

# Κεφάλαιο 1

## Εισαγωγή

Δίπλα στον εντυπωσιακό όγκο της υπάρχουσας βιβλιογραφίας σχετικά με τα συστήματα P.L.C. και την γλώσσα τους, ένα ακόμη βιβλίο θα φανόταν ίσως περιττό.

Παρόλα αυτά ο σκοπός αυτού του βιβλίου είναι διπτός. Απο την μια σαν σκοπό έχει να υπενθυμίσει στον αναγνώστη ορισμένα θέματα γύρω από την τεχνολογία P.L.C., τόσο σε επίπεδο υλικού (hardware), όσο και σε επίπεδο λογισμικού (software), και την γλώσσα που αυτή χρησιμοποιεί, προκειμένου να μπορέσει να προχωρήσει στην βαθύτερη κατανόηση του αντικειμένου μέσω παραδειγμάτων και εφαρμογών βήμα προς βήμα.

Από την άλλη σαν σκοπό έχει να καταδείξει το ίδιο το λογισμικό, δηλαδή το πώς και κάτω απο ποιες συνθήκες δημιουργείται ένα πρόγραμμα, τι προβλήματα μπορεί να παρουσιάσουν κατά την πορεία της δημιουργίας ενός προγράμματος και πώς μπορούμε να τα αντιμετωπίσουμε, όπως και το να δικαιολογηθούν οι όποιες επιλογές γίνονται για την ολοκλήρωση ενός προγράμματος.

Σε αυτό το σημείο θα πρέπει να υπενθυμίσουμε ότι το βιβλίο αυτό δημιουργήθηκε με χρήση του P.L.C. **Twido** της **Telemecanique** και επομένως τα παραδείγματα είναι άμεσα συνδεδεμένα μαζί του.

Πάνω από όλα όμως το βιβλίο αυτό αποτελεί μια προσπάθεια να οδηγήσει τον αναγνώστη, χρήστη και προγραμματιστή, με ξεκάθαρα βήματα στην εκμάθηση, μέσα από μια όσο το δυνατόν κατανοητή και σε βάθος προσέγγιση και μεθοδολογία, στον τρόπο σκέψης και χρήσης ενός σύγχρονου συστήματος P.L.C..

Φυσικά δεν πρέπει να ξεχνάμε ότι το βιβλίο αυτό δημιουργήθηκε κατόπιν απαιτήσεως μεγάλης μερίδας αυτοματιστών, ηλεκτρολόγων και εγκαταστατών της βιομηχανίας, στις ανάγκες των οποίων εστιάστηκε η ανάλυση και εξήγηση των λύσεων που προτείνονται σε αυτό το βιβλίο.

Φυσικά, οι λύσεις αυτές σε καμία περίπτωση δεν πρέπει να θεωρούνται μονόδρομος, άλλα και όπως σε όλα τα άλλα πράγματα στον αυτοματισμό, μια πιθανή λύση σε ένα επίσης πιθανό σενάριο.

Επίσης μεγαλύτερη μέριμνα δόθηκε στον τρόπο χρήσης και στην επεξήγηση με τον απλούστερο κατα το δυνατόν τρόπο, των εργαλείων αυτών που θα μας βοηθήσουν στην δημιουργία και την ανάπτυξη μιας ολοκληρωμένης εφαρμογής παρά στην στείρα αναπαραγωγή και παράθεση έτοιμων λύσεων προς αντιγραφή.

Και αυτό γιατί κάτι τέτοιο δεν εξυπηρετεί ούτε την μάθηση, άλλα σίγουρα ούτε και την εφαρμογή αυτομάτων συστημάτων, μια και για την δημιουργία των εφαρμογών, δύο είναι τα απαραίτητα συστατικά από αρχής χρόνου: Γνώση και φαντασία.

Για αυτόν ακριβώς τον λόγο, αν και αυτό αναφέρεται και παρακάτω, οι όποιες λύσεις και βοήθειες προσφέρονται σε αυτό το βιβλίο, μπορούν να δικαιολογηθούν αναγνωρίζοντας ότι ουσιαστικά προχωρήσαμε στην δημιουργία ενός βιβλίου – πιλότου ίσως όχι πρώτου, αλλά σίγουρα μοναδικού στα δεδομένα της Ελληνικής βιβλιογραφίας, με σκοπό όχι μόνο να ξεφύγει από τα στενά πλαίσια της εκπαίδευσης, αλλά με βαθύτερη φιλοδοξία να σταθεί κοντά στις ανάγκες της βιομηχανίας με την μορφή ενός καλού βιβλίου - συνεργάτη.

Σε αυτό το σημείο θα πρέπει να δώσουμε **ιδιαίτερη προσοχή**, στην ανάγκη ανάγνωσης **κάθε φορά** των οδηγιών χρήσης του εκάστοτε ελεγκτή ή συναφών εξαρτημάτων του, όπως κάρτες για παράδειγμα, για την εξασφάλιση μιας ασφαλούς εγκατάστασης και λειτουργίας.

Έτσι οποιαδήποτε διαγράμματα ή συνδεσμολογίες περιγράφονται σε αυτό το βιβλίο, σκοπό έχουν να διευκολύνουν τον αναγνώστη, αλλά **σε καμία περίπτωση** να αντικαταστήσουν τις οδηγίες χρήσης και εγκατάστασης που συνοδεύουν τον ελεγκτή, και χορηγούνται από την κατασκευαστρια εταιρεία Schneider Electric.

*Για οποιαδήποτε άλλη απορία σχετικά με τις συνδέσεις ή την εφαρμογή σας, μπορείτε να επικοινωνήσετε με την Συνετάρ. Ηλεκτρικ ΑΕ, και την γραμμή Εξυπηρέτησης Πελατών, στο 8001162900 (χωρίς χρέωση από όλη την Ελλάδα).*

Τελειώνοντας θα πρέπει να πούμε ότι αυτή η εργασία και το αντικείμενο της επιστήμης που αυτή μελετά, ίσως μετά από μερικά χρόνια να ξεπεραστεί από την ίδια την επιστήμη, αλλά ελπίζω σε αυτό το μεταξύ χρονικό διάστημα να προσφέρει έστω λίγη βοήθεια και έργο σε όσους θελήσουν να μάθουν τα βασικά γύρω από την τεχνολογία των προγραμματιζόμενων λογικών ελεγκτών (P.L.C.) και την γλώσσα τους, χρησιμοποιώντας τα λόγια του Ναπο στο θρυλικό έργο του Ιουλιού Βέρν, “Είκοσι χιλιάδες λέυγες υπό την θάλασσα”:

«...Έχετε διεξάγει τις έρευνές σας όσο η επίγεια επιστήμη σας έχει επιτρέψει. Μα δεν ξέρετε όλα όσα μπορεί να μάθει κανείς, ούτε και έχετε δει ακόμη τα πάντα. Αφήστε με να σας διαβεβαιώσω τότε, ότι δεν θα μετανιώσετε για τον χρόνο που ξοδέψατε πάνω στο σκάφος μου. Πρόκειται να επισκεφθείτε ένα υπέροχο κόσμο θαυμάτων...»

Θεσσαλονίκη, Απρίλιος 2008

**Χρήστος Παπαζαχαρίας**

## 2.1. Γενική δομή ενός συστήματος αυτοματισμού

### 2.1.1 Σύστημα αυτοματισμού

Σύστημα αυτοματισμού εννοούμε ένα σύστημα το οποίο, **βασισμένο σε πληροφορίες** το οποίο παίρνει, εκτελεί **προκαθορισμένες ενέργειες** σε σχέση με το περιβάλλον του.



Εικόνα 2.1 : Σύστημα αυτοματισμού

Οι **ενέργειες** εκτελούνται σύμφωνα με μια **ακριβή διαδικασία**, η οποία εξαρτάται **αποκλειστικά** από τις πληροφορίες που δόθηκαν και υπολογίστηκαν, ή από προκαθορισμένες παραμέτρους.

<p>Παραδείγματα μερών (εξαρτημάτων) ενός συστήματος αυτοματισμού</p>		
	<p>Εγκατάσταση</p>	<p>Δεξαμενή</p>
	<p>Αισθητήρια</p>	<p>Αισθητήριο Υπερήχων</p>
	<p>Κινητήρια στοιχεία</p>	<p>Αντλία</p>
	<p>Έλεγχος ισχύος</p>	<p>Θερμομαγνητικός Διακόπτης</p>
	<p>Μονάδα επεξεργασίας δεδομένων</p>	<p>PLC</p>
	<p>Στοιχεία διαλόγου ανθρώπου-μηχανής, οθόνη, χειριστήρια</p>	<p>Οθόνη απεικόνισης</p>

● **Το σύστημα**, μπορεί να είναι να είναι πολύπλοκο όπως μια γραμμή συναρμολόγησης, ή παραγωγής σε ένα εργοστάσιο. Είναι επίσης δυνατόν να περιέχει έναν συνδυασμό από απλούστερες συσκευές, όπως μια γκαραζόπορτα, μια πισίνα, ή ένα σύστημα άρδευσης.

● **Τα αισθητήρια**, τα οποία όπως η όραση του οδηγού στο παράδειγμα του αυτοκινήτου, παρέχουν τις απαραίτητες **πληροφορίες από το περιβάλλον** για την εφαρμογή.

Τέτοια αισθητήρια μπορεί να είναι:

- Αισθητήρια στάθμης, ορίου
- Αισθητήρια θερμοκρασίας
- Αισθητήρια κίνησης

Για παράδειγμα, για να ανιχνεύσουμε ένα αυτοκίνητο στην είσοδο ενός parking, χρησιμοποιείται ένα φωτοηλεκτρικό αισθητήριο (φωτοκύτταρο)

● **Τα κινητήρια στοιχεία**, επιτρέπουν ενέργειες να εκτελεστούν στο σύστημα αυτοματισμού. Μιλάμε επομένως για κινητήρες, όπως αντλίες, αναδευτήρες κ.α. αλλά και κυλίνδρους γενικότερα.

● **Ο έλεγχος ισχύος**, όπως ρελέ, αυτόματοι διακόπτες, κ.λ.π., παρέχει την απαραίτητη ενέργεια για τα κινητήρια στοιχεία π.χ. αντλίες, σε συνδυασμό με τον επεξεργαστή, και τα λοιπά στοιχεία ελέγχου ισχύος.

