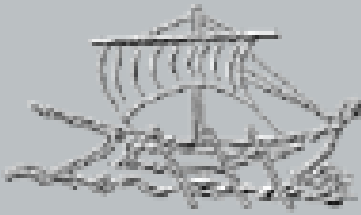


Τρίτη, 6 Σεπτεμβρίου 2016



Α.Ε.Ι. ΠΕΙΡΑΙΑ Τ.Τ.

**ΑΝΩΤΑΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΕΙΡΑΙΑ
ΤΕΧΝΟΛΟΓΙΚΟΥ ΤΟΜΕΑ**

Πτυχιακή Εργασία

*Βιομηχανικά Συστήματα
Ελέγχου SCADA &
Προγραμματισμός με το
SIMATIC WinCC 7.2*

Σπουδαστές:

Στασινός Συμεών 34652

Αλεξίου Παναγιώτης 34322

Υπεύθυνη

Καθηγήτρια:

Βελώνη Αναστασία

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

***Βιομηχανικά Συστήματα Ελέγχου SCADA &
Προγραμματισμός με το SIMATIC WinCC 7.2***

Στασινός Συμεών 34652

Αλεξίου Παναγιώτης 34322

Εισηγητής: Βελώνη Αναστασία

Εξεταστική Επιτροπή:

Ημερομηνία Εξέτασης:

Ευχαριστήριες

Η παρούσα εργασία αποτελεί την πτυχιακή μας εργασία για την απόκτηση του πτυχίου μας ως μηχανικοί ηλεκτρονικών συστημάτων. Υπήρχαν αρκετές δυσκολίες λόγω των επαγγελματικών μας υποχρεώσεων αλλά και στην ανεύρεση υλικού σχετικά με τον προγραμματισμό των συστημάτων SCADA με το λογισμικό SIMATIC WinCC της εταιρίας SIEMENS όπου εστιάζεται η εργασία μας. Θα ήταν παράληψη να μην ευχαριστήσουμε την καθηγήτρια **κ. Αναστασία Βελώνη** για την ανάθεση της διπλωματικής εργασίας αλλά και για την πολύτιμη βοήθεια που μας παρείχε καθ' όλη τη διάρκεια της εκπόνησης της.

Περίληψη

Η εργασία συνίσταται στην αναφορά διαφόρων συστημάτων βιομηχανικού ελέγχου και συγκεκριμένα στα συστήματα SCADA καθώς και στον προγραμματισμό τους μέσω του λογισμικού SIMATIC WinCC της εταιρίας Siemens, για την παρακολούθηση σε πραγματικό χρόνο ορισμένων βασικών φυσικών μεγεθών του βιομηχανικού συστήματος. Η εργασία χωρίζεται σε δύο βασικά μέρη. Στο πρώτο μέρος γίνεται είναι μία εκτενής παρουσίαση των συστημάτων βιομηχανικού ελέγχου SCADA καθώς επίσης η ανάπτυξη των διαφόρων αυτοματισμών ελέγχου κατά την πάροδο των ετών. Το δεύτερο μέρος αφορά την υλοποίηση σεναρίων βασικών εφαρμογών βιομηχανικού αυτοματισμού SCADA μέσω του λογισμικού SIMATIC WinCC με σκοπό την δημιουργία εποπτικού σταθμού παρακολούθησης των τιμών των μεγεθών που μας ενδιαφέρουν σε ένα σύστημα βιομηχανικού ελέγχου SCADA. Εκμεταλλευόμενοι τις δυνατότητες του λογισμικού παρακολούθησης και ελέγχου SIMATIC WinCC της εταιρίας SIEMENS που χρησιμοποιήσαμε για την υλοποίηση του ελέγχου της βιομηχανικής μονάδος που εξετάσαμε, κατορθώσαμε και εστιάσαμε τις προσπάθειές μας στο προγραμματιστικό μέρος των εφαρμογών, αποφεύγοντας ηλεκτρομηχανολογικές κατασκευές. Με τον τρόπο αυτό, ασχοληθήκαμε με την διασύνδεση και ανάπτυξη κατάλληλων περιβάλλοντων παρακολούθησης και ελέγχου. Δημιουργήσαμε, έτσι, ολοκληρωμένα συστήματα SCADA, τα οποία αποτέλεσαν μία πολύτιμη γνωστική εμπειρία. Στα παρακάτω κεφάλαια θα γίνει μία ιστορική αναφορά στην δομή των βιομηχανικών μονάδων παλαιότερα, στην εξέλιξή τους στην πάροδο των ετών καθώς επίσης και στις ανάγκες που κατέστησαν απαραίτητο τον πλήρη έλεγχο αυτοτελών των μονάδων μίας βιομηχανίας. Φτάνοντας στο σήμερα και συγκεκριμένα στα συστήματα ελέγχου SCADA, πραγματοποιείται μία εκτενέστερη ανάλυση των δομικών στοιχείων που αποτελούν το σύστημα παρακολούθησης όπως επίσης και του τρόπου λειτουργίας τους. Συνεχίζοντας στα επόμενα κεφάλαια, προχωράμε σε μία παρουσίαση του λογισμικού SIMATIC WinCC, από την φάση της εγκατάστασης στο υπολογιστικό μας σύστημα έως και την φάση της παραμετροποίησής του για την βιομηχανική μονάδα που επιθυμούμε να παρακολουθήσουμε. Επίσης, θα δημιουργήσουμε μία δοκιμαστική εφαρμογή για την περαιτέρω εμπέδωση των στοιχείων του λογισμικού υλοποιώντας παράλληλα έναν εικονικό προσομοιωτή, ακριβώς όμοιο με τον πραγματικό για την πλήρη κατανόηση της χρήσης αλλά και της σημασίας του SIMATIC WinCC. Τέλος, παραθέτουμε τα συμπεράσματά μας από την ενασχόληση με την παρούσα εργασία και στο παράρτημα, επίσης, παραθέτουμε τα προγράμματα αυτοματισμού που χρησιμοποιήσαμε.

ABSTRACT

This pre-graduation assignment consists of mentioning of various automated industrial control systems and specifically in SCADA systems as well as in their programming through the Siemens SIMATIC WinCC software, for the purpose of real time monitoring of variables in the industrial system. The pre-graduation assignment is composed of two main parts. In the first part an extended automated industrial control systems SCADA takes place as well as the development of various automated control systems over the years. The second part consists of the implementation of basic application scenarios that are being carried out with the Siemens SIMATIC WinCC software, aiming to create an observatory station for the variables that are of interest. By exploiting the advantages of the SIMATIC WinCC monitoring and control software that we used, we achieved on focusing our efforts in programming the applications, thus avoiding electro – mechanic constructions. By doing this, we got involved with the connection and development of the appropriate monitoring and control environments. In this way, we created completed SCADA systems, which were a valuable learning experience. In the following chapters, there will be a historic reference in the previous structure of industrial units, in their development through the ages as well as the needs which have become necessary to the complete control of independent industrial units. Nowadays, in SCADA systems, there is an extended analysis of structural elements that make up the monitoring system and its functioning. Continuing to the next chapters, we move on to a SIMATIC WinCC software presentation, from its installation up to the configuration of the plant we want to monitor. Furthermore, we will create beta applications aiming to better understand the details of the software while implementing a virtual simulator, exactly the same with the real one for the complete understanding of the SIMATIC WinCC software use and meaning. Finally, we present our conclusions and the beta applications that we used.

Περιεχόμενα

Κεφάλαιο 1 – Η εξέλιξη των βιομηχανιών

1.1 Από την προβιομηχανική στη βιομηχανική εποχή.....	9
1.2 Δομή των πρώτων βιομηχανιών	10
1.3 Παραγωγική διαδικασία και έλεγχος	12
1.4 Συστήματα Ελέγχου Βιομηχανιών	13
1.4.1 Τύποι συστημάτων	13
1.4.2 Ελεγκτές	14
1.5 Τύποι βιομηχανικού ελέγχου	15
1.6 Συστήματα αυτομάτου βιομηχανικού ελέγχου (ICS)	16
1.6.1 Λόγοι αυτοματοποίησης.....	16
1.6.2 Κατηγορίες Συστημάτων Αυτομάτου Ελέγχου	17
1.6.2.1 Προγραμματιζόμενοι Λογικοί Ελεγκτές (PLC's).....	17
1.6.2.2 Κατανεμημένα Συστήματα Ελέγχου (DCS).....	18
1.6.2.3 Ενσωματωμένος Έλεγχος (Embedded Control).....	19
1.6.2.4 Συστήματα Αυτόματου Εποπτικού Ελέγχου (SCADA)	19

Κεφάλαιο 2 – Συστήματα Βιομηχανικού Ελέγχου SCADA

2.1 Τα συστήματα SCADA.....	21
2.2 Ιστορική Αναδρομή	22
2.3 Αρχιτεκτονική συστήματος SCADA.....	24
2.3.1 Κεντρική Τερματική Μονάδα (MTU).....	26
2.3.2 Αισθητήρες και ευφυείς συσκευές (IED's).....	26
2.3.3 Απομακρυσμένες Τερματικές Μονάδες (RTU)	27
2.3.4 Προγραμματιζόμενοι Λογικοί Ελεγκτές (PLC).....	27
2.3.5 Διεπαφή Ανθρώπου – Μηχανής (HMI).....	28
2.3.6 Τηλεμετρία.....	29
2.3.7 Λογισμικό Προγραμματισμού SCADA.....	31
2.3.7.1 Χαρακτηριστικά λογισμικού SCADA.....	32
2.4 Πρωτόκολλα επικοινωνίας.....	33
2.4.1 Αρχή λειτουργίας πρωτοκόλλων επικοινωνίας	33
2.4.1.1 Το πρωτόκολλο Modbus	34
2.4.1.2 Το πρωτόκολλο Profibus	36
2.4.1.3 Το πρωτόκολλο IEC 60870.....	37
2.4.1.4 Το πρωτόκολλο DNP3.....	39
2.4.1.5 Το πρωτόκολλο RP – 570.....	40
2.4.1.6 Το πρωτόκολλο Hart	41
2.4.1.7 Το πρωτόκολλο Conitel	41

2.5 Σύγχρονα καταναμημένα συστήματα SCADA.....	42
2.6 Σκοπός των συστημάτων SCADA	44
2.7 Στόχος των συστημάτων SCADA.....	44
2.8 Κύριες λειτουργίες ενός συστήματος SCADA.....	45
2.9 Εξέλιξη των συστημάτων SCADA	47
2.9.1 1η Γενιά – Τα μονολιθικά συστήματα	48
2.9.2 2η Γενιά – Τα καταναμημένα συστήματα	50
2.9.3 3η Γενιά – Τα δικτυωμένα συστήματα	52
2.9.4 4η Γενιά – Internet Of Things.....	53
2.10 Οφέλη και πλεονεκτήματα των συστημάτων SCADA	55
2.11 Προβλήματα και μειονεκτήματα των συστημάτων SCADA	58
2.12 Ασφάλεια στα συστήματα SCADA.....	60
2.12.1 Επιθέσεις στα συστήματα SCADA.....	61
2.12.1.1 Είδη επιθέσεων	62
2.12.2 Αντιμετώπιση επιθέσεων	62
2.12.2.1 Δακτυλίδι αμυνών - Ring of Defenses	63
2.13 Εμπορικές εφαρμογές συστημάτων SCADA.....	66
2.14 Το μέλλον των συστημάτων SCADA	68

Κεφάλαιο 3 – Εγκατάσταση του λογισμικού Siemens SIMATIC WinCC 7.2

3.1 Εγκατάσταση του λογισμικού SIMATIC WinCC	69
3.2 Ενεργοποίηση Λογισμικού WinCC μέσω προγράμματος crack	81

Κεφάλαιο 4 – Εισαγωγή στο λογισμικού SIMATIC WinCC 7.2

4.1 Η δημιουργία του λογισμικού SIMATIC WinCC.....	86
4.2 Εκκίνηση του λογισμικού WinCC 7.2.....	87
4.3 Το περιβάλλον εργασίας του SIMATIC WinCC 7.2	87
4.3.1 Τμήμα Πλοήγησης (Navigation Window)	89
4.3.2 Τμήμα Δεδομένων (Data Window)	89
4.3.3 Μπάρα μενού (Menu Bar)	89
4.3.4 Μπάρα εργαλείων (Toolbar)	90
4.3.5 Μπάρα κατάστασης (Status Bar)	90
4.4 Βοηθητικά προγράμματα σχεδίασης εφαρμογών στο WinCC.....	91
4.4.1 Computer (Υπολογιστής)	91
4.4.2 Tag Managements (Διαχείριση Μεταβλητών).....	93
4.4.3 Data Types (Τύποι Μεταβλητών).....	93
4.4.4 Editors	94
4.4.4.1 Graphics Designer	94
4.4.4.2 Tag Logging.....	94

4.4.4.3 Alarm Logging	94
4.4.4.4 Report Designer.....	94
4.4.4.5 Global Script	95

Κεφάλαιο 5 – Δημιουργία Εφαρμογής με το SIMATIC WinCC7.2

5.1 Μεθοδολογία Δημιουργίας Εφαρμογής	97
5.2 Αναλυτική παρουσίαση βημάτων δημιουργίας project	97
5.2.1 Εκκίνηση του WinCC	97
5.2.1.1 Single – User Project.....	98
5.2.1.2 Multi – User Project	98
5.2.1.3 Client Project	99
5.2.2 Δημιουργία εφαρμογής (project)	99
5.2.3 Ρύθμιση υπολογιστή εφαρμογής	100
5.2.4 Επιλογή και εγκατάσταση driver	104
5.2.5 Ορισμός των μεταβλητών (tags).....	108
5.2.5.1 Κατηγορίες Μεταβλητών (tags)	108
5.2.5.1.1 Δημιουργία Εξωτερικών Μεταβλητών (Process Tags)	109
5.2.5.1.2 Δημιουργία Εσωτερικών Μεταβλητών (InternalTags)	111
5.2.6 Δημιουργία και εμφάνιση των εικόνων της εφαρμογής.....	112
5.2.6.1 Το πρόγραμμα Graphics Designer	112
5.2.6.2 Δημιουργία εικόνων εφαρμογής	113
5.2.6.2.1 Menu Bar	116
5.2.6.2.2 Standard Toolbar	116
5.2.6.2.3 Zoom Palette	116
5.2.6.2.4 Font Palette	116
5.2.6.2.5 Object Palette	116
5.2.6.2.6 Alignment Palette	117
5.2.6.2.7 Color Palette	117
5.2.6.2.8 Workspace	117
5.2.6.2.9 Style Palette	117
5.2.6.2.10 Object Properties.....	117
5.2.6.2.11 Layer Bar	118
5.2.6.2.12 Status Bar.....	118
5.2.6.3 Δυναμική μεταβολή αντικειμένων.....	119
5.2.7 Καθορισμός των ιδιοτήτων του WinCC Runtime	126

Κεφάλαιο 6 – Προσομοίωση Εφαρμογών στο SIMATIC WinCC 7.2

6.1 Χειρισμός και λειτουργία φωτεινών σηματοδοτών	128
6.2 Εποπτεία και έλεγχος στάθμης δεξαμενής σε μία βιομηχανική μονάδα	140

Κεφάλαιο 1

Η Εξέλιξη των Βιομηχανιών



1.1 Από την προβιομηχανική στη βιομηχανική εποχή

Ετυμολογικά, η λέξη **βιομηχανία** μπορεί να αναλυθεί ως η κατασκευή ενός αγαθού ή μίας υπηρεσίας μέσα σε μία οικονομία. Παρόλο που η βιομηχανία αποτελεί μία ευρεία έννοια για κάθε είδος οικονομικής παραγωγής, στα οικονομικά και τον αστικό σχεδιασμό η βιομηχανία είναι συνώνυμη της δευτερογενούς παραγωγής, η οποία είναι το είδος της οικονομικής δραστηριότητας που ασχολείται με την κατασκευή αγαθών και προϊόντων. Η εμφάνιση των πρώτων βιομηχανιών στην Ευρώπη αλλά και στις Η.Π.Α παρατηρείται από τις αρχές του 19^{ου} αιώνα. Κατά βάση τότε, η οικονομία στηριζόταν κυρίως σε αγροτικές πηγές οι οποίες παράγονταν κατά κύριο λόγο σε μικρά εργαστήρια ή ακόμα και στα σπίτια των ιδίων των τεχνιτών. Παράλληλα, τα περισσότερα παραγωγικά σύνολα ήταν μικρά σε μέγεθος αλλά και σε δυναμικότητα παραγωγής. Χρησιμοποιούνταν ως επί το πλείστον σαν τόποι ανταλλαγής των καρπών της γης και των προϊόντων της βιοτεχνίας, ενώ οι ελάχιστες μονάδες με μεγαλύτερη παραγωγική δύναμη, οι οποίες έκτοτε χαρακτηρίστηκαν ως οι πρώτες βιομηχανίες, ήταν αρκετά διάσπαρτες. Οι περισσότερες βρίσκονταν στην Αγγλία και λιγότερες στην ηπειρωτική Ευρώπη. Αυτές οι βιομηχανικές μονάδες συνιστούσαν την αφετηρία ενός νέου παραγωγικού συστήματος, του εργοστασιακού συστήματος.



Βιομηχανία πριν την Βιομηχανική Επανάσταση

Το νέο εργοστασιακό σύστημα, με την έννοια της παρασκευής-κατασκευής, έγινε ο κύριος τομέας παραγωγής και εργασίας στις Ευρωπαϊκές και Βορειοαμερικανικές χώρες κυρίως κατά τη διάρκεια της **Βιομηχανικής Επανάστασης**. Η περίοδος αυτή αναφέρονταν σε μία διαδικασία με βαθύ και διαρκή μετασχηματισμό των οικονομικών και κοινωνικών συνθηκών,

καθώς και των συνθηκών εργασίας, που ξεκίνησε στο δεύτερο μισό του 18ου Αιώνα και ενισχύθηκε στο 19^ο αιώνα, αρχικά στην Βρετανία και, εν συνεχεία, στο σύνολο της Δυτικής Ευρώπης.

Τα κύρια χαρακτηριστικά της **Βιομηχανικής Επανάστασης** ήταν:

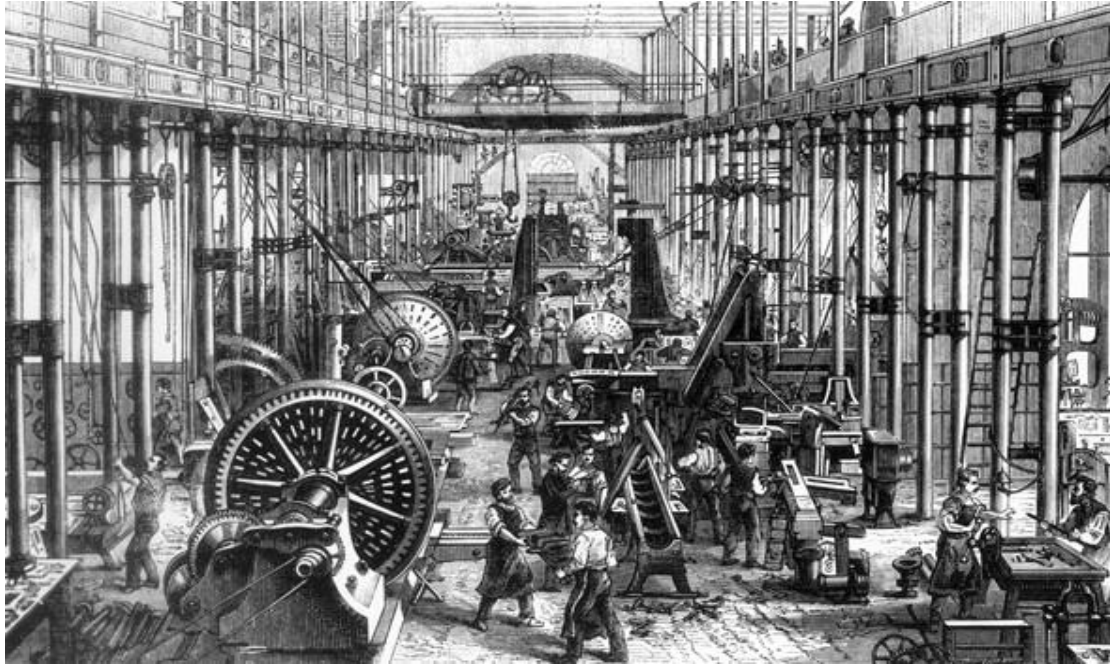
1. Η εκτεταμένη χρήση νέων τεχνικών μέσων που περιόριζαν τη χειρωνακτική εργασία, αυξάνοντας την παραγωγή και μειώνοντας ταυτόχρονα το κόστος των προϊόντων,
2. Η αξιοποίηση νέων μορφών ενέργειας,
3. Η εφαρμογή καινοτομιών κυρίως στη μεταλλουργία,
4. Η ανάδειξη της βιομηχανικής μονάδος ως του βασικού τόπου παραγωγής, όπου συγκεντρώθηκε η πλειοψηφία των εργατών και τέλος
5. Οι υψηλοί ρυθμοί ανάπτυξης.

Ως φυσικό επακόλουθο, οι βιομηχανικές χώρες άρχισαν να υιοθετούν καπιταλιστική οικονομική πολιτική. Από την Βιομηχανική επανάσταση έως και σήμερα, πάνω από το μισό της παγκόσμιας οικονομίας προέρχεται από τη βιομηχανική παραγωγή, ποσοστό μεγαλύτερο από το αντίστοιχο ποσοστό της αγροτικής παραγωγής.

1.2 Δομή των πρώτων βιομηχανιών

Τα αποτελέσματα της Βιομηχανικής επανάστασης ήταν ήδη ορατά από την πρώτη στιγμή. Πολλές ευρωπαϊκές χώρες αναγκάστηκαν να στραφούν προς μία συνεχόμενη αναπτυξιακή κατεύθυνση, απειλώντας με ολοσχερή καταστροφή τις παλαιές βιοτεχνίες. **Η εμφάνιση νέων τεχνικών παραγωγής** καθώς επίσης **η ενσωμάτωση μέσα σε αυτές νέων μηχανημάτων** όπου ευνοούσαν την μαζική παραγωγή, όπως οι μηχανές εσωτερικής καύσης, έδωσαν την δυνατότητα ανάπτυξης των μικρών βιοτεχνιών σε εργοστάσια και βιομηχανικές μονάδες. Παράλληλα, **οι υψηλοί ρυθμοί ανάπτυξης και παραγωγής** αγαθών όπως επίσης **η ραγδαία ανάγκη συγκέντρωσης περισσότερων εργατών** στα εργοστάσια συντέλεσε θετικά στην αναβάθμιση της απλής βιοτεχνίας σε μεγάλη παραγωγική μονάδα. Δεν είναι τυχαίο ότι μετά το πέρας της Βιομηχανικής επανάστασης, αρκετές βιομηχανικές πόλεις γεννήθηκαν σε κορυφές λόφων αποτελούμενες από 30 με 40 μικρότερα εργοστάσια.

Γύρω από τα εργοστάσια βρίσκονταν μικρά, φτωχικά εργατικά σπιτάκια, στα οποία οι άνθρωποι ζούσαν μέσα στην αθλιότητα και το θάνατο. Έτσι, η παραγωγή ενός αγαθού αρχίζει να γίνεται μαζική, αυτοματοποιημένη, δημιουργώντας υλικά ανώτερης ποιότητας.



Εσωτερικό βιομηχανίας μετά την Βιομηχανική Επανάσταση

Παρατηρώντας στο εσωτερικό μίας πρώιμης βιομηχανίας, μπορούμε εύκολα να αντιληφθούμε κατ' αρχάς την αντικατάσταση αρκετών εργατικών χεριών με ογκώδη μηχανήματα. Χάρη την έκρηξη της τεχνολογίας που έλαβε χώρα κατά την διάρκεια της Βιομηχανικής επανάστασης, βλέπουμε την μαζική εισαγωγή της σύγχρονης για τότε τεχνολογίας στην παραγωγική διαδικασία. Οι τυποποιημένες, μονότονες, επαναλαμβανόμενες, κουραστικές και συχνά επικίνδυνες σωματικές εργασίες εκτελούνται πλέον από προγραμματιζόμενα και αυτοματοποιημένα παραγωγικά συστήματα και μηχανήματα, που αντικαθιστούν τους ανειδίκευτους εργάτες, ακόμα και τους ειδικευμένους τεχνίτες, παράγοντας παράλληλα προϊόντα ανώτερης ποιότητας μαζικά, στον μισό χρόνο με το μισό κόστος. Επίσης, για τον χειρισμό των παραπάνω μηχανημάτων, απαιτείται η συμβολή ανθρώπων με εξειδικευμένες γνώσεις. Οι ραγδαίες τεχνολογικές εξελίξεις μειώνουν συνεχώς την ανάγκη για χειρωνακτική εργασία και αυξάνουν τη ζήτηση για εργαζόμενους με πρόσθετα προσόντα και εξοικειωμένους με τη χρήση της νέας τεχνολογίας. Έτσι, με την εισαγωγή χειριστών και εργατών για το συγκεκριμένο μηχάνημα, μπορούμε πλέον να παράγουμε αγαθά αρκετά πιο πολύπλοκα και εξειδικευμένα, αναβαθμίζοντας έτσι την παραγωγική διαδικασία μίας βιομηχανίας.

1.3 Παραγωγική διαδικασία και έλεγχος

Ως αποτέλεσμα των προαναφερθέντων, η εισαγωγή καινοτομιών και νέων τεχνικών παραγωγής σε μία νεοσύστατη βιομηχανία, έφερε αυτομάτως την ανάγκη **ενός άμεσου και αδιάλειπτου ελέγχου** στο όλο σύστημα παραγωγής. Ανάγκη η οποία σίγουρα ήταν σημαντική ακόμα και στην προ Βιομηχανικής επανάστασης εποχή, πλέον όμως έχει αποκτήσει υποχρεωτικό χαρακτήρα, εάν μία βιομηχανική μονάδα παραγωγής θέλει να διατηρεί την λειτουργία και την ποιότητα των αγαθών που παράγει. Άλλωστε όπως είναι γνωστό, όπου υπάρχει τεχνολογία, ο έλεγχος πρέπει να είναι ακόμα πιο ανώτερος!!! Ερχόμενοι στις αρχές του 20^{ου} αιώνα, είναι εμφανής η τάση κατά την οποία το περιβάλλον μίας βιομηχανίας απαιτεί την παραγωγική διαδικασία να μην εξετάζεται σαν μεμονωμένη δραστηριότητα αλλά σαν ολοκληρωμένο σύστημα το οποίο περιλαμβάνει όλα τα απαραίτητα στοιχεία που επηρεάζουν σημαντικά την λειτουργία και την παραγωγική δύναμη της βιομηχανικής μονάδος.

Ολοένα και περισσότερες νέες τεχνολογικές εφαρμογές ενθουλακώνονται στο παλαιό παραγωγικό μοντέλο, αναβαθμίζοντας όλο και περισσότερο το παραγόμενο προϊόν αλλά και την παραγωγική διαδικασία γενικότερα.

Παράλληλα όμως, όλο και μεγαλύτερη γίνεται η ανάγκη για έλεγχο όχι ενός μεμονωμένου μηχανήματος ή εφαρμογής, αλλά ολόκληρου του συστήματος παραγωγής. Ο έλεγχος της συμβατότητας των μηχανημάτων μεταξύ τους καθώς και με τα νέα παραγόμενα υλικά καθώς και η παράλληλη λειτουργία τους είναι πλέον πιο σημαντικός από ποτέ, για να μπορέσουμε να επιτύχουμε ένα σωστό αγαθό, σταθερά υψηλής ποιότητας.



Ένα άλλο σημαντικό στοιχείο που καθιστά τον βιομηχανικό έλεγχο απαραίτητο κομμάτι στην λειτουργία της σύγχρονης βιομηχανικής μονάδος, είναι **η αύξηση της ανταγωνιστικότητας**. Όλες οι σύγχρονες παραγωγικές μονάδες έχουν επενδύσει ένα αρκετά μεγάλο μερίδιο του κεφαλαίου τους στην ορθή και σωστή λειτουργία του συστήματος παραγωγής τους, για να μπορέσουν να έχουν ένα προϊόν άρτιο, σύγχρονο και συνάμα ασφαλές. Μόνο έτσι θα καταφέρουν να κρατηθούν ψηλά στην μάχη της αγοράς. Τέλος, με την πάροδο των ετών η τεχνολογία ολοένα και πλησιάζει τα όριά της. Τα κυριότερα

συστήματα αυτοματοποίησης, δεν είναι μόνο ακριβά, αλλά παρουσιάζουν μεγάλες δυσκολίες κατά την εφαρμογή, προσαρμογή, λειτουργία και ιδίως στην συντήρησή τους. Τις περισσότερες φορές μάλιστα παρέχουν ελάχιστες δυνατότητες επέκτασης ή προσαρμογής στις εξελισσόμενες απαιτήσεις της μονάδας. Γι' αυτόν τον λόγο, κρατώντας παράλληλα υψηλό το επίπεδο ανταγωνισμού, η κάθε παραγωγική μονάδα θα πρέπει να έχει δώσει βασικό ρόλο στο κομμάτι του ελέγχου. Το κάθε μηχάνημα που απαρτίζει την βιομηχανική μονάδα έχει μία ημερομηνία λήξης. Με τον σωστό έλεγχο και την άμεση συντήρησή τους, η ημερομηνία αυτή μπορεί να παραταθεί αρκετά, ανεβάζοντας έτσι το επίπεδο ασφάλειας της μονάδας αλλά και του ανθρώπινου δυναμικού, επιτυγχάνοντας παράλληλα τον βασικό σκοπό και στόχο της κάθε βιομηχανίας: την άρτια, ασφαλή και αδιάλειπτη λειτουργία της!!!

1.4 Συστήματα Ελέγχου Βιομηχανιών

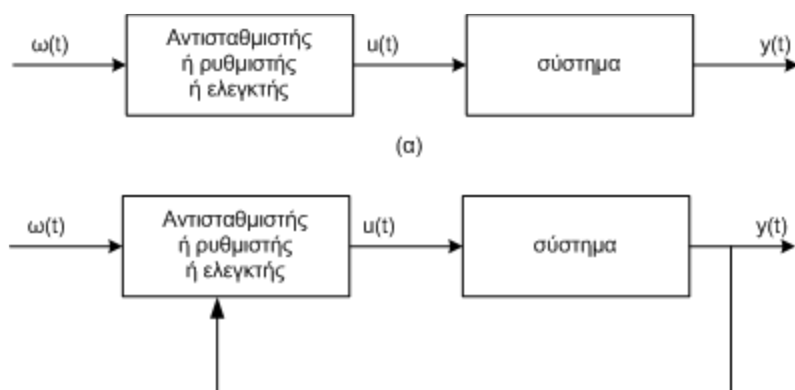
Ως συνέπεια των αιτιών που αναλύθηκαν στις παραπάνω παραγράφους, γεννήθηκε η ανάγκη μίας ολοκληρωμένης και συνεχόμενης βιομηχανικής ελεγκτικής διαδικασίας. Για να μπορέσουμε να κατανοήσουμε καλύτερα την έννοια του όρου, μπορούμε να πούμε ότι ως σύστημα ελέγχου ορίζεται μια συσκευή, ή ένα σύνολο διατάξεων καλύτερα, που διαχειρίζεται εντολές, διευθύνει ή ρυθμίζει τη συμπεριφορά των άλλων συσκευών ή συστημάτων μίας βιομηχανικής μονάδας.

1.4.1 Τύποι συστημάτων

Μπορούμε να διαχωρίσουμε τα συστήματα αυτά σε δύο βασικές κατηγορίες

- Συστήματα Ελέγχου Ανοικτού Βρόχου
- Συστήματα ελέγχου Κλειστού Βρόχου

Ανοιχτό χαρακτηρίζεται το σύστημα που η είσοδος του $u(t)$ δεν είναι συνάρτηση της εξόδου $y(t)$, ενώ **κλειστό** είναι το σύστημα που η είσοδος του $u(t)$ είναι συνάρτηση της εξόδου $y(t)$. Ένα σύστημα κλειστού βρόχου ονομάζεται επίσης ένα **σύστημα ελέγχου ανάδρασης**. Το ανθρώπινο σώμα είναι ένα κλασικό παράδειγμα των συστημάτων ανάδρασης.



Αναφερόμενοι σε συστήματα αυτομάτου ελέγχου κλειστού βρόχου, το σήμα εισόδου $u(t)$ που δέχεται το σύστημα δεν παράγεται από μία γεννήτρια αλλά πρόκειται για την έξοδο ενός συστήματος ελέγχου που περιέχεται. Ο βρόχος ελέγχου αποτελείται από συσκευές όπως διάφοροι αισθητήρες, ελεγκτές, αλγόριθμους ελέγχου καθώς και ενεργοποιητές, τοποθετημένοι βέβαια με τρόπο ώστε να μπορεί να παράγεται άμεσα και χωρίς λάθη το επιθυμητό σήμα εισόδου. Αντιλαμβανόμενοι την αξία και την θέση που έχουν, μπορούμε να πούμε ότι οι **ελεγκτές**, ή όπως αλλιώς αναφέρονται **Ελεγκτές Τριών Όρων**, αποτελούν τα ζωτικά όργανα μιας αυτοματοποιημένης ελεγκτικής διαδικασίας μιας βιομηχανικής μονάδος.

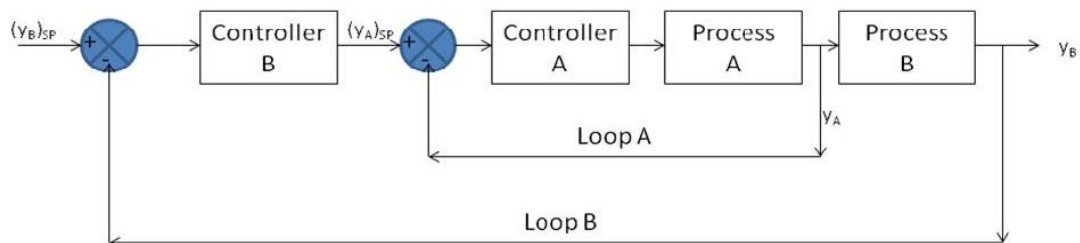
1.4.2 Ελεγκτές

Η γέννηση των συσκευών αυτών παρατηρείται από τα πρώτα χρόνια του 18^{ου} αιώνα, όπου ο James Watt εφάρμοσε αναλογικό έλεγχο υπό την μορφή φυγοκεντρικού κυβερνήτη για τον έλεγχο ενός ανεμόμυλου και μερικά χρόνια αργότερα για τον έλεγχο ατμομηχανών, τα οποία αποτέλεσαν την κινητήρια δύναμη της βιομηχανικής επανάστασης. Οι βιομηχανικοί ελεγκτές έκαναν την εμφάνισή τους κατά την δεύτερη δεκαετία του 20^{ου} αιώνα. Ακόμα και τότε, έστω και σε πρώιμο στάδιο, αποτελούσαν την καρδιά των ελεγκτικών συστημάτων. Έκτοτε, εκατομμύρια βιομηχανικοί ελεγκτές λειτουργούν ικανοποιητικά σε όλων των ειδών τις βιομηχανικές διαδικασίες ανά τον κόσμο, κάτω από δύσκολες και αντίξοες λειτουργικές συνθήκες. Είναι βέβαιο δε, ότι η Βιομηχανική επανάσταση θα ήταν αδύνατη χωρίς αυτούς τους απλούς, αξιόπιστους και ευρέως διαδεδομένους ελεγκτές. Παράλληλα, η ικανοποίηση των σημερινών απαιτήσεων της παραγωγής θα ήταν αδύνατη χωρίς αυτούς. Σε αυτό το σημείο αξίζει να σημειωθεί ότι από διεθνής έρευνες που έχουν γίνει έχει διαπιστωθεί ότι το 95% των κλειστών βρόχων ελέγχου που χρησιμοποιούνται σήμερα βασίζονται στους βιομηχανικούς ελεγκτές τριών όρων.

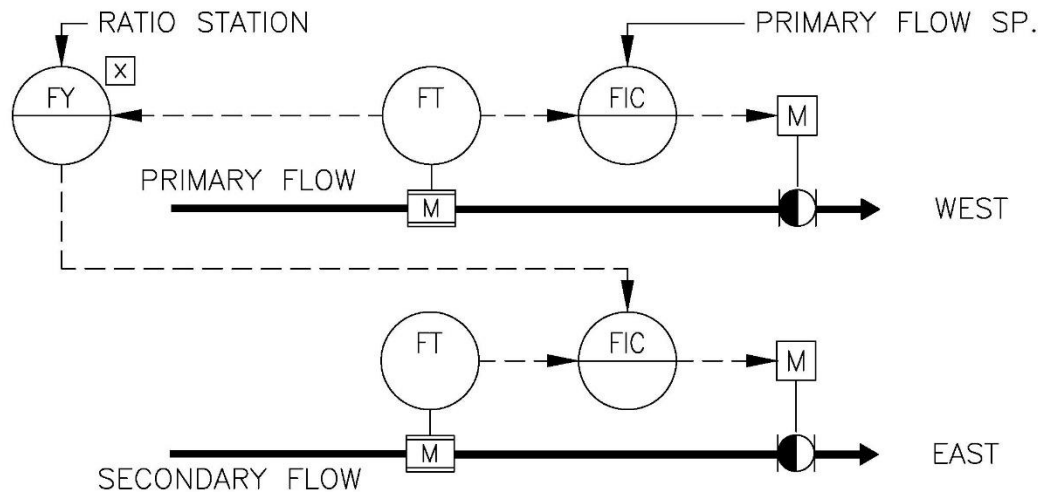
1.5 Τύποι βιομηχανικού ελέγχου

Στην πράξη, μία βιομηχανική διαδικασία συχνά περιέχει ένα πλήθος βρόγχων ελέγχου, κάθε ένα από τους οποίους έχει διαφορετικό σκοπό καθώς και διαφορετικές προδιαγραφές. Σε πολλές περιπτώσεις τα δομικά στοιχεία των διαφόρων βρόγχων αλληλοεπιδρούν και μία πιθανή μεταβολή ενός στοιχείου επηρεάζει άμεσα και όλα τα άλλα. Η ανάλυση και η σύνθεση των πολυμεταβλητών ελεγκτών είναι δύσκολη και απαιτεί βαθιά γνώση της θεωρίας πολυμεταβλητών συστημάτων ελέγχου. Υπάρχουν όμως πάρα πολλές περιπτώσεις όπου τα στοιχεία ενός συστήματος ελέγχου μίας βιομηχανίας δεν επηρεάζουν άλλους, ή μπορούν να θεωρηθούν ανεξάρτητοι. Ουσιαστικά διακρίνονται οι εξής τέσσερις τύποι βιομηχανικού ελέγχου:

1. Ο **έλεγχος μοναδικού βρόγχου** (*single loop control*) όπου μία μεταβλητή της ελεγχόμενης διαδικασίας ελέγχεται αποκλειστικά από ένα ελεγκτή που δεν έχει διασύνδεση με άλλον,
2. Ο **διαδοχικός έλεγχος** (*cascade control*) όπου ο απλός έλεγχος ενός μοναδικού βρόγχου δεν ικανοποιεί τις προδιαγραφές ελέγχου της διαδικασίας λόγω αλληλεπίδρασης των μεταβλητών της διαδικασίας και απαιτούνται δύο ή περισσότεροι διασυνδεδεμένοι βρόγχοι,
3. Ο **έλεγχος λόγου** (*ratio control*) όπου απαιτείται η τήρηση ενός σταθερού λόγου μεταξύ δύο (ή περισσότερων) μεταβλητών της ελεγχόμενης διαδικασίας, και τέλος
4. Ο **έλεγχος πρόσμιξης** (*blending control*) όπου απαιτείται ο ποσοστιαίος έλεγχος δύο ή περισσότερων ποσοτήτων.



Διάγραμμα Cascade Control



Διάγραμμα Ratio Control

1.6 Συστήματα αυτομάτου βιομηχανικού ελέγχου (ICS)

Παρ' όλη όμως αυτή την ευρεία χρήση των διάφορων ελεγκτών και αισθητήρων στις βιομηχανικές μονάδες, σε αρκετές περιπτώσεις οι βρόγχοι ελέγχου δεν ανταποκρίνονται στις απαιτήσεις που έχουν τεθεί, αρχικά. Δεν είναι



άλλωστε λίγες οι έρευνες όπου μας υποδεικνύουν ότι ο έλεγχος της συμπεριφοράς μίας διαδικασίας δεν είναι τόσο εύκολος όσο μπορεί να θεωρήσει κανείς. Η άποψη αυτή μας έχει οδηγήσει σε πλήρη επαναπροσδιορισμό της ελεγκτικής διαδικασίας που πρέπει να προβλέπεται σε μία σύγχρονη βιομηχανική μονάδα.

1.6.1 Λόγοι αυτοματοποίησης

Κατ' αρχήν μία βελτιωμένη διαδικασία ελέγχου, διαμορφωμένη κατάλληλα ώστε να μας δίνει επιπλέον επιλογές όσον αφορά τον ολικό έλεγχο της βιομηχανικής μονάδος (π.χ. κατάλληλη εκλογή των μεταβλητών της εισόδου και της εξόδου) είναι δυνατόν να τελειοποιήσει την απόδοση του συστήματος ελέγχου. Συγκεκριμένα, τα τελευταία χρόνια, έχει δοθεί ιδιαίτερη προσοχή σε μία τεχνική η οποία ονομάζεται **“Interaction between Design and Control”**.

Για να μπορέσει όμως να πραγματοποιηθεί κάτι ένα τέτοιο σύστημα, σημαντικό ρόλο έχουν οι διάφοροι αισθητήρες και ελεγκτές που εμπεριέχονται σε αυτό. Η διαδικασία όμως της πλήρους προσαρμογής τους στο σύστημα ή ακόμα και της αντικατάστασής του από νέους αισθητήρες ελέγχου, προσφέροντάς μας ένα αναμενόμενο και άρτιο επίπεδο συμπεριφοράς όλου του συστήματος ελέγχου, είναι κάτι δύσκολο, ίσως και ακατόρθωτο σε αρκετές περιπτώσεις.

Γίνεται λοιπόν φανερό πλέον η ανάγκη ενός ολοκληρωμένου συστήματος αυτόματης ρύθμισης των διαφόρων ελεγκτών και αισθητήρων ενός συστήματος ελέγχου, για να μπορέσουμε να υπερκεράσουμε τα παραπάνω εμπόδια. Χάρης αυτό, τα δομικά στοιχεία του βρόχου ελέγχου θα μπορούν να βρίσκουν αυτόματα και αυτόνομα τις παραμέτρους ρυθμίσεών τους, χαρίζοντας έτσι στο σύστημα ελέγχου μία ολοκληρωμένη, αδιάληπτη και ομαλή λειτουργία.

1.6.2 Κατηγορίες Συστημάτων Αυτομάτου Ελέγχου

Ανάλογα με την λειτουργικότητα και την πολυπλοκότητα των εκάστοτε βιομηχανικών μονάδων, τα βιομηχανικά συστήματα αυτομάτου ελέγχου (**Industrial Control Systems - ICS**) διαφοροποιούνται σε διάφορους τύπους. Συνηθέστερα όμως και ευρέως χρησιμοποιούμενα στον βιομηχανικό χώρο είναι τα εξής:

1. Προγραμματιζόμενοι Λογικοί Ελεγκτές (*PLC's*)
2. Κατανεμημένα Συστήματα Ελέγχου (*DCS*)
3. Ενσωματωμένος Έλεγχος (*Embedded Control*)
4. Συστήματα Αυτόματου Εποπτικού Ελέγχου και Συλλογής Δεδομένων (*SCADA*)

1.6.2.1 Προγραμματιζόμενοι Λογικοί Ελεγκτές (*PLC's*)

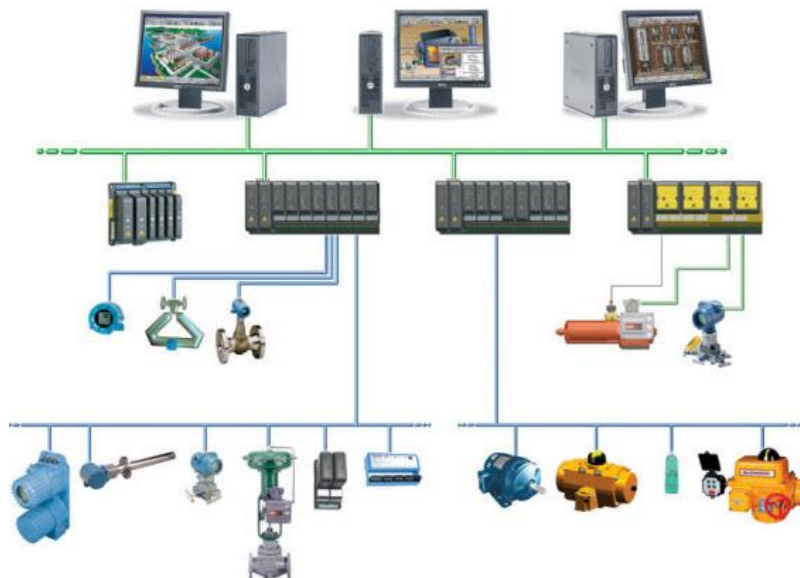


Πρόκειται για υπολογιστικές μονάδες βιομηχανικής χρήσης, ικανές να προγραμματιστούν για να εκτελούν διάφορες εφαρμογές βιομηχανικού ελέγχου. Βασίζουν την λειτουργία τους κυρίως σε λογικές δυαδικές πράξεις, ενώ ορισμένα μοντέλα χρησιμοποιούν συστήματα χρόνου, διαρκούς ελέγχου, ακόμα και αναλογικά.

Τα **PLC's** παρουσιάζονται σαν τα βασικά δομικά κομμάτια σε συστήματα DCS και SCADA, αλλά μπορούν να χρησιμοποιηθούν και αυτόνομα σε μικρότερες βιομηχανικές μονάδες για τον ρυθμιστικό έλεγχο διακριτών διεργασιών όπως σε γραμμές συναρμολόγησης αυτοκινήτων.

1.6.2.2 Κατανεμημένα Συστήματα Ελέγχου (DCS)

Τα **κατανεμημένα συστήματα ελέγχου (DCS)** πρόκειται για συστήματα ελέγχου, ειδικά σχεδιασμένα για τον έλεγχο πολύπλοκων και γεωγραφικά κατανεμημένων εφαρμογών. Όπως υποδηλώνει το όνομα, οι ελεγκτές διανέμονται σε ολόκληρη την περιοχή της εγκατάστασης, όπου συνδεόμενοι με διάφορους αισθητήρες ή ενεργοποιητές και με την βοήθεια ειδικών υπολογιστικών συστημάτων όπου επικοινωνούν μέσω δικτύου υψηλής ταχύτητας, μας δίνουν μία ολική εικόνα της παρατηρούμενης βιομηχανικής μονάδος. Όπως είναι λογικό, η ανθρώπινη αλληλεπίδραση ελαχιστοποιείται, μειώνοντας έτσι το κόστος εργασίας καθώς και τα εργατικά ατυχήματα.



Δομή ενός Συστήματος Κατανεμημένου Ελέγχου

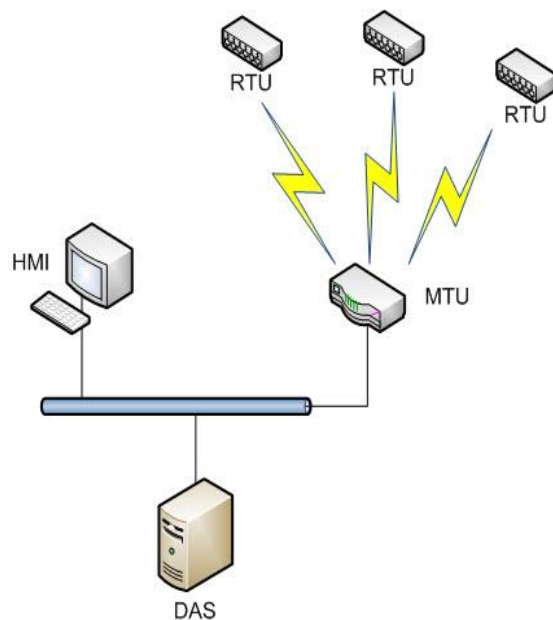
1.6.2.3 Ενσωματωμένος Έλεγχος (Embedded Control)

Στο συγκεκριμένο ελεγκτικό σύστημα, πολλά μικρά στοιχεία ελέγχου ενσωματώνονται στην παραγωγική διαδικασία. Με την βοήθεια ενός υπολογιστή συνδεδεμένος στο βιομηχανικό δίκτυο, μας δίνει την δυνατότητα να ασκούμε έλεγχο στην βιομηχανική μας μονάδα.

1.6.2.4 Συστήματα Αυτόματου Εποπτικού Ελέγχου και Συλλογής

Δεδομένων (SCADA)

Ένα **σύστημα SCADA** πρόκειται βασικά για μία ολόκληρη διαδικασία αυτόματης εποπτείας και ελέγχου όπου συλλέγει και επεξεργάζεται δεδομένα από διάφορα όργανα που βρίσκονται τοποθετημένα είτε μέσα στην παραγωγική μονάδα είτε ακόμα και εκτός αυτής. Αποτελούμενο από διάφορα υποσυστήματα όπως μονάδες τηλεμετρίας, διάφορες διεπαφές ανθρώπου – μηχανής (*HMI*), καθώς ακόμα και *PLC's*, τα συστήματα SCADA παρουσιάζονται ως τα πλέον διαδεδομένα, παραμετροποιήσιμα και λειτουργικά συστήματα αυτομάτου ελέγχου που μπορούν να χρησιμοποιηθούν σε μία βιομηχανία. Στα παρακάτω κεφάλαια της πτυχιακής μας εργασίας, θα εμβαθύνουμε στην δομή, την χρήση και τα πλεονεκτήματα που μπορούν να μας παρέχουν τα συγκεκριμένα συστήματα αυτομάτου ελέγχου.



Κεφάλαιο 2

Συστήματα Βιομηχανικού Ελέγχου SCADA

2.1 Τα συστήματα SCADA

Τα τελευταία 30 χρόνια η ανάπτυξη στο χώρο των βιομηχανικών συστημάτων υπήρξε ιδιαίτερα σημαντική, έχοντας συμβάλλει καθοριστικά σε αυτό η τεχνολογία των ψηφιακών συσκευών, είτε με τη χρησιμοποίηση ευφυών συσκευών είτε με τη χρήση ηλεκτρονικών υπολογιστών και επικοινωνιακών δικτύων. Οι δεξαμενές καυσίμου, η εξοικονόμηση ενέργειας, τα συστήματα φωτισμού καθώς και τα επίπεδα νερού είναι μόνο μερικά από τα πράγματα που ίσως κάθε σύγχρονη βιομηχανική μονάδα θα πρέπει να εστιάσει για να εξασφαλιστεί η ομαλή και αδιάληπτη λειτουργία της. Η ανάγκη για αξιόπιστη ανταλλαγή δεδομένων σε πραγματικό χρόνο και οι ενέργειες ελέγχου που απαιτούνται σε ένα ολοκληρωμένο σύστημα παραγωγής, είχαν ως αποτέλεσμα την ανάπτυξη των συστημάτων SCADA.

Ο όρος **SCADA** είναι ακρωνύμιο του όρου *Supervisory Control And Data Acquisition*, που σημαίνει **Εποπτικός Έλεγχος και Συλλογή Δεδομένων**. Περιγράφει μια γενική κατηγορία συστημάτων βιομηχανικού αυτομάτου ελέγχου και τηλεμετρίας. Πρόκειται για συστήματα ελέγχου μεγάλης κλίμακας για αυτοματοποιημένες βιομηχανικές διεργασίες.



Αίθουσα χειρισμού σταθμών SCADA σε βιομηχανία

Με άλλα λόγια, δεν αναφέρεται σε μία συγκεκριμένη τεχνολογία αλλά σε ένα **μεγάλο λογισμικό πακέτο** που συγκεντρώνει μέσω ενός δικτύου μεγάλης κλίμακας, όλες τις απαιτούμενες πληροφορίες από το ελεγχόμενο σύστημα ώστε να μπορεί να υλοποιήσει μία

σειρά από λειτουργικές συναρτήσεις όπως λήψη και αποθήκευση δεδομένων, παραγωγή γραφημάτων, γραφική απεικόνιση της διαδικασίας με κινούμενη εξομοίωση των μηχανών, αναγγελία σημαντικών καταστάσεων και καταστάσεων κινδύνου, παρουσίαση σημαντικών τιμών σε ενεργά πεδία, στατιστική επεξεργασία των δεδομένων (ακόμα και off-line), παρακολούθηση της κατάστασης του βιομηχανικού δικτύου, επικοινωνία με εξωτερικές βάσεις δεδομένων και άλλες οι οποίες καθιστούν πραγματικότητα τη ρήση “με το σύστημα SCADA έχω το εργοστάσιο στην οθόνη μου”.

