



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής



Πτυχιακή Εργασία :

Σύστημα Πυρανίχνευσης σε κτίριο με ανάπτυξη
συστήματος ελέγχου και κατάλληλου λογισμικού

Χρήση PLC και ανάπτυξη SCADA με Visual Basic

- Επιβλέπων Καθηγητής : Βλησίδης Ανδρέας
- Σπουδαστές : Λουκάς Γεώργιος AM 2205
Δίπλας Παναγιώτης-Στυλιανός AM 2264

Ηράκλειο , Ιούνιος 2014

Ευχαριστίες

Σε αυτό το σημείο θα θέλαμε να ευχαριστήσουμε πρώτα από άλλους τους γονείς μας που μας παρείχαν την δυνατότητα να αρχίσουμε και να ολοκληρώσουμε τις σπουδές μας σε τόσο δύσκολους καιρούς, για την υποστήριξη την εμπιστοσύνη και την υπομονή που έδειξαν. Επίσης τους καθηγητές μας Α.Βλησίδη και Σ.Χαρακόπουλο για το κίνητρο που μας παρείχαν ώστε να ασχοληθούμε με το συγκεκριμένο αντικείμενο και να ολοκληρώσουμε την απαιτητική πτυχιακή ελπίζοντας να έχουμε την ευκαιρία να ασχοληθούμε πάνω σε αυτό τον τομέα.

Abstract

With Great effort, perseverance and patience we were able to complete this thesis which deals with automations and imaging with the help of Scada. What we did was to create, using open source software that all companies provide by payment. Therefore we have created automation for industries and generally large assemblies, where the fire risk is quite high. The hardware used was a PLC S7-300 which was already installed in ENPET and for creating the software Visual Studio and particular language Visual Basic. The demonstration of the thesis was to model depicting a two storey building and operation of the automation.

Σύνοψη

Με μεγάλη προσπάθεια, επιμονή και υπομονή καταφέραμε να ολοκληρώσουμε της παρούσα πτυχιακή η οποία ασχολείται με συστήματα αυτοματισμού και απεικόνιση τους με της βοήθεια των Scada. Αυτό που κάναμε ήταν να δημιουργήσουμε με ελεύθερο λογισμικό αυτό που όλες οι εταιρείες παρέχουν επι πληρωμή. Δημιουργήσαμε λοιπόν αυτοματισμό για βιομηχανίες και γενικότερα για μεγάλα συγκροτήματα για πυρανίχνευση και πυρόσβεση στα οποία ο κίνδυνος είναι αρκετά μεγάλος. Ως Hardware χρησιμοποιήσαμε PLC s7-300, την ήδη υπάρχουσα εγκατάσταση στο κτήριο ENPIET, και για την δημιουργία του λογισμικού το Visual Studio και ειδικότερα την γλώσσα Visual Basic. Η επίδειξη της πτυχιακής έγινε σε μακέτα απεικονίζοντας ένα διόροφο κτήριο και την λειτουργία του αυτοματισμού.



Πίνακας περιεχομένων

Ευχαριστίες.....	2
Abstract	3
Σύνοψη	4
ΚΕΦΑΛΑΙΟ 1	9
ΕΙΣΑΓΩΓΗ.....	9
1.1 Περίληψη.....	9
1.2 Δυσκολίες Εκπόνησης	9
1.3 Σκοπός	9
ΚΕΦΑΛΑΙΟ 2.....	10
ΠΡΟΓΡΑΜΜΑΤΙΖΟΜΕΝΟΙ ΛΟΓΙΚΟΙ ΕΛΕΓΚΤΕΣ (PLC) ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑ	10
2.1 Υλικά για τον έλεγχο μιας εγκατάστασης μέσω P.L.C.....	10
2.2 Πλεονεκτήματα PLC συγκριτικά με τον κλασικό αυτοματισμό	12
2.3 Βιομηχανική Επικοινωνία.....	14
2.3.1 Επίπεδα	14
2.3.2 PROFIBUS.....	15
2.3.3 Industrial Ethernet.....	16
2.3.4 PROFINET	20
ΚΕΦΑΛΑΙΟ 3	22
Η ΔΟΜΗ ΤΩΝ PLC	22
3.1 Βασική δομή των PLC	22
3.1.1 Πλαίσιο Στήριξης (RACK).....	24
3.1.2 Στήριξη Καρτών στο Rack:	24
3.2 Τροφοδοτικό PS (PowerSupply)	26
3.3 Κεντρική Μονάδα Επεξεργασίας - CPU	27
3.3.1 Εσωτερικά η CPU Περιέχει	28
3.3.2 Εξωτερικά μια CPU παρουσιάζει:.....	29
3.4 Ψηφιακές Μονάδες Εισόδων DI (Digital Input)	30
3.5 Ψηφιακές Μονάδες Εξόδων DO (Digital Output).....	31
3.6 Μονάδες Αναλογικών Εισόδων AI (Analog Input)	32
3.7 Μονάδες Αναλογικών Εξόδων AO (Analog Output)	33
ΚΕΦΑΛΑΙΟ 4	34
SIMATIC MANAGER	34

4.1 Προγραμματισμός με το Simatic Step7.....	34
4.2 Η Δομή του προγράμματος.....	36
4.3 Ορισμός και Διαρρύθμιση.....	37
4.3.1 Δημιουργία νέου Project.....	38
4.4 Διαμόρφωση του σταθμού-Hardware configuration	38
4.5 Βασικές πύλες και οργάνωση των blocks	45
4.6 Ανάπτυξη των OB1 , FC , DB	48
ΚΕΦΑΛΑΙΟ 5	53
ΣΥΣΤΗΜΑΤΑ SCADA.....	53
5.1 Τι είναι Scada.....	53
5.2 Πλεονεκτήματα Scada	54
5.3 Σύνομη Περιγραφή του υλοποιηθέντος συστήματος	55
ΚΕΦΑΛΑΙΟ 6	57
ΑΝΑΛΥΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΛΟΓΙΣΜΙΚΟΥ.....	57
6.1 Βάση Δεδομένων Mysql	57
6.2 Ο προγραμματισμός στην Visual basic.....	59
6.3 Κώδικας	63
ΚΕΦΑΛΑΙΟ 7	74
7.1 ΣΥΜΠΕΡΑΣΜΑΤΑ	74
7.2 Μελλοντική εργασία και Επεκτάσεις.	75

ΕΙΚΟΝΑ 1 : ΕΦΑΡΜΟΓΗ ΕΛΕΓΧΟΥ	10
ΕΙΚΟΝΑ 2 : ΠΩΛΗΣΕΙΣ PLC	13
ΕΙΚΟΝΑ 3 : ΕΠΙΠΕΔΑ ΑΥΤΟΜΑΤΟΠΟΙΗΜΕΝΗΣ ΔΙΑΔΙΚΑΣΙΑΣ	14
ΕΙΚΟΝΑ 4 : ΔΙΑΣΥΝΔΕΣΗ PROFIBUS.....	15
ΕΙΚΟΝΑ 5 : ΣΥΣΤΗΜΑ PROFIBUS	15
ΕΙΚΟΝΑ 6 : ΕΞΕΛΙΞΗ ΤΟΥ INDUSTRIAL ETHERNET	16
ΕΙΚΟΝΑ 7 : INDUSTRIAL ETHERNET/PROFINET	18
ΕΙΚΟΝΑ 8 : INDUSTRIAL ETHERNET PROTOCOLS	19
ΕΙΚΟΝΑ 9 : ΧΡΗΣΗ ΤΩΝ ΠΡΩΤΟΚΟΛΩΝ	19
ΕΙΚΟΝΑ 10 : ΧΡΟΝΟΣ ΑΠΟΚΡΗΣΗΣ PROFINET	20
ΕΙΚΟΝΑ 11 : PRONINET & PROFIBUS	21
ΕΙΚΟΝΑ 12 : ΔΟΜΗ ΤΩΝ PLC	22
ΕΙΚΟΝΑ 13 : ΣΤΟΙΧΕΙΑ ΤΟΥ PLC.....	23
ΕΙΚΟΝΑ 14 : RACK.....	24
ΕΙΚΟΝΑ 15 : ΚΑΘΕΤΗ ΚΑΙ ΟΡΙΖΟΝΤΙΑ ΕΓΚΑΤΑΣΤΑΣΗ	25
ΕΙΚΟΝΑ 16 : ΜΕΓΙΣΤΟΣ ΑΡΙΘΜΟΣ MODULE	25
ΕΙΚΟΝΑ 17 : ΤΡΟΦΟΔΟΤΙΚΟ	26
ΕΙΚΟΝΑ 18 : ΚΑΛΩΔΙΩΣΗ ΤΡΟΦΟΔΟΤΙΚΟΥ.....	27
ΕΙΚΟΝΑ 19 : CPU 31X	29
ΕΙΚΟΝΑ 20 : ΨΗΦΙΑΚΗ ΚΑΡΤΑ ΕΙΣΟΔΩΝ.....	30
ΕΙΚΟΝΑ 21 : ΨΗΦΙΑΚΗ ΚΑΡΤΑ ΕΞΟΔΩΝ	31
ΕΙΚΟΝΑ 22 : ΑΝΑΛΟΓΙΚΗ ΚΑΡΤΑ ΕΙΣΟΔΩΝ.....	32
ΕΙΚΟΝΑ 23 : CPU 314C-2DP	33
ΕΙΚΟΝΑ 24 : ΓΕΝΙΚΗ ΔΟΜΗ PROJECT	35
ΕΙΚΟΝΑ 25 : ΔΟΜΗ ΤΟΥ ΔΙΚΟΥ ΜΑΣ PROJECT	35
ΕΙΚΟΝΑ 26 : ΔΟΜΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ.....	36
ΕΙΚΟΝΑ 27 : BLOCK ΚΑΙ ΙΔΙΟΤΗΤΕΣ	36
ΕΙΚΟΝΑ 28 : PC/PG INTRFAVE	37
ΕΙΚΟΝΑ 29 : ΝΕΟ PROJECT.....	38
ΕΙΚΟΝΑ 30 : ΔΗΜΙΟΥΡΓΙΑ STATION.....	38
ΕΙΚΟΝΑ 31 : ΤΟΠΟΘΕΤΗΣΗ RAIL.....	39
ΕΙΚΟΝΑ 32 : ΔΙΑΧΕΙΡΙΣΗ HARDWARE	39
ΕΙΚΟΝΑ 33 : ΑΛΛΑΓΗ ΔΙΕΥΘΥΝΣΕΩΝ ΑΝΑΛΟΓΙΚΩΝ ΚΑΡΤΩΝ	40
ΕΙΚΟΝΑ 34 : ΟΡΙΣΜΟΣ ΤΡΟΠΟΥ ΜΕΤΡΗΣΗΣ.....	41
ΕΙΚΟΝΑ 35 : ΑΛΛΑΓΗ ΔΙΕΥΘΥΝΣΕΩΝ ΨΗΦΙΑΚΗΣ ΚΑΡΤΑΣ	42
ΕΙΚΟΝΑ 36 : ΑΝΑΘΕΣΗ IP ΣΤΟ CP	43
ΕΙΚΟΝΑ 37 : NETPRO TOOL	44
ΕΙΚΟΝΑ 38 : ΠΑΡΑΔΕΙΓΜΑΤΑ FBD/STL.....	45
ΕΙΚΟΝΑ 39 : BLOCK ΣΥΓΚΡΙΣΗΣ	46
ΕΙΚΟΝΑ 40 : ΤΥΠΟΙ ΜΕΤΑΒΛΗΤΩΝ	47
ΕΙΚΟΝΑ 41 : SCALING FUNCTION	48
ΕΙΚΟΝΑ 42 : COMPAIR FUNCTION	48
ΕΙΚΟΝΑ 43 : COMPARE FUNCTION.....	49
ΕΙΚΟΝΑ 44 : ΣΥΝΑΡΤΗΣΕΙΣ ΣΤΟ OB	49
ΕΙΚΟΝΑ 45 : SCALING FUNCTION.....	50
ΕΙΚΟΝΑ 46 : ΔΗΜΙΟΥΡΓΙΑ FC1.....	50
ΕΙΚΟΝΑ 47 : FC1 Η ΣΥΝΕΧΕΙΑ	51
ΕΙΚΟΝΑ 48 : ΔΗΜΙΟΥΡΓΙΑ FC2.....	51
ΕΙΚΟΝΑ 49 : FC2 Η ΣΥΝΕΧΕΙΑ	52

ΕΙΚΟΝΑ 50 : ΠΕΡΙΕΧΟΜΕΝΑ DATABLOCK.....	52
ΕΙΚΟΝΑ 51 : ΣΥΣΤΗΜΑ SCADA	53
ΕΙΚΟΝΑ 52 : ΟΛΟΚΛΗΡΩΜΕΝΟ SCADA	54
ΕΙΚΟΝΑ 53 : ΑΠΛΗ ΠΡΟΤΑΣΗ SCADA	55
ΕΙΚΟΝΑ 54 : Η ΟΛΟΚΛΗΡΩΜΕΝΗ ΠΡΟΤΑΣΗ ΜΑΣ ΓΙΑ SCADA	56
ΕΙΚΟΝΑ 55 : ΈΝΑΡΞΗ ΡΗΡΜΥADMIN	57
ΕΙΚΟΝΑ 56 : ΦΤΙΑΧΝΟΥΜΕ ΤΟ TABLE	57
ΕΙΚΟΝΑ 57 : Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΜΑΣ	58
ΕΙΚΟΝΑ 58 : ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ PROJECT	59
ΕΙΚΟΝΑ 59 : MDI ΦΟΡΜΑ/ΚΥΡΙΟ ΜΕΡΟΣ	59
ΕΙΚΟΝΑ 60 : SCADA ΤΑΒ.ΑΠΕΙΚΟΝΗΣΗ ΕΙΣΟΔΩΝ/ΕΞΟΔΩΝ/ΓΡΑΦΗΜΑΤΑ.....	60
ΕΙΚΟΝΑ 61 : ΓΡΑΦΙΚΗ ΑΠΕΙΚΟΝΙΣΗ ΤΟΥ SCADA.....	61
ΕΙΚΟΝΑ 62 : WEBCAM ΓΙΑ ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΧΩΡΟΥ	62

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Περίληψη

Ως κυρίως στόχος αυτής της πτυχιακής ήταν η δημιουργία μιας πλήρους αυτοματοποιημένης διεργασίας με ελεύθερο λογισμικό. Μελετήθηκαν υπάρχουσες αρχιτεκτονικές αυτοματισμών αλλά σε καμία περίπτωση δεν βρήκαμε κάτι παρόμοιο.

Πιο συγκεκριμένα χρησιμοποιήσαμε την Visual Basic και την ανοιχτού κώδικα βιβλιοθήκη libnodave η οποία χρησιμοποιείτε για την δημιουργία εφαρμογών windows για την επικοινωνία PLC με τον υπολογιστή μας. Στη συνέχεια δημιουργήσαμε ένα πλήρες σύστημα scada το οποίο θα εκτελεί τον έλεγχο για την Πυρανίχνευση και πυρόσβεση ενός κτηρίου ή μιας βιομηχανικής εγκατάστασης. Όλα αυτά τα δεδομένα αποθηκεύονται και ανακτούνται απο την βάση δεδομένων που δημιουργήσαμε για την απεικόνιση γραφημάτων και την εμφάνιση του ιστορικού. Θα μπορούμε να έχουμε επίσης εικόνα για το τί γίνεται στην εγκατάσταση μας μέσω του διαδικτύου απο την στιγμή που όλα τα δεδομένα θα είναι στην βάση δεδομένων.

Έτσι το σύστημα που δημιουργήσαμε είναι πλήρως λειτουργικό και απαραίτητο για κάθε μεγάλη εγκατάσταση αλλά και μικρή. Το επόμενο μελλοντικό βήμα θα είναι η δημιουργία μιας Android εφαρμογής για επίβλεψη της εγκατάστασης μέσω Tablet ή smartphone.

1.2 Δυσκολίες Εκπόνησης

Όπως σε κάθε πρόβλημα που απαιτεί λύση έτσι και στην παρούσα πτυχιακή δεν ήταν όλα εύκολα αλλά υπήρξαν πολλές δυσκολίες. Εκτός από το κομμάτι του αυτοματισμού ένα μεγάλο κομμάτι της πτυχιακής ήταν η υλοποίηση μιας εφαρμογής για την διαχείριση και επίβλεψη του αυτοματισμού (Scada) αλλά και η άντληση των δεδομένων και η παρουσίαση τους στο διαδίκτυο ώστε να υπάρχει πρόσβαση και ενημέρωση από παντού. Έπρεπε λόγω έλλειψης πείρας και από την στιγμή που δεν υπάρχει κάποια παρόμοια εγκατάσταση ούτε μοντέλο να αναφερθώ στον επιβλέπων καθηγητή για βοήθεια και να καταλήξω μετά από πολύ έρευνα στην τελική μορφή τα που θα ανταποκρίνεται στις απαιτήσεις της βιομηχανίας.

1.3 Σκοπός

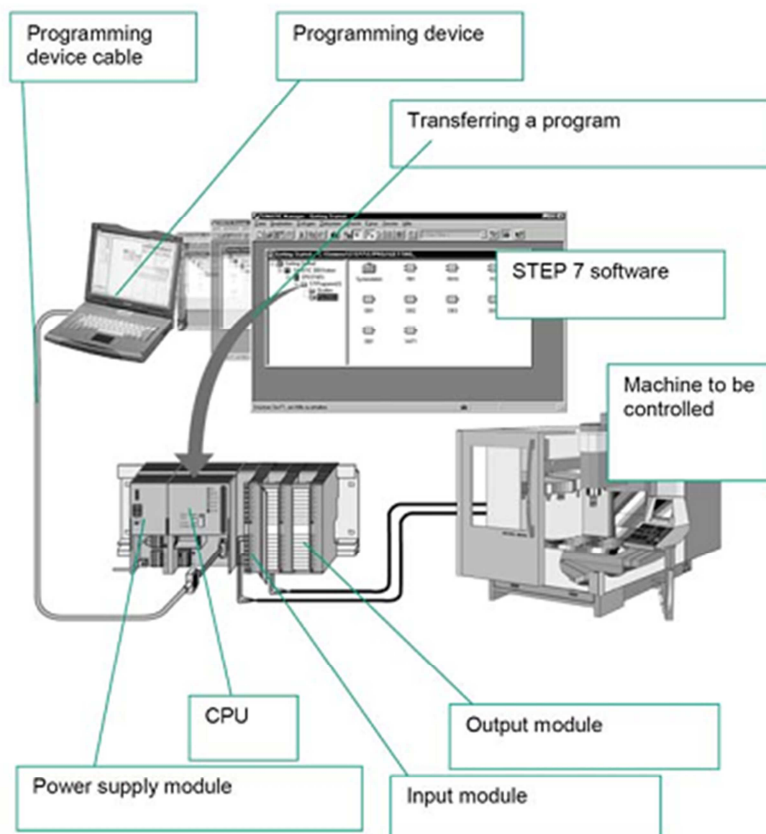
Σκοπός της πτυχιακής είναι να γίνει μια μελέτη και εγκατάσταση για την ασφάλεια μιας βιομηχανίας. Ωστόσο πέρα από τον σκοπό της πτυχιακής, σκοπός μου ως μηχανικός ήταν να εξοικειωθώ με το βιομηχανικό περιβάλλον, να αντλήσω όσο το δυνατόν περισσότερη γνώση και να μπορέσω να σχεδιάσω ένα σύστημα αυτοματισμού το οποίο θα αποσκοπεί σε μια πραγματική εγκατάσταση και θα είναι απαραίτητο για κάθε επιχείρηση με υψηλό ρίσκο κινδύνου χωρίς να επενδύει πολλά χρήματα.

ΚΕΦΑΛΑΙΟ 2

ΠΡΟΓΡΑΜΜΑΤΙΖΟΜΕΝΟΙ ΛΟΓΙΚΟΙ ΕΛΕΓΚΤΕΣ (PLC) ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑ

2.1 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. Αυτή αποτελείται:

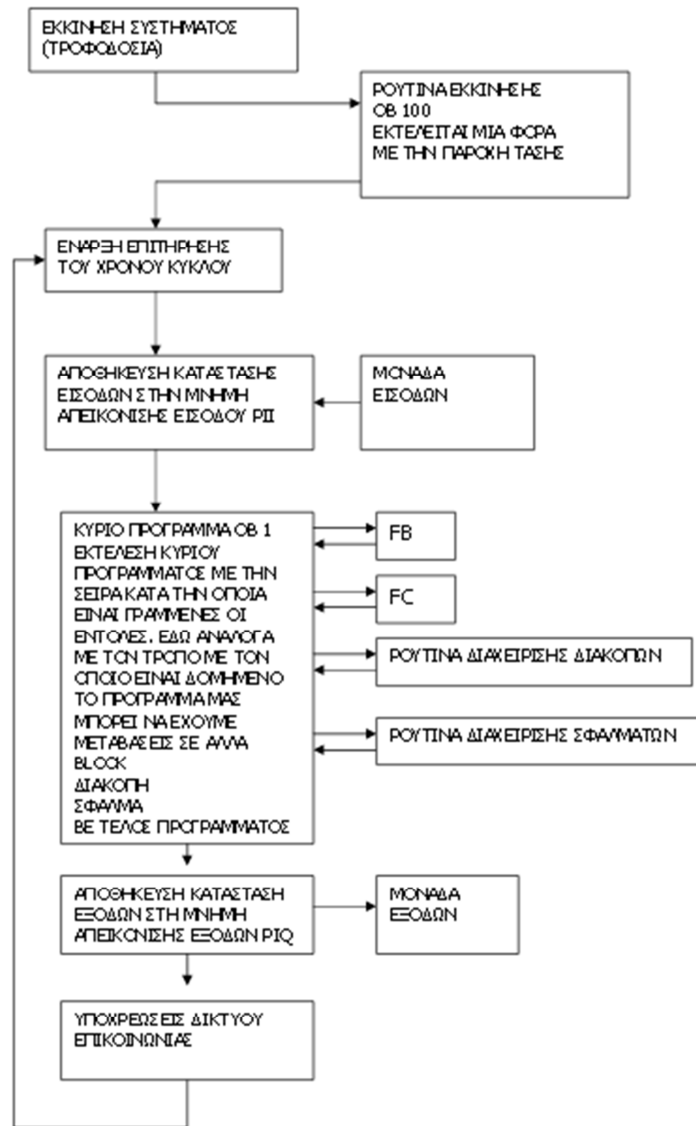


Εικόνα 1 : Εφαρμογή ελέγχου

- ΠΡΟΓΡΑΜΜΑΤΙΣΤΡΙΑ: Είναι το μέσο με το οποίο ο άνθρωπος επικοινωνεί με το PLC.
- ΠΑΚΕΤΟ SOFTWARE: Είναι το πρόγραμμα (γλώσσα) με το οποίο ο άνθρωπος επικοινωνεί με την προγραμματίστρια.
- ΤΡΟΦΟΔΟΤΙΚΟ: Ο ρόλος του είναι να δημιουργεί τις τάσεις που χρειάζεται το PLC για την τροφοδοσία του.
- CPU: Είναι ο εγκέφαλος του συστήματος εδώ περιέχονται και εκτελούνται τόσο το λειτουργικό πρόγραμμα του PLC όσο και το πρόγραμμα του χρήστη.
- ΚΑΡΤΕΣ ΕΙΣΟΔΟΥ: Είτε ψηφιακές, είτε αναλογικές, αυτές έχουν τον ρόλο να μετατρέπουν τα σήματα της εγκατάστασης σε σήματα τα οποία μπορεί να επεξεργαστεί η CPU.
- ΚΑΡΤΕΣ ΕΞΟΔΟΥ: Είτε ψηφιακές, είτε αναλογικές, αυτές έχουν τον ρόλο να μετατρέπουν τα σήματα που έχει ήδη επεξεργαστεί η CPU σε κατάλληλες τάσεις τις οποίες στέλνουμε προς την εγκατάσταση.

Σε εφαρμογές με χρήση των PLC η παρουσίαση καλωδίωσης περιορίζεται μόνο στα περιφερειακά εξαρτήματα (αισθητήρια, διακόπτες, λυχνίες, ...).

Ένα από τα πιο απλά αλλά πολύ σημαντικό σημείο που πρέπει να κατανοήσουμε είναι ο λεγόμενος κύκλος λειτουργίας μιας CPU. Ο κύκλος αυτός (scancycle) παρουσιάζεται στο επόμενο σχήμα!



2.2 Πλεονεκτήματα PLC συγκριτικά με τον κλασικό αυτοματισμό

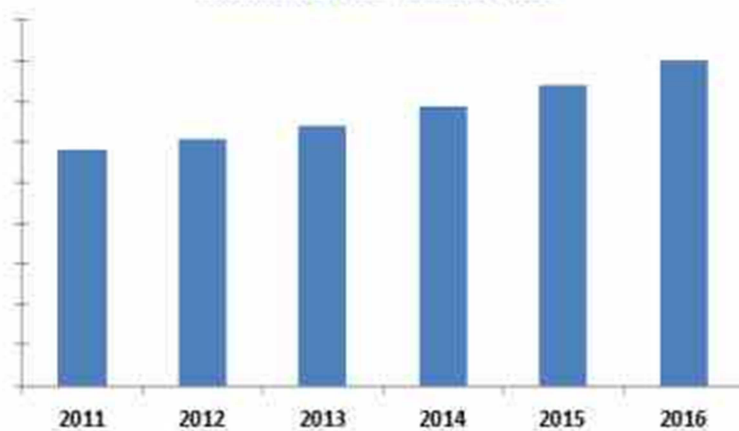
- Είναι συσκευές γενικής χρήσης (δεν είναι κατασκευασμένα για ένα συγκεκριμένο είδος εφαρμογής).
- Δεν ενδιαφέρει ο συνολικός των επαφών χρονικών, απαριθμητών (δεν είναι φυσικά στοιχεία αλλά στοιχεία μνήμης).
- Η λειτουργία του αυτοματισμού μπορεί να αλλάξει σε οποιοδήποτε στάδιο θελήσουμε.
- Εύκολος οπτικός έλεγχος της λειτουργίας ή μη στοιχείων της εγκατάστασης με την βοήθεια των LED που υπάρχουν σε όλες τις κάρτες.
- Με την βοήθεια της προγραμματίστριας μπορούμε να παρακολουθήσουμε την ροή της εκτέλεσης του προγράμματος και μέσω διαγνωστικών εντοπίσουμε τυχόν βλάβες.
- Κάθε αλλαγή στο πρόγραμμα του χρήστη αποθηκεύεται στην μνήμη του PLC, έτσι ο τεχνικός δεν βρίσκεται προ απροόπτου να διαβάσει ένα σχέδιο και άλλο να βρίσκεται πραγματικά στην εγκατάσταση.
- Τα PLC καταλαμβάνουν πολύ μικρό χώρο απ' ότι ένα αντίστοιχος πίνακας αυτοματισμού.
- Μπορούν να τοποθετηθούν και 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.
- Είναι επεκτάσιμα.
- Έχουν μεγάλες δυνατότητες δικτύωσης με πρότυπα βιομηχανικά δίκτυα.
- Μας δίνουν δυνατότητα αντιγραφής εφαρμογών.
- Απαιτούν ελάχιστη συντήρηση.

