



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ Ι

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΙ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΙΑΣ

Γνωστική: Καρβινάκης  
Βενέτας  
Νικολαΐδης

ΠΡΑΞΗ 02/04-02-201

H/Γ  
530



«ΔΙΑΧΕΙΡΙΣΗ ΓΡΑΜΜΗΣ ΠΑΡΑΓΩΓΗΣ ΧΗΜΙΚΟΥ  
ΑΝΤΙΔΡΑΣΤΗΡΑ ΜΕ ΤΗ ΒΟΗΘΕΙΑ PLC S7-300 &  
ΕΠΟΠΤΕΙΑ ΑΠΟ SCADA WinCC»

ΜΟΣΧΑΚΗΣ ΓΙΩΡΓΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΗΡ. ΣΩΤΗΡΑΚΗΣ

ΑΘΗΝΑ, ΙΑΝΟΥΑΡΙΟΣ 2011



Πίνακας Περιεχομένων		
<b>Μέρος 1<sup>ο</sup></b>	Θεωρητικό Υπόβαθρο	
	Εισαγωγή	4
<b>Κεφάλαιο 1<sup>ο</sup></b>	Προγραμματιζόμενοι λογικοί ελεγκτές	5
1.1	Ιστορική αναδρομή	6
1.2	Ορισμός και Πλεονεκτήματα	8
1.3	Υλικά για τον έλεγχο μιας εγκατάστασης μέσω P.L.C	10
<b>Κεφάλαιο 2<sup>ο</sup></b>	Προγραμματιζόμενος λογικός ελεγκτής Simatic s7-300	15
2.1	Δομή του προγραμματιζόμενου ελεγκτή	16
2.1.1	Υλικά	16
2.1.2	Συγκεντρωτική διαμόρφωση	17
2.1.3	Διευθυνσιοδότηση - Ονοματολογία	30
2.2	Δομή του Προγράμματος	36
2.2.1	Τύποι των Διαθέσιμων Μπλοκ	40
2.2.2	Δομή των Μπλόκ	45
<b>Κεφάλαιο 3<sup>ο</sup></b>	S7-300 Software	47
3.1	Simatic Manager	48
3.1.1	Περιβάλλον εργασίας	52
3.1.2	Είδη προγραμματισμού	59
<b>Κεφάλαιο 4<sup>ο</sup></b>	Μετατροπείς συχνότητας - Inverters	60
4.1	Γενικά περί των μετατροπέων συχνότητας	61

4.2	Μετατροπείς Ε.Ρ/Σ.Ρ/Ε.Ρ ή Ρυθμιστές Στροφών Ασύγχρονων κινητήρων	62
<b>Κεφάλαιο 5<sup>ο</sup></b>	<b>Επικοινωνίες</b>	<b>64</b>
5.1	Βασικές έννοιες	65
5.2	Δίκτυο	65
5.3	Υποδίκτυο	65
5.4	Simatic Communications	65
5.5	Είδη υποδικτύων	66
<b>Κεφάλαιο 6<sup>ο</sup></b>	<b>Scada WinCC</b>	<b>72</b>
6.1	Ιστορική αναδρομή	73
6.2	Γενικά περί συστημάτων τηλεδιαχείρισης	75
6.2.1	Το κόστος ενός συστήματος τηλεδιαχείρισης (SCADA)	78
6.3	Ορισμός	79
6.4	Scada WinCC V7.0 * Siemens	83
6.4.1	Περιβάλλον εργασίας / Δημιουργία ενός Project	86
<b>Μέρος 2<sup>ο</sup></b>	<b>Project</b>	<b>95</b>
<b>Μέρος 3<sup>ο</sup></b>	<b>Ηλεκτρολογικά Σχέδια Εφαρμογής</b>	<b>107</b>
<b>Μέρος 4<sup>ο</sup></b>	<b>PLC Software</b>	<b>130</b>
Πηγές		152



# ΜΕΡΟΣ 1<sup>ο</sup>

## Θεωρητικό Υπόβαθρο



## Εισαγωγή

Το project ασχολείται με τη διαχείριση του αυτοματισμού ενός χημικού αντιδραστήρα, μέσα στον οποίο, λαμβάνει χώρα μια εξώθερμη χημική αντίδραση.

Τον αυτοματισμό έχει αναλάβει ένα PLC S7-300 της Siemens, ενώ στο control room θα υπάρχει ένας Η/Υ, όπου θα τρέχει ένα πρόγραμμα απεικόνισης Scada της Siemens, που ονομάζεται WinCC v7.0.

Σε επιμέρους τμήματα του αυτοματισμού χρειάστηκε να χρησιμοποιηθούν Inverters για τον έλεγχο των στροφών κάποιων κινητήρων, καθώς επίσης και αναλογικές βαλβίδες οι οποίες μπορούν να ανοίγουν ποσοστιαία, για καλύτερη μίξη του νερού που ψύχει τον αντιδραστήρα, εξωτερικά.

Το plc, τα inverters καθώς & το πρόγραμμα απεικόνισης Scada, επικοινωνούν μεταξύ τους μέσω ενός πρωτόκολλου επικοινωνίας, γνωστό ως ProfibusDp.

Στο 1<sup>ο</sup> μέρος της πτυχιακής εργασίας αναλύεται σε θεωρητικό επίπεδο , ο προγραμματιζόμενος ελεγκτής, οι μετατροπείς συχνότητας , και το πρόγραμμα απεικόνισης του εργοστασίου Scada.

Το 2<sup>ο</sup> μέρος αναλύει το Project σε πρακτικό επίπεδο , λειτουργίες , χειρισμοί κλπ, ενώ στο τρίτο και τελευταίο μέρος βρίσκεται το software που υλοποιήθηκε στον προγραμματιζόμενο ελεγκτή για τον έλεγχο της ηλεκτρομηχανολογικής εγκατάστασης που πραγματοποιήθηκε.



# ΚΕΦΑΛΑΙΟ 1<sup>ο</sup>



## Προγραμματιζόμενοι Λογικοί Ελεγκτές

## 1.1 Ιστορική αναδρομή

Τα συστήματα αυτομάτου ελέγχου είχαν αποτελέσει αντικείμενο μελετών του ανθρώπου από αρχαιωτάτων χρόνων. Ένας από τους πλέον γνωστούς αυτοματισμούς της εποχής εκείνης είναι και ο ρυθμιστής του Ήρωνος του Αλεξάνδρου.

Σκοπός του αυτοματισμού ήταν να ανοίγει και να κλείνει αυτόματα την πύλη ενός ναού συγχρόνως με το άναμμα και το σβήσιμο της φωτιάς αντίστοιχα. Μέχρι περίπου τα τέλη του 18<sup>ου</sup> αιώνα, η τεχνολογία δεν είχε να επιδείξει κάτι αξιόλογο στον τομέα των αυτοματισμών παρά μόνο το 1769 όταν ο James Watt κατασκεύασε τον πρώτο φυγοκεντρικό ρυθμιστή ταχύτητας, ο οποίος χρησιμοποιήθηκε για τον αυτόματο έλεγχο της ταχύτητας της ατμομηχανής.

Μετά τον 2<sup>ο</sup> παγκόσμιο πόλεμο αρχίζει η ηλεκτρονική εποχή. Έχουμε τις πρώτες ηλεκτρονικές συσκευές, το ραδιόφωνο, αργότερα την τηλεόραση, τους ασυρμάτους και τα ραντάρ. Το κύριο εξάρτημα των συσκευών αυτών ήταν η λυχνία.

Η κλασική θεωρία των ΣΑΕ όπως την γνωρίζουμε σήμερα άρχισε να αναπτύσσεται στη δεκαετία του 1950. Ο αυτοματισμός γνώρισε καινούρια άνθιση τις δύο τελευταίες δεκαετίες με την ανάπτυξη των ηλεκτρονικών κυκλωμάτων και τη χρήση των ψηφιακών και αναλογικών υπολογιστών.

Σε αυτό το σημείο πρέπει να τονίσουμε πως οι υπολογιστές που κατασκευάζονταν πριν από μερικά χρόνια είχαν σαν μοναδικό στόχο την επίλυση ορισμένων επιστημονικών και λογιστικών προβλημάτων που ο χρόνος εξαγωγής των αποτελεσμάτων δεν ήταν και τόσο κρίσιμος.

Προκειμένου λοιπόν να γίνει δυνατή η προσαρμογή και χρησιμοποίησή τους στη βιομηχανία έπρεπε να υποστούν ορισμένες αναγκαίες τροποποιήσεις στα ηλεκτρονικά τους κυκλώματα ώστε να είναι σε θέση να δέχονται στην είσοδό τους, τα διάφορων τύπου ηλεκτρικά σήματα, να τα επεξεργάζονται και να στέλνουν το αποτέλεσμα στην έξοδο προκειμένου να ελέγξουν την βιομηχανική εγκατάσταση.



Τέτοιοι ειδικοί υπολογιστές κατάλληλοι για χρήση σε βιομηχανικούς αυτοματισμούς, είναι οι **Προγραμματιζόμενοι Λογικοί Ελεγκτές (PLC)**, οι οποίοι έχουν αντικαταστήσει σήμερα, κατά το μεγαλύτερο μέρος, τους αυτοματισμούς που λειτουργούν με συμβατική τεχνολογία (ρελέ, χρονικά, απαριθμητές, κτλ ).

Στην αρχή της εφαρμογής τους, τα PLC χρησιμοποιήθηκαν για την επίλυση ειδικών προβλημάτων αυτοματισμού, που είχαν σχέση με τα πλυντήρια αυτοκινήτων, τις αυτόματες εργαλειομηχανές & τα ηλεκτροπαραγωγά ζεύγη.

Με την εξέλιξη της τεχνικής στους μικροεπεξεργαστές, έχουμε και την ταυτόχρονη εξέλιξη των PLC, τα οποία χρησιμοποιούνται τώρα στον ευρύτερο και πολυσύνθετο χώρο της βιομηχανίας.

Τα PLC, μπορεί να θεωρηθεί ότι αποτελούν μια ειδική κατηγορία μικροεπεξεργαστών ή μικροϋπολογιστών, που έχουν τη δυνατότητα να δέχονται ή να παράγουν ηλεκτρικές τάσεις και ρεύματα όμοια με αυτά που χρησιμοποιούνται στη βιομηχανία. Εκτός αυτού, η σχεδίαση και κατασκευή αυτών των συσκευών έχει γίνει με τέτοιο τρόπο ώστε η συμπεριφορά τους, σε ώρα λειτουργίας, να παραμένει ανεπηρέαστη από τις διάφορες μεταβολές θερμοκρασίας ή υγρασίας.

Τελικά, το plc δεν είναι τίποτα άλλο παρά ένας μικροϋπολογιστής, κατάλληλα προσαρμοσμένος ώστε να χρησιμοποιείται για τη λειτουργία των αυτοματισμών. Δημιουργήθηκε για να αντικαταστήσει τον κλασικό πίνακα αυτοματισμού με τους ηλεκτρονόμους, τις βοηθητικές επαφές, τα χρονικά κλπ. Έγινε μια μεγάλη αλλαγή όπου από τους ηλεκτρονόμους περάσαμε στους υπολογιστές παρακάμπτοντας τα ηλεκτρονικά.

Για να μπορέσουν όμως οι τεχνικοί να χρησιμοποιήσουν το νέο προϊόν και να μην “τρομάξουν” από έννοιες όπως υπολογιστής, προγραμματισμός, γλώσσες προγραμματισμού κλπ, οι κατασκευάστριες εταιρείες δεν ανέφεραν τίποτα από τα





παραπάνω. Απλά τους είπαν πως, αντί να σχεδιάζουν το ηλεκτρολογικό σχέδιο, θα το κάνετε με το τρόπο που θα σας πούμε. Ουσιαστικά τους μάθαιναν προγραμματισμό.

Οι πρώτες γλώσσες προγραμματισμού δεν έκαναν τίποτα άλλο από το να αντιγράφουν το ηλεκτρολογικό σχέδιο σε μια ειδική συσκευή προγραμματισμού.

Σήμερα τα PLC έχουν εξελιχθεί πάρα πολύ. Υπάρχουν δε στην αγορά εκατοντάδες μοντέλα από πλήθος διαφορετικών εταιριών.

## 1.2 Ορισμός και Πλεονεκτήματα

### Ορισμός

Ο προγραμματιζόμενος λογικός ελεγκτής είναι ένα σύστημα αυτοματισμού που λειτουργεί με βάση την τεχνολογία των Η/Υ και είναι σε θέση

- να δέχεται διάφορα ηλεκτρικά σήματα ρεύματος ή τάσης (Inputs)
- να τα επεξεργάζεται
- να παράγει τα κατάλληλα σήματα εξόδου (Outputs), τα οποία θα ενεργοποιήσουν τις υπό έλεγχο διατάξεις.

### Πλεονεκτήματα

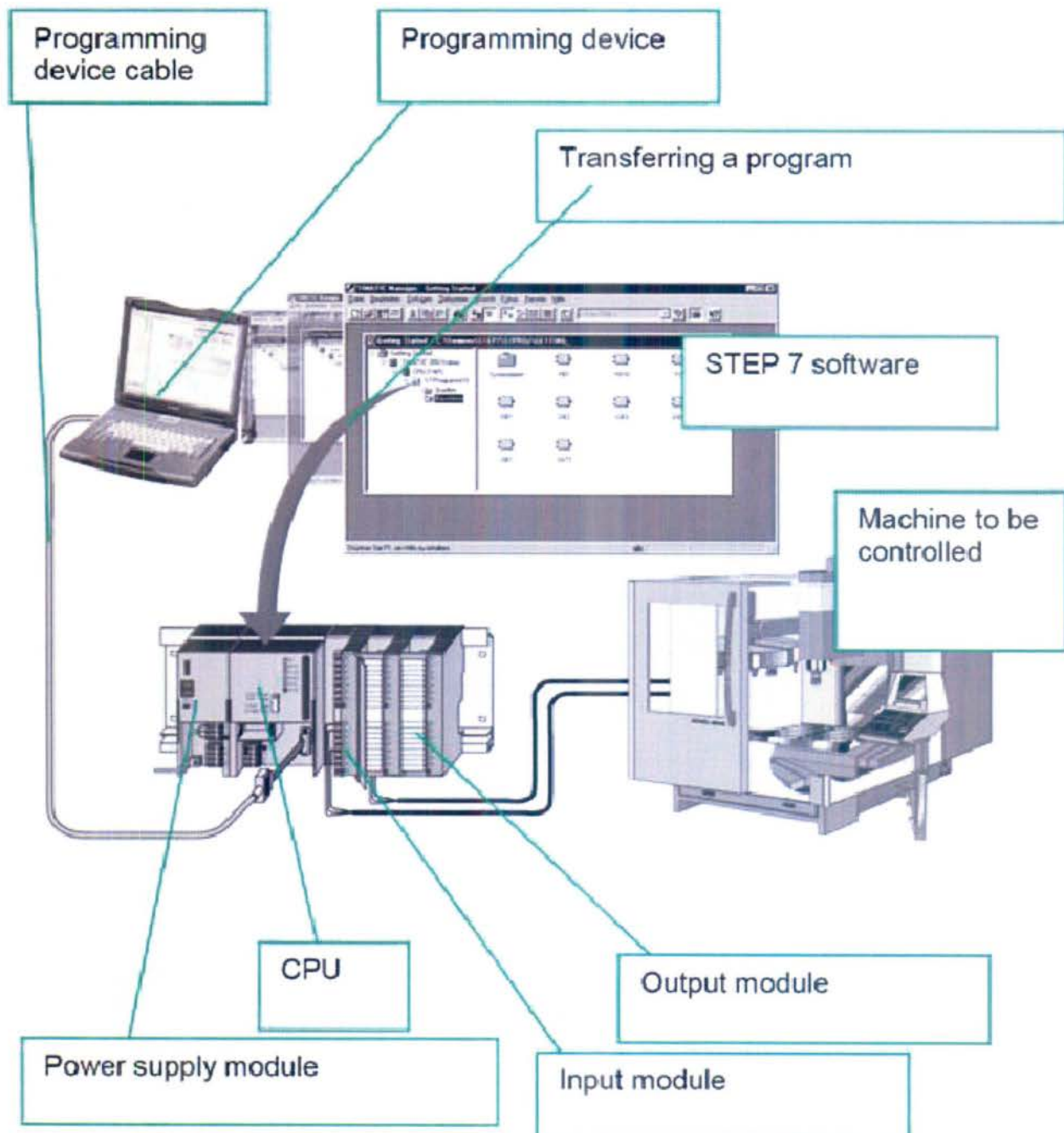
- ➔ Είναι συσκευές γενικής χρήσης (δεν είναι κατασκευασμένα για ένα συγκεκριμένο είδος εφαρμογής).
- ➔ Δεν ενδιαφέρει ο συνολικός αριθμός των επαφών, χρονικών, απαριθμητών (δεν είναι φυσικά στοιχεία, αλλά στοιχεία μνήμης).
- ➔ Ο χρόνος κατασκευής του αυτοματισμού είναι μηδαμινός σε σχέση με την κατασκευή ενός κλασικού πίνακα αυτοματισμού.



- ↳ Ελαχιστοποίηση του κόστους συντήρησης, εφόσον δεν υπάρχουν πλέον τόσο πολλές βλάβες, και αν υπάρξουν μπορούν να διαγνωστούν πολύ πιο εύκολα.
- ↳ Εύκολος οπτικός έλεγχος της λειτουργίας ή μη στοιχείων της εγκατάστασης με την βοήθεια των LED που υπάρχουν σε όλες τις κάρτες.
- ↳ Κάθε αλλαγή στο πρόγραμμα του χρήστη αποθηκεύεται στην μνήμη του PLC, έτσι ο τεχνικός δεν βρίσκεται προ απρόοπτου να διαβάσει ένα σχέδιο και άλλο να βρίσκεται πραγματικά στην εγκατάσταση.
- ↳ Τα PLC καταλαμβάνουν πολύ μικρό χώρο απ' ότι ένα αντίστοιχος πίνακας αυτοματισμού.
- ↳ Τα PLC είναι ευέλικτα στην τροποποίηση της λειτουργίας του αυτοματισμού που «τρέχουν», εφόσον οποιαδήποτε αλλαγή στη λειτουργία και τη ροή του αυτοματισμού, γίνεται μέσα σε λίγα λεπτά, αλλάζοντας μόνο κάποιο μέρος του προγράμματος.
- ↳ Πολύ εύκολα επεκτάσιμα
- ↳ Μπορούμε να υλοποιήσουμε πολύπλοκες και έξυπνες επεξεργασίες που στο κλασσικό αυτοματισμό είναι εξαιρετικά δύσκολο να γίνουν.
- ↳ Ένα PLC μπορεί πολύ εύκολα να συνδεθεί με κάποιο ηλεκτρονικό υπολογιστή, με το σύστημα αποθήκης, με το λογιστήριο κλπ.
- ↳ Μπορούν να τοποθετηθούν και μέσα σε πεδίο ισχύος χωρίς πρόβλημα εφ' όσον τηρήσουμε τις οδηγίες του κατασκευαστή.
- ↳ Έχουμε την δυνατότητα να συνδέσουμε επάνω τους οθόνες, εκτυπωτές, πληκτρολόγια και HMI συστήματα.
- ↳ Οι γλώσσες προγραμματισμού καλύπτουν όλο το φάσμα των ανθρώπων που καλούνται να ασχοληθούν με την τεχνολογία αυτή.
- ↳ Έχουν μεγάλες δυνατότητες δικτύωσης με πρότυπα βιομηχανικά δίκτυα
- ↳ Μας δίνουν δυνατότητα αντιγραφής εφαρμογών, με την κατασκευή υπορουτινών ή βιβλιοθηκών.

### 1.3 Υλικά για τον έλεγχο μιας εγκατάστασης μέσω P.L.C

Στο επόμενο σχήμα παρουσιάζεται η δομή την οποία πρέπει να έχουμε σε μια εφαρμογή ελέγχου μέσω PLC.





➤ **Προγραμματίστρια**

Είναι το μέσο με το οποίο ο άνθρωπος επικοινωνεί με το PLC. Συνήθως είναι ένας ηλεκτρονικός υπολογιστής.

➤ **Πακέτο προγραμματισμού / software**

Είναι το πακέτο εκείνο που κάθε κατασκευαστής δίνει ώστε ο άνθρωπος να επικοινωνεί με την προγραμματίστρια.

➤ **Τροφοδοτικό**

Ο ρόλος του είναι να δημιουργεί τις αναγκαίες τάσεις που χρειάζεται το PLC για την τροφοδοσία του.

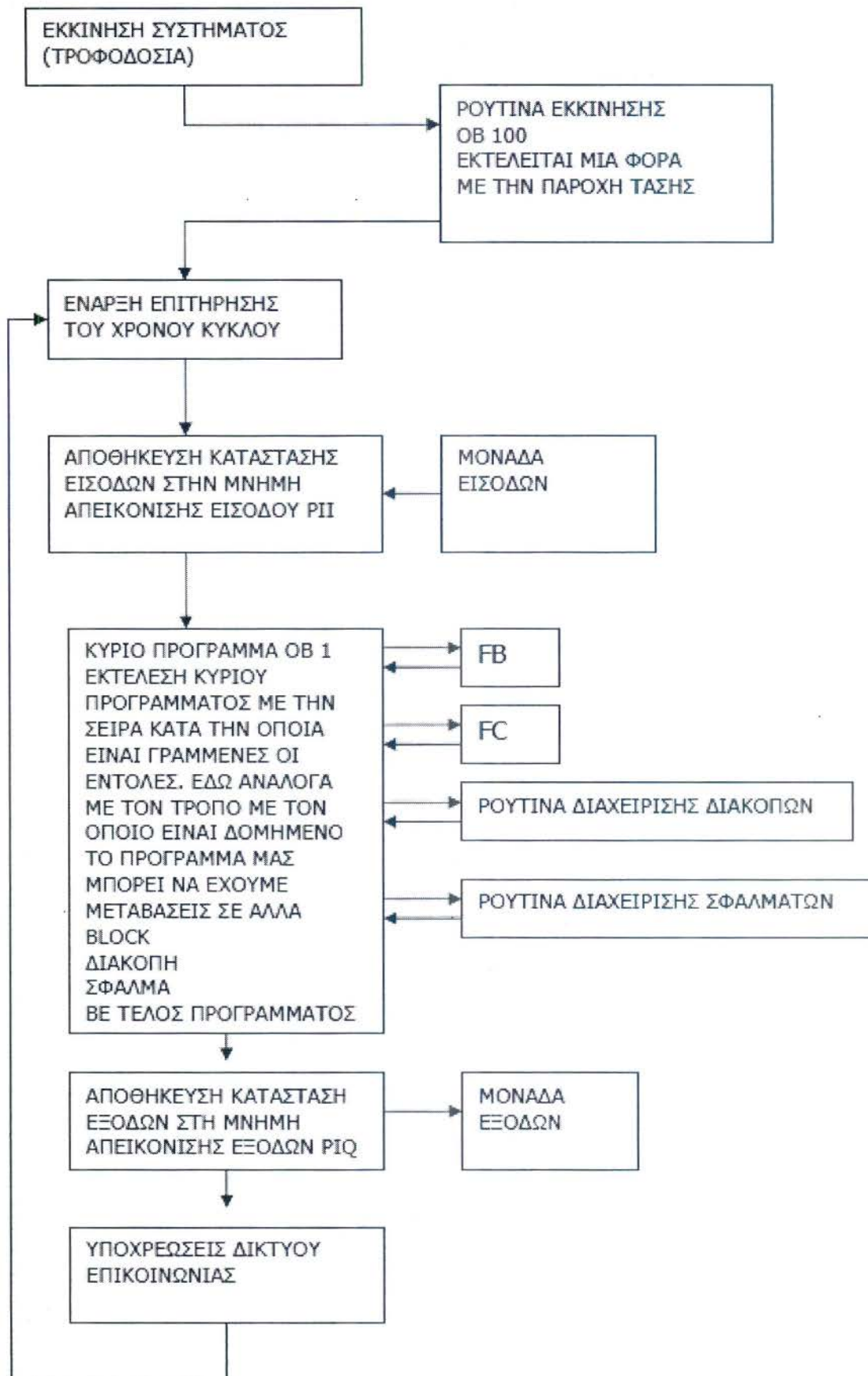
➤ **Κάρτες εισόδου**

Αυτές έχουν τον ρόλο να μετατρέπουν τα σήματα της εγκατάστασης σε σήματα τα οποία μπορεί να επεξεργαστεί η CPU.

➤ **Κάρτες εξόδου**

Αναλαμβάνουν να μετατρέπουν τα σήματα που έχει ήδη επεξεργαστεί η CPU σε κατάλληλες τάσεις τις οποίες στέλνουμε προς την ηλεκτρομηχανολογική εγκατάσταση.

Ξεκινώντας όμως οποιοδήποτε project χρησιμοποιώντας ένα PLC, οποιοδήποτε κατασκευαστή, πρέπει να έχουμε κατανοήσει απόλυτα ένα πολύ απλό αλλά ταυτόχρονα πολύ σημαντικό σημείο. Το σημείο αυτό ονομάζεται κύκλος λειτουργίας (*scan cycle*), το οποίο και περιγράφεται στο παρακάτω σχήμα.



### Παρατηρήσεις:

- Η πληροφορία για την κατάσταση της εισόδου αποκτάται μόνο στην αρχή του κύκλου και η κατάσταση της κατά τον χρόνο εκτέλεσης του προγράμματος θεωρείται σταθερή. Φυσικά για ιδιαίτερα κρίσιμες εισόδους υπάρχουν τεχνικές που επιτρέπουν την ακαριαία πληροφόρηση και δράση της CPU
- Η εκτέλεση μιας εντολής και η ενημέρωση της αντίστοιχης θέσης μνήμης γίνεται με την σειρά με την οποία είναι γραμμένη η εντολή στο πρόγραμμα.

Η συγκεκριμένη πτυχιακή εργασία υλοποιείται με PLC της σειράς **S7 - 300**, ελεγκτές της Siemens που προορίζονται για μεσαίας κλίμακας βιομηχανικές εφαρμογές.

### Τα κυριότερα χαρακτηριστικά τους είναι:

- ↪ Μνήμη προγράμματος μέχρι 512Kb
- ↪ Μέχρι 1024 εισόδους και εξόδους
- ↪ Ενσωματωμένη πολυκομβική διασύνδεση (MPI) για δημιουργία μικρών δικτύων και για σύνδεση με την προγραμματίστρια μονάδα
- ↪ Μεγάλη ταχύτητα, μία CPU μπορεί να εκτελέσει 1024 δυαδικές πράξεις σε 0.1-0.3 sec
- ↪ MODULAR μορφή
- ↪ Δυνατότητα επέκτασης έως και 32 κάρτες.
- ↪ Ενσωματωμένες ειδικές λειτουργίες: counters, positioners, έλεγχος κλειστού βρόχου με τις CPU 3xx IFM.
- ↪ Ενσωματωμένη διασύνδεση PROFIBUS-DP στη σειρά S7-300 2DP. Χρήση της CPU ως master ή slave
- ↪ Ενσωματωμένες λειτουργίες για HMI



- ↳ Εύκολη και γρήγορη διαμόρφωση και προγραμματισμός μέσω λογισμικού STEP7
- ↳ Εκτεταμένες διαγνωστικές λειτουργίες μέσω του STEP7. Μηνύματα σφαλμάτων που αποθηκεύονται στον διαγνωστικό buffer με αναγραφή ημερομηνίας και ώρας.
- ↳ Μεγάλη ποικιλία από CPU για καλύτερη επιλογή αναλόγως εφαρμογής.
- ↳ Μεγάλες δυνατότητες δικτύωσης (MPi, PROFIBUS, Industrial Ethernet)
- ↳ Μία μόνο κάρτα για όλους τους τύπους αναλογικών σημάτων.
- ↳ 32-bit σετ εντολών για μαθηματικές συναρτήσεις.
- ↳ Ελεύθερη διευθυνσιοδότηση των καρτών.



# ΚΕΦΑΛΑΙΟ 2<sup>ο</sup>

---



---

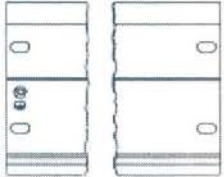
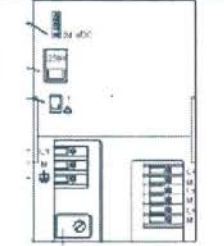

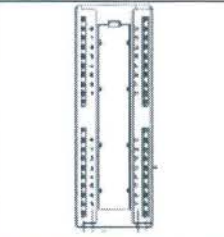

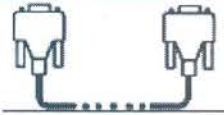
Προγραμματιζόμενος λογικός ελεγκτής Simatic s7-300



## 2.1 Δομή του προγραμματιζόμενου ελεγκτή

### 2.1.1 Υλικά

Ο SIMATIC S7300/400 είναι ένας προγραμματιζόμενος ελεγκτής που αποτελείται από βαθμίδες και περιέχει τα ακόλουθα υλικά:

ΟΝΟΜΑΣΙΑ	ΛΕΙΤΟΥΡΓΙΑ	ΜΟΡΦΟΛΟΓΙΑ
Πλαίσιο στήριξης (Rack)	Ο ρόλος του είναι απλά να στηρίζει τις διάφορες κάρτες που θα συνθέσουν το σύστημα αυτοματισμού.	
Τροφοδοτικό PS (Power Supply)	Μετατρέπει την τάση του δικτύου τροφοδοσίας στην κατάλληλη τάση λειτουργίας του PLC	
Κεντρική μονάδα επεξεργασίας (Central Processing Unit)	Εκτελεί λειτουργικό πρόγραμμα του PLC και το πρόγραμμα του χρήστη. Ελέγχει τις επικοινωνίες σε ένα MPI δίκτυο.	
Κάρτες Εισόδων / Εξόδων Ψηφιακές - αναλογικές (Analog- Digital SM)	Προσαρμόζουν τα ηλεκτρικά σήματα από το εξωτερικό περιβάλλον προς την CPU και αντιστρόφως.	
Καλώδιο Profibus δικτύου με τους bus connector	Συνδέει μεταξύ τους κόμβους ενός MPI ή Profibus δικτύου.	
Καλώδιο σύνδεσης προγραμματιστή (PG cable)	Συνδέει τη CPU με την συσκευή προγραμματισμού PG (μπορεί ως προγραμματιστής να χρησιμοποιηθεί ένας Η/Υ με adaptor cable).	

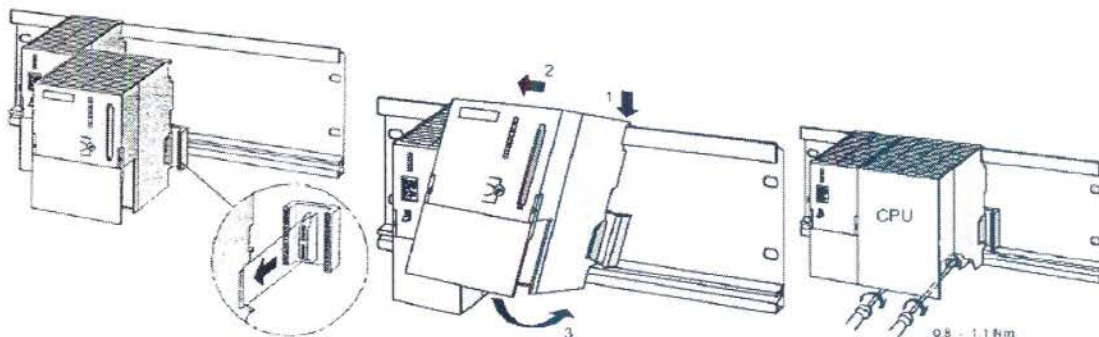
## 2.1.2 Συγκεντρωτική διαμόρφωση

Ας δούμε όμως αναλυτικά τα βασικά στοιχεία μιας S7-300 δομής

### 1. Πλαίσιο στήριξης (Rack)

Ο ρόλος του είναι να στηρίζει απλά τις διάφορες κάρτες που θα συνδέσουν το σύστημα αυτοματισμού. Πάνω σε κάθε rack πρέπει να τηρήσουμε μια ορισμένη σειρά στην σύνθεση του συστήματος μας. Στην πρώτη θέση του rack πρέπει να κουμπώσουμε την κάρτα του τροφοδοτικού, στην δεύτερη θέση πρέπει να τοποθετήσουμε την CPU, την τρίτη θέση είτε χρησιμοποιούμε είτε όχι κάρτα διασύνδεσης των rack (IM) πρέπει να την διαθέσουμε για αυτήν, από την τέταρτη θέση και πέρα πάνω στο rack συνδέω τα υπόλοιπα στοιχεία. Αυτά ισχύουν για το αρχικό rack (rack 0), Στα rack επέκτασης ξεκινάμε από την θέση 3 η οποία είναι αφιερωμένη για την κάρτα διασύνδεσης και πέρα. Κάθε rack εκτός από τα σταθερά που έχει (τροφοδοτικό, CPU, κάρτα διασύνδεσης) μπορεί να πάρει άλλες οκτώ κάρτες. Σ' ένα σύστημα με υλικό της σειράς S7 - 300 μπορούμε συνολικά να έχουμε έως τέσσερα πλαίσια στήριξης (rack).

→ Στήριξη καρτών στο (Rack)



Στην σειρά S7 - 300 το rack χρησιμεύει μόνο για την στήριξη των υλικών που συνθέτουν το σύστημα. Η επικοινωνία μεταξύ καρτών και CPU γίνεται με έναν συνδετήρα σχήματος «Π» στο πίσω μέρος των καρτών.

Μέσω αυτού υλοποιούνται δύο διαύλοι εσωτερικής επικοινωνίας:

❖ **Δίαυλος I/O (ή δίαυλος P - Bus → Peripheral Bus)**

Έχει σαν κύριο στόχο να μεταφέρει πληροφορίες που αφορούν την «περιφέρεια» (επικοινωνία με κάρτες εισόδου ή εξόδου) με ταχύτητα 1,5 Mbps

❖ **Δίαυλος επικοινωνίας (K - Bus → Communication Bus)**

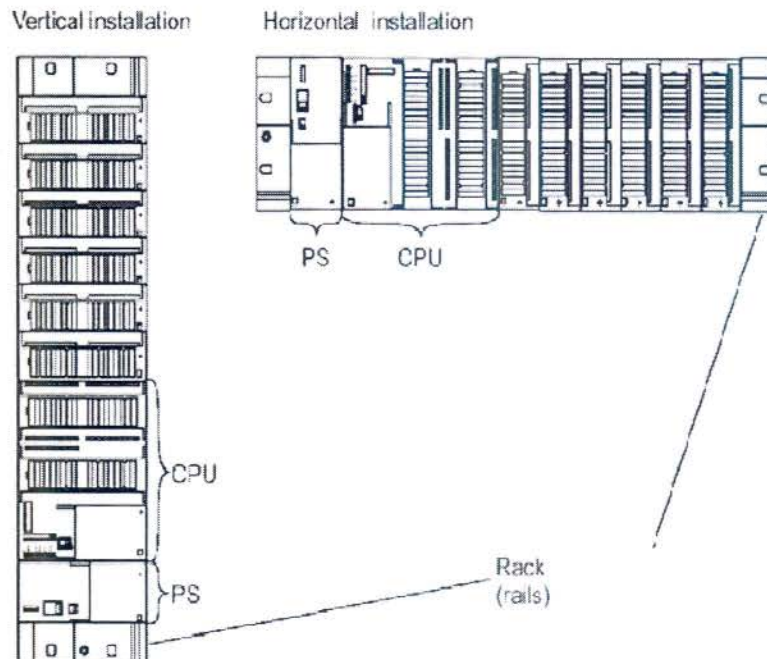
Είναι σχεδιασμένος για την ανταλλαγή μεγάλων ποσοτήτων δεδομένων. Αφορά την επικοινωνία με τις λεγόμενες «ειδικές» κάρτες (κάρτες απαρίθμησης, PID, FM, CP ...). Η πληροφορία μεταφέρεται σειριακά με ταχύτητα 187,5 Kbps.

Επίσης συνδέει την CPU και τη συσκευή διασύνδεσης προγραμματισμού (MPI) με τις βαθμίδες λειτουργίας και τους επεξεργαστές επικοινωνιών.

❖ **Επιμέρους διαύλοι (Local Bus Segment)**

Ένα ιδιαίτερο χαρακτηριστικό όσο αφορά τη διαμόρφωση είναι η χρήση της βαθμίδας FM 356 από τους υπολογιστές αυτοματισμού της οικογένειας M7-300. Η βαθμίδα αυτή μπορεί να διαχωρίσει τον εννιαίο δίαυλο που χρησιμοποιούν κάποιες βαθμίδες σε επιμέρους διαύλους και αναλάβει τον έλεγχο των βαθμίδων που χρησιμοποιούν κάθε έναν από αυτούς τους διαχωρισμένους διαύλους. Οι περιορισμοί που αναφέρθηκαν παραπάνω σχετικά με τον αριθμό των βαθμίδων και την κατανάλωση ισχύος, ισχύουν και στη περίπτωση αυτή.

Ένα σύστημα της σειράς S7-300 μπορεί να τοποθετηθεί οριζόντια ή κάθετα.



Σε έναν ελεγκτή S7-300 μπορούν να τοποθετηθούν στη κεντρική βάση στήριξης μέχρι και 8 βαθμίδες I/O.

Αν αυτή η διαμόρφωση ενός επιπέδου αποδειχθεί ανεπαρκής, τότε έχετε 2 επιλογές για ελεγκτές που είναι εφοδιασμένοι με CPU 314 ή μεγαλύτερη:

- ❖ Είτε θα επιλέξετε μια διαμόρφωση 2 επιπέδων (με IM-365, μέχρι και ένα μέτρο μεταξύ των βάσεων)
- ❖ ή θα επιλέξετε μια διαμόρφωση μέχρι και 4<sup>ων</sup> επιπέδων (με IM-360 & IM-361, μέχρι και 10m μεταξύ των βάσεων)

Μπορείτε να λειτουργήσετε μέχρι 8 βαθμίδες το πολύ σε μία βάση.

Ο αριθμός των βαθμίδων μπορεί να περιοριστεί από το μέγιστο επιτρεπόμενο ρεύμα ανα βάσης στήριξης το οποίο είναι 1,2 A (0,8 A για τη CPU 312 IFM).

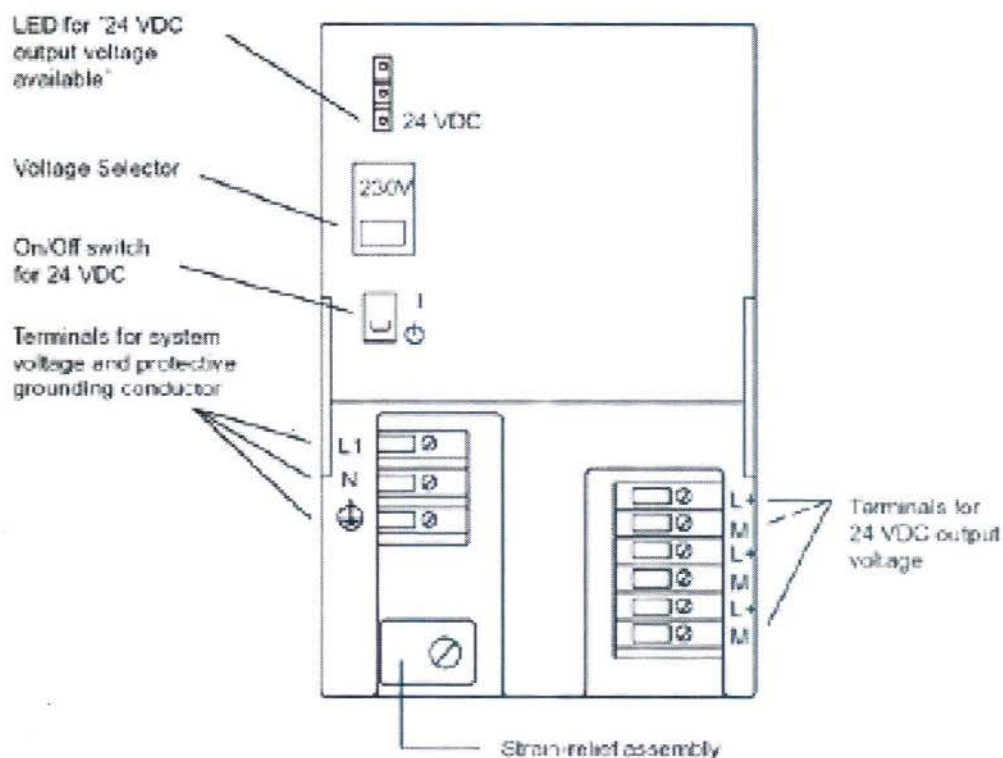
Οι βαθμίδες συνδέονται μεταξύ τους μέσω ενός ενιαίου διαύλου (Backplane Bus) ο οποίος συνδυάζει τις λειτουργίες των διαύλων P και K.

Στην επόμενη εικόνα παρουσιάζεται η μέγιστη δυνατή σύνθεση ενός συστήματος



## 2. ΤΡΟΦΟΔΟΤΙΚΟ PS (Power Supply)

Ο ρόλος του είναι να δημιουργήσει τις αναγκαίες τάσεις που χρειάζεται το PLC για την τροφοδοσία του. Το ονομαστικό ρεύμα εξόδου του τροφοδοτικού πρέπει να είναι πάντα μεγαλύτερο από το ρεύμα που απορροφούν όλες οι κάρτες που είναι τοποθετημένες στο rack.

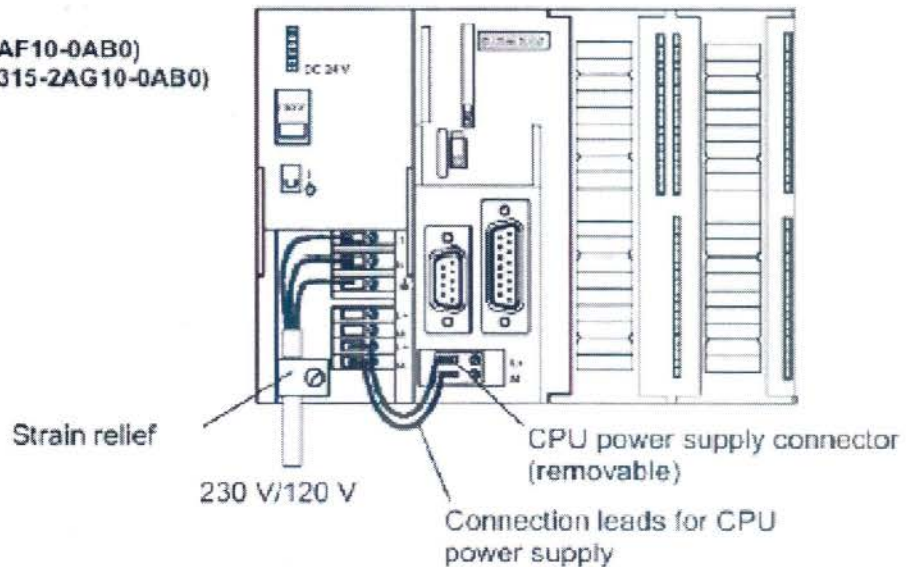


Διαθέτει:

- ❖ Κλέμες για τάση τροφοδοσίας (L1, N) και γείωση
- ❖ Κλέμες για τάση εξόδου 24 V (L+, M)
- ❖ Διακόπτης ON - OFF
- ❖ Επιλογικό διακόπτη τάσης τροφοδοσίας (230 VAC ή 120 VAC)
- ❖ Ενδεικτικά LED ύπαρξης τάσεως εξόδου 24 VDC

### Σύνδεση τροφοδοτικού με CPU

**CPU 31xC,312,  
314 (6ES7314-1AF10-0AB0)  
315-2 DP (6ES7315-2AG10-0AB0)**



### 3. Κεντρική Μονάδα Επεξεργασίας - CPU

Η κεντρική μονάδα επεξεργασίας η οποία συνηθίζεται να συμβολίζεται με CPU (Central Processing Unit) είναι ταυτόχρονα ο εγκέφαλος και η κινητήριος δύναμη ενός PLC.

Η κεντρική μονάδα επεξεργασίας πραγματοποιεί πολλαπλές βασικές λειτουργίες:

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

Κατά μια άποψη εάν συγκρίνουμε την CPU με την καλωδιωμένη λογική, τότε η CPU είναι το στοιχείο εκείνο το οποίο πραγματοποιεί τις καλωδιώσεις οι οποίες ζητούνται από τον κύκλο εργασίας της μηχανής ή της εγκατάστασης.

