



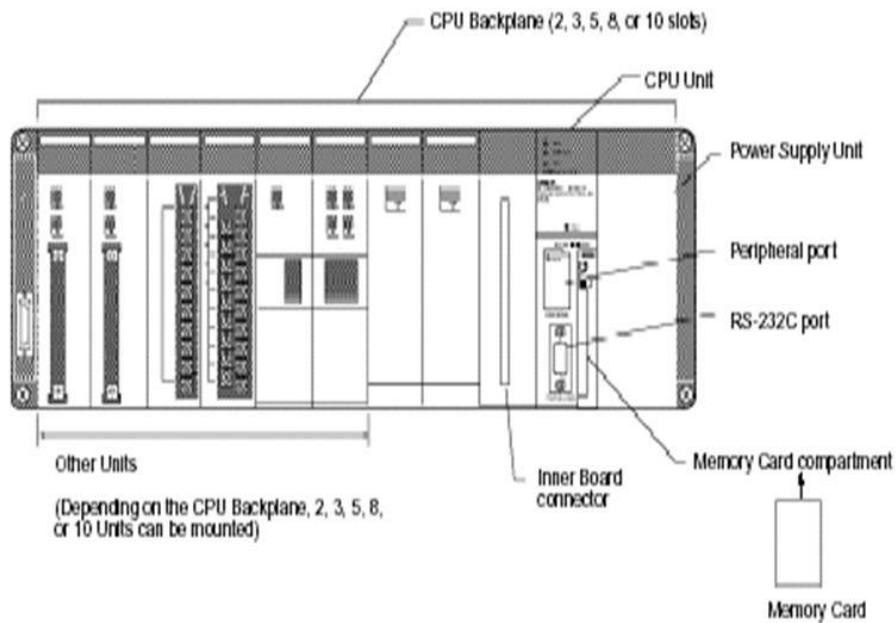
Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Αυτοματισμού

Πτυχιακή Εργασία

Θέμα:

«Δομή και λειτουργία των PLC»



ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΜΙΧΑΛΗΣ ΠΑΠΟΥΤΣΙΔΑΚΗΣ

ΦΟΙΤΗΤΗΣ: ΑΚΡΙΔΑΣ ΑΝΤΩΝΙΟΣ Α.Μ.: 11606

ΑΙΓΑΛΕΩ, ΜΑΪΟΣ 2014

«Δηλώνω υπεύθυνα ότι το παρόν κείμενο αποτελεί προϊόν προσωπικής μελέτης και εργασίας και πως όλες οι πηγές που χρησιμοποιήθηκαν για τη συγγραφή της δηλώνονται σαφώς είτε στις παραπομπές είτε στη βιβλιογραφία. Γνωρίζω πως η λογοκλοπή αποτελεί σοβαρότατο παράπτωμα και είμαι ενήμερος/η για την επέλευση των νομίμων συνεπειών»

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Αθήνα

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣ

- 1.
- 2.
- 3.

Περιεχόμενα

Κεφάλαιο 1 Εισαγωγή	1
1.1 Ιστορική Αναδρομή.....	1
1.2 Πλεονεκτήματα της χρήσης αυτοματισμών	4
Κεφάλαιο 2 - PLC6	
2.1 Ιστορική εξέλιξη του βιομηχανικού αυτοματισμού	7
2.2 Ιστορική εξέλιξη των PLC.....	8
2.3 Βασικά χαρακτηριστικά ενός PLC	9
2.3.1 Κεντρική μονάδα (CPU).....	10
2.3.2 Μνήμη.....	11
2.3.2.1 Μη πτητικές μνήμες.....	11
2.3.2.2 Πτητικές μνήμες.....	12
2.3.3 Μονάδες εισόδου – εξόδου	13
2.3.3.1 Μονάδα εισόδων.....	16
2.3.3.2 Μονάδα εξόδων	16
2.3.3.3 Ειδικές I/O μονάδες.....	17
2.3.4 Τροφοδοτικό (Power Supply).....	19
2.3.5 Το πλαίσιο στήριξης μονάδων	20
2.3.6 Βοηθητικές μονάδες.....	20
2.3.7 Θύρα επικοινωνίας	20
2.3.7.1 Καλώδια επικοινωνίας.....	21
2.3.7.1.1 RS-232.....	21
2.3.7.1 .2 Αναφορά στο RS-485.....	22
2.3.8 Ενδεικτικά LED	22
2.3.9 Συσκευή προγραμματισμού.....	22
2.4 Τύποι PLC.....	24
2.4.1. Μη επεκτάσιμοι ελεγκτές.....	25
2.4.2. Επεκτάσιμοι ελεγκτές.....	26
2.5 Πλεονεκτήματα του PLC	27
2.6 Μειονεκτήματα των PLCs	28
2.7 Αρχή λειτουργίας ενός Προγραμματιζόμενου Λογικού Ελεγκτή	28
2.8 Διαφορές μεταξύ προγραμματιζόμενων λογικών ελεγκτών και ηλεκτρονικών υπολογιστών.....	30
2.9 Κύριες λειτουργίες προγραμματιζόμενων λογικών ελεγκτών.....	30
Κεφάλαιο 3 - Προγραμματισμός	
3 Εισαγωγή.....	32
3.1 Προσπέλαση προγράμματος.....	33
3.1.1 Μέθοδοι προσπέλασης προγράμματος.....	33
3.2 Δομή προγράμματος.....	34
3.2.1 Τύποι μπλοκ.....	34
3.2.1.1 Μπλοκ χρήστη	34
3.2.1.2 Μπλοκ συστήματος.....	35
3.3 Μορφές προγραμματισμού.....	35
3.4 Γλώσσες προγραμματισμού.....	36
3.4.1 Μη γραφικές.....	36
3.4.1.1 Λίστα εντολών Instruction List (IL).....	36
3.4.1.1.1 Πλεονεκτήματα λίστας εντολών (STL) σε σχέση με τις γραφικές μορφές (LAD, FBD).....	43

3.4.1.1.2 Μειονεκτήματα λίστας εντολών (STL) σε σχέση με τις γραφικές μορφές (LAD, FBD).....	43
3.4.1.2 Γλώσσα δομημένου κειμένου Structured Text (ST).....	43
3.4.1.3 Γλώσσα MATRIX.....	43
3.4.2 Γραφικές.....	44
3.4.2.1 Σχέδιο επαφών Ladder Diagram(LAD).....	44
3.4.2.2 Διάγραμμα λογικών πυλών Function Block Diagram (FBD)	46
3.4.2.3 Διάγραμμα διαδοχικών λειτουργιών Sequential Function Chart (SFC).	47
3.5 Οι εντολές S (SET) και R (RESET).....	48
3.6 Ανάπτυξη προγραμμάτων με χρονικές λειτουργίες.....	51
3.6.1 Χρονικά.....	52
3.6.2Ανάλυση χρονικών.....	52
3.7 Απαριθμητές (Counters).....	57
3.7.1 Κατηγορίες απαριθμητών.....	57
3.7.2 Παράδειγμα χρήσης απαριθμητών στην πράξη.....	58
3.8 Συγκρίσεις.....	60
3.9 Αριθμητικές πράξεις.....	61
3.10 Βοηθητικές μνήμες.....	66
3.11 Συσσωρευτές (ACCUMULATORS).....	66
3.12Διακοπές κύκλου προγράμματος.....	66
3.13 Γρήγοροι απαριθμητές.....	66
3.14 ACCUMULATORS (Συσσωρευτές)	67
3.15 Προγραμματιστικά χαρακτηριστικά και ονοματολογία των στοιχείων ενός προγραμματιζόμενου λογικού ελεγκτή.....	67
Κεφάλαιο 4 - Διάφορα παραδείγματα.....	68
4.1 Έλεγχος διάβασης πεζών	68
4.2 Σύστημα συναγερμού.....	71
4.3 Ταινίες μεταφοράς υλικών.....	74
4.4 Χώρος στάθμευσης οχημάτων.....	80
4.5 Αντλιοστάσιο τριών αντλιών.....	84
Βιβλιογραφία.....	90

Κεφάλαιο 1

1.1 Ιστορική Αναδρομή

Ο αυτοματισμός είναι μια πολύ παλιά ιστορία. Και σε μεγάλο βαθμό είναι Ελληνική. Η λέξη «αυτόματο» είναι Ελληνική και τη συναντάμε κατ' αρχάς στα Ομηρικά έπη. Στην αρχαιότητα οι Έλληνες, αρχικά φαντάζονταν, οραματίζονταν αυτόματα συστήματα και στη συνέχεια οι Έλληνες μηχανικοί της αρχαιότητας μελετούσαν, σχεδίαζαν και κατασκεύαζαν αυτόματα και επιπλέον έγραφαν γι' αυτά [1],[36].

Ο αυτόματος έλεγχος, και ειδικά η εφαρμογή της ανάδρασης, ήταν θεμελιώδους σημασίας για την ανάπτυξη του αυτοματισμού. Αρχικά, η ανάγκη για ακριβή μέτρηση του χρόνου έδωσε την ώθηση για την μελέτη αυτόματων συστημάτων. Γύρω στο 270 π.Χ. στην Αλεξάνδρεια της Αιγύπτου ο Έλληνας Κτεσίβιος εφεύρε ένα είδος ρυθμιστή στάθμης για τη χρήση του σε ένα ρολόι νερού. Ο ρυθμιστής αυτός διατηρούσε σταθερή τη στάθμη του νερού μιας δεξαμενής το οποίο είχε ως αποτέλεσμα τη σταθερή ροή νερού προς μια δεύτερη δεξαμενή, η στάθμη της οποίας εξαρτιόταν από το χρόνο που είχε περάσει από την έναρξη της διαδικασίας. Χρησιμοποιήθηκαν από Έλληνες και Άραβες για να ελέγχουν συσκευές όπως κλεψύδρες, λάμπες λαδιού και διανομείς κρασιού, όπως και το επίπεδο του νερού στις δεξαμενές. Τα πιο γνωστά Ελληνικά ονόματα, που ασχολήθηκαν με τέτοια συστήματα, είναι του Κτεσίβιου, του Φίλωνα (3ος αιώνας π.Χ.) και του Χίρωνα (πρώτος αιώνας μ.Χ.), οι οποίοι δραστηριοποιήθηκαν στην ανατολική Μεσόγειο (Αλεξάνδρεια). Ακόμη, πολύ Άραβες μηχανικοί σχεδίασαν συστήματα που βασιζόταν στην αρχή της ανατροφοδότησης. Δυστυχώς, όμως, οι έρευνες τους διακόπηκαν το 1258 όταν οι Μογγόλοι κατέλαβαν τη Βαγδάτη.

Η εφεύρεση της ατμομηχανής από τον J Watt το 1769, σηματοδοτεί την έναρξη της Βιομηχανικής Επανάστασης στην Ευρώπη. Η ραγδαία εξέλιξη της τεχνολογίας έκανε δύσκολο το χειροκίνητο χειρισμό των προηγμένων, για την εποχή, μηχανημάτων. Αυτό οδήγησε στην εμφάνιση ποικίλων συσκευών ελέγχου κάποιες από τις οποίες αφορούσαν ρυθμιστές στάθμης, ρυθμιστές θερμοκρασίας, πίεσης και ταχύτητας. Το 1788 ο Watt χρησιμοποίησε τη φυγόκεντρο συσκευή αυτομάτου ελέγχου flyball (centrifugal governor) για τη ρύθμιση της ταχύτητας της περιστρεφόμενης ατμομηχανής.

Καθώς τα συστήματα ελέγχου άρχισαν να γίνονται ιδιαίτερα περίπλοκα, η μέθοδος σχεδίασης δοκιμής και λάθους συνδυασμένη με τη διαίσθηση των μηχανικών άρχισε να αποδεικνύεται ανεπαρκής. Στα μέσα του 19ου αιώνα χρησιμοποιήθηκαν για πρώτη φορά τα μαθηματικά για τον έλεγχο της ευστάθειας των συστημάτων ανατροφοδότησης. Από τις διαφορικές εξισώσεις έως το κριτήριο ευστάθειας του Routh και τις συναρτήσεις μεταφοράς, αυτή η μαθηματική μοντελοποίηση προωθήθηκε μέχρι το 1900 οπότε και κάνει την εμφάνισή της η θεωρία συστημάτων ελέγχου. Το 1840, ο Βρετανός αστρονόμος G.B. Airy ανέπτυξε μία συσκευή που έδινε τη δυνατότητα για εκτενέστερη παρατήρηση ενός άστρου με το τηλεσκόπιο. Η συσκευή διέθετε ένα σύστημα ελέγχου ταχύτητας το οποίο περιστρέφει το τηλεσκόπιο ώστε να αντιτίθεται στην αλλαγή θέσης λόγω της περιστροφής της γης. Ο Airy ήταν ο πρώτος που χρησιμοποίησε διαφορικές εξισώσεις για την μελέτη της ευστάθειας του συστήματος κλειστού βρόγχου. Λίγο αργότερα, το 1868, ο J.C. Maxwell στο έγγραφό του "On Governors", ήταν σε θέση να εξηγήσει τις αστάθειες που παρουσιάστηκαν στο σύστημα αυτομάτου ελέγχου flyball, χρησιμοποιώντας διαφορικές εξισώσεις για να περιγράψει το σύστημα ελέγχου.

Μέχρι τον 20ο αιώνα η ανάλυση των συστημάτων ελέγχου γινόταν στο πεδίο του χρόνου. Μόνο στη δεκαετία του 20 και του 30 το πεδίο της συχνότητας με τα μαθηματικά του Laplace, Fourier και Cauchy άρχισε να παίζει σημαντικό ρόλο. Το κύριο πρόβλημα στα συστήματα μαζικής επικοινωνίας ήταν η ανάγκη για περιοδική ενίσχυση των σημάτων φωνής σε μακρές τηλεφωνικές γραμμές χωρίς την ταυτόχρονη ενίσχυση και του θορύβου. Σε αυτή την περιοχή ήταν που ο H. S. Black έδειξε τη χρησιμότητα της αρνητικής ανατροφοδότησης το 1927 με τους H. Nyquist και H. W. Bode να εισάγουν νέες τεχνικές σχεδίασης την επόμενη δεκαετία. Κατά τη διάρκεια των παγκόσμιων πολέμων, η ανάπτυξη του ελέγχου με ανατροφοδότηση έγινε ένα θέμα επιβίωσης. Ήταν το κλειδί στην ανάπτυξη συστημάτων ελέγχου των πλοίων, συστημάτων πλοήγησης καθώς και ακριβείας σκόπευσης των όπλων. Επίσης, η εξάλειψη του θορύβου έγινε και πάλι σημαντικό ζήτημα κατά τη διάρκεια μελέτης των προβλημάτων της επεξεργασίας των πληροφοριών η οποία συνδέθηκε με την εφεύρεση του «ραντάρ». Η δουλειά του MIT Radiation Lab οδήγησε στην ανάπτυξη προηγμένων τεχνικών σχεδίασης.

Μετά τον δεύτερο παγκόσμιο πόλεμο αρχίζει η ηλεκτρονική εποχή. Ήδη από τις αρχές του 20ου αιώνα έχουμε τις πρώτες ηλεκτρονικές συσκευές, το ραδιόφωνο και, αργότερα, την τηλεόραση, τους ασύρματους και τα ραντάρ. Το κύριο εξάρτημα αυτών των συσκευών ήταν η ηλεκτρονική λυχνία. Η ανακάλυψη του τρανζίστορ το 1950 ήταν η αρχή της ηλεκτρονικής επανάστασης των ημιαγωγών. Το θαυματουργό αυτό στοιχείο αντικατέστησε την ακριβή, ογκώδη και ενεργειακά βόρα ηλεκτρονική λυχνία και έκανε τις ηλεκτρονικές συσκευές μικρότερες, εύκολες στην κατασκευή και απείρως πιο φθηνές.

Το 1945 κατασκευάστηκε ο πρώτος ηλεκτρονικός υπολογιστής, ο ENIAC, ο οποίος χρησιμοποιούσε λυχνίες. Ο ENIAC δεν θύμιζε σε τίποτα τους σημερινούς υπολογιστές, ήταν ένα ολόκληρο εργοστάσιο το οποίο έλυνε μαθηματικές εξισώσεις. Μετά το 1950 και με τη χρήση των τρανζίστορ έχουμε τους πρώτους πραγματικούς υπολογιστές, οι οποίοι χρησιμοποιούνται κυρίως στο θέμα της μηχανογράφησης, δηλαδή στην αποθήκευση και διαχείριση μεγάλων αρχείων δεδομένων.

Προς το τέλος της δεκαετίας του '50, Η σχεδίαση συστημάτων ελέγχου στράφηκε και πάλι προς το πεδίο του χρόνου. Μπορεί αρχικά να φαίνεται παράξενο αλλά τελικά έχει νόημα με την εισαγωγή σύνθετων, μη γραμμικών πολλών μεταβλητών συστημάτων που έχουν σχέση με εφαρμογές της αεροδιαστημικής.

Από τη δεκαετία του '60 ήδη οι μηχανικοί άρχισαν να σκέφτονται τρόπους για να αξιοποιήσουν τις καταπληκτικές δυνατότητες των υπολογιστών στη βιομηχανία. Από τις πρώτες εφαρμογές των υπολογιστών στη βιομηχανία ήταν οι αυτόματες εργαλειομηχανές (τόρνοι, φρέζες κ.λπ.), οι οποίες μέχρι τότε χρησιμοποιούσαν κυρίως μηχανολογικούς και λιγότερο ηλεκτρολογικούς αυτοματισμούς. Η επιτυχημένη αυτή εφαρμογή οδήγησε τους μηχανικούς να αρχίσουν να σκέφτονται την αντικατάσταση όλων των αυτοματισμών ενός εργοστασίου από ένα υπολογιστή.

Σήμερα οι ψηφιακοί υπολογιστές έχουν ολοκληρωτικά επικρατήσει των αναλογικών. Εκατομμύρια απ' αυτούς είναι εγκατεστημένοι στη βιομηχανία όπου ελέγχουν διεργασίες παρακολουθώντας και ελέγχοντας πλήθος μεταβλητών.

Αρχικά τα συστήματα αυτόματου ελέγχου εισήχθησαν στην βιομηχανία λόγω της ανάγκης μείωσης του κόστους παραγωγής. Η ανάγκη περιορισμού των ελαττωματικών προϊόντων που παράγονται σε μια γραμμή παραγωγής καθώς και η ανάγκη παραγωγής προϊόντων υψηλής ποιότητας – εντός αυστηρών προδιαγραφών οδήγησαν στην αυξανόμενη εισαγωγή αυτοματισμών στην βιομηχανία. Φυσική συνέπεια της τάσης αυτής είναι ο άνθρωπος να αποσύρεται σταδιακά από τη χειρονακτική εργασία, που εκτελείται πλέον από «έξυπνες» προγραμματιζόμενες μηχανές.

Μέχρι την δεκαετία του 1960, τα συστήματα υλοποίησης των αλγορίθμων ελέγχου ήταν μηχανικά. Αποτελούνται από ηλεκτρομηχανικούς διακόπτες, τους ηλεκτρονόμους ή αλλιώς

ρελέ (relays) που ανοίγουν ή κλείνουν μία επαφή ανάλογα με την ροή ηλεκτρικού ρεύματος. Η χρήση των ρελέ παρουσίαζε πολλές δυσκολίες όπως ο δύσκολος προγραμματισμός τους (κάθε ρελέ έπρεπε να καλωδιωθεί ξεχωριστά), η περίπλοκη κατασκευή και η περιορισμένη διάρκεια ζωής. Επίσης, καθώς οι αλγόριθμοι γίνονταν όλο και πιο απαιτητικοί σε μνήμη και αριθμητικές πράξεις, τα συστήματα αυτά δεν μπορούσαν να ανταποκριθούν. Αυτά τα προβλήματα οδήγησαν στην εμφάνιση των PLC (Προγραμματιζόμενοι Λογικοί Ελεγκτές) στα τέλη της δεκαετίας του '60. Τα PLCs αναπτύχθηκαν για να αντικαταστήσουν ξεχωριστούς διακόπτες που χρησιμοποιούνταν για διαδοχική (και συνδυαστική) λογική ελέγχου σε διάφορους τομείς της βιομηχανίας. Οι πρώτες plugboard συσκευές εμφανίστηκαν στα μέσα της δεκαετίας του 1960, αλλά το πρώτο PLC ήταν προφανώς το MODICON (Modular Digital Controller – Αρθρωτός Ψηφιακός Ελεγκτής), που αναπτύχθηκε για τη General Motors για την αντικατάσταση ηλεκτρομηχανικών διακοπών στην παραγωγή εξαρτημάτων αυτοκινήτων. Και άλλες παρόμοιες συσκευές προτάθηκαν από διάφορες εταιρίες αλλά το Modicon έφερε το πρώτο PLC στην εμπορική παραγωγή. Τα PLC αποτέλεσαν μια μεγάλη καινοτομία στον τρόπο χειρισμού των διαδικασιών ελέγχου. Πλέον, οι όποιες αλλαγές στο σύστημα ελέγχου γίνονται με λιγότερο κόπο, η διάγνωση των σφαλμάτων είναι γρηγορότερη και ευκολότερη και η συντήρηση διευκολύνθηκε. Τα μοντέρνα PLCs προσφέρουν ευρύ φάσμα επιλογών ελέγχου, συμπεριλαμβάνοντας αλγόριθμους ελέγχου κλειστού βρόχου όπως PID και λογικές συναρτήσεις. Παρά την άνοδο των ανθεκτικών PCs σε πολλές βιομηχανικές εφαρμογές, τα PLCs χρησιμοποιούνται ακόμα ευρέως χάρη στην αξιοπιστία τους και την εξοικείωση που έχουμε με αυτά.

Στις αρχές της δεκαετίας του '70 εμφανίζονται οι πρώτες δυνατότητες επικοινωνίας. Το σύστημα αυτό ονομάστηκε Modbus. Έτσι, τα PLC που βρισκόταν σε μακρινές αποστάσεις είχαν τη δυνατότητα να επικοινωνούν μεταξύ τους όπως επίσης και η μηχανή που ελέγχονταν μπορούσε να είναι απομακρυσμένη. Ακόμη, απέκτησαν τη δυνατότητα αποστολής και λήψης διαφόρων τιμών ηλεκτρικών τάσεων ώστε να μπορούν να συνδέονται με αναλογικές συσκευές. Η έλλειψη όμως τυποποίησης και η συνεχώς μεταβαλλόμενη τεχνολογία κατέστησε την επικοινωνία των PLCs σε μαρτύριο.

Μία προσπάθεια τυποποίησης έγινε στη δεκαετία του '80 με το πρωτόκολλο MAP της General Motors, με το οποίο μπορούσαν να επικοινωνούν PLC διαφορετικών κατασκευαστών. Ένα εξίσου σημαντικό σημείο είναι ότι πλέον ο προγραμματισμός των PLC μπορεί να γίνει με οποιοδήποτε προσωπικό υπολογιστή, εφόσον είναι εφοδιασμένος με το κατάλληλο λογισμικό προγραμματισμού και όχι με τη χρήση ειδικών τερματικών προγραμματισμού.

Η βιομηχανία μέχρι και τη δεκαετία του 80 μπορούμε να πούμε ότι χρησιμοποιούσε ελάχιστα τα ηλεκτρονικά. Το 90% και πλέον των αυτοματισμών καταλάμβαναν οι αυτοματισμοί με ηλεκτρονόμους. Τα ηλεκτρονικά χρησιμοποιούνταν κυρίως για κάποιες "ευφυείς" εργασίες, και οι πλακέτες αυτές τοποθετούνταν μέσα στους πίνακες των ηλεκτρονόμων.

Κι ενώ η τεχνολογία προχωρά, φθάνουμε στη δεκαετία του 90 όπου τεχνολογικά έγινε μεγάλο άλμα (συσκευές μικρότερες, φθηνότερες, με σημαντικά αυξημένες δυνατότητες συγκριτικά με αυτές της προηγούμενης δεκαετίας) αλλά παράλληλα αυξήθηκε δυσανάλογα το κόστος εκπόνησης των προγραμμάτων και της θέσης σε λειτουργία των εγκαταστάσεων. Ακόμη, εκδόθηκε το πρότυπο IEC 1131-3 με σκοπό την ενοποίηση των γλωσσών προγραμματισμού των PLCs διαφορετικών κατασκευαστών. Οι κατασκευαστές ρίχνουν πλέον σημαντικό βάρος στο λογισμικό όπου παρέχονται έτοιμες λύσεις για τομείς του αυτοματισμού με τη βοήθεια βιβλιοθηκών, εκμεταλλεύονται την πρόοδο των ηλεκτρονικών υπολογιστών και χρησιμοποιούν την εξέλιξη στο λειτουργικό τους σύστημα για να μειώσουν τους χρόνους στον προγραμματισμό των. Εμφανίζονται νέες γλώσσες προγραμματισμού για τεχνολόγους σε γραφική μορφή (block διαγράμματα συναρτήσεων), όπου ο χρήστης μέσω βιβλιοθηκών κι έχοντας γνώση μόνο της παραγωγικής διαδικασίας "συνθέτει" τον

αυτοματισμό του. Τα υπόλοιπα γίνονται στο παρασκήνιο για λογαριασμό του. Υποστηρίζεται τέλος και η εξέλιξη στις γλώσσες προγραμματισμού των ηλεκτρονικών υπολογιστών (Pascal, C++) για χρήστες που είναι εξοικειωμένοι σε τέτοια περιβάλλοντα.

Τέλος, ιδιαίτερη έμφαση δίνεται πλέον στη δικτύωση – ασύρματη ή ενσύρματη για τον προγραμματισμό και την επιτήρηση εξ αποστάσεως μέσω ειδικών συσκευών επικοινωνίας και λογισμικού για ηλεκτρονικό υπολογιστή (SCADA) καθώς και στις επικοινωνίες Internet. [1],[11],[17],[20].

1.2 Πλεονεκτήματα της χρήσης αυτοματισμών

Βασικά πλεονεκτήματα της χρήσης αυτοματισμών είναι:

1. Μείωση του λειτουργικού κόστους και αύξηση της αξιοπιστίας του συστήματος, τα οποία οφείλονται στους εξής παράγοντες:

- Απαιτείται ελάχιστο ανθρώπινο δυναμικό, καθώς οι αυτοματισμοί αναλαμβάνουν σημαντικό τμήμα της εργασίας το οποίο φέρουν εις πέρας με μεγαλύτερη αποτελεσματικότητα. Παράλληλα, με αυτό τον τρόπο, μειώνονται τα ανθρώπινα σφάλματα σε λειτουργίες ρουτίνας τα οποία είναι και τα πλέον συνήθη.
- Εντοπίζονται και αντιμετωπίζονται ταχύτερα και αποτελεσματικότερα τα διάφορα σφάλματα αλλά και οι καταστάσεις συναγερμού (alarms). Προλαμβάνονται έτσι καταστάσεις, που θα μπορούσαν να οδηγήσουν σε μη αναστρέψιμες ή εκτεταμένες βλάβες και επιπλέον οι όποιες αστοχίες αντιμετωπίζονται στον ελάχιστο δυνατό χρόνο.
- Είναι δυνατή πλέον, η εγκατάσταση και λειτουργία προηγμένων συστημάτων ελέγχου, συντονισμού και διαχείρισης του δικτύου σε όλα τα επίπεδα εφαρμόζοντας σύγχρονες τεχνικές όπως νευρωνικά δίκτυα, έμπειρα συστήματα (expert systems) ή ασαφή λογική (fuzzy logic).

2. Μείωση του κόστους συντήρησης, το οποίο αποτελεί έναν πάγιο στόχο εκείνων που σχεδιάζουν και υλοποιούν τέτοια συστήματα. Αυτό επιτυγχάνεται με :

- Μείωση των αστοχιών και του χρόνου αποκατάστασής τους. Παλαιότερα, και το πλέον απλό σύστημα αυτοματισμού και διαχείρισης μιας εγκατάστασης, απαιτούσε αρκετά χιλιόμετρα καλωδίωσης αλλά και πολύπλοκο εξοπλισμό, γεγονός που δημιουργούσε πολλές εστίες πιθανών σφαλμάτων και επιπλέον, όταν αυτά συνέβαιναν, ο εντοπισμός τους αλλά και η διόρθωσή τους αποτελούσαν σύνθετη και επίπονη διαδικασία. Η εισαγωγή ταχύτατων δικτύων υπολογιστών έλυσε σε σημαντικό βαθμό όλα αυτά τα προβλήματα, αφού πλέον η βασική απαίτηση σε καλωδίωση είναι ένα απλό καλώδιο του Profibus, Fieldbus ή του Modbus στις συνηθέστερες των περιπτώσεων. Επιπλέον, οι αναρίθμητες διαφορετικές συσκευές αντικαταστάθηκαν από όμοιες κάρτες στις οποίες το μόνο που διαφοροποιείται είναι το λογισμικό και έτσι χρειαζόμαστε ελάχιστα εξαρτήματα για τις ανάγκες της συντήρησης.

- Μείωση του κόστους τακτικής συντήρησης του κύριου εξοπλισμού μίας εγκατάστασης, αφού μπορούμε να έχουμε μια πραγματική εικόνα του τρόπου λειτουργίας αλλά και των καταπονήσεών του. Παραδείγματος χάριν, μέχρι τώρα η συντήρηση ενός μετασχηματιστή γινόταν με βάση στατιστικούς πίνακες. Αντίθετα τώρα, με μικρό κόστος, μπορούμε να έχουμε στην διάθεση μας ένα μεγάλο αριθμό πληροφοριών για την πραγματική του κατάσταση, όπως τα φορτία του, η ισχύς φόρτισής του, η θερμοκρασία του, ένα πλήρες ιστορικό ρευμάτων βραχυκύκλωσης διαμέσου αυτού και άλλες τέτοιες πληροφορίες, οι οποίες μπορούν να μας καθορίσουν με μεγαλύτερη ακρίβεια το πότε πρέπει να γίνουν εργασίες συντήρησης. Είναι σαφές ότι τέτοιου είδους πληροφορίες δε μπορούν να αποκτηθούν με τους συμβατικούς

(αναλογικούς) τρόπους και το κυριότερο να τύχουν επεξεργασίας προκειμένου να βγάλουμε κάποιο συμπέρασμα.

- Μείωση του κόστους συντήρησης και λειτουργίας των συστημάτων ελέγχου και προστασίας. Η εισαγωγή των PLC (προγραμματιζόμενων λογικών ελεγκτών) αλλά και των μικροεπεξεργαστών, έδωσε τη δυνατότητα πραγματοποίησης εξαιρετικά σύνθετων λειτουργιών ελέγχου και προστασίας, οι οποίες με χρήση απλών αναλογικών ηλεκτρονόμων και διακοπών θα ήταν όχι απλώς δύσκολες αλλά και σε πολλές περιπτώσεις αδύνατες. Οι σύγχρονες εγκαταστάσεις δεν είναι μόνο πιο πολύπλοκες, αλλά έχουν και μεγαλύτερες διαστάσεις, οπότε η χρήση σύγχρονων πληροφοριακών συστημάτων είναι μονόδρομος.

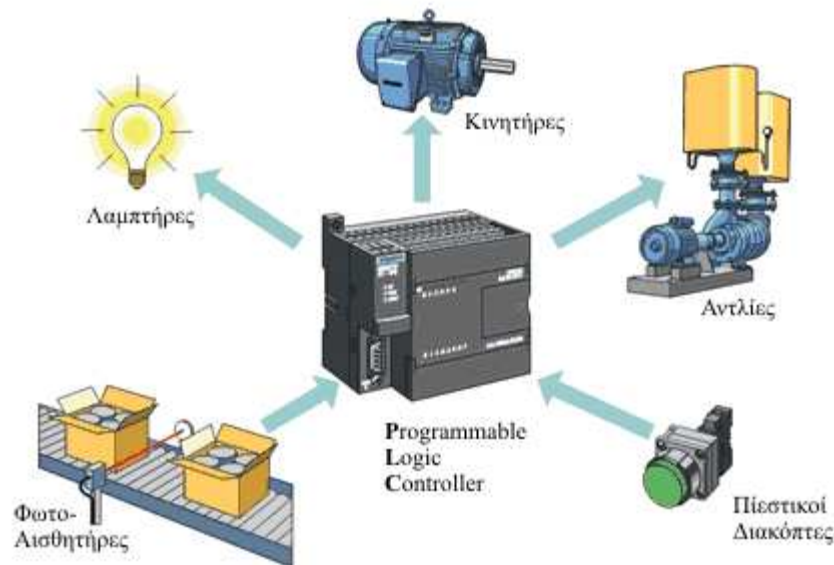
3. Μείωση του κόστους του εγκατεστημένου υλικού, η οποία μπορεί να προέλθει από τα εξής σημεία:

- Μείωση του κόστους καλωδίωσης για τα συστήματα ελέγχου και προστασίας. Κάθε μεγάλη εγκατάσταση έλεγχου και προστασίας, απαιτεί αρκετά χιλιόμετρα καλωδίων που πέρα από το υψηλό κόστος προκαλεί και άλλα προβλήματα και περιορισμούς, όπως αυτεπαγωγές, απώλειες σήματος και ηλεκτρομαγνητικές παρεμβολές.

- Μείωση του κόστους εγκατάστασης εξοπλισμού σχετιζομένου με συγκεκριμένες λειτουργίες. Με τα συμβατικά μέσα, κάθε νέα λειτουργία απαιτούσε την αγορά και εγκατάσταση νέου εξοπλισμού καθώς και καλωδίωσης. Κάτι τέτοιο δεν είναι απαραίτητο με τα νέα συστήματα SCADA, αφού το μόνο που πιθανόν να απαιτείται επιπλέον, είναι καινούργιοι αισθητήρες, οι οποίοι προσαρμόζονται στον υπάρχοντα εξοπλισμό με κατάλληλη επέμβαση στο λογισμικό. Επιπλέον, η εγκατάσταση του νέου εξοπλισμού, στις περισσότερες περιπτώσεις, δεν έχει σημαντικές απαιτήσεις ως προς την κτιριακή υποδομή, αφού τέτοια συστήματα καταλαμβάνουν πολύ μικρό χώρο.

4. Καλύτερη και αποτελεσματικότερη διαχείριση της παραγωγής και κατανάλωσης ενέργειας. Το πρόβλημα της βέλτιστης χρησιμοποίησης της ηλεκτρικής αλλά και των λοιπών μορφών ενέργειας, είναι ένα θέμα που απασχολεί σήμερα όλα τα αναπτυγμένα κράτη, όχι μόνο επειδή αυξάνεται η τιμή της kWh, αλλά και επειδή απαιτούνται όλο και μεγαλύτερα ποσά ενέργειας, ενώ παράλληλα, τα ενεργειακά αποθέματα εξαντλούνται. Προκειμένου οι βιομηχανίες να αντιμετωπίσουν το φλέγον αυτό ζήτημα κινούνται σε δύο βασικούς άξονες, ένα μακροπρόθεσμο και ένα βραχυπρόθεσμο. Μακροπρόθεσμο, οι βιομηχανίες θέλουν να γνωρίζουν με τη μέγιστη δυνατή ακρίβεια, τη ζήτηση φορτίου ανά ώρα, ημέρα, μήνα και έτος, ώστε να ελαχιστοποιούν την κατανάλωσή τους, με κριτήριο την εξοικονόμηση ενέργειας και επιπλέον να σχεδιάζουν τη στρατηγική τους για τις μελλοντικές χρονικές περιόδους. Βραχυπρόθεσμο, απαιτείται η -κατά το δυνατόν- ομαλοποίηση της ροής της ενέργειας, προκειμένου να αποφεύγουμε εξαιρετικά οδυνηρές καταστάσεις, όπως οι υπερβάσεις του φορτίου αλλά και του συντελεστή ισχύος. [9],[20].

Κεφάλαιο 2 - PLC



Εισαγωγή

Οι προγραμματιζόμενοι λογικοί ελεγκτές αναφέρονται στην αγγλική ορολογία με το όνομα PLCs (Programmable Logic Controllers). Τα PLCs είναι οι ηλεκτρονικοί υπολογιστές της βιομηχανίας.

Ο Προγραμματιζόμενος Λογικός Ελεγκτής (PLC) είναι μια ειδική συσκευή, η οποία έρχεται να αντικαταστήσει στον πίνακα του κλασικού αυτοματισμού όλους τους βοηθητικούς ηλεκτρονόμους, τα χρονικά και τους απαριθμητές. Αντί για την κατασκευή ενός πίνακα με πολύπλοκες συνδεσμολογίες μεταξύ των παραπάνω υλικών, που έχουμε στον κλασικό αυτοματισμό, με την χρήση του PLC η λειτουργία του αυτοματισμού "προγραμματίζεται" μέσω μιας ειδικής συσκευής (προγραμματιστή) ή μέσω ενός ηλεκτρονικού υπολογιστή με τη βοήθεια ειδικού λογισμικού.

Έχει εισόδους και εξόδους που συνδέονται με τα στοιχεία μιας εγκατάστασης και βέβαια έναν αλγόριθμο που καθορίζει ότι κάποιος συνδυασμός εισόδων παράγει ένα αποτέλεσμα στις εξόδους. Οι ομοιότητες όμως σταματούν εδώ μιας και το ιδιαίτερο χαρακτηριστικό των PLC είναι ότι οι "κανόνες" που καθορίζουν την συμπεριφορά των εξόδων δεν είναι σταθεροί και "καλωδιωμένοι" όπως σε ένα κλασικό πίνακα αυτοματισμού αλλά μπορούν να μεταβάλλονται με την επέμβαση στο πρόγραμμα του PLC χωρίς καμία επέμβαση στο Hardware του συστήματος. Δηλαδή η λογική της λειτουργίας που ενσωματώνεται στο PLC μέσω του προγραμματισμού του είναι μεταβαλλόμενη. Το σύστημα PLC είναι η κατάλληλη εναλλακτική λύση για επεξεργασίες χαμηλής πολυπλοκότητας που απαιτούν ένα σύστημα ελέγχου με γρήγορους χρόνους αντίδρασης.[5],[12],[19],[20]

Μερικές από τις εφαρμογές που χρησιμοποιούνται τα PLC είναι : ασανσέρ , διωλιστήρια , καράβια , υδροηλεκτρικά φράγματα , συστήματα γεννητριών , ανεμογεννήτριες, βιολογικοί καθαρισμοί , αντλιοστάσια , φανάρια σε διασταυρώσεις δρόμων , κυλιόμενες σκάλες, τούνελ κυκλοφορίας αυτοκινήτων, «έξυπνα» σπίτια, συναγερμοί, γραμμές παραγωγής στην

βιομηχανία, αυτόματες μηχανές συσκευασίας – εμφιάλωσης, γκαραζόπορτες, κυλιόμενες διαφημιστικές πινακίδες.[21]

2.1 Ιστορική εξέλιξη του βιομηχανικού αυτοματισμού

Η Βιομηχανία αναπτύχθηκε παράλληλα με τα διάφορα τεχνολογικά επιτεύγματα, στις αρχές του περασμένου αιώνα, όταν η ανθρώπινη εργασία αντικαταστάθηκε σε μεγάλο βαθμό από τις μηχανές. Οι παραγωγικές διαδικασίες είχαν αρχικά διακεκομμένη μορφή με χειροκίνητες επεμβάσεις μεταξύ των διαφόρων τμημάτων τους. Σύντομα η ανάγκη για συνεχή παραγωγή γινόταν όλο και πιο επιτακτική. Αυτή η τάση ενισχύθηκε ακόμη περισσότερο από το γεγονός ότι η παραγωγή ήταν βασισμένη κυρίως σε συνεχείς ροές πρώτων υλών. Ο τύπος αυτός της βιομηχανικής παραγωγής απαιτούσε την εφαρμογή μεθόδων αυτοματισμού και έτσι συνδέθηκε άμεσα η λειτουργία των βιομηχανικών διαδικασιών συνεχούς λειτουργίας με την πρόοδο της τεχνολογίας του βιομηχανικού αυτοματισμού. Από τα πρώτα στάδια, ο έλεγχος και η εποπτεία της παραγωγής ήταν στην πλήρη αρμοδιότητα των ανθρώπων. Με την ανάπτυξη νέου εξοπλισμού, μερικές λειτουργίες ελέγχου και εποπτείας σταδιακά αυτοματοποιήθηκαν. Παράλληλα παρατηρείται άνθηση στη βιομηχανία παραγωγής οργάνων και συσκευών ελέγχου. Ο αυτοματισμός από τα αρχικά του βήματα περιλαμβάνει τους εξής τρεις βασικούς παράγοντες:

- τα αισθητήρια που συγκεντρώνουν πληροφορίες από το περιβάλλον παραγωγής
- τους ενεργοποιητές που επιτρέπουν την υλοποίηση των αποφάσεων ελέγχου και
- τα συστήματα αποφάσεων που αποφασίζουν, προγραμματίζουν και κατευθύνουν τις ενέργειες ελέγχου.

Τα πρώτα χρόνια της βιομηχανικής εποχής, οι παράγοντες αυτοί ήταν τα μάτια, τα χέρια, και το μυαλό των υπευθύνων χειριστών αντίστοιχα. Στη συνέχεια, στα πρώτα συστήματα ελέγχου, που είχαν αρκετά πρωτόγονη μορφή, οι λειτουργίες ελέγχου είχαν ενσωματωμένες τις λειτουργίες αισθητήρων και ενεργοποιητών. Προς τα μέσα της δεκαετίας του 20 υπάρχει ευρεία χρήση του διακοπτικού ελέγχου ενώ προς το τέλος της ίδιας δεκαετίας άρχισε να εφαρμόζεται η χρήση αντισταθμητών PID. Στη δεκαετία τέλος του 30 αρχίζει η ευρεία χρήση των αντισταθμητών PID.

Οι τεχνολογικές εξελίξεις στη διάρκεια του δεύτερου Παγκοσμίου πολέμου είχαν σημαντική επίδραση στον τομέα του βιομηχανικού ελέγχου.

Στις αρχές της δεκαετίας του 50 είναι πλέον δεδομένη η χρήση ηλεκτρονικών οργάνων στον έλεγχο με παράλληλη καθιέρωση της τυποποίησης στη μετάδοση σημάτων. Το σύστημα ελέγχου έγινε περισσότερο ευέλικτο και αποκεντρωμένο, αποτελούμενο από αισθητήρες, ρυθμιστές, ενεργοποιητές, και καταγραφικά όργανα. Αυτό διευκόλυνε ιδιαίτερα το σχεδιασμό, την εγκατάσταση, τη λειτουργία, την επέκταση και τη συντήρηση του εξοπλισμού ελέγχου. Με τη καθιέρωση της τυποποίησης απλοποιήθηκε η διαδικασία συνδυασμού του εξοπλισμού από διαφορετικούς κατασκευαστές.

