοχές εξόδου τάσης  $\pm 10V$ , 0-10V 1-5V και περιοχές εξόδου ρεύματος 4..20mA,  $\pm 20mA$ , 0-20mA πού μπορούν να αλλαχθούν με μηχανικά Jumpers πάνω στην κάρτα ενώ διάφορες ρυθμίσεις μπορούν να γίνουν από το Software.
- Αντίσταση φορτίου για τα κανάλια τάσης min 1K $\Omega$
- Αντίσταση φορτίου για τα κανάλια ρεύματος max 0.5 K $\Omega$ .
- Χωρητικά φορτία max 1 $\mu F$
- Επαγωγικά φορτία max 1 mH
- Προστασία από βραχυκύκλωμα με ρεύμα βραχυκύκλωσης 25mA για εξόδους τάσης.
- Ισχύς εξόδου με τάση ανοικτού κυκλώματος 18V
- Η ανάλυση του D/A Converter είναι 11 bits+Πρόσημο ( $\pm 10V$ , 4..20mA,  $\pm 20mA$ , 1-5V) , 12 Bits (0-10V, 0-20mA)
- Ο κύκλος μετατροπής για κάθε κανάλι είναι max 0.8 msec και ειδικά για Ωμικά φορτία 0.1ms για χωρητικά φορτία 3.3 ms για επαγωγικά φορτία 0.5.ms
- Όρια λειτουργίας (0..60°C στην περιοχή εξόδου)  
Τάση  $\pm 0,5\%$   
Ρεύμα  $\pm 0.6\%$
- Βασικό σφάλμα (0-25°C στην περιοχή εξόδου)  
Τάση  $\pm 0,2\%$   
Ρεύμα  $\pm 0,3\%$
- Το μήκος καλωδίου μέχρι το στοιχείο ενεργοποίησης είναι 200μ με μπλενταρισμένο καλώδιο
- Φίσσα καλωδίου με την ίδια λογική όπως της αναλογικής κάρτας εισόδου.
- Δυνατότητα διάγνωσης μέσω κόκκινου Led για σφάλματα καναλιών

### 1.3.5. Κάρτα επικοινωνίας

- Interfaces 1 ( RS232C-V24,RS485)
- ταχύτητα μετάδοσης 2.4 Kbit/sec - 19.2 Kbit/sec
- LEDS για SEND-RECEIVE- ERROR
- Πρωτόκολλα : ASCII ( μήκος πληροφορίας max 1024 bytes , ταχύτητα 9.6 Kbit/sec), 3964R( μήκος πληροφορίας max 1024 bytes , ταχύτητα 19.2 Kbit/sec), RK512 ( με ειδικό Software εφαρμογή ) , Printer Driver (ταχύτητα 9.6 Kbit/sec)

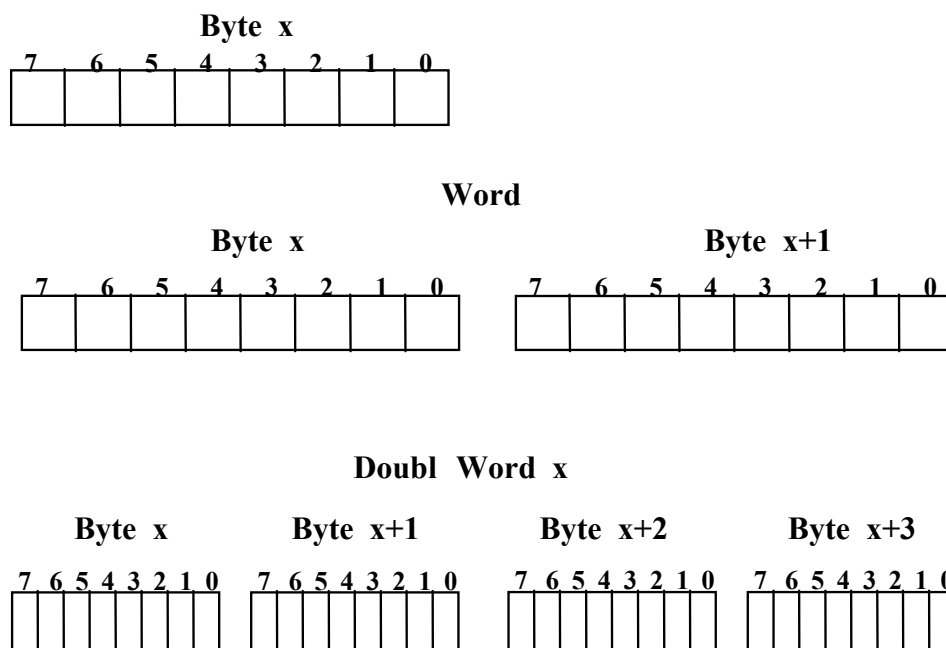
Ο χρήστης με την βοήθεια πακέτου παραμετροποίησης πού φορτώνεται στο βασικό Software , καθορίζει το πρωτόκολλο και τα χαρακτηριστικά του και αυτά υπό μορφή System Data φορτώνονται στην Load Memory της CPU.

Το PLC ικανοποιεί τα βιομηχανικά Standards λόγω της συμβατότητας τους σε υψηλά ηλεκτρομαγνητικά πεδία, της λειτουργίας του σε μεγάλη θερμοκρασία 0...60°C και της αντοχής του σε κραδασμούς, δεν απαιτεί κατά την λειτουργία του επιπλέον ανεμιστήρες.

### 1.4. Ονοματολογία

Τα PLC βασίζονται στην ψηφιακή τεχνολογία, γεγονός που σημαίνει ότι χρησιμοποιούν το δυαδικό σύστημα. Αυτό συμβαδίζει με την ήδη υπάρχουσα φιλοσοφία στην τεχνολογία των αυτοματισμών, αφού και οι συμβατικοί αυτοματισμοί (ρελέ, διακόπτες, μπουτόν) βασίζονται σε δύο επίσης καταστάσεις «ανοιχτό - κλειστό».

Η οργάνωση των bit μνήμης, στη μνήμη του PLC είναι κατά byte, λέξεις (2 bytes), ή διπλή λέξη (4 bytes) όπως φαίνεται στο ακόλουθο σχήμα:



Σχ. 1.3. Οργάνωση μνήμης σε byte. Word = λέξη, Double Word = διπλή λέξη.

Η διεύθυνση μιας εισόδου ή μιας εξόδου, αντίστοιχα, στο PLC ορίζεται με τη μορφή:

I x.y όπου x = καθορίζει τη διεύθυνση του byte εισόδων/ εξόδων μόνο για bit  
Q x.y y = καθορίζει τη διεύθυνση του bit εισόδων/ εξόδων

I xy όπου x = καθορίζει το είδος για τις  
Qxy y = καθορίζει την αρχική υπόλοιπες  
διεύθυνση του bit εισόδων / εξόδων μορφές

ενώ I είναι η είσοδος  
Q είναι η έξοδος

Η ίδια διευθυνσιοδότηση ακολουθείται και στα βοηθητικά, δηλ τα bit μιας ειδικής περιοχής της μνήμης, που χρησιμοποιούνται από το πρόγραμμα για την αποταμίευση ορισμένων καταστάσεων (παίζουν το ρόλο των βοηθητικών ρελέ του συμβατικού ηλεκτρισμού). Σε αντίθεση με τις εισόδους / εξόδους, τα βοηθητικά δεν έχουν «επικοινωνία» με τον έξω κόσμο (π.χ. κάποια κλέμα ή LED) και η κατάσταση τους μπορεί να αναγνωριστεί μόνο με τη βοήθεια της μιας συσκευής προγραμματισμού.

#### 1.4.1. Ονοματολογία χρονικών

Με τον όρο χρονικό εννοούμε μια λέξη (2 byte) μέσα σε μια ειδική περιοχή της μνήμης - την περιοχή των χρονικών.