● **Η μονάδα επεξεργασίας δεδομένων**, είναι ο εγκέφαλος της εγκατάστασης (ή μηχανήματος). Κάποτε δημιουργούνταν χρησιμοποιώντας ένα δίκτυο από βοηθητικά ρελέ και βοηθητικές επαφές τους, σήμερα αποτελείται από Προγραμματιζόμενους Ελεγκτές P.L.C. με ολοένα αυξανόμενες δυνατότητες σε υπολογισμούς και αποφάσεις, και ολοένα μικρότερες διαστάσεις!

● **Τα Στοιχεία διαλόγου ανθρώπου-μηχανής**, είναι απαραίτητα για την επίβλεψη ή έλεγχο κάθε αυτόματου συστήματος από τον άνθρωπο.

Τέτοια στοιχεία μπορεί να είναι:

- Μπουτόν, χειριστήρια
- Πληκτρολόγια
- Οθόνες απεικόνισης

## Τι πρέπει να θυμόμαστε:

Σύστημα αυτοματισμού εννοούμε ένα σύστημα το οποίο, **βασισμένο σε πληροφορίες** το οποίο παίρνει, εκτελεί **προκαθορισμένες ενέργειες** σε σχέση με το περιβάλλον του.

Αποτελείται από τα ακόλουθα στοιχεία:

- ▣ Το Σύστημα αυτοματισμού (Εγκατάσταση ή μηχανήμα)
- ▣ Τα αισθητήρια
- ▣ Τον έλεγχο ισχύος
- ▣ Τα κινητήρια στοιχεία
- ▣ Την μονάδα επεξεργασίας δεδομένων
- ▣ Τα στοιχεία διαλόγου ανθρώπου-μηχανής

### 2.1.2 Διάφορες Τεχνολογίες

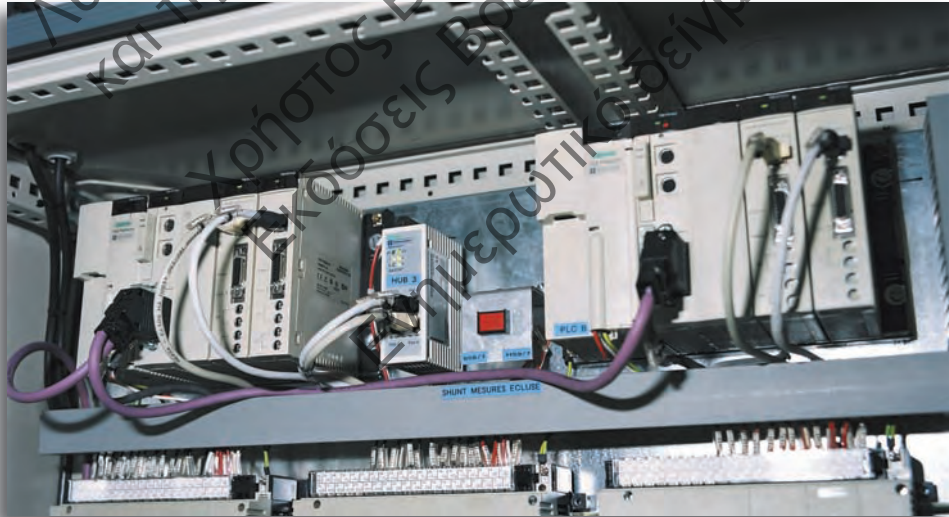
Μέχρι σήμερα μπορούμε να διακρίνουμε δυο τεχνολογικές επιλογές, όσον αφορά τον αυτοματισμό.

Αυτήν της **καλωδιωμένης τεχνολογίας** ή του **κλασσικού αυτοματισμού**, όπως συνήθως αποκαλείται,



Εικόνα 2.2 : Σύστημα με κλασσικό αυτοματισμό

και αυτήν της **προγραμματιζόμενης τεχνολογίας** ή **P.L.C.**, όπως επίσης συνήθως αποκαλείται.



Εικόνα 2.3 : Σύστημα με τεχνολογία P.L.C.

Μετά από μια σύντομη ματιά και στις δυο θα συγκεντρώσουμε τα κύρια στοιχεία τους, με τέτοιο τρόπο ώστε να μας βοηθήσει στην επιλογή μεταξύ των δυο διαφορετικών προσεγγίσεων ανά περίπτωση.



### Χαρακτηριστικά της καλωδιωμένης τεχνολογίας ή κλασσικού αυτοματισμού

Σε αυτή την περίπτωση, ο τρόπος λειτουργίας της εγκατάστασης καθορίζεται **αποκλειστικά και μόνο από την καλωδίωση** μεταξύ των διαφορετικών στοιχείων και εξαρτημάτων, όπως ρελέ, χρονικά, μετρητές, κ.α.

Το μεγάλο πλεονέκτημα που προσφέρει η καλωδιωμένη τεχνολογία είναι αναμφίβολα το **χαμηλό επίπεδο τεχνογνωσίας** σε επίπεδο κατασκευής και κυρίως συντήρησης της εγκατάστασης, καθώς δεν απαιτούνται γνώσεις προγραμματισμού ή ειδικά εργαλεία και λογισμικά (software).



Εικόνα 2.4 : Σύστημα με κλασσικό αυτοματισμό

Φυσικά, όσο πιο πολύπλοκη μια εγκατάσταση, τόσο πιο πολύπλοκη και εκτεταμένη και η καλωδίωση. Αυτό φυσικά συνεπάγεται **αυξημένη δυσκολία και πολυπλοκότητα** της μελέτης, αποσφαλμάτωσης και θέσης σε λειτουργία (start-up) της εγκατάστασης.

Για αυτόν ακριβώς το λόγο και οποιαδήποτε τροποποίηση της λειτουργίας απαιτηθεί στην συνέχεια, απαιτεί νέα μελέτη και πολλές φορές δυσανάλογα **μεγάλη και δύσκολη τροποποίηση της καλωδίωσης**.

Αυτό είναι απαραίτητο πολλές φορές, εστώ και για μια μικρή αλλαγή στην λειτουργία της εγκατάστασης, και αυτό συμβαίνει συνήθως λόγω του πλήθους των διαφορετικών στοιχείων και εξαρτημάτων που είναι καλωδιωμένα.

Επιπλέον όσο πιο πολύπλοκη εγκατάσταση, άρα και εκτεταμένη καλωδίωση, τόσο και πιο **μεγάλες οι απαιτήσεις σε χώρο και εξαερισμό**, ενώ ταυτόχρονα απαιτείται και η ανάλογη τεχνογνωσία σε επίπεδο κατασκευής κατάλληλων πινάκων και πεδίων ικανών να φιλοξενήσουν ένα τόσο μεγάλο αριθμό και όγκο καλωδίωσης, διαφορετικών στοιχείων και εξαρτημάτων.

Τέλος μια τέτοια τροποποίηση μπορεί να επιφέρει και ένα **επιπλέον κόστος στην εγκατάσταση**, καθώς μπορεί να προκύψουν καινούργια ή επιπλέον στοιχεία και εξαρτήματα που πρέπει να τοποθετηθούν εκ νέου, ή να αντικαταστήσουν κάποια από τα υπάρχοντα, με ανάλογο φυσικά κόστος.



### ► Την Μονάδα Τροφοδοσίας ή Τροφοδοτικό,

Το τροφοδοτικό είναι απαραίτητο για την παροχή τροφοδοσίας στα μέρη του P.L.C.. Είναι ενσωματωμένο στο P.L.C., και ανάλογα με το μοντέλο μπορεί να είναι συνήθως **220 Vac**, π.χ **Twido Compact**, ή **24 Vdc**, π.χ **Twido Modular**

### ► Τον Επεξεργαστή ή CPU,

Ο επεξεργαστής αποτελεί τον εγκέφαλο του P.L.C.

Αυτός είναι που:

διαβάζει τις τιμές των εισόδων, π.χ. αισθητήρια, εκτελεί το πρόγραμμα με τα δεδομένα που περιέχει, και διαμορφώνει την κατάσταση των εξόδων, π.χ. κινητήρες.

### ► Την Μνήμη,

Η μνήμη περιέχει το πρόγραμμα το οποίο πρέπει να εκτελεστεί, όπως επίσης και τα δεδομένα που χρησιμοποιούνται και διαμορφώνονται από το πρόγραμμα, π.χ. τιμές χρονικών, μετρητών, αριθμητικών πράξεων κ.λ.π.

Εκεί επομένως αποθηκεύονται όλα τα ζωτικής σημασίας για την λειτουργία του P.L.C. δεδομένα.

### ► Την Μονάδα Επικοινωνίας,

Η μονάδα επικοινωνίας είναι αυτή με την βοήθεια της οποίας επικοινωνεί ο ελεγκτής, προκειμένου να μεταφερθεί σε αυτόν το πρόγραμμα που πρόκειται να εκτελέσει π.χ. από ένα H/Y, και ανάλογα με την περίπτωση, δεδομένα π.χ. τιμές χρονικών, μετρητών, αριθμητικών πράξεων κ.λ.π. σε ένα δίκτυο P.L.C..

Στην περίπτωση του **Twido**, η μονάδα επικοινωνίας είναι ενσωματωμένη, και το μόνο ορατό σημείο της, είναι η ενσωματωμένη θύρα προγραμματισμού Mini-Din.

### ► Τις Μονάδες Εισόδων και Εξόδων

Οι Μονάδες Εισόδων ή απλά Είσοδοι, μεταφέρουν τις τιμές των εισόδων, π.χ. αισθητήρια, στο P.L.C., ενώ οι Μονάδες Εξόδων ή απλά Έξοδοι, μεταφέρουν την κατάσταση των εξόδων, π.χ. κινητήρες.