Η γενική αντιμετώπιση στο πρόβλημα του αυτοματισμού ήταν η χρήση αισθητήρων και ενεργοποιητών σε τοπικό επίπεδο, ενώ σε κεντρικό θάλαμο ελέγχου υπήρχαν ελεγκτές PID για τη λήψη των αποφάσεων ελέγχου. Τα τυποποιημένα συστήματα ελέγχου χρησιμοποιούσαν έλεγχο ονομαστικών τιμών, δηλαδή διατήρηση της τιμής μιας μεταβλητής όσο γίνεται πιο κοντά σε μια επιλεγμένη, από πριν, τιμή. Λόγω του γεγονότος ότι οι βρόχοι ελέγχου ήταν αυτόνομα σχεδιασμένοι, δεν ελάμβαναν υπόψη τους αλληλεπιδράσεις από άλλες μεταβλητές. Αυτό πολλές φορές δημιουργούσε αρκετά σημαντικά προβλήματα. Προβλήματα επίσης παρουσιάζονταν κατά τη συνεργασία συστημάτων ελέγχου και ηλεκτρονόμων (τηλεχειριζόμενων διακοπών ή κοινώς ρελέ). Η χρήση των υπολογιστών άρχισε από τα μέσα της δεκαετίας του 60, αλλά σε πολύ περιορισμένη κλίμακα λόγω του μεγάλου όγκου τους και της χαμηλής ταχύτητας επεξεργασίας. Η πραγματική τομή στην

τεχνολογία των υπολογιστών ήρθε με την ανάπτυξη των μικροϋπολογιστών στις αρχές της δεκαετίας του 70. Το μικρό μέγεθος, το χαμηλό κόστος και η υψηλή ταχύτητα του νέου τύπου υπολογιστή κατέστησαν την υπόθεση του αυτομάτου ελέγχου προσιτή για οποιοδήποτε είδος εφαρμογής. Με τη χρήση των μικροϋπολογιστών ήταν επίσης δυνατή η αλλαγή τεχνολογίας σε λειτουργίες ελέγχου που πραγματοποιούνται ήδη με κλασικά συστήματα ηλεκτρονόμενων. Έτσι κάνει την εμφάνισή του το PLC. Η τεχνολογία των μικροϋπολογιστών έφερε μια επανάσταση στα ψηφιακά ηλεκτρονικά και το PLC, προϊόν αυτής της επανάστασης, εισήλθε στην βιομηχανία σαν βασικό κύτταρο αυτοματισμού. Το αναλογικό υλικό αντικαθίσταται γρήγορα από μικροϋπολογιστές ακόμη και για τον έλεγχο απλών βρόχων, ενώ εγκαθίστανται "αφοσιωμένα" μικροϋπολογιστικά συστήματα για άμεσο ψηφιακό έλεγχο. Παράλληλα τα συστήματα τηλεχειρισμού που προσφέρονται στους χειριστές στο θάλαμο ελέγχου, έχουν βελτιωθεί και χρησιμοποιούν έγχρωμες οθόνες για την παρουσίαση των πληροφοριών. Παράλληλα αρχίζει η ανάπτυξη ιεραρχικών συστημάτων ελέγχου που περιέχουν μεγάλο αριθμό μικροϋπολογιστών. Τα PLCs έχουν γίνει ευρέως αποδεκτά και κυριαρχούν στη βιομηχανία σαν συσκευές ελέγχου. Οι βιομηχανικές εφαρμογές έδειξαν ότι η συνύπαρξη υπολογιστών και PLCs στο ίδιο σύστημα έχουν μεγάλα πλεονεκτήματα. Η σημερινή μορφή αυτοματισμού των περισσότερων εκσυγχρονισμένων βιομηχανιών έχει σαν βάση τα συστήματα ιεραρχικού ελέγχου που υλοποιούνται με συνδυασμό υπολογιστών και PLCs.

Τη δεκαετία του 70 είχαμε ζήτηση για συστήματα ελέγχου που διεκπεραιώνουν διοικητικές εργασίες, π.χ. εκθέσεις παραγωγής και αποθήκης, υπολογισμοί συνθέσεων μιγμάτων, κλπ.. Την εποχή εκείνη η ανάπτυξη του ελέγχου επεξεργασιών με H/Y ήταν αρκετά προχωρημένη, ειδικά οι γλώσσες προγραμματισμού είχαν αναπτυχθεί σε τέτοιο βαθμό που να καθιστούν τα συστήματα αυτά προσιτά σε χειριστές χωρίς ειδικές γνώσεις H/Y. Για να προσαρμοστούν τα συστήματα PLC στις νέες απαιτήσεις, συμπεριλήφθηκαν στην κεντρική μονάδα τους μικροϋπολογιστές, έτσι ο σύγχρονος εξοπλισμός PLC είναι στην πραγματικότητα H/Y με τεχνολογία PLC.

Για τη καλή λειτουργία των ιεραρχικών συστημάτων μεγάλη συμβολή έχουν και τα τοπικά δίκτυα επικοινωνίας τα οποία κάνουν εφικτή την πλήρη αξιοποίησή τους. Τα οικονομικά οφέλη που προκύπτουν είναι τεράστια, όπως έχει αποδειχθεί στην πράξη, από πλευράς αύξησης της ποιότητας, μείωσης του κόστους παραγωγής και ευελιξίας καθώς και αξιοπιστίας της βιομηχανικής παραγωγής. [3],[4]

2.2 Ιστορική εξέλιξη των PLC

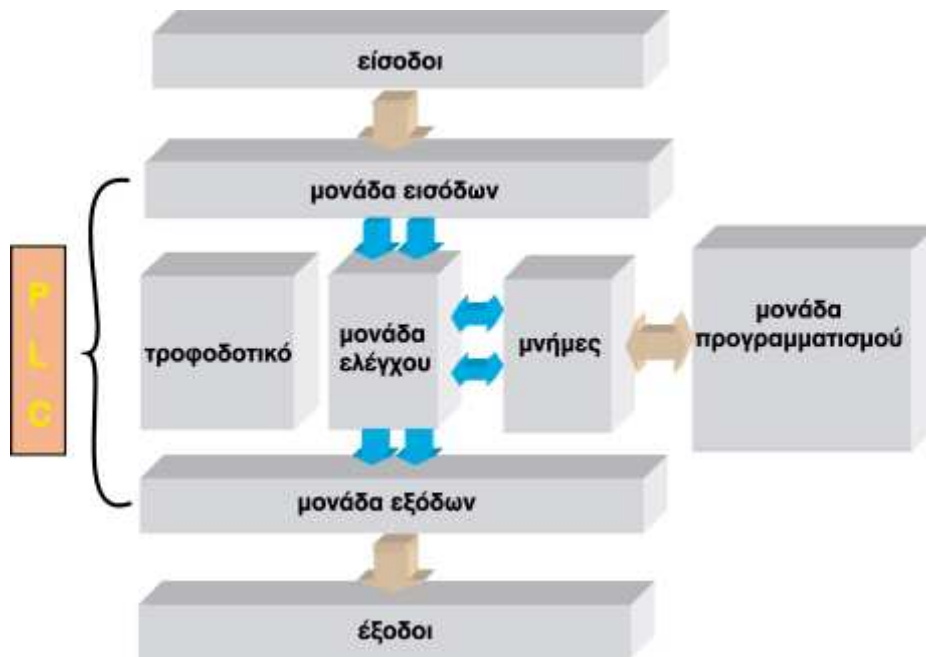
Ο Προγραμματιζόμενος Λογικός Ελεγκτής (PLC) αναπτύχθηκε στις Η.Π.Α. στο τέλος της δεκαετίας του 60. Στόχος ήταν ν' αναπτυχθεί ένα ευέλικτο σύστημα στο οποίο να υπάρχει η δυνατότητα εύκολης αλλαγής του προγράμματος.

Αντί για σύνδεση και συγκόλληση καλωδίων, οι χειρισμοί θα προγραμματιζόταν μέσα στο σύστημα, όπως ακριβώς γίνεται με τον H/Y. Η αρκετά πολύπλοκη γλώσσα προγραμματισμού των ημερών εκείνων, όμως, θα αντικαθιστούταν από μια απλούστερη γλώσσα, έτσι ώστε να είναι δυνατό το γράψιμο προγραμμάτων από χειριστές που δεν είναι ειδικοί στους H/Y.

Το σύστημα PLC αποτέλεσε ένα σπουδαίο βήμα προόδου στον αυτοματισμό, και εξοπλισμός αυτού του τύπου εγκαταστάθηκε σε ένα μεγάλο αριθμό εργοστασίων σ' όλο τον κόσμο.[8]

2.3 Βασικά χαρακτηριστικά ενός PLC

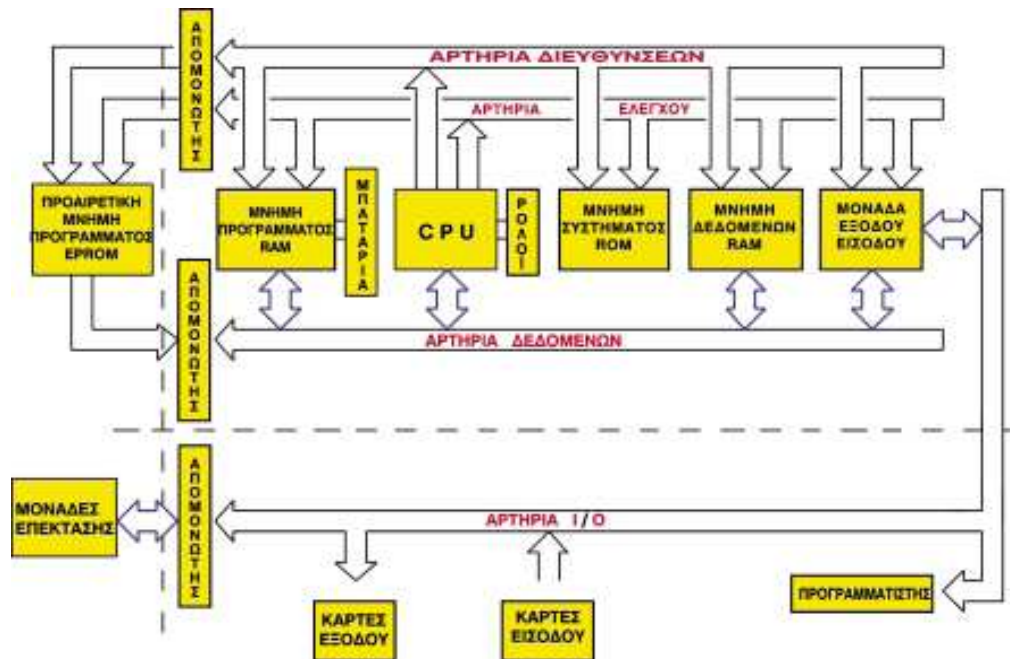
Από άποψη σχεδιασμού, ένα σύστημα PLC μοιάζει με Η/Υ στο ότι αποτελείται από μια κεντρική μονάδα επεξεργασίας (CPU) ή επεξεργαστή (processor), μια μονάδα μνήμης, μονάδα εισόδου για παραλαβή σημάτων από το μηχανολογικό εξοπλισμό της γραμμής επεξεργασίας, από μονάδα εξόδου για αποστολή σημάτων εξόδου από την κεντρική μονάδα προς τον εξοπλισμό της γραμμής επεξεργασίας και από μια μονάδα προγραμματισμού. Το πρόγραμμα αποτελείται από έναν αριθμό οδηγιών και αποθηκεύεται στη μνήμη.[8]



Σχ.2.1 Δομή του PLC[14]

Μέγεθος των PLC:

- Μικρά: μονάδες μέχρι 128 I/O και μνήμες μέχρι 2Kbytes
- Μεσαία: μονάδες μέχρι 2048 I/O και μνήμη μέχρι 32Kbytes+ ειδικές I/O μονάδες
- Μεγάλοι: μονάδες μέχρι 16000 I/O και μνήμη μέχρι 2Mbytes.[13]



Σχ.2.2 Αρχιτεκτονική ενός Προγραμματιζόμενου Λογικού Ελεγκτή [14]

2.3.1 Κεντρική μονάδα (CPU)

Η κεντρική μονάδα επεξεργασίας (Central Processing Unit)-CPU, έχει ακριβώς την ίδια δομή με τη CPU ενός ψηφιακού Ηλεκτρονικού Υπολογιστή. Τα ηλεκτρονικά στοιχεία που την αποτελούν επιλέγονται πάρα πολύ προσεκτικά, ώστε να πληρούν τις αυστηρότερες προδιαγραφές αξιοπιστίας, δηλαδή, θα πρέπει να λειτουργούν σε ένα αρκετά ευρύ θερμοκρασιακό φάσμα και βέβαια η μηχανική τους στήριξη, θα πρέπει να είναι τέτοια ώστε να μην διαταράσσεται η λειτουργία τους από συνήθεις κραδασμούς.

Ο μικροεπεξεργαστής της κεντρικής μονάδας (CPU), αφού δεχθεί τα υπό ψηφιακά σήματα εισόδου, τα επεξεργάζεται και πραγματοποιεί λογικές αποφάσεις σύμφωνα με τις εντολές του προγράμματος που βρίσκεται αποθηκευμένο στην μνήμη του. Η επεξεργασία του προγράμματος γίνεται συνεχώς κυκλικά. Δηλαδή, ο μικροεπεξεργαστής ελέγχει συνεχώς τις διάφορες εισόδους, αν έχουν τάση ή όχι (επαφές κλειστές ή ανοικτές), επεξεργάζεται τις εντολές του προγράμματος και βάσει των λογικών αποφάσεων που λαμβάνει, εξαναγκάζει τις εξόδους να διεγερθούν ή όχι, ενεργοποιώντας ή απενεργοποιώντας έτσι τα διάφορα εξωτερικά στοιχεία (ρελέ, βαλβίδες κ.λ.π.) που βρίσκονται συνδεδεμένα σε αυτές. Συνήθως ένα PLC έχει μία μόνο CPU, η οποία όμως μπορεί να εξυπηρετεί ταυτόχρονα πολλές εισόδους και εξόδους.

Η κεντρική μονάδα επεξεργασίας πραγματοποιεί πολλαπλές βασικές λειτουργίες:

- Διάβασμα, ερμηνεία και εκτέλεση, με τη σωστή διαδοχή, των οδηγιών, που περιέχονται στην μνήμη.
- Έλεγχο του πρωτοκόλλου επικοινωνίας που έχουμε καθορίσει στο σύστημα μας.
- Αποθήκευση των πληροφοριών
- Εκτέλεση αριθμητικών πράξεων

Τα δεδομένα εισόδου που επεξεργάζεται μια CPU ενός PLC είναι δυαδικής μορφής και για το λόγο αυτό λέμε συνήθως ότι τα PLCs είναι 1 bit Boolean Processors (επεξεργαστές του 1 bit). Υπάρχουν βέβαια και επεξεργαστές με δυνατότητα επεξεργασίας πολλαπλών bits.

Η κεντρική μονάδα διαβάσει τη μνήμη βήμα προς βήμα με σταθερή ταχύτητα. Ανάλογα με τον αριθμό των καταχωρημένων (στη μνήμη) οδηγιών, χρειάζονται 10-20 χιλιοστά του δευτερολέπτου (ms) για να διαβασθεί ολόκληρη η μνήμη.

Έτσι, όταν ο κύκλος ανάγνωσης του προγράμματος (κύκλος προγράμματος) είναι 20 ms, μια οδηγία σ' ένα συγκεκριμένο σημείο του προγράμματος επεξεργάζεται (διαβάζεται και προωθείται ανάλογα) 50 φορές το δευτερόλεπτο. Υπάρχει επομένως ένας μέγιστος ενδιάμεσος χρόνος 0.02 sec (δευτερολέπτων) που μεσολαβεί πριν ένα γεγονός, π.χ. ένα σήμα από κάποιο δείκτη θερμοκρασίας ή ορίου, εντοπισθεί από την κεντρική μονάδα.[8],[9],[10]

2.3.2 Μνήμη

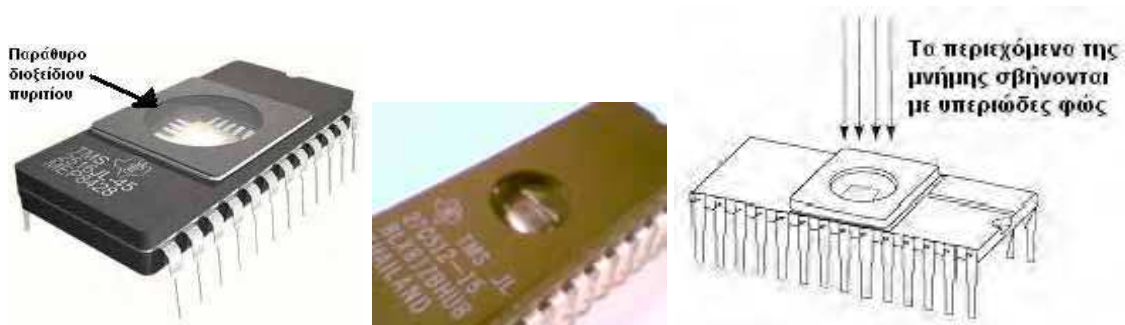
2.3.2.1 Μη πτητικές μνήμες

Οι μη πτητικές μνήμες έχουν την ικανότητα να διατηρήσουν τις αποθηκευμένες πληροφορίες τους σε περίπτωση που η τροφοδοσία τους διακοπεί τυχαία ή σκόπιμα. Αν και οι μη πτητικές μνήμες είναι αμετάβλητες, υπάρχουν κάποιοι ειδικοί τύποι που προτιμώνται από τους κατασκευαστές και στους οποίους οι αποθηκευμένες πληροφορίες μπορούν να αλλάξουν.

- Μνήμη ROM (Read Only Memory). Η μνήμη ROM είναι σχεδιασμένη ώστε οι πληροφορίες που περιέχει να είναι μόνο για ανάγνωση. Τα δεδομένα της μνήμης αυτής είναι τοποθετημένα από τον κατασκευαστή για εσωτερική χρήση και λειτουργία του PLC. Οι μόνοι για ανάγνωση μνήμες είναι αμετάβλητες και χρησιμοποιούνται από τον ελεγκτή ως λειτουργικό σύστημα. Το λειτουργικό σύστημα εισάγεται στη μνήμη ROM από τον κατασκευαστή του ελεγκτή και ελέγχει το λογισμικό που χρησιμοποιεί ο χρήστης για τον προγραμματισμό.

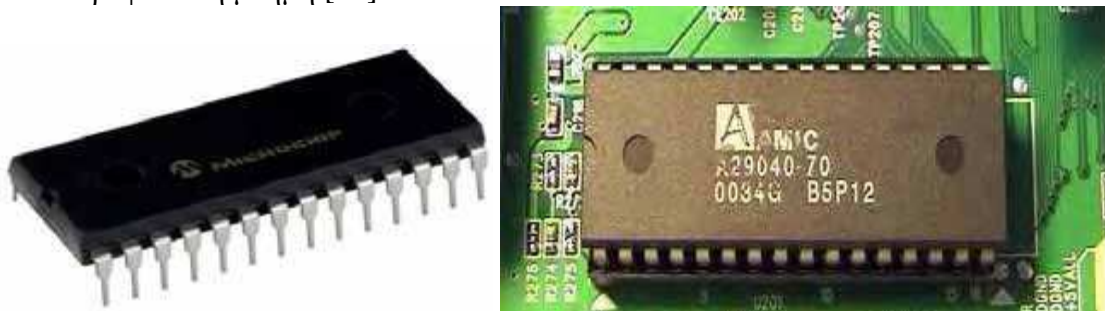
- Μνήμη PROM (Programmable Read Only Memory). Πρόκειται για έναν ειδικό τύπο μνήμης ROM η οποία είναι επίσης αμετάβλητη και είναι μόνο για ανάγνωση. Η μνήμη PROM μπορεί να δεχτεί γραφή μόνο μία φορά και αυτή από τον κατασκευαστή. Ο προγραμματισμός επιτυγχάνεται από παλμούς ρεύματος που λιώνουν τις εύτηκτες συνδέσεις στο ολοκληρωμένο, εμποδίζοντας τον αναπρογραμματισμό του. Ελάχιστοι ελεγκτές χρησιμοποιούν τη μνήμη PROM για τη μνήμη προγράμματος, γιατί οποιαδήποτε αλλαγή στο πρόγραμμα θα απαιτούσε ένα νέο σύνολο από PROM ολοκληρωμένα.[22]

- Μνήμη EPROM (Erasable Programmable Read Only Memory). Η προγραμματιζόμενη μνήμη μόνο για ανάγνωση που μπορεί να σβηστεί είναι μια ειδικά σχεδιασμένη μνήμη PROM που μπορεί να προγραμματιστεί ξανά αφού διαγραφεί εντελώς με τη χρήση μιας πηγής υπεριώδους φωτός. Αυτό το ολοκληρωμένο που επίσης ονομάζεται υπεριώδης PROM μνήμη (UV PROM) έχει ένα παράθυρο διοξειδίου του πυριτίου (quartz) πάνω από το τσιπ πυριτίου. Αυτό το παράθυρο συνήθως είναι καλυμμένο από ένα αδιαφανές υλικό. Όταν το αδιαφανές υλικό απομακρυνθεί και το κύκλωμα εκτεθεί στο υπεριώδες φως για 20 περίπου λεπτά, το περιεχόμενο της μνήμης μπορεί να διαγραφεί. Αφού διαγραφεί, το ολοκληρωμένο EPROM μπορεί να προγραμματιστεί ξανά χρησιμοποιώντας τη συσκευή προγραμματισμού. Η μνήμη EPROM ή UV PROM χρησιμοποιείται για να στηρίζει, να αποθηκεύσει ή να μεταφέρει PLC προγράμματα. Ο επεξεργαστής PLC μπορεί να διαβάσει μόνο από αυτόν τον τύπο συσκευής μνήμης. Ένας εξωτερικός προγραμματιστής μνήμης PROM χρησιμοποιείται για να προγραμματίσει (να γράψει) τη συσκευή. Η UV PROM είναι μη πτητική συσκευή μνήμης και δεν απαιτεί υποστήριξη μπαταρίας.[22]



Σχ.2.3 EPROM[22]

• Μνήμη EEPROM (Electrically Erasable Programmable Read Only Memory). Ηλεκτρικά Διαγραφόμενη Μνήμη Μόνο Ανάγνωσης. Η μνήμη EEPROM είναι μια μνήμη όπου μπορεί να προσφέρει την ίδια ελαστικότητα με μία μνήμη RAM, μπορεί να σβηστεί και να γραφούν πάνω της νέα δεδομένα και πληροφορίες. Η διαγραφή τέτοιου είδους μνημών γίνεται μόνο ηλεκτρικά και όχι με τη χρήση υπεριώδους φωτός. Έχει την δυνατότητα μόνιμης αποθήκευσης του προγράμματος, ένα πρόγραμμα το οποίο μπορεί εύκολα να αλλάξει. Οι μνήμες EEPROM χρησιμοποιούνται για αποθήκευση, εκτέλεση και μεταφορά προγραμμάτων. Ο επεξεργαστής PLC μπορεί να διαβάσει και να γράψει σε μια EEPROM. Μην εκθέτετε τον επεξεργαστή σε επιφάνειες ή άλλες περιοχές που μπορεί να έχουν ηλεκτροστατική φόρτιση. Οι ηλεκτροστατικές φορτίσεις μπορεί να αλλάξουν ή να καταστρέψουν τη μνήμη.[22]



Σχ.2.4 EEPROM[22]

• Προαιρετική μνήμη :FEPROM (Flash Erasable PROM: PROM που μπορεί να σβηστεί ηλεκτρονικά). Σε αυτή μπορεί να αποθηκευτεί το πρόγραμμα αφού πάρει την τελική του μορφή απελευθερώνοντας έτσι τη μνήμη RAM.

2.3.2.2 Πτητικές μνήμες

Οι πτητικές μνήμες είναι οι μνήμες που δεν έχουν την δυνατότητα να διατηρήσουν τα δεδομένα τους σε περίπτωση διακοπής της τροφοδοσίας τους. Σε αυτήν την κατηγορία ανήκει η μνήμη RAM. Οι μνήμες τυχαίας προσπέλασης (RAM) είναι σχεδιασμένες έτσι ώστε οι πληροφορίες να μπορούν να γράφονται σε αυτήν και να διαβάζονται από αυτή. Το πρόγραμμα του χρήστη, οι τιμές των μετρητών και των χρονιστών, οι καταστάσεις των εισόδων και εξόδων αποθηκεύονται σε αυτήν την μνήμη. Στη μνήμη RAM η κεντρική μονάδα αποθηκεύει μια σειρά από πληροφορίες σε ξεχωριστές περιοχές εργασίας. Μπορούμε να διακρίνουμε τις εξής περιοχές:

α. Περιοχή μνήμης όπου αποθηκεύονται οι καταστάσεις των εισόδων και των εξόδων. Η περιοχή αυτή ονομάζεται για τις εισόδους «εικόνα εισόδου» και για τις εξόδους «εικόνα εξόδου».

β. Περιοχή μνήμης όπου αποθηκεύονται οι ενδιαμέσες πληροφορίες που αφορούν τη λειτουργία του αυτοματισμού.

γ. Περιοχή μνήμης των χρονικών.

δ. Περιοχή μνήμης των απαριθμητών.

ε. Περιοχή μνήμης όπου αποθηκεύονται τα προγράμματα του χρήστη, δηλαδή τα προγράμματα που λειτουργούν ένα συγκεκριμένο αυτοματισμό.

Κατά τη λειτουργία του προγραμματιζόμενου ελεγκτή, το περιεχόμενο της μνήμης RAM μπορεί να μεταβληθεί αρκετές φορές. Η μνήμη RAM δεν έχει την ικανότητα διατήρησης των δεδομένων της σε περίπτωση διακοπής της τροφοδοσίας της. Για αυτόν τον λόγο, προστατεύεται από μια μπαταρία. Οι μνήμες CMOS-RAM έχουν χαμηλή κατανάλωση ρεύματος και μπορούν να διατηρηθούν για μεγάλο χρονικό διάστημα υπό την τροφοδοσία μιας μπαταρίας. Αυτές οι μπαταρίες ποικίλουν όσο αφορά τον χρόνο ζωής τους. Μπαταρίες με μικρό χρόνο ζωής είναι οι αλκαλικές και οι μπαταρίες υδραργύρου οι οποίες αντικαθίστανται περιοδικά (από 6 μήνες έως ένα έτος). Αντίθετα, υπάρχουν και μπαταρίες με μεγάλο χρόνο ζωής, όπως οι μπαταρίες λιθίου οι οποίες αντικαθίστανται κάθε 10 χρόνια καθώς και επαναφορτιζόμενες μπαταρίες, όπως οι μπαταρίες Νικελίου-Καδμίου και Μολύβδου. [9],[14],[17],[18],[22]

2.3.3 Μονάδες εισόδου – εξόδου

Οι μονάδες των εισόδων και των εξόδων αποτελούν τις μονάδες επικοινωνίας της κεντρικής μονάδας με τους αισθητήρες, τους διακόπτες, τα μπουτόν κ.α., που δίνουν τις πληροφορίες (εντολές) στη κεντρική μονάδα, καθώς και με τα ρελέ ισχύος των κινητήρων, ηλεκτρομαγνητικές βαλβίδες, ενδεικτικές λυχνίες και γενικά τους αποδέκτες που εκτελούν τις εντολές της κεντρικής μονάδας.

Οι αγωγοί σύνδεσης (καλώδια) από τα αισθητήρια (sensors) της παραγωγικής διαδικασίας συνδέονται στις κλέμμες (ακροδέκτες) των μονάδων εισόδου (INPUT MODULES). Αντίστοιχα, τα καλώδια που πηγαίνουν προς τους ενεργοποιητές (actuators) συνδέονται στις κλέμμες των μονάδων εξόδου (OUTPUT MODULES).

Ο προγραμματιζόμενος λογικός ελεγκτής αντιλαμβάνεται ότι ένα αισθητήριο είναι ανοικτό ή κλειστό από το αν εμφανίζεται ή όχι τάση στην αντίστοιχη κλέμα εισόδου. Επίσης, αν από το πρόγραμμα δοθεί εντολή για διέγερση π.χ. μιας βαλβίδας, τότε κλείνει ο διακόπτης ή εμφανίζεται τάση στην αντίστοιχη κλέμα εξόδου. Η τάση αυτή μπορεί να παρέχεται είτε από τη μονάδα τροφοδοσίας του PLC είτε από κατάλληλο τροφοδοτικό DC ή μετασχηματιστή τάσης χειρισμού (AC).

Η κεντρική μονάδα μπορεί να δεχτεί ψηφιακά σήματα εισόδου και εξόδου χαμηλής τάσης και πολύ μικρού ρεύματος. Η τάση που δέχεται είναι συνήθως 0 Volt για το λογικό “0” και 5 Volt για το λογικό “1”. Το ρεύμα εισόδου καθώς και το ρεύμα εξόδου δεν μπορεί να ξεπεράσει τα λίγα mA. Οι μονάδες εισόδων και εξόδων αναλαμβάνουν να προσαρμόσουν τα σήματα εισόδου και εξόδου, που έχουμε στον αυτοματισμό, σε σήματα που μπορεί να δεχτεί η κεντρική μονάδα, τόσο από άποψη τάσεων όσο και από άποψη ρευμάτων. Η προσαρμογή αυτή γίνεται με χρήση ηλεκτρονικών στοιχείων ισχύος, είτε με τη χρήση των κατάλληλων μικρό-ρελέ.

Κάθε σύστημα PLC καταλήγει πάντα σε ακροδέκτες (κλέμες). Οι ακροδέκτες αυτοί ανήκουν στις μονάδες εισόδων και εξόδων του PLC. Στους ακροδέκτες εισόδων καταλήγουν οι αγωγοί που έρχονται από αισθητήρες (τερματικούς διακόπτες, πιεζοστάτες, κ.λπ.), διακόπτες, μπουτόν κ.λπ. Στους ακροδέκτες εξόδων καταλήγουν οι αγωγοί που τροφοδοτούν

πηνία ηλεκτρονόμων ισχύος, ηλεκτρομαγνητικές βαλβίδες, λυχνίες ένδειξης και λοιπούς αποδέκτες.

Στους διάφορους τύπους των PLC οι μονάδες εισόδων και εξόδων αντιμετωπίζονται με διαφορετικό τρόπο. Γενικά ισχύουν τα παρακάτω:

Μία μονάδα εισόδων ή εξόδων μπορεί να λειτουργεί με συνεχή τάση ή με εναλλασσόμενη τάση. Τυπικές τάσεις που συναντούμε στα PLC είναι: DC 24V, 48V, 60V και AC 24V, 48V, 115V, 230V, με συνηθέστερες τις DC 24V και AC 115V και 230V.

Η τάση αυτή δεν παρέχεται συνήθως από τη μονάδα τροφοδοσίας του PLC. Πρέπει να τη δημιουργήσουμε εμείς με άλλη τροφοδοτική μονάδα.

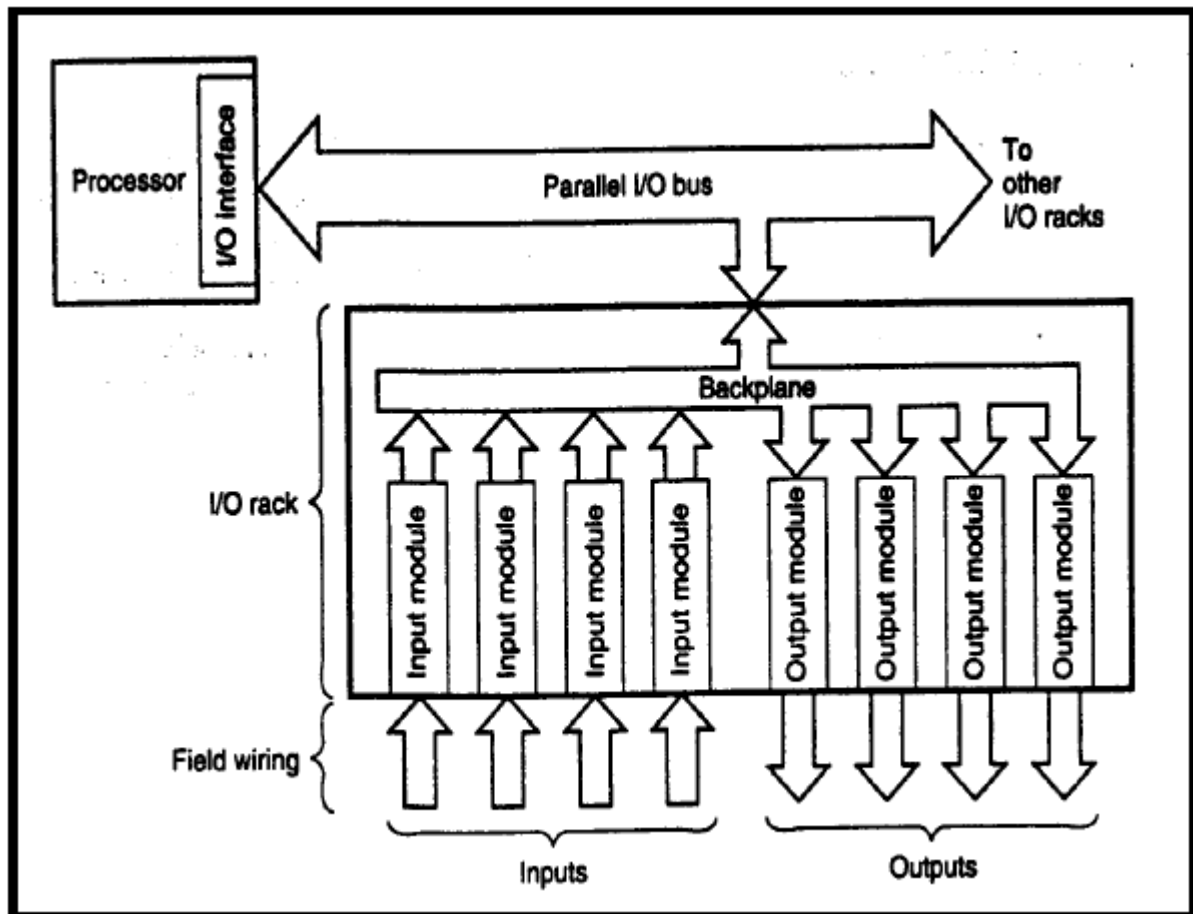
Τα κυκλώματα και οι τάσεις των εισόδων είναι τελείως ανεξάρτητα από τα κυκλώματα και τις τάσεις των εξόδων. Επομένως η τάση για τις εισόδους μπορεί να είναι διαφορετική από την τάση για τις εξόδους. Αν η τάση εξόδων είναι η ίδια με την τάση των εισόδων μπορεί να χρησιμοποιηθεί το ίδιο τροφοδοτικό (για τάσεις DC), ή μετασχηματιστής χειρισμού (για τάσεις AC) για τις εισόδους και τις εξόδους.

Η τάση εισόδων (δηλαδή η τάση που θα φθάσει σε μια είσοδο, όταν ενεργοποιηθεί ο αντίστοιχος αισθητήρας) διαχωρίζεται συνήθως γαλβανικά από το υπόλοιπο εσωτερικό κύκλωμα του PLC. Τα ίδια ισχύουν και για τις εξόδους. Αν σε κάποιες μονάδες εισόδων ή εξόδων δεν έχουμε γαλβανική απομόνωση πρέπει να προσέξουμε ιδιαίτερα το θέμα των γειώσεων.

Τα συστήματα εισόδων-εξόδων των PLC διακρίνονται στις παρακάτω κατηγορίες:

- Άμεσο Σύστημα I/O (Direct I/O Systems). Αυτός ο τύπος συστημάτων χρησιμοποιείται σε μικρά PLC, τα οποία διαθέτουν συγκεκριμένο πλήθος εισόδων/εξόδων στο ίδιο πακέτο με τον επεξεργαστή και αναφέρονται συχνά ως εσωτερικές είσοδοι-έξοδοι I/O (Internal I/O). Το σημαντικότερο πλεονέκτημα των άμεσων συστημάτων είναι η οικονομικότητά τους.

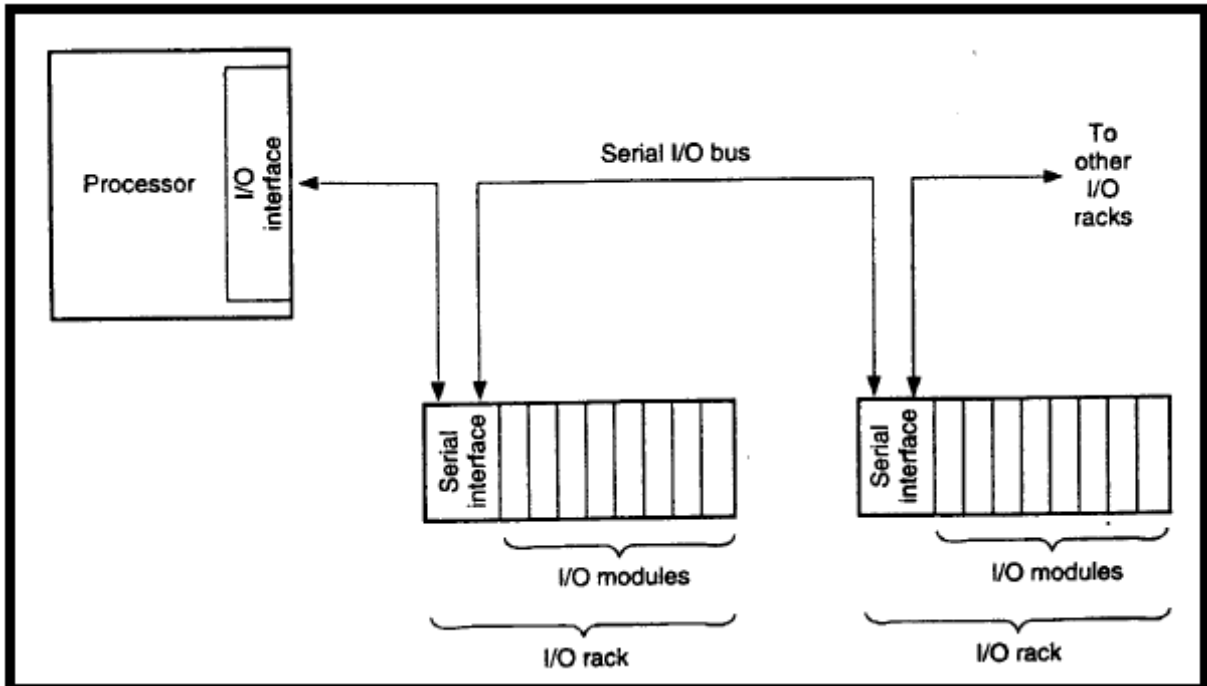
- Παράλληλα Συστήματα (Parallel I/O System): Σε ένα παράλληλο σύστημα, ένας διάυλος ανεξαρτήτων μονάδων I/O είναι συνδεδεμένος στο τμήμα εισόδων – εξόδων του επεξεργαστή. Οι μονάδες αυτές περιλαμβάνουν κατάλληλα κυκλώματα για την αποκωδικοποίηση των σημάτων του διαύλου και τη μετατροπή τους σε επίπεδα τάσης τα οποία μπορούν να οδηγήσουν τα φορτία. Κάθε μια από αυτές τις μονάδες έχει έναν ορισμένο αριθμό σημείων εισόδων ή εξόδων, που λέγεται συναρμολογησιμότητα (modularity). Τα περισσότερα συστήματα εισόδων-εξόδων παρουσιάζουν συναρμολογησιμότητα των 4, 8, 16 και 32 σημείων.



Σχ.2.5 Παράλληλο σύστημα I/O [9]

- Σειριακό Σύστημα (Serial I/O System): Το σειριακό σύστημα παρουσιάζει το πλεονέκτημα της μετάδοσης των δεδομένων εισόδου-εξόδου, σε μεγάλες αποστάσεις, δηλαδή από 300m έως 3000m, σε αντίθεση με το παράλληλο σύστημα, το οποίο μπορεί να επεκτείνει το δίαυλο εισόδου-εξόδου σε απόσταση μικρότερη από 15m. Αυτό σημαίνει ότι αν η απαιτούμενη απόσταση είναι, για παράδειγμα, 30m, τότε θα χρειαζόμασταν 2 PLC με παράλληλο σύστημα εισόδου – εξόδου. Ένας σειριακός δίαυλος μπορεί να συνδεθεί με έναν παράλληλο μέσω κατάλληλου μετατροπέα.

Τα PLC τελευταίας τεχνολογίας, περιλαμβάνουν αμιγώς σειριακά συστήματα εισόδου-εξόδου, τα οποία έχουν μικρότερο κόστος. Ιδιαίτερη προσοχή θα πρέπει να δοθεί σε εφαρμογές κατά τις οποίες προσπελούνται και οι δύο δίαυλοι εισόδου-εξόδου, αντί μόνο ο ένας. Αυτό μπορεί να έχει ως αποτέλεσμα το σειριακό σύστημα να είναι πιο αργό από ένα παράλληλο. Ένα ακόμα μειονέκτημα των σειριακών συστημάτων είναι ότι μπορεί να αποσυγχρονιστούν από τη λογική σάρωση (logic scanning) των εισόδων-εξόδων, με αποτέλεσμα να μη μπορούν να προβλεφθούν οι αποκρίσεις εισόδων και εξόδων σε σήματα ταχέως μεταβαλλόμενα.[9],[20]



Σχ.2.6 Σειριακό σύστημα I/O.[9]

2.3.3.1 Μονάδα εισόδων.

Σκοπός αυτής της μονάδας είναι να λαμβάνει σήματα από τις διάφορες συσκευές εισόδου και να τα προσαρμόζει σε σήματα που αναγνωρίζει ο επεξεργαστής. Κάθε ακροδέκτης εισόδου έχει ένα μοναδικό αριθμό ή όνομα, την ταυτότητα του, η οποία κάνει κάθε είσοδο μοναδική ώστε να μην συγχέεται με άλλο στοιχείο. Η ταυτότητα αναγνώρισης ονομάζεται διεύθυνση και καθορίζεται από τον κατασκευαστή. Έχει επικρατήσει οι εισοδοί να συμβολίζονται με το γράμμα I (Input). Τα νούμερα τα οποία πάντα ακολουθούν αμέσως μετά δείχνουν τον αριθμό της μονάδας βάσης (base unit) ή της μονάδας προέκτασης (extension unit) και τον αριθμό της εισόδου αντίστοιχα.

