

**ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Ανάπτυξη εφαρμογών αυτοματισμού με το λογισμικό STEP 7**

**Παναγιώτης Ν. Δημητρακόπουλος**

**Εισηγητής: Βελώνη Αναστασία, Καθηγήτρια**

**ΑΘΗΝΑ  
ΙΑΝΟΥΑΡΙΟΣ 2016**



**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Ανάπτυξη εφαρμογών αυτοματισμού με το λογισμικό STEP 7**

**Παναγιώτης Ν. Δημητρακόπουλος  
Α.Μ. 39275**

**Εισηγητής/ια:**

**Βελώνη Αναστασία, Καθηγήτρια**

**Εξεταστική Επιτροπή:**

**Γεώργιος Αρμένης, Καθηγητής  
Θεμιστοκλής Σιάντος, Επίκουρος Καθηγητής**

**Ημερομηνία εξέτασης \_\_/\_\_/2016**



## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/Η κάτωθι υπογεγραμμένος/η .....  
του ..... με αριθμό μητρώου .....  
φοιτητής/τρια του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά  
Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι  
ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του  
συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και  
πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται  
αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια  
πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα  
πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο  
συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και  
άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το  
Ίδρυμα του έχει απονεμίσει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης  
του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση  
του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και  
διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να  
ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία  
ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ.  
5 του ισχύοντος Εσωτερικού Κανονισμού.»



## ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, γεμάτες υπομονή και στήριξη πάνω σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό της δημιουργίας προσομοιώσεων για προγραμματιζόμενους λογικούς ελεγκτές με την χρήση του προγράμματος Step 7 από την Siemens. Θα ήθελα να ευχαριστήσω την κ. Βελώνη Αναστασία για την αποδοχή του αιτήματός μου και για την βοήθεια της σε αυτή την προσπάθεια.

Επιπλέον, θα ήθελα να ευχαριστήσω την οικογένειά μου για την στήριξη τους, τους συναδέλφους μου για την παροχή βοήθειας και πληροφοριών και τέλος τους φίλους μου που μου σταθήκανε όλα αυτά τα χρόνια.

Τέλος, θα ήθελα να ευχαριστήσω την εταιρεία Aventurine S.A. για την ευκαιρία που μου έδωσε να εργαστώ αυτά τα δύο χρόνια μαζί με την ομάδα της όπως και τον Θάνο Ψαρογένη, ιδιοκτήτη της ενημερωτικής ιστοσελίδας για τεχνολογία και παιχνίδια υπολογιστών, Alt-tab.gr για την θέση που μου πρόσφερε στην σελίδα καθώς και την ενημέρωση πάνω σε θέματα τεχνολογίας και προγραμματισμού.





## ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία επικεντρώνεται στην ανάπτυξη εφαρμογών αυτοματισμού με την χρήση του λογισμικού Simatic Step 7 της εταιρείας Siemens. Πάνω σε αυτό το πεδίο, έχουν στηριχθεί αρκετοί σύγχρονοι αυτοματισμοί που χρησιμοποιούνται στην καθημερινότητά μας καθώς και στην βιομηχανία γενικότερα. Το πρόγραμμα αυτό έχει αρκετά πλεονεκτήματα αλλά και μειονεκτήματα τα οποία καλούμαστε να αναλύσουμε. Το λογισμικό αυτό μας βοηθάει στην συγγραφή των αντίστοιχων εφαρμογών αυτοματισμού και στην κατάλληλη προσωμοίωσή τους.

Για την εργασία αυτή, μελετήθηκε η δομή αλλά και η λειτουργία των προγραμματιζόμενων λογικών ελεγκτών και έγινε αναφορά στις γλώσσες προγραμματισμού που χρησιμοποιούνται. Δίνονται οδηγίες για την εγκατάσταση, την παραμετροποίηση, την χρήση του λογισμικού και τέλος, μερικά παραδείγματα που μας βοηθούν να καταλάβουμε τον τρόπο λειτουργίας του λογισμικού αλλά και των βοηθητικών προγραμμάτων.

## ABSTRACT

The present thesis focuses on the developing process of automation application with the use of Simatic Step 7 from Siemens. On this field, several modern automations used in our life as well as in the current industry, are being supported by this software. The mentioned software has several advantages and disadvantages which we analyze. It helps us in writing and compiling the respective applications and also in their proper simulation.

For this thesis, we study the structure and function of the programmable logic controllers and the language used to program them. Instructions for the installation of the software are provided, some configurations and the way we can use the software. Lastly, we are using some examples in order to understand the way the software and its utilities operate.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Ανάπτυξη εφαρμογών, Προγραμματισμός  
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: λογισμικό, ελεγκτής, Step 7, ανάπτυξη εφαρμογών, PLC



## ΠΕΡΙΕΧΟΜΕΝΑ

<b>1.</b>	<b>ΕΙΣΑΓΩΓΗ.....</b>	<b>15</b>
1.1.	Περιγραφή του αντικειμένου της πτυχιακής εργασίας .....	15
1.2.	Ιστορική Αναδρομή .....	16
1.3.	Σχεδίαση και Προγραμματισμός.....	18
1.4.	Πλεονεκτήματα και μειονεκτήματα των PLCs .....	20
1.5.	Δομή ενός Προγραμματιζόμενου Λογικού Ελεγκτή .....	21
1.6.	Αρχή Λειτουργίας ενός PLC .....	24
1.7.	Λειτουργικά χαρακτηριστικά των PLC.....	26
1.8.	Διαδεδομένοι λογικοί ελεγκτές.....	30
1.9.	Προγραμματισμός των PLC .....	31
1.10.	Κυριότερες γλώσσες προγραμματισμού PLC .....	34
<b>2.</b>	<b>ΤΟ ΛΟΓΙΣΜΙΚΟ STEP 7 .....</b>	<b>37</b>
2.1.	Εγκατάσταση του STEP 7 .....	39
2.2.	Εγκατάσταση άδειας χρήσης .....	44
2.3.	Συνδέοντας την διεπαφή του PG/PC .....	45
2.4.	Απεγκατάσταση του STEP 7.....	47
<b>3.</b>	<b>ΔΗΜΙΟΥΡΓΙΑ ΝΕΩΝ ΕΡΓΑΣΙΩΝ ΣΤΟ STEP 7 .....</b>	<b>49</b>
3.1.	Περιβάλλον Εργασίας του SIMATIC Manager .....	49
3.2.	Δημιουργία νέας εργασίας.....	55
3.3.	Το λογισμικό PLCSim.....	63
<b>4.</b>	<b>ΠΑΡΑΔΕΙΓΜΑΤΑ .....</b>	<b>67</b>
4.1.	Παράδειγμα 1 <sup>ο</sup> : Χρονιστές.....	68
4.2.	Παράδειγμα 2 <sup>ο</sup> : Μετρητές .....	72
4.3.	Παράδειγμα 3 <sup>ο</sup> : Χρήση της κλήσης συναρτήσεων.....	78
4.4.	Παράδειγμα 4 <sup>ο</sup> : Δεξαμενή.....	82
4.5.	Παράδειγμα 5: Κυλιόμενη Σκάλα .....	86
4.6.	Παράδειγμα 6: Στάθμη Δοχείου .....	89
<b>5.</b>	<b>ΣΥΜΠΕΡΑΣΜΑΤΑ - ΠΡΟΟΠΤΙΚΕΣ.....</b>	<b>95</b>
5.1.	Πλεονεκτήματα και μειονεκτήματα χρήσης .....	96
5.2.	Προοπτικές .....	97
<b>6.</b>	<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>99</b>

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

<b>Σχήμα 1.1:</b> Περιβάλλον Ladder.....	<b>33</b>
<b>Σχήμα 1.2:</b> Μετατροπή προγράμματος LAD σε STL.....	<b>34</b>
<b>Σχήμα 1.3:</b> Λογικές πύλες ως Function Block Diagrams.....	<b>35</b>
<b>Σχήμα 4.1:</b> Δίκτυο 1 – S_PULSE.....	<b>71</b>
<b>Σχήμα 4.2:</b> Δίκτυο 2 – S_PEXT.....	<b>71</b>
<b>Σχήμα 4.3:</b> Δίκτυο 3 – S_ODT.....	<b>72</b>
<b>Σχήμα 4.4:</b> Δίκτυο 4 – S_ODTS.....	<b>72</b>
<b>Σχήμα 4.5:</b> Δίκτυο 5 – S_OFFDT.....	<b>73</b>
<b>Σχήμα 4.6:</b> Μετρητής Down με χρήση της FBD γλώσσας.....	<b>74</b>
<b>Σχήμα 4.7:</b> Δίκτυα 1 και 2 – Αρχική τιμή και μετρητής προς τα άνω.....	<b>76</b>
<b>Σχήμα 4.8:</b> Δίκτυο 3 – Μορφή μετρητή προς τα κάτω.....	<b>77</b>
<b>Σχήμα 4.9:</b> Τα δίκτυα κατά την διάρκεια λειτουργίας του Debug Mode.....	<b>78</b>
<b>Σχήμα 4.10:</b> Τα δίκτυα για την σύγκριση.....	<b>79</b>
<b>Σχήμα 4.11:</b> Τα δίκτυα του μπλοκ συναρτήσεων FC1.....	<b>80</b>
<b>Σχήμα 4.12:</b> Το δίκτυο του μπλοκ συναρτήσεων FC2.....	<b>81</b>
<b>Σχήμα 4.13:</b> Το βασικό Block του προγράμματος.....	<b>81</b>
<b>Σχήμα 4.14:</b> Η δεξαμενή TANK.....	<b>84</b>
<b>Σχήμα 4.15:</b> Δίκτυο 1 - Ενεργοποίηση PUMP 1.....	<b>85</b>
<b>Σχήμα 4.16:</b> Δίκτυο 2 - Ενεργοποίηση PUMP 2.....	<b>86</b>
<b>Σχήμα 4.17:</b> Δίκτυο 3 - Ενεργοποίηση MIXER.....	<b>87</b>
<b>Σχήμα 4.18:</b> Δίκτυο 4 - Ενεργοποίηση PUMP 3 και VALVE.....	<b>87</b>
<b>Σχήμα 4.19:</b> Δίκτυο 1 - Ενεργοποίηση σκάλας.....	<b>89</b>
<b>Σχήμα 4.20:</b> Δίκτυο 2 – Λειτουργία μετά από την έξοδο.....	<b>90</b>
<b>Σχήμα 4.21:</b> Δοχείο με όλα τα απαραίτητα στοιχεία.....	<b>91</b>
<b>Σχήμα 4.22:</b> Δίκτυο 1 – Είσοδος αναλογικής κάρτας.....	<b>92</b>
<b>Σχήμα 4.23:</b> Δίκτυο 2 – Η συνάρτηση αναλογικής κλιμάκωσης.....	<b>92</b>
<b>Σχήμα 4.24:</b> Δίκτυα 3 και 4 – Συγκρίσεις λίτρων.....	<b>93</b>
<b>Σχήμα 4.25:</b> Δίκτυα 3 και 4 – Συγκρίσεις λίτρων.....	<b>94</b>

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

<b>Πίνακας 3.1:</b> Λειτουργίες γραμμής κατάστασης του SIMATIC Manager.....	<b>54</b>
<b>Πίνακας 3.2:</b> Λειτουργίες κουμπιών στο SIMATIC Manager.....	<b>56</b>
<b>Πίνακας 3.3:</b> Βασικά εικονίδια του Project.....	<b>64</b>
<b>Πίνακας 3.4:</b> Κουμπιά γραμμής εργαλείων επεξεργαστή προγράμματος.....	<b>65</b>
<b>Πίνακας 3.5:</b> Βασικά κουμπιά γραμμής εργαλείων του προσωμοιωτή PLCSim...	<b>67</b>
<b>Πίνακας 4.1:</b> Συμβολική ονομασία στοιχείων μετρητή προς τα κάτω.....	<b>75</b>
<b>Πίνακας 4.2:</b> Σύμβολα δεξαμενής.....	<b>85</b>
<b>Πίνακας 4.3:</b> Σύμβολα κυλιόμενης σκάλας.....	<b>88</b>

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

**ALM** Automation License Manager

**CAN** Computer Automatic Network

**CD** Compact Disk

**CPU** Central Processing Unit

**DVD** Digital Versatile Disk

**EPROM** Erasable programmable Read-only memory

**MODICON** Modular Digital CONTroller

**MPI** Message Passing Interface

**PC** Personal Computer

**PS** Power Supply

**PLC** Programmable Logic Controller

**PLCSim** Programmable Logic Controller Simulation

**PROFIBUS** Process Field Bus

**RAM** Random-Access Memory

**RLS** Relay Logic Systems

**USB** Universal Serial Bus

## ΚΕΦΑΛΑΙΟ 1

### ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο αναλύεται το αντικείμενο ενασχόλησής μας και γίνεται μια ιστορική αναδρομή γύρω από την προέλευση του, τα διάφορα είδη και τις μεθόδους ανάπτυξης και προσομοίωσης που χρησιμοποιούνται σήμερα. Επιπλέον, ορίζεται το αντικείμενο μελέτης και προσδιορίζεται η σημασία του στην εκπαίδευση.

Αρχικά αναφερόμαστε στον ορισμό των P.L.C. και αναλύουμε όλα τα βασικά χαρακτηριστικά των εφαρμογών του, των λειτουργιών του καθώς και πλεονεκτήματα – μειονεκτήματα στον σύγχρονο αυτοματισμό και την βιομηχανία. Η εύρεση των πηγών για να μπορέσουμε να ορίσουμε και να περιγράψουμε το θέμα της εργασίας αυτής έγινε μέσω βιβλίων σχετικών με θεωρητικά αλλά και προγραμματιστικά στοιχεία για τους λογικούς ελεγκτές, με την βοήθεια επιστημονικών ιστοσελίδων αλλά και με την χρήση οπτικοακουστικού υλικού.

Τέλος, δίνονται αρκετές λεπτομέρειες πάνω στους λογικούς ελεγκτές έτσι ώστε να είμαστε σε θέση να κατανοήσουμε την λειτουργία τους, τον τρόπο προγραμματισμού τους και το μέσο προγραμματισμού δηλαδή το λογισμικό SIMATIC STEP 7.

#### 1.1. Περιγραφή του αντικειμένου της πτυχιακής εργασίας

Η παρούσα πτυχιακή εργασία αναλύει την δημιουργία και επεξεργασία προγραμμάτων στο λογισμικό SIMATIC STEP 7 της γερμανικής εταιρείας Siemens. Κύριο αντικείμενο είναι η παρουσίαση και η περιγραφή κάθε λειτουργίας του προγράμματος, η ανάδειξη ορισμένων δυνατοτήτων του και η δημιουργία παραδειγμάτων πάνω σε αυτό. Κατά την διάρκεια της ανάλυσής μας θα δούμε την σημασία του λογισμικού αυτού στα πλαίσια της εκπαιδευτικής διαδικασίας ενώ επιχειρείται η διευκόλυνση στην εκμάθηση του προγράμματος από τους μελλοντικούς τεχνικούς και αυτοματιστές.

Ο Προγραμματιζόμενος Λογικός Ελεγκτής (PLC – Programmable Logic Controller), είναι ένας ψηφιακός υπολογιστής που χρησιμοποιείται σε βιομηχανικές εφαρμογές αυτοματισμού. Τέτοιες είναι ο έλεγχος των μηχανών στις γραμμές συναρμολόγησης, στους διαδρόμους ή στους φωτεινούς σηματοδότες των

εργοστασίων. Τα PLCs χρησιμοποιούνται σε πολλές μηχανές που σχετίζονται με την βιομηχανία και είναι σχεδιασμένα για πολλαπλές ρυθμίσεις ψηφιακών και αναλογικών εισόδων και εξόδων, επεκτάσεις θερμοκρασιακών διακυμάνσεων, την αντιμετώπιση του ηλεκτρονικού θορύβου και αντοχή σε ζημιές και κραδασμούς. Αποτελούν συστήματα πραγματικού χρόνου διότι η ανταπόκριση μεταξύ της εισόδου και της εξόδου πρέπει να είναι άμεση για να μην υπάρχει ανεπιθύμητη λειτουργία. Είναι μια ειδική συσκευή η οποία έρχεται να αντικαταστήσει όλους τους βοηθητικούς ηλεκτρονόμους, τα χρονικά και τους απαριθμητές. Είναι μια ηλεκτρονική διάταξη με εισόδους και εξόδους, που συνδέονται με τα στοιχεία μιας εγκατάστασης και έναν αλγόριθμο που καθορίζει τους συνδιασμούς εισόδων που παράγουν ένα συγκεκριμένο αποτέλεσμα στις εξόδους. Με άλλα λόγια αποτελεί ένα ψηφιακό κύκλωμα που χρησιμοποιείται ευρέως στην βιομηχανία αφού διαθέτει επεξεργαστή και εσωτερική μνήμη στην οποία μπορούμε να φορτώσουμε πρόγραμμα για την εκτέλεση διάφορων λειτουργιών, αριθμητικών και μη.

## **1.2. Ιστορική Αναδρομή**

Στις πρώτες μέρες των αυτοματισμών, τα αυτόματα μηχανήματα και εργαλεία που χρησιμοποιούνται από τις επιχειρήσεις και τα εργοστάσια ήταν συνδεδεμένα με μηχανολογικά υλικά όπως μοχλούς, γρανάζια και άλλα. Ωστόσο, η εξέλιξη της τεχνολογίας βοήθησε στην ανάπτυξη του τομέα αυτού αφού οι τεχνικοί προσπαθούσαν από νωρίς να βρουν τρόπους για την σύνδεση των μηχανολογικών αυτοματισμών με τους ηλεκτρονικούς υπολογιστές. Από την δεκαετία του 1960 και μετά οι πρώτες εφαρμογές των υπολογιστών φάνηκαν με τις εργαλειομηχανές που χρησιμοποιούσαν κυρίως μηχανολογικούς αυτοματισμούς και λιγότερο ηλεκτρικούς. Λόγω όμως της ακρίβειας των υπολογιστών και της δυσκολίας χρήσης τους από τους τεχνικούς οι σκέψεις αυτές παρέμειναν ανεκμετάλευτες μέχρι τις αρχές του 1980.

Ανάμεσα στις δεκαετίες αυτές, οι υπολογιστές πέρασαν από διάφορα στάδια εξέλιξης με τους πρώτους μικρουπολογιστές να κατασκευάζονται το 1975. Ένα νέο κεφάλαιο ανοίγει στην τεχνολογία και την καθημερινή μας ζωή αφού η εφεύρεση αυτή εφαρμόζεται σε πολλούς τομείς και αλλάζει ριζικά τον αυτοματισμό. Σημειώνεται πως μέχρι το 1980 το μεγαλύτερο κομμάτι των αυτοματισμών



εξακολουθούσε να πραγματοποιείται με τους αυτοματισμούς ρελέ (relay) και οι ηλεκτρονικοί χρησιμοποιούνταν για άλλες έξυπνες εργασίες.



**Εικόνα 1.1:** Βιομηχανικό Ρελέ Λυχνίας

Την δεκαετία του 1980 και πιο συγκεκριμένα το 1986, η Hydra-Matic (το τμήμα αυτοματισμού της General Motors) εξέδωσε ένα αίτημα για την πρόταση και πιθανή δημιουργία ενός ηλεκτρονικού αντικατάστατου για ενσύρματα συστήματα ρελέ. Η πρόταση που επικράτησε ήταν εκείνη του Dick Morley της Bedford Associates. Έτσι δημιουργήθηκε το πρώτο PLC με όνομα 084 – μιας και αποτελούσε την 84<sup>ο</sup> δουλειά (project) της εταιρείας – μαζί με την έναρξη μιας νέας εταιρείας που είχε ως σκοπό την σχεδίαση, κατασκευή και πώληση του MODICON.

Για να μπορέσουμε όμως να καταλάβουμε την σημασία των PLCs στον Αυτοματισμό, πρέπει πρώτα να ορίσουμε τι είναι Αυτοματισμός. Εξάλλου, πολλές αυτόματες λειτουργίες γίνονταν αρχικά με συμβατικά υλικά (ρελέ, επαφές κλπ) τα οποία συνδέονταν μεταξύ τους ηλεκτρικά, με καλώδια.

Ο αυτοματισμός είναι η τυποποίηση μιας διαδικασίας μέσω της εύρεσης καλά ορισμένων βημάτων. Αυτά τα βήματα πρέπει να ακολουθηθούν κατάλληλα για να παραχθεί το επιθυμητό αποτέλεσμα. Με πιο απλά λόγια, ο αυτοματισμός είναι η εύρεση λύσης σε ένα πρόβλημα ή η δημιουργία ενός μηχανισμού που θα δέχεται κάποιον αλγόριθμο για την επίλυση ενός προβλήματος χωρίς ανθρώπινη παρέμβαση. Σήμερα όμως, αυτοματισμός είναι και η επιστήμη του μηχανισμού εκείνου που ασχολείται με τον έλεγχο διεργασιών και την διατήρησή τους σε μια συγκεκριμένη κατάσταση όπως είναι η διατήρηση της θερμοκρασίας ενός θερμοστάτη. Για την εκτέλεση αυτοματισμών, χρειάζονται ηλεκτρονόμοι (δηλαδή ηλεκτρονικές συσκευές). Το γεγονός αυτό αποτελεί και ένα μειονέκτημα αυτών των αυτοματισμών αφού η κατανάλωση ενέργειας είναι μεγάλη, ενώ ταυτόχρονα

ευθύνονται και για την παραγωγή μεγάλου ηλεκτρονικού θορύβου με αποτέλεσμα να μην είναι εύκολη η επιδιόρθωση των βλαβών. Έτσι λοιπόν, για να μπορέσουν οι μηχανές αυτοματισμού να λειτουργήσουν επιθυμητά, ο προγραμματισμός και η εφαρμογή ενός PLC έγινε μια απαραίτητη προϋπόθεση αλλά και διευκόλυνση γενικότερα.



**Εικόνα 1.2:** Κεντρικές Μονάδες PLC της Simatic

### 1.3. Σχεδίαση και Προγραμματισμός

Προκειμένου να σχεδιαστεί και να κατασκευαστεί ένας αυτοματισμός με PLC, υπάρχουν ορισμένα στάδια τα οποία πρέπει να ολοκληρωθούν. Αρχικά γίνεται η περιγραφή του αυτοματισμού που επιθυμούμε. Έπειτα γίνεται η ανάπτυξη του σχεδίου καλωδίωσης μαζί με την κατασκευή του πίνακα εγκατάστασης. Αφού ολοκληρωθούν αυτά τα κομμάτια, προχωρούμε στο κομμάτι του ελεγκτή με την ανάπτυξη του προγράμματος που θα χρειαστεί για την σωστή λειτουργία του αυτοματισμού. Μετά την επιτυχή εγγραφή και εισαγωγή του προγράμματος στον

ελεγκτή, και την εγκατάσταση των αισθητήρων και των λοιπών συσκευών, γίνεται δοκιμή της λειτουργίας του. Τα PLC λόγω του ότι δεν απαιτούν πολλά υλικά για την κατασκευή τους, χρειάζονται ελάχιστο χώρο για την εγκατάστασή τους και λιγότερο χρόνο για την κατασκευή τους σε σχέση με τους ηλεκτρονόμους.

Τα πρώτα PLCs σχεδιάστηκαν με σκοπό την αντικατάσταση των λογικών συστημάτων ρελέ (Relay Logic Systems). Τα συστήματα αυτά προγραμματίζονταν με την λογική Ladder, δηλαδή ενός διαγράμματος λογικής ρελέ. Αυτός ο τρόπος βοήθησε κυρίως στην μείωση του χρόνου εκπαίδευσης των υπαρχόντων τεχνικών και υπαλλήλων.

Σήμερα τα PLCs μπορούν να προγραμματιστούν με πολλούς τρόπους, από την απλή LADDER γλώσσα μέχρι και βασικές γλώσσες προγραμματισμού όπως η BASIC και η C. Δεν είχαν τερματικά προγραμματισμού τα οποία μπορούσαν να προσομοιώσουν την λειτουργία τους. Ωστόσο, με την εξέλιξη της βιομηχανίας και του προγραμματισμού, η λογική της LADDER γλώσσας άρχισε να χρησιμοποιείται περισσότερο. Ένας από τους κυριότερους λόγους είναι ότι τα PLCs λύνουν την λογική αυτή με έναν προβλεπόμενο τρόπο σε μια επαναλαμβανόμενη ακολουθία και έτσι η λογική της LADDER επιτρέπει στον προγραμματιστή να εντοπίσει προβλήματα στον συγχρονισμό. Μην ξεχνάμε πως ο χρόνος στην αυτοματοποίηση παίζει έναν πολύ μεγάλο και σημαντικό ρόλο.

Ο προγραμματισμός ενός ελεγκτή γίνεται ευκολότερα σήμερα. Η λογική ενός αυτοματισμού παρουσιάζεται με γραφική μορφή αντί με σύμβολα. Ο υπολογιστής συνδέεται στο μικροελεγκτή μέσω καλωδίωσης Ethernet, RS-232, RS-485, RS-422 και γίνεται σε αυτόν η φόρτωση του προγράμματος. Εκτός αυτού, μπορεί να γίνει επανέλεγχος και επιδιόρθωση του προγράμματος για την αντιμετώπιση τυχόν προβλημάτων. Το λογισμικό που χρησιμοποιείται για τα προγράμματα μπορεί να ανεβάσει και να κατεβάσει αντίστοιχα το πρόγραμμα ή να το σώσει σε περίπτωση κατακερματισμού.

Η λειτουργικότητα των ελεγκτών έχει εξελιχθεί με αποτέλεσμα να περιλαμβάνονται έλεγχοι για ρελέ, έλεγχος κίνησης και διεργασίας καθώς και δικτύωση. Η διαχείριση των δεδομένων, η αποθήκευση, η επεξεργαστική ισχύ και οι δυνατότητες επικοινωνίας κάποιων σύγχρονων ελεγκτών είναι περίπου ισοδύναμη με αυτή των επιτραπέζιων υπολογιστών. Ο λόγος όμως που οι σταθεροί επιτραπέζιοι υπολογιστές δεν χρησιμοποιούνται στην βιομηχανία έχουν να κάνουν κυρίως με την κατασκευή τους. Πιο συγκεκριμένα, οι υπολογιστές αυτοί δεν έχουν

σταθερό λειτουργικό σύστημα ενώ η κατασκευή τους δεν είναι τέτοια ώστε να αντέχουν μεγάλες θερμοκρασίες, υγρασία, δονήσεις και διάρκεια ζωής ίση με αυτή των λογικών ελεγκτών.

Μια από τις βασικότερες διαφορές μεταξύ ελεγκτών και υπολογιστών, εκτός από την αντοχή τους σε ακραίες συνθήκες, είναι η δυνατότητα καταχώρησης πολλών ρυθμίσεων εισόδου / εξόδου.

#### **1.4. Πλεονεκτήματα και μειονεκτήματα των PLCs**

Υπάρχουν πολλοί λόγοι για τους οποίους οι προγραμματιζόμενοι λογικοί ελεγκτές είναι πιο επιθυμητοί στην βιομηχανία σήμερα. Τα πλεονεκτήματα των ελεγκτών αυτών είναι περισσότερα σε σχέση με τα μειονεκτήματα και αυτό φαίνεται και από το γεγονός ότι αποτελούν καλύτερα συστήματα σε σχέση με τους ηλεκτρονόμους. Για να μπορέσουμε να καταλάβουμε όμως την σημασία τους, θα πρέπει να αναφερθούμε συγκεκριμένα σε αυτές τις διαφορές.

##### **1.1.1. Πλεονεκτήματα**

Ο προγραμματισμός των λογικών ελεγκτών γίνεται με αρκετά εύκολο τρόπο και ο χρήστης μπορεί να παρέμβει σε αυτόν χωρίς να χρειάζεται η επέμβαση του κατασκευαστή υλικού, δηλαδή μπορούμε να αλλάξουμε την λειτουργία του ελεγκτή χωρίς όμως να χρειαστεί να αλλάξουμε κάποιο κομμάτι του κυκλώματος. Περιέχει στην μνήμη του βοηθητικές επαφές, χρονικά, απαριθμητές, μετρητές και άλλα εξαρτήματα στα οποία έχει πρόσβαση ο χρήστης και μπορεί να παρέμβει. Τα διαγνωστικά μηνύματα λάθους αποτελούν μια επιπλέον βοήθεια για τον εντοπισμό σφαλμάτων και την έγκαιρη παρέμβασή.

Αναφερθήκαμε παραπάνω στον χώρο τον οποίο χρειάζονται τα PLC για να εγκατασταθούν. Τα κυκλώματα αυτά είναι κατασκευασμένα έτσι ώστε να μπορούν να προστατεύονται από αλλαγές στις συνθήκες εργασίας όπως την υγρασία, τις υψηλές ή χαμηλές θερμοκρασίες, τις δονήσεις κλπ. Γενικότερα, υπάρχει σημαντική οικονομία στον χώρο, την συντήρηση και την κατανάλωση ενέργειας. Η οικονομία στον χώρο δεν είναι το μοναδικό πλεονέκτημα όμως. Οι λογικοί ελεγκτές δεν χρειάζονται αρκετά χρήματα για να κατασκευαστούν λόγω του μικρού κόστους των επιμέρους υλικών.

Τέλος, παρέχονται δυνατότητες σύνδεσης με τον κεντρικό ηλεκτρονικό υπολογιστή και το ενδοεταιρικό δίκτυο

### **1.1.2. Μειονεκτήματα**

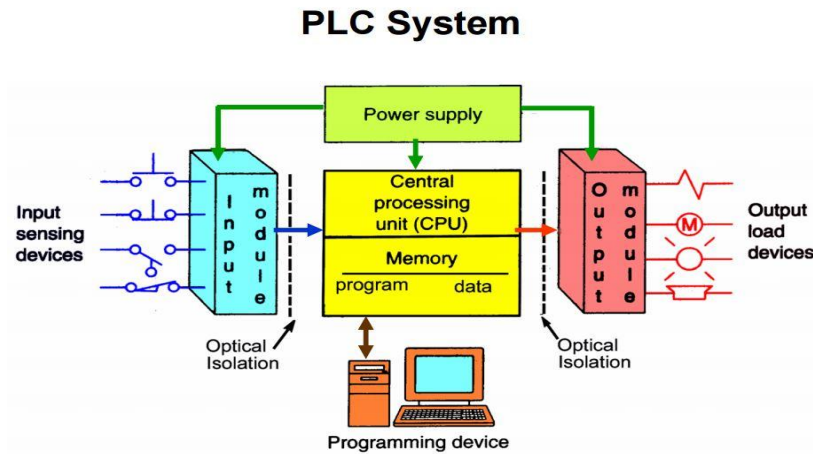
Υπάρχουν ορισμένα μειονεκτήματα τα οποία είναι αρκετά εμφανή σε ορισμένες περιπτώσεις. Για παράδειγμα, όταν έχουμε μια απλή εφαρμογή, η χρήση ενός προγραμματιζόμενου ελεγκτή είναι πιο ακριβή απ'οτι ένα πεδίο με ηλεκτρονόμους.

Εαν παρουσιαστεί μια βλάβη στον προγραμματιζόμενο ελεγκτή, είναι δυνατόν να μην επισκευάζεται και να χρειάζεται αντικατάσταση ενός τμήματος ή ολόκληρου του ελεγκτή, ενώ αν είχαμε ένα πεδίο με ρελέ θα ήταν αρκετό να αντικαταστήσουμε ένα μόνο από αυτά. Εξάλλου, οι λογικοί ελεγκτές είναι ευαίσθητοι στον ηλεκτρονικό θόρυβο, γεγονός που απαιτεί ειδικές κατασκευές και προστασίες.

Τέλος, η εγκατάσταση, η παρακολούθηση της λειτουργίας και η συντήρηση ενός προγραμματιζόμενου ελεγκτή απαιτεί εξειδικευμένο προσωπικό ή την εκπαίδευση του υπάρχοντος, πράγμα που σημαίνει αυξημένο κόστος συντήρησης.

### **1.5. Δομή ενός Προγραμματιζόμενου Λογικού Ελεγκτή**

Ένας προγραμματιζόμενος ελεγκτής αποτελείται από ορισμένα απαραίτητα στοιχεία τα οποία εξασφαλίζουν την ορθή λειτουργία του. Οι ελεγκτές αποτελούν έναν μικρό επεξεργαστή (ή αλλιώς έναν μικρό υπολογιστή) οπότε θα αποτελούνται από παρόμοια εξαρτήματα. Μέσα σε αυτά έχουμε, πλαίσιο για τοποθέτηση μονάδων (Rack), μονάδα τροφοδοσίας για να παρέχεται η απαραίτητη ενέργεια (PS), η κεντρική μονάδα επεξεργασίας στην οποία γίνεται η εκτέλεση του αλγορίθμου (CPU) και τέλος μονάδες εισόδων και εξόδων και η μονάδα προγραμματισμού.



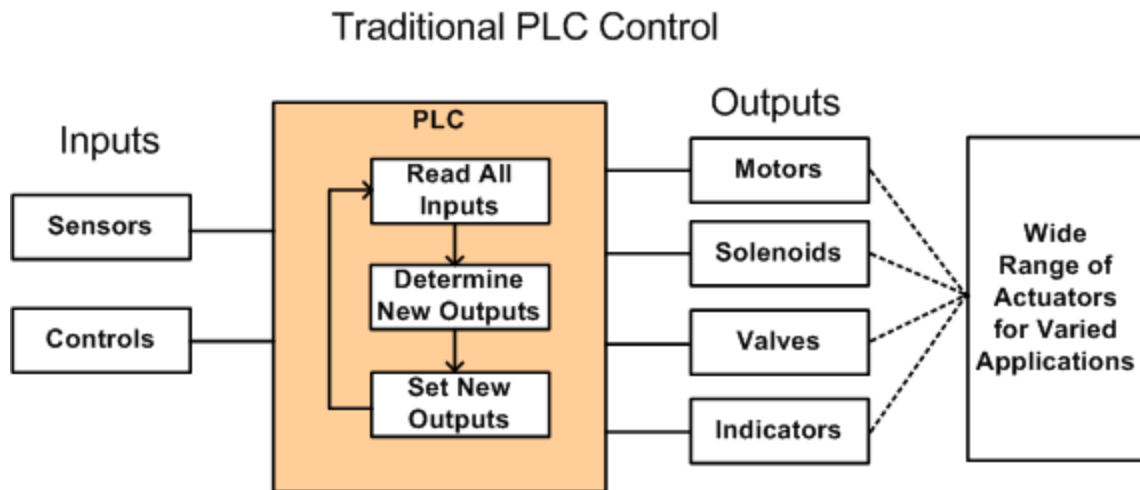
**Εικόνα 1.3:** Δομικό Διάγραμμα ενός PLC

Σε κάθε σύστημα ελέγχου υπάρχει ένας λογικός ελεγκτής που ενεργεί ως εγκέφαλος πίσω από την διαδικασία. Σε ένα τυπικό σύστημα ελέγχου PLC, ο ελεγκτής είναι μια ενιαία οντότητα με πολλές εισόδους και εξόδους όπου ταιριάζει σε μια ευρεία ποικιλία συσκευών και αισθητήρων.

