

CPU

DI

DO

**Προγραμματιζόμενοι
Λογικοί
Ελεγκτές
(PLC)**

μP



Επιδιωκόμενοι στόχοι:

- ✓ να καταλάβουν οι μαθητές την αξία και την αναγκαιότητα των PLC, τα πλεονεκτήματά τους, την έκταση των εφαρμογών τους,
- ✓ να μάθουν από τι αποτελείται ένα PLC, και να γνωρίζουν τα κοινά χαρακτηριστικά των PLC διαφόρων εταιρειών,
- ✓ να γνωρίσουν την αγορά όσον αφορά τα PLC,
- ✓ να γνωρίσουν ποιες είναι οι κύριες γλώσσες προγραμματισμού, ποια είναι τα κοινά χαρακτηριστικά τους, ποιες είναι οι αρχές προγραμματισμού της κάθε γλώσσας,
- ✓ να συσχετίζουν και να συνδέουν όλα όσα διδάχθηκαν στα προηγούμενα κεφάλαια για τον κλασικό αυτοματισμό με τα PLC.

Οι κεντρικοί στόχοι είναι να μπορούν οι μαθητές μετά το τέλος του κεφαλαίου:

1. να μετατρέπουν μια εφαρμογή κλασικού αυτοματισμού (το ηλεκτρολογικό σχέδιο του αυτοματισμού) σε πρόγραμμα για PLC, σε μια από τις τρεις πιο γνωστές γλώσσες, δηλαδή τη γλώσσα λίστα εντολών, τη γλώσσα LADDER, τη γλώσσα λογικών γραφικών,
2. να σχεδιάζουν μια απλή εφαρμογή αυτοματισμού κατευθείαν για PLC.

1. Η εξέλιξη των αυτοματισμών και οι προγραμματιζόμενοι λογικοί ελεγκτές (PLC).

Η εξέλιξη των αυτοματισμών, όπως ήταν φυσικό, ακολούθησε την πορεία εξέλιξης της τεχνολογίας. Οι πρώτοι αυτοματισμοί ήταν καθαρά μηχανικοί, όλοι οι έλεγχοι δηλαδή καθοριζόταν από την κίνηση γραναζιών και μοχλών. Το μεγάλο άλμα στους αυτοματισμούς έγινε με τη χρήση του ηλεκτρισμού. Το κύριο εξάρτημα των ηλεκτρολογικών αυτοματισμών είναι ο ηλεκτρονόμος.

Μετά τον δεύτερο παγκόσμιο πόλεμο αρχίζει η ηλεκτρονική εποχή. Ήδη από τις αρχές του 20^{ου} 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 ενεργειακόβορα 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 έλυσε 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.

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. Μέχρι όμως την δεκαετία του '80 αυτό ήταν αδύνατο, διότι ο υπολογιστής ήταν μια πανάκριβη και δύσκολη στην χρήση της συσκευή.

Η επανάσταση της πληροφορικής ξεκινά το 1975 με την κατασκευή του πρώτου μικροϋπολογιστή. Πολλά από όλα όσα σήμερα θεωρούμε αυτονόητα δημιουργήθηκαν μετά το 1980. Η τεχνολογία άλλαξε πορεία, αλλάζοντας πορεία σε όλους τους τομείς της καθημερινής ζωής. Ο μικροϋπολογιστής "τρύπωσε" παντού, σε οποιοδήποτε τομέα, σε οποιαδήποτε εφαρμογή.

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.

Στις αρχές της δεκαετίας του '80 οι εταιρείες παραγωγής ηλεκτρολογικού υλικού εμφανίζουν στους μηχανικούς και τεχνικούς της βιομηχανίας ένα νέο προϊόν αυτοματισμού, το οποίο ονόμασαν PLC. Η πλήρης ονομασία αυτής της νέας συσκευής είναι Programmable Logic Controller (Προγραμματιζόμενος Λογικός Ελεγκτής). Οι εταιρείες δεν χρησιμοποίησαν αρχικά στην αγορά την πλήρη ονομασία, μιλώντας απλά για PLC, πράγμα που ίσως έγινε έντεχνα για να μην τρομάζουν το τεχνικό κατεστημένο της Βιομηχανίας.

Το PLC δεν είναι τίποτα άλλο παρά ένας μικροϋπολογιστής, κατάλληλα προσαρμοσμένος ώστε να χρησιμοποιείται για τη λειτουργία αυτοματισμών. Τα PLC προορίζονταν να αντικαταστήσουν τον κλασικό πίνακα αυτοματισμού με τους ηλεκτρονόμους. Όπως γίνεται εύκολα κατανοητό μιλάμε για μια τεράστια αλλαγή στον τρόπο που μέχρι τότε δούλευε η βιομηχανία, δηλαδή έπρεπε να περάσει κατευθείαν από τους ηλεκτρονόμους στους υπολογιστές! Εδώ ήταν που οι εταιρείες παραγωγής PLC έπαιξαν ένα σπουδαίο "παιχνίδι" μάρκετινγκ. Προσάρμοσαν τον τρόπο χρήσης του PLC στον τρόπο που δούλευε μέχρι τότε η βιομηχανία, δηλαδή:

- Έντεχνα απέφυγαν να χρησιμοποιήσουν λέξεις που θα "τρόμαζαν" το τεχνικό κατε-

στημένο της βιομηχανίας, όπως για παράδειγμα υπολογιστής, προγραμματισμός κ.λπ. Ακόμη και το όνομα του νέου προϊόντος απέφευγαν να το χρησιμοποιήσουν ολοκληρωμένο και προτιμούσαν να αναφέρουν τη συσκευή σαν PLC.

- Προσπάθησαν να μην αλλάξουν τον μέχρι τότε τρόπο εργασίας στον τομέα των αυτοματισμών. Δεν άλλαξαν δηλαδή τίποτα σε σχέση με τον σχεδιασμό ενός αυτοματισμού. Απλά είπαν, στους τεχνικούς: *“αυτό το σχέδιο αντί να το δώσετε στον ηλεκτρολόγο για να το κατασκευάσει, θα το φτιάξετε με τον τρόπο που θα σας δείξουμε”*, και στην ουσία τους μάθαιναν προγραμματισμό.
- Οι πρώτες γλώσσες προγραμματισμού δεν έκαναν τίποτα παραπάνω από το να αντιγράφουν με πλήκτρα σε μία ειδική συσκευή προγραμματισμού το σχέδιο του ηλεκτρολογικού αυτοματισμού.

Με τον τρόπο αυτό η είσοδος του PLC στην βιομηχανία υπήρξε επιτυχής και ομαλή. Σήμερα, ο κλασικός αυτοματισμός με ηλεκτρονόμους τείνει να εκλείψει. Όλες οι καινούργιες εγκαταστάσεις χρησιμοποιούν PLC. Μετά από λίγα χρόνια ελάχιστες εγκαταστάσεις θα χρησιμοποιούν πίνακες κλασικού αυτοματισμού.

Σήμερα, τα PLC έχουν εξελιχτεί πάρα πολύ σε σχέση με τα πρώτα μοντέλα της δεκαετίας του '80. Και βέβαια το προσωπικό της βιομηχανίας έχει εκπαιδευτεί κατάλληλα στον χειρισμό και προγραμματισμό τους. Σήμερα ένας ηλεκτρολόγος πρέπει να γνωρίζει στοιχειώδη πράγματα από τα ηλεκτρονικά και τις βασικές αρχές των υπολογιστών, αλλιώς θα είναι πολύ δύσκολο να διαβάσει και να καταλάβει ακόμη και το πιο απλό εγχειρίδιο ενός PLC.

Η χρήση των PLC παρέχει πάρα πολλά πλεονεκτήματα σε σχέση με τον κλασικό αυτοματισμό. Η καθολική όμως γενίκευση της χρήσης τους δεν οφείλεται μόνο στα πλεονεκτήματα που παρέχουν στον τελικό χρήστη.

Η χρήση των PLC σε σχέση με τον κλασικό αυτοματισμό συμφέρει πρώτιστα στις εταιρείες που παράγουν είδη αυτοματισμού. Φανταστείτε μόνο πόσο κοστίζει σε μια εταιρεία παραγωγής ηλεκτρολογικού εξοπλισμού η παραγωγή ενός τεράστιου αριθμού βοηθητικών ηλεκτρονόμων και ενός μεγάλου αριθμού χρονικών και απαριθμητών. Σε αντίθεση με τα υλικά αυτά αυτοματισμού, όσον αφορά τα PLC η εταιρεία τι παράγει; Η απάντηση είναι: *“Μια και μοναδική συσκευή!”*

Ναι! Η ψηφιοποίηση σε όλους τους τομείς (και όχι μόνο στον τομέα των αυτοματισμών) οδηγεί σε τρομακτική μείωση του κόστους παραγωγής των αντίστοιχων συσκευών. Γι' αυτό μην απορείτε που οι τιμές στα ηλεκτρονικά πέφτουν! Να είστε σίγουροι ότι τα κέρδη των εταιρειών ανεβαίνουν.

Πάντως θα μπορούσαμε να πούμε ότι, ενώ σε όλους τους τομείς της παραγωγής περάσαμε από τις ηλεκτρολογικές συσκευές, στις ηλεκτρονικές με λυχνίες, μετά στις ηλεκτρονικές με ημιαγωγούς(τρανζίστορ) και τέλος φθάσαμε στις συσκευές με μικροϋπολογιστές, στις ψηφιακές συσκευές δηλαδή, στον τομέα των αυτοματισμών περάσαμε σχεδόν κατευθείαν από τους ηλεκτρολογικούς αυτοματισμούς στους αυτοματισμούς με

PLC. Αν θέλουμε να αναζητήσουμε την αιτία γι' αυτό, θα λέγαμε ότι μάλλον δεν προλάβουμε! Οι εξελίξεις στην ηλεκτρονική ήταν ραγδαίες, ενώ αντίθετα η βιομηχανική τεχνολογία αλλάζει με πολύ πιο αργούς ρυθμούς.

2. Τι είναι ο προγραμματιζόμενος λογικός ελεγκτής.

Στον κλασικό αυτοματισμό της καλωδιωμένης λογικής (τον αυτοματισμό με ηλεκτρονόμους) τα στάδια εργασίας από το σχεδιασμό και την κατασκευή ενός αυτοματισμού μέχρι το σημείο της πλήρους λειτουργίας είναι τα εξής:

1. Περιγραφή τους αυτοματισμού.
2. Ανάπτυξη του λειτουργικού σχεδίου του αυτοματισμού.
3. Ανάπτυξη του σχεδίου εφαρμογής του πίνακα (σχέδιο καλωδίωσης).
4. Κατασκευή του πίνακα της εγκατάστασης.
5. Εγκατάσταση και σύνδεση στους ακροδέκτες (κλέμες) του πίνακα των αισθητήρων που δίνουν τις πληροφορίες (εντολές) και των συσκευών (αποδεκτών) που εκτελούν τις εργασίες.
6. Δοκιμή λειτουργίας της εγκατάστασης.
7. Πλήρης λειτουργία του αυτοματισμού.

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. Τα στάδια εργασίας για το σχεδιασμό και την κατασκευή ενός αυτοματισμού στην προγραμματιζόμενη λογική είναι τα εξής:

1. Περιγραφή τους αυτοματισμού.
2. Ανάπτυξη του σχεδίου εφαρμογής του πίνακα (σχέδιο καλωδίωσης).
3. Κατασκευή του πίνακα της εγκατάστασης.
4. Ανάπτυξη του προγράμματος λειτουργίας του αυτοματισμού και εισαγωγή του προγράμματος στο PLC μέσω του προγραμματιστή.
5. Εγκατάσταση και σύνδεση στους ακροδέκτες (κλέμες) του πίνακα των αισθητήρων που δίνουν τις πληροφορίες (είσοδοι) και των συσκευών (αποδεκτών) που εκτελούν τις εργασίες (έξοδοι).
6. Δοκιμή λειτουργίας της εγκατάστασης.
7. Πλήρης λειτουργία του αυτοματισμού.

Παρατηρούμε ότι τα στάδια, τα οποία αλλάζουν στις εργασίες σχεδιασμού και κατασκευής ενός αυτοματισμού, όταν χρησιμοποιούμε την προγραμματιζόμενη λογική, είναι τα

2, 3 και 4. Αντί για την κατασκευή ενός πίνακα, με πλήθος υλικών και πολύπλοκες καλωδιώσεις, έχουμε την κατασκευή ενός πίνακα με ελάχιστα υλικά, απλές καλωδιώσεις και τον προγραμματισμό του PLC. Ο χρόνος, που απαιτείται για τον προγραμματισμό του PLC και την κατασκευή του μικρού και απλού πίνακα αυτοματισμού, είναι πολύ μικρός σε σχέση με τον χρόνο, που απαιτείται για τη μελέτη και την κατασκευή του αντίστοιχου πολύπλοκου πίνακα κλασικού αυτοματισμού. Αυτό όμως δεν είναι και το μοναδικό πλεονέκτημα που προκύπτει από την εφαρμογή των PLC.

2.1 Πλεονεκτήματα των προγραμματιζόμενων λογικών ελεγκτών.

Τα πρώτα μεγάλα πλεονεκτήματα των PLC αφορούν τους κατασκευαστές εξοπλισμού αυτοματισμών και πινάκων αυτοματισμού και ήδη τα έχουμε αναφέρει:

1. Το κόστος κατασκευής ενός PLC είναι σημαντικά μικρότερο από το κόστος παραγωγής ενός μεγάλου αριθμού βοηθητικών ηλεκτρονόμων, χρονικών και απαριθμητών.
2. Ο χρόνος κατασκευής του αυτοματισμού είναι μηδαμινός σε σχέση με την κατασκευή ενός κλασικού πίνακα αυτοματισμού.

Υπάρχουν όμως πολλά πλεονεκτήματα που έχουν σχέση με τον τελικό χρήστη, τις βιομηχανίες δηλαδή που εφαρμόζουν τους αυτοματισμούς, και είναι αυτά που μας ενδιαφέρουν περισσότερο. Κατά σειρά σπουδαιότητας αναφέρουμε:

- *Τα PLC ελαχιστοποιούν το κόστος συντήρησης του πίνακα αυτοματισμού.* Το κόστος αυτό αναλύεται ως εξής: Συχνότητα βλαβών, χρόνος εντοπισμού μιας βλάβης και αποκατάστασής της. Δηλαδή, όταν υπάρχει μια βλάβη στον πίνακα μιας εγκατάστασης κλασικού αυτοματισμού, υπάρχει καθυστέρηση στην παραγωγή μέχρι να εντοπιστεί η βλάβη. Αφού εντοπιστεί, πρέπει να έχουμε διαθέσιμο το κατάλληλο ανταλλακτικό στην αποθήκη, γιατί διαφορετικά θα υπάρξει σημαντική καθυστέρηση, αφού θα χρειαστεί να γίνει η σχετική παραγγελία και η προμήθεια. Στον αυτοματισμό με PLC δεν υπάρχει ουσιαστικό θέμα βλάβης εσωτερικά στον πίνακα της εγκατάστασης. Θα πείτε, δεν “χαλά” το PLC; Αυτό συμβαίνει σπάνια και οι εγγυήσεις είναι πάρα πολύ μεγάλες.
- *Τα PLC είναι ευέλικτα στην τροποποίηση της λειτουργίας του αυτοματισμού.* 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 ένα πίνακα κλασικού αυτοματισμού, τέτοιου είδους αλλαγές είναι πράγμα πολύ δύσκολο και χρονοβόρο.
- *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 κλασικό αυτοματισμό είναι πολύ δύσκολη.
- *Ο αυτοματισμός με PLC μας παρέχει καταπληκτικές δυνατότητες.* Μπορούμε να δημιουργούμε πολύ εύκολα πολύπλοκες και έξυπνες επεξεργασίες, οι οποίες στον κλασικό αυτοματισμό είναι εξαιρετικά δύσκολο να υλοποιηθούν.

- Σε μια εγκατάσταση, που χρησιμοποιεί αυτοματισμούς με PLC, σήμερα παρέχονται δυνατότητες σύνδεσης με κεντρικό ηλεκτρονικό υπολογιστή, σύνδεσης με το σύστημα αποθήκης, λογιστηρίου κ.λπ.
- Το PLC καταλαμβάνει ελάχιστο χώρο σε σχέση με τον αντίστοιχο πίνακα κλασικού αυτοματισμού.

Βλέπουμε ότι από τη χρήση των PLC προκύπτουν μόνο πλεονεκτήματα. Υπάρχουν άραγε μειονεκτήματα; Θα μπορούσαμε ίσως να θεωρήσουμε μειονέκτημα την έλλειψη επαρκούς ενημέρωσης των τεχνικών όλων των βαθμίδων, ειδικά στην Ελλάδα, πράγμα το οποίο δυσκολεύει και δημιουργεί προβλήματα στην εφαρμογή των PLC.

Η τελική ερώτηση, που προκύπτει, είναι: Πόσο κοστίζουν τελικά τα PLC; συμφέρει να χρησιμοποιούμε σε κάθε εγκατάσταση PLC; Η απάντηση είναι ότι οι τιμές “πέφτουν” καθημερινά και οι εταιρείες βγάζουν συνεχώς νέα μοντέλα που κάνουν για όλων των επιπέδων τις εφαρμογές. Παρ’ όλα αυτά δεν συμφέρει ακόμη η χρήση του PLC, όταν έχουμε πολλούς κινητήρες (πολλούς ηλεκτρονόμους ισχύος) και απλό αυτοματισμό (λίγους βοηθητικούς ηλεκτρονόμους, χρονικά και απαριθμητές). Όμως είναι σίγουρο ότι η χρήση βοηθητικών ηλεκτρονόμων και άλλων κλασικών υλικών αυτοματισμού έχει περιορισθεί και θα περιορίζεται συνεχώς.

2.2 Η δομή ενός προγραμματιζόμενου λογικού ελεγκτή.

Στην αγορά υπάρχουν σήμερα εκατοντάδες μοντέλα PLC κατασκευασμένα από πλήθος διαφορετικών εταιρειών. Γενικά, σε ένα PLC μπορούμε να διακρίνουμε τα παρακάτω μέρη:

- Την κεντρική μονάδα επεξεργασίας (Central Processing Unit, CPU), που αποτελεί και την καρδιά ή μάλλον τον εγκέφαλο του PLC.
- Τη μονάδα τροφοδοσίας.
- Τις μονάδες εισόδων - εξόδων (I/O modules).

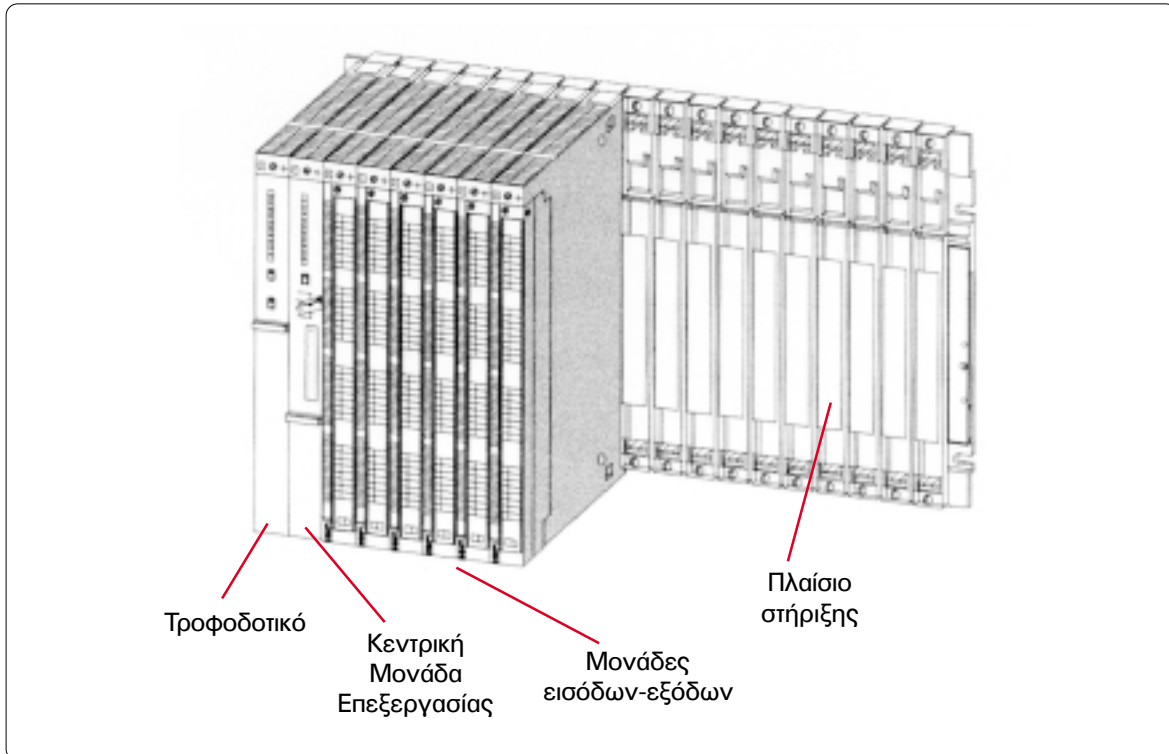
Η κεντρική μονάδα, η μονάδα τροφοδοσίας και οι μονάδες εισόδων - εξόδων αποτελούν την κύρια μονάδα αυτοματισμού, δηλαδή το κύριο μέρος του PLC.

Σε πολλά μοντέλα, κυρίως στα μικρά μοντέλα των εταιρειών, οι τρεις παραπάνω μονάδες βρίσκονται ενσωματωμένες σε μια συσκευή.

Εκτός από την κεντρική μονάδα αυτοματισμού, σε ένα PLC είναι ακόμη απαραίτητα:

- Το πλαίσιο (ή τα πλαίσια) για την τοποθέτηση των μονάδων και των τυχόν επεκτάσεών τους.
- Η συσκευή προγραμματισμού (προγραμματιστής, programmer) για τον προγραμματισμό του PLC.

Ο προγραμματιστής είναι μια συσκευή τελείως ξεχωριστή από την μονάδα αυτοματισμού. Χρησιμοποιείται για την εισαγωγή του προγράμματος στο PLC και την παρακολούθηση της εξέλιξης του αυτοματισμού μέσα από την οθόνη, που διαθέτει. Με ένα μόνο προγραμματιστή μπορούμε να χειριζόμαστε όλες τις μονάδες PLC μιας αυτοματοποιημένης εγκατάστασης (της ίδιας εταιρείας εννοείται).



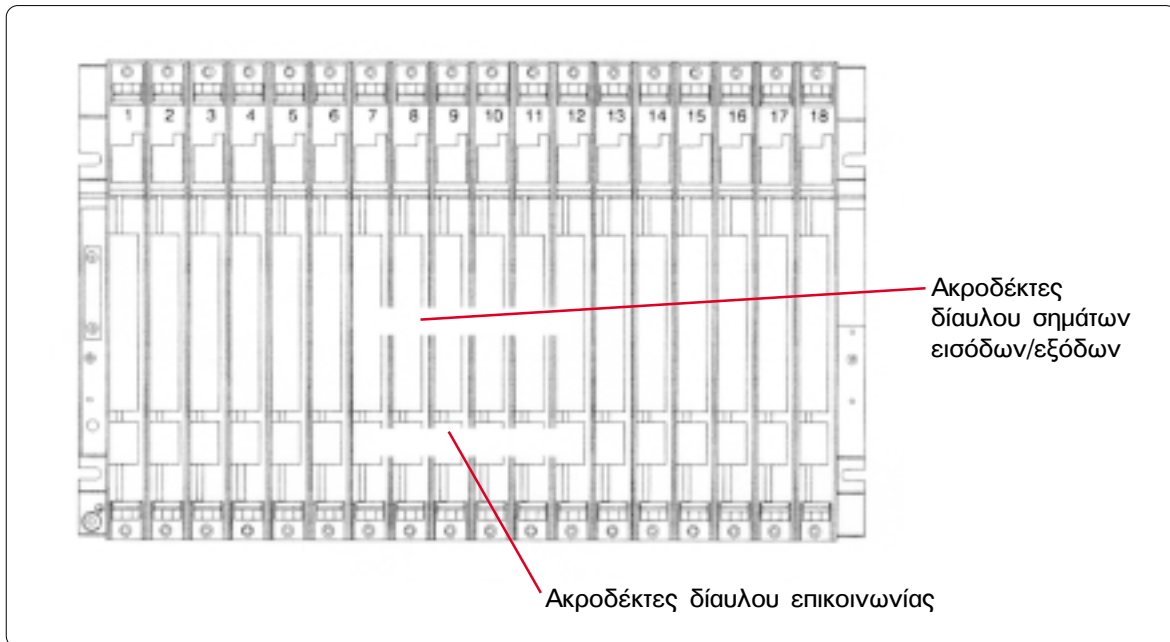
Σχήμα 1: Δομή ενός PLC.

Στη συνέχεια θα δούμε με λεπτομέρεια τις μονάδες ενός PLC.

Πλαίσιο τοποθέτησης μονάδων.

Οι μονάδες ενός μεγάλου PLC τοποθετούνται σε ένα κεντρικό πλαίσιο. Στο πλαίσιο αυτό είναι ενσωματωμένο ένα σύστημα αγωγών (σύστημα ζυγών) μέσω των οποίων επικοινωνούν οι διάφορες μονάδες με την Κεντρική Μονάδα Επεξεργασίας.

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 ένα ή περισσότερα πλαίσια επέκτασης για την τοποθέτηση των πρόσθετων μονάδων. Κάθε πλαίσιο επέκτασης διασυνδέεται με το κεντρικό πλαίσιο ή με τα άλλα πλαίσια επέκτασης μέσω ειδικής μονάδας διασύνδεσης και καλωδίου. Κάθε εταιρεία και το κάθε μοντέλο έχει το δικό του σύστημα πλαισίου. Στο σχήμα 2 δείχνεται ένα πλαίσιο στήριξης PLC.



Σχήμα 2: Πλαίσιο στήριξης PLC.

Μονάδα τροφοδοσίας.

Η μονάδα τροφοδοσίας ενός PLC έχει σκοπό να δημιουργήσει από την τάση του δικτύου τροφοδοσίας τις απαραίτητες εσωτερικές τάσεις, που απαιτούνται για την τροφοδοσία των ηλεκτρονικών στοιχείων (τρανζίστορ, ολοκληρωμένα κυκλώματα κ.λπ.) του PLC. Οι τυπικές εσωτερικές τάσεις των PLC είναι συνήθως: DC 5 V, DC 9 V, DC 24 V.

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.

Χρειάζεται προσοχή ώστε να μην προκαλέσουμε υπερφόρτιση της μονάδας τροφοδοσίας. Για τον σκοπό αυτό συμβουλευόμαστε τα τεχνικά φυλλάδια της εταιρείας κατασκευής του PLC.

Κεντρική Μονάδα Επεξεργασίας.

Είναι η βασική μονάδα του PLC, η οποία είναι υπεύθυνη για τη λειτουργία του αυτοματισμού. Η κεντρική μονάδα επεξεργασίας είναι στην ουσία ένας μικροϋπολογιστής και διακρίνουμε σ' αυτήν όλα τα κύρια μέρη ενός μικροϋπολογιστή, δηλαδή τον μικροεπεξεργαστή και τη μνήμη.

Ο μικροεπεξεργαστής είναι ένα ολοκληρωμένο κύκλωμα, το οποίο αποτελεί τον "εγκέφαλο" κάθε μικροϋπολογιστή. Οι μικροεπεξεργαστές εξελίσσονται με ταχύτατους ρυθ-

μούς, το όνομά τους δίνει συνήθως και το όνομα στο μοντέλο του μικροϋπολογιστή (π.χ λέμε PC 386, 486, Pentium κ.λπ.). Στα PLC πολύ λίγο μας ενδιαφέρει να ξέρουμε ποιον μικροεπεξεργαστή χρησιμοποιεί η κεντρική μονάδα, αν και πολλές φορές μπορούμε να το διαβάσουμε στα χαρακτηριστικά που δίνουν οι εταιρείες. Ο μικροεπεξεργαστής για το PLC είναι ο κύριος υπεύθυνος για όλες τις λειτουργίες του (τη λειτουργία του PLC περιγράφουμε στην επόμενη ενότητα).

Η μνήμη της κεντρικής μονάδας διακρίνεται σε μνήμη RAM, ROM και EEPROM.

Μνήμη RAM. Η μνήμη RAM (Random Access Memory, μνήμη τυχαίας προσπέλασης) είναι εκείνη στην οποία μπορούμε να γράφουμε και να σβήνουμε, και η οποία σβήνει μόλις λείπει η ηλεκτρική τροφοδοσία της. Στη μνήμη RAM η κεντρική μονάδα αποθηκεύει μια σειρά από πληροφορίες σε ξεχωριστές περιοχές εργασίας. Μπορούμε να διακρίνουμε τις εξής περιοχές:

- Περιοχή μνήμης όπου αποθηκεύονται οι καταστάσεις των εισόδων και των εξόδων. Η περιοχή αυτή ονομάζεται για τις εισόδους “εικόνα εισόδων” και για τις εξόδους “εικόνα εξόδων”.
- Περιοχή μνήμης όπου αποθηκεύονται οι ενδιάμεσες πληροφορίες, που αφορούν τη λειτουργία του αυτοματισμού.
- Περιοχή μνήμης των χρονικών.
- Περιοχή μνήμης των απαριθμητών.
- Περιοχή μνήμης όπου αποθηκεύονται τα προγράμματα του χρήστη, δηλαδή τα προγράμματα που λειτουργούν ένα συγκεκριμένο αυτοματισμό.

ΠΡΟΣΟΧΗ! Η μνήμη RAM σβήνει μόλις σταματήσει η τροφοδοσία της. Όμως το πρόγραμμα που λειτουργεί τον αυτοματισμό πρέπει να παραμένει αναλλοίωτο στη μνήμη και αφού κλείσουμε την τροφοδοσία του PLC. Γι’ αυτό το λόγο η μνήμη RAM παραμένει πάντα σε τροφοδοσία μέσω μιας μπαταρίας (συνήθως λιθίου).

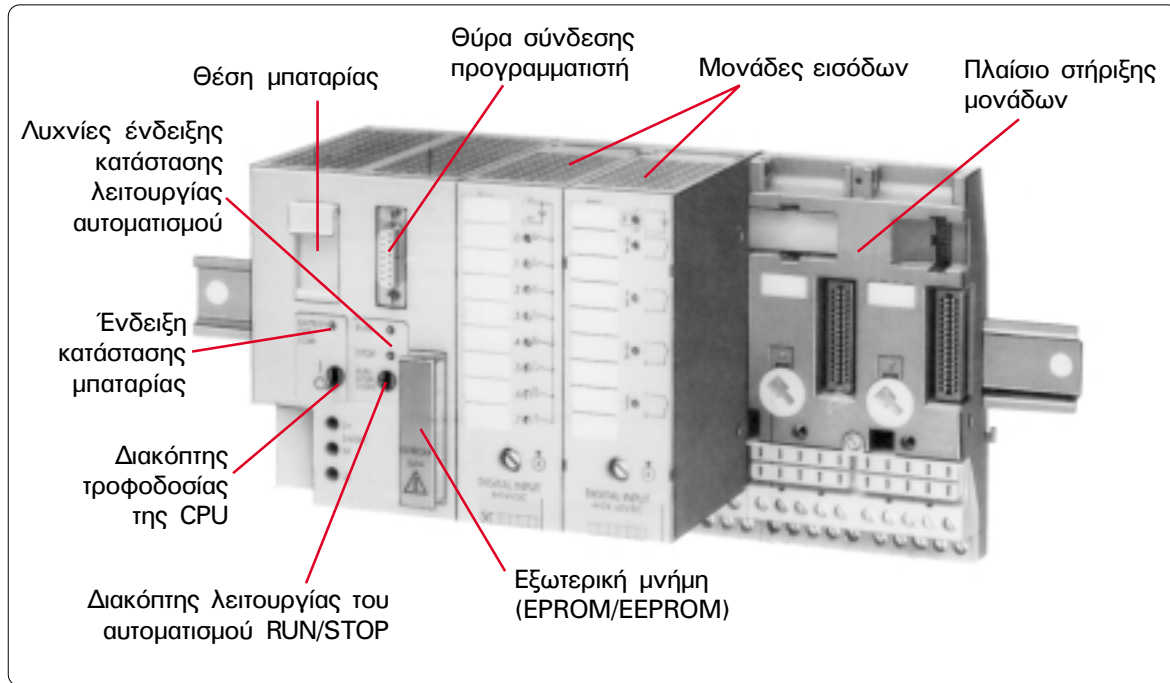
Μνήμη EEPROM. Τα διάφορα PLC δεν χρησιμοποιούν μόνο τον παραπάνω τρόπο, της “πάντα τροφοδοτούμενης μνήμης RAM”, για να διατηρήσουν το πρόγραμμα στη μνήμη. Ένας πιο ασφαλής τρόπος είναι η χρήση της μνήμης EEPROM (Electrically Erasable Programmable Read Only Memory), μνήμης η οποία προγραμματίζεται και σβήνει ηλεκτρικά. Πρόκειται για μνήμη που δε σβήνει, όταν μείνει χωρίς τροφοδοσία, στην οποία μπορούμε να γράφουμε, να σβήνουμε και να ξαναγράψουμε μέσω ειδικού μηχανήματος. Σε πολλά PLC η EEPROM χρησιμοποιείται σαν “κασέτα” για την εύκολη αλλαγή του τρόπου λειτουργίας του αυτοματισμού από ένα απλό χειριστή. Δηλαδή έχουμε “γραμμένο” το εναλλακτικό πρόγραμμα σε ένα “τσιπάκι” EEPROM και απλά αλλάζουμε την ηλεκτρονική πλακέτα του PLC, όταν θέλουμε να αλλάξουμε το πρόγραμμα λειτουργίας του αυτοματισμού.

Μνήμη ROM. Στη μνήμη ROM ο κατασκευαστής του PLC αποθηκεύει το λειτουργικό σύστημα του PLC, δηλαδή τις οδηγίες (το πρόγραμμα) για όλες τις βασικές λειτουργίες που είναι απαραίτητες για να δουλέψει το PLC.

Εξωτερικά σε μια κεντρική μονάδα επεξεργασίας συνήθως υπάρχουν (σχήμα 3):

- Θέση σύνδεσης (ειδικός κονέκτορας) της συσκευής προγραμματισμού.
- Θέση σύνδεσης επεκτάσεων.
- Διακόπτης δύο θέσεων (συνήθως) ο οποίος θέτει το PLC σε κατάσταση RUN ή STOP, δηλαδή σε κατάσταση λειτουργίας του αυτοματισμού (RUN) ή όχι (STOP).
- Λυχνίες ένδειξης, όπως: λυχνία ένδειξης ότι το PLC είναι σε τροφοδοσία, λυχνία ένδειξης ότι το PLC είναι σε κατάσταση RUN (λειτουργεί ο αυτοματισμός), λυχνία

ένδειξης ότι το PLC είναι σε κατάσταση STOP (δεν λειτουργεί ο αυτοματισμός) και λυχνία που δείχνει, εάν έχει πρόβλημα η μπαταρία του PLC.



Σχήμα 3: Κεντρική μονάδα επεξεργασίας και μονάδες εισόδων ενός PLC.

Μονάδες εισόδων - εξόδων.

Οι μονάδες των εισόδων και των εξόδων αποτελούν τις μονάδες επικοινωνίας της κεντρικής μονάδας με τον έξω κόσμο, δηλαδή με τους αισθητήρες, τους διακόπτες και τα μπουτόνς, που δίνουν τις πληροφορίες (εντολές), καθώς και με τους ηλεκτρονόμους ισχύος των κινητήρων, ηλεκτρομαγνητικές βαλβίδες, ενδεικτικές λυχνίες και γενικά τους αποδέκτες που εκτελούν τις εντολές του αυτοματισμού.

Η κεντρική μονάδα μπορεί να δεχτεί ψηφιακά σήματα εισόδου και εξόδου χαμηλής τάσης και πολύ μικρού ρεύματος. Η τάση που δέχεται η κεντρική μονάδα είναι συνήθως 0Volt για το λογικό "0" και 5Volt για το λογικό "1". Το ρεύμα εισόδου καθώς και το ρεύμα εξόδου δεν μπορεί να περάσει τα μερικά mA. Οι μονάδες εισόδων και εξόδων αναλαμβάνουν να προσαρμόσουν τα σήματα εισόδου και εξόδου, που έχουμε στον αυτοματισμό, με τα σήματα που μπορεί να δεχτεί η κεντρική μονάδα, τόσο από άποψη τάσεων όσο και από άποψη ρευμάτων.

Η προσαρμογή αυτή γίνεται με τη χρήση ηλεκτρονικών στοιχείων ισχύος, δηλαδή τρανζίστορ ισχύος, θυρίστορ και triac, είτε ακόμη με τη χρήση κατάλληλων μικροηλεκτρονικών.

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 (θερματικούς διακόπτες, πιεζοστάτες, κ.λπ.), διακόπτες, μπουτόνς 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.

2.3 Οι προγραμματιζόμενοι λογικοί ελεγκτές της αγοράς.

Η κατάσταση που έχει διαμορφωθεί σήμερα στην αγορά από τις εταιρείες, που κατασκευάζουν PLC, είναι η εξής:

Περίπτωση 1 - τα **Modular PLC** (συνήθως τα μεγάλα PLC, σχήμα 4)

Σ' αυτή την περίπτωση το PLC πωλείται σε modular μορφή, δηλαδή κομμάτι-κομμάτι. Τα βασικά κομμάτια ενός τέτοιου PLC είναι:

- **Η μονάδα τροφοδοσίας.**
- **Η κεντρική μονάδα επεξεργασίας,** η οποία έχει τη δυνατότητα να οδηγήσει ένα ανώτατο αριθμό εισόδων και εξόδων. Π.χ. το PLC SIMATIC S7-300 (CPU 316) της SIEMENS μπορεί να οδηγήσει μέχρι 1024 ψηφιακές εισόδους και εξόδους (συνολικά).

- **Οι μονάδες εισόδων και εξόδων.** Στα modular PLC πωλούνται και αυτές σε κομμάτια-μονάδες. Κάθε μονάδα εισόδων (ή εξόδων) μπορεί να έχει 4, 8, 16 ή 32 εισόδους (ή εξόδους). Μ' αυτό τον τρόπο μπορούμε να επιλέγουμε μία μονάδα εισόδων ή εξόδων η οποία να έχει τα τεχνικά χαρακτηριστικά που επιθυμούμε. Γίνεται κατανοητό ότι σε ένα modular PLC μπορούμε να έχουμε μονάδες εισόδων ή εξόδων που να λειτουργούν σε διαφορετικές τάσεις τροφοδοσίας.

Περίπτωση 2 - Συμπαγή PLC (μικρά συνήθως, σχήμα 5)

Όλες οι εταιρείες διαθέτουν και μικρά PLC, στα οποία όλες οι μονάδες τους (τροφοδοσίας, κεντρική μονάδα και μονάδες εισόδων - εξόδων) είναι ενσωματωμένες σε μια συσκευή. Σ' αυτού του είδους τα PLC είσοδοι και έξοδοι είναι συνήθως μέχρι 20 και όλες οι είσοδοι (ή έξοδοι) έχουν τα ίδια τεχνικά χαρακτηριστικά.

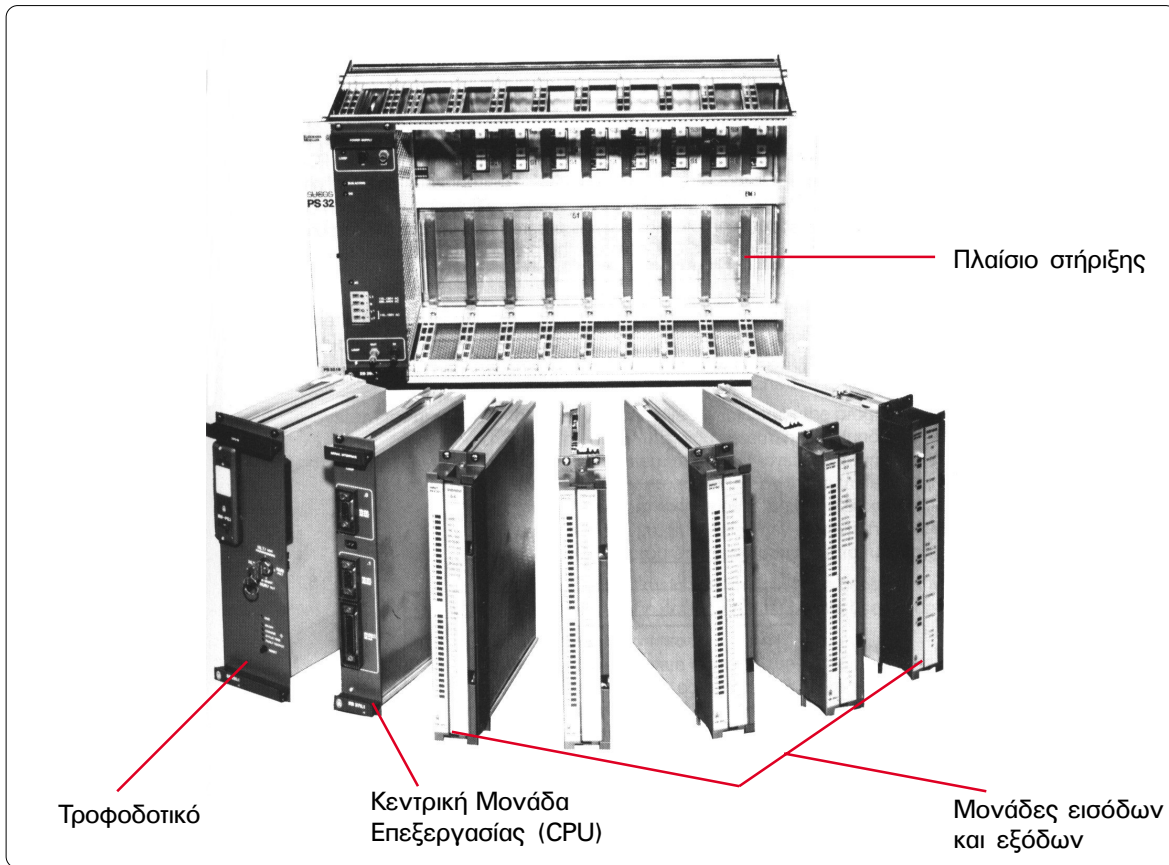
Το σημείο που πρέπει κάποιος να προσέξει σχετικά με τις εισόδους και εξόδους είναι ότι κάθε είσοδος ή έξοδος είναι για το PLC ακριβώς καθορισμένη, δηλαδή έχει καθορισμένο όνομα με το οποίο αναφέρεται και στο πρόγραμμα. Στα συμπαγή PLC σε κάθε ακροδέκτη αναγράφεται το όνομα της εισόδου ή της εξόδου. Στα modular PLC υπάρχει σαφές σύστημα με το οποίο αναγνωρίζουμε το όνομα της εισόδου (ή εξόδου) σε κάθε ακροδέκτη μιας μονάδας εισόδων (ή εξόδων).

2.4 Αρχή λειτουργίας ενός προγραμματιζόμενου λογικού ελεγκτή.

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

Βήμα 1ο. Στην αρχή ο μικροεπεξεργαστής “διαβάζει” τις 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 μνήμης η οποία ονομάζεται **εικόνα εισόδων**. Την εικόνα εισόδων μπορείτε να τη φανταστείτε σαν ένα πίνακα, όπου ο μικροεπεξεργαστής “σημειώνει” τις τιμές, που διάβασε. Π.χ. είσοδος I1=“1”, I2=“0”, I3=“0” κ.ο.κ.

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.



Σχήμα 4: Modular PLC. Αποτελείται από ανεξάρτητες μονάδες οι οποίες προσαρμόζονται στο πλαίσιο στήριξης.