Σε έναν ακροδέκτη εισόδου μπορούν να συνδεθούν μπουτόν, αισθητήρες και γενικά διακόπτες που αλλάζουν κατάσταση λόγω εξωτερικών γεγονότων ή φυσικών μεγεθών.[18]

Υπάρχουν δύο βασικοί τύποι μονάδων εισόδων:

Ψηφιακές:(ON-OFF), στις οποίες η είσοδος μπορεί να αναγνωρίζει δύο μόνο τιμές τάσης (υψηλή - χαμηλή). Η τάση αυτή μπορεί να δημιουργείται είτε από το τροφοδοτικό του PLC, είτε από δικό μας εξωτερικό τροφοδοτικό. Η τιμή της στα περισσότερα PLC είναι $24 V_{DC}$.

Αναλογικές: στις οποίες το σήμα εισόδου μπορεί να είναι ένα αναλογικό σήμα. Συνήθως τα σήματα αυτά είναι τάσεις 0 έως 10V ή -10V έως 10V, ή εντάσεις ρεύματος 0 έως 20mA ή 4 έως 20mA.

Μια μονάδα εισόδου μπορεί να περιλαμβάνει 4, 8, 16 ή 32 ψηφιακές εισόδους, ανάλογα με τον τύπο του PLC, ο οποίος μπορεί να περιλαμβάνει πολλές τέτοιες μονάδες. Ο μέγιστος αριθμός των αναλογικών εισόδων που μπορεί να διαθέτει ο ελεγκτής δίνεται από τον κατασκευαστή και διαφέρει από εταιρεία σε εταιρεία, αλλά ακόμη και σε μοντέλα της ίδιας εταιρείας.[14]

2.3.3.2 Μονάδα εξόδων.

Σκοπός αυτής τη μονάδας είναι να μεταφέρει τα σήματα εξόδου από τον επεξεργαστή στους ακροδέκτες εξόδου όπου συνδέονται τα διάφορα υπό έλεγχο φορτία. Κάθε ακροδέκτης

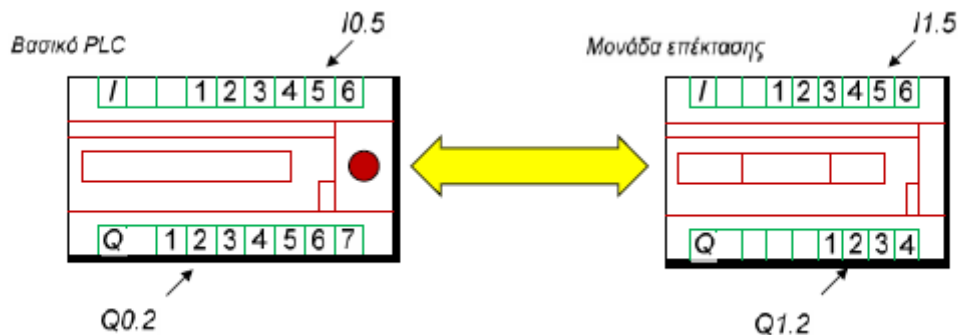
εξόδου έχει ένα μοναδικό όνομα, την διεύθυνση του, το οποίο καθορίζεται από τον κατασκευαστή και δεν μπορεί να αλλαχτεί. Η διεύθυνση είναι το μέσο που χρησιμοποιεί ο ελεγκτής για να στείλει κάποιο σήμα σε μια από τις συσκευές εξόδου.

Οι έξοδοι είναι το αποτέλεσμα των λογικών καταστάσεων των εισόδων σε συνδυασμό με τις εντολές του προγράμματος. Στους ακροδέκτες εξόδου συνδέονται τα ηλεκτρικά κυκλώματα που επιθυμούμε να ελέγξουμε.

Έχει επικρατήσει ο συμβολισμός των εξόδων να γίνεται με το γράμμα Q. Το νούμερο αμέσως μετά που ακολουθεί δείχνει τον αριθμό της βάσης ή προέκτασης και το τελευταίο νούμερο, τον αριθμό της εξόδου.[18]

Και οι μονάδες εξόδου διακρίνονται σε ψηφιακές και αναλογικές. Τυπικές τιμές τάσης ψηφιακών εξόδων είναι 24 V_{DC}, 115 V_{AC}, 220 V_{AC}. Τα τυποποιημένα ηλεκτρικά σήματα που παίρνουμε από μία μονάδα αναλογικών εξόδων έχουν συνήθως τάση -10 V έως +10V, 0 έως 10V ή ένταση ρεύματος 0 έως 20 mA, 4 έως 20mA. Μια μονάδα ψηφιακών εξόδων περιλαμβάνει 4, 8, 16, ή 32 εξόδους.

Ένα PLC περιλαμβάνει έναν καθορισμένο μέγιστο αριθμό μονάδων εισόδων και εξόδων που εξαρτάται από τις δυνατότητες της CPU. Τον αριθμό αυτό τον καθορίζει ο εκάστοτε κατασκευαστής.[14]



Σχ.2.7 Βασικό PLC και μονάδα επέκτασης [18]

2.3.3.3 Ειδικές I/O μονάδες

Οι ειδικές I/O μονάδες έχουν αναπτυχθεί για ορισμένες ανάγκες. Αυτές περιλαμβάνουν:

- Μετρητή υψηλής ταχύτητας

Ο μετρητής υψηλής ταχύτητας χρησιμοποιείται για να παρέχει μια διασύνδεση για τις εφαρμογές που απαιτούν μετρητή ταχυτήτων ο οποίος υπερβαίνει την ικανότητα του βαθμωτού προγράμματος PLC. Οι μετρητές υψηλής ταχύτητας χρησιμοποιούνται για να μετρούν παλμούς από ανιχνευτές, κωδικοποιητές και διακόπτες σε πολύ υψηλές ταχύτητες. Μονάδες οι οποίες μπορούν να μετρούν παλμούς μέχρι 75KHz είναι συνηθισμένες.

- Χειροκίνητη Μονάδα

Η χειροκίνητη μονάδα επιτρέπει τη χρήση του ρυθμιζόμενου με τα δάκτυλα διακόπτη για να τροφοδοτεί πληροφορίες παράλληλα στο PLC, για να χρησιμοποιηθούν στο σύστημα ελέγχου. Η πληροφορία του ρυθμιζόμενου με τα δάκτυλα διακόπτη είναι συνήθως σε δεκαδική μορφή δυαδικά κωδικοποιημένη (BCD) και καθιστά ικανό κάποιον να αλλάξει τα προκαθορισμένα σημεία εξωτερικά, χωρίς να τροποποιήσει το πρόγραμμα ελέγχου.

- Μονάδα TTL

Η μονάδα TTL επιτρέπει τη μετάδοση και τη λήψη των σημάτων TTL για επικοινωνία με τον επεξεργαστή του PLC. Τα σήματα επιπέδου TTL είναι σε μια μορφή που ο επεξεργαστής μπορεί να επιτύχει, και απαιτείται μόνο καταχώρηση.

- Μονάδα κωδικοποιητή-Μετρητή (Encoder-Counter)

Η μονάδα κωδικοποιητή-μετρητή επιτρέπει συνεχή παρακολούθηση από ένα αυξανόμενο ή απόλυτο κωδικοποιητή. Οι κωδικοποιητές κρατούν ίχνος από τη θέση των αξόνων. Ο κώδικας Gray είναι συνηθισμένος για απόλυτους κωδικοποιητές, με τη θέση να καθορίζεται από την αποκωδικοποίηση του κώδικα Gray.

- Μονάδα BASIC ή ASCII

Η μονάδα ASCII επιτρέπει τη μετάδοση και τη λήψη αρχείων ASCII. Αυτά τα αρχεία είναι συνήθως προγράμματα ή κατασκευαστικά δεδομένα. Οι μονάδες είναι φυσιολογικά προγραμματισμένες με εντολές BASIC. Ο χρήστης γράφει το πρόγραμμα σε γλώσσα (language) BASIC. Η μονάδα BASIC μπορεί να χρησιμοποιηθεί σε κείμενο εξόδου σε εκτυπωτή ή τερματικό, για να ενημερώσει ένα χρήστη.

- Μονάδα βηματικού κινητήρα

Η μονάδα βηματικού κινητήρα παρέχει ακολουθία παλμών σε ένα μεταφραστή βηματικού κινητήρα, και έτσι τον καθιστά ικανό να ελέγχει ένα βαθμωτό κινητήρα. Οι εντολές για τη μονάδα καθορίζονται από τον έλεγχο του προγράμματος στον PLC.

- Μονάδα BCD-Εξόδου

Η μονάδα BCD-Εξόδου καθιστά ικανό ένα PLC να λειτουργεί συσκευές οι οποίες απαιτούν BCD-κωδικοποιημένα (binary-coded decimal) σήματα, όπως απεικονίσεις επτά ψηφίων. Ορισμένες ειδικές μονάδες αναφέρονται σαν έξυπνα I/O επειδή έχουν το δικό τους μικροεπεξεργαστή σε πλακέτα που μπορεί να λειτουργεί παράλληλα με ένα PLC. Αυτές περιλαμβάνουν:

- Μονάδα PID

Η αναλογική μονάδα παραγωγής και ολοκλήρωσης [(PID) proportional-integral derivative)], χρησιμοποιείται στην επεξεργασία εφαρμογών ελέγχου που συσσωματώνουν PID αλγόριθμους. Ο αλγόριθμος είναι ένα σύνθετο πρόγραμμα βασισμένο σε μαθηματικούς υπολογισμούς. Η μονάδα PID επιτρέπει τον έλεγχο επεξεργασίας να πραγματοποιείται έξω από τη CPU. Αυτή η διεύθυνση προστατεύει τη CPU από επιβαρύνσεις με σύνθετους υπολογισμούς. Ο μικροεπεξεργαστής στην PID μονάδα επεξεργάζεται δεδομένα, συγκρίνει τα δεδομένα με τα σταθερά σημεία που παρέχονται από τη CPU και καθορίζει το κατάλληλο σήμα εξόδου.

- Μονάδα σερβομηχανισμού

Η μονάδα σέρβο χρησιμοποιείται σε εφαρμογές ελέγχου επεξεργασίας κλειστού βρόχου. Ο έλεγχος κλειστού βρόχου ολοκληρώνεται μέσω ανατροφοδότησης από τη συσκευή. Ο προγραμματισμός αυτής της μονάδας γίνεται μέσω του PLC, αλλά αν προγραμματιστεί μια φορά μπορεί να ελέγχει μια συσκευή ανεξάρτητα, χωρίς να παρεμβαίνει στην ομαλή λειτουργία του PLC.

- Μονάδα επικοινωνίας

Καθώς ολοκληρώνονται διάφορα συστήματα, τα δεδομένα πρέπει να δοκιμάζονται σε όλο το σύστημα. Οι PLC πρέπει να είναι ικανά να επικοινωνούν με τους υπολογιστές με μηχανές αριθμητικού ελέγχου [(CNC) Computer Numerical Controls)], με robot και με άλλους PLC.

Αυτή η μονάδα επιτρέπει στο χρήστη να συνδέσει το PLC στα τοπικά δίκτυα υψηλής ταχύτητας, κάτι το οποίο μπορεί να είναι διαφορετικό από την επικοινωνία δικτύου που παρέχεται με το PLC.

- Μονάδα γλώσσας

Η μονάδα γλώσσας καθιστά ικανό το χρήστη να γράφει προγράμματα σε γλώσσα υψηλού επιπέδου. Μέσω ενός μεταφραστή υψηλού επιπέδου γλώσσας, μετατρέπει τις εντολές υψηλού επιπέδου σε γλώσσα μηχανής η οποία είναι κατανοητή από τον επεξεργαστή του PLC. Η BASIC είναι η πιο δημοφιλής γλώσσα. Άλλες διαθέσιμες γλώσσες είναι οι C, Forth και Pascal.

- Μονάδα ομιλίας

Οι μονάδες ομιλίας τυπικά χρησιμοποιούνται για να ψηφιοποιήσουν μια ανθρώπινη φωνή που προσφέρει μια επιθυμητή λέξη, φράση ή πρόταση. Ο ψηφιοποιημένος ήχος αποθηκεύεται στη μονάδα μνήμης. Κάθε λέξη, φράση ή πρόταση είναι δοσμένη με αριθμό. Η λογική κλίμακα χρησιμοποιείται για να εξάγουμε το κατάλληλο μήνυμα στον κατάλληλο χρόνο.[22]

2.3.4 Τροφοδοτικό (Power Supply)

Η μονάδα τροφοδοσίας έχει ως σκοπό την εξασφάλιση των απαιτούμενων εσωτερικών τάσεων τροφοδοσίας των ηλεκτρονικών στοιχείων (τρανζίστορ, ολοκληρωμένα κυκλώματα κ.τ.λ.) που συνθέτουν ένα PLC. Τυπικές εσωτερικές τάσεις που χρησιμοποιούνται είναι: 5V DC, 9V DC, 24V DC.

Σε ορισμένα μοντέλα PLC, όταν το PLC δεν τροφοδοτείται από το δίκτυο, η μονάδα τροφοδοσίας διατηρεί το περιεχόμενο της μνήμης του PLC με την βοήθεια μιας μπαταρίας (συνήθως λιθίου), που διαθέτει. Σε άλλα μοντέλα PLC η παραπάνω μπαταρία βρίσκεται στην Κεντρική Μονάδα Επεξεργασίας.

Η κάρτα τροφοδοσίας δεν παράγει την απαιτούμενη τάση χειρισμού για εξωτερικές επαφές, ρελέ, ενδεικτικές λυχνίες κ.λ.π. Για την τροφοδοσία των παραπάνω υπάρχουν χωριστά κυκλώματα τα οποία είναι συνήθως γαλβανικά απομονωμένα από την υπόλοιπη συσκευή (L+, L-).

Ανάλογα με τον τύπο των καρτών εισόδων-εξόδων, η τάση αυτή δημιουργείται είτε από ιδιαίτερο τροφοδοτικό, για τάση χειρισμού 24V DC είτε από μετασχηματιστή, για τάση χειρισμού 115/220V. Υπάρχει η δυνατότητα ξεχωριστής τάσης χειρισμού για τις εισόδους και τις εξόδους. Για τη γαλβανική απομόνωση των τάσεων χειρισμού εισόδων- εξόδων χρησιμοποιούνται οπτοηλεκτρονικά στοιχεία ζεύξης (optocouplers). Αποτελούνται από μία διόδο, η οποία μετατρέπει το ηλεκτρικό ρεύμα σε φως και από ένα φωτοτρανζίστορ, που λειτουργεί ως διακόπτης, διεγερόμενο από την ακτινοβολία της διόδου.

Στα κυκλώματα εισόδου, η τάση εξόδου του φωτοτρανζίστορ μετατρέπεται, μέσω κυκλώματος προσαρμογής, σε μια τάση κατάλληλη για επεξεργασία από τη συσκευή (συνήθως 5V). Αντίθετα, στα κυκλώματα εξόδου το κύκλωμα προσαρμογής παράγει το απαραίτητο ρεύμα για τη διέγερση της διόδου. Η τάση που βγαίνει από το φωτοτρανζίστορ, ενισχύεται μέχρι σημείου που να μπορεί να διεγείρει τα διάφορα στοιχεία εντολής (ρελέ ισχύος, βαλβίδες κ.λ.π.). Η τάση λειτουργίας του ενισχυτή, αντιστοιχεί στην εξωτερική τάση χειρισμού του κυκλώματος εξόδου (π.χ. 24 V DC ή 220V AC).

Σχετικά με την τροφοδοσία του PLC, θα πρέπει να τηρούνται οι ακόλουθες προδιαγραφές:

1. Το κάλυμμα της τροφοδοσίας θα πρέπει να είναι κατάλληλο ώστε η θερμότητα που παράγεται από τα τροφοδοτικά κυκλώματα του PLC, κατά την λειτουργία τους, να απάγεται

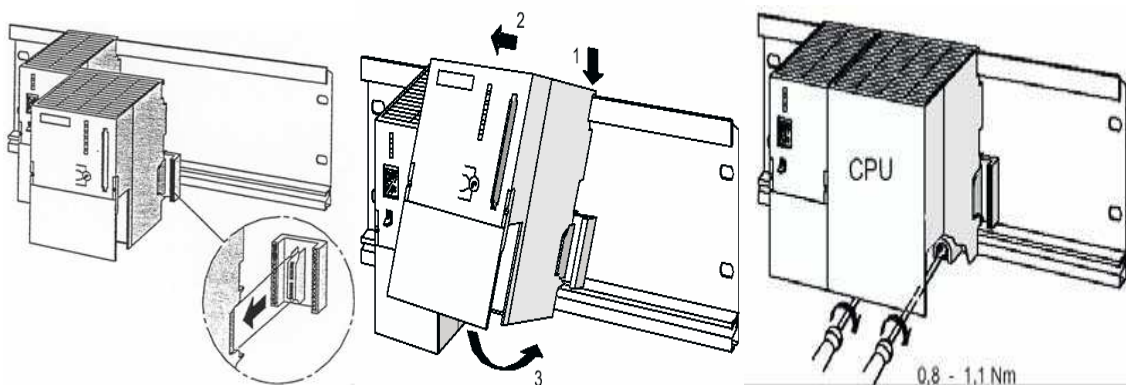
και έτσι να αποφεύγεται η υπερθέρμανση. Με τον τρόπο αυτό αυξάνεται η αξιοπιστία του συστήματος.

2. Η τροφοδοσία θα πρέπει να ελέγχεται από εργαστήρια ή φορείς, όπως το Underwriters Laboratories (UL) και το Canadian Standards Association (CSA), ώστε να πληρούν συγκεκριμένα πρότυπα. Στα προαναφερθέντα εργαστήρια γίνεται έλεγχος της θερμοκρασίας και της ηλεκτρικής απομόνωσης.

3. Η τροφοδοσία θα πρέπει επίσης να πληροί πρότυπα σχετικά με την απομόνωση από ανεπιθύμητο θόρυβο.[9],[20]

2.3.5 Το πλαίσιο στήριξης μονάδων

Αρκετοί τύποι PLC φέρουν ένα πλαίσιο επάνω στο οποίο τοποθετούνται όλες οι επιμέρους βαθμίδες που το συγκροτούν. Σε αυτό το πλαίσιο είναι επίσης ενσωματωμένο και το σύστημα ζυγών, δηλαδή το σύστημα των αγωγών (BUS) μέσω των οποίων επικοινωνούν οι διάφορες βαθμίδες για την τροφοδοσία τους αλλά και για την απαραίτητη ανταλλαγή πληροφοριών. Αν οι θέσεις του κεντρικού πλαισίου που διατίθεται, δεν επαρκούν για να τοποθετηθούν οι μονάδες εισόδων και εξόδων που απαιτούνται σε μια συγκεκριμένη εφαρμογή, τότε χρησιμοποιούνται περισσότερα πλαίσια επέκτασης για την τοποθέτηση των επιπλέον μονάδων. Κάθε πλαίσιο επέκτασης συνδέεται με το κεντρικό πλαίσιο ή με τα άλλα πλαίσια μέσω ειδικής μονάδας διασύνδεσης και καλωδίου. [14],[17]



Σχ.2.8 Πλαίσια στήριξης μονάδων

2.3.6 Βοηθητικές μονάδες

Πρόκειται για συσκευές που δεν είναι απαραίτητες για τη λειτουργία του PLC, σίγουρα όμως δίνουν καλύτερη εποπτεία και έλεγχο του αυτοματισμού. Οι κυριότερες είναι:

- Μονάδαπροσομοίωσης: Είναι μία σειρά από διακόπτες με τους οποίους μπορούμε να κάνουμε εργαστηριακό έλεγχο του αυτοματισμού.
- Modem: Είναι συσκευές με τις οποίες μπορούμε να διαβιβάσουμε πληροφορίες και να δώσουμε εντολές μέσω τηλεφωνικής γραμμής.
- Οθόνες (monitors): Για έγχρωμες απεικονίσεις μιμικών διαγραμμάτων υψηλής ακρίβειας.
- Εκτυπωτές όλων των τύπων.[14]

2.3.7 Θύρα επικοινωνίας

Η ανταλλαγή πληροφοριών μεταξύ χρήστη και συσκευής μπορεί να γίνει είτε παράλληλα είτε σειριακά. Τα PLCs χρησιμοποιούν συνήθως σειριακή ανταλλαγή πληροφοριών με θύρα RS 232C.

Το PLC συνδέεται με τον H/Y ή τον Hand Programmer μέσω ειδικού καλωδίου επικοινωνίας. Η επικοινωνία γίνεται με τις σειριακές θύρες RS232 και εναλλακτικά με RS485 και ethernet. Συνήθως όμως στον υπολογιστή χρησιμοποιείται μία 25pin θύρα COM. Στην περίπτωση αυτή απαιτείται ειδικός μετατροπέας 25pin-9pin. Στην περίπτωση προγραμματισμού μέσω H/Y η θύρα επικοινωνίας πρέπει να δηλώνεται μέσω του προγράμματος.

Το καλώδιο επικοινωνίας συνήθως διαθέτει μια σειρά μικροδιακοπών, μέσω των οποίων ρυθμίζεται η ταχύτητα μετάδοσης δεδομένων.

Ακόμη χρησιμοποιείται και το πρωτόκολλο Modbus ή αλλιώς DF1. Άλλες επιλογές είναι η χρήση fieldbus, το οποίο δεν είναι κάτι άλλο από μια κατηγορία πρωτοκόλλων δικτυακής επικοινωνίας για διανεμημένα συστήματα ελέγχου πραγματικού χρόνου, το οποίο πλέον έχει κυρωθεί και επίσημα ως IEC 61158. Δύο μέθοδοι επικοινωνίας οι οποίες ανήκουν στην κατηγορία fieldbuses είναι οι DeviceNet και Profibus.

Τα σύγχρονα PLC επικοινωνούν μέσα από ένα δίκτυο με άλλα συστήματα, όπως ένας υπολογιστής ο οποίος τρέχει ένα πρόγραμμα τύπου επίβλεψης ελέγχου και ανάκτησης δεδομένων SCADA (Supervisory Control And Data Acquisition) ή μέσω ενός Web Browser.

Τα PLC, τα οποία χρησιμοποιούνται σε μεγαλύτερα συστήματα εισόδου/εξόδου, μπορεί να χρησιμοποιούν peer-to-peer (P2P) επικοινωνία μεταξύ των μικροεπεξεργαστών. Αυτό επιτρέπει στα επιμέρους τμήματα ενός πολύπλοκου συστήματος να έχουν αυτονομία. Αυτοί οι σύνδεσμοι επικοινωνίας χρησιμοποιούνται και την διάδραση ανθρώπου-μηχανής μέσω συσκευών όπως πληκτρολόγια ή υπολογιστικού τύπου σταθμών εργασίας (PC-type workstations) ή μέσω οθονών touch. Κάποια από τα σημερινά PLC μπορούν να επικοινωνούν με διάφορους τρόπους συμπεριλαμβανομένων RS-485, Coaxial ακόμη και Ethernet για συστήματα ελέγχου εισόδου/εξόδου με ταχύτητες δικτύου μέχρι και 100 Mbit/s. [14],[15]

2.3.7.1 Καλώδια επικοινωνίας

2.3.7.1.1 RS-232

Στις τηλεπικοινωνίες το RS-232 (Recommended Standard 232) είναι ένα πρωτόκολλο για σειριακή μετάδοση ψηφιακών σημάτων δεδομένων, το οποίο συνδέει ένα Τερματικό δεδομένων εξοπλισμού (Data Terminal Equipment) με ένα κύκλωμα δεδομένων - τερματικού εξοπλισμού (Data Circuit-terminating Equipment). Η επικοινωνία μέσω του πρωτοκόλλου αυτού είναι κατά βάση ασύγχρονη. Στο Σχήμα 1.1 απεικονίζεται χαρακτηριστικά η δομή ενός προς μετάδοση frame είναι. Τα D_x , όπου $x=0..7$ είναι τα ως προς μετάδοση bits.

Οι πιο κοινοί ρυθμοί μετάδοσης είναι αυτοί των 300, 600, 1200, 1800, 2000, 2400, 4800, 9600, 19200 bits ανά δευτερόλεπτο.

Τα επίπεδα των τάσεων οι οποίες αντιστοιχούν στα δυαδικά ψηφία 1 και 0 είναι τα 12V και -12V αντίστοιχα. Για την ομαλή διάδοση πληροφορίας σύμφωνα με το πρότυπο RS-232 χρειάζονται κάποιες μονάδες τερματικού (DTE) και μονάδες επικοινωνίας (DCE).

Γενικότερα ενώ το πρότυπο RS-232 προβλέπει ένα μεγάλο αριθμό pin/σημάτων κυρίως για χειραγία τα οποία και διασφαλίζουν την ορθότητα και ακεραιότητα των δεδομένων, υπάρχει επιπλέον η δυνατότητα επικοινωνίας με χρήση μόνο τριών pin RxD, TxD και GND. Είναι προφανές ότι σε αυτή την περίπτωση μειώνουμε το κόστος σε hardware συνδέσεις έχοντας όμως μικρότερη αξιοπιστία για την ορθότητα της ληφθείσας από τον δέκτη πληροφορίας.

2.3.7.1.2 Αναφορά στο RS-485

Στις τηλεπικοινωνίες το EIA-485 (πρώην RS-485) είναι ένα πρωτόκολλο για 2 γραμμές επικοινωνίας, μονόδρομης, πολλαπλών κόμβων. Έχει την δυνατότητα επέκτασης σε αποστάσεις μέχρι και 1200 μέτρα.

Αυτό το πρωτόκολλο έχει πλέον περάσει υπό την εποπτεία και ευθύνη του φορέα TIA (Telecommunications Industry Association) και φέρει το όνομα " TIA-485-A Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems (ANSI/TIA/EIA-485-A-98) (R2003)" το οποίο πρακτικά σημαίνει ότι το πρωτόκολλο επανεκδόθηκε χωρίς κάποιες επιμέρους αλλαγές.

Κάνοντας χρήση του RS-485 υπάρχει η δυνατότητα επίτευξης ταχυτήτων μέχρι και 10 Mbits/sec, ενώ το συνολικό μήκος των αντίστοιχων γραμμών επικοινωνίας ανέρχεται μέχρι τα 1000 μέτρα. Ακόμη υπάρχει η δυνατότητα σύνδεσης μέχρι και 32 διαφορετικών μονάδων οδήγησης και λήψης στον ίδιο δίαυλο.[15]

2.3.8 Ενδεικτικά LED

Τα ενδεικτικά led βρίσκονται πάνω στο PLC, από ένα για κάθε είσοδο και έξοδο, δίπλα στο led αναγράφεται ο αριθμός της εισόδου ή εξόδου που παριστάνει. Όταν η είσοδος ή η έξοδος τροφοδοτείται, τότε ανάβει το αντίστοιχο led. Είναι πολύ χρήσιμα όταν κάνουμε δοκιμή του προγράμματος, χωρίς να έχουμε συνδέσει ακόμη τις εισόδους και εξόδους.

2.3.9 Συσκευή προγραμματισμού

Η συσκευή προγραμματισμού είναι μια τελείως ξεχωριστή συσκευή από τη μονάδα αυτοματισμού. Χρησιμοποιείται για την εισαγωγή του προγράμματος στο PLC και την παρακολούθηση της εξέλιξης του αυτοματισμού μέσα από την οθόνη που διαθέτει. Με έναν μόνο προγραμματιστή μπορούμε να χειριζόμαστε όλες τις μονάδες της ίδιας εταιρίας PLC, σε μια αυτοματοποιημένη εγκατάσταση.

Η μονάδα προγραμματισμού μπορεί να είναι μια οθόνη υγρών κρυστάλλων (LCD) με ακροδέκτες που χειρίζονται με το χέρι, μια απλής γραμμής μονάδα LED ή ένα πληκτρολόγιο και μια μονάδα οθόνης. Η οθόνη προσφέρει το πλεονέκτημα της έκθεσης μεγάλων ποσών της λογικής στην οθόνη, απλοποιώντας την ερμηνεία του προγράμματος.

Η μονάδα προγραμματισμού επικοινωνεί με τον επεξεργαστή μέσω μιας σειριακής ή παράλληλης γραμμής επικοινωνίας δεδομένων. Εάν η μονάδα προγραμματισμού δεν είναι σε λειτουργία, μπορεί να είναι αποσυνδεδεμένη και απομακρυσμένη. Η απομάκρυνση της μονάδας προγραμματισμού δεν θα επηρεάσει τη λειτουργία του προγράμματος του χρήστη.

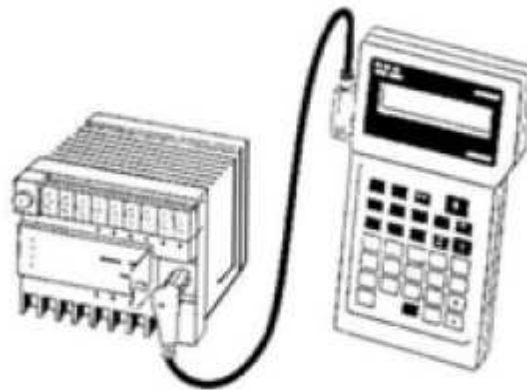
Οι συσκευές προγραμματισμού επιτρέπουν στο χρήστη να εισάγει (enter), να αλλάζει (change), ή να παρακολουθεί (monitor) το πρόγραμμα ενός PLC ελεγκτή. Ο εύκολος στη χρήση προγραμματιστικός εξοπλισμός είναι ένα σημαντικό χαρακτηριστικό του PLC. Βιομηχανικά CRT τερματικά μπορούν να χρησιμοποιηθούν για τον προγραμματισμό του ελεγκτή. Αυτά τα τερματικά περιέχουν μονάδες απεικόνισης με πληκτρολόγιο και τα απαραίτητα ηλεκτρονικά για να επικοινωνούν με τη CPU και για να απεικονίζουν δεδομένα. Το CRT, όπως οι μονάδες απεικόνισης, προσφέρουν το πλεονέκτημα να απεικονίζουν μεγάλη ποσότητα λογικής στην οθόνη, κάτι το οποίο απλοποιεί την ερμηνεία του προγράμματος.

Τα βιομηχανικά τερματικά αφιερώνονται σε ένα είδος του PLC. Στις περισσότερες περιπτώσεις πρέπει να προσαρτώνται στο PLC για να είναι ικανά να προγραμματίζουν. Τα αφιερωμένα τερματικά μπορούν να χρησιμοποιηθούν για να ανιχνεύουν λάθη στην κλιμακωτή λογική, ενώ ο PLC εκτελεί (miming). Μπορούν να εξαναγκάσουν τις εισόδους ή

τις εξόδους σε ON ή OFF για ανίχνευση λάθους. Το βασικό μειονέκτημα αυτών των τερματικών είναι ότι απαιτείται ένα τερματικό για κάθε διαφορετικό είδος PLC που μπορεί να έχει η βιομηχανία. Είναι επίσης αρκετά ακριβά και χρησιμοποιούνται για να προγραμματίζουν ένα είδος PLC.

Οι μίνι προγραμματιστές, γνωστοί επίσης και σαν προγραμματιστές χειρός, είναι ένας φτηνός και φορητός τρόπος προγραμματισμού μικρών PLC. Η οθόνη είναι συνήθως LED ή οθόνη υγρών κρυστάλλων (LCD) και το πληκτρολόγιο αποτελείται από αριθμητικά πλήκτρα, πλήκτρα εντολών προγράμματος και πλήκτρα ειδικών λειτουργιών.

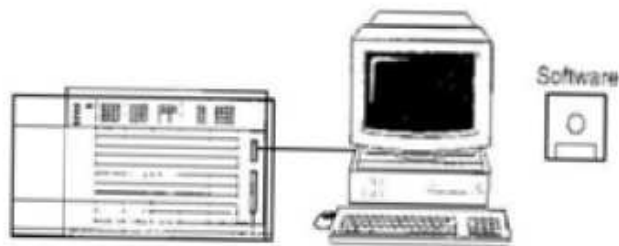
Οι προγραμματιστές χειρός είναι φτηνοί και εύχρηστοι. Τυπικά έχουν πληκτρολόγια μεμβράνης πάνω στην οθόνη, τα οποία έχουν ανοσία στις μολύνσεις του εργοστασιακού περιβάλλοντος. Οι προγραμματιστές χειρός κατατάσσονται στις βουβές συσκευές. Αυτό σημαίνει ότι δεν υπάρχει εξυπνάδα στη συσκευή. Έτσι, όλο το μυαλό για τη λειτουργία του PLC και για το σύστημα προγραμματισμού, βρίσκεται στο PLC. Πρέπει να προσαρτώνται στο PLC για να τεθούν σε χρήση. Είναι ειδικευμένοι στην ανίχνευση λαθών και μπορούν εύκολα να μεταφερθούν στο σύστημα κατασκευής και να συνδεθούν στο PLC. Όταν συνδεθούν, μπορούν να χρησιμοποιηθούν για να ελέγχουν την κατάσταση των εισόδων, εξόδων, μεταβλητών, μετρητών, χρονιστών κλπ. Αυτό εξαλείφει την ανάγκη μεταφοράς μεγάλων συσκευών προγραμματισμού στο πάτωμα του εργοστασίου.



Σχ.2.9 Συσκευή προγραμματισμού χειρός[22]

Ο προσωπικός υπολογιστής είναι, αυτή τη στιγμή, η πιο συνηθισμένη συσκευή που χρησιμοποιείται για προγραμματισμό. Μετατρέπεται σε προγραμματιστή για προγραμματιζόμενους ελεγκτές με τη βοήθεια του software προγραμματισμού που παρέχεται σαν ένα πακέτο δισκετών. Ο ίδιος υπολογιστής μπορεί να προγραμματίζει οποιοδήποτε είδος PLC, το οποίο έχει διαθέσιμο λογισμικό γι' αυτό. Αυτό κάνει πιθανή την απομάκρυνση του προγραμματισμού μακριά από τη φυσική του θέση από τους προγραμματιζόμενους ελεγκτές. Όταν το πρόγραμμα ολοκληρωθεί, αποθηκεύεται σε κάποια μορφή μαζικής αποθήκευσης και διαβιβάζεται στους προγραμματιζόμενους ελεγκτές, όταν απαιτείται.

Ο υπολογιστής μπορεί επίσης να χρησιμοποιηθεί για να τεκμηριώσει το πρόγραμμα του PLC. Σημειώσεις για τους τεχνικούς μπορεί να προστεθούν, και το κλιμακωτό διάγραμμα μπορεί να εξαχθεί σε ένα εκτυπωτή για να αποκτήσουμε ένα αντίγραφο. Αυτή η τεκμηρίωση είναι ανεκτίμητη για την κατανόηση και για την ανίχνευση λαθών στα κλιμακωτά διαγράμματα. Ο προγραμματιστής μπορεί να προσθέσει σημειώσεις, ονόματα για τις συσκευές εισόδου ή εξόδου και σχόλια τα οποία μπορεί να είναι χρήσιμα για την ανίχνευση λαθών και για τη συντήρηση.[22]



Ο προσωπικός υπολογιστής χρησιμοποιεί λογισμικό προγραμματισμού για να το μετατρέψει σε ένα προγραμματιστή για PLC.

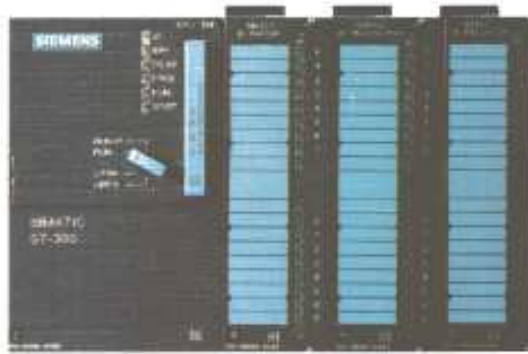
Σχ.2.10 Προσωπικός υπολογιστής ως συσκευή προγραμματισμού[22]

2.4 Τύποι PLC

Τα PLCs χωρίζονται σε δύο κατηγορίες (ανεξάρτητες των εταιρειών). Τα compact (συμπαγούς μορφής) και τα modular (δομοστοιχειωτής δομής).



Σχήμα 2.11 Compact PLC

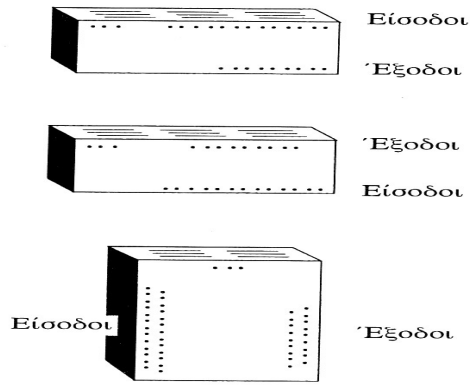


Σχήμα 2.12 Modular

Τα πρώτα είναι μία συμπαγής συσκευή με CPU, τροφοδοτικό και συγκεκριμένο αριθμό I/O (που ποικίλει ανάλογα με την εταιρεία). Τα δεύτερα περιλαμβάνουν μία βάση, στην οποία "κουμπώνουν" οι μονάδες επεξεργασίας, τροφοδοσίας, εισόδων, εξόδων. Ένα PLC μπορεί να διαθέτει περισσότερες από μια μονάδες εισόδων και εξόδων, ανάλογα με τον επιθυμητό αριθμό εισόδων ή εξόδων. Επομένως, αν σε κάποιο αυτοματισμό, προκειμένου να τον επεκτείνουμε, χρειαστούμε κι άλλες εισόδους ή εξόδους, που δεν υπάρχουν στην αρχική κατασκευή, έχουμε τη δυνατότητα να προσθέσουμε μία ή περισσότερες μονάδες εισόδων ή εξόδων, διατηρώντας την ίδια CPU και το ίδιο τροφοδοτικό.[14]

Compact PLC

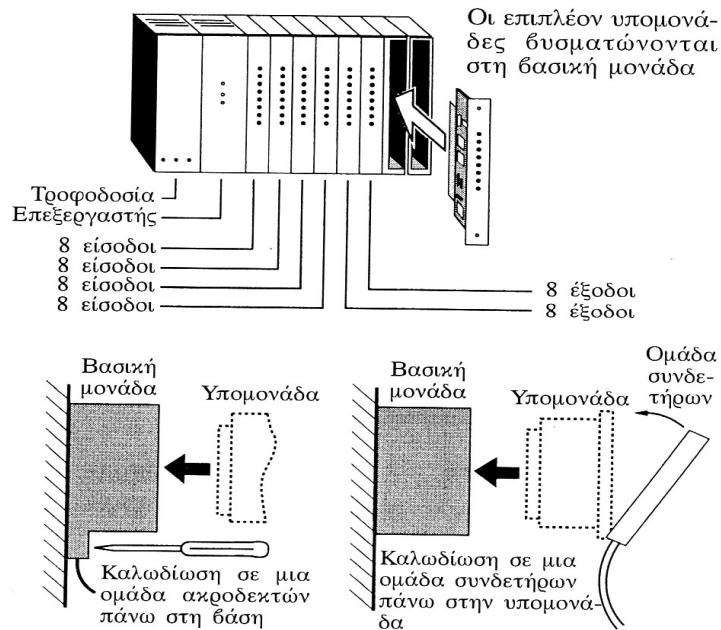
Σ' αυτήν την κατηγορία ανήκουν τα PLC που όλα τα επιμέρους στοιχεία, που απαρτίζουν ένα PLC, είναι ενσωματωμένα σε μια συσκευή. Είναι περιορισμένων δυνατοτήτων καθώς έχουν 48 το πολύ εισόδους και εξόδους, όλες με τα ίδια χαρακτηριστικά, καθώς και μικρό αριθμό χρονικών και απαριθμητών. Τα παλαιότερα μοντέλα δεν ήταν επεκτάσιμα. Στα νεότερα μοντέλα υπάρχει δυνατότητα περιορισμένης επέκτασης. Το πλεονέκτημά τους είναι το χαμηλό κόστος τους.



Σχ.2.13 Compact

Modular PLC

Σ' αυτήν την κατηγορία κάθε μονάδα (module) του PLC είναι ξεχωριστή και συνδέονται όλες μαζί πάνω στο πλαίσιο τοποθέτησης μονάδων. Είναι επεκτάσιμα και χρησιμοποιούνται συνήθως όταν έχουμε μεγάλο αριθμό εισόδων και εξόδων. Έτσι μπορούμε να διαλέξουμε την κεντρική μονάδα και τις μονάδες εισόδων / εξόδων με τα χαρακτηριστικά που επιθυμούμε.[17]



Σχ.2.14 Modular

2.4.1. Μη επεκτάσιμοι ελεγκτές

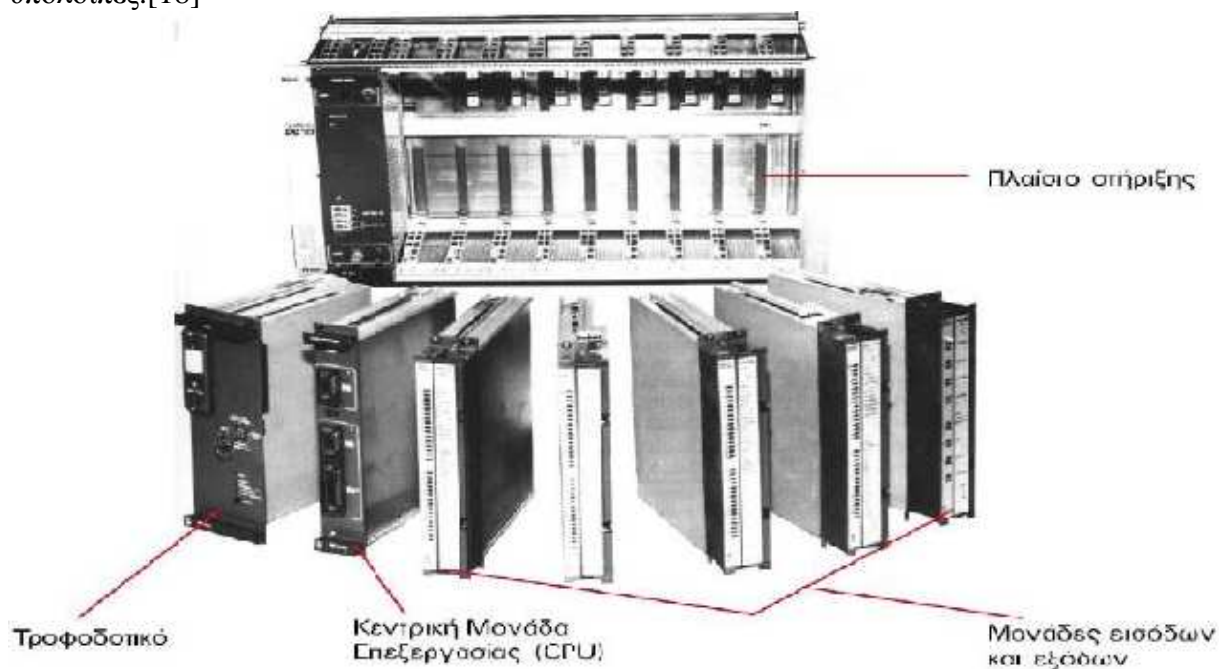
Πρόκειται για ελεγκτές μικρών δυνατοτήτων και διατίθεται σε ενιαία, συμπαγή συσκευασία. Η συσκευασία αυτή περιλαμβάνει τον επεξεργαστή, την μνήμη, το τροφοδοτικό και ένα συγκεκριμένο αριθμό εισόδων και εξόδων. Αποτελείται από επιπρόσθετες λειτουργίες, όπως συγκριτές, μαθηματικές και λογικές συναρτήσεις κ.α., ενώ πλεονεκτεί εκεί όπου υπάρχει έλλειψη χώρου.