### Τι πρέπει να θυμόμαστε:

Ένα P.L.C. αποτελείται από τα ακόλουθα μέρη:

- ▣ Τον δίαυλο επικοινωνίας ή Bus
- ▣ Την Μονάδα Τροφοδοσίας ή Τροφοδοτικό
- ▣ Τον Επεξεργαστή ή CPU
- ▣ Την Μνήμη
- ▣ Την Μονάδα Επικοινωνίας
- ▣ Τις Μονάδες Εισόδων και Εξόδων

## 2.1.4 Βασικές έννοιες

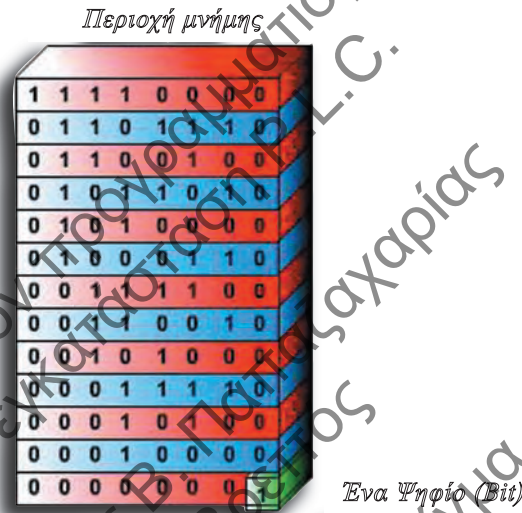
### Ψηφίο ή Bit

Μια από τις πιο βασικές έννοιες, ίσως και η πιο **θεμελιώδης**, πάνω στην οποία στηρίζεται όλο το οικοδόμημα του αυτοματισμού, είναι η έννοια του ψηφίου ή πιο γνωστού σε όλους μας ως **Bit**.

Ένα **Bit** μπορεί να έχει μόνο δυο καταστάσεις:

**1** , π.χ. ON ενός διακόπτη και

**0** , π.χ. OFF ενός διακόπτη



Εικόνα 2.7 : Ένα Ψηφίο(Bit) στην περιοχή μνήμης

Σε αυτό το σημείο πρέπει να αναφέρουμε ότι στα P.L.C., συναντούμε **τέσσερα είδη Bit**:

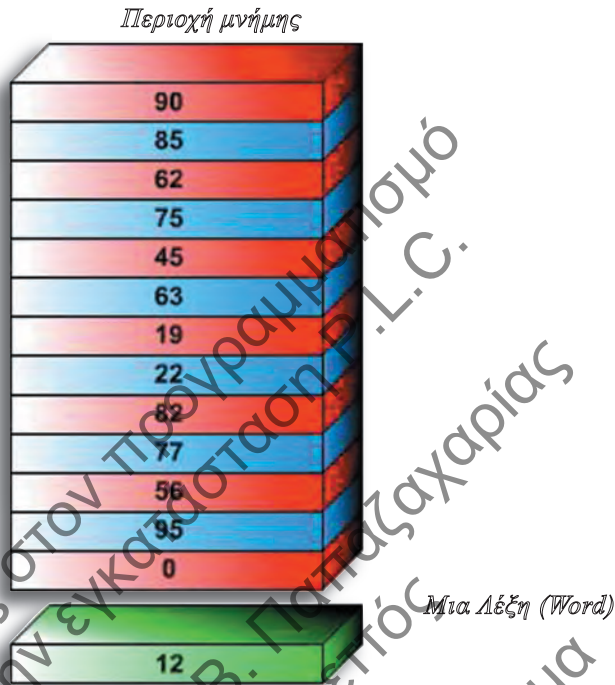
- Το Bit εισόδου, ή Input Bit (**%I**), π.χ. ένας διακόπτης
- Το Bit εξόδου, ή Output Bit (**%Q**), π.χ. ένας κινητήρας
- Το Bit μνήμης, ή Memory Bit (**%M**), π.χ. ένα βοηθητικό ρελέ
- Το Bit συστήματος, ή System Bit (**%S**), π.χ. ένα alarm

Ο συμβολισμός στην παρένθεση είναι αυτός που χρησιμοποιείται για τον συμβολισμό του κάθε **Bit**, όσον αφορά τον προγραμματισμό τους.

Πρέπει να τονίσουμε ότι τον συμβολισμό αυτό ακολουθεί όχι μόνο το **TwidoSuite** με το οποίο θα ασχοληθούμε στα επόμενα κεφάλαια, **άλλα όλα τα Software της Telemecanique**, όπως και όλα τα Software που υπακούουν στο πρότυπο **IEC 1131**.

## Λέξη ή Word

Μια ακόμη βασική έννοια, πάνω στην οποία στηρίζονται όλες οι εφαρμογές, ακόμη και οι πιο απλές που περιέχουν μετρήσεις και επομένως αριθμητικές τιμές, είναι η έννοια της λέξης ή πιο γνωστής σε όλους μας ως **Word**.



Εικόνα 2.8: Μία Λέξη (Word) στην περιοχή μνήμης

Σε αυτό το σημείο πρέπει να αναφορευθούμε ότι στα P.L.C., όπως και στην περίπτωση των **Bit**, συναντούμε τέσσερα είδη **Word**:

- Το **Word** εισόδου, ή **Input Word** ( **%IW** ), π.χ. μια θερμοκρασία
- Το **Word** εξόδου, ή **Output Word** ( **%QW** ), π.χ. μια τιμή 0-10V
- Το **Word** μνήμης, ή **Memory Word** ( **%MW** ), π.χ. μία τιμή ενός χρονικού
- Το **Word** συστήματος, ή **System Word** ( **%SW** ), π.χ. ώρα συστήματος

Ο συμβολισμός στην παρένθεση είναι αυτός που χρησιμοποιείται για τον συμβολισμό του κάθε **Word**, όσον αφορά τον προγραμματισμό τους, και ισχύουν όσα αναφέραμε και στην προηγούμενη παράγραφο που ασχολούνταν με τα **Bit**.

Όπως μπορεί πολύ εύκολα να καταλάβει κανείς, μια λέξη περιέχει ένα νούμερο, το οποίο για παράδειγμα θα μπορούσε να είναι η τιμή ενός μετρητή, ενός χρονικού, κ.λ.π.

Με άλλα λόγια μια λέξη δεν είναι τίποτα άλλο από μια διεύθυνση μνήμης στην οποία περιέχεται η τιμή αυτή.

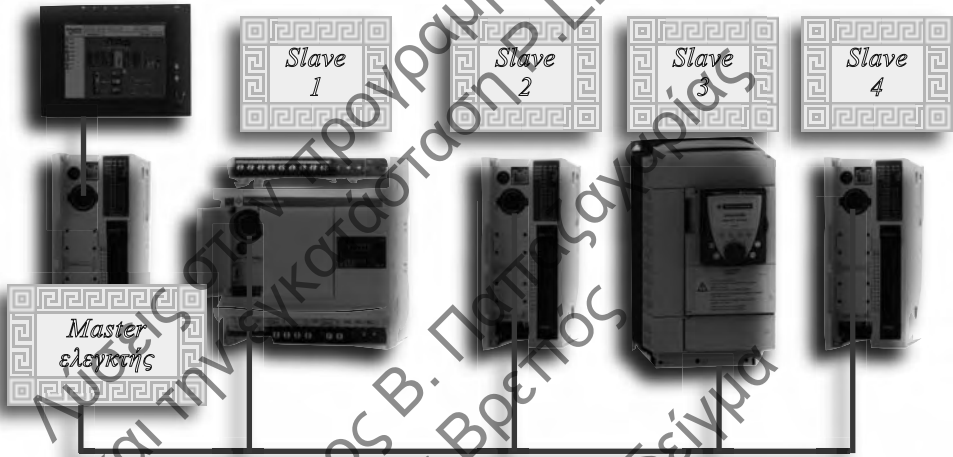
### 3.6.5 Δίκτυο Modbus

Η σύνδεση δικτύου Modbus είναι μια τοπολογία δικτύου μορφής διαύλου(bus) υψηλής ταχύτητας, αρχιτεκτονικής master/slave.

Σκοπός της είναι η μεταφορά δεδομένων μεταξύ του κύριου ελεγκτή (master) και μέχρι 31 απομακρυσμένων (slave) συσκευών, όπως ελεγκτές, οθόνες απεικόνισης, Η/Υ, ρυθμιστές στροφών (inverters), αναλυτές ενέργειας κλπ..

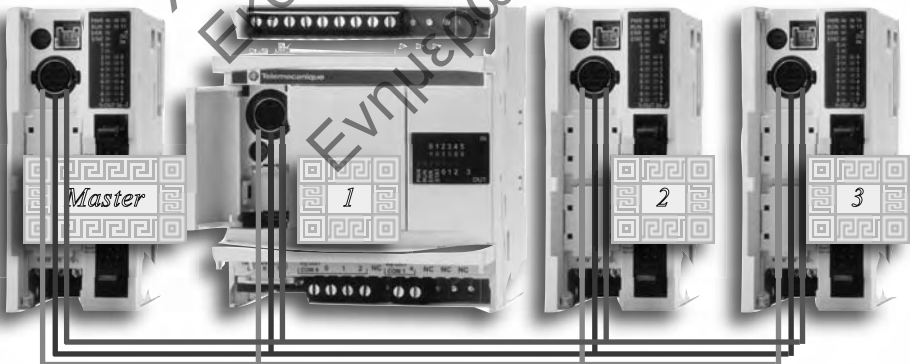
Με αυτό τον τρόπο δεδομένα της κάθε απομακρυσμένης συσκευής μπορούν να μεταφερθούν σε ένα μέγιστο μήκος δικτύου 1000 m, με μέγιστη ταχύτητα 19,2 Kbit/sec, ανάλογα με τη ζήτηση και το πρόγραμμα του master ελεγκτή.

Όπως προαναφέραμε, σε κάθε ελεγκτή μπορούν να χρησιμοποιηθούν δυο θύρες επικοινωνίας για σύνδεση μέχρι και σε δυο δίκτυα Modbus, όπου ο ελεγκτής μπορεί να λειτουργήσει ως Slave στην μια θύρα, και ταυτόχρονα ως master στην άλλη θύρα, όπως στην εφαρμογή της εικόνας 3.29.



Εικόνα 3.29 : Εφαρμογή με δίκτυο Modbus

Σε αυτό το παράδειγμα συνδέουμε τρεις ίδιους ελεγκτές, όπου κάθε ένας έχει μια κάρτα επέκτασης HMI, και δεύτερη θύρα RS485 με κλέμμες, όπως στην εικόνα 3.30.