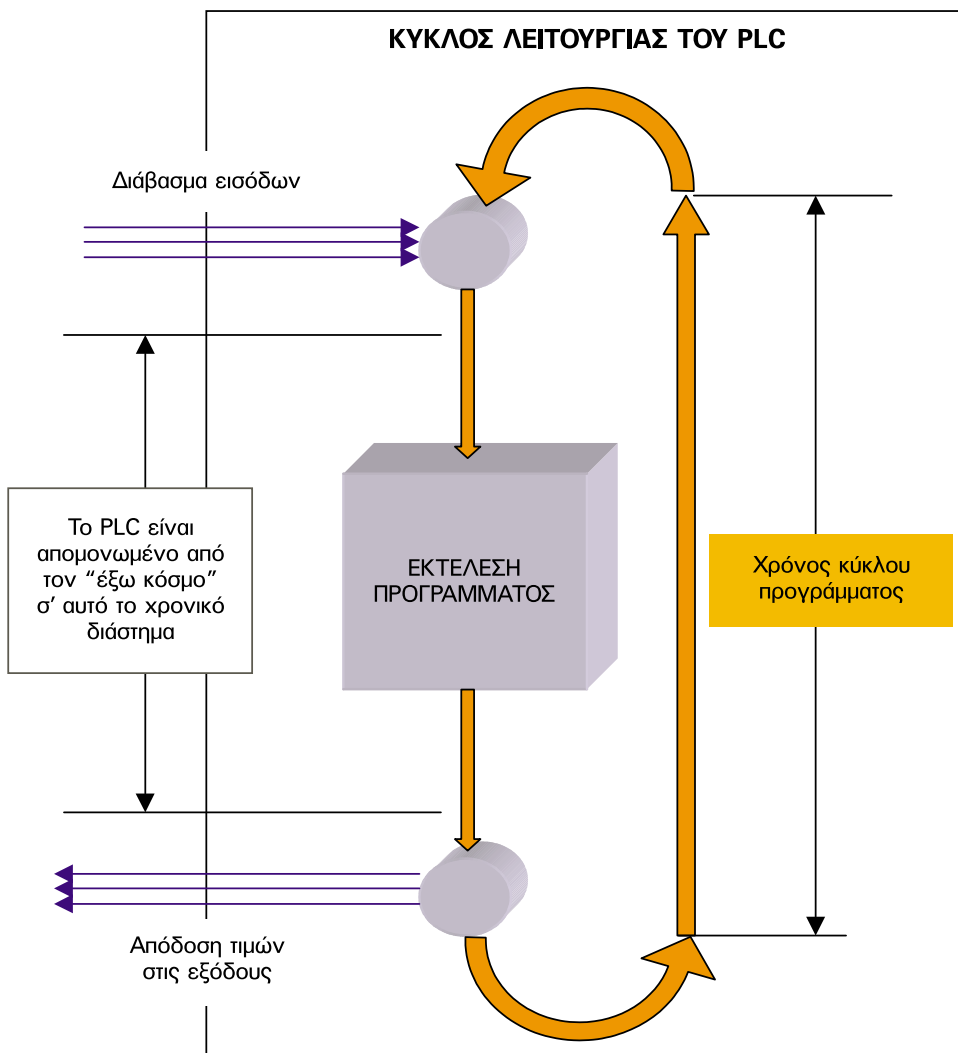


Σχήμα 5: Συμπαγές PLC. Περιλαμβάνει τροφοδοτικό, κεντρική μονάδα επεξεργασίας, εισόδους και εξόδους, όλα ενσωματωμένα σε μια ενιαία συσκευή.

Η εκτέλεση του προγράμματος θα δώσει αποτελέσματα για τις εξόδους. Τα αποτελέσματα αυτά αποθηκεύονται στην ειδική περιοχή της μνήμης που ονομάζεται **εικόνα εξόδων**. Όπως η εικόνα εισόδων, η εικόνα εξόδων περιέχει την τιμή ("0" ή "1") για κάθε έξοδο, π.χ. Q1="1", Q2="1", Q3="0" κ.ο.κ. Σημειώνουμε ότι οι τιμές αυτές προκύπτουν από την εκτέλεση των λογικών πράξεων του προγράμματος.

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

Με τη συμπλήρωση του 3^{ου} βήματος συμπληρώνεται ένας πλήρης κύκλος λειτουργίας και η διαδικασία ξαναρχίζει 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.



Σχήμα 6: Κύκλος λειτουργίας PLC.

Ο χρόνος που χρειάζεται για να εκτελέσει το PLC ένα πλήρη κύκλο λειτουργίας ονομάζεται **χρόνος κύκλου** και εξαρτάται από τη “ταχύτητα” του μικροεπεξεργαστή 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 και ένα μέτρο σύγκρισης μεταξύ των PLC. Για να μπορούν να συγκριθούν τα PLC ως προς τη ταχύτητα εκτέλεσης ενός προγράμματος, ορίζουμε το **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 μερικές εκατοντάδες χιλιοστά του δευτερολέπτου.

Θα θέλαμε στο σημείο αυτό να τονίσουμε την ουσιαστική διαφορά στη λειτουργία ενός αυτοματισμού με PLC από ένα κλασικό αυτοματισμό με ηλεκτρονόμους.

Στην περίπτωση του κλασικού αυτοματισμού, όταν έχουμε αλλαγή της κατάστασης ενός διακόπτη εισόδου, η αλλαγή αυτή προκαλεί εκείνη τη στιγμή διαδοχικές αλλαγές στα στοιχεία του κυκλώματος που τροφοδοτούνται από το συγκεκριμένο διακόπτη. Έχουμε δηλαδή διαδικασία που συμβαίνει σε πραγματικό χρόνο.

Αν μελετήσουμε τον κύκλο λειτουργία του PLC, θα δούμε ότι το PLC “δεν βλέπει” συνεχώς τον “έξω κόσμο” (την εξωτερική εγκατάσταση), παρά μόνο κατά τα χρονικά διαστήματα που διαβάζει τις εισόδους και αποδίδει τιμές στις εξόδους. Στον υπόλοιπο χρόνο του κύκλου, το PLC είναι ένας υπολογιστής ο οποίος εκτελεί πράξεις (λογικές βέβαια ή και αριθμητικές) απομονωμένο από τον έξω κόσμο. Για να γίνει αυτό κατανοητό υποθέστε ότι αλλάζει η κατάσταση μιας εισόδου, κατά την διάρκεια του χρόνου κατά τον οποίον εκτελούνται οι εντολές προγράμματος. Στην περίπτωση αυτή στο τέλος του κύκλου το PLC θα δώσει αποτελέσματα στις εξόδους, στα οποία δεν θα έχει ληφθεί υπ’ όψη η αλλαγή στην κατάσταση της συγκεκριμένης εισόδου. Αυτό γιατί το PLC θα ενημερώσει την εικόνα των εισόδων για την αλλαγή στην κατάσταση της συγκεκριμένης εισόδου στην αρχή του επόμενου κύκλου επεξεργασίας του προγράμματος. Το PLC θα δώσει αποτελέσματα στις εξόδους, στα οποία θα έχει ληφθεί υπ’ όψη η αλλαγή στην κατάσταση της παραπάνω εισόδου στο τέλος του επόμενου κύκλου επεξεργασίας του προγράμματος. Λαμβάνοντας τα παραπάνω υπ’ όψη θα έλεγε κάποιος ότι τελικά το PLC ανταποκρίνεται πολύ καθυστερημένα στις αλλαγές μιας αυτοματοποιημένης διαδικασίας. Όμως, αυτό δεν είναι η πραγματικότητα, αφού ο χρόνος πραγματοποίησης ενός κύκλου προγράμματος από ένα PLC είναι πάρα πολύ μικρός, το πολύ 300ms σε πολύπλοκες εγκαταστάσεις αυτοματισμού.

3. Προγραμματισμός ενός προγραμματιζόμενου λογικού ελεγκτή.

Το βασικότερο κομμάτι σε ένα σύστημα αυτοματισμού με PLC δεν είναι το υλικό μέρος αλλά το λογισμικό, δηλαδή το πρόγραμμα που υλοποιεί τον επιθυμητό αυτοματισμό. Το

πρόγραμμα αναπτύσσεται σε μια γλώσσα προγραμματισμού. Δυστυχώς στα PLC δεν υπήρξε τυποποίηση σε κανέναν τομέα, λόγω του ανταγωνισμού των εταιρειών, ούτε βέβαια στο θέμα των γλωσσών προγραμματισμού. Δηλαδή δεν υπάρχουν γλώσσες προγραμματισμού για PLC που να ισχύουν ανεξάρτητα από εταιρεία, όπως για παράδειγμα συμβαίνει στον προγραμματισμό των ηλεκτρονικών υπολογιστών. Παρ' όλα αυτά οι γλώσσες των PLC των διαφόρων εταιρειών μοιάζουν πολύ μεταξύ τους, έτσι που να μπορούμε να μιλάμε σήμερα για μια "τυποποίηση της αγοράς".

3.1 Γλώσσες προγραμματισμού των προγραμματιζόμενων λογικών ελεγκτών.

Τρεις είναι σήμερα οι κυριότερες κατηγορίες γλωσσών προγραμματισμού για PLC, τις οποίες συναντούμε με μικρές διαφορές στα PLC όλων των εταιρειών:

1. Γλώσσα LADDER ή γλώσσα ηλεκτρολογικών γραφικών (σχήμα 7).

Είναι η πρώτη γλώσσα που αναπτύχθηκε ιστορικά. Η γλώσσα Ladder στην ουσία επιτρέπει τη μεταφορά του ηλεκτρολογικού σχεδίου, μέσω της συσκευής προγραμματισμού στο PLC. Με τη γλώσσα αυτή η εκπαίδευση των τεχνικών, που ήταν συνηθισμένοι στον κλασικό αυτοματισμό, γινόταν εύκολα και γρήγορα, αφού δεν άλλαζε ουσιαστικά την εργασία σχεδιασμού του αυτοματισμού. Η γλώσσα LADDER χρησιμοποιεί όχι την Ευρωπαϊκή τυποποίηση στο σχεδιασμό των ηλεκτρικών επαφών, αλλά την Αμερικάνικη. 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.

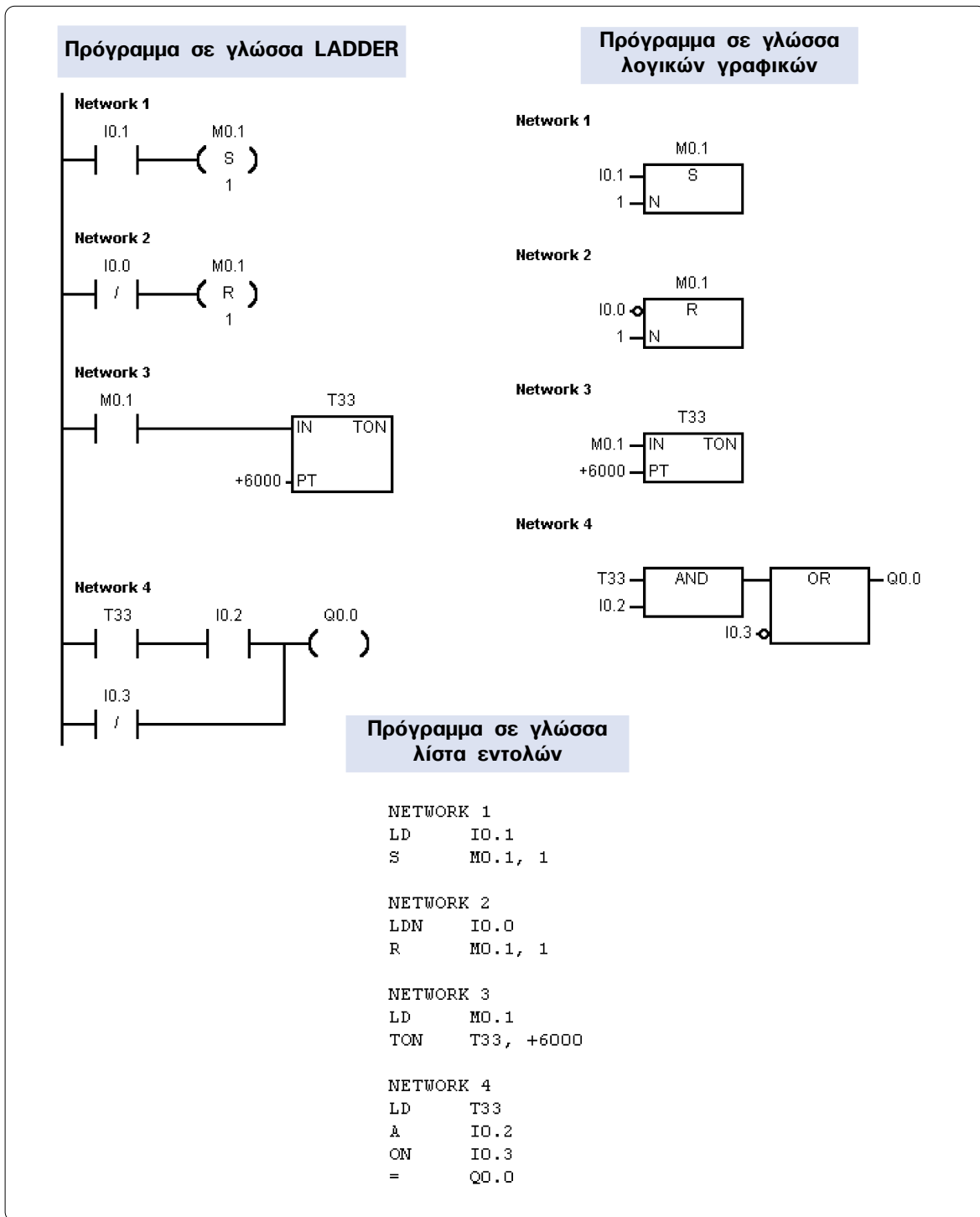
2. Γλώσσα λίστα εντολών (Statement List, STL) ή γλώσσα λογικών εντολών (σχήμα 7).

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. Η γλώσσα αυτή δημιουργεί λίστα προγράμματος με εντολές, οι οποίες αντιστοιχούν στις λογικές πύλες (AND, OR, NOT κ.λπ.). 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.

3. Γλώσσα λογικών γραφικών ή λογικού διαγράμματος (σχήμα 7).

Η γλώσσα αυτή είναι επίσης γραφική, αλλά αντί του ηλεκτρολογικού σχεδίου του

αυτοματισμού, χρησιμοποιεί το αντίστοιχο λογικό κύκλωμα. Η γλώσσα αυτή είναι νεότερη και δεν χρησιμοποιείται από όλες τις εταιρείες.



Σχήμα 7: Γλώσσες προγραμματισμού PLC.

3.2 Συσκευές προγραμματισμού των προγραμματιζόμενων λογικών ελεγκτών.

Αφού συντάξουμε το πρόγραμμα στο “χαρτί” σε οποιαδήποτε γλώσσα προγραμματισμού, πρέπει να το εισάγουμε στο PLC. Αυτό συνήθως γίνεται μέσω μιας συσκευής προγραμματισμού, ενός “προγραμματιστή”, που συνδέεται με το PLC. Ορισμένα μικρά PLC προγραμματίζονται με τη βοήθεια ενός αριθμού πλήκτρων που είναι ενσωματωμένα επάνω στη συσκευή του PLC και δε χρειάζονται συσκευή προγραμματισμού (βλέπε σχήματα 52 και 53 στη σελίδα 64).

Μια συσκευή προγραμματισμού μπορεί να είναι μιας από τις παρακάτω μορφές:

1) Ειδικός προγραμματιστής χειρός.

Κάθε PLC συνοδεύεται από μια ειδική συσκευή προγραμματιστή, η οποία είναι συνήθως “χειρός”, δηλαδή φορητή. Αυτές οι συσκευές προγραμματισμού διαθέτουν μια μικρή οθόνη υγρών κρυστάλλων και τυποποιημένα πλήκτρα προγραμματισμού. Συνήθως οι ειδικοί “προγραμματιστές” μπορούν να προγραμματίσουν τα PLC μόνο σε γλώσσα λίστα εντολών. Υπάρχουν όμως και “προγραμματιστές” με τους οποίους μπορούμε να προγραμματίσουμε και σε κάποια από τις γραφικές γλώσσες. Για να προγραμματίσουμε το PLC πρέπει να το συνδέσουμε με τον προγραμματιστή. Η σύνδεση πραγματοποιείται μέσω της ειδικής θύρας που υπάρχει στην κεντρική μονάδα επεξεργασίας του PLC. Αφού πληκτρολογήσουμε το πρόγραμμα, το μεταφέρουμε στη μνήμη του PLC. Όταν ολοκληρώσουμε τη διαδικασία αυτή, ο προγραμματιστής μπορεί να αποσυνδεθεί. Ο τρόπος χειρισμού του προγραμματιστή είναι τελείως ειδικός για κάθε PLC. Οι προγραμματιστές των διαφόρων εταιριών δεν μοιάζουν πολύ μεταξύ τους και αυτό είναι μια δυσκολία στην εκμάθηση του προγραμματισμού ενός νέου PLC.



Σχήμα 8: Προγραμματιστής χειρός PLC.

Οι προγραμματιστές χειρός σήμερα διαθέτουν και άλλες δυνατότητες, όπως για παράδειγμα:

- Μπορούν να συνδεθούν με εκτυπωτή, για να εκτυπώσουμε το πρόγραμμα.
- Μπορούν να συνδεθούν με προσωπικό υπολογιστή (PC) με όσα πλεονεκτήματα μπορεί αυτό να έχει, π.χ. μπορούμε να αποθηκεύσουμε σε δισκέτα το πρόγραμμα, να κάνουμε εκτύπωση του προγράμματος κ.λπ.
- Μπορούν να συνδεθούν με ειδική συσκευή προγραμματισμού EEPROM, με την οποία μπορούμε να θέτουμε το πρόγραμμα σε πλακέτες EEPROM.
- Ακόμη με τον προγραμματιστή μπορούμε να ελέγχουμε την λειτουργία του προγράμματος-αυτοματισμού και να κάνουμε ανίχνευση βλαβών. Αυτή η δυνατότητα είναι ίσως το σημαντικότερο πλεονέκτημα του προγραμματιστή χειρός, γιατί μπορεί να μεταφερθεί σε οποιαδήποτε εγκατάσταση PLC, να συνδεθεί στο PLC και να ψάξουμε για βλάβες στη λειτουργία του αυτοματισμού.



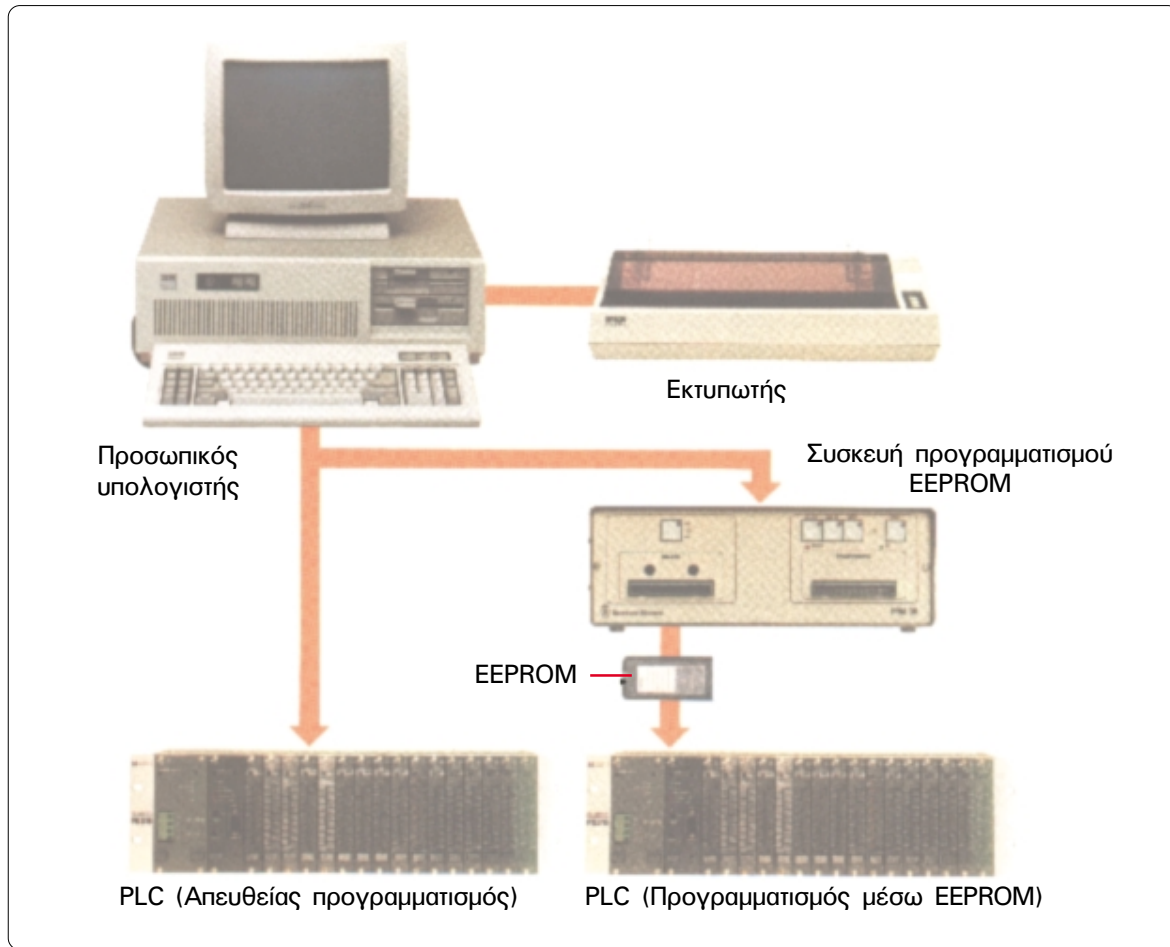
Σχήμα 9: Έλεγχος βλαβών σε εγκατάσταση αυτοματισμού με PLC.

2) Προσωπικός υπολογιστής (PC) και χρήση ειδικού λογισμικού.

Ο πιο εύκολος τρόπος προγραμματισμού ενός PLC σήμερα είναι μέσω ενός προσωπικού υπολογιστή (PC). Με την χρήση ειδικού λογισμικού, το οποίο δίνεται από την εταιρεία, το PC μετατρέπεται σε “προγραμματιστή”. Για τη σύνδεση του PC με το PLC ή με την συσκευή προγραμματισμού EEPROM χρειάζεται ειδική κάρτα σύνδεσης (interface), η οποία τοποθετείται στο PC. Ο προγραμματισμός μέσω PC είναι πολύ ευκολότερος από τον προγραμματισμό με τον ειδικό προγραμματιστή χειρός, ειδικά για κάποιον που είναι εξοικειωμένος με την χρήση του PC. Ο προγραμματισμός στις γραφικές γλώσσες γίνεται με τρόπο ιδανικό στην οθόνη του PC.

Τα υπόλοιπα πλεονεκτήματα νομίζουμε είναι προφανή, δηλαδή:

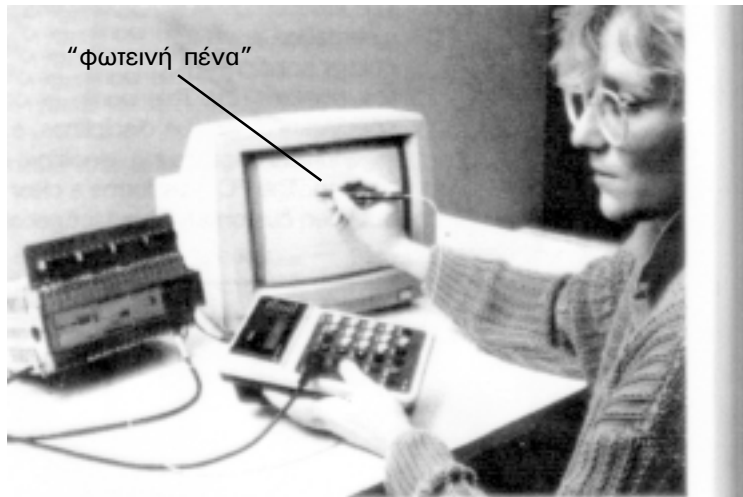
- Μπορούμε να αποθηκεύουμε και να αρχειοθετούμε τα προγράμματά μας.
- Μπορούμε να τυπώνουμε τα προγράμματα.



Σχήμα 10: Προγραμματισμός PLC με την βοήθεια προσωπικού υπολογιστή.

3) Ειδικές συσκευές προγραμματισμού.

Εκτός από τις δύο μορφές που προαναφέραμε, υπάρχουν κάποιες ειδικές συσκευές με τις οποίες ο προγραμματισμός, κυρίως στις γραφικές γλώσσες γίνεται πολύ εύκολα. Μία τέτοια συσκευή είναι η “φωτεινή πένα” (light pen). Πρόκειται για μια συσκευή η οποία περιλαμβάνει μια οθόνη, επάνω στην οποία σχεδιάζουμε με μια ειδική “φωτεινή πένα” (σχήμα 11).



Σχήμα 11: Προγραμματισμός PLC με “φωτεινή πένα” (Light Pen).

3.3 Τρόπος ανάπτυξης ενός συστήματος αυτοματισμού με τη χρήση προγραμματιζόμενου λογικού ελεγκτή.

Είδαμε ότι αρχικά οι εταιρείες παραγωγής PLC προσπάθησαν να μη διαταράξουν τον τρόπο σχεδιασμού των αυτοματισμών, δηλαδή όλες οι γλώσσες προγραμματισμού στηρίχθηκαν στην ανάπτυξη του προγράμματος από το υπάρχον σχέδιο του κλασικού αυτοματισμού της καλωδιωμένης λογικής. Αντί να προχωρήσουμε στην κατασκευή της πολύπλοκης καλωδίωσης του πίνακα, προχωρούσαμε στον προγραμματισμό του PLC, μετατρέποντας στην ουσία το κλασικό σχέδιο σε πρόγραμμα. Όλες οι γλώσσες προγραμματισμού, είτε επρόκειτο για τη γλώσσα λίστα εντολών είτε για τις γραφικές, είχαν προσαρμοστεί σε αυτό το μοντέλο. Με την πάροδο του χρόνου οι γλώσσες προγραμματισμού εξελίχθηκαν, απομακρύνονται όλο και περισσότερο από το μοντέλο του ηλεκτρολογικού σχεδίου και πλησιάζουν τις ανώτερες γλώσσες προγραμματισμού. Για παράδειγμα οι εντολές ελέγχου και διακλάδωσης (π.χ. IF..., GO TO...) και υποπρογραμμάτων (π.χ. GO SUB..., CALL... κ.λπ.) είναι από τις σημαντικότερες εντολές στις ανώτερες γλώσσες προγραμματισμού. Τέτοιου είδους εντολές δεν υπήρχαν αρχικά στις γλώσσες των PLC. Σιγά-σιγά όμως αρχίζουν να εμφανίζονται πολλές τέτοιες εντολές στις νεότερες εκδόσεις όλων των γλωσσών. Η εξέλιξη αυτή έχει αρχίσει να αλλάζει τον τρόπο σχεδιασμού και ανάπτυξης των αυτοματισμών με PLC. Ήδη υπάρχει μια κατηγορία νέων σχεδιαστών αυτοματισμού, οι οποίοι δουλεύουν περισσότερο σαν προγραμματιστές.

4. Ανάπτυξη προγράμματος σε προγραμματιζόμενο λογικό ελεγκτή.

4.1 Γενικά.

Στις ενότητες που ακολουθούν θα δούμε, πως προγραμματίζουμε ένα PLC. Για τις απαιτήσεις του μαθήματος αυτού θα θεωρήσουμε ότι διαθέτουμε το ηλεκτρολογικό σχέδιο του αυτοματισμού, για τον οποίο θέλουμε να αναπτύξουμε πρόγραμμα σε PLC. Πρέπει να σημειώσουμε εδώ ότι το ηλεκτρολογικό σχέδιο ενός αυτοματισμού δεν είναι απαραίτητο προκειμένου να αναπτυχθεί πρόγραμμα σε PLC για τον αυτοματισμό. Ένας σχετικά έμπειρος στον προγραμματισμό τεχνικός δεν “περνά” από το ηλεκτρολογικό σχέδιο του αυτοματισμού, προκειμένου να αναπτύξει το πρόγραμμα. Σε σύνθετους (πολύπλοκους) αυτοματισμούς η ανάπτυξη του ηλεκτρολογικού σχεδίου του αυτοματισμού είναι πολύ δυσκολότερη από την ανάπτυξη του προγράμματος. Μάλιστα, σε ορισμένες περιπτώσεις είναι πιο εύκολο να αναπτυχθεί το πρόγραμμα άμεσα από τα δεδομένα του αυτοματισμού παρά χρησιμοποιώντας ένα έτοιμο ηλεκτρολογικό σχέδιο του αυτοματισμού.

Το πρόβλημα που υπάρχει σχετικά με τον προγραμματισμό των PLC είναι αυτό στο οποίο έχουμε ήδη αναφερθεί, δηλαδή το γεγονός ότι οι γλώσσες προγραμματισμού των PLC δεν είναι τυποποιημένες, αλλά διαφέρουν από εταιρεία σε εταιρεία. Διαφέρουν ακόμη και μεταξύ των μοντέλων της ίδιας εταιρείας. Βέβαια η λογική όλων των γλωσσών προγραμματισμού σε όλα τα PLC είναι ίδια. Αλλά και οι εντολές προγραμματισμού στις διάφορες γλώσσες μοιάζουν μεταξύ τους σε ένα σημαντικό ποσοστό. Έτσι, όποιος μάθει να χρησιμοποιεί πολύ καλά τις γλώσσες προγραμματισμού ενός μοντέλου PLC, αρκετά εύκολα μαθαίνει τις γλώσσες προγραμματισμού ενός άλλου μοντέλου, εντοπίζοντας πολύ γρήγορα τις διαφορές.

Θα παρουσιάσουμε τον προγραμματισμό των PLC σε δύο ενότητες. Στην πρώτη ενότητα θα δούμε πως προγραμματίζουμε σε ένα PLC συνδυαστικούς αυτοματισμούς και στη δεύτερη ενότητα πως προγραμματίζουμε ακολουθιακούς αυτοματισμούς. Αυτό το κάνουμε, γιατί οι βασικές διαφορές στον προγραμματισμό των PLC εμφανίζονται, όταν έχουμε χρήση χρονικών, απαριθμητών και των λοιπών ειδικών συναρτήσεων των ακολουθιακών αυτοματισμών.

Τα διάφορα προγράμματα θα αναπτυχθούν και στις τρεις βασικές γλώσσες προγραμματισμού, που αναφέραμε προηγούμενα.

Συνδυαστικοί και ακολουθιακοί αυτοματισμοί

Υιοθετώντας την ορολογία από τα λογικά κυκλώματα, ονομάζουμε:

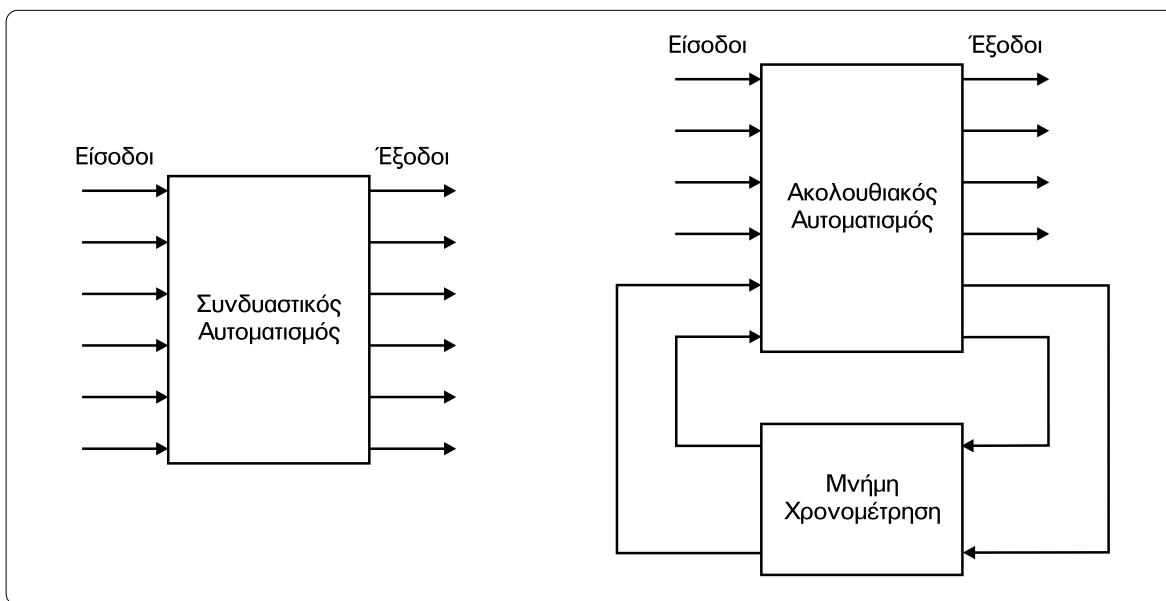
Συνδυαστικό αυτοματισμό, τον αυτοματισμό εκείνο στον οποίον οι έξοδοι εξαρτώνται μόνο από τις εισόδους. Αυτό σημαίνει ότι οι κινητήρες, βαλβίδες και οι λοιποί αποδέκτες του αυτοματισμού λαμβάνουν εντολές μόνο από τους αισθητήρες και τους διακόπτες εισόδου και δεν εξαρτώνται από το χρόνο ή από προηγούμενες καταστάσεις των εξόδων.

Ακολουθιακό αυτοματισμό, τον αυτοματισμό εκείνο, στον οποίο οι εξοδοί εξαρτώνται όχι μόνο από τις εισόδους, αλλά και από το χρόνο ή και από προηγούμενες καταστάσεις των εξόδων.

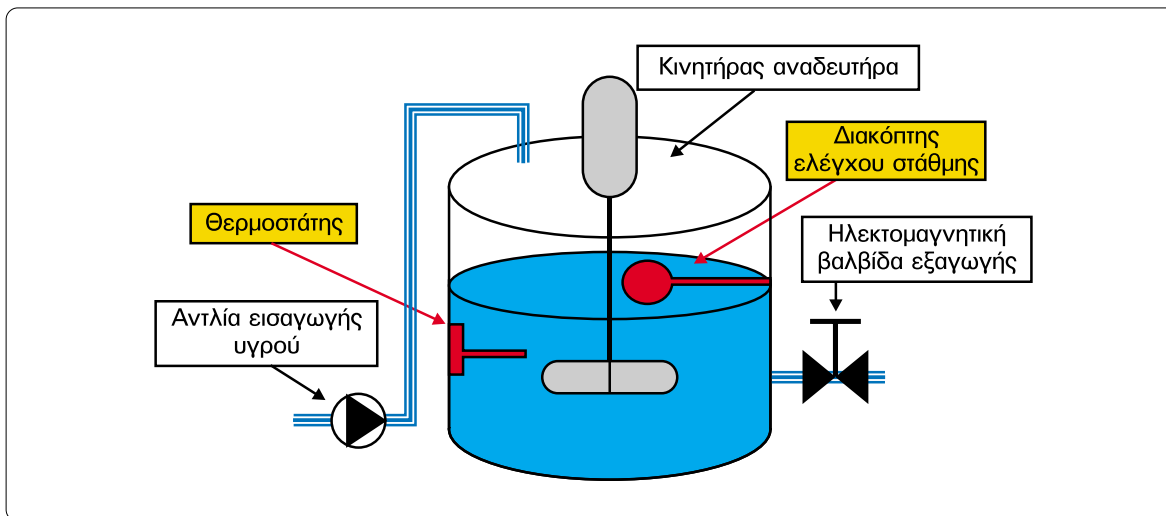
Σχηματικά οι δύο κατηγορίες αυτοματισμών φαίνονται στο σχήμα 12.

Παράδειγμα συνδυαστικού και ακολουθιακού αυτοματισμού.

Ο αυτοματισμός του σχήματος 13 που περιγράψαμε στην ενότητα 2.9 του Α' τόμου (σελίδες 76-82) είναι συνδυαστικός αυτοματισμός, γιατί οι καταστάσεις των εξόδων (της αντλίας, του αναδευτήρα και της βαλβίδας εξαγωγής) εξαρτώνται μόνο από τις καταστάσεις των εισόδων (του πλωτήρα και του θερμοστάτη).



Σχήμα 12



Σχήμα 13

Ο αυτοματισμός θα ήταν ακολουθιακός, αν θεωρήσουμε ότι ο αναδευτήρας θα λειτουργήσει για κάποιο σταθερό χρόνο T , ανεξάρτητα από την κατάσταση του θερμοστάτη.

4.2 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 εισόδοι σχεδόν σε όλα τα PLC χαρακτηρίζονται με το γράμμα I (Input). Στα μικρά συμπαγή PLC το γράμμα I ακολουθεί ένας απλός αύξοντας αριθμός, ξεκινώντας από το 1 (ή το 0) και φθάνοντας στο πλήθος των εισόδων π.χ. I1, I2, I3, κ.λπ. Στα modular PLC, όπου οι εισόδοι βρίσκονται σε μονάδες εισόδων, το γράμμα I ακολουθούν δύο αριθμοί, που χωρίζονται με μια τελεία. Ο πρώτος αριθμός χαρακτηρίζει συνήθως τη θέση (τη σειρά) της μονάδας, που βρίσκεται η είσοδος, και ο δεύτερος αριθμός χαρακτηρίζει την είσοδο πάνω στην μονάδα (σχήμα 14). Π.χ. έχουμε εισόδους I 0.0, I 0.1, I 0.2, ..., I 1.1, I 1.2, κ.λπ.

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

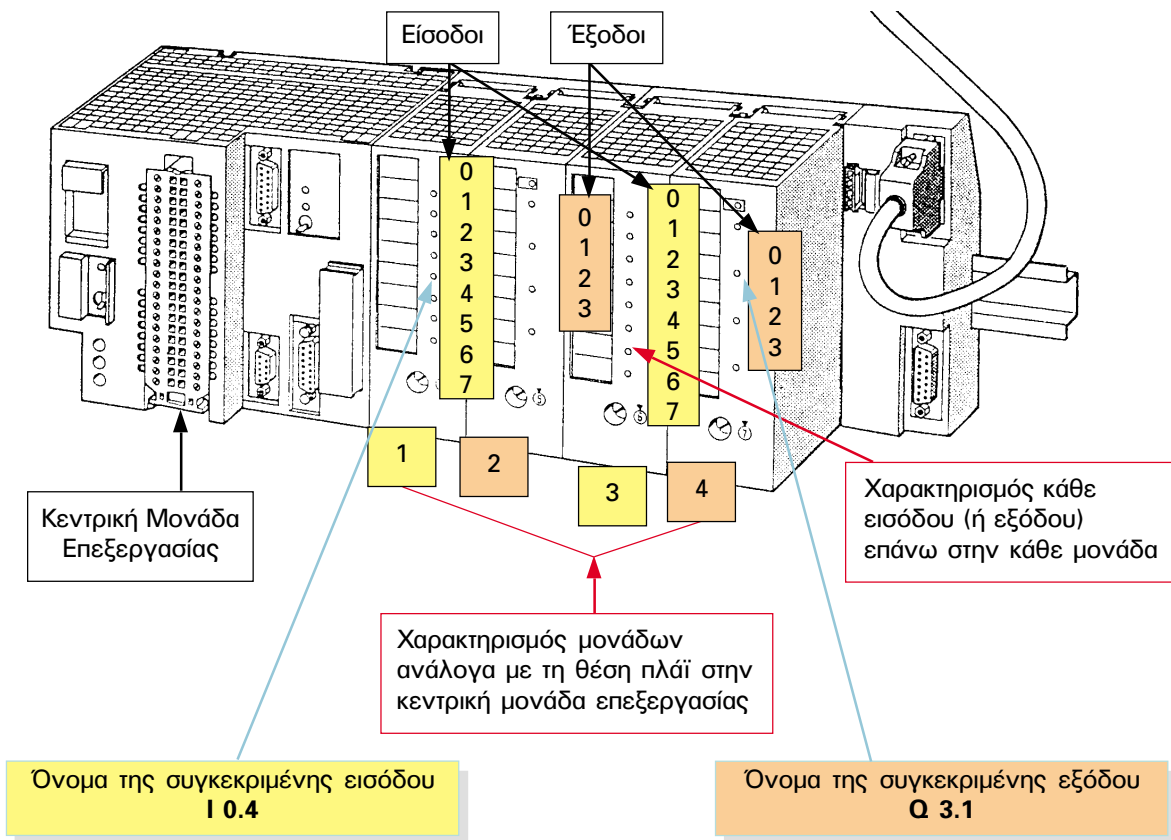
Τα ίδια, που ισχύουν για τις εισόδους, ισχύουν και για τις εξόδους. Το γράμμα με το οποίο χαρακτηρίζονται οι εξόδοι στα διάφορα PLC είναι συνήθως το Q ή το O (Output). Για τους αριθμούς, που ακολουθούν το γράμμα, ισχύει ότι και για τις εισόδους.

- Πόσες βοηθητικές μνήμες έχει και πως τις ονομάζουμε.

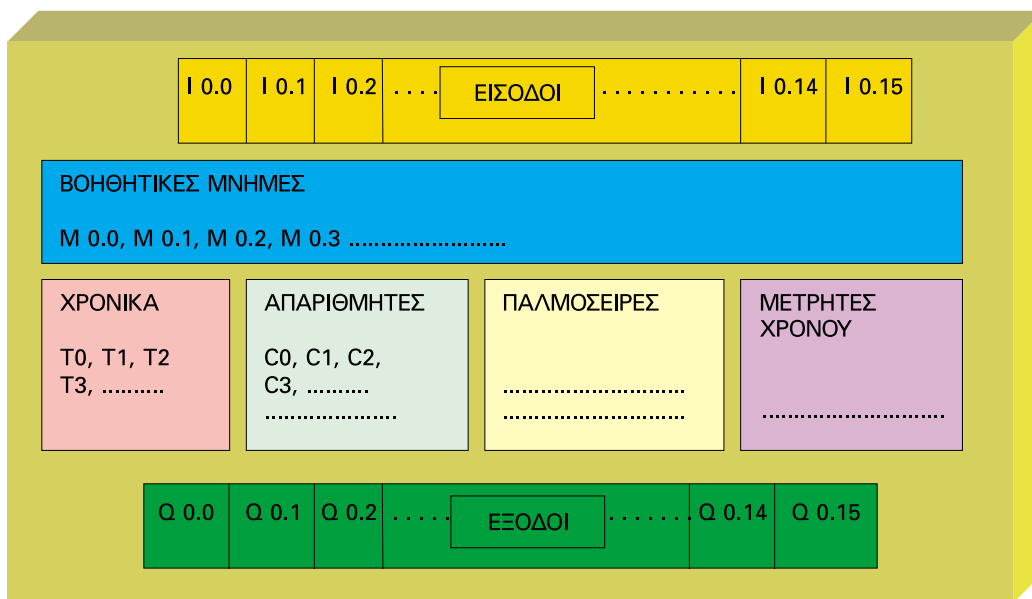
Στα διάφορα PLC θα τις συναντήσουμε με το όνομα Markers ή Flags. Πρόκειται για θέσεις μνήμης, στις οποίες αποθηκεύονται ενδιάμεσες λογικές καταστάσεις και πληροφορίες. Όπως ισχύει για τις εισόδους και τις εξόδους, χαρακτηρίζονται με ένα γράμμα ακολουθούμενο από έναν αριθμό ή δύο αριθμούς, που χωρίζονται με τελεία. Το γράμμα στα διάφορα PLC είναι το M (Marker) ή το F (Flag). Συνήθως οι βοηθητικές μνήμες σε ένα PLC δεν είναι λιγότερες από 255. Έτσι, έχουμε βοηθητικές μνήμες:

M 0.1 έως M 0.15
M 1.0 έως M 1.15	M 31.0 έως M 31.15
.....	

Ο λόγος, για τον οποίον εξελίσσονται οι αριθμοί της ονοματολογίας με αυτό τον τρόπο, δεν είναι αντικείμενο του παρόντος να το εξηγήσουμε.