Παρακάτω παρατίθενται δύο σημαντικοί ορισμοί της έννοιας **SCADA** και η προέλευση τους.

- *Με τον όρο SCADA αναφερόμαστε στην τεχνολογία που επιτρέπει στο χρήστη να «συλλέγει» δεδομένα από μία ή περισσότερες εγκαταστάσεις και/ή να αποστέλλει στοιχειώδεις εντολές ελέγχου σε αυτές.*

(SCADA: Supervisory Control and Data Acquisition, Stuart A. Boyer, ISA The Instrumentation, Systems, and Automation Society; 3rd edition)

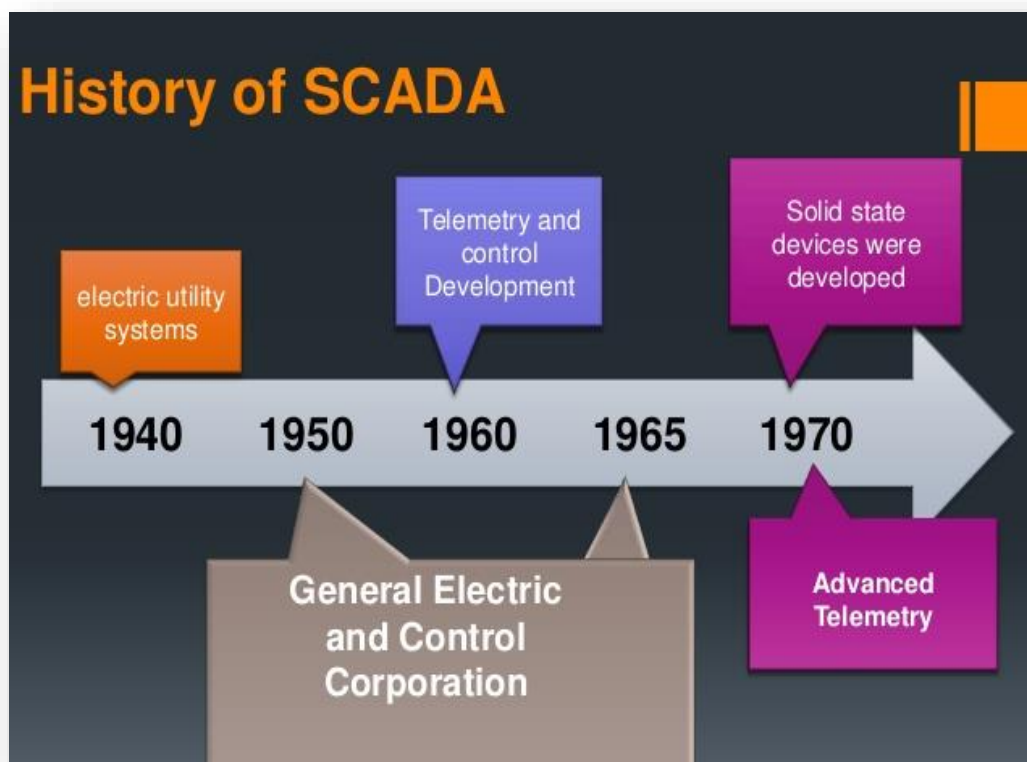
- *Σύστημα το οποίο λειτουργεί με κωδικοποιημένα σήματα σε διαύλους επικοινωνίας με σκοπό την παροχή ενεργειών ελέγχου σε εξοπλισμό συσκευών RTU.*

(IEEE Standard C37.1-1994, Definition, Specification, and Analysis of System Used for Supervisory Control, Data Acquisition, and Automatic Control)

2.2 Ιστορική Αναδρομή

Η ανάγκη του εποπτικού ελέγχου, σαν μία γενικότερη έννοια, αναπτύχθηκε για πρώτη φορά σε ηλεκτρικά συστήματα βιομηχανικών μονάδων. Η ανάγκη για την δημιουργία ενός απομακρυσμένου υποσταθμού όπου θα μπορεί, χωρίς την παρουσία του ανθρώπινου παράγοντα, να ελέγχει σε πραγματικό χρόνο την κατάσταση των συσκευών που αποτελούν την παραγωγική διαδικασία έγινε γρήγορα αντιληπτή. Έτσι, στην αρχή της δεκαετίας του 1940, παρατηρούμε την προσπάθεια ένωσης μέσω ενός απλού ζεύγους καλωδίων, όλων των εξοπλισμών της βιομηχανικής μονάδας με τον απομακρυσμένο υποσταθμό. Αναλυτικότερα, για πρώτη φορά εμφανίζεται η προσπάθεια αξιοποίησης της **πολυπλεξίας** σε ένα ζεύγος τηλεφωνικών γραμμών λαμβάνοντας μάλιστα ιδέες από το μαγνητικό βαθμωτό διακόπτη που αναπτύχθηκε από εταιρείες τηλεφωνίας στη δεκαετία του '30. Επίσης, η ασφάλεια του όλου συστήματος παίζει βασικό ρόλο στην ομαλή λειτουργία μίας μονάδας. Γι' αυτόν τον λόγο, υιοθετήθηκε ένα **πρότυπο επιλογής** –

επιβεβαίωσης λειτουργίας (select – check – operate) κατά το οποίο ο χειριστής περίμενε την επιβεβαίωση από τη συσκευή πριν τελικά αρχίσει να λειτουργεί. Λαμβάνοντας τα αποτελέσματα των παραπάνω κινήσεων, οι εταιρίες **Westinghouse** και **North Electric Company** ξεκινούν την ανάπτυξη του ελεγκτικού συστήματος **Visicode**. Η **General Electric** καθώς και η **Control Corporation**, στις αρχές τις δεκαετίας του 1950 και το 1965, ξεκινούν επίσης την ανάπτυξη ενός ανεξάρτητου προγράμματος οπτικού ελέγχου. Τα προγράμματα αυτά βρήκαν αμέσως χρήση σε αγωγούς μεγάλων αποστάσεων, εταιρείες φυσικού αερίου και ακόμη και στα αεροδρόμια για τα φώτα του διαδρόμου προσγείωσης. Παράλληλα, κατά το 1960, η **τηλεμετρία** αρχίζει να κάνει την εμφάνισή της για σκοπούς παρακολούθησης της παραγωγικής διαδικασίας.



Διάγραμμα δημιουργίας των συστημάτων SCADA

Όπως αναφέραμε και παραπάνω, η σύνδεση των συσκευών της βιομηχανικής μονάδος, πριν το 1970, γινόταν κυρίως με την χρήση καλωδίων, καθώς οι συσκευές σταθερής κατάστασης ήταν ακόμα σε πρώιμο στάδιο. Όμως, με την έλευση των υπολογιστών στην βιομηχανία, το κόστος σύνδεσης μειώθηκε αισθητά, μετακινώντας τους εργάτες από την χειρωνακτική παραγωγή στον χειρισμό των υπολογιστικών συστημάτων. Η αναβάθμιση από τους αρχικούς υπολογιστές **τεχνολογίας 8bit σε νέους 16bit** όπως και η **εξέλιξη των μικροεπεξεργαστών** τα επόμενα χρόνια, πρόσφεραν μία ευελιξία στον προγραμματισμό και την επικοινωνία μεταξύ

των μονάδων συλλογής δεδομένων του βιομηχανικού τομέα. Αυτή ήταν η γέννηση του συστήματος εποπτικού ελέγχου **SCADA**

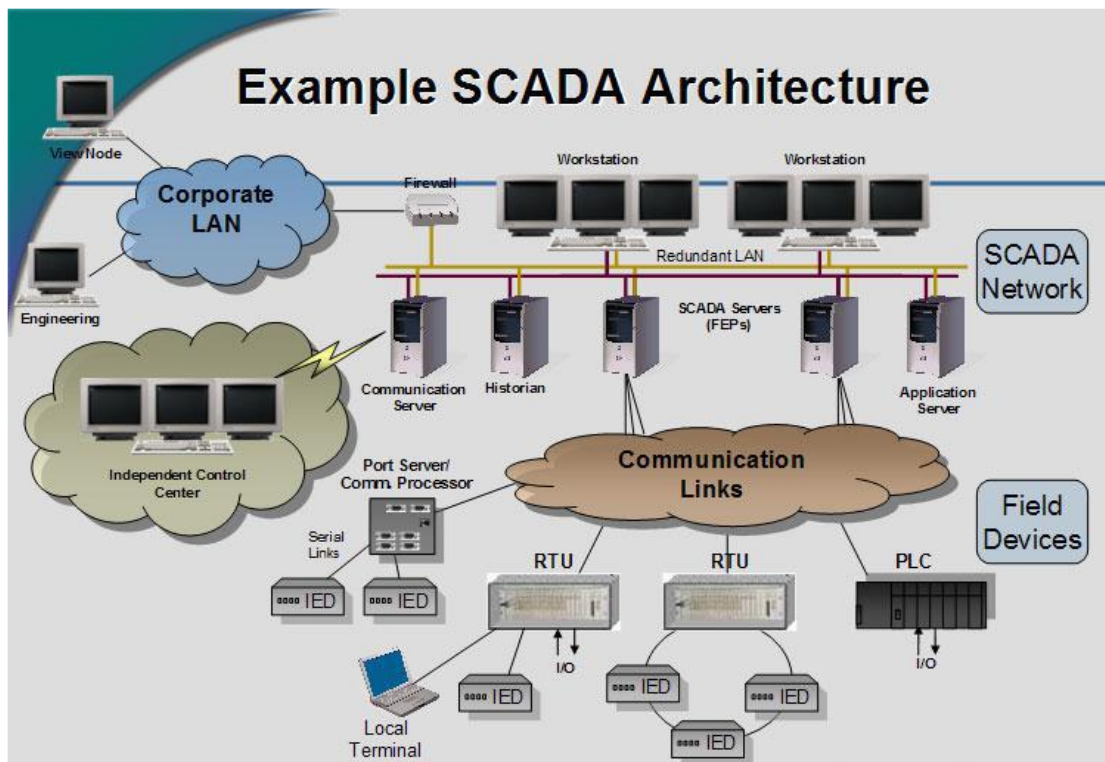
2.3 Αρχιτεκτονική συστήματος SCADA

Εκτός από το λογισμικό εποπτείας και ελέγχου, από άποψη υλικοτεχνικής υποδομής, ένα σύστημα SCADA αποτελείται, και από μια συλλογή αισθητηρίων και ευφυών συσκευών. Αναλυτικότερα, μπορούμε να τα χωρίσουμε στις εξής κατηγορίες:

1. Ένα **δίκτυο έξυπνων συσκευών (Intelligent Electronic Devices - IED's)** που συνδέεται με το σύστημα όπου επιθυμούμε να εποπτεύσουμε μέσω αισθητήρων και ελεγκτών. Το δίκτυο αυτό μας δίνει τη δυνατότητα να μετρήσουμε και να ελέγξουμε συγκεκριμένα στοιχεία του συστήματος.
2. Ο **Κεντρικός Υπολογιστικός Σταθμός (Master Terminal Unit – MTU)**. Αποτελείται από ένα κεντρικό υπολογιστικό σύστημα, αρκετά μεγάλης ισχύος, στον οποίο βρίσκεται το λογισμικό SCADA εγκατεστημένο, όπως και το πρόγραμμα της εκάστοτε εφαρμογής. Με άλλα λόγια, πρόκειται για τον πυρήνα του εποπτικού συστήματος SCADA.
3. **Τερματικές μονάδες (Remote Telemetry Unit - RTU's)** όπου συνδέονται στους διάφορους αισθητήρες. Έχουν την δυνατότητα ανάγνωσης και μετατροπής των δεδομένων που δέχονται από τους αισθητήρες σε ψηφιακά δεδομένα, σύμφωνα με το πρωτόκολλο επικοινωνίας που υποστηρίζουν. Επίσης αποτελούν βασικό κομμάτι της τηλεμετρίας του εποπτικού συστήματος διότι είναι υπεύθυνα για την αποστολή των δεδομένων στο Κεντρικό Εποπτικό Σύστημα καθώς και λήψης εντολών από το σύστημα.
4. **Προγραμματιζόμενους λογικούς ελεγκτές (Programmable Logic Controller - PLC's)**. Είναι συσκευές με παρόμοια λειτουργία με τις τερματικές μονάδες. Σε αντίθεση όμως με τα RTU, τα PLC's δεν παρέχουν δυνατότητες τηλεμετρίας. Διαθέτουν όμως μεγάλες δυνατότητες ελέγχου καθώς υποστηρίζουν και περισσότερους αλγορίθμους ελέγχου (*IEC 61131-3*). Επίσης, λόγω του μικρότερου όγκου τους, είναι πιο οικονομικά.
5. Ένα **σύστημα τηλεμετρίας** για την διασύνδεση όλων των ελεγκτών. Παρέχει διασύνδεση των ευφυών και περιφερειακών συσκευών (IED's, PLC's, RTU's) με κέντρα ελέγχου και βάσεις δεδομένων είτε ενσύρματα είτε ασύρματα. Για την πιο

ομαλή διασύνδεση των παραπάνω συσκευών, χρησιμοποιούνται σύγχρονα βιομηχανικά πρωτόκολλα επικοινωνίας, για την σύνδεση των υπηρεσιών λογισμικού.

6. Μία **διεπαφή ανθρώπου – μηχανής (Human Machine Interface – HMI)**. Είναι η συσκευή η οποία βοηθά στην αλληλεπίδραση του ανθρώπου στην βιομηχανική διαδικασία καθώς και στον έλεγχο της. Πρακτικά, αποτελεί τον τρόπο με τον οποίο παρουσιάζονται δεδομένα σχετικά με την βιομηχανική λειτουργία στον άνθρωπο, δίνοντάς του έτσι την δυνατότητα να έχει έναν πλήρη έλεγχο καθώς και να επιδρά και ο ίδιος στην λειτουργία της βιομηχανικής διαδικασίας.



Αρχιτεκτονική ενός συστήματος SCADA

Η επικοινωνία μεταξύ του MTU και των RTU's μπορεί να είναι **ενσύρματη** (σειριακή επικοινωνία RS232, RS485, δίκτυα Profibus, Ethernet, τηλεφωνική σύνδεση, Internet) ή και **ασύρματη** (ραδιοκύματα, δορυφορική σύνδεση, μικρόκύματα). Οι πληροφορίες μεταφέρονται από τις RTU's στον MTU, όπου αφού επεξεργαστούν κατάλληλα καταγράφονται και προβάλλονται σε υπολογιστές που «τρέχουν» HMI λογισμικό όπου πραγματοποιείται ο μη αυτόματος έλεγχος και η εποπτεία των διεργασιών. Τυχόν αυτόματα σήματα ελέγχου που παράγονται στην MTU αποστέλλονται πίσω στις RTUs, οι οποίες με τη σειρά τους ενεργοποιούν τις διατάξεις μετατροπής και τους ελεγκτές των μηχανών

2.3.1 Κεντρική Τερματική Μονάδα (MTU)

Αποτελεί την **καρδιά** σε ένα σύστημα βιομηχανικού ελέγχου SCADA. Πρόκειται για ένα υπολογιστικό σύστημα το οποίο λαμβάνει τα δεδομένα που αποστέλλονται από τα διάφορα περιφερειακά συστήματα του ελεγκτικού μηχανισμού (αισθητήρες, IED's, RTU's), τα επεξεργάζεται και αποστέλλει τα ανάλογα αποτελέσματα για την ορθή και άρτια διαχείριση του συστήματος.

2.3.2 Αισθητήρες και ευφυείς συσκευές (IED's)

Πρόκειται για συσκευές όπου μας δίνουν την δυνατότητα ανίχνευσης τυχών αλλαγών και μεταβολών ορισμένων βασικών τιμών που επιθυμούμε να παρακολουθούμε στο βιομηχανικό μας σύστημα. Παρουσιάζοντας μία γενική έννοια του όρου, σαν αισθητήρα ορίζουμε μία συσκευή όπου μας δίνει την δυνατότητα να ανιχνεύσουμε αλλαγές ή γεγονότα σε κάποιο συγκεκριμένο χαρακτηριστικό που έχουμε ορίσει



Αισθητήρας Κίνησης

αυτόματα. Δέχεται ως είσοδο τυχών αλλαγές στην μορφή ή στην ποσότητα του χαρακτηριστικού που έχουμε ορίσει και μας δίνει μία αντίστοιχη έξοδο, πολλές φορές είτε ως ηλεκτρικό είτε ως οπτικό σήμα, μαζί με συγκεκριμένα δεδομένα που αφορούν την μεταβολή αυτή. Σαν εξέλιξη των συσκευών αυτών, οι **έξυπνοι αισθητήρες** πρόκειται για συσκευές πάνω στις οποίες έχει ολοκληρωθεί τουλάχιστον ένα αισθητήριο στοιχείο και ένα κύκλωμα επεξεργασίας σήματος. Ο όρος **ευφυής** (*smart*) διατηρείται για να δηλώσει την επιμέρους ή την ολοκληρωτική ενσωμάτωση της κύριας μονάδας επεξεργασίας, η οποία προσθέτει ευφυΐα στο όλο σύστημα. Ορισμένα παραδείγματα χρήσης αισθητήρων σε βιομηχανικές μονάδες είναι οι αισθητήρες καπνού, υγρασίας καθώς και αισθητήρες ανίχνευσης της κίνησης.



Αισθητήρας Καπνού

2.3.3 Απομακρυσμένες Τερματικές Μονάδες (RTU)

Μία απομακρυσμένη τερματική μονάδα (RTU) αποτελείται από μία υπολογιστική μονάδα, υποκινούμενη από μικροεπεξεργαστές, όπου εγκαθίσταται σε μία απομακρυσμένη τοποθεσία στο ελεγχόμενο περιβάλλον της βιομηχανίας μας. Βασικό του μέλημα είναι η συλλογή δεδομένων από την ελεγχόμενη βιομηχανική μονάδα, η κωδικοποίησή τους σε μορφή εύκολα μεταδιδόμενη καθώς και η μετάδοσή τους, μέσω του βιομηχανικού δικτύου, είτε σε άλλες συσκευές στο δίκτυο, είτε στο κύριο εποπτικό υπολογιστή (MTU) είτε ακόμα και από τον master προς όλο το δίκτυο. Μπορούμε, με άλλα λόγια, να παρομοιάσουμε την σχέση των RTU's με τα MTU's σαν την σχέση master – slave! Τα RTU's είναι υπεύθυνα για την ύπαρξη της διασύνδεσης (τηλεμετρίας) μεταξύ των συσκευών που απαρτίζουν το σύστημα. Γι' αυτόν τον λόγο είναι εξοπλισμένα με κανάλια εισόδου, αναλογικά και ψηφιακά, για την ανίχνευση ή μέτρηση των χαρακτηριστικών που μας αφορούν, κανάλια εξόδου, επίσης αναλογικά και ψηφιακά, για τον έλεγχο και την επικοινωνία με το δίκτυο, ένδειξη συναγερμού όπως επίσης και ρελέ εξόδου.



Remote Telemetry Unit - RTU

2.3.4 Προγραμματιζόμενοι Λογικοί Ελεγκτές (PLC)

Όπως αναφέρεται και παραπάνω, οι προγραμματιζόμενοι λογικοί ελεγκτές (PLC's), παρουσιάζουν πολλά κοινά με τους RTU που αναλύθηκαν παραπάνω, και από πλευράς κατασκευής αλλά και από πλευράς λειτουργίας. Πρόκειται με άλλα λόγια για ένα ψηφιακό υπολογιστικό σύστημα, το οποίο χρησιμοποιείται για την αυτοματοποίηση των βιομηχανικών λειτουργιών, όπως είναι η παραγωγή αλλά και ο έλεγχός της. Αποτελούνται από μικροεπεξεργαστές όπου δίνουν την δυνατότητα επεξεργασίας των αποτελεσμάτων που συλλέγουν. Η κυριότερη διαφορά τους με τα RTU's βρίσκεται στον τομέα της αποστολής των δεδομένων. Τα RTU's διαθέτουν μηχανισμούς τηλεμετρίας έχοντας έτσι την δυνατότητα χρησιμοποίησής τους ακόμα και σε απομακρυσμένες περιοχές της βιομηχανικής μονάδος, κάτι που τα PLC's δεν έχουν την δυνατότητα να το υποστηρίξουν. Λόγω του παραπάνω, διαφοροποιείται και το κόστος χρήσης τους. Τα PLC's, λόγω του μικρότερου μεγέθους που έχουν, είναι πιο οικονομικά σε σχέση με τα ογκώδη RTU's, κάτι που τα κάνει ευρέως διαδεδομένα στα συστήματα SCADA.



Εξοπλισμός PLC εν λειτουργία

2.3.5 Διεπαφή Ανθρώπου - Μηχανής (HMI)

Η τεχνολογία HMI επιτρέπει με απλά λόγια την επικοινωνία του ανθρώπου με την μηχανή. Πρόκειται δηλαδή για το όργανο το οποίο παρουσιάζει δεδομένα από κάποια διεργασία σε έναν χειριστή και μέσω αυτής, ο χειριστής έχει τη δυνατότητα να ελέγξει και να εποπτεύσει τη διεργασία. Συνήθως συνδέεται με βάσεις δεδομένων του συστήματος SCADA και προγράμματα λογισμικού, με την παροχή διαγνωστικών ελέγχων και πληροφοριών διαχείρισης, όπως η προγραμματισμένη διαδικασία συντήρησης, έχοντας συγκεκριμένες σχηματικές παραστάσεις για ένα συγκεκριμένο αισθητήρα ή μηχανή, καθώς και σύστημα αντιμετώπισης προβλημάτων. Για την ορθότερη και ταχύτερη ανταπόκριση του χειριστή, το σύστημα HMI παρουσιάζει τις πληροφορίες αυτές γραφικά, με τη μορφή μιμικού διαγράμματος. Αυτό σημαίνει ότι ο χειριστής μπορεί να δει μια σχηματική παράσταση της μονάδας που ελέγχεται. Στα συστήματα SCADA, τα HMI εμφανίζονται κατά κόρον με την μορφή **Push Button Panels** καθώς και **Operator Interfaces** τα οποία μπορούν να χωριστούν σε τρεις βασικές κατηγορίες: **Text Displays (TD)**, **Operator Panels (OP)** και **Touch Panels (TP)**.

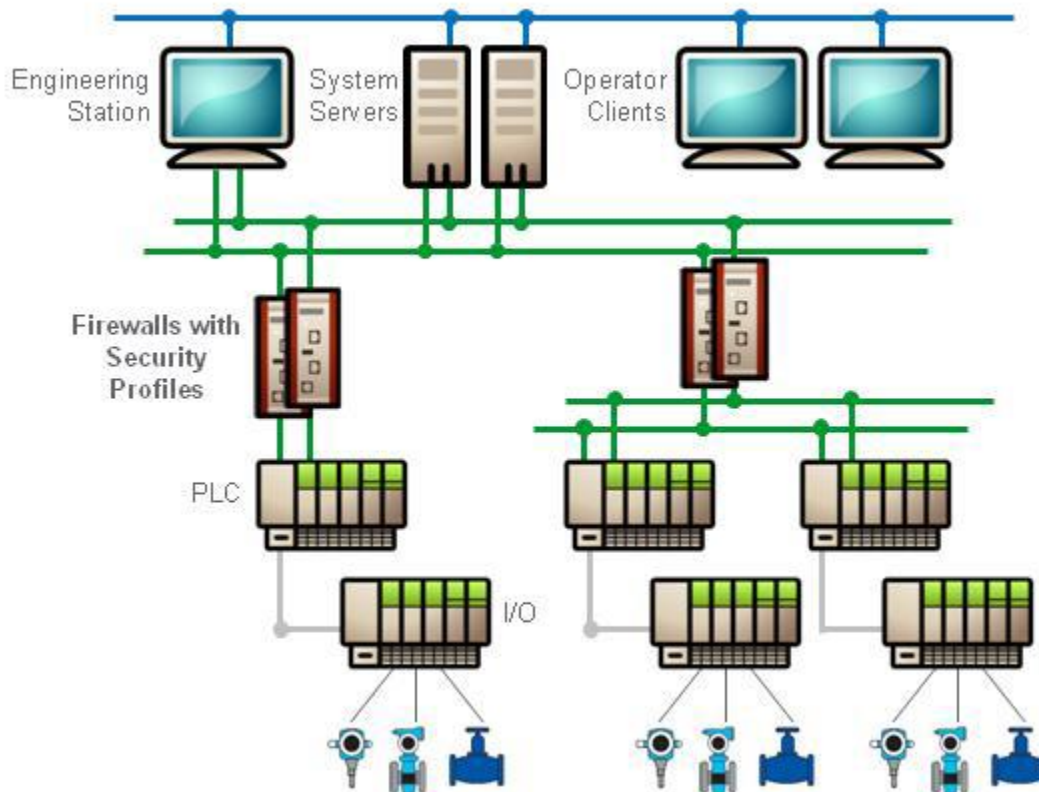


Σύστημα HMI για βιομηχανικό έλεγχο

2.3.6 Τηλεμετρία

Εξ' ορισμού, η **τηλεμετρία** είναι η τεχνολογία που μας επιτρέπει την διασύνδεση απομακρυσμένων συσκευών καθώς και την αποστολή δεδομένων μεταξύ τους, είτε ενσύρματα είτε ασύρματα.

Αποτελεί βασικό κομμάτι στην ελεγκτική και εποπτική διαδικασία μίας βιομηχανικής μονάδος διότι μας επιτρέπει την επιμέρους σύνδεση όλων των συσκευών σε ένα ενιαίο δίκτυο από το οποίο διακινούνται πληροφορίες, μετρήσεις, ακόμα και εντολές για τον σωστό και ποιοτικό έλεγχο της εκάστοτε μονάδος.



Παράδειγμα διασύνδεσης για έλεγχο απομακρυσμένων συσκευών

Για την δικτύωση του κεντρικού υπολογιστή με τους χρήστες (πληροφοριακό δίκτυο), χρησιμοποιούνται κυρίως τα πρωτόκολλα **Ethernet** ή **Token Ring** ενώ για την δικτύωση των περιφερειακών συσκευών (βιομηχανικό δίκτυο), έχουμε πρωτόκολλα όπως τα **Modbus**, **Profibus**, **DNP3** και **IEC 60870 - 5**, 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. Επίσης, σε ελεγκτικά συστήματα μεγαλύτερης έκτασης, υπάρχει η δυνατότητα **ασύρματης διασύνδεσης μέσω Wi-Fi, μικροκυμάτων, δορυφορικής σύνδεσης** ή ακόμα με την βοήθεια **ηλεκτρομαγνητικών κυμάτων** μεταφοράς δεδομένων.

2.3.7 Λογισμικό Προγραμματισμού SCADA

Η σχεδίαση SCADA εφαρμογών γίνεται μέσα από ειδικά λογισμικά πακέτα που προορίζονται για αυτόν το σκοπό. Τα λογισμικά αυτά βρίσκονται εγκατεστημένα στην κεντρική μονάδα ελέγχου (MTU) και υλοποιούν το σημαντικότερο έργο των προϊόντων με την ορολογία **Automation Software**. Συνήθως οι κατασκευαστές βιομηχανικού εξοπλισμού (*αισθητήρια όργανα, ελεγκτές, δίκτυα κλπ.*) παρέχουν και αντίστοιχο λογισμικό ελέγχου διεργασιών τύπου SCADA ή άλλου τύπου όπως Human - Machine Interface (HMI), οπτικοποίησης μεγεθών, καταγραφής συμβάντων, γραφικής απεικόνισης κλπ. Παρακάτω παρατίθεται ένας ενδεικτικός πίνακας με διάφορα λογισμικά SCADA και τους αντίστοιχους κατασκευαστές.

Λογισμικά SCADA

Λογισμικό SCADA	Πάροχος		Λογισμικό SCADA	Πάροχος
RsView	RsView		Entivity VLC	Entivity
InTouch	Wonderware		FactoryTack	Rockwell Automation
WinCC	Siemens		ISaGRAF	ISaGRAF
Cimplicity	GeFanuc		Prodigy	TASCOMP
CX-Supervisor	Omron		GlobalSCREEN	Ordinal Technologies
CitectSCADA	Schneider Electric		IGSS	7-Technologies
AutomationX	AutomationX		AGP 3000	Pro-face
Genesis32	Iconics		Wizcon	ELUTIONS
MicroScada				
SattGraph 5000	ABB		TRACE MODE	AdAstra
ScadaNet				
ProccessNet	Matrikon		Factorylink	Siemens
EclipseScada	Eclipse Software		PcVue	ARC Informatique
	Santis			
Intellution	Engineering		Adroit	Adroit Technologies
			Advantech	
P-CIM	AFCON		Studio	Advantech

Πίνακας λογισμικών SCADA

Στην τωρινή εποχή, όλο και περισσότερα λογισμικά SCADA εμφανίζονται από διάφορους κατασκευαστές (όπως το *TopKari Vision* της *Areal*) με σκοπό να καλυφθούν οι απαιτήσεις της σύγχρονης βιομηχανίας. Στην παρούσα πτυχιακή εργασία θα ασχοληθούμε εκτενέστερα με το σχεδιασμό συστημάτων SCADA στο περιβάλλον **SIMATIC WinCC** της εταιρίας **Siemens**.

2.3.7.1 Χαρακτηριστικά λογισμικού SCADA

Το λογισμικό SCADA μπορεί να διαιρεθεί σε **δύο τύπους**, σε **λογισμικό κλειστού κώδικα** και σε **λογισμικό ανοικτού κώδικα**. Οι επιχειρήσεις, για λόγους ιδιωτικότητας και ασφάλειας, αναπτύσσουν συνήθως **ιδιόκτητο λογισμικό** με σκοπό την αποκλειστική επικοινωνία με το υλικό τους. Τα συστήματα αυτά πωλούνται ως βασικές «λύσεις» συστημάτων SCADA. Το βασικό όμως πρόβλημα με το συγκεκριμένο τύπο συστήματος βρίσκεται στο γεγονός της συντριπτικής εμπιστοσύνης στον προμηθευτή του συστήματος. Στην αντίπερα όχθη, τα ανοικτά συστήματα λογισμικού έχουν κερδίσει τη δημοτικότητα λόγω της διαλειτουργικότητας τους και της ευκολίας στην παραμετροποίηση που φέρνουν στο σύστημα. Η διαλειτουργικότητα είναι η δυνατότητα να αναμιχθεί ο εξοπλισμός των διαφορετικών κατασκευαστών στο ίδιο σύστημα. Αθροίζοντας τα κύρια χαρακτηριστικά που πρέπει να έχει ένα λογισμικό SCADA, μπορούμε να αναφέρουμε τα εξής:

- ✓ Διεπιφάνεια χρήστη
- ✓ Γραφικό περιβάλλον
- ✓ Συναγερμοί και ειδοποιήσεις για τον χειριστή
- ✓ Εξελιξιμότητα
- ✓ Πρόσβαση στα δεδομένα
- ✓ Ταυτόχρονη σύνδεση με βάσεις δεδομένων
- ✓ Ανοχή ελαττωμάτων
- ✓ Διασύνδεση Client/server μεταξύ των περιφερειακών συσκευών

2.4 Πρωτόκολλα επικοινωνίας

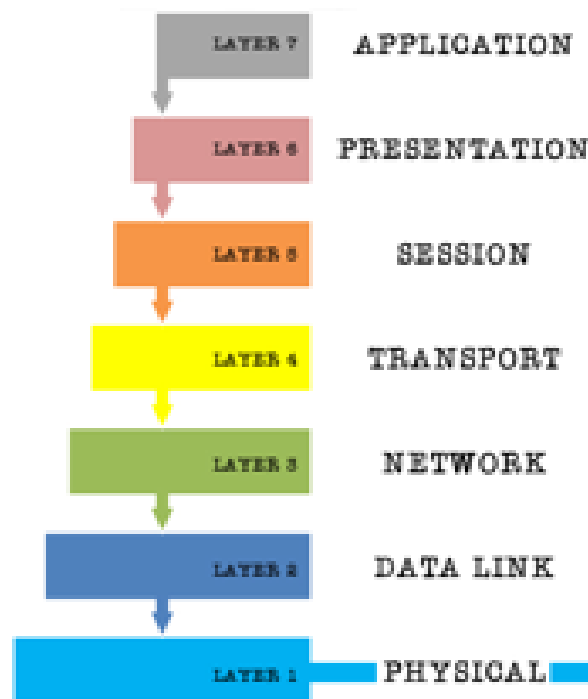
Όπως είδαμε και στις παραπάνω αναφορές μας σχετικά με την αρχιτεκτονική και την δομή ενός συστήματος SCADA, τα RTU's συμμετέχουν κατά κόρον στην τηλεμετρία ενός συστήματος. Είναι εργοστασιακά προγραμματισμένα για να δέχονται εντολές για την ορθή λειτουργία των σημείων ελέγχου της βιομηχανικής μονάδος, να αντλούν δεδομένα μέσω της αναλογικής εξόδου αυτών και να ανταποκρίνονται στα αιτήματα λειτουργείας τους. Παράλληλα, είναι υπεύθυνα για την αποστολή μετρήσεων και εντολών στις βάσεις δεδομένων και στον κεντρικό σταθμό ελέγχου του συστήματος SCADA αντίστοιχα. Για να διασφαλιστεί η **ασφάλεια**, η **ορθότητα** και κυρίως η **αύξηση της ταχύτητας αποστολής** των δεδομένων αυτών, από και προς το βιομηχανικό δίκτυο, απαιτείται η κρυπτογράφηση των πακέτων επικοινωνίας με βιομηχανικά πρωτόκολλα, ικανά να παρέχουν στην βιομηχανική μονάδα μια αξιόπιστη μεταφορά των δεδομένων στους προορισμούς, χωρίς σφάλματα, κάτω από οποιαδήποτε κατάσταση.

2.4.1 Αρχή λειτουργίας πρωτοκόλλων επικοινωνίας

Κάθε πρωτόκολλο μπορεί να διαχωριστεί σε **δύο βασικά μέρη**. Το πρώτο μέρος αποτελεί το **κύριο πρωτόκολλο** (*master protocol*), όπου περιέχει δεδομένα σχετικά με την εγκαθίδρυση επικοινωνίας με τον κεντρικό σταθμό ελέγχου του συστήματος. Αντιθέτως, το δεύτερο μέρος, **το πρωτόκολλο RTU**, περιέχει δεδομένα για την εγκατάσταση επικοινωνίας για το RTU. Όπως και οι περισσότερες αρχιτεκτονικές δικτύου, έτσι και ένας μεγάλος αριθμός από πρωτόκολλα επικοινωνίας βασίζονται στο **Διεθνές Πρότυπο OSI των 7 επιπέδων**. Με αυτόν τον τρόπο, το δίκτυο επικοινωνίας γίνεται άμεσα επεκτάσιμο και παραμετροποιήσιμο, δίνοντας την δυνατότητα σύνδεσης και ανταλλαγής δεδομένων με διάφορες συσκευές ελέγχου.

Παρατηρώντας το πρότυπο, βλέπουμε ότι τα τέσσερα κατώτερα στρώματα καλύπτουν την φυσική ένωση, τις συσκευές δικτύωσης, τα πρωτόκολλα επικοινωνίας και την μεταφορά των δεδομένων στα συνδεδεμένα δίκτυα αντίστοιχα. **Το πρωτόκολλο TCP/IP** (*Transport Control protocol / Internet Protocol*) όπου τα βιομηχανικά εποπτικά συστήματα βασίζονται, αφορά ένα σύνολο κανόνων που χρησιμοποιούνται κυρίως στο **τρίτο επίπεδο OSI** (επίπεδο δικτύου). Τα επίπεδα 5 και 6, συνεδρίας και παρουσίασης αντίστοιχα, αναλαμβάνουν την περάτωση της συνόδου επικοινωνίας. Τέλος, το επίπεδο 7 (επίπεδο εφαρμογής), πρόκειται

για το επίπεδο επικοινωνίας μεταξύ του κεντρικού σταθμού ελέγχου και των περιφερειακών συσκευών ελέγχου (RTU, PLC).



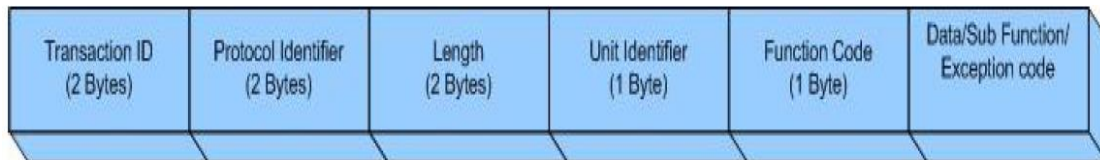
Το πρότυπο OSI των επτά επιπέδων

Όπως αναφέρθηκε και παραπάνω, στις σύγχρονες βιομηχανικές μονάδες σήμερα, τα κυριότερα βιομηχανικά πρωτόκολλα επικοινωνίας που συναντούμε είναι τα πρωτόκολλα **Modbus**, **Profibus**, **IEC 60870** καθώς και το πρωτόκολλο **DNP3**. Επίσης, τα πρωτόκολλα **ASCII**, **RP – 570**, **Hart** και **Conitel** εμφανίζονται εξίσου σε εποπτικά συστήματα, με μικρότερη όμως κλίμακα. Παρακάτω, θα δούμε μία αναλυτική παρουσίαση των πρωτοκόλλων αυτών.

2.4.1.1 Το πρωτόκολλο Modbus

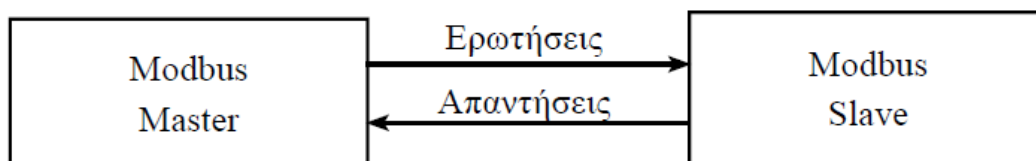
Το **πρωτόκολλο Modbus** είναι ένα ευρέως διαδεδομένο πρωτόκολλο για την αποστολή και λήψη δεδομένων κυρίως σε εφαρμογές αυτοματισμού και ευφυές συσκευές (*IED's*). Πρόκειται για ένα de facto πρότυπο, ανοικτού κώδικα. Αναπτύχθηκε αρχικά από την Εταιρεία Modicon στα τέλη της δεκαετίας του 1970 ως σειριακό πρωτόκολλο για την επικοινωνία με PLC και RTU. Πολύ σύντομα όμως καθιερώθηκε ως το πλέον χρησιμοποιούμενο πρότυπο σε περιβάλλοντα βιομηχανικής παραγωγής λόγω των παρακάτω πλεονεκτημάτων του:

- Είναι απλό στην χρήση του και σχετικά εύκολο στην υλοποίησή του
- Πρόκειται για ένα πρωτόκολλο ανοικτού κώδικα, όπου μπορεί να υποστηρίξει μεγάλη γκάμα συσκευών
- Αναπτύχθηκε ειδικά για βιομηχανικές εφαρμογές



Πρότυπο του πρωτοκόλλου Modbus

Το πρωτόκολλο Modbus αποτελεί ένα **πρωτόκολλο επιπέδου εφαρμογής ανταλλαγής μηνυμάτων** που τοποθετείται στο **επίπεδο 7 του μοντέλου OSI** και παρέχει client / server επικοινωνία μεταξύ συσκευών που συνδέονται με διαφορετικούς τύπους μέσων σύνδεσης και δικτύων πρωτόκολλο. Επιτρέπει την ανταλλαγή διακριτών αναλογικών σημάτων μεταξύ συσκευών, κάνοντας χρήση των στάνταρ **RS - 232** και **RS - 422/485**. Κάποιες νεότερες εκδόσεις του πρωτοκόλλου μπορούν να υποστηρίξουν αποστολές ακόμα και μέσω **Ethernet** ή **TCP/IP**, στέλνοντας πακέτα 1bit ή 16bit. Το Modbus ορίζει δύο τύπους συσκευών οι οποίες επικοινωνούν: τον **Modbus Master** και τον **Modbus Slave**. Ο Modbus Master είναι η συσκευή η οποία στέλνει μηνύματα στον Modbus Slave και ο Modbus Slave απαντά στα μηνύματα τα οποία δέχεται. Σε περίπτωση σφάλματος αποστολής του μηνύματος προς τον Modbus Master, στέλνεται μήνυμα λάθος από τον Modbus Slave, ζητώντας παράλληλα την επαναποστολή του μηνύματος. Ο Modbus Master είναι συνήθως η συσκευή η οποία συλλέγει τα δεδομένα από πολλούς περιφερειακούς σταθμούς μετρήσεων και αυτοματισμού. Ένας Modbus Master μπορεί να επικοινωνεί μέχρι και με 247 Modbus Slaves οι οποίοι είναι συνδεδεμένοι στο ίδιο δίκτυο και έχει ο καθένας την δικιά του μοναδική διεύθυνση επικοινωνίας ή αναγνωριστικό αριθμό σταθμού.



Υπάρχουν διάφορες παραλλαγές του πρωτοκόλλου Modbus για την μετάδοση σε διαφορετικά είδη δικτύων, οι κυριότερες από τις οποίες είναι οι παρακάτω:

- **Modbus RTU** για σειριακή επικοινωνία
- **Modbus ASCII** για σειριακή επικοινωνία
- **Modbus TCP** για επικοινωνία με το πρωτόκολλο TCP.
- **Modbus X**, σαν επέκταση του από δικτύου Modbus, κυρίως για σύγχρονα συστήματα SCADA

Αναλόγως του τύπου πρωτοκόλλου ορίζονται και διαφορετικές δομές πακέτων για την αποστολή και μετάδοση μηνυμάτων.

2.4.1.2 Το πρωτόκολλο Profibus

Το **πρωτόκολλο Profibus (PROcess Field BUS)** είναι ένα διεθνές και ευρέως αποδεκτό πρωτόκολλο, το οποίο συναντάται συνήθως σε εποπτικά συστήματα βιομηχανιών μεγάλου μεγέθους. Όπως και το πρωτόκολλο Modbus, πρόκειται για ένα πρωτόκολλο ανοικτού κώδικα, εμπίπτοντας και αυτό, όπως και τα πρωτόκολλα που αναφέρθηκαν παραπάνω, στο μοντέλο 7 επιπέδων του OSI. Με τον τρόπο αυτό εξασφαλίζεται η εύκολη και αποδοτική επικοινωνία περιφερειακών συσκευών από διαφορετικούς κατασκευαστές. Οριστικοποιήθηκε σαν πρωτόκολλο επικοινωνίας στη Γερμανία ως το **DIN-19245** και αργότερα υπό το European National Standard **EN 50170**. Η ανάπτυξη του ξεκίνησε από την BMFT (Γερμανικό Υπουργείο Έρευνας και Τεχνολογίας) το 1989, σε συνεργασία με έναν μεγάλο αριθμό αυτοκινητοβιομηχανιών, και έπειτα χρησιμοποιήθηκε κυρίως από την εταιρία Siemens ως το βασικό πρωτόκολλο επικοινωνιών. Βασίζεται στο πρωτόκολλο IEC – 61158.

Από τεχνικής απόψεως, το πρωτόκολλο Profibus υποστηρίζει μέχρι και 127 κόμβους με μέγιστη απόσταση από άκρο σε άκρο τα 27km κάνοντας χρήση κυρίως οπτικών ινών και αναμεταδοτών. Τα πακέτα μηνυμάτων έχουν μήκος 244 bytes/κόμβο συν 12 bytes overhead, συνολικά δηλαδή 256 bytes και στηρίζονται στην τεχνική **polling-token passing**. Το Profibus βασίζεται και αυτό στην αρχιτεκτονική **Master-Active/Slave-Passive** ως εξής:

- ❖ Οι συσκευές Master ελέγχουν το bus όταν και όποτε τους παραχωρείται το δικαίωμα. Σε αυτήν την κατάσταση μεταφέρουν τα μηνύματα χωρίς να προηγηθεί απομακρυσμένο αίτημα.

- ❖ Οι συσκευές Slave είναι περιφερειακές συσκευές όπως λ.χ. αισθητήρες, μετατροπείς κλπ. Επιτρέπεται μόνο να αναγνωρίσουν τη λήψη μηνυμάτων ή να αποστείλουν μήνυμα κατόπιν αιτήματος της Master.

Μπορούμε να χωρίσουμε το πρωτόκολλο Profibus σε 3 βασικές κατηγορίες:

1. **Πρωτόκολλο Profibus DP** (Distributed Peripheral) το οποίο επιτρέπει τη χρήση πολλαπλών Master συσκευών με τις συσκευές Slave να αντιστοιχούν σε καθένα από αυτές. Αυτό σημαίνει πως παρότι όλες οι συσκευές Master έχουν πρόσβαση σε όλες τις Slave, μόνο η κύρια Master (στην οποία αντιστοιχεί η εκάστοτε Slave) έχει τη δυνατότητα να εγγράψει δεδομένα σε αυτή.
2. **Πρωτόκολλο Profibus FMS** (Fielbus Message Specification) το οποίο κάνει χρήση ενός peer – to – peer format και επιτρέπει την επικοινωνία μεταξύ των Master με γενικό σύνολο 127 συσκευών πάνω στο bus. Σημείωση πως στο FMS είναι δυνατόν και οι 127 συσκευές να είναι Master.
3. **Πρωτόκολλο Profibus PA**, το οποίο αποτελεί νεότερη γενιά του DP στην οποία τα επίπεδα της τάσης και έντασης του ηλεκτρικού ρεύματος είναι χαμηλότερα ώστε να ικανοποιούνται οι απαιτήσεις ενδογενούς ασφάλειας βιομηχανικών εγκαταστάσεων (Class I, division II).

2.4.1.3 Το πρωτόκολλο IEC 60870

Το **πρωτόκολλο IEC 60870** δημιουργήθηκε τις αρχές του 1990 από την Διεθνή Ηλεκτρονική Επιτροπή (**International Electrotechnical Commission - IEC**) βρίσκοντας χρήση αρχικά σε εποπτικά συστήματα μετάδοσης και διανομής ηλεκτρικής ενέργειας. Χωρίζεται σε **έξι βασικές κατηγορίες**, καθορίζοντας τις γενικές πληροφορίες που σχετίζονται με το πρωτόκολλο, τις συνθήκες λειτουργείας του και τις απαιτήσεις του για την μεταφορά των δεδομένων στο δίκτυο. Στα σύγχρονα βιομηχανικά συστήματα εποπτικού ελέγχου SCADA, ιδιαίτερη χρήση παρουσιάζει η πέμπτη κατηγορία του πρωτοκόλλου (**IEC 60870 – 5**). Η κατηγορία αυτή είναι δυνατό να μας προσφέρει ένα απλό και συνάμα γρήγορο προφίλ επικοινωνίας για την αποστολή δεδομένων ακόμα και σε αργά ασύγχρονα μέσα μετάδοσης. Κάνοντας ακόμα πιο εξειδικευμένη την χρήση του πρωτοκόλλου IEC 60870 – 5 , μπορούμε να το διαχωρίσουμε σε έξι εκδόσεις:

- **IEC 60870-5-1**, όπου χρησιμοποιείται για τις υπηρεσίες αποστολής δεδομένων
- **IEC 60870-5-2**, όπου χρησιμοποιείται για τις υπηρεσίες μετάδοσης δεδομένων
- **IEC 60870-5-3**, όπου χρησιμοποιείται για την γενική δομή των δεδομένων των εφαρμογών
- **IEC 60870-5-4**, όπου χρησιμοποιείται για την κωδικοποίηση των στοιχείων των εφαρμογών
- **IEC 60870-5-5**, όπου χρησιμοποιείται για την μετάδοση των βασικών λειτουργιών της εφαρμογής
- **IEC 60870-5-6**, όπου χρησιμοποιείται παρέχοντας επιπλέον ασφάλεια για την έκδοση IEC 60870-5-5

Σαν αναβάθμιση των παραπάνω εκδόσεων, η IEC στις αρχές του 2000, δημιούργησε τα παρακάτω πρότυπα, καλύπτοντας έτσι αρκετές ατέλειες των εκδόσεων που υπήρχαν:

- Πρότυπο **IEC 60870-5-101**, για την μεταφορά δεδομένων ιδίως σε εποπτικά συστήματα ελέγχου SCADA
- Πρότυπο **IEC 60870-5-102**, για την μεταφορά δεδομένων κυρίως σε συστήματα διανομής ηλεκτρικής ενέργειας
- Πρότυπο **IEC 60870-5-103**, για την διασύνδεση του εξοπλισμού προστασίας του συστήματος
- Πρότυπο **IEC 60870-5-104**, για την ασφαλέστερη μεταφορά δεδομένων σε συνδυασμό με το πρότυπο IEC 60870-5-101

Στις βιομηχανικές μονάδες του σήμερα, το πλέον διαδεδομένο πρότυπο όπου χρησιμοποιείται κατά κόρον από τα συστήματα εποπτικού ελέγχου SCADA είναι το **IEC 60870 – 5 – 101**. Πρόκειται για ένα διεθνές πρότυπο πρωτοκόλλου επικοινωνιών, πλήρως συμβατό με τα περισσότερα βιομηχανικά δίκτυα, λόγω τις ασύγχρονης σύνδεσης χρησιμοποιεί με τις περιφερειακές συσκευές του συστήματος. Παρέχεται για τον έλεγχο του τηλεχειρισμού των συστημάτων

μετάδοσης

δεδομένων.

Η δομή του πρωτοκόλλου αυτού απαρτίζεται από **4 βασικά σκέλη: ένα bit έναρξης του πακέτου, ένα bit λήξης του, ένα bit ισοτιμίας και συγχρονισμού** με τον δέκτη καθώς και **οκτώ bits με τα δεδομένα προς μετάδοση.**

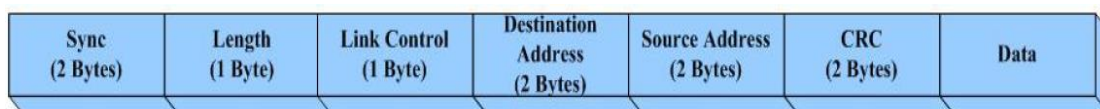
Παρόλα αυτά, το πρότυπο του πρωτοκόλλου IEC 60870 – 5 – 101, σύμφωνα με το είδος της εργασίας που έχει οριστεί, μπορεί να χωριστεί με την σειρά του σε **τρία διαφορετικά είδη**: Στο είδος **single character**, όπου χρησιμοποιείται για αναγνωρίσεις και επιβεβαιώσεις πακέτων δεδομένων, στο είδος **fixed length**, βρίσκοντας χρήση στην αποστολή εντολών προς

το σύστημα και στο είδος **variable length** όπου χρησιμοποιείται για την αποστολή δεδομένων.

Όπως αναφέραμε και παραπάνω, το IEC 60870 – 5 – 101 πρόκειται για ένα ευρέως διαδεδομένο πρωτόκολλο στα σύγχρονα συστήματα βιομηχανικού και εποπτικού ελέγχου SCADA. Λόγω της δομής και του τρόπου λειτουργίας μετάδοσης των πακέτων δεδομένων, μπορεί να υποστηρίξει μονόπλευρες (*master only*) και αμφίπλευρες (*master/slave*) μεταφορές πακέτων δεδομένων. Επίσης, τα διάφορα τμήματα του συστήματος καθώς και τα δεδομένα προς αποστολή διαφοροποιούνται μεταξύ τους με την παρουσία διαφορετικών διευθύνσεων όπου χρησιμοποιεί το πρωτόκολλο, επιτυγχάνοντας έτσι καλύτερη οργάνωση και ταχύτερη αποστολή των εντολών και των δεδομένων του συστήματος. Παράλληλα, η διευκόλυνση στην ταξινόμηση των πακέτων δεδομένων με την χρήση ομάδων αλλά και προτεραιοτήτων, παρέχει στον χρήστη την δυνατότητα επιλογής των δεδομένων που επιθυμεί να αποστείλει στο σύστημα.