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 85K
- Μέχρι 1024 εισόδους και εξόδους
- Ενσωματωμένη πολυκομβική διασύνδεση (MPI) για δημιουργία μικρών δικτύων και για σύνδεση με τη προγραμματίστρια μονάδα.
- Μεγάλη ταχύτητα . Μια CPU μπορεί να 1024 δυαδικές πράξεις σε 0.1-0.3 sec.
- MODULAR μορφή.
- Δυνατότητα επέκτασης έως και 32 κάρτες.
- Ενσωματωμένες ειδικές λειτουργίες: counters, positioners, έλεγχος κλειστού βρόγχου με τις CPU 3xxIFM.
- 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 STEP 7.
- Εκτεταμένες διαγνωστικές λειτουργίες μέσω του STEP 7. Μηνύματα σφαλμάτων που αποθηκεύονται στο διαγνωστικό Buffer με αναγραφή ημερομηνίας και ώρας.
- Μεγάλη ποικιλία από CPU για καλύτερη επιλογή αναλόγως εφαρμογής.
- Μεγάλες δυνατότητες δικτύωσης (MPI, Profibus, Industrial Ethernet)
- Μια μόνο κάρτα για όλους τους τύπους αναλογικών σημάτων.
- 32-bit σετ εντολών για μαθηματικές συναρτήσεις.
- Ελεύθερη διευθυνσιοδότηση των καρτών.

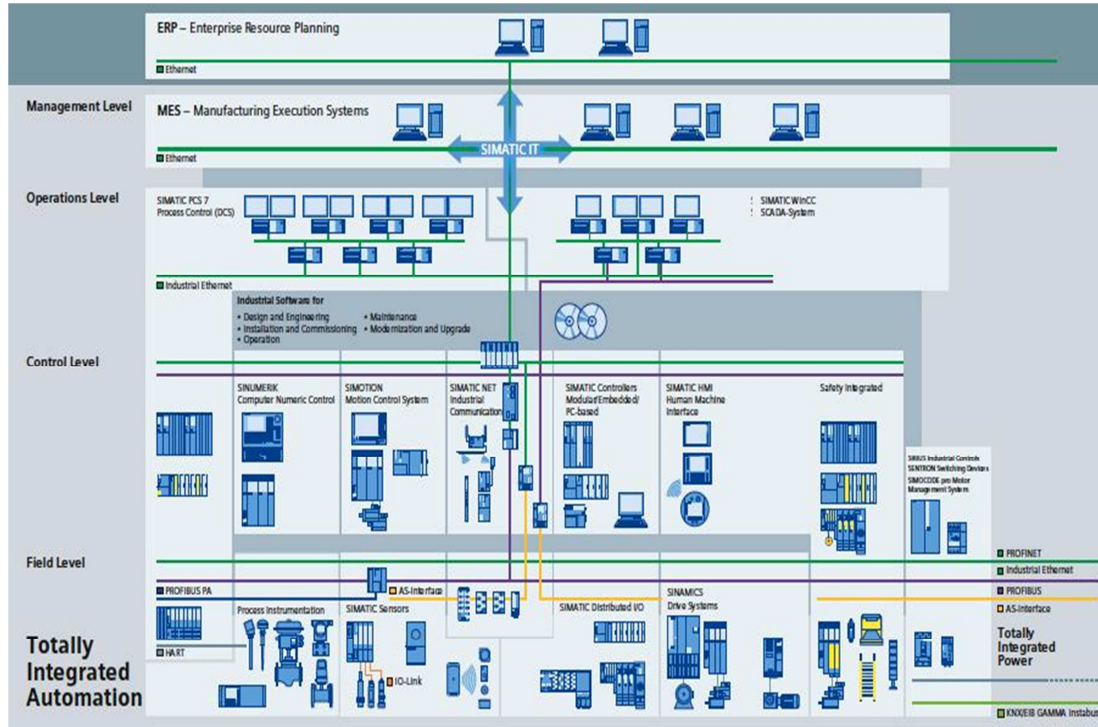
**The Worldwide Programmable Logic Controller (PLC) Market
Including PLC-based PACs**



Εικόνα 2 : Πωλήσεις PLC

2.3 Βιομηχανική Επικοινωνία

2.3.1 Επίπεδα



Εικόνα 3 : Επίπεδα αυτοματοποιημένης διαδικασίας

➤ Επίπεδο παραγωγής :

Το επίπεδο παραγωγής είναι το χαμηλότερο επίπεδο ενός ολοκληρωμένου συστήματος αυτοματισμού και ονομάζεται επίσης επίπεδο των συσκευών. Όλοι οι αισθητήρες και βρίσκονται στο επίπεδο παραγωγής. Σε αυτό το επίπεδο θα βρούμε πολλούς τρόπους για να επικοινωνούν: έχουμε AS-Interface, όπως ένα αισθητήρα διαύλου, PROFIBUS-DP ως διάυλο συσκευής και το PROFIBUS-PA ως fieldbus.

➤ Επίπεδο Ελέγχου :

Στο Επίπεδο ελέγχου θα βρούμε πολλούς πίνακες ελέγχου ή πολλά PLC's που χρησιμοποιούνται για τον έλεγχο των συσκευών που βρίσκονται στο επίπεδο παραγωγής.

➤ Επίπεδο Λειτουργίας :

Σε αυτό το επίπεδο έχουμε τον έλεγχο, παρακολούθηση και τον συντονισμό της διαδικασίας αυτοματισμού.

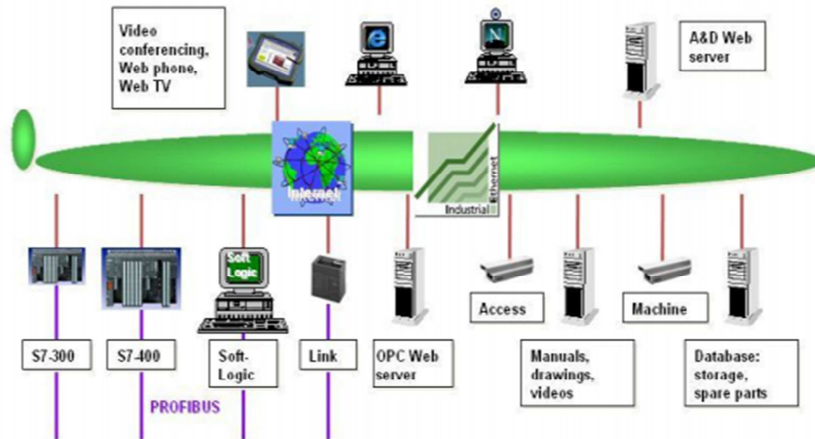
➤ Επίπεδο Διαχείρισης /Διοίκησης :

Manufacturing Executive Systems (MES) βρίσκονται σε αυτό το επίπεδο. MES είναι ένα πληροφοριακό σύστημα που ελέγχει και παρακολουθεί την παραγωγική διαδικασία. Μπορεί να υπολογίσει πολλά πράγματα (κέρδη παραγωγής ...) με βάση την παραγωγική διαδικασία και δημιουργεί αυτόματα τα έγγραφα δεδομένα.

2.3.3 Industrial Ethernet

Η εξέλιξη της βιομηχανικής επικοινωνίας έφερε τη χρήση των δικτύων στην κορυφή όλων. Η Παρακολούθηση και τα συστήματα ελέγχου, όπου τα δεδομένα μεταφέρονται μέσω ενός δικτύου, χρησιμοποιούνται ήδη στις περισσότερες βιομηχανικές ελέγχου και στην παρακολούθηση εφαρμογών. Χρησιμοποιώντας μια κατανομημένη αρχιτεκτονική έχει πολλά πλεονεκτήματα έναντι ενός σχεδιασμού σημείου-προς-σημείο, όπως χαμηλό κόστος εγκατάσταση, εύκολη συντήρηση και την ευελιξία. Σήμερα, οι κορυφαίοι κατασκευαστές συστημάτων ελέγχου και τεχνολογίας παρακολούθησης προσφέρουν διεπαφές δικτύου για τις συσκευές τους. Η Μείωση του κόστους και η αυξανόμενη ζήτηση για ένα ενιαίο, στάνταρ πρωτόκολλο δικτύου, από την αίθουσα συνεδριάσεων έως το εργοστάσιο, έχουν οδηγήσει στην ανάπτυξη των Industrial Ethernet.

Σύμφωνα με τα αποτελέσματα μιας πρόσφατης μελέτης, η παγκόσμια αγορά για τις Ethernet συσκευές αναμένεται να αυξηθεί με ρυθμό περίπου 30 τοις εκατό έως το 2013. Στην αρχή θεωρήθηκε μια λύση που περιορίζεται σε εταιρικά περιβάλλοντα δικτύου, η Ethernet τεχνολογία έχει αποδειχθεί ότι είναι μια ισχυρή εναλλακτική λύση που μπορεί να ικανοποιήσει τις μοναδικές ανάγκες του επιπέδου παραγωγής .



Εικόνα 6 : Εξέλιξη του Industrial Ethernet

Υπάρχουν τρεις κύριες τάσεις που παρατηρούνται στα σύγχρονα βιομηχανικά συστήματα ελέγχου:

- Διανομή και αποκέντρωση των κατασκευών των αυτοματισμών, όπως η ``ευφυΐα`` μετατοπίζεται στο πεδίο των εξαρτημάτων και των μηχανών στην παραγωγή.
- Η αύξηση της κάθετης επικοινωνίας μεταξύ όλων των επιπέδων του συστήματος ελέγχου.
- Η αυξανόμενη ζήτηση για την εφαρμογή των προτύπων IT.

Τα περισσότερα από τα σύγχρονα βιομηχανικά συστήματα αυτοματισμού υιοθετούν πολυεπίπεδη, κάθετη αρχιτεκτονική ελέγχου.

Τα σύγχρονα συστήματα επικοινωνίας για την αυτοματοποίηση περιλαμβάνουν διαφορετικά πρωτόκολλα. Αυτό είναι ένα σημαντικό μειονέκτημα, που οδηγεί στην ανάγκη χρησιμοποίησης υλικού και λογισμικού από συγκεκριμένο πάροχο, τα οποία αυξάνουν το κόστος εγκατάστασης και συντήρησης. Επιπλέον, οι τεχνολογίες που χρησιμοποιούνται σήμερα fieldbus κάνουν την κατακόρυφη επικοινωνία σε όλα τα επίπεδα των συστημάτων αυτοματισμού δύσκολη. Οι πύλες θα πρέπει να χρησιμοποιηθούν για την αποκατάσταση συνδέσεων μεταξύ των διαφόρων ειδών συστημάτων fieldbus που χρησιμοποιείται στο κάτω επίπεδο, και Ethernet που χρησιμοποιούνται στα ανώτερα επίπεδα. Διαφορές στα format των δεδομένων εμποδίζει επίσης την επικοινωνία.

Ethernet Σημαντικά πλεονεκτήματα :	SIMATIC NET Σημαντικές προσθήκες στο Ethernet :
<ul style="list-style-type: none"> - Γρήγορη ανάθεση χάρη σε μια απλή τεχνολογία σύνδεσης. - Υψηλή Διαθεσιμότητα. - Δικτύωση πολλών διαφορετικών εφαρμογών. - Ευρεία επικοινωνία της επιχείρησης χάρη στο WAN (Wireless Area Networking) - ... 	<ul style="list-style-type: none"> - Δικτυακά εξαρτήματα που χρησιμοποιούνται σε σκληρό βιομηχανικό περιβάλλον - Συνεχής επίβλεψη των δικτυακών εξαρτημάτων. - Επικοινωνία Gigabit σε επίπεδο ελέγχου για μεγάλες ποσότητες δεδομένων, όπως για παράδειγμα: εφαρμογές Web.... - ...

Διαθέσιμες λειτουργίες / υπηρεσίες που προσφέρονται από το Industrial Ethernet:

➤ PG / OP επικοινωνία :

Υφίσταται από ενσωματωμένες λειτουργίες επικοινωνίας που επιτρέπει στα SIMATIC συστήματα αυτοματισμού να επικοινωνούν με κάθε HMI (Human Machine Interaction) της συσκευής και SIMATIC PG (Siemens φορητό υπολογιστή για βιομηχανικές εργασίες).

➤ Επικοινωνία S7 :

Επικοινωνία S7 είναι τα μπλοκ λειτουργίας για S7-300, τα οποία έχουν βελτιστοποιηθεί στο πλαίσιο SIMOTION, SINUMERIK και SIMATIC S7/WinAC. Ενεργοποιεί, επίσης, υπολογιστές και σταθμούς εργασίας που πρόκειται να συνδεθεί. Η επικοινωνία S7 προσφέρει απλή, ισχυρή επικοινωνία και μια διασύνδεση λογισμικού ανεξάρτητα από το δίκτυο.

➤ Open.IE/S5-Συμβατή επικοινωνία. (Αποστολή / λήψη) :

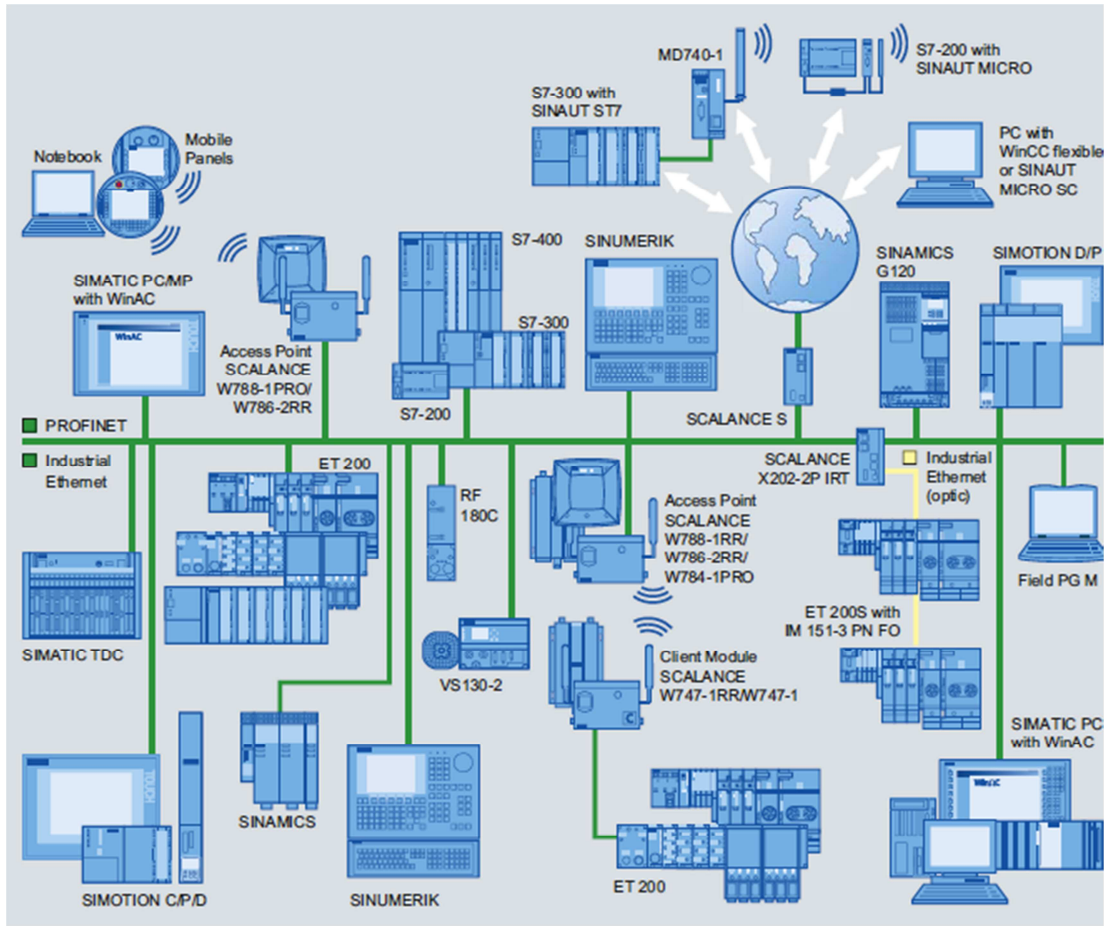
Αυτό επιτρέπει στο SIMATIC S7 να επικοινωνεί με τα υπάρχοντα συστήματα, κυρίως με SIMATIC S5, αλλά και με τον υπολογιστή μέσω του PROFIBUS και Industrial Ethernet.

➤ Τυπική Επικοινωνία :

Περιλαμβάνει τυποποιημένα πρωτόκολλα για την επικοινωνία δεδομένων: FTP, e-mail, το πρωτόκολλο TCP / IP, ...

➤ Υπηρεσίες επικοινωνίας PROFINET :

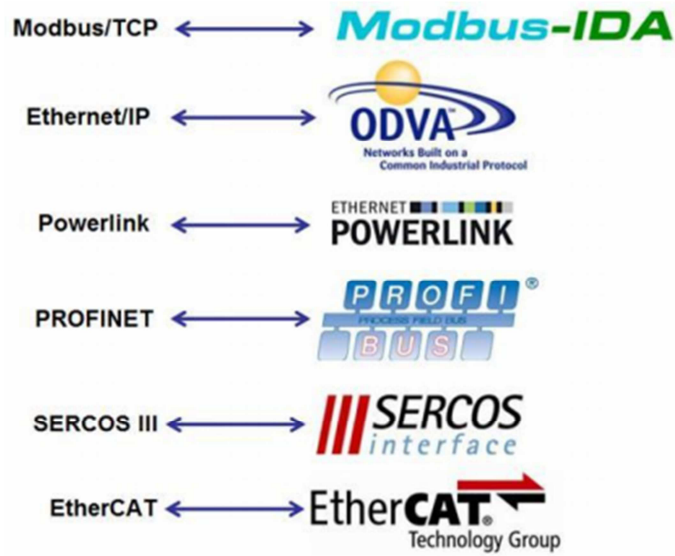
PROFINET IO για τη σύνδεση των συσκευών του χαμηλότερου επιπέδου που διανέμονται σε Industrial Ethernet και για την απλή machine-machine επικοινωνία. PROFINET CBA (Component Based Automation) για σύγχρονο εργοστάσιο με σκοπό να συμπεριλάβει διανεμημένες δομές αυτοματισμού βάσει των προκατασκευασμένων στοιχείων.



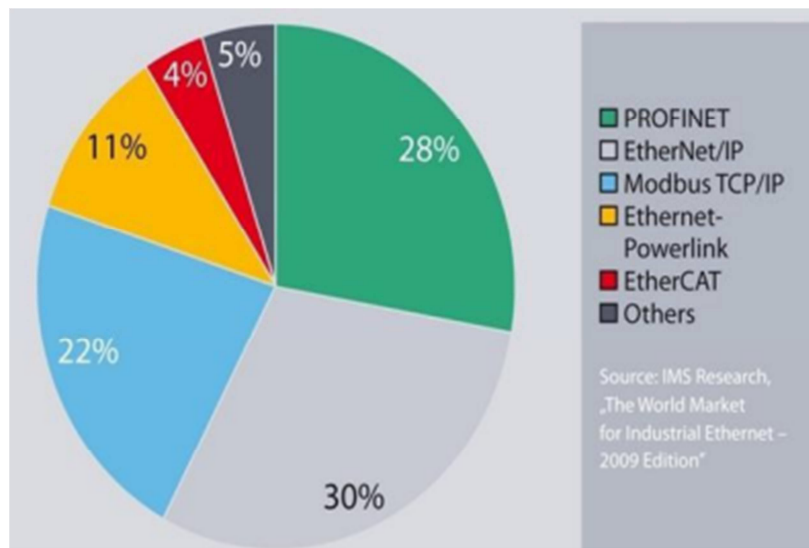
Εικόνα 7 : Industrial Ethernet/PROFINET

Η βασική διαφορά ανάμεσα Industrial Ethernet και παραδοσιακό Ethernet είναι το είδος του υλικού που χρησιμοποιείται. Ο εξοπλισμός του βιομηχανικού ethernet έχει σχεδιαστεί για να λειτουργεί σε σκληρές περιβαλλοντικές συνθήκες. Περιλαμβάνει βιομηχανικής ποιότητας υλικά, μετάδοση θερμότητας, και τη σηματοδότηση εξόδου ρελέ. Και είναι σχεδιασμένα για να λειτουργούν σε ακραίες θερμοκρασίες και κάτω από ακραίες δονήσεις και κρούσεις. Οι απαιτήσεις ισχύος για βιομηχανικά περιβάλλοντα διαφέρουν από τα δίκτυα δεδομένων, έτσι ώστε η εξοπλισμός τρέχει χρησιμοποιώντας 24 βολτ συνεχούς ρεύματος. Για τη μεγιστοποίηση της διαθεσιμότητας του δικτύου, το βιομηχανικό ethernet είναι ανεκτικό σε σφάλματα, όπως η εναλλαγή στην τάση του ρεύματος. Ο εξοπλισμός είναι επίσης ολοκληρωμένος, προκειμένου να ανταποκριθεί στις εξαιρετικά ποικίλες απαιτήσεις στον χώρο του εργοστασίου.

Τα βιομηχανικά δίκτυα με industrial ethernet θα γίνουν αναπόσπαστο μέρος των συστημάτων ελέγχου σε πραγματικό χρόνο για τον έλεγχο της διαδικασίας και των βιομηχανικές εφαρμογές. Τα πιο δημοφιλή πρωτόκολλα φαίνονται παρακάτω :



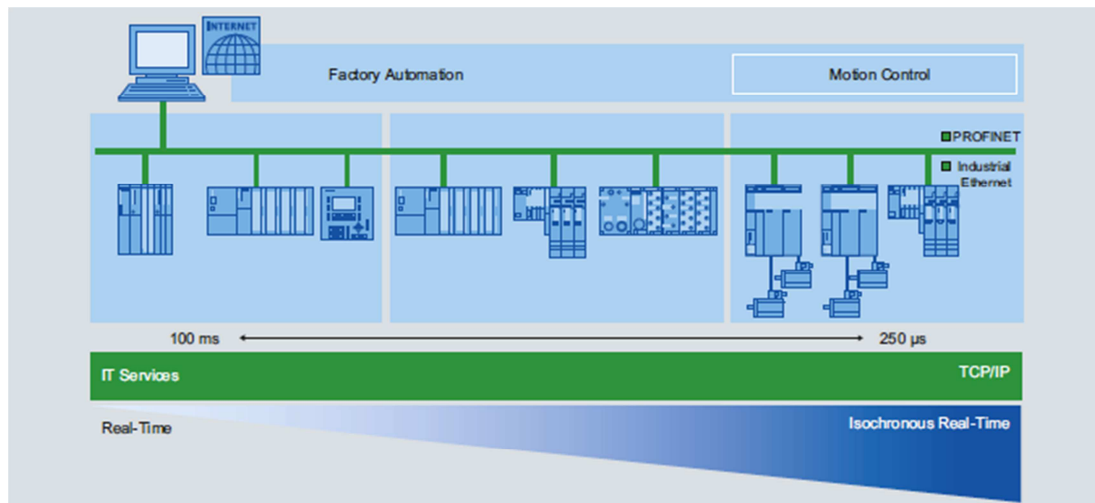
Εικόνα 8 : Industrial Ethernet Protocols



Εικόνα 9 : Χρήση των πρωτοκόλων

2.3.4 PROFINET

PROFINET είναι το ανοιχτό βιομηχανικό πρότυπο Ethernet για βιομηχανικό αυτοματισμό. Με PROFINET, συσκευές από το επίπεδο παραγωγής (χαμηλότερο επίπεδο) μπορούν να συνδεθούν μέσω των MES (Manufacturing Execution Systems) επίπεδο. Το PROFINET επιτρέπει το σύστημα ευρείας επικοινωνίας, αυτό σημαίνει ότι μπορούμε εύκολα να συνδεθούν όλα τα είδη των συσκευών, επειδή το PROFINET χρησιμοποιεί τα πληροφοριακά πρότυπα κατευθείαν στο επίπεδο παραγωγής. Όπως μπορούμε να δούμε στην παρακάτω εικόνα, το PROFINET χρησιμοποιεί τυποποιημένο πρωτόκολλο TCP / IP για την παραμετροποίηση, τη διαμόρφωση και τη διάγνωση. Η Επικοινωνία σε πραγματικό χρόνο για τη διαβίβαση των δεδομένων στους χρήστες εκτελείται στην ίδια γραμμή.