Σ' αυτή τη λέξη προτοποθετείται με το πρόγραμμα η τιμή του χρόνου (π.χ. 100 sec). Όταν - από το πρόγραμμα επίσης - ξεκινήσει να «μετράει» ο χρόνος, τότε η τιμή αυτή αρχίζει και μειώνεται με το ρυθμό της χρονικής μονάδας (π.χ. 1 sec). Όταν περάσει ο χρόνος, (δηλ. όταν η τιμή του χρονικού μέσα στη λέξη γίνει μηδέν), τότε παίρνουμε αντίστοιχο σήμα, το οποίο μπορούμε να αξιοποιήσουμε.

Ένα χρονικό χαρακτηρίζεται μονοσήμαντα ως εξής:

Tx

όπου T το χρονικό  
x ο αριθμός του χρονικού

#### 1.4.2. Ονοματολογία απαριθμητών

Με τον όρο απαριθμητής εννοούμε μια λέξη (2byte) μέσα σε μια ειδική περιοχή της μνήμης - την περιοχή των απαριθμητών.

Σ' αυτή την λέξη μέσα βρίσκεται κάθε στιγμή το «περιεχόμενο» του απαριθμητή, το οποίο αυξάνεται ή ελαττώνεται κατά 1 με κατάλληλες εντολές από το πρόγραμμα. Επίσης, το περιεχόμενο αυτό μπορεί να ζητηθεί και να αξιολογηθεί.

Ένας απαριθμητής χαρακτηρίζεται μονοσήμαντα ως εξής:

Cx

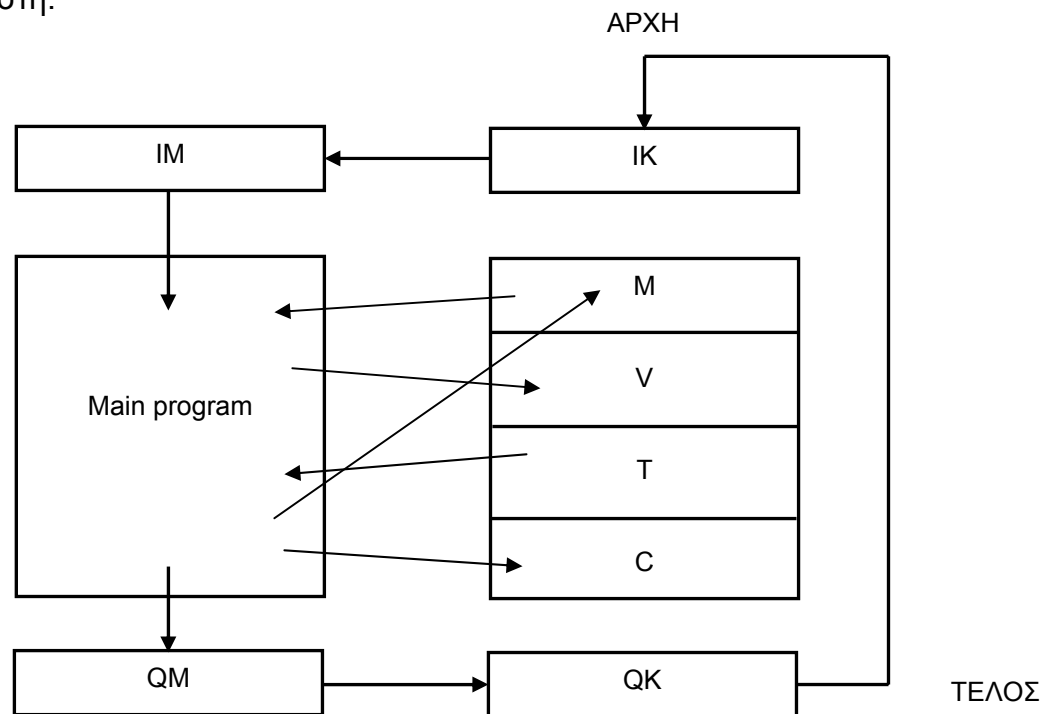
όπου C ο απαριθμητής  
x ο αριθμός του απαριθμητή

## 1.5. Κυκλική επεξεργασία

Βασικό χαρακτηριστικό ενός ελεγκτή είναι η συνεχής (κυκλική) επεξεργασία του προγράμματος που είναι γραμμένο από το χειριστή στη μνήμη. Ένας υπολογιστής επεξεργάζεται τις εντολές του προγράμματος του, καταλήγει στα αποτελέσματα και μετά σταματά.

Αντίθετα σ' έναν ελεγκτή συμβαίνουν τα εξής:

1. Ο μικροεπεξεργαστής διαβάζει την τρέχουσα κατάσταση των εισόδων και γράφει τις τιμές που αντιστοιχούν (0 ή 1) σε έναν πίνακα απεικόνισης καταστάσεων εισόδων (Process-Image Input Register) που βρίσκεται στην μνήμη του επεξεργαστή.



Σχ. 1.4. Κυκλική επεξεργασία προγράμματος

2. Ο μικροεπεξεργαστής διαβάζει και ταυτόχρονα εκτελεί μία, μία τις εντολές που είναι γραμμένες στη μνήμη, αρχίζοντας από την πρώτη και τελειώνοντας στην τελευταία
3. Μετά την εκτέλεση της τελευταίας εντολής, ο μικροεπεξεργαστής «οδηγεί» τις εξόδους ανάλογα με την τιμή (0 ή 1) που έχουν στον πίνακα απεικόνισης εξόδων (Process-Image Output Register).
4. Τέλος, ξαναγυρνά στο πρώτο βήμα και η διαδικασία επαναλαμβάνεται συνεχώς. Επομένως, σ' έναν ελεγκτή έχουμε μια διαρκή σάρωση και εκτέλεση των εντολών του προγράμματος.

Είναι ιδιαίτερα σημαντικό να τονίσουμε ότι η πληροφορία για την κατάσταση της εισόδου αποκτάται μόνο στην αρχή του κύκλου. Παρόλο που η κατάσταση της εισόδου, κατά τον χρόνο εκτέλεσης του προγράμματος, θεωρείται σταθερή (γεγονός

που βεβαίως μπορεί και να μην συμβαίνει), ο κύκλος του PLC είναι τόσο σύντομος (τυπικά μερικά ms) που ακόμα και αν αλλάξει κατάσταση η είσοδος, η CPU θα το αντιληφθεί στον αμέσως επόμενο κύκλο και θα δράσει ανάλογα, με καθυστέρηση μόνο χιλιοστών του δευτερολέπτου.

### 1.5.1. Χρόνος κύκλου

Σαν χρόνος κύκλου ορίζεται ο χρόνος που απαιτείται για την επεξεργασία και εκτέλεση (μία φορά) των εντολών που είναι γραμμένες στην μνήμη, από την πρώτη μέχρι την τελευταία. Είναι αυτονόητο, ότι όσο πιο μεγάλο είναι το πρόγραμμα τόσο μεγαλύτερος είναι και ο χρόνος κύκλου.

Για να μπορούν να συγκριθούν διαφορετικοί τύποι ελεγκτών, οι κατασκευαστές δίνουν συνήθως τον μέσο χρόνο κύκλου για 1 K (= 1024) εντολές προγράμματος. Όσο μικρότερος είναι αυτός ο χρόνος, τόσο ταχύτερος είναι ο ελεγκτής. Η έννοια μέσος χρόνος κύκλου οφείλεται στο γεγονός ότι όλες οι εντολές δεν απαιτούν ίδιο χρόνο για την επεξεργασία τους. Έτσι, ανάλογα με τις εντολές που θα χρησιμοποιήσουμε, διαφοροποιείται και ο χρόνος αυτός.