2.4.1.4 Το πρωτόκολλο DNP3

Το **πρωτόκολλο DNP3** (*Distributed Network Protocol*) πρόκειται για ένα σύνολο πρωτοκόλλων επικοινωνίας που λαμβάνει χώρα μεταξύ των συσκευών ελέγχου σε ένα εποπτικό βιομηχανικό σύστημα. Αναπτύχθηκε από την εταιρία GE-Harris Canada (γνωστή και ως Westronic, Inc.) το 1990 σαν μία ορθότερη έκδοση του πρωτοκόλλου IEC 60870-5 και δόθηκε προς χρήση στο βιομηχανικό κοινό το 1993. Στις μέρες μας διαδραματίζει βασικό ρόλο στα βιομηχανικά συστήματα SCADA καθώς αποτελεί πρωτόκολλο επικοινωνίας μεταξύ των Κεντρικών Σταθμών Ελέγχου, των RTU's και των ευφυών ηλεκτρονικών συσκευών (*Intelligent Electronic Devices - IED's*) που είναι συνδεδεμένα πάνω στο σύστημα. Από τεχνικής πλευράς, το πρωτόκολλο DNP3 ορίζεται κυρίως στο **επίπεδο ζεύξης δεδομένων** (2ο επίπεδο) του μοντέλου αναφοράς OSI. Παρέχει μηχανισμούς πολυπλεξίας και κατακερματισμού των δεδομένων που αποστέλλονται, ελέγχου σφαλμάτων και εγγύησης της ακεραιότητας των δεδομένων (*CRC Check*). Παράλληλα όμως ορίζεται και στα επίπεδα μεταφοράς και εφαρμογής (6ο και 7ο επίπεδο κατά OSI), καθορίζοντας έτσι διάφορες λειτουργίες και δεδομένα για την κατάλληλη επικοινωνία τους με το σύστημα SCADA.



Πρότυπο του πρωτοκόλλου DNP3

Σύμφωνα με το μοντέλο του πρωτοκόλλου DNP3 που παρατίθεται παραπάνω, κάθε πακέτο μπορεί να έχει μέγεθος έως και 16 bytes, αναλόγως με το μέγεθος των δεδομένων που περιέχει. Για να εκκινήσει η αποστολή των δεδομένων από τους τερματικούς σταθμούς (RTU's) προς τον Κεντρικό Εποπτικό Σταθμό (MTU), θα πρέπει να εδραιωθεί ένας διάυλος επικοινωνίας μεταξύ τους. Αυτό πραγματοποιείται με κατάλληλα πακέτα που αποστέλλονται από τα RTU's προς τον MTU και το αντίθετο (*send - receive messages*), στα πρότυπα της σχέσης master - slave. Η αποστολή των δεδομένων από τα RTU's του συστήματος SCADA προς τον MTU πραγματοποιείται με τον διαχωρισμό των δεδομένων σε **2 βασικές κατηγορίες: τις στατικές δομές (data objects) και τις δομές γεγονότων (event objects)**. Οι στατικές δομές περιλαμβάνουν πληροφορίες σχετικά με την τρέχουσα τιμή και κατάσταση της συσκευής. Σε αντίθεση, οι δομές γεγονότων παράγονται από μεταβολές στις τιμές των δεδομένων. Οι μεταβολές αυτές με την σειρά τους κατακερματίζονται σε 4 κλάσεις: την κλάση 0 όπου περιέχει όλα τα δεδομένα από τις στατικές δομές και τις κλάσεις 1 – 3 όπου περιέχουν τα δεδομένα από τις δομές γεγονότων που έχουν παραχθεί. Η ανάθεση των δεδομένων στην ανάλογη κλάση γίνεται είτε μέσω ειδικού αλγόριθμου είτε βάση προτεραιότητας των δεδομένων, από την υψηλότερη προς την χαμηλότερη. Όταν κάποιο από τα δεδομένα δεν αντιστοιχηθεί σε κάποια από τις κλάσεις 1 – 3, τότε αυτόματα ανατίθεται στην κλάση 0. Για να διασφαλιστεί η ακεραιότητα των δεδομένων, τοποθετούνται 2 bytes ελέγχου (CRC) κάθε 16 bytes των απεσταλλόμενων δεδομένων. Η ανάγνωση από τον MTU πραγματοποιείται με παρόμοιο τρόπο. Πιο συγκεκριμένα, ο κεντρικός σταθμός, λαμβάνοντας το απεσταλλόμενο πακέτο, πραγματοποιεί ανάγνωση είτε όλων των πακέτων ταυτόχρονα είτε ξεχωριστά, αναλόγως με την προτεραιότητα που τους έχει τεθεί. Παράλληλα, λόγω της δυνατότητας συγχρονισμού που παρέχει το πρωτόκολλο DNP3, μπορεί να υλοποιηθεί η αποστολή πακέτων ακόμα και από συσκευές όπου λειτουργούν με ασύγχρονο τρόπο επικοινωνίας.

2.4.1.5 Το πρωτόκολλο RP – 570

Το **πρωτόκολλο RP – 570** αποτελεί ένα πρωτόκολλο επικοινωνίας μεταξύ μίας απομακρυσμένης τερματικής συσκευής (RTU) και του λογισμικού του εποπτικού συστήματος SCADA, όπου βρίσκεται στον Κεντρικό Σταθμό (MTU). Αναπτύχθηκε από την εταιρία ABB στις αρχές τις δεκαετίας του '90 βασιζόμενο, όπως και το DNP3, στο πρωτόκολλο IEC – 60870. Σύμφωνα με την δομή του πρωτοκόλλου RP – 570, παρατηρούμε ότι χρησιμοποιεί παρόμοιο

μηχανισμό με το IEC – 60870 – 5 – 101 για την μετάδοση των πακέτων δεδομένων. Αποτελείται από ένα bit αρχής και ένα bit όπου υποδεικνύει το τέλος του πακέτου, ένα bit ισοτιμίας και συγχρονισμού και 8 bits δεδομένων.

2.4.1.6 Το πρωτόκολλο Hart

Το **πρωτόκολλο HART** (*Highway Addressable Remote Transducer*) είναι ένα πρώιμο πρωτόκολλο Fieldbus. Εισήχθη από την εταιρία Rosemount το 1986 ως ένα πρωτόκολλο ανοικτού τύπου και αργότερα έγινε ευρύτερα γνωστό από το ίδρυμα HART Communication Foundation (*HCF*). Το κυριότερο γνώρισμα που έχει είναι το ότι μπορεί να εγκατασταθεί σε όργανα τα οποία είναι σχεδιασμένα να συλλέγουν δεδομένα και πληροφορίες από αισθητήρες και ενεργοποιητές που λειτουργούν σε μία υβριδική κατάσταση (αναλογική – ψηφιακή) τύπου 4 – 20mA. Ακριβώς επειδή το πλήθος των εγκαταστάσεων βασισμένων στο 4-20mA είναι τεράστιο, πιστεύεται ότι το HART είναι από τα πιο δημοφιλή πρωτόκολλα βιομηχανικής δικτύωσης σήμερα.

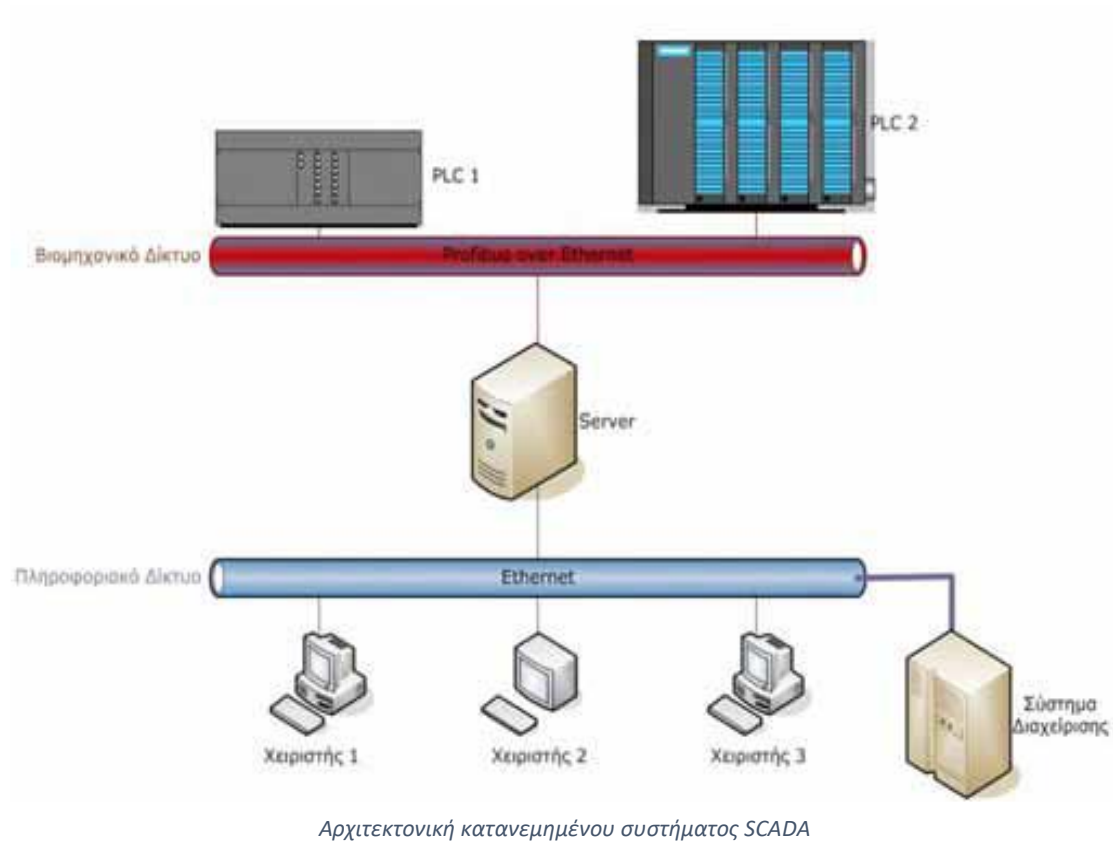
2.4.1.7 Το πρωτόκολλο Conitel

Το **πρωτόκολλο Conitel** πρόκειται για ένα πρωτόκολλο ασύγχρονης επικοινωνίας όπου βρίσκει συχνά χρήση στα συστήματα εποπτικού ελέγχου SCADA. Ακολουθώντας την βάση λειτουργίας των προαναφερθέντων πρωτοκόλλων, το Conitel αποτελείται από ένα bit εκκίνησης και ένα λήξης του πακέτου δεδομένων καθώς και 31 bits όπου αποθηκεύονται τα δεδομένα και οι εντολές προς μεταφορά. Χρησιμοποιείται από τα συστήματα SCADA είτε για την απευθείας επικοινωνία μίας περιφερειακής συσκευής με τον Κεντρικό Εποπτικό Σταθμό (*MTU*) είτε για την μετάδοση δεδομένων και εντολών όλων των συσκευών του συστήματος με τον MTU και αντιστρόφως. Για να διασφαλιστεί η ασφάλεια του βιομηχανικού δικτύου, το πρωτόκολλο Conitel ενθυλακώνει στο μεταδιδόμενο πακέτο ένα κώδικα μεγέθους 5 bits, σύμφωνα με τα πρότυπα του κυκλικού κώδικα κρυπτογράφησης Bose Chaudhuri. Όλα τα πακέτα όπου αποστέλλονται με το πρωτόκολλο Conitel, ελέγχονται αυστηρά από τον MTU. Αυτό σημαίνει ότι καμία απομακρυσμένη τερματική συσκευή δεν μπορεί να αποστείλει άμεσα πακέτα δεδομένων και εντολών προς μία άλλη απομακρυσμένη τερματική συσκευή, χωρίς αυτό να φιλτραριστεί από τον κεντρικό υπολογιστή του συστήματος. Ο κεντρικός υπολογιστής, ελέγχοντας τα 5 bits του κώδικα Bose Chaudhuri, προχωράει στην απόρριψη ή την αποστολή του μηνύματος προς τις τερματικές συσκευές. Το τερματικό όπου

θα δεκτεί το πακέτο θα αποστείλει ένα μήνυμα επιβεβαίωσης και ορθής λήψης του μεταδιδόμενου μηνύματος προς τον κεντρικό υπολογιστή. Εξαίρεση στον κανόνα αυτό αποτελούν τα πακέτα μηνυμάτων τύπου broadcast.

2.5 Σύγχρονα κατανεμημένα συστήματα SCADA

Στις μέρες μας, όπου οι ηλεκτρονικοί μικροϋπολογιστές έχουν κατακλύσει την αγορά, οι προγραμματιστές λογισμικού SCADA καλούνται να αναπτύξουν λογισμικά προγράμματα SCADA τα οποία θα παρέχουν την δυνατότητα να «τρέχουν» σε προσωπικούς ηλεκτρονικούς υπολογιστές (PC), θα είναι σχεδιασμένα με βάση την αρχιτεκτονική 32bit και σύντομα την 64bit των σύγχρονων λειτουργικών συστημάτων καθώς επίσης θα δύναται να υποστηρίζουν τα πιο διαδεδομένα πρωτόκολλα διασύνδεσης υπολογιστών (*Ethernet, Token Ring*) αλλά και περιφερειακών εποπτικών συσκευών (*Profibus, Modbus, DNP3*). Παράλληλα, τα σύγχρονα συστήματα SCADA απαιτείται να είναι **εξαιρετικά φιλικά ως προς τον χρήστη** αλλά και **να μην απαιτούν ιδιαίτερες γνώσεις προγραμματισμού**. Τέλος, πρέπει να είναι **πλήρως αντικειμενοστραφή** (*object-oriented*) σε επίπεδο δεδομένων (*data*) και λειτουργιών (*functions*), έτσι ώστε να περιέχουν όλους τους απαραίτητους οδηγούς (*drivers*) για επικοινωνία με τις εισόδους, τις εξόδους και τις μνήμες του PLC. Τα συστήματα SCADA υλοποιούνται με βάση την **κατανεμημένη αρχιτεκτονική** της οποίας η τοπολογία φαίνεται στο σχήμα παρακάτω



2.6 Σκοπός των συστημάτων SCADA

Ανακεφαλαιώνοντας τις παραπάνω παραγράφους, η λειτουργία των συστημάτων SCADA περιλαμβάνει την συλλογή των δεδομένων από τις διάφορες περιφερειακές συσκευές (*PLC's*, *IED's*, *RTU's*) που απαρτίζουν το εποπτικό σύστημα, την αποστολή τους στο κέντρο ελέγχου του συστήματος (*MTU*), την ανάλυση και επεξεργασία τους και τέλος την απεικόνισή τους σε οθόνες ελέγχου ή μιμικά διαγράμματα (*HMI*), όπου ελέγχονται από τους εκάστοτε χειριστές των μηχανημάτων, όλα αυτά σε πραγματικό χρόνο. Ο έλεγχος των δεδομένων μπορεί να πραγματοποιείται αυτόματα ή μετά από αίτημα του χειριστή. Το λογισμικό που χρησιμοποιούν τα συστήματα SCADA επιτρέπει την υλοποίηση εξαιρετικά πολύπλοκων λειτουργιών και υπολογισμών, χωρίς κανένα απολύτως κίνδυνο αλλοίωσης των αποτελεσμάτων ελέγχου που διεξάγουν οι μονάδες ελέγχου του συστήματος. Τα συστήματα εποπτικού ελέγχου δίνουν τη δυνατότητα στους χειριστές τους να εποπτεύουν και να παρατηρούν διεργασίες οι οποίες τοπολογικά μπορεί να βρίσκονται διάσπαρτες σε μια μεγάλη γεωγραφική περιοχή. Σε αυτό το σημείο γίνεται ορατό το μεγαλύτερο ίσως

πλεονέκτημα των συστημάτων SCADA. Δηλαδή, αντί να στέλνεται ανθρώπινο δυναμικό για ελέγχους και μετρήσεις σε διάσπαρτα σημεία, ο έλεγχος ολόκληρου του συστήματος πραγματοποιείται από μια τοποθεσία σε ελάχιστο χρονικό διάστημα

2.7 Στόχος των συστημάτων SCADA

Η εποπτική διαδικασία για μία σύγχρονη παραγωγική μονάδα αποτελεί όπως προαναφέραμε βασικό κεφάλαιο. **Η ομαλή και απροβλημάτιστη λειτουργία της βιομηχανικής μονάδος** αποτελεί βασικό στόχο όχι μόνο των ανθρώπων που συμμετέχουν σε αυτή αλλά όλου του συνόλου της βιομηχανίας. Έτσι, κατηγοριοποιώντας τους στόχους που δύναται να υλοποιηθούν με την χρήση ενός συστήματος βιομηχανικού ελέγχου SCADA, μπορούμε να αναφέρουμε τα παρακάτω:

- ❖ Η **διασφάλιση της ποιότητας** του παραγόμενου προϊόντος.
- ❖ Η **ατέρμονη παρακολούθηση** του συστήματος
- ❖ Η **μεγιστοποίηση της παραγωγής** με χρήση των ελάχιστων δυνατών (*ενεργειακών*) πόρων.
- ❖ Η **μείωση των επιπέδων των λειτουργικών αποθεμάτων** μέσω της αυτοματοποίησης ή μέσω της λειτουργίας του συστήματος από μία απλή κεντρική εγκατάσταση
- ❖ Η **παροχή πληροφοριών** για την απόδοση του συστήματος και εφαρμογή αποτελεσματικών διαδικασιών διαχείρισης για το σύστημα
- ❖ Η **παροχή ενός συστήματος ελέγχου** που θα επιτρέπει να επιτυγχάνονται οι αντικειμενικοί σκοποί
- ❖ Η **βέλτιστη διαχείριση** του εξοπλισμού, των υλικών και της ενέργειας της εγκατάστασης.
- ❖ Η **ασφάλεια του εξοπλισμού** και του προσωπικού παρακολούθησης της διεργασίας μέσω συστήματος έκτακτης ειδοποίησης που θα επιτρέπει την διάγνωση των σφαλμάτων από ένα κεντρικό σημείο, ώστε να επιτυγχάνονται οι διαδικασίες επίλυσης των προβλημάτων από εξειδικευμένο προσωπικό και να αποφεύγονται περιστατικά που μπορεί να είναι επιζήμια για το περιβάλλον.

Είναι σημαντικό όλοι οι παραπάνω στόχοι να **επιτυγχάνονται παράλληλα**. Δεν είναι δυνατό να επιδιώκουμε την μεγιστοποίηση της παραγωγικής διαδικασίας χωρίς πρώτα να έχουμε εξασφαλίσει την βέλτιστη διαχείριση των διαθέσιμων πόρων και υλικών. Επιπλέον ο έλεγχος των διεργασιών θα πρέπει να είναι εξαιρετικά γρήγορος, ώστε να έχουμε επίγνωση της κατάστασης των επιτηρούμενων μεγεθών σε πραγματικά χρόνο (*real time*). Αυτό γίνεται αμέσως εμφανές αν αναφερθούμε στο θέμα της ασφάλειας. Η επέμβαση στις διεργασίες σε περίπτωση κινδύνου θα πρέπει να είναι άμεση και αποτελεσματική ώστε να αποφεύγουμε μερική ή ολική καταστροφή του εξοπλισμού, ακόμα και ανθρώπινες απώλειες.



Εποπτικός Έλεγχος με σύστημα SCADA

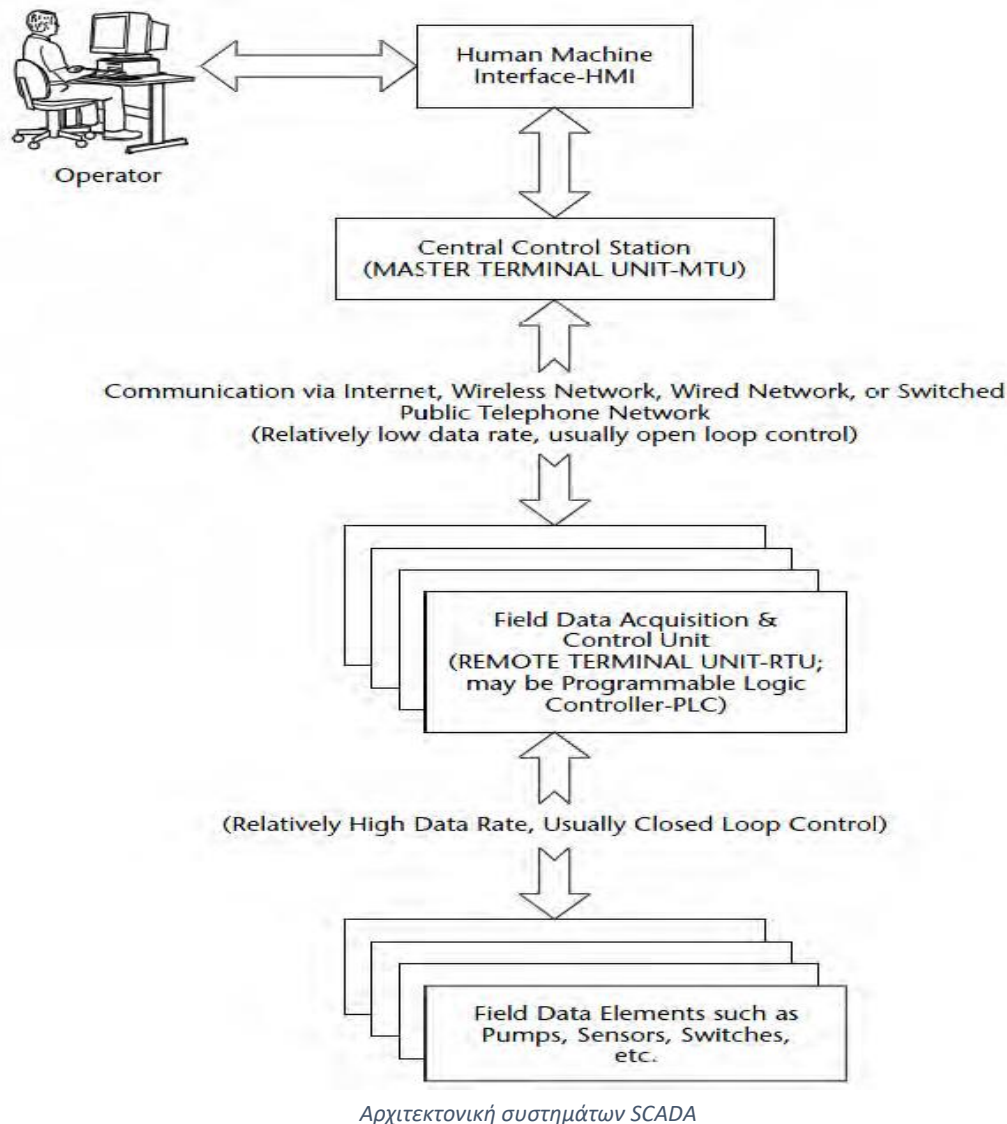
2.8 Κύριες λειτουργίες ενός συστήματος SCADA

Επιγραμματικά, μπορούμε να αναφέρουμε τις ενέργειες που έχουμε την δυνατότητα να υλοποιούμε μέσω των συστημάτων αυτομάτου ελέγχου SCADA. Συγκεκριμένα:

- Συλλογή δεδομένων από τα PLC's, τα IED's και τα RTU's. Όλα τα επιθυμητά σήματα μεταδίδονται προς το σύστημα SCADA μέσω του δικτύου βιομηχανικού αυτοματισμού.
- Αποθήκευση των πληροφοριών σε βάση δεδομένων και αναπαράσταση τους μέσω γραφημάτων. Οι επιλεγμένες πληροφορίες αναπαρίστανται είτε αυτούσιες είτε έπειτα από κατάλληλη επεξεργασία.
- Ανάλυση δεδομένων και ειδοποίηση του προσωπικού σε περιπτώσεις σφάλματος. Όταν τα δεδομένα πάρουν μη κανονικές τιμές, το σύστημα SCADA ειδοποιεί με

οπτική ή ακουστική σήμανση τους χειριστές, ώστε να αποφευχθούν δυσάρεστες επιπτώσεις.

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



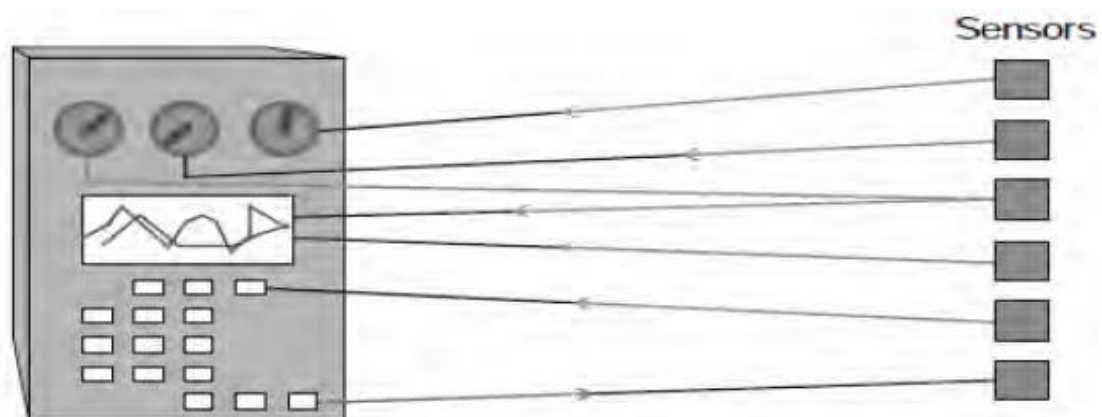
2.9 Εξέλιξη των συστημάτων SCADA

Τα συστήματα SCADA χρονολογούνται όσο και τα πρώτα συστήματα ελέγχου. Αρχικά αποτελούνταν από διάφορους μετρητές, συσκευές ένδειξης και καταγραφικά. Εντολές ελέγχου μπορούσαν να εκτελεστούν μόνο χειροκίνητα από το χρήστη. Ορισμένες εγκαταστάσεις χρησιμοποιούν ακόμα τέτοια συστήματα. Ωστόσο στα σύγχρονα βιομηχανικά περιβάλλοντα η **τηλεμετρία** είναι αυτή που χρησιμοποιείται ώστε να δικτυωθούν οι διάφορες συσκευές μεταξύ τους ειδικά όταν αυτές βρίσκονται σε πολύ μεγάλες αποστάσεις μεταξύ τους (από μερικά μέτρα μέχρι χιλιάδες χιλιόμετρα). Υπενθυμίζουμε ότι η τηλεμετρία χρησιμοποιείται για να γίνεται διάδοση δεδομένων όπως εντολές, προγράμματα, δεδομένα και φυσικές μεταβλητές.

Τα συστήματα SCADA λοιπόν συνδυάζουν μαζί την τηλεμετρία και τη συλλογή δεδομένων. Η αρχιτεκτονική SCADA ξεκινάει με τη **συλλογή δεδομένων**, τη **μεταφορά τους** σε ένα κεντρικό σταθμό όπου γίνεται η απαραίτητη ανάλυση και επεξεργασία, την **απεικόνιση** σημαντικών πληροφοριών και ενδείξεων σε μία ή περισσότερες οθόνες και τέλος την **εκτέλεση και αποστολή εντολών ελέγχου** πίσω στο πραγματικό περιβάλλον. Η εξέλιξη όμως των συστημάτων αυτών, για να μπορέσουν να φτάσουν στην μορφή που απαντώνται μέχρι σήμερα, πέρασε από αρκετά στάδια. Συγκεκριμένα, τα συστήματα βιομηχανικού ελέγχου SCADA αναπτύχθηκαν σε **4 βασικές γενιές**:

1. Τα μονολιθικά συστήματα SCADA
2. Τα κατανεμημένα συστήματα SCADA
3. Τα δικτυωμένα συστήματα SCADA
4. Internet Of Things (IOT)

Παρακάτω θα γίνει μία αναλυτική παρουσίαση για κάθε μία από τις προαναφερθείσες γενιές των συστημάτων

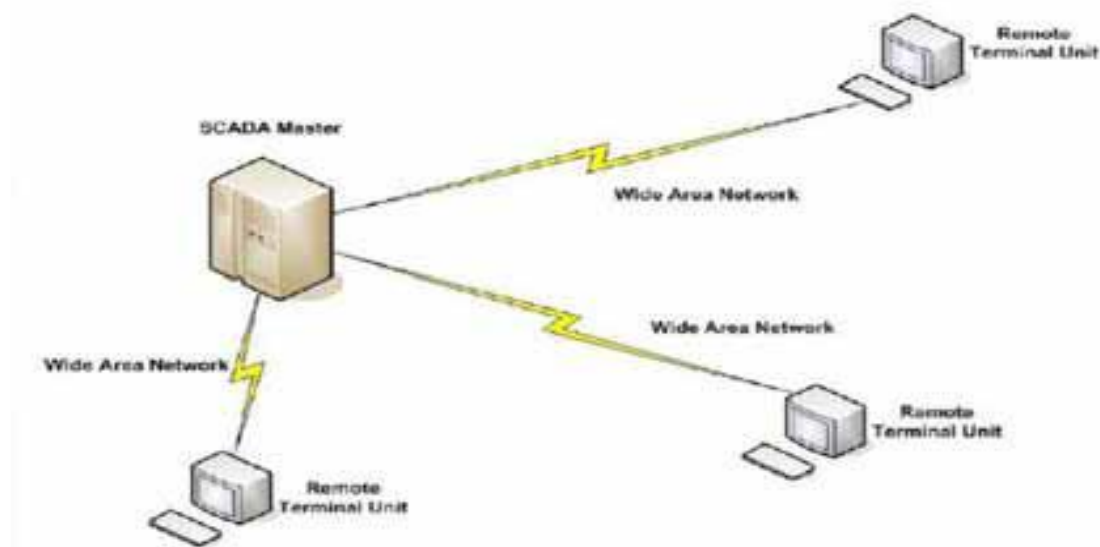


Καταγραφικά συστήματα 1^{ης} γενιάς

2.9.1 1^η Γενιά – Τα μονολιθικά συστήματα

Τα συστήματα πρώτης γενιάς είχαν ως κύριο στοιχείο έναν **Mainframe υπολογιστή** μεγάλης ισχύος, όπως ο **PDP-11** της Digital Equipment Corporation, μέσα από τον οποίο γινόταν οι απαραίτητες λειτουργίες ανάλυσης και συλλογής δεδομένων. Ήταν εντελώς **αυτόνομα συστήματα** με ουσιαστικά καμία συνδεσιμότητα με άλλα συστήματα. Τα δίκτυα ευρείας περιοχής (**WAN's**) που εφαρμόστηκαν για να επικοινωνήσουν με τις μονάδες απομακρυσμένων τερματικών (**RTU's**) σχεδιάστηκαν με έναν ενιαίο και μοναδικό σκοπό, αυτόν της επικοινωνίας με RTU's στον τομέα.

Ωστόσο όμως, είχε προβλεφθεί η δυνατότητα εγκατάστασης εφεδρικού συστήματος σε περίπτωση αστοχίας ή βλάβης του κύριου υπολογιστή.



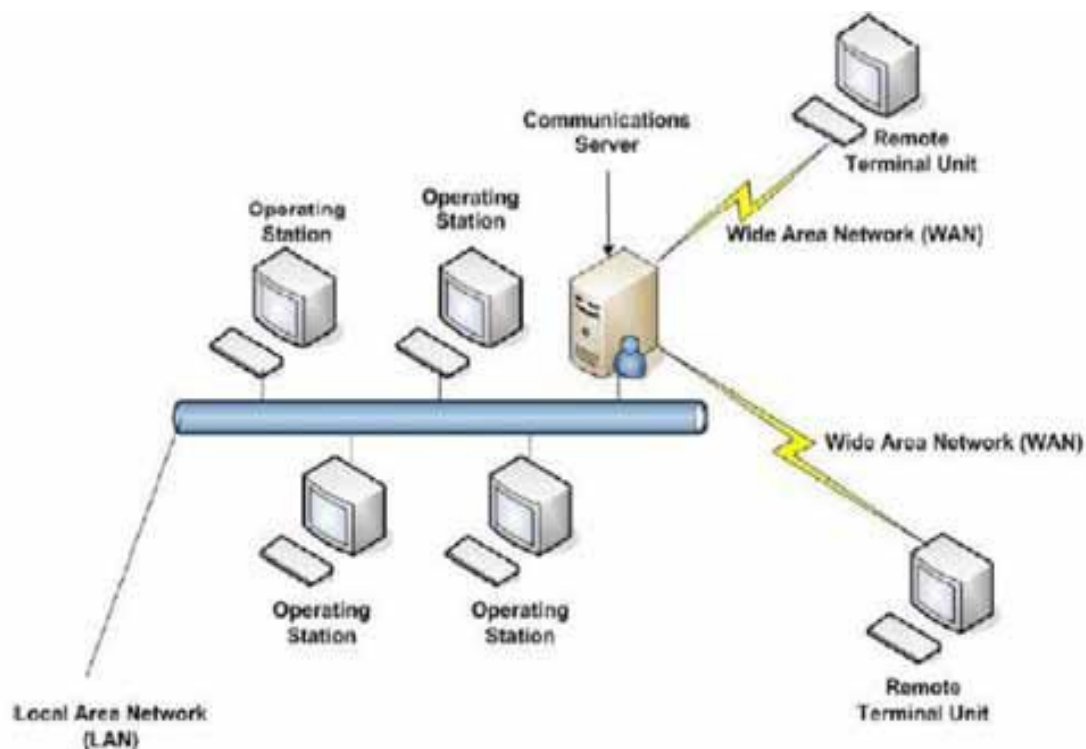
Υπόδειγμα μονολιθικού συστήματος SCADA

Τα πρωτόκολλα επικοινωνίας σχετικά με τα δίκτυα SCADA αναπτύχθηκαν από τους προμηθευτές του εξοπλισμού RTU και ήταν κυρίως **ιδιωτικά**. Επιπλέον, τα πρωτόκολλα αυτά μπορούν να χαρακτηριστούν ως «**αδύναμα**», μη ενισχύοντας ουσιαστικά καμία λειτουργία πέρα από εκείνη την απαραίτητη, δηλαδή την ανίχνευση και τον έλεγχο των σημείων μέσα στη μακρινή συσκευή. Παράλληλα, δεν ήταν γενικά εφικτό να αναμιχθούν και άλλοι τύποι διάδοσης δεδομένων. Η συνδεσιμότητα στον κύριο σταθμό ήταν αρκετά περιορισμένη και δύσκολη για τον προμηθευτή των τότε συστημάτων. Οι συνδέσεις στον κύριο υπολογιστή γινόταν με την χρήση ενός **διαύλου** (*bus*) ή ενός **προσαρμογέα δικτύου** που συνδεόταν απευθείας με την Κεντρική Μονάδα Επεξεργασίας (*CPU*) του υπολογιστή. Ο λειτουργικότητα σε αυτά τα συστήματα πρώτης γενιάς ολοκληρώθηκε με την χρήση δύο όμοια εξοπλισμένων συστημάτων κεντρικών υπολογιστών, ένας αρχικός και ένα στήριγμα, που συνδέθηκε στον κοινό δίαυλο (*bus*). Η αρχική λειτουργία του εφεδρικού συστήματος ήταν να ελεγχθεί ο αρχικός και να αναλάβει σε περίπτωση ανιχνευμένης αποτυχίας. Αυτός ο τύπος εφεδρικής λειτουργίας σήμανε ότι ελάχιστη ή καμία επεξεργασία δεν γινόταν στο εφεδρικό σύστημα.

2.9.2 2^η Γενιά – Τα κατανεμημένα συστήματα

Η επόμενη γενιά των συστημάτων SCADA εκμεταλλεύτηκε τις εξελίξεις και τη βελτίωση στη **μικρογράφιση συστημάτων** και την **τοπική τεχνολογία δικτύωσης (τοπικό LAN)** για να διανεμίει την επεξεργασία σε πολλαπλάσια συστήματα. Οι πολλαπλάσιοι σταθμοί, κάθε ένας με μια συγκεκριμένη λειτουργία, συνδέθηκαν με το τοπικό δίκτυο LAN και μοιράστηκαν τις πληροφορίες ο ένας με τον άλλον σε πραγματικό χρόνο. Αυτοί οι σταθμοί είχαν χαρακτηριστικά της κατηγορίας των μίνι υπολογιστών, μικρότεροι και λιγότερο ακριβοί από τα παρόμοια υπολογιστικά συστήματα της πρώτης γενιάς. Μερικοί από αυτούς τους διανεμημένους σταθμούς χρησιμοποιήθηκαν επίσης ως επεξεργαστές επικοινωνιών, με πρωταρχικό τους έργο να μπορούν να επικοινωνούν με τις συσκευές ελέγχου. Μερικοί χρησιμοποιήθηκαν ακόμα και ως ο χείριστες διασύνδεσης, παρέχοντας την διασύνδεση ανθρώπου - μηχανής για τους χειριστές συστημάτων. Τα υπόλοιπα υπολογιστικά συστήματα χρησιμοποιήθηκαν ως επεξεργαστές υπολογισμού ή ως κεντρικοί υπολογιστές βάσεων δεδομένων.

Η διανομή των μεμονωμένων λειτουργιών των συστημάτων SCADA στα πολλαπλάσια συστήματα παρείχε περισσότερη εξουσία επεξεργασίας για το σύστημα συνολικά από την χρήση ενός ενιαίου και μοναδικού επεξεργαστή. Τα δίκτυα που σύνδεσαν αυτά τα μεμονωμένα συστήματα βασίστηκαν γενικά στα πρωτόκολλα του τοπικού LAN και δεν ήταν σε θέση να αναπτύξουν πέρα από τα όρια του τοπικού περιβάλλοντος. Μερικά από τα πρωτόκολλα του τοπικού LAN που χρησιμοποιήθηκαν ήταν ιδιωτικής φύσης, στα οποία ο προμηθευτής δημιούργησε το πρωτόκολλο δικτύων ή την έκδοσή του. Αυτό επέτρεψε στον εκάστοτε προμηθευτή να βελτιστοποιήσει το πρωτόκολλο του τοπικού δικτύου LAN του για την κυκλοφορία σε πραγματικό χρόνο, αλλά περιόρισε τη σύνδεση του από άλλους προμηθευτές στο τοπικό LAN SCADA, δημιουργώντας ταυτόχρονα σοβαρά προβλήματα ασφάλειας, αφού πέρα από τους σχεδιαστές του συστήματος, κανείς άλλος δεν μπορούσε να ξέρει πόσο απλή είναι η πρόσβαση σε ένα σύστημα SCADA. Έτσι, τα συστήματα ασφαλείας αρχικά ήταν υποτιμημένα, κάνοντας έτσι σε επίδοξους hackers πιο εύκολο το έργο τους.

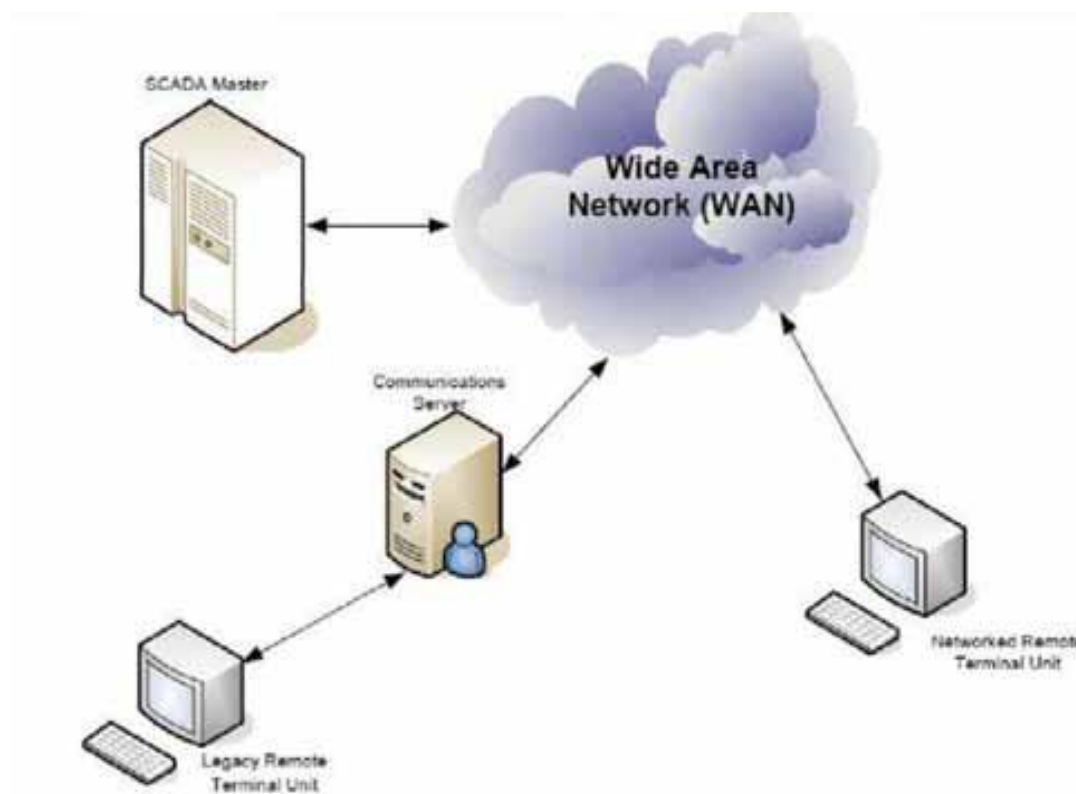


Υπόδειγμα κατακεντρωμένου συστήματος SCADA

Η διανομή της λειτουργίας συστημάτων στα συνδεδεμένα στο δίκτυο συστήματα χρησίμευσε όχι μόνο για την αύξηση της επεξεργαστικής ισχύος, αλλά και στην βελτίωση του πλεονασμού και της αξιοπιστίας του συστήματος συνολικά. Παρά το απλό σχέδιο με την εμφάνιση αρχικού και εφεδρικού εξοπλισμού που χρησιμοποιήθηκε σε πολλά συστήματα πρώτης γενιάς, η διανεμημένη αρχιτεκτονική κράτησε όλους τους σταθμούς που βρίσκονταν συνδεδεμένοι στο τοπικό δίκτυο LAN σε απευθείας σύνδεση πραγματικού χρόνου. Παραδείγματος χάριν, εάν ένας σταθμός HMI επρόκειτο να αποτύχει, ένας άλλος σταθμός HMI θα μπορούσε να χρησιμοποιηθεί για να ενεργοποιήσει το σύστημα, χωρίς αναμονή από το αρχικό σύστημα στο δευτεροβάθμιο. Το δίκτυο (LAN) που χρησιμοποιήθηκε για να επικοινωνήσει με τις συσκευές στον τομέα ήταν κατά ένα μεγάλο μέρος αμετάβλητο από την ανάπτυξη της συνδεσιμότητας του τοπικού LAN μεταξύ των τοπικών σταθμών στον κύριο SCADA. Αυτά τα δίκτυα εξωτερικών επικοινωνιών περιορίστηκαν σε συγκεκριμένους τύπους πρωτοκόλλων και δεν ήταν διαθέσιμα για άλλους τύπους κυκλοφορίας στα δίκτυα. Όπως συνέβη με την πρώτη γενιά των συστημάτων, η δεύτερη γενιά των συστημάτων SCADA περιορίστηκε επίσης στο υλικό, το λογισμικό, και τις περιφερειακές συσκευές που ήταν αναγκαίες.

2.9.3 3^η Γενιά – Τα δικτυωμένα συστήματα

Η σημαντικότερη βελτίωση που παρατηρείται στην τρίτη γενιά είναι αυτή του **ανοίγματος της αρχιτεκτονικής συστημάτων**, της χρησιμοποίησης δηλαδή των ανοικτών προτύπων και των πρωτοκόλλων και την δυνατότητα να διανείμει τη λειτουργία SCADA σε ένα από έναν μεγάλο WAN και όχι μόνο σε ένα τοπικό LAN. Τα ανοικτά πρότυπα αποβάλλουν διάφορους περιορισμούς των προηγούμενων γενεών των συστημάτων SCADA. Η χρησιμοποίηση των **off - the - shelf συστημάτων** διευκολύνει το χρήστη να συνδεθούν περιφερειακές συσκευές τρίτων (όπως τα όργανα ελέγχου, οι εκτυπωτές, οι κινήσεις δίσκων, οι κινήσεις ταινιών, κ.λπ.) με το σύστημα ή και το δίκτυο.



Υπόδειγμα δικτυωμένου συστήματος SCADA

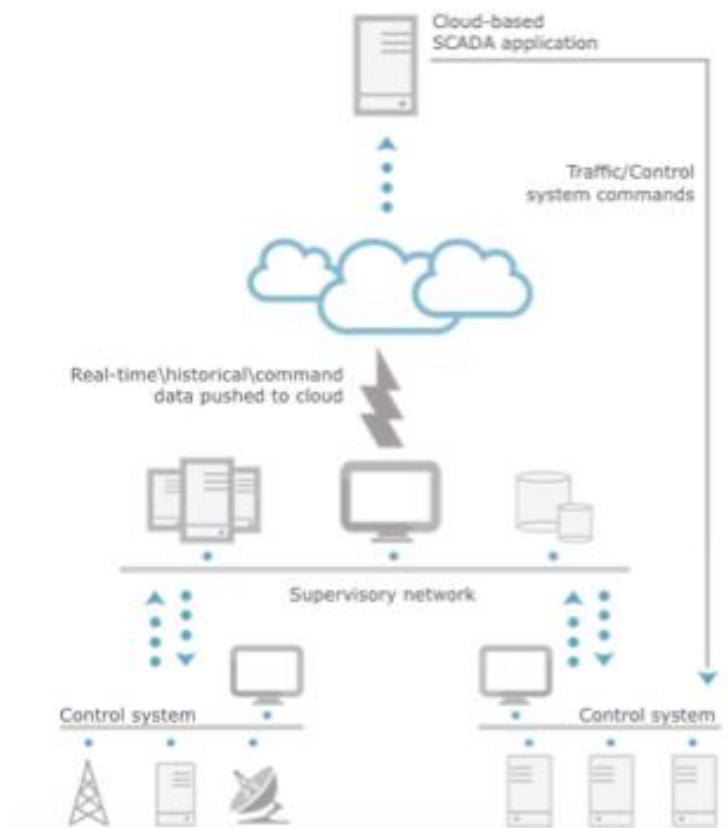
Δεδομένου ότι έχουν κινηθεί για «να ανοίξουν» τα συστήματα, οι προμηθευτές SCADA έχουν πάρει βαθμιαία την επιχείρηση ανάπτυξης υλικού. Αυτοί οι προμηθευτές έχουν κοιτάξει στους προμηθευτές συστημάτων όπως η Compaq, η Hewlett Packard, και η Microsoft για την πείρα τους, στην ανάπτυξη των βασικών πλατφόρμων υπολογιστών και του λογισμικού λειτουργικών συστημάτων. Αυτό επιτρέπει στους προμηθευτές SCADA να συγκεντρώσουν την ανάπτυξή τους σε μια περιοχή όπου μπορούν να προσθέσουν τη συγκεκριμένη αξία σύστημα στο σύστημα, δηλαδή το λογισμικό κύριων σταθμών SCADA. Η σημαντικότερη βελτίωση στα συστήματα τρίτης γενιάς SCADA προέρχεται από τη χρήση των **WAN πρωτοκόλλων** όπως το **πρωτόκολλο Διαδικτύου (IP Internet)** για την επικοινωνία μεταξύ του κύριου σταθμού (MTU) και του εξοπλισμού επικοινωνιών. Αυτό επιτρέπει το τμήμα του MTU που είναι αρμόδιο για τις επικοινωνίες με τις συσκευές τομέων, να είναι ανεξάρτητο από το WAN. Οι προμηθευτές παράγουν τώρα RTU's που να μπορούν να επικοινωνήσουν με τον κύριο σταθμό χρησιμοποιώντας μια σύνδεση Ethernet. Από την όψη του νομίματος όμως, η συνδεσιμότητα με το Internet έχει μεταμορφώσει το σύστημά μας σε ένα σύστημα αρκετά ευάλωτο σε εξωτερικούς κινδύνους. Τα αναβαθμισμένα συστήματα ασφάλειας, η συχνή συντήρηση και η συχνή ενημέρωση των λογισμικών αποτελούν τα κύρια όπλα για την αντιμετώπιση τέτοιων προβλημάτων. Ένα άλλο πλεονέκτημα που προέρχεται από τη διανομή της λειτουργίας SCADA πέρα από ένα δίκτυο WAN είναι αυτό της **ικανότητας επιβίωσης σε περίπτωση καταστροφής**. Η διανομή του συστήματος SCADA πέρα από το τοπικό δίκτυο LAN, στα συστήματα δεύτερης γενιάς, βελτιώνει την αξιοπιστία, αλλά σε περίπτωση συνολικής απώλειας του κύριου σταθμού SCADA, ολόκληρο το σύστημα θα μπορούσε να χαθεί επίσης. Με τη διανομή της επεξεργασίας στις χωριστούς σταθμούς, είναι δυνατό να χτίσει ένα σύστημα SCADA που μπορεί να επιζήσει μιας συνολικής απώλειας οποιασδήποτε θέσης, ακόμη και του κεντρικού σταθμού. Για μερικούς οργανισμούς ή βιομηχανίες που βλέπουν το σύστημα SCADA ως εξαιρετικά κρίσιμη λειτουργία, αυτό είναι ένα πραγματικό όφελος.

2.9.4 4η Γενιά – Internet Of Things

Το **Internet Of Things (IOF)** μπορεί να επεξηγηθεί σαν την δυνατότητα σύνδεσης μέσω δικτύου internet μεμονωμένων απομακρυσμένων εποπτικών συσκευών, σε μία ήδη υπάρχουσα υποδομή. Έτσι, οι επιχειρήσεις μπορούν να δημιουργήσουν νέα συστήματα ή να εγκαταστήσουν ενημερώσεις πολύ εύκολα σε όλες τις συσκευές του συστήματός τους. Η τέταρτη γενιά εξέλιξης των συστημάτων SCADA βασίζεται κυρίως στην τεχνολογία του

Cloud Computing, η οποία παίζει μεγάλο ρόλο στο σημερινό επιχειρηματικό κόσμο γενικότερα.

Χάρη την τεχνολογία αυτή, μας δίνεται πλέον η δυνατότητα δικτύωσης διακομιστών όπου απέχουν ακόμα και χιλιάδες μίλια μακριά. Η διασύνδεση αυτή επιτυγχάνεται μέσω ενός «σύννεφου», μέσω δηλαδή μίας συσκευής η οποία βρίσκεται εγκατεστημένη σε ένα απομακρυσμένο μέρος, τις περισσότερες φορές άγνωστο ακόμα και στον χρήστη. Η πρόσβασή μας σε αυτή την συσκευή γίνεται όπως είναι λογικό, μέσω του internet, παρέχοντάς μας έτσι την δυνατότητα κεντρικής αποθήκευσης αλλά και ταυτόχρονης πρόσβασης σε διάφορες υπηρεσίες από οποιοδήποτε μέρος. Σαν απόρροια όλων των παραπάνω, τα συστήματα SCADA υιοθετούν όλο και περισσότερο την τεχνολογία IOT μειώνοντας σημαντικά το κόστος υποδομών και συντήρησης των συστημάτων τους, αυξάνοντας παράλληλα την ευκολία πρόσβασης και την εποπτεία των βιομηχανικών συστημάτων. Παράλληλα, τα συστήματα SCADA, χρησιμοποιώντας την οριζόντια κλίμακα των συννέφων, μπορούν πλέον να επεκταθούν με περιφερειακές συσκευές (IED's, PLC's) ακόμα πιο εύκολα, άμεσα, και βεβαίως με το λιγότερο δυνατό κόστος.



