



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΗΣ ΙΣΧΥΟΣ

Εργαστηριακές εφαρμογές SCADA

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΕΩΡΓΙΟΣ Σ. ΣΚΙΑΔΟΠΟΥΛΟΣ

Επιβλέπων: Γεώργιος Ν. Κορρές
Αν. Καθηγητής Ε.Μ.Π

Αθήνα, Σεπτέμβριος 2007



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΗΣ ΙΣΧΥΟΣ

Εργαστηριακές εφαρμογές SCADA

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΕΩΡΓΙΟΣ Σ. ΣΚΙΑΔΟΠΟΥΛΟΣ

Επιβλέπων: Γεώργιος Ν. Κορρές
Αν. Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την

.....
Γ. Κορρές
Αν. Καθηγητής Ε.Μ.Π

.....
Κ. Βουρνάς
Καθηγητής Ε.Μ.Π

.....
Πεკμεστζή Κιαμάλ
Καθηγητής Ε.Μ.Π

Αθήνα, Σεπτέμβριος 2007

.....
ΓΕΩΡΓΙΟΣ Σ. ΣΚΙΑΔΟΠΟΥΛΟΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © ΓΕΩΡΓΙΟΣ Σ.ΣΚΙΑΔΟΠΟΥΛΟΣ, 2007
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία εκπονήθηκε στον τομέα Ηλεκτρικής Ισχύος της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του ΕΜΠ, στα πλαίσια των ερευνητικών δραστηριοτήτων του Αναλογικού Εργαστηρίου ΣΗΕ. Αντικείμενό της είναι η υλοποίηση εργαστηριακών εφαρμογών με το σύστημα SCADA.

Υπεύθυνος κατά την εκπόνηση της διπλωματικής ήταν ο Αναπληρωτής Καθηγητής κ. Γ. Κορρές, στον οποίο οφείλω ιδιαίτερες ευχαριστίες τόσο για την ανάθεση αυτής όσο και για την δυνατότητα που μου δόθηκε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα. Επιπλέον, θα ήθελα επίσης να ευχαριστήσω τον επιστημονικό συνεργάτη κ. Γεώργιο Αρβανίτη για τις πολύτιμες συμβουλές του και το αμέριστο ενδιαφέρον που επέδειξε όλο αυτό τον καιρό, όπως επίσης και τον υπεύθυνο του εργαστηρίου κ. Γεώργιο Κατσαρό για τις πολύτιμες τεχνικές πληροφορίες που μου παρείχε.

Τέλος, θα ήθελα να αφιερώσω τον παρόν έργο στους γονείς μου, τους ανθρώπους που, με τη μακροχρόνια και εμπνευσμένη καθοδήγηση τους, συνέβαλλαν τα μέγιστα ώστε να διευρύνω το γνωστικό μου πεδίο και να είμαι ικανός να πετυχαίνω τους στόχους που κατά καιρούς θέτω.

Αθήνα, Σεπτέμβριος 2007

ΓΕΩΡΓΙΟΣ Σ. ΣΚΙΑΔΟΠΟΥΛΟΣ

ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας εργασίας ήταν η υλοποίηση σεναρίων βασικών εφαρμογών αυτοματισμού με το σύστημα SCADA της εταιρείας SIEMENS. Εκμεταλλευόμενοι τις δυνατότητες του προσομοιωτή Προγραμματιζόμενου Λογικού Ελεγκτή (PLC) του εργαστηρίου, κατορθώσαμε και εστίασαμε τις προσπάθειές μας στο προγραμματιστικό μέρος των εφαρμογών, αποφεύγοντας ηλεκτρομηχανολογικές κατασκευές.

Συγκεκριμένα, ο προσομοιωτής αποτελείται από ένα PLC και έχει επιπλέον τη δυνατότητα προσομοίωσης ψηφιακών και αναλογικών σημάτων. Με τον τρόπο αυτό, ασχοληθήκαμε με την διασύνδεση και ανάπτυξη κατάλληλων περιβάλλοντων παρακολούθησης και ελέγχου, με τη χρήση του λογισμικού WinCC της εταιρείας SIEMENS. Δημιουργήσαμε, έτσι, ολοκληρωμένα συστήματα SCADA, τα οποία αποτέλεσαν τόσο μία πολύτιμη γνωστική εμπειρία όσο και εκπαιδευτικά εργαλεία στα πλαίσια των διαλέξεων του μαθήματος “Κέντρα Ελέγχου Ενέργειας” του 8^{ου} εξαμήνου.

Ξεκινώντας από την περιγραφή του προσομοιωτή στο κεφάλαιο 1, συνεχίζουμε με ανάλυση και περιγραφή του πακέτου WinCC, το οποίο χρησιμοποιήσαμε, στο κεφάλαιο 2. Στη συνέχεια, στο κεφάλαιο 3, δημιουργούμε μία δοκιμαστική εφαρμογή για περαιτέρω εμπέδωση των στοιχείων του WinCC που παρουσιάσαμε προηγουμένως και στο κεφάλαιο 4 υλοποιούμε έναν εικονικό προσομοιωτή, ακριβώς όμοιο με τον πραγματικό. Στο κεφάλαιο 5 υλοποιούμε το πρώτο σενάριο με τον έλεγχο Μ/Σ ισχύος και στο κεφάλαιο 6 τον έλεγχο σηματοδοτών σε διάβαση πεζών και οχημάτων. Στα κεφάλαια 7 και 8 έχουμε την υλοποίηση ελέγχου συστήματος διασυνδεδεμένων δεξαμενών και ελέγχου και προστασίας γραμμής μεταφοράς διπλού κυκλώματος, αντίστοιχα. Τέλος, στο κεφάλαιο 9 παραθέτουμε τα συμπεράσματά μας από την ενασχόληση με την παρούσα εργασία και στο παράρτημα, επίσης, παραθέτουμε τα προγράμματα αυτοματισμού που χρησιμοποιήσαμε.

ABSTRACT

The purpose of this diploma thesis has been the creation of basic scenarios of automation applications according to the SCADA system developed by SIEMENS. Taking advantage of the Programmable Logic Unit (PLC) simulator that exists in the lab, we had the potential to focus on the programming part of the applications, avoiding extra electromechanical manufactures.

The simulator consists of a PLC unit, having also the ability to simulate digital and analog signals. In such way, we worked on the connection and development of suitable monitoring and control environments, using the WinCC software developed by SIEMENS. Thereby, we have created full SCADA systems which are proved to be both a unique knowledgeable experience and a valuable teaching tool for the 8th semester's course "Energy Control Centers".

Starting with chapter 1, we describe the simulator and continue with a description of the WinCC software which we used, in chapter 2. Furthermore, in chapter 3, we create a test application for the comprehension of WinCC's elements being previous presented and in chapter 4 we effectuate a virtual simulator, exactly the same with the real one. In chapter 5 we effectuate the first scenario referring to the control of af a power transformer and in chapter 6 the control of traffic lights in a vehicles and pedestrians crossing. In chapters 7 and 8 we create the control of two connected watertanks and the control and protection of a double circuit power line. Last but no least, we present our conclusions drawn in chapter 9 and the automation programs in the appendix.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1 Το σύστημα προσομοίωσης

1.1	Εισαγωγή στον προσομοιωτή	10
1.2	Δομή και λειτουργία	10
1.2.1	Βασικά στοιχεία.	10
1.2.2	Εξομοίωση ψηφιακών σημάτων	11
1.2.3	Εξομοίωση αναλογικών σημάτων	13
1.3	Συμβολή του προσομοιωτή	16

ΚΕΦΑΛΑΙΟ 2 Το περιβάλλον SCADA WinCC της SIEMENS

2.1	Εισαγωγή στα SCADA	18
2.1.1	Γενικά.	18
2.1.2	Τα βασικά μέρη ενός SCADA	18
2.2	Το περιβάλλον WinCC	21
2.2.1	Βασική δομή WinCC	22
2.2.2	Computer (Υπολογιστής)	22
2.2.3	Tag Management (Διαχείριση Μεταβλητών)	23
2.2.4	Data Types (Τύποι Μεταβλητών)	24
2.2.5	Editor	25

ΚΕΦΑΛΑΙΟ 3 Δημιουργία εφαρμογής με το WinCC

3.1	Εισαγωγή	29
3.2	Διαδικασία δημιουργίας εφαρμογής	29
3.2.1	Μεθοδολογία	29
3.2.2	Εκκίνηση του Wincc	30
3.2.3	Δημιουργία της εφαρμογής	30
3.2.4	Επιλογή και εγκατάσταση ενός driver	32
3.2.5	Ορισμός των μεταβλητών (tags)	34
3.2.6	Δημιουργία και εμφάνιση των εικόνων της εφαρμογής	38
3.2.7	Καθορισμός των ιδιοτήτων του WinCC Runtime	44

ΚΕΦΑΛΑΙΟ 4 Ο Προσομοιωτής ως εφαρμογή WinCC

4.1	Γενικά	46
4.2	Δημιουργία εφαρμογής	46
4.2.1	Περιγραφή	46
4.2.2	Ορισμός μεταβλητών	46
4.2.3	Σχεδιαστικό περιβάλλον	48

ΚΕΦΑΛΑΙΟ 5 Σενάριο 1: Έλεγχος Μ/Σ Ισχύος

5.1	Αντικείμενο ζητούμενης εφαρμογής	58
5.2	Μετασχηματιστής και σήματα εντοπισμού σφαλμάτων	58
5.3	Δημιουργία εφαρμογής	61
5.3.1	Πρόγραμμα αυτοματισμού	61
5.3.2	Δημιουργία περιβάλλοντος εργασίας	63

ΚΕΦΑΛΑΙΟ 6 Σενάριο 2 : Έλεγχος σηματοδοτών

6.1	Αντικείμενο ζητούμενης εφαρμογής	72
6.2	Διάβαση πεζών και επίσπευση αναμονής	72
6.3	Δημιουργία εφαρμογής	73
6.3.1	Πρόγραμμα αυτοματισμού	73
6.3.2	Δημιουργία περιβάλλοντος εργασίας	74

ΚΕΦΑΛΑΙΟ 7 Σενάριο 3 : Έλεγχος συστήματος δεξαμενών

7.1	Αντικείμενο ζητούμενης εφαρμογής	80
7.2	Σύστημα δύο διασυνδεδεμένων δεξαμενών	80
7.3	Δημιουργία εφαρμογής	82
7.3.1	Πρόγραμμα αυτοματισμού	82
7.3.2	Δημιουργία περιβάλλοντος εργασίας	82

ΚΕΦΑΛΑΙΟ 8 Σενάριο 4 : Έλεγχος & Προστασία Γραμμής Μεταφοράς Διπλού Κυκλώματος

8.1	Αντικείμενο ζητούμενης εφαρμογής	94
8.2	Γραμμή μεταφοράς διπλού κυκλώματος	94
8.2.1	Πρόγραμμα αυτοματισμού	98
8.2.2	Δημιουργία περιβάλλοντος εργασίας	98

ΚΕΦΑΛΑΙΟ 9 Συμπεράσματα

9.1	Συμπεράσματα – Προοπτικές	106
-----	---------------------------	-----

ΚΕΦΑΛΑΙΟ 1

Το Σύστημα Προσομοίωσης

1.1 Εισαγωγή στον προσομοιωτή.

Η παρούσα διπλωματική εργασία στηρίχθηκε στο σύστημα προσομοίωσης, το οποίο αποτελείται από δύο πίνακες αυτοματισμού που ο καθένας χωριστά απεικονίζει ένα πλήρες εργαστηριακό σύστημα PLC και είναι αποτέλεσμα της διπλωματικής εργασίας με τίτλο «**Εργαστηριακό σύστημα προσομοίωσης προγραμματιζόμενου λογικού ελεγκτή**» του Θωμά Β.Σιαμπαλιώτη. Σκοπός του συστήματος αυτού είναι η δημιουργία ενός πραγματικού περιβάλλοντος στο οποίο οι φοιτητές του μαθήματος «**ΚΕΝΤΡΑ ΕΛΕΓΧΟΥ ΕΝΕΡΓΕΙΑΣ**» του 8^{ου} εξαμήνου της σχολής **Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών** θα έχουν τη δυνατότητα να δοκιμάζουν τα προγράμματά τους, γεγονός που προσφέρει ποικίλλα πλεονεκτήματα σε σχέση με την δοκιμή στον εικονικό προσομοιωτή του **Simatic Manager** της εταιρείας **SIEMENS**. Εργαζόμενοι με έναν από τους προσομοιωτές αυτούς κάθε φορά, μπορέσαμε και αναπτύξαμε ένα πλήρες σύστημα SCADA, υλοποιώντας τα ζητούμενα σενάρια διαφόρων εφαρμογών. Παρακάτω ακολουθεί μία περιληπτική περιγραφή της βασικής δομής και λειτουργίας ενός τέτοιου προσομοιωτή.

1.2 Δομή και λειτουργία.

1.2.1 Βασικά στοιχεία.

Τα βασικά στοιχεία τα οποία συνθέτουν έναν από τους παραπάνω προσομοιωτές είναι τα ακόλουθα :

- Μια τροφοδοτική μονάδα PS307/2A εξόδου 24 VDC ικανότητας 2 A.
 <<PS 307 2A (307-1BA00-0AA0) >>
- Μία CPU 312c, με ενσωματωμένη κάρτα 10 ψηφιακών εισόδων και 6 ψηφιακών εξόδων.
 <<CPU 312C (312-5BD00-0AB0)>>
- Μια αναλογική κάρτα 8 εισόδων.
 <<SM331 (331-7KF01-0AB0)>>

Τα παραπάνω βασικά μέρη του προσομοιωτή ολοκληρώνονται με τα απαραίτητα ηλεκτρολογικά στοιχεία, όπως φαίνεται και στις εικόνες 1.1 και 1.2, παρακάτω.

1.2.2 Εξομοίωση ψηφιακών σημάτων.

Όπως προαναφέρθηκε στην 1.2.1, η CPU έχει ενσωματωμένη ψηφιακή κάρτα 10 εισόδων και 6 εξόδων. Επομένως, μπορεί να δεχτεί μέχρι 10 ψηφιακά σήματα για επεξεργασία, τα οποία δημιουργούνται με τη βοήθεια 10 μονοπολικών διακοπών δύο θέσεων, ικανών να προσομοιώσουν πληθώρα ψηφιακών σημάτων που μπορεί να προέρχονται π.χ από πιεστικούς διακόπτες (buttons) Start/Stop για την εκκίνηση ενός κινητήρα, τερματικούς διακόπτες, επαφές θερμικών ενός κινητήρα ή μίας αντλίας κ.τ.λ. Γενικά, υλοποιήθηκαν με βάση την ακόλουθη λογική:

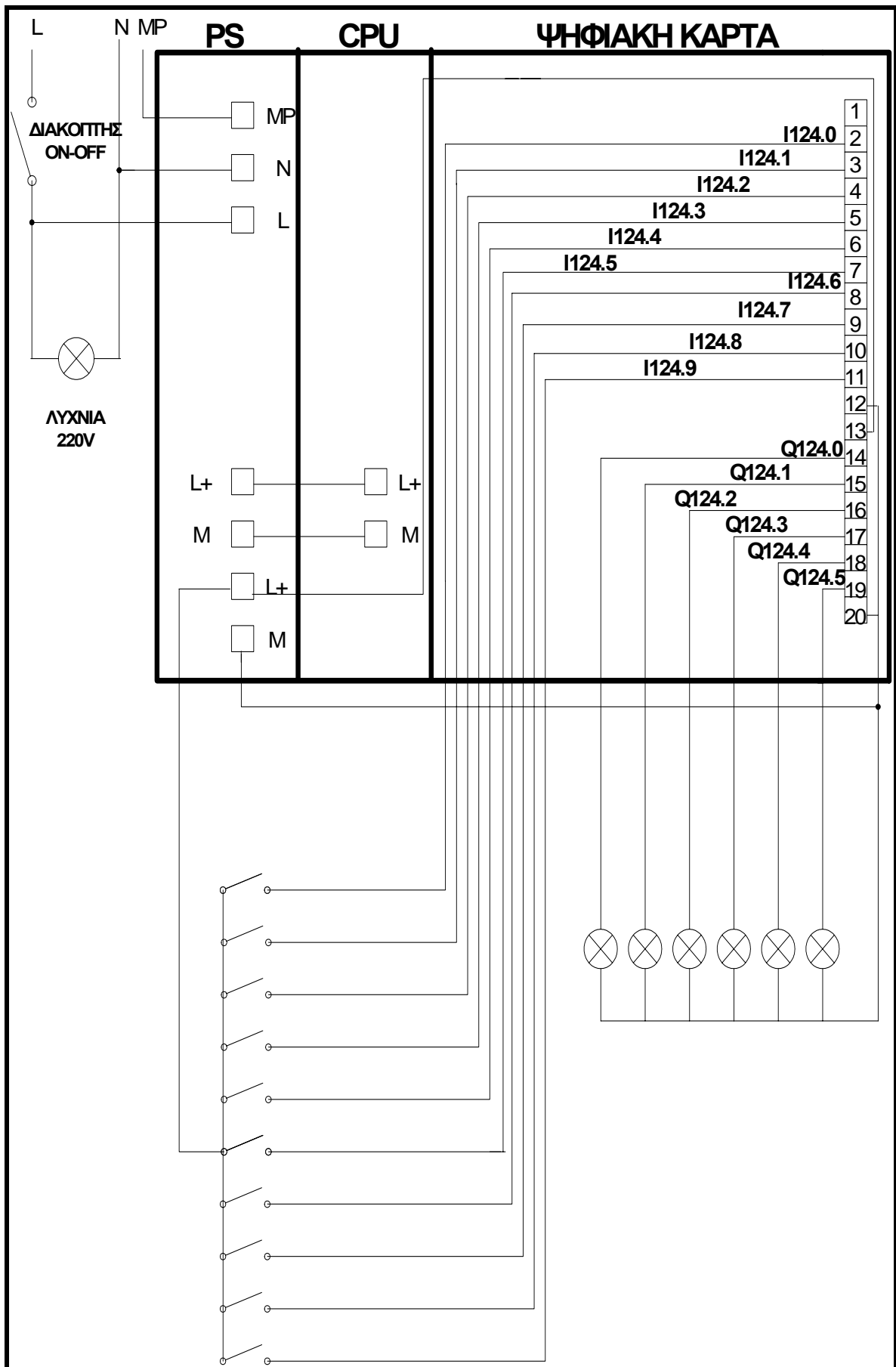
- Όταν ο διακόπτης είναι ανοικτός, στην είσοδο της ψηφιακής κάρτας του PLC έχουμε 0V DC, δηλαδή λογικό-“0”.
- Όταν ο διακόπτης είναι κλειστός, στην είσοδο της ψηφιακής κάρτας του PLC έχουμε 24V DC, δηλαδή λογικό-“1”.

Τέλος, όπως δείχνει και το σχήμα 1, οι 10 ψηφιακές εισοδοι αντιστοιχούν στις διευθύνσεις :

*I124.0 , I124.1 , I124.2 , I124.3 , I124.4 , I124.5 , I124.6 , I124.7 , I124.8 , I124.9 ,
I125.0 , I125.1*

και οι 6 ψηφιακές εξοδοι στις διευθύνσεις :

Q124.0 , Q124.1 , Q124.2 , Q124.3 , Q124.4 , Q124.5

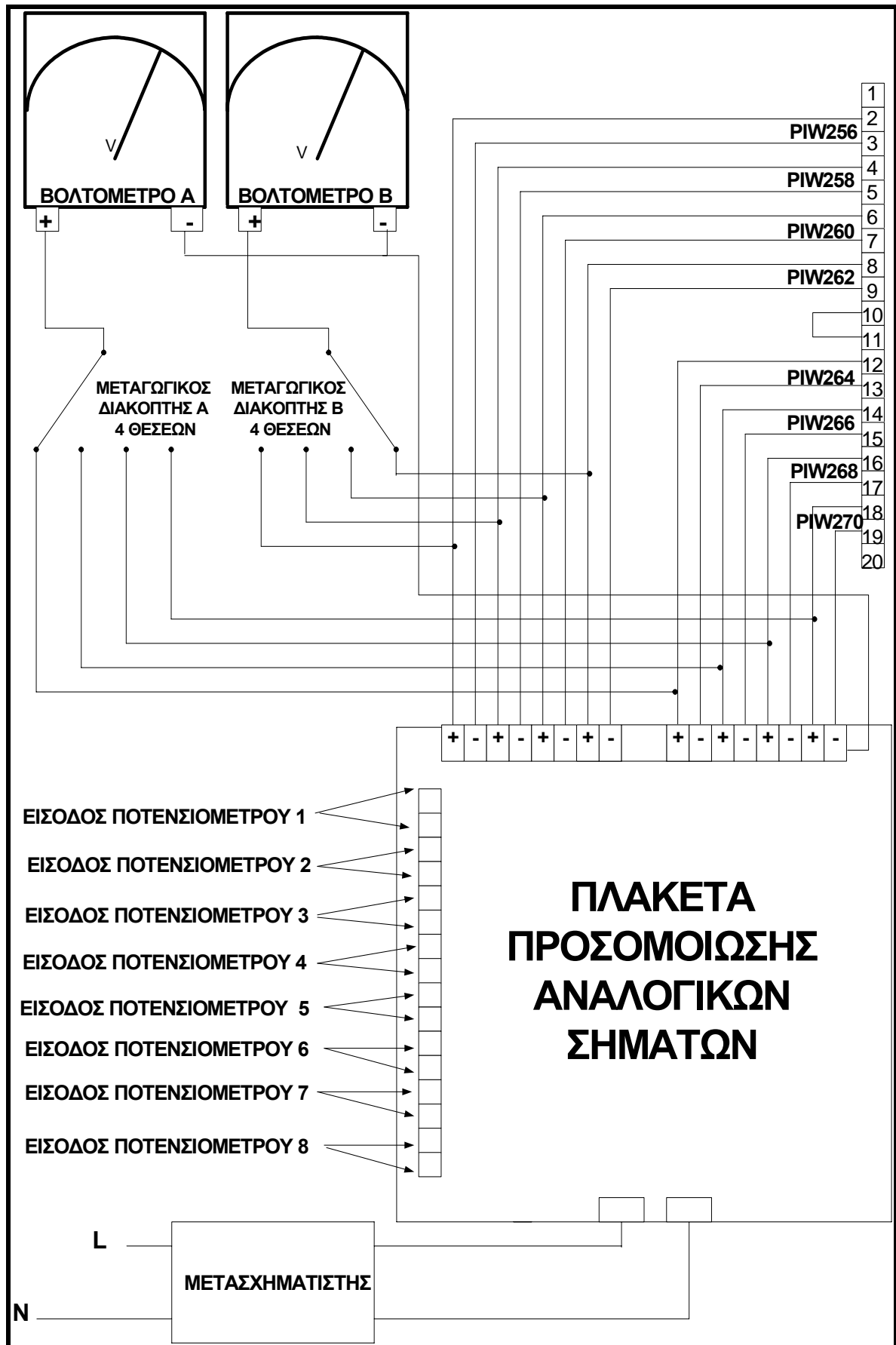


Σχήμα 1.1. Εξομοίωση ψηφιακών σημάτων εισόδων/εξόδων.

1.2.3 Εξομοίωση αναλογικών σημάτων.

Στο ίδιο rack με την CPU υπάρχει και μία κάρτα αναλογικών σημάτων 8 εισόδων, στην οποία είναι συνδεδεμένη μία πλακέτα που παράγει «εικονικά» αναλογικά σήματα. Η πλακέτα αυτή, δέχεται στην είσοδό της 12V AC -μετά την παρεμβολή ενός μετασχηματιστή- και στην έξοδό της παράγει 8 DC τάσεις μεταβλητής μέσης τιμής. Μεταβάλλοντας την μέση τιμή προσομοιώσαμε τα διαφορετικά αναλογικά σήματα που ήταν αναγκαία για την ανάπτυξη των ζητούμενων εφαρμογών, όπως τάση, συχνότητα, πίεση, στάθμη ρευστού κτλ.

Τα σήματα που προκύπτουν από την παραπάνω επεξεργασία, επειδή είναι DC τάσεις, απεικονίζονται με τη βοήθεια δύο DC βολτομέτρων, συνδεδεμένα απευθείας με δύο μεταγωγικούς διακόπτες τεσσάρων θέσεων, κάθε μία από τις οποίες αντιστοιχεί σε διαφορετικές διευθύνσεις της κάρτας αναλογικών σημάτων της CPU, όπως φαίνεται καθαρά στο σχήμα 2. Με τον τρόπο αυτό, ξεκινώντας από την διεύθυνση *PIW256* και καταλήγοντας στην *PIW270*, έχουμε μία άκρως ικανοποιητική πληθώρα αναλογικών επιλογών για την ανάπτυξη των εργαστηριακών εφαρμογών μας.



Σχήμα 1.2. Εξομοίωση αναλογικών σημάτων.



Εικόνα 1.1. Εμπρόσθια όψη των δύο προσομοιωτών.



Εικόνα 1.2. Προσομοιωτής σε λειτουργία.

1.3 Συμβολή του προσομοιωτή.

Η παρούσα διπλωματική εργασία αποτελεί συνέχεια της διπλωματικής εργασίας του συναδέλφου Θωμά Σιαμπαλιώτη, αξιοποιώντας όλα τα πλεονεκτήματα που αυτή προσφέρει ώστε να δημιουργηθεί ένα πλήρες και αξιόπιστο σύστημα SCADA για κάθε μία από τις ζητούμενες εφαρμογές. Τα πλεονεκτήματα αυτά είναι:

- ❖ Αποτελεί ένα ολοκληρωμένο σύστημα PLC με έτοιμες υποδοχές εισόδου/εξόδου για περαιτέρω αξιοποίησή του με άλλα ηλεκτρολογικά στοιχεία (ρελέ, αισθητήρες κτλ)
- ❖ Προσφέρει αξιόπιστη εξομοίωση ψηφιακών/αναλογικών σημάτων
- ❖ Είναι μία συμπαγής κατασκευή, εύκολη στη διασύνδεση και χρήση απο τον πιο προχωρημένο μέχρι τον πιο αρχάριο σε θέματα PLC.

Συμπερασματικά και λαμβάνοντας υπόψη τα παραπάνω, η ύπαρξη του προσομοιωτή συνέβαλλε τα μέγιστα για την αποπεράτωση της παρούσας εργασίας. Εξοικονομήθηκε χρόνος, μιας που όλα τα ηλεκτρολογικά μέρη που ήταν απαραίτητα για την διεξαγωγή της ήταν ήδη έτοιμα, ο οποίος αφιερώθηκε στην περαιτέρω βελτιστοποίηση των εφαρμογών, ώστε να ανταποκρίνονται, όσο το δυνατόν, σε πραγματικές συνθήκες λειτουργίας. Έτσι, δημιουργήθηκε ένα πλήρες, κομψό και αποδοτικό σύστημα SCADA κάθε φορά, το οποίο αναδεικνύει τα πλεονεκτήματα που δύναται να προσφέρει σε βιομηχανικές και μη εφαρμογές, ενώ παράλληλα αποτελεί ένα πολύτιμο εκπαιδευτικό εργαλείο για τους φοιτητές οι οποίοι θέλουν να εντρυφήσουν περισσότερο στην πολύπλοκη δομή και λειτουργία του.

ΚΕΦΑΛΑΙΟ 2

Το Περιβάλλον SCADA WinCC της SIEMENS

2.1 Εισαγωγή στα SCADA.

2.1.1 Γενικά.

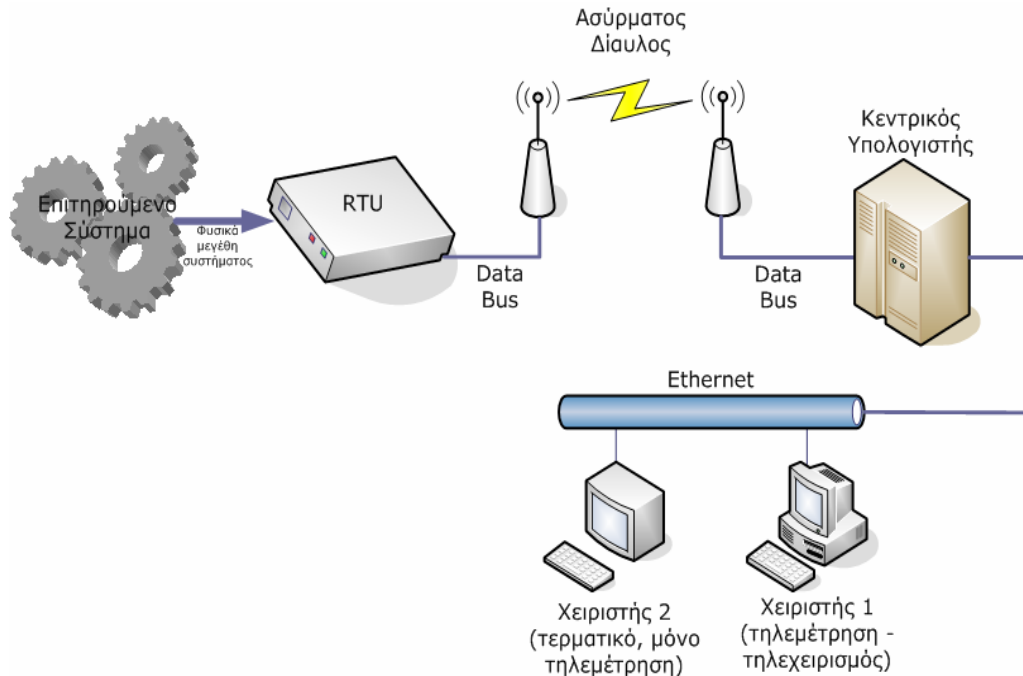
Πριν ξεκινήσουμε την ανάλυση του λογισμικού που χρησιμοποιήσαμε για την ανάπτυξη των εφαρμογών μας, είναι σκόπιμο να αναφέρουμε περιληπτικά τα βασικά στοιχεία που συνθέτουν ένα πλήρες σύστημα SCADA. Η λέξη SCADA αποτελεί τα αρχικά των λέξεων Supervisory, Control And Data Acquisition System, δηλαδή σύστημα εποπτείας, ελέγχου και συλλογής πληροφοριών. Είναι, συνεπώς, σύστημα τηλεμετρίας και τηλεχειρισμού, το οποίο συλλέγει πληροφορίες από διάφορες διεργασίες και τις αξιοποιεί για τον εποπτικό τους έλεγχο. Τα συστήματα SCADA είναι ιδιαίτερα δημοφιλή, τόσο σε ιδιαίτερα σύνθετες βιομηχανικές εφαρμογές, όπου απαιτούνται αυτοματισμοί υψηλής αξιοπιστίας και ακρίβειας όσο και σε συστήματα παραγωγής και διανομής ηλεκτρικής ενέργειας τα οποία απαιτούν συνεχή και λεπτομερή έλεγχο όλων των κρίσιμων παραμέτρων τους.

Παρόλο που τα συστήματα αυτά προορίζονται, κυρίως, για εφαρμογές μεγάλης κλίμακας, στις μέρες μας, ωστόσο, γνωρίζουν μία επιπλέον αλματώδη άνθηση, κυρίως λόγω της επέκτασης τους και στην καθημερινή ζωή. Τα λεγόμενα «έξυπνα σπίτια» στηρίζουν την λειτουργία τους στις αρχές των συστημάτων αυτών και με τη χρήση κατάλληλων PLC και ηλεκτρολογικών υλικών δείχνουν τον δρόμο στο πως η τεχνολογία μπορεί, κάτω από συγκεκριμένες συνθήκες, να κάνει την ανθρώπινη ζωή ακόμα πιο άνετη και ταυτόχρονα φιλική στο περιβάλλον.

2.1.2 Τα βασικά μέρη ενός SCADA.

Ένα σύνθετο σύστημα SCADA χρησιμοποιεί σαν κεντρικό πυρήνα έναν κεντρικό υπολογιστή, αρκετά μεγάλης υπολογιστικής ισχύος, στον οποίο βρίσκεται το λογισμικό SCADA εγκατεστημένο, όπως και το πρόγραμμα της εκάστοτε εφαρμογής. Η ζητούμενη τηλεμετρία στον επιθυμητό χώρο, επιτυγχάνεται με την εγκατάσταση σταθμών τηλεμετρίας RTU (Remote Telemetry Units). Οι σταθμοί αυτοί «διαβάζουν» τις τιμές διαφόρων μεγεθών που μας ενδιαφέρουν (τάση, πίεση, θερμοκρασία κτλ), τις μετατρέπουν σε ηλεκτρικά σήματα και τα σήματα αυτά τα μεταδίδουν ενσύρματα ή ασύρματα με κατάλληλες τηλεπικοινωνιακές ζεύξεις στον κεντρικό υπολογιστή, ανά τακτά χρονικά διαστήματα. Από εκεί και πέρα, αρχίζει η παρακολούθηση και

επεξεργασία τους από τους χρήστες του κεντρικού υπολογιστή και εξάγονται χρήσιμα συμπεράσματα για τη λειτουργία της εκάστοτε διεργασίας (σχήμα 2.1).

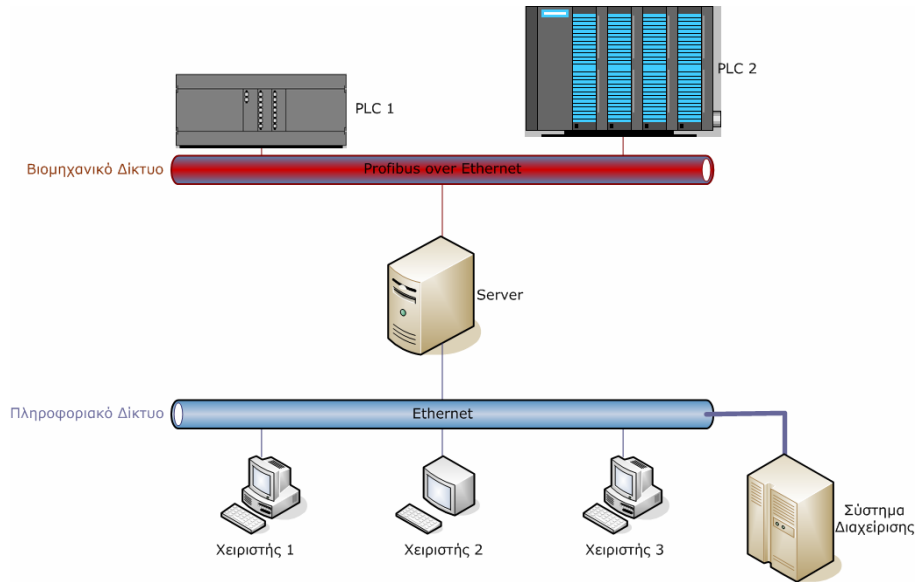


Σχήμα 2.1. Τοπολογία συστήματος SCADA.

Ένα πολύ σημαντικό κεφάλαιο στην ολοκλήρωση ενός συστήματος SCADA αποτελούν οι τύποι των δικτύων, με τα οποία επικοινωνούν τόσο ο κεντρικός υπολογιστής με τους χρήστες όσο και τα διάφορα PLC μεταξύ τους. Για μία σύνθετη βιομηχανική εφαρμογή, συνήθως απαιτούνται αρκετά PLC, το καθένα επιφορτισμένο με ένα συγκεκριμένο τμήμα και η διασύνδεση αυτών έχει άμεσες συνέπειες στην ορθή λειτουργία και άμεση ανταπόκριση του SCADA.

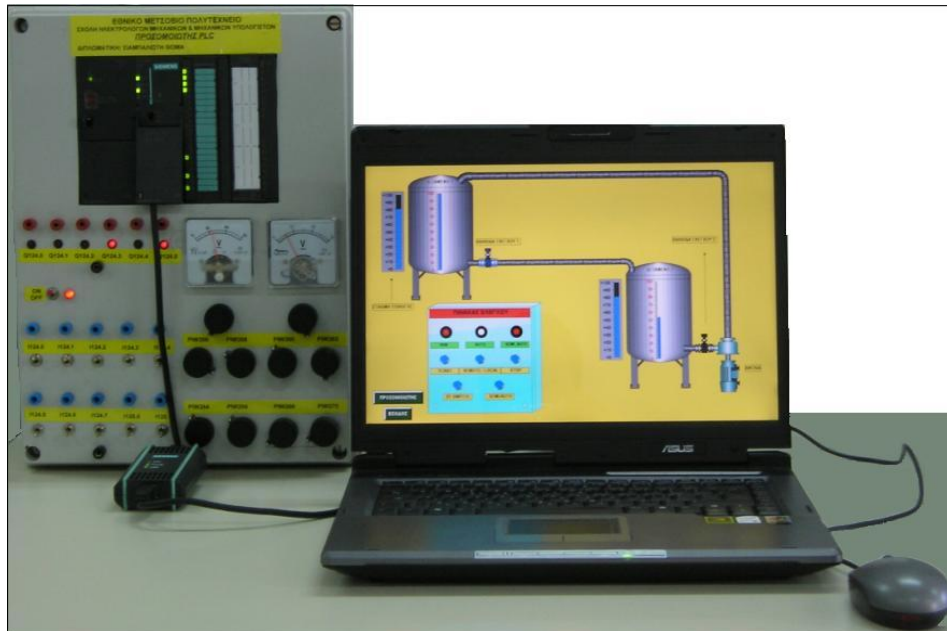
Όπως δείχνει και το σχήμα 2.2, για την δικτύωση του κεντρικού υπολογιστή με τους χρήστες (πληροφοριακό δίκτυο), χρησιμοποιούνται τα πρωτόκολλα Ethernet ή Token Ring ενώ για την δικτύωση των PLC (βιομηχανικό δίκτυο), έχουμε τα πρωτόκολλα Profibus, TCP/IP και Industrial Ethernet, με το πρώτο να είναι και το πιο ευρέως διαδεδομένο. Αξίζει να σημειωθεί ότι σε συστήματα SCADA εγκατεστημένα εντός Ηνωμένων Πολιτειών Αμερικής, κυριαρχεί το πρωτόκολλο ModBus, αν και τα τελευταία χρόνια γίνεται μία, ανεπιτυχής προς το παρόν, προσπάθεια να επικρατήσει το Profibus παγκοσμίως, για λόγους συμβατότητας και ευκολίας. Τέλος, για την περίπτωση των «έξυπνων σπιτιών» για τα οποία έγινε λόγος παραπάνω, έχει αναπτυχθεί ένα πολύ ενδιαφέρον και ευέλικτο δίκτυο στις αρχές των παραπάνω, το Instabus EIB,

ένα σύστημα μεταφοράς και επεξεργασίας δεδομένων μεγάλης ευελιξίας, σχεδιασμένο για τέτοιες διεργασίες μικρής κλίμακας, αλλά μεγάλης αξιοπιστίας και εντυπωσιακών, ομολογουμένως, αποτελεσμάτων.

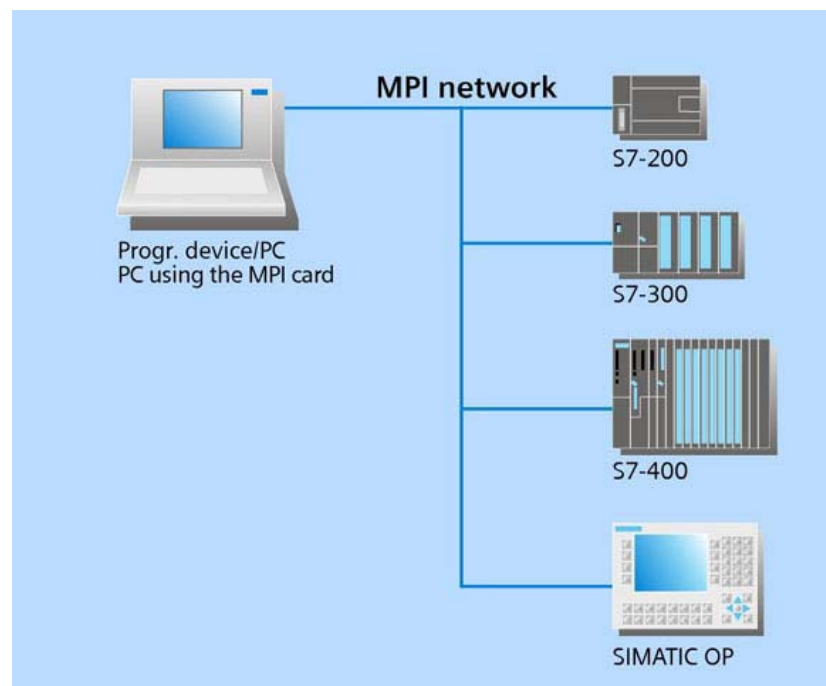


Σχήμα 2.2. Δικτύωση ενός συστήματος SCADA.

Στην παρούσα διπλωματική εργασία τον ρόλο του κεντρικού υπολογιστή έπαιξε ένας φορητός υπολογιστής, αρκετά μεγάλης υπολογιστικής ισχύος (εικόνα 2.1), ενώ το PLC και οι υποτιθέμενοι σταθμοί RTU βρίσκονταν, τοπολογικά, στον προσομοιωτή. Χρησιμοποιώντας την εξομοίωση ψηφιακών και αναλογικών σημάτων, όπως περιγράφηκε στο κεφάλαιο 1, μπορέσαμε να δημιουργήσουμε ένα εικονικό περιβάλλον τηλεμετρίας το οποίο, γενικά, διέφερε ελάχιστα από ένα πραγματικό SCADA. Μοναδικό μειονέκτημα ήταν η αργή διασύνδεση κεντρικού υπολογιστή και PLC μέσω της «αργής» διεπιφάνειας MPI (σχήμα 2.3), αφού διασύνδεσή τους με το «γρήγορο» πρωτόκολλο PROFIBUS ήταν τόσο αδύνατη όσο και ασύμφορη στους χώρους του εργαστηρίου, γεγονός όμως που ουδόλως επηρέασε τόσο το αποτέλεσμα όσο και την εκπαιδευτική αξία του όλου εγχειρήματος. Τέλος, το λογισμικό SCADA που χρησιμοποιήσαμε ήταν το πακέτο SIMATIC WinCC V6.0.1.0 της εταιρείας SIEMENS.



Εικόνα 2.1. Ο κεντρικός υπολογιστής των εφαρμογών μας.



Σχήμα 2.3. Δικτύωση μέσω της διεπιφάνειας MPI.

2.2 Το περιβάλλον WinCC.

Για την εκπόνηση της διπλωματικής εργασίας, αναπτύξαμε τα ζητούμενα σενάρια εφαρμογών με τη βοήθεια του πακέτου SIMATIC WinCC V6.0.1.0 της εταιρείας

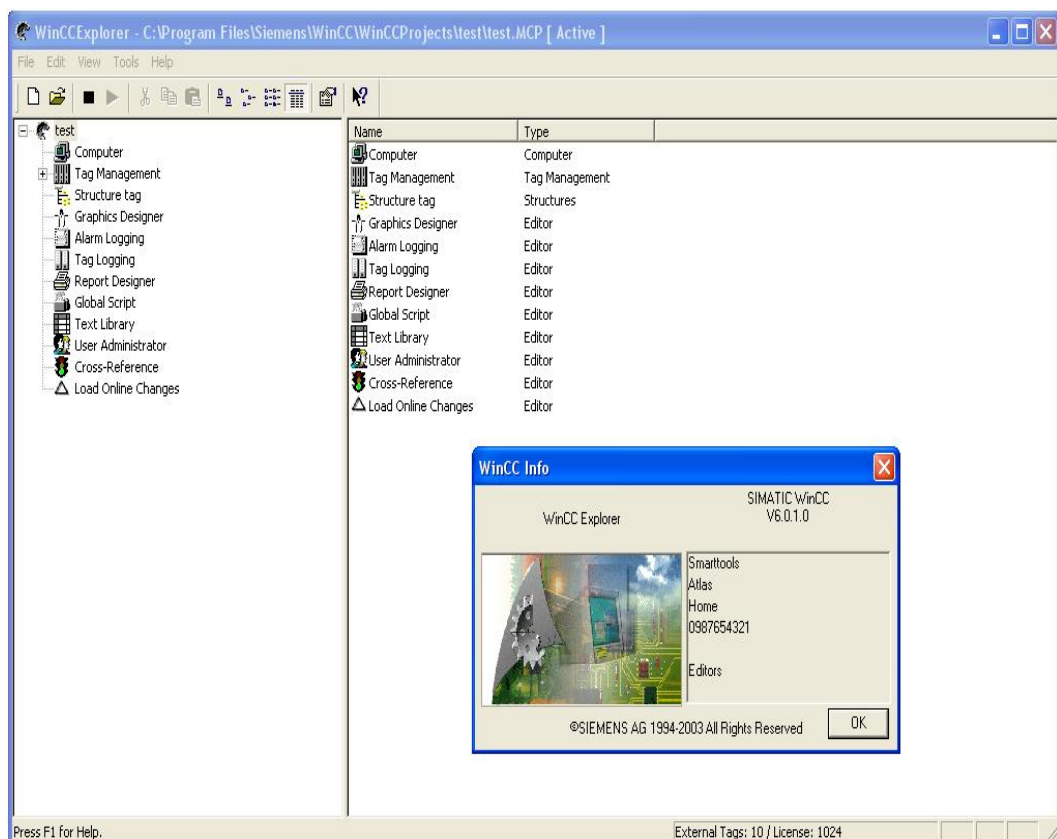
SIEMENS (εικόνα 2.2). Παρακάτω γίνεται μία περιληπτική ανάλυση και περιγραφή του πακέτου αυτού, ώστε να γίνει κατανοητός ο βασικός τρόπος λειτουργίας του.

2.2.1 Βασική δομή WinCC.