Οι περισσότεροι προγραμματιζόμενοι ελεγκτές τύπου compact έχουν την δυνατότητα να συνδεθούν με μονάδες επεκτάσεων. Ουσιαστικά πρόκειται για μονάδες οι οποίες αποτελούνται από επιπλέον εισόδους και εξόδους και σκοπό έχουν την αύξηση δυνατοτήτων σε εφαρμογές υψηλότερων απαιτήσεων. Χαρακτηριστικό αυτού του τύπου P.L.C. είναι το μικρό κόστος ενώ σε περίπτωση βλάβης μιας από εσωτερικές μονάδες του, χρειάζεται αντικατάσταση ολόκληρο το σύστημα.[18]

2.4.2. Επεκτάσιμοι ελεγκτές

Πρόκειται για ελεγκτές όπου οι διάφορες λειτουργικές μονάδες, όπως ο επεξεργαστής, το τροφοδοτικό και είσοδοι/έξοδοι, είναι τοποθετημένες σε ανεξάρτητες υπομονάδες. Οι υπομονάδες αυτές έχουν την δυνατότητα βυσμάτωσης πάνω σε μία βάση, (πλαίσιο στήριξης-rack). Όταν μια μονάδα τοποθετείται ('κουμπώνει') στη βάση, δημιουργείται μια ηλεκτρική σύνδεση πάνω σε μια πλακέτα τυπωμένου κυκλώματος (μονάδα συρταρωτής σύνδεσης). Ο επεξεργαστής του P.L.C. συνδέεται επίσης στην πλακέτα του κυκλώματος (πλαίσιο στήριξης) και μπορεί να επικοινωνεί με όλες τις μονάδες στη βάση.

Η αύξηση των επιμέρους μονάδων έχει σαν αποτέλεσμα να πολλαπλασιάζονται οι δυνατότητες τους και να καθίστανται ικανοί να καλύψουν πιο απαιτητικές εφαρμογές. Πλεονεκτούν εκεί που χρειάζεται μεγάλος αριθμός εισόδων και εξόδων. Σε περίπτωση βλάβης σε μία από τις υπομονάδες, γίνεται αντικατάσταση της χωρίς να πειραχτούν οι υπόλοιπες.[18]



Σχ.2.15 Βασικές υπομονάδες ενός επεκτάσιμου P.L.C[18]



Σχ.2.16Επεκτάσιμοι προγραμματιζόμενοι λογικοί ελεγκτές (modular P.L.C.) [18]

2.5 Πλεονεκτήματα του PLC

Τα πρώτα μεγάλα πλεονεκτήματα των PLC αφορούν στους κατασκευαστές εξοπλισμού αυτοματισμών και πινάκων αυτοματισμού που είναι το κόστος κατασκευής ενός PLC είναι σημαντικά μικρότερο από το κόστος παραγωγής ενός μεγάλου αριθμού βοηθητικών ρελέ, χρονικών και απαριθμητών. Ο χρόνος κατασκευής του αυτοματισμού είναι μηδαμινός σε σχέση με την κατασκευή ενός κλασσικού πίνακα αυτοματισμού.

Υπάρχουν όμως πολλά πλεονεκτήματα που έχουν σχέση με τον τελικό χρήστη, δηλαδή τις βιομηχανίες που χρησιμοποιούν τους αυτοματισμούς.

Τα PLC ελαχιστοποιούν το κόστος συντήρησης του πίνακα αυτοματισμού. Το κόστος αυτό αναλύεται ως εξής: α) Συχνότητα βλαβών, β) χρόνος εντοπισμού μιας βλάβης και αποκατάστασής της. Δηλαδή, όταν υπάρχει μια βλάβη στον πίνακα μιας εγκατάστασης κλασσικού αυτοματισμού, υπάρχει καθυστέρηση στην παραγωγή μέχρι να εντοπιστεί η βλάβη. Αφού εντοπιστεί, πρέπει να έχουμε διαθέσιμο στην αποθήκη το κατάλληλο ανταλλακτικό, γιατί διαφορετικά θα υπάρξει σημαντική καθυστέρηση, κατά τη παραγγελία και προμήθεια. Στον αυτοματισμό με PLC δεν υπάρχει ουσιαστικό θέμα βλάβης εσωτερικά του πίνακα της εγκατάστασης (δεν υπάρχουν μηχανικές επαφές) και κάνει οικονομία και στην κατανάλωση ενέργειας. Βέβαια το PLC σπάνια χαλάει, όμως οι εγγυήσεις είναι συνήθως πάρα πολύ μεγάλες.

Τα PLC είναι ευέλικτα στην τροποποίηση της λειτουργίας του αυτοματισμού σε οποιοδήποτε στάδιο (μελέτη, σχεδίαση, κατασκευή, λειτουργία). Δηλαδή αν υποθέσουμε ότι θέλουμε να κάνουμε μια αλλαγή στον αυτοματισμό, αυτή μπορεί να γίνει μέσα σε λίγα λεπτά, αρκεί μόνο να αλλάξουμε το πρόγραμμα. Σε έναν πίνακα κλασσικού αυτοματισμού τέτοιες αλλαγές είναι πολύ δύσκολες, ακριβές και χρονοβόρες. Το ίδιο μηχάνημα μπορεί να χρησιμοποιηθεί σε εντελώς διαφορετικές εφαρμογές.

Ο αυτοματισμός με PLC επεκτείνεται πολύ εύκολα. Αυτό γίνεται είτε απλά αλλάζοντας το πρόγραμμα, είτε με την τοποθέτηση νέων μονάδων εισόδων και εξόδων. Κάθε επέκταση στον κλασσικό αυτοματισμό είναι πολύ δύσκολη.

Διευκολύνεται αφάνταστα ο εντοπισμός βλαβών διότι σε κάθε εξωτερική εντολή υπάρχει και το αντίστοιχο LED. Είναι πάρα πολύ εύκολη η παρακολούθηση της ροής του αυτοματισμού, της εκτέλεσης του προγράμματος και μέσω διαγνωστικών να εντοπίσουμε τυχόν βλάβες.

Ο αυτοματισμός με PLC μας παρέχει μεγάλες δυνατότητες. Μπορούμε να δημιουργήσουμε πολύ εύκολα πολύπλοκες και «έξυπνες» επεξεργασίες.

Η τοποθέτησή του μπορεί να γίνει χωρίς κίνδυνο ακόμα και μέσα σε μαγνητικά πεδία, τα οποία καθιστούν προβληματική τη λειτουργία των ηλεκτρονόμων.

Έχουν μεγάλες δυνατότητες δικτύωσης με πρότυπα βιομηχανικά δίκτυα. Μπορεί να συνδεθεί με περιφερειακές μονάδες ή υπολογιστές, για έλεγχο, επιτήρηση συντονισμό και κεντρική οργάνωση των εγκαταστάσεων, ανάλογα με τις απαιτήσεις του αυτοματισμού. Σε μια μοντέρνα εγκατάσταση που χρησιμοποιεί αυτοματισμούς με PLC, παρέχονται δυνατότητες σύνδεσης με τον κεντρικό ηλεκτρονικό υπολογιστή, και το ενδοεταιρικό δίκτυο.

Το PLC καταλαμβάνει ελάχιστο χώρο σε σχέση με τον πίνακα κλασσικού αυτοματισμού.

Δεν απαιτεί κάποια ανώτερη γλώσσα προγραμματισμού, αλλά προγραμματίζονται: 1) με η γλώσσα προγραμματισμού Ladder (διάγραμμα LADDER) που είναι προσαρμοσμένη στο βιομηχανικό αυτοματισμό και άρα είναι προσιτή στο προσωπικό που μέχρι σήμερα συντηρούσε τους κλασσικούς πίνακες αυτοματισμού με βάση το γνωστό στους ηλεκτρολόγους συνδεσμολογικό σχέδιο με επαφές, χρονικά κ.τ.λ. 2) με λογικό διάγραμμα(FBD) και 3) με τη γλώσσα STL (Statement List). Μας δίνουν δυνατότητα αντιγραφής εφαρμογών.

Ο αυτοματισμός παραδίδεται συντομότερα σε λειτουργία, γιατί η μελέτη μπορεί να γίνεται παράλληλα με την τοποθέτηση και συρμάτωση του PLC.

Είναι αξιόπιστα, έχουν μεγάλη ταχύτητα επεξεργασίας δεδομένων, σχετικά χαμηλό κόστος και πολύ μεγάλη διάρκεια ζωής.

Τα σήματα στις εισόδους και εξόδους μπορεί να είναι: Ψηφιακά (digital): σήματα με σταθερό πλάτος των οποίων η διάρκεια μπορεί να μεταβάλλεται, π.χ. ένας παλμός. Αναλογικά (analog): π.χ. μια ημιτονοειδή τάση, κάποιο μεταβαλλόμενο συνεχές μέγεθος (πίεση, θερμοκρασία, υγρασία κ.τ.λ.)

Είναι οικονομικότερα για αυτοματισμούς στους οποίους χρησιμοποιείται μεγάλος αριθμός ρελέ.

Η εμπειρία έχει δείξει ότι μόνο το 5% των σφαλμάτων αυτοματοποιούμενου συστήματος προέρχεται από σφάλμα του PLC. [9],[10],[17],[18]

2.6 Μειονεκτήματα των PLCs :

- Μικρό μέγεθος μνήμης / Αδυναμία αποθήκευσης μεγάλων ποσοτήτων δεδομένων.
- Το πρωτόκολλο επικοινωνίας των PLCs (π.χ. Ethernet) πρέπει να λειτουργεί ξεχωριστά από το αντίστοιχο πρωτόκολλο του εταιρικού δικτύου γιατί σε περιπτώσεις συνεχούς επικοινωνίας, οποιαδήποτε υπερφόρτωση του δικτύου προκαλεί βλάβες στο σύστημα αυτοματισμού.
- Υπάρχει αυξημένος κίνδυνος ύπαρξης σφάλματος στο πρόγραμμα λειτουργίας του PLC.
 - Μειωμένη αξιοπιστία λόγω περίπλοκων κυκλωμάτων.
 - Περιορισμένος αριθμός εισόδων και εξόδων. [1]

2.7 Αρχή λειτουργίας ενός Προγραμματιζόμενου Λογικού Ελεγκτή

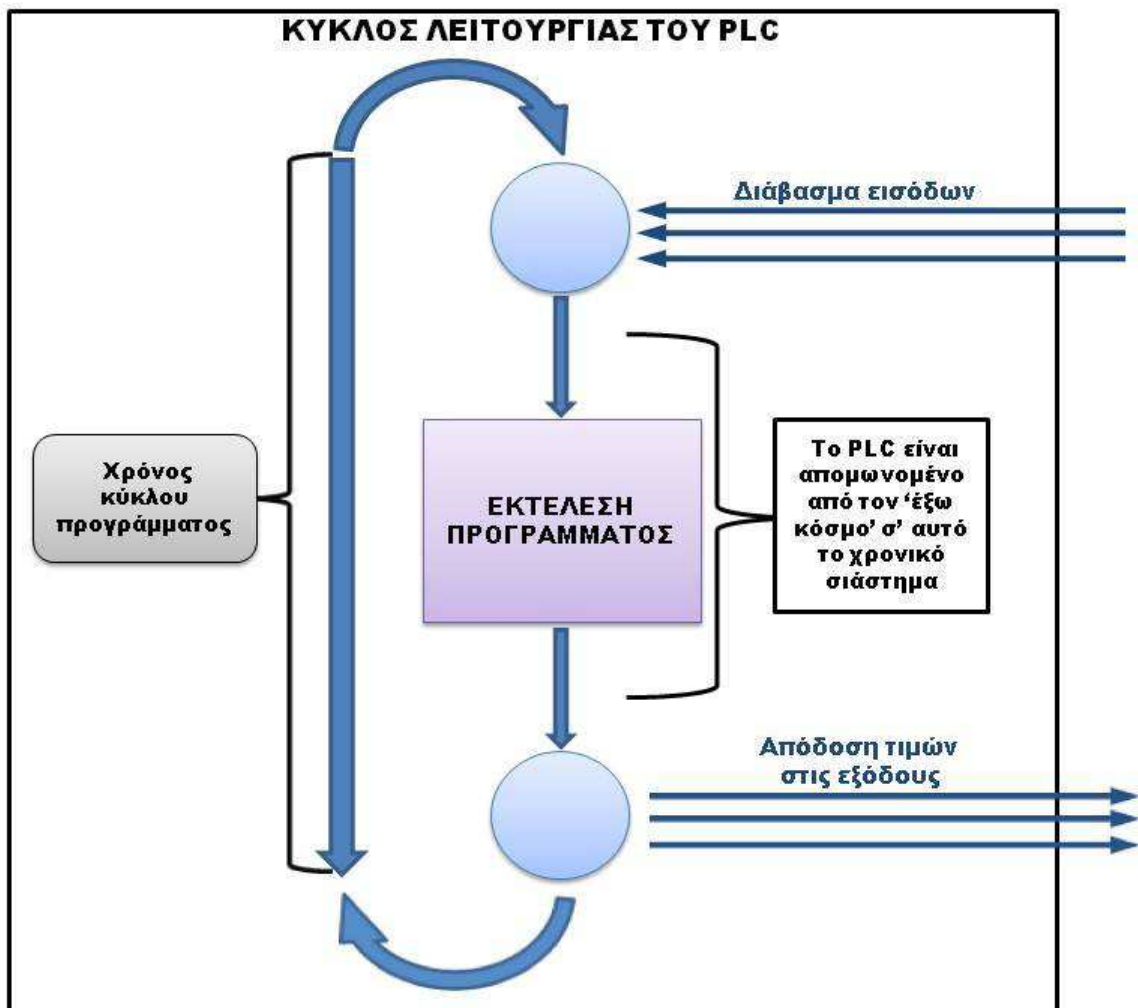
Ας υποθέσουμε ότι ένα PLC βρίσκεται σε κατάσταση εκτέλεσης του αυτοματισμού (RUN). Τα βήματα που ακολουθεί κατά τη λειτουργία του είναι τα εξής:

Βήμα 1ο: Στην αρχή ο μικροεπεξεργαστής “διαβάζει” της εισόδους. Αυτό σημαίνει ότι για κάθε είσοδο ελέγχει αν έχει “υψηλή” τάση (λογικό “1”) ή “χαμηλή” τάση (λογικό “0”). Η τιμή “0” ή “1” για κάθε είσοδο αποθηκεύεται σε μια ειδική περιοχή της μνήμης η οποία ονομάζεται εικόνα εισόδων (input image). Την εικόνα εισόδων μπορείτε να την φανταστείτε σαν έναν πίνακα, όπου ο μικροεπεξεργαστής σημειώνει τις τιμές που διάβασε. π.χ. είσοδος I1=“1”. I2=“0”, I3=“0” κοκ.

Βήμα 2ο: Στη συνέχεια ο μικροεπεξεργαστής χρησιμοποιώντας σαν δεδομένα τις τιμές των εισόδων, που διάβασε, εκτελεί τις εντολές του προγράμματος, το οποίο λειτουργεί τον αυτοματισμό. Το πρόγραμμα αυτό στην ουσία περιέχει μια σειρά από λογικές πράξεις. Η εκτέλεση του προγράμματος θα δώσει αποτελέσματα για τις εξόδους. Τα αποτελέσματα αυτά αποθηκεύονται στην ειδική περιοχή της μνήμης που ονομάζεται εικόνα εξόδων (output image). Όπως η εικόνα εισόδων, έτσι και η εικόνα εξόδων περιέχει την τιμή (“0” ή “1”) για κάθε έξοδο. Σημειώνουμε ότι οι τιμές αυτές προκύπτουν από την εκτέλεση των λογικών πράξεων του προγράμματος.

Βήμα 3ο: Στη συνέχεια ο μικροεπεξεργαστής θέτει τις τιμές της εικόνας εξόδων στις εξόδους. Αυτό σημαίνει ότι θα δοθεί “υψηλή” τάση σε όποια έξοδο έχει “1” και χαμηλή τάση σε όποια έξοδο έχει “0”.

Με τη συμπλήρωση του 3ου βήματος συμπληρώνεται ένας πλήρης κύκλος λειτουργίας και η διαδικασία αρχίζει από την αρχή. Ο κύκλος λειτουργίας εκτελείται συνεχώς όσο το PLC βρίσκεται σε κατάσταση RUN. Δηλαδή ένα PLC εκτελεί συνεχώς τα βήματα του κύκλου λειτουργίας. Στο παρακάτω σχήμα φαίνεται ένας κύκλος λειτουργίας PLC.



Σχ.2.17Κύκλος λειτουργίας του PLC

Ο χρόνος που χρειάζεται για να εκτελέσει το PLC ένα πλήρη κύκλο λειτουργίας ονομάζεται χρόνος κύκλου και εξαρτάται από την ταχύτητα του επεξεργαστή του PLC, αλλά και από τον αριθμό και το είδος των εντολών του προγράμματος. Δηλαδή στο ίδιο PLC για ένα μεγαλύτερο πρόγραμμα έχουμε μεγαλύτερο χρόνο κύκλου. Ο χρόνος κύκλου αποτελεί

ένα μέτρο σύγκρισης μεταξύ των PLC. Για να μπορούν να συγκριθούν τα PLC ως προς την ταχύτητα εκτέλεσης ενός προγράμματος, ορίζουμε τον μέσο χρόνο κύκλου, σαν το χρόνο κύκλου ενός προγράμματος που περιλαμβάνει 1 Kbyte δυαδικές εντολές. Πάντως στη χειρότερη περίπτωση και σε ένα αργό PLC, ο χρόνος κύκλου δεν ξεπερνά τις μερικές εκατοντάδες millisecond (ms).[17]

2.8 Διαφορές μεταξύ προγραμματιζόμενων λογικών ελεγκτών και ηλεκτρονικών υπολογιστών

Η αρχιτεκτονική ενός προγραμματιζόμενου λογικού ελεγκτή είναι ίδια με αυτή ενός ηλεκτρονικού υπολογιστή. Ένας H/Y μπορεί να μετατραπεί σε έναν προγραμματιζόμενο ελεγκτή. Παρέχοντας του μια δίοδο για να λαμβάνονται πληροφορίες από κάποιες συσκευές, όπως χειροκίνητους διακόπτες και ένα πρόγραμμα για να επεξεργάζεται τις εισόδους και να αποφασίζει τους τρόπους αλλαγής των φορτίων των συσκευών, μπορεί εύκολα να μετατραπεί σε έναν ελεγκτή όπως ο P.L.C. Όμως υπάρχουν κάποια σημαντικά χαρακτηριστικά που διαχωρίζουν τους προγραμματιζόμενους ελεγκτές από τους H/Y.

Ένας H/Y είναι σχεδιασμένος για βέλτιστη απόδοση σε εργασίες υπολογισμών και απεικόνισης. Ένας προγραμματιζόμενος λογικός ελεγκτής είναι σχεδιασμένος για βέλτιστη απόδοση σε εργασίες ελέγχου και ρύθμισης. Αυτή είναι η βασική διαφορά ανάμεσα σε έναν ηλεκτρονικό υπολογιστή και έναν προγραμματιζόμενο λογικό ελεγκτή. Κάποιες ακόμα διαφορές είναι οι εξής:

- σε ένα βιομηχανικό περιβάλλον συμβαίνουν αρκετά συχνά μεγάλες μεταβολές της υγρασίας, της θερμοκρασίας και του ηλεκτρικού θορύβου. Ένας P.L.C. είναι κατασκευασμένο για προσαρμογή και λειτουργία σε τέτοιου είδους περιβάλλον. Ένας καλοσχεδιασμένος προγραμματιζόμενος λογικός ελεγκτής συνήθως δεν επηρεάζεται από τους παραπάνω παράγοντες που υπάρχουν στους περισσότερους βιομηχανικούς χώρους. Αντίθετα, ένας ηλεκτρονικός υπολογιστής είναι υπερβολικά ευαίσθητος σε τέτοιους χώρους.
- ένας προγραμματιζόμενος λογικός ελεγκτής έχει τη δική του γλώσσα προγραμματισμού η οποία είναι αποθηκευμένη στη μόνιμη μνήμη του ελεγκτή. Ένας ηλεκτρονικός υπολογιστής χρειάζεται δικό του λειτουργικό σύστημα το οποίο μάλιστα δεν είναι αποθηκευμένο εξ αρχής στη μνήμη του. Απαιτείται χρόνος εγκατάστασης και θεωρείται αρκετά δύσκολο, ειδικά για κάποιους που δεν έχουν τις απαραίτητες γνώσεις.
- ένας ηλεκτρονικός υπολογιστής πρόκειται για μία σύνθετη υπολογιστική μηχανή ικανή να εκτελεί διάφορα προγράμματα και λειτουργίες ταυτόχρονα ή με οποιαδήποτε σειρά. Από την άλλη μεριά, οι προγραμματιζόμενοι λογικοί ελεγκτές είναι ικανοί να εκτελούν ένα μόνο πρόγραμμα κάθε φορά και αυτό σε ακολουθιακή φορά, από την πρώτη εντολή μέχρι την τελευταία.

2.9 Κύριες λειτουργίες προγραμματιζόμενων λογικών ελεγκτών

Τα PLC σήμερα έχουν και επιπλέον λειτουργίες που βοηθούν στην δημιουργία του αυτοματισμού. Οι λειτουργίες αυτές αυξάνουν συνεχώς καθώς τα PLC εξελίσσονται με ταχύτερους ρυθμούς. Αναφέρονται ενδεικτικά οι σημαντικότερες από αυτές.

- Λειτουργία απαριθμητών. Οι απαριθμητές αποτελούν ακόμα ένα πολύ σημαντικό στοιχείο των PLC. Οι απαριθμητές μπορούν να απαριθμούν εξωτερικούς ή εσωτερικούς παλμούς. Η απαρίθμηση μπορεί να είναι προς τα πάνω (count up) ή προς τα κάτω (count down). Η λειτουργία των απαριθμητών δεν είναι ίδια σε όλα τα PLC.

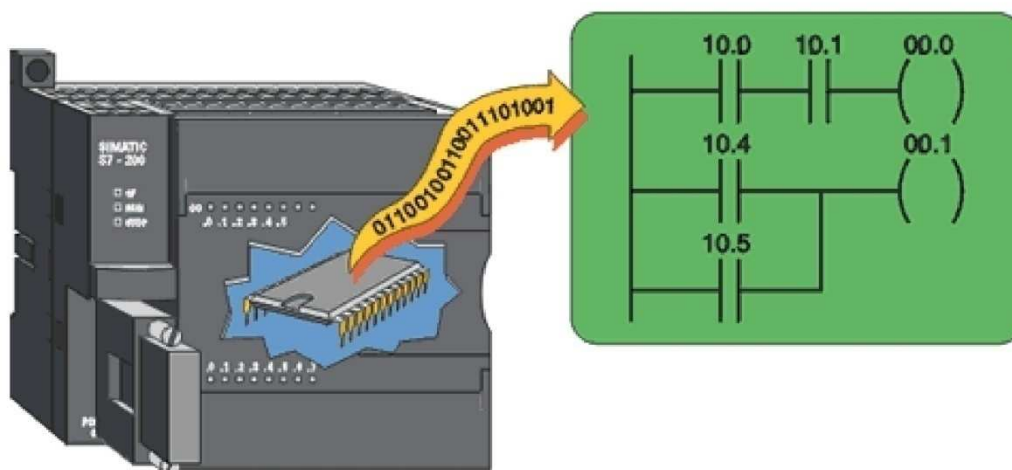
- Δυνατότητα πραγματικού ρολογιού, μέσω του οποίου μπορούμε να προγραμματίσουμε κάποιες εξόδους σε πραγματικό χρόνο, ημερομηνία και ώρα.

- Αριθμητικές επεξεργασίες. Τα σύγχρονα PLC έχουν προσεγγίσει πάρα πολύ τις δυνατότητες των ηλεκτρονικών υπολογιστών. Σχεδόν όλα τα PLC έχουν σήμερα τη δυνατότητα να επεξεργάζονται αριθμητικές πράξεις.

- Αναλογικές εισοδοί-εξοδοί. Τα PLC ενώ αρχικά ήρθαν για να αντικαταστήσουν τους αυτοματισμούς καλωδιωμένης λογικής (αυτοματισμούς με ρελέ), οι δυνατότητές τους έχουν εξαπλωθεί με προοπτική να καλύψουν πλήρως και τα συστήματα αυτομάτου ελέγχου, όπως είναι αναλογικοί έλεγχοι θερμοκρασίας, πίεσης, στάθμης, στροφών κινητήρων κλπ. Αυτό γίνεται δυνατό με την δυνατότητα των PLC να δέχονται και να επεξεργάζονται αναλογικές εισόδους, όπως και να παρέχουν αναλογικές εξόδους. Το PLC μετατρέπει τις αναλογικές τιμές των εισόδων σε ψηφιακές τιμές και στη συνέχεια επεξεργάζεται τις τιμές αυτές αξιοποιώντας τις δυνατότητες για επεξεργασία ψηφιακών αριθμών όπως ήδη προαναφέρθηκε. Η δυνατότητα επεξεργασίας αναλογικών σημάτων έχει δώσει άλλη δυναμική στην εξέλιξη στα PLC.

- Δικτύωση PLC – Συνεργασία μεταξύ τους και με ηλεκτρονικούς υπολογιστές. Η εξέλιξη των PLC σήμερα αλλάζει τη μορφή της βιομηχανίας. Τα PLC μπορούν να συνδέονται μεταξύ τους ανταλλάσσοντας πληροφορίες, όπως και να συνεργάζονται με ηλεκτρονικούς υπολογιστές, οι οποίοι ασχολούνται με τον έλεγχο όλης της παραγωγής και ακόμη με τον έλεγχο της αποθήκης και του λογιστηρίου του εργοστασίου. Όλα αυτά μαζί αποτελούν ένα βασικό Βιομηχανικό Δίκτυο Αυτοματισμού (Computer Automatic Network, CAN).[12]

Κεφάλαιο 3 Προγραμματισμός



Εισαγωγή

Με την έννοια προγραμματισμός ενός P.L.C. εννοείται η διαδικασία της δημιουργίας μιας σειράς εντολών οι οποίες λύνουν ένα συγκεκριμένο πρόβλημα αυτοματισμού. Η υλοποίηση του αντίστοιχου αλγορίθμου επίλυσης του προβλήματος αποτελεί το πρόγραμμα. Πρόγραμμα λοιπόν είναι ένα 'σύνολο από κανόνες ή εντολές' σύμφωνα με τις οποίες συμπεριφέρεται ο προγραμματιζόμενος λογικός ελεγκτής. Οι κανόνες αυτοί καθορίζουν τις λειτουργίες ελέγχου του ελεγκτή. Σκοπός του προγράμματος είναι να καθορίσει τις ακριβείς συνθήκες για την ενεργοποίηση κάθε εξόδου του P.L.C. [18]

Ο προγραμματισμός του P.L.C. μπορεί να γίνει με διάφορους τρόπους. Οι τρόποι αυτοί έχουν καθοριστεί και τυποποιηθεί από το πρότυπο IEC 1131-3 και ονομάζονται γλώσσες προγραμματισμού. [18]

Αρχικά μέχρι και τα μέσα της δεκαετίας του 1980, τα PLC προγραμματίζονταν μέσω προγραμματιστικών πάνελ, τα οποία παρέχονταν από την κατασκευάστρια εταιρία, ή δια μέσω ειδικών προγραμματιστικών τερματικών. Τα τελευταία σε πολλές περιπτώσεις περιείχαν πλήκτρα τα οποία αναπαριστούσαν τα διάφορα λογικά στοιχεία των PLC προγραμμάτων. Τα προγράμματα αποθηκεύονταν σε μια κεφαλή μαγνητικής κασέτας. Λόγο της έλλειψης χωρητικότητας μνήμης δεν υπήρχαν μεγάλες δυνατότητες για εκτύπωση και δημιουργία επεξηγηματικών εγγράφων (documentation). Αυτή η έλλειψη χώρου αποθήκευσης προέκυπε από το γεγονός ότι οι παλιοί υπολογιστές χρησιμοποιούσαν

μαγνητική μνήμη πυρήνα (magnetic core memory), με την ανάπτυξη της τεχνολογίας υπήρξε και αύξηση της χωρητικότητας.[15]

3.1 Προσπέλαση προγράμματος

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

Το λειτουργικό σύστημα 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.[24]

3.1.1 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 προσπελαύνεται 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.[24]

3.2 Δομή προγράμματος

Μπορούμε να χωρίσουμε το πρόγραμμα σε όσα μέρη θέλουμε με σκοπό να το διαβάσουμε 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.

3.2.1 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.

3.2.1.1 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).

• Μπλοκ δεδομένων (DB).

Αυτά τα μπλοκ περιέχουν τα δεδομένα του προγράμματος μας. Προγραμματίζοντας 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 όνομα τους.

3.2.1.2 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.

3.3 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 προγράμματος 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 πρόγραμμα 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.[24]

3.4 Γλώσσες προγραμματισμού

Ο προγραμματισμός των PLC δεν απαιτεί κάποια ανώτερη γλώσσα προγραμματισμού. Οι πιο διαδεδομένες τυποποιημένες μορφές προγραμματισμού που έχουν επικρατήσει διεθνώς είναι:

1. Σχέδιο επαφών (LAD – Ladder Diagram)
2. Λίστα εντολών (IL – Instruction List) και
3. Διάγραμμα λογικών πυλών (FBD – Function Block Diagram).

Αυτό συμβαίνει διότι με τις συγκεκριμένες γλώσσες προγραμματισμού είναι εύκολη η παρακολούθηση της ροής του προγράμματος και συνεπώς ο εντοπισμός κάποιου σφάλματος.

Οι γλώσσες προγραμματισμού, ανάλογα με το είδος των στοιχείων και των εντολών που χρησιμοποιούν, ταξινομούνται σε μη γραφικές και σε γραφικές.[18]

3.4.1 Μη γραφικές

Οι μη γραφικές γλώσσες προγραμματισμού χρησιμοποιούν εντολές αντίστοιχες της γλώσσας μηχανής. Τέτοιες εντολές θα έκαναν τον προγραμματισμό ενός P.L.C. αφενός χρονοβόρο και αφετέρου, εφικτό μόνο από χρήστες με αρκετές γνώσεις στη δομή και λειτουργία μικροεπεξεργαστών. Παρ' όλα αυτά όμως χρησιμοποιούνται, αν και δεν είναι τόσο διαδεδομένες.

3.4.1.1 Λίστα εντολών Instruction List (IL) ή (Statement List, STL) ή λογικών εντολών

Η λίστα εντολών πρόκειται για μία μη γραφική γλώσσα προγραμματισμού με τη μορφή κειμένου. Η γλώσσα αυτή αναπτύχθηκε σχεδόν ταυτόχρονα με τη γλώσσα LADDER, αν και οι εταιρείες έδειξαν στην αρχή δισταγμό στο να την «προωθήσουν», φοβούμενες μην τρομάξουν το τεχνικό κατεστημένο της βιομηχανίας. Έχει την ικανότητα βέλτιστης χρήσης της μνήμης και εκτέλεσης του προγράμματος. Αποτελείται από μία σειρά εντολών, οι οποίες αντιστοιχούν στις λογικές πύλες (AND, OR, NOT κ.λπ.), τοποθετημένες η μία κάτω από την άλλη σε γραμμές, οι οποίες καταχωρούνται με συντομογραφικό τρόπο. Κάθε γραμμή του προγράμματος περιέχει δύο στοιχεία. Το ένα στοιχείο είναι η εντολή. Αυτή αναγράφεται με λατινικούς χαρακτήρες και θα είναι είτε εντολή φόρτωσης, Load (LD), είτε κάποια εντολή από τις εκφράσεις της άλγεβρας του Boole. Μπορεί επίσης να είναι μια έτοιμη ρουτίνα του ελεγκτή, όπως κάποιος χρονιστής ή κάτι άλλο. Το άλλο στοιχείο είναι η μεταβλητή, η οποία συνήθως περιέχει την διεύθυνση κάποιας εισόδου ή εξόδου. Κάποιοι κατασκευαστές

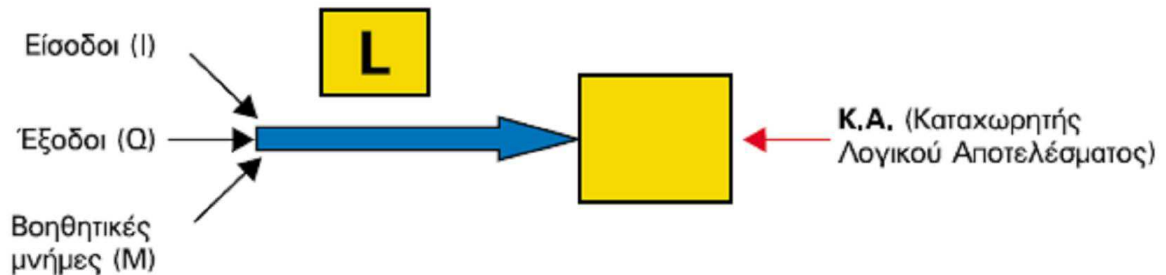
προσθέτουν και ένα τρίτο στοιχείο σε κάθε γραμμή, τον αριθμό της γραμμής, ο οποίος τοποθετείται αυτόματα κάθε φορά που επιλέγεται αλλαγή γραμμής .

Στην αρχή η γλώσσα λίστα εντολών ήταν πολύ φτωχή και περιοριζόταν μόνο στις βασικές λογικές εντολές, οι οποίες αντιστοιχούσαν αμέσως στις γραφικές εντολές της γλώσσας LADDER. Σήμερα οι γλώσσες αυτές έχουν εξελιχθεί πάρα πολύ και συναντά κανείς σε αυτές στοιχεία από τις γλώσσες των υπολογιστών και κυρίως των γλωσσών Assembly. Ο προγραμματισμός σε λίστα εντολών απαιτεί από τον ηλεκτρολόγο να έχει έστω στοιχειώδεις γνώσεις προγραμματισμού.[18],[20]

AND	A
OR	O
NOT	N
Exclusive - OR	XO
Assignment	=
Set	S
Reset	R
Count (forwards)	ZV
Count (backwards)	ZR
Add	ADD
Subtract	SUB
Multiply	MUL
Divide	DIV
Greater than	GR
Greater than or equal to	GRG
Equal to	GL
Smaller than	SL
Smaller than or equal to	SLG
Convert code (decimal / binary)	DEB
Convert code (binary / decimal)	BID
Non operation	NOP
Load	L
Brackets open	(
Brackets closed)
Jump (unconditional)	SP
JUMP Module call (unconditional)	B
Module call (conditional)	BAB
Call module end	BE
Program end	PE
Comment (start/end)	PE
Constants	K
Input	I
Output	O
Flag	F
Timer	T
Counter	C
Βασικά στοιχεία προγραμματισμού γλώσσας STL. [34]	

- Η εντολή L (Load)

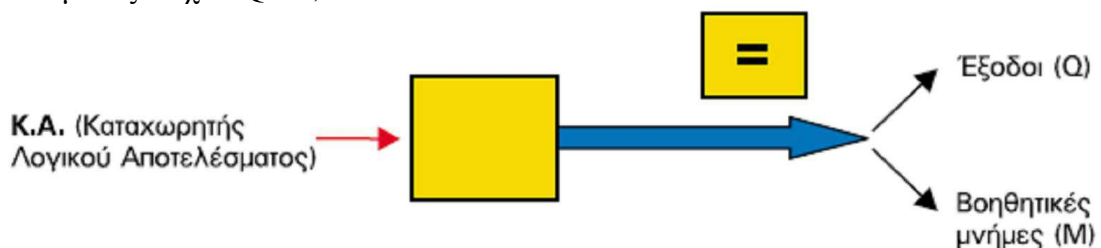
Το πρόγραμμα το οποίο αντιστοιχεί σε μια πύλη λογικού κυκλώματος (ή κλάδο ηλεκτρολογικού κυκλώματος αυτοματισμού) ξεκινά με την εντολή L (Load, φόρτωση). Το PLC με την εντολή Load διαβάζει τη λογική κατάσταση (0 ή 1) μιας εισόδου, εξόδου, βοηθητικής μνήμης, χρονικού κ.λπ., και τη φορτώνει σε ένα καταχωρητή (μια ειδική θέση μνήμης) τον οποίο θα ονομάζουμε Καταχωρητή Λογικού Αποτελέσματος (Κ.Α.). Η εντολή L μπορεί να αναφέρεται σε εισόδους, εξόδους, βοηθητικές μνήμες, χρονικά, κ.λπ. Π.χ. L I 0.1, L Q 0.2, L M 0.1.



Σχ.3.1 Σχηματική παράσταση της εντολής Load.

- Η εντολή = (ίσον)

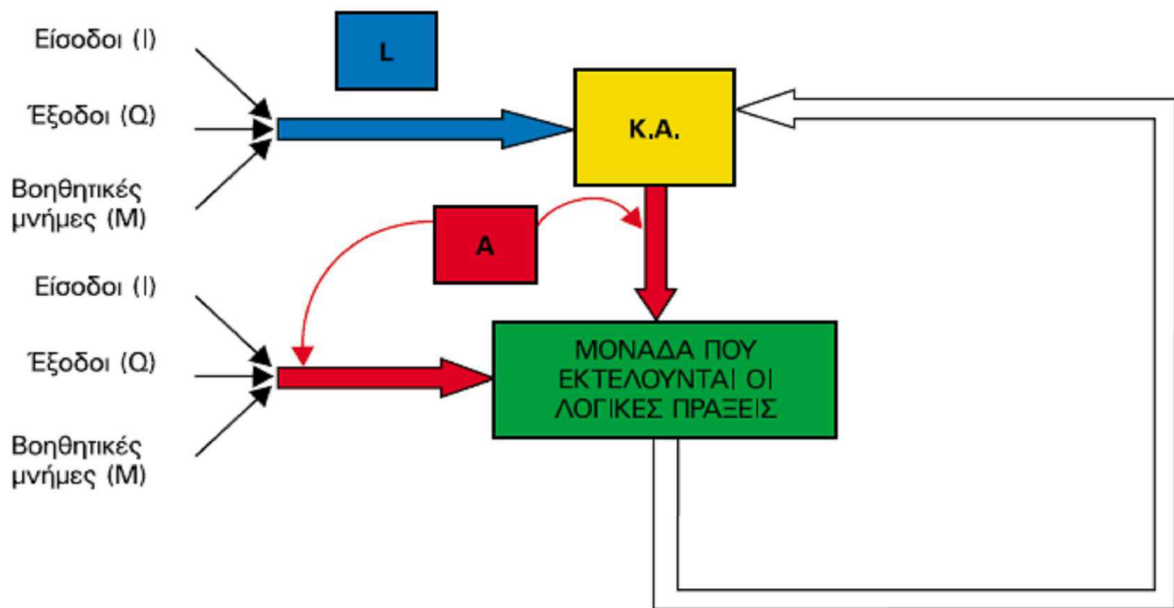
Το πρόγραμμα που αντιστοιχεί σε μια πύλη λογικού κυκλώματος (ή κλάδο ηλεκτρολογικού κυκλώματος αυτοματισμού) καταλήγει πάντα με την εντολή = (ίσον). Η εντολή αναφέρεται σε εξόδους και βοηθητικές μνήμες. Το PLC με την εντολή = μεταφέρει στις εξόδους ή στις βοηθητικές μνήμες το περιεχόμενο του καταχωρητή λογικού αποτελέσματος. Π.χ. = Q 1.2, = M 0.1



Σχ.3.2 Σχηματική παράσταση της εντολής =

- Η εντολή A (AND)

Η εντολή A υπαγορεύει στο PLC να εκτελέσει τη λογική πράξη AND. Η εντολή AND αναφέρεται σε εισόδους, εξόδους, βοηθητικές μνήμες, χρονικά κ.λπ. Η λογική πράξη γίνεται μεταξύ της λογικής κατάστασης της εισόδου, εξόδου, βοηθητικής μνήμης, χρονικού, κ.λπ. το οποίο αναφέρεται στην εντολή και του περιεχομένου του Κ.Α. Το αποτέλεσμα επιστρέφει στον Κ.Α. Προσέξτε το παρακάτω σχήμα, το οποίο αποδίδει πολύ καλά τον τρόπο που το PLC εκτελεί τη λογική πράξη AND.



Σχ.3.3 Σχηματική παράσταση της εντολής AND.

Παράδειγμα: Έστω ότι έχουμε δύο διακόπτες S1 και S2 (επαφές σε ηρεμία ανοικτές), συνδεδεμένους σε σειρά και θέλουμε να ανάβει η λυχνία L1 όταν είναι κλειστοί και οι δύο διακόπτες ταυτόχρονα. Όπως έχουμε αναφέρει η ύπαρξη τάσης σε μια είσοδο-έξοδο του PLC μεταφράζεται σε σήμα "1". Έτσι όταν ρωτάμε για την ύπαρξη σήματος "1" η εντολή στο πρόγραμμα είναι η A I0.0. Για να φορτώσουμε την εντολή στην είσοδο που έχουμε επιλέξει χρησιμοποιούμε την εντολή LD I0.0. Έτσι όπως θα δούμε και παρακάτω οι εντολές LD I0.0 και LD I0.1 φορτώνουν τις εντολές (A) στις εισόδους I0.0 και I0.1. Η εντολή A I0.0 ρωτάει αν και στην είσοδο I0.0 έχουμε σήμα "1" και η εντολή A I0.1 ρωτάει αν και στην είσοδο I0.1 έχουμε σήμα "1". Αν ισχύουν οι δύο αυτές συνθήκες τότε ενεργοποιείται η έξοδος Q0.0 που έχουμε ορίσει με την εντολή = Q0.0 που αντιστοιχεί στην λυχνία L1. Παρακάτω ακολουθεί το πρόγραμμα και ο πίνακας που μας δείχνει τις πιθανές καταστάσεις και αντίστοιχα πώς συμπεριφέρεται η έξοδος.

```

LD I0.0
LD I0.1
A I0.0      ή      LD I0.0
A I0.1      = Q0.0
= Q0.0
END
END

```

I0.0	I0.1	Q0.0
0	0	0
0	1	0
1	0	0
1	1	1

Όπως παρατηρούμε η μοναδική περίπτωση που ανάβει η λυχνία εξόδου είναι όταν και οι δύο διακόπτες είναι κλειστοί δηλαδή έχουν σήμα "1". Σε κάθε άλλη περίπτωση η έξοδος δεν ενεργοποιείται.