Υπόδειγμα συστήματος SCADA βασισμένο στην τεχνολογία Cloud Computing

Μία άλλη σημαντική αλλαγή που συντελείται με την γενιά αυτή είναι η **κοινή χρήση πρωτοκόλλων ανοικτού κώδικα**, όπως τα πρωτόκολλα για την ασφάλεια του επιπέδου μεταφοράς (*TLS*). Η χρήση του πρωτοκόλλου αυτού παρέχει στον χρήστη ένα όριο ασφαλείας ακόμα πιο κατανοητό και διαχειρίσιμο, επιδιορθώνοντας έτσι ακόμα περισσότερα τρωτά σημεία του συστήματος σε σχέση με την παρουσία πολλαπλών διαφορετικών πρωτοκόλλων στα ιδιωτικά δίκτυα, που μέχρι πρότινος χρησιμοποιούνταν στα συστήματα SCADA. Η **αποκέντρωση των δεδομένων**, όπως αναφέρθηκε παραπάνω, γεννά επίσης μία διαφορετική προσέγγιση σε σχέση με παραδοσιακά συστήματα εποπτικού ελέγχου. Τα μέχρι πρότινος συστήματα περιλαμβάνουν τη απεικόνιση μέσω γραφικών στο περιβάλλον εργασίας του χρήστη, χρησιμοποιώντας δεδομένα και μετρήσεις οι οποίες αντλούνταν σε τοπικό επίπεδο, από διαφορετικές συσκευές ελέγχου (IED's, PLC's). Σαν αποτέλεσμα αυτού, η γραφική αναπαράσταση 1 προς 1 των δεδομένων αυτών γίνεται ολοένα και πιο προβληματική. Την λύση σε αυτό μας την δίνει το **Data Modeling**, μία έννοια η οποία βασίζεται κυρίως στον αντικειμενοστραφή προγραμματισμό. Σύμφωνα με το μοντέλο αυτό, το εκάστοτε λογισμικό SCADA αναλαμβάνει την εικονική αναπαράσταση καθεμιάς από τις συσκευές που υπάρχουν στο εποπτικό σύστημα. Οι αναπαραστάσεις αυτές περιλαμβάνουν όχι μόνο την καταγραφή της διεύθυνσης μνήμης όπου βρίσκεται η κάθε συσκευή αλλά και κάθε είδους πληροφορία που μπορεί να χρησιμοποιηθεί από άλλες πτυχές του συστήματος SCADA.

2.10 Οφέλη και πλεονεκτήματα των συστημάτων SCADA

Για να χαρακτηριστεί άρτια και απροβλημάτιστη η λειτουργία σε έναν βιομηχανικό οργανισμό, είναι απαραίτητη η πραγματοποίηση του βέλτιστου χειρισμού, με την βοήθεια των διάφορων μηχανικών μέσων, των ρών ενέργειας και υλών μέσω της παρακολούθησης της ροής πληροφορίας. Για να επιτευχθεί αυτός ο στόχος απαιτείται συνήθως να γίνει ένας συμβιβασμός μεταξύ κάποιων οικονομικών και ποιοτικών παραγόντων που θα οδηγήσει σε κάποια άλλα οφέλη. Οφέλη βεβαίως τα οποία μπορούν να διαρθρωθούν σε πολλά επίπεδα τόσο για τον ανθρώπινο παράγοντα όσο και για την παραγωγική διαδικασία. Ορισμένα ενδεικτικά οφέλη που αφορούν κυρίως την παραγωγική διαδικασία αναφέρονται παρακάτω.

- Αύξηση της παραγωγής λόγω αξιοποίησης όλων των παρεχόμενων πόρων στο μέγιστο δυνατό βαθμό, π.χ. λειτουργώντας τη βιομηχανική μονάδα στα ανώτερα όριά της

- Μείωση του κόστους της παραγωγικής διαδικασίας ανά μονάδα προϊόντος λόγω βέλτιστης χρήσης των εσωτερικών πηγών ενέργειας και μείωσης του κόστους εργασίας
- Βελτίωση της ποιότητας των προϊόντων που παράγονται χάρις την δυνατότητα να διατηρούνται οι συνθήκες λειτουργίας μέσα σε στενά όρια ανοχών καθώς επίσης διότι τα σφάλματα πλέον μπορούν να εντοπίζονται σε πολύ σύντομο χρονικό διάστημα
- Ελαχιστοποίηση του κόστους συντήρησης των επιμέρους μηχανών
- Ευελιξία παραγωγής κάτω από συνεχώς μεταβαλλόμενες συνθήκες αγοράς



Απεικόνιση ελέγχου δεξαμενών μέσω συστημάτων SCADA

Ειδικότερα τα συστήματα SCADA προσφέρουν:

- Παρακολούθηση της διαδικασίας παραγωγής, μέσω της οποίας επιτυγχάνεται η εξομάλυνσή της και η μέγιστη απόδοση της βιομηχανικής μονάδας
- Αύξηση της επικοινωνίας μεταξύ των επιπέδων της βιομηχανικής μονάδας, και κυρίως μεταξύ της διοίκησης και της παραγωγής
- Δυνατότητα στο προσωπικό να λαμβάνει αποφάσεις μετά από πληρέστερη ενημέρωση, ώστε να εκπληρώνει τις υποχρεώσεις του με μεγαλύτερη επιτυχία
- Γρηγορότερος εντοπισμός και αντιμετώπιση σφαλμάτων, που εκτός από την βελτίωση της απόδοσης προσφέρει και μείωση του κόστους συντήρησης

- Βελτίωση των συνθηκών ασφαλείας, και εργασίας εν γένει
- Πιο εύστοχες και έγκαιρες πληροφορίες για τη διοίκηση

Όσον αφορά τον ανθρώπινο παράγοντα ορισμένα πλεονεκτήματα είναι τα εξής:

- Περιορισμός του ρόλου του ανθρωπίνου παράγοντα στον τομέα του ελέγχου και μόνο.
- Ανάληψη χειρωνακτικών εργασιών από τις μηχανές.
- Ελαχιστοποίηση κινδύνου εργατικού ατυχήματος σε βιομηχανίες. Οι μηχανές αναλαμβάνουν πλέον τα επικίνδυνα κομμάτια της εργασίας.
- Ελαχιστοποίηση του ανθρωπίνου λάθους.

Κάνοντας μία εκτενής αναφορά σε μερικά πρακτικά παραδείγματα που παρουσιάζουν το όφελος της χρήσης των συστημάτων SCADA στην παραγωγική διαδικασία μπορούμε να αναφέρουμε ότι:

- Παρέχεται η δυνατότητα στη διοίκηση και τους μηχανικούς να προβάλουν πληροφορίες στην επιφάνεια εργασίας του υπολογιστή τους, είτε η πηγή των δεδομένων βρίσκεται κοντά τους είτε μίλια μακριά.
- Αυξάνεται η ικανότητα της πλήρους παραμετροποίηση του εποπτικού συστήματος κατ' ανάγκη του χρήστη, με την βοήθεια χιλιάδων περιφερειακών συσκευών μέτρησης και ελέγχου, όπου μπορούν να ενσωματωθούν στο σύστημα άμεσα και χωρίς επιπλέον κόστος.
- Οι χειριστές μπορούν να χρησιμοποιούν οθόνες PC's που τρέχουν Microsoft Windows για να εποπτεύουν ή και να ελέγχουν τον εξοπλισμό, μέσω εύχρηστων GUI's (Graphical User Interfaces – Γραφικά Περιβάλλοντα Χρήσης).
- Οι ενημερώσεις κρίσιμων καταστάσεων (Alarms) μπορούν να γίνονται με ηχογραφημένα μηνύματα τα οποία μπορούν να εκπέμπονται αυτόματα μέσω τηλεφώνων, ασυρμάτων, δικτύων ηλεκτρονικών υπολογιστών κτλ.
- Ο οποιοσδήποτε χειριστής μπορεί πλέον να αποθηκεύσει μεγαλύτερο όγκο καθώς και να ανακτήσει «ιστορικά» δεδομένων, για περαιτέρω συγκρίσεις, συμπεράσματα ή διάγνωση σφαλμάτων.
- Ο απομακρυσμένος εξοπλισμός μπορεί να ρυθμιστεί από απόσταση, χωρίς την φυσική παρουσία του χρήστη στον χώρο.
- Ο χειριστής είναι πλέον σε θέση να ενσωματώνει σε πραγματικό χρόνο (real time) δεδομένα και μετρήσεις στο εποπτικό σύστημα.

- Επιτρέπεται η χρήση φθηνών και εύρηστων προσωπικών υπολογιστών ως τερματικές συσκευές. Οι υπολογιστές αυτοί είναι πιο εύκολο και οικονομικό να αναβαθμιστούν ή να υποστούν μετατροπές από ότι ο εξειδικευμένος εξοπλισμός.
- Παράλληλα, επιτρέπεται η χρήση σύγχρονων και συνηθισμένων πρωτοκόλλων και υλικών δικτύων, που είναι επίσης εύκολο και οικονομικό να αναβαθμιστούν, να προσαρμοστούν ή να αντικατασταθούν. Με αυτό τον τρόπο επιτυγχάνεται η αξιόπιστη δικτυακή επικοινωνία μεταξύ υλικού διαφορετικών κατασκευαστών.
- Η ύπαρξη τεχνικής υποστήριξης και συντήρησης του συστήματος από τον προμηθευτή του.

2.11 Προβλήματα και μειονεκτήματα των συστημάτων SCADA

Η διασύνδεση των συστημάτων SCADA στον εποπτικό μηχανισμό μίας βιομηχανικής μονάδος σαφώς και μπορεί να μας δώσει άπειρες λύσεις στην διαχείριση του ελέγχου της παραγωγικής διαδικασίας. Ταυτόχρονα όμως, εμφανίζει αρκετά προβλήματα κατά την ενσωμάτωσή τους σε αρκετές σύγχρονες βιομηχανίες. Ενδεικτικά, τα μειονεκτήματα που παρουσιάζουν τα συστήματα αυτά είναι:

- Κατασκευαστικά, τα συστήματα SCADA μπορούν να χαρακτηριστούν ως **πολύπλοκα**, **κοστοβόρα** και αρκετά **απαιτητικά** από πλευράς υλικών. Οι βιομηχανίες παραγωγής ηλεκτρονικών ειδών είναι αναγκασμένες σε έναν ατέρμονο κύκλο αναβάθμισης των παραγόμενων συσκευών για να είναι σε θέση να ακολουθήσουν την συνεχόμενη εξέλιξη των συστημάτων
- Ο χειριστής έχει την δυνατότητα να παρέμβει στην λειτουργία ορισμένων περιφερειακών συσκευών.
- Σε βιομηχανικές μονάδες μεγάλης έκτασης απαιτούνται πολύπλοκες συνδέσεις όπου ορισμένες φορές είναι δύσκολο να υλοποιηθούν.
- Λόγω της πολυπλοκότητας που χαρακτηρίζει τα συστήματα, είναι απαραίτητη η παρουσία ανθρωπίνου δυναμικού με πολλές και εξειδικευμένες τεχνικές γνώσεις όπως αναλυτής συστήματος ή προγραμματιστής αισθητήρων.
- Τα σύγχρονα εποπτικά συστήματα εμφανίζουν αρκετά κενά ασφαλείας λόγω της ύπαρξης του internet ως μέσω επικοινωνίας και διασύνδεσης μεταξύ των συσκευών, δίνοντας έτσι την δυνατότητα υποκλοπής και παραποίησης της παραγωγικής διαδικασίας.

Συνοψίζοντας τις παραπάνω αναφορές, μπορούμε να δούμε συγκεντρωτικά τα πλεονεκτήματα και τα μειονεκτήματα των συστημάτων SCADA, όπως παρουσιάζονται στον επόμενο πίνακα.

Συστήματα SCADA	
<i>Πλεονεκτήματα</i>	<i>Μειονεκτήματα</i>
+ Προβολή και διαχείριση δεδομένων απ' όπου και αν βρίσκεσαι	- Πολύπλοκα, κοστοβόρα και απαιτητικά
+ Πλήρη παραμετροποίηση συστήματος	- Διαρκή αναβάθμιση περιφερειακών συσκευών
+ Εύχρηστος χειρισμός μέσω GUI	- Παρέμβαση σε περιορισμένες συσκευές ελέγχου
+ Άμεση ενημέρωση alarms	- Εγκαταστάσεις με δύσκολη υλοποίηση
+ Μεγαλύτερη δυνατότητα αποθήκευσης δεδομένων	- Παρουσία ανθρωπίνου δυναμικού με εξιδεικευμένες γνώσεις
+ Δυνατότητα ανάκτησης ιστορικού	- Αρκετά κενά ασφαλείας λόγω διασύνδεσης μέσω internet
+ Απομακρυσμένος προγραμματισμός συστήματος	
+ Καταγραφή και ενσωμάτωση δεδομένων σε πραγματικό χρόνο	
+ Χρήση PC's σαν τερματικές συσκευές	
+ Δικτύωση μέσω σύγχρονων πρωτοκόλλων και υλικών	
+ Εύκολη τεχνική υποστήριξη	

Συγκεντρωτικός πίνακας θετικών και αρνητικών στοιχείων συστημάτων SCADA

2.12 Ασφάλεια στα συστήματα SCADA

Στο πρώιμο στάδιο δημιουργίας τους, τα συστήματα SCADA δεν χρειαζόταν να ενσωματώνουν προστασία από εξωτερικούς παράγοντες. Ο λόγος ήταν ότι ελάχιστοι γνωρίζανε τα συστήματα SCADA, πάρα μόνον οι προγραμματιστές και οι διαχειριστές τους. Στη σημερινή εποχή όμως τα εποπτικά συστήματα αποτελούν κύριο στόχο επιθέσεων και κακόβουλων ενεργειών σε μία βιομηχανική μονάδα. Η μετάβαση από τα ιδιωτικά δίκτυα σε μία πιο τυποποιημένη και ανοικτή λύση, παράλληλα με την αύξηση των συζεύξεων των συστημάτων SCADA στα διάφορα εταιρικά δίκτυα, μεταξύ αυτών και του διαδικτύου, μπορεί μεν να έχει δημιουργήσει καλύτερες συνθήκες ενσωμάτωσης διαφορετικών συστημάτων και συσκευών σε μία βιομηχανική μονάδα, αλλά έχει καταστήσει τα βιομηχανικά συστήματα πιο ευάλωτα αυξάνοντας τον κίνδυνο δικτυακών επιθέσεων από τρίτους. Ανατρέχοντας στην ιστορία, την 1η Οκτωβρίου 2003, ο Robert F. Dacey, Διευθυντής για Ζητήματα Ασφάλειας Πληροφοριών στο Γενικό Λογιστήριο του Κράτους (*General Accounting Office - GAO*), κατά την διάρκεια της κατάθεσής του ενώπιον της Υποεπιτροπής Τεχνολογίας, Πολιτικής Πληροφόρησης και Διακυβερνητικών Σχέσεων των Η.Π.Α ανέφερε:

«Για αρκετά χρόνια, κίνδυνοι ασφαλείας έχουν αναφερθεί σε συστήματα ελέγχου, στα οποία βασίζονται πολλές από τις κρίσιμες υποδομές της χώρας για την παρακολούθηση και τον έλεγχο των διαδικασιών αρκετών ευαίσθητων λειτουργιών της χώρας. Πέραν των γενικών κυβερνοαπειλών, οι οποίες έχουν μία σταθερά αυξανόμενη τροχιά, διάφοροι άλλοι παράγοντες έχουν συμβάλει στην κλιμάκωση των κινδύνων που σχετίζονται με τα συστήματα ελέγχου, συμπεριλαμβανομένης (1) της έγκρισης τυποποιημένων τεχνολογιών με γνωστά θέματα ευπάθειας, (2) της σύνδεσης των συστημάτων ελέγχου σε άλλα δίκτυα, (3) των περιορισμών που τέθηκαν σχετικά με τη χρήση των υφιστάμενων τεχνολογιών και πρακτικών ασφαλείας, (4) των μη ασφαλών απομακρυσμένων συνδέσεων και (5) την ευρεία διαθεσιμότητα των τεχνικών πληροφοριών σχετικά με τα συστήματα ελέγχου»

Με λίγα λόγια, οι υπεύθυνοι ασφαλείας των συστημάτων SCADA στις βιομηχανικές μονάδες θα πρέπει να επικεντρωθούν στα παρακάτω ζητήματα:

- Την έλλειψη ενδιαφέροντος για την ασφάλεια και τον έλεγχο ταυτότητας στο σχεδιασμό, την εγκατάσταση και τη λειτουργία ορισμένων δικτύων SCADA που βρίσκονται ήδη εν λειτουργία
- Την πεποίθηση ότι τα συστήματα SCADA υπερτερούν σε θέματα ασφαλείας λόγω της χρήσης εξειδικευμένων πρωτοκόλλων επικοινωνίας και ιδιωτικών διεπαφών σύνδεσης

- Την πεποίθηση ότι ορισμένα παλαιότερα συστήματα SCADA δεν διατρέχουν κίνδυνο υποκλοπής λόγω της μη ύπαρξης του διαδικτύου ως μέσω επικοινωνίας.

Στο επιχειρησιακό περιβάλλον του σήμερα, όπως έχει αναφερθεί και σε προηγούμενες παραγράφους, όλα τα συστήματα εσωτερικής λειτουργίας μίας βιομηχανίας, συμπεριλαμβανομένου και του εποπτικού συστήματος SCADA, συνδέονται με δίκτυα όπου στηρίζονται σε **πρωτόκολλα επικοινωνίας TCP/IP**. Η δικτύωση των περιφερειακών συσκευών ενός συστήματος SCADA (*PLC's, IED's, RTU's*), όπως αναφέρθηκε και σε προηγούμενη παράγραφο, επιτυγχάνεται μέσω διαφορετικών πρωτοκόλλων όπως τα **Modbus, IEC – 60870 - 5** και **DNP3**. Όμως η διασύνδεση των συσκευών αυτών με τους τερματικούς σταθμούς γίνεται συνήθως μέσω τηλεφωνικών γραμμών, μισθωμένων δικτύων, δορυφορικών συνδέσεων, ακόμα και μέσω ραδιοσυχνοτήτων, δημιουργώντας έτσι μία ψευδή αίσθηση ασφάλειας, από την στιγμή που οι τερματικές συσκευές δεν είναι συνδεδεμένες στο ίδιο δίκτυο.

2.12.1 Επιθέσεις στα συστήματα SCADA

Η ασφάλεια σε ένα βιομηχανικό δίκτυο μπορεί να εκτεθεί σε κίνδυνο υποκλοπής μέσω πολλών και διαφορετικών καταστάσεων. Ένας βασικός τρόπος είναι η **μη εξουσιοδοτημένη πρόσβαση** στο κέντρο ελέγχου του λογισμικού, είτε μέσω ανθρώπινης πρόσβασης είτε μέσω αλλαγών που έχουν προκληθεί σκόπιμα ή κατά λάθος στο λογισμικό του συστήματος από μολύνσεις ιών. Επίσης, η **παραβίαση των πακέτων επικοινωνίας** που διακινούνται μέσω του δικτύου του συστήματος SCADA είναι ένας εξίσου συνηθισμένος τρόπος επίθεσης. Σε πολλές περιπτώσεις, το πρωτόκολλο ελέγχου στερείται κάθε μορφής κρυπτογράφησης, επιτρέποντας έτσι στον εισβολέα του συστήματος να ελέγχει εξ' ολοκλήρου μια συσκευή SCADA. Δεν είναι λίγες άλλωστε οι φορές όπου οι χειριστές των συστημάτων SCADA έχουν την λανθασμένη πεποίθηση ότι χρησιμοποιώντας ένα δίκτυο VPN, είναι διασφαλισμένοι από εξωτερικές απειλές. Αγνωούν όμως ότι η ασφάλεια αυτή μπορεί εύκολα να διαρρηχθεί με την φυσική πρόσβαση στο εποπτικό περιβάλλον.

2.12.1.1 Είδη επιθέσεων

Μόλις το εταιρικό δίκτυο διαρρηχθεί, εξαιτίας της συνεχούς και αδιάληπτης λειτουργίας του, ο επιτιθέμενος αποκτά πλέον την δυνατότητα διαχείρισης οποιουδήποτε κομματιού της βιομηχανικής μονάδος επιθυμεί. Τα κυριότερα είδη επιθέσεων αναγράφονται παρακάτω:

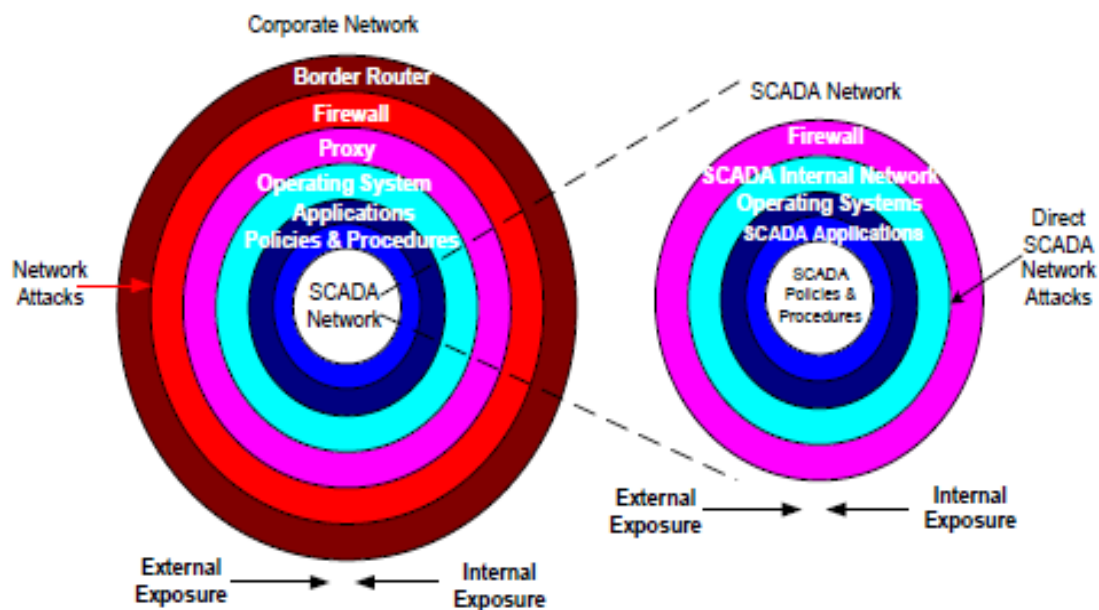
- ❖ Η χρήση μίας επίθεσης DoS (Denial of Service) κατά την οποία στέλνονται στον εξυπηρετητή του συστήματος SCADA συνεχόμενες αιτήσεις προς διεκπεραίωση, με αποτέλεσμα την αδυναμία εξυπηρέτησης τους και τελικά την κατάρρευση του συστήματος.
- ❖ Η πρόσβαση στον κεντρικό υπολογιστή του συστήματος, όπου με μία απλή αλλαγή των διευθύνσεων δικτύου (IP Spoofing), μπορεί να θέσει σε κίνδυνο ολόκληρη την βιομηχανική μονάδα.
- ❖ Η απευθείας διαγραφή βασικών αρχείων στον κεντρικό υπολογιστή του συστήματος SCADA.
- ❖ Η αποστολή στο σύστημα ενός μολυσμένου αρχείου εφαρμογών (virus, Trojan) με σκοπό την απόκτηση του ολικού ελέγχου του συστήματος.
- ❖ Η υποκλοπή των κωδικών σύνδεσης των χειριστών στις διάφορες τερματικές συσκευές. Με αυτόν τον τρόπο, ο επιτιθέμενος μπορεί πλέον να διαγράψει έναν χρήστη από το σύστημα ή ακόμα να αλλάξει τα στοιχεία σύνδεσής του ώστε να μην μπορεί να έχει πρόσβαση στο σύστημα.
- ❖ Η απευθείας πρόσβαση στις βάσεις δεδομένων όπου βρίσκονται οι μετρήσεις που λαμβάνονται από τους αισθητήρες του συστήματος SCADA. Πραγματοποιώντας μία μικρή αλλαγή στα δεδομένα αυτά, αναγκάζει το σύστημα να λαμβάνει λάθος μετρήσεις και πολλές φορές να κλείνει για λόγους προστασίας.

2.12.2 Αντιμετώπιση επιθέσεων

Για να είναι ικανή μία βιομηχανική μονάδα να ανταπεξέλθει στα παραπάνω είδη επιθέσεων, είναι φρόνιμο να αναπτυχθεί μία **στρατηγική ασφαλείας** περιλαμβάνοντας μία γκάμα μέτρων για την προστασία του συστήματος SCADA. Η δομή της στρατηγικής αυτής περιλαμβάνει μέτρα ικανά να καλύψουν επιθέσεις τόσο στην αρχιτεκτονική του εποπτικού συστήματος όσο και γενικότερα σε όλο το βιομηχανικό δίκτυο, προερχόμενες είτε από το εσωτερικό του εταιρικού δικτύου είτε ακόμα από τα ανώτερα και κατώτερα στρώματα, εφαρμογές και περιφερειακές συσκευές αντίστοιχα, του εποπτικού συστήματος.

2.12.2.1 Δακτυλίδι αμυνών – Ring of Defenses

Βασικά χαρακτηριστικά των στρατηγικών ασφαλείας είναι η **ευελιξία** καθώς και η **συντονισμένη λειτουργία** που πρέπει να έχουν όλα τα υποσυστήματα για να μπορέσουν να ανταπεξέλθουν στις επιθέσεις. Χαρακτηριστικά τα οποία μπορούν εύκολα να παρατηρηθούν στην πλέον διαδεδομένη στρατηγική ασφαλείας που χρησιμοποιείτε από το μεγαλύτερο ποσοστό των βιομηχανικών μονάδων, το "**δακτυλίδι των αμυνών**" (*ring of defenses*). Μπορούμε πιο εύκολα να κατανοήσουμε την αρχή λειτουργίας παρατηρώντας το παρακάτω διάγραμμα.



Συσχέτιση μεταξύ βιομηχανικών δικτύων και SCADA

Όπως γίνεται αντιληπτό, το κάθε επίπεδο χρησιμοποιεί μια σειρά αντισταθμιστικών μέτρων για την διασφάλιση των υπηρεσιών τους. Κάνοντας μία σύντομη περιγραφή των καθηκόντων τους έχουμε τα εξής:

❖ Border Router και Firewalls

Το τείχος προστασίας, ρυθμισμένο και συντονισμένο κατάλληλα, είναι ικανό να αποτρέπει επιθέσεις που έχουν να κάνουν με τους κωδικούς πρόσβασης, τις διευθύνσεις IP των συσκευών και με τα διάφορα αρχεία που βρίσκονται αποθηκευμένα σε βάσεις δεδομένων του συστήματος. Παρόλα αυτά, χωρίς την ύπαρξη ενός δυνατού και αξιόπιστου λειτουργικού, οι επιτιθέμενοι μπορούν να

διεισδύσουν στο σύστημα είτε άμεσα μέσω των ιδιωτικών δικτύων είτε κάνοντας μία επίθεση DoS

❖ **Proxy Servers**

Με τον όρο proxy server εννοούμε έναν διακομιστή διαδικτύου, ο οποίος εποπτεύει την σύνδεση μεταξύ του εσωτερικού δικτύου της βιομηχανίας με το internet. Πιο συγκεκριμένα, επιβλέπει οτιδήποτε πακέτα αποστέλλονται από ή προς τις διάφορες εφαρμογές με πρωτόκολλα επικοινωνίας HTTP ή SMTP. Ωστόσο, δεν εξαλείφεται πλήρως η απειλή των επιθέσεων σε επίπεδο εφαρμογών.

❖ **Λειτουργικά Συστήματα (Operating Systems)**

Πρόκειται για τον βασικό πυρήνα κάθε υπολογιστικού συστήματος. Ως αποτέλεσμα αυτού, τα λειτουργικά συστήματα αποτελούν τον πρωταρχικό στόχο κάθε επιτιθέμενου. Τα σχεδιαστικά και λειτουργικά χαρακτηριστικά τους είναι ευρέως γνωστά, κάνοντάς τα έτσι ευάλωτα σε πολλών ειδών επιθέσεις. Η συχνή ανανέωση καθώς και οι αναβαθμίσεις των είδη υπάρχοντων λειτουργικών συστημάτων είναι ίσως οι μόνες λύσεις που να δώσουν ένα μικρό επίπεδο προστασίας στο εποπτικό μας σύστημα.

❖ **Εφαρμογές (Applications)**

Αποτελούν το δεύτερο πιο συχνά παραβιαζόμενο στρώμα σε ένα σύστημα λόγω των πολλών κενών που μπορεί να υπάρχουν στους κώδικες των εφαρμογών. Υπερχειλίσσεις της μνήμης buffer, ιοί και κακόβουλα λογισμικά είναι μερικές από τις πλέον συχνές τεχνικές που χρησιμοποιούνται, εξαφανίζοντας εξ' ολοκλήρου την ύπαρξη οποιουδήποτε προγράμματος ασφαλείας υπάρχει στο βιομηχανικό σύστημα.

❖ **Πολιτικές και Διαδικασίες (Policies and Procedures)**

Χαρακτηρίζονται ως οι θεμέλιοι λίθοι στην σχεδίαση μίας στρατηγικής ασφαλείας σε ένα εποπτικό σύστημα. Περιλαμβάνουν μία σειρά κανόνων για την ενιαία και άρτια λειτουργία του συστήματος. Η επιλογή ασφαλών κωδικών πρόσβασης από τους χρήστες, όπου να μην βασίζεται σε μία γνωστή τους λέξη και κάθε κωδικός να περιέχει τουλάχιστον ένα γράμμα μικρό, ένα κεφαλαίο, έναν αριθμό και ένα σύμβολο πρόκειται για ένα χαρακτηριστικό παράδειγμα της πολιτικής προστασίας του συστήματος.

Η λίστα που αναλύθηκε παραπάνω παρουσιάζεται κοινή σε όλες τις βιομηχανικές μονάδες που έχουν εταιρικά δίκτυα. Στην παρακάτω λίστα αναφέρονται τρόποι προστασίας του δικτύου SCADA σε συνδυασμό με το εταιρικό δίκτυο.

✓ **SCADA Firewalls**

Όπως και τα βιομηχανικά, έτσι και τα συστήματα SCADA μπορούν να εκτεθούν στον κίνδυνο παραβίασης από τρίτους με παρόμοιες μεθόδους. Για την οχύρωση της βιομηχανικής μας μονάδας, προτείνεται η χρησιμοποίηση τοίχων προστασίας τόσο στο εσωτερικό όσο και στο δίκτυο SCADA, για την ολοκληρωμένη κάλυψη της μονάδος από επιθέσεις τρίτων.

✓ **Εσωτερικό δίκτυο σχεδιασμού SCADA (Internal Network)**

Για την προστασία και την ομαλή λειτουργία του βιομηχανικού συστήματος, συνίσταται η οριοθέτηση του δικτύου SCADA σε ένα ξεχωριστό τμήμα. Το τμήμα αυτό διαφοροποιείται έχοντας ξεχωριστές IP διευθύνσεις μέσω κατάλληλων τεχνικών υποδικτύωσης, μειώνοντας έτσι την διακίνηση δεδομένων σε όλο το δίκτυο.

✓ **Λειτουργικό SCADA Server (SCADA Operating System)**

Οι προαναφερθείσες ενέργειες δεν σου εξασφαλίζουν ένα απαραβίαστο βιομηχανικό δίκτυο. Ένας επιτιθέμενος με γνώση και πείρα μπορεί εύκολα να παρακάμψει οποιοδήποτε τοίχος προστασίας και με διάφορες τεχνικές, όπως η τεχνική **ARP (Address Resolution Protocol)** και το **IP spoofing**, έχοντας πρόσβαση σε οποιοδήποτε μέρος του δικτύου επιθυμεί. Έτσι, η έννοια του λειτουργικού συστήματος σε ένα δίκτυο SCADA μπορεί να χαρακτηριστεί ως ζωτικής σημασίας για την προστασία του δικτύου. Η συστηματική συντήρηση και αναβάθμιση του λειτουργικού του SCADA server μπορεί να καλύψει αρκετά τρωτά σημεία του συστήματος.

✓ **Εφαρμογές SCADA (SCADA Applications)**

Ένα άλλο εξίσου ευάλωτο σε επιθέσεις κομμάτι του δικτύου. Όπως και στις εφαρμογές του βιομηχανικού δικτύου που αναλύθηκε παραπάνω, ιοί και διάφορα κακόβουλα λογισμικά αποτελούν τις πιο συνηθισμένες και συνάμα αποτελεσματικές τεχνικές επίθεσης για την διαχείριση των δεδομένων ή των εντολών που εφαρμόζονται στο δίκτυο SCADA. Οι εφαρμογές μπορούν να μολυνθούν είτε εκ των έσω, μέσω εμφύτευσης του ιού σε κάποια συσκευή του δικτύου, είτε από εξωτερικά μολυσμένα αρχεία που αποστέλλονται μέσω mail ή μέσω δικτύου. Για την προστασία του δικτύου από μολύνσεις τέτοιου είδους προβλέπεται η χρήση εξιδεικευμένων αντιϊοικών λογισμικών στα συστήματα όπου η εφαρμογή τρέχει καθώς και η

απενεργοποίηση της δυνατότητας εγκατάστασης μη εξουσιοδοτημένων προγραμμάτων.

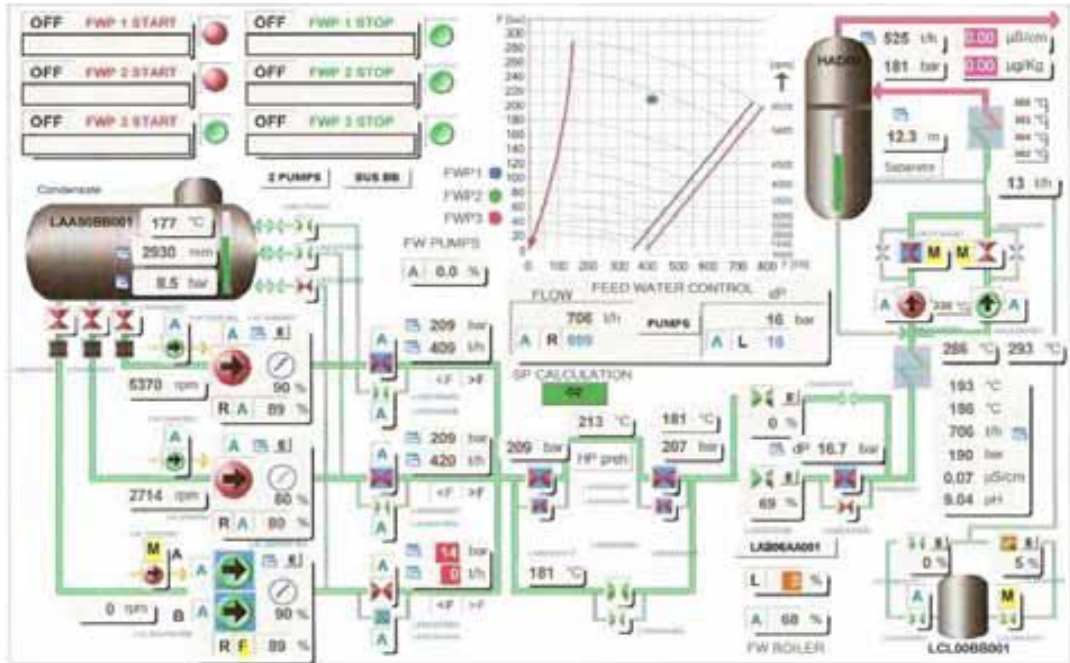
✓ **Πολιτικές και Διαδικασίες SCADA (SCADA Policies and Procedures)**

Οι πολιτικές και οι διαδικασίες που σχετίζονται με την πρόσβαση στα εποπτικά συστήματα του δικτύου, όπως η διαχείριση των κωδικών πρόσβασης, μπορούν να επηρεάσουν σημαντικά την σταθερότητα του δικτύου SCADA.

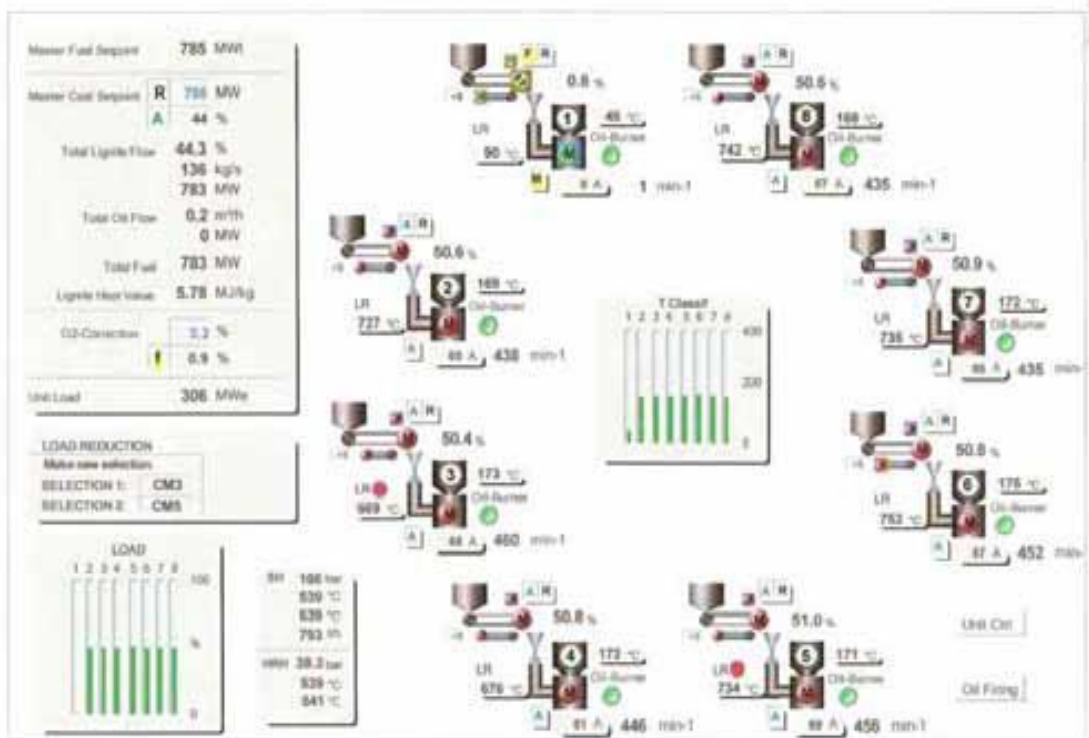
2.13 Εμπορικές εφαρμογές συστημάτων SCADA

Όπως έχουμε ήδη αναφέρει τα συστήματα SCADA έχουν τεράστια εφαρμογή στη βιομηχανία και τα συστήματα ηλεκτρικής ενέργειας. Η πιο ολοκληρωμένη και πιο μεγάλη σε μέγεθος εφαρμογή SCADA στη χώρα μας, έχει γίνει από τη ΔΕΗ για τον έλεγχο και την διαχείριση του δικτύου της. Σε κάθε υποσταθμό (ομάδα ζυγών) υπάρχουν εγκατεστημένες συσκευές εποπτικού ελέγχου (PLC's, IED's, RTU's), τα οποία καταγράφουν την τάση κάθε ζυγού, την ενεργό και άεργο ισχύ που παράγεται ή δαπανάται ανάλογα με το είδος του ζυγού (παραγωγής ή φορτίου αντίστοιχα). Ακόμη λαμβάνονται μετρήσεις σχετικά με τους μετασχηματιστές, όπως τάσεις και ρεύματα σε πρωτεύον και δευτερεύον, θερμοκρασία κλπ. Όλα αυτά τα δεδομένα αποστέλλονται με τη χρήση ενσύρματων τηλεπικοινωνιακών ζεύξεων, στο **Κέντρο Ελέγχου Ενέργειας** της ΔΕΗ, το οποίο είναι εγκατεστημένο στον **Άγιο Στέφανο**. Εκεί τα δεδομένα αυτά εισάγονται σε μια βάση δεδομένων πραγματικού χρόνου (*real time*) και χρησιμοποιούνται από ειδικά προγράμματα, οικονομικής ανάλυσης και εκτίμησης κατάστασης τα οποία χρησιμοποιούν αλγορίθμους βελτιστοποίησης και τεχνητής νοημοσύνης. Έτσι ελέγχεται αν υπάρχει πτώση τάσης σε κάποιο ζυγό ή αν υπάρχει υπερφόρτιση κάποιας γραμμής. Επίσης δίνονται εντολές για την εκκίνηση ή το σταμάτημα μιας γεννήτριας ενός σταθμού παραγωγής (θερμικού, υδροηλεκτρικού κλπ), Η βάση δεδομένων που αναφέραμε πιο πάνω μεταβάλλεται δυναμικά με την σάρωση των πληροφοριών από τους σταθμούς ελέγχου. Υπάρχει όμως και μια δεύτερη βάση δεδομένων στην οποία αποθηκεύονται οι τιμές των μεταβλητών μιας δεδομένης χρονικής στιγμής. Έτσι είναι δυνατόν αργότερα να μελετηθεί η συμπεριφορά του συστήματος, για παράδειγμα σε ένα βραχυκύκλωμα ή ένα black-out και να βρεθεί ο καλύτερος χειρισμός που θα μπορούσε να γίνει σε περίπτωση επανάληψης του φαινομένου αυτού. Στην επόμενη σελίδα, παρατίθενται φωτογραφίες του συστήματος εποπτικού ελέγχου SCADA που χρησιμοποιεί της ΔΕΗ, όπως αναλύθηκε παραπάνω. Συγκεκριμένα, στην *εικόνα 2.12.1* φαίνεται η κατάσταση στο σύστημα δεξαμενών νερού ενώ στην *εικόνα 2.12.2*

παρατηρούμε ένα παράδειγμα του συστήματος εποπτείας καύσης λιγνίτη ενός σταθμού παραγωγής ηλεκτρικής ενέργειας.



Εικόνα 2.12.1



Εικόνα 2.12.2

2.14 Το μέλλον των συστημάτων SCADA

Σύμφωνα με πρόσφατες έρευνες που υλοποιήθηκαν πάνω στο θέμα, οι εταιρείες σχεδίασης λογισμικών SCADA κυκλοφορούν συνήθως μια σημαντική έκδοση λογισμικού, με το πολύ δύο πρόσθετες δευτερεύουσες εκδόσεις το χρόνο. Η ταχύτατη εξέλιξη των συστημάτων αυτών έγκειται στο γεγονός ότι πρέπει να εκμεταλλευτούν τις νέες ευκαιρίες αγοράς, για να καλύψουν τις νέες απαιτήσεις των πελατών τους και τις νέες τεχνολογίες. Δεδομένου ότι οι ανάγκες των βιομηχανιών αυξάνονται, οι νέες εκδόσεις SCADA έχουν ως σκοπό να χειριστούν τις συσκευές όπως και ολόκληρα τα συστήματα ως πλήρεις οντότητες (*classes*) που τοποθετούν όλες τις συγκεκριμένες ιδιότητες και τη λειτουργία τους σε αυτά. Επιπλέον, θα υποστηρίζουν την ανάπτυξη πολύ-ομάδων (*multi team development*). Οι μελλοντικές τάσεις είναι τόσο το λογισμικό των PLC όσο και το λογισμικό HMI/SCADA να ενοποιηθούν και να θεωρούνται έως ένα πλήρες και ενοποιημένο πακέτο.

Κεφάλαιο 3

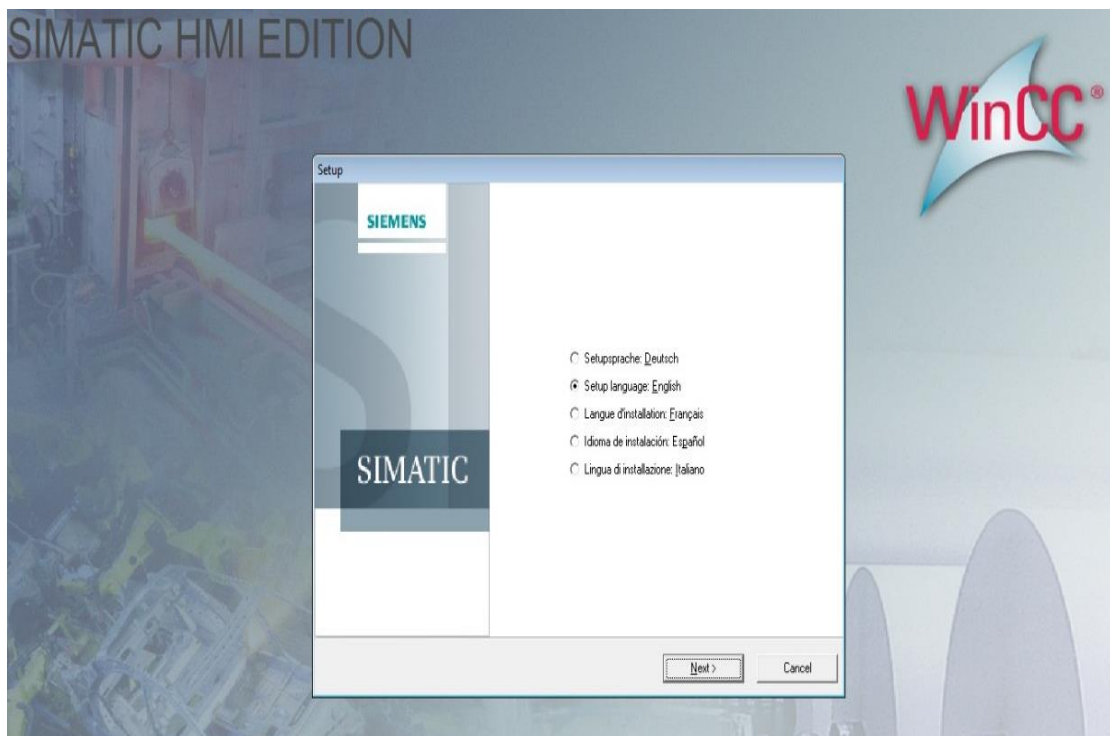
Εγκατάσταση του λογισμικού
Siemens SIMATIC WinCC 7.2

3.1 Εγκατάσταση του λογισμικού SIMATIC WinCC

Για την εκπόνηση της πτυχιακής μας εργασίας, ήταν επιβεβλημένη η εγκατάσταση του λογισμικού SIMATIC WinCC σε έναν ηλεκτρονικό υπολογιστή, αναλαμβάνοντας τον ρόλο του HMI, έτσι ώστε να γίνει η ανάπτυξη των πειραματικών εφαρμογών που θα αναλυθούν σε παρακάτω κεφάλαια. Στην συνέχεια, παρουσιάζονται αναλυτικά τα βήματα που απαιτήθηκαν για την εγκατάσταση του λογισμικού αυτού. Για την εγκατάσταση του **SIMATIC WinCC** χρησιμοποιήθηκε ένας εικονικός ηλεκτρονικός υπολογιστής με χαρακτηριστικά τα οποία αναγράφονται παρακάτω:

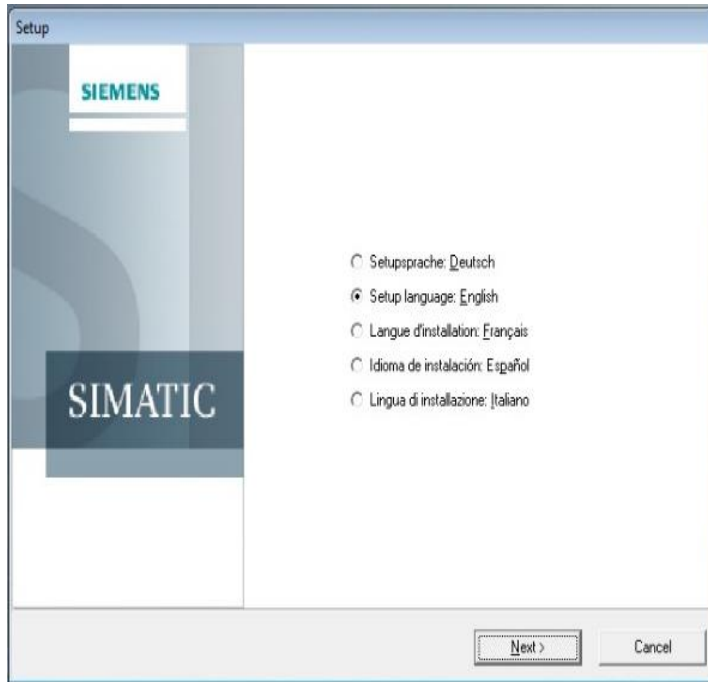
- **Επεξεργαστική ισχύς (CPU)** 1,8MHz Dual Core
- **Μνήμη RAM** 2GB
- **Σκληρός δίσκος HDD** 100GB
- **Λογισμικό** Window 7 Home Edition 32bit

Εισάγοντας το DVD με το πρόγραμμα SIMATIC WinCC εμφανίζεται το παράθυρο υποδοχής του λογισμικού, όπως παρουσιάζεται στην *εικόνα 3.1.1*.



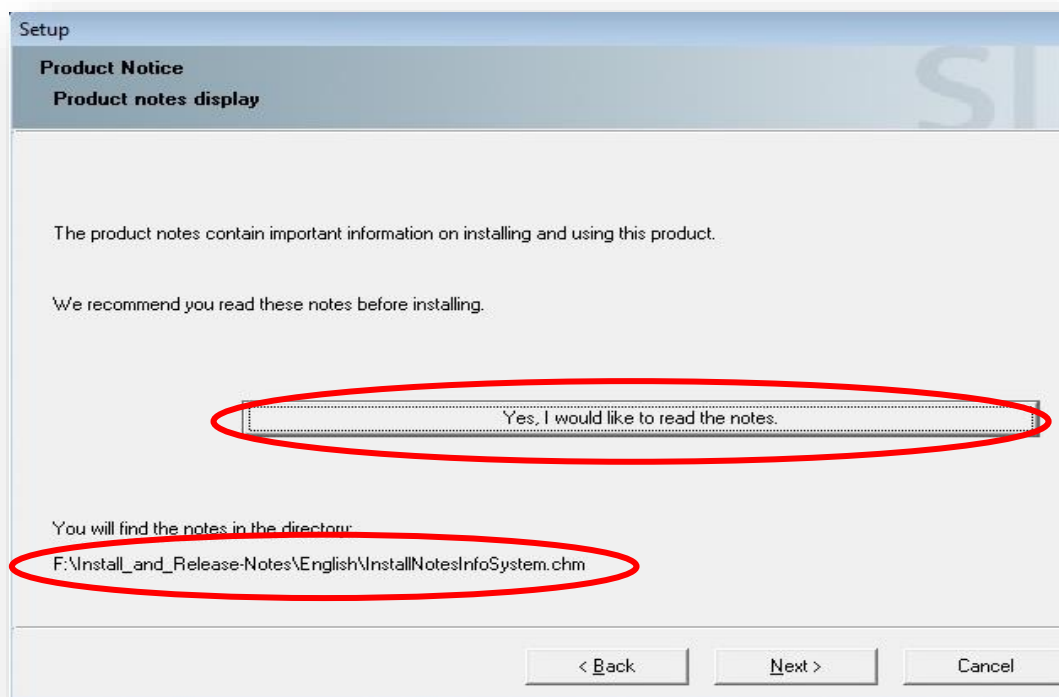
Εικόνα 3.1.1

Στο αναδυόμενο παράθυρο θα πρέπει να επιλέξουμε την γλώσσα με την οποία θα συνεχίσουμε στην εγκατάσταση του προγράμματος. Στην εγκατάστασή μας, επιλέξαμε την **αγγλική γλώσσα** (εικόνα 3.1.2).



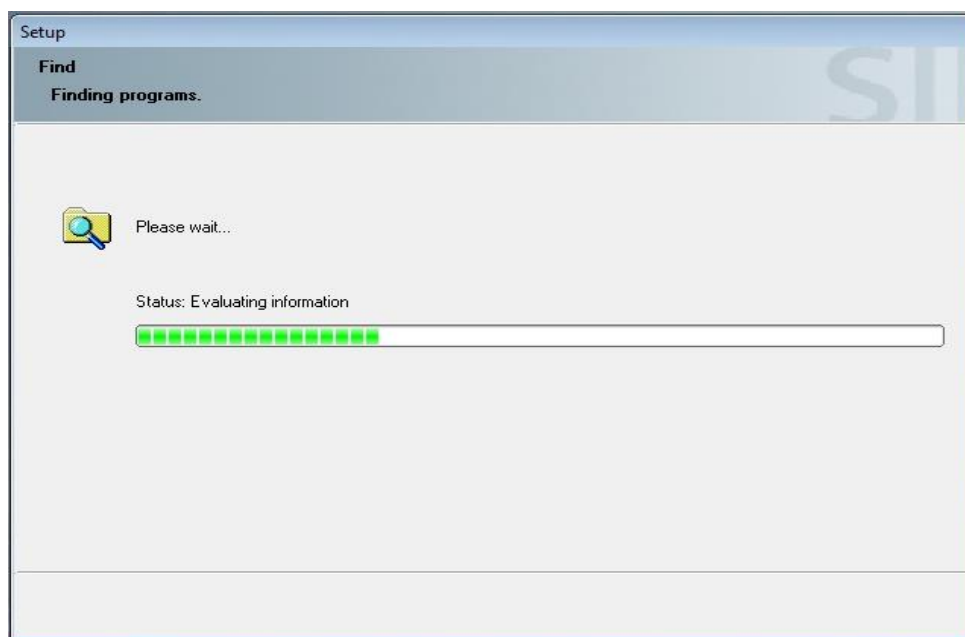
Εικόνα 3.1.2

Κάνοντας κλικ στο **κουμπί Next**, μας εμφανίζεται το παράθυρο πληροφοριών και ενημέρωσης σχετικά με τα δικαιώματα και τις δυνατότητες του προγράμματος. Μπορούμε να τις διαβάσουμε είτε κατά την εγκατάσταση είτε αργότερα ακολουθώντας το installation path που αναγράφεται, όπως φαίνεται στην *εικόνα 3.1.3*. Προχωρούμε την εγκατάσταση κάνοντας κλικ στο **κουμπί Next**.