Εικόνα 3.30 : Συνδέσεις εφαρμογής Modbus

### 3.6.6 Δίκτυο CANOpen

Το δίκτυο CANOpen είναι επίσης μια τοπολογία δικτύου μορφής διαύλου(bus) υψηλής ταχύτητας, αρχιτεκτονικής master/slave.

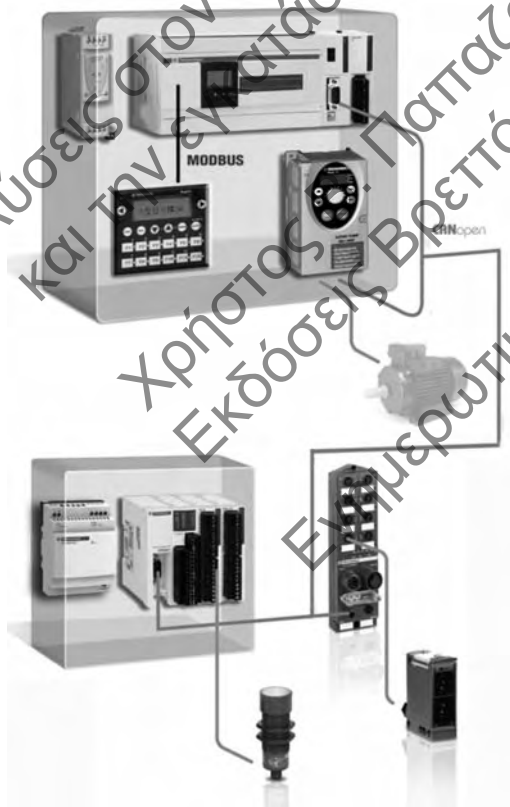


Σκοπός της είναι η μεταφορά δεδομένων μεταξύ του κύριου ελεγκτή (master) και μέχρι των 16 απομακρυσμένων (slave) συσκευών, όπως ελεγκτές, ρυθμιστές στροφών (inverters), ταχογεννητριών (encoders) κλπ., για απλές ή σύνθετες μηχανές, μεταφορά και διαχείριση υλικών, κ.α..

Με αυτό τον τρόπο δεδομένα της κάθε απομακρυσμένης συσκευής μπορούν να μεταφερθούν σε ένα μέγιστο μήκος δικτύου 1000 m, με μέγιστη ταχύτητα 1Mbit/sec, ανάλογα με τη ζήτηση και το πρόγραμμα του master ελεγκτή.

Έτσι με την μορφή μιας κάρτας επέκτασης που μπορεί να χρησιμοποιηθεί σε κάθε ελεγκτή που επιδέχεται επέκταση, με κωδικό TWDNCO1M για σύνδεση και διαχείριση, σαν master του CANOpen δικτύου αν το επιθυμούμε, ή απλά σαν slave, όπως στην εφαρμογή της εικόνας 3.32.

Εικόνα 3.31: Κάρτα CANOpen



Η κάρτα επέκτασης με κωδικό TWDNCO1M της Schneider Electric ακολουθεί τις προδιαγραφές του CANOpen Standard. Έτσι έχουμε την δυνατότητα να την ρυθμίσουμε με το λογισμικό TwidoSuite.

Επιπλέον μπορούμε να εισάγουμε αρχεία ρυθμίσεων άλλων στοιχείων και συσκευών CANOpen με μορφή EDS, μέσα στον κατάλογο TwidoSuite, για να τα χρησιμοποιήσουμε στο δίκτυο της εφαρμογής μας.

Ακόμη, ειδικά για τους ρυθμιστές στροφών (inverters) της σειράς Altivar της Schneider Electric, με το TwidoSuite παραδίδονται ειδικές εντολές (macros) για την απλούστερη εκτέλεση των εντολών κινήσεων τους μέσω δικτύου CANOpen, όπως Περιστροφή δεξιά, Περιστροφή αριστερά, και στάση.

Εικόνα 3.32 : Εφαρμογή με δίκτυο CANOpen



### 3.6.7 Δίκτυο Ethernet

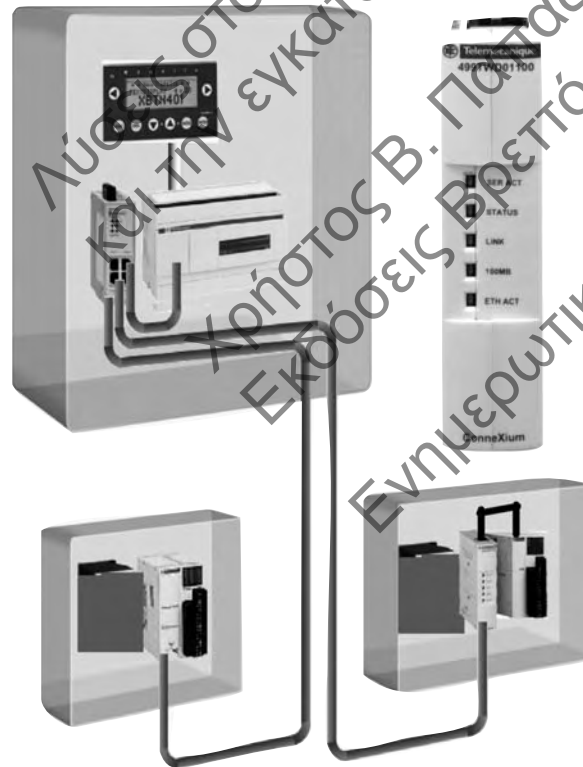
Η σύνδεση δικτύου Ethernet είναι μια τοπολογία δικτύου μορφής αστέρα υψηλής ταχύτητας, αρχιτεκτονικής client/server.



Σκοπός της είναι η μεταφορά δεδομένων μεταξύ του κύριου ελεγκτή (server) και μέχρι όσων απομακρυσμένων συσκευών (clients) επιτρέπουν οι IP διευθύνσεις του δικτύου, όπως ελεγκτές, οθόνες απεικόνισης, Η/Υ, inverters, Gateways κλπ..

Με αυτό τον τρόπο δεδομένα του κάθε απομακρυσμένου ελεγκτή μπορούν να μεταφερθούν, σε ένα μέγιστο μήκος δικτύου 100 m μέχρι το switch ή hub, με μέγιστη ταχύτητα 100 Mbit/sec, ανάλογα με τη ζήτηση και το πρόγραμμα του Client ελεγκτή.

Εικόνα 3.33 : Ελεγκτής TWDLcxE40DRF



Έτσι, για να επικοινωνήσει μέσω δικτύου Ethernet, όπως ο ελεγκτής TWDLcxE40DRF ο οποίος διαθέτει ενσωματωμένη θύρα επικοινωνίας RJ45 Ethernet, το ίδιο μπορεί να γίνει και με τους υπόλοιπους ελεγκτές της σειράς Twido με λειτουργικό τουλάχιστον V3.0, με την χρήση του μετατροπέα (Gateway) TwidoPort από Modbus σε Ethernet της σειράς ConneXium, με κωδ. 499TWD01100.

Η μόνη διαφορά έτσι που έχει ο ελεγκτής που χρησιμοποιεί το TwidoPort, σε σχέση με τον ελεγκτή TWDLcxE40DRF ο οποίος διαθέτει ενσωματωμένη θύρα επικοινωνίας RJ45 Ethernet, είναι ότι **δεν μπορεί** δώσει εντολή ανάγνωσης ή γραφής αιτημάτων (requests) από το Twido μέσω του TwidoPort.

Εικόνα 3.34 : Εφαρμογή με Ethernet

### Τι πρέπει να θυμόμαστε:

Όταν μιλάμε για επικοινωνία αποκλειστικά μεταξύ ελεγκτών Twido, μιλάμε για σύνδεση των ελεγκτών σε **Απομακρυσμένη σύνδεση**.

Σε αυτή την περίπτωση, μπορούμε να έχουμε δυο είδη δικτύωσης:

- ▣ Απομακρυσμένες I/O
- ▣ Απομακρυσμένοι Ελεγκτές

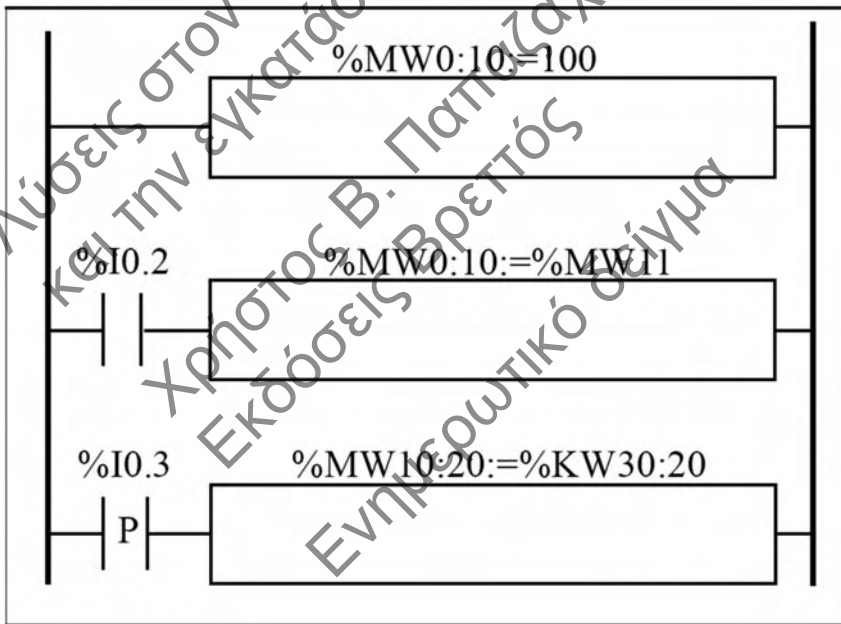
Αντίστοιχα, όταν μιλάμε για επικοινωνία μεταξύ ελεγκτών Twido αλλά και άλλων συσκευών, μιλάμε για σύνδεση των ελεγκτών σε **δίκτυο**, το οποίο μπορεί να είναι:

- ▣ ASCII
- ▣ Modbus
- ▣ CanOpen
- ▣ Ethernet
- ▣ AS-i

## ■ Ανάθεση Πινάκων λέξεων ή αλλιώς Word Tables

Η εντολή της ανάθεσης μπορεί φυσικά να εκτελεσθεί και στους ακόλουθους πίνακες λέξεων, όπως φαίνεται και από την εικόνα με τα παραδείγματα 4.55, παρόμοια με τους πίνακες ψηφίων:

- Άμεση τιμή => Πίνακα λέξεων (Γραμμή 1) ή Πίνακα διπλών λέξεων
- Λέξη => Πίνακα λέξεων (Γραμμή 2)
- Πίνακας λέξεων => Πίνακα λέξεων (Γραμμή 3)
- Διπλή λέξη => Πίνακα διπλών λέξεων
- Πίνακας διπλών λέξεων => Πίνακα διπλών λέξεων
- Άμεση δεκαδική τιμή => Πίνακα δεκαδικών λέξεων
- Δεκαδική λέξη => Πίνακα δεκαδικών λέξεων
- Πίνακα δεκαδικών λέξεων => Πίνακα δεκαδικών λέξεων



Εικόνα 4.55 : Παραδείγματα εντολών ανάθεσης πινάκων λέξεων

Ιδιαίτερη προσοχή θα πρέπει να δοθεί ώστε σε περιπτώσεις που έχουμε ανάθεση τιμής από πίνακα λέξεων σε πίνακα λέξεων όπως το παράδειγμα της γραμμής 3 και όχι μόνο βέβαια, το μήκος του πίνακα (L) πρέπει να είναι το ίδιο και για τους δύο πίνακες. Ειδικότερα, η σύνταξη όπως και οι τύποι των τελεστών για τις εντολές ανάθεσης πινάκων λέξεων είναι:

Τελεστής	Σύνταξη και επεξήγηση	Τύπος	Τύπος τελεστού 1 (Op1)	Τύπος τελεστού 2 (Op2)
:=	<p><b>Σύνταξη:</b></p> <p>[Op1: = Op2 ]</p> <p><b>Επεξήγηση :</b></p>	Πίνακας Λέξεων (Word Table)	%MWi:L, %SWi:L	Άμεση τιμή, %MWi:L, %SWi:L, %MWi, %KW, %IW, %QW, %IWA, %QWA, %SWi, %BLK.x
	<p>Ο τελεστής 1 (Op1) λαμβάνει την τιμή του τελεστού 2 (Op2)</p>	Πίνακας Διπλών λέξεων (Double Word Table)	%MDi:L	Άμεση τιμή, %MDi, %KDi, %MDi:L, %KDi:L
		Πίνακας Δεκαδικών λέξεων (Floating point Table)	%MFi:L]	Άμεση δεκαδική τιμή, %MFi, %KFi, %MFi:L, %KFi:L

## 4.8.2. Εντολή σύγκρισης

Η εντολή σύγκρισης αποτελεί μετά από αυτήν της ανάθεσης, την επόμενη συνηθέστερη μορφή αριθμητικής εντολής. Η χρήση τους αποσκοπεί στην **σύγκριση της τιμής που έχει ο τελεστής Op1 σε σχέση με τον τελεστή Op2.**

Για παράδειγμα αν ο τελεστής 1 είναι η αναλογική τιμή μιας θερμοκρασίας από ένα αισθητήριο PT100 που έχουμε συνδέσει σε μια αναλογική είσοδο του ελεγκτή, όπως λόγου χάριν η %IW3.0, μπορούμε να την συγκρίνουμε με ένα όριο που έχουμε θέσει στην τιμή αυτή, όποια και να είναι αυτή, σε μια λέξη μνήμης, λόγου χάριν την %MW32.

Η σύνταξη για τις εντολές σύγκρισης είναι:

### [Op1 Τελεστής Op2]

Έτσι για το παραπάνω παράδειγμα μας θα μπορούσαμε να είχαμε π.χ, %IW3.0>%MW32. Ο ακόλουθος πίνακας απαριθμεί τους τύπους εντολών σύγκρισης, και τους ανάλογους τελεστές.

Τελεστής	Λειτουργία
>	Σύγκριση εάν ο τελεστής 1 είναι <b>μεγαλύτερος</b> από τον τελεστή 2
>=	Σύγκριση εάν ο τελεστής 1 είναι <b>μεγαλύτερος ή ίσος</b> από τον τελεστή 2
<	Σύγκριση εάν ο τελεστής 1 είναι <b>μικρότερος</b> από τον τελεστή 2
<=	Σύγκριση εάν ο τελεστής 1 είναι <b>μικρότερος ή ίσος</b> από τον τελεστή 2
=	Σύγκριση εάν ο τελεστής 1 είναι <b>ίσος</b> με τον τελεστή 2
<>	Σύγκριση εάν ο τελεστής 1 είναι <b>διάφορος</b> από τον τελεστή 2

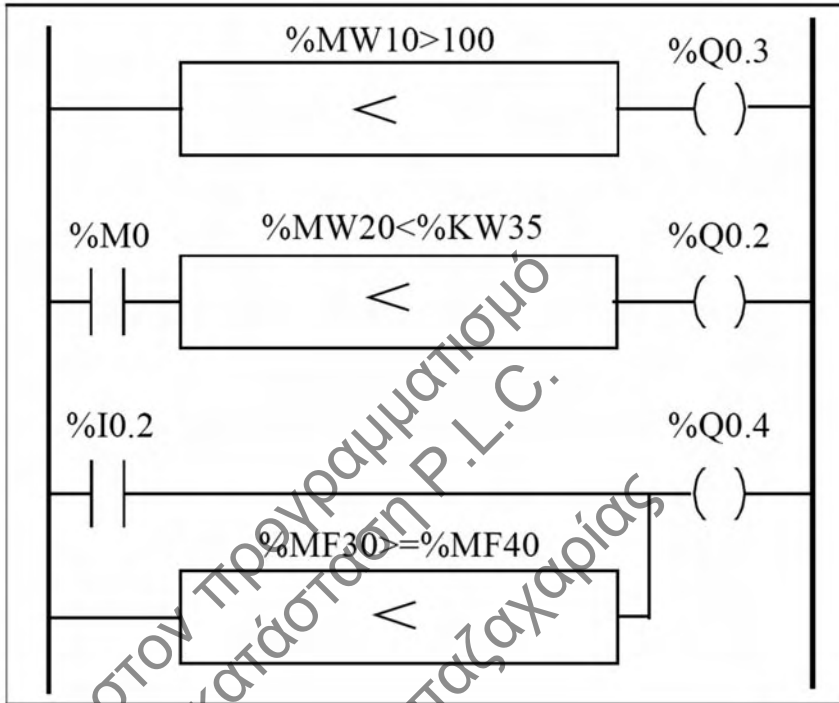
Όπως αναφέραμε και στην εισαγωγή του κεφαλαίου, η σύνταξη των αριθμητικών εντολών σύγκρισης ακολουθεί κατά κανόνα την εισαγωγή τους σε αγκύλες.

Ταυτόχρονα με την χρήση της εντολής σύγκρισης γίνεται εφικτή η εκτέλεση δράσεων και εντολών, όπως λόγου χάριν η ενεργοποίηση ενός πηνίου ή μια εντολή ανάθεσης, κάτω από συνθήκες ή έστω υπό μια συνθήκη που ορίζει ο χρήστης.

Έτσι, εάν το αποτέλεσμα της προηγούμενης εντολής σύγκρισης ήταν αληθές (Λογική Boole = 1), τότε η αριθμητική εντολή εκτελείται.

Αντίστοιχα, εάν το αποτέλεσμα της προηγούμενης εντολής σύγκρισης ήταν ψευδές (Λογική Boole = 0), τότε η αριθμητική εντολή δεν εκτελείται και ο τελεστής παραμένει αμετάβλητος.

Για να το κατανοήσουμε καλύτερα ας δούμε το παράδειγμα της εικόνας 4.56.



Εικόνα 4.56 : Παραδείγματα εντολών σύγκρισης

Οι τύποι των τελεστών ειδικότερα για τις εντολές σύγκρισης είναι:

Τύπος	Τύπος τελεστών 1 (Op1)	Τύπος τελεστών 2 (Op2)
Λέξη (Word)	%MWi, %KW <sub>i</sub> , %INW <sub>i</sub> , %IW <sub>i</sub> , %IWA <sub>i</sub> , %QNW <sub>i</sub> , %QW <sub>i</sub> , %QWA <sub>i</sub> , %QNW <sub>i</sub> , %SW <sub>i</sub> , %BLK.x	Άμεση τιμή, %MW <sub>i</sub> , %KW <sub>i</sub> , %INW <sub>i</sub> , %IW <sub>i</sub> , %IWA <sub>i</sub> , %QNW <sub>i</sub> , %QW <sub>i</sub> , %QWA <sub>i</sub> , %SW <sub>i</sub> , %BLK.x, %MW <sub>i</sub> [%MW <sub>i</sub> ], %KW <sub>i</sub> [%MW <sub>i</sub> ]
Διπλή λέξη (Double Word)	%MDi, %KDi	Άμεση τιμή, %MDi, %KDi, %MDi[%MW <sub>i</sub> ], %KD[%MW <sub>i</sub> ]
Δεκαδική λέξη (Floating Word)	%MFi, %KFi	Άμεση δεκαδική τιμή, %MFi, %KFi, %MFi [%MW <sub>i</sub> ], %KFi [%MW <sub>i</sub> ]



## 5.1 Γενικά για την δημιουργία προγράμματος

Για τους περισσότερους απο εμάς που ασχολούμαστε με τον αυτοματισμό, η ίδια η ιδέα του λογικού ελεγκτή είναι σχεδόν απόλυτα συνυφασμένη με τον ίδιο τον προγραμματισμό του. Έτσι όπως είναι φυσικό η συγγραφή ενός βιβλίου με θέμα τον αυτοματισμό και ειδικότερα τους λογικούς ελεγκτές, δεν θα μπορούσε παρα να αφιερώσει ένα κεφάλαιο ειδικά στην διαδικασία δημιουργίας του προγράμματος του λογικού ελεγκτή, ώστε να παρέχει την μεγαλύτερη ευελιξία και ευκολία εκμάθησης.