- Η εντολή AN (AND NOT)

Η εντολή AN υπαγορεύει στο PLC να εκτελέσει τη λογική πράξη AND NOT, η οποία είναι στην ουσία το συμπλήρωμα της εντολής AND. Η εντολή AND NOT αναφέρεται σε εισόδους, εξόδους, βοηθητικές μνήμες, χρονικά κ.λπ. Η λογική πράξη γίνεται μεταξύ της λογικής κατάστασης της εισόδου, εξόδου, βοηθητικής μνήμης, χρονικού, κ.λπ. το οποίο αναφέρεται στην εντολή και του περιεχομένου του Κ.Α. Το αποτέλεσμα επιστρέφει στον Κ.Α.

Παράδειγμα: Έστω ότι έχουμε δύο διακόπτες S1 και S2 (επαφές σε ηρεμία ανοικτές), συνδεδεμένους σε σειρά και θέλουμε να ανάβει η λυχνία L1 όταν είναι ανοιχτοί και οι δύο διακόπτες ταυτόχρονα. Σε αυτήν την περίπτωση η ερώτηση για σήμα μηδέν σε μια είσοδο-έξοδο του PLC γίνεται με την εντολή AN (AND NOT). Για να φορτώσουμε την εντολή αυτή στην είσοδο που έχουμε επιλέξει χρησιμοποιούμε την εντολή LDN (Load Not). Έτσι στο παρακάτω παράδειγμα οι εντολές LDN I0.0 και LDN I0.1 φορτώνουν τις εντολές AN I0.0 και AN I0.1 αντίστοιχα στους διακόπτες που έχουμε επιλέξει. Έτσι λοιπόν η εντολή AN I0.0 ρωτάει αν και στην είσοδο I0.0 έχουμε σήμα "0" και η εντολή AN I0.1 ρωτάει αν και στην είσοδο I0.1 έχουμε σήμα "0". Αν ισχύουν οι δύο αυτές συνθήκες τότε ενεργοποιείται η έξοδος Q0.0 που έχουμε ορίσει με την εντολή = Q0.0 που αντιστοιχεί στην λυχνία L1. Παρακάτω ακολουθεί το πρόγραμμα και ο πίνακας που μας δείχνει τις πιθανές καταστάσεις και αντίστοιχα πώς συμπεριφέρεται η έξοδος.

LDN I0.0		LDN I0.0
LDN I0.1		AN I0.1
AN I0.0	ή	= Q0.0
AN I0.1		END
= Q0.0		
END		

- Η εντολή O (OR)

Η εντολή OR υπαγορεύει στο PLC να εκτελέσει τη λογική πράξη OR. Η εντολή OR αναφέρεται σε εισόδους, εξόδους, βοηθητικές μνήμες, χρονικά, κ.λπ. Εκτελείται με ανάλογο τρόπο, με αυτόν που εκτελείται η εντολή AND.

Παράδειγμα: Έστω ότι έχουμε δύο διακόπτες S1 και S2 (επαφές σε ηρεμία ανοικτές), και θέλουμε να ανάβει η λυχνία L1 όταν πατηθεί τουλάχιστον ένας ή και οι δύο διακόπτες ταυτόχρονα. Σε αυτήν την περίπτωση η ερώτηση για σήμα "1" σε μια είσοδο-έξοδο του PLC γίνεται με την εντολή O (OR). Για να φορτώσουμε την εντολή αυτή στην είσοδο που έχουμε επιλέξει χρησιμοποιούμε την εντολή LD (Load). Έτσι στο παρακάτω παράδειγμα οι εντολές LD I0.0 και LD I0.1 φορτώνουν τις εντολές O I0.0 και O I0.1 αντίστοιχα στους διακόπτες που έχουμε επιλέξει. Έτσι λοιπόν η εντολή O I0.0 ρωτάει είτε στην είσοδο I0.0 έχουμε σήμα "1" και η εντολή O I0.1 ρωτάει είτε στην είσοδο I0.1 έχουμε σήμα "1". Δηλαδή αν ένας από τους δύο διακόπτες είναι κλειστός ή και οι δύο είναι κλειστοί τότε ενεργοποιείται η έξοδος Q0.0 που έχουμε ορίσει με την εντολή = Q0.0 που αντιστοιχεί στην λυχνία L1. Παρακάτω ακολουθεί το πρόγραμμα και ο πίνακας που μας δείχνει τις πιθανές καταστάσεις και αντίστοιχα πώς συμπεριφέρεται η έξοδος.

LD I0.0		LD I0.0
LD I0.1		O I0.1
O I0.0	ή	= Q0.0
O I0.1		END
= Q0.0		
END		

- Η εντολή ON (OR NOT)

Η εντολή OR NOT υπαγορεύει στο PLC να εκτελέσει τη λογική πράξη OR NOT, η οποία είναι το συμπλήρωμα της εντολής OR. Η εντολή OR NOT αναφέρεται σε εισόδους, εξόδους, βοηθητικές μνήμες, χρονικά, κ.λπ. Εκτελείται με ανάλογο τρόπο, με αυτόν που εκτελείται η εντολή AND.

Παράδειγμα: Έστω ότι έχουμε δύο διακόπτες S1 και S2 (επαφές σε ηρεμία ανοικτές), και θέλουμε να ανάβει η λυχνία L1 είτε όταν και οι δύο διακόπτες είναι ανοιχτοί είτε όταν ένας από τους δύο είναι ανοιχτός. Σε αυτήν την περίπτωση η ερώτηση για σήμα "0" σε μια είσοδο-έξοδο του PLC γίνεται με την εντολή ON (OR NOT). Για να φορτώσουμε την εντολή αυτή στην είσοδο που έχουμε επιλέξει χρησιμοποιούμε την εντολή LDN (Load Not). Έτσι στο παρακάτω παράδειγμα οι εντολές LDN I0.0 και LDN I0.1 φορτώνουν τις εντολές ON I0.0 και ON I0.1 αντίστοιχα στους διακόπτες που έχουμε επιλέξει. Έτσι λοιπόν η εντολή ON I0.0 ρωτάει είτε στην είσοδο I0.0 έχουμε σήμα "0" και η εντολή ON I0.1 ρωτάει είτε στην είσοδο I0.1 έχουμε σήμα "0". Αν ισχύουν οι δύο αυτές συνθήκες τότε ενεργοποιείται η έξοδος Q0.0 που έχουμε ορίσει με την εντολή = Q0.0 που αντιστοιχεί στην λυχνία L1. Παρακάτω ακολουθεί το πρόγραμμα και ο πίνακας που μας δείχνει τις πιθανές καταστάσεις και αντίστοιχα πώς συμπεριφέρεται η έξοδος.

LDN I0.0		LDN I0.0
LDN I0.1		ON I0.1
ON I0.0	ή	= Q0.0
ON I0.1		END
= Q0.0		
END		

- Η εντολή N (NOT)

Η εντολή NOT κάνει αλλαγή στην κατάσταση των εισόδων ή των εξόδων. Δηλαδή αν το σήμα μιας ή παραπάνω εισόδων-εξόδων είναι "1" τότε το κάνει "0" και αντίστροφα. Έτσι λοιπόν στο παράδειγμα που ακολουθεί αν βάλουμε ένα NOT πριν την εντολή εξόδου τότε θα έχουμε τα αντίστροφα αποτελέσματα σύμφωνα με τον πίνακα που ακολουθεί.

LD I0.0		LD I0.0
LD I0.1		A I0.1
A I0.0	ή	NOT
A I0.1		= Q0.0
NOT		END
= Q0.0		
END		

LD I0.0 φόρτωση της εισόδου I0.0 (normally open)
 OR Q0.0 τοποθέτηση μιας ανοιχτής επαφής της εξόδου παράλληλα με την είσοδο I0.0
 LDN I0.1 φόρτωση της εισόδου I0.1 (normally closed)
 AN I0.2 τοποθέτηση της επαφής I0.2 (normally closed) σειρά με την I0.1
 A LD τοποθέτηση των δύο παραπάνω κλάδων σε σειρά
 =Q0.0 έξοδος
 I0.0: START I0.1: STOP I0.2: θερμικό Q0.0: έξοδος (ρελέ)
 Παράδειγμα κώδικα σε γλώσσα STL, απλός αυτόματος διακόπτης.

3.4.1.1.1 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.

3.4.1.1.2 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.[24]

3.4.1.2 Γλώσσα δομημένου κειμένου Structured Text (ST)

Πρόκειται για μία γλώσσα προγραμματισμού η οποία έχει πολλές ομοιότητες με τη γλώσσα C και ως προς την δομή, αλλά και ως προς τις εντολές. Η γλώσσα δομημένου κειμένου απαιτεί ειδικές γνώσεις προγραμματισμού και για αυτό τον λόγο τείνει να εξαφανιστεί, μιας και δεν χρησιμοποιείται πολύ.[18]

3.4.1.3 Γλώσσα MATRIX

Γλώσσα MATRIX (είναι μια γλώσσα προγραμματισμού που έχει κατασκευαστεί από την FESTO, φιλική στο χρήστη, ειδικά για ηλεκτροπνευματικά και ηλεκτροϋδραυλικά συστήματα).[20]

Ο προγραμματισμός του P.L.C. με την γλώσσα MATRIX είναι απλούστερος, σε σχέση με τις υπόλοιπες γλώσσες που αναφέραμε. Στη περίπτωση αυτή οι είσοδοι που κάθε φορά ενεργοποιούνται δηλώνονται με το ψηφίο L αυτές που δεν ενεργοποιούνται με το 0 και αυτές

που δεν χρησιμοποιούνται καθόλου με το X . Τα ίδια ισχύουν και για τις εξόδους . Το πρόγραμμα χωρίζεται σε βήματα και κάθε βήμα μπορεί να επαναληφθεί πολλές φορές πριν γίνει το επόμενο (Counter) . Επίσης μπορεί να καθοριστεί η χρονική διάρκεια (Timer) μετά από την οποία το πρόγραμμα θα προχωρήσει στο επόμενο ή κάποιο άλλο βήμα . Στο πρόγραμμα υπάρχει η δυνατότητα ορισμού των εξόδων που θα απενεργοποιηθούν σε περίπτωση λανθασμένου σήματος εισόδου .[25]

Δημιουργία προγράμματος MATRIX

Αφού επεξεργαστούμε προσεκτικά την διαδικασία ελέγχου που διαχειρίζεται του πρόγραμμα του P.L.C. ακολουθούμε τα παρακάτω βήματα :

- 1) Αριθμούμε τα βήματα στα οποία ολοκληρώνεται η διεργασία .
- 2) Αριθμούμε τις εισόδους (inputs) με όποια σειρά θέλουμε .
- 3) Αριθμούμε τις εξόδους (outputs) που θα χρησιμοποιήσουμε .
- 4) Ορίζουμε ποιες εισόδους δεν χρησιμοποιούμε.
- 5) Προσδιορίζουμε τις συνθήκες ελέγχου.
- 6) Καθορίζουμε το emergency stop.

Σε κάθε βήμα σημειώνουμε με κάποιο κύκλο , τις εισόδους που δέχονται τα σήματα καθώς και τις εξόδους που ενεργοποιούνται . Ολοκληρώνοντας τα βήματα έχουμε μια συνοπτική εικόνα και το μόνο που απομένει είναι η βηματική εισαγωγή της κατάστασης των εισόδων - εξόδων στον ελεγκτή.[25]

3.4.2 Γραφικές

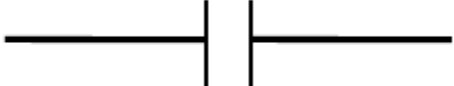

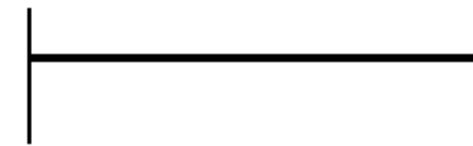
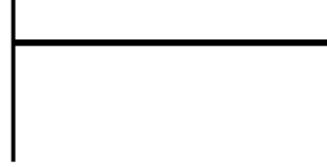
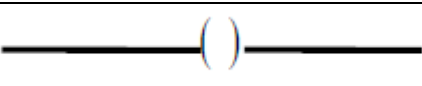
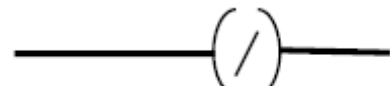
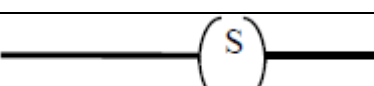
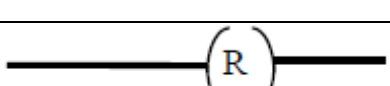
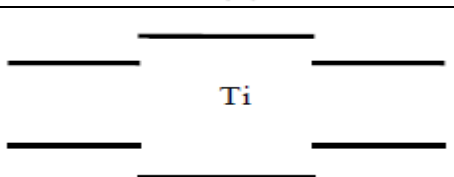
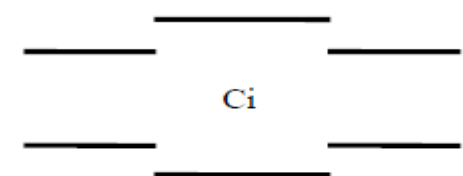
Οι γραφικές γλώσσες προγραμματισμού χρησιμοποιούν εντολές και στοιχεία αρκετά όμοια με τα σύμβολα που χρησιμοποιούνται στα συμβατικά, κλασικά συστήματα αυτοματισμού καθώς επίσης και σύμβολα λογικών πυλών. Είναι πιο προσιτές και πλεονεκτούν ως προς την καλύτερη και ευκολότερη εποπτεία του προγράμματος.[18]

3.4.2.1 Σχέδιο επαφών Ladder Diagram(LAD)ή γλώσσα ηλεκτρολογικών γραφικών

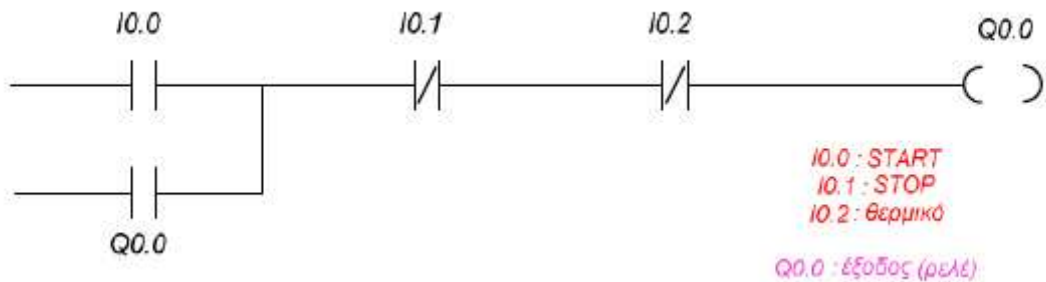
Είναι η πρώτη γλώσσα που αναπτύχθηκε ιστορικά. Το σχέδιο επαφών, ή αλλιώς Ladder Diagram, είναι μία γραφική γλώσσα προγραμματισμού η οποία θυμίζει το διάγραμμα κυκλώματος κλασικού αυτοματισμού, στην ουσία επιτρέπει τη μεταφορά του ηλεκτρολογικού σχεδίου, μέσω της συσκευής προγραμματισμού στο PLC. Σε αυτήν την γλώσσα παρακολουθείται εύκολα η ροή του σήματος και η εκπαίδευση των τεχνικών που ήταν συνηθισμένοι στον κλασικό αυτοματισμό, γίνεται εύκολα και γρήγορα, αφού δεν αλλάζει ουσιαστικά την εργασία σχεδιασμού του αυτοματισμού για αυτό και προτιμάται από αυτούς που είναι εξοικειωμένοι με το ηλεκτρολογικό σχέδιο. Αυτός είναι και ο λόγος που είναι η πιο ευρέως χρησιμοποιούμενη γλώσσα για τον προγραμματισμό τέτοιου είδους ελεγκτών.

Με τη χρήση γραφικών εργαλείων δομείται ένα λογικό πρόγραμμα, ικανό να ακολουθήσει τη λογική της συνδεσμολογίας ενός κυκλώματος κλασικού, συμβατικού αυτοματισμού. Οι δυνατότητες της βέβαια είναι πολύ περισσότερες καθώς εκτελούνται λειτουργίες σύγκρισης, μεταφοράς και μαθηματικής επεξεργασίας δεδομένων. Όλοι οι είσοδοι εισάγονται με τη μορφή συμβολικών επαφών και όλοι οι έξοδοι εισάγονται με τη μορφή συμβολικών πηνίων. Στο σχεδιασμό επαφών έχει επικρατήσει η Αμερικάνικη τυποποίηση. Αυτό ίσως οφείλεται στο γεγονός ότι τα πρώτα PLC αναπτύχθηκαν στην Αμερική. Όμως στη συνέχεια ο τρόπος αυτός σχεδιασμού «βόλεψε» και έτσι διατηρήθηκε και από τις Ευρωπαϊκές εταιρείες, με αποτέλεσμα σήμερα να είναι πλέον καθιερωμένος.[18],[20]

Τα δομικά στοιχεία του προγράμματος φαίνονται παρακάτω :

1 . κανονικά ανοικτή επαφή : NO	
2. κανονικά κλειστή επαφή : NC	
3. οριζόντια σύνδεση (συνδέει τα στοιχεία του προγράμματος σε σειρά)	
4. κάθετη σύνδεση (συνδέει τα στοιχεία του προγράμματος παράλληλα) Μερικοί κατασκευαστές χρησιμοποιούν την τεχνική των κόμβων αντί των οριζοντίων και κάθετων συνδέσεων .	
5. άμεση έξοδος που ενεργοποιείται όταν περνάει ρεύμα	
6. αντίστροφη έξοδος (ενεργοποιείται όταν δεν περνάει ρεύμα)	
7. έξοδος SET (είναι συνεχώς ενεργοποιημένη όταν περάσει μια φορά ρεύμα)	
8. έξοδος RESET (είναι συνεχώς απενεργοποιημένη)	
9. χρονικά στοιχεία (στοιχεία με τα οποία πετυχαίνουμε χρονικές καθυστερήσεις)	
10. μετρητές UP - DOWN (είναι στοιχεία με τα οποία μπορούμε να κάνουμε απαρίθμηση)	
11. Drum controller	Είναι ένα λειτουργικό μπλοκ για αυτοματισμούς που εκτελούνται κατά βήματα . Ο drum controller προχωρά κατά ένα βήμα όταν εκπληρωθεί κάποια συνθήκη τέλος άλλου βήματος , τέλος χρόνου , άνοιγμα ή κλείσιμο κάποιας επαφής . Σε κάθε βήμα ενεργοποιούνται οι έξοδοι που θέλουμε.

Σχ.3.4Δομικά στοιχεία του προγράμματος. [25]



Σχ.3.5 Πρόγραμμα σε γλώσσα Ladder, απλός αυτόματος διακόπτης[18]

Κάθε οριζόντια γραμμή, αποτελούμενη από τουλάχιστον μία επαφή, ονομάζεται rung (βαθμίδα). Ο ελεγκτής εκτελεί κάθε rung ξεχωριστά και με τη σειρά που είναι τοποθετημένα στο πρόγραμμα, από πάνω προς τα κάτω. Οι εντολές που υπάρχουν σε κάθε rung εκτελούνται πάντα από τα αριστερά προς τα δεξιά και ποτέ το ανάποδο.

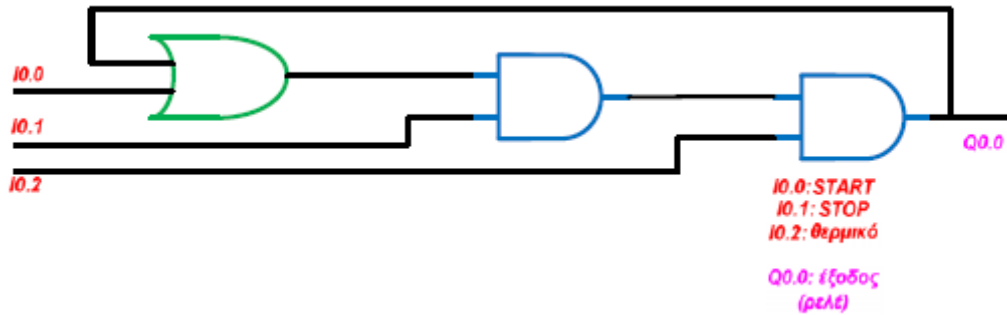
Αρκετοί P.L.C. έχουν την δυνατότητα, με ειδικό λογισμικό, να μετατρέπουν κάποιο πρόγραμμα που είναι γραμμένο σε λίστα εντολών, σε γλώσσα Ladder ή και το ανάποδο. Μάλιστα, η μετατροπή αυτή μπορεί να υλοποιηθεί όχι μόνο σε ολόκληρο το πρόγραμμα, αλλά και σε μεμονωμένα τμήματα εντολών.[18]

3.4.2.2 Διάγραμμα λογικών πυλών Function Block Diagram (FBD) ή CONTROL SYSTEM FLOWCHART (CSF)

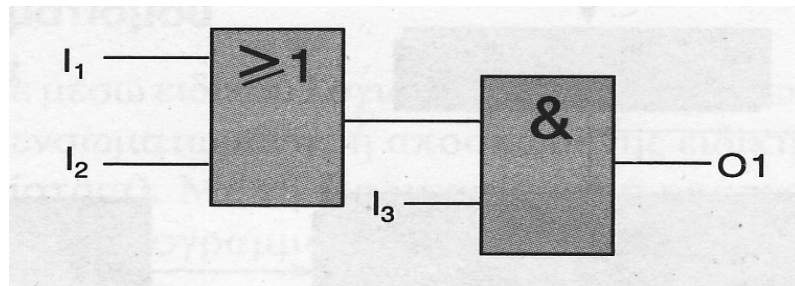
Ο κώδικας της συγκεκριμένης γλώσσας προγραμματισμού περιέχει εντολές οι οποίες αναπαρίστανται με blocks λογικών πυλών. Τα διαγράμματα λογικών πυλών, Function block diagrams, θυμίζουν περισσότερο τη λογική των ψηφιακών ηλεκτρονικών, για αυτό και σαν γλώσσα προγραμματισμού προτιμάται περισσότερο από εκείνους οι οποίοι είναι εξοικειωμένοι με την ψηφιακή σχεδίαση. Η γλώσσα αυτή είναι νεότερη και δεν χρησιμοποιείται από όλες τις εταιρείες.[18],[20]

Αρχή (ένδειξη έναρξης)	
Βήμα έναρξης	
Βήμα	
Συνθήκη συνέχισης κύκλου	
Συνεχώς ενεργοποιημένες γραμμές	
Έξοδος συνθήκης	
Κατευθύνσεις σήματος	

Σχ.3.6 Βασικά στοιχεία προγραμματισμού γλώσσας FBD[25]



Σχ.3.7 Απλός αυτόματος διακόπτης σε γλώσσα FBD.[18]



Σχ.3.8 Πρόγραμμα σε γλώσσα FBD

3.4.2.3 Διάγραμμα διαδοχικών λειτουργιών Sequential Function Chart (SFC)

Πρόκειται για μία γλώσσα προγραμματισμού η οποία αποτελεί εξέλιξη της γλώσσας Grafset. Η γλώσσα προγραμματισμού των διαδοχικών λειτουργιών, Sequential Function Chart, χρησιμοποιεί ένα λειτουργικό διάγραμμα για να περιγράψει γραφικά τη συμπεριφορά ενός συστήματος αυτοματισμού. Ουσιαστικά, αντιπροσωπεύει τη λειτουργία μιας διαδοχικής αυτοματοποιημένης διεργασίας σε μία δομημένη και γραφική μορφή. Ένα πρόγραμμα βασισμένο σε γλώσσα SFC αποτελείται από ένα σύνολο διαδοχικών ενεργειών και βημάτων τα οποία επιτρέπουν σε μια σύνθετη δράση να χωριστεί σε μικρότερες και πιο απλές καθορίζοντας τη σειρά εκτέλεσης. Ταυτόχρονα, το διάγραμμα διαδοχικών λειτουργιών καθορίζεται από κανόνες προγραμματισμού και γραφικά στοιχεία. Πρέπει να σημειωθεί ότι, αν και επίσημα, η γλώσσα προγραμματισμού SFC είναι αναγνωρισμένη σαν μία από τις γλώσσες που υπακούουν στις προδιαγραφές τυποποίησης του προτύπου IEC 1131-3, για τη χρήση της όμως είναι απαραίτητη η εισαγωγή εντολών και δεύτερης γλώσσας. Αυτό κάνει τα διαγράμματα διαδοχικών λειτουργιών να θεωρούνται από πολλούς προγραμματιστές περισσότερο σαν μια μέθοδος διευκόλυνσης στην κατανόηση και ανάλυση πολύπλοκων εφαρμογών παρά σαν μια αυτόνομη γλώσσα προγραμματισμού.

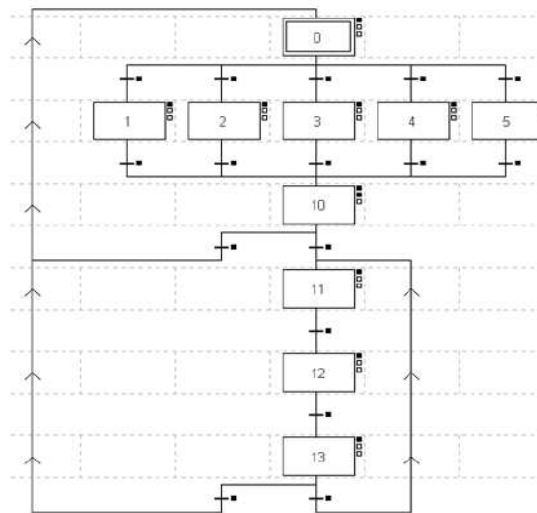
Κάθε πρόγραμμα το οποίο περιέχει συναρτήσεις και εντολές της γλώσσας SFC, αποτελείται από βήματα και μεταβάσεις. Τα βήματα είναι τοποθετημένα διαδοχικά και ελέγχονται από τις μεταβάσεις. Μόλις ενεργοποιηθεί κάποιο βήμα, αμέσως εκτελείται η συνθήκη της αντίστοιχης μετάβασης. Η ενεργοποίηση του επόμενου βήματος συμβαίνει μόνο με το τέλος της μετάβασης. Με αυτόν τον τρόπο ακολουθείται μια διαδοχική ενεργοποίηση όλων των βημάτων του προγράμματος και ταυτόχρονα και των αντίστοιχων συνθηκών. Κατά την διάρκεια μετάβασης από το ένα βήμα στο επόμενο μπορούν να ελεγχθούν είσοδοι, έξοδοι, χρονικά και πολλές άλλες λειτουργίες.

Ένα πρόγραμμα γραμμένο σε κώδικα SFC φαίνεται στο παρακάτω σχήμα. Κάθε βήμα μπορεί να ελέγχει διάφορες ενέργειες, όπως το άνοιγμα ή το κλείσιμο επαφών. Τα βήματα του συγκεκριμένου παραδείγματος ελέγχουν τα φώτα ενός τρίχρωμου φωτεινού σηματοδότη. Για να ανάψει ένας λαμπτήρας, πρέπει το αντίστοιχο βήμα να είναι ενεργό. Το βήμα με το

διπλό τετράγωνο κουτί αποτελεί το πρώτο προς εκτέλεση βήμα αμέσως μετά την έναρξη του προγράμματος ή αμέσως μετά την επανάκτηση της τροφοδοσίας, για αυτό και λέγεται αρχικό βήμα (1). Τα υπόλοιπα βήματα αποκαλούνται απλά βήματα (2) καθώς αποτελούν την συνέχεια του προγράμματος.[18]



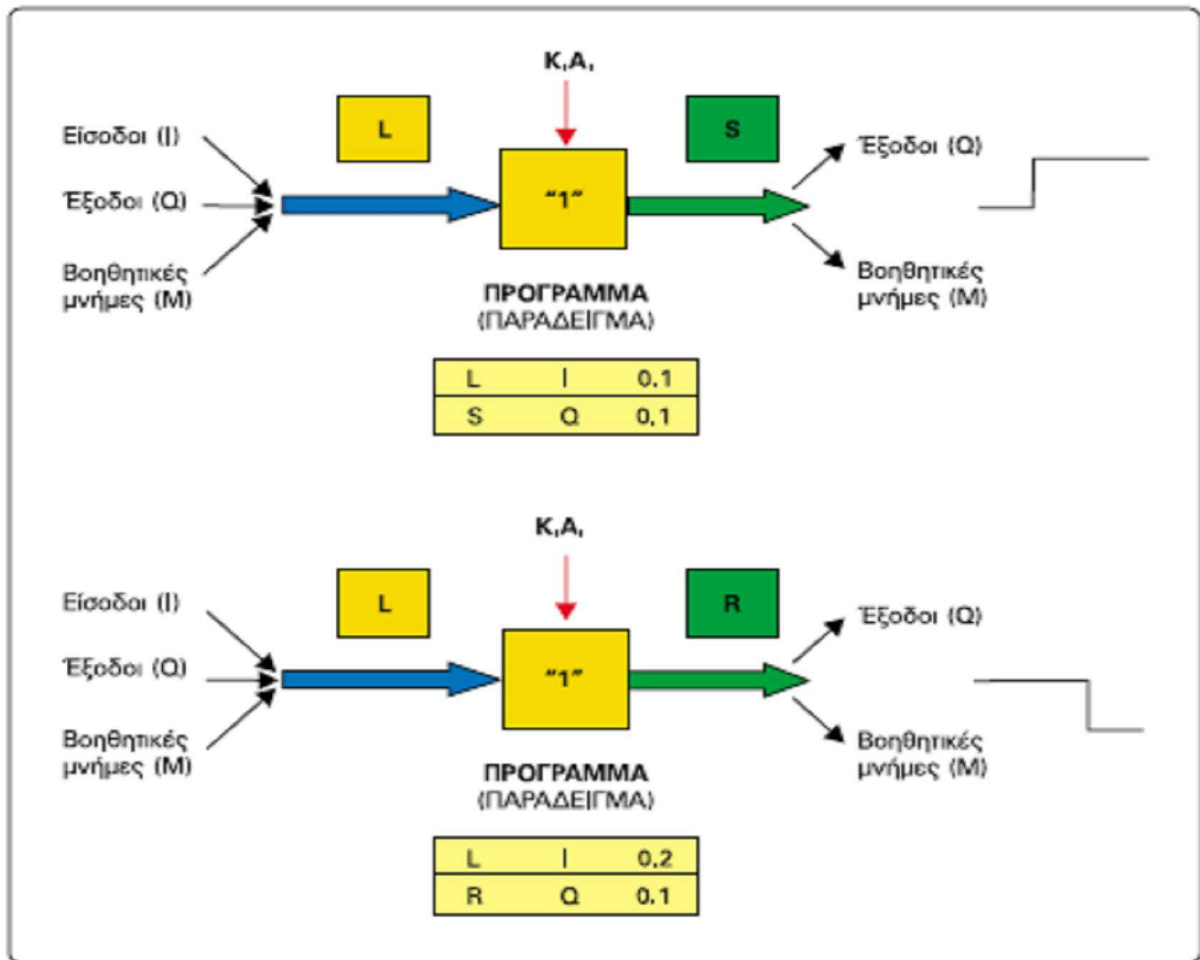
Σχ.3.9 Έλεγχος τρίχρωμου φωτεινού σηματοδότη σε γλώσσα SFC.[18]



Σχ.3.10 Πρόγραμμα σε γλώσσα SFC

3.5 Οι εντολές S (SET) και R (RESET).

Σε όλα τα PLC υπάρχουν οι εντολές S (SET) και R (RESET), οι οποίες αντιστοιχούν στο S-R flip - flop. Αναφέρονται όπως και η εντολή (=) σε εξόδους και βοηθητικές μνήμες. Η εντολή SET θέτει την έξοδο ή τη βοηθητική μνήμη, στην οποία αναφέρεται σε κατάσταση 1, όταν στον Κ.Α. υπάρχει λογική τιμή 1 κατά την εκτέλεση της εντολής. Η έξοδος ή η βοηθητική μνήμη διατηρεί την κατάσταση 1 έστω και αν σε επόμενη εκτέλεση της εντολής SET στον Κ.Α. υπάρχει λογική τιμή 0. Η εντολή RESET θέτει την έξοδο ή τη βοηθητική μνήμη στην οποία αναφέρεται σε κατάσταση 0 όταν στον Κ.Α. υπάρχει λογική τιμή 1 κατά την εκτέλεση της εντολής.



Σχ.3.11 Σχηματική παράσταση των εντολών SET και RESET.

Για να γίνει ξεκάθαρη η διαφορά της εντολής SET από την εντολή (=) δίνουμε το παρακάτω παράδειγμα:

Εντολή =

L	I	0,1
=	Q	0,1

Εντολές S, R.

L	I	0,1
S	Q	0,1
L	I	0,2
R	Q	0,1

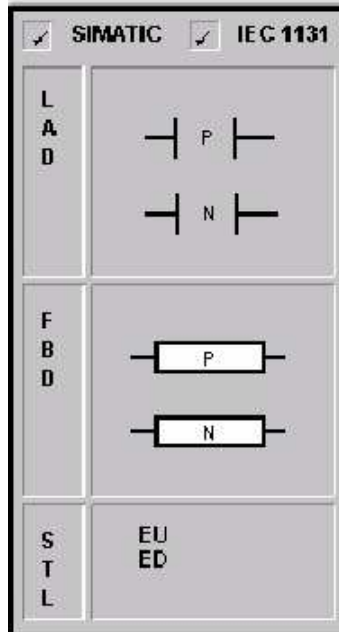
Στο πρόγραμμα που χρησιμοποιούμε την εντολή (=), η έξοδος Q 0.1 είναι σε κατάσταση 1, όσο η είσοδος I 0.1 είναι σε κατάσταση 1. Μόλις η είσοδος I 0.1 αποκτήσει κατάσταση 0 τότε και η έξοδος Q 0.1 αποκτά κατάσταση 0. Στο πρόγραμμα που χρησιμοποιούμε την εντολή SET, μόλις η είσοδος I 0.1 αποκτήσει κατάσταση 1, η έξοδος Q 0.1 αποκτά κατάσταση 1.

Αλλά η έξοδος παραμένει σε κατάσταση 1 ακόμη και όταν η είσοδος I 0.1 επανέλθει σε κατάσταση 0. Για να επανέλθει η έξοδος Q 0.1 σε κατάσταση 0 πρέπει να ενεργοποιηθεί η εντολή RESET, δηλαδή πρέπει η είσοδος I 0.2 να αποκτήσει κατάσταση 1. [35]

Εντολές Θετικής - Αρνητικής μετάβασης

Η σύνταξη των εντολών Positive – Negative transition στις γλώσσες (LAD/STL/FBD) καθώς και η διευθυνσηδότηση φαίνονται.

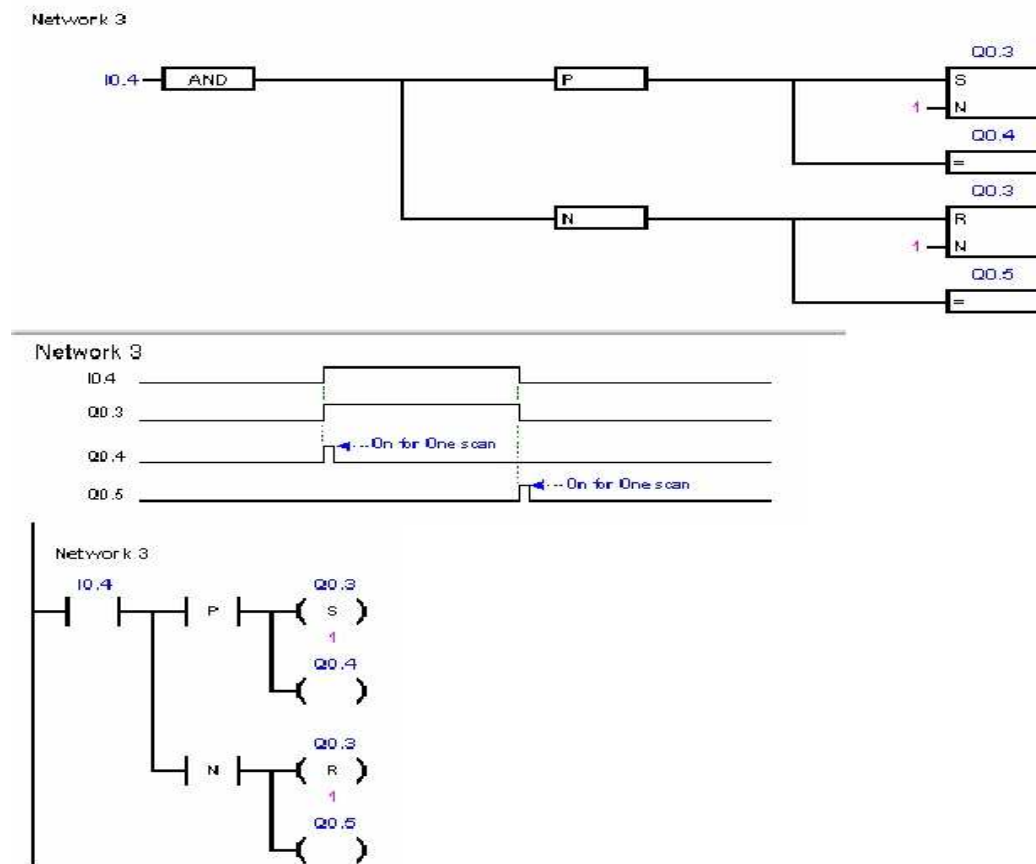
Inputs/Outputs	Data Type	Operands
Bit	BOOL	I, Q, V, M, SM, S, T, C, L, Power Flow
Bit (immediate)	BOOL	I



Σχ.3.12 Εντολές Positive – Negative transition και διευθυνσηδότηση.

Η Positive transition αφήνει να περάσει σήμα για 1 scan του προγράμματος κατά την μετάβαση από την κατάσταση off σε on.

Η Negative transition αφήνει να περάσει σήμα για 1 scan του προγράμματος κατά την μετάβαση από την κατάσταση on σε off.[36]



Σχ.3.13 Παράδειγμα LADDER και FBD εντολών Positive–Negative transition και οι κυματομορφές

```

LD    I0.4
LPS
EU
S     Q0.3, 1
=     Q0.4
LPP
ED
R     Q0.3, 1
=     Q0.5

```

Παράδειγμα σε STL

Στα παραπάνω παραδείγματα, όταν η I0.4 μεταβεί από 0 σε 1 η Q0.4 γίνεται 1 για 1 scan time ενώ παράλληλα η Q0.3 γίνεται set δηλ 1 μέχρις ότου έρθει το reset και γίνει 0. Όταν τώρα η I 0.4 μεταβεί από 1 σε 0 η Q0.5 γίνεται για 1 scan time ενώ παράλληλα η Q0.3 παίρνει reset και μεταβαίνει στο 0.[36]

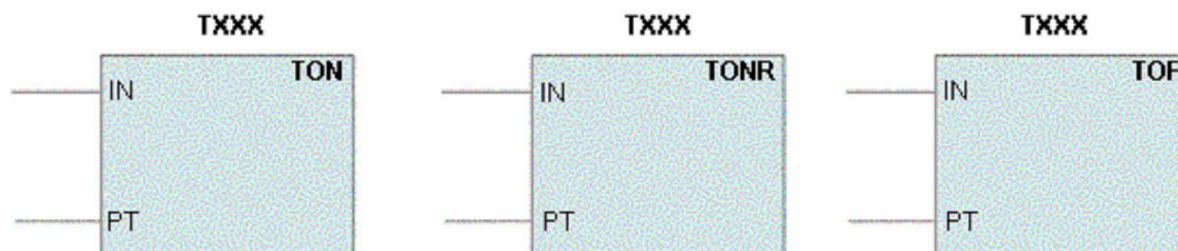
3.6 Ανάπτυξη προγραμμάτων με χρονικές λειτουργίες.

Τα περισσότερα PLC διαθέτουν σημαντικές ευκολίες όσον αφορά στον προγραμματισμό χρονικών λειτουργιών με χρησιμοποίηση των χρονικών λειτουργιών που διαθέτουν. Το κακό είναι ότι στον προγραμματισμό των χρονικών λειτουργιών υπάρχουν μεγάλες διαφορές

μεταξύ των PLC της αγοράς, αντίθετα με ότι συμβαίνει με τις εντολές που είδαμε μέχρι τώρα, όπου οι διαφορές είναι ελάχιστες έως και ανύπαρκτες. Εμείς θα προσπαθήσουμε να παρουσιάσουμε τον προγραμματισμό των χρονικών λειτουργιών με τέτοιο τρόπο, ώστε να μπορεί κάποιος πολύ εύκολα να καταλάβει πως χειρίζεται τα χρονικά το κάθε PLC με το οποίο θα ασχοληθεί. [35]

3.6.1 Χρονικά

Τα χρονικά είναι προγραμματιστικές δομές που υλοποιούν και επιτηρούν χρονικά συνδεδεμένες διαδικασίες. Οι εντολές των χρονικών επιτρέπουν στο πρόγραμμα μας να εκτελούν λειτουργίες, όπως χρόνος αναμονής, χρόνος επιτήρησης, δημιουργία παλμοσειρών και μέτρηση χρόνου. Υπάρχουν διάφορων τύπων χρονικά στα PLC. Στο S7-200 συγκεκριμένα, υπάρχουν χρονικά 3 τύπων. Τα χρονικά αναπαρίστανται στη γλώσσα Ladder με ορθογώνια. Όταν ένα χρονικό δεχθεί το σήμα έναυσης (IN) ξεκινά η μέτρηση του χρόνου και η μετρούμενη τιμή συγκρίνεται συνεχώς με την προκαθορισμένη τιμή (PT). Όσο διάστημα η μετρούμενη τιμή είναι μικρότερη από την προκαθορισμένη τιμή, η έξοδος του χρονικού είναι OFF (λογικό 0). Όταν η μετρούμενη τιμή γίνει μεγαλύτερη από την προκαθορισμένη τιμή, τότε η έξοδος του χρονικού αλλάζει και γίνεται ON (λογικό 1). Οι τρεις τύποι χρονικού του S7 είναι: καθυστέρησης έλξης (On Delay TON), καθυστέρησης έλξης με αυτοσυγκράτηση (Retentive On Delay TONR) και καθυστέρησης πτώσης (Off Delay TOF). Οι αναλύσεις χρόνου για τα χρονικά είναι: ανάλυση 1 millisecond, 10 millisecond και 100 millisecond με μέγιστη τιμή μέτρησης 32,767 second, 327,67 second και 3.276,7 second αντίστοιχα. Για μεγαλύτερους χρόνους χρησιμοποιούμε επιπλέον πρόγραμμα. [35]



Σχ.3.14 Σχηματική αναπαράσταση χρονικών στη γλώσσα LADDER

3.6.2 Ανάλυση χρονικών

Τα χρονικά μετρούν χρονικά διαστήματα. Η ανάλυση (βάση χρόνου) καθορίζει το ποσό του χρόνου, που αντιστοιχεί σε ένα χρονικό διάστημα. Ο χρόνος που μετράει το χρονικό είναι λοιπόν το γινόμενο της ανάλυσης επί τον αριθμό των διαστημάτων. Παρακάτω βλέπουμε τον κάθε τύπο χρονικού πιο αναλυτικά.