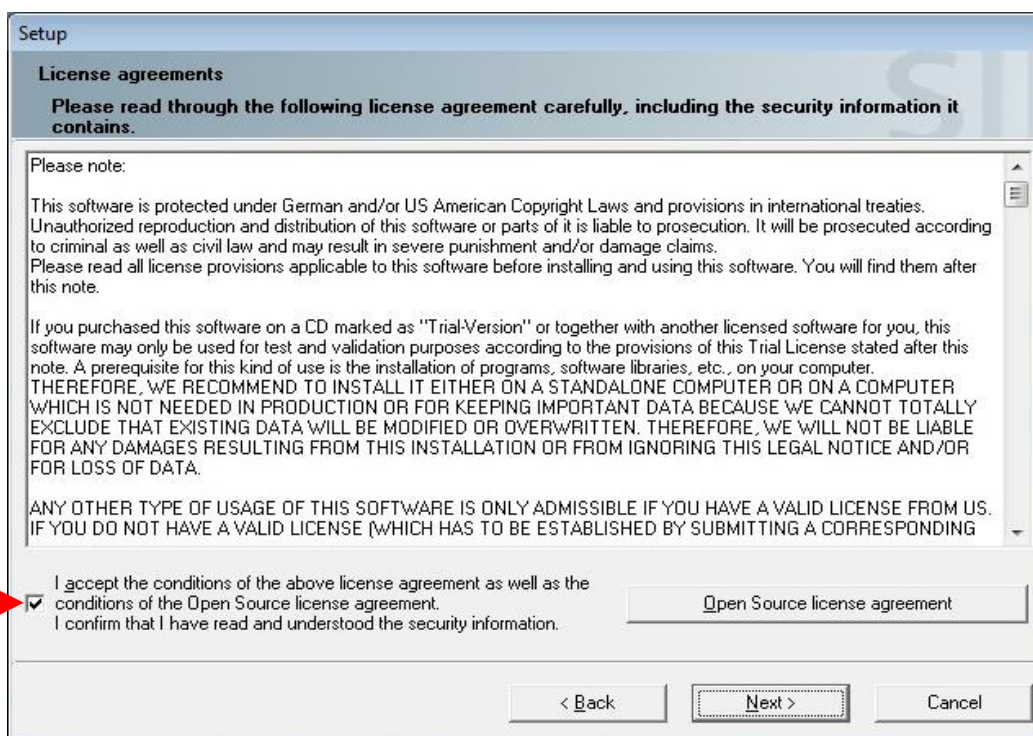


Εικόνα 3.1.3

Έπειτα, το πρόγραμμα αναλύει και επεξεργάζεται τις δυνατότητες του συστήματός μας προκειμένου να δει εάν είναι κατάλληλο για την εγκατάσταση του λογισμικού SIMATIC WinCC (εικόνα 3.1.4). Εάν ναι, εμφανίζει παράθυρο όπου αναλύονται οι κανονισμοί και τα δικαιώματα που έχει το λογισμικό. Για να προχωρήσουμε παρακάτω, θα πρέπει να τικάρουμε το κουτί στο κάτω αριστερά μέρος του παραθύρου και εν συνεχεία να κάνουμε κλικ στο **κουμπί Next** (εικόνα 3.1.5).

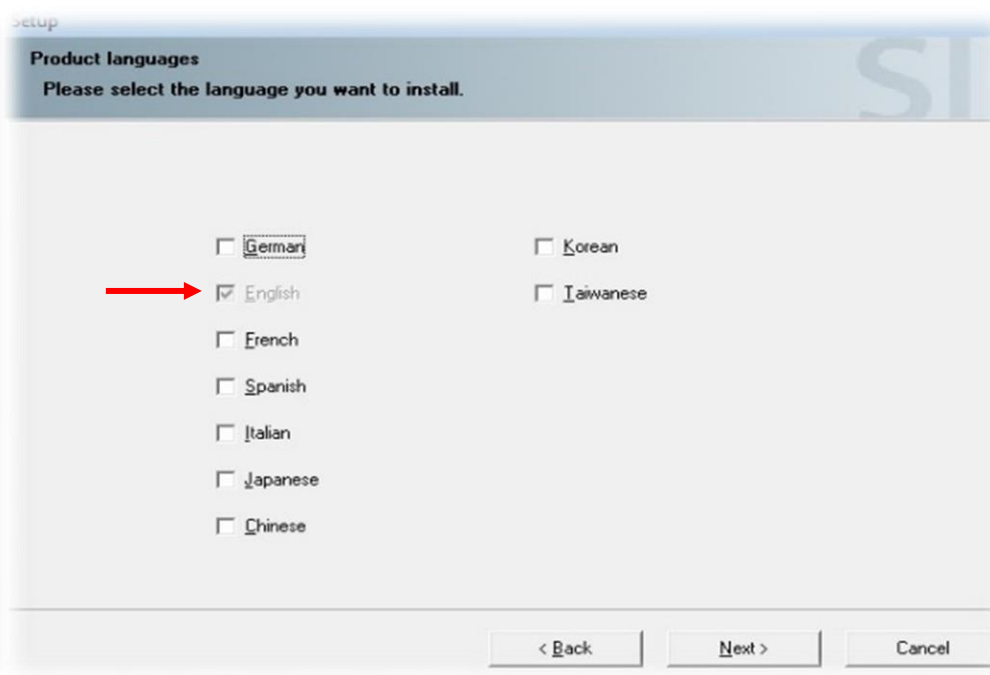


Εικόνα 3.1.4

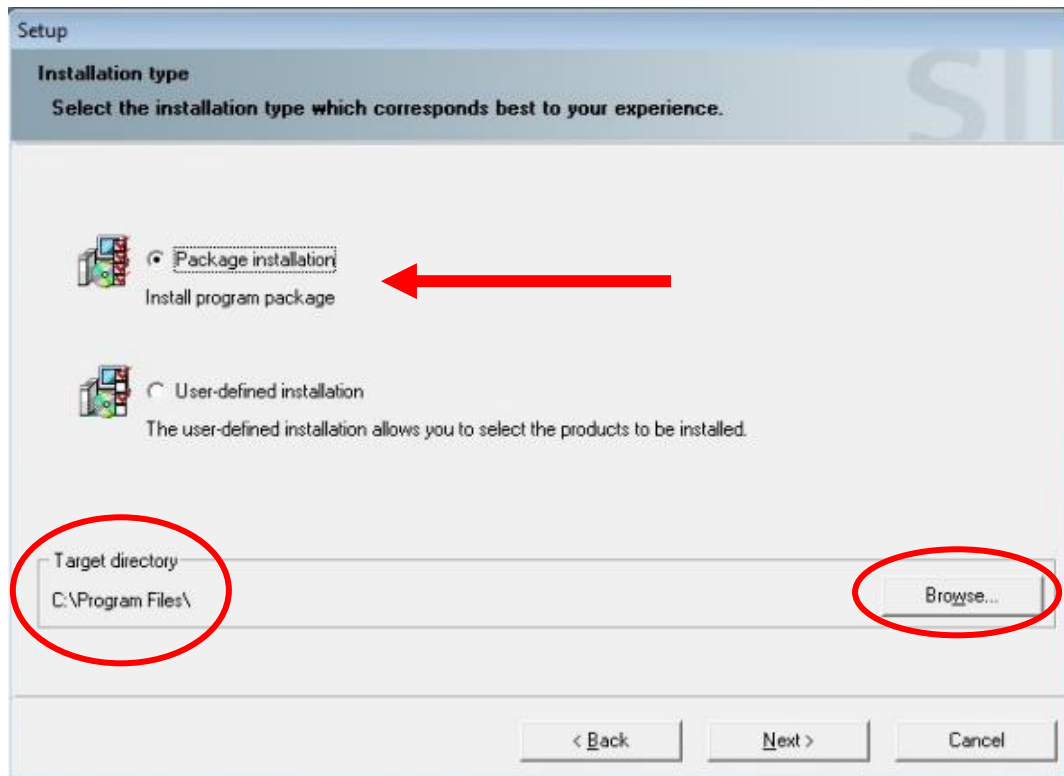


Εικόνα 3.1.5

Στην συνέχεια μπαίνουμε στο κυρίως κομμάτι της εγκατάστασης του λογισμικού SIMATIC WinCC. Κάνοντας κλικ στο κουμπί Next στο προηγούμενο παράθυρο, μας εμφανίζονται οι **επιλογές γλώσσας** που έχουμε για το πρόγραμμά μας (εικόνα 3.1.6). Στην εργασία μας έχουμε επιλέξει μόνο την **αγγλική γλώσσα**. Κλικάροντας το Next, μας τίθεται πλέον το ερώτημα του **τύπου εγκατάστασης** που θέλουμε ακολουθήσουμε για το λογισμικό μας. Στον τύπο **package installation** επιτρέπουμε στο πρόγραμμα εγκατάστασης να επιλέξει εκείνο τα απαραίτητα για το λογισμικό μας υποπρογράμματα που πρέπει να εγκατασταθούν στο σύστημά μας. Αντιθέτως, επιλέγοντας τον τύπο **user-defined installation**, ο χρήστης είναι σε θέση να επιλέξει εκείνος ποια υποπρογράμματα θέλει να εγκαταστήσει στο σύστημά του. Στο παράδειγμά μας, επιλέγουμε τον τύπο **package installation**. Επίσης, στο κάτω μέρος του παραθύρου μας, στο πεδίο **target directory**, εμφανίζεται η τοποθεσία στον σκληρό μας δίσκο όπου θα εγγραφεί το λογισμικό μας. Από το κουμπί **browse**, μπορούμε κάλλιστα να ορίσουμε διαφορετικό path που επιθυμούμε να γίνει εγκατάσταση το WinCC (εικόνα 3.1.7). Για λόγους λειτουργικότητας, έχουμε αφήσει το **default path** εγκατάστασης που μας έχει ορίσει το πρόγραμμα.

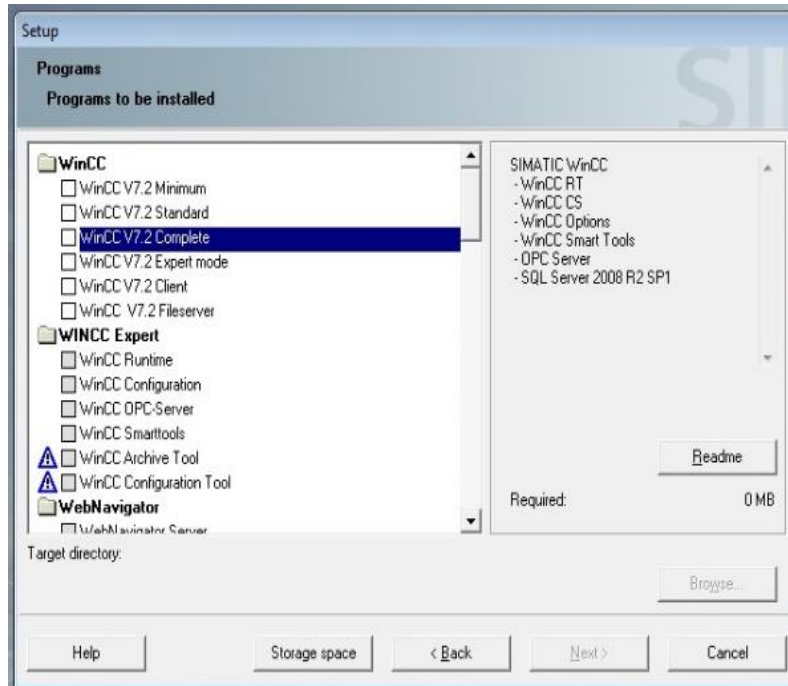


Εικόνα 3.1.6



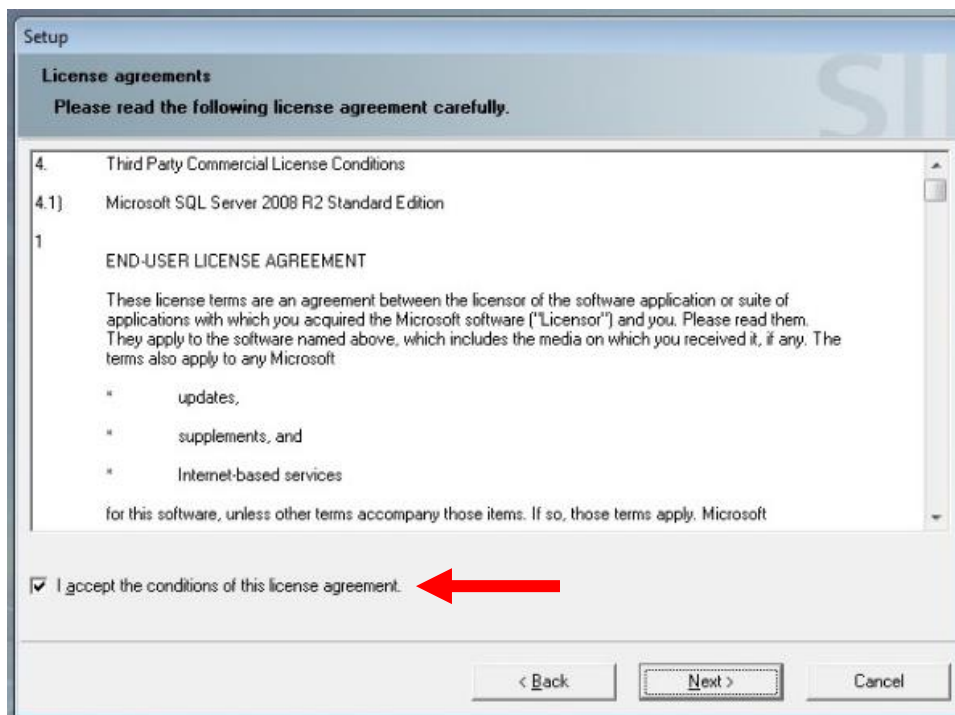
Εικόνα 3.1.7

Έχοντας επιλέξει τον τύπο εγκατάστασης που θα ακολουθήσουμε, στο επόμενο παράθυρο (εικόνα 3.1.8) παρουσιάζονται τα **υποπρογράμματα** που πρόκειται να εγκατασταθούν στο σύστημά μας σχετικά με το λογισμικό SIMATIC WinCC. Τικάροντας το ανάλογο πακέτο προγραμμάτων μας εμφανίζει επιλεγμένα τα υποπρογράμματα που θα εγκατασταθούν. Αντιθέτως, στα μη επιλεγμένα, με λευκό χρώμα παρουσιάζονται τα υποπρογράμματα που μπορούν να προστεθούν σαν συμπληρωματικά στο λογισμικό μας ενώ με ανοικτό γκρι τα υποπρογράμματα τα οποία δεν μπορούν να συνδυαστούν με το πακέτο που έχουμε επιλέξει. Παράλληλα, τα προγράμματα όπου φέρουν το εικονίδιο προσοχής (⚠) στην αριστερή τους πλευρά, δεν δίνονται προς εγκατάσταση λόγω απουσίας κάποιων βασικών παραμέτρων στο σύστημά μας. Μπορούμε εύκολα να δούμε ποιες παράμετροι απουσιάζουν από το παράθυρο στα δεξιά μας, απλά επιλέγοντάς τες. Για την πτυχιακή μας εργασία, έχουμε επιλέξει το πακέτο **WinCC V7.2 Complete**.



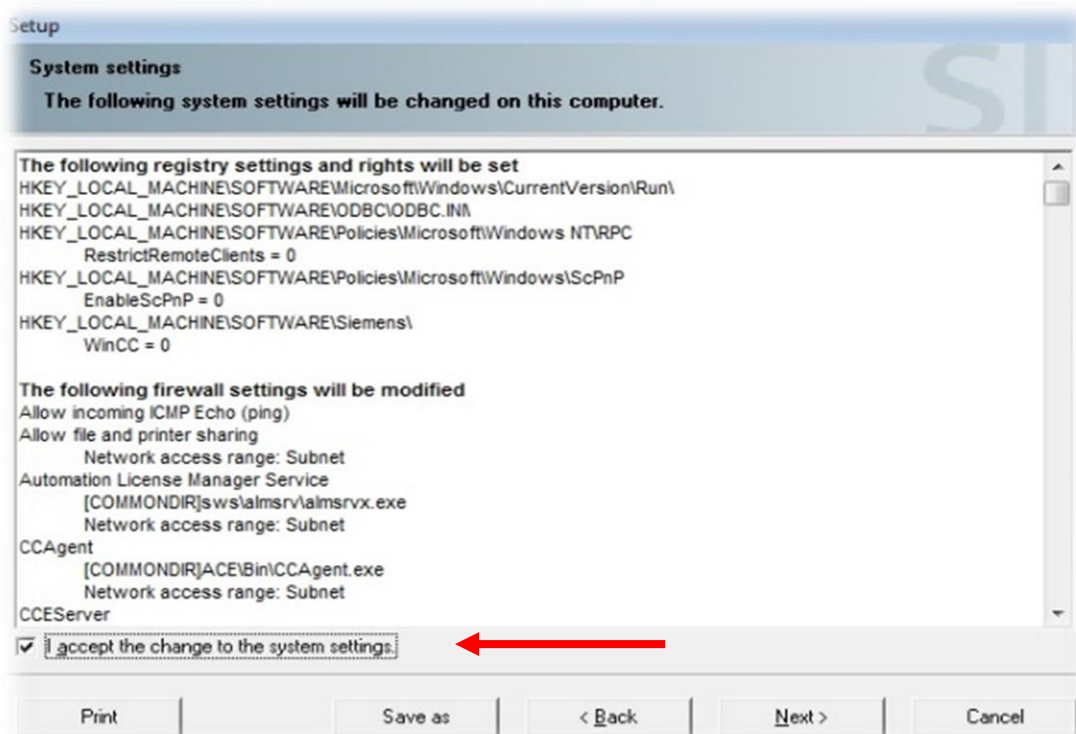
Εικόνα 3.1.8

Κάνοντας κλικ στο **κουμπί Next**, παρουσιάζονται οι κανονισμοί και τα δικαιώματα των επιπρόσθετων προγραμμάτων του SIMATIC WinCC που θα εγκατασταθούν στο σύστημά μας (εικόνα 3.1.9). Για να προχωρήσουμε παρακάτω, θα πρέπει να **τικάρουμε το κουτί** στο κάτω αριστερά μέρος του παραθύρου και εν συνεχεία να κάνουμε κλικ στο **κουμπί Next**.



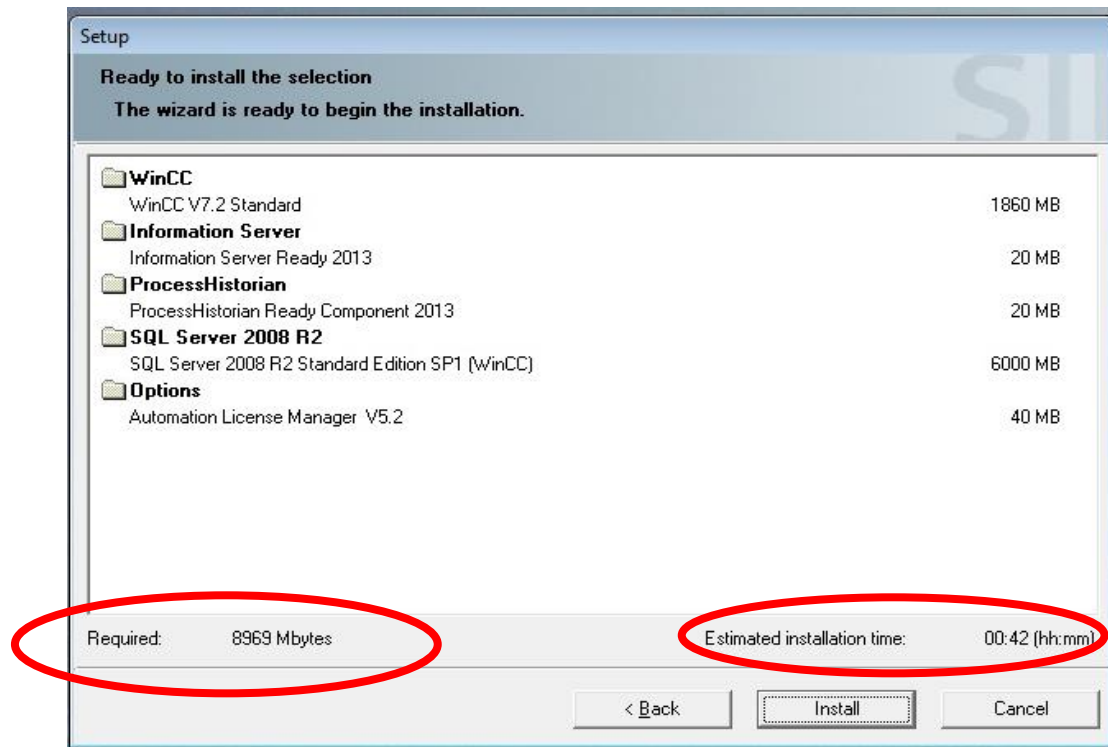
Εικόνα 3.1.9

Στο επόμενο βήμα, μας παρουσιάζονται οι αλλαγές στις ρυθμίσεις που πρόκειται να επιφέρει το λογισμικό στο σύστημά μας. Στο παράθυρο, όπως φαίνεται παρακάτω, στην *εικόνα 3.1.10*, εμφανίζονται αναλυτικά οι αλλαγές και οι παραμετροποιήσεις που θα λάβουν χώρα μετά το πέρας της εγκατάστασης του λογισμικού SIMATIC WinCC στο σύστημά μας. Ομοίως με παραπάνω, θα πρέπει να **τικάρουμε το κουτί** στο κάτω αριστερά μέρος του παραθύρου και παράλληλα να κάνουμε κλικ στο **κουμπί Next** για να προχωρήσουμε στο βήμα της εγκατάστασης.



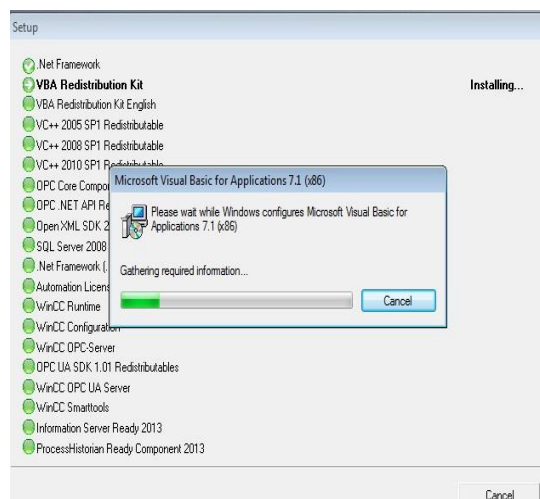
Εικόνα 3.1.10

Εν συνεχεία, μας παρουσιάζεται μία σύνοψη με τις επιλογές που έχουμε κάνει όσον αφορά την εγκατάσταση του λογισμικού στο σύστημά μας. Στο αναδυόμενο παράθυρο μπορούμε να πάρουμε εύκολα πληροφορίες σχετικά με τα προγράμματα που έχουμε επιλέξει να εγκαταστήσουμε, το μέγεθός τους, καθώς και με τον εκτιμώμενο χρόνο εγγραφής του προγράμματος SIMATIC WinCC. Το συγκεκριμένο βήμα αποτελεί το τελικό μας βήμα πριν την εγκατάσταση του λογισμικού. Εάν οι επιλογές μας είναι σωστές, κάνουμε κλικ στο **κουμπί Install**, για να ξεκινήσει η διαδικασία εγκατάστασης (*εικόνα 3.1.11*). Πληροφοριακά, όπως φαίνεται και στην παρακάτω εικόνα, για την εκπόνηση της πτυχιακή μας εργασίας, το συνολικό αποθηκευτικό μέγεθος που χρειάστηκε για το λογισμικό και τα υποπρογράμματα του SIMATIC WinCC ήταν 8969MB ενώ ο εκτιμώμενος χρόνος εγγραφής του στο σύστημά μας ήταν περίπου 42 λεπτά.

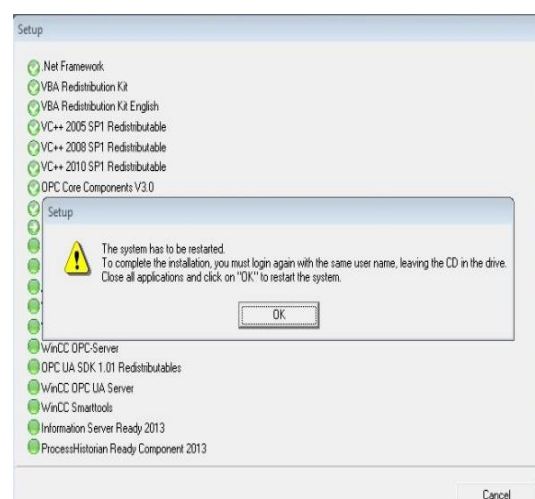


Εικόνα 3.1.11

Μπαίνοντας πλέον στην **διαδικασία της εγκατάσταση του λογισμικού SIMATIC WinCC** καθώς και των υποπρογραμμάτων στο σύστημά, μας παρουσιάζεται μία λίστα όπου μας παρέχει συνεχή ενημέρωση σχετικά με την κατάσταση όπου βρίσκεται η εγγραφή του λογισμικού όπως επίσης και με τα προγράμματα όπου εγκαθίστανται στο σύστημά μας (εικόνα 3.1.12). Λόγω της ύπαρξης των διαφόρων συμπληρωματικών προγραμμάτων που είναι απαραίτητα για την ορθή λειτουργία του λογισμικού μας, δεν είναι λίγες οι φορές όπου το σύστημα θα πρέπει να επανεκκινήσει, ενημερώνοντάς μας με ανάλογο μήνυμα (εικόνα 3.1.13).

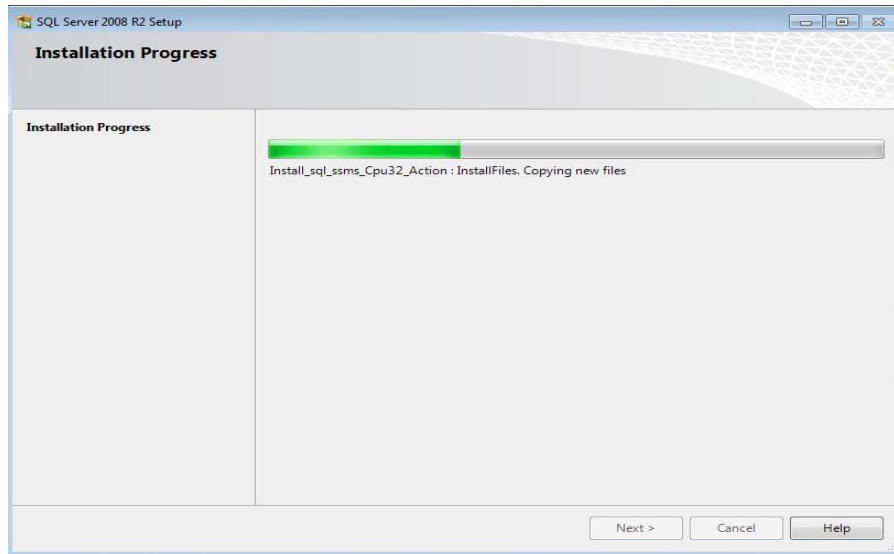


Εικόνα 3.1.12

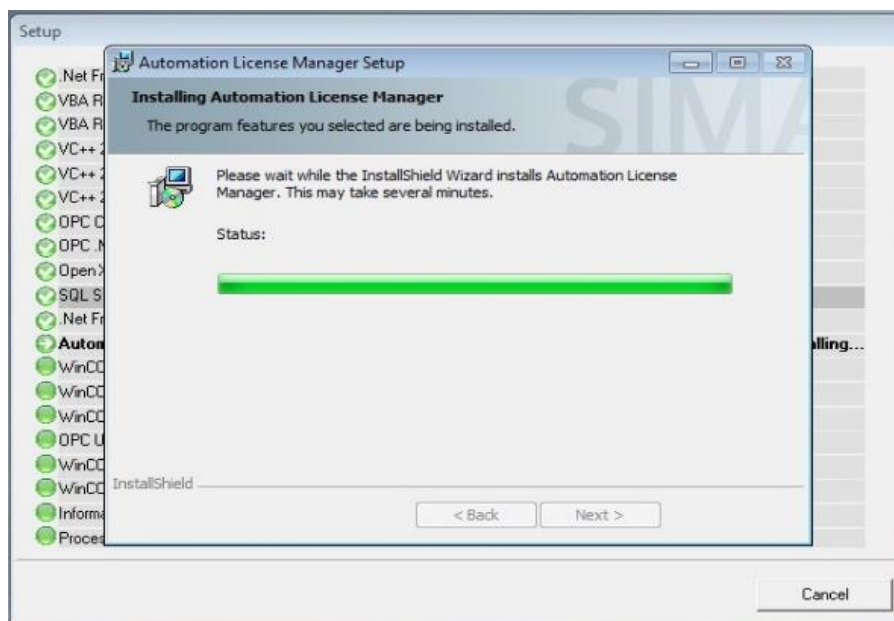


Εικόνα 3.1.13

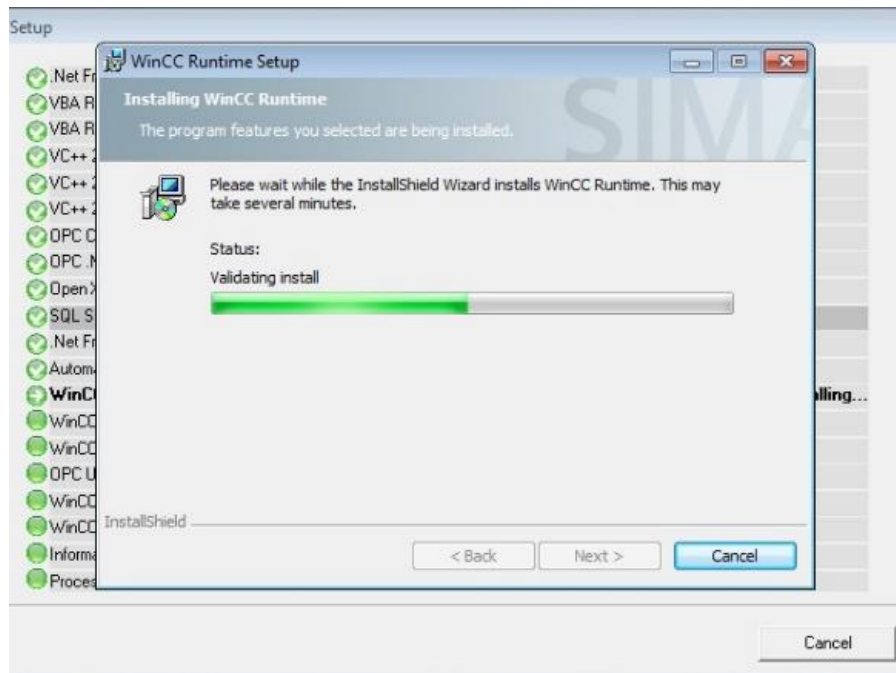
Αμέσως μετά την πρώτη επανεκκίνηση του συστήματος, ο οδηγός εγκατάστασης εκκινεί αυτόματα και συνεχίζει την εγγραφή του λογισμικού SIMATIC WinCC στον σύστημά μας, προχωρώντας παράλληλα στην εγκατάσταση των εναπομεινάντων συμπληρωματικών προγραμμάτων, όπως ο **SQL Server 2008** (εικόνα 3.1.14), ο **.Net Framework v4.0**, το **Automation License Manager** (εικόνα 3.1.15), αλλά και τα βασικά δομικά κομμάτια του λογισμικού μας, το **WinCC Runtime** (εικόνα 3.1.16) και το **WinCC OPC - Server**. Ο χρήστης στο σημείο αυτό δεν χρειάζεται να προβεί σε καμία ενέργεια.



Εικόνα 3.1.14

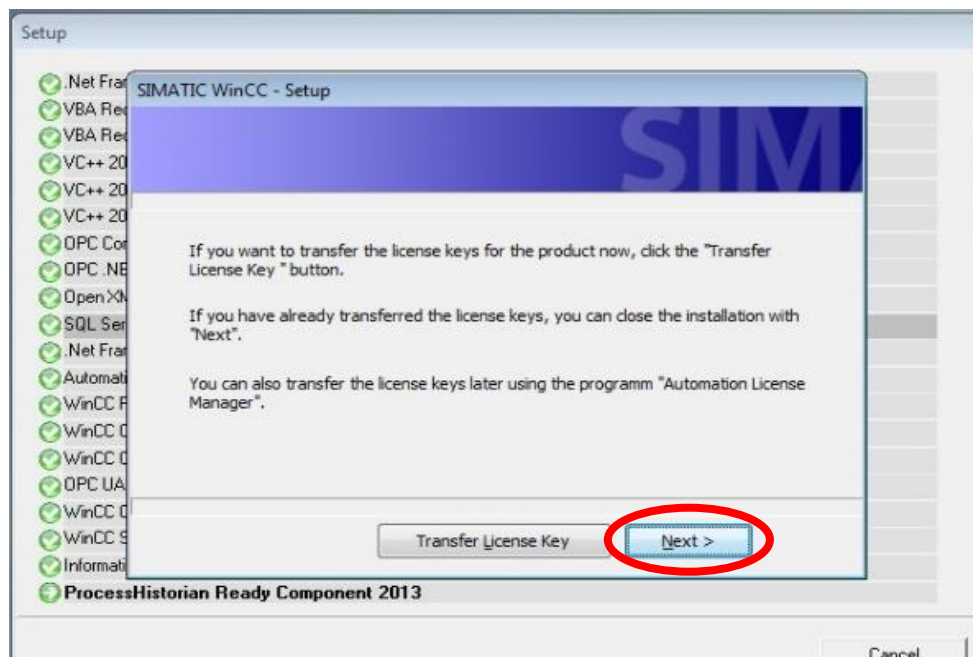


Εικόνα 3.1.15



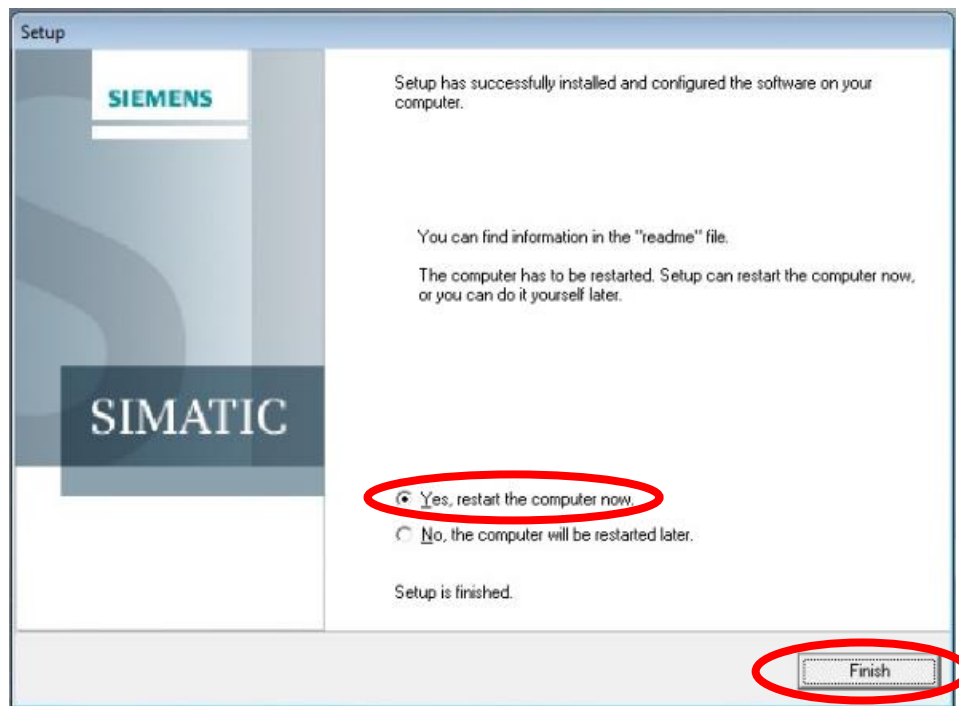
Εικόνα 3.1.16

Μετάπειτα τις παραπάνω εγγραφές των προγραμμάτων, ο οδηγός εγκατάστασης μας εμφανίζει ένα μήνυμα στην οθόνη όπου αφορά την **ενεργοποίηση** αλλά και τις **άδειες χρήσης του λογισμικού** (εικόνα 3.1.17). Στις επιλογές που μας δίνει, θα κάνουμε **κλικ στην επιλογή Next**, για να μπορέσουμε να έχουμε την δυνατότητα ενεργοποίησης του SIMATIC WinCC αργότερα, με την βοήθεια του **Automation License Manager** αλλά και της **διαδικασίας Crack** όπου θα παρουσιάσουμε εκτενέστερα παρακάτω.



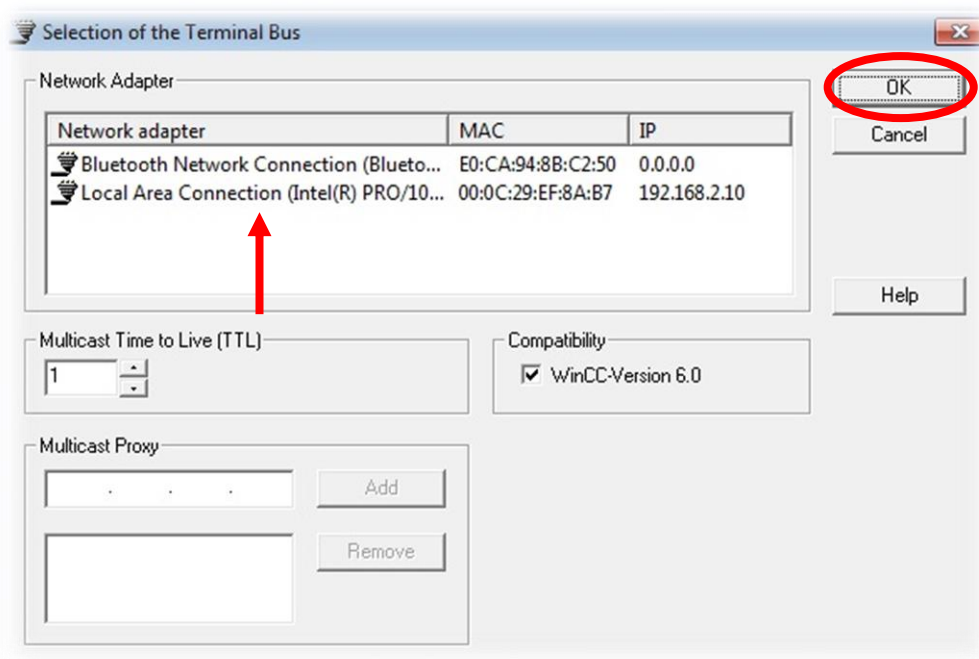
Εικόνα 3.1.17

Κάνοντας κλικ στο κουμπί Next, όπως αναφέραμε παραπάνω, μας εμφανίζεται το **παράθυρο σύνοψης της εγκατάστασης** όπου μας ενημερώνει ότι η εγκατάσταση μας πραγματοποιήθηκε επιτυχώς. Για να αποθηκευτούν και οι τελευταίες παραμετροποιήσεις, θα πρέπει να κάνουμε επανεκκίνηση στο σύστημά μας, κλικάροντας την ανάλογη επιλογή στο παράθυρο μας και πατώντας έπειτα το **κουμπί Finish** (εικόνα 3.1.18).



Εικόνα 3.1.18

Κατά την εκκίνηση του συστήματος, μας εμφανίζεται ένα αναδυόμενο παράθυρο στο οποίο θα πρέπει να οριστεί **ο τρόπος σύνδεσης μας με το βιομηχανικό δίκτυο της μονάδας εποπτείας μας**. Για την εκπόνηση της πτυχιακής μας εργασίας, επιλέγουμε **σύνδεση μέσω τοπικού δικτύου Ethernet**, κάνοντας απλά κλικ στην επιλογή **Local area connection**, όπως αυτή εμφανίζεται στην *εικόνα 3.1.19*. Από το παράθυρο αυτό, μπορούμε παράλληλα να ρυθμίσουμε τον χρόνο **TTL (Time To Live)** που επιθυμούμε καθώς και να ορίσουμε ακόμα και **proxy server**. Στο συγκεκριμένο παράδειγμα, ορίζουμε τον χρόνο **TTL ίσο με 1**. Έχοντας τελειώσει με τις τελευταίες παραμετροποιήσεις, τις αποθηκεύουμε κάνοντας κλικ στο **κουμπί OK**. Το λογισμικό SIMATIC WinCC είναι πλέον εγκατεστημένο ορθά στο σύστημά μας!!!

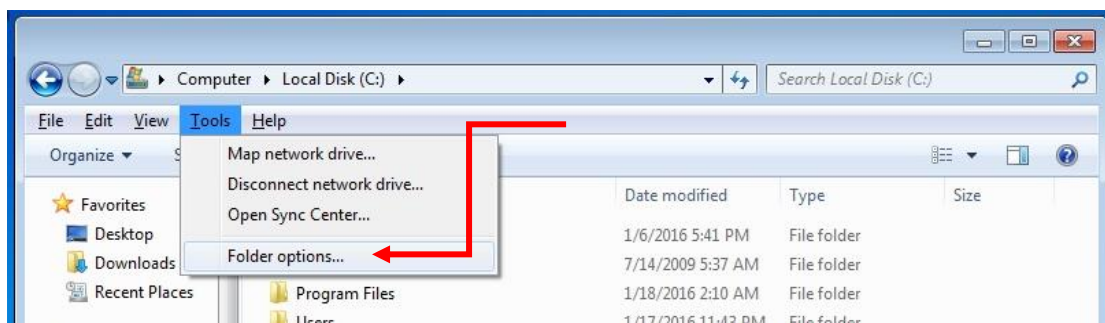


Εικόνα 3.1.19

3.2 Ενεργοποίηση Λογισμικού WinCC μέσω προγράμματος crack

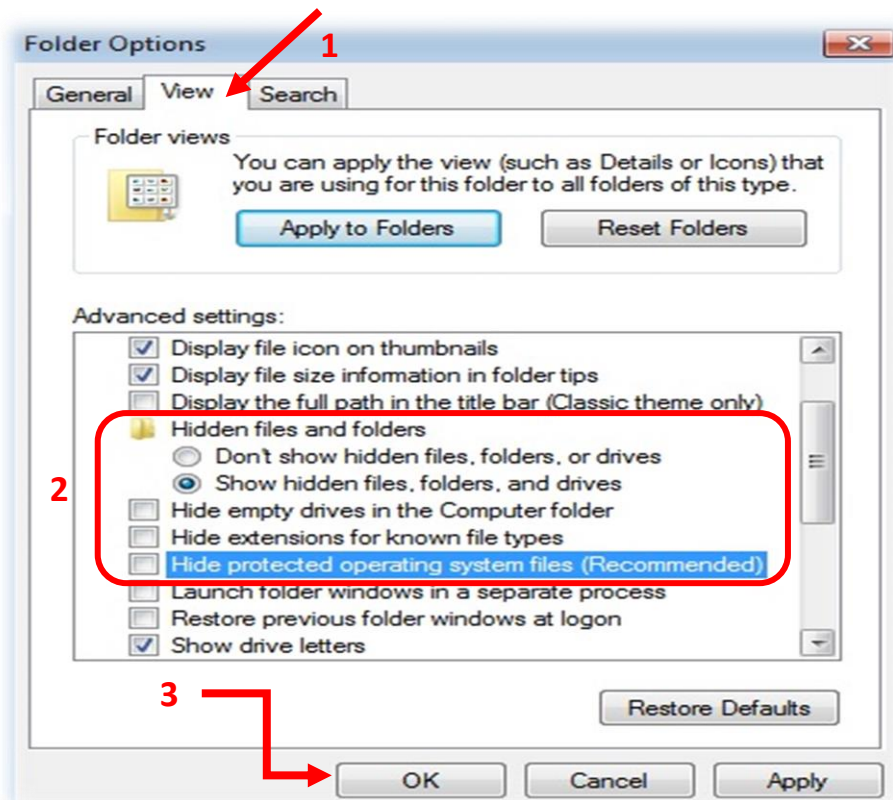
Έχοντας υλοποιήσει πλέον την ορθή εγκατάσταση και παραμετροποίηση του λογισμικού SIMATIC WinCC, όπως αναλύθηκε και στην παραπάνω ενότητα, για να έχουμε την δυνατότητα ολοκληρωτικής εποπτείας και ελέγχου της βιομηχανικής μας μονάδος, για τους σκοπούς της πτυχιακής μας εργασίας, απομένει μόνο η ενεργοποίηση του προγράμματος στο σύστημά μας. Για τον λόγο αυτό, χρίζεται επιβεβλημένη η εφαρμογή ορισμένων ρυθμίσεων στο λογισμικό αλλά και στο σύστημά μας, όπως αυτές παρουσιάζονται εκτενέστερα παρακάτω.

Το πρώτο βήμα που θα πρέπει να ακολουθήσουμε είναι να **εμφανίσουμε τα κρυμμένα αρχεία και φακέλους** που υπάρχουν εγκατεστημένα στο σύστημά μας. Για να πραγματοποιηθεί αυτό θα πρέπει να πάμε στις **Επιλογές Φακέλων (Folder Options)** από το μενού **Εργαλεία (Tools)**, όπως αυτό παρουσιάζεται στην **εικόνα 3.2.1**.



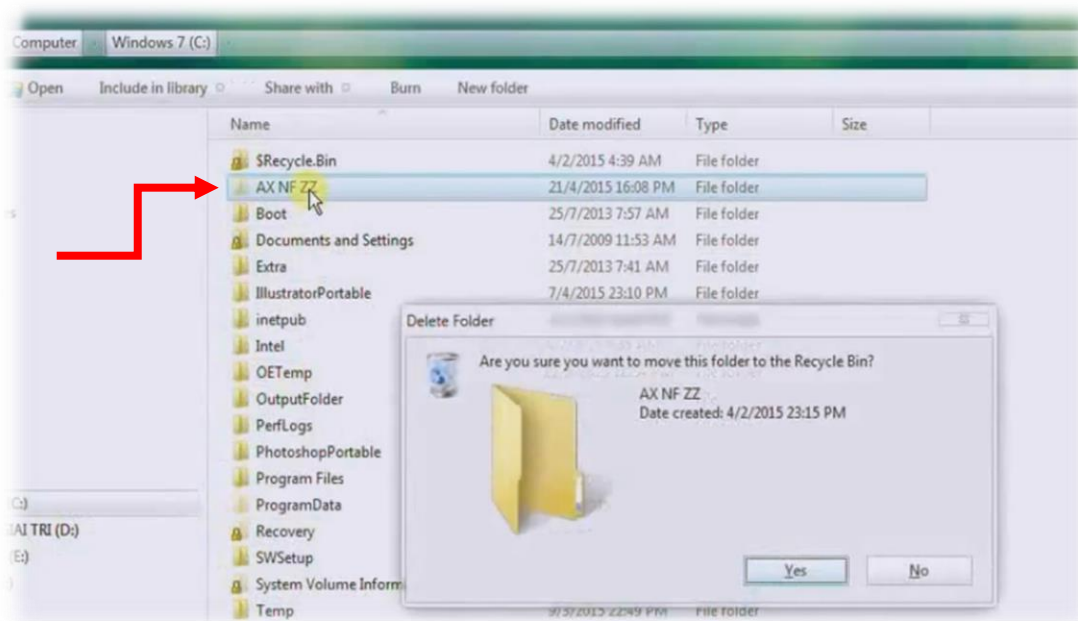
Εικόνα 3.2.1

Εν συνεχεία, επιλέγουμε την καρτέλα **Προβολή (View)** και επιλέγουμε την **Εμφάνιση Κρυμμένων Αρχείων, Φακέλων και Οδηγών (Show hidden files, folders and drivers)** καθώς επίσης απενεργοποιούμε τις **επιλογές Μη Εμφάνισης Άδειων Αρχείων, Μη Εμφάνισης Επεκτάσεων Γνωστών Τύπων Αρχείων και Μη Εμφάνισης Προστατευόμενων Αρχείων Συστήματος**. Στο αναδυόμενο παράθυρο προειδοποίησης που θα μας εμφανιστεί, επιλέγουμε **OK**. Για την αποθήκευση των ρυθμίσεών μας, κάνουμε κλικ στο **κουμπί OK** (εικόνα 3.2.2).



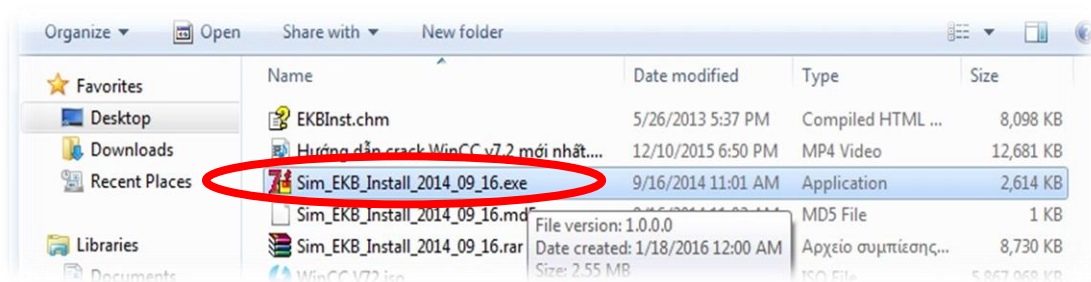
Εικόνα 3.2.2

Αφού ολοκληρώσουμε τις παραπάνω ρυθμίσεις, θα πρέπει μπούμε στο στον φάκελο **Ο Υπολογιστής μου (My Computer)**, στον σκληρό δίσκο όπου έχουμε επιλέξει να γίνει η εγγραφή του λογισμικού SIMATIC WinCC κατά την εγκατάσταση του, και να **διαγράψουμε τον κρυφό φάκελο AX NF ZZ** που υπάρχει (εικόνα 3.3.3)

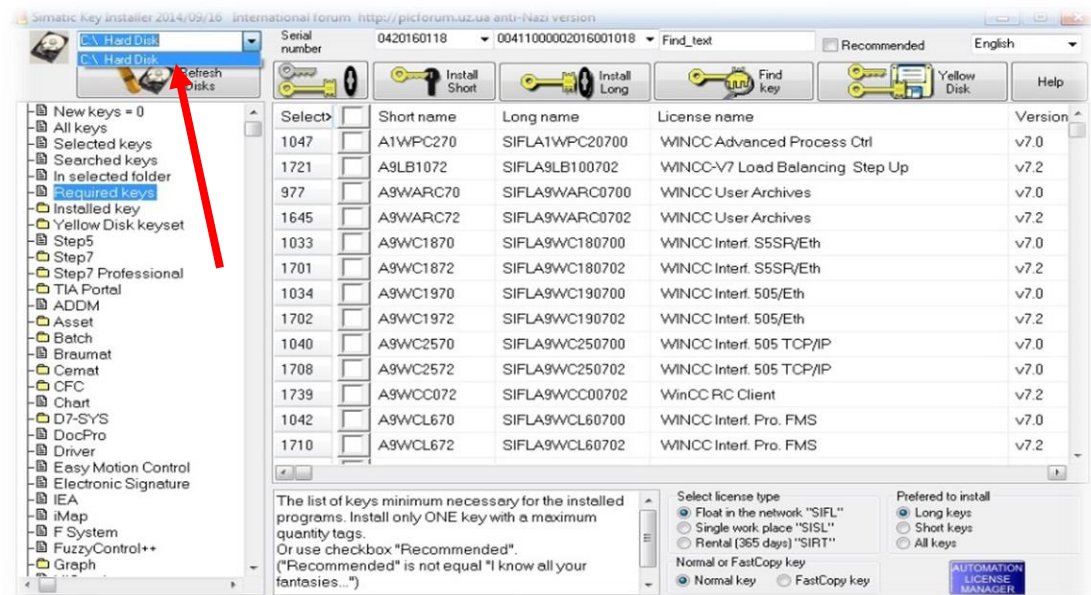


Εικόνα 3.2.3

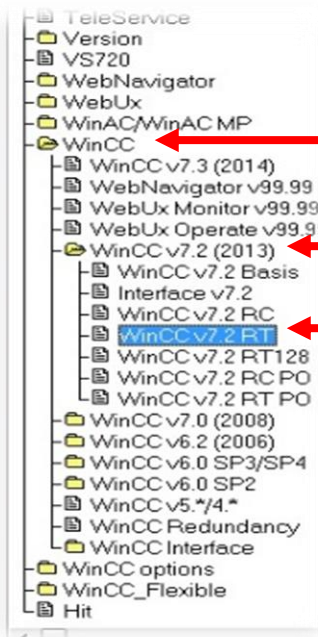
Συνεχίζοντας την διαδικασία ενεργοποίησης, θα χρησιμοποιήσουμε την **εφαρμογή SIM_EKB_Install_2014**, η οποία υπάρχει στον αρχικό φάκελο του λογισμικού μας (εικόνα 3.2.4). Ανοίγοντας την εφαρμογή αυτή με διπλό αριστερό κλικ, μας εμφανίζεται το βασικό παράθυρο ρυθμίσεων, πάνω στο οποίο θα ασχοληθούμε για να ενεργοποιήσουμε το SIMATIC WinCC (εικόνα 3.2.5). Πρώτο μας βήμα είναι να επιλέξουμε **τον αποθηκευτικό χώρο όπου βρίσκεται εγκατεστημένο το λογισμικό** μας από την αναδυόμενη λίστα στο πάνω αριστερό μέρος της οθόνης μας. Έπειτα, από την αριστερή στήλη επιλέγουμε **τον φάκελο προορισμού** με όνομα **WinCC**, στην συνέχεια ανοίγουμε τον φάκελο με όνομα **WinCC v7.2_(2013)** και επιλέγουμε το αρχείο **WinCC v7.2RT**, όπως εμφανίζεται στην εικόνα 3.2.6.