Σε αντίθεση όμως από την καλωδιωμένη λογική της οποίας η λειτουργία είναι «παράλληλη», το PLC εκτελεί τις λειτουργίες του με «σειριακό» τρόπο, για τον λόγο αυτό στα PLC είναι χαρακτηριστική η ταχύτητα λειτουργίας των κυκλωμάτων.

#### Εσωτερικά μια CPU περιλαμβάνει:

- Τον μικροεπεξεργαστή  
Αυτός εκτελεί τις εντολές των προγραμμάτων που έχει αποθηκευμένες η μνήμη, καθορίζει την σειρά εκτέλεσης των λειτουργιών του συστήματος και ελέγχει για τυχόν σφάλματα.
  
- Τη μνήμη
  - Η μνήμη μιας CPU χωρίζεται σε τρεις κατηγορίες
  - Τη μνήμη φόρτωσης (Load Memory)
  - Μνήμη εργασίας (Work Memory)
  - Μνήμη συστήματος (System Memory)

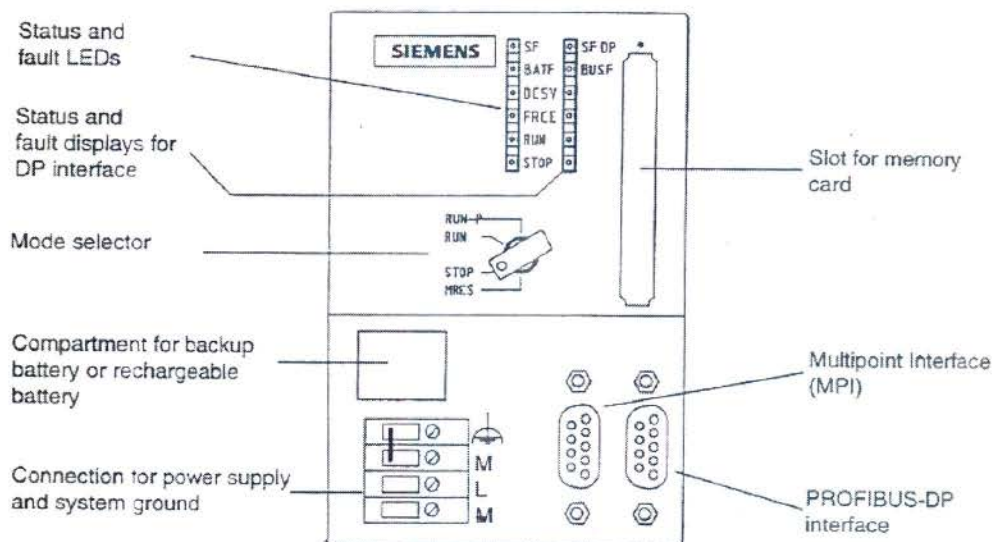
Οι περιοχές (ομάδες) που χωρίζεται η μνήμη συστήματος είναι:

- ↳ Μνήμη απεικόνισης εισόδων PII  
Σ' αυτήν την περιοχή αποθηκεύονται οι τιμές των εισόδων που διαβάζει η CPU από τις κάρτες εισόδου στην αρχή κάθε κύκλου λειτουργίας.
- ↳ Μνήμη απεικόνισης εξόδων PIQ  
Σ' αυτήν την περιοχή αποθηκεύεται η τιμή κάθε μια από τις χρησιμοποιούμενες εξόδους κατά την χρονική περίοδο του κύκλου λειτουργίας κατά την οποία εκτελείται το πρόγραμμα του χρήστη. Αυτή η περιοχή μνήμης στο τέλος του κύκλου στέλνεται για να ενημερώσει τις κάρτες εξόδου.
- ↳ Βοηθητικά Μ (Memory)  
Σ' αυτήν την περιοχή της μνήμης αποθηκεύονται ενδιάμεσα αποτελέσματα τα οποία έχουν υπολογιστεί κατά την εκτέλεση του προγράμματος.



- ↪ Χρονικά T (Timers)  
 Είναι η περιοχή της μνήμης του συστήματος όπου αποθηκεύονται οι χρόνοι των χρονικών που χρησιμοποιούμε.
- ↪ Απαριθμητές C (Counters)  
 Είναι η περιοχή της μνήμης του συστήματος όπου αποθηκεύονται τα περιεχόμενα των απαριθμητών.
- ↪ Τοπικά βοηθητικά L (Local Data)  
 Είναι η περιοχή της μνήμης του συστήματος όπου αποθηκεύονται προσωρινά δεδομένα ενός μπλοκ που περιέχει κώδικα (π.χ. ενός OB, FB, FC)  
 Τα τοπικά βοηθητικά έχουν ισχύ όσο τρέχει το συγκεκριμένο μπλοκ στο οποίο περιέχονται.
- ↪ Διαγνωστικά (Diagnostics)  
 Καταχωρούνται διάφορες ενέργειες που έχουν γίνει στο σύστημα με ώρα και ημερομηνία όπως CPU σε RUN/STOP, βραχυκυκλωμένη κάρτα αναλογικών κλπ.

Εξωτερικά μια CPU περιλαμβάνει:





- Ακροδέκτες τροφοδοσίας
- Θέση για μπαταρία (οι CPU που χρησιμοποιούν CF cards δεν έχουν).
- Διακόπτη με κλειδί RUN - P/RUN/STOP/MRES
- Ενδεικτικά LED για την κατάσταση της CPU
- Ενδεικτικά LED για την κατάσταση του PROFIBUS δικτύου
- Θέση για τοποθέτηση εξωτερικής μνήμης
- Θέση σύνδεσης συσκευής προγραμματισμού ή MPI δικτύου
- Θέση σύνδεσης PROFIBUS δικτύου.

Στην οικογένεια S7-300 υπάρχει μια μεγάλη γκάμα από διαφορετικές CPU στην διάθεση του χρήστη. Διαφέρουν κυρίως ως προς το:

- Εάν έχουν ή όχι ενσωματωμένες εισόδους / εξόδους
- Εάν έχουν ή όχι ενσωματωμένο profibus DP interface
- Πλήθος των εισόδων/ εξόδων που υποστηρίζουν
- Μέγεθος της ενσωματωμένης μνήμης RAM

#### 4. Ψηφιακές Μονάδες Εισόδων DI (Digital Input)

Η χρήση των μονάδων ψηφιακών εισόδων έχει τον σκοπό να μεταφέρει στην CPU τις καταστάσεις των διαφόρων αισθητηρίων ή διακοπών ελέγχου που χρησιμοποιούμε στην εγκατάσταση.

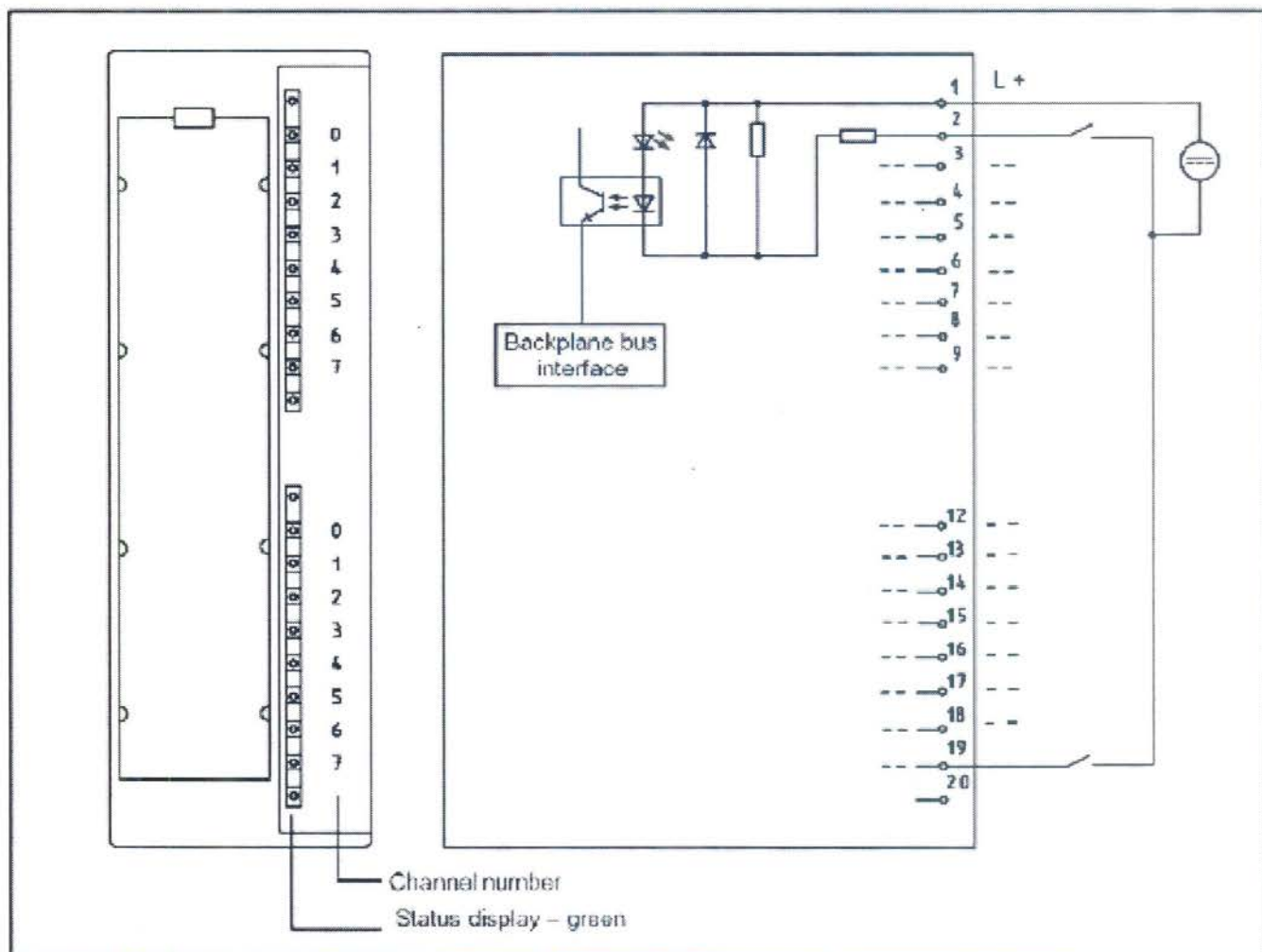
Μια μονάδα εισόδων έχει 8,16 ή 32 εισόδους ανάλογα με τον τύπο και τάση που χρησιμοποιεί. Οι περισσότερες συνηθισμένες τάσεις για τα σήματα εισόδου είναι 24 VDC ή 230 VAC.

Στα όρια μιας κάρτας πρέπει να χρησιμοποιείται η ίδια τάση, στα όρια όμως όλου του συστήματος μπορούμε να χρησιμοποιήσουμε μονάδες ψηφιακών εισόδων με διαφορετικές τάσεις.

Μια κάρτα ψηφιακών εισόδων των 24 VDC αναγνωρίζει σαν σήμα «+1» τα +24 VDC και σαν σήμα «0» τα 0 V. Στις περιπτώσεις εκείνες που υπάρχει διακόμανση

στην τάση (μη σταθεροποιημένο τροφοδοτικό) οι ψηφιακές κάρτες εισόδων έχουν ανοχές.

Έτσι σαν σήμα καταλαβαίνει τις τάσεις από +13 - +30 VDC και σαν σήμα «0» τις τάσεις από -3 - + 5 VDC. Για τις ενδιάμεσες τιμές τάσεων δηλαδή από +6 - +12 VDC δεν είναι δυνατόν να προκαθοριστεί για το πώς θα τις κατανοήσει το PLC. Στην κάτω εικόνα παρουσιάζεται η μορφολογία και η αρχή λειτουργίας μιας ψηφιακής κάρτας εισόδων.



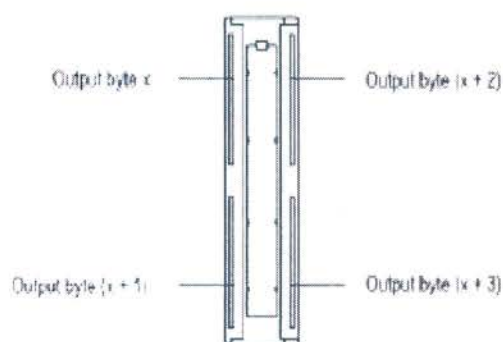
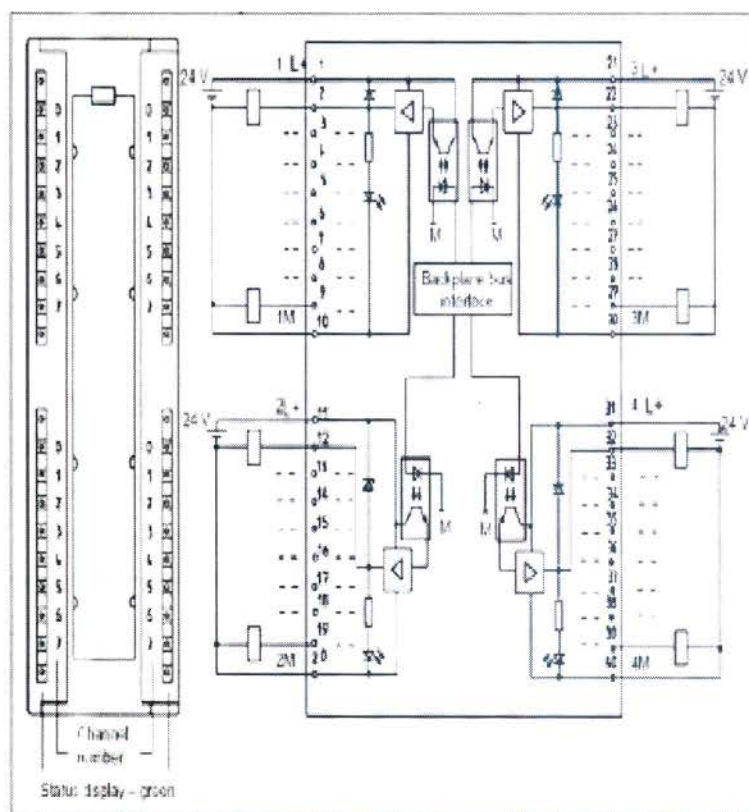
## 5. Ψηφιακές Μονάδες Εξόδου DO (Digital Output)

Ο ρόλος τους είναι να μετατρέπουν τις αποφάσεις που παίρνει η CPU σε εντολές προς την εγκατάσταση.

Οι αποφάσεις αυτές βρίσκονται καταχωρημένες στην μνήμη απεικόνισης των εξόδων στην CPU και μετατρέπονται σε ηλεκτρικά σήματα από τις κάρτες εξόδων.

Οι κάρτες εξόδων λειτουργούν σαν διακόπτες, στους οποίους δίνουμε εμείς την τάση (εξωτερικά) και όταν κλείσει ο διακόπτης η τάση περνάει και πηγαίνει προς το υπόλοιπο κύκλωμα.

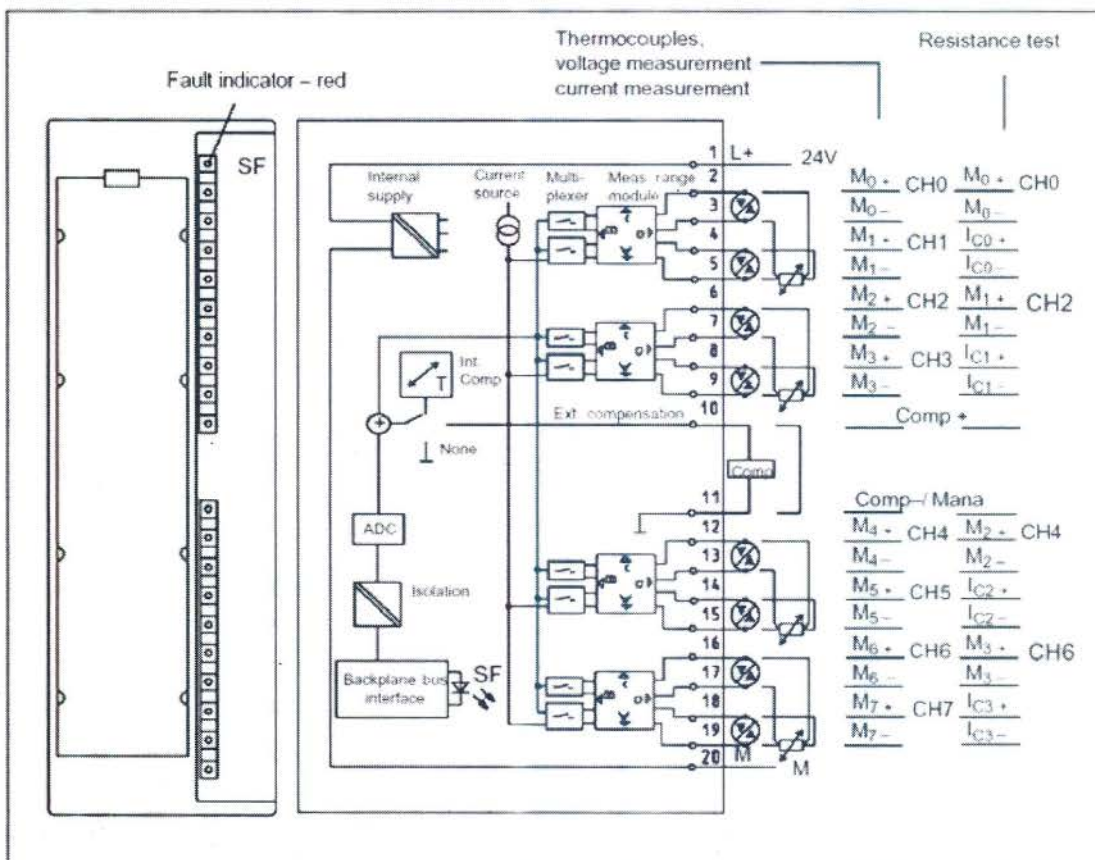
Σε αντιστοιχία με τις κάρτες εισόδου το πρώτο χαρακτηριστικό που πρέπει να λάβουμε υπ' όψη μας είναι η τάση και το ρεύμα εξόδου της κάρτας, αυτά θα πρέπει να συμφωνούν με τα αντίστοιχα του φορτίου (π.χ. ρελέ) που θα συνδέσουμε σε κάθε ψηφιακή έξοδο. Μια κάρτα ψηφιακών εξόδων έχει 8, 16, ή 32 εξόδους ανάλογα με τον τύπο και την τάση που έχουν. Στα όρια μιας κάρτας χρησιμοποιείται πάντοτε η ίδια τάση. Στην κάτω εικόνα παρουσιάζεται η μορφολογία και η αρχή λειτουργίας μιας ψηφιακής κάρτας εξόδων.



Ένα επί πλέον ιδιαίτερο χαρακτηριστικό των DO είναι το στοιχείο εξόδου (αυτό που παρέχει την ισχύ στο φορτίο). Αυτό συνήθως είναι τρανζίστορ αν πρόκειται για DC κάρτα εξόδων ή triac ή ρελέ εάν πρόκειται για AC κάρτα εξόδου. Όλες οι ψηφιακές εξοδοί είναι γαλβανικά απομονωμένες.

### 6. Μονάδες Αναλογικών Εισόδων AI (Analog Input)

Για να επεξεργαστούμε ηλεκτρικά σήματα, με συνεχή μεταβολή της τιμής τους, στο PLC χρειαζόμαστε κάρτες αναλογικών σημάτων. Οι κάρτες αναλογικών εισόδων έχουν τον ρόλο να διαβάζουν ένα ηλεκτρικό μέγεθος και να το μετατρέπουν σε ένα αριθμό (δυαδική αναπαράσταση) το οποίο πλέον μπορεί η CPU να αναγνωρίσει και να επεξεργαστεί. Οι κάρτες αναλογικών εισόδων δέχονται ηλεκτρικά σήματα τάσης ή έντασης.



Οι

τυποποιημένες τιμές έντασης τις οποίες μπορεί να διαβάσει μια αναλογική κάρτα εισόδων είναι 0 -20 mA ή 4 - 20 mA για δε τα σήματα τάσης έχουμε 0 - 10 V ή  $\pm 10$  V. Ένα άλλο μέγεθος που μας ενδιαφέρει στην επιλογή μιας κάρτας αναλογικών εισόδων είναι η διακριτική τους ικανότητα (ακρίβεια). Κάθε αναλογικό σήμα καταλαμβάνει χώρο 16 bit.

Ένα ακόμα μεγάλο πλεονέκτημα της σειράς S7 είναι ότι μια αναλογική κάρτα εισόδων μπορεί να γίνει τάσης ή έντασης και να μεταβάλουμε την περιοχή μέτρησης της επεμβαίνοντας τόσο εξωτερικά πάνω στην ίδια την κάρτα όσο και στο software.

#### 7. Μονάδες Αναλογικών Εξόδων A/O (Analog Output)

Οι κάρτες αναλογικών εξόδων έχουν τον ρόλο να μετατρέψουν το αριθμητικό μέγεθος με το οποίο επεξεργάζεται τα δεδομένα η CPU στην κατάλληλη τιμή έντασης ή τάσης ώστε να μπορεί να οδηγηθεί το ανάλογο εξάρτημα που ελέγχει το φυσικό μέγεθος της εγκατάστασης μας.

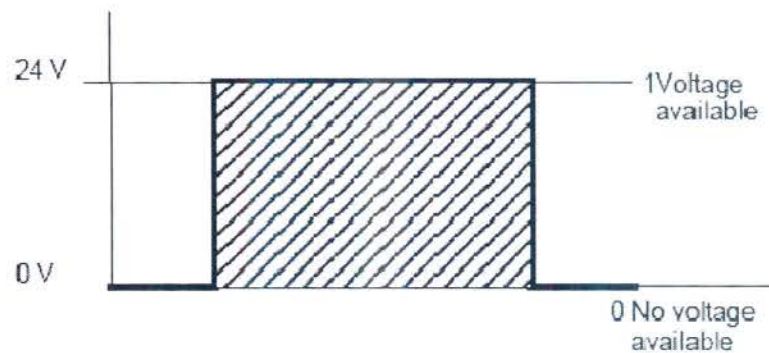
Όλα τα χαρακτηριστικά των καρτών είναι σε πλήρη αντιστοιχία με αυτή των αναλογικών εισόδων μια και εκτελούν απλώς την αντίστροφη διαδικασία όποτε δεν απαιτείται κάποια ιδιαίτερη συζήτηση.

### 2.1.3 Διευθυνσιοδότηση - Ονοματολογία

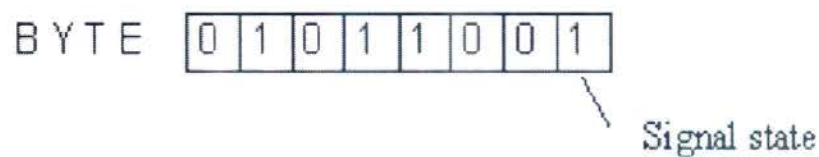
➤ Διευθυσιοδότηση

Έννοιες bit, byte, word, double word

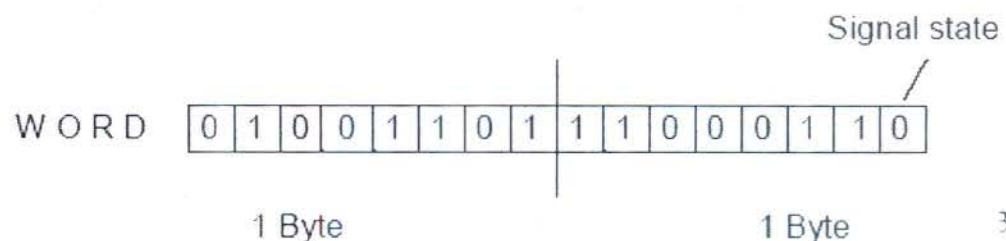
- ❖ **Bit:** Το bit είναι η μικρότερη μονάδα αποθήκευσης της κατάστασης ενός ψηφιακού σήματος. Το bit είναι ο χώρος μιας κυψέλης μνήμης και μπορεί να πάρει δύο τιμές την κατάσταση «0» η οποία αντιστοιχεί στην μη ύπαρξη τάσης στο ψηφιακό σήμα και την κατάσταση «1» η οποία αντιστοιχεί στην ύπαρξη τάσης στο ψηφιακό σήμα.



- ❖ **Byte:** Μια ομάδα από 8 συνεχόμενα bit ορίζει ένα byte

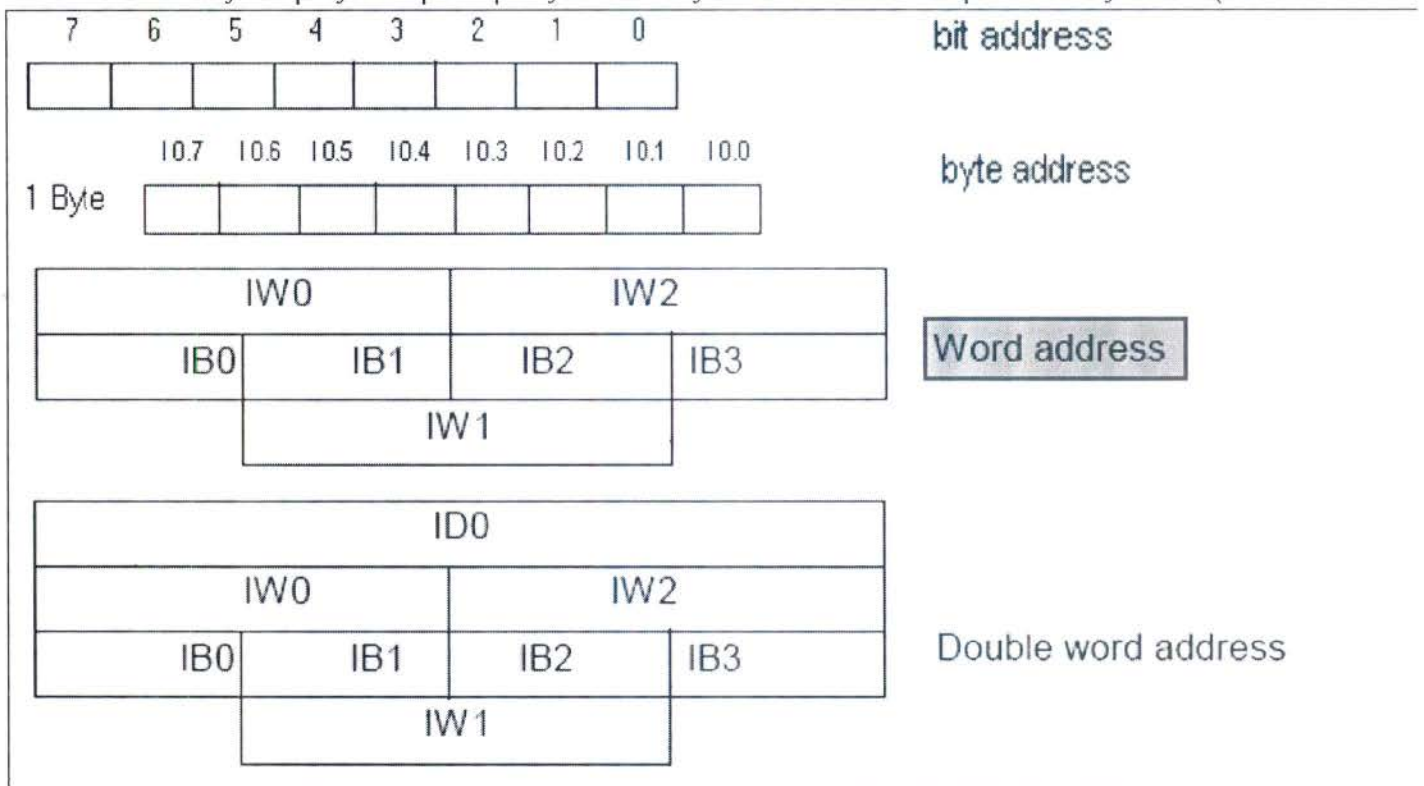


- ❖ **Word:** 2 συνεχόμενα Byte ορίζουν ένα Word



❖ **Double Word:** 2 συνεχόμενα Word ορίζουν ένα Double Word

Το κάθε bit έχει μια συγκεκριμένη διεύθυνση (αριθμό). Η αριθμηση γίνεται πάντοτε από τα δεξιά προς τα αριστερά ξεκινώντας από το bit 0 και φθάνοντας στο 7 (στα



byte) 15 (στις word) ή 31 (στις double word). Στην παρακάτω εικόνα δίνεται η γραφική αναπαράσταση των εννοιών bit, byte, word, double word.



### Διεύθυνση θέσης

Κάθε μονάδα η οποία ανήκει σ' ένα σύστημα αυτοματισμού με PLC της σειράς S7 έχει μια διεύθυνση θέσης. Αυτή αποτελείται από τον αριθμό του rack που είναι τοποθετημένη η μονάδα και τον αριθμό της θέσης της. Την πρώτη πάντα θέση καταλαμβάνει το τροφοδοτικό, την δεύτερη θέση η CPU, η τρίτη θέση ανήκει στην κάρτα διασύνδεσης του Rack (IM). Εάν έχουμε τέτοια κάρτα χρησιμοποιούμε την τρίτη θέση εάν το σύστημα μας δεν διαθέτει τέτοια κάρτα η θέση 3 παραμένει υποχρεωτικά κενή.

### Λογική διεύθυνση

Πέρα από την διεύθυνση θέσης, κάθε μονάδα έχει μια αρχική διεύθυνση (διεύθυνση του πρώτου byte που καταλαμβάνει στον χώρο μνήμης που ανήκει) η οποία καθορίζει τη θέση της στο χώρο των λογικών διευθύνσεων. Ο χώρος των λογικών διευθύνσεων ξεκινάει από την διεύθυνση 0 και τελειώνει σε διεύθυνση που εξαρτάται από την χρησιμοποιούμενη CPU. Η λογική διεύθυνση σε ένα σύστημα εξαρτάται από την θέση που βρίσκεται η κάρτα σε σχέση με την CPU και από το εάν είναι ψηφιακή ή αναλογική. Ανάλογα λοιπόν με την θέση και το είδος της κάρτας ισχύει ο παρακάτω πίνακας.

Rack	Αρχική διεύθυνση της μονάδας	Αριθμός θέσης										
		1	2	3	4	5	6	7	8	9	10	11
0	Ψηφιακά	PS	CPU	IM	0	4	8	12	16	20	24	28
	Αναλογικά				256	272	288	304	320	336	352	368
1 <sup>1</sup>	Ψηφιακά	-	-	IM	32	36	40	44	48	52	56	60
	Αναλογικά				384	400	416	432	448	464	480	496
2 <sup>1</sup>	Ψηφιακά	-	-	IM	64	68	72	76	80	84	88	92
	Αναλογικά				512	528	544	560	576	592	608	624
3 <sup>1</sup>	Ψηφιακά	-	-	IM	96	100	104	108	112	116	120	124 <sup>2</sup>
	Αναλογικά				640	656	672	688	704	720	735	752 <sup>2</sup>

1 - όχι με τις CPU312IFM/313

2 - όχι με τη CPU314IFM



### ➤ Ονοματολογία

Για να ορίσουμε μια παράμετρο σ' ένα σύστημα αυτοματισμού με PLC χρησιμοποιούμε ένα συνδυασμό γραμμάτων και αριθμών. Τα μεν γράμματα είναι τα διευκρινιστικά εκείνα στοιχεία που κατατάσσουν την παράμετρο σε μια ομάδα (π.χ. εισόδοι, έξοδοι, εσωτερικά, βοηθητικά) οι δε αριθμοί είναι τα στοιχεία εκείνα τα οποία ορίζουν την διεύθυνση μιας συγκεκριμένης παραμέτρου. Για την σειρά S7 και στην αγγλική γλώσσα χρησιμοποιείται η εξής ονοματολογία.

#### Εισόδοι I (Inputs)

Μια ψηφιακή είσοδος συμβολίζεται με το γράμμα I και η ονοματολογία της έχει τη μορφή.

**I x.y** όπου x: Διεύθυνση byte (0 ...n)  
y: Διεύθυνση bit (0 ... 7)

Έχουμε την δυνατότητα να παρουσιάσουμε ή να ζητήσουμε:

#### ❖ **Byte εισόδων: IBX**

Παράδειγμα: IB3 με αυτήν την ονοματολογία δηλώνουμε τις εισόδους I3.0 ... I3.7

#### ❖ **Word εισόδων: !WX**

Παράδειγμα: IW2 με αυτήν την ονοματολογία δηλώνουμε τις εισόδους I2.0 ... I2.7, I3.0, I3.1, ... I3.7

#### ❖ **Double word εισόδων: !DX**

Παράδειγμα: ID4 με αυτήν την ονοματολογία δηλώνουμε τις εισόδους I4.0 ..... I4.7, I5.0 ... I5.7, I6.0 ... I6.7, I7.0 ... I7.7

### Εξοδοι Q (Output)

Μια ψηφιακή έξοδος συμβολίζεται με το γράμμα Q και η ονοματολογία της έχει τη μορφή.

**Q x.y** όπου x: διεύθυνση byte  
y: διεύθυνση bit (0..7).

Όπως στις ψηφιακές εισόδους έτσι και για τις ψηφιακές εξόδους έχουμε byte εξόδων, Word εξόδων, double word εξόδων.

### Βοηθητικά M (Memory bit)

Τα βοηθητικά παίζουν τον ρόλο των βοηθητικών ρελέ στον κλασικό αυτοματισμό, τα χρησιμοποιούμε στο πρόγραμμα για να αποθηκεύσουμε λογικό αποτέλεσμα τμήματος του προγράμματος (ειδικά όταν αυτό είναι επαναλαμβανόμενο). Είναι ρελέ του οποίου

το λογικό αποτέλεσμα δεν μπορώ να πάρω απ' ευθείας στην κάρτα εξόδου. Ένα βοηθητικό συμβολίζεται με το γράμμα M και η ονοματολογία του έχει τη μορφή:

**M x.y** όπου x: διεύθυνση byte (0...n)  
y: διεύθυνση bit (0...7)

Και εδώ έχουμε **MBX, MWX, MDX**

### Χρονικά T (Timers)

Η λειτουργία χρονικών χρησιμοποιείται για να υλοποιήσει αλγορίθμους που έχουν σχέση με χρόνο ( επιτήρηση, αναμονή, μέτρηση χρονικών διαστήματος, δημιουργία παλμών). Με τον όρο «χρονικό» εννοούμε μια λέξη (word) σε μια ειδική περιοχή της μνήμης, αυτή των χρονικών. Τα χρονικά συμβολίζονται με το γράμμα T και η ονοματολογία του έχει τη μορφή :

**Tx** όπου x: αριθμός του χρονικού (0... n)

### Απαριθμητές C (counters)

Οι λειτουργίες απαριθμητή μας δίνουν τη δυνατότητα να εκτελούμε εργασίες απαρίθμησης απ' ευθείας από την CPU. Με τον όρο απαριθμητής εννοούμε μια λέξη (Word) σε μια ειδική περιοχή της μνήμης, αυτή των απαριθμητών. Ο απαριθμητής συμβολίζονται με το γράμμα C και η ονοματολογία που έχει τη μορφή:

**Cx**, όπου x: αριθμός του απαριθμητή (0. n)

## .2 Δομή του Προγράμματος

### ➤ Δομή Project

Κατά τη φάση του σχεδιασμού του project μας ένα από τα πρώτα πράγματα που πρέπει να κάνουμε είναι στο να αποφασίσουμε με ποιόν τρόπο θα δομήσουμε το πρόγραμμα μας δηλαδή στο τι μπλοκ θα περιέχει και πως θα συνδέονται μεταξύ τους αυτά τα μπλοκ. Ας δούμε όμως πρώτα πως είναι οργανωμένο ένα πρόγραμμα στην CPU.

Κάθε CPU περιλαμβάνει δύο προγράμματα ανεξάρτητα το ένα από το άλλο:

### ❖ Λειτουργικό Σύστημα

Το λειτουργικό σύστημα είναι το σύνολο των ορισμών και εντολών που ελέγχουν τους πόρους του συστήματος. Είναι αυτό που ενημερώνει το ρολόι του πραγματικού χρόνου στη CPU, που ελέγχει την κατάσταση του διακόπτη της CPU, (RUN, STOP, ...), ελέγχει να ανάψει τα LED στη CPU, να ρυθμίσει τις επικοινωνίες μέσα απ το MPI interface, ... Στο λειτουργικό σύστημα δεν μπορούμε να κάνουμε μεταβολές,

μπορούμε όμως να διαβάσουμε ή να χρησιμοποιήσουμε ορισμένα αποτελέσματα αυτού (π.χ. το ρολόι πραγματικού χρόνου).

### ❖ Πρόγραμμα Εφαρμογής

Το πρόγραμμα εφαρμογής είναι το σύνολο των εντολών και ορισμών που χρειάζεται το PLC για τον έλεγχο της εγκατάστασης. Η δομή ενός προγράμματος εφαρμογής δίνεται στην παρακάτω εικόνα.



### ➔ Πρόγραμμα Χρήστη

Είναι το πρόγραμμα που εμείς γράφουμε για τις λειτουργικές ανάγκες της εγκατάστασης και του αυτοματισμού. Αυτό μπορεί να περιέχει **μπλοκ λογικής** (εντολές) και **μπλοκ δεδομένων** (όπου καταχωρούνται λίστες με αριθμούς).

#### → Μπλόκ Συστήματος

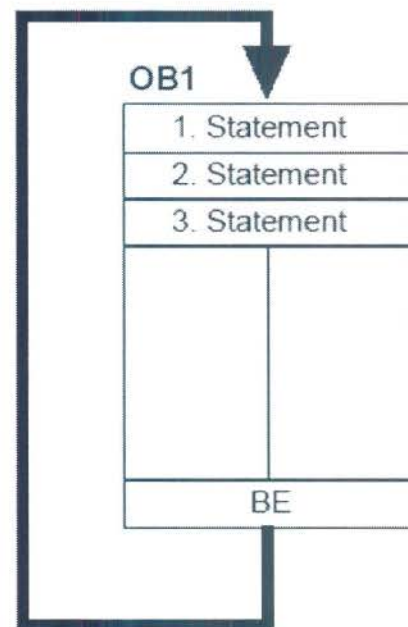
Είναι λειτουργίες που είναι από πριν ορισμένες και καταχωρημένες στο λειτουργικό σύστημα του PLC. Στο πρόγραμμα του ο χρήστης καλεί αυτά τα μπλοκ σε οποιοδήποτε σημείο θέλει, τους δίνει κάποιες παραμέτρους και παίρνει μόνο τα αποτελέσματα, χωρίς να ενδιαφέρεται για το πώς έχουν αυτά παραχθεί.

#### → Standard Μπλόκ

Είναι μπλοκ που μας προσφέρουν έτοιμες λύσεις για τυποποιημένες εργασίες αυτοματισμού που πιθανόν να μας ενδιαφέρουν.

### Γραμμικό Πρόγραμμα

Όλο το πρόγραμμα του χρήστη βρίσκεται σ' ένα συνεχόμενο μπλοκ (OB1 που καλείται αυτόματα κάθε κύκλο λειτουργίας). Η CPU επεξεργάζεται τις εντολές την μια μετά την άλλη μέχρι το τέλος του μπλοκ και ξαναρχίζει η ίδια διαδικασία πάλι από αρχή. Έχει το πλεονέκτημα ότι εύκολα και γρήγορα αρχίζει κάποιος τη φάση του προγραμματισμού. Έχει το μειονέκτημα ότι σε μεγάλα προγράμματα είναι δύσκολο να εντοπίσουμε που γίνεται μια συγκεκριμένη εργασία. Χρησιμοποιείται για μικρές εφαρμογές.

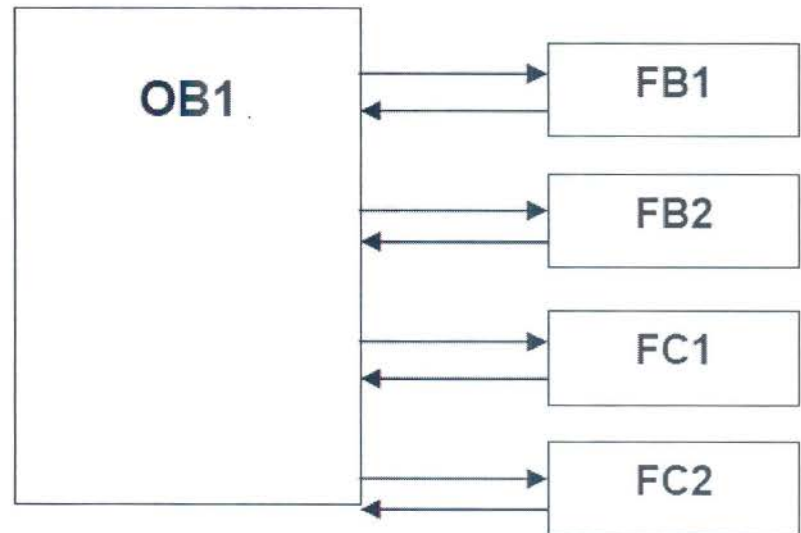


σε

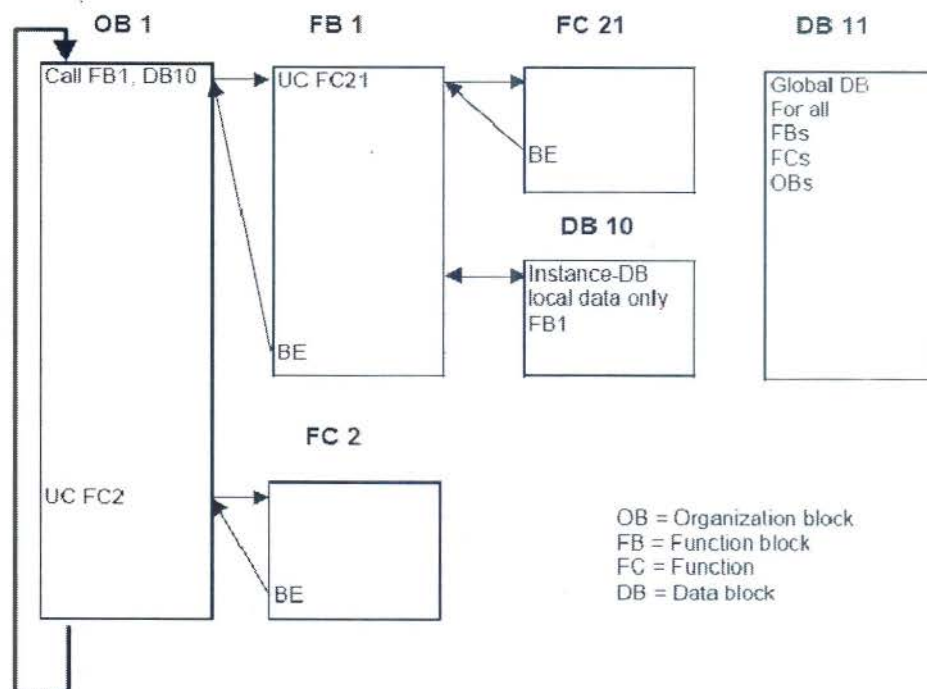
την

### Τμηματοποιημένο Πρόγραμμα

Το πρόγραμμα χωρίζεται σε μπλοκ όπου κάθε ένα από αυτά υλοποιεί μια συγκεκριμένη εργασία. Για τον τρόπο κλήσης, την σωστή λειτουργία τους καθώς και την σωστή σειρά εκτέλεσης τους φροντίζει ένα ειδικό μπλοκ το οποίο λέγεται μπλοκ οργάνωσης (OB1).



### Δομημένο Πρόγραμμα



Ένα δομημένο πρόγραμμα μπορεί να περιλαμβάνει παραμετροποιημένα μπλοκ. Αυτά τα μπλοκ είναι έτσι σχεδιασμένα ώστε να μπορούν να είναι γενικής χρήσης. Όταν καλείται ένα τέτοιο μπλοκ του δίνουμε τιμές στις παραμέτρους για την διαδικασία που μας ενδιαφέρει (διευθύνσεις εισόδων, εξόδων, χρονικά). Ο δομημένος προγραμματισμός μας προσφέρει πολλά πλεονεκτήματα, όπως :

- Εξοικονόμηση μνήμης (δεν επαναλαμβάνουμε το γράψιμο ίδιων προγραμμάτων)
- Οποιαδήποτε αλλαγή στη λογική του αυτοματισμού την περνάμε μια φορά στο πρόγραμμα και αυτόματα γίνεται η διόρθωση της λειτουργίας όπου χρειάζεται (εξοικονόμηση χρόνου και ελαχιστοποίηση της πιθανότητας σφάλματος από λανθασμένη πληκτρολόγηση).