Σχήμα 14: Ονοματολογία σε modular PLC.



Σχήμα 15: Προγραμματιστικό μοντέλο ενός PLC.

• 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.

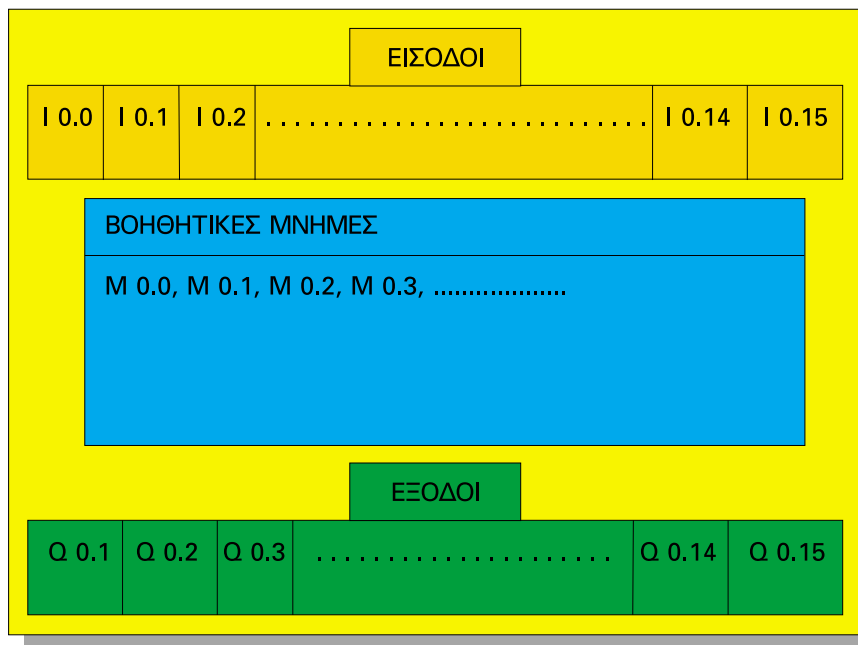
Όλα τα παραπάνω στοιχεία λέμε ότι αποτελούν το προγραμματιστικό μοντέλο ενός PLC. Για να ξεκινήσουμε τον προγραμματισμό πρέπει να γνωρίζουμε το προγραμματιστικό μοντέλο του συγκεκριμένου PLC, που διαθέτουμε.

Στη συνέχεια θα χρησιμοποιήσουμε το υποθετικό μοντέλο PLC του σχήματος 16, του οποίου τα χαρακτηριστικά ομοιάζουν αρκετά με τα χαρακτηριστικά των PLC της αγοράς. Για το υποθετικό μοντέλο PLC θεωρούμε:

Είσοδοι: I 0.1, I 0.2, έως I 0.15

Έξοδοι: Q 0.1, Q 0.2, έως Q 0.15

Βοηθητικές μνήμες: M 0.0, M 0.1, M 0.2, M 0.3, ..., M 0.15, M 1.0, M 1.1, ..., M 15.14, M 15.15



Σχήμα 16: Προγραμματιστικό μοντέλο του PLC που θα χρησιμοποιήσουμε στον προγραμματισμό.

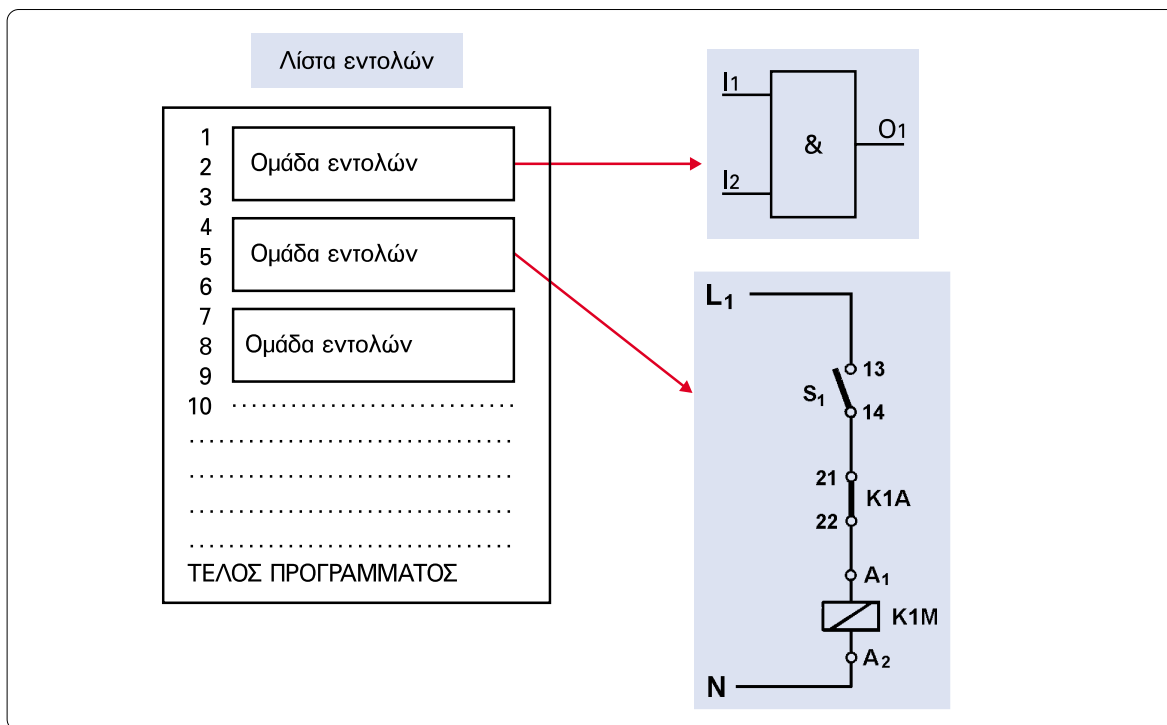
Τις ειδικές συναρτήσεις του υποθετικού μοντέλου θα τις δούμε αργότερα, όταν θα μελετήσουμε παραδείγματα ακολουθιακών αυτοματισμών.

5. Ανάπτυξη προγράμματος σε γλώσσα λίστα εντολών.

5.1 Βασικές εντολές προγραμματισμού στη γλώσσα λίστα εντολών.

Μορφή του προγράμματος.

Το πρόγραμμα αποτελείται από μια σειρά εντολών. Κάθε εντολή αποτελεί μια γραμμή προγράμματος. Οι εντολές κατανέμονται σε ομάδες εντολών. Κάθε ομάδα εντολών αντιστοιχεί σε μία λογική πύλη, ή αλλιώς σε ένα κλάδο του ηλεκτρολογικού κυκλώματος (σχήμα 17).



Σχήμα 17: Μορφή προγράμματος στη γλώσσα λίστα εντολών.

Μορφή εντολής.

Κάθε εντολή του προγράμματος αποτελείται από δύο μέρη (σχήμα 18). Το πρώτο μέρος καθορίζει την ενέργεια την οποία θα εκτελέσει το PLC, δηλαδή χαρακτηρίζει την ίδια την εντολή. Το δεύτερο μέρος καθορίζει την παράμετρο, δηλαδή καθορίζει σε ποια είσοδο, έξοδο, βοηθητική μνήμη κ.λπ. αναφέρεται η ενέργεια της εντολής.



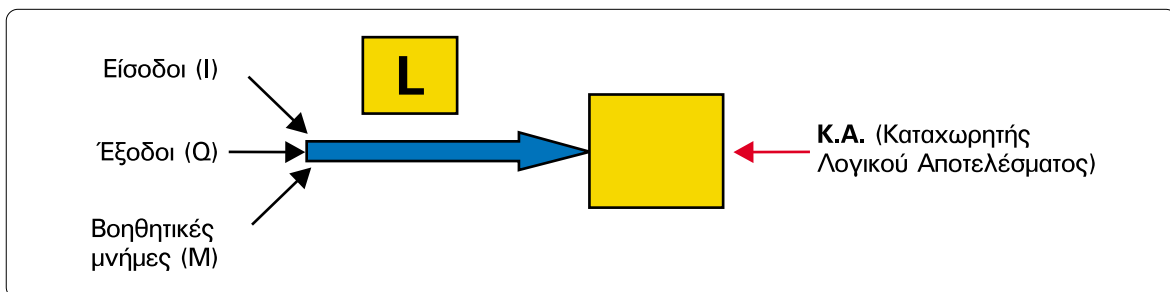
Σχήμα 18: Μορφή εντολής στη γλώσσα λίστα εντολών.

Παρουσίαση εντολών.

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.)**.

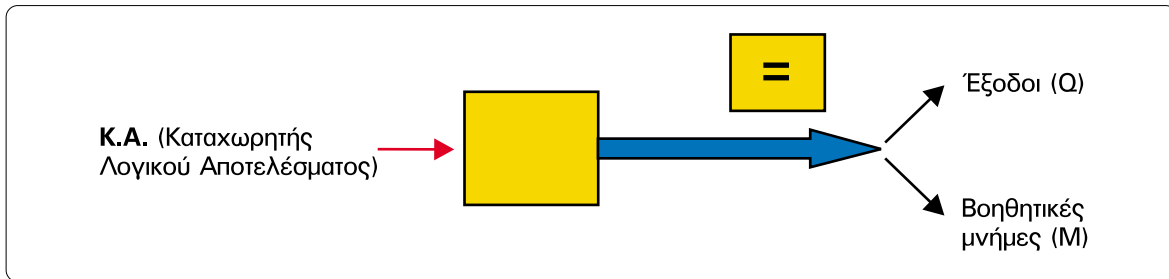
Η εντολή L μπορεί να αναφέρεται σε εισόδους, εξόδους, βοηθητικές μνήμες, χρονικά, κ.λπ. Π.χ. L I 0.1, L Q 0.2, L M 0.1.



Σχήμα 19: Σχηματική παράσταση της εντολής Load.

Η εντολή = (ίσον)

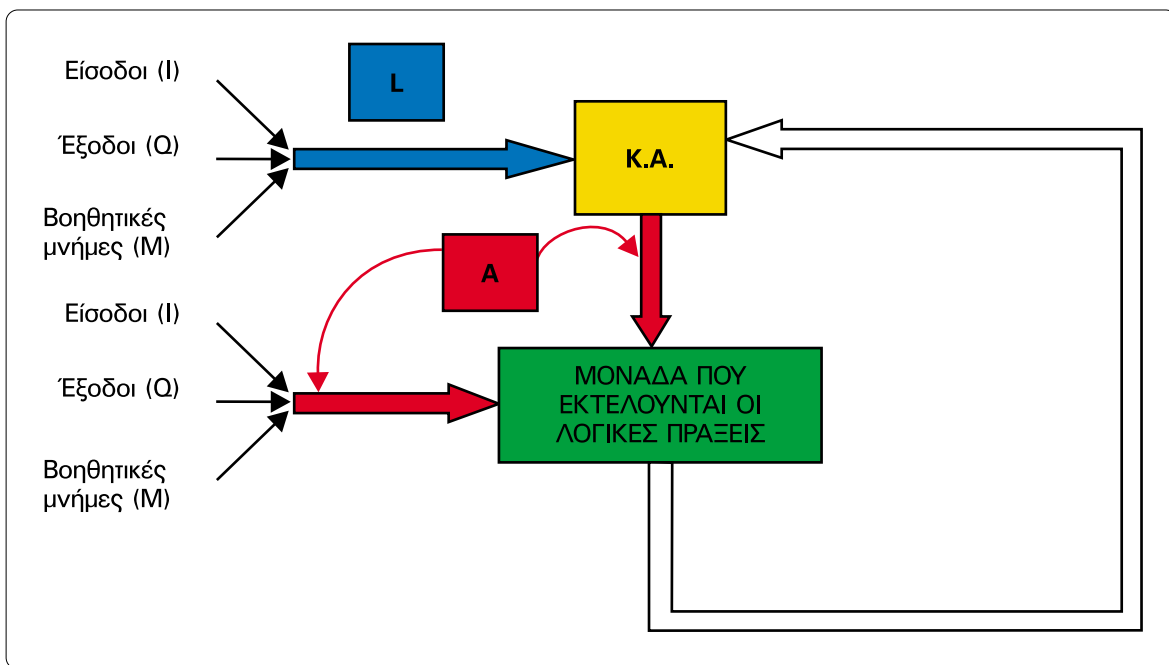
Το πρόγραμμα που αντιστοιχεί σε μια πύλη λογικού κυκλώματος (ή κλάδο ηλεκτρολογικού κυκλώματος αυτοματισμού) καταλήγει πάντα με την εντολή = (ίσον). Η εντολή αναφέρεται σε εξόδους και βοηθητικές μνήμες. Το PLC με την εντολή = μεταφέρει στις εξόδους ή στις βοηθητικές μνήμες το περιεχόμενο του Καταχωρητή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



Σχήμα 20: Σχηματική παράσταση της εντολής =.

Η εντολή A (AND)

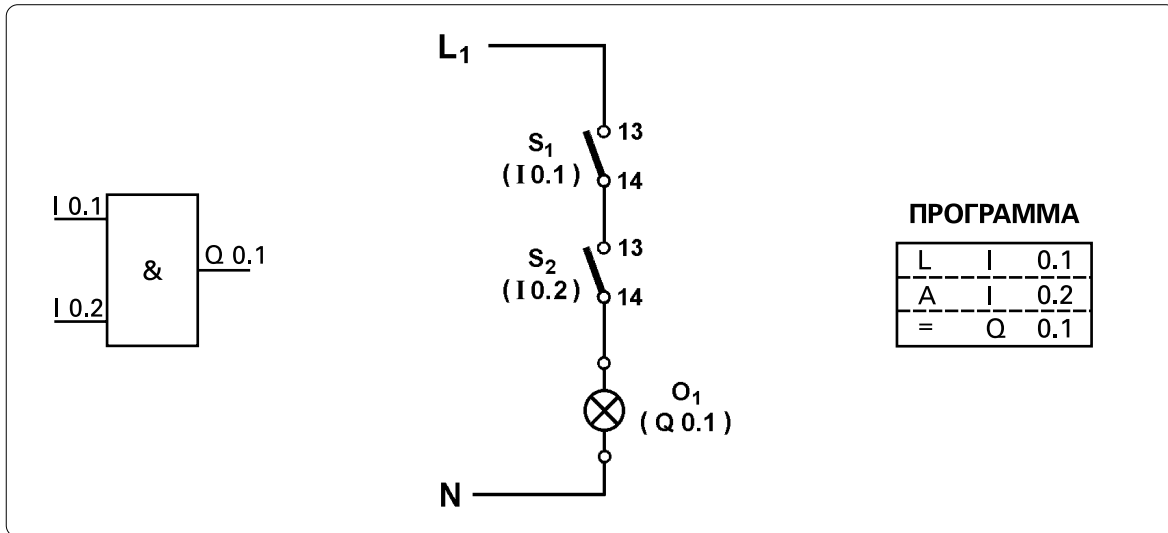
Η εντολή **A** υπαγορεύει στο PLC να εκτελέσει τη λογική πράξη AND. Η εντολή AND αναφέρεται σε εισόδους, εξόδους, βοηθητικές μνήμες, χρονικά κ.λπ. Η λογική πράξη γίνεται μεταξύ της λογικής κατάστασης της εισόδου, εξόδου, βοηθητικής μνήμης, χρονικού, κ.λπ. το οποίο αναφέρεται στην εντολή και του περιεχομένου του Κ.Α. Το αποτέλεσμα επιστρέφει στον Κ.Α. Προσέξτε το σχήμα 21, το οποίο αποδίδει πολύ καλά τον τρόπο που το PLC εκτελεί τη λογική πράξη AND.



Σχήμα 21: Σχηματική παράσταση της εντολής AND.

5.2 Αναπτύσσοντας τα πρώτα προγράμματα στη γλώσσα λίστα εντολών.

Με τις τρεις εντολές, που μάθαμε, είμαστε σε θέση να φτιάξουμε το πρώτο πρόγραμμα, που αντιστοιχεί σε μια πύλη AND (ή τον αντίστοιχο κλάδο ηλεκτρολογικού κυκλώματος αυτοματισμού). Αν παρατηρήσουμε στο σχήμα 21 τον τρόπο με τον οποίο εκτελεί την εντολή AND το PLC, εύκολα καταλαβαίνουμε πώς να γράψουμε το πρόγραμμα. Δηλαδή:



Σχήμα 22

Επεξήγηση:

Γραμμή 1: Η εντολή Load φέρνει στον Κ.Α. τη λογική κατάσταση (λογική τιμή “0” ή “1”) της εισόδου I 0.1.

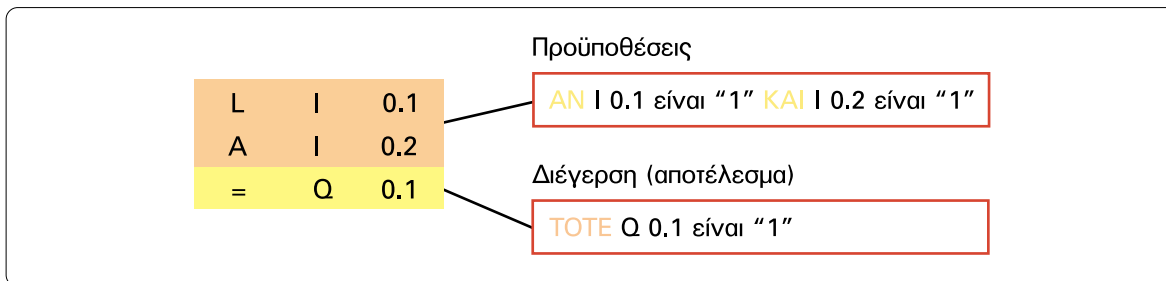
Γραμμή 2: Η εντολή AND εκτελεί τη λογική πράξη μεταξύ της λογικής κατάστασης της εισόδου I 0.2 και του περιεχομένου του Κ.Α. (δηλαδή της λογικής κατάστασης της εισόδου I 0.1), και το λογικό αποτέλεσμα επιστρέφει στον Κ.Α.

Γραμμή 3: Η εντολή = οδηγεί το περιεχόμενο του Κ.Α. (που είναι το λογικό αποτέλεσμα της πράξης AND) στην έξοδο Q 0.1.

ΠΡΟΣΟΧΗ

Το πρόγραμμα που αντιστοιχεί σε μία πύλη αποτελεί μια ομάδα εντολών.

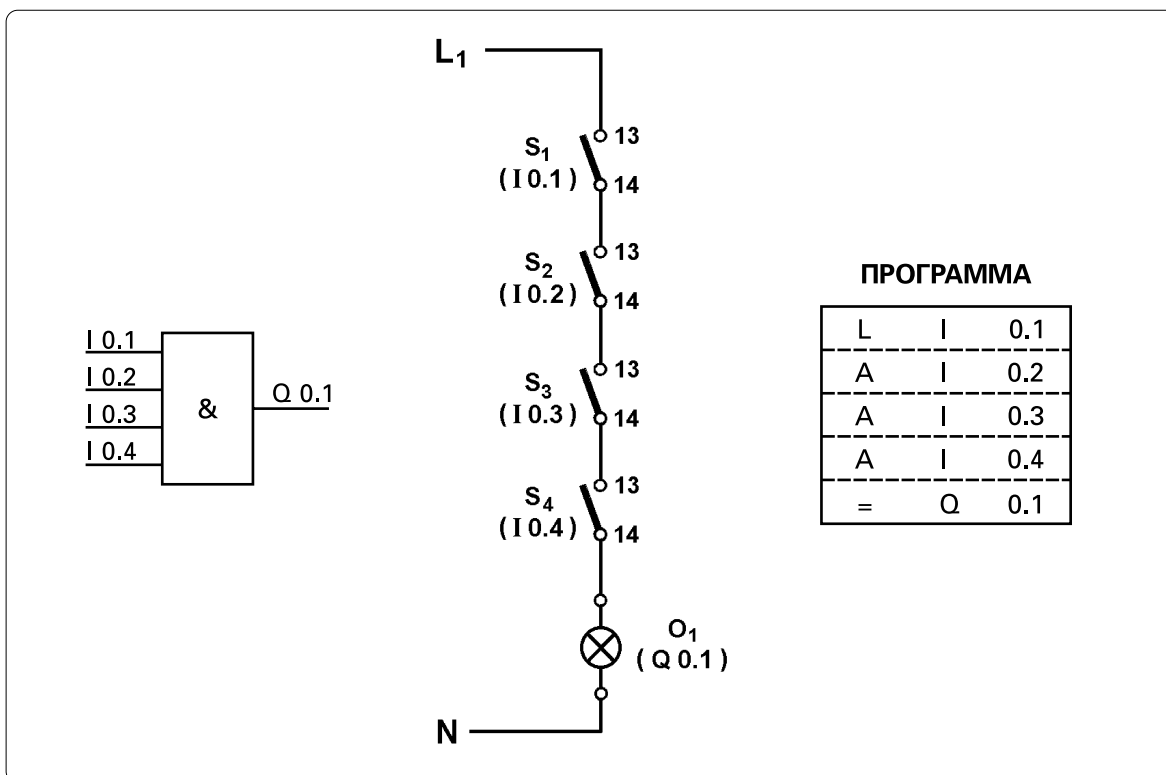
Μπορούμε να πούμε ότι στο πρόγραμμα αυτό, που αντιστοιχεί σε μια πύλη, διακρίνουμε δύο μέρη. Το πρώτο μέρος αποτελούν όλες οι εντολές πριν την εντολή =. Οι εντολές αυτές θέτουν τις προϋποθέσεις, τις ερωτήσεις. Το δεύτερο μέρος αποτελεί η εντολή =. Η εντολή αυτή δίνει το αποτέλεσμα, τη διέγερση. Το πρόγραμμα επομένως διαβάζεται ως εξής:



Σχήμα 23

Πύλη AND πολλών εισόδων.

Το πρόγραμμα για μια πύλη AND πολλών εισόδων είναι σχεδόν το ίδιο με το πρόγραμμα της πύλης δύο εισόδων.



Σχήμα 24

Επεξήγηση:

Αφού γνωρίζουμε ότι στον Κ.Α. βρίσκεται πάντα το αποτέλεσμα της προηγούμενης λογικής πράξης, εύκολα μπορούμε να καταλάβουμε τη διαδικασία.

Γραμμή 1: Η εντολή Load φέρνει στον Κ.Α. τη λογική τιμή της εισόδου I 0.1.

Γραμμή 2: Η εντολή AND εκτελεί τη λογική πράξη μεταξύ της λογικής τιμής της εισόδου I 0.2 και του περιεχομένου του Κ.Α. (δηλαδή της λογικής τιμής της εισόδου I 0.1) και το αποτέλεσμα επιστρέφει στον Κ.Α.

Γραμμή 3: Η εντολή AND εκτελεί τη λογική πράξη AND μεταξύ του προηγούμενου λογικού αποτελέσματος (που βρίσκεται στον Κ.Α.) και της λογικής τιμής της εισόδου I 0.3. Το αποτέλεσμα επιστρέφει και πάλι στον Κ.Α.

Γραμμή 4: Επαναλαμβάνεται η διαδικασία της εντολής της γραμμής 3, δηλαδή εκτελείται η λογική πράξη AND μεταξύ του νέου περιεχομένου του Κ.Α. και της λογικής τιμής της εισόδου I 0.4.

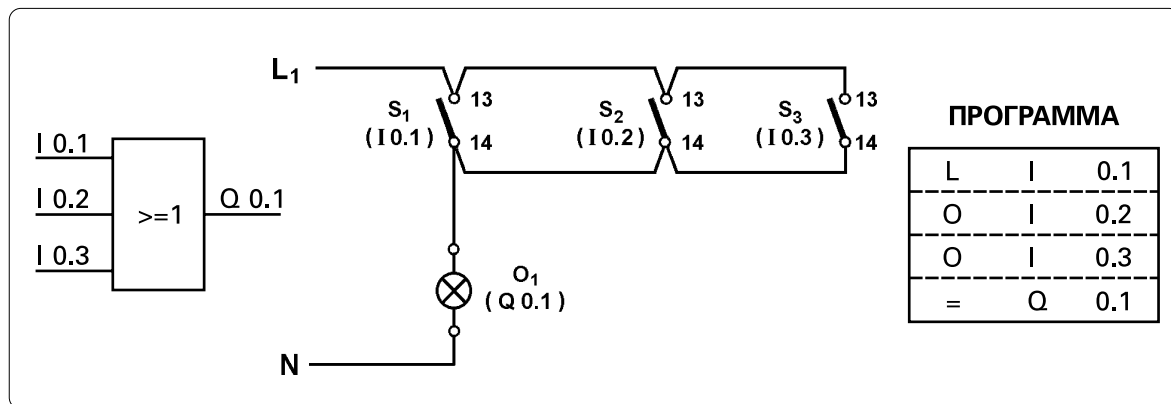
Γραμμή 5: Το περιεχόμενο του Κ.Α., δηλαδή η λογική τιμή, που είναι το τελικό λογικό αποτέλεσμα της προηγούμενης ομάδας εντολών, μεταφέρεται στην έξοδο Q 0.1.

5.3 Παρουσίαση λοιπών εντολών στη γλώσσα λίστα εντολών.

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 προγράμματος:

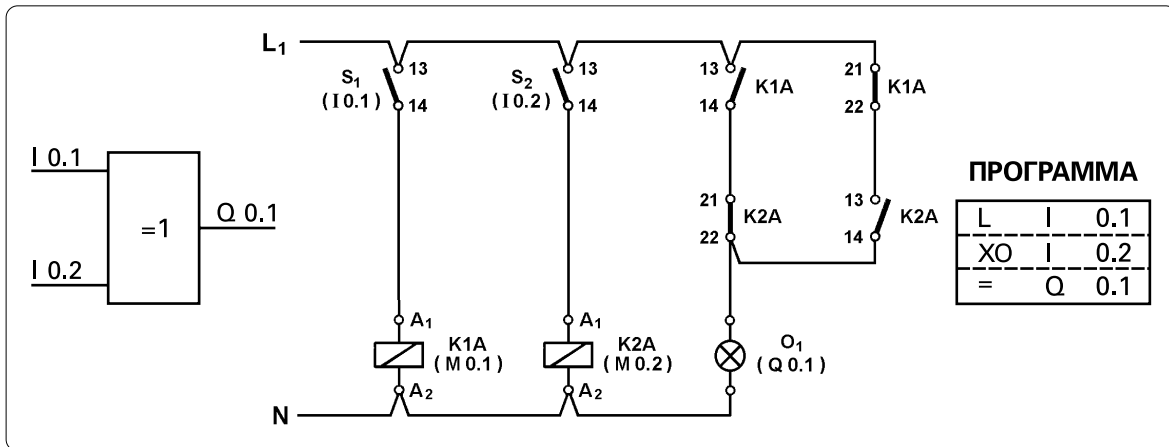


Σχήμα 25

Η εντολή XO (EXOR)

Η εντολή EXOR υπαγορεύει στο PLC να εκτελέσει τη λογική πράξη EXOR. Η εντολή EXOR αναφέρεται σε εισόδους, εξόδους, βοηθητικές μνήμες, χρονικά, κ.λπ. Εκτελείται με ανάλογο τρόπο, με αυτόν που εκτελούνται οι εντολές AND και OR. Υπενθυμίζουμε ότι η λογική πύλη EXOR αποτελεί σύνθετο λογικό κύκλωμα και όχι βασική πύλη.

Παράδειγμα προγράμματος:



Σχήμα 26

Αντιστοιχία συμβολισμών των ηλεκτρολογικών κυκλωμάτων αυτοματισμού με τους συμβολισμούς των PLC.

Όπως ήδη έχουμε δει, στα PLC διακρίνουμε τις εισόδους, στις οποίες δίνουμε τις ονομασίες I 0.0, I 0.1, I 0.2, I 0.3, ..., τις βοηθητικές μνήμες, τις οποίες ονομάζουμε M 0.0, M 0.1, M 0.2, M 0.3, ..., και τις εξόδους, τις οποίες ονομάζουμε Q 0.0, Q 0.1, Q 0.2, Q 0.3, Είναι επίσης γνωστοί οι συμβολισμοί που χρησιμοποιούμε στα ηλεκτρολογικά κυκλώματα αυτοματισμού (S₁, S₂,... για διακόπτες, μπουτόνς και αισθητήρες, K1M, K2M,... για ηλεκτρονόμους ισχύος, K1A, K2A,... για βοηθητικούς ηλεκτρονόμους κ.λπ.). Στα παραδείγματα που θα ακολουθήσουν, όταν αναφερόμαστε σε ηλεκτρολογικά κυκλώματα αυτοματισμού θα υπάρχουν στα σχέδια των κυκλωμάτων και οι ονομασίες των αντίστοιχων στοιχείων του PLC.

Υπενθυμίζουμε ότι οι διακόπτες, τα μπουτόνς και οι αισθητήρες είναι είσοδοι σε ένα PLC, οι ηλεκτρονόμοι ισχύος είναι έξοδοι του PLC και οι βοηθητικοί ηλεκτρονόμοι είναι βοηθητικές μνήμες του PLC.

Η εντολή N (NOT)

Δεν πρόκειται για ανεξάρτητη λογική εντολή όπως οι τρεις εντολές λογικών πράξεων AND, OR, EXOR. Η εντολή N είναι συμπλήρωμα όλων των εντολών που είδαμε προηγουμένως. Μπορούμε επομένως να πούμε ότι έχουμε όχι μια εντολή NOT, αλλά πέντε εντολές:

- LN → LOAD NOT
- AN → AND NOT
- ON → OR NOT
- XON → XOR NOT
- = N → = NOT

Η εντολή NOT χρησιμοποιείται ως εξής:

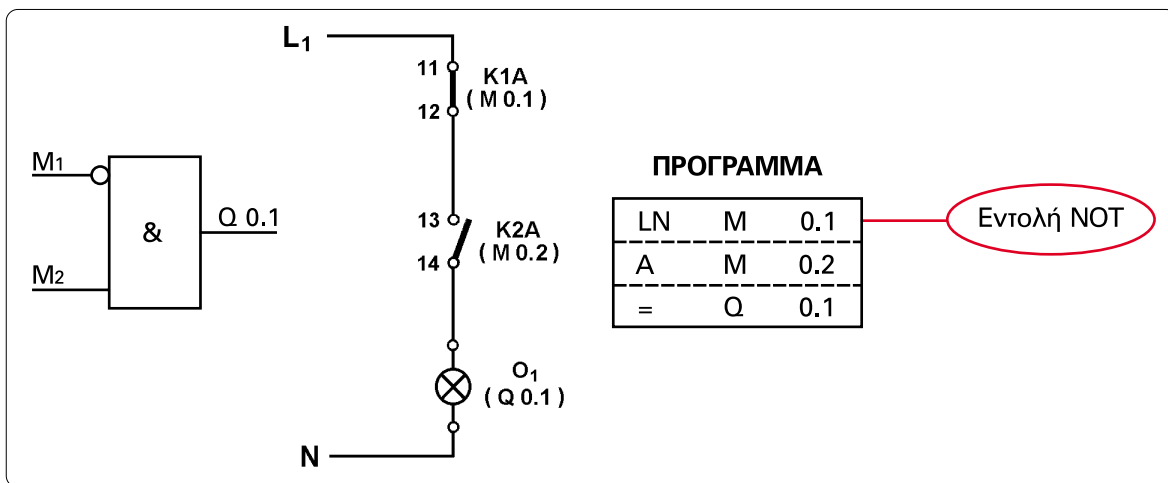
- Όταν έχουμε το σχέδιο λογικού κυκλώματος του αυτοματισμού.

Όταν στο λογικό κύκλωμα έχουμε πύλη NOT, στην εντολή που θα είχαμε αν δεν υπήρχε η πύλη NOT (A, O, XO, L, =) προσθέτουμε το N.

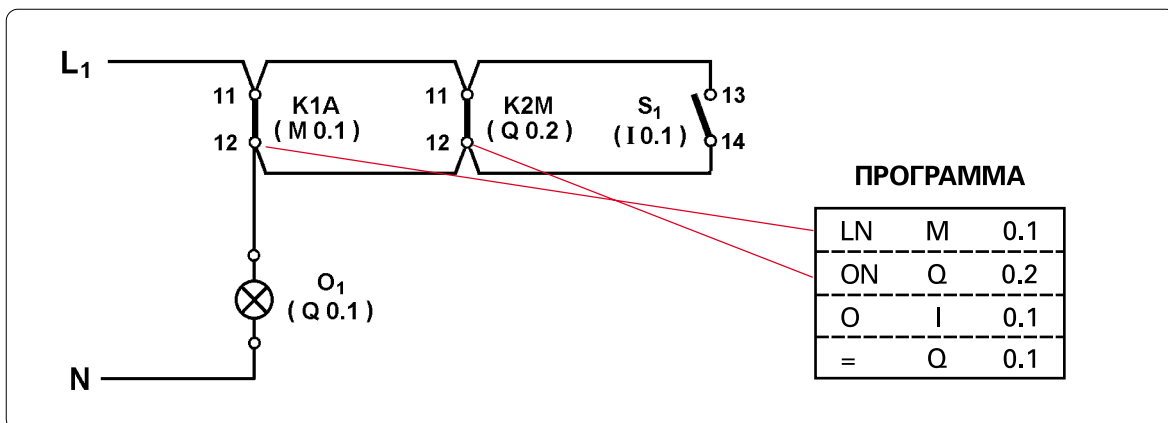
- Όταν έχουμε το ηλεκτρολογικό σχέδιο του αυτοματισμού.

Όταν στο ηλεκτρολογικό σχέδιο έχουμε “κανονική κλειστή” επαφή ηλεκτρονόμου, στην εντολή που θα είχαμε αν η επαφή του ηλεκτρονόμου ήταν “κανονικά ανοικτή” (A, O, L) προσθέτουμε το N. **ΠΡΟΣΟΧΗ! Αυτό δεν ισχύει όταν έχουμε “κανονική κλειστή” επαφή εξωτερικού διακόπτη, μπουτόν ή αισθητήρα.**

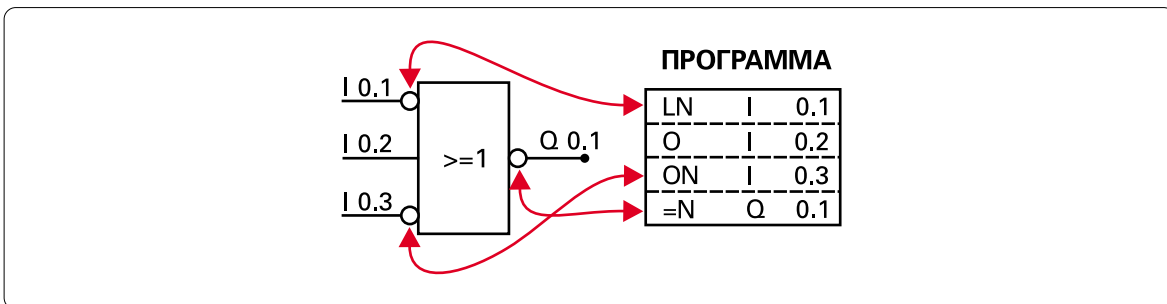
Παραδείγματα με την εντολή NOT:



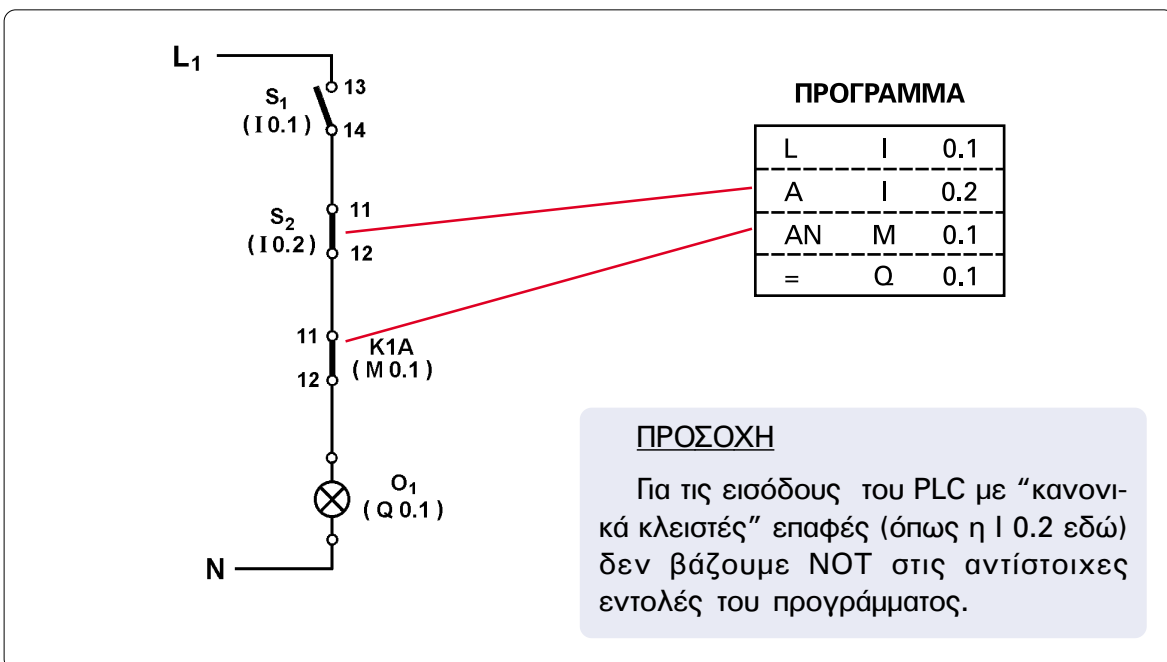
Σχήμα 27



Σχήμα 28



Σχήμα 29



Σχήμα 30

5.4 Παραδείγματα ανάπτυξης προγράμματος σε γλώσσα λίστα εντολών.

Αφού είδαμε τις πρώτες βασικές εντολές του PLC και τα πρώτα στοιχειώδη προγράμματα που αντιστοιχούν σε μια πύλη (ή αντίστοιχα σε έναν κλάδο ηλεκτρολογικού κυκλώματος αυτοματισμού), είμαστε τώρα έτοιμοι να φτιάξουμε τα πρώτα προγράμματα που αντιστοιχούν σε ένα πλήρες λογικό κύκλωμα με πολλές λογικές πύλες ή αντίστοιχα σε ένα πλήρες σχέδιο ηλεκτρολογικού αυτοματισμού με πολλούς κλάδους.

Βασικοί κανόνες για την ανάπτυξη του προγράμματος.

- Όταν έχουμε το σχέδιο λογικού κυκλώματος του αυτοματισμού:

1. Αντιστοιχούμε τις εξόδους των ενδιάμεσων πυλών (δηλαδή τις εξόδους πυλών που

είναι είσοδοι σε άλλες πύλες) με βοηθητικές μνήμες M 0.0, M 0.1, M 0.2, M 0.3, κ.λπ.

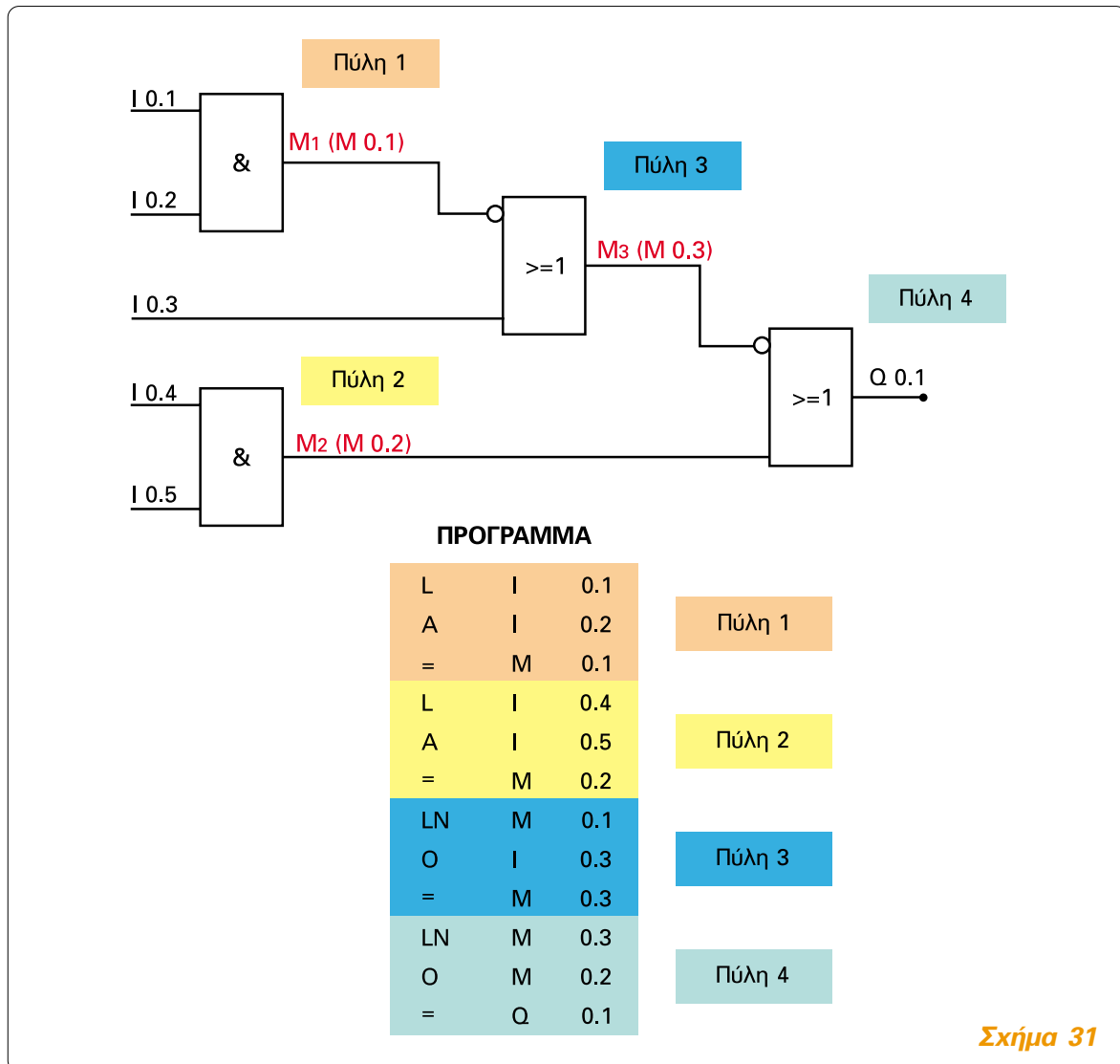
2. Για κάθε πύλη γράφουμε την ομάδα εντολών όπως ακριβώς είδαμε προηγουμένως στις ενότητες 5.2 και 5.3.