Εικόνα 3.2.4

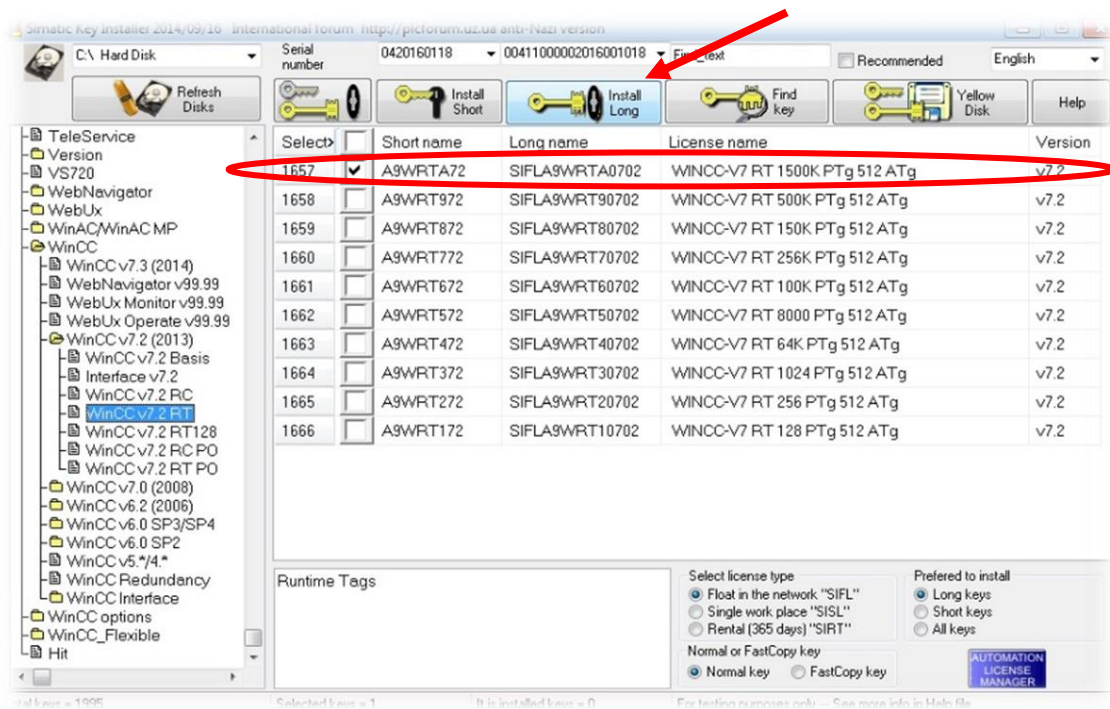


Εικόνα 3.2.5



Εικόνα 3.2.6

Επιλέγοντας το παραπάνω αρχείο, στα αριστερά της οθόνης μας μπορούμε εύκολα να παρατηρήσουμε ότι εμφανίστηκαν διάφοροι κωδικοί ενεργοποίησης και άδειες λειτουργίας για το λογισμικό μας. Από την λίστα αυτή, θα επιλέξουμε **το πρώτο κλειδί ενεργοποίησης** και θα κάνουμε κλικ στο **κουμπί Install Long**, όπου βρίσκεται ακριβώς από πάνω μας (εικόνα 3.2.7). Θα δούμε άμεσα ότι το κλειδί που είχαμε επιλέξει, θα αποεπιλεγθεί αυτόματα, επισημαίνοντας μας έτσι ότι το κλειδί ενεργοποίησης περάστηκε στο λογισμικό μας. Για να αποθηκευτούν οι ρυθμίσεις αυτές, θα πρέπει να βγούμε από την εφαρμογή αυτή και να κάνουμε **επανεκκίνηση του συστήματός μας**.



Εικόνα 3.2.7

Έχοντας πλέον ολοκληρώσει όλες τις απαιτούμενες ενέργειες, το λογισμικό SIMATIC WinCC είναι πλέον ενεργοποιημένο στο σύστημά μας και **έτοιμο προς χρήση!!!**

Κεφάλαιο 4

Εισαγωγή στο λογισμικό
SIMATIC WinCC 7.2

4.1 Η δημιουργία του λογισμικού SIMATIC WinCC

Το λογισμικό **SIMATIC WinCC** αναπτύχθηκε και κυκλοφόρησε στις αγορές από την εταιρία Siemens κατά το έτος 1996. Στην σημερινή του μορφή, χρησιμοποιείται κυρίως για την δημιουργία εφαρμογών με σκοπό την εποπτεία και τον έλεγχο συστημάτων SCADA και εφαρμογών HMI. Μέσα από τις εφαρμογές αυτές ο εκάστοτε χρήστης θα μπορεί να παρακολουθεί σε γραφικό περιβάλλον και με αποδοτικό τρόπο οποιοδήποτε κομμάτι της παραγωγικής διαδικασίας μίας βιομηχανικής μονάδος. Το λογισμικό SIMATIC WinCC χρησιμοποιεί έναν SQL Server της Microsoft, για την ορθή και εύκολη καταγραφή των δεδομένων που αντλεί από το σύστημα SCADA, καθώς και VBScript και ANSI C, για τον προγραμματισμό των εφαρμογών που το απαρτίζουν. Συνοπτικά, μερικές από τις δυνατότητες που μπορεί να προσφέρει το SIMATIC WinCC είναι:

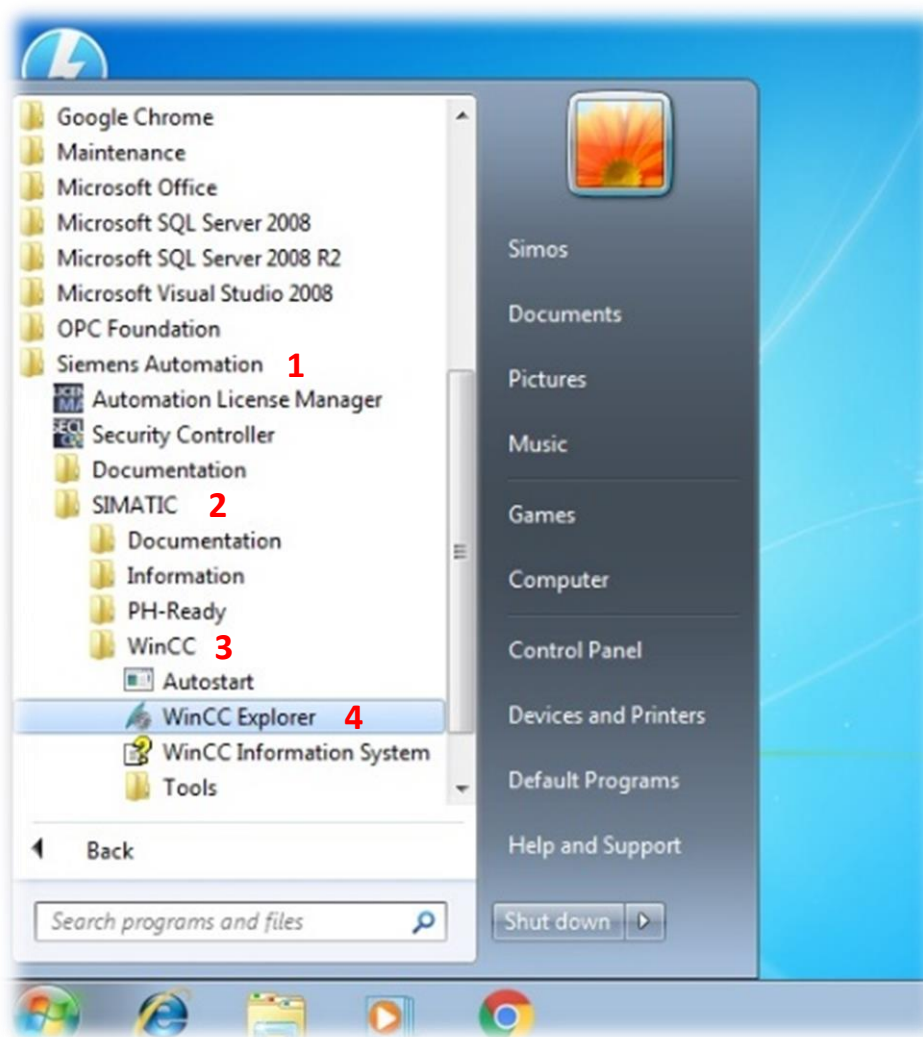
- Επιτρέπει στον χρήστη να παρακολουθεί μία διεργασία. Η διεργασία αυτή απεικονίζεται γραφικά σε μία οθόνη και η κατάστασή της ενημερώνεται κάθε φορά που τροποποιείται η διεργασία.
- Επιτρέπει στο χρήστη να ελέγξει την διαδικασία που παρακολουθεί. Για παράδειγμα μπορεί να τροποποιήσει την επιθυμητή τιμή μίας παραμέτρου ή να ανοίξει μία βαλβίδα μέσω του γραφικού περιβάλλοντος.
- Μία ένδειξη συναγερμού (Alarm) μπορεί να προειδοποιήσει το χρήστη για κρίσιμες εξελίξεις που σχετίζονται με μία διεργασία. Για παράδειγμα αν η τιμή κάποιας μεταβλητής ξεπεράσει κάποιο άνω όριο τότε υπάρχει η δυνατότητα εμφάνισης ενός προειδοποιητικού παραθύρου.
- Κατά την διάρκεια λειτουργίας του WinCC, οι παράμετροι μίας διεργασίας μπορούν, είτε να τυπωθούν, είτε να αποθηκευτούν ηλεκτρονικά σε κάποιο αρχείο για τη δημιουργία κάποιου αρχείου παρακολούθησης της όλης πορείας της διεργασίας.

Άλλες πιο ειδικευμένες δυνατότητες του SIMATIC WinCC είναι:

- Η διασύνδεσή του με εφαρμογές Πληροφοριακών Συστημάτων όπως είναι εφαρμογές ERP της εταιρίας SAP για παράδειγμα ή με εφαρμογές λογιστικών φύλλων όπως είναι το Microsoft Excel, όπως επίσης και με οποιαδήποτε εφαρμογή απαιτεί ο χρήστης αφού το περιβάλλον διασύνδεσης είναι ανοιχτό.
- Η δημιουργία κατανεμημένων συστημάτων SCADA με τη χρήση της ενσωματωμένης τεχνολογίας του διαδικτύου, όπως επίσης και συστημάτων αρχιτεκτονικής Client – Server.

4.2 Εκκίνηση του λογισμικού WinCC 7.2

Έχοντας υλοποιήσει επιτυχώς, ακολουθώντας κατά γράμμα τα βήματα που αναφέρθηκαν στην προηγούμενη ενότητα, μπορούμε πλέον να έχουμε μία πρώτη επαφή με το λογισμικό SIMATIC WinCC 7.2. Για να το εκκινήσουμε, σε λειτουργικό σύστημα Windows 7, αρκεί απλά να πραγματοποιήσουμε τα βήματα, όπως αυτά εμφανίζονται στην *εικόνα 4.2.1*



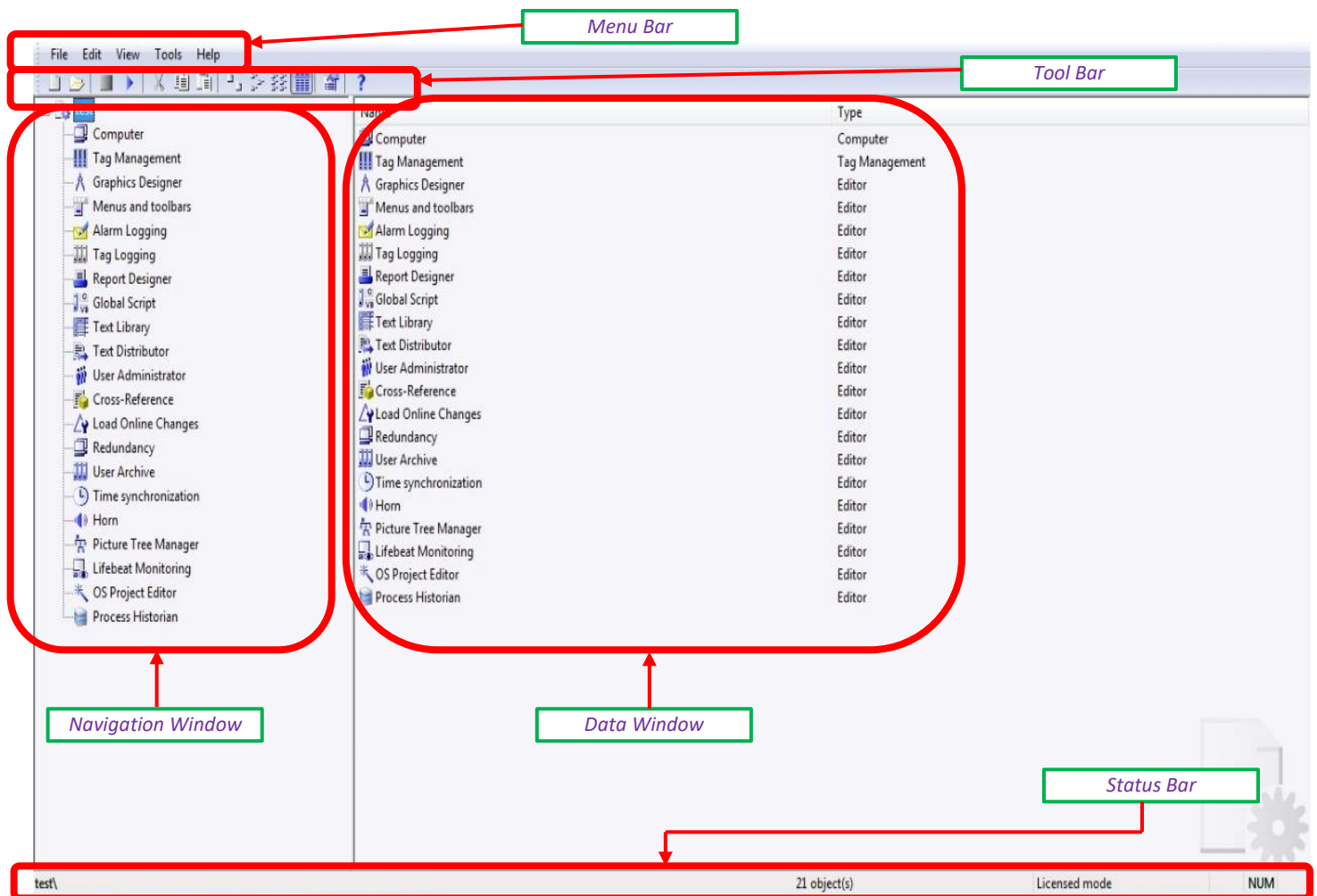
Εικόνα 4.2 1

4.3 Το περιβάλλον εργασίας του SIMATIC WinCC 7.2

Αφού εκκινήσουμε το λογισμικό SIMATIC WinCC, όπως υποδείξαμε παραπάνω, μας ανοίγει αυτόματα το πρόγραμμα **WinCC Explorer**, όπου αποτελεί και το κύριο πρόγραμμα δημιουργίας και διαμόρφωσης οποιασδήποτε εφαρμογής επιθυμούμε να υλοποιήσουμε. Με το πρόγραμμα **WinCC Explorer** μας δίνονται δυνατότητες όπως:

- Δημιουργίας μίας εφαρμογής
- Ανοίγματος και διαχείρισης μίας ήδη υπάρχουσας εφαρμογής
- Ανοίγματος διαφόρων βοηθητικών προγραμμάτων διαμόρφωσης μίας εφαρμογής

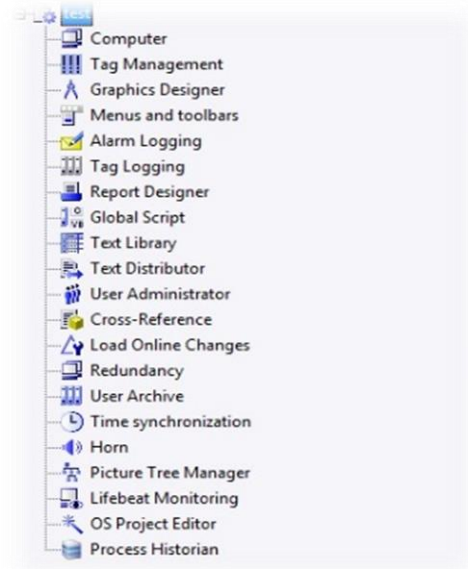
Όπως διακρίνουμε και στην *εικόνα 4.3.1*, το βασικό περιβάλλον εργασίας του WinCC Explorer διαχωρίζεται σε επιμέρους τμήματα, από τα οποία μπορούμε να διαχειριστούμε γρήγορα και αξιόπιστα την εφαρμογή όπου εποπτεύουμε. Βασικότερα στοιχεία του προγράμματος για έναν χρήστη μπορούν να χαρακτηριστούν το **τμήμα πλοήγησης (Navigation Window)** καθώς και το **τμήμα δεδομένων (Data Window)**. Παρακάτω παρουσιάζεται μία συνοπτική ανάλυση των τμημάτων αυτών.



Εικόνα 4.3.1

4.3.1 Τμήμα Πλοήγησης (Navigation Window)

Το τμήμα ή το παράθυρο πλοήγησης αποτελεί ένα βασικό κομμάτι στην δημιουργία εφαρμογών εποπτείας. Μέσα σε αυτό περιέχονται όλα τα βοηθητικά προγράμματα και οι λειτουργίες που μπορούν να υλοποιηθούν μέσω του WinCC Explorer. Παράλληλα, μέσω του τμήματος πλοήγησης, παρέχεται η δυνατότητα στον χρήστη μίας γρήγορης πρόσβασης σε όλα τα δεδομένα της εφαρμογής που έχει δημιουργηθεί. Ο χειρισμός του είναι απλός και εύκολος, παρόμοιος με τον χειρισμό του windows explorer σε ένα απλό υπολογιστικό σύστημα.



Το Τμήμα Πλοήγησης του WinCC

4.3.2 Τμήμα Δεδομένων (Data Window)

Το τμήμα δεδομένων λειτουργεί ως παράθυρο απεικόνισης των προγραμμάτων που λαμβάνουν χώρα στο τμήμα πλοήγησης του λογισμικού. Συγκεκριμένα, το τμήμα δεδομένων εμφανίζει τις παραμέτρους και τα στοιχεία των βοηθητικών προγραμμάτων και των λειτουργιών, εφόσον αυτές επιλεγούν από το τμήμα πλοήγησης. Έτσι, με ένα απλό διπλό κλικ στα στοιχεία που εμφανίζονται στο τμήμα αυτό, μπορούμε εύκολα να διαχειριστούμε αλλά και να επεξεργαστούμε οποιοδήποτε κομμάτι της εφαρμογής επιθυμούμε.

Name	Type
Computer	Computer
Tag Management	Tag Management
Graphics Designer	Editor
Menus and toolbars	Editor
Alarm Logging	Editor
Tag Logging	Editor
Report Designer	Editor
Global Script	Editor
Text Library	Editor
Text Distributor	Editor
User Administrator	Editor
Cross-Reference	Editor
Load Online Changes	Editor
Redundancy	Editor
User Archive	Editor
Time synchronization	Editor
Horn	Editor
Picture Tree Manager	Editor
Lifebeat Monitoring	Editor
OS Project Editor	Editor
Process Historian	Editor

Το Τμήμα Δεδομένων του WinCC

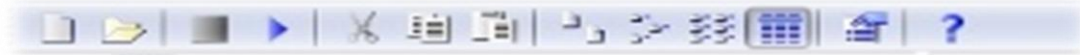
4.3.3 Μπάρα μενού (Menu Bar)



Η μπάρα μενού του WinCC Explorer περιέχει τις περισσότερες εντολές και λειτουργίες που δύναται να υλοποιήσει ο χρήστης με το πρόγραμμα. Παρέχει πρόσβαση σε μία ευρεία γκάμα λειτουργιών που απαιτούνται για την ρύθμιση της εποπτευόμενης εφαρμογής. Επίσης,

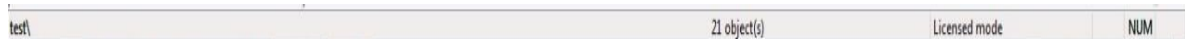
καθώς ο κέρσορας του ποντικιού μετακινηθεί πάνω σε μια εντολή, μας εμφανίζεται αυτόματα η επεξήγηση της λειτουργίας αυτής.

4.3.4 Μπάρα εργαλείων (Toolbar)



Χάρη την **μπάρα εργαλείων** μπορούμε να έχουμε μία γρήγορη και άμεση πρόσβαση σε βασικές εντολές του προγράμματος, όσον αφορά το χειρισμό μίας εφαρμογής. Η δημιουργία μίας νέας εφαρμογής, το άνοιγμα μίας ήδη υπάρχουσας ή ακόμα η ενεργοποίηση και η απενεργοποίησή της, μπορούν εύκολα να πραγματοποιηθούν, απλά με το πάτημα του ανάλογου κουμπιού. Η μπάρα εργαλείων είναι ολικά παραμετροποιήσιμη.

4.3.5 Μπάρα κατάστασης (Status Bar)



Η **μπάρα κατάστασης** παρέχει στον χρήστη γενικές πληροφορίες όπως το path της εφαρμογής που βρίσκεται ενεργοποιημένη στο πρόγραμμα και των αριθμό των αντικειμένων που υπάρχουν στην εφαρμογή αυτή.

4.4 Βοηθητικά προγράμματα σχεδίασης εφαρμογών στο WinCC

Όπως αναφέρθηκε και στην παραπάνω παράγραφο, το τμήμα πλοήγησης (navigation window) του λογισμικού SIMATIC WinCC μας παρέχει εκτός των άλλων, μια ευρεία γκάμα βοηθητικών προγραμμάτων και λειτουργιών, ικανά να σχεδιάσουν οποιαδήποτε εφαρμογή επιθυμεί ο χρήστης. Διαχωρίζοντας τες αναλόγως με τις επιλογές και τις δυνατότητες που μας παρέχουν, μπορούμε να χαρακτηρίσουμε ως πιο σημαντικές τις παρακάτω επιλογές:

- Computer
- Tag Managements
- Data types
- Editors



Βοηθητικά προγράμματα του λογισμικού WinCC

Στην συνέχεια ακολουθεί μία συνοπτική παρουσίαση των χαρακτηριστικών αλλά και του ρόλου που κατέχει η καθεμία από τις επιλογές αυτές, έχοντας πάντα ως γνώμονα την δημιουργία μίας εφαρμογής ικανής και αξιόπιστης για την εποπτεία ενός βιομηχανικού συστήματος SCADA.

4.4.1 Computer (Υπολογιστής)

Η λειτουργία αυτή καθορίζει, σε γενικές γραμμές, τις υπηρεσίες που θέλουμε να μας προσφέρει το λογισμικό SIMATIC WinCC. Πιο συγκεκριμένα, εδώ γίνεται η παραμετροποίηση των μεταβλητών που θέλουμε να εποπτεύουμε σε συνεχόμενη βάση καθώς επίσης και η καταγραφή των τιμών τους. Παράλληλα, παρέχεται δυνατότητα να γίνεται συγκεκριμένη αναφορά στην περίπτωση την οποία ξεπεραστεί κάποιο κρίσιμο όριο που εμείς έχουμε θέσει κτλ. Επιπλέον, μέσω της λειτουργίας αυτής, μπορούμε να έχουμε μία απευθείας συνεργασία του λογισμικού SIMATIC WinCC με διάφορα προγράμματα ανταλλαγής δεδομένων, όπως το Word, το Excel και το Autocad, τα οποία είναι ιδιαίτερα χρήσιμα σε περιπτώσεις διεξοδικής επεξεργασίας δεδομένων και κομψής παρουσίασής τους. Τέλος, στο μενού αυτό υπάρχουν διάφορες επιλογές για την παρουσίαση των γραφικών των εικόνων μας, τις οποίες και εκμεταλλευτήκαμε για την ολοκλήρωση των εφαρμογών μας.

4.4.2 Tag Managements (Διαχείριση Μεταβλητών)

Εδώ γίνεται ο καθορισμός του πρωτοκόλλου επικοινωνίας του υπολογιστή με την εκάστοτε περιφερειακή συσκευή (αισθητήρας, PLC, IED) και καθορίζονται οι μεταβλητές του συστήματος. Οι εν λόγω μεταβλητές διακρίνονται σε **εξωτερικές μεταβλητές (external ή proccess tags)** και **εσωτερικές μεταβλητές (internal tags)**. Οι **εξωτερικές μεταβλητές** πρόκειται στην ουσία μεταβλητές στην μνήμη δεδομένων της συνδεδεμένης περιφερειακής συσκευής, ενώ **οι εσωτερικές** χρησιμοποιούνται για διάφορες λειτουργίες εντός του λογισμικού. Όπως είναι εύκολα αντιληπτό, πρόκειται για ένα από τα πιο βασικά και πολύπλοκα τμήματα του SIMATIC WinCC, στο οποίο διαμορφώνεται η βασική δομή του βιομηχανικού συστήματος SCADA.

4.4.3 Data Types (Τύποι Μεταβλητών)

Αναφερόμενοι στις μεταβλητές που χρησιμοποιούνται από το λογισμικό, παρακάτω γίνεται μία εκτενής αναφορά σε όλους τους δυνατούς τύπους μεταβλητών που χρησιμοποιεί το SIMATIC 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

Αξίζει βεβαίως να αναφέρουμε ότι, κάθε μεταβλητή μπορεί να λάβει συγκεκριμένους τύπους δεδομένων, ανάλογα βεβαίως με το είδος και τη λειτουργία που επιτελεί.

4.4.4 Editors

Για την ευκολότερη κατανόηση τη χρήσης του, το βοηθητικό πρόγραμμα editor αποτελείται από **πέντε βασικά υποτμημάτα**:

4.4.4.1 Graphics Designer

Πρόκειται για το σχεδιαστικό κομμάτι 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 εμποτεύσει το σύστημα με μεγάλη ακρίβεια και λεπτομέρεια που πολύ δύσκολα θα είχε αλλιώς τη δυνατότητα να κάνει.

4.4.4.2 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.

4.4.4.3 Alarm Logging

Ό,τι υπέρβαση ορίων στις τιμές μεταβλητών, τα οποία έχουν τεθεί από τον χρήστη του συστήματος, παρατηρείται κατά τη λειτουργία του συστήματος καταγράφεται εδώ. Ως εκ τούτου, ο ρόλος του υποτμήματος 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.

4.4.4.4 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.

4.4.4.5 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.

Κεφάλαιο 5

Δημιουργία Εφαρμογής με το
SIMATIC WinCC 7.2

5.1 Μεθοδολογία Δημιουργίας Εφαρμογής

Στην προηγούμενο κεφάλαιο έγινε μία εκτενής αναφορά στα βασικά εργαλεία της καρδιάς ενός συστήματος SCADA, στο λογισμικό WinCC. Επίσης πραγματοποιήθηκε ανάλυση στα βοηθητικά προγράμματα και τις λειτουργίες όπου μας καθιστούν ικανούς να δημιουργήσουμε μία εφαρμογή για την εποπτεία και την παρακολούθηση μίας βιομηχανικής μονάδος.

Στις παρακάτω ενότητες θα γίνει μία αναλυτική παρουσίαση των βημάτων που ακολουθήθηκαν για την δημιουργία και την παραμετροποίηση των εφαρμογών που υλοποιήθηκαν στην παρούσα πτυχιακή εργασία. Με αυτόν τον τρόπο θα μπορέσουμε να επιτύχουμε μία ολοκληρωμένη και πιο ομαλή κατανόηση της λειτουργίας του λογισμικού WinCC.

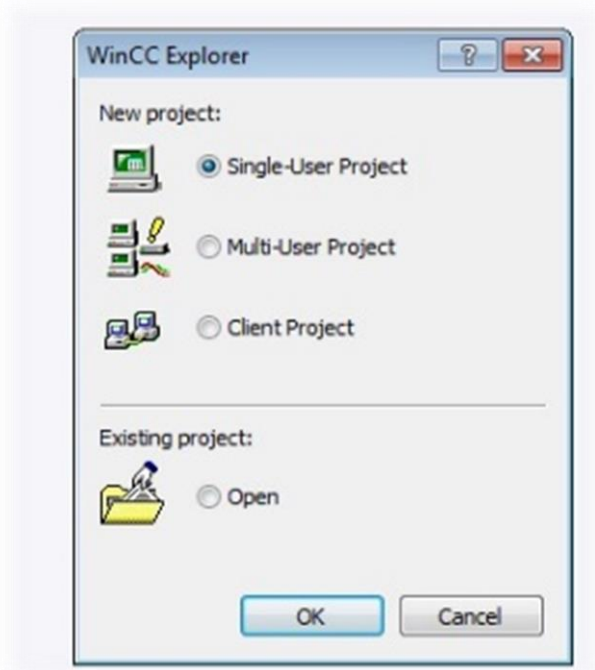
Για να δημιουργήσουμε μία νέα εφαρμογή στο WinCC, ακολουθούμε απαραίτητως τα παρακάτω βασικά βήματα :

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

5.2 Αναλυτική παρουσίαση βημάτων δημιουργίας project

5.2.1 Εκκίνηση του WinCC

Ακολουθώντας λοιπόν τα βήματα που αναφέραμε παραπάνω, κατά την εκκίνηση του λογισμικού, μας εμφανίζεται αυτόματα ένα αναδυόμενο παράθυρο όπου είτε επιλέγουμε το είδος της εφαρμογής (project) που επιθυμούμε να δημιουργήσουμε είτε ανοίγουμε μία ήδη αποθηκευμένη εφαρμογή (εικόνα 5.2.1.1). Εδώ αξίζει να αναφέρουμε ότι εάν κατά το κλείσιμο του προγράμματος δεν απενεργοποιηθεί η εφαρμογή που εργαζόμαστε, το παράθυρο αυτό μπορεί να μην μας εμφανιστεί. Όπως παρατηρούμε, μας εμφανίζει τρεις επιλογές δημιουργίας εφαρμογής, τις **Single – User Project**, **Multi – User Project** και **Client Project**. Η επεξήγηση των επιλογών αυτών αναλύεται παρακάτω.



Εικόνα 5.2.1.1

5.2.1.1 Single – User Project

Η επιλογή **single – user project** παρέχει στον χρήστη την δυνατότητα να εργαστεί από έναν μόνο υπολογιστή, ο οποίος αναλαμβάνει ταυτόχρονα τον ρόλο του server του συστήματος, για την επεξεργασία των δεδομένων, και τον ρόλο ενός απλού εποπτικού σταθμού. Μέσω της εφαρμογής αυτής, δεν είναι δυνατή η προέκταση του εποπτικού συστήματος σε περισσότερους υπολογιστές.

5.2.1.2 Multi – User Project

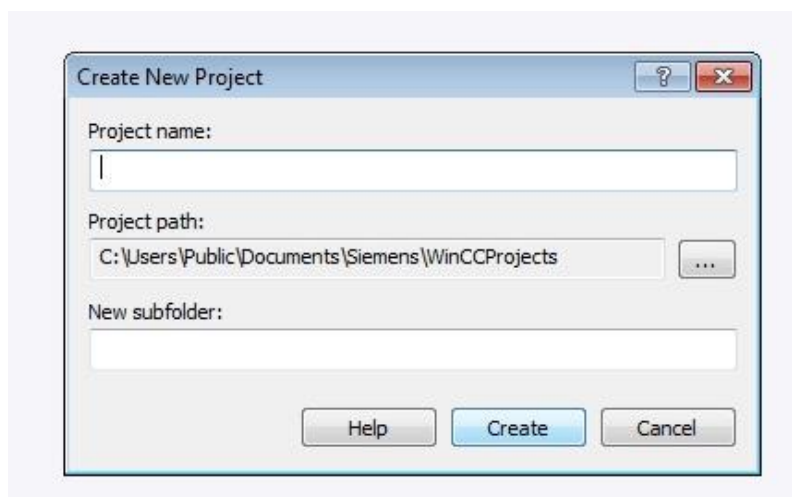
Η επιλογή αυτή έρχεται ως συμπλήρωμα της προηγούμενης, επιτρέποντας στον χρήστη την διαχείριση μίας εφαρμογής από διαφορετικούς υπολογιστές ταυτόχρονα. Με το **multi – user project** μπορείς να έχεις ένα σύστημα πολλαπλών χρηστών με έναν ή περισσότερους servers ταυτόχρονα. Με αυτόν τον τρόπο, ο εκάστοτε χρήστης μπορεί να έχει πρόσβαση σε δεδομένα αλλά και γενικά στο εποπτικό σύστημα, το οποίο μπορεί να βρίσκεται σε διαφορετικούς servers.

5.2.1.3 Client Project

Η επιλογή αυτή βρίσκει χρήση **μόνο** στην περίπτωση εφαρμογής **multi – user**. Όπως αναφέραμε παραπάνω, λόγω της ύπαρξης αρκετών χρηστών, είτε εάν η εφαρμογή βρίσκεται εγκατεστημένη σε έναν είτε σε πολλούς server, ο διαχειριστής θα πρέπει να ορίσει τους χρήστες όπου θα μπορούν να έχουν πρόσβαση στην εφαρμογή. Έτσι, με την επιλογή **client project** μπορεί να οριστούν, εκτός από τον αριθμό των χρηστών, και τα δικαιώματα που μπορούν να έχουν πάνω στην εφαρμογή.

5.2.2 Δημιουργία εφαρμογής (project)

Για τις ανάγκες του παραδείγματος της πτυχιακής μας εργασίας, θα επιλέξουμε την δημιουργία εφαρμογής **single – user project**, χρησιμοποιώντας το υπολογιστικό μας σύστημα ως server και εποπτικού σταθμού παράλληλα. Κλικάροντας πάνω στην επιλογή αυτή και πατώντας το κουμπί **OK**, μας εμφανίζεται το παράθυρο **Create New Project** (εικόνα 5.2.2.1). Στο σημείο αυτό μπορούμε να δώσουμε ένα όνομα στην εφαρμογή που επιθυμούμε να δημιουργήσουμε, συμπληρώνοντας ανάλογα το πεδίο **Project Name**, καθώς και να διαλέξουμε την διαδρομή στην οποία θέλουμε να αποθηκευτεί, επιλέγοντάς την στο πεδίο **Project Path**. Αφού ολοκληρώσουμε τις ρυθμίσεις αυτές, είμαστε έτοιμοι να δημιουργήσουμε την εφαρμογή μας, πατώντας το κουμπί **Create**.

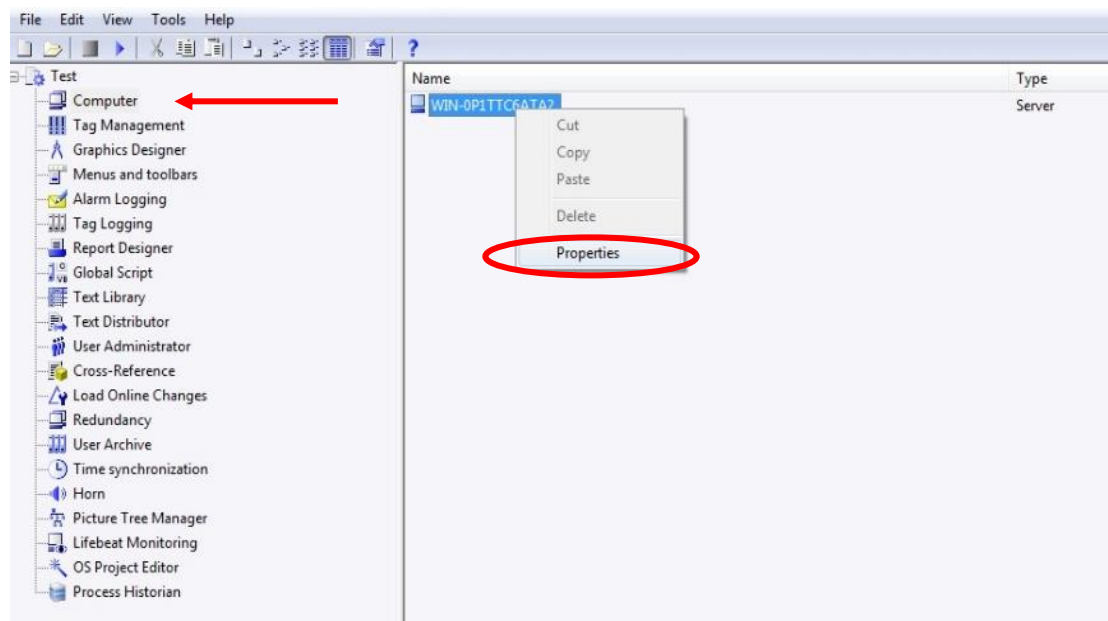


Εικόνα 5.2.2.1

5.2.3 Ρύθμιση υπολογιστή εφαρμογής

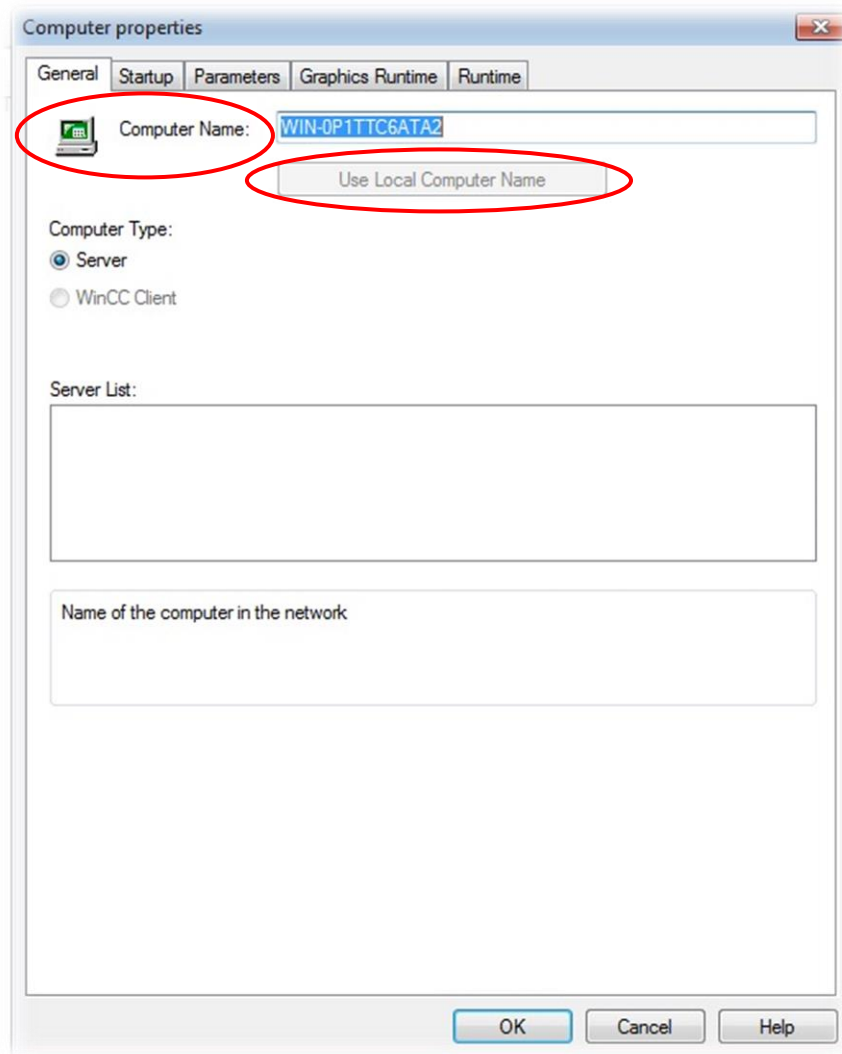
Έχοντας δημιουργήσει επιτυχώς την εφαρμογή (project) σύμφωνα με τα παραπάνω, το επόμενο μας βήμα είναι ο καθορισμός των ιδιοτήτων του **ηλεκτρονικού υπολογιστή (computer)** στον οποίο θα είναι ενεργή η εφαρμογή μας. Αξίζει να επισημάνουμε ότι λόγω του **single – user project** που έχουμε δημιουργήσει για την ανάγκες του παραδείγματός μας, η ρύθμιση γίνεται μόνο στον υπολογιστή που υπάρχει εγκατεστημένη η εφαρμογή. Στην περίπτωση δημιουργίας multi – user project, θα πρέπει να γίνει ρύθμιση των ιδιοτήτων του κάθε υπολογιστή ξεχωριστά.

Για να αποκτήσουμε πρόσβαση στις ιδιότητες του υπολογιστή, κάνουμε **διπλό κλικ** στην επιλογή **Computer**, όπου βρίσκεται στην **περιοχή πλοήγησης (navigation window)**, όπως αυτό γίνεται εμφανές στην **εικόνα 5.2.3.1**. Έπειτα, στην **περιοχή δεδομένων (data window)** όπου εμφανίζεται πλέον το επιλεγόμενο υπολογιστικό σύστημα, κάνουμε πάνω του **δεξί κλικ** και διαλέγουμε την επιλογή **Properties**.



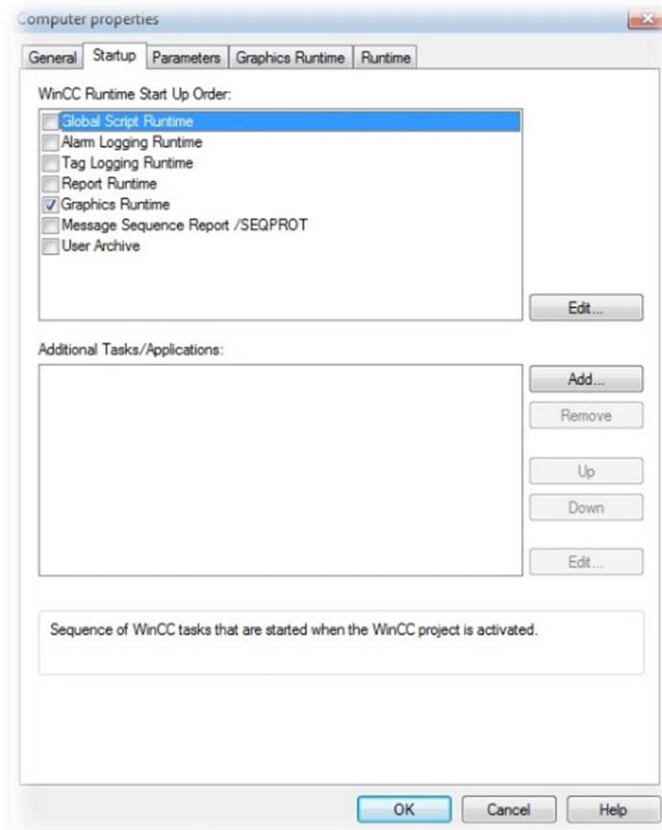
Εικόνα 5.2.3.1

Στο παράθυρο το οποίο μας ανοίγεται (εικόνα 5.2.3.2), πρώτο μας μέλημα είναι να ελέγξουμε εάν το όνομα του υπολογιστή είναι αυτό που επιθυμούμε. Για να πραγματοποιηθεί αυτό, αρκεί να κοιτάξουμε το πεδίο **Computer Name** όπου βρίσκεται στην καρτέλα **General**. Το όνομα του υπολογιστή μπορεί επίσης να ρυθμιστεί και αυτόματα, παίρνοντας το όνομα του υπολογιστικού συστήματος όπου είναι εγκατεστημένο, κάνοντας απλά **αριστερό κλικ** στην επιλογή **Use Local Computer Name**. Τελευταία μας ενέργεια είναι να καθορίσουμε τον ρόλο του υπολογιστή στο εποπτικό μας σύστημα. Συγκεκριμένα, κάνοντας κλικ στην ανάλογη επιλογή, μπορούμε να ορίσουμε τον υπολογιστή μας είτε σε ρόλο **client** είτε σε ρόλο **server**. Στο παράδειγμα που παρουσιάζεται, ο υπολογιστή μας λαμβάνει αυτόματα την θέση του **server** λόγω του **single – user project** που επιλέξαμε κατά την δημιουργία της εφαρμογής. Σε **multi – user projects**, μπορούμε να δούμε την λίστα των διαθέσιμων clients ή servers από την λίστα που εμφανίζεται στο κάτω μέρος του παραθύρου.



Εικόνα 5.2.3.2

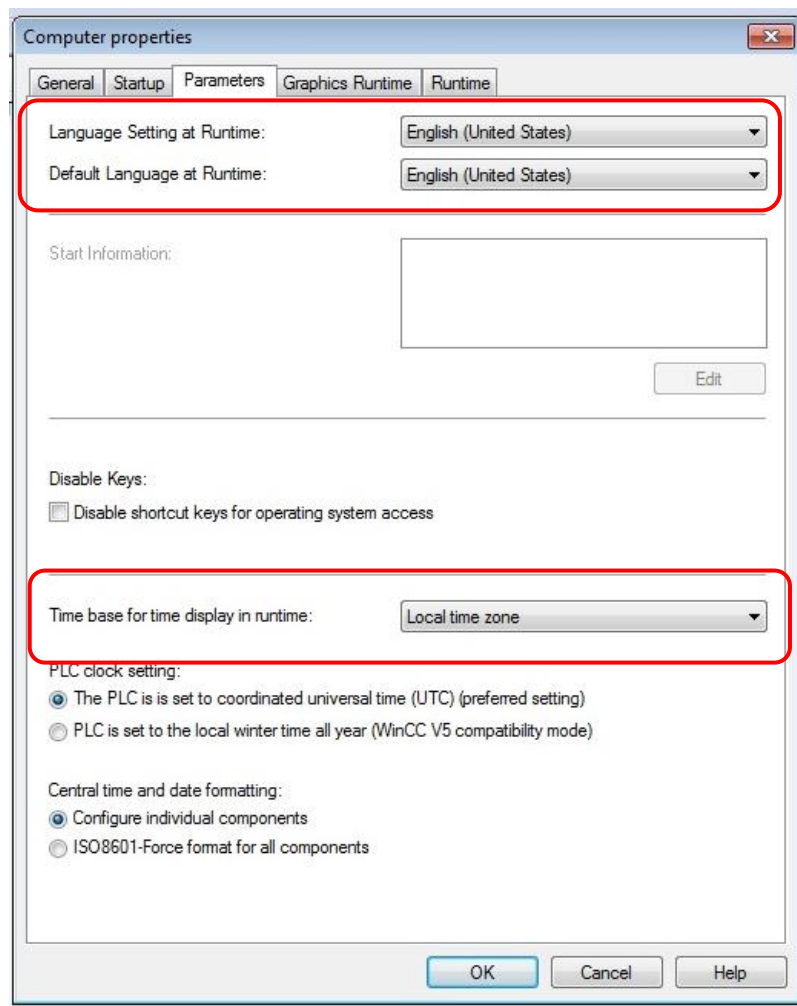
Επιπροσθέτως, από τις ιδιότητες του υπολογιστή, μας δίνεται η δυνατότητα να ορίσουμε εμείς τα βοηθητικά προγράμματα που επιθυμούμε να εκκινούν μαζί με την εφαρμογή μας. Ο ορισμός αυτός γίνεται από την καρτέλα **Startup** (εικόνα 5.2.3.3). Επιλέγοντάς την, βλέπουμε μία έτοιμη λίστα με τα υποστηριζόμενα προγράμματα τα οποία, κάνοντας απλά ένα κλικ πάνω τους και πατώντας το κουμπί Add, μπορούμε να τα ενεργοποιήσουμε ώστε να εκκινούν στο παρασκήνιο την εφαρμογής μας, υποβοηθώντας το στον εποπτικό του ρόλο.



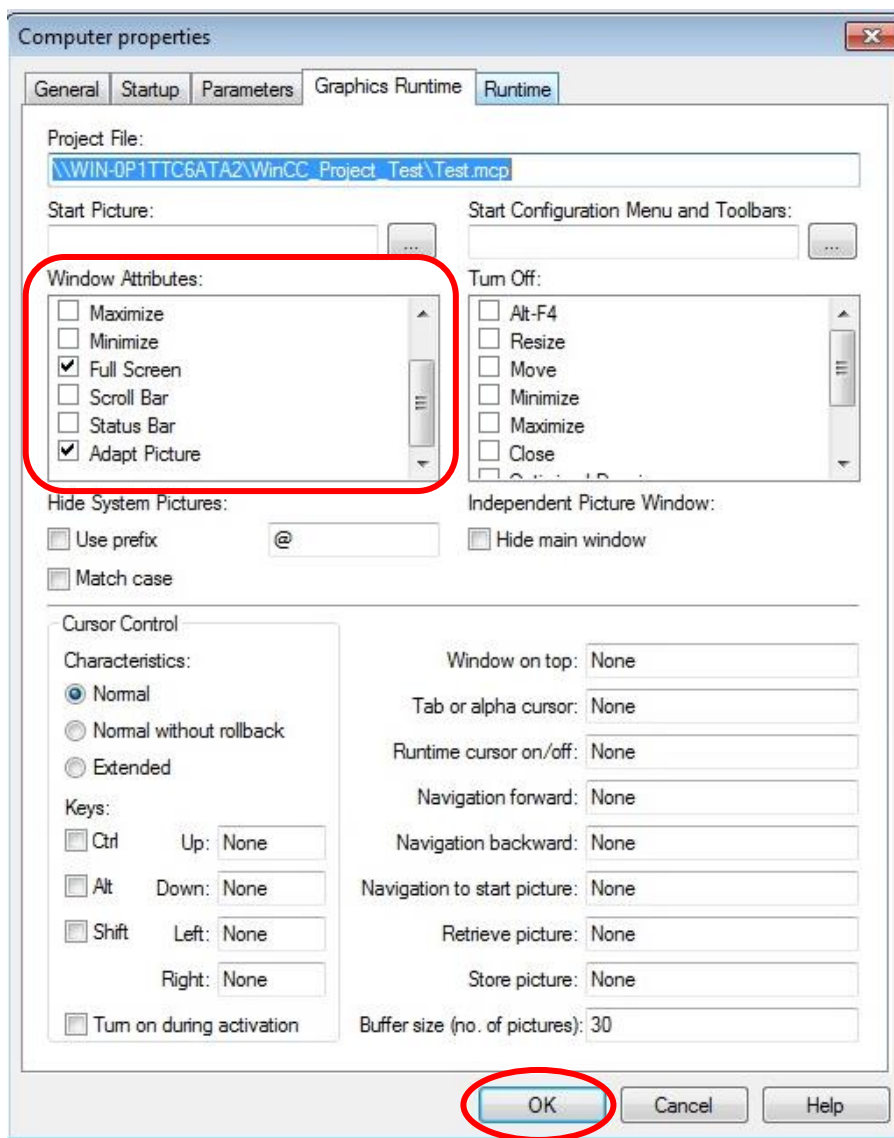
Εικόνα 5.2.3.3

Από τις επιλογές του μενού **Computer** μας δίνεται η δυνατότητα επίσης να χειριστούμε και πιο τοπικές λειτουργίες όπως είναι η **γλώσσα εισαγωγής** αλλά και διαχείρισης του προγράμματος όπως η **ζώνη ώρας** όπου επιθυμούμε να ακολουθήσουμε. Οι ρυθμίσεις αυτές πραγματοποιούνται από την καρτέλα **Parameters**. Όπως φαίνεται και στην *εικόνα 5.2.3.4*, έχουμε ορίσει σαν προεπιλεγμένη γλώσσα για την εφαρμογή μας την Αγγλική καθώς και η ζώνη ώρας που ακολουθούμε είναι η ίδια με αυτή που έχει οριστεί στο σύστημά μας.