### 2.2.1 Τύποι των Διαθέσιμων Μπλόκ

Για το χτίσιμο της εφαρμογής μας έχουμε στην διάθεση μας διαφορετικά είδη μπλοκ προγραμματισμού. Το τι θα χρησιμοποιήσουμε και πως θα τα διασύνδεουμε είναι τις περισσότερες φορές υποκειμενική υπόθεση και εξαρτάται από την εφαρμογή που έχουμε να προγραμματίσουμε.

Οι διάφοροι τύποι των διαθέσιμων μπλοκ είναι:

- Μπλόκ Οργάνωσης OB (Organization Blocks).

Έχουν τον ρόλο του διαμεσολαβητή μεταξύ του λειτουργικού συστήματος και του προγράμματος του χρήστη. Κατά την εκδήλωση κάποιων ειδικών γεγονότων, όπως για παράδειγμα μιας χρονικής διακοπής, μιας διακοπής τροφοδοσίας, το λειτουργικό σύστημα της CPU καλεί το αντίστοιχο μπλοκ οργάνωσης. Ένα από τα διάφορα μπλοκ οργάνωσης, σημαντικότερο απ' όλα είναι το OB1. Αυτό είναι ένα μπλοκ το οποίο η CPU καλεί αυτόματα και το εκτελεί συνεχώς κυκλικά. Μέσα σ' αυτό το μπλοκ βρίσκεται το κύριο πρόγραμμα του χρήστη.



Άλλο σημαντικό μπλοκ είναι το OB100 που εκτελείται μία φορά όταν δίνουμε τάση στο σύστημα.

Τα μπλοκ οργάνωσης έχουν τάξεις προτεραιότητας από 0 ως 29. Αν ένα μπλοκ έχει μεγαλύτερη προτεραιότητα από κάποιο άλλο, τότε μπορεί να το διακόψει και να εκτελεστεί το ίδιο. Π.χ. το OB1 που έχει προτεραιότητα 1 μπορεί να διακοπεί από όλα τα άλλα μπλοκ.

Στον κάτω πίνακα δίνονται όλα τα διαθέσιμα μπλοκ οργάνωσης SIMATIC S7, το καθένα μαζί με την προτεραιότητα του.

Μπλοκ οργάνωσης	Συνθήκες κλήσης	Προτεραιότητα	
		Προεπιλεγμένη	Τροποποιήσιμη
Ελεύθερος κύκλος OB 1	Κυκλικά μέσω του λειτουργικού συστήματος	1	Όχι
Χρονικές διακοπές (TOD) OB 10 ως OB 17	Σε συγκεκριμένη ώρα της ημέρας ή σε τακτά χρονικά διαστήματα (π.χ. μηνιαίως)	2	2 ως 24
Διακοπές καθυστέρησης OB 20 ως OB 23	Μετά από προγραμματισμένο χρόνο, ελεγχόμενο από το πρόγραμμα χρήστη	3 ως 6	2 ως 24
Διακοπές χρονιστή επιτήρησης/: OB 30 ως OB 38	Τακτικά σε προγραμματισμένα χρονικά διαστήματα (π.χ. κάθε 100ms)	7 ως 15	2 ως 24
Διακοπές επεξεργασίας OB 40 ως OB 47	Σε σήματα διακοπών από τις βαθμίδες I/O	16 ως 23	2 ως 24
Διακοπή πολυεπεξεργασίας OB 60	Κλήση υπό συνθήκες μέσω του προγράμματος χρήστη σε κατάσταση πολυεπεξεργασίας	25	Όχι
Εφεδρικά σφάλματα OB 70, OB 72, OB 73	Στην περίπτωση απώλειας εφεδρικού στοιχείου που απορρέει από σφάλμα I/O, Στην περίπτωση εφεδρικού σφάλματος της CPU Στην περίπτωση εφεδρικού σφάλματος επικοινωνιών	25 28 25	2 ως 26 2 ως 28 2 ως 26

Ασύγχρονα σφάλματα OB 80, OB81 ως OB 84, OB 86, OB87, OB 85	Λάθη που δεν σχετίζονται με την εκτέλεση του προγράμματος (π.χ. χρονικά σφάλματα, σφάλματα ΣΕ, διαγνωστικές διακοπές, διακοπές εγκατάστασης ή απεγκατάστασης βαθμίδων, αποτυχία βάσης στήριξης ή σταθμού)	26 <sup>2)</sup> 26 <sup>2)</sup>	26 2 ως 26 24 ως 26
Εκτέλεση στο παρασκήνιο OB90	Ελάχιστη διάρκεια χρονικού κύκλου που δεν έχει επιτευχθεί ακόμα	29 <sup>1)</sup>	Όχι
Ρουτίνα εκκίνησης OB 100, OB 101, OB 102	Σε προγραμματιζόμενη εκκίνηση ελεγκτή	27	
Σύγχρονα σφάλματα OB 121, OB 122	Σφάλματα που σχετίζονται με την εκτέλεση του προγράμματος (π.χ. σφάλματα πρόσβασης I/O)	Η προτεραιότητα των OB που προκαλεί τα σφάλματα	

➔ Συναρτήσεις FC (Functions)

Οι συναρτήσεις είναι μπλοκ τα οποία προγραμματίζονται από τον χρήστη. Τα FC είναι μπλοκ κώδικα «στερούμενο μνήμης». Οι προσωρινές μεταβλητές (temporary variables) των FC αποθηκεύονται στην περιοχή των τοπικών δεδομένων (local data stack). Μετά την επεξεργασία των FC αυτά τα δεδομένα χάνονται. Για την αποθήκευση των δεδομένων τα FC μπορούν να χρησιμοποιήσουν DB (shared data blocks).

Ένα FC περιέχει ένα πρόγραμμα το οποίο εκτελείται όταν το FC καλείται από ένα άλλο μπλοκ που περιέχει κώδικα.

Τα FC χρησιμοποιούνται για:

- Υπολογισμό κάποιας συνάρτησης και απόδοσης τιμής στο μπλοκ που το έχει καλέσει (π.χ. υπολογισμός μαθηματικών συναρτήσεων).
- Έλεγχο μιας τεχνολογικής συνάρτησης (π.χ. έλεγχος ανεξάρτητων τμημάτων εγκατάστασης).
- Συχνά επαναλαμβανόμενες λειτουργίες αυτοματισμού

Τα FC παραμετροποιούνται και επομένως μπορούν να χρησιμοποιηθούν για περιπτώσεις στις οποίες έχουμε επαναλαμβανόμενη λογική στο πρόγραμμα μας με διαφορετικές παραμέτρους.

→ Μπλόκ Συναρτήσεων FB (Function Block)

Τα μπλοκ συναρτήσεων προγραμματίζονται και αυτά από τον χρήστη και περιέχουν κώδικα. Ένα μπλοκ συνάρτησης «έχει μνήμη», δηλαδή σε αυτό διατίθεται ένα μπλοκ δεδομένων (DB) σαν δικιά του μνήμη. Αυτό το DB

λέγεται (instance data block) και είναι μόνιμα δεσμευμένα με το μπλοκ συνάρτησης και για την ακρίβεια με την κλήση (call) του μπλοκ συνάρτησης. Επίσης είναι δυνατόν σε κάθε κλήση μπλοκ συνάρτησης να εκχωρηθεί ένα διαφορετικό μπλοκ δεδομένων (με την ίδια δομή αλλά με διαφορετικές τιμές).

Τα FB παραμετροποιούνται όπως και τα FC επομένως και αυτά χρησιμοποιούνται σε περιπτώσεις που έχουν επαναλαμβανόμενη λογική. Όταν δεν παραμετροποιούνται η λειτουργία τους δεν διαφέρει σε τίποτα από τα FC. Τόσο οι παράμετροι οι οποίες μεταβιβάζονται στα FB όσο και οι στατικές μεταβλητές (static variables) αποθηκεύονται στο instance data block. Οι προσωρινές μεταβλητές (temporary variables) αποθηκεύονται στην περιοχή των τοπικών δεδομένων. Στο τέλος της επεξεργασίας του FB όσα δεδομένα αποθηκεύτηκαν στο instance data block δεν χάνονται ενώ αυτά τα δεδομένα τα οποία αποθηκεύονται στην περιοχή των τοπικών δεδομένων (local data stack) χάνονται. Τα FB περιέχουν πρόγραμμα το οποίο εκτελείται

κάθε φορά που τα FB καλούνται από άλλο μπλοκ που περιέχει κώδικα. Τα μπλοκ συναρτήσεων (FB) διευκολύνουν τον προγραμματισμό συχνά χρησιμοποιούμενων και σύνθετων συναρτήσεων.



#### ↪ Μπλοκ Δεδομένων DB (Data Blocks)

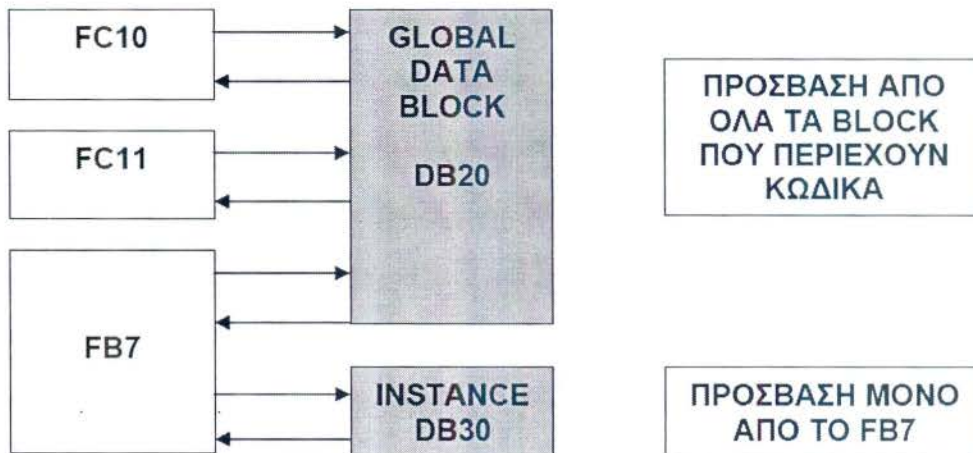
Τα μπλοκ δεδομένων δεν περιέχουν κώδικα, αλλά περιέχουν δεδομένα του προγράμματος μας. Προγραμματίζοντας τα μπλοκ δεδομένων καθορίζουμε σε ποια μορφή θα αποθηκευτούν τα δεδομένα (σε ποια μπλοκ, με ποια σειρά και με ποιο τύπο δεδομένων). Υπάρχουν δύο βασικοί τρόποι χρησιμοποίησης των μπλοκ δεδομένων:

#### ↪ Μπλοκ γενικών δεδομένων (Global data block GD)

Προγραμματίζονται για κοινή χρήση σε όλο το πρόγραμμα. Ένα μπλοκ γενικών δεδομένων είναι, κατά κάποιο τρόπο, ένα «ελεύθερο», μπλοκ μέσα στο πρόγραμμα του χρήστη και δεν εκχωρείται σε κάποιο μπλοκ «κώδικα».

#### ↪ Πρότυπα μπλοκ δεδομένων (Instance data block)

Αντίθετα ένα «instance data block» (πρότυπο μπλοκ δεδομένων) εκχωρείται σ' ένα μπλοκ συνάρτησης (FB) και αποθηκεύει ένα μέρος των τοπικών δεδομένων αυτού του μπλοκ συνάρτησης. Το μέγεθος των DB είναι μεταβαλλόμενο, όσον αφορά το μέγιστο μέγεθος αυτού αυτό εξαρτάται από την χρησιμοποιούμενη CPU. Όταν ένα μπλοκ κώδικα (FC, FB, OB) καλείται, αυτό μπορεί ταυτόχρονα να καταλάβει χώρο μνήμης και στην περιοχή των τοπικών δεδομένων (L-Stack) και υπό μορφή ενός DB. Αντίθετα με τα τοπικά δεδομένα, τα δεδομένα οποία περιέχονται σε ένα DB δεν χάνονται όταν κλείσει το DB ή στο τέλος της επεξεργασίας του μπλοκ που περιέχει κώδικα. Κάθε FB, FC, OB έχει πρόσβαση στο διάβασμα ή γράψιμο ενός DB. Ένα μπλοκ κώδικα έχει την δυνατότητα να ανοίγει ταυτόχρονα ένα global data block και ένα instance data block.



## 2.2.2 Δομή των Μπλόκ

Σε γενικές γραμμές ένα μπλοκ που περιέχει κώδικα αποτελείται από τα εξής μέρη:

- Την κεφαλή του μπλοκ (block header). Αυτό περιλαμβάνει τις ιδιότητες του μπλοκ και το όνομα του
- Την περιοχή των δηλώσεων (declarations) όπου δηλώνονται οι τοπικές μεταβλητές του μπλοκ (Local Variables - L)
- Τέλος την περιοχή η οποία περιλαμβάνει τον κώδικα του χρήστη και τα τυχόν σχόλια.

Στην παρακάτω εικόνα παρουσιάζεται η δομή ενός FC

Contents Of: 'Environment\Interface\IN_OUT'		
Name	Data Type	Comment

Interface

- IN
- OUT
- IN\_OUT**
- TEMP
- RETURN
- RET\_VAL

Δήλωση  
Μεταβλητών

Τα

FC40 : Title:

Comment:

**Network 1** : Title:

Comment:

```

CALL "Start_Stop"
Start_Stop:="RD".V2_Open
ON      :=M30.0
Released :=M30.1
    
```

FC4  
DB102.DBX110.6

-- RD V2 Open Bit

Σχόλια

μπλοκ δεδομένων (DB) είναι και αυτά δομημένα με παρόμοιο τρόπο.

Κώδικας

- ➔ Την κεφαλή του μπλοκ (block header) που περιλαμβάνει τις ιδιότητες του μπλοκ.
- ➔ Την περιοχή των δηλώσεων declarations όπου δηλώνονται οι τοπικές μεταβλητές του μπλοκ (οι διευθύνσεις των δεδομένων και ο τύπος τους).
- ➔ Το τμήμα με τις αρχικές τιμές, τις τιμές δηλαδή που θα έχουν κατά την πρώτη εκκίνηση του συστήματος.



# ΚΕΦΑΛΑΙΟ 3<sup>ο</sup>



S7-300 Software

### 3.1 Simatic Manager



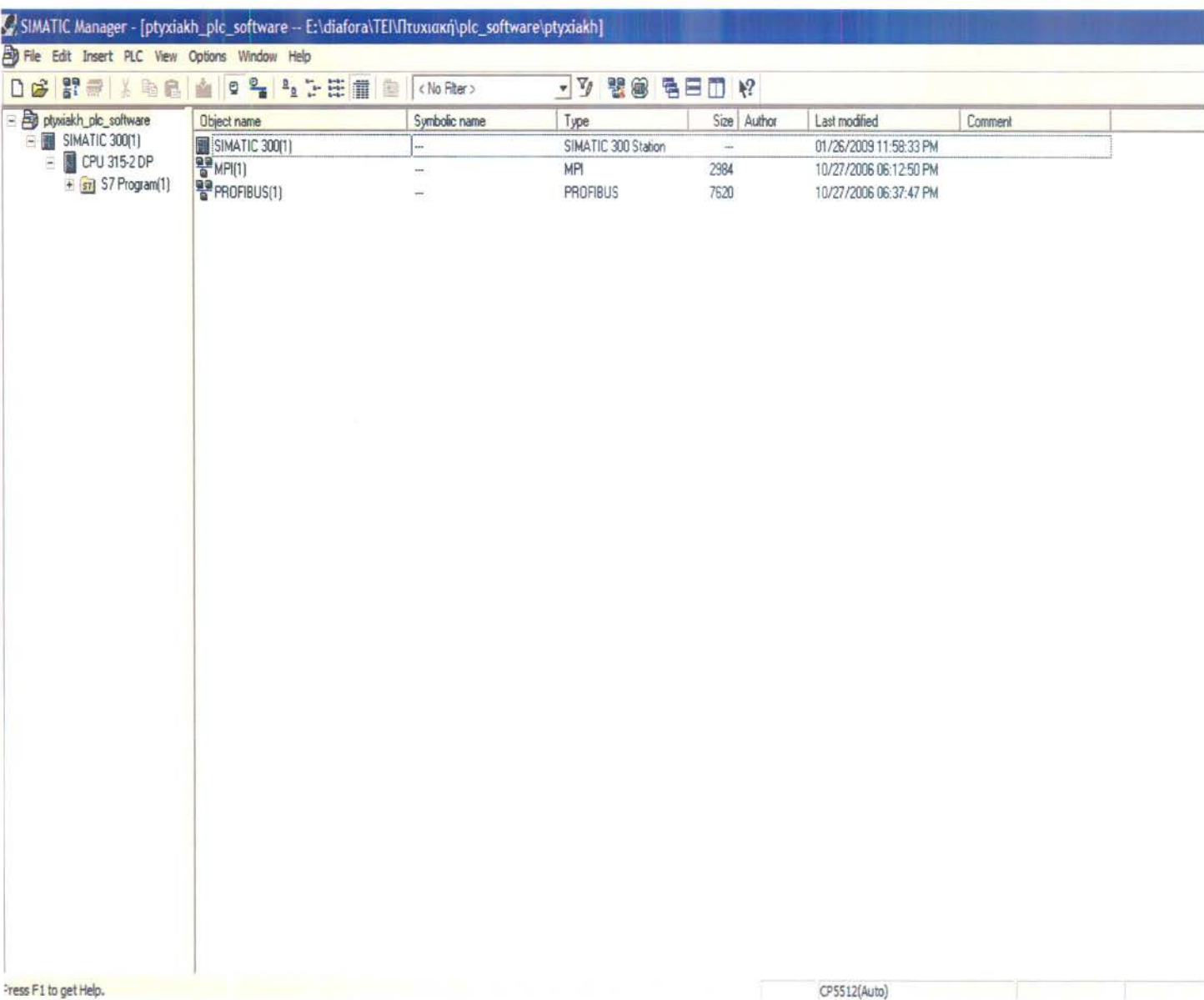
Ο Simatic Manager είναι το κύριο εργαλείο της step 7. Το εικονίδιό του

θα το βρείτε στα windows και μπορείτε να τον ξεκινήσετε κάνοντας διπλό click πάνω σε αυτό.

Μετά από την πρώτη εκτέλεση του προγράμματος εμφανίζεται ο οδηγός δημιουργίας έργων (project wizard), ο οποίος μπορεί να χρησιμοποιηθεί για την απλή δημιουργία νέων έργων (project). Μπορείτε να τον απενεργοποιήσετε τσεκάροντας στο κουτάκι «display wizard on starting simatic manager» αφού μπορείτε να τον καλέσετε, αν το θέλετε, μέσω της εντολής Menu file → New Project Wizard. Ο προγραμματισμός ξεκινά με το άνοιγμα ή τη δημιουργία ενός έργου.

Όταν ανοίγετε το έργο από το Menu File → Open θα δείτε το χωρισμένο παράθυρο του έργου:



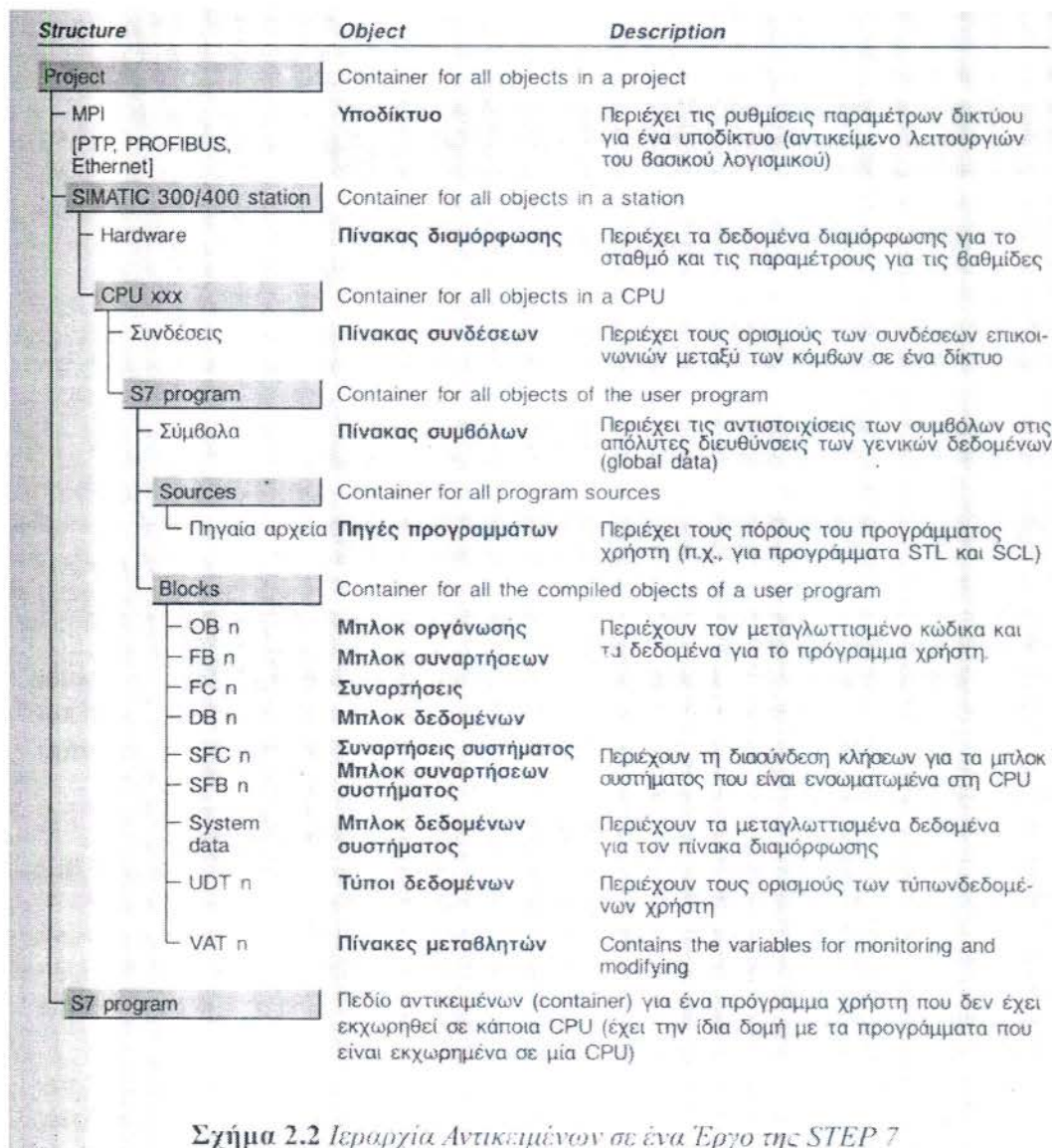


Στα αριστερά βρίσκεται η δομή του ανοιχτού αντικειμένου (ιεραρχία αντικειμένου) και στα δεξιά βρίσκεται το επιλεγμένο αντικείμενο.

Κάνοντας κλικ στο κουτάκι που περιέχει το σύμβολο + στο αριστερο παράθυρο, εμφανίζονται τα επιμέρους επίπεδα της δομής.

Επιλέγοντας ένα αντικείμενο στο αριστερό μέρος του παραθύρου, εμφανίζονται τα περιεχόμενά του στο δεξί μέρος.

Στο περιβάλλον του Simatic Manager εργαζόμαστε με τα αντικείμενα της Step7. Αυτά τα “λογικά” αντικείμενα αντιστοιχούν στα “πραγματικά” αντικείμενα του σχεδίου μας. Ένα έργο περιέχει ολόκληρο το σχέδιο και ένας σταθμός αντιστοιχεί σε έναν προγραμματιζόμενο ελεγκτή. Ένα έργο μπορεί να περιλαμβάνει πολλούς σταθμούς συνδεδεμένους μεταξύ τους, για παράδειγμα μέσω ενός υποδικτύου MPI. Ένας σταθμός περιέχει μία CPU και η CPU περιέχει ένα πρόγραμμα, εν προκειμένω ένα πρόγραμμα s7. Αυτό το πρόγραμμα, με τη σειρά του, περιέχει αντικείμενα, όπως το αντικείμενο Blocks το οποίο περιέχει μεταξύ άλλων τα μεταγλωτισμένα Block.



Σχήμα 2.2 Ιεραρχία Αντικειμένων σε ένα Έργο της STEP 7

Τα αντικείμενα της step7 συνδέονται μεταξύ τους μέσω μιας δενδρικής δομής. Το παραπάνω σχήμα απεικονίζει τα πιο σημαντικά τμήματα της δενδρικής δομής (κύριος κλάδος) όταν εργαζόμαστε με το βασικό πακέτο της step7 για εφαρμογές s7 σε offline προβολή. Τα πεδία αντικειμένων (με τους έντονους χαρακτήρες) ή container περιέχουν άλλα αντικείμενα. Όλα τα αντικείμενα του σχεδίου είναι διαθέσιμα στη offline προβολή και βρίσκονται στον σκληρό δίσκο της συσκευής προγραμματισμού. Αν η συσκευή είναι συνδεδεμένη online σε μια CPU ( συνήθως σε σύστημα PLC), μπορείται να κάνετε μεταγωγή σε ONLINE προβολή επιλέγοντας View → Online. Αυτή η επιλογή εμφανίζει ένα ακόμη παράθυρο του έργου το οποίο περιέχει τα αντικείμενα της συσκευής προορισμού στο οποίο όμως, δεν περιέχονται τα αντικείμενα με τους πλαγιαστούς χαρακτήρες.

Από την γραμμή τίτλου του ενεργού παραθύρου του έργου μπορούμε να δούμε αν εργαζόμαστε σε offline ή online προβολή. Για καλύτερη διαφοροποίηση, μπορούμε να ορίσουμε διαφορετικό χρώμα στη γραμμή τίτλου και στον τίτλο παραθύρου απο αυτό στο offline παράθυρο.

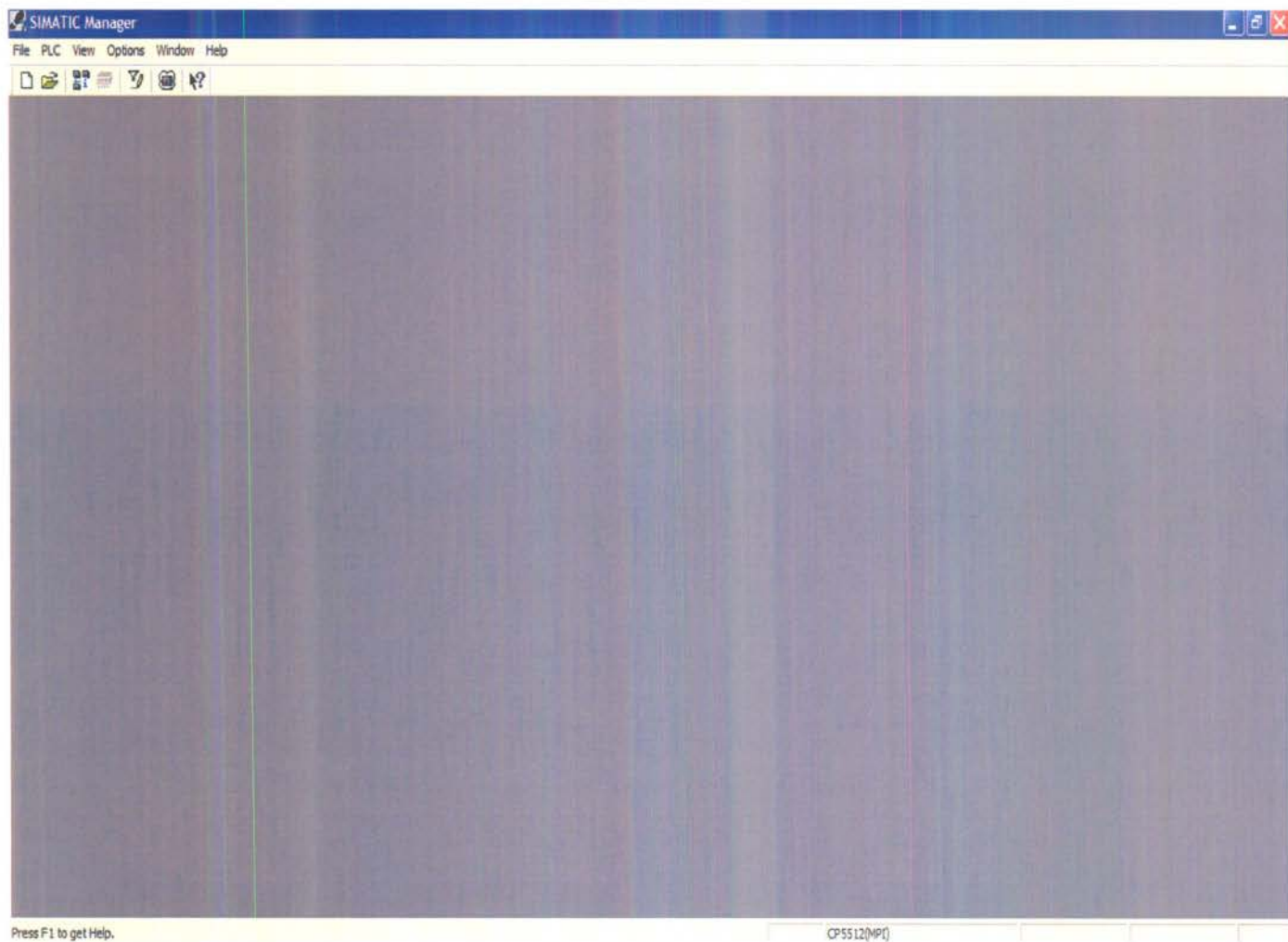
Απο την επιλογή του Menu Options →Customize, μπορούμε να αλλάξουμε τις βασικές ρυθμίσεις του Simatic Manager, όπως η γλώσσα, το αρχείο προγράμματος και η θέση αποθήκευσης των έργων και των βιβλιοθηκών, καθώς και να διανορφώσουμε το αρχείο προγράμματος.



### 3.1.1 Περιβάλλον εργασίας

Διπλό κλικ στον Simatic Manager

Αρχική οθόνη

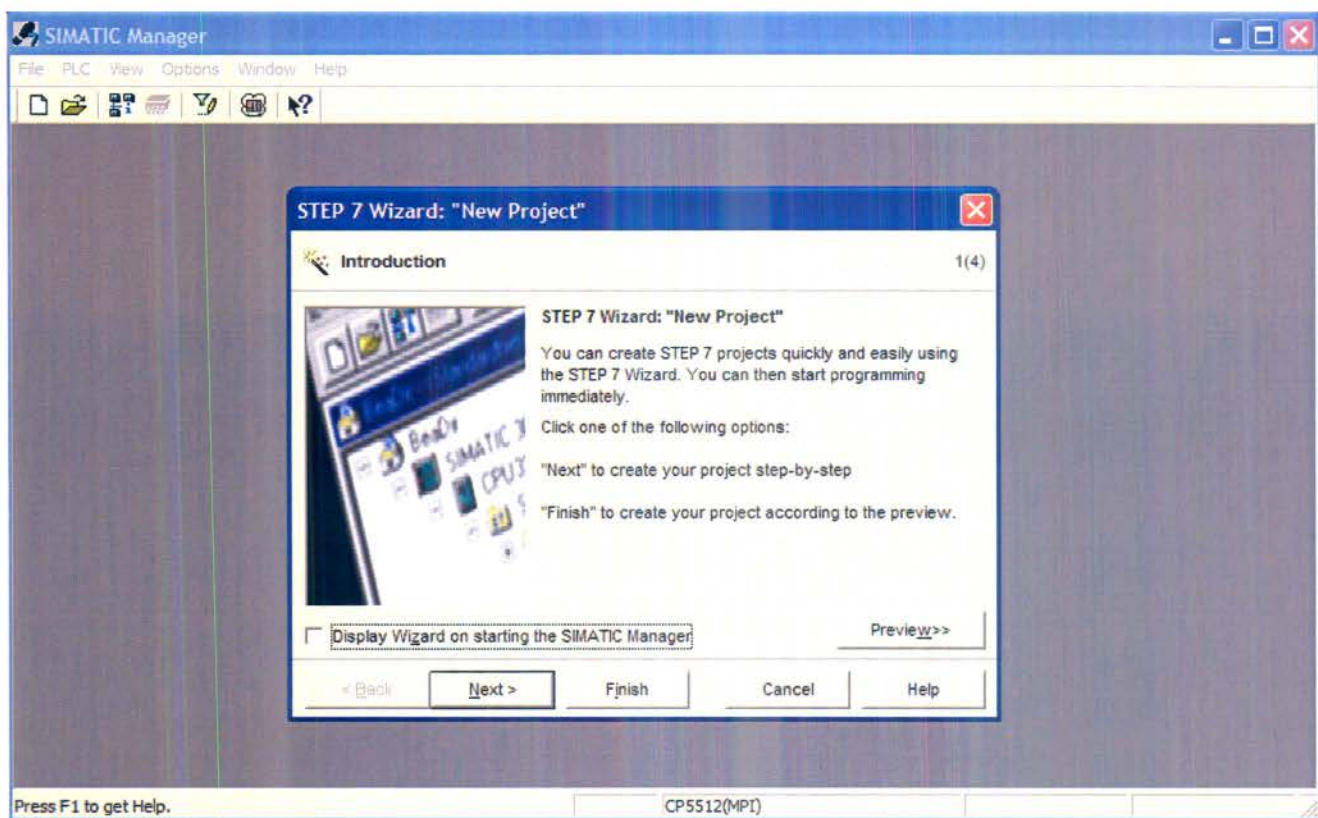




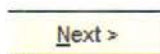
### 3.1.2 Δημιουργία Project

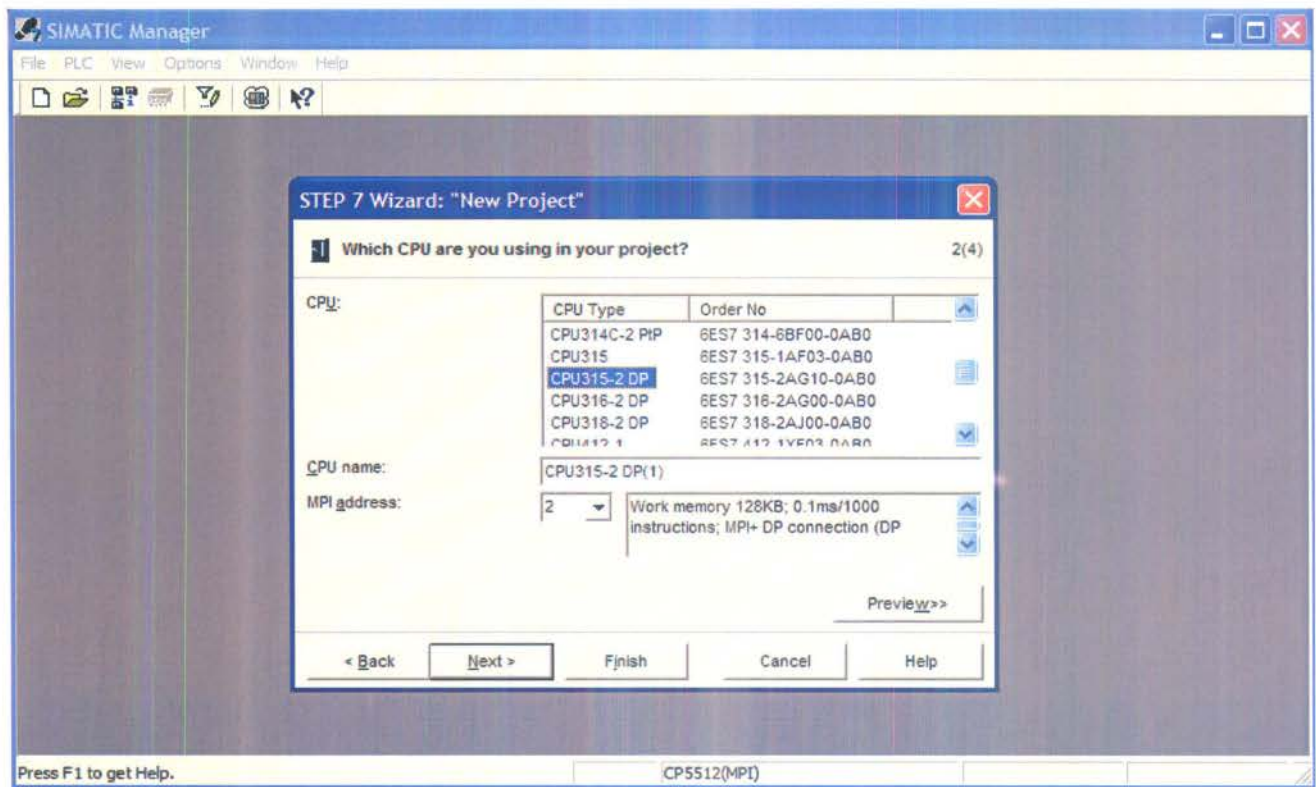
Υπάρχουν 2 τρόποι για να δημιουργήσουμε ένα Project.

Ο πρώτος (ενδεικνύεται για αρχάριους χρήστες), είναι με τον οδηγό δημιουργίας ενός Project του Simatic Manager, που ονομάζεται *New Project Wizard*.



Κλικ

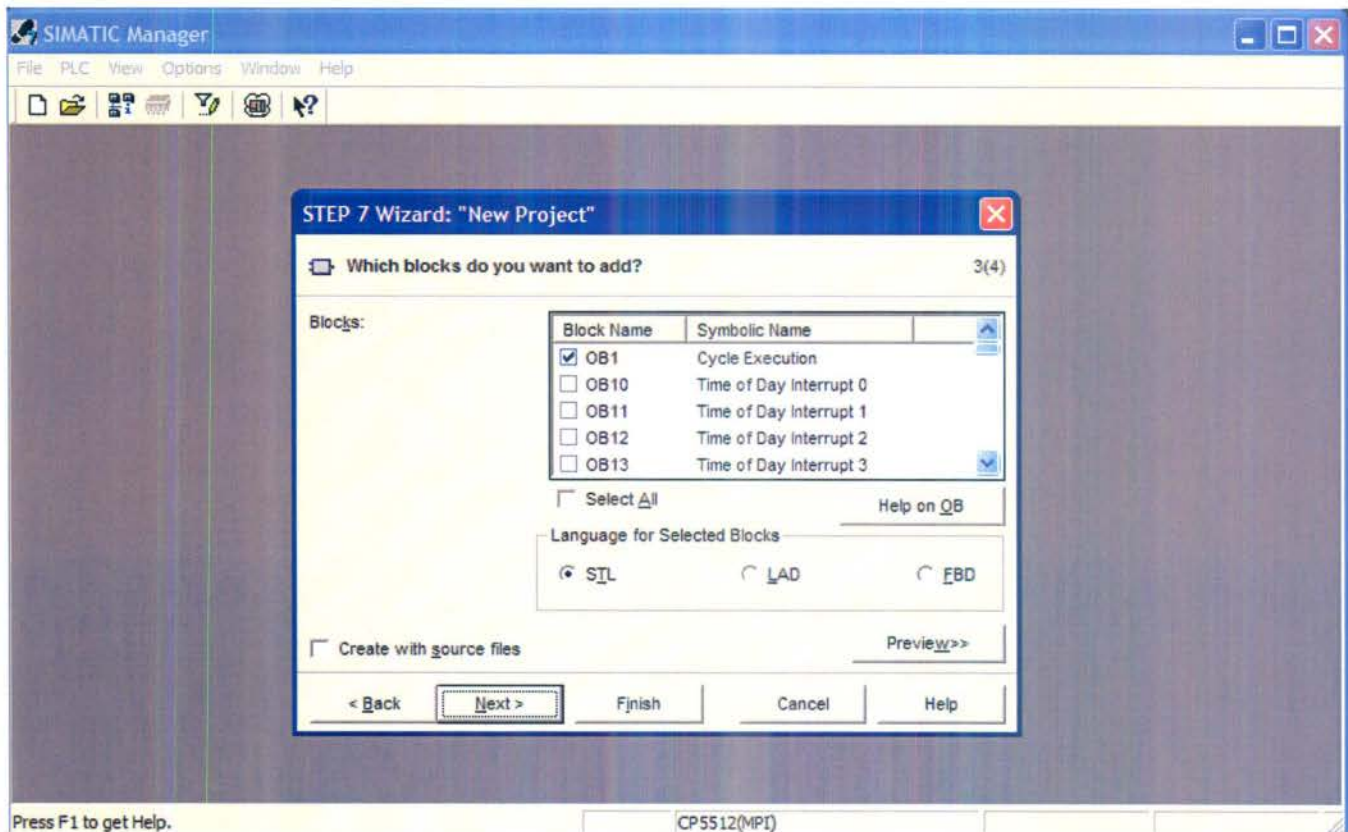




επιλέγουμε την CPU του σταθμού.

Next >

στη συνέχεια επιλέγουμε τα OB (Organization Blocks), που θα χρησιμοποιηθούν στο Project.



Παρατηρείτε πως το OB1 είναι προεπιλεγμένο, αφού αυτό είναι που τρέχει το εκτελέσιμο πρόγραμμα.

(Cycle Execution, Εκτελέσιμος κύκλος προγράμματος)

Τα υπόλοιπα OB είναι περισσότερο blocks ασφαλείας, όπως τα:

- OB10-17 Time of Day Interrupt
- OB20-23 Time Delay Interrupt
- OB30-38 Cyclic Interrupt
- OB40-47 Hardware Interrupt
- OB80 Cycle Time Fault
- OB81 Power Supply Fault
- OB82-83 I/O Point Fault
- OB84 CPU Fault
- OB 85 OB Not Loaded Fault
- OB 86 Loss of Rack Fault
- OB 87 Communication Fault
- OB 100 Complete Restart
- OB 101 Restart
- OB 102 Cold Restart
- OB 121 Programming Error
- OB 122 Module Access Error

με τα σημαντικότερα εξ αυτών να είναι από το OB80 έως το OB122.

Στο ίδιο παράθυρο επιλέγουμε και τη γλώσσα προγραμματισμού που θα χρησιμοποιήσουμε STL / LAD / FBD.

- STL: λίστα εντολών, μοιάζει με γλώσσα Assembly
- LAD: κλιμακωτή λογική, μοιάζει με τα λογικά διαγράμματα ρελέ
- FD: Μπλοκ διάγραμμα λειτουργίας, function block diagram

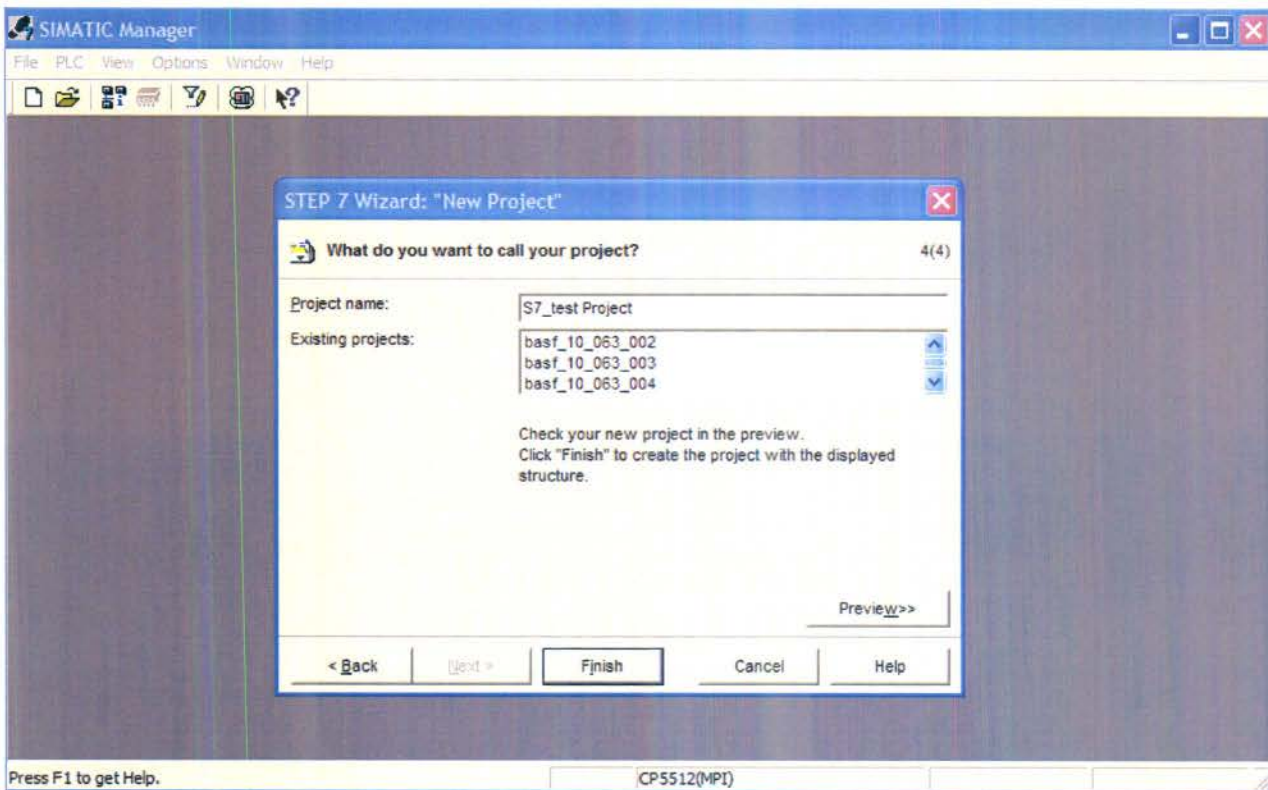




Πέρα από τις παραπάνω γλώσσες προγραμματισμού υπάρχουν και επιπλέον προαιρετικά πακέτα όπως η SCL (υψηλή γλώσσα προγραμματισμού παρόμοια με την pascal), το S7-GRAPH (ακολουθιακός έλεγχος), το S7-HiGRAPH (προγραμματισμός με διαγράμματα καταστάσεων) και η CFC (σύνδεση μπλοκ, παρόμοια με το μπλοκ διάγραμμα λειτουργίας).