Μόλις το PLC προγραμματιστεί με τις ιδιαίτερες απαιτήσεις της εφαρμογής και έχει συνδεθεί με όλες τις εισόδους και εξόδους του, είναι έτοιμο για τον έλεγχο. Το PLC θα επαναλάβει τυπικά τον επόμενο βρόχο καθώς επιτελεί ταυτόχρονα τα καθήκοντά του:

1. Διάβασμα όλων των εισόδων
2. Προσδιορισμών νέων εξόδων σύμφωνα με την παρούσα κατάσταση και τις παρούσες εισόδους
3. Θέτονται νέες εξόδους

Αυτό φαίνεται και στο σχήμα παρακάτω:



**Εικόνα 1.4:** Προσομοίωση ελέγχου PLC

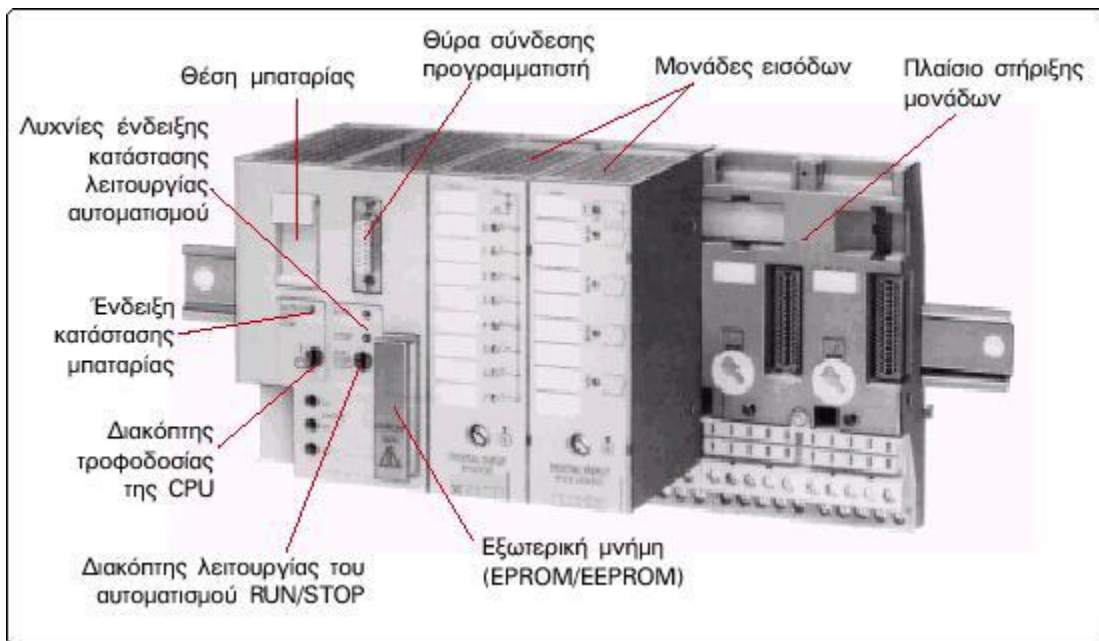
Για να μπορέσουμε όμως να καταλάβουμε πως λειτουργούν οι προγραμματιζόμενοι ελεγκτές και ποια είναι η σημασία των εξαρτημάτων από τα οποία αποτελούνται, πρέπει να αναφερθούμε στο καθένα ξεχωριστά:

- **Πλαίσιο για τοποθέτηση μονάδων (Rack):** Στην ουσία αποτελεί ένα «ντουλάπι» για τις μονάδες του PLC.
- **Μονάδα τροφοδοσίας:** Πρόκειται για εκείνο το σύστημα που τροφοδοτεί τον ελεγκτή με ηλεκτρικό ρεύμα. Οι μονάδες εισόδου/εξόδου μπορούν να λειτουργήσουν με συνεχή ή εναλλασσόμενη τάση τις τάξεως των: DC 24-48-60V και AC 24-48-115-230V.
- **Κεντρικός Επεξεργαστής/Μικροεπεξεργαστής:** Ο εγκέφαλος του συστήματος. Εκεί λαμβάνονται οι αποφάσεις που έχουν αποθηκευτεί σαν οδηγίες μέσα στην μνήμη. Το πρόγραμμα αποθηκεύεται σε αυτήν και έπειτα εκτελείται από τον μικροεπεξεργαστή. Τα σήματα που μπορούν να γίνουν δεκτά είναι χαμηλής τάσης και πολύ μικρού ρεύματος.
- **Συσκευές Εισόδου/Εξόδου:** Οι συσκευές εισόδου είναι απαραίτητες για την ανάγνωση των εισόδων και των δεδομένων που δέχεται ο ελεγκτής. Η εισαγωγή γίνεται στην είσοδο του κύριου κυκλώματος για παροχή εξισορόπησης τάσεως. Στις συσκευές αυτές συγκαταλέγονται αισθητήρες κλπ.

Από την άλλη, οι συσκευές εξόδου εκτελούν περίπου τις ίδιες λειτουργίες με αυτές των συστημάτων εισόδου, δηλαδή βρίσκονται εκεί για ηλεκτρική απομόνωση και

την εξισσορόπηση των τάσεων εξόδου. Σε αυτές τις συσκευές συγκαταλέγονται ρελέ, βαλβίδες και λυχνίες.

- **Μνήμη:** Όπως και τα περισσότερα επεξεργαστικά συστήματα, έτσι και τα PLC αποτελούνται από μια κεντρική μνήμη. Αυτές οι μνήμες ποικίλλουν σε μνήμες μόνο ανάγνωσης (ROM), μνήμες τυχαίας προσπέλασης (RAM), προγραμματιζόμενες μνήμες ανάγνωσης (EPROM) και Προγραμματιζόμενη μνήμη ανάγνωσης με δυνατότητα ηλεκτρικής διαγραφής (EEPROM).
- **Η κάρτα επικοινωνίας:** Για να μπορέσουμε να φορτώσουμε τον αλγόριθμό μας μέσα στο PLC πρέπει να βρούμε έναν τρόπο επικοινωνίας ανάμεσα στον ελεγκτή και τον υπολογιστή στον οποίο δημιουργήσαμε τον αλγόριθμο. Η βασικότερη κάρτα είναι η RS232 και είναι το πιο διαδεδομένο πρωτόκολλο επικοινωνίας.

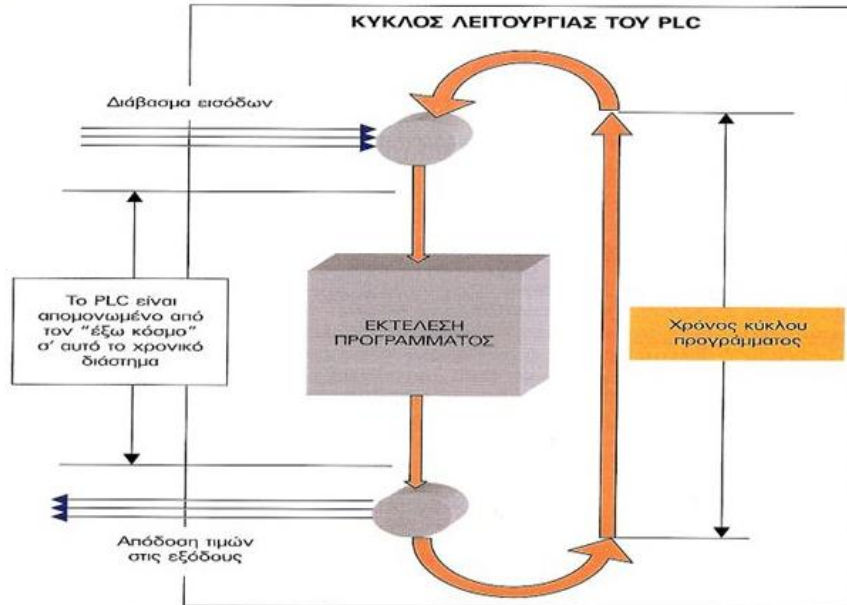


Εικόνα 1.5: PLC με τα βασικά του μέρη

### 1.6. Αρχή Λειτουργίας ενός PLC

Όταν ένα PLC βρίσκεται σε κατάσταση εκτέλεσης, υπάρχουν ορισμένα στάδια που ακολουθούνται κατά την διάρκεια της λειτουργίας. Τα βήματα αυτά αποτελούν τον κύκλο λειτουργίας του ελεγκτή και συνήθως είναι τρία (3). Ο κύκλος λειτουργίας είναι συνεχόμενος όταν βρισκόμαστε στην κατάσταση RUN.





**Εικόνα 1.6:** Κύκλος λειτουργίας PLC

Στην αρχή ο μικροεπεξεργαστής διαβάζει τις εισόδους, δηλαδή για κάθε είσοδο ελέγχεται αν υπάρχει υψηλή τάση (λογικό '1') ή χαμηλή τάση (λογικό '0'). Η τιμή '0' και '1' για κάθε είσοδο αποθηκεύεται σε μια ειδική περιοχή της μνήμης η οποία ονομάζεται εικόνα εισόδων. Η εικόνα αυτή αποτελεί έναν πίνακα στον οποίο ο μικροεπεξεργαστής σημειώνει τις τιμές που διάβασε.

Στην συνέχεια ο μικροεπεξεργαστής χρησιμοποιώντας σαν δεδομένα τις τιμές των εισόδων που διάβασε, εκτελεί τις εντολές του προγράμματος, σαν ένας απλός αυτοματισμός. Η εκτέλεση του προγράμματος θα δώσει αποτελέσματα για τις εξόδους που αποθηκεύεται στην εικόνα εξόδων. Όλες αυτές οι τιμές προκύπτουν από την εκτέλεση των λογικών πράξεων του προγράμματος.

Τέλος ο επεξεργαστής θέτει τις τιμές της εικόνας εξόδων στις διαθέσιμες εισόδους. Θα δοθεί υψηλή τάση σε όποια έξοδο έχει '1' και χαμηλή σε όποια έξοδο έχει '0'. Έτσι ολοκληρώνεται ένας κύκλος λειτουργίας.

Ο χρόνος που χρειάζεται για να γίνει ένας πλήρης κύκλος λέγεται χρόνος κύκλου και εξαρτάται από την ταχύτητα επεξεργαστή αλλά και το είδος των εντολών του προγράμματος. Ο χρόνος αυξάνεται αναλογικά ανάλογα με το μέγεθος του προγράμματος οπότε αν έχουμε μεγάλο πρόβλημα τότε έχουμε και μεγαλύτερο χρόνο λειτουργίας. Ο χρόνος αυτός αποτελεί και μέτρο σύγκρισης μεταξύ των υπάρχοντων PLC.

Η επικοινωνία του PLC με τον έξω κόσμο γίνεται μόνο κατά την διάρκεια του κύκλου λειτουργίας του δηλαδή όταν δέχεται εισόδους και στέλνει εξόδους. Την υπόλοιπη ώρα αποτελεί έναν μικρουπολογιστή που εκτελεί αυτόνομα πράξεις.

### **1.7. Λειτουργικά χαρακτηριστικά των PLC**

Με την εξέλιξη της τεχνολογίας σήμερα, πολλές συσκευές και τομείς που σχετίζονται με τον αυτοματισμό υπόκεινται στις αντίστοιχες βελτιώσεις και αναβαθμίσεις. Έτσι και τα PLC εξελίσσονται ολοένα και περισσότερο και μάλιστα με αρκετά γρήγορους ρυθμούς. Τα βασικά εξαρτήματα που αναφέραμε στο προηγούμενο κεφάλαιο είναι μόνο μερικά από τα στοιχεία που μπορεί να αποτελούν έναν σύγχρονο λογικό ελεγκτή. Για παράδειγμα, μπορεί να δούμε την λειτουργία απαριθμητών, στοιχείων σημαντικών στην απαρίθμηση εξωτερικών και εσωτερικών παλμών. Ο υπολογισμός αυτός όμως, δεν είναι ο μόνος μιας και τα σύγχρονα PLC έχουν την δυνατότητα να επεξεργάζονται αριθμητικές πράξεις. Αυτή η δυνατότητα αξιοποιείται από άτομα που γνωρίζουν από ψηφιακά συστήματα και μικρουπολογιστές.

Εκτός από την εισαγωγή ενός αλγορίθμου στην μονάδα του PLC, έχουμε και το εσωτερικό ρολόι μέσω του οποίου γίνεται να προγραμματίσουμε άμεσα κάποιες από τις εξόδους σε πραγματικό χρόνο, ημερομηνία και ώρα.

Οι βοηθητικές μεταβλητές που υπάρχουν για τον απλούστερο και ευκολότερο έλεγχο του προγράμματος που φτιάχνουμε αντιστοιχούν σε ρελέ κλασσικού αυτοματισμού τα οποία μπορούμε να χρησιμοποιήσουμε χωρίς πρόβλημα. Οι λογικοί ελεγκτές είχαν ως σκοπό την αντικατάσταση των αναλογικών αυτοματισμών – δηλαδή των αυτοματισμών ρελέ – και το πετυχαίνουν. Σήμερα οι δυνατότητές τους είναι πολύ περισσότερες αφού γίνεται η προσπάθεια να καλυφθούν πλήρως τα συστήματα αυτομάτου ελέγχου. Για να γίνει αυτό, οι προγραμματιζόμενοι ελεγκτές δέχονται και επεξεργάζονται αναλογικές, τις μετατρέπουν σε ψηφιακές τιμές για να είναι ευκολότερη η επεξεργασία των διαφόρων ψηφιακών αριθμών και παράγουν αντίστοιχες αναλογικές εξόδους.

Τέλος, ένα ακόμα σημαντικό χαρακτηριστικό των σύγχρονων PLC είναι το γεγονός ότι μπορούν να επικοινωνούν μεταξύ τους ανταλλάσσοντας πληροφορίες, μέσω αντίστοιχης δικτύωσης. Το ίδιο μπορεί να γίνει και για επικοινωνία με ηλεκτρονικούς υπολογιστές, οι οποίοι ασχολούνται με τον έλεγχο όλης της

βιομηχανικής παραγωγής. Η δικτύωση αυτή ονομάζεται Βιομηχανικό Δίκτυο Αυτοματισμού (CAN).

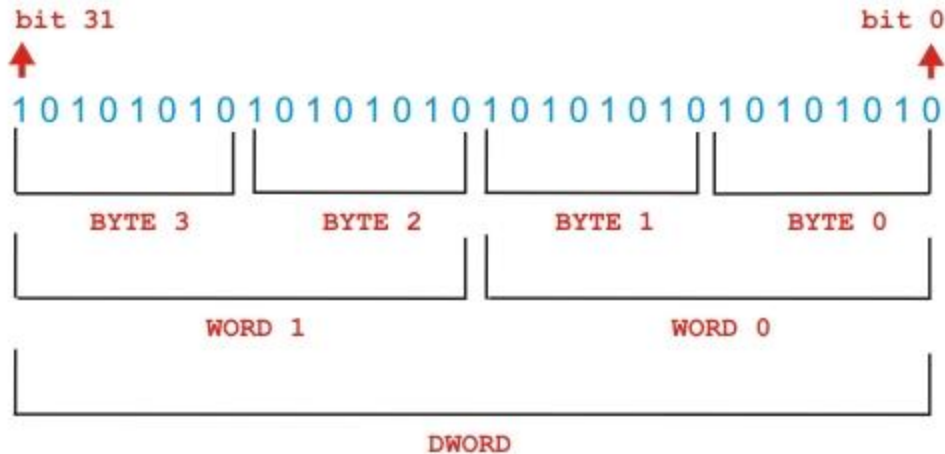
Όταν φτάσουμε στην επιλογή ενός PLC υπάρχουν κάποια κριτήρια επιλογής τα οποία λαμβάνονται υπόψιν. Ο χρήστης έχει να επιλέξει ανάμεσα σε μια μεγάλη ποικιλία υλικών και τεχνικών χαρακτηριστικών που αποτελούν έναν ελεγκτή. Αυτά είναι κυρίως τεχνοοικονομικά, δηλαδή έχουν να κάνουν με τα τεχνικά χαρακτηριστικά ενός ελεγκτή, το σύνολο εισόδων και εξόδων, την τιμή αγοράς και την τιμή των εξαρτημάτων ενώ θα πρέπει να καλύπτουν και τις ανάγκες του αγοραστή. Για παράδειγμα, σε μια εγκατάσταση ανάλογα με τα υλικά που καλούνται να καλυφθούν, το σύνολο των εισόδων και των εξόδων ποικίλλει και ο υπολογισμός τους προκύπτει από την ύπαρξη στοιχείων όπως Διακόπτες, Λυχνίες, Σειρήνες κλπ.

Η επιλογή της CPU γίνεται εξίσου ανάλογα με τον αριθμό των εισόδων/εξόδων ενώ αξιόπιστη απάντηση για το πόση μνήμη θα χρειαστούμε για την ομαλή λειτουργία ενός συστήματος δεν υπάρχει. Αυτό κυρίως υπόκειται στην εμπειρία του χρήστη κάτι όχι και τόσο αρνητικό αφού η κοστολόγηση της μνήμης σήμερα είναι πολύ μικρότερη σε σχέση με το παρελθόν.

### **1.1.3. Διευθυνσιοδότηση**

Στην παρούσα μελέτη, θα κάνουμε λόγο για τις ψηφιακές κάρτες του Simatic οπότε θα πρέπει να μελετήσουμε ορισμένες έννοιες όπως εκείνες των Bit, Byte, Word και Doubleword. Η σημασία τους βρίσκεται στο γεγονός ότι αποτελούν πληροφοριακές μονάδες για τις κάρτες εισόδων των PLC.

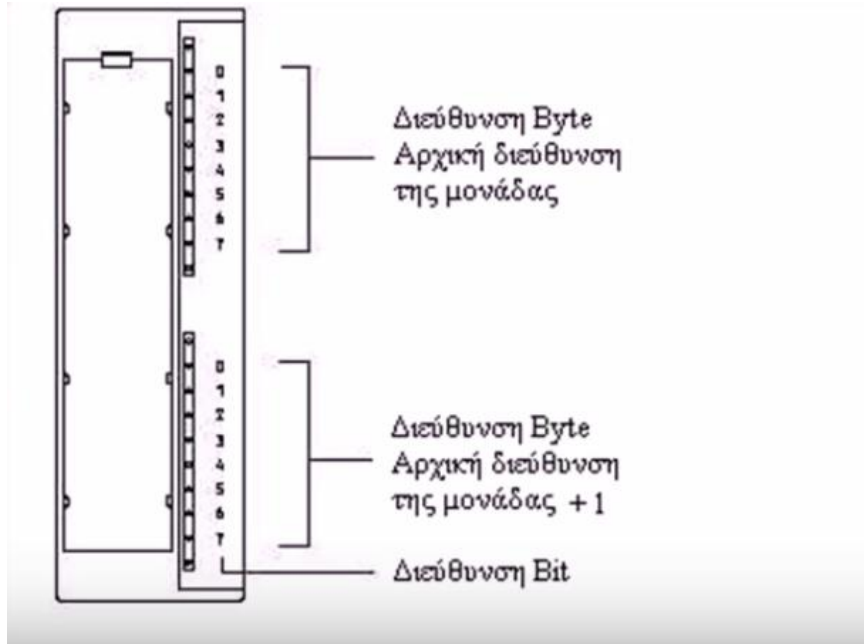
Το Bit είναι η μικρότερη πληροφοριακή μονάδα και μπορεί να έχει μόνο δύο τιμές 0 και 1. Έστω ότι έχουμε μια κάρτα εισόδων με δύο τερματικούς διακόπτες με τον ένα να έχει τάση ενώ ο άλλος όχι. Ο Διακόπτης με τιμή 0 δεν έχει τάση και ο διακόπτης με τιμή 1 έχει.



Εικόνα 1.7: Διευθυνσιοδότηση

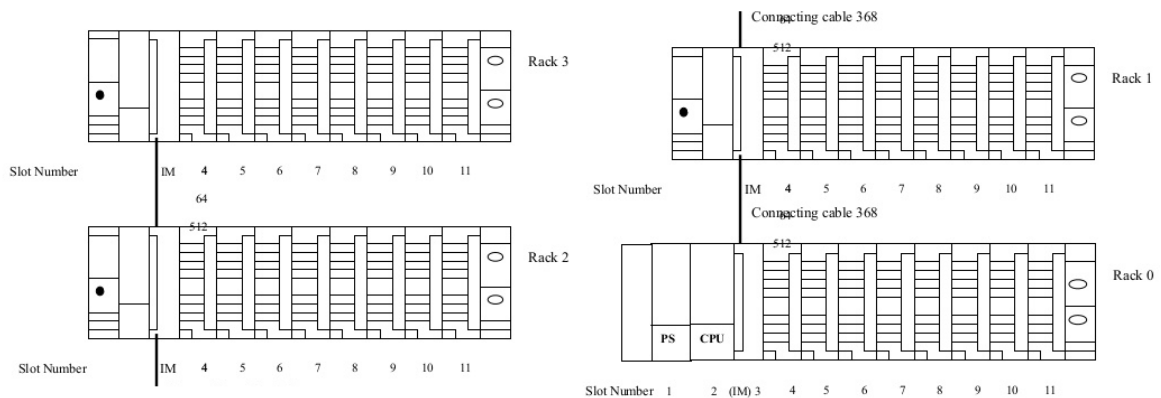
- **Bit:** είναι η στοιχειώδης μονάδα πληροφορίας στην Επιστήμη Υπολογιστών και στις Τηλεπικοινωνίες. Είναι η ποσότητα της πληροφορίας που μπορεί να αποθηκευτεί από μία δυαδική συσκευή, ή από άλλο φυσικό σύστημα το οποίο μπορεί να υπάρχει σε μία από δύο διακριτές καταστάσεις. Σε ένα PLC ή κάποιο άλλο κύκλωμα αποτελεί τις δυο τάσεις του ηλεκτρικού ρεύματος.
- **Byte:** Το byte είναι και η βασική μονάδα μέτρησης (χώρου και πληροφορίας) στα υπολογιστικά συστήματα. Ένα byte ισοδυναμεί με 8 bit.
- **Word:** Στους υπολογιστές, η “λέξη” είναι μια μονάδα μέτρησης που αποτελείται από 16 bit ή 2 bytes.
- **Doubleword (DWord):** Αντίστοιχα η “διπλολέξη” είναι μια μονάδα μέτρησης που αποτελείται από δύο λέξεις δηλαδή 32 bit ή 4 bytes.

Η αρίθμηση ξεκινάει πάντα από τα δεξιά προς τα αριστερά, από το μηδενικό bit και φτάνει το 7 στα bit, το 15 στις Word και το 31 στις DWord. Οι ψηφιακές κάρτες στο Simatic είναι οργανωμένες κατά byte. Η αρίθμηση σε μια κάρτα ψηφιακών εισόδων ξεκινάει από κάτω από την διεύθυνση του bit που είναι κάθε επιμέρους είσοδος της κάρτας. Για να προσδιοριστεί η είσοδος πρέπει να προσδιοριστεί σε ποια οχτάδα βρίσκεται και σε ποιο ακριβώς σημείο της.



**Εικόνα 1.8:** Διευθύνσεις ψηφιακής κάρτας

Κάθε θέση στην σειρά S7 έχει σταθερή διεύθυνση με με συγκεκριμένη τοποθέτηση συνοδευόμενη από τον αριθμό της θέσης. Η διεύθυνση της θέσης σε μια κάρτα της σειράς S7-300 φαίνεται παρακάτω:



**Εικόνα 1.9:** Διευθυνσιοδότηση ψηφιακής κάρτας S7 - 300

Πέρα από την διεύθυνση της θέσης, μια μονάδα έχει και μια αρχική διεύθυνση που καθορίζει την θέση της στον χώρο των λογικών διευθύνσεων, τον χώρο δηλαδή εισόδων και εξόδων. Αυτές οι διευθύνσεις ξεκινούν από το μηδέν και φτάνουν σε μια διεύθυνση που εξαρτάται από την CPU που χρησιμοποιείται.

Οι μνήμες των ελεγκτών είναι οργανωμένες κατά bytes και κάθε πληροφορία αποθηκεύεται σε διαφορετική τοποθεσία της μνήμης η οποία έχει και μοναδική

διεύθυνση. Όταν θέλουμε να προσπελάσουμε μια διεύθυνση μνήμης, αρκεί να προσδιορίσουμε την επιθυμητή διεύθυνση. Γενικά, πρέπει να προσδιορίζουμε τις φυσικές εισόδους και εξόδους καθώς και πως ονομάζονται αυτές. Κάθε στοιχείο που χρησιμοποιούμε έχει και διαφορετική διεύθυνση μνήμης οπότε η προσπέλασή τους γίνεται με αποκλειστικό τρόπο.

### 1.8. Διαδεδομένοι λογικοί ελεγκτές

Στην σύγχρονη αγορά υπάρχουν δύο τύποι προγραμματιζόμενων λογικών ελεγκτών PLC. Αυτά είναι τα Compact και Modular PLC.

**Compact PLC:** Στα Compact (Συμπαγή) PLC ανήκουν όλοι οι ελεγκτές που έχουν τα επιμέρους στοιχεία τους ενσωματωμένα σε μια συσκευή. Έχουν περιορισμένες δυνατότητες αφού έχουν σαρανταοχτώ(48) εισόδους και εξόδους με παλιά χαρακτηριστικά αλλά και μικρό αριθμό χρονικών απαριθμητών. Τα παλαιότερα μοντέλα δεν ήταν επεκτάσιμα ενώ στα νεότερα υπάρχει δυνατότητα περιορισμένης επέκτασης. Ως πλεονέκτημά τους χαρακτηρίζεται το χαμηλό κόστος.



Εικόνα 1.10: Compact PLC

**Modular PLC:** Η μονάδα κάθε PLC είναι ξεχωριστή και συνδέονται όλες μαζί στο πλαίσιο τοποθέτησης μονάδων. Είναι επεκτάσιμα και χρησιμοποιούνται συνήθως όταν έχουμε μεγάλο αριθμό εισόδων και εξόδων. Αυτό έχει ως αποτέλεσμα να

μπορούμε να διαλέξουμε την κεντρική μονάδα και τις μονάδες εισόδων και εξόδων με τα χαρακτηριστικά που επιθυμούμε.

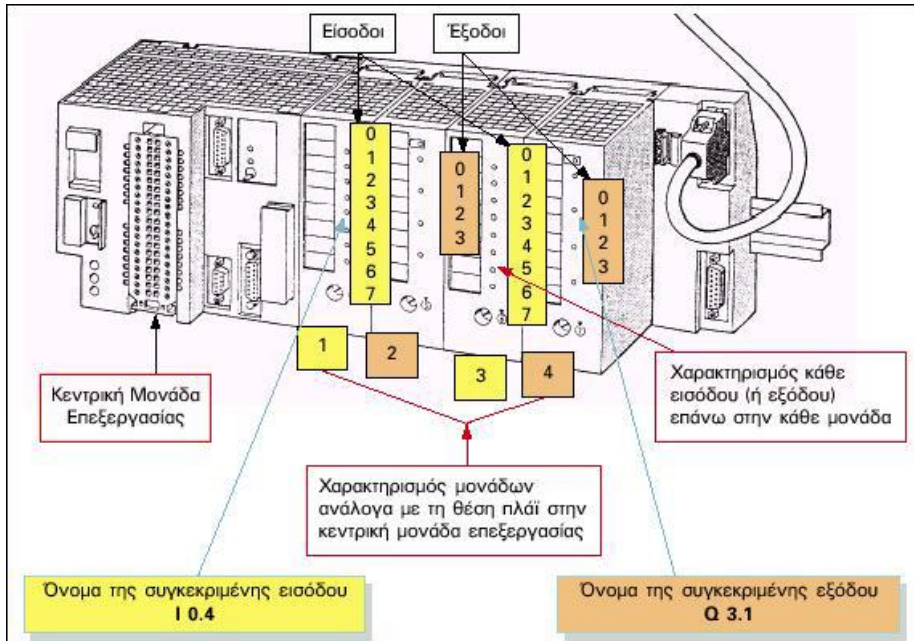


**Εικόνα 1.11:** Modular PLC

### **1.9. Προγραμματισμός των PLC**

Αναφέραμε προηγουμένως πως η επιλογή των PLC εξαρτάται από διάφορα κριτήρια. Η επιλογή αυτή δεν γίνεται μόνο για τον σκοπό λειτουργίας αλλά και για τον προγραμματισμό. Για να γίνουμε όμως πιο συγκεκριμένοι, υπάρχουν ορισμένοι παράγοντες που θα πρέπει να προσέξουμε πριν προγραμματίσουμε έναν ελεγκτή.

Το βασικότερο στοιχείο που πρέπει να γνωρίζουμε πριν τον προγραμματισμό είναι πόσες εισόδους έχουμε. Οι εισοδοί αυτές, ανάλογα το είδος του ελεγκτή, έχουν διαφορετικό διακριτικό και όνομα. Στα μικρά PLC, οι εισοδοί ξεκινούν με το γράμμα I (Input) και ακολουθούνται από έναν αριθμό που δηλώνει και την θέση της εισόδου στο πλήθος (I0, I1, I2 ...). Από την άλλη, τα μεγαλύτερα modular PLCs έχουν το γράμμα I ενώ ακολουθούνται από δύο αριθμούς χωρισμένους με μια τελεία. Ο διαχωρισμός αυτός δηλώνει πως ο πρώτος αριθμός είναι η θέση της μονάδας ενώ ο δεύτερος την είσοδο πάνω στην μονάδα. Τα ίδια που ισχύουν για τις εισόδους, ισχύουν και για τις εξόδους με την μόνη διαφορά χρησιμοποιείται το γράμμα O. Να σημειωθεί πως μιας και μιλάμε για ψηφιακές εισόδους και εξόδους η ονοματολογία μπορεί να αλλάξει προσθέτοντας απλά ένα D μπροστά από το I και Q.



**Εικόνα 1.12:** Ονοματολογία σε Modular PLC

Εκτός από το πλήθος των εισόδων, πρέπει να γνωρίζουμε και το πλήθος των βοηθητικών μνημών που υπάρχουν. Αυτές θα τις βρούμε με την ονομασία Markers ή Flags και εκεί γίνεται η αποθήκευση ενδιάμεσων λογικών καταστάσεων και πληροφοριών. Εδώ η ονομασία γίνεται με το γράμμα M/F/W μαζί με έναν ή δύο αριθμούς χωριζόμενους από τελεία.

Τέλος, θα πρέπει να γνωρίζουμε πληροφορίες για τις ειδικές συναρτήσεις και την διευθυνσιοδότησή τους. Κυριότερα χρειάζεται να ξέρουμε ποιες είναι, πως ονομάζονται, πως τις χειρίζεται το PLC και πόσες διαθέτει από την κάθε μίας. Οι ειδικές συναρτήσεις κατά σειρά σπουδαιότητας είναι:

- **Τα βοηθητικά (Memory Bit) – Mx.y:** Υπάρχουν περιπτώσεις που θα χρειαστεί να επαναλάβουμε ένα κομμάτι του κώδικα. Αυτή η επανάληψη ίσως χρειαστεί να γίνει πολλές φορές συνεχόμενα. Αν γίνει χειροκίνητα τότε κινδυνεύουμε να δαπανήσουμε αρκετό χρόνο καθώς και χώρο στην μνήμη. Έτσι χρησιμοποιώντας τα βοηθητικά, μπορούμε να επαναλάβουμε τον κώδικά μας όσες φορές χρειάζεται καταγράφοντας μία φορά την λογική. Το Memory bit χαρακτηρίζεται από δύο μεταβλητές – σε ποια οκτάδα ανήκει και τα όρια της στις επιμέρους θέσεις [Mx.y]



- **Οι απαριθμητές (Counters) - Cx:** Μας δίνεται η δυνατότητα αξιοποίησης λειτουργιών απαρίθμησης απευθείας από τον κεντρικό επεξεργαστή. Μετρούν προς τα πάνω και προς τα κάτω από 0 έως 999. Για να γίνει η μέτρηση από τον επεξεργαστή, πρέπει να παρατηρηθεί αλλαγής κατάστασης στην είσοδο, από 0 σε 1. Το X είναι ο αριθμός του απαριθμητή ανάλογα με την CPU που χρησιμοποιείται.
- **Τα χρονικά (timers) – Tx:** Απαραίτητα για την υλοποίηση αλγορίθμων που έχουν σχέση με χρόνο (Επιπήρηση, αναμονή, δημιουργία παλμών κλπ). Τα χρονικά χρησιμοποιούνται συνήθως στην διαχείριση λειτουργιών που έχουν να κάνουν με την χρονική διάρκεια. Για παράδειγμα, εκκίνηση ενός αυτοματισμού δέκα δευτερόλεπτα μετά από την εκκίνηση ενός άλλου.
- **Συγκριτές**
- **Γεννήτριες Παλμοσειρών**
- **Μετρητής Πραγματικού Χρόνου**

#### 1.1.4. Στάδια προγραμματισμού PLC

Πριν προχωρήσουμε στον προγραμματισμό ενός Προγραμματιζόμενου Λογικού Ελεγκτή ακολουθούμε ορισμένα στάδια όπως θα κάναμε για την λύση ενός λογικού ή μαθηματικού προβλήματος. Τα στάδια αυτά είναι αρκετά απλά αλλά χρειάζονται προσοχή ώστε να καταφέρουμε να προγραμματίσουμε σωστά και επιθυμητά την λειτουργία του ελεγκτή.

Πρώτα, διατυπώνουμε πλήρως το πρόβλημα και όσο πιο κατανοητά γίνεται. Έτσι θα μπορέσουμε να καταλάβουμε ποια είναι τα ζητούμενα και θα βρεθεί πιο εύκολα η λύση. Αφού διατυπωθεί κατάλληλα το πρόβλημα, καθορίζουμε τα στοιχεία εισόδων και εξόδων και έπειτα τα επικαλούμαστε για την κατασκευή του πίνακα αληθείας. Μέσω του πίνακα αληθείας κατασκευάζουμε τις λογικές εξισώσεις που προκύπτουν και τις απλοποιούμε μέσω του πίνακα Karnaugh ή της άλγεβρας Boole.

Ακολουθώντας όλα τα παραπάνω βήματα είμαστε έτοιμοι να ξεκινήσουμε τον προγραμματισμό μέσω της γλώσσας προγραμματισμού της επιλογής μας. Μετά την σχεδίασή του, ακολουθεί η σχεδίαση του λογικού διαγράμματος FUNCTION CHART για την κατασκευή του πίνακα αντιστοιχιών ή για την κωδικοποίηση των

στοιχείων εισόδου-εξόδου. Ολοκληρώνουμε με την συγγραφή του προγράμματος και την εισαγωγή του στην μνήμη του PLC για την εκκίνηση της λειτουργίας του.