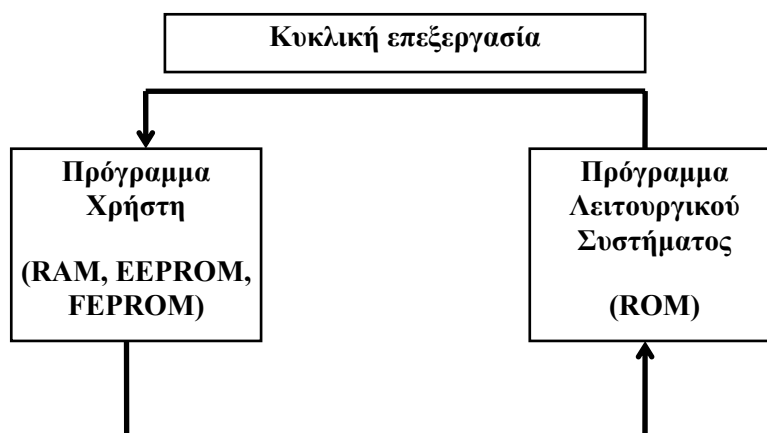
Συνήθως ο χρόνος κύκλου για ένα 1 K εντολές είναι της τάξεως μερικών msec.

## 1.6. Λειτουργικό σύστημα ελεγκτή προγραμματιζόμενης λογικής

Ορισμένες ενέργειες του ελεγκτή γίνονται αυτόματα, χωρίς καμία απολύτως εντολή από το χρήστη, π.χ.

- Όταν επανέρχεται η τάση μετά από μία διακοπή του δικτύου γίνεται μηδενισμός των βοηθητικών που ανήκουν στην περιοχή μνήμης χωρίς συγκράτηση.
- Πριν από την έναρξη κάθε κύκλου προγράμματος γίνεται μεταφορά σημάτων από τις κλέμες των μονάδων εισόδων στη μνήμη απεικόνισης καταστάσεων εισόδων (Process - Image Input Register).
- Μετά το τέλος κάθε κύκλου προγράμματος γίνεται μεταφορά της μνήμης απεικόνισης καταστάσεων εξόδων (Process - Image Output Register) στις αντίστοιχες κλέμες των μονάδων εξόδων.

Όλες αυτές οι απαραίτητες ενέργειες, οι οποίες προσδιορίζουν τι άλλο πρέπει να κάνει ο ελεγκτής παράλληλα με το κυρίως πρόγραμμα του χρήστη, το οποίο εμείς προγραμματίζουμε, αποτελούν το λειτουργικό πρόγραμμα του ελεγκτή. Το πρόγραμμα αυτό είναι συνήθως αποθηκευμένο σε μια μνήμη ROM μέσα στην κεντρική μονάδα και περιέχει εντολές που δεν μπορούν να διαβαστούν από εμάς, και οι οποίες καθορίζουν τις παραπάνω αντιδράσεις.



Σχ.1.5. Κυκλική επεξεργασία.

## 1.7. Προσπέλαση προγράμματος

Το γενικό πρόγραμμα μιας κεντρικής μονάδας αποτελείται από το λειτουργικό σύστημα και το πρόγραμμα του χρήστη.

Το **λειτουργικό σύστημα** όπως προαναφέραμε, αποτελεί το σύνολο που περιέχει όλες τις εντολές και τις δηλώσεις που ελέγχουν τις πηγές του συστήματος, τις διαδικασίες που χρησιμοποιούν αυτές τις πηγές, καθώς και περιοχές λειτουργίας όπως αποθήκευση δεδομένων στην περίπτωση πτώση της τάσης του δικτύου, ενεργοποίηση τάξεων προτεραιότητας, κλπ. Το λειτουργικό σύστημα αποτελεί ένα μέρος της κεντρικής μονάδας, στο οποίο ο χρήστης δεν έχει πρόσβαση γραφής. Εντούτοις, μπορούμε να φορτώσουμε ξανά το σύστημα αυτό από μια μονάδα μνήμης, π.χ. στην περίπτωση της ενημέρωσης με τις τελευταίες αλλαγές του προγράμματος.

Το **πρόγραμμα του χρήστη** αποτελεί το σύνολο όλων των εντολών και δηλώσεων, στην περίπτωση αυτή τα στοιχεία του προγράμματος, για την οδήγηση των σημάτων, μέσα από την οποία η όλη διαδικασία επηρεάζεται ανάλογα με τη προκαθορισμένη εργασία ελέγχου.

### 1.7.1. Μέθοδοι προσπέλασης προγράμματος

Το πρόγραμμα του χρήστη μπορεί να αποτελείται από διάφορα μέρη τα οποία η κεντρική μονάδα επεξεργασίας εκτελεί ανάλογα με το αν λαμβάνουν χώρα κάποια συγκεκριμένα γεγονότα. Ένα τέτοιο γεγονός μπορεί να είναι η εκκίνηση του αυτόματου συστήματος, μια διακοπή ή η ανίχνευση σφάλματος. Τα προγράμματα που εξαρτώνται από τέτοια γεγονότα χωρίζονται σε τάξεις προτεραιότητας που καθορίζουν την σειρά εκτέλεσης των μερών του προγράμματος όταν συμβαίνουν συγκεκριμένα γεγονότα.

Το χαμηλής τάξης προτεραιότητας πρόγραμμα είναι το κυρίως πρόγραμμα, το οποίο προσπελάζεται κυκλικά από την κεντρική μονάδα. Όλα τα άλλα γεγονότα μπορούν να διακόψουν το κυρίως πρόγραμμα σε οποιοδήποτε σημείο. Η CPU τότε



εκτελεί την ανάλογη ρουτίνα διακοπής ή ρουτίνα αποσφαλμάτωσης και επιστρέφει στο κυρίως πρόγραμμα.

Ένα συγκεκριμένο μπλοκ οργάνωσης (organization block OB) συναντάται σε κάθε γεγονός. Τα μπλοκ οργάνωσης αναπαριστούν τις τάξεις προτεραιότητας στο πρόγραμμα του χρήστη. Όταν ένα γεγονός λαμβάνει χώρα, η κεντρική μονάδα καλεί το ανάλογο μπλοκ οργάνωσης. Ένα τέτοιο μπλοκ αποτελεί ένα μέρος του προγράμματος του χρήστη.

Πριν ακόμα ξεκινήσει η CPU να εκτελεί το κυρίως πρόγραμμα, εκτελεί μια ρουτίνα εκκίνησης. Αυτή η ρουτίνα μπορεί να ενεργοποιηθεί με το άνοιγμα της κεντρικής τάσης τροφοδοσίας, από τον επιλογικό διακόπτη που υπάρχει στην CPU ή μέσω ενός προγραμματιστή.

Το κυρίως πρόγραμμα βρίσκεται μέσα στο μπλοκ οργάνωσης OB1, το οποίο η κεντρική μονάδα επεξεργασίας εκτελεί. Αφού έχει τελειώσει η εκτέλεση του OB1 (τέλος προγράμματος), η CPU επιστρέφει στο λειτουργικό σύστημα και μόλις καλέσει για εκτέλεση διάφορες λειτουργίες του συστήματος, όπως ενημέρωση του πίνακα διευθύνσεων των εξόδων, καλεί, ξανά, για άλλη μια φορά το OB1.

Γεγονότα που μπορούν να διακόψουν το πρόγραμμα είναι οι αιτήσεις διακοπών και τα σφάλματα. Οι διακοπές μπορούν να ζητηθούν από την διαδικασία (hardware interrupts) ή από την CPU (διακοπές εποπτείας). Όσον αφορά τα σφάλματα, υπάρχει ένας διαχωρισμός μεταξύ συγχρονισμένων και ασύγχρονων σφαλμάτων. Το ασύγχρονο σφάλμα είναι ανεξάρτητο από τον κύκλο του προγράμματος, για παράδειγμα διακοπή τροφοδοσίας σε μια μονάδα επέκτασης ή διακοπή που έχει προκληθεί από την αντικατάσταση της μονάδας. Το σύγχρονο σφάλμα προκαλείται από την εκτέλεση του προγράμματος, όπως η αναφορά σε μη υπαρκτή διεύθυνση ή η δημιουργία σφάλματος μετατροπής τύπων δεδομένων. Ο τύπος και ο αριθμός των καταχωρημένων γεγονότων, καθώς και τα ανάλογα μπλοκ οργάνωσης εξαρτώνται από την κεντρική μονάδα επεξεργασίας.