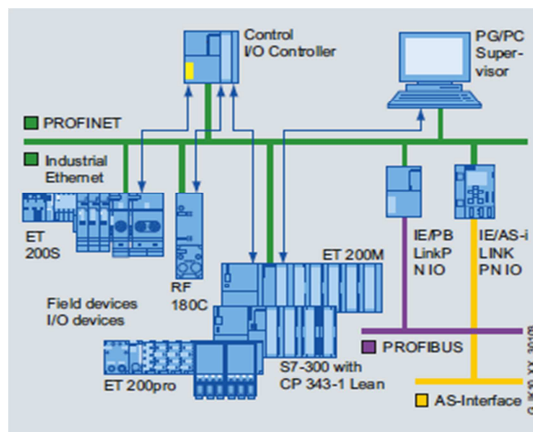
Outstanding feature of PROFINET: Integrated real-time communication with simultaneous, unrestricted TCP/IP communication

Εικόνα 10 : Χρόνος απόκρισης PROFINET

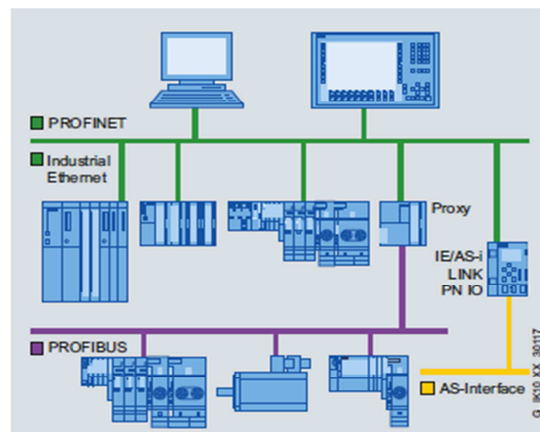
Σημαντικές πτυχές του PROFINET :

<p>Real-Time Επικοινωνία :</p>	<p><u>Real-Time (RT)</u> Χρησιμοποιείται για την επεξεργασία δεδομένων όταν ο χρόνος είναι κρίσιμος. PROFINET χρησιμοποιεί ένα βελτιστοποιημένο κανάλι επικοινωνίας σε πραγματικό χρόνο, στο οποίο η απόδοση υπερτερεί αυτής ενός συμβατικού fieldbus .RT είναι η επικοινωνία σε πραγματικό χρόνο για τις συνήθεις εφαρμογές, προκειμένου να συνδεθούν συσκευές στο επίπεδο παραγωγής.</p> <p><u>Ισόχρονη Real-Time (IRT)</u> IRT χρησιμοποιείται για ιδιαίτερα απαιτητικές εφαρμογές, όπως για παράδειγμα: εφαρμογές ελέγχου κίνησης. Ο κύκλος του χρόνου στην IRT είναι πολύ πιο γρήγορος από ότι στην RT.</p>
---------------------------------------	--

<p>ERTEC (Ενισχυμένος Real-Time Ethernet Ελεγκτής)</p>	<p>Υποστηρίζει την επικοινωνία σε πραγματικό χρόνο (RT και IRT) και είναι η βασική τεχνολογία για σύστημα ολοκληρωμένων λύσεων με PROFINET.</p>
<p>Κατανεμημένες συσκευές (PROFINET IO)</p>	<p>PROFINET επιτρέπει κατανεμημένες συσκευές στο επίπεδο παραγωγής να ενσωματωθούν άμεσα σε Industrial Ethernet. Υπάρχουσες μονάδες ή συσκευές μπορούν να συνεχίσουν να χρησιμοποιούνται με συμβατές PROFINET διεπαφές ή συνδέσμους (IE / PB & IE / AS).</p>
<p>Ενσωμάτωση Fieldbus (Via Proxy)</p>	<p>PROFINET επιτρέπει την εύκολη ενσωμάτωση της υφιστάμενης fieldbus. Αυτό μπορεί να είναι ένα PROFIBUS ή ένα σύστημα AS-Interface. Γι 'αυτό χρειαζόμαστε ένα Proxy, το οποίο είναι master του PROFIBUS ή του συστήματος AS-Interface από τη μία πλευρά και ένας σταθμός στο Βιομηχανικό Ethernet από την άλλη πλευρά.</p>



PROFINET with distributed field devices



Fieldbus integration via a proxy

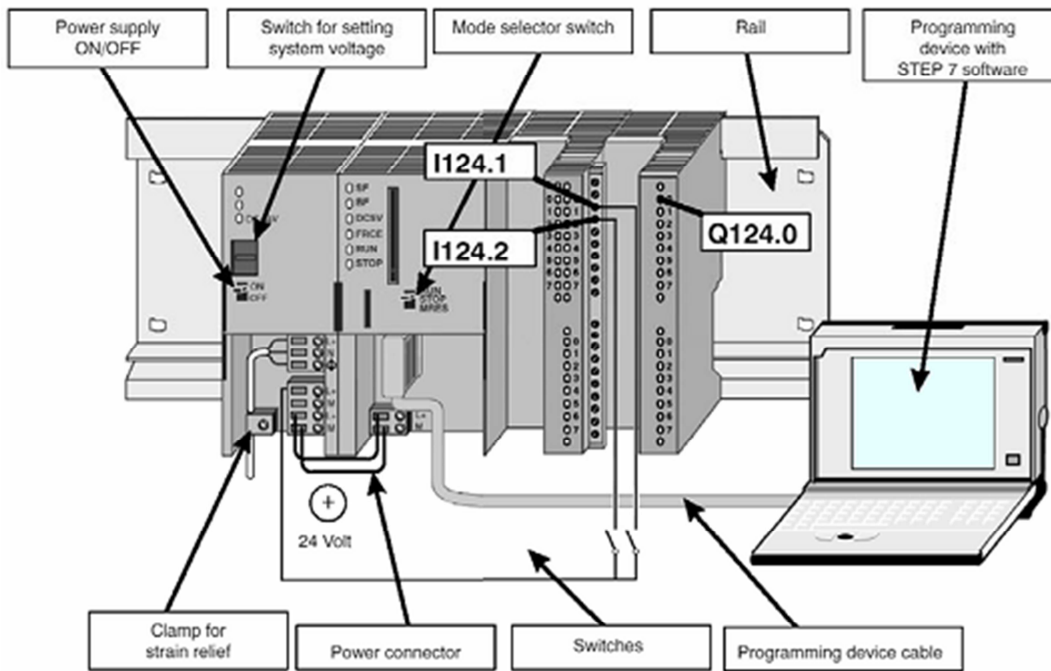
Εικόνα 11 : PRONINET & PROFIBUS

ΚΕΦΑΛΑΙΟ 3

Η ΔΟΜΗ ΤΩΝ PLC


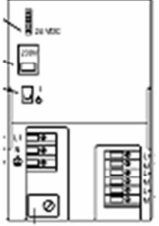
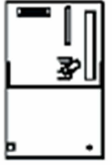
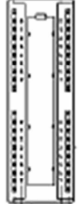


3.1 Βασική δομή των PLC

Κάθε PLC μπορεί να δομηθεί από επιμέρους μονάδες ανάλογα με την εφαρμογή για την οποία θα χρησιμοποιηθεί.



Εικόνα 12 : Δομή των PLC

Τα σημαντικότερα στοιχεία μιας εφαρμογής με PLC της σειράς S7-300 δίνονται στους παρακάτω πίνακες.

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 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).	

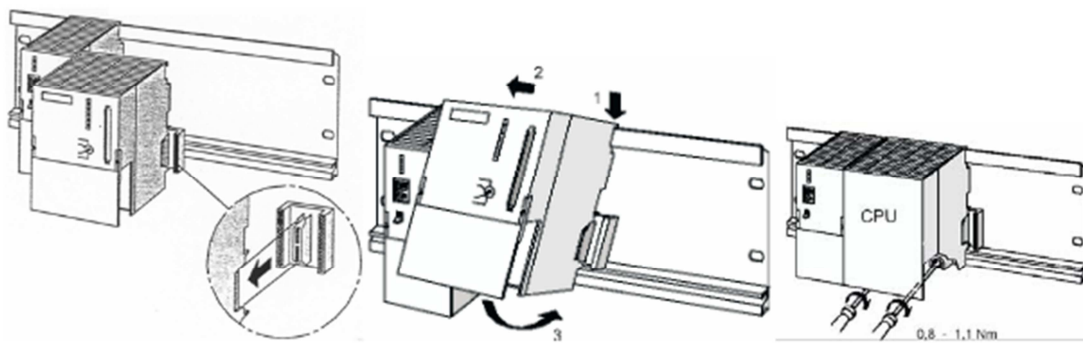
Εικόνα 13 : Στοιχεία του PLC

Ας δούμε όμως αναλυτικά τα βασικά στοιχεία μιας S7 – 300 δομής.

3.1.1 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 3 η οποία είναι αφιερωμένη για την κάρτα διασύνδεση 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).

3.1.2 Στήριξη Καρτών στο Rack:



Εικόνα 14 : 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 συνδεδηρά σχήματος «Π» στο πίσω μέρος των καρτών. Μέσω αυτού υλοποιούνται δύο δίαυλοι εσωτερικής επικοινωνίας:

P – Bus (PeripheralBus):

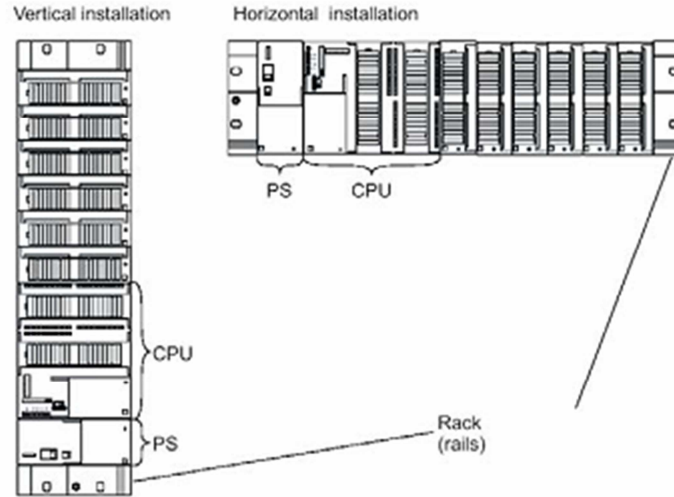
Αυτό έχει σαν κύριο στόχο να μεταφέρει πληροφορίες που αφορούν την «περιφέρεια» (επικοινωνία με κάρτες εισόδου ή εξόδου) με ταχύτητα 1,5Mbps.

K – Bus (CommunicationBus) :

Αφορά την επικοινωνία με τις λεγόμενες «ειδικές» κάρτες (κάρτες απαρίθμησης, PID, FM, CP ...). Και στο K – Bus η Πληροφορία μεταφέρεται σειριακή με ταχύτητα 187,5 Kbps.

Εγκατάσταση :

Ένα σύστημα της σειράς S7-300 μπορεί να τοποθετηθεί οριζόντια ή κάθετα όπως δείχνει η επόμενη εικόνα. Σε κάθε περίπτωση το τροφοδοτικό και η CPU ή θα βρίσκονται αριστερά του συστήματος ή κάτω.



Εικόνα 15 : Κάθετη και οριζόντια εγκατάσταση

Στην επόμενη εικόνα παρουσιάζεται η μέγιστη δυνατή σύνθεση ενός συστήματος S7-300.

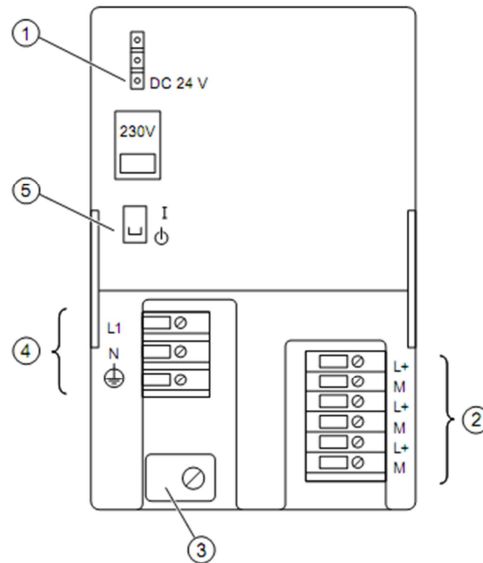


Εικόνα 16 : Μέγιστος αριθμός Module

3.2 Τροφοδοτικό PS (PowerSupply)

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. Για την σειρά S7 – 300 έχουμε τις εξής επιλογές:

Στην κατασκευή μας, χρησιμοποιούμε ένα τροφοδοτικό PS 307: 5A. Μορφολογικά αυτό παρουσιάζεται στην επόμενη εικόνα.

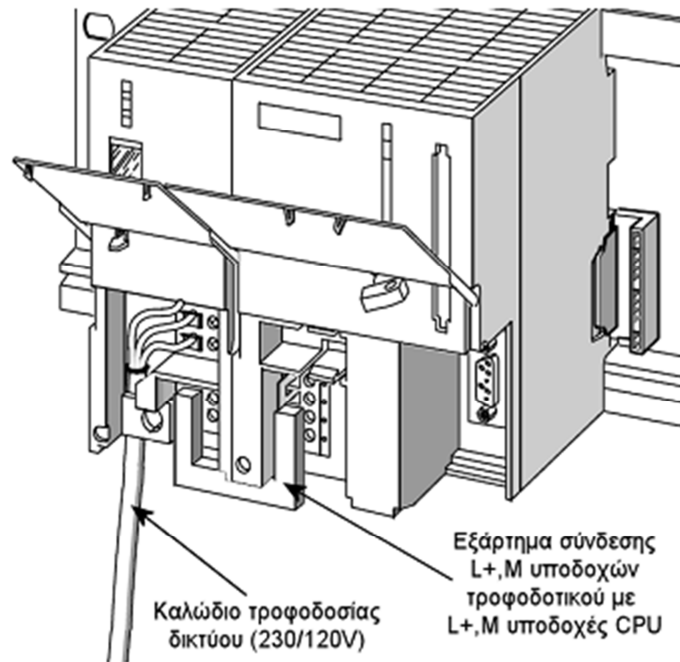


Εικόνα 17 : Τροφοδοτικό

Διαθέτει:

- ① "24 VDC output voltage present" display
- ② Terminals for 24 VDC output voltage
- ③ Strain relief
- ④ Mains and protective conductor terminals
- ⑤ 24 VDC On/Off switch
- ⑥ Mains selector switch

Στην επόμενη εικόνα παρουσιάζεται ο τρόπος καλωδίωσης μεταξύ τροφοδοτικού και CPU.



Εικόνα 18 : Καλωδίωση Τροφοδοτικού

3.3 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 μας.
- Αποθήκευση των πληροφοριών.
- Εκτέλεση αριθμητικών πράξεων.

3.3.1 Εσωτερικά η CPU Περιέχει

I. 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.

II. Την Μνήμη

Η μνήμη μιας CPU χωρίζεται σε τρεις κατηγορίες.

1. Μνήμη φόρτωσης (LoadMemory)
2. Μνήμη εργασίας (Workmemory)
3. Μνήμη συστήματος (Systemmemory)

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

- **Μνήμη απεικόνισης εισόδων ΡΙΙ**

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 ΡΙQ**

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 οποία έχουν υπολογιστεί κατά την εκτέλεση του προγράμματος.

- **Χρονικά Τ (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.

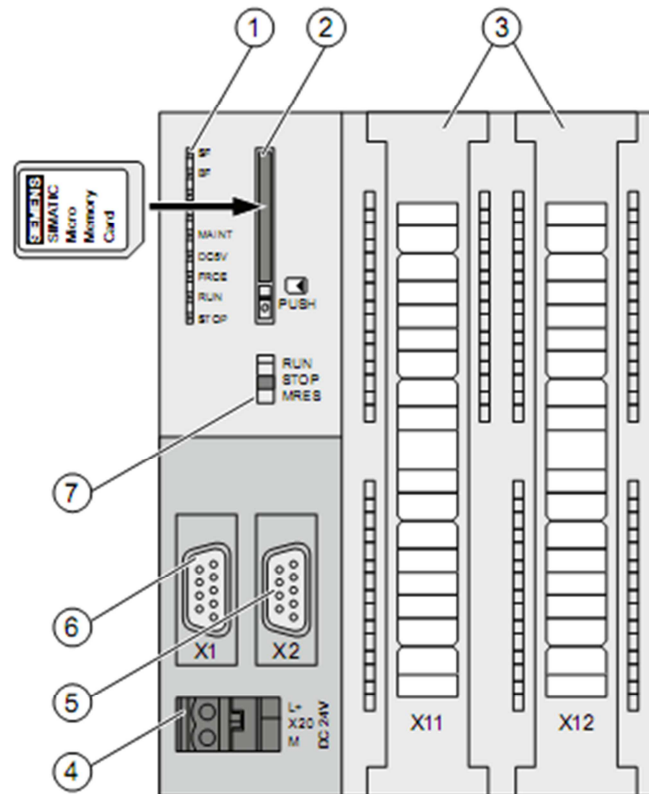
- **Τοπικά βοηθητικά L (LocalData)**

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 το οποίο το περιέχει.

- **Διαγνωστικά (Diagnostics)**

Καταχωρούνται διάφορες ενέργειες που έχουν γίνει στο σύστημα με ώρα και ημερομηνία όπως CPU σε RUN/STOP, βραχυκυκλωμένη κάρτα αναλογικών.

3.3.2 Εξωτερικά μια CPU παρουσιάζει:



Εικόνα 19 : CPU 31x

- ① Status and error indicators
- ② Slot for the SIMATIC Micro Memory Card incl. the ejector
- ③ Terminals of the integrated inputs and outputs
- ④ Power supply connection
- ⑤ 2. interface X2 (DP)
- ⑥ 1. interface X1 (MPI)
- ⑦ Mode selector

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 intarface.
- Πλήθος των εισόδων/ εξόδων που υποστηρίζουν
- Μέγεθος της ενσωματωμένης μνήμης RAM

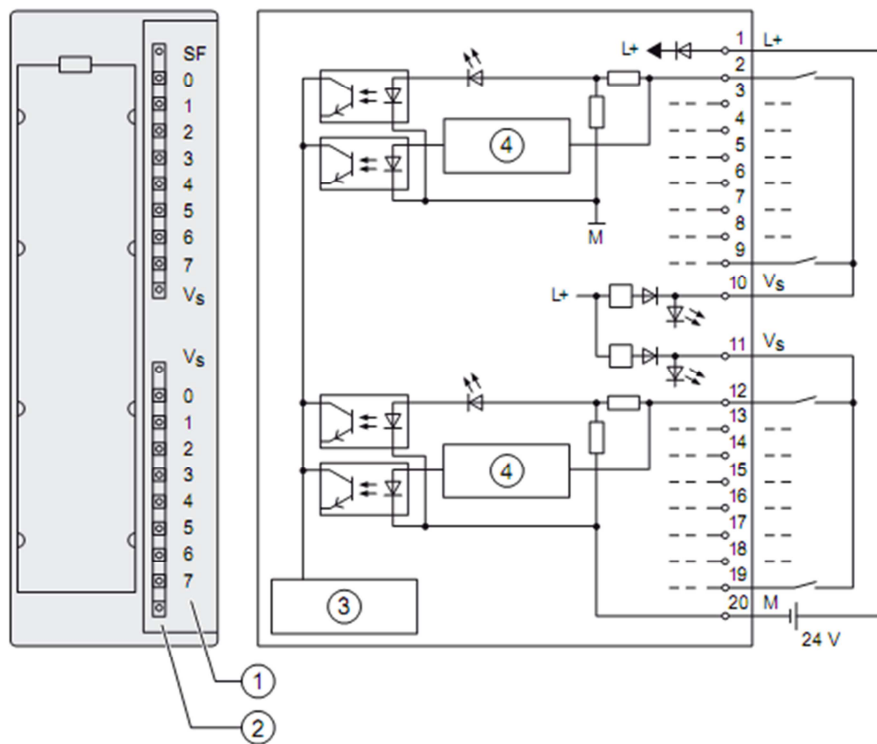
3.4 Ψηφιακές Μονάδες Εισόδων DI (Digital Input)

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

Έτσι σαν σήμα «+1» 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.



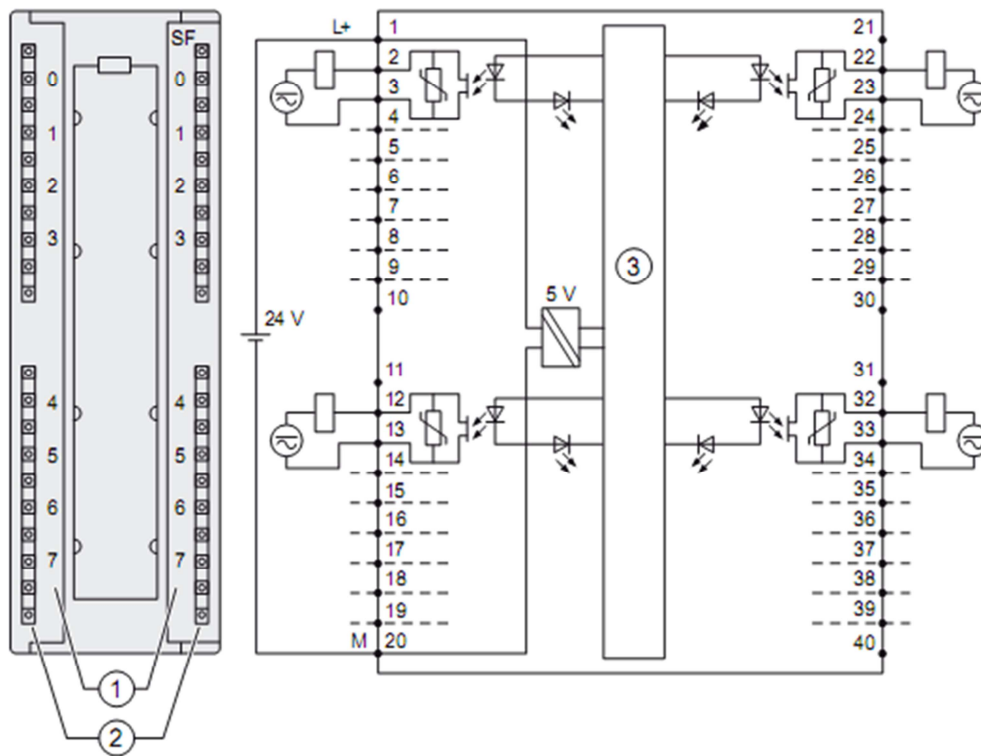
Εικόνα 20 : Ψηφιακή κάρτα εισόδων

3.5 Ψηφιακές Μονάδες Εξόδων 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.



Εικόνα 21 : Ψηφιακή κάρτα εξόδων

- ① Channel number
- ② Status LEDs - green
- ③ Backplane bus interface

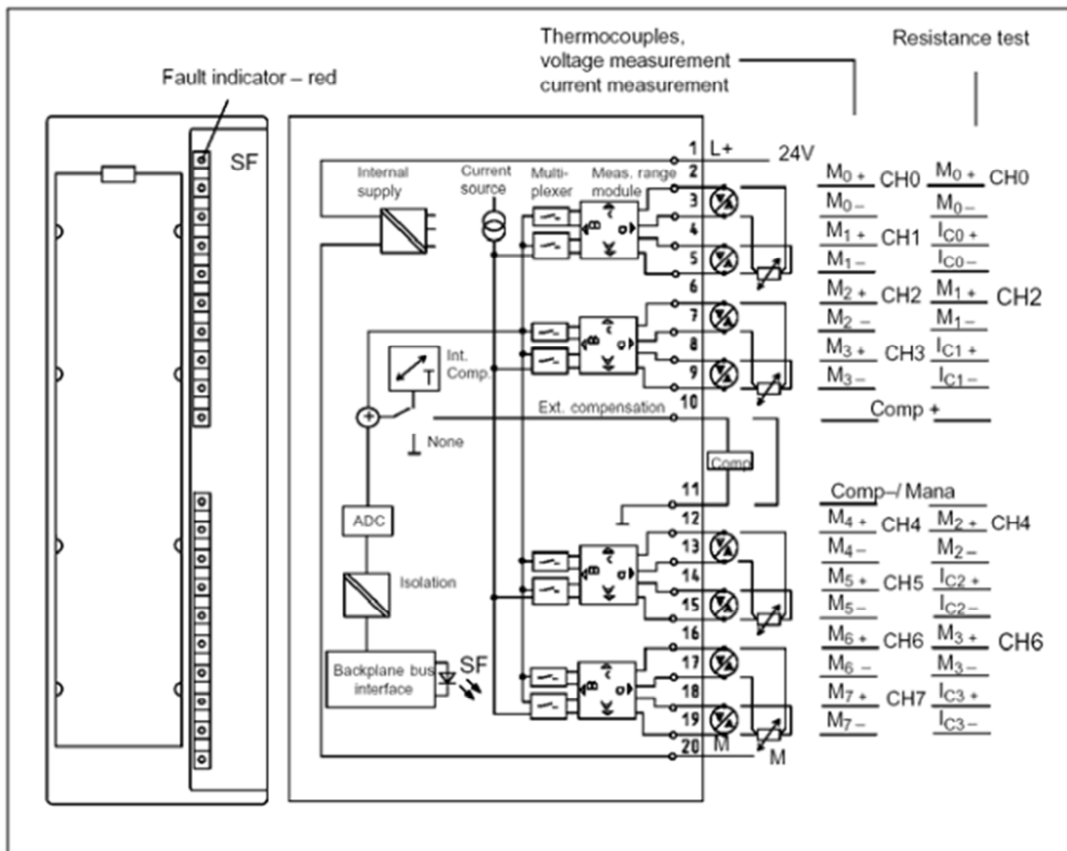
Ένα επιπλέον ιδιαίτερο χαρακτηριστικό των DO είναι το στοιχείο εξόδου (αυτό που παρέχει την ισχύ στο φορτίο). Αυτό συνήθως είναι τρανζίστορ αν πρόκειται για DC κάρτα εξόδων ή τριακ ή ρελέ εάν πρόκειται για AC κάρτα εξόδων. Όλες οι ψηφιακές έξοδοι είναι γαλβανικά απομονωμένες.

3.6 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 (δυναδική 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 ± 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.

Στην κάτω εικόνα παρουσιάζεται η μορφολογία και η αρχή λειτουργίας μιας αναλογικής κάρτας εισόδων.