Για να μπορέσουμε να καταλάβουμε καλύτερα τον προγραμματισμό ενός PLC πρέπει να κατανοήσουμε τις αρχές στις οποίες βασίζεται η λειτουργία του. Κάθε ενέργεια του PLC γίνεται μέσω των εντολών που καταχωρούμε στην κεντρική του μνήμη. Στην πραγματικότητα το πρόγραμμα εφαρμογής είναι μια σειρά οδηγιών που διαδέχονται η μία την άλλη και εκτελούνται κυκλικά (δηλαδή γίνεται επανάληψη της σειράς από την αρχή).

Ο ελεγκτής είναι σε θέση να εκτελέσει απλές λογικές πράξεις (AND, OR, NOT) ενώ μπορεί να προγραμματιστεί με τις αντίστοιχες εντολές ώστε να εκτελέσει και επιπρόσθετε πράξεις όπως XOR κλπ. Μέσα στο κύκλωμα υπάρχει ο καταχωρητής RESULT REGISTER στον οποίο αποθηκεύουμε την κατάσταση 0,1 οποιασδήποτε εισόδου ενώ μπορούμε να διατηρήσουμε το αποτέλεσμα κάποιας πράξης μέσα σε αυτόν.

### 1.10. Κυριότερες γλώσσες προγραμματισμού PLC

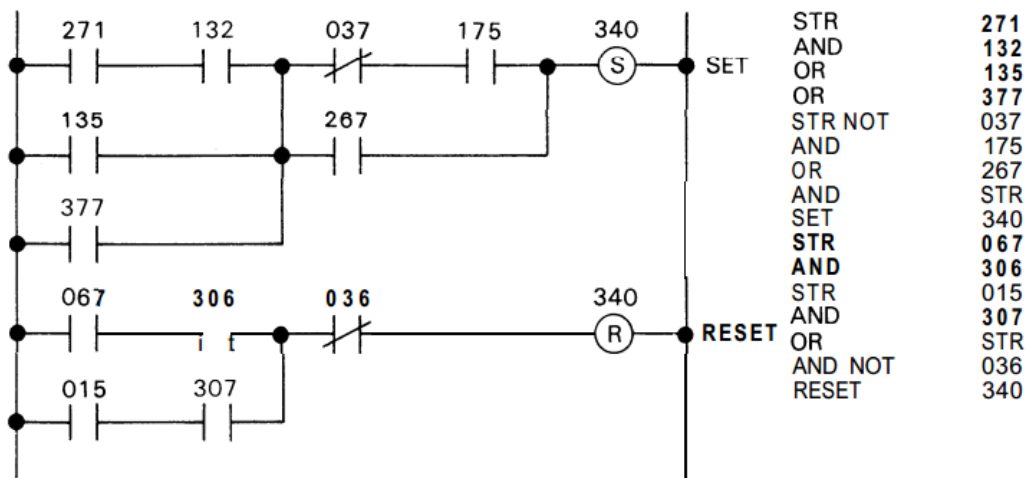
Η πρώτη γλώσσα προγραμματισμού είναι η LADDER (LAD), μια γλώσσα γραφικών που χρησιμοποιεί ηλεκτρομηχανικά σύμβολα και επιτρέπει την μεταφορά ηλεκτρολογικών σχεδίων στον ελεγκτή. Η γλώσσα αυτή είναι αρκετά εύκολη στην χρήση και κατανοητή οπότε χρησιμοποιήθηκε αρκετά για την εκπαίδευση των τεχνικών στην βιομηχανία.

Τα δύο πιο διαδεδομένα λογισμικά εκμάθησης είναι το LogixPRO και το SIMATIC Manager της Siemens.

Η γλώσσα που θα χρησιμοποιήσουμε σε μεγάλο βαθμό είναι η **LADDER**. Η γλώσσα αυτή έχει ένα ευνόητο περιβάλλον και είναι η πιο διαδεδομένη γλώσσα προγραμματισμού στην βιομηχανία αλλά και στην σύγχρονη εκπαίδευση.

Η Ladder αποτελεί μια γλώσσα που χρησιμοποιείται ευρέως στον προγραμματισμό των PLCs, εκεί όπου ο διαδοχικός έλεγχος της βιομηχανικής διαδικασίας ή των μηχανημάτων είναι απαραίτητος. Είναι αρκετά απλή και εύκολη στην εκμάθησή της. Η ίδια η γλώσσα μπορεί να θεωρηθεί ως ένα σύνολο συνδέσεων μεταξύ λογικών ελεγκτών (επαφές – τετραγωνικό, παράλληλο σχήμα) και ενεργοποιητών (τηνία – εληπτικό σχήμα). Το όνομα της γλώσσας αυτής

προέρχεται από την απεικόνισή της αφού κάθε πρόγραμμα μοιάζει ως σκάλα, με δύο κάθετες ράγες και μια σειρά από οριζόντιες βαθμίδες μεταξύ τους.

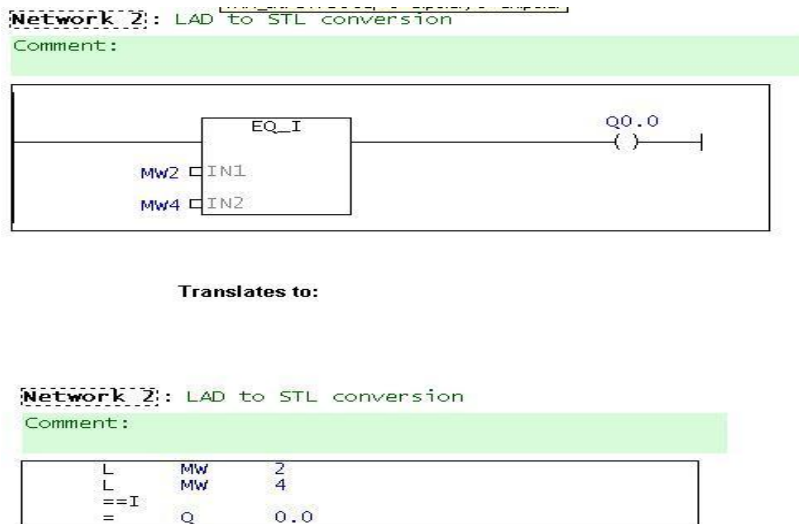


Σχήμα 1.1: Περιβάλλον Ladder

Η δεύτερη γλώσσα προγραμματισμού είναι η **Statement List (STL)**. Η γλώσσα αυτή δημιουργήθηκε περίπου την ίδια στιγμή με την LADDER. Εδώ δημιουργείται μια λίστα προγράμματος με τις εντολές που αντιστοιχούν στις λογικές πύλες οπότε υπάρχουν περιορισμοί στις Boolean εντολές. Αργότερα, με την συνεχόμενη ανάπτυξή τους είδαμε πολλά χαρακτηριστικά που υπήρχαν και σε άλλες γλώσσες και πιο συγκεκριμένα στην Assembly. Εδώ απαιτούνται γνώσεις στοιχειώδους προγραμματισμού.

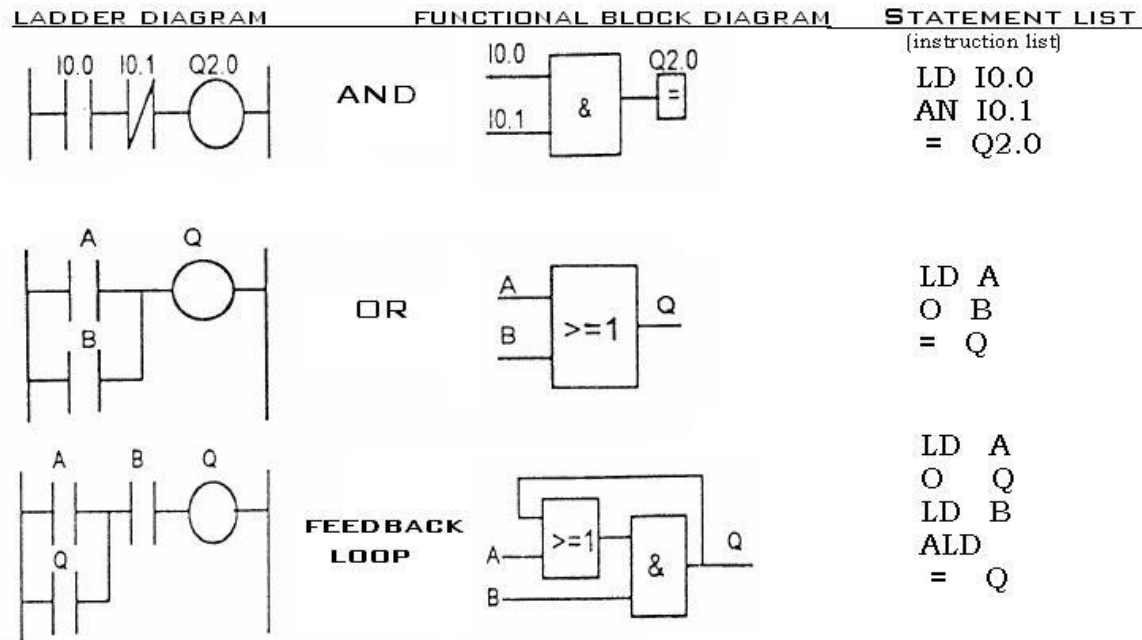
Γενικότερα, η Statement List είναι μια προγραμματιστική γλώσσα η οποία χρησιμοποιεί μνημονική συντομογραφία των λογικών πράξεων της Boolean. Οι λογικοί τελεστές εργάζονται σε έναν συνδιασμό μεταβλητών από Αληθείς και Ψευδείς.

- Οδηγία φόρτωσης - Load (L)
- Οδηγία Και - And (A)
- Οδηγία Ή - Or (O)
- Έξοδος - Output (=)



**Σχήμα 1.2:** Μετατροπή προγράμματος LAD σε STL

Τρίτη και τελευταία είναι η γλώσσα λογικών γραφικών (**Function Block Diagram**). Χρησιμοποιεί γραφικά αλλά αντί για το ηλεκτρολογικό χρησιμοποιεί το αντίστοιχο λογικό σχέδιο της άλγεβρας Boolean. Σύνθετες λειτουργίες όπως οι μαθηματικές μπορούν να εμφανίζονται απ'ευθείας σε συνδιασμό με τα λογικά κουτιά. Ένα απλό παράδειγμα για τις λογικές πύλες σε απεικόνιση Function Block Diagram φαίνεται στην επόμενη εικόνα:



**Σχήμα 1.3:** Λογικές πύλες ως Fuction Block Diagrams

Και οι τρεις γλώσσες αποτελούν γραφικές αναπαραστάσεις ή αναπαραστάσεις κειμένου του λογισμικού STEP 7.

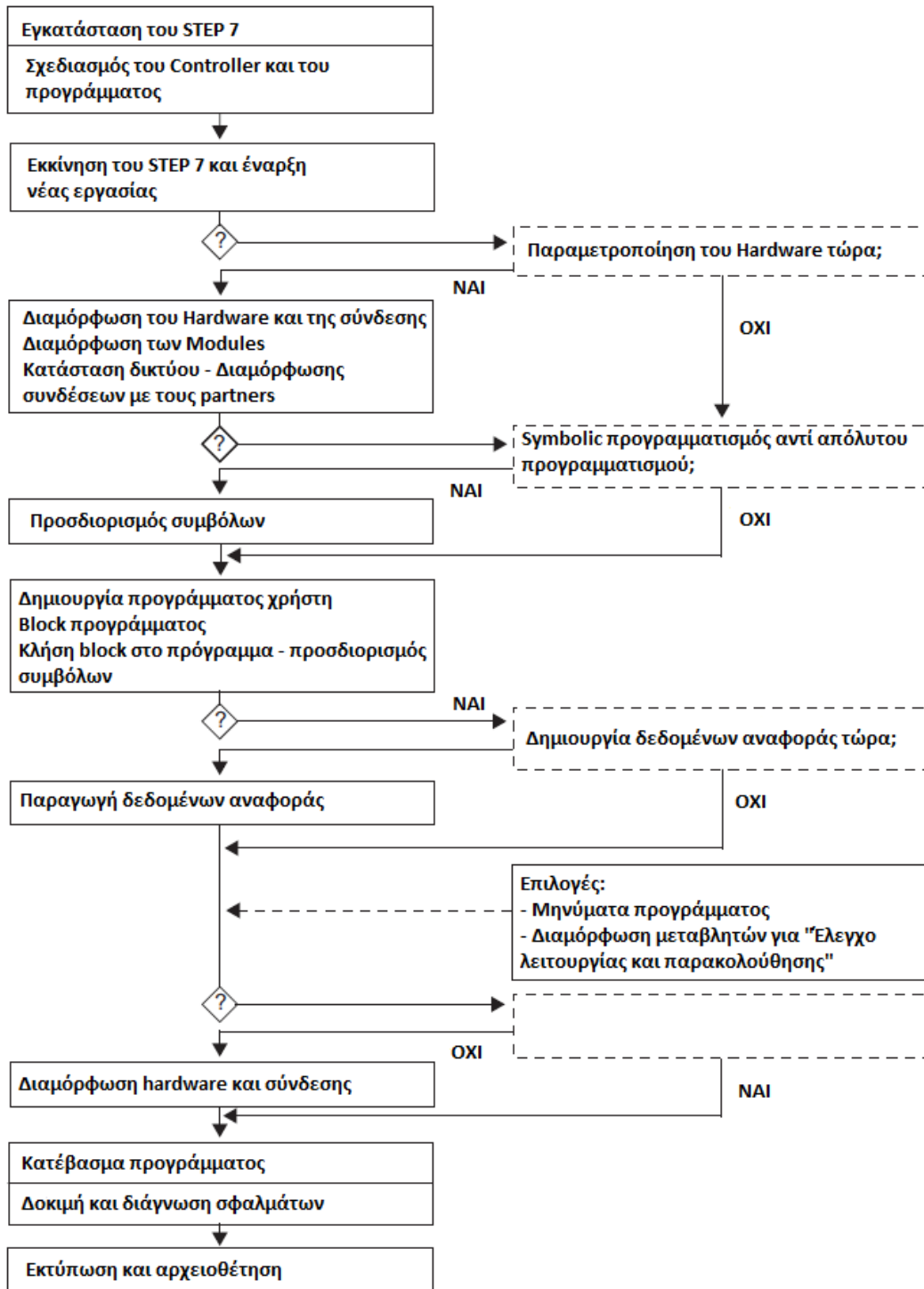
## ΚΕΦΑΛΑΙΟ 2:

### ΤΟ ΛΟΓΙΣΜΙΚΟ STEP 7

Ένα από τα πλέον διαδεδομένα λογισμικά δημιουργίας προγραμμάτων για την ρύθμιση και λειτουργία ενός λογικού ελεγκτή είναι, εκτός από το LogixPRO, το Simatic STEP 7 από την Γερμανική εταιρεία ηλεκτρονικών, Siemens.

Το Simatic STEP 7 είναι ένα πακέτο λογισμικών που χρησιμοποιείται για τον προγραμματισμό των SIMATIC PLCs. Όταν δημιουργούμε μια λύση αυτοματισμού με το λογισμικό αυτό, υπάρχουν ορισμένα βήματα που πρέπει να ακολουθήσουμε:

- 1) Αφού εγκαταστήσουμε το πρόγραμμα στον υπολογιστή μας, σχεδιάζουμε τον ελεγκτή και προετοιμάζουμε την αρχική δομή του κύριου προγράμματός μας.
- 2) Με την δημιουργία ενός νέου έργου, καλούμαστε να κάνουμε δύο επιλογές: Αν υπάρχει κάποιο κύκλωμα συνδεδεμένο, τότε γίνεται η επικύρωσή του από το λογισμικό ενώ αν δεν υπάρχει προχωράμε κατευθείαν στην συγγραφή του προγράμματός μας.
- 3) Διαλέγουμε ποια θα είναι η γλώσσα προγραμματισμού μας, τα σύμβολά της και τα προγραμματιστικά σύμβολα.
- 4) Επιλέγουμε δεδομένα αναφοράς για να μπορέσουμε να κάνουμε γρηγορότερο και καλύτερο έλεγχο (Debugging).
- 5) Με την εύρεση αυτών των δεδομένων, περνάμε τα αντίστοιχα μηνύματα για τον έλεγχο της διαδικασίας του λογικού ελεγκτή.
- 6) Αφού συνδέσουμε το υλικό στον υπολογιστή και δούμε ότι λειτουργεί πλήρως, τότε κατεβάζουμε το πρόγραμμα, το τρέχουμε και αναζητούμε τυχόν σφάλματα.
- 7) Ολοκληρώνοντας εκτυπώνουμε το πρόγραμμα και το αρχειοθετούμε κατάλληλα.



Εικόνα 2.1: Διάγραμμα Προγραμματισμού με STEP 7

## 2.1. Εγκατάσταση του STEP 7

Το Step 7 είναι ένα λογισμικό σαν όλα τα άλλα που χρησιμοποιούμε στον υπολογιστή μας. Η εγκατάστασή του γίνεται μέσω του αντίστοιχου εκτελέσιμου που βρίσκεται είτε στον δίσκο CD ή DVD-ROM που ενδέχεται να αγοράσουμε ή μέσα στο φάκελο που κατεβάσαμε από την αντίστοιχη ιστοσελίδα.

Τα βήματα που θα αναλυθούν παρακάτω, εξηγούν την διαδικασία εγκατάστασης του πακέτου SIEMENS SIMATIC STEP 7 έκδοση 5.5 SP1 και του PLCSim έκδοση 5.4 SP5 για συστήματα 32-bit και 64-bit.

Πριν ξεκινήσουμε, καλό θα ήταν να σιγουρέψουμε ότι το σύστημά μας είναι ικανό να υποστηρίξει το λογισμικό μέσω των απαιτήσεων (System Requirements). Οι απαιτήσεις αυτές είναι:

**Λειτουργικό σύστημα:** Microsoft Windows 2000, Windows XP, Windows Server 2003, Windows Vista ή/και Windows 7 x32/x64.

**Βασικό κύκλωμα:** Προσωπικός Υπολογιστής (PC) ή Προγραμματιζόμενη συσκευή

- **Προσωπικός Υπολογιστής:**
  - Επεξεργαστής Pentium (600MHz)
  - Μνήμη RAM στα 256MB
  - Έγχρωμη Οθόνη
  - Πληκτρολόγιο και ποντίκι, υποστηριζόμενα από τα Microsoft Windows
  
- **Προγραμματιστική Συσκευή (Programming Device – PG)<sup>1</sup>:**
  - Χωρικότητα σκληρού δίσκου
  - Διεπαφή MPI (προαιρετικό): Απαραίτητη μόνο αν υπάρχει σύνδεση ενός PG/PC με το PLC. Χρήσιμο στην επικοινωνία του υπολογιστή με τον ελεγκτή. Αν υπάρχει MPI τότε χρειαζόμαστε ένα καλώδιο USB για την επικοινωνία ή μιας μονάδας (π.χ. CP 5611).
  - Εξωτερικό PROMMER: Εξάρτημα απαραίτητο μόνο αν θέλουμε να προγραμματίσουμε EPROMs.

---

<sup>1</sup> **Programming Device:** Ειδικός συμπανής σχεδιασμός. Είναι υπολογιστής βιομηχανικής χρήσης, εξοπλισμένος πλήρως με SIMATIC PLCs.

Αφού ελέγξουμε ότι το σύστημα πληρεί τις παραπάνω προϋποθέσεις, τότε προχωράμε κανονικά στην κύρια εγκατάσταση.

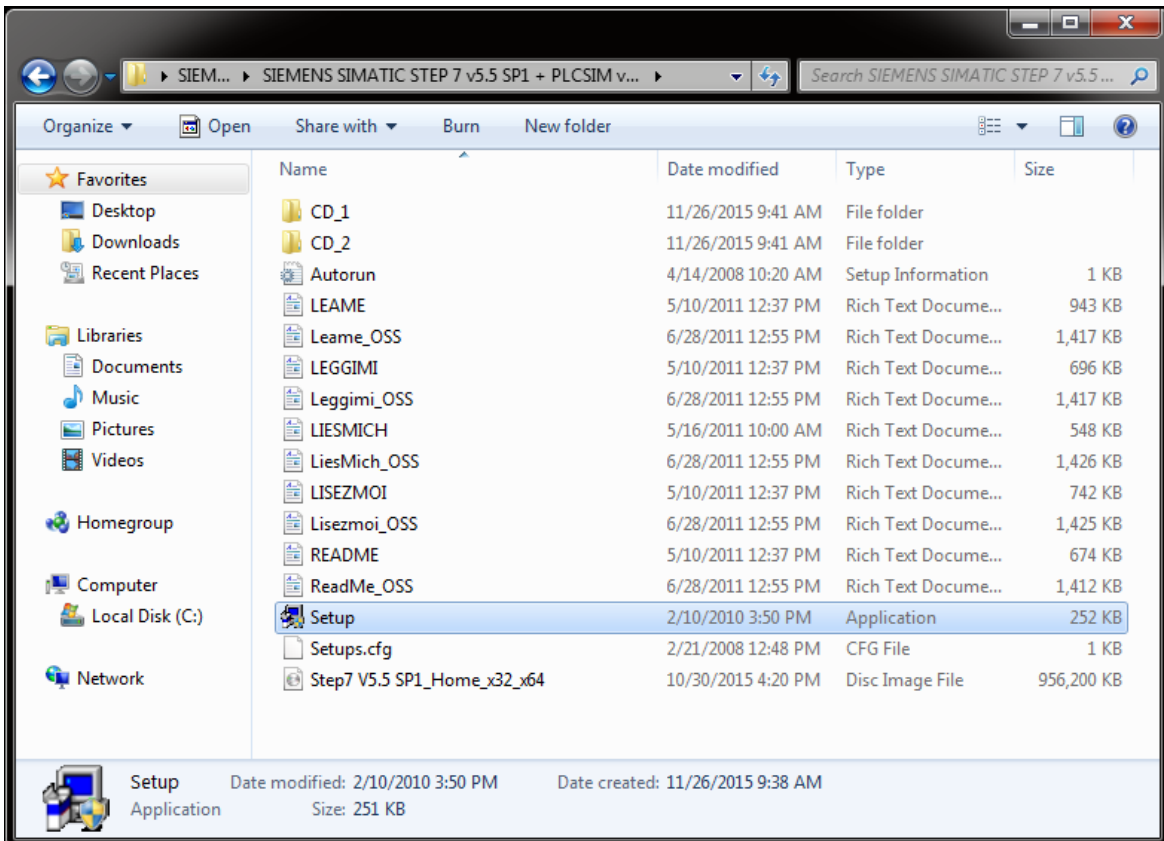
### Βήμα πρώτο:

Αρχικά, πρέπει να σιγουρευτούμε πως το λειτουργικό μας σύστημα υποστηρίζει το Simatic Step 7. Σύμφωνα με την Siemens, η έκδοση 5.5 υποστηρίζεται μέχρι και από τα Windows 7 οπότε αν έχουμε νεότερες εκδόσεις (π.χ. Windows 8, Windows 8.1 ή Windows 10) δεν θα καταφέρουμε να το περάσουμε στον υπολογιστή μας.

Αφού έχουμε συγκεντρώσει και έχουμε εφαρμόσει τα μέσα εγκατάστασης (CD-Rom, DVD-Rom, USB ή ένα απλό εκτελέσιμο αρχείο) τότε αφού τα ενεργοποιήσουμε, προχωράμε στα επόμενα βήματα.

### Βήμα Δεύτερο:

Για την εγκατάσταση του προγράμματος εισάγουμε τον δίσκο, τον “ανοίγουμε” και τρέχουμε το αρχείο με όνομα “Setup.exe”.

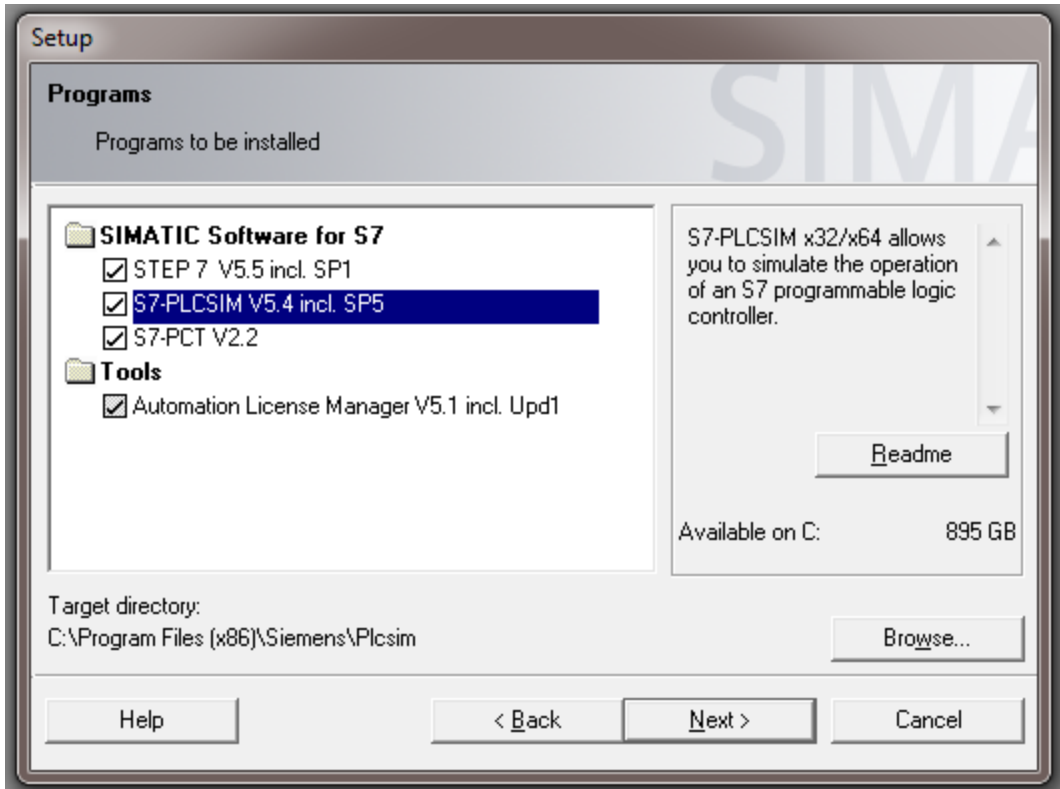


Εικόνα 2.2: Επιλογή αρχείου εγκατάστασης



Ακολουθούμε τα βήματα που μας εμφανίζονται στο αντίστοιχο παράθυρο. Αυτόματα, το Setup μας καθοδηγεί σε όλα τα βήματα της εγκατάστασης μέσω του κουμπιού “Next”.

Το εκτελέσιμο αρχείο μας δίνει τα εξής προγράμματα προς εγκατάσταση:



Εικόνα 2.3: Προγράμματα προς εγκατάσταση

Εδώ ενδέχεται να συναντήσουμε δύο περιπτώσεις:

- **Αν υπάρχει ήδη μια έκδοση του προγράμματος στο σύστημα:**

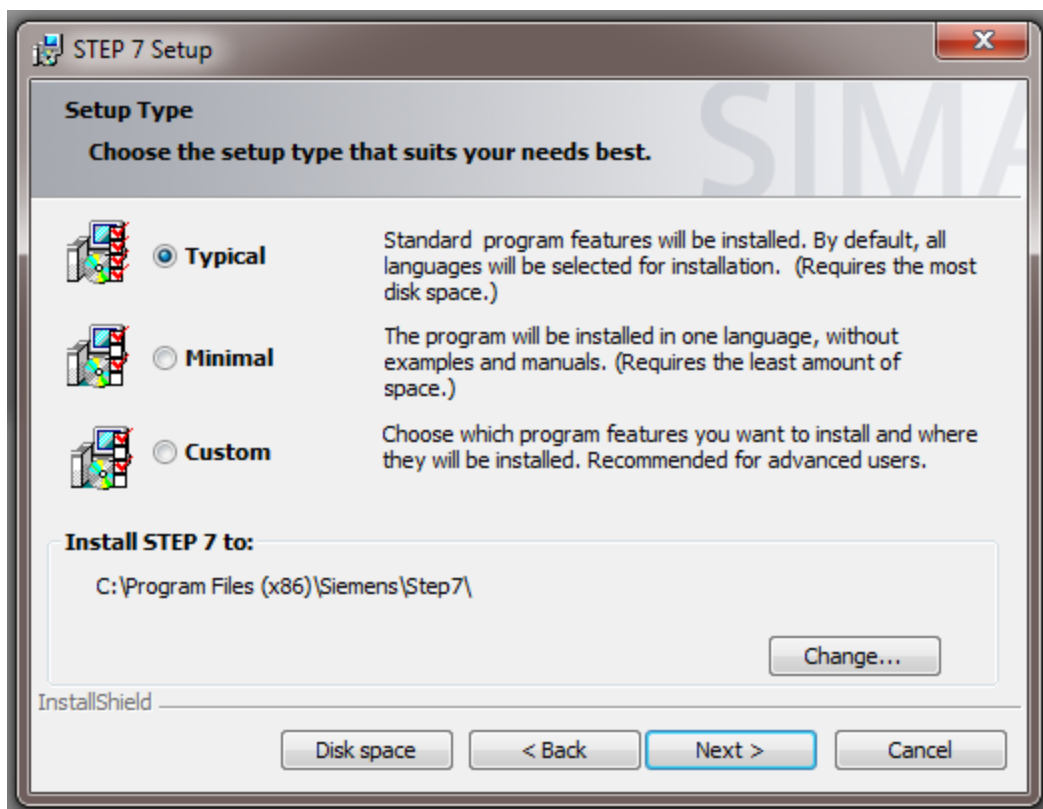
Ένα νέο μήνυμα θα εμφανιστεί στην οθόνη που θα μας ενημερώνει για την προηγούμενη έκδοση του Step 7. Οι επιλογές μας είναι είτε να κρατήσουμε την παλιά έκδοση, ακυρώνοντας την εγκατάσταση ή να συνεχίσουμε και να αντικαταστήσουμε με την νέα έκδοση. Για την καλύτερη απόδοση, συστήνεται να απεγκαταστήσουμε τις προηγούμενες εκδόσεις και να εγκαταστήσουμε “καθαρά”<sup>2</sup> την νέα έκδοση. Ο λόγος είναι πως υπάρχει περίπτωση να παραμένουν αρχεία της παλιάς έκδοσης στην καινούργια, κάτι που μπορεί να προκαλέσει σφάλματα.

<sup>2</sup> **Καθαρά:** Όρος που χρησιμοποιείται κατά την εγκατάσταση ή απεγκατάσταση ενός προγράμματος. Με αυτό δηλώνεται η ολοκληρωτική απεγκατάσταση ενός στοιχείου/προγράμματος από το σύστημά μας.

- **Αν εγκαθιστούμε το πρόγραμμα για πρώτη φορά:**

Η πρώτη εγκατάσταση αποτελεί απλή διαδικασία. Στην αρχή, μας δίνονται τρεις κύριες επιλογές:

- **Προκαθορισμένη εγκατάσταση** (Standard Setup): Η εγκατάσταση προσθέτει μέσα στο σύστημά μας όλες τις γλώσσες διαλόγου για την διεπαφή του χρήστη, όλες τις εφαρμογές και όλα τα διαθέσιμα παραδείγματα.
- **Βασική εγκατάσταση** (Basic Setup): Μόνο μια γλώσσα διαλόγου, χωρίς εφαρμογές και παραδείγματα. Τα αρχεία που εγκαθίστανται είναι πιο μικρά σε μέγεθος μνήμης.
- **Προσαρμοσμένη εγκατάσταση** (Custom Setup): Ο χρήστης μπορεί να επιλέγει ποια αρχεία θα εγκατασταθούν και σε ποια θέση μνήμης (διεύθυνση δίσκου κλπ.)



Εικόνα 2.4: Επιλογές τυπικής, ολοκληρωτικής και προσαρμοσμένης εγκατάστασης

### Βήμα Τρίτο:

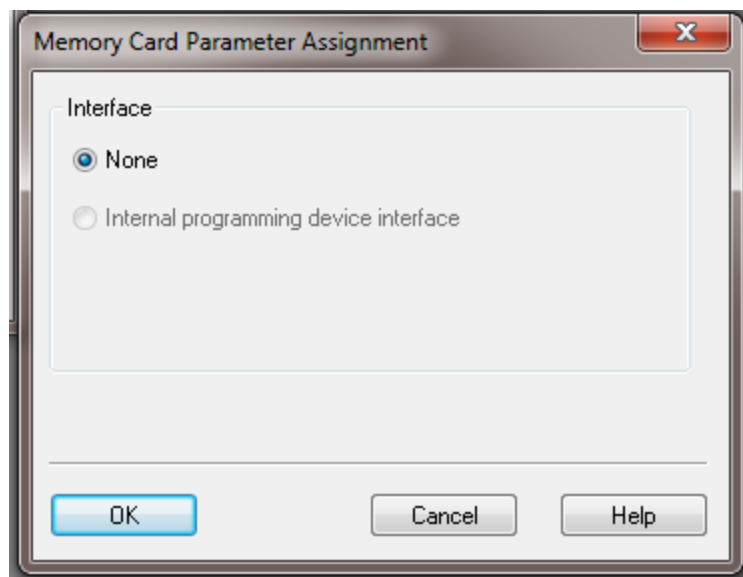
Αφού ολοκληρωθεί η εγκατάσταση, τότε ένα νέο παραθυράκι θα μας ζητήσει το κλειδί άδειας χρήσης. Το κλειδί δίνεται μέχρι και σήμερα σε δεύτερο δίσκο που

κυκλοφορεί μαζί με το βασικό πρόγραμμα ή στον αντίστοιχο οδηγό. Αν λάβαμε το πρόγραμμά μέσω φακέλου κατεβασμένου από το διαδίκτυο ή μεταφέραμε από κάποιον εξωτερικό αποθηκευτικό χώρο, τότε το κλειδί (ή το αρχείο που περιέχει το κλειδί) δίνεται ως κομμάτι του πακέτου.

Από το τέταρτο βήμα και μετά, ξεκινάει η παραμετροποίηση εξωτερικών εξαρτημάτων και τμημάτων μνήμης. Συνήθως, αυτή η πράξη γίνεται με το τρέξιμο του προγράμματος αλλά σε βιομηχανικές χρήσεις η πλήρωση αυτών των απαιτούμενων θεωρείται απαραίτητη για την εξοικονόμηση χρόνου.

### **Βήμα Τέταρτο:**

Ερώτηση για την παραμετροποίηση των διεπαφών του υπολογιστή, της εσωτερικής κάρτας μνήμης ή του PG.



**Εικόνα 2.5:** Παραμετροποίηση κάρτας μνήμης και PG/υπολογιστή

### **Βήμα Πέμπτο:**

Ερώτηση για την παραμετροποίηση των καρτών μνήμης. Αν δεν χρησιμοποιούμε κάρτες μνήμης τότε δεν είναι απαραίτητη και η ύπαρξη οδηγών για EPROM. Έτσι, επιλέγουμε “No EPROM Driver” διαφορετικά επιλέγουμε την είσοδο του PG μας.