• Όταν έχουμε ηλεκτρολογικό σχέδιο του αυτοματισμού:

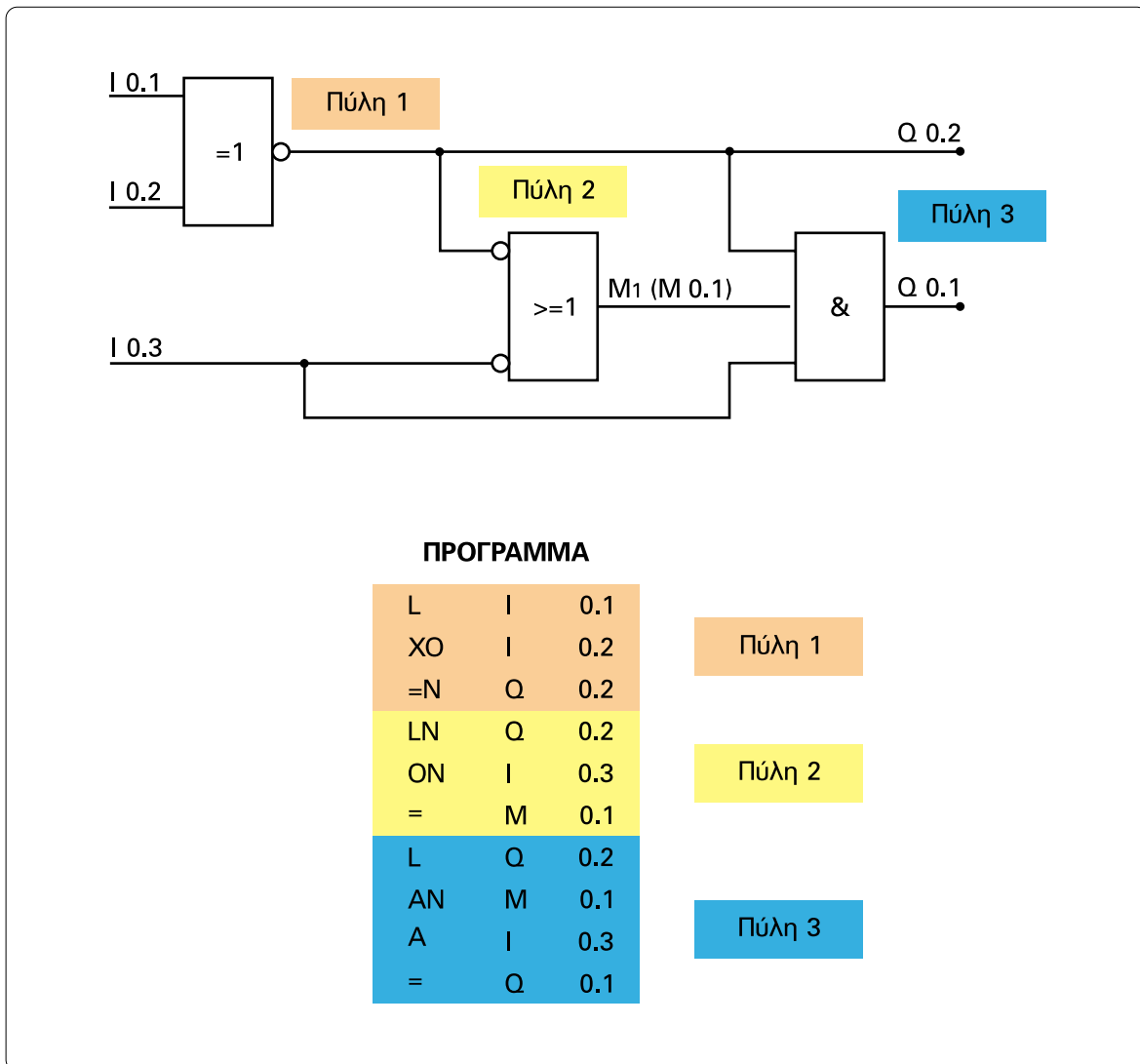
1. Αντιστοιχούμε τα πηνία των βοηθητικών ηλεκτρονόμων με βοηθητικές μνήμες M 0.0, M 0.1, M 0.2, M 0.3, κ.λπ.

2. Για κάθε “αυτοτελές” τμήμα του κυκλώματος γράφουμε την ομάδα εντολών, όπως ακριβώς είδαμε προηγουμένως. Υπενθυμίζουμε ότι ένα “αυτοτελές” τμήμα κυκλώματος χαρακτηρίζεται από το πηνίο του ηλεκτρονόμου ή από την κατανάλωση στην οποία καταλήγει (βλέπε Συστήματα Αυτοματισμών - Α΄ τόμος, ενότητα 2.8, σελίδες 70-71).

Παράδειγμα 1. Να γραφεί το πρόγραμμα που αντιστοιχεί στο παρακάτω λογικό κύκλωμα.

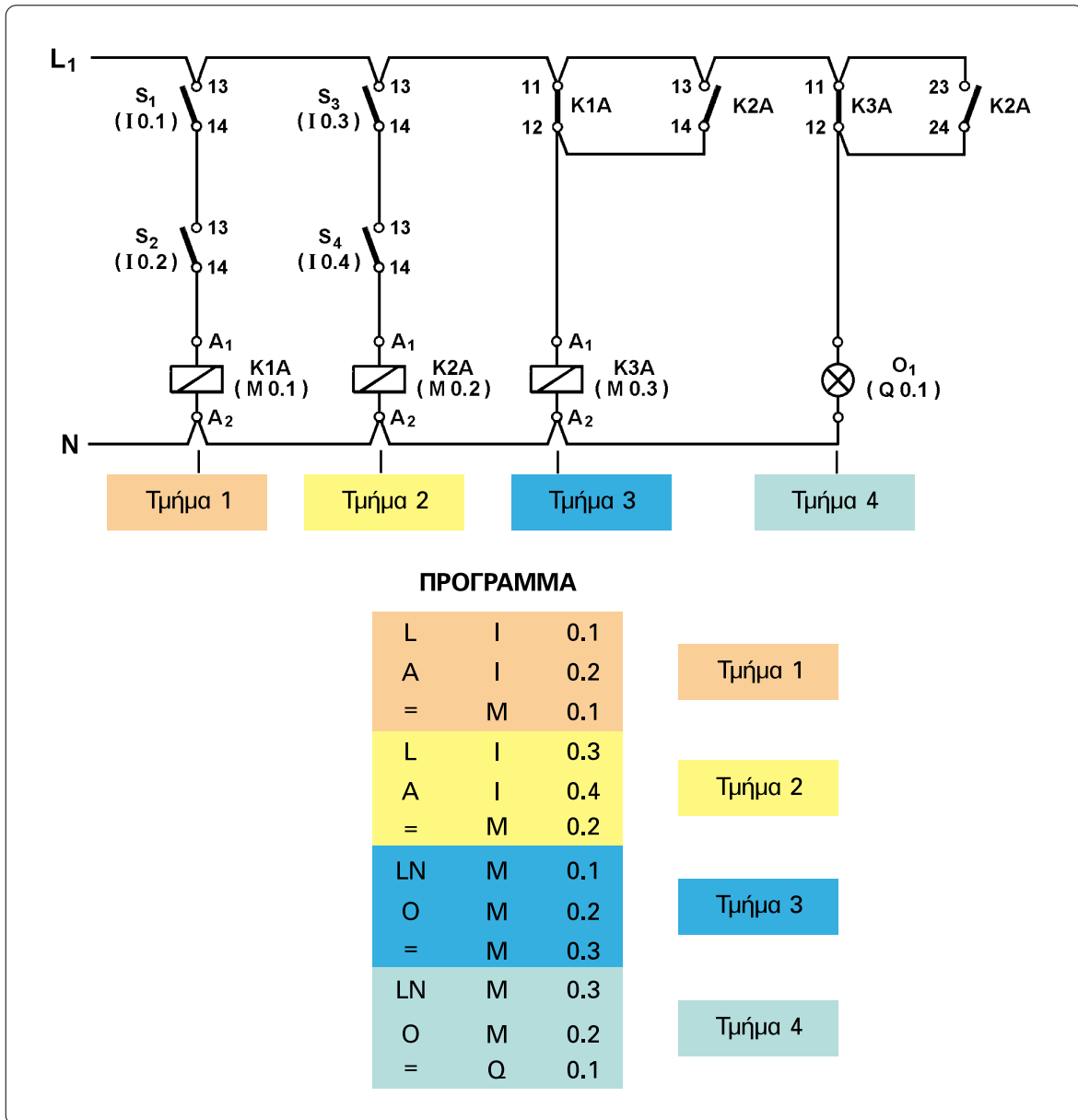


Παράδειγμα 2. Να γραφεί το πρόγραμμα που αντιστοιχεί στο παρακάτω λογικό κύκλωμα.



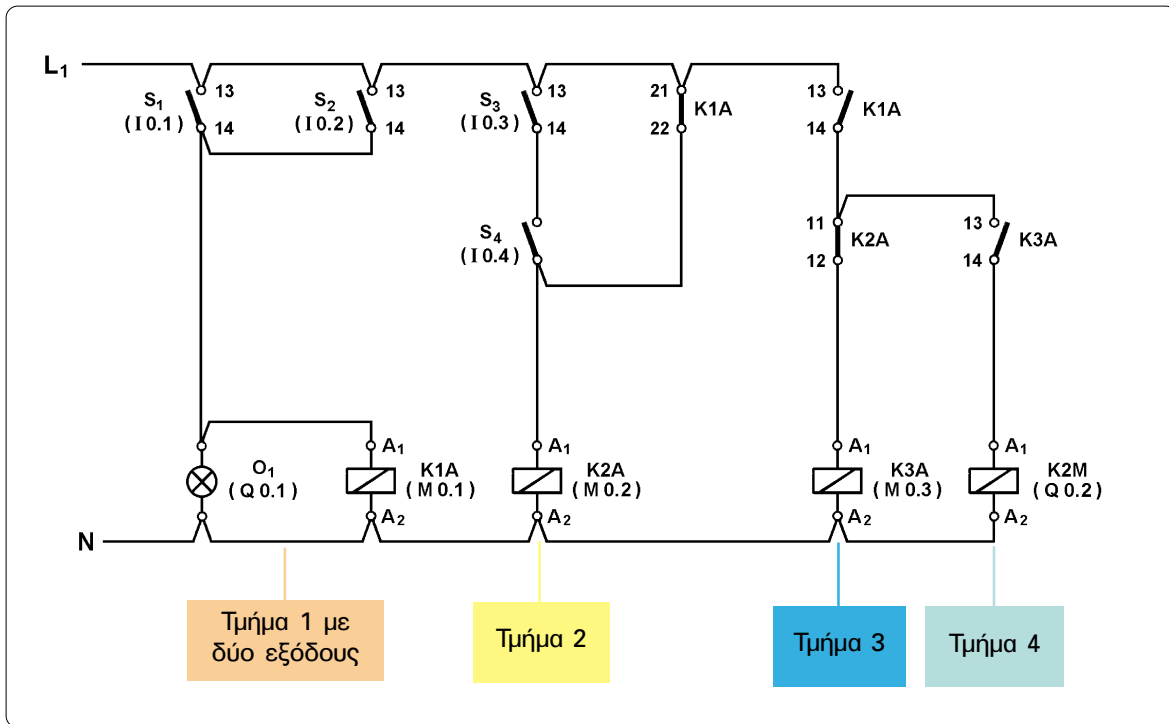
Σχήμα 32

Παράδειγμα 3. Να γραφεί το πρόγραμμα που αντιστοιχεί στο παρακάτω ηλεκτρολογικό σχέδιο αυτοματισμού.



Σχήμα 33

Παράδειγμα 4. Να γραφεί το πρόγραμμα σε γλώσσα λίστα εντολών, που αντιστοιχεί στο ηλεκτρικό κύκλωμα αυτοματισμού του σχήματος 34.



Σχήμα 34

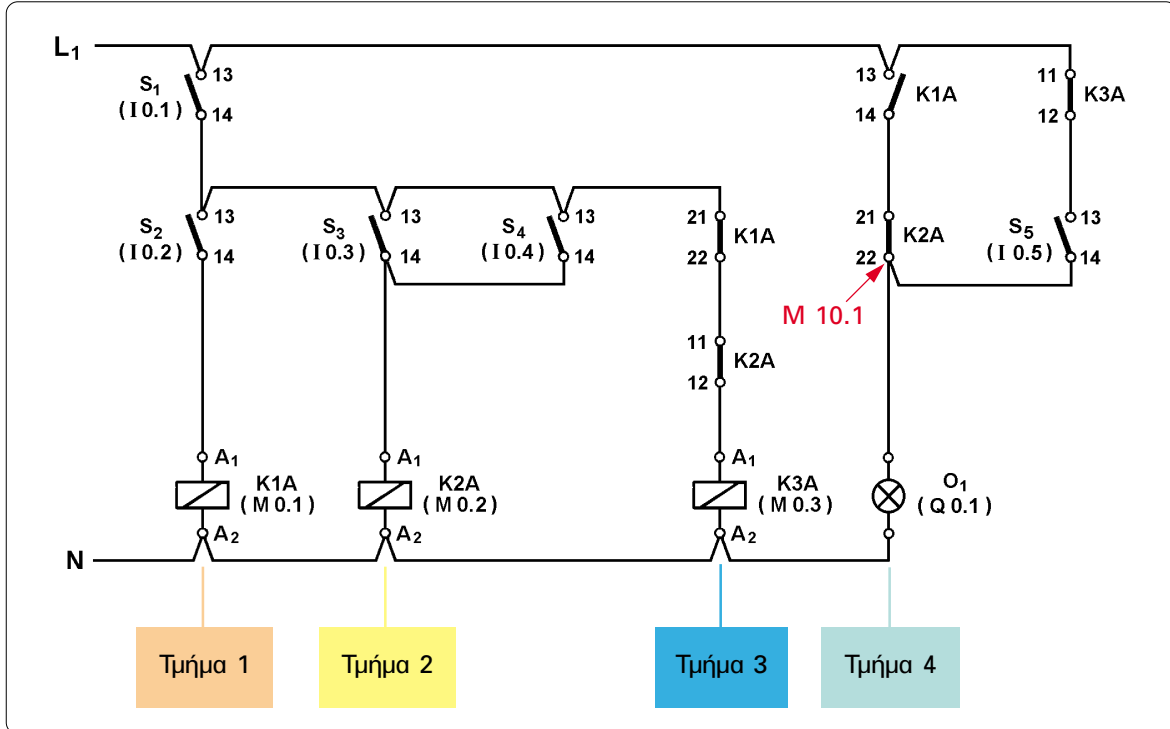
Για να προχωρήσουμε στην ανάπτυξη του προγράμματος, πρέπει να θυμηθούμε όσα έχουμε πει για τη μετατροπή των ηλεκτρικών κυκλωμάτων αυτοματισμού σε λογικά κυκλώματα. Και συγκεκριμένα πως καθορίζουμε τα “αυτοτελή” τμήματα ενός ηλεκτρικού κυκλώματος και πως μετατρέπουμε πολύπλοκες συνδεσμολογίες επαφών (βλέπε Συστήματα Αυτοματισμών - Α΄ τόμος, ενότητα 2.8, σελίδες 68-76). Αν προσέξουμε αυτά τα δύο σημεία, η ανάπτυξη του προγράμματος γίνεται εύκολα γράφοντας μια ομάδα εντολών για κάθε “αυτοτελές” τμήμα του ηλεκτρικού κυκλώματος αυτοματισμού.

ΠΡΟΓΡΑΜΜΑ

L	I	0.1	
O	I	0.2	← Τμήμα 1. ΠΡΟΣΟΧΗ! Εδώ έχουμε δύο εξόδους σε ένα κλάδο.
=	Q	0.1	
=	M	0.1	
L	I	0.3	
A	I	0.4	← Τμήμα 2. ΠΡΟΣΟΧΗ! Στις περιπτώσεις που έχουμε πολύπλοκες συνδεσμολογίες (όπως εδώ που έχουμε δύο επαφές σε σειρά και το σύνολο αυτών παράλληλα σε μια τρίτη επαφή) θα ξεκινάμε το πρόγραμμα από τις επιμέρους συνδεσμολογίες. Δοκιμάστε να ξεκινήσετε το πρόγραμμα από την επαφή 21-22 του K1A. Τι παρατηρείτε;
ON	M	0.1	
=	M	0.2	
L	M	0.1	
AN	M	0.2	← Τμήμα 3.
=	M	0.3	
L	M	0.1	
A	M	0.3	← Τμήμα 4.
=	Q	0.2	

Σχήμα 35

Παράδειγμα 5. Να γραφεί το πρόγραμμα σε γλώσσα λίστα εντολών, που αντιστοιχεί στο ηλεκτρικό κύκλωμα αυτοματισμού του σχήματος 36.



Σχήμα 36

Το καινούργιο στο παράδειγμα αυτό είναι η πολύπλοκη συνδεσμολογία του κλάδου 4. Αν δοκιμάσουμε να γράψουμε το πρόγραμμα για τον κλάδο 4 με τον ίδιο τρόπο που δουλέψαμε μέχρι τώρα, θα δούμε ότι κάπου θα δυσκολευτούμε.

Ας παρακολουθήσουμε μαζί το παράδειγμα για να δούμε το λόγο. Έστω ότι ξεκινάμε το πρόγραμμα από τη συνδεσμολογία σειράς των επαφών K1A (M 0.1) και K2A (M 0.2), οπότε έχουμε:

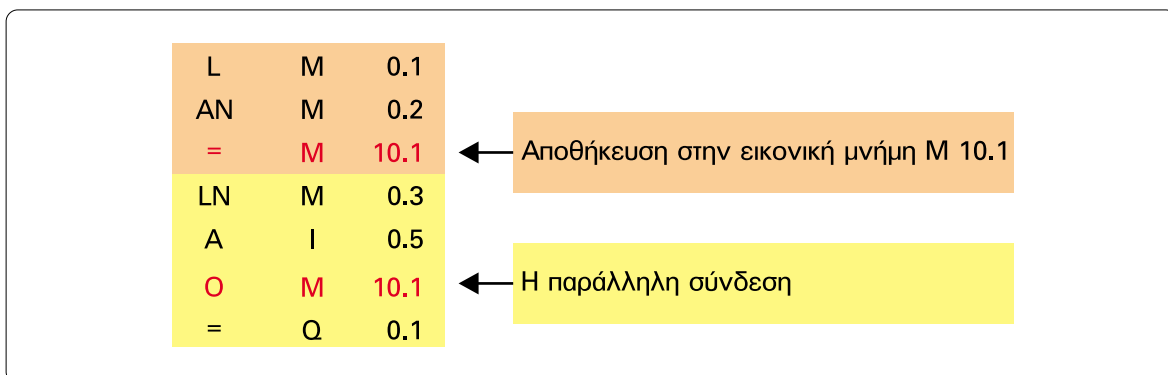
L M 0.1
AN M 0.2

Το αποτέλεσμα που βρίσκεται στον Κ.Α. μετά την εκτέλεση των παραπάνω εντολών δεν οδηγείται πουθενά, αφού δεν ακολουθεί εντολή =, ούτε μπορεί να χρησιμοποιηθεί στην επόμενη πράξη, αφού θα πρέπει να ακολουθήσει η λογική πράξη της άλλης επιμέρους συνδεσμολογίας σειράς των επαφών K3A (M 0.3) και S₅ (I 0.5), δηλαδή:

LN M 0.3
A I 0.5

Αν γράψουμε το πρόγραμμα με αυτή τη σειρά, το αποτέλεσμα της πρώτης λογικής πράξης χάνεται από τον Κ.Α., μόλις εκτελεστεί η εντολή LN M 0.3.

Η λύση στην περίπτωση αυτή μπορεί να δοθεί ως εξής: Χρησιμοποιούμε μια ενδιάμεση μνήμη M 10.1 για την αποθήκευση του αποτελέσματος της πρώτης λογικής πράξης, το οποίο καλούμε στην συνέχεια μετά την ολοκλήρωση της δεύτερης λογικής πράξης. Η ενδιάμεση αυτή μνήμη είναι “εικονική”, αφού δεν ανταποκρίνεται σε βοηθητικό ηλεκτρονόμο του κυκλώματος. Χρειάζεται προσοχή όμως να μην χρησιμοποιήσουμε αριθμό μνήμης, που χρησιμοποιείται σε άλλο σημείο του προγράμματος. Έτσι το πρόγραμμα του κλάδου 4 θα γίνει:



Σχήμα 37

Το συνολικό πρόγραμμα του κυκλώματος δίνεται στη συνέχεια.

ΠΡΟΓΡΑΜΜΑ			
L	I	0.1	← Τμήμα 1.
A	I	0.2	
=	M	0.1	
L	I	0.3	← Τμήμα 2. ΠΡΟΣΟΧΗ! Ξεκινάμε από τις δύο παράλληλες επαφές
O	I	0.4	
A	I	0.1	
=	M	0.2	
L	I	0.1	← Τμήμα 3.
AN	M	0.1	
AN	M	0.2	
=	M	0.3	
L	M	0.1	← Τμήμα 4.
AN	M	0.2	
=	M	10.1	
LN	M	0.3	
A	I	0.5	
O	M	10.1	
=	Q	0.1	

Σχήμα 38

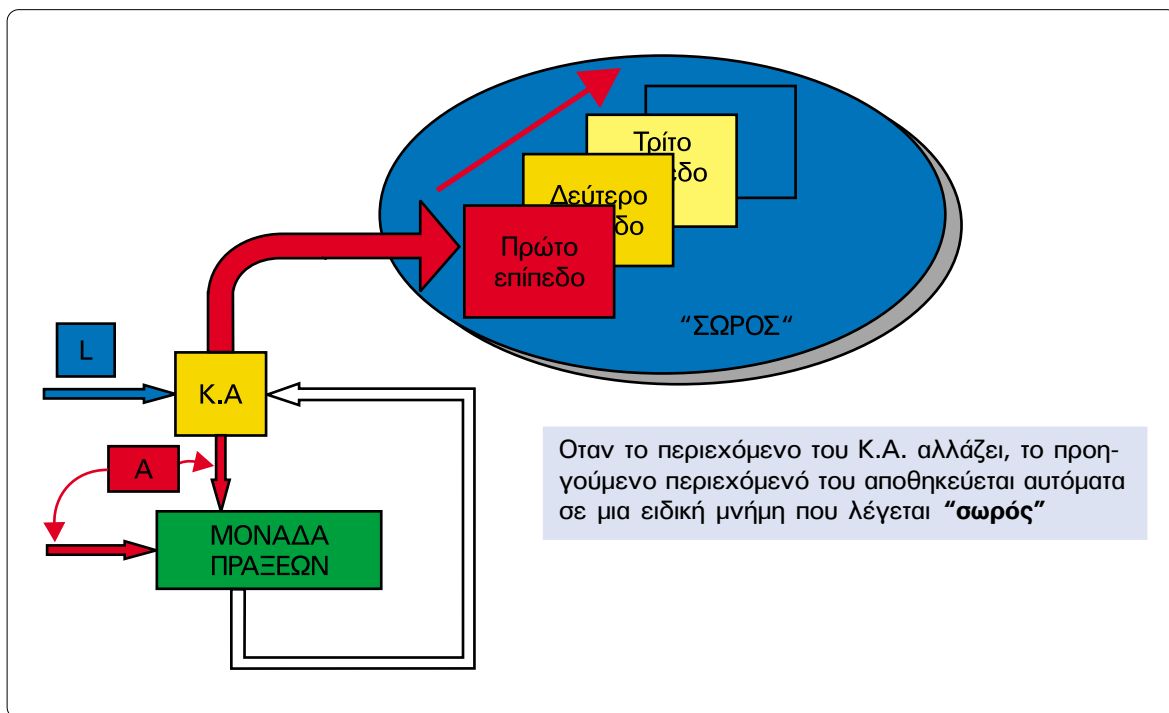
5.5 Πολύπλοκες συνδεσμολογίες με την χρήση του “σωρού”.

Στα περισσότερα PLC υπάρχει τρόπος αντιμετώπισης του παραπάνω προβλήματος της πολύπλοκης συνδεσμολογίας του παραδείγματος 5, χωρίς να είμαστε αναγκασμένοι να χρησιμοποιήσουμε ενδιάμεση “εικονική” μνήμη. Αυτό γίνεται με τον παρακάτω τρόπο:

Όταν το περιεχόμενο του Κ.Α. αλλάζει, το προηγούμενο περιεχόμενό του αποθηκεύεται αυτόματα σε μια περιοχή της μνήμης, που ονομάζεται “σωρός”. Ο “σωρός” μπορεί να αποτελείται από μία ή και περισσότερες θέσεις μνήμης. Στην περίπτωση που αποτελείται από περισσότερες θέσεις, αυτές διακρίνονται σε επίπεδα. Το πρώτο δεδομένο που θα έρθει από τον Κ.Α., θα αποθηκευτεί στο πρώτο επίπεδο. Το δεύτερο δεδομένο που θα έρθει, θα αποθηκευτεί και πάλι στο πρώτο επίπεδο, ωθώντας το προηγούμενο δεδομένο στο δεύτερο επίπεδο κ.ο.κ. Με αυτόν τον τρόπο το PLC φυλάσσει τα τελευταία περιεχόμενα του Κ.Α. (τόσα, όσες είναι και οι θέσεις του “σωρού”). Μάλιστα στα πιο κάτω επίπεδα βρίσκονται τα πιο πρόσφατα περιεχόμενα του Κ.Α.

Βλέπουμε επομένως ότι στις πολύπλοκες συνδεσμολογίες δεν είναι ανάγκη να χρησιμοποιήσουμε “εικονική” μνήμη για να αποθηκεύσουμε το ενδιάμεσο αποτέλεσμα, αφού

το κάνει μόνο του το PLC με την χρήση του “σωρού”. Το θέμα βέβαια είναι να δούμε πως θα χρησιμοποιήσουμε τα περιεχόμενα του “σωρού” στο πρόγραμμά μας. Τα διάφορα PLC της αγοράς κάνουν χρήση του “σωρού” με τον τρόπο που αναφέραμε. Εκεί που διαφέρουν τα PLC μεταξύ τους είναι στις εντολές με τις οποίες ανακαλούν τα περιεχόμενα του “σωρού”. Εμείς θα παρουσιάσουμε τις περιπτώσεις δύο PLC, τις οποίες θεωρούμε αντιπροσωπευτικές στο θέμα αυτό.



Σχήμα 39: Σχηματική παράσταση της λειτουργίας του “σωρού”.

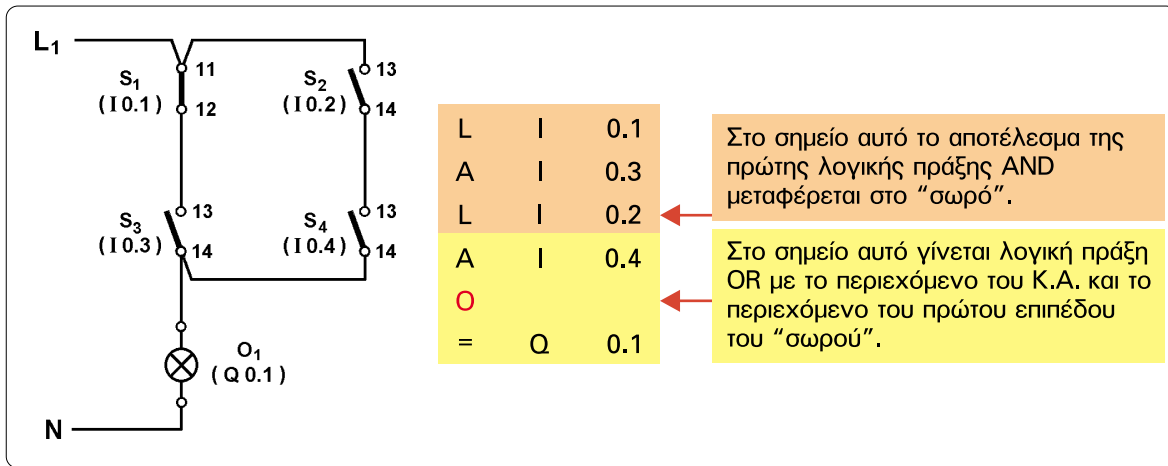
Η περίπτωση του PLC **SUCOS PS 3** της **KLOCKNER MOELLER**.

Ο τρόπος με τον οποίο γίνεται η χρήση των περιεχομένων του “σωρού” στο συγκεκριμένο PLC είναι ο παρακάτω:

Σε μια εντολή **O** ή **A** χωρίς παράμετρο, το PLC αναλαμβάνει να κάνει την αντίστοιχη λογική πράξη μεταξύ του περιεχομένου του Κ.Α. και του περιεχομένου του πρώτου πάντα επιπέδου του “σωρού”. Μόλις όμως εκτελεστεί μια λογική πράξη με το περιεχόμενο του πρώτου επιπέδου του σωρού, τα περιεχόμενα του “σωρού” ολισθαίνουν προς τα κάτω κατά ένα επίπεδο, δηλαδή το περιεχόμενο του δεύτερου επιπέδου “κατεβαίνει” στο πρώτο, το περιεχόμενο του τρίτου επιπέδου “κατεβαίνει” στο δεύτερο κ.ο.κ.

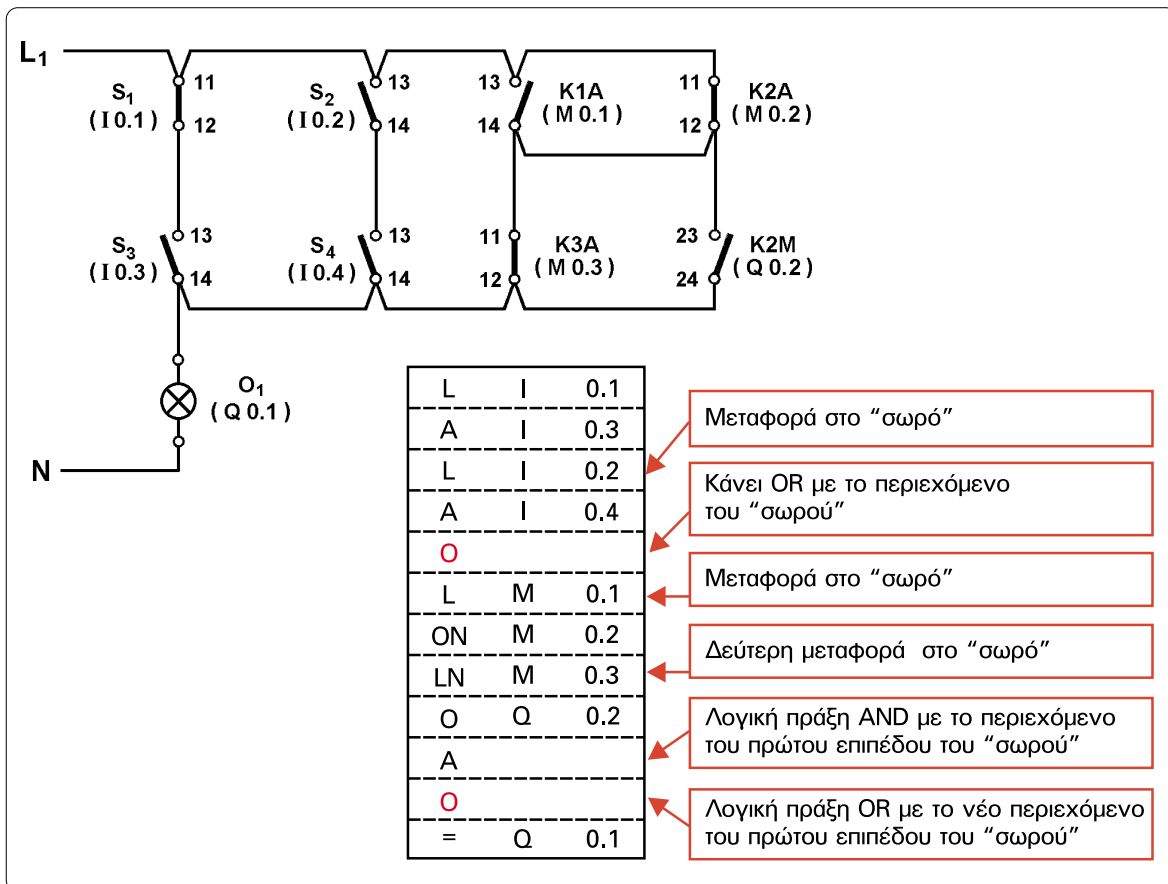
Θα δούμε πως χρησιμοποιείται ο “σωρός” στο συγκεκριμένο PLC μέσω των παρακάτω παραδειγμάτων:

Παράδειγμα 1.



Σχήμα 40

Παράδειγμα 2.



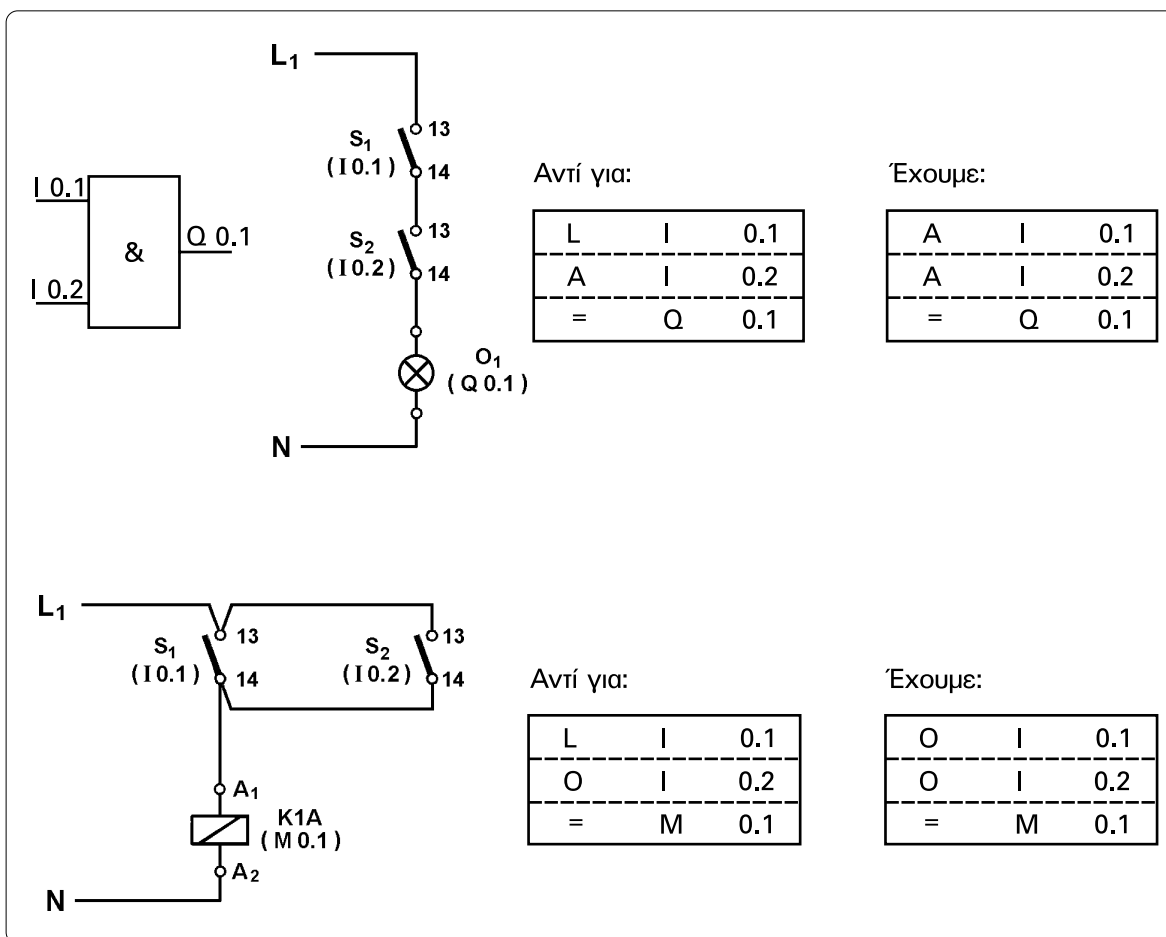
Σχήμα 41

Η περίπτωση των PLC SIMATIC S7-300 και S7-400 της SIEMENS.

Η σειρά των PLC SIMATIC-S7 της SIEMENS παρέχει πολλές διευκολύνσεις στο προγραμματισμό, όσον αφορά το θέμα της αντιμετώπισης των πολύπλοκων συνδεσμολογιών.

Πρέπει εδώ να τονίσουμε μια μικρή διαφοροποίηση που υπάρχει στα PLC SIMATIC S7-300 και S7-400 και αφορά όσα έχουμε αναφέρει μέχρι τώρα για τα διάφορα PLC:

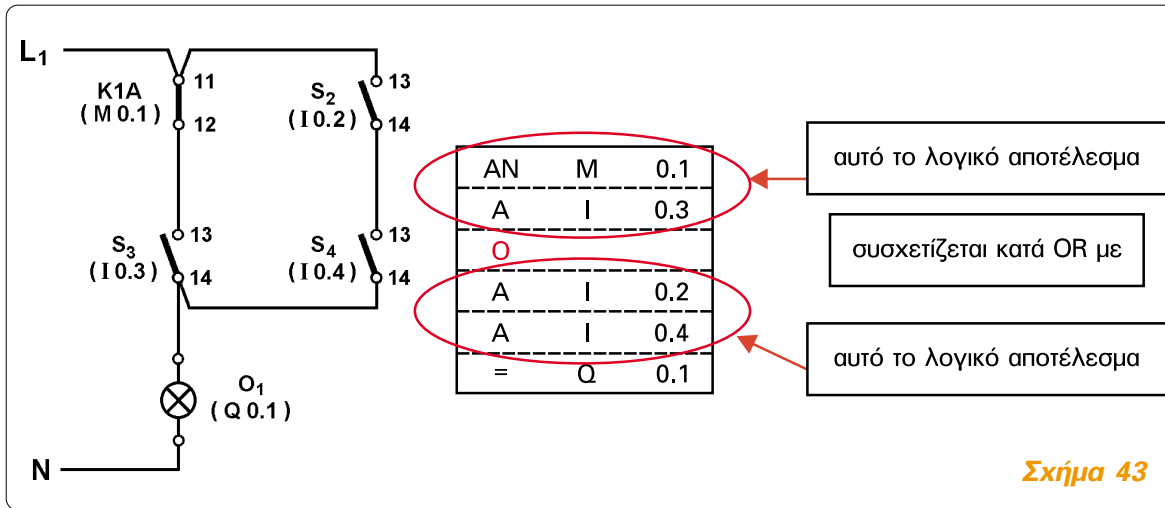
Στον προγραμματισμό των παραπάνω PLC δεν χρησιμοποιείται η εντολή L (Load), αλλά στη θέση της χρησιμοποιείται η εντολή **A** ή **O** ανάλογα με το είδος της συνδεσμολογίας που προγραμματίζεται. Δηλαδή:



Σχήμα 42

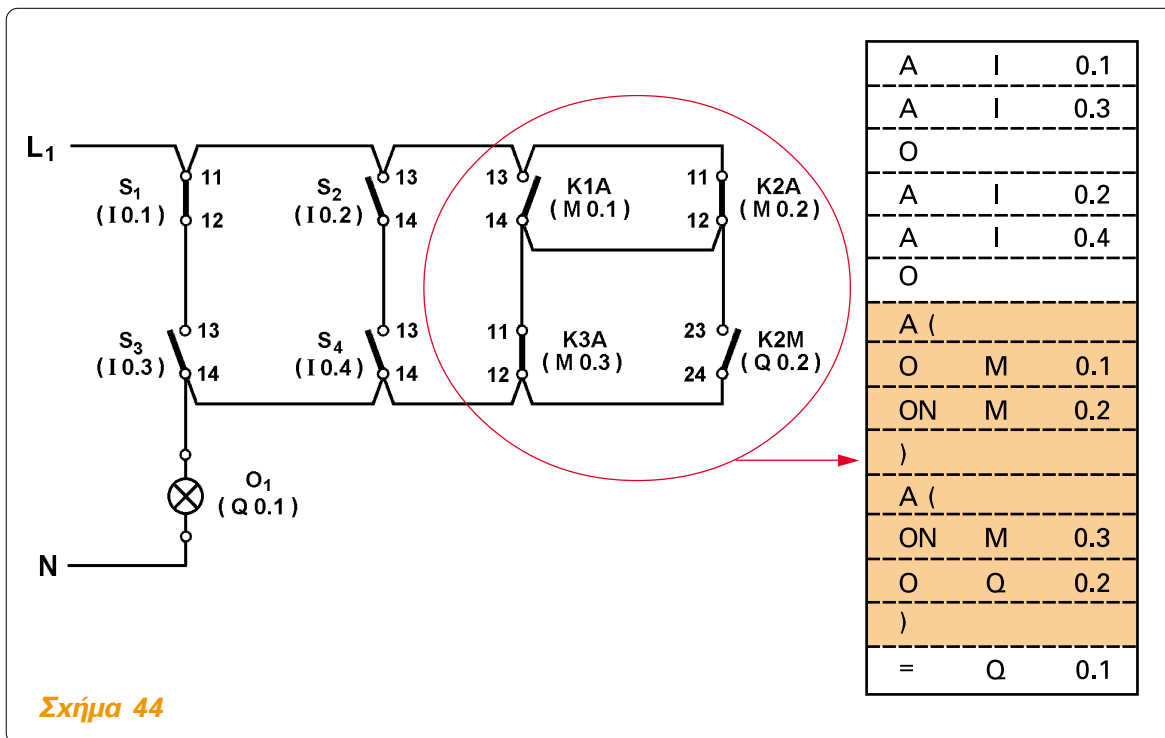
Η χρήση του “σωρού” γίνεται με πολύ απλό τρόπο:

Περίπτωση 1η: Απλές συνδεσμολογίες



Περίπτωση 2η: Χρήση παρενθέσεων

Μια πολύ μεγάλη διευκόλυνση στην αντιμετώπιση των πολύπλοκων συνδεσμολογιών είναι η χρήση παρενθέσεων. Μέσα στην παρένθεση περικλείουμε τις εσωτερικές συνδεσμολογίες. Ας δούμε ένα παράδειγμα.



6. Ανάπτυξη προγράμματος σε γλώσσα LADDER (LAD).

6.1 Γενικά.

Η γλώσσα Ladder (LAD) είναι γλώσσα που χρησιμοποιεί τα ηλεκτρολογικά γραφικά. Το πρόγραμμα σε γλώσσα LADDER μοιάζει με το ηλεκτρολογικό σχέδιο του αυτοματισμού. Οι ιδιαιτερότητες που έχουμε να αντιμετωπίσουμε στη γλώσσα αυτή είναι:

- Χρησιμοποιούνται σύμβολα από την Αμερικάνικη τυποποίηση και όχι από την Ευρωπαϊκή με την οποία είμαστε εξοικειωμένοι.
- Το “σχέδιο-πρόγραμμα” είναι τυποποιημένο, δεν έχουμε δηλαδή την ελευθερία που έχουμε κατά τη σχεδίαση. Για παράδειγμα σε κάθε κλάδο μπορούμε να έχουμε περιορισμένο αριθμό στοιχείων προγράμματος (διακόπτες και επαφές). Επίσης, δεν μπορούμε να κάνουμε οποιασδήποτε μορφής διακλάδωση.

Άρα η δουλειά που έχει να κάνει ο προγραμματιστής στη γλώσσα LADDER είναι να προσαρμόσει το σχέδιο του αυτοματισμού, στα δεδομένα που απαιτεί η γλώσσα.

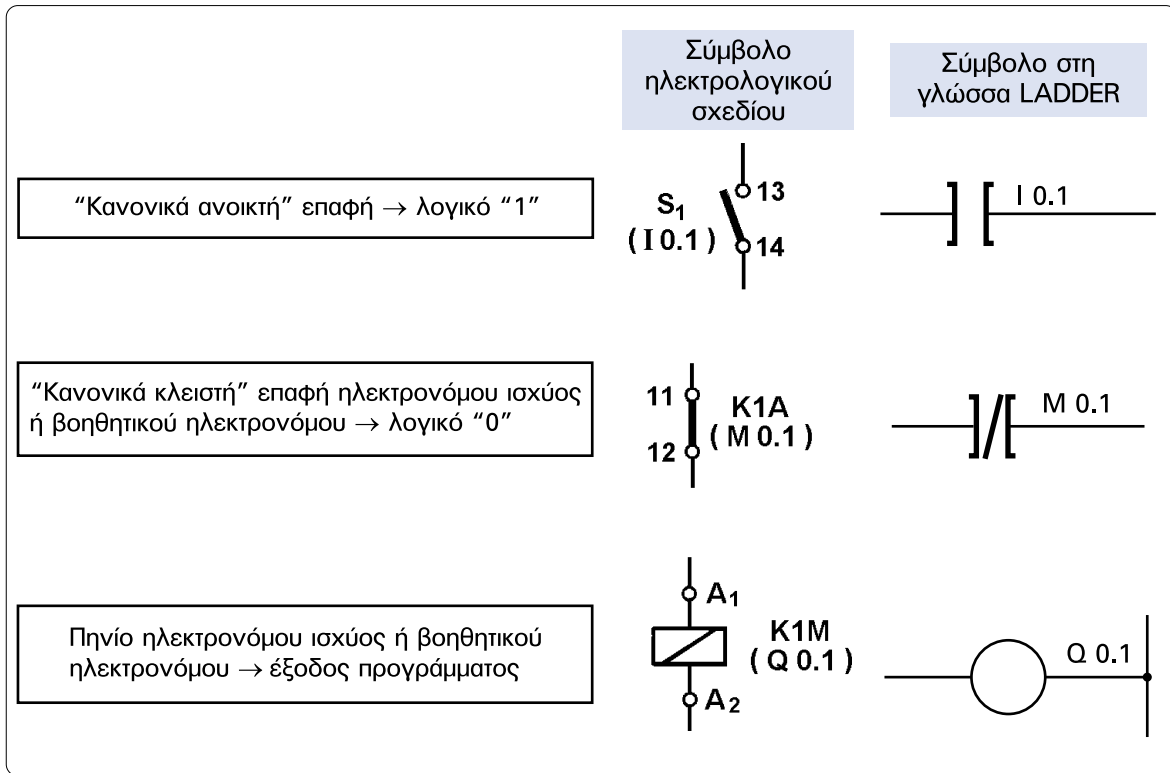
6.2 Δομή προγράμματος στη γλώσσα LADDER.