## **1.8. Δομή προγράμματος**

Μπορούμε να χωρίσουμε το πρόγραμμα σε όσα μέρη θέλουμε με σκοπό να το διαβάζουμε και να το αντιλαμβανόμαστε καλύτερα και ευκολότερα. Κάθε μέρος του προγράμματος πρέπει να έχει τεχνολογική και λειτουργική βάση. Αυτού του είδους τα μέρη ονομάζονται «Μπλοκ». Ένα μπλοκ αποτελεί ένα μέρος του προγράμματος του χρήστη που καθορίζεται από τις λειτουργίες του, τη δομή και τον σκοπό της ύπαρξής του.

### **1.8.1. Τύποι μπλοκ**

Υπάρχουν διάφοροι τύποι μπλοκ για διάφορους σκοπούς:

- Μπλοκ χρήστη.

Τα μπλοκ αυτά περιέχουν το πρόγραμμα και τα δεδομένα του χρήστη.

- Μπλοκ συστήματος.

Τα μπλοκ αυτά περιέχουν το πρόγραμμα και τα δεδομένα του συστήματος.

- Στάνταρτ μπλοκ.

Τα μπλοκ αυτά αποτελούν το κλειδί λειτουργίας των οδηγών (drivers) των ειδικών καρτών CP και FM.

## **Μπλοκ χρήστη**

Τα μεγάλα και περίπλοκα προγράμματα «δομούνται» (διαχωρίζονται) σε μπλοκ τα οποία εν μέρη είναι απαραίτητα. Μπορούμε να διαλέξουμε μεταξύ των διαφόρων τύπων των μπλοκ, ανάλογα με την εφαρμογή:

- *Μπλοκ οργάνωσης (OB).*

Τα προαναφερόμενα μπλοκ συμβάλουν στην επικοινωνία μεταξύ του λειτουργικού συστήματος και του προγράμματος του χρήστη. Οι κεντρικές μονάδες επεξεργασίας καλούν τα μπλοκ οργάνωσης όταν συγκεκριμένα γεγονότα λαμβάνουν χώρα, π.χ. στην περίπτωση διακοπής. Το κυρίως πρόγραμμα βρίσκεται στο μπλοκ οργάνωσης OB1. Τα άλλα μπλοκ οργάνωσης έχουν συγκεκριμένους αριθμούς βασισμένους στο είδος των γεγονότων, τα οποία καλούνται να χειριστούν.

- *Μπλοκ λειτουργίας (FB).*

Αποτελούν μέρος του προγράμματος του οποίου οι κλήσεις μπορούν να προγραμματιστούν μέσω παραμέτρων του μπλοκ. Οι μεταβλητές μνήμης που περιέχονται σε ένα μπλοκ δεδομένων το οποίο με την σειρά του περιλαμβάνεται στην **κλήση** του μπλοκ λειτουργίας. Επίσης είναι δυνατόν σε κάθε κλήση να περιέχεται και διαφορετικό μπλοκ δεδομένων (με την ίδια δομή δεδομένων άλλα διαφορετικές τιμές μεταβλητών).

- *Λειτουργία (Fc).*

Οι λειτουργίες χρησιμοποιούνται για το προγραμματισμό περίπλοκων αυτόματων λειτουργιών. Μπορούν να παραμετροποιηθούν και να επιστρέψουν μια τιμή στο καλούμενο μπλοκ. Η τιμή της λειτουργίας είναι προαιρετική ενώ οι λειτουργίες μπορούν, επίσης, να στέλνουν προς τα έξω διαφορετικές παραμέτρους. Οι λειτουργίες δεν αποθηκεύουν πληροφορίες και δεν περιέχουν μπλοκ δεδομένων.

- *Μπλοκ δεδομένων (DB).*

Αυτά τα μπλοκ περιέχουν τα δεδομένα του προγράμματός μας. Προγραμματίζοντάς τα καθορίζουμε σε ποια μορφή θα σωθούν τα δεδομένα (σε ποιο μπλοκ, με ποια σειρά και με τι τύπο δεδομένων). Υπάρχουν δύο τρόποι χρήσης των μπλοκ δεδομένων: ως καθολικά και ως στιγμιαία μπλοκ. Ένα καθολικό μπλοκ δεδομένων είναι ένα «ελεύθερο» μπλοκ μέσα στο πρόγραμμα του χρήστη και δεν περιέχεται σε ένα κωδικοποιημένο μπλοκ. Ένα στιγμιαίο μπλοκ δεδομένων όμως, περιέχεται σε ένα μπλοκ λειτουργίας και αποθηκεύει μέρος των δεδομένων του μπλοκ λειτουργίας.

Ο αριθμός των μπλοκ ανά τύπο μπλοκ και το μήκος τους εξαρτάται από την CPU. Οι αριθμοί των μπλοκ οργάνωσης και το πλήθος τους είναι καθορισμένα. Αναθέτονται από το λειτουργικό σύστημα της κεντρικής μονάδας. Μπορούμε να ορίσουμε μόνοι μας τον αριθμό του μπλοκ των άλλων ειδών των μπλοκ, αρκεί αυτός να βρίσκεται μέσα σε καθορισμένα όρια. Επίσης έχουμε την επιλογή να ονομάσουμε κάθε μπλοκ μέσω του πίνακα συμβόλων και στη συνέχεια να αναφερόμαστε σ' αυτά με το όνομά τους.

## **Μπλοκ συστήματος**

Τα μπλοκ συστήματος αποτελούν μέρος του λειτουργικού συστήματος. Μπορούν να περιέχουν προγράμματα (λειτουργίες συστήματος) ή μπλοκ λειτουργιών ή δεδομένα (μπλοκ δεδομένων συστήματος). Τα μπλοκ συστήματος πραγματοποιούν έναν αριθμό από σημαντικές λειτουργίες του συστήματος, προσβάσιμες στο χρήστη, όπως είναι ο χειρισμός του εσωτερικού ρολογιού της CPU, ή οι διάφορες λειτουργίες επικοινωνίας.

Μπορούμε να καλέσουμε τις λειτουργίες του συστήματος και τα μπλοκ λειτουργιών του συστήματος, αλλά δεν μπορούμε να τα διαμορφώσουμε ή να τα προγραμματίσουμε. Τα μπλοκ από μόνα τους δεν διατηρούν χώρο στην μνήμη. Μόνο οι κλήσεις των μπλοκ και τα στιγμιαία μπλοκ δεδομένων των μπλοκ λειτουργιών του συστήματος είναι στην μνήμη.

### **1.9. Δομή μπλοκ**

Συνήθως τα μπλοκ αποτελούνται από τρία μέρη :

- Τον αριθμό του μπλοκ που περιέχει τις ιδιότητες του μπλοκ, όπως το όνομα του.
- Το μέρος των δηλώσεων όπου οι τοπικές μεταβλητές του μπλοκ δηλώνονται.
- Το μέρος του προγράμματος όπου περιέχονται οι εντολές του προγράμματος.

Ένα μπλοκ δεδομένων έχει παρόμοια μορφή:

- Την επικεφαλίδα του μπλοκ που περιέχει τις ιδιότητες του μπλοκ.
- Το μέρος των δηλώσεων που περιέχει τις δηλώσεις των τοπικών λειτουργιών του μπλοκ, στην περίπτωση αυτή τις διευθύνσεις των δεδομένων με τους τύπους των δεδομένων.
- Το μέρος της αρχικοποίησης, στο οποίο μπορούν να οριστούν αρχικές τιμές για μεμονωμένες διευθύνσεις δεδομένων.

### **1.10. Μορφές προγραμματισμού**

Για να αναλύσουμε έναν περίπλοκο αυτοματισμό θα πρέπει να χωρίσουμε την εφαρμογή σε μικρότερα μέρη ανάλογα με την δομή της διαδικασίας που πρέπει να ελεγχθεί. Μετά μπορούμε να διαμορφώσουμε τα επιμέρους κομμάτια καθορίζοντας τις λειτουργίες και διοχετεύοντας τα εσωτερικά σήματα προς την διαδικασία ή άλλα μέρη. Αυτός ο διαχωρισμός μπορεί να εφαρμοστεί και στον προγραμματισμό μας. Μ' αυτόν τον τρόπο η δομή του προγράμματός μας ανταποκρίνεται στον διαχωρισμό της εφαρμογής.