Το πακέτο του λογισμικού WinCC αποτελείται γενικά από τέσσερα βασικά τμήματα :

- Computer
- Tag Managements
- Data Types
- Editor

Παρακάτω ακολουθεί περιγραφή των χαρακτηριστικών και του ρόλου που καθένα από αυτά τα τμήματα παίζει, όσον αφορά την δημιουργία και αξιόπιστη λειτουργία ενός συστήματος SCADA.

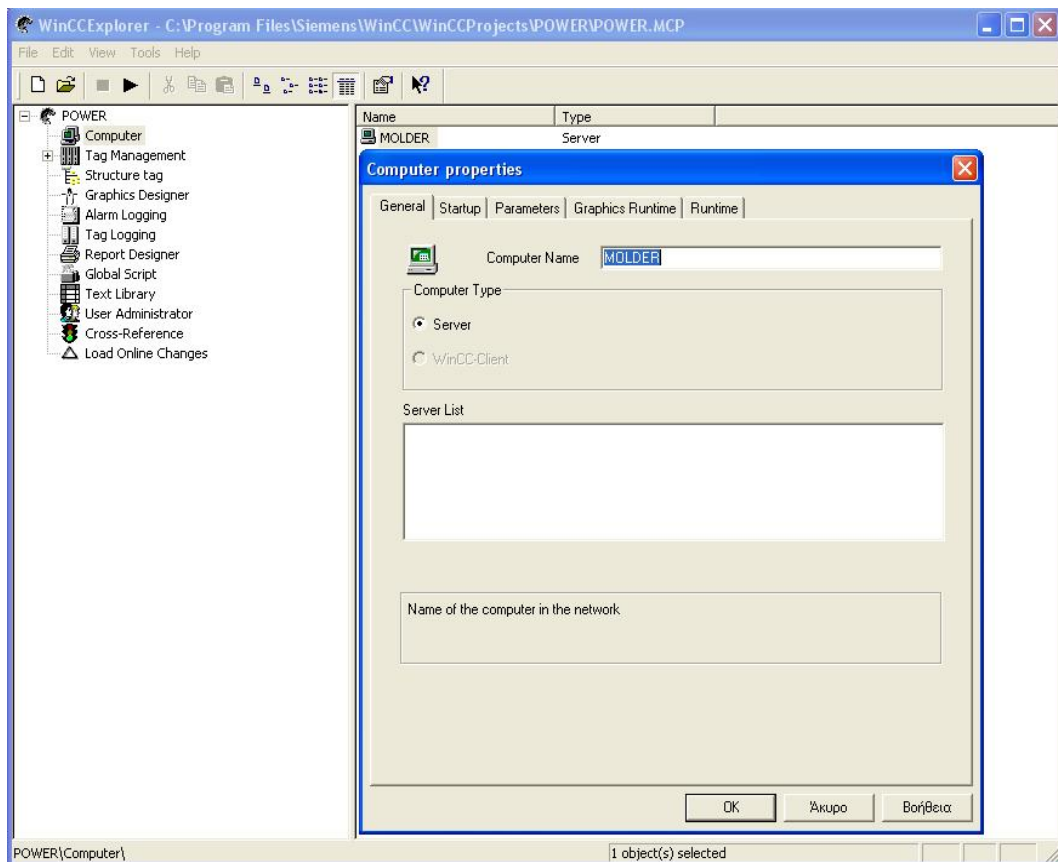


Εικόνα 2.2. Το πακέτο WinCC της εργασίας μας.

2.2.2 Computer (Υπολογιστής).

Στο τμήμα αυτό καθορίζονται, σε γενικές γραμμές, οι υπηρεσίες που θέλουμε να μας προσφέρει το WinCC. Πιο συγκεκριμένα, εδώ γίνεται η παραμετροποίηση των μεταβλητών που

θέλουμε να παρακολουθούνται συνεχώς και να γίνεται καταγραφή των τιμών τους ή να γίνεται συγκεκριμένη αναφορά στην περίπτωση την οποία ξεπεραστεί κάποιο κρίσιμο όριο που εμείς έχουμε θέσει κτλ. Επιπλέον, υπάρχει η δυνατότητα να ορίζονται προγράμματα ανταλλαγής δεδομένων, σε απευθείας συνεργασία με το WinCC, όπως το *Word*, το *Excel* και το *Autocad*, τα οποία είναι ιδιαίτερα χρήσιμα σε περιπτώσεις διεξοδικής επεξεργασίας δεδομένων και κομψής παρουσίασής τους. Τέλος, εδώ υπάρχουν διάφορες επιλογές για την παρουσίαση των γραφικών των εικόνων μας, τις οποίες και εκμεταλλευτήκαμε για την ολοκλήρωση των εφαρμογών μας, όπως θα δούμε στη συνέχεια (εικόνα 2.3).

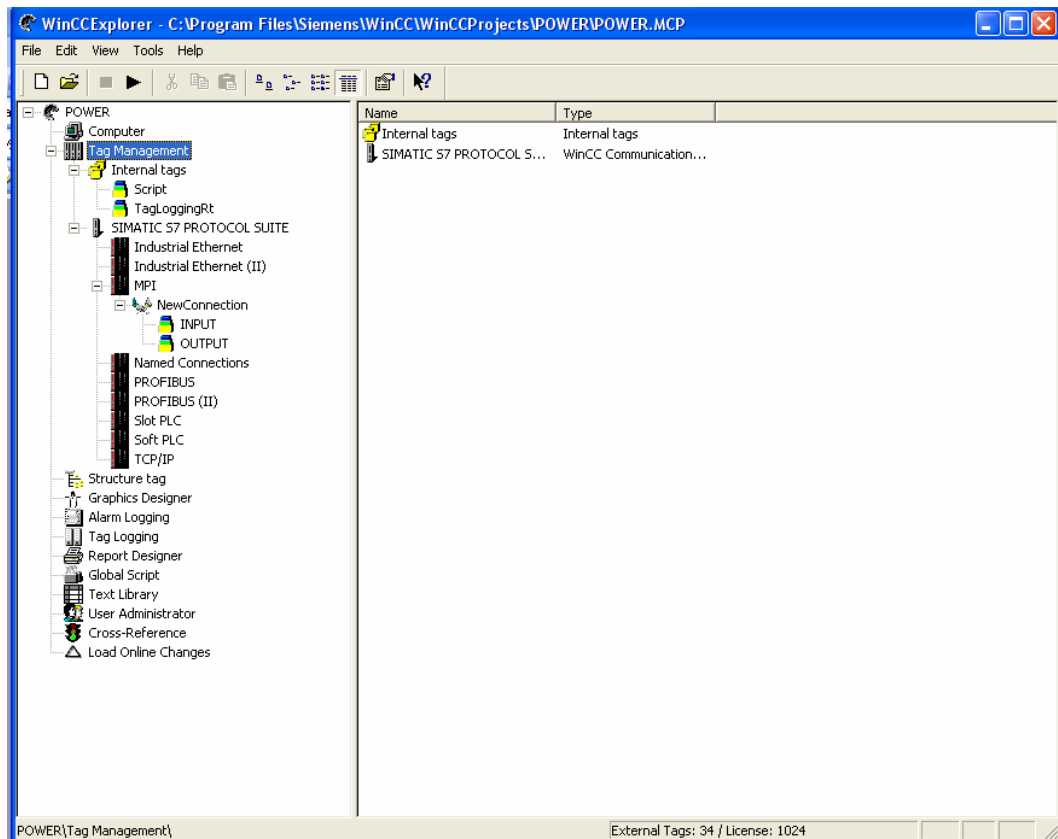


Εικόνα 2.3. Το υπομήμα “Computer”.

2.2.3 Tag Management (Διαχείριση Μεταβλητών).

Εδώ γίνεται ο καθορισμός του πρωτοκόλλου επικοινωνίας του υπολογιστή με το PLC και καθορίζονται οι μεταβλητές του συστήματος. Οι εν λόγω μεταβλητές διακρίνονται στις εξωτερικές μεταβλητές (external tags) και στις εσωτερικές μεταβλητές (internal tags). Οι μεν εξωτερικές είναι στην ουσία μεταβλητές στην μνήμη δεδομένων του PLC, ενώ οι εσωτερικές χρησιμοποιούνται για διάφορες λειτουργίες εντός του WinCC. Όπως καταλαβαίνει κανείς,

πρόκειται για ένα από τα πιο βασικά και πολύπλοκα τμήματα του WinCC, στο οποίο διαμορφώνεται η βασική δομή του συστήματος SCADA (εικόνα 2.4).



Εικόνα 2.4. Το υπομήμα “Tag Management”.

2.2.4 Data Types (Τύποι Μεταβλητών).

Εδώ αναφέρονται όλοι οι δυνατοί τύποι μεταβλητών που χρησιμοποιεί το WinCC :

- Binary tag
- Unsigned 8-bit value
- Signed 8-bit value
- Unsigned 16-bit value
- Signed 16-bit value
- Unsigned 32-bit value
- Signed 32-bit value

- Floating-point number 32-bit IEEE 754
- Floating-point number 64-bit IEEE 754
- Text tag 8-bit character set
- Text tag 16-bit character set
- Raw Data Type
- Text reference
- Structure tag

Συμπερασματικά, κάθε μεταβλητή μπορεί να λάβει συγκεκριμένους τύπους δεδομένων, αναλόγως το είδος και τη λειτουργία που επιτελεί. Για παράδειγμα, όπως θα δούμε και στην ανάπτυξη των εφαρμογών μας στη συνέχεια, οι μεταβλητές που αφορούν αναλογικά σήματα, άρα πρόκειται για αριθμούς κινητής υποδιαστολής, μπορούν να λάβουν μόνο τον τύπο δεδομένων Floating Point Number IEEE 754 32/64 bit, ενώ μεταβλητές που αφορούν τμήματα μνήμης του PLC, μπορούν να λάβουν μόνο τη μορφοποίηση binary tag. Με τον τρόπο αυτό, είναι ξεκάθαρο ότι σωστή δήλωση των μεταβλητών συνεπάγεται και ορθή λειτουργία της εκάστοτε εφαρμογής.

2.2.5 Editor.

Ο Editor του WinCC αποτελείται από πέντε υπομήματα τα οποία είναι :

Graphics Designer

Πρόκειται για το σχεδιαστικό κομμάτι (εικόνα 2.5), το οποίο χρησιμοποιείται για τη δημιουργία του γραφικού περιβάλλοντος της εκάστοτε εφαρμογής. Παρέχει μία επαρκέστατη σειρά από εργαλεία, τα οποία μπορούν να αναπαραστήσουν οποιοδήποτε τμήμα της παραγωγικής διαδικασίας σε εικόνες, καθεμία από τις οποίες αναπαριστά τα σημεία εκείνα που χρήζουν ιδιαίτερης έμφασης. Με τον τρόπο αυτό, ο χρήστης μπορεί να εποπτεύσει το σύστημα με μεγάλη ακρίβεια και λεπτομέρεια που πολύ δύσκολα θα είχε αλλιώς τη δυνατότητα να κάνει.



Εικόνα 2.5. Εκκίνηση του “Graphics Designer”.

Tag Logging

Εδώ γίνεται η καταγραφή των τιμών μίας ή περισσότερων μεταβλητών σε σχέση με το χρόνο. Τα αποτελέσματα των καταγραφών, καθώς και οι γραφικές παραστάσεις, παρουσιάζονται σε πραγματικό χρόνο αναλόγως του χρόνου ανακύκλωσης που έχουμε ορίσει. Ο χρόνος αυτός έχει να κάνει με τη συχνότητα ενημέρωσης των τιμών μίας μεταβλητής και εξαρτάται από το πόσο συχνά απαιτείται η ενημέρωσή τους.

Alarm Logging

Ό,τι υπέρβαση ορίων στις τιμές μεταβλητών, τα οποία έχουμε θέσει εμείς, παρατηρείται κατά τη λειτουργία του συστήματος καταγράφεται εδώ. Ως εκ τούτου, ο ρόλος του υπομνήματος αυτού είναι ιδιαίτερα σημαντικός σε περιπτώσεις παρακολούθησης μεταβλητών ζωτικής σημασίας, τόσο για το σύστημα όσο και για την ασφάλεια του ανθρώπου. Τέτοιες υπερβάσεις καταγράφονται και εμφανίζονται στην κατάσταση Runtime.

Report Designer

Με το εργαλείο αυτό γίνεται αναφορά, σχετική με τις λεπτομέρειες της εφαρμογής που δημιουργήθηκε. Αυτές έχουν να κάνουν με μεταβλητές, εικόνες και στοιχεία που απεικονίζουν, τις ιδιότητες των στοιχείων αυτών κτλ.

Global Script

Με τη χρήση του Global Script, ο χρήστης μπορεί να προγραμματίσει σε γλώσσα προγραμματισμού C. Χρησιμοποιώντας έτοιμες συναρτήσεις ή δημιουργώντας νέες, ο χρήστης έχει τη δυνατότητα να επεκτείνει την εφαρμογή του και να διεισδύσει σε λεπτομέρειες μεγάλου βάθους, αναδεικνύοντας για άλλη μία φορά την πρακτική αξία που έχει το φερόμενο πακέτο λογισμικού.

ΚΕΦΑΛΑΙΟ 3

Δημιουργία εφαρμογής με το WinCC

3.1 Εισαγωγή.

Στο προηγούμενο κεφάλαιο αναφέραμε, με τρόπο περιεκτικό, τα βασικά σημεία τα οποία απαρτίζουν ένα σύστημα SCADA. Πυρήνας ενός τέτοιου συστήματος αποτελεί η δημιουργία του περιβάλλοντος εργασίας με το πακέτο WinCC, το οποίο αποτελεί το βασικό εργαλείο καθορισμού των παραμέτρων του συστήματός μας. Για το λόγο αυτό, προτού προχωρήσουμε στην παρουσίαση των εφαρμογών που δημιουργήσαμε στην παρούσα εργασία, κρίνεται σκόπιμο να περιγράψουμε τον τρόπο δημιουργίας μίας εφαρμογής ώστε να είναι πιο ομαλή η κατανόηση του τρόπου με τον οποίο τις υλοποιήσαμε.

3.2 Διαδικασία δημιουργίας εφαρμογής.

3.2.1 Μεθοδολογία.

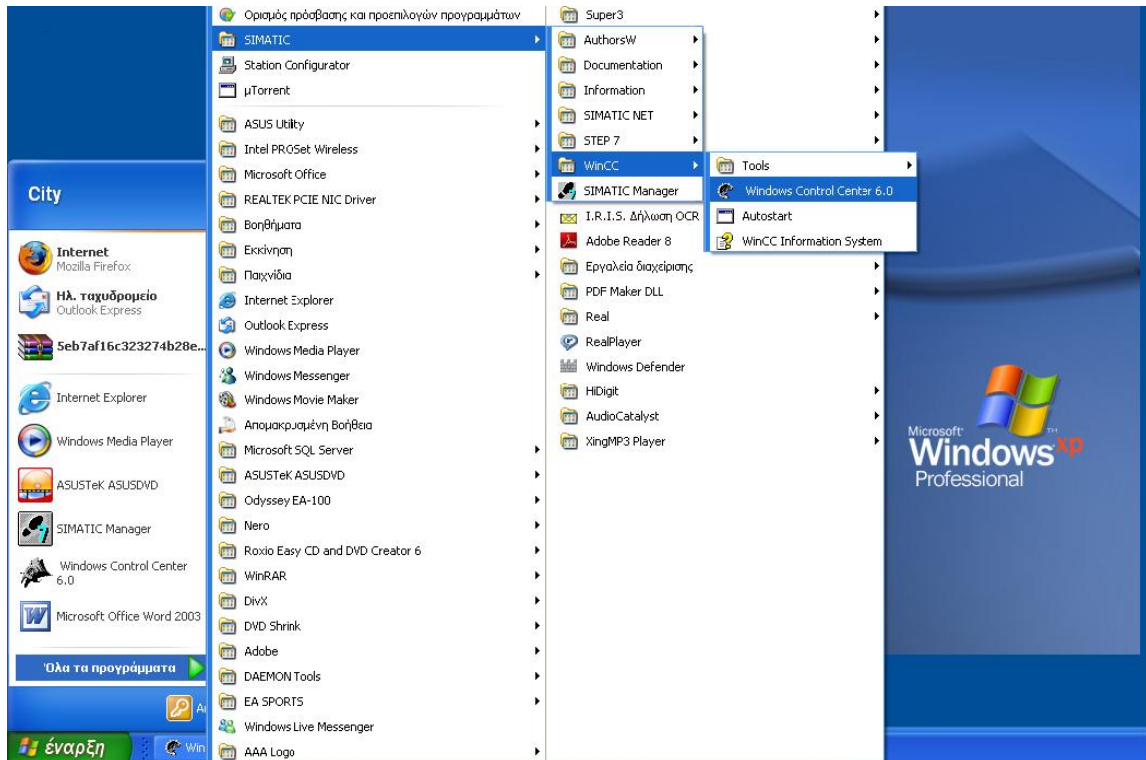
Για να δημιουργήσουμε μία νέα εφαρμογή στο WinCC, ακολουθούμε απαραίτητως τα παρακάτω βασικά βήματα :

1. Εκκίνηση του *WinCC*.
2. Δημιουργία της εφαρμογής.
3. Επιλογή και εγκατάσταση ενός driver.
4. Ορισμός των μεταβλητών (*tags*).
5. Δημιουργία και εμφάνιση των εικόνων της εφαρμογής.
6. Καθορισμός των ιδιοτήτων του *WinCC Runtime*.

Στη συνέχεια ακολουθεί περιγραφή του κάθε βήματος με δημιουργία μίας δοκιμαστικής εφαρμογής για καλύτερη εμπέδωση.

3.2.2 Εκκίνηση του Wincc.

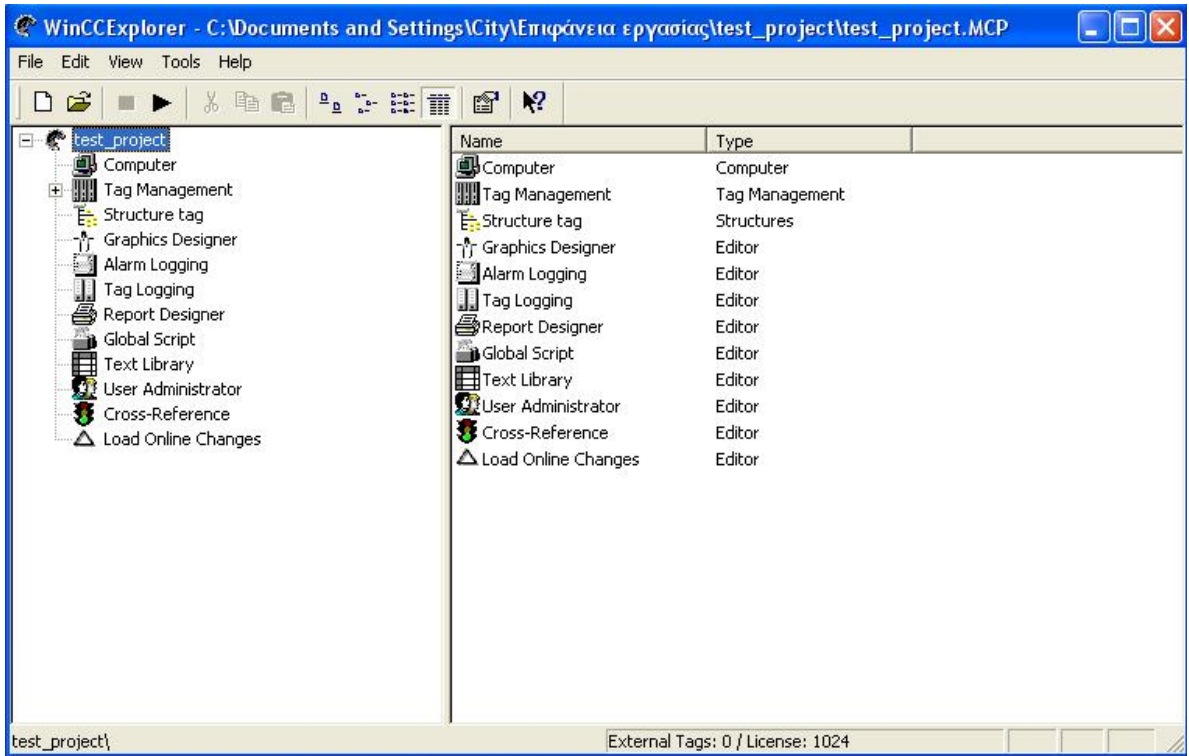
Η εκκίνηση του WinCC, σε λειτουργικό σύστημα Windows XP, γίνεται όπως δείχνει η εικόνα 3.1.



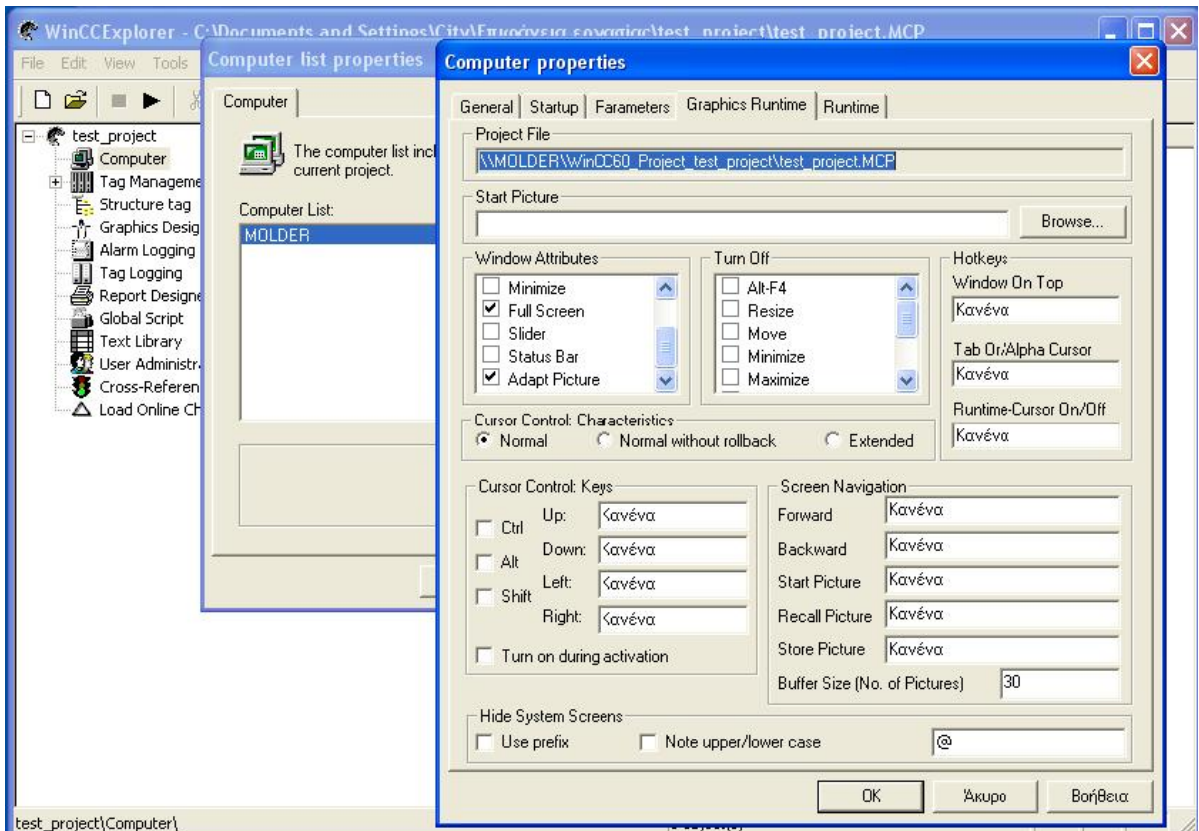
Εικόνα 3.1. Εκκίνηση του WinCC.

3.2.3 Δημιουργία της εφαρμογής.

Αφού εκκινήσουμε το WinCC, τότε πηγαίνουμε και δημιουργούμε την δοκιμαστική μας εφαρμογή με το όνομα “test_project”. Υπάρχει περίπτωση όμως να είναι ήδη ενεργοποιημένη κάποια εφαρμογή, από προηγούμενη χρήση του λογισμικού, οπότε πρώτα την απενεργοποιούμε πατώντας το μπουτόν “■” και δημιουργούμε τη καινούρια μας εφαρμογή (εικόνα 3.2). Έπειτα, πηγαίνουμε στο τμήμα Computer και κάνουμε διπλό κλικ, οπότε αναδύεται ένα νέο παράθυρο, στο οποίο κάνοντας αριστερό κλικ στο μπουτόν “Properties”, πηγαίνουμε σε ένα άλλο παράθυρο το οποίο περιλαμβάνει όλες τις ιδιότητες του WinCC Runtime, δηλαδή της κατάστασης κατά την οποία «τρέχει» η η εφαρμογή μας (εικόνα 3.3). Η κατάσταση αυτή αφορά ποικίλλες παραμέτρους, σχετικές με εμφάνιση των εικόνων εργασίας στην οθόνη του υπολογιστή, ρύθμιση της ανάλυσης των εικόνων αυτών, καθορισμό τρόπων εξόδου από αυτή και άλλα πολλά.



Εικόνα 3.2. Η εφαρμογή μας έτοιμη για ρύθμιση των βασικών παραμέτρων της.



Εικόνα 3.3. Ρύθμιση των ιδιοτήτων του WinCC Runtime.

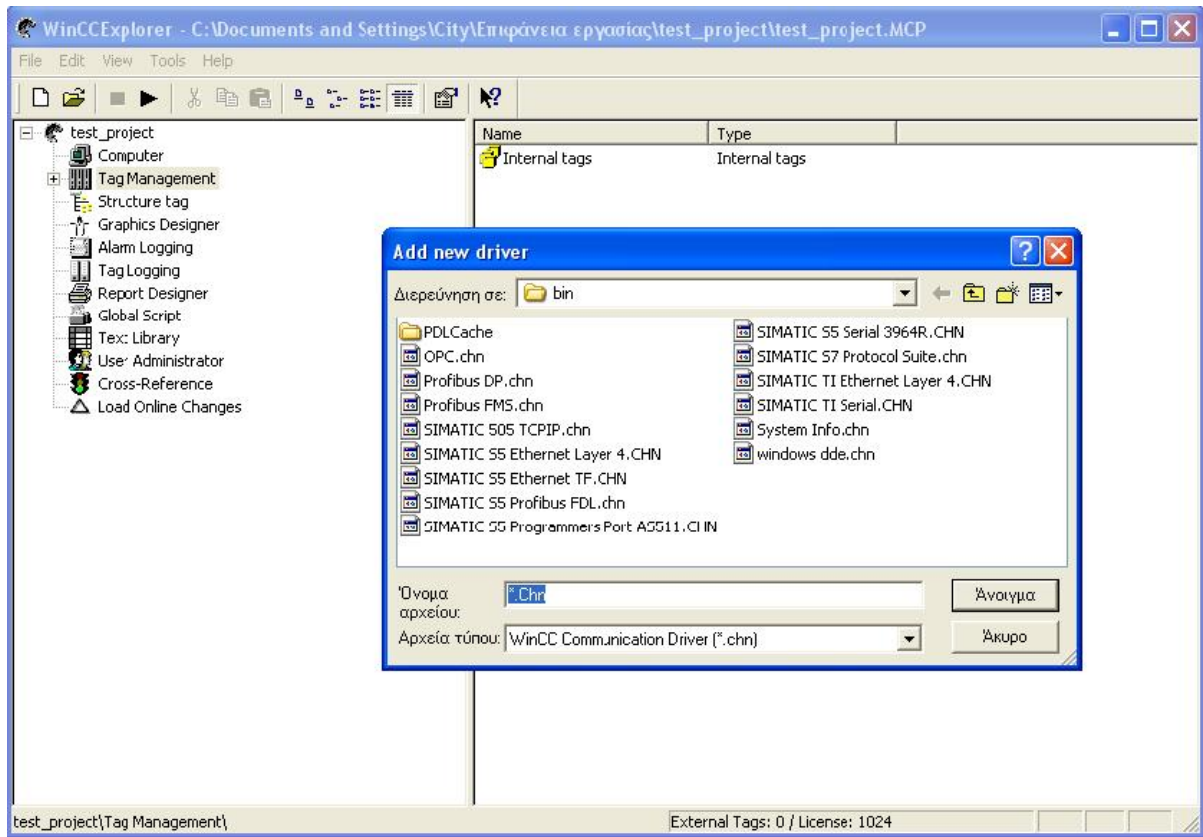
Στο παράθυρο το οποίο βρισκόμαστε τώρα κάνουμε δύο σημαντικές ρυθμίσεις, οι οποίες μας ήταν ιδιαίτερα χρήσιμες και τις εφαρμόσαμε σε όλες τις εφαρμογές που υλοποιήσαμε. Συγκεκριμένα, όταν ενεργοποιούσαμε μία εφαρμογή, είχαμε δύο βασικά προβλήματα κατά την εμφάνισή της. Η εικόνα του περιβάλλοντος εργασίας δεν κάλυπτε όλη την οθόνη του υπολογιστή, με αποτέλεσμα να υπάρχουν κενά πέρα από τα όριά της, δημιουργώντας ένα άκομφο οπτικό αποτέλεσμα. Το άλλο πρόβλημα ήταν ότι ενώ δημιουργούσαμε κάποια εικόνα με αντικείμενα αυστηρά χωροθετημένα και οργανωμένα σε ομάδες της επιλογής μας, μόλις «τρέχαμε» την εφαρμογή ο υπολογιστής τα εμφάνιζε με ακανόνιστη μορφή και τυχαία σειρά.

Τα προβλήματα αυτά λύθηκαν με τρόπο που δείχνει η εικόνα 3.3. Στο πλαίσιο “Window Attributes” επιλέξαμε “Full Screen” και “Adapt Picture”. Με τον τρόπο αυτό, η πρώτη επιλογή εφαρμόζει την εικόνα μας σε όλη την οθόνη, χωρίς κενά και ασυμμετρίες, ενώ η δεύτερη διατηρεί την δομή και την ομαδοποίηση των αντικειμένων, όπως ακριβώς έχουμε ορίσει. Έτσι, επιτεύχθηκε ένα ικανοποιητικό οπτικό αποτέλεσμα και παράλληλα ένα αποδοτικό περιβάλλον εργασίας.

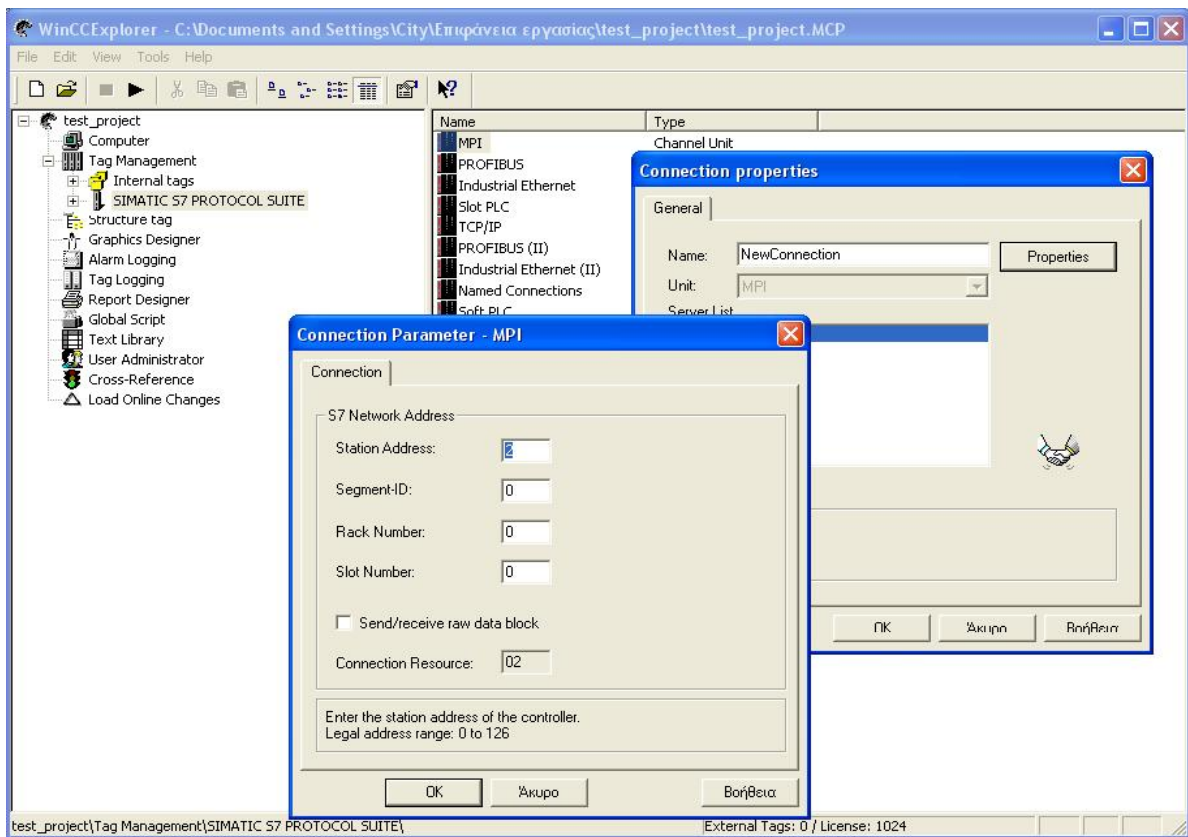
3.2.4 Επιλογή και εγκατάσταση ενός driver.

Από τα πιο βασικά βήματα τα οποία καλούμαστε να ακολουθήσουμε, μιας και εδώ εγκαθιστούμε τον τρόπο επικοινωνίας του PLC μας με τον υπολογιστή και το WinCC. Όπως γίνεται εύκολα αντιληπτό, αν δεν εισάγουμε τις σωστές παραμέτρους τότε η επικοινωνία δε θα επιτευχθεί. Επομένως, οφείλουμε να γνωρίζουμε το μοντέλο του PLC και το πρωτόκολλο επικοινωνίας, καθώς και την ακριβή διεύθυνση ράγας (rack) και θέση ράγας (slot) στην οποία βρίσκεται το PLC. Στην παρούσα εφαρμογή, επικοινωνία επιτεύχθηκε μέσω της διεπαφής MPI, όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, οποία αφορά επικοινωνία μεταξύ υπολογιστή και PLC μέτριας ταχύτητας, σε σχέση με βιομηχανικά πρωτόκολλα τα οποία αναφέρθηκαν, αλλά επαρκέστατης για τις εργαστηριακές μας ανάγκες.

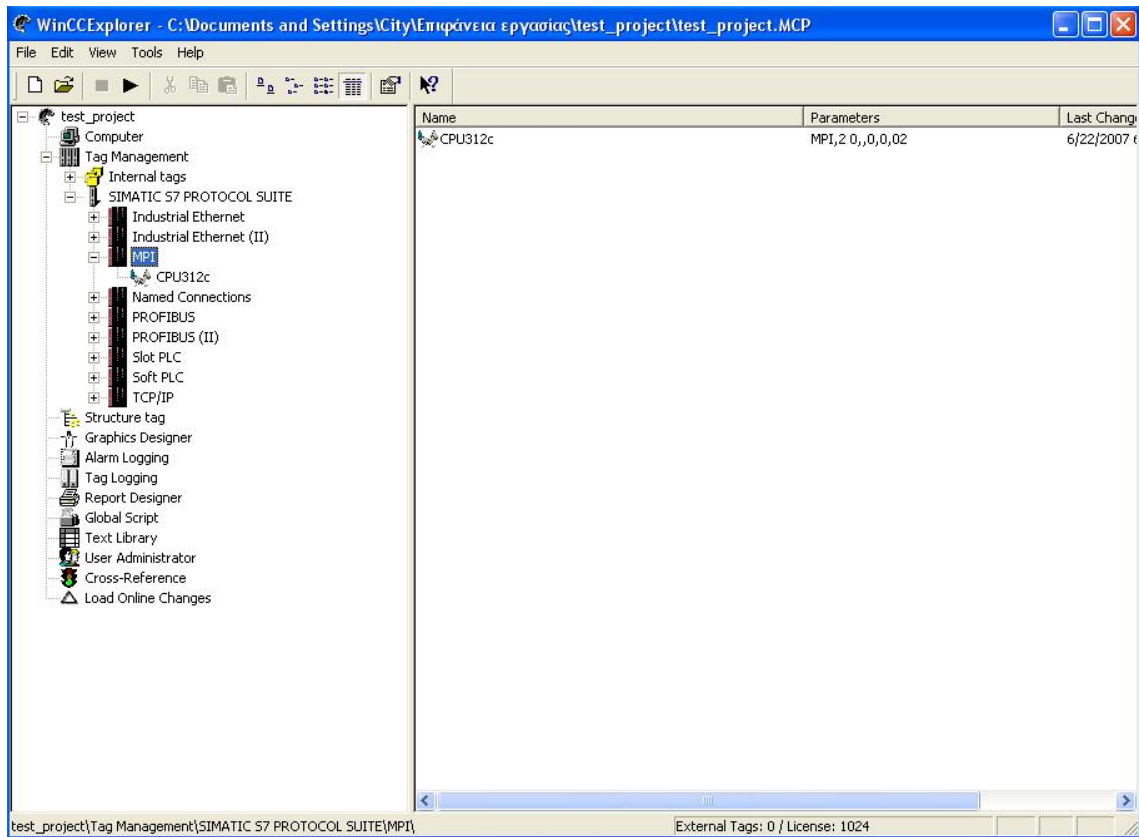
Κάνοντας δεξιά κλικ στο υποτιμήμα “Tag Management” εμφανίζεται το πλαίσιο της εικόνας 3.4, όπου επιλέγουμε “Simatic S7 Protocol Suite”. Στις επιλογές που εμφανίζονται επιλέγουμε “MPI” και έπειτα “New Driver Connection”, με δεξιά κλικ πάνω στην επιλογή αυτή. Ονομάζουμε την επικοινωνία “CPU312c” και έπειτα εισάγουμε τις κατάλληλες παραμέτρους, όπως δείχνουν οι εικόνες 3.5 και 3.6.



Εικόνα 3.4. Εγκαθιστώντας την επικοινωνία υπολογιστή – PLC.



Εικόνα 3.5. Ρύθμιση παραμέτρων επικοινωνίας μέσω MPI.

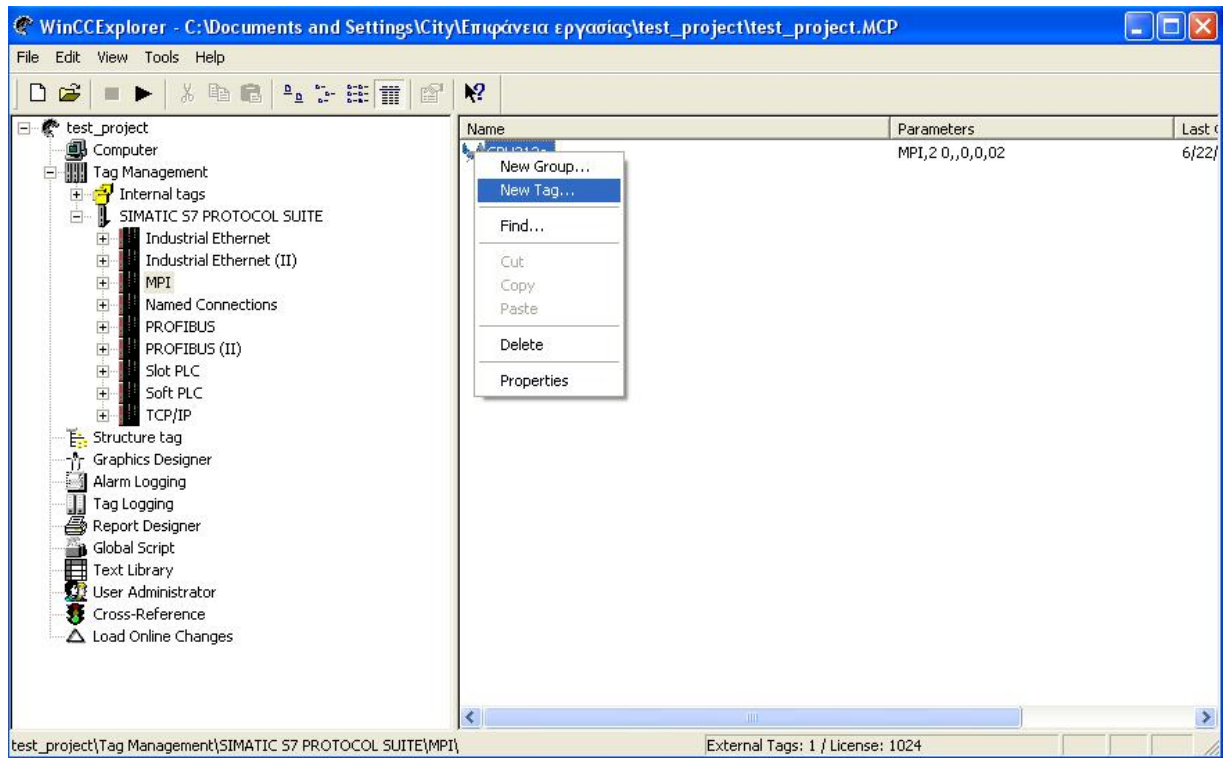


Εικόνα 3.6. Δημιουργία επικοινωνίας “CPU312c”.

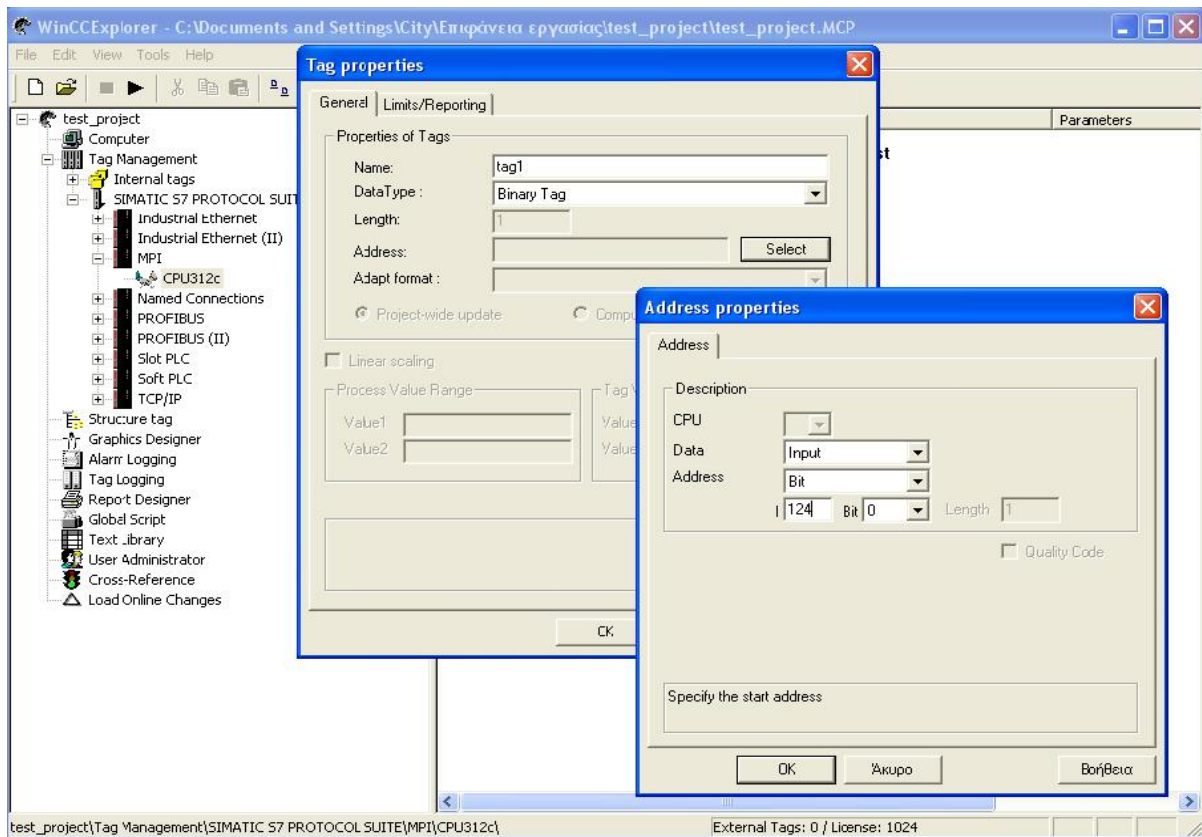
3.2.5 Ορισμός των μεταβλητών (tags).

Το επόμενο βήμα και εξίσου σημαντικό με την αποκατάσταση της επικοινωνίας είναι η δημιουργία των κατάλληλων εξωτερικών (external tags) – εσωτερικών (internal tags) μεταβλητών. Οι εξωτερικές μεταβλητές χρησιμοποιούνται για να μεταφέρουν τιμές από την περιοχή δεδομένων μνήμης του PLC (ή του RTU) στο WinCC, όπου γίνεται η επεξεργασία και εποπτεία τους. Οι εσωτερικές μεταβλητές αφορούν κάποια τμήματα της μνήμης του WinCC όμως, τα οποία χρησιμοποιούνται για εσωτερικά θέματα του λογισμικού και για λειτουργίες προσομοίωσης. Ο ορισμός και των δύο τύπων μεταβλητών είναι σημαντικός για σωστή αναπαράσταση των παρατηρούμενων διεργασιών και παίζει σημαντικό ρόλο στη σωστή λειτουργία του SCADA.