Το πρόγραμμα σε γλώσσα LADDER είναι ένα “σχέδιο”, “ένα διάγραμμα επαφών”, δηλαδή σχεδόν το ηλεκτρολογικό σχέδιο του αυτοματισμού.

Το διάγραμμα επαφών της γλώσσας LADDER σχεδιάζεται όχι “κατακόρυφα” αλλά “οριζόντια”. Δηλαδή σε ένα πρόγραμμα LADDER έχουμε δύο παράλληλες κατακόρυφες γραμμές (μπάρες), η αριστερή γραμμή παριστάνει τη μπάρα τροφοδοσίας με το υψηλό δυναμικό (+) και η δεξιά γραμμή τη μπάρα τροφοδοσίας με το χαμηλό δυναμικό (-). Μεταξύ των δύο γραμμών σχεδιάζουμε οριζόντια τους κλάδους του “κυκλώματος”.

Κάθε κλάδος του διαγράμματος Ladder, που ξεκινά από την αριστερή μπάρα και καταλήγει στη δεξιά μπάρα, αποτελεί μια “γραμμή προγράμματος”, η οποία αντιστοιχεί στην ομάδα εντολών της γλώσσας λίστα εντολών.

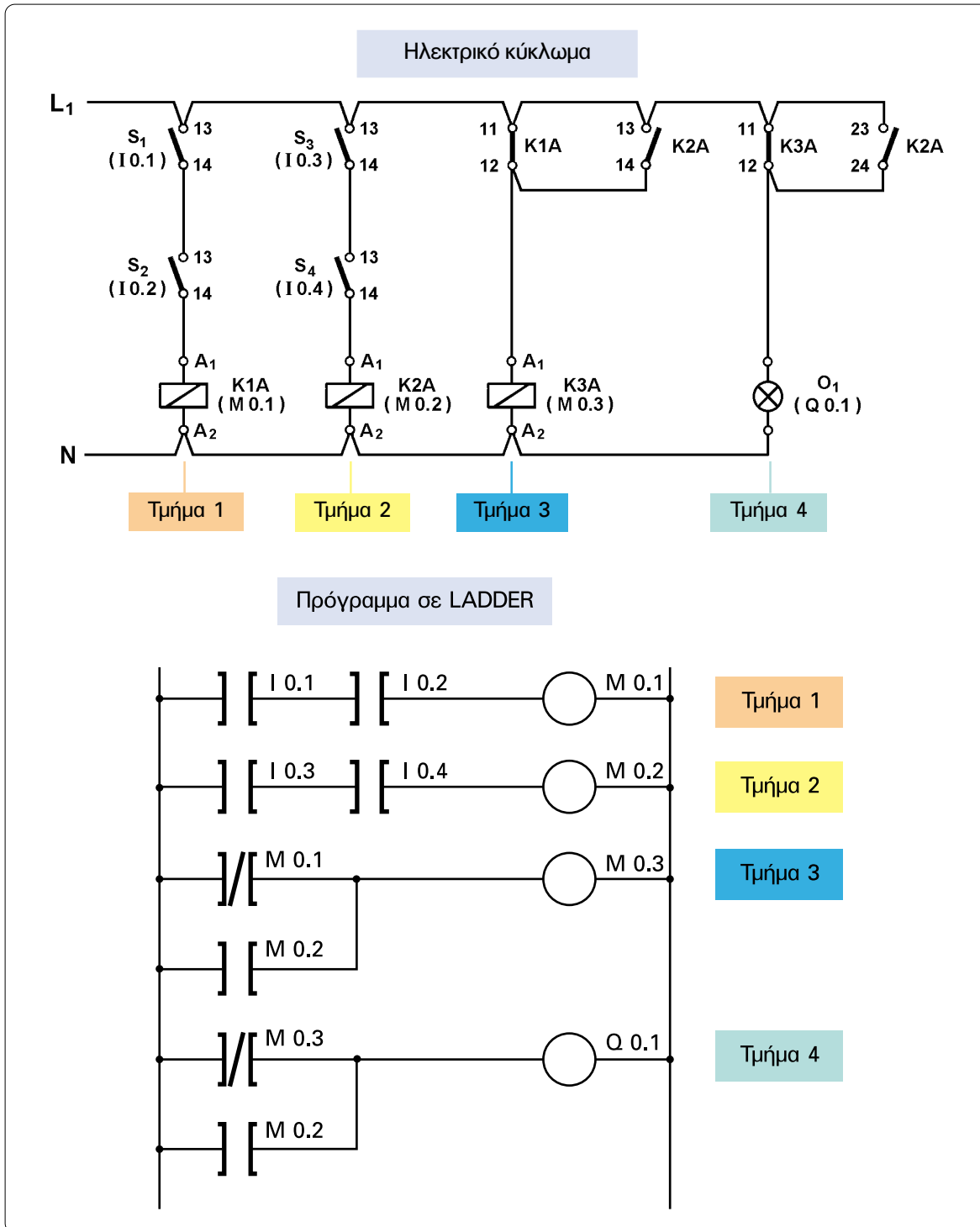
Τα σύμβολα που χρησιμοποιούνται είναι σύμβολα από την Αμερικανική τυποποίηση του ηλεκτρολογικού σχεδίου (ANSI). Τα βασικά σύμβολα δίνονται σε σχήμα 45. Δίπλα σε κάθε σύμβολο γράφεται το στοιχείο (η παράμετρος) στο οποίο αναφέρεται το σύμβολο.



Σχήμα 45

6.3 Παραδείγματα ανάπτυξης προγράμματος σε γλώσσα LADDER.

Θα γράψουμε το πρόγραμμα για το ηλεκτρικό κύκλωμα αυτοματισμού του παραδείγματος 3 της ενότητας 5.4.

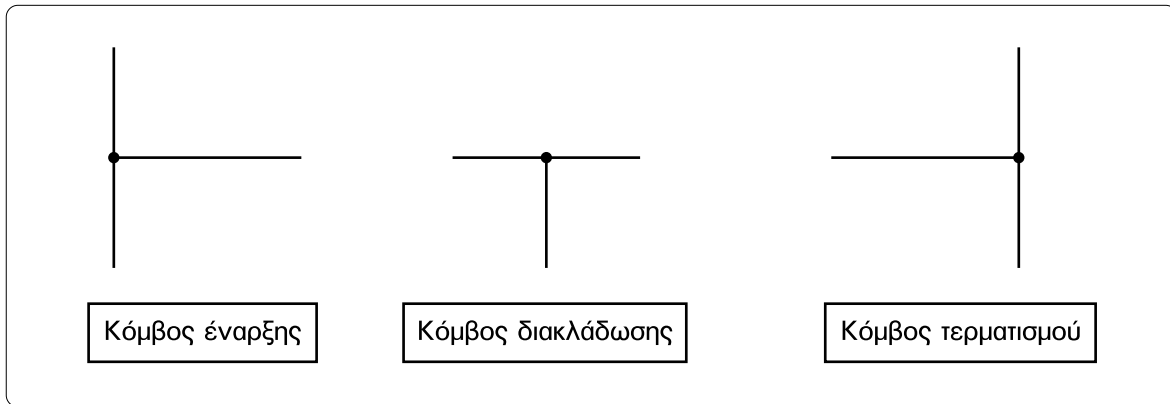


Σχήμα 46

ΠΑΡΑΤΗΡΗΣΕΙΣ

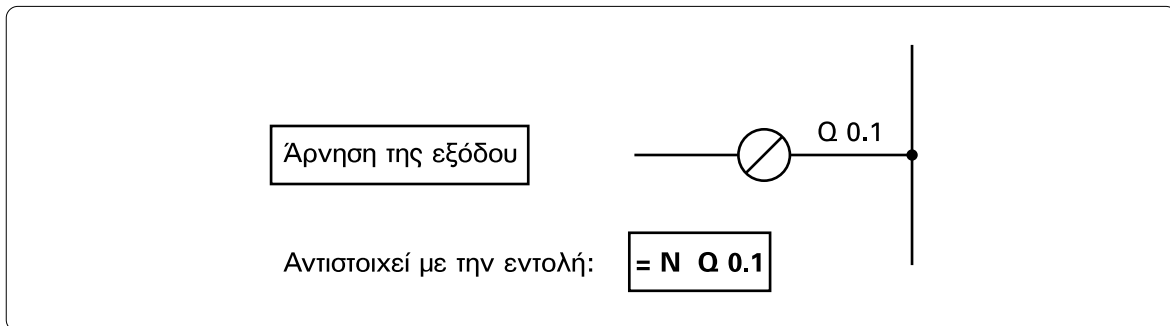
Ο σχεδιασμός ενός διαγράμματος επαφών LADDER υπόκειται σε περιορισμούς οι οποίοι εξαρτώνται από το κάθε PLC. Οι περιορισμοί αυτοί συνίστανται στα εξής:

- Ο αριθμός των στοιχείων (επαφές, πηνία) που μπορούμε να τοποθετήσουμε σε ένα κλάδο είναι περιορισμένος (εξαρτάται από το PLC).
- Η μορφή των διακλαδώσεων δεν μπορεί να είναι οποιαδήποτε. Για παράδειγμα δεν μπορούμε να έχουμε μεταγωγική επαφή. Οι διακλαδώσεις γίνονται μέσω ειδικών συμβόλων "κόμβων". Υπάρχει ο κόμβος έναρξης, κόμβος τερματισμού και ο κόμβος ενδιάμεσης διακλάδωσης (σχήμα 47).



Σχήμα 47: Στοιχεία της γλώσσας Ladder.

- Στη γλώσσα επαφών Ladder υπάρχει και η άρνηση της εξόδου πράγμα που δεν υπάρχει στα σύμβολα του ηλεκτρολογικού αυτοματισμού. Το σύμβολο είναι το εξής:

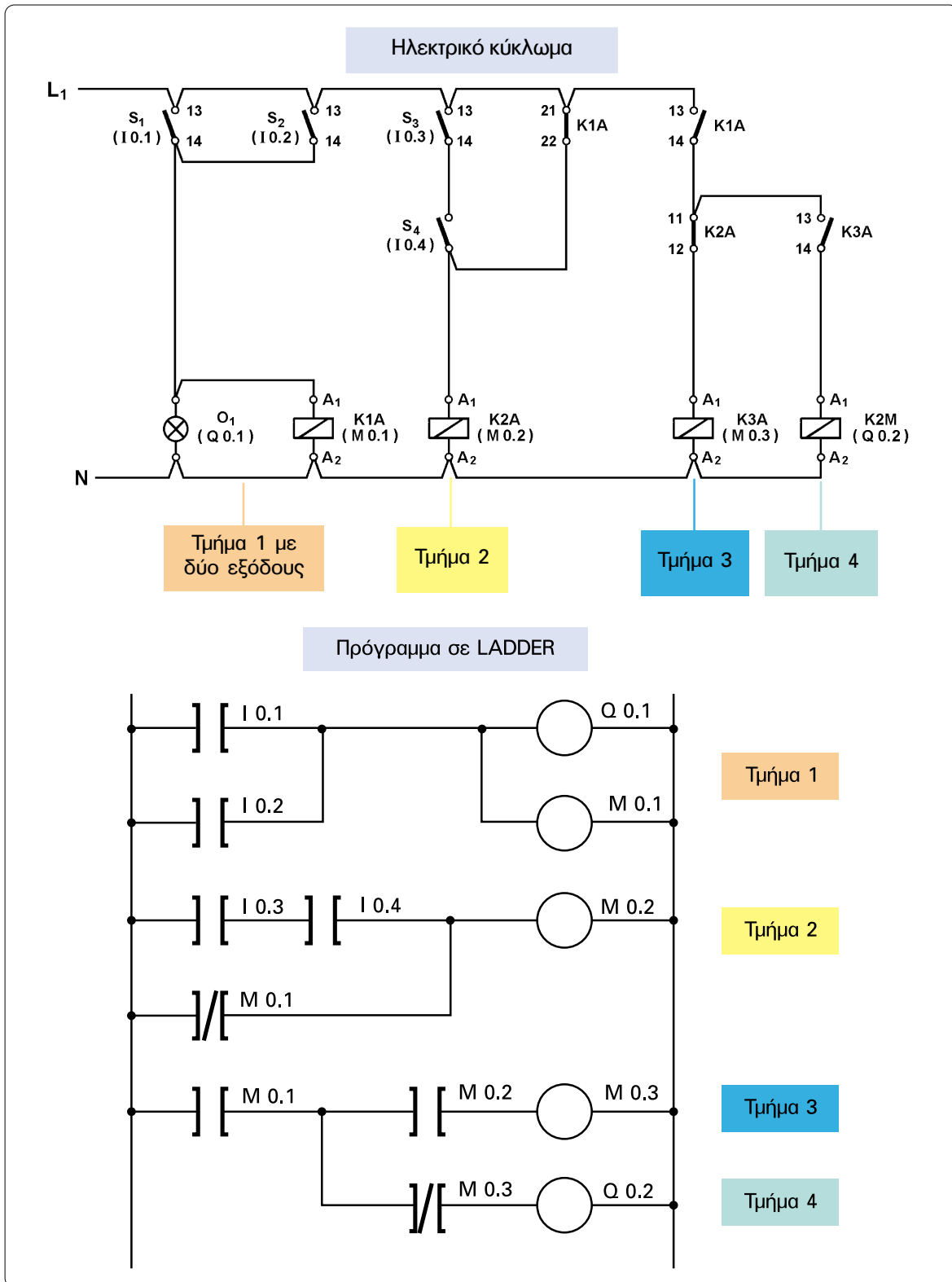


Σχήμα 48: Στοιχεία της γλώσσας Ladder.

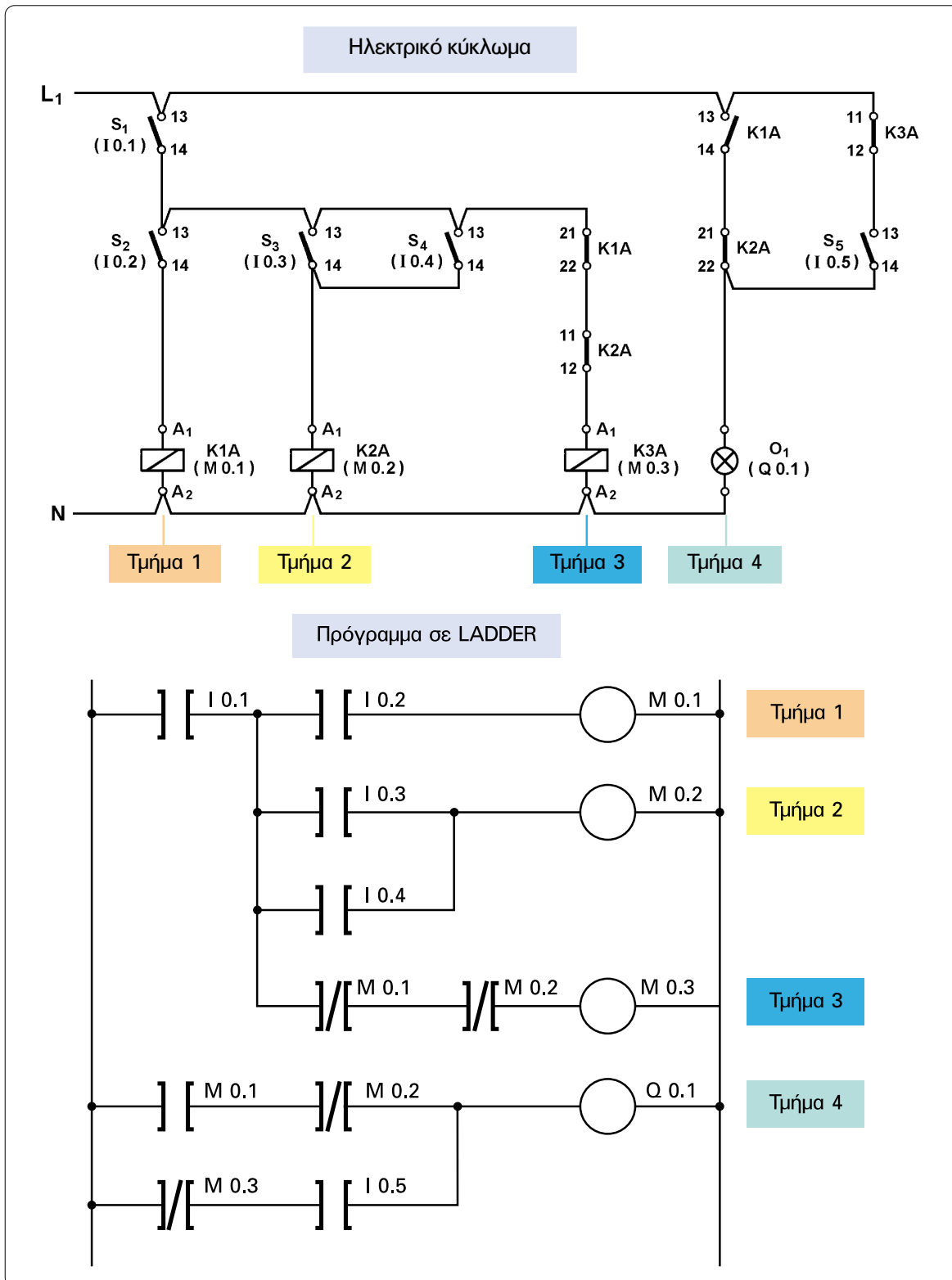
Παραδείγματα προγραμμάτων στη γλώσσα LADDER.

Θα δώσουμε στη συνέχεια το αντίστοιχο διάγραμμα Ladder για τα παραδείγματα που δώσαμε στη γλώσσα λίστα εντολών.

Παράδειγμα 4 της ενότητας 5.4.



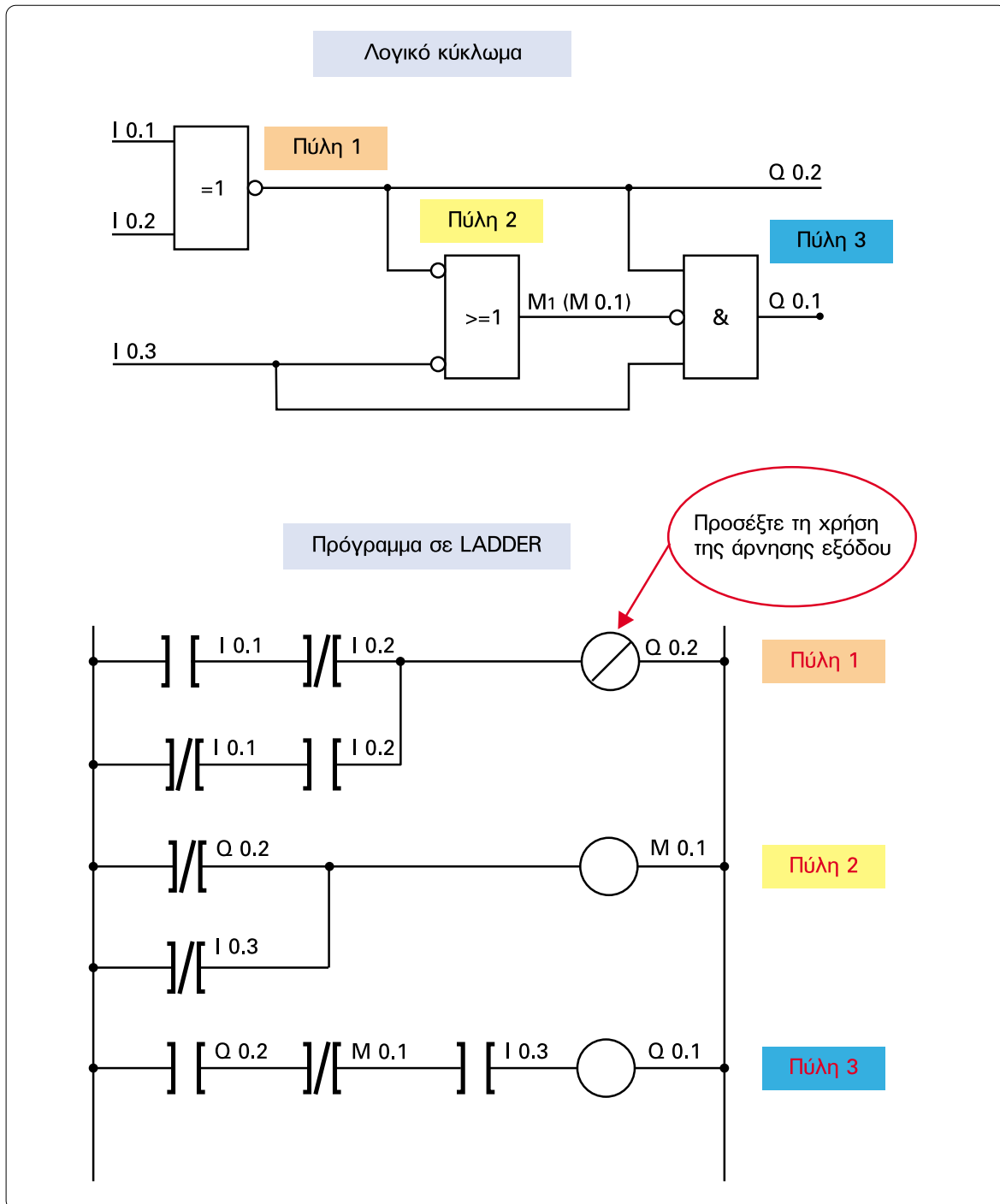
Παράδειγμα 5 της ενότητας 5.4.



Σχήμα 50

Παράδειγμα 2 της ενότητας 5.4.

Στο παράδειγμα αυτό θα φτιάξουμε το πρόγραμμα σε Ladder από το λογικό κύκλωμα. Δε θα δυσκολευτείτε γιατί έχουμε ήδη μάθει πως μετατρέπουμε λογικά κυκλώματα σε ηλεκτρολογικά.



Σχήμα 51

7. Ανάπτυξη προγράμματος στη γλώσσα λογικών γραφικών.

7.1 Γενικά.

Η γλώσσα λογικών γραφικών (ή λογικών διαγραμμάτων) δεν παρουσιάζει καμία δυσκολία για κάποιον, που γνωρίζει να μετατρέπει ηλεκτρολογικά σχέδια αυτοματισμού σε σχέδια με λογικές πύλες. Το πρόγραμμα είναι αυτό το ίδιο το λογικό κύκλωμα.

Είναι αλήθεια ότι οι εταιρείες κατασκευής PLC απέφευγαν στην αρχή να υιοθετήσουν αυτή τη γλώσσα, γιατί το τεχνικό κατεστημένο της βιομηχανίας δεν είχε γνώσεις πάνω στα λογικά κυκλώματα. Σήμερα η κατάσταση αυτή έχει αρχίσει να αλλάζει. Οι τεχνικοί είναι πλέον γνώστες των λογικών διαγραμμάτων και έτσι οι εταιρείες έχουν αρχίσει τελευταία να χρησιμοποιούν τη γλώσσα των λογικών γραφικών.

Τα PLC Millenium της Crouzet και LOGO της SIEMENS.

Σε γλώσσα λογικών γραφικών και μόνο σ' αυτήν προγραμματίζονται δύο μικρά PLC, το Millenium της Crouzet και το LOGO της SIEMENS. Πρόκειται για μικρά και χαμηλού κόστους συμπαγή PLC τα οποία έχουν:

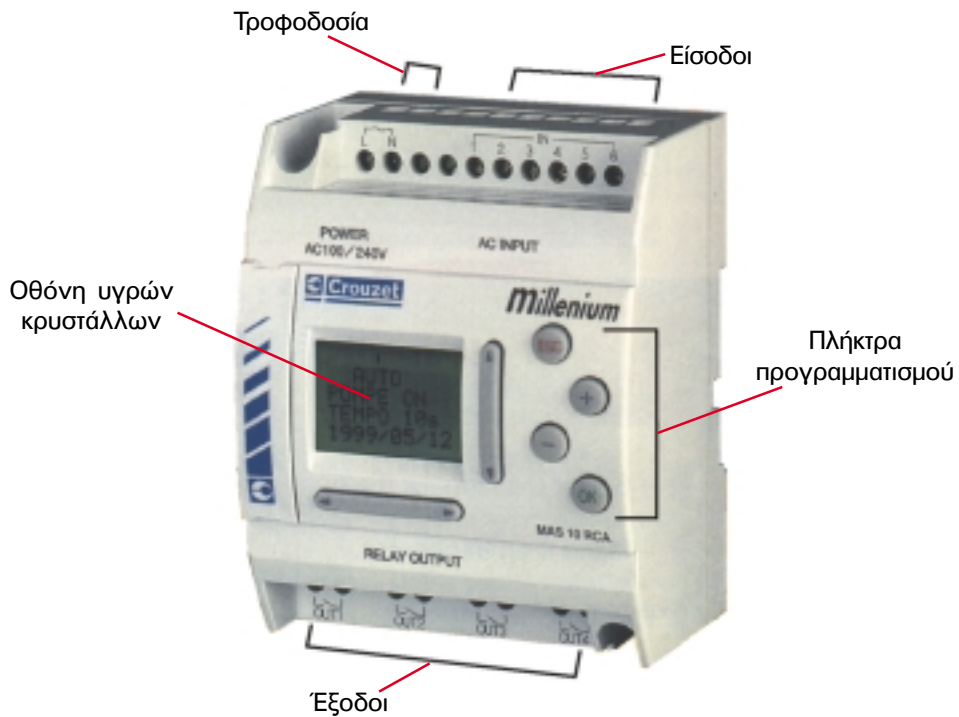
- 6 ψηφιακές εισόδους και 4 ψηφιακές εξόδους (ή 12 εισόδους και 8 εξόδους).
- Ενσωματωμένο τροφοδοτικό.
- Ενσωματωμένο προγραμματιστή. Ο προγραμματιστής αποτελείται από μια οθόνη υγρών κρυστάλλων και από 6 πλήκτρα. Με αυτά μπορούμε να προγραμματίσουμε σχεδιάζοντας το λογικό διάγραμμα.
- Ρολόι πραγματικού χρόνου. Μπορούν να προγραμματιστούν σ' αυτά ενέργειες σε ένα βάθος χρόνου από 1 μέχρι 4 έτη περίπου, ανάλογα με το μοντέλο.

Επίσης:

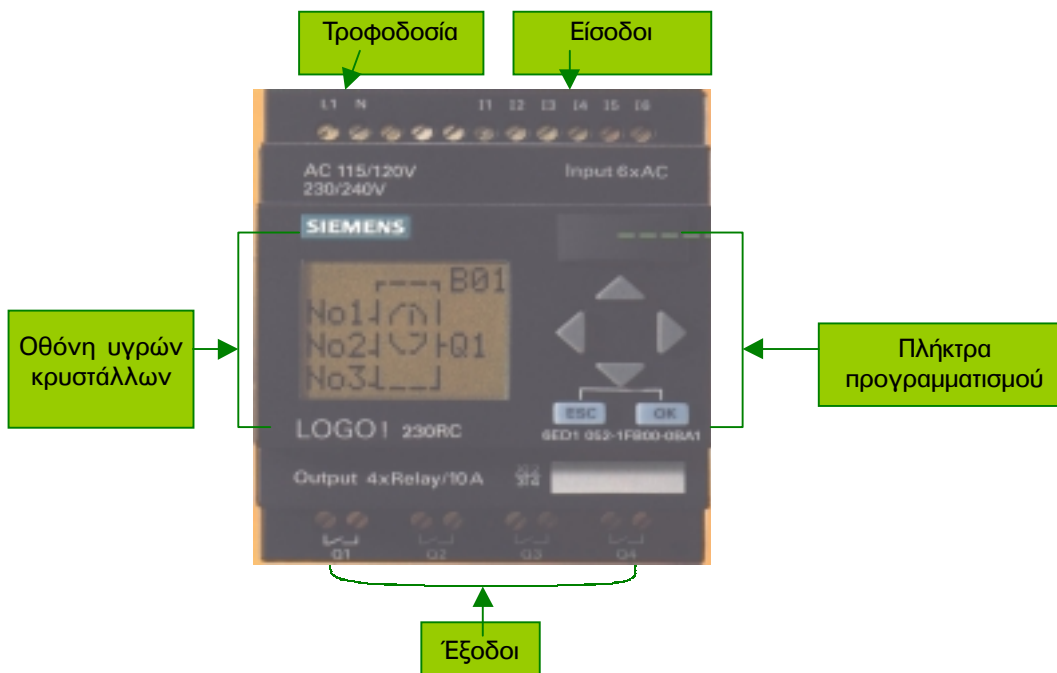
- Μπορούν να συνδεθούν με PC όποτε ο προγραμματισμός γίνεται πολύ πιο εύκολος.
- Μπορούν να επικοινωνούν και να ανταλλάσσουν σήματα και πληροφορίες με άλλες μεγαλύτερες συσκευές ελέγχου.
- Η γλώσσα προγραμματισμού τους είναι αρκετά πλούσια. Διαθέτουν αρκετές από τις εντολές και συναρτήσεις, που διαθέτει και ένα μεγάλο PLC.
- Στο Millenium της Crouzet οι 6 εισοδοι μπορούν να χρησιμοποιηθούν και ως εισοδοι αναλογικών σημάτων (στο μοντέλο με τροφοδοσία 24V DC).

Όσον αφορά τα ηλεκτρικά τους χαρακτηριστικά κυκλοφορούν σε διαφορετικά μοντέλα. Ενδεικτικά αναφέρουμε:

- Τροφοδοσία: 115V/230V AC ή 24V DC.
- Εξοδοι: μικροηλεκτρονόμοι με μέγιστο ρεύμα 8-10A για ωμικό φορτίο ή τρανζίστορ με μέγιστο ρεύμα 0,3A (στο LOGO της SIEMENS).



Σχήμα 52: PLC Millenium της Crouzet.

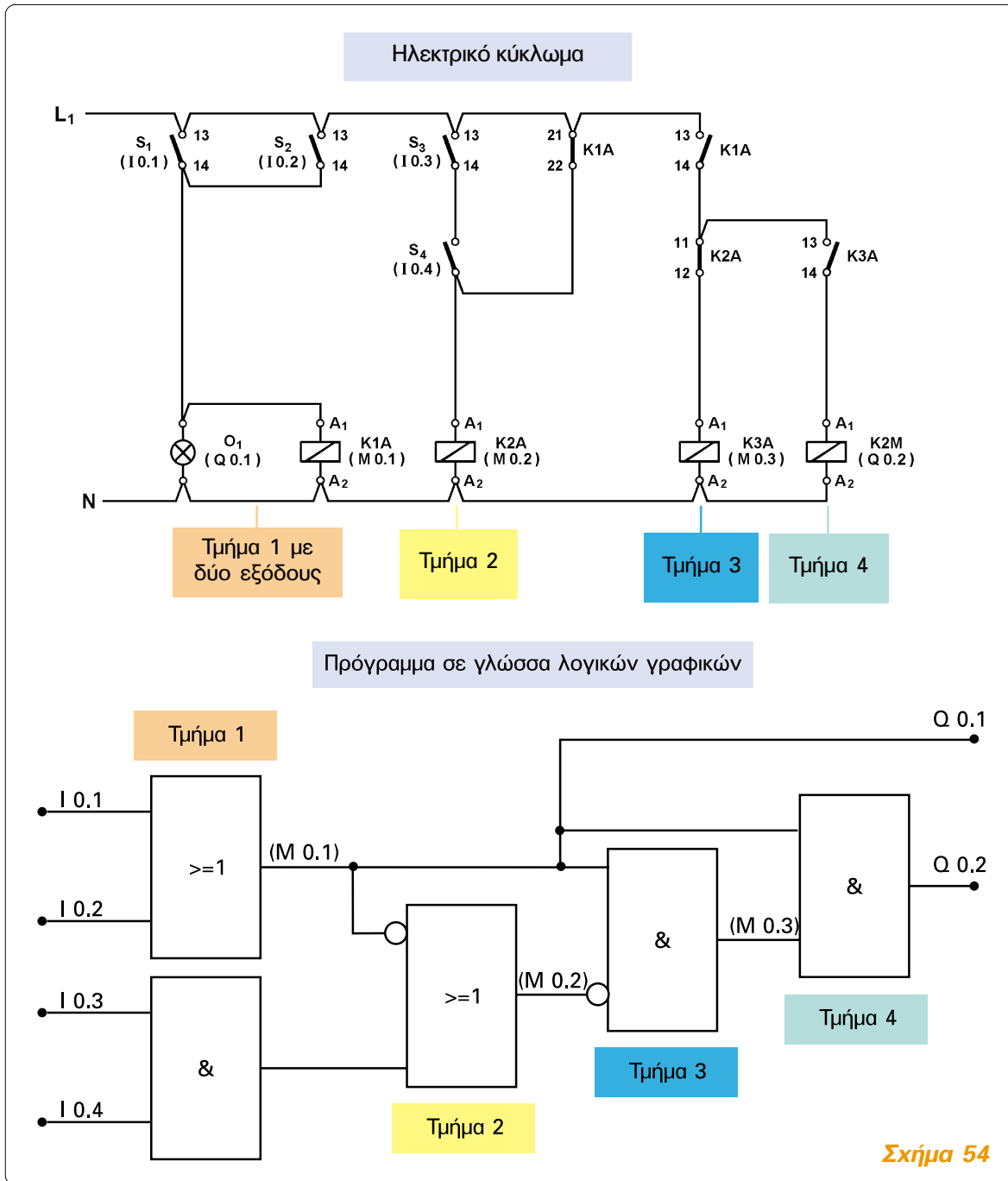


Σχήμα 53: PLC LOGO της SIEMENS.

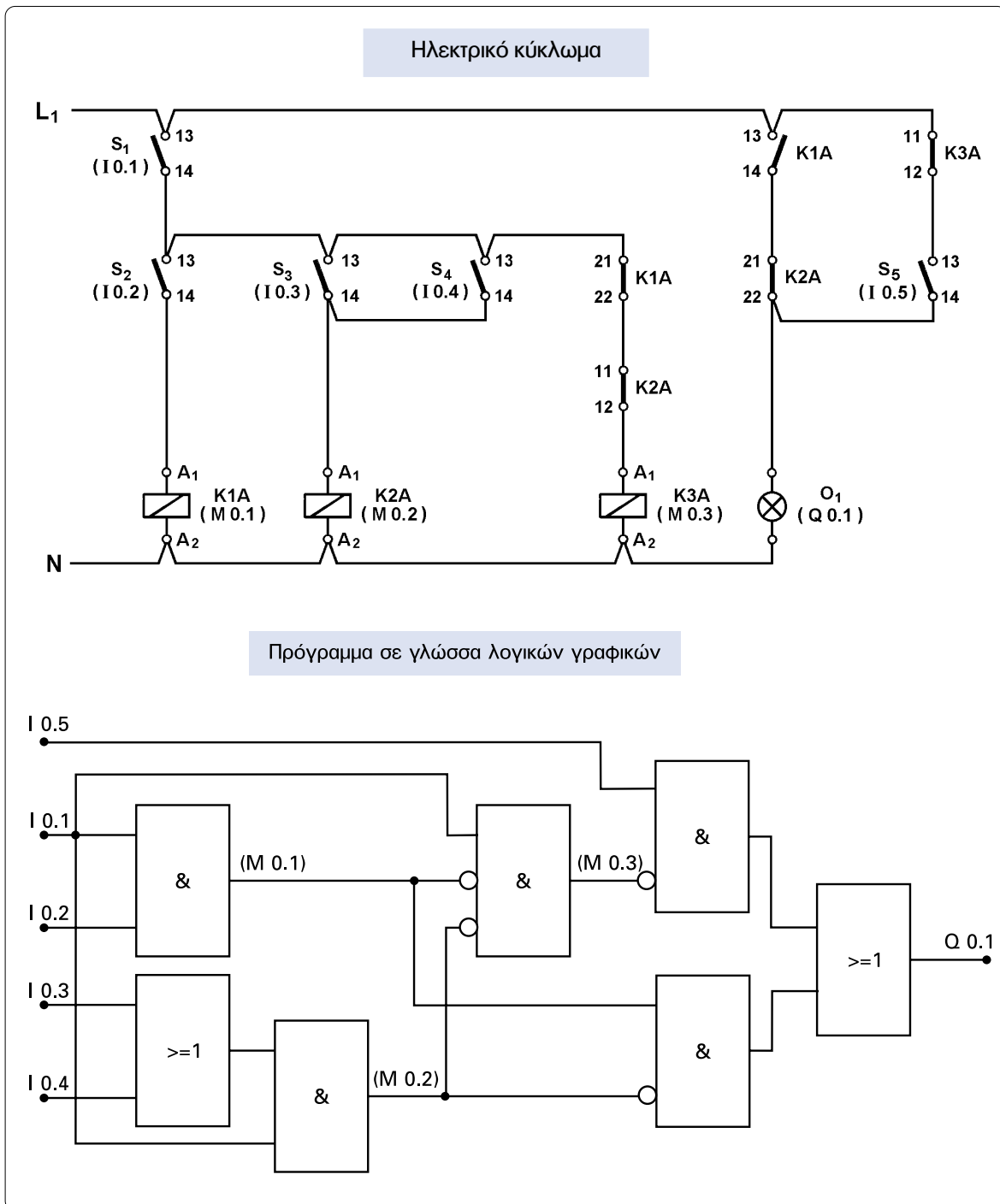
7.2 Παραδείγματα ανάπτυξης προγράμματος σε γλώσσα λογικών γραφικών.

Στα παραδείγματα που ακολουθούν θα προσπαθήσουμε να φτιάξουμε προγράμματα των προηγούμενων παραδειγμάτων σε γλώσσα λογικών γραφικών.

Παράδειγμα 4 της ενότητας 5.4.



Παράδειγμα 5 της παραγράφου 5.4.



Σχήμα 55

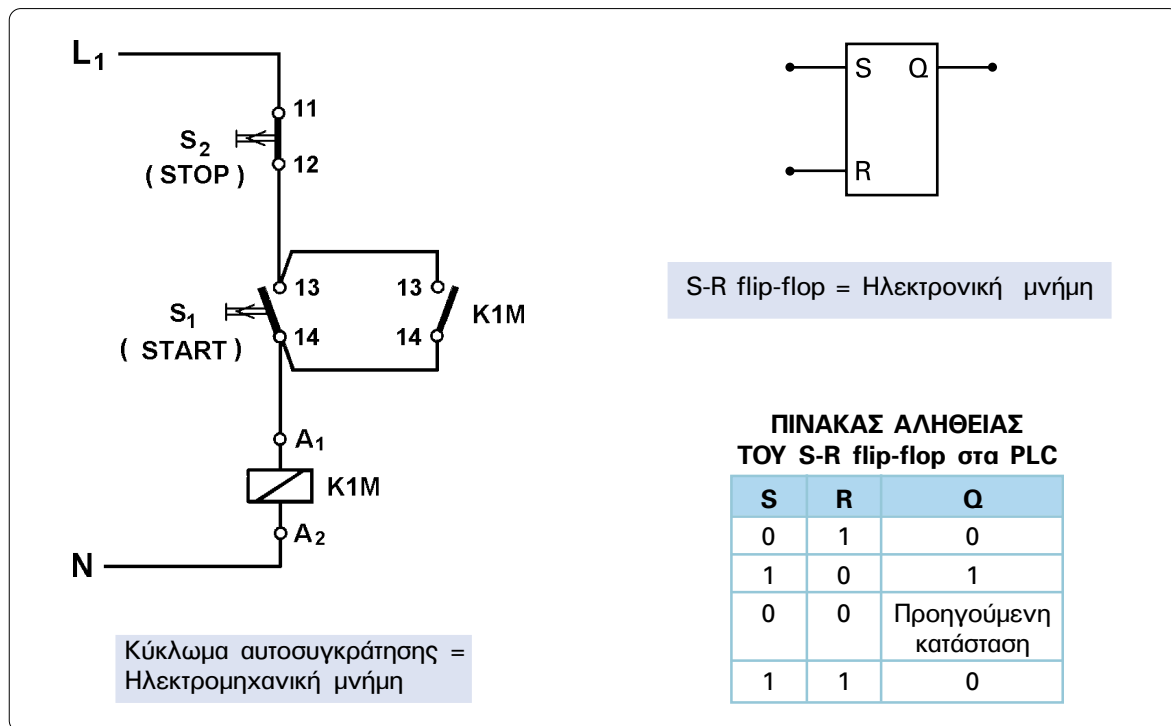
8. Ανάπτυξη προγραμμάτων σε ακολουθιακά κυκλώματα αυτοματισμού.

8.1 Γενικά.

Υπενθυμίζουμε ότι ακολουθιακό κύκλωμα αυτοματισμού ονομάζουμε εκείνο, στο οποίο έχουμε εξάρτηση των καταστάσεων κάποιων εξόδων του κυκλώματος από το χρόνο ή από προηγούμενες καταστάσεις στοιχείων του κυκλώματος. Στην περίπτωση που οι καταστάσεις κάποιων εξόδων του κυκλώματος εξαρτώνται από προηγούμενες καταστάσεις στοιχείων του κυκλώματος, λέμε ότι το κύκλωμα αυτοματισμού έχει μνήμη, θυμάται δηλαδή τις προηγούμενες καταστάσεις.

Ηλεκτρομηχανική μνήμη και PLC.

Ένα κύκλωμα αυτοσυγκράτησης με ηλεκτρονόμους, είναι μια ηλεκτρομηχανική μνήμη. Πιέζοντας το μπουτόν S_1 (START) ο ηλεκτρονόμος K1M ενεργοποιείται και δεν αλλάζει κατάσταση παρά μόνο αν πιέσουμε το μπουτόν S_2 (STOP). Δηλαδή, το κύκλωμα “θυμάται”, διατηρεί την κατάσταση που είχε, μέχρι εμείς να την αλλάξουμε, οπότε διατηρεί την νέα κατάσταση μέχρι και πάλι να την αλλάξουμε κ.ο.κ.



Σχήμα 56

Στα λογικά κυκλώματα την ηλεκτρονική μνήμη αποτελεί το κύκλωμα, που ονομάζουμε S-R flip-flop και λειτουργεί ακριβώς με τον ίδιο τρόπο. Δηλαδή, αν θέσουμε "1" στην είσοδο S, τότε η έξοδος Q γίνεται "1". Η έξοδος παραμένει σε "1" έστω και αν η είσοδος S έρθει σε "0". Για να επαναφέρουμε την έξοδο σε "0", πρέπει να θέσουμε "1" στην είσοδο R.