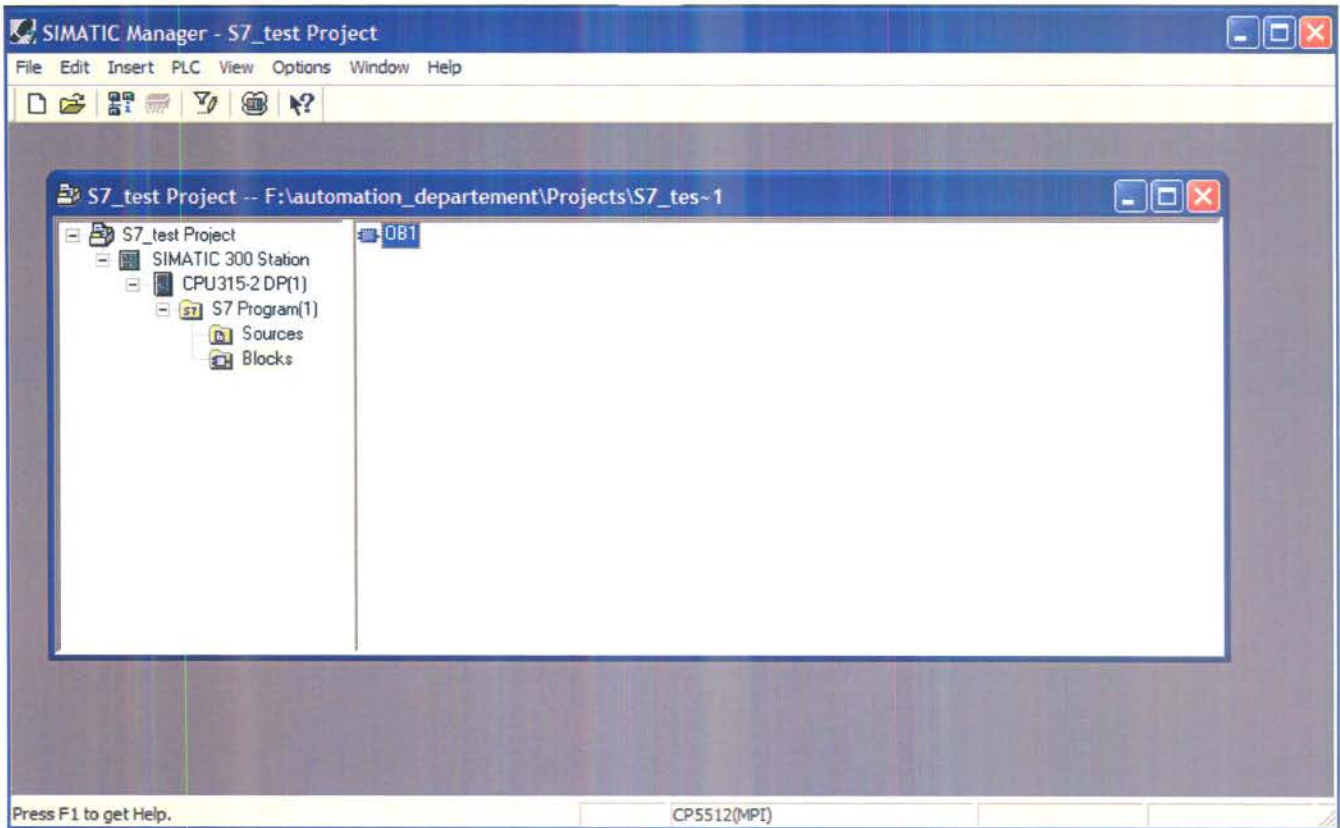
*Οι πιο διαδεδομένες γλώσσες σήμερα είναι η STL και η SCL.*

Στη συνέχεια δίνουμε ένα όνομα στο Project



κλικ στο





και το Project δημιουργήθηκε.

### 3.1.2 Είδη προγραμματισμού

#### ❖ Άμεσος Προγραμματισμός

Επιλέγοντας στον Simatic Manager το αντικείμενο Blocks στο αριστερό μέρος του παραθύρου του project και δημιουργώντας ένα νέο μπλοκ με την επιλογή Insert - S7 BLOCK - ...

Τότε θα εμφανιστεί ένα παράθυρο διαλόγου με την επικεφαλίδα του μπλοκ (αριθμός μπλοκ, γλώσσα, ιδιότητες)

#### ❖ Source Oriented Προγραμματισμός

Εκινώντας ένα πρόγραμμα με την μέθοδο Source Oriented προγραμματισμού πρέπει πρώτα να δημιουργήσετε ένα άδειο πηγαίο αρχείο προγράμματος στον Simatic Manager.

Εκινάτε τον συντάκτη ανοίγοντας το πηγαίο αρχείο προγράμματος και αρχίζοντας να γράφεται το πρόγραμμα αμέσως, πχ, με μια δεσμευμένη λέξη για ένα μπλοκ συνάρτησης.

Ο Source Oriented προγραμματισμός υπερτερεί σε σχέση με τον άμεσο, αφού μπορείς να κάνεις copy κάποιο source και να το χρησιμοποιήσεις όπως είναι, σε ένα άλλο project, ειδικότερα αν χρησιμοποιούμε συμβολικές ονομασίες και έχουμε δημιουργήσει κάποια standards σε αυτές.

Το να επαναλαμβάνονται κάποιες διαδικασίες σε διάφορα project στη βιομηχανία, είναι ένα πολύ συχνό φαινόμενο, που με τον Source Oriented προγραμματισμό, γλιτώνουμε πολύτιμο χρόνο προγραμματισμού.



# ΚΕΦΑΛΑΙΟ 4<sup>ο</sup>



Μετατροπείς συχνότητας - Inverters

#### 4.1 Γενικά περί των μετατροπέων συχνότητας

Για την αποδοτική ρύθμιση των στροφών του ασύγχρονου κινητήρα σε ευρεία όρια λειτουργίας, απαιτείται η ταυτόχρονη μεταβολή της τάσης και της συχνότητας τροφοδοσίας, κατά ένα συγκεκριμένο τρόπο. Αυτό επιτυγχάνεται μέσω ειδικών διατάξεων, οι οποίες καλούνται μετατροπείς συχνότητας. Μια χονδρική διάκριση των μετατροπέων αυτών είναι στους στρεφόμενους και τους στατούς.

Οι στρεφόμενοι μετατροπείς συχνότητας, οι οποίοι δεν είναι τίποτε άλλο παρά ένα ζεύγος κινητήρα σύγχρονης γεννήτριας, αποτελούν προϊόν παλαιάς τεχνολογίας και χρησιμοποιούνταν κατά κόρων πριν μερικές δεκαετίες.

Οι μετατροπείς αυτοί, είχαν μεγάλο κόστος, όγκο, χρόνο συντήρησης, αλλά το πιο σημαντικό μειονέκτημά τους ήταν πως τόσο η συχνότητα όσο και το μέγεθος της τάσης εξόδου της σύγχρονης γεννήτριας, επηρεάζονταν σημαντικά από το φορτίο.

Τη λύση σε αυτά τα προβλήματα ήρθαν να δώσουν οι στατοί μετατροπείς συχνότητας, οι οποίοι αποτελούν αντικείμενο νέας τεχνολογίας και είναι προϊόντα τεχνολογίας των Ηλεκτρονικών Ισχύος.

Τα κύρια διακοπτικά στοιχεία των μετατροπέων αυτών, είναι στοιχεία στερεάς κατάστασης (θυρίστορ, GTO, BTJ, MOSFET, IGBT κλπ), δηλαδή ταχύτατοι ηλεκτρονικοί διακόπτες, με υψηλό βαθμό απόδοσης. Το κόστος αλλά και ο όγκος των εν λόγω μετατροπέων, είναι σημαντικά μικρότερα των αντίστοιχων στρεφόμενων. Το σημαντικότερο όμως πλεονέκτημά τους είναι ότι, τόσο η συχνότητα όσο και το μέγεθος της τάσης εξόδου, δεν επηρεάζονται από το είδος και το μέγεθος του φορτίου.

Οι βασικοί τύποι των στατών μετατροπέων είναι δύο, ο μετατροπέας E.P/Σ.P/E.P (DC linked inverter) και ο κυκλομετατροπέας (cyclonverter).

Η χρήση των τελευταίων στις συνήθεις βιομηχανικές εφαρμογές, είναι περιορισμένη αφού μειονεκτούν έναντι των μετατροπέων E.P/Σ.P/E.P, λόγω του ότι η συχνότητα της τάσης εξόδου, πρέπει πρακτικά να είναι μικρότερη από το 1/3 της συχνότητας εισόδου.

#### 4.2 Μετατροπείς E.P/Σ.P/E.P ή Ρυθμιστές Στροφών Ασύγχρονων κινητήρων

Ο ανορθωτής μπορεί να είναι ελεγχόμενος ή μη, μονοφασικός ή τριφασικός, ανάλογα με το είδος και την ισχύ του μετατροπέα. Στις περισσότερες περιπτώσεις, είναι τριφασικός πλήρως ελεγχόμενος 6 παλμών. Ο αντιστροφέας, αποτελείται από 6 βασικά διακοπτικά στοιχεία. Με κατάλληλη παλμοδότηση των στοιχείων αυτών, η συνεχής τάση εισόδου μετατρέπεται στην έξοδο σε μια τριφασική εναλλασσόμενη μη ημιτονοειδή τάση.

Η μορφή της τάσης εξόδου, ποικίλει ανάλογα με την τεχνική του κυκλώματος παλμοδότησης. Με διάφορες δε χρησιμοποιούμενες μεθόδους εξάλειψης των αρμονικών, είναι δυνατόν να πλησιάσει την ιδανική ημιτονοειδή μορφή.

Η συχνότητα εξόδου στους συνήθεις μετατροπείς κυμαίνεται από 2-3 Hz, μέχρι και μερικές δεκάδες Hz.

Με πιο απλά λόγια, ο ρυθμιστής στροφών είναι ένας οδηγός ο οποίος μπορεί να μεταβάλει τις στροφές ενός ασύγχρονου κινητήρα (σ.σ. οποιοσδήποτε ασύγχρονος κινητήρας που τροφοδοτείται από την ονομαστική τάση και

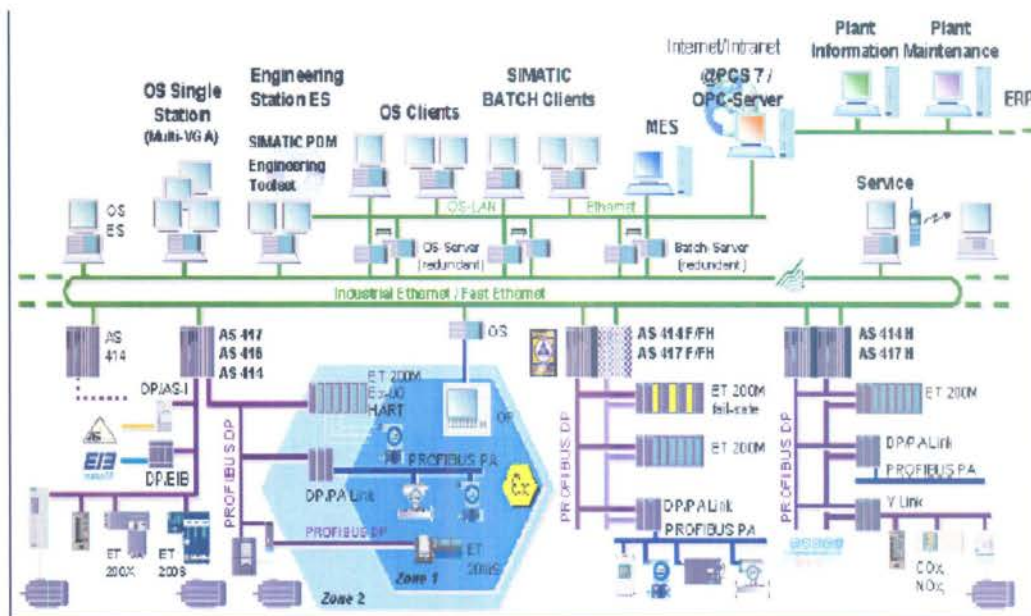
ένταση ηλεκτρικού ρεύματος, θα δουλεύει με τις ονομαστικές του στροφές - συνήθως 1450 σ/λεπτό).



Ο ρυθμιστής στροφών μπορεί να μεταβάλει κατά το δοκούν τις τιμές της τροφοδοσίας του κινητήρα με αποτέλεσμα να αλλάζουν οι στρόφες/λειπό, ο χρόνος επιτάχυνσης από ηρεμία στις ονομαστικές στρόφες, ο χρόνος επιβράδυνσης από ονομαστικές στρόφες σε ηρεμία και άλλα.

Πως το κάνει αυτό? Πολύ απλά εξομοιώνει την εναλλασσόμενη τάση τροφοδοσίας με μια σειρά από στιγμιαίες τιμές τάσης DC- οι οποίες προοδευτικά αλλάζουν σε πλάτος και φάση (+ / - από τον άξονα κλπ).

# ΚΕΦΑΛΑΙΟ 5<sup>ο</sup>



## Επικοινωνίες



## 5.1 Βασικές έννοιες

### 5.2 Δίκτυο

Το δίκτυο είναι η σύνδεση διαφόρων συσκευών μεταξύ τους με σκοπό την επικοινωνία. Αποτελείται από ένα ή περισσότερα ίδια ή διαφορετικά υποδίκτυα (subnet) συνδεδεμένα μεταξύ τους.

### 5.3 Υποδίκτυο

Σε ένα υποδίκτυο όλοι οι κόμβοι επικοινωνίας διασυνδέονται μέσω Hardware συνδέσεων με ομογενή φυσικά χαρακτηριστικά και παραμέτρους μετάδοσης, όπως είναι ο ρυθμός μεταφοράς δεδομένων, όπου ανταλλάσσουν δεδομένα μέσω μιας κοινής διαδικασίας μετάδοσης.

### 5.4 Simatic Communications

Οι επικοινωνίες - ανταλλαγή δεδομένων μεταξύ προγραμματιζόμενων βαθμίδων - είναι αναπόσπαστο τμήμα των συστημάτων SIMATIC-S7.

Σχεδόν όλες οι επικοινωνιακές λειτουργίες κατευθύνονται από το λειτουργικό σύστημα.

Μπορείτε να ανταλλάξετε δεδομένα χωρίς την προσθήκη ξεχωριστού Hardware και μόνο με ένα καλώδιο σύνδεσης μεταξύ 2 CPU.

Αν χρησιμοποιείται βαθμίδες CP (επεξεργαστών επικοινωνιών) μπορείτε να επιτύχετε ισχυρούς δικτυακούς συνδέσμους και την ευκολία να συνδέεστε και με συσκευές που δεν είναι Siemens.

Ο όρος somatic net (δίκτυο Simatic), είναι ο γενικός όρος που καλύπτει τις επικοινωνίες των Simatic και απεικονίζει την ανταλλαγή πληροφοριών μεταξύ τους, αλλά και μεταξύ προγραμματιζόμενων ελεγκτών και συσκευών HMI.

Υπάρχουν διάφορες υλοποιήσεις επικοινωνιών διαθέσιμες ανάλογα με τις απαιτήσεις σε απόδοση.

Το σύστημα Simatic αναγνωρίζει τις συνδέσεις MPI, PROFIBUS, Industrial ETHERNET, και την «από σημείο σε σημείο» σύνδεση (point-to-point, PTP), υποδίκτυα.

## 5.5 Είδη υποδικτύων

Τα υποδίκτυα είναι επικοινωνιακές διαδρομές (οδοί) με τα ίδια φυσικά χαρακτηριστικά και τις ίδιες επικοινωνιακές διαδικασίες τα οποία αποτελούν για τον διαχειριστή Simatic τα βασικά στοιχεία επικοινωνίας.

Η διαφορά μεταξύ των υποδικτύων έγκειται στην απόδοσή τους.

### ➤ MPI

Αποτελεί μία χαμηλού κόστους μέθοδο για τη δικτύωση μερικών συσκευών Simatic με μικρού όγκου δεδομένων. Στην οικογένεια Simatic κάθε CPU έχει την δυνατότητα πολυκομβικής διασύνδεσης (multipoint interface, MPI), η οποία επιτρέπει τη σύσταση υποδικτύων στα οποία αναλαμβάνει χώρα

ανταλλαγή δεδομένων μεταξύ συσκευών, όπως για παράδειγμα, CPU, HMI, και συσκευές προγραμματισμού.

Η ανταλλαγή δεδομένων διαχειρίζεται από ένα κλειστό πρωτόκολλο επικοινωνίας της Siemens.

Ως μέσο μετάδοσης σε ένα δίκτυο MPI χρησιμοποιείται ή μονωμένο καλώδιο ενός ζεύγους ή καλώδιο οπτικής ίνας από γυαλί ή πλαστικό.

Το μήκος του καλωδίου σε ένα τμήμα διαύλου μπορεί να έχει μήκος μέχρι και 50 μέτρα, το οποίο όμως μπορεί να αυξηθεί με τη χρήση ενισχυτών RS485 (μέχρι και 1100 μέτρα) ή με βαθμίδες διασύνδεσης οπτικής ίνας (μέχρι και 100Km). Ο ρυθμός μεταφοράς δεδομένων είναι συνήθως 187.5Kbits/sec.

Ο μέγιστος αριθμός κόμβων είναι 32. Κάθε κόμβος έχει πρόσβαση στο δίαυλο για συγκεκριμένο χρονικό διάστημα κατά το οποίο μπορεί να στείλει πακέτα δεδομένων. Με την πάροδο του χρονικού αυτού διαστήματος, τα δικαιώματα πρόσβασης περνάνε στον επόμενο κόμβο (διαδικασία προσπέλασης).

#### ➤ ProfibusDp

Πρόκειται για ένα από τους πιο διαδεδομένους διαύλους επικοινωνίας, αφού πρόκειται για ένα ανοιχτό πρωτόκολλο το οποίο υποστηρίζεται από τους περισσότερους κατασκευαστές, και έτσι οι επικοινωνίες γίνονται εύκολα και γρήγορα χωρίς να παρεμβάλλονται έξτρα κάρτες κλπ.

Ο όρος **Profibus** είναι η σύντμηση των λέξεων του όρου «Process Fieldbus» που σε ελληνική μετάφραση σημαίνει «επεξεργασία διαύλου πεδίου» και είναι ένα πρότυπο παγκόσμιας αποδοχής σύμφωνα με το πρότυπο EN50170 για τη δικτύωση συσκευών πεδίου.

Ως μέσο μετάδοσης χρησιμοποιείται μονωμένο καλώδιο ενός ζεύγους ή καλώδιο οπτικής ίνας από γυαλί ή πλαστικό. Το μήκος του καλωδίου σε ένα τμήμα διαύλου εξαρτάται από τον απαιτούμενο ρυθμό μεταφοράς δεδομένων. Έτσι το μέγιστο μήκος για ρυθμούς μεταφοράς της τάξεως των 12Mbits/sec είναι 100 μέτρα, ενώ για ρυθμούς μεταφοράς της τάξεως των 9,6Kbits/sec είναι 1000 μέτρα.

Η εμβέλεια του δικτύου μπορεί να αυξηθεί με τη χρήση ενισχυτών ή με βαθμίδες διασύνδεσης οπτικής ίνας.

Ο μέγιστος αριθμός κόμβων είναι 127 και χωρίζονται σε 2 κατηγορίες, στους ενεργητικούς και στους παθητικούς κόμβους.

Ένας ενεργητικός κόμβος έχει δικαιώματα πρόσβασης στο δίαυλο για συγκεκριμένο χρονικό διάστημα κατά το οποίο μπορεί να στείλει πακέτα δεδομένων. Με την πάροδο του χρονικού αυτού διαστήματος, τα δικαιώματα πρόσβασης περνάνε στον επόμενο κόμβο (διαδικασία προσπέλασης).

Αν σε έναν ενεργητικό κόμβο έχουν εκχωρηθεί παθητικοί κόμβοι (εξαρτημένοι), ο κύριος εκτελεί τις ανταλλαγές δεδομένων με τους εκχωρημένους σε αυτόν εξαρτημένους κόμβους κατά το χρονικό διάστημα όμως, που έχει τα δικαιώματα πρόσβασης.

Τέλος, να σημειωθεί ότι ένας παθητικός κόμβος δεν μπορεί να πάρει δικαιώματα πρόσβασης.

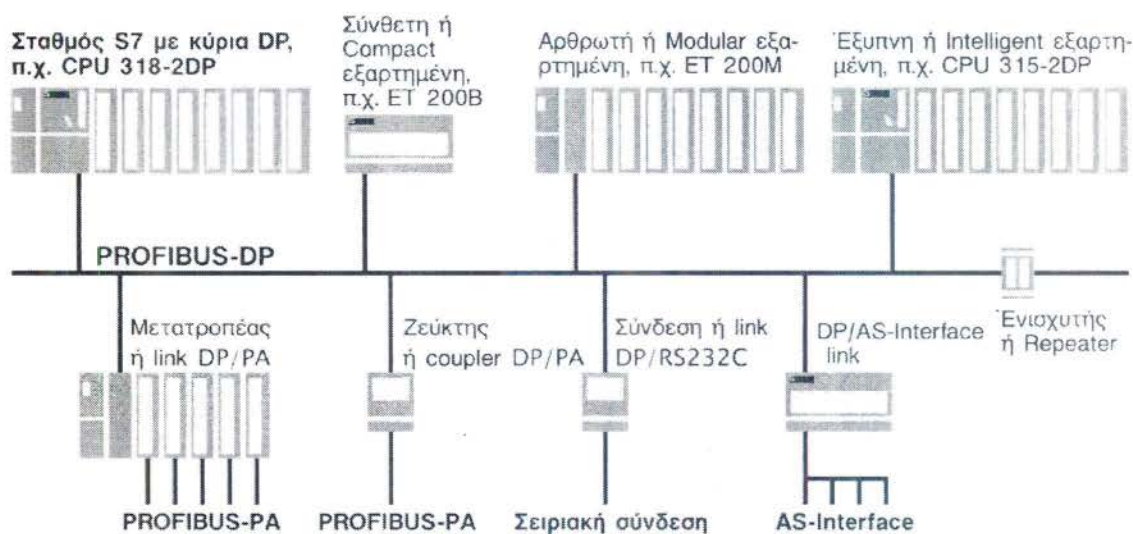
Η υλοποίηση σύνδεσης κατανεμημένων I/O γίνεται μέσω ενός δικτύου Profibus, γεγονός που υποδηλώνει τη χρήση της υπηρεσίας επικοινωνιών Profibus - DP.

Μπορείτε να χρησιμοποιήσετε CPU με ενσωματωμένες ή πρόσθετες κύριες βαθμίδες DP ή επεξεργαστές επικοινωνιών CP.

Μπορείτε επίσης, σε αυτό το δίκτυο να χρησιμοποιήσετε επικοινωνίες SFC εσωτερικές ή επικοινωνίες SFB.

Η μεταφορά δεδομένων μπορεί να γίνει με τις υπηρεσίες Profibus - FMS και Profibus - FDL , χρησιμοποιώντας τους κατάλληλους CP.

Υπάρχουν κάποια διαθέσιμα Block (διασύνδεση FMS ή διασύνδεση SEND/RECEIVE) όπως η διασύνδεση για το πρόγραμμα χρήστη.



### ➤ Δίκτυο Βιομηχανικού (Industrial) Ethernet

Το βιομηχανικό Ethernet είναι το υποδίκτυο που χρησιμοποιείται για τη σύνδεση υπολογιστών και προγραμματιζόμενων ελεγκτών, με έμφαση στο βιομηχανικό τομέα, όπως ορίζεται από το διεθνές πρότυπο IEEE 802.3.

Η ηλεκτρική φυσική σύνδεση είναι ένα διπλής μόνωσης ομοαξονικό καλώδιο ενός ζεύγους, και η οπτική σύνδεση είναι ένα καλώδιο οπτικής ίνας από γυαλί.

Σε ένα ηλεκτρικό δίκτυο, η εμβέλεια είναι 1.5km και σε ένα δίκτυο οπτικής ίνας είναι 4,5Km.

Ο ρυθμός μεταφοράς δεδομένων είναι της τάξεως των 10Mbits/s.

Ο αριθμός των κόμβων που μπορούν να συνδεθούν με το βιομηχανικό Ethernet είναι πάνω από 1000. Όλοι οι κόμβοι έχουν τα ίδια δικαιώματα πρόσβασης.

➤ Δίκτυο σύνδεσης «από σημείο σε σημείο»

Μια σύνδεση «από σημείο σε σημείο» επιτρέπει την ανταλλαγή δεδομένων μέσω μιας σειριακής διασύνδεσης .

Το μέσο μετάδοσης είναι ένα ηλεκτρικό καλώδιο ο τύπος του οποίου εξαρτάται από την διασύνδεση που θα επιλεγεί. Οι διαθέσιμες συνδέσεις είναι η RS232C(V.24), η 20mA (TTY), και η RS422/485 στα 19.2Kbits/s.

➤ Δίκτυο AS-Διασύνδεσης

Το δίκτυο AS-Διασύνδεσης καλύπτει δικτυακά τους κατάλληλα σχεδιασμένους δυαδικούς αισθητήρες και ενεργοποιητές σύμφωνα με τις προδιαγραφές της AS-Διασύνδεσης IEC TG178.

Το μέσο μετάδοσης είναι ένα μη μονωμένο καλώδιο ενός ζεύγους που παρέχει στους ενεργοποιητές και στους αισθητήρες , όχι μόνο δεδομένα αλλά και τροφοδοσία. Η εμβέλεια του δικτύου μπορεί να φτάσει μέχρι και τα 300 μέτρα με ενισχυτές. Ο ρυθμός μεταφοράς δεδομένων είναι 167Kbits/s.

Μια κύρια συσκευή AS-I ελέγχει μέχρι και 31 εξαρτημένες μέσω κυκλικής σάρωσης, γεγονός που εγγυάται καθορισμένη χρονική απόκριση.



# ΚΕΦΑΛΑΙΟ 6<sup>ο</sup>



Scada WinCC

## 6.1 Ιστορική αναδρομή

Ταυτόχρονα με τα PLC εμφανίστηκαν στην αγορά και τα συστήματα Scada (Supervisory control and data acquisition). Τα συστήματα αυτά αναπτύχθηκαν με κύριο σκοπό την συλλογή δεδομένων από απόμακρα κέντρα και την διάθεσή τους σε κεντρικά σημεία για πληροφόρηση και έλεγχο. Τα SCADA είχαν αρκετά εκτεταμένη δυνατότητα συλλογής, καταγραφής και απεικόνισης πληροφοριών αλλά υστερούσαν στη δυνατότητα της επεξεργασίας τους για την διαχείριση της διαδικασίας μιας συνολικής παραγωγής.

Η αλματώδης ανάπτυξη των μικροϋπολογιστών και γενικότερα της τεχνολογίας επικοινωνίας και αυτοματισμού έφερε και μεγάλη εξέλιξη στις δυνατότητες των PLC.

Τα εξελιγμένα συστήματα PLC/SCADA, έχουν σήμερα την δυνατότητα να διαχειρίζονται αξιόπιστα μεγάλη ποσότητα πληροφοριών, είναι ιδιαίτερος ευέλικτα, ταχύτατα, έχουν τη δυνατότητα επικοινωνίας και διασύνδεσης μεταξύ τους και με μεγάλη ποικιλία δικτύων και συσκευών.

Τα συστήματα βασισμένα σε PLC και SCADA είναι προσανατολισμένα στην ταχύτατη περισυλλογή και επεξεργασία στοιχείων και πληροφοριών με το κέντρο του κόσμου να είναι η πληροφορία που συλλέγεται.

Παρόλο που ένα τέτοιο σύστημα μεριμνά και για την παραγωγική διαδικασία, κύριος στόχος είναι η ταχύτητα απόδοσης ενός αποτελέσματος στον χρήστη και όχι ο έλεγχος της συνολικής διαδικασίας της παραγωγής.



Εκτός από την ταχύτητα, η γενική φιλοσοφία των συστημάτων αυτών είναι η απόλυτη ευελιξία στην διασύνδεση τους με αλλά συστήματα για την ανταλλαγή και την διάθεση δεδομένων.

Έτσι παρόλο που ένα σύστημα PLC/SCADA υστερεί στην εξειδίκευση, υπερτερεί στην ταχύτητα και στην ευελιξία της εγκατάστασης.

Λόγω αυτού, η δομή και τα υλικά που χρησιμοποιούνται μπορεί και να διαφέρουν από εγκατάσταση σε εγκατάσταση ανάλογα με τις απαιτήσεις και την ζητούμενη εφαρμογή. Και εδώ όμως η πιο συνήθης δομή είναι ιεραρχική με τοπικά υποσυστήματα περιουλογής και επεξεργασίας δεδομένων που διασυνδέονται σε ένα κεντρικό SCADA το οποίο και αναλαμβάνει την γενικότερη επιτήρηση της εγκατάστασης. Τα τοπικά αυτά υποσυστήματα είναι συνήθως εγκαταστάσεις PLC (ενδεχόμενος με ένα τοπικό SCADA/HMI) που αναλαμβάνουν τόσο την συλλογή και την πρωτογενή επεξεργασία των δεδομένων όσο και τον έλεγχο της τοπικής διαδικασίας.

Ανάμεσα στα πλεονέκτημα του είναι η ταχύτερη «ζωντανή» πληροφόρηση των δεδομένων, γεγονότων και καταστάσεων, η εύκολη διασύνδεση και ενσωμάτωση υφισταμένων συστημάτων αυτοματισμού και ελέγχου και η δυνατότητα του αξιόπιστου έλεγχου της εγκατάστασης από μεγάλες αποστάσεις (remote locations). Η ευελιξία που παρέχεται από τον ελεύθερο προγραμματισμό των PLCs και του SCADA είναι επίσης ένα πλεονέκτημα μιας και μπορεί να παρέχει το απόλυτα επιθυμητό για τον χρήστη αποτέλεσμα.

## 6. 2 Γενικά περί συστημάτων τηλεδιαχείρισης

Έχει οριστικά παρέλθει η εποχή που οι μηχανικοί που ήταν υπεύθυνοι για τη λειτουργία μιας βιομηχανικής εγκατάστασης αντίκριζαν σε καθημερινή βάση τεράστιες κονσόλες χειρισμού με αμέτρητα κομβία και διακόπτες .

Την εποχή εκείνη η κονσόλα ήταν το μέσο αλληλεπίδρασης ανάμεσα στον μηχανικό - χειριστή και στον ηλεκτρομηχανολογικό εξοπλισμό και παρείχε δυνατότητες που περιορίζονταν ως επί το πλείστον στον χειρισμό και στην οπτική ( στιγμιαία ) απεικόνιση λειτουργιών και βλαβών.

Εδώ και 30 χρόνια περίπου οι κονσόλες αυτές έχουν αντικατασταθεί από ηλεκτρονικούς υπολογιστικές και λογισμικά τηλεδιαχείρισης με συνεχώς αυξανόμενες δυνατότητες σε ότι αφορά την εποπτεία και την τηλεδιαχείριση του συνόλου μιας ηλεκτρομηχανολογικής εγκατάστασης.

Τα πλέον διαδεδομένα συστήματα τηλεδιαχείρισης είναι τα λεγόμενα (λογισμικά εποπτικού ελέγχου και ανάκτησης δεδομένων) ή SCADA (Supervisory Control And Data Acquisition). Τα λογισμικά αυτά εγκαθίστανται σε προσωπικούς Η/Υ (δηλ σε ένα απλό PC), συνδέονται δικτυακά με τους λογικούς ελεγκτές (PLC ή RTU) και παρέχουν στους χειριστές τη δυνατότητα πλήρους εποπτείας και ελέγχου του βιομηχανολογικού εξοπλισμού μέσω γραφικών παραστάσεων της εγκατάστασης.

Σε μια τυπική εφαρμογή SCADA, το σύνολο της εγκατάστασης ομαδοποιείται σε ζώνες, ενώ η κάθε ζώνη απεικονίζεται σε ξεχωριστή οθόνη της εφαρμογής. Για παράδειγμα, σε μια εφαρμογή SCADA ενός βιολογικού καθαρισμού θα υπάρχει μια οθόνη στην οποία απεικονίζονται τα μηχανήματα της δεξαμενής αερισμού (αεριστήρες, αναδευτήρες, αντλίες, όργανα μέτρησης κλπ), ενώ σε μια άλλη οθόνη θα βρούμε τον εξοπλισμό της μονάδας χλωρίωσης (δοσομετρικές αντλίες, αναδευτήρες , όργανα μέτρησης κλπ).

Η κατάσταση λειτουργίας των μηχανημάτων απεικονίζεται συνήθως μέσω διαφοροποίησης χρωματισμού. Για παράδειγμα, όταν ο κινητήρας έχει πράσινο χρώμα σημαίνει ότι βρίσκεται σε κατάσταση αυτόματης λειτουργίας ενώ όταν είναι κόκκινος σημαίνει ότι έχει υποστεί βλάβη. Επίσης, στις οθόνες απεικονίζονται και οι μετρήσεις των αναλογικών οργάνων (θερμόμετρα, μανόμετρα, παροχόμετρα, σταθμήμετρα και άλλα).

Επιπλέον ο χειριστής έχει τη δυνατότητα να τηλεχειρίζεται τα μηχανήματα με το πάτημα ενός κουμπιού στο ποντίκι. Ο τηλεχειρισμός περιλαμβάνει το ξεκίνημα, το σταμάτημα, τη θέση σε αυτόματη λειτουργία ή τον ορισμό παραμέτρων αυτόματης λειτουργίας ενός κινητήρα.

Όμως, πέρα της στιγμιαίας και σε πραγματικό χρόνο απεικόνισης της κατάστασης λειτουργίας των μηχανημάτων, το ισχυρότερο εργαλείο που παρέχει μια εφαρμογή SCADA είναι η καταγραφή και διαχείριση δεδομένων.

Η εφαρμογή πραγματοποιεί συνεχή καταγραφή των μετρήσεων και των συμβάντων σε βάση δεδομένων, έτσι ώστε να εξάγονται ιστορικά στοιχεία συμβάντων (συναγερμών, λειτουργιών, χειρισμών και λοιπά) και μετρήσεων σε μορφή πινάκων ή ιστορικών γραφημάτων.

Η ανάπτυξη ενός σωστού λογισμικού εφαρμογής τηλεδιαχείρισης SCADA πραγματοποιείται σε δυο στάδια. Το πρώτο αφορά την ανάλυση και τον σχεδιασμό του συνόλου του συστήματος. Η εργασία αυτή περιλαμβάνει αρχικά την πλήρη καταγραφή των προς έλεγχο μηχανημάτων & οργάνων, και τον σχεδιασμό της διάταξης αυτών στις οθόνες απεικόνισης του SCADA. Στη συνέχεια πραγματοποιείται η ανάλυση των σημάτων που θα απεικονίζονται για το κάθε μηχανήμα.

Τα σήματα αυτά μπορεί να είναι είτε ψηφιακά είτε αναλογικά. Για παράδειγμα, ένας μετρητής παροχής νερού διαθέτει συνήθως ένα αναλογικό

σήμα για τη στιγμιαία μέτρηση της παροχής σε λίτρα/ ώρα, ένα ψηφιακό σήμα για τη μέτρηση της συνολικής παροχής σε λίτρα (το σήμα αυτό έχει τη μορφή παλμού, ο οποίος ισούται για παράδειγμα με 11 λίτρα) και ένα ψηφιακό σήμα που δηλώνει ύπαρξη σφάλματος στο όργανο.

Έπειτα θα πρέπει να γίνει η ανάλυση των λειτουργικών στοιχείων της εφαρμογής, όπως είναι η τηλεχειρισμοί μηχανημάτων και η παραμετροποίηση αυτόματων λειτουργιών των μηχανημάτων.

Ακολουθεί η σημαντικότερη φάση της ανάλυσης, αυτή που αφορά την καταγραφή και επεξεργασία των σημάτων. Στη φάση αυτή πραγματοποιείται ο σχεδιασμός της βάσης δεδομένων των καταγραφόμενων σημάτων (μετρήσεις οργάνων, λειτουργίες μηχανημάτων, συναγερμοί, χειρισμοί κλπ). Η βάση αυτή θα χρησιμοποιηθεί για τη δημιουργία ιστορικών γραφημάτων, πινάκων συναγέρμων και συμβάντων, καθώς και ιστορικών αναφορών.

Το πρώτο στάδιο που περιγράψαμε παραπάνω δεν απαιτεί τη χρήση του λογισμικού ανάπτυξης εφαρμογών SCADA. Αυτό θα απαιτηθεί στη δεύτερη φάση, όπου θα πρέπει να σχεδιαστούν στο πρόγραμμα οι διάφορες οθόνες της εφαρμογής και να προγραμματιστούν όλες οι λειτουργίες που μελετήθηκαν στη φάση της ανάλυσης.

Είναι προφανές ότι η πρώτη φάση πρέπει να υλοποιείται από μηχανολόγο ή ηλεκτρολόγο – μηχανικό με εμπειρία στη λειτουργία ηλεκτρομηχανολογικών εγκαταστάσεων, ενώ δεν απαιτούνται από αυτόν ιδιαίτερες γνώσεις προγραμματισμού. Η δεύτερη φάση απαιτεί γνώσεις προγραμματισμού με εξειδίκευση στις πλατόφορμες ανάπτυξης συστημάτων SCADA. Καταλαβαίνει κανείς ότι η πρώτη και η δεύτερη φάση δεν είναι εύκολο να υλοποιηθούν από τον ίδιο μηχανικό, μια και οι γνώσεις και η εμπειρία που απαιτούνται στις δυο περιπτώσεις είναι τελείως διαφορετικές.

Δυστυχώς, η Ελληνική πραγματικότητα έχει δείξει το αντίθετο, καθώς η απαίτηση για μείωση του κόστους πολλές φορές έχει ως αποτέλεσμα την υλοποίηση ολόκληρης της εφαρμογής SCADA από έναν και μόνο μηχανικό. Έτσι οι μηχανικοί γίνονται προγραμματιστές και οι προγραμματιστές γίνονται μηχανικοί.

### 6.2.1. Το κόστος ενός συστήματος τηλεδιαχείρισης (SCADA)

Μια ιδιαίτερα σημαντική παράμετρος είναι το κόστος ενός συστήματος τηλεδιαχείρισης (υλικό και λογισμικό), το οποίο είναι άμεση συνάρτηση του πλήθους των προς έλεγχο σημάτων και της πολυπλοκότητας του συστήματος. Σε ότι αφορά το υλικό Η/Υ (Hardware) δεν απαιτείται ιδιαίτερη δαπάνη μια και ένας Η/Υ χαμηλής γκάμας διαθέτει επαρκεί υπολογιστική ισχύ για να «τρέξει» τις πλέον σύνθετες εφαρμογές SCADA. Δεν ισχύει όμως το ίδιο και για τη δαπάνη που αφορά το ίδιο το λογισμικό, του οποίου η συνολική τιμή μπορεί να ξεπεράσει τις μερικές δεκάδες χιλιάδες ευρώ όταν πρόκειται για μια μεγάλη βιομηχανία, όπου τα προς έλεγχο δεδομένα (TAGS στην ορολογία των συστημάτων SCADA) ξεπερνούν το 1.000.000. Η δαπάνη αυτή αφορά μόνο την άδεια χρήσης του λογισμικού και δεν περιλαμβάνει την ανάπτυξη του λογισμικού εφαρμογής, το οποίο σε ορισμένες περιπτώσεις απαιτεί εργασίας μελέτης, σχεδιασμού και προγραμματισμού από έμπειρους μηχανικούς και προγραμματιστές, χρονικής διάρκειας πολλών μηνών. Κατά συνέπεια, η εξεύρεση λύσεων για τη μείωση του κόστους (άδεια χρήσης και ανάπτυξη) ενός συστήματος τηλεδιαχείρισης αποτελεί ισχυρό δέλεαρ, τόσο για τις εταιρείες που δραστηριοποιούνται στον χώρο του βιομηχανικού αυτοματισμού και της τηλεδιαχείρισης, όσο και για τις ίδιες τις βιομηχανίες που υιοθετούν τέτοια συστήματα.

Είδαμε ότιο το συνολικό κόστος μιας εφαρμογής SCADA ισούται με το άθροισμα του κόστους αγοράς της άδειας χρήσης του λογισμικού και του κόστους ανάπτυξης της εφαρμογής. Στη μεν περίπτωση της άδειας χρήσης του λογισμικού το κόστος δεν ποικίλλει σημαντικά, μια και οι περισσότεροι οίκοι ανάπτυξης λογισμικών SCADA ακολουθούν κοινές πολιτικές κοστολόγησης. Στη περίπτωση όμως της ανάπτυξης λογισμικού εφαρμογής τα πράγματα διαφέρουν, καθώς το κόστος είναι στην ουσία κόστος εργασίας. Η εταιρεία που θα αναλάβει την ανάπτυξη ενός συστήματος SCADA θα κοστολογήσει την εργασία αυτή ανάλογα με το κόστος της εργατοώρας του μηχανικού, το πλήθος των μηχανικών που εμπλέκονται στην εργασία και το σύνολο των εργατοωρών που απαιτούνται για την ολοκλήρωση της εφαρμογής SCADA.

### 6.3 Ορισμός

Όταν αναφερόμαστε στον όρο SCADA στην ουσία μιλάμε για έναν συνδυασμό Ηλεκτρονικού Υπολογιστή - Μηχανημάτων). Ο υπολογιστής αναλαμβάνει την εποπτεία της μηχανής και την συλλογή πληροφοριών από αυτήν μέσω αισθητήρων. Με την εποπτεία και συλλογή πληροφοριών από πολλές μηχανές έχουμε τη δυνατότητα παρακολούθησης μιας ολόκληρης παραγωγής.

Έτσι λοιπόν όσον αφορά το ρόλο που έχει ο υπολογιστής στην παραπάνω διαδικασία έχουμε:

- Συγκέντρωση πληροφοριών μέσω αισθητήρων

Θερμοκρασία, υγρασία, βάρος, στάθμη, θέση, ταχύτητα, πυκνότητα, πίεση, ροή, χρόνος, χρόνος λειτουργίας, χρόνος σταματήματος, και γενικά κάθε



φυσική τιμή η οποία μπορεί να μετρηθεί με κάποιο όργανο που κυκλοφορεί αυτή την στιγμή στην αγορά και μπορεί να μεταδώσει την μέτρηση του είτε ψηφιακά είτε αναλογικά.

Έχοντας λοιπόν τις παραπάνω πληροφορίες ο υπολογιστής είναι σε θέση να υπολογίσει χρόνους προληπτικής συντήρησης, κατανάλωση ρεύματος ή καύσιμου, χρόνους παραγωγής, παραγωγή, κοστολόγια, χρόνους βλαβών, συχνότητα βλαβών κ.λπ.

Όλα τα παραπάνω στοιχεία πέραν του υπολογισμού έχουν την δυνατότητα να καταγράφονται στον σκληρό δίσκο του υπολογιστή με επιλεκτική συχνότητα στον υπολογιστή από 0.05 sec έως . . . Δηλαδή σε μια περίπτωση μπορούμε να καταγράψουμε την πίεση του αέρα κάθε 0.05 sec, ενώ σε κάποια άλλη να καταγράψουμε την θερμοκρασία κάποιου υγρού μόνο όταν αυτή έχει ξεπεράσει για παράδειγμα τους 60° C.