Αν χρησιμοποιούμε προσωπικό υπολογιστή, τότε επιλέγουμε τον οδηγό του εξωτερικού prommer. Διευκρινίζουμε πάντα την θύρα στην οποία συνδεόμαστε.

Μπορούμε να αλλάξουμε όλες τις παραμέτρους μέσω του πίνακα ελέγχου του STEP 7 επιλέγοντας το “Memory Card Parameter Assignment”. Μαζί με τις

παραμέτρους, μπορούμε να εγκαταστήσουμε και ένα “flash file” σύστημα το οποίο επιτρέπει σε ανεξάρτητα αρχεία να περαστούν σε μια EPROM κάρτα μνήμης χωρίς να αλλάξουμε στοιχεία της κάρτας.

Η διαδρομή που ακολουθούμε για τον πίνακα ελέγχου και για το “Memory Card parameter assignment” είναι:

- Εκκίνηση → Πίνακας Ελέγχου → Προβολή σε λίστα → Memory Card parameter assignment

### **Αν υπάρξει πρόβλημα κατά την εγκατάσταση του STEP 7:**

Η εγκατάσταση ίσως διακοπεί για τους εξής λόγους:

- Σφάλμα κατά την εκκίνηση που ίσως να οφείλεται στο γεγονός ότι η εγκατάσταση δεν γίνεται σε λειτουργικό Windows.
- Περιορισμένη μνήμη στον σκληρό δίσκο: Κάθε πρόγραμμα που εγκαθιστούμε στον υπολογιστή μας απαιτεί τουλάχιστον 100MB ελεύθερο χώρο στον σκληρό δίσκο.
- Χαλασμένο CD-ROM: Αν το CD είναι ελαττωματικό τότε καλό θα ήταν να απευθυνθούμε στον κατασκευαστή.
- Σφάλμα χρήστη: Παρέμβαση του χρήστη κατά την εγκατάσταση ή σφάλμα συστήματος. Αναγκαία η επανεκκίνηση της εγκατάστασης

### **Μετά την ολοκλήρωση:**

Το αντίστοιχο παράθυρο που αναδύεται μας ενημερώνει για την επιτυχή εγκατάσταση του προγράμματος. Αν έχουν γίνει αλλαγές στα αρχεία κατά την εγκατάσταση τότε πρέπει να επανεκκινήσουμε τον υπολογιστή. Μετά μπορούμε να κάνουμε εκκίνηση του STEP 7 – SIMATIC Manager.

## **2.2. Εγκατάσταση άδειας χρήσης**

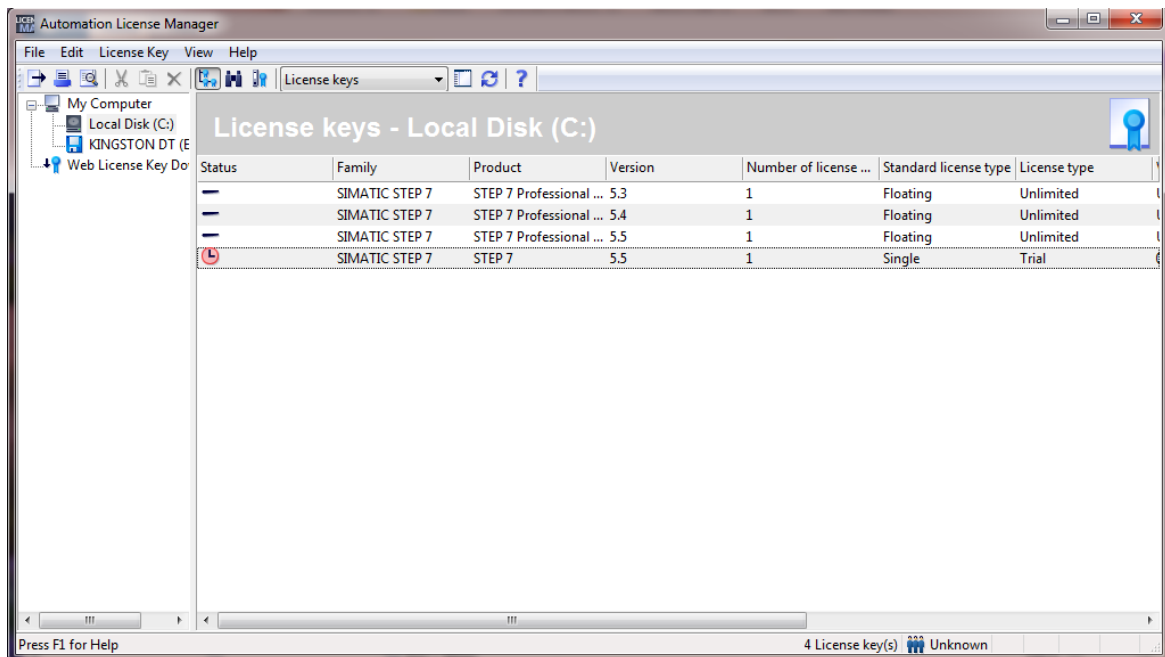
Ίσως σε πολλούς να φαίνεται κάπι απλό αλλά η εγκατάσταση και η επικύρωση ενός κωδικού άδειας χρήσης δεν είναι εύκολη υπόθεση. Όπως είπαμε παραπάνω, μαζί με το πακέτο λογισμικών, εμπεριέχεται και ένας παραπάνω δίσκος που περιέχει το κλειδί που θα χρειαστούμε για την ενεργοποίηση.

Το κυριότερο μέσο που χρησιμοποιείται για την διανομή του κλειδιού, είναι η χαρακτηριστική κίτρινη δισκέτα. Όλα τα μεγάλα προϊόντα και softwares των

εταιρειών όπως η Siemens, διαρκούν μόνο 15 μέρες στα πλαίσια της απλής δοκιμής (Trial). Ο μοναδικός αριθμός που υπάρχει στον δίσκο είναι απαραίτητος για να ζητήσουμε από την κεντρική ιστοσελίδα υποστήριξης, τον αντίστοιχο μοναδικό σειριακό αριθμό. Μιας και όμως, οι δισκέτες αποτελούν πλέον ένα κομμάτι της ιστορίας των υπολογιστών, η διανομή αυτή γίνεται πλέον με την μορφή δίσκου.

Για να εισάγουμε ένα κλειδί επιτυχώς στο πρόγραμμα προτίμησής μας, πρέπει αφού το λάβουμε και το επικυρώσουμε, να το εφαρμόσουμε στο λογισμικό μας μέσω του Automation License Manager. Το ALM είναι ένα από τα προγράμματα που περιέχονται στο πακέτο εγκατάστασης και είναι υπεύθυνο για την σωστή εγκατάσταση και επικύρωση των σειριακών κωδικών για όλα τα προγράμματα αυτοματισμού και ηλεκτρονικών της Siemens.

Τελειώνοντας, ελέγχουμε την κατάσταση των κλειδιών.



Εικόνα 2.6: Παράθυρο του Automation License Manager

### 2.3. Συνδέοντας την διεπαφή του PG/PC

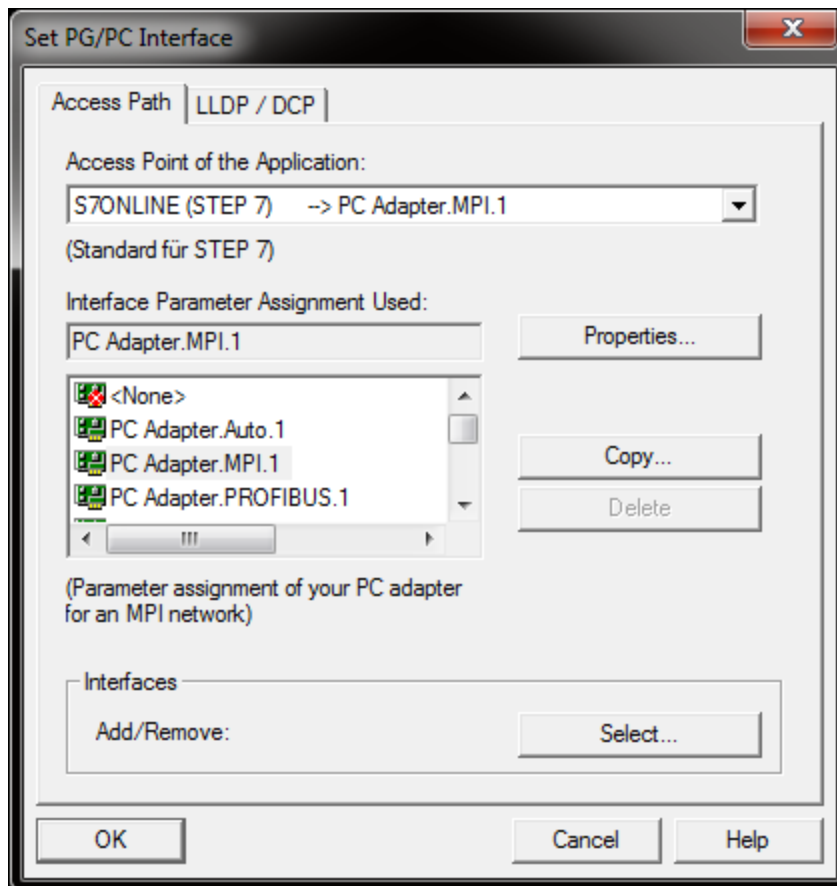
Σε αυτό το σημείο καλούμαστε να ρυθμίσουμε την επικοινωνία μεταξύ του PG/PC και του PLC. Ως προϋπόθεση θέτεται η ύπαρξη ενός PLC συνδεδεμένου κατά την διάρκεια της παραμετροποίησης. Αν δεν υπάρχει κάποιος ελεγκτής συνδεδεμένος, τότε δεν γίνεται κάποια επικοινωνία και θα πρέπει να ρυθμίσουμε ως μέσω εικονικής επικοινωνίας κάποιο εσωτερικό εξάρτημα του υπολογιστή.

### 2.3.1. Βασική διαδικασία

Πριν ξεκινήσουμε την παραμετροποίηση, πρέπει να ελέγξουμε ότι το σύστημά μας έχει τις κατάλληλες ρυθμίσεις ενεργές και πως οι ρυθμίσεις της διεπαφής είναι αξιόπιστες.

Αν χρησιμοποιούμε μια κάρτα επικοινωνίας MPI ή κάποιον επεξεργαστή επικοινωνιών, θα πρέπει να ελέγξουμε τις διακοπές και τη διευθυνσιοδότηση των Windows και να σιγουρέψουμε ότι δεν υπάρχουν προβλήματα, μέσω του Πίνακα ελέγχου. Για ευκολότερη παραμετροποίηση, τα Windows μας αναδύουν ένα παράθυρο διαλόγου με όλες τις διαθέσιμες επιλογές για τις ρυθμίσεις μας.

Η διαδικασία που ακολουθούμε για την ενεργοποίηση της επικοινωνίας ανάμεσα στα δύο κυκλώματα είναι η παρακάτω:



**Εικόνα 2.7:** Ρυθμίσεις επικοινωνίας μεταξύ PC/PG και PLC

- Πηγαίνοντας στον Πίνακα Ελέγχου των Windows, κάνουμε διπλό κλικ στην επιλογή “Setting PG/PC Interface”
- Θέτουμε το “Access Point of Application” σε “S7ONLINE”.

- Στην λίστα “Interface Parameter Set Used”, επιλέγουμε το σύνολο των παραμέτρων που επιθυμούμε. Αν δεν υπάρχει αυτό το σύνολο, τότε το εγκαθιστούμε χειροκίνητα μέσω των δομοστοιχείων (modules) από την επιλογή “select”.
  - Αν η επαφή μας μπορεί να αναγνωρίσει τις παραμέτρους αυτόματα, τότε μπορούμε να συνδέσουμε το PG/PC στο MPI ή το PROFIBUS<sup>3</sup> χωρίς να παρέμβουμε στην ρύθμιση. Τα προαπαιτούμενα για την αυτόματη αναγνώριση είναι η σύνδεση για την εκπομπή των σημάτων διαύλου.
  - Αν από την άλλη, δεν έχουμε αυτόματη αναγνώριση των παραμέτρων, τότε μπορούμε να εμφανίσουμε τις ιδιότητες και να τις θέσουμε έτσι ώστε να υπάρχει επιτυχημένη σύνδεση στο δίκτυο.

Αν επιθυμούμε να μην ρυθμίσουμε την επικοινωνία, τότε μας δίνεται η δυνατότητα παράλειψης. Πρέπει να θυμόμαστε όμως, πως για να μπορέσουμε να κατεβάσουμε και να δοκιμάσουμε ή να προσωμοιώσουμε ένα έτοιμο πρόγραμμα, θα πρέπει να έχουμε ολοκληρώσει την ρύθμιση. Ως λύση σε αυτή την προϋπόθεση είναι η άμεση ρύθμιση μέσω του SIMATIC Manager.

#### **2.4. Απεγκατάσταση του STEP 7**

Η διαδικασία που ακολουθούμε για να απεγκαταστήσουμε το STEP 7 από τον υπολογιστή μας είναι απλή αλλά χρονοβόρα. Αρχικά, ανοίγουμε το παράθυρο του Πίνακα Ελέγχου και επιλέγουμε το “Προγράμματα και εφαρμογές (Programs and Features)”. Έπειτα, αναζητούμε το πλήθος των προγραμμάτων που εγκαταστάθηκαν.

Επιλέγουμε με δεξί κλικ την εφαρμογή που επιθυμούμε και τότε το αντίστοιχο παράθυρο διαλόγου αναδύεται. Αφού γίνει η απεγκατάσταση, θα πρέπει να κάνουμε επανεκκίνηση του συστήματός μας για να συνεχίσουμε με τις υπόλοιπες εφαρμογές. Αν επιχειρήσουμε να συνεχίσουμε τις απεγκαταστάσεις, τότε το πρόγραμμα θα μας δίνει σφάλμα ύπαρξης εκκρεμούς επανεκκίνησης συστήματος από κάποια προηγούμενη εφαρμογή.

---

<sup>3</sup> **PROFIBUS**: Πρότυπο για την επικοινωνία διαύλων στον αυτοματισμό. Εμφανίστηκε στην Γερμανία το 1989 από την BMBF και αργότερα χρησιμοποιήθηκε από την Siemens (πηγή: Wikipedia)

Τέλος, θα πρέπει να σιγουρέψουμε ότι έχουν διαγραφεί από το σύστημά μας όλα τα αρχεία των σχετικών προγραμμάτων. Για να το κάνουμε αυτό, θα πρέπει να ψάξουμε στα Αρχεία Προγραμμάτων (Program Files) μέσα στον σκληρό μας δίσκο. Η σουίτα του SIMATIC βρίσκεται συνήθως στις παρακάτω διαδρομές:

**Για συστήματα Windows των 32-bit:**

C:\Program Files\Siemens

**Για συστήματα Windows των 64-bit:**

C:\Program Files (x86)\Siemens

Αν ελέγχοντας τις διαδρομές αυτές, διαπιστώσουμε ότι δεν υπάρχουν φακέλοι ή αρχεία που καθιστούν δυνατή την εκκίνηση ή την παρέμβαση σε κάποια εφαρμογή, τότε έχουμε απεγκαταστήσει επιτυχώς την σουίτα.

Ένας δεύτερος τρόπος απεγκατάστασης είναι η άμεση διαγραφή των φακέλων μέσα από τις διαδρομές του σκληρού δίσκου που δίνονται παραπάνω. Ο τρόπος αυτός είναι ίσως και ο πιο γρήγορος, μπορεί όμως να προκαλέσει προβλήματα στο σύστημά μας. Οι συντομεύσεις στην επιφάνεια εργασίας εξακολουθούν να υπάρχουν ενώ ίχνη του προγράμματος ίσως υπάρχουν σε άλλους τομείς της κύριας μνήμης του υπολογιστή αποτρέποντάς μας έτσι από μελλοντική επανεγκατάσταση. Συστίνεται η απεγκατάσταση μέσω του Πίνακα Ελέγχου των Windows.



## ΚΕΦΑΛΑΙΟ 3:

### ΔΗΜΙΟΥΡΓΙΑ ΝΕΩΝ ΕΡΓΑΣΙΩΝ ΣΤΟ STEP 7

Έχοντας ολοκληρώσει την εγκατάσταση του STEP 7 στο σύστημά μας και έχοντας εισάγει το κλειδί αδείας, τώρα είμαστε έτοιμοι να ξεκινήσουμε την εφαρμογή για τους προγραμματιστικούς μας σκοπούς.

Στην επιφάνεια εργασίας μας έχουν δημιουργηθεί τρία εικονίδια: “SIMATIC Manager”, “S7-PCT Port Configuration” και “Automation License Manager”. Έχουμε αναφέρει ήδη πως το ALM χρησιμεύει στην επαλήθευση των κωδικών ενεργοποίησης. Το S7-PCT Port Configuration μας χρησιμεύει στην σωστή διασύνδεση και επικοινωνία μεταξύ των ελεγκτών μας και του υπολογιστή ή της ηλεκτρονικής συσκευής.

Η εφαρμογή για την οποία ενδιαφερόμαστε, δηλαδή το SIMATIC Manager είναι η βασική εφαρμογή για τον προγραμματισμό, την προσωμοίωση και τον έλεγχο λειτουργίας των ελεγκτών. Παρακάτω θα αναλυθούν το περιβάλλον εργασίας, τα διαθέσιμα κουμπιά καθώς και πως εργαζόμαστε.

#### 3.1. Περιβάλλον Εργασίας του SIMATIC Manager

Για να εκτελέσουμε το SIMATIC Manager, κάνουμε διπλό κλικ πάνω στο εικονίδιο που φέρει την αντίστοιχη ονομασία. Ένα παράθυρο ανοίγει και είναι αυτό που μας δίνει πρόσβαση σε όλες τις λειτουργίες. Εναλλακτικά, η εκκίνηση μπορεί να γίνει μέσω της εκκίνησης των Windows και έπειτα στο σχετικό εικονίδιο.

Οι λειτουργίες στις οποίες έχουμε πρόσβαση είναι η δημιουργία νέων εργασιών, η παραμετροποίηση και ανάθεση των παραμέτρων στο υλικό, η παραμετροποίηση των δικτύων επικοινωνίας, ο προγραμματισμός των μπλοκ, η εκτέλεση και ο έλεγχος των προγραμμάτων. Η εξοικείωση είναι εύκολη και αυτό οφείλεται στον φιλικό του σχεδιασμό. Για ακόμα μεγαλύτερη ευκολία μπορούμε να εργαστούμε σε δυο καταστάσεις:

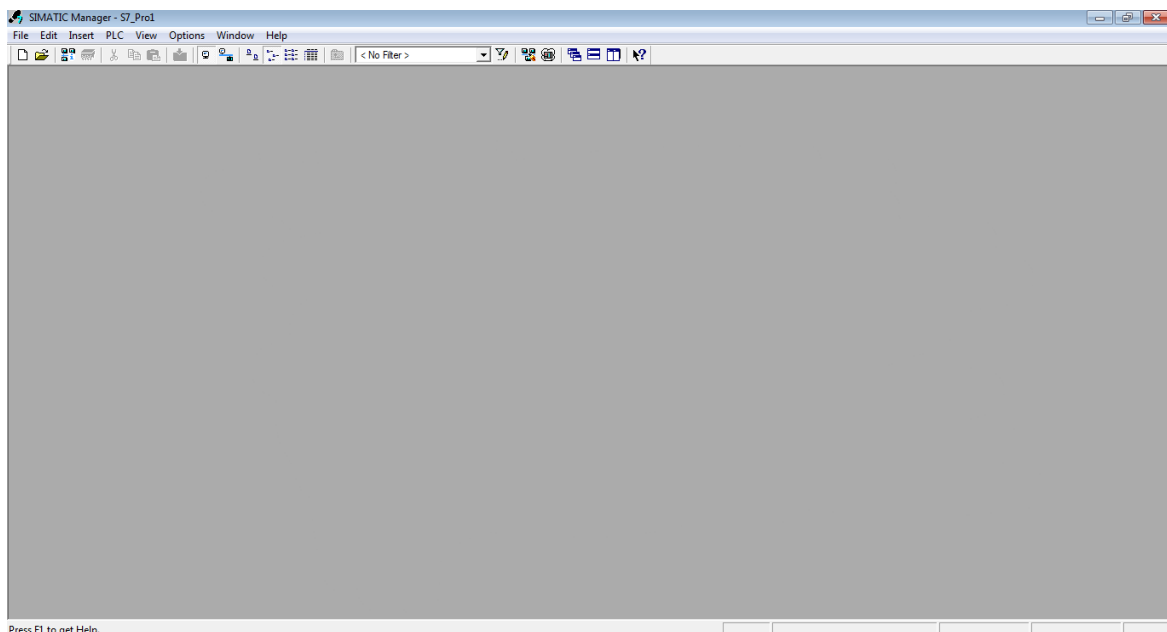
- Εκτός Σύνδεσης: δεν υπάρχει συνδεδεμένος ελεγκτής
- Σε Σύνδεση: Υπάρχει συνδεδεμένος ελεγκτής. Η σύνδεση είναι απαραίτητη για να προσδιορίζονται τα αποτελέσματά της εκτέλεσης καθώς και η λειτουργία του ελεγκτή



**Εικόνα 3.1:** Εικονίδιο εκκίνησης SIMATIC Manager

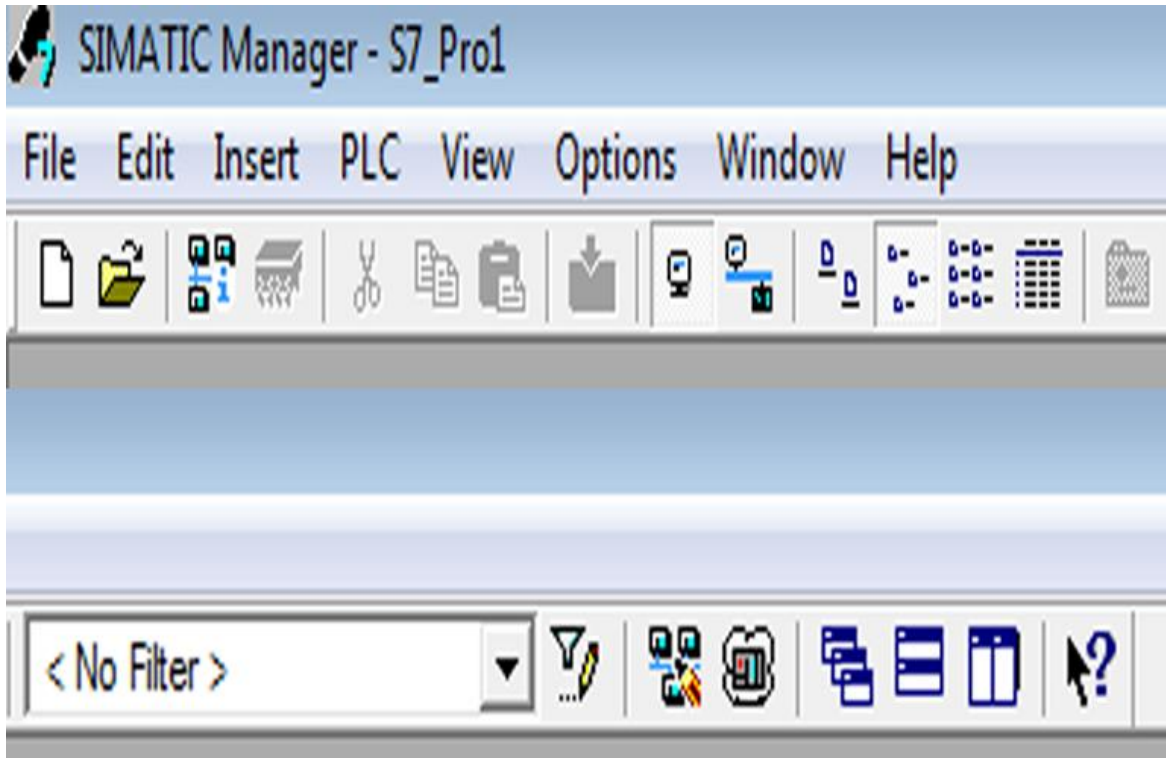
Το SIMATIC Manager έχει ένα απλό περιβάλλον εργασίας με όλα όσα θα χρειαστούμε για τον προγραμματισμό και την προσωμοίωση των ελεγκτών μας. Όπως τα περισσότερα προγράμματα στο εμπόριο, έτσι και το Manager αποτελείται από την βασική γραμμή επιλογών μέσω της οποίας μπορούμε να δημιουργήσουμε μια νέα εργασία, να επεξεργαστούμε τα αρχεία μας ή και να τα αποθηκεύσουμε, και την γραμμή εργαλείων που περιέχει όλες τις συντομεύσεις των βασικών λειτουργιών με την μορφή εικονιδίων. Οι περισσότερες από αυτές τις επιλογές δεν είναι προσβάσιμες μέχρι να δημιουργήσουμε μια νέα εργασία ή 'Project'.

Το περιβάλλον εργασίας του SIMATIC Manager φαίνεται παρακάτω:



**Εικόνα 3.2:** Περιβάλλον εργασίας SIMATIC Manager

Και η γραμμή εργαλείων:



Εικόνα 3.3: Γραμμή εργαλείων SIMATIC Manager (σε μεγέθυνση)

Βλέπουμε πως η γραμμή μενού είναι παρόμοια με εκείνη άλλων προγραμμάτων. Αρχικά, υπάρχουν οι βασικές επιλογές:







Μενού	Ανάλυση Λειτουργιών
<b>File</b> (Αρχείο)	Είναι δυνατή η δημιουργία, το άνοιγμα και το κλείσιμο ενός αρχείου, η δημιουργία ενός νέου project χειροκίνητα ή αυτόματα, το διάβασμα και η ρύθμιση μιας κάρτας μνήμης S7, η αποθήκευση, η διαγραφή, οργάνωση, ο έλεγχος, η αρχειοθέτηση και η εκτύπωση μιας εργασίας.
<b>Edit</b> (Επεξεργασία)	Γίνεται η ανιγραφή, αποκοπή και επικόλληση ενός στοιχείου της εργασίας. Επιπλέον, μπορεί να γίνει η επιλογή, η διαγραφή, η μετονομασία και παραμετροποίηση ενός αντικειμένου καθώς και η εξαγωγή του.












<p><b>Insert</b> (Εισαγωγή)</p>	<p>Αποτελεί την κύρια επιλογή για προσθήκες αντικειμένων στην εργασία μας. Μέσω αυτής μπορούμε να εισάγουμε σταθμούς (Stations), Υποδίκτυα (Subnets), έτοιμα Προγράμματα (Program) καθώς και στοιχεία του S7 και M7 (Software, Blocks). Δίνεται και πρόσβαση στον πίνακα σημάτων, κειμένων και στην βιβλιοθήκη.</p>
<p><b>PLC</b></p>	<p>Όσες επιλογές χρειαζόμαστε για να γίνει επιτυχής επικοινωνία με το PLC. Δίνονται δικαιώματα επεξεργασίας και εισόδου στον χρήστη, σύνδεση με PC/PG, πρόσβαση στην κάρτα μνήμης, αναβάθμιση του σχετικού Firmware αλλά και πρόσβαση στα στοιχεία της CPU όπως σε μηνύματα και μεταβλητές.</p>
<p><b>View</b> (Προβολή)</p>	<p>Εδώ μπορεί να γίνει η αλλαγή από την κατάσταση “Αποσυνδεδεμένος” σε κατάσταση “Συνδεδεμένος”. Επίσης, καθορίζονται τα μεγέθη των εικονιδίων και των αντικειμένων και η επιλογή προβολής τους. Επιλέγουμε επίσης την εμφάνιση της μπάρας εργαλείων και κατάστασης.</p>
<p><b>Options</b> (Επιλογές)</p>	<p>Επιλογές για εισαγωγή και εξαγωγή κώδικα και αρχείων, προστασία δικαιωμάτων, πρόσβαση στα logs και τα Libraries και σύγκριση των δεδομένων. Τέλος, γίνεται και ο Ορισμός του Interface των PG/PC.</p>
<p><b>Window</b> (Παράθυρο)</p>	<p>Τα παράθυρα που θα είναι ορατά και αναδυόμενα στην εργασία μας (π.χ. Project Window, Arrange Icons κ.λ.π.)</p>
<p><b>Help</b> (Βοήθεια)</p>	<p>Υπάρχουν όλα τα διαθέσιμα βοηθήματα, οδηγοί και διαδικτυακές συνδέσεις καθώς και πληροφορίες για το SIMATIC Manager και την έκδοσή του. Εδώ ο χρήστης μπορεί να ανατρέξει στα βασικά βήματα που θα πρέπει να ακολουθήσει για την δημιουργία και επεξεργασία ενός νέου Project.</p>

**Πίνακας 3.1:** Αναφορά λειτουργιών γραμμής κατάστασης του SIMATIC Manager

Κάθε μια από τις παραπάνω επιλογές μας δίνει ένα νέο μενού με διάφορες επιλογές που αντιστοιχούν στο μενού αυτό. Για παράδειγμα, το “Αρχείο” επιτρέπει την δημιουργία ενός νέου αρχείου ή εργασίας μέσω του “Νέα Εργασία” (New Project) ή μέσω της συντόμευσης Ctrl+N από το πληκτρολόγιο. Οι περισσότερες επιλογές είναι συνδεδεμένες με συντομεύσεις πληκτρολογίου για την εξοικονόμηση χρόνου.

Εκτός από την γραμμή μενού, έχουμε και την γραμμή εργαλείων με όλα τα βασικά εικονίδια που μας βοηθούν στην άμεση πρόσβαση:

Κουμπιά	Αναφορά Λειτουργιών
	<b>New – Νέο αρχείο:</b> Ανάλογα την επιλογή μας, το κουμπί αυτό δημιουργεί ένα νέο project, library ή multiproject
	<b>Open – Άνοιγμα Αρχείου:</b> Άνοιγμα ενός υπάρχοντος project, library ή multiproject
	<b>Accessible Nodes – Προσβάσιμοι κόμβοι:</b> Άνοιγμα παραθύρου στο οποίο εμφανίζονται όλες οι προγραμματιζόμενες μονάδες με τις διευθύνσεις τους στο δίκτυο.
	<b>S7 Memory Card – Κάρτα μνήμης S7:</b> Εμφάνιση των περιεχομένων της κάρτας μνήμης. Η κάρτα θα πρέπει να είναι συνδεδεμένη στην συσκευή προγραμματισμού (PG/PC).
	<b>Cut, Copy, Paste – Αποκοπή, Αντιγραφή, Επικόλληση:</b> Τα κουμπιά αυτά βοηθούν στην αποκοπή, αντιγραφή και επικόλληση αντικειμένων και στοιχείων μέσα σε μια εργασία.
	<b>Download – Κατέβασμα:</b> Με αυτό το κουμπί γίνεται κατέβασμα του προγράμματος στο PLC για επιτυχή εκτέλεση. Πρέπει να γίνεται κάθε φορά που έχουμε ολοκληρώσει το πρόγραμμά μας και θέλουμε να το δοκιμάσουμε είτε στην προσωμοίωση είτε στο πραγματικό μας κύκλωμα. Για την δεύτερη περίπτωση θα πρέπει να έχουμε συνδέση τον ελεγκτή/συσκευή μας με τον υπολογιστή.

	<b>Online – Σε σύνδεση:</b> Εναλλαγή σε κατάσταση συνδεδεμένου. Χρησιμοποιείται για πρόσβαση σε ελεγκτές προς προγραμματισμό.
	<b>Offline – Αποσυνδεδεμένο:</b> Το αντίθετο του Online. Γυρίζει σε αποσυνδεδεμένη κατάσταση.
	<b>Large Icons – Μεγάλα Εικονίδια:</b> Μεγαλώνει όλα τα αντικείμενα και εικονίδια της εργασίας.
	<b>Small Icons – Μικρά εικονίδια:</b> Μικραίνει όλα τα αντικείμενα της εργασίας.
	<b>List – Λίστα:</b> Εμφάνιση όλων των αντικειμένων στην δεξιά μεριά του παραθύρου στην μορφή λίστας.
	<b>Details – Λεπτομέρειες:</b> Εμφάνιση των αντικειμένων σε λίστα μαζί με λεπτομέρειες για το καθένα (όπως ημερομηνία αλλαγής κ.λ.π.)
	<b>Filter Command – Φιλτράρισμα:</b> Με το κουμπί αυτό αναδύεται ένα παράθυρο μέσα στο οποίο μπορούμε να επιλέξουμε, να δημιουργήσουμε ή να επεξεργαστούμε τα φίλτρα των εμφανιζόμενων αντικειμένων.
	<b>Up One Level – Αλλαγή επιπέδου προς τα πάνω:</b> Με το πάτημα του κουμπιού ανεβαίνουμε ένα επίπεδο πάνω, προς τον αρχικό φάκελο.
	<b>Simulate Module – Προσομοίωση:</b> Μέσω αυτού του κουμπιού μπορούμε να προσομοιώσουμε την λειτουργία ενός PLC χωρίς την ύπαρξη υλικού, αρκεί να έχουμε σταθερή σύνδεση και επικοινωνία στον υπολογιστή μας.
	<b>Διάταξη (Οριζόντια, κάθετα, συνεχόμενα) – Arrange (Horizontally, Vertically, Cascade):</b> Γίνεται αναδιάταξη όλων των ανοιχτών παραθύρων μέσα στο STEP 7. Η αναδιάταξη μπορεί να γίνει κάθετα, οριζόντια ή σε αλληλουχία.
	<b>Help – Βοήθεια:</b> Μετάβαση στο παράθυρο βοήθειας. Εκεί εμπεριέχονται βοηθήματα και βήματα για το πως δουλεύει η εφαρμογή και όλα τα βασικά.

**Πίνακας 3.2:** Αναφορά λειτουργιών των κουμπιών στο SIMATIC Manager

Αν θέλουμε να μάθουμε περισσότερα για το κάθε πλήκτρο ή επιλογή ή ακόμα και για τις πιο εξειδικευμένες λειτουργίες του Manager, τότε μπορούμε να επιλέξουμε το μενού “Βοήθεια” από την γραμμή των μενού (όπως φαίνεται στον πίνακα 4.1) ή να επιλέξουμε το αντίστοιχο κουμπί βοήθειας από την γραμμή εργαλείων (όπως φαίνεται από τον πίνακα 4.2).

Σημειώνεται πως πριν την έναρξη μιας νέας εργασίας, η πλειοψηφία των επιλογών στα μενού και στα κουμπιά είναι μη διαθέσιμες. Αυτό συμβαίνει για τον λόγο ότι αφού δεν υπάρχουν εργασίες σε εφαρμογή, οι επιλογές αυτές παραμένουν ανενεργές για την καλύτερη αξιοποίηση των πόρων του συστήματος και του προγράμματος.

Αφού έχουμε διαβάσει και εξοικειωθεί με τις λειτουργίες των βασικών πλήκτρων, ήρθε η ώρα να φτιάξουμε το πρώτο Project μας.