Ένα τέτοιο πρόγραμμα μπορεί να διαμορφωθεί πιο εύκολα και να προγραμματιστεί σε μέρη, ακόμα και από διαφορετικά άτομα, στην περίπτωση που το πρόγραμμα είναι πολύ μεγάλο. Τέλος, χωρίζοντας το πρόγραμμα σε μέρη είναι πιο εύκολη η δοκιμή και η αποσφαλμάτωση του. Η δομή του προγράμματος του χρήστη εξαρτάται από το μέγεθος και τις λειτουργίες του.

Οι μορφές προγραμματισμού είναι οι εξής:

- *Γραμμικός προγραμματισμός.*

Εδώ όλο το κυρίως πρόγραμμα είναι το μπλοκ οργάνωσης OB1. Κάθε τρέχον μονοπάτι είναι σε ξεχωριστό network. Όταν διορθώνουμε και αποσφαλματώνουμε, μπορούμε να αναφέρουμε το κάθε network απευθείας από τον αριθμό του.

- *Μερικός προγραμματισμός.*

Ο μερικός προγραμματισμός βασίζεται στον γραμμικό προγραμματισμό μόνο που το πρόγραμμα χωρίζεται σε μπλοκ. Οι αιτίες για τον διαχωρισμό του προγράμματος σε μικρότερα μέρη είναι είτε το γεγονός ότι το πρόγραμμα είναι πολύ μεγάλο για το OB1, είτε επειδή θέλουμε να διαβάζεται πιο εύκολα. Τα μπλοκ τότε καλούνται με την σειρά. Μπορούμε επίσης να χωρίσουμε το πρόγραμμα ενός μπλοκ σε άλλα μπλοκ όπως κάναμε με το OB1. Αυτή η μέθοδος μας επιτρέπει να καλούμε συσχετισμένες λειτουργίες της διαδικασίας μέσα από ένα και το αυτό μπλοκ. Το πλεονέκτημα αυτής της μορφής προγραμματισμού είναι ότι αν και το πρόγραμμα είναι γραμμικό μπορούμε να το αποσφαλματώσουμε σε μέρη (απλά μόνο καλώντας τα μπλοκ).

- *Δομημένος προγραμματισμός.*

Ο δομημένος προγραμματισμός χρησιμοποιείται όταν το επινοημένο σχέδιο είναι εξαιρετικά ακριβό, όταν θέλουμε να δημιουργήσουμε λειτουργίες προγράμματος και όταν πρέπει να λυθούν περίπλοκα προβλήματα. Μ' αυτήν την μέθοδο χωρίζουμε το πρόγραμμα σε κομμάτια (μπλοκ) με ενσωματωμένες λειτουργίες ή σε μπλοκ που εξυπηρετούν έναν συγκεκριμένο σκοπό λειτουργίας και τα οποία ανταλλάσσουν όσο το δυνατόν λιγότερα σήματα με τα άλλα μπλοκ. Αναθέτοντας σε κάθε κομμάτι μια συγκεκριμένη λειτουργία δημιουργούμε ευανάγνωστα μπλοκ με απλούστερη επικοινωνία με τα άλλα μπλοκ.

Τέλος, η οργάνωση του προγράμματος καθορίζει την σειρά με την οποία η κεντρική μονάδα επεξεργασίας θα εκτελέσει τα μπλοκ που έχουμε δημιουργήσει. Για να οργανώσουμε το πρόγραμμά μας, προγραμματίζουμε τις κλήσεις των μπλοκ με την σειρά που επιθυμούμε. Η σειρά αυτή θα πρέπει να είναι ανάλογη με την σειρά των επιμέρους λειτουργιών της διαδικασίας που θέλουμε να ελέγξουμε.

## **1.11. Γλώσσες προγραμματισμού**

Οι ελεγκτές προγραμματίζονται συνήθως σε μια (ή περισσότερες) από τις παρακάτω πια διεθνώς τυποποιημένες μορφές γλωσσών:

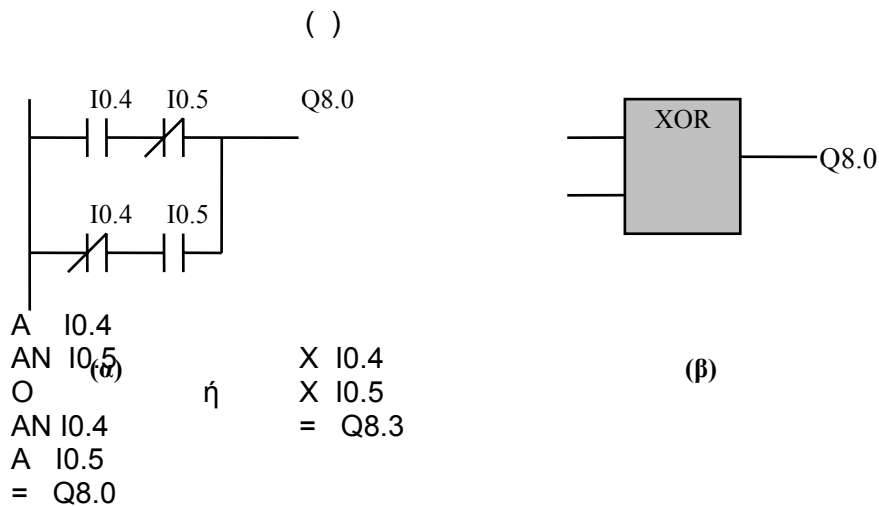
- **Λίστα εντολών**  
(STATEMENT LIST = **STL**)
- **Σχέδιο επαφών**  
(LADDER DIAGRAM = **LAD**)
- **Λογικό διάγραμμα**  
(FUNCTION BLOCK DIAGRAM = **FBD**)

Οι μορφές αυτές έχουν τυποποιηθεί κατά DIN και IEC και αποτελούν κατά κάποιον τρόπο τη «γλώσσα» στον προγραμματιζόμενο αυτοματισμό. Μπορούμε να

πούμε από την αρχή, ότι οι μορφές σχεδιασμού επαφών (LAD) και λογικού διαγράμματος (FBD) είναι γραφικές μορφές παράστασης, δηλ. το πρόγραμμα «ζωγραφίζεται» πάνω στην οθόνη μιας συσκευής προγραμματισμού.

Το σχέδιο επαφών χρησιμοποιεί λίγο πολύ σύμβολα του κλασσικού συνδεσμολογικού σχεδίου, π.χ. επαφές, πηνία, κτλ. Αντίθετα, το λογικό διάγραμμα χρησιμοποιεί σύμβολα λογικών πυλών, π.χ. πύλη AND, πύλη OR, κλπ.

Παρακάτω φαίνεται η αντιστοιχία που υπάρχει ανάμεσα στις τρεις μορφές προγραμματισμού.



Σχ. 1.6. (α) πρόγραμμα σε LAD, (β) πρόγραμμα σε FBD, (γ) πρόγραμμα σε STL.

(γ)

## 1.12. Σύγκριση μορφών προγραμματισμού

Η «μητρική» γλώσσα κάθε ελεγκτή είναι αναμφίβολα η λίστα εντολών, η οποία έχει και τις μεγαλύτερες δυνατότητες και ευελιξία. Οποσδήποτε, και οι δύο γραφικές μορφές (σχέδιο επαφών, λογικό διάγραμμα) έχουν το μεγάλο πλεονέκτημα της καλύτερης εποπτείας «με μία ματιά». Στη συνέχεια θα επιχειρήσουμε μια παρουσίαση των σημαντικότερων πλεονεκτημάτων και μειονεκτημάτων.

### 1.12.1. Πλεονεκτήματα λίστας εντολών (STL) σε σχέση με τις γραφικές μορφές (LAD, FBD)

- Έχει τις μεγαλύτερες δυνατότητες, γιατί υπάρχουν εντολές, οι οποίες δεν είναι δυνατόν να παρασταθούν γραφικά, αν και στο κοντινό μέλλον αυτό θα διορθωθεί.