- Χρονικό καθυστερημένης έλξης (On Delay Timer)

Αυτό το χρονικό μετράει χρόνο όσο η είσοδος είναι "1" (αυτό συμβαίνει για ένα χρονικό διάστημα). Έτσι όταν η είσοδος του χρονικού μηδενιστεί για κάποιο λόγο τότε μηδενίζεται και ο χρόνος του χρονικού. Σε νέα ενεργοποίηση της εισόδου ο χρόνος αρχίζει να μετρά από το μηδέν. Όταν ο χρόνος που έχει μετρήσει το χρονικό είναι μεγαλύτερος ή ίσος από την προκαθορισμένη τιμή το BIT του χρονικού ενεργοποιείται. Στον παρακάτω πίνακα βλέπουμε τις χρονικές τιμές που μπορεί να πάρει ο TON.

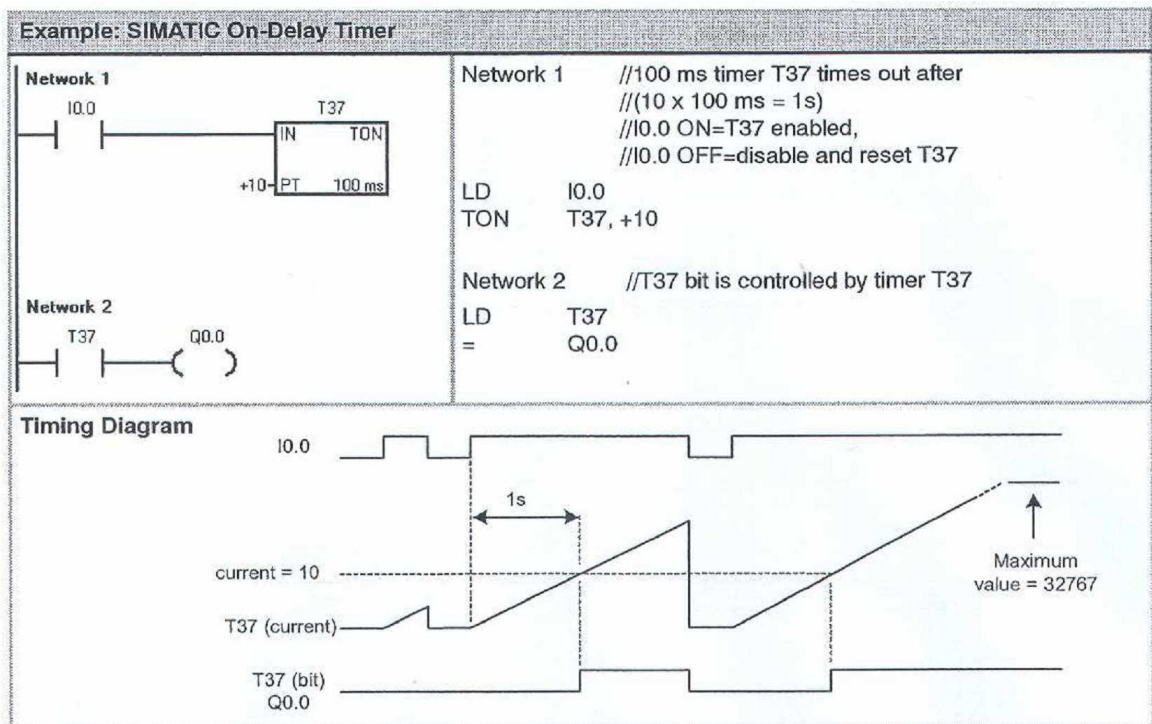
Τύπος χρονικού	Τιμές σε ms	Max τιμή σε sec	Αριθμός Timer
TON	1ms	32,767 sec	T32,T96
	10ms	327,67 sec	T33-T36,T97- T100
	100ms	3276,7 sec	T36-T63,T101- T255

Για παράδειγμα η εντολή TON T32,+50 μας δηλώνει ότι ο timer TON μετράει για $50 * 1\text{ms} = 50\text{ms}$.

Ομοίως η εντολή TON T35,+60 μας δηλώνει ότι ο timer TON μετράει για $60 * 10\text{ms} = 600\text{ms} = 0,6\text{sec}$.

Ομοίως η εντολή TON T60,+80 μας δηλώνει ότι ο timer TON μετράει για $80 * 100\text{ms} = 8000\text{ms} = 8\text{sec}$.

Στο παρακάτω σχήμα φαίνεται το χρονικό διάγραμμα του χρονικού TON T36,+10 δηλαδή όπως προείπαμε $10 * 100\text{ms} = 1\text{sec}$.



Σχ.3.15 Παραδείγματα χρονικού καθυστερημένης έλξης (On Delay Timer)

Παράδειγμα: Με το πάτημα του διακόπτη S1 θέλουμε να ανάβει η λυχνία εξόδου L1 και να παραμένει ανοικτή για 2 sec.

```

LD I0.0
S Q0.0, 1
TON T36, +20
LD T36
R Q0.0, 1
END

```

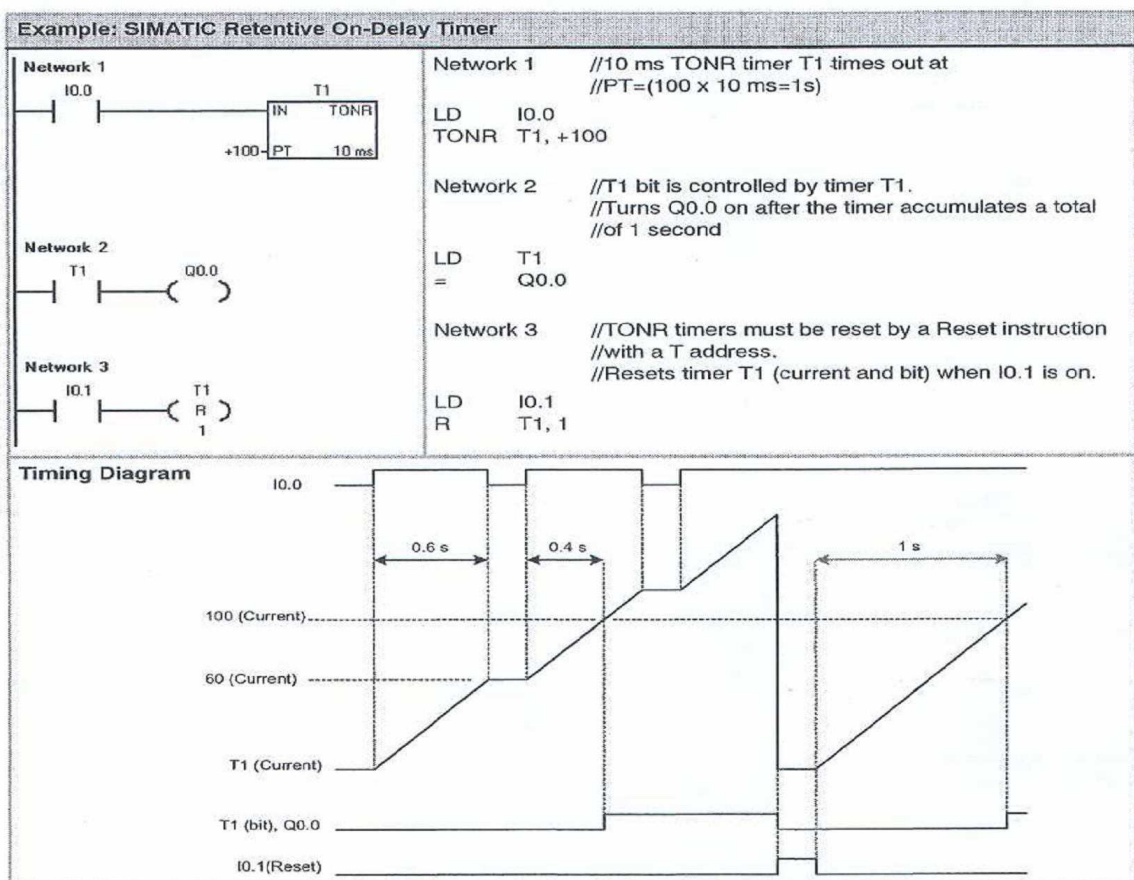
Επεξήγηση: Στο πρόγραμμα αυτό πατώντας τον διακόπτη I0.0 κάνουμε SET στην έξοδο Q0.0 με την εντολή S Q0.0,1 δηλαδή ανάβουμε την λυχνία. Ταυτόχρονα αρχίζει να μετράει ο timer

με την εντολή TON T36,+20 για $20 \times 100\text{ms} = 2\text{sec}$ και όταν φτάσει την τιμή αυτή γίνεται reset στην έξοδο με την εντολή R Q0.0,1.

- Χρονικό καθυστερημένης έλξης με αυτοσυγκράτηση (Retentive On Delay Timer)

Το χρονικό αυτό ενεργοποιείται εφόσον συμβεί μεταβολή από "0" σε "1". Για όση ώρα έχουμε κατάσταση "1" το χρονικό θα μετράει ωστόσο φτάσει την τελική του τιμή που εμείς του έχουμε δώσει μέσα στο πρόγραμμα μας. Αν υπάρχει κατάσταση "1" περισσότερη ώρα από τον χρόνο που έχουμε ορίσει στο χρονικό να μετρήσει τότε αυτό αφού πάρει την τελική του τιμή θα παραμείνει σε αυτήν μέχρι την αλλαγή κατάστασης. Αν έχουμε αλλαγή κατάστασης από "1" σε "0" πριν το χρονικό φτάσει στην τελική του τιμή τότε κρατάει την τιμή που έχει εκείνη την χρονική στιγμή της αλλαγής της κατάστασης μέχρι την επόμενη αλλαγή δηλαδή από "0" σε "1" όπου και συνεχίζει να μετράει από την τιμή αυτή ως την τελική του τιμή. Από τα παραπάνω βγάζουμε το συμπέρασμα ότι το χρονικό αυτό δεν μηδενίζεται όσες αλλαγές καταστάσεων και αν γίνουν. Για να καταφέρουμε τον μηδενισμό του χρησιμοποιούμε την εντολή Reset και μόνο αυτήν. Στον παρακάτω πίνακα βλέπουμε τις χρονικές τιμές που μπορεί να πάρει ο TONR.

Τύπος χρονικού	Τιμές σε ms	Max τιμή σε sec	Αριθμός Timer
TONR	1ms	32,767 sec	T0,T64
	10ms	327,67 sec	T1-T4,T65-T68
	100ms	3276,7 sec	T5-T31,T69-T95



Σχ.3.16 Παράδειγμα χρονικού καθυστερημένης έλξης με αυτοσυγκράτηση (Retentive On Delay Timer)

Παράδειγμα: Με το πάτημα του διακόπτη S1 θέλουμε να ανάβει η λυχνία εξόδου L1 και να παραμένει ανοικτή. Θέλουμε η λυχνία να κλείσει όταν ο συνολικός χρόνος που ο διακόπτης είναι κλειστός (έχει σήμα "1") να είναι 5 sec ανεξάρτητα αν ανοίξουμε τον διακόπτη για κάποια χρονικά διαστήματα πριν την συμπλήρωση του χρόνου αυτού. Επίσης να δοθεί δυνατότητα επανάληψης του προγράμματος για όσες φορές επιθυμούμε.

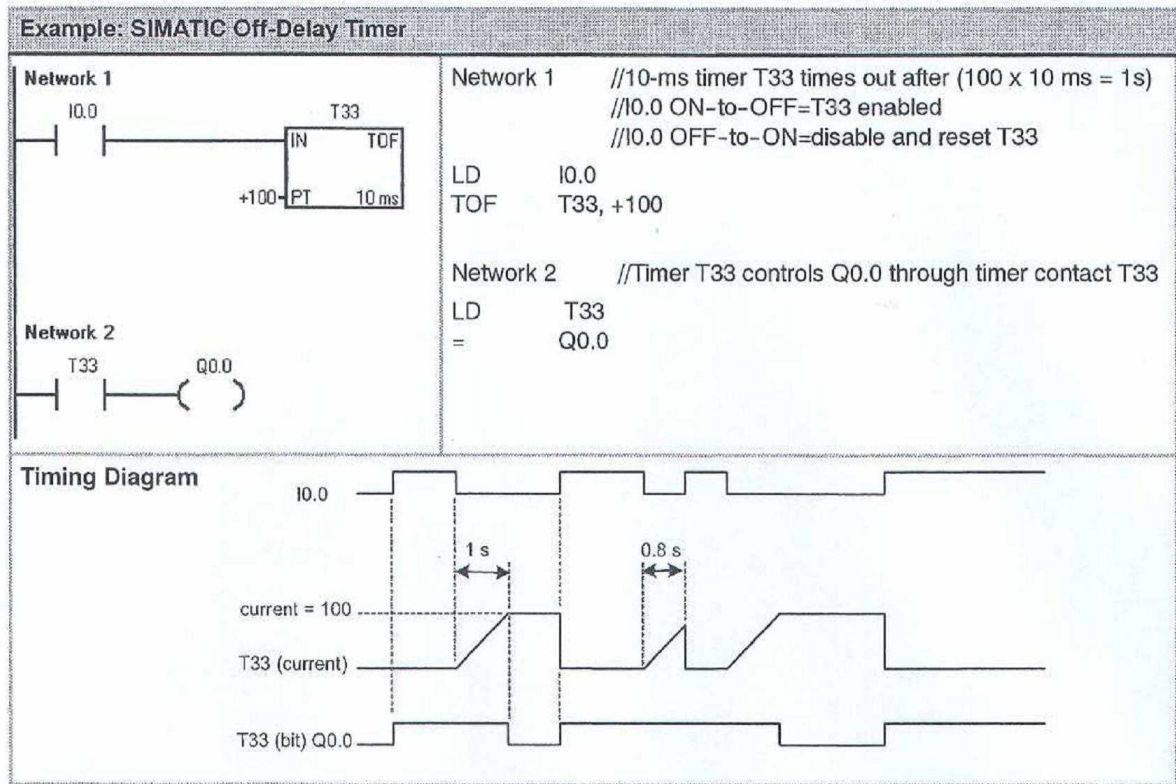
```
LD I0.0
S Q0.0, 1
TO NR T5, +50
LD T5
R Q0.0, 1
LD I0.1
R T5, 1
END
```

Επεξήγηση: Στο πρόγραμμα αυτό πατώντας τον διακόπτη I0.0 κάνουμε SET στην έξοδο Q0.0 με την εντολή S Q0.0,1 δηλαδή ανάβουμε την λυχνία. Όταν ανοίξουμε τον διακόπτη αρχίζει να μετράει ο timer με την εντολή TONR T5,+50 για $50 \cdot 100\text{ms} = 5 \text{ sec}$. Αν πριν την συμπλήρωση του χρόνου ο διακόπτης ανοίξει τότε το χρονικό κρατάει την τιμή που έχει εκείνη την χρονική στιγμή μέχρι να ξανακλείσει όπου και συνεχίζει την μέτρηση από αυτήν την τιμή μέχρι τα 5 sec. Για να επαναλάβουμε την μέτρηση πρέπει να κάνουμε reset στο χρονικό με κάποιον διακόπτη που έχουμε ορίσει. Το reset αυτό γίνεται με την εντολή R T5,1.

- Χρονικό καθυστερημένης πτώσης (Off Delay Timer)

Αυτό το χρονικό χρησιμοποιείται για να καθυστερήσει την απενεργοποίηση μιας εξόδου για ένα συγκεκριμένο χρονικό διάστημα από την απενεργοποίηση της εισόδου. Μόλις λοιπόν η είσοδος απενεργοποιηθεί τότε το χρονικό ξεκινάει την μέτρηση του για το χρονικό διάστημα που του έχουμε προσδώσει. Αν η είσοδος παραμένει απενεργοποιημένη για περισσότερο χρονικό διάστημα από την διάρκεια του χρονικού τότε αυτό θα σταματήσει την μέτρηση και θα παραμείνει σταθερό στην τελική του τιμή ώσπου να ενεργοποιήσουμε την είσοδο και πάλι όπου το χρονικό μηδενίζεται. Με την επόμενη ενεργοποίηση της εισόδου ξεκινάει η ίδια διαδικασία. Στον παρακάτω πίνακα βλέπουμε τις χρονικές τιμές που μπορεί να πάρει ο TOF.

Τύπος χρονικού	Τιμές σε ms Max τιμή σε sec	Αριθμός Timer
TOF	1ms 32,767 sec	T32,T96
	10ms 327,67 sec	T33-T36,T97-T100
	100ms 3276,7 sec	T36-T63,T101-T255



Σχ.3.17 Παράδειγμα χρονικού καθυστερημένης πτώσης (Off Delay Timer)

Παράδειγμα: Με το πάτημα του διακόπτη S1 θέλουμε να ανάβει η λυχνία εξόδου L1 και να παραμένει ανοικτή. Όταν κλείσουμε τον διακόπτη θέλουμε η λυχνία να κλείσει μετά από 5 sec.

```
LD I0.0
A I0.0
S Q0.0, 1
A I0.0
TOF T33, +500
LDN T33
R Q0.0, 1
END
```

Επεξήγηση: Στο πρόγραμμα αυτό πατώντας τον διακόπτη I0.0 κάνουμε SET στην έξοδο Q0.0 με την εντολή S Q0.0,1 δηλαδή ανάβουμε την λυχνία. Όταν ανοίξουμε τον διακόπτη αρχίζει να μετράει ο timer με την εντολή TOF T33,+500 για $500 \times 10\text{ms} = 5 \text{ sec}$ και όταν φτάσει την τιμή αυτή γίνεται reset στην έξοδο με την εντολή R Q0.0,1

Παράδειγμα: Να δημιουργήσετε ένα πρόγραμμα στο οποίο η έξοδος Q0.0 να δουλεύει σαν clock με διάρκεια παλμού 2 sec.

```
LD I1.0
A I1.0
AN T38
TON T36, +20
LD T36
= Q0.0
A T36
TON T38, +20
```

END

Επεξήγηση: Στο πρόγραμμα αυτό με το πάτημα του διακόπτη I0.0 έχουμε σβήσιμο της εξόδου για 2 sec με την εντολή TON T36, +20 και στη συνέχεια άναμμα της εξόδου για 2 sec με την εντολή TON T38, +20. [35]

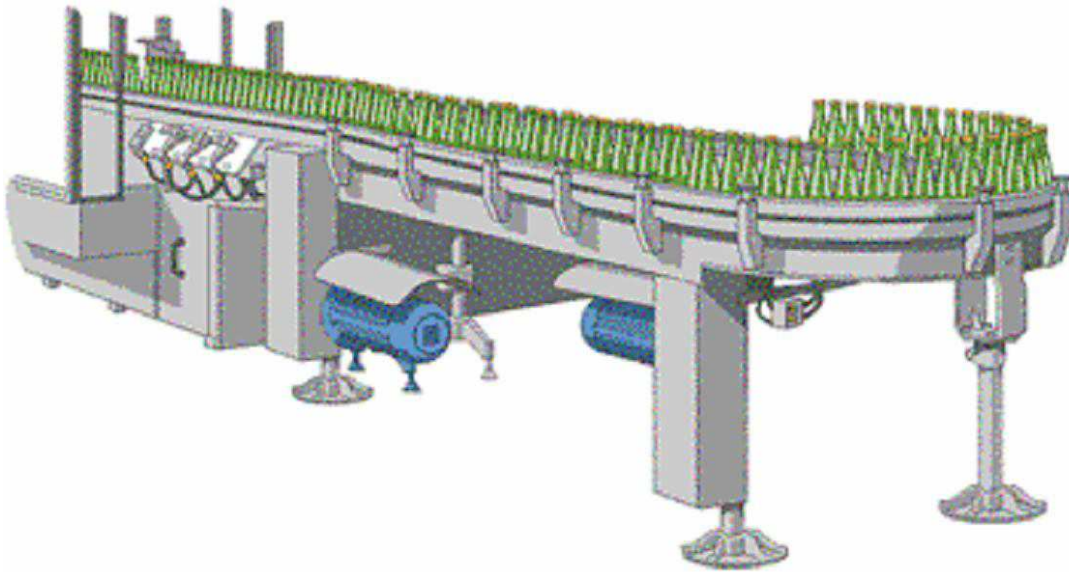
3.7 Απαριθμητές (Counters)

Οι απαριθμητές μας δίνουν την δυνατότητα να εκτελούμε λειτουργίες απαρίθμησης μέσα στην CPU. Οι απαριθμητές μετρούν προς τα πάνω ή προς τα κάτω. Για να γίνει μια μέτρηση η CPU πρέπει να αντιληφθεί αλλαγή στην κατάσταση του σήματος σε μια είσοδο.

Οι απαριθμητές στα PLC εκτελούν λειτουργία αντίστοιχη με αυτή των μηχανικών απαριθμητών. Η μετρούμενη τιμή συγκρίνεται με μία προκαθορισμένη τιμή. Συνήθως οι απαριθμητές λειτουργούν με δύο διαφορετικές λογικές:

- Μετρούν μέχρι την προκαθορισμένη τιμή και προκαλούν ένα γεγονός
- Προκαλούν ένα γεγονός μέχρι η μέτρηση να φτάσει την προκαθορισμένη τιμή.

Μία μηχανή (γραμμή παραγωγής) εμφιάλωσης για παράδειγμα μπορεί να χρησιμοποιεί έναν απαριθμητή για να μετρά βάζες μπουκαλιών που μετά θα μπουν μαζί σε μία συσκευασία. Σε αυτήν την περίπτωση ο απαριθμητής λειτουργεί με την πρώτη από τις δύο λογικές που περιγράφηκαν παραπάνω.



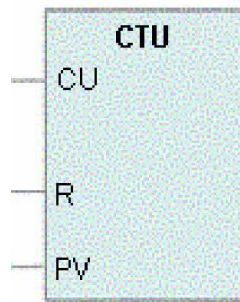
Σχ.3.18 Παράδειγμα χρήσης απαριθμητή σε γραμμή παραγωγής.

3.7.1 Κατηγορίες απαριθμητών

Υπάρχουν τρεις κατηγορίες απαριθμητών:

- Counter up (CTU).

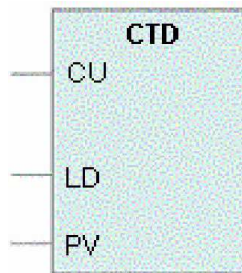
Σε αυτόν τον μετρητή αυξάνεται η τιμή του κατά ένα κάθε φορά που η είσοδος CU μεταβαίνει από το μηδέν στο ένα. Όταν η τιμή του μετρητή γίνει ίση ή μεγαλύτερη από την προκαθορισμένη τιμή PV το bit του απαριθμητή γίνεται ένα. Όταν ο μετρητής φθάσει την επιθυμητή τιμή σταματάει να απαριθμεί. Ο απαριθμητής απενεργοποιείται-μηδενίζεται, όταν η είσοδος του reset μεταβεί από το μηδέν στο ένα. Παρακάτω απεικονίζεται ο CTU απαριθμητής στην γλώσσα Ladder.



Σχ.3.19 Ο CTU απαριθμητής.

- Counter down (CTD).

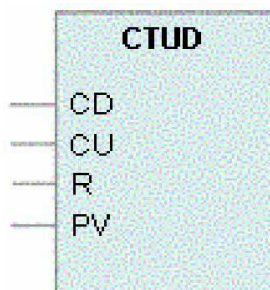
Σε αυτόν τον μετρητή μειώνεται η τιμή του κατά ένα κάθε φορά που η είσοδος CU μεταβαίνει από το μηδέν στο ένα. Όταν η τιμή του μετρητή γίνει ίση με το μηδέν το bit του απαριθμητή γίνεται ένα. Όταν ο μετρητής φθάσει την τιμή μηδένσταματάει να απαριθμεί. Ο απαριθμητής απενεργοποιείται και η τρέχουσα τιμή του τίθεται ίση με την προκαθορισμένη τιμή PV όταν η είσοδος του reset μεταβεί από το μηδέν στο ένα. Παρακάτω απεικονίζεται ο CTD απαριθμητής στην γλώσσα Ladder.



Σχ.3.20 Ο CTD απαριθμητής.

- Counter up/down (CTUD).

Αυτός ο μετρητής απαριθμεί προς τα πάνω, όταν η είσοδος CU μεταβαίνει από το μηδέν στο ένα και απαριθμεί προς τα κάτω, όταν η είσοδος του CD μεταβαίνει από το μηδέν στο ένα. Όταν η τιμή του μετρητή είναι μεγαλύτερη ή ίση από την προκαθορισμένη τιμή, ενεργοποιείται το bit του απαριθμητή. Ο απαριθμητής παύει να απαριθμεί όταν φτάσει την προκαθορισμένη τιμή PV. Ο απαριθμητής απενεργοποιείται όταν ενεργοποιείται η είσοδος reset. Παρακάτω απεικονίζεται ο CTUD απαριθμητής στην γλώσσα Ladder. [35]

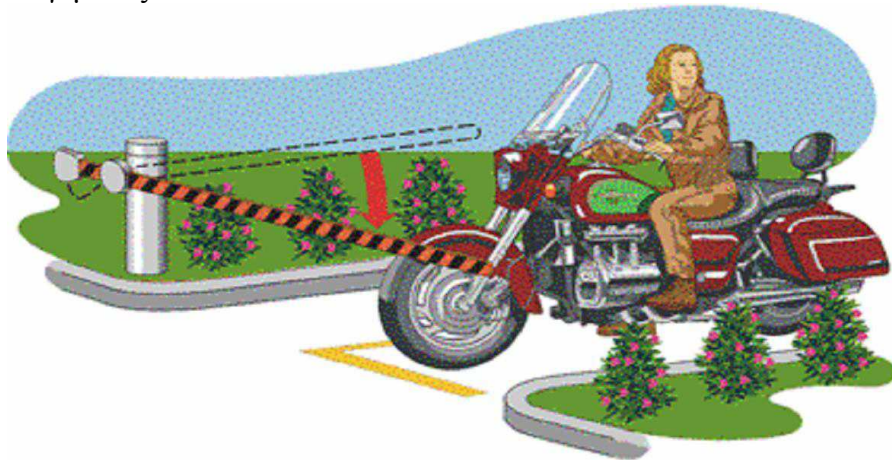


Σχ.3.21 Ο CTUD απαριθμητής.

3.7.2 Παράδειγμα χρήσης απαριθμητών στην πράξη.

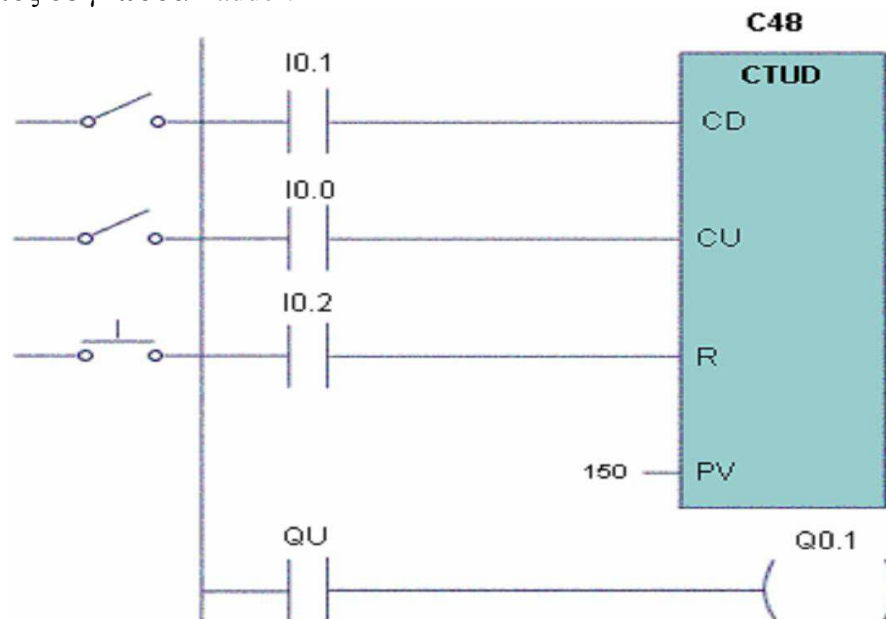
Ένας απαριθμητής μπορεί να χρησιμοποιηθεί για να ελέγχεται ο αριθμός των οχημάτων σε ένα χώρο στάθμευσης. Όταν τα οχήματα μπαίνουν στο χώρο από την

είσοδο, ο απαριθμητής μετρά προς τα πάνω. Όταν βγαίνουν από την έξοδο ο απαριθμητής μετρά προς τα κάτω. Όταν η μετρούμενη τιμή φτάσει την προκαθορισμένη τιμή που αντιστοιχεί στο μέγιστο αριθμό οχημάτων που χωρά ο χώρος στάθμευσης τότε ενεργοποιείται μια ενδεικτική λυχνία στην είσοδο του παρκινγκ που μας ειδοποιεί ότι ο χώρος στάθμευσης είναι γεμάτος.



Σχ.3.22 Παράδειγμα χρήσης απαριθμητών σε χώρο στάθμευσης οχημάτων.

Για να υλοποιηθεί η εφαρμογή χρησιμοποιείται ένας απαριθμητής αύξησης/μείωσης, συγκεκριμένα ο up/down counter 48. Ένας διακόπτης που έχει προσαρμοσθεί στην είσοδο του χώρου στάθμευσης συνδέεται στην είσοδο I0.0. Ένας άλλος διακόπτης, προσαρμοσμένος στην έξοδο, συνδέεται στην είσοδο I0.1. Ο χώρος στάθμευσης έχει χώρο για 150 οχήματα. Αυτή είναι και η τιμή που δίνεται στον απαριθμητή ως προκαθορισμένη τιμή μέτρησης PV. Στην έξοδο Q0.1 συνδέεται η ενδεικτική λυχνία “πληρότητας” του χώρου στάθμευσης. Αν η μέτρηση στον απαριθμητή φθάσει το 150 η έξοδος Q0.1 ενεργοποιείται και ανάβει η ενδεικτική λυχνία. Αν ένα αυτοκίνητο βγει από το χώρο στάθμευσης, η μέτρηση γίνεται 149 και η ενδεικτική λυχνία σβήνει. Στο παρακάτω σχήμα φαίνεται η υλοποίηση του εν λόγω προγράμματος σε γλώσσα Ladder.



Σχ.3.23 Υλοποίηση προγράμματος σε γλώσσα Ladder.

Παράδειγμα: Σε μια διαδικασία ελέγχου θέλουμε να γίνονται τα εξής: Όταν περάσουν 5 αυτοκίνητα από την είσοδο του παρκινγκ να ανάβει μια λυχνία η οποία θα φωτοβολεί για 3 sec. Η διαδικασία να επαναλαμβάνεται συνέχεια.

```
LD I0.0
LD I0.1
CTU C34, +5
LD C34
= Q0.0
TON T36, +30
LD T36
R Q0.0, 1
END
```

Επεξήγηση: Στο πρόγραμμα αυτό με τον διακόπτη I0.0 μεταβάλλουμε τον απαριθμητή ενώ με τον I0.1 μηδενίζουμε τον απαριθμητή. Όταν θα περάσουν 5 οχήματα ανάβει η λυχνία Q0.0. Ταυτόχρονα ενεργοποιείται ο timer για 3 δευτερόλεπτα. Με την εντολή R Q0.0, 1 κάνουμε reset στην έξοδο. [35]

3.8 Συγκρίσεις

Οι λειτουργίες σύγκρισης συγκρίνουν δύο ψηφιακές τιμές αριθμητικές ή λογικές. Το αποτέλεσμα της σύγκρισης επηρεάζει το RLO και με αυτό μπορούμε να πάρουμε κάποιες αποφάσεις όπως για παράδειγμα όταν η στάθμη μιας δεξαμενής φτάσει το 90% του μέγιστου επιτρεπτού να ανάβει μια προειδοποιητική λυχνία. Οι συγκρίσεις γίνονται στην CPU και εκτελούνται ανεξάρτητα από την ικανοποίηση ή μη ορισμένων συνθηκών ή λογικών μανδαλώσεων. Ανάλογα με το είδος των αριθμών που πρόκειται να συγκρίνουμε (Integer, Real, Byte) υπάρχει και η αντίστοιχη εντολή.

• ΣΥΓΚΡΙΣΗ ΑΚΕΡΑΙΩΝ

Οι εντολές σύγκρισης ακεραίων χρησιμοποιούνται για την σύγκριση δύο ακεραίων τιμών έστω INT 1 και INT 2 και περιλαμβάνουν τις παρακάτω πράξεις:

- Εντολή ισότητας: LDW= INT 1, INT 2
- Εντολή διάφορου: LDW<> INT 1, INT 2
- Εντολή μικρότερου από: LDW< INT 1, INT 2
- Εντολή μεγαλύτερου από: LDW> INT 1, INT 2
- Εντολή μικρότερου ή ίσου: LDW<= INT 1, INT 2
- Εντολή μεγαλύτερου ή ίσου: LDW>= INT 1, INT 2

Παράδειγμα: Να γίνει πρόγραμμα που να συγκρίνει δύο ακεραίους.

```
LDW>= +12, +5
= Q0.0
END
```

Με την εκτέλεση του προγράμματος θα γίνει η παραπάνω σύγκριση στην CPU και αν αληθεύει θα ενεργοποιηθεί η έξοδος Q0.0.

• ΣΥΓΚΡΙΣΗ ΠΡΑΓΜΑΤΙΚΩΝ

Οι εντολές σύγκρισης πραγματικών χρησιμοποιούνται για την σύγκριση δύο πραγματικών τιμών έστω INT 1 και INT 2 και περιλαμβάνουν τις παρακάτω πράξεις:

- Εντολή ισότητας: LDR= INT 1,INT 2
- Εντολή διάφορου: LDR<> INT 1,INT 2
- Εντολή μικρότερου από: LDR< INT 1,INT 2
- Εντολή μεγαλύτερου από: LDR> INT 1,INT 2
- Εντολή μικρότερου ή ίσου: LDR<= INT 1,INT 2
- Εντολή μεγαλύτερου ή ίσου: LDR>= INT 1,INT 2

Παράδειγμα: Να γίνει πρόγραμμα που να συγκρίνει δύο πραγματικούς.

```
LDR<>    17.8, 12.7
=        Q0.0
END
```

Με την εκτέλεση του προγράμματος θα γίνει η παραπάνω σύγκριση στην CPU και αν αληθεύει θα ενεργοποιηθεί η έξοδος Q0.0.

• ΣΥΓΚΡΙΣΗ BYTE

Οι εντολές σύγκρισης byte χρησιμοποιούνται για την σύγκριση δύο τιμών έστω INT1 και INT 2 και περιλαμβάνουν τις παρακάτω πράξεις:

- Εντολή ισότητας: LDB= INT 1,INT 2
 - Εντολή διάφορου: LDB<> INT 1,INT 2
 - Εντολή μικρότερου από: LDB< INT 1,INT 2
 - Εντολή μεγαλύτερου από: LDB> INT 1,INT 2
 - Εντολή μικρότερου ή ίσου: LDB<= INT 1,INT 2
 - Εντολή μεγαλύτερου ή ίσου: LDB>= INT 1,INT 2
- [35]

3.9 Αριθμητικές πράξεις

Οι αριθμητικές πράξεις εφαρμόζονται πάνω σε δύο ψηφιακές τιμές που βρίσκονται η μια στον Accumulator 1 και η άλλη στον Accumulator 2. Οι πράξεις εκτελούνται ανεξάρτητα από την τιμή του RLO δηλαδή άσχετα από την ικανοποίηση ή μη ορισμένων συνθηκών ή λογικών μανδαλώσεων.

• ΠΡΟΣΘΕΣΗ ΚΑΙ ΑΦΑΙΡΕΣΗ ΑΚΕΡΑΙΩΝ.

Στην περίπτωση πρόσθεσης ή αφαίρεσης ακεραίων αριθμών οι πράξεις αυτές θα γίνουν μόνο μεταξύ των δεξιών τμημάτων των δύο Accumulators επειδή όπως είπαμε και παραπάνω οι πράξεις αυτές γίνονται μόνο μεταξύ των περιεχομένων δύο Accumulators. Για να κάνουμε λοιπόν πρόσθεση ή αφαίρεση ακεραίων αριθμών πρέπει να φορτώσουμε κάθε αριθμό σε κάποιον Accumulator.

Η σύνταξη μιας εντολής πρόσθεσης δύο ακεραίων αριθμών έχει ως εξής:

```
+I    IN1, OUT
```

Η σύνταξη μιας εντολής αφαίρεσης δύο ακεραίων αριθμών έχει ως εξής:

```
-I    IN1, OUT
```

Τα IN1 και OUT συμβολίζουν δύο ακεραίους αριθμούς και τα +I, -I την πράξη της πρόσθεσης και της αφαίρεσης αντίστοιχα. Μετά την εκτέλεση κάποιας από τις δύο παραπάνω

πράξεις ο IN1 παραμένει αμετάβλητος ενώ στον OUT θα αποθηκευτεί το αποτέλεσμα της αριθμητικής πράξης που εκτελέστηκε. Δηλαδή ισχύουν οι δύο παρακάτω ισότητες:

IN1+OUT = OUT
OUT-IN1 = OUT

Παράδειγμα: Να γίνει ένα πρόγραμμα το οποίο θα προσθέτει δύο ακεραίους αριθμούς και ένα άλλο το οποίο θα τους αφαιρεί.

LD	I0.0	LD	I0.0
A	I0.0	A	I0.0
MOVW	+4, AC1	MOVW	+5, AC1
MOVW	+7, AC2	MOVW	+8, AC2
+I	AC2, AC1	-I	AC2, AC1
END		END	

Στα δύο παραπάνω προγράμματα χρησιμοποιούμε τον διακόπτη I0.0 για να αποθηκεύσουμε τους δύο ακεραίους αριθμούς στους δύο Accumulators και στην συνέχεια να εκτελεστούν οι αριθμητικές πράξεις. Όπως είπαμε και παραπάνω το περιεχόμενο του AC2 δεν θα αλλάξει μετά από τις πράξεις όμως το περιεχόμενο του AC1 θα αλλάξει και θα περιλαμβάνει το αποτέλεσμα της πράξης που γίνεται δηλαδή το άθροισμα των AC1 και AC2 για την πρόσθεση και το υπόλοιπο AC1 και AC2 για την αφαίρεση αντίστοιχα.

•ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΣ ΚΑΙ ΔΙΑΙΡΕΣΗ ΑΚΕΡΑΙΩΝ.

Για να κάνουμε λοιπόν πολλαπλασιασμό και διαίρεση κατά όμοιο τρόπο με την πρόσθεση και την αφαίρεση ακεραίων αριθμών πρέπει να φορτώσουμε κάθε αριθμό σε κάποιον Accumulator.

Η σύνταξη μιας εντολής πολλαπλασιασμού δύο ακεραίων αριθμών έχει ως εξής:

***I IN1, OUT**

Η σύνταξη μιας εντολής διαίρεσης δύο ακεραίων αριθμών έχει ως εξής:

/I IN1, OUT

Τα IN1 και OUT συμβολίζουν δύο ακεραίους αριθμούς και τα *I, /I την πράξη του πολλαπλασιασμού και της διαίρεσης αντίστοιχα. Μετά την εκτέλεση κάποιας από τις δύο παραπάνω πράξεις ο IN1 παραμένει αμετάβλητος ενώ στον OUT θα αποθηκευτεί το αποτέλεσμα της αριθμητικής πράξης που εκτελέστηκε. Δηλαδή ισχύουν οι δύο παρακάτω ισότητες:

IN1*OUT = OUT
OUT/IN1 = OUT

Παράδειγμα: Να γίνει ένα πρόγραμμα το οποίο θα πολλαπλασιάζει δύο ακεραίους αριθμούς και ένα άλλο το οποίο θα τους διαιρεί.

LD	I0.0	LD	I0.0
A	I0.0	A	I0.0
MOVW	+7, AC1	MOVW	+9, AC1

*I	2, AC1	/I	AC1, +3
END		END	

Στα δύο παραπάνω προγράμματα χρησιμοποιούμε τον διακόπτη I0.0 για να αποθηκεύσουμε τον ακέραιο αριθμό στον AC1 και στην συνέχεια να εκτελεστούν οι αριθμητικές πράξεις. Μπορούμε να χρησιμοποιήσουμε και άλλον ένα AC αποθηκεύοντας σε αυτόν τον αριθμό που θέλουμε να πολλαπλασιάσουμε ή να διαιρέσουμε όπως στην πρόσθεση αφαίρεση.

•ΠΡΟΣΘΕΣΗ ΚΑΙ ΑΦΑΙΡΕΣΗ ΠΡΑΓΜΑΤΙΚΩΝ

Στην περίπτωση πρόσθεσης ή αφαίρεσης πραγματικών αριθμών οι πράξεις αυτές θα γίνουν μόνο μεταξύ των δεξιών τμημάτων των δύο Accumulators επειδή όπως είπαμε και παραπάνω οι πράξεις αυτές γίνονται μόνο μεταξύ των περιεχομένων δύο Accumulators. Για να κάνουμε λοιπόν πρόσθεση ή αφαίρεση πραγματικών αριθμών πρέπει να φορτώσουμε κάθε αριθμό σε κάποιον Accumulator.

Η σύνταξη μιας εντολής πρόσθεσης δύο πραγματικών αριθμών έχει ως εξής:

+R IN1, OUT

Η σύνταξη μιας εντολής αφαίρεσης δύο πραγματικών αριθμών έχει ως εξής:

-R IN1, OUT

Τα IN1 και OUT συμβολίζουν δύο πραγματικών αριθμούς και τα +R, -R την πράξη της πρόσθεσης και της αφαίρεσης αντίστοιχα. Μετά την εκτέλεση κάποιας από τις δύο παραπάνω πράξεις ο IN1 παραμένει αμετάβλητος ενώ στον OUT θα αποθηκευτεί το αποτέλεσμα της αριθμητικής πράξης που εκτελέστηκε. Δηλαδή ισχύουν οι δύο παρακάτω ισότητες:

IN1+OUT = OUT

OUT-IN1 = OUT

Παράδειγμα: Να γίνει ένα πρόγραμμα το οποίο θα προσθέτει δύο πραγματικού αριθμούς και ένα άλλο το οποίο θα τους αφαιρεί.