Σε αυτό το κεφάλαιο θα δούμε πως περατώνεται μια τέτοια διαδικασία, θα ορίσουμε τα στάδια της, και θα τα αναλύσουμε με τέτοιο τρόπο, ώστε να γίνει απόλυτα κατανοητή και ταυτόχρονα απόλυτα προσιτή η διαδικασία δημιουργίας ενός προγράμματος.

Είναι γεγονός βέβαια, ότι ο αποτελεσματικότερος τρόπος εκμάθησης αλλά και εμπέδωσης ενός αντικειμένου, πόσο μάλλον μιας διαδικασίας, είναι αυτός που εμπεριέχει την πράξη της θεωρίας στην πράξη.

Για τον λόγο αυτό ταυτόχρονα με την ανάλυση της διαδικασίας, θα δούμε ταυτόχρονα και ένα παράδειγμα εφαρμογής, το οποίο θα επιλύσουμε κατά την διάρκεια του κεφαλαίου, ώστε να ρίξουμε όσο το δυνατόν πιο άπλετο φώς στο κάθε στάδιο, αλλά και στην ολοκλήρωση του προγραμματισμού ενός λογικού ελεγκτή.



Εικόνα 5.1: Εφαρμογή ελέγχου μονάδας σιλό

## 5.2 Διάρθρωση προγράμματος

Όπως είδαμε μέχρι στιγμής λέγοντας πρόγραμμα εννοούμε μια **σειρά εντολών** οι οποίες υποδεικνύουν στον επεξεργαστή του ελεγκτή, τις δράσεις που πρέπει να αναλάβει και να εκτελέσει.

Έτσι, ακόμη και αν δεν δώσουμε τις εντολές με μια λογική σειρά, ή δεν ακολουθούμε πάντα μια συγκεκριμένη διαδικασία προγραμματισμού, αν αυτές οι εντολές είναι οι κατάλληλες, το πρόγραμμα θα επιτύχει τον σκοπό του.

Όμως, μια και η ίδια η διαδικασία του προγραμματισμού δεν αποτελεί παρά την επικοινωνία μεταξύ του προγραμματιστή (άνθρωπου) και του ελεγκτή (μηχανή), θα πρέπει όπως και σε όλες τις διαδικασίες επικοινωνίας, να ακολουθούμε μια κατά το δυνατόν συγκεκριμένη δομή προγραμματισμού.

Και αυτό, γιατί για τον άνθρωπο που συμμετέχει στην διαδικασία, για εμάς με άλλα λόγια, μια ορισμένη και επαναλήψιμη διαδικασία, μπορεί να μας βοηθήσει στην ενίσχυση της ικανότητας επίλυσης σύνθετων προβλημάτων, όπως αυτά εμφανίζονται σχεδόν πάντα στην βιομηχανία.

Με αυτό τον τρόπο θα πετύχουμε δραστική μείωση σε:

- Σφαλμάτα
- Επαναλήψεις
- Ελλείψεις και κενά
- Χρόνο προγραμματισμού
- Χρόνο αποσφαλμάτωσης
- Χρόνο αλλαγών και βελτιώσεων

Το διάγραμμα που ακολουθεί, απεικονίζει την διαδικασία διάρθρωσης και δημιουργίας ενός προγράμματος. Σε αυτό το διάγραμμα βλέπουμε αναλυτικά τα στάδια που θα ακολουθήσουμε, και το οποίο όπως θα δούμε αναμφίβολα μπορεί να μας βοηθήσει στα παραπάνω, και όχι μόνο.

### 5.3 Ορισμός εισόδων/εξόδων και συμβόλων

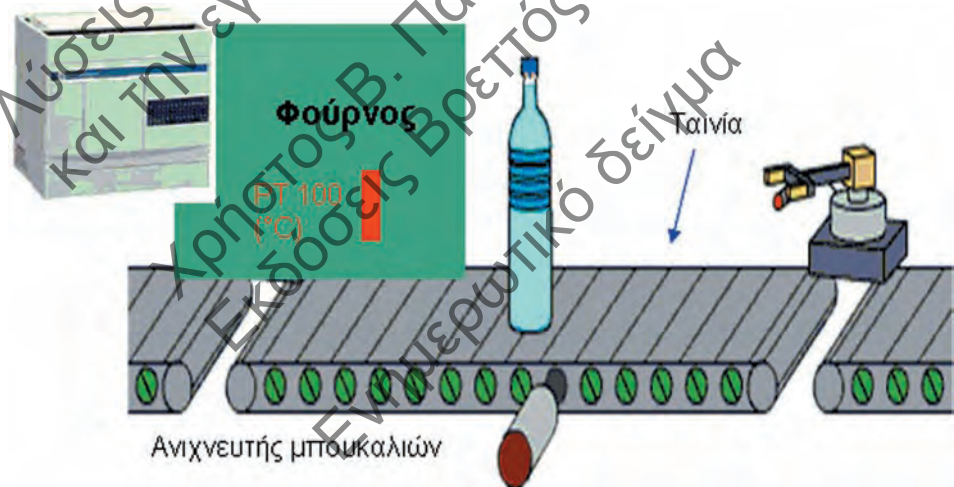
Το παράδειγμα που θα χρησιμοποιήσουμε για να κατανοήσουμε καλύτερα την διαδικασία δημιουργίας ενός προγράμματος, είναι μια τυπική εφαρμογή, παραλλαγές της οποίας μπορεί να συναντήσουμε σε πλήθος βιομηχανιών.

Σε αυτή την εφαρμογή, ο ελεγκτής μας Twido, δίνει εντολή run/stop σε έναν ιμάντα μεταφοράς φιαλών, ενώ τατόχρονα μετρά τον αριθμό των μπουκαλιών.

Όταν ο αριθμός φθάσει στον επιθυμητό αριθμό που έθεσε ο χειριστής, ο ιμάντας σταματά και ενεργοποιείται ένας βραχίονας παλετοποίησης. Ο βραχίονας παλετοποίησης είναι εφοδιασμένος με έναν τερματικό διακόπτη ορίου. Μόλις ο τερματικός του βραχίονα παλετοποίησης δώσει σήμα, που σημαίνει ότι ο βραχίονας ολοκλήρωσε την κίνηση του, ο κύκλος ξεκινά από την αρχή.

Ο ελεγκτής μας Twido ελέγχει επίσης την θερμοκρασία του φούρνου. Αν η θερμοκρασία είναι υψηλότερη από το όριο που θέτει ο χειριστής, δημιουργείται σήμα alarm. Φυσικά η εντολή Run και Stop δίνεται από δυο απλά μπουτόν.

Επίσης ο ιμάντας κινείται από κινητήρα με προστασία θερμικού, το οποίο δίνει με την σειρά του σε περίπτωση υπερθέρμανσης του κινητήρα σήμα alarm στο Twido. Στην εικόνα 5.2, βλέπουμε μια απεικόνιση της εφαρμογής μας.



Εικόνα 5.2: Παράδειγμα μεταφοράς και παλετοποίησης μπουκαλιών

Όπως αναφέραμε και στην προηγούμενη ενότητα, η διαδικασία προγραμματισμού ενός ελεγκτή, μπορεί να διαχωριστεί σε επτά κύρια στάδια, απο τα οποία το πρώτο είναι αυτό του ορισμού των εισόδων / εξόδων και συμβόλων.

Ας τα πάρουμε λοιπόν ένα προς ένα:

Πρώτη κίνηση μας, είναι να καταγράψουμε σε ένα πίνακα τις εισόδους της εφαρμογής μας, ξεκινώντας από τις ψηφιακές εισόδους (DI) :

**%I0.0 : Run**

**%I0.1 : Stop**

**%I0.2 : Ανιχνευτής μπουκαλιών (φωτοκύτταρο)**

**%I0.3 : Τερματικός διακόπτης (βραχίονα παλετοποίησης)**

**%I0.4 : θερμικό κινητήρα (επαφή )**

Αντίστοιχα για τις ψηφιακές εξόδους (DO), έχουμε:

**%Q0.0 : Κινητήρας Ιμάντα**

**%Q0.1 : Βραχίονας παλετοποίησης**

**%Q0.2 : Alarm θερμοκρασίας φούρνου (ενδεικτική λυχνία)**

**%Q0.3 : Ένδειξη λειτουργίας ιμάντα (ενδεικτική λυχνία)**

**%Q0.4 : Alarm θερμικού κινητήρα (ενδεικτική λυχνία)**

Τέλος για τις αναλογικές εισόδους (AI), έχουμε:

**%IW0.0 : Θερμοκρασία φούρνου (αισθητήριο PT100)**

Όπως βλέπουμε στην εφαρμογή μας, δεν έχουμε αναλογικές εξόδους, αν είχαμε όμως θα τις προσθέταμε στην λίστα που δημιουργήσαμε μαζί με όσες άλλες επιπλέον εισόδους και εξόδους χρειαζόμαστε.

Όπως βλέπουμε με την δημιουργία του πίνακα εισόδων / εξόδων, έχουμε ολοκληρώσει το πρώτο πολύ σημαντικό στάδιο της διαδικασίας δημιουργίας του προγράμματος μας, το οποίο ήταν:

## 1. Ορισμός εισόδων/εξόδων και συμβόλων

### Τι πρέπει να θυμόμαστε:

Η διαδικασία προγραμματισμού ενός ελεγκτή, μπορεί να διαχωριστεί σε επτά κύρια στάδια, από τα οποία το πρώτο είναι αυτό του ορισμού των εισόδων/εξόδων και συμβόλων.