Η καταγραφή μας δίνει την δυνατότητα να δούμε πια ήταν η ιδανικότερη κοστολογικά - παραγωγικά ημέρα και να προσαρμόσουμε τον τρόπο λειτουργίας σε αυτές τις συνθήκες. Όλες οι παραπάνω μετρήσεις έχουν την δυνατότητα να καταγράφονται συνεχόμενα στον σκληρό δίσκο του υπολογιστή για διάστημα τόσο όσο επιτρέπει ο ελεύθερος χώρος που έχει ο σκληρός δίσκος. Σήμερα με την εξέλιξη που έχουμε στους σκληρούς δίσκους είναι εφικτή η συνεχόμενη καταγραφή 100 αισθητήρων με συχνότητα 0.05 sec για ένα έτος.

Φυσικά είναι πάντα δυνατόν η μεταφορά αυτών των καταγράφων σε άλλο δίσκο ή να εκτυπωθούν σαν ιστορικό αρχείο ελευθερώνοντας χώρο στο σκληρό δίσκο που καταγράφει.

### ➤ Εποπτεία

Έχοντας στη διάθεση του όλες τις παραπάνω πληροφορίες ο υπολογιστής έχει την δυνατότητα εποπτείας μιας μηχανής και κατά συνέπεια μιας παραγωγής.

Για παράδειγμα ένας χειρίστης μιας μηχανής βλέποντας ένα ενδεικτικό alarm θα σταματούσε κάποιο κινητήρα, ή βλέποντας μια υψηλή θερμοκρασία θα έβαζε σε λειτουργία κάποιο ανεμιστήρα, ή παρατηρώντας μια υψηλή στάθμη σε ένα σιλό θα άνοιγε κάποια ηλεκτροβάννα ή θα έβαζε σε λειτουργία μια αντλία. Τους ίδιους παραπάνω χειρισμούς μπορεί να τους κάνει ένας

υπολογιστής με το πλεονέκτημα ότι η αντίδραση του θα είναι πιο έγκαιρη και ότι μπορεί να εποπτεύει περισσότερες της μια μηχανές. Φυσικά ο υπολογιστής δεν μπορεί να διορθώσει μια βλάβη, αλλά όπως ο χειρίστης θα φωνάζει έναν τεχνικό για να τη διορθώσει έτσι και στον υπολογιστή μπορούμε να έχουμε οπτικό ή ηχητικό σήμα βλάβης, τον ακριβή χρόνο σταματήματος, διάρκεια σταματήματος αλλά και το σταμάτημα μηχανών που έχουν άμεση ροή παραγωγής με την μηχανή που έχει βλάβη.

Η δυνατότητα ύπαρξης συνταγολογίου ελαχιστοποιεί το χρόνο αλλαγής παραγωγής. Παράδειγμα σε μια βιομηχανία η οποία κατασκευάζει μπίρα το παραγόμενο Α προϊόν αποτελείται από 100gr αλάτι, 200gr λεκιθίνη, 700gr νερό και χρειάζεται αυτά τα υλικά να ανακατευθούν για Τ χρόνο σε Θ βαθμούς θερμοκρασία. Έχοντας λοιπόν καταχωρημένες στον υπολογιστή τις συνταγές παραγωγής και επιλέγοντας μια από αυτές, αναλαμβάνει ο υπολογιστής την ζύγιση, ανάμειξη, και βρασμό τους.





Επομένως, συγκεντρωτικά, σε ένα σύστημα Scada μπορούμε πολύ εύκολα να έχουμε τις παρακάτω λειτουργίες.

- Αυτόματο / Χειροκίνητο από τον υπολογιστή.
- Χειροκίνητο από το PLC.
- Χειροκίνητο καλωδιακά.
- Συνταγολογία.
- Καταγραφή θεωρητικά απεριόριστη.
- Ανίχνευση επικοινωνίας Υπολογιστή - PLC.
- Αυτόματη μεταφορά σε Χειροκίνητο σε περίπτωση βλάβης του υπολογιστή.
- Επαναφορά On Fly ακόμα και σε χρονικά μεγάλες διακοπές ρεύματος.
- Εύκολη επέκταση σε περίπτωση προσθήκης νέων λειτουργιών χωρίς την διακοπή λειτουργίας.
- Ελληνικά μενού και μηνύματα σφαλμάτων.
- Χειρισμός χωρίς πληκτρολόγιο.
- Έως και με 8 διαφορετικά PLC επικοινωνία σε ένα Υπολογιστή.
- Αυτόματη μεταφορά των πληροφοριών σε Excel.

και πάρα πολλές ακόμα , ανάλογα με τις απαιτήσεις της εφαρμογής.

## 6.4 Scada WinCC V7.0 \*Siemens

Τα συστήματα SCADA αποτελούν εφαρμογή της βιομηχανικής πληροφορικής για την εποπτεία της παραγωγής. Με χρήση των συστημάτων SCADA επιτυγχάνεται on-line παρακολούθηση, μέσω μονάδων PLCs, και συνεχής καταγραφή σε ηλεκτρονικούς υπολογιστές όλων των κρίσιμων παραμέτρων της παραγωγικής διαδικασίας, για την επίτευξη εποπτείας σε πραγματικό χρόνο.

Ένα σύστημα SCADA περιλαμβάνει απεικόνιση σε μινικά διαγράμματα όλων των διεργασιών παραγωγής, ενδείξεις των τιμών των μετρούμενων μεγεθών, διαρκή συλλογή και αποθήκευση δεδομένων σε Η/Υ, γνωστοποίηση σφαλμάτων κ.α. Επίσης σημαντική είναι η δυνατότητα παρακολούθησης του συστήματος μέσω κατάλληλα διαμορφωμένων οελίδων του διαδικτύου.

Οι κύριες λειτουργίες ενός συστήματος SCADA είναι οι ακόλουθες:

- Συλλογή δεδομένων από τα PLCs και τις Απομακρυσμένες Τερματικές Μονάδες (RTU). Όλα τα επιθυμητά σήματα μεταδίδονται προς το σύστημα SCADA μέσω του δικτύου βιομηχανικού αυτοματισμού.
- Αποθήκευση των πληροφοριών στη βάση δεδομένων και αναπαράστασή τους μέσω γραφημάτων. Οι επιλεγμένες πληροφορίες αναπαρίστανται είτε αυτούσιες είτε έπειτα από κατάλληλη επεξεργασία.
- Ανάλυση δεδομένων και ειδοποίηση του προσωπικού σε περιπτώσεις σφάλματος. Όταν τα δεδομένα πάρουν τιμές μη κανονικές το σύστημα SCADA ειδοποιεί με οπτική ή ακουστική σήμανση τους χειριστές, ώστε να αποφευχθούν δυσάρεστες επιπτώσεις.
- Έλεγχος κλειστού βρόχου διεργασιών. Υπάρχει η δυνατότητα εφαρμογής τεχνικών ελέγχου, αυτόματες ή χειροκίνητες.

- Γραφική απεικόνιση των τμημάτων της διεργασίας σε μμικά διαγράμματα και παρουσιάσεις των δεδομένων σε ενεργά πεδία. Τα μμικά διαγράμματα απεικονίζουν ρεαλιστικά τμήματα της διεργασίας με στόχο την ευκολότερη εποπτεία και την κατανόηση των δεδομένων από τους χειριστές του συστήματος.
- Καταγραφή όλων των συμβάντων κανονικών και μη για την δημιουργία ιστορικού αρχείου. Σε κάθε βιομηχανία υπάρχει καταγραφή όλων των κρίσιμων παραμέτρων. Παλιότερα γινόταν με χειρόγραφη καταγραφή, ενώ σήμερα την ευθύνη αυτή έχει αναλάβει η βάση δεδομένων του συστήματος SCADA.
- Υποστήριξη διπλού υπολογιστικού συστήματος με αυτόματη εναλλαγή, αν αυτό κρίνεται σκόπιμο βάση της υπό έλεγχο διεργασίας. Σε διεργασίες υψηλής επικινδυνότητας πρέπει να ελαχιστοποιηθεί όσο το δυνατόν περισσότερο η εμφάνιση σφάλματος λόγω βλάβης του εξοπλισμού. Για τον λόγο αυτό τα συστήματα SCADA υποστηρίζουν δεύτερο υπολογιστικό σύστημα που αναλαμβάνει σε περίπτωση σφάλματος.
- Μεταφορά δεδομένων σε άλλα τμήματα του κεντρικού συστήματος πληροφόρησης και διαχείρισης.
- Έλεγχος της πρόσβασης χειριστών στα διάφορα υποσυστήματα του συστήματος SCADA.
- Ειδικές εφαρμογές λογισμικού όπως εκτέλεση κώδικα C++ ή ανάπτυξη ευφών συστημάτων.



Ένα σύγχρονο σύστημα ελέγχου και εποπτείας (SCADA) καλείται να επεξεργαστεί ένα τεράστιο όγκο δεδομένων και παράλληλα να δώσει τι κατάλληλες εντολές ελέγχου. Επιπλέον θα πρέπει να γίνει κατανοητό ότι ο ανθρώπινος παράγοντας έχει περιορισμένες ικανότητες στη επεξεργασία μεγάλου όγκου πληροφοριών, ωστόσο είναι σε θέση να λάβει σύνθετες και νοήμονες αποφάσεις.

Το λογισμικό καλείται να δώσει στο χειριστή του συστήματος με κατανοητό για τον τελευταίο τρόπο, έγκυρες πληροφορίες για την κατάσταση του συστήματος. Περνάνε λοιπόν σε ένα τελευταίο επίπεδο στην ιεραρχία των δικτύων, αυτό της διαχείρισης (management).

### 6.4.1 Περιβάλλον εργασίας / Δημιουργία ενός Project

➤ Εισαγωγή στο πρόγραμμα WinCC

Η επικοινωνία μεταξύ ενός plc, και ενός συστήματος Scada εκτός από το υποδίκτυο που χρησιμοποιείται, MPI, ProfibusDp κλπ, γίνεται κατά βάση μέσω των tags.

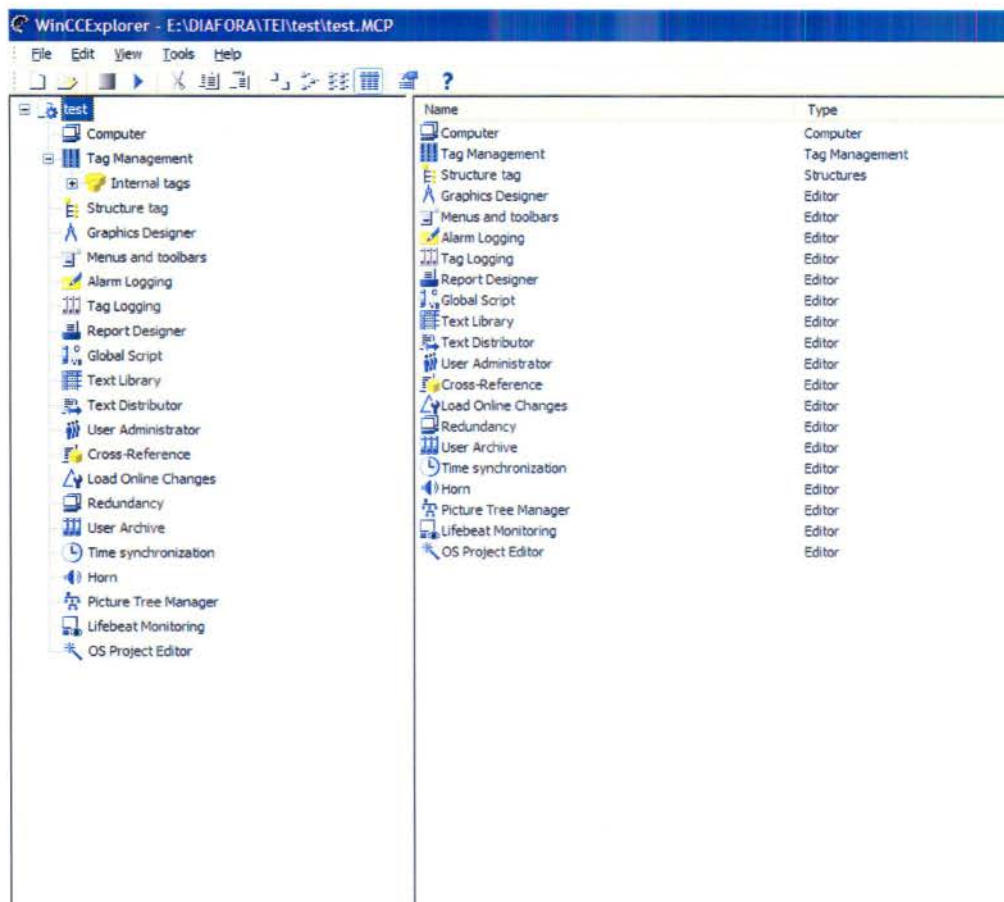
Τα tags δεν είναι τίποτα άλλο παρά κάποια δεσμευμένα data block από την μεριά του plc, τα οποία ανεξαρτήτως data type (bit, byte, int, floating points), χρησιμοποιούνται μόνο για την ανταλλαγή δεδομένων Scada - Plc.

Για παράδειγμα το memory bit που βρίσκεται στο Data Block 200, byte 1 και bit 6 , γίνεται «1», όταν δουλεύει το μοτέρ M1.

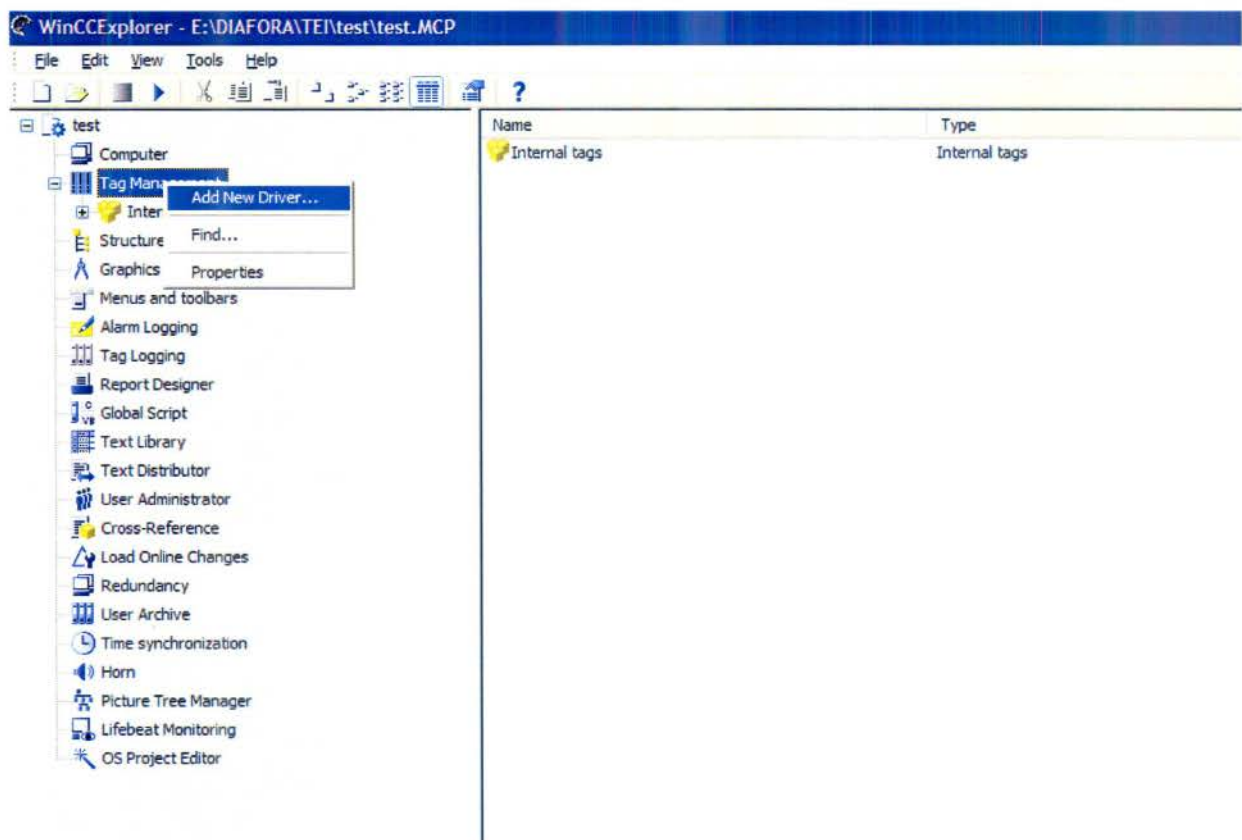
Αρα από τη μεριά του scada όταν το παραπάνω bit θα έχει την τιμή 1, τότε θα εμφανίζει ένα πράσινο μοτέρ, δείχνοντας έτσι πως το μοτέρ είναι σε λειτουργία. Κάπως έτσι γίνεται με όλα τα στοιχεία που προβάλλονται πάνω σε μία οθόνη Scada.

➤ Δημιουργία προγράμματος WinCC

Από file-new και αφού τοποθετήσουμε μέσω ενός project wizard τα βασικά του project, όπως name, storage path κλπ βγαίνουμε στη βασική οθόνη του project

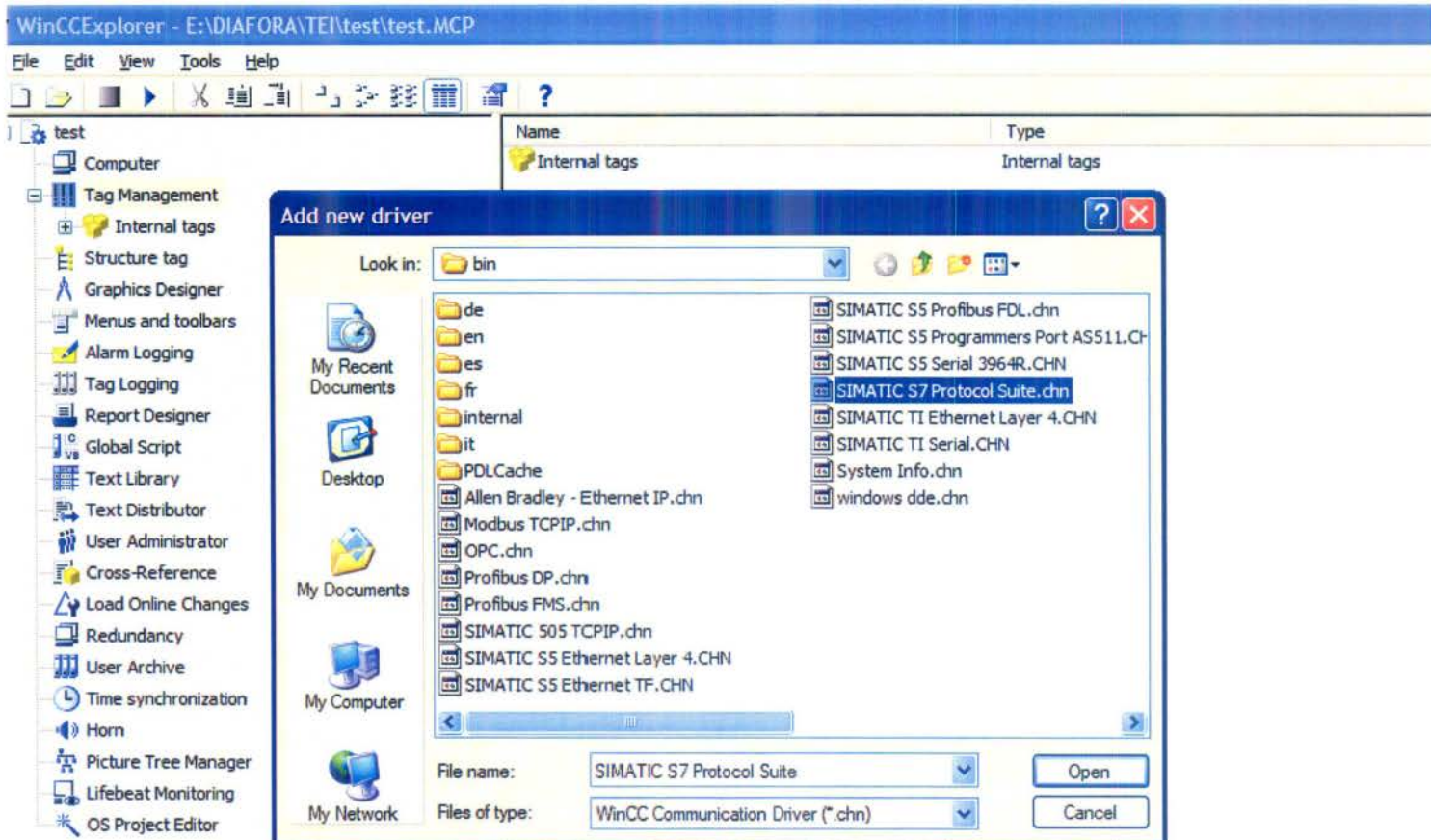


για το λόγο ότι η επικοινωνία του scada θα γίνει με ένα plc s7-300 της Siemens, κάνοντας δεξί κλικ στο tag management, κάνουμε κλικ add new Driver



KCU

επιλέγουμε τον SIMATIC S7 Protocol Suite



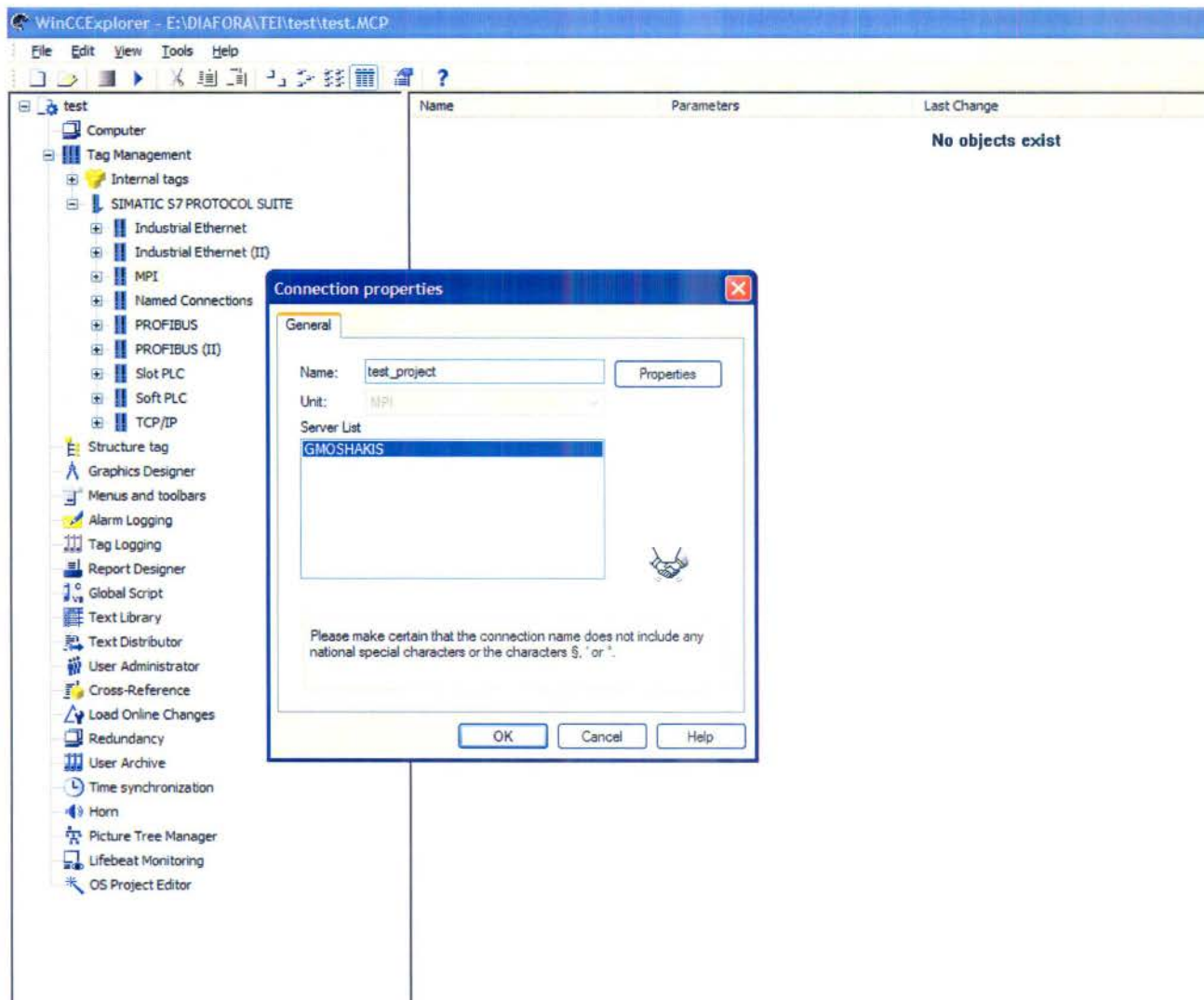
ο driver δημιουργείται

και στην επιλογή MPI , κάνουμε δεξί κλικ και επιλέγουμε

New Driver Connection

εμφανίζεται η οθόνη connection properties





πατώντας Properties εμφανίζεται ένα window που μπορούμε να αλλάξουμε ή να επιβεβαιώσουμε κάποιους παραμέτρους για το PLC όπως είναι η διεύθυνση του αντίστοιχου s7-station με το οποίο θα ανταλλάξει δεδομένα το Scada κλπ.



WinCCExplorer - E:\DIAFORA\TEI\test\test.MCP

File Edit View Tools Help

test

Name	Parameters	Last Change
No objects exist		

Computer

Tag Management

Internal tags

SIMATIC S7 PROTOCOL SUITE

- Industrial Ethernet
- Industrial Ethernet (II)
- MPI

Named Connections

PROFIBUS

PROFIBUS (II)

Slot PLC

Soft PLC

TCP/IP

Structure tag

Graphics Designer

Menus and toolbars

Alarm Logging

Tag Logging

Report Designer

Global Script

Text Library

Text Distributor

User Administrator

Cross-Reference

Load Online Changes

Redundancy

User Archive

Time synchronization

Horn

Picture Tree Manager

Lifebeat Monitoring

OS Project Editor

**Connection properties**

General

Name: test\_project Properties

Unit: MPI

Server List

- GMOSHAKIS

OK Cancel Help

**Connection Parameter - MPI**

Connection

S7 Network Address

Station Address: 2

Segment-ID: 0

Rack Number: 0

Slot Number: 0

Send/receive raw data block

Connection Resource: 02

Enter the station address of the controller.  
Legal address range: 0 to 126

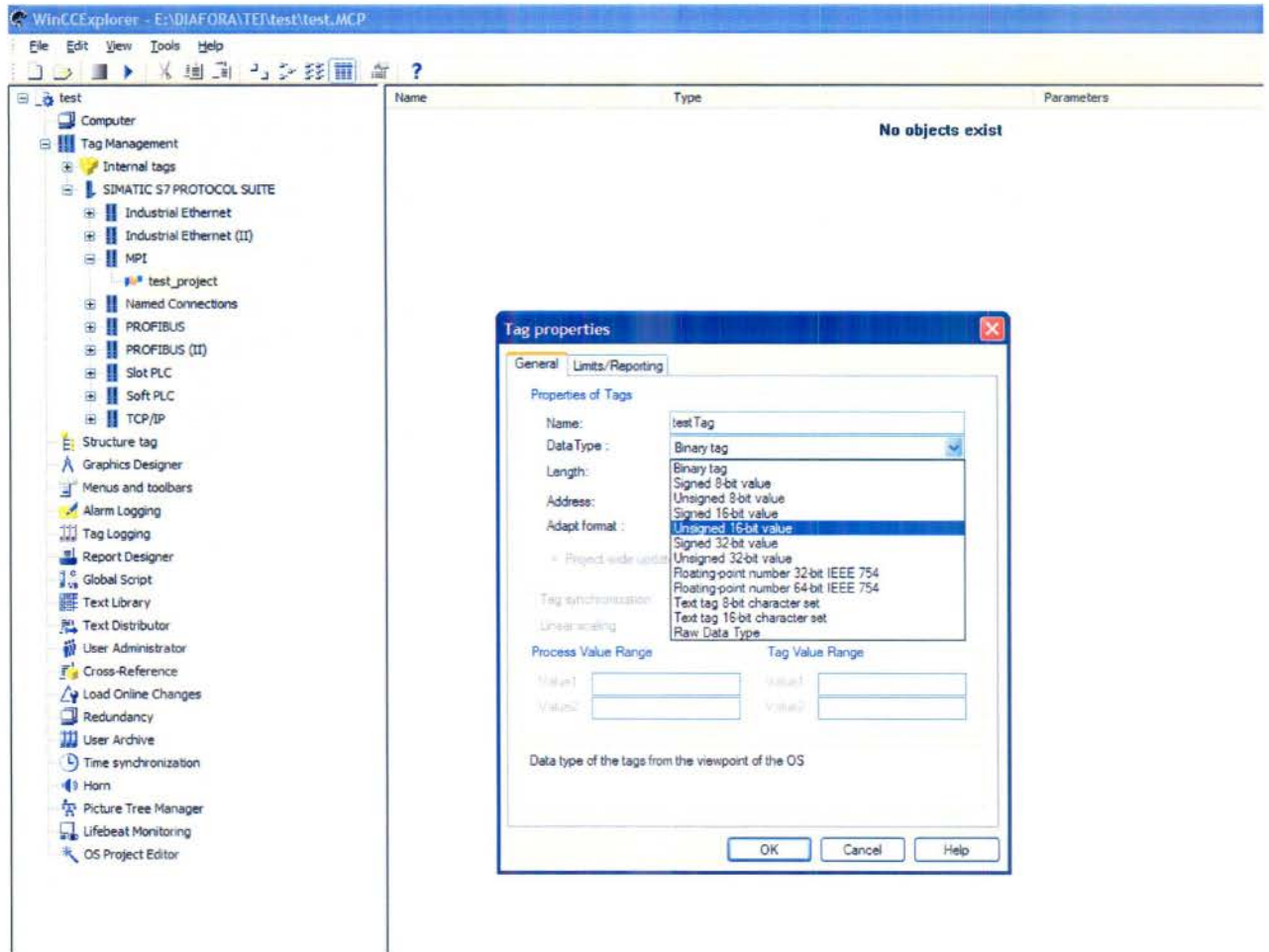
OK Cancel Help

ΒΙΒΛΙΟΘΗΚΗ  
ΤΕΙ ΠΕΙΡΑΙΑ

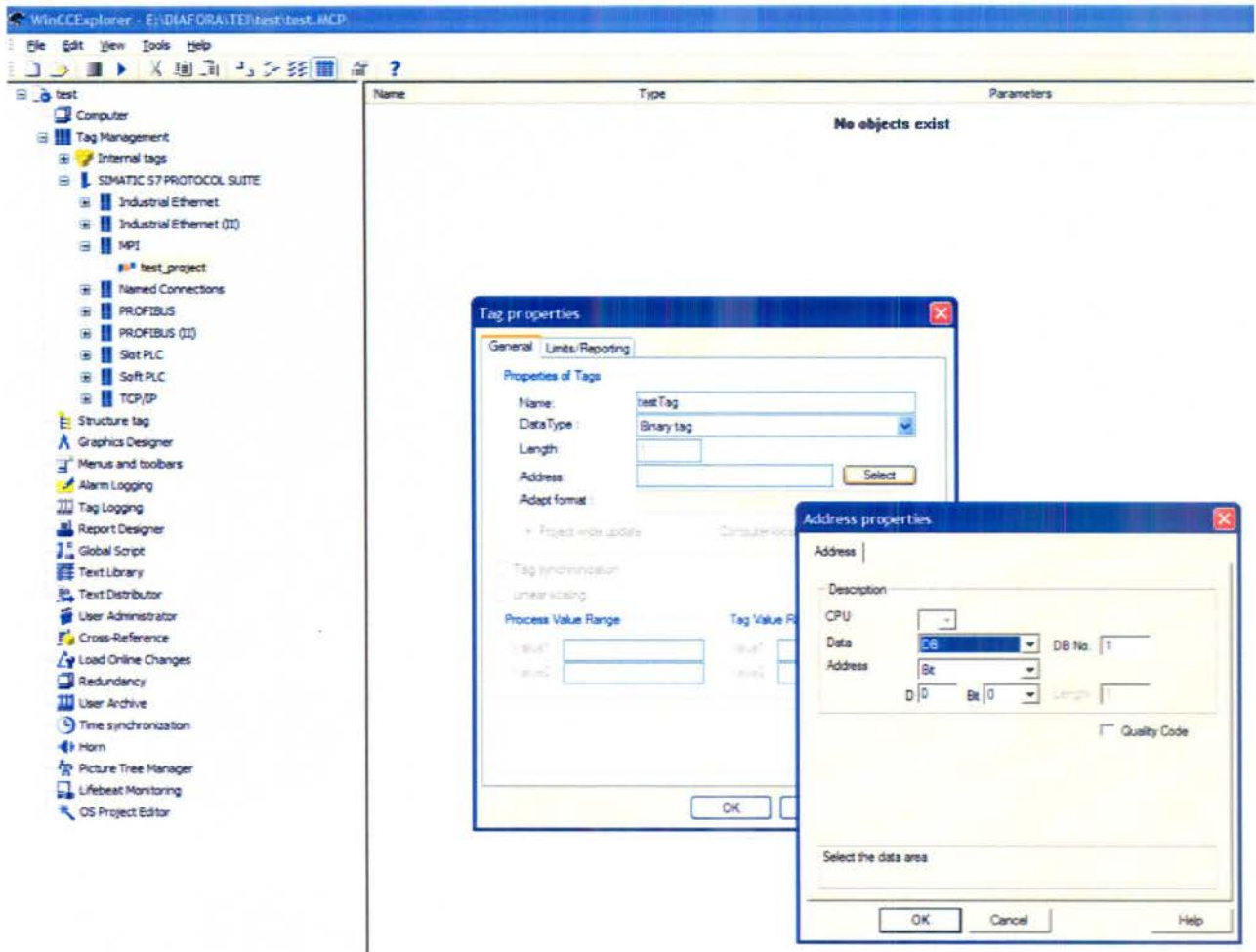


The screenshot shows the WinCC Explorer application window. The title bar reads "WinCC Explorer - E:\DIAFORA\TEI\test\test.MCP". The menu bar includes "File", "Edit", "View", "Tools", and "Help". The left pane displays a tree view of the project structure, including folders like "Computer", "Tag Management", "Internal tags", "SIMATIC S7 PROTOCOL SUITE", "Industrial Ethernet", "Industrial Ethernet (II)", "MPI", "test\_project", "Named Connections", "PROFIBUS", "PROFIBUS (II)", "Slot PLC", "Soft PLC", "TCP/IP", "Structure tag", "Graphics Designer", "Menus and toolbars", "Alarm Logging", "Tag Logging", "Report Designer", "Global Script", "Text Library", "Text Distributor", "User Administrator", "Cross-Reference", "Load Online Changes", "Redundancy", "User Archive", "Time synchronization", "Horn", "Picture Tree Manager", "Lifebeat Monitoring", and "OS Project Editor".

The right pane is a table with columns "Name" and "Type". The table is currently empty, displaying the text "No objects exist". A context menu is open over the table, listing the following options: "New Group...", "New Tag..." (highlighted), "Find...", "Cut", "Copy", "Paste", "Delete", and "Properties".



Click Select παρουσιάζεται ένα παράθυρο στο οποίο επιλέγουμε την διεύθυνση του bit από τη μεριά του PLC.



WinCC Explorer - E:\DIAFORA\TEI\test\test.MCP

File Edit View Tools Help

test

- Computer
- Tag Management
  - Internal tags
    - SMATIC 57 PROTOCOL SUITE
      - Industrial Ethernet
      - Industrial Ethernet (I)
      - MP1
        - test\_project
    - Named Connections
    - PROFIBUS
    - PROFIBUS (I)
    - Slot PLC
    - Soft PLC
    - TCP/IP
  - Structure tag
  - Graphics Designer
  - Menus and toolbars
  - Alarm Logging
  - Tag Logging
  - Report Designer
  - Global Script
  - Text Library
  - Text Distributor
  - User Administrator
  - Cross-Reference
  - Load Online Changes
  - Redundancy
  - User Archive
  - Time synchronization
  - Hom
  - Picture Tree Manager
  - Lifbeat Monitoring
  - OS Project Editor

Name Type Parameters  
**No objects exist**

**Tag properties**

General Limits/Reporting

Properties of Tags

Name: testTag  
 Data Type: Binary tag  
 Length:   
 Address:    
 Adapt format:

Tag synchronization

Process Value Range Tag Value Range

**Address properties**

Address

Description

CPU:   
 Data:  DB No.:   
 Address:

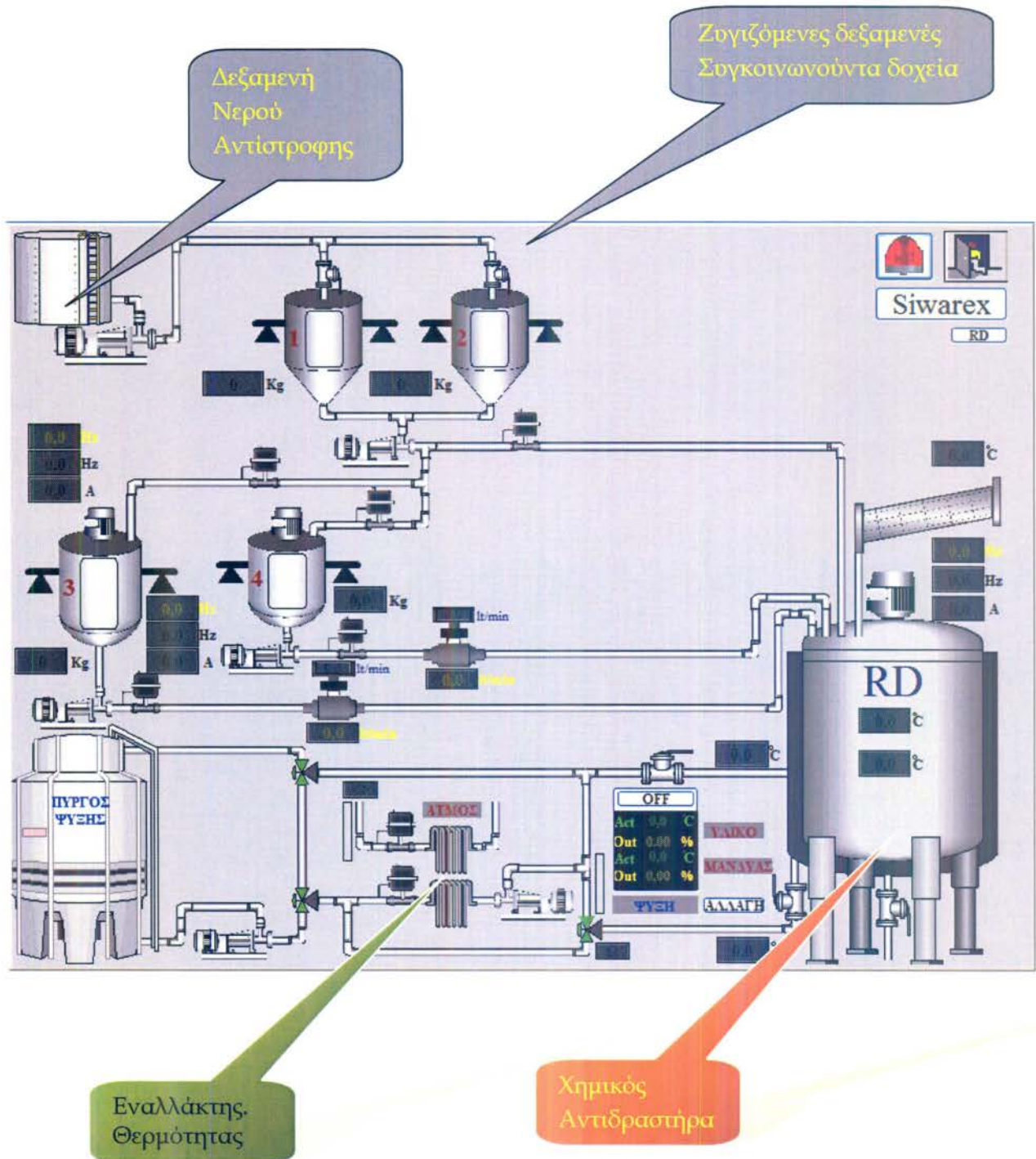
Quality Code

Select the data area



# ΜΕΡΟΣ 2<sup>ο</sup>

## Project





Ειδικός Εξοπλισμός που χρησιμοποιήθηκε στο Project

- PLC Siemens S7-300 CPU 315-2DP
- 8 Analog Input, 4 Analog Out
- 48 Digital Input
- 32 Digital Output
- Siwarex U, 2chanel
- Τρία Inverter με κάρτα Profibus Dp (1τεμ 15kw και 2τεμ 2,2 kw)
- Δύο όργανα θερμοκρασίας Dicon 400 με Profibus Dp
- Δύο όργανα θερμοκρασίας Jumo 316 με Analog Out
- Δύο PC 13 , 4-20mA - 3-15Psi



## ➤ Λειτουργία

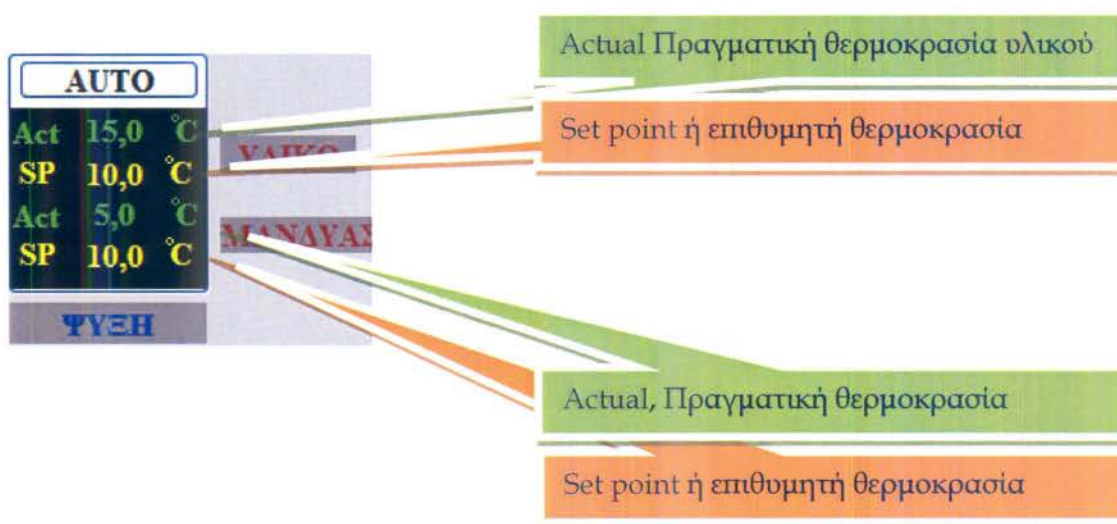
Στην ουσία το συγκεκριμένο Project, έχει δημιουργηθεί με σκοπό την αυτόματη ψύξη - θέρμανση ενός αντιδραστήρα, μέσα στον οποίο εκτελείται μια εξώθερμη χημική αντίδραση.

Ο αντιδραστήρας είναι μια διπλοχίτωνη δεξαμενή (υπάρχει κενό μεταξύ της εξωτερικής και της εσωτερικής επιφάνειας) 5000lt, όπου ψύχεται και θερμαίνεται με νερό που κυκλοφορεί ανάμεσα στα 2 χιτώνια.

Υπάρχουν 3 επιλογές για την ψύξη - θέρμανση του αντιδραστήρα.