Για τη δημιουργία των εξωτερικών μεταβλητών, πηγαίνουμε στο “Tag Management” και διαδοχικά με διπλό κλικ “SIMATIC PROTOCOL SUITE” > “MPI” > “CPU312c”. Εκεί με δεξιά κλικ επιλέγουμε “New Tag...” (εικόνα 3.7) και εμφανίζεται το πλαίσιο της εικόνας 3.8. Στα πεδία τα οποία εμφανίζονται δημιουργούμε την μεταβλητή “Tag1”, η οποία είναι διασυνδεδεμένη με την ψηφιακή είσοδο I124.0.

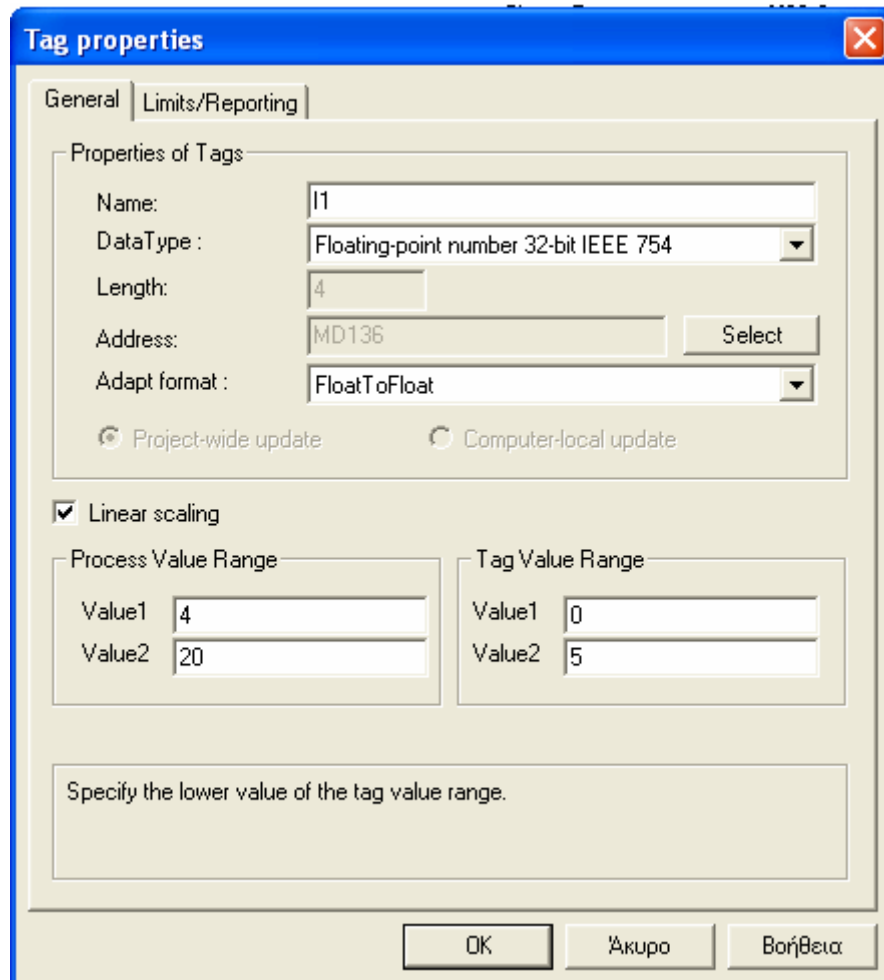


Εικόνα 3.6. Δημιουργώντας μία εξωτερική μεταβλητή.



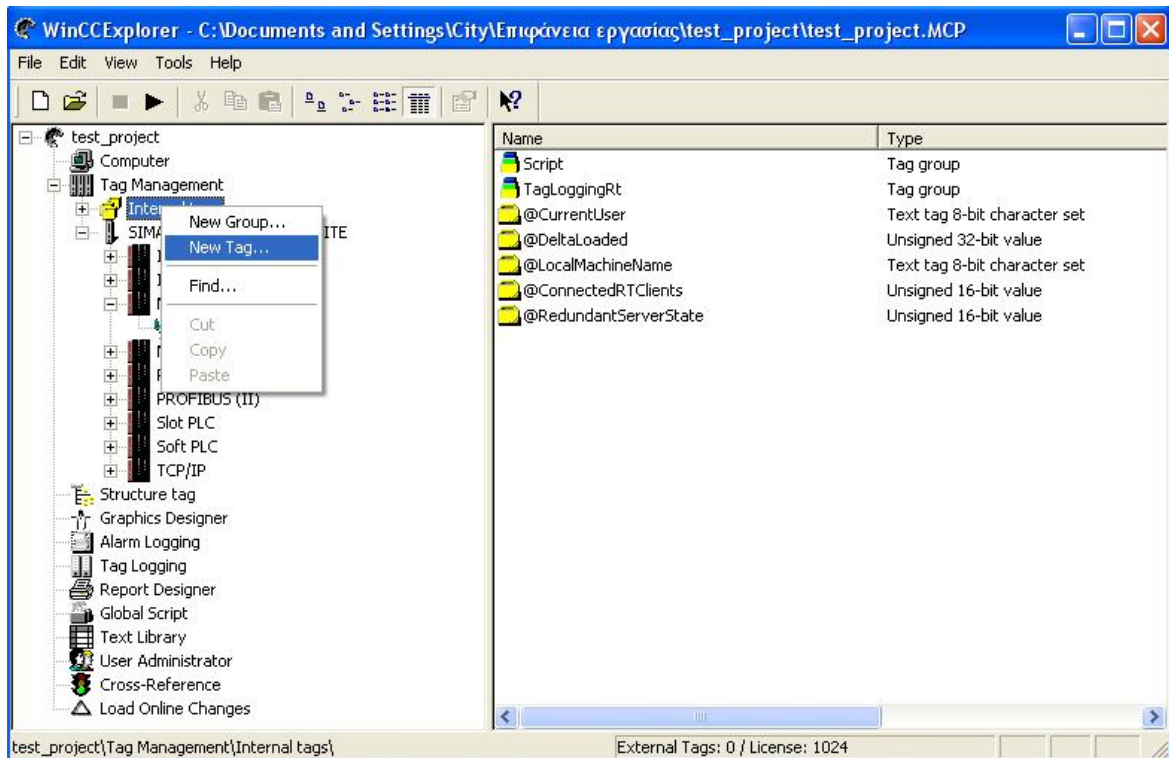
Εικόνα 3.7. Ρυθμίζοντας τις παραμέτρους της εξωτερικής μεταβλητής.

Παρατηρούμε ότι υπάρχει πληθώρα επιλογών για ορισμό μεταβλητών όλων των απαιτήσεων, όπως και αναπαράστασης αναλογικών μεγεθών. Γι' αυτά τα μεγέθη υπάρχει, επιπλέον, η δυνατότητα της γραμμικής κλίμακας, όπως δείχνει το παράθυρο της εικόνας 3.8. Έστω ότι έχουμε έναν μορφοτροπέα ρεύματος 0 – 5 A με έξοδο 4 – 20 mA. Τότε συμπληρώνουμε τα πεδία όπως δείχνει η εικόνα και ορίζουμε κατάλληλα την μεταβλητή μας σαν αριθμό κινητής υποδιαστολής, αριθμό δηλαδή ο οποίος αναπαριστά πραγματικό αριθμό στη μνήμη του υπολογιστή.

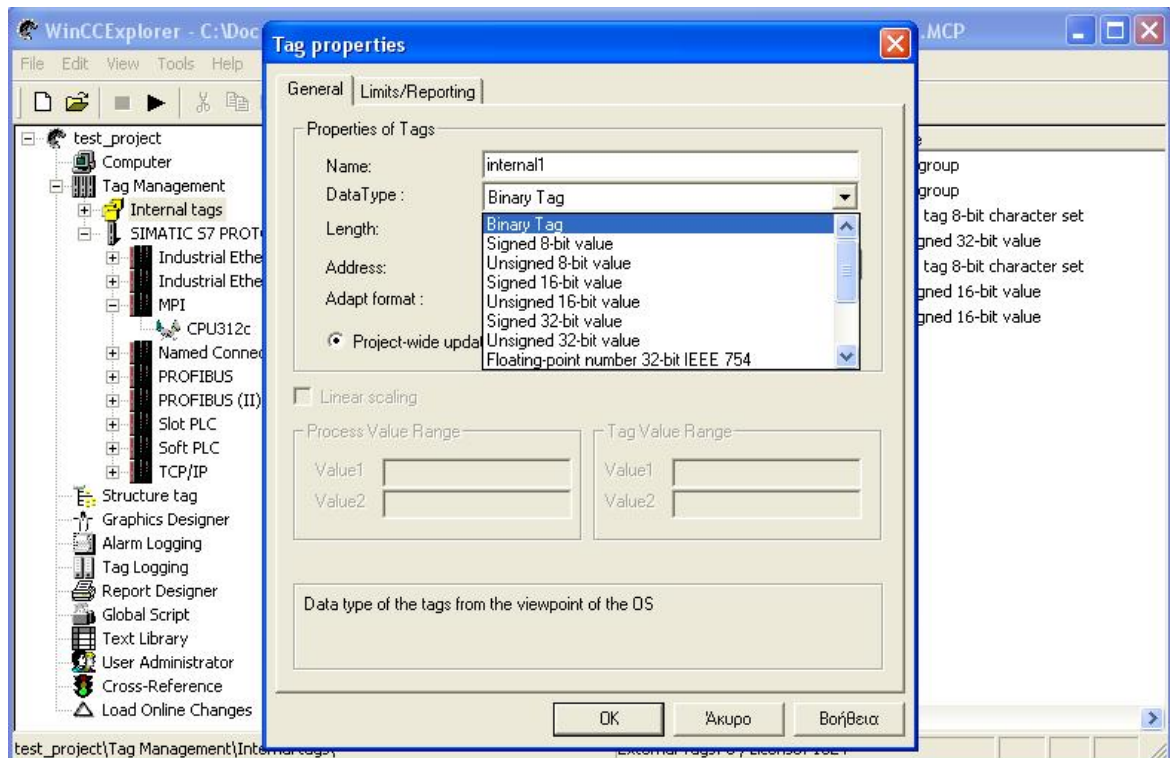


Εικόνα 3.8. Γραμμική κλίμακα αναλογικού μεγέθους.

Στην περίπτωση των εσωτερικών μεταβλητών, μεταφερόμαστε στην επιλογή “Internal tags” και επιλέγουμε “New Tag...” (εικόνα 3.9). Εκεί δημιουργούμε την εσωτερική μεταβλητή “internal1” (εικόνα 3.10). Το πλεονέκτημα το οποίο παρουσιάζουν οι εσωτερικές μεταβλητές είναι ότι αντιγράφονται και μεταφέρονται, σε αντίθεση με τις εξωτερικές οι οποίες είναι στάσιμες. Αυτό το εκμεταλλευτήκαμε για τη μεταφορά εσωτερικών μεταβλητών, κατά τη δημιουργία του εικονικού προσομοιωτή, στο κεφάλαιο 4.



Εικόνα 3.9. Δημιουργώντας εσωτερική μεταβλητή.

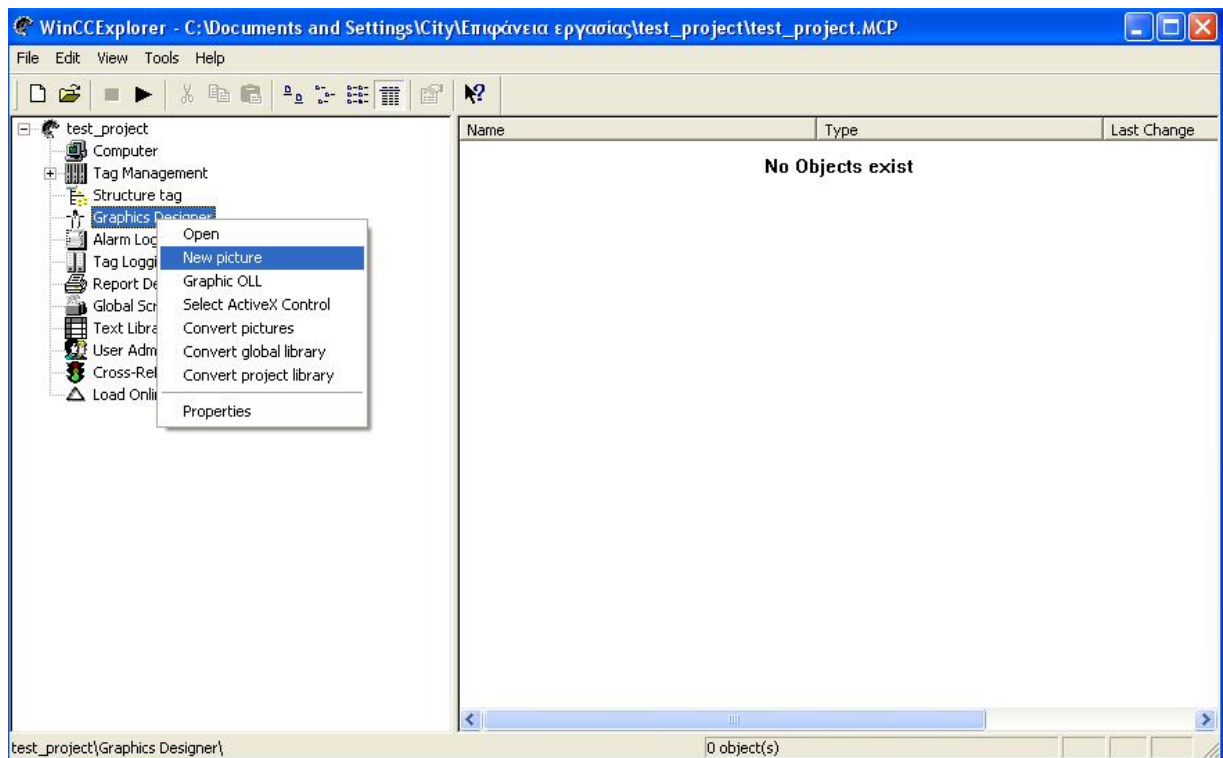


Εικόνα 3.10. Δημιουργώντας εσωτερική μεταβλητή.

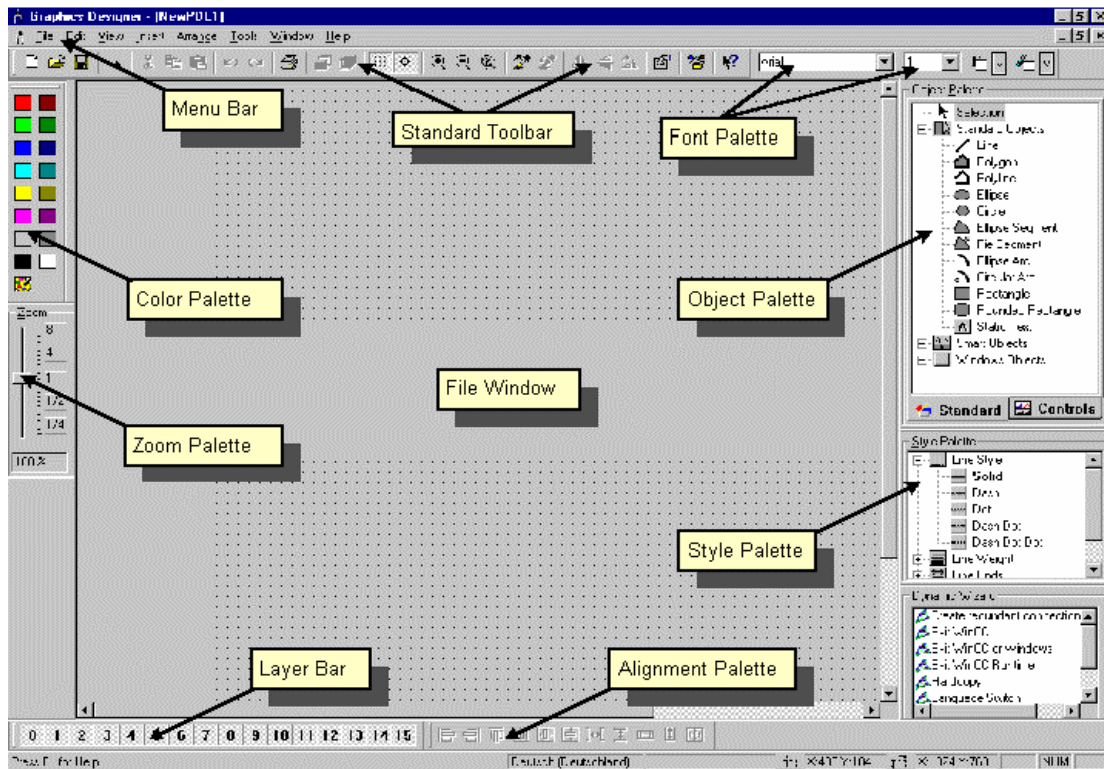
Αξίζει να σημειωθεί ότι τόσο οι εσωτερικές όσο και οι εξωτερικές μεταβλητές μπορούν να ομαδοποιηθούν σε ομάδες μεταβλητών, τα Tag Groups. Δεν αλλάζει κάτι λειτουργικά με την κίνηση αυτή, απλά μας προσφέρει καλύτερη οργάνωση και ευκολότερη αναζήτηση, κάτι το οποίο έχει βαρύνουσα σημασία σε αρκετά πολύπλοκα και μακροσκελή συστήματα. Η δημιουργία ενός Tag Group γίνεται όμοια με τη δημιουργία μίας μεταβλητής, όπου αντί για “New Tag” επιλέγεται “New Group”.

3.2.6 Δημιουργία και εμφάνιση των εικόνων της εφαρμογής.

Για τη δημιουργία του περιβάλλοντος εργασίας στην οθόνη του χρήστη, δηλαδή των κατάλληλων εικόνων, χρησιμοποιούμε τον σχεδιαστή γραφικών του WinCC, τον Graphics Designer. Όπως δείχνει η εικόνα 3.11, κάνουμε δεξιά κλικ πάνω στον σχεδιαστή και επιλέγουμε “New Picture...” οπότε μεταφερόμαστε στο περιβάλλον του σχήματος 3.1. Κατόπιν ακολουθεί μία περιγραφή των βασικών στοιχείων που αποτελούν το περιβάλλον του σχεδιαστή γραφικών.



Εικόνα 3.11. Δημιουργώντας εικόνες με τον σχεδιαστή γραφικών.



Σχήμα 3.1. Το περιβάλλον του σχεδιαστή γραφικών.

➤ Color Palette

Χρησιμοποιείται για να χρωματιστούν τα αντικείμενα που έχουν επιλεγεί. Εκτός από τα κύρια χρώματα, υπάρχει η δυνατότητα δημιουργίας ποικίλων συνδυασμών και αποθήκευσής τους από τον χρήστη.

➤ Object Palette

Περιέχει όλα τα κύρια σχήματα: Standard Objects, (όπως πολύγωνο, έλλειψη, παραλληλόγραμμο κτλ.), Smart Objects (OLE Control, OLE Element, I/O Fields κτλ.) και Windows Objects (Button, Check Box κτλ.).Γενικά περιέχει βιβλιοθήκη βασικών αντικειμένων τα οποία επιτελούν βασικές λειτουργίες σε μία εφαρμογή.

➤ Style Palette

Αλλάζει την εμφάνιση του επιλεγμένου αντικειμένου. Ανάλογα με τον τύπο του αντικειμένου, μπορεί να αλλαχθεί ο τύπος της γραμμής ή του πλαισίου, το πλάτος της γραμμής ή του πλαισίου, το "στυλ" (είδος) του τέλους της γραμμής ή το "γέμισμα" (fill pattern).

➤ Alignment Palette

Επιτρέπει την αλλαγή της απόλυτης θέσης ενός ή περισσότερων αντικειμένων, την αλλαγή της θέσης επιλεγμένου αντικειμένου σε σχέση με κάποιο άλλο, ή τον καθορισμό του ύψους και του πλάτους κάποιων αντικειμένων.

➤ Zoom Palette

Καθορίζει τον συντελεστή εστίασης (zoom) για το ενεργοποιημένο παράθυρο.

➤ Menu Bar

Περιέχει όλες τις μεταβλητές του μενού για τον Graphics Designer. Οι εντολές που δεν είναι την τρέχουσα στιγμή διαθέσιμες εμφανίζονται με "γκρι" χρώμα.

➤ Toolbar

Περιέχει κουμπιά για τη γρήγορη και εύκολη χρήση των πιο συνηθισμένων εντολών.

➤ Font Palette

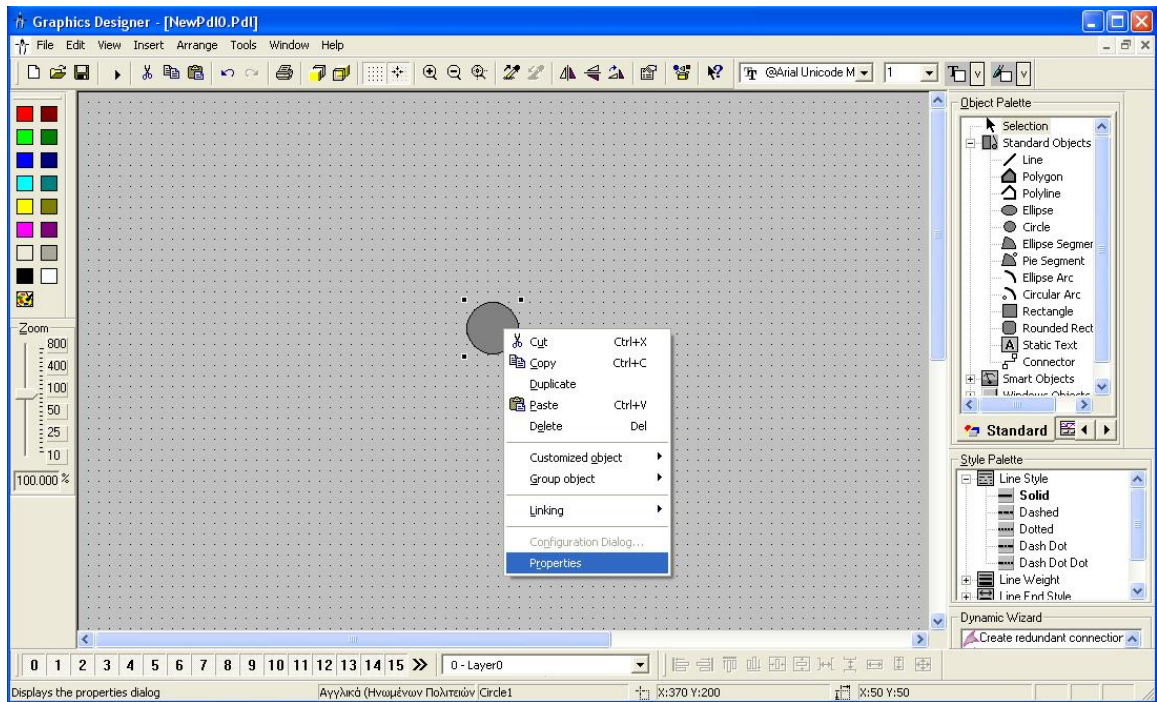
Επιτρέπει την αλλαγή του τύπου, του μεγέθους και του χρώματος των γραμμάτων σε αντικείμενα κειμένου (text objects), όπως και του χρώματος της γραμμής των κύριων αντικειμένων.

➤ Layer Bar

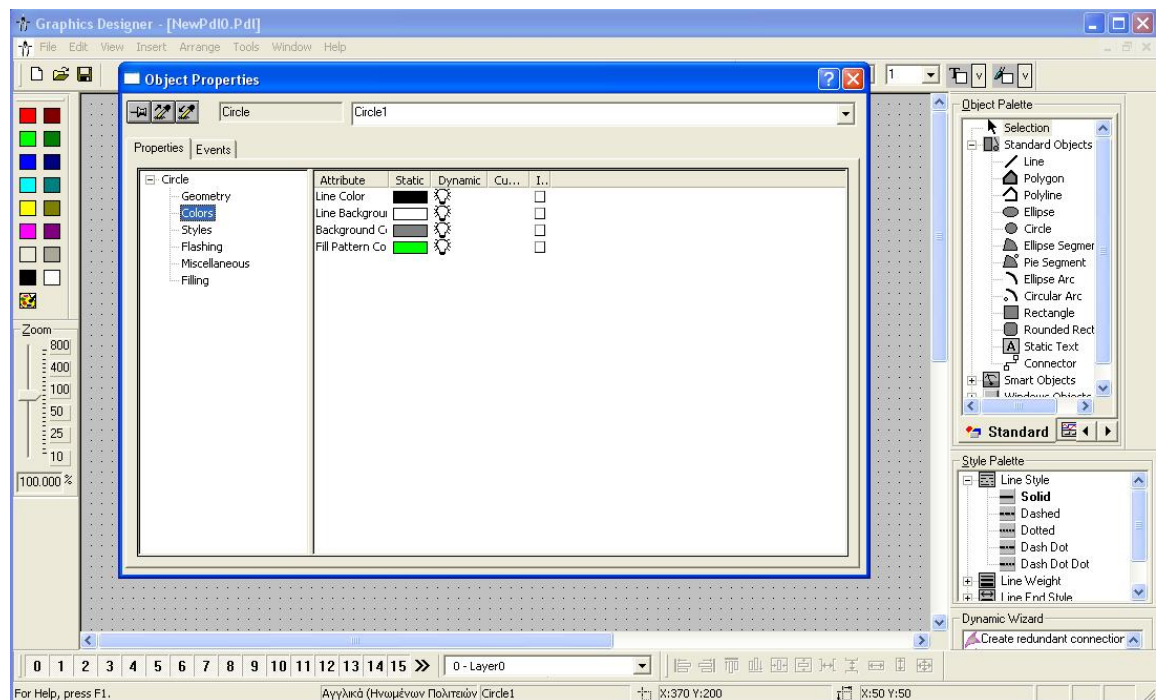
Χρησιμοποιείται για να επιλεγεί ποιο από τα 16 layers (layer 0 έως 15) θα είναι ορατό. Το layer 0 είναι η επιλογή default.

Μία πολύ σημαντική παράμετρος την οποία παρέχει ο σχεδιαστής είναι η δυναμική διασύνδεση ενός ή περισσότερων αντικειμένων, ώστε να μεταβάλλονται οι ιδιότητές τους, σύμφωνα με κάποια μεταβλητή ή να αναπαριστούν τιμές διαφόρων μεταβλητών. Για να γίνει αντιληπτή η διαδικασία αυτή θα παρουσιάσουμε ένα παράδειγμα διασύνδεσης ενός κύκλου ο οποίος, με κατάλληλες μεταβολές, δύναται να αναπαραστήσει μία μεταβλητή εισόδου.

Εισάγουμε έναν κύκλο από το "Object Palette" με διπλό κλικ πάνω του και κάνοντας δεξιά κλικ επιλέγουμε "Properties" (εικόνα 3.12). Με τον τρόπο αυτό αναδύεται ένα παράθυρο, το οποίο περικλείει όλες τις ιδιότητες του συγκεκριμένου αντικειμένου (εικόνα 3.13). Εμείς θέλουμε να διασυνδέσουμε δυναμικά το χρώμα του κυκλικού δίσκου με τη μεταβλητή εξόδου "tag1", την οποία δημιουργήσαμε πριν.



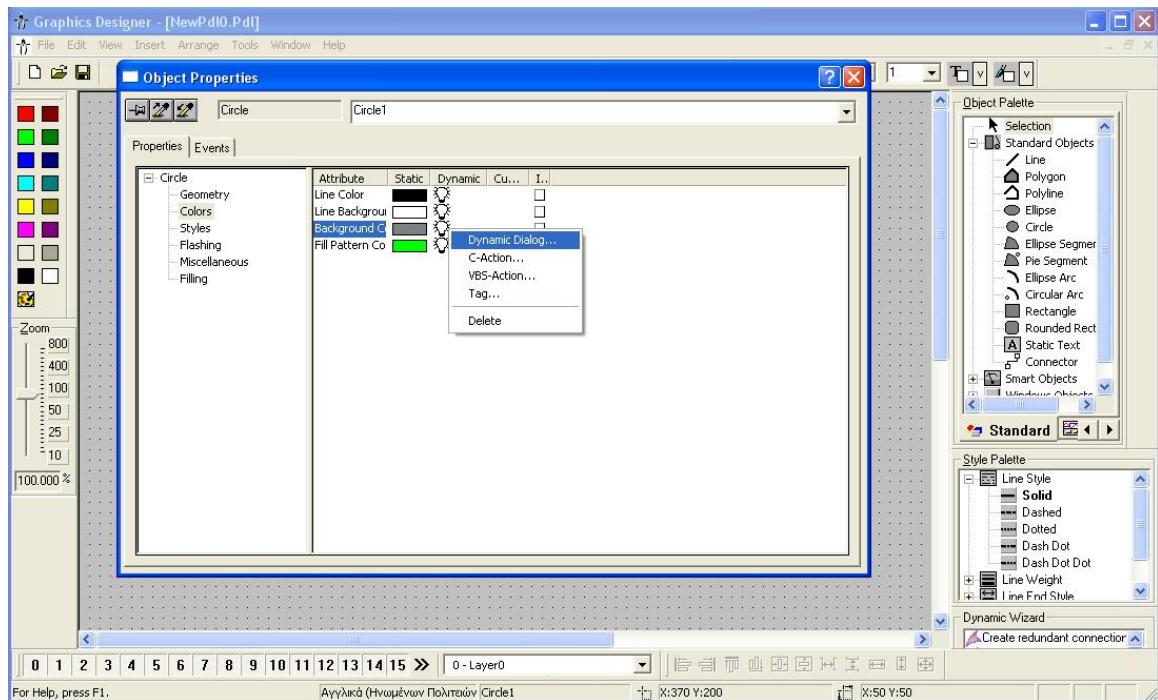
Εικόνα 3.12. Εισαγωγή κύκλου και προσπέλαση στο παράθυρο ιδιοτήτων του.



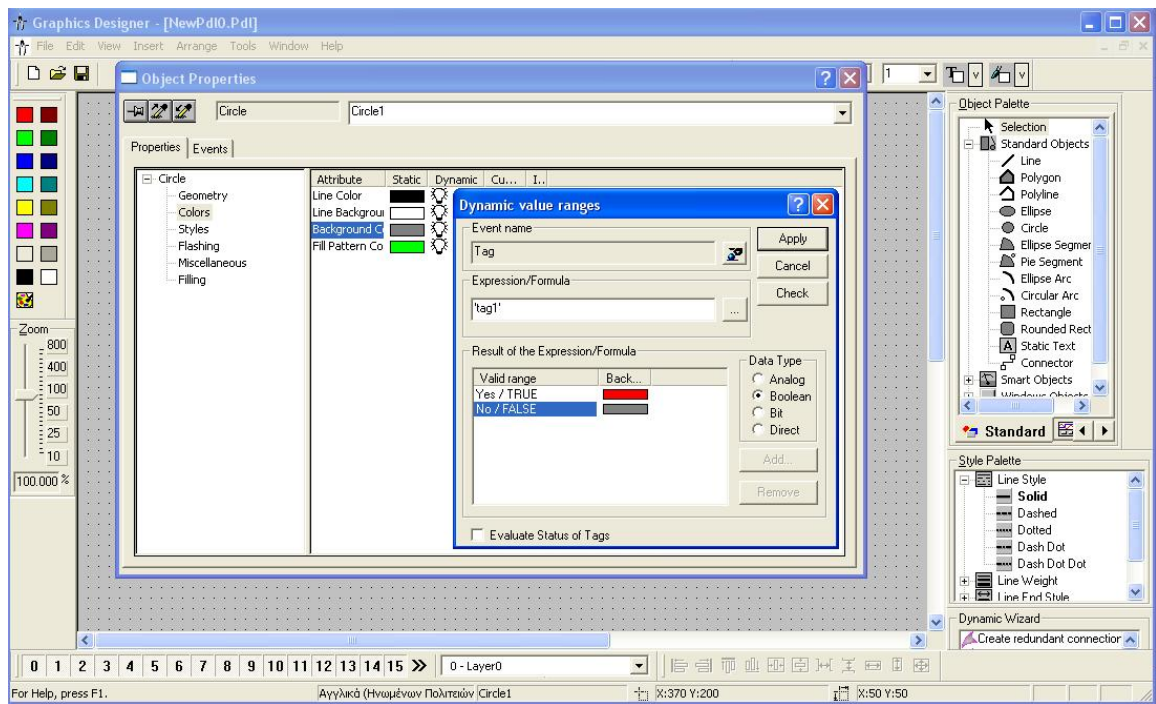
Εικόνα 3.13. Το παράθυρο ιδιοτήτων του κύκλου.

Πηγαίνουμε στην επιλογή “Colors” της καρτέλας “Properties” και δίπλα στο εικονίδιο της επιλογής “Background Color” κάνουμε δεξιά κλικ, όπως δείχνει η εικόνα 3.14, όπου επιλέγουμε “Dynamic Dialog”. Στο νέο παράθυρο που αναδύεται έχουμε τη δυνατότητα να διασυνδέσουμε δυναμικά την ιδιότητα του χρώματος του κυκλικού δίσκου με την μεταβλητή της αρεσκείας μας. Για του λόγου το αληθές, κάνοντας τις επιλογές που δείχνει η εικόνα 3.15, διασυνδέσαμε τη

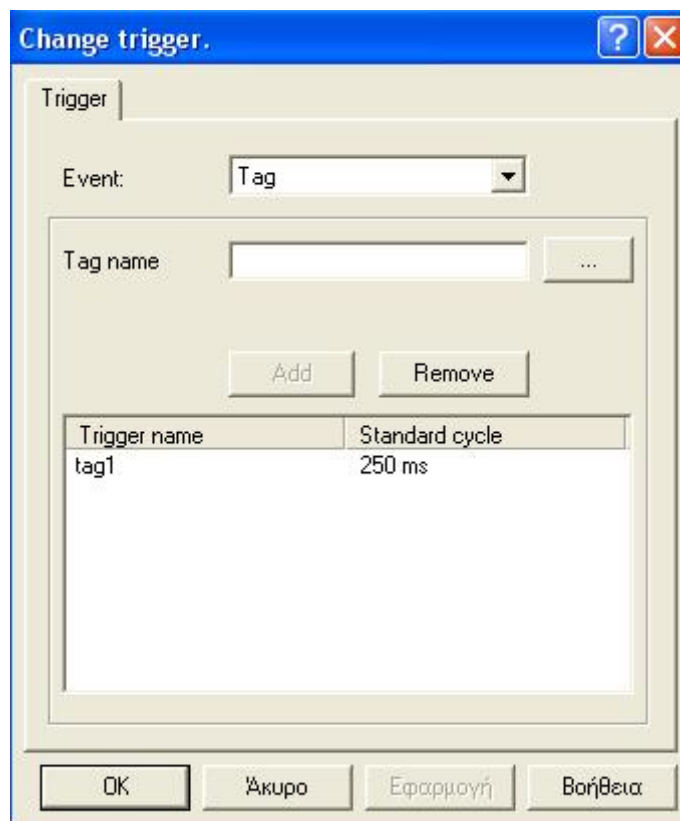
μεταβλητή “tag1” και επιλέξαμε τα κατάλληλα χρώματα. Αξίζει να σημειωθεί ότι αν πάμε στο πλαίσιο “Event name” και πατήσουμε το ανάλογο μπουτόν, καταλήγουμε στο παράθυρο της εικόνας 3.16, όπου κάτω αναγράφεται αριστερά το όνομα της μεταβλητής και δεξιά “Standard cycle”. Η επιλογή αυτή αφορά τη συχνότητα που θέλουμε το WinCC να ενημερώνεται για την τιμή της μεταβλητής και μεταβάλλεται με δεξιά κλικ πάνω της, όπου εμφανίζεται ένας κατάλογος διαφόρων συχνοτήτων. Αυτό είναι ιδιαίτερα σημαντικό σε περιπτώσεις όπου οι μεταβλητές θέλουμε να ενημερώνονται άμεσα και σε περιπτώσεις που δεν έχει και τόση σημασία η συχνότητα ενημέρωσή τους, οπότε μπορούμε να επιλέξουμε κάποια μέτρια συχνότητα και να αποφορτίσουμε τον υπολογιστή. Εδώ απλά επιλέγουμε “250 ms” που γενικά μας καλύπτει, ενώ κατά την πορεία της εργασίας αργότερα, επιλέξαμε και “Upon change”, όταν απαιτούσαμε αναπαράσταση τιμών σε πραγματικό χρόνο και αφορά σε ενημέρωση σχεδόν στιγμιαίας χρονικής διάρκειας.



Εικόνα 3.14. Διασυνδέοντας δυναμικά το χρώμα του κυκλικού δίσκου.



Εικόνα 3.15. Ρυθμίσεις παραμέτρων δυναμικής διασύνδεσης.



Εικόνα 3.16. Ρυθμίζοντας τη συχνότητα ενημέρωσης της τιμής της μεταβλητής.

Τελικά, αυτό που επιτύχαμε είναι ο κυκλικός δίσκος να είναι χρώματος γκρι, όταν η μεταβλητή μας είναι στο λογικό-“0” (FALSE) και να γίνεται κόκκινος όταν η μεταβλητή πάει στο λογικό-“1” (TRUE). Έτσι, μπορεί η διαδικασία αυτή να φαίνεται πολύπλοκη, στην πραγματικότητα όμως είναι αρκετά απλή και μας επιτρέπει να προσομοιώσουμε τη λειτουργία LEDs εισόδου-εξόδου και γενικά να προχωρήσουμε σε πιο ειδικές λειτουργίες, όπως θα δούμε και στην πορεία της εργασίας.

3.2.7 Καθορισμός των ιδιοτήτων του WinCC Runtime.

Το τελευταίο βήμα αφορά στη ρύθμιση των ιδιοτήτων που λαμβάνει χώρα στο υποτιμήμα “Computer”, τις οποίες αναφέραμε αρχικά. Γενικά, το βήμα αυτό μπορεί να γίνει είτε στην αρχή είτε στο τέλος της διαδικασίας, χωρίς να υπάχουν περιορισμοί. Όλες οι ρυθμίσεις που αφορούν τον τρόπο παρουσίασης, από τον ορισμό της αρχικής εικόνας μέχρι και πιο σύνθετα θέματα, γίνονται εδώ καταδεικνύοντας τη σημασία του συγκεκριμένου βήματος.

Ακολουθώντας τα παραπάνω βήματα, δημιουργήσαμε τη δοκιμαστική μας εφαρμογή. Το μόνο που απομένει είναι να την ενεργοποιήσουμε από το μενού του Editor, πατώντας “▶”. Την διαδικασία αυτή ακολουθήσαμε προκειμένου να υλοποιήσουμε τις ζητούμενες εφαρμογές της εργασίας μας, με κάποιες αλλαγές προχωρημένου επιπέδου, οι οποίες καταδεικνύονται και επεξηγούνται στα επόμενα κεφάλαια.

ΚΕΦΑΛΑΙΟ 4

Ο Προσομοιωτής ως εφαρμογή WinCC

4.1 Γενικά.

Σε όλα τα σενάρια εφαρμογών, τα οποία υλοποιήσαμε στην παρούσα εργασία, ήταν επιτακτική η δημιουργία ενός περιβάλλοντος που προσομοιάζει τον ίδιο τον προσομοιωτή ως προς τις λειτουργίες του, για εκπαιδευτικούς λόγους. Πρόκειται για έναν εικονικό προσομοιωτή που παρουσιάζει ακριβώς την ίδια λειτουργία με τον πραγματικό, από κάθε άποψη. Για ευκολία, δημιουργήσαμε το ζητούμενο περιβάλλον μία φορά και με διαδοχικές αντιγραφές το μεταφέραμε σε κάθε εφαρμογή, εκμεταλλευόμενοι πλήρως αυτήν την ευελιξία του WinCC. Παρακάτω ακολουθεί μία διεξοδική παρουσίαση του τρόπου δημιουργίας αλλά και του αποτελέσματος του ζητούμενου περιβάλλοντος.

4.2 Δημιουργία εφαρμογής.

4.2.1 Περιγραφή.

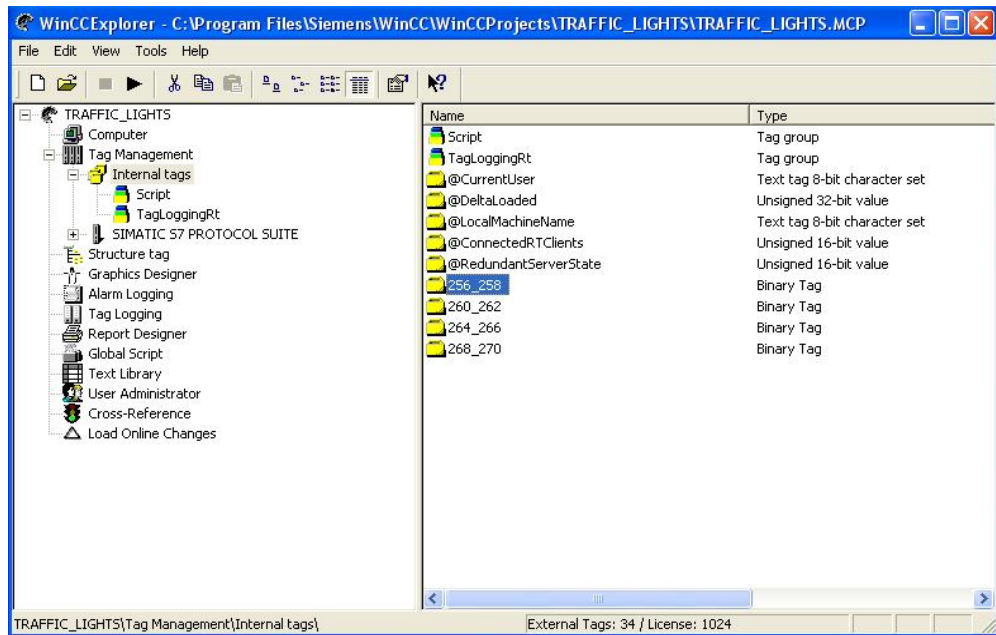
Προτού ξεκινήσουμε την περιγραφή του τρόπου εργασίας μας, πρέπει να τονιστεί ότι αναφερόμαστε όχι σε μία ολοκληρωμένη εφαρμογή, όπως ορίστηκε στα προηγούμενα κεφάλαια, αλλά στη δημιουργία ενός απλού περιβάλλοντος εργασίας, δηλαδή μίας εικόνας. Η εικόνα αυτή του σχεδιαστή γραφικών αντιγράφηκε σε κάθε μία από τις ολοκληρωμένες μας εφαρμογές σαν απλό αρχείο και κάθε φορά γινόταν τροποποίησή της, αναλόγως την εφαρμογή. Μιλώντας για τροποποίηση αναφερόμαστε στη δυναμική διασύνδεση διαφορετικών εξωτερικών μεταβλητών και γενικά σε τυχόν προσθήκες επιπλέον αντικειμένων για καλύτερη λειτουργικότητα.

4.2.2 Ορισμός μεταβλητών.

Στην παρούσα διαδικασία δεν προχωρήσαμε σε ορισμό εξωτερικών μεταβλητών, όπως προαναφέρθηκε, αλλά δημιουργήσαμε κατάλληλες εσωτερικές μεταβλητές, κάτι το οποίο και παρουσιάζει μεγάλο πρακτικό ενδιαφέρον, αλλά και μας έδωσε τη δυνατότητα να κατανοήσουμε πλήρως αυτόν τον ιδιαίτερο τύπο μεταβλητών.

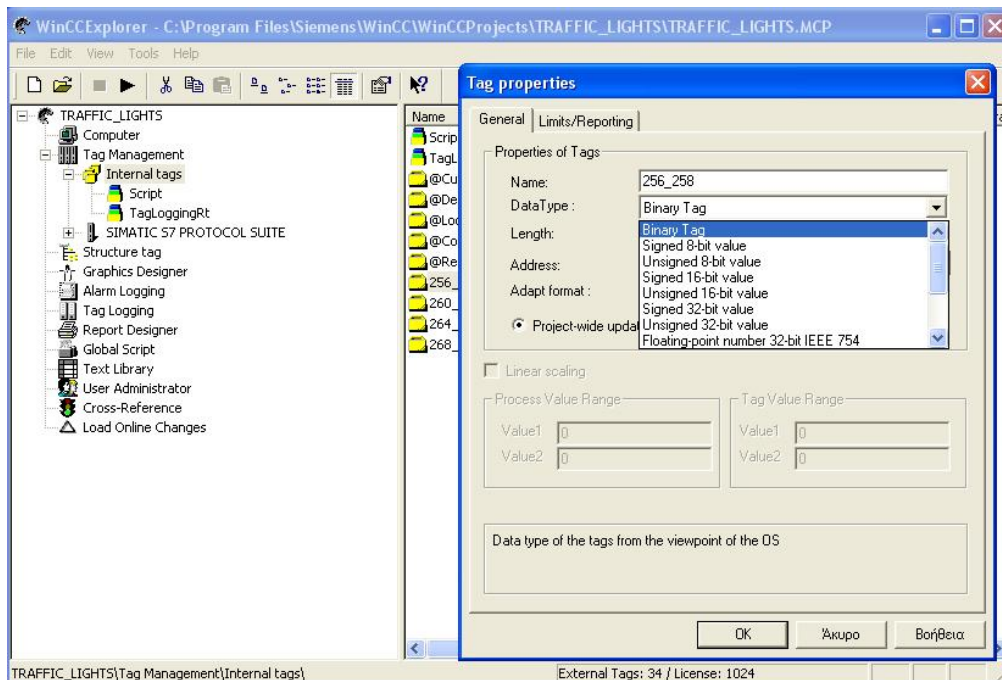
Ακολουθώντας τον τρόπο που περιγράφηκε στο κεφάλαιο 2 ορίσαμε τις εξής εσωτερικές μεταβλητές (εικόνα 4.1) :

- 256_258
- 260_262
- 264_266
- 268_270



Εικόνα 4.1. Οι εσωτερικές μεταβλητές της εφαρμογής.