LD	I0.0	LD	I0.0
A	I0.0	A	I0.0
MOVR	4.7, AC1	MOVR	3.5, AC1
MOVR	7.2, AC2	MOVR	5.8, AC2
+R	AC2, AC1	-R	AC2, AC1
END		END	

Στα δύο παραπάνω προγράμματα χρησιμοποιούμε τον διακόπτη I0.0 για να αποθηκεύσουμε τους δύο πραγματικούς αριθμούς στους δύο Accumulators και στην συνέχεια να εκτελεστούν οι αριθμητικές πράξεις. Όπως είπαμε και παραπάνω το περιεχόμενο του AC2 δεν θα αλλάξει μετά από τις πράξεις όμως το περιεχόμενο του AC1 θα αλλάξει και θα περιλαμβάνει το αποτέλεσμα της πράξης που γίνεται δηλαδή το άθροισμα των AC1 και AC2 για την πρόσθεση και το υπόλοιπο AC1 και AC2 για την αφαίρεση αντίστοιχα.

•ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΣ ΚΑΙ ΔΙΑΙΡΕΣΗ ΠΡΑΓΜΑΤΙΚΩΝ.

Για να κάνουμε λοιπόν πολλαπλασιασμό και διαίρεση κατά όμοιο τρόπο με την πρόσθεση και την αφαίρεση πραγματικών αριθμών πρέπει να φορτώσουμε κάθε αριθμό σε κάποιον Accumulator.

Η σύνταξη μιας εντολής πολλαπλασιασμού δύο πραγματικών αριθμών έχει ως εξής:

***R IN1, OUT**

Η σύνταξη μιας εντολής διαίρεσης δύο πραγματικών αριθμών έχει ως εξής:

/R IN1, OUT

Τα IN1 και OUT συμβολίζουν δύο πραγματικούς αριθμούς και τα *R, /R την πράξη του πολλαπλασιασμού και της διαίρεσης αντίστοιχα. Μετά την εκτέλεση κάποιας από τις δύο παραπάνω πράξεις ο IN1 παραμένει αμετάβλητος ενώ στον OUT θα αποθηκευτεί το αποτέλεσμα της αριθμητικής πράξης που εκτελέστηκε. Δηλαδή ισχύουν οι δύο παρακάτω ισότητες:

$$\text{IN1} * \text{OUT} = \text{OUT}$$

$$\text{OUT} / \text{IN1} = \text{OUT}$$

Παράδειγμα: Να γίνει ένα πρόγραμμα το οποίο θα πολλαπλασιάζει δύο πραγματικούς αριθμούς και ένα άλλο το οποίο θα τους διαιρεί.

LD	I0.0	LD	I0.0
A	I0.0	A	I0.0
MOVR	7.0, AC1	MOVR	9.1, AC1
*R	2.3, AC1	/R	AC1, +3.5
END		END	

Στα δύο παραπάνω προγράμματα χρησιμοποιούμε τον διακόπτη I0.0 για να αποθηκεύσουμε τον πραγματικό αριθμό στον AC1 και στην συνέχεια να εκτελεστούν οι αριθμητικές πράξεις. Μπορούμε να χρησιμοποιήσουμε και άλλον ένα AC αποθηκεύοντας σε αυτόν τον αριθμό που θέλουμε να πολλαπλασιάσουμε ή να διαιρέσουμε όπως στην πρόσθεση αφαίρεση.

Παράδειγμα: Έστω ότι η είσοδος S1 απαριθμεί προϊόντα τύπου A που περνάνε από μπροστά της αλλά και μια είσοδος S2 η οποία απαριθμεί τον ίδιο τύπο προϊόντων. Στην διαδικασία αυτή θέλουμε να ανάβει η ενδεικτική λυχνία L1 αν κατά την διάρκεια της έχει περάσει ο ίδιος αριθμός προϊόντων από τις δύο εισόδους, η ενδεικτική λυχνία L2 αν έχει περάσει περισσότερος αριθμός προϊόντων από την S1 και η ενδεικτική λυχνία L3 αν έχει περάσει περισσότερος αριθμός προϊόντων από την S2.

LD	I0.0
LD	I0.1
CTU	C1, +0
LD	I0.2
LD	I0.3
CTU	C2, +0
LDW=	C1, C2
=	Q0.0
LDW>	C1, C2
=	Q0.1
LDW<	C1, C2
=	Q0.2
END	

Επεξήγηση: Στο πρόγραμμα αυτό οι διακόπτες I0.0 και I0.1 αποτελούν την είσοδο S1 ενώ οι διακόπτες I0.2 και I0.3 αποτελούν την είσοδο S2. Συμπεραίνουμε λοιπόν ότι ο μετρητής C1

είναι ο μετρητής εισόδου S1 και ο C2 ο μετρητής εισόδου S2. Αν $C1=C2$ τότε ενεργοποιείται η λυχνία L1, αν $C1>C2$ τότε ενεργοποιείται η λυχνία L2 και αν $C1<C2$ τότε ενεργοποιείται η λυχνία L3.

Παράδειγμα: Σε μια διαδικασία ελέγχου η οποία χρονομετρείται για 10 sec ζητάμε μετά το πέρας 3 sec να γίνεται ενεργοποίηση του relay K1 και μετά το πέρας 6 sec να γίνεται ενεργοποίηση και του relay K2. Τα δύο αυτά relay να παραμένουν ενεργοποιημένα μέχρι την επανάληψη της διαδικασίας αυτής.

```
LD      I0.0
A       I0.0
TON    T36, +100
LDW=   T36, +30
S      Q0.0, 1
LDW=   T36, +60
S      Q0.1, 1
END
```

Επεξήγηση: Στο πρόγραμμα αυτό πατώντας τον διακόπτη I0.0 ξεκινά να μετράει ο timer για 10 δευτερόλεπτα. Όταν αυτός γίνει ίσος με 3 δευτερόλεπτα ενεργοποιείται η έξοδος Q0.0 με την εντολή S Q0.0, 1 ενώ όταν γίνει ίσος με 6 δευτερόλεπτα ενεργοποιείται η έξοδος Q0.1. [35]

3.10 Βοηθητικές μνήμες

Πολλές φορές κατά την εκτέλεση ενός προγράμματος καλούμαστε να επαναλάβουμε τμήματα του κώδικα για να εκτελέσουμε κάποιες διαδικασίες. Ένας τρόπος είναι να γράψουμε τον επαναλαμβανόμενο κώδικα τόσες φορές όσες χρειαζόμαστε. Αυτό όμως στοιχίζει σε χρόνο και σε μνήμη προγράμματος. Η ενδεδειγμένη λύση είναι η χρήση βοηθητικών. Καταγράφεται μια φορά η λογική, αποθηκεύεται σε ένα βοηθητικό και το βοηθητικό αυτό το χρησιμοποιούμε όσες φορές και σε όποιο σημείο του προγράμματος μας θέλουμε. Τα βοηθητικά παίζουν το ρόλο των βοηθητικών ρελέ στον κλασσικό αυτοματισμό. Τα βοηθητικά ρελέ είναι πολυάριθμες, φθηνές “έξοδοι” του προγραμματιζόμενου λογικού ελεγκτή που δεν εμφανίζονται στην έξοδο του και δεν έχουμε πρόσβαση σε αυτές. Συμβολίζονται με το γράμμα M π.χ. M0.0, και χρησιμοποιούνται για την αποθήκευση ενδιάμεσων αποτελεσμάτων. [35]

3.11 Συσσωρευτές (ACCUMULATORS)

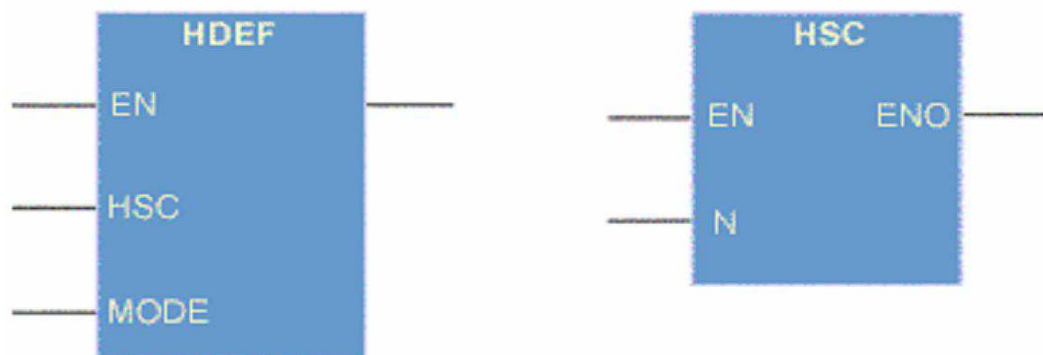
Οι συσσωρευτές είναι στοιχεία ανάγνωσης / εγγραφής που μπορούν να χρησιμοποιηθούν σαν μνήμη. Μπορούμε για παράδειγμα να χρησιμοποιήσουμε συσσωρευτές για να περάσουμε παραμέτρους από και προς τις υπορουτίνες, αλλά και να αποθηκεύσουμε ενδιάμεσες τιμές που χρησιμοποιούνται στον υπολογισμό. Το S7-200 μας παρέχει τέσσερις 32-bit συσσωρευτές (ACO, AC1, AC2, AC3). Μπορούμε να απευθυνθούμε στα δεδομένα στους συσσωρευτές σαν bits, bytes, words η double words. Το μέγεθος των δεδομένων που χρησιμοποιείται καθορίζεται από την εντολή που χρησιμοποιείται για την πρόσβαση στον συσσωρευτή.

3.12 Διακοπές κύκλου προγράμματος

Όπως αναφέραμε και πιο πριν, οι προγραμματιζόμενοι λογικοί ελεγκτές εκτελούν το πρόγραμμα τους κυκλικά. Ο χρόνος κάθε κύκλου εξαρτάται από το μέγεθος του προγράμματος, τον αριθμό εισόδων και εξόδων και τον όγκο των επικοινωνιών που πρέπει να υλοποιηθούν. Ο κύκλος ξεκινά με έλεγχο της κατάστασης των εισόδων. Στη συνέχεια εκτελείται το πρόγραμμα με βάση τη λογική που περιέχει και την κατάσταση των εισόδων. Όταν ολοκληρωθεί η εκτέλεση του προγράμματος η CPU εκτελεί ορισμένες εσωτερικές διαγνωστικές λειτουργίες και πραγματοποιεί τις επικοινωνίες. Στο τέλος, ενημερώνονται οι καταστάσεις των εξόδων και ο κύκλος ξεκινά από την αρχή. Υπάρχουν όμως κάποιες περιπτώσεις που ο προγραμματιζόμενος λογικός ελεγκτής πρέπει να αντιδράσει άμεσα σε κάποια γεγονότα που συμβαίνουν πριν να ολοκληρωθεί ο κύκλος του PLC. Αυτό γίνεται με τις λεγόμενες “διακοπές” του προγράμματος (interrupts) και με εντολές ειδικού τύπου (high speed instructions). [35]

3.13 Γρήγοροι απαριθμητές

Οι γρήγοροι απαριθμητές είναι σε θέση να μετρήσουν ψηφιακά σήματα που αλλάζουν κατάσταση (0-1) πολύ γρήγορα (παλμούς). Οι γρήγοροι απαριθμητές του S7-200 μπορούν να μετρήσουν παλμούς μέγιστης συχνότητας από 10 έως 50 KHz ανάλογα με το μοντέλο, τον τρόπο λειτουργίας κ.α. Στη γλώσσα Ladder οι γρήγοροι απαριθμητές αναπαρίστανται με ορθογώνια. Οι γρήγοροι απαριθμητές έχουν δώδεκα διαφορετικούς τρόπους λειτουργίας. Μαζί με την εντολή του γρήγορου απαριθμητή υπάρχει πάντα και μια συνοδευτική εντολή που στη Ladder αναπαρίσταται επίσης με ορθογώνιο, ονομάζεται “definitionbox” και αυτό που κάνει είναι να καθορίζει τον τρόπο λειτουργίας του γρήγορου απαριθμητή.[35]



Σχ.3.24 Αναπαράσταση γρήγορου απαριθμητή σε γλώσσα Ladder.

3.14 ACCUMULATORS (Συσσωρευτές)

Οι συσσωρευτές είναι στοιχεία ανάγνωσης / εγγραφής που μπορούν να χρησιμοποιηθούν σαν μνήμη. Μπορούμε για παράδειγμα να χρησιμοποιήσουμε συσσωρευτές για να περάσουμε παραμέτρους από και προς τις υπορουτίνες, αλλά και να αποθηκεύσουμε ενδιάμεσες τιμές που χρησιμοποιούνται στον υπολογισμό. Το S7-200 μας παρέχει τέσσερις 32-bit συσσωρευτές (ACO, AC1, AC2, AC3). Μπορούμε να απευθυνθούμε στα δεδομένα στους συσσωρευτές σαν bits, bytes, words η double words. Το μέγεθος των δεδομένων που χρησιμοποιείται καθορίζεται από την εντολή που χρησιμοποιείται για την πρόσβαση στον Συσσωρευτή.

3.15 Προγραμματιστικά χαρακτηριστικά και ονοματολογία των στοιχείων ενός προγραμματιζόμενου λογικού ελεγκτή.

Όταν ξεκινάμε να μελετάμε, πως θα προγραμματίσουμε ένα PLC, πρέπει να γνωρίζουμε:

- Πόσες εισόδους έχει, πως τις ονομάζουμε και πως τις αναγνωρίζουμε.

Οι εισοδοί σχεδόν σε όλα τα PLC χαρακτηρίζονται με το γράμμα I (Input). Το κάθε γράμμα συνοδεύεται από δύο αριθμούς που χωρίζονται από μια τελεία (π.χ. είσοδος I1.3).

Αν χρησιμοποιείτε συσκευή με 16 εισόδους θα δείτε ότι η αρίθμηση δεν είναι από I1 έως I16. Αντίθετα χωρίζονται σε δύο ομάδες των οκτώ και ο πρώτος αριθμός μετά το γράμμα μας δίνει τον αριθμό της οκτάδας. Κάθε οκτάδα αντιστοιχεί σε ένα byte. Επίσης παρατηρήστε ότι η αρίθμηση ξεκινά από το μηδέν. Έτσι, π.χ. η είσοδος I1.3 αντιστοιχεί στην τέταρτη είσοδο της δεύτερης ομάδας εισόδων, δηλαδή στην $8+4=12$ είσοδο, όπως φαίνεται στο σχήμα.



Σχ.3.25 Συμβολισμός εισόδων

- Πόσες εξόδους έχει, πως τις ονομάζουμε και πως τις αναγνωρίζουμε.

Τα ίδια, που ισχύουν για τις εισόδους, ισχύουν και για τις εξόδους. Το γράμμα με το οποίο χαρακτηρίζονται οι έξοδοι στα διάφορα PLC είναι συνήθως το Q ή το O (Output). Για τους αριθμούς, που ακολουθούν το γράμμα, ισχύει ότι και για τις εισόδους.

- Πόσες βοηθητικές μνήμες έχει και πως τις ονομάζουμε.

Στα διάφορα PLC θα τις συναντήσουμε με το όνομα Markers ή Flags. Πρόκειται για θέσεις μνήμης, στις οποίες αποθηκεύονται ενδιάμεσες λογικές καταστάσεις και πληροφορίες. Όπως ισχύει για τις εισόδους και τις εξόδους, χαρακτηρίζονται με ένα γράμμα ακολουθούμενο από έναν αριθμό ή δύο αριθμούς, που χωρίζονται με τελεία. Το γράμμα στα διάφορα PLC είναι το M (Marker) ή το F (Flag). Ο αριθμός τους συνήθως είναι πάνω από 1000 και είναι οργανωμένες σε οκτάδες.

- Τις ειδικές συναρτήσεις του PLC.

Πρέπει να γνωρίζουμε ποιες είναι, πως ονομάζονται, πως τις χειρίζεται το συγκεκριμένο PLC και πόσες από την κάθε μία διαθέτει. Οι ειδικές συναρτήσεις κατά σειρά σπουδαιότητας είναι:

- Τα χρονικά.
- Οι απαριθμητές.
- Οι συγκριτές.
- Οι γεννήτριες παλμοσειρών.
- Ο μετρητής πραγματικού χρόνου.

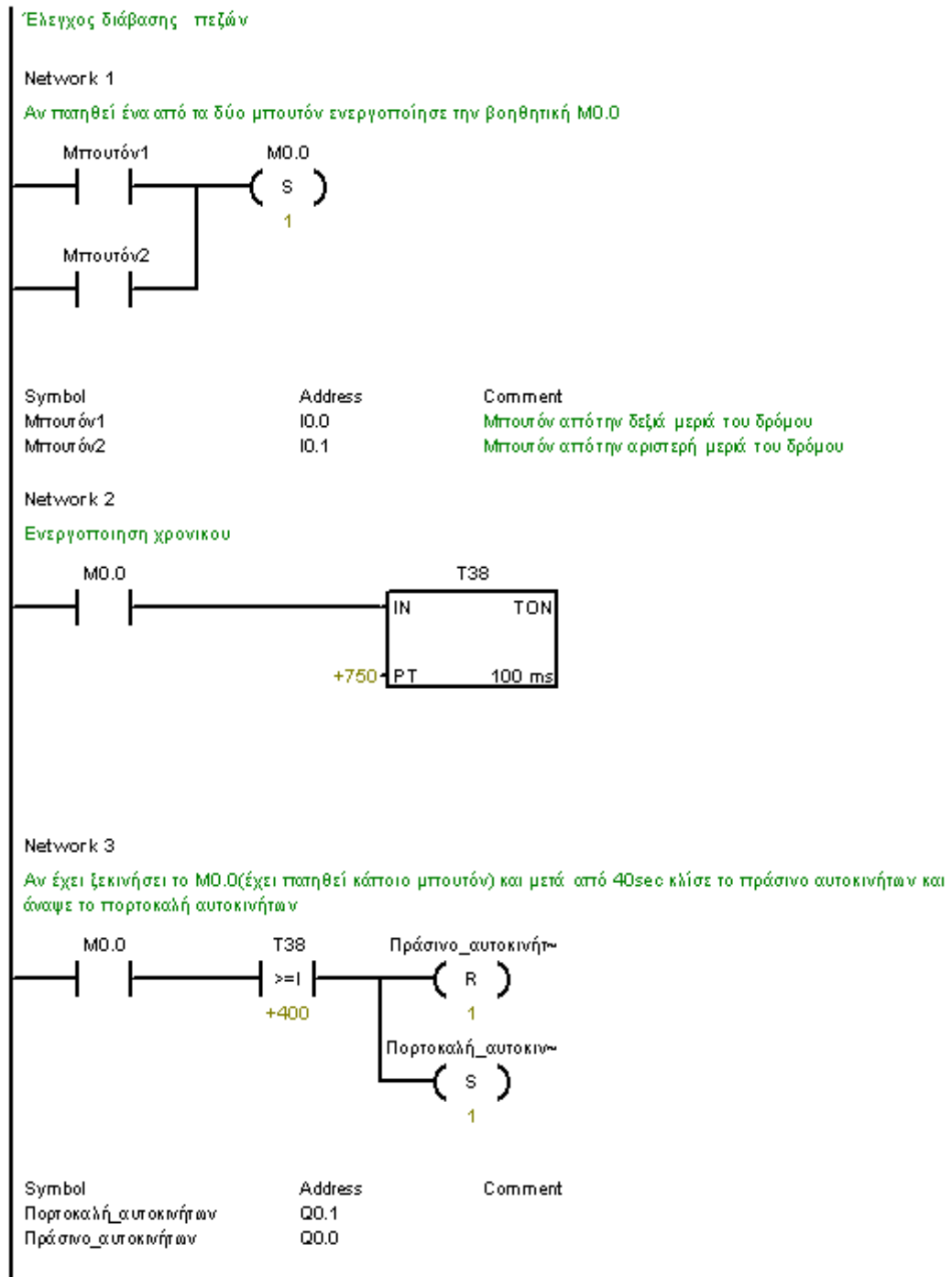
Όλα τα παραπάνω στοιχεία λέμε ότι αποτελούν το προγραμματιστικό μοντέλο ενός PLC. Για να ξεκινήσουμε τον προγραμματισμό πρέπει να γνωρίζουμε το προγραμματιστικό μοντέλο του συγκεκριμένου PLC, που διαθέτουμε.[35]

4 Διάφορα παραδείγματα

4.1 Έλεγχος διάβασης πεζών

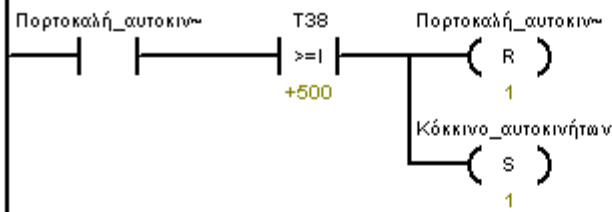
Υπάρχουν δύο μπουτόν ενεργοποίησης ένα από την δεξιά μεριά του δρόμου και ένα από την αριστερή. Για να ανάψει πράσινο για τους πεζούς πρέπει να πιεστή το ανάλογο μπουτόν. Όταν πιεστή ένα από τα δύο μπουτόν έχουμε: μετά από 40sec ανάβει πορτοκαλή για τα αυτοκίνητα και μετά από 50 sec ανάβει κόκκινο για τα αυτοκίνητα, μετά από 55 sec ανάβει πράσινο για τους πεζούς για 15sec και μετά γίνεται κόκκινο, μόλις περάσουν 5 sec από την στιγμή που άναψε κόκκινο για τους πεζούς ανάβει πράσινο για τα αυτοκίνητα.

Symbol	Address	Comment
Μπουτόν1	I0.0	Μπουτόν από την δεξιά μεριά του δρόμου
Μπουτόν2	I0.1	Μπουτόν από την αριστερή μεριά του δρόμου
Πράσινο_αυτοκινήτων	Q0.0	
Πορτοκαλή_αυτοκινήτων	Q0.1	
Κόκκινο_αυτοκινήτων	Q0.2	
Πράσινο_Πεζών	Q0.3	
Κόκκινο_Πεζών	Q0.4	



Network 4

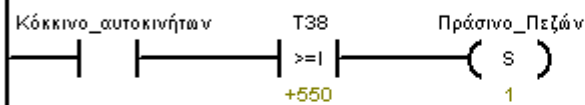
Αν έχει ανάψει πορτοκαλί και σε 10sec μετά το άναμμα του κλίσει το πορτοκαλή αυτοκινήτων και άναψε το κόκκινο αυτοκινήτων



Symbol	Address	Comment
Κόκκινο_αυτοκινήτων	Q0.2	
Πορτοκαλή_αυτοκινήτων	Q0.1	

Network 5

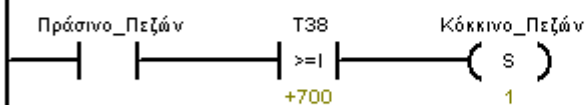
5sec μετά το άναμμα του κόκκινου αυτοκινήτων άναψε πράσινο για τους πεζούς



Symbol	Address	Comment
Κόκκινο_αυτοκινήτων	Q0.2	
Πράσινο_Πεζών	Q0.3	

Network 6

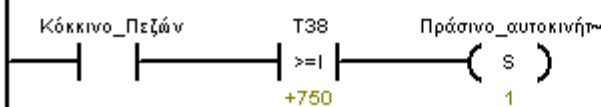
15sec μετά το άναμμα του πράσινου πεζών άναψε κόκκινο πεζών



Symbol	Address	Comment
Κόκκινο_Πεζών	Q0.4	
Πράσινο_Πεζών	Q0.3	

Network 7

5sec μετά το άναμμα του κόκκινου πεζών άναψε πράσινο αυτοκινήτων



Symbol	Address	Comment
Κόκκινο_Πεζών	Q0.4	
Πράσινο_αυτοκινήτων	Q0.0	

4.2 Σύστημα συναγερμού

Σε ένα σύστημα συναγερμού που ελέγχετε από ένα PLC έχουμε μία σειρά αισθητήρων κινήσες που όσο δεν ανιχνεύουν κίνηση δίνουν λογικό 0 ενώ όταν ανιχνεύσουν κίνηση λογικό 1, μία σειρά αισθητήρων παραβιάσεις που αποτελούνται από δύο μέρει αν διακοπή η επαφή τους δίνουν λογικό 0 αλλιώς λογικό 1. Ο συναγερμός ενεργοποιείτε με το πάτημα ενός διακόπτη, 30 sec μετά το πάτημα του διακόπτη ενεργοποιείτε η λυχνία που δηλώνει ότι ο συναγερμός είναι ενεργός τα φώτα ασφαλείς. Ο συναγερμός απενεργοποιείτε με το πάτημα του ίδιου διακόπτη. Αν υπάρξει αληγή στους αισθητήρες μετά από 30 sec από τη παραβίαση ενεργοποιείτε μία σειρήνα και ένας φάρος όπου απενεργοποιούνται με το πάτημα του διακόπτη.

Symbol	Address	Comment
Διακόπτης	I0.0	διακόπτης (ON/FF)
Αισθητήρες2	I0.2	αισθητήρες παραβίασης
Αισθητήρες1	I0.1	αισθητήρες παραβίασης
Λυχνία	Q0.0	ενδειξη λειτουργίας συναγερμού
Φώτα	Q0.1	ανάβουν όταν λειτουργι ο συναγερμός
Φάρος	Q0.2	ανάβει όταν υπάρχει παραβίαση
Σειρήνα	Q0.3	ανάβει όταν υπάρχει παραβίαση

PROGRAM COMMENTS

Network 1

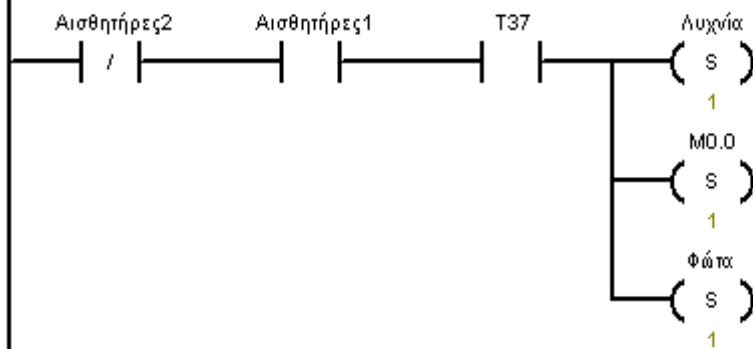
Από την στιγμή που θα ενεργοποιηθεί ο διακόπτης περίμενε 30 sec



Symbol	Address	Comment
Διακόπτης	I0.0	διακόπτης (ON/FF)

Network 2

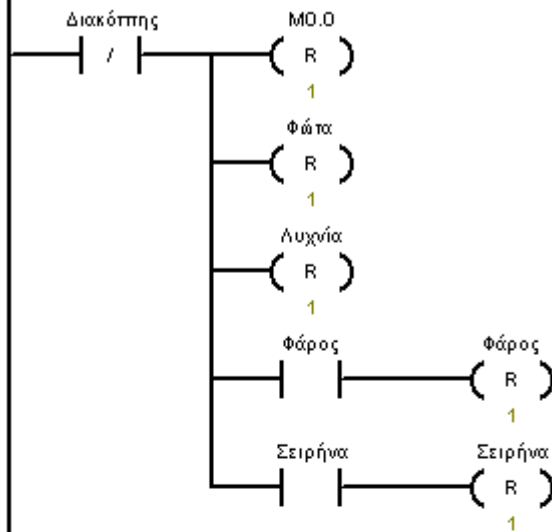
Μόλις περάσουν τα 30 sec και δεν υπάρχει παραβίαση στους αισθητήρες ενεργοποίησε τα φάτα ασφαλείας ,την ένδειξη ενεργοποίησης του συναγερμού και την βοηθητική μεταβλητή M0.0.



Symbol	Address	Comment
Αισθητήρες1	I0.1	αισθητήρες παραβίασης
Αισθητήρες2	I0.2	αισθητήρες παραβίασης
Λυχνία	Q0.0	ένδειξη λειτουργίας συναγερμού
Φάτα	Q0.1	ανάβουν όταν λειτουργι ο συναγερμός

Network 3

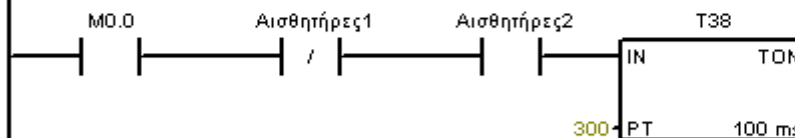
Όταν κλίσει ο διακόπτης κλίσει τα φώτα ασφαλείας την ένδειξη λειτουργίας του συναγερμού και αν έχουν ενεργοποιηθεί, τον φάρο και την σειρήνα



Symbol	Address	Comment
Διακόπτης	I0.0	διακόπτης (ON/FF)
Λυχνία	Q0.0	ένδειξη λειτουργίας συναγερμού
Σειρήνα	Q0.3	ανάβει όταν υπάρχει παραβίαση
Φάρος	Q0.2	ανάβει όταν υπάρχει παραβίαση
Φώτα	Q0.1	ανάβουν όταν λειτουργεί ο συναγερμός

Network 4

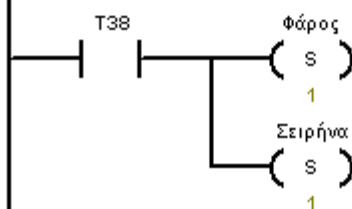
Αν έχει ενεργοποιηθεί ο συναγερμός και κάποιο αισθητήριο δώσει σήμα παραβίασης περίμενε 30 sec



Symbol	Address	Comment
Αισθητήρες1	I0.1	αισθητήρες παραβίασης
Αισθητήρες2	I0.2	αισθητήρες παραβίασης

Network 5

Μόλις περάσουν τα 30 sec ενεργοποίησε τον φάρο και την σειρήνα



4.3 Ταινίες μεταφοράς υλικών

Το σύστημα αποτελείται από τρεις ταινίες μεταφοράς την T1, T2 και T3 όπου κινούνται από ηλεκτροκινητήρες τους M1, M2 και M3 αντίστοιχα.

Πατώντας το μπουτόν S1 ξεκινάει η ταινία T1 ενώ πατώντας το μπουτόν S2 ξεκινάει η T2.

Για να υπάρχει ομαλή ροή προς την ταινία T3 πρέπει να βρίσκεται σε λειτουργία μόνο μια από τις T1, T2. Η ταινία T3 ξεκινά όταν τεθεί σε λειτουργία μια από τις ταινίες T1, T2. Για να σταματήσει να λειτουργεί η ταινία T1 και η T2 πιέζεται το μπουτόν S3 Προκειμένου να μεταφερθούν στη θέση αποθήκευσης τα υλικά που βρίσκονται επάνω στις ταινίες T1 και T3, οι T1 και T3 σταματούν αντίστοιχα 30 sec και 70 sec μετά από το πάτημα του S3. Αντίστοιχα για τα υλικά που βρίσκονται επάνω στην T2 και T3 οι ταινίες σταματούν 30 sec και 70 sec με την πίεση του S3.

Όταν σταματήσει η ταινία T1 ή η ταινία T2 λόγω υπερφόρτισης του κινητήρα που την κινεί προκειμένου να μεταφερθούν στη θέση αποθήκευσης τα υλικά που βρίσκονται επάνω στην ταινία T3, η T3 συνεχίζει να λειτουργεί ακόμη για 40 sec και στη συνέχεια σταματά.

Όταν σταματήσει η ταινία T3 λόγω υπερφόρτισης του κινητήρα που την κινεί, σταματά αμέσως και η ταινία T1 ή η ταινία T2 .

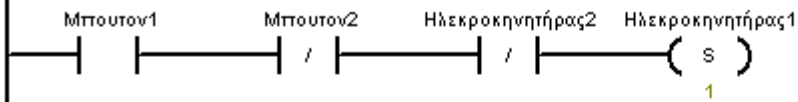
Η υπερφόρτιση των κινητήρων M1, M2 και M3 δηλώνεται από τις λυχνίες ένδειξης H4, H5 και H6 αντίστοιχα, οι οποίες αναβοσβήνουν με συχνότητα 2 Hz. Το αναβόσβημα μιας λυχνίας σταματά με το πάτημα του S4 .

Symbol	Address	Comment
Μπουτον1	I0.0	Μπουτον εκίνησης της T1
Μπουτον2	I0.1	Μπουτον εκίνησης της T2
Μπουτον3	I0.2	Σταματά την T1 ή την T2
Μπουτον4	I0.3	Σβήνει τις λυχνίες
Θερμικό1	I0.5	
Θερμικό2	I0.6	
Ηλεκτροκινητήρας1	Q0.0	Ηλεκτροκινητήρας της T1 ταινίας
Ηλεκτροκινητήρας2	Q0.1	Ηλεκτροκινητήρας της T2 ταινίας
Λυχνία1	Q0.3	Υπερφόρτιση του κινητήρα M1
Λυχνία2	Q0.4	Υπερφόρτιση του κινητήρα M2
Λυχνία3	Q0.5	Υπερφόρτιση του κινητήρα M3
Ηλεκτροκινητήρας3	Q0.2	Ηλεκτροκινητήρας της T3 ταινίας
Θερμικό3	I0.7	

Ταινίες μεταφοράς υλικών

Network 1

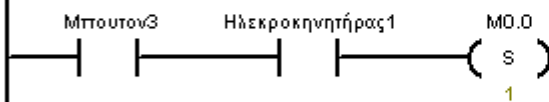
Εάν έχει πατηθεί το μπουτόν για να εκινήθει η T1 και δεν έχει πατηθεί το μπουτόν 2 όπως επίσης δεν λειτουργεί η T2 ταινία ξεκίνα την T1



Symbol	Address	Comment
Μπουτον1	I0.0	Μπουτον εκίνησης της T1
Μπουτον2	I0.1	Μπουτον εκίνησης της T2
Ηλεκτροκνητήρας1	Q0.0	Ηλεκτροκνητήρας της T1 ταινίας
Ηλεκτροκνητήρας2	Q0.1	Ηλεκτροκνητήρας της T2 ταινίας

Network 2

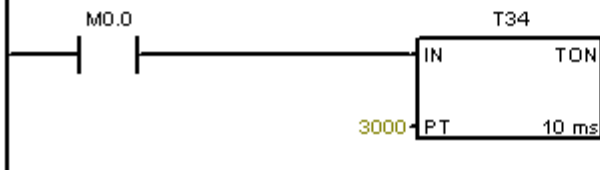
Εάν πατηθεί το μπουτόν 3 και λειτουργεί η T1 ξεκίνα την βοηθητική M0.0



Symbol	Address	Comment
Μπουτον3	I0.2	Σταματά την T1 ή την T2
Ηλεκτροκνητήρας1	Q0.0	Ηλεκτροκνητήρας της T1 ταινίας

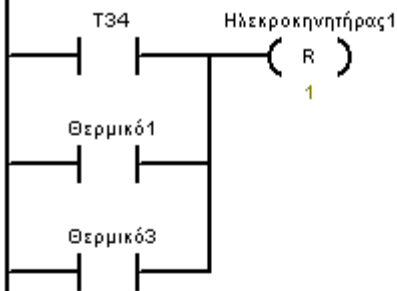
Network 3

Αν έχει ξεκινήσει η βοηθητική τότε περίμενε 30 sec



Network 4

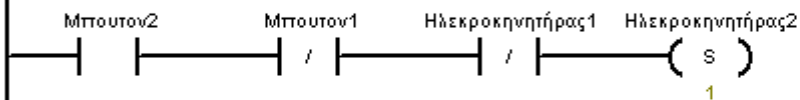
Όταν περάσουν τα 30 sec ή ανάψει το θερμικό 1 ή το θερμικό3 τότε κλίσε τη ταινία T1



Symbol	Address	Comment
Ηλεκτροκνητήρας1	Q0.0	Ηλεκτροκνητήρας της T1 ταινίας
Θερμικό1	I0.5	
Θερμικό3	I0.7	

Network 5

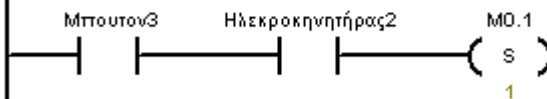
Εάν έχει πατηθεί το μπουτόν για να εκινήθει η T2 και δεν έχει πατηθεί το μπουτόν 1 πώς επίσης δεν λειτουργεί η T1 ταινία ξεκίνα την T2



Symbol	Address	Comment
Μπουτον1	I0.0	Μπουτον εκίνησης της T1
Μπουτον2	I0.1	Μπουτον εκίνησης της T2
Ηλεκτροκνητήρας1	Q0.0	Ηλεκτροκνητήρας της T1 ταινίας
Ηλεκτροκνητήρας2	Q0.1	Ηλεκτροκνητήρας της T2 ταινίας

Network 6

Εάν πατηθεί το μπουτόν 3 και λειτουργεί η T2 ξεκίνα την βοηθητική M0.1



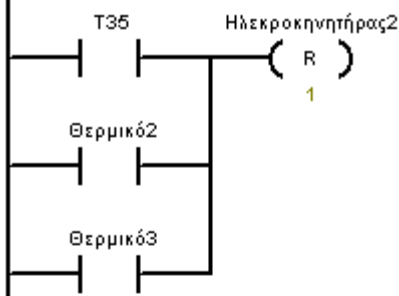
Symbol	Address	Comment
Μπουτον3	I0.2	Σταματά την T1 ή την T2
Ηλεκτροκνητήρας2	Q0.1	Ηλεκτροκνητήρας της T2 ταινίας

Network 7

Αν έχει ξεκινήσει η βοηθητική τότε περίμενε 30 sec

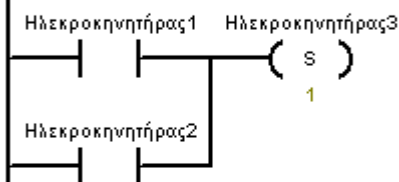


Network 8
Όταν περάσουν τα 30 sec ή ανάψει το θερμικό 2 ή το θερμικό 3 τότε κλίσε τη ταινία T2



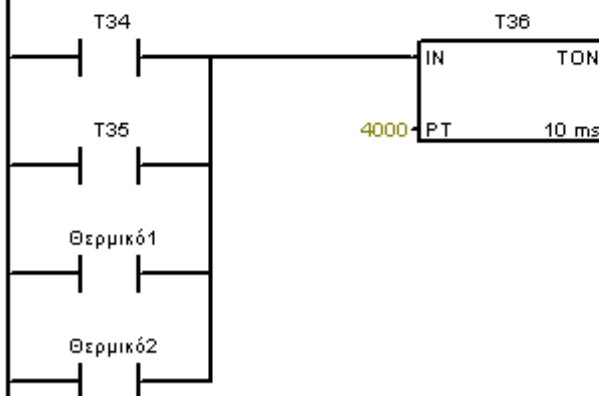
Symbol	Address	Comment
Ηλεκτροκνητήρας2	Q0.1	Ηλεκτροκνητήρας της T2 ταινίας
Θερμικό2	I0.6	
Θερμικό3	I0.7	

Network 9
Αν ξεκινήσει η T1 ή T2 τότε ξεκίνα και την T3



Symbol	Address	Comment
Ηλεκτροκνητήρας1	Q0.0	Ηλεκτροκνητήρας της T1 ταινίας
Ηλεκτροκνητήρας2	Q0.1	Ηλεκτροκνητήρας της T2 ταινίας
Ηλεκτροκνητήρας3	Q0.2	Ηλεκτροκνητήρας της T3 ταινίας

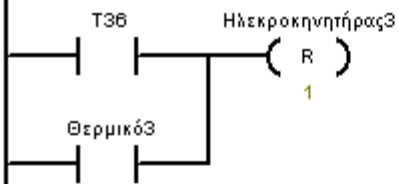
Network 10
Αν έχει σταματήσει η T1 ή T2 τότε περίμενε 40sec



Symbol	Address	Comment
Θερμικό1	I0.5	
Θερμικό2	I0.6	

Network 11

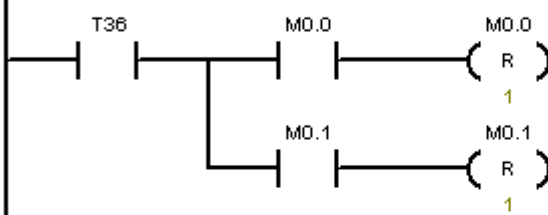
Όταν περάσουν 40 sec ή ανάψει το θερμικό 3 σταμάτα και την T3



Symbol	Address	Comment
Ηλεκτροκνητήρας3	Q0.2	Ηλεκτροκνητήρας της T3 τακίας
Θερμικό3	I0.7	

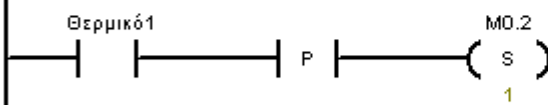
Network 12

Αν σταματήσει η T3 σταμάτα την βοηθητική



Network 13

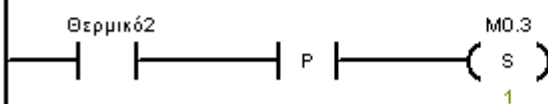
Αν ανάψει το θερμικό ξεκίνα την βοηθητική



Symbol	Address	Comment
Θερμικό1	I0.5	

Network 14

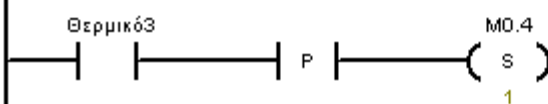
Αν ανάψει το θερμικό ξεκίνα την βοηθητική



Symbol	Address	Comment
Θερμικό2	I0.6	

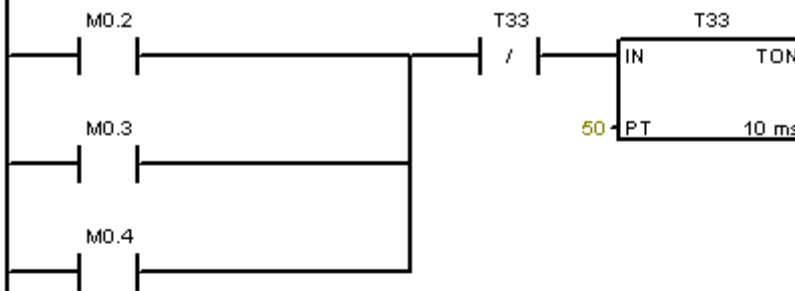
Network 15

Αν ανάψει το θερμικό ξεκίνα την βοηθητική



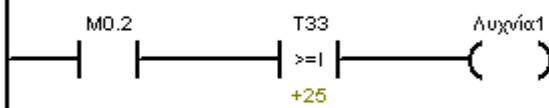
Network 16

Αν έχει ανάψει η βοηθητική και δεν έχουν περάσει 5 sec περιμένε 5 sec



Network 17

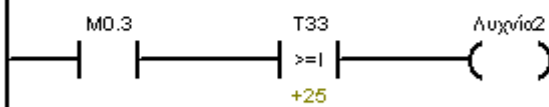
Αν έχει ξεκινήσει η βοηθητική και έχουν περάσει 24 sec άναψε την λυχνία



Symbol	Address	Comment
Λυχνία1	Q0.3	Υπερφόρτιση του κινητήρα M1

Network 18

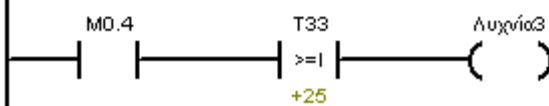
Αν έχει ξεκινήσει η βοηθητική και έχουν περάσει 24 sec άναψε την λυχνία



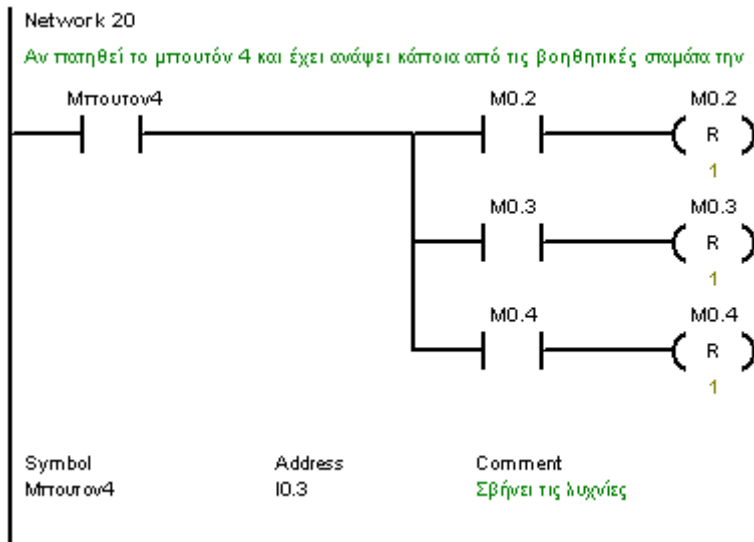
Symbol	Address	Comment
Λυχνία2	Q0.4	Υπερφόρτιση του κινητήρα M2

Network 19

Αν έχει ξεκινήσει η βοηθητική και έχουν περάσει 24 sec άναψε την λυχνία



Symbol	Address	Comment
Λυχνία3	Q0.5	Υπερφόρτιση του κινητήρα M3



4.4 Χώρος στάθμευσης οχημάτων

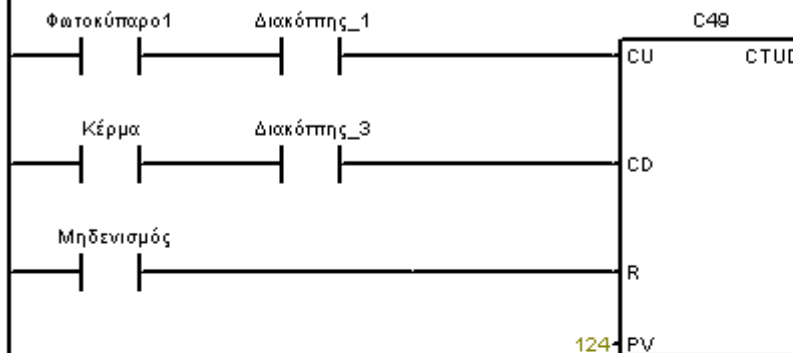
Σε ένα χώρο στάθμευσης υπάρχουν, ένα φωτοκύτταρο στην είσοδο και μια μπάρα εισόδου οχημάτων καθώς ένα φωτοκύτταρο στην έξοδο και δύο μπάρες εξόδου σε σειρά ανάμεσα από τις δύο μπάρες υπάρχει ένας κερματοδέκτης .Κάθε φορά που ανιχνεύει ένα φωτοκύτταρο κάποιο όχημα στέλνει λογικό '1'. Τα οχήματα που εισέρχονται στο χώρο καταγράφονται και όταν φτάσουν τα 124 που είναι ο μέγιστος αριθμός χωρητικότητας οχημάτων ανάβει ενδεικτική λυχνία. Ο μέγιστος αριθμός εξαρτάται και από τα οχήματα που φεύγουν απ' το χώρο στάθμευσης και καταγράφονται από το φωτοκύτταρο εξόδου. Υπάρχει και ένα κουμπί που μηδενίζει τον απαριθμητή. Η είσοδος ελέγχεται αυτόματα μέσω μιας μπάρας που ανοίγει κάθε φορά που ένα αυτοκίνητο εντοπιστεί από το φωτοκύτταρο και υπάρχει κενή θέση Όταν ανοίγει τελείως η μπάρα ενεργοποιείται ένας τερματικός διακόπτης. Μετά από 15sec από την ενεργοποίηση του η μπάρα κλείνει αυτόματα. Στην έξοδο υπάρχουν δύο μπάρες ανοίγοντας η πρώτη ενεργοποιεί και αυτή έναν τερματικό διακόπτη, 15 sec μετά την ενεργοποίηση του η μπάρα κλίνει, για να ανοίξει η δεύτερη μπάρα πρέπει να τοποθετηθεί κέρμα στον κερματοδέκτη έπειτα η μπάρα ανοίγει και 15 sec μετά την ενεργοποίηση του τερματικού διακόπτη της μπάρας η μπάρα κλίνει.