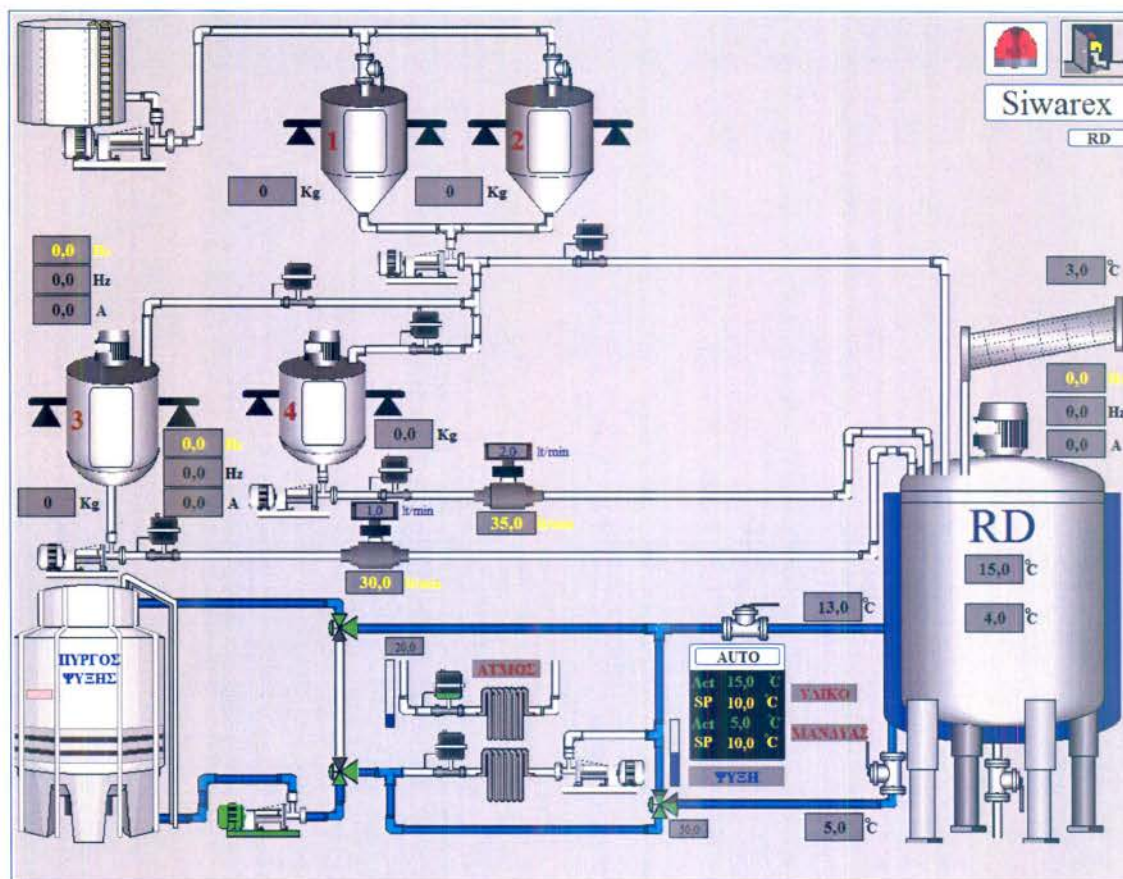
### ❖ Auto Procedure

→ Ψύξη:



Για την ψύξη του αντιδραστήρα θα πρέπει η actual θερμοκρασία του υλικού να είναι μεγαλύτερη από το set point το οποίο έχουμε ορίσει.

Τότε η ροή του υλικού θα κατευθυνθεί ως εξής:



Έτσι ο αντιδραστήρας αρχίζει να ψύχεται, ώσπου η πραγματική θερμοκρασία του υλικού να γίνει ίση με την επιθυμητή θερμοκρασία που έχουμε ορίσει (Set point).

Όταν γίνει αυτό, η διαδικασία σταματάει, ώσπου η θερμοκρασία του υλικού να ξεπεράσει και πάλι το set point.

→ Θέρμανση:

Ο χειριστής θα πρέπει να επιλέξει από το **ΑΛΛΑΓΗ** button

μόνο όταν το combo box βρίσκεται στην **OFF** επιλογή

την διαδικασία **Θέρμανσης**.



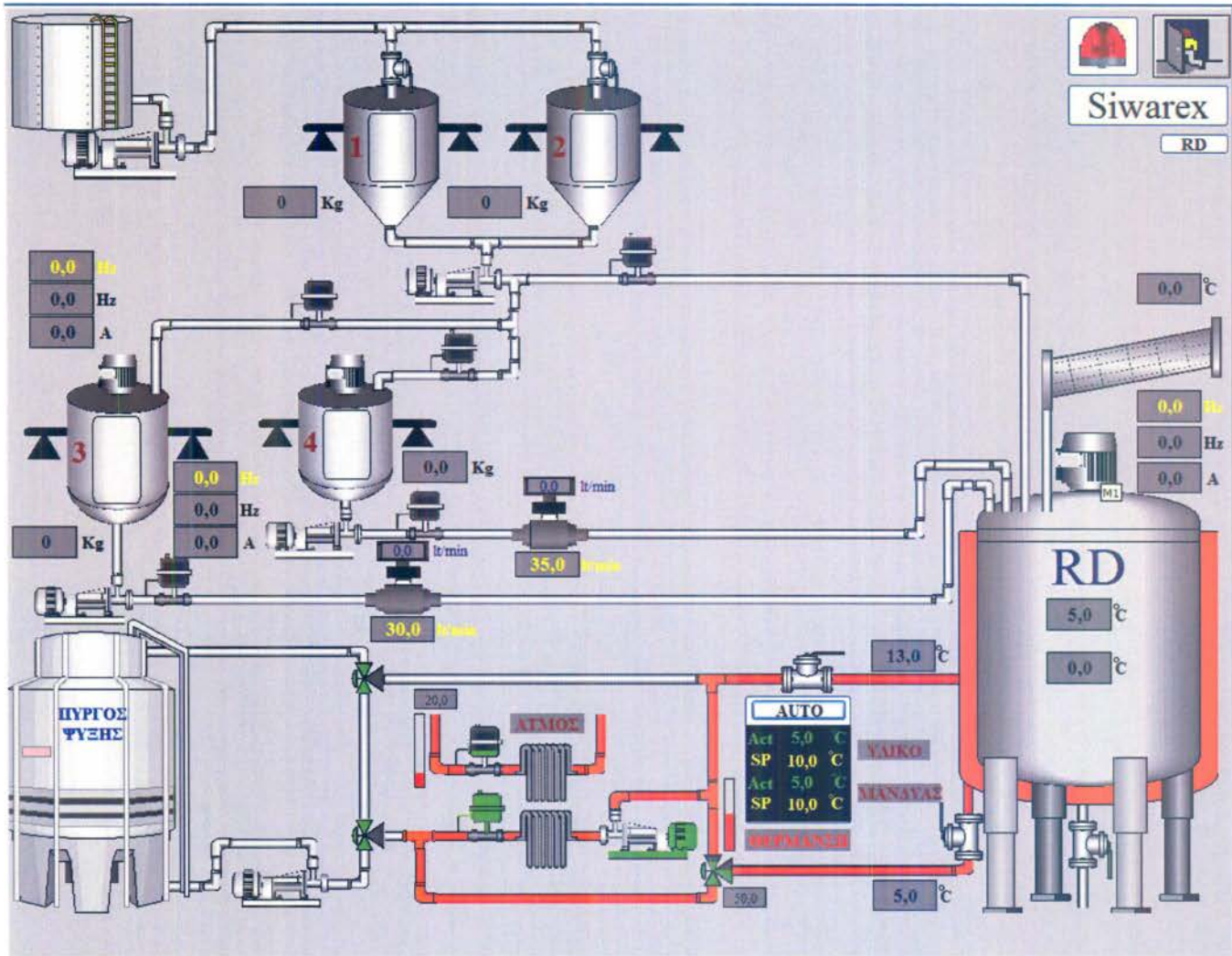
Για την θέρμανση του υλικού θα πρέπει η πραγματική θερμοκρασία του υλικού (**Act**) να είναι χαμηλότερη από την επιθυμητή θερμοκρασία (**SP**)

Για να ξεκινήσει η διαδικασία θα πρέπει να τοποθετηθεί στο combo box η επιλογή

**AUTO**



Η διαδικασία ξεκινάει και η ροή του υλικού κατευθύνεται ως εξής:



Με τη βοήθεια του εναλλάκτη θερμότητας και του ατμού, το νερό που κυκλοφορεί στις σωληνώσεις θερμαίνεται με στόχο την αύξηση της θερμοκρασίας του χημικού αντιδραστήρα. Όταν η θερμοκρασία του υλικού αρχίζει να φτάνει την επιθυμητή, οι αναλογικές βαλβίδες αρχίζουν σταδιακά να κλείνουν έτσι ώστε να επιτευχθεί η μεγαλύτερη δυνατή ακρίβεια.

Η θερμοκρασία του υλικού φτάνει την επιθυμητή και η διαδικασία σταματάει, ενώ οι αναλογικές βαλβίδες έχουν κλείσει στο 100%.

### ➤ Combi Procedure

Όταν στο Combo box επιλεχτεί η διαδικασία Combi, πρόκειται για μια διαδικασία τελείως αυτόματη, αφού δεν επεμβαίνει καθόλου ο χειριστής στο αν το σύστημα θα εκτελέσει ψύξη ή θέρμανση.

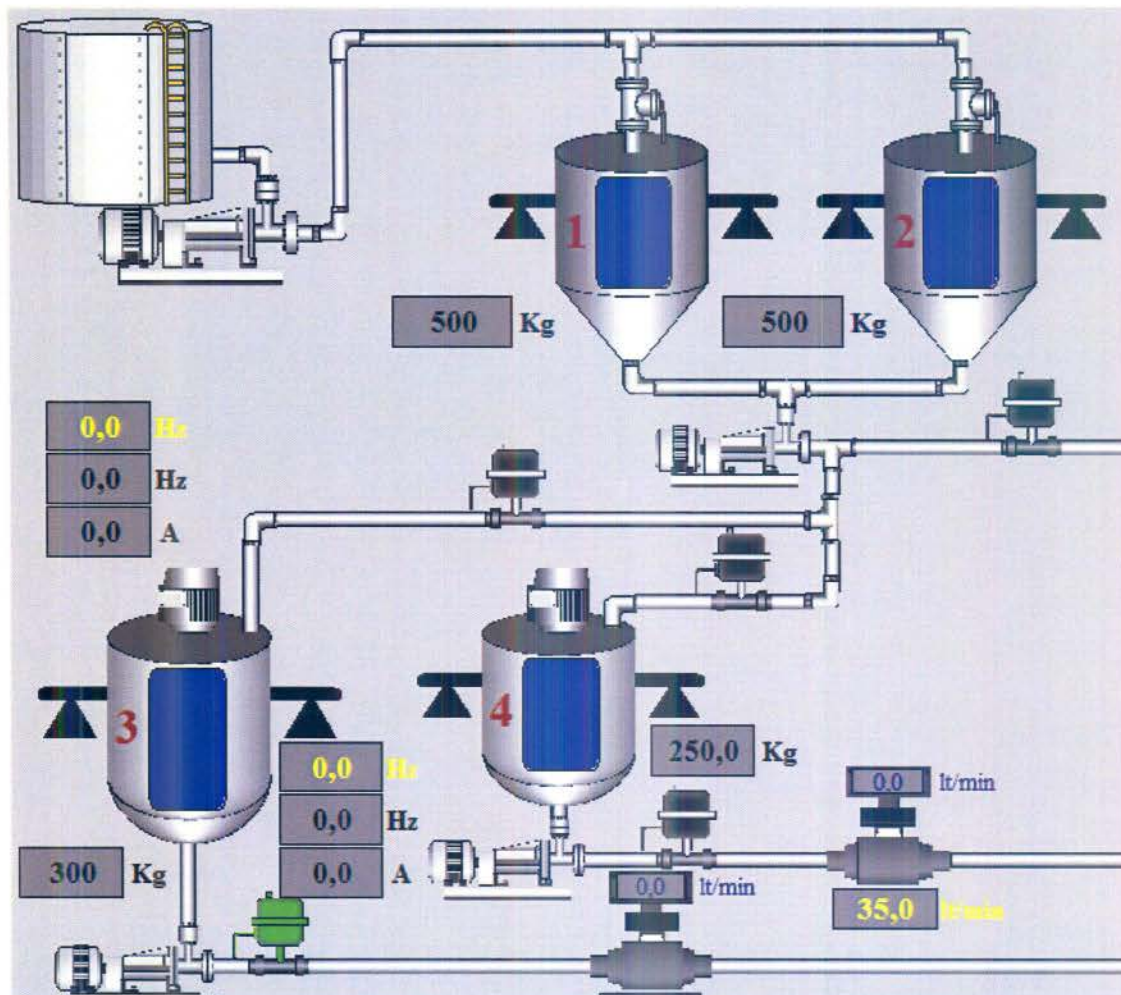
Οι διαδικασίες ψύξης, θέρμανσης του αντιδραστήρα ενεργοποιούνται πλήρως αυτόματα ανάλογα με τις πραγματικές θερμοκρασίες (Actual Values) του υλικού και του χιτωνίου του αντιδραστήρα, σε σχέση με τις επιθυμητές (Set points)

### ➤ Manual Procedure

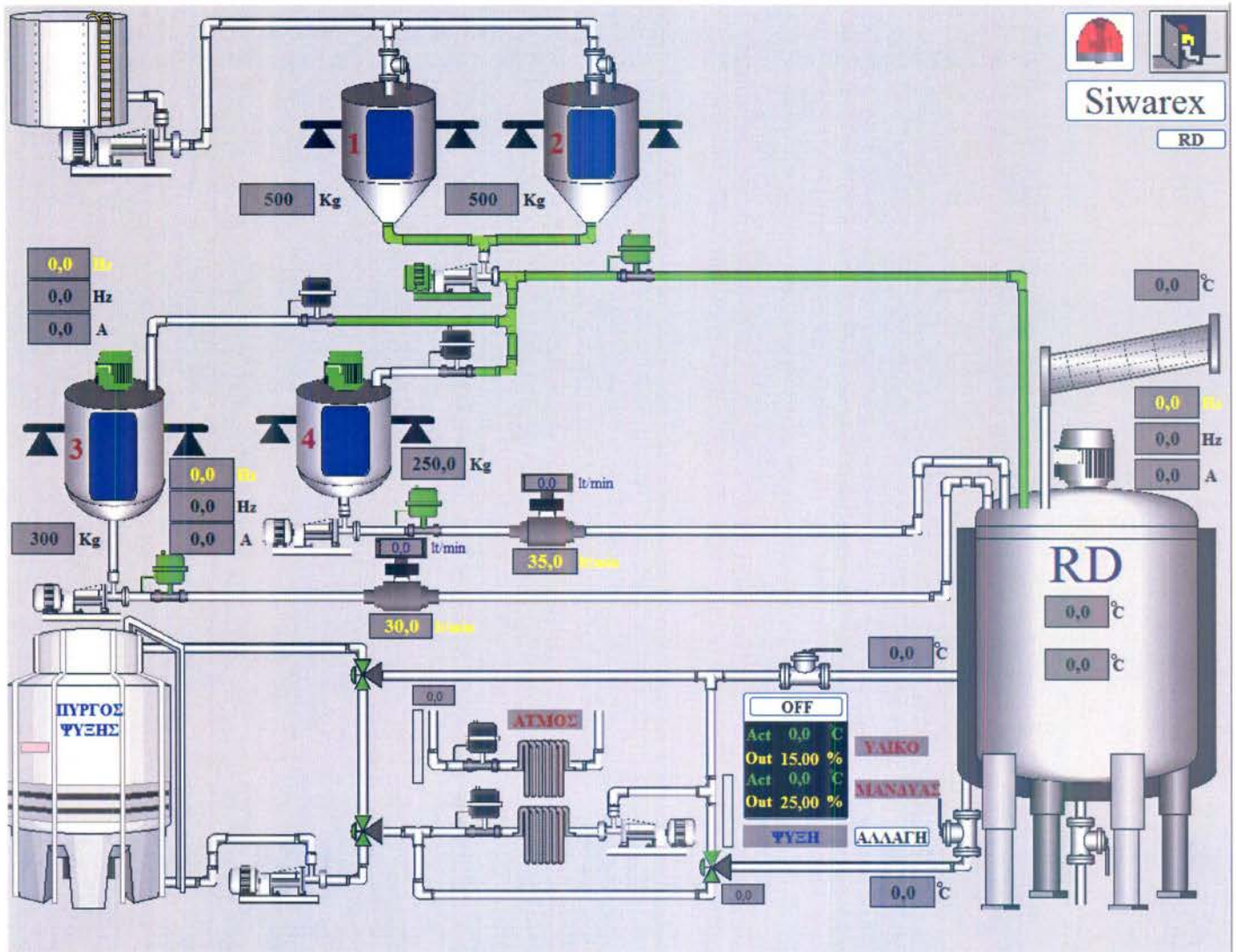
Η διαδικασία αυτή ενεργοποιείται όταν το combo box βρίσκεται στην αντίστοιχη επιλογή. Στην διαδικασία αυτή ο χειριστής είναι αυτός που πρέπει να καθορίσει αν θα γίνει ψύξη του αντιδραστήρα ή θέρμανση. Πέρα από αυτό όμως είναι και αυτός που πρέπει να καθορίσει το ποσοστό που θα ανοίξουν οι αναλογικές βαλβίδες ώστε να ψύξουν ή να θερμάνουν επαρκώς τον αντιδραστήρα

Η συγκεκριμένη διαδικασία χρησιμοποιείται σε περιπτώσεις βλάβης, που δεν μπορούν να χρησιμοποιηθούν οι 2 παραπάνω, ή και σε περιπτώσεις παραγωγής ενός νέου προϊόντος που χρειάζεται κάποιες δοκιμές. Όταν να μπορούν να καθοριστούν οι συγκεκριμένες θερμοκρασίες.





Εκτός όμως από την ψύξη - θέρμανση του αντιδραστήρα, υπάρχουν και άλλες 4 δεξαμενές οι οποίες περιέχουν διαλύτες, νερό ή και άλλα διάφορα χημικά που προστίθενται μέσα στον αντιδραστήρα έτσι ώστε να εκτελεστεί με επιτυχία η χημική αντίδραση.



Οι δεξαμενές 1 & 2, είναι ζυγιζόμενες δεξαμενές, ενώ η σύνδεση στο κάτω μέρος τους, δημιουργεί ένα σύστημα συγκοινωνούντων δοχείων.

Οι δεξαμενές 3, 4, είναι κι αυτές ζυγιζόμενες δεξαμενές, αλλά ανεξάρτητες μεταξύ τους.

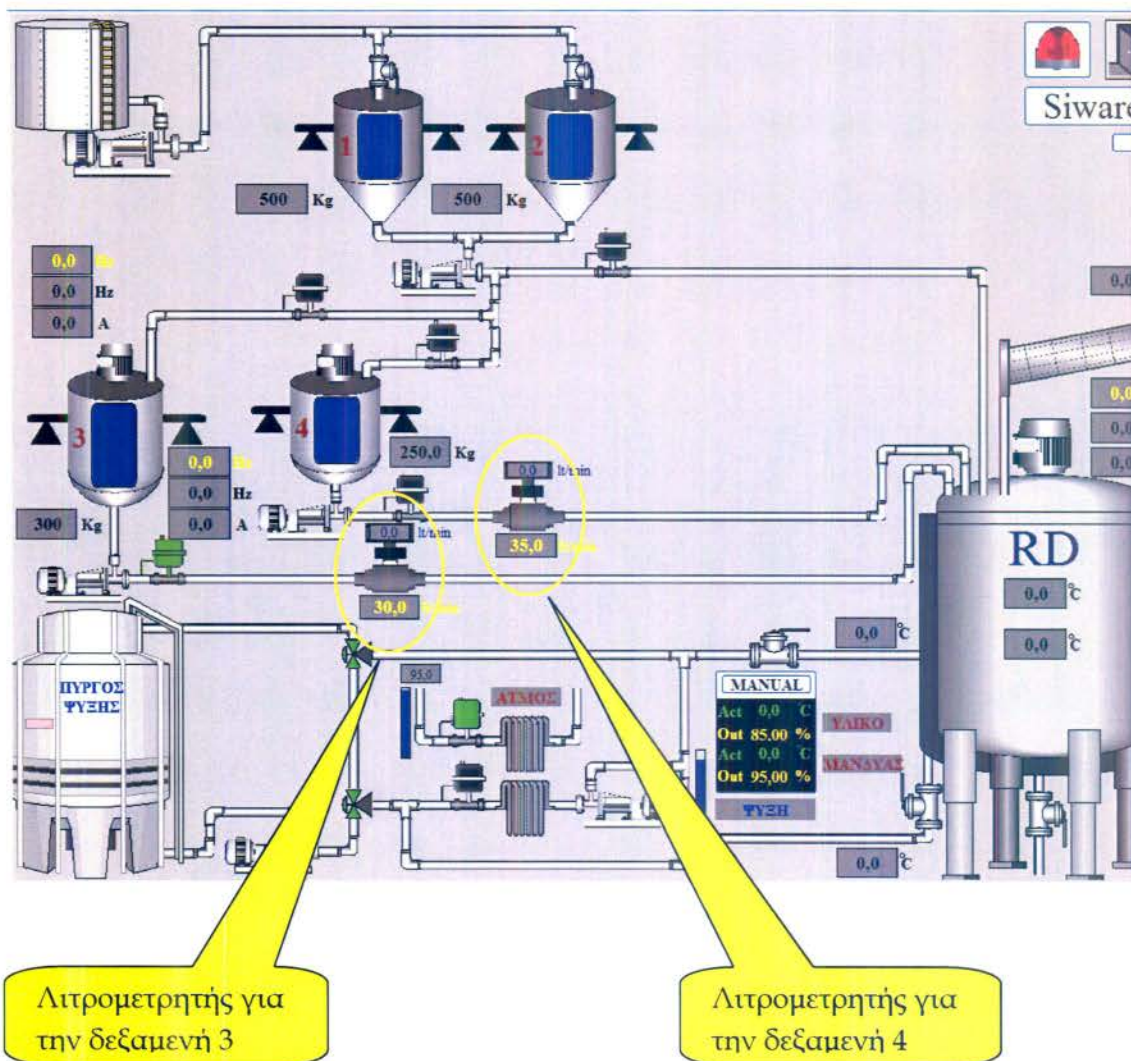
Η προσθήκη των υλικών που περιέχουν οι δεξαμενές 1-4 στον αντιδραστήρα, δεν γίνεται με κάποιο συνταγολόγιο, παρά μόνο χειροκίνητα.

Δηλαδή κάνοντας click πάνω σε μια βαλβίδα με το ποντίκι του υπολογιστή, η βαλβίδα ανοίγει.

Αντίστοιχα ενεργοποιούνται και οι αντλίες στην εκκένωση των δεξαμενών.

Στις δεξαμενές όμως 3 & 4 ο χειριστής έχει την εξής δυνατότητα:

Παρατηρείτε πως στην έξοδο των δεξαμενών αυτών υπάρχει ένας λιτρομετρητής ο οποίος μετράει την ποσότητα του υλικού που κατευθύνεται στον αντιδραστήρα.



Παρατηρείτε πως ο κάθε λιτρομετρητής έχει 2 πεδία αριθμών.



Το πάνω πεδίο δείχνει την actual (πραγματική) τιμή των λίτρων, ενώ το κάτω είναι το set point, δηλ η επιθυμητή τιμή των λίτρων που τοποθετείται από τον χειριστή.

Οι αντλίες εκκένωσης των εν λόγω δεξαμενών οδηγούνται από Inverter (Altivar 71).

Όταν ο χειριστής γράψει μια τιμή lt/min στο αντίστοιχο πεδίο, το Inverter προσαρμόζει έτσι τις στροφές του ώστε να επιτύχει την επιθυμητή τιμή σε lt/min.

Αντίστοιχα ο χειριστής μπορεί να επιλέξει και επιθυμητή τιμή των Hz των αντλιών 3 & 4 στο κατάλληλο πεδίο.