Οι παραπάνω μεταβλητές είναι τύπου “binary tag”, δηλαδή δυαδικής λογικής όπως δείχνει και η εικόνα 4.2.



Εικόνα 4.2. Ο τύπος των εσωτερικών μεταβλητών.

Η λειτουργία την οποία επιτελούν οι μεταβλητές αυτές είναι πράγματι πολύτιμη. Θέλοντας να αναπαραστήσουμε ακριβώς τον προσομοιωτή μας σε εικονικό περιβάλλον, αντιμετωπίσαμε

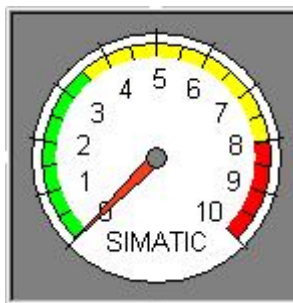
τη δυσκολία αναπαράστασης όλων των διευθύνσεων της αναλογικής κάρτας. Στην πραγματικότητα, έχοντας τους μεταγωγικούς διακόπτες τεσσάρων θέσεων κάθε φορά, επιλέγαμε ποια αναλογική διεύθυνση το σήμα θα απεικόνιζε το εκάστοτε βολτόμετρο. Για παράδειγμα, όταν ο μεταγωγικός διακόπτης B ήταν στη θέση 1 (σχήμα 1.2) το βολτόμετρο B απεικόνιζε το σήμα της διεύθυνσης *PIW256* κοκ.

Δυστυχώς τέτοιοι αυτούσιοι μεταγωγικοί διακόπτες δεν υποστηρίζονται από τον σχεδιαστή γραφικών και έπρεπε με κάποιο τρόπο να επιλέξουμε μία παρεμφερή σχεδίαση που να πλησιάζει κατά πολύ την πραγματικότητα. Δηλαδή, εργαζόμενοι σαν μηχανικοί να επιλέξουμε μία πρακτική λύση με το ίδιο αποτέλεσμα. Η λύση δόθηκε με επιλογή τεσσάρων κατάλληλων πιεστικών διακοπών (“Round Button”), οι οποίοι διασυνδέθηκαν δυναμικά με τις εν λόγω μεταβλητές. Έτσι, το κάθε μπουτόν καθορίζει ποια αναλογική διεύθυνση θα παριστάνεται στο εκάστοτε εικονικό μας βολτόμετρο κάθε φορά, αναλόγως αν κάνουμε αριστερό ή δεξί κλικ πάνω του. Η διαδικασία αυτή μπορεί να φαίνεται πολύπλοκη, δικαιολογημένα, σε έναν αρχάριο γνώστη του WinCC, αλλά με τη βοήθεια της λεπτομερούς περιγραφής που ακολουθεί παρακάτω, εύκολα διαπιστώνει κανείς πόσο απλή και πρακτική είναι.

4.2.3 Σχεδιαστικό περιβάλλον.

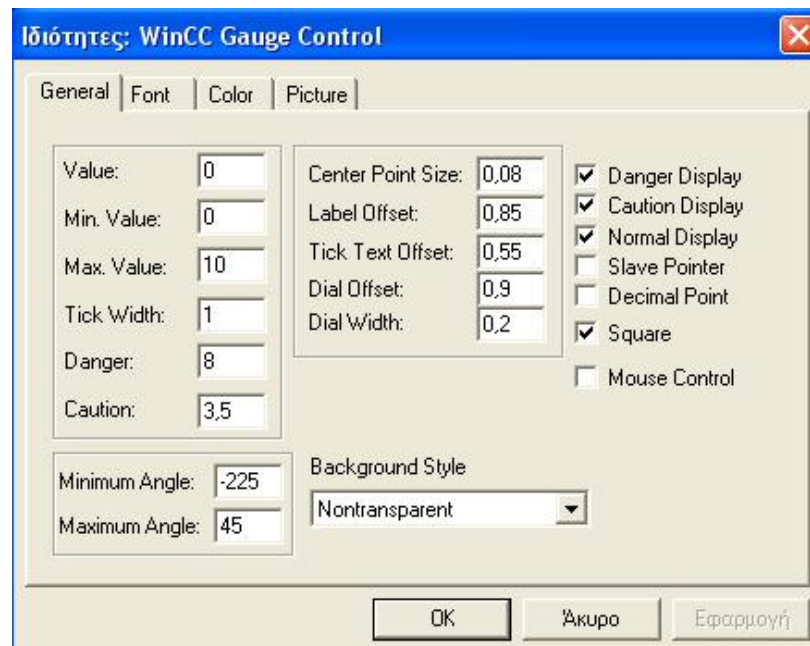
Για να δημιουργήσουμε το ζητούμενο περιβάλλον, ανοίγουμε τον σχεδιαστή γραφικών, ασχέτως του project που “τρέχει” στον Editor, μιας που μας ενδιαφέρει μόνο το αρχείο που θα δημιουργηθεί και μόνο, ώστε να είμαστε σε θέση να το αντιγράψουμε έπειτα στα projects της αρεσκείας μας. Ακολουθεί μία περιγραφή τόσο των αντικειμένων βιβλιοθήκης τα οποία χρησιμοποιήθηκαν όσο και των σχημάτων εκείνων που δημιουργήθηκαν, αλλά και των ιδιοτήτων τους εκείνων που αξιοποιήσαμε καταλλήλως.

➤ WinCC Gauge Control



Εικόνα 4.3. WinCC Gauge Control.

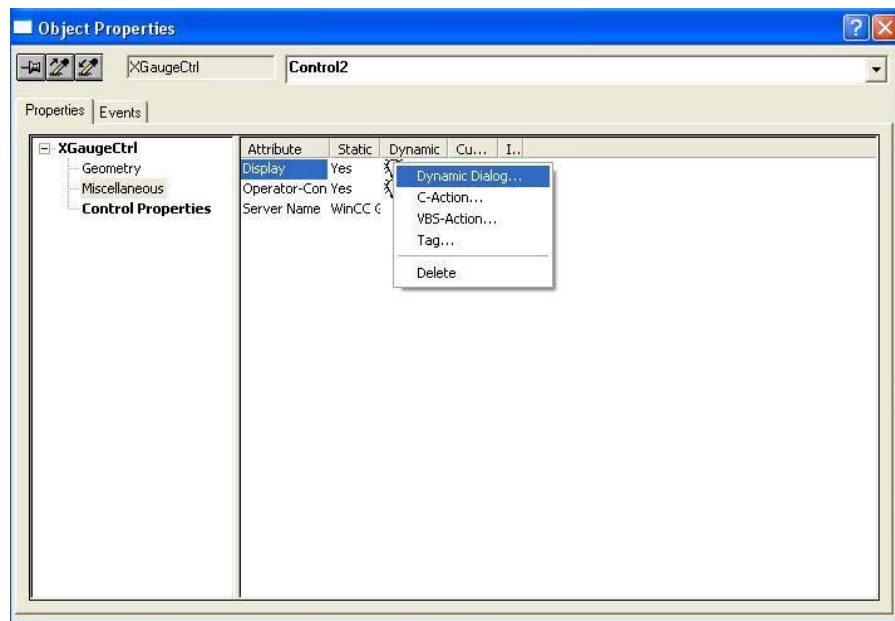
Αποτελεί αντικείμενο της βιβλιοθήκης που βρίσκεται στο “*Object Palette*” και χρησιμοποιείται για την απεικόνιση της τιμής μίας μεταβλητής. Αποτελείται από έναν κύκλο ρολογιού τον οποίο διαγράφει ένας δείκτης, ο οποίος έχει άμεση σχέση με την τιμή που καλείται να αναπαραστήσει. Κάνοντας διπλό κλικ πάνω του και αφού το έχουμε εισάγει στο περιβάλλον εργασίας, εμφανίζεται ένα παράθυρο ιδιοτήτων, το οποίο και συμπληρώσαμε σύμφωνα με τις απαιτήσεις μας (εικόνα 4.4).



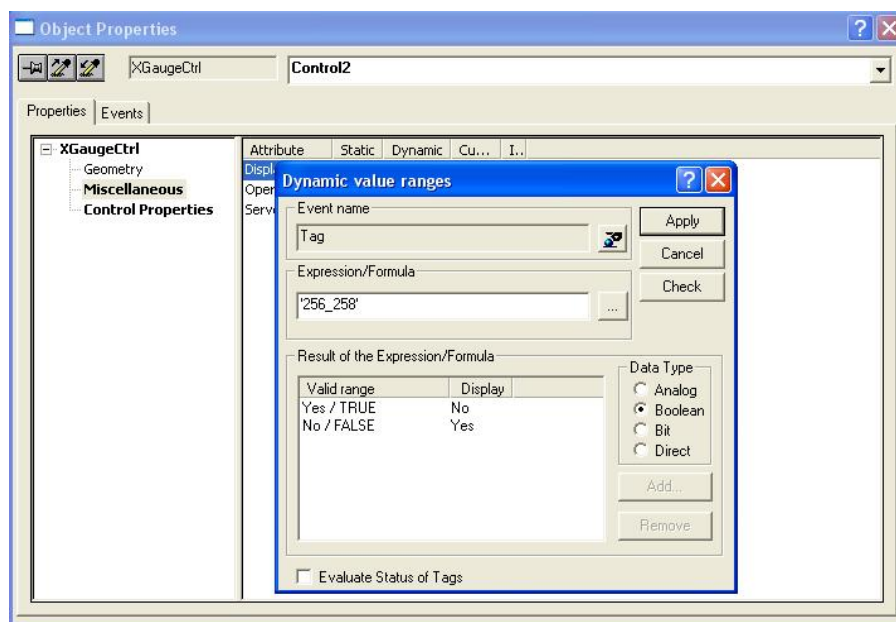
Εικόνα 4.4. Παράμετροι του WinCC Gauge Control.

Για να πετύχουμε το στόχο μας εφαρμόσαμε το εξής τέχνασμα : Τοποθετήσαμε ανά δύο αντικείμενα “Gauge Control” το ένα πάνω στο άλλο, δημιουργώντας έτσι ένα στρώμα το οποίο οπτικά φαίνεται να αποτελείται από τέσσερα τέτοια όμοια αντικείμενα, αλλά στην πραγματικότητα περικλείει και άλλα τέσσερα κρυμμένα από κάτω. Καθορίσαμε ότι το καθένα σετ θα αντιπροσωπεύει και διαφορετικό σετ αναλογικών διευθύνσεων στη σειρά, δηλαδή *PIW256-PIW258*, *PIW260-PIW262* κοκ. Εν συνεχεία, ακολούθησε η δυναμική διασύνδεση των αντικειμένων με τις κατάλληλες εσωτερικές μεταβλητές που έχουν ήδη δημιουργηθεί.

Ειδικότερα, εργαζόμενοι με ένα αντικείμενο από το πρώτο σετ, με τρόπο που υποδείχτηκε στο προηγούμενο κεφάλαιο περί δυναμικής διασύνδεσης αντικειμένων, προκύπτει το παράθυρο της εικόνας 4.5, όπου επεμβαίνουμε στο τμήμα “*Display*” και κάνουμε πάνω του επίσης δεξί κλικ και επιλέγουμε “*Dynamic Dialog*”. Εκεί τώρα, δημιουργούμε διασύνδεση με μία εσωτερική μεταβλητή, για παράδειγμα, όπως φαίνεται στην εικόνα 4.6, την “*256_258*”.



Εικόνα 4.5. Δημιουργώντας δυναμική διασύνδεση στο Gauge Control.



Εικόνα 4.6. Προγραμματίζοντας κατάλληλα το Gauge Control.

Κάνοντας τις κατάλληλες επιλογές προγραμματίζουμε το αντικείμενο ως εξής :

- ❖ Αν η μεταβλητή “256_258” είναι **TRUE** (σε λογικό-“1”) τότε το αντικείμενο να μην εμφανίζεται στην επιφάνεια.
- ❖ Αν η μεταβλητή “256_258” είναι **FALSE** (σε λογικό-“0”) τότε το αντικείμενο να εμφανίζεται στην επιφάνεια.

Στη συνέχεια, με τον ίδιο ακριβώς τρόπο, επιλέγουμε το δεύτερο αντικείμενο του σετ, το διασυνδέουμε με την ίδια μεταβλητή, προγραμματίζοντάς το όμως αντίστροφα. Δηλαδή :

- ❖ Αν η μεταβλητή “256_258” είναι **TRUE** (σε λογικό-“1”) τότε να εμφανίζεται το αντικείμενο στην επιφάνεια.
- ❖ Αν η μεταβλητή “256_258” είναι **FALSE** (σε λογικό-“0”) τότε το αντικείμενο να μην εμφανίζεται στην επιφάνεια.

Με την παραπάνω διαδικασία έχουμε επιτύχει κάτι πολύ σημαντικό. Όταν η μεταβλητή “256_258” είναι **FALSE**, τότε το πρώτο αντικείμενο επικαλύπτει πλήρως το δεύτερο και οπτικά φαίνεται μόνο αυτό. Με κάποιο τρόπο, μεταβάλλοντας την μεταβλητή από **FALSE** σε **TRUE**, τότε αυτομάτως το πρώτο αντικείμενο κρύβεται πίσω από το δεύτερο, το οποίο πλέον έρχεται στο προσκήνιο, χωρίς να το αντιλαμβανόμαστε οπτικά και χωρίς να καταλαμβάνουμε επιπλέον χώρο. Ο τρόπος με τον οποίο μεταβάλλουμε την τιμή της μεταβλητής επιτυγχάνεται με τη χρήση ενός μπουτόν, όπως θα δείξουμε παρακάτω.

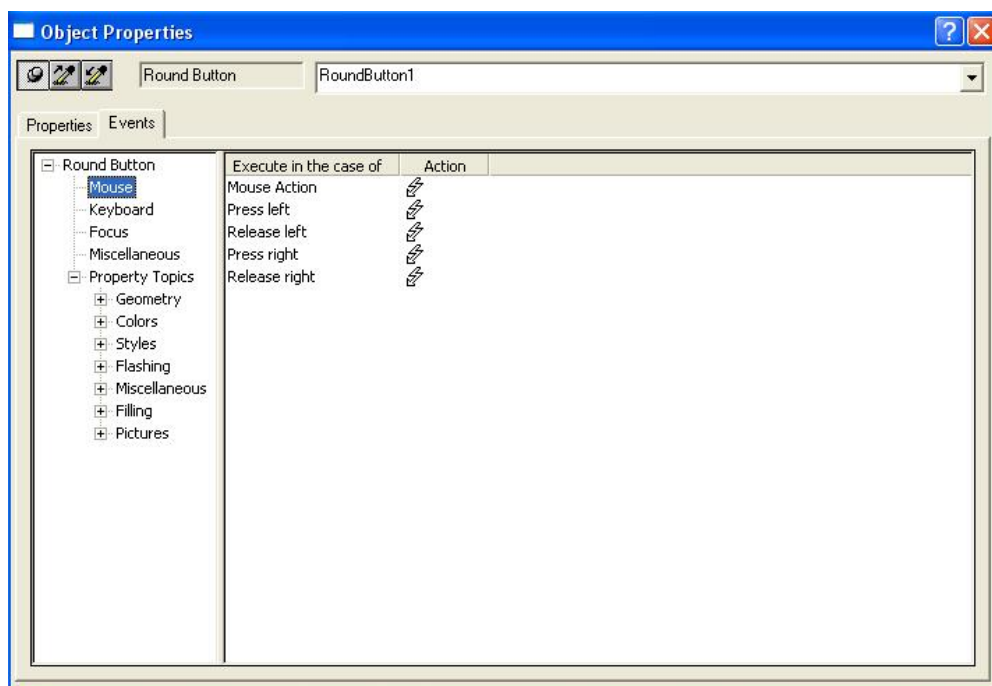
Για να ολοκληρώσουμε πλήρως τον προγραμματισμό των αντικειμένων αυτών, διασυνδέουμε το καθένα, επιπλέον, με κάποια εξωτερική μεταβλητή, η οποία αντιστοιχεί στην εκάστοτε διεύθυνση αναλογικής κάρτας που θέλουμε να παραστήσουμε. Στο παράδειγμά μας, το πρώτο αντικείμενο το διασυνδέουμε με την αναλογική διεύθυνση *PIW256* και το δεύτερο με την *PIW258*. Οι μεταβλητές αυτές δημιουργούνται ξεχωριστά σε κάθε εφαρμογή, γι’ αυτό και προς το παρόν η διασύνδεση αυτή εκκρεμεί. Τέλος, αν και είναι αυτονόητο, οφείλουμε να αναφέρουμε ότι ποτενσιόμετρα δεν ήταν δυνατό να προσθέσουμε στο περιβάλλον που δημιουργήσαμε, αφού τα σήματα προέρχονται από τον πραγματικό προσομοιωτή και εδώ απλά τα παριστάνουμε χωρίς να μπορούμε να μεταβάλλουμε τίποτα.

➤ **Round Button**



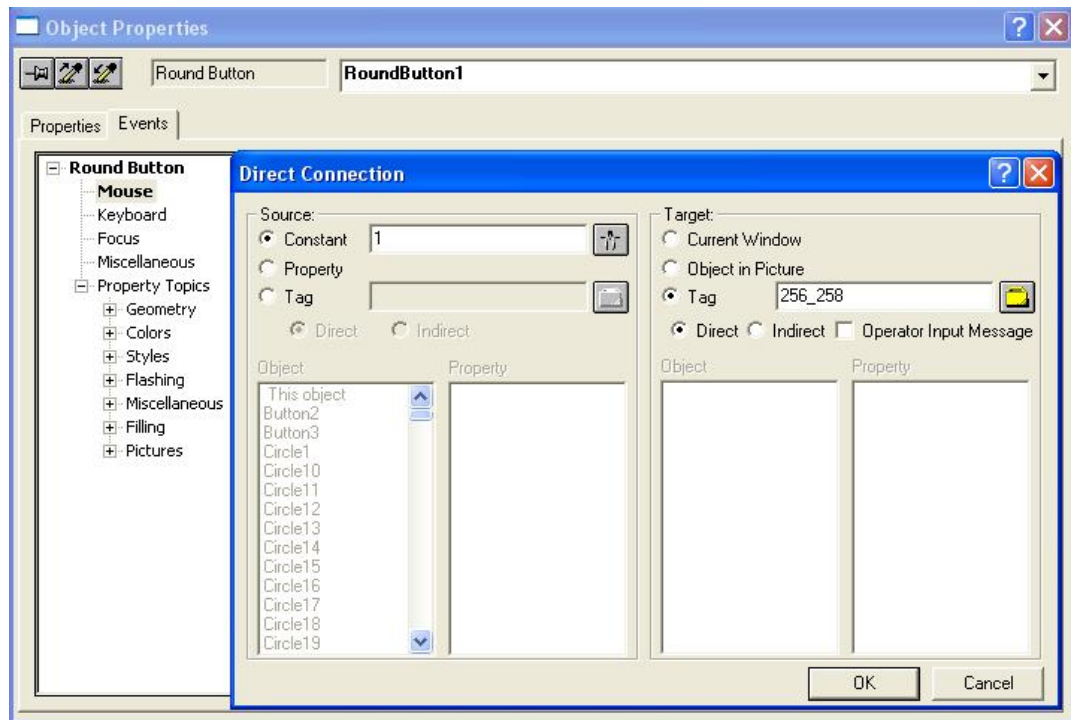
Εικόνα 4.7. WinCC Round Button.

Χρησιμοποιώντας το αντικείμενο αυτό, ολοκληρώσαμε ίσως το πιο δύσκολο κομμάτι της παρούσας εφαρμογής. Διασυνδέσαμε δυναμικά το αντικείμενο με μία εσωτερική μεταβλητή, για παράδειγμα την “256_258” με τρόπο που δείχνει η εικόνα 4.9. Εκεί τώρα το προγραμματίζουμε να κάνει τη μεταβλητή **TRUE**, όταν πατάμε πάνω του αριστερό κλικ, ενώ με δεξί κλικ η τιμή της γίνεται **FALSE**.

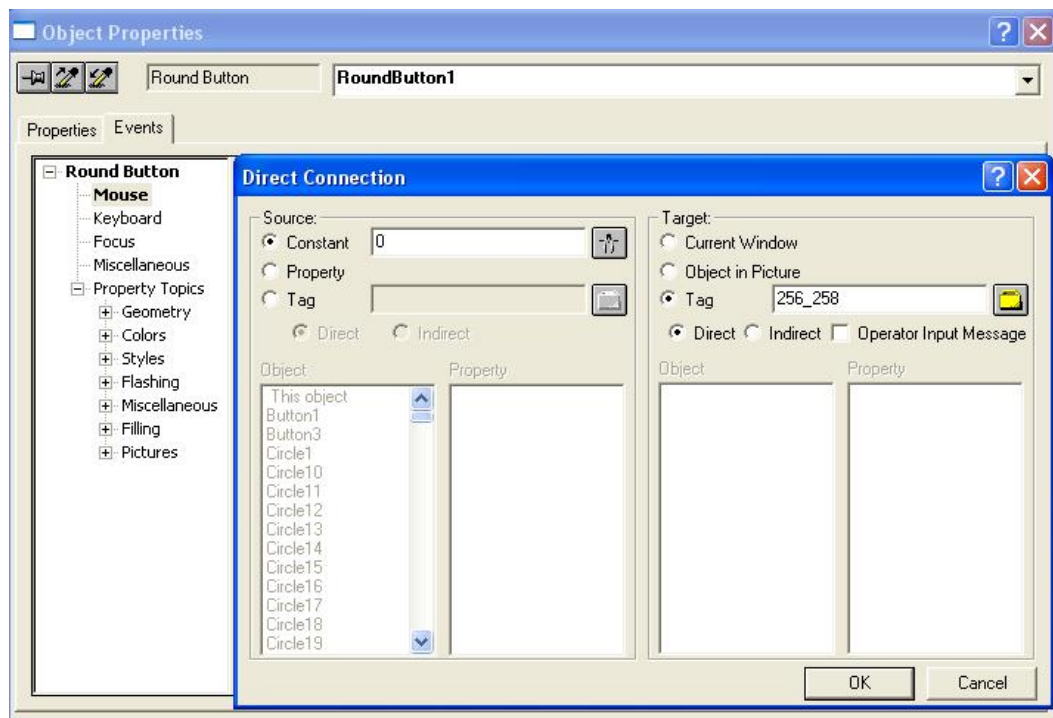


Εικόνα 4.8. Παράθυρο δυναμικής διασύνδεσης του Round Button.

Πιο συγκεκριμένα, για να πετύχουμε την παραπάνω λειτουργία, στο μενού της εικόνας 4.8 κάνουμε δεξί κλικ στο τμήμα “Press left”, όπου τώρα όμως προκύπτει ένα διαφορετικό πλαίσιο από αυτό που είχαμε συνηθίσει και επιλέγουμε “Direct connection”. Το αναδυόμενο παράθυρο που προκύπτει, μας δίνει τη δυνατότητα να ορίσουμε ότι με αριστερό κλικ του ποντικιού θα δίνουμε στην μεταβλητή “256_258” λογικό-“1” (εικόνα 4.9). Το ίδιο επαναλαμβάνουμε στο τμήμα “Press right”, όπου τώρα με δεξί κλικ η μεταβλητή θα παίρνει λογικό-“0” (εικόνα 4.10). Έτσι, έχουμε καταφέρει να εμφανίζεται το σήμα μίας αναλογικής διεύθυνσης στον εικονικό προσομοιωτή μας και με αριστερό κλικ του “Round Button” που βρίσκεται από κάτω του να εναλλάσσουμε την αναλογική αυτή διεύθυνση με την αμέσως επόμενη, ενώ με δεξί κλικ να εμφανίζεται πάλι η πρώτη. Η διαδικασία αυτή εφαρμόζεται για την αναπαράσταση όλων των αναλογικών διευθύνσεων απαιτώντας τελικά τέσσερα “Round Button”.



Εικόνα 4.9. Δίνοντας τιμή λογικό-“1” σε μεταβλητή με το Round Button.

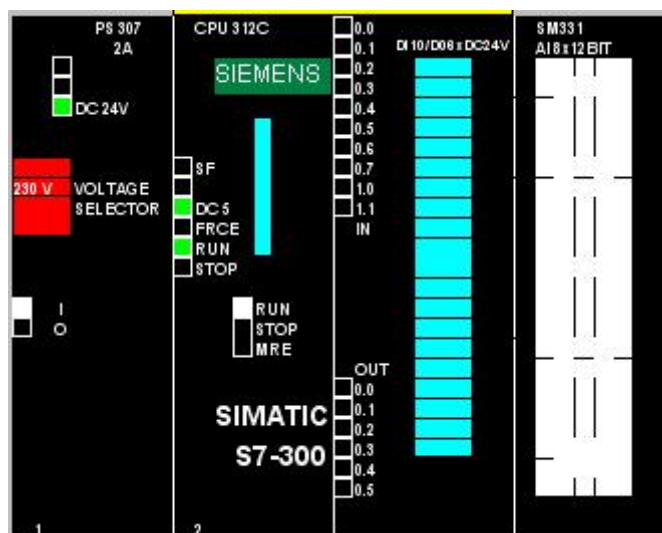


Εικόνα 4.10. Δίνοντας τιμή λογικό-“0” σε μεταβλητή με το Round Button.

➤ Βασικά γεωμετρικά σχήματα

Τις υπόλοιπες λειτουργίες του προσομοιωτή τις αναπαραστήσαμε με κατάλληλα βασικά γεωμετρικά σχήματα. Χρησιμοποιώντας κύκλους, αναπαραστήσαμε τα LEDs εξόδου, διασυνδέοντας απλά τα αντικείμενα αυτά με τις αρμόδιες εξωτερικές μεταβλητές, οι οποίες δημιουργούνται σε κάθε ζητούμενη εφαρμογή. Ο τρόπος διασύνδεσης είναι ακριβώς ο ίδιος με το παράδειγμα του κεφαλαίου 3, αλλάζοντας μόνο τα χρώματα και τις διαστάσεις.

Συνδυάζοντας τα βασικά σχήματα και χρώματα, καθώς και τις ιδιότητές τους, φτιάξαμε ένα εικονικό PLC (εικόνα 4.11), το οποίο οπτικά κρίνεται ιδιαίτερα επιτυχημένο και πολύ κοντά στο πραγματικό. Το γεγονός αυτό αποδεικνύει την μεγάλη επάρκεια του WinCC σε σχήματα, βιβλιοθήκες και γενικά σε δυνατότητα επέμβασης σχεδόν σε όλες τις ιδιότητες των αντικειμένων που μας ενδιαφέρουν. Αξίζει να σημειωθεί ότι καταφέραμε να προσομοιώσουμε ακόμη και τα LEDs εισόδου/εξόδου του ίδιου του PLC, χρησιμοποιώντας τετράγωνα στα οποία διασυνδέουμε τις εξωτερικές μεταβλητές εισόδου/εξόδου, με τα ίδια χρώματα.



Εικόνα 4.11. Το εικονικό μας PLC.

Κρίνεται επιτακτική η ανάγκη να αναφερθεί ο τρόπος με τον οποίο παραστήσαμε τους απλούς διακόπτες ON-OFF, με απλό συνδυασμό πάλι βασικών σχημάτων. Όπως δείχνει η εικόνα 4.12, έχουμε χρησιμοποιήσει τρεις απλές γραμμές, οι οποίες διαφέρουν σε πάχος.

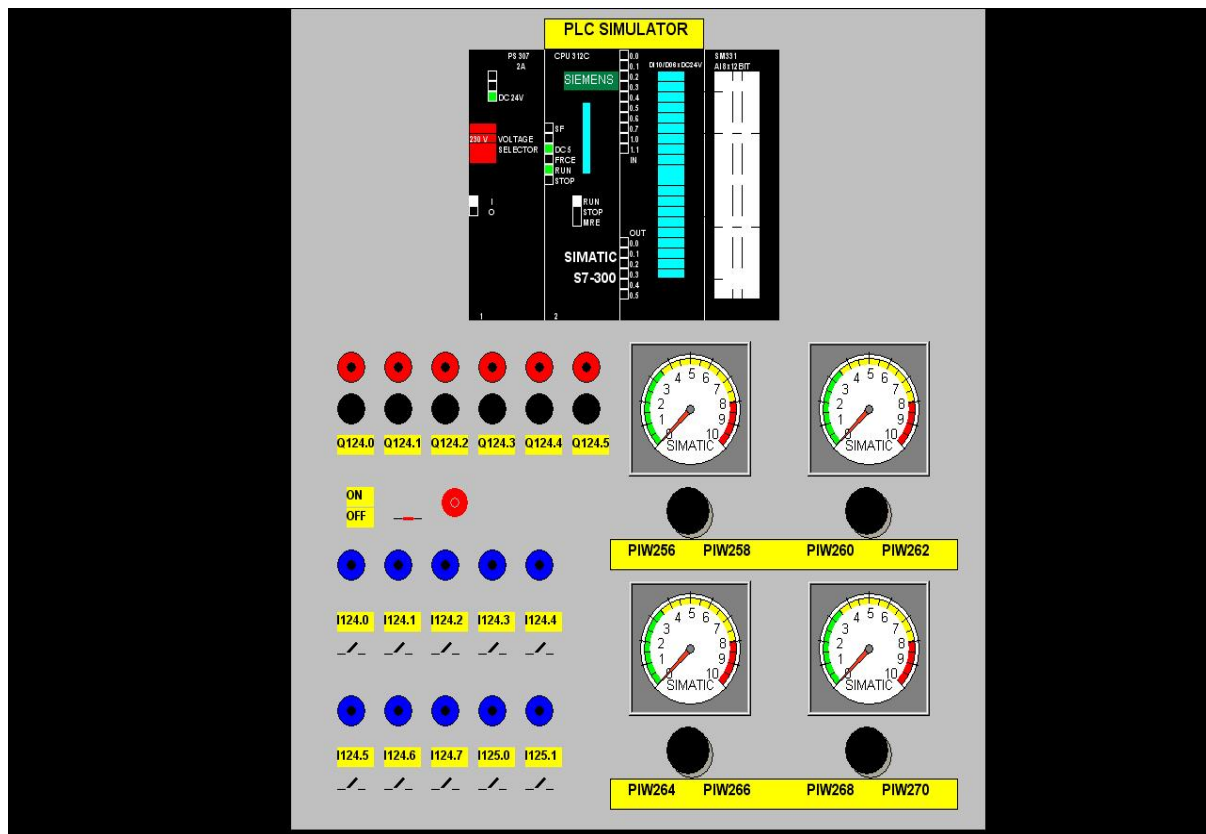


Εικόνα 4.12. Ο εικονικός μας διακόπτης.

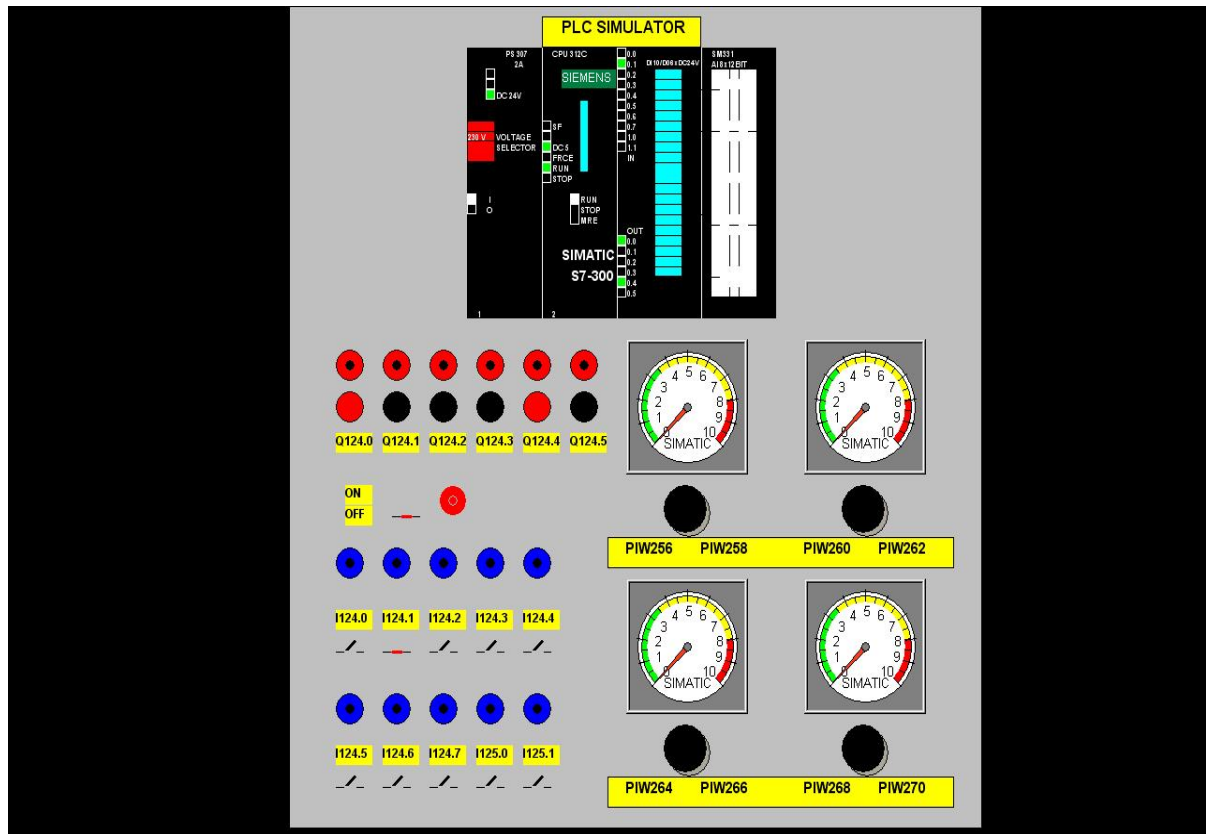
Όταν ο αντίστοιχος διακόπτης, άρα και η αντίστοιχη ψηφιακή είσοδος είναι **TRUE**, τότε η οριζόντια παχιά γραμμή γίνεται κόκκινη ενώ η μαύρη κεκλιμένη γραμμή παίρνει το χρώμα της επιφάνειας που βρίσκεται, δηλαδή οπτικά γίνεται αόρατη (εικόνα 4.13). Με αυτή την απλή λογική, καταφέραμε να παραστήσουμε τους διακόπτες με τρόπο απλό μα και εντυπωσιακό και με ιδιαίτερα καλή απόκριση από πλευράς δικτύου και συστήματος.



Εικόνα 4.13. Ο διακόπτης είναι OFF και ON.



Εικόνα 4.15. Ο εικονικός προσομοιωτής στην τελική του μορφή.



Εικόνα 4.16. Ο εικονικός προσομοιωτής σε λειτουργία.

Τελικά ο προσομοιωτής πήρε τη μορφή που δείχνουν οι προηγούμενες εικόνες, προσθέτοντας και τα απαραίτητα κείμενα και ρυθμίζοντας και τις τελευταίες λεπτομέρειες. Τώρα πια σώζουμε το αρχείο ως “simulator.pdl” και πλέον μπορούμε να το αντιγράψουμε από το περιβάλλον του Editor σε όποια εφαρμογή θέλουμε, προσέχοντας δύο βασικές λεπτομέρειες : να αντιγράψουμε τις εσωτερικές μεταβλητές και να διασυνδέσουμε τα αντικείμενα με τις κατάλληλες εξωτερικές μεταβλητές.

ΚΕΦΑΛΑΙΟ 5

Σενάριο 1: Έλεγχος Μ/Σ Ισχύος

5.1 Αντικείμενο ζητούμενης εφαρμογής.

Από το παρόν κεφάλαιο ξεκινάμε πλέον την περιγραφή των ολοκληρωμένων εφαρμογών οι οποίες ήταν και τα ζητούμενα σημεία στην παρούσα διπλωματική εργασία. Σύμφωνα με το σενάριο 1, καλούμαστε να δημιουργήσουμε ένα σύστημα SCADA για τον έλεγχο ενός Μετασχηματιστή Ισχύος (εδώ Μέσης Τάσης), ο οποίος υποτίθεται ότι βρίσκεται εγκατεστημένος σε ένα απομακρυσμένο μέρος, τροφοδοτώντας μία βιοτεχνική μονάδα. Ο εποπτικός έλεγχος και το γενικό σενάριο που καλούμαστε να δημιουργήσουμε έχει τρία σκέλη :

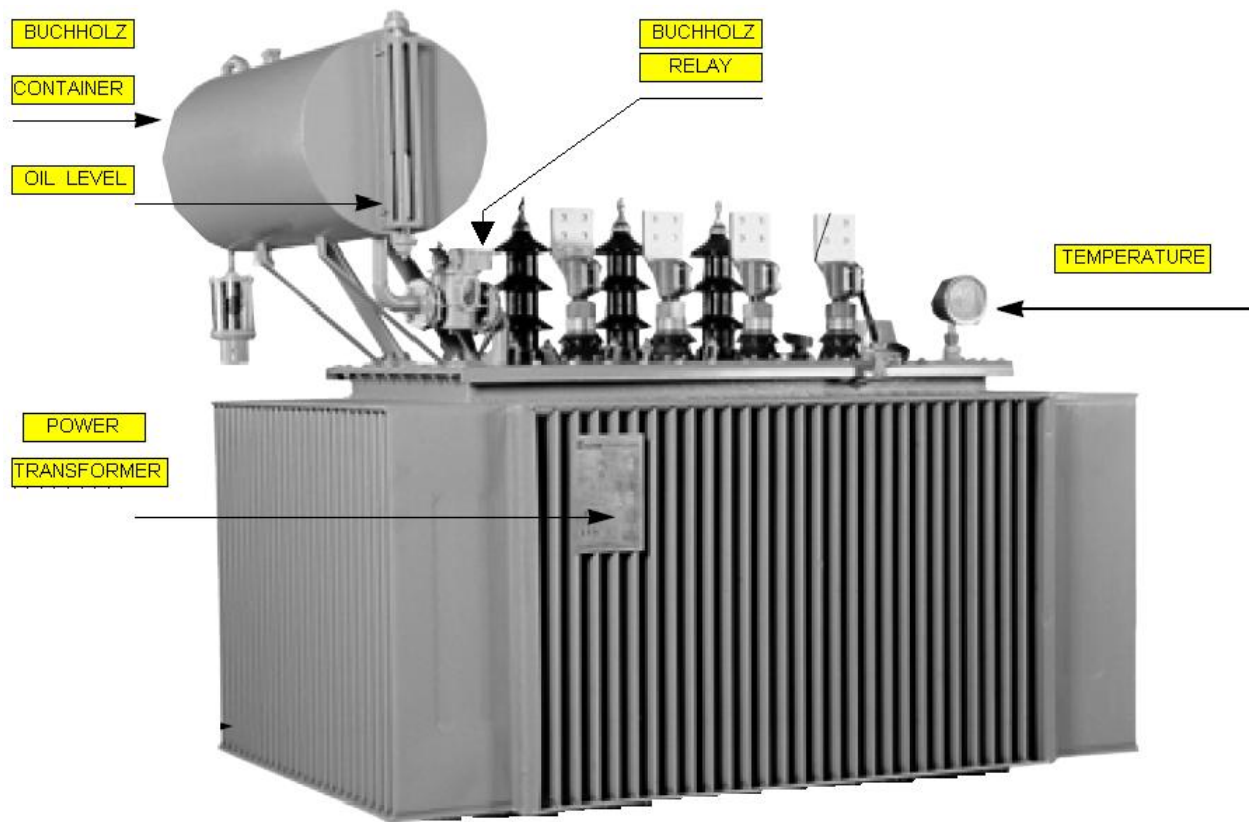
- Ανίχνευση και εποπτεία όλων των σφαλμάτων που δύναται να παρουσιάσει ο Μ/Σ.
- Δυνατότητα απομακρυσμένου ελέγχου του Μ/Σ αλλά και του προσομοιωτή επιδεικνύοντας τις δυνατότητες ενός SCADA.
- Δημιουργία εικονικού προσομοιωτή ως ξεχωριστής εικόνα εργασίας.

5.2 Μετασχηματιστής και σήματα εντοπισμού σφαλμάτων.

Αναφερόμενοι στο παρόν τύπο Μ/Σ Ισχύος αναφέρουμε ότι είναι ο κύριος τύπος που χρησιμοποιεί η ΔΕΗ στις γραμμές Μέσης Τάσης για μετατροπή σε Χαμηλή Τάση. Όπως δείχνει και το σχήμα 5.1, τα βασικά σημεία στα οποία μπορεί να παρουσιάσει σφάλμα ο Μ/Σ είναι η θερμοκρασία και η ποιότητα/στάθμη λαδιού. Το λάδι το οποίο βρίσκεται στο εσωτερικό ενός τέτοιου Μ/Σ καταλήγει σε ένα δοχείο διαστολής, το δοχείο Buchholz. Εκεί μόλις πέσει η στάθμη του κάτω από ένα όριο υπάρχει αισθητήρας, ο οποίος κλείνει την κατάλληλη επαφή ενός ηλεκτρονόμου (Buchholz relay) και ο οποίος δίνει σήμα, με τη σειρά του στην εγκατεστημένη μονάδα RTU. Με τον ίδιο τρόπο γίνεται ανίχνευση σήματος υπερβολικής θερμοκρασίας. Η μονάδα RTU ενημερώνει το PLC και γενικά το σύστημα SCADA μέσω της τηλεπικοινωνιακής ζεύξης, η οποία υφίσταται ανάμεσά τους.

Αναλόγως του σφάλματος ή των σφαλμάτων που δημιουργούνται και τη χρονική διάρκεια αυτών, ο Μ/Σ δύναται να αποκοπεί από το δίκτυο, ώστε να αποφευχθούν χειρότερες καταστάσεις και ακόμη να ελεγχθεί αν όντως αυτή η αποκοπή συνέβησε, μιας που υπάρχει η πιθανότητα να μην λειτουργήσει ο ηλεκτρονόμος που βγάζει τον Μ/Σ από το δίκτυο και να συνεχίσει η λειτουργία του αδιαλείπτως, προξενώντας μεγαλύτερα προβλήματα. Η πολύ

σημαντική αυτή λειτουργία επιτυγχάνεται με χρήση ηλεκτρονόμου ανάδρασης, τις επαφές του οποίου προσομοιώνουμε είτε τοπικά είτε απομακρυσμένα.



Σχήμα 5.1. Μετασχηματιστής Ισχύος Μέσης Τάσης.

Οι παρακάτω πίνακες παρουσιάζουν αναλυτικά τα σήματα λάθους που μπορεί να προκύψουν, τις χρονικές διάρκειές τους καθώς και την προγραμματισμένη αντίδραση των LEDs εξόδου, ώστε να ερμηνευτούν κατάλληλα από τον χρήστη του περιβάλλοντος εργασίας. Μάλιστα, σε περίπτωση που κάποια σήματα συμπέσουν προβλέπεται διαφορετική αντίδραση, αφού μεγαλώνει έτσι ο κίνδυνος δυσλειτουργίας του Μ/Σ. Αν πάλι τα σήματα σταματήσουν να υφίσταται μέσα στα προκαθορισμένα χρονικά διαστήματα, τότε ο Μ/Σ αφήνεται να λειτουργήσει κανονικά, έχοντας όμως παρατηρήσει και αποθηκεύσει αυτά με τη βοήθεια του WinCC, είμαστε σε θέση να προβούμε σε προληπτικές ενέργειες (αποκατάσταση λαδιού, συντήρηση Μ/Σ) ώστε να προλάβουμε μία κατάσταση προτού γίνει μη αναστρέψιμη και καταστρεπτική τόσο λειτουργικά όσο και οικονομικά.

ΣΗΜΑ	ΧΡΟΝΙΚΟ ΔΙΑΣΤΗΜΑ (sec)	ΑΠΟΤΕΛΕΣΜΑ
ALARM TH	0-60	ALARM TEMPERATURE LED ON
	60-119	ALARM TEMPERATURE LED FLASH
Π	120	TRANSFORMER OFF
TRIP TH	0-24	TRIP TEMPERATURE LED ON
	24-30	TRIP TEMPERATURE LED FLASH
	36	TRANSFORMER OFF
ALARM BUCH	0-60	ALARM BUCCHOLZ LED ON
	60-83	ALARM BUCCHOLZ LED FLASH
	84	TRANSFORMER OFF
TRIP BUCH	0-24	TRIP BUCCHOLZ LED ON
	24-30	TRIP BUCHHOLZ LED FLASH
	30	TRANSFORMER OFF

Πίνακας 5.1. Σήματα που καταφθάνουν χωρίς παρεμβολές από άλλα.

ΣΗΜΑ	ΧΡΟΝΙΚΟ ΔΙΑΣΤΗΜΑ (sec)	ΑΠΟΤΕΛΕΣΜΑ
ALARM TH		
+TRIP TH	<36	TRANSFORMER OFF
ALARM BUCH		
+TRIP BUCH	<36	TRANSFORMER OFF

Πίνακας 5.2. Σήματα που καταφθάνουν και ακολουθούν παρεμβολές από άλλα.