Εικόνα 22 : Αναλογική κάρτα εισόδων

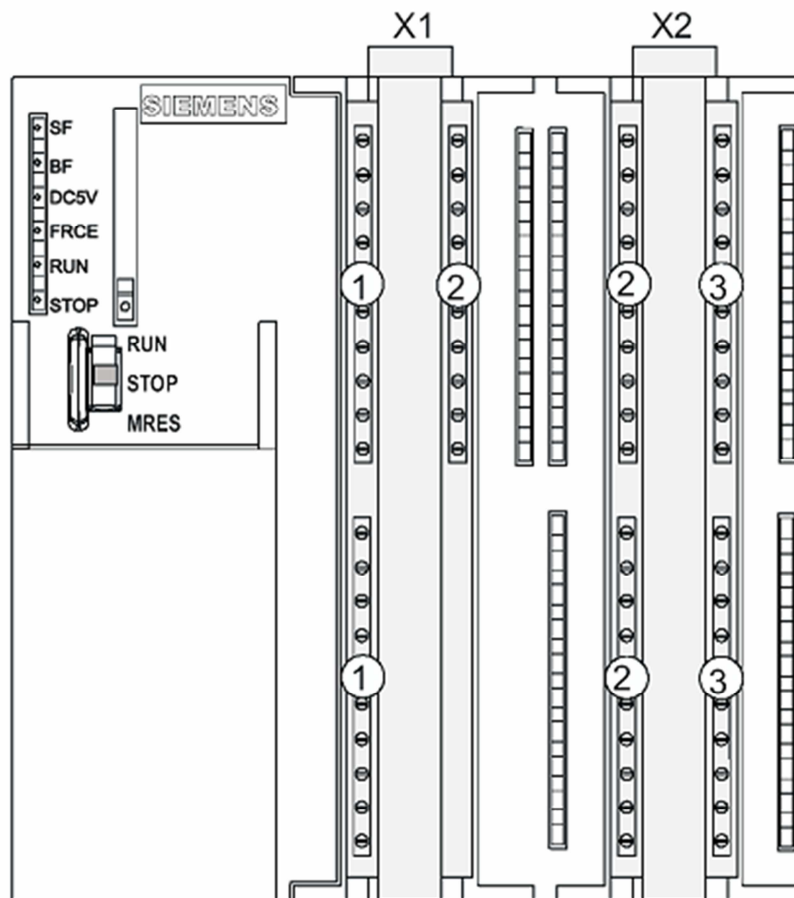
Ένα ακόμα μεγάλο πλεονέκτημα της σειράς S7 είναι ότι μια αναλογική κάρτα εισόδων μπορεί να γίνει τάσης ή έντασης και να μεταβάλουμε την περιοχή μέτρησης της επεμβαίνοντας τόσο εξωτερικά πάνω στην ίδια την κάρτα όσο και στο software.

3.7 Μονάδες Αναλογικών Εξόδων AO (Analog Output)

Οι κάρτες αναλογικών εξόδων έχουν τον ρόλο να μετατρέψουν το αριθμητικό μέγεθος με το οποίο «σκέπτεται» 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.

*Το σύστημα αυτοματισμού στο οποίο βασίζεται η πυρηνική εργασία διαθέτει την **CPU314C-2DP**.*



Εικόνα 23 : Cpu 314C-2DP

- ① Analog outputs and analog inputs
- ② Digital inputs
- ③ Digital outputs

ΚΕΦΑΛΑΙΟ 4

SIMATIC MANAGER

4.1 Προγραμματισμός με το Simatic Step7

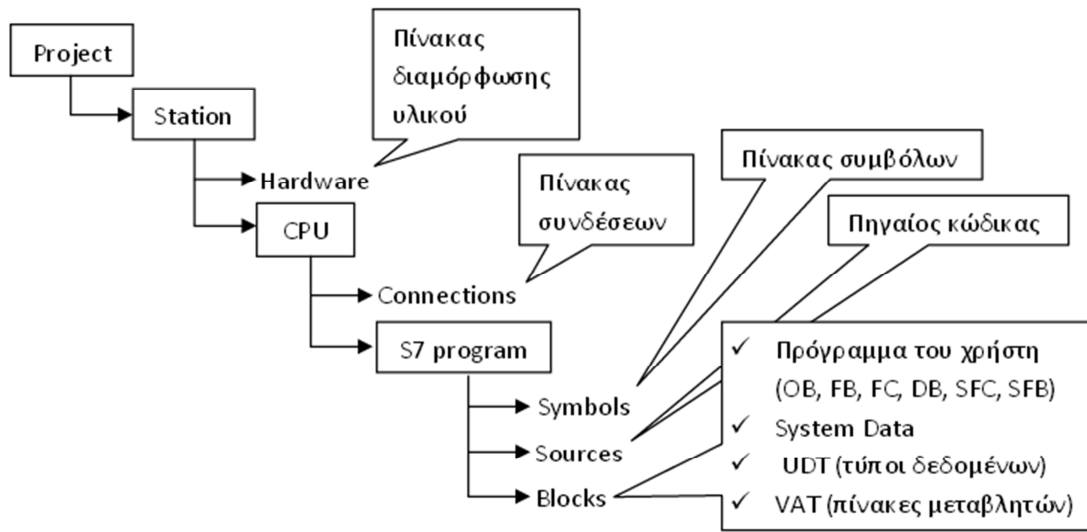
Ο προγραμματισμός των ελεγκτών S7-300 γίνεται με το πρόγραμμα STEP7. Το STEP7 είναι κάτι παραπάνω από ένα απλό πακέτο προγραμματισμού ενός PLC.

Καταρχήν, έχει ενσωματωμένες τουλάχιστον τρεις γλώσσες προγραμματισμού (LAD/S TL/FBD) ενώ με κάποια επιπρόσθετα πακέτα γίνονται διαθέσιμες και άλλες γλώσσες προγραμματισμού όπως η SCL (Structured Control Language) και η GRAPH (ανάλογη με την SFC-Sequential Function Chart του προτύπου IEC11313). Ο κώδικας μπορεί να γραφεί σε οποιαδήποτε από τις τρεις ενσωματωμένες γλώσσες.

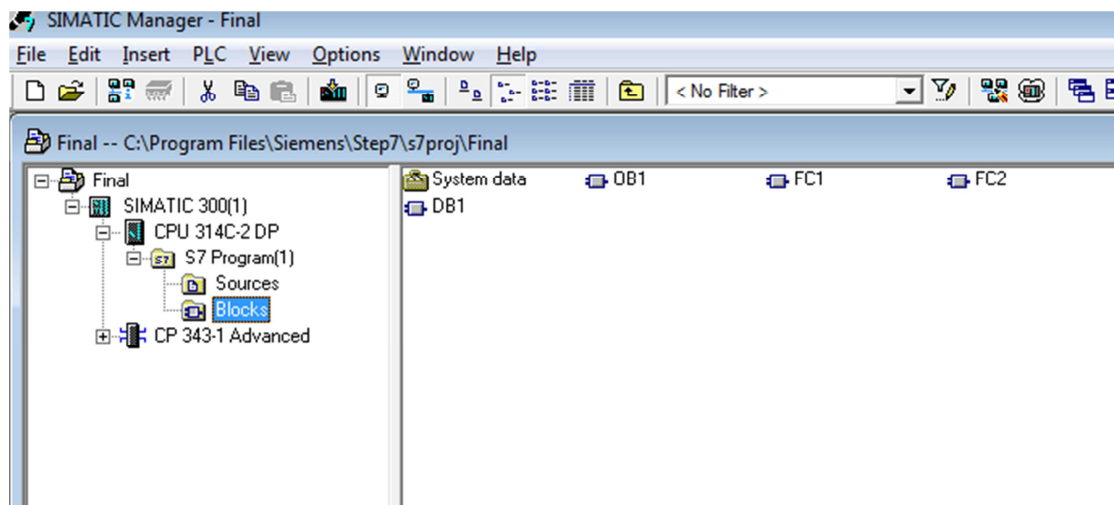
Για την κατασκευή του προγράμματος υποστηρίζονται διάφοροι τύποι υποπρογραμμάτων και κάθε υποπρόγραμμα μπορεί να γραφεί σε οποιαδήποτε από τις γλώσσες ανεξάρτητα από αυτή που χρησιμοποιήθηκε για τα υπόλοιπα. Έτσι, ανάλογα με την περίπτωση χρησιμοποιείτε και η κατάλληλη γλώσσα διευκολύνοντας έτσι το προγραμματισμό. Επιπρόσθετα το Step7 διαθέτει εργαλεία διαρρύθμισης του υλικού (Hardware configuration), δημιουργίας και παραμετροποίησης δικτύων (Netpro), εύρεσης σφαλμάτων (debugger), παρακολούθησης κατάστασης του ελεγκτή (διαγνωστικά) και λειτουργίες εποπτείας των περιεχομένων της μνήμης.

Ένα άλλο πλεονέκτημα του Step7 είναι η δυνατότητα απομακρυσμένης πρόσβασης σε αυτό μέσω modem και διαδικτύου. Δεν χρειάζεται να είναι κανείς τοπικά συνδεδεμένος στο σταθμό αυτοματισμού ή το δίκτυο για να προγραμματίσει, να κάνει ενημερώσεις ή να ανιχνεύσει προβλήματα. Το πακέτο Teleservice του Step7 παρέχει τη δυνατότητα σύνδεσης μιας μονάδας προγραμματισμού ή ενός υπολογιστή σε ένα PLC μέσω του τηλεφωνικού δικτύου ή του διαδικτύου. Όσον αφορά τα διαγνωστικά δεν χρειάζεται να γραφεί η διαγνωστική λογική για να διαπιστωθεί τι είναι λανθασμένο με το υλικό ή το λογισμικό. Όταν η CPU αντιληφθεί ένα σφάλμα τότε καταχωρείται η αιτία του σφάλματος σε μια λίστα που μπορούμε να τη διαβάσουμε μέσα από το Step7, σταματάει η λειτουργία της και δίνει ένα μπλοκ διακοπής που περιέχεται με λεπτομέρεια ο λόγος διακοπής. Στο Step7 όλες οι απαιτήσεις σε λογισμικό και υλικό μιας διαδικασίας αυτοματισμού οργανώνονται και διαχειρίζονται μέσα από ένα project. Το project περιλαμβάνει τη διαμόρφωση του υλικού, τη διαμόρφωση του δικτύου, όλα τα προγράμματα και ολόκληρη τη διαχείριση δεδομένων για μια λύση αυτοματισμού. Ο Simatic Manager είναι ο κορμός του Step7 αφού αυτός είναι που διαχειρίζεται τα projects.

Η δομή ενός project φαίνεται στην παρακάτω εικόνα.



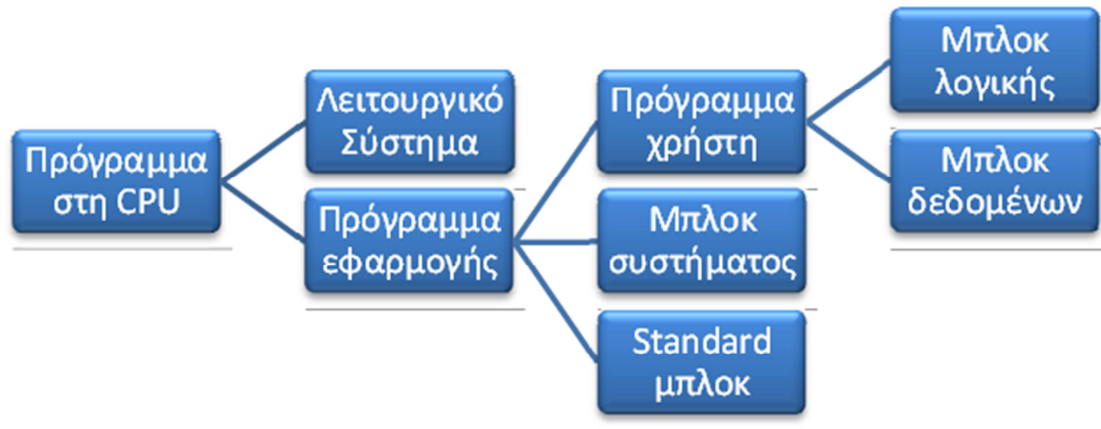
Εικόνα 24 : Γενική δομή project



Εικόνα 25 : Δομή του δικού μας project

4.2 Η Δομή του προγράμματος

Κάθε CPU περιλαμβάνει δύο προγράμματα, ανεξάρτητα το ένα από το άλλο, το λειτουργικό σύστημα και το πρόγραμμα εφαρμογής.



Εικόνα 26 : Δομή του προγράμματος

Το λειτουργικό σύστημα είναι το σύνολο των ορισμών και εντολών που ελέγχουν τους πόρους του συστήματος, Είναι συνιστώσα της CPU και δεν έχουμε δυνατότητα να το τροποποιήσουμε. Μπορούμε όμως να διαβάσουμε και να εκμεταλλευτούμε τα αποτελέσματα των ενεργειών του στο δικό μας πρόγραμμα. Το πρόγραμμα εφαρμογής χωρίζεται στο πρόγραμμα του χρήστη, στα μπλοκ συστήματος και στα standard (IEC) μπλοκ. Το πρόγραμμα του χρήστη είναι το πρόγραμμα που εμείς γράφουμε για να υλοποιήσουμε το αυτοματισμό. Αυτό μπορεί να περιέχει μπλοκ λογικής και μπλοκ δεδομένων.

Όνομα μπλοκ	Περιγραφή
Organization Blocks (OB)	Παίζουν το ρόλο του μεσάζοντα μεταξύ του λειτουργικού συστήματος και του προγράμματος μας. Το λειτουργικό σύστημα της CPU καλεί τα OB όταν συμβεί κάποιο ειδικό γεγονός (πχ κάποιο σφάλμα).
Function Blocks (FB)	Τα FB είναι συναρτήσεις που έχουν «μνήμη», δέχονται δηλαδή ένα μπλοκ δεδομένων (IDB) ως περιοχή αποθήκευσης των δεδομένων.
Functions (FC)	Τα FC περιέχουν κομμάτια κώδικα που επαναλαμβάνονται συχνά.
Data blocks (DB)	Τα DB περιέχουν δεδομένα. Ορίζουμε σε ποια μορφή θα σώζονται, με ποια σειρά και τι τύπου είναι.
Instance Data Blocks (IDB)	Τα IDB ανατίθενται στα FB ή τα SFB όταν καλούνται από το πρόγραμμα και αποτελούν τη μνήμη τους.

Εικόνα 27 : Block και ιδιότητες

Τα OB καθορίζουν τη δομή του προγράμματος. Το σημαντικότερο και άκρως απαραίτητο σε κάθε πρόγραμμα είναι το OB1, αφού αυτό είναι που επεξεργάζεται κυκλικά η CPU και από εδώ καλούμε όλα τα υπόλοιπα μπλοκ του προγράμματος μας. Ένα άλλο σημαντικό μπλοκ είναι το OB100, το οποίο εκτελείται μόνο μια φορά, μετά την παροχή τάσης στο σύστημα ή μετά από warmrestart και πριν την κυκλική επεξεργασία του OB1.

Τα μπλοκ συστήματος (SFB – system FB και SFC system FC) περιέχουν χρήσιμες λειτουργίες που είναι από πριν ορισμένες και καταχωρημένες στο λειτουργικό σύστημα του PLC. Αυτά δεν καταλαμβάνουν χώρο από τη μνήμη που είναι διαθέσιμη στο χρήστη για να γράψει το δικό του πρόγραμμα. Ο χρήστης καλεί αυτά τα μπλοκ και παίρνει τα αποτελέσματα χωρίς να τον ενδιαφέρει το πώς παράχθηκαν. Τα standard (IEC) μας προσφέρουν έτοιμες λύσεις για τυποποιημένες εργασίες αυτοματισμού που πιθανόν μας ενδιαφέρουν.

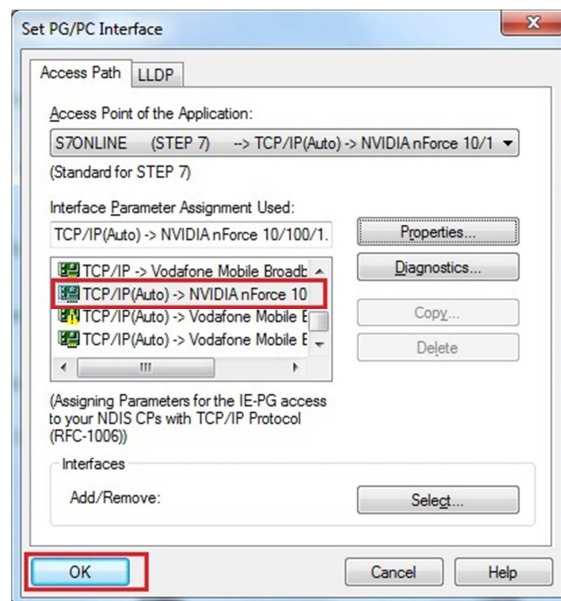
4.3 Ορισμός και Διαρρύθμιση

Αρχικά πριν προχωρήσουμε στον ορισμό του πρώτου σταθμού και τη διαρρύθμιση του πρέπει να ορίσουμε τον τρόπο επικοινωνίας της συσκευής προγραμματισμού με το PLC. Αυτό θα δούμε πως γίνεται στις παρακάτω εικόνες.

Ανοίγοντας το Control Panel βλέπουμε πως το πρόγραμμα μας έχει εγκαταστήσει το Set PC/PG Interface το οποίο και επιλέγουμε.

Στην Συγκεκριμένη περίπτωση θέλουμε να επικοινωνούμε με το PLC μέσω του δικτύου οπότε επιλέγουμε TCP/IP (Auto).

Οι ρυθμίσεις που απομένουν θα ολοκληρωθούν μέσω του Simatic Manager αφού πρώτα φτιάξουμε το πρώτο μας Station στην συνέχεια.

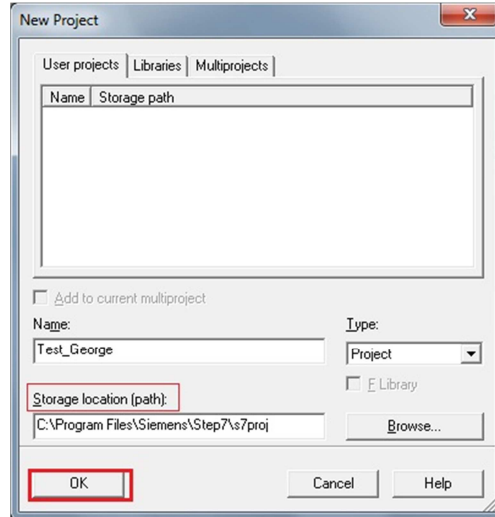


Εικόνα 28 : PC/PG interface

4.3.1 Δημιουργία νέου Project

Τώρα θα δημιουργήσουμε το πρώτο μας Project. Για να τα καταφέρουμε πρέπει να εκτελέσουμε τα παρακάτω βήματα.

Για την δημιουργία επιλέγουμε **File**→**New** και στο παράθυρο που μας ανοίγει δίνουμε το όνομα που θα έχει το project μας.

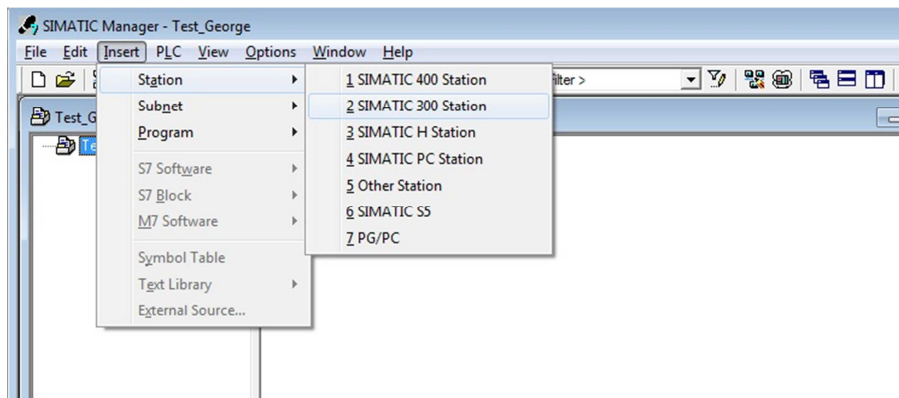


Εικόνα 29 : Νέο Project

Το «Storage Location (path)» μας δείχνει την διαδρομή αποθήκευσης των αρχείων project που δημιουργούμε. Μπορούμε να το αλλάξουμε από το Options→Customize αν και δεν θα το συνιστούσαμε. Πατώντας OK αποθηκεύουμε το project στο προκαθορισμένο path και είμαστε έτοιμοι να οργανώσουμε το Station.

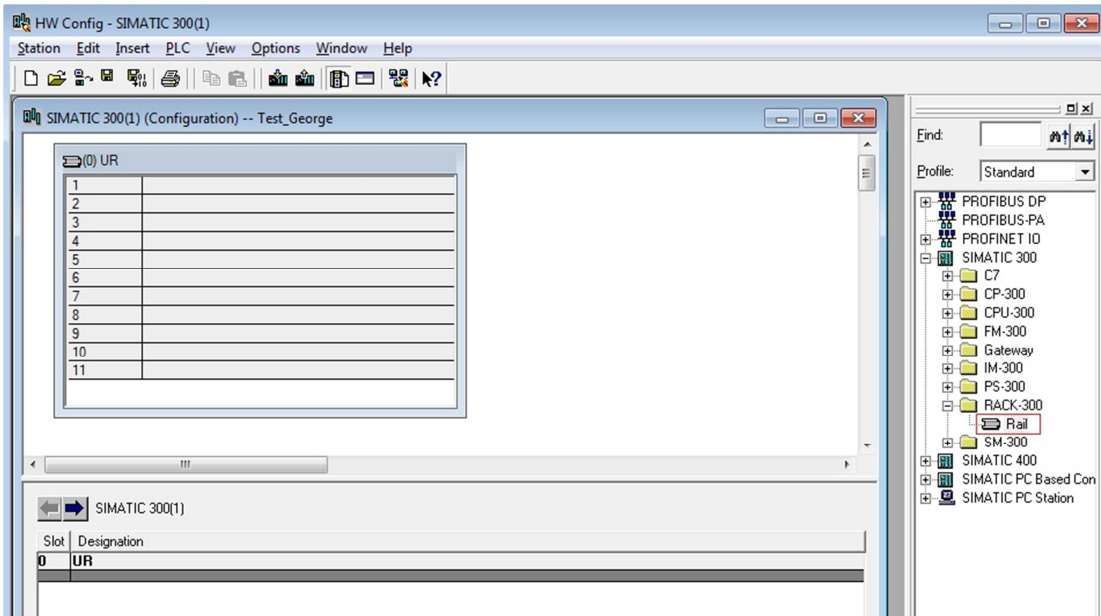
4.4 Διαμόρφωση του σταθμού-Hardware configuration

Για να εισάγουμε σταθμό κάνουμε Insert→Station→Siamatic 300 Station.



Εικόνα 30 :Δημιουργία Station

Κάνοντας double click στο hardware που εμφανίζεται στην διπλανή καρτέλα εμφανίζεται η καρτέλα διαμόρφωσης του σταθμού και η βιβλιοθήκη για την προσθήκη των Modules.

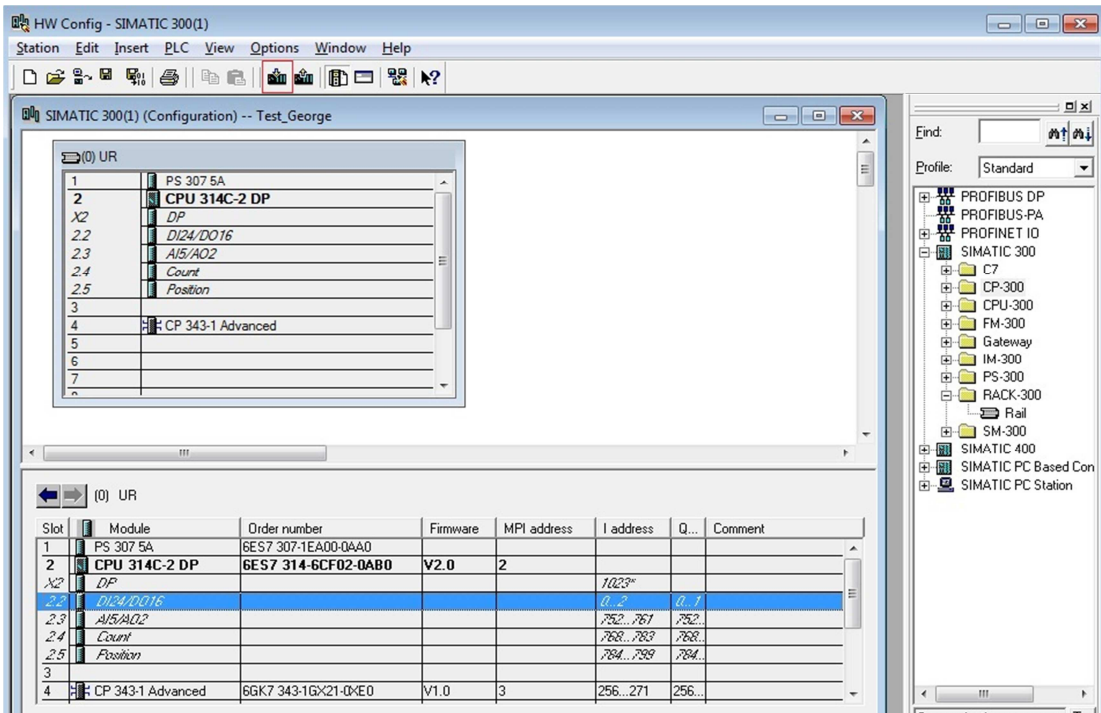


Εικόνα 31 : Τοποθέτηση Rail

Αρχικά εισάγουμε μια Rail όπου και θα στηρίζονται τα Modules(Simatic 300→Rack 300→Rail).

- Στην πρώτη θέση του Rail Βάζουμε το τροφοδοτικό (PS-300→PS 307 5A).
- Στην Δεύτερη θέση την Cpu (CPU-300→Cpu 314-2DP→6ES7 314-6CF02-0AB0).
- Η Τρίτη θέση είναι δεσμευμένη για IM (σε περίπτωση που επεκτεινόμαστε και σε άλλα Rack).Οπότε εάν έχουμε επέκταση του σταθμού μας τοποθετούμε την κατάλληλη IM κάρτα , αλλιώς αφήνουμε κενή την θέση.
- Στην τέταρτη θέση έχουμε το CP (CP-300→Ind.Ethernet→343-1 advanced-IT) το οποίο είναι το module που αναλαμβάνει την επικοινωνία μέσω Ethernet.