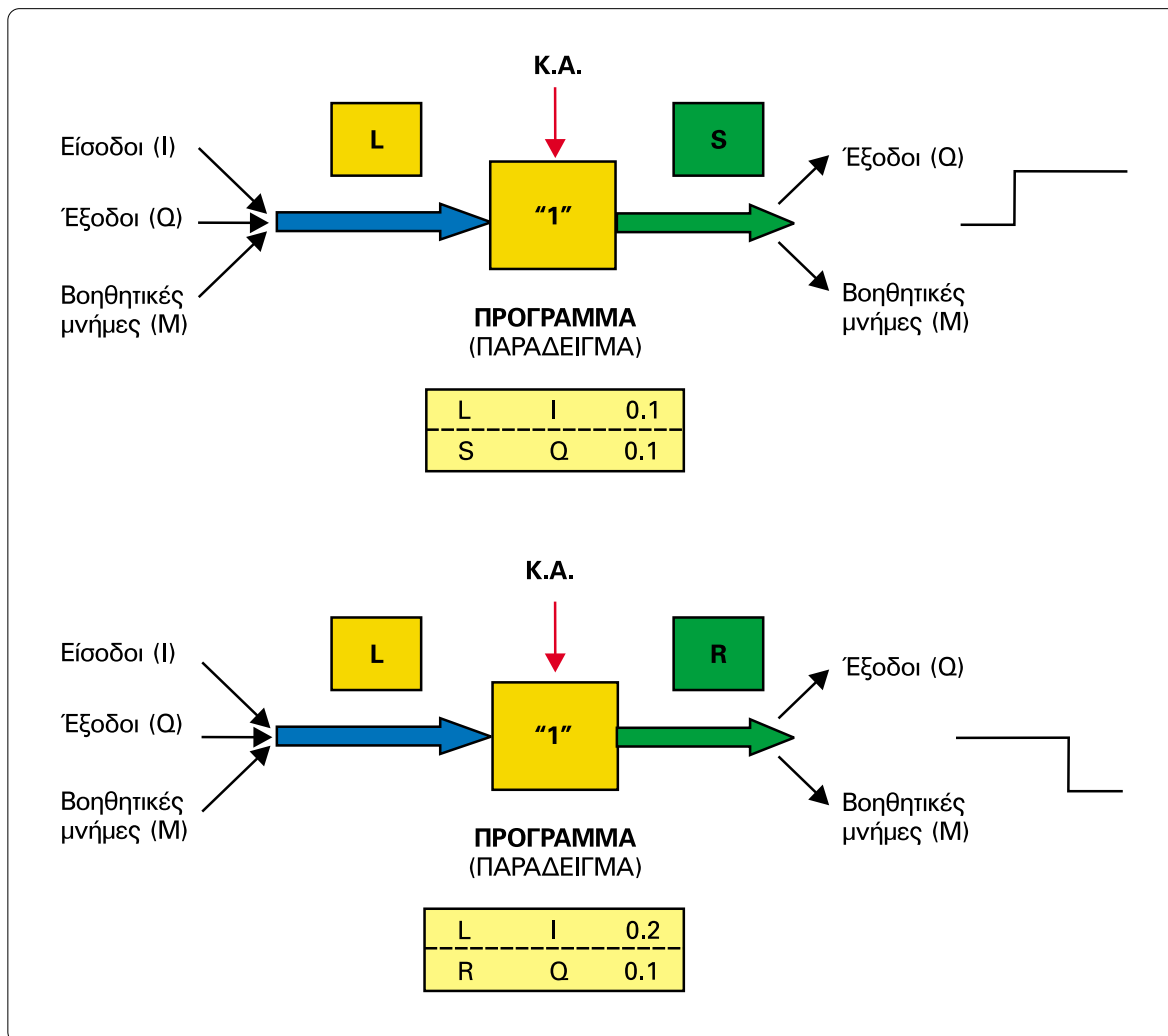
ΠΡΟΣΟΧΗ! Στο ηλεκτρονικό στοιχείο S-R flip-flop η κατάσταση των εισόδων $S = "1"$, $R = "1"$ συνεπάγεται ασταθή έξοδο.

Στα PLC και στο S-R flip-flop η κατάσταση $S = "1"$, $R = "1"$ οδηγεί την έξοδο σε λογική κατάσταση "0", όταν έχουμε προτεραιότητα στην είσοδο R, που είναι η συνηθισμένη περίπτωση.

8.2 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 (ή στο κύκλωμα αυτοσυγκράτησης όταν αναφερόμαστε στα ηλεκτρικά κυκλώματα). 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.



Σχήμα 57: Σχηματική παράσταση των εντολών SET και RESET.

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

Εντολή =			Εντολές S, R.		
L	I	0.1	L	I	0.1
=	Q	0.1	S	Q	0.1
			L	I	0.2
			R	Q	0.1

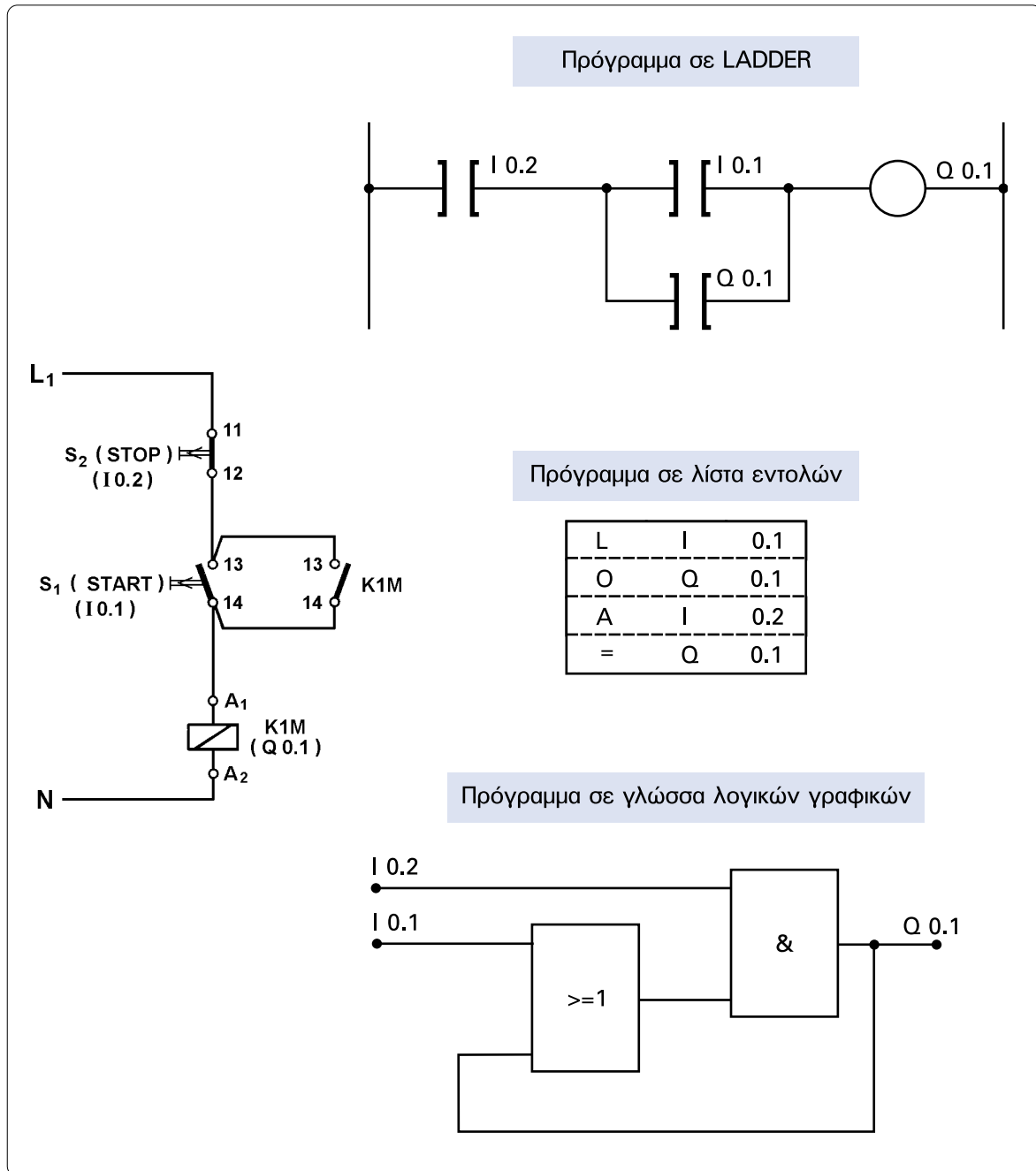
Στο πρόγραμμα που χρησιμοποιούμε την εντολή =, η έξοδος Q 0.1 είναι σε κατάσταση "1" όσο η είσοδος I 0.1 είναι σε κατάσταση "1". Μόλις η είσοδος I 0.1 αποκτήσει κατάσταση "0", τότε και η έξοδος Q 0.1 αποκτά κατάσταση "0".

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".

Στα διάφορα PLC στη γλώσσα Ladder αλλά και στη γλώσσα λογικών γραφικών το σύμβολο των εντολών SET και RESET είναι συνήθως το ίδιο το σύμβολο του S-R flip-flop, που δώσαμε παραπάνω στο σχήμα 56.

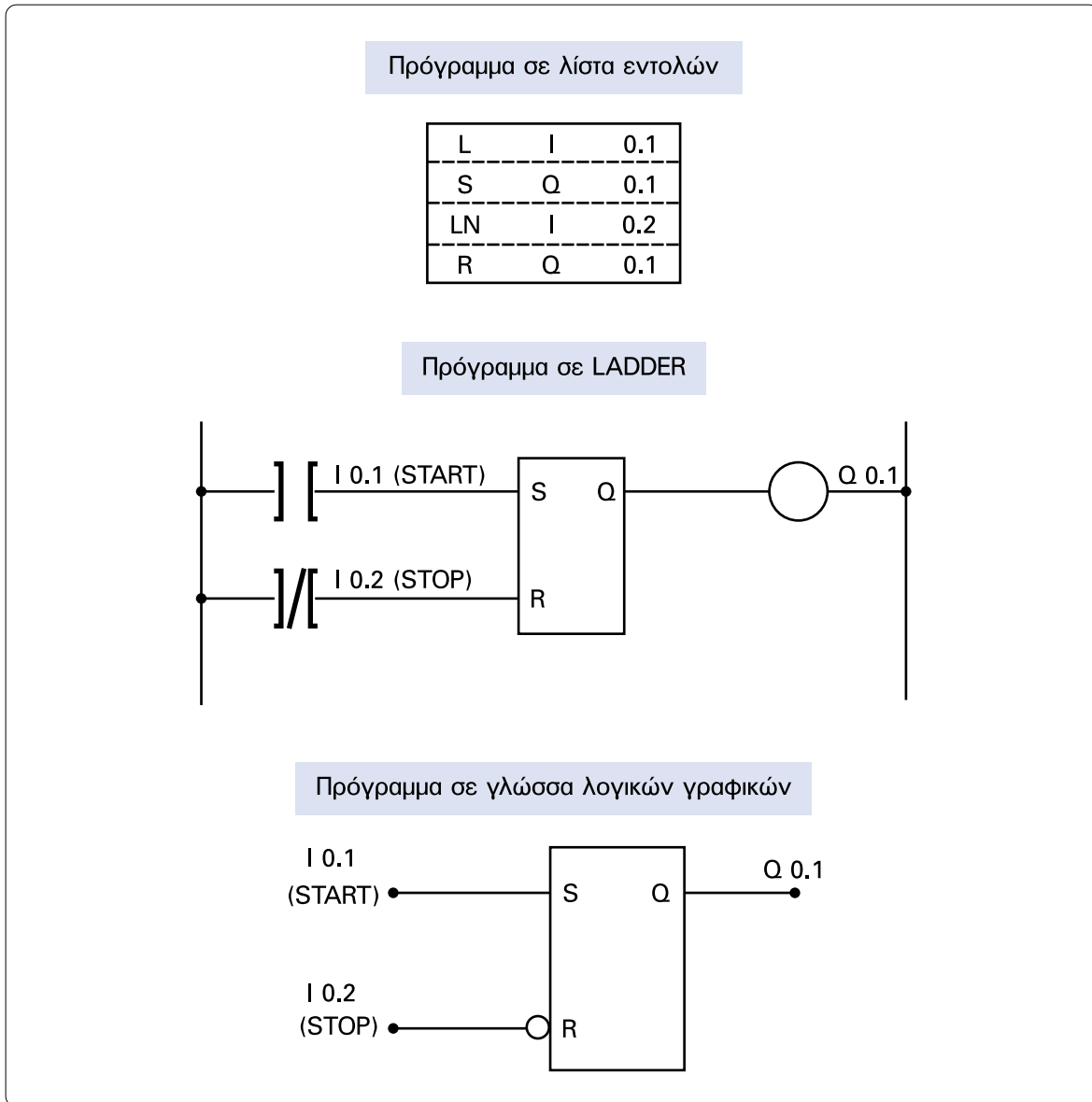
8.3 Το πρόγραμμα του κυκλώματος αυτοσυγκράτησης και στις τρεις γλώσσες με δύο τρόπους (με τις εντολές S και R και χωρίς αυτές).

1. Πρόγραμμα χωρίς τις εντολές SET και RESET.



Σχήμα 58: Πρόγραμμα για το κύκλωμα αυτοσυγκράτησης χωρίς τη χρησιμοποίηση των εντολών SET και RESET.

2. Πρόγραμμα με τη χρησιμοποίηση των εντολών SET και RESET (S-R flip-flop).



Σχήμα 59: Πρόγραμμα για το κύκλωμα αυτοσυγκράτησης με τη χρησιμοποίηση των εντολών SET και RESET (S-R flip-flop).

ΠΡΟΣΟΧΗ

Στην είσοδο RESET του flip-flop, που αντιστοιχεί το μπουτόν STOP, πρέπει να χρησιμοποιήσουμε NOT. Ο λόγος είναι ο εξής: η εντολή RESET ενεργοποιείται όταν στην είσοδό της έχουμε λογική τιμή "1". Όταν το μπουτόν STOP πιεστεί, η επαφή του ανοίγει και δίνει λογική κατάσταση "0" στην είσοδο I 0.2 του PLC. Με την αναστροφή η είσοδος RESET λαμβάνει τότε λογική τιμή "1".

9. Ανάπτυξη προγραμμάτων με χρονικές λειτουργίες.

9.1 Γενικά.

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.

Γενικά για τα χρονικά των PLC μπορούμε να πούμε ότι ισχύουν τα εξής:

- Σε όλα τα PLC υπάρχει το αντίστοιχο του χρονοηλεκτρονόμου με καθυστέρηση στην ενεργοποίηση (delay on).
- Πολλά PLC καλύπτουν τους χρονοηλεκτρονόμους που έχουμε ήδη περιγράψει, δηλαδή το χρονοηλεκτρονόμο με καθυστέρηση στην απενεργοποίηση (delay off) και τους χρονοηλεκτρονόμους δημιουργίας παλμού.
- Στα PLC που διαθέτουν πρόγραμμα (στοιχείο προγράμματος) μόνο για τη χρονική λειτουργία καθυστέρηση στην ενεργοποίηση, όλες οι άλλες χρονικές λειτουργίες δημιουργούνται προγραμματιστικά με τη χρησιμοποίηση αυτού.
- Στα διάφορα PLC το στοιχείο προγράμματος (σύμβολο) των χρονικών λειτουργιών, που χρησιμοποιείται στις γραφικές γλώσσες προγραμματισμού (Ladder και λογικών γραφικών) είναι ίδιο όσον αφορά στη μορφή του, στις εισόδους και εξόδους του.

Ονομασία και αριθμός των στοιχείων προγράμματος χρονικών λειτουργιών.

Στα διάφορα PLC χρησιμοποιούνται ονομασίες για τα στοιχεία προγράμματος χρονικών λειτουργιών, όπως T0, T1, T2, T3 κ.λπ. ή TR0, TR1, TR2, TR3 κ.λπ. Ο αριθμός των στοιχείων προγράμματος χρονικών λειτουργιών, που έχει ένα PLC, εξαρτάται από την δυναμικότητά του. Ο αριθμός αυτός μπορεί να είναι μικρός (π.χ. 8) μέχρι και πολύ μεγάλος (512 ή και μεγαλύτερος).

Το στοιχείο προγράμματος (σύμβολο) μιας χρονικής λειτουργίας και οι εισοδοί/εξοδοί αυτού.

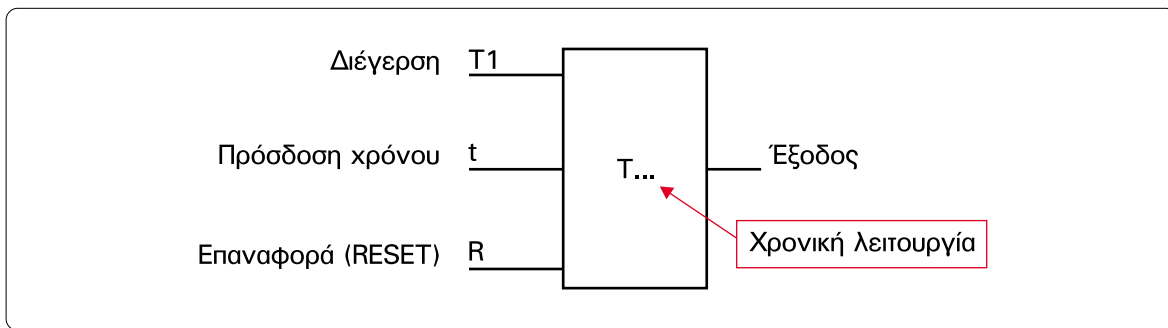
Τα στοιχεία προγράμματος των χρονικών λειτουργιών στις γραφικές γλώσσες προγραμματισμού έχουν συνήθως τρεις εισόδους και μία έξοδο. Οι εισοδοί είναι:

- *Είσοδος διέγερσης.* Στην είσοδο αυτή συνδέεται το λογικό κύκλωμα που ελέγχει τη χρονική λειτουργία.
- *Είσοδος που δίνεται ο χρόνος, ο οποίος θα μετρηθεί στη χρονική λειτουργία.* Ο τρόπος με τον οποίο προσδιορίζεται ο χρόνος ποικίλει στα διάφορα PLC.
- *Είσοδος επαναφοράς (είσοδος μηδενισμού, RESET).* Αν στην είσοδο αυτή έχουμε κατάσταση "1", η κατάσταση της εξόδου του χρονικού γίνεται "0" και μηδενίζεται ο χρόνος του σε όποιο σημείο της διαδικασίας και αν βρίσκεται.

Ο χαρακτηρισμός των εισόδων/εξόδων του στοιχείου προγράμματος μιας χρονικής λειτουργίας δεν είναι ο ίδιος σε όλα τα PLC.

Στην περίπτωση που το PLC υποστηρίζει πολλές διαφορετικές χρονικές λειτουργίες, ο προσδιορισμός μιας χρονικής λειτουργίας γίνεται με διαφορετικό τρόπο στα διάφορα PLC (αλλάζει η ονομασία της εισόδου διέγερσης ή προσδιορίζεται από τον αύξοντα αριθμό του χρονικού ή διαφοροποιείται εν μέρει το σύμβολο του στοιχείου προγράμματος της χρονικής λειτουργίας κ.λπ.)

Στις εφαρμογές, που θα αναπτύξουμε στη συνέχεια στις γραφικές γλώσσες προγραμματισμού, το στοιχείο προγράμματος, που θα χρησιμοποιήσουμε για τις χρονικές λειτουργίες είναι αυτό του σχήματος 60.

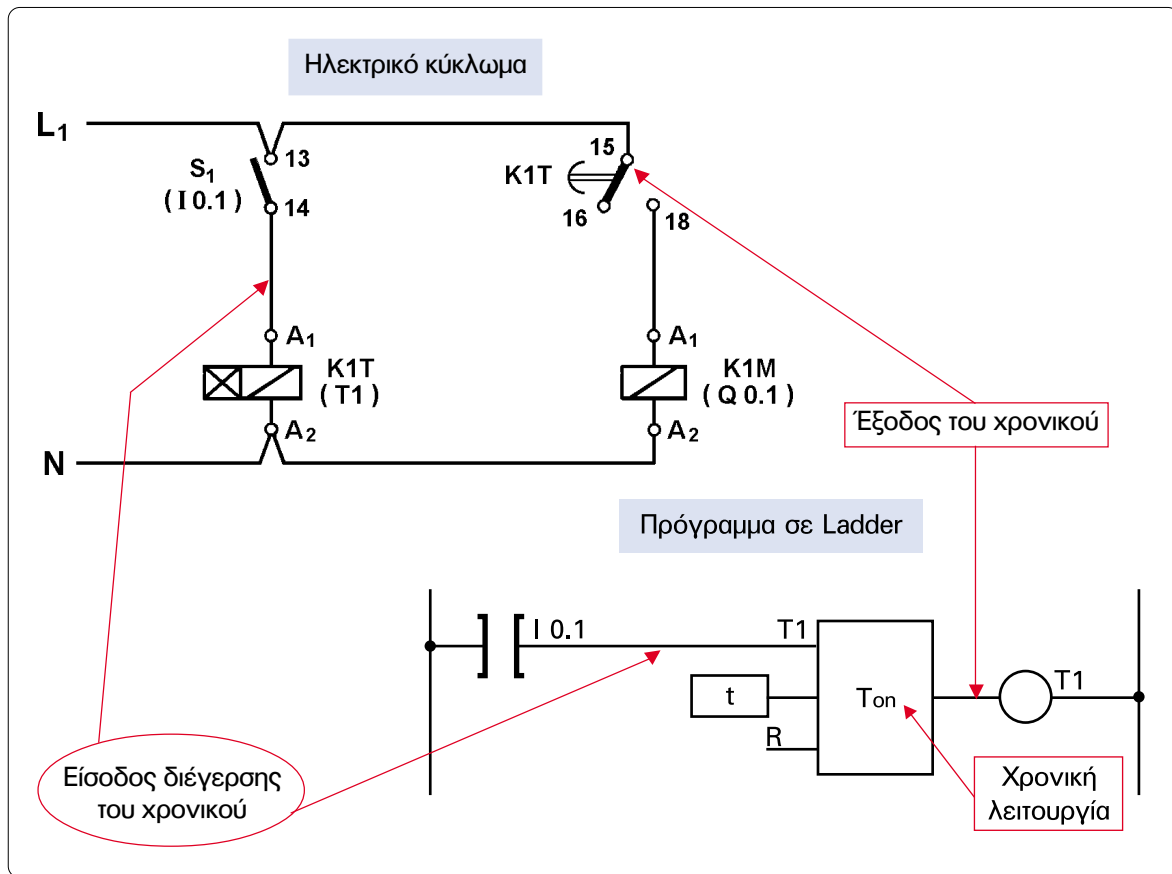


Σχήμα 60

9.2 Παραδείγματα προγραμμάτων με χρονικές λειτουργίες σε γλώσσα Ladder.

Οι εφαρμογές για τις οποίες θα φτιάξουμε το πρόγραμμα είναι εφαρμογές που έχουμε μελετήσει στην ενότητα 6.1.8 του βιβλίου Συστήματα Αυτοματισμών - Α΄ τόμος (σελίδες 238-253).

Παράδειγμα 1. Πρόγραμμα χρονικής λειτουργίας καθυστέρησης στην ενεργοποίηση (delay on).

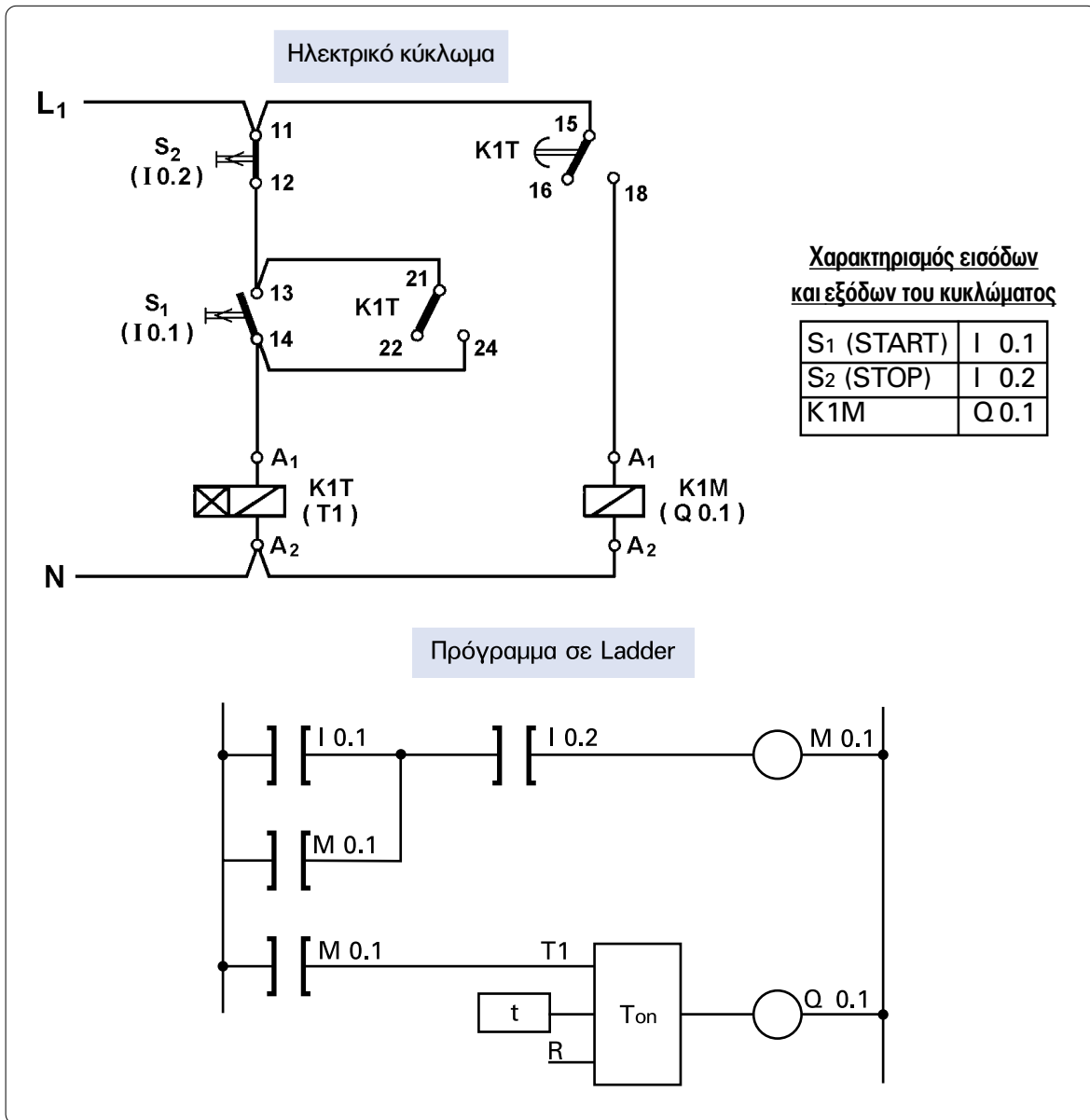


Σχήμα 61

- Χρειάζεται προσοχή στο "σχεδιασμό" του προγράμματος, γιατί ενώ στο σχέδιο του κλασικού αυτοματισμού ο χρονοηλεκτρονόμος έχει τη μορφή πηνίου ηλεκτρονόμου, στη γλώσσα Ladder είναι ένα ενιαίο "κουτί". Στην είσοδο διέγερσης του χρονικού συνδέουμε τις "επαφές" (τα στοιχεία προγράμματος) διέγερσης και στην έξοδο τα "πηνιά" (τις εξόδους) του προγράμματος.
- Στην περίπτωση που το PLC διαθέτει διάφορα είδη χρονικών, το πρόγραμμα είναι ίδιο, απλά αλλάζει ο χαρακτηρισμός του χρονικού. Π.χ. αν στο προηγούμενο πρόγραμμα χρησιμοποιήσουμε χρονικό με καθυστέρηση στην απενεργοποίηση (delay off) αντί για χρονικό με καθυστέρηση στην ενεργοποίηση, έχουμε το πρόγραμμα, που δίνει τη χρονική λειτουργία καθυστέρηση στην απενεργοποίηση.
- Ιδιαίτερη προσοχή χρειάζεται, όταν έχουμε στο σχέδιο του κλασικού αυτοματισμού επαφές του χρονοηλεκτρονόμου χωρίς χρονική καθυστέρηση. Στην περίπτωση αυτή

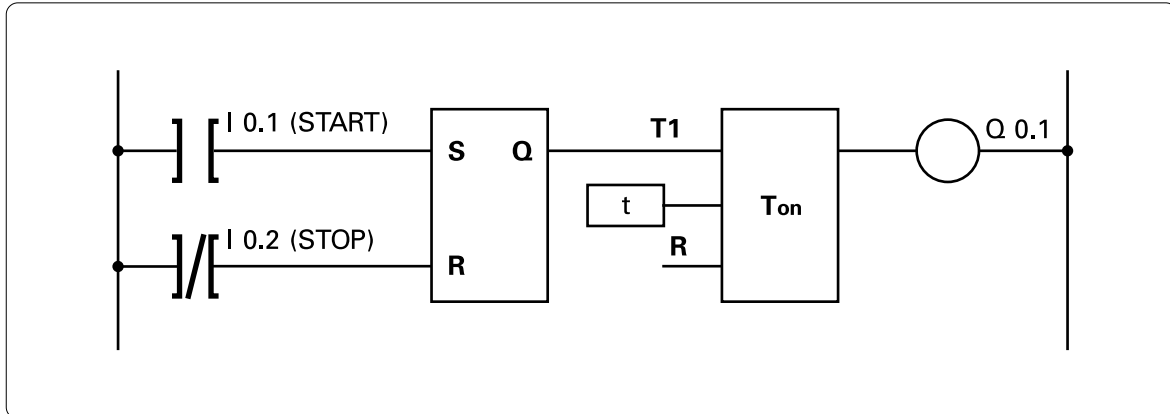
χρησιμοποιούμε στο πρόγραμμα μια βοηθητική μνήμη, η οποία διεγείρεται ταυτόχρονα με το χρονικό και για τις επαφές του χρονικού χωρίς χρονική καθυστέρηση χρησιμοποιούμε τη βοηθητική μνήμη. Αυτό φαίνεται στο επόμενο παράδειγμα.

Παράδειγμα 2. Πρόγραμμα χρονικής λειτουργίας καθυστέρησης στην ενεργοποίηση με αυτοσυγκράτηση (retentive delay on).



Σχήμα 62

Το παραπάνω πρόγραμμα μπορεί να γραφεί, όπως δίνεται στη συνέχεια (σχήμα 63), με τη χρησιμοποίηση του S-R flip-flop (των εντολών SET, RESET), που ήδη έχουμε δει στην ενότητα 8.3.



Σχήμα 63

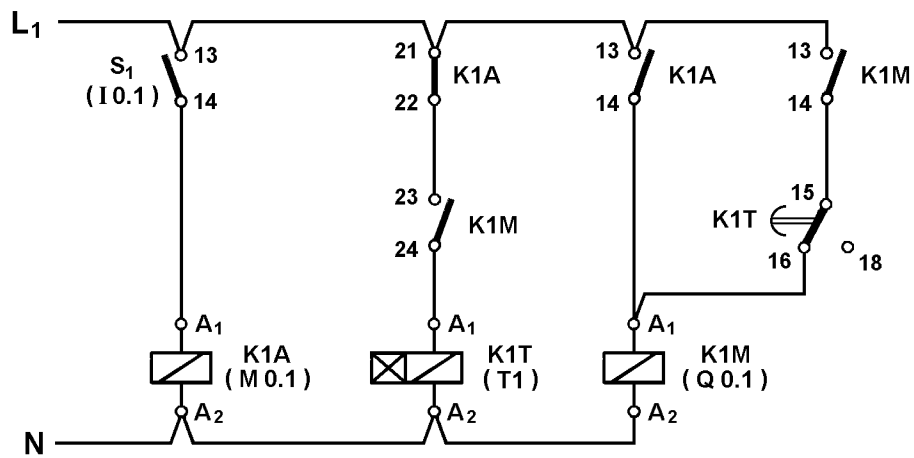
Παράδειγμα 3. Πρόγραμμα χρονικής λειτουργίας καθυστέρησης στην απενεργοποίηση (delay off) με τη χρησιμοποίηση του στοιχείου προγράμματος της χρονικής λειτουργίας καθυστέρησης στην ενεργοποίηση (delay on).

Σε πολλά PLC τα διαθέσιμα χρονικά δεν καλύπτουν όλες τις χρονικές λειτουργίες, που αναπτύχθηκαν στο κεφάλαιο 6 (ενότητα 6.1.7, σελίδες 232-237) του βιβλίου Συστήματα Αυτοματισμών, Α' τόμος. Σε αρκετές περιπτώσεις PLC όλες οι χρονικές λειτουργίες προγραμματίζονται με τη χρησιμοποίηση της χρονικής λειτουργίας καθυστέρησης στην ενεργοποίηση (delay on). Στο παράδειγμα αυτό δημιουργούμε ένα κύκλωμα χρονικής λειτουργίας καθυστέρησης στην απενεργοποίηση (delay off) με τη χρησιμοποίηση του χρονικού καθυστέρησης στην ενεργοποίηση (delay on).

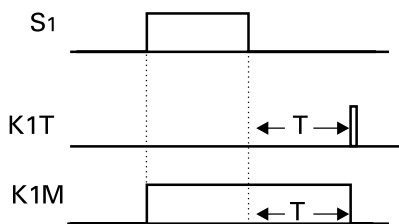
Εξήγηση της λειτουργίας του ηλεκτρικού κυκλώματος:

Το κλείσιμο του διακόπτη S_1 ενεργοποιεί το βοηθητικό ηλεκτρονόμο K1A. Στη συνέχεια, μέσω της επαφής 13-14 του τελευταίου ενεργοποιείται ο ηλεκτρονόμος ισχύος K1M (έξοδος). Ο ηλεκτρονόμος K1M αυτοσυγκρατείται με την επαφή του 13-14. Όταν ανοίξει ο διακόπτης S_1 , απενεργοποιείται ο ηλεκτρονόμος K1A και μέσω της επαφής 21-22 του K1A τροφοδοτείται το χρονικό K1T. Μετά την παρέλευση του χρόνου, που έχει ρυθμιστεί στο χρονικό, η επαφή του χρονικού 15-16/18 αλλάζει κατάσταση και διακόπτει την τροφοδοσία του K1M. Στη συνέχεια διακόπτεται και η τροφοδοσία του χρονικού μέσω της επαφής 23-24 του K1M.

Ηλεκτρικό κύκλωμα



Χρονοδιάγραμμα

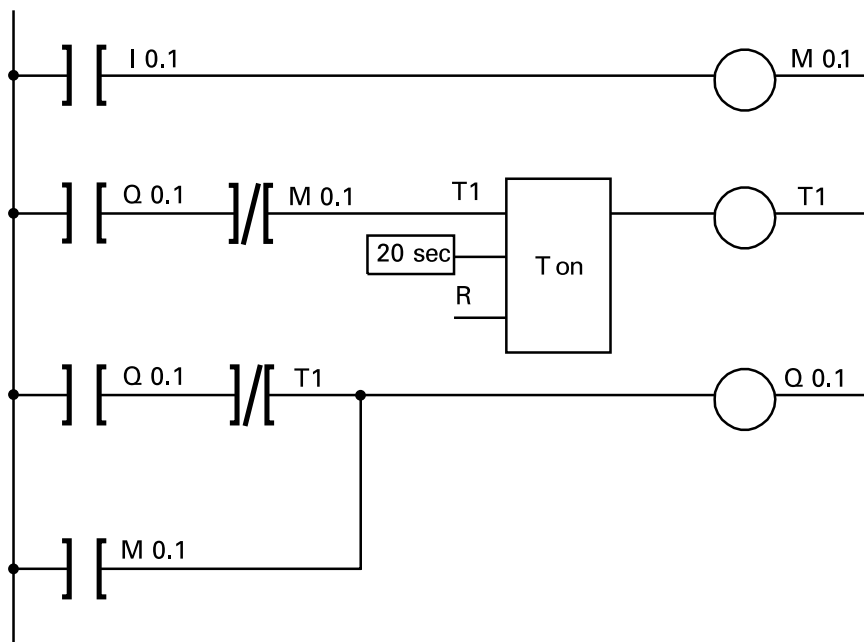


Χαρακτηρισμός εισόδων
εξόδων του κυκλώματος

S ₁	I 0.1
K1M	Q 0.1

Χρόνος καθυστέρησης T=20 sec

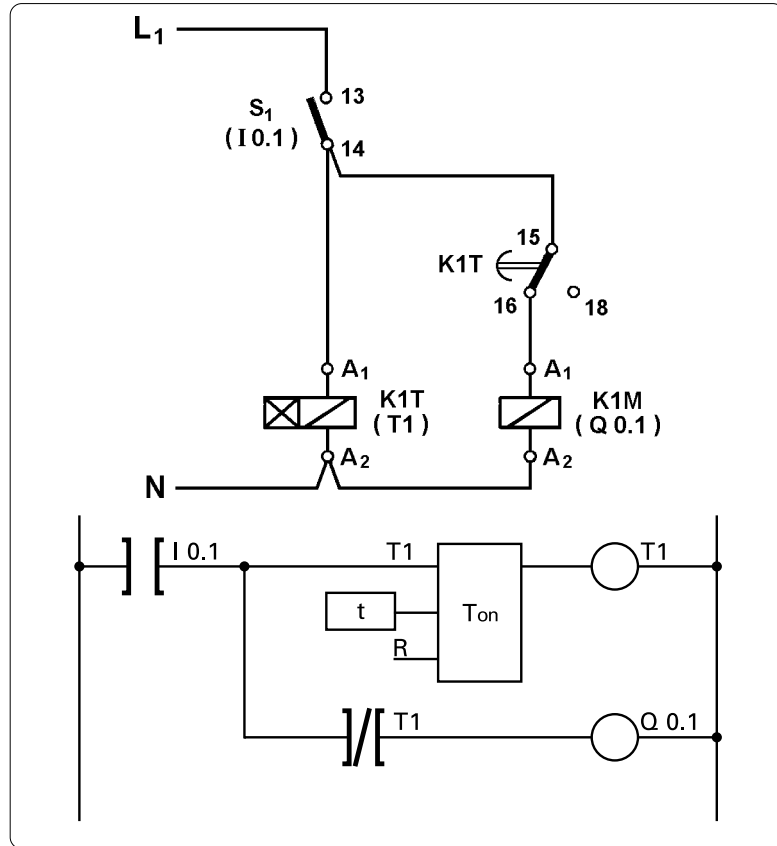
Πρόγραμμα σε Ladder



Σχήμα 64

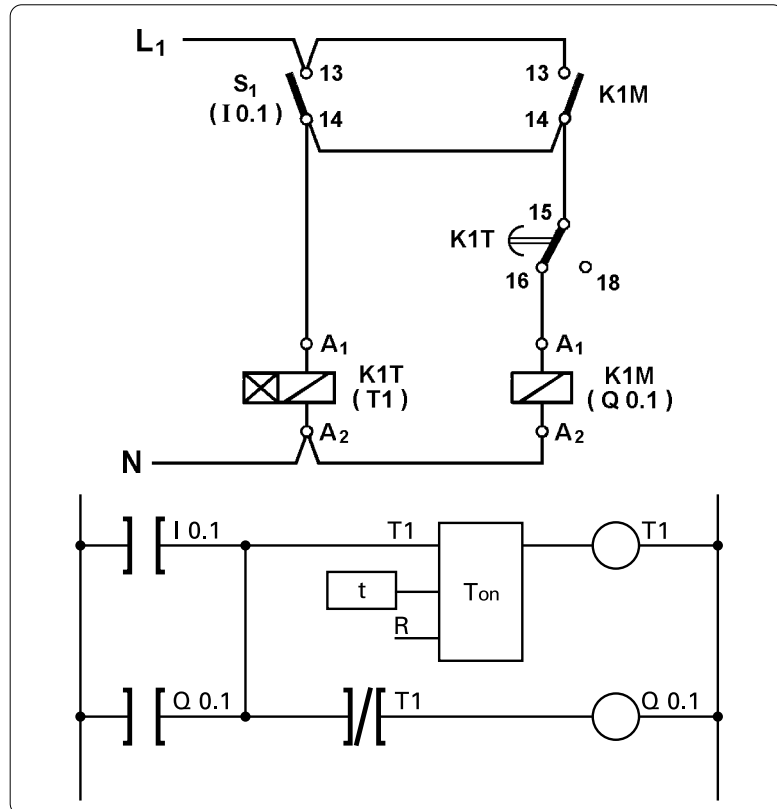
Παράδειγμα 4. Πρόγραμμα χρονικής λειτουργίας δημιουργίας παλμού εξαρτώμενου και από τη χρονική διάρκεια του σήματος διέγερσης (one shot) με τη χρησιμοποίηση του στοιχείου προγράμματος της χρονικής λειτουργίας καθυστέρησης στην ενεργοποίηση (delay on).

Σχήμα 65

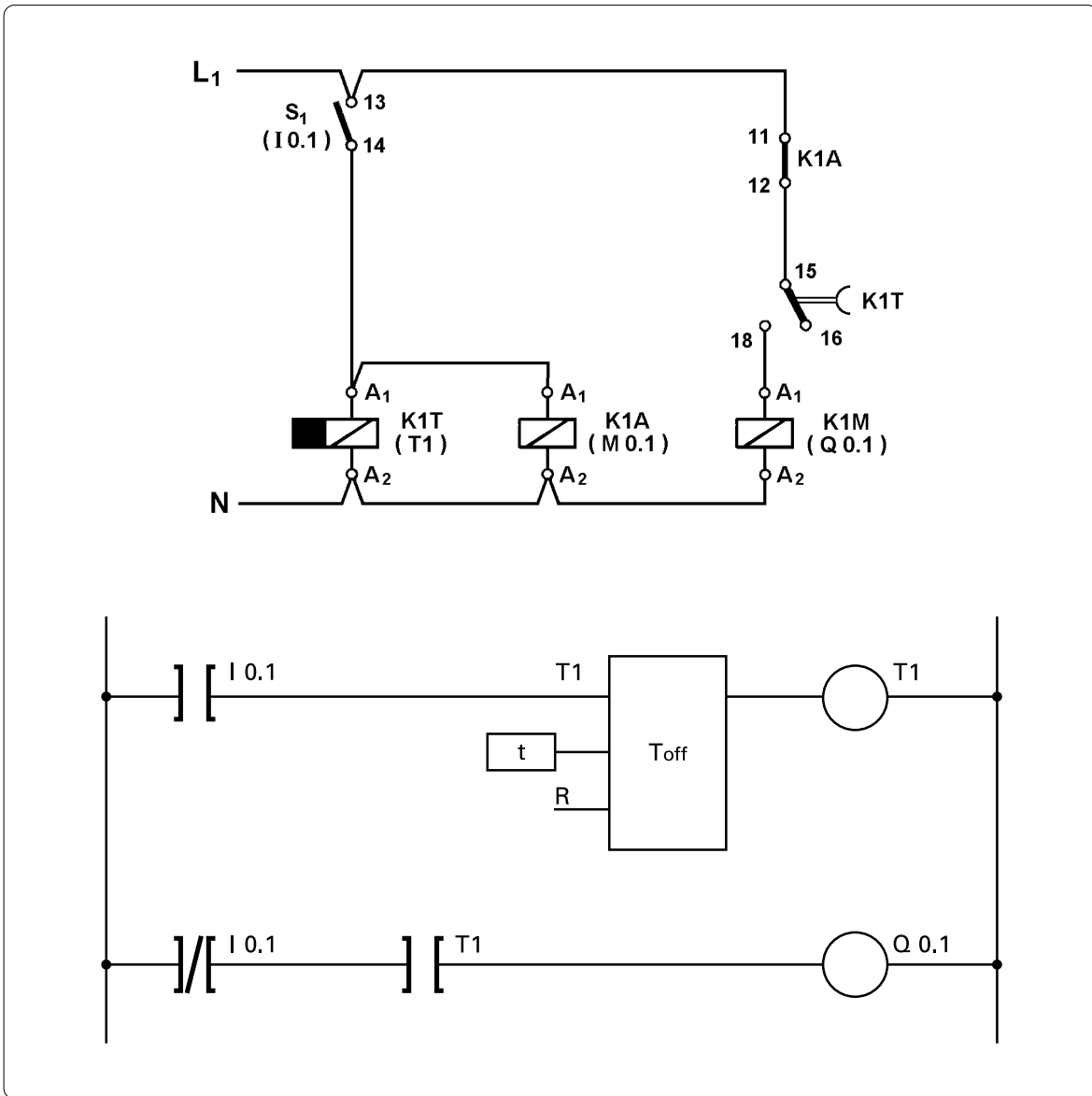


Παράδειγμα 5. Πρόγραμμα χρονικής λειτουργίας δημιουργίας παλμού ανεξάρτητου από τη χρονική διάρκεια του σήματος διέγερσης με τη χρησιμοποίηση του στοιχείου προγράμματος της χρονικής λειτουργίας καθυστέρησης στην ενεργοποίηση (delay on).