Στους παραπάνω πίνακες εμφανίζονται τα σήματα για τα οποία έγινε λόγος πριν. Συγκεκριμένα, ο ηλεκτρονόμος Buchholz φέρει δύο ανοικτές επαφές, την «**ALARM BUCH**» και την «**TRIP BUCH**», που ελέγχουν ποιότητα και στάθμη λαδιού. Επιπροσθέτως, για τον ηλεκτρονόμο που ελέγχει τη θερμοκρασία υπάρχουν επίσης δύο τέτοιες ανοικτές επαφές, «**ALARM TH**» και «**TRIP TH**». Όταν αναφέρουμε ότι κάποιο LED κάνει «**FLASH**», εννοούμε ότι αναμοσβήνει ενώ το σύμβολο “+” υπονοεί ότι σε ένα σήμα λάθους που έχει ήδη καταφθάσει στο σύστημά μας παρεμβάλλεται ακόμη άλλο ένα, σε διάστημα που καθορίζεται στη μεσαία στήλη και συνιστά αυξημένο κίνδυνο για τον Μ/Σ .

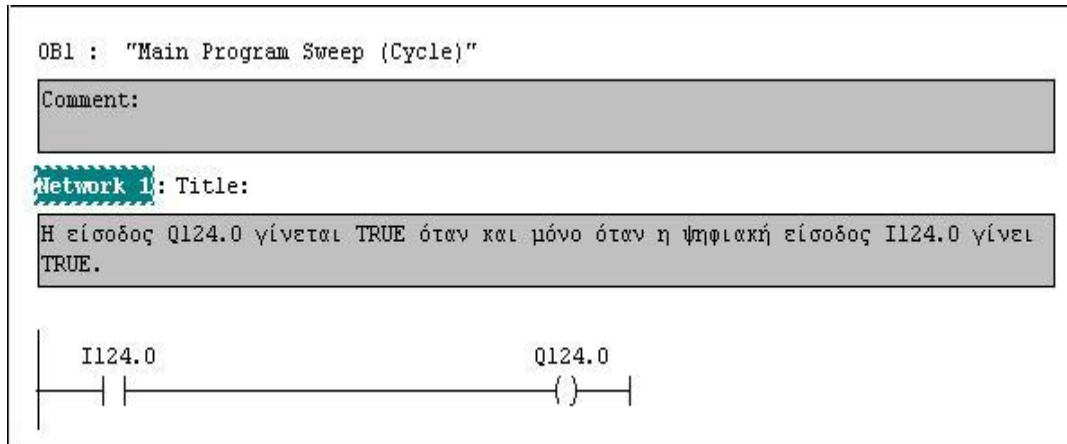
5.3 Δημιουργία εφαρμογής.

5.3.1 Πρόγραμμα αυτοματισμού.

Προκειμένου να δημιουργηθεί η παρούσα εφαρμογή, επιβάλλεται πρώτα απ'όλα ο προγραμματισμός του PLC. Το πρόγραμμα που χρησιμοποιήθηκε δημιουργήθηκε με το πακέτο «**SIMATIC STEP7 V5.2 + SP1**» της **SIEMENS** σε γλώσσα προγραμματισμού **LADDER**. Εκτός του κλασικού τρόπου προγραμματισμού, ο οποίος απαιτείται για μία τέτοια εφαρμογή, χρειάστηκε να χρησιμοποιήσουμε μία νέα λογική ώστε να πετύχουμε τον απομακρυσμένο έλεγχο του Μ/Σ από το περιβάλλον εργασίας και όχι απλά την εποπτεία του.

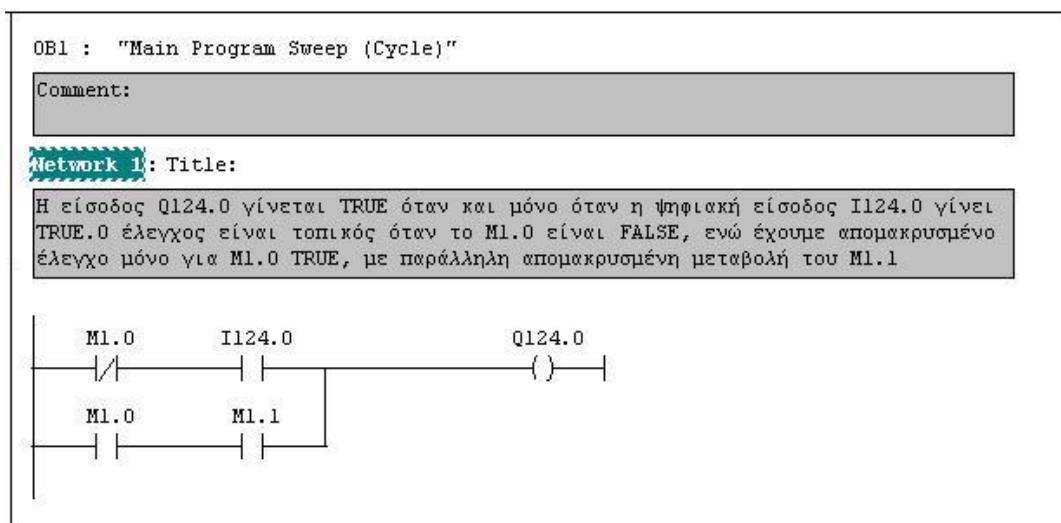
Συγκεκριμένα, θέλουμε με επιλογή ενός μπουτόν από τον υπολογιστή μας να αποκλείσουμε το χειρισμό από τους διακόπτες του προσομοιωτή, δηλαδή να επιβάλλουμε στο PLC να αγνοεί τυχόν τοπικές μεταβολές από πραγματικούς διακόπτες στις εισόδους του και να δέχεται μόνο ο,τι του υποδεικνύει το λογισμικό και το αντίστροφο. Εδώ είχαμε ένα σημαντικό πρόβλημα να αντιμετωπίσουμε διότι η CPU του PLC είναι προγραμματισμένη να δέχεται με αυστηρή προτεραιότητα σήματα του υλικού της εγκατάστασης και να αγνοεί τυχόν εντολές του απομακρυσμένου υπολογιστή που έρχονται μετά ή και το αντίστροφο, δηλαδή να αναιρεί εντολές που έχει ήδη δεχτεί από το WinCC για να εξυπηρετήσει μετέπειτα εντολές που προέρχονται από κάποιον διακόπτη, άμεσα συνδεδεμένο ηλεκτρολογικά με το PLC.

Σαν μία πρώτη σκέψη μία λύση θα ήταν ο επαναπρογραμματισμός της CPU, αλλά κάτι τέτοιο απορρίπτεται μονομιάς λόγω της πολυπλοκότητας ενός τέτοιου εγχειρήματος το οποίο πιθανότατα δε θα είχε κανένα αποτέλεσμα. Οπότε η άλλη λύση που απέμενε ήταν να παρακάμψουμε κατά κάποιο τρόπο την διαδικασία αυτή της CPU, με κατάλληλη προγραμματιστική τεχνική. Στο σχήμα 5.2 έχουμε δημιουργήσει ένα παράδειγμα που αντικατοπτρίζει την τεχνική αυτή. Έστω ότι με ένα απλό πρόγραμμα θέλουμε να δίνουμε την τιμή **TRUE** στην ψηφιακή έξοδο **Q124.0**, όταν η ψηφιακή είσοδος **I124.0** γίνει **TRUE**, δηλαδή εκμεταλλευόμενοι τον προσομοιωτή, όταν ο διακόπτης **I124.0** είναι **ON**. Το σχήμα αυτό το απλούστατο πρόγραμμα σε **LADDER**.



Σχήμα 5.2. Απλό πρόγραμμα-παράδειγμα.

Χρειαζόμαστε τώρα να έχουμε τη δυνατότητα να επιλέγουμε από το περιβάλλον εργασίας του WinCC αν η μεταβολή της **Q124.0** θα γίνεται τοπικά από τον διακόπτη **I124.0** ή απομακρυσμένα από το WinCC και έπειτα να την μεταβάλλουμε από εκεί. Αυτή η τεχνική επιτυγχάνεται με χρήση δύο bits μνήμης τα M1.0 και M1.1. Το M1.0 καθορίζει αν ο έλεγχος θα είναι τοπικός ή απομακρυσμένος και τοποθετείται όπως δείχνει το σχήμα 5.3, ενώ το M1.1 μεταβάλλει την έξοδό μας, όταν το επιτρέψει η κατάσταση του M1.0. Αξίζει να παρατηρήσει κανείς ότι όταν το M1.0 είναι **TRUE**, τότε ο έλεγχος θα είναι μόνο απομακρυσμένος αγνοώντας τον διακόπτη, ενώ όταν είναι **FALSE** τότε μόνο ο διακόπτης θα αλλάζει την έξοδο με μεταβολή της κατάστασης του M1.1. Ο τρόπος με τον οποίο έγινε η δυναμική διασύνδεση των bits αυτών έχει ήδη περιγραφεί και έτσι, με τα κατάλληλα αντικείμενα του WinCC, εφαρμόζουμε αυτήν την πραγματικά απλή τεχνική, της οποίας όμως η αξία αποδείχτηκε μεγάλη για την υλοποίηση της εργασίας μας.



Σχήμα 5.3. Το πρόγραμμα-παράδειμά μας ολοκληρωμένο.

Η παραπάνω τεχνική ακολουθήθηκε σε όλα τα σενάρια που υλοποιήσαμε, ενώ το πρόγραμμα αυτοματισμού της παρούσας εφαρμογής παρατίθεται στο παράρτημα στο τέλος της εργασίας.

5.3.2 Δημιουργία περιβάλλοντος εργασίας.

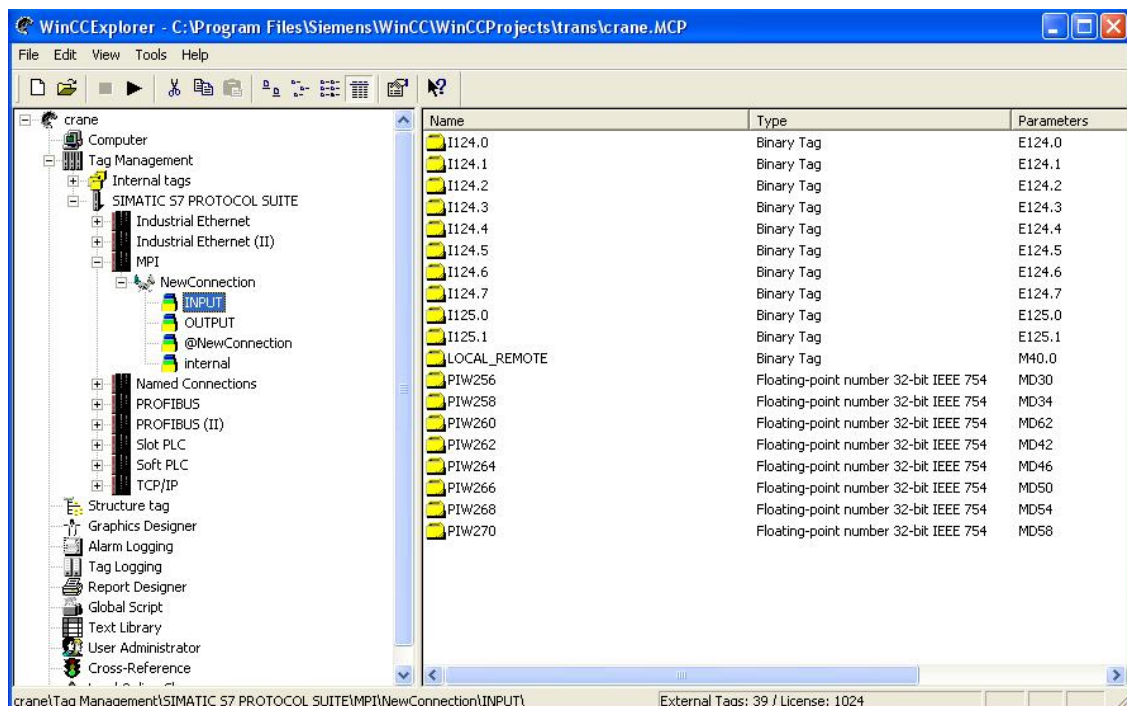
Έχοντας δημιουργήσει και φορτώσει το πρόγραμμά μας στο PLC, περάσαμε στην δημιουργία του περιβάλλοντος εργασίας στο WinCC. Καταρχάς δημιουργήσαμε μία νέα εφαρμογή με το όνομα “crane” και εφαρμόσαμε τα παρακάτω :

Editor

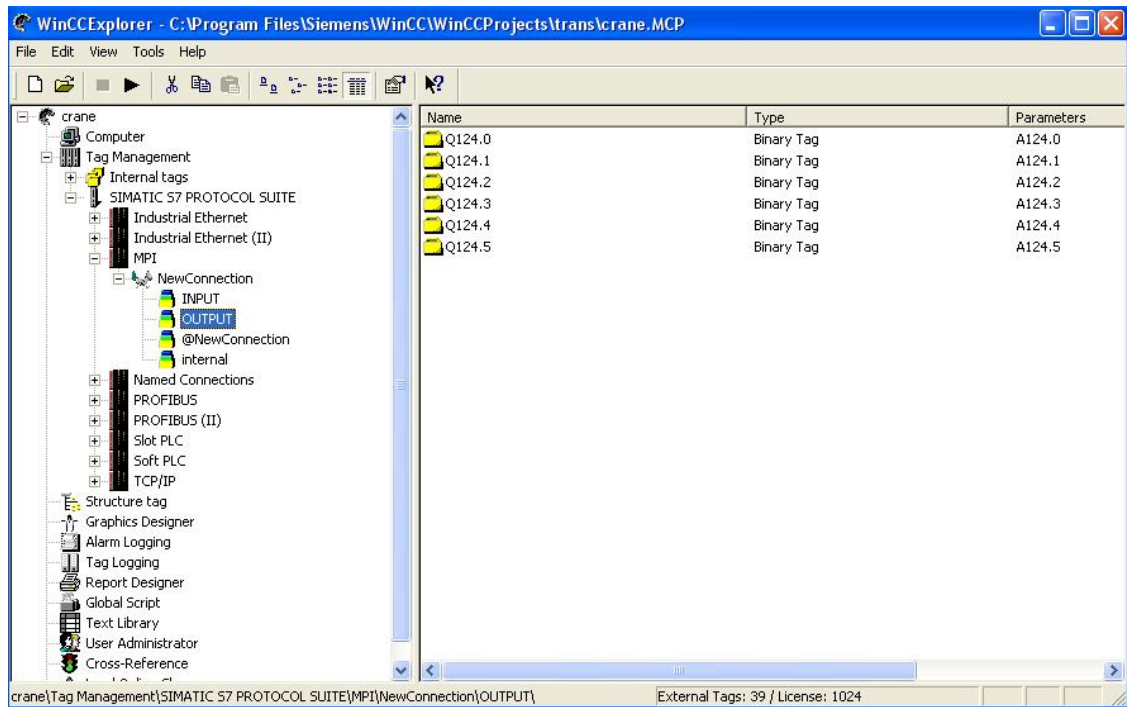
Στο υπομήμα “Computer” εφαρμόσαμε τις ρυθμίσεις που αναφέραμε στο κεφάλαιο 2, σχετικά με την ευπαρουσίαστη εικόνα της εφαρμογής μας.

Tag Management

Οι εξωτερικές μεταβλητές που δημιουργήσαμε χωρίστηκαν σε δύο ομάδες (Groups) οι οποίες ονομάστηκαν “INPUT” και “OUTPUT” (εικόνες 5.1 & 5.2).



Εικόνα 5.1. Εξωτερικές μεταβλητές του Group “INPUT”.

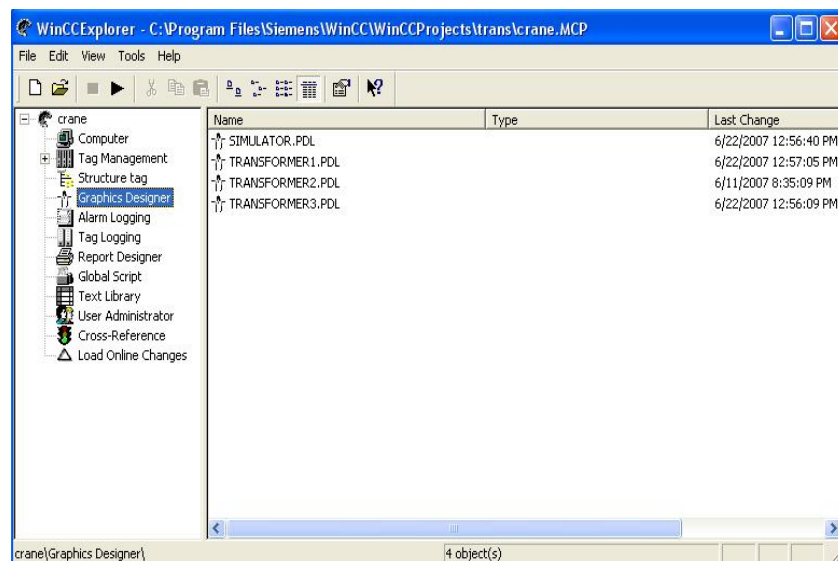


Εικόνα 5.2. Εξωτερικές μεταβλητές του Group “OUTPUT”.

Οι εσωτερικές μεταβλητές, όπως αναφέρθηκε στο κεφάλαιο 4, αντιγράφηκαν μαζί με το αρχείο του προσομοιωτή ώστε να τον χρησιμοποιήσουμε σαν εικόνα εργασίας.

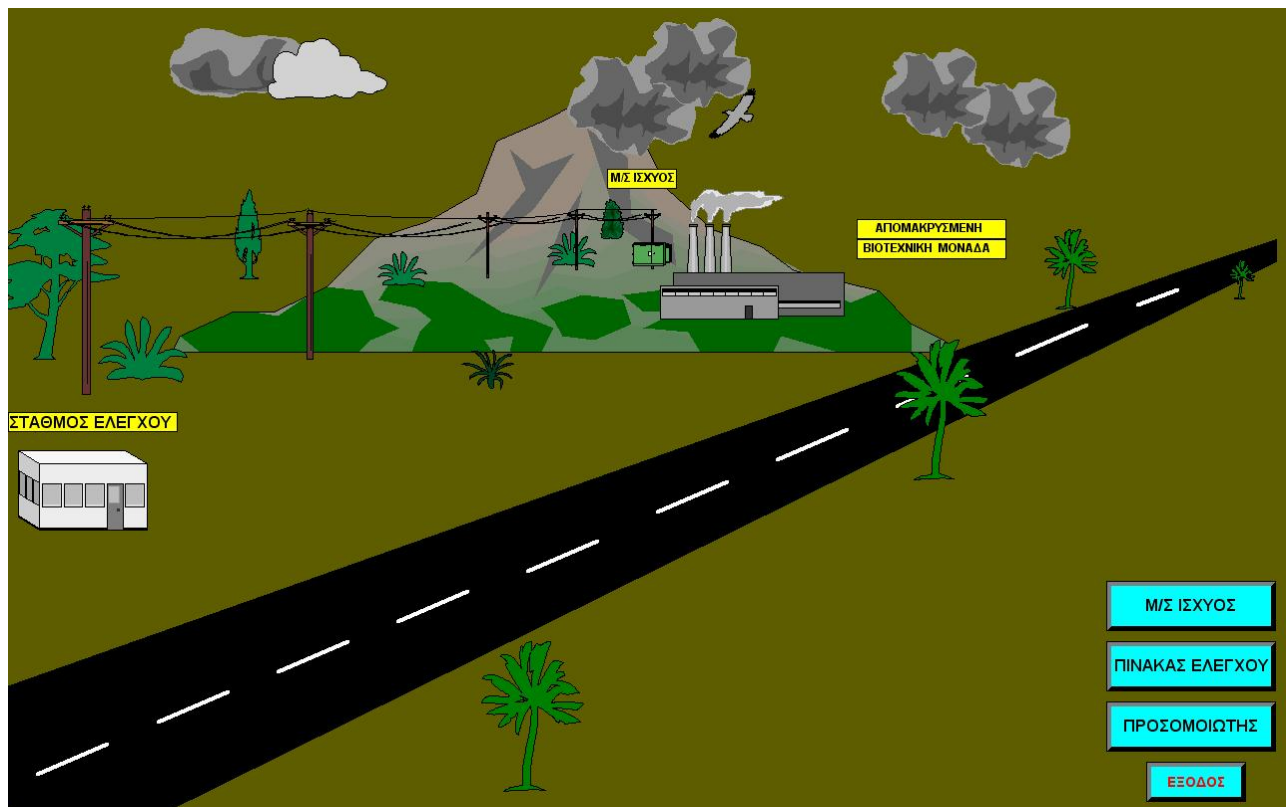
Graphics Designer

Με τη βοήθεια του σχεδιαστή γραφικών δημιουργήσαμε τρία αρχεία και αντιγράψαμε εδώ το ήδη έτοιμο αρχείο του προσομοιωτή (εικόνα 5.3). Στη συνέχεια παρουσιάζουμε την προεπισκόπηση της κάθε εικόνας εργασίας και περιγράφουμε τη δημιουργία της, όπου κρίνεται αναγκαίο.



Εικόνα 5.3. Τα αρχεία του Graphics Designer.

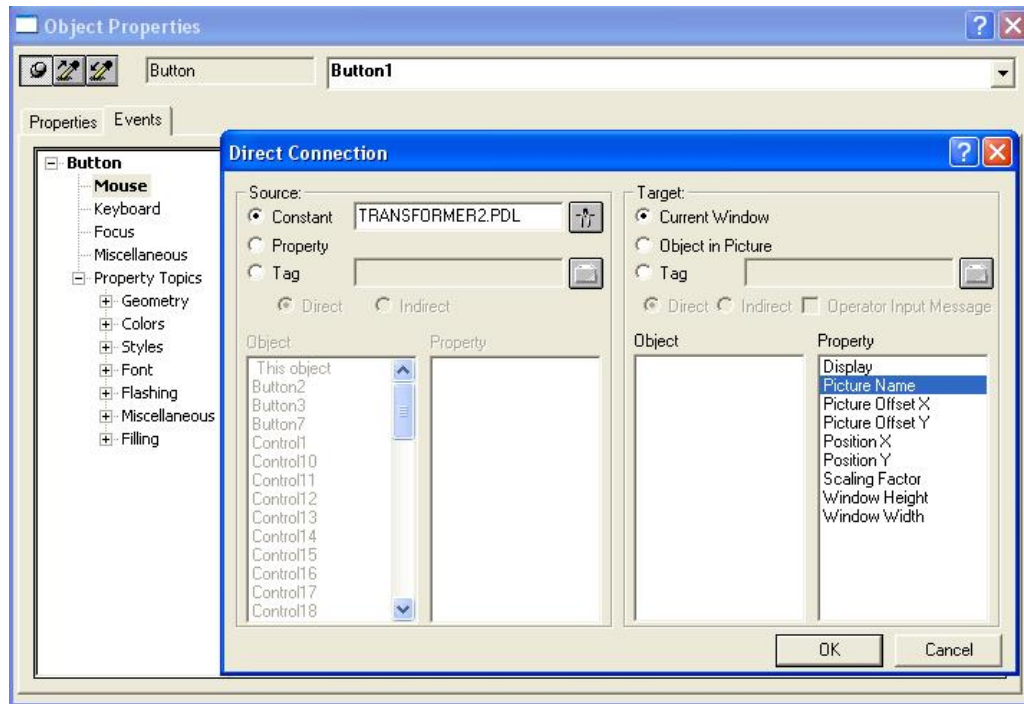
➤ Transformer1.pdl



Εικόνα 5.4. Η πρώτη εικόνα της εφαρμογής.

Η εικόνα αυτή, η οποία ορίστηκε και σαν αρχική όταν ξεκινάει η εφαρμογή να “τρέχει”, δημιουργήθηκε με τη λογική εισαγωγής στο περιβάλλον του παρόντος συστήματος SCADA. Αναπαραστήσαμε χωροταξικά ένα υποθετικό σύστημα, όπου φαίνεται καθαρά η απόσταση που δύναται να υπάρχει μεταξύ του χρήστη που βρίσκεται στον σταθμό ελέγχου και του αντικείμενου που εποπτεύουμε και εδώ είναι ο Μ/Σ. Πράγματι, με τη βοήθεια της πληρέστατης βιβλιοθήκης του Graphics Designer δημιουργήσαμε ένα εντυπωσιακό αποτέλεσμα, το οποίο, επίσης, μας δείχνει γιατί πολλές φορές είναι αναγκαίος ο έλεγχος και η μεταβολή από απόσταση.

Όσον αφορά τα αντικείμενα που εισάγαμε, τα μπουτόν που βρίσκονται δεξιά κάτω στην εικόνα παρουσιάζουν ιδιαίτερο ενδιαφέρον. Τα τρία πρώτα χρησιμεύουν στην αλλαγή από την μία εικόνα στην άλλη, ενώ είμαστε σε λειτουργία. Για παράδειγμα, κάνοντας δεξιά κλικ στο πρώτο μπουτόν και πηγαίνοντας στην επιλογή “*Press left*”, όπως κάναμε και στα “*Round button*” στο κεφάλαιο 4, καταλήγουμε στο γνώριμο αναδυόμενο παράθυρο της εικόνας 5.5, όπου κάνουμε τις επιλογές που φαίνονται. Πιο συγκεκριμένα, προγραμματίζουμε το μπουτόν, όταν κάνουμε αριστερό κλικ πάνω του να μεταφερόμαστε στην εικόνα που θέλουμε, εδώ στην εικόνα του αρχείου «*transformer2.pdl*». Με τον ίδιο τρόπο προγραμματίστηκαν και τα υπόλοιπα δύο μπουτόν, τα οποία παραπέμπουν στην εκάστοτε εικόνα που αναγράφουν.



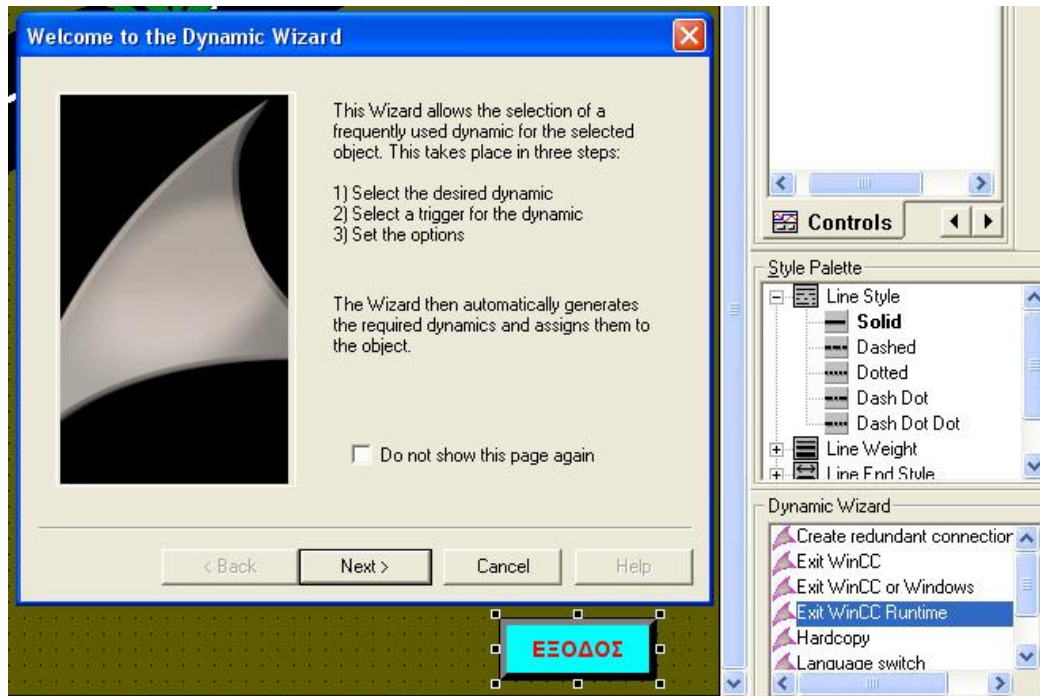
Εικόνα 5.5. Προγραμματισμός των μπουτόν αλλαγής περιβάλλοντος.

Μείζονος σημασίας μπουτόν είναι και το τελευταίο το οποίο αναγράφει «**ΕΞΟΔΟΣ**». Έχοντας ρυθμίσει το “Computer” όπως αναφέραμε παραπάνω, εκμεταλλευόμαστε ολόκληρη την οθόνη του υπολογιστή, χωρίς όμως να έχουμε τη δυνατότητα να κλείσουμε την εικόνα που βλέπουμε εκείνη τη στιγμή, με τον κλασικό τρόπο που κλείνουν τα παράθυρα στο λειτουργικό σύστημα των *WindowsXP*. Με το μπουτόν όμως αυτό, πατώντας με αριστερό κλικ πάνω του, φεύγουμε από το *Runtime* και επανερχόμαστε στον Editor ή όπου αλλού βρισκόμαστε στο WinCC.

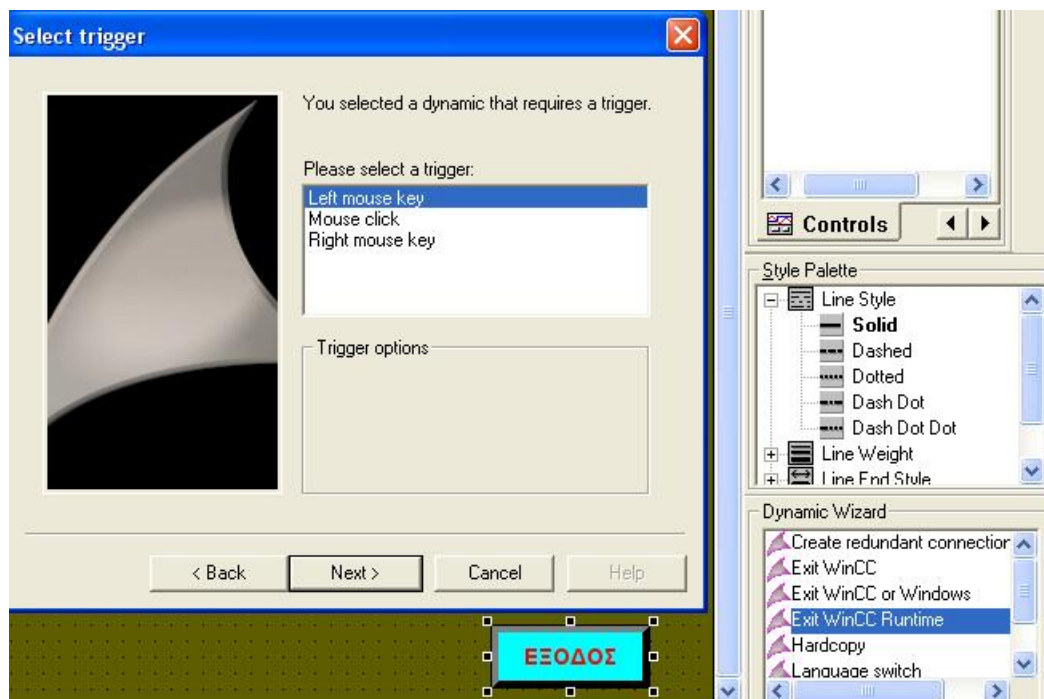
Για να το προγραμματίσουμε να επιτελεί τη συγκεκριμένη λειτουργία, εφαρμόζουμε κάτι το οποίο μέχρι τώρα δεν είχαμε αναφέρει. Εισάγουμε ένα μπουτόν με τον τετριμμένο τρόπο και, αφού το επιλέξουμε με ένα απλό κλικ πάνω του, πηγαίνουμε κάτω δεξιά στον “*Dynamic Wizard*” και επιλέγουμε “*Exit WinCC Runtime*”. Τότε εμφανίζεται ένας οδηγός με συνεχόμενα παράθυρα, στα οποία επιλέγουμε ότι θέλουμε να επιτελείται η συγκεκριμένη λειτουργία με αριστερό κλικ και πατάμε “*Finish*” (εικόνες 5.6, 5.7, 5.8). Έτσι δημιουργήσαμε ένα μπουτόν που αλληλεπιδρά με το περιβάλλον εργασίας μας και δείχνει την μεγάλη ευελιξία του WinCC.

Αξιοσημείωτο είναι το γεγονός ότι εάν περιηγηθούμε στις ιδιότητες του μπουτόν και κάνουμε διπλο κλικ στο σημείο που δείχνει η εικόνα 5.8, παρατηρούμε ότι τώρα έχουμε έτοιμες διαδικασίες γραμμένες σε γλώσσα προγραμματισμού C. Αυτό ενισχύει τον προηγούμενο ισχυρισμό μας περί ευελιξίας του πακέτου, αφού ένας έμπειρος προγραμματιστής είναι σε θέση να προχωρήσει σε λεπτομέρειες και λειτουργίες, που με μία πρώτη ματιά φαίνονται, ίσως,

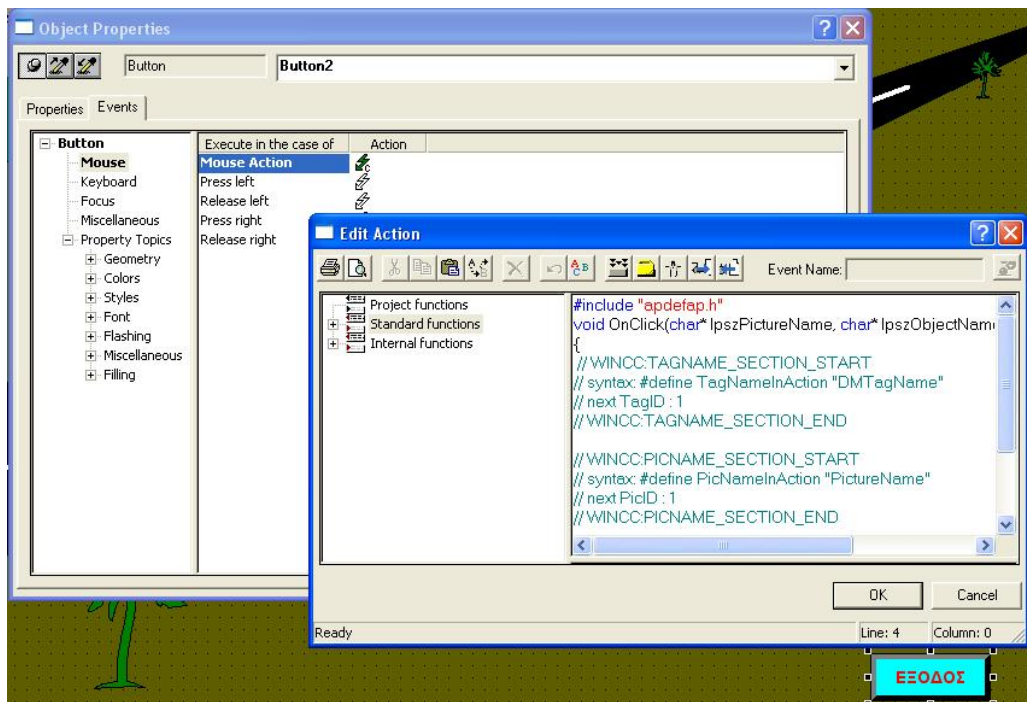
απλησίαστες και να προσαρμόσει το περιβάλλον σύμφωνα με τις ανάγκες του, αλλά με τη δημιουργία των δικών του συναρτήσεων στο περιβάλλον “C-editor” του υπομητήματος “Global Script”.



Εικόνα 5.6. Δημιουργώντας την επιθυμητή δυναμική διασύνδεση εξόδου.

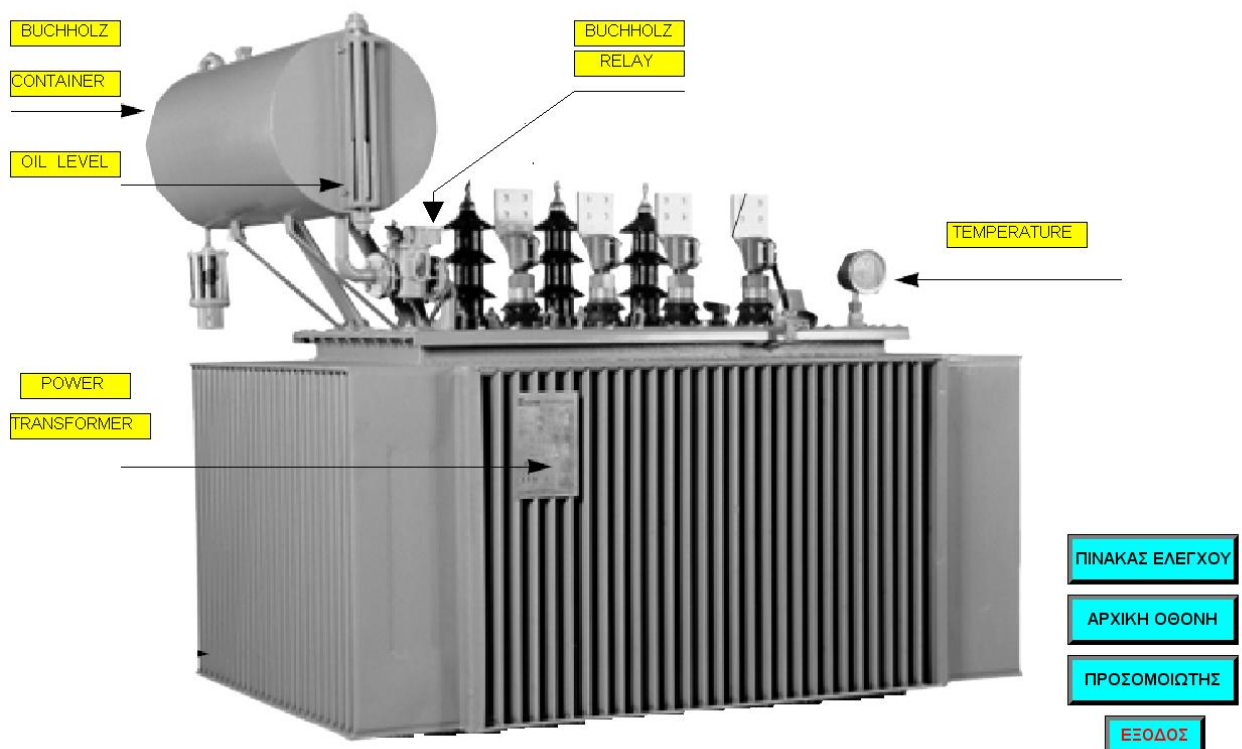


Εικόνα 5.7. Επιλέγοντας έξοδο με αριστερό κλικ.



Εικόνα 5.8. Οι συναρτήσεις C που βρίσκονται πίσω από τη λειτουργία εξόδου.

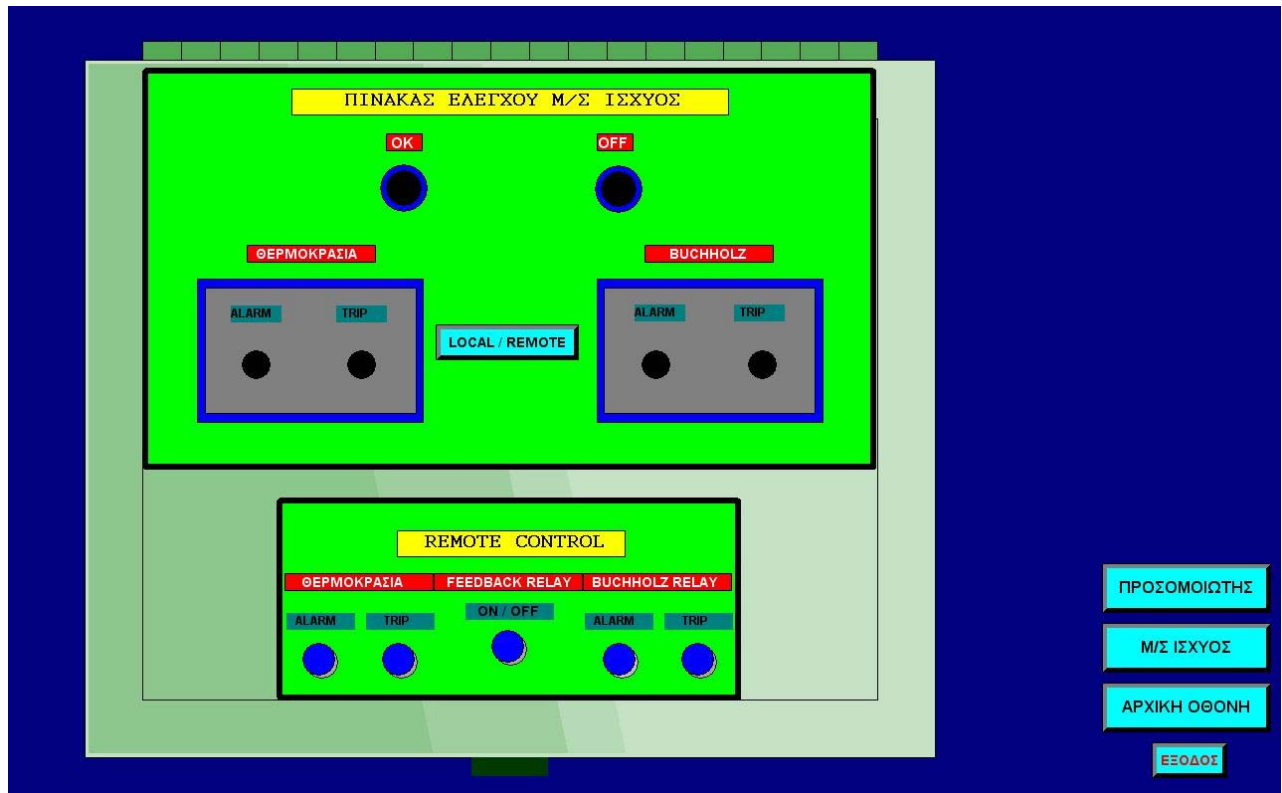
➤ Transformer2.pdl



Εικόνα 5.9. Η δεύτερη εικόνα της εφαρμογής.

Παρατηρούμε ότι η εικόνα αυτή δεν είναι παρά μία παρουσίαση του αντικειμένου που μελετάμε, ώστε να είναι σε θέση κάποιος χρήστης να αντιληφθεί την γενική έννοια του Μ/Σ Ισχύος, μαζί με τους ηλεκτρονόμους για τους οποίους έγινε λόγος παραπάνω. Η προσθήκη μπουτόν μας βοηθάει ώστε να περιηγηθούμε στις εικόνες της εφαρμογής ή να φύγουμε από το *Runtime*.

➤ Transformer3.pdl



Εικόνα 5.10. Η τρίτη εικόνα της εφαρμογής.

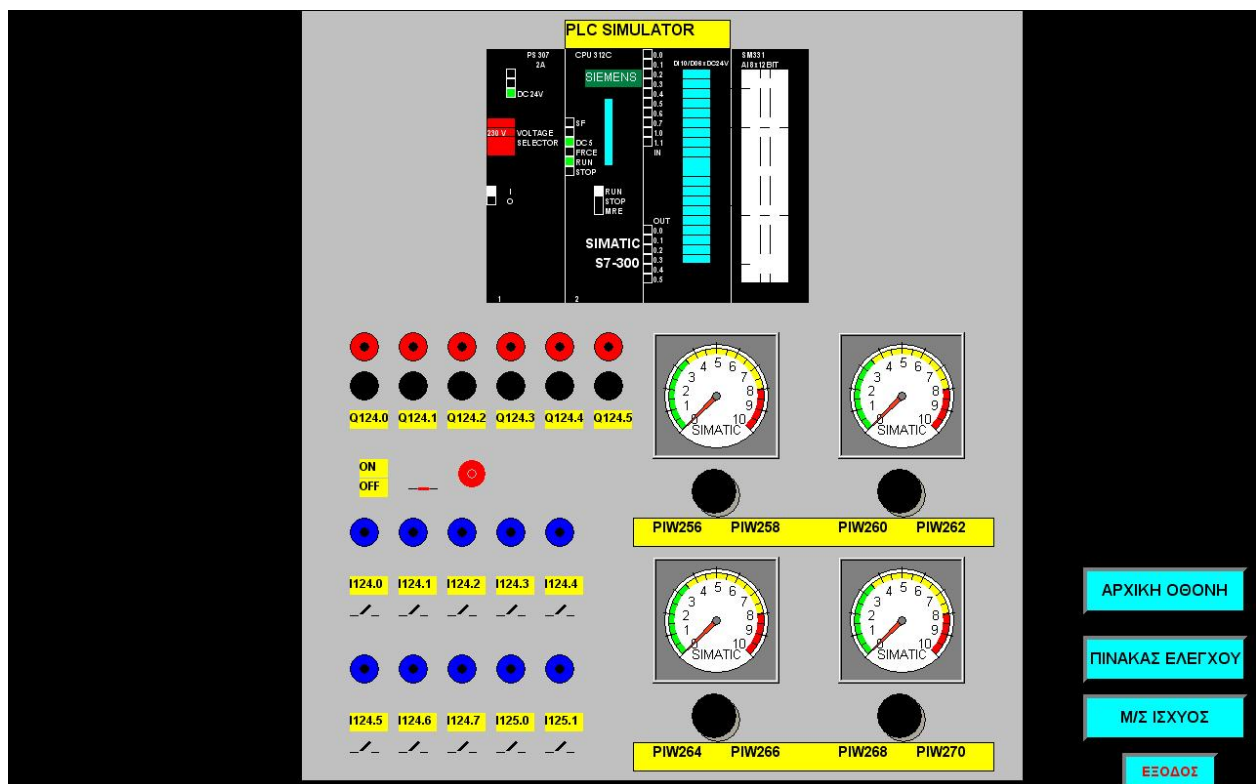
Η τρίτη εικόνα της εφαρμογής μας είναι και η πλέον ουσιαστική, μιας και εδώ γίνεται η εποπτεία και έλεγχος του Μ/Σ. Όλα τα αντικείμενα που έχουν εισαχθεί έχουν διασυνδεθεί δυναμικά με τις κατάλληλες εξωτερικές μεταβλητές και με τρόπο που ήδη έχει περιγραφεί. Το μόνο που απομένει είναι να εξηγήσουμε τον τρόπο λειτουργίας του εικονικού μας πίνακα ελέγχου.

