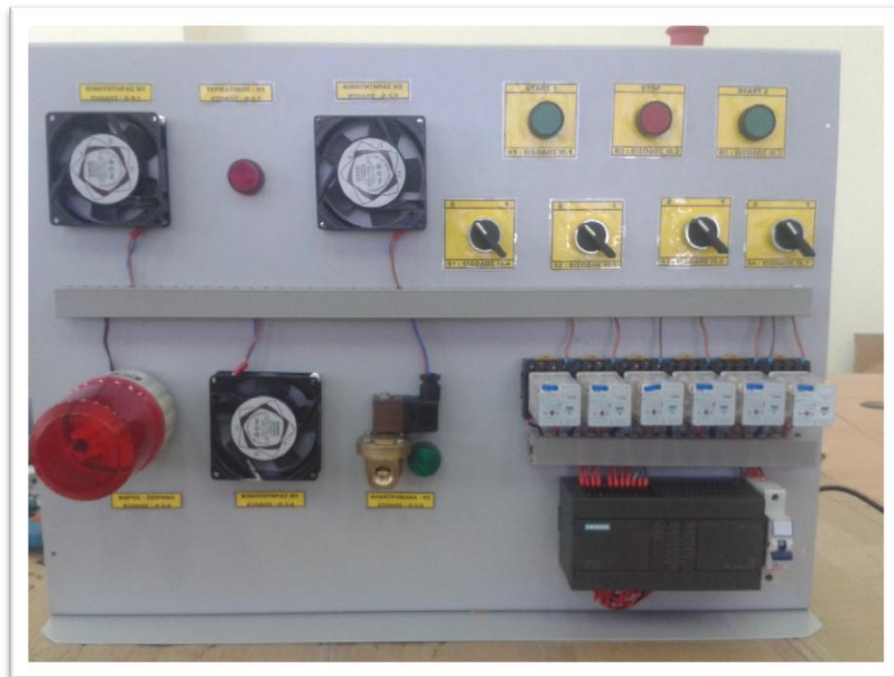

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ**

Π Τ Υ Χ Ι Α Κ Η Ε Ρ Γ Α Σ Ι Α

**Θέμα : Σχεδίαση και υλοποίηση εκπαιδευτικού
αναπτύγματος με τη χρήση του PLC S7-200,
για την προσομοίωση εφαρμογών
αυτοματισμού**



**Όνοματεπώνυμο
ΑΕΜ**

**: Ιωακείμ Μαμάτας
: 3203**

Επιβλέπων

: Ιωάννης Καλόμοιρος, Επίκουρος Καθηγητής

ΣΕΡΡΕΣ, 2015

Υπεύθυνη Δήλωση

Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Πληροφορικής & Επικοινωνιών του Τ.Ε.Ι. Σερρών.

Περίληψη

Σκοπός της παρούσας πτυχιακής εργασίας είναι η εισαγωγή και η εκμάθηση των σπουδαστών του τμήματος Πληροφορικής του ΤΕΙ Κεντρικής Μακεδονίας, σε βασικές έννοιες αυτοματισμού με τη χρήση των PLC. Αρχικά γίνεται η μελέτη των κύριων βαθμίδων των PLC γενικά και στη συνέχεια ακολουθεί η εξειδικευμένη περιγραφή για τον ελεγκτή S7-200 που χορηγήθηκε από το τμήμα και χρησιμοποιήθηκε στην εργασία. Η εκμάθηση των PLC ολοκληρώνεται με τη χρήση της εκπαιδευτικής πλατφόρμας που σχεδιάστηκε και κατασκευάστηκε για τις ανάγκες της πτυχιακής εργασίας, η οποία αν και έχει εκπαιδευτικό χαρακτήρα τηρεί προδιαγραφές που χρησιμοποιούνται στη βιομηχανία. Ο χρήστης καλείται μέσω των εφαρμογών – ασκήσεων που έχουν συγγραφεί, να υλοποιήσει τα δικά του προγράμματα και μέσω της διάταξης να ελέγξει την ορθότητα των προγραμμάτων καθώς και να κατανοήσει τις βασικές έννοιες του προγραμματιζόμενου αυτοματισμού.

Θεματική ενότητα : Βιομηχανικοί αυτοματισμοί

Λέξεις κλειδιά : PLC, Προγραμματιζόμενοι Λογικοί Ελεγκτές, Εκπαιδευτικό ανάπτυγμα

Περιεχόμενα	
Υπεύθυνη Δήλωση	2
Περίληψη	3
Εισαγωγή.....	6
ΚΕΦΑΛΑΙΟ 1 – ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΑΥΤΟΜΑΤΙΣΜΟ ΚΑΙ ΣΤΑ PLC	7
1.1 Εισαγωγή στον αυτοματισμό	7
1.1.1 Κλαστικός αυτοματισμός.....	8
1.1.2 Αυτοματισμός με PLC	9
1.2 Σύγκριση των αυτοματισμών	9
1.2.1 Στάδια υλοποίησης αυτοματισμών	9
1.2.2 Πλεονεκτήματα - Μειονεκτήματα	10
1.3 Εφαρμογές των PLC.....	11
1.4 Τύποι PLC	11
1.5 Δομή PLC	12
1.6 Αρχή λειτουργίας των PLC	15
1.6.1 Χρόνος κύκλου	16
1.7 Ονοματολογία - Διευθυνσιοδότηση	16
ΚΕΦΑΛΑΙΟ 2 – ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ PLC.....	20
2.1 Γλώσσες προγραμματισμού	20
2.1.1 Η γλώσσα LADDER - LAD	20
2.1.2 Η γλώσσα Statement List – STL	21
2.1.3 Η γλώσσα Function Block (FDB).....	21
2.2 Η γλώσσα LADDER	22
2.2.1 Εντολές λογικής bit (bit logic).....	24
2.2.2 Εντολές σύγκρισης (COMPARE).....	27
2.2.3 Εντολές χρονικών - μετρητών	27
2.3 Συσκευές προγραμματισμού	28
ΚΕΦΑΛΑΙΟ 3 – ΤΑ PLC ΤΗΣ ΕΤΑΙΡΕΙΑΣ SIEMENS	29
3.1 Η οικογένεια Simatic - S7	29
3.1.1 Simatic S7 - 200	29
3.1.2 Simatic S7 - 300.....	30
3.1.3 Simatic S7 - 400.....	30
3.2 PLC S7-200	32
3.2.1 Χαρακτηρίστηκα του PLC με τη CPU-212	32
3.2.2 Το εξωτερικό του PLC.....	32

3.2.3	Το εσωτερικό του PLC	33
3.3	Χαρακτηριστικά του S7 – 200	35
3.3.1	Ονοματολογία - Διευθυνσιοδότηση	35
3.3.2	Συμβολικός προγραμματισμός	39
3.4	Εργαλεία προγραμματισμού για το S7 - 200	40
3.4.1	Το περιβάλλον Step 7 / MicroWin	41
3.4.2	Ρυθμίσεις επικοινωνίας	42
3.4.3	Δημιουργία Project	43
3.4.4	Έλεγχος προγράμματος με τον Program Status	47
ΚΕΦΑΛΑΙΟ 4 – Η ΕΡΓΑΣΤΗΡΙΑΚΗ ΔΙΑΤΑΞΗ ΤΗΣ ΠΤΥΧΙΑΚΗΣ		48
4.1	Σκοπός της κατασκευής	48
4.2	Στάδια υλοποίησης	48
4.2.1	Επιλογή υλικών	48
4.2.2	Βάση τοποθέτησης υλικών – Λαμαρίνα	52
4.2.3	Κατεργασία πρόσοψης και τοποθέτηση των υλικών	53
4.2.4	Καλωδίωση	54
4.3	Η κατασκευή ολοκληρωμένη	54
4.3.1	Είσοδοι και έξοδοι του PLC	55
ΚΕΦΑΛΑΙΟ 5 – ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΕΦΑΡΜΟΓΕΣ		56
5.1	Έλεγχος κινητήρων	56
5.1.1	Έλεγχος κινητήρα M1	56
5.1.2	Έλεγχος κινητήρων M1 και M2	58
5.1.3	Έλεγχος κινητήρων M1 και M2	60
5.2	Χρονικά - Timers	62
5.3	Απαριθμητές - Counters	64
5.4	Έλεγχος μεταφορικών ταινιών	67
5.5	Χώρος στάθμευσης 5 θέσεων	70
Μελλοντικές επεκτάσεις		73
Βιβλιογραφία		74

Εισαγωγή

Ένα από τα πιο σημαντικά πεδία της επιστήμης και της τεχνολογίας, αποτελούν αδιαμφισβήτητα τα συστήματα αυτομάτου ελέγχου. Παρατηρώντας κανείς γύρω του θα διαπιστώσει ότι ολοένα και περισσότερα από τα επιτεύγματα που κατασκευάζει σήμερα ο άνθρωπος σχετίζονται άμεσα με τον αυτοματισμό, τόσο στις καθημερινές συσκευές που χρησιμοποιεί στη ζωή του όσο και στο εργασιακό του περιβάλλον. Βασικό χαρακτηριστικό, ενός συστήματος αυτοματισμού είναι ότι μπορεί να λειτουργεί χωρίς να χρειάζεται ανθρώπινη επιτήρηση ή παρέμβαση και να λειτουργεί με τέτοιο τρόπο, ώστε να επιτευχθεί κάποιο επιθυμητό αποτέλεσμα με προκαθορισμένη ακρίβεια. Ένα απλό και κλασσικό παράδειγμα είναι αυτό του θερμοστάτη τον οποίο χρησιμοποιούμε για να διατηρήσουμε τη θερμοκρασία ενός χώρου σταθερή σε μια επιθυμητή τιμή που έχει καθορίσει ο άνθρωπος.

Πέρα όμως από την εξέλιξη των συσκευών της καθημερινότητας μας, εδώ και αρκετά χρόνια τα συστήματα αυτοματισμού βρίσκουν εφαρμογή σε πολύ μεγάλο ποσοστό σε εργασιακά περιβάλλοντα και διεργασίες όπως οι βιομηχανίες, απαλλάσσοντας από τη μία τον άνθρωπο από εργασίες επικίνδυνες και ιδιαίτερα κοπιαστικές και από τη άλλη του δίνει νέο ρόλο και μορφή εργασίας. Αποτέλεσμα αυτού είναι να έχουν βελτιωθεί οι χρόνοι παραγωγής ενός προϊόντος καθώς και η ποιότητα του μιας και η χρήση μηχανημάτων παρέχει σαφώς μεγαλύτερη και καλύτερη ακρίβεια στο τελικό αποτέλεσμα.

Για την υλοποίηση ενός αυτοματισμού και γενικότερα την αυτοματοποίηση μιας παραγωγικής διαδικασίας εδώ και αρκετά χρόνια χρησιμοποιείται ευρέως και ιδιαίτερα σε βιομηχανίες οι λεγόμενοι Προγραμματιζόμενοι Λογικοί Ελεγκτές ή αλλιώς οι Programmable Logic Controllers (PLC). Τα PLC είναι ουσιαστικά μια ηλεκτρονική διάταξη που διαθέτουν ένα μικροϋπολογιστικό σύστημα, με πλήθος εισόδων και εξόδων, που συνδέονται με τα στοιχεία μιας εγκατάστασης και παράγουν ένα επιθυμητό αποτέλεσμα στις εξόδους ανάλογα με το συνδυασμό των εισόδων και το πρόγραμμα που έχουν εκτελούν. Το γεγονός το ότι είναι προγραμματιζόμενοι, τους προσδίδει ένα από τα πιο βασικά χαρακτηριστικά τους, διότι δεν απευθύνονται μόνο για μια συγκεκριμένη διεργασία, αλλά μπορούν να υλοποιήσουν πλήθος εφαρμογών, ακόμα και να βελτιστοποιήσουν την εφαρμογή που ήδη χρησιμοποιούνται, επεμβαίνοντας μόνο στον κώδικα. Κάτι το οποίο δεν ήταν εφικτό πριν από την εμφάνιση τους, μιας και ο κλασσικός αυτοματισμός που χρησιμοποιούνταν τα παλαιότερα χρόνια σε μια συγκεκριμένη διεργασία προοριζόταν αποκλειστικά για αυτή και η οποιαδήποτε μεταβολή ήταν αρκετά επίπονη και χρονοβόρα.

Ένα τέτοιο μικροϋπολογιστικό σύστημα είναι και το αντικείμενο της παρούσας πτυχιακής εργασίας και συγκεκριμένα του ελεγκτή S7 – 200 της εταιρείας Siemens. Αρχικά, γίνεται η μελέτη των κύριων χαρακτηριστικών του και του λογισμικού προγραμματισμού του και στη συνέχεια η σχεδίαση και η κατασκευή μιας μονάδας εκμάθησης με τη χρήση υλικών βιομηχανικού τύπου, στην οποία ο φοιτητής θα μπορεί να κατανοήσει και να εμβαθύνει τον τρόπο λειτουργίας και προγραμματισμού ενός PLC, μέσω διάφορων ασκήσεων βιομηχανικού αυτοματισμού.

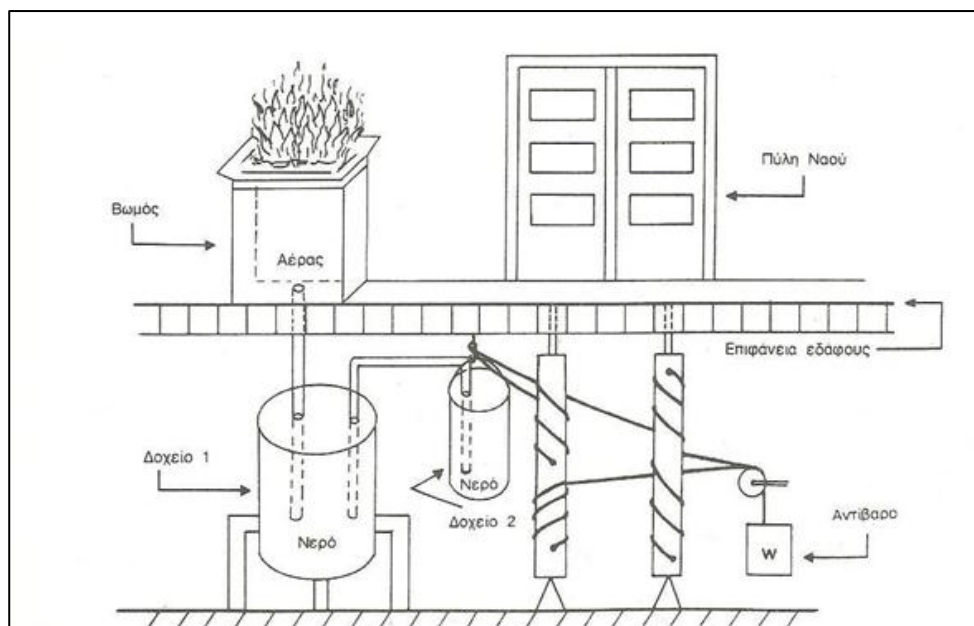
ΚΕΦΑΛΑΙΟ 1 – ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΑΥΤΟΜΑΤΙΣΜΟ ΚΑΙ ΣΤΑ PLC

1.1 Εισαγωγή στον αυτοματισμό

Ο αυτοματισμός δεν είναι μια ανάγκη που δημιουργήθηκε στην εποχή μας, απεναντίας έχει τις ρίζες του στα αρχαία χρόνια. Ένα πολύ χαρακτηριστικό παράδειγμα, που συναντάμε συχνά σε αναφορές της σύγχρονης βιβλιογραφίας, είναι μια κατασκευή γνωστή και ως “Ρυθμιστής του Ήρωος του Αλεξάνδρου” (εικόνα 1.1) η οποία καθόριζε τη θέση της πύλης ενός ναού με αυτόματο τρόπο χρησιμοποιώντας τα μέσα εκείνης της εποχής.

Η παρακάτω διάταξη λειτουργούσε ως εξής :

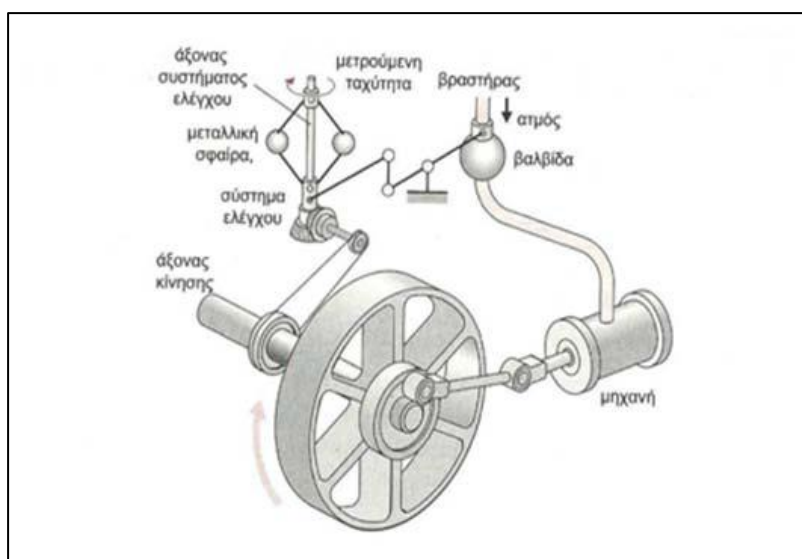
- Ανάβοντας φωτιά στον βωμό ο αέρας που ήταν εγκλωβισμένος κάτω από αυτόν ζεσταινόταν, με αποτέλεσμα να ωθεί το νερό που βρισκόταν από το δοχείο 1 στο άδειο δοχείο 2. Το δοχείο αυτό ήταν κρεμασμένο με σχοινιά τυλιγμένα κατάλληλα σε ένα μηχανισμό με ένα αντίβαρο. Όσο γέμιζε το δοχείο 2 με νερό αυξάνονταν το βάρος του και κατέβαινε προς τα κάτω ενώ το αντίβαρο ανέβαινε προς τα πάνω με αποτέλεσμα τα σχοινιά να ανοίγουν τις πύλες. Μόλις έσβηνε η φωτιά το νερό επέστρεφε στο δοχείο 1 και το αντίβαρο κατέβαινε προς τα κάτω, ενώ το δοχείο 2 ανέβαινε προς τα πάνω, με αποτέλεσμα η πύλη να κλείνει.



Εικόνα 1.1 : Ο ρυθμιστής του Ήρωος του Αλεξανδρέως

Συνεχίζοντας μια σύντομη ιστορική αναδρομή, οι αυτοματισμοί που ακολούθησαν ήταν αποκλειστικά μηχανικοί και βασιζόταν στην κίνηση γραναζιών και μοχλών. Ένα ακόμη χαρακτηριστικό παράδειγμα αποτελεί ο φυγοκεντρικός ρυθμιστής του WATT (εικόνα 1.2) ο οποίος χρησιμοποιήθηκε για να ελέγχει την ταχύτητα των ατμομηχανών. Ο τρόπος με το οποίο μεταβάλλονταν η ταχύτητα καθοριζόταν από την ποσότητα του ατμού που θα περνούσε στη μηχανή. Συγκεκριμένα, όσο η ταχύτητα περιστροφής αυξανόταν, οι σφαίρες από τον άξονα του συστήματος απομακρυνόταν λόγο

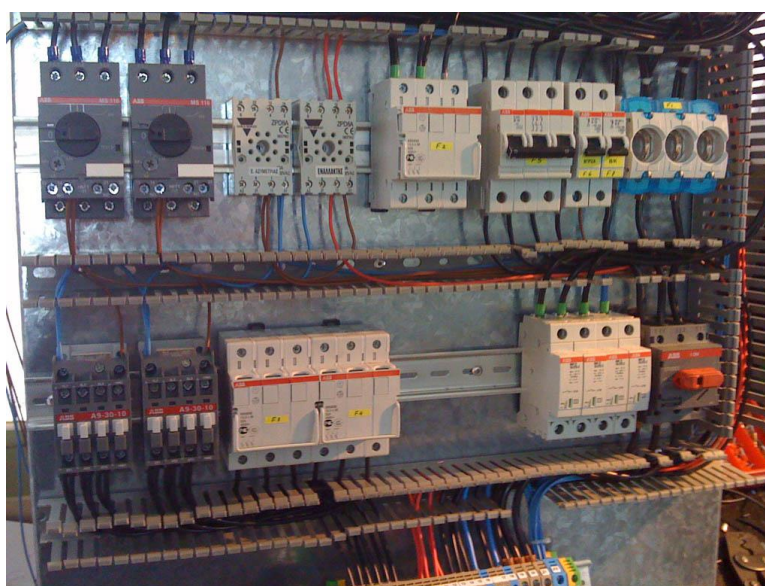
της φυγόκεντρου δύναμης οπότε έκλινει και η βαλβίδα και η μηχανή επιβράδυνε.



Εικόνα 1.2 : Ο φυγόκεντρικός ρυθμιστής του Watt

1.1.1 Κλασικός αυτοματισμός

Περνώντας στην πιο σύγχρονη εποχή, οι σημαντικότερες εξελίξεις στον αυτοματισμό έγιναν με τη χρήση του ηλεκτρισμού. Η υλοποίηση των αυτοματισμών στις προηγούμενες δεκαετίες στηρίχτηκε σε ηλεκτρολογικά στοιχεία όπως ρελέ, χρονικά, μετρητές κλπ και στην ενσύρματη λογική. Δηλαδή, για να επιτευχθεί ο αυτοματισμός όλα τα στοιχεία συνδεόταν μεταξύ τους με καλώδια. Για να ολοκληρωθεί ο αυτοματισμός και να κατασκευαστεί ο πίνακας (εικόνα 1.3) θα έπρεπε πιο μπροστά να έχει ολοκληρωθεί το ηλεκτρολογικό σχέδιο της εφαρμογής και μόνο τότε ήταν δυνατή η ολοκλήρωση της εφαρμογής. Όπως μπορεί να γίνει αντιληπτό, στον κλασικό αυτοματισμό η οποιαδήποτε μελλοντική αλλαγή έχει και σαν αποτέλεσμα αλλαγή και στην καλωδίωση του πίνακα, αύξηση του όγκου στο τελικό αποτέλεσμα μια διαδικασία χρονοβόρα που απαιτεί αρκετές γνώσεις.



Εικόνα 1.3 : Πίνακας κλασικού αυτοματισμού με ενσύρματη λογική

1.1.2 Αυτοματισμός με PLC

Με την πάροδο των χρόνων ολοένα και περισσότερο οι απαιτήσεις για αυτοματισμούς αυξανόταν και γινόταν πιο σύνθετοι. Ταυτόχρονα σημειώθηκε και αλματώδη εξέλιξη στην τεχνολογία και στα ηλεκτρονικά και κάτι τέτοιο δεν θα μπορούσε να αφήσει ανεπηρέαστο και τον τομέα των αυτοματισμών. Έτσι, περάσαμε σε αυτοματισμούς με ηλεκτρονικές διατάξεις, οι οποίες χρησιμοποιήθηκαν σε συνδυασμό με τα ηλεκτρολογικά κυκλώματα και κατάφεραν να επιλύσουν αρκετά προβλήματα.

Και γύρω στη δεκαετία του `80, όπου ο Η/Υ έχει εξελιχθεί αρκετά και έχει μπει στο χώρο της βιομηχανίας, οι εταιρείες κατασκευής παρουσιάζουν στους μηχανικούς ένα νέο προϊόν, τον Προγραμματιζόμενο Λογικό Ελεγκτή (εικόνα 1.4) όπου πλέον μιλάμε για προγραμματιζόμενο αυτοματισμό.



Εικόνα 1.4 : Αυτοματισμός με PLC

1.2 Σύγκριση των αυτοματισμών

Κάνοντας μια σύγκριση στα είδη των αυτοματισμών μπορούμε να διαπιστώσουμε πολύ σημαντικές διαφορές με αρκετά πλεονεκτήματα και μειονεκτήματα, τόσο κατά τη διαδικασία υλοποίησης των εφαρμογών όσο και στο τελικό αποτέλεσμα.

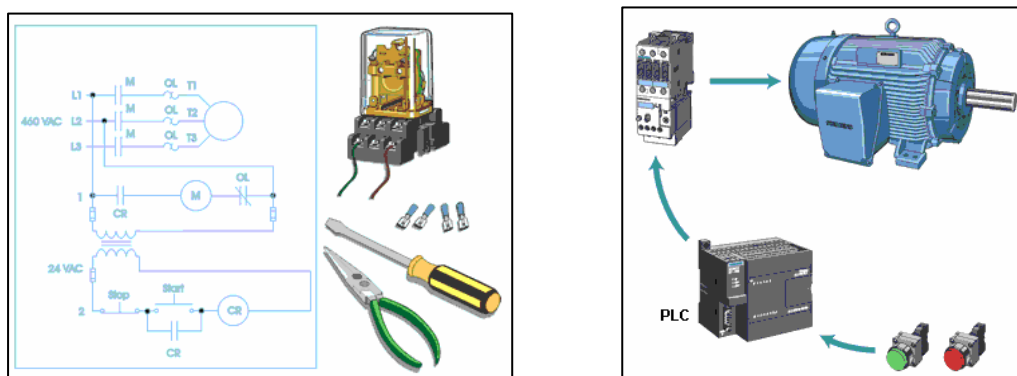
1.2.1 Στάδια υλοποίησης αυτοματισμών

Πιο συγκεκριμένα για την υλοποίηση μιας εφαρμογή με χρήση κλασσικού αυτοματισμού και ενσύρματης λογικής (εικόνα 1.5α) ακολουθούμε τα παρακάτω στάδια :

- i. Εκπόνηση ηλεκτρολογικού σχεδίου
- ii. Επιλογή υλικών, ρελέ, βοηθητικών ρελέ, χρονικών μετρητών κλπ βάση του ηλεκτρολογικού σχεδίου.
- iii. Εγκατάσταση όλων των στοιχείων μέσα στον ηλεκτρολογικό πίνακα
- iv. Καλωδίωση των εξωτερικών στοιχείων της εγκατάστασης (τερματικοί διακόπτες, μπουτόν, ενδεικτικές λυχνίες, βαλβίδες) στις κλέμες του πίνακα
- v. Καλωδίωση των υλικών αυτοματισμού μέσα στον πίνακα μεταξύ τους, σύμφωνα με το ηλεκτρολογικό σχέδιο.

Ενώ για την υλοποίηση μιας εφαρμογής με τη χρήση PLC (εικόνα 1.5β) ακολουθούνται τα παρακάτω στάδια :

- i. Επιλογή κατάλληλου PLC, ανάλογα με τις απαιτήσεις της εφαρμογής
- ii. Εγκατάσταση στον πίνακα
- iii. Καλωδίωση των εξωτερικών στοιχείων της εγκατάστασης (τερματικοί διακόπτες, μπουτόν, ενδεικτικές λυχνίες) στις κλέμες του ελεγκτή
- iv. Προγραμματισμός



α) Κλασσικός αυτοματισμός

β) Αυτοματισμός με PLC

Εικόνα 1.5 : Είδη αυτοματισμών

1.2.2 Πλεονεκτήματα - Μειονεκτήματα

Με μια αρχική εκτίμηση μπορεί να γίνει αντιληπτό ότι οι αυτοματισμοί με PLC παρουσιάζουν αρκετά πλεονεκτήματα έναντι του κλασσικού. Ενδεικτικά θα μπορούσαμε να αναφέρουμε τα παρακάτω :

- Ένας πίνακας με PLC καταλαμβάνει μικρότερο χώρο
- Τόσο ο χρόνος όσο και το κόστος κατασκευής του πίνακα που φιλοξενεί τα υλικά είναι ελάχιστος και αυτό λόγω του μεγάλου αριθμού βοηθητικών επαφών που παρέχει ένα PLC
- Για να επεκτείνουμε ή για να τροποποιήσουμε μια εφαρμογή αρκεί να γίνει επέμβαση στον κώδικα ή και να τοποθετήσουμε καινούργιες μονάδες εισόδων / εξόδων κάτι που δεν υφίσταται στον κλασσικό αυτοματισμό, όπου εκεί μια τέτοια διαδικασία είναι αρκετά δύσκολη
- Ελαχιστοποιείται το κόστος συντήρησης και αποκατάστασης της βλάβης : στον κλασσικό αυτοματισμό όταν χαλάει ένα υλικό, π.χ. ένα χρονικό, θα πρέπει να είναι διαθέσιμο ακριβώς το ίδιο και να γίνει αντικατάσταση με καινούργιο. Στους αυτοματισμούς με PLC, δεν ισχύει κάτι τέτοιο, καθώς το να εμφανίσουν κάποια βλάβη είναι πολύ σπάνιο για αυτό και οι κατασκευάστριες εταιρείες παρέχουν μεγάλες εγγυήσεις
- Ο εντοπισμός της τρέχουσας κατάστασης εισόδων και εξόδων είναι πολύ εύκολος καθώς κάθε PLC διαθέτει ενδεικτικές λυχνίες LED που μας το δείχνουν
- Ένα PLC μπορεί να χρησιμοποιηθεί σε διαφορετικές εφαρμογές

Σαν μειονέκτημα θα μπορούσαμε να πούμε ότι ένα PLC δεν θα προτιμηθεί για την υλοποίηση ενός απλού αυτοματισμού με ελάχιστες απαιτήσεις, διότι κάτι τέτοιο θα ήταν ασύμφορο οικονομικά και η επιλογή κλασσικού αυτοματισμού σαφώς και αποτελεί την καλύτερη. Ένα ακόμα εξίσου σημαντικό μειονέκτημα είναι ότι οι γλώσσες προγραμματισμού των PLC δεν είναι τυποποιημένες, πράγμα που σημαίνει ότι κάθε εταιρεία έχει τις δίκες της ιδιομορφίες στις αντίστοιχες γλώσσες. Ακόμα και για την ίδια εταιρεία, από μοντέλο σε μοντέλο συναντάμε διαφορές στις γλώσσες κάτι που σημαίνει ότι αυτός που θα το προγραμματίσει θα πρέπει πιο μπροστά να αφιερώσει χρόνο και να μελετήσει το αντίστοιχο εγχειρίδιο.