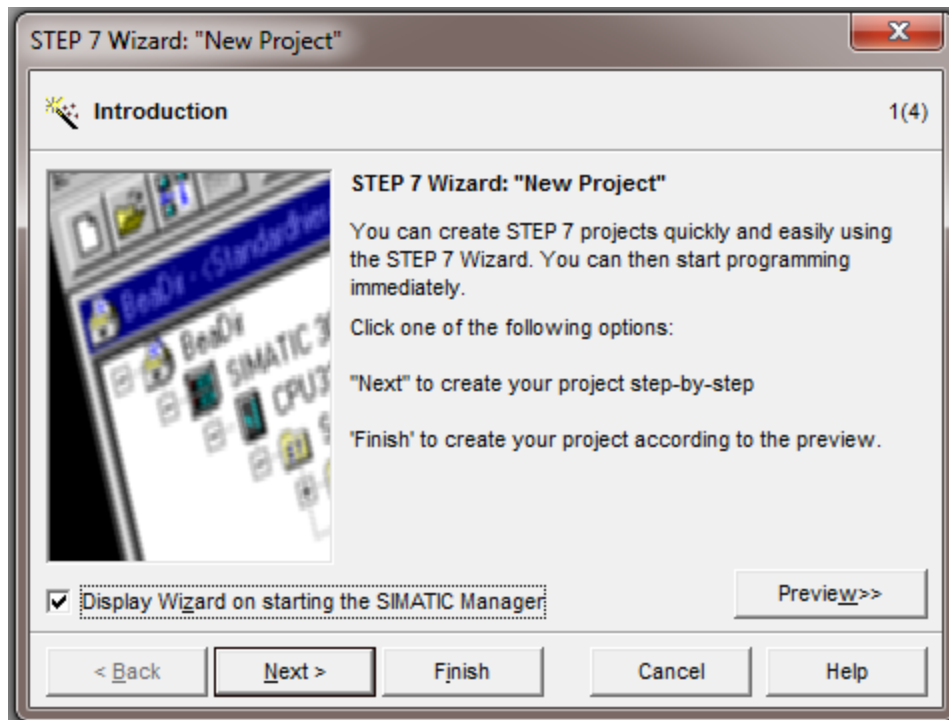
### **3.2. Δημιουργία νέας εργασίας**

Το SIMATIC Manager αποτελεί ένα λογισμικό ιδανικό για αρχάριους αλλά και για επαγγελματίες. Μπορούμε να το χρησιμοποιήσουμε είτε για προσωπική εκμάθηση πάνω στο αντικείμενο του προγραμματισμού λογικών ελεγκτών, είτε για να προγραμματίσουμε άμεσα έναν λογικό ελεγκτή που ενδέχεται να έχουμε αγοράσει και θέλουμε να χρησιμοποιήσουμε. Γι'αυτό το λόγο, η εταιρεία ανάπτυξης έχει προβλέψει και για τις δύο περιπτώσεις οπότε η δημιουργία ενός νέου project μπορεί να γίνει με δύο τρόπους. Αυτόματα και χειροκίνητα.

Υπάρχουν πολλά είδη προσωμοιώσεων τα οποία μπορούμε να δημιουργήσουμε και αυτά ξεκινούν από απλά χρονικά και μετρητές μέχρι και πιο περίπλοκα όπως η λειτουργία των δεξαμενών ή κυλιόμενων σκαλών. Ξεκινώντας την εφαρμογή μας θα έρθουμε σε πρώτη επαφή με τον “οδηγό δημιουργίας” (STEP 7 Wizard: New Project). Αυτή η λειτουργία μας επιτρέπει την αυτόματη δημιουργία μιας νέα εργασίας με τις επιλογές που επιθυμούμε όπως το είδος του επεξεργαστή, την μονάδα σταθμού, τα αντικείμενα που θα εμπεριέχονται (μαζί με την συμβολική τους ονομασία για μεγαλύτερη ευκολία στον εντοπισμό) και την γλώσσα προγραμματισμού (LAD, STL, FBD) στην οποία θα βασίσουμε το πρόγραμμα.

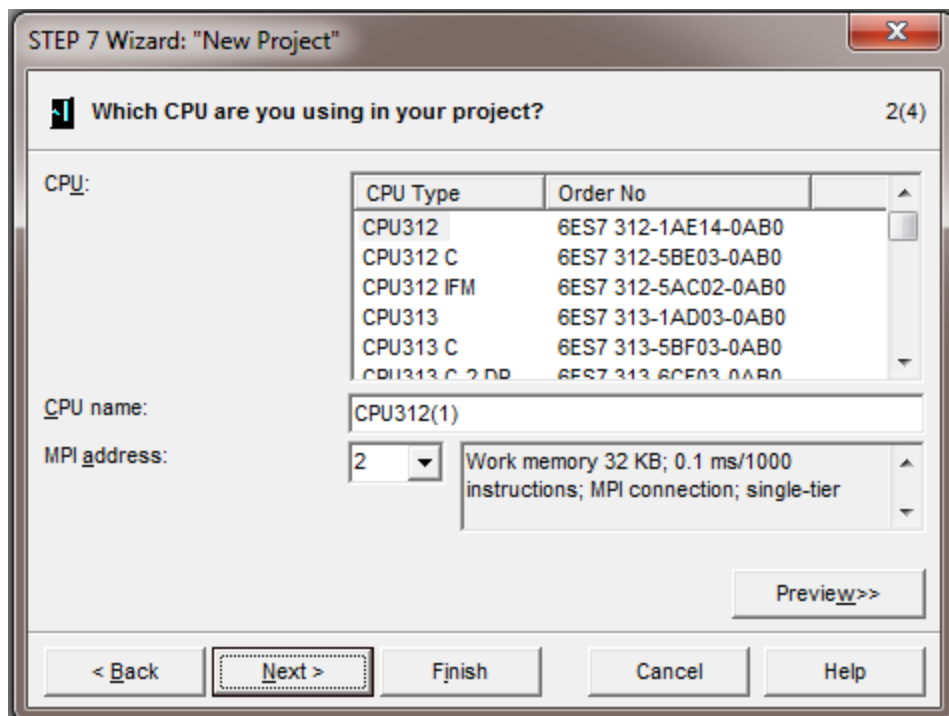
Τα βήματα που προκύπτουν από την αυτόματη δημιουργία είναι τα εξής:

### Βήμα 1<sup>ο</sup>: Έναρξη του οδηγού.



Εικόνα 3.4: Παράθυρο έναρξης οδηγού

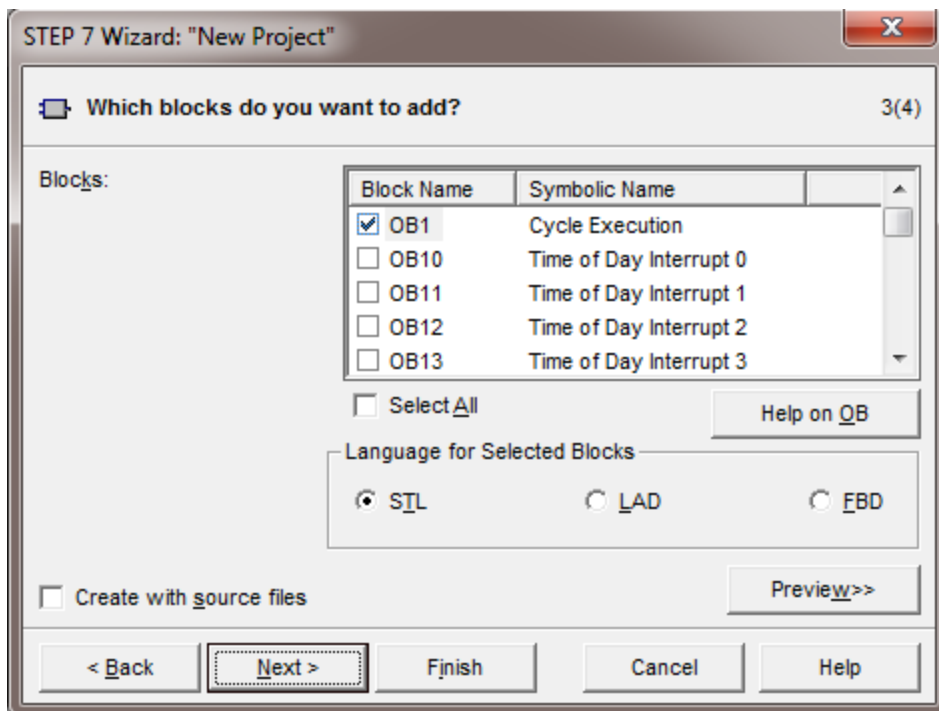
### Βήμα 2<sup>ο</sup>: Επιλογή επεξεργαστή



Εικόνα 3.5: Παράθυρο επιλογής επεξεργαστών και διεύθυνσης MPI

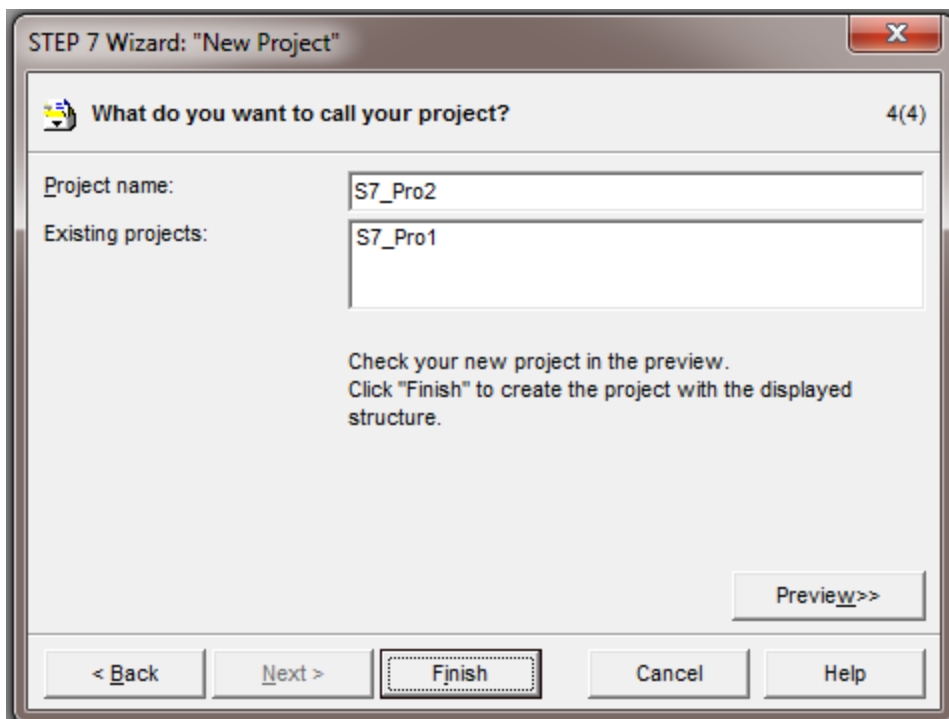


**Βήμα 3<sup>ο</sup>:** Επιλογή και αντικειμένων και γλώσσας προγραμματισμού



**Εικόνα 3.6:** Παράθυρο επιλογής αντικειμένων και γλώσσας

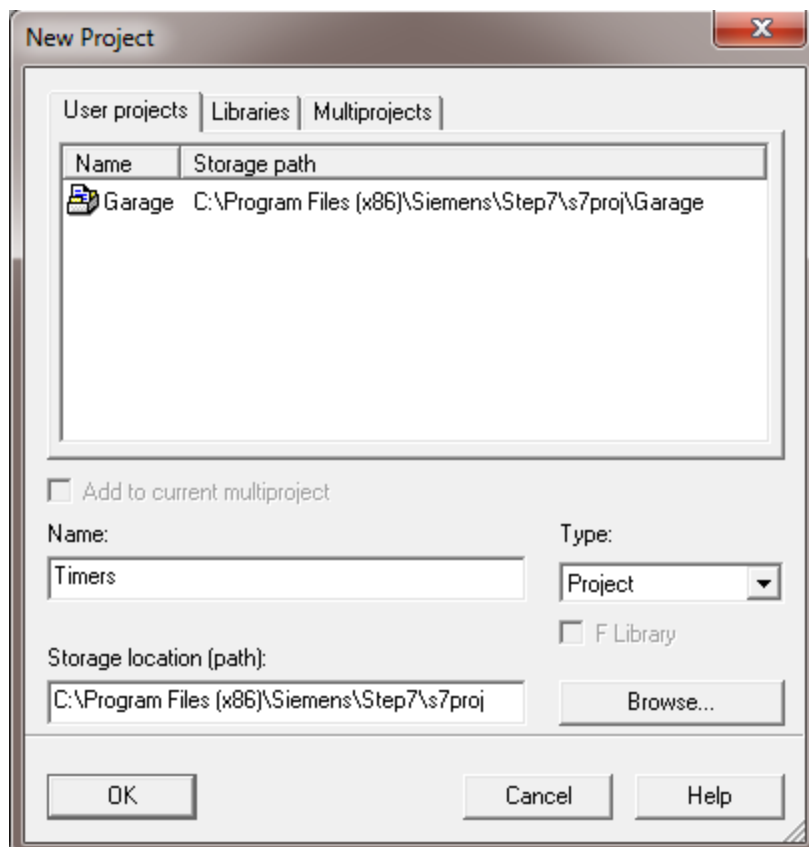
**Βήμα 4<sup>ο</sup>:** Ολοκλήρωση με ονομασία της εργασίας μας



**Εικόνα 3.7:** Παράθυρο με ονομασία εργασίας

Αφού ολοκληρώσουμε αυτά τα βήματα τότε έχουμε έτοιμο τον σταθμό μας και όλα τα αντίστοιχα αντικείμενα που θα μας χρειαστούν. Εκτός από την αυτόματη δημιουργία είπαμε πως υπάρχει και η χειροκίνητη η οποία είναι αρκετά πιο αξιόπιστη αν αναλογιστούμε ότι εμείς επιλέγουμε όλα εκείνα τα στοιχεία που ίσως να έχουν παραληφθεί στον πρώτο τρόπο. Τέτοια αντικείμενα μπορεί να είναι η τροφοδοσία και ο αριθμός εισόδων και εξόδων. Ωστόσο, ο αριθμός εισόδων και εξόδων καθορίζεται και από την CPU που επιλέγουμε κατά την διάρκεια της αυτόματης δημιουργίας.

Αν αποφασίσουμε να διαλέξουμε τα στοιχεία χειροκίνητα, τότε θα πρέπει να ακυρώσουμε (cancel) τον οδηγό δημιουργίας. Κάνοντας κάτι τέτοιο θα μας οδηγήσει σε κενό παράθυρο στον Manager. Από εκεί επιλέγουμε το κουμπί νέου αρχείου που αναφέραμε παραπάνω ή από την γραμμή μενού την διαδρομή: **File → New → Εμφάνιση παραθύρου ονομασίας νέας εργασίας**



**Εικόνα 3.8:** Χειροκίνητη επιλογή ονομασίας νέας εργασίας

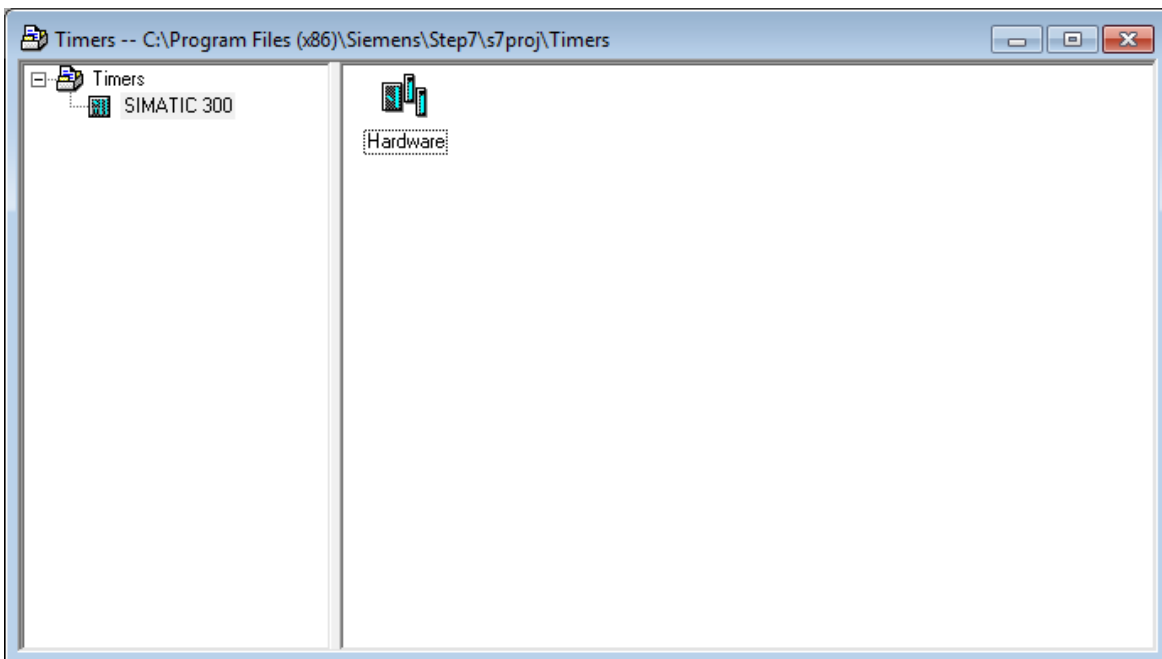
Στο παραπάνω παράθυρο βλέπουμε πως εκτός από την μπάρα ονομασίας, υπάρχει μια λίστα με όλα τα προηγούμενα διαθέσιμα projects. Πατώντας 'OK'

μεταφερόμαστε στην επιφάνεια εργασίας απ'όπου μπορούμε να ξεκινήσουμε με την δουλειά μας.

Με μια πρώτη ματιά θα δούμε πως δεν υπάρχει κάτι στην εργασία μας πέρα από τον ριζικό φάκελο. Για να ξεκινήσουμε την προσθήκη των σταθμών μας πρέπει να κάνουμε κλικ πάνω στον ριζικό φάκελο και μετά στην γραμμή μενού ώστε να επιλέξουμε: **Insert** → **Station** → **Τον σταθμό επιλογής μας**. Μπορούμε να επιλέξουμε από τους σταθμούς:

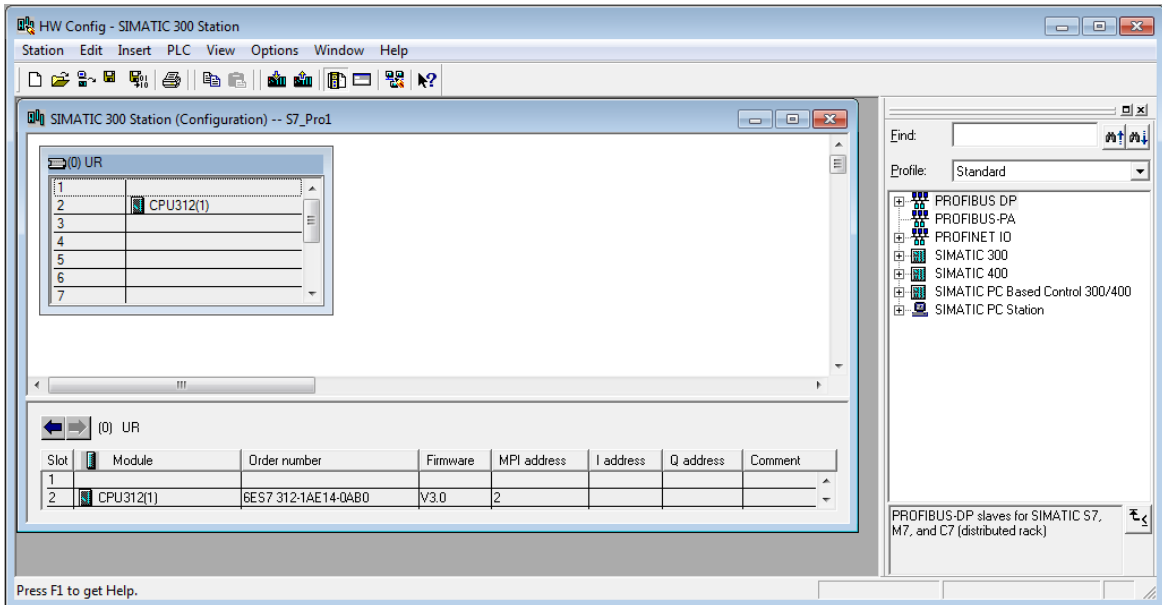
- SIMATIC 300 station
- SIMATIC 400 station
- SIMATIC H station
- SIMATIC PC station
- PC/programming device
- SIMATIC S5
- Other stations, meaning non- SIMATIC S7/M7 and SIMATIC S5

Κάτω από τον ριζικό φάκελο, δημιουργείτε η προσθήκη με το εικονίδιο του σταθμού που επιλέξαμε. Πατώντας πάνω θα δούμε ότι υπάρχει ένα εικονίδιο με τον τίτλο "Hardware". Στο παρακάτω παράδειγμα διαλέγουμε τον σταθμό SIMATIC 300 με το "Hardware" να δηλώνει όλα τα κυκλώματα που είναι απαραίτητα για να λειτουργήσει ο ελεγκτής σωστά.



**Εικόνα 3.9:** Τα στοιχεία της εργασίας μας

Με διπλό κλικ πάνω στο εικονίδιο, ένα νέο παράθυρο ανοίγει και φέρει τον τίτλο “H/W Config”. Το H/W Config περιέχει διάφορα στοιχεία που προστίθενται σιγά-σιγά για την δημιουργία ενός κυκλώματος. Για να μπορέσουμε να προσθέσουμε αυτά τα στοιχεία, μας δίνεται μια λίστα στην αριστερή μεριά του πεδίου εργασίας.



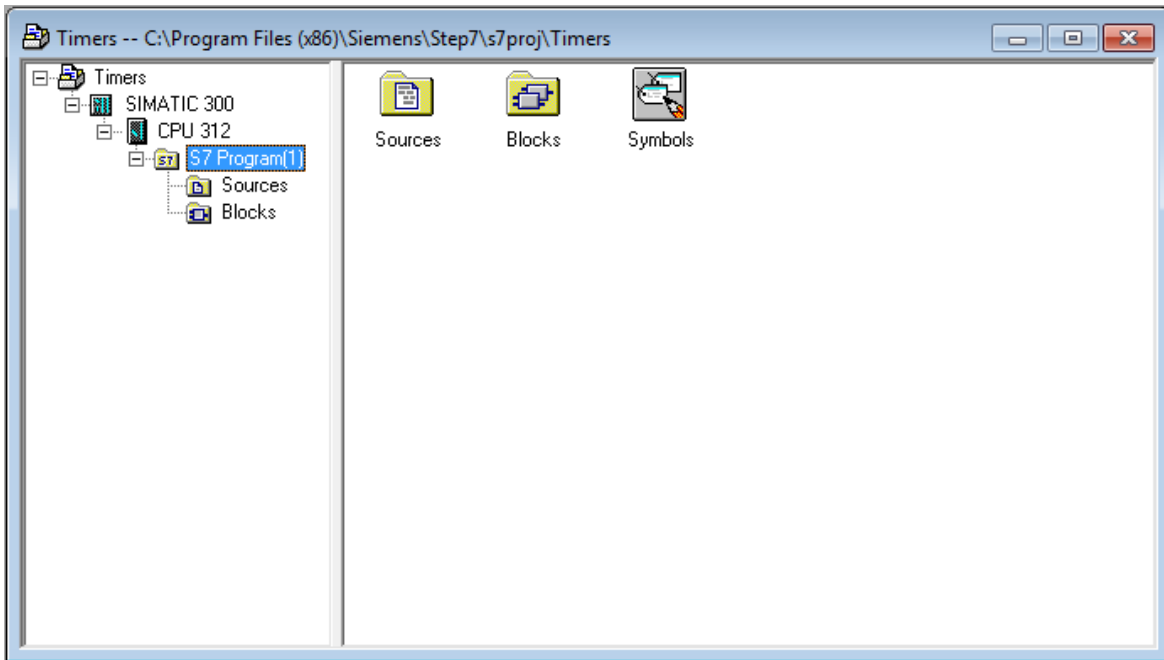
Εικόνα 3.10: H/W Config

Οι περισσότερες προσωμοιώσεις που θα χρησιμοποιήσουμε στο SIMATIC STEP 7 αποτελούνται από το ίδιο Hardware. Μέσω του H/W Config επιλέγουμε το SIMATIC 300. Κάτω από εκεί θα βρούμε αρκετούς φακέλους που περιέχουν τροφοδοτικά και modules για εισόδους, εξόδους και επεξεργαστές. Για να τα προσθέσουμε στο υλικό μας, επιλέγουμε πρώτα τον φάκελο “Rack-300”. Με τον κέρσορα του ποντικιού επιλέγουμε το Rail κάνοντας διπλό κλικ. Ένας νέος πίνακας εμφανίζεται και έτσι μπορούμε να προσθέσουμε νέα αντικείμενα στα κελιά του rack.

Για την τροφοδοσία επιλέγουμε τον φάκελο PS-300 και με τον ίδιο τρόπο που αναφέραμε πιο πριν, και τοποθετούμε στην πρώτη γραμμή το **PS 307 2A**. Αντίστοιχα, για τον επεξεργαστή, επιλέγουμε **CPU-300 → Επεξεργαστή που απαιτεί το πρόβλημα**. Είπαμε πως για την εξαγωγή αποτελεσμάτων, εκτός από την παρουσία εξόδων πρέπει να έχουμε και τις αντίστοιχες εισόδους.






Η εισαγωγή μονάδων εισόδου και εξόδου γίνεται από τους φακέλους SM-300 → DI-300/DO-300 αντίστοιχα. Έπειτα επιλέγουμε μονάδες δεκάεξι εισόδων και εξόδων ως **SM-321 DI16xDC24V για την είσοδο** και **SM-322 DO16Xdc24V/0.5A**






**για την έξοδο.** Για να ολοκληρώσουμε κάθε προσθήκη στο υλικό μας πατάμε το κουμπί “Download to Module” και μετά “Upload to Programming Device”.



**Εικόνα 3.11:** Ολοκληρωμένος σχηματισμός της εργασίας






Στο παράθυρο με τα εξαρτήματα του συστήματος, θα δούμε ότι προστέθηκαν κάτω από τον σταθμό, ο επεξεργαστής που επιλέξαμε και ένας ακόμα φάκελος που περιέχει το κύριο πρόγραμμα. Από την εικόνα 4.11, βλέπουμε πως μερικά νέα εικονίδια δημιουργήθηκαν. Τα εικονίδια αυτά είναι:





Κουμπιά	Λειτουργία
	Εικονίδιο έργου (Project). Δίπλα από αυτό το εικονίδιο έχουμε την ονομασία της εργασίας μας.
	Το εικονίδιο του σταθμού. Με διπλό κλικ μπορούμε να δούμε πληροφορίες για τα αντικείμενα του προγράμματος, να προσθέσουμε σταθμό, να ανεβάσουμε πληροφορίες, να κατεβάσουμε μια έτοιμη προσομοίωση και να δείξουμε πληροφορίες.
	Η συσκευή προγραμματισμού. Συνήθως, αυτό το εικονίδιο υποδηλώνει τον επεξεργαστή που χρησιμοποιείται.
	 Εικονίδιο που παραπέμπει στο παράθυρο δικτύου και συνδέσεων.

		Σύμβολα: Συντόμευση για το παράθυρο συμβόλων. Εκεί δηλώνονται οι μεταβλητές που θα χρησιμοποιήσουμε στα διαγράμματα και τα σχήματά μας.
		Ο φάκελος με τα μπλοκ αντικείμενά μας. Τα αντικείμενα αυτά μπορεί να είναι Object Blocks, Function Blocks, Function Chart Blocks κλπ.
		Εδώ βρίσκονται τα πηγαία αρχεία και οι απαραίτητες πληροφορίες τους. Τα αρχεία αυτά μπορεί να περιέχουν το πηγαίο πρόγραμμα ή πρότυπα διαγραμμάτων.
		Εικονίδιο που δηλώνει αντικείμενο. Με διπλό κλικ πάνω του μεταφερόμαστε στην επεξεργασία των περιεχομένων. Τα περιεχόμενα αυτά αποτελούν και τα διαγράμματα (το κύριο πρόγραμμα) απαραίτητα για την λειτουργία του PLC. Το εικονίδιο βρίσκεται μέσα στον φάκελο των μπλοκ.

**Πίνακας 3.3:** Βασικά εικονίδια του Project

Όταν μπαίνουμε στο εσωτερικό ενός αντικειμένου μπλοκ για την επεξεργασία του, μας ανοίγεται μια νέα επιφάνεια εργασίας στην οποία μας παρέχονται μετρητές, χρονικά και άλλα στοιχεία των γλωσσών προγραμματισμού. Αυτά τα στοιχεία μπορούμε να τα εισάγουμε από την γραμμή εργαλείων (Toolbar):

	Οθόνη προβολής (Debug Mode). Με αυτό το κουμπί μπορούμε να δούμε τον τρόπο λειτουργίας του κυκλώματός μας. Αν οι επικοινωνίες είναι χρωματισμένες με πράσινο σημαίνει ότι υπάρχει λειτουργία.
	Αλλαγή του PLC σε Online Mode – Κατάσταση συνδεδεμένου.
	Προσθήκη ανοιχτής επαφής – Open Contact
	Προσθήκη κλειστής επαφής – Closed Contact
	Προσθήκη πηνίου – Insert Coil

	Προσθήκη κενού κουτιού – Insert empty box. Ανάλογα την ονομασία που θα δώσουμε στο κουτί, μετατρέπεται αυτόματα και στο αντίστοιχο αντικείμενο. Για παράδειγμα, αν βάλουμε όνομα S_PULSE παίρνουμε χρονιστή ως αποτέλεσμα.
	Άνοιγμα νέου βρόγχου – Open Branch
	Κλείσιμο ανοιχτού βρόγχου – Close branch
	Δημιουργία σύνδεσης - Connection

**Πίνακας 3.4:** Βασικά κουμπιά της γραμμής εργαλείων του επεξεργαστή προγράμματος

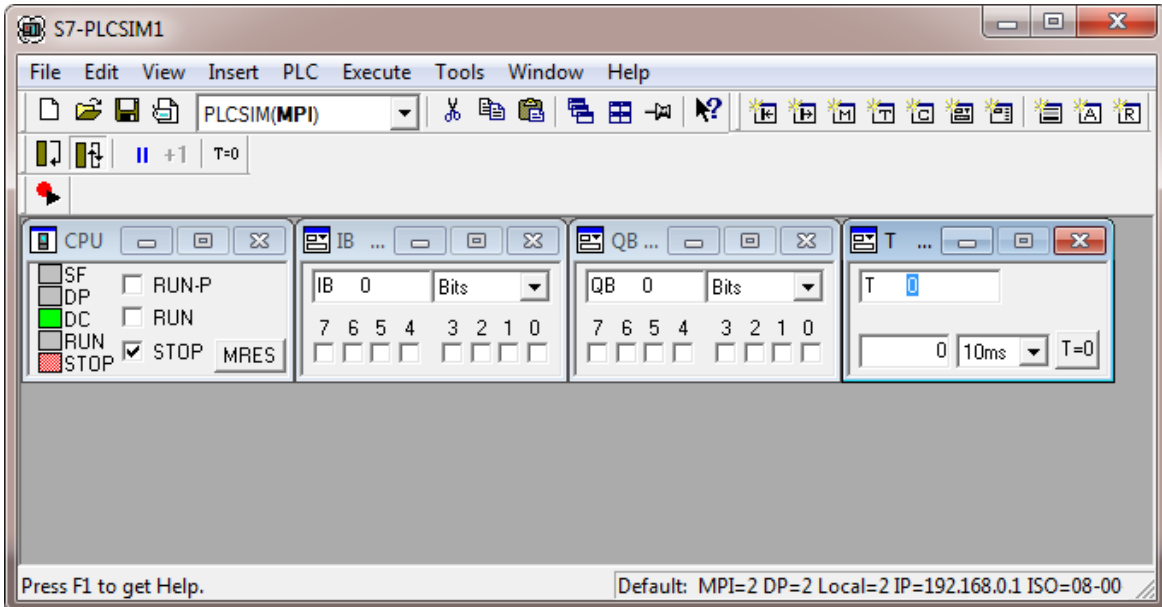
Πριν προχωρήσουμε στην δημιουργία του προγράμματος και την επεξεργασία του, θα πρέπει να λάβουμε υπόψιν μερικά πράγματα για τον προσωμοιωτή.

### 3.3. Το λογισμικό PLCSim

Στο SIMATIC Step 7 γίνεται η χρήση του λογισμικού ελέγχου και προσωμοίωσης PLCSim. Εκεί μπορούμε να εκτελέσουμε τα προγράμματά μας, να επικυρώσουμε την σωστή λειτουργία τους και να συγκεντρώσουμε τα αντίστοιχα αποτελέσματα. Η επιφάνεια εργασίας είναι παρόμοια με αυτή του Manager καθώς υπάρχουν και τα απαραίτητα κουμπιά για στοιχεία όπως μετρητές, χρονικά και μονάδες Bit.









Η επιτυχής φόρτωση του ολοκληρωμένου προγράμματός μας βασίζεται σε απλές κινήσεις που πρέπει να ακολουθήσουμε. Συγκεκριμένα, θα πρέπει να κατεβάσουμε το πρόγραμμα στον ελεγκτή μας αφού έχουμε ανοίξει το PLCSim. Πιο συγκεκριμένα, αφού ολοκληρώσουμε την συγγραφή εργαζόμαστε ως εξής:

- Αποθηκεύουμε το πρόγραμμα και κλείνουμε το παράθυρο επεξεργασίας
- Επιλέγουμε το OB1 αντικείμενο από το παράθυρο της εργασίας μας
- Ανοίγουμε το PLCSim μέσω του αντίστοιχου κουμπιού από την γραμμή εργαλείων
- Πατάμε το κουμπί Download από τον Manager για να κατεβάσουμε το πρόγραμμα στον εικονικό ελεγκτή
- Πηγαίνουμε στο PLCSim και αλλάζουμε από κατάσταση ‘Stop’ σε κατάσταση ‘Run’ για ενεργοποίηση λειτουργίας και προσωμοίωσης του προγράμματος







**Εικόνα 3.12:** Προσωμοιωτής PLCSim

Στην εικόνα 4.12 βλέπουμε τον πίνακα της CPU που έχει τους διακόπτες λειτουργίες ενώ τα υπόλοιπα 3 παραθυράκια ανηππροσωπεύουν την είσοδο (IB), την έξοδο (QB) και έναν χρονιστή. Κάθε στοιχείο που απεικονίζεται εδώ μπορεί να κληθεί με τα αντίστοιχα κουμπιά στην μπάρα πλήκτρων.