0,0	Hz
0,0	Hz
0,0	A



# ΜΕΡΟΣ 3ο

Ηλεκτρολογικά Σχέδια Εφαρμογής



Βιομηχανικοί Αυτοματισμοί  
Συστήματα διαχείρισης Α΄Υλών

57ο ΧΛΜ Εθν. Οδού Αθηνών-Λαμίας Τηλ: 6973697134

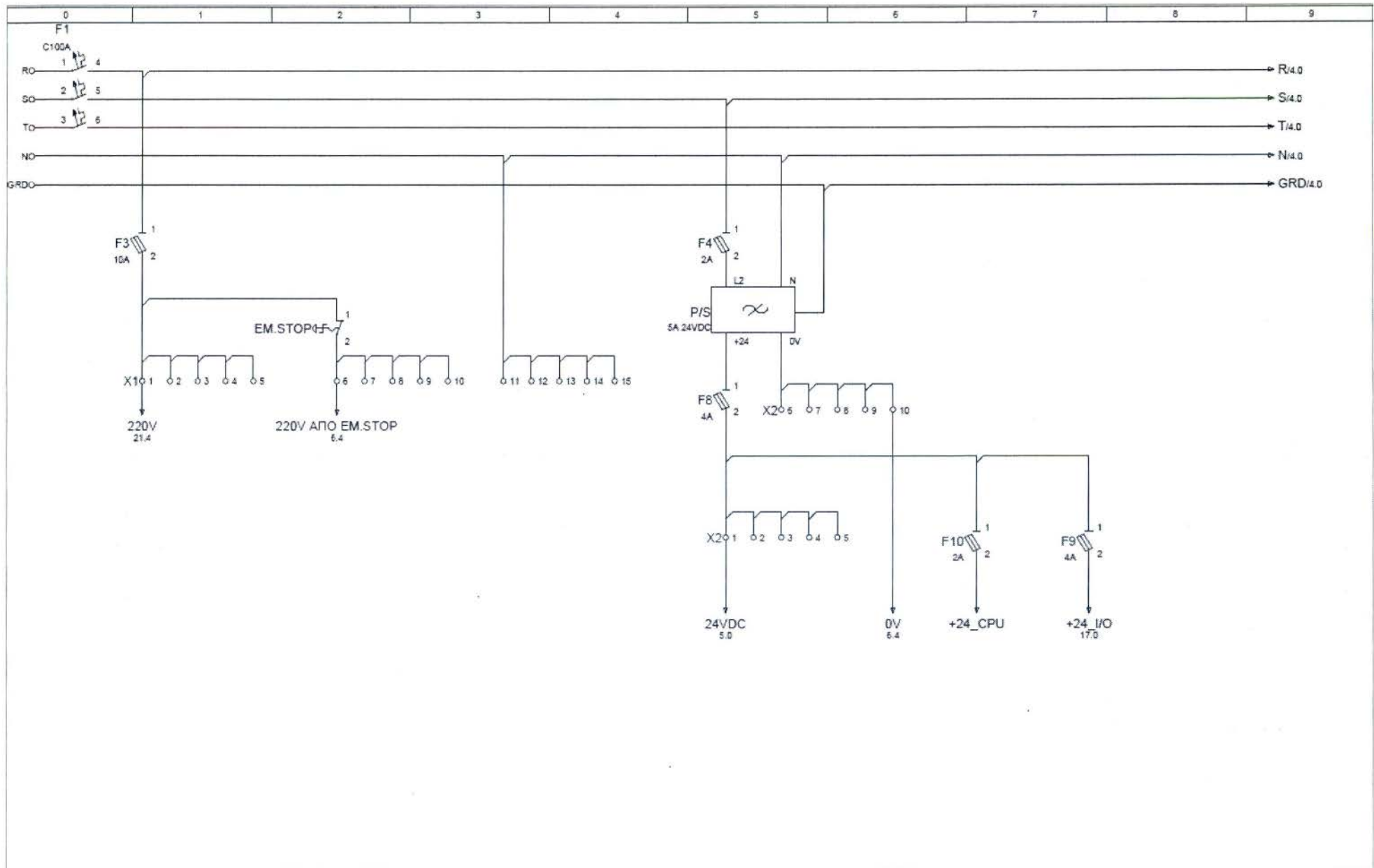
**ΠΕΛΑΤΗΣ: ΤΕΙ ΠΕΙΡΑΙΑ**

**ΕΡΓΟ: ΑΥΤΟΜΑΤΙΣΜΟΣ ΑΝΤΙΔΡΑΣΤΗΡΑ**

ΙΑΝΟΥΑΡΙΟΣ 2011

ΥΠΕΥΘΥΝΟΣ ΜΗΧ/ΚΟΣ: Μοσχάκης Γεώργιος

ΟΝΟΜΑ ΑΡΧΕΙΟΥ: D:\SXEDIA\ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ



ΣΕΛΙΔΑ :1

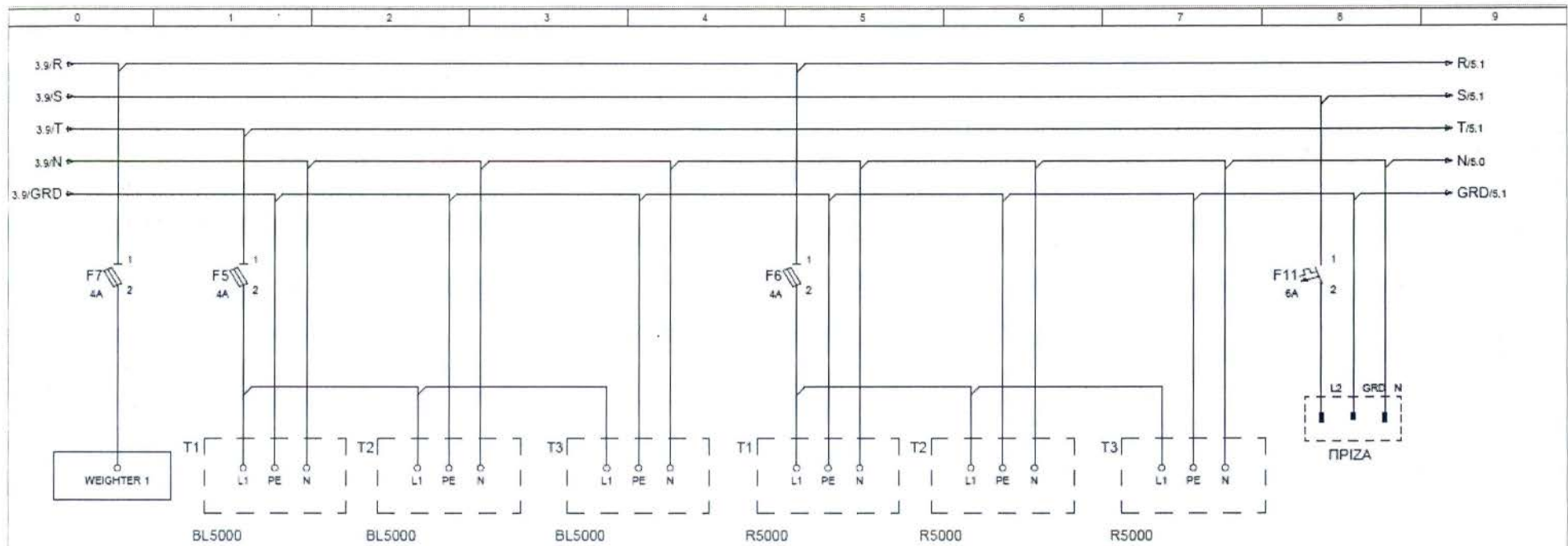
ΡΑΘΗ :D:\ΣΧΕΔΙΑΠΤΥΧΙΑΚΗ

### ΑΥΤΟΜΑΤΙΣΜΟΣ ΑΝΤΙΔΡΑΣΤΗΡΑ

ΗΜ/ΝΙΑ : 20/01/2011

ΚΩΔ ΕΡΓΟΥ

100.011



ΣΕΛΙΔΑ :2

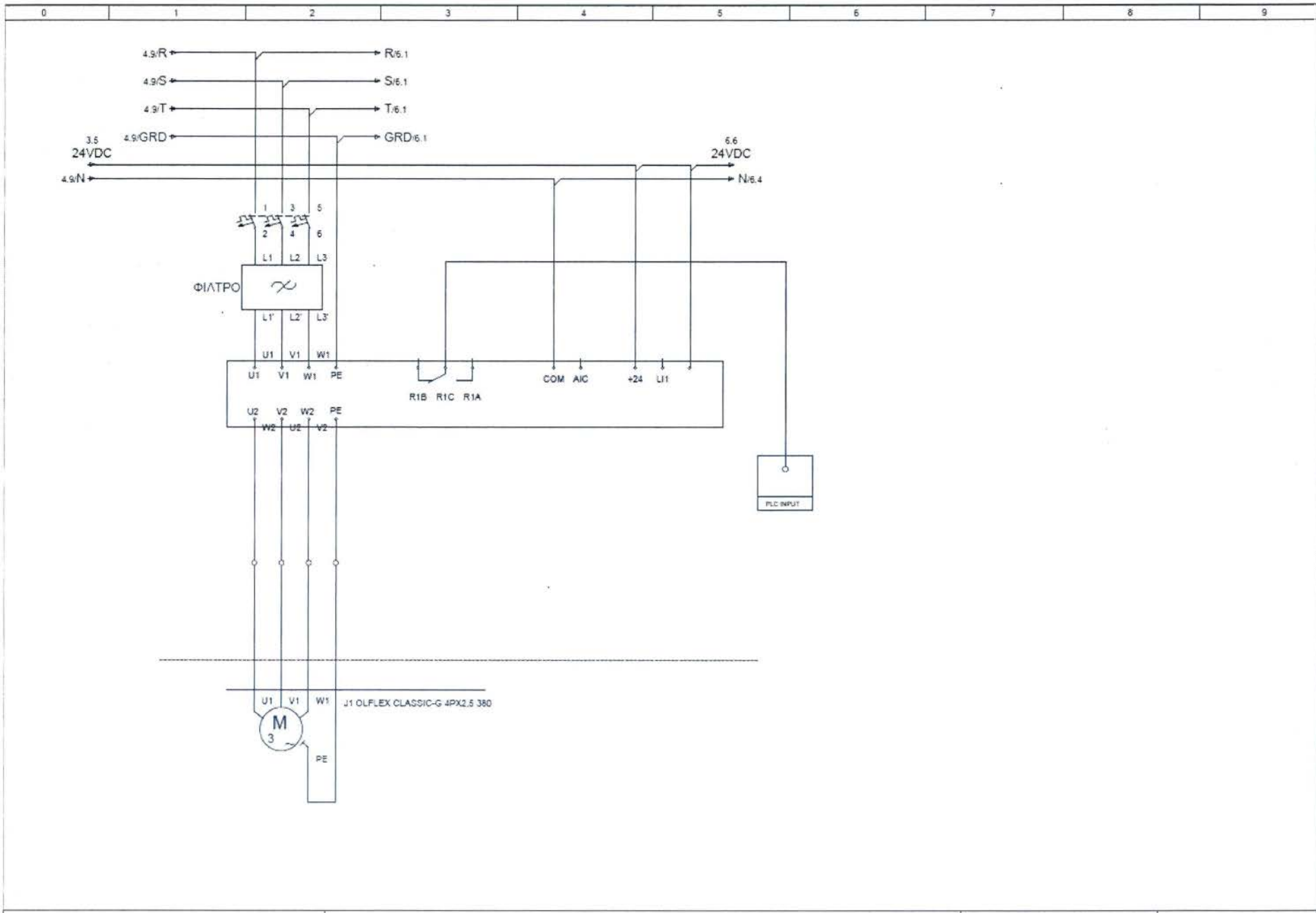
ΡΑΘΗ :D:\ΣΧΕΔΙΑ\ΠΤΥΧΙΑΚΗ

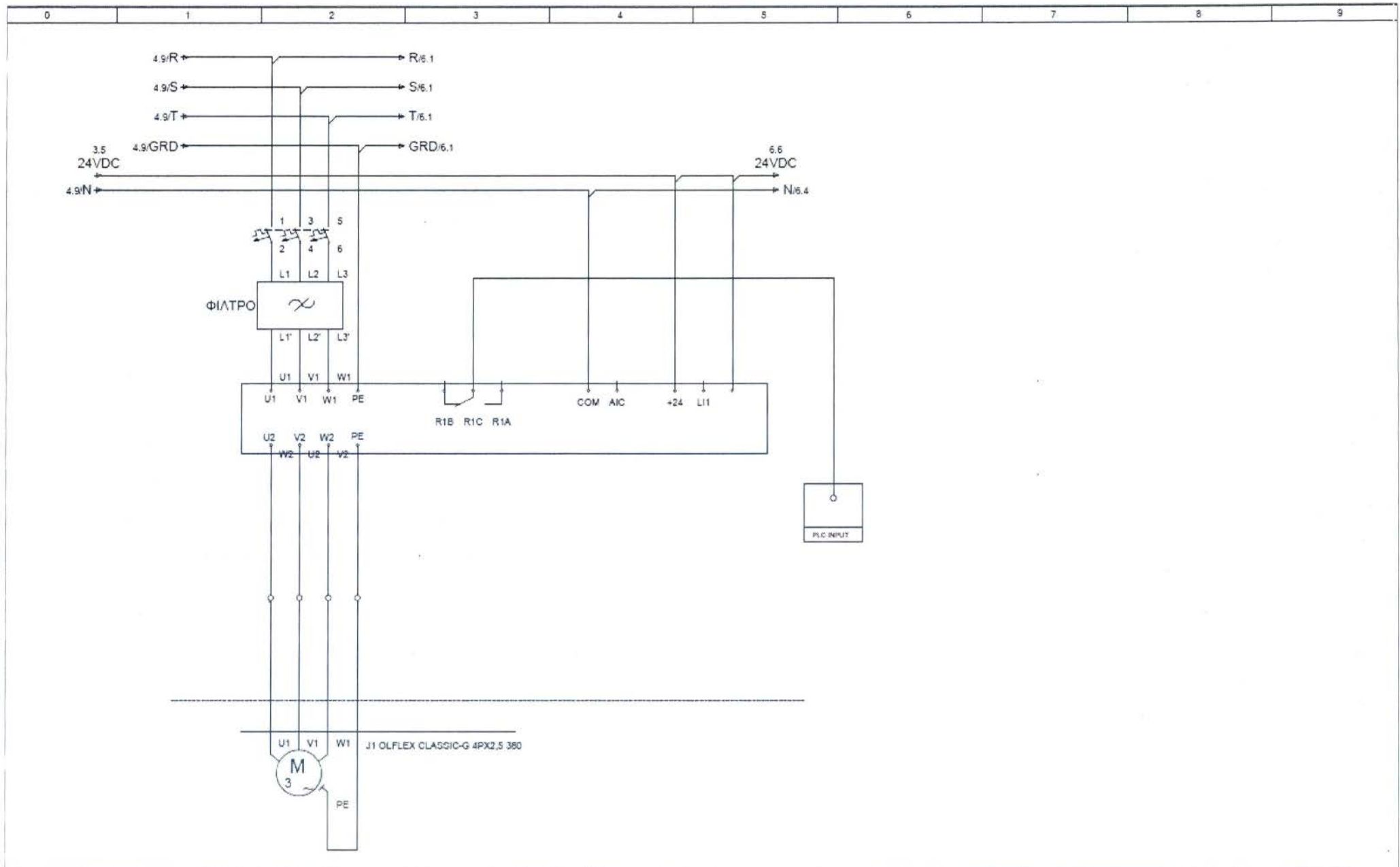
### ΑΥΤΟΜΑΤΙΣΜΟΣ ΑΝΤΙΔΡΑΣΤΗΡΑ

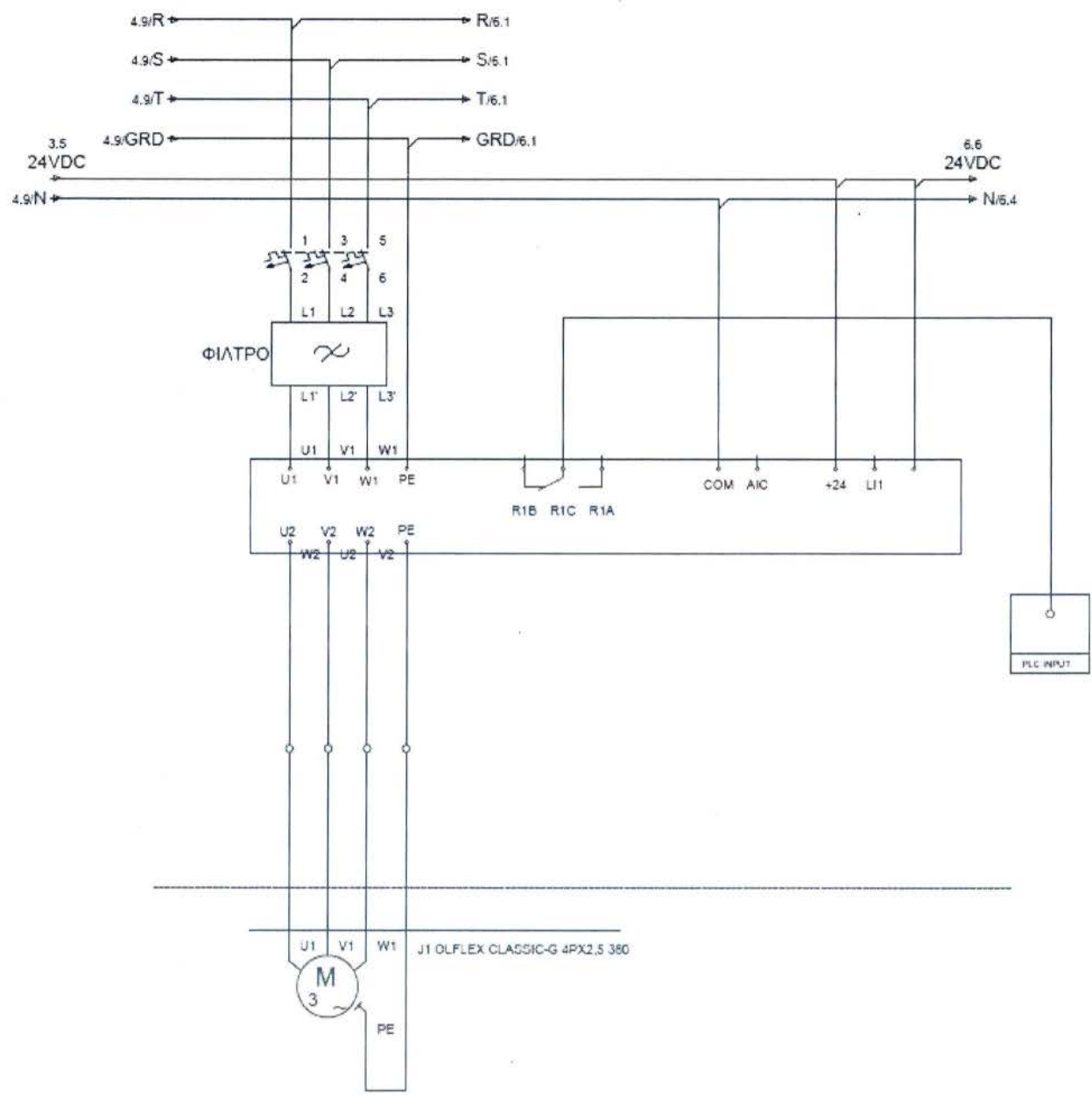
ΗΜ/ΝΙΑ : 20/01/2011

ΚΩΔ ΕΡΓΟΥ

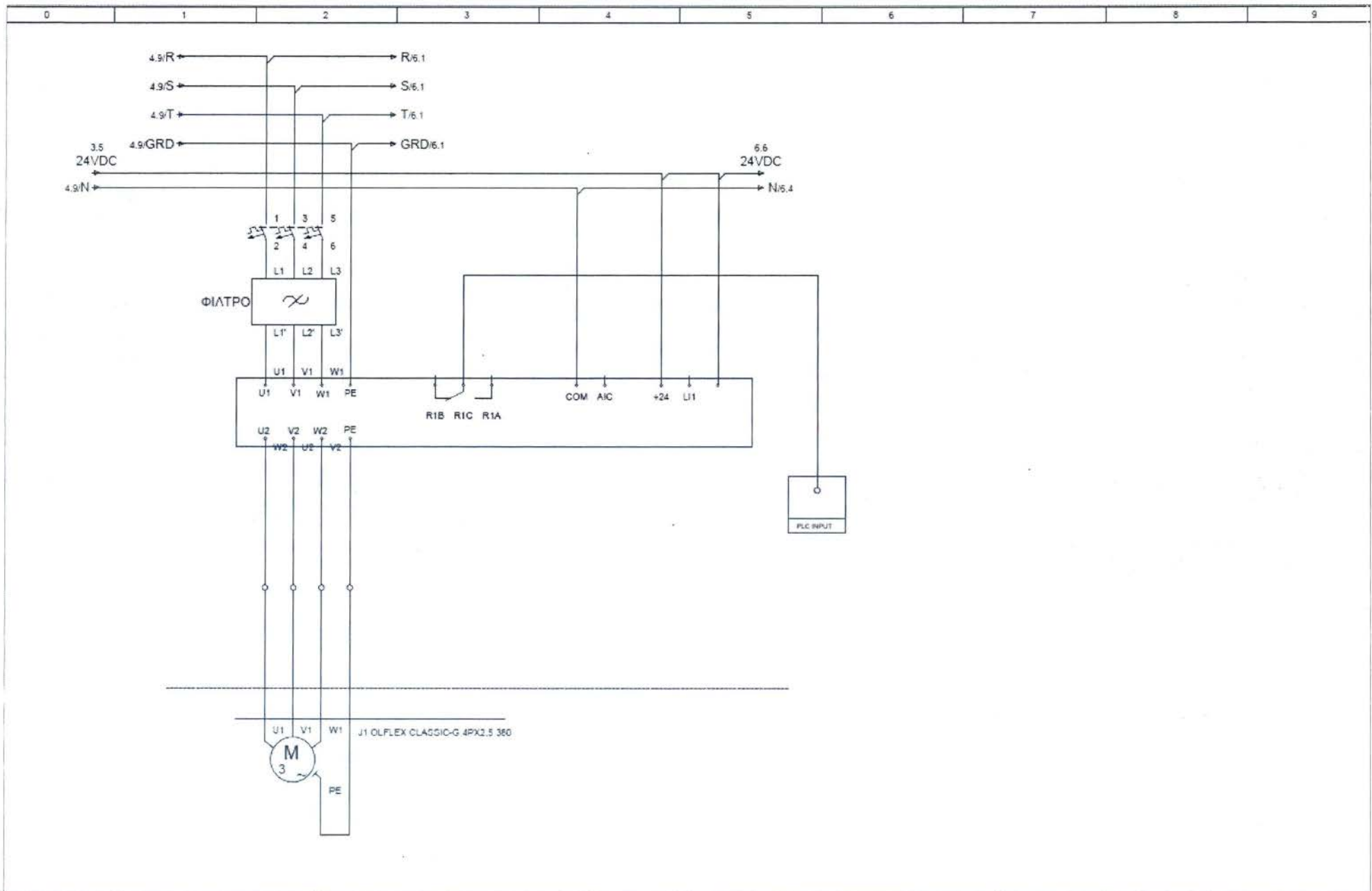
100.011

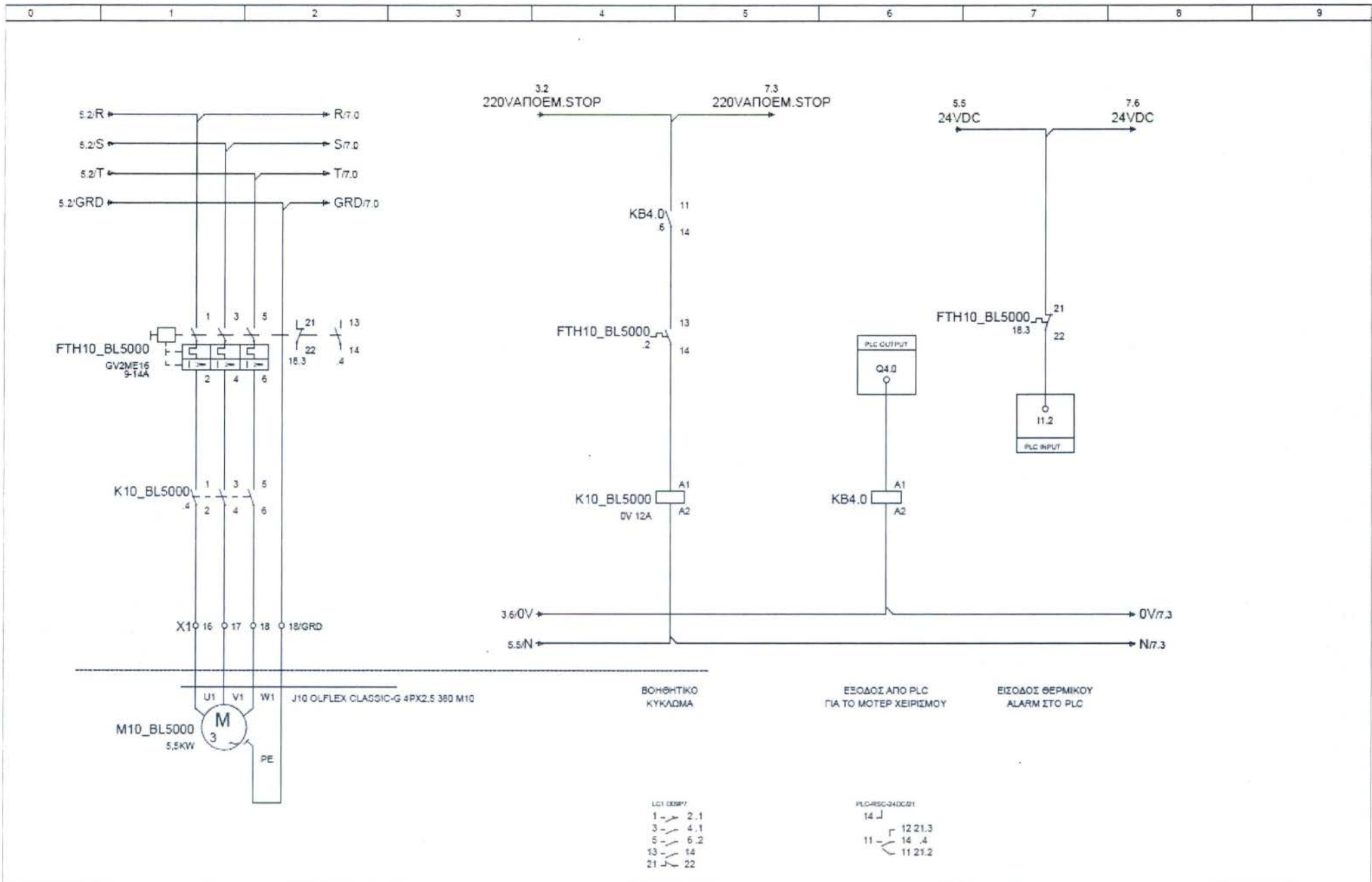


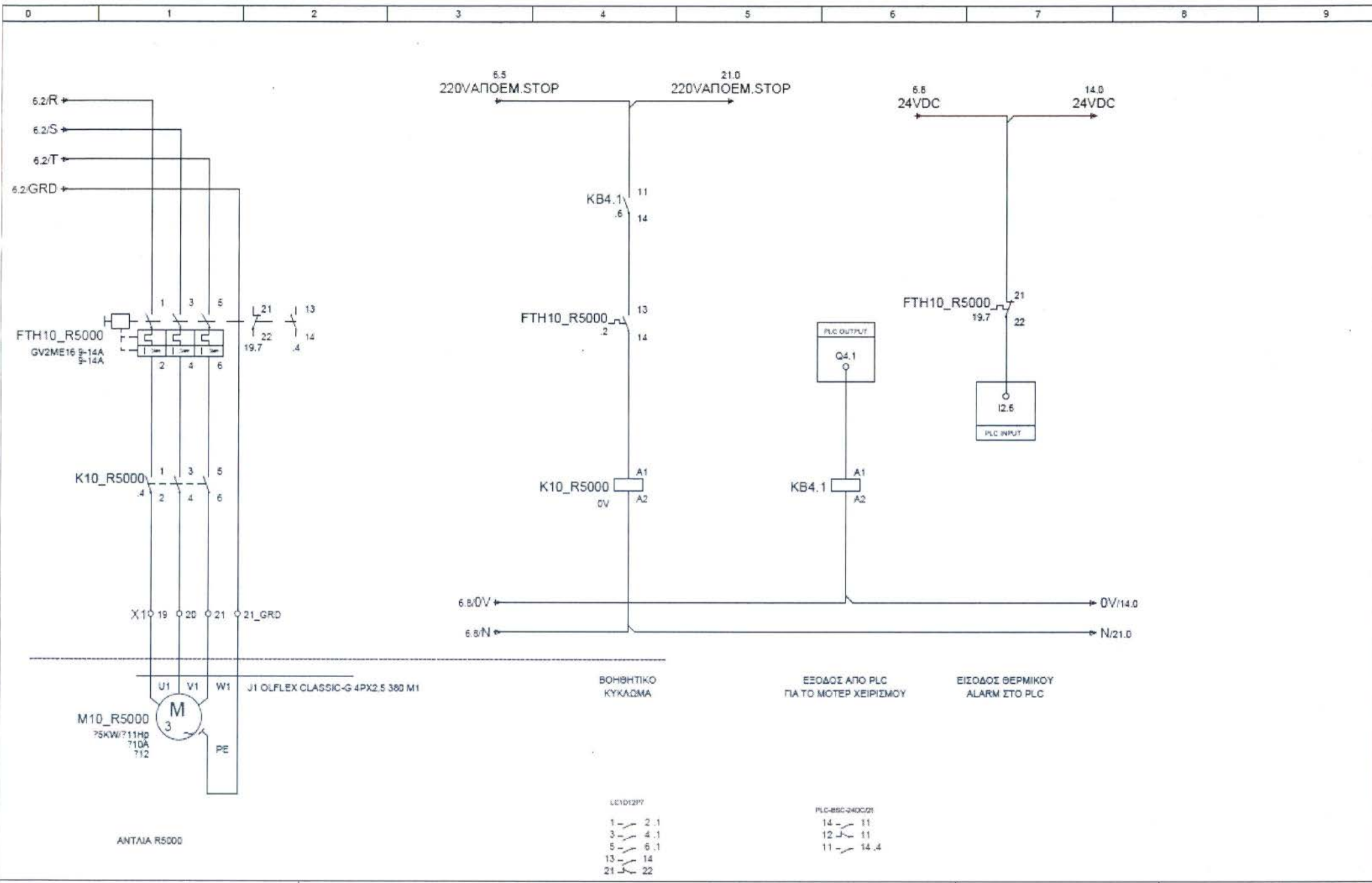












CPU 315-2DP (128KB)

6ES73152AG100AB0

MEMORY CARD: 6ES795038LG110AA0

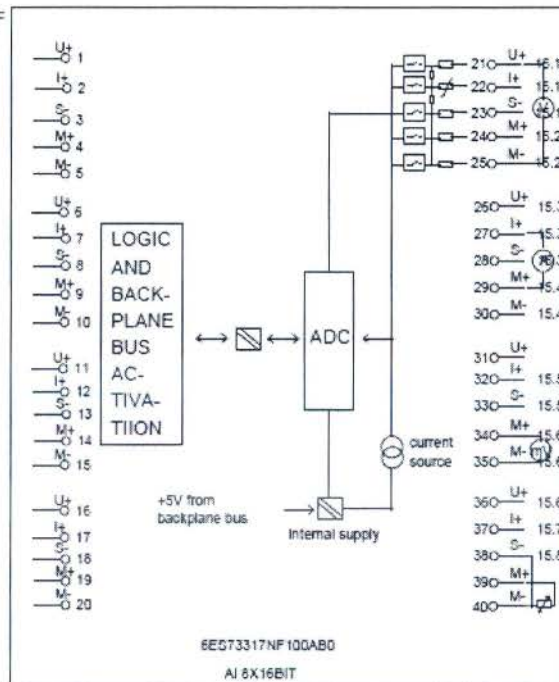
14.1  
14.1  
14.1  
14.3  
14.3  
14.3  
14.4  
14.5  
14.5  
14.5  
14.6  
14.7  
14.7  
14.7  
14.8  
14.8

RESERVE

RESERVE

RESERVE

-331\_1KF

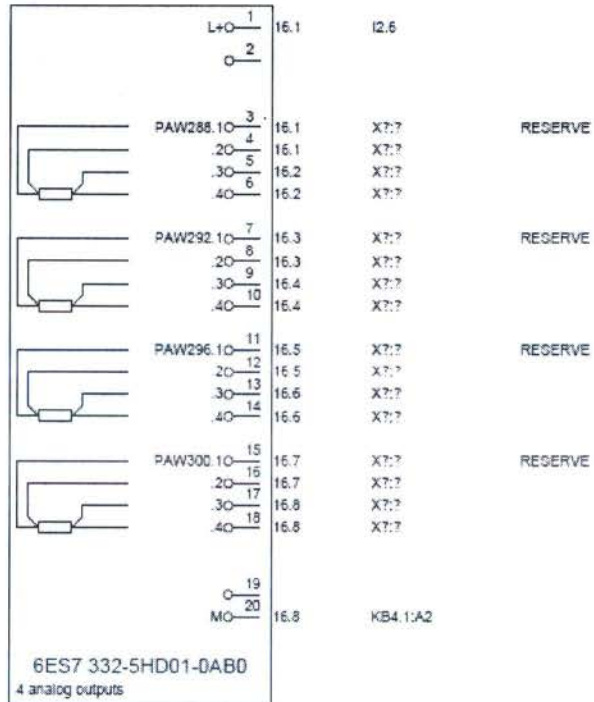


X4:1  
X4:2  
X4:3

RESERVE RESERVE

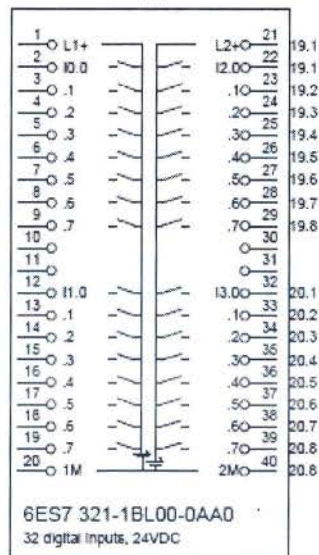
RESERVE

-A1

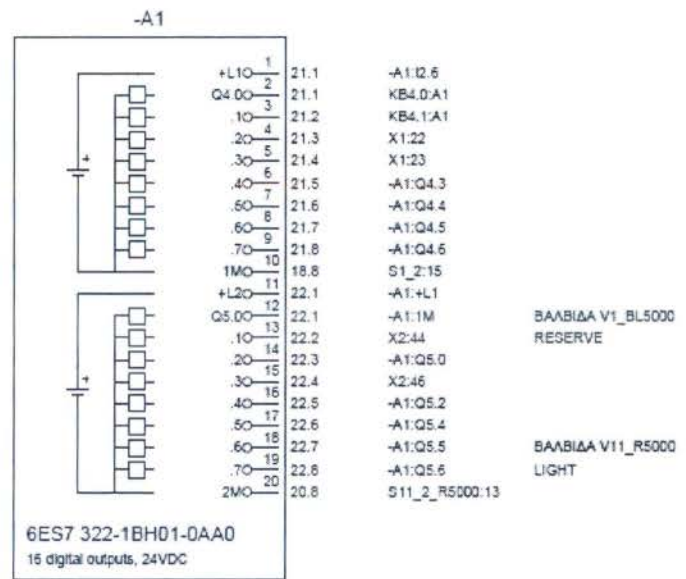


-A1

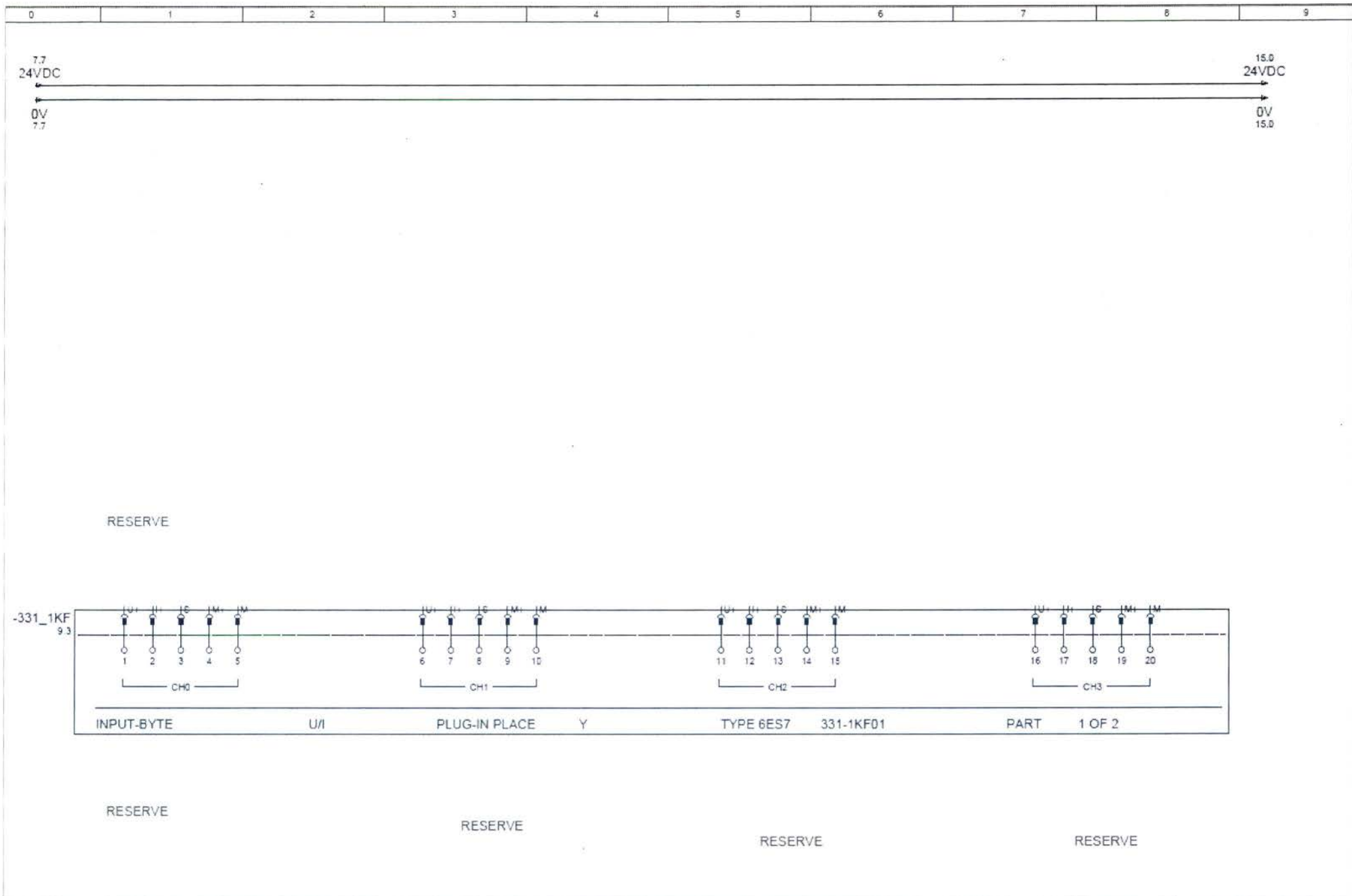
17.1		
17.1	-A1:1.14	EMERGENCY STOP
17.2	-A1:10.0	RESET
17.3	X2:13	RESERVE
17.4	X2:14	RESERVE
17.5	X2:15	RESERVE
17.6	X2:16	RESERVE
17.7	X2:17	RESERVE
17.8	X2:18	RESERVE
18.2	F1:14	RESERVE
18.3	-A1:11.1	ΘΕΡΜΙΚΟ Μ10_BLS000
18.4	X2:22	FEEDBACK BL5000 M10
18.5	S1_1:14	ΤΕΡΜΑΤΙΚΟΣ ΔΙΑΚΟΠΤΗΣ S1_1 NO
18.6	S1_2:16	ΤΕΡΜΑΤΙΚΟΣ ΔΙΑΚΟΠΤΗΣ S1_2 NC
18.7	X2:25	RESERVE
18.8	X2:26	RESERVE
18.8	S1_2:15	

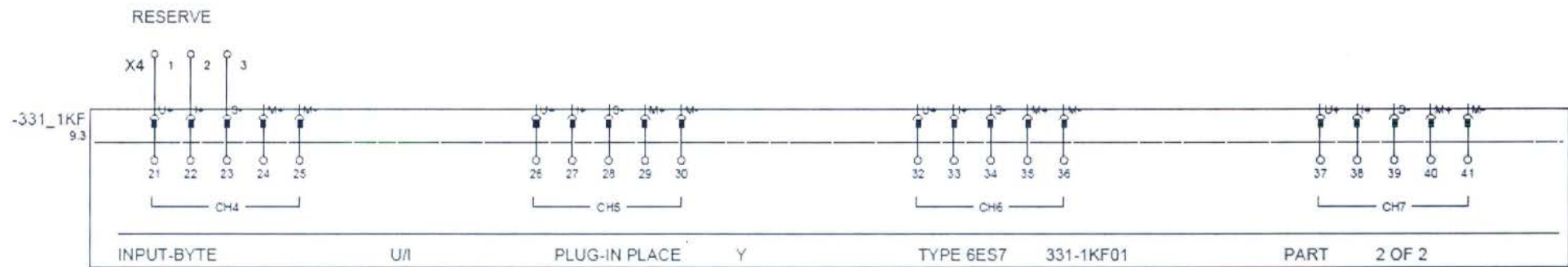
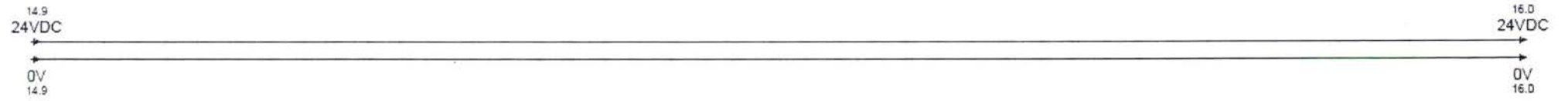


19.1	-A1:10.1	
19.1	S11_1_BLS000:14	ΤΕΡΜΑΤΙΚΟΣ ΔΙΑΚΟΠΤΗΣ S11_1_BLS000 NO
19.2	S11_2_BLS000:16	ΤΕΡΜΑΤΙΚΟΣ ΔΙΑΚΟΠΤΗΣ S11_2_BLS000 NC
19.3	X2:29	RESERVE
19.4	X2:30	RESERVE
19.5	X2:31	RESERVE
19.6	X2:32	RESERVE
19.7	S11_2_BLS000:15	RESERVE
19.8	X2:34	FEEDBACK R5000 M10
20.1	S1_1_R5000:14	ΤΕΡΜΑΤΙΚΟΣ ΔΙΑΚΟΠΤΗΣ S1_1_R5000 NO
20.2	S1_2_R5000:14	ΤΕΡΜΑΤΙΚΟΣ ΔΙΑΚΟΠΤΗΣ S1_2_R5000 NC
20.3	S2_1_R5000:14	ΤΕΡΜΑΤΙΚΟΣ ΔΙΑΚΟΠΤΗΣ S2_1_R5000 NO
20.4	S2_2_R5000:14	ΤΕΡΜΑΤΙΚΟΣ ΔΙΑΚΟΠΤΗΣ S2_2_R5000 NC
20.5	S11_1_R5000:14	ΤΕΡΜΑΤΙΚΟΣ ΔΙΑΚΟΠΤΗΣ S11_1_R5000 NO
20.6	S11_2_R5000:14	ΤΕΡΜΑΤΙΚΟΣ ΔΙΑΚΟΠΤΗΣ S11_2_R5000 NC
20.7	X2:41	RESERVE
20.8	X2:42	RESERVE
20.8	S11_2_R5000:13	







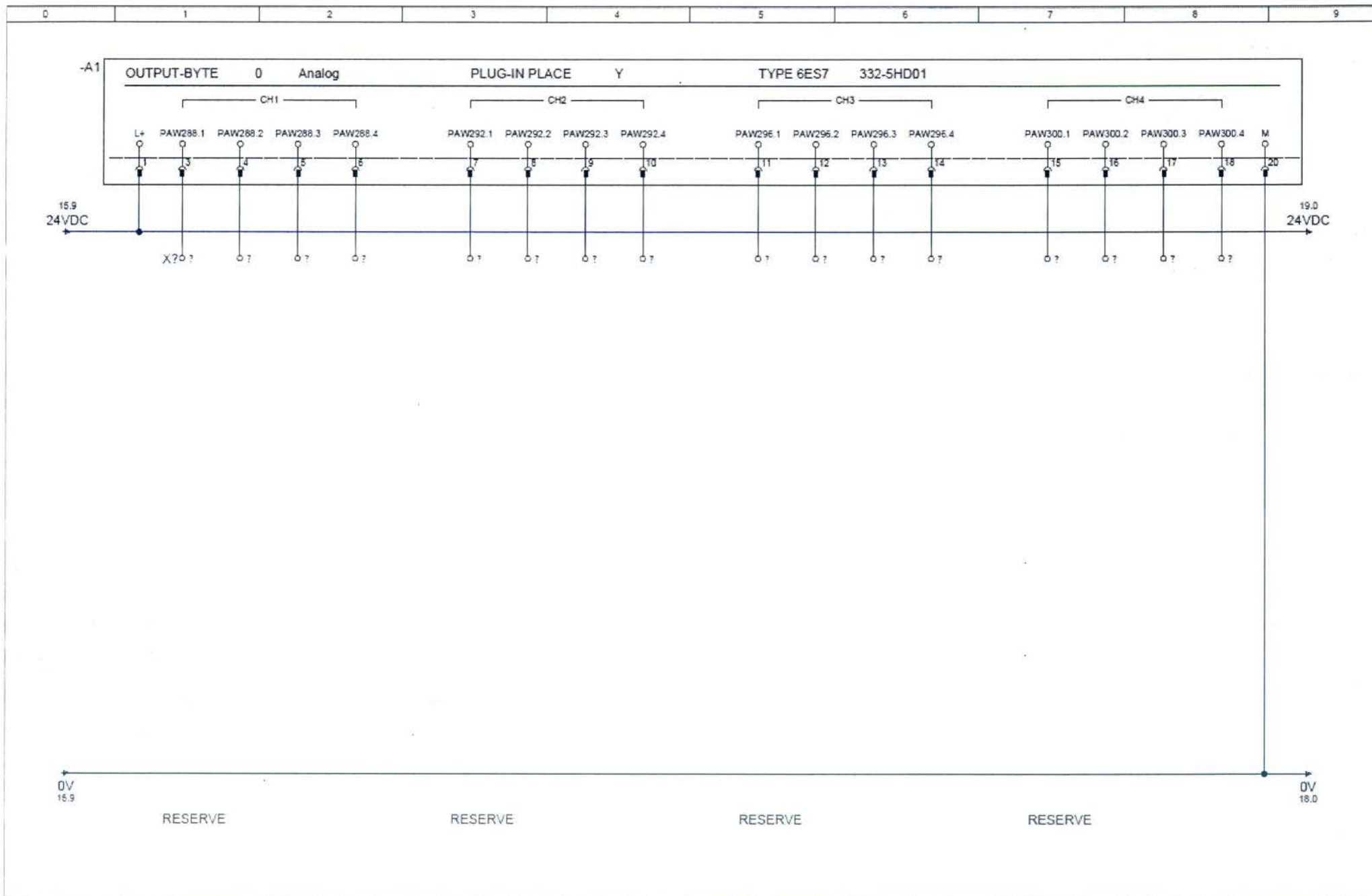


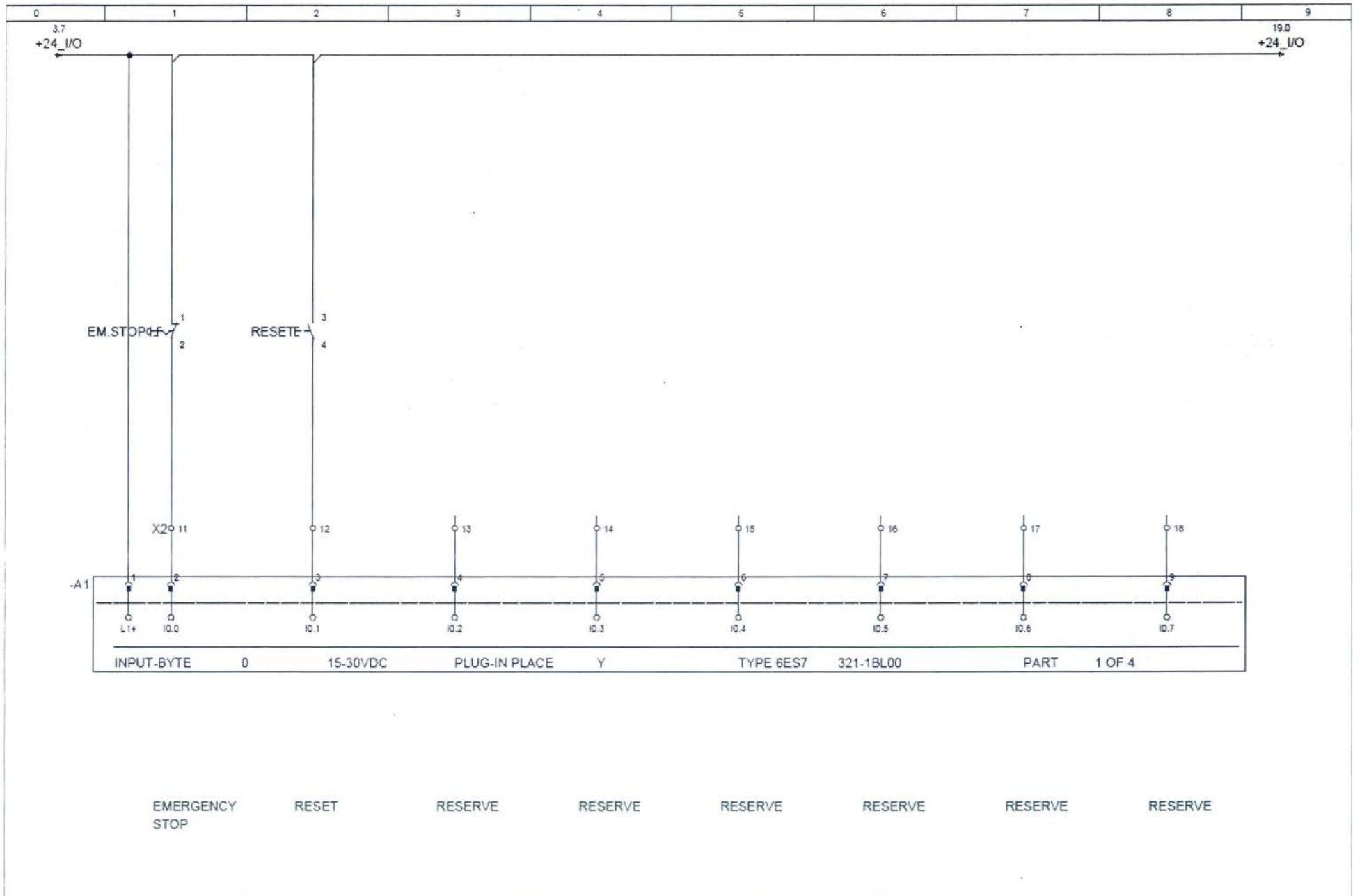
RESERVE

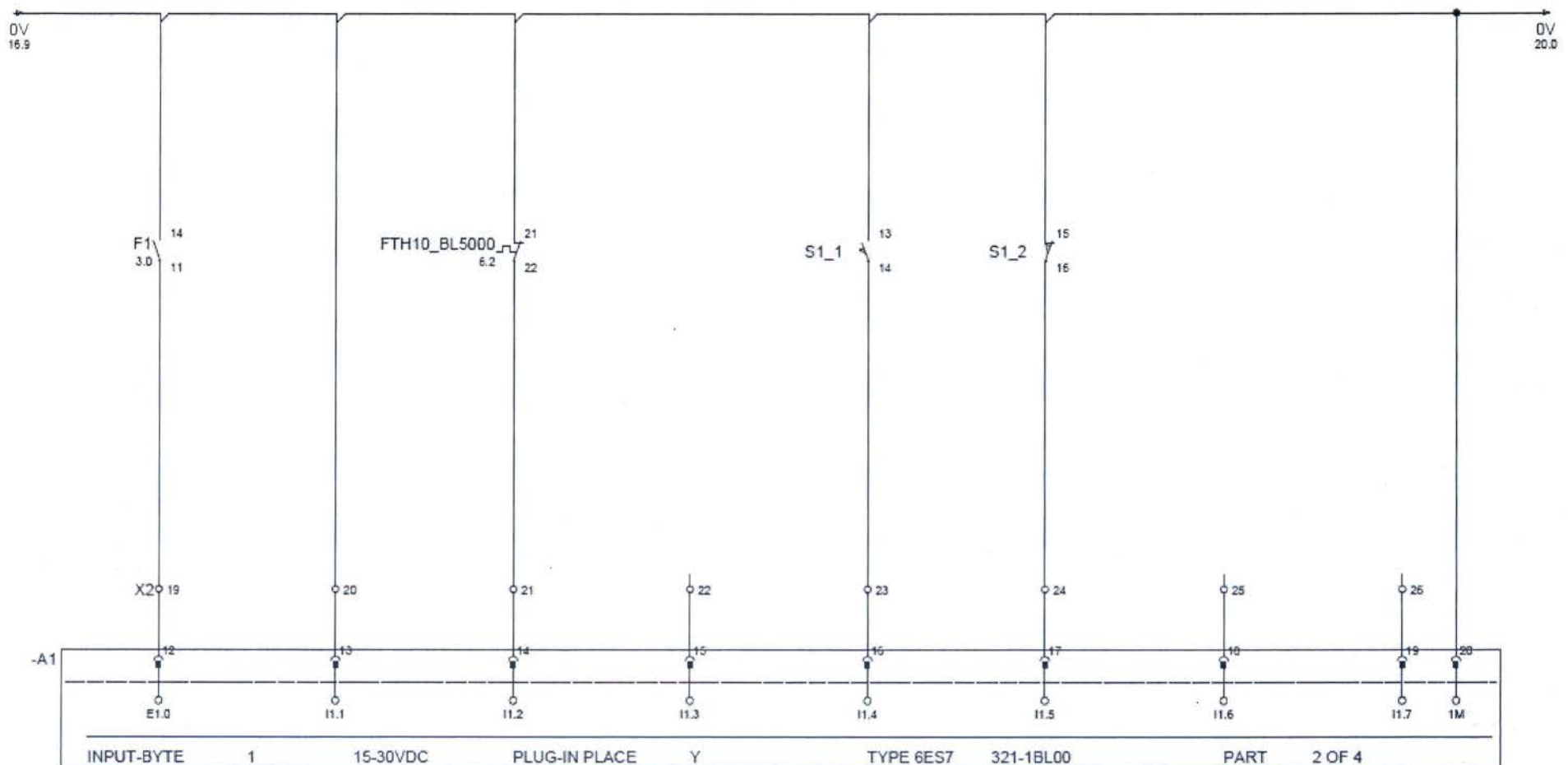
RESERVE

RESERVE

RESERVE







ΠΤΩΣΗ  
ΑΣΦΑΛΕΙΑΣ M1

RESERVE

ΘΕΡΜΙΚΟ  
M10\_BL5000

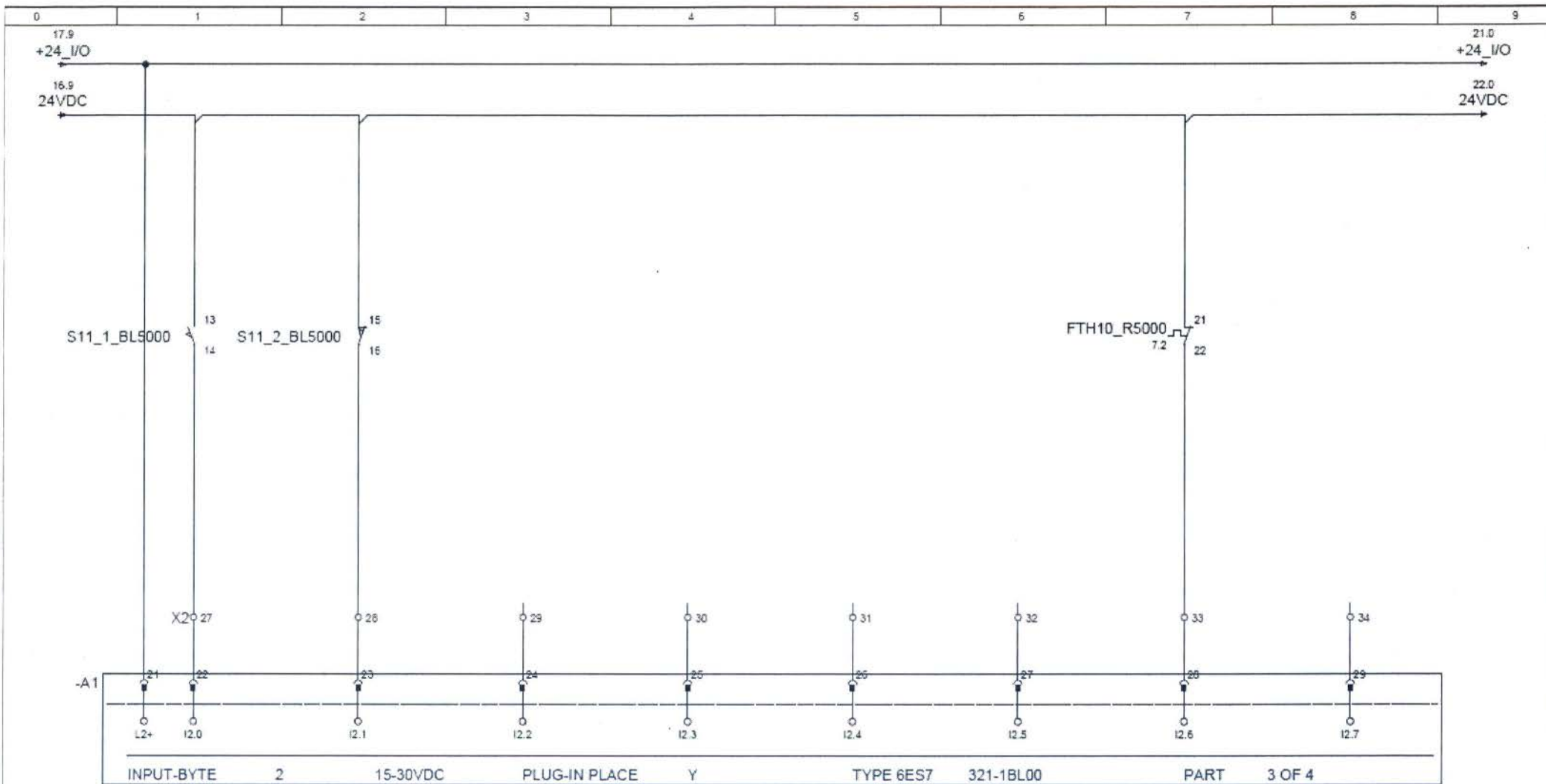
FEEDBACK  
BL5000 M10

ΤΕΡΜΑΤΙΚΟΣ  
ΔΙΑΚΟΠΤΗΣ S1\_1  
NO

ΤΕΡΜΑΤΙΚΟΣ  
ΔΙΑΚΟΠΤΗΣ S1\_2  
NC

RESERVE

RESERVE



ΤΕΡΜΑΤΙΚΟΣ  
ΔΙΑΚΟΠΤΗΣ  
S11\_1\_BL5000  
NO

ΤΕΡΜΑΤΙΚΟΣ  
ΔΙΑΚΟΠΤΗΣ  
S11\_2\_BL5000  
NC

RESERVE

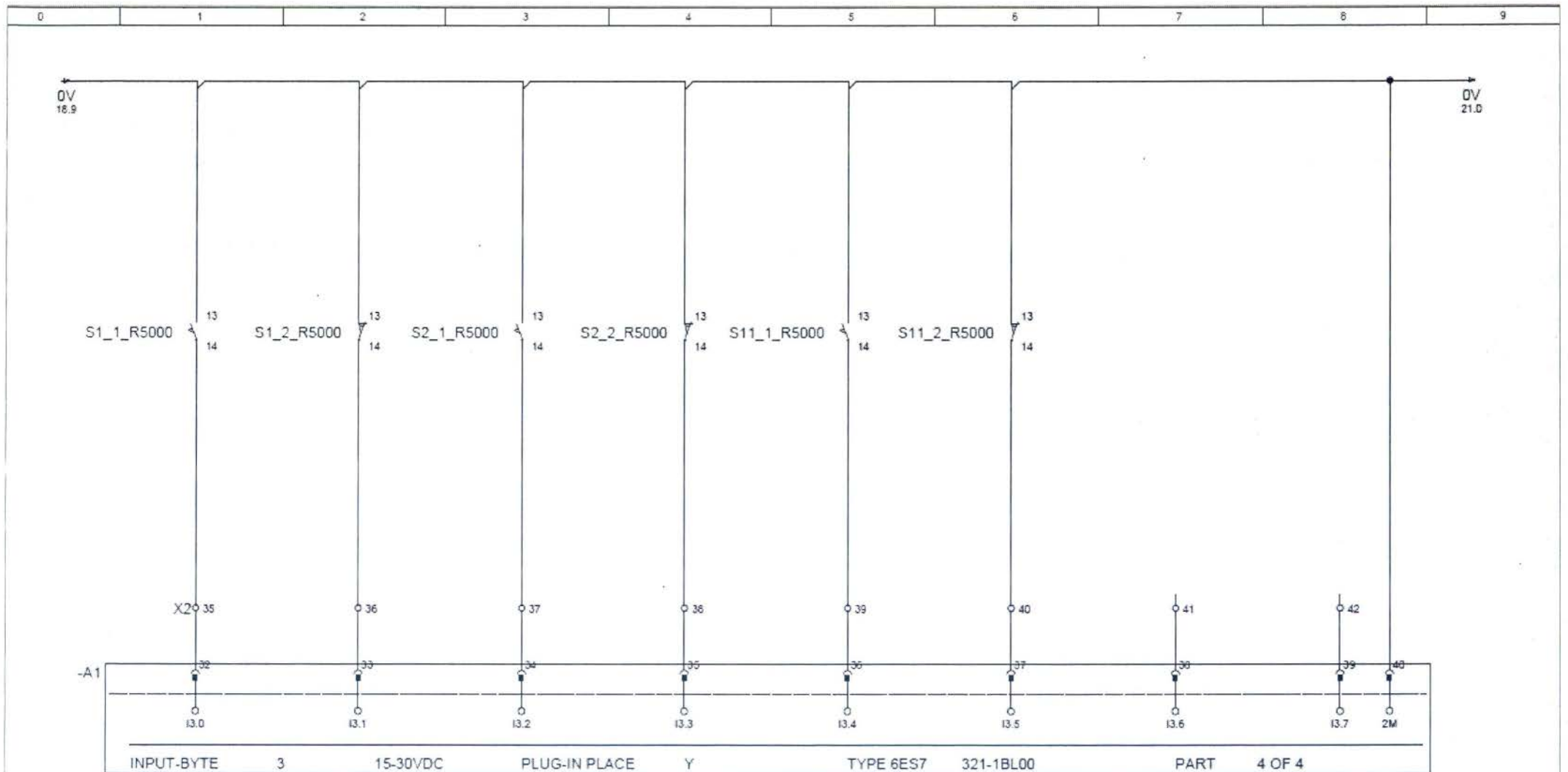
RESERVE

RESERVE

RESERVE

RESERVE

FEEDBACK  
R5000 M10



INPUT-BYTE 3 15-30VDC PLUG-IN PLACE Y TYPE 6ES7 321-1BL00 PART 4 OF 4

ΤΕΡΜΑΤΙΚΟΣ  
ΔΙΑΚΟΠΤΗΣ  
S1\_1\_R5000  
NO

ΤΕΡΜΑΤΙΚΟΣ  
ΔΙΑΚΟΠΤΗΣ  
S1\_2\_R5000  
NC

ΤΕΡΜΑΤΙΚΟΣ  
ΔΙΑΚΟΠΤΗΣ  
S2\_1\_R5000  
NO

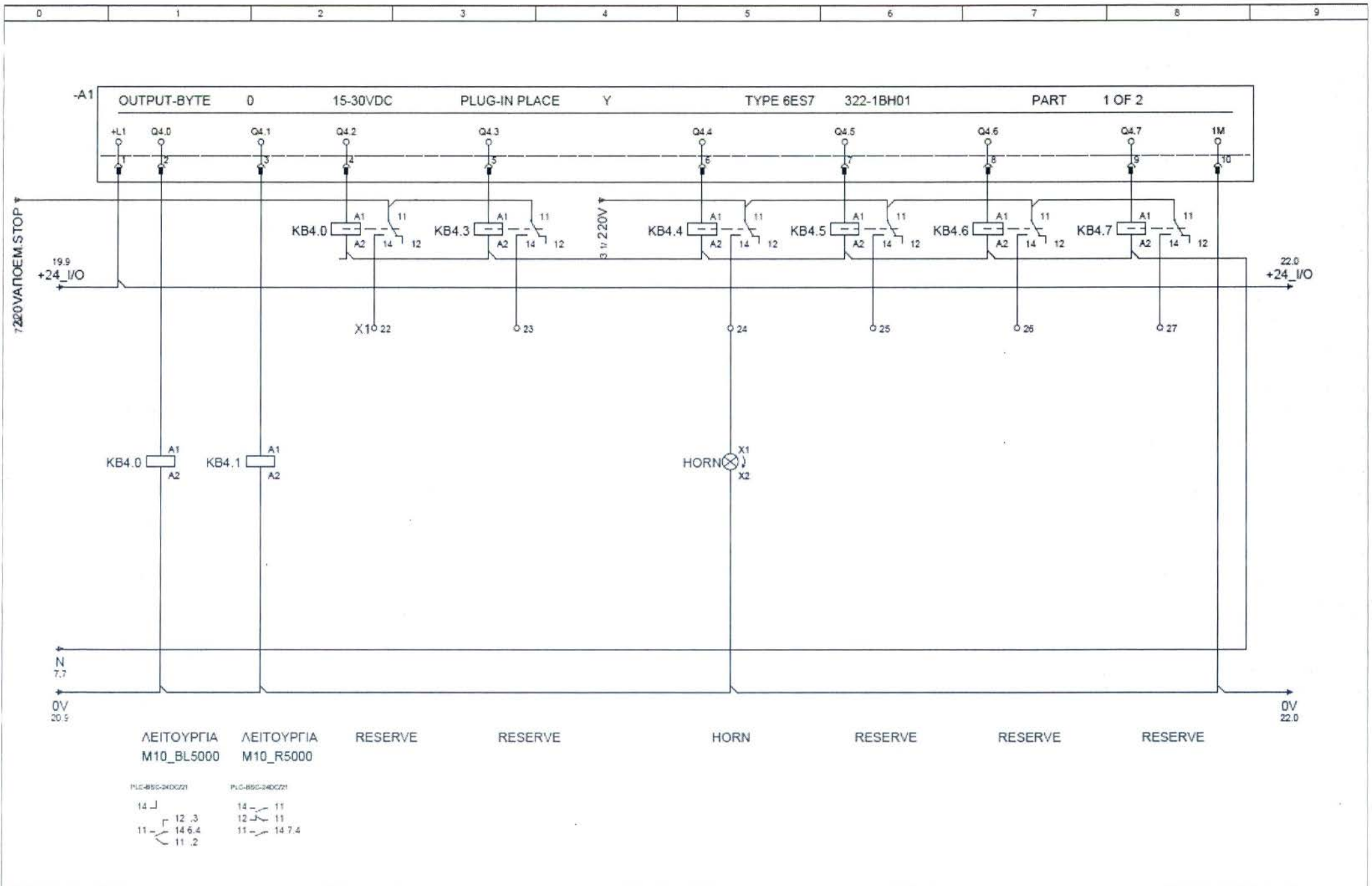
ΤΕΡΜΑΤΙΚΟΣ  
ΔΙΑΚΟΠΤΗΣ  
S2\_2\_R5000  
NC