Ο ορισμός των εισόδων/εξόδων και συμβόλων πραγματοποιείται με την δημιουργία μιας λίστας των στοιχείων που απαιτεί η εφαρμογή μας, όπως τα ακόλουθα:

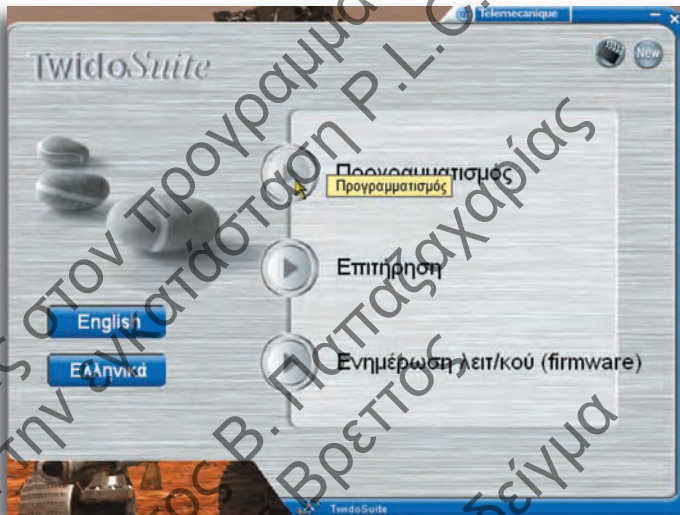
- ▶ Ψηφιακές εισόδους
- ▶ Ψηφιακές εξόδους
- ▶ Αναλογικές εισόδους
- ▶ Αναλογικές εξόδους
- ▶ Λειτουργικά στοιχεία (Χρονικά, Μετρητές, κλπ)

## 5.4 Επιλογή υλικού

Όπως αναφέραμε στην προηγούμενη ενότητα, για τις ανάγκες της εφαρμογής του παραδείγματος μας, έχουμε ήδη υπολογίσει ότι απαιτούνται τουλάχιστον:

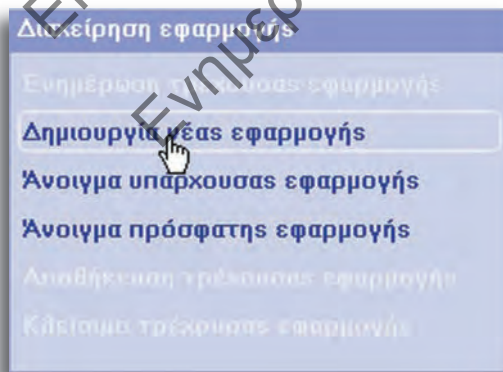
- 5 Ψηφιακές εισοδοι
- 5 Ψηφιακές έξοδοι
- 1 Αναλογική έξοδος

Σε αυτό το σημείο, ανοίγουμε το λογισμικό TwidoSuite, και επιλέγουμε **Προγραμματισμός**, όπως βλέπουμε στην εικόνα 5.3:



Εικόνα 5.3: Παράθυρο επιλογής Προγραμματισμού και γλώσσας

Στην συνέχεια μέσα στο λογισμικό TwidoSuite, επιλέγουμε **Δημιουργία νέας εφαρμογής**, όπως βλέπουμε στην εικόνα 5.4:



Εικόνα 5.4: Παράθυρο επιλογής διαχείρισης εφαρμογής

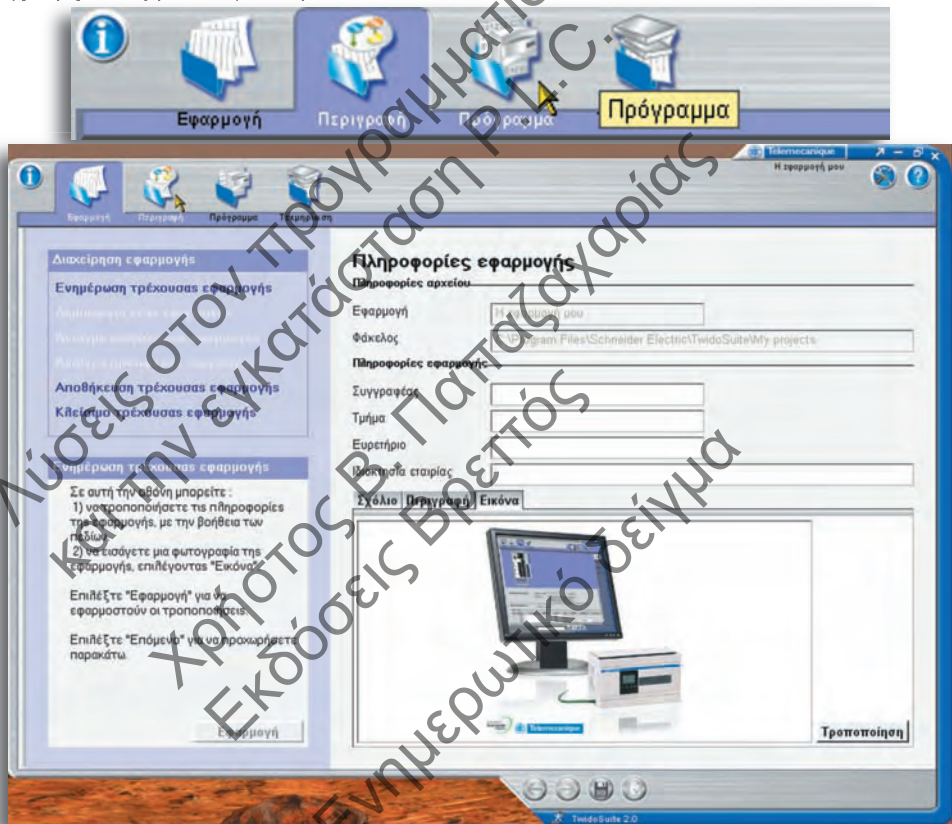


Ακολουθώντας, στις **Πληροφορίες εφαρμογής**, δίνουμε τα στοιχεία που θέλουμε να έχει η εφαρμογή μας, όπως ονομασία εφαρμογής, φάκελλο που θα αποθηκευθεί στον Η/Υ, όνομα συγγραφέα, τμήμα, κλπ.

Παράλληλα, μας δίνεται η δυνατότητα μέσα από τις καρτέλλες με τίτλους **Σχόλιο**, **Περιγραφή** και **Εικόνα**, να αποθηκεύσουμε κάποια σχόλια για την εφαρμογή μας, μια περιγραφή της, ακόμη και κάποια εικόνα της.

Όλα τα παραπάνω στοιχεία είναι φυσικά προαιρετικά, και σκοπεύουν στην δημιουργία μιας όσο το δυνατόν πιο ολοκληρωμένης τεκμηρίωσης, η οποία μπορεί να μας βοηθήσει και να μας κατατοπίσει στο μέλλον, να θυμηθούμε ή να προσέξουμε τις οποιεσδήποτε ιδιαιτερότητες περιέχει η εφαρμογή μας.

Στην συνέχεια, προχωρούμε στην **επιλογή του υλικού** επιλέγοντας την καρτέλα **Περιγραφή**, όπως βλέπουμε στην εικόνα 5.5:



Εικόνα 5.5: Παράθυρο επιλογής καρτέλλας Περιγραφή

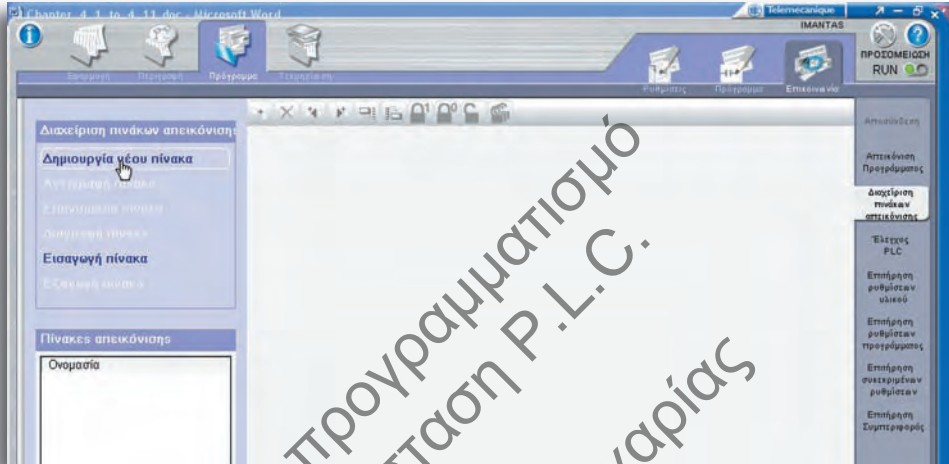
Στην συνέχεια εμφανίζεται το περιβάλλον σχεδίασης της εφαρμογής μας, μέσα απο την οποία επιλέγουμε το υλικό που θα χρησιμοποιήσουμε.

Η εύρεση του κατάλληλου υλικού είναι κυριολεκτικά παιχνίδι, ακόμη και αν δεν διαθέτουμε οποιοδήποτε έντυπο κατάλογο του υλικού ή οδηγό επιλογής, μέσα από το λογισμικό μας.

Στην καρτέλα **Περιγραφή** επιλέγουμε από μια λεπτομερέστατη αναδιπλούμενη λίστα μορφής δέντρου (tree), που βρίσκεται στα δεξιά, το κατάλληλο υλικό για την εφαρμογή μας.