Αρχίζοντας, παρατηρούμε την ύπαρξη μπουτόν με την ετικέτα “LOCAL / REMOTE”. Αυτό το μπουτόν καθορίζει και το είδος του ελέγχου, δηλαδή μόλις κάνουμε αριστερό κλικ πάνω του έχουμε τοπικό έλεγχο, οπότε τα σήματα προέρχονται μόνο από τους διακόπτες του προσομοιωτή, ενώ με δεξί κλικ τα σήματα αυτά αγνοούνται και πλέον μεταφερόμαστε στο πλαίσιο “REMOTE CONTROL”. Εκεί, για εκπαιδευτικούς λόγους, προσομοιώνουμε τα σήματα

σφαλμάτων πατώντας τα αντίστοιχα μπουτόν, όπως και προσομοιώνουμε τις επαφές του ηλεκτρονόμου ανάδρασης. Η λογική των μπουτόν είναι ότι με αριστερό κλικ ενεργοποιούνται (γίνονται **TRUE**) οι αντίστοιχες διασυνδεδεμένες μεταβλητές , ενώ με δεξί κλικ απενεργοποιούνται (γίνονται **FALSE**) οι αντίστοιχες μεταβλητές. Θα μπορούσε, τέλος, να παρατηρήσει κανείς ότι, αγνοώντας το πλαίσιο “REMOTE CONTROL”, έχουμε ήδη έτοιμο ένα αξιόπιστο περιβάλλον εποπτείας του Μ/Σ, ένα ολοκληρωμένο σύστημα SCADA .

➤ Simulator.pdl

Τελευταία εικόνα της εφαρμογής μας είναι ο προσομοιωτής. Το αρχείο αυτό το έχουμε ήδη αντιγράψει, μαζί με τις εσωτερικές μεταβλητές και το μόνο που χρειαζόταν να κάνουμε ήταν να διασυνδέσουμε τα αντικείμενά του με τις εξωτερικές μεταβλητές της παρούσας εφαρμογής. Αξίζει να σημειωθεί ότι αν και το πρόγραμμα σε STEP7 δεν το απαιτούσε, δημιουργήσαμε κατάλληλες εξωτερικές μεταβλητές για τα αναλογικά σήματα, με τις οποίες διασυνδέσαμε τα αντικείμενα προσομοίωσης των βολτομέτρων. Προσθέτοντας και τα απαιτούμενα μπουτόν περιήγησης, καταλήξαμε στην εικόνα 5.12, ένα ολοκληρωμένο εικονικό προσομοιωτή για το παρόν περιβάλλον μας.



Εικόνα 5.11. Η τέταρτη εικόνα της εφαρμογής.

ΚΕΦΑΛΑΙΟ 6

Σενάριο 2: Έλεγχος σηματοδοτών

6.1 Αντικείμενο ζητούμενης εφαρμογής.

Σύμφωνα με το παρόν σενάριο, καλούμαστε να υλοποιήσουμε ένα σύστημα SCADA, το οποίο να εποπτεύει σηματοδότες κίνησης σε μία συνηθισμένη διάβαση πεζών. Πρόκειται για μία συνηθισμένη εφαρμογή των PLC, όπου έχουμε και μία διαδραστικότητα από πλευράς του χρήστη μιας και τα σκέλη υλοποίησης είναι τα ακόλουθα:

- Κυκλική προκαθορισμένη λειτουργία των σηματοδοτών τόσο των οχημάτων όσο και των πεζών.
- Δυνατότητα επίσπευσης της αναμονής του πεζού για διάσχιση της διάβασης, με πάτημα των ανάλογων μπουτόν.
- Δημιουργία εικονικού προσομοιωτή ως ξεχωριστής εικόνας εργασίας.

6.2 Διάβαση πεζών και επίσπευση αναμονής.

Σε μία συνήθη διάβαση πεζών, υπάρχουν σηματοδότες οι οποίοι ρυθμίζουν την κυκλοφορία των οχημάτων και σηματοδότες που καθοδηγούν τους πεζούς. Δεδομένου ότι οι πεζοί έχουν κάποιου είδους προτεραιότητα σε τέτοιες περιπτώσεις, σύμφωνα με τον Κώδικα Οδικής Κυκλοφορίας, οι σηματοδότες των πεζών φέρουν από ένα μπουτόν το οποίο εάν πατηθεί, (οποιοδήποτε από αυτά), τότε ο χρόνος αναμονής των πεζών μικραίνει χαρακτηριστικά και τα οχήματα ακινητοποιούνται σε σύντομο χρονικό διάστημα και για περισσότερο χρόνο.

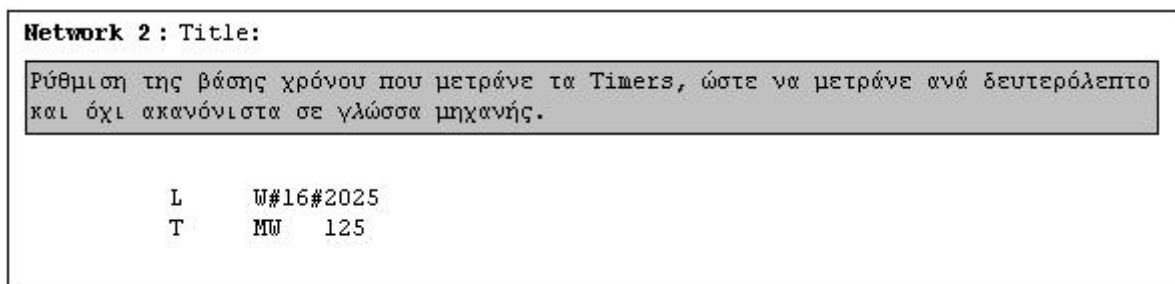
Στην εφαρμογή που υλοποιήσαμε, δημιουργήσαμε μία εικονική διάβαση πεζών με κατάλληλους σηματοδότες και δημιουργήσαμε και τα κατάλληλα μπουτόν για τους πεζούς. Αξιοσημείωτη ήταν η δημιουργία πλαισίων τα οποία παρουσιάζουν το χρόνο που απομένει για να απενεργοποιηθεί ένας σηματοδότης, ώστε να συγκρίνουμε τους χρόνους ανάμονης πριν και μετά το πάτημα των επίμαχων μπουτόν. Το γεγονός αυτό, σε συνδυασμό με τον πίνακα ελέγχου που δημιουργήθηκε, έκανε την εφαρμογή αρκετά λειτουργική και ιδιαίτερα ευπαρουσίαστη.

6.3 Δημιουργία εφαρμογής.

6.3.1 Πρόγραμμα αυτοματισμού.

Κατά τα γνωστά, δημιουργήσαμε το πρόγραμμα αυτοματισμού σε γλώσσα **LADDER** του λογισμικού **STEP7** της **SIEMENS**. Στο πρόγραμά μας χρησιμοποιήσαμε την τεχνική που αναφέρθηκε στο προηγούμενο κεφάλαιο, ώστε να υπάρξει το κατάλληλο υπόβαθρο από πλευράς εξωτερικών μεταβλητών για τον απομακρυσμένο έλεγχο από το WinCC.

Ένα πρόβλημα το οποίο αντιμετωπίσαμε κατά την προεπισκόπηση του, υπό κατασκευή, περιβάλλοντος εργασίας μας ήταν η πραγματική αναπαράσταση του χρόνου που μέτραγαν αντίστροφα τα Timers, τύπου “ON-DELAY”, τα οποία χρησιμοποιήσαμε στο πρόγραμμα αυτοματισμού (σχήμα 6.1). Συγκεκριμένα, παρατηρήσαμε ότι ο χρόνος εμφανιζόταν με μία ακανόνιστη μορφή, η οποία έμοιαζε με γλώσσα χαμηλού επιπέδου, επομένως δυσνόητη στον άνθρωπο. Η λύση δόθηκε αλλάζοντας την βάση χρόνου με την οποία θέλαμε να μετράνε τα Timers. Συγκεκριμένα, τα ρυθμίσαμε να μετράνε ανά δευτερόλεπτο, με τη βοήθεια της εντολής που φαίνεται στο σχήμα 6.2.



Σχήμα 6.2. Καθορισμός της βάσης χρόνου.

Είναι αξιοσημείωτο το γεγονός ότι ο καθορισμός της βάσης χρόνου έγινε σε γλώσσα προγραμματισμού STL. Η γλώσσα αυτή, η οποία έχει αρκετές ομοιότητες με τη γλώσσα assembly, η οποία χρησιμοποιείται για να προγραμματίσει απευθείας μικροπολογιστικές μονάδες, παρέχει όλα τα εφόδια για να επέμβουμε και να ρυθμίσουμε ό,τι θεωρούμε απαραίτητο στο πρόγραμά μας, σε αντίθεση με τη γλώσσα LADDER που έχει αρκετούς περιορισμούς. Στις εφαρμογές μας όμως, για λόγους ευκολίας, χρησιμοποιήσαμε κατά κόρον την τελευταία, σε συνδυασμό με μικρά τμήματα κώδικα σε STL, όπως εδώ. Γενικότερα πάντως, σε εφαρμογές πολύπλοκες και μεγάλης κλίμακας, η STL είναι η γλώσσα στην οποία υλοποιούνται μακροσκελή και απαιτητικά προγράμματα αυτοματισμού.

6.3.2 Δημιουργία περιβάλλοντος εργασίας.

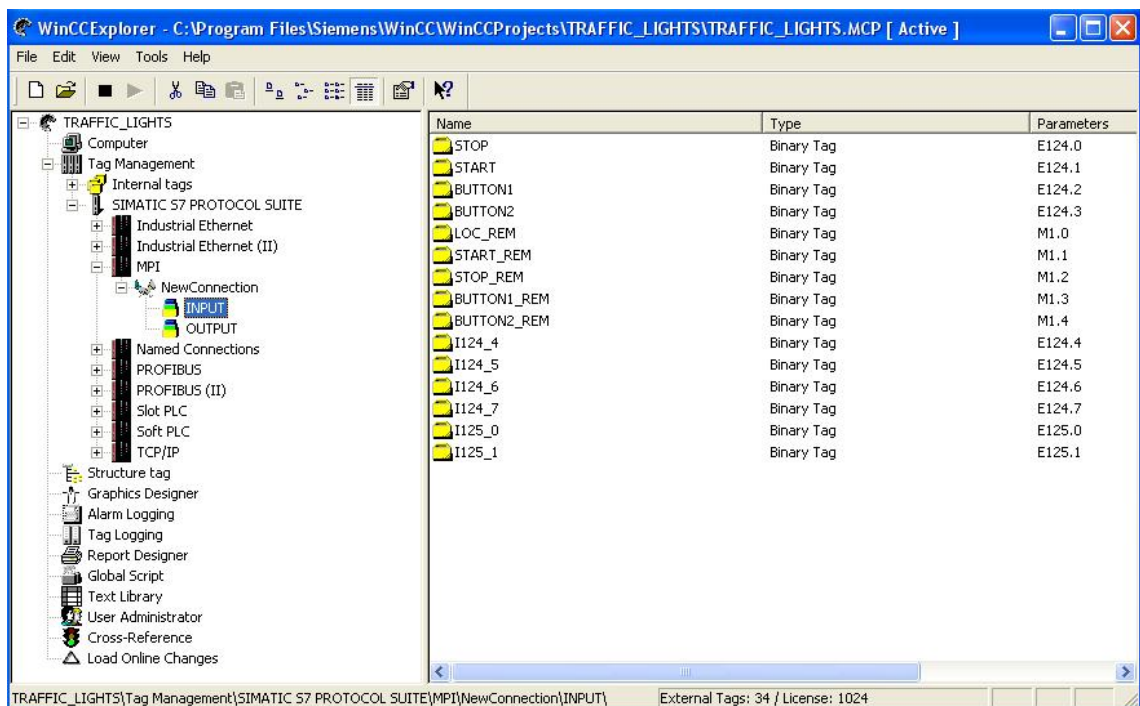
Έχοντας δημιουργήσει και φορτώσει το πρόγραμμά μας στο PLC, περάσαμε στην δημιουργία του περιβάλλοντος εργασίας στο WinCC. Καταρχάς δημιουργήσαμε μία νέα εφαρμογή με το όνομα “traffic_lights” και εφαρμόσαμε τα παρακάτω :

Editor

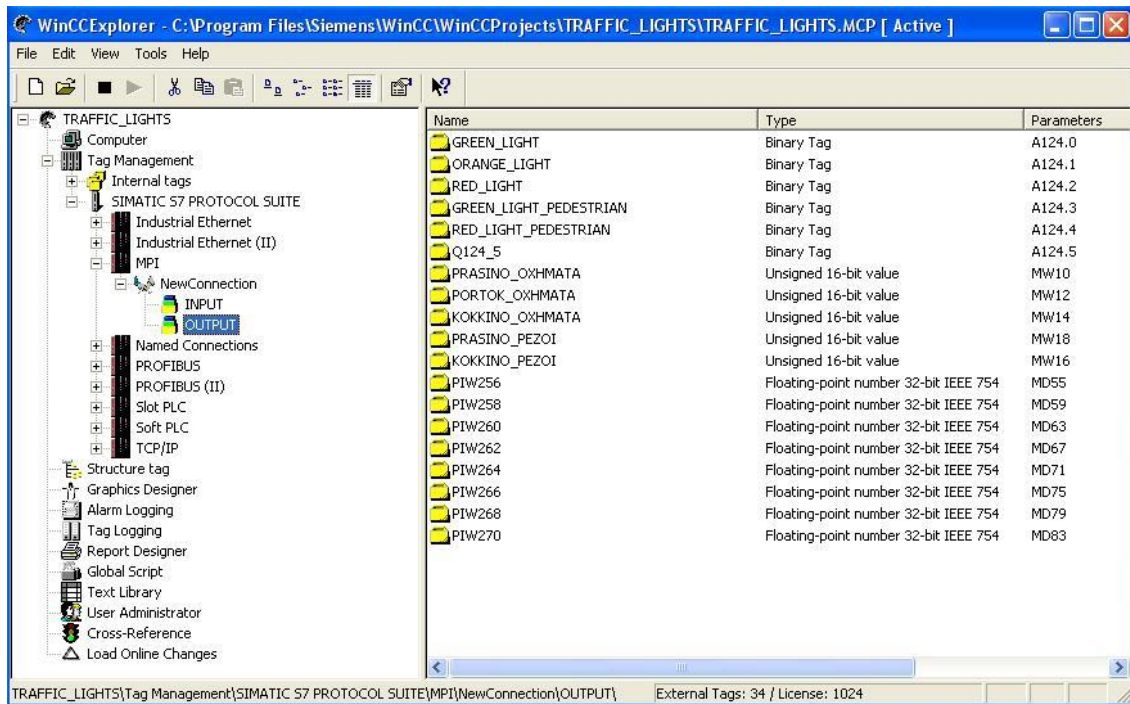
Στο υπομήμα “Computer” εφαρμόσαμε τις ρυθμίσεις που αναφέραμε στο κεφάλαιο 2, σχετικά με την ευπαρουσίαση εικόνα της εφαρμογής μας.

Tag Management

Οι εξωτερικές μεταβλητές που δημιουργήσαμε χωρίστηκαν σε δύο ομάδες (Groups) οι οποίες ονομάστηκαν “INPUT” και “OUTPUT” (εικόνες 6.1 & 6.2).



Εικόνα 6.1. Εξωτερικές μεταβλητές του Group “INPUT”.

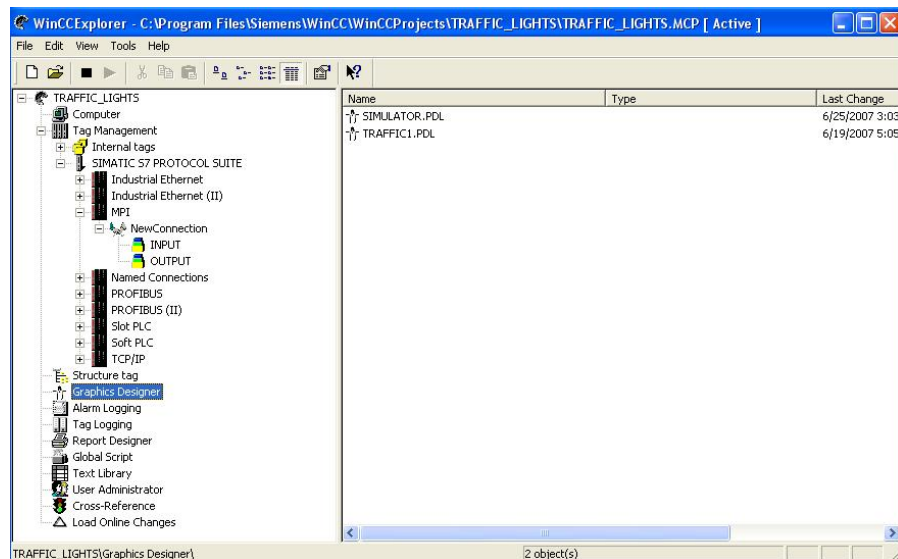


Εικόνα 6.2. Εξωτερικές μεταβλητές του Group “OUTPUT”.

Οι εσωτερικές μεταβλητές, όπως αναφέρθηκε στο κεφάλαιο 4, αντιγράφηκαν μαζί με το αρχείο του προσομοιωτή ώστε να τον χρησιμοποιήσουμε σαν έτοιμη εικόνα εργασίας.

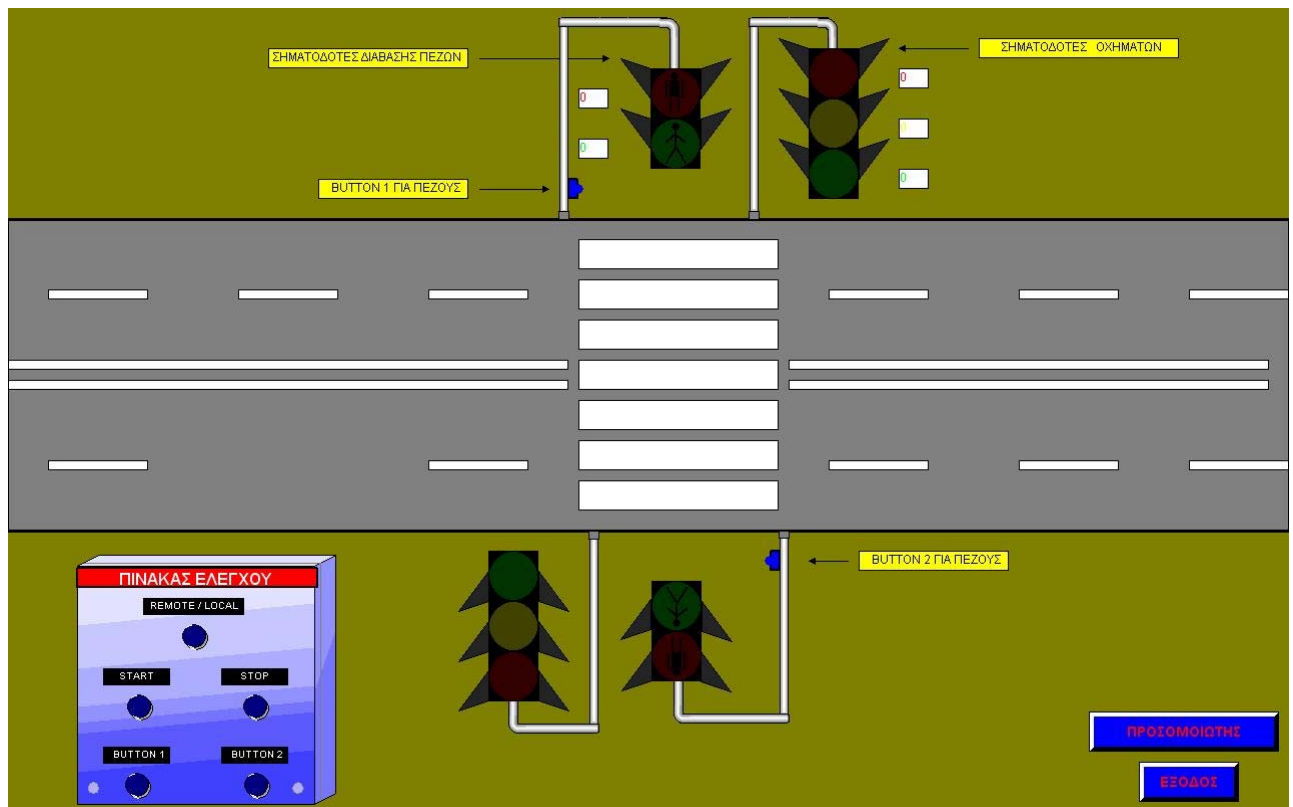
Graphics Designer

Με τη βοήθεια του σχεδιαστή γραφικών δημιουργήσαμε ένα αρχείο και αντιγράψαμε εδώ το ήδη έτοιμο αρχείο του προσομοιωτή (εικόνα 6.3). Στη συνέχεια, παρουσιάζουμε την προεπισκόπηση της κάθε εικόνας εργασίας και περιγράφουμε τη δημιουργία της, όπου κρίνεται αναγκαίο.



Εικόνα 6.3. Τα αρχεία του Graphics Designer.

➤ Traffic1.pdl

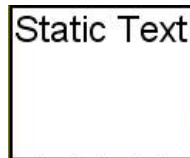


Εικόνα 6.4. Η πρώτη εικόνα της εφαρμογής.

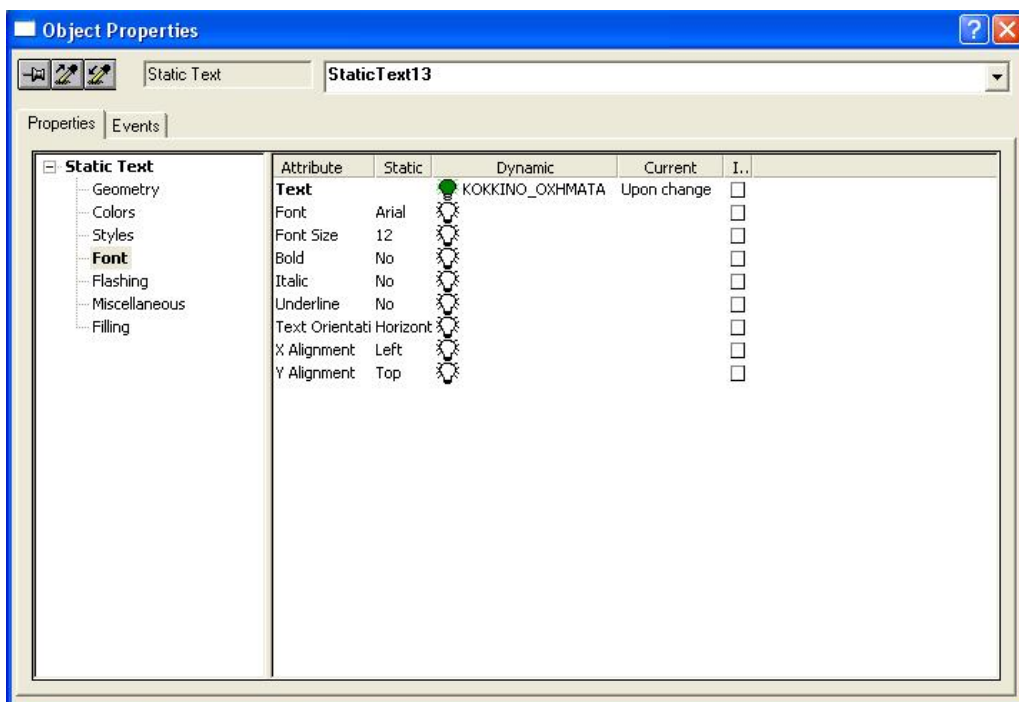
Η εικόνα αυτή περιλαμβάνει όλες τις ζητούμενες λειτουργίες ελέγχου της εφαρμογής. Όπως παρατηρεί κανείς, επιλέξαμε να σχεδιάσουμε τους σηματοδότες κατοπτρικά, με άξονα συμμετρίας το οδόστρωμα, ώστε να δημιουργήσουμε ένα πιο εντυπωσιακό πλάνο σχεδίασης. Όλα τα αντικείμενα που χρησιμοποιήσαμε τα βρήκαμε έτοιμα στη βιβλιοθήκη, ενώ χρησιμοποιώντας τα εργαλεία σχεδίασης, καταφέραμε να δημιουργήσουμε ένα πιστό, εικονικό περιβάλλον.

Ο πίνακας ελέγχου έχει επιλεγεί σε μικρότερες διαστάσεις ενσωματωμένος στην αρχική εικόνα, για το λόγο ότι οι ενέργειες μας τώρα έχουν άμεσο αντίκτυπο στη συμπεριφορά του συστήματος και είναι ανάγκη να το παρακολουθούμε άμεσα. Ειδικότερα, μόλις πατήσουμε ένα από τα μπουτόν των πεζών (“**BUTTON 1**” ή “**BUTTON 2**”), παρακολουθούμε άμεσα την αλλαγή των χρόνων στα ειδικά πλαίσια και τη μεταβολή όλων των σηματοδοτών προς εξυπηρέτηση των πεζών. Επιπλέον, υπάρχει και εδώ μπουτόν (“**REMOTE / LOCAL**”) για απομακρυσμένο ή τοπικό έλεγχο του συστήματος, ενώ για πρώτη φορά εισάγουμε μπουτόν εκκίνησης (“**START**”) και παύσης (“**STOP**”) του προγράμματος αυτοματισμού. Ο τρόπος λειτουργίας των μπουτόν αυτών είναι ο συνήθης, δηλαδή με αριστερό κλικ πάνω τους ενεργοποιούμε (τις κάνουμε **TRUE**) τις εξωτερικές μεταβλητές με τις οποίες είναι δυναμικά διασυνδεδεμένα και με δεξί κλικ τις απενεργοποιούμε (τις κάνουμε **FALSE**).

Καινούριο στοιχείο το οποίο συναντάμε εδώ είναι η απεικόνιση του χρόνου των διαφόρων Timers. Το αντικείμενο που εισάγαμε για να επιτύχουμε αυτή τη λειτουργία είναι το “**Static text**” (σχήμα 6.1). Μάλιστα, δεν είναι η πρώτη φορά που το χρησιμοποιούμε, αφού ο,τι ετικέτες δημιουργούμε στον σχεδιαστή γραφικών γίνονται με το αντικείμενο αυτό, αλλά είναι η πρώτη φορά που αλλάζουμε κάποια από τις ιδιότητές του, ώστε να το διασυνδέσουμε με κάποια εξωτερική μεταβλητή.



Σχήμα 6.1. Το Static text

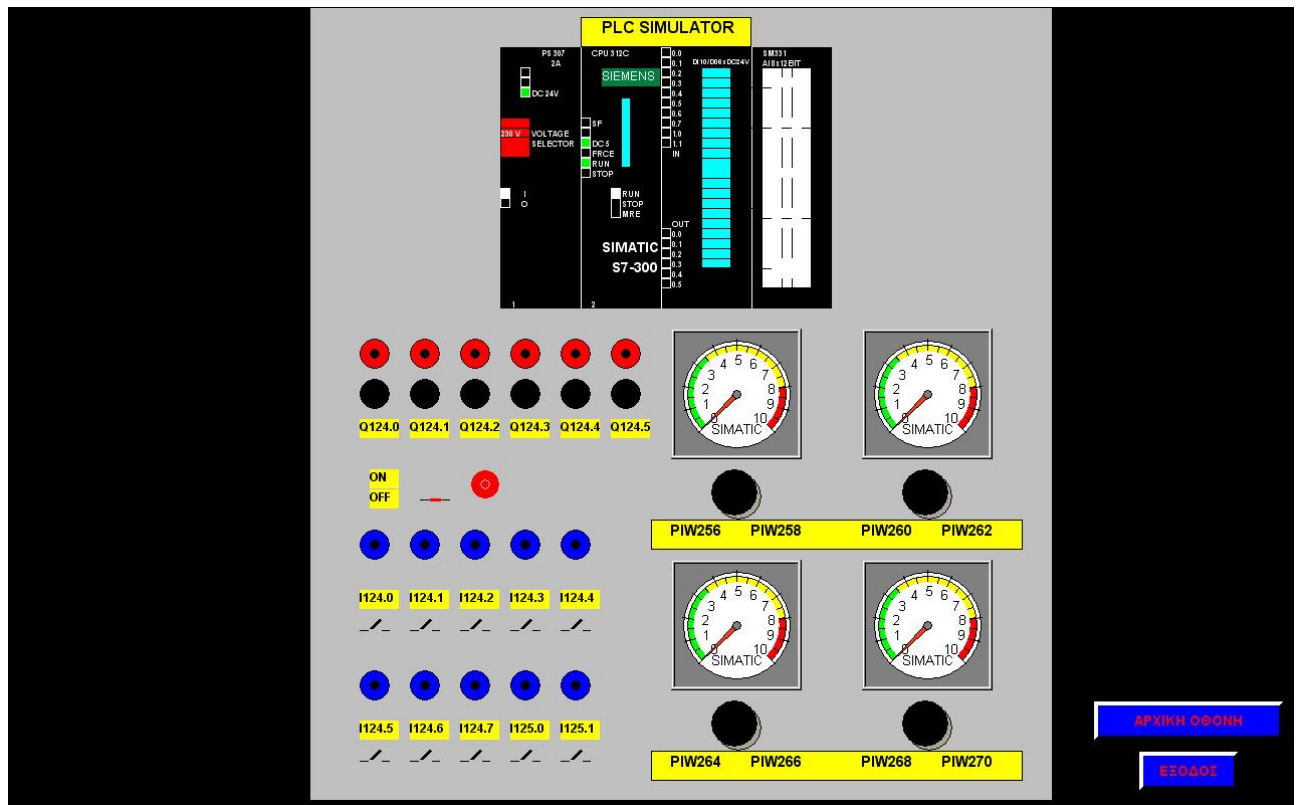


Εικόνα 6.5. Ρυθμίζοντας την δυναμική διασύνδεση του Static text.

Όπως φαίνεται και στην εικόνα 6.5, στο πλαίσιο των ιδιοτήτων του αντικειμένου αυτού, διασυνδέουμε την ιδιότητα “Text” με την εξωτερική μεταβλητή της αρεσκείας μας. Εδώ έχουμε τη μεταβλητή “KOKKINO_OXHMATA”, η οποία αναπαριστά το χρόνο που απομένει ώστε ο κόκκινος σηματοδότης των οχημάτων να απενεργοποιηθεί. Η μεταβλητή αυτή είναι απευθείας συνδεδεμένη με την έξοδο “BCD” του αρμόδιου Timer και έτσι σαν κείμενο μεταφέρει τον ζητούμενο χρόνο μετρώντας αντίστροφα. Ακριβώς την ίδια διαδικασία ακολουθήσαμε, ώστε να διασυνδέσουμε και τα άλλα πλαίσια με τις κατάλληλες μεταβλητές.

➤ Simulator.pdl

Τελευταία εικόνα είναι και πάλι ο προσομοιωτής (εικόνα 6.6). Ομοίως, αντιγράψαμε το ήδη έτοιμο αρχείο με τις εσωτερικές μεταβλητές που επαναλαμβάνονται σε όλες τις εφαρμογές. Επίσης, δημιουργήσαμε τις κατάλληλες εξωτερικές μεταβλητές για τα αναλογικά σήματα και προσθέσαμε τα κατάλληλα μπουτόν περιήγησης στο συνολικό περιβάλλον της εφαρμογής.



Εικόνα 6.6. Ο προσομοιωτής της εφαρμογής.

ΚΕΦΑΛΑΙΟ 7

Σενάριο 3: Έλεγχος συστήματος δεξαμενών

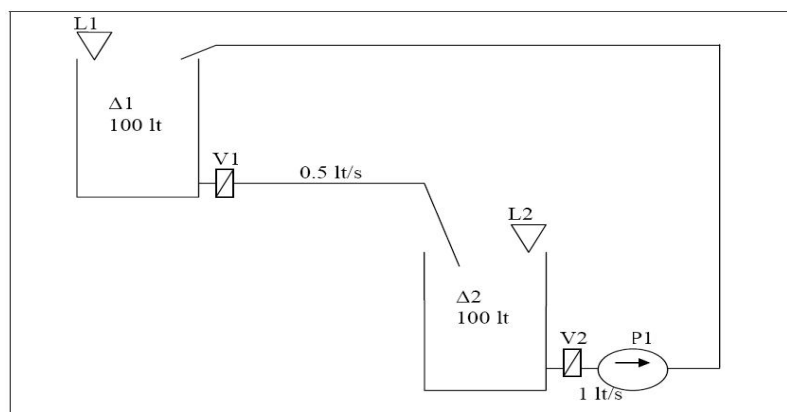
7.1 Αντικείμενο ζητούμενης εφαρμογής.

Η παρούσα εφαρμογή αφορά την υλοποίηση ενός συστήματος SCADA, το οποίο να υλοποιεί και να εποπτεύει την πλήρωση δύο διασυνδεδεμένων δεξαμενών με ρευστό, τόσο τοπικά όσο και εξ'αποστάσεως. Συγκεκριμένα, τα σκέλη τα οποία καλούμαστε να υλοποιήσουμε είναι τα ακόλουθα:

- Αυτόματη/Ημιαυτόματη λειτουργία του συστήματος.
- Καθορισμός της λειτουργίας του συστήματος τοπικά και απομακρυσμένα με επιπλέον καθορισμό τελικής στάθμης του ρευστού από τον χρήστη .
- Δημιουργία εικονικού προσομοιωτή ως ξεχωριστής εικόνας εργασίας.

7.2 Σύστημα δύο διασυνδεδεμένων δεξαμενών.

Στην περίπτωση αυτή έχουμε το σύστημα των δύο δεξαμενών που φαίνεται στο σχήμα 7.1. Πρόκειται για τυπικές δεξαμενές οι οποίες διασυνδέονται με δύο σωληνώσεις. Μία σωληνώση ελεγχόμενη από την ηλεκτρονική δίοδη βαλβίδα V1, με φυσική ροή από την δεξαμενή 1 στην δεξαμενή 2, ικανότητας 0,5 lt/s. Η δεύτερη σωληνώση επιτρέπει τη ροή του ρευστού από την δεξαμενή 2 στην δεξαμενή 1, διαμέσου μίας αντλίας P1 ικανότητας 1 lt/s και μίας δίοδης βάνας V2. Οι δεξαμενές είναι εφοδιασμένες με αισθητήρες στάθμης 0-10 VDC L1 (PIW 256) και L2 (PIW 258) για την 1^η και 2^η δεξαμενή αντίστοιχα.



Σχήμα 7.1. Το σύστημα των δύο δεξαμενών.

Η λειτουργία του συστήματος διακρίνεται σε αυτόματη και ημιαυτόματη :

- Ημιαυτόματη λειτουργία : Ενεργοποιείται όταν η είσοδος *I124.1* έχει λογικό-“1” (TRUE). Στην ημιαυτόματη λειτουργία δηλώνουμε την κατεύθυνση του ρευστού μέσω του διακόπτη B1 (*I124.0*), ο οποίος όταν έχει λογικό-“1” τότε έχουμε κατεύθυνση του ρευστού από την $\Delta 1 \rightarrow \Delta 2$, ενώ η αντίστροφη διαδρομή επιτυγχάνεται όταν ο διακόπτης έχει λογικό-“0”. Εδώ έχουμε τη δυνατότητα να ορίζουμε, σε πραγματικό χρόνο, την τελική στάθμη L3 του υγρού στην δεξαμενή άφιξης, μέσω του ποτενσιόμετρου που είναι συνδεδεμένο με την αναλογική διέθυνση *PIW 260*. Αν δηλώσουμε όγκο μικρότερο από τον υφιστάμενο όγκο της δεξαμενής, τότε το σύστημα μας ειδοποιεί ότι πρέπει να αντιστραφεί η ροή, βγάζει ένα μήνυμα λάθους και σταματά. Το μήνυμα λάθους αφορά την απενεργοποίηση όλων των εν ενεργεία λειτουργιών και εξόδων του συστήματος.
- Αυτόματη λειτουργία : Ενεργοποιείται όταν η είσοδος *I124.1* έχει λογικό-“0” (FALSE). Δηλώνουμε μόνο τον τελικό όγκο του ρευστού, μέσω του ποτενσιόμετρου *PIW 260* και γενικά η ανταπόκριση του συστήματος πρέπει να καλύπτει τις απαιτήσεις του πίνακα 7.1.

Περιπτώσεις	Αποτέλεσμα
1. $L1 > L2$ & $L3 > L2$	V1 ON
2. $L2 > L1$ & $L3 > L1$	V2 ON, P1 ON
3. $L1 > L2$ & $L3 < L2$	V2 ON, P1 ON
4. $L2 > L1$ & $L3 < L1$	V1 ON

Πίνακας 7.1. Περιπτώσεις αυτόματης λειτουργίας.

Η λειτουργία του συστήματος ξεκινάει με πάτημα του μπουτόν “**Start**” (*I124.3*) και διακόπτεται είτε με πάτημα του μπουτόν “**Stop**” (*I124.4*), είτε με ενεργοποίηση του θερμικού προστασίας της αντλίας (*I124.5*). Τα μπουτόν του “**Stop**” και του θερμικού έχουν λειτουργία λογικής ψηφιακής πύλης NOT, δηλαδή όταν είναι και τα δύο FALSE τότε το πρόγραμμα αυτοματισμού της εγκατάστασης λειτουργεί κανονικά, εάν όμως ένα από τα δύο πατηθεί και γίνει TRUE, τότε η λειτουργία της εγκατάστασης διακόπτεται ακαριαία.

7.3 Δημιουργία εφαρμογής.

7.3.1 Πρόγραμμα αυτοματισμού.

Ως γνωστόν, δημιουργήσαμε το πρόγραμμα αυτοματισμού σε γλώσσα **LADDER** του λογισμικού **STEP7** της **SIEMENS**. Στο πρόγραμμά μας χρησιμοποιήσαμε τις απαραίτητες προγραμματιστικές τεχνικές, τις οποίες περιγράψαμε στα προηγούμενα κεφάλαια.

7.3.2 Δημιουργία περιβάλλοντος εργασίας.

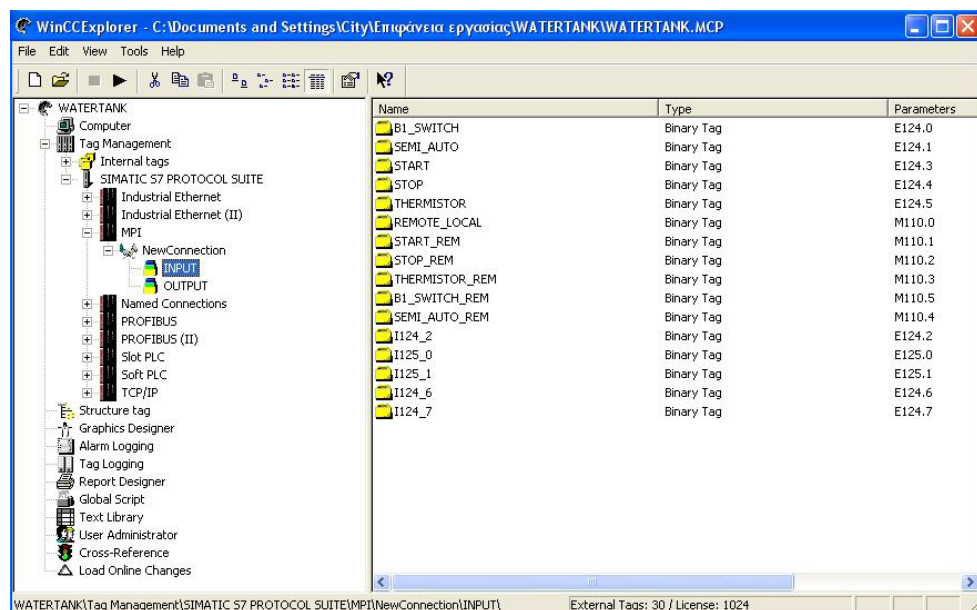
Έχοντας δημιουργήσει και φορτώσει το πρόγραμμά μας στο PLC, περάσαμε στην δημιουργία του περιβάλλοντος εργασίας στο WinCC. Δημιουργήσαμε μία νέα εφαρμογή με το όνομα “watertank” και εφαρμόσαμε τα παρακάτω :

Editor

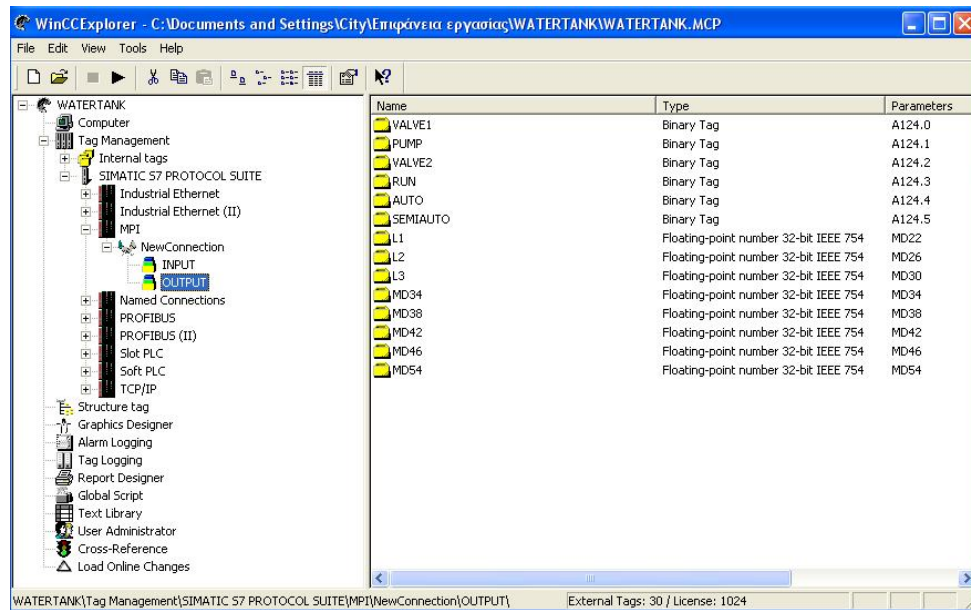
Στο υπομήμα “Computer” εφαρμόσαμε τις ρυθμίσεις που αναφέραμε στο κεφάλαιο 2, σχετικά με την ευπαρουσίαστη εικόνα της εφαρμογής μας.

Tag Management

Οι εξωτερικές μεταβλητές που δημιουργήσαμε χωρίστηκαν σε δύο ομάδες (Groups), οι οποίες ονομάστηκαν “INPUT” και “OUTPUT” (εικόνες 7.1 & 7.2).



Εικόνα 7.1. Εξωτερικές μεταβλητές του Group “INPUT”.

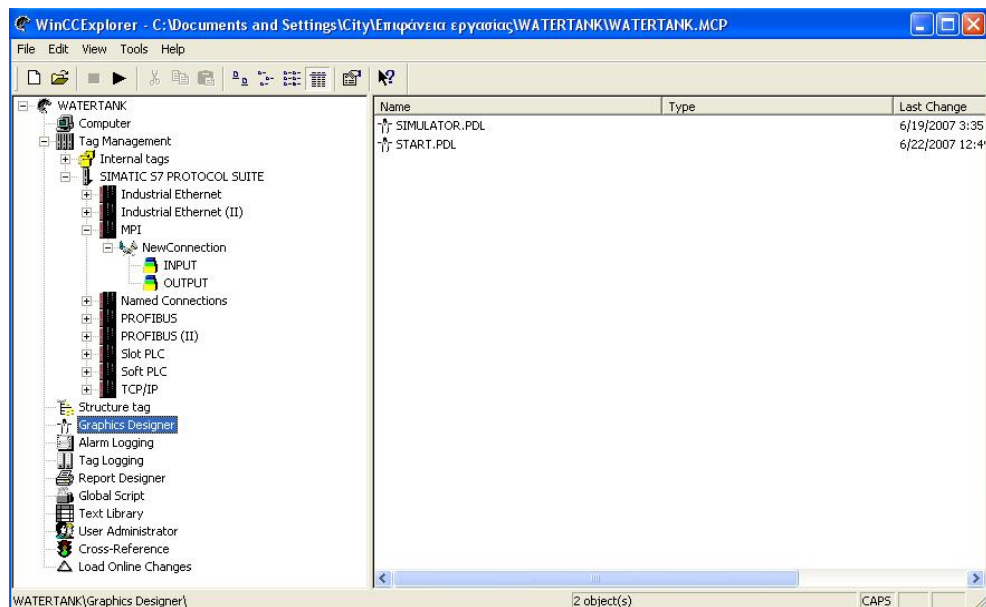


Εικόνα 7.2. Εξωτερικές μεταβλητές του Group “OUTPUT”.

Οι εσωτερικές μεταβλητές, όπως αναφέρθηκε στο κεφάλαιο 4, αντιγράφηκαν μαζί με το αρχείο του προσομοιωτή ώστε να τον χρησιμοποιήσουμε σαν έτοιμη εικόνα εργασίας.