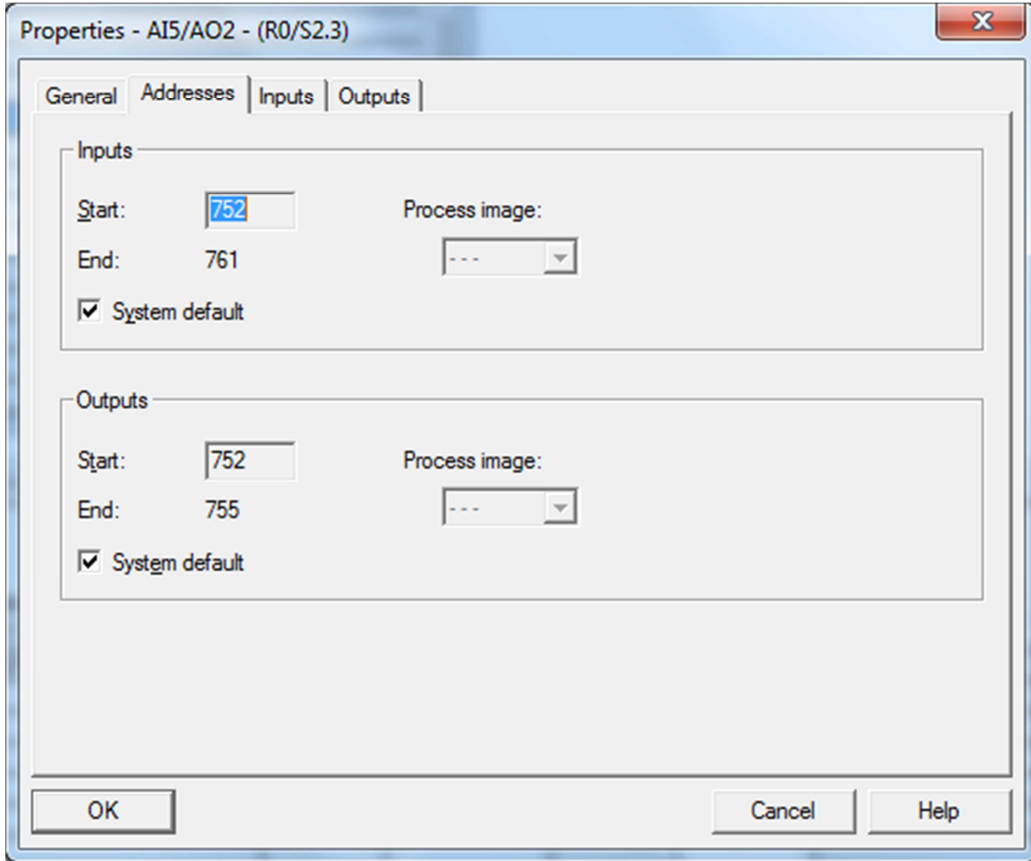
Κάνοντας τις παραπάνω ρυθμίσεις έχουμε τον παρακάτω σταθμό :



Εικόνα 32 : Διαχείριση Hardware

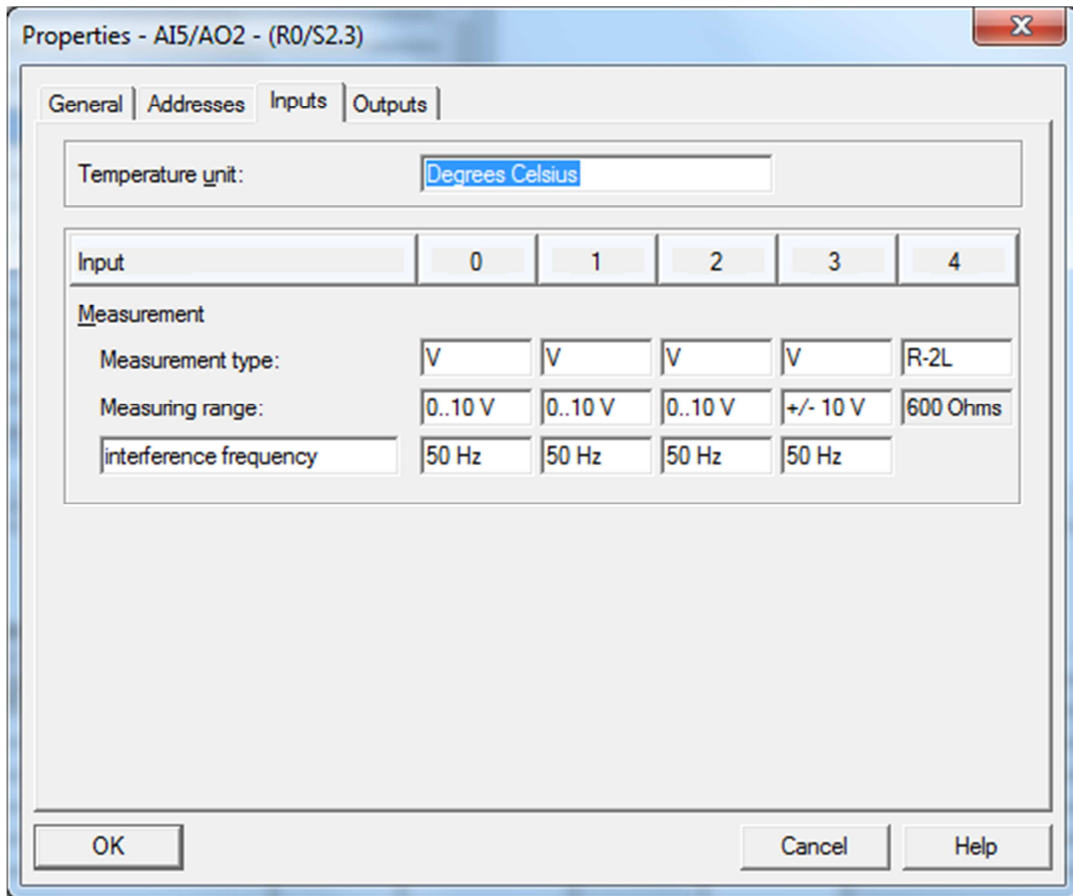
Μεταβολή διευθύνσεων I/O καρτών:

Εδώ θα δούμε πως θα μεταβάλουμε τις διευθύνσεις των εισόδων και των εξόδων. Κάνοντας αριστερό κλικ σε μια μονάδα εισόδων μας ανοίγει ένα παράθυρο στο οποίο επιλέγουμε την καρτέλα address που είναι χωρισμένη σε input/output. Αλλάζοντας την τιμή εκκίνησης των εισόδων ή των εξόδων αυτόματα ενημερώνεται και η τελική τιμή που θα έχει σύμφωνα με τον σταθμό που έχουμε δημιουργήσει παραπάνω. Πατάμε OK και αφού κάνουμε Save ο σταθμός βλέπει τις νέες διευθύνσεις.



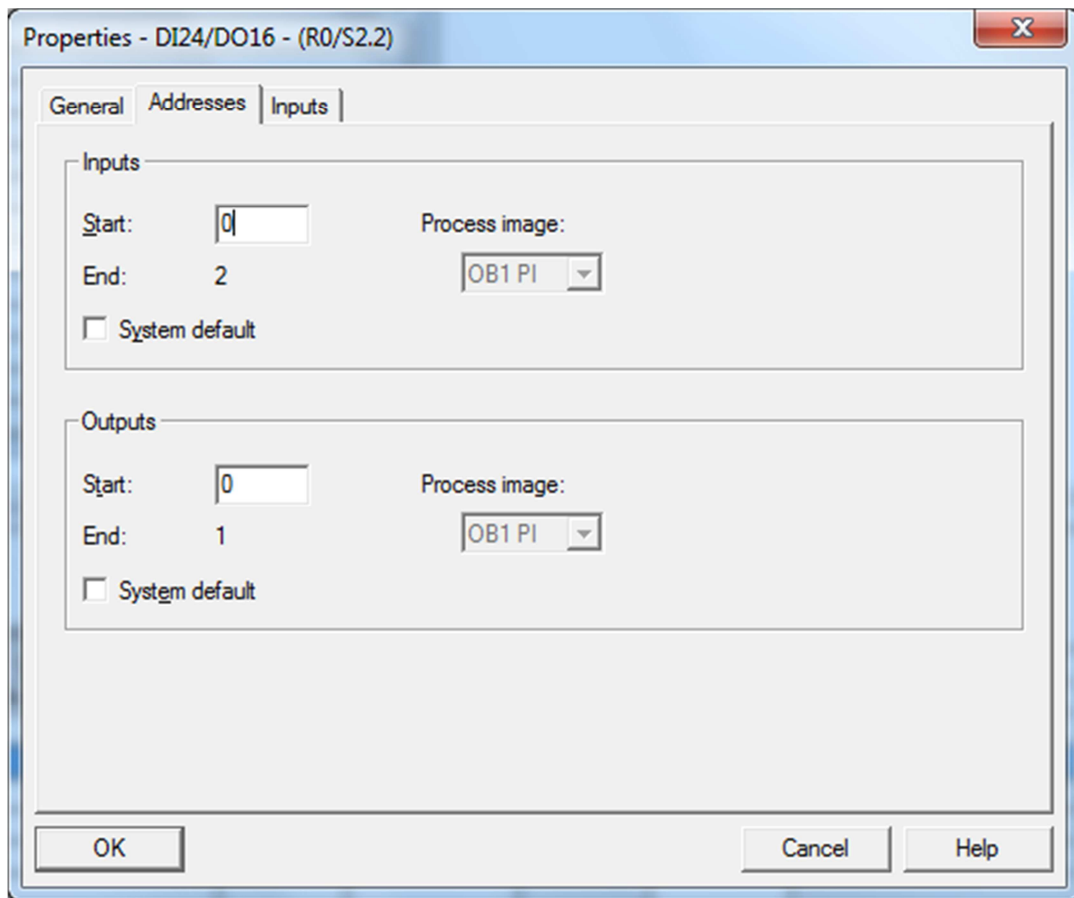
Εικόνα 33 : Αλλαγή διευθύνσεων αναλογικών καρτών

Στην παρακάτω καρτέλα μεταβάλλουμε τις εισόδους της αναλογικής κάρτας. Επειδή θέλουμε να μετρήσουμε θερμοκρασία ορίζουμε ως μονάδα μέτρησης Degrees Celsius Και κατά την μέτρηση ορίζουμε V(volt) και η κλίμακα 0...10V.



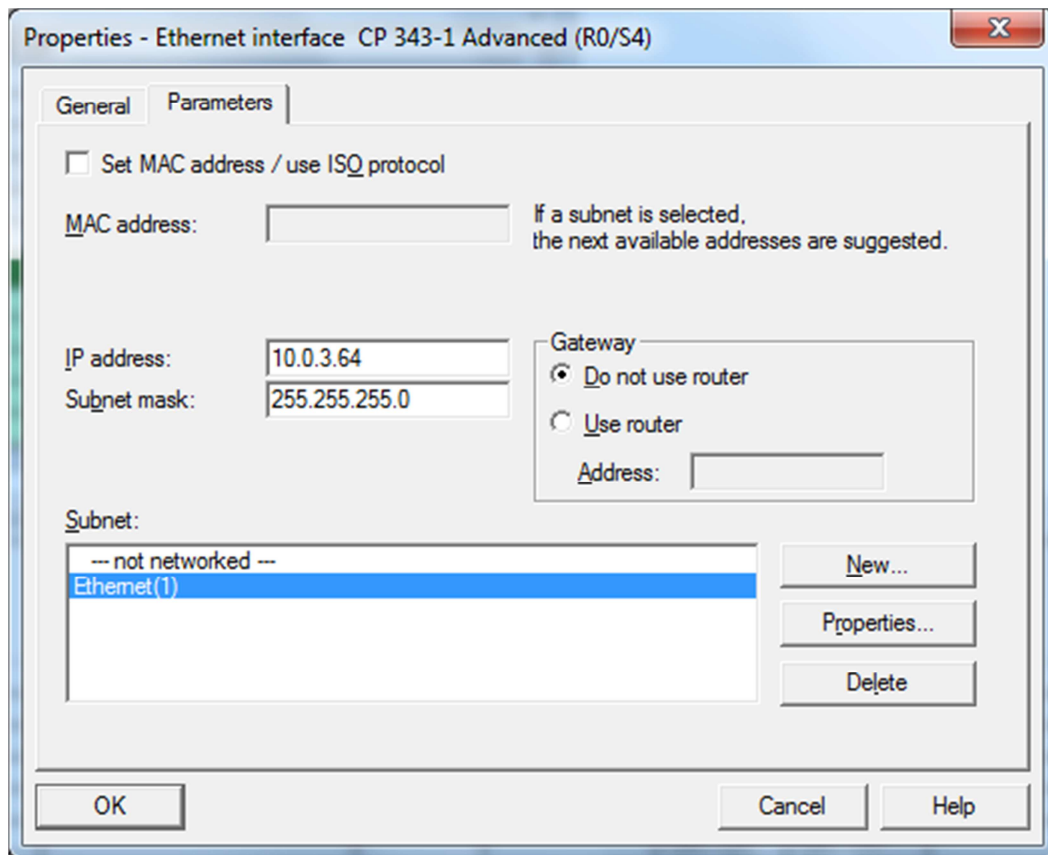
Εικόνα 34 : Ορισμός τρόπου μέτρησης

Στην επόμενη καρτέλα φαίνεται η αλλαγή των διευθύνσεων στις ψηφιακές εισόδους. Αυτό γίνεται γιατί επιλέξαμε να φτιάξουμε το πρόγραμμα μας με αρχικοποίηση των εισόδων και των εξόδων θέτοντας το 0.



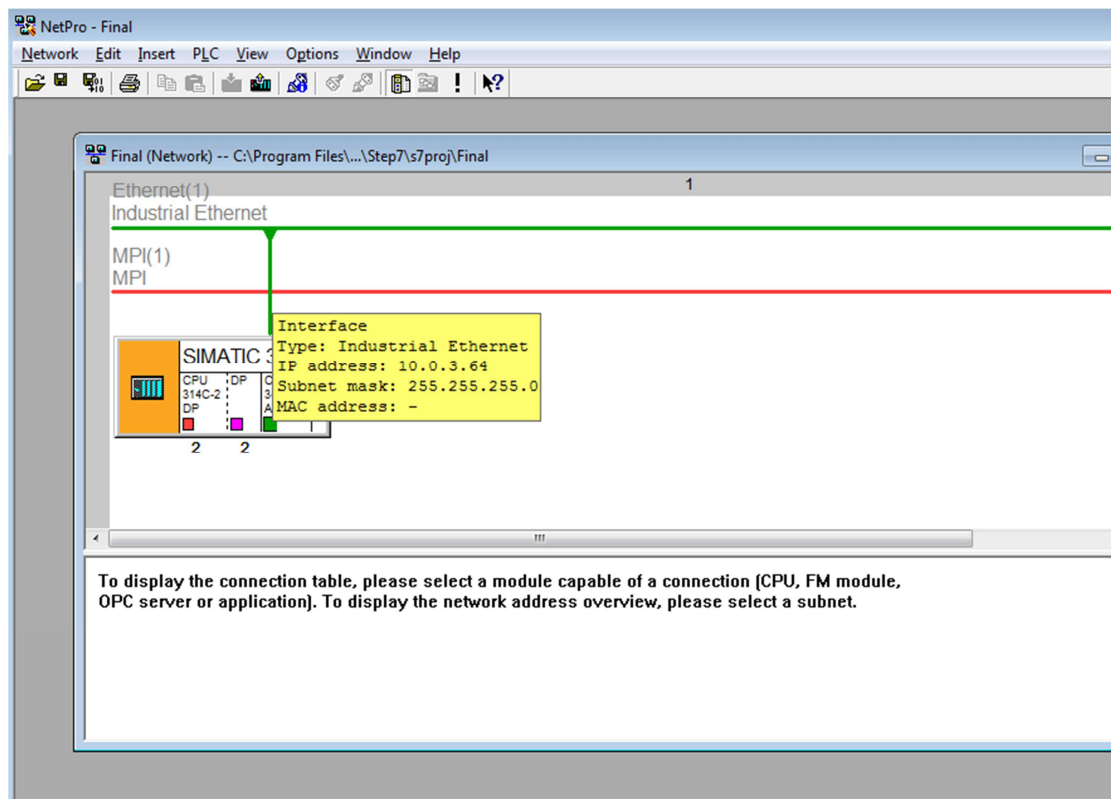
Εικόνα 35 : Αλλαγή διευθύνσεων ψηφιακής κάρτας

Βρισκόμαστε στην επιλογή properties του 343-1 advanced και αυτό για να δικτυώσουμε το industrial Ethernet module. Πατώντας New θέτουμε την νέα IP:10.0.3.64 χωρίς να χρησιμοποιούμε το router.



Εικόνα 36 : Ανάθεση IP στο CP

Με το εργαλείο NetPro μπορούμε να ελέγξουμε την κατάσταση που βρίσκεται ο σταθμός μας και αν είναι συνδεδεμένος στο δίκτυο.



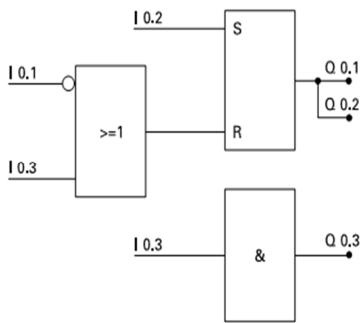
Εικόνα 37 : NetPro Tool

Φόρτωση προγράμματος στο PLC :

Αφού ολοκληρώσουμε τις ρυθμίσεις κάνουμε download πατώντας το κουμπί που έχουμε σημειώσει με κόκκινο χρώμα στην παραπάνω εικόνα. Αν σε κάποια θέση του Rail έχουμε δηλώσει κάποιο module που δεν υπάρχει τότε το simatic το θεωρεί ως λάθος και σε αυτή την περίπτωση έχουμε SF (System Failure). Προτείνουμε πριν το download να κάνουμε Check station consistency ώστε να αποφύγουμε διάφορα σφάλματα.

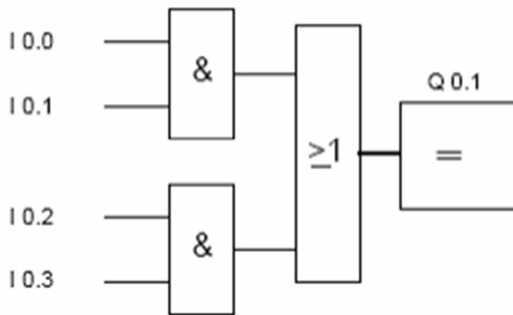
4.5 Βασικές πύλες και οργάνωση των blocks

Παρακάτω μπορούμε να δούμε ένα πρόγραμμα σε γλώσσα FBD και STL



L	I	0.2
S	Q	0.1
LN	I	0.1
O	I	0.3
R	Q	0.1
L	Q	0.1
=	Q	0.2
A	I	0.3
=	Q	0.3

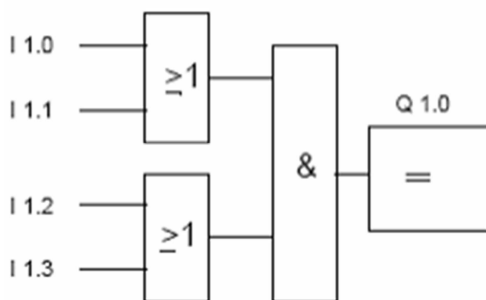
FBD



STL

```
A I 0.0
A I 0.1
O
A I 0.2
A I 0.3
= Q 0.1
```

FBD



STL

```
A(
O I 1.0
O I 1.1
)
A(
O I 1.2
O I 1.3
)
= Q 1.0
```

Εικόνα 38 : Παραδείγματα FBD/STL

Βλέπουμε στον παρακάτω πίνακα τα Block που θα συναντήσουμε και που θα μας χρησιμεύσουν για να κάνουμε τον απαραίτητο έλεγχο στις μεταβολές των μεταβλητών και να καταχωρούμε το αποτέλεσμα για περαιτέρω επεξεργασία.

	INT	DINT	REAL
ΙΣΟ ==			
ΔΙΑΦΟΡΕΤΙΚΟ <>			
ΜΕΓΑΛΥΤΕΡΟ >			
ΜΕΓΑΛΥΤΕΡΟ ή ΙΣΟ >=			
ΜΙΚΡΟΤΕΡΟ <			
ΜΙΚΡΟΤΕΡΟ ή ΙΣΟ <=			

Εικόνα 39 : Block σύγκρισης

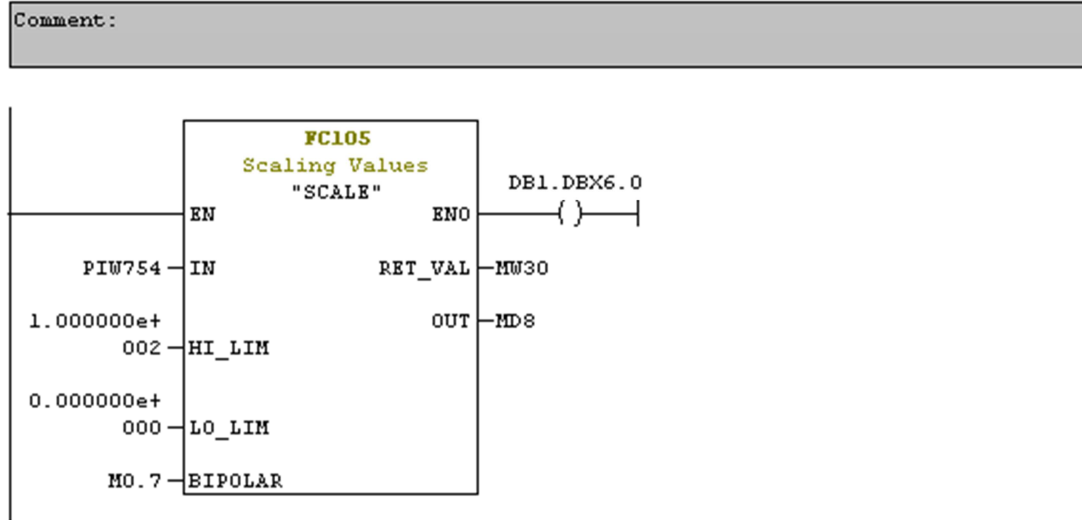
Είναι απαραίτητο να ξέρουμε όλους τους τύπους των μεταβλητών και το εύρος των διευθύνσεων για να ολοκληρώσουμε αποτελεσματικά ένα πρόγραμμα. Η μελέτη έγινε από τον παρακάτω πίνακα.

IEC	SIMATIC	Description	Data Type	Address Range
I	E	Input bit	BOOL	0.0 to 65535.7
IB	EB	Input byte	BYTE, CHAR	0 to 65535
IW	EW	Input word	WORD, INT, S5TIME, DATE	0 to 65534
ID	ED	Input double word	DWORD, DINT, REAL, TOD, TIME	0 to 65532
Q	A	Output bit	BOOL	0.0 to 65535.7
QB	AB	Output byte	BYTE, CHAR	0 to 65535
QW	AW	Output word	WORD, INT, S5TIME, DATE	0 to 65534
QD	AD	Output double word	DWORD, DINT, REAL, TOD, TIME	0 to 65532
M	M	Memory bit	BOOL	0.0 to 65535.7
MB	MB	Memory byte	BYTE, CHAR	0 to 65535
MW	MW	Memory word	WORD, INT, S5TIME, DATE	0 to 65534
MD	MD	Memory double word	DWORD, DINT, REAL, TOD, TIME	0 to 65532
PIB	PEB	Peripheral input byte	BYTE, CHAR	0 to 65535
PQB	PAB	Peripheral output byte	BYTE, CHAR	0 to 65535
PIW	PEW	Peripheral input word	WORD, INT, S5TIME, DATE	0 to 65534
PQW	PAW	Peripheral output word	WORD, INT, S5TIME, DATE	0 to 65534
PID	PED	Peripheral input double word	DWORD, DINT, REAL, TOD, TIME	0 to 65532
PQD	PAD	Peripheral output double word	DWORD, DINT, REAL, TOD, TIME	0 to 65532
T	T	Timer	TIMER	0 to 65535
C	Z	Counter	COUNTER	0 to 65535
FB	FB	Function block	FB	0 to 65535
OB	OB	Organization block	OB	1 to 65535
DB	DB	Data block	DB, FB, SFB, UDT	1 to 65535
FC	FC	Function	FC	0 to 65535
SFB	SFB	System function block	SFB	0 to 65535
SFC	SFC	System function	SFC	0 to 65535
VAT	VAT	Variable table		0 to 65535
UDT	UDT	User-defined data type	UDT	0 to 65535

Εικόνα 40 : Τύποι μεταβλητών

4.6 Ανάπτυξη των OB1 , FC , DB

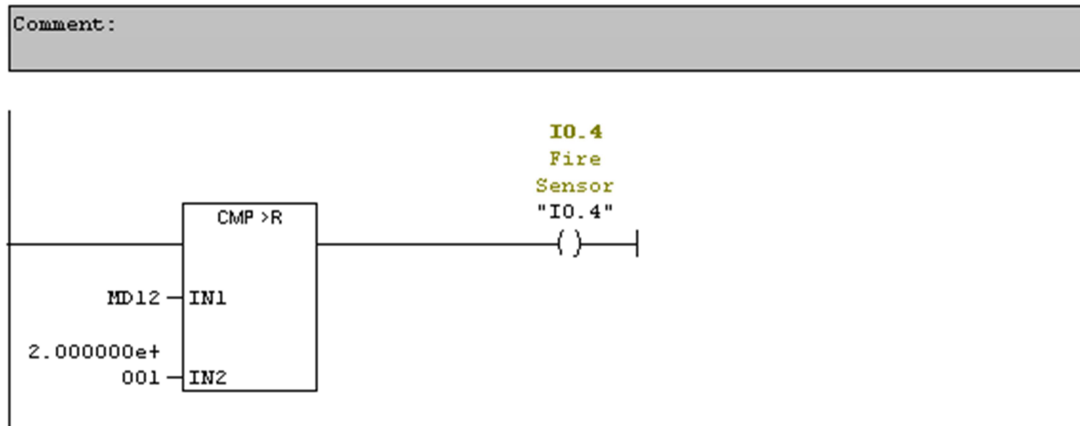
Network 2: Scaling the temp with FC 105



Εικόνα 41 : Scaling function

Χρησιμοποιώντας την ειδική συνάρτηση FC105 μπορούμε να κάνουμε scaling στην τιμή που παίρνουμε από την αναλογική μας είσοδο PIW 754 και να την μετατρέψουμε σε θερμοκρασία. Θέτουμε τα όρια HI_LIM και LO_LIM. Τοποθετούμε την τιμή στο datablock και σε μια θέση μνήμης και την επιστρέφουμε ως WORD

Network 1: Checking Current Temp In Room 1

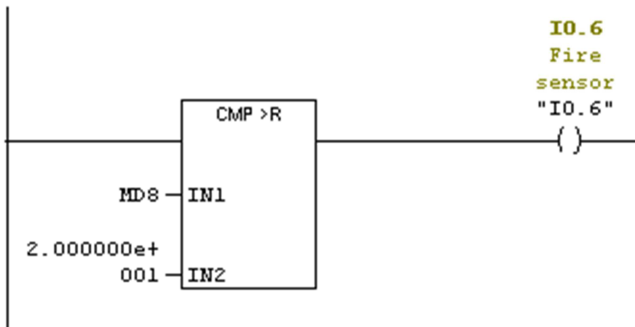


Εικόνα 42 : Compar Function

Με το παραπάνω block compare greater than συγκρίνουμε την τιμή που παίρνουμε από την μία αναλογική είσοδο η οποία εκχωρείται σε μια θέση μνήμης με μία τιμή που έχουμε προκαθορίσει εμείς Π.χ 20°C. Οπότε όταν η τιμή μας ξεπεράσει την προκαθορισμένη τιμή θα ενεργοποιηθεί η είσοδος I 0.4 η οποία θεωρείται αισθητήρας θερμοκρασίας.