Εν συνεχεία, για να ρυθμίσουμε την κατάσταση και τις παραμέτρους όπου θα τρέχει η εφαρμογή μας, θα πρέπει κάνουμε κλικ στην καρτέλα **Graphics Runtime** (εικόνα 5.2.3.5). Η καρτέλα αυτή περιέχει ποικίλες παραμέτρους, σχετικές με την εμφάνιση των εικόνων εργασίας στην οθόνη του υπολογιστή μας, την ρύθμιση της ανάλυσης των εικόνων αυτών, τον καθορισμό των τρόπων εξόδου από αυτή και άλλα πολλά. Στο παράδειγμά μας, θα χρειαστεί να ρυθμίσουμε μόνο δύο βασικά πράγματα που αφορούν κυρίως την εμφάνιση και την παρουσίαση της εφαρμογής: την εμφάνιση του περιβάλλοντος εργασίας σε όλο το εύρος της διαθέσιμης οθόνης μας και την σταθερή διάταξη και τοποθέτηση των αντικειμένων της εφαρμογής. Οι ρυθμίσεις αυτές μπορούν να υλοποιηθούν κάνοντας **κλικ** στις επιλογές **Full Screen** και **Adapt Picture** από το πλαίσιο **Window Attributes**, όπως αυτές παρουσιάζονται στην παραπάνω εικόνα. Για να αποθηκευτούν οι αλλαγές μας, κάνουμε κλικ στο κουμπί **OK**.



Εικόνα 5.2.3.4

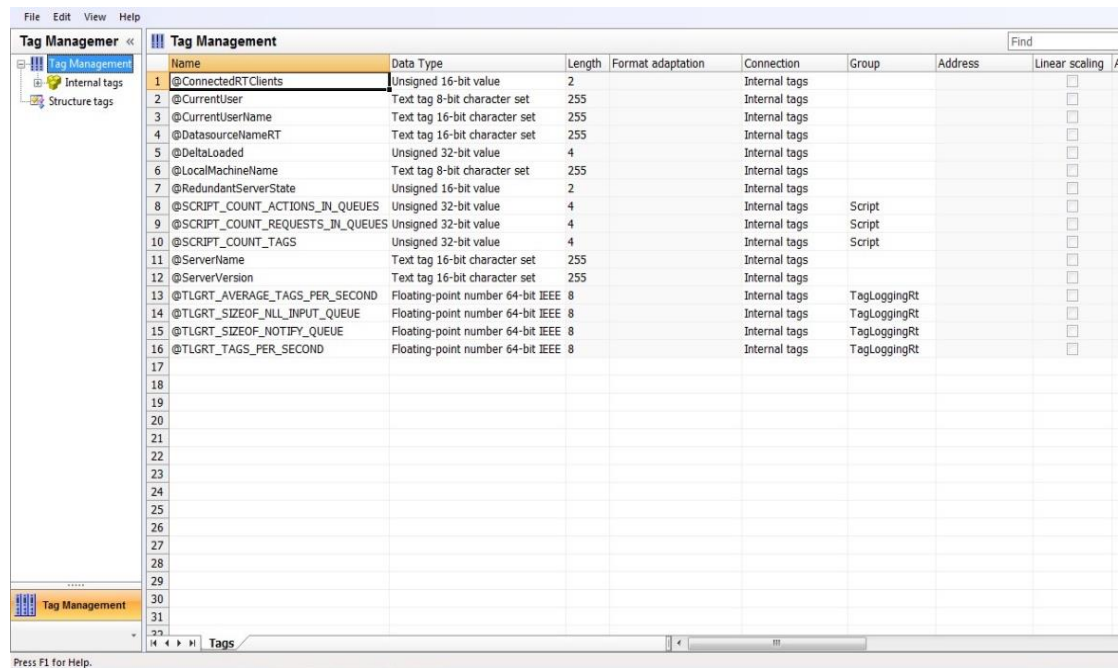


Εικόνα 5.2.3 5

5.2.4 Επιλογή και εγκατάσταση driver

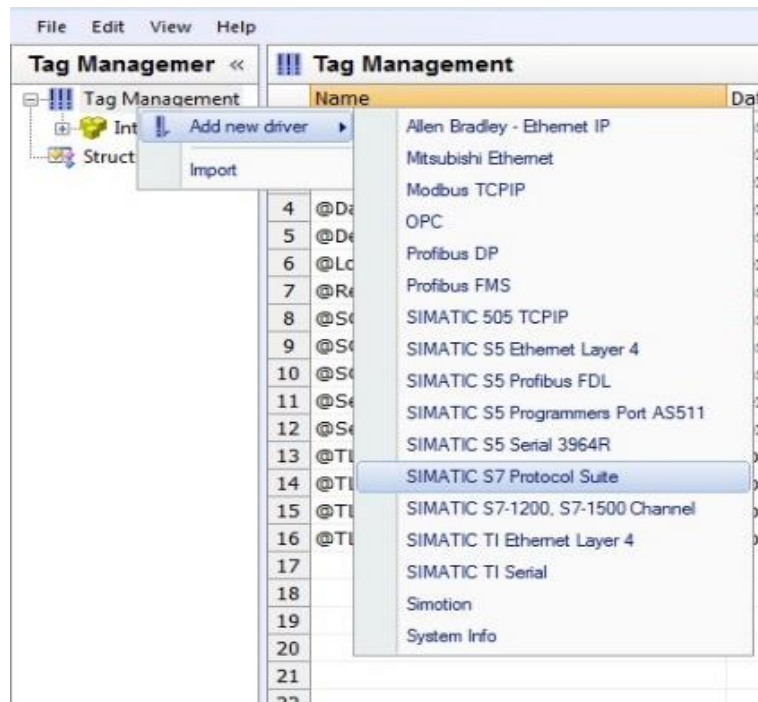
Κάνοντας ήδη τα απαραίτητα βήματα οργάνωσης του τρόπου παρουσίασης της εφαρμογής μας, επόμενο μας βήμα, βάση της μεθοδολογίας που αναφέρθηκε παραπάνω, είναι η επιλογή και η **εγκατάσταση των προγραμμάτων οδήγησης (driver)** των περιφερειακών συσκευών που βρίσκονται συνδεδεμένες στο βιομηχανικό σύστημα SCADA. Όπως γίνεται εύκολα αντιληπτό, η επιλογή και η εγκατάσταση των drivers αποτελεί κομβικό βήμα για την ορθή και απροβλημάτιστη λειτουργία του συστήματός μας. Μία λάθος επιλογή παραμέτρων ενός αισθητήρα ή PLC μπορεί να οδηγήσει σε προβληματική λειτουργία και, κατά συνέπεια, σε λάθος εποπτεία και κατάρρευση του συστήματος.

Για να ορίσουμε λοιπόν τις παραμέτρους σύνδεσης των περιφερειακών συσκευών αρκεί να κάνουμε **διπλό κλικ** στην λειτουργία **Tag Management** από την **περιοχή πλοήγησης (navigation window)**. Στο νέο παράθυρο στο οποίο μας εμφανίζεται (εικόνα 5.2.4.1), θα χρειαστεί να προσθέσουμε την ανάλογο πακέτο πρωτοκόλλων που αφορούν την σύνδεση του λογισμικού με τις περιφερειακές μας συσκευές.



Εικόνα 5.2.4.1

Για την εκπόνηση του παραδείγματος της πτυχιακής μας εργασίας, θα χρησιμοποιήσουμε το πρωτόκολλο **SIMATIC S7 Protocol Suite** και η επικοινωνία μεταξύ περιφερειακής συσκευής και υπολογιστή έχει πραγματοποιηθεί μέσω της διεπαφής **MPI**. Για να πραγματοποιηθεί η σύνδεση αυτή, κάνουμε **δεξί κλικ** στην επιλογή **Tag Management** και επιλέγουμε το ανάλογο πρωτόκολλο από το μενού **Add a new driver**, όπως αυτό εμφανίζεται στην **εικόνα 5.2.4.2**.

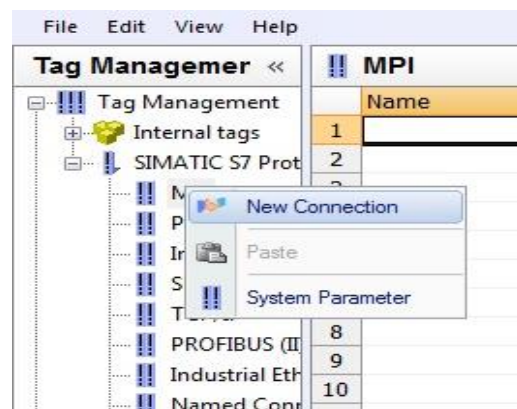


Εικόνα 5.2.4.2

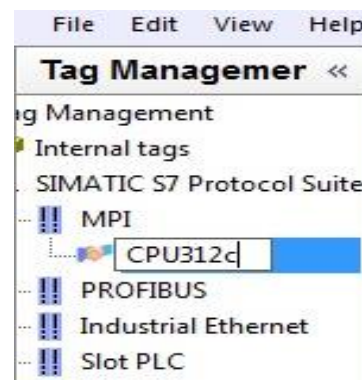
Έπειτα, κάνοντας κλικ στο **SIMATIC S7 Protocol Suite**, στις επιλογές που μας εμφανίζονται, κάνουμε δεξί κλικ στην διεπαφή **MPI** και επιλέγουμε την **δημιουργία νέας σύνδεσης (New Connection)** (εικόνα 5.2.4.3).

Αυτόματα δημιουργείται μία νέα επικοινωνία η οποία για τους λόγους του παραδείγματός μας, θα ονομαστεί **CPU312c** (εικόνα 5.2.4.4).

Για να ρυθμίσουμε τις παραμέτρους λειτουργίας της επικοινωνίας που μόλις δημιουργήσαμε, αρκεί απλά να κάνουμε ένα **δεξί κλικ** πάνω της και να επιλέξουμε το μενού **Connection Parameters** (εικόνα 5.2.4.5).

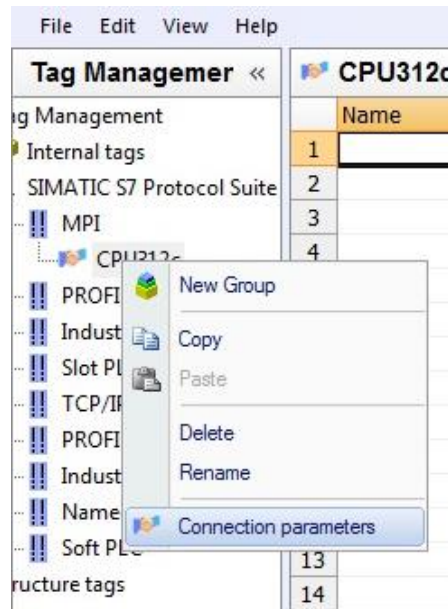


Εικόνα 5.2.4.3

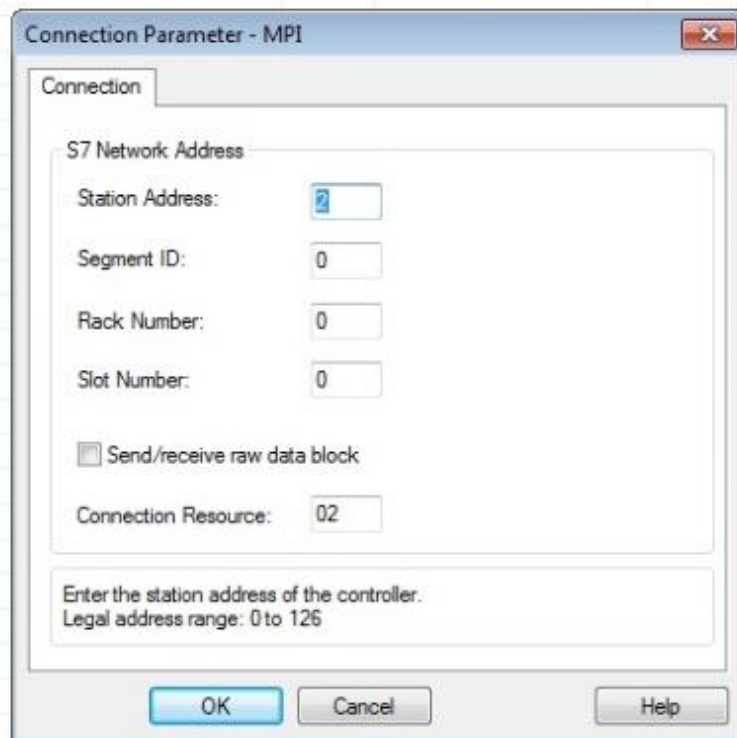


Εικόνα 5.2.4.4

Στο αναδυόμενο παράθυρο (**Connection Parameters**) όπου μας εμφανίζεται εισάγουμε τις κατάλληλες τιμές στις ανάλογες επιλογές. Στην περίπτωση μας, στην παράμετρο **Station Address** εισάγουμε την **τιμή 2** και ορίζουμε την επιλογή **Connection Resource** ίση με **2** και αυτή. Τις υπόλοιπες παραμέτρους τις αφήνουμε με τις αρχικές τους ρυθμίσεις (εικόνα 5.2.4.6). Για να αποθηκευτούν οι αλλαγές μας, κάνουμε κλικ στο κουμπί **OK**.



Εικόνα 5.2.4.5



Εικόνα 5.2.4.6

5.2.5 Ορισμός των μεταβλητών (tags)

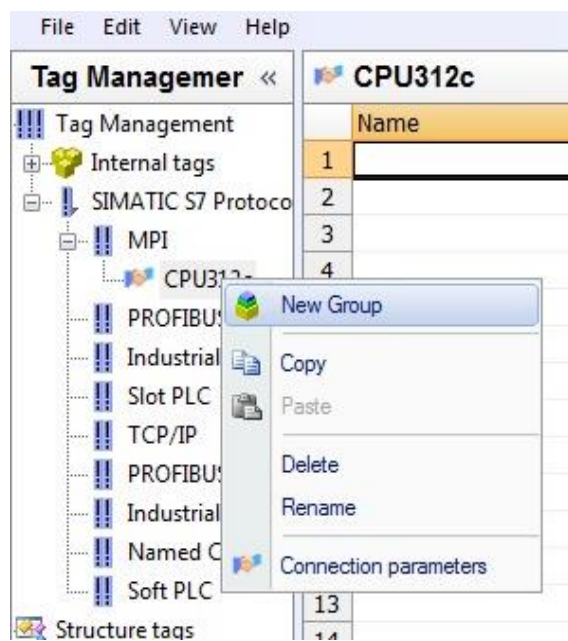
Η εποπτεία ενός βιομηχανικού συστήματος SCADA, μέσω των διαφόρων περιφερειακών συσκευών (PLC's, RTU's) και αισθητήρων, έχει ως αποτέλεσμα την αποθήκευση και αποστολή δεδομένων και συμβάντων από το σύστημα προς το λογισμικό μας. Τα δεδομένα οπύ αντλούνται από αυτό, με σκοπό την καλύτερη και ασφαλέστερη διαχείρισή τους, περνάνε μέσα στο WinCC με την μορφή των **μεταβλητών (tags)**. Μία μεταβλητή χαρακτηρίζεται από μία διεύθυνση δεδομένων, η οποία χρησιμοποιείται για την άμεση επικοινωνία και μεταφορά των τιμών των περιφερειακών συσκευών, και ένα συμβολικό όνομα, όπου απαιτείται για την λειτουργία της εφαρμογής που έχουμε δημιουργήσει.

5.2.5.1 Κατηγορίες Μεταβλητών (tags)

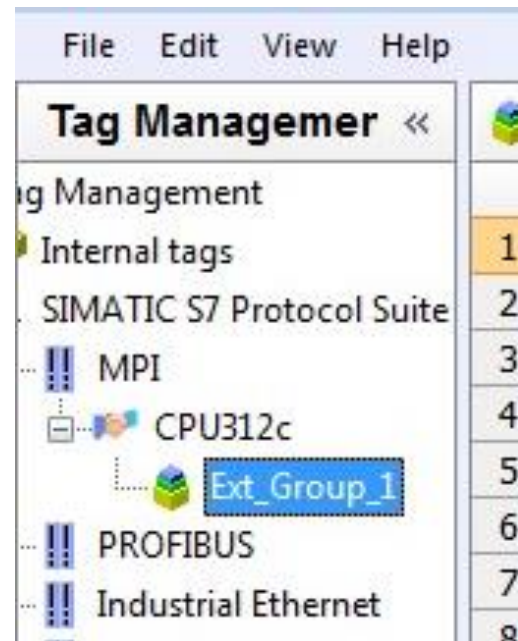
Οι μεταβλητές αυτές μπορούν να κατηγοριοποιηθούν σε δύο βασικές κατηγορίες: στις **εξωτερικές μεταβλητές (external tags ή process tags)** και στις **εσωτερικές μεταβλητές (internal tags)**. **Εξωτερικές** ονομάζονται οι μεταβλητές όπου είναι υπεύθυνες για την μεταφορά τιμών και δεδομένων από την περιοχή δεδομένων μνήμης της εποπτικής συσκευής (PLC's, RTU's, IED's) προς το λογισμικό WinCC, όπου γίνεται η επεξεργασία και η εποπτεία τους. Από την αντίπερα όχθη, οι **εσωτερικές μεταβλητές** αφορούν κάποια τμήματα της μνήμης του WinCC, τα οποία χρησιμοποιούνται κυρίως για εσωτερικά θέματα του λογισμικού και για λειτουργίες προσομοίωσης. Ο ορισμός και των δύο τύπων μεταβλητών είναι απαραίτητος και εξίσου σημαντικός με την εγκατάσταση driver για την διασφάλιση της σωστής αναπαράστασης των παρατηρούμενων εφαρμογών και παίζει σημαντικό ρόλο στη σωστή λειτουργία του SCADA. Στο παράδειγμα που ακολουθεί παρακάτω, τόσο για τις εσωτερικές όσο και για τις εξωτερικές, θα πραγματοποιηθεί **ομαδοποίηση των μεταβλητών (Tag Groups)**. Η κίνηση αυτή, λόγω της απλότητας της εφαρμογής μας, δεν αποφέρει κάποια αλλαγή στην λειτουργία της. Παρέχει όμως καλύτερη οργάνωση και ευκολότερη αναζήτηση των μεταβλητών.

5.2.5.1.1 Δημιουργία Εξωτερικών Μεταβλητών (Process Tags)

Από το μενού **Tag Management** όπου βρισκόμαστε, κάνουμε **δεξί κλικ** στην σύνδεση που έχουμε ήδη δημιουργήσει, ακολουθώντας τα βήματα που αναλύθηκαν στην παραπάνω ενότητα, και επιλέγουμε το μενού **New Group** (εικόνα 5.2.5.1.1.1). Έτσι δημιουργούμε μία νέα ομάδα, όπου μέσα της θα συμπεριληφθούν οι εξωτερικές μεταβλητές της εφαρμογής μας. Για τις ανάγκες της εργασίας μας, θα ονομάσουμε την ομάδα αυτή **Ext_Group_1** (εικόνα 5.2.5.1.1.2).

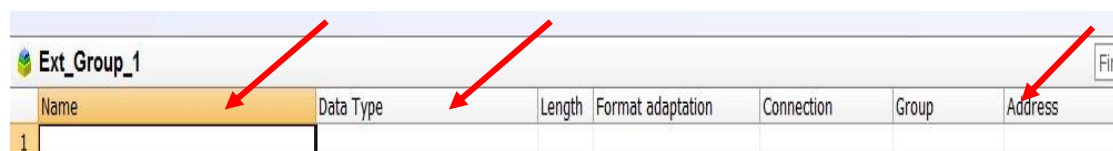


Εικόνα 5.2.5.1.1.1



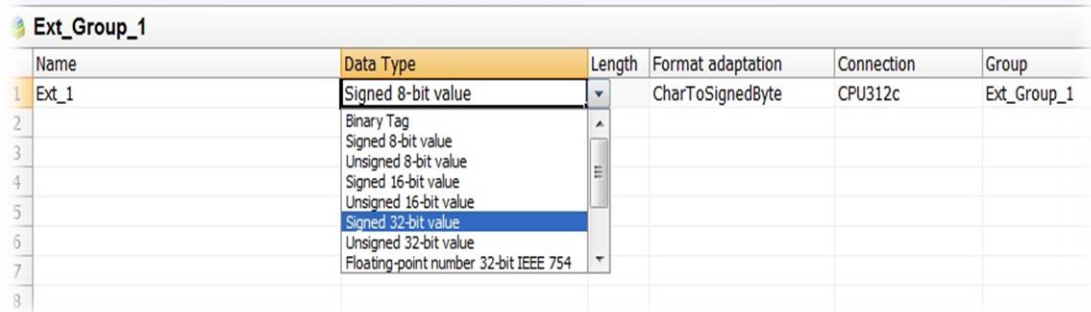
Εικόνα 5.2.5.1.1.2

Προχωρώντας στον ορισμό τους, θα χρησιμοποιήσουμε το **τμήμα δεδομένων (data window)** της οθόνης μας. Συγκεκριμένα, στα πεδία που έχουν δημιουργηθεί, όπως αυτό είναι φανερό στην εικόνα 5.2.5.1.1.3, θα συμπληρώσουμε τις κατάλληλες τιμές. Θα πρέπει να συμπληρωθούν απαραίτητα για την δημιουργία των εξωτερικών μεταβλητών τα πεδία **Name**, **Data Type** και **Address**.



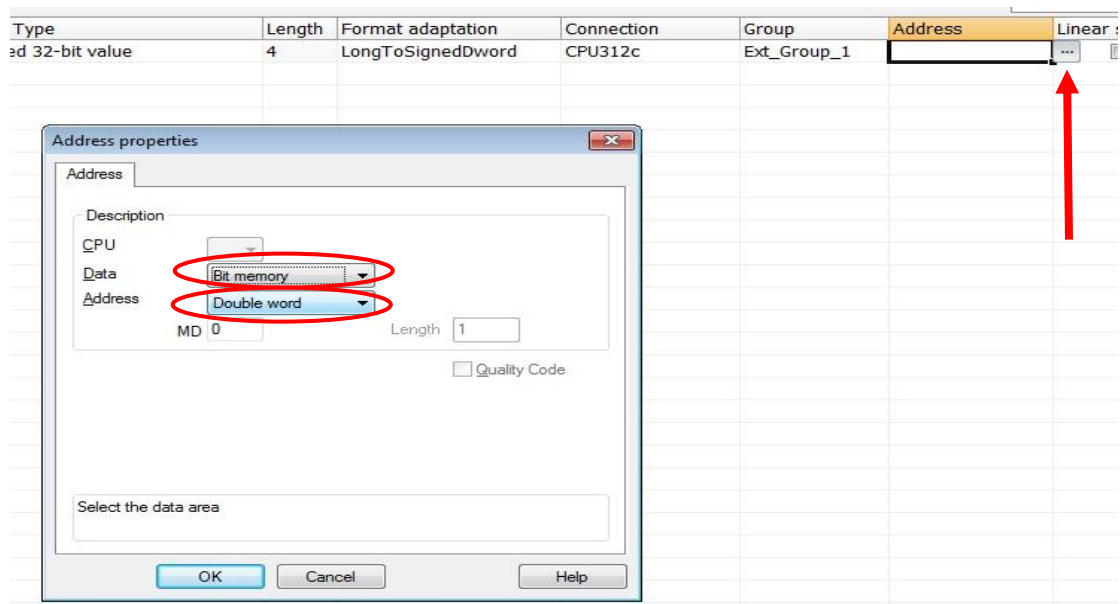
Εικόνα 5.2.5.1.1.3

Στο παράδειγμά μας, θα δημιουργήσουμε μία εξωτερική μεταβλητή με όνομα **Ext_1**. Εισάγοντας το όνομά της, παρατηρούμε ότι τα υπόλοιπα πεδία συμπληρώνονται αυτόματα. Παραμετροποιώντας την σύμφωνα με τις ανάγκες της εφαρμογής μας, στο πεδίο **Data Type** ορίζουμε το μήκος και το είδος των δεδομένων που αντλούνται από τις εποπτικές περιφερειακές συσκευές του συστήματος SCADA. Στο παράδειγμά μας, θα ορίσουμε ως Data Type την επιλογή **Signed 32 – bit value** (εικόνα 5.2.5.1.1.4).



Εικόνα 5.2.5.1.1.4

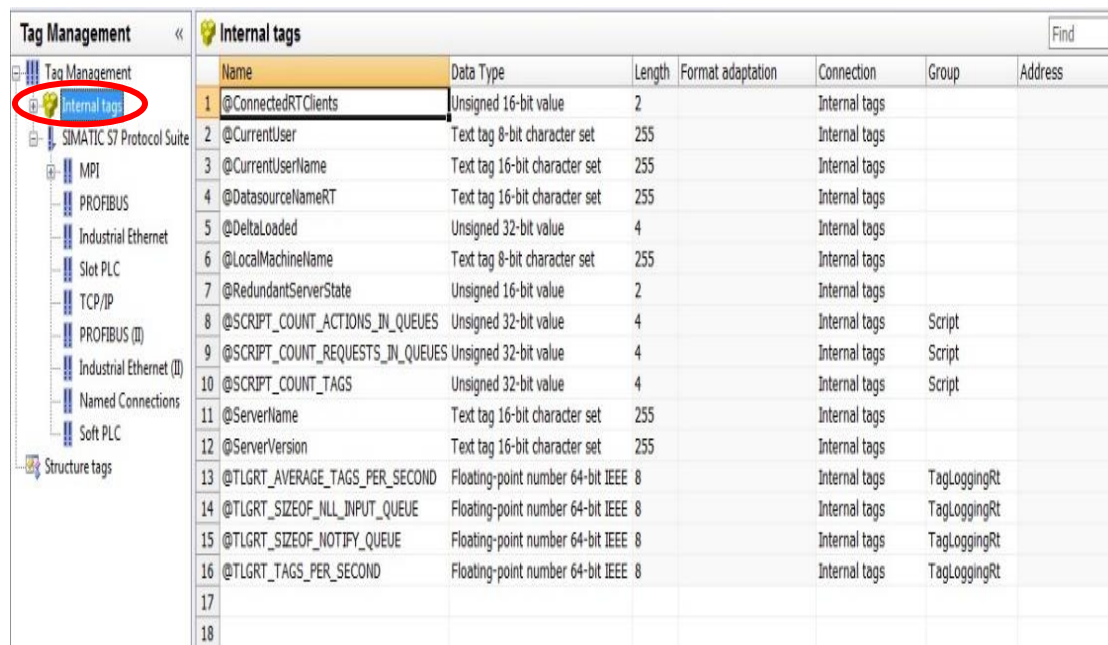
Παράλληλα, κάνοντας κλικ στο κουμπί όπου βρίσκεται δεξιά του πεδίου **Address**, μας εμφανίζεται ένα νέο αναδυόμενο παράθυρο. Στο μενού **Data**, επιλέγουμε σαν είδος δεδομένων το **Bit Memory** και κάνουμε κλικ στο κουμπί **OK** για να αποθηκευτούν οι αλλαγές μας (εικόνα 5.2.5.1.1.5). Στα υπόλοιπα πεδία διατηρούμε τις τιμές που ορίστηκαν αυτόματα κατά την δημιουργία της εξωτερικής μεταβλητής μας.



Εικόνα 5.2.5.1.1.5

5.2.5.1.2 Δημιουργία Εσωτερικών Μεταβλητών (InternalTags)

Η μέθοδος της δημιουργίας εσωτερικών μεταβλητών παρουσιάζει αρκετές ομοιότητες με την δημιουργία των εξωτερικών μεταβλητών που αναλύθηκε στην παραπάνω παράγραφο. Γι' αυτόν τον λόγο, ανατρέχουμε στο **τμήμα πλοήγησης (navigation window)** του μενού **Tag Management** και κάνουμε κλικ στην επιλογή **Internal Tags**, όπως εμφανίζεται στην *εικόνα 5.2.5.1.2.1*. Στο τμήμα δεδομένων του παραθύρου μας, προχωράμε απλά στην συμπλήρωση των πεδίων **Name** και **Data type**, ομοίως όπως και στις εξωτερικές μεταβλητές. Στο παράδειγμά μας, θα δημιουργήσουμε μία εσωτερική μεταβλητή ονόματος **Int_1**. Προς αποφυγή προβλημάτων συνδεσιμότητας των δύο μεταβλητών, η εσωτερική μεταβλητή θα πρέπει να έχει ίδιο μήκος με την εξωτερική μεταβλητή. Σαν αποτέλεσμα αυτού, στο πεδίο **Data type** θα επιλέξουμε το **Unsigned 32 – bit value** (*εικόνα 5.2.5.1.2.2*). Στα υπόλοιπα πεδία διατηρούμε τις τιμές που ορίστηκαν αυτόματα κατά την δημιουργία της εσωτερικής μεταβλητής μας.



Name	Data Type	Length	Format adaptation	Connection	Group	Address
1 @ConnectedRTClients	Unsigned 16-bit value	2		Internal tags		
2 @CurrentUser	Text tag 8-bit character set	255		Internal tags		
3 @CurrentUserName	Text tag 16-bit character set	255		Internal tags		
4 @DataSourceNameRT	Text tag 16-bit character set	255		Internal tags		
5 @DeltaLoaded	Unsigned 32-bit value	4		Internal tags		
6 @LocalMachineName	Text tag 8-bit character set	255		Internal tags		
7 @RedundantServerState	Unsigned 16-bit value	2		Internal tags		
8 @SCRIPT_COUNT_ACTIONS_IN_QUEUES	Unsigned 32-bit value	4		Internal tags	Script	
9 @SCRIPT_COUNT_REQUESTS_IN_QUEUES	Unsigned 32-bit value	4		Internal tags	Script	
10 @SCRIPT_COUNT_TAGS	Unsigned 32-bit value	4		Internal tags	Script	
11 @ServerName	Text tag 16-bit character set	255		Internal tags		
12 @ServerVersion	Text tag 16-bit character set	255		Internal tags		
13 @TLGRT_AVERAGE_TAGS_PER_SECOND	Floating-point number 64-bit IEEE	8		Internal tags	TagLoggingRt	
14 @TLGRT_SIZEOF_NULL_INPUT_QUEUE	Floating-point number 64-bit IEEE	8		Internal tags	TagLoggingRt	
15 @TLGRT_SIZEOF_NOTIFY_QUEUE	Floating-point number 64-bit IEEE	8		Internal tags	TagLoggingRt	
16 @TLGRT_TAGS_PER_SECOND	Floating-point number 64-bit IEEE	8		Internal tags	TagLoggingRt	
17						
18						

Εικόνα 5.2.5.1.2.1

Internal tags						
	Name	Data Type	Length	Format adaptation	Connection	Group
1	@ConnectedRTClients	Unsigned 16-bit value	2		Internal tags	
2	@CurrentUser	Text tag 8-bit character set	255		Internal tags	
3	@CurrentUserName	Text tag 16-bit character set	255		Internal tags	
4	@DatasourceNameRT	Text tag 16-bit character set	255		Internal tags	
5	@DeltaLoaded	Unsigned 32-bit value	4		Internal tags	
6	@LocalMachineName	Text tag 8-bit character set	255		Internal tags	
7	@RedundantServerState	Unsigned 16-bit value	2		Internal tags	
8	@SCRIPT_COUNT_ACTIONS_IN_QUEUES	Unsigned 32-bit value	4		Internal tags	Script
9	@SCRIPT_COUNT_REQUESTS_IN_QUEUES	Unsigned 32-bit value	4		Internal tags	Script
10	@SCRIPT_COUNT_TAGS	Unsigned 32-bit value	4		Internal tags	Script
11	@ServerName	Text tag 16-bit character set	255		Internal tags	
12	@ServerVersion	Text tag 16-bit character set	255		Internal tags	
13	@TLGRT_AVERAGE_TAGS_PER_SECOND	Floating-point number 64-bit IEEE	8		Internal tags	TagLoggingRt
14	@TLGRT_SIZEOF_NULL_INPUT_QUEUE	Floating-point number 64-bit IEEE	8		Internal tags	TagLoggingRt
15	@TLGRT_SIZEOF_NOTIFY_QUEUE	Floating-point number 64-bit IEEE	8		Internal tags	TagLoggingRt
16	@TLGRT_TAGS_PER_SECOND	Floating-point number 64-bit IEEE	8		Internal tags	TagLoggingRt
17	Int_1	Unsigned 32-bit value	4		Internal tags	
18						

Εικόνα 5.2.5.1.2.2

5.2.6 Δημιουργία και εμφάνιση των εικόνων της εφαρμογής

Στο βήμα αυτό θα ασχοληθούμε με το γραφικό κομμάτι της εφαρμογής και συγκεκριμένα με την δημιουργία και τον σχεδιασμό του περιβάλλοντος εργασίας της εφαρμογής όπου θα εποπτεύει ο χρήστης. Για τον σκοπό αυτό, θα χρησιμοποιηθεί ένα βοηθητικό πρόγραμμα σχεδίασης γραφικών του λογισμικού WinCC, το **Graphics Designer**.




5.2.6.1 Το πρόγραμμα Graphics Designer

Το πρόγραμμα σχεδίασης Graphics Designer αποτελεί ένα από τα βασικά δομικά κομμάτια του λογισμικού μας. Είναι αρμόδιο για την δημιουργία γραφικών και εικόνων της εφαρμογής μας όπως:

- ❖ Την δημιουργία και την εμφάνιση στατικών και δυναμικών αντικειμένων όπως κείμενα, γραφικά ή κουμπιά χειρισμού.
- ❖ Την δυναμική ενημέρωση, σε πραγματικό χρόνο, διαφόρων αντικειμένων που απαρτίζουν την εφαρμογή όπως την μεταβολή του μήκους ενός ιστογράμματος σε σχέση με τα δεδομένα που αντλεί από μία διαδικασία.

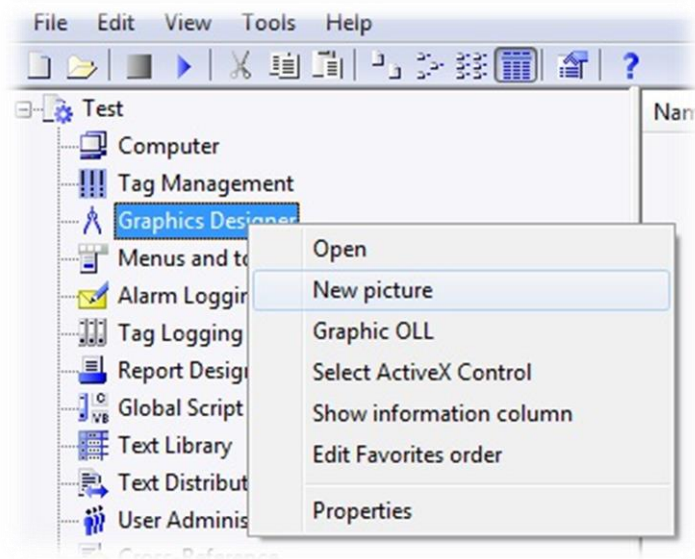
- ❖ Την αντίδραση της εφαρμογής αναλόγως με τον χειρισμό των αντικειμένων από τον εκάστοτε χρήστη. Το κλικ ενός κουμπιού ή η εισαγωγή ενός μηνύματος σε ένα πεδίο εισαγωγής μπορεί να επιφέρει διαφορετική αντίδρασή της εφαρμογής.

Μπορεί να υποστηρίξει εικόνες και γραφικά διαφόρων τύπων, όπως παρουσιάζεται αναλυτικότερα στον πίνακα παρακάτω. Μπορούμε εύκολα να ξεχωρίσουμε τις εικόνες αυτές με την βοήθεια του εικονιδίου που εμφανίζεται.

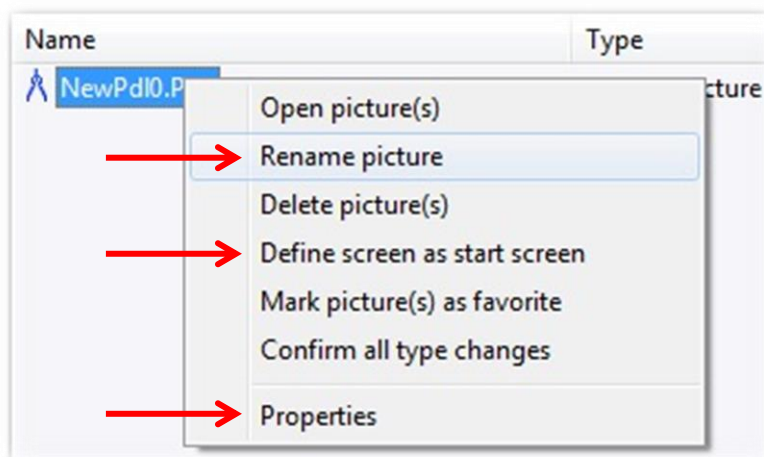
Εικονίδιο	Επεξήγηση
	Απλή εικόνα εφαρμογής
	Εικόνα πρόσοψης όπου μπορεί να χρησιμοποιηθεί σαν εικόνα εφαρμογής
	Εικόνα εφαρμογής όπου μπορεί να επηρεαστεί από μεταβολές

5.2.6.2 Δημιουργία εικόνων εφαρμογής

Για να ξεκινήσουμε να δημιουργούμε το περιβάλλον εργασίας της εφαρμογής μας, εκκινήσουμε το πρόγραμμα **Graphics Designer**. Όπως φαίνεται και στην *εικόνα 5.2.6.2.1*, κάνουμε δεξί κλικ πάνω στον σχεδιαστή, στο τμήμα πλοήγησης του λογισμικού μας, και επιλέγουμε την **δημιουργία νέας εικόνας (New Picture)**, μορφής αρχείου **.Pdl**. Για την καλύτερη οργάνωση της εφαρμογής μας, είναι χρήσιμο να μετονομάσουμε την εικόνα μας, ανάλογα με την χρήση ή την θέση που θα έχει στην εφαρμογή μας. Για να υλοποιηθεί αυτό, κάνουμε **δεξί κλικ** πάνω της και επιλέγουμε την **Μετονομασία (Rename Picture)**, όπως παρουσιάζεται στην *εικόνα 5.2.6.2.2*. Παράλληλα, μπορούμε να δούμε τις ιδιότητες και τα χαρακτηριστικά της δημιουργηθείσας εικόνας, κάνοντας κλικ στην επιλογή **Ιδιότητες (Properties)** ή επίσης να ορίσουμε την επιλεγμένη εικόνα ως **εικόνα εκκίνησης** της εφαρμογής, κάνοντας κλικ στην επιλογή **Define screen as start screen**.

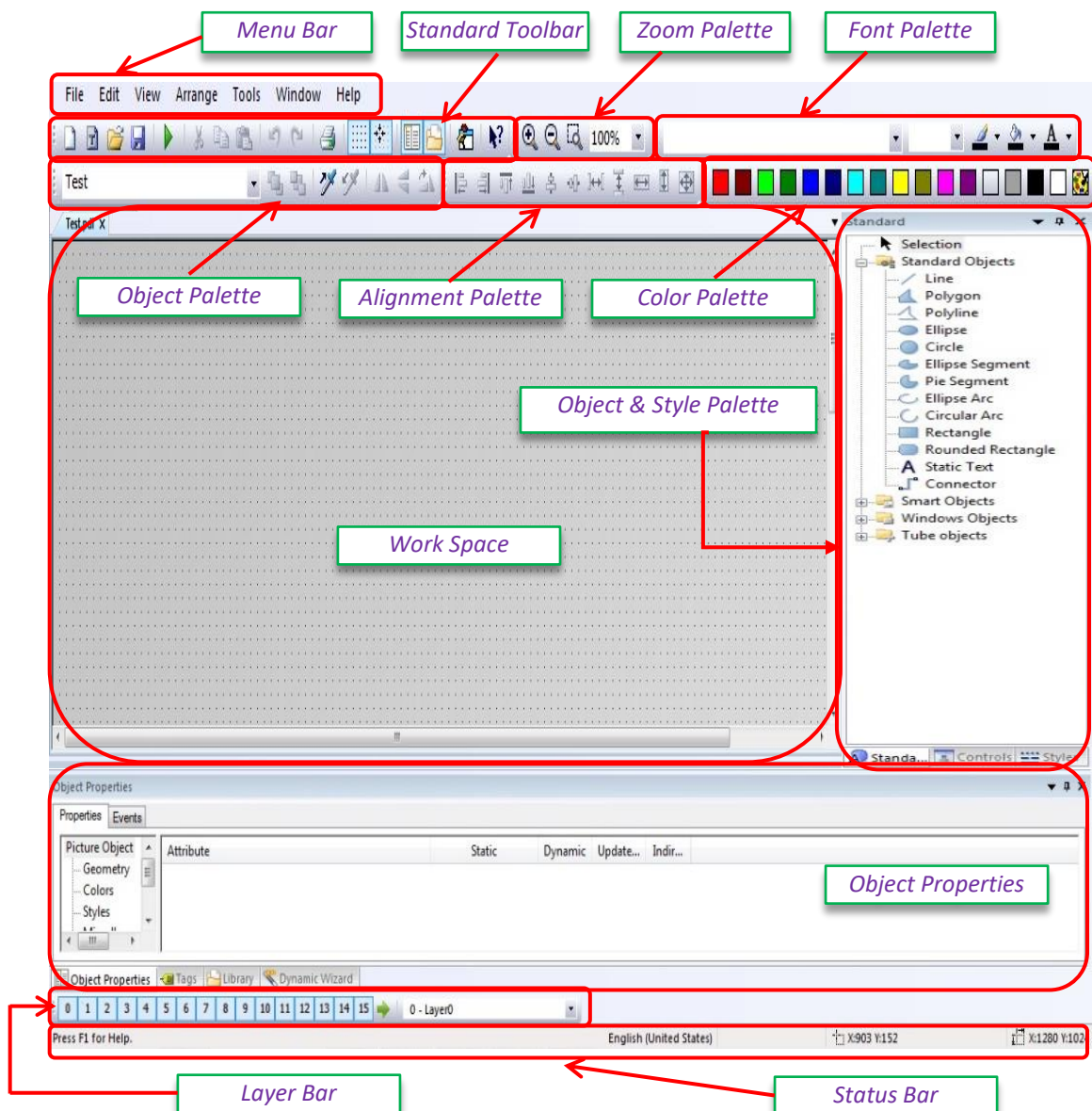


Εικόνα 5.2.6.2.1



Εικόνα 5.2.6.2.2

Έχοντας δημιουργήσει πλέον όλες τις εικόνες που χρειάζονται για την εφαρμογή μας, για να προχωρήσουμε στον σχεδιασμό αλλά και την παραμετροποίησή τους, κάνουμε **διπλό κλικ** πάνω στην εικόνα που επιθυμούμε. Αυτομάτως, μας ανοίγει ένα νέο παράθυρο όπου αποτελεί το βασικό περιβάλλον εργασίας του σχεδιαστή μας.



Εικόνα 5.2.6.2.3

Με μία πρώτη ματιά, όπως αυτό εμφανίζεται στην εικόνα 5.2.6.2.3, το πρόγραμμα **Graphics Designer**, για να διασφαλιστεί η ευκολία και η άνεση στην χρήση του, είναι δομημένο σύμφωνα με τις αρχές λειτουργίας του λογισμικού των Windows. Εμβαθύνοντας σε αυτό, βλέπουμε ότι το πρόγραμμα απαρτίζεται από μία βασική επιφάνεια εργασίας και διάφορες μπάρες μενού και εργαλείων, πλήρως παραμετροποιήσιμες. Παρακάτω πραγματοποιείται μία εκτενής ανάλυση σε καθεμία από τις περιοχές εργασίας του προγράμματος.

5.2.6.2.1 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.

5.2.6.2.2 Standard Toolbar

Περιέχει κουμπιά για τη γρήγορη και εύκολη χρήση τόσο για τις απλές εντολές διαχείρισης των αρχείων εικόνας (Open, Save, Copy) όσο και για τις ειδικές εντολές του σχεδιαστή όπως την εντολή προσομοίωσης (Runtime).

5.2.6.2.3 Zoom Palette

Εκτός των κουμπιών εστίασης όπου υπάρχουν στην Βασική Μπάρα (Standard Toolbar), το πρόγραμμα Graphics Designer μας παρέχει μία επιπλέον μπάρα για τον εύκολο καθορισμό του συντελεστή εστίασης (zoom) για το ενεργοποιημένο παράθυρο.

5.2.6.2.4 Font Palette

Όπως και σε έναν απλό κειμενογράφο, η παλέτα επεξεργασίας γραμματοσειράς επιτρέπει την αλλαγή του τύπου, του μεγέθους και του χρώματος των γραμμάτων σε αντικείμενα κειμένου (text objects), όπως επίσης και του χρώματος της γραμμής των κύριων αντικειμένων.

5.2.6.2.5 Object Palette

Η παλέτα αντικειμένων (Object palette) περιέχει όλα τα κύρια σχήματα όπου μπορούν να χρησιμοποιηθούν για την κατασκευή μίας εφαρμογής. Παράλληλα, παρέχει μία βιβλιοθήκη αποθήκευσης αντικειμένων, όπου ο χρήστης μπορεί να αποθηκεύει ακόμα και δικά του σχέδια.

5.2.6.2.6 Alignment Palette

Επιτρέπει την αλλαγή της απόλυτης θέσης των αντικειμένων που απαρτίζουν την εφαρμογή. Επίσης παρέχει την δυνατότητα ευθυγράμμισης και στοίχισης ενός ή περισσότερων αντικειμένων, χρησιμοποιώντας ως βάση είτε ένα αντικείμενο είτε ακόμα και όλο το εύρος της επιφάνειας εργασίας.

5.2.6.2.7 Color Palette

Η παλέτα διαχείρισης χρωμάτων χρησιμοποιείται για να αλλαγή του χρώματος και της εμφάνισης των επιλεγόμενων αντικειμένων. Καθ' ομοίωση μιας απλής παλέτας χρωμάτων σε ένα σύστημα windows, εκτός από τα 16 κύρια χρώματα όπου υπάρχουν, ο χρήστης έχει την δυνατότητα δημιουργίας ποικίλλων συνδυασμών και αποθήκευσής τους.

5.2.6.2.8 Workspace

Αποτελεί την κύρια περιοχή εργασίας του χρήστη όπου τοποθετούνται όλα τα αντικείμενα που ο χρήστης επιθυμεί να εμφανίζονται στην εφαρμογή μας.

5.2.6.2.9 Style Palette

Παρέχει την άμεση αλλαγή της εμφάνισης του αντικειμένου όπου επιλέγει ο χρήστης. 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 της γραμμής ή το γέμισμα του αντικειμένου.

5.2.6.2.10 Object Properties

Παρέχει στον χρήστη πληροφορίες σχετικά με το μέγεθος, το χρώμα και τις ιδιότητες του αντικειμένου που διαχειρίζεται ο χρήστης.

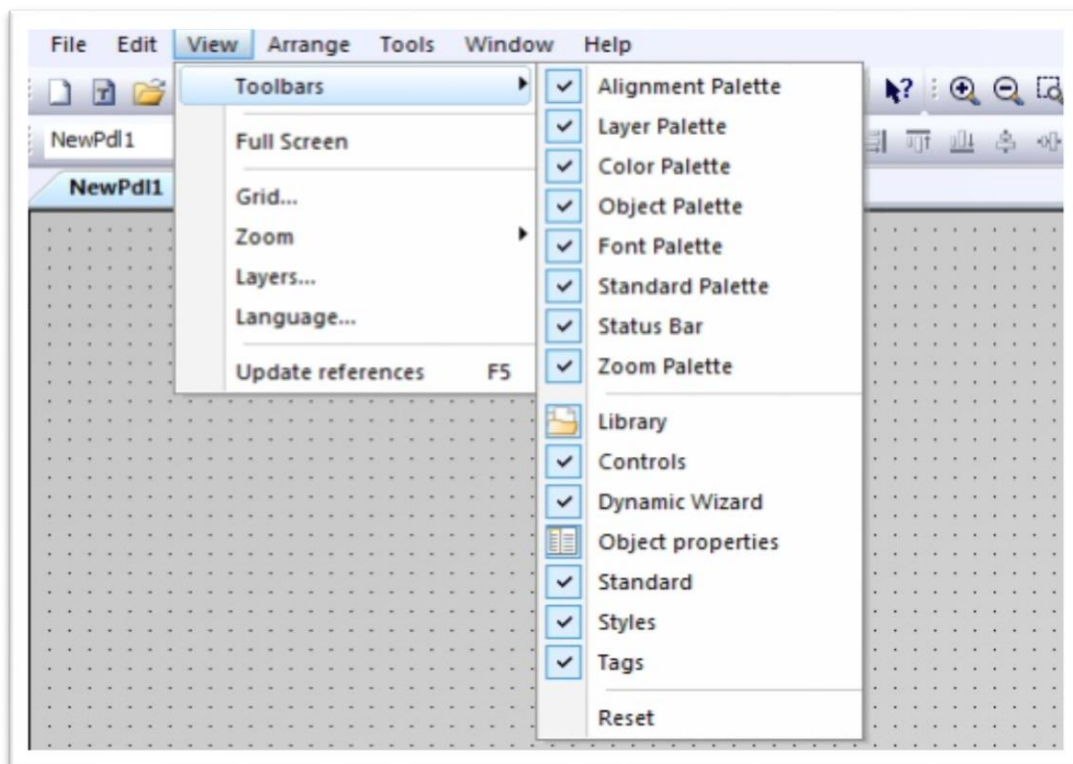
5.2.6.2.11 Layer Bar

Μας δίνει την δυνατότητα εναλλαγής μεταξύ των επιπέδων (layers) όπου υπάρχουν στο αντικείμενό μας. Σαν αρχικό επίπεδο είναι εξ' ορισμού το layer 0.

5.2.6.2.12 Status Bar

Βρίσκεται στο κάτω μέρος του παραθύρου και παρέχει χρήσιμες πληροφορίες σχετικές με την θέση και τις ακριβές συντεταγμένες όπου βρίσκονται τα διάφορα αντικείμενα, με το μέγεθος το οποίο έχουν με βάση την περιοχή εργασίας της εφαρμογής καθώς και με γενικές πληροφορίες χρήσης του προγράμματος από τον χρήστη.