Κουμπιά	Λειτουργία
	Εισαγωγή καταχωρητή εισόδου (Input Bits).
	Εισαγωγή καταχωρητή εξόδου (Output Bits).
	Εισαγωγή καταχωρητή μνήμης (Memory Bits).
	Εισαγωγή χρονιστή (Timer).
	Εισαγωγή μετρητή (Counter).
	Εισαγωγή γενικής μεταβλητής (Insert Generic Variable).
	Εισαγωγή κατακόρυφου bit (Insert Vertical Bit).
	Εισαγωγή παραθύρου εκφωλιασμένης στοίβας (Nesting Stack). Η Nested Stack είναι ένα πεπερασμένο αυτόματο που κάνει χρήση



	μιας στοίβας με δεδομένα που μπορεί να αποτελούν εξίσου στοίβες. (RLO, OR).
	Εισαγωγή παραθύρου ελέγχου κατάστασης προγράμματος και επεξεργαστή. Η Λέξη Κατάστασης (Status Word) είναι μια αρχιτεκτονική για τους διαδόχους καταχωρητές ελέγχου που πραγματοποιεί την συνάρτηση καταχωρητών κατάστασης (Status Register) και του μετρητή προγράμματος (Program Counter).
	Εισαγωγή παραθύρου για καταχωρητές Block (Block Regs).
	Κουμπί αλλαγής σε μονή σάρωση (Single Scan).
	Κουμπί αλλαγής σε συνεχή σάρωση (Continuous Scan).

**Πίνακας 3.5:** Βασικά κουμπιά γραμμής εργαλείων του προσωμοιωτή PLCSim

Μαθαίνοντας τα απαραίτητα κουμπιά μπορούμε να προχωρήσουμε στην δημιουργία των προγραμμάτων. Πρέπει να θυμόμαστε πάντα πως μετά την συγγραφή και κατασκευή των σχημάτων επιβάλλεται πρώτα το κατέβασμα του προγράμματος στον εικονικό (ή πραγματικό) ελεγκτή και μετά η εκκίνησή του.

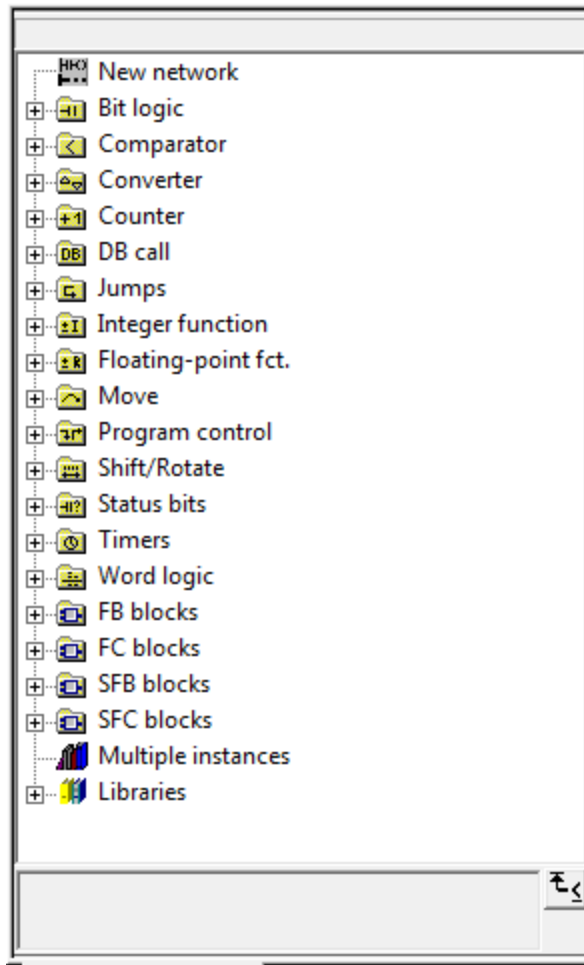


## ΚΕΦΑΛΑΙΟ 4:

### ΠΑΡΑΔΕΙΓΜΑΤΑ

Μετά από την εγκατάσταση, την ρύθμιση του SIMATIC Manager και την σύνδεσή του με όλα τα υπόλοιπα βοηθητικά προγράμματα (PLCSim, WinCC Flexible κ.λ.π.) είμαστε έτοιμοι να σχεδιάσουμε τα προγράμματά μας. Μπορούμε να δημιουργήσουμε προγράμματα για απλή χρονομέτρηση, αντίστροφη ή κανονική μέτρηση αλλά και αυτοματισμούς για μηχανήματα καθημερινής χρήσης όπως θύρες γκαράζ, κυλιόμενες σκάλες και διαδρόμους εργοστασίων.

Το SIMATIC Manager μας προσφέρει μια μεγάλη ποικιλία από έτοιμα μπλοκ διαγράμματα για πιο γρήγορο και αποτελεσματικό προγραμματισμό. Για να καταλάβουμε πιο άμεσα την ευκολία για την οποία κάνουμε λόγο, θα ήταν καλύτερο να προχωρήσουμε σε μερικά παραδείγματα. Ξεκινάμε με απλά παραδείγματα ενώ καταλήγουμε σε μερικά πιο περίπλοκα, χρήσιμα όμως για σύγχρονους αυτοματισμούς. Κάθε παράδειγμα μεταφέρεται από το SIMATIC Manager σε μορφή εικόνων ενώ τα δίκτυα, οι πίλοι και οι σχολιασμοί σε μορφή πινάκων. Η επεξήγηση των βημάτων καθώς και τα αποτελέσματα γίνονται στα πλαίσια των σχολίων. Τέλος, όταν κάνουμε διπλό κλικ πάνω στο εικονίδιο του μπλοκ που θέλουμε να επεξεργαστούμε, τότε μεταβαίνουμε στον χώρο επεξεργασίας ο οποίος περιέχει διάφορα έτοιμα αντικείμενα για το πρόγραμμά μας όπως χρονιστές, μετρητές και μαθηματικές συναρτήσεις:



Εικόνα 4.1: Αντικείμενα του επεξεργαστή μπλοκ

#### 4.1. Παράδειγμα 1<sup>ο</sup>: Χρονιστές

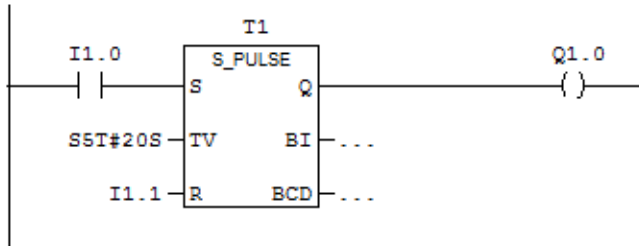
Ένα από τα αρχικά παραδείγματα που θα μας βοηθήσουν στην εξοικείωση με το περιβάλλον εργασίας αλλά και με τον τρόπο λειτουργίας είναι οι χρονιστές. Στο παράδειγμα παρακάτω, θα χρησιμοποιήσουμε τέσσερις τύπους χρονιστών. Πρώτος είναι ο χρονιστής παλμού. Αυτός ο χρονιστής χρησιμοποιείται κυρίως όταν υπάρχει κάποια βλάβη στον αυτοματισμό μας και θέλουμε να ηχεί συναγερμός για συγκεκριμένο χρονικό διάστημα. Επιλέγουμε το **S\_PULSE** Timer από την στήλη ή πατάμε το αντίστοιχο κουμπί για προσθήκη κουπού της γραμμής εργαλείων (επαφές: , κουπά: , πηνία: ). Το πρώτο δίκτυο θα πρέπει να είναι ως εξής:

OB1 : TIMERS

PULSE (SP)	- Pulse Timer (S_PULSE)
EXTENDED PULSE TIMER (SE)	- Extended Pulse Timer (S_PEXT)
ON-DELAY TIMER (SD)	- On-Delay Timer (S_ODT)
RETENTIVE ON-DELAY TIMER (SS)	- Retentive On-Delay Timer (S_ODTS)
OFF-DELAY TIMER (SF)	- Off-Delay Timer (S_OFFDT)

Network 1 : Pulse Timer (S\_PULSE) (SP)

When an error exists from a drop in a thermal engine we want an alarm to go off only for 20 seconds, even if the error persists for more hours.



Σχήμα 4.1: Δίκτυο 1 – S\_PULSE

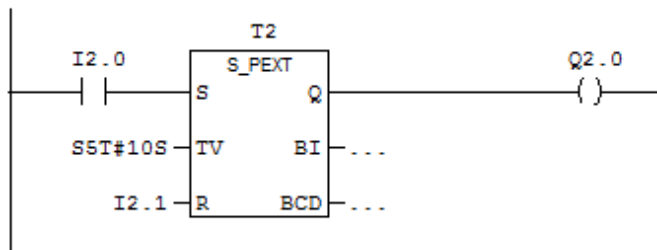
Στην παραπάνω εικόνα βλέπουμε πως το δίκτυό μας έχει τον δικό του τίτλο καθώς και ένα γκρι κενό στο οποίο καταχωρείται το σχόλιό μας. Αυτά τα κενά, δεν χρειάζεται να συμπληρώνονται, βοηθούν όμως σε μια πιο οργανωμένη εργασία και την επεξήγηση της λειτουργίας των ανίστοιχων δικτύων (Networks).

Οι επόμενοι χρονιστές είναι:

- **S\_PEXT** (Extended Pulse Timer – Παλμού με αυτοσυγκράτηση): Χρησιμοποιείται σε περιπτώσεις όπου η επαφή μπορεί να έρθει για μια στιγμή ενώ εμείς θέλουμε έξοδο για μεγαλύτερο χρονικό διάστημα.

Network 2 : ssExtended Pulse Timer (S\_PEXT) (SE)

This timer is used when the contact may happen only for a moment and we want to have an output for a more extended time. So, if we supervise the extension of the upper pressure point in a valve, this may come and go in seconds. With the use of this timer, we can activate an alarm which can sound for much more time.

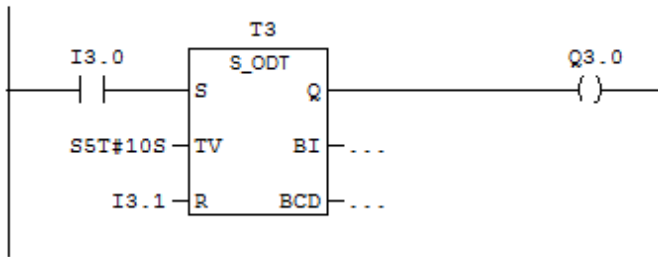


Σχήμα 4.2: Δίκτυο 2 – S\_PEXT

- **S\_ODT** (On-Delay Timer – Καθυστέρησης Έλξης): Χρησιμοποιείται στην περίπτωση ανίχνευσης για ύπαρξη ανάδρασης μετά από μια εντολή ‘Open’ στην βάνα. Αν η εντολή δεν έρθει μετά από κάποιο χρονικό διάστημα, τότε βγαίνει μήνυμα σφάλματος.

Network 3 : On Delay Timer (S\_ODT) (SD)

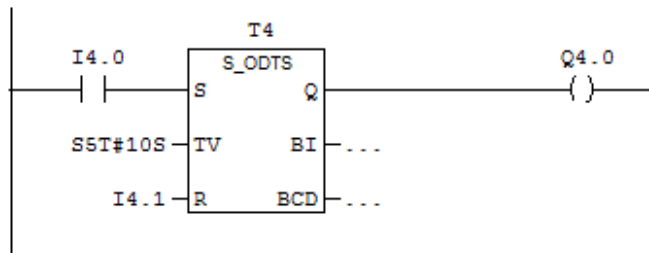
With this timer we can detect any feedback from an "Open" command in a valve. If the Input does not come after 3 seconds, the system will give a failure message because a command was given but was not executed.



Σχήμα 4.3: Δίκτυο 3 – S\_ODT

- **S\_ODTs** (Retentive On-Delay Timer – Καθυστέρησης Έλξης με αυτοσυγκράτηση)

Network 4 : Retentive On-delay timer (S\_ODTS)

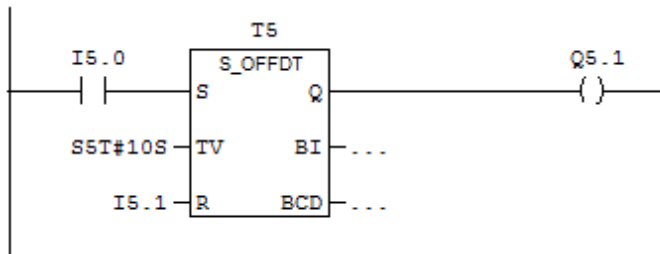


Σχήμα 4.4: Δίκτυο 4 – S\_ODTS

- **S\_OFFDT** (Off-Delay Timer – Καθυστέρησης Πώσης): Χρησιμοποιείται όταν θέλουμε να έχουμε έξοδο για χρονικό διάστημα, ακόμα και μετά την αιτία πρόκλησης της ενεργοποίησης της εξόδου αυτής.

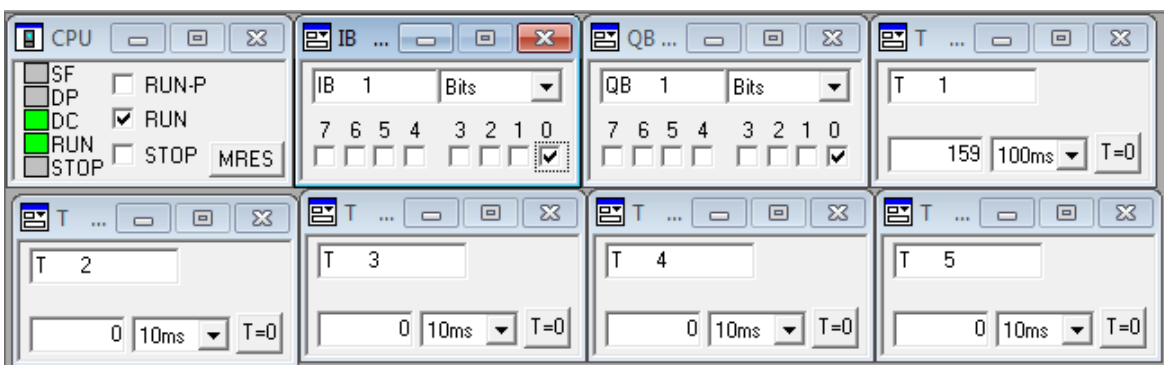
**Network 5 : Off-delay Timer (S\_OFFDT) (SP)**

This timer is used when we want to produce an output for a specific duration even after we receive the reason of the activation. A known example is the lights of a staircase. We want the lights to turn on automatically when the move detector receives the existence of a human on the stairs, and to turn off when the human leaves.



**Σχήμα 4.5:** Δίκτυο 5 – S\_OFFDT

Παρατηρούμε πως σε κάθε χρονιστή έχουμε βάλει στο Time Variable την πιμή S5T#10s. Το S5T αποτελεί την δήλωση χρόνου ενώ το κομμάτι μετά το Sharp (#) είναι το χρονικό διάστημα που θέτουμε σε μονάδες χρόνου (συνήθως δευτερόλεπτα). Με το που προσθέσουμε όλα τα χρονικά με τα αντίστοιχα ονόματα, τις εισόδους και τις εξόδους ανοίγουμε το PLCsim και κάνουμε “Download” το OB1 Block στο εικονικό PLC. Έπειτα, πάμε στο PLCsim και από την κατάσταση ‘STOP’ το θέτουμε στην κατάσταση ‘RUN’. Για κάθε είσοδο με όνομα Ix.x έχουμε και την αντίστοιχη έξοδο Qx.x+1. Οπότε, αν θέλουμε να έχουμε αποτελέσματα για το χρονιστή T1 τότε θέτουμε την είσοδο I1 στο bit 0 και ενεργοποιείται ο χρονιστής ενώ βλέπουμε το αποτέλεσμα της λειτουργίας με την βοήθεια του bit εξόδου Q1. Οπότε:



**Εικόνα 4.2:** Αντίστροφη μέτρηση του χρονιστή S\_PULSE

Στον πρώτο Timer έχουμε θέσει τον χρόνο να ξεκινάει από τα 20 δευτερόλεπτα. Αυτό γιατί θεωρούμε πως έχουμε έναν αυτοματισμό στον οποίο προέκυψε σφάλμα

αλλά η ειδοποίηση θέλουμε να διαρκεί μόνο 20 δευτερόλεπτα ακόμα και αν το σφάλμα συνεχίζει να υφίσταται. Με τον ίδιο τρόπο βλέπουμε και τα αποτελέσματα για τους υπόλοιπους χρονιστές.

## 4.2. Παράδειγμα 2<sup>ο</sup>: Μετρητές

Από τα βασικά στοιχεία που χρησιμοποιούνται στον προγραμματισμό με STEP 7 αλλά και στον προγραμματισμό ελεγκτών γενικότερα, είναι οι μετρητές (Counters). Τα κυριότερα είδη μετρητών είναι ο μετρητής προς τα άνω (Counter Up - CU), ο μετρητής προς τα κάτω (Counter Down - CD) και ο μετρητής άνω-κάτω (Counter Up/Down – CUD). Αντίθετα με τους χρονιστές που χρονομετρούν αυτόματα και αυτόνομα, οι μετρητές χρειάζονται πολλές φορές την παρέμβασή μας για να κάνουν την μέτρηση. Ωστόσο, μπορούμε να συνδιάσουμε τους χρονιστές με τους μετρητές για να γίνεται η μέτρηση αυτόματα ή ακόμα καλύτερα να συνδιάσουμε διαδρόμους με φωτοκύπαρα για ευκολότερη μέτρηση.

Παρακάτω, εισάγουμε τους μετρητές που αναφέραμε και ελέγχουμε την λειτουργία τους με το PLCsim.

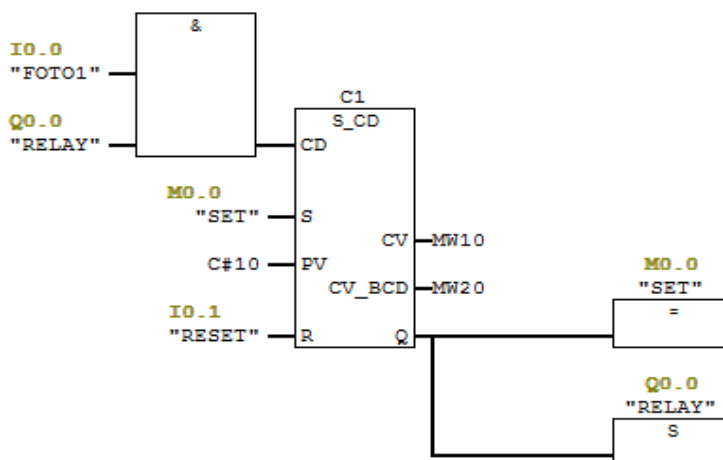
- Μετρητής προς τα κάτω (DOWN Counter)

OBI : Example with DOWN Counter

We have a photoelectric cell. In front of it we have passing boxes. With a SET Button we preset the counter with the value '10'. With a RESET button we revert back to 0. When 10 boxes have passed, a relay output is activated.

Network 1: Title:

Automatic setting of the counter value



Σχήμα 4.6: Μετρητής Down με χρήση της FBD γλώσσας



Στο σχήμα της εικόνας 4.8 βλέπουμε πως γίνεται η χρήση της γλώσσας FBD. Αν θέλουμε να αλλάξουμε σε LAD το κάνουμε εύκολα από την γραμμή μενού στις επιλογές. Συνήθως, η μετάφραση γίνεται αυτόματα χωρίς πρόβλημα.

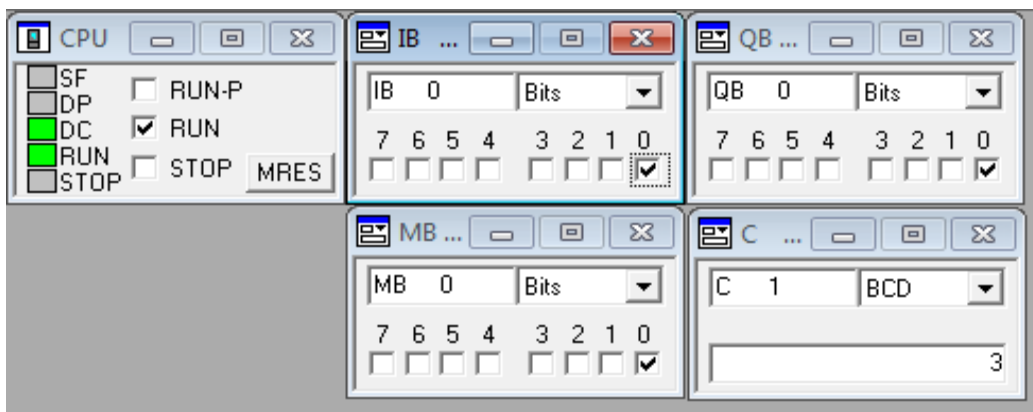
Στον μετρητή down βλέπουμε τέσσερα μπλοκς. Ένα μπλοκ AND που θέτουμε την είσοδο και την έξοδο στον μετρητή, το μπλοκ του μετρητή, το μπλοκ “set” το οποίο θέτει την τιμή από την οποία θα αρχίσει η μέτρηση και το “relay” που αποτελεί και την κύρια έξοδό μας.

Στο συγκεκριμένο παράδειγμα, έχουμε έναν εργοστασιακό αυτοματισμό διαδρόμου στον οποίο υπάρχουν φωτοκύπαρα. Μετά από το πέρασμα δέκα κιβωτίων, έχουμε ενεργοποίηση του ρελέ εξόδου ενώ αν θέλουμε να επανεκκινήσουμε την διαδικασία, μπορούμε να κάνουμε “reset” το σύστημα.

Τα συμβολικά ονόματα αναθέτονται από το εικονίδιο “Symbols” του φακέλου “S7 Program”. Εκεί μπορούμε να βάλουμε ένα όνομα σε κάθε είσοδο, έξοδο και μεταβλητή που χρησιμοποιούμε ώστε να την καλούμε ευκολότερα στα στοιχεία μας. Εδώ οι συμβολικές ονομασίες είναι οι:

Status	Symbol	Address	Data Type	Comment
	Cycle execution	OB 1	OB 1	
	FOTO1	I 0.0	BOOL	
	RELAY	Q 0.0	BOOL	
	RESET	I 0.1	BOOL	
	SET	M 0.0	BOOL	

**Πίνακας 4.1:** Συμβολική ονομασία στοιχείων μετρητή προς τα κάτω



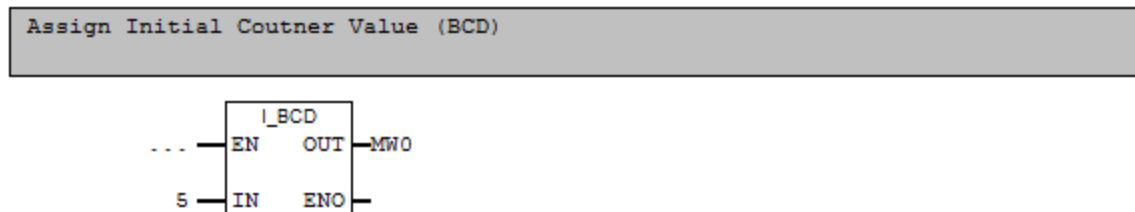
**Εικόνα 4.3:** Μετρητής Down με χρήση της FBD γλώσσας

Βλέπουμε πως στην κατάσταση 'Run' η είσοδος και η έξοδος θέτονται ως I0.0 και Q0.0 αντίστοιχα με το Memory Bit (MB) να είναι και αυτό M0.0. Αυτό σημαίνει πως όταν το Memory Bit είναι στην κατάσταση αυτή, τότε ο αυτοματισμός μας ξεκινάει να μετράει ανίστροφα από το 10 με την ύπαρξη εισόδου. Τα CV και CV\_BCD στα πλάγια του μετρητή χρησιμοποιούνται για να δούμε την τιμή του μετρητή με το CV να είναι μια δυαδική αναπαράσταση της τιμής και το CV\_BCD μια δυαδικά κωδικοποιημένη σε δεκαδικό, αναπαράσταση.

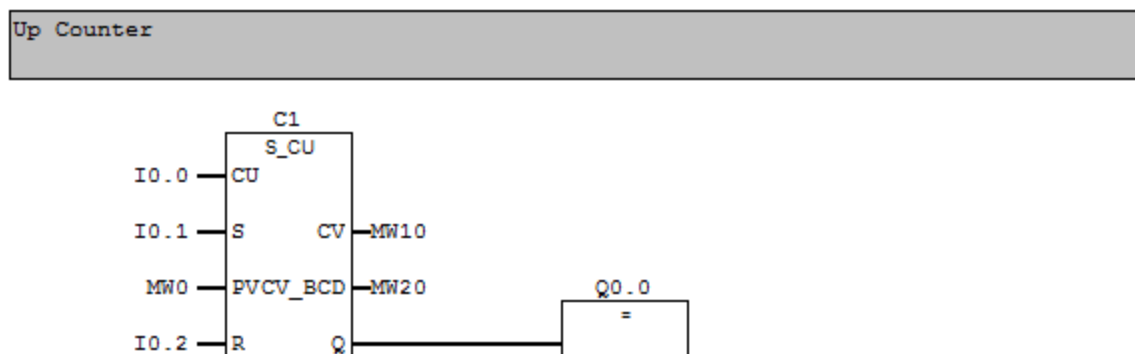
Είπαμε παραπάνω, πως η παρούσα διαδικασία γίνεται χειροκίνητα. Οπότε, κάθε φορά που περνάει ένα κιβώτιο ενεργοποιούμε τον μετρητή να κατεβάσει την τιμή του, κλείνοντας και ανοίγοντας την είσοδο I0.0 (υπάρχει το τετραγωνάκι επιλογής γύρω από το Bit 0 του IB 0).

Εκτός από τον Down μετρητή, μπορούμε να δούμε την συμπεριφορά όλων των τριών μετρητών, ταυτόχρονα. Ξεκινούμε λοιπόν, φτιάχνοντας το πρώτο δίκτυο και θέτοντας την αρχική τιμή όλων των μετρητών.

**Network 1 :**



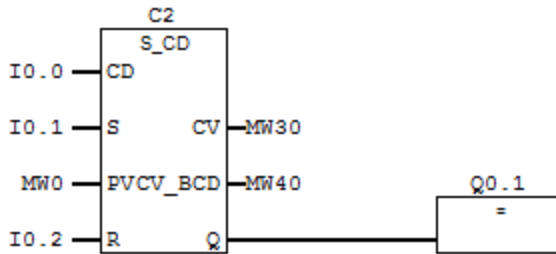
**Network 2 :**



**Σχήμα 4.7:** Δίκτυα 1 και 2 – Αρχική τιμή και μετρητής προς τα άνω

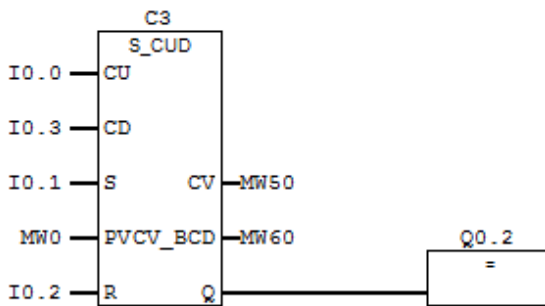
Και συνεχίζουμε με τους υπόλοιπους δύο μετρητές:

Network 3 :



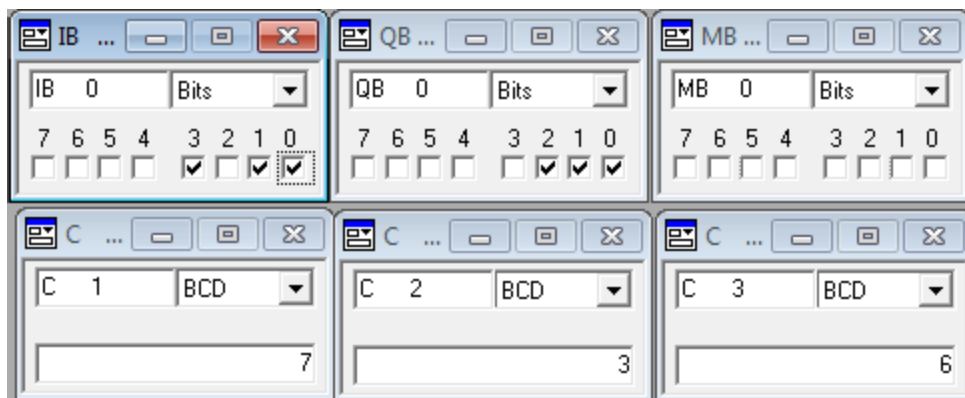
Σχήμα 4.8: Δίκτυο 3 – Μορφή μετρητή προς τα κάτω

Network 4 :



Σχήμα 4.9: Δίκτυο 4 – Μορφή μετρητής προς άνω και κάτω

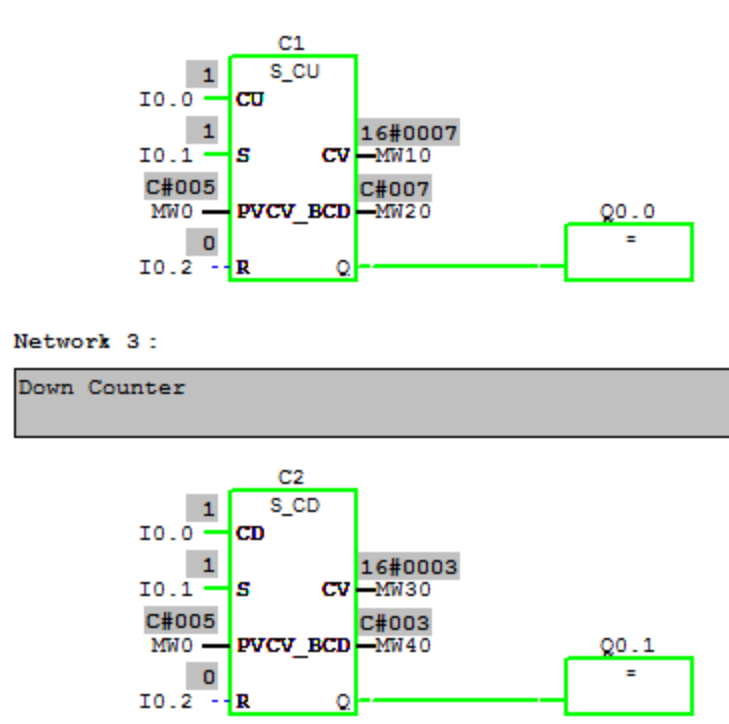
Κατεβάζοντας το πρόγραμμα στο PLCSim, το εκκινούμε με το 'Run'. Αφού γίνει αυτό συνεχίζουμε με τις εισόδους I0.1 για να θέσουμε τις τιμές στους μετρητές και το I0.0 για να αρχίσει η μέτρηση.



Εικόνα 4.4: Αποτελέσματα μετρητών

Παρατηρούμε πως ο μετρητής προς τα άνω μετράει κανονικά προς τα πάνω, ο μετρητής προς τα κάτω συνεχίζει την αντίστροφη μέτρηση κατά ένα ενώ ο μετρητής προς τα άνω/κάτω μετράει αλλά και αφαιρεί κατά ένα.

Γυρνώντας το πρόγραμμά μας σε Debug mode, θα δούμε ποια κομμάτια των μετρητών λειτουργούν κατά την διάρκεια της προσωμοίωσης:

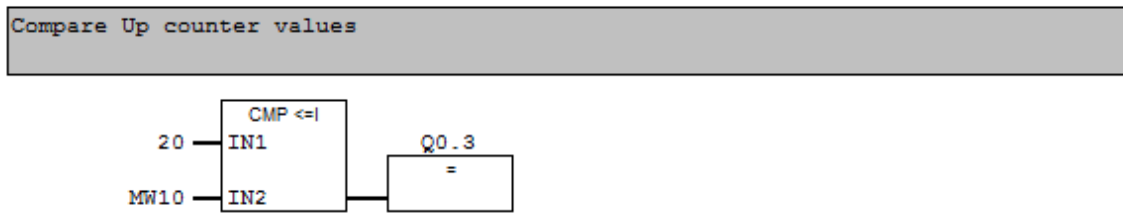


**Σχήμα 4.10:** Τα δίκτυα κατά την διάρκεια λειτουργίας του Debug Mode

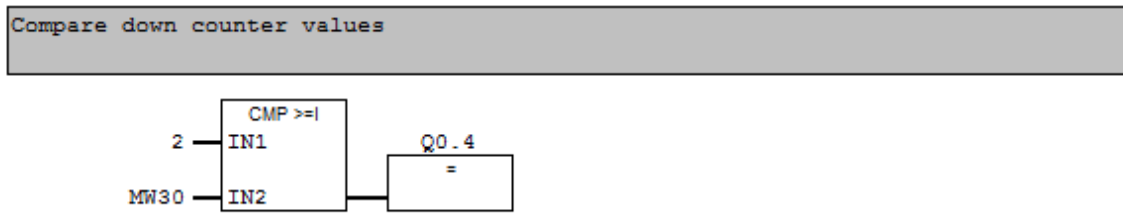
Εκτός από τις πράσινες γραμμές λειτουργίας, πάνω σε κάθε στοιχείο (CU, S, κ.λ.π.) υπάρχει κάποια μεταβλητή. Αυτή η μεταβλητή δηλώνει μια καταχωρημένη τιμή ή την κατάσταση στην οποία βρίσκεται εκείνη την στιγμή το στοιχείο αυτό. Στον μετρητή C1, το Counter Up (CU) έχει την τιμή 1. Αυτό σημαίνει πως είναι ενεργοποιημένο και έτσι ενεργοποιείται ο μετρητής. Το PVCV έχει την τιμή C#005 δηλαδή ο μετρητής C ξεκινάει από την ήδη δοθείσα τιμή 5.