Network 3: Checking Current Temp In Room 2

Comment:

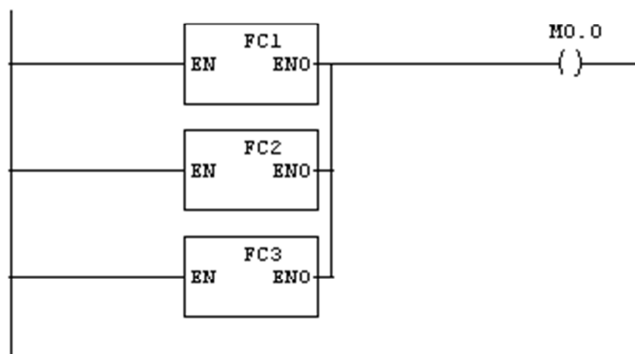


Εικόνα 43 : Compare Function

Κάνουμε την ίδια δουλειά με παραπάνω συγκρίνοντας την προκαθορισμένη τιμή με αυτή της άλλης αναλογικής εισόδου και ενεργοποιούμε την είσοδο I 0.6.

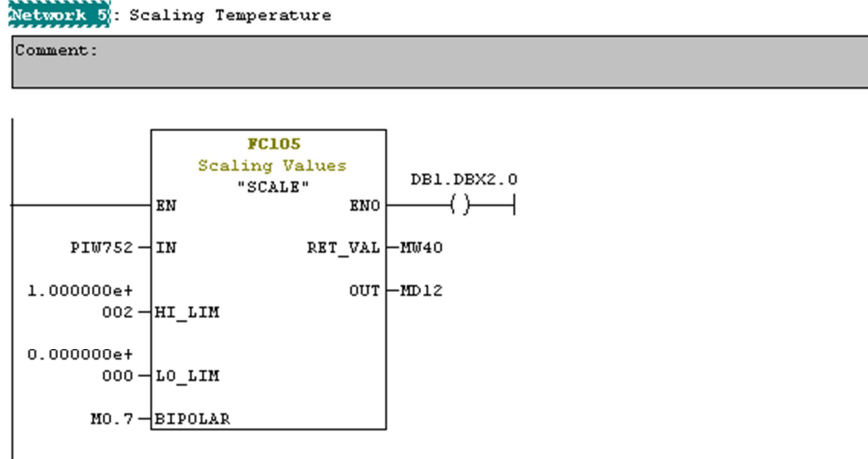
Network 4: Running Functions From Building

Comment:



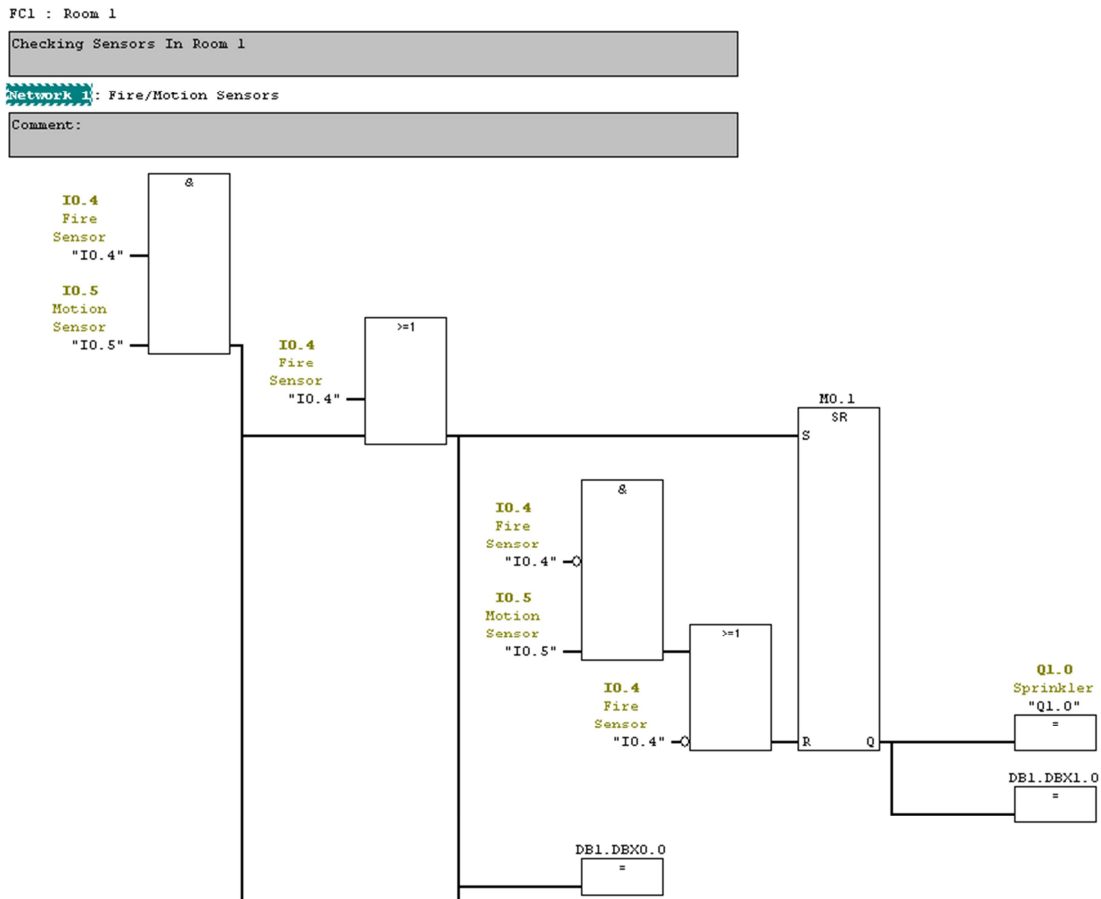
Εικόνα 44 : Συναρτήσεις στο OB

Στο παραπάνω network καλούμε όλες τις συναρτήσεις που έχουμε φτιάξει οι οποίες έχουν να κάνουν με την διαχείριση του προγράμματος μας.



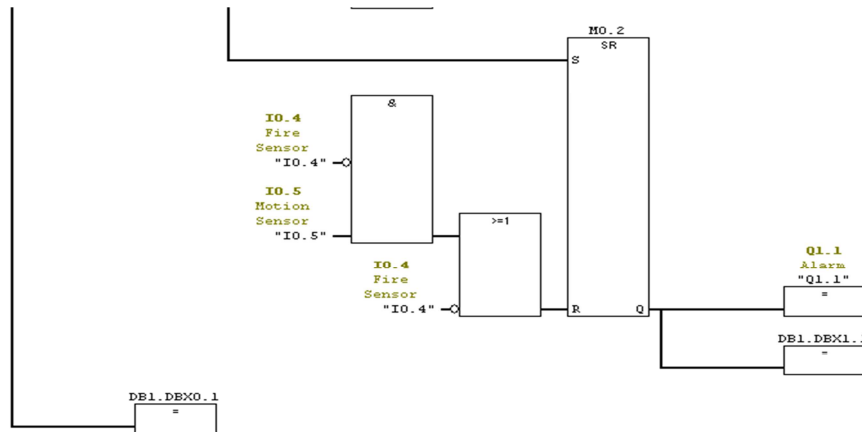
Εικόνα 45 : Scaling Function

Για ακόμη μια φορά κάνουμε Scale την άλλη αναλογική μας τιμή. Έτσι καταφέρνουμε να έχουμε δύο διαφορετικές τιμές σε διαφορετικές θέσεις μνήμης και σε διαφορετικό σημείο στο databock.



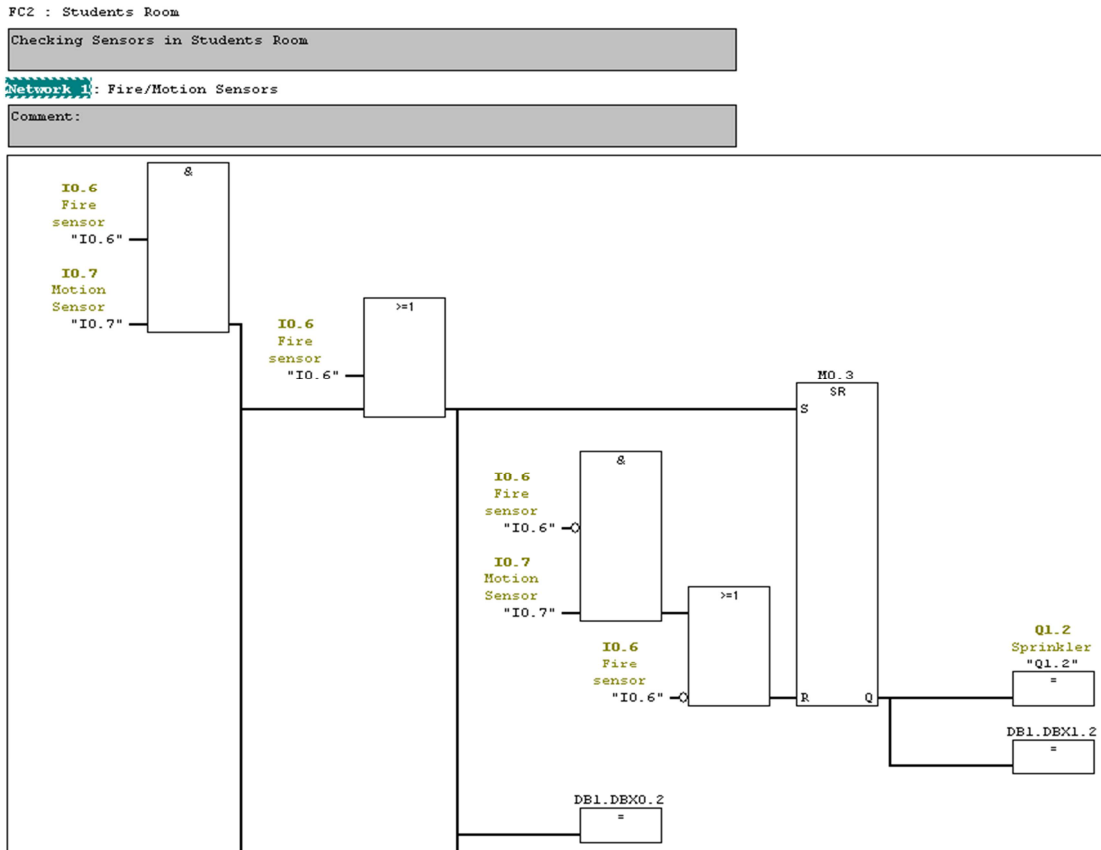
Εικόνα 46 : Δημιουργία FC1

Στην συνάρτηση FC1 ελέγχουμε για πρόβλημα πυρκαγιάς στον πρώτο χώρο. Εάν IO.4 (fire sensor) και IO.5(motion sensor) ενεργοποιηθούν είτε μόνο η IO.4 τότε ενεργοποιείτε η Q1.0 (Sprinkler) και το reset στο flip flop γίνεται μόνο αν απενεργοποιηθεί η IO.4.



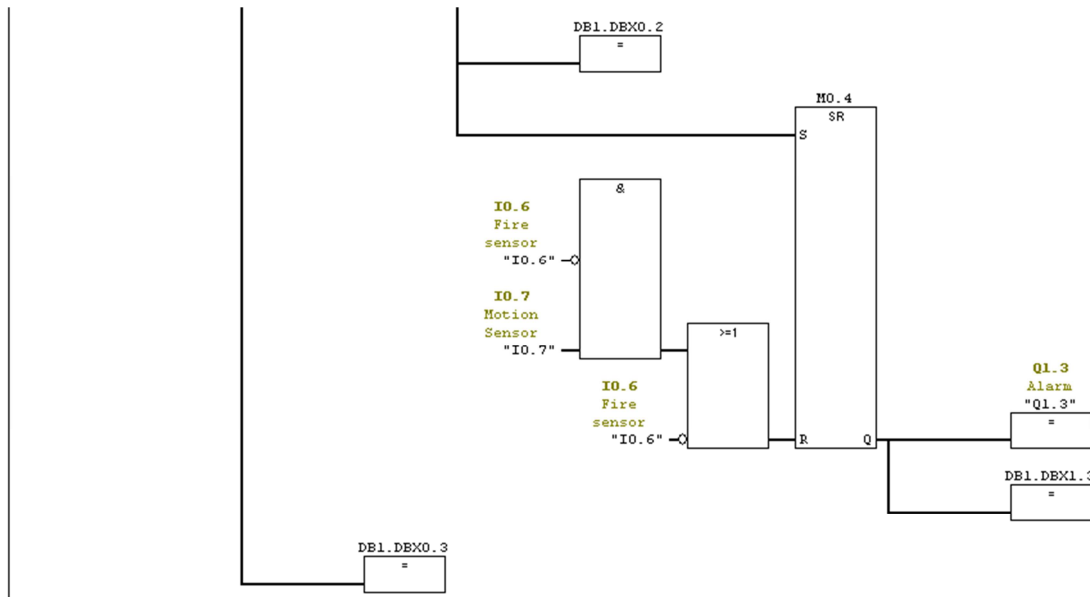
Εικόνα 47 : FC1 η συνέχεια

Επίσης ενεργοποιείτε και η Q1.1(Alarm) η οποία απενεργοποιείτε με τον ίδιο τρόπο. Οι μεταβλητές εκχωρούνται στο datablock για περαιτέρω επεξεργασία.



Εικόνα 48 : Δημιουργία FC2

Στην συνάρτηση FC2 ελέγχουμε για πρόβλημα πυρκαγιάς στον δεύτερο χώρο. Εάν IO.6 (fire sensor) και IO.7 (motion sensor) ενεργοποιηθούν είτε μόνο η IO.6 τότε ενεργοποιείτε η Q1.2 (Sprinkler) και το reset στο flip flop γίνεται μόνο αν απενεργοποιηθεί η IO.6.



Εικόνα 49 : FC2 η συνέχεια

Επίσης ενεργοποιείτε και η Q1.3(Alarm) η οποία απενεργοποιείτε με τον ίδιο τρόπο. Οι μεταβλητές εκχωρούνται στο datablock για περαιτέρω επεξεργασία.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	I04	BOOL	FALSE	FIRE SENSOR AS DIGITAL
+0.1	I05	BOOL	FALSE	MOTION SENSOR
+0.2	I06	BOOL	FALSE	FIRE SENSOR
+0.3	I07	BOOL	FALSE	MOTION SENSOR
+0.4	I10	BOOL	FALSE	FIRE SENSOR
+0.5	I11	BOOL	FALSE	MOTION SENSOR
+0.6	I12	BOOL	FALSE	FIRE SENSOR
+0.7	I13	BOOL	FALSE	MOTION SENSOR
+1.0	Q10	BOOL	FALSE	SPRINKLER
+1.1	Q11	BOOL	FALSE	ALARM
+1.2	Q12	BOOL	FALSE	SPRINKLER
+1.3	Q13	BOOL	FALSE	ALARM
+1.4	Q04	BOOL	FALSE	SPRINKLER
+1.5	Q15	BOOL	FALSE	ALARM
+1.6	Q16	BOOL	FALSE	SPRINKLER
+1.7	Q17	BOOL	FALSE	ALARM
+2.0	PIW752	REAL	0.000000e+000	LM35
+6.0	piw754	REAL	0.000000e+000	LM35
=10.0		END_STRUCT		

Εικόνα 50 : Περιεχόμενα DataBlock

Στο Datablock Βλέπουμε όλες τις μεταβλητές που έχουμε εκχωρήσει. Όταν η μεταβλητή ενεργοποιηθεί τότε και στο datablock θα γίνει True.

ΚΕΦΑΛΑΙΟ 5

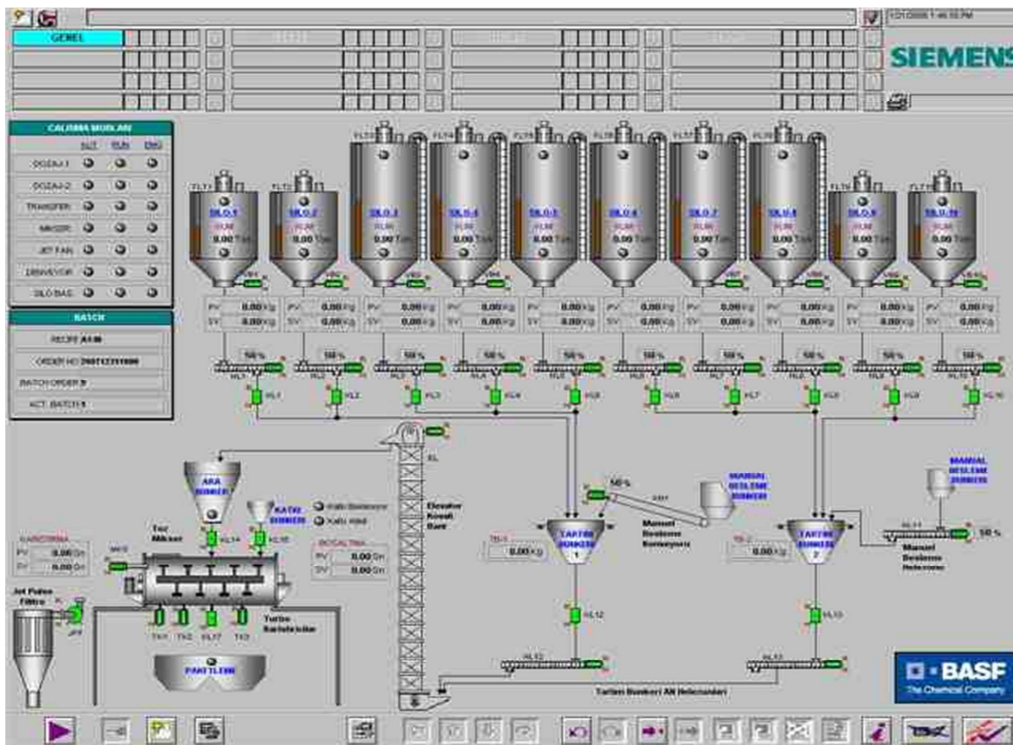
ΣΥΣΤΗΜΑΤΑ SCADA

5.1 Τι είναι Scada

Ο όρος scada είναι ακρωνύμιο του όρου Supervisory Control And Data Acquisition που σημαίνει εποπτικός έλεγχος και συλλογή δεδομένων. Ένα τέτοιο σύστημα επιτρέπει στο χειριστή να εποπτεύει και να ελέγχει διεργασίες οι οποίες βρίσκονται κατακευματισμένες μεταξύ διαφόρων απομακρυσμένων περιοχών. Όπως φαίνεται και από την ονομασία του δεν είναι ένα πλήρες σύστημα ελέγχου αλλά εστιάζει κυρίως στην εποπτεία.

Η διαδικασία λειτουργίας ενός τέτοιου συστήματος είναι η συλλογή των πληροφοριών, η αποστολή τους σε ένα κεντρικό σημείο επεξεργασίας, η εκτέλεση της απαραίτητης ανάλυσης και ελέγχου και τέλος η παρουσίαση των αποτελεσμάτων σε διάφορες οθόνες χειρισμού και εποπτείας. Ένα σύστημα scada επιτρέπει στους χειριστές του να παρατηρούν διαδικασίες με μεγάλη τοπολογική διανομή από μια κεντρική τοποθεσία.

Τα συστήματα HMI (Human Machine Interface) αποτελούν το μέρος λειτουργίας των Scada που αλληλεπιδρά με τον τελικό χρήστη. Συνήθως είναι μια οπτική απεικόνιση της διεργασίας πάνω στην οποία εμφανίζονται τιμές μεταβλητών, καταστάσεις και διαγράμματα. Αυτά τα συστήματα επιτρέπουν την κατ' απαίτηση εμφάνιση ιστορικών των δεδομένων. Οι πληροφορίες αυτές αντλούνται από την βάση δεδομένων του scada οπότε scada και HMI είναι αλληλένδετα περισσότεροι κατασκευαστές scada δίνουν την δυνατότητα ενσωμάτωσης του συστήματος τους σε HMI. Το θέμα είναι ότι μια HMI εφαρμογή μπορεί να κατασκευαστεί ανεξάρτητα από τον πυρήνα καταγραφής ενός Scada ώστε να αντλεί πληροφορίες και να κάνει τον απαραίτητο έλεγχο.



Εικόνα 51 : Σύστημα Scada

5.2 Πλεονεκτήματα Scada

Τα οφέλη που προκύπτουν από τον έλεγχο του βιομηχανικού οργανισμού στην γενικότερη μορφή του είναι :

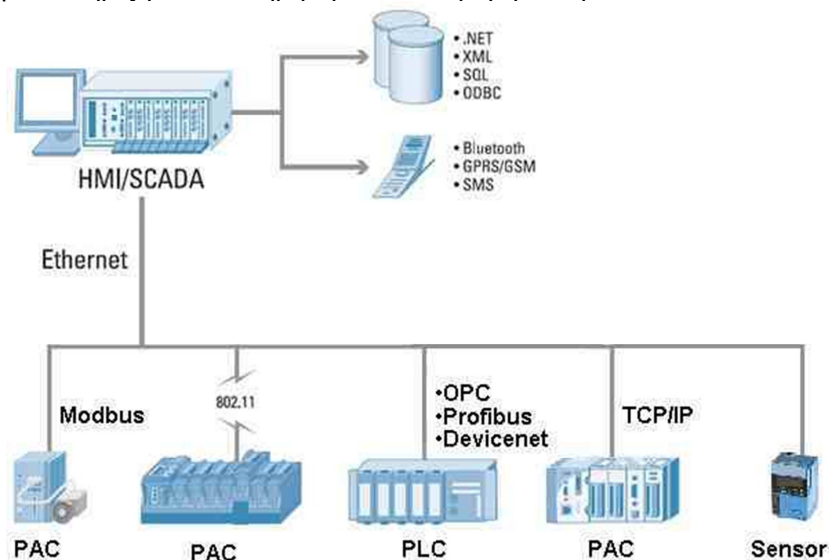
- Αύξηση της παραγωγής λόγω της καλύτερης αξιοποίησης των δυνατοτήτων των μέσων παραγωγής.
- Μείωση του κόστους παραγωγής ανά μονάδα παραγόμενου προϊόντος, λόγω βέλτιστης χρήσης των εσωτερικών πηγών ενέργειας και μείωση του κόστους εργασίας.
- Βελτίωση της ποιότητας των παραγόμενων προϊόντων λόγω της δυνατότητας να διατηρούνται οι συνθήκες λειτουργίας μέσα σε στενά όρια ανοχών.
- Ευελιξία παραγωγής κάτω από συνεχώς μεταβαλλόμενες συνθήκες αγοράς.

Ειδικότερα τα συστήματα Scada προσφέρουν :

- Παρακολούθηση της διαδικασίας παραγωγής μέσω της οποίας επιτυγχάνεται η μέγιστη απόδοση της βιομηχανικής μονάδας.
- Αύξηση της επικοινωνίας μεταξύ των επιπέδων της βιομηχανικής μονάδας και κυρίως της διοίκησης και της παραγωγής.
- Δυνατότητα στο προσωπικό να λαμβάνει αποφάσεις μετά από πληρέστερη ενημέρωση.
- Γρηγορότερος εντοπισμός και αντιμετώπιση σφαλμάτων.
- Βελτίωση των συνθηκών ασφαλείας.
- Πιο εύστοχες και έγκαιρες πληροφορίες για την διοίκηση.