1.3 Εφαρμογές των PLC

Γενικά τα PLC βρίσκουν αρκετές εφαρμογές σε διάφορους τομείς, εκτός των βιομηχανιών. Ενδεικτικά τέτοια συστήματα τα συναντάμε :

- Σε συστήματα ανελκυστήρων
- Στη ναυτιλία και συγκεκριμένα σε πλοία
- Αυτόματες μηχανές συσκευασίες
- Στη μηχανολογία σε μηχανές CNC
- Στον έλεγχο σηματοδότησης
- Σε αυτοματισμούς θυρών
- Σε συστήματα παραγωγής ενέργειας, όπως φωτοβολταϊκά και ανεμογεννήτριες
- Συναγερμοί
- Σε χώρους ελέγχου αντλιοστασίων

Όπως μπορεί να γίνει αντιληπτό, καλύπτουν ένα μεγάλο φάσμα εφαρμογών για αυτό το λόγω αρκετοί μηχανικοί ασχολούνται με το προγραμματισμό τους και μπορούν να υλοποιήσουν αυτοματισμούς με υψηλές δυνατότητες κερδίζοντας αρκετό χρόνο.

1.4 Τύποι PLC

Κάνοντας κανείς μια έρευνα στην αγορά, θα διαπιστώσει πως κυκλοφορούν αρκετοί τύποι προγραμματιζόμενων λογικών ελεγκτών με διαφορές στο μέγεθος και στις δυνατότητες τους, που ποικίλουν ανάλογα με την εταιρία κατασκευής. Γενικά, θα μπορούσαμε να πούμε πως διακρίνουμε δύο τύπους PLC, τα **compact** και τα **modular**. Τα πρώτα (εικόνα 1.2α) είναι μια συμπαγής και ενιαία συσκευή όπου όλες οι περιφερειακές του βαθμίδες (τροφοδοσία, CPU, είσοδοι – έξοδοι) βρίσκονται στην ίδια μονάδα, ενώ τα modular κάθε βαθμίδα αποτελεί ξεχωριστή συσκευή (εικόνα 1.2β)

Οι κυριότερες διαφορές μεταξύ των δυο τύπων θα μπορούσαμε να πούμε ότι είναι οι εξής :

Τα compact PLC έχουν περιορισμένες δυνατότητες, δηλαδή, έχουν μικρό αριθμό εισόδων και εξόδων και έχουν μικρή δυνατότητα επέκτασης και μικρότερη επεξεργαστική ισχύ. Το βασικό τους πλεονέκτημα τους όμως είναι ότι αποτελούν μια πολύ οικονομική λύση σε σχέση με τα modular.

Αντιθέτως, τα modular PLC χρησιμοποιούνται για να υλοποιήσουμε απαιτητικούς αυτοματισμούς με μεγάλο αριθμό εισόδων και εξόδων και σε περίπτωση που μελλοντικά χρειαστεί να επεκτείνουμε την εφαρμογή μας, αρκεί να προσθέσουμε μια ή περισσότερες βαθμίδες εισόδων ή εξόδων

διατηρώντας την ίδια CPU και το ίδιο τροφοδοτικό. Αν και πιο ισχυρά, αποτελούν σίγουρα μια αρκετή ακριβή και απαγορευτική λύση, ειδικά για πιο εφαρμογές χαμηλών απαιτήσεων.



α) Compact



β) Modular

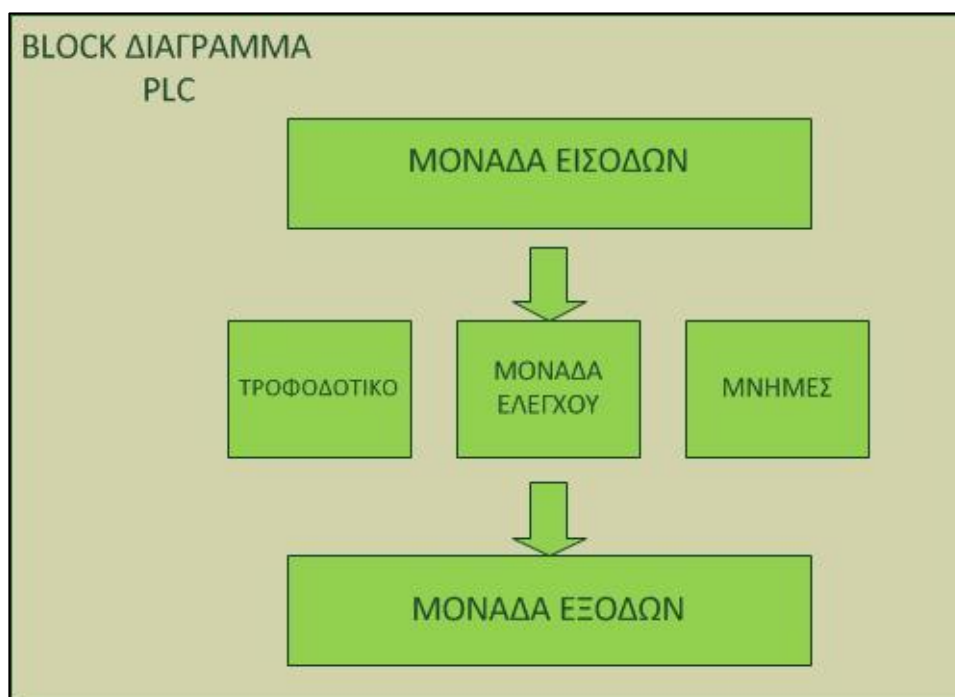
Εικόνα 1.2 : Τύποι PLC

1.5 Δομή PLC

Σε οποιαδήποτε κατασκευάστρια εταιρεία και να ανήκει ένα PLC και ανεξάρτητα από το τι τύπος είναι, η βασική δομή ενός PLC παραμένει η ίδια καθώς αποτελεί ένα μικροϋπολογιστικό σύστημα με το υλικό του μέρος να μοιάζει με έναν Η/Υ. Συνεπώς, σε ένα PLC διακρίνουμε τις εξής βαθμίδες :

- i. Μονάδες εισόδων
- ii. Μονάδες εξόδων
- iii. Κεντρική μονάδα επεξεργασίας (CPU)
- iv. Μνήμες
- v. Μονάδες τροφοδοσίας

Στην εικόνα 1.3 απεικονίζονται οι διάφορες βαθμίδες από τις οποίες αποτελείται ένα PLC.



Εικόνα 1.3 : Μπλοκ διάγραμμα βαθμίδων

i) Μονάδες εισόδων

Αποτελούν τις μονάδες επικοινωνίας όπου συνδέεται η κεντρική μονάδα επεξεργασίας με τα στοιχεία μιας εγκατάστασης. Αυτά τα στοιχεία συνήθως είναι διακόπτες, μπουτόνς και οποιοδήποτε άλλο αισθητήριο εισάγει μια πληροφορία όταν θα συμβεί ένα γεγονός. Διακρίνουμε δυο τύπους εισόδων, τις ψηφιακές και τις αναλογικές.

- Οι ψηφιακές εισοδοι αναγνωρίζουν μόνο δύο τιμές τάσης (υψηλή – χαμηλή). Τα αισθητήρια ή τα στοιχεία εισόδου είναι συσκευές που μετατρέπουν μια φυσική κατάσταση σε ηλεκτρικό σήμα που μεταφέρεται στην είσοδο του PLC. Το πιο κλασσικό παράδειγμα είναι το μπουτόν. Όταν το πιέζουμε η “φυσική” του κατάσταση μεταβάλλεται και η πληροφορία (μέσω της φυσικής αλλαγής της κατάστασης της επαφής του) μεταφέρεται σαν ηλεκτρικό σήμα (ρεύμα) στην είσοδο του PLC.
- Οι αναλογικές εισοδοι αντιλαμβάνονται περισσότερες καταστάσεις από δύο και συγκεκριμένα μια κατάσταση που μεταβάλλονται συνεχώς. Για παράδειγμα, μια αναλογική είσοδος χρησιμοποιείται για την μέτρηση της στάθμης ενός υγρού υλικού σε μια δεξαμενή. Η μεταβαλλόμενη στάθμη του υγρού “μεταφράζεται” από το αισθητήριο σε ένα αντίστοιχα μεταβαλλόμενο ηλεκτρικό σήμα που κυμαίνεται σε μια τυποποιημένη κλίμακα έντασης ρεύματος (π.χ. 4 έως 20 mA) ή τάσης ρεύματος (π.χ. 0 έως 10 V). Η ειδική τύπου αναλογική είσοδος του PLC “αντιλαμβάνεται” τις διαφοροποιήσεις (αυξομειώσεις του ηλεκτρικού ρεύματος ή τάσης) και τις “μεταφράζει” σε μεταβολές (αυξομειώσεις) του φυσικού φαινομένου, δηλαδή της στάθμης του υγρού.

ii) Μονάδες εξόδων

Οι μονάδες εξόδων αποτελούν τις βαθμίδες με τις οποίες συνδέεται το PLC με τα φορτία της εφαρμογής μας. Και οι μονάδες εξόδων διακρίνονται σε ψηφιακές και αναλογικές :

- Οι ψηφιακές έξοδοι μπορούν να έχουν κατάσταση ON ή OFF. Σ αυτές συνδέουμε φορτία τα οποία είναι δύο καταστάσεων, όπως ενδεικτικές λυχνίες. Η σύνδεση των φορτίων με τις εξόδους γίνεται είτε απ’ ευθείας ή μέσω διατάξεων ενεργοποίησης όπως ρελέ.
- Στις αναλογικές εξόδους, υπάρχουν περισσότερες από δύο καταστάσεις και μεταβάλλονται συνεχώς. Για παράδειγμα μια αναλογική έξοδος μπορεί να παρέχει ηλεκτρικό σήμα και να ελέγχει τη θέση μιας ηλεκτροβάνας.

iii) Κεντρική μονάδα επεξεργασίας

Όπως και στους υπολογιστές, έτσι και στα PLC, η CPU είναι ένα ψηφιακό κύκλωμα που αποτελείται ουσιαστικά από έναν μικροελεγκτή και αποτελεί τον εγκέφαλο ενός PLC. Στη CPU εκτελείται το πρόγραμμα το οποίο του έχουμε φορτώσει. Είναι αυτή που θα “πάρει” τις αποφάσεις και θα μεταβάλλει την κατάσταση των εξόδων ανάλογα με την κατάσταση των εισόδων και με τη λογική με την οποία το έχουμε προγραμματίσει. Ανάλογα ποιον τύπο PLC χρησιμοποιούμε και σε ποια οικογένεια ανήκει ποικίλουν και οι δυνατότητες της CPU.

iv) Μνήμες

Όσον αφορά τις μνήμες στις CPU των PLC, διακρίνουμε τρεις γενικές κατηγορίες :

- i) Τη μνήμη φόρτωσης,
- ii) την κύρια μνήμη και
- iii) τη μνήμη συστήματος

Στη μνήμη φόρτωσης γίνεται η αποθήκευση του προγράμματος χρήστη, καθώς επίσης και όλες οι πληροφορίες διαμόρφωσης που αποτελείται το hardware του PLC. Η συγκεκριμένη μνήμη, μπορεί να επεκταθεί με εξωτερικές μνήμες RAM, ανάλογα με τον τύπο PLC που χρησιμοποιείται.

Η κύρια μνήμη, είναι τύπου RAM και είναι πλήρως ενσωματωμένη στη CPU και περιέχει τα κυριότερα τμήματα του προγράμματος του χρήστη που είναι ο κώδικας του προγράμματος και τα δεδομένα του χρήστη. Η μεταφορά των απαραίτητων στοιχείων από τη μνήμη φόρτωσης προς τη μνήμη εργασίας γίνεται από το λειτουργικό σύστημα της CPU.

Ενώ στην μνήμη συστήματος, υπάρχουν όλες οι μεταβλητές που αναφερόμαστε στο πρόγραμμα μας. Αυτές είναι ομαδοποιημένες στη μνήμη και το μέγεθος τους εξαρτάται από τη CPU που χρησιμοποιούμε. Γενικά στη συγκεκριμένη μνήμη υπάρχουν οι παρακάτω διευθύνσεις :

- **Μνήμη απεικόνισης εισόδων** : Εδώ καταγράφονται οι τιμές των εισόδων κατά την ανάγνωση από τη CPU
- **Μνήμη απεικόνισης εξόδων** : Αφού εκτελεστεί το πρόγραμμα, η CPU διαβάζει τη συγκεκριμένη περιοχή και στέλνει τις τιμές που πρόέκυψαν στις κάρτες
- **Βοηθητικά** : Αποθηκευμένες πληροφορίες προσπελάσιμες σε όλο το πρόγραμμα
- **Χρονικά** : Θέσεις που χρησιμοποιούνται για την υλοποίηση μέτρηση χρόνων
- **Μετρητές** : Βρίσκεται αποθηκευμένο το περιεχόμενο των μετρητών
- **Διαγνωστικά – Buffers** : στο συγκεκριμένο χώρο αποθηκεύονται πληροφορίες που έχουν να κάνουν με εργασίες επικοινωνίας και μηνύματα του συστήματος π.χ αν συμβεί κάποιο σφάλμα.

v) Μονάδες τροφοδοσίας

Ο σκοπός του τροφοδοτικού, όπως και σε κάθε συσκευή, είναι να δημιουργεί τις απαραίτητες τάσεις που χρειάζεται το PLC για να λειτουργήσει. Η τάση με την οποία τροφοδοτείται ένα PLC εξαρτάται καθαρά από την κατηγορία και το μοντέλο στην οποία ανήκει. Έτσι συναντάμε PLC που για να λειτουργήσουν αρκεί άπλα να συνδέσουμε τις αντίστοιχες κλέμες στην τάση δικτύου στα 220 Volt, όπως αυτό της εφαρμογής μας, ενώ υπάρχουν άλλα που η κεντρική τους τροφοδοσία είναι 24 Volt DC, οπότε σε μια τέτοια περίπτωση θα πρέπει να χρησιμοποιήσουμε εξωτερικό τροφοδοτικό. Σε κάθε περίπτωση θα πρέπει να είμαστε ιδιαίτερα προσεκτικοί και να συμβουλευόμαστε τα φύλλα δεδομένων του κατασκευαστή για να μην προκληθεί καταστροφή του ελεγκτή.

1.6 Αρχή λειτουργίας των PLC

Κάνοντας μια αντιπαράθεση ενός PLC με έναν Η/Υ θα λέγαμε ότι ένας υπολογιστής επεξεργάζεται τις εντολές και τα στοιχεία που του έχουμε δώσει, τα εκτελεί καταλήγει στα αντίστοιχα αποτελέσματα και στη συνέχεια σταμάτα. Με τα PLC όμως δεν ισχύει κάτι τέτοιο, αφού η συνεχής κυκλική επεξεργασία (εικόνα 1.6) και εκτέλεση του προγράμματος αποτελεί βασικό χαρακτηριστικό τους. Πιο συγκεκριμένα η αρχή λειτουργίας τους στηρίζεται στα παρακάτω στάδια :

I. Στάδιο 1^ο :

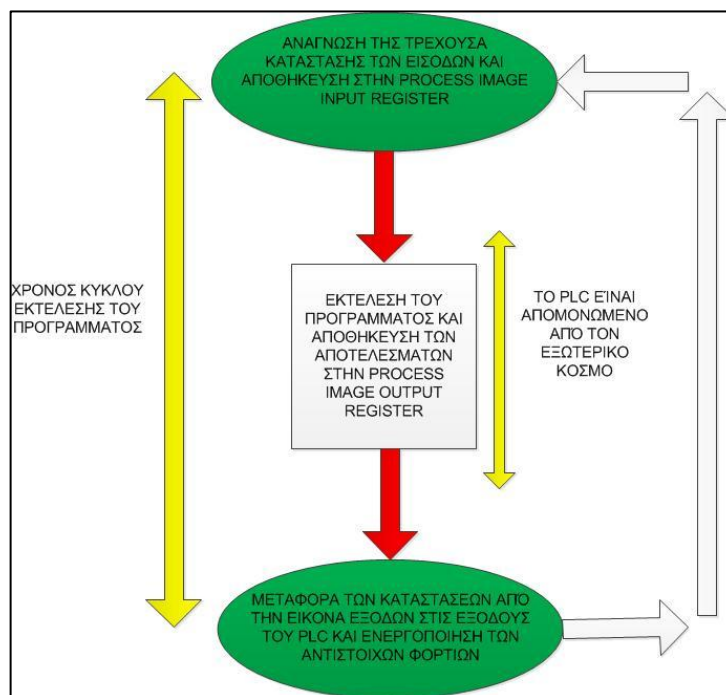
Πριν την εκτέλεση του κυρίως προγράμματος, ο ελεγκτής ελέγχει την τρέχουσα κατάσταση των εισόδων. Έτσι, αν μία είσοδος βρίσκεται σε υψηλό ή αν βρίσκεται σε χαμηλό δυναμικό, αποθηκεύει την τιμή 1 ή 0 αντίστοιχα, σε μια ειδική περιοχή μνήμης που ονομάζεται εικόνα εισόδων (process image input register).

II. Στάδιο 2^ο :

Ακολουθεί η εκτέλεση του προγράμματος. Η δομή του προγράμματος αποτελεί μια σειρά από λογικές πράξεις, όπου ο ελεγκτής παίρνει σαν δεδομένα τις καταστάσεις των εισόδων και προχωράει στην εκτέλεση των εντολών, που υλοποιούν τον αυτοματισμό, ξεκινώντας από την πρώτη και καταλήγοντας στην τελευταία. Αφού ολοκληρωθεί η εκτέλεση του προγράμματος, ο ελεγκτής αποδίδει τα αποτελέσματα στην ειδική περιοχή που λέγεται εικόνα εξόδων (process image output register) η οποία θα περιέχει τις τιμές των καταστάσεων για κάθε έξοδο ή λογικό 0 ή λογικό 1.

III. Στάδιο 3^ο :

Ο ελεγκτής, τέλος, μεταφέρει τις τιμές που βρίσκονται στην περιοχή εικόνας εξόδων στις εξόδους του οπότε θα ενεργοποιηθούν και τα αντίστοιχα φορτία που αποτελούν τον αυτοματισμό



Εικόνα 1.4 : Κύκλος λειτουργίας και χρόνος κύκλου

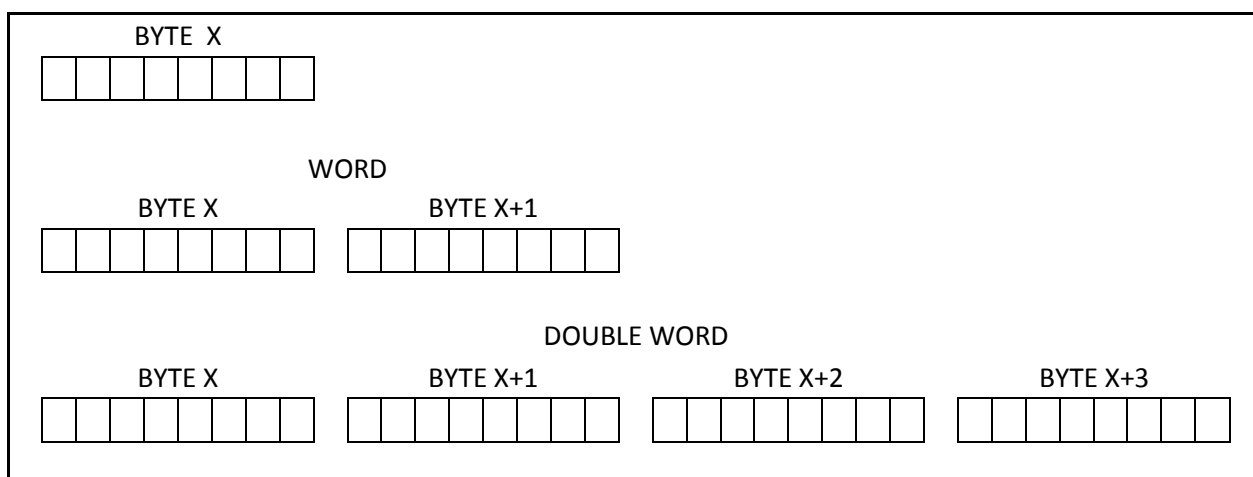
Αφού ολοκληρωθεί και το 3^ο στάδιο, έχει συμπληρωθεί ένας πλήρης κύκλος λειτουργίας και η διαδικασία ξεκινάει από την αρχή κυκλικά. Άρα σαν αρχή λειτουργίας τα PLC θα μπορούσαμε να πούμε ότι βρίσκονται σε μια διαρκή σάρωση και εκτέλεση εντολών. Ένα πολύ σημαντικό στοιχείο που πρέπει να αναφερθεί είναι ότι η κατάσταση των εισόδων διαβάζονται από το PLC μόνο κατά την διάρκεια του πρώτου σταδίου. Όταν βρίσκεται στο 2^ο στάδιο το PLC είναι ουσιαστικά απομονωμένο από τον έξω κόσμο. Με άλλα λόγια αν συμβεί μια αλλαγή σε μία είσοδο το PLC θα την διαβάσει στο επόμενο κύκλο, ο οποίος φυσικά ξεκινάει μετά από μερικά msec.

1.6.1 Χρόνος κύκλου

Το χρονικό διάστημα που χρειάζεται ένα PLC για να εκτελέσει έναν πλήρη κύκλο ονομάζεται χρόνος κύκλου και εξαρτάται καθαρά από την ταχύτητα του επεξεργαστή και από το είδος και το πλήθος των εντολών που εκτελεί. Αποτελεί σίγουρα ένα μέσο σύγκρισης των PLC και συγκεκριμένα οι κατασκευαστές ορίζουν σαν μέσο χρόνο κύκλου το χρονικό διάστημα που χρειάζεται να εκτελέσει 1Kbyte ή 1024 εντολές, ο οποίος γενικά είναι της τάξεως των msec. Χρόνος αρκετά μικρός, ώστε να θεωρηθεί ότι κάτι τέτοιο αποτελεί μειονέκτημα η χρήση των PLC.

1.7 Ονοματολογία - Διευθυνσιοδότηση

Όπως έχει αναφερθεί, το PLC είναι ένα μικροϋπολογιστικό σύστημα, που περιέχει τον μικροελεγκτή και τις μνήμες. Αναμενόμενο είναι, πως για να γίνει μια προσέγγιση και περιγραφή, να χρησιμοποιηθούν έννοιες όπως bits και bytes. Στην εικόνα 1.5 φαίνεται των πλήθος των bits που αποτελούν ένα, δύο και τέσσερα bytes.



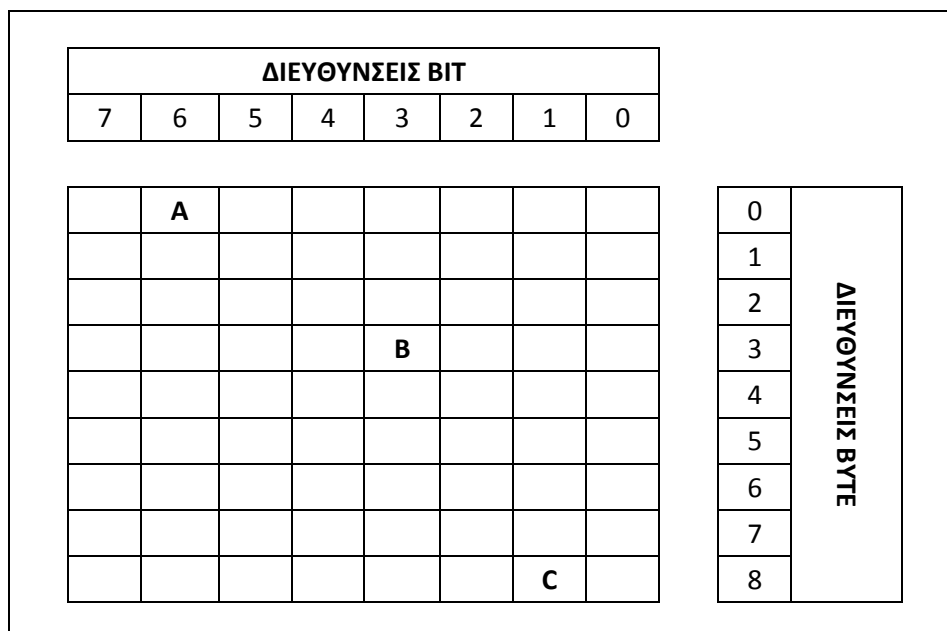
Εικόνα 1.5 : Bytes

Οι μνήμες των PLC είναι οργανωμένες κατά bytes και κάθε πληροφορία αποθηκεύεται σε διαφορετική τοποθεσία της μνήμης η οποία έχει μοναδική διεύθυνση. Έτσι, όταν θέλουμε να προσπελάσουμε μια διεύθυνση μνήμης αρκεί να προσδιορίσουμε την επιθυμητή διεύθυνση.

Γενικά, όταν εργαζόμαστε με PLC, ένα από τα πρώτα πράγματα που πρέπει να κάνουμε είναι να προσδιορίσουμε το πλήθος των φυσικών εισόδων και εξόδων που διαθέτει και πως αυτές ονομάζονται. Εκτός από τις φυσικές

εισόδους και εξόδους στα PLC συναντάμε και τα βοηθητικά, τα χρονικά και τους μετρητές τα οποία χρησιμοποιούνται για την υλοποίηση των αυτοματισμών.

Κάθε στοιχείο που χρησιμοποιούμε κατέχει και διαφορετική διεύθυνση μνήμης, όποτε η προσπέλαση τους γίνεται με αποκλειστικό τρόπο. Για παράδειγμα, έστω έχουμε τις παρακάτω περιοχές διευθύνσεων (εικόνα 1.6) στην οποία αντιστοιχεί οποιοδήποτε στοιχείο του PLC (είσοδος / έξοδος κλπ).



Εικόνα 1.6 : Περιοχές διευθύνσεων

Το στοιχείο A θα αντιστοιχεί στη διεύθυνση 0.6, το στοιχείο B στην 3.3 και το C στην 8.1. Όπως γίνεται αντιληπτό ο πρώτος αριθμός αντιστοιχεί στην διεύθυνση των byte και ο δεύτερος στη θέση bit.

• **Φυσικές εισοδοι (I)**

Οι φυσικές εισοδοι στα PLC συνήθως συμβολίζονται με το γράμμα I και για τα μικρά PLC ακολουθεί ένας αύξον αριθμός που ξεκινάει από το 0 ή το 1 και φτάνει μέχρι το συνολικό πλήθος τους. Για παράδειγμα αν είχαμε ένα PLC με 8 εισόδους η διευθύνσεις τους θα ξεκινούσε από το I0 μέχρι το I7. Στα μεγαλύτερα PLC, όπως αυτό της εφαρμογής μας το γράμμα I ακολουθείται από δύο αριθμούς, όπου ο πρώτος αντιστοιχεί στην διεύθυνση του byte και ο δεύτερος στη θέση του bit του συγκεκριμένου byte, σύμφωνα με την παραπάνω αναφορά.

Παράδειγμα : I x.y : x (0-n) διεύθυνση byte, ανάλογα με το πλήθος εισόδων
y (0-7) διεύθυνση bit

• **Φυσικές έξοδοι (Q)**

Οι φυσικές έξοδοι συμβολίζονται με το γράμμα Q ή O, ανάλογα με την εταιρία. Ότι ισχύει για τις εισόδους το ίδιο ισχύει με και για τις εξόδους.

Παράδειγμα : Q x.y : x (0-n) διεύθυνση byte, ανάλογα με το πλήθος εξόδων
y (0-7) διεύθυνση bit

- **Βοηθητικά (M)**

Τα βοηθητικά συμβολίζονται με το γράμμα M και αντιστοιχούν στα βοηθητικά ρελέ που χρησιμοποιούμε στον κλασικό αυτοματισμό. Χρησιμοποιούνται σε περιπτώσεις όπου αντί να επαναλάβουμε τμήματα κώδικα, καταγράφεται μία φορά η επιθυμητή λογική, αποθηκεύεται το αποτέλεσμα στο βοηθητικό και χρησιμοποιούμε το βοηθητικό σαν επαφή όσες φορές χρειάζεται. Θα μπορούσαμε να τα παρομοιάσουμε με εξόδους των PLC, μόνο που δεν οδηγούνται απευθείας στην εγκατάσταση και δεν απεικονίζεται η κατάσταση τους στα LED των PLC.