Σχήμα 66

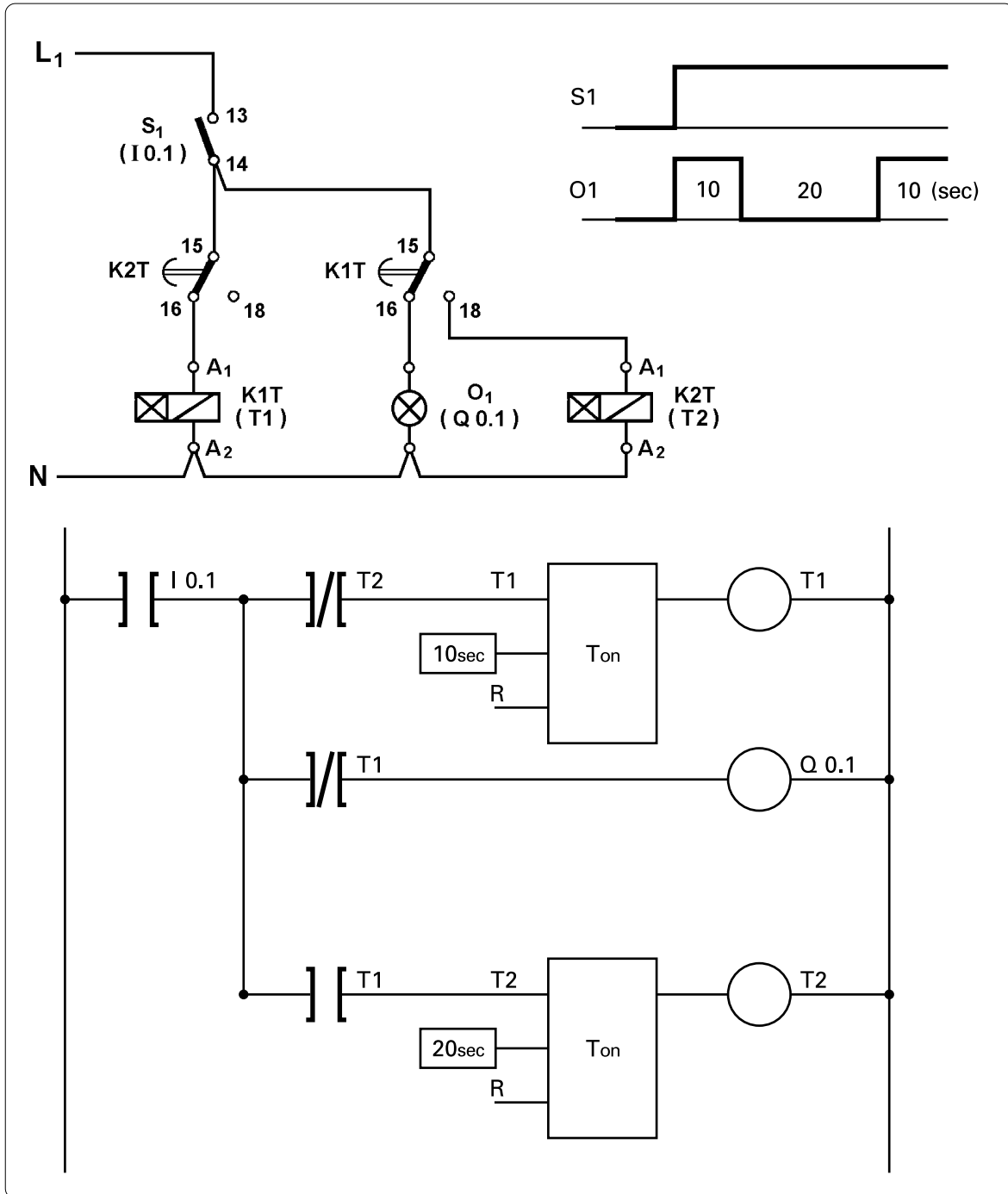


Παράδειγμα 6. Πρόγραμμα χρονικής λειτουργίας δημιουργίας παλμού κατά το άνοιγμα μιας ηλεκτρικής επαφής.



Σχήμα 67

Παράδειγμα 7. Πρόγραμμα δημιουργίας παλμοσειράς.

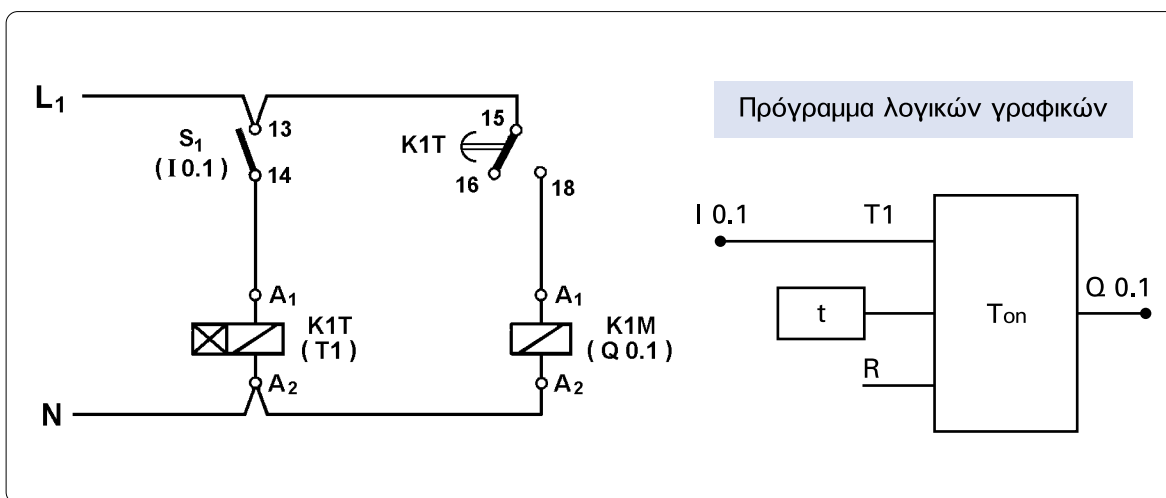


Σχήμα 68

9.3 Παραδείγματα προγραμμάτων με χρονικές λειτουργίες στη γλώσσα λογικών γραφικών.

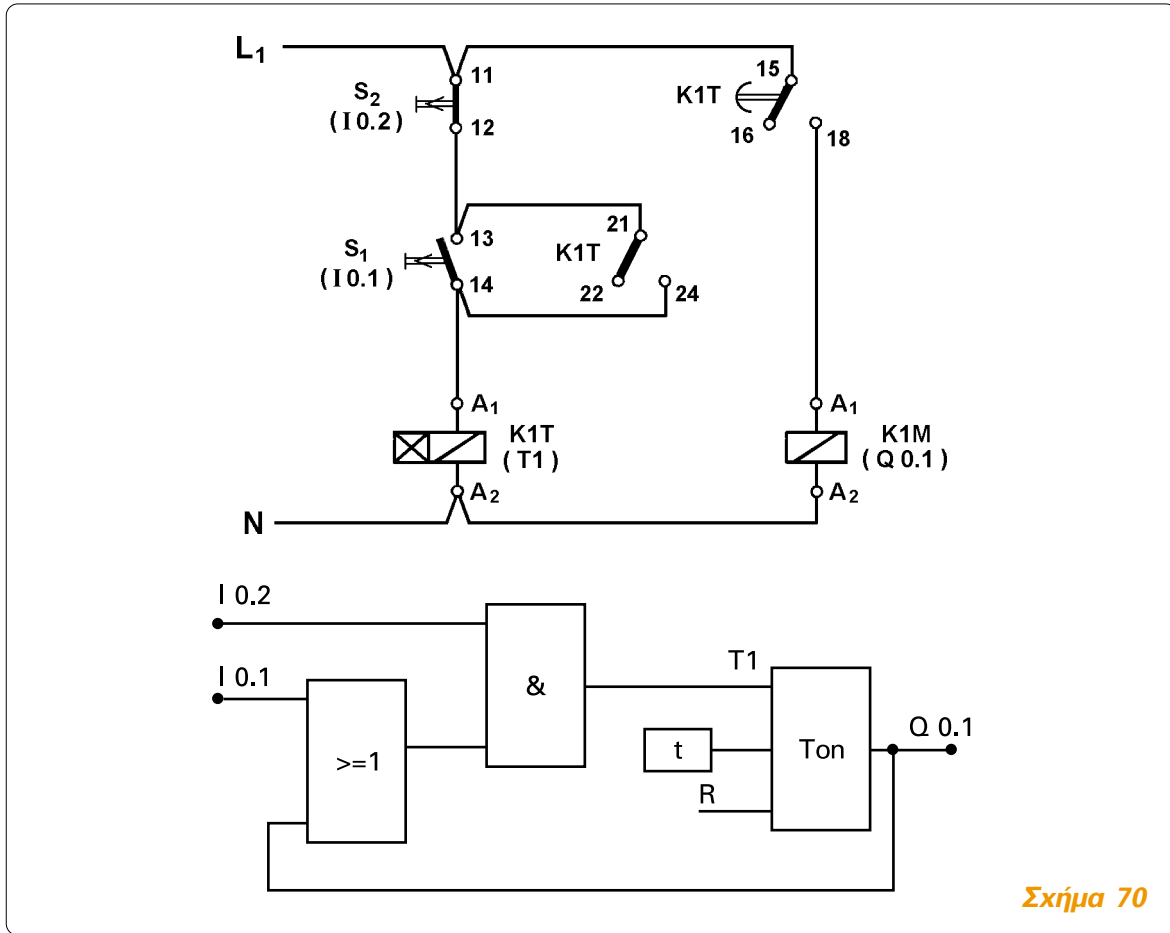
Το στοιχείο προγράμματος (σύμβολο), που χρησιμοποιούμε για τα χρονικά στη γλώσσα λογικών γραφικών, είναι το ίδιο με αυτό, που χρησιμοποιούμε και στη γλώσσα Ladder. Κατά τα άλλα ισχύουν όσα ακριβώς έχουμε πει μέχρι τώρα για τον προγραμματισμό στη γλώσσα λογικών γραφικών. Ας δούμε λοιπόν, πως θα γίνουν στη γλώσσα των λογικών γραφικών τα προγράμματα με τα χρονικά, που φτιάξαμε στη γλώσσα Ladder. Στα παραδείγματα δίνεται πάλι και το ηλεκτρολογικό κύκλωμα και δείχνεται με χρώματα η αντιστοιχία των στοιχείων του κυκλώματος με τις λογικές πύλες.

Παράδειγμα 1. Πρόγραμμα χρονικής λειτουργίας καθυστέρησης στην ενεργοποίηση (delay on).



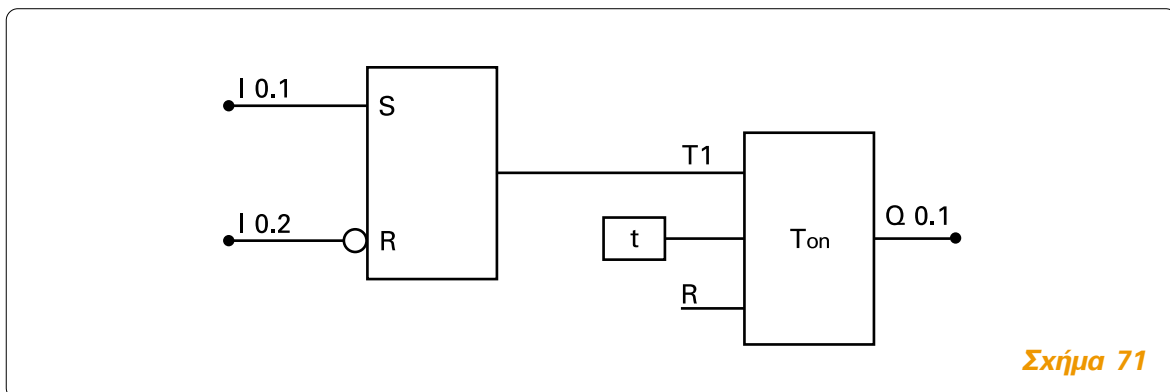
Σχήμα 69

Παράδειγμα 2. Πρόγραμμα χρονικής λειτουργίας καθυστέρησης στην ενεργοποίηση με αυτοσυγκράτηση (retentive delay on).



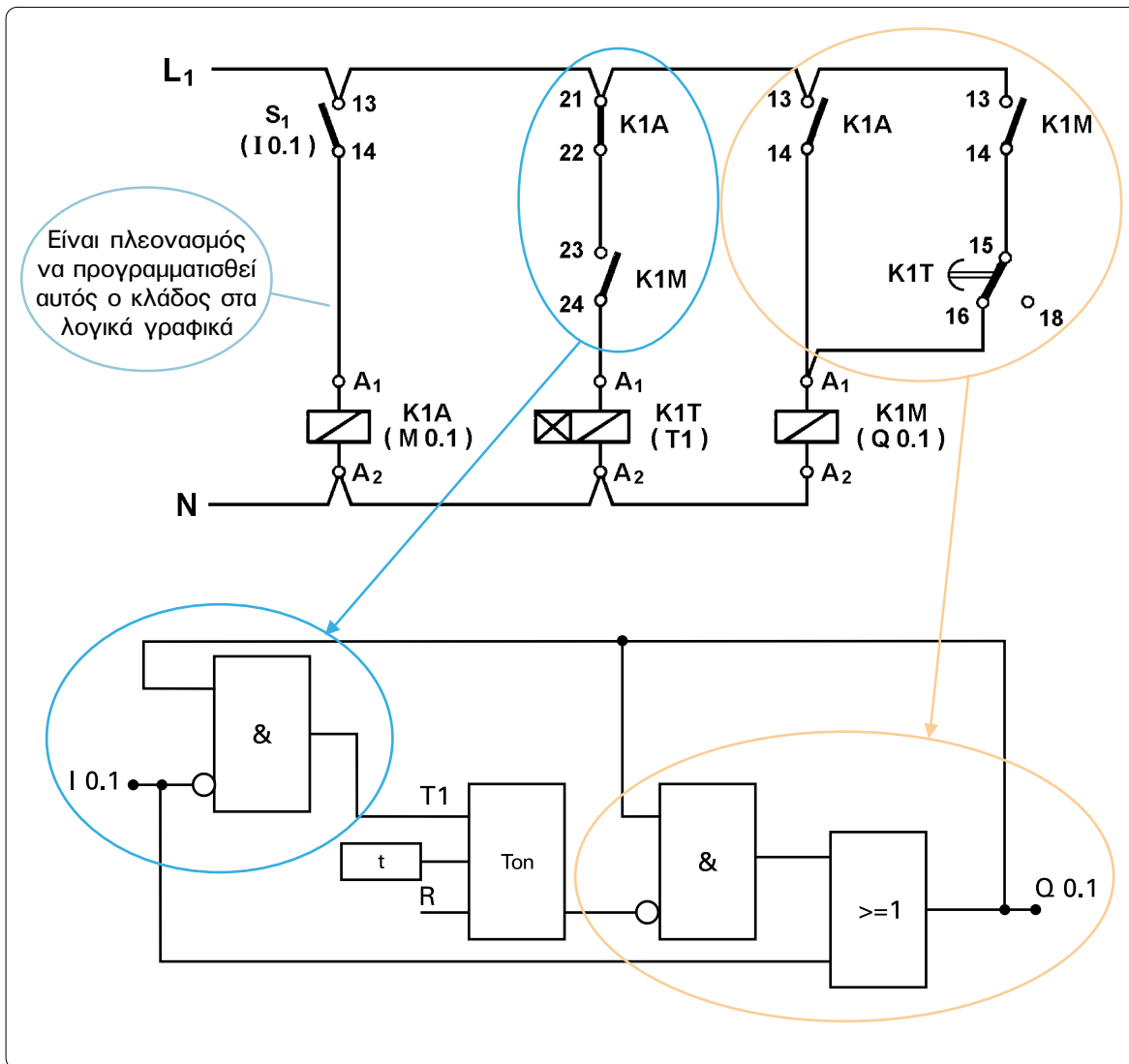
Σχήμα 70

Το ίδιο πρόγραμμα με τη χρήση του S-R flip-flop γίνεται:



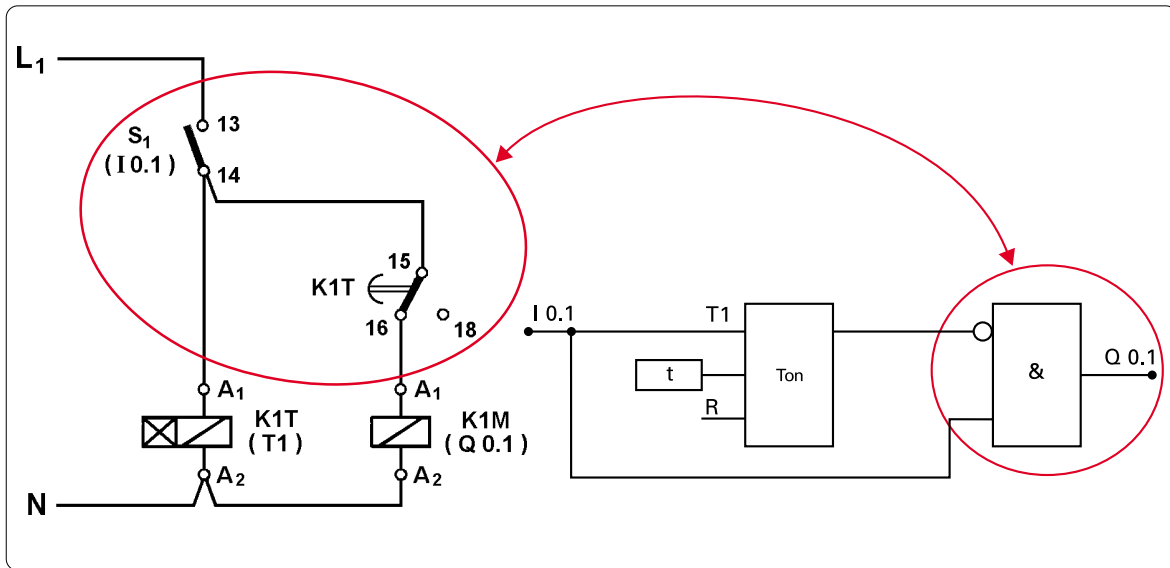
Σχήμα 71

Παράδειγμα 3. Πρόγραμμα χρονικής λειτουργίας καθυστέρησης στην απενεργοποίηση (delay off) με τη χρησιμοποίηση του στοιχείου προγράμματος της χρονικής λειτουργίας καθυστέρησης στην ενεργοποίηση (delay on).



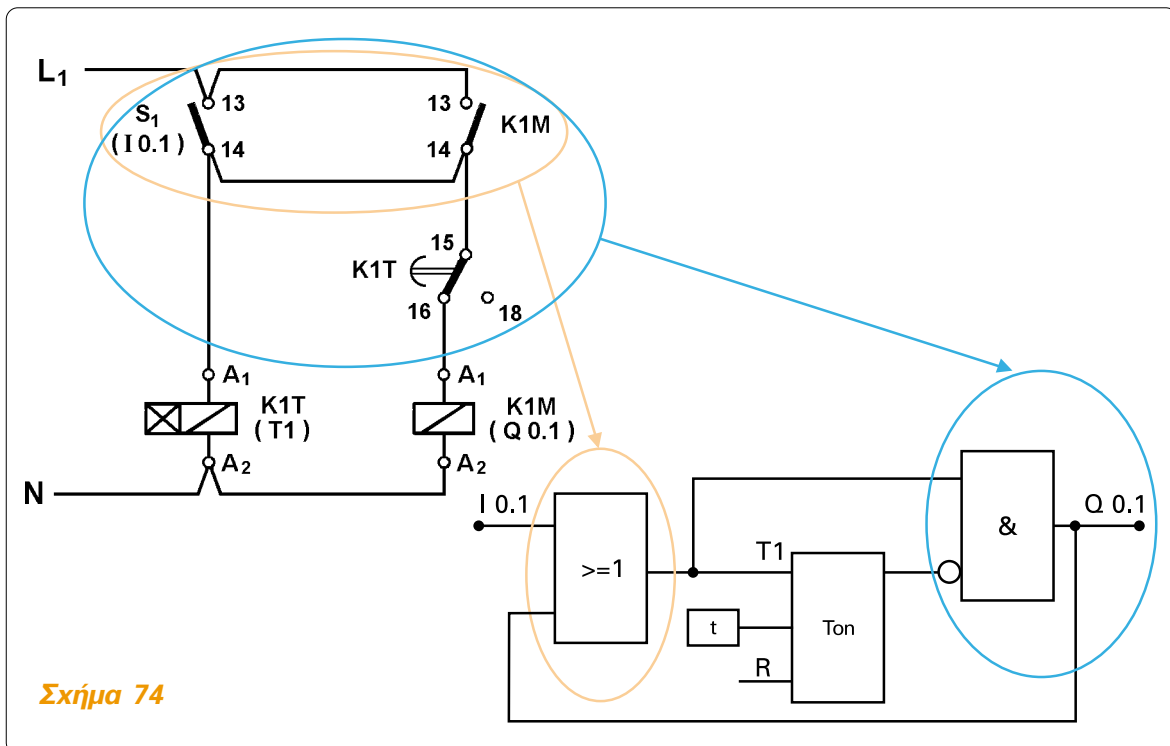
Σχήμα 72

Παράδειγμα 4. Πρόγραμμα χρονικής λειτουργίας δημιουργίας παλμού εξαρτώμενου και από τη χρονική διάρκεια του σήματος διέγερσης (one shot) με τη χρησιμοποίηση του στοιχείου προγράμματος της χρονικής λειτουργίας καθυστέρησης στην ενεργοποίηση (delay on).



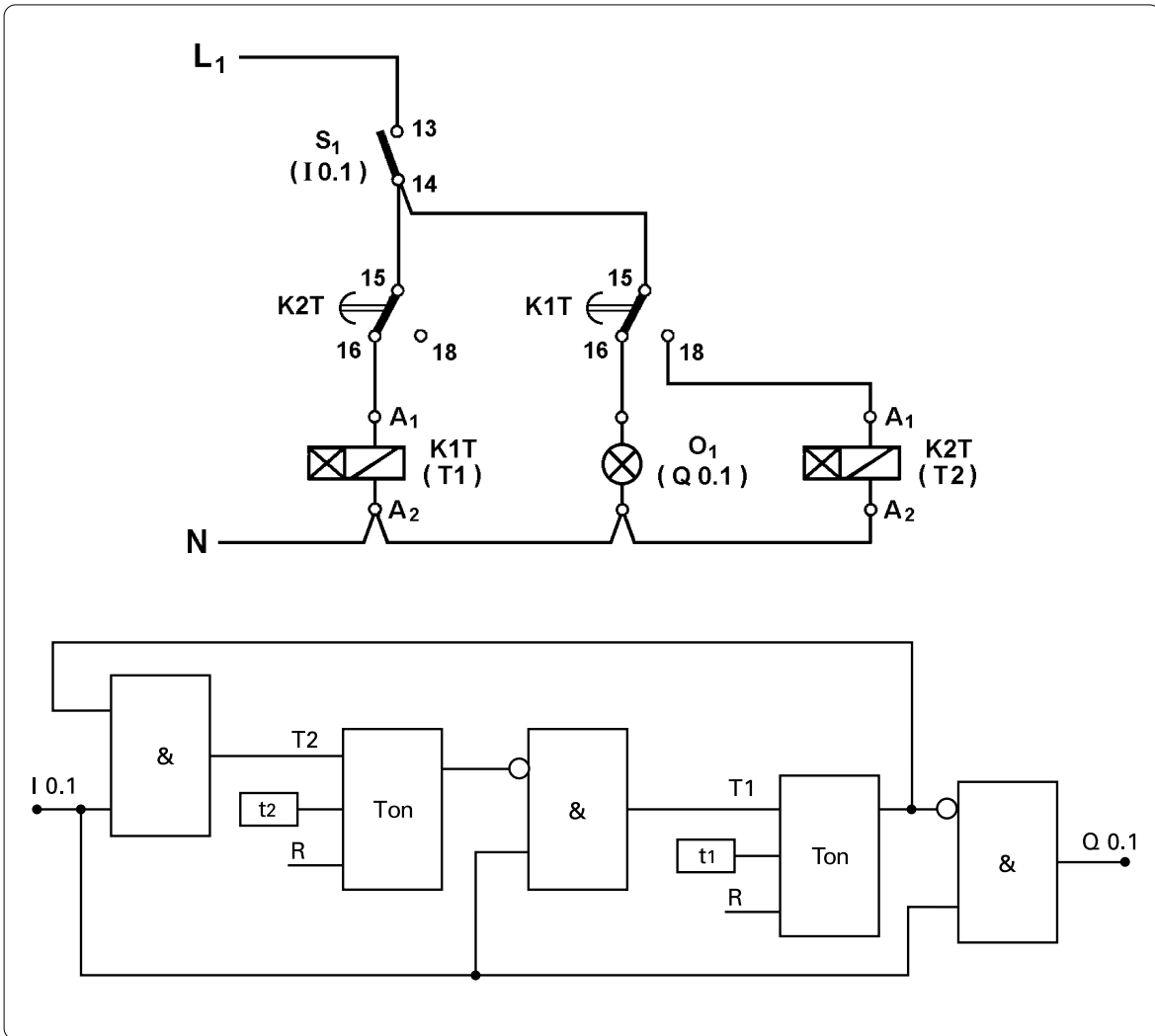
Σχήμα 73

Παράδειγμα 5. Πρόγραμμα χρονικής λειτουργίας δημιουργίας παλμού ανεξάρτητου από τη χρονική διάρκεια του σήματος διέγερσης με τη χρησιμοποίηση του στοιχείου προγράμματος της χρονικής λειτουργίας καθυστέρησης στην ενεργοποίηση (delay on).



Σχήμα 74

Παράδειγμα 6. Πρόγραμμα δημιουργίας παλμοσειράς.

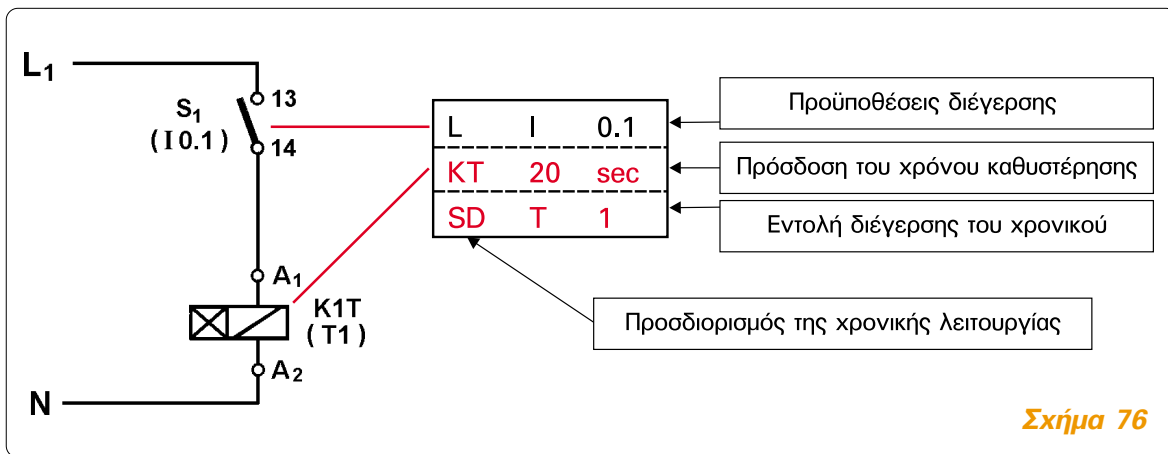


Σχήμα 75

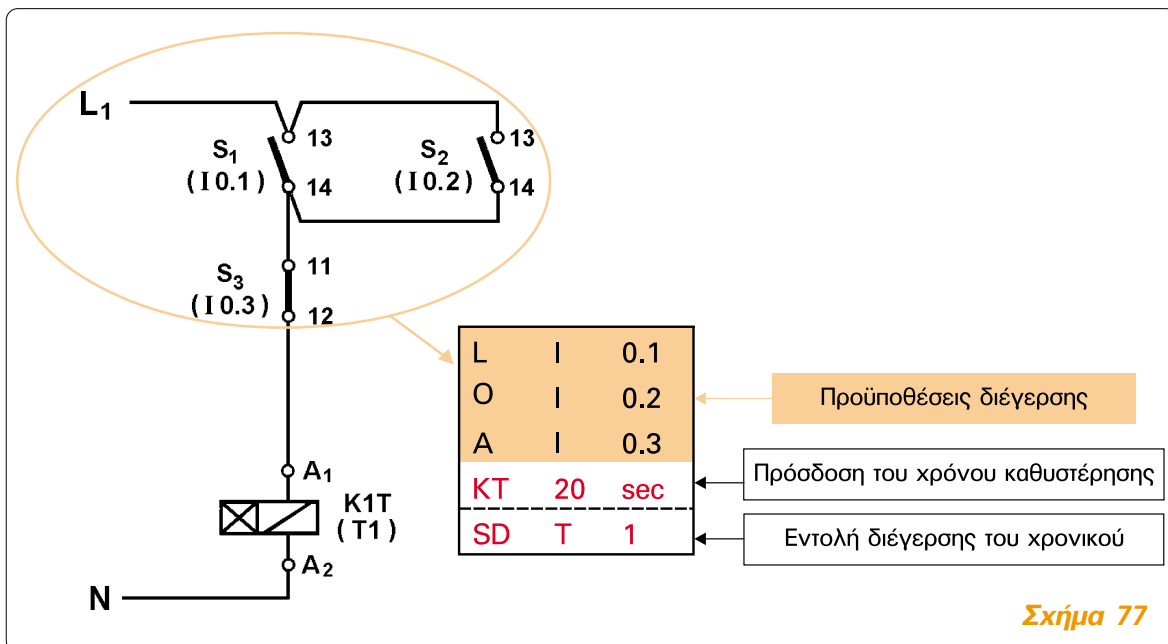
9.4 Παραδείγματα προγραμμάτων με χρονικές λειτουργίες σε γλώσσα λίστα εντολών.

Στη γλώσσα λίστα εντολών οι διαφορές στον προγραμματισμό χρονικών λειτουργιών μεταξύ των διαφόρων PLC είναι μεγαλύτερες απ' ό,τι στις γραφικές γλώσσες. Στη συνέχεια περιγράφουμε τις εντολές χρονικών λειτουργιών χωρίς να αναφερόμαστε σε κάποιο συγκεκριμένο PLC. Με μικρές παραλλαγές οι εντολές ισχύουν σε πολλά PLC.

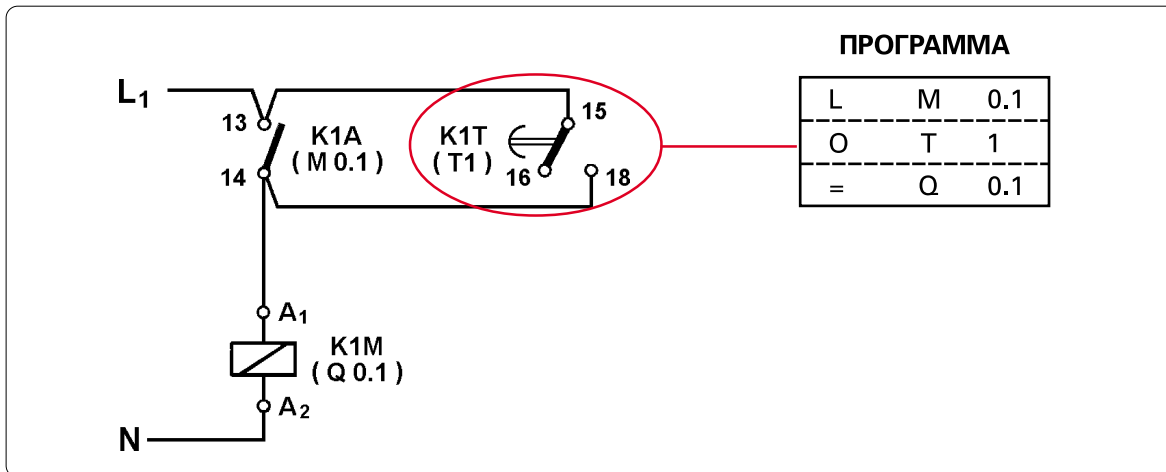
Στις εντολές προγράμματος μιας χρονικής λειτουργίας έχουμε τις προϋποθέσεις διέγερσης, την εντολή πρόσδοσης χρόνου και την εντολή διέγερσης όπως φαίνεται στο σχήμα 76.



Για τις προϋποθέσεις διέγερσης μπορεί να έχουμε ένα πολύπλοκο κλάδο όπως φαίνεται στο παράδειγμα του σχήματος 77.



Στις εντολές προγράμματος μιας χρονικής λειτουργίας μπορεί να περιλαμβάνεται και εντολή αποδιέγερσης της χρονικής λειτουργίας (Reset). Έχουμε ήδη δει το ρόλο που παίζει η εντολή **R** σε ένα χρονικό. Η εντολή Reset χρησιμοποιείται όπως ακριβώς η γνωστή μας εντολή Reset σε μια οποιαδήποτε παράμετρο.



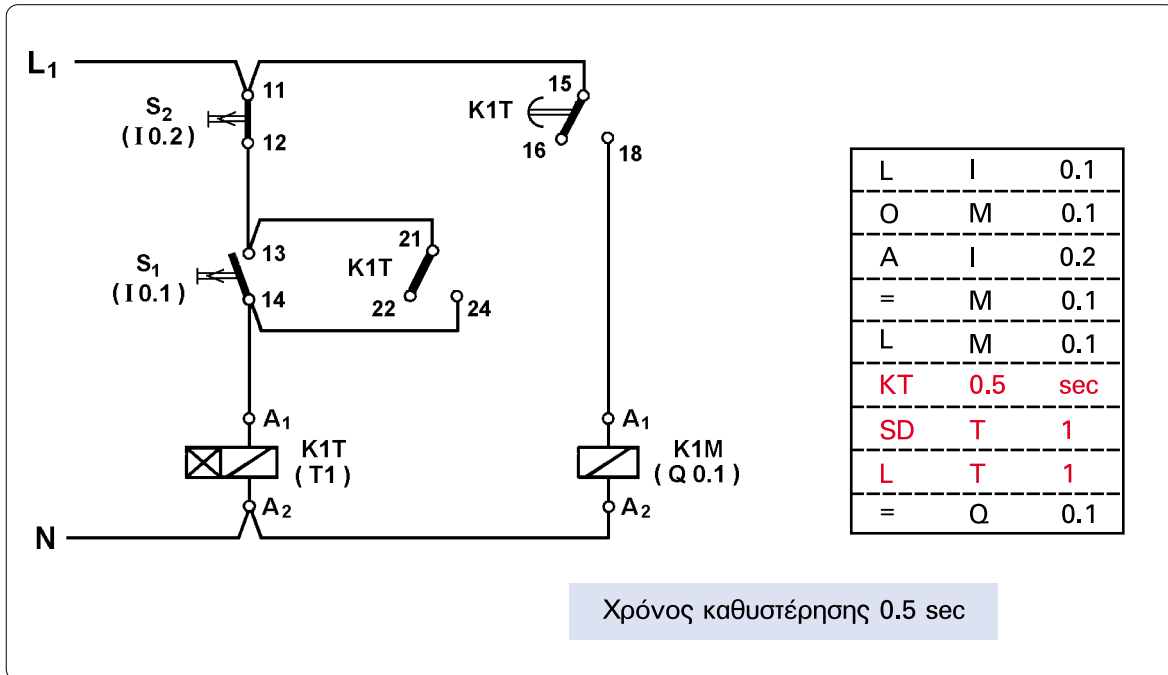
Σχήμα 78

Το αντίστοιχο της επαφής χρονικής λειτουργίας ενός χρονοηλεκτρονόμου είναι μια εντολή με παράμετρο T1 (σχήμα 78). Η παράμετρος T1 έχει λογική κατάσταση ("0" ή "1") που προσδιορίζεται από τη χρονική λειτουργία που καθορίζεται στην εντολή διέγερσης.

Στην περίπτωση που ένα PLC υποστηρίζει πολλών ειδών χρονικές λειτουργίες, αλλάζει ο χαρακτηρισμός της χρονικής λειτουργίας στην εντολή διέγερσης της χρονικής λειτουργίας. Αντί για χρονική λειτουργία SD έχουμε χρονική λειτουργία SS, SP κ.λπ. Στο βιβλίο αυτό με το χαρακτηρισμό SD θα εννοούμε χρονική λειτουργία με καθυστέρηση στην ενεργοποίηση (delay on).

Παραδείγματα προγραμμάτων με χρονικές λειτουργίες σε γλώσσα λίστα εντολών.

Παράδειγμα 1. Πρόγραμμα χρονικής λειτουργίας καθυστέρησης στην ενεργοποίηση με αυτοσυγκράτηση (retentive delay on).

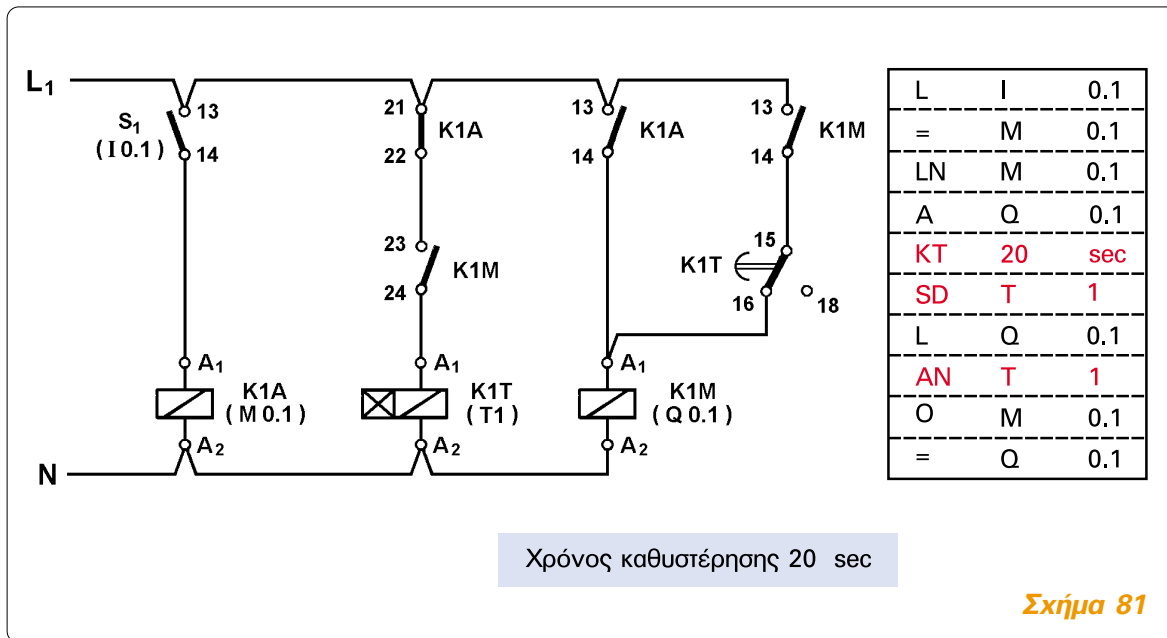


Σχήμα 79

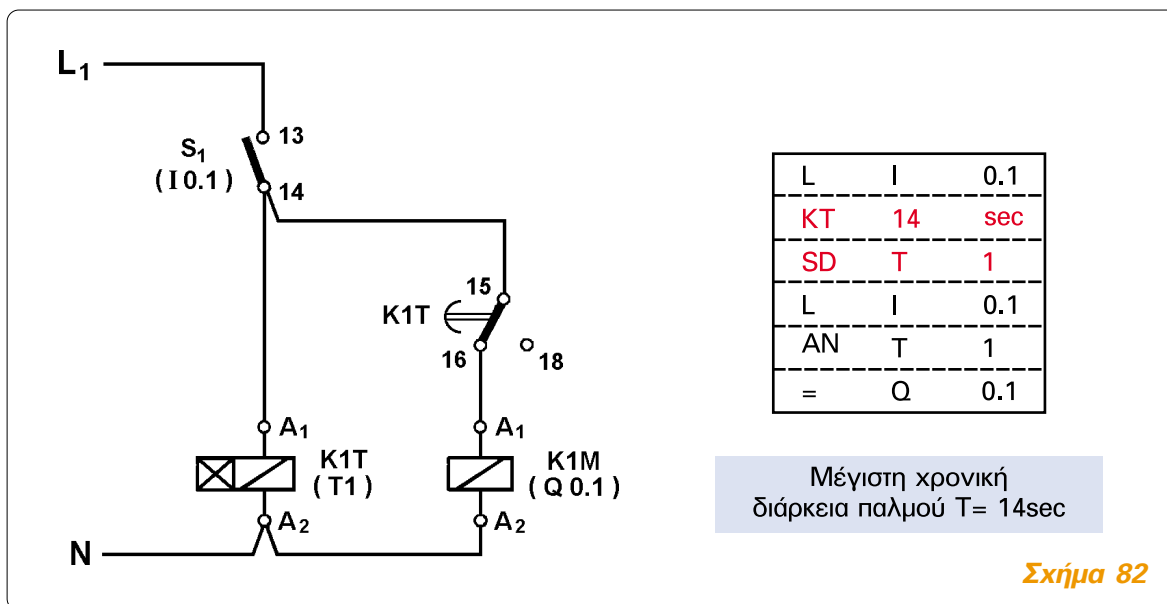
L	I	0.1
S	M	0.1
LN	I	0.2
R	M	0.1
L	M	0.1
KT	0.5	sec
SD	T	1
L	T	1
=	Q	0.1

Σχήμα 80

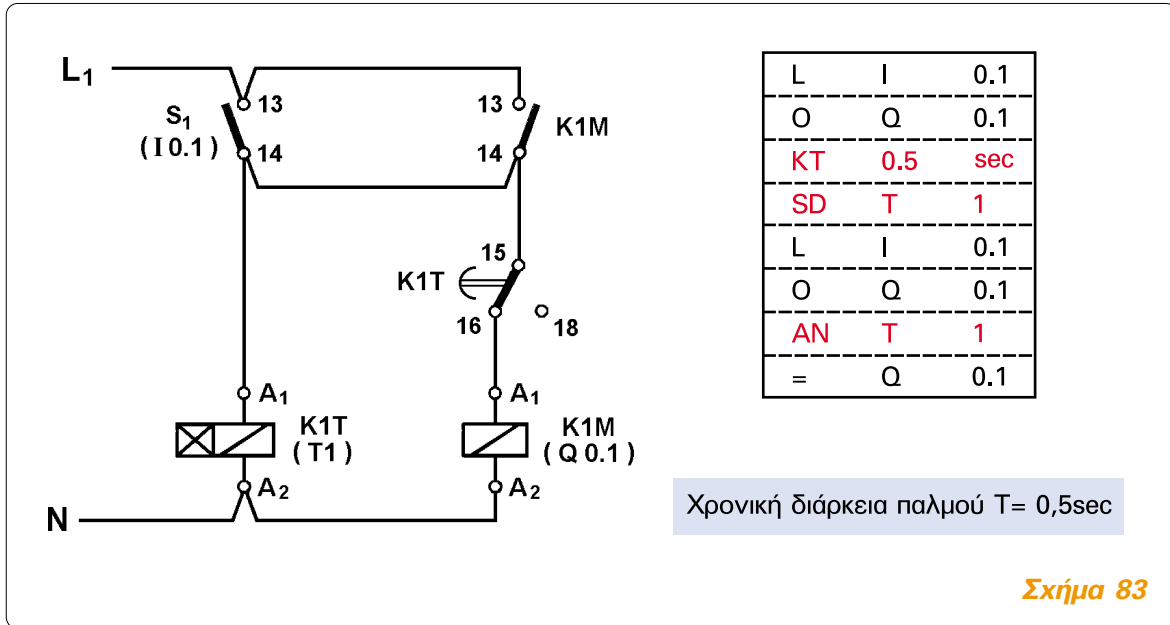
Παράδειγμα 2. Πρόγραμμα χρονικής λειτουργίας καθυστέρησης στην απενεργοποίηση (delay off) με τη χρησιμοποίηση του προγράμματος της χρονικής λειτουργίας καθυστέρησης στην ενεργοποίηση (delay on).