- Γνωρίζουμε με απόλυτη ακρίβεια τη σειρά, με την οποία ο μικροεπεξεργαστής επεξεργάζεται το πρόγραμμα (τη μία εντολή ύστερα από την άλλη).
- Καταλαμβάνει μικρότερο χώρο στη μνήμη για την αποθήκευση του προγράμματος.
- Είναι πολύ προσιτή στην χρήση σε όποιον έχει ασχοληθεί ήδη με προγραμματισμό κάθε είδους.
- Μπορούν να χρησιμοποιηθούν μικροί, φτηνοί, φορητοί προγραμματιστές χειρός (ενώ αντίθετα για τη «σχεδίαση» μιας γραφικής μορφής απαιτείται οθόνη, αν θέλουμε να έχουμε εποπτεία).
- Ο χειρισμός κατά την πληκτρολόγηση του προγράμματος είναι πολύ απλούστερος. Αντίθετα, για την πληκτρολόγηση ενός στοιχείου στις γραφικές μορφές, π.χ. μιας επαφής, πρέπει ο δείκτης (cursor) στην οθόνη να βρίσκεται στη σωστή θέση.
- Αν σαν βάση για τον προγραμματισμό χρησιμοποιηθεί ένα κλασσικό συνδεσμολογικό σχέδιο με ρελέ ή ένα λογικό διάγραμμα (flow-chart), τότε η «μετάφραση» τους σε λίστα εντολών είναι το ίδιο εύκολη με την «μετάφραση» τους σε σχέδιο επαφών ή λογικό διάγραμμα αντίστοιχα (αν όχι ευκολότερη πολλές φορές).
- Πρέπει να τονιστεί, ότι ένα ηλεκτρολογικό συνδεσμολογικό σχέδιο, πολύ σπάνια μπορεί να προγραμματιστεί όπως είναι, χωρίς μετατροπές, σε σχέδιο επαφών.

### **1.12.2. Μειονεκτήματα λίστας εντολών (STL) σε σχέση με τις γραφικές μορφές (LAD, FBD)**

- Ένα πρόγραμμα γραμμένο σε λίστα εντολών δεν έχει την ίδια εποπτεία «με μια ματιά», την οποία έχουν οι γραφικές μορφές.  
Με τις δυνατότητες όμως σχολιασμού προγράμματος, που παρέχουν οι σύγχρονες συσκευές προγραμματισμού, το μειονέκτημα αυτό παύει να είναι ιδιαίτερα σημαντικό.
- Η παρακολούθηση του αυτοματισμού σε λειτουργία (πάνω σε μια συσκευή προγραμματισμού οθόνης συνδεδεμένη στον ελεγκτή) είναι απλούστερη και πιο εποπτική, αν το πρόγραμμα είναι γραμμένο σε κάποια από τις δύο γραφικές μορφές.

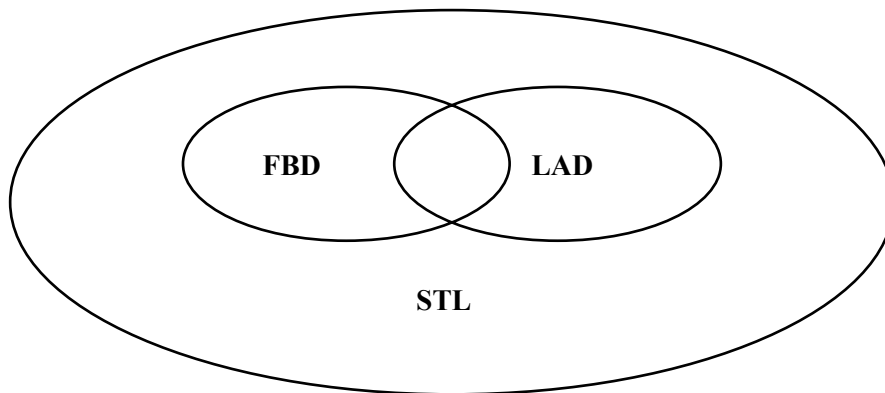
### **1.12.3. Συμπεράσματα**

Καλό είναι οι ελεγκτές να έχουν τη δυνατότητα να προγραμματιστούν και στις τρεις μορφές που προαναφέρθηκαν και να αφήνεται σ' αυτόν που θα φτιάξει το πρόγραμμα η επιλογή της μορφής προγραμματισμού. Θεωρείται αυτονόητο ότι οι τρεις μορφές είναι συμβατές μεταξύ τους, δηλ. σε όποια μορφή κι αν προγραμματίσουμε, έχουμε τη δυνατότητα να πάρουμε το πρόγραμμα και στις άλλες δύο, ζητώντας το από τη συσκευή.

Η χρήση περισσότερων από μία μορφή παράστασης ενός προγράμματος είναι πολλές φορές επιθυμητή και για άλλους λόγους:

Π.χ. σ' ένα μεγάλο εργοστάσιο, αυτός που θα φτιάξει το πρόγραμμα μπορεί να επιλέξει π.χ. τη λίστα εντολών, αλλά η ηλεκτρολογική συντήρηση πιθανόν να προτιμάει στο αρχείο της την παράσταση σχεδίου επαφών, για την ανεύρεση βλαβών.

Όσοι έχουν εμπειρία στον τομέα του αυτοματισμού επιλέγουν συνήθως τη μορφή προγραμματισμού που ταιριάζει καλύτερα στην εμπειρία τους, λαμβάνοντας υπ' όψη και τα πλεονεκτήματα - μειονεκτήματα που προαναφέραμε.



Σχ. 1.7. Συμβατότητα μεταξύ των γλωσσών.

### 1.13. Θέση σε λειτουργία ελεγκτή προγραμματιζόμενης λογικής

Η θέση σε λειτουργία χωρίζεται σε δύο τελείως ξεχωριστά μέρη:

- Θέση σε λειτουργία και έλεγχος όλων των μονάδων του ελεγκτή (HARDWARE).  
Μπορεί να γίνει και από μη ειδικευμένο προσωπικό και προηγείται πάντοτε οποιοδήποτε ελέγχου.
- Θέση σε λειτουργία του προγράμματος (SOFTWARE).  
Γίνεται οπωσδήποτε από άτομο με γνώσεις προγραμματισμού.

#### 1.13.1. Θέση σε λειτουργία και έλεγχος των μονάδων ενός ελεγκτή (HARDWARE)

Συνήθως, ο ελεγκτής παραδίνεται επί τόπου του έργου, τοποθετημένος και συρματωμένος μέσα σ' έναν πίνακα. Όλες οι εισοδοί και οι εξοδοί, που πρόκειται να συνδεθούν με την εξωτερική εγκατάσταση, πρέπει να είναι συρματωμένες σε κλέμες. Ο πίνακας περιέχει, συνήθως, και τα τροφοδοτικά για τις τάσεις εισόδων - εξόδων, με τους αντίστοιχους μικροαυτόματους κατανομής προς τις μονάδες.

Η θέση σε λειτουργία των μονάδων ενός ελεγκτή διακρίνεται σε 4 διαδοχικά στάδια:

- Προετοιμασία.
- Έλεγχος μονάδας τροφοδοσία και CPU.
- Έλεγχος των εισόδων και της συρμάτωσης τους.
- Έλεγχος των εξόδων και της συρμάτωσης τους.

#### Προετοιμασία

1. Βγάζουμε τις ασφάλειες από όλα τα κυκλώματα ισχύος (π.χ. κινητήρες ) και «αδρανοποιούμε» τα υδραυλικά κυκλώματα, ώστε αν τυχαία οπλίσει ένα ρελέ να μην πάρει μπρος ο αντίστοιχος κινητήρας.
2. Βγάζουμε εκτός όλους τους μικροαυτόματους, οι οποίοι τροφοδοτούν τον ελεγκτή και τις εισόδους / εξόδους του .
3. Ξεσυρματώνουμε και τραβάμε έξω τις πρίζες καλωδίων από όλες τις μονάδες εισόδων - εξόδων.
4. Ελέγχουμε, βάσει του σχεδίου, αν σε κάθε θέση του πλαισίου υπάρχει ο σωστός τύπος μονάδας με τις σωστές διευθύνσεις.