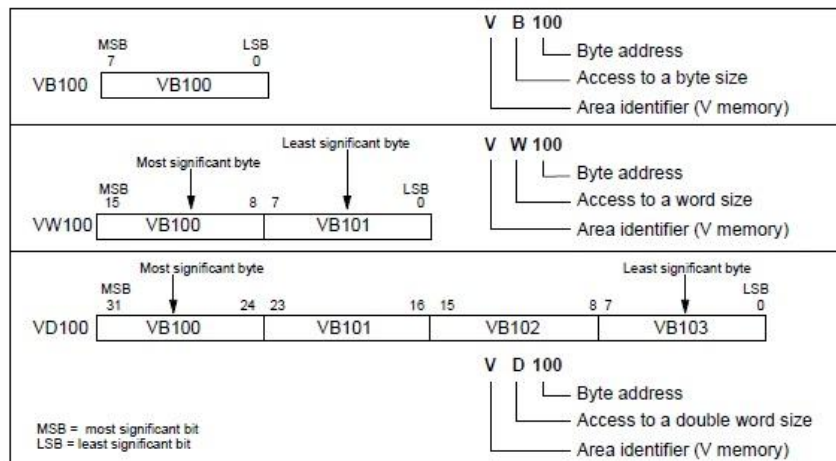
Παράδειγμα : M x.y :x (0-n) διεύθυνση byte
y (0-7) διεύθυνση bit

- **Μνήμη μεταβλητών**

Σε κάθε PLC υπάρχει μια μεγάλη περιοχή μνήμης στην οποία μπορούμε να αποθηκεύουμε τα ενδιάμεσα αποτελέσματα των πράξεων που εκτελούνται από το πρόγραμμα μας. Επίσης, μπορούμε να αποθηκεύσουμε και άλλα δεδομένα που αφορούν την διεργασία μας. Για κάθε PLC το μέγεθος αλλά και η ονοματολογία για αυτήν την περιοχή ποικίλει και θα πρέπει να συμβουλευόμαστε τα φύλλα δεδομένων

Παράδειγμα : V x.y :x (0-n) διεύθυνση byte
y (0-7) διεύθυνση bit

Ένας άλλο τρόπος για να αποκτήσουμε πρόσβαση στα παραπάνω στοιχεία είναι αυτός με τη χρήση διευθύνσεων και ονομάτων bytes, words ή double words ανάλογα που αυτό που μας εξυπηρετεί καλύτερα. Δηλαδή, ενώ στα παραπάνω παραδείγματα η πρόσβαση γίνεται δηλώνοντας τον στοιχείο που θέλουμε, την διεύθυνση του byte που ανήκει και τη θέση του bit που θέλουμε να τροποποιήσουμε, με αυτόν τον τρόπο πρέπει να δηλώσουμε το στοιχείο που θέλουμε (π.χ. I, Q, M κλπ) το μέγεθος των δεδομένων (Byte, Word, Double Word) και το byte εκκίνησης. Για παράδειγμα αν θέλουμε να τροποποιήσουμε τα οκτώ πρώτα Bits του βοηθητικού που βρίσκεται στη διεύθυνση 0, αρκεί να χρησιμοποιήσουμε σαν όνομα το MB0. Στην εικόνα 1.7 που ακολουθεί φαίνεται η εναλλακτική πρόσβαση στα διάφορα στοιχεία :



Εικόνα 1.7 : Πρόσβαση σε δεδομένα με τη χρήση byte, word, double word

•Χρονικά (T)

Τα χρονικά συμβολίζονται με το γράμμα T και τα χρησιμοποιούμε για να πετύχουμε εφαρμογές που έχουν σχέση με το χρόνο, όπως αναμονή κάποιου χρονικού διαστήματος. Βρίσκονται στην περιοχή της μνήμης των χρονικών και αφού τους δίνουμε την τιμή του χρόνου που θέλουμε να μετρήσουμε, μόλις ενεργοποιηθούν η τιμή τους καταγράφεται σε έναν 16 bit αριθμό και μόλις ξεπεραστεί η προκαθορισμένη τιμή τους ενεργοποιείται και η αντίστοιχη επαφή τους. Το είδος των χρονικών εξαρτάται από το PLC που χρησιμοποιείται.

Παράδειγμα : T_x : x ο αριθμός του χρονικού που χρησιμοποιείται

•Απαριθμητές (C)

Οι απαριθμητές συμβολίζονται με το γράμμα C και χρησιμοποιούνται σε εφαρμογές όπως η μέτρηση κάποιου προϊόντος. Στη μνήμη βρίσκονται σε ξεχωριστή περιοχή, στην περιοχή των απαριθμητών και η τιμή τους καταγράφεται σε έναν 16 bit καταχωρητή. Ανάλογα με το PLC που χρησιμοποιείται υπάρχουν απαριθμητές που μετρούν προς τα πάνω και άλλοι που μετρούν με φθίνουσα μέτρηση.

Παράδειγμα : C_x : x ο αριθμός του μετρητή που χρησιμοποιείται

ΚΕΦΑΛΑΙΟ 2 – ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ PLC

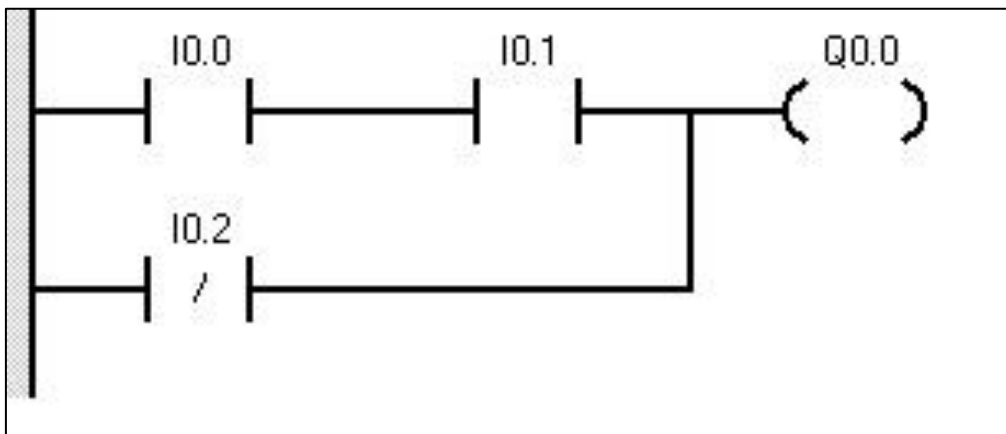
2.1 Γλώσσες προγραμματισμού

Υλοποιώντας έναν αυτοματισμό με PLC το πιο δύσκολο κομμάτι δεν είναι τόσο το hardware όσο ο τρόπος με το οποίο θα δουλέψει το υλικό μέρος, με τέτοιο τρόπο ώστε να διασφαλιστεί η αξιοπιστία της εφαρμογής. Ο τρόπος με τον οποίο θα δουλέψει το υλικό μέρος εξαρτάται καθαρά από το πρόγραμμα το οποίο έχουμε φορτώσει στο PLC και αποτελεί αδιαμφισβήτητα το πιο δύσκολο κομμάτι όσον αφορά τη χρήση τους. Λόγω του μεγάλου ανταγωνισμού από τις κατασκευάστριες εταιρείες, δεν υπάρχει κάποια τυποποίηση στις γλώσσες, όπως ισχύει στους Η/Υ και παρόλο που χρησιμοποιούνται τρεις βασικές γλώσσες, από εταιρεία σε εταιρεία παρατηρούνται σημαντικές διαφορές στον τρόπο χρήσης τους.

Το ποια γλώσσα θα χρησιμοποιηθεί εξαρτάται καθαρά από αυτόν που θα προγραμματίσει και την εμπειρία του και σαφώς από την εφαρμογή που πρέπει να υλοποιηθεί. Τρεις είναι οι επικρατέστερες γλώσσες που συναντάμε για να προγραμματίσουμε ένα PLC. Η πρώτη, που είναι και η πιο γνωστή, είναι η γλώσσα **LADDER (LAD)** ή αλλιώς σχέδιο επαφών, η δεύτερη είναι η **STATEMENT LIST (STL)** ή λίστα εντολών και τρίτη η **FUNCTION BLOCK DIAGRAM (FBD)** ή γλώσσα λογικού διαγράμματος.

2.1.1 Η γλώσσας LADDER - LAD

Είναι η πρώτη γλώσσα που χρησιμοποιήθηκε στα PLC. Θυμίζει πάρα πολύ τη δομή ενός ηλεκτρολογικού σχεδίου αυτοματισμού και αυτό δεν είναι τυχαίο διότι η εταιρείες ήθελαν να είναι προσιτό στους μηχανικούς και να μην είναι τελείως άγνωστο απ' ό,τι είχαν συνηθίσει μέχρι τότε. Αποτελεί ουσιαστικά τη μετατροπή του ηλεκτρολογικού σχεδίου σε γλώσσα κατανοητή από το PLC. Στο σχήμα 2.1 φαίνεται ένα παράδειγμα προγράμματος σε γλώσσα LADDER. Η γλώσσα LADDER περιγράφεται αναλυτικότερα σε επόμενη ενότητα.



Σχήμα 2.1 : Η γλώσσα LADDER

2.1.2 Η γλώσσα Statement List – STL

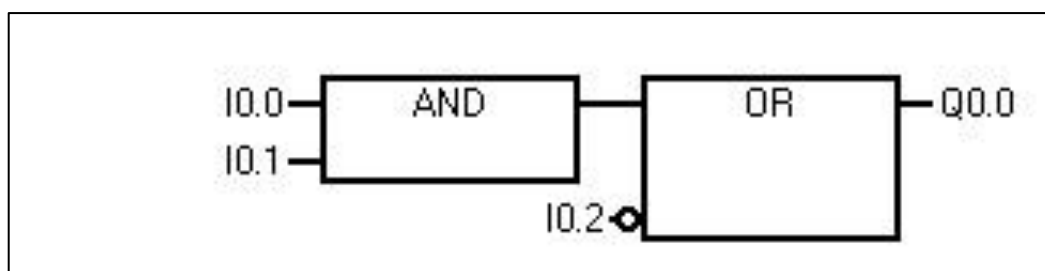
Αν παρατηρήσει κάποιος τη συγκεκριμένη γλώσσα, θυμίζει πάρα πολύ τη γλώσσα assembly που χρησιμοποιούμε στους μικροελεγκτές. Η STL (σχήμα 2.2) αποτελεί και την πιο ισχυρή γλώσσα γιατί μπορούμε εκμεταλλευτούμε πλήρως τις δυνατότητες των PLC.

```
LD      I0.0
A       I0.1
ON      I0.2
=       Q0.0
```

Σχήμα 2.2 : Η γλώσσα STL

2.1.3 Η γλώσσα Function Block (FDB)

Η FDB (σχήμα 2.3) χρησιμοποιείται πιο σπάνια και δεν υποστηρίζεται από όλους τους τύπους των PLC. Αποτελεί επίσης μια γραφική γλώσσα με τη διαφορά ότι αναπαριστά το λογικό κύκλωμα του αυτοματισμού χρησιμοποιώντας λογικές πύλες.



Σχήμα 2.3 : Η γλώσσα FDB

Κάνοντας μια γρήγορη σύγκριση μεταξύ των γλωσσών, θα μπορούσαμε να πούμε πως κάθε μία έχει τα πλεονεκτήματά της και τα μειονεκτήματά της. Ενδεικτικά κάποια πλεονεκτήματα όσον αφορά την STL είναι τα εξής :

- Χρειάζεται μικρότερη μνήμη για την αποθήκευση του προγράμματος
- Η εκτέλεση της γίνεται η μία εντολή μετά την άλλη κάτι που μας επιτρέπει να γνωρίζουμε με ακρίβεια σε ποιο σημείο βρίσκεται το πρόγραμμά μας
- Αρκετές φορές η χρήση της για την υλοποίηση κάποιων ρουτινών του κώδικα μας “λύνει” τα χέρια διότι κερδίζουμε αρκετό χρόνο και κόπο σε σχέση με τη χρήση των γραφικών συμβόλων.

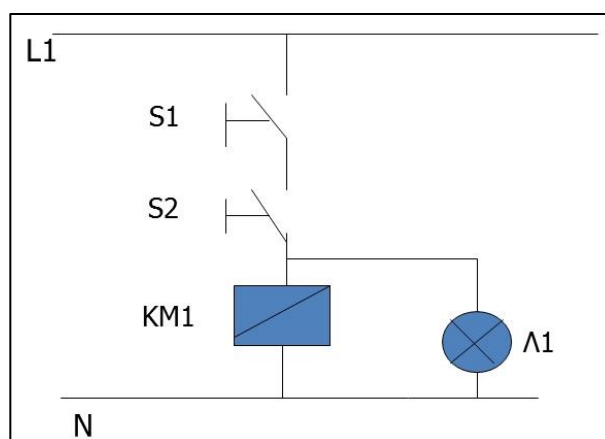
Το μειονέκτημα, όμως της συγκεκριμένης γλώσσας, είναι ότι σε καμία περίπτωση δεν παρέχει την ίδια δυνατότητα ελέγχου και εποπτείας που μας παρέχουν οι γραφικές γλώσσες σε online έλεγχο της εφαρμογής. Σίγουρα

αποτελεί την πιο απαιτητική γλώσσα και απαιτεί βασικές γνώσεις και καλύτερη αντίληψη σε ζητήματα προγραμματισμού από τον προγραμματιστή.

Στην οικογένεια ελεγκτών της Siemens υποστηρίζονται και οι τρεις γλώσσες και χαρακτηριστικό αποτελεί ότι μπορεί στην ίδια εφαρμογή να χρησιμοποιηθούν ταυτόχρονα και οι 3 γλώσσες ανάλογα με αυτό που μας είναι πιο βολικό.

2.2 Η γλώσσα LADDER

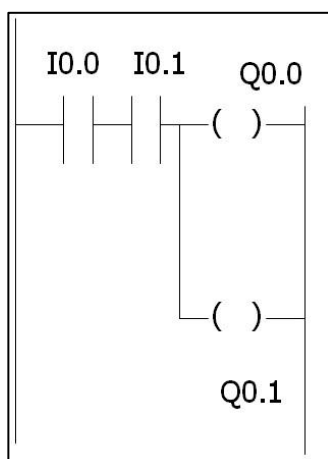
Όπως προαναφέραμε η γλώσσα LADDER, θυμίζει πάρα πολύ το ηλεκτρολογικό σχέδιο. Στην εικόνα 2.3 φαίνεται ένα απλό ηλεκτρολογικό σχέδιο :



Εικόνα 2.3 : Ηλεκτρολογικό σχέδιο

Δίνοντας μια απλή περιγραφή για το παραπάνω κύκλωμα, παρατηρούμε ότι αποτελείται από δύο οριζόντιες γραμμές, την L που αποτελεί το υψηλό δυναμικό και τη M που αποτελεί το αρνητικό δυναμικό της τροφοδοσίας. Στη συνέχεια βλέπουμε δύο διακόπτες Normal Open συνδεδεμένους σε σειρά, S1 και S2 και έναν ηλεκτρονόμο KM1 που είναι συνδεδεμένος παράλληλα με την ενδεικτική λυχνία Λ1. Για να σπλίσει ο ηλεκτρονόμος και να ανάψει η λυχνία Λ1 θα πρέπει να κλείσουν και οι δυο διακόπτες, έτσι ώστε το θετικό δυναμικό να περάσει στο άλλο άκρο των φορτίων.

Αν θέλαμε να υλοποιήσουμε το κύκλωμα της εικόνας 2.3 με PLC προγραμματίζοντας σε γλώσσα LADDER ο κώδικας μας θα ήταν όπως στην εικόνα 2.4 :

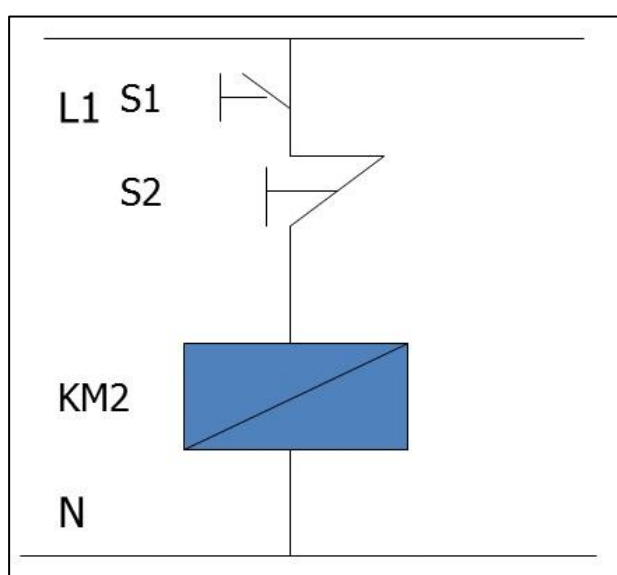


Εικόνα 2.4

Οι διακόπτες S1 και S2 σε αυτήν την περίπτωση θα ήταν συνδεδεμένοι στις εισόδους I0.0 και I0.1 και τα φορτία KM1 και Λ1 στις εξόδους του PLC Q0.0 και Q0.1 αντίστοιχα.

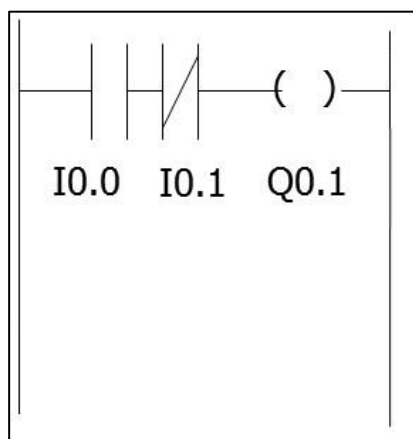
Κάνοντας αντιπαράθεση με το αντίστοιχο ηλεκτρολογικό σχέδιο παρατηρούμε ότι με τη γλώσσα LADDER ο σχεδιασμός γίνεται οριζόντια. Δηλαδή έχουμε δύο κατακόρυφες παράλληλες μπάρες, όπου η αριστερή αποτελεί το θετικό δυναμικό και η αριστερή το χαμηλό δυναμικό της τροφοδοσίας. Και μεταξύ αυτών των 2 γραμμών γίνεται ο σχεδιασμός των κλάδων του κυκλώματος. Κάθε κλάδος του διαγράμματος LADDER που ξεκινάει από την αριστερή και καταλήγει στη δεξιά αποτελεί μία γραμμή προγράμματος.

Στο επόμενο παράδειγμα έχουμε πάλι δύο διακόπτες σε σειρά, μόνο που ο ένας είναι Normal Close επαφής. Συνεπώς το αντίστοιχο ηλεκτρολογικό σχέδιο θα ήταν όπως αυτό της εικόνας 2.5 :



Εικόνα 2.5 : Ηλεκτρολογικό κύκλωμα με NC διακόπτη

Στο συγκεκριμένο κύκλωμα, για να ενεργοποιηθεί το φορτίο KM2 αρκεί να κλείσει μόνο ο διακόπτης S1 και να παραμείνει στη φυσική του κατάσταση ο S2. Το αντίστοιχο πρόγραμμα σε γλώσσα LADDER φαίνεται στην εικόνα 2.6 C



Εικόνα 2.6 : Πρόγραμμα σε γλώσσα LADDER με NC διακόπτη

Και σε αυτήν την περίπτωση οι διακόπτες S1 και S2 είναι συνδεδεμένοι στις εισόδους I0.0 και I0.1 και το φορτίο KM2 στην έξοδο Q0.1 του PLC αντιστοιχώς.

Στα παραπάνω παραδείγματα, είδαμε πως τροποποιείται ένα ηλεκτρολογικό σχέδιο στη γλώσσα προγραμματισμού LADDER. Η γλώσσα LADDER μας δίνει αρκετές δυνατότητες και ένα ισχυρό σετ εντολών, με τις οποίες μπορούμε να υλοποιήσουμε όλες μας τις εφαρμογές.

Οι πιο βασικές εντολές που συναντάμε θα μπορούσαμε να τις χωρίσουμε στις εξής κατηγορίες :

- Εντολές λογικής bit (bit logic)
- Εντολές σύγκρισης (Compare)
- Εντολές μετρητών (Counters)
- Εντολές χρονικών (Timers)

2.2.1 Εντολές λογικής bit (bit logic)

Από το όνομα τους και μόνο μπορεί να γίνει αντιληπτό ότι αναφέρονται σε εντολές δύο καταστάσεων, ή λογικού 0 ή λογικού 1. Πιο συγκεκριμένα διακρίνουμε τις εξής :

- **---| |--- Κανονικά ανοιχτή / Normally Open (NO)**
Η επαφή NO είναι ανοιχτή, δηλαδή δεν επιτρέπει τη διέλευση ρεύματος από το ένα άκρο στο άλλο. Διέλευση του ρεύματος θα έχουμε όταν η τιμή του bit, που αντιστοιχεί στη διεύθυνση που του έχουμε ορίσει γίνει 1. Για παράδειγμα όταν πατηθεί ένα μπουτόν που βρίσκεται συνδεδεμένο στην είσοδο του PLC ή αν έχει ενεργοποιηθεί ένα χρονικό.
- **---| / |--- Κανονικά κλειστή / Normally Closed (NC)**
Η επαφή NC είναι κλειστή, δηλαδή επιτρέπει τη διέλευση ρεύματος από το ένα άκρο στο άλλο, όταν η τιμή του bit στη διεύθυνση που αντιστοιχεί είναι 0. Για παράδειγμα αν δεν πατηθεί ένα μπουτόν, όπως το Emergency Stop, που είναι κλειστής επαφής και βρίσκεται συνδεδεμένο στην είσοδο του PLC.

Στην αντίθετη περίπτωση, αν για παράδειγμα πατηθεί το συγκεκριμένο κουμπί επαφής NC, τότε ανοίγει και δεν επιτρέπει τη διέλευση ρεύματος.

- **--|NOT|-- Αναστροφή / Invert Power Flow**
Με την εντολή NOT γίνεται αλλαγή της κατάστασης της ροής του ρεύματος. Δηλαδή, αν στο ένα άκρο έχουμε ρεύμα τότε με τη NOT δεν επιτρέπεται η διέλευση του ρεύματος. Ενώ όταν δεν φτάνει το ρεύμα στο ένα άκρο της από το άλλο υπάρχει ρεύμα.
- **--|P|-- , --|N|-- Εντολές θετικής, αρνητικής μετάβασης**
Η εντολή θετικής μετάβασης, επιτρέπει τη διέλευση του ρεύματος για έναν κύκλο σάρωσης όταν θα συμβεί μια μεταβολή από 0 σε 1 ενώ η αρνητική μετάβαση επιτρέπει τη διέλευση για έναν κύκλο όταν θα υπάρχει μεταβολή από 1 σε 0.

- **---() Εντολές εξόδου / Output Coil**

Η εντολές εξόδου λειτουργούν όπως τα πηνία στον κλασικό αυτοματισμό. Δηλαδή, αν υπάρχει ρεύμα στο άκρο του πηνίου τότε ενεργοποιείται η αντίστοιχη έξοδος ή το βοηθητικό.

Μια πολύ σημαντική λεπτομέρεια είναι ότι όταν χρησιμοποιούμε την συγκεκριμένη εντολή, θα πρέπει να τοποθετείται τέρμα δεξιά στο δικτύωμα του κλάδου του προγράμματος.

- **---(S), ---(R) Εντολή ενεργοποίησης / Απενεργοποίησης Set, Reset Coil**

Όταν εκτελούνται οι εντολές SET / RESET τότε οι αντίστοιχες παράμετροι-έξοδοι ενεργοποιούνται και γίνεται αυτοσυγκράτηση. Με την έννοια αυτοσυγκράτηση εννοούμε ότι για να ενεργοποιηθεί το φορτίο αρκεί να πατηθεί μια φορά το μπουτόν και χωρίς να χρειάζεται να είναι συνέχεια πατημένο.

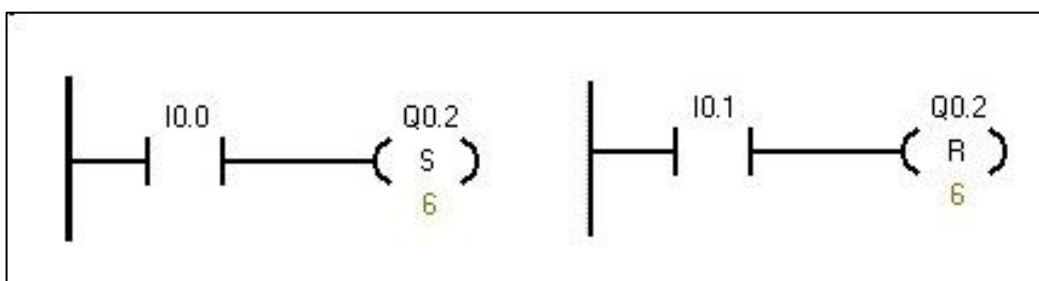
Ειδικά για τα s7 – 200 η σύνταξη το συγκεκριμένων εντολών έχει την ιδιομορφία να μπορούμε να ενεργοποιούμε ή να απενεργοποιούμε περισσότερες εξόδους – bits παραμέτρων με μία εντολή. Στο παράδειγμα που ακολουθεί μπορεί να γίνει κατανοητό :

Έστω ότι έχουμε των κώδικα της εικόνας 2.7 :

Μόλις κλείσει η επαφή I0.0 τότε θα ενεργοποιηθούν ταυτόχρονα 6 έξοδοι του PLC ξεκινώντας από την Q0.2. Δηλαδή οι έξοδοι Q0.2, Q0.3, Q0.4, Q0.5 Q0.6 και Q0.7.

Το ίδιο σαφώς και ισχύει και την περίπτωση της Reset εντολής, όπου μόλις θα κλείσει η επαφή I0.1 τότε θα απενεργοποιηθούν οι αντίστοιχες έξοδοι.

Εκτός από τις φυσικές εξόδους, οι εντολές SET και RESET, χρησιμοποιούνται για τα βοηθητικά, τα χρονικά και τους μετρητές.



Εικόνα 2.7 : Εντολή SET – RESET

- **SET – RESET FLIP FLOP**

Η συγκεκριμένη εντολή χρησιμοποιείται σαν στοιχείο μνήμης. Δηλαδή αρκεί να ενεργοποιηθεί μια φορά η είσοδος S και η έξοδος θα παραμείνει ενεργοποιημένη μέχρι να η είσοδος R.

Ένα πολύ χαρακτηριστικό παράδειγμα είναι η λειτουργία Start, Stop ενός κινητήρα. Έστω συνδέουμε δυο μπουτόν σε δύο διαφορετικές εισόδους του PLC και επιθυμούμε πατώντας και αφήνοντας το πρώτο μπουτόν ο κινητήρας να δουλεύει μέχρι να πατηθεί το δεύτερο. Το πρόγραμμα της συγκεκριμένης λειτουργίας φαίνεται στην εικόνα 2.8

Αν λοιπόν κλείσει η επαφή I0.1 (πατηθεί το αντίστοιχο μπουτόν) τότε ενεργοποιείται το βοηθητικό M0.0 και κατ' επέκταση η έξοδος Q0.0. Αν κλείσει όμως η I0.2 (το δεύτερο μπουτόν) το απενεργοποιείται το χρονικό, οπότε και η έξοδος Q0.0

• **RESET - SET FLIP FLOP**

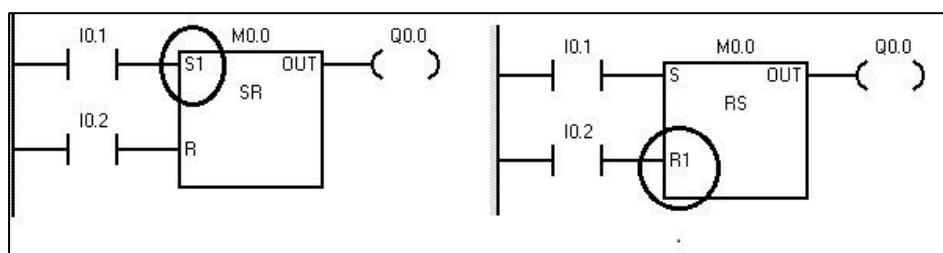
Η συγκεκριμένη εντολή λειτουργεί όπως ακριβώς η προηγούμενη με τη διαφορά ότι η προτεραιότητα είναι στην είσοδο RESET. Δηλαδή, αν είναι ταυτόχρονα και οι δυο εισοδοι του flip flop ενεργοποιημένες προηγείται η RESET οπότε δεν θα ενεργοποιηθεί καμία έξοδος. Στους πίνακες 2.1 και 2.2 φαίνονται οι πίνακες αληθείας με την ακριβή λειτουργία και για τα δύο flip flops.

ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ SET - RESET FLIP FLOP		
S1	R	OUT
0	0	Προηγούμενη τιμή
0	1	0
1	0	1
1	1	1

Πίνακας 2.1

ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ RESET - SET FLIP FLOP		
S	R1	OUT
0	0	Προηγούμενη τιμή
0	1	0
1	0	1
1	1	0

Πίνακας 2.2



Εικόνα 2.8 : Εντολές SET – RESET και RESET - SET

2.2.2 Εντολές σύγκρισης (COMPARE)

Στη συγκεκριμένη κατηγορία έχουμε να κάνουμε με εντολές με τις οποίες συγκρίνουμε αριθμούς, όπου αν ισχύει ο τελεστής της σύγκρισης τότε ενεργοποιείται και η αντίστοιχη επαφή. Συγκεκριμένα για τη CPU 212 οι συγκρίσεις που μπορούμε να κάνουμε είναι μεταξύ bytes, word integer και double integer.