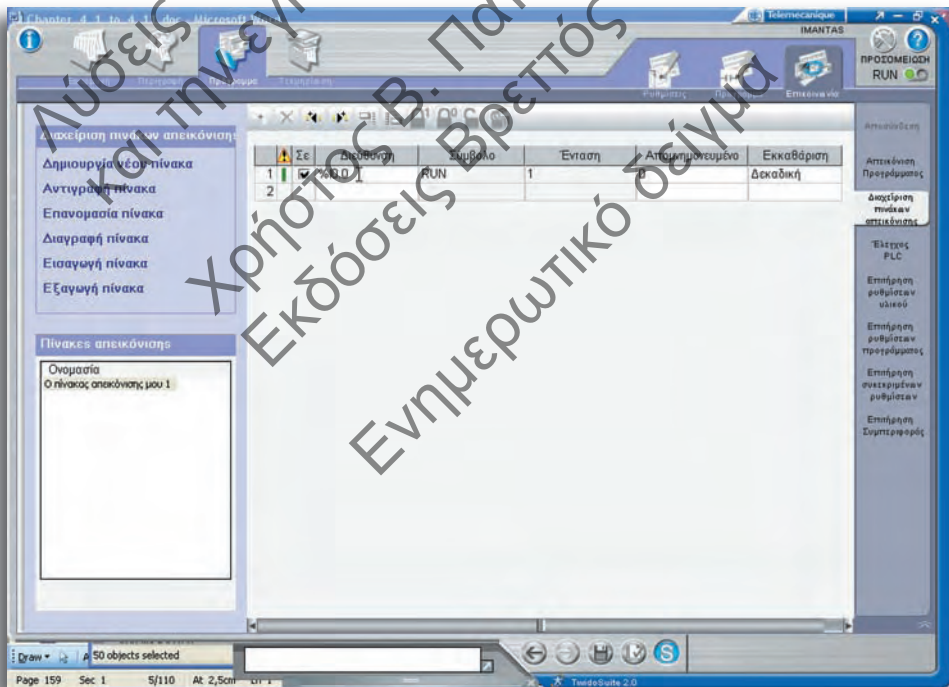
### 5.12.3 Πίνακας Απεικόνισης (Animation Table)

Αντίστοιχα, αν επιθυμούμε να δούμε και να ελέγξουμε συγκεκριμένες μεταβλητές και στοιχεία της εφαρμογής μας σε μορφή λίστας, επιλέγουμε Καρτέλλα Διαχείρισης Πινάκων Απεικόνισης, και **Δημιουργία νέου πίνακα**, όπως στην εικόνα 5.71:



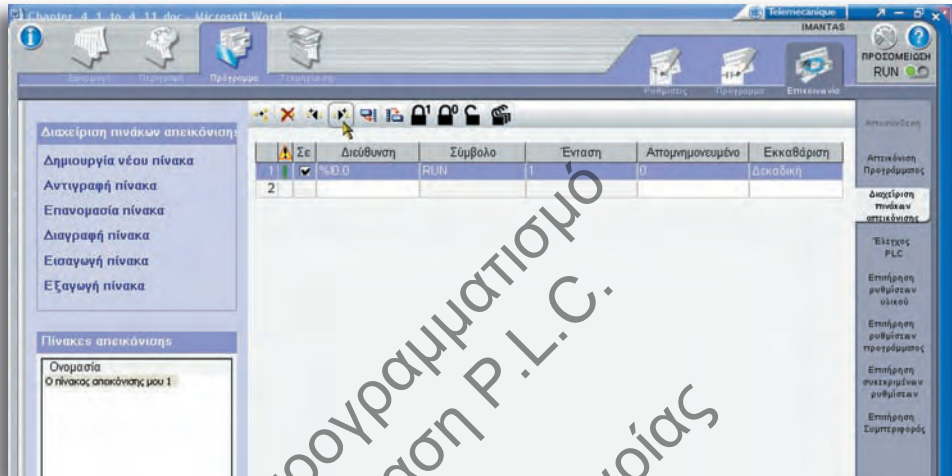
Εικόνα 5.71: Δημιουργία Πίνακα Απεικόνισης

Έπειτα, στο αντίστοιχο κελί προσθέτουμε το στοιχείο όπως μια είσοδο, ή την τιμή που μας ενδιαφέρει να παρατηρήσουμε ή να τροποποιήσουμε την τιμή του, όπως στην εικόνα 5.72.



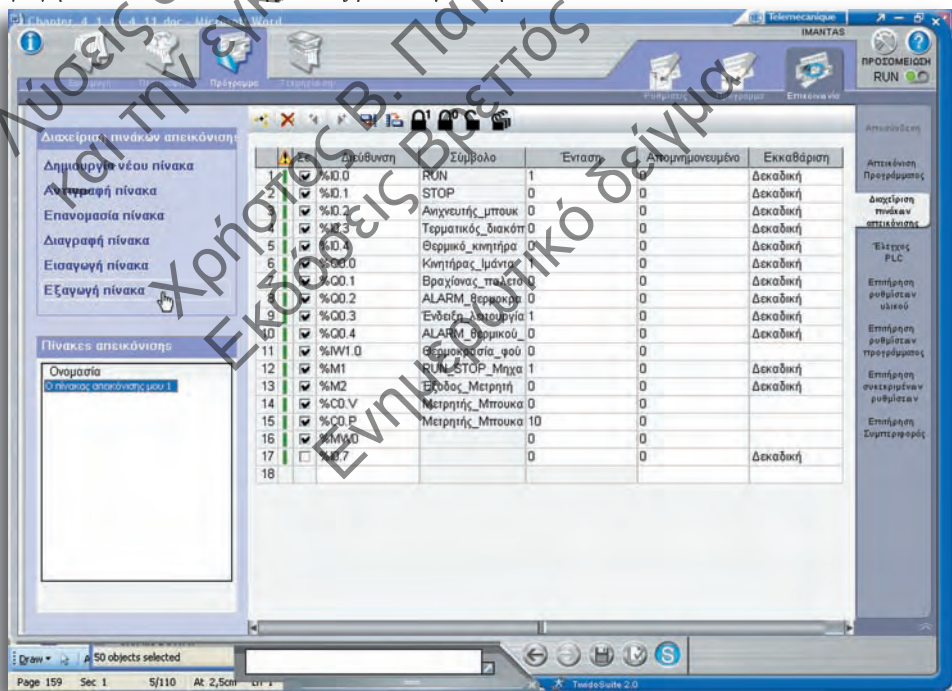
Εικόνα 5.72: Τροποποίηση Πίνακα Απεικόνισης

Στη συνέχεια θα μπορούσαμε πολύ απλά στο επόμενο κάτωθεν κελί να προσθέσουμε το επόμενο στοιχείο όπως π.χ. την επόμενη είσοδο, εδώ %Ι0.1, %Ι0.2, κ.ο.κ. ή ακόμη πιο εύκολα να κάνουμε κλικ στο **Πλήκτρο προσθήκης συγγενών στοιχείων**, και να το κάνει το πρόγραμμα για εμάς, όσες φορές και αν χρειαστεί, αυτόματα και ταχύτατα.



Εικόνα 5.73: Προσθήκη συγγενών στοιχείων

Αν βέβαια δημιουργήσουμε έναν πίνακα απεικόνισης, που μας ενδιαφέρει να χρησιμοποιήσουμε και στο μέλλον, κάτι που συμβαίνει αρκετά συχνά, απλά επιλέγουμε την Εξαγωγή πίνακα απεικόνισης, όπως βλέπουμε στην εικόνα 5.74.



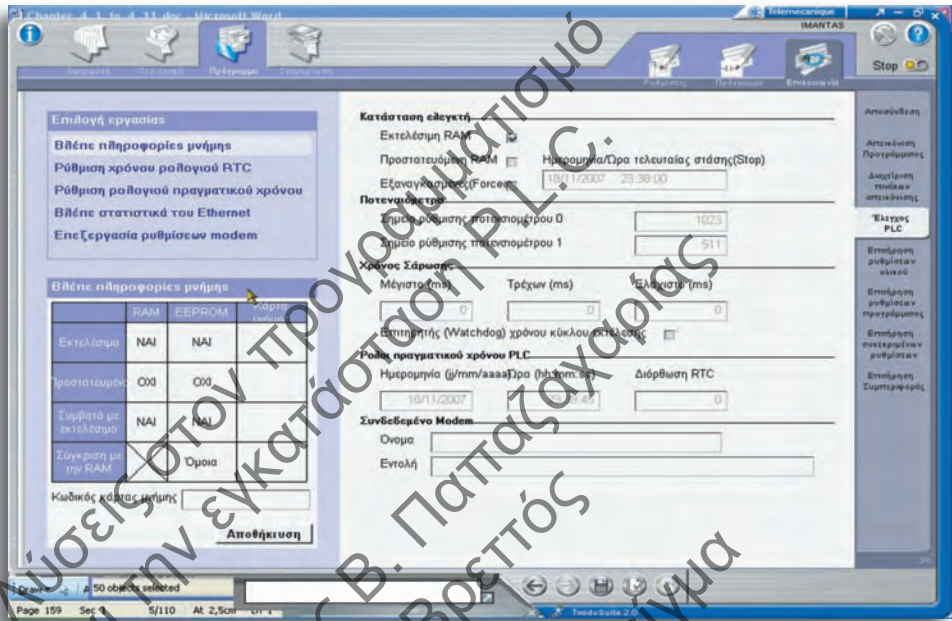
Εικόνα 5.74: Εξαγωγή πίνακα απεικόνισης



### 5.12.5 Διαγνωστικές λειτουργίες με το TwidoSuite

Όπως έχουμε παρατηρήσει, το λογισμικό TwidoSuite είναι δομημένο με τέτοιο τρόπο ώστε να μπορεί να μας παρέχει τις απαραίτητες σε μας πληροφορίες ακολουθώντας όσο περισσότερο γίνεται την πορεία της ίδιας της εφαρμογής μας.

Με αυτό το σκεπτικό, έχοντας συνδεθεί στον ελεγκτή μας δίνονται επιπλέον διαγνωστικές λειτουργίες και δυνατότητες ρυθμίσεων.



Εικόνα 5.78 : Πληροφορίες μνήμης ελεγκτή

Η πρώτη που θα αναφέρουμε είναι αυτή που φαίνεται στην εικόνα 5.78, και στην οποία μπορούμε να δούμε όλα τα στοιχεία σχετικά με την μνήμη του ελεγκτή μας, αρκεί να επιλέξουμε το πλήκτρο στα δεξιά μέρος της καρτέλας Έλεγχος PLC.

Εκεί μπορούμε να δούμε πληροφορίες σχετικά με το πρόγραμμα μας όπως:

- Αν έχει αποθηκευτεί εκτός από την RAM, και στην EEPROM (Backup).
- Αν το αποθηκευμένο πρόγραμμα στην RAM είναι το ίδιο με της EEPROM.
- Αν έχει προστατευθεί με κωδικό είτε στην RAM, είτε στην EEPROM.
- Όλα τα παραπάνω και για την προαιρετική κάρτα μνήμης

Ταυτόχρονα όμως στην ίδια καρτέλλα, μπορούμε να δούμε σε πραγματικό χρόνο (Live) στοιχεία όπως:

- Ρύθμιση χρόνου ρολογιού πραγματικού χρόνου RTC
- Στατιστικά της σύνδεσης σε δίκτυο Ethernet
- Επεξεργασία ρυθμίσεων Modem