ΤΕΡΜΑΤΙΚΟΣ  
ΔΙΑΚΟΠΤΗΣ  
S11\_1\_R5000  
NO

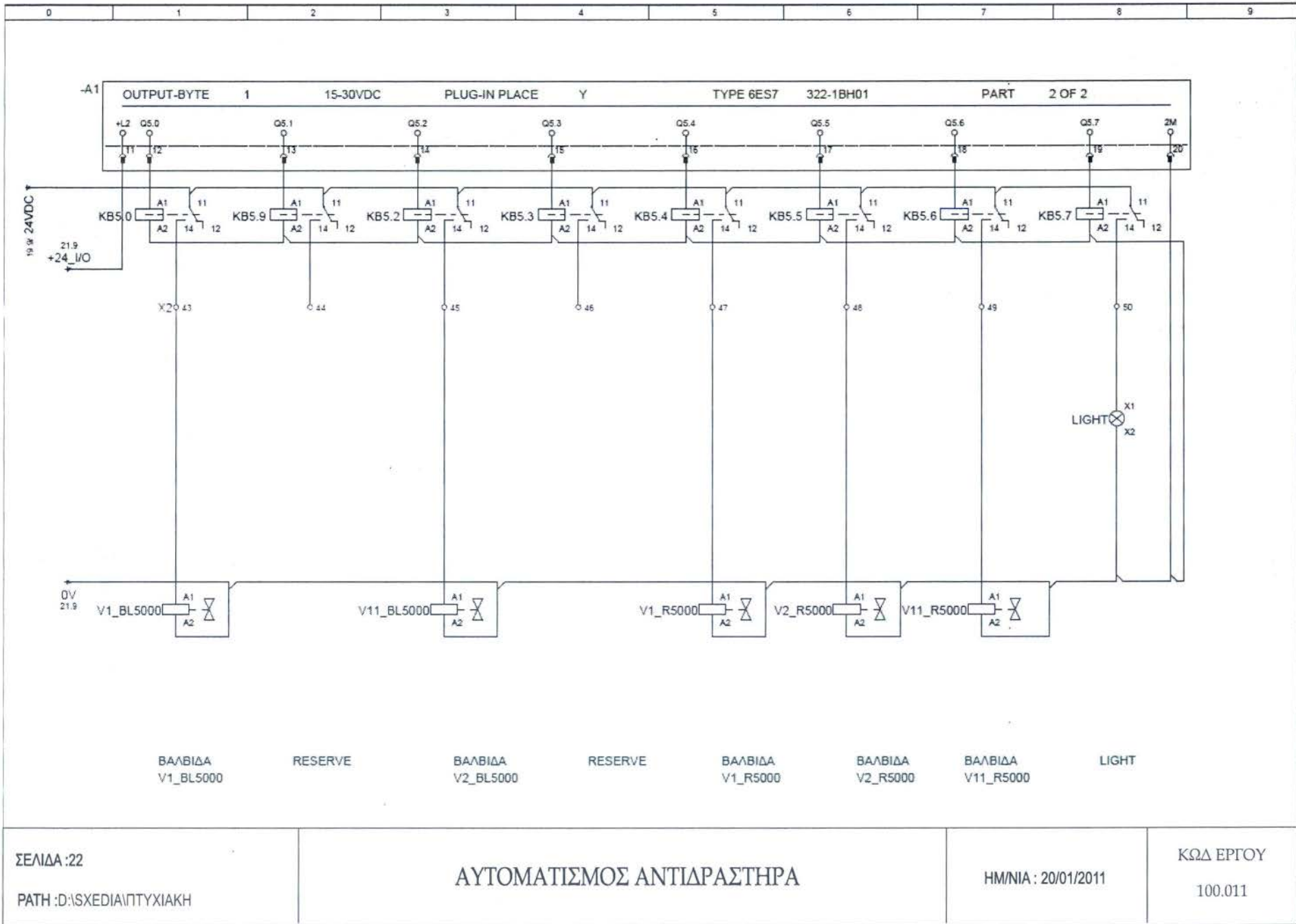
ΤΕΡΜΑΤΙΚΟΣ  
ΔΙΑΚΟΠΤΗΣ  
S11\_2\_R5000  
NC

RESERVE

RESERVE









# ΜΕΡΟΣ 4ο

PLC Software

```
1 ORGANIZATION_BLOCK OB1
2
3 VAR_TEMP
4     HEADER : ARRAY [1..20] OF BYTE ; //20 bytes for opsy
5     dummy : INT;
6 END_VAR
7
8 BEGIN
9
10
11     siwarexU();
12     RD_Cmd();
13     RD_Charging();
14     RW_Dicon();
15     Calc_Analog();
16     Inv_Comm();
17
18
19
20 END_ORGANIZATION_BLOCK
21
```

## 40 - &lt;offline&gt;

"Charging"

Name: George                      Family:  
                                     Version: 0.1  
                                     Block version: 2  
 Date stamp Code:                   08/02/2007 05:06:50 PM  
                   Interface:       08/02/2007 04:12:23 PM  
 Paths (block/logic/data): 00684 00544 00002

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC40

Network: 1

```

CALL "Start_Stop"      FC4
  Start_Stop:="RD".V2_Open  DB102.DBX110.6    -- RD V2 Open Bit
  ON      :=M30.0
  Released :=M30.1
  
```

Network: 2      X2:54

```

A      M      30.0
AN     "RD".V2_Alm  DB102.DBX110.4    -- RD V2 Alarm Bit
=      "V2"        Q25.6             -- X2:54
=      "RD".V2_Run DB102.DBX110.5    -- RD V2 Running Bit
  
```

Network: 3

```

CALL "Start_Stop"      FC4
  Start_Stop:="RD".M2_Start  DB102.DBX59.0    -- M2 Command to Start
  ON      :=M30.2
  Released :=M30.3
  
```

Network: 4

```

ON     "V2"        Q25.6             -- X2:54
ON     "Em_Stop"   I16.0
R      M      30.2
  
```

---

tworok: 5

---

A M 30.2  
A "M2\_Inv\_Br" I16.3  
= "M2\_LI1" Q25.1

---

tworok: 6 M2 Inverter is Running

---

A "M2\_LI1" Q25.1  
A "ATV71\_DB".Inv[2].Read.Operation\_enabled DB50.DBX37.2 -- 3201 Running  
= "RD".M2\_Run DB102.DBX58.1 -- M2 Inverter is Running

---

tworok: 7

---

CALL "Start\_Stop" FC4  
Start\_Stop:="RD".M4\_Start DB102.DBX90.2 -- M4 Command to Start  
ON :=M30.4  
Released :=M30.5

---

tworok: 8 X1: 16,17,18

---

A M 30.4  
AN "Fth\_M4" I16.6  
= "M4" Q24.1 -- X1: 16,17,18  
= "RD".M4\_Run DB102.DBX90.1 -- M4 Run

---

tworok: 9

---

CALL "Start\_Stop" FC4  
Start\_Stop:="RD".V3\_Open DB102.DBX111.2 -- RD V3 Open Bit  
ON :=M31.0  
Released :=M31.1

---

tworok: 10 X2:54

---

A M 31.0  
AN "RD".V3\_Alm DB102.DBX111.0 -- RD V3 Alarm Bit  
= "V3" Q25.7 -- X2:54  
= "RD".V3\_Run DB102.DBX111.1 -- RD V3 Running Bit

---

tworok: 11

---

CALL "Start\_Stop" FC4  
Start\_Stop:="RD".M3\_Start DB102.DBX89.0 -- M3 Command to Start  
ON :=M31.2  
Released :=M31.3

---

---

tworck: 12

---

ON "V3" Q25.7 -- X2:55  
ON "Em\_Stop" I16.0  
R M 31.2

---

---

tworck: 13

---

A M 31.2  
A "M3\_Inv\_Br" I16.4  
= "M3\_LI1" Q25.2

---

---

tworck: 14 M3 Inverter is Running

---

A "M3\_LI1" Q25.2  
A "ATV71\_DB".Inv[3].Read.Operation\_enabled DB50.DBX73.2 -- 3201 Running  
= "RD".M3\_Run DB102.DBX88.1 -- M3 Inverter is Running

---

---

tworck: 15

---

CALL "Start\_Stop" FC4  
Start\_Stop:="RD".M5\_Start DB102.DBX90.6 -- M4 Command to Start  
ON :=M31.4  
Released :=M31.5

---

---

tworck: 16 X1: 19,20,21

---

A M 31.4  
AN "Fth\_M5" I16.7  
= "M5" Q24.2 -- X1: 19,20,21  
= "RD".M5\_Run DB102.DBX90.5 -- M4 Run

---

---

tworck: 17

---

CALL "Start\_Stop" FC4  
Start\_Stop:="RD".V7\_1\_Open DB102.DBX111.6 -- RD V7\_1 Open Bit  
ON :=M32.0  
Released :=M32.1

---

---

tworck: 18 X2:54

---

A M 32.0  
AN "RD".V7\_1\_Alm DB102.DBX111.4 -- RD V7\_1 Alarm Bit  
= "V7\_1" Q25.3 -- X2:51  
= "RD".V7\_1\_Run DB102.DBX111.5 -- RD V7\_1 Running Bit

---

Network: 19

```
CALL "Start_Stop"          FC4
Start_Stop:="RD".V7_2_Open DB102.DBX112.2  -- RD V7_2 Open Bit
ON      :=M32.2
Released :=M32.3
```

Network: 20 X2:54

```
A      M      32.2
AN     "RD".V7_2_Alm DB102.DBX112.0  -- RD V7_2 Alarm Bit
=      "V7_2"      Q25.4             -- X2:52
=      "RD".V7_2_Run DB102.DBX112.1  -- RD V7_2 Running Bit
```

Network: 21

```
CALL "Start_Stop"          FC4
Start_Stop:="RD".V7_3_Open DB102.DBX112.6  -- RD V7_3 Open Bit
ON      :=M32.4
Released :=M32.5
```

Network: 22 X2:54

```
A      M      32.4
AN     "RD".V7_3_Alm DB102.DBX112.4  -- RD V7_3 Alarm Bit
=      "V7_3"      Q25.5             -- X2:53
=      "RD".V7_3_Run DB102.DBX112.5  -- RD V7_3 Running Bit
```

Network: 23

```
CALL "Start_Stop"          FC4
Start_Stop:="RD".M7_Start  DB102.DBX91.6   -- M4 Command to Start
ON      :=M32.6
Released :=M32.7
```

Network: 24

```
O(
AN     "V7_1"      Q25.3             -- X2:51
AN     "V7_2"      Q25.4             -- X2:52
AN     "V7_3"      Q25.5             -- X2:53
)
ON     "Em_Stop"   I16.0
R      M      32.6
```

Network: 25 X1: 28,29,30

```
A      M      32.6
AN     "Fth_M7"    I17.2             -- X2:11
=      "M7"        Q24.5             -- X1: 28,29,30
=      "RD".M7_Run DB102.DBX91.5   -- M4 Run
```

## 30 - &lt;offline&gt;

\_Cmd"  
 ie:  
 hor: George           Family:  
                       Version: 0.1  
                       Block version: 2  
 ie stamp Code:       09/06/2007 12:57:24 PM  
                   Interface: 08/30/2005 11:01:57 AM  
 gths (block/logic/data): 00806 00640 00018

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
Controller_OUT	Real	0.0	
Out	Int	4.0	
Ret_Value	Word	6.0	
RETURN		0.0	
RET_VAL		0.0	

---

Block: FC30

---

Network: 1       OFF CHOICE

---

```

O(
L   "RD".Mode  DB102.DBW102    -- 0=OFF, 1=Manual, 2=Auto, 3=Combi
L   0
==I
)
=   "RD_OFF"   M18.0

```

---

Network: 2       MANUAL CHOICE

---

```

O(
L   "RD".Mode  DB102.DBW102    -- 0=OFF, 1=Manual, 2=Auto, 3=Combi
L   1
==I
)
=   "RD_MANUAL" M18.1

```

---

Network: 3       AUTOMATIC CHOICE

---

```

L   "RD".Mode  DB102.DBW102    -- 0=OFF, 1=Manual, 2=Auto, 3=Combi
L   2
==I
=   "RD_AUTO"  M18.2

```



---

 twork: 4 COMBI CHOICE
 

---

```

L   "RD".Mode      DB102.DBW102      -- 0=OFF, 1=Manual, 2=Auto, 3=Combi
L   3
==I
=   "RD_COMBI"    M18.3

```

---

 twork: 5 MANUAL OPERATION
 

---

```

O   "RD_OFF"      M18.0
O   "RD_MANUAL"   M18.1
=   "Temp".RD_T1a.Command.Manual_Operation DB11.DBX22.2  -- MANUAL OPERATION
=   "Temp".RD_T2a.Command.Manual_Operation DB11.DBX60.2  -- MANUAL OPERATION

```

---

 twork: 6 AUTOMATIC OPERATION
 

---

```

O   "RD_AUTO"     M18.2
O   "RD_COMBI"    M18.3
=   "Temp".RD_T1a.Command.Automatic_Operation DB11.DBX22.3  -- AUTOMATIC OPERATION
=   "Temp".RD_T2a.Command.Automatic_Operation DB11.DBX60.3  -- AUTOMATIC OPERATION

```

---

 twork: 7 RD Heating Mode=1, Cooling Mode=0
 

---

```

A   "RD_COMBI"    M18.3
AN(
L   "DB90".Anal[9].Output DB90.DBD772  -- Output Value
L   "DB90".Anal[9].SP[1]  DB90.DBD778  -- Set Point
>R
)
S   "RD".Heat_ON  DB102.DBX100.0  -- RD Heating Mode=1, Cooling Mode=0

A   "RD_COMBI"    M18.3
A(
L   "DB90".Anal[9].Output DB90.DBD772  -- Output Value
L   "DB90".Anal[9].SP[1]  DB90.DBD778  -- Set Point
>R
)
R   "RD".Heat_ON  DB102.DBX100.0  -- RD Heating Mode=1, Cooling Mode=0

```

---

 twork: 8
 

---

```

A   "Temp".RD_T1a.Command.Manual_Operation DB11.DBX22.2  -- MANUAL OPERATION
A   "Temp".RD_T1a.Status.Manual           DB11.DBX1.0      -- Manual Operation = 1
JNB _001

L   "RD".V10_SP  DB102.DBD134  -- -100.00 to +100.00%
T   "Temp".RD_T1a.Command.Controller_Man_Out DB11.DBD28     -- Controller Manual Output

```

l: NOP 0

---

 twork: 9
 

---

```

A   "Temp".RD_T2a.Command.Manual_Operation DB11.DBX60.2  -- MANUAL OPERATION
A   "Temp".RD_T2a.Status.Manual           DB11.DBX39.0     -- Manual Operation = 1
JNB _002

L   "RD".V15_SP  DB102.DBD142  -- -100.00 to +100.00%
T   "Temp".RD_T2a.Command.Controller_Man_Out DB11.DBD66     -- Controller Manual Output

```

2: NOP 0

twork: 10 X2:46

```

AN  "RD".Heat_ON          DB102.DBX100.0  -- RD Heating Mode=1,  Cooling Mode=0
A(
O   "RD_MANUAL"          M18.1
O   "RD_AUTO"            M18.2
O   "RD_COMBI"           M18.3
)
A(
L   "DB90".Anal[9].Output DB90.DBD772  -- Output Value
L   "DB90".Anal[9].SP[1] DB90.DBD778  -- Set Point
>R
)
=   "RD_V12_V13"         Q26.0          -- X2:56
=   "RD".V12_V13_Run     DB102.DBX120.5  -- RD V12_V13 Running Bit

```

twork: 11

```

A   "Em_Stop"           I16.0
AN  "Fth_M12"           I17.0
A   "RD_V12_V13"       Q26.0          -- X2:56
L   S5T#10S
SD  T    12

```

twork: 12 X1: 22,23,24

```

A   T    12
=   "M12_RD"           Q24.3          -- X1: 22,23,24

```

twork: 13 M12 Run

```

A   "M12_RD"           Q24.3          -- X1: 22,23,24
=   "RD".M12_Run       DB102.DBX92.5  -- M4 Run

```

twork: 14

```

L   "Temp".RD_Tla.Status.Controller_Out DB11.DBD10  -- Controller Output
ABS
T   #Controller_OUT
T   "RD".V10_Actual     DB102.DBD130  -- -100.00 to +100.00%

```

twork: 15

```

O   "RD_OFF"           M18.0
JNB _005
L   0.000000e+000
T   #Controller_OUT
T   "RD".V10_Actual     DB102.DBD130  -- -100.00 to +100.00%
5: NOP 0

```

twork: 16

```
CALL "UNSCALE"      FC106      -- Unscaling Values
  IN      :=#Controller_OUT
  HI_LIM :=1.000000e+002
  LO_LIM :=0.000000e+000
  BIPOLAR:=FALSE
  RET_VAL:=#Ret_Value
  OUT     :=#Out
NOP      0
```

twork: 17

```
L      #Out
T      MW      38
T      "RD_V10" PQW304      -- Tree Way Valve Water
```

twork: 18

```
A      "RD".Heat_ON      DB102.DBX100.0      -- RD Heating Mode=1, Cooling Mode=0
JNB    _003
L      "Temp".RD_T2a.Status.Controller_Out DB11.DBD48      -- Controller Output
ABS
T      #Controller_OUT
T      "RD".V15_Actual    DB102.DBD138      -- -100.00 to +100.00%
3: NOP 0
```

twork: 19

```
ON     "RD".Heat_ON      DB102.DBX100.0      -- RD Heating Mode=1, Cooling Mode=0
O      "RD_OFF"          M18.0
JNB    _004
L      0.000000e+000
T      #Controller_OUT
T      "RD".V15_Actual    DB102.DBD138      -- -100.00 to +100.00%
4: NOP 0
```

twork: 20

```
CALL "UNSCALE"      FC106      -- Unscaling Values
  IN      :=#Controller_OUT
  HI_LIM :=1.000000e+002
  LO_LIM :=0.000000e+000
  BIPOLAR:=FALSE
  RET_VAL:=#Ret_Value
  OUT     :=#Out
NOP      0
```

twork: 21

```
L      #Out
T      MW      40
T      "RD_V15" PQW306      -- Two Way Valve Steam
```

---

Network: 22 X2:44

---

```
A "Em_Stop" I16.0
A "RD".Heat_ON DB102.DBX100.0 -- RD Heating Mode=1, Cooling Mode=0
A(
O "RD_MANUAL" M18.1
O "RD_AUTO" M18.2
O "RD_COMBI" M18.3
)
= "RD_V11" Q26.1 -- X2:57
```

---

Network: 23 RD V11 Running Bit

---

```
A "RD_V11" Q26.1 -- X2:57
= "RD".V11_Run DB102.DBX120.1 -- RD V11 Running Bit
```

---

Network: 24

---

```
A "Em_Stop" I16.0
AN "Fth_M11" I17.1
A "RD_V11" Q26.1 -- X2:57
L S5T#10S
SD T 11
```

---

Network: 25 X1: 25,26,27

---

```
A T 11
= "M11_RD" Q24.4 -- X1: 25,26,27
```

---

Network: 26 M4 Run

---

```
A "M11_RD" Q24.4 -- X1: 25,26,27
= "RD".M11_Run DB102.DBX92.1 -- M4 Run
```

---

Network: 27

---

```
CALL "Start_Stop" FC4
Start_Stop:="RD".M1_Start DB102.DBX29.0 -- M1 Command to Start
ON :=M20.0
Released :=M20.1
```

---

Network: 28

---

```
A M 20.0
A "M1_Inv_Br" I16.2
= "M1_L11" Q25.0
```

Network: 29 M1 Inverter is Running

---

A	"M1_LI1"	Q25.0	
A	"ATV71_DB".Inv[1].Read.Operation_enabled	DB50.DBX1.2	-- 3201 Running
=	"RD".M1_Run	DB102.DBX28.1	-- M1 Inverter is Running

---

Network: 30

---

BE

## 4 - &lt;offline&gt;

art\_Stop"

Name: George                      Family:  
                                     Version: 0.1  
                                     Block version: 2  
 Date stamp Code:                07/04/2006 06:19:18 PM  
                     Interface:    08/31/2004 11:07:16 AM  
 Lengths (block/logic/data): 00154 00050 00000

Name	Data Type	Address	Comment
IN		0.0	
Start_Stop	Bool	0.0	
OUT		0.0	
IN_OUT		0.0	
ON	Bool	2.0	
Released	Bool	2.1	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

---

 Block: FC4
 

---



---

 Network: 1
 

---

```

#Start_
Stop      #ON          #Released
  /-----/-----/ (S)

```

---

 Network: 2
 

---

```

#Start_
Stop      #ON          #Released
  /-----/-----/ (R)

```

---

 Network: 3
 

---

```

#Start_
Stop      #Released    #ON
  /-----/-----/ (S)

```

twork: 4

---



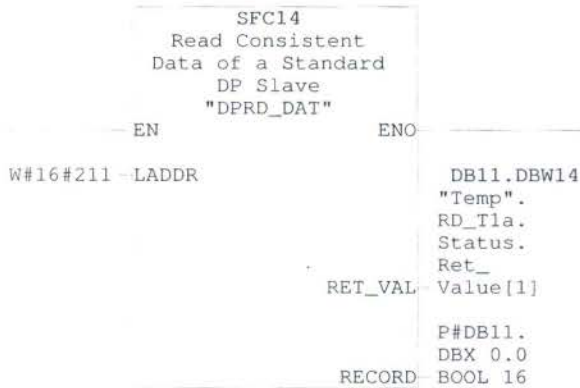
1 - <offline>

'\_Dicon" Read Write Jumo Dicon400 through Profibus  
 Name: George Family:  
 Version: 0.1  
 Block version: 2  
 Date stamp Code: 08/01/2007 02:20:47 PM  
 Interface: 09/03/1998 08:09:28 AM  
 Paths (block/logic/data): 01412 01242 00016

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

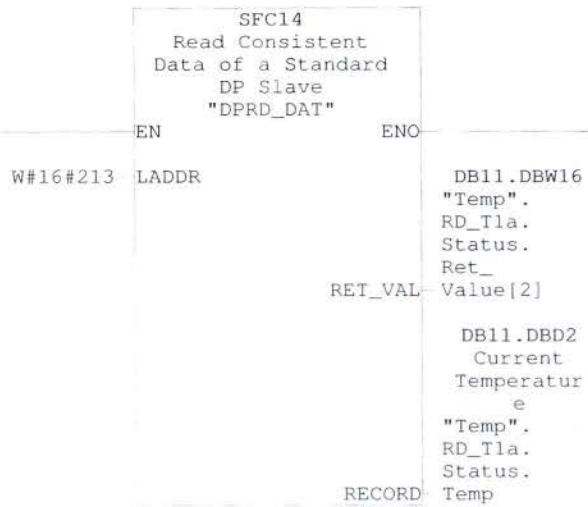
Block: FC1

Network: 1

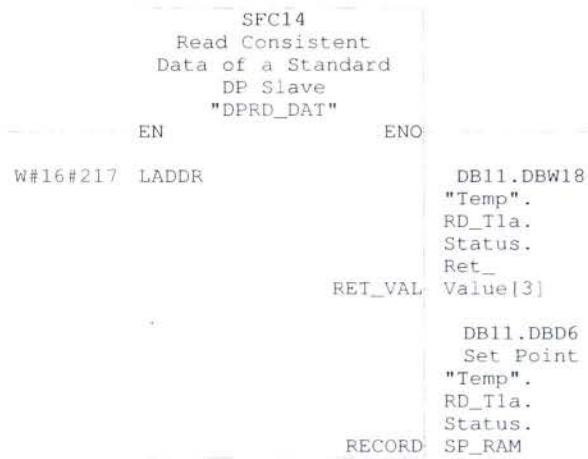




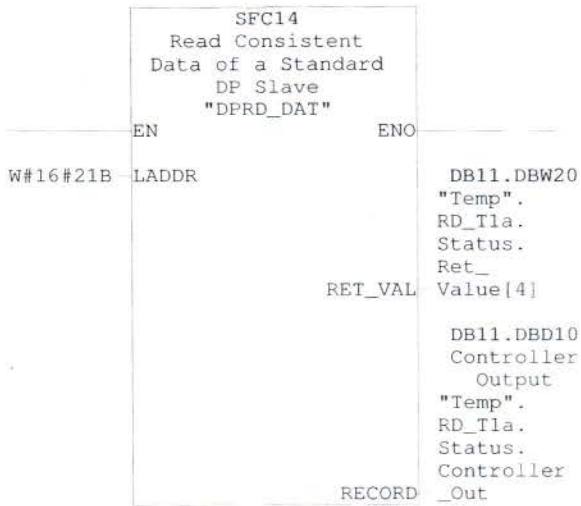
twork: 2



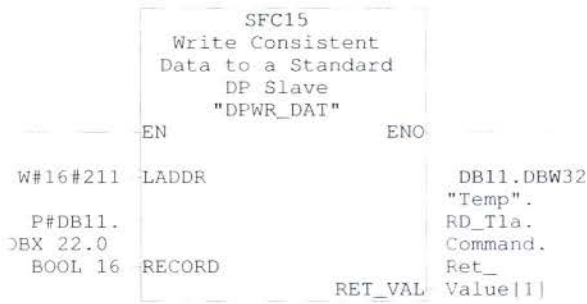
twork: 3



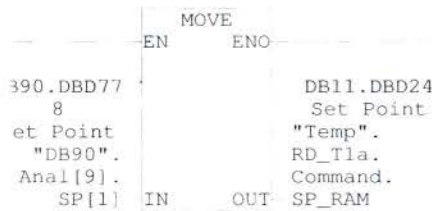
Network: 4



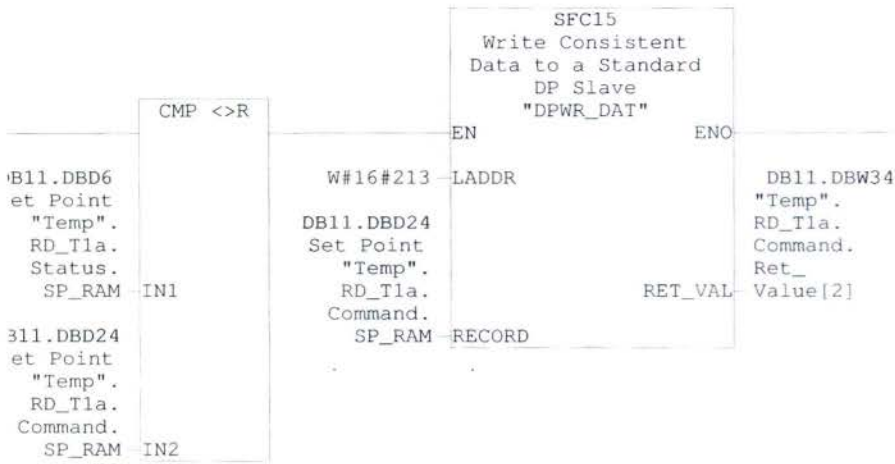
Network: 5



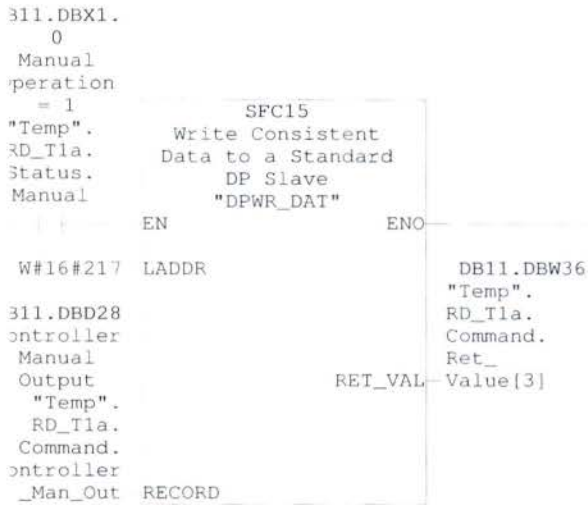
Network: 6



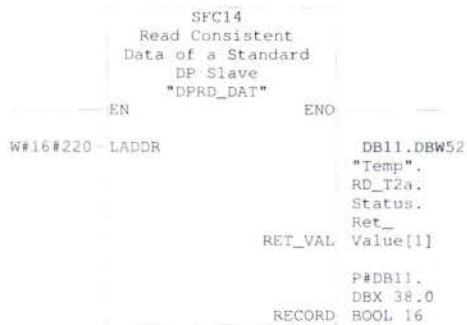
Network: 7



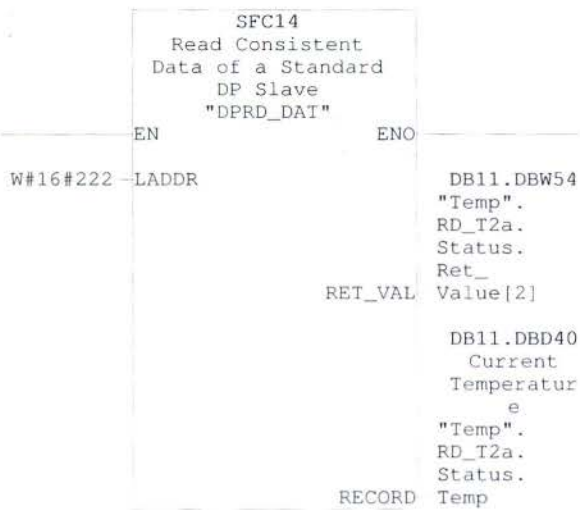
Network: 8



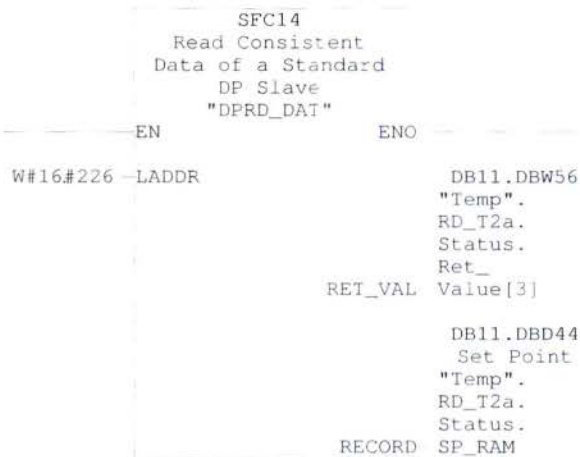
Network: 9



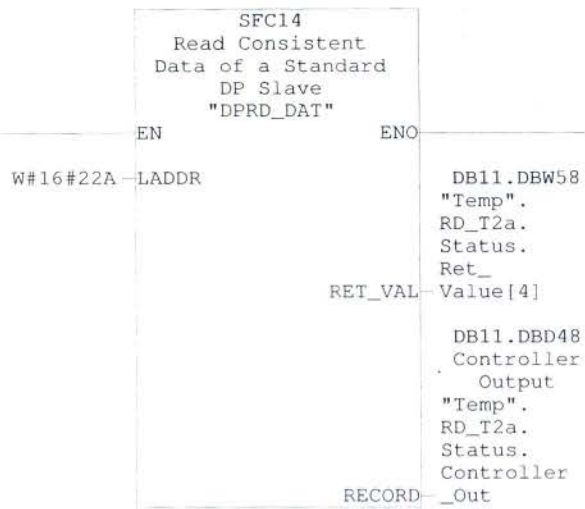
Network: 10



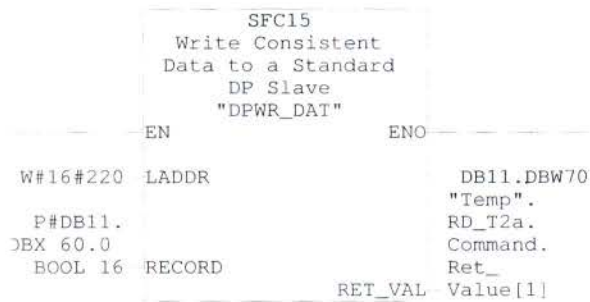
Network: 11



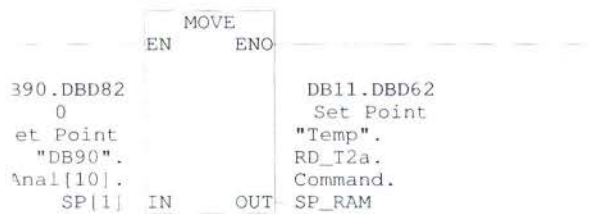
twork: 12



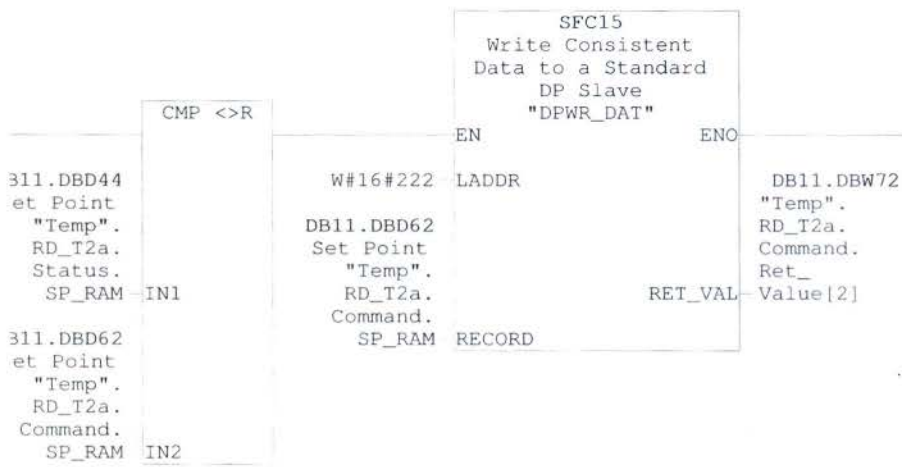
twork: 13



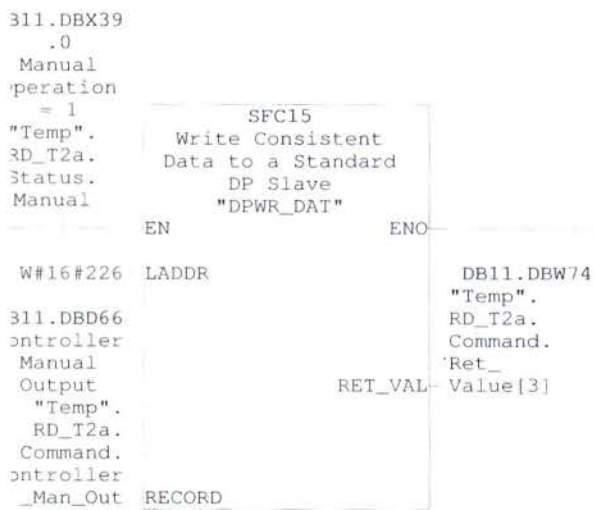
twork: 14



Network: 15



Network: 16



## 90 - &lt;offline&gt;

lc\_Analog"

```

ie:
hor: George
ie stamp Code:
Interface:
gths (block/logic/data): 00804 00690 00022

```

```

Family:
Version: 0.1
Block version: 2

```

```

08/01/2007 01:03:15 PM
08/12/2005 10:08:23 AM

```

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

---

 block: FC90
 

---

 network: 1
 

---

 V 288 Analog Input 4-20mA from Liter Meter Big
 

---

```

CALL "Analog"          FC92
  PIW      := "Liter_Meter_Big"    PIW288
  Max_Div := "DB90".Anal[1].Max_Div  DB90.DBW420  -- Maximun Divisions for maximum Value
  Min_Div := "DB90".Anal[1].Min_Div  DB90.DBW422  -- Minimum Divisions for minimum Value
  Hi_Real := "DB90".Anal[1].Hi_Real  DB90.DBD424  -- Maximum Output for Maximum Analog Input Value
  Low_Real:= "DB90".Anal[1].Low_Real  DB90.DBD428  -- Minimum Output for Minimum Analog Input Value
  Alm_Prc := "DB90".Anal[1].Alm_prc  DB90.DBD432  -- Percentage Underrange and Overrange Alarm
  Output  := "DB90".Anal[1].Output   DB90.DBD436  -- Output Value
  Under   := "DB90".Anal[1].Under    DB90.DBX440.0 -- Underrange Scale Alarm Bit
  Over    := "DB90".Anal[1].Over     DB90.DBX440.1 -- Overrange Scale Alarm Bit
NOP      0

```

 network: 2
 

---

 V 290 Analog Input 4-20mA from Liter Meter Small
 

---

```

CALL "Analog"          FC92
  PIW      := "Liter_Meter_Small"    PIW290
  Max_Div := "DB90".Anal[2].Max_Div  DB90.DBW462  -- Maximun Divisions for maximum Value
  Min_Div := "DB90".Anal[2].Min_Div  DB90.DBW464  -- Minimum Divisions for minimum Value
  Hi_Real := "DB90".Anal[2].Hi_Real  DB90.DBD466  -- Maximum Output for Maximum Analog Input Value
  Low_Real:= "DB90".Anal[2].Low_Real  DB90.DBD470  -- Minimum Output for Minimum Analog Input Value
  Alm_Prc := "DB90".Anal[2].Alm_prc  DB90.DBD474  -- Percentage Underrange and Overrange Alarm
  Output  := "DB90".Anal[2].Output   DB90.DBD478  -- Output Value
  Under   := "DB90".Anal[2].Under    DB90.DBX482.0 -- Underrange Scale Alarm Bit
  Over    := "DB90".Anal[2].Over     DB90.DBX482.1 -- Overrange Scale Alarm Bit
NOP      0

```

 network: 3
 

---

 V 292 Analog Input 4-20mA from T3 RD Temperature Exhaust Pipe
 

---

```

CALL "Analog"          FC92
  PIW      := "RD_T3"                PIW292
  Max_Div := "DB90".Anal[3].Max_Div  DB90.DBW504  -- Maximun Divisions for maximum Value
  Min_Div := "DB90".Anal[3].Min_Div  DB90.DBW506  -- Minimum Divisions for minimum Value
  Hi_Real := "DB90".Anal[3].Hi_Real  DB90.DBD508  -- Maximum Output for Maximum Analog Input Value
  Low_Real:= "DB90".Anal[3].Low_Real  DB90.DBD512  -- Minimum Output for Minimum Analog Input Value
  Alm_Prc := "DB90".Anal[3].Alm_prc  DB90.DBD516  -- Percentage Underrange and Overrange Alarm
  Output  := "DB90".Anal[3].Output   DB90.DBD520  -- Output Value
  Under   := "DB90".Anal[3].Under    DB90.DBX524.0 -- Underrange Scale Alarm Bit

```

```
Over      := "DB90".Anal[3].Over      DB90.DBX524.1      -- Overange Scale Alarm Bit
NOP      0
```

---

```
Network: 4
```

---

```
√ 294 Analog Input 4-20mA from T1b RD Temperature Material Spare
```

```
CALL "Analog"          FC92
PIW      := "RD_T1b"    PIW294
Max_Div := "DB90".Anal[4].Max_Div  DB90.DBW546      -- Maximun Divisions for maximum Value
Min_Div := "DB90".Anal[4].Min_Div  DB90.DBW548      -- Minimum Divisions for minimum Value
Hi_Real := "DB90".Anal[4].Hi_Real  DB90.DBD550      -- Maximum Output for Maximum Analog Input Value
Low_Real := "DB90".Anal[4].Low_Real DB90.DBD554      -- Minimum Output for Minimum Analog Input Value
Alm_Prc := "DB90".Anal[4].Alm_prc  DB90.DBD558      -- Percentage Underange and Overange Alarm
Output  := "DB90".Anal[4].Output   DB90.DBD562      -- Output Value
Under   := "DB90".Anal[4].Under    DB90.DBX566.0    -- Underange Scale Alarm Bit
Over    := "DB90".Anal[4].Over     DB90.DBX566.1    -- Overange Scale Alarm Bit
NOP      0
```

---

```
Network: 5
```

---

```
√ 296 Analog Input 4-20mA from T2b RD Heating/Cooling Output Jacket
perature
```

```
CALL "Analog"          FC92
PIW      := "RD_T2b"    PIW296
Max_Div := "DB90".Anal[5].Max_Div  DB90.DBW588      -- Maximun Divisions for maximum Value
Min_Div := "DB90".Anal[5].Min_Div  DB90.DBW590      -- Minimum Divisions for minimum Value
Hi_Real := "DB90".Anal[5].Hi_Real  DB90.DBD592      -- Maximum Output for Maximum Analog Input Value
Low_Real := "DB90".Anal[5].Low_Real DB90.DBD596      -- Minimum Output for Minimum Analog Input Value
Alm_Prc := "DB90".Anal[5].Alm_prc  DB90.DBD600      -- Percentage Underange and Overange Alarm
Output  := "DB90".Anal[5].Output   DB90.DBD604      -- Output Value
Under   := "DB90".Anal[5].Under    DB90.DBX608.0    -- Underange Scale Alarm Bit
Over    := "DB90".Anal[5].Over     DB90.DBX608.1    -- Overange Scale Alarm Bit
NOP      0
```

---

```
Network: 6
```

---

```
√ fibus DP from T1a Reactor RD Material Temperature
```

```
L      "Temp".RD_T1a.Status.Temp  DB11.DBD2        -- Current Temperature
T      "DB90".Anal[9].Output      DB90.DBD772      -- Output Value
NOP      0
```

---

```
Network: 7
```

---

```
√ fibus DP from T2a Reactor RD Steam Temperature
```

```
L      "Temp".RD_T2a.Status.Temp  DB11.DBD40       -- Current Temperature
T      "DB90".Anal[10].Output     DB90.DBD814      -- Output Value
NOP      0
```





## ΠΗΓΕΣ

### Βιβλιογραφία

- Εργαστηριακές ασκήσεις Ηλεκτρικής Κίνησης Με Συνοπτική Θεωρία (Μαλατέστας/ Βυλλιώτης)
- Σχεδίαση Εφαρμογών Αυτοματισμού με τη γλώσσα STEP 7 σε STL και SCL (Hans Berger)
- <http://www.btelectric.gr/?page=scada>
- [www.metadosi-ischios.gr](http://www.metadosi-ischios.gr)
- Σύγχρονη Τεχνική επιθεώρηση (τεύχος 217)
- <http://www.siemens.gr>