- $$\begin{array}{ccc} n1 & n1 & n1 \\ --|==B|--, & --|<=B|--, & --|>=B|-- \\ n2 & n2 & n2 \end{array}$$
 , **Compare Byte**

Με τη συγκεκριμένη εντολή, συγκρίνουμε δύο bytes τα n1 και n2 και ανάλογα με το αν ισχύει η ισότητα ή ανισότητα η επαφή γίνεται 1 οπότε θα έχουμε τη ροή ρεύματος.

Σαν τελεστές σύγκρισης μπορούμε να χρησιμοποιήσουμε VB, IB, QB, MB ή σταθερό αριθμό.

- $$\begin{array}{ccc} n1 & n1 & n1 \\ --|==I|--, & --|<=I|--, & --|>=I|-- \\ n2 & n2 & n2 \end{array}$$
 , **Compare Word Integer**

Με τη συγκεκριμένη εντολή, συγκρίνουμε δύο λέξεις n1 και n2 και ανάλογα με το αν ισχύει η ισότητα ή ανισότητα η επαφή γίνεται 1 οπότε θα έχουμε τη ροή ρεύματος.

Σαν τελεστές σύγκρισης μπορούμε να χρησιμοποιήσουμε VB, IB, QB, MB, T, C ή σταθερό.

- $$\begin{array}{ccc} n1 & n1 & n1 \\ --|==D|--, & --|<=D|--, & --|>=D|-- \\ n2 & n2 & n2 \end{array}$$
 , **Compare Double Word Integer**

Ομοίως με τα παραπάνω ισχύουν και σε αυτή την περίπτωση με τη διαφορά ότι εδώ μπορούμε η σύγκριση γίνεται με αριθμούς μεγέθους Double Word

Σαν τελεστές σύγκρισης μπορούμε να χρησιμοποιήσουμε VD, ID, QD, MD.

2.2.3 Εντολές χρονικών - μετρητών

Η λειτουργία των χρονικών και των μετρητών περιγράφεται αναλυτικά στο κεφάλαιο 3.3.1

2.3 Συσσκευές προγραμματισμού

Για να προγραμματίσουμε ένα PLC χρειαζόμαστε το μέσο στο οποίο θα γράψουμε τον κώδικα μας και το οποίο είναι ικανό να το μεταφέρει και να το μεταφράσει στην κατάλληλη γλώσσα που κατανοεί το PLC. Με λίγα λόγια αποτελεί το μέσο επικοινωνίας του PLC με τον προγραμματιστή. Για να πετύχουμε αυτήν την επικοινωνία έχουμε δύο συσκευές. :

- **Ηλεκτρονικός υπολογιστής**

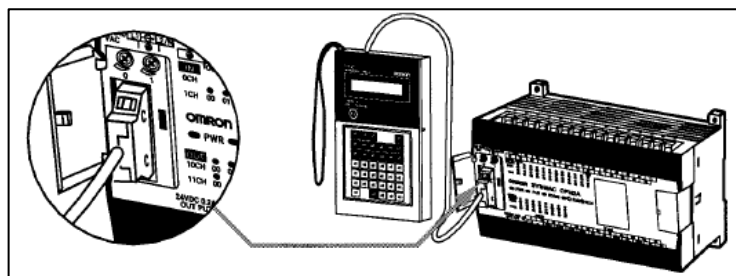
Ο Η / Υ που είναι και το πλέον χρησιμοποιούμενο μέσο μπορεί να είναι είτε σταθερός, όπου και η όλη διαδικασία μπορεί να γίνει σε περιβάλλον γραφείου, είτε φορητός με τον οποίο μεταφέρουμε το πρόγραμμα στο χώρο που βρίσκεται το PLC και το προγραμματίζουμε εκείνη τη στιγμή. Εκτός όμως από τον προγραμματισμό μπορούμε να εκτελέσουμε και κάποιες διαγνωστικές λειτουργίες για να εντοπίσουμε κάποια βλάβη ή να παρατηρούμε την ροή του προγράμματος ταυτόχρονα με την εκτέλεση της εφαρμογής.



Σχήμα 2.9 : Laptop για προγραμματισμό PLC

- **Φορητές συσκευές προγραμματισμού :**

Οι φορητές συσκευές χρησιμοποιήθηκαν τα παλαιότερα χρόνια και κυρίως εξυπηρετούσαν σε μικρά προγράμματα αλλά και σε μικρές τροποποιήσεις, όπως για παράδειγμα για αλλαγή σε κάποια τιμή χρονικού.



Σχήμα 2.10 : Φορητή συσκευή προγραμματισμού

ΚΕΦΑΛΑΙΟ 3 – ΤΑ PLC ΤΗΣ ΕΤΑΙΡΕΙΑΣ SIEMENS

3.1 Η οικογένεια Simatic - S7

Ανάλογα, λοιπόν, με τις απαιτήσεις του αυτοματισμού που καλούμαστε να υλοποιήσουμε η εταιρεία Siemens μας δίνει τη δυνατότητα να επιλέξουμε μεταξύ 3 διαφορετικών τύπων PLC. Αν και οι βασικοί τύποι είναι τρεις, ωστόσο κάθε τύπος περιλαμβάνει μια πληθώρα από διαφορετικά χαρακτηριστικά και δυνατότητες (CPU, αριθμό εισόδων / εξόδων κλπ) έτσι ώστε να επιλέξουμε το καταλληλότερο λαμβάνοντας υπόψη το συνολικό κόστος και τα επιμέρους στοιχεία που θα χρησιμοποιηθούν.

3.1.1 Simatic S7 - 200

Το S7-200 (εικόνα 3.1) αποτελεί το μικρότερο μέλος της οικογένειας Simatic - S7 και η χρήση του προορίζεται κυρίως για μικρές αυτόνομες εφαρμογές όπως ανελκυστήρες, πλυντήρια αυτοκινήτων. Βασικό χαρακτηριστικό τους, όπως και για κάθε τύπο PLC, αποτελεί η δυνατότητα να προσθέσουμε επιπλέον μονάδες εισόδων και εξόδων και έτσι να χρησιμοποιηθούν σε αυτοματισμούς υψηλότερων απαιτήσεων και σε σύνθετες διεργασίες όπως μηχανές ή γραμμές παραγωγής εμφιάλωσης, συσκευασίας.

Τα κυριότερα χαρακτηριστικά του είναι :

- Ταχύτητα
- Ευελιξία
- Αποκρίσεις σε πραγματικό χρόνο
- Σύνδεση συσκευών απεικόνισης και χειρισμών
- Δικτύωση
- Επεκτασιμότητα, πέραν των ενσωματωμένα
- Δυνατότητα επεξεργασίας αναλογικών σημάτων
- Σύνδεση μονάδων επέκτασης σε δύο σειρές
- Πάρα πολύ μικρές διαστάσεις
- Αναλογικά ποτενσιόμετρα για εύκολη τοποθέτηση set point
- Βρόγχοι ελέγχου με PID
- Πακέτο προγραμματισμού MicroWin με δυνατότητα προγραμματισμού σε όλες τις γλώσσες των PLC



Εικόνα 3.1 : PLC S7-200 – Το PLC της εφαρμογής μας

3.1.2 Simatic S7 - 300

Το αμέσως μεγαλύτερο μέλος της οικογένειας, είναι το S7 – 300 (εικόνα 3.2) το οποίο χρησιμοποιείται για εφαρμογές μεσαίας εμβέλειας στις οποίες συγκαταλέγονται και οι περισσότερες ελληνικές βιομηχανίες. Διατίθεται σε μορφή Compact και modular και είναι επίσης επεκτάσιμο. Τα κυριότερα χαρακτηριστικά του είναι τα εξής :

- Μεγάλη ποικιλία από CPU για τη βέλτιστη επιλογή ανάλογα με την επιθυμητή απόδοση
- Επεκτασιμότητα με έως 32 κάρτες.
- Δικτυώνεται με όλα τα πρότυπα δίκτυα (Profibus, Industrial Ethernet)
- Δεν έχει περιορισμό για τη θέση των επιμέρους καρτών
- Δεν υπάρχουν μικροδιακόπτες για τη παραμετροποίηση – όλα γίνονται μέσω λογισμικού
- Έχει πλήρες 32-bit σετ εντολών (ακόμα και για ημίτονο, συνημίτονο, λογάριθμο, τετραγωνική ρίζα).
- Ενσωματωμένη δυνατότητα δικτύωσης (MPI) στη κεντρική μονάδα επεξεργασίας.
- Ενσωματωμένες δυνατότητες διασύνδεσης με HMI - δεν απαιτείται προγραμματισμός
- Μνήμη διαγνωστικών - αυτόματη αποθήκευση με χρόνο και ημερομηνία όλων των συμβάντων στο PLC
- Μια μόνο κάρτα για όλους τους τύπους αναλογικών – η επιλογή γίνεται μέσω του λογισμικού



Εικόνα 3.2 : PLC S7-300

3.1.3 Simatic S7 - 400

Το S7-400 (εικόνα 3.3) αποτελεί τον πλέον «ισχυρό» ελεγκτή της Siemens για εφαρμογές υψηλών απαιτήσεων σε αριθμό σημάτων, χρόνους επεξεργασίας μέγεθος προγράμματος και επικοινωνίες. Χρησιμοποιείται σε πολύ απαιτητικές εφαρμογές και το κόστος του είναι αρκετά υψηλό.

Σαν κύρια χαρακτηριστικά διαθέτει ότι και η σειρά S7-300 και επιπλέον :

- Πολύ μεγάλο αριθμό σημάτων (πάνω από 130.000 ψηφιακά και 8.000 αναλογικά)
- Πολύ μεγάλες μνήμες (πάνω από 8 MB)
- Ταυτόχρονη χρήση μέχρι και 4 κεντρικών μονάδων επεξεργασίας (CPU)
- Ελεύθερη τοποθέτηση των καρτών, ακόμα και των CPU
- Δυνατότητα αφαίρεσης των καρτών ακόμα και κατά τη διάρκεια λειτουργίας του συστήματος χωρίς πρόβλημα



Εικόνα 3.3 : PLC S7-400

Η επιλογή μεταξύ οικογενειών συστημάτων (S7 - 200/ S7 - 300/ S7 - 400) σχετίζεται με τις απαιτήσεις και την πολυπλοκότητα της εφαρμογής καθώς και με τις προοπτικές πιθανών μελλοντικών επεκτάσεων. Αξίζει να σημειωθεί πως τα S7 – 300 και S7 – 400 έχουν διαφορετικό περιβάλλον προγραμματισμού το Step7 - Simatic Manager ενώ το περιβάλλον προγραμματισμού S7 - 200 λέγεται Step7 – Microwin.

Επίσης, όπως προαναφέραμε, κάθε σειρά περιλαμβάνει και αρκετές εκδόσεις CPU, όπου η κάθε μία έχει και τις ανάλογες δυνατότητες εκτέλεσης προγράμματος και επεξεργασίας σημάτων. Ακόμη, ανάλογα με την έκδοση της CPU υπάρχουν και διαφορές στην κεντρική τροφοδοσία των PLC (230 Volt AC ή 24 Volt DC) καθώς και ποικιλία στις εισόδους και εξόδους, όσον αφορά το πλήθος τους και την τάση που μπορούν να αναγνωρίσουν σαν είσοδο και να τροφοδοτήσουν σε επιθυμητά φορτία.

Ενδεικτικά, αναφέρουμε τον παρακάτω πίνακα με τα διάφορα χαρακτηριστικά της σειράς S7-200 και των αντίστοιχων CPU :

CPU 212	DC Power Supply, DC Inputs, DC Outputs
CPU 214	AC Power Supply, DC Inputs, Relay Outputs
CPU 216	DC Power Supply, DC Inputs, DC Outputs
CPU 221	DC Power Supply, DC Inputs, Relay Outputs

Πίνακας 3.1 : Χαρακτηριστικά των CPU της σειράς S7-200

Τα χαρακτηριστικά του S7-200 που χρησιμοποιήθηκε στην πτυχιακή εργασία, περιγράφονται αναλυτικότερα στην επόμενο ενότητα.

3.2 PLC S7-200

Όπως έχει ήδη αναφερθεί, η σειρά S7 – 200, είναι η “μικρή” σειρά των PLC με κύρια χαρακτηριστικά της compact κατασκευής, χαμηλό κόστος και ένα ισχυρό σετ εντολών που αδιαμφισβήτητα αποτελεί μια αξιόλογη λύση για μικρές εφαρμογές αυτοματισμού. Υπάρχει πλήθος από διαφορετικά “μοντέλα” των S7 – 200 με διαφορετικές CPU με πλήθος εισόδων και εξόδων για να διαλέξουμε ανάλογα με την εφαρμογή μας. Στις επόμενες ενότητες θα περιοριστούμε στην περιγραφή μόνο της CPU 212, μιας και το PLC που χρησιμοποιήθηκε διαθέτει τη συγκεκριμένη CPU.

3.2.1 Χαρακτηρίστηκα του PLC με τη CPU-212

Στον πίνακα 2 που ακολουθεί, είναι συγκεντρωμένα τα κυριότερα χαρακτηριστικά του S7 – 200 με τη CPU 212 όπως τα δίνει η SIEMENS στα αντίστοιχα φύλλα δεδομένων :

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ	CPU 212
ΦΥΣΙΚΕΣ ΔΙΑΣΤΑΣΕΙΣ	160mm X 80mm X 82mm
ΒΑΡΟΣ	0.4 kg
ΜΝΗΜΗ	
ΠΡΟΓΡΑΜΜΑΤΟΣ	512 WORDS
ΔΕΔΟΜΕΝΑ ΧΡΗΣΤΗ	512 WORDS
ΕΣΩΤΕΡΙΚΑ BITS ΜΝΗΜΗΣ	128
ΕΙΣΟΔΟΙ / ΕΞΟΔΟΙ	
ΤΟΠΙΚΕΣ	8 ΕΙΣΟΔΟΙ / 6 ΕΞΟΔΟΙ
ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΕΞΟΔΩΝ	
ΤΥΠΟΣ ΕΞΟΔΟΥ	ΡΕΛΕ
ΤΑΣΗ	5-30 V DC / 250 V AC
ΧΡΟΝΙΚΑ	64
ΜΕΤΡΗΤΕΣ	64
ΤΡΟΦΟΔΟΣΙΑ	85 – 264 Volt AC
ΤΡΟΦΟΣΙΑ ΑΙΣΘΗΤΗΡΩΝ DC	20.4 – 28.8 Volt DC

Πινάκας 3.2 : Βασικά χαρακτηριστικά του S7-200 με CPU 212

3.2.2 Το εξωτερικό του PLC

Παρατηρώντας εξωτερικά το PLC (εικόνα 3.4), βλέπουμε τρεις στήλες με συνολικά 17 leds. Πιο συγκεκριμένα στην πρώτη στήλη υπάρχουν τρία leds με ονομασία SF, RUN και STOP:

- **Led SF (System Fault)**
Το συγκεκριμένο led ενεργοποιείται όταν συμβεί κάποιο κρίσιμο λάθος που δημιουργεί πρόβλημα κατά τη λειτουργία της CPU και οδηγεί το PLC σε κατάσταση STOP
- **Led RUN**
Το συγκεκριμένο Led είναι ενεργοποιημένο όταν ο διακόπτης στο εσωτερικό του PLC βρίσκεται στη θέση RUN ή έχουμε δώσει εντολή RUN μέσα από το περιβάλλον MicroWin και δηλώνει ότι το PLC βρίσκεται σε λειτουργία και το πρόγραμμα μας εκτελείται κανονικά
- **Led STOP**
Το Led STOP ενεργοποιείται όταν ο διακόπτης βρίσκεται στη θέση Stop ή το έχουμε δώσει εντολή Stop από το λογισμικό. Σε αυτήν την περίπτωση το πρόγραμμα μας σταματάει να εκτελείται όποτε ο αυτοματισμός δεν λειτουργεί

Η μεσαία στήλη των Leds αντιστοιχεί στις εισόδους. Κάθε φορά που μια είσοδος είναι ενεργοποιημένη, δηλαδή περνάνε τα 24 volt στην είσοδο του PLC ενεργοποιείται και το αντίστοιχο led.

Ομοίως και η τελευταία στήλη που αντιστοιχεί στις εξόδους του PLC, ενεργοποιείται το led όταν θα ενεργοποιηθεί η έξοδος ανάλογα με την κώδικα.



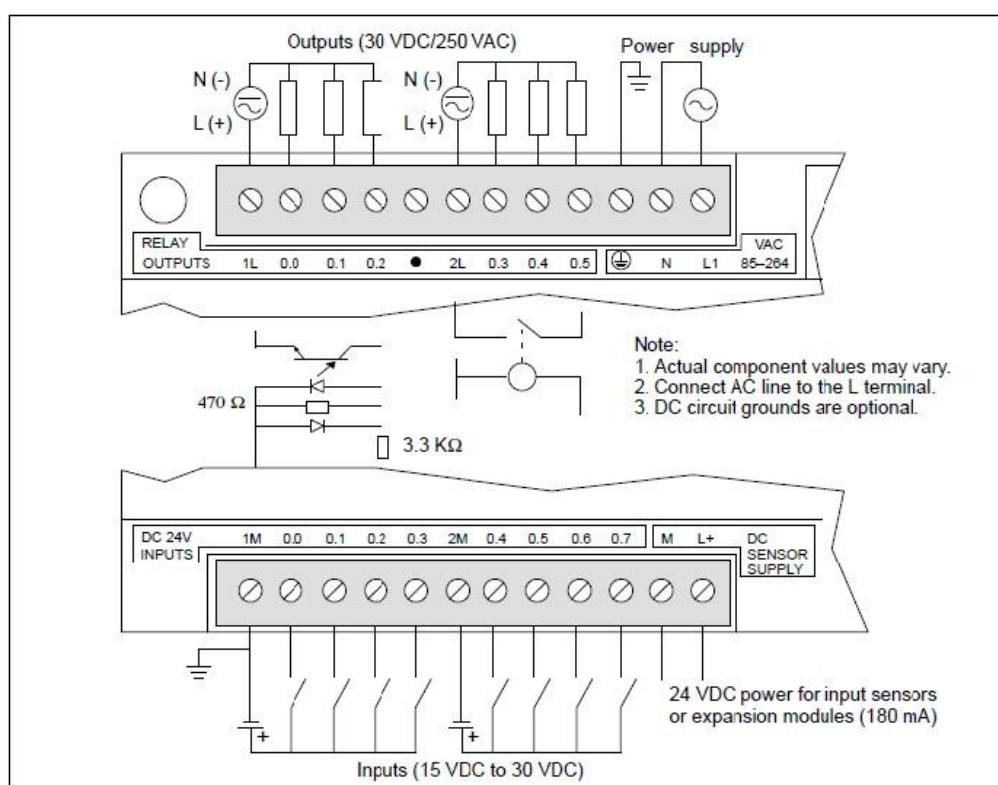
Εικόνα 3.4 : Το PLC εξωτερικά

3.2.3 Το εσωτερικό του PLC

Αν σηκώσουμε τα δύο προστατευτικά καπάκια του PLC της εφαρμογής μας, από κάτω βρίσκονται οι κλέμες στις οποίες πρέπει να γίνουν όλες οι καλωδιώσεις με τις εισόδους και εξόδους της εφαρμογής μας. Όπως φαίνεται και στην εικόνα 3.4 η κάτω πλευρά διαθέτει 8 αριθμημένες κλέμες από I0.0 έως I0.7 οι οποίες αποτελούν τις 8 εισόδους του PLC. Σε αυτές έχουμε συνδέσει όλα τα μπουτόνς και τους διακόπτες της εφαρμογής μας. Επίσης διακρίνουμε δύο κλέμες με την ονομασία DC SENSOR SUPPLY (+L, M). Από αυτές το PLC μας δίνει 24 Volt, το οποίο χρησιμοποιήθηκαν σαν τάση τροφοδοσίας των αισθητήρων. Τέλος, έχουμε άλλες 2 κλέμες με ονομασία 1M

και 2M οι οποίες αποτελούν τη γείωση για την τροφοδοσία των αισθητήρων μας.

Το πάνω κομμάτι της εικόνας αποτελείται από την κεντρική τροφοδοσία και την πλευρά των εξόδων του PLC. Στις τρεις δεξιά κλέμες είναι η τροφοδοσία που θα πάρουμε από το δίκτυο 230Volt AC. Οι κλέμες που αριθμούνται με αριθμούς από Q0.0 έως Q0.5 αποτελούν τις 6 ψηφιακές εξόδους του PLC οι οποίες είναι συνδεδεμένες με τα ρελέ που χρησιμοποιήσαμε για ενεργοποιήσουμε τα φορτία. Οι δύο κλέμες με ονομασία 1L και 2L αποτελούν την τάση που θέλουμε να τροφοδοτήσουμε τις εξόδους του PLC. Δηλαδή, αν ήταν επιθυμητό όταν θα ενεργοποιούνταν οι έξοδοι των PLC να παίρναμε τάση 230 Volt θα έπρεπε τις κλέμες 1L και 2L να τις συνδέσουμε σε αυτήν την τάση. Στην προκειμένη περίπτωση, συνδέθηκαν στα 24 Volt, όποτε οι έξοδοι μας θα μας δίνουν 24 volt κάθε φορά που θα ενεργοποιούνται.



Εικόνα 3.5 : Οι κλέμες της CPU - 212

Στο εσωτερικό του PLC βλέπουμε και έναν διακόπτη τριών θέσεων με τις ενδείξεις RUN, TERM και STOP (εικόνα 3.6). Με το συγκεκριμένο διακόπτη επιλέγουμε τη λειτουργία που θέλουμε να εκτελεί το PLC. Πιο συγκεκριμένα :

- **Θέση RUN**

Στη θέση αυτή το PLC βρίσκεται σε λειτουργία και το πρόγραμμα μας εκτελείται κανονικά. Το PLC εκτελεί τον κώδικα μας δέχεται εισόδους και ανάλογα ανταποκρίνεται και μεταβάλλει τις εξόδους.

- **Θέση TERM**
Στη θέση αυτή συνδέουμε τον υπολογιστή με το PLC. Μπορούμε να προγραμματίσουμε το PLC, να παρατηρήσουμε την λειτουργία και την ομαλότητα του κώδικα μας και να τοποθετήσουμε το PLC σε θέση RUN ή STOP.
- **Θέση STOP**
Σε αυτή τη θέση το πρόγραμμα σταματά να εκτελείται.



Εικόνα 3.6 : Διακόπτης επιλογής κατάστασης

3.3 Χαρακτηριστικά του S7 – 200

Σε προηγούμενη ενότητα αναφερθήκαμε στην ονοματολογία και σε χαρακτηριστικά που αφορούν τα PLC γενικότερα. Παρακάτω, ακολουθεί μια αντίστοιχη περιγραφή σε όλα αυτά τα στοιχεία, επικεντρωμένη στο S7 -200 με τη CPU – 212.

3.3.1 Ονοματολογία - Διευθυνσιοδότηση

Το PLC που χρησιμοποιήθηκε, διαθέτει 8 ψηφιακές εισόδους και 6 ψηφιακές εξόδους. Η ονοματολογία που χρησιμοποιείται είναι η εξής :

- **Ψηφιακές εισοδοι – Digital Inputs**
Για τις εισόδους χρησιμοποιούμε το γράμμα I και η αρίθμηση τους ξεκινάει από το I0.0 μέχρι I0.7
- **Ψηφιακές έξοδοι – Digital Outputs**
Οι έξοδοι χρησιμοποιούν το γράμμα Q και η αρίθμηση τους ξεκινάει από την Q0.0 μέχρι την Q0.5
- **Βοηθητικά – Memory Bits**
Τα βοηθητικά bits – επαφές χρησιμοποιούν το γράμμα M και η αρίθμηση τους ξεκινάει από το M0.0 μέχρι M15.7
- **Μεταβλητές μνήμης – Variable Memory**
Οι μεταβλητές μνήμης χρησιμοποιούν το γράμμα V και η αρίθμηση τους ξεκινάει από V0.0 μέχρι V1023.7

- **Χρονικά – Timers**

Για τα χρονικά χρησιμοποιείται το γράμμα T και η αρίθμηση τους ξεκινάει από T0 μέχρι T63. Όπως έχουμε αναφέρει τα χρονικά χρησιμοποιούνται σε εφαρμογές όπου απαιτείται κάποια χρονική καθυστέρηση αφού συμβεί ένα γεγονός. Για παράδειγμα αν πατηθεί ένα μπουτόν μετά από ένα διάστημα 5 δευτερολέπτων να ξεκινάει ένας κινητήρας.

Υπάρχουν δύο μεταβλητές που σχετίζονται με τα χρονικά :

- Η πρώτη είναι η τρέχουσα τιμή. Είναι ένας 16 bit αριθμός στον οποίο βρίσκεται αποθηκευμένη η τρέχουσα τιμή που έχει μετρήσει το χρονικό
- Η δεύτερη είναι το bit του χρονικού. Το bit αυτό γίνεται 1 όταν ο χρόνος που έχει μετρηθεί είναι ίσος ή μεγαλύτερος από την τιμή που του έχουμε ορίσει.

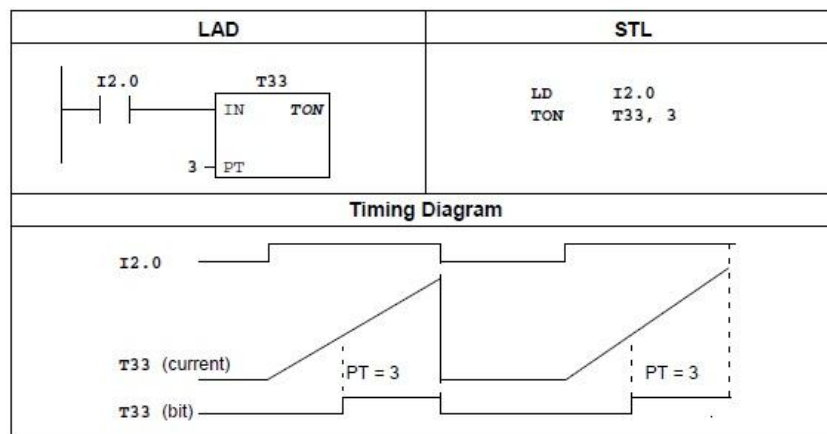
Η πρόσβαση και στις δύο μεταβλητές γίνεται με τη χρήση του γράμματος T και τον αριθμό του χρονικού που χρησιμοποιούμε.

Ειδικά για το S7 – 200 με τη CPU212 έχουμε δύο ειδών χρονικά (εικόνα 3.7) το ON Delay Timer (TON) και το Retentive On Delay (TONR).

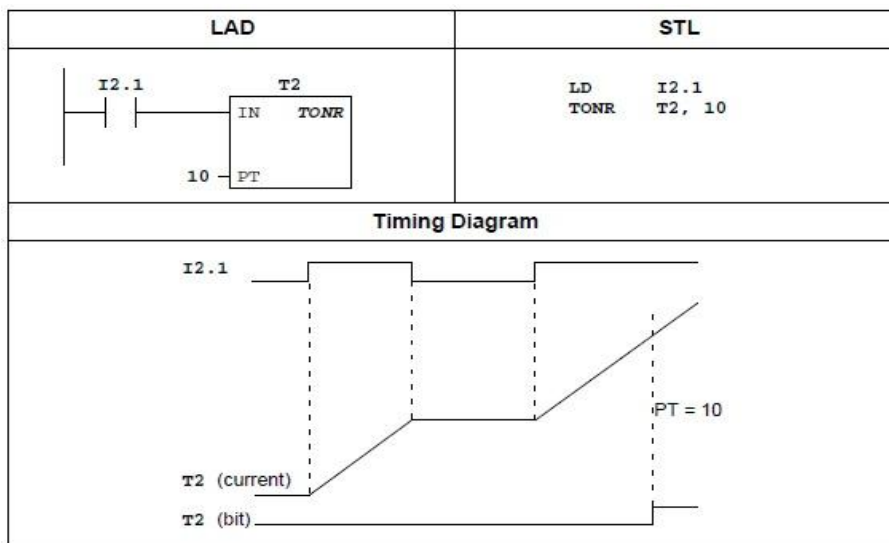


Εικόνα 3.7 : Τα χρονικά TON και TONR

Όσον αφορά το χρονικό TON, για να ξεκινήσει να μετρήσει θα πρέπει να ενεργοποιηθεί η είσοδος IN. Μόλις ενεργοποιηθεί και για όσο χρόνο είναι ενεργοποιημένη το χρονικό συνεχίζει να μετράει. Αν η είσοδος IN σταματήσει να είναι ενεργή τότε το χρονικό μηδενίζει. Σε αντίθεση το TONR όταν η είσοδος γίνει 0 κρατάει την τιμή που έχει μετρήσει και όταν ενεργοποιηθεί ξανά συνεχίζει να μετράει. Η λειτουργία των δύο χρονικών φαίνεται πιο καθαρά στις εικόνες 3.8α και 3.8β.



Εικόνα 3.8 α : Λειτουργία TON timer



Εικόνα 3.8 β : Λειτουργία TONR timer

Όταν ξεπεραστεί ο χρόνος με τον οποίο έχουμε προκαθορίσει το χρονικό ο χρόνος θα συνεχίσει να μετράει μέχρι τη μέγιστη τιμή που μπορεί να μετρήσει και εκεί θα σταματήσει.

Ένα ακόμη πολύ σημαντικό στοιχείο που πρέπει να αναφέρουμε, σχετικά με την ονοματολογία, είναι ότι το συγκεκριμένο PLC διαθέτει διαφορετική αρίθμηση στα χρονικά ανάλογα με το είδος αλλά και με την ανάλυση που θέλουμε να χρησιμοποιούμε. Συγκεκριμένα, όπως φαίνεται και από τον πίνακα 3 υπάρχουν τρεις διαφορετικές χρονικές αναλύσεις και για τα δύο είδη χρονικών και αναλόγως καθορίζεται και το όνομα του χρονικού που έχουμε τη δυνατότητα να χρησιμοποιήσουμε.

TIMER	ΑΝΑΛΥΣΗ	ΜΕΓΙΣΤΗ ΤΙΜΗ	CPU 212
TON	1ms	32.767 s	T32
	10ms	327.67 s	T33 - T36
	100ms	3276.7 s	T37 - T63
TONR	1ms	32.767 s	T0
	10ms	327.67 s	T1 - T4
	100ms	3276.7 s	T5 - T31

Πίνακας 3.3 : Χρονικές αναλύσεις και ονοματολογία χρονικών

Για παράδειγμα, έστω ότι θέλουμε να μετρήσουμε ένα χρόνο 200 ms. Όπως φαίνεται από τον παραπάνω πίνακα μπορούμε να το υλοποιήσουμε ως εξής :

- Με τη χρήση χρονικού ανάλυσης 10 ms, επιλέγουμε ένα όνομα από το διαθέσιμο εύρος π.χ. T33 και στην είσοδο PT θα πρέπει να δώσουμε την τιμή 20.

- Με τη χρήση χρονικού 100 ms, επιλέγουμε ένα όνομα από το διαθέσιμο εύρος π.χ. T35 και στην είσοδο PT θα πρέπει να δώσουμε την τιμή 2.

Ουσιαστικά η είσοδος PT των χρονικών σε συνδυασμό με την ανάλυση που επιλέγουμε λειτουργεί σαν πολλαπλασιαστής.

- **Απαριθμητές - Counters**

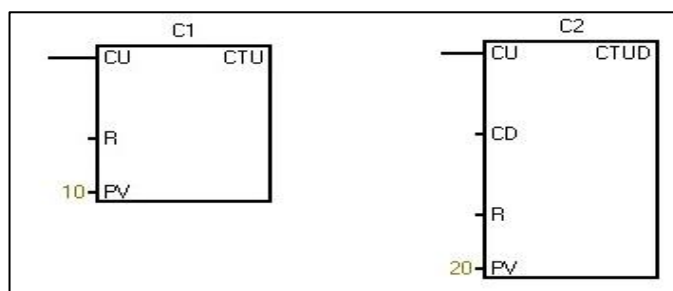
Για τους Counters Χρησιμοποιούμε το γράμμα C και η αρίθμηση τους ξεκινάει από το 0 μέχρι το 63. Τα χρονικά χρησιμοποιούνται για την καταμέτρηση προϊόντων ή να μετράμε πόσες φορές έχει συμβεί ένα γεγονός. Για παράδειγμα, αν σε μια διεργασία περνούσαν δέκα προϊόντα να κλείσει ένας κινητήρας που τροφοδοτεί τη γραμμή με επιπλέον προϊόντα.

Η χρήση των μετρητών γίνεται με τη χρήση του γράμματος C και τον αριθμό του μετρητή. Για κάθε μετρητή που χρησιμοποιούμε θα πρέπει να δίνουμε διαφορετικό αριθμό.

Όπως και για τα χρονικά έτσι και για τους μετρητές υπάρχουν δύο μεταβλητές που σχετίζονται με τη λειτουργία τους :

- Η πρώτη είναι η τρέχουσα τιμή, που είναι ουσιαστικά ένας 16 Bit αριθμός που βρίσκεται αποθηκευμένη η τιμή που έχει μετρηθεί από τον απαριθμητή.
- Και η δεύτερη μεταβλητή είναι το bit του μετρητή το οποίο ενεργοποιείται, μόλις η τρέχουσα τιμή του μετρητή ξεπεράσει την επιθυμητή τιμή που του έχουμε δώσει στην είσοδο PV (Preset Value)

Συγκεκριμένα η CPU 212 διαθέτει δύο ειδών μετρητών (εικόνα 3.9), τον Counter Up (CTU) και τον Counter Up / Down (CTUD)



Εικόνα 3.9 : Οι μετρητές – Counters της CPU – 212

Για τον CTU, προγραμματίζουμε την είσοδο PV με την επιθυμητή τιμή. Για να ξεκινήσει να μετράει ο μετρητής θα πρέπει η είσοδος CU να αλλάξει κατάσταση από 0 σε 1. Κάθε φορά που συμβαίνει αυτό ο μετρητής αυξάνει κατά 1 και όταν γίνει η τιμή γίνει ίση ή μεγαλύτερη από την τιμή PV τότε το bit του μετρητή γίνεται 1. Για να μηδενιστεί η τρέχουσα τιμή του μετρητή αρκεί να γίνει 1 η είσοδος R (Reset)

Για τον μετρητή CTUD ισχύει ότι και για τον CTU με τη διαφορά ότι διαθέτει άλλη μία είσοδο την CD η οποία κάθε φορά που ενεργοποιείται μειώνει το περιεχόμενο του counter κατά ένα.

3.3.2 Συμβολικός προγραμματισμός

Λόγω του πλήθους των εισόδων και εξόδων που συναντάμε στα PLC και λόγω τις ιδιαιτερότητας των ονομάτων που χρησιμοποιούν, μπορούμε να χρησιμοποιήσουμε συμβολικά ονόματα της επιλογής μας έτσι ώστε η εισαγωγή των επαφών και η αντιστοίχηση τους με τη σωστή ονομασία να γίνεται με πιο εύκολο τρόπο.

Για να γίνει αυτό, αρκεί να επιλέξουμε από το περιβάλλον MicroWin τη συντόμευση Symbol Table (εικόνα 3.10). Αμέσως μας εμφανίζεται ένας πίνακας (εικόνα 3.10) με τρεις στήλες στο οποίο αποδίδουμε τα συμβολικά ονόματα ως εξής :

- **1^η Στήλη - Symbol**

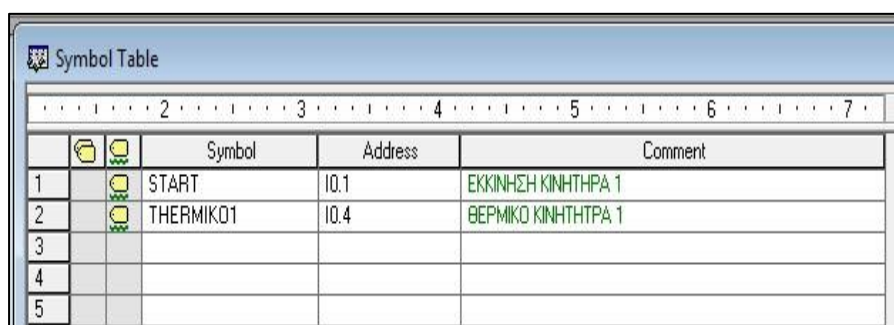
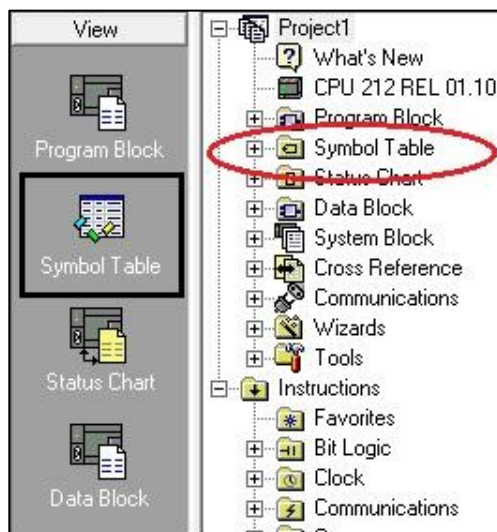
Επιθυμητό όνομα (π.χ Start, Stop, Emergency Stop)

- **2^η Στήλη - Address**

Διεύθυνση της αντίστοιχης εισόδου – εξόδου (π.χ I0.1, I0.0, Q0.4)

- **3^η Στήλη – Comment**

Η συγκεκριμένη στήλη είναι προαιρετική. Αν το επιθυμούμε γράφουμε κάποια σχόλια σχετικά με το αντίστοιχο στοιχείο για να διευκολυνθούμε.



	Symbol	Address	Comment
1	START	I0.1	ΕΚΚΙΝΗΣΗ ΚΙΝΗΤΗΡΑ 1
2	THERMIKO1	I0.4	ΘΕΡΜΙΚΟ ΚΙΝΗΤΗΡΑ 1
3			
4			
5			

Εικόνα 3.10 : Η επιλογή Symbol Table – Ο πίνακας συμβολικών ονομάτων

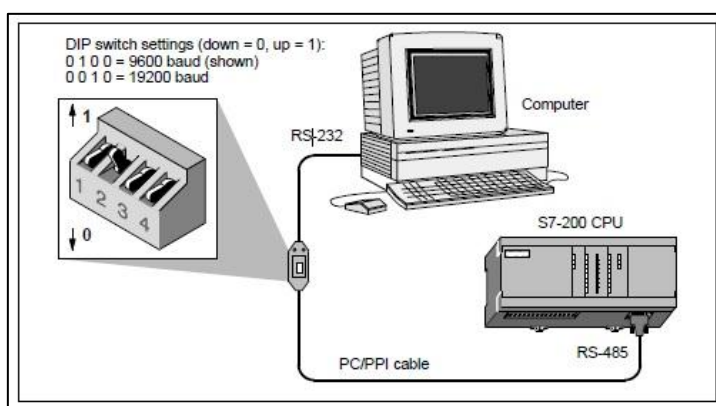
3.4 Εργαλεία προγραμματισμού για το S7 - 200

Για να προγραμματίσουμε όλα τα PLC S7 – 200 με οποιαδήποτε CPU χρειαζόμαστε δύο βασικά εργαλεία. Το πρώτο είναι το λογισμικό προγραμματισμού της συγκεκριμένης σειράς το οποίο ονομάζεται STEP 7 / MICROWIN (εικόνα 3.11) και είναι αποκλειστικό προϊόν της SIEMENS. Η έκδοση που χρησιμοποιήθηκε είναι η τέταρτη, η οποία είναι πλήρως συμβατή με Windows XP και μεταγενέστερη.



Εικόνα 3.11 : Η εφαρμογή STEP 7 / MICROWIN

Το δεύτερο εργαλείο που χρειαζόμαστε είναι το καλώδιο προγραμματισμού. Το συγκεκριμένο καλώδιο λέγεται PPI (εικόνα 3.12) και αποτελεί το μέσο με το οποίο συνδέουμε το υπολογιστή με το PLC μέσω μιας σειριακής θύρας ή μια θύρας USB ανάλογα ποια έκδοση διαθέτουμε. Αν παρατηρήσουμε το καλώδιο, θα δούμε στο σημείο που βρίσκεται το ηλεκτρονικό κύκλωμα υπάρχουν 4 μικροδιακόπτες. Αυτοί οι διακόπτες καθορίζουν το baud rate με το PLC. Για τη συγκεκριμένη CPU το baud rate πρέπει να είναι 9.6 Kbaud οπότε ρυθμίζουμε ανάλογα και τους διακόπτες.



Εικόνα 3.12 : Το καλώδιο προγραμματισμού PPI

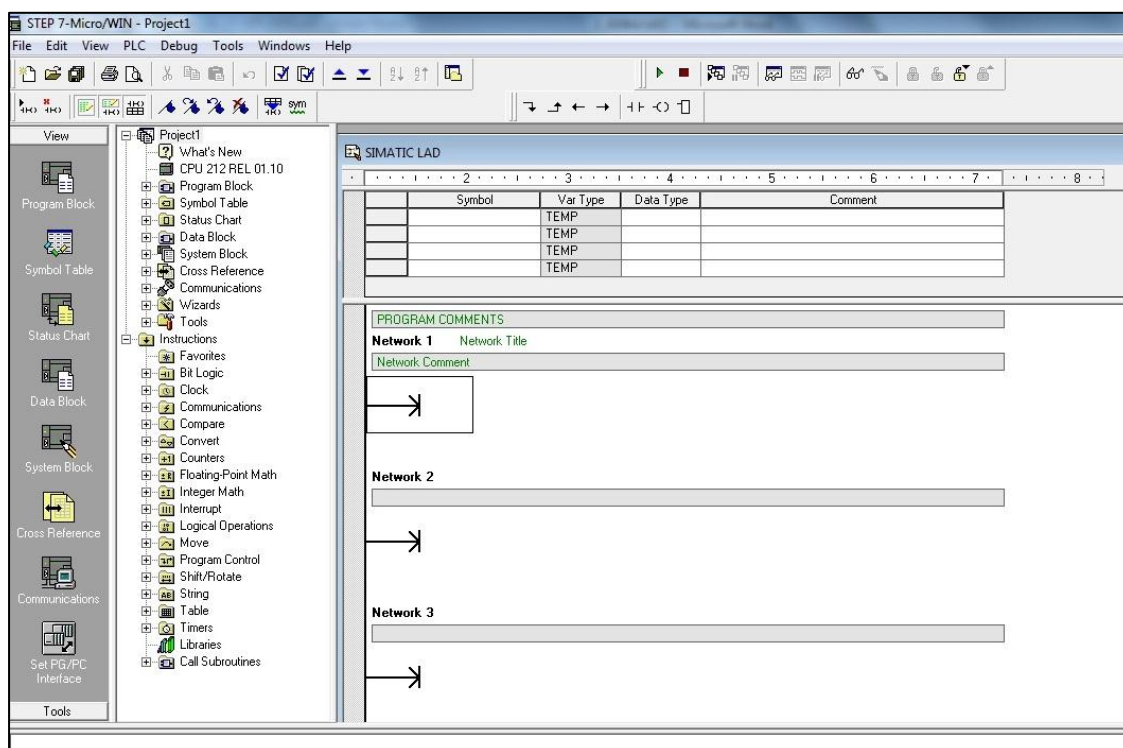
3.4.1 Το περιβάλλον Step 7 / MicroWin

Ανοίγοντας το λογισμικό MicroWin μπαίνουμε στο περιβάλλον όπως αυτό της εικόνας 3.13. Με μια πρώτη ματιά, γίνεται αντιληπτό ότι αποτελεί μια εφαρμογή που ακολουθεί τη φιλοσοφία των εφαρμογών που τρέχουν σε περιβάλλον Windows. Για τη διευκόλυνση μας, μας δίνεται η δυνατότητα αφήνοντας το κέρσορα από το ποντίκι πάνω σε κάποια επιλογή, αυτομάτως εμφανίζεται το όνομα αυτής της επιλογής.

Ενδεικτικά να αναφέρουμε ότι από την αριστερή πλευρά παρατηρούμε κάποιες συντομεύσεις επιλογών, με τις οποίες μπορούμε να κάνουμε αρκετά εύκολη την πλοήγηση μας στο περιβάλλον και να βρούμε εύκολα τις εντολές που θέλουμε να χρησιμοποιήσουμε.

Στην πάνω πλευρά, επίσης υπάρχουν κάποιες γνώστες συντομεύσεις, όπως η αποθήκευση, το άνοιγμα, εκτύπωση του project καθώς και οι επιλογές με τις οποίες “κατεβάζουμε” ή “ανεβάζουμε” ένα πρόγραμμα στο PLC. Επίσης υπάρχουν οι επιλογές με τις οποίες μπορούμε να θέσουμε το PLC σε κατάσταση RUN ή STOP. Σε κάθε περίπτωση πριν την οποιαδήποτε ενέργεια θα πρέπει να αφιερώσουμε λίγο χρόνο, έτσι ώστε να εξοικειωθούμε το περιβάλλον

Στο κέντρο της εφαρμογής βρίσκεται το πεδίο στο οποίο γράφουμε το πρόγραμμα μας. Στη συγκεκριμένη εικόνα φαίνεται το περιβάλλον στο οποίο συντάσσουμε με τη γραφική γλώσσα LADDER.

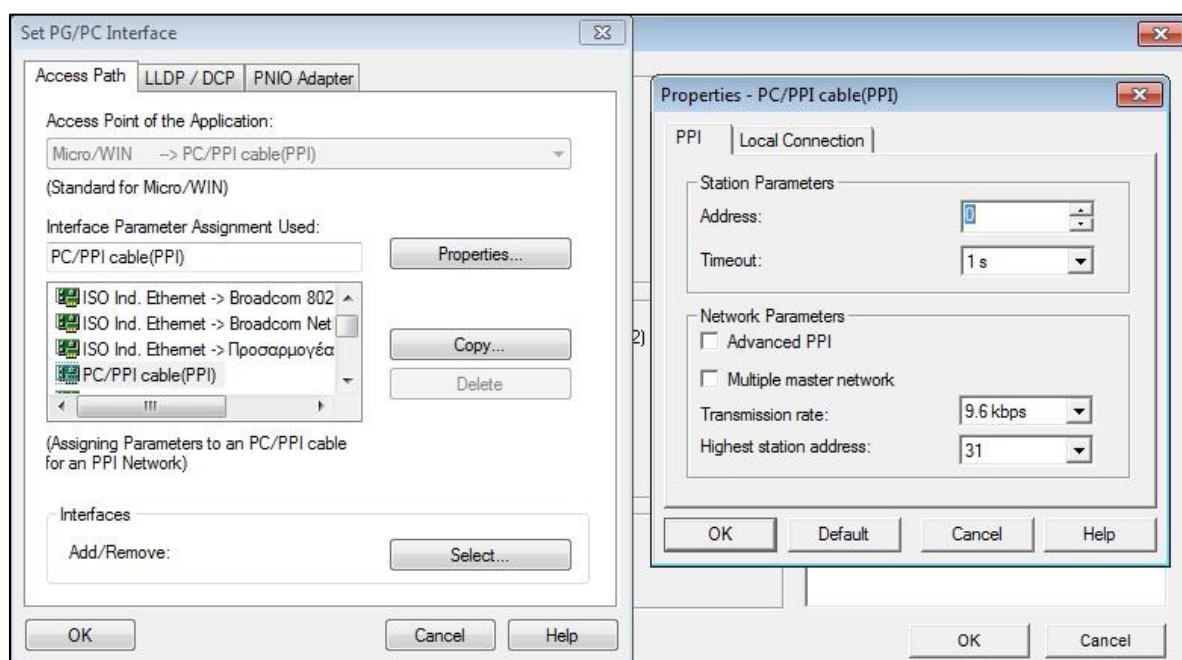


Εικόνα 3.13 : Το περιβάλλον προγραμματισμού MicroWin

3.4.2 Ρυθμίσεις επικοινωνίας

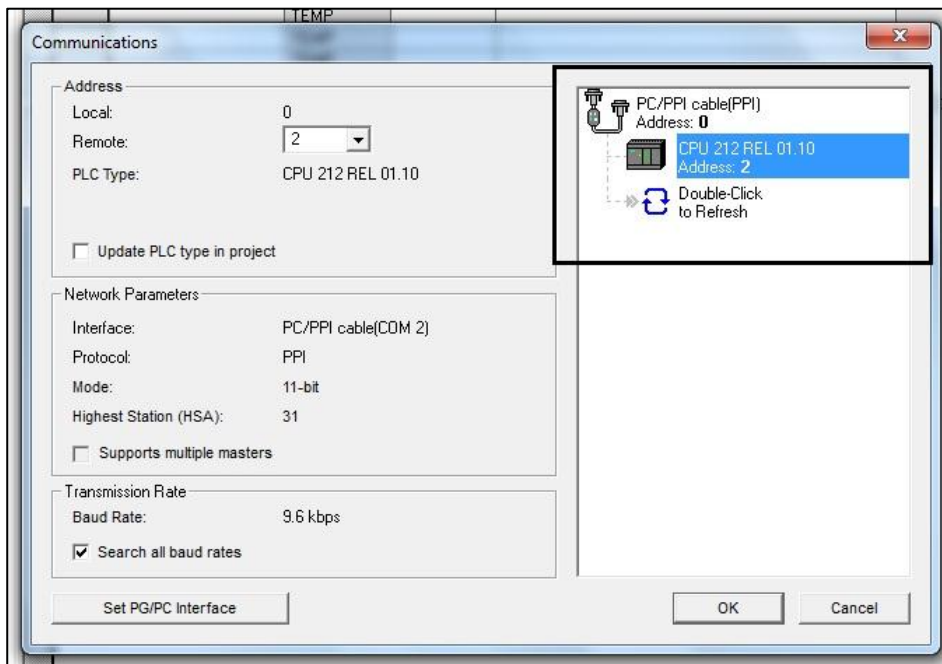
Πριν ξεκινήσουμε να γράφουμε τον κώδικα που θέλουμε, πρέπει πρώτα να κάνουμε κάποιες ρυθμίσεις που έχουν να κάνουν με την επικοινωνία του PLC και του υπολογιστή. Πιο συγκεκριμένα θα πρέπει να ακολουθήσουμε τα παρακάτω βήματα :

- i. Συνδέουμε το PLC με τον υπολογιστή με το καλώδιο PPI
- ii. Σηκώνουμε την ασφάλεια που βρίσκεται στη μονάδα έτσι ώστε να τροφοδοτηθεί το PLC
- iii. Τοποθετούμε τον διακόπτη που βρίσκεται στο PLC στη θέση TERM και ανοίγουμε την τροφοδοσία.
- iv. Επιλέγουμε το εικονίδιο Communications από την εφαρμογή Microwin και στη συνέχεια στο σχετικό πεδίο για ανανέωση, όπου και θα πρέπει να εμφανιστεί η αντίστοιχη CPU
- v. Σε περίπτωση που δεν αναγνωριστεί η CPU, τότε επιλέγουμε το εικονίδιο SET PG/PC INTERFACE και από εκεί την επιλογή properties
- vi. Από την καρτέλα PPI βάζουμε Address 0 και την καρτέλα LOCAL CONNECTION επιλέγουμε τη θύρα που συνδέσαμε το καλώδιο (εικόνα 3.14)



Εικόνα 3.14 : Ιδιότητες PG/PC interface

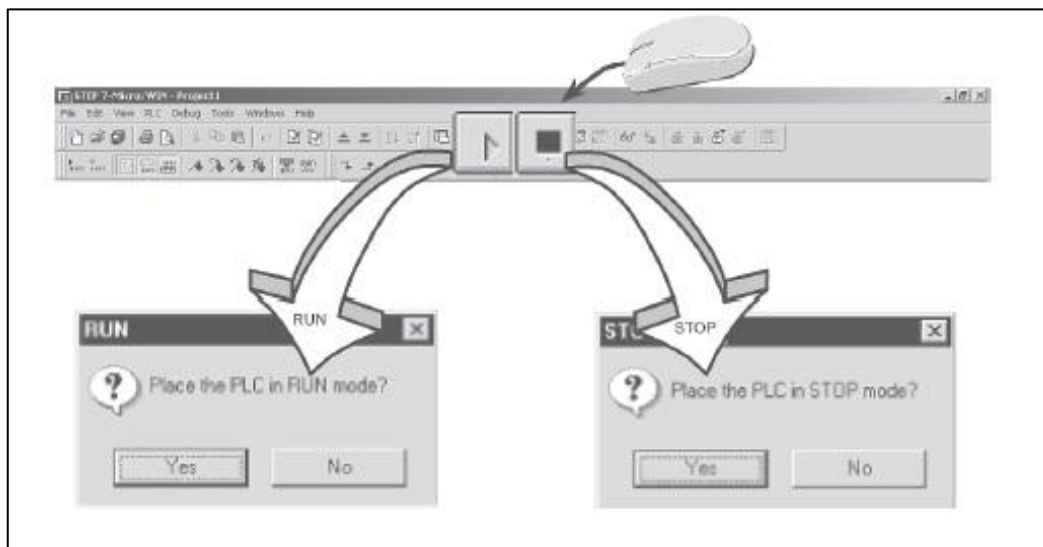
- vii. Πατώντας ξανά το πεδίο ανανέωση από το COMMUNICATIONS η CPU θα πρέπει να μας αναγνωριστεί (εικόνα 3.14)



Εικόνα 3.14 : Αναγνώριση της CPU - 212

Ένας ακόμη τρόπος για να βεβαιωθούμε ότι η επικοινωνία μεταξύ PLC και υπολογιστή έχει γίνει σωστά, είναι ο εξής :

Με τον διακόπτη του PLC στη θέση TERM, από την εφαρμογή MicroWin κάνουμε διαδοχικά κλικ στις επιλογές RUN και STOP (εικόνα 3.15) και παρατηρούμε τα ενδεικτικά Led πάνω στο PLC. Όταν επιλέγουμε RUN θα πρέπει να ανάβει το πράσινο Led ενώ στο STOP το κίτρινο Led.



Εικόνα 3.15 : Έλεγχος ορθής επικοινωνίας

3.4.3 Δημιουργία Project

Σε αυτήν την ενότητα θα δούμε τα βήματα που ακολουθούμε, προκειμένου να δημιουργήσουμε έναν απλό κώδικα από τη σύνταξη του μέχρι το “κατέβασμα” στο PLC.

Έστω έχουμε την παρακάτω εφαρμογή :

Σε ένα σύστημα αυτοματισμού έχουμε δύο κινητήρες M1 και M2 οι οποίοι διαθέτουν από μία ασφάλεια ο καθένας την S1 και S2 αντίστοιχα. Πατώντας το μπουτόν B1 ξεκινάει ο κινητήρας M1 και μετά από 5 δευτερόλεπτα ενεργοποιείται και ο κινητήρας M2, εφόσον οι ασφάλειες τους είναι OK. Για να σταματήσει το σύστημα αρκεί να πατηθεί το μπουτόν B2. Τα στοιχεία είναι συνδεδεμένα στο PLC ως έξης :

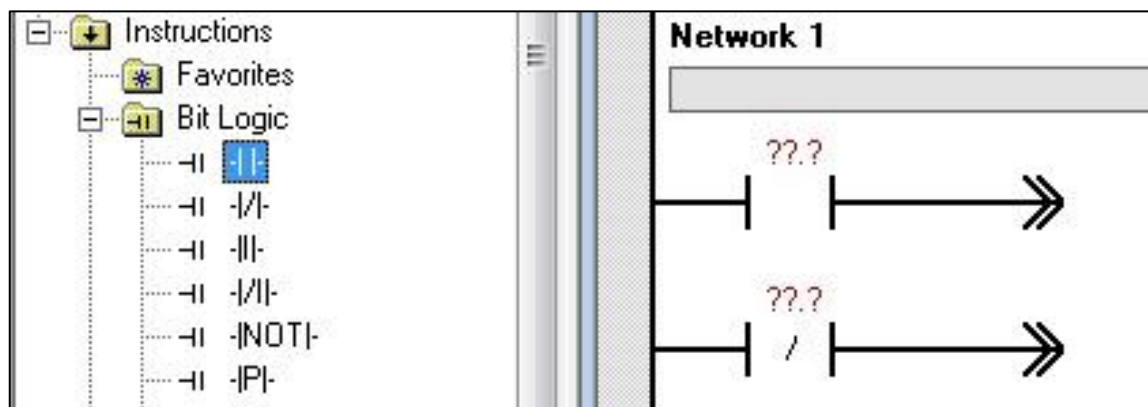
ΕΙΣΟΔΟΙ	
B1	I0.1
B2	I0.2
S1	I0.4
S2	I0.5

ΕΞΟΔΟΙ	
M1	Q0.1
M2	Q0.3

- **Εισαγωγή επαφών**

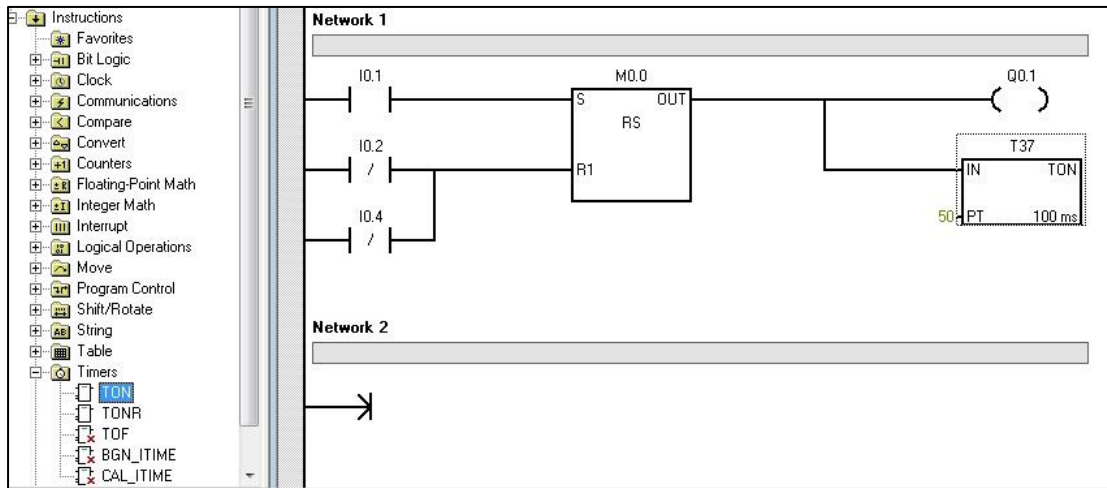
Αφού μελετήσουμε τις απαιτήσεις της εφαρμογής αυτό που πρέπει να κάνουμε πρώτα είναι να αρχίσουμε να εισάγουμε τις επαφές (εικόνα 3.14). Αυτό γίνεται από το μενού συντομεύσεων επιλέγοντας τις επιθυμητές επαφές, με διπλό κλικ αφού πρώτα επιλέξουμε τη θέση που θέλουμε να μπει η επαφή στον editor.

Όπως φαίνεται και από την εικόνα 3.16 πάνω από κάθε επαφή μπαίνουν αυτόματα ερωτηματικά. Αυτό συμβαίνει γιατί δεν έχουμε τοποθετήσει τα ονόματα των εισόδων που αντιστοιχούν στις εισόδους του PLC



Εικόνα 3.16 : Εισαγωγή επαφών

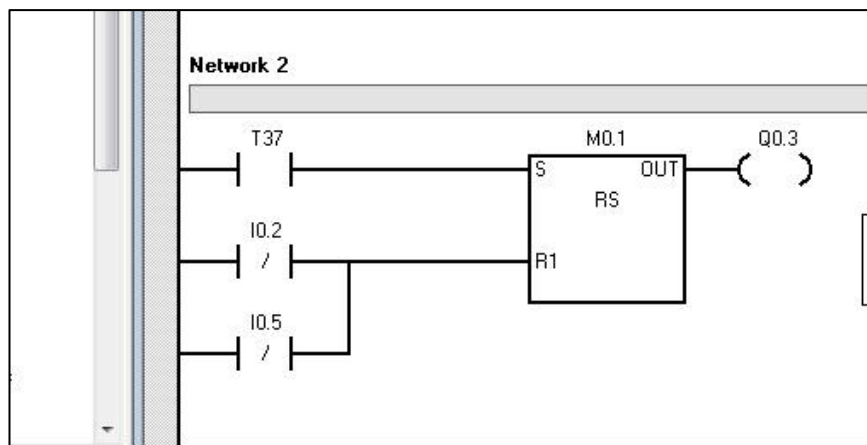
Συνεχίζοντας την τοποθέτηση των απαραίτητων στοιχείων και τοποθετώντας τα αντίστοιχα ονόματα στις επαφές το πρώτο network του προγράμματος μας θα είναι όπως τις εικόνας 3.17. Στο πρώτο network έχουν προγραμματιστεί όλες οι συνθήκες εκκίνησης και σταματήματος για τον M1.



Εικόνα 3.17 : Το πρώτο network

- Ένας πολύ σημαντικός κανόνας που πρέπει να τηρείται κατά τη συγγραφή κώδικα σε γλώσσα LADDER είναι ότι σε κάθε network θα πρέπει να υπάρχει μόνο μια έξοδος Q, διαφορετικά το πρόγραμμα θα εμφανίσει σφάλματα.

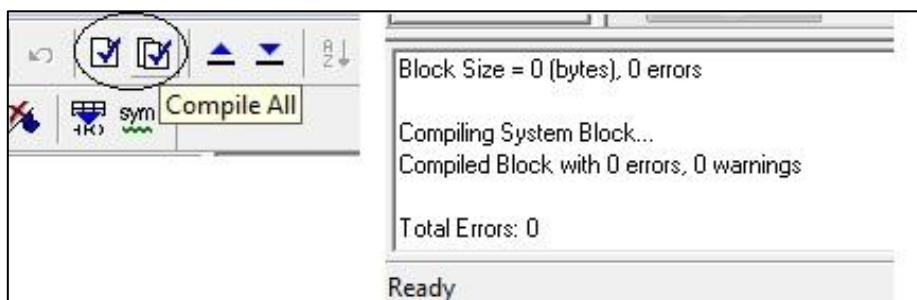
Στο δεύτερο network τοποθετούμε τον κώδικα για τον M2. Όπως φαίνεται για να ενεργοποιηθεί χρησιμοποιούμε την ανοιχτή επαφή από το χρονικό που μετράει τα 5 δευτερόλεπτα.



Εικόνα 3.18 : Το δεύτερο network

- Έλεγχος για συντακτικά λάθη – **Compile**

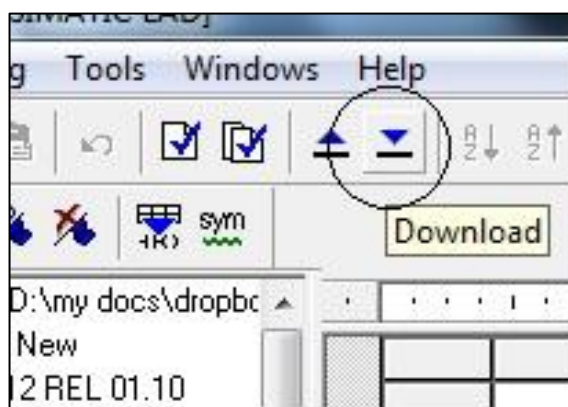
Αφού ολοκληρωθεί το πρόγραμμα, ελέγχουμε για τυχόν συντακτικά λάθη με την επιλογή **Compile all** που βρίσκεται σε συντόμευση στην πάνω μπάρα εργαλείων και στη συνέχεια ελέγχουμε στο πλαίσιο διαλόγου στο οποίο, εφόσον το πρόγραμμα μας είναι σωστό θα μας εμφανίσει μήνυμα ότι δεν υπάρχουν λάθη (εικόνα 3.19).



Εικόνα 3.19 : Έλεγχος για συντακτικά λάθη

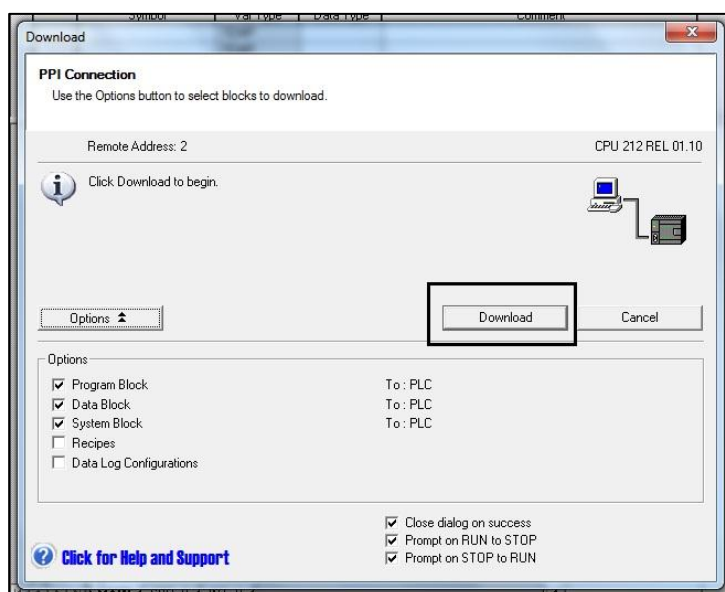
- **Download**

Το τελευταίο βήμα είναι να φορτώσουμε το πρόγραμμα στη μνήμη του PLC. Αυτό μπορεί να γίνει επίσης από τη συντόμευση που βρίσκεται στην πάνω μπάρα



Εικόνα 3.20 : Η επιλογή download

Αμέσως μας εμφανίζεται το επόμενο παράθυρο (εικόνα 3.21) στο οποίο πρέπει να επιλέξουμε την επιλογή Download και εφόσον όλα είναι σωστά το πρόγραμμά μας θα φορτωθεί στο πρόγραμμα.

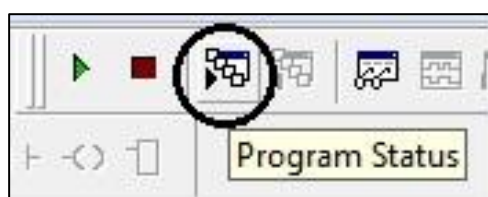


Εικόνα 3.21 : Download το πρόγραμμα στο PLC

Πλέον είμαστε έτοιμοι να ελέγξουμε την ορθότητα του προγράμματος μας και τη συμπεριφορά του σε πραγματικές συνθήκες. Για ξεκινήσει η εκτέλεση του προγράμματος αρκεί είτε να επιλέξουμε την επιλογή RUN από το MicroWin ή να θέσουμε τον διακόπτη επιλογής κατάστασης στη θέση RUN.

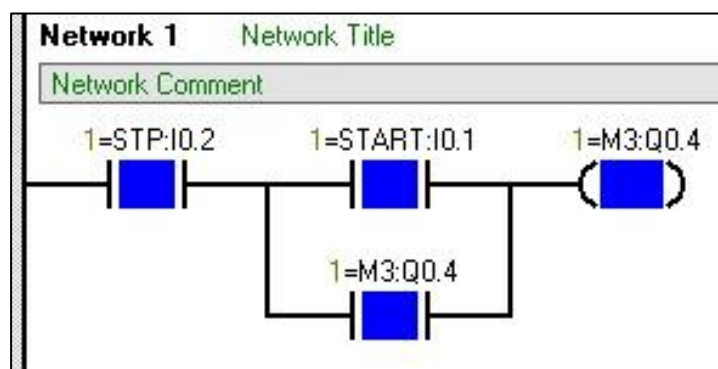
3.4.4 Έλεγχος προγράμματος με τον Program Status

Ένα πολύ σημαντικό εργαλείο που μας παρέχει το MicroWin είναι η επιλογή Program Status το οποίο χρησιμοποιείται για την γραφική απεικόνιση των καταστάσεων των εισόδων και εξόδων του PLC. Με άλλα λόγια, μπορούμε να ελέγξουμε σε πραγματικό χρόνο, την κατάσταση των εντολών του προγράμματος που εκτελούνται από τη CPU και να μπορούμε να ελέγξουμε καλύτερα τη συμπεριφορά του κώδικα μας.



Εικόνα 3.22 : Η επιλογή Program Status

Έστω για παράδειγμα έχουμε το παρακάτω πρόγραμμα (εικόνα 3.23)το οποίο ξεκινάει τον κινητήρα M3 του αναπτύγματος με το button B1 και τον σταματάει με το button B2. Αφού έχει γίνει το Download στο PLC επιλέγουμε το εικονίδιο Program Status. Αυτό το οποίο παρατηρούμε είναι ότι μόλις πιάσουμε το button B1 η επαφή στον κώδικα (Start I0.1) γίνεται μπλε και εφόσον ο M3 ενεργοποιείται ο M3 θα γίνει ενεργοποιηθεί με μπλε χρώμα και η επαφή M3:Q0.4. Με αυτόν τον τρόπο έχουμε καλύτερη εποπτεία του προγράμματος και μπορούμε να κάνουμε τις κατάλληλες τροποποιήσεις προκειμένου να λειτουργήσει σωστά η εφαρμογή μας.



Εικόνα 3.23 : Εποπτεία του κώδικα με τον Program Status

- Επιλέγοντας τη λειτουργία Program Status υπάρχει πιθανότητα να εμφανιστεί κάποια προειδοποίηση. Σε αυτή την περίπτωση επιλέγουμε OK και συνεχίζουμε κανονικά την εργασία μας. Για να μην εμφανιστεί, πριν από κάθε μας ενέργεια θα πρέπει από το μενού : Debug να «ξετικάρουμε » την επιλογή Use Execution Status.

ΚΕΦΑΛΑΙΟ 4 – Η ΕΡΓΑΣΤΗΡΙΑΚΗ ΔΙΑΤΑΞΗ ΤΗΣ ΠΤΥΧΙΑΚΗΣ

4.1 Σκοπός της κατασκευής

Όταν γράφουμε ένα πρόγραμμα στο λογισμικό οποιουδήποτε PLC και να χρησιμοποιούμε, ο μοναδικός τρόπος για να ελέγξουμε την ορθότητα του προγράμματος είναι μέσα από το ίδιο το λογισμικό και μέσω ενός εικονικού προσομοιωτή που μας παρέχει κάθε εταιρεία. Το μειονέκτημα μιας τέτοιας διαδικασίας είναι ότι τα αποτελέσματα φαίνονται μόνο στην οθόνη του Η/Υ και τέτοια διαδικασία είναι αρκετά κοπιαστική και επίπονη ιδιαίτερα σε μεγάλα προγράμματα. Με την κατασκευή όμως, μπορεί να γίνει ο έλεγχος σε πραγματικές συνθήκες, αφού ο χρήστης φορτώνει το πρόγραμμα σε πραγματικό PLC και μπορεί να βλέπει την πραγματική συμπεριφορά του προγράμματος του, με πιο ευχάριστο και προσιτό τρόπο.

4.2 Στάδια υλοποίησης

Παρόλο που η κατασκευή έχει εκπαιδευτικό χαρακτήρα, δεν απέχει καθόλου από τις προδιαγραφές που ισχύουν στη βιομηχανία. Ο λόγος είναι ότι τόσο η σχεδίαση όσο και τα υλικά που χρησιμοποιήθηκαν βασιστήκαν σε βιομηχανικά πρότυπα. Παρακάτω γίνεται μια περιγραφή όλων των βημάτων που ακολουθήθηκαν για την υλοποίηση της κατασκευής.

4.2.1 Επιλογή υλικών

Η επιλογή των υλικών στηρίχθηκε καθαρά στον τύπο του PLC που χρησιμοποιήθηκε, το οποίο όπως προαναφέραμε διαθέτει 8 ψηφιακές εισόδους, 6 ψηφιακές εξόδους και μία βαθμίδα τροφοδοσίας 24 Volt.

i. Είσοδοι

Για τις εισόδους χρησιμοποιήθηκαν 3 push buttons και 4 διακόπτες 0–1. Πιο αναλυτικά :

- 2 push buttons “πράσινα” επαφής Normal Open (NO). Μόλις πατηθούν, τότε τα 24 Volt “περνάνε” από την επαφή τους και οδηγούνται στην είσοδο του PLC



Εικόνα 1 : Push Button NO

- 1 push button “κόκκινο” επαφής Normally Close (NC). Τα 24 Volt “περνάνε” συνεχώς από την επαφή του και κατ’επέκταση στην είσοδο του PLC. Μόλις πατηθεί στην είσοδο του PLC θα έχουμε 0 Volt.



Εικόνα 2 : Push Button NC

- 4 διακόπτες περιστροφικοί 0-1 Normally Open (NO). Όταν ο διακόπτης είναι στη θέση 0 τότε είναι ανοιχτός ενώ όταν βρίσκεται στη θέση 1 είναι κλειστός οπότε τα 24 Volt οδηγούνται στην είσοδο του PLC



Εικόνα 3 : Διακόπτης περιστροφικός NO

- 1 διακόπτης Normally Close (NC) ή αλλιώς μανιτάρι. Ο συγκεκριμένος διακόπτης είναι γνωστός και σαν emergency stop. Στην κανονική του κατάσταση τα 24 Volt “περνάνε” από την επαφή του και οδηγούνται στη είσοδο του PLC. Μόλις πατηθεί ο διακόπτης ανοίγει οπότε στην είσοδο του PLC θα έχουμε 0 volt.



Εικόνα 4 : Διακόπτης Emergency Stop NC

ii. **Έξοδοι**

Η επιλογή των εξόδων ήταν αυθαίρετη και στηρίχθηκε περισσότερο στις εφαρμογές – σενάρια που θα υλοποιηθούν έτσι ώστε το οπτικό αποτέλεσμα να συμβαδίζει με το αντίστοιχο σενάριο. Πιο αναλυτικά χρησιμοποιήθηκαν τα εξής υλικά :

- 3 ανεμιστήρες 220 Volt. Η οποίοι θα έχουν το ρόλο των κινητήρων. Η επιλογή τους στηρίχθηκε περισσότερο σε οικονομικά κριτήρια, μιας και σε αντίστοιχες διαστάσεις και προδιαγραφές, αντίστοιχοι κινητήρες έχουν αρκετά υψηλό κόστος.



Εικόνα 5 : Ανεμιστήρας 220 Volt

- 1 ενδεικτική λυχνία κόκκινη 220 Volt.



Εικόνα 6 : Ενδεικτική λυχνία κόκκινη

- 1 ενδεικτική λυχνία πράσινη 220 Volt



Εικόνα 7 : Ενδεικτική λυχνία πράσινη

- 1 ηλεκτροβάνα 220 Volt. Η ηλεκτροβάνα είναι ένα στοιχείο που χρησιμοποιείται για την διέλευση υγρών ή αερίων. Η λειτουργία της είναι πολύ απλή. Διαθέτει ένα πηνίο το οποίο όταν τροφοδοτείται με 220 Volt τότε η βάνα ανοίγει και επιτρέπει τη διέλευση του υλικού.



Εικόνα 8 : Ηλεκτροβάνα

- 1 περιστρεφόμενος φάρος 220 Volt. Ο φάρος είναι ένα όργανο το οποίο χρησιμοποιείται για να μας ειδοποιήσει για μια κατάσταση έκτακτης ανάγκης.



Εικόνα 9 : Περιστρεφόμενος φάρος

- Όπως φαίνεται από την περιγραφή των υλικών, όλα τα φορτία είναι 220 Volt, για το λόγο αυτό χρησιμοποιήθηκαν και 6 ρελέ με βάσεις ράγας 24 Volt 2 επαφών τα οποία συνδέονται απευθείας στις εξόδους των PLC και τα φορτία ενεργοποιούνται μέσω των επαφών των ρελέ.



Εικόνα 10 : Βάση ρελέ και ρελέ 24 Volt

4.2.2 Βάση τοποθέτησης υλικών – Λαμαρίνα

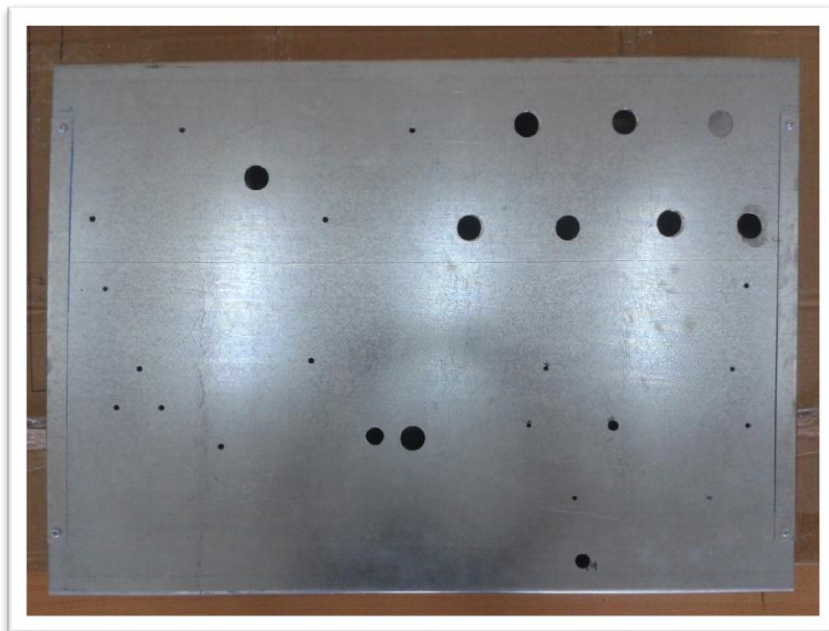
Αφού έγινε η διαστασιολόγηση των υλικών, ακολούθησε ο σχεδιασμός της πλατφόρμας που θα φιλοξενήσει όλα τα υλικά. Το υλικό που χρησιμοποιήθηκε είναι λαμαρίνα και αποτελείται από 3 κομμάτια. Στο πρώτο, που αποτελεί και την πρόσοψη της κατασκευής, έγινε η μεγαλύτερη κατεργασία λόγω του ότι τοποθετήθηκαν όλα τα υλικά και συνδεθήκαν μεταξύ τους. Το δεύτερο κομμάτι αποτελεί το “καπάκι” της πρόσοψης και χρησιμοποιείται περισσότερο για λόγους ασφαλείας και αισθητικής. Και ένα τρίτο κομμάτι χρησιμοποιήθηκε σαν βάση στήριξης για να εξασφαλιστεί η σταθερότητα της κατασκευής.



Εικόνα 11 : Η λαμαρίνα πριν την κατεργασία

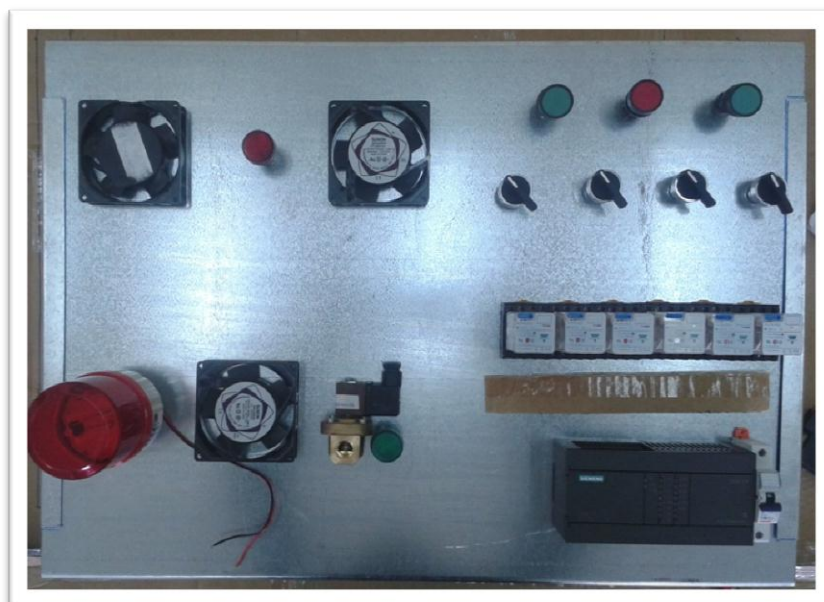
4.2.3 Κατεργασία πρόσοψης και τοποθέτηση των υλικών

Στη συνέχεια ακολούθησε μια εκτίμηση για τη χωροταξική τοποθέτηση των υλικών, με κριτήριο τόσο την αισθητική άποψη όσο και την εύκολη πρόσβαση και διαχείριση την μονάδας από τον φοιτητή και αναλόγως έγινε και η διάτρηση της λαμαρίνας. Όλη η κατεργασία έγινε στο μηχανουργείο του Τμήματος Μηχανολογίας και το αποτέλεσμα φαίνεται στην εικόνα 12.



Εικόνα 12 : Διάτρηση της λαμαρίνας

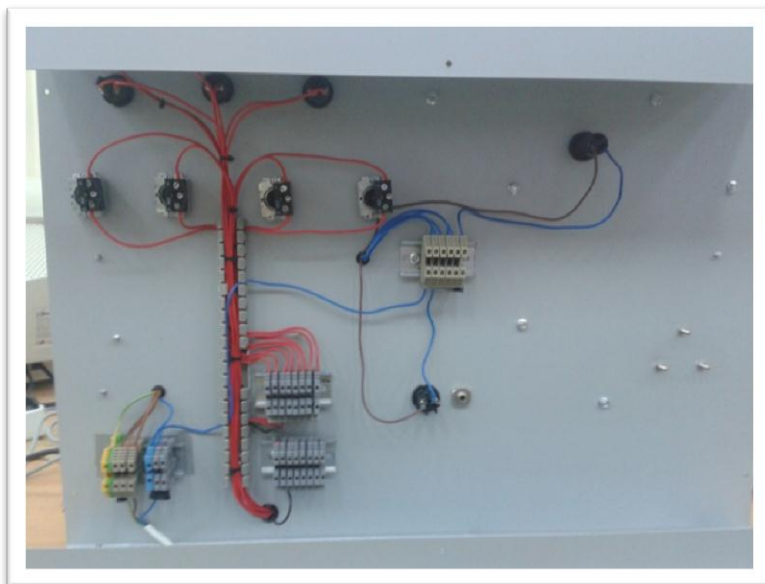
Στην εικόνα 13 φαίνεται η τελική τοποθέτηση των υλικών :



Εικόνα 13 : Τοποθέτηση των υλικών

4.2.4 Καλωδίωση

Το τελευταίο στάδιο ήταν η καλωδίωση των υλικών. Ολόκληρη η καλωδίωση έγινε κυρίως στην πίσω πλευρά της πρόσοψης μέσα σε ειδικά κανάλια, έτσι ώστε ο τελικός χρήστης και να μην έχει πρόσβαση, για λόγους ασφαλείας και να επικεντρωθεί μόνο στον χειρισμό της μονάδας. Στην εικόνα 14 φαίνεται η πίσω πλευρά της πρόσοψης με όλα τα υλικά συνδεδεμένα μεταξύ τους.



Εικόνα 14 : Καλωδίωση των υλικών

4.3 Η κατασκευή ολοκληρωμένη

Αφού ολοκληρώθηκαν όλα τα στάδια της υλοποίησης το τελικό αποτέλεσμα απεικονίζεται στην εικόνα 15:



Εικόνα 15 : Το τελικό αποτέλεσμα

4.3.1 Είσοδοι και έξοδοι του PLC

Στους πίνακες 4.1 και 4.2 παρουσιάζονται όλα τα υλικά που χρησιμοποιήθηκαν και σε ποιες εισόδους και εξόδους του PLC έχουν συνδεθεί. Σε κάθε εφαρμογή που καλούμαστε να υλοποιήσουμε θα πρέπει ο χρήστης να συμβουλευτεί τον συγκεκριμένο πίνακα για να μπορεί να προχωρήσει στη συγγραφή του κώδικα.

ΕΙΣΟΔΟΙ			
ΟΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	ΕΙΣΟΔΟΣ ΣΤΟ PLC
EMERGENCY STOP	ΔΙΑΚΟΠΤΗΣ	NORMALLY CLOSE	I0.0
START 1	PUSH BUTTON ΠΡΑΣΙΝΟ	NORMALLY OPEN	I0.1
STOP	PUSH BUTTON ΚΟΚΚΙΝΟ	NORMALLY OPEN	I0.2
START 2	PUSH BOTTON ΠΡΑΣΙΝΟ	NORMALLY CLOSE	I0.3
S1	ΔΙΑΚΟΠΤΗΣ 0-1	ΠΕΡΙΣΤΡΟΦΙΚΟΣ	I0.4
S2	ΔΙΑΚΟΠΤΗΣ 0-1	ΠΕΡΙΣΤΡΟΦΙΚΟΣ	I0.5
S3	ΔΙΑΚΟΠΤΗΣ 0-1	ΠΕΡΙΣΤΡΟΦΙΚΟΣ	I0.6
S4	ΔΙΑΚΟΠΤΗΣ 0-1	ΠΕΡΙΣΤΡΟΦΙΚΟΣ	I0.7

Πίνακας 4.1 : Υλικά και εισόδου του PLC

ΕΞΟΔΟΙ			
ΟΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	ΕΞΟΔΟΣ ΣΤΟ PLC
ΣΕΙΡΙΝΑ	ΦΑΡΟΣ ΕΚΤΑΚΤΗΣ ΑΝΑΓΚΗΣ	220 VOLT	Q0.0
ΚΙΝΗΤΗΡΑΣ M1	ΕΞΑΕΡΙΣΤΙΚΟ ΤΕΤΡΑΓΩΝΟ	220 VOLT	Q0.1
H1	ΕΝΔΕΙΚΤΙΚΗ ΛΥΧΝΙΑ ΚΟΚΚΙΝΗ	220 VOLT	Q0.2
ΚΙΝΗΤΗΡΑΣ M2	ΕΞΑΕΡΙΣΤΙΚΟ ΤΕΤΡΑΓΩΝΟ	220 VOLT	Q0.3
ΚΙΝΗΤΗΡΑΣ M3	ΕΞΑΕΡΙΣΤΙΚΟ ΤΕΤΡΑΓΩΝΟ	220 VOLT	Q0.4
ΗΛΕΚΤΡΟΒΑΝΑ / H2	ΒΑΝΑ ON - OFF / ΕΝΔΕΙΚΤΙΚΗ ΛΥΧΝΙΑ ΠΡΑΣΙΝΗ	220 VOLT	Q0.5

Πίνακας 4.2 : Υλικά και έξοδοι του PLC

- Στην έξοδο Q0.5 έχουν συνδεθεί δύο φορτία και η ηλεκτροβάνα και η πράσινη ενδεικτική λυχνία. Ο λόγος είναι για να υπάρχει και ο οπτικός έλεγχος ότι έχει ενεργοποιηθεί η αντίστοιχη έξοδος.

ΚΕΦΑΛΑΙΟ 5 – ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΕΦΑΡΜΟΓΕΣ

5.1 Έλεγχος κινητήρων

5.1.1 Έλεγχος κινητήρα M1

Να γραφεί πρόγραμμα σε γλώσσα Ladder που να υλοποιεί τον παρακάτω αυτοματισμό :

Ο κινητήρας M1 ξεκινάει μόνο με την προϋπόθεση οι διακόπτες S1, S2, S3 και S4 βρίσκονται στη θέση 1. Για να σταματήσει ο κινητήρας θα πρέπει να πατηθεί το Emergency Stop.