Symbol	Address	Comment
Διακόπτης_1	IO.5	διακόπτης μπάρας εισόδου
ΑΜΕισόδου	Q0.1	Άνοιγμα Μπάρας Εισόδου
ΚΜΕισόδου	Q0.2	Κλείσιμο Μπάρας Εισόδου
Φωτοκύτταρο1	IO.1	φωτοκύτταρο εισόδου
Κέρμα	IO.4	τοποθέτηση κέρματος στο μηχανήμα
Λυχνία_πληρότητας	Q0.0	ανάβει όταν γεμίσει το παρκινγκ
Μηδενισμός	IO.2	
Διακόπτης_2	IO.3	διακόπτης μπάρας εξόδου1
Φωτοκύτταρο_2	IO.0	φωτοκύτταρο εξόδου
ΑΜΕξόδου1	Q0.3	Άνοιγμα Πρώτης Μπάρας Εξόδου
ΚΜΕξόδου1	Q0.4	Κλείσιμο Πρώτης Μπάρας Εξόδου
ΑΜΕξόδου2	Q0.5	Άνοιγμα Δεύτερης Μπάρας Εξόδου
ΚΜΕξόδου2	Q0.6	Κλείσιμο Δεύτερης Μπάρας Εξόδου
Διακόπτης_3	IO.6	Διακόπτης μπάρας εξόδου2

Χώρος στάθμευσης οχημάτων

Network 1

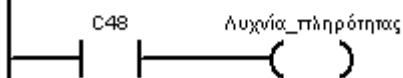
Όταν έχουμε είσοδο οχήματος(το φωτοκύπαρο εισόδου ανίχνευση όχημα και ανοίξει η μπάρα εισόδου) αύξησε των αριθμό των οχημάτων κατά 1. Όταν έχουμε έξοδο οχήματος (γίνει εισαγωγή κέρματος και η μπάρα εξόδου ανοίξει) μείωσε των αριθμό οχημάτων κατά 1. Αν πατηθεί το κουμπι μηδενισμού μηδένισε των μετρητή.



Symbol	Address	Comment
Διακόπτης_1	I0.5	διακόπτης μπάρας εισόδου
Διακόπτης_3	I0.6	Διακόπτης μπάρας εξόδου2
Κέρμα	I0.4	τοποθέτηση κέρματος στο μηχανήμα
Μηδενισμός	I0.2	
Φωτοκύπαρο1	I0.1	φωτοκύπαρο εισόδου

Network 2

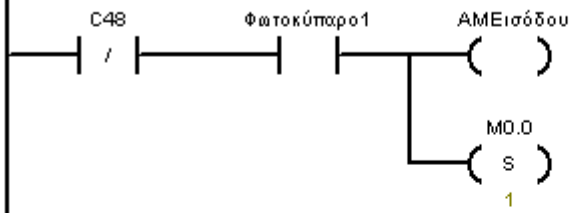
Αν έχουμε 124 οχήματα μέσα στο παρκινγκ άναψε την λυχνία πληρότητας



Symbol	Address	Comment
Λυχνία_πληρότητας	Q0.0	ανάβει όταν γεμίσει το παρκινγκ

Network 3

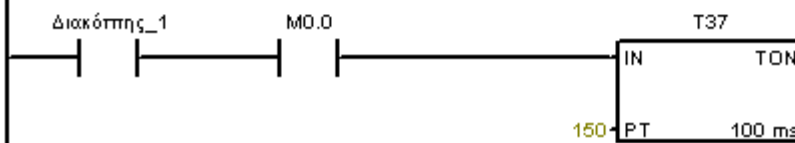
Εάν δεν έχει γεμίσει το παρκινγκ και το φωτοκύπαρο εισόδου αφήνευση όχημα ανοίξε την μπάρα εισόδου



Symbol	Address	Comment
ΑΜΕισόδου	Q0.1	Άνοιγμα Μπάρας Εισόδου
Φωτοκύπαρο1	I0.1	φωτοκύπαρο εισόδου

Network 4

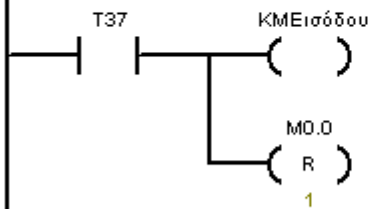
Όταν ο διακόπτης εισόδου πατηθεί(έχει ανοίξει η μπάρα εισόδου) περιμένε 15sec



Symbol	Address	Comment
Διακόπτης_1	I0.5	διακόπτης μπάρας εισόδου

Network 5

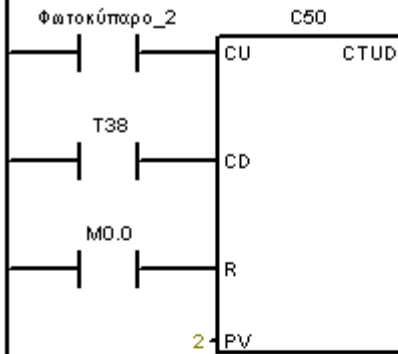
Μόλις περάσουν τα 15sec κλίσε την μπάρα εισόδου .



Symbol	Address	Comment
ΚΜΕισόδου	Q0.2	Κλίσιμο Μπάρας Εισόδου

Network 6

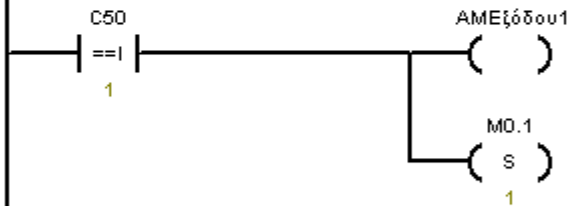
Αν ανοιχθεί ο όχημα το φωτοκύτταρο εξόδου αύξησε τον μετρητή κατά 1, αν αρχίζει να κλίνει η δεύτερη μπάρα εξόδου μείωσε των μετρητή κατά 1



Symbol	Address	Comment
Φωτοκύτταρο_2	I0.0	φωτοκύτταρο εξόδου

Network 7

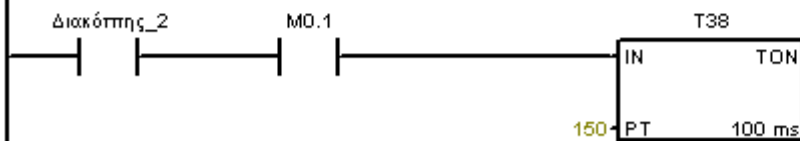
Όταν ο μετρητής προσέλευσης εξόδου είναι ένα (δεν υπάρχει άλλο όχημα πριν την δεύτερη μπάρα εξόδου) ανοίξει την πρώτη μπάρα εξόδου και μία βοηθητική μεταβλητή



Symbol	Address	Comment
AMEξόδου1	Q0.3	Άνοιγμα Πρότης Μπάρας Εξόδου

Network 8

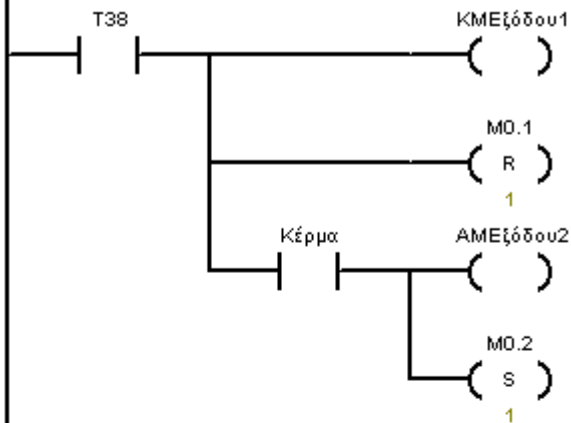
Όταν ανοίξει η πρώτη μπάρα εξόδου περίμενε 15 Sec



Symbol	Address	Comment
Διακόπτης_2	I0.3	διακόπτης μπάρας εξόδου1

Network 9

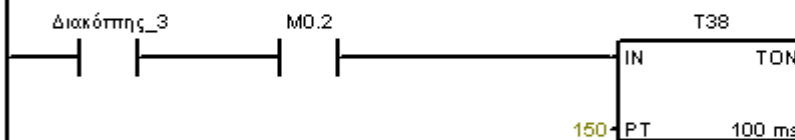
Όταν περάσουν τα 15sec κλίσε την πρώτη μπάρα εξόδου και αν γίνει εισαγωγή κέρματος κλίσε την βοηθητική άνοιξε την δεύτερη μπάρα εξόδου και μια νέα βοηθητική.



Symbol	Address	Comment
AMEξόδου2	Q0.5	Άνοιγμα Δεύτερης Μπάρας Εξόδου
KMEξόδου1	Q0.4	Κλείσιμο Πρώτης Μπάρας Εξόδου
Κέρμα	I0.4	τοποθέτηση κέρματος στο μηχανήμα

Network 10

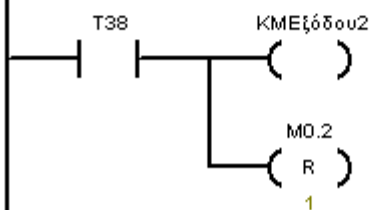
Αν ενεργοποιηθεί ο διακόπτης 3 και έχει ενεργοποιηθεί και το MO.2 (αν δηλαδή έχει ανοίξει η δεύτερη μπάρα εξόδου) τότε περιμένε 15 sec



Symbol	Address	Comment
Διακόπτης_3	I0.6	Διακόπτης μπάρας εξόδου2

Network 11

Όταν περάσουν τα 15 sec κλίσε την δεύτερη μπάρα εξόδου και το MO.2



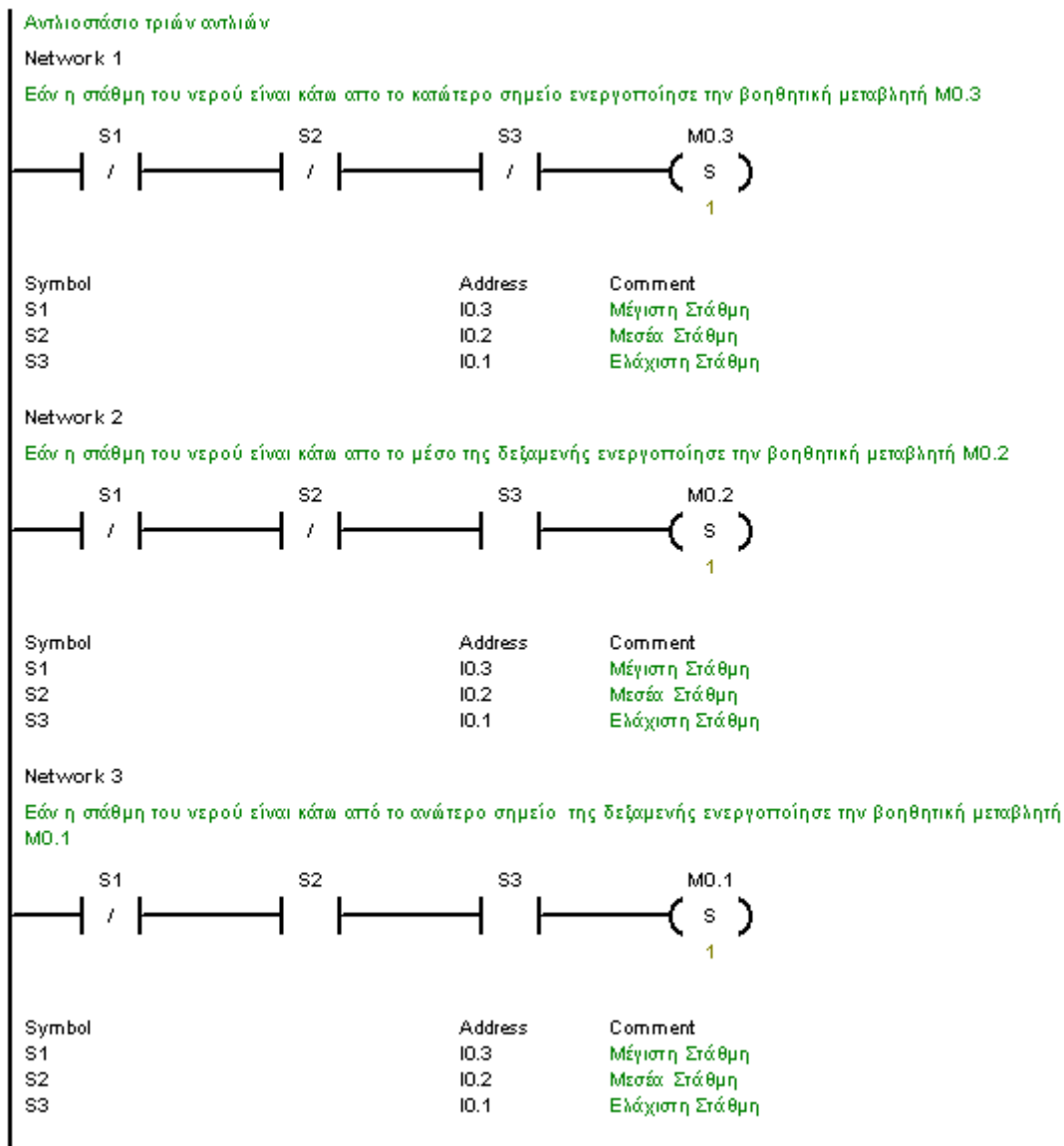
Symbol	Address	Comment
KMEξόδου2	Q0.6	Κλείσιμο Δεύτερης Μπάρας Εξόδου

4.5 Αντλιοστάσιο τριών αντλιών

Σε ένα αντλιοστάσιο έχουμε μία δεξαμενή με τρεις στάθμες μέτρησης ,ελάχιστη στάθμη την μεσαία στάθμη και την μέγιστη στάθμη επίσης έχουμε τρεις αντλίες. Ανάλογα με το ύψος του

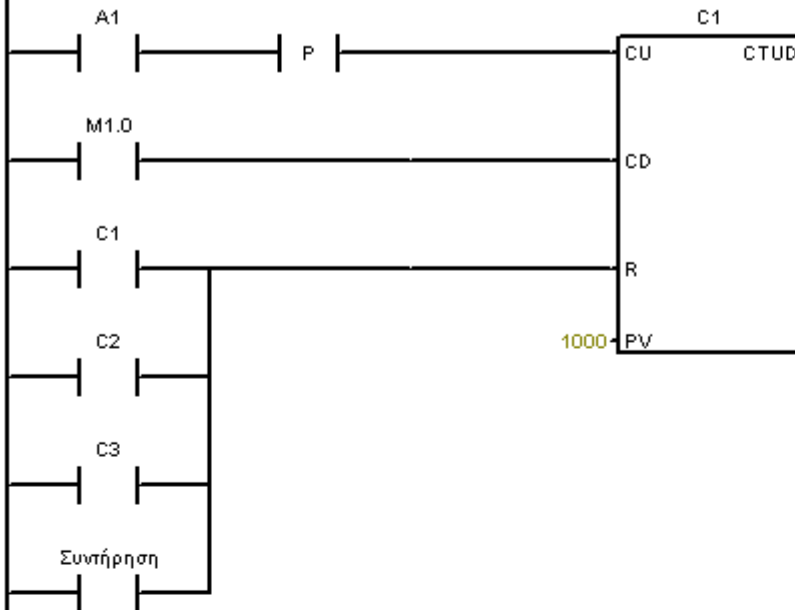
νερού έχουμε και αντίστοιχο αριθμό κινητήρων που δουλεύουν, αν στάθμη του νερού πέσει κάτω από την ελάχιστη στάθμη έχουμε τρεις κινητήρες για κάτω από την μεσαία δύο και για κάτω από την μέγιστη 1 κινητήρα. Όποιος κινητήρας έχει δουλέψει τελευταίος ξεκινάει και τελευταίος.

Symbol	Address	Comment
Συντήρηση	I0.0	
S3	I0.1	Ελάχιστη Στάθμη
S2	I0.2	Μεσαία Στάθμη
S1	I0.3	Μέγιστη Στάθμη
Θερμικό1	I0.4	Βλάβη αντλίας 1
Θερμικό2	I0.5	Βλάβη αντλίας 2
Θερμικό3	I0.6	Βλάβη αντλίας 3
A1	Q0.1	Αντλία 1
A2	Q0.2	Αντλία 2
A3	Q0.3	Αντλία 3



Network 4

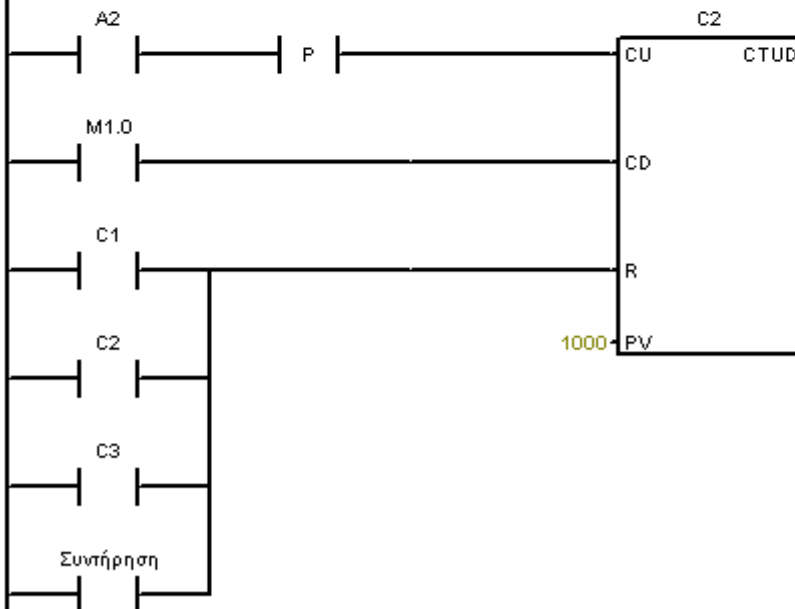
Κάθε φορά που ξανά εκκινείτε ο κινητήρας 1 αύξησε τον μετρητή κατά ένα ,αν κάποιος από τους κινητήρες εκκινωθεί 1000 φορές ή γίνει συντήρηση μηδένισε τον μετρητή



Symbol	Address	Comment
A1	Q0.1	Αυτίκι 1
Συντήρηση	I0.0	

Network 5

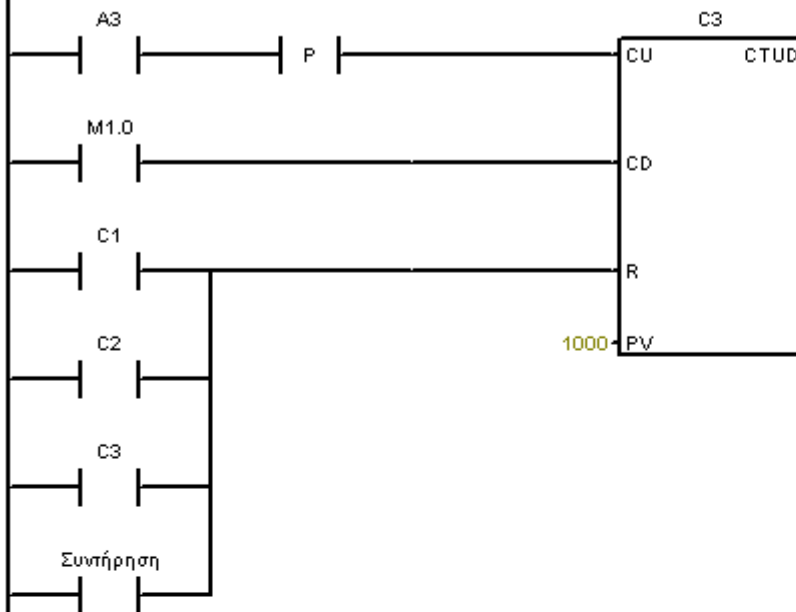
Κάθε φορά που ξανά εκκινείτε ο κινητήρας 2 αύξησε τον μετρητή κατά ένα ,αν κάποιος από τους κινητήρες εκκινωθεί 1000 φορές ή γίνει συντήρηση μηδένισε τον μετρητή



Symbol	Address	Comment
A2	Q0.2	Αυτίκι 2
Συντήρηση	I0.0	

Network 6

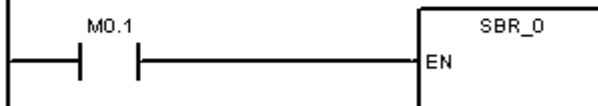
Κάθε φορά που ξανά εκκινείτε ο κινητήρας 3 αύξησε τον μετρητή κατά ένα ,αν κάποιος από τους κινητήρες εκκινωθεί 1000 φορές ή γίνει συντήρηση μηδένισε τον μετρητή



Symbol	Address	Comment
A3	Q0.3	Αυτίκι 3
Συντήρηση	I0.0	

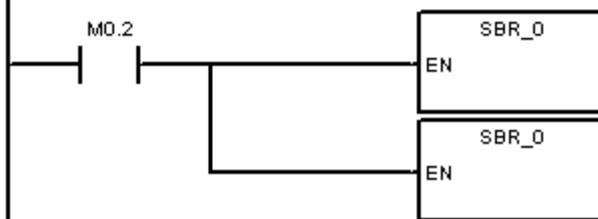
Network 7

Εάν έχει ενεργοποιηθεί το M0.1 τότε κάλεσε την υπορουτίνα SBR 0



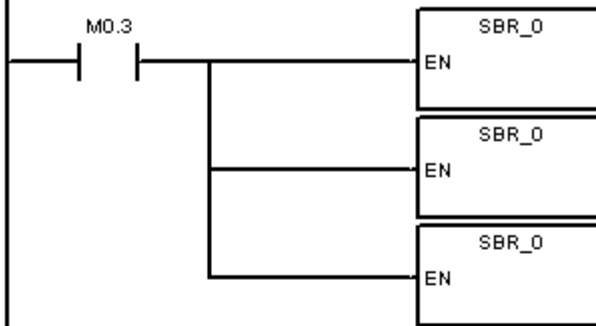
Network 8

Εάν έχει ενεργοποιηθεί το M0.2 τότε κάλεσε δυο φορές την υπορουτίνα SBR 0



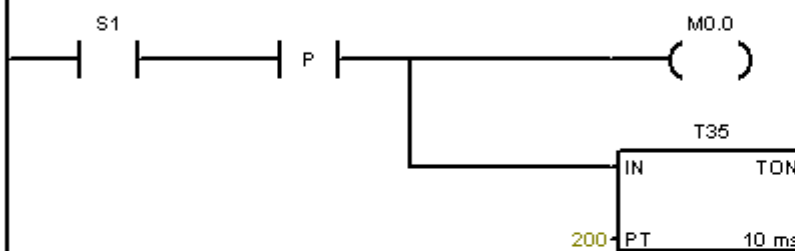
Network 9

Εάν έχει ενεργοποιηθεί το M0.3 τότε κάλεσε τρεις φορές την υπορουτίνα SBR_0



Network 10

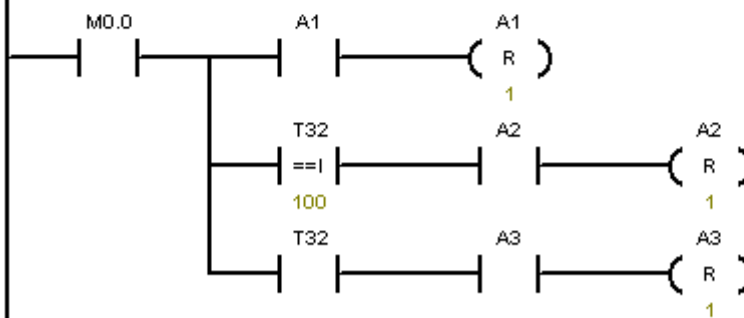
Εάν το νερό φτάσει στην ανώτερη στάθμη τότε ενεργοποίησε το M0.0 και το χρονικό για 2sec



Symbol	Address	Comment
S1	I0.3	Μέγιστη Στάθμη

Network 11

Εάν έχει ενεργοποιηθεί το M0.0 και είναι σε λειτουργία ο πρώτος κινητήρας τότε σταμάτησε των ,αν έχει περάσει 1sec και ο δεύτερος κινητήρας λειτουργεί σταμάτησε των και όταν περάσουν 2sec και ο τρίτος κινητήρας λειτουργεί σταμάτησέ των

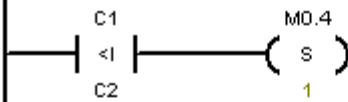


Symbol	Address	Comment
A1	Q0.1	Αντλία 1
A2	Q0.2	Αντλία 2
A3	Q0.3	Αντλία 3

SBR_0

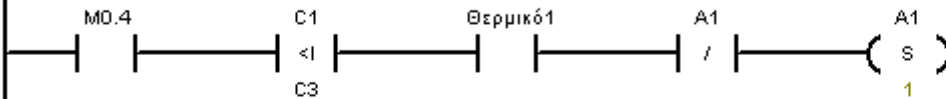
Network 1

Αν ο πρώτος κινητήρας έχει δουλέψει λιγότερες φορές από των δεύτερο ξεκίνα την M0.4



Network 2

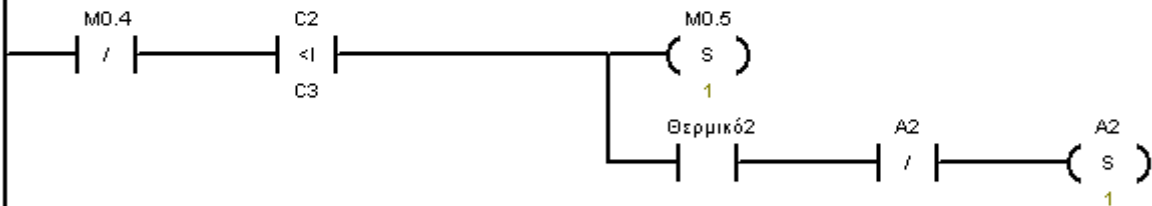
Αν έχει ενεργοποιηθεί η M0.4 και ο πρώτος κινητήρας έχει δουλέψει λιγότερες φορές από των τρίτο όπως επίσης αν δεν έχει βλάβη ο κινητήρας 1 και δεν λειτουργεί τότε ενεργοποίησε των



Symbol	Address	Comment
A1	Q0.1	Αντλία 1
Θερμικό1	I0.4	Βλάβη αντλίας 1

Network 3

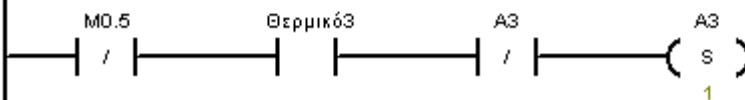
Αν δεν έχει ενεργοποιηθεί η M0.4 και ο δεύτερος κινητήρας έχει δουλέψει λιγότερες φορές από των τρίτο όπως επίσης αν δεν έχει βλάβη ο κινητήρας 2 και δεν λειτουργεί τότε ενεργοποίησε των και ενεργοποίησε το M0.5



Symbol	Address	Comment
A2	Q0.2	Αντλία 2
Θερμικό2	I0.5	Βλάβη αντλίας 2

Network 4

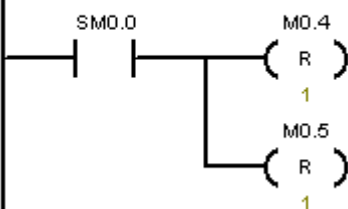
Αν δεν έχει ενεργοποιηθεί το M0.5 και δεν έχει βλάβη ο κινητήρας 3 και δεν λειτουργεί τότε ενεργοποίησε των



Symbol	Address	Comment
A3	Q0.3	Αντλία 3
Θερμικό3	I0.6	Βλάβη αντλίας 3

Network 5

Το SM0.0 είναι πάντα 1 . Απενεργοποίησε το M0.4 και M0.5



Βιβλιογραφία

- [1] http://vivliothmyy.ee.auth.gr/707/2/Zisidou_Salamani_diplomatiki.pdf
- [2] <http://www.sml.ee.upatras.gr/Courses/IndustrialAutomationNetworks/IndustrialAutomationNetworks.aspx>
- [3] http://www.sml.ee.upatras.gr/UploadedFiles/%CE%94%CE%92%CE%91_%CE%9A%CE%B5%CF%86%CE%B1%CE%BB%CE%B1%CE%B9%CE%BF1.pdf
- [4] http://agro-fst.web.auth.gr/3-Spoydes_proptyxiaka/ANAKOINVSEIS_DIDAKONTVN_gr/%CE%9B%CE%91%CE%96%CE%91%CE%A1%CE%99%CE%94%CE%97%CE%A3-%CE%A3%CE%A5%CE%A3%CE%A4%CE%97%CE%9C%CE%91%CE%A4%CE%91%20%CE%91%CE%A5%CE%A4%CE%9F%CE%9C%CE%91%CE%A4%CE%99%CE%A3%CE%9C%CE%9F%CE%A5_gr/%CE%99%CF%83%CF%84%CE%BF%CF%81%CE%AF%CE%B1%20%CE%BA%CE%B1%CE%B9%20%CE%B5%CE%BE%CE%AD%CE%BB%CE%B9%CE%BE%CE%B7%20%CF%84%CF%89%CE%BD%20%CF%83%CF%85%CF%83%CF%84%CE%B7%CE%BC%CE%AC%CF%84%CF%89%CE%BD%20%CE%B5%CE%BB%CE%AD%CE%B3%CF%87%CE%BF%CF%85-BW.pdf
- [5] www.mtee.net/Presentations/MM/PLCs.ppt
- [6] http://nemertes.lis.upatras.gr/dspace/bitstream/123456789/4183/1/teliko_keimeno.pdf
- [7] <http://plc.openforall.net/html/theory.html>
- [8] http://agro-fst.web.auth.gr/3-Spoydes_proptyxiaka/ANAKOINVSEIS_DIDAKONTVN_gr/%CE%9B%CE%91%CE%96%CE%91%CE%A1%CE%99%CE%94%CE%97%CE%A3-%CE%A3%CE%A5%CE%A3%CE%A4%CE%97%CE%9C%CE%91%CE%A4%CE%91%20%CE%91%CE%A5%CE%A4%CE%9F%CE%9C%CE%91%CE%A4%CE%99%CE%A3%CE%9C%CE%9F%CE%A5_gr/%CE%A3%CE%A5%CE%A3%CE%A4%CE%97%CE%9C%CE%91%CE%A4%CE%91%20%CE%91%CE%A5%CE%A4%CE%9F%CE%9C%CE%91%CE%A4%CE%99%CE%A3%CE%9C%CE%9F%CE%A5%20%CE%99%20_%CE%9A%CE%B5%CF%86.%201-6_%20.pdf
- [9] http://artemis.cslab.ntua.gr/el_thesis/artemis.ntua.ece/DT2006-0018/DT2006-0018.pdf
- [10] http://eureka.lib.teithe.gr:8080/bitstream/handle/10184/1231/gkor_papa_main.pdf?sequence=3
- [11] http://vivliothmyy.ee.auth.gr/94/1/%CE%94%CE%B9%CF%80%CE%BB%CF%89%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CE%AE_%CE%A7%CE%B1%CF%81%CE%AF%CF%83%CE%B7%CF%82_%CE%92%CE%B1%CF%83%CE%AF%CE%BB%CE%B7%CF%82.pdf
- [12] <http://nefeli.lib.teicrete.gr/browse/stef/hle/2007/PitsavasNikolaos/attached-document-1295347419-226739-13679/pitsabas2007.pdf>
- [13] <http://users.cs.teilar.gr/iatrellis/files/%CE%94%CE%99%CE%9A%CE%A4%CE%A5%CE%A9%CE%A3%CE%97%20%CE%9A%CE%91%CE%99%20%CE%91%CE%A5%CE%A4%CE%9F%CE%9C%CE%91%CE%A4%CE%9F%CE%A0%CE%9F%CE%99%CE%97%CE%A3%CE%97%>

[CE%92%CE%99%CE%9F%CE%9C%CE%97%CE%A7%CE%91%CE%9D%CE%99%CE%9A%CE%91%20%CE%A3%CE%A5%CE%A3%CE%A4%CE%97%CE%9C%CE%91%CE%A4%CE%91%20%CE%95%CE%9B%CE%95%CE%93%CE%A7%CE%9F%CE%A5%20PLC%20b.pdf](#)

[14] http://www.jimkava.com/wp-content/uploads/2011/10/Ergasthrio_aytomatismwn_sae_ask1-7.pdf

[15] http://vivliothmmy.ee.auth.gr/294/1/anafora_final.pdf

[16]

http://www.google.gr/url?sa=t&source=web&cd=1&sqi=2&ved=0CC8QFjAA&url=http%3A%2F%2Ffikaros.teipir.gr%2Fphyche%2FSubjects%2FRoutoulas%2FAutomatismoiVIKTE%2FBA3.ppt&ei=wqnOTZLFG4busgbXj_2SCw&usq=AFQjCNEK15-OtMN_pqxhGuswkue4H2Nq9g&sig2=HRzVRRmKZDfzX8GUjnVM0A

[17]

http://nemertes.lis.upatras.gr/dspace/bitstream/123456789/2672/1/4.%CE%A4%CE%B5%CE%BB%CE%B9%CE%BA%CE%BF_%CE%94%CE%B9%CE%B1%CE%BC%CE%BF%CF%81%CF%86%CF%89%CE%BC%CE%B5%CE%BD%CE%BF.pdf

[18] <http://nefeli.lib.teicrete.gr/browse/stef/hle/2010/DarakisNikolaos/attached-document-1276675269-859171-8202/Darakis2010.pdf>

[19] http://www.lme.ntua.gr/repos/lessons/control_eidika_systimata/ch3_engine_autom.pdf

[20] http://micro-kosmos.uoa.gr/gr/magazine/ergasies_foititon/ettap/2010-11/h-taxi/var/SAE_B_TOMOS_SEL9_106.PDF

[21]

<http://nefeli.lib.teicrete.gr/browse/stef/hlk/2010/GiakoumakisEmmanouil,PollakisGerasimos/attached-document-1277883811-853544-3107/2010giakoumakispollakis.pdf>

[22]

http://nemertes.lis.upatras.gr/dspace/bitstream/123456789/1811/3/Nimertis_Papadopoulos%28ele%29.pdf

[23] <http://nemertes.lis.upatras.gr/jspui/handle/10889/4912>

[24] http://artemis.cs.lab.ntua.gr/el_thesis/artemis.ntua.ece/DT2011-0164/DT2011-0164.pdf

[25]

<http://ikaros.teipir.gr/phyche/Subjects/Routoulas/AutomatismoiVIKTE/ViomixanikosAutomatismos.pdf>

[26] <http://www.hlektronika.gr/forum/showthread.php?t=55104>

[27] <http://vivliothmmy.ee.auth.gr/577/3/ergasia.pdf>

[28] <http://thesis.ekt.gr/thesisBookReader/id/17710#page/1/mode/2up>

[29] <http://www.el.teithe.gr/File.aspx?UI=1&ID=364&PID=795&File=ANNOUNCEMENTS>

[30] <http://invnio.lib.auth.gr/record/100444/files/gri-2007-921.pdf?version=1>

[31]

[http://conta.uom.gr/conta/ekpaideysh/metapyxiaka/technologies_diktywn/ergasies/2007/Industrial%20\(Automation\)%20Network%20protocols.pdf](http://conta.uom.gr/conta/ekpaideysh/metapyxiaka/technologies_diktywn/ergasies/2007/Industrial%20(Automation)%20Network%20protocols.pdf)

[32] http://nefeli.lib.teicrete.gr/browse/stef/epp/2008/MarkakisFotios/attached-document/Markakis_Fotios.pdf

[33] http://library.tee.gr/digital/m2107/m2107_kontogiannis.pdf

[34] <http://nefeli.lib.teicrete.gr/browse/stef/hlk/2007/DolianitisSpyridon/attached-document/2007Dolianitis.pdf>

[35]

https://docs.google.com/viewer?a=v&q=cache:xJesxxgZfqIJ:ikaros.teipir.gr/phyche/Subjects/Routoula/s/AutomatismoiVIKTE/BA3.ppt+%CE%92%CE%99%CE%9F%CE%9C%CE%97%CE%A7%CE%91%CE%9D%CE%99%CE%9A%CE%9F%CE%99+%CE%91%CE%A5%CE%A4%CE%9F%CE%9C%CE%91%CE%A4%CE%99%CE%A3%CE%9C%CE%9F%CE%99+%CE%91.%CE%A4.%CE%A1%CE%9F%CE%A5%CE%A4%CE%9F%CE%A5%CE%9B%CE%91%CE%A3&hl=el&gl=gr&pid=bl&srcid=ADGEESH1q1swp8IgmJOcmq_6SGdlsyA36F6jLny04gbb0p8ux83kX481GJHrfPy9fLuFtj6RuIbwNebyCuzj6ALPxdaKcz_GbFsZoyu-CdCY9jXXpC3x6k8cyjTb0b2lafXGos3M0NqY&sig=AHIEtbQ4qplPzJTmaninkmgCgL-MSB557w

[36] <http://nefeli.lib.teicrete.gr/browse/stef/hle/2007/PitsavasNikolaos/attached-document-1295347419-226739-13679/pitsabas2007.pdf>