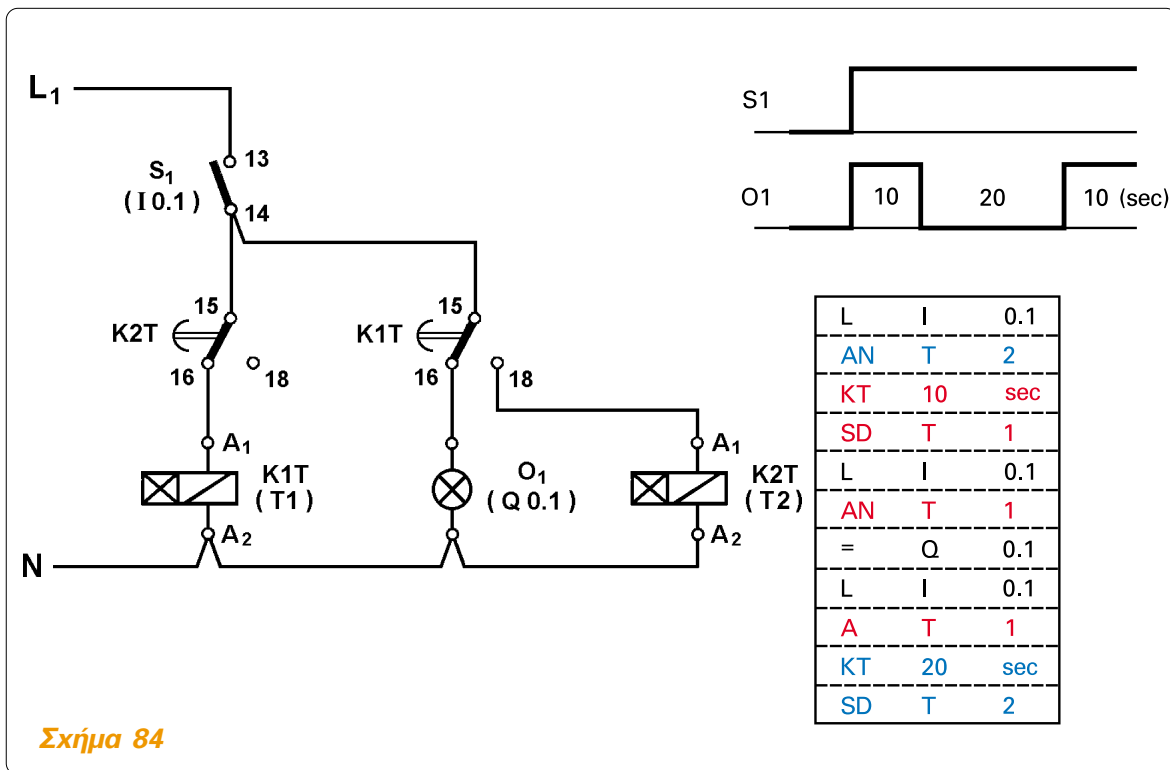
Παράδειγμα 3. Πρόγραμμα χρονικής λειτουργίας δημιουργίας παλμού εξαρτώμενου και από τη χρονική διάρκεια του σήματος διέγερσης (one shot) με τη χρησιμοποίηση του προγράμματος της χρονικής λειτουργίας καθυστέρησης στην ενεργοποίηση (delay on).



Παράδειγμα 4. Πρόγραμμα χρονικής λειτουργίας δημιουργίας παλμού ανεξάρτητου από τη χρονική διάρκεια του σήματος διέγερσης με χρησιμοποίηση του προγράμματος της χρονικής λειτουργίας καθυστέρησης στην ενεργοποίηση (delay on).



Παράδειγμα 5. Πρόγραμμα δημιουργίας παλμοσειράς.



10. Άλλες δυνατότητες των προγραμματιζόμενων λογικών ελεγκτών.

Τα PLC σήμερα έχουν και πολλές άλλες δυνατότητες. Οι δυνατότητες αυτές συνεχώς αυξάνουν, καθώς τα PLC εξελίσσονται με ταχύτατους ρυθμούς. Αναφέρουμε ενδεικτικά τις σημαντικότερες από αυτές τις δυνατότητες.

- Λειτουργία απαριθμητών. Οι απαριθμητές αποτελούν ακόμη 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 και το γραφικό σύμβολο και οι εντολές ποικίλουν όπως ακριβώς και στα χρονικά.

- Δυνατότητες πραγματικών ωρολογίων μέσω των οποίων 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. Για να μπορέσει κάποιος να καταλάβει και να αξιοποιήσει τις δυνατότητες αυτές πρέπει να έχει γνώσεις ψηφιακών ηλεκτρονικών και μικροϋπολογιστών.

- Αναλογικές εισοδοί-εξοδοί. Τα PLC ενώ αρχικά ήρθαν για να αντικαταστήσουν τους αυτοματισμούς καλωδιωμένης λογικής (αυτοματισμούς με ηλεκτρονόμους), οι δυνατότητές τους έχουν εξαπλωθεί με προοπτική να καλύψουν πλήρως και τα συστήματα αυτομάτου ελέγχου, όπως είναι αναλογικοί έλεγχοι θερμοκρασίας, πίεσης, στάθμης, στροφών κινητήρων κ.λπ. 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 προαναφέραμε. Η δυνατότητα επεξεργασίας αναλογικών σημάτων έχει δώσει άλλη προοπτική εξέλιξης 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 των 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 Βιομηχανικό Δίκτυο Αυτοματισμού (Computer Automatic Network, CAN). Τα PLC βρίσκονται στη βάση κάθε βιομηχανικού δικτύου αυτοματισμού.

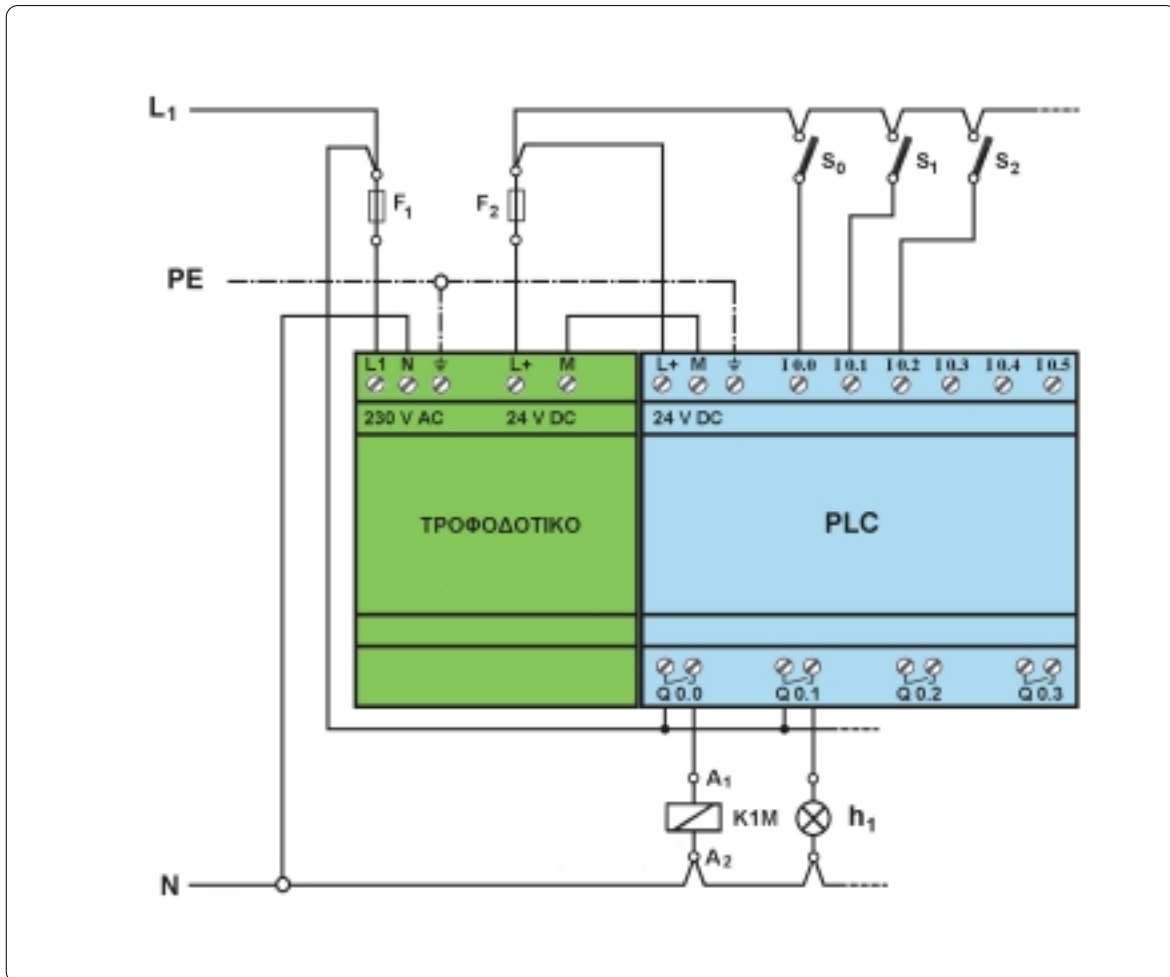
11. Εφαρμογή των προγραμματιζόμενων λογικών ελεγκτών στον έλεγχο λειτουργίας ηλεκτροκινητήρων.

Είμαστε έτοιμοι τώρα να δούμε μια πρώτη εφαρμογή στον έλεγχο λειτουργίας ηλεκτροκινητήρων με χρήση PLC. Πριν προχωρήσουμε στην εφαρμογή θα δούμε λίγα πράγματα για την καλωδίωση των PLC.

Καλωδίωση ενός PLC.

Η καλωδίωση ενός PLC εξαρτάται από το συγκεκριμένο μοντέλο του PLC, αλλά σε γενικές γραμμές ισχύουν τα εξής:

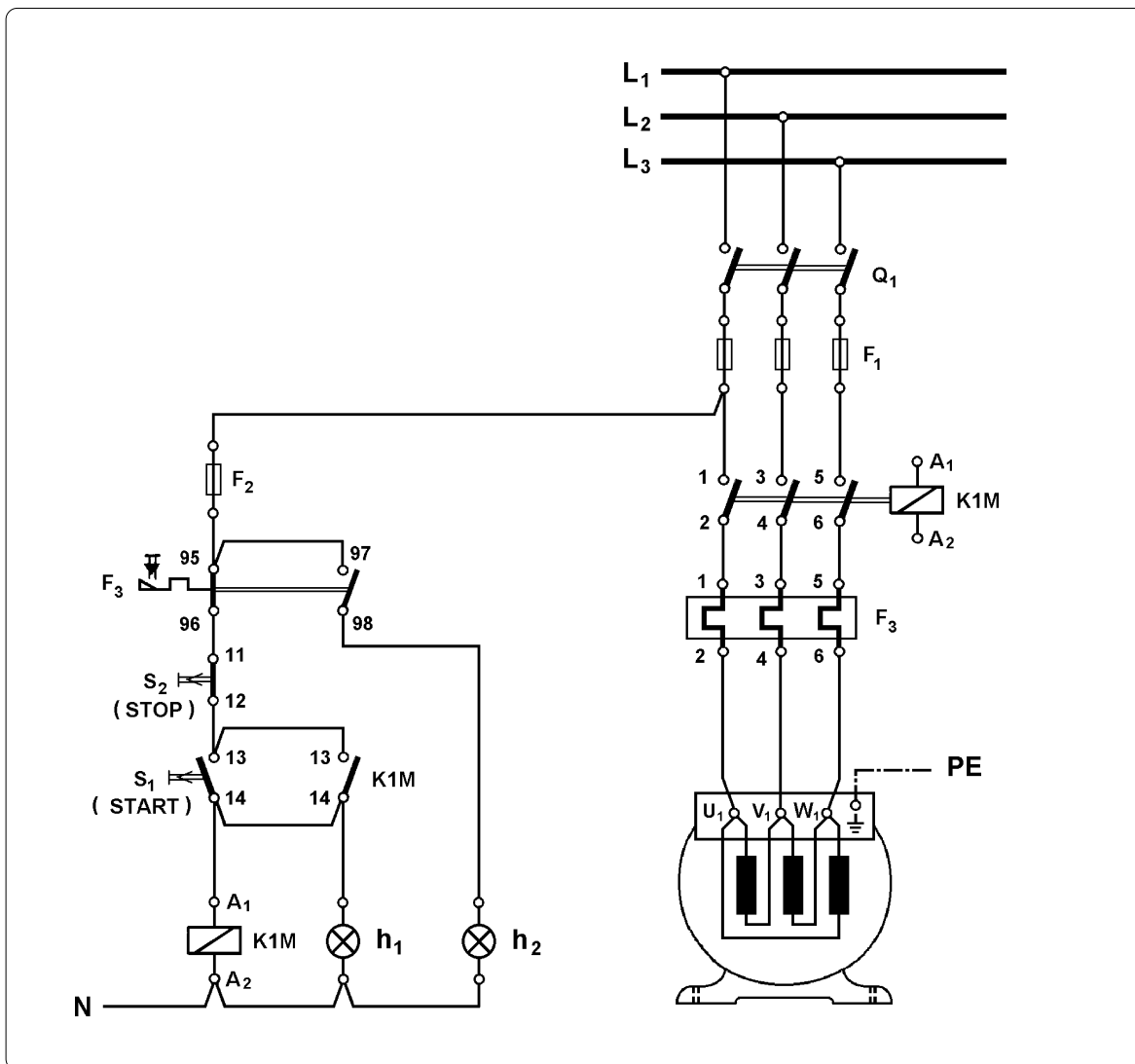
- Οι εισοδοι τροφοδοτούνται συνήθως με τάση 24V DC. Η τάση αυτή παρέχεται από το ίδιο το τροφοδοτικό του PLC.
- Για τις εξόδους συνηθέστερο είναι να επιλέγονται μονάδες εξόδου με τάση 220V AC. Στην περίπτωση αυτή οι έξοδοι λαμβάνουν τροφοδοσία εξωτερική.



Σχήμα 85: Παράδειγμα καλωδίωσης ενός PLC.

Εφαρμογή: Έλεγχος της λειτουργίας τριφασικού ηλεκτροκινητήρα βραχυκυκλωμένου δρομέα με απ' ευθείας εκκίνηση.

Πρόκειται για την απλή εκκίνηση τριφασικού ηλεκτροκινητήρα βραχυκυκλωμένου δρομέα με μπουτόνς START, STOP. Το πλήρες ηλεκτρολογικό σχέδιο του αυτοματισμού δίνεται στο σχήμα 86.



Σχήμα 86: Απλός αυτόματος διακόπτης για τη λειτουργία τριφασικού ηλεκτροκινητήρα βραχυκυκλωμένου δρομέα.

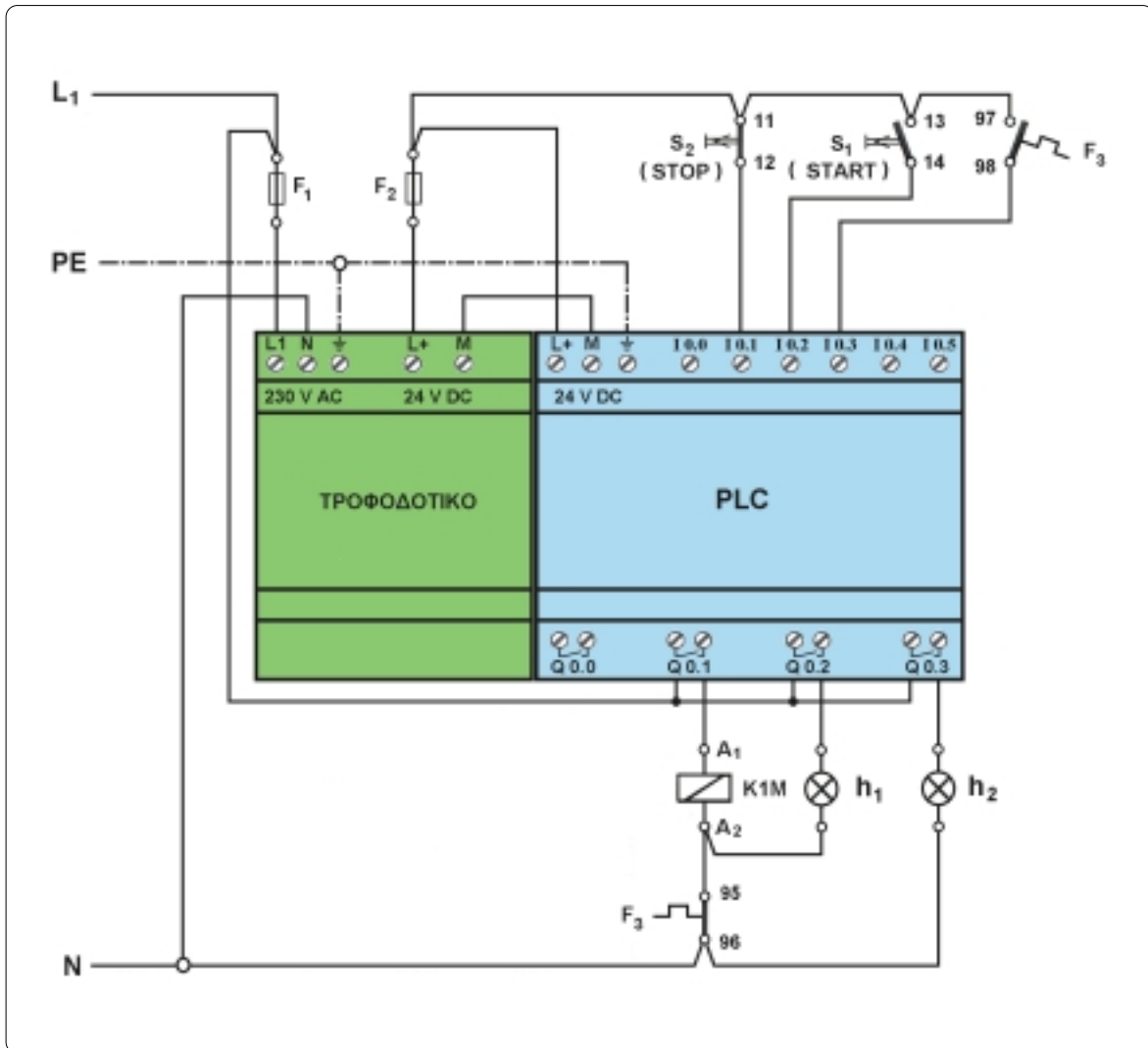
Στο σχήμα 87 δίνεται ο πίνακας των εισόδων και των εξόδων του PLC. Η μη χρησιμοποίηση της εισόδου I 0.0 και της εξόδου Q 0.0 του PLC δεν έγινε για κάποιο συγκεκριμένο λόγο.

Είσοδοι		Έξοδοι	
Μπουτόν STOP	I 0.1	Πηνίο ηλεκτρονόμου ισχύος (K1M)	Q 0.1
Μπουτόν START	I 0.2	Λυχνία ένδειξης λειτουργίας h1	Q 0.2
Επαφή 97-98 του θερμικού	I 0.3	Λυχνία ένδειξης βλάβης h2	Q 0.3

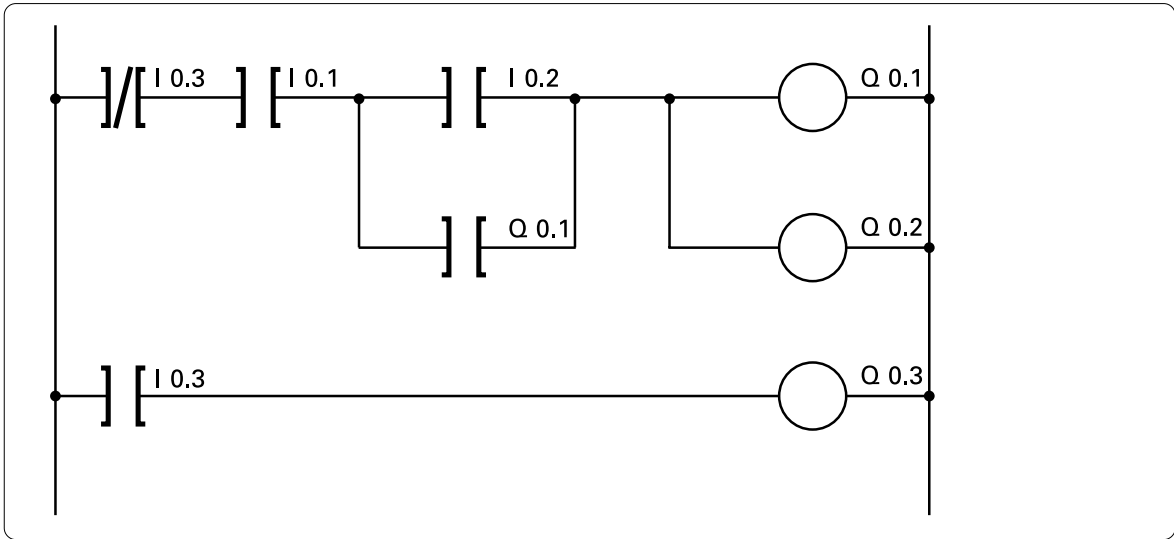
Σχήμα 87: Πίνακας των εισόδων και των εξόδων του PLC.

Στο σχήμα 88 δίνεται η καλωδίωση του PLC της εφαρμογής. Όπως φαίνεται σ' αυτό το σχήμα, στο εξωτερικό κύκλωμα των εξόδων του PLC έχει συνδεθεί η "κανονικά κλειστή" επαφή 95-96 του θερμικού που προστατεύει τον ηλεκτροκινητήρα από υπερφόρτιση. Η συνδεσμολογία αυτή της επαφής 95-96 του θερμικού είναι υποχρεωτική και γίνεται για να εξασφαλισθεί καλύτερη προστασία στον ηλεκτροκινητήρα. Με τη συνδεσμολογία αυτή, όταν ενεργοποιηθεί το θερμικό ύστερα από υπερφόρτιση του ηλεκτροκινητήρα, θα διακοπεί η τροφοδοσία του πηνίου του ηλεκτρονόμου ισχύος K1M και θα διακοπεί η τροφοδοσία του ηλεκτροκινητήρα έστω και αν υπάρχει πρόβλημα στο υλικό (hardware) του PLC (π.χ. να έχει βραχυκυκλωθεί η έξοδος Q 0.1 του PLC και να έχει συνεχώς τάση ο ακροδέκτης της εξόδου Q 0.1).

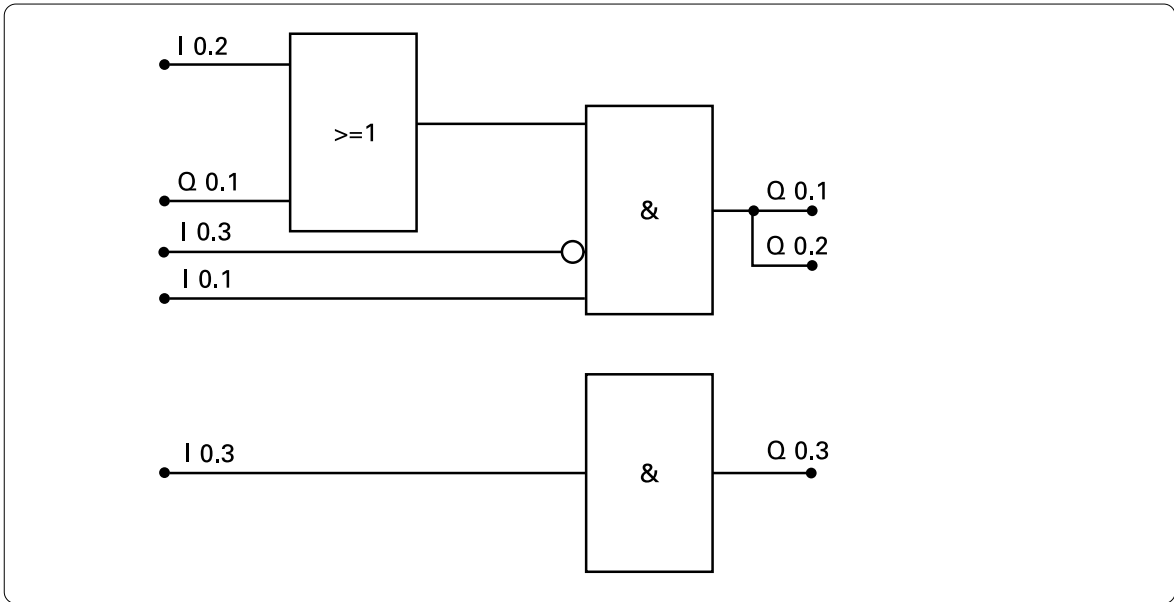
Στα σχήματα 89, 90 και 91 δίνεται το πρόγραμμα που ελέγχει τη λειτουργία του ηλεκτροκινητήρα στις γλώσσες Ladder, λογικά γραφικά και λίστα εντολών αντίστοιχα.



Σχήμα 88: Καλωδίωση του PLC της εφαρμογής.



Σχήμα 89: Το πρόγραμμα σε γλώσσα Ladder.

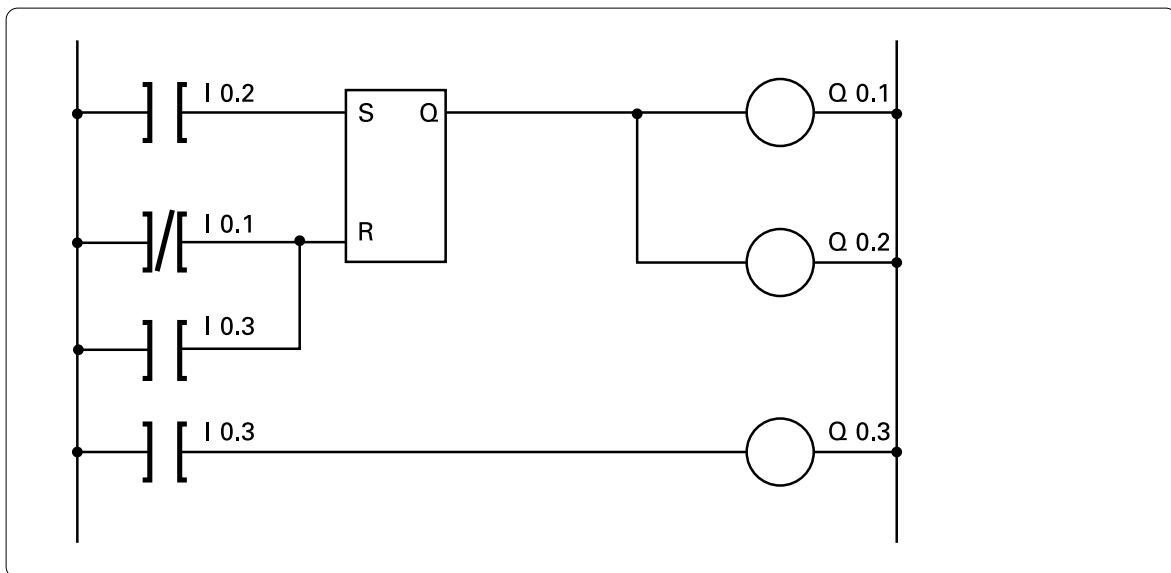


Σχήμα 90: Το πρόγραμμα σε γλώσσα λογικών γραφικών.

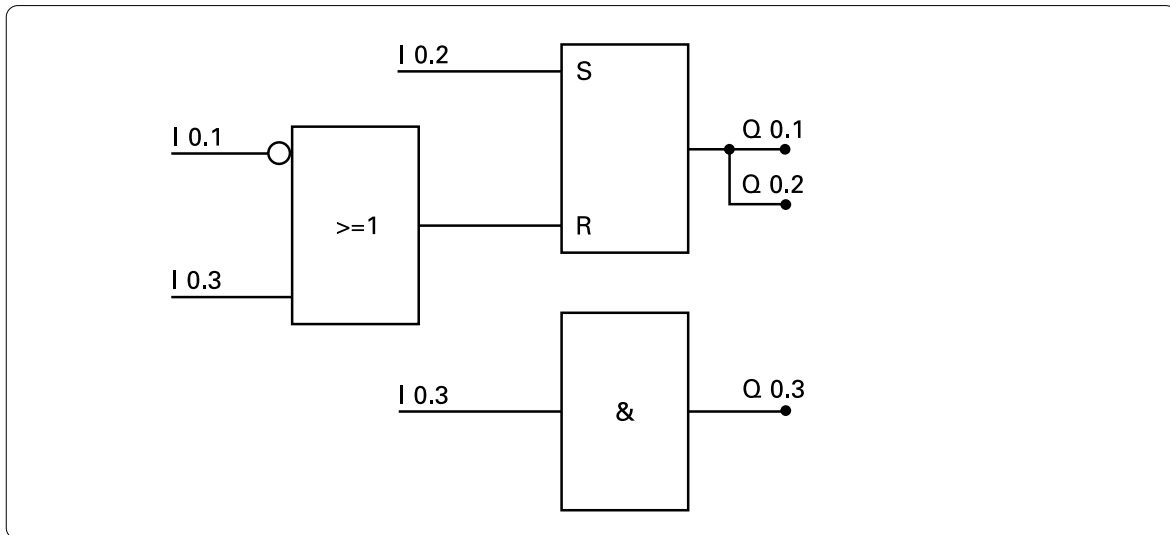
L	I	0.2
O	Q	0.1
AN	I	0.3
A	I	0.1
=	Q	0.1
=	Q	0.2
A	I	0.3
=	Q	0.3

Σχήμα 91: Το πρόγραμμα σε γλώσσα λίστα εντολών.

Το πρόγραμμα της εφαρμογής με χρήση των εντολών SET και RESET στις γλώσσες Ladder, λογικά γραφικά και λίστα εντολών δίνεται στα σχήματα 92, 93 και 94 αντίστοιχα.



Σχήμα 92: Το πρόγραμμα σε Ladder με χρήση των εντολών SET και RESET (S-R flip-flop).



Σχήμα 93: Το πρόγραμμα σε γλώσσα λογικών γραφικών με χρήση των εντολών SET και RESET (S-R flip-flop).

L	I	0.2
S	Q	0.1
LN	I	0.1
O	I	0.3
R	Q	0.1
L	Q	0.1
=	Q	0.2
A	I	0.3
=	Q	0.3

Σχήμα 94: Το πρόγραμμα σε λίστα εντολών με χρήση των εντολών SET και RESET.

12. Ανακεφαλαίωση.

- ☞ Τα PLC είναι ειδικές συσκευές, οι οποίες αντικαθιστούν τον κλασικό αυτοματισμό της καλωδιωμένης λογικής. Τα πλεονεκτήματά τους είναι πολλά. Τα σπουδαιότερα είναι ότι είναι ευέλικτα, επεκτάσιμα και έχουν μικρό κόστος.
- ☞ Ένα PLC αποτελείται από τη κεντρική μονάδα επεξεργασίας, τη μονάδα τροφοδοσίας, τις μονάδες εισόδων / εξόδων και τα πλαίσια στήριξης των μονάδων. Στην αγορά τα PLC κυκλοφορούν κυρίως στις μορφές modular PLC και συμπαγή. Στα συμπαγή PLC όλα τα παραπάνω στοιχεία βρίσκονται ενσωματωμένα σε μία ενιαία συσκευή.
- ☞ Η κεντρική μονάδα επεξεργασίας ενός PLC περιέχει το μικροεπεξεργαστή, τις μνήμες RAM, ROM και EEPROM. Η λειτουργία του στηρίζεται στη συνεχή εκτέλεση του προγράμματος σε μια ατέρμονα διαδικασία. Ο χρόνος, που διαρκεί ένας κύκλος λειτουργίας, αποτελεί ένδειξη της ταχύτητας ενός PLC και ονομάζεται χρόνος κύκλου.
- ☞ Οι γλώσσες προγραμματισμού των PLC δεν είναι τυποποιημένες, παρ' όλα αυτά τρεις κατηγορίες γλωσσών έχουν επικρατήσει στην αγορά. Αυτές είναι η γλώσσα Ladder, η γλώσσα λίστα εντολών και η γλώσσα λογικών διαγραμμάτων.
- ☞ Για να προχωρήσουμε στον προγραμματισμό ενός PLC χρειάζεται να γνωρίζουμε ποια είναι τα προγραμματιστικά στοιχεία του PLC και πως ονομάζονται στο συγκεκριμένο PLC. Αυτά τα στοιχεία είναι οι είσοδοι, οι έξοδοι, οι βοηθητικές μνήμες, και επιπλέον τα χρονικά, οι απαριθμητές και οι ειδικές συναρτήσεις.
- ☞ Το πρόγραμμα σε γλώσσα λίστα εντολών αποτελείται από μια σειρά εντολών. Κάθε εντολή αποτελεί μια γραμμή προγράμματος. Οι εντολές κατανέμονται σε ομάδες εντολών. Κάθε ομάδα εντολών αντιστοιχεί σε μια λογική πύλη. Κάθε εντολή αποτελείται από δύο μέρη. Το πρώτο μέρος καθορίζει την ενέργεια την οποία θα εκτελέσει το PLC, ενώ το δεύτερο μέρος καθορίζει την παράμετρο, στην οποία αναφέρεται (είσοδο, έξοδο, βοηθητική μνήμη κ.λπ.). Βασικές εντολές της γλώσσας λίστα εντολών είναι:
 - > L (Load)
 - > = (ίσον)
 - > A (AND)
 - > O (OR)
 - > N (NOT)
 - > S (Set)
 - > R (Reset)

☞ Η γλώσσα Ladder είναι μια γραφική γλώσσα. Στην ουσία ένα πρόγραμμα ladder είναι περίπου το ίδιο το ηλεκτρολογικό σχέδιο αυτοματισμού. Στη γλώσσα Ladder χρησιμοποιείται ο αμερικανικός συμβολισμός των στοιχείων των ηλεκτρονόμων και οι κλάδοι του ηλεκτρολογικού σχεδίου σχεδιάζονται οριζόντια και όχι κατακόρυφα.

☞ Η γλώσσα λογικών γραφικών είναι επίσης γραφική. Στη γλώσσα αυτή το πρόγραμμα είναι ίδιο με το λογικό κύκλωμα που αντιστοιχεί στο ηλεκτρολογικό σχέδιο του αυτοματισμού.

13. Ερωτήσεις - Ασκήσεις.

1. Αναφέρετε τα πλεονεκτήματα που προκύπτουν από τη χρήση των PLC στη βιομηχανία σε σύγκριση με τον κλασικό αυτοματισμό. Ποιο νομίζετε ότι είναι το σπουδαιότερο πλεονέκτημα και γιατί;
2. Τα βήματα ανάπτυξης ενός κλασικού αυτοματισμού καλωδιωμένης λογικής είναι τα παρακάτω:
 - 1) Περιγραφή του αυτοματισμού.
 - 2) Ανάπτυξη του λειτουργικού σχεδίου του αυτοματισμού.
 - 3) Ανάπτυξη του σχεδίου εφαρμογής του πίνακα (σχέδιο καλωδίωσης).
 - 4) Κατασκευή του πίνακα της εγκατάστασης.
 - 5) Εγκατάσταση και σύνδεση στις κλέμες του πίνακα των αισθητήρων, που δίνουν τις πληροφορίες (εντολές), και των συσκευών (αποδεκτών), που εκτελούν τις εργασίες.
 - 6) Δοκιμή λειτουργίας της εγκατάστασης.
 - 7) Πλήρης λειτουργία του αυτοματισμού.

Αναφέρετε ποια από τα παραπάνω βήματα αλλάζουν, όταν σχεδιάζουμε ένα αυτοματισμό με PLC.

3. Ποιες είναι οι διαφορές στα εξής είδη μνήμης: RAM, ROM, EEPROM.
4. Γράψτε τον όρο που αντιστοιχεί στην επεξήγηση που δίνεται παρακάτω

	Μια μονάδα του PLC, η οποία λαμβάνει τα εξωτερικά σήματα από τους αισθητήρες.
	Ο χρόνος που χρειάζεται το PLC να εκτελέσει ένα πλήρη κύκλο λειτουργίας, από ένα διάβασμα των εισόδων έως το επόμενο διάβασμα των εισόδων
	Ο χρόνος κύκλου ενός προγράμματος που περιλαμβάνει 1 KByte δυαδικές εντολές
	Το είδος της μνήμης όπου αποθηκεύονται τα προγράμματα του PLC και η οποία δεν χρειάζεται μπαταρία για να διατηρήσει τα στοιχεία της αμετάβλητα
	Η ειδική περιοχή της μνήμης όπου αποθηκεύονται οι τιμές των εισόδων που "διαβάζει" το PLC κατά την λειτουργία του.

5. Καταγράψτε τα πλεονεκτήματα και τα μειονεκτήματα της κάθε μιας από τις παρακάτω γλώσσες προγραμματισμού:

1. Ladder

2. Λίστα εντολών

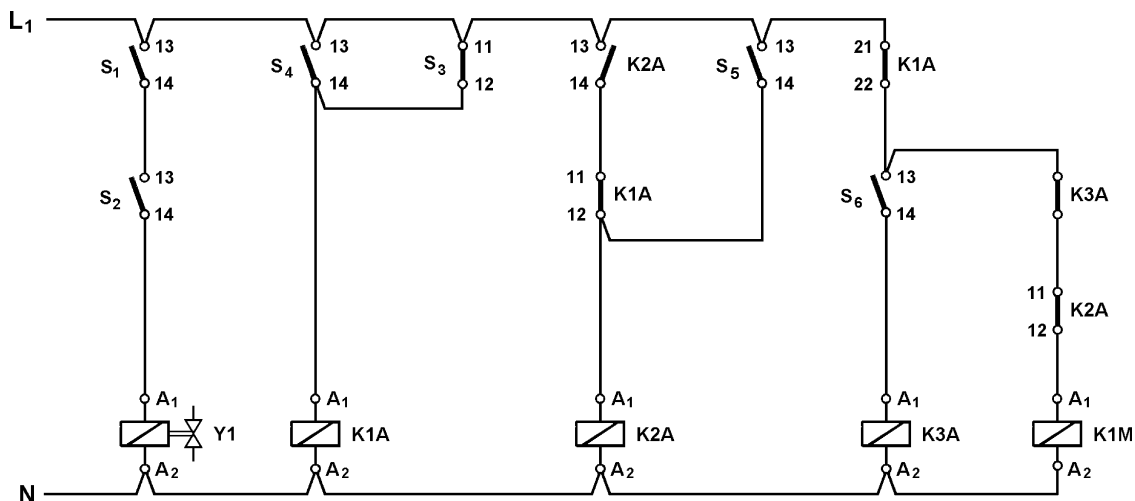
3. Λογικά γραφικά

6. Δώστε το πρόγραμμα σε ladder, σε λίστα εντολών και σε γλώσσα λογικών γραφικών για κάθε μια από τις παρακάτω πύλες:

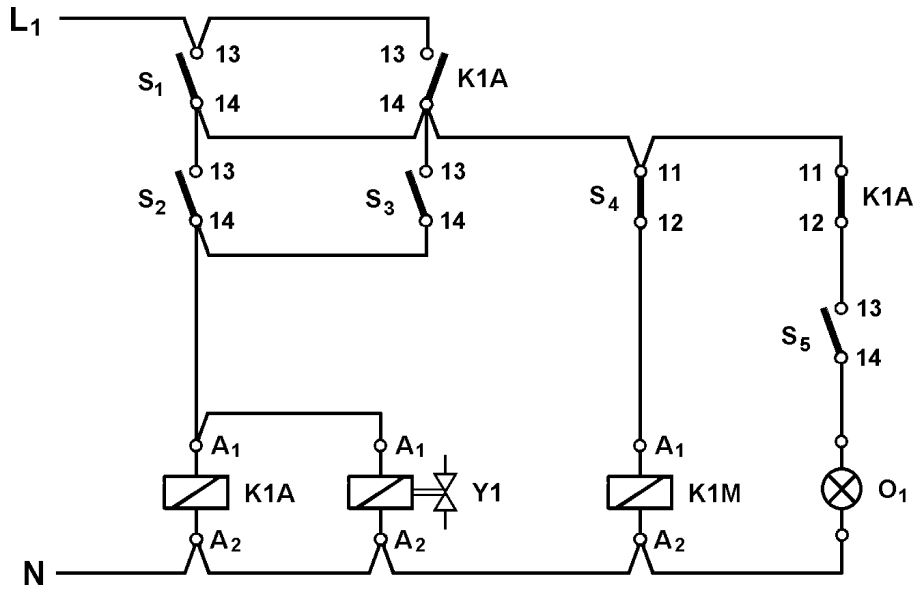
- > AND τριών εισόδων
- > OR τεσσάρων εισόδων
- > NAND δύο εισόδων
- > NOT
- > NOR πέντε εισόδων
- > XOR

7. Δώστε στις γλώσσες ladder, λίστα εντολών και λογικών γραφικών τα προγράμματα που αντιστοιχούν στα παρακάτω ηλεκτρολογικά κυκλώματα:

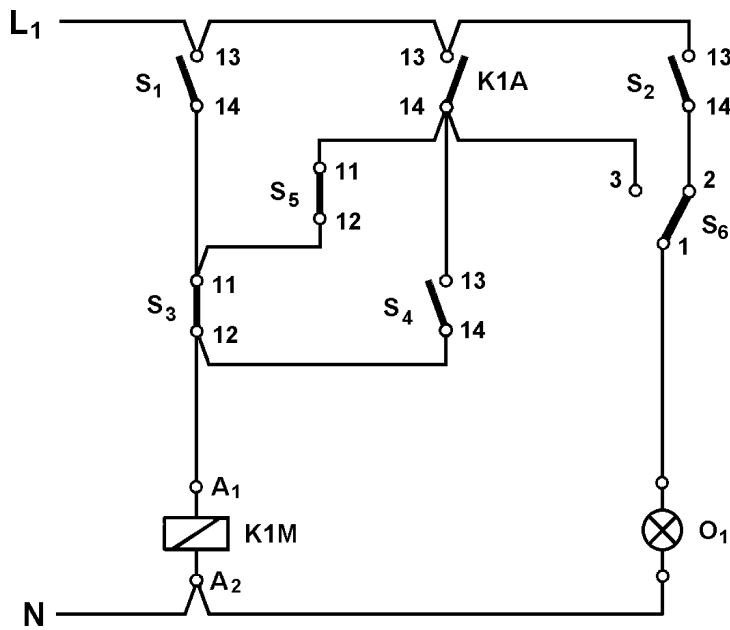
α)



β)



γ)



8. Σχεδιάστε τα βέλτιστα λογικά κυκλώματα που προκύπτουν από τους παρακάτω πίνακες αληθείας και γράψτε τα προγράμματα που αντιστοιχούν σε γλώσσα λίστα εντολών.

α)

I1	I2	I3	O1
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

β)

I1	I2	I3	O1
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

9. Δώστε το πρόγραμμα σε γλώσσα ladder και γλώσσα λίστα εντολών ενός κυκλώματος αυτοματισμού που παράγει τις παρακάτω κυματομορφές:

