



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΝΑΥΠΗΓΩΝ ΝΗΧΑΝΙΚΩΝ**

**Διπλωματική Εργασία**

Ανάπτυξη μονάδων εκμάθησης προγραμματισμού/διασύνδεσης  
βιομηχανικού τύπου ελεγκτή (*PLC*).

**Συγγραφέας:**

Ειρήνη Π. Μιχελή

**ΑΜ:** 51113018

**Επιβλέπων:**

Δημήτριος Νικόλαος Παγώνης

Αιγάλεω, Νοέμβριος 2021

**- Η σελίδα αυτή είναι σκοπίμως κενή -**



**UNIVERSITY OF WEST ATTICA  
SCHOOL OF ENGINEERING  
DEPARTMENT OF NAVAL ARCHITECTURE**

## **Diploma Thesis**

Development of PLC programming training boards.

**Student name and surname:**

Eirini Micheli

**Registration Number:** 51113018

**Supervisor name and surname:**

Dimitrios Nikolaos Pagonis

**Aigaleo, November 2021**

**- This page intentionally left blank –**



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΝΑΥΠΗΓΩΝ ΜΗΧΑΝΙΚΩΝ**

**Διπλωματική εργασία:**

Ανάπτυξη μονάδων εκμάθησης προγραμματισμού/διασύνδεσης βιομηχανικού τύπου ελεγκτή (PLC).

**Συγγραφέας:**

Ειρήνη Π. Μιχελή (51113018)

**Επιβλέπων:**

Δημήτριος Νικόλαος Παγώνης,  
Αναπληρωτής Καθηγητής, ΠΑ.Δ.Α

**Ημερομηνία εξέτασης:** 08/11/2021

**Εξεταστική Επιτροπή:**

**Δημήτριος Ν. Παγώνης,**  
**Αναπληρωτής Καθηγητής**  
**ΠΑ.Δ.Α**

**Σοφία Πέππα,**  
**Επίκουρος Καθηγήτρια**  
**ΠΑ.Δ.Α**

**Γεώργιος Λιβανός,**  
**Αναπληρωτής Καθηγητής**  
**ΠΑ.Δ.Α**

**- Η σελίδα αυτή είναι σκοπίμως κενή -**

## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Η κάτωθι υπογεγραμμένη **Μιχελή Ειρήνη** του **Παναγιώτη**, με αριθμό μητρώου **51113018** φοιτήτρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής **Μηχανικών** του **Τμήματος Ναυπηγών Μηχανικών**, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Η Δηλούσα



(Υπογραφή)

## Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της διπλωματικής μου εργασίας, Δρ. Δημήτριο - Νικόλαο Παγώνη, καθηγητή του τμήματος Ναυπηγών Μηχανικών, για την ευκαιρία που μου έδωσε να ασχοληθώ και να διευρύνω παράλληλα τις γνώσεις μου γύρω από το θέμα του προγραμματισμού και τη λειτουργία των προγραμματιζόμενων λογικών ελεγκτών, αλλά και για την άριστη συνεργασία που είχαμε καθ' όλη τη διάρκεια εκπόνησής της.

Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου, και κυρίως τον πατέρα μου, για την ψυχολογική και οικονομική ενίσχυση κατά τη διάρκεια της εκπόνησης της παρούσας διπλωματικής εργασίας.



## Περιεχόμενα

Περίληψη .....	12
Abstract.....	14
Εισαγωγή .....	16
Κεφάλαιο 1 <sup>ο</sup> : Ιστορική Αναδρομή.....	18
Κεφάλαιο 2 <sup>ο</sup> : Συστήματα Αυτόματου Ελέγχου (Σ.Α.Ε).....	20
2.1 Συστήματα ανοικτού βρόγχου ( <i>open loop</i> ).....	20
2.2 Συστήματα κλειστού βρόγχου ( <i>closed loop</i> ).....	21
2.3 Αξιολόγηση συστημάτων αυτόματου ελέγχου .....	22
2.4 Εισαγωγή στις λογικές πύλες.....	24
Κεφάλαιο 3 <sup>ο</sup> : Προγραμματιζόμενοι Λογικοί Ελεγκτές (PLC).....	29
3.1 Πλεονεκτήματα χρήσης ΠΛΕ.....	29
3.2 Μειονεκτήματα χρήσης ΠΛΕ .....	30
3.3 Η δομή ενός προγραμματιζόμενου λογικού ελεγκτή (PLC).....	30
3.4 Λειτουργία ενός Προγραμματιζόμενου Λογικού Ελεγκτή (PLC) .....	32
3.5 Γλώσσα προγραμματισμού ενός Προγραμματιζόμενου Λογικού Ελεγκτή (PLC).....	34
3.5.1 Γλώσσα προγραμματισμού Ladder Logic (Γλώσσα ηλεκτρολογικών γραφικών).....	35
3.5.2 Γλώσσα προγραμματισμού Statement List (Γλώσσα λογικών εντολών).....	35
3.5.3 Γλώσσα προγραμματισμού Functional Block Diagram (Γλώσσα λογικών γραφικών ή λογικού διαγράμματος) .....	36
3.5.4 Γραφικά στοιχεία της γλώσσας προγραμματισμού Ladder .....	37
Κεφάλαιο 4 <sup>ο</sup> : Δημιουργία νέου project με χρήση του λογισμικού TIA Portal V13 .....	47
4.1 Απαραίτητα βήματα για τη δημιουργία νέου κώδικα/project.....	48
4.2 Εγκαθίδρυση επικοινωνίας PC – PLC και ορισμός IP address.....	53
Κεφάλαιο 5 <sup>ο</sup> : Σχεδίαση πρότυπης εκπαιδευτικής μονάδας με χρήση του λογισμικού Rhinoceros.....	59
Κεφάλαιο 6 <sup>ο</sup> : Κατασκευή και συναρμολόγηση μονάδων εκμάθησης προγραμματισμού και διασύνδεσης βιομηχανικού τύπου PLC.....	65
6.1 Κατασκευή προτύπου μονάδας εκμάθησης.....	65
6.2 Κατασκευή μονάδων εκμάθησης.....	71
6.3 Χωροταξική διαρρύθμιση των στοιχείων της μονάδας .....	77
Κεφάλαιο 7 <sup>ο</sup> : Χρήση τρισδιάστατου εκτυπωτή στην οπτικοποίηση της εκπαιδευτικής μονάδας.....	83
7.1 Προετοιμασία του τρισδιάστατου εκτυπωτή για την εκτύπωση.....	90
Κεφάλαιο 8 <sup>ο</sup> : Υλικά και κόστος παραγωγής εκπαιδευτικών μονάδων.....	92
8.1 Μονάδες εκμάθησης PLC της εταιρίας GTEE και του ΠΑ.Δ.Α .....	94
Κεφάλαιο 9 <sup>ο</sup> : Αισθητήρες (Sensors) και Ηλεκτρονόμος (Relay) .....	97
9.1 Εισαγωγή στους ηλεκτρονόμους .....	97
9.2 Εισαγωγή στους αισθητήρες .....	101

9.2.1	Χαρακτηριστικά Αισθητήρων.....	102
9.2.2	Συνδεσμολογία αισθητήρων .....	103
9.2.3	Αισθητήρες στροφών.....	103
9.2.4	Αισθητήρες ροής.....	104
9.2.5	Αισθητήρια στάθμης υγρού .....	105
Κεφάλαιο 10 <sup>ο</sup> : Διάταξη ελέγχου στάθμης και μεταφοράς ρευστού σε δεξαμενές .....		106
10.1	Εισαγωγή.....	106
10.2	Διάταξη δικτύου δεξαμενών εκπαιδευτικών μονάδων .....	107
Κεφάλαιο 11 <sup>ο</sup> : Έλεγχος ασύγχρονου κινητήρα .....		110
11.1	Αρχή λειτουργίας ασύγχρονου τριφασικού κινητήρα.....	112
11.2	Εκκινήτης Αστέρα-Τριγώνου.....	112
Κεφάλαιο 12 <sup>ο</sup> : Παράρτημα .....		118
12.1	Εισαγωγή.....	118
12.2	Άσκηση 1: Εξοικείωση με το περιβάλλον «TIA Portal V13» και τη γλώσσα προγραμματισμού Ladder.....	118
12.2.1	Επεξήγηση ενδεικτικού κώδικα .....	122
12.2.2	Συνδεσμολογία εκπαιδευτικής μονάδας .....	129
12.3	Άσκηση 2: Έλεγχος στάθμης και μεταφοράς ρευστού σε δεξαμενές.....	137
12.3.1	Επεξήγηση ενδεικτικού κώδικα .....	138
12.3.2	Συνδεσμολογία εκπαιδευτικής μονάδας .....	146
12.4	Άσκηση 3: Εκκίνηση τριφασικού κινητήρα με χρήση προγραμματιζόμενου λογικού ελεγκτή 151 .....	
12.4.1	Επεξήγηση ενδεικτικού κώδικα .....	155
12.4.2	Συνδεσμολογία εκπαιδευτικής μονάδας .....	165
Κεφάλαιο 13 <sup>ο</sup> : Συμπεράσματα.....		170
Κεφάλαιο 14 <sup>ο</sup> : Προτάσεις.....		171
Βιβλιογραφική Αναφορά .....		172

## Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η κατασκευή κατάλληλων εκπαιδευτικών μονάδων, με στόχο τη χρήση τους στα πλαίσια εργαστηριακού μαθήματος του προγράμματος σπουδών του Πανεπιστημίου Δυτικής Αττικής του τμήματος Ναυπηγών Μηχανικών. Αρχικά, πραγματοποιήθηκε ο σχεδιασμός και η κατασκευή κατάλληλης πρότυπης μονάδας με χρήση του λογισμικού *Rhinoceros*.

Στη συνέχεια, πραγματοποιήθηκε εξοικείωση με τον προγραμματιζόμενο λογικό ελεγκτή (*PLC*) *S-1200* της εταιρίας *SIEMENS* και με το αντίστοιχο λογισμικό πακέτο περιβάλλοντος εργασίας «*Simatic Step 7*», χρησιμοποιώντας ως γλώσσα προγραμματισμού τη *Ladder*. Τέλος, αναπτύχθηκαν τυπικές εφαρμογές, σε μορφή εργαστηριακών ασκήσεων. Πιο συγκεκριμένα, οι εφαρμογές αυτές αναφέρονται στην εκκίνηση ενός τριφασικού ασύγχρονου κινητήρα, στη χρήση αισθητήρων για τη μέτρηση των στροφών της μηχανής και στον έλεγχο στάθμης και μεταφοράς ρευστού μεταξύ δύο δεξαμενών.

**Λέξεις κλειδιά:** Προγραμματιζόμενος Λογικός Ελεγκτής, ΠΛΕ, εκπαιδευτικές μονάδες εκμάθησης, Siemens, ναυτική μηχανολογία, Γλώσσα προγραμματισμού Ladder, αισθητήρες, διάταξη ελέγχου στάθμης, ασύγχρονος τριφασικός κινητήρας.

**- Η σελίδα αυτή είναι σκοπίμως κενή -**

## **Abstract**

*The main purpose of this thesis was the design and construction of appropriate PLC-programming training boards. The specific units will be employed for educational purposes at the undergraduate/postgraduate program of the Naval Architecture department. Firstly, an appropriate prototype board was designed and constructed, using the 3D CAD software Rhinoceros.*

*In the next step, familiarization with programming the PLC S-1200 unit followed, utilizing the SIEMENS software package “Simatic Step 7”; the type of the programming language preferred was Ladder. Finally, typical applications employing the PLC S-1200 have been developed and analyzed in the form of laboratory exercises. In more detail, these exercises are relative to engine speed measurement, control of tank level and electric motor starting.*

**Key words:** *Programmable Logic Controller, PLC, programming training board, Siemens, marine engineering, Ladder Logic, sensors, level control device, three – phase electric motor.*

**- This page intentionally left blank –**

## Εισαγωγή

Γενικότερα ως «Αυτοματισμός», θεωρείται το πεδίο της επιστήμης και της τεχνολογίας που ασχολείται με τη τυποποίηση μίας διαδικασίας μέσω της εύρεσης καλώς ορισμένων βημάτων/εντολών τα οποία πρέπει να ακολουθηθούν για να παραχθεί κάποιο επιθυμητό αποτέλεσμα. Συνεπώς το αντικείμενο του αυτοματισμού, είναι γενικό και πολύπλευρο, ενώ πολυάριθμες εφαρμογές του βρίσκονται στη καθημερινή ζωή και στη βιομηχανία, με αποτέλεσμα να αποκτά σημαντικό ρόλο στη ζωή μας.

Η ραγδαία εξέλιξη της τεχνολογίας και της πληροφορικής σε συνδυασμό με τις αυξημένες ανάγκες της σημερινής κοινωνίας, επηρεάζει άμεσα όλες τις σύγχρονες αυτοματοποιημένες λειτουργίες σε επαγγελματικό και ατομικό επίπεδο, δημιουργούν μεγαλύτερες απαιτήσεις ποσότητας και ποιότητας που μόνο με την εφαρμογή της πληροφορικής και ειδικά των συστημάτων αυτομάτου ελέγχου θα μπορούσαν να ικανοποιηθούν. Αυτό έχει ως αποτέλεσμα την ανάγκη για τη στελέχωση των εταιριών και των βιομηχανιών με εξειδικευμένο προσωπικό, ώστε να μπορεί να ανταπεξέλθει στις απαιτήσεις των αυτοματοποιημένων μέσων είτε ελέγχοντας εάν το σύστημα λειτουργεί ομαλά, επιδιορθώνοντας τυχόν βλάβες, είτε με την ανάγκη προγραμματισμού ενός συστήματος από την αρχή.

Η παρούσα διπλωματική εργασία αποτελείται από 14 κεφάλαια:

Στο 1<sup>ο</sup> κεφάλαιο γίνεται μια σύντομη αναφορά στην εξέλιξη των αυτοματισμών στο πέρασμα του χρόνου.

Το 2<sup>ο</sup> κεφάλαιο περιλαμβάνει εισαγωγικά στοιχεία για τα συστήματα αυτόματου ελέγχου (Σ.Α.Ε.). Συγκεκριμένα πραγματοποιείται μια σύντομη αναφορά στις κατηγορίες τις οποίες τα διακρίνουμε καθώς και στα είδη λογικών πυλών και τα χαρακτηριστικά τους.

Το 3<sup>ο</sup> κεφάλαιο αναφέρεται στους προγραμματιζόμενους λογικούς ελεγκτές (*PLC*) και πιο συγκεκριμένα στον ελεγκτή S-1200 ο οποίος χρησιμοποιήθηκε και στις μονάδες εκμάθησης. Επιπλέον γίνεται μια αναφορά στις γλώσσες προγραμματισμού ενός *PLC*, εμβαθύνοντας στη γλώσσα *Ladder* η οποία χρησιμοποιείται για τον προγραμματισμό του ελεγκτή του εργαστηρίου.

Στο 4<sup>ο</sup> κεφάλαιο αναφέρονται τα βήματα δημιουργίας ενός «*project*» στο περιβάλλον εργασίας «*TIA PORTAL V13*» καθώς και τα απαιτούμενα βήματα σύνδεσης ενός προγραμματιζόμενου λογικού ελεγκτή (*PLC*) με έναν ηλεκτρονικό υπολογιστή (H/Y), με σκοπό την ανάπτυξη απαιτούμενου κώδικα.

Στο 5<sup>ο</sup> κεφάλαιο γίνεται αναφορά στο τρόπο σχεδίασης της μονάδας εκμάθησης *PLC* κάνοντας χρήση κατάλληλου σχεδιαστικού προγράμματος (*Rhinoceros*), ώστε να ελεγχθεί η λειτουργικότητα της μονάδας.

Στο 6<sup>ο</sup> κεφάλαιο αναφέρονται τα βήματα και ο τρόπος κατασκευής των μονάδων εκμάθησης καθώς και η κατασκευή μιας πρότυπης μονάδας για έλεγχο λειτουργικότητας.

Το 7<sup>ο</sup> κεφάλαιο περιλαμβάνει τη διαδικασία για τη χρήση του τρισδιάστατου εκτυπωτή «*Cube X*» του εργαστηρίου. Ο εκτυπωτής χρησιμοποιήθηκε για την εκτύπωση αντικειμένων (*Star – Delta*), τα οποία ήταν μέρος της οπτικοποίησης των μονάδων εκμάθησης.

Στο 8<sup>ο</sup> κεφάλαιο γίνεται μια σύντομη αναφορά στα απαραίτητα υλικά και τις ποσότητες που χρησιμοποιήθηκαν για την υλοποίηση των εκπαιδευτικών μονάδων, καθώς και το συνολικό κόστος της κατασκευής. Τέλος, θα αναφέρονται οι ομοιότητες και οι διαφορές της μονάδας που κατασκευάστηκε σε σχέση με μια παρόμοια μονάδα εκμάθησης του εμπορίου.

Το 9<sup>ο</sup> κεφάλαιο αναφέρεται στη λειτουργία των ηλεκτρονόμων και στους αισθητήρες. Γενικότερα γίνεται αναφορά στα χαρακτηριστικά των αισθητήρων (τα οποία αποτελούν κριτήριο επιλογής) στο τρόπο συνδεσμολογίας τους και στη λειτουργία τους. Τέλος, γίνεται αναφορά επίσης σε συγκεκριμένους αισθητήρες που χρησιμοποιούνται στις ασκήσεις (ροής, στάθμης και στροφών).

Στο 10<sup>ο</sup> κεφάλαιο γίνεται αναφορά στη λειτουργία του τμήματος δικτύου έρματος του εργαστηρίου «*Ναυτικών μηχανών και εργαλειομηχανών*». Επιπλέον, γίνεται αναφορά στη λειτουργία μιας ήδη υπάρχουσας εγκατάστασης έλεγχου στάθμης και μεταφορά έρματος μεταξύ δύο δεξαμενών, του ίδιου εργαστηρίου με χρήση ελεγκτή. Τέλος, παρουσιάζεται η διάταξη του «δικτύου έρματος» το οποίο θα χρησιμοποιηθεί στις εκπαιδευτικές μονάδες.

Στο 11<sup>ο</sup> κεφάλαιο γίνεται αναφορά στη λειτουργία ενός ασύγχρονου τριφασικού κινητήρα καθώς και στην απαιτούμενη συνδεσμολογία των τυλιγμάτων για τη συνδεσμολογία αστέρα και τριγώνου ( $Y - \Delta$ ). Επιπλέον, γίνεται αναφορά στη λειτουργία μιας ήδη υπάρχουσας εγκατάστασης λειτουργίας του κινητήρα του τμήματος με χρήση ελεγκτή. Τέλος γίνεται αναφορά στη συνδεσμολογία «αστέρα - τριγώνου» η οποία θα χρησιμοποιηθεί στις εκπαιδευτικές μονάδες.

Στο 12<sup>ο</sup> κεφάλαιο γίνεται αναφορά στην ανάπτυξη τυπικών εφαρμογών με χρήση των μονάδων που αναπτύχθηκαν, με τη μορφή εργαστηριακών ασκήσεων. Αναφέρονται τα βήματα και η εξήγηση του κώδικα σε γλώσσα *Ladder* των βημάτων, καθώς και η σωστή συνδεσμολογία της μονάδας εκμάθησης για την ομαλή λειτουργία του συστήματος με το επιθυμητό αποτέλεσμα.

Το 13<sup>ο</sup> κεφάλαιο περιλαμβάνει χρήσιμα συμπεράσματα και παρατηρήσεις, τα οποία προκύπτουν από τη περάτωση αυτής της διπλωματικής εργασίας.

Τέλος στο 14<sup>ο</sup> κεφάλαιο υπάρχουν κάποιες προτάσεις τροποποίησης/εξέλιξης της παρούσας διπλωματικής εργασίας.



## Κεφάλαιο 1<sup>ο</sup>: Ιστορική Αναδρομή

Στις μέρες μας διανύουμε την εποχή με την μεγαλύτερη τεχνολογική ανάπτυξη. Κάθε μέρα ανακαλύπτονται νέα πιο αυτοματοποιημένα μέσα με σκοπό την βελτίωση του ανθρώπινου βίου και την αυτοματοποίηση της βιομηχανικής παραγωγής.

Τα συστήματα του αυτοματισμού φαίνεται ότι είχαν κάνει την εμφάνισή τους αιώνες πριν, στα χρόνια των αρχαίων Ελλήνων με χρήση του αυτόματου ρυθμιστή της στάθμης του νερού στις δεξαμενές. Στη συνέχεια, το 5<sup>ο</sup> αιώνα π.Χ. ο Αρχιμήδης χρησιμοποίησε τον παραπάνω μηχανισμό για τη δημιουργία του υδραυλικού ωρολογίου. Αργότερα ο Ήρων ο Αλεξανδρινός (1<sup>ο</sup> αιώνα μ.Χ.) δημιούργησε διάφορα συστήματα αυτοματισμού τα οποία βασίζονταν στην υδραυλική πίεση. Το πιο γνωστό του σύστημα ήταν η αιολόσφαιρας ή ρυθμιστής (θεωρείται η πρώτη ατμομηχανή), η οποία αποτελούσε σύστημα για την αυτοματοποίηση των θυρών των ναών καθώς και στη δημιουργία ενός αυτόματου θεάτρου, δημιουργώντας ειδικά εφέ και άμεση εναλλαγή των σκηνών.

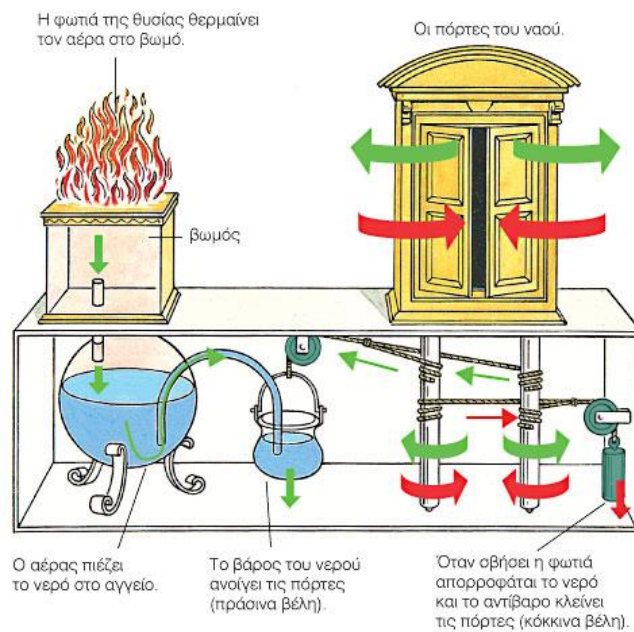
Κατά το έτος 1690 επινοήθηκε από τον *Denis Papin* ο αυτόματος ρυθμιστής πίεσης για λέβητες. Ο *James Watt* (1769) κατάφερε να ρυθμίσει αυτόματα την ταχύτητα της ατμομηχανής χρησιμοποιώντας το φυγοκεντρικό ρυθμιστή ταχύτητας των ατμομηχανών ή αλλιώς ρυθμιστή με σφαίρες. Με το μηχανισμό αυτό ρυθμίζεται το άνοιγμα της βαλβίδας και η ποσότητα του ατμού που εισέρχεται στη μηχανή. Ο *James Clerk Maxwell* (1868) δημοσίευσε ένα μαθηματικό μοντέλο για ένα ρυθμιστή ελέγχου μιας ατμομηχανής, στα πρακτικά της Βασιλικής Εταιρείας. Ο *Henry Ford* σχεδίασε την πρώτη κινητή γραμμή παραγωγής (1913) και έφερε επανάσταση στην κατασκευαστική διαδικασία. Σκοπός του *Ford* ήταν να παράγει το μεγαλύτερο αριθμό αυτοκινήτων, με το απλούστερο σχέδιο, για το χαμηλότερο δυνατό κόστος. Ο *Harry Nyquist* (1932) αναπτύσσει μια μέθοδο για την ανάλυση της ευστάθειας των συστημάτων αυτόματου ελέγχου. Ο *Hendrik Wade Bode* (1945) αναλύει τους ενισχυτές ανάδρασης (*feedback amplifiers*), εισάγοντάς μας σε αυτό που είναι γνωστό ως διάγραμμα *Bode*, για τον σχεδιασμό και την ανάλυση ενός Σ.Α.Ε.

Ο πρωτότυπος σχεδιασμός για το *PLC* (*programmable logic controller*), που επινοήθηκε από τους *Bradford Associates*, βασίστηκε σε ένα έγγραφο που γράφτηκε από έναν μηχανικό που ονομάζεται *Edward R. Clark*. Ο *Richard Morley* γνωστός και ως *Dick Morley*, ο οποίος βοήθησε στη παραγωγή του πρώτου *PLC* (1968) για τη *General Motors*, θεωρήθηκε ο πατέρας του *PLC*.

Σήμερα το *PLC* είναι μια συσκευή που βασίζεται σε κατάλληλο μικροεπεξεργαστή και αποτελεί κυρίαρχο μέσο για την αυτοματοποίηση βιομηχανικών ηλεκτρομηχανολογικών διαδικασιών. Παρόλο που η τεχνολογία για το *PLC* έχει καταστεί πιο εξελιγμένη τα τελευταία χρόνια, ο σκοπός της παραμένει σχεδόν ο ίδιος, με τις προσθήκες της αναλογικής παρακολούθησης, όπως τον έλεγχο της κίνησης σε υψηλή ταχύτητα και την ανταλλαγή δεδομένων μέσω δικτύων επικοινωνίας.



Εικόνα 1: Υδραυλικό ρολόι του Αρχιμήδη (<http://kotsanas.com>).



Εικόνα 2: Αιολόσφαιρα του Ήρωνος (<https://1gymkomotrobotics.wordpress.com>).

## Κεφάλαιο 2<sup>ο</sup>: Συστήματα Αυτόματου Ελέγχου (Σ.Α.Ε)

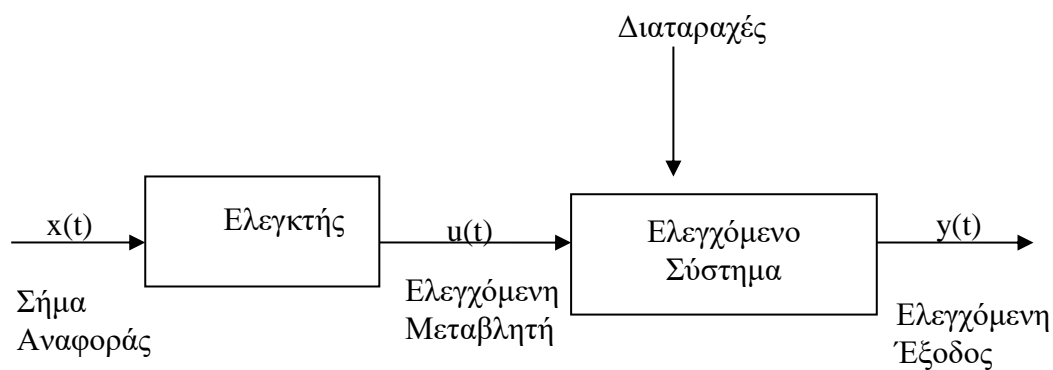
Τα τελευταία χρόνια η επέμβαση των συστημάτων αυτόματου ελέγχου, τόσο στην καθημερινότητα των ανθρώπων όσο και στην πλειοψηφία όλων των σύγχρονων εφαρμογών, έχει αυξηθεί ραγδαία. Ένας από τους κύριους σκοπούς της τεχνολογίας του αυτοματισμού είναι η κατά το δυνατόν μείωση της ανθρώπινης παρέμβασης στη λειτουργία των μηχανημάτων. Αυτό έχει ως σκοπό τον περιορισμό εκτέλεσης χρονοβόρων, κοπιαστικών και επαναλαμβανόμενων δράσεων, ενώ παράλληλα αυξάνουν τον αριθμό παραγωγής, μειώνοντας το σφάλμα που προκαλείται από τον ανθρώπινο παράγοντα.

Έτσι, συστήματα αυτόματου ελέγχου (Σ.Α.Ε) ονομάζουμε κάθε διάταξη φυσικών στοιχείων, τα οποία είναι διασυνδεδεμένα με τέτοιο τρόπο ώστε να καθοδηγούν, ελέγχουν ή ρυθμίζουν τον εαυτό τους ή άλλα συστήματα, ώστε να λειτουργούν αυτόματα με ένα προκαθορισμένο τρόπο. Τα συστήματα αυτόματου ελέγχου διακρίνονται σε δύο βασικές κατηγορίες: α) ανοικτού βρόγχου (*open loop*) και β) κλειστού βρόγχου (*closed loop*) ανάλογα με το εάν η είσοδος  $u(t)$  αποτελεί συνάρτηση της εξόδου  $y(t)$ .

### 2.1 Συστήματα ανοικτού βρόγχου (*open loop*)

Ανοικτό σύστημα είναι ένα σύστημα όπου η είσοδος  $x(t)$  δεν είναι συνάρτηση της εξόδου  $y(t)$ . Ένα σύστημα ανοικτού βρόχου ρυθμίζει την απόκριση μιας εγκατάστασης χωρίς τη χρήση της ανάδρασης (αισθητήρων). Στο ανοικτό σύστημα δεν υπάρχει ανατροφοδότηση. Αντίθετα χρησιμοποιείται μία ενεργή συσκευή, που παράγει το σήμα εισόδου, για να ελέγξει απευθείας την διεργασία.

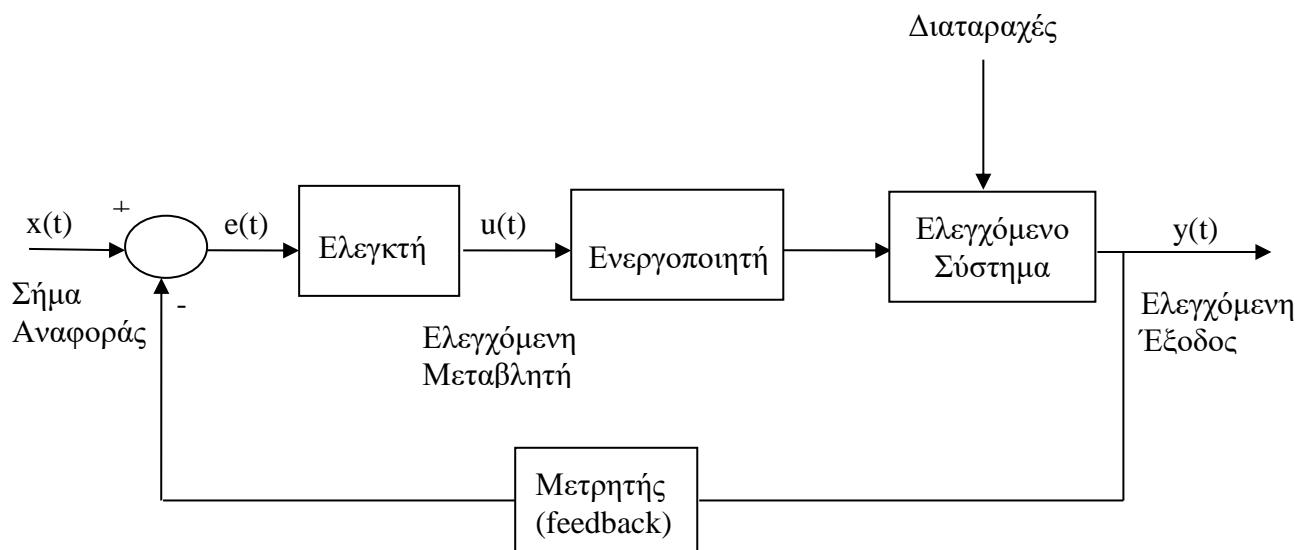
Τα συστήματα ανοικτού βρόχου είναι ευαίσθητα στις διαταραχές από το περιβάλλον του συστήματος και στις μεταβολές των παραμέτρων του συστήματος.



## 2.2 Συστήματα κλειστού βρόχου (closed loop)

Σύστημα κλειστού βρόχου είναι εκείνο στο οποίο η ποσότητα που ενεργοποιεί το σύστημα εξαρτάται από την έξοδο. Τα συστήματα ελέγχου κλειστού βρόχου ονομάζονται και συστήματα ανάδρασης. Ένα σύστημα κλειστού βρόχου χρησιμοποιεί ανάδραση (*feedback*) μεταβλητής και σύγκρισή της με την επιθυμητή τιμή, με σκοπό την επίτευξη της επιθυμητής απόκρισης (προσέγγιση επιθυμητού αποτελέσματος).

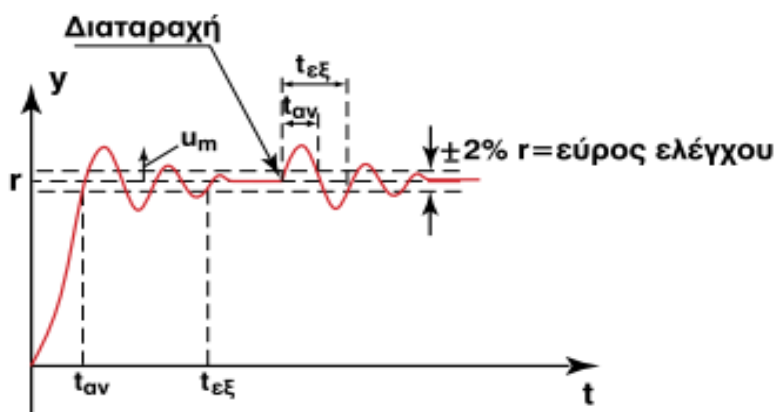
Ένα σύστημα μπορεί να χρησιμοποιεί πολλές αναδράσεις. Κάθε σύστημα ανάδρασης αποτελείται από αισθητήρες που μετρούν τις εξόδους για να διασφαλίσουν ότι οι ενεργοποιητές εκτελούν το σκοπό τους, με αποτέλεσμα να έχουν μεγάλη ακρίβεια σαν σύστημα ελέγχου, εξασφαλίζοντας την ευστάθειά του. Η ανάδραση είναι το χαρακτηριστικό των συστημάτων κλειστού βρόχου η οποία επιτρέπει να γίνεται σύγκριση του σήματος εξόδου του με την είσοδο στο σύστημα. Εάν δεν υπάρχει επικοινωνία μεταξύ εισόδου και εξόδου, έχουμε σύστημα ανοικτού βρόχου, ενώ όταν ελέγχουμε την έξοδο και την οδηγούμε στο επιθυμητό αποτέλεσμα, έχουμε σύστημα κλειστού βρόχου.



## 2.3 Αξιολόγηση συστημάτων αυτόματου ελέγχου

Σε αντίθεση με τα ανοιχτά συστήματα ελέγχου, των οποίων η έξοδος δεν αποτελεί συνάρτηση της τιμής της εισόδου, τα κλειστά συστήματα ελέγχου αξιολογούνται ως προς την ικανότητα και την ποιότητα ελέγχου τους. Τα βασικά κριτήρια αξιολόγησης ενός κλειστού συστήματος είναι:

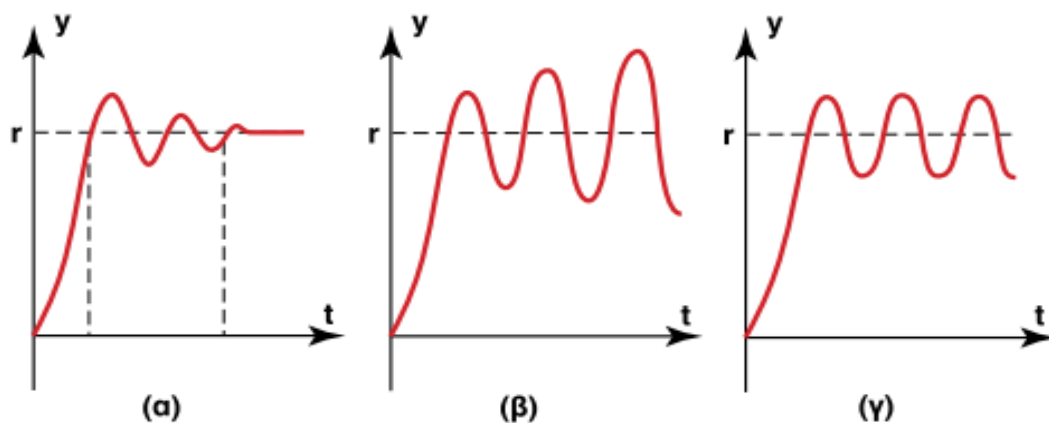
1. Η ακρίβεια ελέγχου (*control accuracy*): Σε ένα κλειστό σύστημα ελέγχου, το οποίο βρίσκεται σε στατική κατάσταση, μόλις επιδράσει μια διαταραχή από το περιβάλλον του συστήματος ή αλλάξει η επιθυμητή τιμή, και η ελεγχόμενη τιμή μετά από μια ταλάντωση γίνει ξανά ίση με την επιθυμητή τιμή, τότε το σύστημα είναι ακριβές, επειδή μηδενίζεται το σφάλμα.
2. Ο χρόνος απόκρισης (χρόνος ανόδου): Σε ένα κλειστό σύστημα ελέγχου, το οποίο βρίσκεται σε στατική κατάσταση, όταν η ελεγχόμενη μεταβλητή εισέρθει σε ένα ευρύ πλάτος ( $\pm 2\%$ ) της επιθυμητής τιμής, προκύπτει ο χρόνος ανόδου ( $T_{av}$ ) και ο χρόνος καθόδου ( $T_{εξ}$ ). Γνωρίζοντας τους χρόνους αυτούς, μπορούμε να αξιολογήσουμε τη δυναμική συμπεριφορά του συστήματος. Εάν οι χρόνοι είναι μικροί, τότε το σύστημα θεωρείται γρήγορο.



Εικόνα 3: Βηματική χρονική απόκριση για τον ορισμό του εύρους ελέγχου και των χρόνων  $T_{av}$  και  $T_{εξ}$  (<http://edume.myds.me>).

3. Ο βαθμός απόσβεσης: Καθορίζει τη μορφή ταλάντωσης του σήματος εξόδου του συστήματος. Ένα χαρακτηριστικό μέγεθος της ταλάντωσης είναι η μέγιστη υπερέψωση ( $U_m$ ), δηλαδή η μέγιστη απόκλιση της ταλάντωσης από την επιθυμητή τιμή.

4. Ευστάθεια συστήματος: Η επίτευξη της ευστάθειας του κλειστού συστήματος αποτελεί ένα πρόβλημα λόγω της συνεχούς διόρθωσης των λαθών που δημιουργούνται στο σύστημα, με άμεση συνέπεια την δημιουργία ταλαντώσεων σταθερού ή μεταβαλλόμενου πλάτους. Ένα αυτόματο σύστημα κλειστού ελέγχου λέμε ότι είναι ευσταθές, όταν μετά από μια μεταβολή του συστήματος (διαταραχή, μεταβολή επιθυμητής τιμής, κ.ά.) η ελεγχόμενη μεταβλητή τείνει με τη πάροδο του χρόνου σε μια σταθερή κατάσταση. Εάν η ελεγχόμενη μεταβλητή διατηρεί σταθερή τη διαταραχή χωρίς να επανέρχεται, τότε λέμε ότι το σύστημα είναι οριακά ευσταθές. Τέλος, εάν η ελεγχόμενη μεταβλητή αυξάνει συνεχώς με τη πάροδο του χρόνου, τότε λέμε ότι το σύστημα είναι ασταθές.



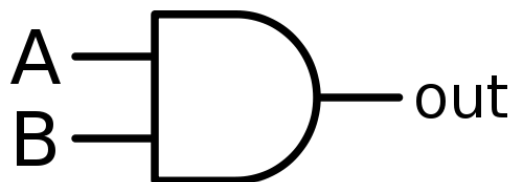
Εικόνα 4: Κυματομορφές της ελεγχόμενης μεταβλητής ενός α) ευσταθούς, β) ασταθούς, γ) οριακά ευσταθούς κλειστού συστήματος (<http://edume.myds.me>).

## 2.4 Εισαγωγή στις λογικές πύλες

Μια λογική πύλη είναι ένα στοιχειώδες δομικό στοιχείο ενός ψηφιακού κυκλώματος. Οι περισσότερες λογικές πύλες έχουν δύο εισόδους και μία έξοδο. Σε κάθε δεδομένη στιγμή, κάθε τερματικό βρίσκεται σε μία από τις δύο δυαδικές συνθήκες, χαμηλή (0) ή υψηλή (1), που αντιπροσωπεύονται από διαφορετικά επίπεδα τάσης. Η λογική κατάσταση ενός τερματικού μπορεί, γενικά, να αλλάζει συχνά καθώς το κύκλωμα επεξεργάζεται τα δεδομένα. Στις περισσότερες λογικές πύλες, η χαμηλή κατάσταση είναι περίπου 0V, ενώ η υψηλή κατάσταση είναι περίπου 5V. Χρησιμοποιώντας συνδυασμούς λογικών πυλών, μπορούμε να εκτελέσουμε σύνθετες λειτουργίες. Θεωρητικά, δεν υπάρχει όριο στον αριθμό των πυλών που μπορούν να τοποθετηθούν μαζί, σε μία μόνο συσκευή. Ωστόσο στην πράξη υπάρχει ένα όριο στον αριθμό των πυλών που μπορούν να συσκευαστούν σε ένα συγκεκριμένο φυσικό χώρο. Υπάρχουν επτά βασικές λογικές πύλες οι οποίες αναφέρονται παρακάτω μαζί με τους πίνακες αλήθειας και το κυκλωματικό σχεδιάγραμμα της κάθε πύλης.

- Πύλη «AND»: Η πύλη AND εκτελεί τη λογική πράξη «και» μεταξύ των εισόδων της. Η πύλη έχει δύο εισόδους και μια έξοδο. Η έξοδος είναι «αληθής» όταν οι εισοδοί είναι «αληθείς». Διαφορετικά, η έξοδος είναι «ψευδής». Παρακάτω βλέπουμε τον πίνακα αλήθειας της λογικής πύλης «AND». Χρησιμοποιώντας τις πράξεις ψευδής (*False* = 0) ή αληθής (*True* = 1) ελέγχουμε το αποτέλεσμα της λογικής πράξης.

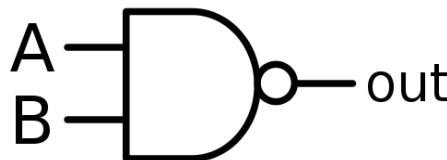
ΕΙΣΟΔΟΙ		ΕΞΟΔΟΣ
A	B	Out (A και B)
0	0	0
0	1	0
1	0	0
1	1	1



Εικόνα 5: Κυκλωματικό σχεδιάγραμμα λογικής πύλης «AND» (<https://el.wikipedia.org>).

- Πύλη «*NAND*»: Η πύλη *NAND* δίνει την αντίθετη έξοδο από μια πύλη *AND*. Λειτουργεί δηλαδή ως πύλη *AND* ακολουθούμενη από μια πύλη *NOT*. Η έξοδος είναι «ψευδής» αν και οι δύο είσοδοι είναι «αληθείς». Διαφορετικά, η έξοδος είναι «αληθής». Παρακάτω βλέπουμε τον πίνακα αλήθειας της λογικής πύλης «*NAND*». Χρησιμοποιώντας τις πράξεις ψευδής (*False* = 0) ή αληθής (*True* = 1) ελέγχουμε το αποτέλεσμα της λογικής πράξης.

ΕΙΣΟΔΟΙ		ΕΞΟΔΟΣ
A	B	Out (A <i>Nand</i> B)
0	0	1
0	1	1
1	0	1
1	1	0

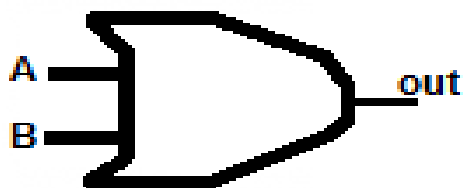


Εικόνα 6: Κυκλωματικό σχεδιάγραμμα λογικής πύλης «*NAND*» (<https://el.wikipedia.org>).

- Πύλη «*OR*»: Το όνομα της πύλης «*OR*» αιτιολογείται από το γεγονός ότι συμπεριφέρεται ακολουθώντας τη λογική της λογικής «ή». Η έξοδος είναι «αληθής» αν έστω και μια από τις δύο εισόδους της είναι «αληθής». Αν και οι δύο είσοδοι είναι «ψευδείς», τότε η έξοδος είναι «ψευδής». Παρακάτω βλέπουμε τον πίνακα αλήθειας της λογικής πύλης «*OR*». Χρησιμοποιώντας τις πράξεις ψευδής (*False* = 0) ή αληθής (*True* = 1) ελέγχουμε το αποτέλεσμα της λογικής πράξης.

ΕΙΣΟΔΟΙ		ΕΞΟΔΟΣ
A	B	Out (A ή B)
0	0	0
0	1	1
1	0	1
1	1	1

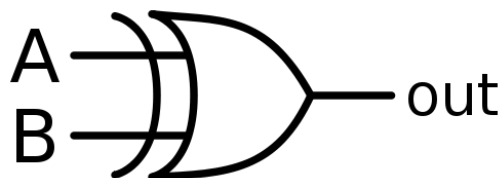




Εικόνα 7: : Κυκλωματικό σχεδιάγραμμα λογικής πύλης «OR» (<http://support.infocenter.gr>).

- Πύλη «XOR»: Η πύλη *XOR* (αποκλειστική-OR) ενεργεί με τον ίδιο τρόπο που ενεργεί η πύλη *OR*. Η έξοδος είναι «αληθής» αν έστω και μια από τις δύο εισόδους της είναι «αληθής». Αν και οι δύο εισοδοι είναι «ψευδής» ή «αληθής», τότε η έξοδος είναι «ψευδής». Παρακάτω βλέπουμε τον πίνακα αλήθειας της λογικής πύλης «XOR». Χρησιμοποιώντας τις πράξεις ψευδής (*False* = 0) ή αληθής (*True* = 1) ελέγχουμε το αποτέλεσμα της λογικής πράξης.

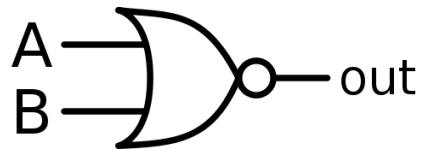
ΕΙΣΟΔΟΙ		ΕΞΟΔΟΣ
A	B	Out (A Xor B)
0	0	0
0	1	1
1	0	1
1	1	0



Εικόνα 8: Κυκλωματικό σχεδιάγραμμα λογικής πύλης «XOR» (<https://el.wikipedia.org>).

- Πύλη «NOR»: Η πύλη *NOR* είναι μια πύλη που δίνει το αντίθετο αποτέλεσμα απ' ότι μια πύλη *OR*. Δηλαδή πρόκειται για ένα συνδυασμό πύλης *OR* που ακολουθείται από έναν αντιστροφέα. Η έξοδος της είναι «αληθής» εάν και οι δύο εισοδοι είναι «ψευδείς». Διαφορετικά, η έξοδος είναι «ψευδής». Παρακάτω βλέπουμε τον πίνακα αλήθειας της λογικής πύλης «NOR». Χρησιμοποιώντας τις πράξεις ψευδής (*False* = 0) ή αληθής (*True* = 1) ελέγχουμε το αποτέλεσμα της λογικής πράξης.

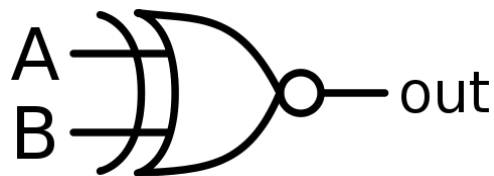
ΕΙΣΟΔΟΙ		ΕΞΟΔΟΣ
A	B	Out (A Nor B)
0	0	1
0	1	0
1	0	0
1	1	0



Εικόνα 9: Κυκλωματικό σχεδιάγραμμα λογικής πύλης «NOR» (<https://el.wikipedia.org>).

- Πύλη «XNOR»: Η πύλη XNOR (αποκλειστική-NOR) δίνει το αντίθετο αποτέλεσμα απ' ότι μια πύλη XOR. Δηλαδή πρόκειται για μια πύλη XOR ακολουθούμενη από ένα μετατροπέα. Η έξοδος της είναι «αληθής» αν οι είσοδοι είναι ίδιες και «ψευδής» όταν οι είσοδοι είναι διαφορετικές. Παρακάτω βλέπουμε τον πίνακα αλήθειας της λογικής πύλης «XNOR». Χρησιμοποιώντας τις πράξεις ψευδής (*False* = 0) ή αληθής (*True* = 1) ελέγχουμε το αποτέλεσμα της λογικής πράξης.

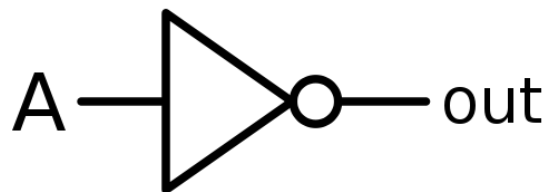
ΕΙΣΟΔΟΙ		ΕΞΟΔΟΣ
A	B	Out (A Xnor B)
0	0	1
0	1	0
1	0	0
1	1	1



Εικόνα 10: Κυκλωματικό σχεδιάγραμμα λογικής πύλης «XNOR» (<https://el.wikipedia.org>).

- Πύλη «NOT»: Ένας λογικός αντιστροφέας, που μερικές φορές ονομάζεται πύλη *NOT* για διαφοροποίηση από άλλους τύπους ηλεκτρονικών συσκευών αντιστροφέα. Έχει μόνο μία είσοδο και δίνει μόνο μία έξοδο. Η λειτουργία της είναι η αντιστροφή του λογικού σήματος της εισόδου. Παρακάτω βλέπουμε τον πίνακα αλήθειας της λογικής πύλης «NOT». Χρησιμοποιώντας τις πράξεις ψευδής (*False* = 0) ή αληθής (*True* = 1) ελέγχουμε το αποτέλεσμα της λογικής πράξης.

ΕΙΣΟΔΟΣ	ΕΞΟΔΟΣ
A	Out ( Not A)
0	1
1	0



Εικόνα 11: Κυκλωματικό σχεδιάγραμμα λογικής πύλης «NOT» (<https://el.wikipedia.org>).

## Κεφάλαιο 3<sup>ο</sup>: Προγραμματιζόμενοι Λογικοί Ελεγκτές (PLC)

Ο έλεγχος των ψηφιακών συστημάτων ελέγχου στηρίζεται σε μεγάλο βαθμό στους προγραμματιζόμενους λογικούς ελεγκτές (*Programmable Logic Controller – PLC*). Τα *PLC* είναι μονάδες που έχουν σαν βάση τους τη μαθηματική λογική. Οι σύγχρονες συσκευές δέχονται κατάλληλα σήματα εισόδου, τα οποία επεξεργάζονται μέσω ενός προγράμματος (ορισμένο από το χρήστη) ενώ δίνουν στην έξοδό τους το αποτέλεσμα του προγράμματος.

### 3.1 Πλεονεκτήματα χρήσης ΠΛΕ

Τα πλεονεκτήματα που προσφέρει ο προγραμματισμός και η χρήση *PLC* γενικότερα είναι τα κάτωθι:

- Το κόστος κατασκευής ενός *PLC* είναι μικρότερο από το κόστος παραγωγής ενός μεγάλου αριθμού βοηθητικών ηλεκτρονόμων, χρονικών και απαριθμητών.
- Τα *PLC* ως συσκευές καταλαμβάνουν μικρότερο χώρο στο πίνακα και καταναλώνουν πολύ λιγότερη ενέργεια από άλλα πιο κλασικά συστήματα αυτοματισμού.
- Έχουν την δυνατότητα σύνδεσης με κεντρικό υπολογιστή του συστήματος ή άλλο εταιρικό δίκτυο.
- Έχουν μεγάλη ευελιξία στην τροποποίηση του συστήματος αυτοματισμού. Καθώς δεν είναι κατασκευασμένα για ένα συγκεκριμένο είδος συστήματος. Συγκεκριμένα, η λογική της λειτουργίας πού ενσωματώνεται στο *PLC* μέσω του προγραμματισμού του μπορεί να τροποποιηθεί εύκολα.
- Λόγω των *LED* που υπάρχουν σε όλες τις κάρτες εισόδου και εξόδου της συσκευής, βοηθάει στον εύκολο οπτικό εντοπισμό σφάλματος στη λειτουργία του προγράμματος, ενώ παράλληλα διευκολύνεται η παρακολούθηση της ροής εκτέλεσης του προγράμματος.
- Οι γλώσσες προγραμματισμού καλύπτουν όλο το φάσμα των επαγγελματιών που καλούνται να ασχοληθούν με την τεχνολογία αυτή. Υπάρχει γλώσσα προγραμματισμού για επαγγελματίες με γνώση στο συμβατικό αυτοματισμό (*Ladder*), γλώσσες για όσους έχουν υπόβαθρο σε υπολογιστές (*Statement List, SCL, FBD, C++*) καθώς και γλώσσες εξειδικευμένες σε διάφορες τεχνολογίες (*GRAPH 7, HIGRAPH, CSF*).

### 3.2 Μειονεκτήματα χρήσης ΠΛΕ

Τα μειονεκτήματα του προγραμματισμού και της χρήσης ενός *PLC* είναι:

- Εάν παρουσιαστεί μία βλάβη στον προγραμματιζόμενο ελεγκτή, είναι δυνατόν να μην επισκευάζεται και να χρειάζεται αντικατάσταση τμήματος ή και ολόκληρου του ελεγκτή, ενώ αν είχαμε ένα πεδίο με ηλεκτρονόμους θα ήταν αρκετό να αντικαταστήσουμε ένα μόνο ηλεκτρονόμους. Αξίζει να σημειωθεί ότι ο λόγος ανάπτυξης αυτής της τεχνολογίας ήταν η αντικατάσταση παλιότερων συστημάτων με *relay*.
- Οι προγραμματιζόμενοι λογικοί ελεγκτές είναι ευαίσθητοι στον ηλεκτρονικό θόρυβο, γεγονός που απαιτεί ειδικές κατασκευές και προστασίες.
- Η εγκατάσταση, η παρακολούθηση της λειτουργίας και η συντήρηση ενός προγραμματιζόμενου ελεγκτή απαιτεί εξειδικευμένο προσωπικό.

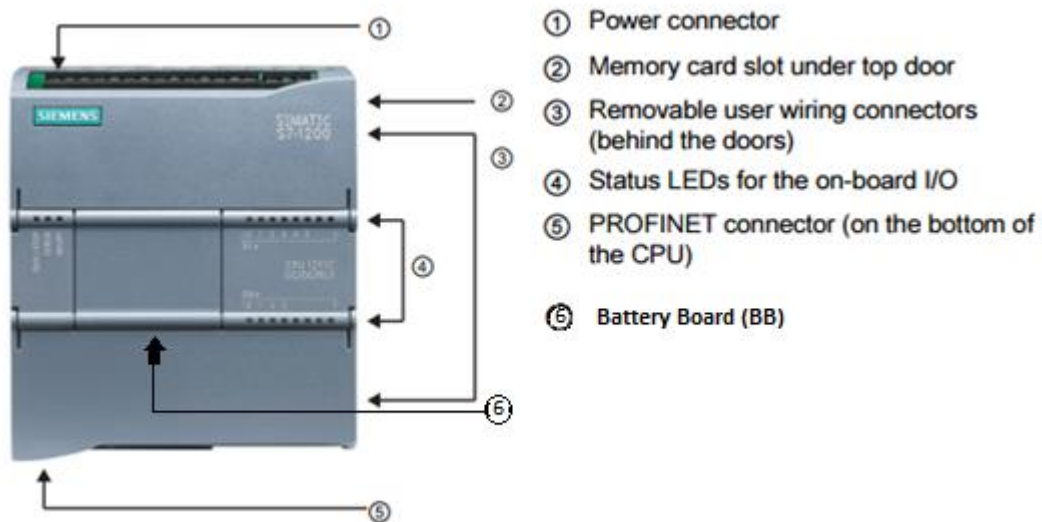
### 3.3 Η δομή ενός προγραμματιζόμενου λογικού ελεγκτή (PLC)

Το κύριο μέρος ενός *PLC* μπορούμε να το διακρίνουμε στα παρακάτω μέρη:

1. Κεντρική μονάδα επεξεργασίας (*Central Processing Unit – CPU*). Αποτελεί τη βασική μονάδα του *PLC* η οποία είναι υπεύθυνη για την λειτουργία του αυτοματισμού. Μπορούμε να την παρομοιάσουμε με έναν μικροϋπολογιστή, μιας και διακρίνουμε μικροεπεξεργαστή και μνήμη. Ο μικροεπεξεργαστής στα *PLC* είναι υπεύθυνος για όλες τις λειτουργίες του. Η μνήμη της κεντρικής μονάδας διακρίνεται σε:
  - Μνήμη τυχαίας προσπέλασης (*Random Access Memory – RAM*). Σε αυτή μπορούμε να γράφουμε και να σβήνουμε το πρόγραμμα το οποίο θα εκτελεί το *PLC*. Όταν δεν υπάρχει ηλεκτρική τροφοδοσία, δεν διατηρείται.
  - Μνήμη ανάγνωσης (*Read Only Memory – ROM*). Σε αυτή τη μνήμη ο κατασκευαστής του *PLC* έχει αποθηκεύσει το λειτουργικό σύστημα του *PLC*.
  - Μνήμη *EEPROM* είναι μία νεότερη εξέλιξη των μνήμων *ROM*. Οι μνήμες *EEPROM*, σε αντίθεση με τις μνήμες *RAM*, διατηρούν τα περιεχόμενά τους μετά τη διακοπή της τροφοδοσίας τους.

2. Τη μονάδα τροφοδοσίας (*Power Supply – SP*). Η μονάδα τροφοδοσίας έχει σκοπό να δημιουργήσει από την τάση του δικτύου τροφοδοσίας τις απαραίτητες εσωτερικές τάσεις που απαιτούνται για την τροφοδοσία των ηλεκτρικών στοιχείων του *PLC*. Οι τυπικές εσωτερικές τάσεις των *PLC* είναι συνήθως: *DC 5V*, *DC 9V*, *DC 24V*. Σε ορισμένα μοντέλα *PLC* , όταν αυτό δεν τροφοδοτείται από το δίκτυο η μονάδα τροφοδοσίας διατηρεί το περιεχόμενο της μνήμης μέσω μιας μπαταρίας (λιθίου).
3. Τις μονάδες εισόδου και εξόδου (*Input modules [I]* και *Output modules [Q]*). Πρόκειται για διατάξεις ηλεκτρονικών κυκλωμάτων για τη λήψη των εξωτερικών σημάτων ή για τη δημιουργία των κατάλληλων σημάτων εξόδου, αντίστοιχα. Συνήθως οι ακροδέκτες εισόδου συνδέονται με αισθητήρες, διακόπτες, κουμπιά, κ.ά, ενώ οι ακροδέκτες εξόδου συνδέονται με πηνία, ηλεκτρομαγνητικές βαλβίδες, αντλίες, λυχνίες ένδειξης, κτλ. Ο αριθμός των εισόδων και των εξόδων διαφέρει ανάλογα το μοντέλο της συσκευής και την εταιρία.

Παρακάτω παρουσιάζεται φωτογραφίες συγκεκριμένες μονάδες ΠΛΕ. Η εν λόγω μονάδες χρησιμοποιούνται στα πλαίσια της παρουσίασης της διπλωματικής.

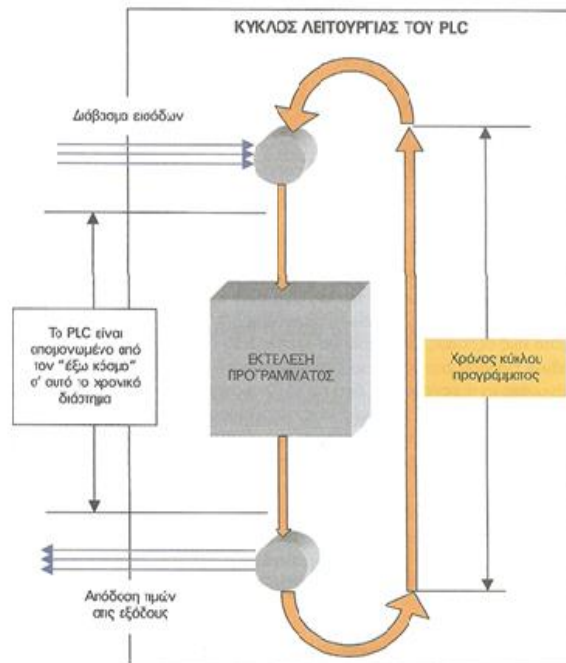


Εικόνα 12: PLC SIEMENS S7-1200 (<https://support.industry.siemens.com>).



αποθηκεύονται σε ειδική περιοχή της μνήμης που ονομάζεται εικόνα εξόδων (*output image*). Όπως η εικόνα εισόδων, έτσι και η εικόνα εξόδων περιέχει την τιμή (0 ή 1) για κάθε έξοδο.

Βήμα 3<sup>ο</sup>: Ο μικροεπεξεργαστής αποδίδει τις τιμές της εικόνας εξόδων στις φυσικές εξόδους του συστήματος. Αυτό σημαίνει ότι θα δοθεί υψηλή τάση σε όποια έξοδο όταν έχει λογικό «1» και χαμηλή τάση σε όποια έξοδο έχει λογικό «0». Με την ολοκλήρωση και του 3<sup>ου</sup> βήματος συμπληρώνεται ένας πλήρης κύκλος λειτουργίας και η διαδικασία. Ο επαναληπτικός κύκλος λειτουργίας εκτελείται συνεχώς όσο το PLC (Κατάσταση *RUN* - Μπορούμε να το αντιληφθούμε από τα *Led* πάνω στη συσκευή) είναι ενεργοποιημένο.



Εικόνα 14: Κύκλος λειτουργίας του PLC (<https://oedhlectrologoi.blogspot.gr/2015/03/plc.html>).



### 3.5 Γλώσσα προγραμματισμού ενός Προγραμματιζόμενου Λογικού Ελεγκτή (PLC)

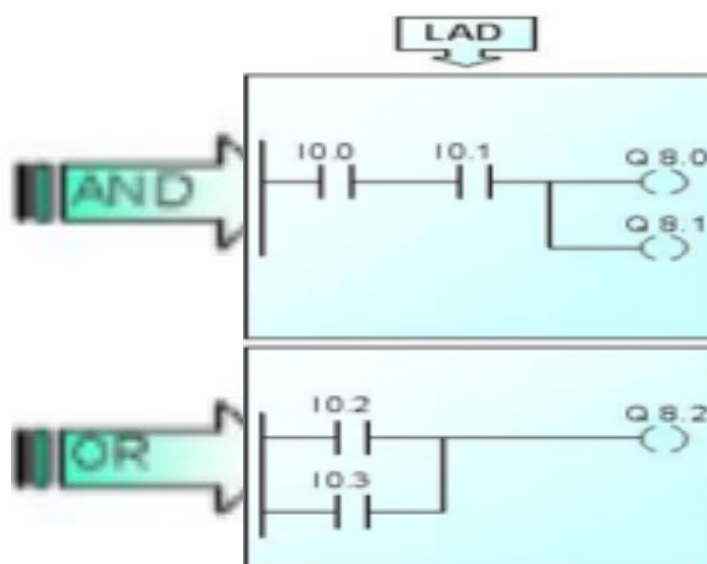
Στην εν λόγω προηγούμενη ενότητα αναπτύξαμε τη δομή λειτουργίας ενός Προγραμματιζόμενου Λογικού Ελεγκτή. Το σημαντικότερο μέρος, σε αυτά τα συστήματα, είναι το πρόγραμμα το οποίο θα υλοποιήσει το επιθυμητό αποτέλεσμα. Λόγω της μεγάλης ζήτησης και χρήσης των μονάδων, αναπτύχθηκε έντονος ανταγωνισμός μεταξύ αντίστοιχων κατασκευαστικών εταιριών, η κάθε μια «δημιούργησε» το δικό της τύπου *PLC*. Μερικές εταιρίες κατασκευής *PLC* είναι:

- *Rockwell Automation* (Αμερικάνικη βιομηχανική εταιρεία προϊόντων αυτοματισμού και πληροφορικής).
- *ABB* (Σουηδική - Ελβετική εταιρία που παρέχει τεχνολογίες ενέργειας και αυτοματισμού).
- *Omron* (Ιαπωνική εταιρία ηλεκτρολογικών προϊόντων).
- *Siemens* (Γερμανική εταιρεία κατασκευής ηλεκτρικών, ηλεκτρονικών και μηχανολογικών προϊόντων και συσκευών).
- *Yokogawa Electric Corporation* (Ιαπωνική εταιρία με επιχειρήσεις που βασίζονται στις τεχνολογίες μέτρησης, ελέγχου και πληροφορικής).

Υπάρχουν τρεις τύπου γλώσσας προγραμματισμού για *PLC* που ισχύουν ανεξάρτητα από την εταιρία. Αυτές είναι οι παρακάτω.

### 3.5.1 Γλώσσα προγραμματισμού Ladder Logic (Γλώσσα ηλεκτρολογικών γραφικών)

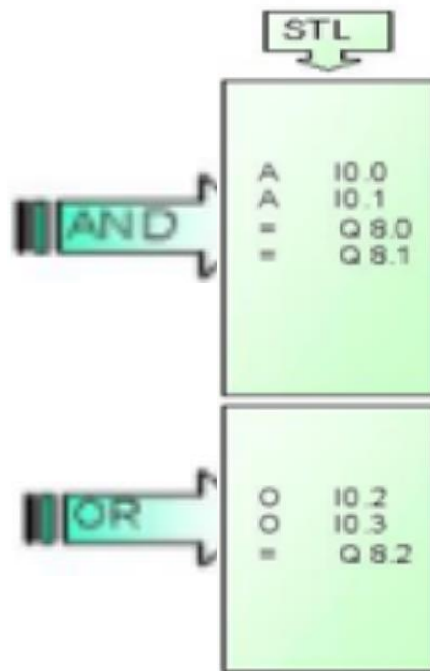
Η *Ladder* είναι η πρώτη γλώσσα προγραμματισμού *PLC* η οποία αναπτύχθηκε. Πρόκειται για μια γλώσσα γραφικών, η οποία χρησιμοποιεί ηλεκτρομηχανικά σύμβολα και επιτρέπει τη μεταφορά του ηλεκτρολογικού σχεδίου στο *PLC*. Ονομάστηκε *Ladder* (σκάλα) διότι οι γραμμές ενός συμπληρωμένου διαγράμματος μοιάζουν με τις βαθμίδες μιας σκάλας. Αυτή η γλώσσα προγραμματισμού χρησιμοποιεί την αμερικάνικη τυποποίηση στο σχεδιασμό των ηλεκτρικών επαφών. Αυτό οφείλεται στο γεγονός ενώ αρχικά τα *PLC* αναπτύχθηκαν στην Αμερική, στη συνέχεια που η χρήση τους έγινε παγκόσμια δεν άλλαξε ο σχεδιασμός των ηλεκτρικών επαφών μιας και θεωρήθηκε εύκολο στη χρήση και κατανοητό.



Εικόνα 15: Λειτουργικές λογικές δραστηριότητες σε γλώσσα Ladder ([www.PLC\\_Intro.gr](http://www.PLC_Intro.gr)).

### 3.5.2 Γλώσσα προγραμματισμού Statement List (Γλώσσα λογικών εντολών)

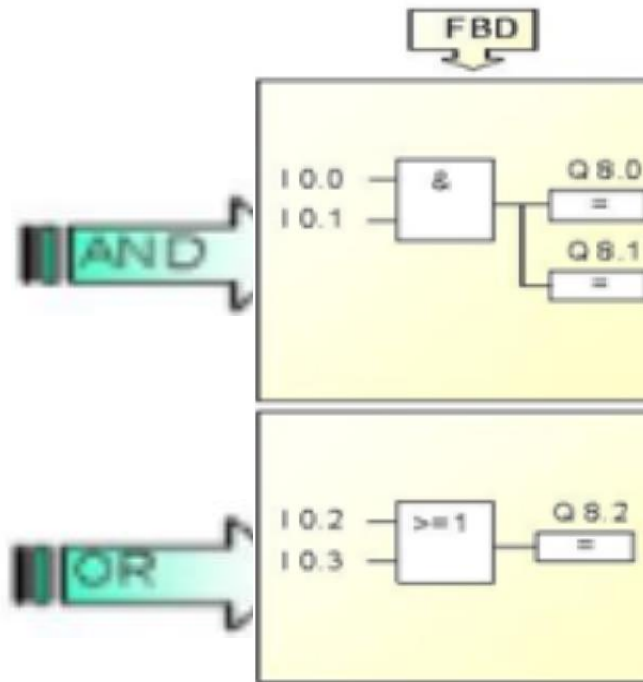
Η *Statement List (STL)* αναπτύχθηκε σχεδόν ταυτόχρονα με τη *Ladder*. Πρόκειται για μια γλώσσα λογικών εντολών, η οποία είναι παραπλήσια με αυτή του κώδικα μηχανής (*Machine Code*), όπου οι εντολές και οι λειτουργίες ακολουθούνται από διευθύνσεις. Πιο συγκεκριμένα δημιουργεί λίστα προγράμματος με εντολές οι οποίες αντιστοιχούν στις λογικές πύλες (βλ. κεφάλαιο: Εισαγωγή στις λογικές πύλες). Σε σχέση με τη *Ladder*, η *STL* απαιτεί γνώσεις προγραμματισμού.



Εικόνα 16: Λειτουργικές λογικές δραστηριότητες σε γλώσσα Statement List ([www.PLC\\_Intro.gr](http://www.PLC_Intro.gr)).

### 3.5.3 Γλώσσα προγραμματισμού Functional Block Diagram (Γλώσσα λογικών γραφικών ή λογικού διαγράμματος)

Η *Functional Block Diagram (FBD)* είναι κι αυτή, σαν τη *Ladder*, μια γλώσσα προγραμματισμού με γραφικά, αλλά αντί για το ηλεκτρολογικό σχέδιο του αυτοματισμού χρησιμοποιεί το αντίστοιχο λειτουργικό δομικό διάγραμμα του κυκλώματος. Το λειτουργικό δομικό διάγραμμα μπορεί να απεικονίσει λειτουργίες ενός συστήματος που αναπαριστώνται με λογικά *blocks* (συναντώνται στην άλγεβρα *Boole*). Παρ' όλο που τα λειτουργικά διαγράμματα (*block*) έχουν χρησιμοποιηθεί σε εφαρμογές ευρείας κλίμακας, από τη μηχανική συστημάτων έως τη μηχανική λογισμικού, από τα τέλη της δεκαετίας του 1950, η *FBD* δεν χρησιμοποιείται τόσο όσο οι άλλες δύο γλώσσες στον προγραμματισμό των *PLC*.



Εικόνα 17: Λειτουργικές λογικές δραστηριότητες σε γλώσσα Function Block Diagram ([www.PLC\\_Intro.gr](http://www.PLC_Intro.gr)).

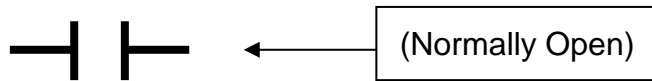
### 3.5.4 Γραφικά στοιχεία της γλώσσας προγραμματισμού Ladder

Στη *Ladder* χρησιμοποιούμε διάφορα γραφικά στοιχεία για τη δημιουργία εντολών. Αυτά χωρίζονται σε 4 κατηγορίες:

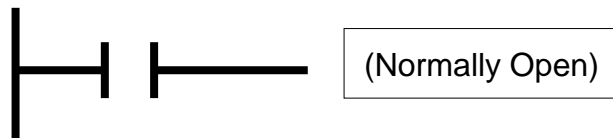
1. *Contacts* (Επαφές).
2. *Coils* (Πηνία).
3. *Timers* (Χρονικά στοιχεία).
4. *Counters* (Μετρητές).

Πιο αναλυτικά αναφέρονται παρακάτω κάποια σημαντικά γραφικά στοιχεία τα οποία ανήκουν στις προαναφερόμενες κατηγορίες, καθώς και ο συμβολισμός τους στη γλώσσα *Ladder*.

- Ανοικτή επαφή (*Normally Open – NO*): Πρόκειται για μια επαφή, στην οποία σε κατάσταση ηρεμίας, στην κανονική της δηλαδή κατάσταση, δεν παρατηρείται ροή από ρεύμα. Η ενεργοποίησή της επαφής (συνήθως είναι κάποιος διακόπτης ή κουμπί) θα κλείσει την επαφή, προκαλώντας την παροχή ροής ρεύματος στο σύστημα. Στη γλώσσα *Ladder* συμβολίζεται ως:

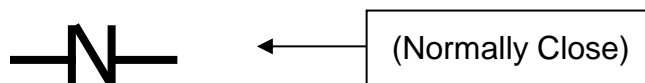


Συμβολισμός στο πλέγμα προγραμματισμού:

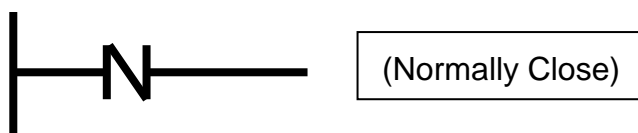


Χρησιμοποιείται για να αντιπροσωπεύσει τις πραγματικές εισόδους του *PLC*.

- Κλειστή επαφή (*Normally Close – NC*): Πρόκειται για μια επαφή, στην οποία σε κατάσταση ηρεμίας, στην κανονική δηλαδή της κατάσταση, παρατηρείται ροή από ρεύμα. Η ενεργοποίησή της επαφής (συνήθως είναι κάποιος διακόπτης ή κουμπί) θα ανοίξει την επαφή, προκαλώντας την αποφυγή ροής ρεύματος στο σύστημα. Στη γλώσσα *Ladder* συμβολίζεται ως:



Συμβολισμός στο πλέγμα προγραμματισμού:

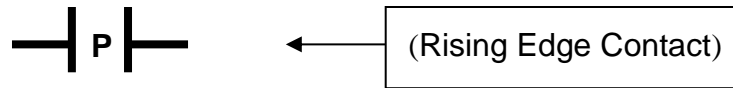


Χρησιμοποιείται για να αντιπροσωπεύσει τις πραγματικές εισόδους του *PLC*.

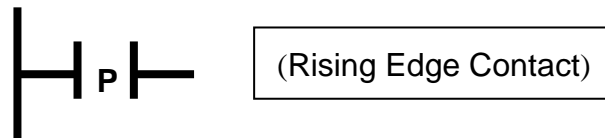
- Επαφή ανερχόμενου παλμού (*Rising Edge Contact*): Στα ηλεκτρονικά συστήματα, ένα άκρο σήματος είναι μία μετάβαση σε ψηφιακό σήμα, είτε από χαμηλό σε υψηλό (0 έως 1), είτε από υψηλό σε χαμηλό (1 έως 0). Με τον όρο «0» εννοούμε την κατάσταση ηρεμίας της επαφής, ενώ με τον όρο «1» εννοούμε την κατάσταση ενεργοποίησης της επαφής. Ονομάζεται «άκρο» (*Edge*) επειδή το τετράγωνο κύμα που αντιπροσωπεύει ένα σήμα έχει άκρα σε αυτά τα σημεία.

Η επαφή ανερχόμενου παλμού (*Rising Edge Contact*) είναι η μετάβαση από χαμηλή σε υψηλή μετάβαση, δηλαδή από κατάσταση ηρεμίας (λογικό 0) και σε κατάσταση ενεργοποίησης

(λογικό 1). Συμβολίζεται με ένα *P* (*Positive*). Όταν βρίσκεται σε κατάσταση ηρεμίας η επαφή είναι κλειστή, ενώ σε κατάσταση ενεργοποίησης, δηλαδή όταν απενεργοποιείται μια είσοδος του *PLC*, η επαφή ανοίγει. Μένει σε κατάσταση ενεργοποίησης όσο διαρκεί ένας κύκλος λειτουργίας του *PLC*. Στη γλώσσα *Ladder* συμβολίζεται ως:

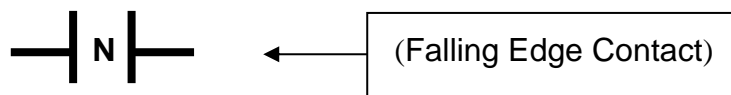


Συμβολισμός στο πλέγμα προγραμματισμού:

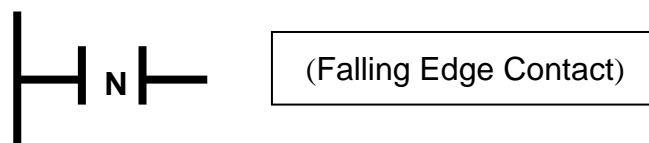


Χρησιμοποιείται για να αντιπροσωπεύσει τις πραγματικές εισόδους του *PLC*.

- Επαφή κατερχόμενου παλμού (*Falling Edge Contact*): Η επαφή κατερχόμενου παλμού (*Falling Edge Contact*) είναι η μετάβαση από υψηλή σε χαμηλή μετάβαση, δηλαδή από κατάσταση ενεργοποίησης (λογικό 1) και σε κατάσταση ηρεμίας (λογικό 0). Συμβολίζεται με ένα *N* (*Negative*). Όταν βρίσκεται σε κατάσταση ηρεμίας, η επαφή είναι ανοιχτή επιτρέποντας τη διέλευση ρεύματος, ενώ σε κατάσταση ενεργοποίησης η επαφή κλείνει. Μένει σε κατάσταση ενεργοποίησης όσο διαρκεί ένας κύκλος λειτουργίας του *PLC*. Στη γλώσσα *Ladder* συμβολίζεται ως:



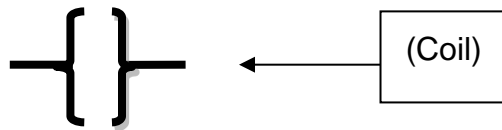
Συμβολισμός στο πλέγμα προγραμματισμού:



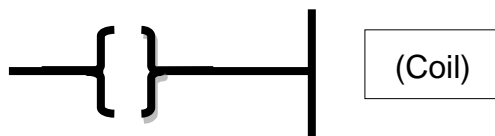
Χρησιμοποιείται για να αντιπροσωπεύσει τις πραγματικές εισόδους του *PLC*.

- Πηνίο (*Coil*): Αντιπροσωπεύει τις πραγματικές εξόδους του *PLC*. Είναι σαν ένα εσωτερικό *relay* με μια φυσική σύνδεση που μπορεί να παρέχει ηλεκτρική ισχύ για τον

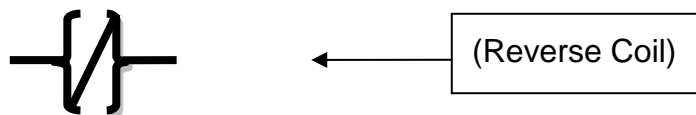
έλεγχο ενός εξωτερικού φορτίου. Ενεργοποιείται όταν περνάει ρεύμα από το σύστημα. Στη γλώσσα *Ladder* συμβολίζεται ως:



Συμβολισμός στο πλέγμα προγραμματισμού:



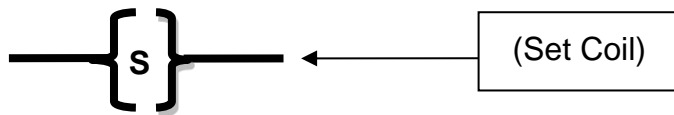
- Αντίστροφο πηνίο (*Reverse Coil*): Αντιπροσωπεύει τις πραγματικές εξόδους του *PLC*. Είναι σαν ένα εσωτερικό *relay* με μια φυσική σύνδεση που μπορεί να παρέχει ηλεκτρική ισχύ για τον έλεγχο ενός εξωτερικού φορτίου. Ενεργοποιείται όταν δεν περνάει ρεύμα από το σύστημα. Στη γλώσσα *Ladder* συμβολίζεται ως:



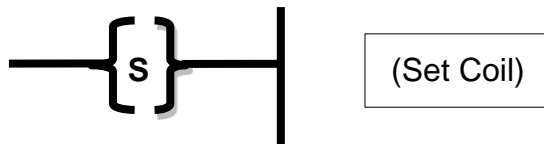
Συμβολισμός στο πλέγμα προγραμματισμού:



- Πηνίο Εξόδου αυτομανδάλωσης (*Set Coil*): Όταν δεν έχει ενεργοποιηθεί το σύστημα, το πηνίο δεν είναι οπλισμένο. Το πηνίο είναι οπλισμένο όταν ενεργοποιηθεί το σύστημα. Μόλις πάρει τάση στα άκρα του ενεργοποιείται και παραμένει ενεργοποιημένο, ανεξάρτητα από το εάν τροφοδοτείται από τάση. Για να το απενεργοποιήσουμε και να το επαναφέρουμε στην κατάσταση ηρεμίας μπορούμε να κάνουμε χρήση του πηνίου *Reset*. Στη γλώσσα *Ladder* συμβολίζεται ως:

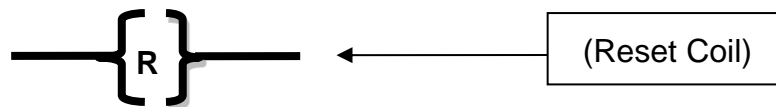


Συμβολισμός στο πλέγμα προγραμματισμού:

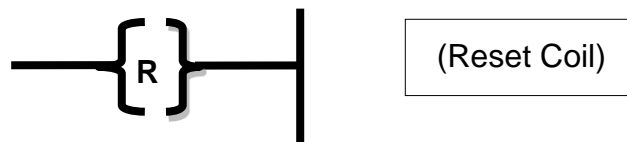


Χρησιμοποιείται για να αντιπροσωπεύσει τις πραγματικές εξόδους του PLC.

- Πηνίο απομανδάλωσης (*Reset Coil*): Εάν η ροή ισχύος στο πηνίο *Reset Coil* είναι «λογικό 1», δηλαδή να είναι οπλισμένο το πηνίο, η καθορισμένη «διεύθυνση» του στοιχείου επαναφέρεται στο «λογικό 0», δηλαδή στην κατάσταση ηρεμίας του. Εάν η ροή ισχύος στο πηνίο *Reset Coil* είναι «λογικό 0», δηλαδή να μην είναι οπλισμένο το πηνίο, η καθορισμένη «διεύθυνση» του στοιχείου δεν επηρεάζεται. Η «διεύθυνση» μπορεί επίσης να είναι ένας χρονομετρητής του οποίου η τιμή χρονοδιακόπτη επαναφέρεται σε «λογικό 0». Στη γλώσσα *Ladder* συμβολίζεται ως:



Συμβολισμός στο πλέγμα προγραμματισμού:

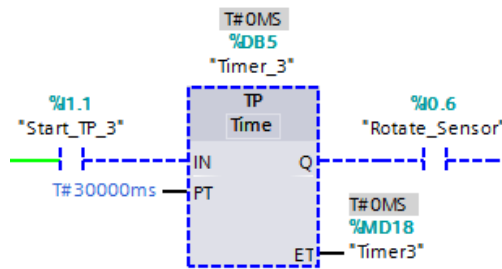


- Χρονικά στοιχεία (*Timers*): Τα χρονικά χρησιμοποιούνται για ένα πλήθος εφαρμογών, όπως η δημιουργία παλμών συγκεκριμένου εύρους και συχνότητας, η καταγραφή του χρόνου φαινομένων και σημάτων, δίνοντας τη δυνατότητα στη ροή του προγράμματος να χειρίζεται τον πραγματικό χρόνο με ακρίβεια. Υπάρχουν διάφορα είδη χρονικών που



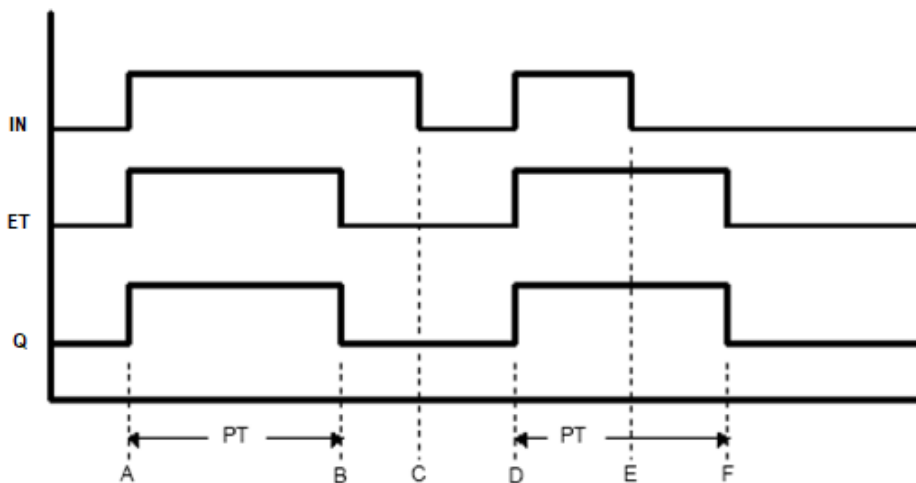
χρησιμοποιούνται ανάλογα με τη φύση της εφαρμογής (χρονικά καθυστέρησης, χρονικά παλμού, κλπ):

- *Pulse Timer (TP)*: Στον εν λόγω μετρητή με ενεργοποίηση της εισόδου (*IN*), η έξοδος (*Q*) παραμένει ενεργοποιημένη για χρήση διαστήματος (*PT*). Η έξοδος *ET* δείχνει ανά πάσα στιγμή τη μέτρηση.



Εικόνα 18: Pulse Timer (TP).

Για τη καλύτερη κατανόηση της λειτουργίας ενός *controller TP* ακολουθεί το παρακάτω διάγραμμα.



Εικόνα 19: Λειτουργία ενός TP (<https://www.hmisource.com>).

Σημείο *A*: Ο χρονοδιακόπτης ενεργοποιείται και ξεκινά τη μέτρηση (*ET*). Η έξοδος *Q* του χρονοδιακόπτη ενεργοποιείται.

Σημείο *B*: Ο χρόνος που έχει παρέλθει (*ET*) ισούται με τον προκαθορισμένο χρόνο (*PT*). Η έξοδος (*Q*) απενεργοποιείται. Ο χρονοδιακόπτης σταματά τη μέτρηση (*ET*). Ο χρόνος που έχει παρέλθει (*ET*) παραμένει ίσος με τον προκαθορισμένο χρόνο ( $ET = PT$ ).

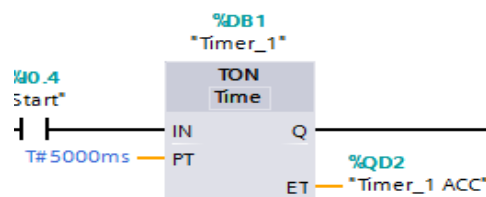
Σημείο C: Ο χρονοδιακόπτης απενεργοποιείται. Ο χρόνος που έχει περάσει (ET) επαναφέρεται στο 0 (έχει παρέλθει ο χρόνος ET).

Σημείο D: Ο χρονοδιακόπτης ενεργοποιείται. Ο χρονοδιακόπτης ξεκινά τη μέτρηση (ET). Το bit εξόδου χρονοδιακόπτη (Q) ενεργοποιείται.

Σημείο E: Ο χρονοδιακόπτης απενεργοποιείται. Ο χρονομετρητής συνεχίζει τη μέτρηση (το TI παραμένει ενεργοποιημένο). Το bit εξόδου χρονοδιακόπτη (Q) παραμένει ενεργοποιημένο.

Σημείο F: Ο χρόνος που έχει παρέλθει (ET) ισούται με τον προκαθορισμένο χρόνο (PT). Το bit εξόδου χρονοδιακόπτη (Q) απενεργοποιείται. Ο χρονοδιακόπτης σταματά τη μέτρηση (TI). Επειδή το bit εισόδου χρονοδιακόπτη (IN) είναι απενεργοποιημένο, ο χρόνος που έχει περάσει (ET) επαναφέρεται στο 0.

- *Timer On Delay (TON)*: Η εντολή «χρονοκαθυστέρηση» αρχίζει να μετράει το χρονικό διάστημα (PT). Η συσσωρευμένη τιμή (ET).

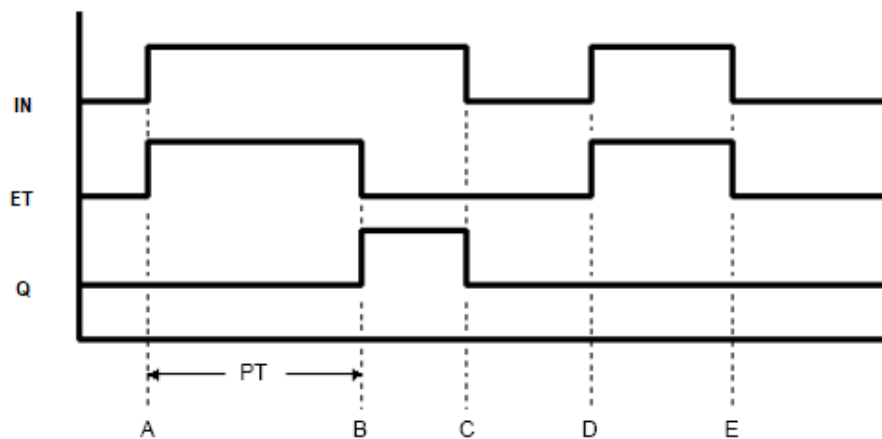


Εικόνα 20: Timer On Delay.

Πίνακας 1: Στοιχεία χρονικών μεταβλητών του Timer On Delay (TON).

Μεταβλητή	Περιγραφή
IN	Ενεργοποιεί το χρονοδιακόπτη (μεταβλητή εισόδου)
Q	Ενεργοποιείται μετά την λήξη της καθυστέρησης (μεταβλητή εξόδου)
PT	Ορίζουμε τη τιμή της χρονικής καθυστέρησης-Προσοχή στη μονάδα μέτρησης (μεταβλητή εισόδου)
ET	Η τρέχουσα τιμή μέτρησης (μεταβλητή εξόδου)

Για τη καλύτερη κατανόηση της λειτουργίας ενός *TON* ακολουθεί το παρακάτω διάγραμμα.



Εικόνα 21: Λειτουργία ενός *TON* (<https://www.hmisource.com>).

Σημείο *A*: Ο χρονοδιακόπτης τίθεται σε λειτουργία και το *bit* μέτρησης χρονοδιακόπτη ενεργοποιείται. Η μέτρηση χρονοδιακόπτη αρχίζει και ο χρόνος που έχει περάσει (*ET*) αυξάνεται. Το *bit* εξόδου χρονοδιακόπτη (*Q*) παραμένει εκτός λειτουργίας.

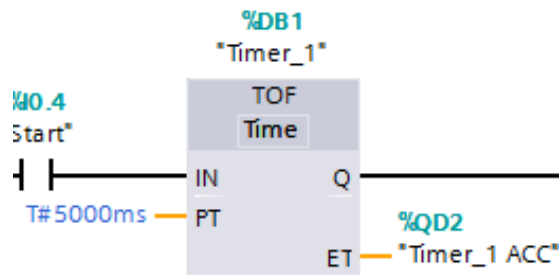
Σημείο *B*: Όταν ο χρόνος που έχει περάσει (*ET*) ισούται με τον προκαθορισμένο χρόνο (*PT*), το *bit* εξόδου χρονοδιακόπτη (*Q*) ενεργοποιείται. Όταν το *ET* παραμένει το ίδιο με τον προκαθορισμένο χρόνο (*PT*), το *bit* μέτρησης του χρονοδιακόπτη (*ET*) απενεργοποιείται.

Σημείο *C*: Το *bit* εισόδου χρονοδιακόπτη (*IN*) απενεργοποιείται και το *bit* εξόδου χρονοδιακόπτη (*Q*) απενεργοποιείται. Ο χρόνος που έχει περάσει (*ET*) επαναφέρεται στο 0.

Σημείο *D*: Ο χρονοδιακόπτης τίθεται σε λειτουργία και το *bit* μέτρησης χρονοδιακόπτη ενεργοποιείται. Η μέτρηση χρονοδιακόπτη αρχίζει και ο χρόνος που έχει περάσει (*ET*) αυξάνεται.

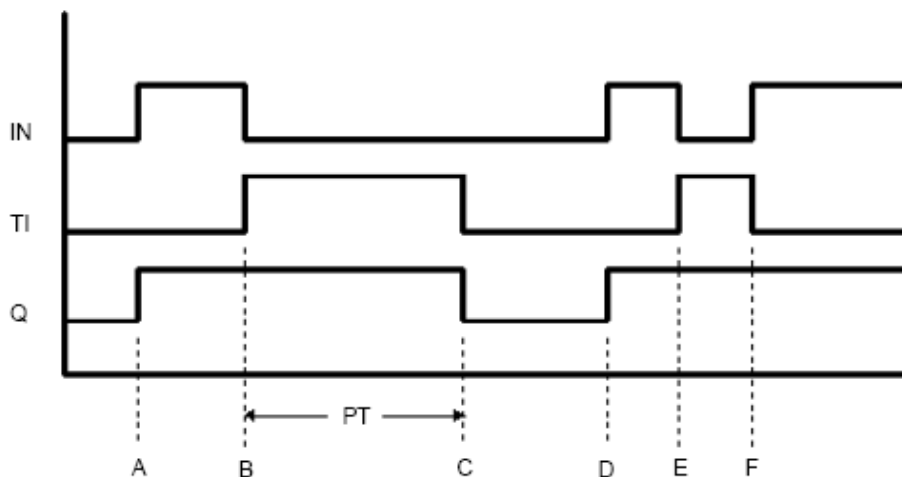
Σημείο *E*: Ο χρονοδιακόπτης απενεργοποιείται πριν από τη λήξη του χρόνου (*ET*) φτάσει στον προκαθορισμένο χρόνο (*PT*). Ενώ το *bit* εξόδου χρονοδιακόπτη (*Q*) παραμένει απενεργοποιημένο, ο χρόνος που έχει περάσει (*ET*) επανέρχεται στο 0.

- *Timer Off Delay (TOF)*: Στον εν λόγω μετρητή με απενεργοποίηση της εισόδου (*IN*), η έξοδος (*Q*) παραμένει ενεργοποιημένη για χρήση διαστήματος (*PT*). Η έξοδος *ET* δείχνει ανά πάσα στιγμή τη μέτρηση.



Εικόνα 22: Timer Off Delay (www. PLC\_Int.gr).

Ο προγραμματισμός του *Timer* είναι όμοιος με του *TON* (*Timer On Delay*) με τη διαφορά ότι σε έναν *TOF* ισχύει:



Εικόνα 23: Λειτουργία ενός TOF (<https://www.hmisource.com>).

Σημείο A: Ο χρονοδιακόπτης ανάβει. Το *bit* μέτρησης χρονομέτρου (*TI*) παραμένει απενεργοποιημένο. Το *bit* εξόδου χρονοδιακόπτη (*Q*) ενεργοποιείται. Ο χρόνος που έχει περάσει (*ET*) επαναφέρεται στο 0.

Σημείο B: Ο χρονοδιακόπτης απενεργοποιείται και ξεκινά τη μέτρηση (ο *TI* ενεργοποιείται). Το *bit* εξόδου χρονοδιακόπτη (*Q*) παραμένει ενεργοποιημένη.

Σημείο C: Ο χρόνος που έχει παρέλθει (*ET*) ισούται με τον προκαθορισμένο χρόνο (*PT*). Το *bit* εξόδου χρονοδιακόπτη (*Q*) απενεργοποιείται. Ο χρονοδιακόπτης σταματά τη μέτρηση (ο *TI* απενεργοποιείται). Ο χρόνος που έχει παρέλθει (*ET*) παραμένει ίσος με τον χρόνο ρύθμισης (*ET* = *PT*).

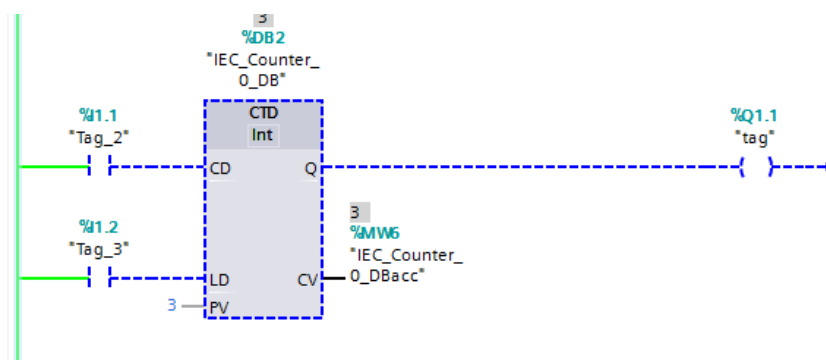
Σημείο *D*: Ο χρονοδιακόπτης ενεργοποιείται. Το *bit* μέτρησης χρονομέτρου (*TI*) παραμένει απενεργοποιημένο. Το *bit* εξόδου χρονοδιακόπτη (*Q*) παραμένει ενεργοποιημένο. Ο χρόνος που έχει περάσει (*ET*) επαναφέρεται στο 0.

Σημείο *E*: Ο χρονοδιακόπτης απενεργοποιείται. Ο χρονοδιακόπτης αρχίζει τη μέτρηση (ο *TI* ενεργοποιείται). Το *bit* εξόδου χρονοδιακόπτη (*Q*) παραμένει ενεργοποιημένο.

Σημείο *F*: Ο χρονοδιακόπτης ενεργοποιείται πριν η πάροδος του χρόνου (*ET*) φτάσει στον προκαθορισμένο χρόνο (*PT*) και ο χρονομετρητής σταματήσει τη μέτρηση (ο *TI* απενεργοποιείται). Το *bit* εξόδου χρονοδιακόπτη (*Q*) παραμένει ενεργοποιημένο και ο χρόνος που έχει περάσει (*ET*) επαναρυθμίζεται στο 0.

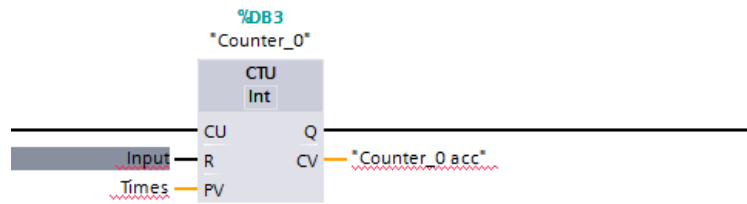
- Μετρητικά στοιχεία (*Counters*): Ένας μετρητής έχει προγραμματιστεί σε κάποια τιμή προεπιλογής και όταν έχει ληφθεί αυτή η τιμή παλμών εισόδου, θα ενεργοποιήσει κάποιες επαφές εξόδου του. Οι ανοικτές επαφές (*Normally Open*) θα είναι κλειστές και οι κλειστές επαφές (*Normally Close*) θα ανοίξουν. Υπάρχουν δύο βασικοί τύποι μετρητών: *down counters* και *up counters*.

- *Down Counters (CTD)*: Η επαφή *Q* ενεργοποιείται κάθε φορά που ενεργοποιείται η επαφή εισόδου *CD*, ενώ η συσσωρευμένη τιμή μέτρησης εμφανίζεται στην επαφή *CV*.



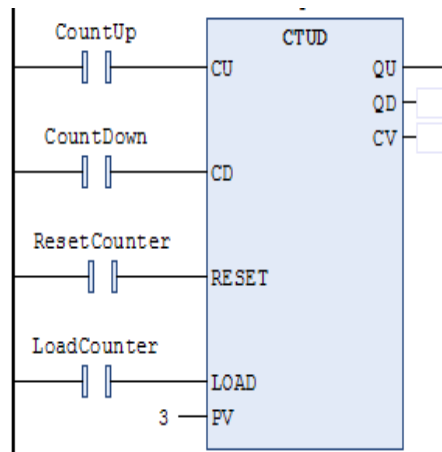
Εικόνα 24: Counter Down.

- *Up Counters (CTU)*: Η επαφή *Q* ενεργοποιείται κάθε φορά που απενεργοποιείται η επαφή εισόδου *CD*, ενώ η συσσωρευμένη τιμή μέτρησης εμφανίζεται στην επαφή *CV*.



Εικόνα 25: Counter Up.

- *Counter Up and Down (CTUD)*: Αυτός ο counter αποτελεί συνδυασμό των δύο προηγούμενων.



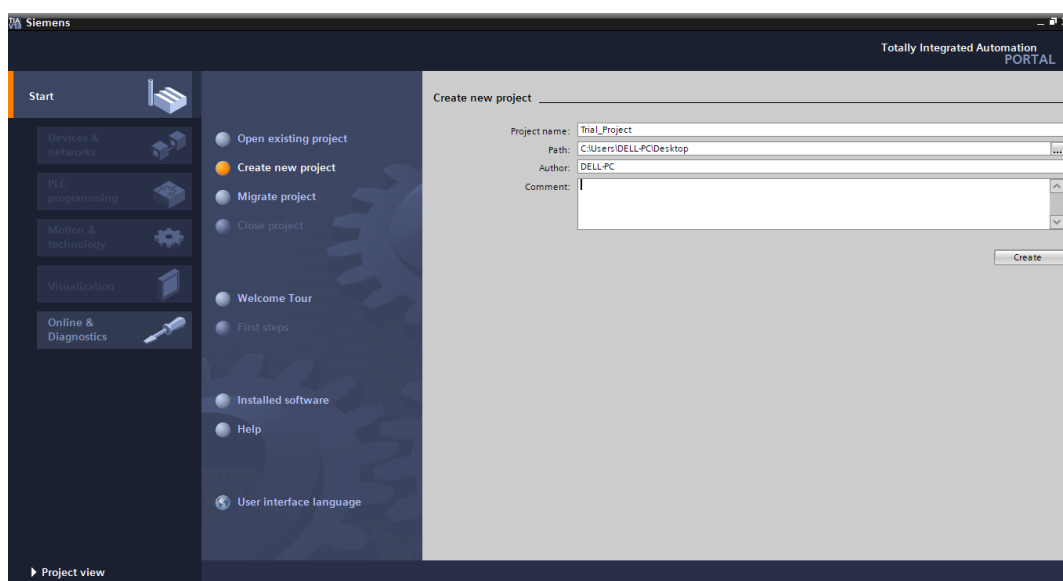
Εικόνα 26: Counter Up and Down (<https://www.youtube.com>).

## Κεφάλαιο 4<sup>ο</sup>: Δημιουργία νέου project με χρήση του λογισμικού TIA Portal V13

Στην ενότητα αυτή γίνεται παρουσίαση του περιβάλλοντος του *TIA Portal V13* και εξηγείται ο τρόπος δημιουργίας ενός νέου *project - προγράμματος*. Για να επιτευχθεί αυτό θα πρέπει να έχουμε συνδέσει το *PLC* και τον υπολογιστή μας με καλώδιο *Ethernet* και να έχουμε θέσει σε λειτουργία το *PLC*. Παρακάτω ακολουθούν τα βασικά στάδια, σύμφωνα με το εγχειρίδιο χρήσης του προγράμματος.

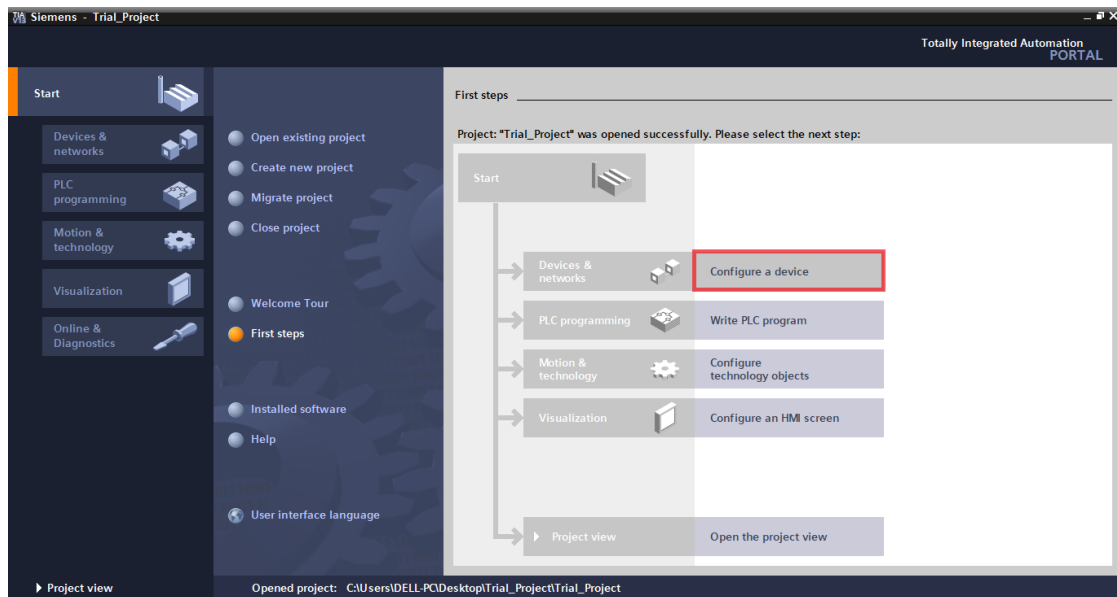
## 4.1 Απαραίτητα βήματα για τη δημιουργία νέου κώδικα/project

Ανοίγουμε το αρχείο με όνομα *TIA Portal V13*. Για τη δημιουργία ενός νέου *project*. Μαρκάρουμε την επιλογή «*Create new project*» και θα οδηγηθούμε στο παράθυρο που φαίνεται στην ακόλουθη εικόνα, όπου και ορίζουμε το όνομα (*project name*) και το σημείο που θα αποθηκευθεί το αρχείο μας με το όνομα που του δώσαμε (*path*).



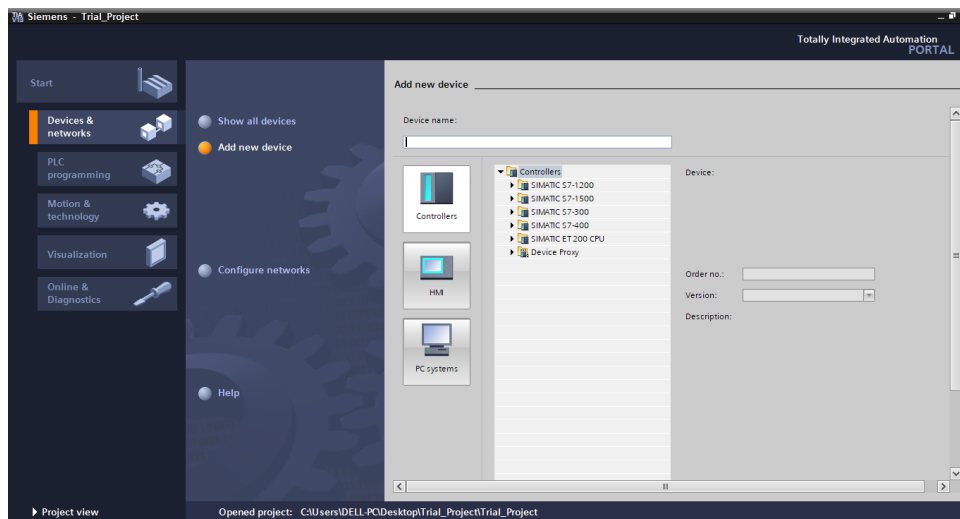
Εικόνα 27: Δημιουργία νέου project μέσω του προγράμματος TIA PORTAL V13.

Στη συνέχεια, μαρκάροντας την επιλογή «*First steps*», εμφανίζεται ένα νέο παράθυρο, όπου ο χρήστης καλείται να επιλέξει το πλαίσιο στο οποίο αναγράφεται «*Configure a device*» που ανήκει στο πεδίο «*Devices & networks*», ώστε να εμφανιστεί η λίστα με τις συσκευές που υποστηρίζει η συγκεκριμένη έκδοση του λογισμικού και να ορίσει εκείνη που θα χρησιμοποιήσει για το εκάστοτε *project*.



Εικόνα 28: Διαμόρφωση συσκευής με χρήση του προγράμματος TIA PORTAL V13.

Μετά την περάτωση του προηγούμενου βήματος, προκύπτει μια νέα επιφάνεια στην οποία ο χρήστης συναντά τρεις επιλογές, από τις οποίες καλείται να επιλέξει την «Add new device». Στη συνέχεια θα οδηγηθεί σε ένα νέο παράθυρο. Στο πάνω μέρος του διακρίνουμε ένα πεδίο με τίτλο «Device name», όπου ο χρήστης δύναται να εισάγει ένα όνομα της επιλογής του, για τη συσκευή που πρόκειται να χρησιμοποιηθεί. Στη συγκεκριμένη περίπτωση που θα χρησιμοποιήσουμε *PLC* θα επιλέξουμε το εικονίδιο με όνομα *controllers*.

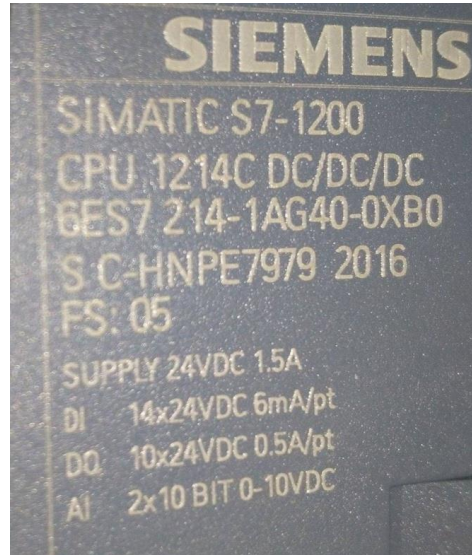


Εικόνα 29: Επιλογή συσκευής την οποία θα προγραμματίσουμε του προγράμματος.

Ακριβώς από κάτω υπάρχει μία στήλη που υποδεικνύει τις διαθέσιμες κατηγορίες συσκευών, οι οποίες διαφέρουν ανάλογα με την έκδοση του λογισμικού πακέτου (δηλαδή εάν πρόκειται για το *TIA Portal V13 basic* ή *professional* κλπ). Για να μπορέσουμε να επιλέξουμε την κατάλληλη

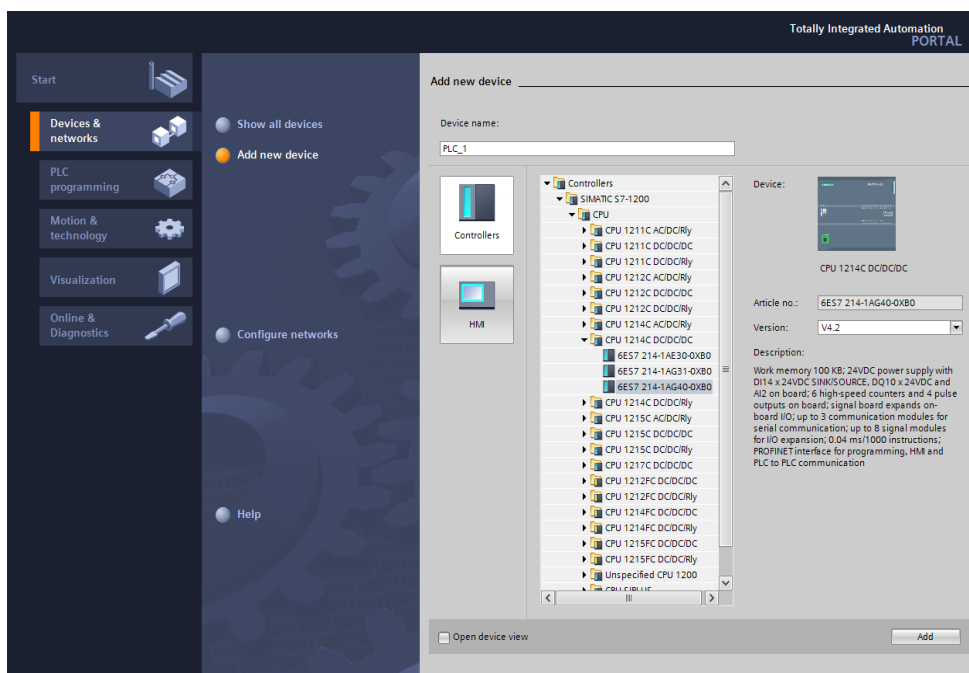


συσκευή, στη συγκεκριμένη περίπτωση για τα *PLC* που θα χρησιμοποιηθούν κατά τη διάρκεια του εργαστηρίου, κοιτάμε την μονάδα *PLC* που έχουμε. Στη μια πλευρά της μονάδας προγραμματιζόμενου λογικού ελεγκτή αναγράφονται όλα τα βασικά στοιχεία της μονάδας.



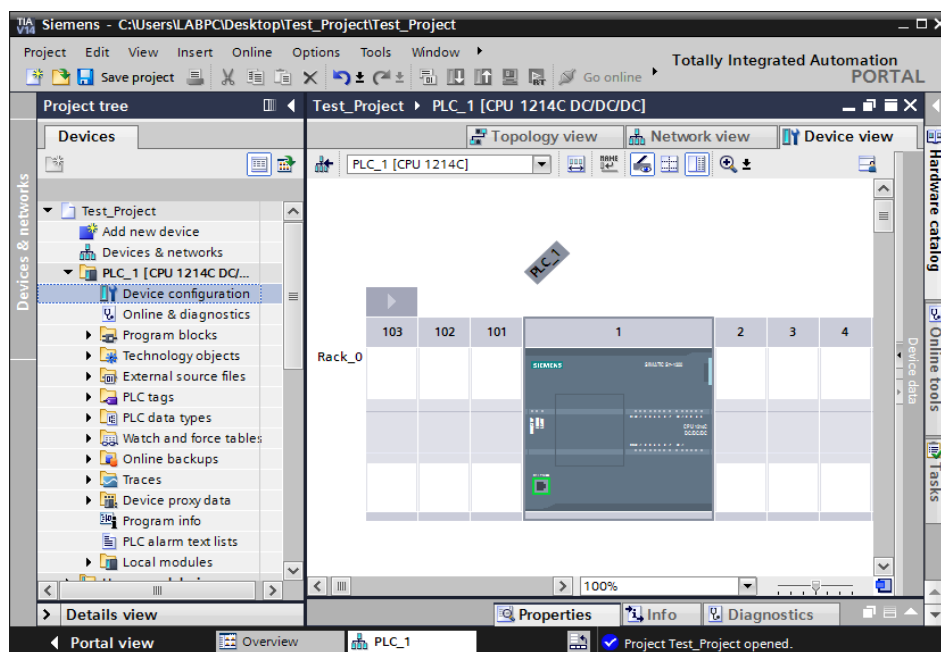
Εικόνα 30: Στοιχεία μονάδας PLC.

Έτσι, με βάση την κατηγορία που επιλέγουμε, στο δεξί μέρος της οθόνης εμφανίζεται μια δεύτερη λίστα με τις συμβατές συσκευές, από όπου και επιλέγεται η επιθυμητή. Στη παρακάτω εικόνα, διακρίνονται τα προαναφερόμενα.



Εικόνα 31: Επιλογή PLC με βάση τα στοιχεία της μονάδας.

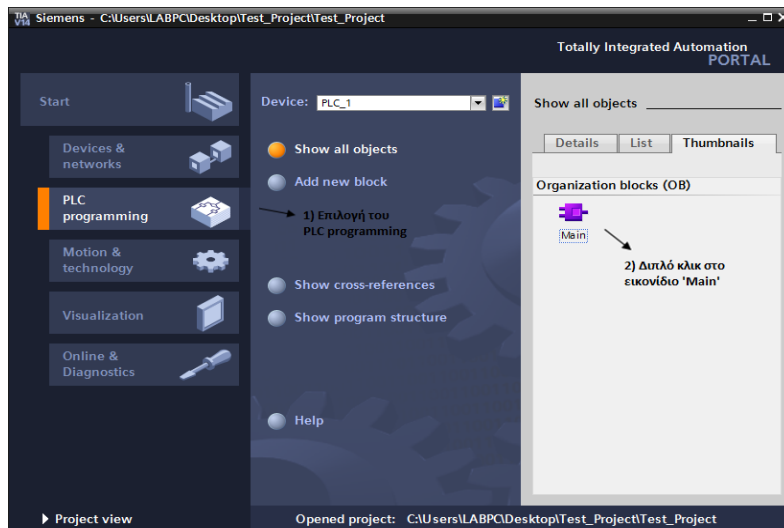
Μόλις επιλεγθεί κάποια συσκευή, στο δεξί μέρος της οθόνης παρουσιάζεται μια απεικόνιση της, καθώς και τα χαρακτηριστικά - δυνατότητές της. Επιλέγουμε την 4.2 *version*. Η κύρια διαφορά στις εκδόσεις είναι η μνήμη της συσκευής. Για την επιτυχή εισαγωγή οποιασδήποτε συσκευής στο νέο *project*, επιλέγουμε την εντολή «Add» που βρίσκεται δεξιά, στο κάτω μέρος της οθόνης. Στην περίπτωση που μαρκαριστεί το πεδίο «Open device view», που βρίσκεται στην ίδια ευθεία με την εντολή Add, γίνεται μεταφορά στο παράθυρο του «Device configuration».



Εικόνα 32: Η μονάδα PLC στο πρόγραμμα TIA PORTAL V13.

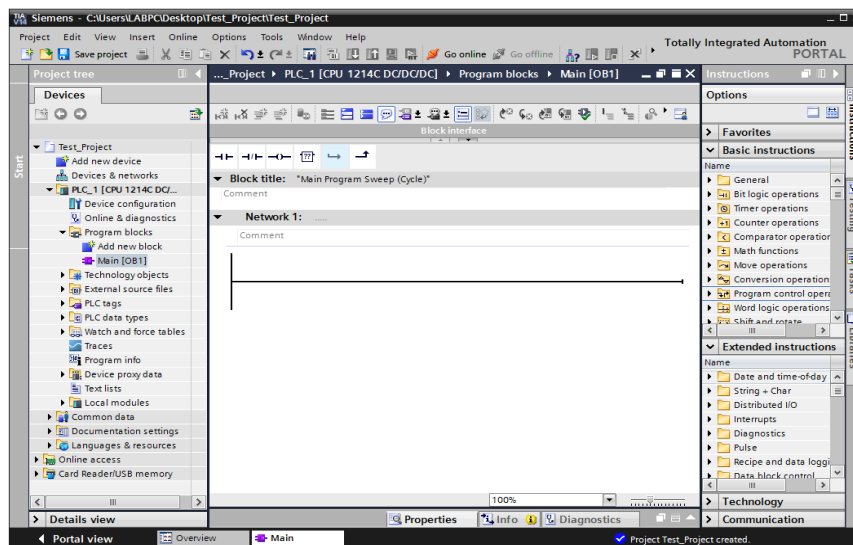
Το νέο παράθυρο στο οποίο μας μετέφερε το πρόγραμμα μας δείχνει το PLC με το οποίο συνδέσαμε τον υπολογιστή μας. Αφού έχει οριστεί η συσκευή που θα χρησιμοποιηθεί, ο χρήστης είναι σε θέση να προχωρήσει στη σύνταξη του κατάλληλου κώδικα. Για να ξεκινήσουμε τον προγραμματισμό της μονάδας γυρίζουμε στο αρχικό παράθυρο μέσω της εντολής «Portal view» που βρίσκεται κάτω αριστερά, όπως φαίνεται στην παραπάνω εικόνα.

Στη συνέχεια ο χρήστης θα επιλέξει το πλαίσιο «PLC programming» που βρίσκεται στα αριστερά της οθόνης και ύστερα με διπλό *click*, το εικονίδιο «Main» του πεδίου «Organization blocks (OB)», όπως φαίνεται και στην παρακάτω εικόνα.



Εικόνα 33: Δημιουργία προγράμματος μέσω του TIA PORTAL V13.

Με την ολοκλήρωση του προηγούμενου σταδίου, εμφανίζεται ένα νέο παράθυρο στο οποίο γίνεται η ανάπτυξη-σύνταξη του επιθυμητού κώδικα σε γλώσσα *Ladder*.



Εικόνα 34: Παράθυρο προγραμματισμού του προγράμματος TIA PORTAL V13.

Εφόσον έχει δημιουργηθεί επιτυχώς κάποιο νέο *project*, ο χρήστης είναι σε θέση να αναπτύξει τον επιθυμητό κώδικα - πρόγραμμα και εν συνεχεία να πραγματοποιήσει την εγγραφή του στη μνήμη του *PLC* που χρησιμοποιείται. Παρακάτω περιγράφεται η διαδικασία εγκαθίδρυσης επικοινωνίας *PC – PLC*.

## 4.2 Εγκαθίδρυση επικοινωνίας PC – PLC και ορισμός IP address

Απαραίτητη προϋπόθεση για την επίτευξη της εγγραφής οποιαδήποτε *project* σε ένα *PLC*, είναι η εγκαθίδρυση της επικοινωνίας μεταξύ του *PLC* και του ηλεκτρονικού υπολογιστή (*PC*) ή άλλης περιφερειακής συσκευής (*Peripheral Gateway - PG*) που χρησιμοποιείται για τον προγραμματισμό των *PLC*. Αυτό επιτυγχάνεται με τον ορισμό κατάλληλης *IP* διεύθυνσης (*IP Address*), μάσκας υποδικτύου (*Subnet Mask*) και προεπιλεγμένη πύλη (*Default Gateway*) στις συσκευές. Σε αυτό το σημείο καλό είναι να σημειωθεί πως η διεύθυνση *IP* αποτελεί ένα μοναδικό αριθμό που περιγράφει κάθε συσκευή που βρίσκεται σε ένα δίκτυο. Κάθε συσκευή χαρακτηρίζεται από τη δίκη της *IP* διεύθυνση. Η διεύθυνση αυτή ορίζεται με σκοπό την αναγνώριση και τη συνεννόηση μεταξύ των συσκευών. Προκειμένου να αποφασίσει ο χρήστης ποιά *IP* διεύθυνση θα ορίσει στο *PLC*, οφείλει να γνωρίζει την *IP* της συσκευής με την οποία το προγραμματίζει, δηλαδή την *IP* του ηλεκτρονικού του υπολογιστή. Μια *IP* διεύθυνση αποτελείται ουσιαστικά από δύο τμήματα, το *network address* και το *host address*. Όλοι οι υπολογιστές που βρίσκονται σε ένα δίκτυο μοιράζονται το ίδιο *network address*, ενώ το *host address* είναι μοναδικό για κάθε συσκευή. Η μάσκα υποδικτύου αποτελεί έναν 32-bit αριθμό (π.χ. 255.255.255.0) και υποδεικνύει τα «όρια» των δύο τμημάτων που προαναφέρθηκαν. Για τον εντοπισμό της διεύθυνσης *IP* ενός υπολογιστή «ρωτάμε» τον ίδιο τον υπολογιστή μέσω της γραμμής εντολών (*Command Prompt – cmd*) των Windows και την εντολή «*ipconfig*» που του ορίζουμε, όπως διακρίνεται και στην παρακάτω εικόνα.

```
cmd. Γραμμή εντολών
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\Eιρήνη>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix . . : zte.com.cn
    IPv6 Address. . . . . : 2a02:2149:8105:1000:947d:b58d:acf8:431c
    Temporary IPv6 Address. . . . . : 2a02:2149:8105:1000:e41d:6ef0:96f6:d22f
    Link-local IPv6 Address . . . . . : fe80::947d:b58d:acf8:431c%10
    IPv4 Address. . . . . : 192.168.1.13
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::1%10
                                192.168.1.1
```

Εικόνα 35: Πληροφορίες υπολογιστή όταν δεν είναι συνδεδεμένο σε router.

```
ca: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>IPCONFIG

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::257a:c223:7fe2:af2a%11
    IPv4 Address. . . . . : 192.168.88.254
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.88.1

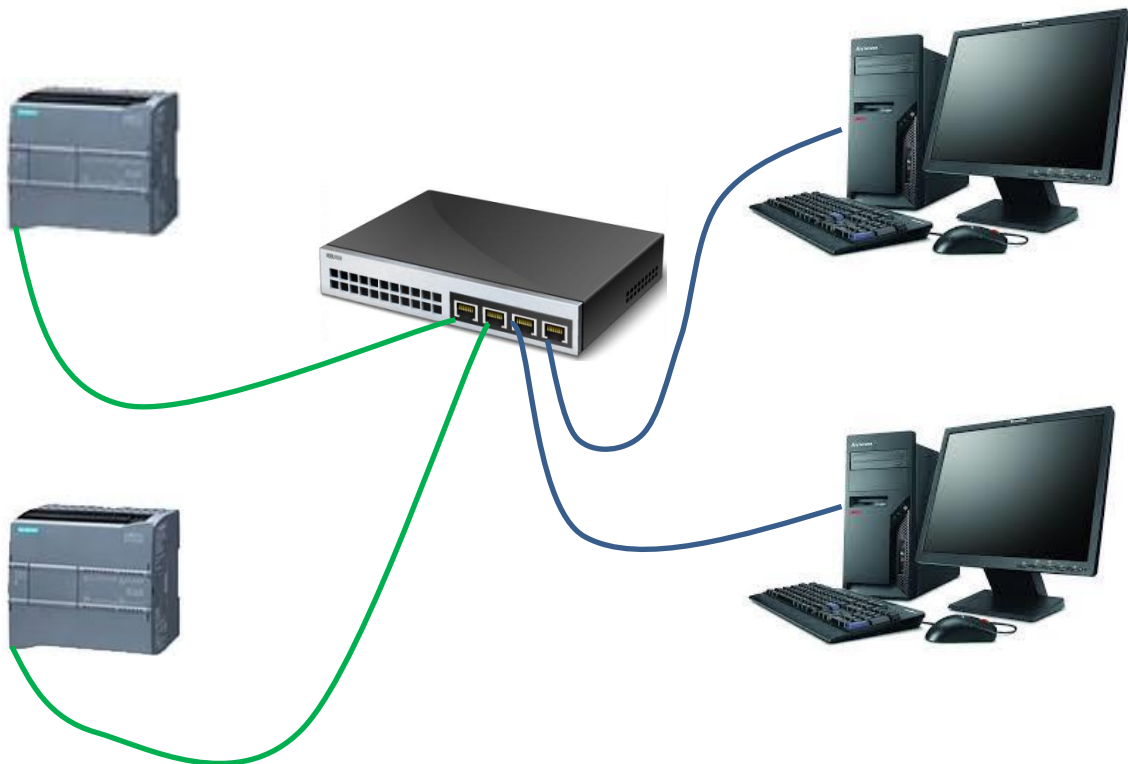
Tunnel adapter isatap.{B4E940E3-C5BC-4DDF-93A2-FADD77A1E5FB}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

C:\Users\User>
```

Εικόνα 36: Πληροφορίες υπολογιστή όταν είναι συνδεδεμένο σε router.

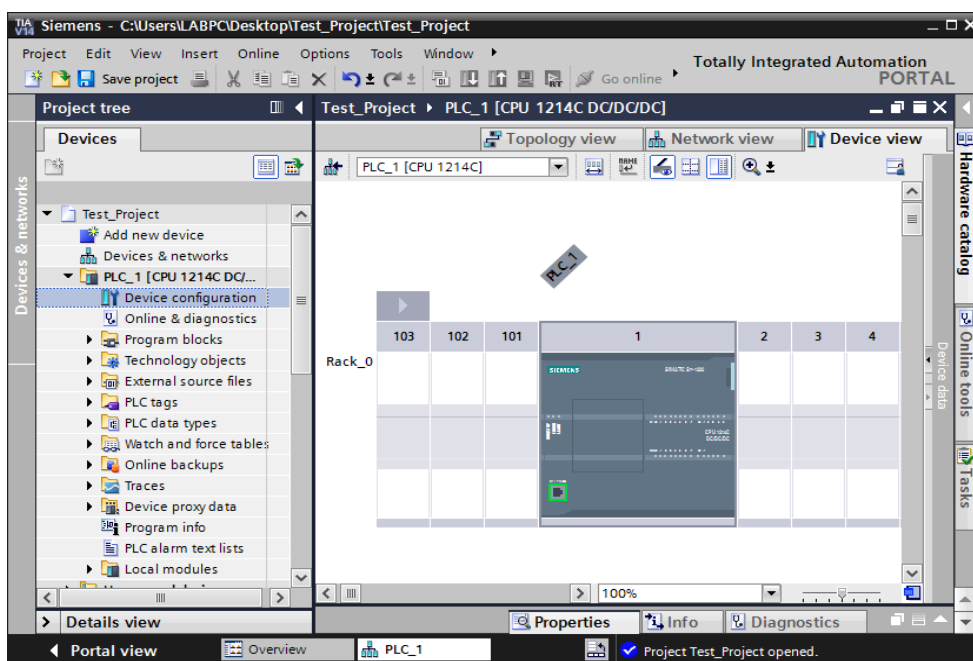
Οι εκπαιδευτικές μονάδες και οι υπολογιστές του εργαστηρίου συνδέονται μέσω *router*, όπως φαίνεται στο παρακάτω διάγραμμα. Επομένως όλες οι μονάδες θα έχουν την ίδια προεπιλεγμένη πύλη (*Default Gateway*).



Εικόνα 37: Σύνδεση PLC και υπολογιστών μέσω router.

Μόλις εντοπιστεί η *IP* του υπολογιστή, ο χρήστης είναι σε θέση να προχωρήσει στην εκχώρηση διεύθυνσης *IP* στο *PLC* που πρόκειται να χρησιμοποιηθεί, ακολουθώντας τα παρακάτω βήματα:

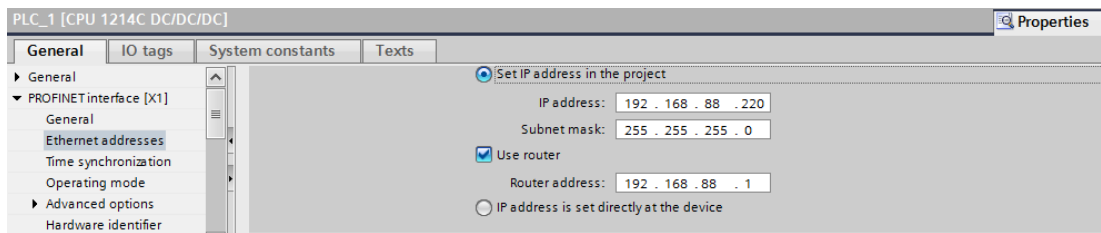
Αρχικά, πρέπει να επιλεχθεί το πεδίο «*Device Configuration*» που βρίσκεται στα αριστερά της οθόνης της κεντρικής επιφάνειας εργασίας (όπου και γίνεται η σύνταξη του κώδικα) εντός του πλαισίου «Project tree». Μόλις εμφανιστεί το νέο παράθυρο, διακρίνεται η συσκευή που έχει επιλεχθεί, τοποθετημένη σε μία ράγα στην οποία βρίσκονται κενές θέσεις για τυχόν προεκτάσεις εισόδων και εξόδων στο συγκεκριμένο *PLC*. Στο πεδίο «*Device Configuration*» εμπεριέχονται πληροφορίες που σχετίζονται με τις προδιαγραφές της εκάστοτε συσκευής που χρησιμοποιείται, αλλά και πλήθος άλλων ιδιοτήτων.



Εικόνα 38: Η μονάδα που επιλέξαμε.

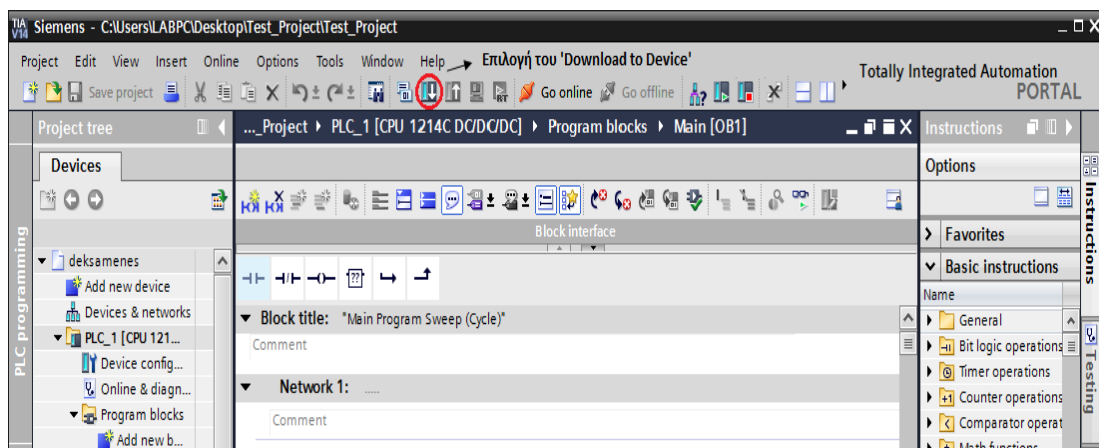
Στη συνέχεια, από τις ιδιότητες (*Properties*) του *PLC*, που βρίσκονται στη μπάρα στο κάτω μέρος της οθόνης, επιλέγουμε το πεδίο «*PROFINET interface*» και από εκεί το «*Ethernet addresses*», όπου και γίνεται ο ορισμός της *IP* διεύθυνσης. Ο χρήστης είναι πλέον σε θέση να εισάγει την επιθυμητή *IP* διεύθυνση στο *PLC* λαμβάνοντας όμως υπ' όψιν του ότι η *Subnet mask* θα πρέπει να είναι ίδια με αυτήν του υπολογιστή. Στο συγκεκριμένο παράδειγμα όπου η μάσκα υποδικτύου του υπολογιστή μας είναι 255.255.255.0, τότε και η *Subnet mask* που θα ορίσουμε θα είναι 255.255.255.0. Όταν θα ορίσουμε *IP address* οι πρώτες τρεις ομάδες αριθμών της *IP* διεύθυνσης του υπολογιστή που χρησιμοποιούμε και αυτής του *PLC* θα πρέπει να είναι ίδιες, ενώ η τελευταία ομάδα των δύο *IP* είναι η μόνη που πρέπει να διαφέρει. Το γεγονός αυτό γίνεται εμφανές στο συγκεκριμένο παράδειγμα, καθώς η *IP* του υπολογιστή είναι η 192.168.88.254 ενώ του *PLC* ορίσαμε ότι είναι 192.168.88.220. Η προεπιλεγμένη πύλη σύνδεσης με τις μονάδες *PLC* και *PC* είναι ένα *router* τότε επιλέγω και «*Router address*». Εάν συνδέουμε τον υπολογιστή και την μονάδα προγραμματιζόμενου λογικού ελεγκτή απευθείας, τότε δεν έχουμε *Default Gateway*.

Πάνω σε κάθε εκπαιδευτική μονάδα υπάρχουν τα στοιχεία *ip\_address*, *subnet\_mask* και *default gateway*.



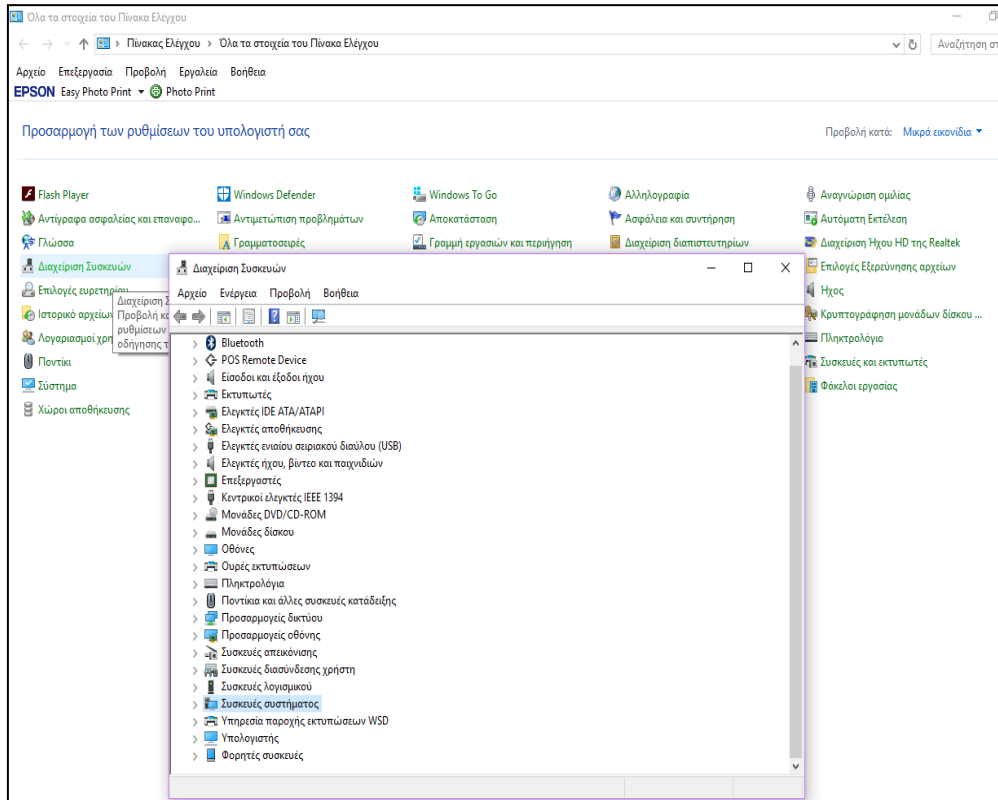
Εικόνα 39: Ορισμός στοιχείων συσκευής PLC για να ταιριάζουν με του υπολογιστή.

Μόλις ολοκληρωθεί η σύνταξη του κώδικα, μπορεί να πραγματοποιηθεί η εγγραφή του στη μνήμη του PLC σύμφωνα με την ακόλουθη διαδικασία. Αρχικά επιλέγουμε το εικονίδιο «Download to device» όπως διακρίνεται στην παρακάτω εικόνα.



Εικόνα 40: Επιλογή Download to Device για την εγκατάσταση του προγράμματος στη μονάδα PLC.

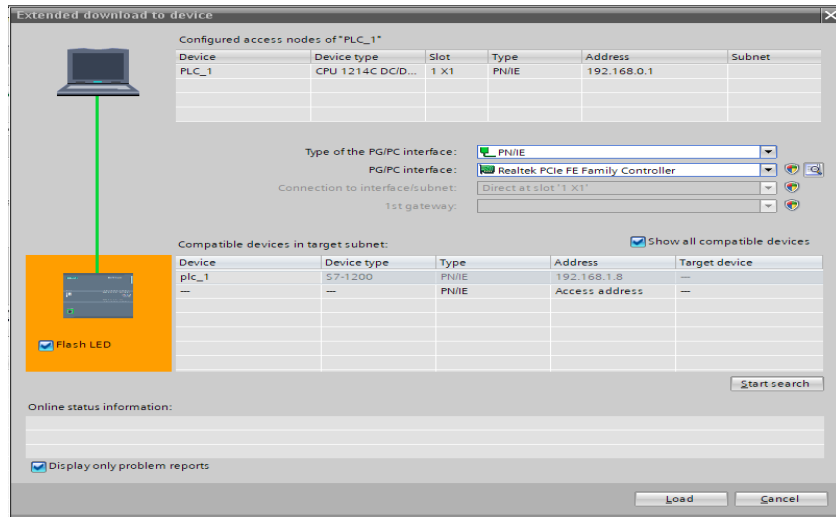
Σε αυτό το σημείο εμφανίζεται ένα νέο παράθυρο (βλ. εικόνα 42), στο οποίο ορίζεται το μοντέλο της κάρτας δικτύου του υπολογιστή που χρησιμοποιείται για τον προγραμματισμό του PLC στο πεδίο «PC/PG interface», αλλά και ο τύπος της διεπαφής (*Type of PC/PG interface*). Ο χρήστης μπορεί να ενημερωθεί για την κάρτα δικτύου που χρησιμοποιεί ο υπολογιστής του από την κατηγορία «Διαχείριση Συσκευών» του πίνακα ελέγχου των *Windows*.



Εικόνα 41: Έλεγχος συσκευών συστήματος.

Μόλις ολοκληρωθεί η εισαγωγή των πληροφοριών αυτών, το λογισμικό ξεκινά την αναζήτηση των διαθέσιμων συσκευών. Εάν δεν ξεκινήσει η αυτόματη αναζήτηση τότε μαρκάρουμε το πλαίσιο «*Show all compatible devices*». Αξίζει να αναφερθεί, πως σε περίπτωση που υπάρξει ασυμφωνία μεταξύ των *IP* διευθύνσεων του υπολογιστή και του *PLC*, εάν δηλαδή οι δύο *Subnet masks* είναι διαφορετικές άρα κάποιες από τις ομάδες αριθμών (εκτός από την τελευταία) των δυο *IP* διαφέρουν μεταξύ τους, το λογισμικό αποδίδει αυτόματα μια νέα *IP* στο *PLC*. Η νέα αυτή *IP* διεύθυνση, έχει την ίδια *Subnet mask* με αυτήν του υπολογιστή, γεγονός που επιτρέπει την εγκαθίδρυση της επικοινωνίας μεταξύ των δύο συσκευών και την εκκίνηση της διαδικασίας εγγραφής του κώδικα στο *PLC*.

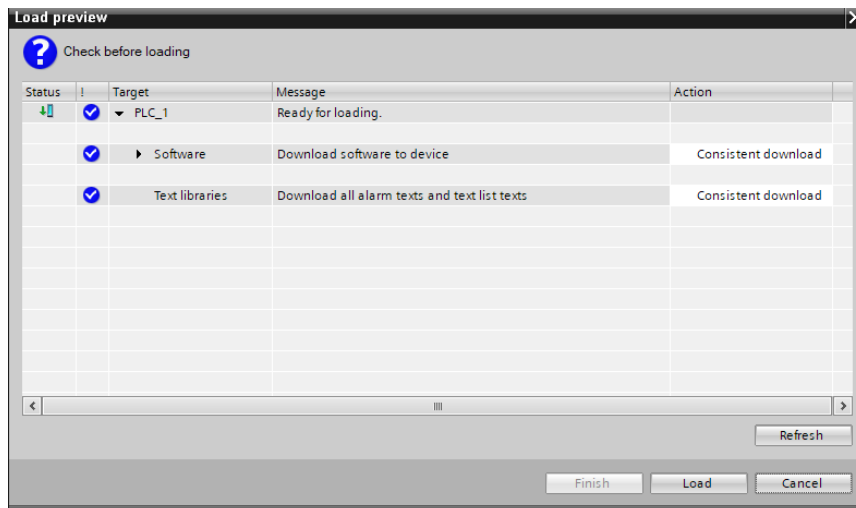




Εικόνα 42: Σύνδεση υπολογιστή με το PLC.

Τέλος, αφού ολοκληρωθεί επιτυχώς η αναζήτηση συσκευών, επιλέγουμε εκείνη με το *device* που είχαμε ορίσει εμείς (βλ. εικόνα 39) και στη συνέχεια επιλέγουμε την εντολή «*Load*» το οποίο βρίσκεται κάτω δεξιά στο παράθυρό μας. Για να ελέγξουμε ότι επιλέξαμε τη σωστή μονάδα επιλέγουμε την εντολή «*Flash LED*» και παρατηρούμε τα *LED* που βρίσκονται πάνω στη μονάδα *PLC*, στην οποία εργαζόμαστε, εάν αναβοσβήνουν.

Ως αποτέλεσμα έχουμε τη δημιουργία ενός νέου παραθύρου (*Load preview*), στο οποίο επιλέγεται εκ νέου το «*Load*».



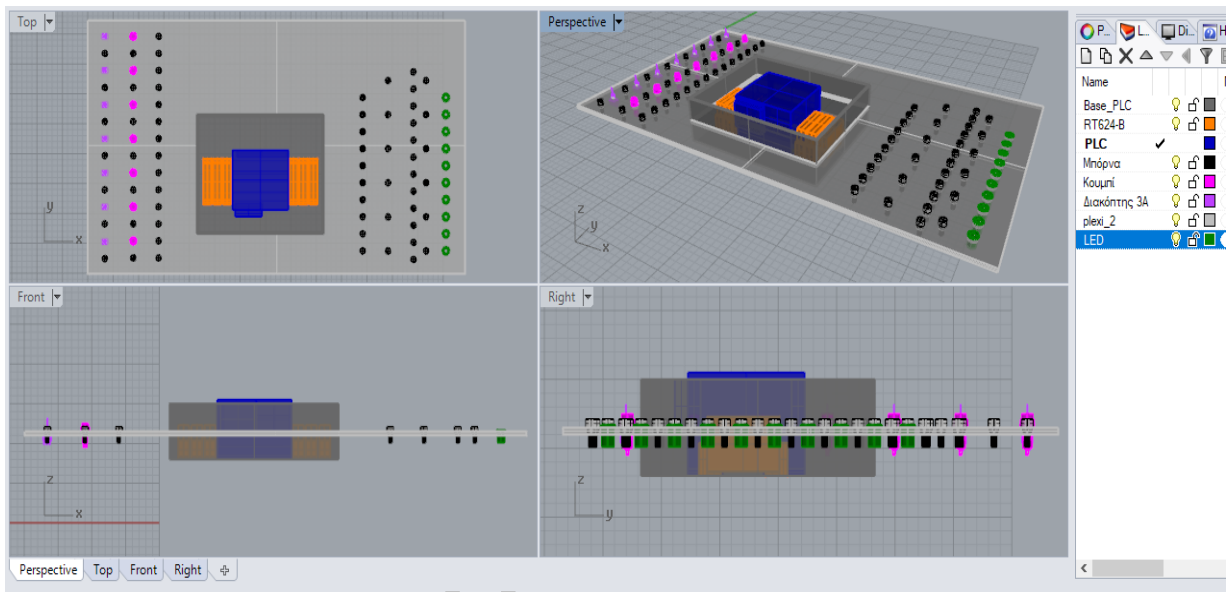
Εικόνα 43: Τελικός έλεγχος πριν τη φόρτωση του προγράμματος στο PLC.

## Κεφάλαιο 5<sup>ο</sup>: Σχεδίαση πρότυπης εκπαιδευτικής μονάδας με χρήση του λογισμικού Rhinoceros

Σε αυτό το κεφάλαιο θα γίνει αναφορά στο σχεδιαστικό τρόπο των μονάδων. Η οπτικοποίηση του πρωτοτύπου αποτελεί ένα σημαντικό κομμάτι της κατασκευής των εκπαιδευτικών μονάδων διότι πέρα από την εμφάνιση, βοηθάει στην βελτιστοποίηση του συστήματος.

Αρχικά με χρήση του προγράμματος *Rhinoceros*, έγινε η οπτικοποίηση μιας μονάδας, με σκοπό να είναι εύχρηστη, να γίνεται εύκολα κατανοητή η δομή της από τους σπουδαστές και να είναι όσο το δυνατόν πιο εύκολη στην αποθήκευση. Για να επιτευχθεί η οπτικοποίηση χρειάστηκε να περάσουμε σε μορφή 3D σχεδίου κάποια υλικά, τα οποία γνωρίζαμε ότι θα χρειαστούμε για την ομαλή λειτουργία των μονάδων όπως διακόπτες, ηλεκτρολογικό κουτί, *PLC* κ.ά. Με αυτό τον τρόπο είχαμε μια πραγματική απεικόνιση των μονάδων και έτσι μπορούσαμε να υπολογίσουμε πιο εύκολα και τις διαστάσεις των δομικών υλικών που θα χρησιμοποιούσαμε (οι μονάδες είναι κατασκευασμένες από *plexiglass* και ξύλο).

Πριν καταλήξουμε στο τελικό σχέδιο όπως είναι λογικό, πραγματοποιήθηκαν πολλές δοκιμές. Στις παρακάτω εικόνες μπορούμε να διακρίνουμε την πορεία τροποποίησης της μορφής της μονάδας στις τέσσερις όψεις (*Top*, *Perspective*, *Front* και *Right*).

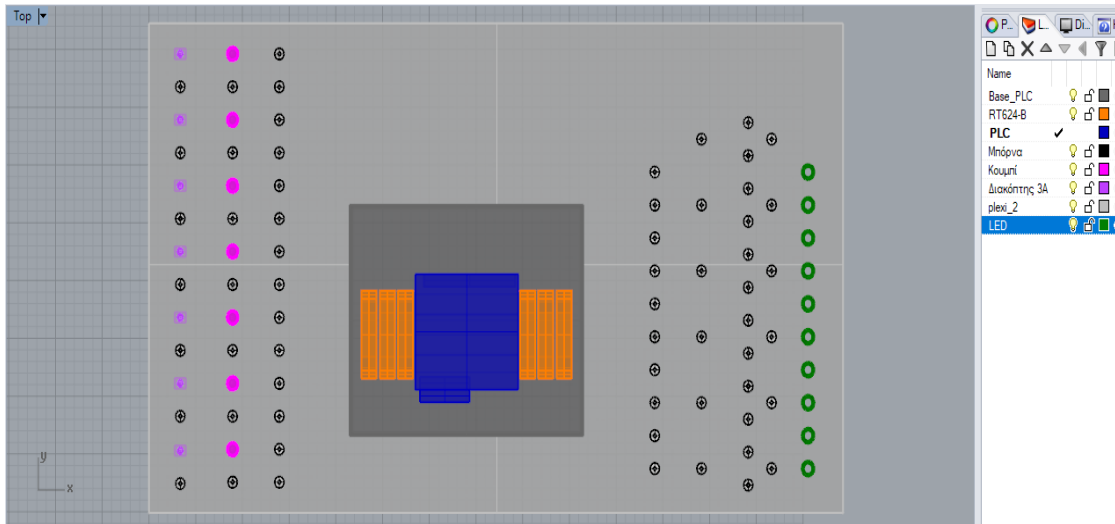


Εικόνα 44: Πρώτο σχέδιο για την οπτικοποίηση της εκπαιδευτικής μονάδας προγραμματισμού.

Κύριο βασικό χαρακτηριστικό των μονάδων είναι οι διασυνδέσεις (μπόρνες) των εισόδων, εξόδων και ηλεκτρονόμων (*Input*, *Output* και *relays*). Οι «μπόρνες» έχουν στόχο να διευκολύνουν τους χρήστες των μονάδων ώστε να μην χρειαστεί να συνδέουν καλώδια στις εισόδους και τις εξόδους του *PLC* (αλλά αφού έχει προηγηθεί ο προγραμματισμός των προγραμματιζόμενων λογικών ελεγκτών) να τα χειρίζονται μέσω διακοπών (*switches*) και

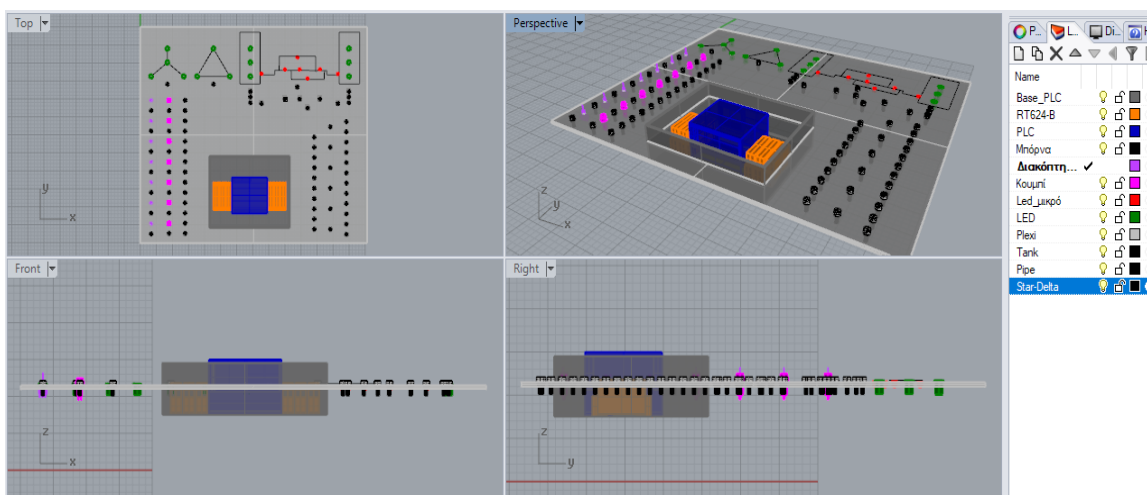
κουμπιών (*buttons*). Επιπλέον, με αυτό τον τρόπο προστατεύονται και οι ίδιες οι μονάδες των *PLC*.

Στη παρακάτω εικόνα μπορούμε να διακρίνουμε δεξιά τα *Layers* που χρησιμοποιήθηκαν, το χρωματικό κώδικα που επιλέξαμε με σκοπό την καλύτερη κατανόηση, και το είδος των υλικών.



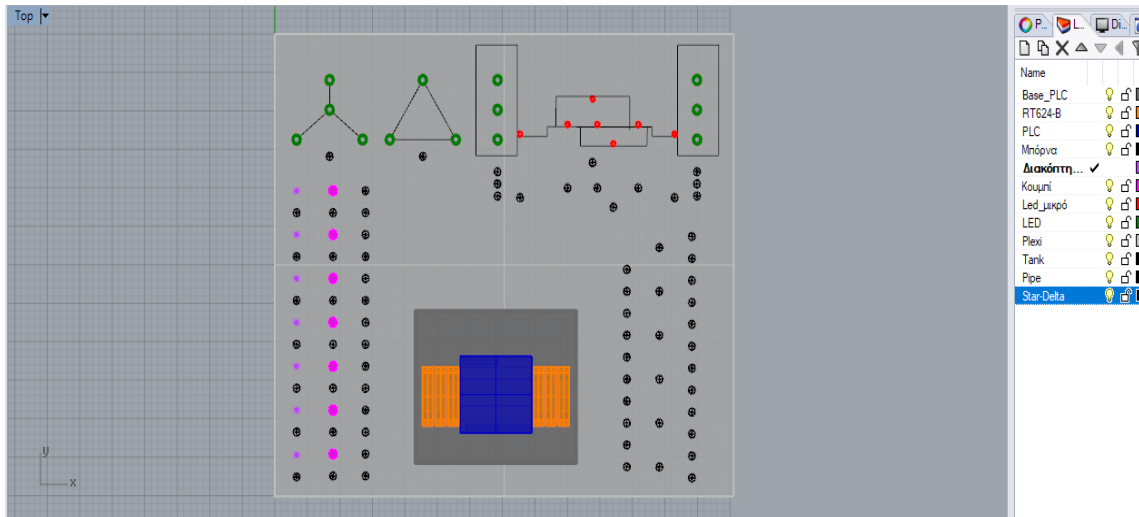
Εικόνα 45: Πρώτο σχέδιο για την οπτικοποίηση της εκπαιδευτικής μονάδας προγραμματισμού - Top όψη.

Στη συνέχεια το μοντέλο τροποποιήθηκε κατάλληλα και έγινε πιο συγκεκριμένο, για να μπορέσει να στηρίξει καλύτερα τις εργαστηριακές ασκήσεις, παίρνοντας την παρακάτω μορφή (βλ. εικόνα 46).



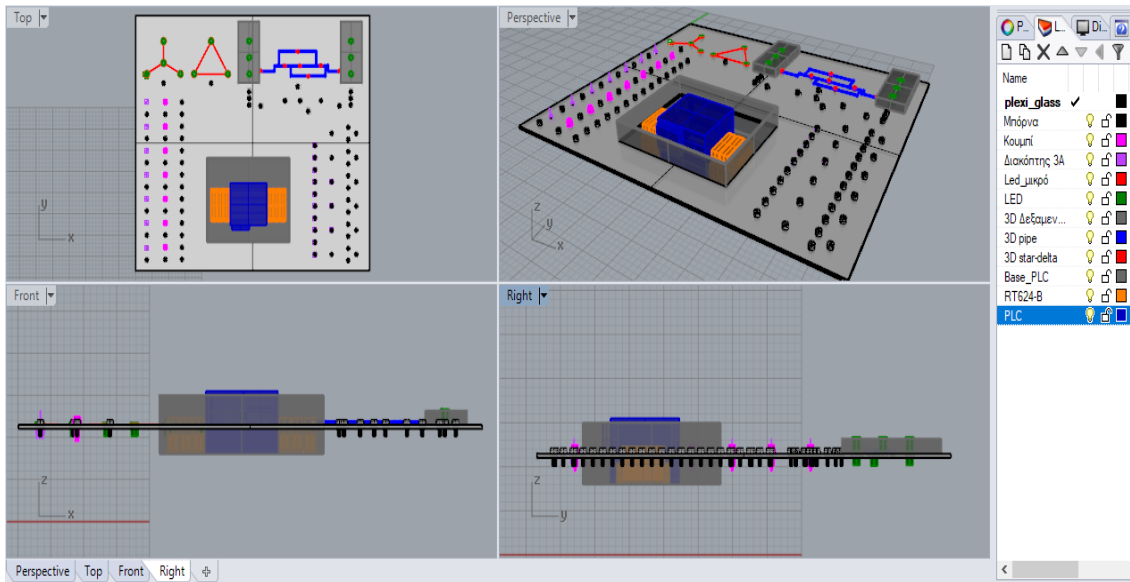
Εικόνα 46: Δεύτερο σχέδιο οπτικοποίησης της εκπαιδευτικής μονάδας προγραμματισμού.

Οι οπτικές ενδείξεις (*led*), τα οποία βρίσκονταν στη δεξιά πλευρά στο πρώτο σχέδιο, αντικαταστάθηκαν με μπόρνες που αντιστοιχούν σε ηλεκτρονόμους (*relays*), τα οποία βρίσκονται στο ηλεκτρολογικό κουτί. Καθώς, σκοπός των μονάδων εκμάθησης είναι οι χρήστες τους να κατανοήσουν την συνδεσμολογία και τον προγραμματισμό των *PLC*, όσο το δυνατό σε πραγματικές συνθήκες λειτουργίας σε σχέση με βιομηχανικές εφαρμογές ή εφαρμογές σε πλοία. Επιπλέον, συμπεριλήφθηκαν μονάδες ηλεκτρονόμων στην εκπαιδευτική μονάδα (Μπορεί στο εργαστήριο η τάση ενεργοποίησης διαφόρων *led* και εξαρτημάτων να είναι της τάσης των 24V, αυτό όμως δεν ισχύει σε μια τυπική εφαρμογή).



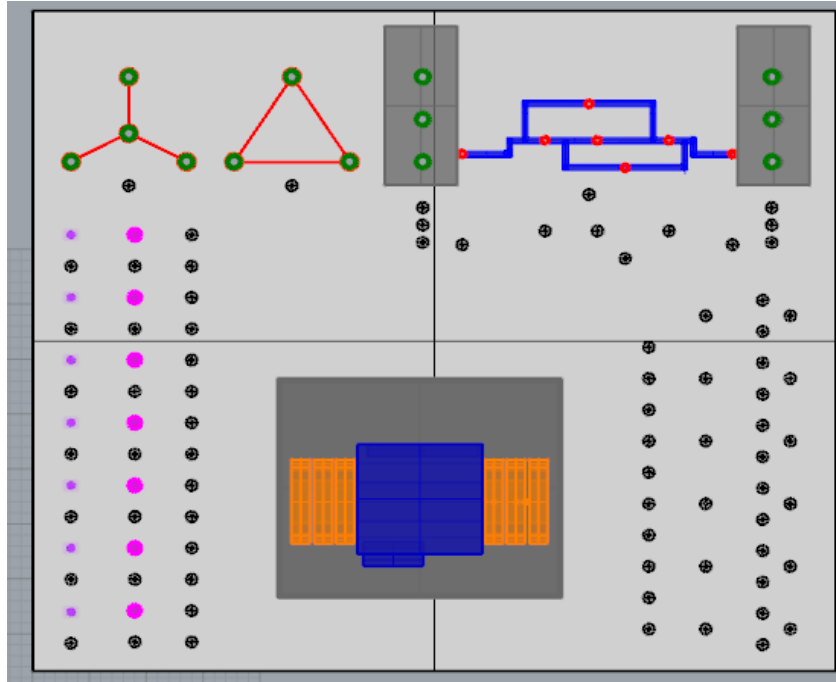
Εικόνα 47: Δεύτερο σχέδιο οπτικοποίησης της εκπαιδευτικής μονάδας - Top όψη.

Επιπλέον παρατηρούμε ότι υπάρχουν μαύρες ευθείες οι οποίες δημιουργούν την όψη δεξαμενών, δικτύου, έναν αστέρα και ένα τρίγωνο. Σκοπός των σχεδίων, τα οποία διαγράφονται με μαύρες γραμμές ήταν να εκτυπωθούν ψηφιακά πάνω στο *plexiglass*, με χρήση *UV* εκτυπωτή. Όμως η ιδέα τροποποιήθηκε μετατρέποντας τα επίπεδα σχέδια σε 3D σχέδια. Όπως παρατηρούμε στις παρακάτω εικόνες.



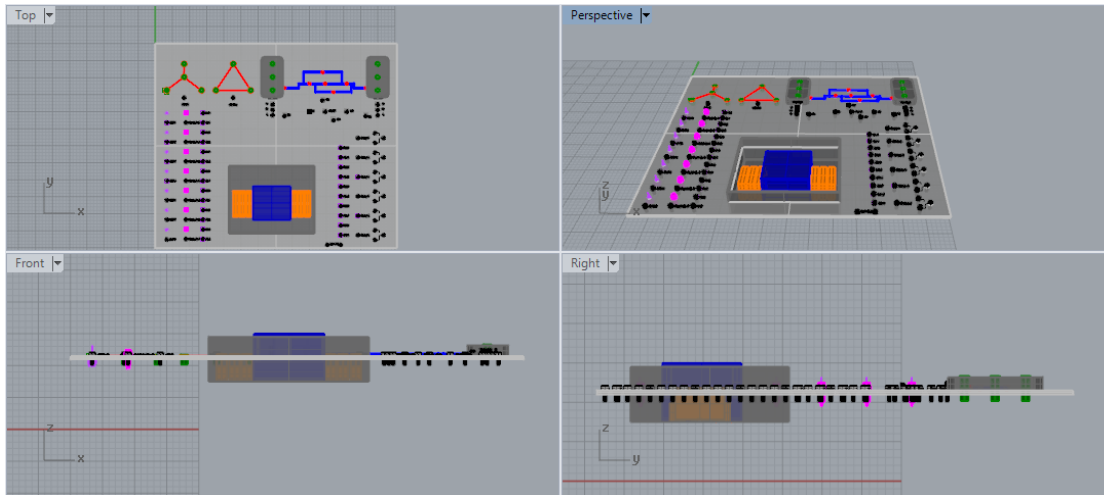
Εικόνα 48: Τρίτο σχέδιο οπτικοποίησης της εκπαιδευτικής μονάδας .

Όπως παρατηρούμε, το τρίτο σχέδιο είναι πιο συγκεκριμένο, εφόσον βασίζεται στις εργαστηριακές ασκήσεις. Παρατηρούμε ότι υπάρχουν 3D σχέδια στη θέση των γραμμών, τα οποία εκτυπώθηκαν με το 3D εκτυπωτή του εργαστηρίου.



Εικόνα 49: Τρίτο σχέδιο οπτικοποίησης της εκπαιδευτικής μονάδας - Top όψη.

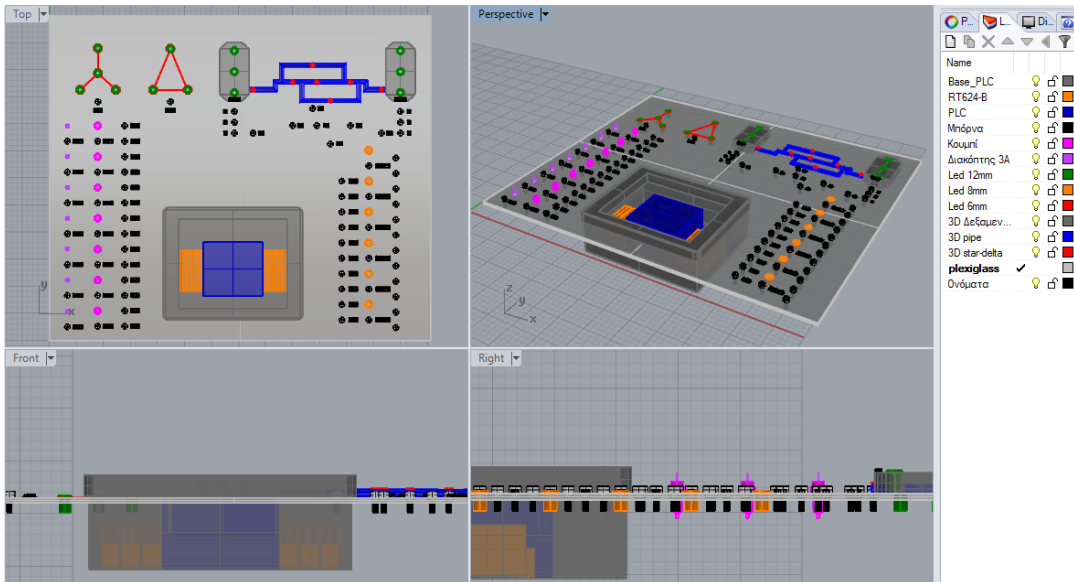
Στη συνέχεια, στο επόμενο σχέδιο οπτικοποίησης (4<sup>ο</sup> σχέδιο), έγιναν ακόμη μερικές τροποποιήσεις. Συγκεκριμένα, τοποθετήθηκαν μερικές ακόμη οπτικές ενδείξεις (*led*) στο σημείο όπου είναι οι ηλεκτρονόμοι. Επομένως, κάθε φορά όπου ένα «*relay*» ενεργοποιείται, τότε ενεργοποιείται και το «*led*». Έτσι είναι πιο εύκολο να κατανοήσουμε πότε θέτουμε αντίστοιχα ένα *relay* σε λειτουργία και να διακρίνουμε πιο εύκολα πιθανά λάθη συνδεσμολογίας. Παράλληλα έγιναν κάποιες διορθώσεις στα 3D σχέδια για καλύτερη εμφάνιση των μονάδων και τοποθετήθηκαν μπόρνες για την τροφοδοσία του συστήματος.



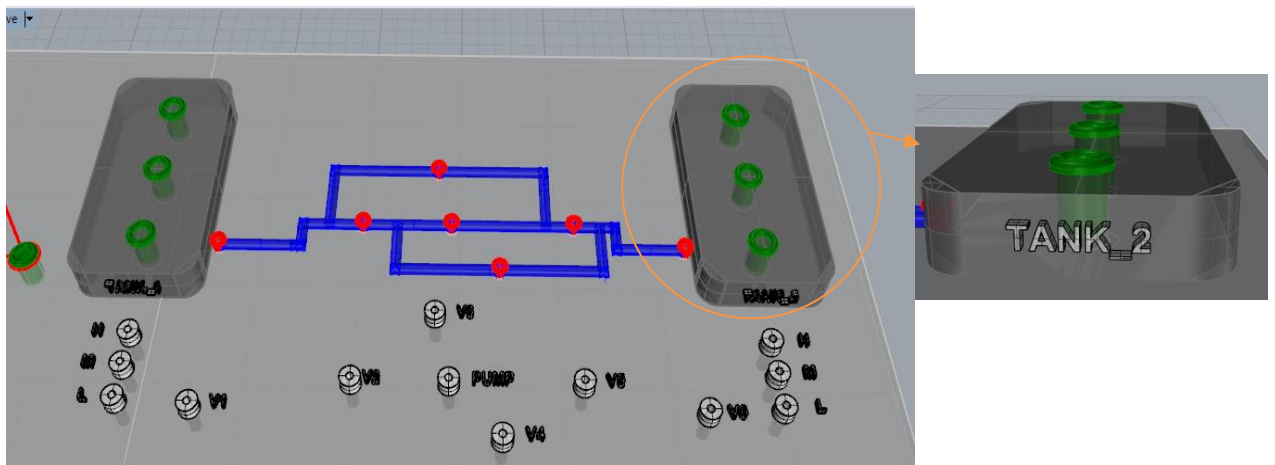
Εικόνα 50: Τέταρτο σχέδιο οπτικοποίησης της εκπαιδευτικής μονάδας.

Τέλος με σκοπό να προστατεύονται τα *relays* και το *plc*, έγινε μια διαμόρφωση στο καπάκι του ηλεκτρολογικού κουτιού, με σκοπό ο χρήστης να έχει ορατότητα του τι συμβαίνει στη μονάδα. Αποτελεί βασικό στοιχείο, διότι πάνω στο *plc* έχει αντίστοιχα *led* για την κατάσταση των *inputs* και των *outputs*. Επιπλέον σχεδιάστηκαν 3D ονόματα για να τοποθετηθούν πάνω στις μονάδες εκμάθησης, με σκοπό να βοηθήσουν στη καλύτερη κατανόηση της συνδεσμολογίας από το χρήστη, η οποία θα είναι ορατή λόγω του *plexiglass*.

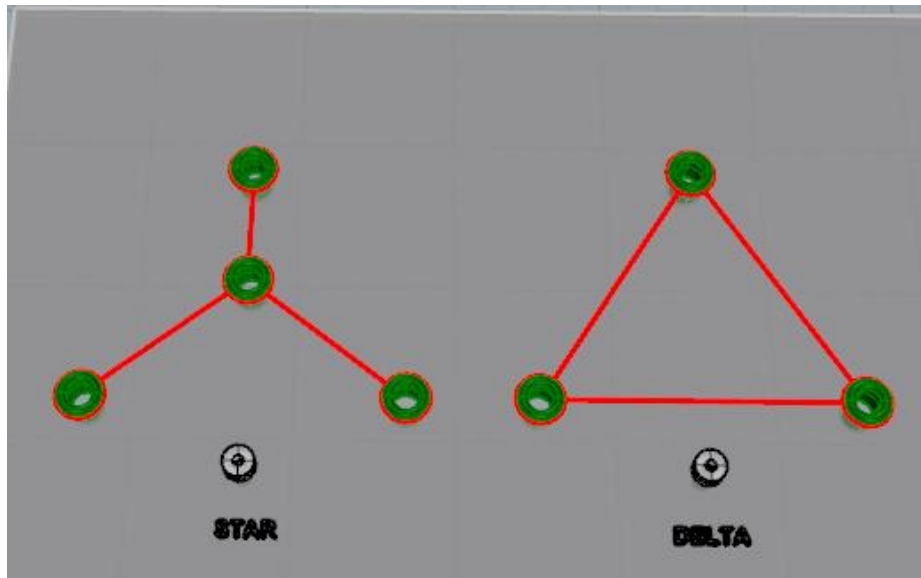
Παρακάτω ακολουθούν χαρακτηριστικές φωτογραφίες του τελικού σχεδίου.



Εικόνα 51: Τελικό σχέδιο οπτικοποίησης της εκπαιδευτικής μονάδας.



Εικόνα 52: Μεγέθυνση της μονάδας στο σημείο που είναι οι δεξαμενές. Η δεξιά εικόνα αποτελεί μεγέθυνση της δεξαμενής 2.



Εικόνα 53: Μεγέθυνση της μονάδας στο σημείο που είναι η συνδεσμολογία αστέρα – τρίγωνο.

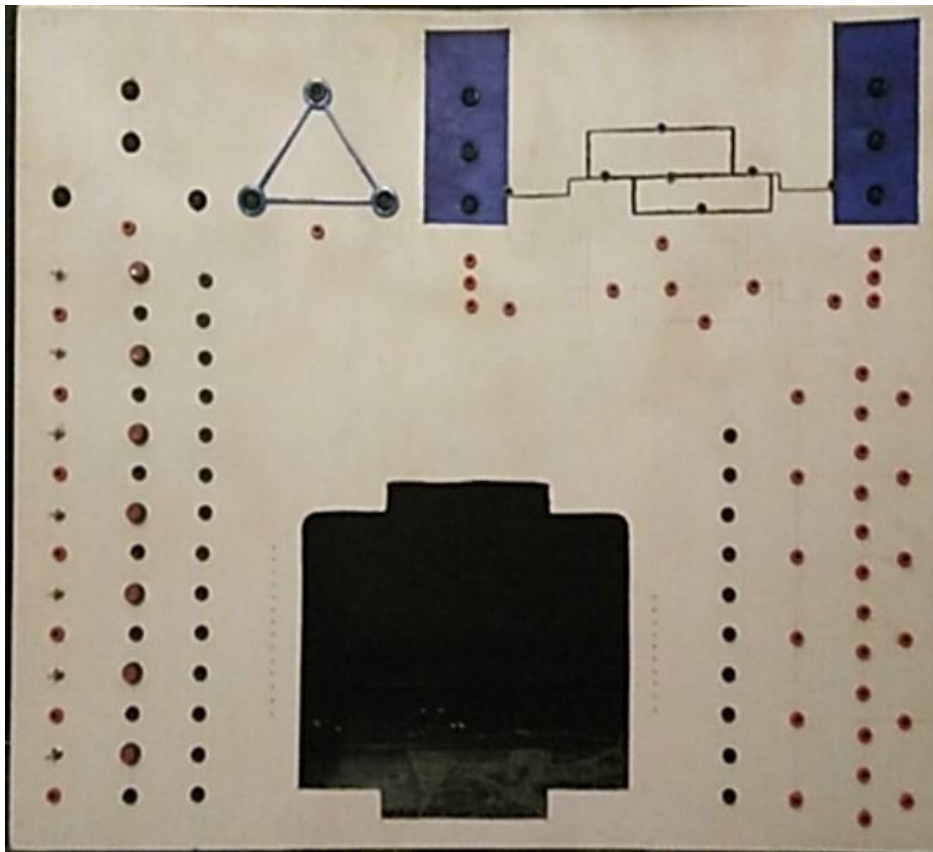
## Κεφάλαιο 6<sup>ο</sup>: Κατασκευή και συναρμολόγηση μονάδων εκμάθησης προγραμματισμού και διασύνδεσης βιομηχανικού τύπου PLC

Σε αυτό το κεφάλαιο θα αναφερθούμε στον τρόπο με τον οποίο κατασκευάστηκαν και συναρμολογήθηκαν οι μονάδες εκμάθησης προγραμματισμού και διασύνδεσης βιομηχανικού τύπου (PLC), οι οποίες θα χρησιμοποιηθούν στο πρόγραμμα σπουδών του τμήματος Ναυπηγών Μηχανικών του ΠΑ.Δ.Α.

### 6.1 Κατασκευή προτύπου μονάδας εκμάθησης

Αρχικά, προκειμένου να σιγουρευτούμε για τη λειτουργικότητα της αρχικής ιδέας, πριν ξεκινήσουμε την εργασία με τα *plexiglass*, κατασκευάστηκε ένα μοντέλο από ξύλο, παρόμοιο με τις διαστάσεις του *plexiglass* που είδαμε ότι θα χρειαστούμε από το σχεδιαστικό πρόγραμμα *Rhinoceros* (βλ. εικόνα 54).





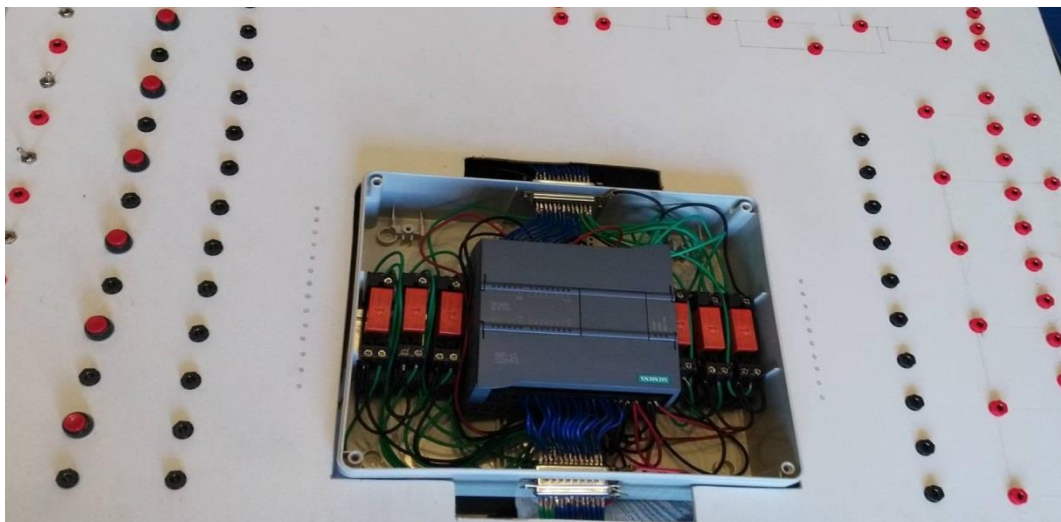
Εικόνα 54: Εύλινο μοντέλο.

Για την προστασία της μονάδας του προγραμματιζόμενου λογικού ελεγκτή (*PLC*), έχουν πραγματοποιηθεί όλες οι συνδέσεις που αφορούν τον αρνητικό πόλο της τροφοδοσίας (μαύρο καλώδιο), ώστε να μην χρειάζεται επέμβαση από το φοιτητή. Επομένως, ο σπουδαστής συνδέει μόνο με εξωτερικά καλώδια - βύσματα (τύπου μπανάνες) το θετικό πόλο (κόκκινο χρώμα) για να δημιουργήσει οποιοδήποτε κύκλωμα.

Για την καλύτερη κατανόηση και για την αποφυγή λαθών, τα χρώματα των καλωδίων που χρησιμοποιήσαμε είναι συγκεκριμένα και αναφέρονται στον παρακάτω πίνακα.

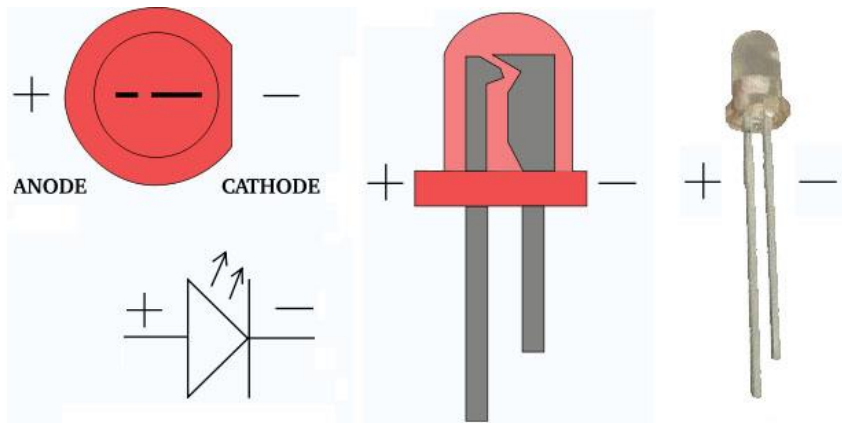
Πίνακας 2: Καλώδια εκπαιδευτικής μονάδας.

Χρήση	Χρώμα	Αγωγός
Inputs & Outputs	Μπλε	
Θύρες relays	Πράσινο	
Επιστροφή (αρνητικό ρεύμα)	Μαύρο	
Τροφοδοσία (θετικό ρεύμα)	Κόκκινο	



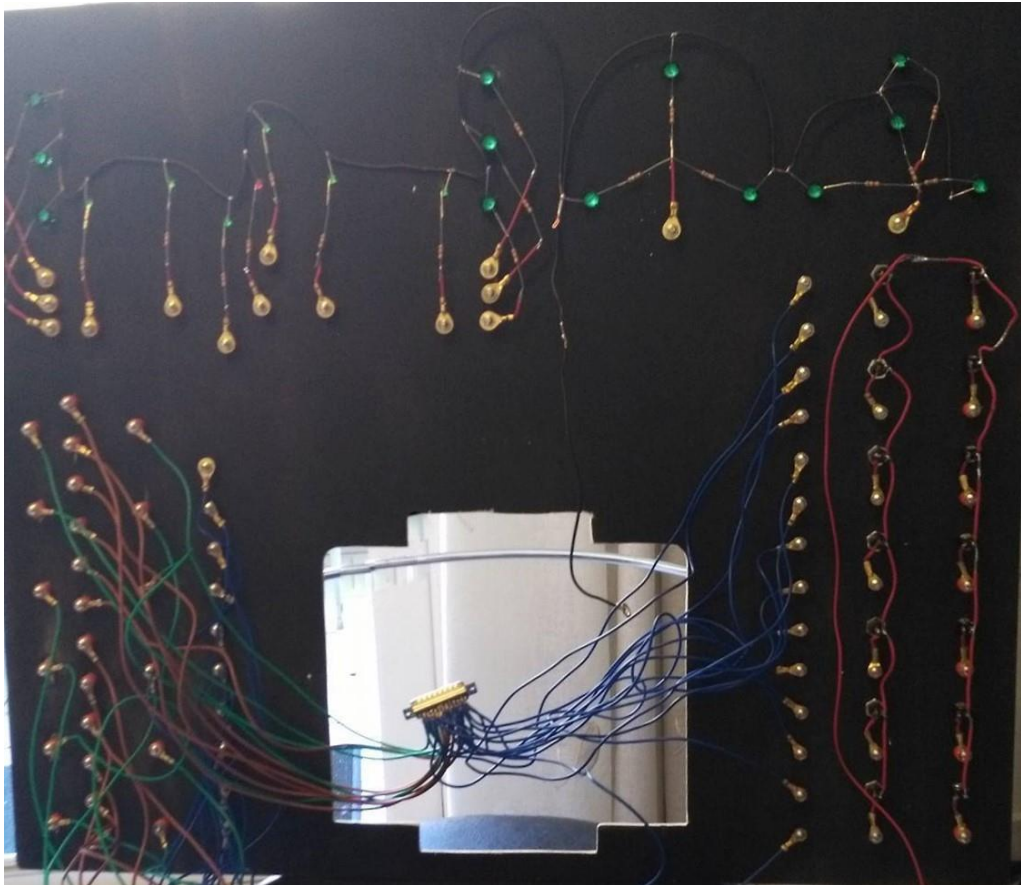
Εικόνα 55: Συνδεσμολογία PLC και relays.

Γενικότερα, τα *led* είναι διαφόρων ειδών αλλά πρέπει πάντα να είναι σε σειρά συνδεδεμένα με κατάλληλο αντιστάτη ώστε να περιορίζεται η ένταση του ρεύματος που τα διαπερνά (Πρακτικά το *led* αποτελεί μία δίοδο - αυτό σημαίνει ότι έχει πολικότητα, όπως φαίνεται στην παρακάτω εικόνα).



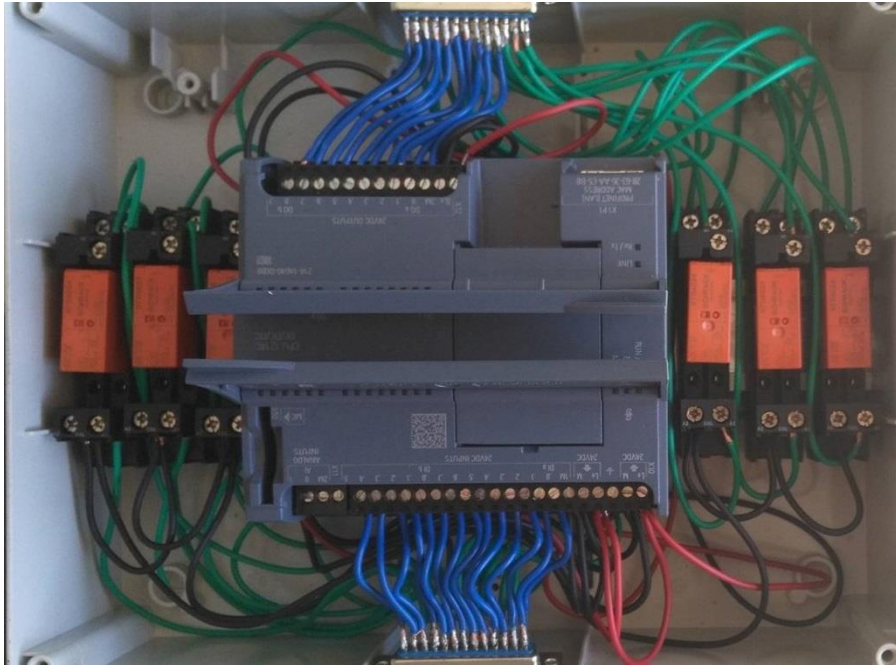
Εικόνα 56: Πολικότητα μικροελεγκτή led(<http://thecontroltheory.blogspot.gr>).

Επομένως κάθε *led* συνδέεται, με το θετικό πόλο (κόκκινο καλώδιο) σε σειρά με μια αντίσταση (3,9 ΚΩ), ενώ με τον αρνητικό πόλο (μαύρο καλώδιο) συνδέεται παράλληλα με τα υπόλοιπα στοιχεία (*relays*, υπόλοιπα *led* και *plc*). Σε κάθε επιμέρους στοιχείο, όπως οι διακόπτες, τα *led*, σήματα εισόδων και εξόδων του *PLC*, ηλεκτρονόμους, ο θετικός ακροδέκτης είναι συνδεδεμένος με κατάλληλο βύσμα «μπόρνα» διασύνδεσης.

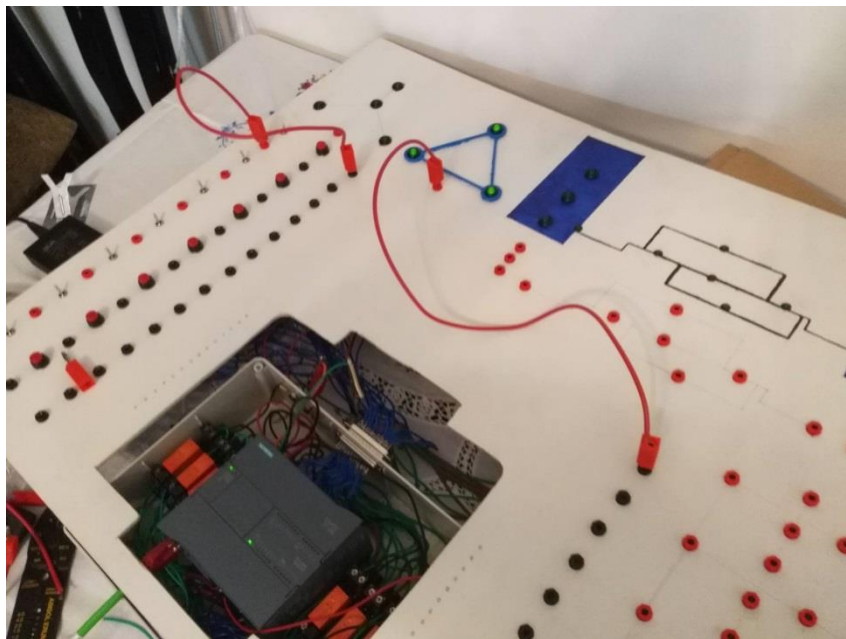


Εικόνα 57: Συνδεσμολογία κάτω από το plexiglass.

Αξίζει να σημειωθεί ότι πάνω στο *PLC* υπάρχουν συγκεκριμένες θύρες για το θετικό πόλο και τον αρνητικό πόλο της παροχής, οι οποίες συμβολίζονται με ένα *L* (θετικός) και με ένα *M* (αρνητικός). Επιπλέον, υπάρχει είσοδος για τη σύνδεση της τροφοδοσίας και έξοδος από την οποία παίρνουμε ρεύμα (θετικό και αρνητικό) για τη παροχή τροφοδοσίας σε επιμέρους συστήματα (π.χ: ηλεκτρονόμων).



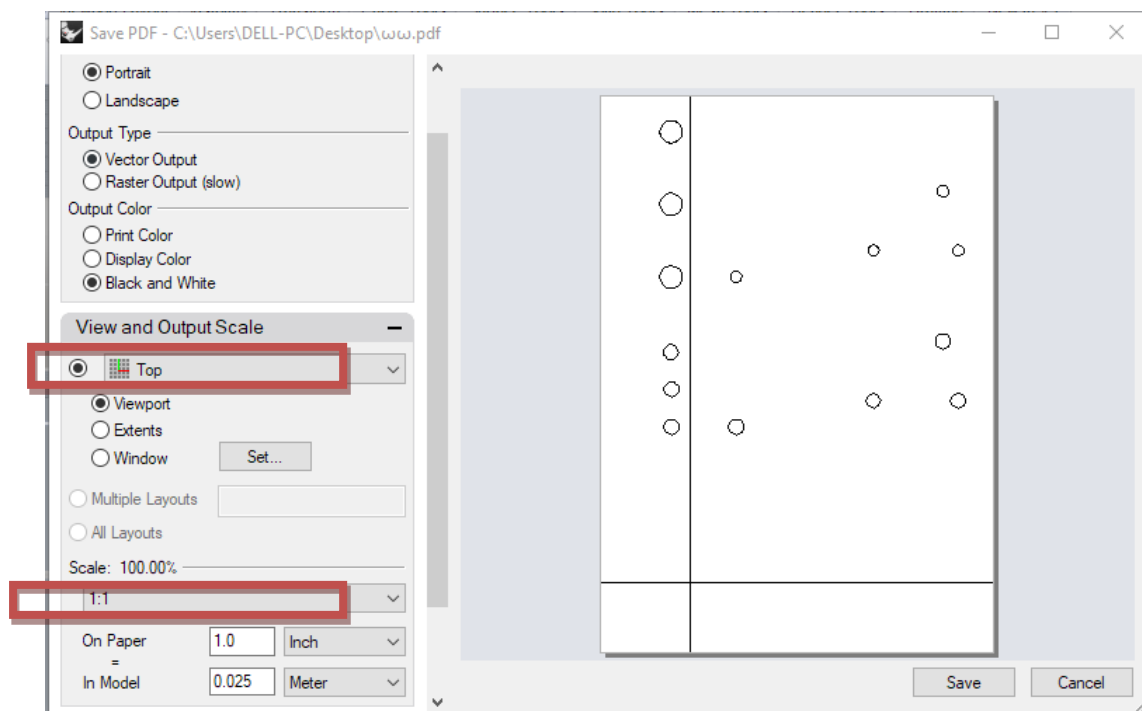
Εικόνα 58: Συνδεσμολογία PLC και relays.



Εικόνα 59: Έλεγχος σωστής συνδεσμολογία.

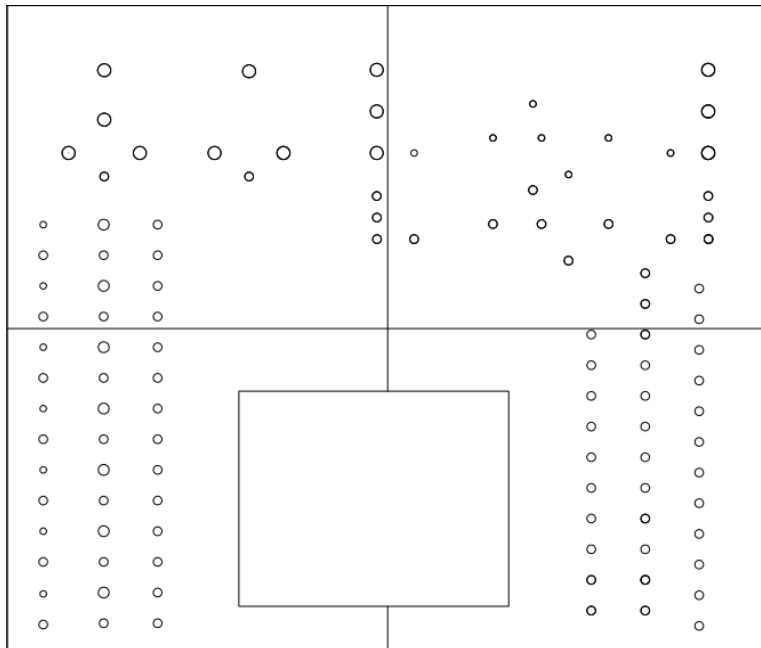
## 6.2 Κατασκευή μονάδων εκμάθησης

Μετά τη κατασκευή της πρότυπης μονάδας, ελέγξαμε την συνδεσμολογία και την ευκολία στη χρήση της. Στη συνέχεια μέσω του λογισμικού *TIA Portal* (βλ. κεφάλαιο 4<sup>ο</sup>) πραγματοποιήθηκε η δομικής λειτουργία της μονάδας με χρήση κατάλληλου κώδικα. Μετά την ολοκλήρωση προχωρήσαμε στην κατασκευή και συναρμολόγηση των μονάδων από *plexiglass*. Για την κατασκευή τους χρησιμοποιήσαμε κατάλληλο οδηγό για τη διάνοιξη των απαραίτητων οπών. Τον οδηγό αυτό τον από το σχέδιο του 3D προγράμματος σχεδίασης (*Rhinceros*) στο οποίο αναφερθήκαμε σε προηγούμενο κεφάλαιο.



Εικόνα 60: Αποθήκευση αρχείου από το Rhinoceros σε pdf μορφή.

Οπότε όταν θα εκτυπώσουμε το *pdf* αρχείο θα έχουμε έναν οδηγό με τις πραγματικές διαστάσεις του *plexiglass*.



Εικόνα 61: Οδηγός για την κατασκευή των plexiglass.

Οι οπές που πρέπει να δημιουργηθούν, σύμφωνα με το σχέδιο, έχουν τις παρακάτω διαμέτρους. Όλες οι τιμές μετριοούνται σε χιλιοστά (*mm*) (βλ. εικόνα 62).

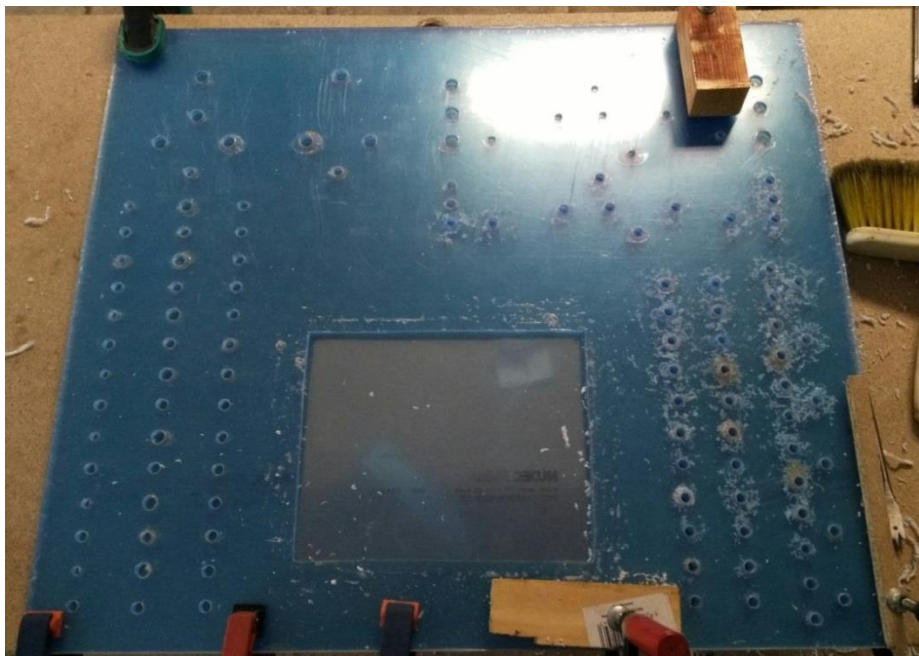


Εικόνα 62: Διάμετρος από χαρακτηριστικές διαστάσεις οπών που χρειάζεται να δημιουργηθούν στο plexiglass και τα αντικείμενα που θα τοποθετηθούν πάνω στην εκπαιδευτική μονάδα.



Εικόνα 63: Κοπή plexiglass με χρήση ηλεκτρικού πριονιού και τρυπανιού.

Να σημειωθεί ότι το υλικό *plexiglass*, σε σχέση με το ξύλο θέλει αρκετή προσοχή στην κοπή (Όταν υπερθερμανθεί μπορεί να σπάσει εύκολα). Στις εικόνες 63 και 64 παρουσιάζονται ενδεικτικά τα εργαλεία κοπής και ένα φύλλο *plexiglass* για την κοπή.



Εικόνα 64: Plexiglass μετά την κοπή.



Τέλος, για να εξασφαλίσουμε την προστασία του *PLC* από πιθανά χτυπήματα, και με στόχο παράλληλα ο χρήστης να μπορεί να ελέγχει τη λειτουργία του, τροποποιήσαμε τα ηλεκτρολογικά κουτιά τοποθετώντας *plexiglass* στην επάνω επιφάνεια (βλ. εικόνα 65).



Εικόνα 65: Καπάκια ηλεκτρολογικών κουτιών (δεξιά είναι πριν την επεξεργασία και αριστερά μετά την επεξεργασία).

Επόμενο βήμα στην κατασκευή των μονάδων είναι η βάση. Σκοπός της βάσης είναι η στήριξη του *plexiglass*, έτσι ώστε να μην σπάσει από την τοποθέτηση και αφαίρεση των βυσμάτων σύνδεσης (μπανάνες) ή γενικά από οποιαδήποτε άλλη δύναμη η οποία θα ασκηθεί στη μονάδα. Το υλικό που επιλέχτηκε ήταν το ξύλο για τη σχετικά εύκολη επεξεργασία του.



Εικόνα 66: Βάση εκπαιδευτικής μονάδας.

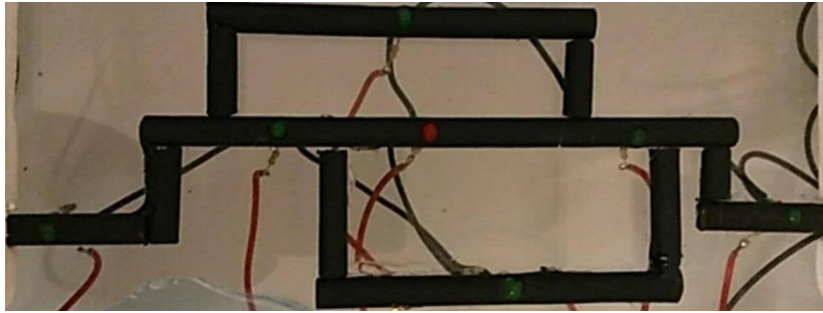
Για την μείωση των τάσεων που αναπτύσσονται στην μονάδα κατά τη διάρκεια της χρήσης της, τοποθετήθηκαν ξύλινα συμπαγή στυλώματα στα σημεία όπου θα είχαμε αυξημένη τάση. Επίσης, για τη συντήρηση και την αντοχή της βάσης από φθορά, τα ξύλα επεξεργάστηκαν με το τριβείο για να λειανθούν, στη συνέχεια βάφτηκαν με βελατούρα (αστάρι νερού για ξύλα) με σκοπό τη κάλυψη ατελειών του ξύλου και προετοιμασία για να δεχτεί την βαφή με έγχρωμο πλαστικό χρώμα.

Τελικό στάδιο της ολοκλήρωσης της οπτικοποίησης της εκπαιδευτικής μονάδας είναι τα τρισδιάστατα αντικείμενα (δεξαμενές, δίκτυο, αστέρας, τρίγωνο) τα οποία θα ενισχύσουν την καλύτερη κατανόηση των ασκήσεων και τη χρήση της μονάδας. Για την κατασκευή των δεξαμενών χρησιμοποιήθηκε μακετόχαρτο στις επιθυμητές διαστάσεις, το οποίο στη συνέχεια επεξεργάστηκε με γυαλόχαρτο για να πάρει σχήμα. Στις παρακάτω εικόνες παρουσιάζεται το εν λόγω υλικό πριν και μετά την επεξεργασία του.



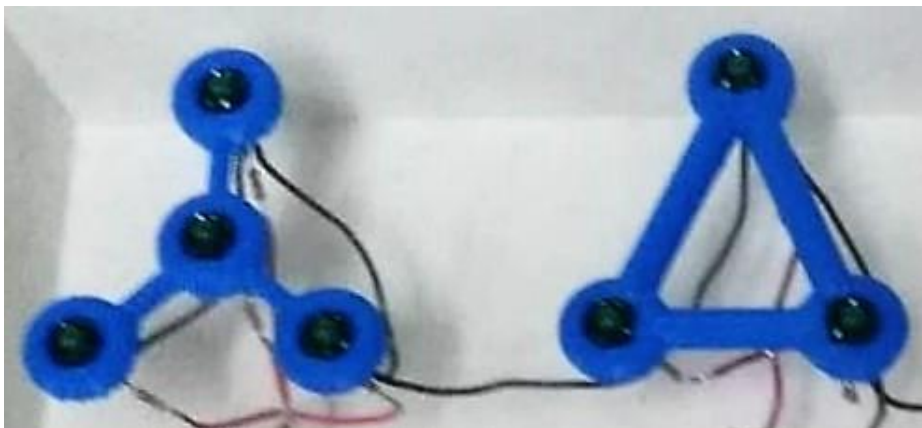
**Εικόνα 67:** Μακετόχαρτο πριν την επεξεργασία (αριστερή εικόνα) και μετά την επεξεργασία (δεξιά εικόνα).

Για την κατασκευή του δικτύου χρησιμοποιήθηκαν ξύλα, τα οποία κόπηκαν και επεξεργάστηκαν και θερμοσυστολικά για την προστασία και την οπτικοποίηση των ξύλων.



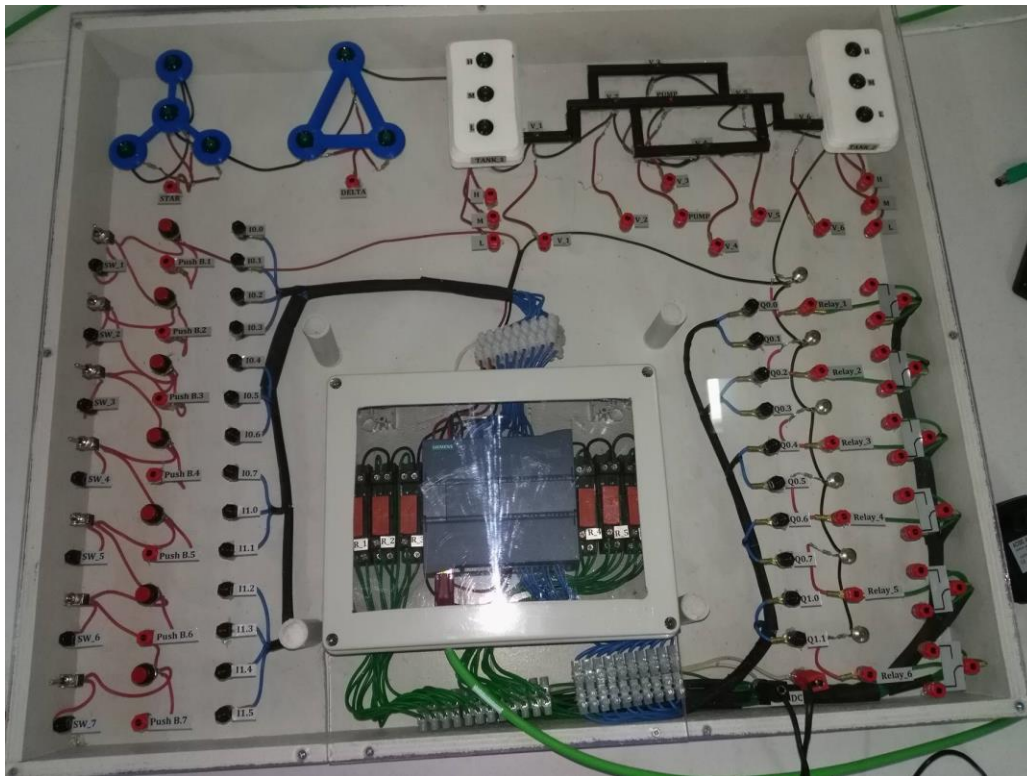
Εικόνα 68: Δίκτυο σωληνώσεων μετά τα θερμοσυστολικά.

Για την κατασκευή των πλαστικών αντικειμένων με σχήμα τριγώνου και αστέρα έγινε χρήση του 3D εκτυπωτή της σχολής. Η χρήση του 3D printer για την κατασκευή του αστέρα και του τριγώνου θα αναλυθεί περαιτέρω σε επόμενο κεφάλαιο (βλ. εικόνα 69).



Εικόνα 69: Αστέρας (αριστερά) και Τρίγωνο (δεξιά).

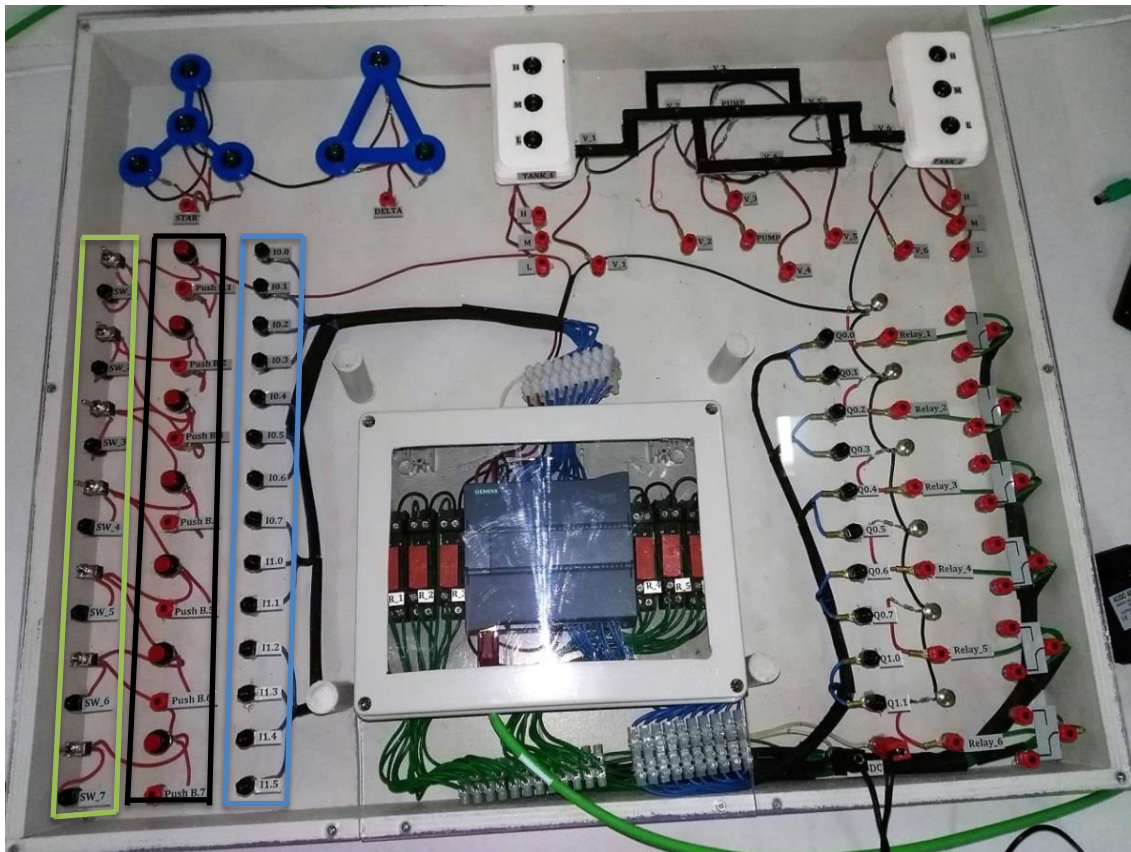
Στην παρακάτω εικόνα παρατηρούμε τη τελική μονάδα, η οποία αποτελείται από τις μπόρνες εισόδου (*Inputs*), τις μπόρνες που αντιστοιχούν σε διακόπτες (*Switches* και *push buttons*), τις μπόρνες εξόδου (*Outputs*), τις μπόρνες των ηλεκτρονόμων (*Relays*), καθώς και τις μπόρνες των φωτοδιόδων (*Led*).



Εικόνα 70: Εκπαιδευτική μονάδα εργαστηρίου.

### 6.3 Χωροταξική διαρρύθμιση των στοιχείων της μονάδας

Πιο αναλυτικά, οι μπόρνες εισόδου (*Inputs*), οι οποίες διακρίνονται στην παρακάτω εικόνα μέσα σε μπλε πλαίσιο, είναι συνδεδεμένες με τις θύρες εισόδου του *PLC*. Ανάλογα με την θύρα με την οποία συνδέεται κάθε μπόρνα, φέρει και το όνομά της, όπως αναγνωρίζεται από τη γλώσσα *Ladder* στο πρόγραμμα *TIA PORTAL V13*. Οι μπόρνες εισόδου συνδέονται τις περισσότερες φορές με κάποιον διακόπτη ή κάποιο κουμπί με σκοπό τον έλεγχο του κυκλώματός μας, πιο συγκεκριμένα συνδέονται στη μονάδα με κάποια μπόρνα διακόπτη, οι οποίες διακρίνονται στην παρακάτω εικόνα μέσα σε πράσινο πλαίσιο, ή κουμπιού, τα οποία διακρίνονται στην παρακάτω εικόνα μέσα σε μαύρο πλαίσιο. Τα ονόματα των *switches* και των *push buttons* οφείλονται στη σειρά τους (από πάνω προς τα κάτω) και δεν παίζουν κάποιο σημαντικό ρόλο στη συνδεσμολογία τους.



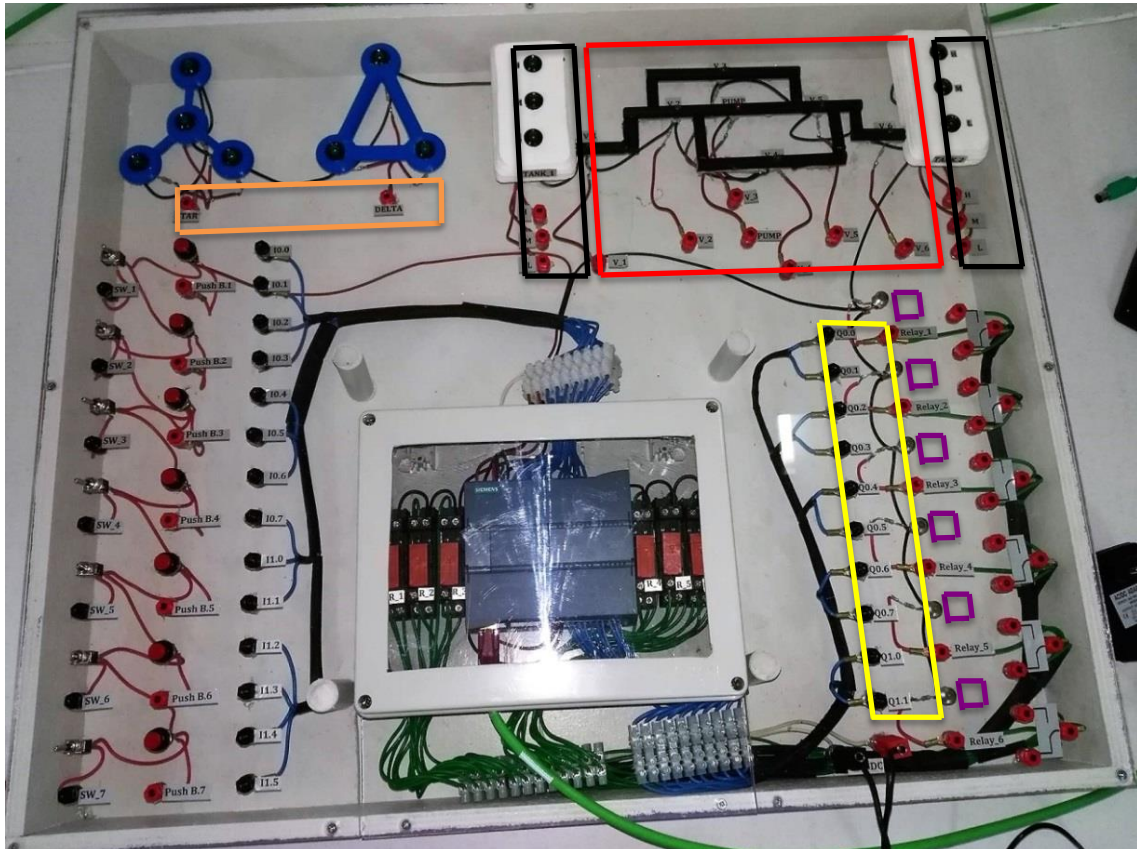
Εικόνα 71: Είσοδοι (inputs), διακόπτες (Switches) και κουμπιά (Push buttons) σε μπλε, πράσινο και μαύρο πλαίσιο αντίστοιχα.

Οι μπόρνες εξόδου (*Outputs*), οι οποίες διακρίνονται στην παρακάτω εικόνα μέσα σε κίτρινο πλαίσιο, είναι συνδεδεμένες με τις θύρες εξόδου του *PLC*. Ανάλογα με την θύρα με την οποία συνδέεται κάθε μπόρνα, φέρει και το όνομά της, όπως αναγνωρίζεται από τη γλώσσα *Ladder* στο πρόγραμμα *TIA PORTAL V13*. Οι μπόρνες εξόδου συνδέονται τις περισσότερες φορές με κάποια λυχνία ένδειξης (*led*), αντλία, μοτέρ, με σκοπό τον έλεγχο του κυκλώματος. Δηλαδή, στη μονάδα συνδέονται με κάποια μπόρνα η οποία αντιστοιχεί σε *led*, τα οποία διακρίνονται σε 4 κατηγορίες.

1. Τα *led* αστέρα (*Star*) και τρίγωνο (*Delta*). Πρόκειται για λυχνίες ένδειξης οι οποίες αναπαριστούν τη συνδεσμολογία αστέρα - τρίγωνο σε έναν τριφασικό κινητήρα. Διακρίνονται στην παρακάτω εικόνα μέσα σε πορτοκαλί πλαίσιο.
2. Τα *led* στις δεξαμενές (*Tanks*). Πρόκειται για λυχνίες ένδειξης οι οποίες αναπαριστούν το επίπεδο της στάθμης του νερού στη δεξαμενή, η οποία μετριέται με αισθητήρες στο εσωτερικό της δεξαμενής. Τα ονόματά των *led* δείχνουν το επίπεδο της στάθμης (*High, Medium, Low*). Διακρίνονται στην παρακάτω εικόνα μέσα σε μαύρο πλαίσιο.
3. Τα *led* των βαλβίδων (*Valve*) και της αντλίας (*Pump*). Πρόκειται για λυχνίες ένδειξης οι οποίες αναπαριστούν την λειτουργία των βαλβίδων και της αντλίας, όταν το *led* είναι αναμμένο. Όταν τα *led* δεν ανάβουν, τότε η αντλία δεν λειτουργεί και οι βαλβίδες είναι

κλειστές με αποτέλεσμα να μην επιτρέπουν την διέλευση του ρευστού από την μια δεξαμενή στην άλλη. Διακρίνονται στην παρακάτω εικόνα μέσα σε κόκκινο πλαίσιο.

4. Τα *led* των ηλεκτρονόμων (*Relays*). Πρόκειται για λυχνίες ένδειξης οι οποίες μας δείχνουν ότι ο ηλεκτρονόμος έχει τεθεί σε λειτουργία. Διακρίνονται στην παρακάτω εικόνα μέσα σε μοβ πλαίσιο.

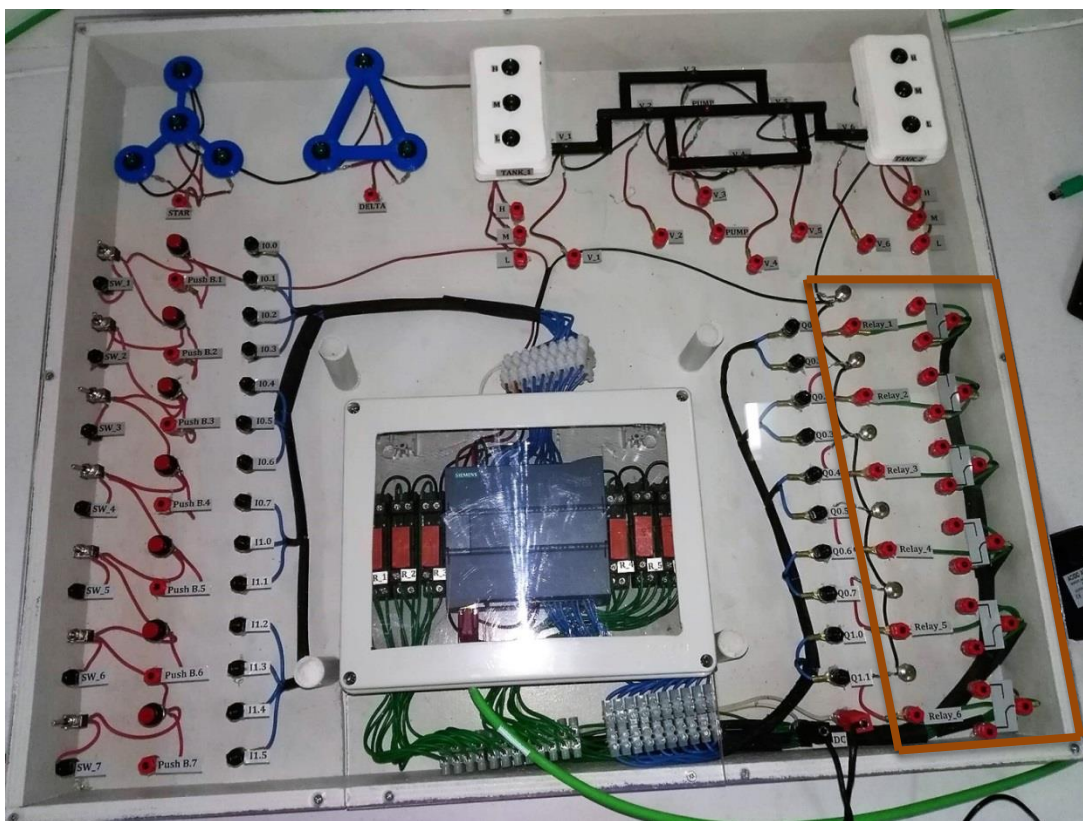


Εικόνα 72: Έξοδοι (Outputs) και Led εκπαιδευτικής μονάδας.

Οι μπόρνες των ηλεκτρονόμων (*Relays*), οι οποίες διακρίνονται στην παρακάτω εικόνα μέσα σε καφέ πλαίσιο, είναι συνδεδεμένες με τις επαφές των ηλεκτρονόμων, τα οποία βρίσκονται μέσα στο ηλεκτρολογικό κουτί. Οι ηλεκτρονόμοι οι οποίοι χρησιμοποιήθηκαν έχουν ένα πόλο. Έχουν μια επαφή η οποία είναι κανονικά κλειστή (*Normal Close*), δηλαδή αποσυνδέει το κύκλωμα όταν ενεργοποιείται ο ηλεκτρονόμος και μια κανονικά ανοικτή (*Normal Open*), η οποία συνδέει το κύκλωμα όταν ενεργοποιείται. Μπορούμε να διακρίνουμε τις επαφές αυτές από το σχέδιο που έχει κολληθεί πάνω στη μονάδα, το οποίο μας δείχνει το είδος της επαφής του ηλεκτρονόμου. Οι επαφές του ηλεκτρονόμου (*Normal Open* και *Normal Close*) συνδέονται συνήθως με κάποια λυχνία ένδειξης (*led*), αντλία, μοτέρ, με σκοπό το έλεγχο του κυκλώματός, ενώ η άλλη επαφή (11) συνδέεται με την τροφοδοσία.



Εικόνα 73: Εσωτερική σύνδεση ηλεκτρονόμου.



Εικόνα 74: Ηλεκτρονόμοι (Relays) εκπαιδευτικής μονάδας σε καφέ πλαίσιο.

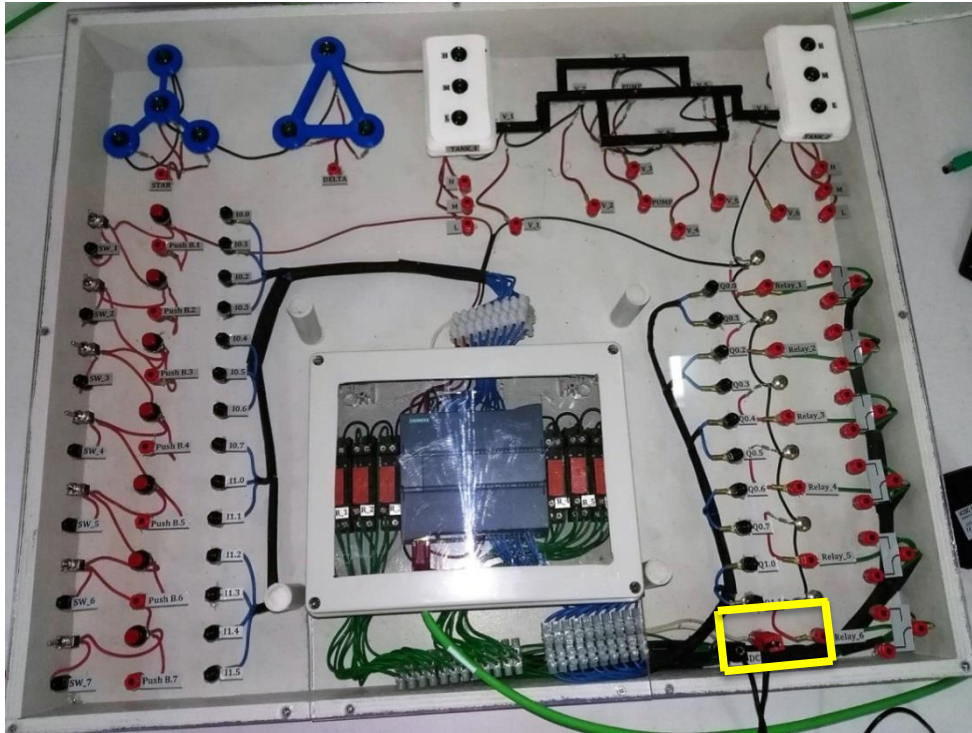
Τέλος, έχουμε τις μπόρνες της τροφοδοσίας της μονάδας, οι οποίες διακρίνονται στην παρακάτω εικόνα μέσα σε κίτρινο πλαίσιο. Το ρεύμα το οποίο παίρνουμε από τις πρίζες είναι εναλλασσόμενο (*Alternating Current – AC*), δηλαδή η έντασή του και η κατεύθυνσή του μεταβάλλονται περιοδικά, ενώ για να λειτουργήσει η μονάδα χρειαζόμαστε συνεχές ρεύμα (*Direct Current – DC*). Γι' αυτό χρησιμοποιούμε έναν προσαρμογέα (*adaptor*), ο οποίος δέχεται ρεύμα εναλλασσόμενο (100 - 240V) και μας δίνει σταθερό συνεχές ρεύμα (24V).



Εικόνα 75: Τροφοδοτικό μονάδας.

Για την προστασία του αυτόματου ελεγκτή προσέχουμε να τοποθετήσουμε τους ακροδέκτες του προσαρμογέα (μπανάνες) στις σωστές μπόρνες. Δηλαδή ο κόκκινος ακροδέκτης (θετικό ρεύμα) στην κόκκινη μπόρνα και ο μαύρος ακροδέκτης (αρνητικό ρεύμα) στην μαύρη μπόρνα της τροφοδοσίας (βλ. εικόνα 76).





Εικόνα 76: Σύνδεση τροφοδοτικού στη μονάδα.

Είναι σημαντικό να αναφερθεί ότι κάθε εκπαιδευτική μονάδα έχει στο μπροστινό της μέρος ένα ταμπελάκι με κάποια στοιχεία, τα οποία είναι σημαντικά για την σύνδεση της μονάδας με τον υπολογιστή. Όλες οι μονάδες έχουν ίδιο *subnet\_mask* και *default gateway* αλλά διαφορετικό *ip\_address*.



Εικόνα 77: Στοιχεία μονάδας για τη σύνδεση με τον υπολογιστή (Ethernet address).

## Κεφάλαιο 7<sup>ο</sup>: Χρήση τρισδιάστατου εκτυπωτή στην οπτικοποίηση της εκπαιδευτικής μονάδας

Γενικότερα, η τρισδιάστατη εκτύπωση (*3D printing*) είναι μια μέθοδος προσθετικής κατασκευής στην οποία κατασκευάζονται αντικείμενα μέσω της διαδοχικής πρόσθεσης επάλληλων στρώσεων υλικού. Χρησιμοποιείται κυρίως για την κατασκευή φυσικών μοντέλων και πρωτοτύπων από σχεδιαστές, μηχανικούς και ομάδες ανάπτυξης νέων προϊόντων. Έχει τη δυνατότητα να εκτυπώνει μέρη και εξαρτήματα από διάφορα υλικά, με διαφορετικές μηχανικές και φυσικές ιδιότητες και συχνά σε μια ενιαία διαδικασία κατασκευής. Για τις τρισδιάστατες εκτυπώσεις χρησιμοποιούνται διάφοροι τύποι υλικού, όπως πολυμερή και κεραμικά.

Σε σύγκριση με άλλες τεχνολογίες και εξοπλισμό προσθετικής κατασκευής, οι τρισδιάστατοι εκτυπωτές υπερτερούν συνήθως λόγω της ταχύτητας παραγωγής του αντικείμενου, το κόστος αγοράς τους και συντήρησής τους, καθώς και της εύκολης χρήσης τους.

Η τρισδιάστατη εκτύπωση ανακαλύφθηκε το 1982, από τον *Chuck Hull*. Ο *Hull* είχε την ιδέα ότι αν μπορούσε να τοποθετήσει χιλιάδες λεπτά στρώματα πλαστικού το ένα πάνω από το άλλο και στη συνέχεια να χαράξει το σχήμα τους, τότε θα ήταν σε θέση να σχηματίσει τρισδιάστατα αντικείμενα. Συνειδητοποίησε ότι το εύρημά του δεν περιοριζόταν σε υγρά στοιχεία και ως εκ τούτου η ευρεσιτεχνία του ονομάστηκε στερεολιθογραφία ή *3D* εκτύπωση, καθώς κάλυπτε κάθε υλικό ικανό προς στερεοποίηση ή ικανό να μεταβάλει τη φυσική του κατάσταση. Η παρεχόμενη τεχνολογία αποτελεί εξέλιξη των μηχανών και τεχνολογιών *RP* (*Rapid Prototyping*) που αναπτύχθηκαν από την ανάγκη «εκτύπωσης» (κατασκευής) των ψηφιακών δεδομένων *CAD*. Οι τεχνικές και οι μέθοδοι που αναπτύχθηκαν για την ταχεία προτυποποίηση (*RP*) χρησιμοποιήθηκαν αποκλειστικά στη βιομηχανία αυτοκινήτων και κάθε μία από αυτές έχει ιδιαίτερα πλεονεκτήματα: υλικά, ποιότητα, ταχύτητα, δυνατότητα χρήσης χρώματος. Στον παρακάτω πίνακα διακρίνουμε τις πιο γνωστές μεθόδους *RP*.

Πίνακας 3: Μέθοδοι *Rapid Prototyping*.

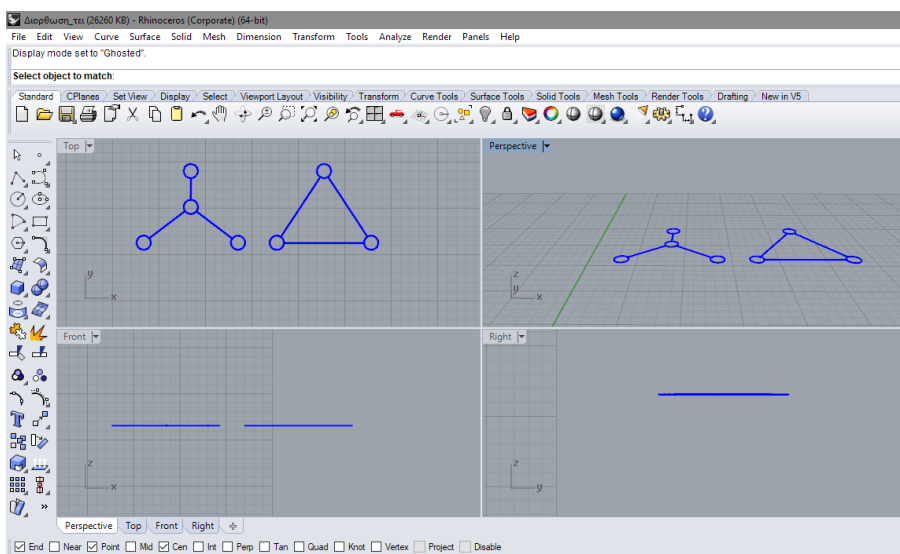
Αρχικά Μέθοδου	Μέθοδος	Περιγραφή
<i>STL</i>	<i>Stereolithography</i>	Στερεολιθογραφία.
<i>SGC</i>	<i>Solid Ground Curing</i>	Παραλλαγή της στερεολιθογραφίας.
<i>SLS</i>	<i>Selective laser sintering</i>	Επιλεκτική σύντηξη με χρήση ακτίνων <i>laser</i> .
<i>LOM</i>	<i>Laminated Object Manufacturing</i>	Συγκόλληση λεπτών φύλων υλικού.
<i>MDF</i>	<i>Fused Deposition Modeling</i>	Κατασκευή μοντέλων μέσω σύντηξης διαδοχικών στρώσεων.
<i>Ink-Jet</i>	<i>3D ink-jet printing</i>	Τεχνική τρισδιάστατης εκτύπωσης <i>ink-jet</i> .
<i>DLP</i>	<i>Digital light processing</i>	Ψηφιακή επεξεργασία φωτός (κάθε διατομή εκτίθεται σε ένα προβολέα <i>Dlp</i> ).
<i>EBM</i>	<i>Electron beam melting</i>	Μέθοδο τήξεως με δέσμη ηλεκτρονίων (σε κενό).

Η τρισδιάστατη εκτύπωση πραγματοποιείται με την αποστολή ενός ψηφιακού αρχείου *CAD*, σε έναν τρισδιάστατο εκτυπωτή με παρόμοιο τρόπο όπως εκτυπώνεται ένα έγγραφο σε έναν εκτυπωτή *ink-jet* ή *laser*. Ελέγχεται η λειτουργικότητα και τα επιμέρους χαρακτηριστικά. Εντοπίζονται λάθη και αστοχίες, που οδηγούν σε πιθανό επανασχεδιασμό και επανεκτύπωση σε σύντομο χρόνο. Κατασκευάζονται εργαλεία ακριβείας (*Rapid Tooling*).

Ανεξάρτητα από τη μέθοδο και τη τεχνολογία που χρησιμοποιεί ένας τρισδιάστατος εκτυπωτής η κατασκευή του μοντέλου πραγματοποιείται σε πέντε στάδια:

### 1. Δημιουργία ψηφιακού μοντέλου *CAD* του σχεδίου.

Με χρήση του προγράμματος *Rhinoceros* κατασκευάστηκαν τα αντικείμενα τα οποία θέλαμε να εκτυπώσουμε στον 3D εκτυπωτή.



Εικόνα 78: Ψηφιακό μοντέλο σε *CAD* πρόγραμμα (*Rhinoceros*).

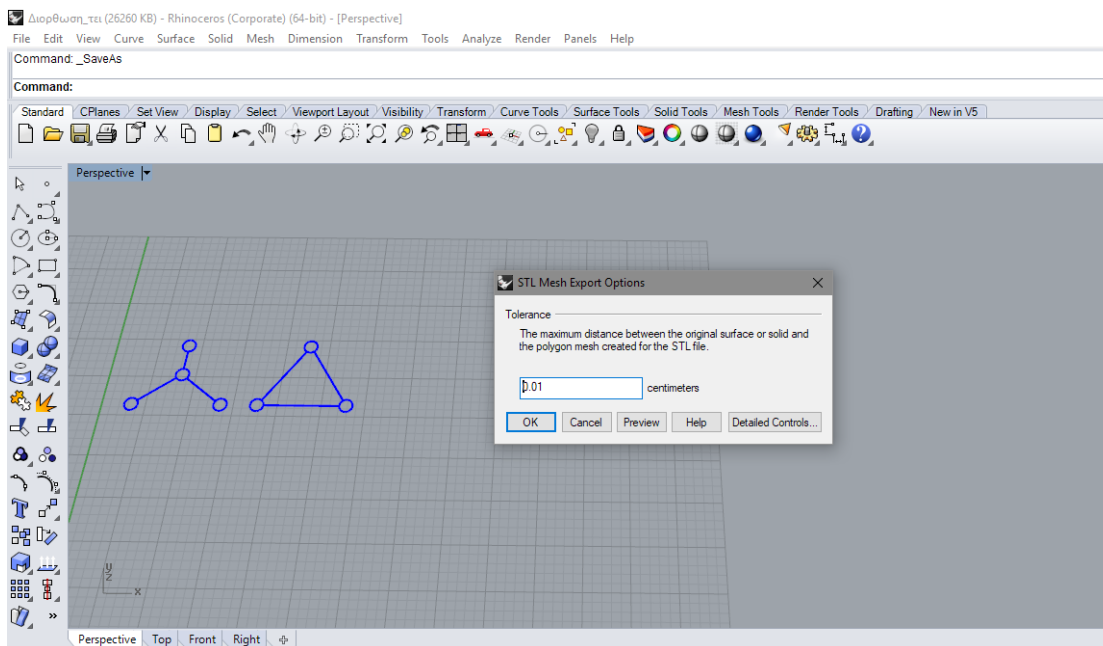
### 2. Μετατροπή του μοντέλου *CAD* σε *format STL* (στερεολιθογραφικό).

Το *STL* αναπαριστά μια τρισδιάστατη επιφάνεια ως ένα σύνολο στοιχειωδών επίπεδων τριγώνων, που δεν μπορούν να αναπαραστήσουν καμπύλες επιφάνειες, παρά μόνο κατά προσέγγιση. Όσο αυξάνονται τα τρίγωνα, αυξάνεται το μέγεθος και η πολυπλοκότητα του αρχείου με αποτέλεσμα να χρειάζεται περισσότερος χρόνος επεξεργασίας και κατασκευής. Το λογισμικό του προγράμματος «τεμαχίζει» το ψηφιακό μοντέλο σε έναν αριθμό λεπτών εγκάρσιων διατομών (πάχους 0,01 έως 0,5 mm, ανάλογα με την τεχνική εκτύπωσης), ενώ

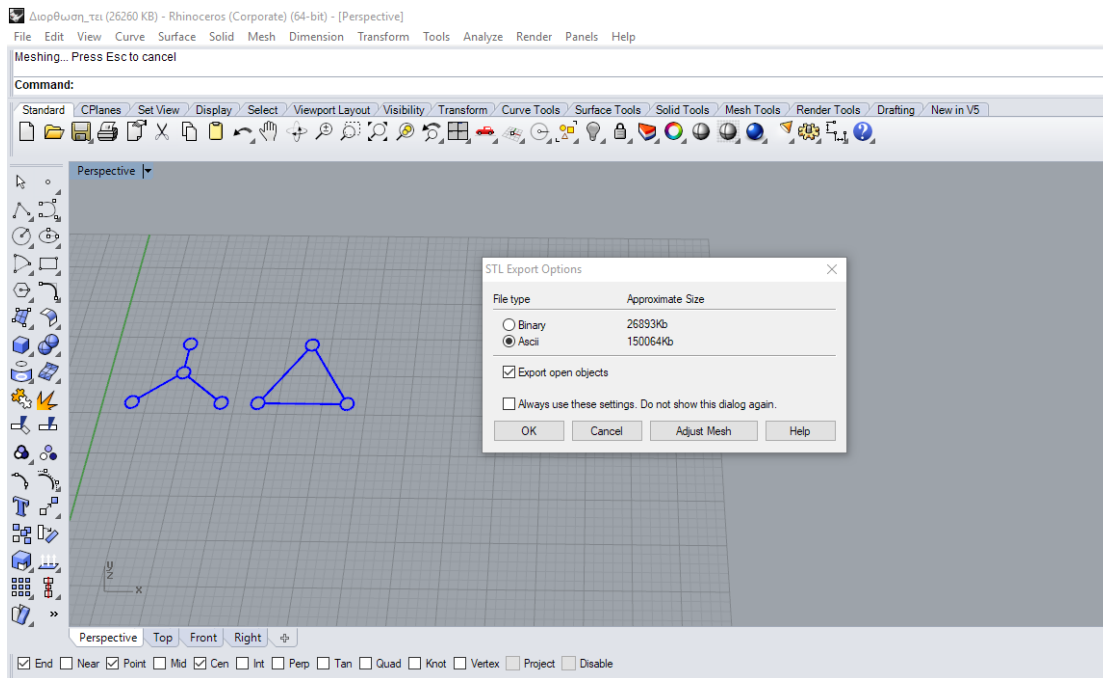
μπορεί να δημιουργήσει και μια βοηθητική δομή, για τη στήριξη του πρωτοτύπου. Επιτρέπουν τη ρύθμιση του μεγέθους, της θέσης και του προσανατολισμού.

Όταν αποθηκεύσουμε το αρχείο μας σε μορφή *STL* θα μας εμφανίσει 2 νέα παράθυρα πριν την αποθήκευσή του. Στο πρώτο (βλ. εικόνα 79) μας ζητάει να επιλέξουμε το πάχος του πλέγματος του αντικειμένου. Ενώ στο δεύτερο (βλ. εικόνα 80) ζητάει το είδος του αρχείου (Συνήθως στις οδηγίες του εκτυπωτή αναφέρει. Στην προκειμένη περίπτωση χρησιμοποιήσαμε *ASCII*).

Στη συνέχεια χρησιμοποιείτε το λογισμικό «*CubeX\_3D System*» για περαιτέρω επεξεργασία του σχεδίου.



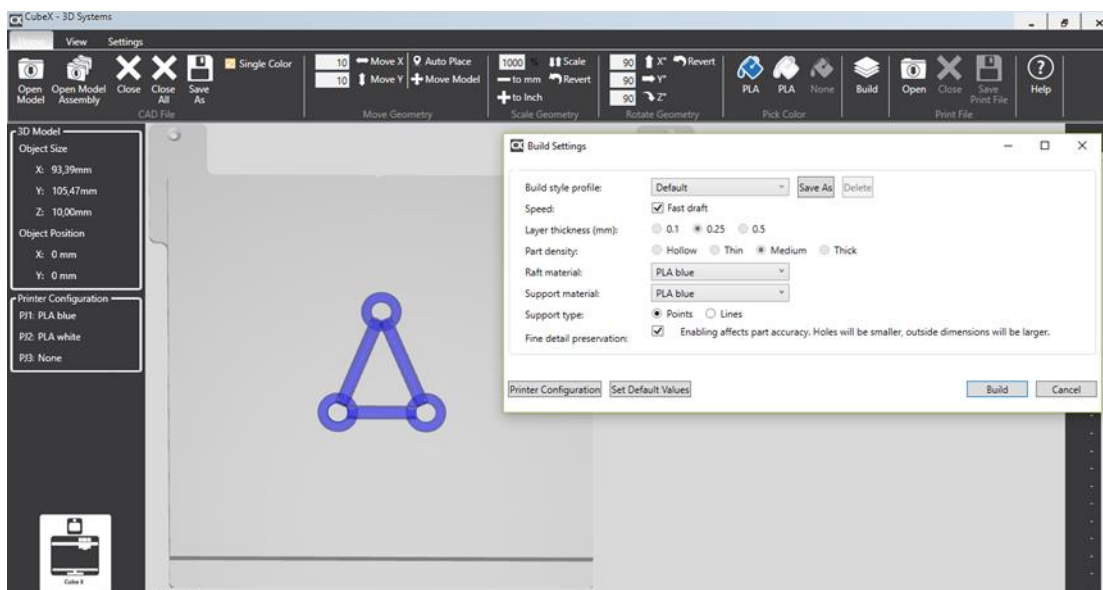
Εικόνα 79: Αποθήκευση σε μορφή αρχείου STL.



Εικόνα 80: Αποθήκευση σε μορφή αρχείου STL.

3. «Τεμαχισμό» με τη χρήση κατάλληλου λογισμικού, του αρχείου *STL* σε λεπτές «διατομές» ελάχιστου πάχους.

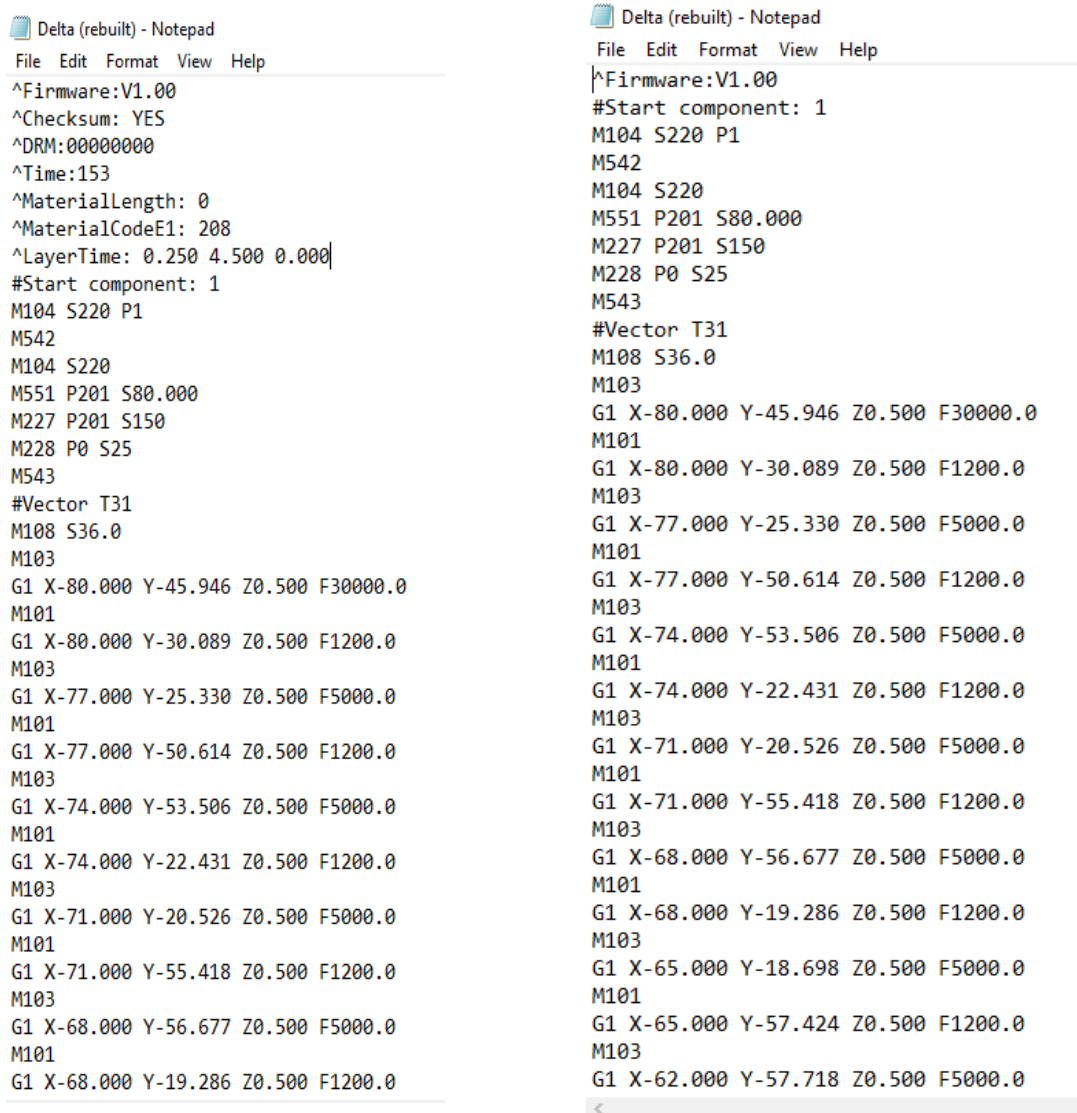
Στη συνέχεια με το πρόγραμμα «*CubeX\_3D System*» (ανάλογα το είδος και την εταιρεία του εκτυπωτή) επεξεργαζόμαστε το αντικείμενο για πιθανές ατέλειες (βλ. εικόνα 81).



Εικόνα 81: Επεξεργασία αντικειμένου πριν την εκτύπωση.

#### 4. Επεξεργασία μοντέλου με χρήση του προγράμματος «CubeX\_3D System».

Για να γίνει η επεξεργασία ανοίγουμε το αρχείο με τη μορφή *txt*, ώστε να επεξεργαστούμε τον κώδικα. Θα εκτυπώσουμε με χρήση μιας κεφαλής του 3D εκτυπωτή, (οι πρώτες έξι γραμμές δεν μας χρειάζονται). Με τον τρόπο αυτό μπορεί να τοποθετηθεί υλικό εντός του αρχικού τύπου που συνοδεύει τον τρισδιάστατο εκτυπωτή.

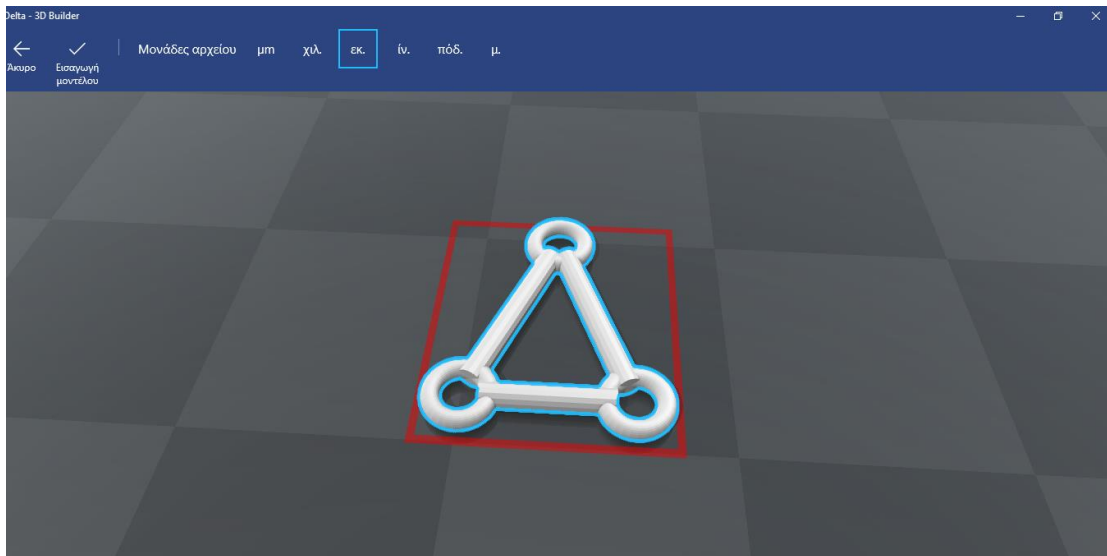


```
Delta (rebuilt) - Notepad
File Edit Format View Help
^Firmware:V1.00
^Checksum: YES
^DRM:00000000
^Time:153
^MaterialLength: 0
^MaterialCodeE1: 208
^LayerTime: 0.250 4.500 0.000
#Start component: 1
M104 S220 P1
M542
M104 S220
M551 P201 S80.000
M227 P201 S150
M228 P0 S25
M543
#Vector T31
M108 S36.0
M103
G1 X-80.000 Y-45.946 Z0.500 F30000.0
M101
G1 X-80.000 Y-30.089 Z0.500 F1200.0
M103
G1 X-77.000 Y-25.330 Z0.500 F5000.0
M101
G1 X-77.000 Y-50.614 Z0.500 F1200.0
M103
G1 X-74.000 Y-53.506 Z0.500 F5000.0
M101
G1 X-74.000 Y-22.431 Z0.500 F1200.0
M103
G1 X-71.000 Y-20.526 Z0.500 F5000.0
M101
G1 X-71.000 Y-55.418 Z0.500 F1200.0
M103
G1 X-68.000 Y-56.677 Z0.500 F5000.0
M101
G1 X-68.000 Y-19.286 Z0.500 F1200.0
M103
G1 X-65.000 Y-18.698 Z0.500 F5000.0
M101
G1 X-65.000 Y-57.424 Z0.500 F1200.0
M103
G1 X-62.000 Y-57.718 Z0.500 F5000.0
<
```

```
Delta (rebuilt) - Notepad
File Edit Format View Help
^Firmware:V1.00
#Start component: 1
M104 S220 P1
M542
M104 S220
M551 P201 S80.000
M227 P201 S150
M228 P0 S25
M543
#Vector T31
M108 S36.0
M103
G1 X-80.000 Y-45.946 Z0.500 F30000.0
M101
G1 X-80.000 Y-30.089 Z0.500 F1200.0
M103
G1 X-77.000 Y-25.330 Z0.500 F5000.0
M101
G1 X-77.000 Y-50.614 Z0.500 F1200.0
M103
G1 X-74.000 Y-53.506 Z0.500 F5000.0
M101
G1 X-74.000 Y-22.431 Z0.500 F1200.0
M103
G1 X-71.000 Y-20.526 Z0.500 F5000.0
M101
G1 X-71.000 Y-55.418 Z0.500 F1200.0
M103
G1 X-68.000 Y-56.677 Z0.500 F5000.0
M101
G1 X-68.000 Y-19.286 Z0.500 F1200.0
M103
G1 X-65.000 Y-18.698 Z0.500 F5000.0
M101
G1 X-65.000 Y-57.424 Z0.500 F1200.0
M103
G1 X-62.000 Y-57.718 Z0.500 F5000.0
<
```

Εικόνα 82: Κώδικας αντικειμένου. Αριστερά διακρίνουμε τον κώδικα πριν την επεξεργασία του, ενώ στη δεξιά εικόνα διακρίνουμε τον κώδικα μετά την επεξεργασία του.

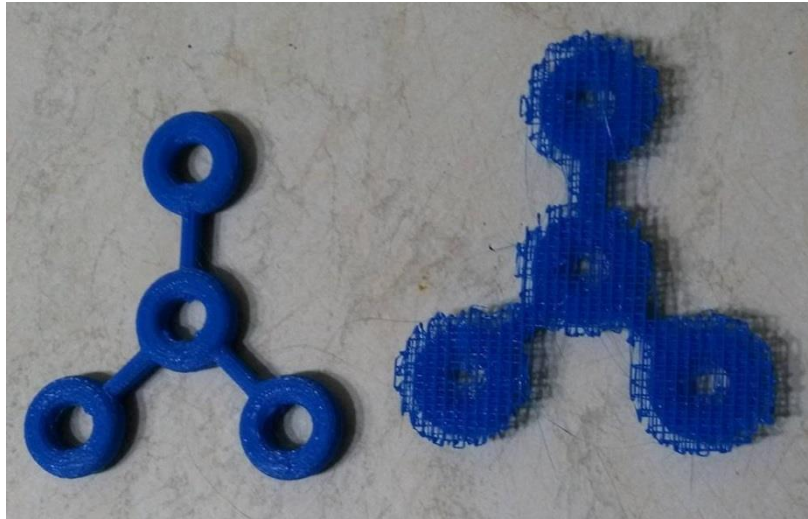
Με χρήση του προγράμματος *CubeX* επιλέγουμε την εντολή «*Build*» για τις τελευταίες λεπτομέρειες πριν την εκτύπωση. Επιλέγουμε την ποιότητα της εκτύπωσης, το πάχος του αντικειμένου καθώς και το είδος του υλικού που θα χρησιμοποιήσουμε. Τέλος, επιλέγουμε την εντολή «*Build*» και το πρόγραμμα μας υπολογίζει το χρόνο εκτύπωσης (Στη συγκεκριμένη περίπτωση χρειάστηκε 1 ώρα και 20 λεπτά). Στη παρακάτω εικόνα διακρίνουμε τη μορφή του μοντέλου όπως ακριβώς θα εκτυπωθεί από τον τρισδιάστατο εκτυπωτή (βλ. εικόνα 83).



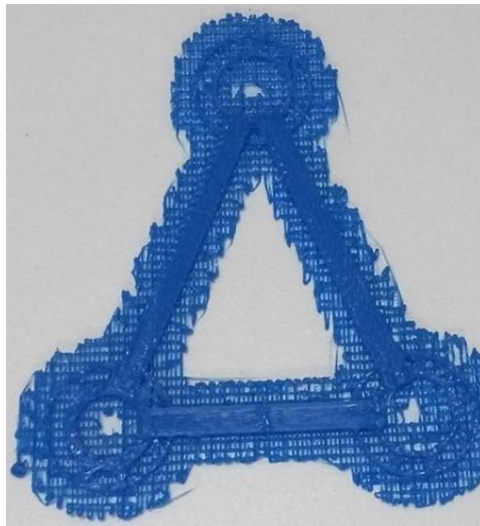
Εικόνα 83: Μορφή αντικειμένου που θα εκτυπωθεί.

##### 5. Κατασκευή και καθαρισμός - τελικό φινίρισμα του μοντέλου.

Το τελικό αρχείο μεταφέρεται στον 3D εκτυπωτή μέσω μονάδας USB για κατασκευή. Μόλις το αντικείμενο είναι έτοιμο, το αφαιρούμε προσεκτικά από την πλατφόρμα του εκτυπωτή με νερό και κατάλληλη σπάτουλα αφαιρούμε προσεκτικά. Το υλικό είναι έτοιμο για καθάρισμα.



Εικόνα 84: Αντικείμενο πριν και μετά την αφαίρεση του περιττού υλικού.



Εικόνα 85: Δομικό υλικό.



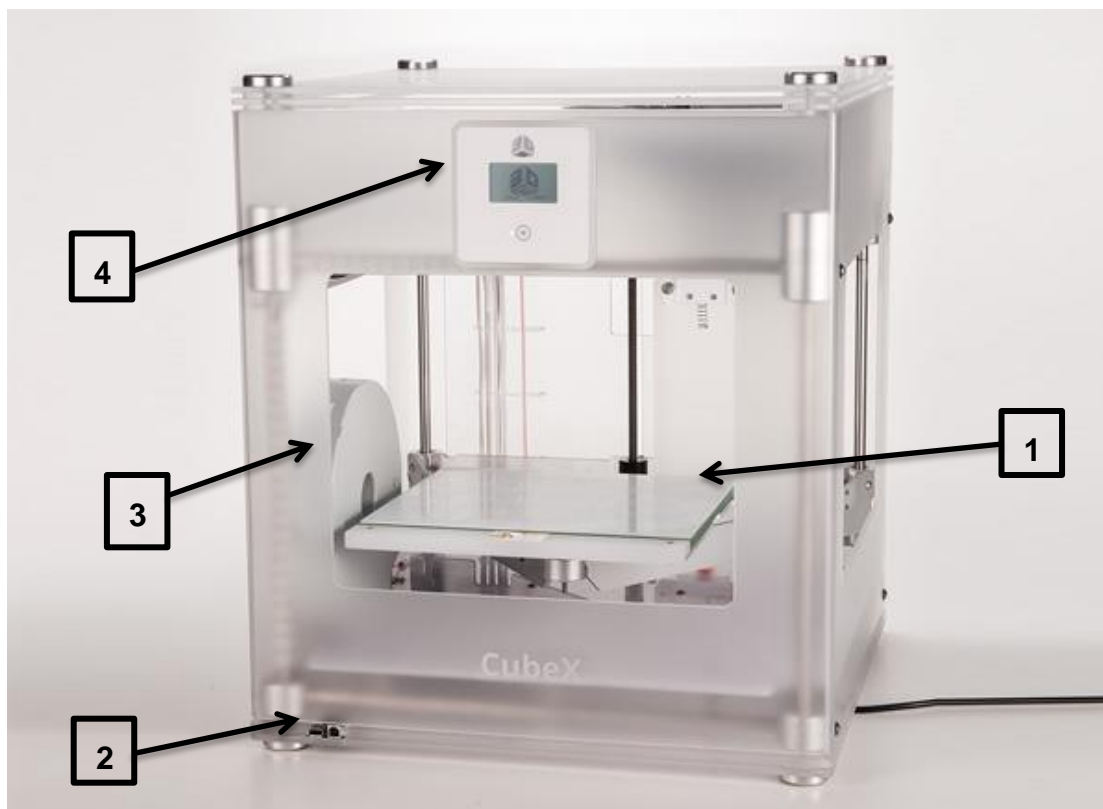


Εικόνα 86: Μέση τομή αντικειμένου. Η εκτύπωση διακόπηκε στο μισό χρόνο.

Το πλέγμα το οποίο αποτελείται από μια λεπτή στρώση υλικού (βλ. εικόνα 85), δημιουργείται με σκοπό στη συνέχεια το υλικό το οποίο θα σχηματίσει το αντικείμενο να μπορεί να κολλήσει πάνω σε μια σταθερή βάση, ώστε να μην μετακινείται και να πάρει τη μορφή που επιθυμούμε.

### 7.1 Προετοιμασία του τρισδιάστατου εκτυπωτή για την εκτύπωση

Η προετοιμασία του εκτυπωτή περιλαμβάνει καταρχάς το γέμισμά του με κατάλληλη πρώτη ύλη το πλαστικό (*PLA*). Γενικότερα, πρέπει να διαλέξουμε το υλικό με το οποίο θα επιτύχουμε καλύτερα τις συγκεκριμένες ιδιότητες που απαιτούνται για το αντικείμενο που θέλουμε να παράγουμε. Επιπλέον, απαιτείται προετοιμασία της πλατφόρμας κατασκευής. Σε ορισμένες περιπτώσεις, ίσως χρειαστεί να καθαριστεί ή να εφαρμοστεί μια κόλλα για να αποτραπεί η μετακίνηση και στρέβλωση του αντικειμένου από τη θερμότητα κατά τη διάρκεια της διαδικασίας εκτύπωσης.

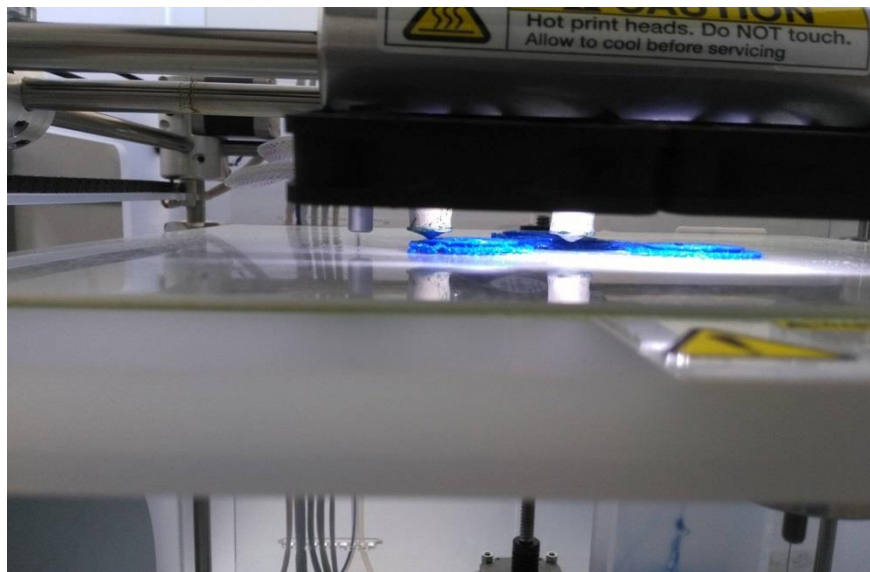


Εικόνα 87: 3D printer (<https://www.consumerreports.org>).

Στη παραπάνω εικόνα διακρίνουμε (βλ. εικόνα 87):

1. Πλατφόρμα εκτύπωσης.
2. Θύρα *usb* για φόρτωση ψηφιακού μοντέλου (χειροκίνητη λειτουργία).
3. Προστατευτική θήκη πρώτης ύλης (Πολυμερές σε μορφή νήματος).
4. Οθόνη ελέγχου του εκτυπωτή.

Μόλις εισαχθεί το ψηφιακό μοντέλο στον εκτυπωτή, το μηχάνημα αναλαμβάνει αυτόματα τη δημιουργία του επιθυμητού αντικειμένου. Το υλικό εκτύπωσης θερμαίνεται μέχρι τη στιγμή που υγροποιείται και εξωθείται μέσω του ακροφυσίου. Χρησιμοποιώντας τις πληροφορίες από το ψηφιακό αρχείο, το μοντέλο εκτυπώνεται σε λεπτές διαστάσεις διατομές, τοποθετώντας το πλαστικό υλικό (πολυμερές) μέσω του ακροφυσίου σε λεπτές στρώσεις κατάλληλα. Το πολυμερές στερεοποιείται γρήγορα και ενσωματώνεται με το κάτω στρώμα του υλικού, πριν η πλατφόρμα χαμηλώσει και η κεφαλή εκτύπωσης προσθέσει καινούριο στρώμα. Ανάλογα με το μέγεθος και την πολυπλοκότητα του αντικειμένου, η όλη διαδικασία μπορεί να διαρκέσει από λεπτά έως ημέρες.



Εικόνα 88: Κατά τη διάρκεια της εκτύπωσης του αντικειμένου.

Αφού ολοκληρωθεί η εκτύπωση, κάθε αντικείμενο απαιτεί μία ελάχιστη επεξεργασία. Αυτή μπορεί να είναι η αποκόλληση του αντικειμένου από την πλατφόρμα εκτύπωσης ή το φινίρισμα, καθώς και η αφαίρεση των «δομών στήριξης» (προσωρινό υλικό που τυπώνεται για τη στήριξη προεξοχών επί του αντικειμένου) από το αντικείμενο.

## Κεφάλαιο 8<sup>ο</sup>: Υλικά και κόστος παραγωγής εκπαιδευτικών μονάδων

Σε αυτό το κεφάλαιο γίνεται αναφορά στο είδος και την ποσότητα των απαραίτητων υλικών για την κατασκευή των έξι εκπαιδευτικών μονάδων για το εργαστήριο «Ναυτικών μηχανών και μηχανολογίας» του τμήματος «Ναυπηγικής» του Τ.Ε.Ι. Αθήνας, καθώς και το συνολικό κόστος κατασκευής των εκπαιδευτικών μονάδων.

Παρακάτω, (βλ. εικόνα 89) μπορούμε να διακρίνουμε με μια πρώτη ματιά τα υλικά που χρησιμοποιήσαμε, καθώς και την ποσότητά τους για την κατασκευή μιας μονάδας.

Για μια μονάδα εκμάθησης

Είδος	Ποσότητα	Κόστος ανά μονάδα (χωρίς ΦΠΑ)	ΦΠΑ	Κόστος ανά μονάδα (Με ΦΠΑ)	Κόστος
Βάση Ράγας 1P RT624-B (Relay Mini)	6	1,94	24%	2,4056	14,4336
Ρελέ Μίνι 1P 24V/12A RT114024	6	2,2	24%	2,728	16,368
Ράγα Ωμέγα Αλουμινίου	1	4,36	24%	5,4064	5,4064
Κουτί Ηλεκτρολογικό με κλειστό καπάκι (226 x 175 x 80)	1	12,75	24%	15,81	15,81
Καλώδιο NYA 1x1 mm (Για την συνδεσμολογία)	65	0,23	24%	0,2852	18,538
Καλώδιο NYA 1x2.5 mm (Για τις μπανανες)	2	0,39	24%	0,4836	0,9672
Μπρόντα 4mm 16A (RC170-SW)	79	0,23	24%	0,2852	22,5308
Μπανάνα κοκ/μαύρη με υποδοχή	40	0,45	24%	0,558	22,32
Διακόπτης Toggle mini on-off 3A/250V 2P	7	0,5	24%	0,62	4,34
Κουμπί (Push Button) 2P	7	0,57	24%	0,7068	4,9476
Βάση LED Μονή 10mm	6	0,27	24%	0,3348	2,0088
Φωτοδίοδος 10mm	6	0,2	24%	0,248	1,488
Βάση LED Μονή 8mm	6	0,27	24%	0,3348	2,0088
Φωτοδίοδος 8mm	6	0,2	24%	0,248	1,488
Βάση LED Μονή 6mm	7	0,3	24%	0,372	2,604
Φωτοδίοδος 6mm	7	0,2	24%	0,248	1,736
Αντίσταση 1/2W 1Ω-10MΩ 5%	19	0,98	24%	1,2152	23,0888
Κλέμα ηλεκτρολογική 10A	10	0,82	24%	1,0168	10,168
Γυμνοί ακροδέκτες-κις	85	0,1	24%	0,124	10,54
Plexy_glass (Πάχος 5mm)	1	35	24%	43,4	43,4
Ξύλο κόντρα πλακέ (Πάχος 9mm)	1	21	24%	26,04	26,04
Ξύλο κόντρα πλακέ (Alfacom T)	6	5,37	24%	6,6588	39,9528
Ξύλινη ραβδος	1	0,27	24%	0,3348	0,3348
Θερμοσυστολικά 12mm	3	1,33	24%	1,6492	4,9476
Θερμοσυστολικά 8mm	6	0,32	24%	0,3968	2,3808
Μακετόχαρτο	1	4,02	24%	4,9848	4,9848
PLC kit	1	716	24%	887,84	887,84
ΣΥΝΟΛΟ					1190,6728

Εικόνα 89: Υλικά και κόστος για την κατασκευή μιας εκπαιδευτικής μονάδας.

Παρατηρούμε ότι το κόστος κατασκευής μιας μονάδας εκμάθησης ανέρχεται στα 1.190,67€. Επομένως το συνολικό κόστος κατασκευής και των έξι εκπαιδευτικών μονάδων ανέρχεται στα 7.144,04€. Να σημειωθεί ότι οι έξι μονάδες PLC προμηθεύονται από την εταιρία Siemens Hellas με σημαντική μείωση κόστους για καθαρά εκπαιδευτικό σκοπό. Η τιμή που αντιστοιχεί στη κάθε μια ήταν περίπου 273€, διαμορφώνοντας το τελικό κόστος ανά μονάδα στα 575,8€.

## 8.1 Μονάδες εκμάθησης PLC της εταιρίας GTEE και του ΠΑ.Δ.Α

Η *GTEE* (*Guangzhou Guangtong Educational Equipment*), είναι μια εταιρία κατασκευής εργαστηριακών μονάδων όλων των ειδών. Η συγκεκριμένη εταιρία έχει κατασκευάσει μονάδες εκμάθησης *PLC*. Η παρακάτω εικόνα (βλ. εικόνα 90) είναι από μια σχετικά όμοια μονάδα εκμάθησης με αυτή που κατασκευάστηκε για το ΠΑ.Δ.Α.



Εικόνα 90: Μονάδα εκμάθησης PLC (<http://www.gteecn.com>).

Η παραπάνω μονάδα περιλαμβάνει τα ακόλουθα:

- Τροφοδοτικό AC 220V.
- *PLC* (S7-200) της εταιρίας *Siemens*. Το συγκεκριμένο *PLC* αποτελείται από 14 θύρες εισόδου και 10 θύρες εξόδου.
- Το λογισμικό «*STEP 7 MicroWIN*».
- 20 καλώδια συνδεσμολογίας (τύπου μπανάνα).

Οι εφαρμογές οι οποίες μπορούν να εφαρμοστούν στη παρούσα μονάδα είναι:

- Εξοικείωση με το περιβάλλον εργασίας «*STEP 7 MicroWIN*» και τη γλώσσα προγραμματισμού *Ladder*.
- Έλεγχος λειτουργίας κινητήρα.
- Εκκίνηση κινητήρα (Συνδεσμολογία Υ-Δ).
- Έλεγχος λειτουργίας φωτεινών σηματοδοτών στους δρόμους.
- Έλεγχος ανάμειξης υγρών.
- Έλεγχος απόκρισης.

Σύμφωνα με τα παραπάνω παρατηρούμε ότι και οι δύο μονάδες έχουν αρκετές ομοιότητες, όπως και διαφορές. Πιο συγκεκριμένα, οι ομοιότητες που εμφανίζουν οι μονάδες εκμάθησης του ΠΑ.Δ.Α και της *GTEE* είναι:

- Οι εφαρμογές που μπορούν να εκτελέσουν μοιάζουν αρκετά.
- Είναι μικρές, ελαφριές και αποθηκεύονται εύκολα.
- Οι Προγραμματιζόμενοι Λογικοί Ελεγκτές είναι της εταιρίας *Siemens* και αποτελούνται από τον ίδιο αριθμό εισόδων και εξόδων.
- Οι συνδέσεις γίνονται όλες στο εσωτερικό της μονάδας ενώ ο χρήστης/φοιτητής με χρήση των καλωδίων τύπου μπανάνα κάνει τις εξωτερικές διασυνδέσεις για τη λειτουργία του συστήματος.

Ενώ οι διαφορές που εμφανίζουν οι μονάδες του Πανεπιστημίου Δυτικής Αττικής και της *GTEE* είναι:

- Στο μοντέλο του *PLC* το οποίο χρησιμοποιούν (*S7-200* της μονάδας της *GTEE* και *S-1200* του Πανεπιστημίου Δυτικής Αττικής).

- Στην δυσκολία επισκευής. Η μονάδα της *GTEE* δυσκολεύει τον εντοπισμό και την επισκευή τυχών βλάβης.
- Στη δυσκολία κατανόησης της εσωτερικής συνδεσμολογίας της μονάδας. Η μονάδα του ΠΑ.Δ.Α έχει κατασκευαστεί από *plexiglass* με σκοπό να επιτρέπει στο φοιτητή να βλέπει και να κατανοεί την εσωτερική συνδεσμολογία της μονάδας. Ενώ της *GTEE* από υλικό το οποίο δεν είναι διάφανο.
- Το κόστος. Η μονάδα του ΠΑ.Δ.Α είναι πολύ πιο οικονομική σε σχέση με τη μονάδα της *GTEE*. Το κόστος της εκπαιδευτικής μονάδας της *GTEE* ανέρχεται στα 1.229,23€, ενώ του ΠΑ.Δ.Α στα 575,8€.
- Προστασία μονάδας *PLC*. Η μονάδα του Πανεπιστημίου Δυτικής Αττικής έχει κατασκευαστεί με τέτοιο τρόπο ώστε να προστατεύεται η μονάδα ΠΛΕ ενώ παράλληλα να είναι ορατή στους φοιτητές (ειδικά διαμορφωμένα καπάκια ηλεκτρολογικού κουτιού). Επιπλέον οι κλέμες έχουν διαμορφωθεί έτσι ώστε να αφαιρείται όλο το ηλεκτρολογικό κουτί, με το περιεχόμενό του, χωρίς να «λυθεί» όλη η μονάδα.

Παρατηρούμε ότι έχουν αρκετές διαφορές, με κύριο χαρακτηριστικό το κόστος τους. Επομένως, η μονάδα εκμάθησης του Πανεπιστημίου Δυτικής Αττικής πέρα από το στόχο μας να είναι λειτουργική και εύχρηστη, παρατηρούμε ότι είναι και αρκετά οικονομική σε σχέση με άλλες μονάδες της αγοράς, οι οποίες είναι σχετικά όμοιες με του ΠΑ.Δ.Α.

## Κεφάλαιο 9<sup>ο</sup>: Αισθητήρες (Sensors) και Ηλεκτρονόμος (Relay)

Σε αυτό το κεφάλαιο γίνεται αναφορά στη λειτουργία και τη χρήση τόσο ενός ηλεκτρονόμου όσο και συγκεκριμένων αισθητήρων (ροής, στάθμης και στροφών), τα οποία χρησιμοποιήθηκαν στα πλαίσια της παρούσας διπλωματικής εργασίας για την πραγματοποίηση των εργαστηριακών ασκήσεων.

### 9.1 Εισαγωγή στους ηλεκτρονόμους

Ο ηλεκτρονόμος, γνωστός και ως ρελέ (*relay*), είναι ένας ηλεκτρικός διακόπτης που συνδέει ή αποσυνδέει μία ή περισσότερες ηλεκτρικές επαφές υπό τον έλεγχο ενός κατάλληλου ηλεκτρικού κυκλώματος ελέγχου. Εξαιτίας του γεγονότος ότι ένας ηλεκτρονόμος είναι ικανός να ελέγχει ένα κύκλωμα εξόδου μεγαλύτερης ισχύος από το κύκλωμα εισόδου, μπορεί να θεωρηθεί ως μία μορφή ηλεκτρικού ενισχυτή.

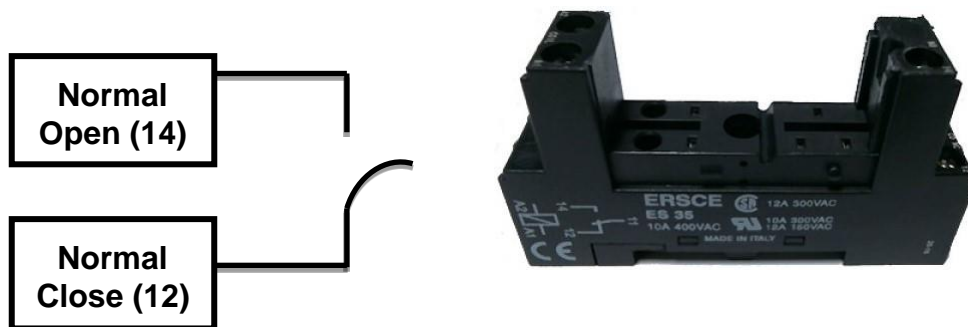


Εικόνα 91: Relay 24VDC (<https://www.ebay.com>).

Οι ηλεκτρονόμοι αποτελούνται από δύο βασικά μέρη. Το ένα περιλαμβάνει τις διάφορες διακοπτικές επαφές και το άλλο το μηχανισμό δημιουργίας της κίνησης που είναι απαραίτητη για την αλλαγή κατάστασης των επαφών. Οι διακοπτικές επαφές που συναντώνται στους ηλεκτρονόμους είναι τριών τύπων :

1. Κανονικά ανοικτή επαφή (*Normally Open – NO*): Κλείνει όταν ο ηλεκτρονόμος ενεργοποιηθεί, επιτρέποντας έτσι τη διέλευση του ρεύματος στο κύκλωμα που ελέγχεται από τον ηλεκτρονόμο.
2. Κανονικά κλειστή επαφή (*Normally Closed – NC*): Ανοίγει όταν ο ηλεκτρονόμος ενεργοποιηθεί, διακόπτοντας έτσι τη διέλευση του ρεύματος στο κύκλωμα που ελέγχεται από τον ηλεκτρονόμο.
3. Μεταγωγική επαφή (*Change Over – CO*): Ελέγχει εκ περιτροπής δύο κυκλώματα.





Εικόνα 92: Διακοπτικές επαφές ρελέ.

Το δεύτερο τμήμα ενός ηλεκτρονόμου που αφορά το μηχανισμό δημιουργίας της κίνησης των διατοπικών επαφών, αποτελείται από ένα πηνίο, ένα κινητό σπλισμό και ένα ελατήριο επαναφοράς.



Εικόνα 93: Μια επαφή του ηλεκτρονόμου. Το μεσαίο είναι το κινητό μέρος της επαφής (<https://greekelectrician.blogspot.gr>).

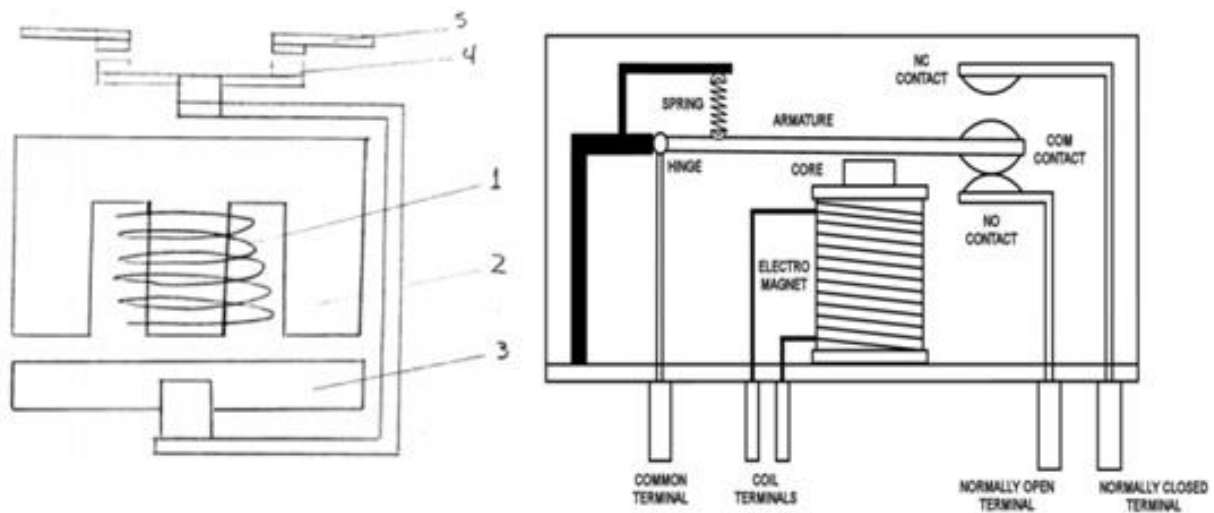
Τα βασικά στοιχεία τα οποία χαρακτηρίζουν ένα ηλεκτρονόμο είναι τα παρακάτω:

1. Το ονομαστικό ρεύμα λειτουργίας: Είναι σημαντικό να γνωρίζουμε το ρεύμα το οποίο μπορεί να διέρχεται το ρελέ, μέσω των κυρίων επαφών του σε μόνιμη λειτουργία.
2. Η τάση λειτουργίας του πηνίου του. Το πηνίο μπορεί να λειτουργεί σε διάφορες τάσεις.
3. Ο αριθμός και το είδος των βοηθητικών επαφών.
4. Η συχνότητα των χειρισμών ανά ώρα που μπορεί να δεχθεί ο ηλεκτρονόμος.

Πιο συγκεκριμένα ο ηλεκτρονόμος αποτελείται από (βλ. εικόνα 94):

1. Το πηνίο: Όταν βρεθεί υπό τάση, δημιουργεί γύρω του μαγνητικό πεδίο. Η τάση λειτουργίας του πηνίου είναι ανεξάρτητη από την τάση λειτουργίας του κυκλώματος ισχύος.
2. Ο μαγνήτης: Μετατρέπει την ηλεκτρική ενέργεια σε μηχανική, έλκοντας προς αυτόν, τον οπλισμό του ηλεκτρονόμου.
3. Ο οπλισμός: Αποτελεί το κινητό μέρος του ηλεκτρονόμου, το οποίο κάτω από την επίδραση του ηλεκτρομαγνήτη έλκεται προς το μέρος του. Αυτή η κίνηση του οπλισμού μεταφέρεται με κατάλληλη μηχανική σύνδεση στις κινητές επαφές του ηλεκτρονόμου.
4. Κινητές επαφές: Σκοπό να δημιουργήσει αγωγή μεταξύ των σταθερών επαφών του ηλεκτρονόμου.
5. Σταθερές ή κύριες επαφές: Στις σταθερές ή κύριες επαφές των ηλεκτρονόμων συνδέονται οι αγωγοί τροφοδοσίας των φορτίων. Ένα φορτίο το οποίο τροφοδοτείται μέσω ενός ηλεκτρονόμου, είναι σε κατάσταση ηρεμίας όταν ο ηλεκτρονόμος είναι σε κατάσταση ηρεμίας, όταν το πηνίο δεν βρίσκεται υπό τάση, γιατί δεν υπάρχει αγωγή μεταξύ των σταθερών επαφών Α και Β. Αντιθέτως, ένα φορτίο βρίσκεται σε κατάσταση λειτουργίας, όταν το πηνίο του ηλεκτρονόμου βρίσκεται υπό τάση γιατί τότε έχει αποκατασταθεί η αγωγή μεταξύ των σταθερών επαφών Α και Β.

Παρακάτω γίνεται μια σύντομη αναφορά της λειτουργίας ενός ηλεκτρονόμου δύο διακοπτικών επαφών, ώστε να μπορέσουμε να κατανοήσουμε ακόμη καλύτερα τη χρήση του (βλ. εικόνα 94).



Εικόνα 94: Λειτουργία ηλεκτρονόμου (<http://robotics.lib-ieronimoub.gr>). Διακρίνονται οι επαφές ελέγχου (Coil Terminals) και οι δύο «διακόπτες» που ενεργοποιούνται/απενεργοποιούνται (Common Terminal-Normally Closed Terminal & Common Terminal-Normally Open Terminal).

Για να λειτουργήσει ένας ηλεκτρονόμος (βλ. εικόνα 94), πρέπει το πηνίο (1) του να βρεθεί υπό τάση. Τότε δημιουργείται γύρω από αυτό μαγνητικό πεδίο. Η σιδερένια μάζα επί της οποίας έχει τοποθετηθεί το πηνίο, τοποθετημένη εντός του μαγνητικού πεδίου, γίνεται μαγνήτης (2) και αυτό έχει σαν αποτέλεσμα την έλξη του σπλισμού (3) του ηλεκτρονόμου προς τον μαγνήτη. Αυτή η κίνηση του σπλισμού μεταφέρεται με κατάλληλη μηχανική σύνδεση στις κινητές επαφές (4) του ηλεκτρονόμου, η κίνηση των οποίων έχει σαν αποτέλεσμα την αγωγή σύνδεση των σταθερών ή κύριων επαφών Α και Β (5) μέσω των οποίων τροφοδοτείται το ελεγχόμενο φορτίο.

## 9.2 Εισαγωγή στους αισθητήρες

Γενικά, με τον όρο αισθητήρες περιγράφονται όλες εκείνες οι συσκευές που μετρούν μια φυσική ποσότητα και τη μετατρέπουν σε ηλεκτρικό σήμα για να γίνει η μέτρησή της. Τα κυριότερα είδη αισθητήρων που μπορούν να συνεργαστούν με ένα Η/Υ, ανάλογα με το φυσικό μέγεθος που μετρούν είναι συνήθως αισθητήρες θερμοκρασίας, φωτεινής στάθμης, πίεσης και βάρους, στάθμης, όγκου, μετατόπισης και κίνησης. Οι αισθητήρες είναι πολύ σημαντικοί στις μέρες μας. Όσο αναπτύσσεται η τεχνολογία και οι απαιτήσεις για ακρίβεια στον εργασιακό χώρο για την ανάπτυξη της παραγωγής, τόσο πιο συχνά χρησιμοποιούνται οι αισθητήρες για τον έλεγχο των συστημάτων, εξασφαλίζοντας έτσι ασφάλεια στον εργασιακό χώρο (π.χ: αισθητήρες ανίχνευσης πυρκαγιάς ή διαρροής αερίου), δυνατότητα αποφυγής βλάβης στο σύστημα (π.χ: έλεγχος στροφών κινητήρα) καθώς και ευκολία στον έλεγχο του συστήματος (π.χ: Πλέον στα πλοία δεν χρειάζεται να πηγαίνουν οι μηχανικοί συχνά στο μηχανοστάσιο και με συγκεκριμένα αισθητήρια μέσα να παίρνουν μετρήσεις για τον έλεγχο της μηχανής και της πίεσης στις σωληνώσεις. Μέσω αισθητήρων όλα τα σήματα μαζεύονται σε έναν κεντρικό υπολογιστή και ελέγχονται αρχικά από εκεί).



Εικόνα 95: Διάφορα είδη αισθητήρων (<http://blog.misumiusa.com>).

## 9.2.1 Χαρακτηριστικά Αισθητήρων

Για την επιλογή του κατάλληλου οργάνου για μια συγκεκριμένη εφαρμογή, σημασία έχει η γνώση των χαρακτηριστικών του αισθητήρα που αποτυπώνουν την απόδοση και την συμπεριφορά του, κατά τη διάρκεια των μετρήσεων. Τα σημαντικότερα από αυτά τα χαρακτηριστικά είναι:

1. Συμβατότητα: Περιγράφει κατά πόσον η εγκατάσταση του οργάνου θα επηρεάσει την τιμή της παραμέτρου που πρόκειται να μετρήσει. Ιδανικό θεωρείται ένα όργανο που δεν την επηρεάζει καθόλου.
2. Εύρος λειτουργίας: Ορίζεται από τα όρια εντός των οποίων έχει υπολογιστεί ότι μπορεί να λειτουργεί αξιόπιστα. Συνήθως, εκφράζεται με την ελάχιστη και τη μέγιστη τιμή που μπορεί να μετρήσει.
3. Ακρίβεια: Καθορίζεται από το μέγιστο σφάλμα που μπορεί να περιέχεται στην ένδειξή του. Γενικά, όλες οι συσκευές παράγουν σφάλμα στις μετρήσεις τους και το ζητούμενο είναι αυτό το σφάλμα να έχουμε το μικρότερο δυνατό σφάλμα.
4. Επαναληψιμότητα: Ονομάζεται ο βαθμός στον οποίο μια συσκευή παρέχει το ίδιο αποτέλεσμα τροφοδοτούμενη με την ίδια είσοδο σε διαφορετικές χρονικές στιγμές.
5. Διακριτική ικανότητα: Ορίζεται ως η μικρότερη μεταβολή στη διέγερση του σήματος η οποία μπορεί να γίνει αντιληπτή από το αισθητήριο.
6. Ευαισθησία: Είναι η ελάχιστη μεταβολή της εισόδου του που είναι σε θέση να δώσει μεταβολή στην έξοδό του.
7. Υστέρηση: Η υστέρηση προκαλεί διαφορές στην έξοδο ενός αισθητήρα όταν η κατεύθυνση μεταβολής της εισόδου αντιστραφεί, με αποτέλεσμα να παράγεται σφάλμα και επηρεάζεται η ακρίβεια της συσκευής.
8. Εξωτερικές διαταραχές: Δημιουργούνται κατά τη διάρκεια μιας μέτρησης από εξωτερικούς παράγοντες, όπως ο θόρυβος. Οι αισθητήρες οι οποίοι δεν επηρεάζονται από εξωτερική διέγερση, όπως ο θόρυβος, είναι πιο αξιόπιστοι για την ύπαρξη ενός σταθερού συστήματος.
9. Διαστάσεις: Οι διαστάσεις ενός αισθητήρα αναφέρονται στο μέγεθός του. Σημαντικός παράγοντας όταν δεν υπάρχει ευχέρεια στο χώρο.
10. Γραμμικότητα: Ονομάζεται ο βαθμός στον οποίο η γραφική παράσταση της εξόδου ως προς την είσοδο του αισθητήρα προσεγγίζει μια ευθεία γραμμή. Ένας αισθητήρας μπορεί να είναι γραμμικός για μια περιοχή τιμών.

11. Κόστος: Δεν πληρούνται όλα τα παραπάνω χαρακτηριστικά από όλους τους αισθητήρες. Εκείνοι που πληρούν τα περισσότερα έχουν και υψηλό κόστος σε σχέση με κάποιους άλλους με λιγότερες δυνατότητες.

### 9.2.2 Συνδεσμολογία αισθητήρων

Οι δύο αισθητήρες που θα μας απασχολήσουν στη παρούσα εργασία, αποτελούν αισθητήρες τύπου *PNP/NPN* (επαγωγικοί αισθητήρες). Οι αισθητήρες αυτοί αποτελούνται από τρία καλώδια τα οποία μπορούν να συνδεθούν με δύο διαφορετικούς τρόπους, κάθε ένας από τους οποίους απευθύνεται στους τύπους *PNP* (στο *PNP*, η βάση πρέπει να είναι 0.6V λιγότερο από το θετικό το ρεύμα. Δηλαδή χρησιμοποιούμε μια αντίσταση στο καλώδιο που συνδέουμε τον αισθητήρα με τη τροφοδοσία) και *NPN* (στο *NPN*, η βάση πρέπει να είναι 0.6V περισσότερα από το αρνητικό ρεύμα. Δηλαδή χρησιμοποιούμε μια αντίσταση στο καλώδιο που συνδέουμε τον αισθητήρα με την επιστροφή) αντίστοιχα. Συγκριμένα τα δύο εκ των τριών καλωδίων κάθε αισθητήρα συνδέθηκαν με το θετικό ρεύμα (+) και το αρνητικό ρεύμα (-) ενώ το τελευταίο, καταλήγει στην επιθυμητή είσοδο του *PLC*. Οι αισθητήρες που χρησιμοποιήθηκαν απεικονίζονται παρακάτω.



Εικόνα 96: Αισθητήρας μέτρησης στροφών (αριστερή εικόνα) και αισθητήρας ροής (δεξιά εικόνα).

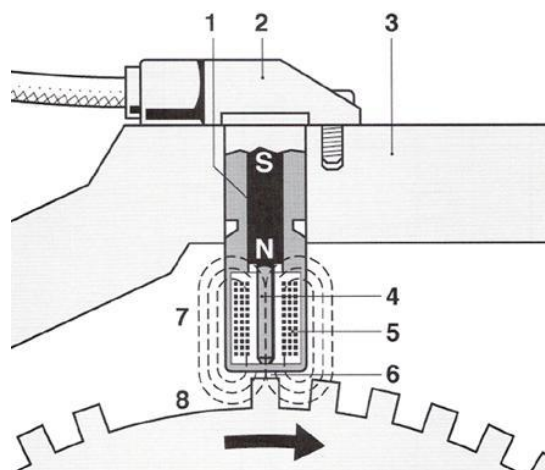
### 9.2.3 Αισθητήρες στροφών

Η ακριβής μέτρηση στροφών στους σύγχρονους κινητήρες είναι σημαντική για τη λειτουργία του ηλεκτρονικού συστήματος ανάφλεξης και ψεκασμού καυσίμου σε αυτούς. Για τον σκοπό αυτό, χρησιμοποιούνται αισθητήρες στροφών που μετράνε τις στροφές του κινητήρα, στέλνοντας κατάλληλο ηλεκτρικό σήμα στον υπολογιστή, όπου συγκεντρώνονται οι μετρήσεις.

Οι αισθητήρες αυτοί είναι επαγωγικού τύπου και μετράνε τις στροφές του στροφαλοφόρου, ενώ ταυτόχρονα ανιχνεύουν και τη χρονική στιγμή που το έμβολο του 1<sup>ου</sup> κυλίνδρου φτάνει στο άνω νεκρό σημείο (ΑΝΣ), δηλαδή εκτελεί διπλή λειτουργία..

Λειτουργία και κατασκευή αισθητήρα:

Αυτού του τύπου οι αισθητήρες παράγουν ένα σήμα που οφείλεται στην μεταβολή του μαγνητικού πεδίου μπροστά από τον αισθητήρα. Το μαγνητικό πεδίο παράγεται από ένα μόνιμο μαγνήτη που βρίσκεται μέσα στο σώμα του αισθητήρα, ενώ η μεταβολή της μαγνητικής ροής στον πυρήνα του, γίνεται από ένα οδοντωτό τροχό ο οποίος περιστρέφεται μπροστά από τον αισθητήρα. Από τη μεταβολή της μαγνητικής ροής παράγεται στο πηνίο μια εναλλασσόμενη τάση από επαγωγή. Οι επαγωγικοί αισθητήρες στροφών είναι τοποθετημένοι εφαιπτόμενα σε έναν οδοντωτό τροχό, ο οποίος συνήθως βρίσκεται στο βολάν του στροφαλοφόρου. Όσο αυξάνεται ο αριθμός στροφών τόσο η συχνότητα και το πλάτος της εναλλασσόμενης τάσης αυξάνονται.



1. Μόνιμος μαγνήτης.
2. Κέλυφος αισθητήρα.
3. Στήριξη του αισθητήρα.
4. Πυρήνας από μαλακό σίδηρο.
5. Πηνίο.
6. Διάκενο.
7. Μαγνητικές γραμμές του μαγνητικού πεδίου.
8. Οδοντωτός τροχός με σημείο αναφοράς (κενό δύο δοντιών).

Εικόνα 97: Λειτουργία επαγωγικού αισθητήρα στροφών (<http://13epal-esp-thess.thess.sch.gr>).

#### 9.2.4 Αισθητήρες ροής

Οι τυπικοί αισθητήρες ροής (όπως αυτός που χρησιμοποιήσαμε στη παρούσα διπλωματική) βασίζεται στη χρήση μιας κατάλληλης περωτής που κινείται λόγω του ρευστού. Δηλαδή, όταν η ροή περάσει μέσα από το αισθητήριο, η περωτή περιστρέφεται με ταχύτητα που, υπό συνθήκες, είναι ανάλογη της παροχής. Συνήθως υπάρχει κάποιο αισθητήριο ταχύτητας, όπου μετράει τις στροφές της περωτής. Τα αισθητήρια ροής πρέπει να παρεμβληθούν εν σειρά στο κύκλωμα παροχής.



Εικόνα 98: Αισθητήρας ροής με στρόβιλο (<http://www.ardumotive.com>).

### 9.2.5 Αισθητήρια στάθμης υγρού

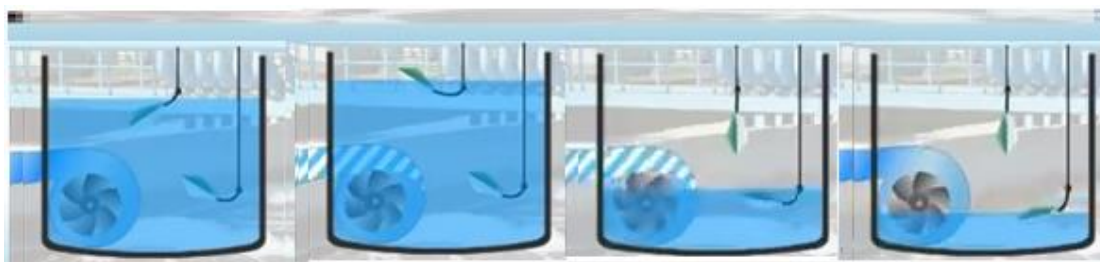
Η παρακολούθηση της στάθμης σε δεξαμενές είναι σημαντική σε πολλές εφαρμογές σε βιομηχανίες καθώς και στα πλοία. Για το σκοπό αυτό υπάρχουν διαφόρων ειδών αισθητήρια στάθμης. Τα περισσότερα απ' αυτά χρησιμοποιούν κάποιο πλωτήρα σε συνδυασμό με ένα αισθητήριο γραμμικής ή γωνιακής θέσης.

Οι αισθητήρες στάθμης σημείου/σημείων χρησιμοποιούνται γενικά για έλεγχο υψηλής/χαμηλής στάθμης, ελάχιστου και μέγιστου ύψους στάθμης ή για ενεργοποίηση συναγερμού. Τέτοιοι αισθητήρες αποτελούνται κυρίως από κινούμενα μέρη (φλοτέρ). Όταν η στάθμη του ρευστού υπερβεί το όριο της δεξαμενής (υπερπλήρωση), ο αισθητήρας, λόγω του κινούμενου μέρους του, ανεβαίνει μέχρι ένα σημείο ενεργοποιώντας έτσι την αντλία της δεξαμενής, με σκοπό τη μείωση της στάθμης του ρευστού. Εάν η στάθμη της δεξαμενής είναι κάτω από το επιθυμητό όριο, τότε ένας άλλος αισθητήρας (μπορεί να είναι ίδιου τύπου με τον προηγούμενο) κινείται προς τα κάτω μέχρι ένα σημείο, απενεργοποιώντας έτσι την αντλία της δεξαμενής (βλ. εικόνα 100). Οι αισθητήρες που χρησιμοποιούμε για συνεχή μέτρηση δεν αποτελούνται από κινούμενα μέρη και δίνουν μια αναλογική έξοδο που αντιστοιχεί με το περιεχόμενο της δεξαμενής.



Εικόνα 99: Αισθητήρας στάθμης ρευστού (Φλοτέρ) (<https://ilektroytomatismoi.blogspot.gr>).





Εικόνα 100: Λειτουργία αισθητήρα στάθμης ρευστού με κινούμενα μέρη (<https://www.youtube.com>).

## Κεφάλαιο 10<sup>ο</sup>: Διάταξη ελέγχου στάθμης και μεταφοράς ρευστού σε δεξαμενές

### 10.1 Εισαγωγή

Σε αυτό το κεφάλαιο γίνεται αναφορά στον έλεγχο στάθμης ενός ρευστού σε ένα σύστημα με δύο δεξαμενές. Για τη συγκεκριμένη εφαρμογή χρησιμοποιήθηκε η διάταξη δεξαμενών που βρίσκεται στο χώρο του εργαστήριου «Ναυτικών Μηχανών και Μηχανολογίας» του τμήματος Ναυπηγών Μηχανικών στο Πανεπιστήμιο Δυτικής Αττικής. Σκοπός αυτής της διάταξης ήταν η προσομοίωση της λειτουργίας των δεξαμενών για τον έλεγχο στάθμης αλλά και μεταφοράς κάποιου ρευστού μεταξύ των δεξαμενών. Οι πληροφορίες σχετικά με τη στάθμη του νερού σε κάθε δεξαμενή δίνονται μέσω κατάλληλων αισθητήρων, ενώ η μεταφορά του νερού πραγματοποιείται με τη χρήση μιας αντλίας νερού. Επιπλέον, στην υπάρχουσα διάταξη υπάρχουν τέσσερις ηλεκτροβάνες που λειτουργούν σε ζεύγη και ορίζουν τη διαδρομή που ακολουθεί το νερό εντός των σωλήνων που συνδέουν τις δύο δεξαμενές. Στη παρακάτω εικόνα (βλ. εικόνα 101) διακρίνουμε το δίκτυο έρματος το οποίο βρίσκετε στο εργαστήριο του ΠΑ.Δ.Α.



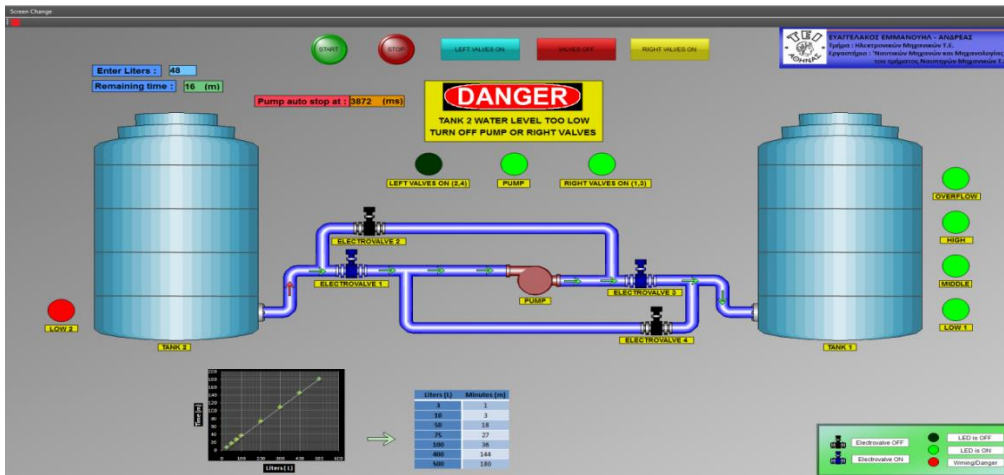
Εικόνα 101: Διάταξη ελέγχου στάθμης και μεταφοράς ρευστού σε δεξαμενές.

## 10.2 Διάταξη δικτύου δεξαμενών εκπαιδευτικών μονάδων

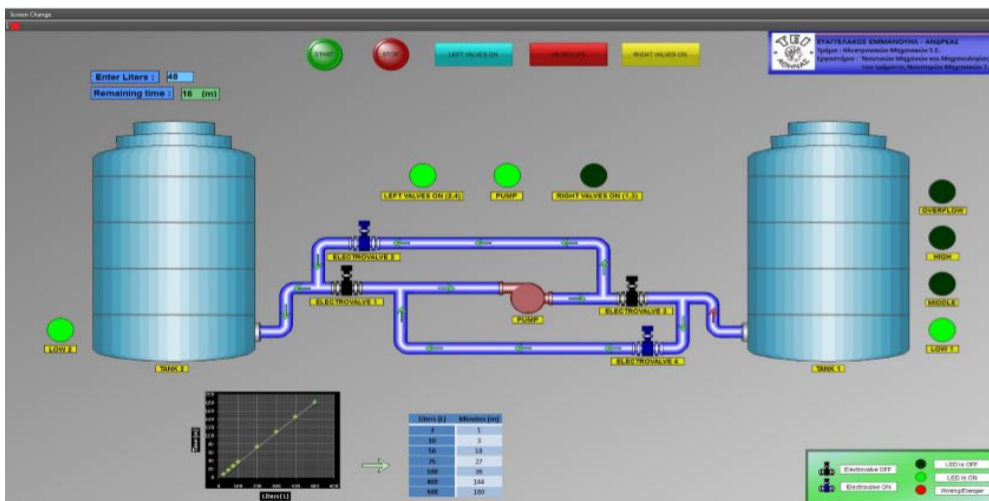
Στις εκπαιδευτικές μονάδες που κατασκευάστηκαν για τη διπλωματική αυτή έχει κατασκευαστεί ένα μοντέλο του δικτύου των δεξαμενών με σκοπό κυρίως, τη κατανόηση του προγραμματισμού ενός προγραμματιζόμενου λογικού ελεγκτή (*PLC*), και στη συνέχεια τον τρόπο συνδεσμολογίας της μονάδας, ώστε να λειτουργεί ο κώδικας και να έχουμε το επιθυμητό αποτέλεσμα.

Για λόγους ασφάλειας και δυσκολίας ύπαρξης πραγματικών δεξαμενών, για όλες τις μονάδες, έχουν τοποθετηθεί λυχνίες ένδειξης (*led*), οι οποίοι αντιστοιχούν στις ηλεκτροβάνες, την αντλία και τους αισθητήρες στάθμης (τρεις σε κάθε δεξαμενή). Επομένως, όταν θα ενεργοποιείται μια ηλεκτροβάνη θα ανάβει το αντίστοιχο *led*.

Στις παρακάτω εικόνες με χρήση του προγράμματος *Graphics Designer* έχει γίνει η απεικόνιση της ροής του νερού (χρήση βέλων) από την αριστερή δεξαμενή στη δεξιά (βλ. εικόνα 102) και αντίστροφα (βλ. εικόνα 103). Επιπλέον διακρίνουμε με μπλε χρώμα τις ανοιχτές ηλεκτροβάνες, οι οποίες επιτρέπουν στο ρευστό να περάσει από μέσα τους, και με μαύρο χρώμα τις κλειστές, οι οποίες δεν επιτρέπουν στο νερό να περάσει από μέσα τους. Η αντλία είναι συνέχεια σε λειτουργία (αλλιώς δεν θα γινόταν η μεταφορά του ρευστού). Τέλος στο πλάι των δεξαμενών διακρίνουμε κάποια *led*, ένα κόκκινο στη μια δεξαμενή και τέσσερα πράσινα στην άλλη. Τα *led* αυτά αντιστοιχούν στους αισθητήρες (φλοτέρ), οι οποίοι ελέγχουν τη στάθμη των δεξαμενών.

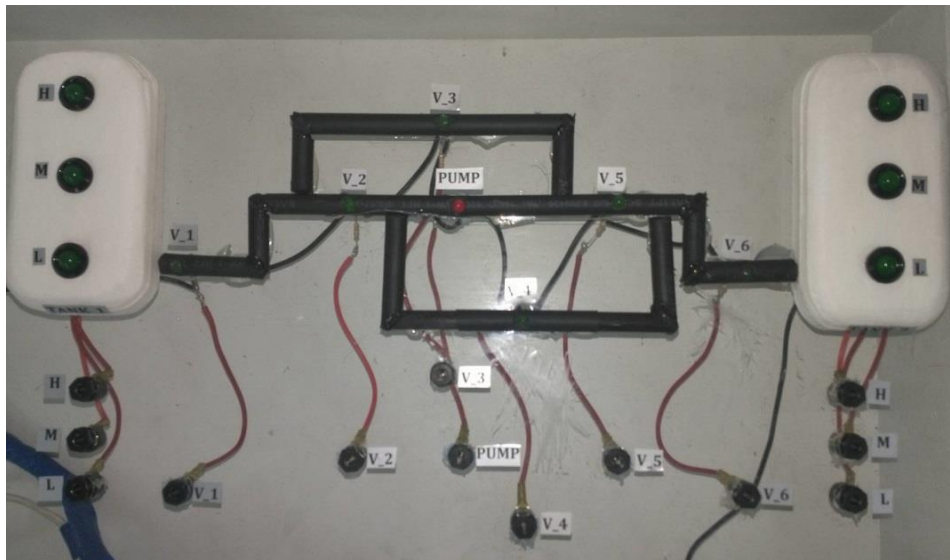


Εικόνα 102: Ροή ρευστού από την αριστερή δεξαμενή στη δεξιά (Ευαγγελάκος Ε. Α. «Ανάπτυξη συστήματος SCADA για βιομηχανικές και ναυτικές εφαρμογές με χρήση PLC», ΤΕΙ Αθήνας, 2017).



Εικόνα 103: Ροή ρευστού από τη δεξιά δεξαμενή στην αριστερή (Ευαγγελάκος Ε. Α. «Ανάπτυξη συστήματος SCADA για βιομηχανικές και ναυτικές εφαρμογές με χρήση PLC», ΤΕΙ Αθήνας, 2017).

Στη παρακάτω εικόνα (βλ. εικόνα 104) ακολουθεί το μοντέλο του υπάρχοντος δικτύου το οποίο κατασκευάστηκε για την καλύτερη κατανόηση των αντίστοιχων εργαστηριακών ασκήσεων. Η λογική μεταφοράς ρευστού από τη μια δεξαμενή στην άλλη είναι η ίδια ε του πραγματικού δικτύου (βλ. εικόνες 102 και 103).



Εικόνα 104: Διάταξη ελέγχου στάθμης νερού και μεταφοράς από τη μια δεξαμενή στην άλλη στην εκπαιδευτική μονάδα εκμάθησης.

Τα led των δεξαμενών (*tanks*) δηλώνουν την ύπαρξη αισθητήρων στάθμης ρευστού σε τρεις καταστάσεις, την υψηλή (*High*), τη μεσαία (*Medium*) και τη χαμηλή (*Low*). Τα ονόματα των ηλεκτροβανών – βαλβίδων (*Valve*) ωστόσο, έχουν αρίθμηση από τα αριστερά προς τα δεξιά. Για περισσότερη ευκολία το led το οποίο αντιστοιχεί στην αντλία (*Pump*) έχει κόκκινο.

## Κεφάλαιο 11<sup>ο</sup>: Έλεγχος ασύγχρονου κινητήρα

Οι ασύγχρονοι τριφασικοί κινητήρες μετατρέπουν την εναλλασσόμενη ηλεκτρική ενέργεια του τριφασικού δικτύου σε μηχανική. Ονομάζονται και επαγωγικοί κινητήρες, διότι το ρεύμα διέγερσης του ρότορα δημιουργείται από επαγωγή αντί να παρέχεται με ηλεκτρική σύνδεση από κάποια πηγή. Ο όρος «ασύγχρονες», αποδίδεται διότι οι στροφές λειτουργίας τους είναι λίγο μικρότερες από το σύγχρονο αριθμό στροφών όταν η μηχανή λειτουργεί σαν κινητήρας και λίγο περισσότερες από τον σύγχρονο αριθμό στροφών όταν η μηχανή λειτουργεί σαν γεννήτρια.

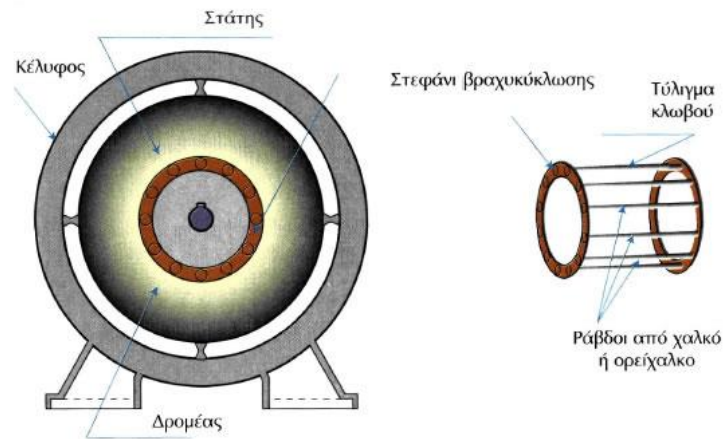
Οι κινητήρες αυτοί αποτελούνται από δύο βασικά τμήματα συζευγμένων τυλιγμάτων, το δρομέα (τα ρεύματα αναπτύσσονται εξ επαγωγής) και το στάτη (τροφοδοτείται με εναλλασσόμενο ρεύμα συχνότητας  $f$ , από μια εξωτερική πηγή). Ο δρομέας (ρότορας) αποτελείται από έναν ηλεκτροφόρο αγωγό, ο οποίος περιβάλλεται από πυκνές περιελίξεις (σπείρες) ώστε να περιέχει όσο μεγαλύτερο μήκος αγωγού γίνεται για δεδομένο όγκο. Από την άλλη, ο στάτης αποτελείται από μόνιμους ή τεχνητού μαγνήτες, οι οποίοι δημιουργούν μαγνητικό πεδίο.

Οι ασύγχρονοι τριφασικοί κινητήρες ανάλογα με τη κατασκευή του δρομέα διακρίνονται σε:

- Κινητήρα με βραχυκυκλωμένο δρομέα (κλειστού κλωβού): Σε αυτή την κατηγορία κινητήρων, ο δρομέας αποτελείται από δύο κυκλικές βάσεις ενωμένες μεταξύ τους με χαλύβδινα ηλεκτρικά μονωμένα ελάσματα. Στην εξωτερική του επιφάνεια συναντώνται κατάλληλες αυλακώσεις, στις οποίες τοποθετούνται χαλύβδινοι ράβδοι. Τα άκρα των ράβδων αυτών είναι ηλεκτρικά συνδεδεμένα μεταξύ τους, επιτρέποντας τη ροή του επαγωγικού ρεύματος το οποίο απαιτείται για την ανάπτυξη ροπής στρέψης (βλ. εικόνα 105). Η εκκίνηση των κινητήρων αυτών επιτυγχάνεται μέσω εξωτερικών μεταβλητών αντιστάσεων οι οποίες είναι συνδεδεμένες με τα άκρα της τριφασικής περιέλιξης του δρομέα, μειώνοντας έτσι το απαιτούμενο ρεύμα εκκίνησης.

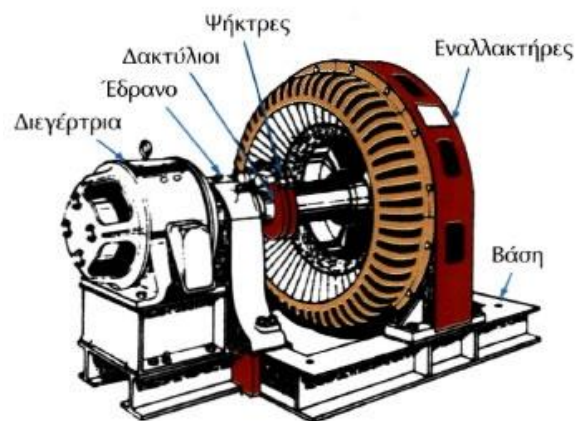


Εικόνα 105: Ασύγχρονος τριφασικός κινητήρας με βραχυκυκλωμένο δρομέα (<http://hlektrologia.gr>).



Εικόνα 106: Ασύγχρονος τριφασικός κινητήρας με βραχυκυκλωμένο δρομέα (<http://ebooks.edu.gr>).

• Κινητήρα με δακτυλιοφόρο δρομέα: Στους κινητήρες αυτούς, ο δρομέας αποτελείται από τριφασική περιέλιξη σε συνδεσμολογία αστέρα. Η σύνδεση των άκρων περιέλιξης επιτυγχάνεται με κατάλληλες διατάξεις δακτυλίων με εξωτερικές μεταβλητές αντιστάσεις. Κατά την εκκίνηση του κινητήρα παρατηρείται μείωση του ρεύματος στο δρομέα μέσω των αντιστάσεων, με αποτέλεσμα την αύξηση της ροπής εκκίνησης. Καθώς αυξάνεται η ταχύτητα οι αντιστάσεις μειώνονται σταδιακά μέχρι να μηδενιστούν.



Εικόνα 107: Ασύγχρονος τριφασικός κινητήρας με δακτυλιοφόρο δρομέα (<https://sites.google.com>).

## 11.1 Αρχή λειτουργίας ασύγχρονου τριφασικού κινητήρα

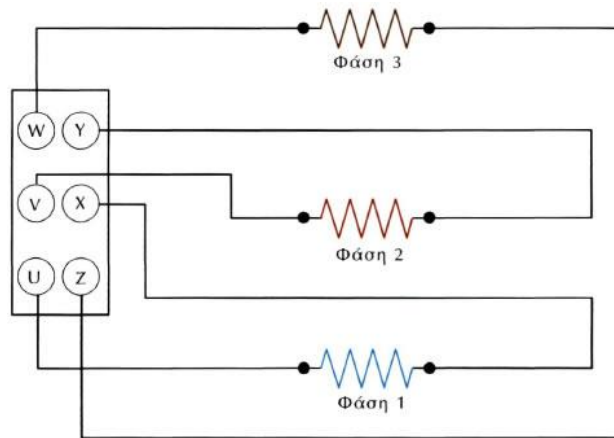
Η περιέλιξη του στάτη αποτελείται από τρία τυλίγματα, ένα για κάθε φάση, συμμετρικά τοποθετημένα. Τα τυλίγματα αυτά τροφοδοτούνται με τριφασικό ρεύμα με αποτέλεσμα τη δημιουργία περιστροφικού μαγνητικού πεδίου με σταθερή γωνιακή ταχύτητα, ανεξάρτητη από το φορτίο του στάτη. Το περιστροφικό μαγνητικό πεδίο οδηγεί στην ανάπτυξη ηλεκτρεγερτικής δύναμης στα άκρα των αγωγών του τυλίγματος του δρομέα. Το ρεύμα που διαρρέει τους αγωγούς επιφέρει τη δημιουργία δύναμης *Laplace* που περιστρέφει το δρομέα σύμφωνα με τη φορά του μαγνητικού πεδίου. Σε περίπτωση αλλαγής της φοράς δύο οποιονδήποτε φάσεων, αλλάζει και η φορά περιστροφής του μαγνητικού πεδίου.

Επιπλέον η ταχύτητα περιστροφής του κινητήρα ( $N$ ), είναι πάντοτε μικρότερη από τη σύγχρονη ταχύτητα ( $N_s$ ) του στρεφόμενου μαγνητικού πεδίου. Η σύγχρονη ταχύτητα εξαρτάται από τη συχνότητα ( $f$ ) του ρεύματος τροφοδοσίας και τα ζεύγη πόλων ( $p$ ) του τυλίγματος του στάτη. Ο δρομέας τείνει να περιστραφεί με την ταχύτητα του κινητήρα ( $N$ ) χωρίς όμως ποτέ να τη φτάνει. Η διαφορά μεταξύ της σύγχρονης ταχύτητας ( $N_s$ ) και της ταχύτητας του δρομέα ( $N$ ) ονομάζεται ολίσθηση. Σε περίπτωση που ο δρομέας αποκτήσει την ίδια ταχύτητα με τη σύγχρονη ταχύτητα ( $N = N_s$ ), τότε δεν θα προλάβαινε το μαγνητικό πεδίο να κόψει τους αγωγούς του δρομέα, με αποτέλεσμα η επαγωγή να μην είναι εφικτή. Ως εκ τούτου η μαγνητική δύναμη θα μηδενιζόταν και η κίνηση του κινητήρα θα τερματιζόταν.

## 11.2 Εκκινητής Αστέρα-Τριγώνου

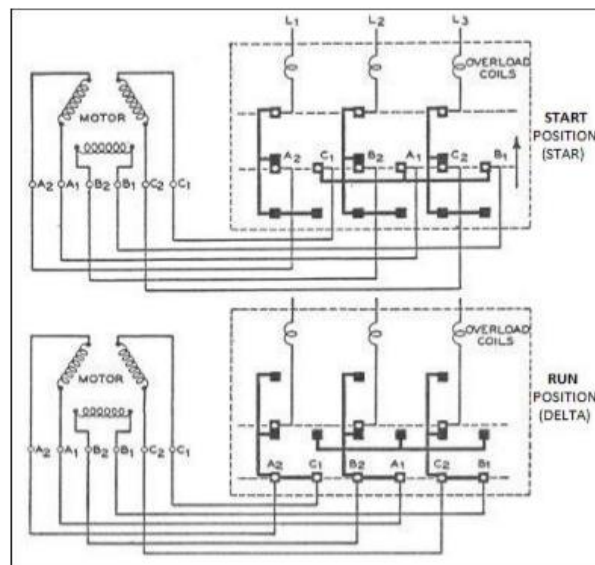
Ο εκκινητής αστέρα-τριγώνου χρησιμοποιείται σε μεγάλους κινητήρες με σκοπό τον περιορισμό του ρεύματος εκκίνησης στο επιθυμητό επίπεδο. Γενικότερα, το ρεύμα εκκίνησης των ασύγχρονων τριφασικών κινητήρων βραχυκυκλωμένου δρομέα είναι τέσσερις έως οκτώ φορές μεγαλύτερο από το ρεύμα κανονικής λειτουργίας. Για το λόγο αυτό, η απευθείας εκκίνηση του κινητήρα προκαλεί συνήθως πτώση τάσης στο δίκτυο τροφοδοσίας. Ως εκ τούτου, για τον περιορισμό του ρεύματος εκκίνησης αλλά και για την προστασία του κινητήρα, είναι συνήθη η χρήση του εκκινητή αστέρα-τριγώνου (Y-Δ). Αξίζει να σημειωθεί ότι το ρεύμα εκκίνησης ενός κινητήρα σε συνδεσμολογία αστέρα, είναι κατά τρεις φορές μικρότερο από το ρεύμα εκκίνησης σε συνδεσμολογία τριγώνου.

Κατά την εκκίνηση του κινητήρα, τα έξι άκρα του τριφασικού τυλίγματος του στάτη συνδέονται μέσω του εκκινητή (βλ. εικόνα 108) σε συνδεσμολογία αστέρα, ενώ μετά από κάποιο διάστημα λειτουργίας, πραγματοποιείται αλλαγή σε συνδεσμολογία τριγώνου. Αρκετοί κινητήρες, εφοδιάζονται με κατάλληλη διάταξη αυτοματισμού, με σκοπό την αυτόματη μετάβαση σε συνδεσμολογία τριγώνου μετά από την πάροδο συγκεκριμένου χρονικού διαστήματος (από 1 έως 10 sec.).



Εικόνα 108: Τα έξι ελεύθερα άκρα των τριών φάσεων του τριφασικού εναλλακτήρα συνδέονται στους έξι ακροδέκτες της μηχανής (<http://ebooks.edu.gr>).

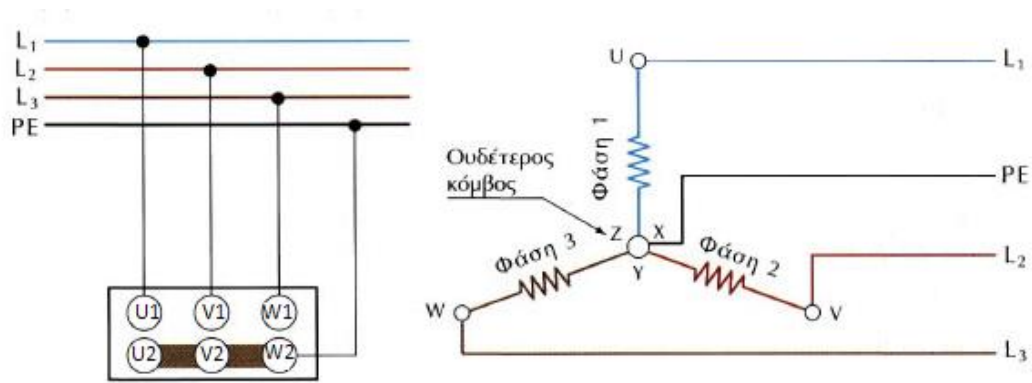
Παρακάτω παρουσιάζεται μια σχηματική απεικόνιση του εκκινητή αστέρα-τριγώνου με τις κατάλληλες συνδεσμολογίες.



Εικόνα 109: Συνδεσμολογία Αστέρα – Τρίγωνο (Ευαγγελάκος Ε. «Ανάπτυξη συστήματος SCADA για βιομηχανικές και ναυτικές εφαρμογές με χρήση PLC»).

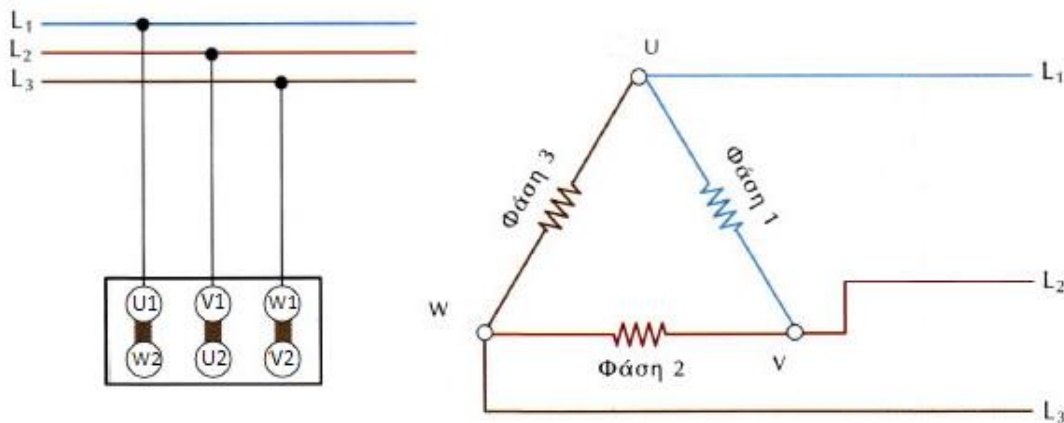
Κατά την εκκίνηση του κινητήρα, τα τυλίγματα πρέπει να είναι συνδεδεμένα σε συνδεσμολογία αστέρα. Στη σύνδεση σε αστέρα συνδέουμε τους ακροδέκτες  $U_2$ ,  $V_2$ ,  $W_2$ . Με αυτό τον τρόπο δημιουργούμε τον ονομαζόμενο ουδέτερο κόμβο της μηχανής. Στους τρεις άλλους ελεύθερους ακροδέκτες  $U_1$ ,  $V_1$ ,  $W_1$  συνδέονται με τις φάσεις του τριφασικού δικτύου  $L_1$ ,  $L_2$  και  $L_3$  αντίστοιχα





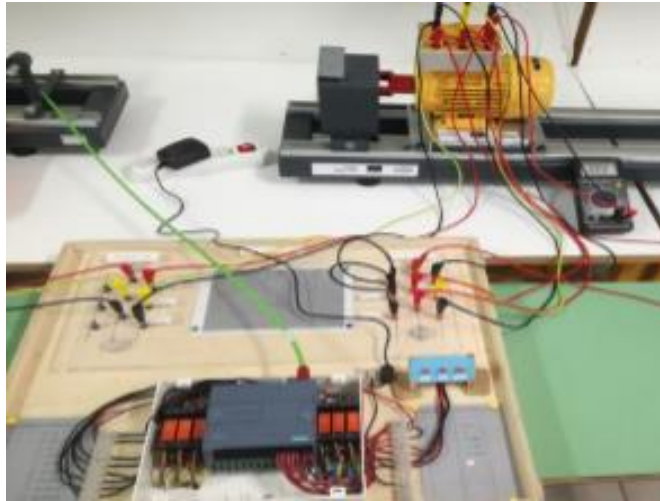
Εικόνα 110: Συνδεσμολογία Αστέρια (<http://ebooks.edu.gr>).

Μετά από την εκκίνηση του κινητήρα και για την αλλαγή σε συνδεσμολογία τριγώνου, απαιτείται η δημιουργία βραχυκυκλωμάτων μεταξύ των άκρων:  $U1-W2$ ,  $V1-U2$  και  $W1-V2$ , αλλά και η σύνδεση τους με τις φάσεις  $L1$ ,  $L2$  και  $L3$  αντίστοιχα.



Εικόνα 111: Συνδεσμολογία Τρίγωνο (<http://ebooks.edu.gr>).

Παρακάτω παρουσιάζεται η τελική συνδεσμολογία ενός μοντέλου ασύγχρονου τριφασικού κινητήρα του εργαστηρίου «Ναυτικών Μηχανών και Μηχανολογίας του Πανεπιστημίου Δυτικής Αττικής» ο οποίος έχει συνδεθεί με έναν προγραμματιζόμενο λογικό ελεγκτή (PLC).



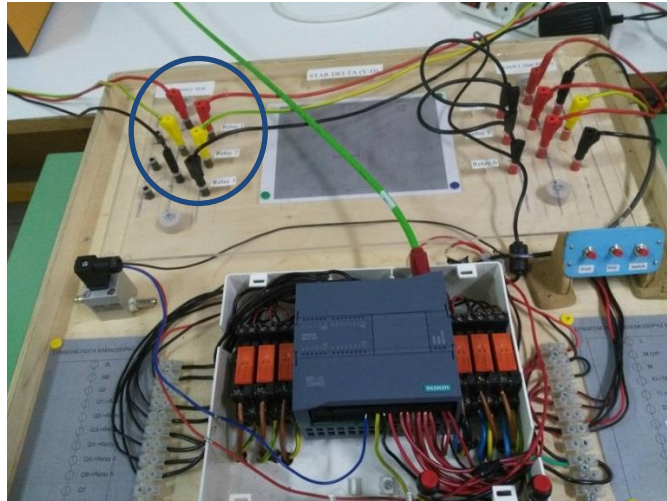
Εικόνα 112: Τελική συνδεσμολογία ενός μοντέλου ασύγχρονου τριφασικού κινητήρα του εργαστηρίου.

Για περαιτέρω κατανόηση της συνδεσμολογίας των τυλιγμάτων του κινητήρα στην προϋπάρχουσα διάταξη αναφέρονται τα εξής βήματα :

1. Οι φάσεις  $L1$ ,  $L2$ ,  $L3$  του τροφοδοτικού, συνδέονται με τους τρεις ηλεκτρονόμους (1,2,3) που βρίσκονται στα αριστερά από το  $PLC$  (μέσα στο μπλε κύκλο). Πιο συγκεκριμένα, οι τρεις αυτές φάσεις συνδέονται στις επαφές  $CO$  (*Change Over*) των ηλεκτρονόμων.

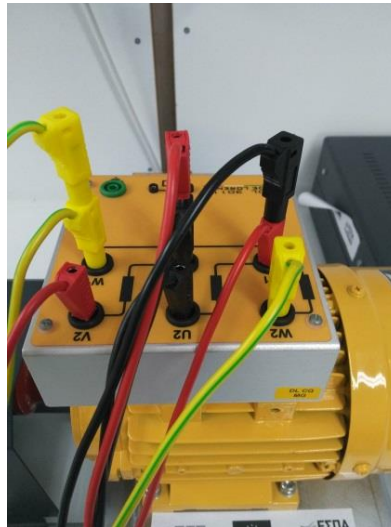


Εικόνα 113: Φάσεις τροφοδοτικού.



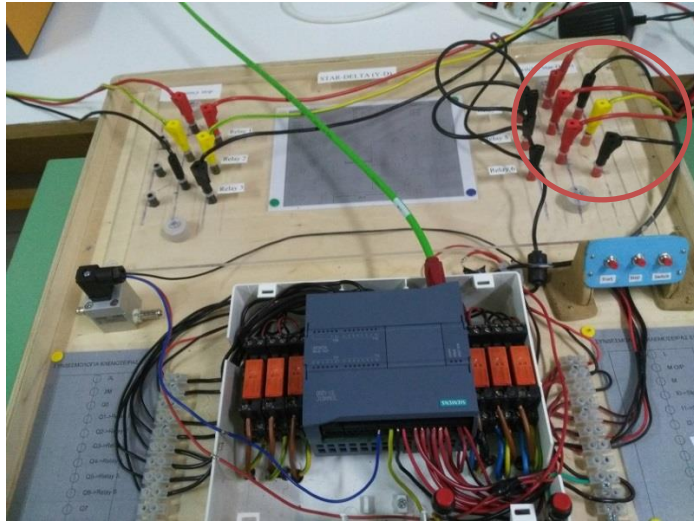
Εικόνα 114: Ρελέ που συνδέονται με το τροφοδοτικό.

2. Στη συνέχεια, οι *NO (Normal Open)* επαφές των τριών αυτών ηλεκτρονόμων, συνδέονται με τα τυλίγματα *W1, V1, U1* του κινητήρα αντίστοιχα.



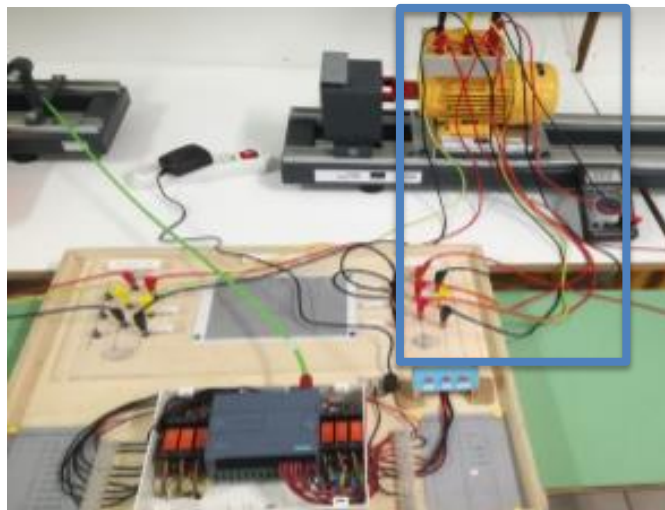
Εικόνα 115: Συνδέσεις στον κινητήρα.

3. Οι επαφές *NC (Normal Close)* των τριών ηλεκτρονόμων (4,5,6) που βρίσκονται δεξιά από το *PLC* (μέσα στο κόκκινο κύκλο), βραχυκυκλώνονται μεταξύ τους.



Εικόνα 116: Ρελέ που βραχυκυκλώνονται μεταξύ τους.

4. Τέλος, τα άκρα  $W2$ ,  $U2$ ,  $V2$  και  $U1$ ,  $V1$ ,  $W1$ , συνδέονται με τις επαφές  $CO$  και  $NO$  των ηλεκτρονόμων (4,5,6) αντίστοιχα.



Εικόνα 117: Σύνδεση άκρων κινητήρα με τις επαφές  $NO$  και  $CO$  των 3<sup>ων</sup> ρελέ, μέσα στο μπλε πλαίσιο.

Στην εκπαιδευτική μονάδα οι μπόρνες που ενώνονται με κινητήρες (λόγο έλλειψης μοντέλων) αντικαθίστανται από μπόρνες οι οποίες ενώνονται με *led* τα οποία δηλώνουν τη συνδεσμολογία αστέρα – τρίγωνο, με σκοπό ο φοιτητής να εξοικειωθεί με τη γλώσσα *Ladder* και τον προγραμματισμό ενός *PLC*.

## Κεφάλαιο 12<sup>ο</sup>: Παράρτημα

### 12.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει αναφορά στην ανάπτυξη τυπικών εφαρμογών, σε μορφή εργαστηριακών ασκήσεων. Πιο συγκεκριμένα οι εφαρμογές αυτές αναφέρονται στην εξοικείωση με το περιβάλλον *TIA PORTAL V13*, της εταιρίας *Siemens*, στον έλεγχο στάθμης και μεταφορέας ρευστού μεταξύ δύο δεξαμενών, στην εκκίνηση ενός τριφασικού ασύγχρονου κινητήρα και τη χρήση αισθητήρων για τη μέτρηση των στροφών της μηχανής. Θα αναφερθούμε στο κώδικα που αναπτύχθηκε στα πλαίσια επίλυσης των ασκήσεων καθώς και η απαραίτητη συνδεσμολογία που εφαρμόσαμε στη μονάδα εκμάθησης, για τη λειτουργία του συστήματος με το επιθυμητό αποτέλεσμα.

### 12.2 Άσκηση 1: Εξοικείωση με το περιβάλλον «TIA Portal V13» και τη γλώσσα προγραμματισμού Ladder.

Σκοπός της πρώτης άσκησης είναι να βοηθήσει το φοιτητή να κατανοήσει τη διαδικασία δημιουργίας σύνδεσης των προγραμματιζόμενων λογικών ελεγκτών (*PLC*) με τον ηλεκτρονικό υπολογιστή, το περιβάλλον προγραμματισμού των *PLC* (*TIA Portal V13*) καθώς και τη γλώσσα προγραμματισμού *Ladder*.

Αρχικά έχουν γίνει τα βήματα σύνδεσης του *PLC* και του υπολογιστή μέσω του προγράμματος *TIA Portal V13*. Για την εξοικείωση του φοιτητή με τη γλώσσα *Ladder* ακολουθούντα παρακάτω βήματα, έτσι ώστε με τη κατάλληλη συνδεσμολογία της εκπαιδευτικής μονάδας να λειτουργεί το πρόγραμμα.

1. Κάνοντας χρήση μιας εισόδου (I0.0 – I1.5), να ενεργοποιηθεί το *led* της βαλβίδας του δικτύου «V1».
2. Κάνοντας χρήση δύο εισόδων (I0.0 – I1.5) **ταυτόχρονα**, να ενεργοποιηθεί το *led* της βαλβίδας του δικτύου «V2».
3. Κάνοντας χρήση **οποιασδήποτε** από δύο εισόδους (I0.0 – I1.5) -θα χρησιμοποιούνται εναλλάξ, να ενεργοποιηθεί το *led* της βαλβίδας του δικτύου «V3».
4. Κάνοντας χρήση μιας εισόδου (I0.0 – I1.5) να ενεργοποιηθεί το *led* της συνδεσμολογίας αστέρα (*Star*) **μετά από πάροδο χρόνου 5 sec.**

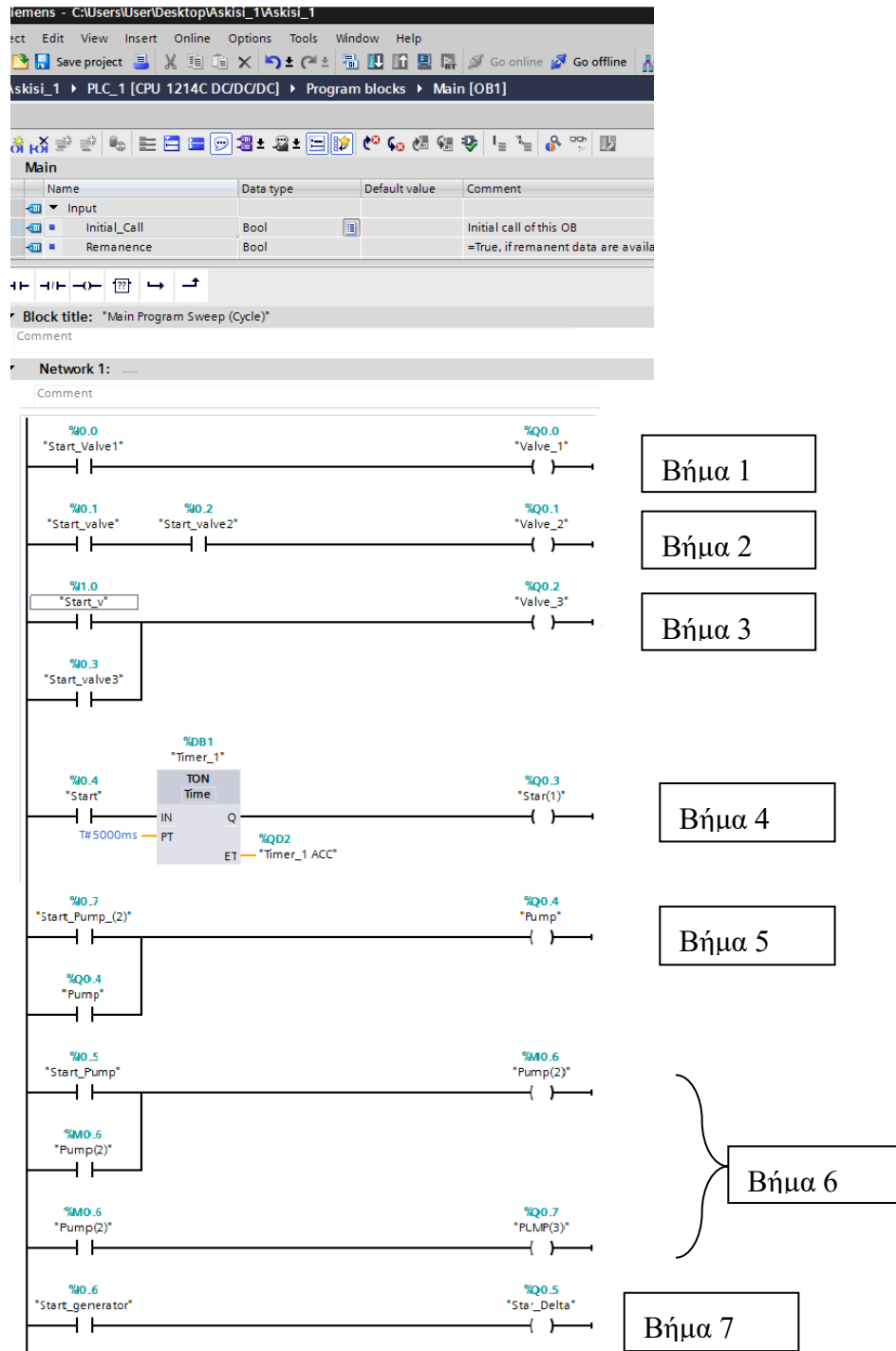
5. Να αναπτυχθεί πρόγραμμα μόνιμης ενεργοποίησης του *led Pump* μέσω της εισόδου «*Push\_B*», με χρήση **αυτοσυγκράτησης** (*feedback*) και μίας εξόδου. Δηλαδή, η συγκεκριμένη μεταβλητή εξόδου (που είναι συνδεδεμένη στο «*led Pump*») να ενεργοποιηθεί μέσω ενός παλμού («πάτημα» διακόπτη *Push\_B*) στην είσοδο.
6. Να αναπτυχθεί πρόγραμμα μόνιμης ενεργοποίησης του *led Pump* μέσω της εισόδου «*Push\_B*», με χρήση **αυτοσυγκράτησης** (*feedback*) και **εσωτερικής μεταβλητής** (μεταβλητή τύπου *Global Memory*). Δηλαδή, η συγκεκριμένη είσοδος να ενεργοποιεί μόνιμα εσωτερική μεταβλητή (π.χ. «*Flag\_1*») η οποία με τη σειρά της θα «οδηγεί» την έξοδο (που είναι συνδεδεμένη στο «*led Pump*»).
7. Κάνοντας χρήση ενός ηλεκτρονόμου (*Relay*) να πραγματοποιηθεί η εναλλαγή ενεργοποίησης των *leds* «αστέρα» και «τριγώνου». Δηλαδή, χωρίς μανδάλωση του ηλεκτρονόμου να ενεργοποιηθεί αρχικά το *led* «αστέρα», ενώ η ακόλουθη μανδάλωση του να έχει ως αποτέλεσμα την ενεργοποίηση του «τριγώνου» και αντίστοιχα την απενεργοποίηση του «αστέρα» (χρήση των δύο επαφών NC-NO).

## **ΠΡΟΣΟΧΗ ΣΤΗ ΣΥΝΔΕΣΗ ΤΟΥ ΤΡΟΦΟΔΟΤΙΚΟΥ ΜΕ ΤΗ ΜΟΝΑΔΑ.**

### ΣΗΜΕΙΩΣΗ:

- Κάθε φορά που δημιουργείται ένα νέο *project* ή διορθώνεται ένα ήδη υπάρχον πρόγραμμα, μέσω του προγράμματος *TIA Portal*, για να το αποθηκεύσουμε από τον υπολογιστή στη μονάδα *PLC* με σκοπό να τεθεί σε λειτουργία, χρειάζεται να επιλέξουμε την εντολή «*Download to Device*» και στη συνέχεια να επιλέξουμε την εντολή «*Start CPU*».
- Πριν προχωρήσουμε σε οποιαδήποτε αλλαγή σε υπάρχον πρόγραμμα, πρώτα θέτουμε σε κατάσταση αδράνειας τη μονάδα ΠΛΕ (επιλέγουμε την εντολή «*Stop CPU*»).
- Όταν αρχικά έχουμε επιλέξει την κατάλληλη μονάδα *PLC* να συνδεθεί με τον υπολογιστή, επιλέγουμε την εντολή «*Flash led*» για να βεβαιωθούμε ότι η διασύνδεση είναι επιτυχής (παρατηρήστε τις οπτικές ενδείξεις στη μονάδα *PLC*).
- Με χρήση της εντολής «*Monitoring on/off*» η οποία βρίσκεται στη γραμμή εργασιών στο παράθυρο «*Main*», μπορούμε να παρατηρούμε σε πραγματικό χρόνο την εκτέλεση του προγράμματος που «τρέχει» εκείνη τη στιγμή.

Παρακάτω ακολουθεί ο κώδικας σε γλώσσα *Ladder* των παραπάνω 7 βημάτων, η εξήγηση του προγράμματος, καθώς και η σωστή συνδεσμολογία της μονάδας για την επίτευξη του επιθυμητού αποτελέσματος.



Εικόνα 118: Κώδικας σε γλώσσα Ladder των 7 βημάτων της εκφώνησης.

Εάν κάνουμε χρήση της εντολής *Monitoring on/off*, τότε ο κώδικας θα μας δείχνει τη ροή του «ρεύματος» στο προγράμματος, επομένως θα μπορούμε να διακρίνουμε καλύτερα τυχόν σφάλματα στον κώδικα ή και στη συνδεσμολογία.

Siemens - C:\Users\User\Desktop\Askisi\_1\Askisi\_1

Project Edit View Insert Online Options Tools Window Help

Save project

Askisi\_1 > PLC\_1 [CPU 1214C DC/DC/DC] > Program blocks > Main [OB1]

Monitoring on/off

PLC programming

Name	Data type	Default value	Comment
1	Input		
2	Initial_Call	Bool	Initial call of this OB
3	Remanence	Bool	=True, if remanent data are available

Network 1: .....

Comment

Bήμα 1

Bήμα 2

Bήμα 3

Bήμα 4

Εικόνα 119: Χρήση της εντολής monitoring on/off.



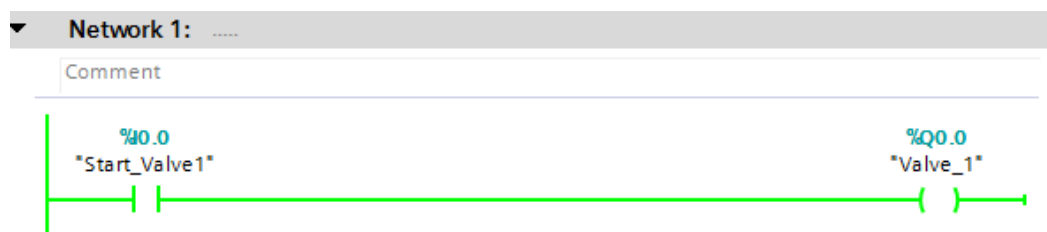
Πίνακας 4: Σύμβολα, τελεστές, ονομασίες του κώδικα και η εξήγησή τους.

ΣΥΜΒΟΛΟ	ΤΕΛΕΣΤΗΣ	ΟΝΟΜΑΣΙΑ	ΕΞΗΓΗΣΗ
<b>I0</b>	I0.0	Start_Valve1	Διακόπτης: Ανοίγει τη βαλβίδα 1
<b>I1</b>	I0.1	Start Valve	Διακόπτης: Βοηθάει στο άνοιγμα της βαλβίδας 2
<b>I2</b>	I0.2	Start_Valve2	Διακόπτης: Ανοίγει τη βαλβίδα 2
<b>I3</b>	I0.3	Start_Valve3	Διακόπτης: Ανοίγει τη βαλβίδα 3
<b>I4</b>	I0.4	Start	Διακόπτης: Ανοίγει το κινητήρα σε αστέρα συνδεσμολογία
<b>I5</b>	I0.5	Start_Pump	Κουμπί: Ενεργοποιεί την αντλία
<b>I6</b>	I0.6	Start_generatory	Διακόπτης: Αλλάζει τις στροφές της μηχανής σε τρίγωνο
<b>I7</b>	I0.7	Start_Pump_(2)	Κουμπί: Ενεργοποιεί την αντλία
<b>I8</b>	I1.0	Start_v	Διακόπτης: Ανοίγει τη βαλβίδα 3
<b>Q0</b>	Q0.0	Valve_1	Led: Βαλβίδα 1
<b>Q1</b>	Q0.1	Valve_2	Led: Βαλβίδα 2
<b>Q2</b>	Q0.2	Valve_3	Led: Βαλβίδα 3
<b>Q3</b>	Q0.3	Star(1)	Led: Αντιστοιχεί στη συνδεσμολογία αστέρα στο 3φασικό κινητήρα.
<b>Q4</b>	Q0.4	Pump	Led: Αντλία
<b>Q5</b>	Q0.5	Star_Delta	Relay: Για την εναλλαγή από αστέρα σε τρίγωνο
<b>Q6</b>	Q0.6	Pump(2)	Led: Αντλία
<b>Q7</b>	Q0.7	PUMP(3)	Led: Αντλία
<b>M1</b>	M2.2	Pump(2)	Global Memory: Διατηρεί τη κατάσταση που αντιστοιχεί στο στοιχείο M1
<b>T1</b>	DB1	Timer_1	Χρονόμετρο: Ρυθμισμένο στα 5sec να ανοίξει την έξοδο Q3
<b>T1_acc</b>	QD2	Timer_1 ACC	Global Memory: Για να μπορούμε να επικαλούμαστε το στοιχείο T1

### 12.2.1 Επεξήγηση ενδεικτικού κώδικα

Σε αυτή την υποενότητα θα γίνει αναφορά στον κώδικα που κατασκευάστηκε με σκοπό τη λύση των 7 βημάτων που αναφέρονται παραπάνω. Αρχικά να σημειωθεί ότι όταν κατασκευάζουμε ένα κώδικα προσέχουμε να μην έχουμε τον ίδιο τελεστή εξόδου σε περισσότερες από μια γραμμές. Για τις εισόδους δεν έχουμε αυτό το περιορισμό μιας και εξαρτάτε από τη συνδεσμολογία που έχουμε κάνει. Παρακάτω αναφέρετε η επεξήγηση του κώδικα με τη βοήθεια της εικόνας 118.

Βήμα 1<sup>ο</sup>: Στη πρώτη γραμμή του κώδικα έχουμε μία είσοδο (I0.0) η οποία είναι *normal open*, δηλαδή χρειάζεται να «κλείσουμε» το κύκλωμα για τη λειτουργία του, και αντιστοιχεί σε ένα διακόπτη (το οποίο το ορίζουμε εμείς με κατάλληλη συνδεσμολογία), ο οποίος ενεργοποιεί την έξοδο (Q0.0). Η έξοδος Q0 ενεργοποιεί το *led* της βαλβίδας του δικτύου «V1».



Εικόνα 120: Όταν ενεργοποιήσουμε την είσοδο I0.0.

Βήμα 2<sup>ο</sup>: Στη δεύτερη γραμμή του κώδικα έχουμε δύο εισόδους (I0.1 και I0.2) οι οποίες είναι *normal open*, δηλαδή χρειάζεται να «κλείσουμε» το κύκλωμα για τη λειτουργία του, και αντιστοιχούν σε δύο διακόπτες (το οποίο το ορίζουμε εμείς με κατάλληλη συνδεσμολογία), έναν για κάθε είσοδο, οι οποίοι όταν τεθούν σε λειτουργία και οι δύο (Λογική πύλη AND) ενεργοποιούν την έξοδο (Q0.1). Η έξοδος Q1 ενεργοποιεί το *led* της βαλβίδας του δικτύου «V2».



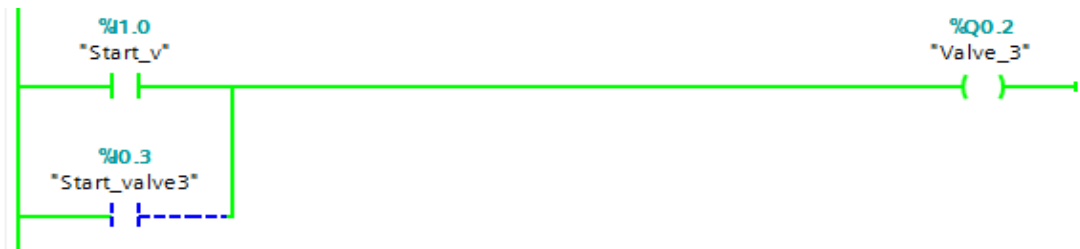
Εικόνα 121: Όταν ενεργοποιήσουμε τη μια είσοδο.



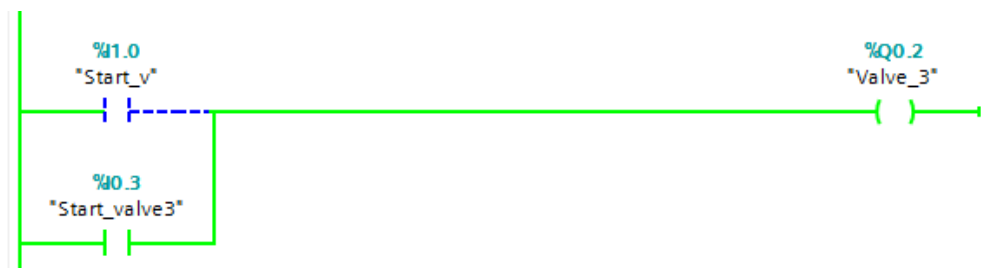
Εικόνα 122: Όταν ενεργοποιήσουμε και τις δύο εισόδους (Λογική πύλη AND).

Βήμα 3<sup>ο</sup>: Στη τρίτη και τέταρτη γραμμή του κώδικα έχουμε μία είσοδο (I1.0) και μια είσοδο (I0.3) αντίστοιχα, οι οποίες είναι *normal open*, δηλαδή χρειάζεται να «κλείσουμε» το κύκλωμα για τη λειτουργία του, και αντιστοιχούν σε δύο διακόπτες (το οποίο το ορίζουμε εμείς με κατάλληλη συνδεσμολογία), έναν για κάθε είσοδο. Όταν ενεργοποιηθεί, έστω, μια από τις δύο

εισόδους (Λογική πύλη *OR*), τότε ενεργοποιείτε η έξοδος (Q0.2). Η έξοδος Q2 ενεργοποιεί το *led* της βαλβίδας του δικτύου «V3».



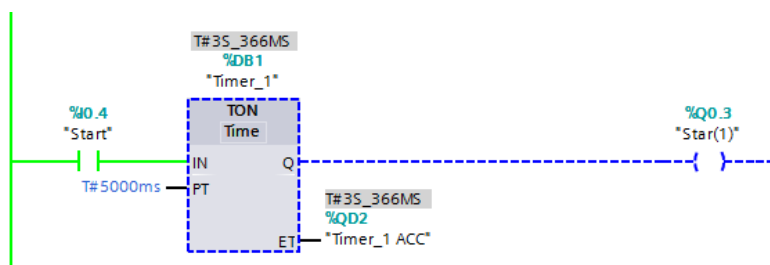
Εικόνα 123: Όταν ενεργοποιήσουμε μια από τις δύο εισόδους (Λογική πύλη *OR*).



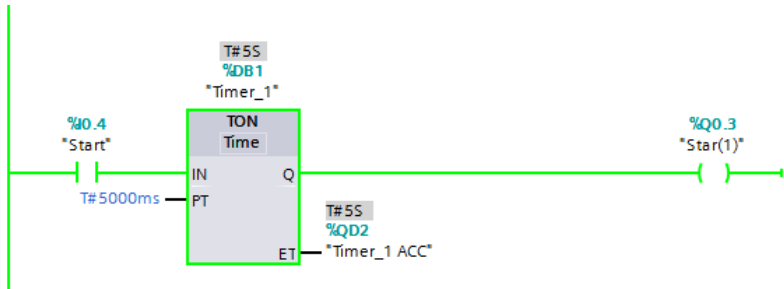
Εικόνα 124: Όταν ενεργοποιήσουμε μια από τις δύο εισόδους (Λογική πύλη *OR*).

Βήμα 4<sup>ο</sup>: Στη πέμπτη γραμμή του κώδικα έχουμε μια εισόδους (I0.4) η οποία είναι *normal open* και αντιστοιχεί σε ένα διακόπτη (το οποίο το ορίζουμε εμείς με κατάλληλη συνδεσμολογία). Όταν ενεργοποιηθεί ο διακόπτης τότε ενεργοποιεί το χρονόμετρο (*Timer – TON*). Το χρονόμετρο έχει οριστεί από εμάς να ενεργοποιήσει σε συγκεκριμένο χρονικό διάστημα (*PT*) την έξοδο (Q0.3). Η έξοδος Q3 ενεργοποιεί το *led* της συνδεσμολογίας του αστέρα «*Star*».

Στην παρακάτω εικόνα (βλ. εικόνα 125) παρατηρούμε ότι η έξοδος δεν ενεργοποιείτε εάν η συσσωρευμένη τιμή (*ET*) δεν γίνει ίση με το χρονικό διάστημα (*PT*).

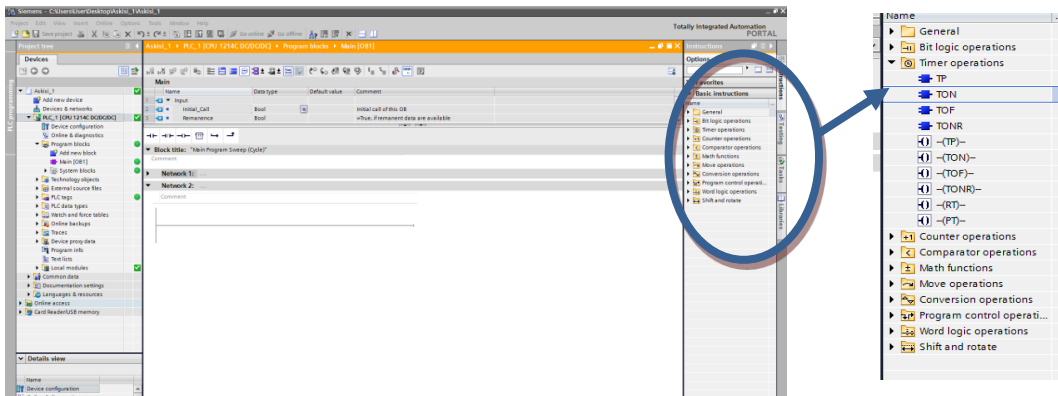


Εικόνα 125: Η συσσωρευμένη τιμή (*ET*) δεν έχει γίνει ακόμη ίση με το χρονικό διάστημα (*PT*).

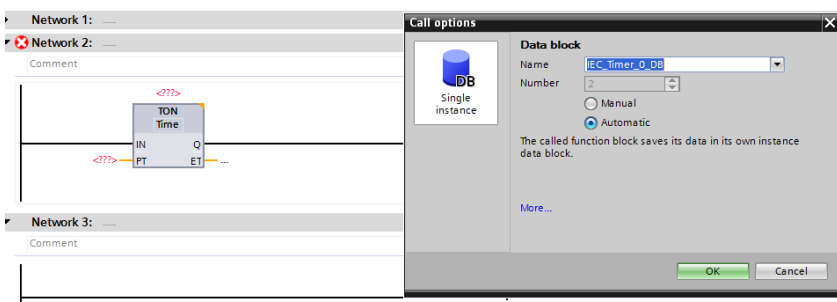


Εικόνα 126: Η συσσωρευμένη τιμή (ET) έγινε ίση με το χρονικό διάστημα (PT).

Η δομή/εντολή *Timer (TON)* είναι διαθέσιμη στο περιβάλλον *TIA PORTAL*. Είναι ο πλέον συνηθής «χρονοδιακόπτης» στο συγκεκριμένο γραφικό περιβάλλον και μπορεί εύκολα να αναζητηθεί (βλ. εικόνα 127).

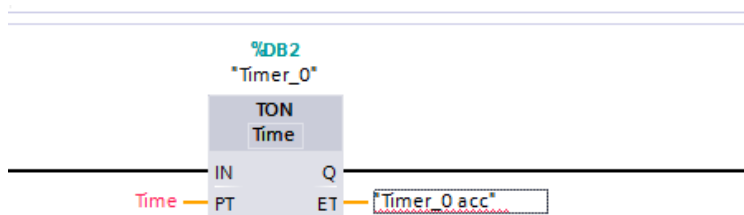


Εικόνα 127: Tools bar με χρήσιμες εντολές για την κατασκευή κώδικα Ladder – Διακρίνεται η εντολή *Timer TON*.



Εικόνα 128: Επιλογή *Timer On Delay*.

Αφού τοποθετήσουμε τη δομή στο πρόγραμμά μας, πρέπει να ορίσουμε τις αντίστοιχες μεταβλητές εισόδου/εξόδου (*IN, Q, PT, ET*). Ο ρόλος της κάθε μεταβλητής παρουσιάζεται στον Πίνακα 5.

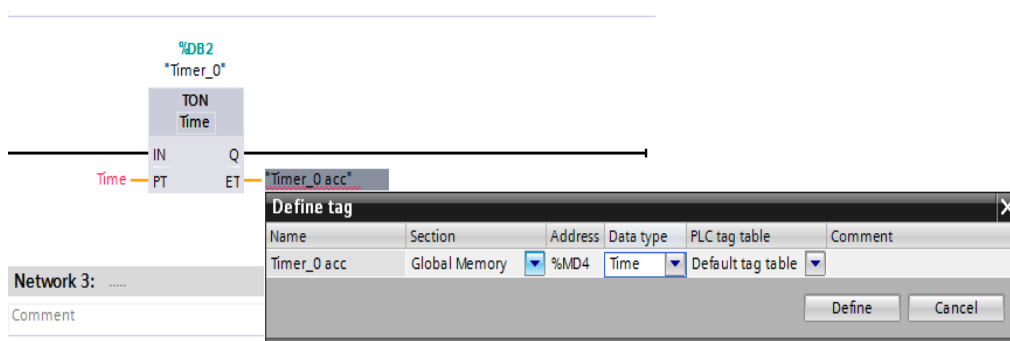


Εικόνα 129: Timer On Delay.

Πίνακας 5: Στοιχεία χρονικών μεταβλητών του Timer On Delay (TON).

Μεταβλητή	Περιγραφή
IN	Ενεργοποιεί το χρονοδιακόπτη (μεταβλητή εισόδου).
Q	Ενεργοποιείται μετά την λήξη της καθυστέρησης (μεταβλητή εξόδου).
PT	Ορίζουμε τη τιμή της χρονικής καθυστέρησης. Προσοχή στη μονάδα μέτρησης (μεταβλητή εισόδου).
ET	Η τρέχουσα τιμή μέτρησης (μεταβλητή εξόδου).

Για να ορίσουμε μία μεταβλητή (π.χ. τη μεταβλητή «ET» στη συγκεκριμένη δομή), ορίζουμε αρχικά μία αυθαίρετη ονομασία και με δεξί κλικ πάνω σε αυτή, επιλέγουμε την επιλογή «Define tag». Με τον τρόπο αυτό μπορούμε να επιλέξουμε το είδος της μεταβλητής (βλ. επόμενη παράγραφο) και να τροποποιήσουμε το όνομά της.

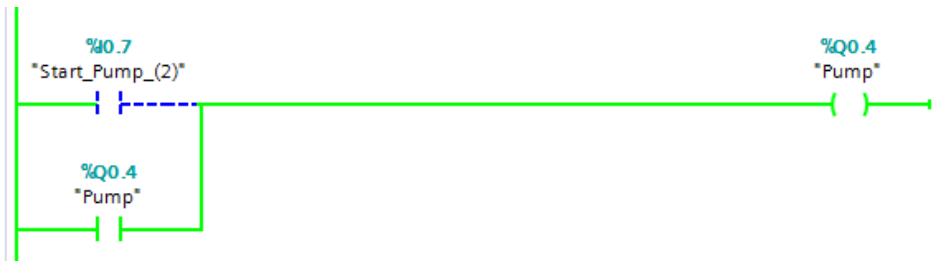


Εικόνα 130: Καθορισμός λεζάντας.

Οι τύποι που μπορεί να έχει μια μεταβλητή γενικότερα είναι τρεις. Συγκεκριμένα, οι τύποι είναι οι: «*Global Memory*» (εσωτερική μεταβλητή που αποθηκεύεται στη μνήμη του *PLC*), «*Global Input*» (μεταβλητή εισόδου - εισοδοί I0.0 - I1.5 του *PLC*), «*Global Output*» (μεταβλητή εξόδου - έξοδοι Q0.0 - Q1.1 του *PLC*).

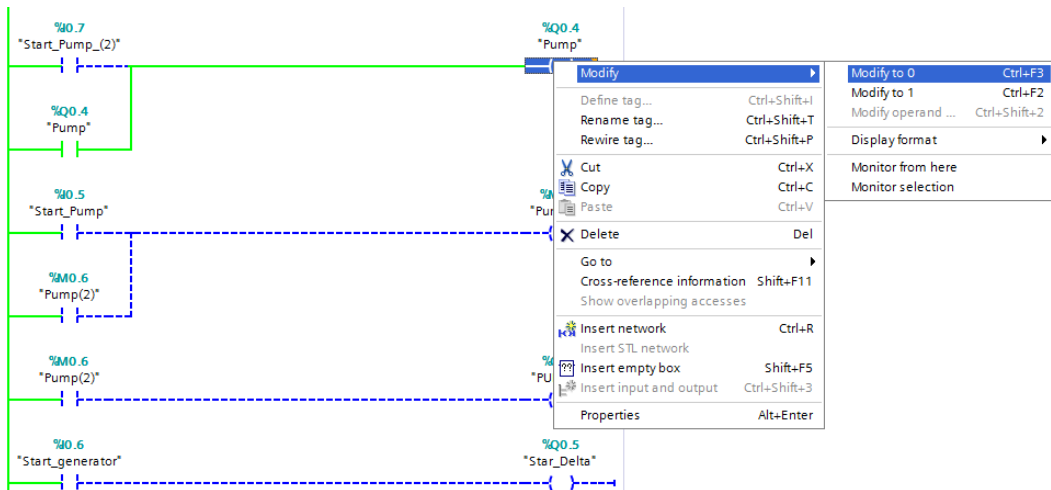
Εάν επιθυμούμε να αλλάξουμε τον τύπο μιας μεταβλητής, πάμε με δεξί κλικ πάνω σε αυτή και επιλέγουμε την αλλαγή του τύπου της μεταβλητής «*Rewire tag*».

Βήμα 5<sup>ο</sup>: Στη έκτη και έβδομη γραμμή του κώδικα έχουμε μία είσοδο (I0.7) και μια είσοδο (Q0.4) αντίστοιχα, οι οποίες είναι *normal open*. Συνήθως το τελεστή «*Q*» το συναντάμε σε εξόδους, όμως μπορούμε να ορίσουμε μια ήδη υπάρχουσα έξοδο ως είσοδο. Όταν πατηθεί το κουμπί (*Push button*), το οποίο το ορίζουμε εμείς με κατάλληλη συνδεσμολογία, ενεργοποιείτε η έξοδος (Q0.4) η οποία αντιστοιχεί στο *led* της αντλίας του δικτύου «*Pump*». Μόλις αφήσουμε το κουμπί, λόγω της εξόδου (Q0.4) (η οποία ενεργοποιήθηκε όταν ενεργοποιήθηκε για πρώτη φορά η έξοδος [Q4]), το σύστημα ανατροφοδοτείτε αυτόματα. Δηλαδή δεν σβήνει το *led*.



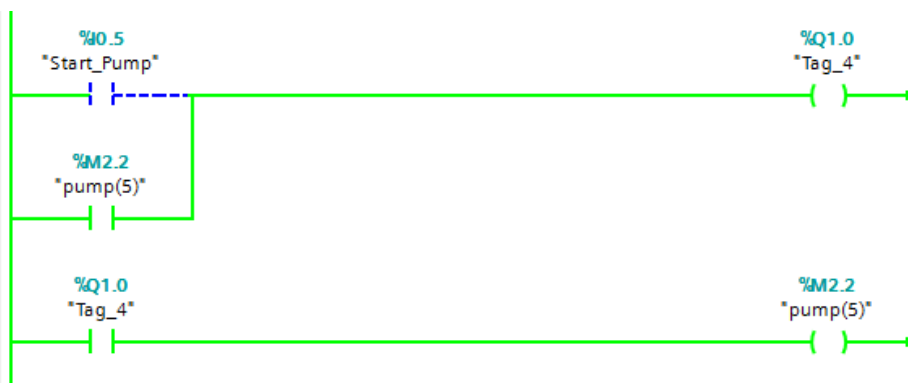
Εικόνα 131: Σύστημα ανατροφοδότησης.

Για να σβήσουμε το *led*, στο κώδικα που φτιάξαμε, πατάμε δεξί κλικ πάνω στην έξοδο (Q4) και επιλέγουμε την εντολή «*Modify*» και στη συνέχεια την εντολή «*Modify to 0*». Επειδή η έξοδος είναι *normal open*, δηλαδή με τη διέλευση σήματος ενεργοποιείτε και γίνεται αληθής (1). Επομένως εμείς τροποποιούμε το σήμα κάνοντάς το ψευδές (0), με αποτέλεσμα το σβήσιμο του συστήματος.



Εικόνα 132: Χρήση της εντολής Modify.

Βήμα 6<sup>ο</sup>: Σε αυτό το βήμα για να αναπτυχθεί πρόγραμμα ενεργοποίησης του *led* της αντλίας του δικτύου «*Pump*», με χρήση ανατροφοδότησης και εσωτερικής μνήμης, χρειάζεται να χρησιμοποιήσουμε κώδικα με έκταση 3<sup>ων</sup> γραμμών. Στην όγδοη γραμμή του κώδικα έχουμε μία είσοδο (I0.5). Όταν ενεργοποιηθεί η είσοδος (I0.5) τότε ενεργοποιείτε η έξοδος (Q1.0). Επιπλέον στη 3<sup>η</sup> γραμμή του 6<sup>ου</sup> βήματος, όταν ενεργοποιηθεί η Q1.0, τότε ταυτόχρονα ενεργοποιείτε μια έξοδος (M2.2), η οποία είναι «*Global Memory*», δηλαδή διαθέτει μια εσωτερική μνήμη του τελευταίου σήματος του αντίστοιχου τελεστή. Συνήθως το τελεστή «*M*» το συναντάμε σε κατηγορίες τελεστών που έχουν την ιδιότητα εσωτερικής μνήμης (*Global memory*). Παρατηρούμε ότι ο τελεστής «*M2.2*» εμφανίζεται ως είσοδος και στην ένατη γραμμή του κώδικα. Αυτό διότι όταν ενεργοποιηθεί η είσοδος (I0.5), μέσω ενός κουμπιού, τότε ενεργοποιείτε η έξοδος (Q1.0). Με τη σειρά της η ενεργοποίηση της (Q1.0), η οποία αντιστοιχεί στο *led* της αντλίας του δικτύου «*Pump*», ενεργοποιεί τη M1 (Βλέπε πίνακα 4). Επειδή θέλουμε και σύστημα ανατροφοδότησης, ορίζουμε την τιμή M1 ως είσοδο η οποία συνδέεται με την όγδοη γραμμή, έτσι εάν σβήσουμε την είσοδο (I0.5), τότε λόγω ανατροφοδότησης μέσω της M1 το *led* δεν θα σβήσει.



Εικόνα 133: Κώδικας εσωτερικής μνήμης και ανατροφοδότησης συστήματος.

Βήμα 7<sup>ο</sup>: Στη τελευταία γραμμή του κώδικα έχουμε μια εισόδους (I0.6) η οποία είναι *normal open* και αντιστοιχεί σε ένα διακόπτη (το οποίο το ορίζουμε εμείς με κατάλληλη συνδεσμολογία). Με την ενεργοποίηση του συστήματος ενεργοποιείτε μια έξοδος (Q0.5) η οποία αντιστοιχεί σε μια από τις δύο πύλες του μηχανισμού κίνησης ενός ηλεκτρονόμου (A1 ή A2) αλλάζοντας τη μανδάλωση του ηλεκτρονόμου από *normal close* (πριν ανοίξουμε το διακόπτη) σε *normal open* (αφού ανοίξουμε το διακόπτη). Επομένως πριν ανοίξουμε το διακόπτη έχουμε ενεργοποιημένο το *led* το οποίο αντιστοιχεί στη συνδεσμολογία «*Star*», ενώ όταν ανοίξουμε το διακόπτη έχουμε ενεργοποιημένο το *led* το οποίο αντιστοιχεί στη συνδεσμολογία «*Delta*».



Εικόνα 134: Κώδικας για χρήση ενός ηλεκτρονόμου.

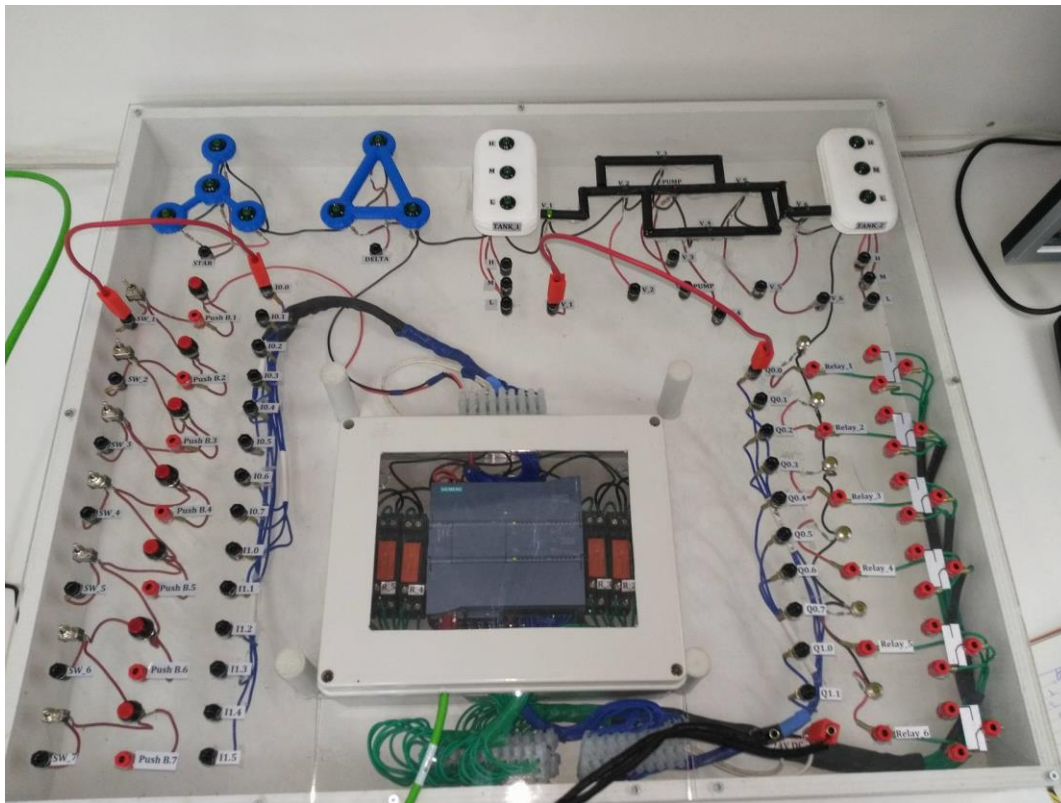
## 12.2.2 Συνδεσμολογία εκπαιδευτικής μονάδας

Πέρα από το σωστό προγραμματισμό μιας μονάδας *PLC*, εξίσου σημαντική είναι και η κατάλληλη συνδεσμολογία του προγραμματιζόμενου λογικού ελεγκτή ώστε να μπορέσει να λειτουργήσει το σύστημά μας με τον επιθυμητό τρόπο. Στην υποενότητα αυτή θα γίνει αναφορά στη σωστή συνδεσμολογία της εκπαιδευτικής μονάδας σύμφωνα με τον κώδικα *Ladder* τον οποίο αναλύσαμε προηγουμένως.

Να υπενθυμίσουμε τη πρώτη σημείωση της άσκησης, ότι οι ακροδέκτες εισόδου (*inputs*) συνδέονται με αισθητήρες (*sensors*), διακόπτες (*switch*), κουμπιά (*buttons*). Ενώ οι ακροδέκτες εξόδου (*outputs*) συνδέονται με πηνία, ηλεκτρομαγνητικές βαλβίδες (*valve*), αντλίες (*pump*), λυχνίες ένδειξης (*led*).

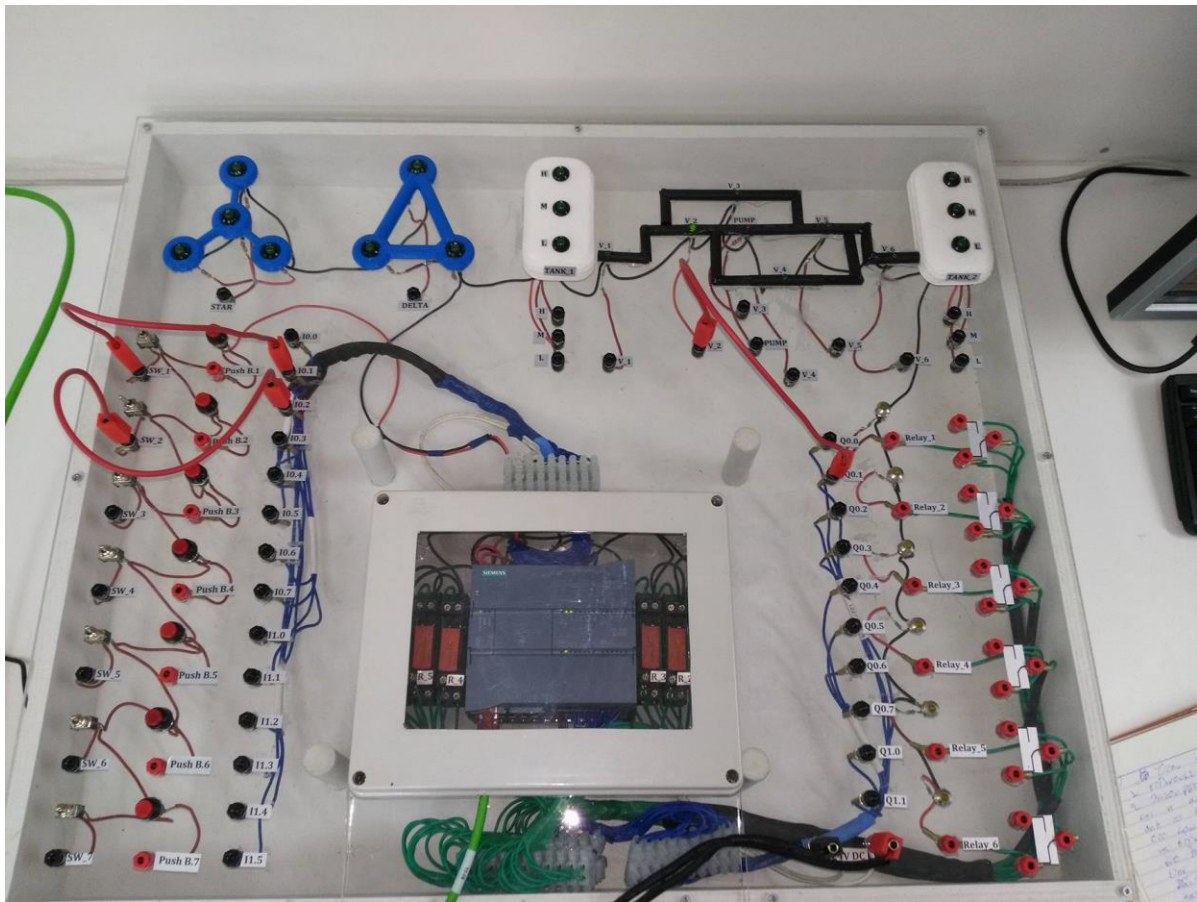
Βήμα 1<sup>ο</sup>: Σκοπός αυτού του βήματος ήταν να ενεργοποιηθεί το *led* της βαλβίδας του δικτύου «V1» χρησιμοποιώντας μια είσοδο, όποια επιθυμούσαμε, του *PLC*. Στον κώδικα ορίσαμε ως είσοδο την I0.0 και ως έξοδο την Q0.0. Σύμφωνα με τη προηγούμενη παράγραφο θα συνδέσουμε τη μπόρνα της εισόδου I0.0 με ένα διακόπτη (SW\_1), ενώ τη μπόρνα της εξόδου με τη μπόρνα που αντιστοιχεί στο *led* «V1». Όταν θα ενεργοποιήσουμε το διακόπτη θα ανάψει το *led* (βλ. εικόνα 135).





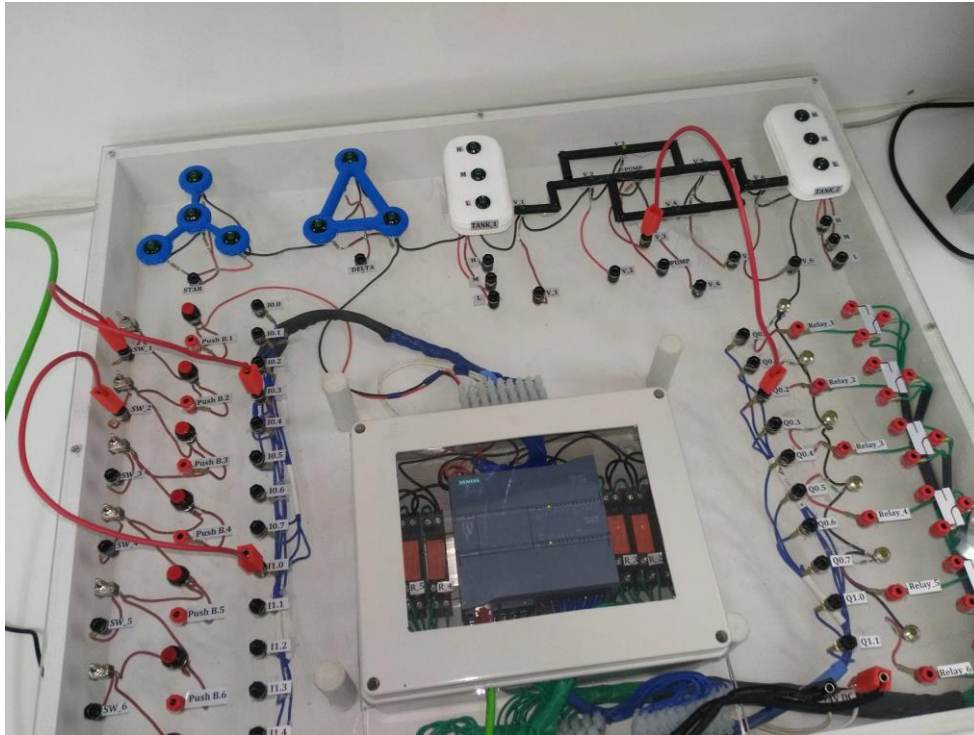
Εικόνα 135: Ενεργοποίηση εισόδου I0.0.

Βήμα 2<sup>ο</sup>: Σκοπός αυτού του βήματος ήταν να ενεργοποιηθεί το *led* της βαλβίδας του δικτύου «V2» χρησιμοποιώντας ταυτόχρονα δύο εισόδους, όποιες επιθυμούσαμε, του *PLC*. Στον κώδικα ορίσαμε ως εισόδους τις I0.1 και I0.2 ενώ ως έξοδο την Q0.1. Σύμφωνα με τη προηγούμενη παράγραφο θα συνδέσουμε κάθε είσοδο με ένα διακόπτη (SW\_1 και SW\_2 αντίστοιχα), ενώ τη μπόρνα της εξόδου με τη μπόρνα που αντιστοιχεί στο *led* «V2». Όταν θα ενεργοποιήσουμε και τους δύο διακόπτες θα ανάψει το *led* (βλ. εικόνα 136).

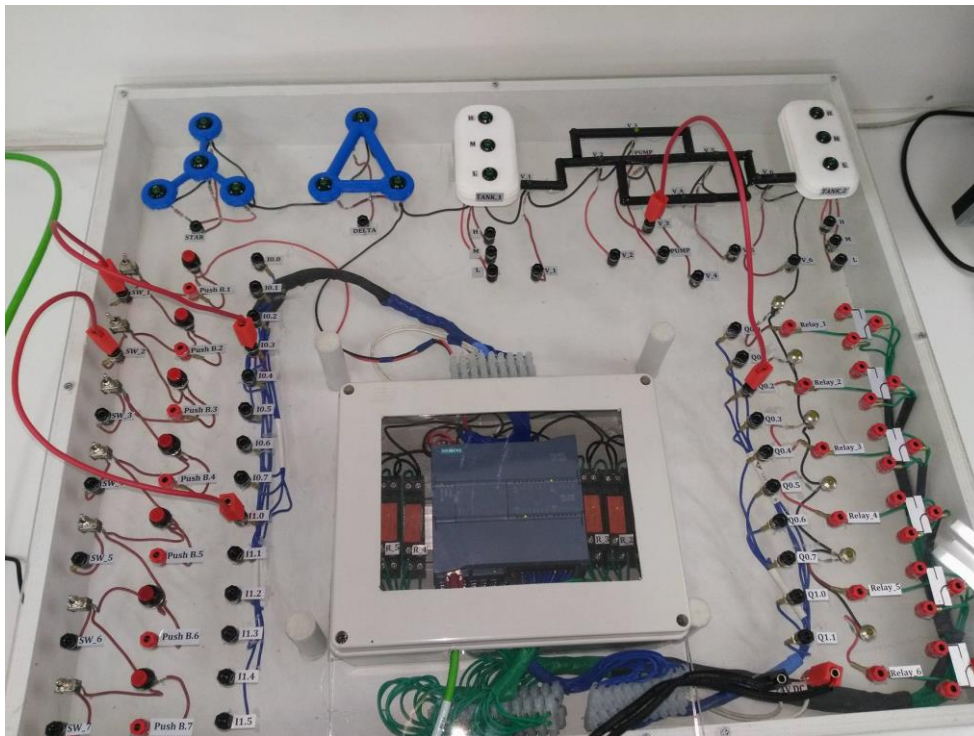


Εικόνα 136: Ενεργοποίηση και των δύο διακοπών για τη λειτουργία του συστήματος.

Βήμα 3<sup>ο</sup>: Σε αυτό το βήμα θέλουμε να ενεργοποιηθεί το *led* της βαλβίδας του δικτύου «V3» χρησιμοποιώντας δύο εισόδους, όποιες επιθυμούσαμε, του *PLC*, εναλλάξ. Στον κώδικα ορίσαμε ως εισόδους τις I0.3 και I1.0 ενώ ως έξοδο την Q0.2. Σύμφωνα με τη σημείωση στην αρχή του κεφαλαίου, θα συνδέσουμε κάθε είσοδο με ένα διακόπτη (SW\_1 και SW\_2 αντίστοιχα), ενώ τη μπόρνα της εξόδου με τη μπόρνα που αντιστοιχεί στο *led* «V3». Όταν θα ενεργοποιήσουμε έστω έναν από τους δύο διακόπτες θα ανάψει το *led* (βλ. εικόνα 137 και 138).



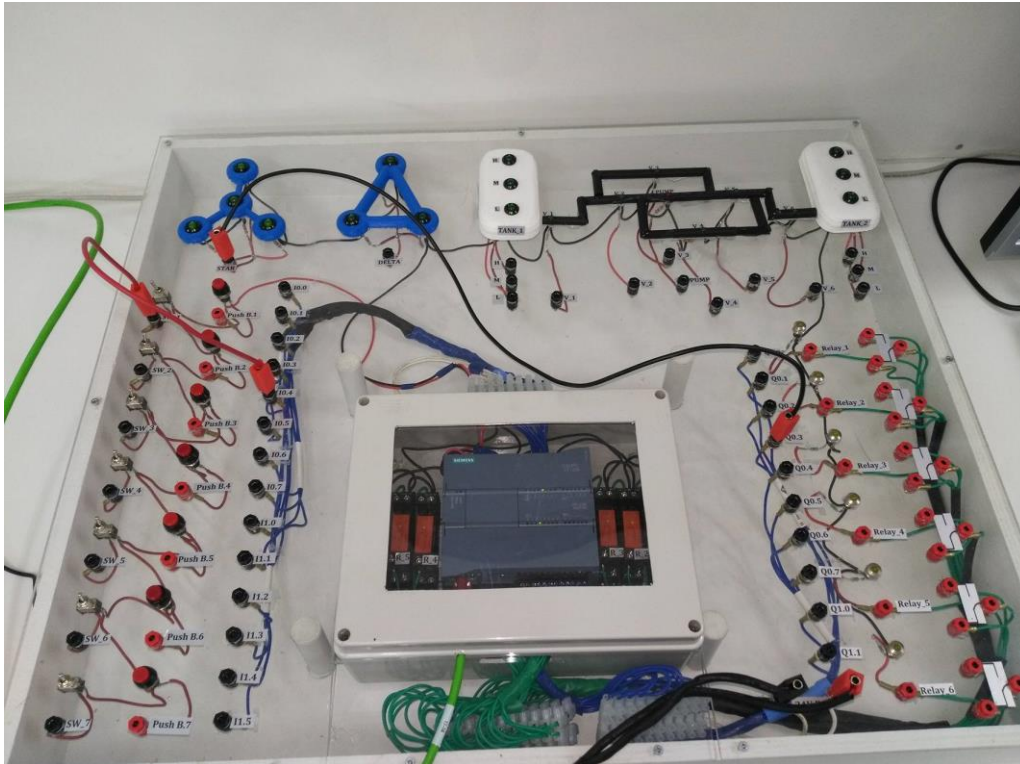
Εικόνα 137: Ενεργοποίηση μόνο της εισόδου I0.3.



Εικόνα 138: Ενεργοποίηση μόνο της εισόδου I1.0.

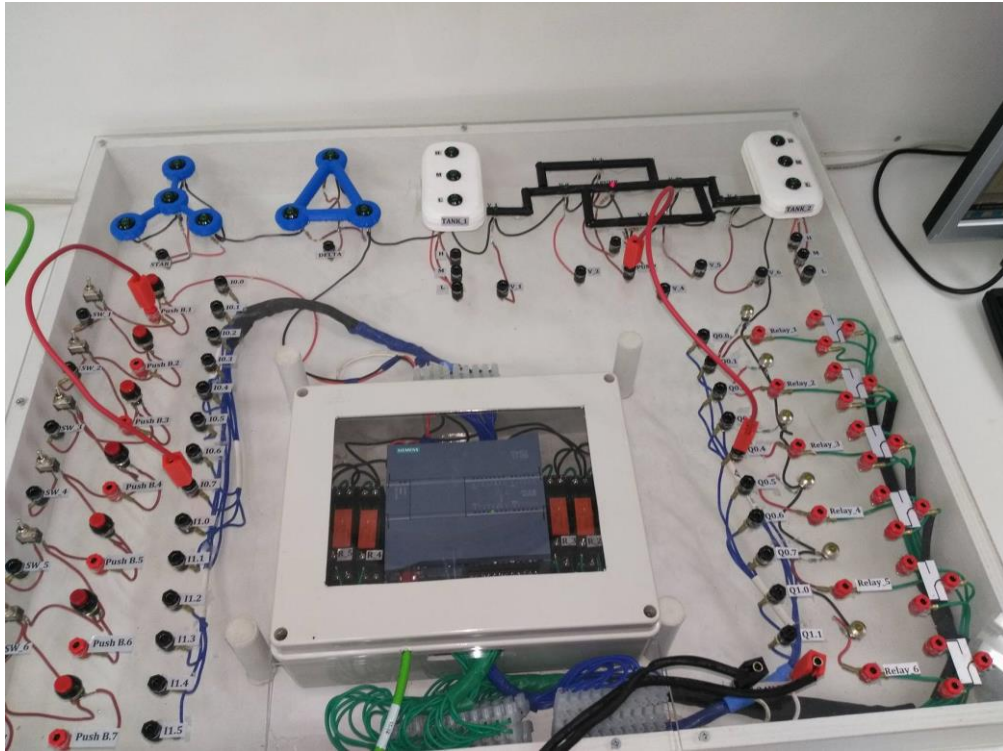
Βήμα 4<sup>ο</sup>: Σε αυτό το βήμα έχουμε ορίσει έναν *Timer*, στο κώδικα, με σκοπό την ενεργοποίηση του *led* της συνδεσμολογίας αστέρα (*Star*) 5 δευτέρα μετά την ενεργοποίηση μιας

οποιασδήποτε εισόδου. Στον κώδικα ορίσαμε ως είσοδο την I0.4 και ως έξοδο την Q0.3. Επομένως συνδέσαμε τη μπόρνα της εισόδου I0.4 με ένα διακόπτη (SW\_1), ενώ τη μπόρνα της εξόδου με τη μπόρνα που αντιστοιχεί στο *led* «Star». Όταν θα ενεργοποιήσουμε το διακόπτη, μετά από 5 δευτερόλεπτα θα ανάψει το *led* (βλ. εικόνα 139).



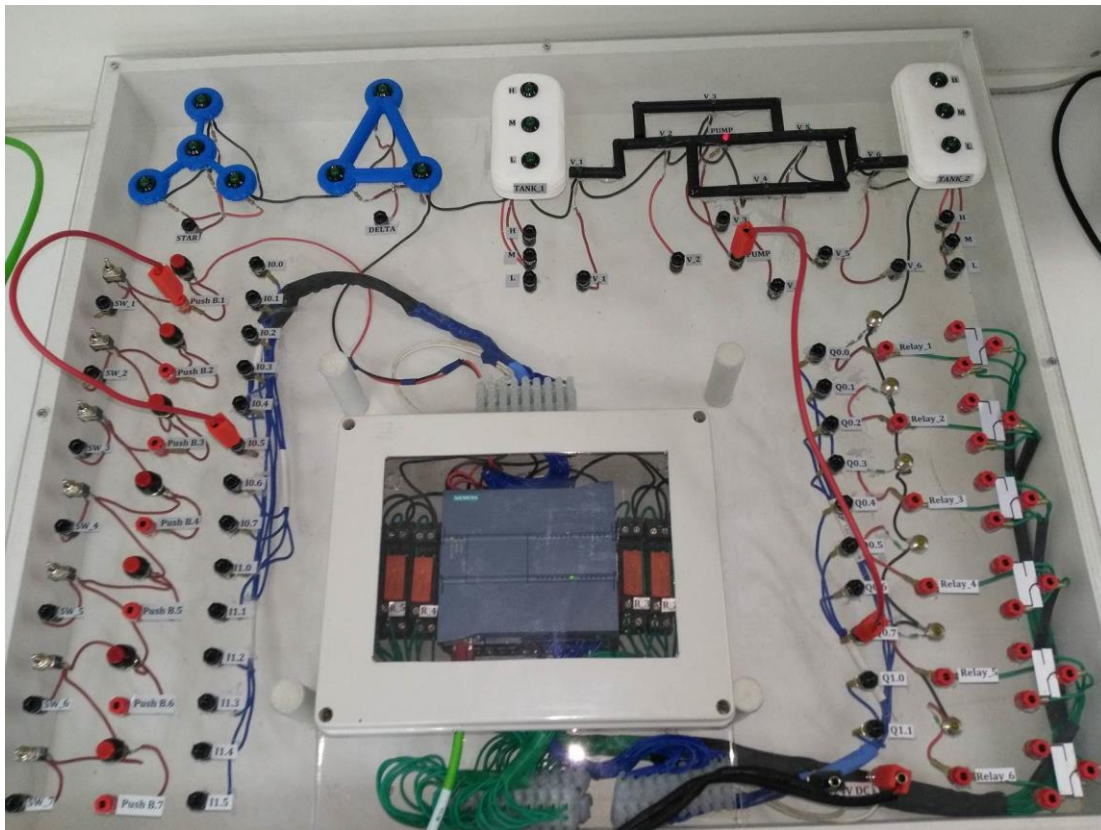
Εικόνα 139: Ενεργοποίηση led αστέρα μετά από το πέρασμα 5 δευτερολέπτων.

Βήμα 5<sup>ο</sup>: Σε αυτό το βήμα έχουμε σύστημα ανατροφοδότησης. Για να μπορέσουμε να το διακρίνουμε έχουμε συνδέσει την είσοδο I0.7 (την οποία την ορίσαμε εμείς στον κώδικα) με ένα κουμπί *push button* (Push B.1). Ενώ την έξοδο Q0.4 (την οποία την ορίσαμε εμείς στον κώδικα) τη συνδέσαμε με το *led* της αντλίας (*Pump*). Λόγο του συστήματος ανατροφοδότησης δεν χρειάζεται να κρατήσουμε πατημένο το κουμπί για να ανάψει το *led*, αρκεί μια φορά ώστε να στείλει το σήμα, μετά έχουμε ανατροφοδότηση του σήματος (βλ. εικόνα 140).



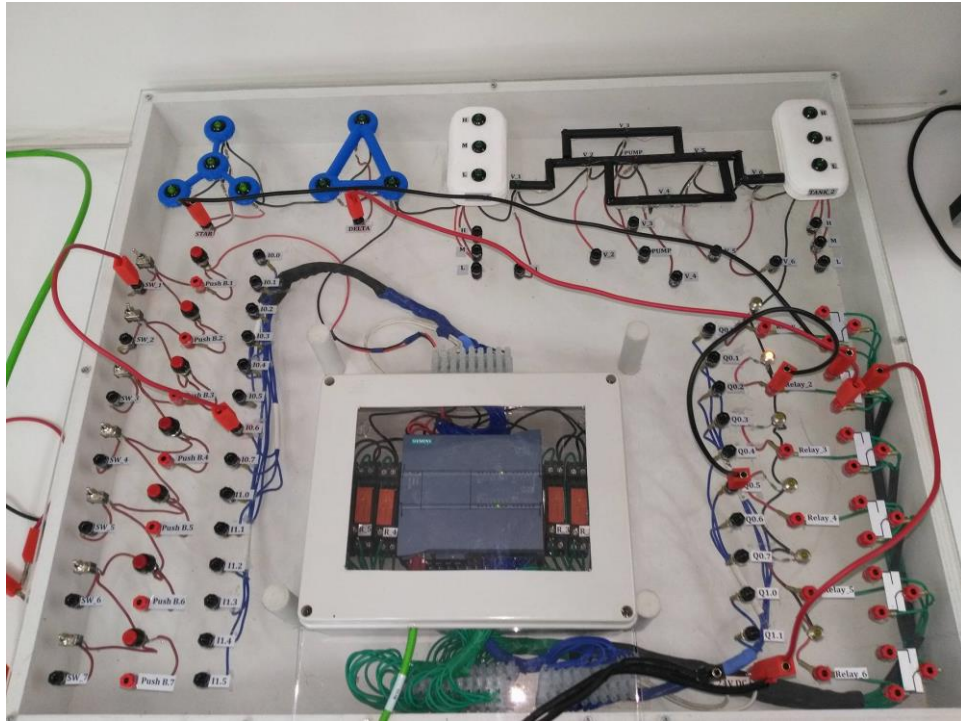
Εικόνα 140: Σύστημα ανατροφοδότησης.

Βήμα 6<sup>ο</sup>: Σε αυτό το βήμα έχουμε σύστημα ανατροφοδότησης και εσωτερικής μνήμης. Για να μπορέσουμε να το διακρίνουμε έχουμε συνδέσει την είσοδο I0.5 (την οποία την ορίσαμε εμείς στον κώδικα) με ένα κουμπί *push button* (Push B.1). Ενώ την έξοδο Q0.7 (την οποία την ορίσαμε εμείς στον κώδικα) τη συνδέσαμε με το *led* της αντλίας (*Pump*). Λόγο του συστήματος ανατροφοδότησης δεν χρειάζεται να κρατήσουμε πατημένο το κουμπί για να ανάβει το *led*, αρκεί μια φορά ώστε να στείλει το σήμα, μετά έχουμε διατήρηση του σήματος λόγω εσωτερικής μνήμης (βλ. εικόνα 141).

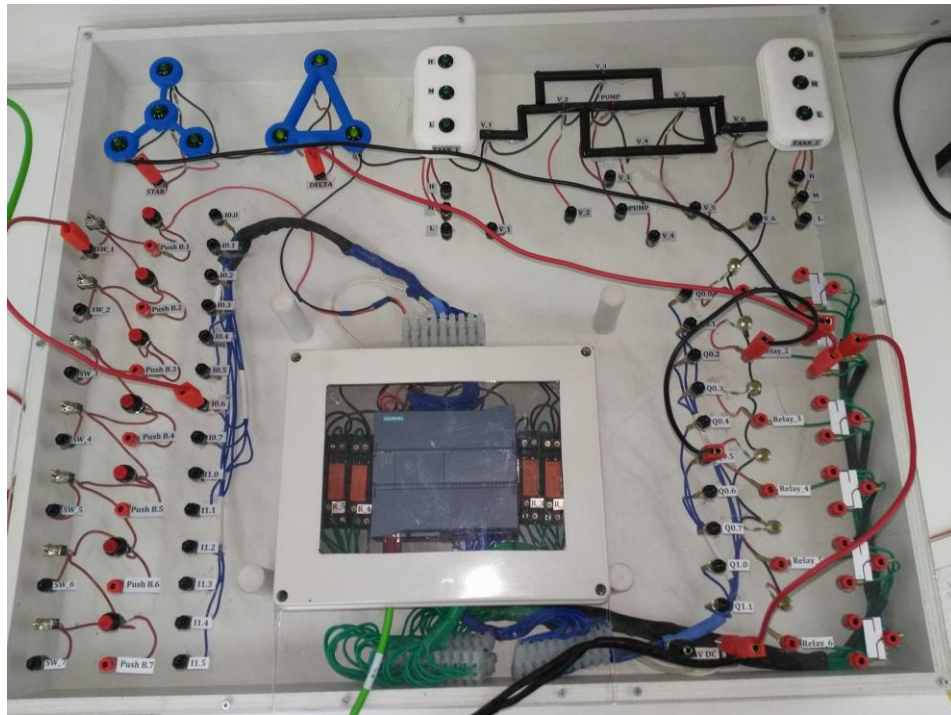


Εικόνα 141: Εσωτερική μνήμη και σύστημα ανατροφοδότησης.

Βήμα 7<sup>ο</sup>: Τέλος έχουμε χρήση ενός ηλεκτρονόμου (*Relay*) με σκοπό την εναλλαγή ενεργοποίησης των *led* «Star» και «Delta». Αρχικά χωρίς μανδάλωση του ηλεκτρονόμου ενεργοποιείτε το *led* «Star» ενώ όταν ακολουθήσει μανδάλωση του ηλεκτρονόμου (βλέπε κεφ.9<sup>ο</sup>) να έχει ως αποτέλεσμα την ενεργοποίηση του *led* «Delta» και αντίστοιχα την απενεργοποίηση του «Star». Αρχικά έχουμε συνδέσει την είσοδο I0.6 (την οποία την ορίσαμε εμείς στον κώδικα) με ένα διακόπτη (SW\_1). Ενώ την έξοδο Q0.5 (την οποία την ορίσαμε εμείς στον κώδικα) τη συνδέσαμε με τη μπόρνα που αντιστοιχεί στο μηχανισμό κίνησης του ηλεκτρονόμου (A1). Τη μπόρνα που αντιστοιχεί στη *normal close* (12) θύρα του ηλεκτρονόμου, τη συνδέσαμε με το *led* του αστερά και τη *normal open* (14) θύρα του ηλεκτρονόμου, τη συνδέσαμε με το *led* του τριγώνου. Τέλος τη μπόρνα, η οποία αντιστοιχεί στη μεταγωγική επαφή (11) τη συνδέσαμε με τη μπόρνα τροφοδοσίας. Έτσι όσο ο διακόπτης είναι κλειστός, ανάβει το *led* που αντιστοιχεί στη συνδεσμολογία αστερά (βλ. εικόνα 142). Μόλις ενεργοποιήσουμε το διακόπτη τότε αλλάζουμε τη μανδάλωση του ηλεκτρονόμου από *normal close* σε *normal open* με αποτέλεσμα την εναλλαγή των *led* από αστερά σε τρίγωνο (βλ. εικόνα 143).



Εικόνα 142: : Λειτουργία συνδεσμολογίας αστέρα με χρήση ηλεκτρονόμου.



Εικόνα 143: Λειτουργία συνδεσμολογίας τρίγωνο με χρήση ηλεκτρονόμου.

## 12.3 Άσκηση 2: Έλεγχος στάθμης και μεταφοράς ρευστού σε δεξαμενές

Η παρούσα άσκηση αποσκοπεί στη κατανόηση ελέγχου στάθμης σε δεξαμενές και μεταφορά ρευστού από μια δεξαμενή σε άλλη με χρήση της εκπαιδευτικής μονάδας. Επιπλέον έχει στόχο την εξοικείωση του φοιτητή με το περιβάλλον *TIA Portal* και τη γλώσσα προγραμματισμού *Ladder* για την επίτευξη του επιθυμητού αποτελέσματος.

Να πραγματοποιηθούν τα παρακάτω βήματα, με χρήση της γλώσσας προγραμματισμού *Ladder*, έτσι ώστε με τη κατάλληλη συνδεσμολογία της εκπαιδευτικής μονάδας να λειτουργεί το πρόγραμμα.

1. Μεταφορά ρευστού από την αριστερή δεξαμενή (*Tank\_1*) στη δεξιά δεξαμενή (*Tank\_2*) χρησιμοποιώντας τις κατάλληλες – σωστές ηλεκτροβάνες (βλ. εικόνα 103 και 104 για επιλογή διαδρομής της ροής).
2. Ροή ρευστού από τη δεξαμενή 1 (*Tank\_1*) στη δεξαμενή 2 (*Tank\_2*) και αντίστροφα με χρήση διακόπτη (τύπου *SW*) για την εναλλαγή της ροής.
3. Όταν η δεξαμενή 1 είναι γεμάτη (επίπεδο **Hight** – *led* ενεργοποιημένο) και η στάθμη της δεξαμενής 2 είναι χαμηλή (επίπεδο **Low** – *led* ενεργοποιημένο), να ενεργοποιούνται κατάλληλα οι ηλεκτροβάνες για μεταφορά έρματος από τη δεξαμενή 1 στη δεξαμενή 2, για χρόνο **8 δευτερόλεπτα**. Μετά τη μεταφορά να μείνουν κλειστές/απενεργοποιηθούν οι ηλεκτροβάνες και η αντλία.

### **ΠΡΟΣΟΧΗ ΣΤΗ ΣΥΝΔΕΣΗ ΤΟΥ ΤΡΟΦΟΔΟΤΙΚΟΥ ΜΕ ΤΗ ΜΟΝΑΔΑ.**

#### ΣΗΜΕΙΩΣΗ:

- Κάθε φορά που δημιουργείται ένα νέο *project* ή διορθώνεται ένα ήδη υπάρχον πρόγραμμα, μέσω του προγράμματος *TIA Portal*, για να το αποθηκεύσουμε από τον υπολογιστή στη μονάδα *PLC* με σκοπό να τεθεί σε λειτουργία, χρειάζεται να επιλέξουμε την εντολή «*Download to Device*» και στη συνέχεια να επιλέξουμε την εντολή «*Start CPU*».
- Στον κώδικα η κάθε έξοδος (τύπου *Global Output* ή *Global Memory*) πρέπει να ορίζεται μόνο μια φορά (Δεν υπάρχει πρόβλημα εάν η ίδια χρησιμοποιείται ως είσοδος/συνθήκη για ορισμό άλλης εξόδου).
- Πριν προχωρήσουμε σε οποιαδήποτε αλλαγή σε υπάρχον πρόγραμμα, πρώτα θέτουμε σε κατάσταση αδράνειας τη μονάδα ΠΛΕ (επιλέγουμε την εντολή «*Stop CPU*»).



- Όταν αρχικά έχουμε επιλέξει την κατάλληλη μονάδα *PLC* να συνδεθεί με τον υπολογιστή, επιλέγουμε την εντολή «*Flash led*» για να βεβαιωθούμε ότι η διασύνδεση είναι επιτυχής (παρατηρήστε τις οπτικές ενδείξεις στη μονάδα *PLC*).
- Με χρήση της εντολής «*Monitoring on/off*» η οποία βρίσκεται στη γραμμή εργασιών στο παράθυρο «*Main*», μπορούμε να παρατηρούμε σε πραγματικό χρόνο την εκτέλεση του προγράμματος που «τρέχει» εκείνη τη στιγμή.

Παρακάτω ακολουθεί ο κώδικας σε γλώσσα *Ladder* των παραπάνω τριών βημάτων, η εξήγηση του προγράμματος, καθώς και η σωστή συνδεσμολογία της μονάδας για την επίτευξη του επιθυμητού αποτελέσματος.

### 12.3.1 Επεξήγηση ενδεικτικού κώδικα

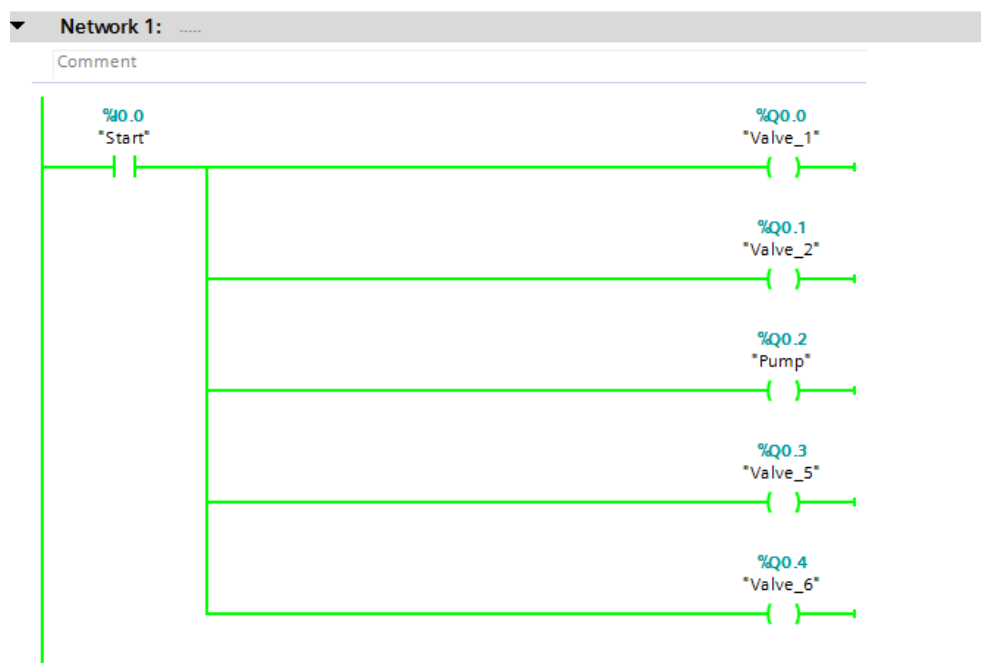
Σε αυτή την υποενότητα θα γίνει αναφορά στον κώδικα που κατασκευάστηκε με σκοπό τη λύση των τριών βημάτων που αναφέρθηκαν παραπάνω. Αρχικά να σημειωθεί ότι όταν κατασκευάζουμε ένα κώδικα προσέχουμε να μην έχουμε τον ίδιο τελεστή εξόδου σε περισσότερες από μια γραμμές. Για τις εισόδους δεν έχουμε αυτό τον περιορισμό, εφόσον εξαρτάται από τη συνδεσμολογία που έχουμε κάνει. Για την καλύτερη κατανόηση του κώδικα χωρίσαμε την άσκηση σε τρία μέρη, ένα για κάθε βήμα.

Βήμα 1<sup>ο</sup>: Το πρώτο βήμα αποτελείται από κώδικα πέντε γραμμών. Στόχος αυτού του βήματος είναι η μεταφορά ρευστού από την αριστερή στη δεξιά δεξαμενή. Σύμφωνα με την ενότητα «Προϋπάρχουσα διάταξη δικτύου δεξαμενών», στην οποία έγινε αναφορά στη χρήση των ηλεκτοβανών – βαλβίδων που χρησιμοποιούμε όταν γίνεται μεταφορά κάποιου ρευστού από τη μια δεξαμενή στην άλλη, θα «ανοίξουμε» τις βαλβίδες 1, 2, 5 και 6 καθώς και την αντλία του συστήματος (βλ. πίνακα 6).

Πίνακας 6: Σύμβολα, τελεστές, ονομασίες του κώδικα και η εξήγησή τους.

ΣΥΜΒΟΛΟ	ΤΕΛΕΣΤΗΣ	ΟΝΟΜΑΣΙΑ	ΕΞΗΓΗΣΗ
I0	I0.0	Start	Διακόπτης: Ενεργοποιεί τη ροή του ρευστού από τα αριστερά προς τα δεξιά.
Q0	Q0.0	Valve_1	Led: Ενεργοποιεί το led της βαλβίδας (V_1).
Q1	Q0.1	Valve_2	Led: Ενεργοποιεί το led της βαλβίδας (V_2).
Q2	Q0.2	Pump	Led: Ενεργοποιεί το led της αντλίας (Pump).
Q3	Q0.3	Valve_5	Led: Ενεργοποιεί το led της βαλβίδας (V_5).
Q4	Q0.4	Valve_6	Led: Ενεργοποιεί το led της βαλβίδας (V_6).

Ο κώδικας, του πρώτου βήματος, αποτελείται από μια είσοδο (I0.0) η οποία με την ενεργοποίησή της, ενεργοποιεί τις εξόδους (Q0.0, Q0.1, Q0.2, Q0.3, Q0.4) οι οποίες αντιστοιχούν στην ενεργοποίηση των *led* των βαλβίδων και των αντλιών. Στη παρακάτω εικόνα διακρίνουμε το κώδικα αυτού το βήματος, όταν ενεργοποιήσαμε την είσοδο με όνομα *Start*.



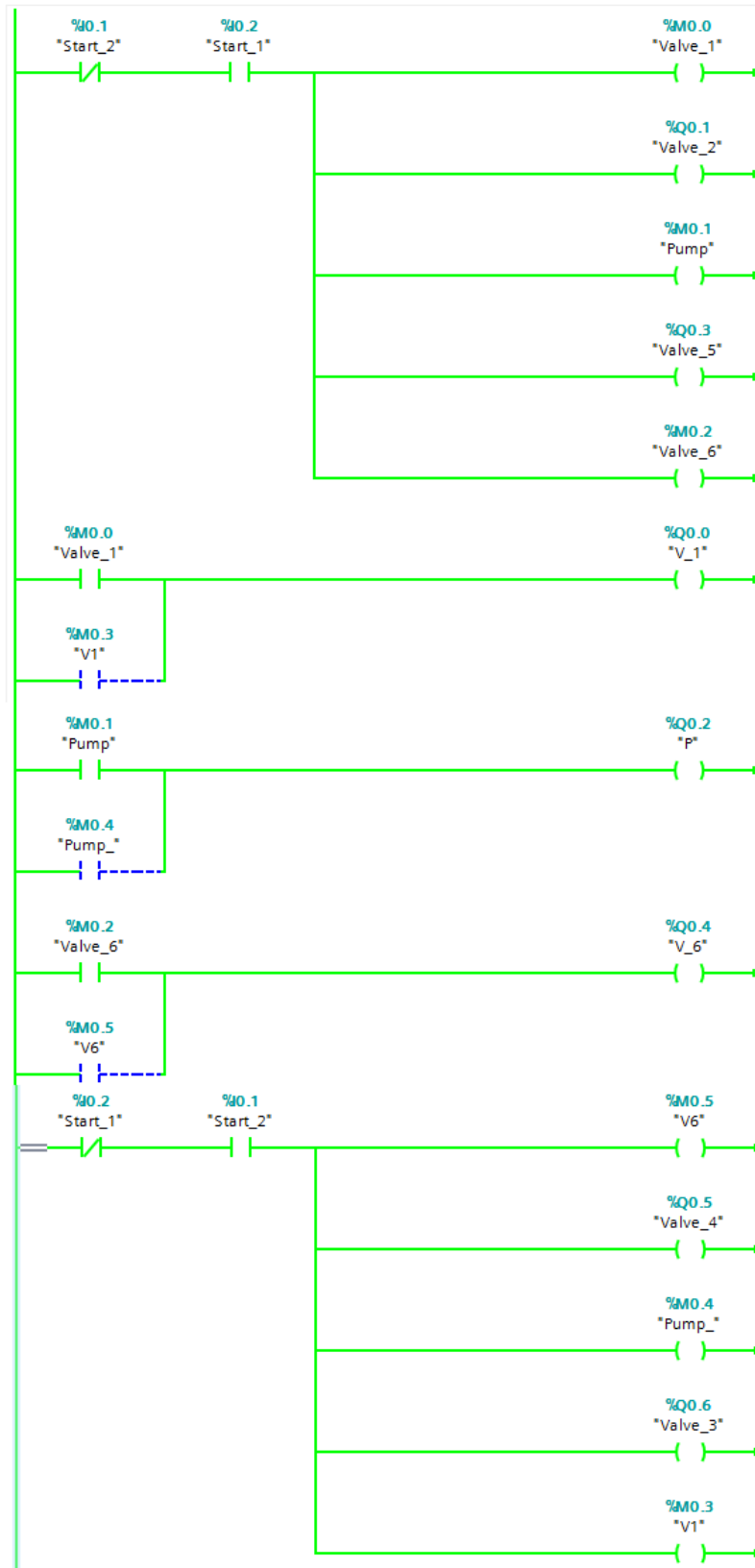
Εικόνα 144: Κώδικας για τη μεταφορά ρευστού από την αριστερή δεξαμενή προς την δεξιά.

Βήμα 2<sup>ο</sup>: Σε αυτό το βήμα φτιάχτηκε κώδικας με σκοπό τη μεταφορά του ρευστού από τη δεξιά δεξαμενή προς την αριστερή και αντίστροφα. Αρχικά προσθέσαμε δύο εισόδους, μια *normal close* (I0.0) και μια *normal open* (I0.1), οι οποίες στη δωδέκατη γραμμή του κώδικα εναλλάσσονται, δηλαδή η είσοδος I0.0 γίνεται *normal open*, ενώ η I0.1 *normal close*. Αυτό αποσκοπούσε στην αδυναμία λειτουργίας του κώδικα σε περίπτωση ενεργοποίησης και των δύο διακοπών (ταυτόχρονη ροή και από τις δύο δεξαμενές). Στο κώδικα οι πρώτες πέντε γραμμές αναφέρονται στην ενεργοποίηση της ροής από την αριστερή δεξαμενή προς τη δεξιά δεξαμενή (ενεργοποίηση βαλβίδων 1, 2, 5 και 6), ενώ οι πέντε τελευταίες γραμμές του κώδικα αναφέρονται στην ενεργοποίηση της ροής από τα δεξιά προς τα αριστερά (ενεργοποίηση

βαλβίδων 6, 4, 3 και 1). Σε κάθε μία από τις δύο περιπτώσεις λειτουργεί η αντλία. Επειδή ο ελεγκτής αδυνατεί να αναγνωρίσει εξόδους οι οποίες εμφανίζονται ξανά σε διαφορετικές γραμμές, για τις βαλβίδες 1 και 6 και για την αντλία, ορίσαμε κάποιους τελεστές «*Global Memory*», οι οποίοι ενεργοποιούν τις αντίστοιχες εξόδους.

Πίνακας 7: Σύμβολα, τελεστές, ονομασίες του κώδικα και η εξήγησή τους.

ΣΥΜΒΟΛΟ	ΤΕΛΕΣΤΗΣ	ΟΝΟΜΑΣΙΑ	ΕΞΗΓΗΣΗ
<b>I0</b>	I0.0	Start_2	Διακόπτης: Ενεργοποιεί τη ροή του ρευστού από τα δεξιά προς τα αριστερά.
<b>I1</b>	I0.1	Start_1	Διακόπτης: Ενεργοποιεί τη ροή του ρευστού από τα αριστερά προς τα δεξιά.
<b>Q0</b>	Q0.0	V_1	Led: Ενεργοποιεί το led της βαλβίδας (V_1).
<b>Q1</b>	Q0.1	Valve_2	Led: Ενεργοποιεί το led της βαλβίδας (V_2).
<b>Q2</b>	Q0.2	P	Led: Ενεργοποιεί το led της αντλίας (Pump).
<b>Q3</b>	Q0.3	Valve_5	Led: Ενεργοποιεί το led της βαλβίδας (V_5).
<b>Q4</b>	Q0.4	V_6	Led: Ενεργοποιεί το led της βαλβίδας (V_6).
<b>Q5</b>	Q0.5	Valve_4	Led: Ενεργοποιεί το led της βαλβίδας (V_4).
<b>Q6</b>	Q0.6	Valve_3	Led: Ενεργοποιεί το led της βαλβίδας (V_3).
<b>M0</b>	M0.0	Valve_1	Global Memory: Ενεργοποιεί την έξοδο Q0.
<b>M1</b>	M0.1	Pump	Global Memory: Ενεργοποιεί την έξοδο Q0.
<b>M2</b>	M0.2	Valve_6	Global Memory: Ενεργοποιεί την έξοδο Q4.
<b>M3</b>	M0.3	V_1	Global Memory: Ενεργοποιεί την έξοδο Q0.
<b>M4</b>	M0.4	Pump_	Global Memory: Ενεργοποιεί την έξοδο Q2.
<b>M5</b>	M0.5	V_6	Global Memory: Ενεργοποιεί την έξοδο Q4.



Εικόνα 145: Κώδικας για τη μεταφορά ρευστού και στις δύο δεξαμενές, εναλλάξ.

Βήμα 3<sup>ο</sup>: Σε αυτό το βήμα χρησιμοποιούμε εισόδους, οι οποίες αντιστοιχούν σε διακόπτες, για τον έλεγχο στάθμης στις δεξαμενές. Όταν η δεξαμενή 1 είναι γεμάτη (Ενεργοποίηση εισόδων I0.0, I0.1 και I0.1) ενώ στην άλλη δεξαμενή η στάθμη είναι στο χαμηλό επίπεδο

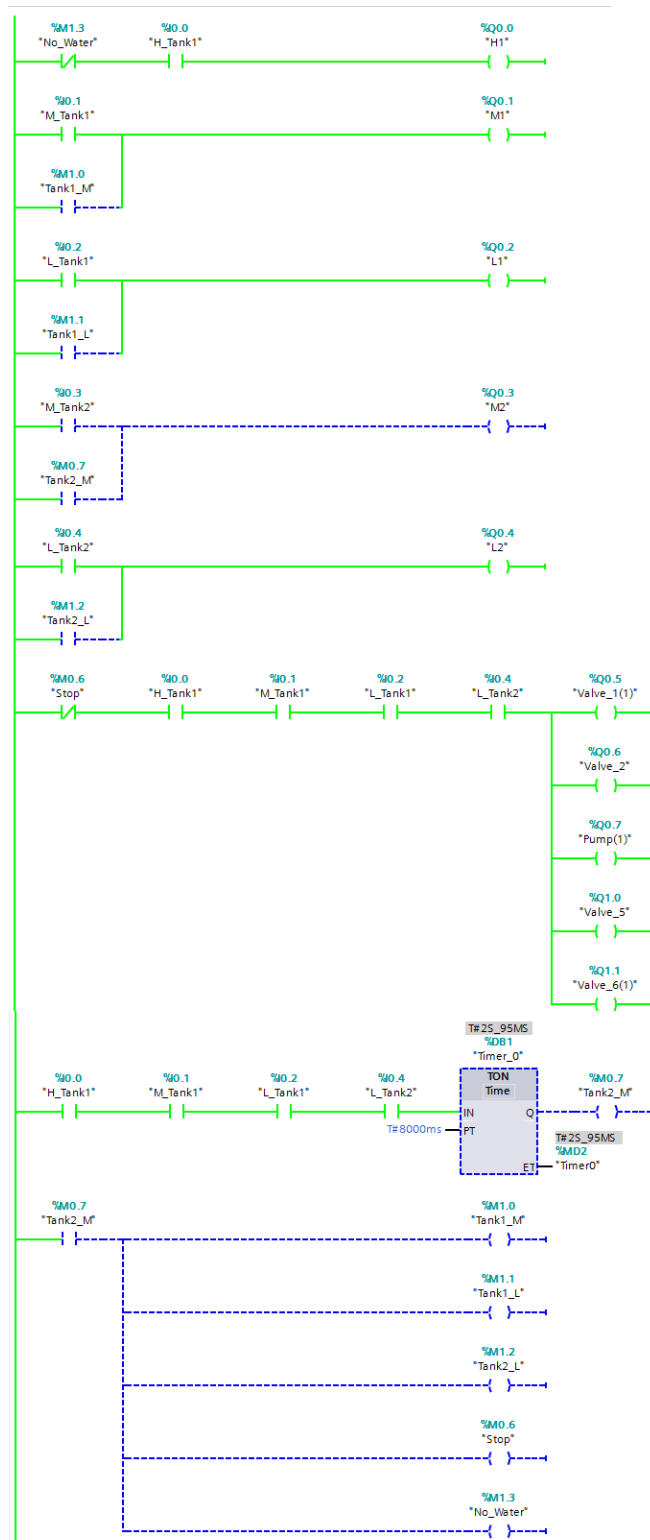
(Ενεργοποίηση εισόδου I0.4), ενεργοποιείται αυτόματα το σύστημα μεταφοράς ρευστού από τη δεξαμενή 1 στη 2, ανοίγοντας συγκεκριμένες βαλβίδες (Εξοδοι Q0.5, Q0.6, Q1.0, Q1.1) και αντλία (έξοδος Q0.7). Την ίδια στιγμή ενεργοποιείται το χρονόμετρο (TON), το οποίο σε χρόνο 8 sec. ενεργοποιεί την έξοδο M0.7.

Λόγω αδυναμίας του ελεγκτή να αναγνωρίσει εξόδους οι οποίες εμφανίζονται ξανά σε διαφορετικές γραμμές του κώδικα, ορίσαμε κάποιους τελεστές «Global Memory», οι οποίοι ενεργοποιούν κάποιες εξόδους οι οποίες χρειάστηκε να χρησιμοποιηθούν παράλληλα και σε άλλες γραμμές του κώδικα (βλ. Πίνακα 8). Τέλος, όταν περάσουν τα 8 δευτερόλεπτα τότε ο αισθητήρας του υψηλού επιπέδου της δεξαμενής σβήνει (Q0.0), εξαιτίας της μεταφοράς ρευστού στην άλλη δεξαμενή. Αυτό συμβαίνει λόγω της εναλλαγής της εισόδου M1.3, η οποία στη πρώτη γραμμή ήταν *normal close* και λόγω της αλλαγής σήματος στην εικοστή γραμμή του κώδικα, άλλαξε σε *normal open*, συγχρόνως, εφόσον είναι τελεστής μνήμης, διατηρεί τη τελευταία κατάσταση. Παρόμοια περίπτωση αποτελεί και η είσοδος M0.6, η οποία απενεργοποίησε τις βαλβίδες και την αντλία που χρησιμοποιήσαμε για τη μεταφορά ρευστού.

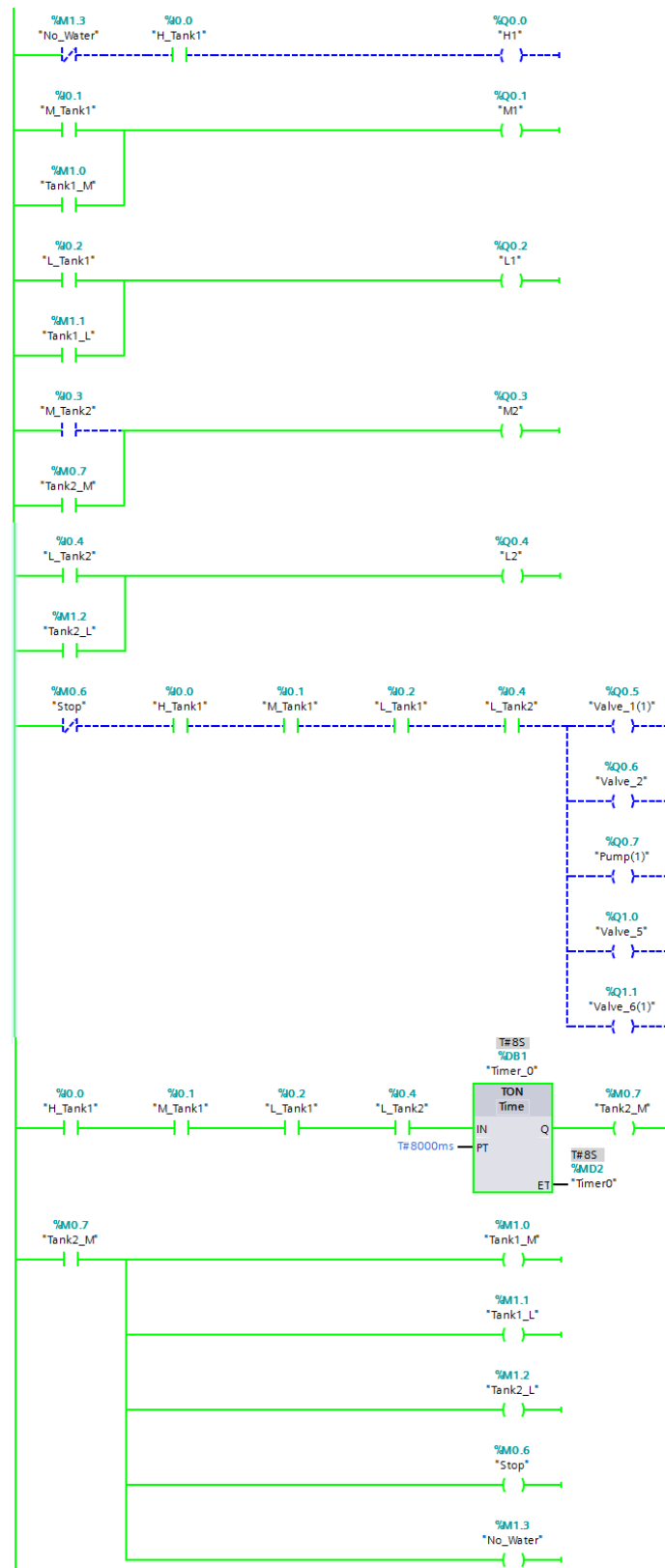
Πίνακας 8: Σύμβολα, τελεστές, ονομασίες του κώδικα και η εξήγησή τους.

ΣΥΜΒΟΛΟ	ΤΕΛΕΣΤΗΣ	ΟΝΟΜΑΣΙΑ	ΕΞΗΓΗΣΗ
I0	I0.0	H_Tank1	Διακόπτης: Ενεργοποιεί τον αισθητήρα (H) της δεξαμενής 1.
I1	I0.1	M_Tank1	Διακόπτης: Ενεργοποιεί τον αισθητήρα (M) της δεξαμενής 1.
I2	I0.2	L_Tank1	Διακόπτης: Ενεργοποιεί τον αισθητήρα (L) της δεξαμενής 1.
I3	I0.3	M_Tank2	Διακόπτης: Ενεργοποιεί τον αισθητήρα (M) της δεξαμενής 2.
I4	I0.4	L_Tank2	Διακόπτης: Ενεργοποιεί τον αισθητήρα (L) της δεξαμενής 2.
Q0	Q0.0	H1	Led: Ενεργοποιεί το led του αισθητήρα (H) της δεξαμενής 1.
Q1	Q0.1	M1	Led: Ενεργοποιεί το led του αισθητήρα (M) της δεξαμενής 1.
Q2	Q0.2	L1	Led: Ενεργοποιεί το led του αισθητήρα (L) της δεξαμενής 1.
Q3	Q0.3	M2	Led: Ενεργοποιεί το led του αισθητήρα (M) της δεξαμενής 2.
Q4	Q0.4	L2	Led: Ενεργοποιεί το led του αισθητήρα (L) της δεξαμενής 2.
Q5	Q0.5	Valve_1(1)	Led: Ενεργοποιεί το led της βαλβίδας (V_1).
Q6	Q0.6	Valve_2	Led: Ενεργοποιεί το led της βαλβίδας (V_2).
Q7	Q0.7	Pump(1)	Led: Ενεργοποιεί το led της αντλίας (Pump).
Q8	Q1.0	Valve_5	Led: Ενεργοποιεί το led της βαλβίδας (V_5).
Q9	Q1.1	Valve_6(1)	Led: Ενεργοποιεί το led της βαλβίδας (V_6).
M0	M0.6	Stop	Global Memory: Απενεργοποιεί τις εξόδους Q5 έως Q9.
M1	M0.7	Tank_2_M	Global Memory: Ενεργοποιεί την έξοδο Q3 και τις M2 έως M5 και τη M0.

<b>M2</b>	M1.0	Tank1_M	Global Memory: Ενεργοποιεί την έξοδο Q1.
<b>M3</b>	M1.1	Tank1_L	Global Memory: Ενεργοποιεί την έξοδο Q2.
<b>M4</b>	M1.2	Tank2_L	Global Memory: Ενεργοποιεί την έξοδο Q4.
<b>M5</b>	M1.3	No_Water	Global Memory: Απενεργοποιεί την έξοδο Q0.
<b>Timer</b>	Timer_0	Timer0	Χρονόμετρο: Σε χρόνο 8 sec ενεργοποιεί την έξοδο M1.



Εικόνα 146: Μεταφορά ρευστού από τη δεξαμενή ένα στη δύο.



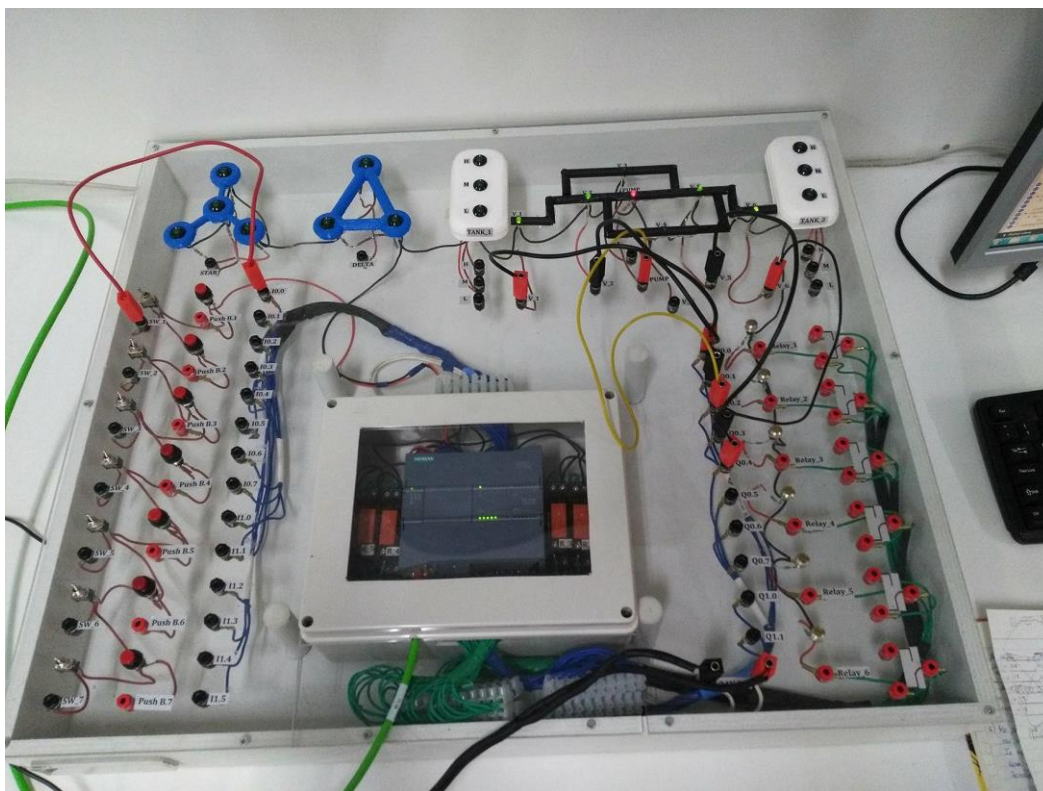
Εικόνα 147: Κώδικας μετά τη μεταφορά του ρευστού από τη δεξαμενή ένα στη δύο.



### 12.3.2 Συνδεσμολογία εκπαιδευτικής μονάδας

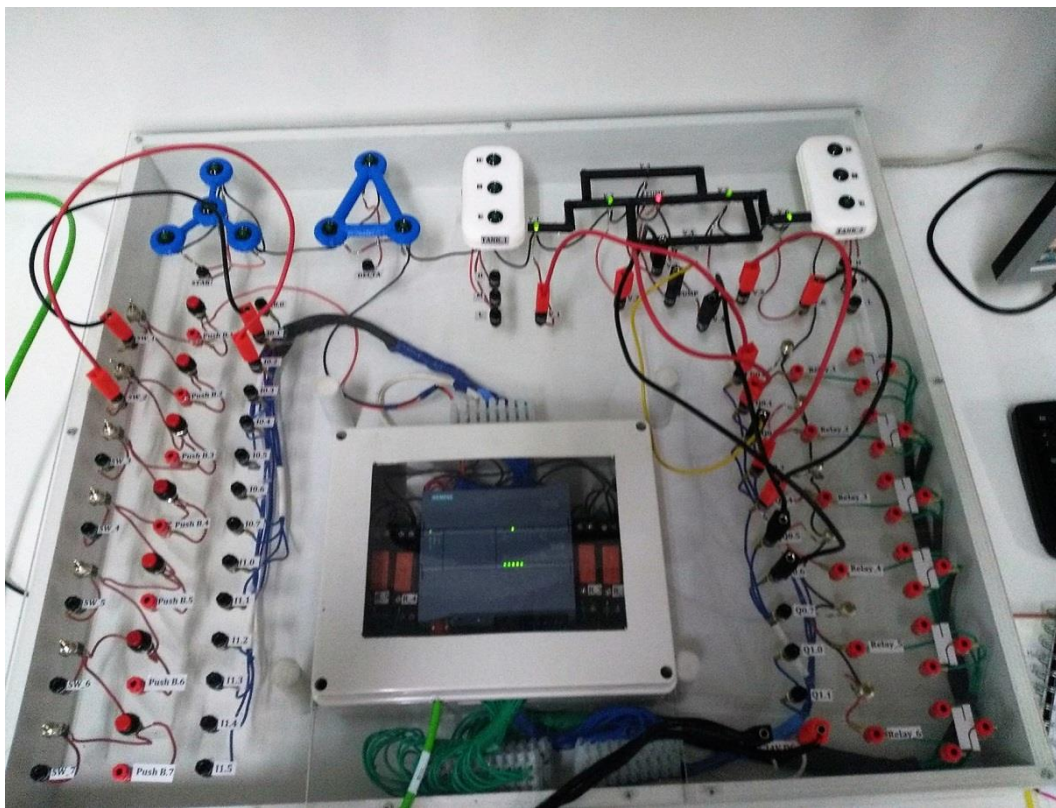
Πέρα από το σωστό προγραμματισμό μιας μονάδας *PLC*, εξίσου σημαντική είναι και η κατάλληλη συνδεσμολογία του προγραμματιζόμενου λογικού ελεγκτή, ώστε να μπορέσει να λειτουργήσει το σύστημά μας με τον επιθυμητό τρόπο. Στην υποενότητα αυτή θα γίνει αναφορά στη σωστή συνδεσμολογία της εκπαιδευτικής μονάδας σύμφωνα με τον κώδικα *Ladder* τον οποίο αναλύσαμε προηγουμένως.

Βήμα 1<sup>ο</sup>: Σκοπός αυτού του βήματος είναι η σωστή συνδεσμολογία, σύμφωνα με τον κώδικα που φτιάξαμε, ώστε να γίνει μεταφορά ρευστού από τη δεξαμενή 1 στη 2. Επομένως συνδέουμε τις βαλβίδες και την αντλία, οι οποίες συμμετέχουν στη ροή προς τα δεξιά, με τις κατάλληλες εξόδους που ορίσαμε στο πρόγραμμα δεν υπάρχει αυστηρό κριτήριο με το οποίο θα συνδέσουμε τις μπόρνες των εξόδων με τις μπόρνες των *led*, αφού η έξοδος, του κώδικα, ενεργοποιεί και τις πέντε εξόδους ταυτόχρονα. Τέλος θα συνδέσουμε την είσοδο *I0.0*, σύμφωνα με τον κώδικα, με έναν διακόπτη, ώστε να εξυπηρετεί στον έλεγχο ενεργοποίησης και απενεργοποίησης του συστήματος. Μόλις ενεργοποιήσουμε το διακόπτη, ενεργοποιούνται οι βαλβίδες και η αντλία που εξυπηρετούν τη ροή προς τα δεξιά.

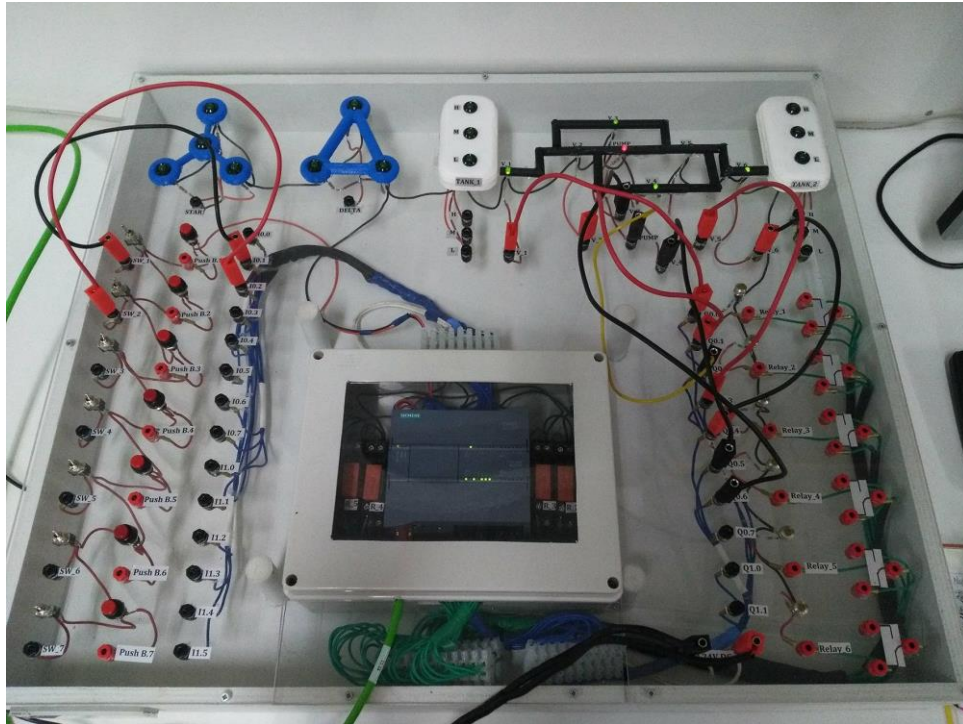


Εικόνα 148: Μεταφορά ρευστού από την αριστερή δεξαμενή (*Tank\_1*) στη δεξιά δεξαμενή (*Tank\_2*) με χρήση κατάλληλων βαλβίδων.

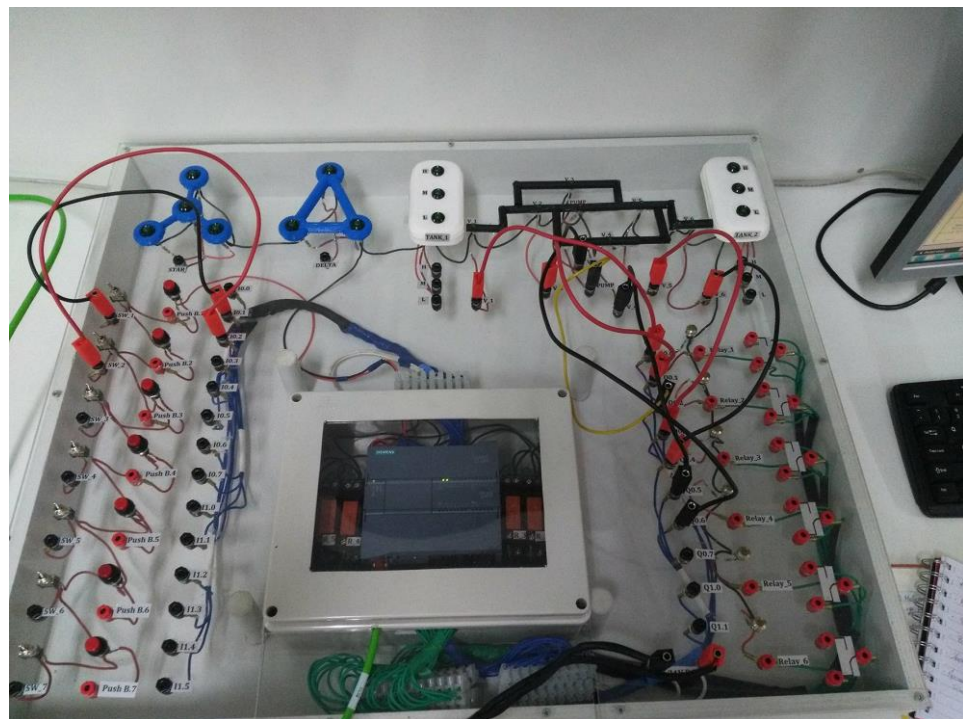
Βήμα 2<sup>ο</sup>: Σε αυτό το βήμα ο στόχος είναι η σωστή συνδεσμολογία, σύμφωνα με τον κώδικα που φτιάξαμε, ώστε να γίνει μεταφορά ρευστού από τη δεξαμενή ένα στη δεξαμενή 2 και αντίστροφα. Επομένως αρχικά συνδέουμε τις βαλβίδες και την αντλία, οι οποίες συμμετέχουν στη ροή προς τα δεξιά και προς τα αριστερά, με τις κατάλληλες εξόδους που ορίσαμε στο πρόγραμμα. Τέλος, θα συνδέσουμε κάθε μία από τις εισόδους I0.0 και I0.1 με έναν διακόπτη, ώστε να εξυπηρετεί στον έλεγχο ενεργοποίησης και απενεργοποίησης του συστήματος. Μόλις ενεργοποιήσουμε το διακόπτη «SW\_1», ο οποίος αντιστοιχεί στην έξοδο «I0.0», ενεργοποιούνται οι βαλβίδες και η αντλία που εξυπηρετούν τη ροή προς τα δεξιά (βλ. εικόνα 149). Αντίθετα, όταν ενεργοποιήσουμε το διακόπτη «SW\_2», ο οποίος αντιστοιχεί στην έξοδο «I0.1», ενεργοποιούνται οι βαλβίδες και η αντλία που εξυπηρετούν τη ροή προς τα αριστερά (βλ. εικόνα 150). Εάν ενεργοποιηθούν και οι δύο, το σύστημα δεν λειτουργεί (βλ. εικόνα 151).



Εικόνα 149: Μεταφορά ρευστού από την αριστερή δεξαμενή (Tank\_1) στη δεξιά δεξαμενή (Tank\_2) με χρήση κατάλληλων βαλβίδων.



Εικόνα 150: Μεταφορά ρευστού από τη δεξιά δεξαμενή (Tank\_2) στην αριστερή δεξαμενή (Tank\_1) με χρήση κατάλληλων βαλβίδων.



Εικόνα 151: Ενεργοποίηση και των δύο διακοπών για τη μεταφορά ρευστού και στις δύο κατευθύνσεις ταυτόχρονα.

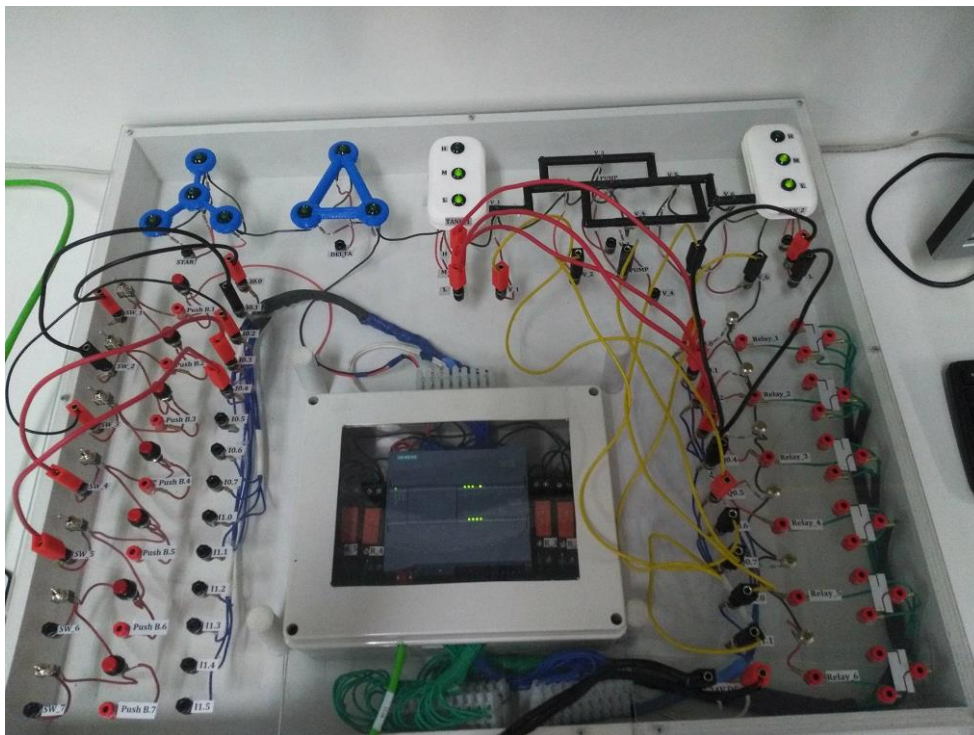
Βήμα 3<sup>ο</sup>: Αυτό το βήμα αποσκοπεί στη σωστή συνδεσμολογία, σύμφωνα με τον κώδικα που φτιάξαμε, ώστε όταν η δεξαμενή 1 είναι γεμάτη ενώ η δεξαμενή 2 είναι μέχρι το χαμηλό της επίπεδο, να γίνει μεταφορά ρευστού από τη δεξαμενή 1 στη 2 με χρήση των κατάλληλων βαλβίδων, με αποτέλεσμα όταν οι δύο δεξαμενές φτάσουν σε ισορροπία, να κλείσουν.

Αρχικά συνδέουμε τις βαλβίδες και την αντλία, οι οποίες συμμετέχουν στη ροή προς τα δεξιά και προς τα αριστερά, με τις κατάλληλες εξόδους που ορίσαμε στο πρόγραμμα (διακρίνονται στην εκπαιδευτική μονάδα με κίτρινα καλώδια). Στη συνέχεια συνδέουμε τα *led*, τις λυχνίες ένδειξης λειτουργίας των αισθητήρων των δεξαμενών, ώστε να ξέρουμε πόσο ρευστό έχει μέσα η δεξαμενή κάθε φορά (διακρίνονται στην εκπαιδευτική μονάδα, στη δεξιά πλευρά. Με κόκκινο καλώδιο συνδέονται οι έξοδοι με τα *led* της δεξαμενής 1, ενώ με μαύρο είναι της δεξαμενής 2). Τα χρώματα των καλωδίων μας βοηθούν να αναγνωρίζουμε την αντιστοιχία των εισόδων με τους αισθητήρες κάθε δεξαμενής. Επομένως, στην αριστερή πλευρά της εκπαιδευτικής μονάδας, έχουμε συνδέσει τις εισόδους του *PLC*, τις οποίες ορίσαμε στον κώδικα, με διακόπτες. Οι διακόπτες με μαύρο καλώδιο αντιστοιχούν στους αισθητήρες της δεξαμενής 1, ενώ εκείνοι με το κόκκινο καλώδιο αντιστοιχούν στη δεξαμενή 2.

Όταν ενεργοποιηθούν οι διακόπτες (*SW*), οι οποίοι αντιστοιχούν στους αισθητήρες *H*, *M* και *L* (της δεξαμενής 1) και *L* (της δεξαμενής 2), τότε το σύστημα μεταφοράς ρευστού από τη δεξαμενή ένα στη δεξαμενή 2, ενεργοποιείται (βλ. εικόνα 152). Ο κώδικας έχει ολοκληρωθεί όταν γίνει η μεταφορά του ρευστού και οι δεξαμενές έχουν ίση ποσότητα από αυτό. Τότε οι βαλβίδες και η αντλία απενεργοποιούνται (απενεργοποιούνται τα *led* ένδειξης λειτουργίας). Ωστόσο παραμένουν, αναμμένα τα *led* ένδειξης λειτουργίας των αισθητήρων των δεξαμενών (βλ. εικόνα 153).



Εικόνα 152: Μεταφορά ρευστού από την αριστερή δεξαμενή (Tank\_1) στη δεξιά δεξαμενή (Tank\_2) με χρήση κατάλληλων βαλβίδων.



Εικόνα 153: Το σύστημα βρέθηκε σε ισορροπία μετά τη μεταφορά ρευστού από τη δεξαμενή ένα στη δύο.

## 12.4 Άσκηση 3: Εκκίνηση τριφασικού κινητήρα με χρήση προγραμματιζόμενου λογικού ελεγκτή

Σκοπός της παρούσας άσκησης είναι να βοηθήσει το φοιτητή να κατανοήσει καλύτερα τη λειτουργία ενός τριφασικού κινητήρα της γλώσσας *Ladder* για τη δημιουργία κώδικα, με σκοπό την εκκίνηση κινητήρα, καθώς και τη χρήση αισθητήρων και ηλεκτρονόμων για την επίτευξη της ομαλής και ασφαλούς λειτουργίας του κινητήρα.

Να πραγματοποιηθούν τα παρακάτω βήματα, με χρήση της γλώσσας προγραμματισμού *Ladder*, έτσι ώστε με τη κατάλληλη συνδεσμολογία της εκπαιδευτικής μονάδας, να λειτουργεί το πρόγραμμα.

- Κάνοντας χρήση κατάλληλης εισόδου, να ενεργοποιηθεί το *led* της συνδεσμολογίας αστέρα (*Star led*). Να τροποποιηθεί ο κώδικας ώστε να ενεργοποιείται η έξοδος με χρήση ανατροφοδότησης/μανδάλωσης (χρήση *push button* και με μεταβλητή μνήμης (*Global Memory*). Να παρέχεται κουμπί έκτακτης ανάγκης για το τερματισμό της λειτουργίας του κινητήρα (*emergency button*).
- Να ενεργοποιείται διαδοχικά (**από μια φορά**) τα *led* συνδεσμολογίας αστέρα και τρίγωνο μέσω ηλεκτρονόμου.
- Με δεδομένο ότι η έξοδος του επαγωγικού αισθητήρα αντιστοιχεί σε ένα *push button*, να φτιαχτεί πρόγραμμα να υπολογιστούν οι στροφές του κινητήρα (*rpm*). Θεωρείστε ένας παλμός του αισθητήρα αντιστοιχεί σε μια περιστροφή του οδοντωτού τροχού του στροφάλου και χρόνο μέτρησης παλμών **60 sec**.
- Θεωρώντας κατάλληλο αισθητήρα με έξοδο *PNP* (επαγωγικός αισθητήρας) και κατάλληλη αντίσταση της τάξης των  $K\Omega$ , να μετρηθούν οι στροφές του κινητήρα του εργαστηρίου (*rpm*). Θεωρήστε χρόνο μέτρησης παλμών **30 sec**. Θεωρήστε ότι η έξοδος του επαγωγικού αισθητήρα αντιστοιχεί σε ένα *push button*,

**Προσοχή:** Δεν αντιστοιχεί 1 παλμός σε 1 περιστροφή του κινητήρα.

- Σε συνέχεια του 3<sup>ου</sup> βήματος, να τροποποιηθεί ο κώδικας ώστε σε περίπτωση που ο κινητήρας ξεπεράσει τις 100 στροφές (*rpm*) να διακοπεί αυτόματα η λειτουργία του.

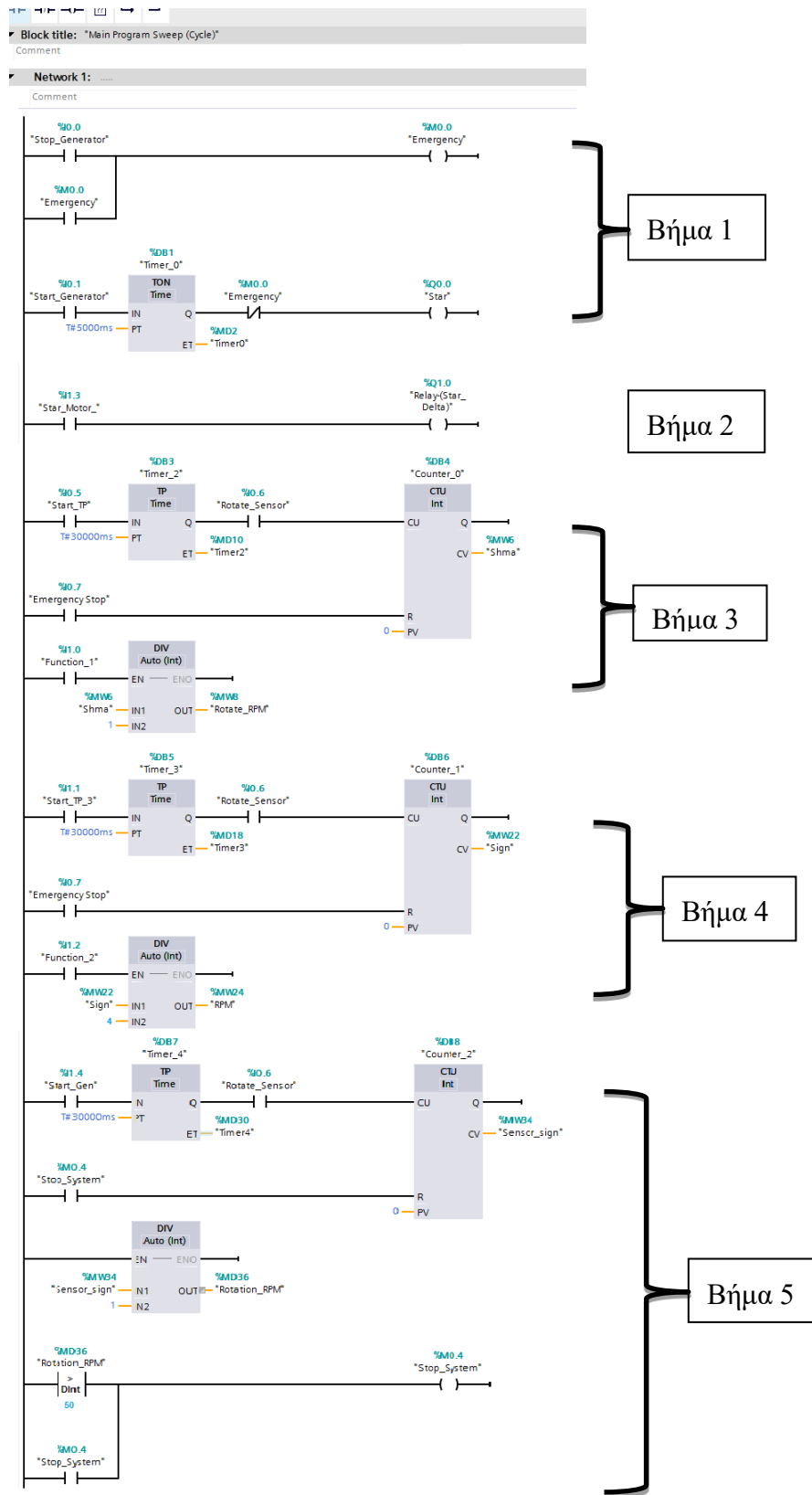
### **ΠΡΟΣΟΧΗ ΣΤΗ ΣΥΝΔΕΣΗ ΤΟΥ ΤΡΟΦΟΔΟΤΙΚΟΥ ΜΕ ΤΗ ΜΟΝΑΔΑ.**

**ΣΗΜΕΙΩΣΗ:**

- Σε όλες τις περιπτώσεις κομβίου μανδάλωσης να γίνει χρήση των *push buttons*.

- Ο αισθητήρας είναι της *IFM* τύπου *IGC222* - βλέπε αντίστοιχο εγχειρίδιο τεχνικών προδιαγραφών του αισθητήρα.
- Κάθε φορά που δημιουργείται ένα νέο *project* ή διορθώνεται ένα ήδη υπάρχον πρόγραμμα, μέσω του προγράμματος *TIA Portal V13*, για να το αποθηκεύσουμε από τον υπολογιστή στο *PLC* με σκοπό να το τρέξει, χρειάζεται να επιλέξουμε την εντολή «*Download to Device*» και στη συνέχεια να επιλέξουμε την εντολή «*Start CPU*».
- Πριν πάμε να κάνουμε αλλαγές στο πρόγραμμα, πρώτα επιλέγουμε την εντολή «*Stop CPU*».
- Όταν επιλέγουμε μονάδα *PLC* να συνδεθεί με τον υπολογιστή που έχουμε κάνει το πρόγραμμα, επιλέγουμε την εντολή «*flash led*» για να είμαστε σίγουροι ότι δεν θα επιλέξουμε άλλη μονάδα.
- Με χρήση της εντολής «*Monitoring on/off*» η οποία βρίσκεται στην μπάρα εργαλείων στο παράθυρο «*Main*», μπορούμε να παρατηρούμε τη ροή του «ρεύματος» στο πρόγραμματος.

Παρακάτω ακολουθεί ο κώδικας σε γλώσσα *Ladder* των παραπάνω 5 βημάτων, η εξήγηση του προγράμματος, καθώς και η σωστή συνδεσμολογία της μονάδας για την επίτευξη του επιθυμητού αποτελέσματος.



Εικόνα 154: Κώδικας σε γλώσσα Ladder των 5 βημάτων της εκφόνησης



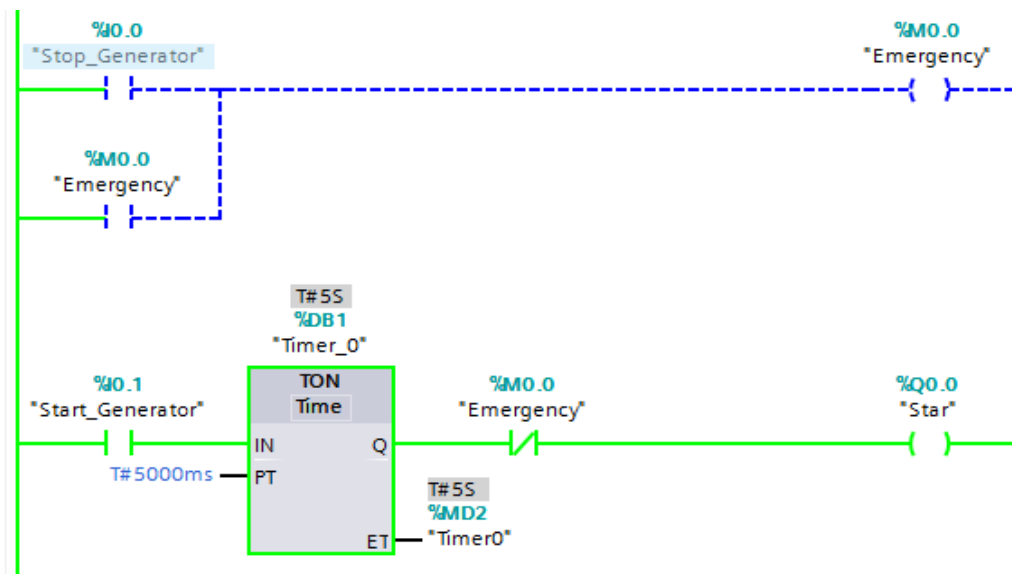
Πίνακας 9: Σύμβολα, τελεστές, ονομασίες του κώδικα και η εξήγησή τους.

ΣΥΜΒΟΛΟ	ΤΕΛΕΣΤΗΣ	ΟΝΟΜΑΣΙΑ	ΕΞΗΓΗΣΗ
<b>I0</b>	I0.0	Stop_Generator	Κουμπί: Διακόπτει τη λειτουργία του κινητήρα.
<b>I1</b>	I0.1	Start_Generator	Διακόπτης: Ενεργοποιεί το κινητήρα.
<b>I2</b>	I0.5	Start TP	Διακόπτης: Ενεργοποιεί το χρονόμετρο Timer_2.
<b>I3</b>	I0.6	Rotate Sensor	Κουμπί: Σήμα αισθητήρα
<b>I4</b>	I0.7	Emergency Stop	Κουμπί: Σταματάει το σύστημα σε περίπτωση έκτακτης ανάγκης
<b>I5</b>	I1.0	Function 1	Κουμπί: Ενεργοποιεί την εντολή υπολογισμού στροφών.
<b>I6</b>	I1.1	Start TP_3	Διακόπτης: Ενεργοποιεί το χρονόμετρο Timer_3.
<b>I7</b>	I1.2	Function 2	Κουμπί: Ενεργοποιεί την εντολή υπολογισμού στροφών.
<b>I8</b>	I1.3	Start_Motor_	Διακόπτης: Ενεργοποιεί το κινητήρα.
<b>I9</b>	I1.4	Start Gen	Διακόπτης: Ενεργοποιεί το χρονόμετρο Timer_4.
<b>Q0</b>	Q0.0	Star	Led: Αστέρα
<b>Q1</b>	Q1.0	Relay-(Star-Delta)	Διακόπτης: Ενεργοποιεί το Relay.
<b>M0</b>	M0.0	Emergency	Global Memory: Διατηρεί τη κατάσταση που αντιστοιχεί στο στοιχείο M0.
<b>M1</b>	M0.4	Emergency stop	Global Memory: Διατηρεί τη κατάσταση που αντιστοιχεί στο στοιχείο M1.
<b>Timer_0</b>	MD2	Timer0	Χρονόμετρο: Ρυθμισμένο στα 5sec να ανοίξει την έξοδο Q0.
<b>Timer_2</b>	MD10	Timer2	Χρονόμετρο: Ρυθμισμένο να παρέχει ρεύμα στο σύστημα για 30sec.
<b>Timer_3</b>	MD18	Timer3	Χρονόμετρο: Ρυθμισμένο να παρέχει ρεύμα στο σύστημα για 30sec.
<b>Timer_4</b>	MD30	Timer4	Χρονόμετρο: Ρυθμισμένο να παρέχει ρεύμα στο σύστημα για 30sec.
<b>DIV</b>	MW13	Rotate-RPM	Μαθηματική εξίσωση: Διαίρεση 2 τιμών.
<b>DIV</b>	MW24	RPM	Μαθηματική εξίσωση: Διαίρεση 2 τιμών.
<b>Counter_0</b>	MW6	Shma	Μετρητής: Μετράει τον αριθμό των σημάτων που δίνουμε.
<b>Counter_1</b>	MW22	Sign	Μετρητής: Μετράει τον αριθμό των σημάτων που δίνουμε.
<b>Counter_3</b>	MW34	Sensor_Sign	Μετρητής: Μετράει τον αριθμό των σημάτων που δίνουμε.

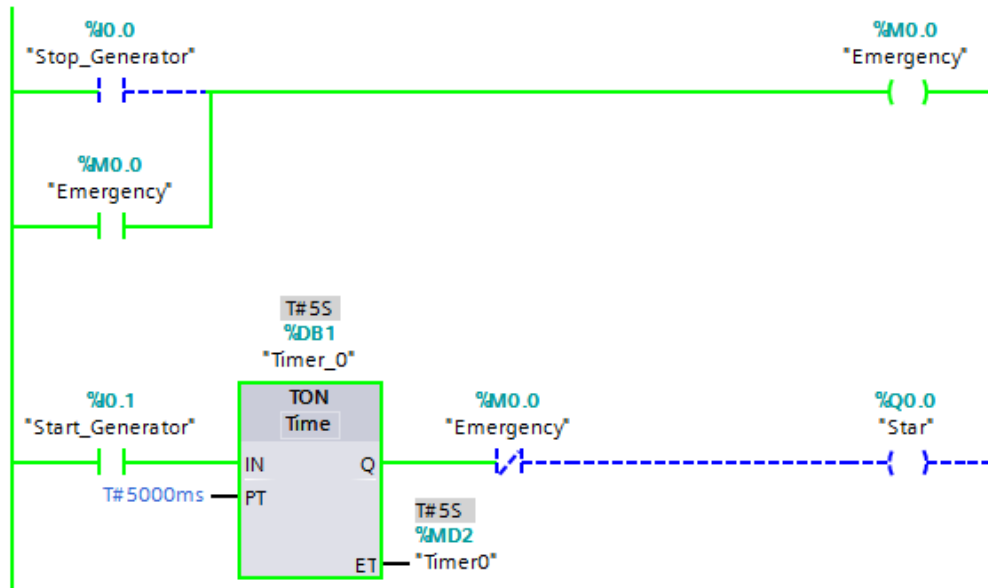
## 12.4.1 Επεξήγηση ενδεικτικού κώδικα

Σε αυτή την υποενότητα θα γίνει αναφορά στον κώδικα που κατασκευάστηκε με σκοπό τη λύση των 5 βημάτων που αναφέρονται παραπάνω. Αρχικά να σημειωθεί ότι όταν κατασκευάζουμε ένα κώδικα προσέχουμε να μην έχουμε τον ίδιο τελεστή εξόδου σε περισσότερες από μια γραμμές. Για τις εισόδους δεν έχουμε αυτόν το περιορισμό, μιας και εξαρτάται από τη συνδεσμολογία που έχουμε κάνει. Παρακάτω αναφέρεται η επεξήγηση του κώδικα με τη βοήθεια της εικόνας 156.

Βήμα 1<sup>ο</sup>: Το πρώτο βήμα αποτελείται από 3 γραμμές κώδικα. Οι δύο πρώτες αποτελούν το σύστημα έκτακτης ανάγκης που θα απενεργοποιήσει τον κινητήρα, ενώ η τρίτη γραμμή θα ενεργοποιήσει τον κινητήρα (συνδεσμολογία αστέρα), μόλις περάσουν 5 sec (*Timer – TON*) από την ενεργοποίηση της εισόδου (I0.1). Στη πρώτη γραμμή του κώδικα έχουμε μία είσοδο (I0.0). Όταν την «κλείσουμε» ενεργοποιείται η έξοδος (M0.0). Η έξοδος M0.0 (*Global Memory*), λόγω του συστήματος ανατροφοδότησης – μανδάλωσης (δεύτερη γραμμή) διατηρεί το σήμα της με αποτέλεσμα να παραμένει ανοιχτή. Αυτό έχει ως αποτέλεσμα την απενεργοποίηση της *normal close* εισόδου, που ορίσαμε στη τρίτη γραμμή. Έτσι το σύστημα σταματάει να τροφοδοτεί τον κινητήρα (βλ. εικόνα 156).

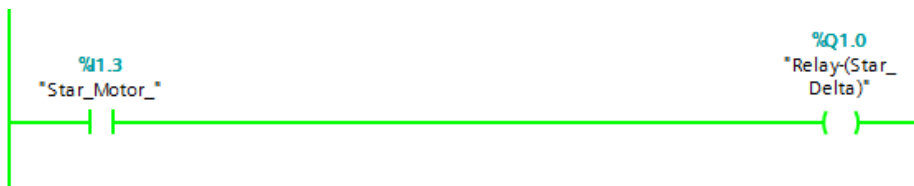


Εικόνα 155: Όταν ενεργοποιήσουμε την είσοδο I0.1.



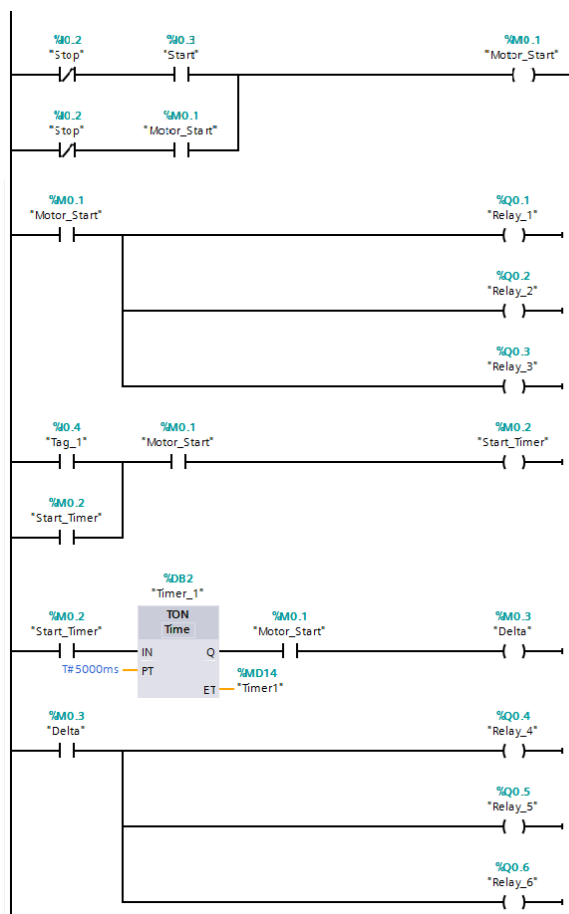
Εικόνα 156: Όταν ενεργοποιείται το κουμπί έκτακτης ανάγκης.

Βήμα 2<sup>ο</sup>: Σε αυτό το βήμα με χρήση ενός ηλεκτρονόμου θα γίνει η εναλλαγή των *led* από αστέρα σε τρίγωνο. Κανονικά, λόγω τριφασικού κινητήρα χρειαζόμαστε 6 ηλεκτρονόμους και συγκεκριμένη συνδεσμολογία των ηλεκτρονόμων με τα τυλίγματα του κινητήρα, για τη λειτουργία σε αστέρα και τρίγωνο. Λόγω μη ικανοποιητικού αριθμού κινητήρων στο εργαστήριο θα γίνει χρήση μόνο ενός ηλεκτρονόμου για την εναλλαγή των *led*. Επομένως, έχουμε μια είσοδο (I1.3) και μια έξοδο (Q1.0). Όταν ενεργοποιηθεί η είσοδος θα δώσει σήμα στην έξοδο να ενεργοποιηθεί. Για να έχουμε το επιθυμητό αποτέλεσμα, σημαντικό ρόλο παίζει η συνδεσμολογία της εκπαιδευτικής μονάδας.



Εικόνα 157: Κώδικας για χρήση ηλεκτρονόμου στην εναλλαγή συνδεσμολογίας αστέρα – τρίγωνο.

Σε περίπτωση χρήσης αισθητήρα και κινητήρα ο κώδικας που θα μπορούσαμε να χρησιμοποιήσουμε είναι ο ακόλουθος.

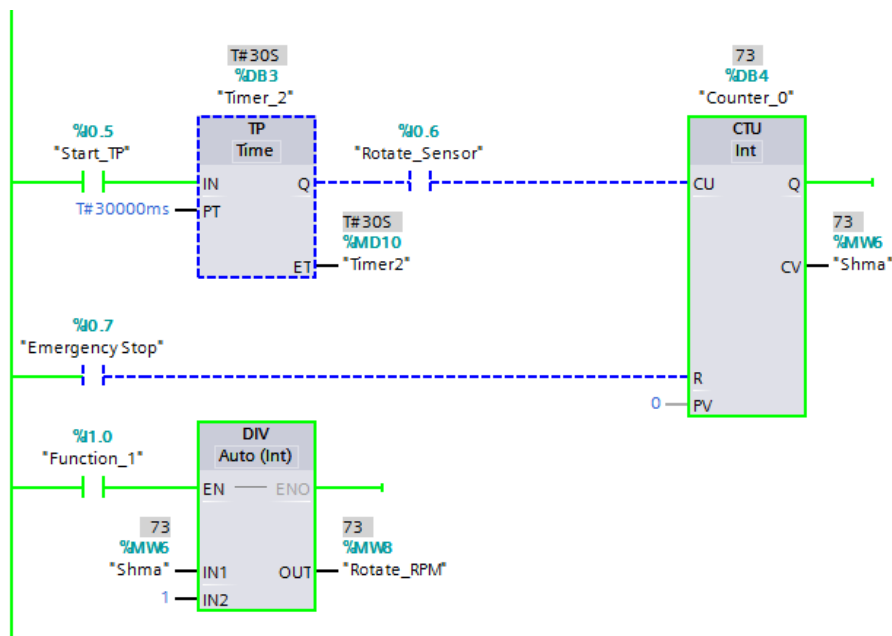


Εικόνα 158: Κώδικας για χρήση έξι ηλεκτρονόμων στην εναλλαγή συνδεσμολογίας αστέρα – τρίγωνο.

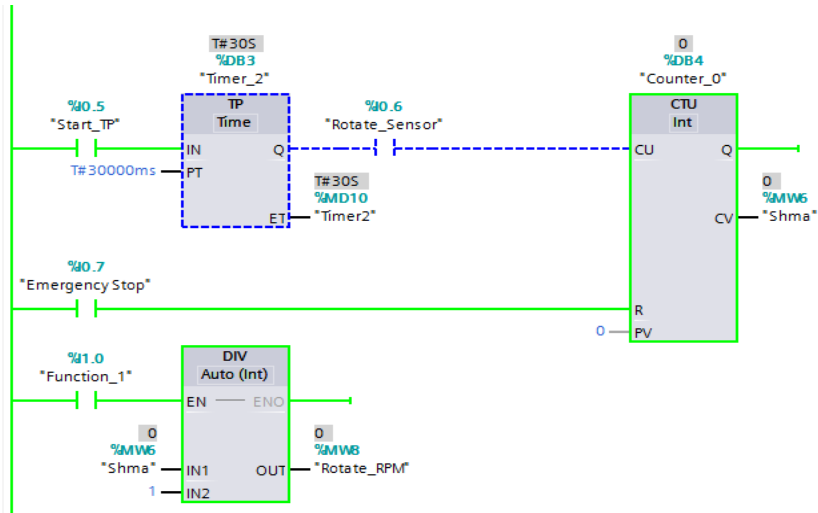
Στις πέντε πρώτες γραμμές του κώδικα επιτυγχάνεται η εκκίνηση και η διακοπή του κινητήρα μόλις πατηθεί το *button* «Start», στο οποίο αντιστοιχεί η επαφή I0.3 (γραμμή 1), δηλαδή μόλις δημιουργείται αγωγίμος δρόμος με αποτέλεσμα την ενεργοποίηση του *flag* M0.1 (*Motor\_Start*). Το *flag* αυτό παραμένει ενεργοποιημένο μέχρι να κλείσει η επαφή I0.2, δηλαδή μέχρι να πατηθεί το *button* «Stop». Η δεύτερη γραμμή του κώδικα αποτελεί μία αυτοσυγκράτηση. Άμεση συνέπεια της ενεργοποίησης του M0.1, είναι η ενεργοποίηση των τριών ηλεκτρονόμων (*Relay\_1*, *Relay\_2*, *Relay\_3*) που ευθύνονται για την εκκίνηση του κινητήρα σε συνδεσμολογία αστέρα. Μόλις πατηθεί το *button* «Stop» τα ρελέ απενεργοποιούνται.

Στη συνέχεια, εφόσον ενεργοποιηθεί η είσοδος I0.4 και η επαφή M0.1 είναι κλειστή, ενεργοποιείται η επαφή M0.2. Η έβδομη γραμμή προσφέρει αυτοσυγκράτηση στο συγκεκριμένο κλάδο. Η ενεργοποίηση της M0.2 έχει ως αποτέλεσμα την εκκίνηση της λειτουργίας ενός χρονόμετρου «TON», το οποίο ενεργοποιεί την έξοδο μετά από την παρέλευση ενός προκαθορισμένου διαστήματος (5 sec). Μετά το πέρας των πέντε δευτερολέπτων και εφόσον η επαφή M0.1 είναι κλειστή, ενεργοποιείται η έξοδος M0.3 (*Delta*). Τέλος, η ενεργοποίηση της M0.3 έχει ως αποτέλεσμα την ενεργοποίηση των τριών ηλεκτρονόμων που οφείλονται για τη λειτουργία του κινητήρα υπό συνδεσμολογία τριγώνου (*Relay\_4*, *Relay\_5*, *Relay\_6*). Ως εκ τούτου, μόλις ενεργοποιηθούν οι τρεις αυτοί ηλεκτρονόμοι, ο κινητήρας λειτουργεί σε συνδεσμολογία τριγώνου.

Βήμα 3<sup>ο</sup>: Σε αυτό το βήμα θεωρούμε ότι ένα *push button* είναι η έξοδος του σήματος ενός επαγωγικού αισθητήρα (είσοδος I0.6). Έχουμε ορίσει έναν *Timer* (*TP*). Αυτό το είδος χρονόμετρου δίνει ρεύμα στην έξοδο από τη στιγμή που θα ενεργοποιηθεί, μέχρι η τιμή *ET* να γίνει ίση με την τιμή που είχαμε ορίσει στην αρχή (*PT*) μετά την είσοδο. Επιπλέον ορίσαμε έναν *Counter* (*CTU*). Ο μετρητής «μετράει» τους παλμούς του σήματος που δέχεται από την είσοδο I0.6 (αισθητήρας). Παράλληλα για τον έλεγχο ενεργοποίησης του συστήματος έχουμε ορίσει δύο εισόδους I0.5 και I1.0. Η πρώτη ενεργοποιεί το χρονόμετρο και η δεύτερη ενεργοποιεί τη μαθηματική εξίσωση (*DIV*), η οποία διαιρεί τον αριθμό των παλμών που εντόπισε ο μετρητής με τον αριθμό των οδοντωτών τροχών του στρόφαλου του κινητήρα. Στη προκειμένη περίπτωση έχουμε υποθέσει ότι ο αισθητήρας δίνει ένα παλμό όταν ο κινητήρας έχει κάνει μια πλήρη περιστροφή. Τέλος, έχουμε μια ακόμη είσοδο (I0.7), η οποία συνδέεται με τον *Counter*. Όταν ενεργοποιηθεί μηδενίζει το μετρητή.

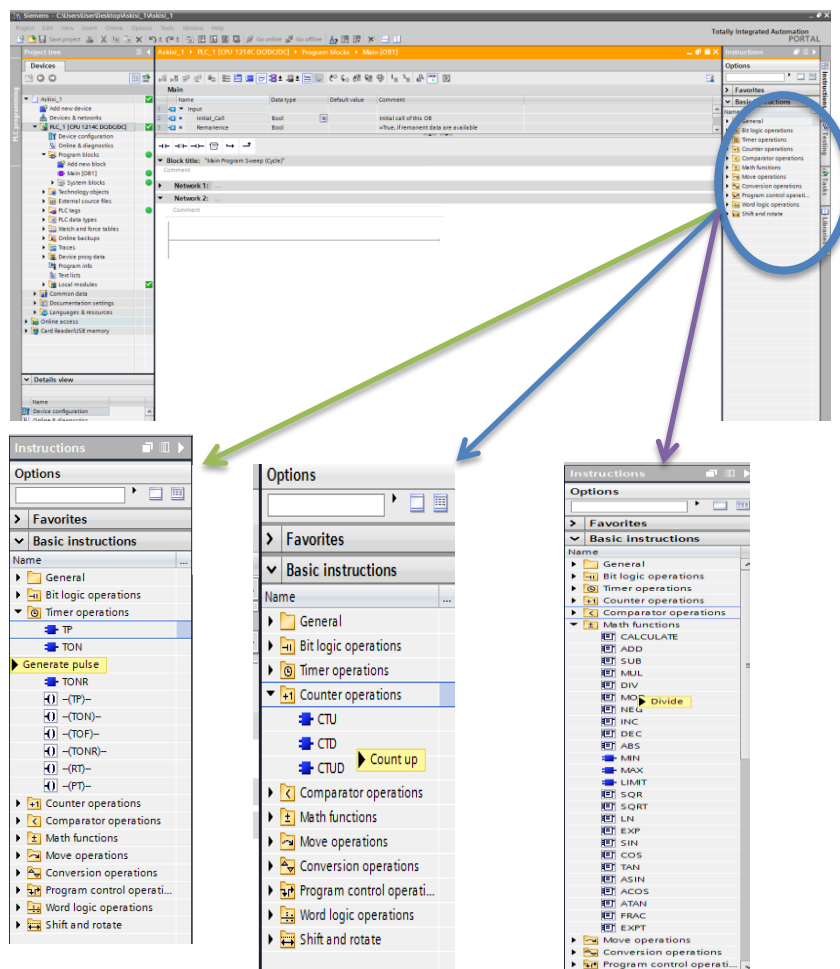


Εικόνα 159: Χρήση αισθητήρα για τον υπολογισμό στροφών (1 παλμό = 1 περιστροφή).



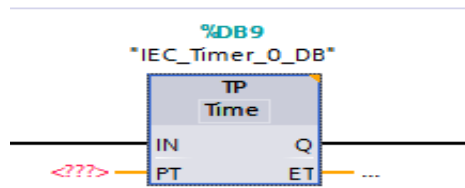
Εικόνα 160: Ενεργοποίηση emergency button.

Οι δομές/εντολές *Timer (TP)*, *Counter (CPU)* και *Divide (DIV)* είναι διαθέσιμες στο περιβάλλον *TIA PORTAL* (βλ. εικόνα 161).



Εικόνα 161: Tools bar με χρήσιμες εντολές για την κατασκευή κώδικα Ladder – Διακρίνεται η εντολή *Timer TP* (Με πράσινο βέλος), η εντολή *Counter CTU* (Με μπλε βέλος) και η εντολή *Divide* (Με μοβ βέλος).

- *Timer TP*: Όταν τοποθετήσουμε τη δομή στο πρόγραμμά, πρέπει να ορίσουμε τις αντίστοιχες μεταβλητές εισόδου/εξόδου (*IN*, *Q*, *PT*, *ET*). Ο ρόλος της κάθε μεταβλητής παρουσιάζεται στον Πίνακα 10.

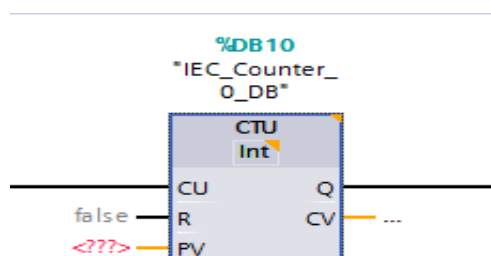


Εικόνα 162: Επιλογή Timer TP.

Πίνακας 10: Στοιχεία χρονικών μεταβλητών του Timer TP.

Μεταβλητή	Περιγραφή
IN	Ενεργοποιεί το χρονοδιακόπτη (μεταβλητή εισόδου).
Q	Είναι ενεργοποιημένο όσο λειτουργεί το χρονόμετρο (μεταβλητή εξόδου).
PT	Ορίζουμε τη τιμή του χρονόμετρου. Προσοχή στη μονάδα μέτρησης (μεταβλητή εισόδου).
ET	Η τρέχουσα τιμή μέτρησης (μεταβλητή εξόδου).

- *Counter (CPU)*: Όταν τοποθετήσουμε τη δομή στο πρόγραμμά, πρέπει να ορίσουμε τις αντίστοιχες μεταβλητές εισόδου/εξόδου (*CU*, *Q*, *CV*, *PV*). Ο ρόλος της κάθε μεταβλητής παρουσιάζεται στον Πίνακα 11.

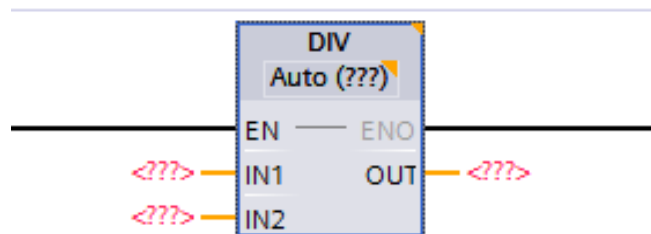


Εικόνα 163: Επιλογή Counter (CTU).

Πίνακας 11: Στοιχεία μετρητικών μεταβλητών του Counter (CTU).

Μεταβλητή	Περιγραφή
CU	Ενεργοποιεί το χρονοδιακόπτη (μεταβλητή εισόδου).
Q	Είναι ενεργοποιημένο όσο λειτουργεί το χρονόμετρο (μεταβλητή εξόδου).
CV	Η τρέχουσα τιμή μέτρησης (μεταβλητή εξόδου).
R	Αντιστοιχεί σε μια είσοδο η οποία μηδενίζει το μετρητή.
PV	Αριθμός σημάτων που θα δεχτεί ο μετρητής για την ενεργοποίηση της μεταβλητής R.

- Divine (DIV): Όταν τοποθετήσουμε τη δομή στο πρόγραμμα, πρέπει να ορίσουμε τις αντίστοιχες μεταβλητές εισόδου/εξόδου (*EN*, *OUT*, *IN1*, *IN2*). Ο ρόλος της κάθε μεταβλητής παρουσιάζεται στον Πίνακα 12.



Εικόνα 164: Επιλογή μαθηματικής εντολής Divine (DIV).

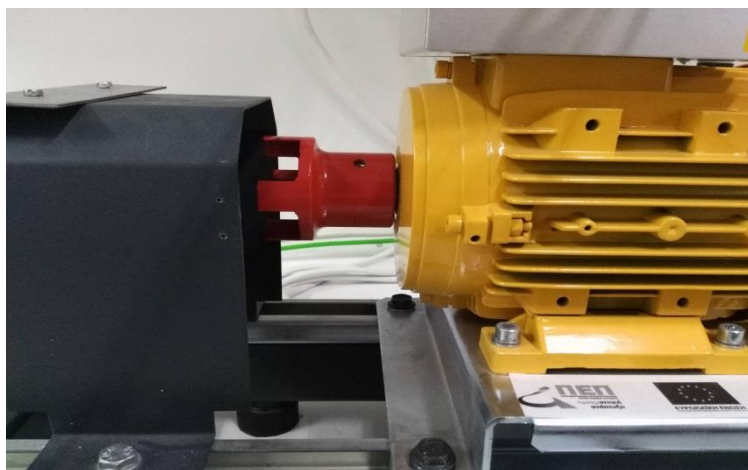


Πίνακας 11: Στοιχεία μαθηματικών μεταβλητών του Divine (DIV).

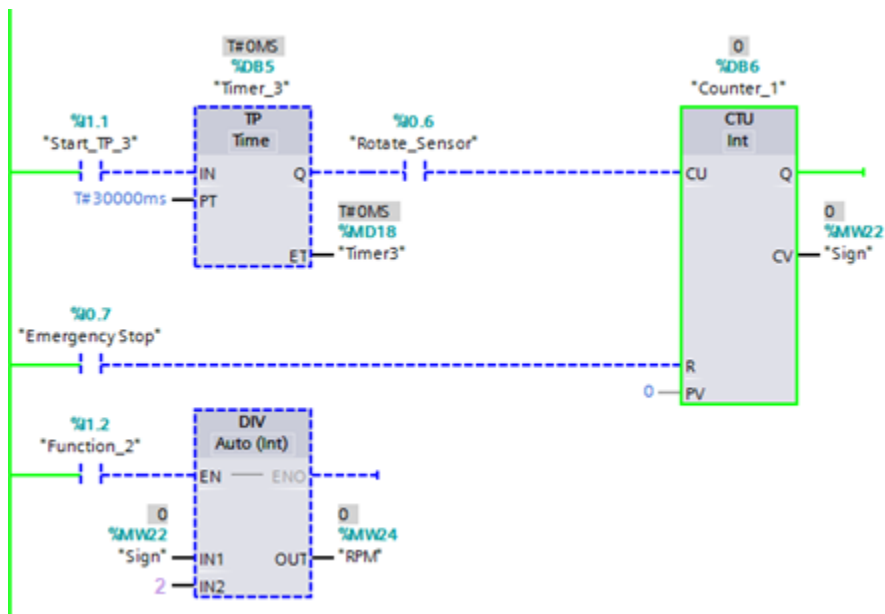
Μεταβλητή	Περιγραφή
EN	Ενεργοποίηση της αριθμητικής εξίσωσης
OUT	Αποτέλεσμα της διαίρεσης $\left(\frac{IN1}{IN2}\right)$
IN1	Αριθμητής διαίρεσης
IN2	Παρανομαστής διαίρεσης

Εάν επιθυμούμε να αλλάξουμε τον τύπο μιας μεταβλητής, πάμε με δεξί κλικ πάνω σε αυτή και επιλέγουμε την αλλαγή του τύπου της μεταβλητής «Rewire tag».

Βήμα 4<sup>ο</sup>: Το 4<sup>ο</sup> βήμα είναι όμοιο με το 3<sup>ο</sup> βήμα. Η μόνη διαφορά είναι ότι ο κινητήρας του εργαστηρίου ναυτικών μηχανών, έχει τέσσερις οδοντωτούς τροχούς (βλ. εικόνα 165). Επομένως στην εντολή *Divine* θα ορίσουμε να διαιρέσει τους παλμούς που εντόπισε ο μετρητής με τον αριθμό των οδοντωτών τροχών. Επιπλέον ο χρόνος μέτρησης των στροφών μειώθηκε στο μισό (σε σχέση με το χρόνο του βήματος 3), επομένως χρειάζεται προσοχή στο κώδικα ώστε να εμφανίζει απευθείας *rpm* και όχι στροφές.

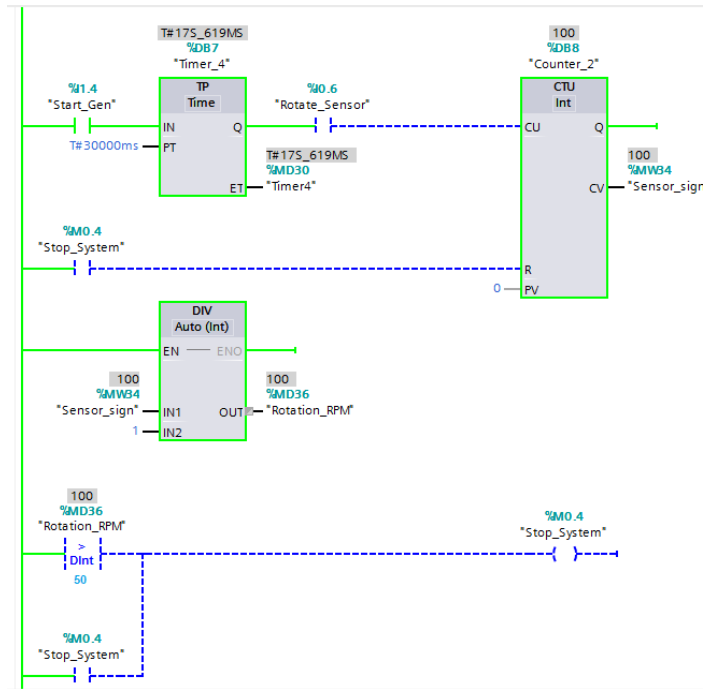


Εικόνα 165: Κινητήρας εργαστηρίου ναυτικών μηχανών και εργαλειομηχανών.

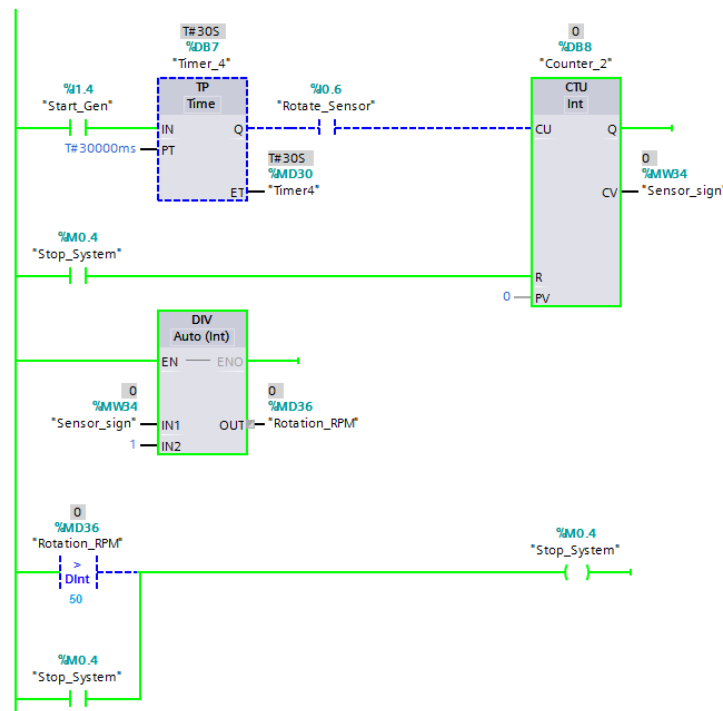


Εικόνα 166: Χρήση αισθητήρα για τον υπολογισμό στροφών (6 παλμοί = 1 περιστροφή).

Βήμα 5<sup>ο</sup>: Το βήμα αυτό αποτελείται από έναν κώδικα πέντε γραμμών. Οι τρεις πρώτες γραμμές είναι όμοιες με του τρίτου βήματος με μόνη διαφορά την αφαίρεση της εισόδου που τοποθετούσαμε πριν την εντολή «Divine», διότι δεν θέλουμε να ελέγχουμε εμείς το τελικό αποτέλεσμα αλλά, σύμφωνα με την εκφώνηση, μόλις ο κινητήρας ξεπεράσει τις 50 στροφές να διακοπεί η λειτουργία του. Στις τελευταίες 2 γραμμές του κώδικα έχουμε ορίσει μια εντολή σύγκρισης «Greater than» η οποία ελέγχει τις τιμές των στροφών με το επιθυμητό όριο στροφών. Μόλις ξεπεράσει το όριο τότε ενεργοποιεί την έξοδο «M0.4». Αυτή η έξοδος αποτελεί και είσοδο μηδενισμού του μετρητή. Αυτό έχει ως αποτέλεσμα το μηδενισμό του μετρητή, χωρίς όμως να απενεργοποιεί τη λειτουργία του κινητήρα. Εμείς όμως, επιθυμούμε τον τερματισμό του κινητήρα. Για το λόγο αυτό στη πέμπτη γραμμή του κώδικα φτιάχνουμε ένα σύστημα ανατροφοδότησης – μανδάλωσης, της μεταβλητής «M0.4», η οποία όχι μόνο θα μηδενίζει το μετρητή, αλλά δεν θα απενεργοποιεί τον κινητήρα. Επομένως ο μετρητής θα είναι συνέχεια μηδέν.

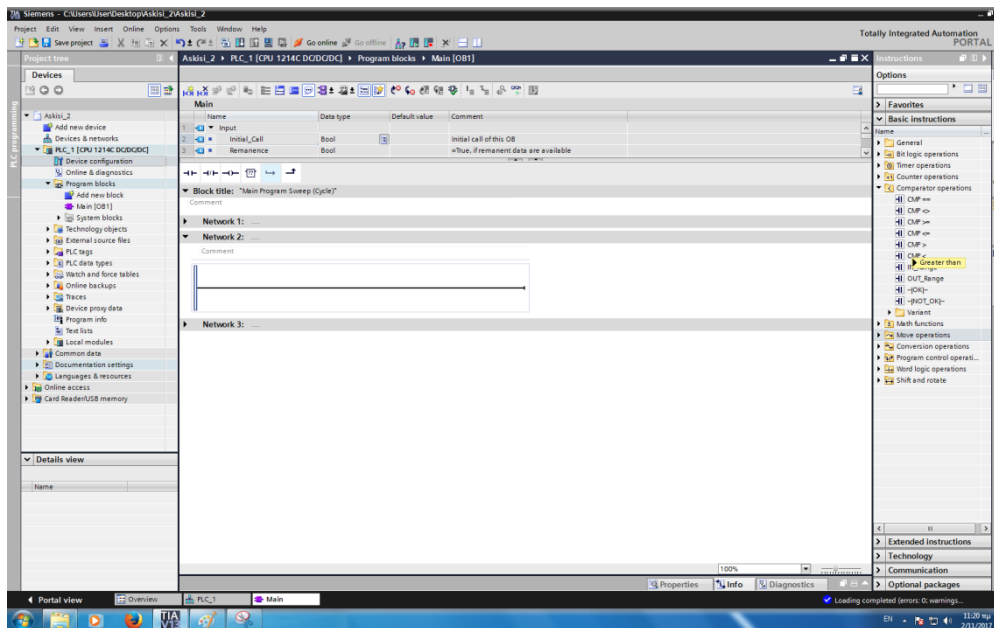


Εικόνα 167: Σύστημα ελέγχου στροφών κινητήρα με διακοπή συστήματος εάν ξεπεράσει τις 50 στροφές.



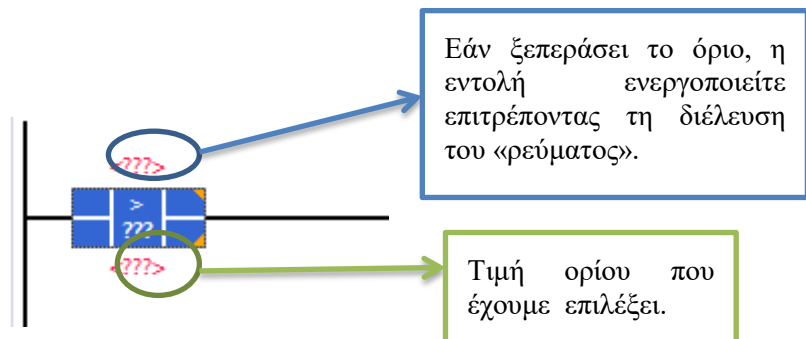
Εικόνα 168: Το σύστημα ξεπέρασε τις 50 στροφές με αποτέλεσμα την απενεργοποίηση του κινητήρα.

Η δομή/εντολή *Greater than* (*CMP >*) είναι διαθέσιμη στο περιβάλλον *TIA PORTAL* (βλ. εικόνα 169).



Εικόνα 169: Tools bar με χρήσιμες εντολές για την κατασκευή κώδικα Ladder – Διακρίνεται η εντολή Greater than (CMP >).

- Greater than (CPM >): Όταν τοποθετήσουμε τη δομή στο πρόγραμμά (*Network*), πρέπει να ορίσουμε τις αντίστοιχες μεταβλητές εισόδου/εξόδου.



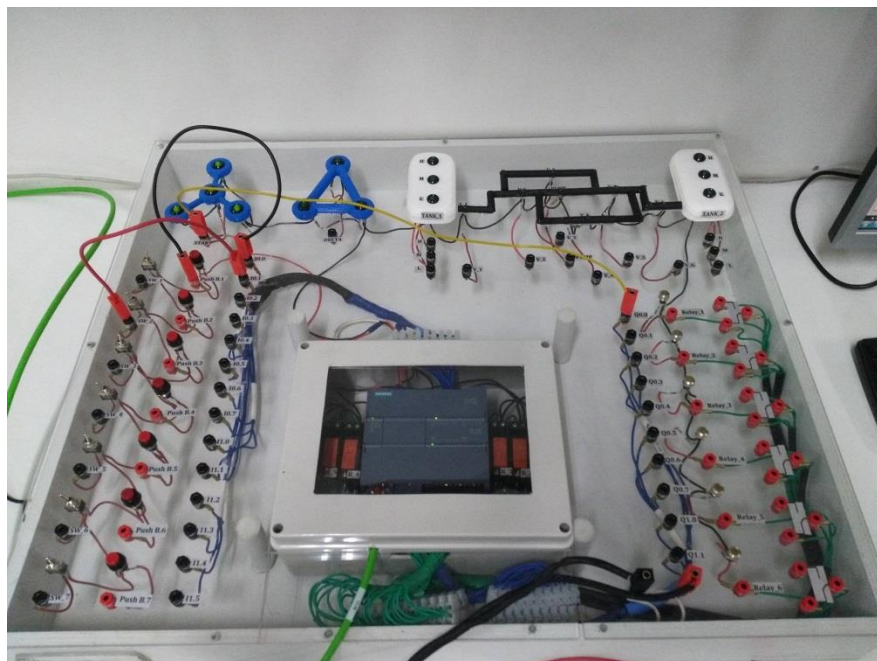
Εικόνα 170: Επιλογή μαθηματικής εντολής Greater than (CMP >).

## 12.4.2 Συνδεσμολογία εκπαιδευτικής μονάδας

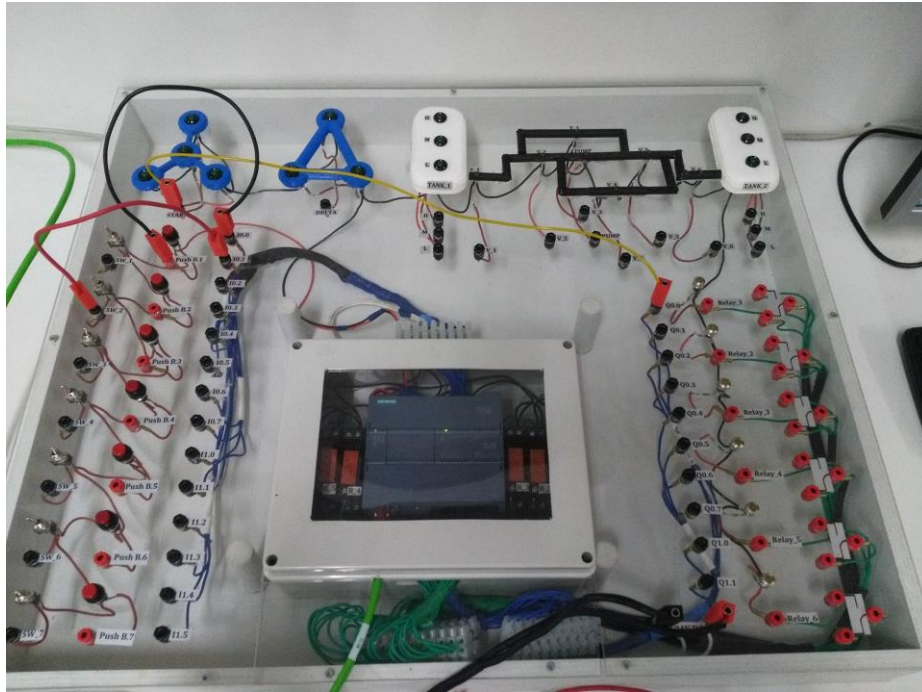
Πέρα από το σωστό προγραμματισμό μιας μονάδας *PLC*, εξίσου σημαντική είναι και η κατάλληλη συνδεσμολογία του προγραμματιζόμενου λογικού ελεγκτή ώστε να μπορέσει να λειτουργήσει το σύστημά με τον επιθυμητό τρόπο. Στην υποενότητα αυτή θα γίνει αναφορά στη σωστή συνδεσμολογία της εκπαιδευτικής μονάδας σύμφωνα με τον κώδικα *Ladder* τον οποίο αναλύσαμε προηγουμένως.

Βήμα 1<sup>ο</sup>: Σκοπός αυτού του βήματος ήταν να ενεργοποιηθεί η συνδεσμολογία του κινητήρα (*Star led*). Επιπλέον, να υπάρχει σύστημα ανατροφοδότησης, με χρήση *push button*, και κουμπί έκτακτης ανάγκης για το τερματισμό της λειτουργίας του κινητήρα (*emergency button*).

Στο κώδικα ορίσαμε ως είσοδο τη μεταβλητή I0.1 και ως έξοδο τη μεταβλητή Q0.0. Σύμφωνα με τη προηγούμενη παράγραφο θα συνδέσουμε τη μπόρνα της εισόδου «I0.0» με ένα διακόπτη (SW\_1), ενώ τη μπόρνα της εξόδου «Q0.0» θα τη συνδέσουμε με τη μπόρνα που αντιστοιχεί στο *led* «*Star*». Επιπλέον για το σύστημα *emergency stop* με ανατροφοδότηση, στο κώδικα ορίσαμε μια είσοδο (I0.0) η οποία αντιστοιχεί σε ένα *push button*. Επομένως συνδέσαμε τη μπόρνα της εξόδου «I0.0» με το κουμπί «*Push B.1*». Όταν θα ενεργοποιήσουμε το διακόπτη «*SW\_1*» θα ανάψει το *led* του αστέρα. Μόλις πατήσουμε το κουμπί, τότε το *led* απενεργοποιείται και παραμένει σε αυτή τη κατάσταση, μέχρι να αλλάξουμε την κατάσταση του συστήματος από τον υπολογιστή με τον οποίο συνδέεται ο προγραμματιζόμενος λογικός ελεγκτής.

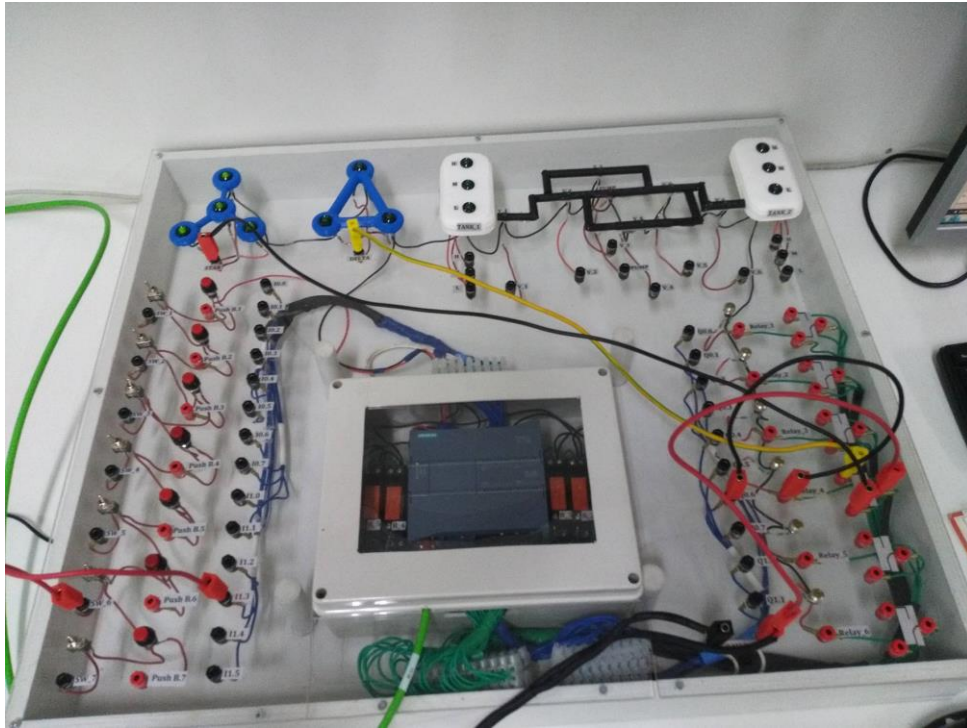


Εικόνα 171: Ενεργοποίηση εισόδου I0.1.

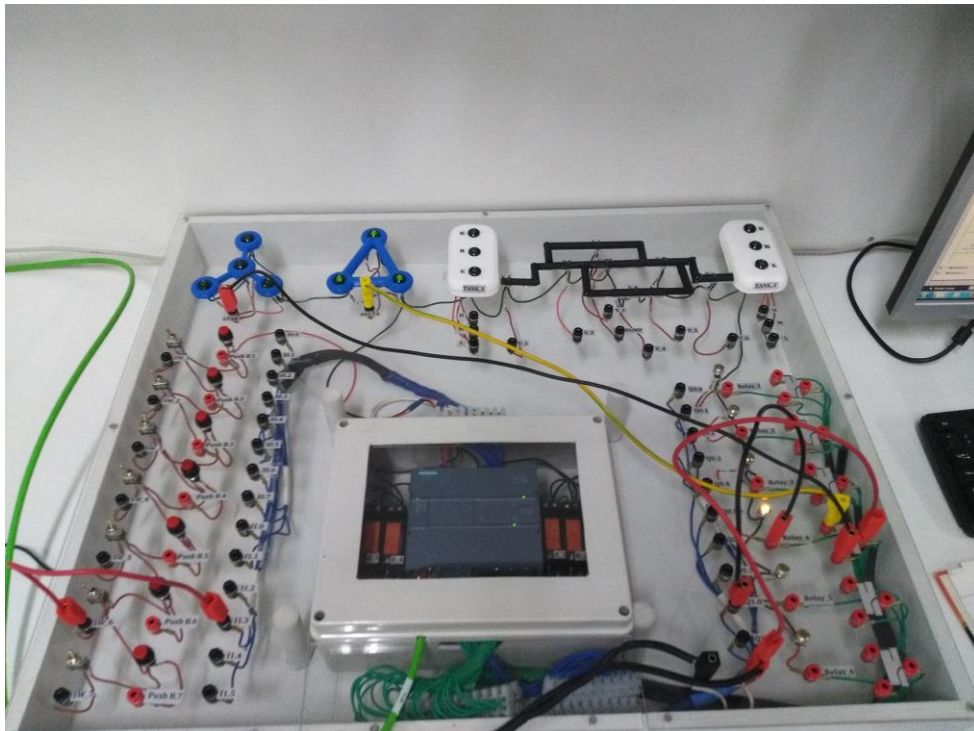


Εικόνα 172: Ενεργοποίηση emergency stop button.

Βήμα 2<sup>ο</sup>: Σε αυτό το βήμα με χρήση ενός ηλεκτρονόμου θα γίνει η εναλλαγή των *led* από αστέρα σε τρίγωνο. Κανονικά λόγω τριφασικού κινητήρα χρειαζόμαστε 3 ηλεκτρονόμους και συγκεκριμένη συνδεσμολογία των ηλεκτρονόμων με τα τυλίγματα του κινητήρα, για τη λειτουργία σε αστέρα και τρίγωνο. Λόγω μη ικανοποιητικού αριθμού κινητήρων στο εργαστήριο θα γίνει χρήση μόνο ενός ηλεκτρονόμου για την εναλλαγή των *led*. Σύμφωνα με τον κώδικα *Ladder* έχουμε μια είσοδο «I1.3» η οποία αντιστοιχεί σε ένα διακόπτη και μια έξοδο «Q1.0», η οποία αντιστοιχεί σε έναν ηλεκτρονόμο. Αρχικά, πριν ενεργοποιήσουμε το διακόπτη θέτοντας έτσι σε λειτουργία την έξοδο, έχουμε αναμμένο το *led* του αστέρα. Μόλις θέσουμε σε λειτουργία το διακόπτη, τότε ανάβει το *led* του ηλεκτρονόμου (δηλαδή δείχνει ότι έχει τεθεί σε λειτουργία ο ηλεκτρονόμος) και γίνεται η εναλλαγή από αστέρα σε τρίγωνο.



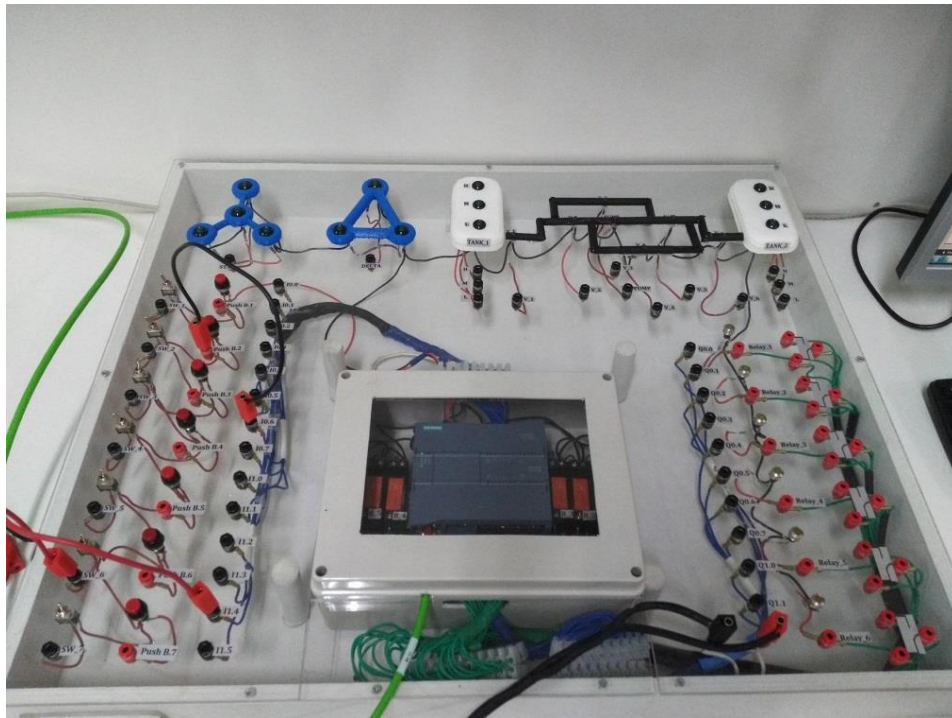
Εικόνα 173: Λειτουργία συνδεσμολογίας αστέρα με χρήση ενός ηλεκτρονόμου.



Εικόνα 174: Λειτουργία συνδεσμολογίας τρίγωνου με χρήση ενός ηλεκτρονόμου.

Βήμα 3<sup>ο</sup>, 4<sup>ο</sup> και 5<sup>ο</sup>: Σκοπός αυτών των βημάτων ήταν η μέτρηση των στροφών ενός κινητήρα με χρήση αισθητήρα στροφών, ο οποίος αντιστοιχεί σ' ένα κουμπί (με κάθε πάτημά του δίνει ένα σήμα). Λόγω αδυναμίας ύπαρξης πολλών κινητήρων στο εργαστήριο, η άσκηση

προσαρμόστηκε έτσι, ώστε να συνδέουμε μόνο τον αισθητήρα – *push button* στην είσοδο «I0.6», την οποία ορίσαμε στον κώδικα και την είσοδο «I1.4» και η οποία ενεργοποιεί το χρονόμετρο. Στα βήματα 3 και 4 έχουμε και άλλη μια είσοδο η οποία συνδέεται σε ένα διακόπτη για την ενεργοποίηση της μαθηματικής εξίσωσης, ώστε να δούμε τις συνολικές στροφές του «κινητήρα» (στη πραγματικότητα είναι το σήμα από το πάτημα του κουμπιού).



Εικόνα 175: Μέτρηση στροφών μηχανής με χρήση push button στη θέση του αισθητήρα.



## Κεφάλαιο 13<sup>ο</sup>: Συμπεράσματα

Κάνοντας μια γενική ανασκόπηση της πορείας της παρούσας διπλωματικής εργασίας, καταλήγουμε στο συμπέρασμα ότι οι προγραμματιζόμενοι λογικοί ελεγκτές (*PLC*) φέρανε την εξέλιξη τόσο στο χώρο του αυτοματισμού, όσο και στο χώρο της βιομηχανίας αφήνοντας πίσω τους μεγάλους πίνακες με τους ηλεκτρονόμους, αφού εξυπηρετούν λειτουργίες αυτοματισμών εξίσου αποτελεσματικά. Όπως είδαμε τα πλεονεκτήματά τους είναι αρκετά, κάτι το οποίο τους κάνει κατάλληλους για μια εγκατάσταση αυτοματισμού. Η χρήση της γλώσσας προγραμματισμού *Ladder* ήταν αρκετά εύκολη στη χρήση, σε σχέση με άλλες γλώσσες προγραμματισμού, διότι δεν χρειάζονται ιδιαίτερες γνώσεις, καθιστώντας την *Ladder* ιδανική για χρήση από άτομα τα οποία δεν είχαν προηγουμένως κάποια επαφή με άλλου είδους γλώσσες προγραμματισμού. Επιπλέον μπορούν και επικοινωνούν άψογα με λογισμικά προγραμματισμού, όπως το περιβάλλον εργασίας «*TIA Portal*», το οποίο χρησιμοποιήθηκε στην εκπόνηση της παρούσας εργασίας.

Οι εκπαιδευτικές μονάδες κατασκευάστηκαν και προσαρμόστηκαν να περιλαμβάνουν τις ασκήσεις, οι οποίες θα χρησιμοποιηθούν στα πλαίσια του μεταπτυχιακού προγράμματος του Πανεπιστημίου Δυτικής Αττικής του τμήματος ναυπηγών μηχανικών με τίτλο «Προηγμένες Τεχνολογίες στη Ναυπηγική και Ναυτική Μηχανολογία». Επομένως, για την προστασία των ελεγκτών και των φοιτητών, η εκμάθηση γίνεται με χρήση *led*, αλλά ο κώδικας ο οποίος θα κατασκευαστεί για τις ασκήσεις, θα είναι ίδιος με αυτόν που θα χρησιμοποιούσαμε σε μια πραγματική, παρόμοια κατάσταση. Παράλληλα ο σκοπός των ασκήσεων είναι να βοηθήσουν στην εκμάθηση του προγραμματισμού ενός αυτοματοποιημένου συστήματος με χρήση προγραμματιζόμενου λογικού ελεγκτή. Για το λόγο αυτό, οι εργαστηριακές ασκήσεις έχουν μια πιο γενική μορφή και δεν απευθύνονται μόνο σε ναυπηγούς μηχανικούς. Τέλος, στη παρούσα διπλωματική υπάρχει μια πιο αναλυτική εξήγηση του κώδικα που φτιάξαμε για την επίλυση των ασκήσεων με τίτλο «Εξοικείωση με το περιβάλλον *TIA Portal V13* και τη γλώσσα προγραμματισμού *Ladder*», «Έλεγχος στάθμης και μεταφοράς ρευστού σε δεξαμενές» και «Εκκίνηση τριφασικού κινητήρα», καθώς και της αντίστοιχης συνδεσμολογίας των εκπαιδευτικών μονάδων για την ομαλή λειτουργία του συστήματος, οι οποίες απέφεραν το επιθυμητό αποτέλεσμα.

## Κεφάλαιο 14<sup>ο</sup>: Προτάσεις

Γνωρίζοντας το «που συμβαίνει τι», μπορούμε να αντιδράσουμε γρήγορα και στοχευμένα με αποτέλεσμα την αύξηση της παραγωγής με χαμηλό κόστος. Αυτού του είδους ο έλεγχος θα μπορούσε να επιτευχθεί με διάφορους τρόπους, ένας από αυτούς είναι η χρήση συστήματος εποπτείας και συλλογής πληροφοριών (*Supervisory Control And Data Acquisition System – SCADA*). Επομένως, πέρα από τη συνδεσμολογία της μονάδας εκμάθησης και τον προγραμματισμό του προγραμματιζόμενου λογικού ελεγκτή (*Programming Logical Controller – PLC*), με τη χρήση του συστήματος *SCADA*, θα είχαμε μια πιο ρεαλιστική αναπαράσταση του κέντρου ελέγχου ενός μηχανοστασίου σε ένα πλοίο ή ενός εργοστασίου το οποίο κατά τη διάρκεια της παραγωγικής διαδικασίας χρησιμοποιεί αυτοματοποιημένα μέσα.

Επιπλέον θα μπορούσαν να διερευνηθούν περαιτέρω δυνατότητες των λειτουργιών του προγραμματιζόμενου λογικού ελεγκτή (*PLC*) S-1200 της εταιρίας *SIEMENS*, όπως η χρήση ελεγκτή *PID*, η δημιουργία και η χρήση ιστοσελίδας (*site*) για τον έλεγχο λειτουργίας του *PLC* εξ' αποστάσεως και άλλες πολλές λειτουργίες.

## Βιβλιογραφική Αναφορά

### Ηλεκτρονική

- <http://www.ford.gr>
- <https://en.wikipedia.org>
- <https://el.wikipedia.org>
- <https://cache.industry.siemens.com>
- <http://repository.cmu.edu>
- <http://www.industrialethernetu.com>
- <https://www.automationmag.com>
- <http://hlektrologia.gr>
- <https://www.theautomationstore.com>
- <http://www.contactandcoil.com>
- <http://plchowto.com>
- <http://www.triplc.com>
- <https://www.wisdomjobs.com>
- <http://www.contactandcoil.com>
- <https://www.allaboutcircuits.com>
- <http://www.chegg.com>
- <http://gun.teipir.gr>
- <https://www.justmakeit.gr>
- <http://auto.teipir.gr>
- <http://hlektrologia.gr>
- <http://hlektrologia.weebly.com>
- <http://www.jimkava.com>.
- <http://eclass.teipat.gr>
- «Ευστάθεια, Τύποι συστημάτων και Σφάλματα», (<http://eclass.teipat.gr>).
- Αξιολόγηση ενός συστήματος αυτόματου ελέγχου (κεφάλαιο 9), (<http://edume.myds.me>).
- Computer Aided Manufacturing TECH «Chaptre 5 Timers and Counters», ([www.personal.kent.edu](http://www.personal.kent.edu)).
- Liju G. Chacko «PLC Programming Language», ([www.slideshare.net](http://www.slideshare.net)).
- Δ. Καλλιγερόπουλος «Αναλυτική σύνθεση συστημάτων αυτομάτου ελέγχου», σεμινάριο αυτόματου ελέγχου, μάθημα 4, ([www.auto.teipir.gr](http://www.auto.teipir.gr)).

- [1] Jonathan D. Taylor «Robust Bode Methods for Feedback Controller Design of Uncertain Systems», Carnegie Mellon University, 2014.
- [2] Ιορδανίδου Δέσποινα «Τρισδιάστατοι εκτυπωτές: Λειτουργία και χρήση», ΕΑΠ, 2011.
- [3] Ευαγγελάκος Εμμανουήλ – Ανδρέας «Ανάπτυξη συστήματος SCADA για βιομηχανικές και ναυτικές εφαρμογές με χρήση PLC», Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας, 2017.
- [4] Φασιλής Α. & Χρηστίδης Β. «Αυτόματος Έλεγχος Με Χρήση Προγραμματιζόμενου Λογικού Ελεγκτή (PLC) – Εφαρμογή Του Στο Εργαστήριο Ναυτικών Μηχανών Και Μηχανολογίας», Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας, 2016.
- [5] Π. Ν. Παρασκευόπουλου «Συστήματα Αυτόματου Ελέγχου Βασικές Έννοιες Με Εφαρμογές», 1<sup>η</sup> έκδοση, Εκδόσεις: Π. Ν. Παρασκευόπουλου, Αθήνα, 2002.
- [6] Δ. Ν. Παγώνης «Αυτοματισμοί Πλοίου», Αθήνα, 2011.
- [7] Μ. Παπουτσιδάκης «Ελεγκτές - Controllers», Αθήνα, 2015.
- [8] Δρόσος Ε. και Ντουντουλάκης Α. «Μελέτη και εφαρμογές ελεγκτών», Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης, Χανιά, 2011.
- [9] Χ. Χαραλάμπους «Εισαγωγή στα συστήματα ελέγχου», Πανεπιστήμιο Κύπρου, 2003.
- [10] Ε. Παπαδόπουλος «Εισαγωγή στη Θεωρία και Τεχνολογία του Αυτόματου Ελέγχου», ΕΜΠ, 2006.
- [11] Αναστασία Βελώνη «Περί Ανάδρασης», Ανώτατο Εκπαιδευτικό Ίδρυμα Πειραιά, Αναστασία Βελώνη «Περί Ανάδρασης», Ανώτατο Εκπαιδευτικό Ίδρυμα Πειραιά, Αναστασία Βελώνη «Περί Ανάδρασης», Ανώτατο Εκπαιδευτικό Ίδρυμα Πειραιά.
- [12] Τσαγκαλίδου Ροδή «Προγραμματισμός Ηλεκτρονικών Υπολογιστών 2», Τεχνολογικό Εκπαιδευτικό Ίδρυμα Δυτικής Μακεδονίας, 2015.

**- Η σελίδα αυτή είναι σκοπίμως κενή -**