Για να φανούν οι τιμές των μετρητών ξεχωριστά και σωστά, βάζουμε τρία παραπάνω δίκτυα που αποτελούν συγκρίσεις των τιμών που εισάγονται. Για τον κάθε μετρητή, η σύγκριση είναι διαφορετική:

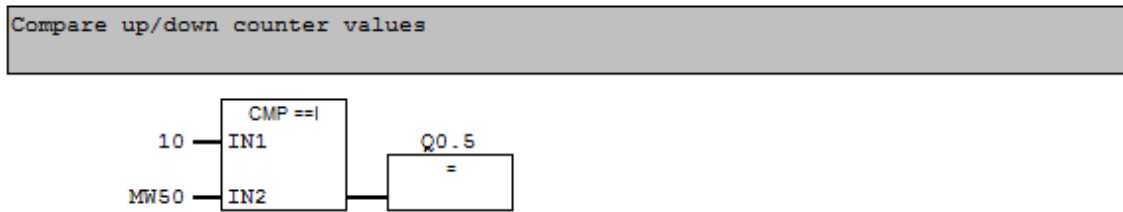
Network 5 :



Network 6 :



Network 7 :



**Σχήμα 4.11:** Τα δίκτυα για την σύγκριση – απαραίτητα για την εμφάνιση των τιμών στις εξόδους

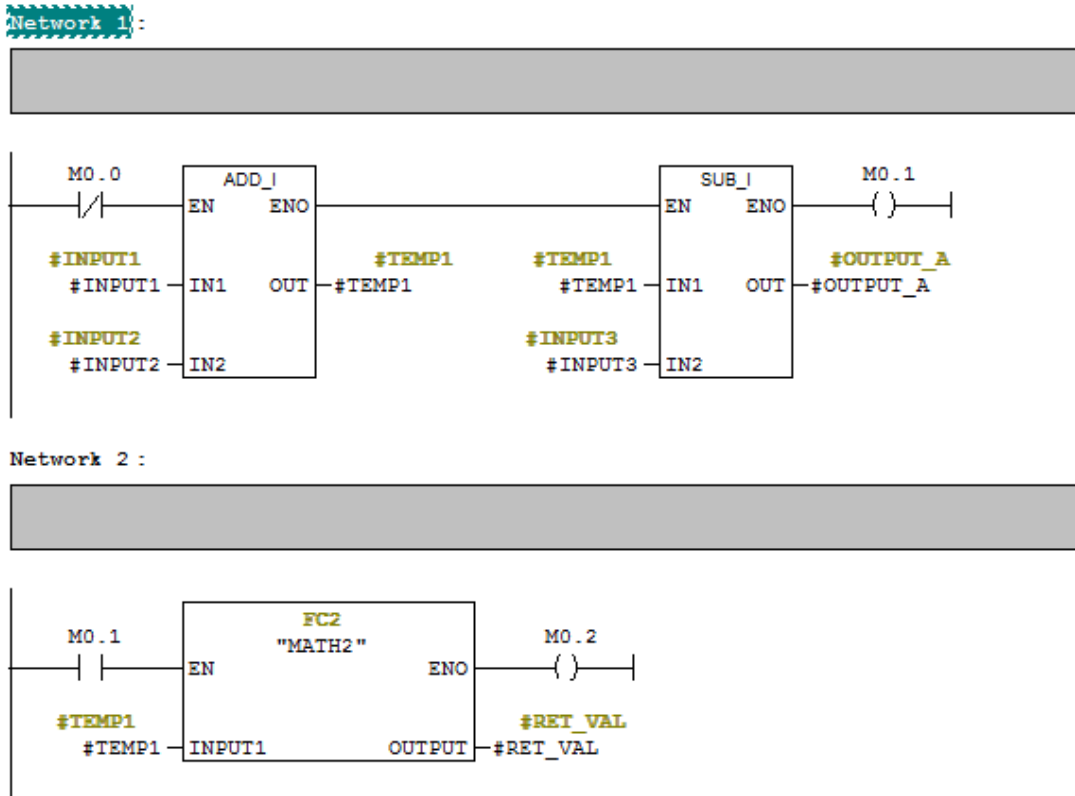
Παραπάνω αναφέραμε πως είναι δυνατόν να δημιουργήσουμε ένα παρόμοιο πρόγραμμα στο οποίο η μέτρηση θα γίνεται αυτόματα με την βοήθεια χρονοστών. Αυτή η εναλλακτική χρησιμοποιείται σε διαδρόμους μέτρησης κιβωτίων με δύο φωτοκύπαρα ή σε κυλιόμενες σκάλες. Αν θέλουμε να προσθέσουμε χρονοστές στους μετρητές μας, εισάγουμε δύο χρονοστές παλμού με διαφορετικές εισόδους και εξόδους. Αυτό που θέλουμε είναι να λειτουργούν εναλλάξ, σε διαφορετικές χρονικές στιγμές. Μετά την είσοδο κάθε χρονοστής, θέτουμε μια κλειστή επαφή οπότε με το που τελειώσει το ένα χρονόμετρο το άλλο αρχίζει και μετράει. Αφού και τα δύο χρονόμετρα ολοκληρώσουν έναν κύκλο λειτουργίας, ο μετρητής που έχει σαν είσοδο την έξοδο του δεύτερου χρονομέτρο, μετράει κατά ένα (προς τα κάτω ή προς τα άνω, ανάλογα το είδος του μετρητή).

### 4.3. Παράδειγμα 3<sup>ο</sup>: Χρήση της κλήσης συναρτήσεων

Στον κλασικό προγραμματισμό, η χρήση των συναρτήσεων για την λύση των προβλημάτων αποτελεί ένα συνηθισμένο τρόπο εργασίας. Τις συναρτήσεις αυτές μπορούμε να τις πληκτρολογήσουμε κατευθείαν μέσα σε ένα πρόγραμμα ή να τις γράψουμε ξεχωριστά σε άλλο αρχείο και να τις καλέσουμε όταν τις χρειαζόμαστε, εξοικονομώντας χωρικότητα στην μνήμη αλλά και γραμμές στον κώδικα.

Εκτός από τη συμβολική ονομασία, το SIMATIC Manager διαθέτει ένα ξεχωριστό μπλοκ για την συγγραφή τέτοιων συναρτήσεων. Το Function Call (FC) θα μας χρησιμεύσει αρκετά προκειμένου να έχουμε μια πιο οργανωμένη εμφάνιση στο πρόγραμμά μας, ειδικά αν αυτό είναι πιο περίπλοκο απ'ότι είχαμε υπολογίσει αρχικά.

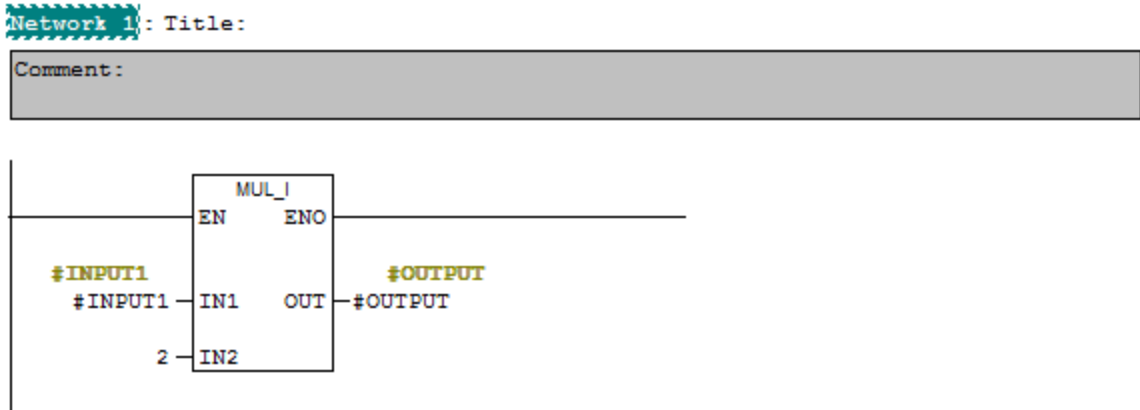
Στο παρακάτω παράδειγμα, κάνουμε μια εισαγωγή στην χρήση της κλήσης συναρτήσεων με την βοήθεια μαθηματικών πράξεων πρόσθεσης και πολλαπλασιασμού.



**Σχήμα 4.12:** Τα δίκτυα του μπλοκ συναρτήσεων FC1

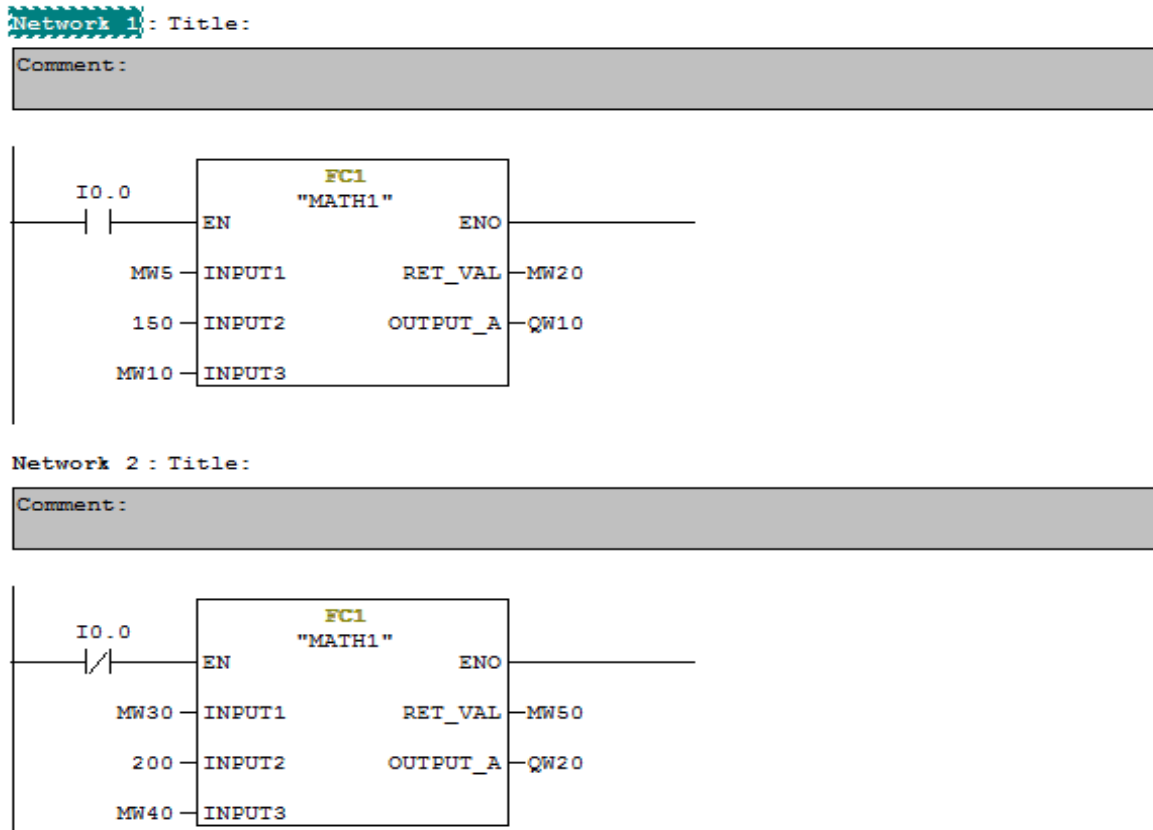
Στο FC1 έχουμε την εισαγωγή δύο αριθμητικών πράξεων με το δεύτερο δίκτυο να καλεί την FC2. Οι πράξεις που θα χρησιμοποιήσουμε είναι η πρόσθεση με την αφαίρεση να παίρνει το αποτέλεσμα της πρόσθεσης και να το αφαιρεί από μια άλλη

μεταβλητή. Το αποτέλεσμα που θα δωθεί από αυτές τις πράξεις, θα πολλαπλασιαστεί επί δύο για να δώσει το τελικό αποτέλεσμα.



Σχήμα 4.13: Το δίκτυο του μπλοκ συναρτήσεων FC2

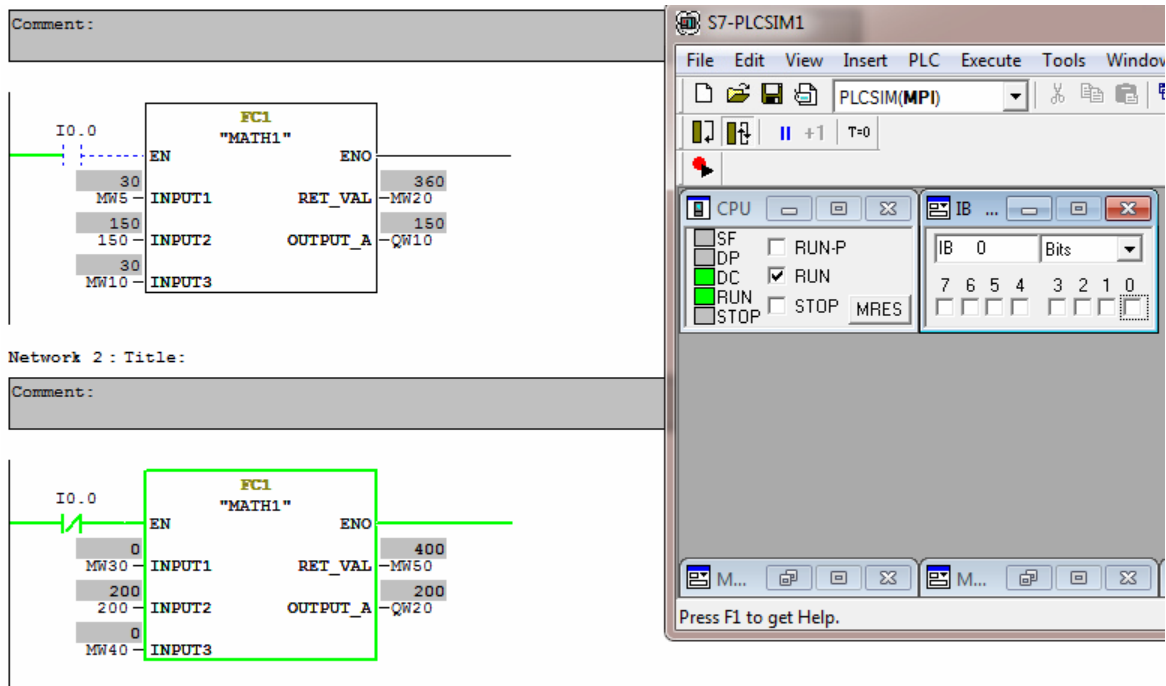
Τέλος, το OB1, δηλαδή το Object Block που έχει το κυρίως πρόγραμμα θα είναι ως εξής:



Σχήμα 4.14: Το βασικό Block του προγράμματος

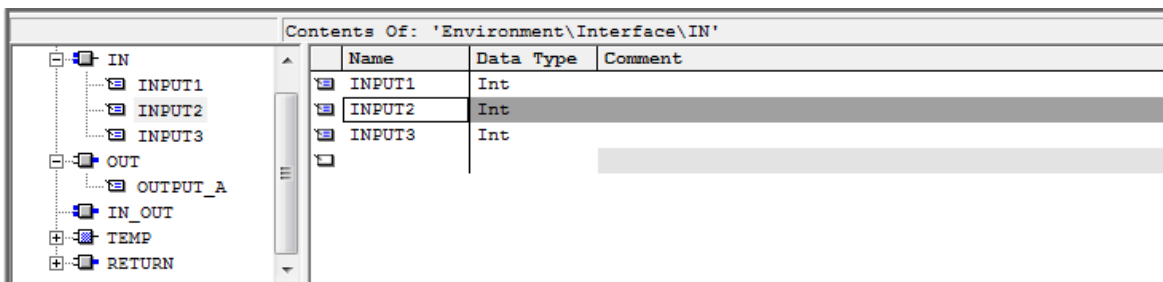
Και στα δύο δίκτυα γίνεται κλήση της ίδιας συνάρτησης με την διαφορά να φαίνεται στις μεταβλητές εισόδου και την συμβολογραφία τους καθώς και στον τρόπο που

γίνεται η ενεργοποίηση. Στο πρώτο δίκτυο, θα πρέπει το Bit 0 να είναι στο λογικό '1' ενώ στο δεύτερο δίκτυο θα πρέπει να βρίσκεται σε λογικό '0'. Για να καταλάβουμε, τρέχουμε το debug mode του λογισμικού:



**Εικόνα 4.5:** Το Bit 0 βρίσκεται σε λογικό '0' και ενεργοποιείται το δεύτερο δίκτυο

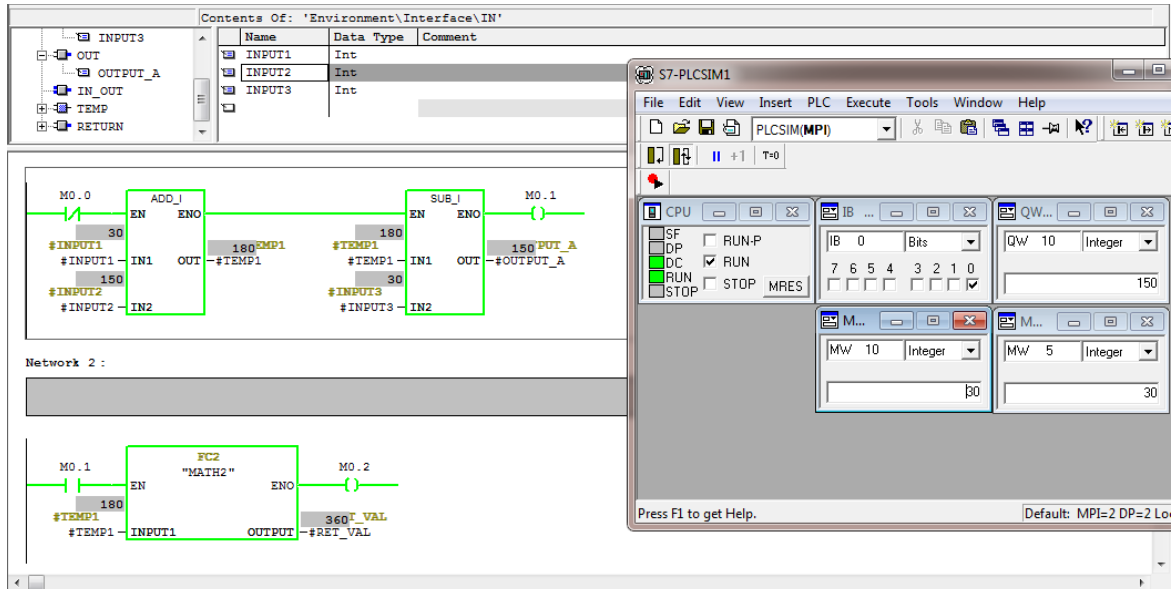
Για να λειτουργήσουν σωστά τα αποτελέσματα, εκτός από τόν συμβολισμό των μεταβλητών (που δείξαμε στο προηγούμενο παράδειγμα), πρέπει να μετατρέψουμε τα INPUT και OUTPUT σε ακεραίους (Integer) αλλιώς το αποτέλεσμα θα δίνει σφάλμα. Για να το κάνουμε αυτό επιλέγουμε τα block συναρτήσεων FC1, FC2 και πάνω από το παράθυρο επεξεργασίας θα βρούμε τα περιεχόμενα των μπλοκ:



**Εικόνα 4.6:** Αλλαγή στο Data Type



Ξεκινούμε την προσωμοίωση και ενεργοποιούμε το Debug Mode στο FC1 για να εξετάσουμε τον τρόπο λειτουργίας των συναρτήσεων. Θα δούμε πως με τις μεταβλητές γενικής σημασίας (MWx) βάζουμε τις δικές μας εισόδους ως ακεραίους ενώ έχουμε ήδη μια είσοδο προκαθορισμένη.



Εικόνα 4.7: Αποτελέσματα πράξεων στην προσωμοίωση

Στο πρώτο μπλοκ γίνεται η πρόσθεση των τριών αριθμών και καταγράφεται το αποτέλεσμα. Μετά αυτό περνάει στο επόμενο μπλοκ από το οποίο γίνεται η αφαίρεσή του με μια τρίτη μεταβλητή εισόδου. Και πάλι, το νέο αποτέλεσμα καταγράφεται στην OUTPUT\_A.

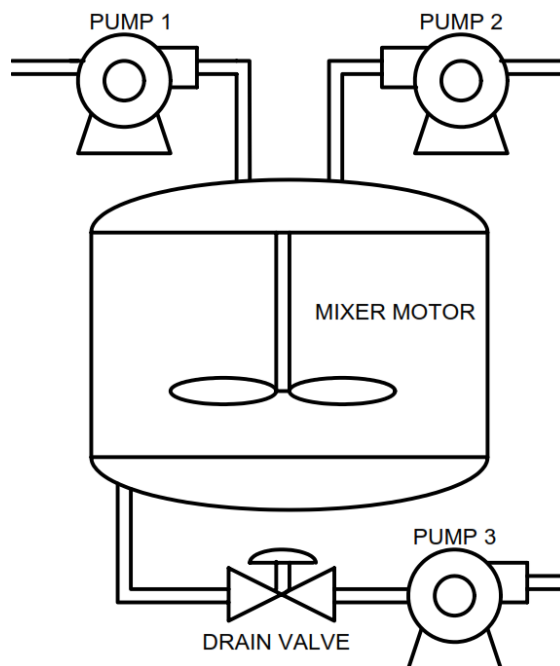
Στο δεύτερο μπλοκ, το άρθροισμα που έχει καταχωρηθεί στο TEMP1 πολλαπλασιάζεται με το δύο και δίνει ένα νέο, διαφορετικό αποτέλεσμα. Στο debug mode βλέπουμε κάθε αριθμό που έχουμε καταχωρήσει πάνω από το αντίστοιχο σύμβολο.

#### 4.4. Παράδειγμα 4<sup>ο</sup>: Δεξαμενή

Η δεξαμενή που φαίνεται στο παρακάτω σχήμα θα γεμίσει με δύο χημικά υγρά, τα οποία πρώτα θα ανακατευτούν και στην συνέχεια θα διοχετευτούν για την υλοποίηση μιας άλλης διαδικασίας. Όταν πατήσουμε το κουμπί εκκίνησης START, δηλαδή όταν θα έχουμε λογικό 1 στην είσοδο I0.0, το πρόγραμμα ενεργοποιεί την αντλία 1 (PUMP 1) η οποία ελέγχεται από την έξοδο Q0.0. Η αντλία 1 ενεργοποιείται για 5 δευτερόλεπτα, γεμίζοντας την δεξαμενή με το πρώτο υγρό και μετά με την πάροδο του χρόνου απενεργοποιείται. Το πρόγραμμα έπειτα ενεργοποιεί την αντλία 2, η οποία ελέγχεται από την έξοδο Q0.1. Η δεύτερη αντλία ενεργοποιείται για 3 δευτερόλεπτα, γεμίζοντας την δεξαμενή με το δεύτερο υγρό. Μετά την πάραδο των 3sec απενεργοποιείται.

Μετά από αυτά, το πρόγραμμα ενεργοποιεί τον αναδευτήρα (Mixer Motor), ο οποίος ελέγχεται από την έξοδο Q0.2 και ανακατεύει τα 2 υγρά για ένα λεπτό. Το πρόγραμμα μετά, ενεργοποιεί την βαλβίδα διοχέτευσης (Drain Valve), η οποία ελέγχεται από την έξοδο Q0.4 όπως επίσης και την αντλία 3 που ελέγχεται από την αντλία Q0.3. Η αντλία 3 και η βαλβίδα διοχέτευσης απενεργοποιούνται μετά από 8 sec και η διαδικασία σταματά. Επιπροσθέτως, έχουμε την δυνατότητα να σταματήσουμε την διαδικασία πατώντας το κουμπί STOP που συνδέεται με την I0.1.

Η δεξαμενή μας φαίνεται παρακάτω:



**Σχήμα 4.15:** Η δεξαμενή TANK

Πριν ξεκινήσουμε την συγγραφή, ξεκινάμε με τον συμβολισμό:

Status	Symbol	Address	Data Type	Comment
	Cycle execution	OB 1	OB 1	
	MIXER	Q 0.2	BOOL	
	PUMP1	Q 0.0	BOOL	
	PUMP2	Q 0.1	BOOL	
	PUMP3	Q 0.3	BOOL	
	START	I 0.0	BOOL	
	STOP	I 0.1	BOOL	
	VALVE	Q 0.4	BOOL	

**Πίνακας 4.2:** Σύμβολα δεξαμενής

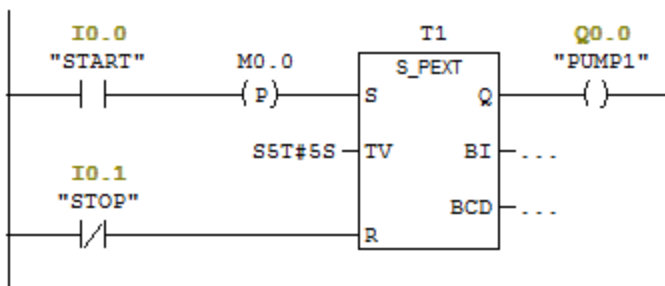
Το πρόγραμμά μας θα αποτελείται από τέσσερα δίκτυα. Το δίκτυο ενεργοποίησης αντλίας 1, το δίκτυο ενεργοποίησης αντλίας 2, η ενεργοποίηση αναδευτήρα και για την ενεργοποίηση αντλίας 3 και βαλβίδας διοχέτευσης.

### Δίκτυο 1: Ενεργοποίηση αντλίας 1

OB1 : TANK



Network 1 : PUMP ACTIVATION



**Σχήμα 4.16:** Δίκτυο 1 - Ενεργοποίηση PUMP 1

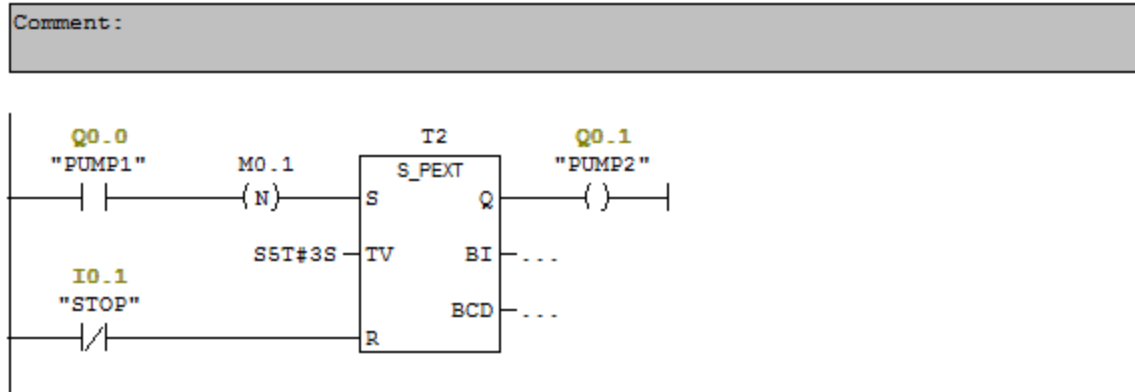
Όταν πατήσουμε το κουμπί εκκίνησης, όταν δηλαδή ενεργοποιηθεί με το λογικό 1 η είσοδος I0.0, θα ενεργοποιείται αντίστοιχα η αντλία 1. Μια καλύτερη τεχνική είναι

να ανιχνεύσουμε την αλλαγή του σήματος από λογικό 0 σε 1 για να ενεργοποιηθεί το PUMP1 για την περίπτωση που το START κουμπί, κολλήσει.

Με την βοήθεια του M0.0 ανιχνεύεται η αλλαγή του σήματος από 0 σε 1 και ενεργοποιείται η αντλία 1 για 5 δευτερόλεπτα και με την πάροδο του χρόνου απενεργοποιείται. Αν κατά την διάρκεια αυτή πατήσουμε το STOP, η αντλία σταματάει.

## Δίκτυο 2: Ενεργοποίησης αντλίας 2

Network 2 : ACTIVATION OF THE SECOND PUMP

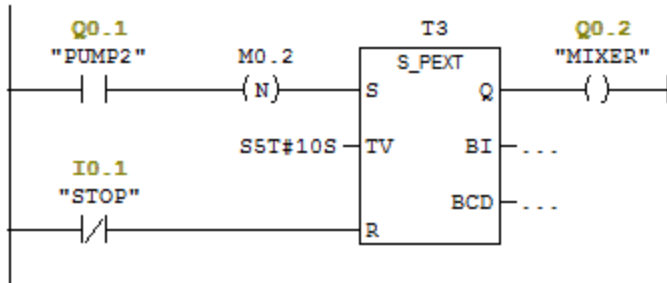


Σχήμα 4.17: Δίκτυο 2 - Ενεργοποίηση PUMP 2

Όταν επενεργοποιείται η αντλία έχουμε αλλαγή σήματος. Με την βοήθεια του M0.1 ανιχνεύουμε την αλλαγή του παλμού σε λογικό '0' ώστε να γίνει η ενεργοποίηση της δεύτερης αντλίας. Η αντλία 2 με τον τρόπο αυτό ενεργοποιείται για 3 δευτερόλεπτα και μετά σταματά. Αν κατά την διάρκεια των 3 δευτερολέπτων πατήσουμε το STOP η αντλία 2 σταματά.

### Δίκτυο 3: Ενεργοποίηση αναδευτήρα

Network 3 : MIXER ACTIVATION

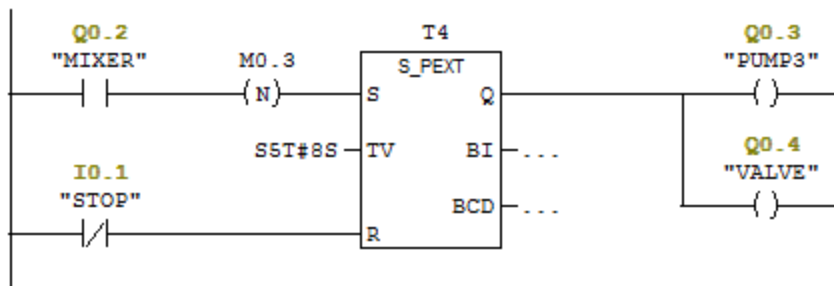


Σχήμα 4.18: Δίκτυο 3 - Ενεργοποίηση MIXER

Με τον ίδιο τρόπο όταν απενεργοποιηθεί η αντλία 2 έχουμε αλλαγή σήματος. Με το M0.2 ανιχνεύουμε την αλλαγή ώστε να ενεργοποιηθεί η έξοδος MIXER, που είναι ο αναδευτήρας της δεξαμενής. Ο αναδευτήρας ενεργοποιείται για 10 δευτερόλεπτα και σταματά.

### Δίκτυο 4: Ενεργοποίηση αντλίας 3 και βαλβίδας διοχέτευσης

Network 4 : PUMP 3 AND DRAIN VALVE ACTIVATION

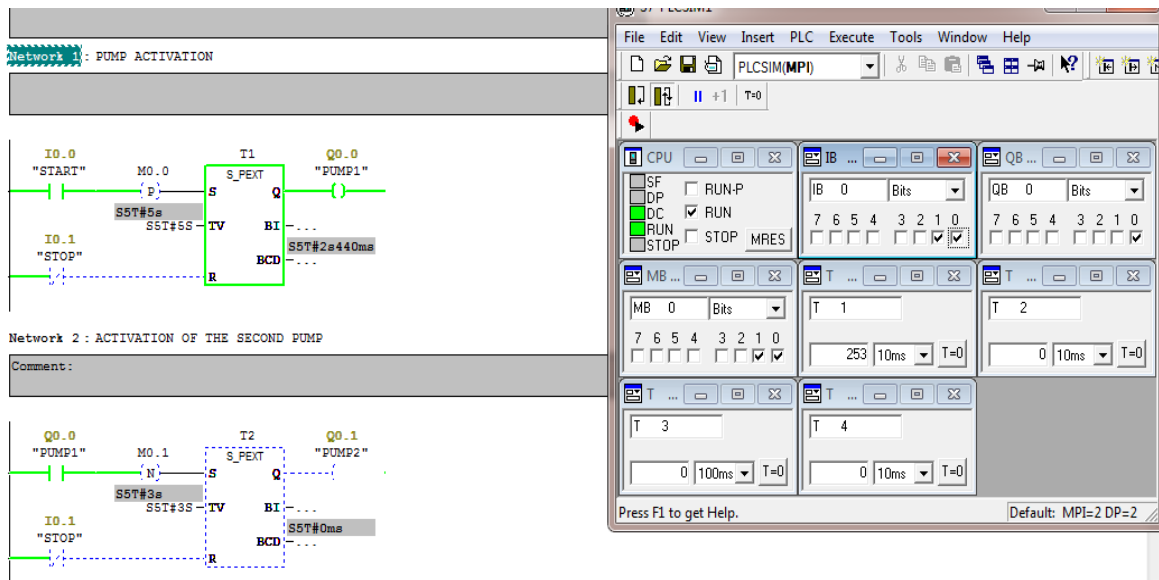


Σχήμα 4.19: Δίκτυο 4 - Ενεργοποίηση PUMP 3 και VALVE

Όταν απενεργοποιηθεί το MIXER έχουμε αλλαγή σήματος. Με το M0.3 ανιχνεύεται η αλλαγή για να ενεργοποιηθούν τα PUMP 3 και VALVE. Αυτές ενεργοποιούνται για 8 δευτερόλεπτα μέχρι να σταματήσουν. Αν θέλουμε, μπορούμε να τις σταματήσουμε πιο πριν.

Η προσωμοίωση του παραπάνω προγράμματος, γίνεται ακριβώς με τον ίδιο τρόπο που αναφέραμε στα προηγούμενα παραδείγματα. Αυτό που παρατηρούμε

είναι πως εδώ έχουμε μια αλυσιδωτή εξάρτηση δηλαδή με το που τελειώσει η λειτουργία στο πρώτο δίκτυο, έχουμε έναρξη λειτουργίας στο δεύτερο μέχρι να φτάσουμε στο τελευταίο που είναι η έξοδος του υγρού από την δεξαμενή. Στην ουσία, εδώ θέλουμε να εισάγονται υγρά από τις ανλίες 1 και 2, μόλις ο χρόνος που απαιτείται για το γέμισμα περάσει να αρχίζουν τα υγρά να ανακατεύονται, και όταν ολοκληρωθεί η διαδικασία ανάδευσης να απελευθερώνονται από την τρίτη ανλία. Εδώ η είσοδος γίνεται χειροκίνητα από εμάς ενώ μετά το πέρας του χρόνου οι εισοδοί στα υπόλοιπα κομμάτια γίνεται αυτόματα. Ωστόσο, αν θέλουμε να έχουμε έξοδο νωρίτερα από το προαπαιτούμενο, μπορούμε να το κάνουμε εξαναγκασμένα μέσω των Bit εξόδου που θέσαμε (Q0.1, Q0.2 κλπ).



**Εικόνα 4.7:** Παραγωγή αποτελεσμάτων για την δεξαμενή – Η ανλία 1 παρέχει υγρό

#### 4.5. Παράδειγμα 5: Κυλιόμενη Σκάλα

Θέλουμε να δημιουργήσουμε ένα σύστημα αυτόματου ελέγχου για μια κλασική κυλιόμενη σκάλα. Η σκάλα θα λειτουργεί με το πάτημα ενός κουμπιού, ενσωματωμένου σε ειδικό χειριστήριο. Το ξεκίνημά της δε θα γίνεται αμέσως αλλά με το που εμφανιστεί ένα άτομο στην είσοδο και θα διακόπτεται με την έξοδο του ατόμου αυτού. Με την έξοδο, η σκάλα θα συνεχίζει να λειτουργεί για άλλα 15 δευτερόλεπτα.

Η διακοπή λειτουργίας θα γίνεται με τους γνωστούς τρόπους, δηλαδή με το πάτημα του κουμπιού Stop ή και μιας εντολής που θα δίνεται από το χειριστήριο αλλά και με άλλα δύο κουμπιά έκτακτης ανάγκης.