#### Έλεγχος μονάδας τροφοδοσίας (PS) και κεντρικής μονάδας (CPU)

1. Ελέγχουμε αν η τάση του δικτύου έχει σωστή τιμή, αντίστοιχη με την ονομαστική τάση της μονάδας τροφοδοσίας. Αν υπάρχει επιλογικός διακόπτης τάσης, εξετάζουμε αν είναι στη σωστή θέση . Βάζουμε εντός τον μικροαυτόματο παροχής τροφοδοσίας και ελέγχουμε, αν τα LED ένδειξης των εσωτερικών τάσεων του ελεγκτή πάνω στη μονάδα τροφοδοσίας ανάβουν όλα.
2. Για την κεντρική μονάδα μπορούμε να κάνουμε ένα απλό τεστ, το οποίο λέει χονδρικά, αν η μονάδα είναι εντάξει.

**Προσοχή!** Το τεστ που ακολουθεί μπορεί να εφαρμοστεί μόνο αν είμαστε σίγουροι ότι δεν υπάρχει πρόγραμμα στη μνήμη. Σε αντίθετη περίπτωση το πρόγραμμα χάνεται.

**Τεστ:** Συνδέουμε μια συσκευή προγραμματισμού και προσπαθούμε να σβήσουμε όλη τη μνήμη, με τους χειρισμούς που λέει ο κατασκευαστής. Σε πολλούς τύπους ελεγκτών το σβήσιμο αυτό είναι δυνατό και χωρίς συσκευή προγραμματισμού, απλά και μόνο με χειρισμούς πάνω στους διακόπτες της μονάδας CPU.

Στη συνέχεια, πηγαίνουμε το διακόπτη της κεντρικής μονάδας στη θέση «RUN». Αν ανάψει το αντίστοιχο LED, σημαίνει ότι η μονάδα μπαίνει σε κύκλο λειτουργίας, άρα με μεγάλη πιθανότητα είναι εντάξει.

#### Έλεγχος των εισόδων και της συρμάτωσης τους

1. Βάζουμε εντός τους μικροαυτόματους τροφοδοσίας των μονάδων εισόδων.
2. Διεγείροντας έναν - έναν τερματικό, μπουτόν, διακόπτη κ.λπ. (π.χ. S1), ελέγχουμε με **πολύμετρο**, αν φτάνει η σωστή τάση στη σωστή κλέμα της τραβηγμένης πρίζας καλωδίων. Ελέγχουμε επίσης, αν φτάνει η τάση L- βάσει του σχεδίου. Έτσι βεβαιωνόμαστε για τον **Έλεγχο** των εισόδων και της συρμάτωσης τους.
3. Βάζουμε τις πρίζες καλωδίων των μονάδων εισόδων στη θέση τους και συνδέουμε μια συσκευή προγραμματισμού στην κεντρική μονάδα. Επιλέγουμε τη λειτουργία «**Έλεγχος κατάστασης στοιχείου**» (STATUS VAR).

Γεφυρώνουμε διαδοχικά το μπαράκι κατανομής δυναμικού προς την εγκατάσταση με μία - μία κλέμα εισόδου πάνω στις πρίζες καλωδίων. Παρατηρούμε αν ανάβει το αντίστοιχο LED εισόδου και αν το αντίστοιχο σήμα της εισόδου στη



συσκευή προγραμματισμού γίνεται «1». Αν δεν υπάρχει αντιστοιχία: «Τάση στην κλέμα» - «LED αναμμένο» = Σήμα εισόδου «1», η υπό έλεγχο είσοδος δεν λειτουργεί σωστά.

Έτσι βεβαιωνόμαστε για τη σωστή λειτουργία των μονάδων εισόδου.

### **Σημείωση 1**

Μελετώντας το ηλεκτρολογικό σχέδιο του συστήματος διαπιστώνουμε, ότι η κατάσταση «LED εισόδου αναμμένο» δεν σημαίνει τίποτε άλλο, παρά το ότι η τάση φτάνει στην αντίστοιχη κλέμα της μονάδας εισόδων.

Ότι η τάση αυτή μετατρέπεται σωστά και «περνάει» στο εσωτερικό του συστήματος ελέγχεται μόνο με τη συσκευή προγραμματισμού, αν επιλέξουμε τη λειτουργία που αναφέραμε.

### **Σημείωση 2**

Από τα προηγούμενα διαπιστώσαμε ότι χρειάστηκε πρακτικά να γίνει η ίδια δουλειά δύο φορές, για τον έλεγχο της συρμάτωσης και της σωστής λειτουργίας εισόδων.

Πολλές φορές στην πράξη αυτό γίνεται σε μία φάση χωρίς τράβηγμα των πριζών καλωδίων και χωρίς χρήση του πολυμέτρου. Διεγείροντας τον τερματικό, παρατηρούμε το LED και το σήμα της εισόδου, αν έχουμε συνδέσει τον προγραμματιστή.

Η μέθοδος αυτή δεν συνιστάται, γιατί μπορεί να προκαλέσει καταστροφή της μονάδας αν έχει γίνει λάθος συρμάτωση.

Σε περίπτωση που δεν υπάρχει, λοιπόν, διαθέσιμος χρόνος, κάνουμε μόνο έλεγχο της συρμάτωσης, όπως περιγράψαμε. Τυχόν λανθασμένη λειτουργία κάποιας εισόδου θα ανακαλυφθεί κατά τον έλεγχο του προγράμματος.

### Έλεγχος των εξόδων και της συρμάτωσης τους

1. Βάζουμε «εντός» όλους τους μικροαυτόματους τροφοδοσίας των μονάδων εξόδων.
2. Ελέγχουμε, βάσει του σχεδίου, αν φτάνει η σωστή τάση τροφοδοσίας στην κλέμα τροφοδοσίας των μονάδων εξόδων, π.χ. κλέμα 1 της τραβηγμένης πρίζας καλωδίων. Ελέγχουμε επίσης, αν φτάνει και η τάση L- (0V) στην αντίστοιχη κλέμα, π.χ. κλέμα 11.
3. Βάζουμε τις πρίζες καλωδίων των μονάδων εξόδων στη θέση τους και συνδέουμε τη συσκευή προγραμματισμού στην κεντρική μονάδα.
4. Χρησιμοποιώντας τη λειτουργία *Εξαναγκασμός σήματος (Force)*, εξαναγκάζουμε διαδοχικά σήμα «1» σε κάθε έξοδο. Πρέπει να ανάβει το αντίστοιχο LED εξόδου και να διεγείρεται το αντίστοιχο στοιχείο που δέχεται την εντολή (π.χ. ρελέ -K1).

Έτσι βεβαιωνόμαστε ταυτόχρονα και για τη σωστή συρμάτωση και για τη σωστή λειτουργία των εξόδων.