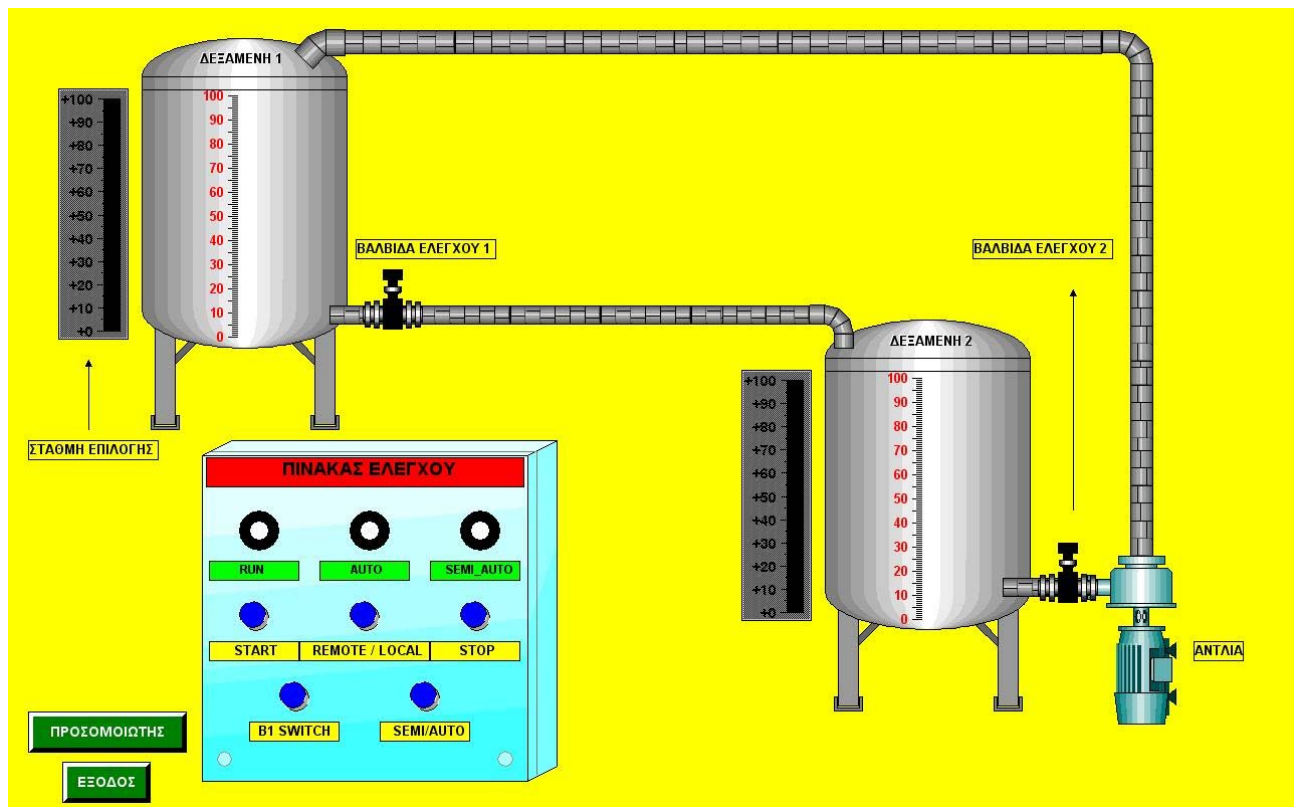
Graphics Designer

Με τη βοήθεια του σχεδιαστή γραφικών δημιουργήσαμε ένα αρχείο και αντιγράψαμε εδώ το ήδη έτοιμο αρχείο του προσομοιωτή (εικόνα 7.3). Στη συνέχεια, παρουσιάζουμε την προεπισκόπηση της κάθε εικόνας εργασίας και περιγράφουμε τη δημιουργία της, όπου κρίνεται αναγκαίο.

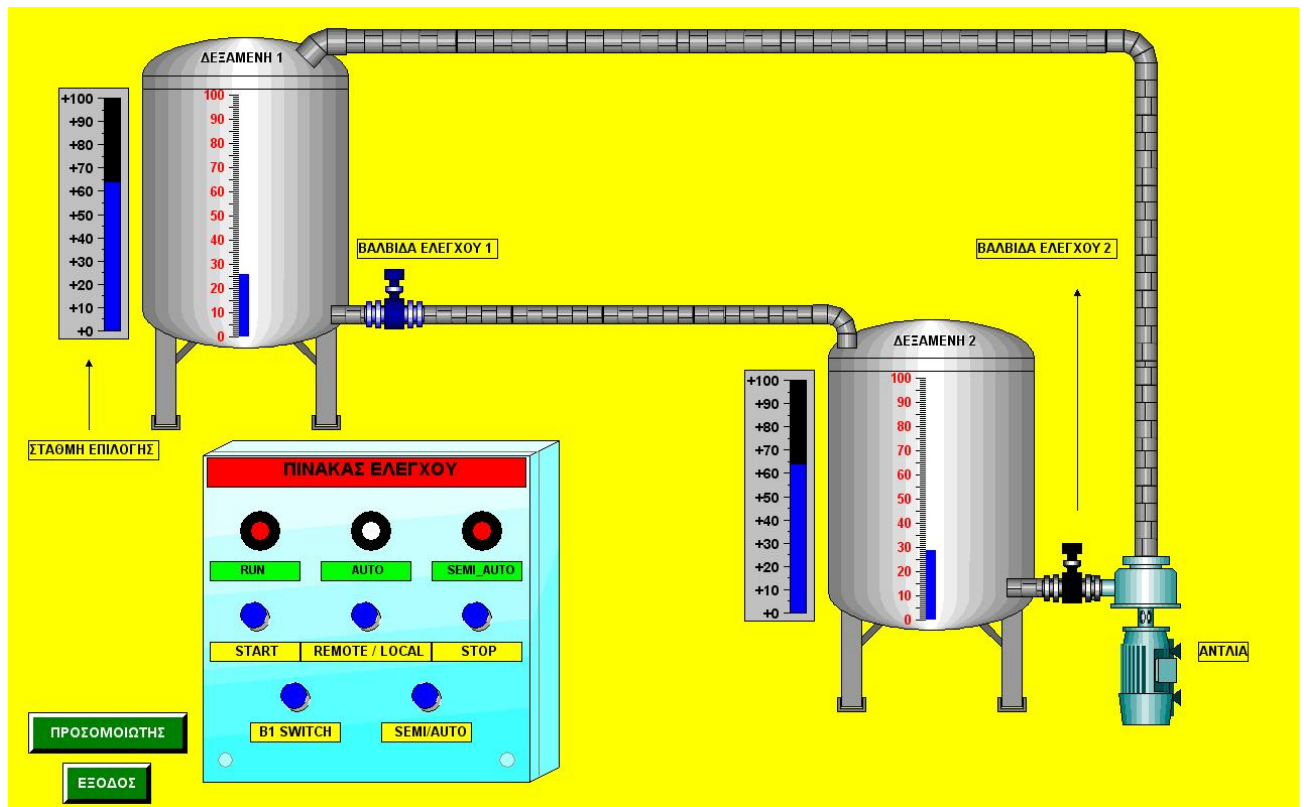


Εικόνα 7.3. Τα αρχεία του Graphics Designer.

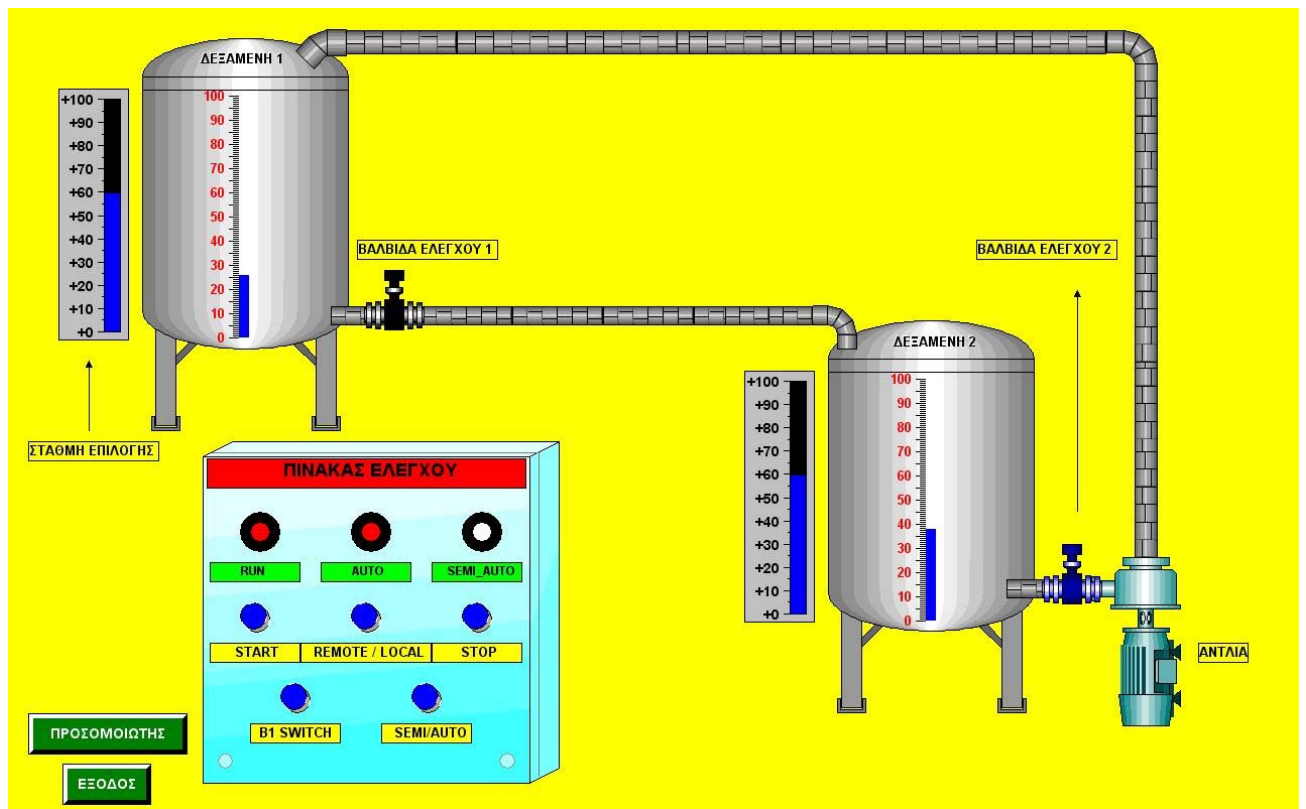
➤ Start.pdl



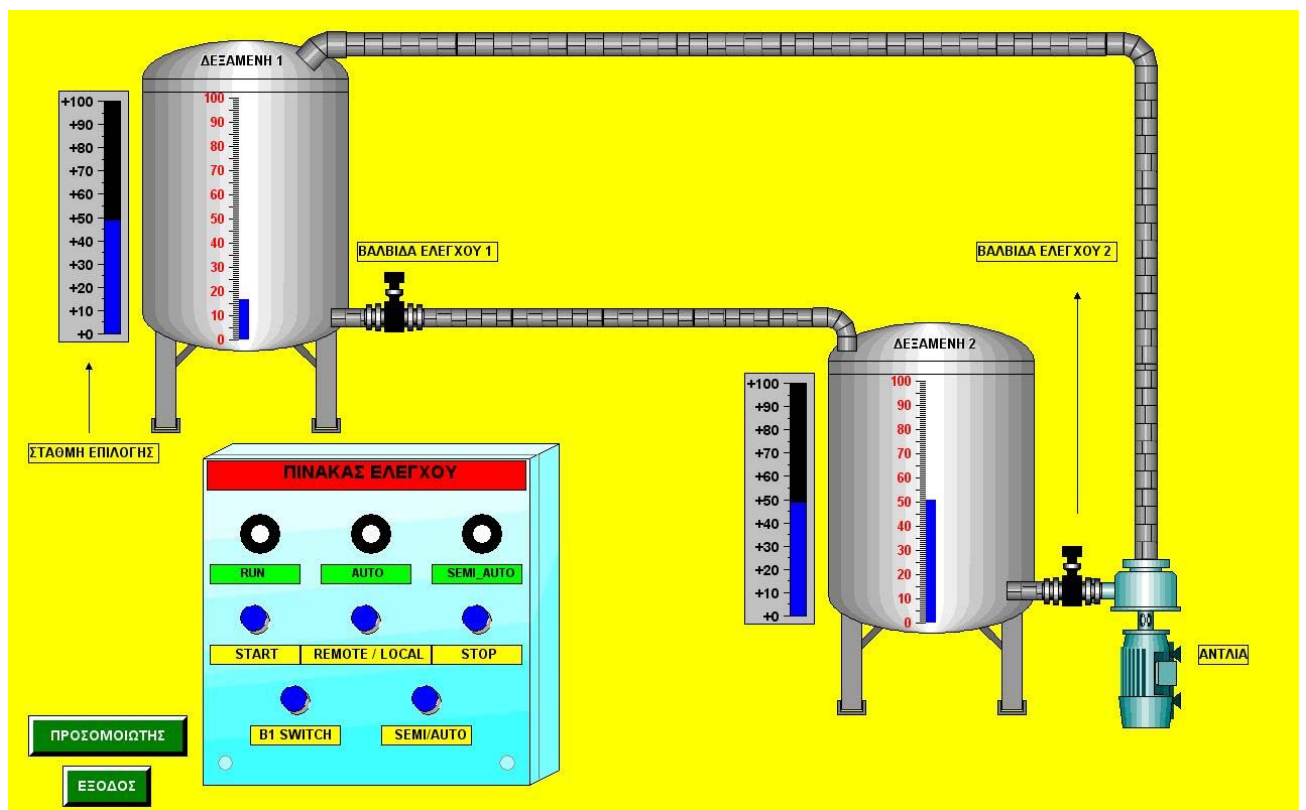
Εικόνα 7.4. Η πρώτη εικόνα της εφαρμογής μας.



Εικόνα 7.5. Η εγκατάσταση σε ημιαντόματη λειτουργία.



Εικόνα 7.6. Η εγκατάσταση σε αυτόματη λειτουργία - περίπτωση 2.

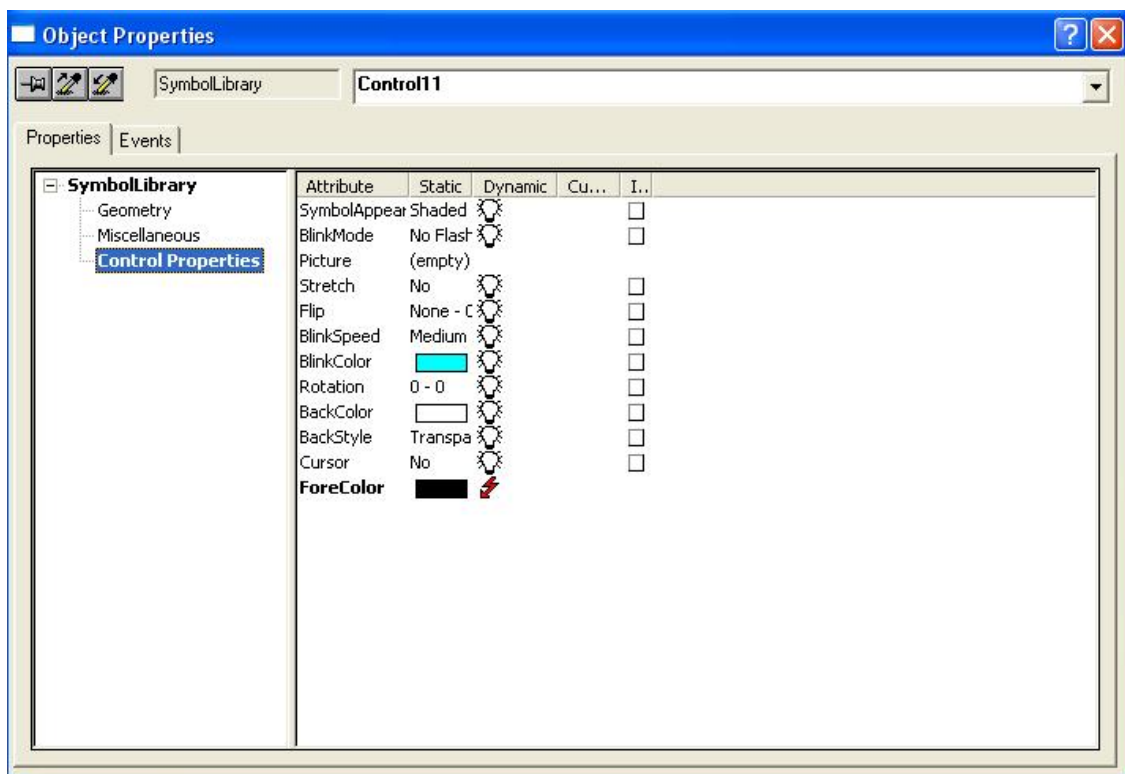


Εικόνα 7.7. Η εγκατάσταση σε κατάσταση σφάλματος.

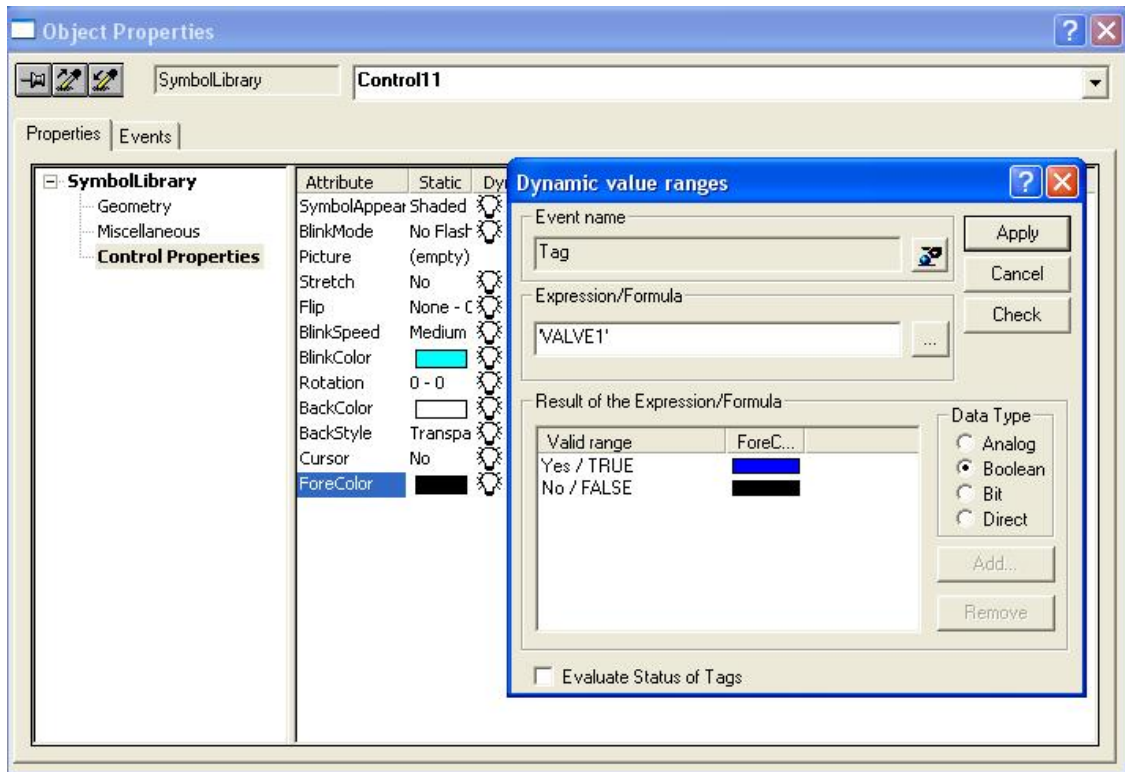
Η παραπάνω εικόνα αποδεικνύει ότι η τρέχουσα εφαρμογή αποτελεί ένα ιδιαίτερα “δυνατό” και ταυτόχρονα κομψό σύστημα SCADA. Επιλέξαμε τον πίνακα ελέγχου να συνυπάρχει στην ίδια εικόνα με όλα τα αντικείμενα, ώστε να βλέπουμε τα αποτελέσματα των ενεργειών μας σε πραγματικό χρόνο.

Καταρχάς, η εγκατάσταση λειτουργεί εξ’ορισμού δεχόμενη εντολές τοπικά, από τον προσομοιωτή μας. Απομακρυσμένος έλεγχος επιτυγχάνεται πατώντας το μπουτόν “**REMOTE / LOCAL**” με αριστερό κλικ, ενώ με δεξί κλικ επαναφέρουμε το σύστημα σε τοπική λειτουργία. Επίσης, σαν μήνυμα λάθους στην περίπτωση αντιστροφής της ροής, κατά την ημιαυτόματη λειτουργία, επιλέξαμε να σβήνουν τα LEDs ακαριαία.

Ξεκινώντας από τα αντικείμενα τα οποία εισάγαμε από τη βιβλιοθήκη, θέσαμε τις βαλβίδες ελέγχου V1 & V2 να έχουν χρώμα μαύρο, σε περίπτωση που είναι κλειστές, ενώ να παίρνουν γαλάζια απόχρωση όταν επιτρέπουν τη διέλευση του υγρού. Ο τρόπος με τον οποίο εφαρμόσαμε τις αλλαγές αυτές φαίνεται στην εικόνα 7.8, όπου έχουμε ανοίξει το σχετικό παράθυρο επιλογών μίας έτοιμης βαλβίδας ελέγχου. Διασυνδέσαμε δυναμικά την επιλογή που σχετίζεται με το χρώμα της βαλβίδας με την κατάλληλη εξωτερική μεταβλητή (εδώ “**VALVE1**”). Εκεί κάναμε τις επιλογές που φαίνονται στην εικόνα 7.9.



Εικόνα 7.8. Διασυνδέοντας δυναμικά τη βαλβίδα ελέγχου.



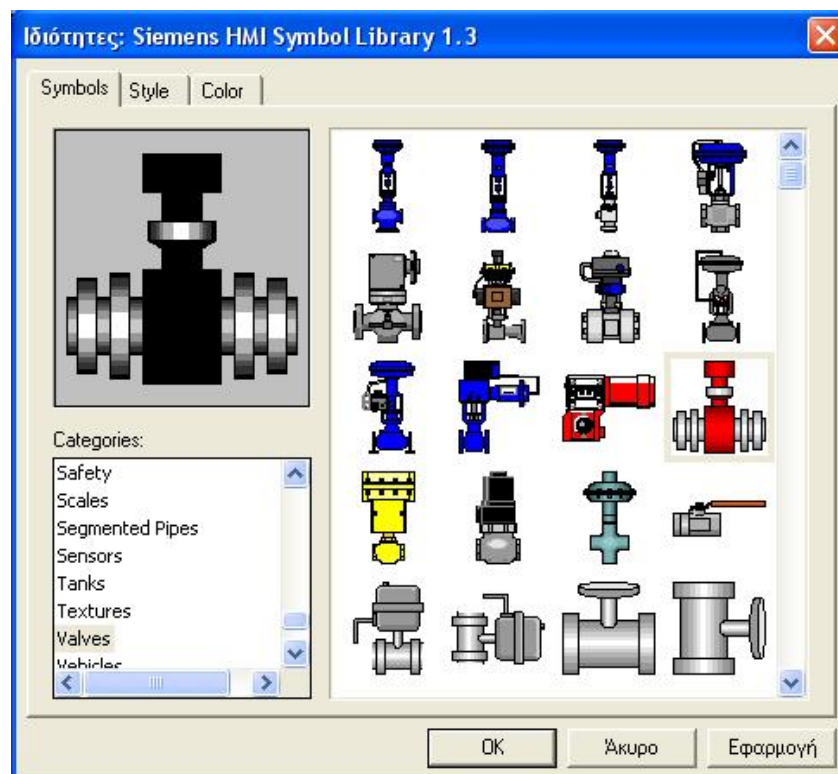
Εικόνα 7.9. Επιλογές διασύνδεσης της βαλβίδας ελέγχου.

Αξίζει να τονίσουμε ότι για να λάβει χώρα η συγκεκριμένη επιλογή πρέπει, πρώτα απ' όλα, να μεταβάλλουμε τις κανονικές ιδιότητες του αντικειμένου. Δηλαδή, με διπλό κλικ πάνω του πηγαίνουμε στο παράθυρο της εικόνας 7.10, όπου επιλέγουμε την καρτέλα "Style" στο πάνω μέρος και εκεί επιλέγουμε το αντικείμενο να είναι "Shaded" (εικόνα 7.11). Αν δεν κάνουμε την επιλογή αυτή και το αφήσουμε στην προκαθορισμένη επιλογή "Original", τότε δεν μπορούμε να μεταβάλλουμε το χρώμα εμφάνισής του, δηλαδή ό,τι χρώμα και να επιλέξουμε αυτό δεν γίνεται αντιληπτό. Έπειτα πατάμε "Εφαρμογή" και "OK" με τη σειρά και αποθηκεύεται η αλλαγή που θέλουμε. Μπορούμε και από εδώ, πηγαίνοντας στην καρτέλα "Color" πάνω δεξιά, να αλλάξουμε τα χρώματα που θέλουμε, απλά δεν μπορούμε να καθορίσουμε κάποια δυναμική διασύνδεση με τον τρόπο αυτό. Την διαδικασία, την οποία μόλις περιγράψαμε, ακολουθήσαμε στις περιπτώσεις μεταβολής ιδιοτήτων αντικειμένων βιβλιοθήκης και δυναμικών μεταβολών αυτών.

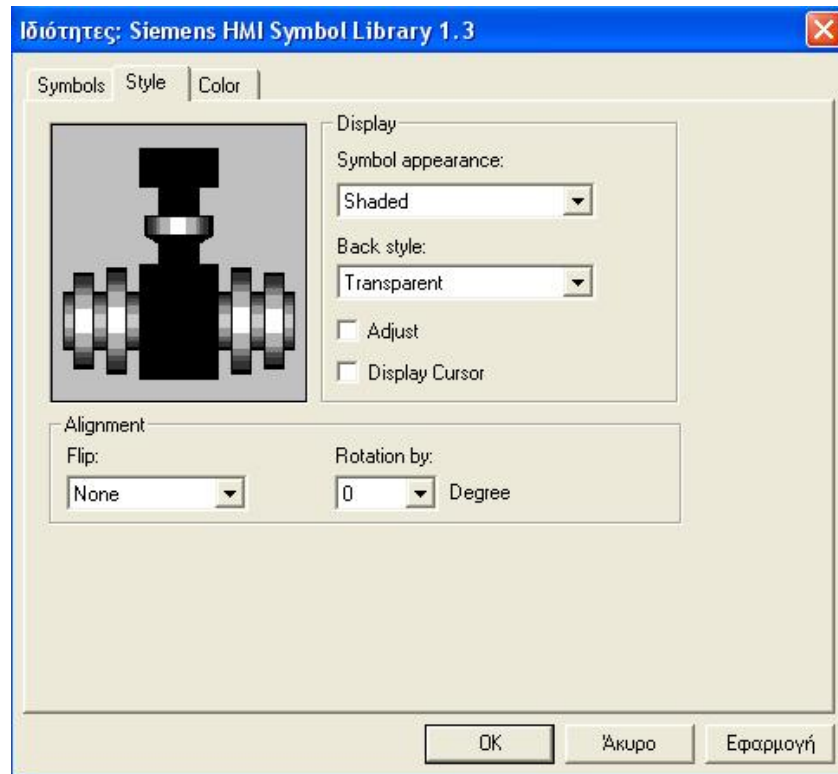
Τα ίδια βήματα ακολουθήσαμε, σε γενικές γραμμές, για τη ρύθμιση των παραμέτρων της αντλίας, με μία επιπλέον ενέργεια. Στην περίπτωση λειτουργίας της αντλίας θέλαμε να "τρέμει" το σχήμα της με σταθερή συχνότητα, δηλαδή να έχει ο χρήστης μία αίσθηση κίνησης του αντικειμένου. Τέτοια δυνατότητα το WinCC δυστυχώς δεν προσφέρει άμεσα, αλλά εμμέσως με μία απλή τεχνική.

Όπως δείχνει και το σχήμα 7.12, πηγαίνουμε στην επιλογή "Blinkspeed", η οποία μας δίνει τη δυνατότητα εναλλαγής του σταθερού χρώματος του σχήματος με το χρώμα της επιλογής

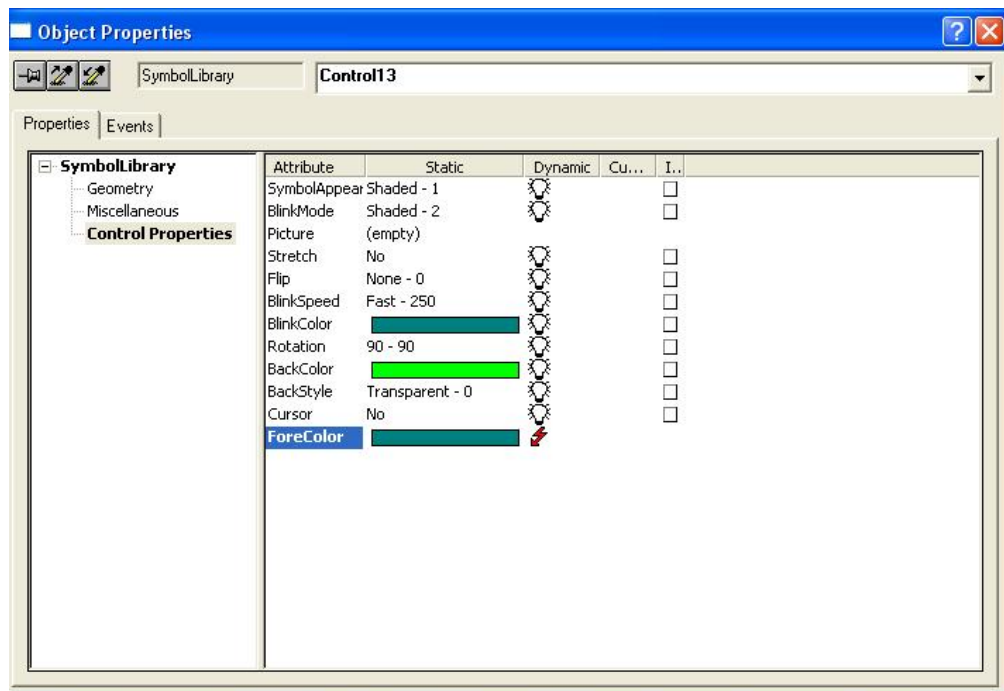
“Blinkcolor”. Ο χρόνος εναλλαγής καθορίζεται στο πλαίσιο δίπλα ακριβώς από την “Blinkspeed”, όπου και επιλέγουμε “Fast – 250”. Ύστερα πηγαίνουμε στην επιλογή “Forecolor”, όπου έχουμε επιλέξει το σταθερό χρώμα και τη διασυνδέουμε δυναμικά με την εξωτερική μεταβλητή “Pump”, η οποία μεταφέρει την κατάσταση της αντλίας (εικόνα 7.13). Εκεί καθορίζουμε σαν σταθερό χρώμα το ίδιο με πριν, όταν η αντλία δε λειτουργεί, ενώ όταν λειτουργεί επιλέγουμε μία οπτικά ανεπαίσθητη παραλλαγή του. Το αποτέλεσμα των ενεργειών μας είναι ότι το αντικείμενο που παριστάνει την αντλία, όταν αυτή δε λειτουργεί, φαίνεται οπτικά ακίνητο αφού εναλλάσσουμε το ίδιο χρώμα με τον εαυτό του, ενώ στην αντίθετη περίπτωση έχουμε μία ελαφριά κίνηση, μιας και εναλλάσσουμε το σταθερό χρώμα με ένα άλλο το οποίο διαφέρει ελάχιστα. Με την τεχνική αυτή κατορθώσαμε να αντιλαμβανόμαστε τη λειτουργία της αντλίας στο περιβάλλον εργασίας, με τρόπο που γίνεται εύκολα αντιληπτός από το ανθρώπινο μάτι και είναι και παράλληλα εντυπωσιακός.



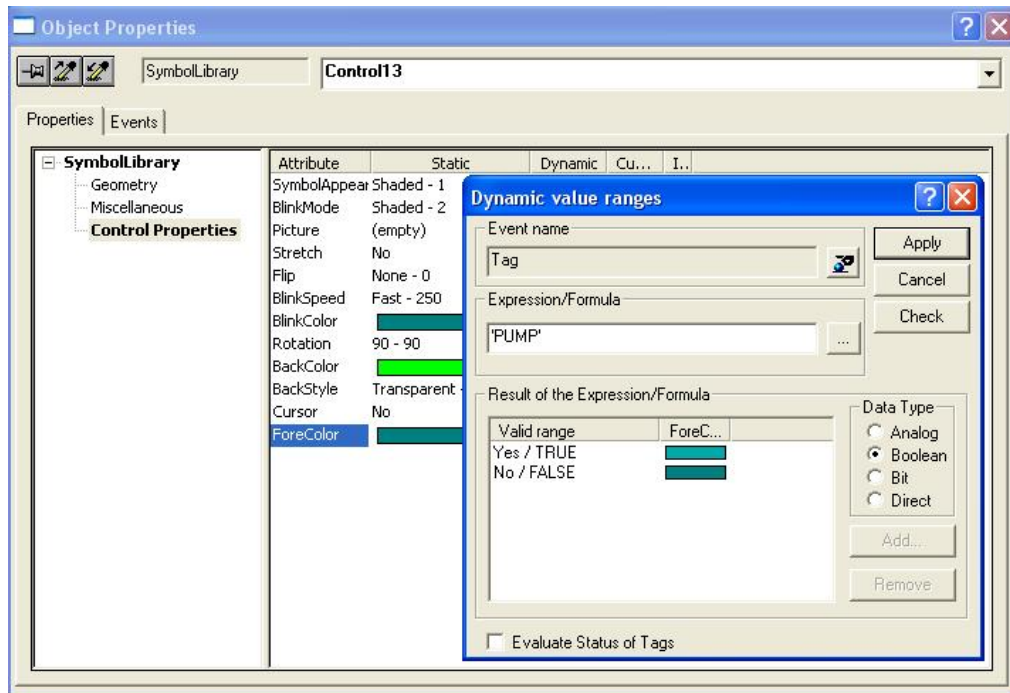
Εικόνα 7.10. Βασικές παράμετροι βαλβίδας ελέγχου.



Εικόνα 7.11. Μεταβάλλοντας τις βασικές παραμέτρους της βαλβίδας ελέγχου.

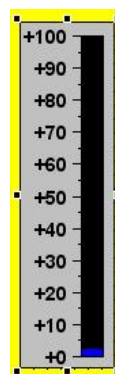


Εικόνα 7.12. Μεταβάλλοντας τις ιδιότητες της αντλίας.



Εικόνα 7.13. Διασυνδέοντας δυναμικά την αντλία.

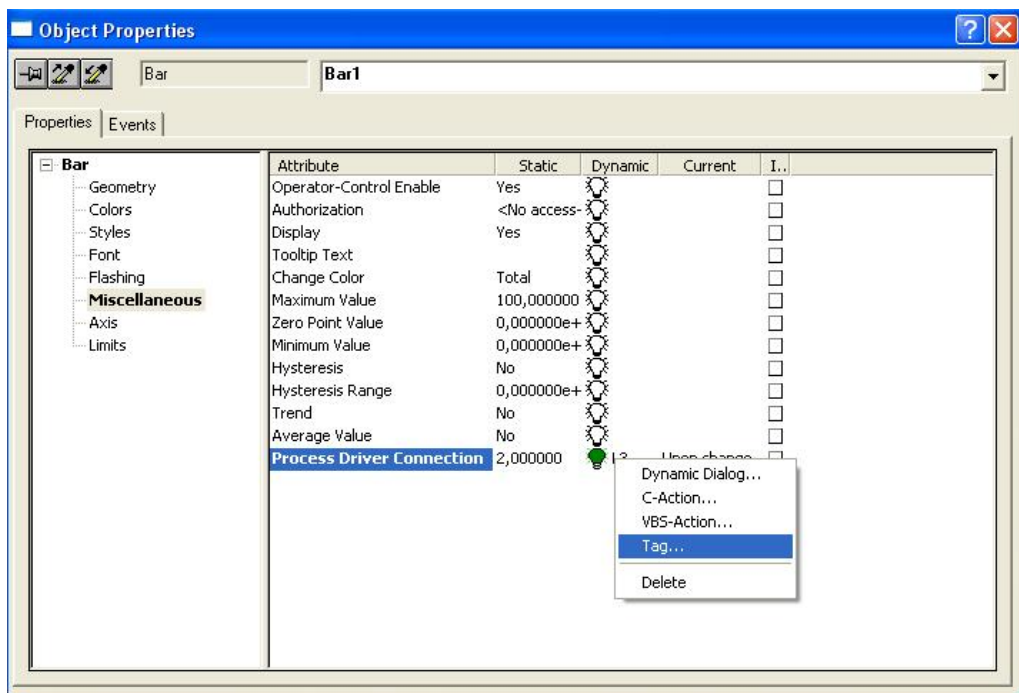
Ένα αντικείμενο το οποίο χρησιμοποιήσαμε εδώ για πρώτη φορά, κατά την ενασχόλησή μας με την παρούσα εργασία, είναι η “**WinCC Bar**” (σχήμα 7.2). Πρόκειται για στοιχείο που χρησιμοποιείται για αναπαράσταση ενός μεγέθους, συνήθως αναλογικού, σε μορφή κατακόρυφης στήλης. Το στοιχείο αυτό το χρησιμοποιήσαμε για να αναπαραστήσουμε τη στάθμη του υγρού, σε δύο περιπτώσεις. Η πρώτη είναι σαν αντικείμενο προσκολλημένο στην εμπρόσθια πλευρική επιφάνεια μίας δεξαμενής, ώστε να βλέπουμε ανά πάσα στιγμή τη στάθμη του υγρού που αυτή περικλείει. Η άλλη περίπτωση είναι σαν πλαίσιο ξεχωριστό δίπλα στην κάθε δεξαμενή, για να παρατηρούμε τη στάθμη του υγρού την οποία θέτουμε σε κάθε περίπτωση λειτουργίας, σε σχέση με την υπάρχουσα στάθμη της κάθε δεξαμενής. Συμπερασματικά, η πρακτική αξία του αντικειμένου είναι πολύ μεγάλη, αφού βελτιστοποιεί οπτικά την εφαρμογή και την καθιστά άκρως πρακτική στον χειρισμό.



Σχήμα 7.2. WinCC Bar.

Ειδικότερα, ανοίγοντας το παράθυρο ιδιοτήτων του αντικειμένου, με τον γνωστό τρόπο, πηγαίνουμε στην επιλογή “Process Driver Connection” και την διασυνδέουμε δυναμικά με τη μεταβλητή που περιέχει το προς αναπαράσταση αναλογικό μέγεθος (εικόνα 7.14). Προσοχή θα πρέπει να δοθεί στο γεγονός ότι επιλέγουμε “Tag” και όχι “Dynamic connection”, διότι δε θέλουμε να αλλάζουμε κάποιες ιδιότητες δυναμικά, όπως προηγουμένως, απλά θέλουμε η μπάρα μας να αναπαριστά το μέγεθος που περιέχει η μεταβλητή μας. Στη συνέχεια, μεταβάλλουμε το αντικείμενο κατά προτίμηση και οδηγούμαστε στο γνωστό και, ομολογουμένως, καλαίσθητο αποτέλεσμα.

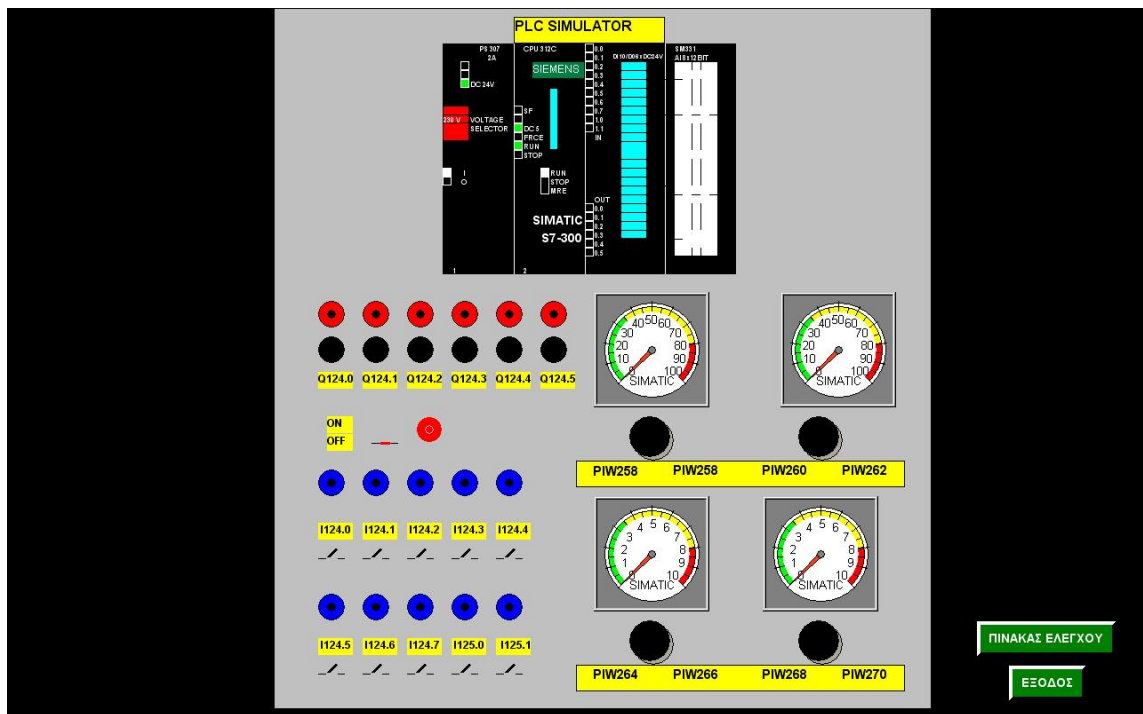
Τέλος, αξίζει να αναφερθεί ότι κατά τον απομακρυσμένο έλεγχο της εγκατάστασης παραλείψαμε κάποιο μπουτόν το οποίο να παραπέμπει στη λειτουργία του θερμικού. Μολονότι τοπικά κάτι τέτοιο το προσομοιώσαμε με ένα διακόπτη εισόδου, στον εξ’αποστάσεως έλεγχο θεωρήσαμε το θερμικό σαν μία αιτία την οποία δεν μπορούμε να καθορίσουμε, όπως συμβαίνει στην πραγματικότητα και απλά δεχόμαστε το αποτέλεσμα της. Έτσι, δημιουργήσαμε έναν ρεαλιστικό πίνακα ελέγχου, ο οποίος θα μπορούσε κάλλιστα να χρησιμοποιηθεί και σε μία πραγματική εφαρμογή, με τις παρούσες ιδιότητές του,



Εικόνα 7.14. Διασυνδέοντας δυναμικά την Bar.

➤ Simulator.pdl

Η εικόνα του προσομοιωτή δημιουργήθηκε ακολουθώντας την ίδια διαδικασία την οποία εφαρμόσαμε και στα προηγούμενα σενάρια εφαρμογών. Διασυνδέσαμε κατάλληλα τις μεταβλητές μας, προσθέσαμε τα κατάλληλα μπουτόν περιήγησης και καταλήξαμε στο αναμενόμενο αποτέλεσμα της εικόνας 7.15.



Εικόνα 7.15. Η εικόνα του προσομοιωτή

ΚΕΦΑΛΑΙΟ 8

Σενάριο 4: Έλεγχος & Προστασία Γραμμής Μεταφοράς Διπλού Κυκλώματος

8.1 Αντικείμενο ζητούμενης εφαρμογής.

Η παρούσα εφαρμογή αφορά την υλοποίηση ενός συστήματος SCADA, το οποίο να ελέγχει και να προστατεύει μία γραμμή μεταφοράς διπλού κυκλώματος, τόσο τοπικά όσο και εξ'αποστάσεως. Συγκεκριμένα, τα σκέλη τα οποία καλούμαστε να υλοποιήσουμε είναι τα ακόλουθα:

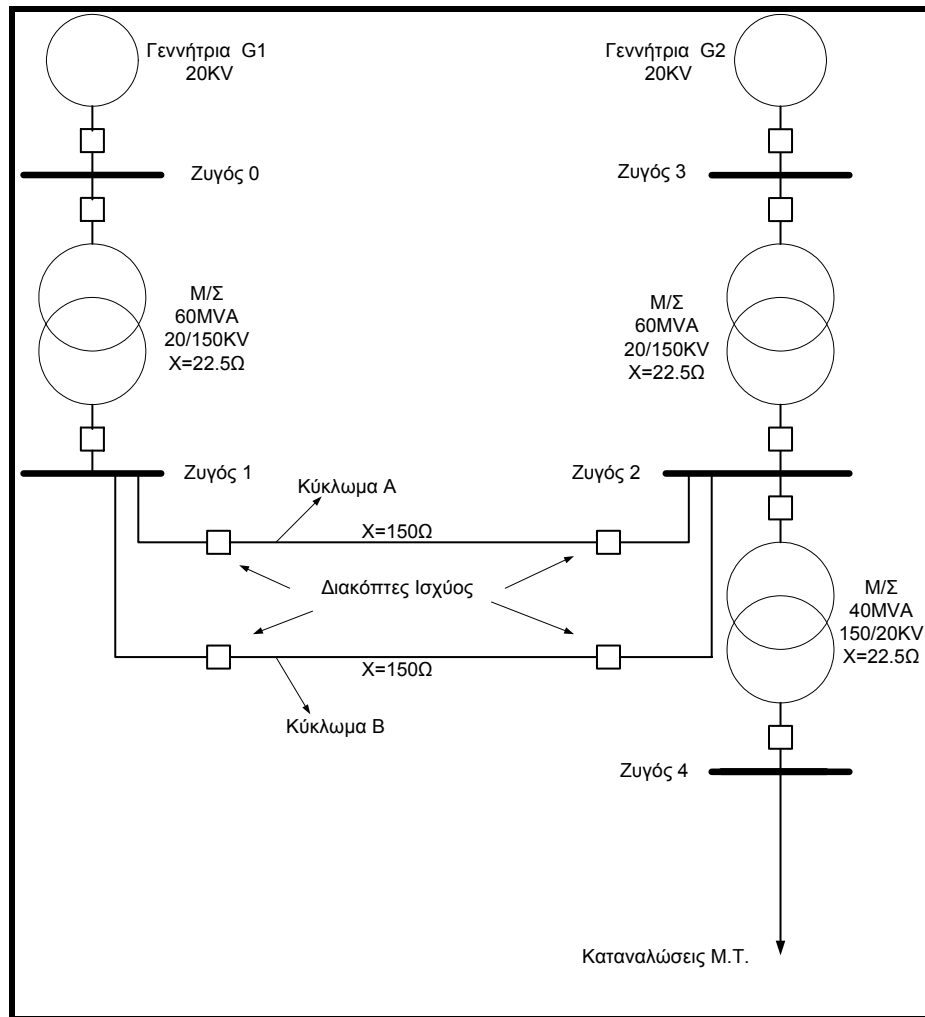
- Καθορισμός της λειτουργίας του συστήματος τοπικά και απομακρυσμένα .
- Παρακολούθηση των ηλεκτρικών μεγεθών της γραμμής σε πραγματικό χρόνο.
- Δημιουργία εικονικού προσομοιωτή ως ξεχωριστής εικόνας εργασίας.