Να χρησιμοποιηθούν οι είσοδοι και οι έξοδοι του παρακάτω πίνακα :

ΕΙΣΟΔΟΙ		
A/A	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ
1	S1	I0.4
2	S2	I0.5
3	S3	I0.6
4	S4	I0.7
5	EMERGENCY_STOP	I0.0

ΕΞΟΔΟΙ		
A/A	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ
1	M1	Q0.1

Βήμα 1 - Ρυθμίσεις επικοινωνίας (ενότητα 3.4.3)

1. Ανοίγουμε το λογισμικό MicroWin
2. Συνδέουμε το PLC με τον υπολογιστή με το καλώδιο PPI
3. Σηκώνουμε την ασφάλεια που βρίσκεται στη μονάδα έτσι ώστε να τροφοδοτηθεί το PLC
4. Τοποθετούμε τον διακόπτη που βρίσκεται στο PLC στη θέση TERM και ανοίγουμε την τροφοδοσία.
5. Επιλέγουμε το εικονίδιο Communications από την εφαρμογή MicroWin και στη συνέχεια στο σχετικό πεδίο για ανανέωση, όπου και θα πρέπει να εμφανιστεί η αντίστοιχη CPU
6. Σε περίπτωση που δεν αναγνωριστεί η CPU, τότε επιλέγουμε το εικονίδιο SET PG/PC INTERFACE και από εκεί την επιλογή properties
7. Από την καρτέλα PPI βάζουμε Address 0 και την καρτέλα LOCAL CONNECTION επιλέγουμε τη θύρα που συνδέσαμε το καλώδιο. Πατώντας ξανά το πεδίο ανανέωση από το COMMUNICATIONS η CPU θα πρέπει να μας αναγνωριστεί

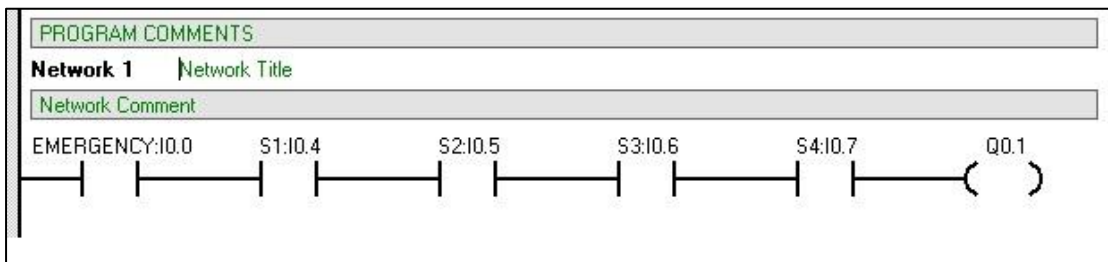
Βήμα 2 – Απόδοση μνημονικών ονομάτων (προαιρετικά)

Μπορούμε να χρησιμοποιήσουμε την επιλογή Symbol Table και να αποδώσουμε τα ονόματα που βρίσκονται στο παραπάνω πίνακα ή οποιαδήποτε επιθυμούμε. Η διαδικασία περιγράφεται αναλυτικά στην ενότητα 3.3.2

Βήμα 3 – Συγγραφή κώδικα

Αφού μελετήσουμε προσεκτικά τις απαιτήσεις της άσκησης μπορούμε να ξεκινήσουμε τη συγγραφή του κώδικα σε γλώσσα LADDER. Για να ξεκινήσει ο κινητήρας, πρέπει όλοι οι διακόπτες να βρίσκονται σε θέση 1 ταυτόχρονα. Ουσιαστικά είναι σαν να θέλουμε να υλοποιήσουμε μια πύλη AND με 5 εισόδους. Οπότε ξεκινάμε να εισάγουμε τις επαφές σε ένα network. Η διαδικασία της συγγραφής του κώδικα και εισαγωγής επαφών περιγράφεται αναλυτικά στην ενότητα 3.4.3.

Συνεπώς, το πρόγραμμα μας θα είναι όπως παρακάτω :



Λύση 5.1.1

Βήμα 4 – Δοκιμή του προγράμματος

Κατεβάζουμε το πρόγραμμα μας στο PLC σύμφωνα με τα βήματα που περιγράφονται στην ενότητα 3.4.3 (παράγραφος Download) και το τοποθετούμε το PLC από το λογισμικό σε θέση RUN.

Για να διαπιστώσουμε την ορθότητα του προγράμματος, τοποθετούμε όλους τους διακόπτες στη θέση 1. Ο διακόπτης Emergency Stop πρέπει να μην είναι πατημένος, γιατί όπως αναφέραμε είναι επαφή κανονικά κλειστή. Ο κινητήρας M1 θα πρέπει να λειτουργεί. Αν κάποιο διακόπτης πάει σε θέση 0 ο κινητήρας σταματάει.

5.1.2 Έλεγχος κινητήρων M1 και M2

Να γραφεί πρόγραμμα σε γλώσσα Ladder πρόγραμμα που να υλοποιεί τον παρακάτω αυτοματισμό :

Ο κινητήρας M1 ξεκινάει μόνο με την προϋπόθεση ότι ένας διακόπτης S1 ή S2 βρίσκεται στη θέση 1. Επίσης ο κινητήρας M2 ξεκινάει μόνο με την προϋπόθεση αν ένας διακόπτης S3 ή S4 βρίσκεται σε θέση ON. Οι κινητήρες σταματάνε αν πατηθεί το Emergency Stop ανεξάρτητα από τη θέση των διακοπών.

Να χρησιμοποιηθούν οι είσοδοι και οι έξοδοι του παρακάτω πίνακα :

ΕΙΣΟΔΟΙ		
A/A	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ
1	S1	I0.4
2	S2	I0.5
3	S3	I0.6
4	S4	I0.7
5	EMERGENCY_STOP	I0.0

ΕΞΟΔΟΙ		
A/A	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ
1	M1	Q0.1
2	M2	Q0.3

Βήμα 1 - Ρυθμίσεις επικοινωνίας (ενότητα 3.4.3)

1. Ανοίγουμε το λογισμικό MicroWin
2. Συνδέουμε το PLC με τον υπολογιστή με το καλώδιο PPI
3. Σηκώνουμε την ασφάλεια που βρίσκεται στη μονάδα έτσι ώστε να τροφοδοτηθεί το PLC
4. Τοποθετούμε τον διακόπτη που βρίσκεται στο PLC στη θέση TERM και ανοίγουμε την τροφοδοσία.
5. Επιλέγουμε το εικονίδιο Communications από την εφαρμογή MicroWin και στη συνέχεια στο σχετικό πεδίο για ανανέωση, όπου και θα πρέπει να εμφανιστεί η αντίστοιχη CPU
6. Σε περίπτωση που δεν αναγνωριστεί η CPU, τότε επιλέγουμε το εικονίδιο SET PG/PC INTERFACE και από εκεί την επιλογή properties
7. Από την καρτέλα PPI βάζουμε Address 0 και την καρτέλα LOCAL CONNECTION επιλέγουμε τη θύρα που συνδέσαμε το καλώδιο. Πατώντας ξανά το πεδίο ανανέωση από το COMMUNICATIONS η CPU θα πρέπει να μας αναγνωριστεί

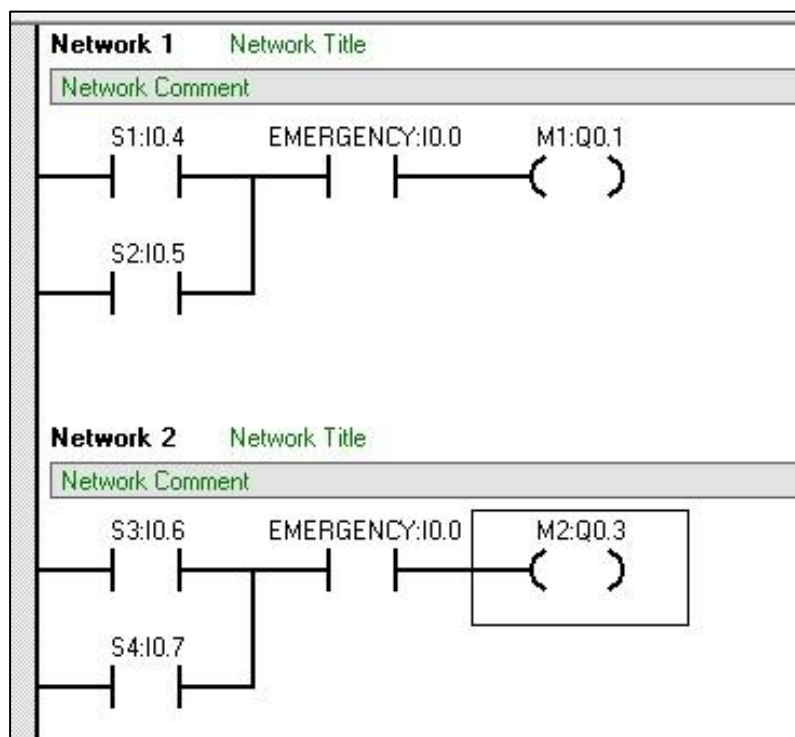
Βήμα 2 – Απόδοση μνημονικών ονομάτων (προαιρετικά)

Μπορούμε να χρησιμοποιήσουμε την επιλογή Symbol Table και να αποδώσουμε τα ονόματα που βρίσκονται στο παραπάνω πίνακα ή οποιαδήποτε επιθυμούμε. Η διαδικασία περιγράφεται στην ενότητα 3.3.2

Βήμα 3 – Συγγραφή κώδικα

Αφού μελετήσουμε προσεκτικά τις απαιτήσεις της άσκησης μπορούμε να ξεκινήσουμε τη συγγραφή του κώδικα σε γλώσσα LADDER. Για να ξεκινήσει ο κινητήρας M1, οι διακόπτες S1 ή S2 να βρίσκονται σε θέση 1. Με λίγα λόγια οι διακόπτες θα πρέπει να είναι σε διάταξη OR ενώ ο διακόπτης Emergency Stop να μπει σε σειρά με το δικτύωμα OR. Οπότε ξεκινάμε να εισάγουμε τις επαφές στο πρώτο network. Με τον ίδιο τρόπο γράφουμε τον κώδικα και για τον κινητήρα M2 σε διαφορετικό network. Η διαδικασία της συγγραφής του κώδικα και εισαγωγής επαφών περιγράφεται αναλυτικά στην ενότητα 3.4.3.

Συνεπώς, το πρόγραμμα μας θα είναι όπως παρακάτω :



Λύση 5.1.2

Βήμα 4 – Δοκιμή του προγράμματος

Κατεβάζουμε το πρόγραμμα μας στο PLC σύμφωνα με τα βήματα που περιγράφονται στην ενότητα 3.4.3 (παράγραφος Download) και το τοποθετούμε το PLC από το λογισμικό σε θέση RUN.

Για να διαπιστώσουμε την ορθότητα του προγράμματος, τοποθετούμε όλους τους διακόπτες στη θέση 0. Ο διακόπτης Emergency Stop πρέπει να μην είναι πατημένος. Περιστρέφουμε τους διακόπτες διαδοχικά και παρατηρούμε την ορθότητα του προγράμματος μας ανάλογα με τη θέση των διακοπών. Οι κινητήρες θα πρέπει να σταματάνε μόλις πατηθεί ο διακόπτης Emergency Stop.

5.1.3 Έλεγχος κινητήρων M1 και M2

Να γραφεί πρόγραμμα σε γλώσσα Ladder που να υλοποιεί τον παρακάτω αυτοματισμό :

Οι κινητήρες M1 και M3 ξεκινάνε μόνο με την προϋπόθεση ότι η ασφάλεια τους και η θερμομαγνητική επαφή, που χρησιμοποιείται για την προστασία από υπερβολικό ρεύμα, είναι σε κατάσταση ON και όταν πατηθεί το Button–Start του κάθε κινητήρα. Οι κινητήρες σταματάνε και οι δυο είτε με το Button Stop είτε με το Emergency Stop. Για να ξεκινήσει ο M1 θα πρέπει να ξεκινήσει ο M2.

Να χρησιμοποιηθούν οι είσοδοι και οι έξοδοι του παρακάτω πίνακα :

ΕΙΣΟΔΟΙ		
A/A	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ
1	START1	I0.1
2	STOP	I0.2
3	START2	I0.3
4	S1 : ASF1	I0.4
5	S2 : THERMIKO1	I0.5
6	S3 : ASF2	I0.6
7	S4 : THERMIKO2	I0.7
8	EMERGENCY_STOP	I0.0

ΕΞΟΔΟΙ		
A/A	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ
1	M1	Q0.1
2	M3	Q0.4

Βήμα 1 - Ρυθμίσεις επικοινωνίας

Ακολουθούμε τα βήματα της ενότητας 3.4.3

Βήμα 2 – Απόδοση μνημονικών ονομάτων (προαιρετικά)

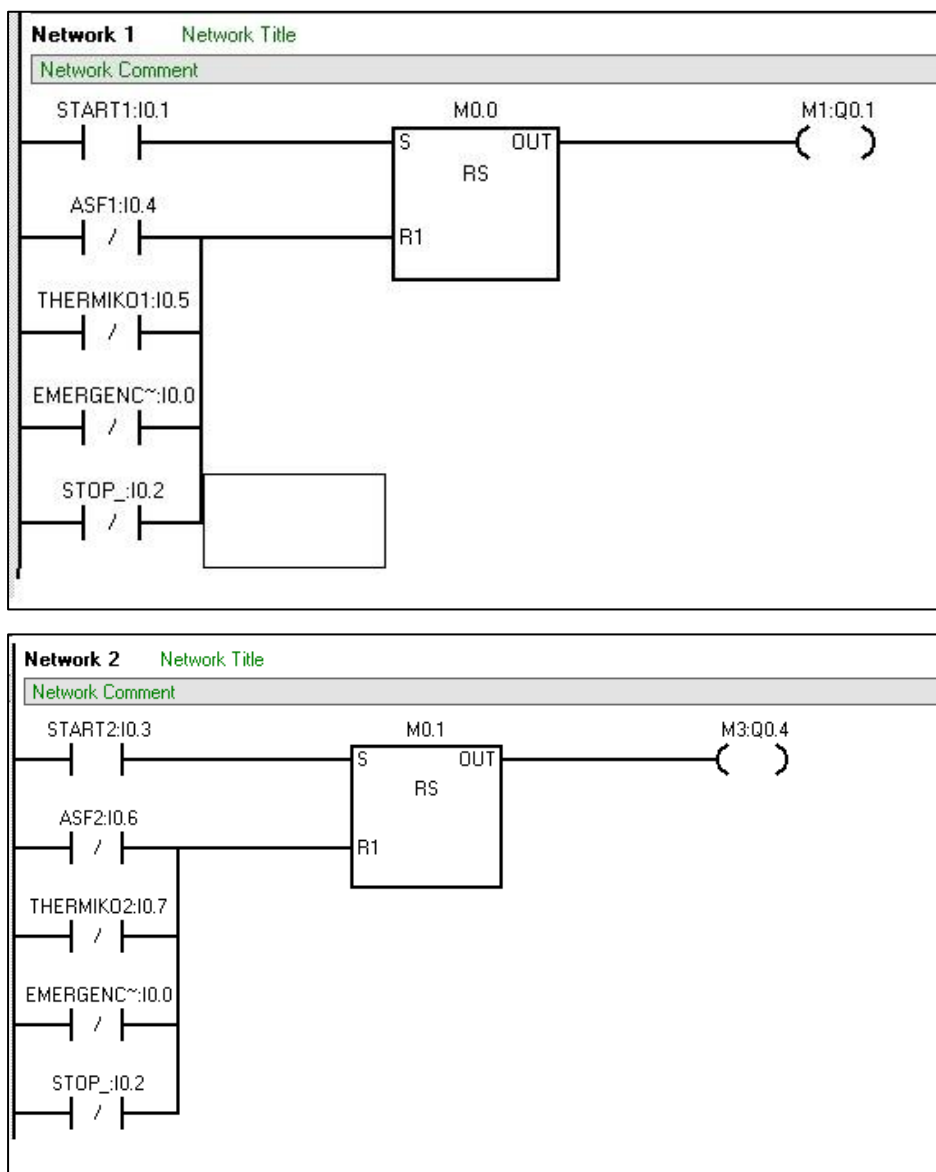
Όπως περιγράφεται στην ενότητα 3.3.2

Βήμα 3 – Συγγραφή κώδικα

Αφού μελετήσουμε προσεκτικά τις απαιτήσεις της άσκησης μπορούμε να ξεκινήσουμε τη συγγραφή του κώδικα σε γλώσσα LADDER.

Στη συγκεκριμένη άσκηση θα χρησιμοποιήσουμε της εντολές Reset – Set που περιγράφονται αναλυτικά στην ενότητα 2.2.1. Στην είσοδο Set τοποθετούμε όλες τις συνθήκες για τις οποίες οι κινητήρες πρέπει να ξεκινάνε. Στην προκειμένη περίπτωση τα Button Start1 και Start2. Ενώ στην είσοδο R όλες τις συνθήκες που οι κινητήρες πρέπει να σταματάνε.

Συνεπώς το πρόγραμμα μας θα πρέπει να είναι όπως παρακάτω :



Λύση 5.1.3

Βήμα 4 – Δοκιμή του προγράμματος

Κατεβάζουμε το πρόγραμμα μας στο PLC σύμφωνα με τα βήματα που περιγράφονται στην ενότητα 3.4.3 (παράγραφος Download) και το τοποθετούμε το PLC από το λογισμικό σε θέση RUN.

Για να διαπιστώσουμε την ορθότητα του προγράμματος, τοποθετούμε όλους τους διακόπτες στη θέση 1. Ο διακόπτης Emergency Stop πρέπει να μην είναι πατημένος.

Πατάμε τα Button Start1 και Start 2 έτσι ώστε να ξεκινήσουν οι κινητήρες και στη συνέχεια αλλάζουμε κατάσταση στους διακόπτες για να ελέγξουμε ότι σταματάνε.

5.2 Χρονικά - Timers

Να γραφεί σε γλώσσα LADDER πρόγραμμα που να υλοποιεί τον παρακάτω αυτοματισμό :

Όταν ο διακόπτης S1 θα είναι ενεργοποιημένος, τότε ενεργοποιείται ο κινητήρας M1 για 15 sec. Μόλις απενεργοποιείται ο M1, ενεργοποιείται επίσης για 15 sec ο M2 και τέλος κατά την απενεργοποίηση του M2 ενεργοποιείται ο M3 για 15 sec. Η κυκλική ενεργοποίηση των κινητήρων διαρκεί για όσο χρόνο ο διακόπτης S1 βρίσκεται στη θέση 1.

Να χρησιμοποιηθούν οι είσοδοι και οι έξοδοι του παρακάτω πίνακα :

ΕΙΣΟΔΟΙ		
A/A	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ
1	S1	I0.4

ΕΞΟΔΟΙ		
A/A	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ
1	M1	Q0.1
2	M2	Q0.3
3	M3	Q0.4

Βήμα 1 - Ρυθμίσεις επικοινωνίας

Ακολουθούμε τα βήματα της ενότητας 3.4.3

Βήμα 2 – Απόδοση μνημονικών ονομάτων (προαιρετικά)

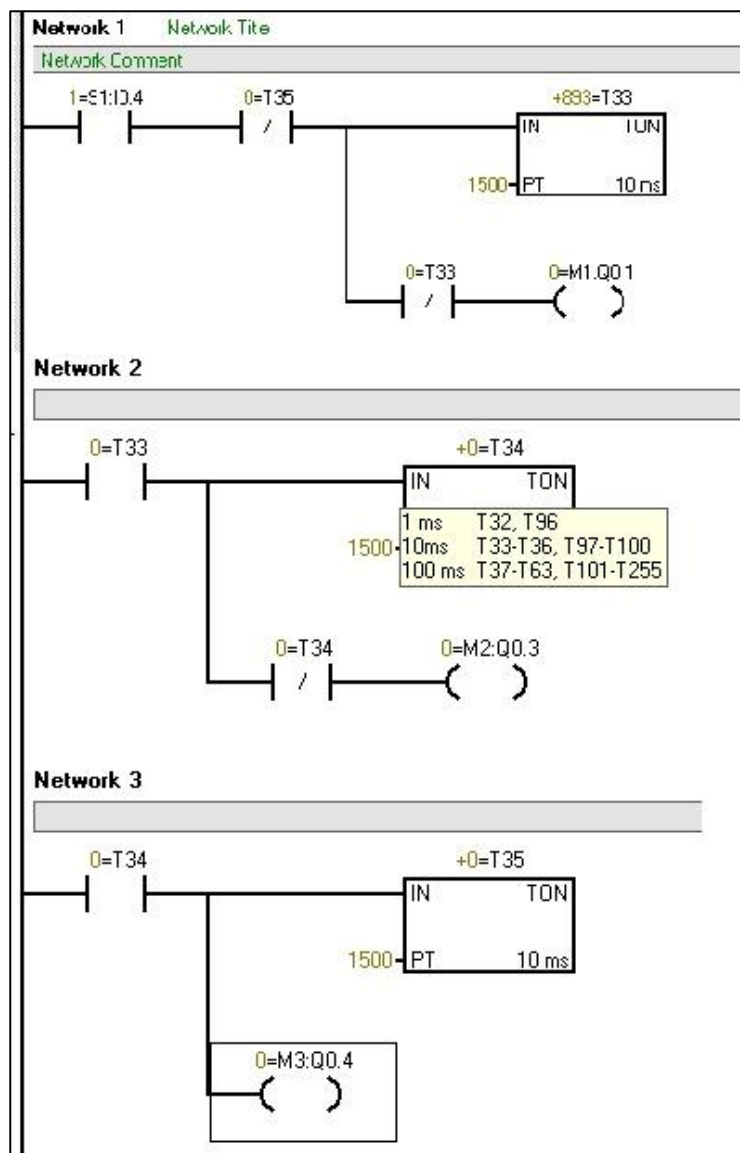
Όπως περιγράφεται στην ενότητα 3.3.2

Βήμα 3 – Συγγραφή κώδικα

Αφού μελετήσουμε προσεκτικά τις απαιτήσεις της άσκησης μπορούμε να ξεκινήσουμε τη συγγραφή του κώδικα σε γλώσσα LADDER.

Στην άσκηση αυτή γίνεται η χρήση χρονικών που περιγράφεται αναλυτικά στην ενότητα 3.3.1. Συγκεκριμένα χρησιμοποιούμε το χρονικά T33 για την λειτουργία του M1, το T34 για την λειτουργία του M2 και το T35 για τη λειτουργία του M3. Σε κάθε χρονικό έχουμε φορτώσει την τιμή 1500 έτσι ώστε να μετράει μέχρι 15 δευτερόλεπτα. Όσο το χρονικό του κάθε κινητήρα είναι κάτω από την τιμή 15 και ο διακόπτης S1 βρίσκεται στη θέση 1, ο κινητήρας θα πρέπει να λειτουργεί. Όταν το χρονικό ξεπεράσει την τιμή που το καθίσαμε το Bit επαφής γίνεται 1, οπότε θα αποτελέσει και τη συνθήκη Start για τον επόμενο κινητήρα και την εκκίνηση του χρονικού του κινητήρα.

Συνεπώς το πρόγραμμα μας θα πρέπει να είναι όπως παρακάτω :



Λύση 5.2

Βήμα 4 – Δοκιμή του προγράμματος

Κατεβάζουμε το πρόγραμμα μας στο PLC σύμφωνα με τα βήματα που περιγράφονται στην ενότητα 3.4.3 (παράγραφος Download) και το τοποθετούμε το PLC από το λογισμικό σε θέση RUN.

Για να διαπιστώσουμε την ορθότητα του προγράμματος, τοποθετούμε το διακόπτη S1 στη θέση 1. Παρατηρούμε ότι ξεκινάει ο M1 για 15 δευτερόλεπτα έπειτα ο M2 και κατόπιν ο M3. Ο κύκλος θα συνεχιστεί μέχρι να τοποθετήσουμε το διακόπτη S1 στη θέση 0.

5.3 Απαριθμητές - Counters

Να γραφεί πρόγραμμα σε γλώσσα LADDER που να υλοποιεί τον παρακάτω αυτοματισμό :

Με τον διακόπτη S2 στη θέση 1 το σύστημα μπαίνει σε ετοιμότητα. Πατώντας το μπουτόν B3 κάθε φορά να ενεργοποιούνται και να απενεργοποιούνται διαδοχικά τα φορτία της εργαστηριακής διάταξης Ο αυτοματισμός θα συνεχίζει μέχρι ο διακόπτης S2 γίνει 0.

Να χρησιμοποιηθούν οι είσοδοι και οι έξοδοι του παρακάτω πίνακα :

ΕΙΣΟΔΟΙ		
A/A	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ
1	S2	I0.5
2	B3	I0.3

ΕΞΟΔΟΙ		
A/A	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ
1	FAROS	Q0.0
2	M1	Q0.1
3	H1	Q0.2
4	M2	Q0.3
5	M3	Q0.4
6	H2	Q0.5

Βήμα 1 - Ρυθμίσεις επικοινωνίας

Ακολουθούμε τα βήματα της ενότητας 3.4.3

Βήμα 2 – Απόδοση μνημονικών ονομάτων (προαιρετικά)

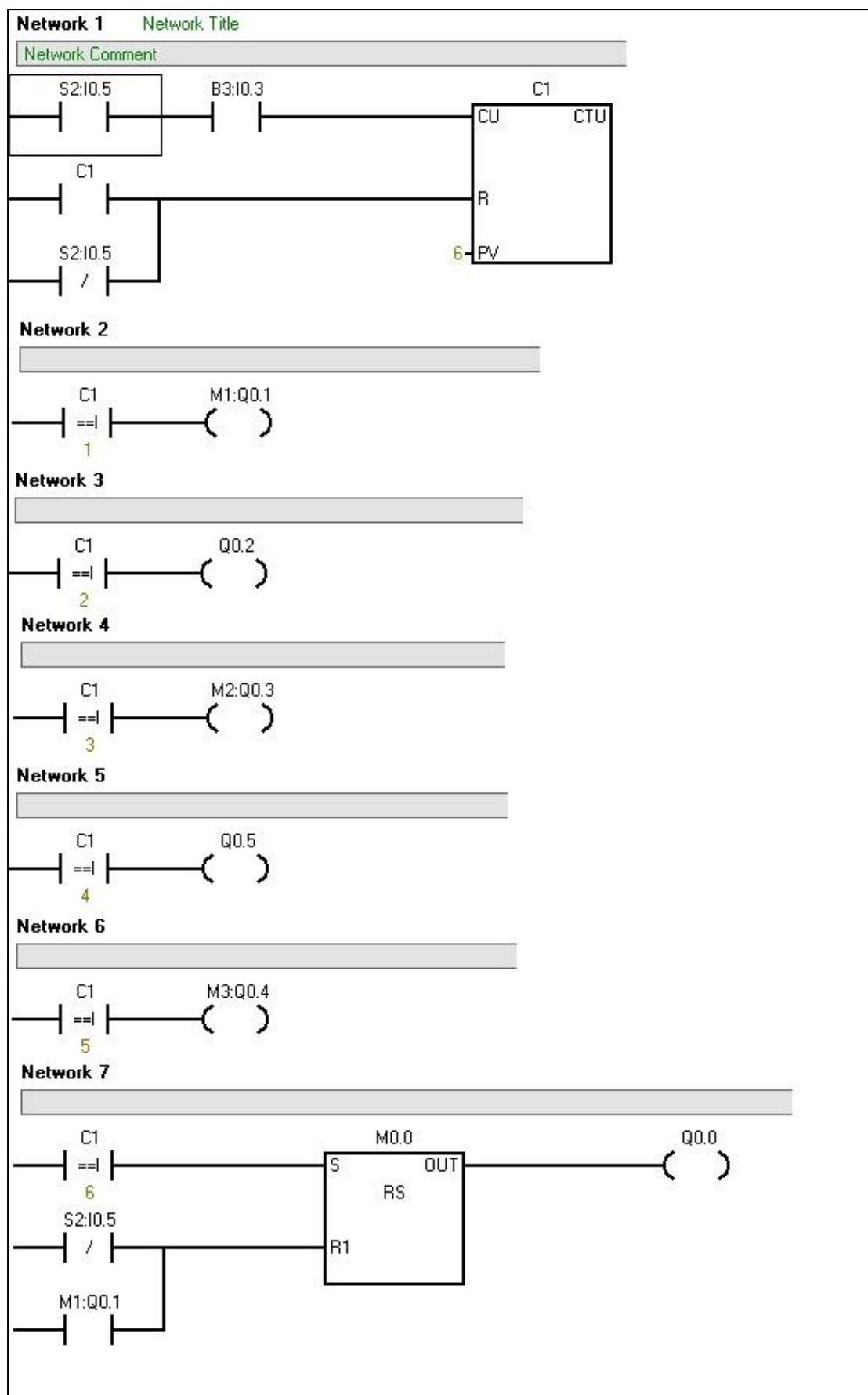
Όπως περιγράφεται στην ενότητα 3.3.2

Βήμα 3 – Συγγραφή κώδικα

Αφού μελετήσουμε προσεκτικά τις απαιτήσεις της άσκησης μπορούμε να ξεκινήσουμε τη συγγραφή του κώδικα σε γλώσσα LADDER.