Τα σύμβολα που θα χρησιμοποιήσουμε είναι:

Status	Symbol	Address	Data Type	Comment
	Cycle execution	OB 1	OB 1	
	MOTOR RELAY	Q 0.0	BOOL	
	PHOTOCELL	I 0.5	BOOL	
	START	I 0.1	BOOL	
	STOP	I 0.2	BOOL	
	STOP1 EMERGENCY	I 0.3	BOOL	
	STOP2 EMERGENCY	I 0.4	BOOL	
	SYSTEM ON	Q 0.1	BOOL	
	THERMIKO	I 0.0	BOOL	

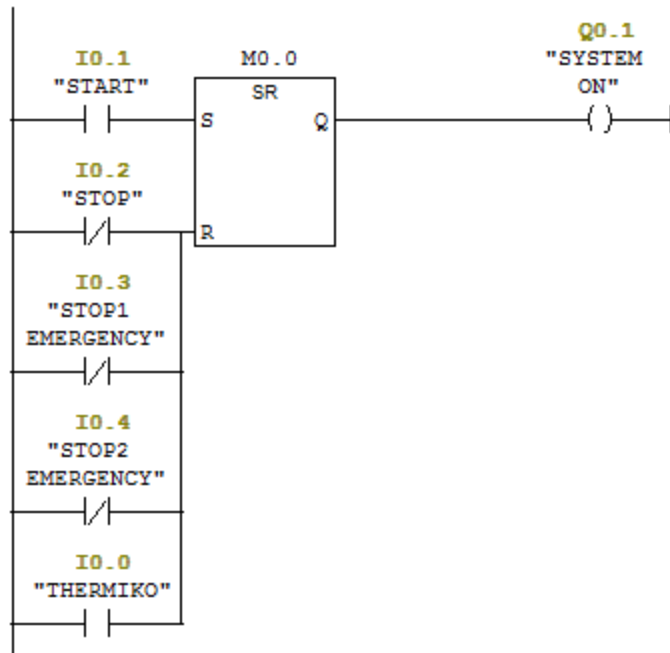
Πίνακας 4.3: Σύμβολα κυλιόμενης σκάλας

### Δίκτυο 1: Ενεργοποίηση εγκατάστασης

OB1 : ESCALATOR



Network 1 :

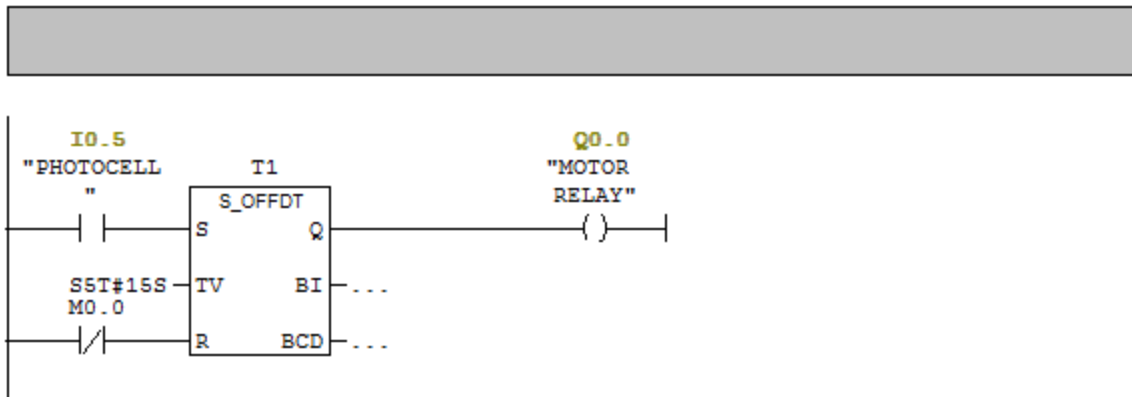


Σχήμα 4.20: Δίκτυο 1 - Ενεργοποίηση σκάλας

Όταν πατάμε το κουμπί εκκίνησης, δηλαδή την είσοδο I0.1, το βοηθητικό M0.0 γίνεται λογικό 1 και ενεργοποιείται η λυχνία SYSTEM ON. Η εγκατάσταση ενεργοποιείται και παραμένει σε αυτή την κατάσταση μέχρι να πατηθεί το κουμπί διακοπής, είτε τα κουμπια έκτακτης ανάγκης STOP1/STOP2 Emergency, είτε αν το θερμικό κλείσει την επαφή. Σε μια από τις περιπτώσεις αυτές, η λυχνία απενεργοποιείται.

## Δίκτυο 2: Λειτουργία 15 δευτερολέπτων

Network 2 :



Σχήμα 4.21: Δίκτυο 2 – Λειτουργία μετά από την έξοδο

Αν το άτομο που ετοιμάζεται να ανέβει, σταματήσει μπροστά από το φωτοκύτταρο εκκίνησης η σκάλα θα πρέπει να παραμένει ενεργή αλλά να μην μετράει τα 15 δευτερόλεπτα. Η αντίστροφη μέτρηση θα γίνεται μετά το πέρασ από την έξοδο. Άρα το χρονικό που χρειαζόμαστε θα αρχίσει να μετράει αφού ανιχνευθεί αρνητική ακμή του παλμού. Το χρονικό που θα χρησιμοποιηθεί εδώ είναι το S\_OFFDT.

Όταν το άτομο διακόψει την δέσμη του φωτοκυτάρου, δηλαδή το σήμα στην είσοδο του T1 είναι λογικό 1, τότε η σκάλα ξεκινάει. Όταν το σήμα στην είσοδο γίνεται λογικό 0, τότε το χρονικό ξεκινάει να μετράει χρόνο 15 δευτερολέπτων και μετά σταματάει την λειτουργία του. Αν το M0.0 γίνει και αυτό λογικό 0 τότε απενεργοποιείται το T1 και η κυλιόμενη σκάλα σταματάει αμέσως.

Η λειτουργία της σκάλας είναι από τις πιο απλές στον αυτοματισμό. Με λίγα λόγια, σε αυτό το παράδειγμα βλέπουμε πως η κυλιόμενη θα δέχεται έξι εισόδους με μόνο δύο από αυτές να θέτουν σε λειτουργία τον μηχανισμό. Αυτό διότι οι υπόλοιπες εισοδοι γίνονται με ανθρώπινη παρέμβαση από το χειριστήριο. Τα φωτοκύτταρα λαμβάνουν διαφορετικές λειτουργίες και παραμέτρους για τον λόγο ότι το



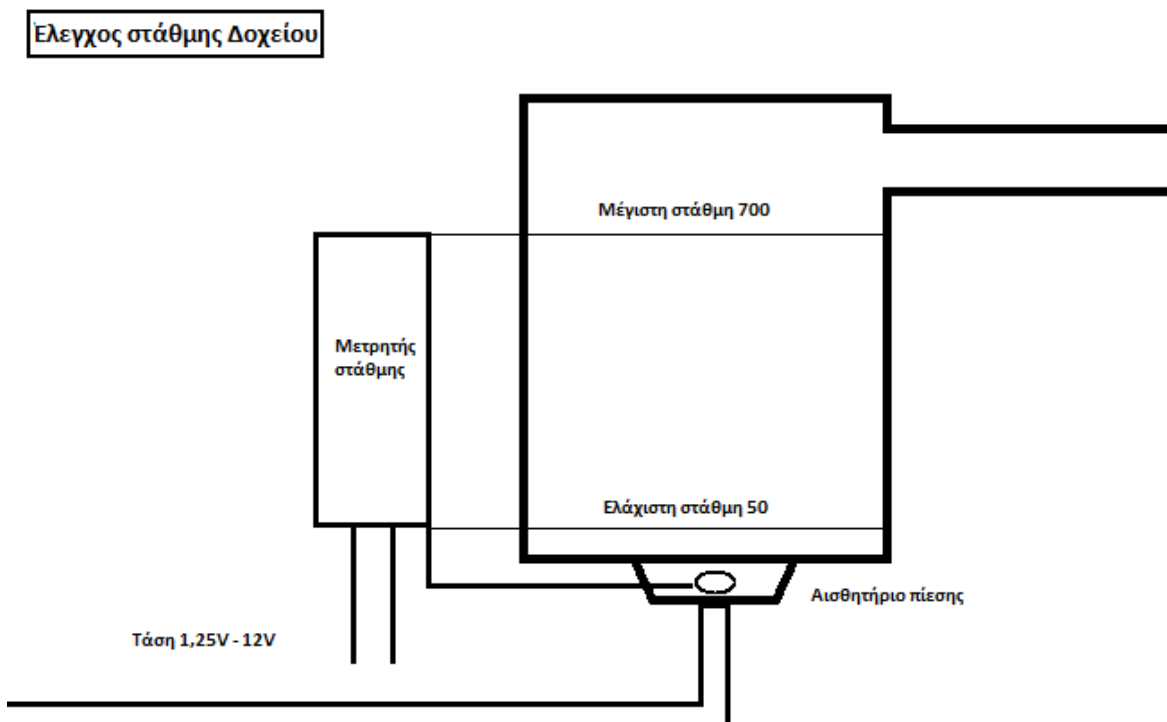
φωτοκύπαρο εισόδου θα πρέπει να ενεργοποιεί την σκάλα και να την κρατάει ενεργοποιημένη ενώ το φωτοκύπαρο εξόδου ελέγχου ανθρώπινη διέλευση και κρατά την σκάλα ενεργή για άλλα 15 δευτερόλεπτα μέχρι να σταματήσει σταδιακά.

#### 4.6. Παράδειγμα 6: Στάθμη Δοχείου

Στην δεξαμενή αναφερθήκαμε στο γέμισμα δοχείου. Σε μερικές δεξαμενές ο αυτοματισμός θα περιέχει και φωτοκύπαρα που θα επιτηρούν το γέμισμα της δεξαμενής και θα εκκινούν την λειτουργία του αναδευτήρα όταν το υγρό φτάσει σε ορισμένο ύψος.

Έστω ότι έχουμε μια δεξαμενή με μέγιστη χωρητικότητα 700 λίτρων. Η στάθμη της μετριέται με την βοήθεια ενός μετρητή, που διαθέτει έναν αισθητήριο στον πυθμένα της δεξαμενής και υπολογίζει την πίεση που ασκεί το υγρό σε αυτόν. Ο μετρητής μετατρέπει την εκάστοτε τρέχουσα πιμή της πίεσης σε μια τάση μεταξύ 1,25 και 12V (μέγιστη στάθμη).

Αυτό που θέλουμε να γίνεται σε αυτό το παράδειγμα είναι να γίνονται διαφορετικές λειτουργίες κάθε φορά που η δεξαμενή γεμίζει με ορισμένα λίτρα υγρού.



**Σχήμα 4.22:** Δοχείο με όλα τα απαραίτητα στοιχεία

Όταν η δεξαμενή περιέχει κάτω από 100 λίτρα υγρού, τότε θα πρέπει να αναβοσβήνουν όλες οι λυχνίες. Μετά από τα 100 λίτρα οι λυχνίες θα ανάβουν με διαφορετική σειρά:

- Από 100-199 λίτρα θα ανάβει η πρώτη λυχνία
- Από 200-299 λίτρα θα ανάβει η πρώτη και η δεύτερη λυχνία
- Από 300-399 λίτρα τότε θα πρέπει να ανάβει μέχρι και η τρίτη λυχνία
- Από 400-499 θα ανάβει μέχρι και η τέταρτη λυχνία
- Από 500-599 θα ανάβει μέχρι και η Πέμπτη λυχνία
- Από 600-700 θα ανάβουν όλες οι λυχνίες

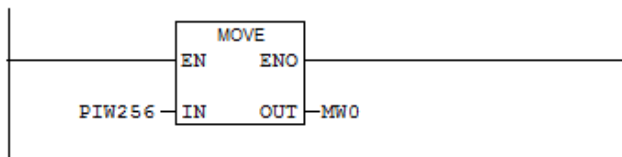
Τα δίκτυά μας θα ανέρχονται στα 14 για την σύγκριση των λίτρων αλλά και για τον έλεγχο κάθε λυχνίας ξεχωριστά. Αναφορικά θα αναφερθούμε στα κυριότερα κομμάτια του προγράμματος ενώ στα υπόλοιπα θα γίνει αναφορά της λειτουργίας τους.

**Δίκτυο 1:** Η είσοδος PIW256 της αναλογικής κάρτας εισόδου μεταφέρεται συνεχώς στο MW0.

OB1 : TANK LEVEL



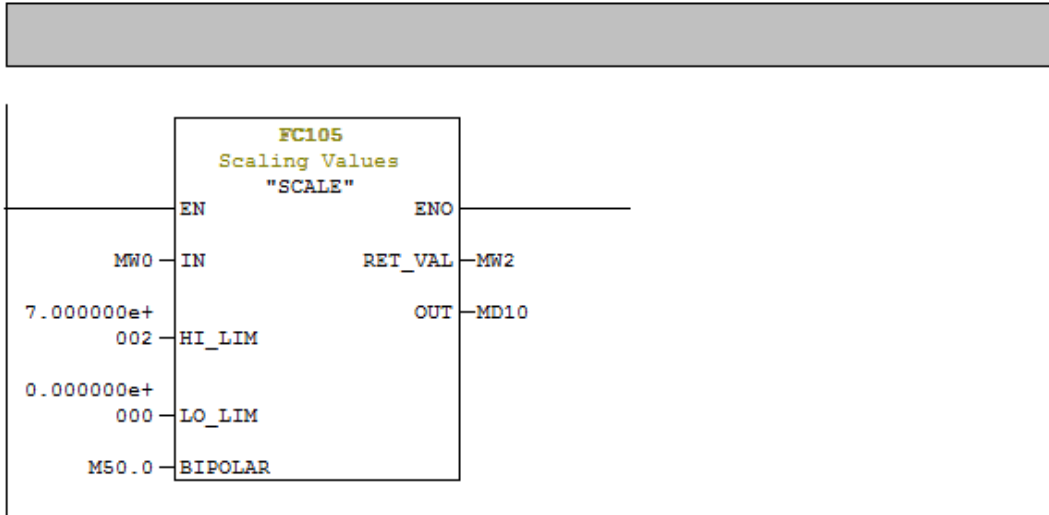
Network 1 :



**Σχήμα 4.23:** Δίκτυο 1 – Είσοδος αναλογικής κάρτας

**Δίκτυο 2:** Κλιμακοποίηση αναλογικών πιμών με το μπλοκ FC 105.

Network 2 :

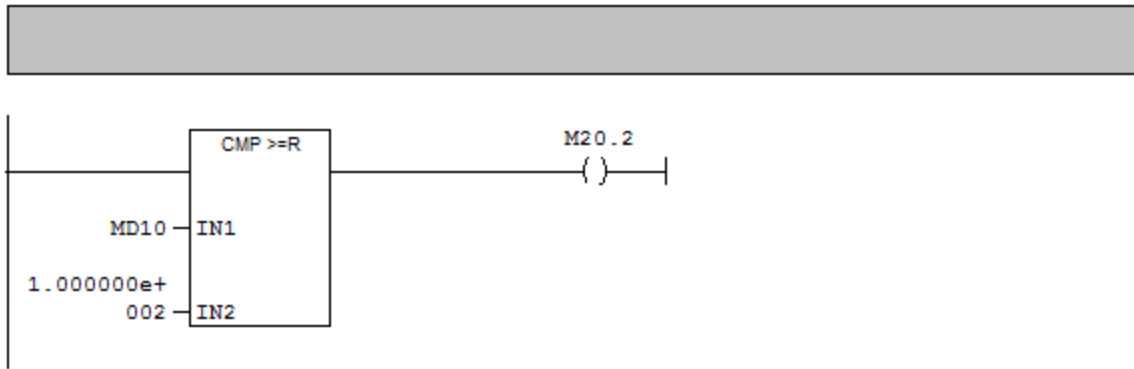


**Σχήμα 4.24:** Δίκτυο 2 – Η συνάρτηση αναλογικής κλιμάκωσης

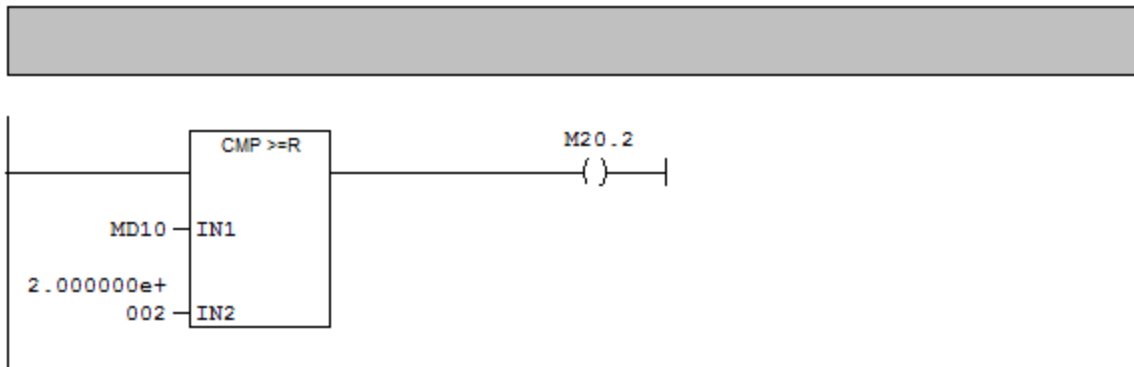
Η κλήση του μπλοκ συναρτήσεων FC 105 μας επιτρέπει την μετατροπή των αναλογικών πιμών σε μορφή ακεραίων. Η τιμή στην είσοδο IN θα διαβαστεί σε INTEGER μορφή. Στο παραπάνω κύκλωμα, τα HI\_LIM και LO\_LIM χρησιμοποιούνται στην οριοθέτηση για την μετατροπή των φυσικών μεγεθών. Η τιμή εδώ κυμαίνεται από 0 έως 700. Η είσοδος BIPOLAR καθορίζει αν πρέπει να μετατρέπονται και οι αρνητικοί αριθμοί. Το M50.0 έχει τιμή λογικό 1, δηλαδή UNIPOLAR. Τέλος, η RET\_VAL έχει τιμή 0 αν η εκτέλεση γίνεται χωρίς σφάλματα.

Παρακάτω, βλέπουμε το πρώτο από τα τρία κομμάτια των δικτύων σύγκρισης. Η διαφορά με τα υπόλοιπα είναι πως η τιμή στο IN2 και η έξοδος M20.x αλλάζουν τιμές κατά ένα (δηλαδή αν η IN2 του πρώτου δικτύου είναι 1.000000e+002 τότε στο επόμενο δίκτυο θα είναι 2.000000e+002):

Network 3 :



Network 4 :

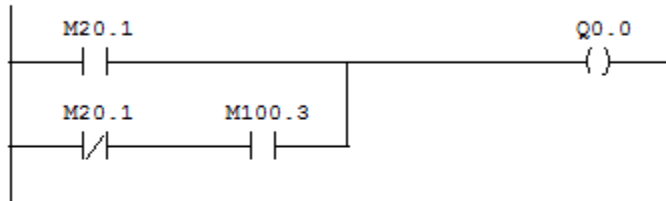


Σχήμα 4.25: Δίκτυα 3 και 4 – Συγκρίσεις λίτρων

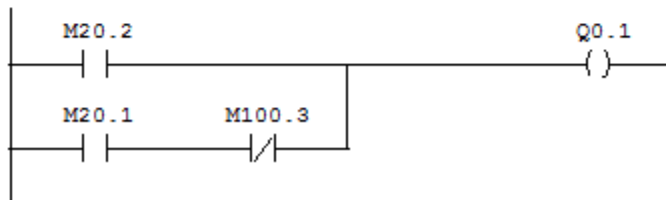
- **Δίκτυο 3:** Αν η πμμή που περιέχει το MD10 είναι μεγαλύτερη ή ίση με το 100, τότε το M20.1 γίνεται λογικό 1 (IN2: 1.000000e+002)
- **Δίκτυο 4:** Αν η πμμή που περιέχει το MD10 είναι μεγαλύτερη ή ίση με το 200, τότε το M20.2 γίνεται λογικό 1 (IN2: 2.000000e+002)
- **Δίκτυο 5:** Αν η πμμή που περιέχει το MD10 είναι μεγαλύτερη ή ίση με το 300, τότε το M20.3 γίνεται λογικό 1 (IN2: 3.000000e+002)
- **Δίκτυο 6:** Αν η πμμή που περιέχει το MD10 είναι μεγαλύτερη ή ίση με το 400, τότε το M20.4 γίνεται λογικό 1 (IN2: 4.000000e+002)
- **Δίκτυο 7:** Αν η πμμή που περιέχει το MD10 είναι μεγαλύτερη ή ίση με το 500, τότε το M20.5 γίνεται λογικό 1 (IN2: 5.000000e+002)
- **Δίκτυο 8:** Αν η πμμή που περιέχει το MD10 είναι μεγαλύτερη ή ίση με το 600, τότε το M20.6 γίνεται λογικό 1 (IN2: 6.000000e+002)

### Δίκτυα 9-14: Ενεργοποίηση λυχνίων

Network 9 :



Network 10 :



Σχήμα 4.26: Δίκτυα 3 και 4 – Συγκρίσεις λίτρων

- **Δίκτυο 9:** Αν το M20.1 είναι λογικό 1, τότε ενεργοποιείται η έξοδος Q0.0. Αν το M20.1 είναι λογικό 0, τότε με την βοήθεια του M100.3, ενεργοποιείται και απενεργοποιείται η έξοδος Q0.0.
- **Δίκτυο 10:** Αν το M20.2 είναι λογικό 1, τότε ενεργοποιείται η έξοδος Q0.1. Αν το M20.1 είναι λογικό 0, τότε με την βοήθεια του M100.3, ενεργοποιείται και απενεργοποιείται η έξοδος Q0.1.
- **Δίκτυο 11:** Αν το M20.3 είναι λογικό 1, τότε ενεργοποιείται η έξοδος Q0.2. Αν το M20.3 είναι λογικό 0, τότε με την βοήθεια του M100.3, ενεργοποιείται και απενεργοποιείται η έξοδος Q0.2.
- **Δίκτυο 12:** Αν το M20.4 είναι λογικό 1, τότε ενεργοποιείται η έξοδος Q0.3. Αν το M20.4 είναι λογικό 0, τότε με την βοήθεια του M100.3, ενεργοποιείται και απενεργοποιείται η έξοδος Q0.3.
- **Δίκτυο 13:** Αν το M20.5 είναι λογικό 1, τότε ενεργοποιείται η έξοδος Q0.4. Αν το M20.5 είναι λογικό 0, τότε με την βοήθεια του M100.3, ενεργοποιείται και απενεργοποιείται η έξοδος Q0.4.

- **Δίκτυο 14:** Αν το M20.6 είναι λογικό 1, τότε ενεργοποιείται η έξοδος Q0.5. Αν το M20.6 είναι λογικό 0, τότε με την βοήθεια του M100.3, ενεργοποιείται και απενεργοποιείται η έξοδος Q0.5.

Το βοηθητικό Byte MB100 χρησιμοποιείται για την ενεργοποίηση της λυχνίας (flashing). Επίσης, έχουμε στο μυαλό μας πως κάθε φορά που έχουμε λογικό 1 σε έξοδο (ή έξοδο) κι έχουμε ενεργοποιημένο το Debug Mode στο Simatic Manager θα δούμε πως το πράσινο χρώμα δηλώνει την ροή. Δηλαδή βλέπουμε πως σε όλα τα δίκτυα υπάρχει κλειστή είσοδος. Αυτό σημαίνει πως όταν το M20.x είναι λογικό 1 (άρα πράσινο) τότε θα λειτουργεί μόνο η αντίστοιχη Q0.x (αντίστοιχα πράσινη).

Όταν η κλειστή είσοδος είναι ενεργή τότε οι λυχνίες αναβασβήνουν. Αυτό το κάνουμε ώστε αν η στάθμη του δοχείου είναι κάτω από 100, να αναβασβήνουν όλες οι λυχνίες.

## ΚΕΦΑΛΑΙΟ 5:

### ΣΥΜΠΕΡΑΣΜΑΤΑ - ΠΡΟΟΠΤΙΚΕΣ

Το λογισμικό της Siemens, Simatic Manager, αποτελεί ένα πολύ καλό εργαλείο, ειδικά για όσους ασχολούνται με τον τομέα του αυτοματισμού. Εκτός από την βιομηχανία, μπορούμε να το χρησιμοποιήσουμε και για εκπαιδευτικούς σκοπούς μιας και ο προγραμματισμός των λογικών ελεγκτών δεν περιορίζεται σε βιομηχανική χρήση αλλά και σε οικιακή. Το υλικό της μελέτης μας χρησιμεύει ακόμα και για εκπαιδευτικούς σκοπούς.

Από την μελέτη μας, είδαμε πως πρόκειται για ένα εύχρηστο λογισμικό με απλό τρόπο εγκατάστασης στους προσωπικούς μας υπολογιστές. Υπάρχουν δύο κύριες εκδόσεις, η επαγγελματική και η Lite με την διαφορά να δίνεται στο πλήθος των διαθέσιμων λειτουργιών. Η έκδοση που χρησιμοποιήσαμε είναι η επαγγελματική μιας και μας προσφέρει περισσότερες εναλλακτικές λύσεις για τον προγραμματισμό ελεγκτών.

Ως απλός χρήστης που έρχεται σε επαφή με το παρόν λογισμικό για πρώτη φορά, οι εντυπώσεις είναι αρκετά θετικές. Η εγκατάσταση στον φορητό υπολογιστή που χρησιμοποιήθηκε για τις προσωμοιώσεις, ήταν γρήγορη και κλασσική με αυτό να σημαίνει πως δεν αντιμετωπίστηκαν προβλήματα κατά την διαδικασία. Όλα τα απαραίτητα στοιχεία προστέθηκαν και η παραμετροποίησή τους ολοκληρώθηκε άμεσα – εκτός από το PLCSim στο οποίο χρειάστηκε να προσθέσουμε τον τρόπο επικοινωνίας του εικονικού λογικού ελεγκτή με τον υπολογιστή.

Μέσα σε αυτόν τον οδηγό, καλύφθηκε το μεγαλύτερο κομμάτι του προγράμματος το οποίο είναι και το βασικότερο προκειμένου να κάνουμε την αρχή αλλά και να εξοικειωθούμε με το περιβάλλον εργασίας του. Οι οδηγίες εγκατάστασης καθώς και οι βασικότερες λειτουργίες, είναι όλα όσα θα χρειαστεί κανείς για να ξεκινήσει τον προγραμματισμό λογικών ελεγκτών με το Simatic Step 7.

Διάφορα προβλήματα που μπορεί να μας δυσκολεύσουν στην αρχή είναι ο περιορισμός του λειτουργικού συστήματος που απαιτείται. Οι περισσότεροι οικιακοί υπολογιστές χρησιμοποιούν πλέον Windows 8/8.1/10 τα οποία δεν υποστηρίζονται. Ο ίδιος περιορισμός όμως, δεν ισχύει ακόμα για τις επιχειρήσεις μιας και η χρήση των Windows XP/7 είναι ακόμα διαδεδομένη. Επιπλέον, οι

φορητοί υπολογιστές παραμένουν η λιγότερο προτεινόμενη λύση λόγω των προβλημάτων που προκύπτουν με την επικοινωνία στο PLCsim (η λύση συνήθως είναι να συνδέουμε τον προσωμοιωτή με την κάρτα δικτύου). Το κυριότερο από αυτά είναι η αδυναμία ρύθμισης της επικοινωνίας λόγω έλλειψης των απαραίτητων εξαρτημάτων από τον υπολογιστή (κάρτα MPI).

### **5.1. Πλεονεκτήματα και μειονεκτήματα χρήσης**

Πέρα από την εύκολη εγκατάσταση και εκμάθηση του λογισμικού Step 7, υπάρχουν αρκετοί λόγοι οι οποίοι θα μας ωθήσουν στην προτίμηση του. Οι αυτοματισμοί με τους οποίους ερχόμαστε σε επαφή καθημερινά είναι εκατοντάδες. Για τον κάθε έναν χρησιμοποιούνται διαφορετικά εξαρτήματα μέσα στα οποία βρίσκονται και οι επεξεργαστές. Το Simatic Manager μας δίνει την δυνατότητα να επιλέξουμε ανάμεσα σε πολλούς και διάφορους από αυτούς ανάλογα με τις απαιτήσεις του προγράμματός μας όπως τον αριθμό εισόδων και εξόδων, την τάση του ρεύματος κλπ.

Η διεπαφή του χρήστη είναι αρκετά απλή δίνοντάς μας όλα όσα θα χρειαστούμε για τον ορθό προγραμματισμό ενός ελεγκτή (εικονικού ή μη) καθώς και πρόσβαση σε όλα τα επιμέρους προγράμματα που ενδέχεται να χρησιμοποιήσουμε.

Η χρήση του σε πόρους συστήματος είναι αρκετά μικρή οπότε δεν θα συναντήσουμε πολλά προβλήματα όσων αφορά την απόδοση στο σύστημά μας. Μπορεί να έχουμε έναν παλιό υπολογιστή όμως θα μπορούμε να προσωμοιώσουμε χωρίς ιδιαίτερα θέματα.

Κατά την συγγραφή ενός προγράμματος, υπάρχουν προεγκατεστημένες βιβλιοθήκες με διάφορα στοιχεία που μπορεί να μας χρειαστούν σε μια προσωμοίωση. Για παράδειγμα, είδαμε στο κεφάλαιο 4, στο παράδειγμα 6 πως καλούμε την συνάρτηση FC 105, απαραίτητη για την μετατροπή σε ακεραίους. Η κλήση της έγινε μέσω της αντίστοιχης βιβλιοθήκης που δίνεται στα αριστερά του παραθύρου συγγραφής. Τα στοιχεία που εισάγουμε, θέτονται πληκτρολογώντας το αντίστοιχο όνομα (π.χ. S\_PEXT) στο διάστημα των τριών ερωτηματικών (???).

Από την άλλη, θα συναντήσουμε ορισμένες περιπτώσεις στις οποίες θα δαπανηθεί χρόνος για την επίλυση προβλημάτων. Ένα από τα κυριότερα προβλήματα που συναντήθηκαν ήταν η αδυναμία προσωμοίωσης λόγω προβλήματος με τις κάρτες επικοινωνίας.



Για κάποιον που θέλει να χρησιμοποιήσει το Simatic Manager στην ελληνική γλώσσα ή απλά να γράψει στα σχόλια και τους τίτλους στα ελληνικά, τότε θα πρέπει να πραγματοποιήσουμε μια χρονοβόρα διαδικασία. Η διαδικασία αυτή απαιτεί μια βιβλιοθήκη ελληνικής μετάφρασης ώστε να μπορέσουμε να την εισάγουμε στο λογισμικό.

Αν θέλουμε να αποθηκεύσουμε το έργο μας σε PDF, θα πρέπει να το κάνουμε μέσω της λειτουργίας εκτύπωσης. Πιο απλά, θα πρέπει να μπούμε μέσα στο γραμμένο κώδικα και να επιλέξουμε “εκτύπωση”. Από εκεί θα πρέπει να κάνουμε την επιλογή “εκτύπωση σε αρχείο” ώστε να μπορέσουμε να το αποθηκεύσουμε κατάλληλα.

Τέλος, κάτι που μπορεί να θεωρηθεί θετικό αλλά και αρνητικό παράλληλα, είναι η δυνατότητα χρήσης εξωτερικών προγραμμάτων για την οπτική προσωμοίωση ενός κώδικα. Αν θέλουμε να δούμε την λειτουργία της δεξαμενής του παραδείγματος 4, τότε θα χρειαστούμε το WinCC Flexible που είναι ένα ακόμα πρόγραμμα της Siemens. Δίνεται η δυνατότητα να το συνδέσουμε με το Simatic Manager όμως αποτελεί μια περίπλοκη διαδικασία. Για να μπορέσουμε να προσωμοιώσουμε το πρόγραμμα, θα πρέπει να φτάσουμε τόσα στοιχεία όσα χρησιμοποιούνται στα σύμβολα που ορίζονται ενώ θα πρέπει να υπάρχει επικοινωνία και με το PLCsim.

## 5.2. Προοπτικές

Το Simatic Manager είναι το κατάλληλο εργαλείο για την εκπαιδευτική διαδικασία αλλά και για προσωπική εξάσκηση αν θελήσουμε να ασχοληθούμε με τον προγραμματισμό και επεξεργασία των λογικών ελεγκτών. Η εξέλιξη του αυτοματισμού συμβαδίζει με την εξέλιξη της τεχνολογίας με αποτέλεσμα να αναμένονται νέα και καλύτερα κυκλώματα για την αξιοποίηση ιδεών. Μπορεί να χρησιμοποιηθεί όχι μόνο σε βιομηχανίες και εκπαιδευτικά ιδρύματα αλλά και σε οικίες, σε εφαρμογές όπως αυτή του θερμοστάτη ή του ελέγχου θυρών και παραθύρων (αυτόματο άνοιγμα και κλείσιμο)

Η εφαρμογή όλων των νέων ιδεών μπορεί να γίνει μέσω προγραμμάτων όπως το Simatic Step 7 ειδικά αν σκεφτεί κανείς το πλήθος των εξαρτημάτων που το συνοδεύουν. Η ύπαρξη των προσωμοιώσεων μας δίνει να καταλάβουμε την λειτουργία του πραγματικού αυτοματισμού. Έτσι αυτό που επιτυγχάνεται είναι η

ακριβής, γρήγορη και εύκολη δημιουργία νέων αυτοματισμών που θα μπορούν να χρησιμοποιηθούν για την διευκόλυνση της ζωής μας.

Η παρούσα εργασία αποτελεί έναν βασικό οδηγό για εκείνους που επιθυμούν να ξεκινήσουν την ασχολία του προγραμματισμού λογικών ελεγκτών. Ενώ δίνονται περιγραφές των βασικών λειτουργιών αλλά και του τρόπου εγκατάστασης δεν υπάρχει μια πιο επαγγελματική προσέγγιση. Τα θέματα που παραμένουν ανοιχτά στην παρούσα φάση είναι η επαγγελματική προσέγγιση και οι πιο ειδικές λειτουργίες του προγράμματος. Τέλος, λόγω της κατασκευής του Step 7 δεν υπάρχει ενσωματωμένη λειτουργία για την απεικόνιση των προσωμιώσεων ως γραφικά. Αν θέλουμε να ασχοληθούμε παραπάνω με το λογισμικό πρέπει να ανατρέξουμε σε συγκεκριμένη βιβλιογραφία καθώς και στους οδηγούς χρήσης που προσφέρονται στα αγγλικά από την εταιρεία Siemens.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] C.T. Jones, “STEP 7 Programming Made Easy in LAD, FBD and STL – A Practical Guide to Programming S7-300/S7-400 Programmable Logic Controllers”, Patrick-Turner Publishing, 2013
- [2] Edward W. Kamen, “Industrial Controls and Manufacturing”, Academic Press, *Ladder Logic Diagrams and PLC Implementations*, Chapter 8, 1999
- [3] Siemens, “SIMATIC: Programming with STEP 7”, Edition 03, 2006
- [4] W. Bolton, “Programmable Logic Controllers”, Fifth Edition, Newnes, 2009
- [5] Μ. Δροσάκης, “Προγραμματιζόμενοι Λογικοί Ελεγκτές”,  
([http://courseware.mech.ntua.gr/ml23194/extras/PLC\\_Lesson1.pdf](http://courseware.mech.ntua.gr/ml23194/extras/PLC_Lesson1.pdf), τελευταία πρόσβαση στις 27/12/2015)