8.2 Γραμμή μεταφοράς διπλού κυκλώματος.

Σύμφωνα με το σενάριο αυτό, έχουμε ένα σύστημα ενέργειας, όπως αυτό του σχήματος 8.1. Μολονότι τα μεγέθη που αναγράφονται στα στοιχεία του συστήματος είναι τυπικά, πρόκειται για ένα σύστημα το οποίο συναντάται συχνά και στη χώρα μας, οπότε καθίσταται ιδιαίτερα ενδιαφέρουσα η εποπτεία και αποτελεσματική προστασία του από ένα σύστημα SCADA.

Ξεκινώντας την ανάλυση του συστήματος, μόνο η γεννήτρια G1 βρίσκεται σε λειτουργία αφού η G2 είναι εκτός. Οι μετασχηματιστές, έχουν αμελητέα ωμική αντίσταση και η επαγωγική τους αντίδραση είναι ανηγμένη στα ονομαστικά τους μεγέθη. Η γραμμή μεταφοράς διπλού κυκλώματος (κυκλώματα A, B) έχει αμελητέα ωμική αντίσταση και επαγωγική αντίδραση 150Ω/ανά κύκλωμα. Σε κανονική κατάσταση λειτουργεί μόνο το ένα κύκλωμα της γραμμής μεταφοράς (κύκλωμα A). Η μέγιστη ικανότητα μεταφοράς ισχύος της γραμμής είναι 50MW/ανά κύκλωμα.

Στο ζυγό 1 υπάρχει υποσταθμός που έχει εγκατεστημένο PLC, το οποίο αποτελεί τμήμα ολοκληρωμένου συστήματος SCADA. Με τη βοήθεια του συστήματος SCADA και με τη χρήση κατάλληλων μορφοτροπέων, επιτηρούνται οι καταστάσεις των διακοπών ισχύος της γραμμής μεταφοράς (κάθε διακόπτης έχει δύο καταστάσεις, ON και OFF), καθώς επίσης και τα μέτρα και οι γωνίες των τάσεων των ζυγών 1 και 2. Οι μορφοτροπείς που έχουμε στη διάθεσή μας μετρούν τις φασικές τάσεις των ζυγών, όπως δείχνουν οι πίνακες 8.1 & 8.2.



Σχήμα 8.1. Το σύστημα μεταφοράς ενέργειας.

Μορφοτροπέας (Transducer) Τάσεως	
Είσοδος	Έξοδος
0 kV	4 mA
100 kV	20 mA

Μορφοτροπέας (Transducer) Φάσεως	
Είσοδος	Έξοδος
0 μοίρες	4 mA
45 μοίρες	20 mA

Πίνακες 8.1 & 8.2. Οι μορφοτροπίες που έχουμε στη διάθεσή μας.

Στη συνέχεια παραθέτουμε τις αριθμητικές σχέσεις που χρησιμοποιήθηκαν για τον υπολογισμό των μεγεθών, τα οποία χρησιμοποιεί το πρόγραμμα αυτοματισμού και τηλεμετράμε στο SCADA μας, σε πραγματικό χρόνο.

- Η ενεργός ισχύς, σε MW, από το ζυγό 1 προς το ζυγό 2 δίνεται από τον τύπο:

$$P_{12} = 3 \frac{V_1 V_2}{X_{12}} \sin(\delta_1 - \delta_2)$$

Επειδή η γραμμή έχει αμελητέα ωμική αντίσταση $P_{12} = -P_{21}$.

- Η άεργος ισχύς, σε MVA_r, από το ζυγό 1 προς το ζυγό 2 δίνεται από τον τύπο:

$$Q_{12} = 3 \frac{V_1^2 - V_1 V_2}{X_{12}} \cos(\delta_1 - \delta_2)$$

- Η άεργος ισχύς, σε MVA_r, από το ζυγό 2 προς το ζυγό 1 δίνεται από τον τύπο:

$$Q_{21} = 3 \frac{V_2^2 - V_1 V_2}{X_{12}} \cos(\delta_2 - \delta_1)$$

- Οι ενεργές απώλειες της γραμμής δίνονται από τον τύπο:

$$P_{losses} = P_{12} + P_{21}$$

- Οι άεργες απώλειες της γραμμής δίνονται από τον τύπο:

$$Q_{losses} = Q_{12} + Q_{21}$$

Σε κανονική κατάσταση λειτουργεί μόνο το ένα κύκλωμα της γραμμής μεταφοράς (κύκλωμα A), με αποτέλεσμα να είναι κλειστοί οι διακόπτες ισχύος D1A και D2A και ανοικτοί οι διακόπτες ισχύος D1B και D2B. Σε αυτή την κατάσταση πρέπει να ανάβει η ενδεικτική λυχνία “LINE A ON”.

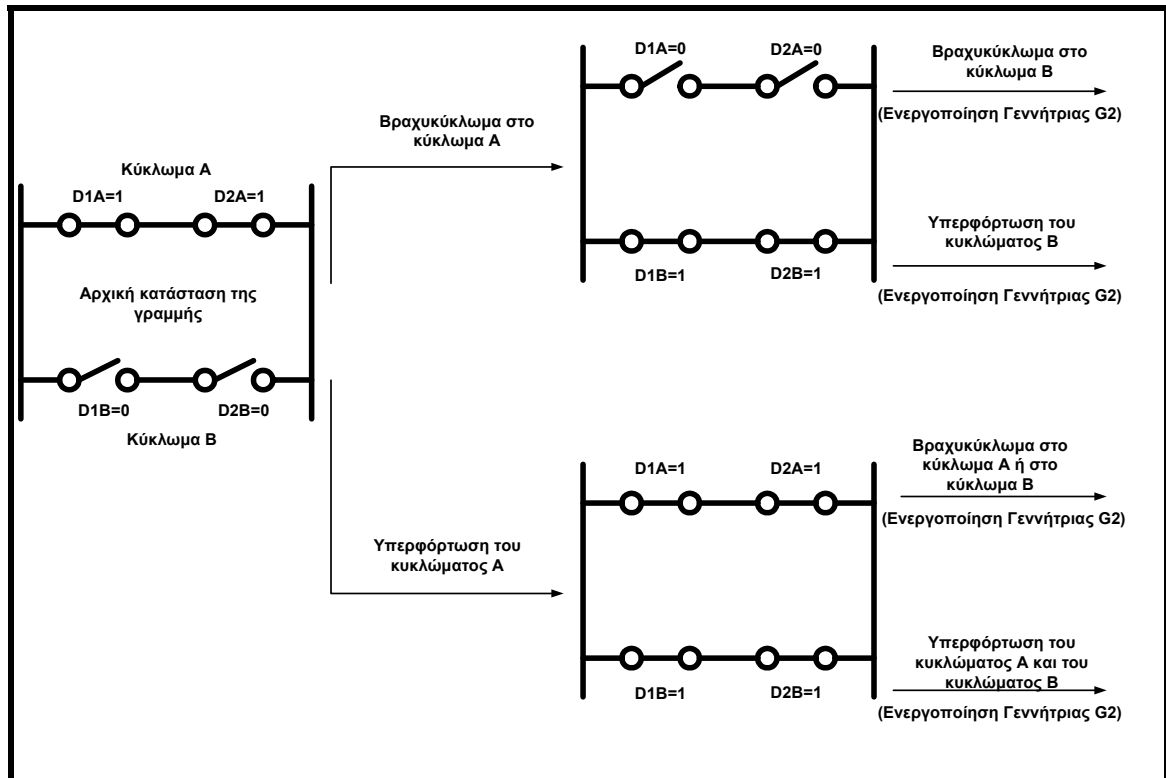
Από την παραπάνω κατάσταση λειτουργίας μπορούν να προκύψουν οι ακόλουθες δύο καταστάσεις λειτουργίας :

1. Να υπάρξει κάποιο βραχυκύκλωμα στο κύκλωμα A της γραμμής μεταφοράς ($D1A = 0$ ή $D2A = 0$). Τότε πρέπει να σβήσει η ενδεικτική λυχνία “LINE A ON” και να ανάψει η ενδεικτική λυχνία “LINE B ON”, καθώς επίσης να ενεργοποιηθεί ένα relay “RELAY B ON” που θα εντάξει σε λειτουργία το κύκλωμα B της γραμμής μεταφοράς. Επειδή όμως στην προσομοίωσή μας δεν έχουμε κάποιο relay, θα κλείσουμε (τοπικά ή απομακρυσμένα) τους διακόπτες ισχύος D1B και D2B..
2. Να υπερφορτωθεί το κύκλωμα A της γραμμής μεταφοράς ($P_{12} \geq 50MW$ ή $P_{21} \geq 50MW$). Σε αυτή την περίπτωση, προκειμένου να μην καταστραφεί το κύκλωμα A της γραμμής μεταφοράς, θα πρέπει να ενταχθεί και το κύκλωμα B της γραμμής μεταφοράς. Τώρα πλέον έχουμε δύο γραμμές παράλληλα, με αποτέλεσμα η ισοδύναμη επαγωγική αντίδραση να είναι η μισή του κάθε κυκλώματος. Τώρα πρέπει να ανάψει η ενδεικτική λυχνία “LINES A-B ON”, να σβήσει η ενδεικτική λυχνία “LINE A ON” και να ενεργοποιηθεί ένα relay “RELAY B ON”, που θα θέσει σε λειτουργία το κύκλωμα B της γραμμής μεταφοράς. Επειδή όμως στην προσομοίωσή μας δεν έχουμε κάποιο relay, θα κλείσουμε (τοπικά ή απομακρυσμένα) τους διακόπτες ισχύος D1B και D2B.

Από τις δύο παραπάνω καταστάσεις λειτουργίας μπορούν να προκύψουν οι ακόλουθες 4 καταστάσεις λειτουργίας.

1. Αν έχει υπάρξει κάποιο βραχυκύκλωμα στο κύκλωμα A της γραμμής μεταφοράς
 - Να υπάρξει κάποιο βραχυκύκλωμα στο κύκλωμα B ($D1B = 0$, ή $D2B = 0$) της γραμμής μεταφοράς. Τότε πρέπει να σβήσει η ενδεικτική λυχνία “LINE B ON”, και να ανάψουν οι ενδεικτικές λυχνίες “GENERATOR ON”, “LINES A-B OFF”.
 - Να υπερφορτωθεί το κύκλωμα B της γραμμής μεταφοράς ($P_{12} \geq 50MW$ ή $P_{21} \geq 50MW$). Τότε πρέπει να σβήσει η ενδεικτική λυχνία “LINE B ON” και να ανάψουν οι ενδεικτικές λυχνίες “GENERATOR ON”, “LINES A-B OFF”.
2. Αν έχει υπάρξει κάποια υπερφόρτιση στο κύκλωμα A της γραμμής μεταφοράς
 - Να υπάρξει κάποιο βραχυκύκλωμα στο κύκλωμα A ή στο κύκλωμα B της γραμμής μεταφοράς. Τότε πρέπει να σβήσει η ενδεικτική λυχνία “LINES A-B ON” και να ανάψουν οι ενδεικτικές λυχνίες “GENERATOR ON”, “LINES A-B OFF”.
 - Να υπάρξει κάποια υπερφόρτιση στα κυκλώματα A και B της γραμμής μεταφοράς. Τότε πρέπει να σβήσει η ενδεικτική λυχνία “LINES A-B ON” και να ανάψουν οι ενδεικτικές λυχνίες “GENERATOR ON”, “LINES A-B OFF”.

Γενικά, οι παραπάνω περιπτώσεις φαίνονται ξεκάθαρα στο σχήμα 8.2.



Σχήμα 8.2. Διαγραμματική απεικόνιση των περιπτώσεων σφάλματος του συστήματος.

8.2.1 Πρόγραμμα αυτοματισμού.

Ως γνωστόν, δημιουργήσαμε το πρόγραμμα αυτοματισμού σε γλώσσα **LADDER** του λογισμικού **STEP7** της **SIEMENS**. Στο πρόγραμμά μας χρησιμοποιήσαμε τις απαραίτητες προγραμματιστικές τεχνικές, τις οποίες περιγράψαμε στα προηγούμενα κεφάλαια.

8.2.2 Δημιουργία περιβάλλοντος εργασίας.

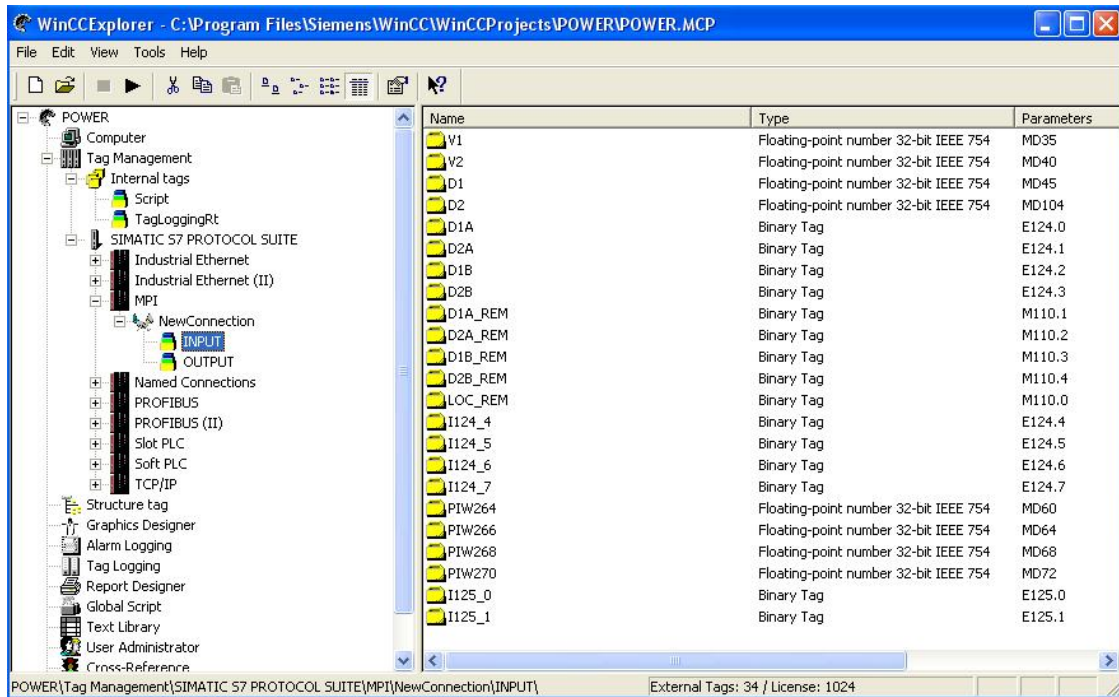
Έχοντας δημιουργήσει και φορτώσει το πρόγραμμά μας στο PLC, περάσαμε στην δημιουργία του περιβάλλοντος εργασίας στο WinCC. Δημιουργήσαμε μία νέα εφαρμογή με το όνομα “power” και εφαρμόσαμε τα παρακάτω :

Editor

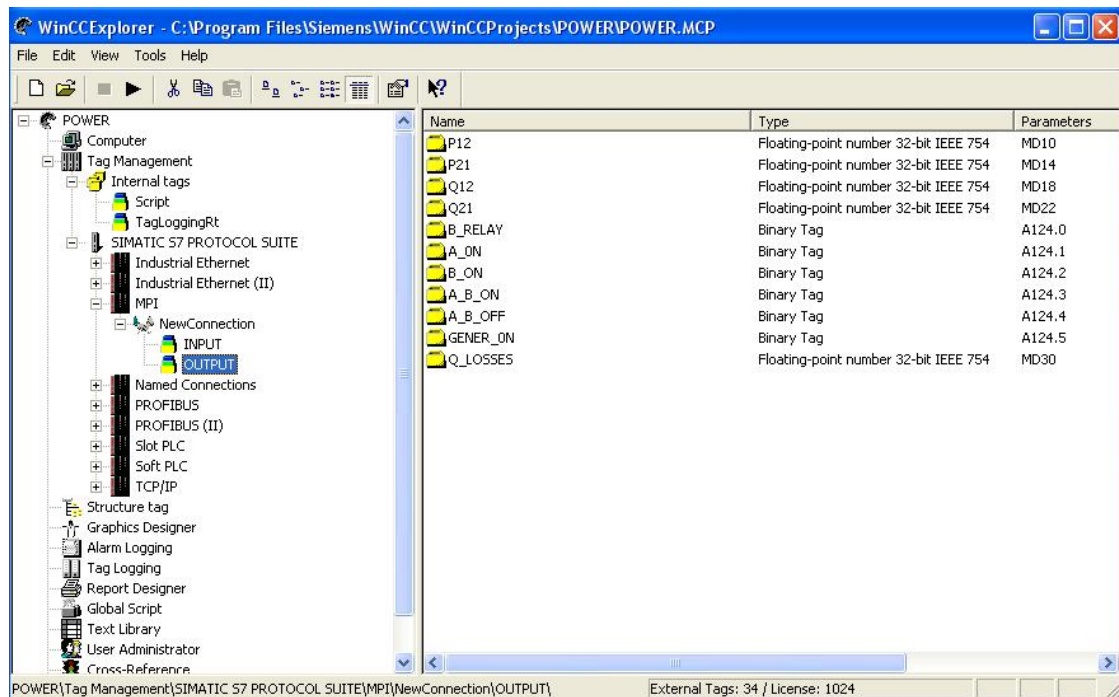
Στο υπομήμη “Computer” εφαρμόσαμε τις ρυθμίσεις που αναφέραμε στο κεφάλαιο 2, σχετικά με την ευπαρουσίαστη εικόνα της εφαρμογής μας.

Tag Management

Οι εξωτερικές μεταβλητές που δημιουργήσαμε χωρίστηκαν σε δύο ομάδες (Groups), οι οποίες ονομάστηκαν “INPUT” και “OUTPUT” (εικόνες 7.1 & 7.2).



Εικόνα 8.1. Εξωτερικές μεταβλητές του Group “INPUT”.

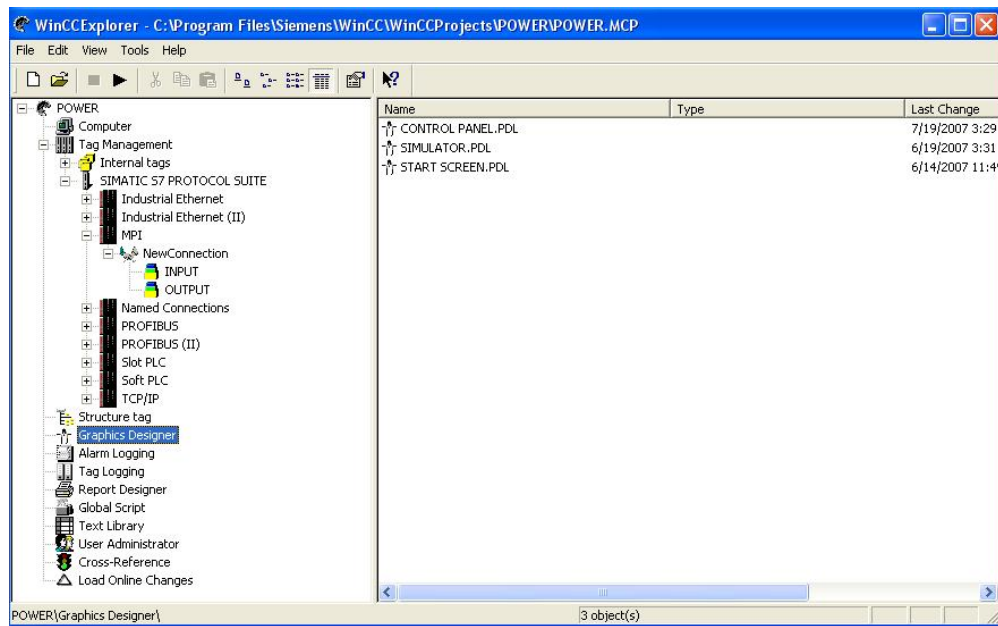


Εικόνα 8.2. Εξωτερικές μεταβλητές του Group “OUTPUT”.

Οι εσωτερικές μεταβλητές, όπως αναφέρθηκε στο κεφάλαιο 4, αντιγράφηκαν μαζί με το αρχείο του προσομοιωτή ώστε να τον χρησιμοποιήσουμε σαν εικόνα εργασίας.

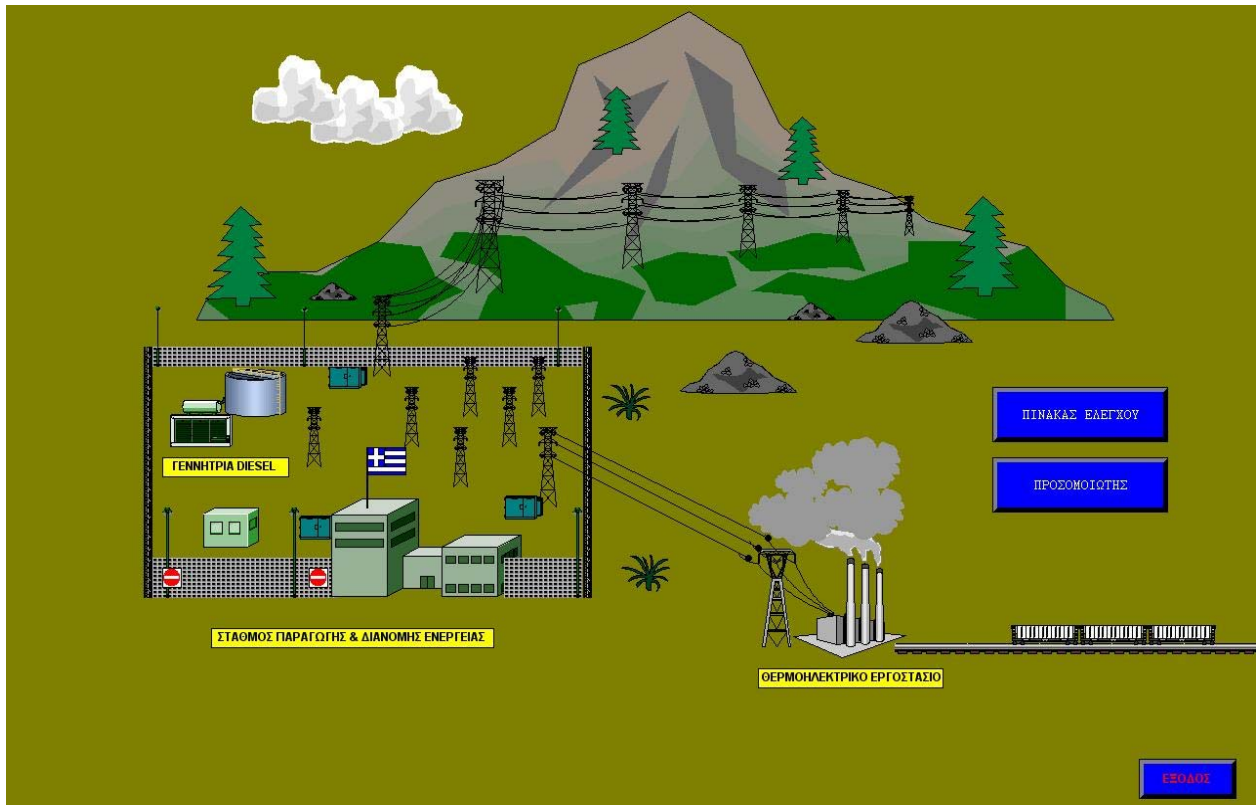
Graphics Designer

Με τη βοήθεια του σχεδιαστή γραφικών δημιουργήσαμε ένα αρχείο και αντιγράψαμε εδώ το ήδη έτοιμο αρχείο του προσομοιωτή (εικόνα 7.3). Στη συνέχεια, παρουσιάζουμε την προεπισκόπηση της κάθε εικόνας εργασίας και περιγράφουμε τη δημιουργία της, όπου κρίνεται αναγκαίο.

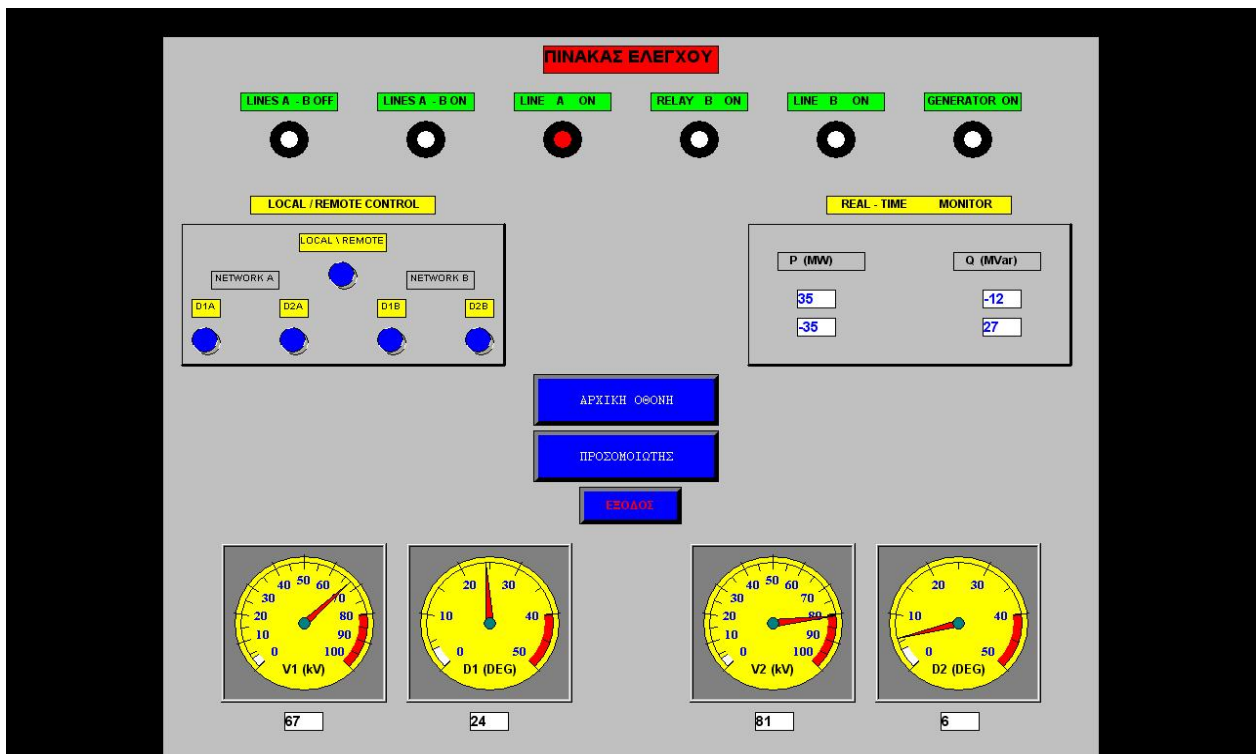


Εικόνα 8.3. Τα αρχεία του Graphics Designer.

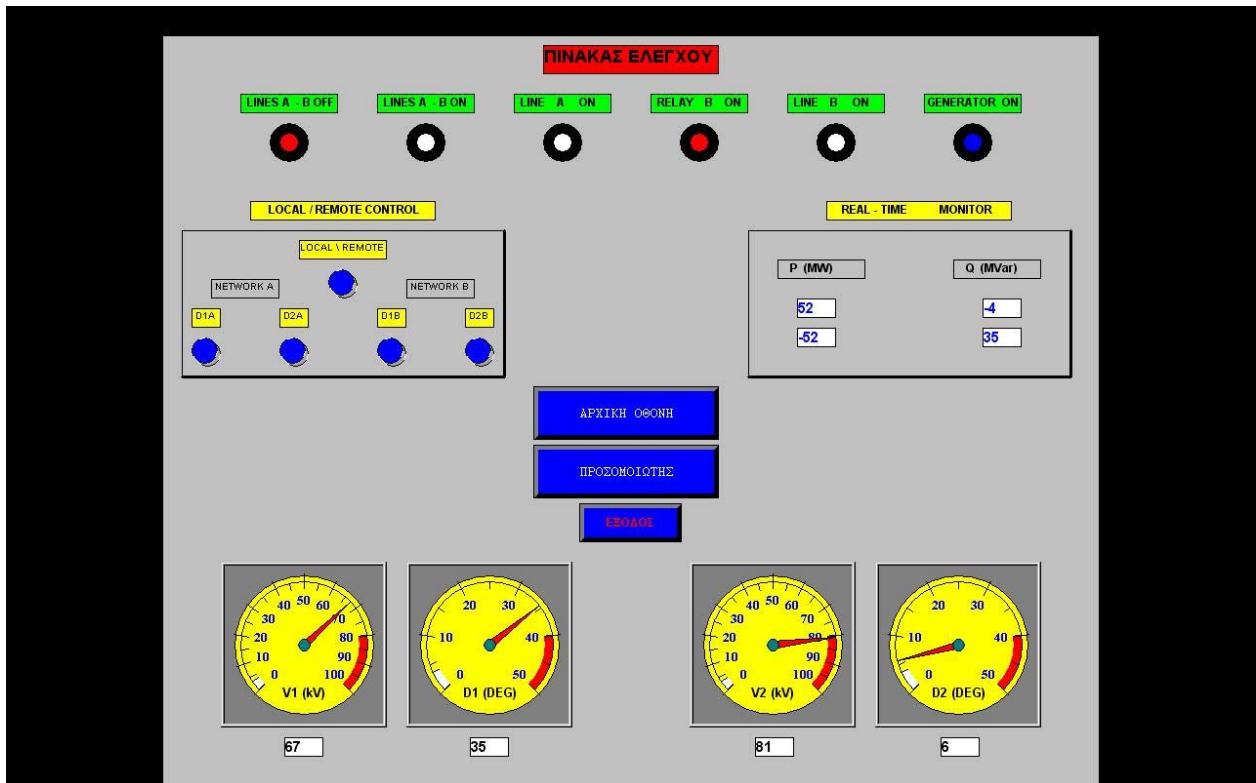
➤ Start Screen.pdl



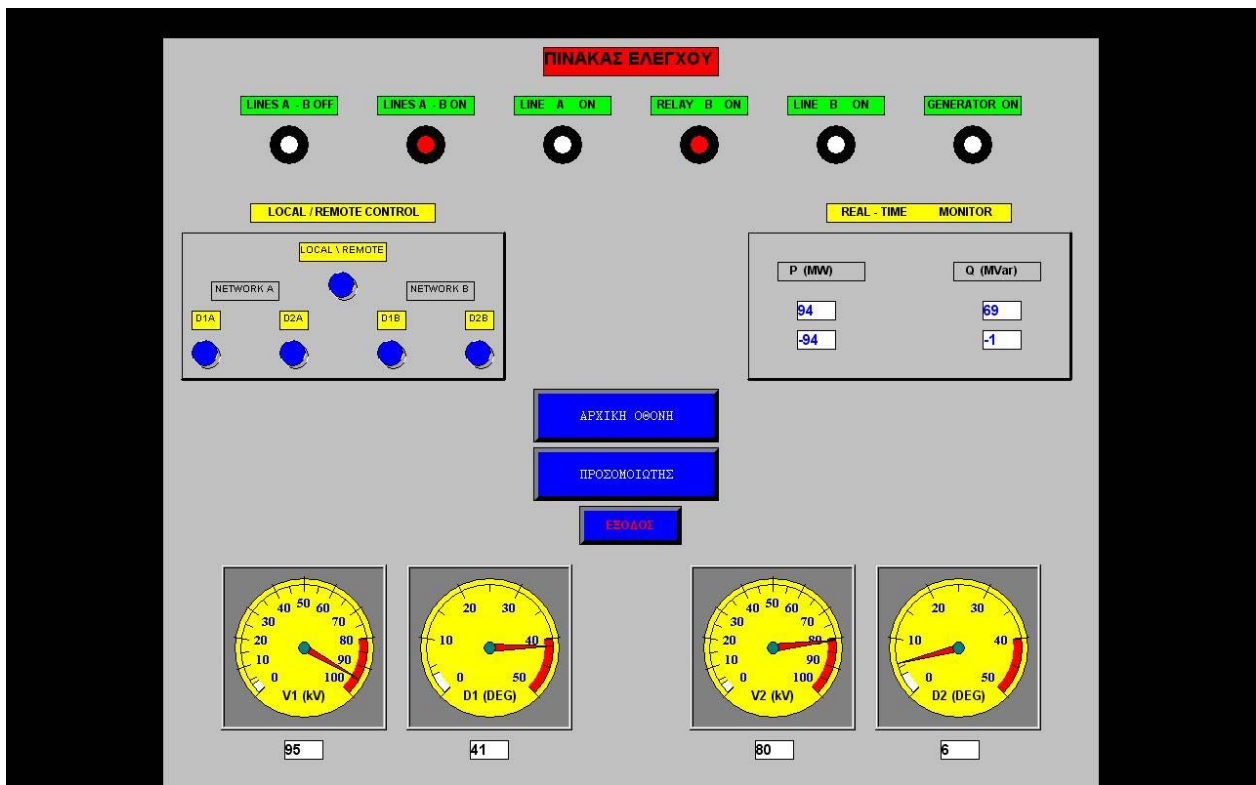
Εικόνα 8.4. Η πρώτη εικόνα της εφαρμογής.



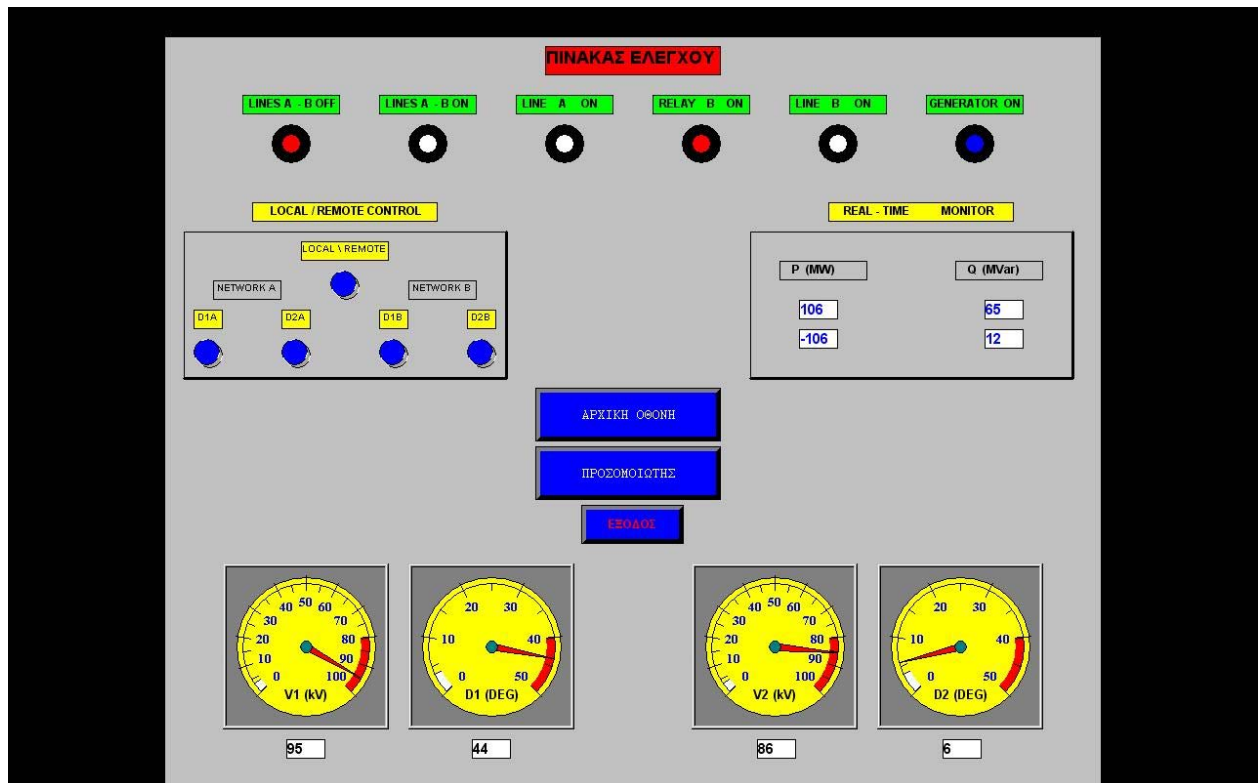
Εικόνα 8.5. Ο πίνακας ελέγχου σε κατάσταση κανονικής λειτουργίας της γραμμής A.



Εικόνα 8.6. Ο πίνακας ελέγχου σε κατάσταση επείγουσας ένταξης και της γραμμής B ή της γεννήτριας.



Εικόνα 8.7. Ο πίνακας ελέγχου σε κατάσταση λειτουργίας και των δύο γραμμών.

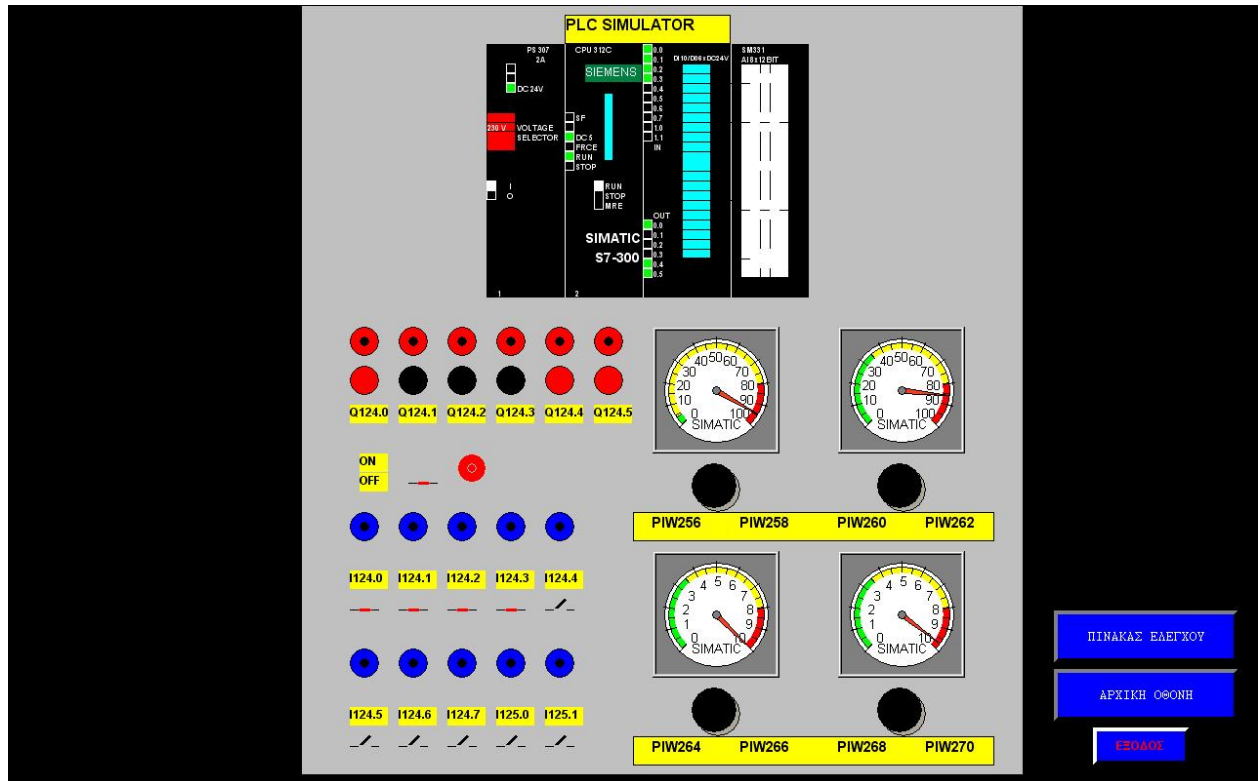


Εικόνα 8.8. Ο πίνακας ελέγχου σε κατάσταση επείγουσας ένταξης της γεννήτριας.

Γενικά, τα αντικείμενα τα οποία χρησιμοποιήθηκαν έχουν ήδη αναλυθεί στα προηγούμενα κεφάλαια, όπως και ο τρόπος δυναμικής διασύνδεσής τους. Προστέθηκαν τα απαραίτητα μπουτόν περιήγησης και καταλήξαμε στο παραπάνω αποτέλεσμα.

➤ Simulator.pdl

Η εικόνα του προσομοιωτή δημιουργήθηκε ακολουθώντας την ίδια διαδικασία την οποία εφαρμόσαμε και στα προηγούμενα σενάρια εφαρμογών. Διασυνδέσαμε κατάλληλα τις μεταβλητές μας, προσθέσαμε τα κατάλληλα μπουτόν περιήγησης και καταλήξαμε στο αναμενόμενο αποτέλεσμα της εικόνας 8.9.



Εικόνα 8.9. Ο προσομοιωτής.

ΚΕΦΑΛΑΙΟ 9

Συμπεράσματα – Προοπτικές

9.1 Συμπεράσματα – Προοπτικές.

Η ενασχόληση με την παρούσα εργασία μου έδωσε την δυνατότητα να εντρυφήσω σε ένα τομέα που πάντα με ενδιέφερε και να εμπλουτίσω τις γνώσεις μου σχετικά με τα σύγχρονα συστήματα SCADA. Παρότι οι παρούσες εφαρμογές αναπτύχθηκαν στο επίπεδο του εργαστηρίου, εντούτοις δε διαφέρουν σημαντικά από συστήματα που λειτουργούν υπό πραγματικές συνθήκες, από την άποψη ότι το μόνο, στην ουσία, που αλλάζει είναι η ταχύτητα διασύνδεσης και η λεπτομερής εξέταση όλων των μεταβλητών σε πραγματικό χρόνο. Από πλευράς προγραμματισμού, το πακέτο WinCC αποτελεί ένα εργαλείο πολύπλοκο, το οποίο απαιτεί συνεχή και επισταμένη ενασχόληση προκειμένου να εκμεταλλευτεί κανείς τα πλεονεκτήματά του. Τα δε πλεονεκτήματα, βέβαια, αφορούν τη δημιουργία κατάλληλων περιβάλλοντων εργασίας, με υψηλής ποιότητας λεπτομέρεια και δυνατότητα επέμβασης σε όλες σχεδόν τις ιδιότητες των αντικειμένων, ώστε να καλυφθούν οι ανάγκες και των πλέον απαιτητικών εφαρμογών.

Επιπροσθέτως, οι παρούσες εφαρμογές συνιστούν και ένα χρήσιμο εκπαιδευτικό οδηγό προκειμένου οι σπουδαστές να εφαρμόσουν τις θεωρητικές τους γνώσεις και να πάρουν μία γεύση από τα πλεονεκτήματα των συστημάτων αυτών, όταν εφαρμόζονται σε παραγωγικές διαδικασίες. Μάλιστα, έχουν έτοιμα παραδείγματα να τροποήσουν και να δουν άμεσα τα αποτελέσματα των επεμβάσεών τους σε αυτά, κάτι που συνδράμει περισσότερο στην εμπέδωση και αξιοποίηση των γνώσεών τους. Συμπερασματικά, το παρόν λογισμικό αποτελεί τον κινητήριο μοχλό ενός SCADA και οι εμπειρίες που αποκόμισα κρίνονται πολύτιμες για την μετέπειτα πορεία μου ως μηχανικός και για το λόγο αυτό θα ήθελα πάλι να ευχαριστήσω τον κύριο Γεώργιο Κορρέ για την καθοδήγησή του πάνω σε ένα τόσο ενδιαφέρον θέμα.

Βιβλιογραφία

- [1] **«Στοιχεία Συστημάτων SCADA»**, Γεώργιος Κορρές, Αναπληρωτής Καθηγητής Ε.Μ.Π., Σημειώσεις Ε.Μ.Π., Αθήνα 2001.
- [2] **«Εφαρμογές Αυτοματισμών με PLC's»**, Γεώργιος Τζουνίδης, Εκδόσεις Παπασωτηρίου, Θεσ/νίκη 2001.
- [3] **«Ολοκληρωμένο Σύστημα Εποπτείας και Ελέγχου Ηλεκτρικών Μηχανών»**, Διπλωματική εργασία του Γεώργιου Αρβανίτη, Αθήνα 2005
- [4] **«Εργαστηριακό Σύστημα Προσομοίωσης Προγραμματιζόμενου Λογικού Ελεγκτή»**, Διπλωματική εργασία του Θωμά Σιαμπαλιώτη, Αθήνα 2006
- [5] Ιστοσελίδα SIEMENS : www.siemens.com