Μερικά πρακτικά παραδείγματα του οφέλους των Scada είναι :

- Οι χειριστές μπορούν να χρησιμοποιούν οθόνες PC με λειτουργικό windows για να εποπτεύουν τον εξοπλισμό μέσω GUI (Graphic user Interface).
- Οι ενημερώσεις κρίσιμων καταστάσεων μπορούν να γίνονται με ηχογραφημένα μηνύματα τα οποία θα εκπέμπονται αυτόματα.
- Δυνατότητα για αποθήκευση και ανάκτηση ιστορικών δεδομένων για περαιτέρω συγκρίσεις ή συμπεράσματα.
- Τεχνική υποστήριξη και συντήρηση από τον προμηθευτή του



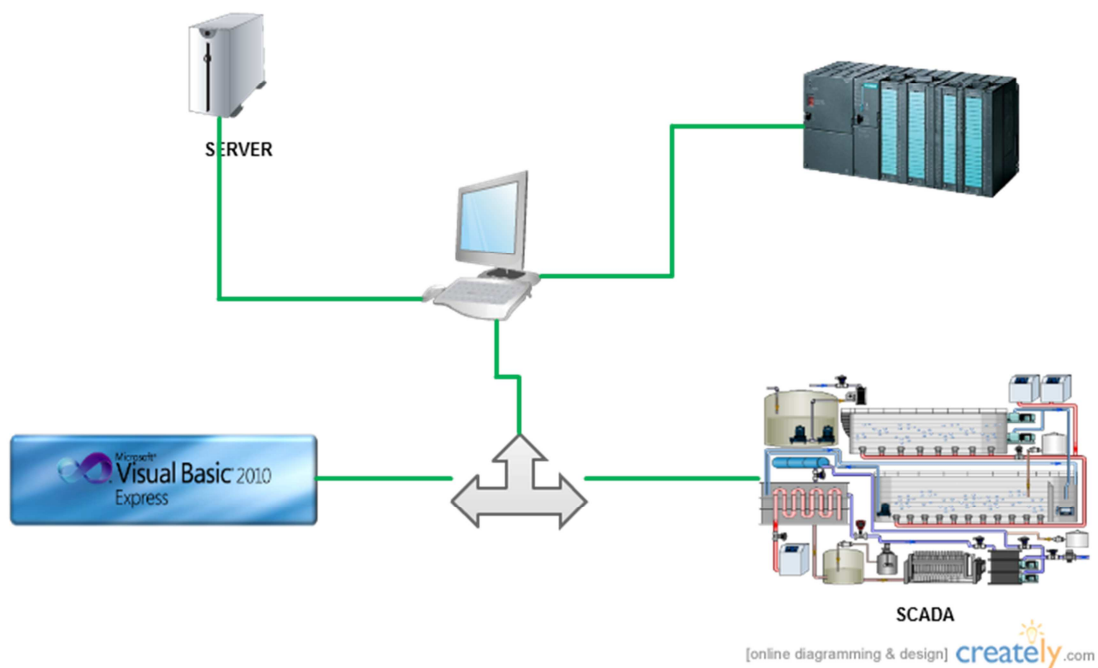
Εικόνα 52 : Ολοκληρωμένο Scada

5.3 Σύντομη Περιγραφή του υλοποιηθέντος συστήματος

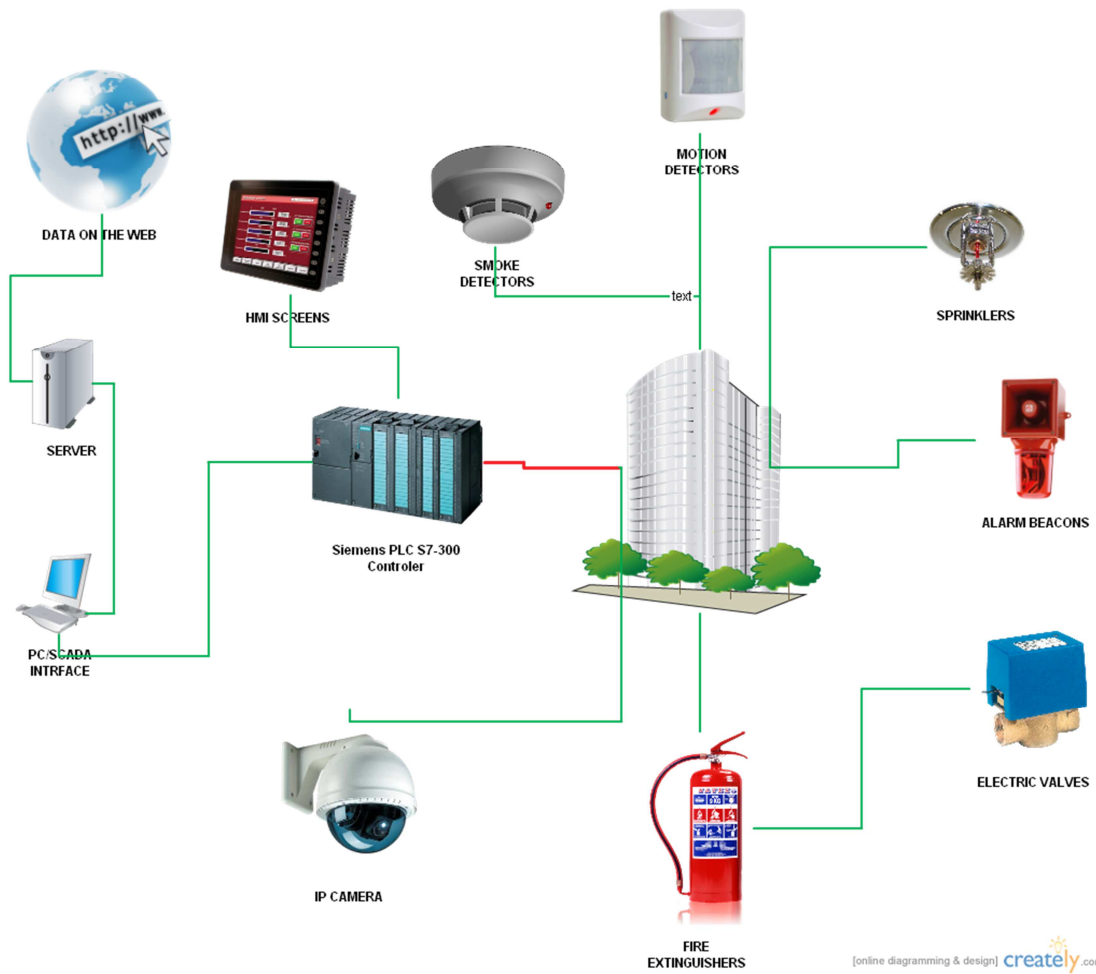
Όπως προαναφέραμε και στην εισαγωγή ο σχεδιασμός του Scada έγινε μέσω της Visual Basic 2010 ενώ χρησιμοποιήσαμε και βάση δεδομένων MySql για την καταγραφή και ανάκτηση δεδομένων.

Η λειτουργία της εφαρμογής διαιρείται σε δυο βασικές ενότητες, τον σχεδιασμό και την εποπτεία. Κατά τον σχεδιασμό εισάγουμε στο σύστημα τα στοιχεία που είναι απαραίτητα για την συλλογή και παρουσίαση των δεδομένων . Αυτά τα στοιχεία είναι :

- Τομείς/Διεργασίες : Αφορά τα τμήματα που πρέπει να χωρίσουμε τον αυτοματισμό μας για καλύτερη απόδοση.
- Μεταβλητές: Είναι το κυριότερο κομμάτι της αυτοματοποιημένης διαδικασίας . Αποτελούν απεικόνιση των μεταβλητών της κατάστασης του εξοπλισμού. Σύμφωνα με τις τιμές των μεταβλητών αυτών ο χρήστης μπορεί να ενημερώνεται για τυχόν ανεπιθύμητες διακυμάνσεις .
- Θύρες επικοινωνίας : Εδώ ορίζουμε πως θα επικοινωνεί ο απομακρυσμένος εξοπλισμός με το σύστημα μας. Στην Περίπτωση μας με σύνδεση TCP/IP.
- Εντολές Ελέγχου : Εντολές που μπορεί να στείλει το σύστημα αυτόματα στα τερματικά είτε μετά από απαίτηση του χρήστη.
- Σύνδεση με DB : Σύνδεση με βάση δεδομένων για καταγραφή συμβάντων.



Εικόνα 53 : Απλή πρόταση Scada



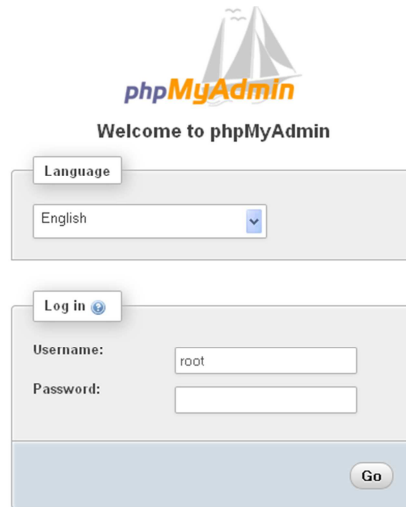
Εικόνα 54 : Η ολοκληρωμένη πρόταση μας για Scada

ΚΕΦΑΛΑΙΟ 6

ΑΝΑΛΥΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΛΟΓΙΣΜΙΚΟΥ

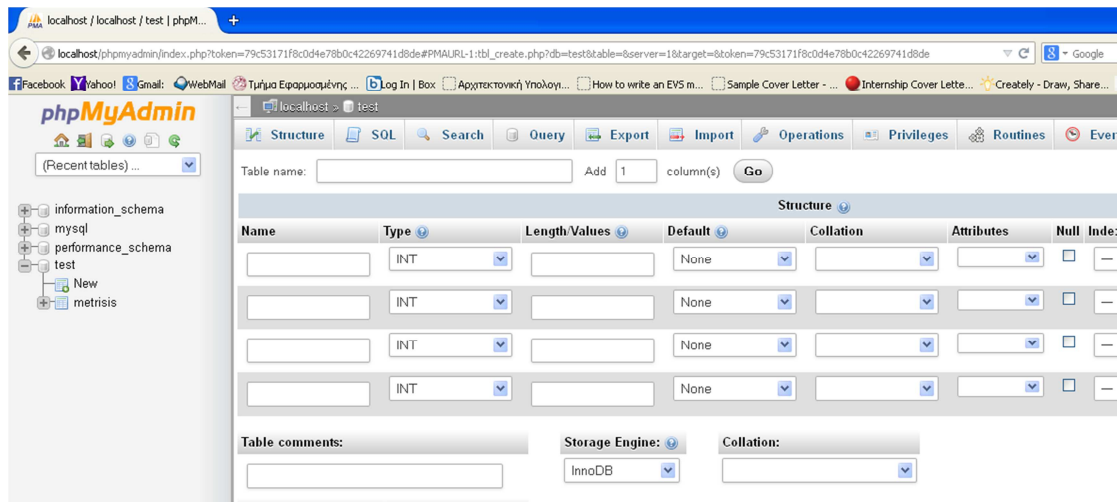
6.1 Βάση Δεδομένων Mysql

Για την αποθήκευση των δεδομένων και για την ανάκτηση τους κατά κύριο μέρος στο πρόγραμμα και κατ' επέκταση στην ιστοσελίδα που θα αναρτηθεί χρησιμοποιήσαμε την MySQL μέσω phpMyAdmin. Θα δούμε παρακάτω τα βασικά βήματα και την ανάπτυξη της βάσης μας.



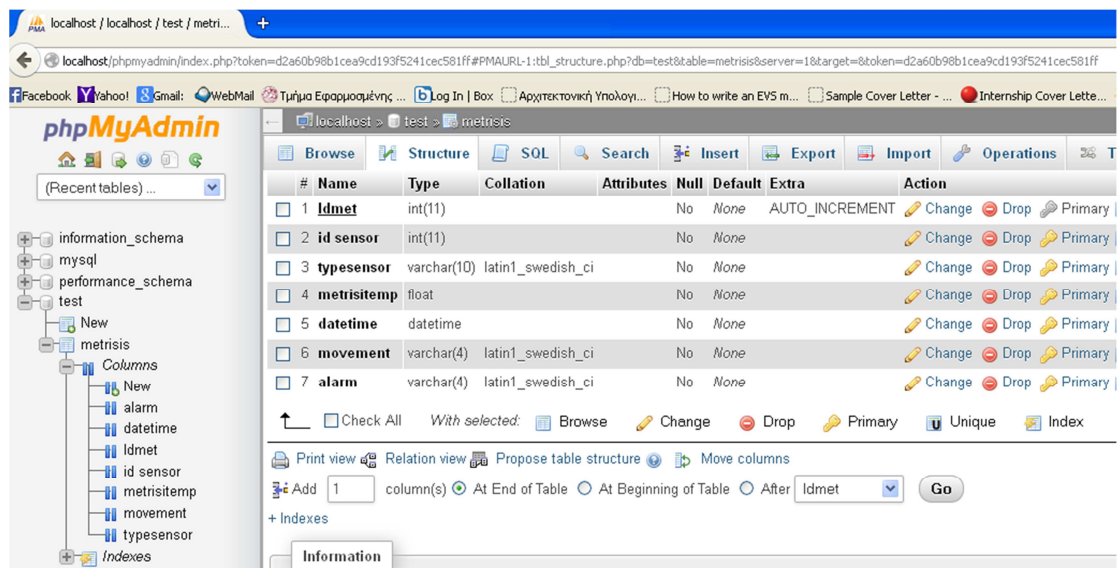
Εικόνα 55 : Έναρξη phpMyAdmin

1. Αρχικά κρατήσαμε το default username και χωρίς password για πιο εύκολη πρόσβαση
2. Πατώντας το GO και κάνοντας new table έχουμε την παρακάτω εικόνα :



Εικόνα 56 : Φτιάχνουμε το Table

3. Συμπληρώνουμε τις στήλες και έχουμε την παρακάτω πίνακα :



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	Idmet	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary
2	id sensor	int(11)			No	None		Change Drop Primary
3	typesensor	varchar(10)	latin1_swedish_ci		No	None		Change Drop Primary
4	metrisitemp	float			No	None		Change Drop Primary
5	datetime	datetime			No	None		Change Drop Primary
6	movement	varchar(4)	latin1_swedish_ci		No	None		Change Drop Primary
7	alarm	varchar(4)	latin1_swedish_ci		No	None		Change Drop Primary

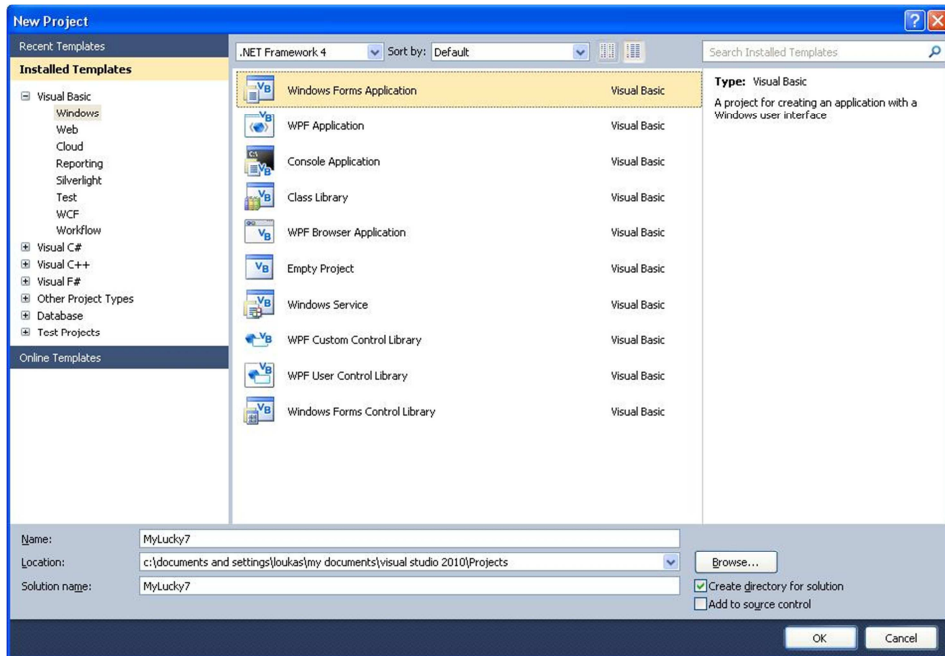
Εικόνα 57 : Η Βάση δεδομένων μας

Έχουμε τα παρακάτω στοιχεία :

- [1]. Idmet τύπου int και auto increment.
- [2]. Idsensor τύπου int .
- [3]. Typesensor τύπου varchar.
- [4]. Metrisitemp τύπου float.
- [5]. Datetime τύπου datetime.
- [6]. Movement τύπου varchar.
- [7]. Alarm τύπου varchar.

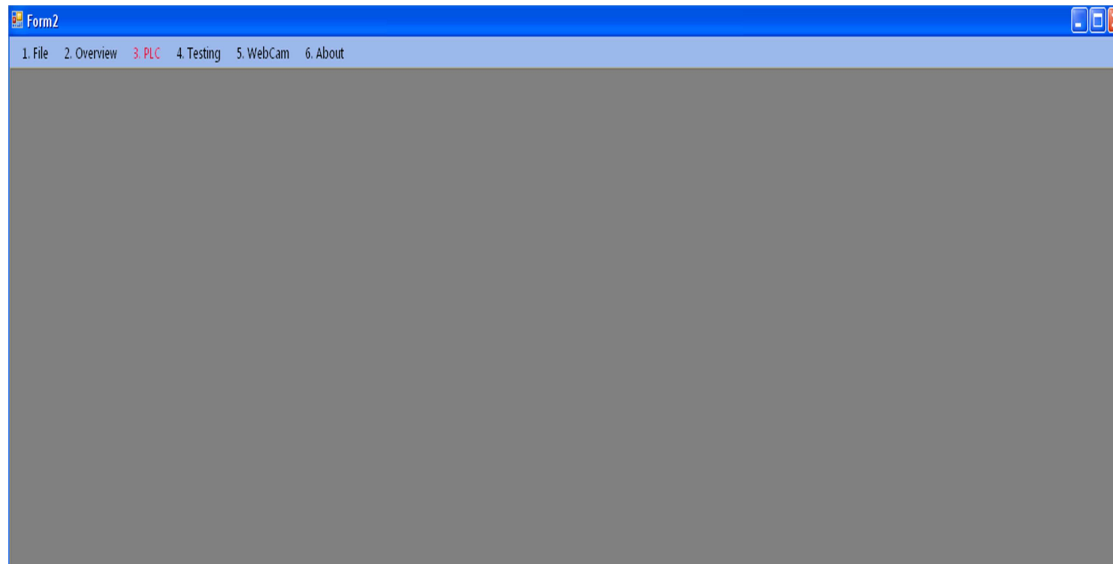
6.2 Ο προγραμματισμός στην Visual basic

1. Ξεκινάμε το Visual Studio 2010
2. Απο το μενού File του visual studio επιλέγουμε New Project
3. Στο πλαίσιο κειμένου Name πληκτρολογούμε LogginProgram2
4. Πατάμε OK και δημιουργούμε το έργο



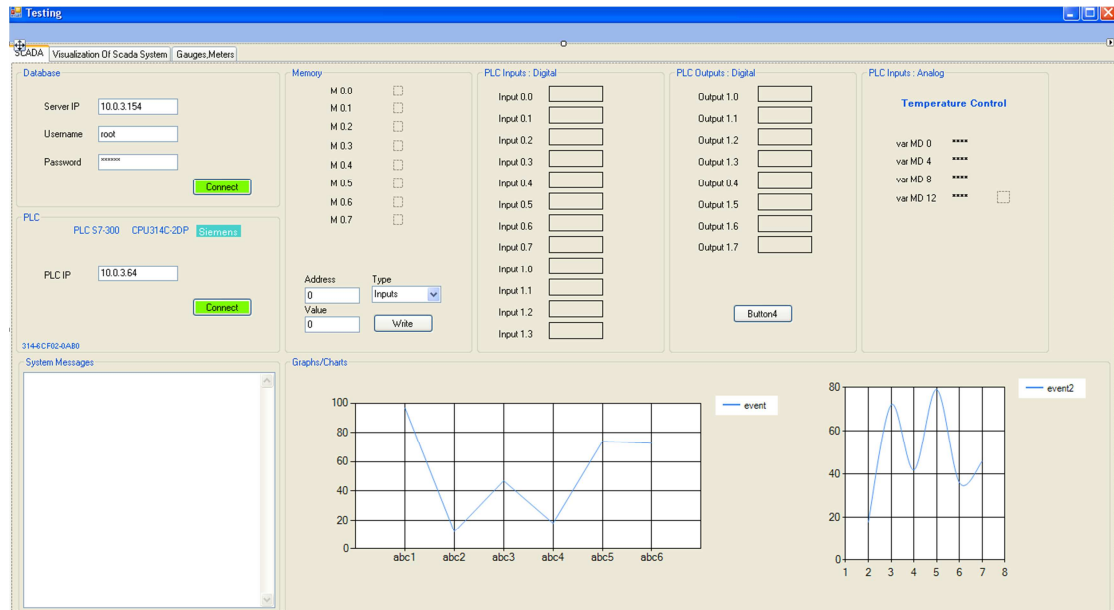
Εικόνα 58 : Δημιουργία του Project

5. Απο το ToolBox τοποθετούμε ένα MenuStrip (Το μενού που βρίσκεται στην κορυφή)
6. Κάνουμε την φόρμα μας MDIparent και έχουμε το παρακάτω :



Εικόνα 59 : MDI φόρμα/Κύριο Μέρος

Πατώντας το PLC Tab εμφανίζεται το παράθυρο που έχουμε δημιουργήσει από το μηδέν για επικοινωνία με το PLC. Είναι η πρώτη καρτέλα του Tab Control που έχουμε προσθέσει και έχουμε δώσει το όνομα SCADA

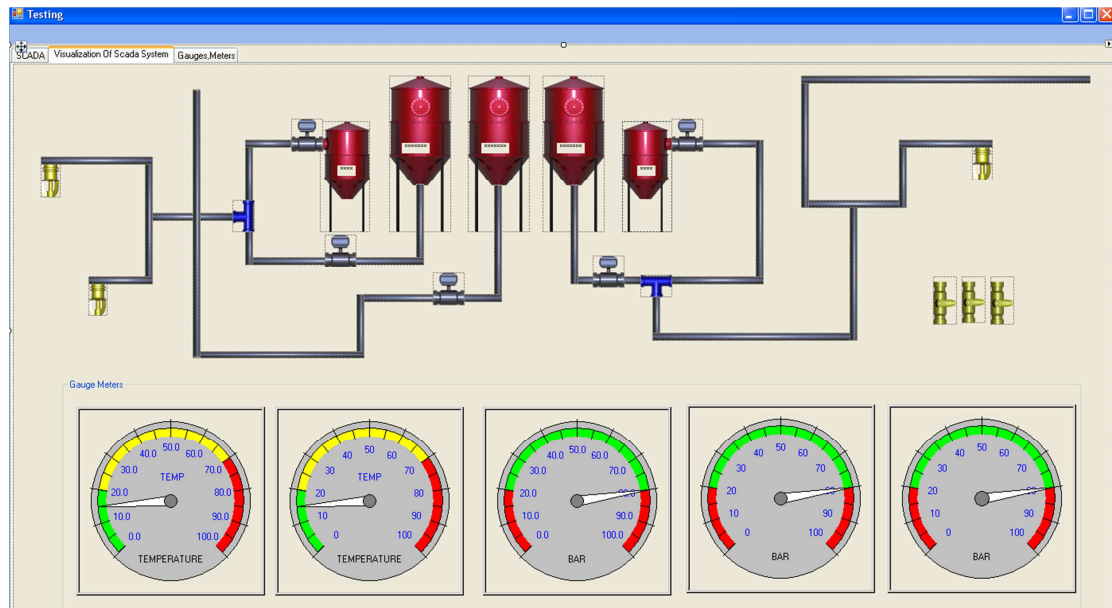


Εικόνα 60 : Scada Tab.Απεικόνιση Εισόδων/εξόδων/γραφήματα

Το παραπάνω παράθυρο περιέχει :

1. Σύνδεση με βάση MySql
2. Σύνδεση με το PLC
3. Panel για απεικόνιση των εισόδων, εξόδων, μνημών και θερμοκρασίας
4. Γραφικές παραστάσεις κατευθείαν από την βάση δεδομένων.
5. Textbox για την αναφορά του συστήματος όσο λειτουργεί με την μορφή μηνυμάτων.
6. Όλα είναι οργανωμένα σε workgroup ανάλογα την διεργασία.

Πατώντας το δεύτερο Tab με όνομα Visualization of scada system πηγαίνουμε στην παρακάτω φόρμα που περιέχει τα κατάλληλα σύμβολα για την απεικόνιση του γραφικού περιβάλλοντος του scada συστήματος μας.



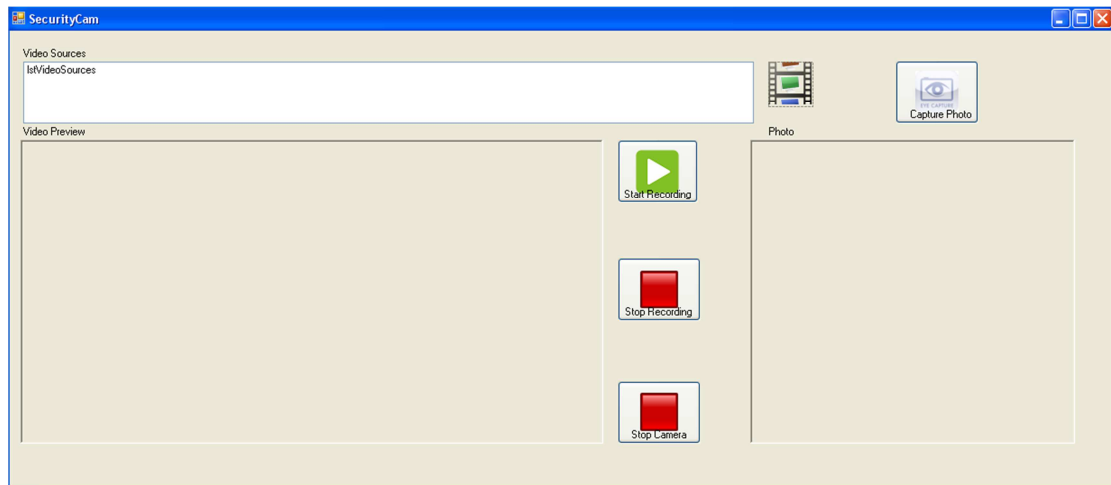
Εικόνα 61 : Γραφική απεικόνιση του Scada

Η παραπάνω φόρμα Περιέχει :

1. Βιομηχανικά σύμβολα
2. Panel για απεικόνιση θερμοκρασίας
3. Gauges για απεικόνιση θερμοκρασίας και στάθμης των Tank.

Τα δεδομένα που απεικονίζονται παραπάνω έχουν άμεση συσχέτιση με την προηγούμενη φόρμα αφού και αυτά αντλούνται από το PLC .Επίσης οι παραπάνω ενέργειες καταγράφονται στην βάση δεδομένων.

Στο κεντρικό μενού, στην MDI φόρμα, πατώντας στην επιλογή 5.webcam έχουμε δημιουργήσει μια κάμερα για την επίβλεψη του χώρου όποτε θέλουμε.