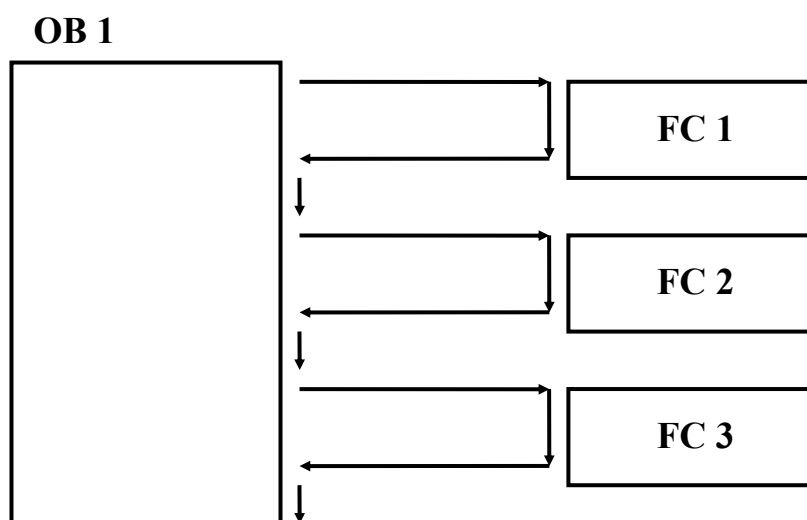
### **Σημείωση**

Αν δεν διαθέτουμε συσκευή προγραμματισμού, μπορούμε να κάνουμε έλεγχο μόνο της σωστής συρμάτωσης, ως εξής:

Χωρίς να βάλουμε τις πρίζες καλωδίων στη σωστή θέση τους, γεφυρώνουμε την κλέμα τροφοδοσίας κάθε μονάδας εξόδων (π.χ. κλέμα 1 της πρίζας καλωδίων) με μία - μία κλέμα εξόδου (π.χ. κλέμα 4) και παρατηρούμε αν διεγείρεται το αντίστοιχο στοιχείο (π.χ. ρελέ -K1).

### **1.13.2. Θέση σε λειτουργία του προγράμματος (SOFTWARE)**

Η θέση σε λειτουργία του προγράμματος διευκολύνεται πολύ, αν έχει γίνει χρήση του δομημένου προγραμματισμού. Έστω λοιπόν, ότι η δομή του προγράμματος είναι η παρακάτω:



Σχ. 1.8. Παράδειγμα δομής προγράμματος.

Τα βήματα που ακολουθούμε είναι τα εξής:

1. Μεταφέρουμε στην μνήμη του ελεγκτή όλες τις λειτουργίες (Fc).
2. Το OB 1 το προγραμματίζουμε κατευθείαν στην μνήμη, γράφοντας αρχικά μόνο μια κλήση.
3. Πηγαίνουμε το σύστημα σε RUN και ελέγχουμε μεμονωμένα κάθε μια από τις λειτουργίες (Fc). Τυχόν σήματα που προέρχονται από άλλα μπλοκ, εξομοιώνονται με την λειτουργία της συσκευής προγραμματισμού Έλεγχος - Εξαναγκασμός Σήματος. Άλλωστε όλα τα μπλοκ που δεν έχουν κληθεί για επεξεργασία από το OB 1, παραμένουν αδρανή μέσα στην μνήμη.
4. Εφόσον ελέγξουμε και διορθώσουμε τις λειτουργίες (Fc), τα μπλοκ (FB), κ.τ.λ. προσθέτουμε μια ακόμα κλήση Fc ή μπλοκ στο OB1.
5. Συνεχίζουμε κατά τον ίδιο τρόπο προσθέτοντας κλήσεις στο OB 1, μέχρι να γίνει έλεγχος όλων των λειτουργιών και μπλοκ, μαζί.

#### **1.14. Σύγκριση μεταξύ κλασσικού αυτοματισμού και μοντέρνου (PLC)**

Σε μια αυτοματοποιημένη παραγωγική διαδικασία (π.χ. ένα τριβείο μεταλλεύματος, ένα σύστημα διακίνησης ή μία μηχανή) διακρίνουμε τα εξής βασικά μέρη, αν ο αυτοματισμός έχει γίνει με το κλασσικό τρόπο.

- i. Ηλεκτρολογικός εξοπλισμός της διαδικασίας (τερματικοί διακόπτες, αναλογικά αισθητήρια, κινητήρες, βαλβίδες).
- ii. Πίνακας ισχύος για την τροφοδοσία των κινητήρων και των βαλβίδων.
- iii. Πίνακας αυτοματισμού με βοηθητικά ρελέ - χρονικά.
- iv. Πίνακας χειρισμών (μπουτόν, διακόπτες).
- v. Μιμικό διάγραμμα (απεικόνιση, λυχνίες, αναλογικά όργανα).

Η «καρδιά» όλων των ανωτέρω είναι ο πίνακας αυτοματισμού ο οποίος :

- Δέχεται εντολές (σήματα) από τα αισθητήρια της εγκατάστασης, από το χειριστήριο και από τον πίνακα ισχύος (βοηθητικές επαφές των ρελέ ισχύος).
- Επεξεργάζεται τις εντολές αυτές με βάση κάποιο συνδεσμολογικό σχέδιο (λειτουργία αυτοματισμού).
- Στέλνει εντολές ενεργοποίησης προς τον πίνακα ισχύος, για την ζεύξη των κινητήρων και των βαλβίδων, και προς το μιμικό διάγραμμα, για ένδειξη στις λυχνίες και τα αναλογικά όργανα.

Το πρώτο βήμα βελτίωσης της «ποιότητας» είναι η αντικατάσταση του κλασσικού πίνακα αυτοματισμού με τον ελεγκτή προγραμματιζόμενης λογικής. Η βασική διαφοροποίηση είναι ότι η λειτουργία του αυτοματισμού δεν «συρματώνεται»

αλλά «προγραμματίζεται» με κάποια συσκευή προγραμματισμού, στην μνήμη της συσκευής αυτοματισμού, με όλα τα σχετικά πλεονεκτήματα, όπως:

- Αξιοπιστία λόγω μη κινούμενων μερών.
- Περισσότερες δυνατότητες, μη πραγματοποιήσιμες με ρελέ.
- Ευκολία αλλαγών χωρίς αλλαγή της συρμάτωσης.
- Δεν υπάρχουν «ανενημέρωτα» σχέδια, γιατί το τρέχον πρόγραμμα είναι μέσα στη μνήμη και μπορεί να διαβαστεί ή να τυπωθεί ανά πάσα στιγμή.
- Στη μελέτη δεν υπάρχει το πρόβλημα αν επαρκούν οι επαφές των ρελέ, των χρονικών ή των εξωτερικών τερματικών.
- Διευκόλυνση στον εντοπισμό εξωτερικών βλαβών λόγω υπάρξεως LED σε κάθε είσοδο, έξοδο.

Το επόμενο βήμα βελτίωσης, το οποίο μελετάται στη συνέχεια, είναι η αντικατάσταση του χειριστηρίου από ένα πληκτρολόγιο και του μιμικού διαγράμματος από ένα έγχρωμο MONITOR. Τα νέα αυτά στοιχεία συνδέονται σε μια κάρτα επικοινωνίας CP (Communicator Processor). Η κάρτα αυτή έχει δικό της μικροεπεξεργαστή και μνήμη για την αποθήκευση των εικόνων και των σχετικών λειτουργιών.

Έτσι η όλη διαδικασία της παρακολούθησης και του χειρισμού γίνεται αυτόνομα χωρίς «επιβάρυνση» της CPU του ελεγκτή (ταχύτερη επεξεργασία).

Ένα ακόμα βήμα βελτίωσης έγκειται στη προσθήκη κάποιων συσκευών εκτύπωσης, με σκοπό την δυνατότητα καταγραφής χρήσιμων στοιχείων που σχετίζονται άμεσα με την διαδικασία. Συνήθως απαιτείται η καταγραφή του χρόνου λειτουργίας κάποιων κινητήρων, φωτιστικών, κ.α. Παράλληλα, είναι πολύ θετικό να γίνεται ενημέρωση μέσω του εκτυπωτή για την ύπαρξη βλάβης σε κάποιο σημείο της διαδικασίας ή την αποκατάσταση αυτής, με ταυτόχρονη εμφάνιση ώρας και ημερομηνίας για εκτίμηση της συχνότητας με την οποία επαναλαμβάνονται τα διάφορα σφάλματα. Η σύνδεση του εκτυπωτή με το PLC επιτυγχάνεται με τη χρήση της ειδικής κάρτας CP.

Στις μέρες μας εξαιτίας της περιπλοκότητας των, βιομηχανικών και μη, διεργασιών που καλείται να ελέγξει ο ελεγκτής προγραμματιζόμενη λογικής, έχουν αναπτυχθεί σε μεγάλο βαθμό τα **Συστήματα Εποπτικού Ελέγχου**.