Στην άσκηση αυτή γίνεται η χρήση των μετρητών που η λειτουργία τους αναλύεται στην ενότητα 3.3.1 καθώς επίσης και των συγκριτών που περιγράφονται στην ενότητα 2.2.2.

Συγκεκριμένα χρησιμοποιούμε τον μετρητή C1 ο οποίος είναι ένας μετρητής Counter Up και του έχουμε φορτώσει την τιμή 6. Όσο ο διακόπτης S2 βρίσκεται στη θέση 1 και κάθε φορά που πιέζουμε το Button B3 το περιεχόμενο του μετρητή αυξάνει κατά 1. Με τη χρήση των συγκριτών ενεργοποιούμε τα φορτία της μονάδας. Για παράδειγμα, όταν ο C1 θα έχει την τιμή 3, η επαφή με την οποία θα γίνει η σύγκριση και της έχουμε δώσει την τιμή 3 θα έχει σαν αποτέλεσμα να ενεργοποιηθεί η επαφή της σύγκρισης. Οπότε ενεργοποιείται και το αντίστοιχο φορτίο.



Λύση 5.3

Βήμα 4 – Δοκιμή του προγράμματος

Κατεβάζουμε το πρόγραμμα μας στο PLC σύμφωνα με τα βήματα που περιγράφονται στην ενότητα 3.4.3 (παράγραφος Download) και το τοποθετούμε το PLC από το λογισμικό σε θέση RUN.

Για να διαπιστώσουμε την ορθότητα του προγράμματος, τοποθετούμε το διακόπτη S2 στη θέση 1. Πιέζουμε τον Button B3 και παρατηρούμε την ενεργοποίηση των φορτίων. Μόλις τοποθετήσουμε τον διακόπτη S2 στη θέση 0 σταματάει η λειτουργία.

5.4 Έλεγχος μεταφορικών ταινιών

Να υλοποιηθεί πρόγραμμα σε γλώσσα LADDER που θα υλοποιεί τον παρακάτω αυτοματισμό :

Η μεταφορική ταινία M1 μεταφέρει μπουκάλια με νερό. Για να ενεργοποιηθεί θα πρέπει ο χειριστής να πατήσει το button Start1. Στο μέσο της ταινίας υπάρχει ένας αισθητήρας μέτρησης οπτού κάθε φορά που περνάει ένα μπουκάλι αυξάνει κατά 1. Μόλις συμπληρωθεί μια εξάδα η M1 σταματάει και ενεργοποιείται για 10 sec η M2 η οποία οδηγεί τα προϊόντα στο επόμενο στάδιο. Η M1 και η M2 έχουν ασφάλειες και θερμικά (διακόπτες S1, S2, S3 και S4). Όταν όλες οι ασφάλειες και τα θερμικά είναι OK (θέση 1) τότε είναι ενεργοποιημένη η πράσινη λυχνία. Σε αντίθετη περίπτωση ενεργοποιείται η κόκκινη. Για να σταματήσει ο αυτοματισμός θα πρέπει να πατηθεί το μπουτόν STOP ή το Emergency Stop όπου σε αυτήν την περίπτωση ενεργοποιείται και ο φάρος.

Να χρησιμοποιηθούν οι είσοδοι και οι έξοδοι του παρακάτω πίνακα :

ΕΙΣΟΔΟΙ		
A/A	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ
1	EMERGENCY_STOP	I0.0
2	START1	I0.1
3	STOP	I0.2
4	COUNTER	I0.3
4	ASF1	I0.4
5	THERMIKO1	I0.5
6	ASF2	I0.6
7	THERMIKO2	I0.7

ΕΞΟΔΟΙ		
A/A	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ
1	FAROS	Q0.0
2	M1	Q0.1
3	ERROR	Q0.2
4	M2	Q0.3
5	H2	Q0.5

Βήμα 1 - Ρυθμίσεις επικοινωνίας

Ακολουθούμε τα βήματα της ενότητας 3.4.3

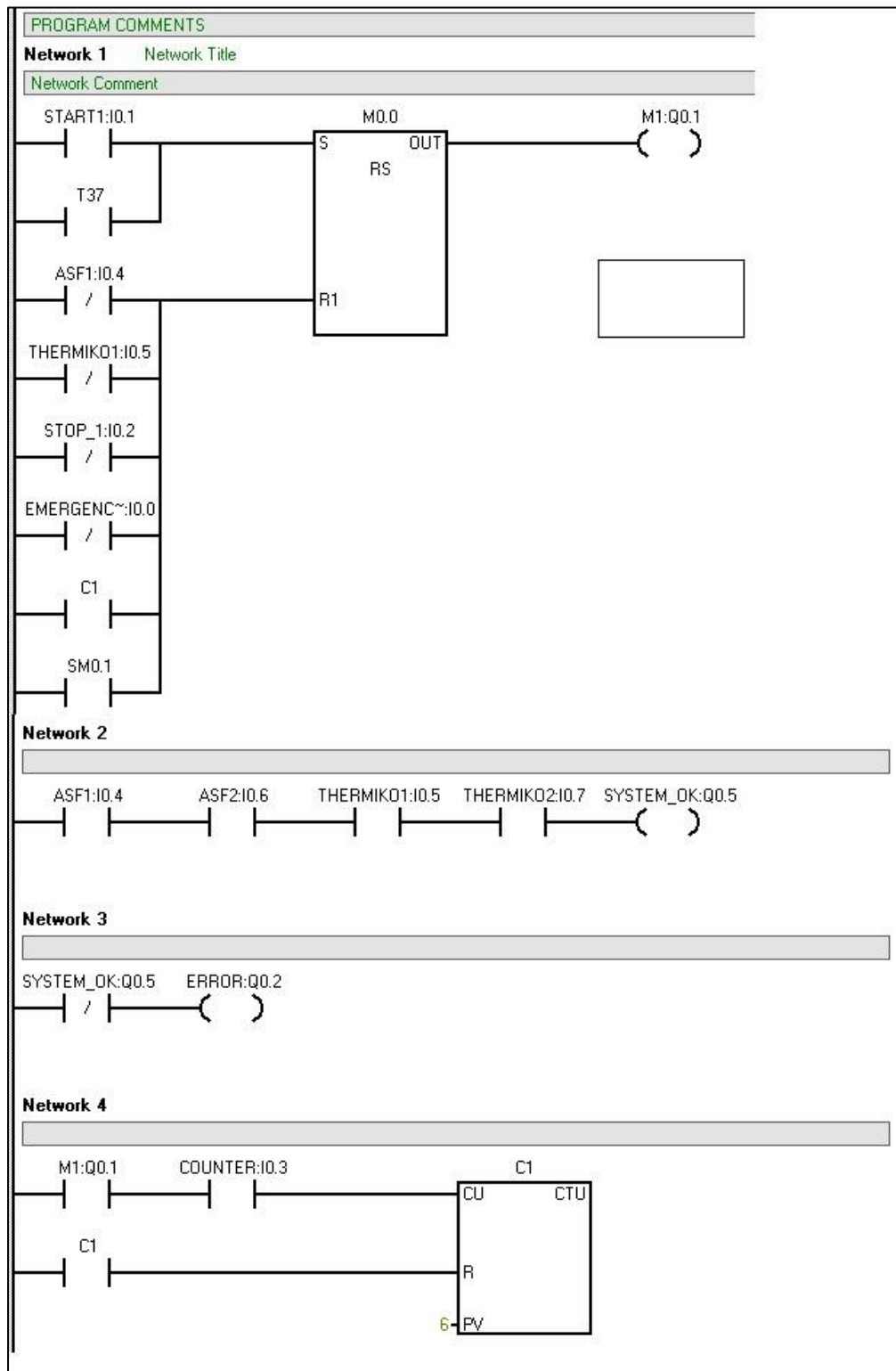
Βήμα 2 – Απόδοση μνημονικών ονομάτων (προαιρετικά)

Όπως περιγράφεται στην ενότητα 3.3.2

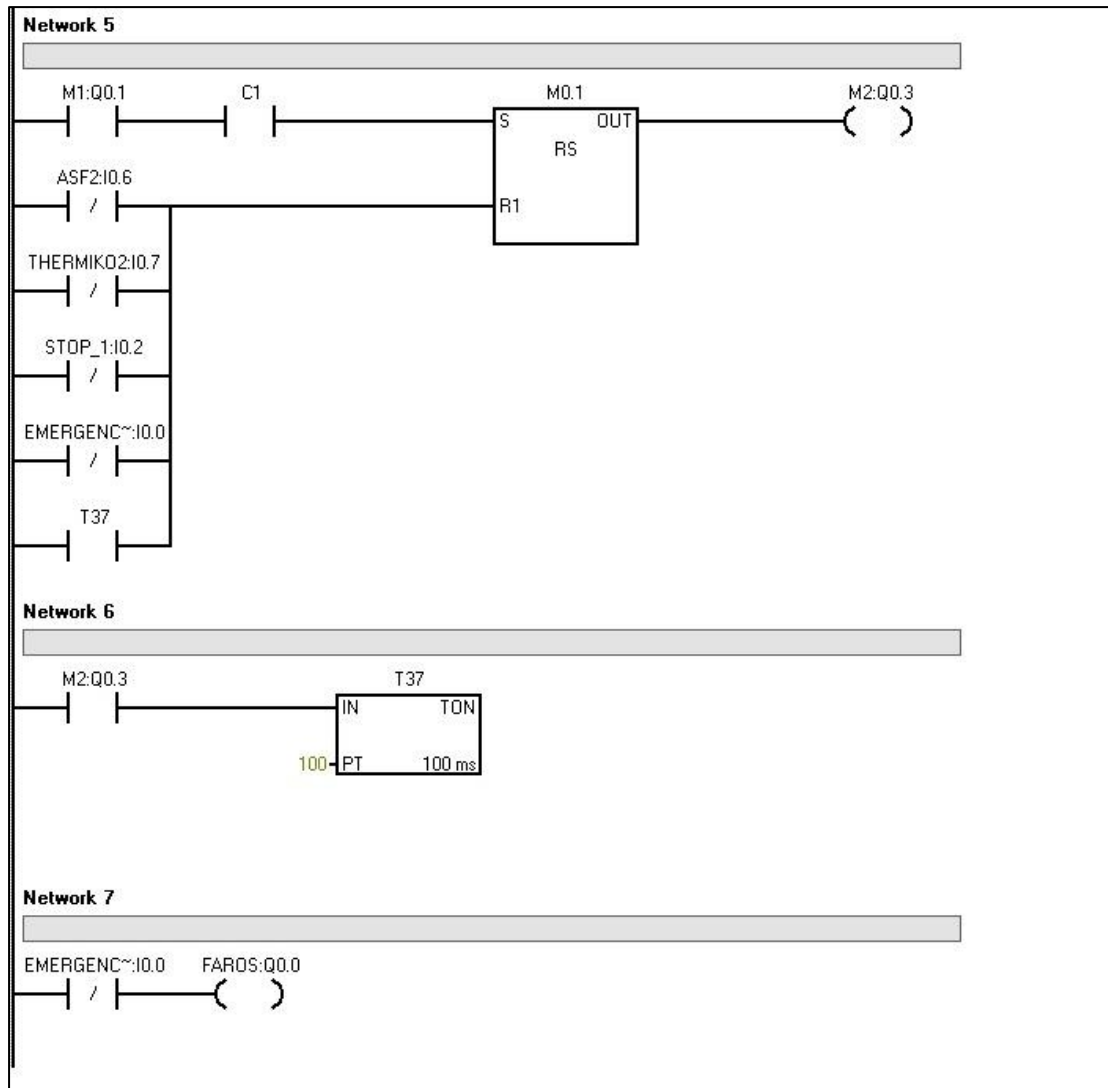
Βήμα 3 – Συγγραφή κώδικα

Αφού μελετήσουμε προσεκτικά τις απαιτήσεις της άσκησης μπορούμε να ξεκινήσουμε τη συγγραφή του κώδικα σε γλώσσα LADDER.

Στη συγκεκριμένη άσκηση θα χρησιμοποιήσουμε της εντολές Reset – Set που περιγράφονται αναλυτικά στην ενότητα 2.2.1. Στην είσοδο Set τοποθετούμαι όλες τις συνθήκες για τις οποίες οι κινητήρες πρέπει να ξεκινάνε. Στην προκειμένη περίπτωση τα Button Start1 και Start2. Ενώ στην είσοδο R όλες τις συνθήκες που οι κινητήρες πρέπει να σταματάνε. Συγκεκριμένα και για τους δύο κινητήρες έχουμε ασφάλειες και θερμικά, οπότε θα πρέπει να τοποθετηθούν στις αντίστοιχες εισόδους R των Reset – Set Flip Flop.



Λύση 5.4



Λύση 5.4

Βήμα 4 – Δοκιμή του προγράμματος

Κατεβάζουμε το πρόγραμμα μας στο PLC σύμφωνα με τα βήματα που περιγράφονται στην ενότητα 3.4.3 (παράγραφος Download) και το τοποθετούμε το PLC από το λογισμικό σε θέση RUN.

Για να διαπιστώσουμε την ορθότητα του προγράμματος, τοποθετούμε τους διακόπτες S1, S2, S3 και S4 στη θέση 1 και να παρατηρήσουμε την πράσινη ένδειξη. Στη συνέχεια πιέζουμε το button Start1 και παρατηρούμε την ενεργοποίηση των φορτίων. Το ρολό του αισθητήρα τον έχει του button Start2 και κάθε φορά που το πιέζουμε θα πρέπει ο μετρητής να αυξάνει κατά 1. Μόλις πατηθεί 6 φορές θα πρέπει να απενεργοποιηθούν και να ενεργοποιηθούν τα αντίστοιχα φορτία.

5.5 Χώρος στάθμευσης 5 θέσεων

Να υλοποιηθεί πρόγραμμα σε γλώσσα LADDER που θα υλοποιεί τον παρακάτω αυτοματισμό :

Κάθε φορά που εισέρχεται ένα αυτοκίνητο στο χώρο στάθμευσης ο οδηγός πιέζει το Button 1. Ο κινητήρας M1 ενεργοποιείται για 6 δευτερόλεπτα, που αντιστοιχεί στην κίνηση της πόρτας εισόδου και για το άνοιγμα και για το κλείσιμο.

Για κάθε αυτοκίνητο που εξέρχεται από το χώρο στάθμευσης, ο οδηγός πιέζει το πλήκτρο button 3. Ο κινητήρας M2 ενεργοποιείται για 6 δευτερόλεπτα, που αντιστοιχεί στην κίνηση της πόρτας εξόδου για το άνοιγμα και το κλείσιμο.

Όσο υπάρχουν διαθέσιμες θέσεις, θα είναι ενεργοποιημένη η πράσινη λυχνία H2. Σε περίπτωση που ο χώρος στάθμευσης είναι πλήρης θα πρέπει να ενεργοποιηθεί η λυχνία H2 και να μην είναι εφικτή η είσοδος κάποιου οχήματος, παρά μόνο αν ελευθερωθεί κάποια θέση. Επίσης, ο εσωτερικός μετρητής θα μπορεί να μηδενίζει οποιαδήποτε στιγμή επιθυμεί ο ιδιοκτήτης πατώντας το button B2 της μονάδας.

Επίσης, σε κάθε κινητήρα που ελέγχει τις πόρτες υπάρχει μια ασφάλεια. Διακόπτες S1 και S2. Όσο οι διακόπτες είναι σε θέση 1 τότε οι κινητήρες μπορούν να λειτουργήσουν. Αν όμως «πέσει» κάποια ασφάλεια, θα πρέπει να ενεργοποιηθεί ο φάρος έτσι ώστε να ενημερωθεί ο ιδιοκτήτης και να επιλυθεί η βλάβη.

Να χρησιμοποιηθούν οι εισοδοί και οι έξοδοι του παρακάτω πίνακα :

ΕΙΣΟΔΟΙ		
A/A	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ
1	ENTRANCE	I0.1
2	EXIT	I0.3
3	ASF1	I0.4
4	ASF2	I0.5
5	RST_Counter	I0

ΕΞΟΔΟΙ		
A/A	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ
1	M1	Q0.1
2	FULL	Q0.2
3	M2	Q0.3
5	OPEN	Q0.5

Βήμα 1 - Ρυθμίσεις επικοινωνίας

Ακολουθούμε τα βήματα της ενότητας 3.4.3

Βήμα 2 – Απόδοση μνημονικών ονομάτων (προαιρετικά)

Όπως περιγράφεται στην ενότητα 3.3.2

Βήμα 3 – Συγγραφή κώδικα

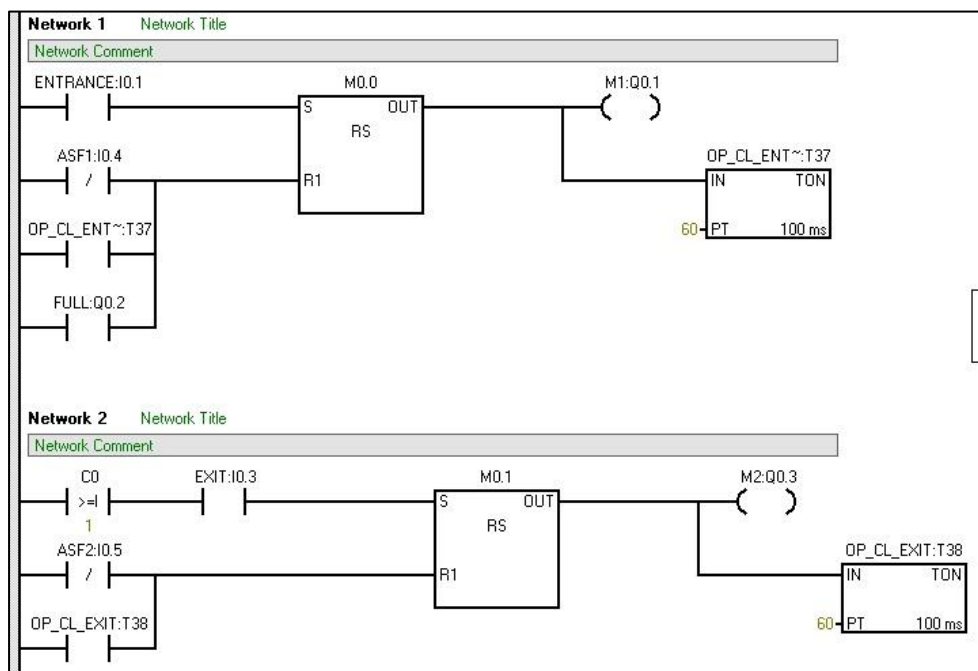
Αφού μελετήσουμε προσεκτικά τις απαιτήσεις της άσκησης μπορούμε να ξεκινήσουμε τη συγγραφή του κώδικα σε γλώσσα LADDER. Στη συγκεκριμένη άσκηση γίνεται χρήση χρονικών, μετρητών, συγκριτών και Set – Reset.

Συγκεκριμένα χρησιμοποιούμε 2 χρονικά τον T37 και T38 τα οποία, αφού πατηθούν τα buttons B1 και B2, ξεκινούν να μετράνε μέχρι 6 δευτερόλεπτα και αντιστοιχούν στην είσοδο και έξοδο των οχημάτων στο χώρο στάθμευσης και στην κίνηση των θυρών. Επίσης, ο Counter C0, που είναι ένας Up-Down μετρητής μετράει μέχρι τον αριθμό 5, που είναι και η χωρητικότητα του χώρου στάθμευσης. Ο μετρητής αυξάνει κατά 1 κάθε φορά που μετράει ο χρόνος που απαιτείται για να ανοίξει και να κλείσει η μπάρα εισόδου και μειώνεται κατά 1 μόλις περάσει ο χρόνος που ανοίγει και κλείνει η μπάρα εξόδου. Ενώ ο μηδενισμός του C0, έχει προγραμματιστεί να γίνεται με το Button B2.

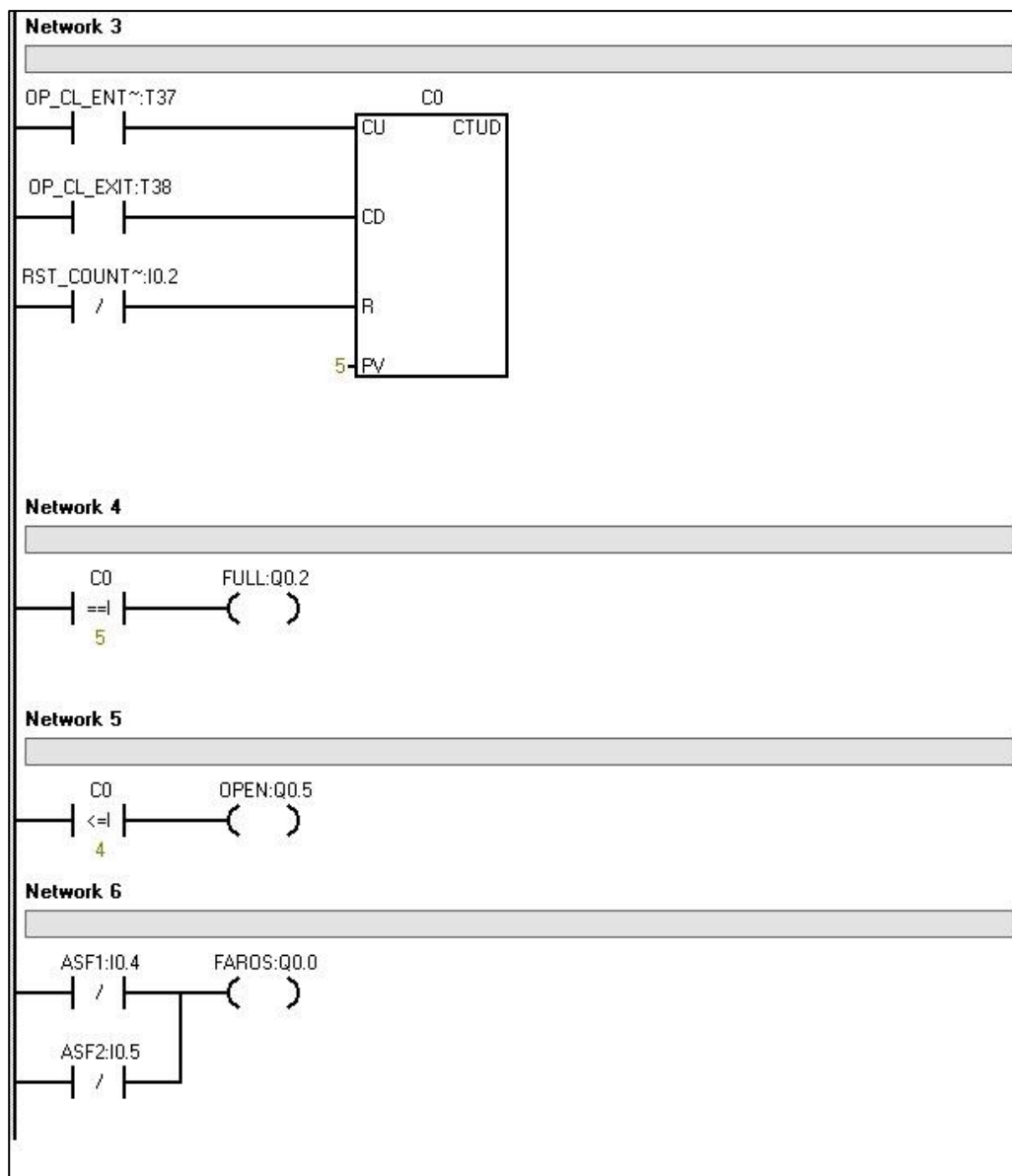
Οι συγκριτές χρησιμοποιούνται για την ενεργοποίηση των ενδείξεων σχετικά με τις διαθέσιμες θέσεις. Έτσι, όταν ο μετρητής έχει τιμή μικρότερη του 5 ενεργοποιείται η πράσινη ένδειξη ενώ όταν ο μετρητής έχει την τιμή 5 (χώρος στάθμευσης πλήρης) ενεργοποιείται η κόκκινη ένδειξη και δεν επιτρέπει την είσοδο σε κάποιο όχημα μέχρι να ελευθερωθεί κάποια θέση.

Τέλος, η ενεργοποίηση του φάρου γίνεται μόλις «πέσει» ασφάλεια από τις πόρτες εισόδου – εξόδου.

Συνεπώς το πρόγραμμα θα πρέπει να είναι όπως το παρακάτω :



Λύση 5.5



Λύση 5.5

Βήμα 4 – Δοκιμή του προγράμματος

Κατεβάζουμε το πρόγραμμα μας στο PLC σύμφωνα με τα βήματα που περιγράφονται στην ενότητα 3.4.3 (παράγραφος Download) και το τοποθετούμε το PLC από το λογισμικό σε θέση RUN.

Για να διαπιστώσουμε την ορθότητα του προγράμματος, τοποθετούμε τους διακόπτες S1 και S2 στη θέση 1, παρατηρούμε την πράσινη ένδειξη που δηλώνει ότι υπάρχουν διαθέσιμες θέσεις. Στη συνέχεια πιέζουμε το button B1 και παρατηρούμε την κίνηση του κινητήρα M1 στη μονάδα ο οποίος θα ενεργοποιηθεί για 6 δευτερόλεπτα. Πιέζουμε ξανά το button B1 όσες φορές χρειαστεί μέχρι να καλυφθούν όλες οι θέσεις, να ενεργοποιηθεί η κόκκινη ένδειξη και να απενεργοποιηθεί η πράσινη. Πιέζουμε το Button B3, αντιστοιχεί στην έξοδο των οχημάτων και παρατηρούμε τις ενδείξεις διαθεσιμότητας. Τέλος, αλλάζουμε την κατάσταση των διακοπών S1 ή S2 για να ενεργοποιηθεί ο φάρος.

Μελλοντικές επεκτάσεις

Όπως έχει ήδη αναφερθεί, ο σκοπός της παρούσας εργασίας είναι καθαρά εκπαιδευτικός και κυρίως εισαγωγικός σε ότι έχει να κάνει με τον αυτοματισμό με PLC. Παρόλο που για το τελικό αποτέλεσμα της συνολικής εκπόνησης της εργασίας χρειάστηκε αρκετός χρόνος, ωστόσο περιοριστήκαμε στην επεξεργασία μόνο ψηφιακών σημάτων. Αυτό έγινε διότι το PLC που χρησιμοποιήσαμε δεν υποστηρίζει την επεξεργασία αναλογικών σημάτων από μόνο του αλλά απαιτεί την χρήση καρτών αναλογικών σημάτων, οι οποίες πρέπει να αγορασθούν και να τοποθετηθούν εκ νέου.

Έτσι, λοιπόν, μια πολύ καλή επέκταση θα αποτελούσε αρχικά η προμήθεια καρτών αναλογικών σημάτων (εισόδων – εξόδων) και κατόπιν η τοποθέτηση και η σύνδεση με το PLC της εφαρμογής μας και η κατασκευή μιας αντίστοιχης πλατφόρμας στην οποία θα μπορεί να γίνει η μελέτη αναλογικών σημάτων με αναλογικά αισθητήρια και την διαχείριση αναλογικών εξόδων. Έτσι, θα μπορεί ο χρήστης να αποκτήσει μια σφαιρική άποψη για τον προγραμματισμό των PLC.

Βιβλιογραφία

1. Νικόλαος Μαραντίνης (2000), *Αυτοματισμός με Simatic S7*, Αθήνα, Εκδόσεις Siemens
2. Νίκ Ι. Ζούλη, *Προγραμματιζόμενοι Λογικοί Ελεγκτές-Εφαρμογές στο PLC S7 - 200*
3. Παρασκευάς Ν. Παρασκευόπουλος (2001), *Εισαγωγή στον αυτόματο έλεγχο – Τόμος Α: Θεωρία*, Πρώτη έκδοση, Αθήνα
4. Hans Berger (2002), *Σχεδίαση εφαρμογών αυτοματισμού με τη γλώσσα Step-7 – Προγραμματιζόμενοι Λογικοί Ελεγκτές S7-300 / S7-400*, Δεύτερη έκδοση, Εκδόσεις Τζιόλα, Θεσσαλονίκη
5. Frank D. Petruzella (2010), *Προγραμματιζόμενοι Λογικοί Ελεγκτές*, Δεύτερη έκδοση, Εκδόσεις Τζιόλα, Θεσσαλονίκη
6. Dennis Collins - Eamonn Lane (1997), *Προγραμματιζόμενοι Λογικοί Ελεγκτές – Ένας πρακτικός οδηγός*, Εκδόσεις Τζιόλας, Θεσσαλονίκη
Νικ. Α. Πανταζής (2001), *PLC – Προγραμματιζόμενοι Λογικοί Ελεγκτές*, Τρίτη έκδοση, Εκδόσεις Ίων, Αθήνα

Ηλεκτρονικά εγχειρίδια :

1. *Simatic S7-200 Programmable Controller System Manual*, Siemens
2. *Βιομηχανικοί αυτοματισμοί με PLC - Εισαγωγικό εγχειρίδιο & παραδείγματα εφαρμογών*, Siemens