Εικόνα 62 : Webcam για παρακολούθηση χώρου

Η φόρμα περιέχει :

1. Λίστα για την προσθήκη πολλών καμερών
2. Κουμπιά εγγραφής
3. Απεικόνιση της εικόνας
4. Δυνατότητα για φωτογραφία

6.3 Κώδικας

Ξεκινώντας την ανάλυση του κώδικα μας αρχίζουμε από την Form2 που είναι η βασική μας φόρμα MDI και όλες οι υπόλοιπες φόρμες θα εμφανίζονται μέσα σε αυτή. Αρχικοποιούμε τις μεταβλητές μας που σχετίζονται με την επικοινωνία του προγράμματος μας με το PLC. Αυτό που μας ενδιαφέρει περισσότερο είναι η IP που θα δώσουμε αφού η σύνδεση μας γίνεται μέσω TCP/IP πρωτόκολλο. Παρακάτω έχουμε την εμφάνιση και την απόκριση των φορμών.

```
Public Class Form2

    Public fds As libnodave.daveO5serialType
    Public di As libnodave.daveInterface
    Public dc As libnodave.daveConnection
    Public res As Integer
    Public buf(100) As Byte
    Public localMPI As Integer = 0
    Public rack As Integer = 0
    Public slot As Integer = 2
    Public plcMPI As Integer = 2
    Public Connection As Boolean = False
    Public IP As String = "192.168.1.100"

    Private Sub WebCamToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles WebCamToolStripMenuItem.Click
        SecurityCam.Show()
        SecurityCam.MdiParent = Me
        Form3.Hide()
        Testing.Hide()

    End Sub

    Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ExitToolStripMenuItem.Click
        System.Windows.Forms.Application.Exit()

    End Sub

    Private Sub ExitToolStripMenuItem1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ExitToolStripMenuItem1.Click
        SecurityCam.Hide()

    End Sub

    Private Sub HomeToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles HomeToolStripMenuItem.Click
        SecurityCam.Hide()
        Form3.Hide()
        Testing.Hide()

    End Sub

    Private Sub AboutToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles AboutToolStripMenuItem.Click
        AboutBox1.Show()

    End Sub

    Private Sub OverviewToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles OverviewToolStripMenuItem.Click
        Form3.Show()
        Form3.MdiParent = Me
        SecurityCam.Hide()
        Testing.Hide()

    End Sub

    Private Sub ExitToolStripMenuItem2_Click(ByVal sender As System.Object,
```

```

ByVal e As System.EventArgs) Handles ExitToolStripMenuItem2.Click
    Form3.Hide()
End Sub

Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
End Sub

Private Sub ConnecToPlcToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
End Sub

Private Sub DisconnectPlcToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    dc.disconnectPLC()
    ToolStripStatusLabel1.Text = "Disconnected."
End Sub

Private Sub PLCToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles PLCToolStripMenuItem.Click
    Testing.Show()
    Testing.MdiParent = Me
    SecurityCam.Hide()
    Form3.Hide()
End Sub
End Class

```

Στο επόμενο κομμάτι κώδικα που είναι και το πιο σημαντικό έχουμε την σύνδεση και την ανταλλαγή των δεδομένων. Μπορούμε με τον κώδικα που έχουμε συντάξει να διαβάσουμε είτε αναλογικές είτε ψηφιακές εισόδους και εξόδους. Ανάλογα με την μεταβολή των εισόδων/εξόδων αλλάζουμε την κατάσταση των Controls των Panels και των Labels. Αλλαγές έχουμε και στα Text boxes τα οποία μας αναφέρουν τα μηνύματα από την CPU.

Για την ανάγνωση των ψηφιακών εισόδων έχουμε συναρτήσεις που επικοινωνούν απευθείας με το PLC μόλις γίνει η σύνδεση και δεν εμπλεκόμαστε με DataBlocks. Μία από αυτές τις συναρτήσεις είναι η IsBitSet όπως φαίνεται παρακάτω.

Επίσης θα δούμε παρακάτω τον κώδικα για την σύνδεση στην βάση και την εισαγωγή των δεδομένων σε αυτή και την ανάκτηση τους για την εμφάνιση των γραφημάτων και οποιαδήποτε άλλη χρήση τους όπως ανάρτηση τους στο διαδίκτυο.

```

Imports MySql.Data.MySqlClient
Imports System.Data.SqlClient
Imports AxXGAUGELib.AxXGauge

Public Class Testing

    Private Sub Testing_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
End Sub

    Public Sub ConnectPLCToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ConnectPLCToolStripMenuItem.Click
        fds.rfd = libnodave.openSocket(102, IP)
        fds.wfd = fds.rfd
        If fds.rfd > 0 Then ' if step 1 is ok
            di = New libnodave.daveInterface(fds, "IF1", 0,
libnodave.daveProtoISOTCP, libnodave.daveSpeed187k)
            di.setTimeout(1000000)
            res = di.initAdapter
        End If
    End Sub
End Class

```



```

    If res = 0 Then      ' init Adapter is ok
        dc = New libnodave.daveConnection(di, 0, rack, slot)
        res = dc.connectPLC()
        If res = 0 Then
            Connection = True
            ToolStripStatusLabel1.Text = "Connected " + IP
            Timer1.Enabled = True
        End If
    End If
End Sub

Private Sub DisconnectPLCToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles DisconnectPLCToolStripMenuItem.Click
    dc.disconnectPLC()
    ToolStripStatusLabel1.Text = "Disconnected."
End Sub

Public Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    Dim a, c As Byte
    Dim data15 As String() = {"The fire sensor is ON"}
    Dim data16 As String() = {"The fire sensor is OFF"}
    If Connection Then
        res = dc.readBytes(libnodave.daveFlags, 0, 0, 16, buf)
        If res = 0 Then
            Data1.Text = Str(dc.getS32)
            Data2.Text = Str(dc.getS32)
            Data3.Text = Str(dc.getFloat)
            Data4.Text = Str(dc.getFloat)
            Label45.Text = Data4.Text

            AxXGauge1.Value = Data4.Text
            Panel29.BackColor = Color.LimeGreen
        Else
            ToolStripStatusLabel1.Text = "Read data. " +
libnodave.daveStrerror(res)
        End If

        res = dc.readBytes(libnodave.daveInputs, 0, 0, 1, buf)
        If res = 0 Then
            a = Str(dc.getU8)
            If IsBitSet(a, 0) Then
                Panel1.BackColor = Color.Lime
            Else
                Panel1.BackColor = Color.DarkGreen
            End If
            If IsBitSet(a, 1) Then
                Panel2.BackColor = Color.Lime
            Else
                Panel2.BackColor = Color.DarkGreen
            End If
            If IsBitSet(a, 2) Then
                Panel3.BackColor = Color.Lime
            Else
                Panel3.BackColor = Color.DarkGreen
            End If
            If IsBitSet(a, 3) Then
                Panel4.BackColor = Color.Lime
            Else
                Panel4.BackColor = Color.DarkGreen
            End If
        End If
    End If

```

```

        If IsBitSet(a, 4) Then
            Panel13.BackColor = Color.Lime

            PictureBox20.Image = My.Resources.Pump1_Green_Right
            TextBox7.Text &= Environment.NewLine & "1 Room 1 Fire
Sensor Is ON In 0.4 Out 1.0 Emergency"
            AxXGauge1.Value = Data4.Text
            AxXGauge2.Value = AxXGauge2.Value - 2
        Else
            PictureBox5.Image = My.Resources.Valve2_Yellow_Closed
            Panel13.BackColor = Color.DarkGreen
            PictureBox20.Image = My.Resources.Pump1_Yellow_Right

            TextBox7.Text &= Environment.NewLine & "1 No Action"
        End If

        If IsBitSet(a, 5) Then
            Panel14.BackColor = Color.Lime
            TextBox7.Text &= Environment.NewLine & "2 Room1 Motion
Is On In 0.5 Out 1.1 JustMovement"
        Else
            Panel14.BackColor = Color.DarkGreen
            TextBox7.Text &= Environment.NewLine & "2 No Action"
        End If

        If IsBitSet(a, 6) Then
            Panel15.BackColor = Color.Lime
            TextBox7.Text &= Environment.NewLine & "3 Room 2 Fire
Sensor Is ON In 0.6 Out 1.2 Emergency"
        Else
            Panel15.BackColor = Color.DarkGreen
            TextBox7.Text &= Environment.NewLine & "3 No Action"
            PictureBox6.Image = My.Resources.Valve2_Yellow_Closed
            PictureBox21.Image = My.Resources.Pump1_Yellow_Right
        End If

        If IsBitSet(a, 7) Then
            Panel16.BackColor = Color.Lime
            TextBox7.Text &= Environment.NewLine & "4 Room 2 Motion
Is ON In 0.7 Out 1.2 Just Movement"
        Else
            Panel16.BackColor = Color.DarkGreen
            TextBox7.Text &= Environment.NewLine & "4 No Action"
        End If

        ToolStripStatusLabel1.Text = "Read data. " +
libnodave.daveStrerror(res)
    End If
    res = dc.readBytes(libnodave.daveInputs, 0, 1, 1, buf)
    If res = 0 Then
        a = Str(dc.getU8)
        If IsBitSet(a, 0) Then
            Panel17.BackColor = Color.Lime
            TextBox7.Text &= Environment.NewLine & "5 Room 3 Fire
Sensor Is ON In 1.0 Out 0.4 Emergency"
        Else
            Panel17.BackColor = Color.DarkGreen
            TextBox7.Text &= Environment.NewLine & "5 No Action"
            PictureBox7.Image = My.Resources.Valve2_Yellow_Closed
            PictureBox22.Image = My.Resources.Pump1_Yellow_Right
        End If
    End If

```

```
        If IsBitSet(a, 1) Then
            Panel18.BackColor = Color.Lime
            TextBox7.Text &= Environment.NewLine & "6 Room 3 Motion
Is ON In 1.1 Out 1.2 Just Movement"
        Else
            Panel18.BackColor = Color.DarkGreen
            TextBox7.Text &= Environment.NewLine & "6 No Action"
        End If
        If IsBitSet(a, 2) Then
            Panel19.BackColor = Color.Lime
        Else
            Panel19.BackColor = Color.DarkGreen
        End If
        If IsBitSet(a, 3) Then
            Panel20.BackColor = Color.Lime
        Else
            Panel20.BackColor = Color.DarkGreen
        End If
    End If

    res = dc.readBytes(libnodave.daveOutputs, 0, 0, 1, buf)
    If res = 0 Then
        a = Str(dc.getS8)
        If IsBitSet(a, 4) Then
            Panel21.BackColor = Color.Lime
        Else
            Panel21.BackColor = Color.DarkGreen
        End If
    End If

    res = dc.readBytes(libnodave.daveOutputs, 0, 1, 1, buf)
    If res = 0 Then
        a = Str(dc.getU8)
        If IsBitSet(a, 0) Then
            Panel5.BackColor = Color.Lime
        Else
            Panel5.BackColor = Color.DarkGreen
        End If
        If IsBitSet(a, 1) Then
            Panel6.BackColor = Color.Lime
        Else
            Panel6.BackColor = Color.DarkGreen
        End If
        If IsBitSet(a, 2) Then
            Panel7.BackColor = Color.Lime
        Else
            Panel7.BackColor = Color.DarkGreen
        End If
        If IsBitSet(a, 3) Then
            Panel8.BackColor = Color.Lime
        Else
            Panel8.BackColor = Color.DarkGreen
        End If

        If IsBitSet(a, 5) Then
            Panel22.BackColor = Color.Lime
        Else
            Panel22.BackColor = Color.DarkGreen
        End If
        If IsBitSet(a, 6) Then
```

```
        Panel23.BackColor = Color.Lime
    Else
        Panel23.BackColor = Color.DarkGreen
    End If
    If IsBitSet(a, 7) Then
        Panel24.BackColor = Color.Lime
    Else
        Panel24.BackColor = Color.DarkGreen
    End If
Else
    ToolStripStatusLabel1.Text = "Read data. " +
libnodave.daveStrerror(res)
End If

res = dc.readBytes(libnodave.daveFlags, 0, 0, 1, buf)
If res = 0 Then
    c = Str(dc.getU8)
    If IsBitSet(c, 0) Then
        Panel9.BackColor = Color.Lime
    Else
        Panel9.BackColor = Color.DarkGreen
    End If
    If IsBitSet(c, 1) Then
        Panel10.BackColor = Color.Lime
    Else
        Panel10.BackColor = Color.DarkGreen
    End If
    If IsBitSet(c, 2) Then
        Panel11.BackColor = Color.Lime
    Else
        Panel11.BackColor = Color.DarkGreen
    End If
    If IsBitSet(c, 3) Then
        Panel12.BackColor = Color.Lime
    Else
        Panel12.BackColor = Color.DarkGreen
    End If

    If IsBitSet(c, 4) Then
        Panel25.BackColor = Color.Lime
    Else
        Panel25.BackColor = Color.DarkGreen
    End If
    If IsBitSet(c, 5) Then
        Panel26.BackColor = Color.Lime
    Else
        Panel26.BackColor = Color.DarkGreen
    End If
    If IsBitSet(c, 6) Then
        Panel27.BackColor = Color.Lime
    Else
        Panel27.BackColor = Color.DarkGreen
    End If
    If IsBitSet(c, 7) Then
        Panel28.BackColor = Color.Lime
    Else
        Panel28.BackColor = Color.DarkGreen
    End If
Else
    ToolStripStatusLabel1.Text = "Read data. " +
libnodave.daveStrerror(res)
End If
```

```
Else
    Data1.Text = "****"
    Data2.Text = "****"
    Data3.Text = "****"
    Data4.Text = "****"
    Panel1.BackColor = Color.Gray
    Panel2.BackColor = Color.Gray
    Panel3.BackColor = Color.Gray
    Panel4.BackColor = Color.Gray
    Panel5.BackColor = Color.Gray
    Panel6.BackColor = Color.Gray
    Panel7.BackColor = Color.Gray
    Panel8.BackColor = Color.Gray
    Panel9.BackColor = Color.Gray
    Panel10.BackColor = Color.Gray
    Panel11.BackColor = Color.Gray
    Panel12.BackColor = Color.Gray
End If
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim a As Integer
    a = Int(TextBox1.Text)
    buf(0) = Int(TextBox2.Text)
    If ComboBox1.SelectedItem = "Inputs" Then
        res = dc.writeBytes(libnodave.daveInputs, 0, a, 1, buf)
    End If
    If ComboBox1.SelectedItem = "Outputs" Then
        res = dc.writeBytes(libnodave.daveOutputs, 0, a, 1, buf)
    End If
    If ComboBox1.SelectedItem = "Flags" Then
        res = dc.writeBytes(libnodave.daveFlags, 0, a, 1, buf)
    End If
End Sub

Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ExitToolStripMenuItem.Click
    If Connection Then
        dc.disconnectPLC()
    End If
    Me.Close()
End Sub

Private Sub Panel5_Paint(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles Panel5.Paint

End Sub

Private Sub Panel13_Paint(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PaintEventArgs)
End Sub

Private Sub Label12_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label12.Click
End Sub

Private Sub Panel16_Paint(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles Panel16.Paint
End Sub
```

```
Private Sub Panel13_Paint_1(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles Panel13.Paint
End Sub

Private Sub Panel8_Paint(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles Panel8.Paint

End Sub

Private Sub ConnectionToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ConnectionToolStripMenuItem.Click
End Sub

Private Function SqlConnection() As MySqlConnection
Throw New NotImplementedException
End Function

Private Sub Button2_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
Dim serverstring As String = "Server=localhost;Database=test;User
Id=root;"
Dim sqlConnection As MySqlConnection = New MySqlConnection
Dim sql As MySqlCommand = New MySqlCommand("SELECT * FROM sensors limit
30 ", sqlConnection)
Dim sql2 As MySqlCommand
Dim rd As MySqlDataReader

Timer2.Enabled = True
sqlConnection.ConnectionString = serverstring
Dim strquery As String = ""

Try
If sqlConnection.State = ConnectionState.Closed Then

strquery = "insert into sensors
(NameSensors,FloorSensors,Alarmsiren,Alarmbeacon,Alarmlights,Typesensors,Date,T
ime,Temp)" & "values('ON',1,'ON','ON','ON','ON',curdate(),curtime(),45)"
sql2 = New MySqlCommand(strquery, sqlConnection)

sqlConnection.Open()

ToolStripStatusLabel1.Text = "Connected To Database"
sql2.ExecuteNonQuery()

rd = sql.ExecuteReader

While rd.Read

Me.Chart1.Series("event").Points.AddXY(rd.GetString("Date"),
rd.GetString("Temp"))

Me.Chart2.Series("event2").Points.AddXY(rd.GetString("Alarmsiren"),
rd.GetString("Typesensors"))

'Me.Chart3.Series("event3").Points.AddXY(rd.GetString("Typesensors"),
rd.GetString("Alarmbeacon"))
End While

Else
```

```

        'strquery = "insert into sensors
(NameSensors,FloorSensors,Alarmsiren,Alarmbeacon,Alarmlights,Typesensors,Date,T
ime)" & "values('ON',1,'ON','ON','ON','ON',curdate(),curtime())"
        'sql2 = New MySqlCommand(strquery, sqlConnection)
    End If

    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try

End Sub

Private Sub Label133_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label133.Click

End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    fds.rfd = libnodave.openSocket(102, IP)
    fds.wfd = fds.rfd
    If fds.rfd > 0 Then ' if step 1 is ok
        di = New libnodave.daveInterface(fds, "IF1", 0,
libnodave.daveProtoISOTCP, libnodave.daveSpeed187k)
        di.setTimeout(1000000) ' Make this longer if you have a very long
response time
        res = di.initAdapter
        '
        If res = 0 Then ' init Adapter is ok
            dc = New libnodave.daveConnection(di, 0, rack, slot) ' rack
amd slot don't matter in case of MPI
            res = dc.connectPLC()
            If res = 0 Then
                Connection = True

                ToolStripStatusLabel1.Text = "Connected " + IP
                Timer1.Enabled = True
            End If
        End If
    End If
End Sub

Private Sub TextBox7_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox7.TextChanged
    TextBox7.Select(TextBox7.TextLength, 0)
    TextBox7.ScrollToCaret()
End Sub

Private Sub MenuStrip2_ItemClicked(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.ToolStripItemClickedEventArgs) Handles
MenuStrip2.ItemClicked
End Sub

Private Sub ConnectToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs)
End Sub

Private Sub FileToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs)
End Sub

```

```
Private Sub ConnectPLCToolStripMenuItem1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs)
    End Sub

    Private Sub ConnectPLCToolStripMenuItem2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        End Sub

        Private Sub StatusStrip1_ItemClicked(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.ToolStripItemClickedEventArgs) Handles
StatusStrip1.ItemClicked
            End Sub

            Private Sub PictureBox9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox9.Click
                End Sub

                Public Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
                    End Sub

                    Private Sub Button4_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
                        Dim ADR As Integer
                        Dim Par As Boolean = True
                        Dim InputNum As Integer = 0
                        Dim BitNum As Integer = 3
                        If Par Then
                            buf(0) = 255 'write 1
                        Else
                            buf(0) = 0 'write 0
                        End If
                        If Connection Then
                            ADR = InputNum * 8 + BitNum

                            'write Input I20.3
                            'res = dc.writeBits(libnodave.daveInputs, 0, ADR, 1, buf)

                            'write Merke M20.3
                            'res = dc.writeBits(libnodave.daveFlags, 0, ADR, 1, buf)

                            'write output Q20.3
                            res = dc.writeBits(libnodave.daveOutputs, 1, ADR, 1, buf)
                        Else
                            MsgBox("No connection with PLC!")
                        End If

                    End Sub

                    Private Sub TabPage2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TabPage2.Click
                        End Sub

                        Private Sub PictureBox26_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox26.Click
                            End Sub

                            Private Sub PictureBox11_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
                                End Sub
```



```
Private Sub PictureBox19_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    End Sub

Private Function dr(ByVal p1 As String) As Object
    Throw New NotImplementedException
End Function

Private Sub Label153_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label153.Click
    End Sub

Private Sub Label156_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label156.Click
    End Sub

Private Sub Data1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Data1.Click
    End Sub
End Class
```

Παρακάτω έχουμε φτιάξει ένα module που μπορούμε να έχουμε εκεί όποιες μεταβλητές όσο αναφορά την επικοινωνία με το PLC και να μην επεμβαίνουμε για κάποια αλλαγή στο κυρίως πρόγραμμα.

```
Module Variables
    Public fds As libnodave.daveOSSerialType
    Public di As libnodave.daveInterface
    Public dc As libnodave.daveConnection
    Public res As Integer
    Public buf(1000) As Byte
    Public localMPI As Integer = 0
    Public rack As Integer = 0
    Public slot As Integer = 2
    Public plcMPI As Integer = 2
    Public Connection As Boolean = False
    Public IP As String = "192.168.1.20"
End Module
```

ΚΕΦΑΛΑΙΟ 7

7.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Το έναυσμα για την εκπόνηση αυτής της πτυχιακής, μας δόθηκε σχεδόν από τα πρώτα εξάμηνα φοίτησης μας όταν είδα την εγκατάσταση των PLC στο εργαστήριο και ενδιαφέρθηκα αμέσως να μάθουμε ποια είναι η χρήση τους. Η τελική απόφαση για ενασχόληση με αυτοματισμούς ήρθε μετά από την παρακολούθηση του εργαστηρίου Βιομηχανικοί Αυτοματισμοί.

Τα κύρια συμπεράσματα που απορρέουν από την πτυχιακή εργασία χωρίζονται σε αρκετούς τομείς. Ξεκινώντας από την εργασία αναφερθήκαμε σε όλα τα επιμέρους συστήματα που χρησιμοποιήσαμε, από τα αισθητήρια μέχρι τους βιομηχανικούς ελεγκτές και τη γλώσσα προγραμματισμού. Η ανάλυση προχώρησε στο καθαρά στο προγραμματιστικό μέρος της εργασίας όπου και αναφερθήκαμε σε όλες τις σημαντικές παραμέτρους, που χρειάζεται ο αναγνώστης για να κατανοήσει την δομή και την λειτουργία των PLC και εν τέλει παραθέσαμε και τα κομμάτια κώδικα αλλά και εικόνες για την κατανόηση του λογισμικού.

Σε μια επίσκεψη μας σε βιομηχανική εγκατάσταση συναντήσαμε το PLC s7-300 το οποίο είχε χρήση σε πολλά σημεία της βιομηχανίας. Οπότε καταλαβαίνουμε πως η υπάρχουσα εγκατάσταση στο εργαστήριο μπορεί να βρει εφαρμογές σε πολλά μέρη. Ο σκοπός της πτυχιακής ήταν να κατανοήσω αφενός τα PLC αλλά και να λάβουμε γνώσεις για την μελλοντική ενασχόληση μας

Αναμφίβολα το μέλλον στην Βιομηχανία αποτελούν οι αυτοματισμοί με PLC και συστημάτων SCADA που όλα μαζί σε συνεργασία προσφέρουν ταχύτητα, ευελιξία και καλύτερης ποιότητας τελικό προϊόν μειώνοντας το κόστος.

Το θέμα της πτυχιακής αποφασίστηκε μετά από πολύ σκέψη με συνεργασία του κ.Βλησίση και του Κ.Χαρακόπουλου όταν ψάχναμε για συστήματα αυτοματισμού που δεν υπάρχουν στην αγορά. Μετά από αυτό ξεκίνησε η εντατική μελέτη για την κατανόηση των PLC και των αυτοματισμών.

Σε όλες τις μεγάλες βιομηχανίες που ψάξαμε είδα πως όλες τις απασχολεί ο κίνδυνος πυρκαγιάς και η επίβλεψη αυτού του αυτοματισμού. Οπότε το μοντέλο που αναπτύξαμε αποτελεί ένα ολοκληρωμένο και άκρως προσιτό αυτοματισμό για μεγάλες αλλά και μικρές βιομηχανίες.

7.2 Μελλοντική εργασία και Επεκτάσεις.

Επειδή τα PLC που υπάρχουν στο εργαστήριο έχουν μια παρά πολύ μεγάλη γκάμα εφαρμογών και πολύ υψηλές δυνατότητες και βρίσκουν εφαρμογές παντού αυτές τις μέρες θα κάνουμε την δική μας πρόταση για την επέκταση της ήδη υπάρχουσας εγκατάστασης αλλά και αν υπάρχει το Budget για εγκατάσταση PLC νέας γενιάς. Σαν μελλοντική πτυχιακή πρώτον θα μπορούσε να εγκατασταθεί στο κτήριο ENPIET η δική μας μελέτη απλά με αγορά και τοποθέτηση του υλικού. Επίσης στο ήδη υπάρχον S-300, S-200, LOGO! ένα δίκτυο HART. Δηλαδή την επικοινωνία των PLC ασύρματα. Γι αυτή την λειτουργία υπάρχει Hardware της Siemens και σχετική καθοδήγηση. Περεταίρω όπως προανέφερα εάν υπάρχει η δυνατότητα αγοράς νέου εξοπλισμού, τοποθέτηση νέας γενιάς PLC και η διαχείριση τους μέσω Android Εφαρμογής.

Βιβλιογραφία

1. **Dennis Collins, Eamon Lane.** *Προγραμματιζόμενοι Ελεγκτές.* Θεσσαλονίκη : Α . Τζιόλα, 2003.
2. **Petruzella, Frank.** *Προγραμματιζόμενοι Λογικοί Ελεγκτές.* Θεσσαλονίκη : Α. Τζιόλα, 2000.
3. **Siemens.** *WinnCC flexible 2008 getting Started.* [First Time Users Getting Started] Germany : Siemens, 2008.
4. —. *Siemens SIAMATIC working with STEP 7.* [Getting Started Edition] Germany : Siemens, 2006.
5. —. Siemens. *www.siemens.com.* [Ηλεκτρονικό] 3 2011. [Παραπομπή: 25 6 2013.]
6. —. *Siemens S7-300 programmable Controllers Module Specifications.* [Siemens Ind] Germany : Siemens, 2011.
7. **Mahony, John O'.** *Accesing MySql from Visual Basic.* s.l. : Risaris Ltd., 2006.
8. **Μάνεσης, Σταμάτης.** *Δίκτυα Βιομηχανικών Αυτοματισμών.* Πάτρα : Πανεπιστήμιο Πατρών, 2003.
9. —. *Συστήματα Βιομηχανικών Αυτοματισμών.* Πάτρα : Πανεπιστήμιο Πατρών, 2003.
10. **Giorgos M. Papadourakis, Dominique Daens.** *Training In Industrial Ethernet.* Heraklion,Belgium : Report, 2010.

Ιστότοποι

- [1]. <http://www.siemens.com>
- [2]. <http://www.plcs.net>
- [3]. <http://www.plcforum.org>
- [4]. <http://support.automation.siemens.com>
- [5]. <http://www.stackoverflow.com>
- [6]. <http://www.codeproject.com>
- [7]. <http://www.parijat.com>