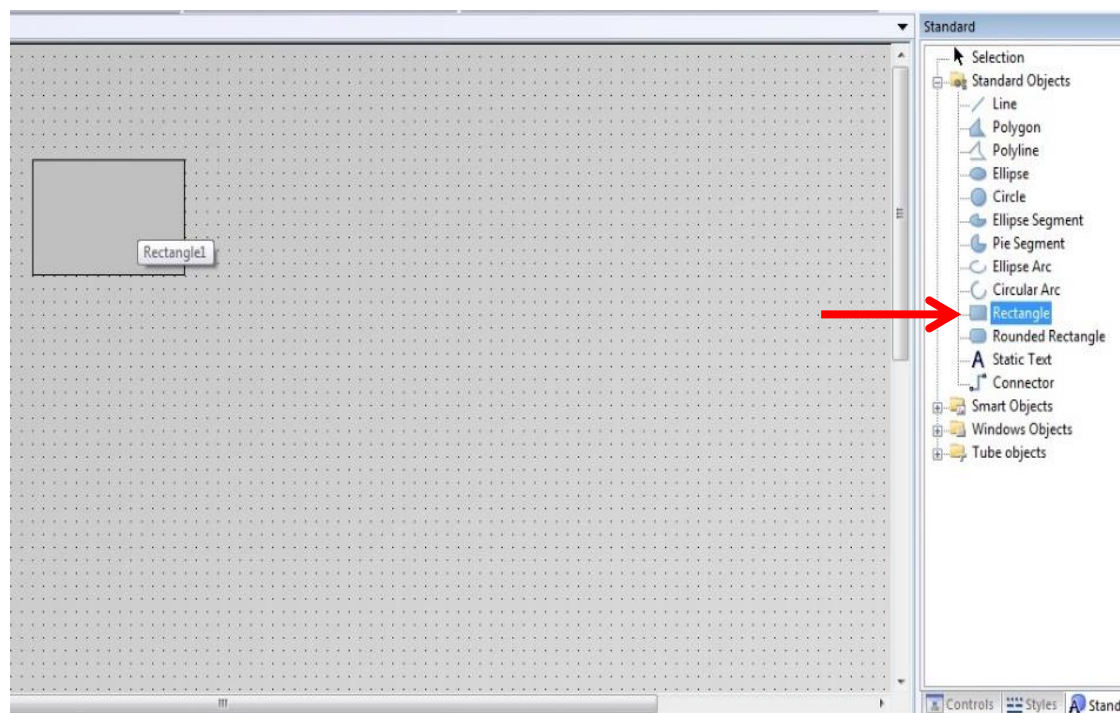
Οι μπάρες μενού και εργαλείων μπορούν να αποκρυφθούν αλλά και να μετακινηθούν σε οποιοδήποτε σημείο του παραθύρου του προγράμματος. Μπορούμε εύκολα να τις διαχειριστούμε πηγαίνοντας από το **Menu Bar** στην λειτουργία **View** και επιλέγοντας στο μενού **Toolbars** τις μπάρες και τα εργαλεία που επιθυμεί ο εκάστοτε χρήστης (εικόνα 5.2.6.2.4)



Εικόνα 5.2.6.2.4

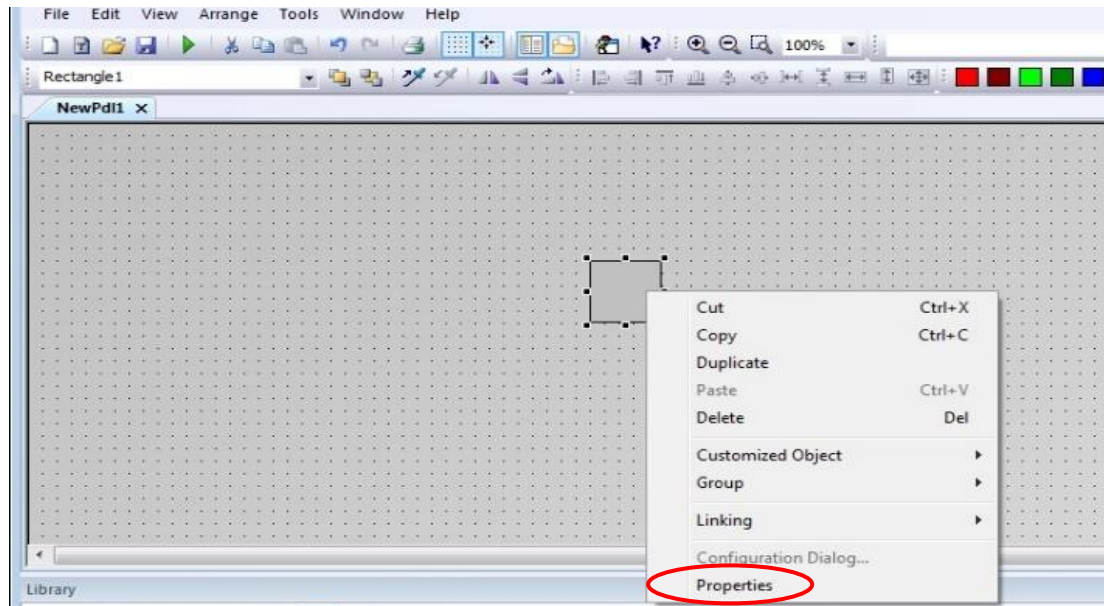
5.2.6.3 Δυναμική μεταβολή αντικειμένων

Όπως αναφέρθηκε και στην παραπάνω παράγραφο, μία από τις δυνατότητες που μας παρέχει το βοηθητικό πρόγραμμα **Graphics Designer** είναι η ικανότητά του να μετασχηματίζει τα απλά στατικά αντικείμενα σε μία εφαρμογή σε δυναμικά, μεταβάλλοντας είτε το περιεχόμενό τους είτε ακόμα και τις ιδιότητές τους. Οι μεταβολές αυτές γίνονται σε άμεση σύνδεση με τις μεταβολές που μπορεί να δεκτεί το λογισμικό απευθείας από τον χρήστη ή μέσω των εξωτερικών μεταβλητών που έχουν οριστεί στο βιομηχανικό σύστημα SCADA όπου εποπτεύουμε. Για να γίνει πιο εύκολα αντιληπτή η σημασία της δυνατότητας αυτής, θα παρουσιάσουμε στον παρακάτω παράδειγμα έναν από τους τρόπους με τον οποίο μπορούμε να ορίσουμε την δυναμική μεταβολή ενός αντικειμένου σύμφωνα με τα εξωτερικά ερεθίσματα που δέχεται από την μεταβλητή εξόδου του συστήματος SCADA. Συγκεκριμένα, θα δημιουργήσουμε μία ένδειξη στην οθόνη μας, η οποία θα εναλλάσσει το χρώμα της από κόκκινο σε πράσινο, ανάλογα με την τιμή της μεταβλητής εξόδου που το έχουμε συνδέσει. Για να το πραγματοποιήσουμε, θα χρειαστεί να εισάγουμε ένα σχήμα από την εργαλειοθήκη **Object palette**. Στο παράδειγμά μας, θα εισάγουμε ένα ορθογώνιο κάνοντας αρχικά κλικ στην επιλογή **Rectangle** της Object Palette και έπειτα κλικ στο **Workspace** για να τοποθετηθεί στην επιφάνεια εργασίας μας, όπως παρουσιάζεται στην εικόνα 5.2.6.3.1.



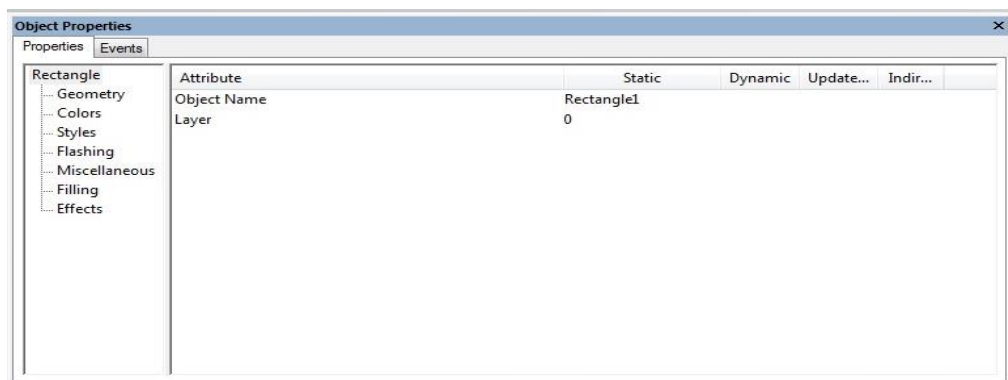
Εικόνα 5.2.6.3.1

Τοποθετώντας το στην περιοχή εργασίας μας, επόμενο μας βήμα είναι να το ρυθίσουμε έτσι ώστε να του δώσουμε την δυναμική αλλαγή που επιθυμούμε. Κάνοντας **δεξί κλικ** πάνω στο αντικείμενό μας, εμφανίζεται το παράθυρο επιλογών του (εικόνα 5.2.6.3.2) Από εκεί επιλέγουμε το μενού **Properties**, για να έχουμε πρόσβαση σε όλες τις ιδιότητές του.

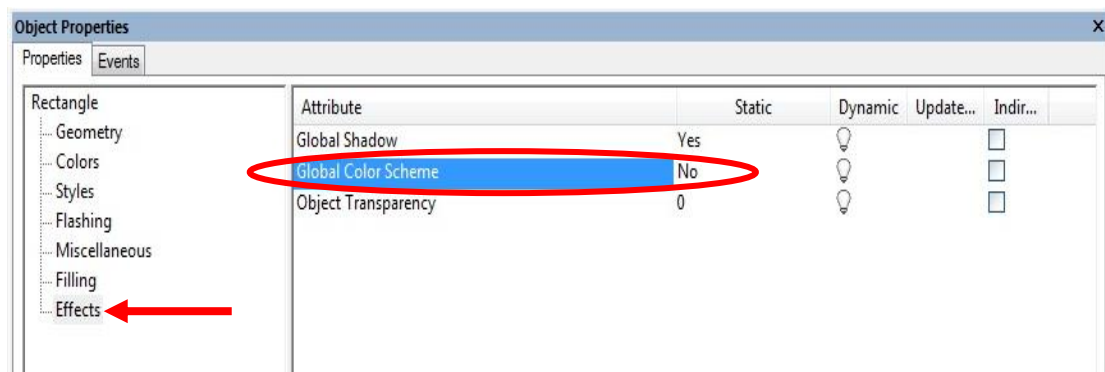


Εικόνα 5.2.6.3.2

Στο παράθυρο το οποίο εμφανίζεται (εικόνα 5.2.6.3.3), παρατηρούμε στην αριστερή στήλη όλες τις επιλογές που δίνονται για το συγκεκριμένο σχήμα, ενώ στην δεξιά στήλη τις υποεπιλογές που μας παρέχονται. Πριν προχωρήσουμε στην ρύθμιση του αντικειμένου μας, θα πρέπει πρώτα να ενεργοποιήσουμε την **χρωματική του μεταβολή**. Για τον λόγο αυτό κάνουμε κλικ στην κατηγορία **Effects** και απενεργοποιούμε την επιλογή **Global Color Scheme**, όπως διακρίνεται στην *εικόνα 5.2.6.3.4*.

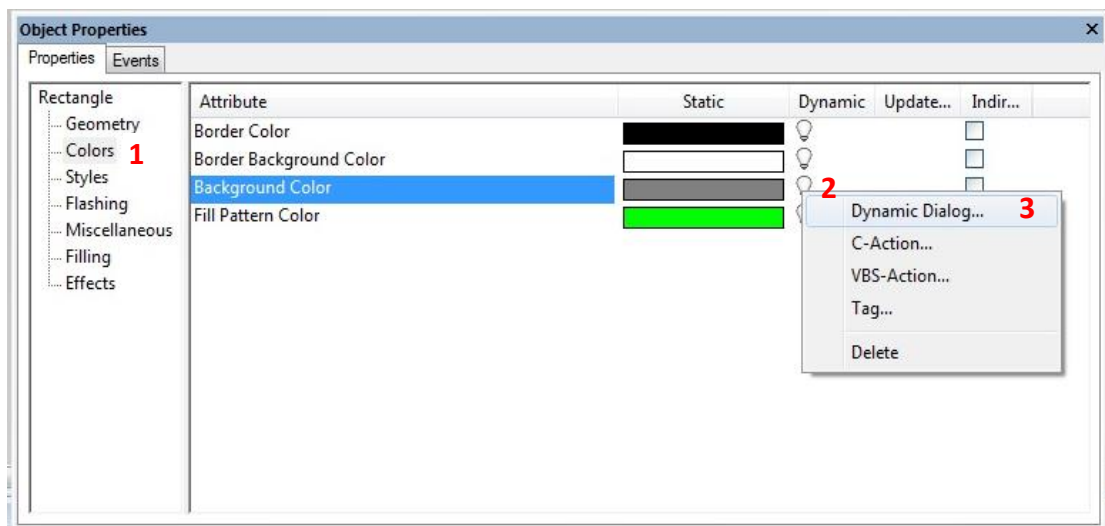


Εικόνα 5.2.6.3.3



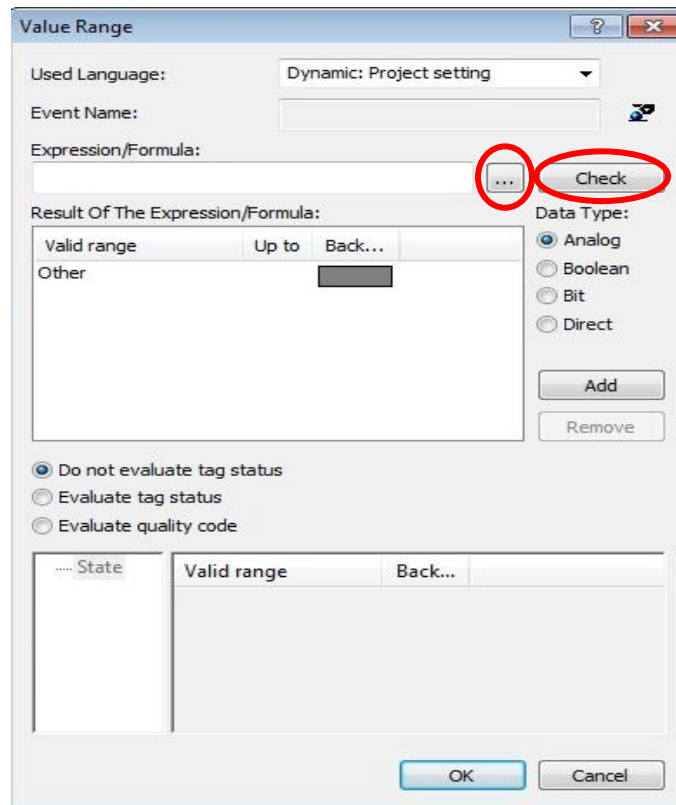
Εικόνα 5.2.6.3.4

Έπειτα, προχωρώντας στην κατηγορία **Colors**, κάνουμε **δεξί κλικ στο εικονίδιο λάμπας** όπου υπάρχει στην επιλογή **Background Color**. Στο παράθυρο που αναδύεται, κάνουμε αριστερό κλικ στην επιλογή **Dynamic Dialog** (εικόνα 5.2.6.3.5), για να συνδέσουμε το αντικείμενό μας με την εξωτερική μεταβλητή έτσι ώστε να μεταβάλλεται δυναμικά.

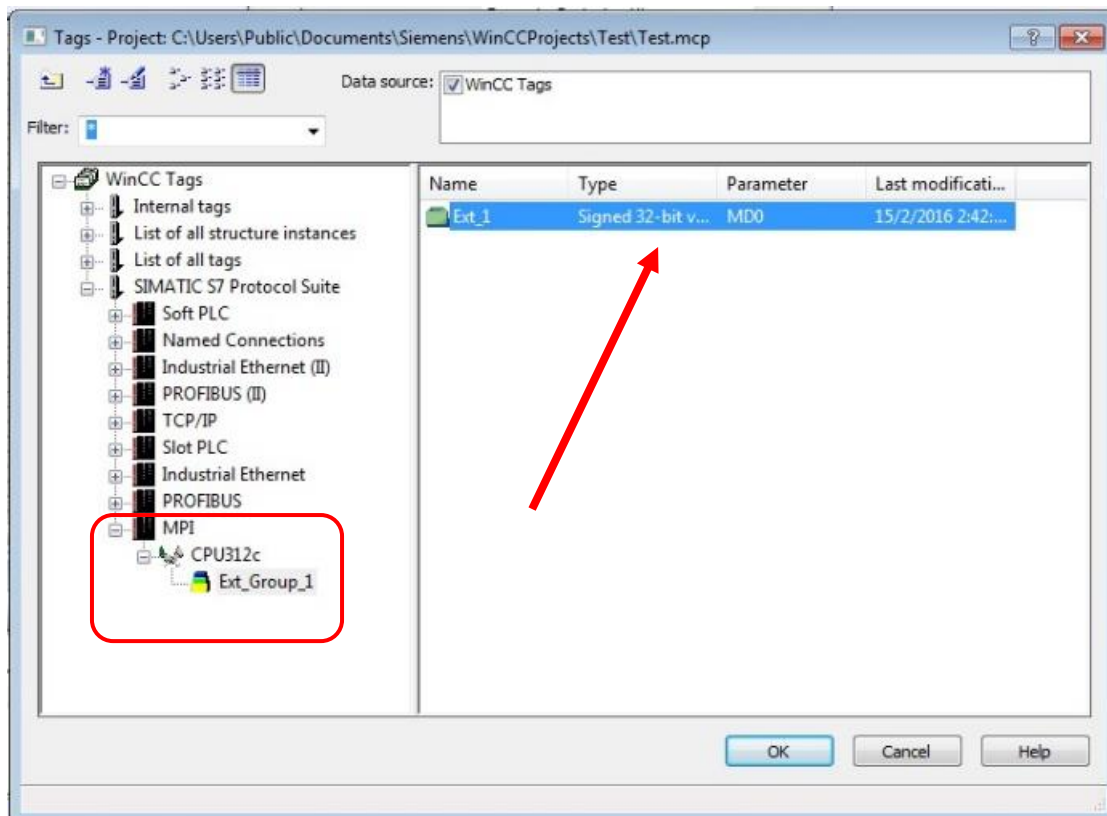


Εικόνα 5.2.6.3.5

Στο νέο πλέον παράθυρο που μας εμφανίζεται, όπως μας παρουσιάζεται στην εικόνα 5.2.6.3.6, πηγαίνουμε στο πεδίο **Expression/Formula** και κάνουμε αριστερό κλικ στο κουμπί όπου βρίσκεται εκεί. Αμέσως μετά επιλέγουμε το μενού **Tag**, για να εισάγουμε την εξωτερική μας μεταβλητή. Αυτόματα μας ανοίγει ένα παράθυρο πλοήγησης στο οποίο θα πρέπει να ανατρέξουμε στην επικοινωνία που είχαμε δημιουργήσει σε προηγούμενο βήμα (εικόνα 5.2.6.3.7). Προς υπενθύμισή μας, η μεταβλητή μας είναι η **Ext_1**. Για να την προσθέσουμε, την επιλέγουμε και κάνουμε κλικ στο κουμπί **OK**. Για να ελέγξουμε εάν έχουμε εισάγει σωστά την εξωτερική μας μεταβλητή, κάνοντας απλά κλικ στο κουμπί **Check**.

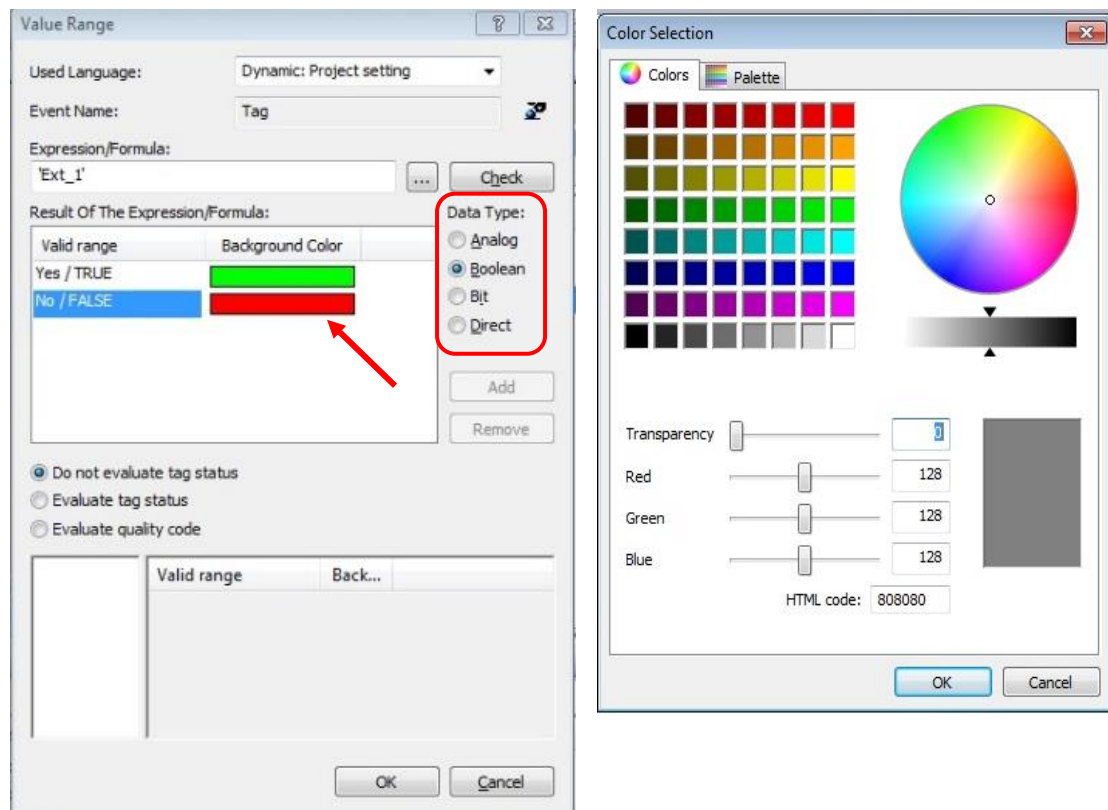


Εικόνα 5.2.6.3.6



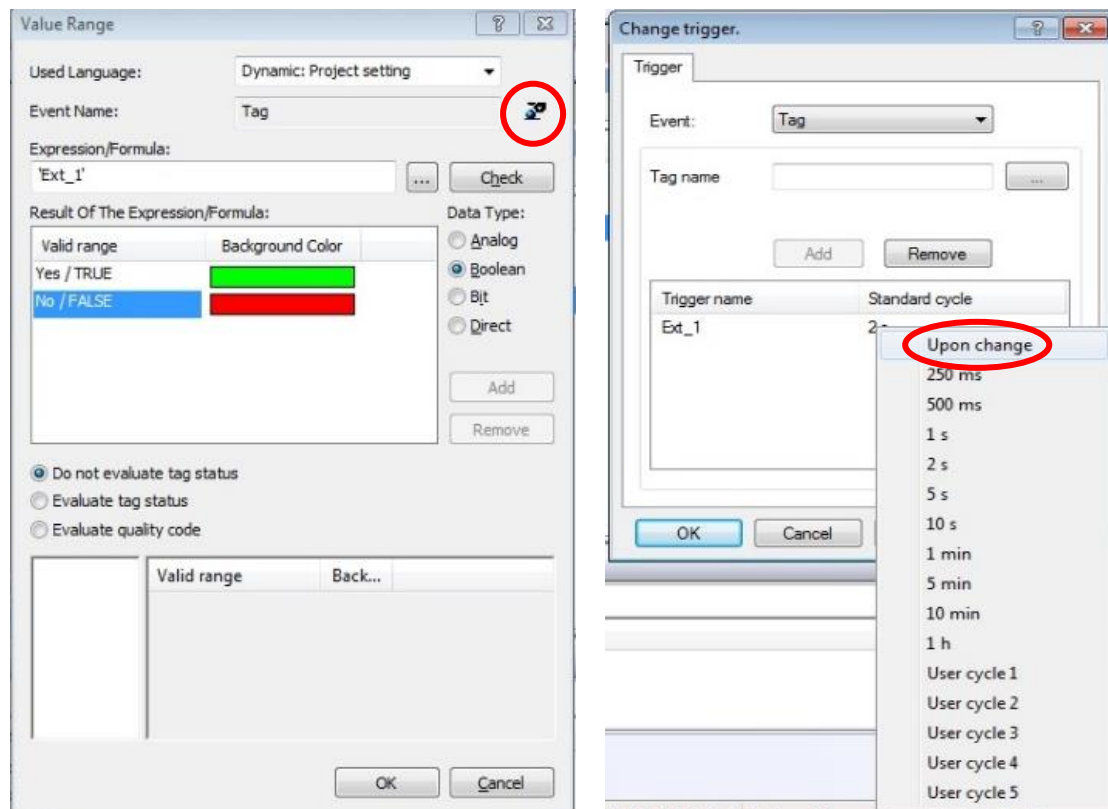
Εικόνα 5.2.6.3.7

Προχωρώντας παρακάτω, στο πεδίο **Results of the Expression/Formula**, επιλέγουμε τα χρώματα και τις συνθήκες όπου επιθυμούμε να πραγματοποιούνται οι μεταβολές στο σχήμα. Για να πραγματοποιηθεί αυτό, επιλέγουμε κατ' αρχάς στο πεδίο **Data Type** την έξοδο της μεταβλητής μας ως **Boolean**, για να διασφαλίσουμε δύο καταστάσεις εξόδου, **TRUE** και **FALSE**. Εν συνεχεία, κάνοντας διπλό κλικ στην στήλη **Background Color**, διαλέγουμε το χρώμα της επιλογής μας για κάθε μία από τις καταστάσεις αυτές. Στο παράδειγμά μας, κατά την κατάσταση TRUE, το αντικείμενό μας θα παίρνει πράσινο χρώμα ενώ κατά την επιλογή FALSE κόκκινο χρώμα (εικόνα 5.2.6.3.8).



Εικόνα 5.2.6.3.8

Στο σημείο αυτό αξίζει να αναφέρουμε ότι μπορούμε επίσης να ορίσουμε και την συχνότητα ενημέρωσης του λογισμικού WinCC για τις μεταβολές στην τιμή της εξωτερικής μεταβλητής μας. Για να υλοποιηθεί αυτό, κάνουμε **κλικ στο κουμπί** όπου βρίσκεται στο πεδίο **Event Name** (εικόνα 5.2.6.3.9). Στο νέο παράθυρο όπου μας εμφανίζεται μπορούμε να προσθέσουμε εξίσου τις μεταβλητές όπου επιθυμούμε να επηρεάζουν το αντικείμενό μας, από το πεδίο **Tag Name**. Όσον αφορά το θέμα της συχνότητας ενημέρωσης, κάνουμε απλά **δεξί κλικ** στην στήλη **Standard cycle**, στην λίστα μεταβλητών όπου βρίσκεται παρακάτω. Με τον τρόπο αυτό μας εμφανίζεται μία λίστα με όλες τις δυνατές επιλογές συχνότητας όπου μας δίνει το λογισμικό. Στο παράδειγμά μας, θα ρυθμίσουμε το αντικείμενο ώστε να έχουμε ενημέρωση των αλλαγών των τιμών σε πραγματικό χρόνο, επιλέγοντας την κατηγορία **Upon Change**. Ολοκληρώνοντας τις απαιτούμενες ρυθμίσεις, κάνουμε κλικ στο κουμπί **OK**.

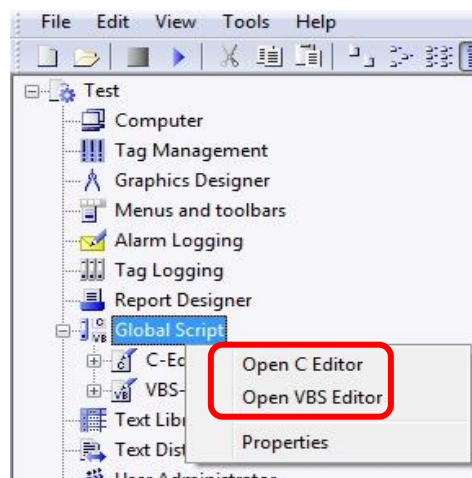
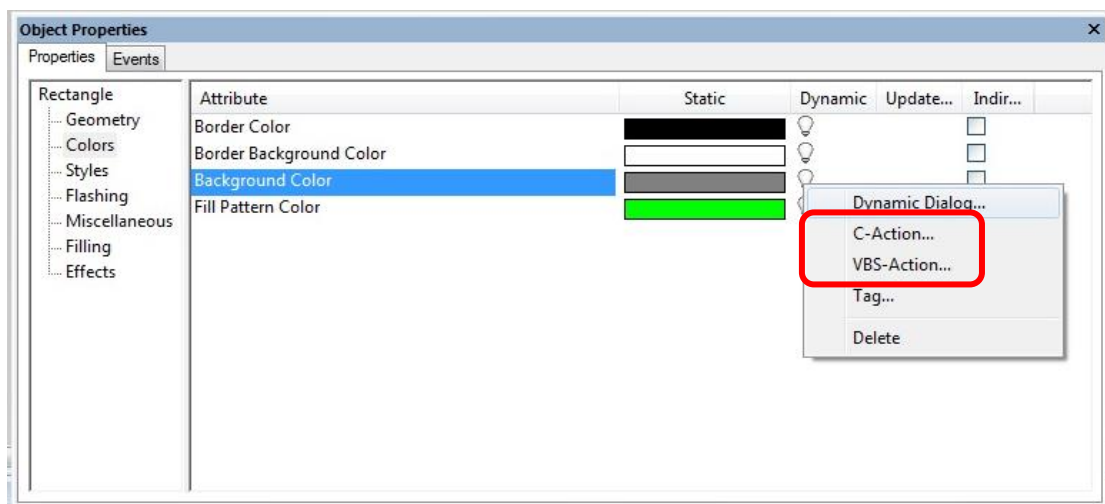


Εικόνα 5.2.6.3.9

Εν κατακλείδι, με τον τρόπο αυτό το αντικείμενο το οποίο έχουμε εισάγει θα αλλάζει το χρώμα του από κόκκινο, όταν η μεταβλητή μας είναι σε λογικό 0, σε πράσινο, όταν η μεταβλητή μας παίρνει την τιμή του λογικού 1.

Το πρόγραμμα Graphics Designer μας δίνει επίσης την δυνατότητα να μετατρέψουμε σε δυναμική την εμφάνιση ενός αντικειμένου και με την βοήθεια μικρών προγραμμάτων (scripts) όπου ο κάθε χρήστης μπορεί να δημιουργήσει. Για τον σκοπό αυτό θα χρησιμοποιήσουμε **Global Script**. Πρόκειται για έναν βασικό editor του λογισμικού στον οποίο ο χρήστης μπορεί να γράψει κώδικα προγραμμάτων σε γλώσσα **Visual Basic** ή **Ansi – C**.

Για να προγραμματίσουμε λοιπόν το αντικείμενό μας ώστε να υλοποιεί μόνο του μία ενέργεια, αρκεί να κάνουμε δεξί κλικ πάνω στο αντικείμενο και να μπούμε το μενού **Ιδιότητες (Properties)**. Από εκεί, όπως αναφέραμε και στην σύνδεση με την εξωτερική μεταβλητή, κάνουμε **δεξί κλικ πάνω στο εικονίδιο λάμπας** και επιλέξουμε το μενού **VBS – Action** ή το μενού **C – Action**. Μπορούμε επίσης να ανοίξουμε τον Editor και από το **τμήμα προορισμού (navigation window)** του λογισμικού WinCC, επιλέγοντας την κατηγορία **Global Script** και κάνοντας πάνω της **δεξί κλικ (εικόνα 5.2.6.3.10)**.



Εικόνα 5.2.6.3.10

5.2.7 Καθορισμός των ιδιοτήτων του WinCC Runtime

Το τελευταίο βήμα αφορά στη ρύθμιση των ιδιοτήτων που λαμβάνει χώρα στην κατηγορία **Computer**, τις οποίες αναφέραμε αρχικά. Σε γενικές γραμμές, το βήμα αυτό μπορεί να γίνει είτε στην αρχή είτε στο τέλος της διαδικασίας δημιουργίας μίας εφαρμογής, χωρίς να υπάρχουν περιορισμοί.

Έχοντας ήδη δημιουργήσει και αποθηκεύσει την εφαρμογή που επιθυμούμε, σύμφωνα με τα βήματα που αναλύθηκαν στις παραπάνω ενότητες, απομένει μόνο να προχωρήσουμε στην ενεργοποίησή της. Για να υλοποιηθεί αυτό, απλά κάνουμε κλικ το **κουμπί ►** όπου βρίσκεται στην **Standard Toolbar** του λογισμικού μας (εικόνα 5.2.7.1). Αυτόματα μας εμφανίζεται η οθόνη εποπτείας της εφαρμογής που έχουμε δημιουργήσει.



Εικόνα 5.2.7.1

Η παραπάνω διαδικασία ακολουθήθηκε κατά γράμμα προκειμένου να υλοποιηθούν οι εφαρμογές της πτυχιακής μας εργασίας που θα αναλυθούν στο επόμενο κεφάλαιο.

Κεφάλαιο 6

Προσομοίωση Εφαρμογών στο
SIMATIC WinCC 7.2

Όλα τα παραδείγματα που χρησιμοποιήθηκαν σε προηγούμενα κεφάλαια, αποτελούν τα βασικά δομικά κομμάτια για την δημιουργία μίας εφαρμογής εποπτικού ελέγχου. Στο κεφάλαιο όμως αυτό, θα προσπαθήσουμε να πραγματοποιήσουμε μία σύνδεση των κομματιών αυτών έτσι ώστε να παρουσιάσουμε τον τρόπο δημιουργίας ενός ολοκληρωμένου πλέον περιβάλλοντος εργασίας με το οποίο θα μπορούμε να υλοποιούμε τον εποπτικό μας έλεγχο σε ένα βιομηχανικό σύστημα SCADA. Παρακάτω θα ακολουθήσει μία διεξοδική παρουσίαση του τρόπου δημιουργίας αλλά και του ζητούμενου αποτελέσματος δύο περιβάλλοντων εργασίας:

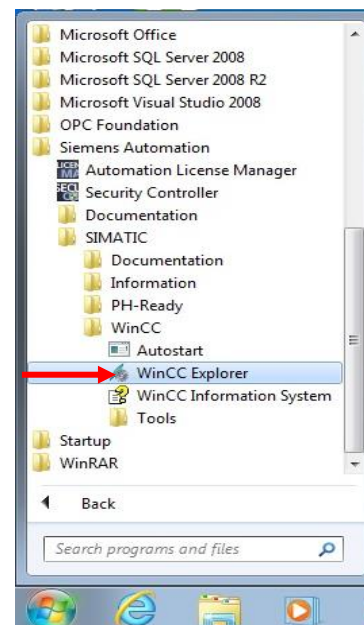
1. Χειρισμός και λειτουργία φωτεινών σηματοδοτών.
2. Εποπτεία και έλεγχος στάθμης δεξαμενής σε μία βιομηχανική μονάδα.

6.1 Χειρισμός και λειτουργία φωτεινών σηματοδοτών

Στην ενότητα αυτή θα δημιουργήσουμε ένα εικονικό σύστημα ελέγχου φωτεινών σηματοδοτών. Θα υλοποιήσουμε μία βασική οθόνη εργασίας και ο χρήστης, με το πάτημα του ανάλογου κουμπιού, θα πραγματοποιεί αλλαγή στο χρώμα και στην ένδειξη του φωτεινού σηματοδότη.

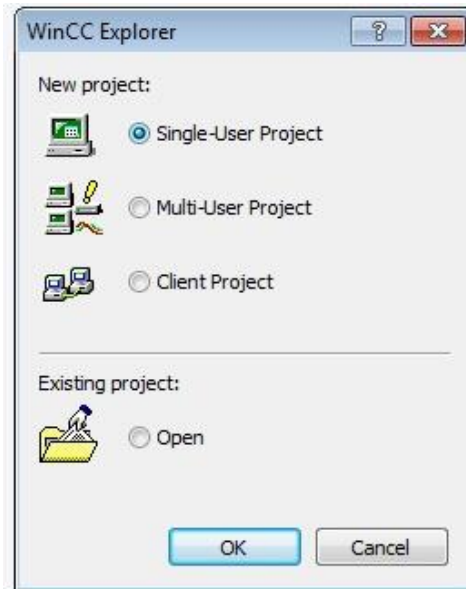
Όπως είδαμε και σε προηγούμενα κεφάλαια, πρώτη μας κίνηση είναι να ανοίξουμε το λογισμικό και να δημιουργήσουμε ένα νέο project στο οποίο θα εργαστούμε. Εκκινώντας το **WinCC** είτε από το μενού **Έναρξη** είτε από την **συντόμευση στην επιφάνεια εργασίας (εικόνα 6.1.1)**, μας εμφανίζεται το αναδυόμενο παράθυρο δημιουργίας νέου project, όπως αυτό παρουσιάζεται στην **εικόνα 6.1.2**.

Στις επιλογές όπου μας εμφανίζονται, κάνουμε κλικ στην δημιουργία ενός **single – user project** και πατάμε το κουμπί **OK** για να προχωρήσουμε παρακάτω. Εν συνεχεία, στο νέο παράθυρο όπου μας εμφανίζεται, στο πεδίο **Project name**, συμπληρώνουμε το όνομα του project που επιθυμούμε.

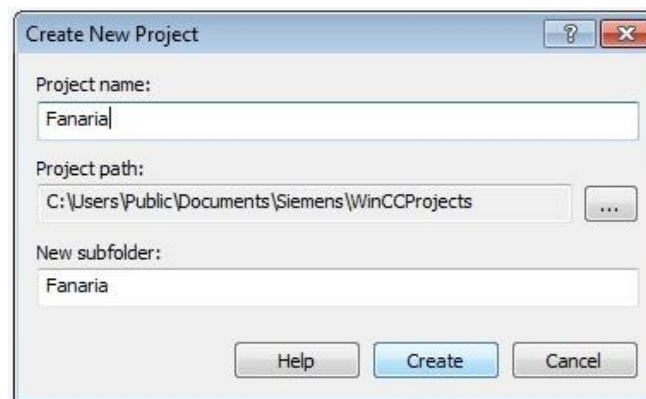


Εικόνα 6.1.1

Στο συγκεκριμένο παράδειγμα, θα δημιουργήσουμε ένα περιβάλλον εργασίας με όνομα **Fanaria** (εικόνα 6.1.3). Αφήνοντας ως έχει ρυθμιστεί το μονοπάτι αποθήκευσης του project, κάνουμε κλικ στο κουμπί **Create** για να το υλοποιήσουμε.



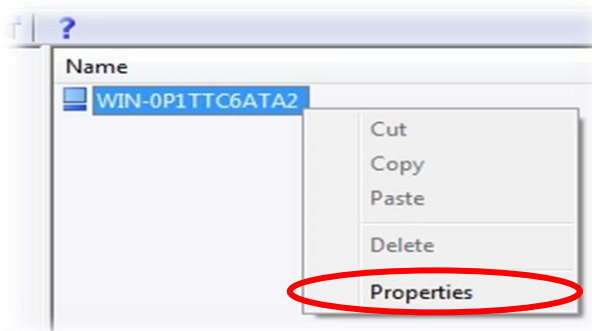
Εικόνα 6.1.2



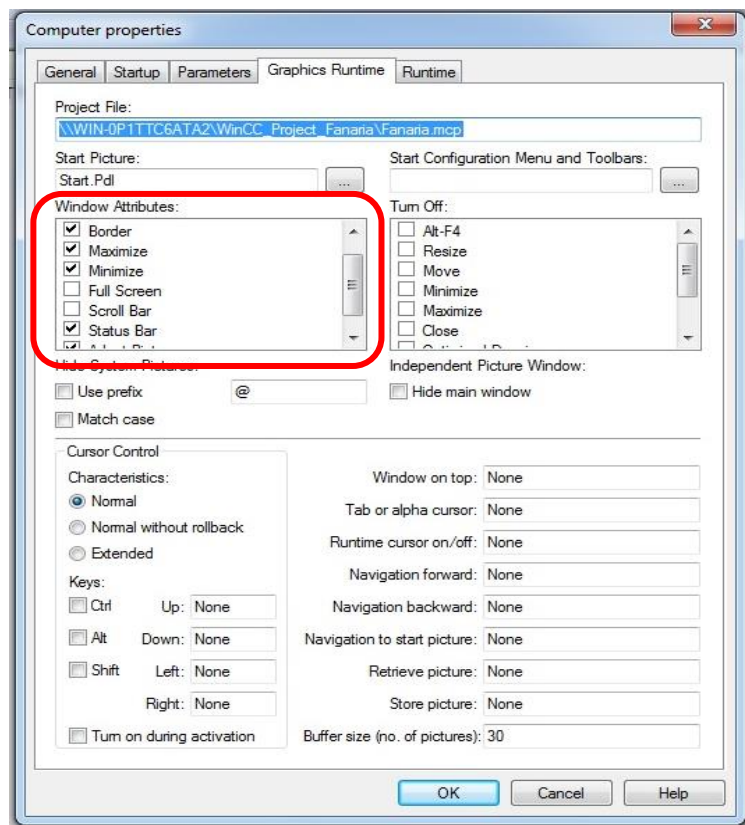
Εικόνα 6.1.3

Ακολουθώντας την μεθοδολογία που αναλύθηκε σε παραπάνω κεφάλαιο, μετά το πέρας της δημιουργίας του project, καθώς έχουν εμφανιστεί όλες οι λειτουργίες στο τμήμα πλοήγησης του λογισμικού, κάνουμε **αριστερό κλικ** στο μενού **Computer** και έπειτα, στον διαθέσιμο υπολογιστή όπου μας εμφανίζεται στο τμήμα δεδομένων του λογισμικού, κάνουμε **δεξί κλικ** και επιλέγουμε το μενού **Properties** (εικόνα 6.1.4).

Στο παράθυρο των ιδιοτήτων που μας ανοίγει, επιβεβαιώνουμε σαν πρώτο βήμα ότι το όνομα του υπολογιστή στο πεδίο **Computer Name** συμπίπτει με το όνομα του συστήματος όπου εργαζόμαστε. Εν συνεχεία, επιβεβαιώνουμε ότι οι ρυθμίσεις στις καρτέλες **Startup** και **Parameters**, σχετικά με τα υποπρογράμματα όπου εκκινούν καθώς και με τις ζώνες ώρας του λογισμικού αντίστοιχα, είναι οι επιθυμητές. Επιπλέον, στην καρτέλα **Graphics Runtime**, ενεργοποιούμε όλες τις επιλογές **εκτός της FullScreen και της Scroll Bar**, όπου αφορούν στην παρουσίαση και την εμφάνιση του περιβάλλοντος εργασίας μας (εικόνα 6.1.5). Η ρύθμιση αυτή γίνεται διότι επιθυμούμε να τρέχει το project μας σε παραθυρικό περιβάλλον. Ολοκληρώνοντας τα ρυθμίσεις αυτές, κάνουμε κλικ στο κουμπί **OK** για να προχωρήσουμε παρακάτω.



Εικόνα 6.1.4

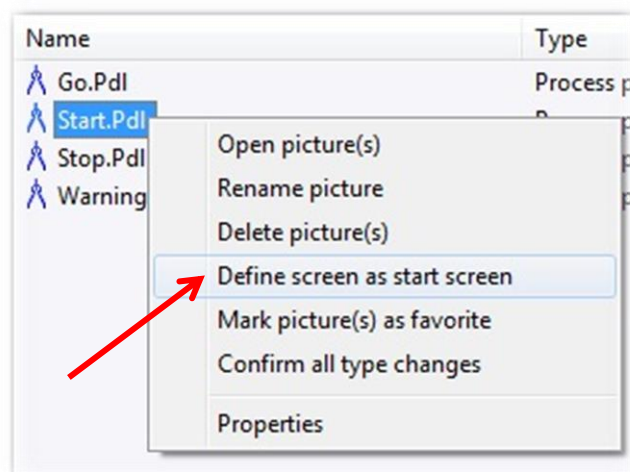


Εικόνα 6.1.5

Επόμενο βήμα μας είναι η δημιουργία των εικόνων και των αντικειμένων που θα απαρτίζουν το περιβάλλον εργασίας μας. Για να πραγματοποιηθεί αυτό, κάνουμε **δεξί κλικ** στο υποπρόγραμμα **Graphic Designer** και επιλέγουμε το μενού **New Picture**. Στην συγκεκριμένη εφαρμογή θα χρειαστούμε **τέσσερις εικόνες**, ανάλογα ονοματισμένες, όπως εμφανίζεται στην *εικόνα 6.1.6*. Στο σημείο αυτό μπορούμε να ορίσουμε και την αρχική εικόνα του περιβάλλοντος εργασίας μας, κάνοντας πάνω της δεξί κλικ και επιλέγοντας την λειτουργία **Define as Start Screen** (*εικόνα 6.1.7*).

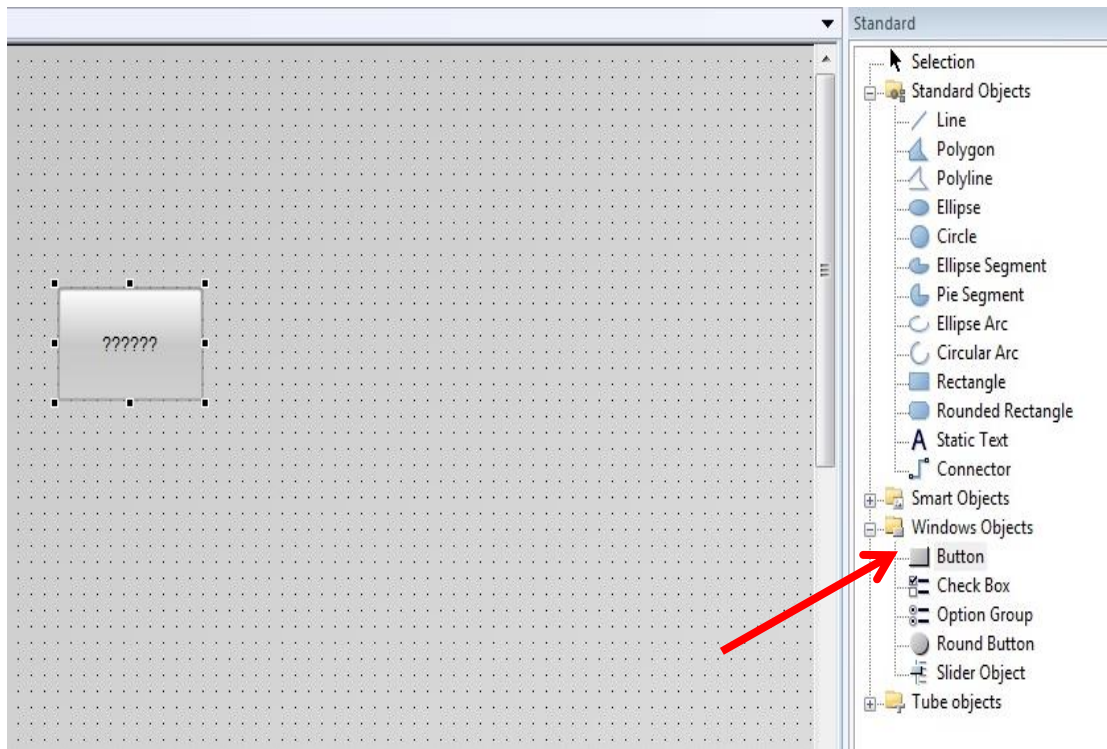
Name	Type	Last Change
Go.Pdl	Process picture	2/23/2016 7:25:44 PM
Start.Pdl	Process picture	2/23/2016 7:22:51 PM
Stop.Pdl	Process picture	2/23/2016 7:14:54 PM
Warning.Pdl	Process picture	2/23/2016 7:15:19 PM

Εικόνα 6.1.6

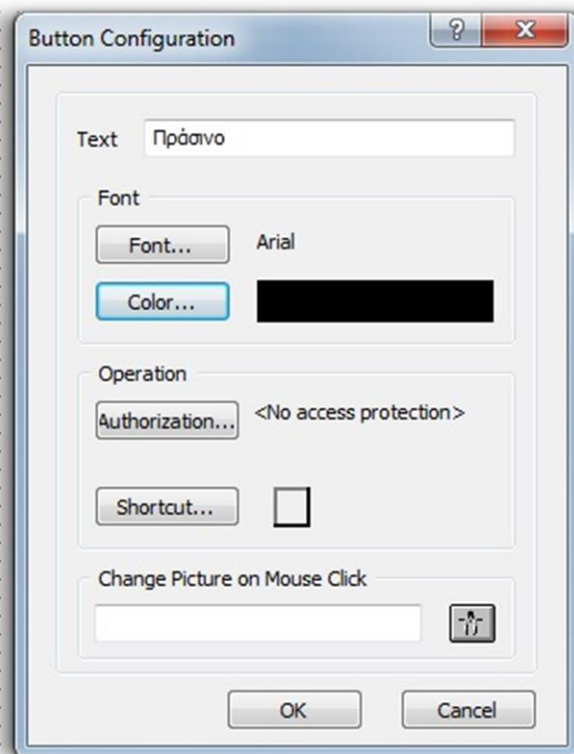


Εικόνα 6.1.7

Προχωρώντας στην υλοποίηση του περιβάλλοντος εργασίας, ανοίγουμε τις εικόνες που έχουμε ήδη δημιουργήσει για να προσθέσουμε τα αντικείμενα όπου θα απαρτίζουν το περιβάλλον εργασίας μας. Αρχικά ορίζουμε το μέγεθος του **Workspace** της εικόνας εκκίνησης με **μήκος 770 και ύψος 460**. Εν συνεχεία, επιλέγουμε από την **Object Palette** του προγράμματος αντικείμενο τύπου **Button** και το προσθέτουμε με **αριστερό κλικ** στο **Workspace** (*εικόνα 6.1.8*). Αυτόματα μας εμφανίζεται ένα παράθυρο βασικών ρυθμίσεων του αντικειμένου, από το οποίο το μετονομάζουμε σε **Πράσινο** στο πεδίο **Text**, όπως διακρίνεται στην *εικόνα 6.1.9*. Επαναλαμβάνουμε την παραπάνω διαδικασία για να δημιουργήσουμε άλλα δύο κουμπιά με ονόματα **Κίτρινο** και **Κόκκινο** αντίστοιχα.

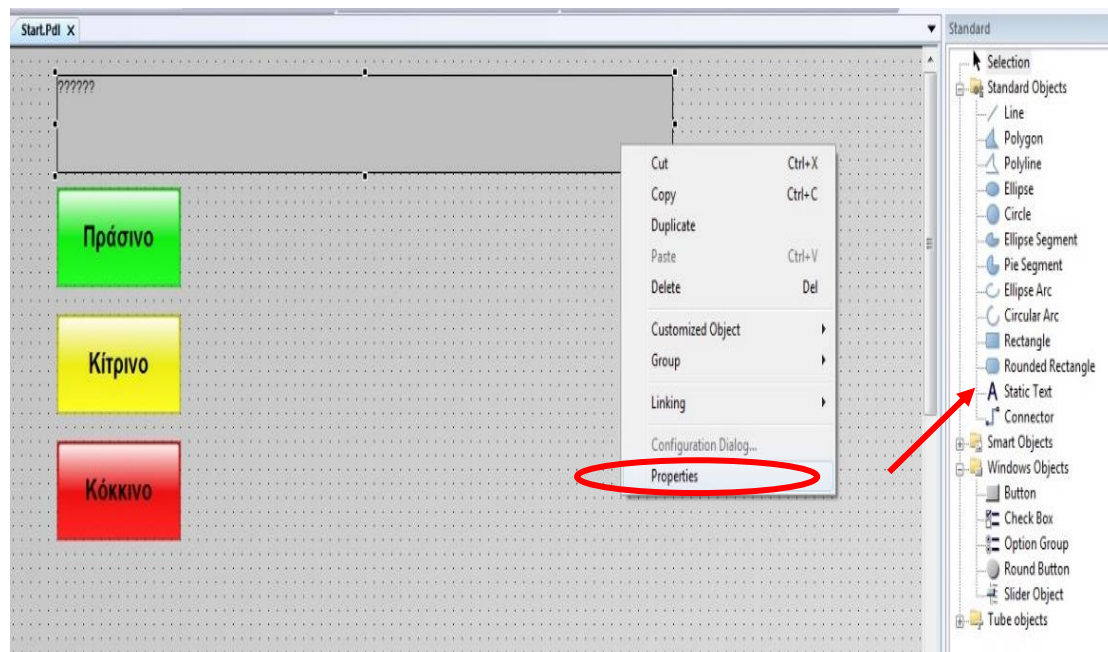


Εικόνα 6.1.8



Εικόνα 6.1.9

Καθώς τα τοποθετήσουμε στο Workspace μας, κάνουμε **δεξί κλικ** πάνω σε ένα από τα κουμπιά και επιλέγουμε το μενού **Properties**, για να παραμετροποιήσουμε την εμφάνισή τους. Για τις ανάγκες του παραδείγματός μας, έχουμε ρυθμίσει την γραμματοσειρά μας σε **Arial** με **έντονη γραφή**, **μεγέθους 20**, χρωματίζοντάς τα ανάλογα με το όνομά τους. Επόμενο αντικείμενο που θα χρειαστούμε είναι μία ετικέτα όπου θα αναφέρεται το όνομα του περιβάλλοντος εργασίας μας. Όπως φαίνεται και στην *εικόνα 6.1.10*, επιλέγουμε από την **Object Palette** το αντικείμενο **Static Text** και το τοποθετούμε κατάλληλα στο **Workspace**. Για να προχωρήσουμε σε αλλαγές στην εμφάνισή του, κάνουμε ομοίως **δεξί κλικ** πάνω του και επιλέγουμε το μενού **Properties**. Για το συγκεκριμένο, η γραμματοσειρά μας θα είναι τύπου **Arial έντονης γραφής, μεγέθους 42**.

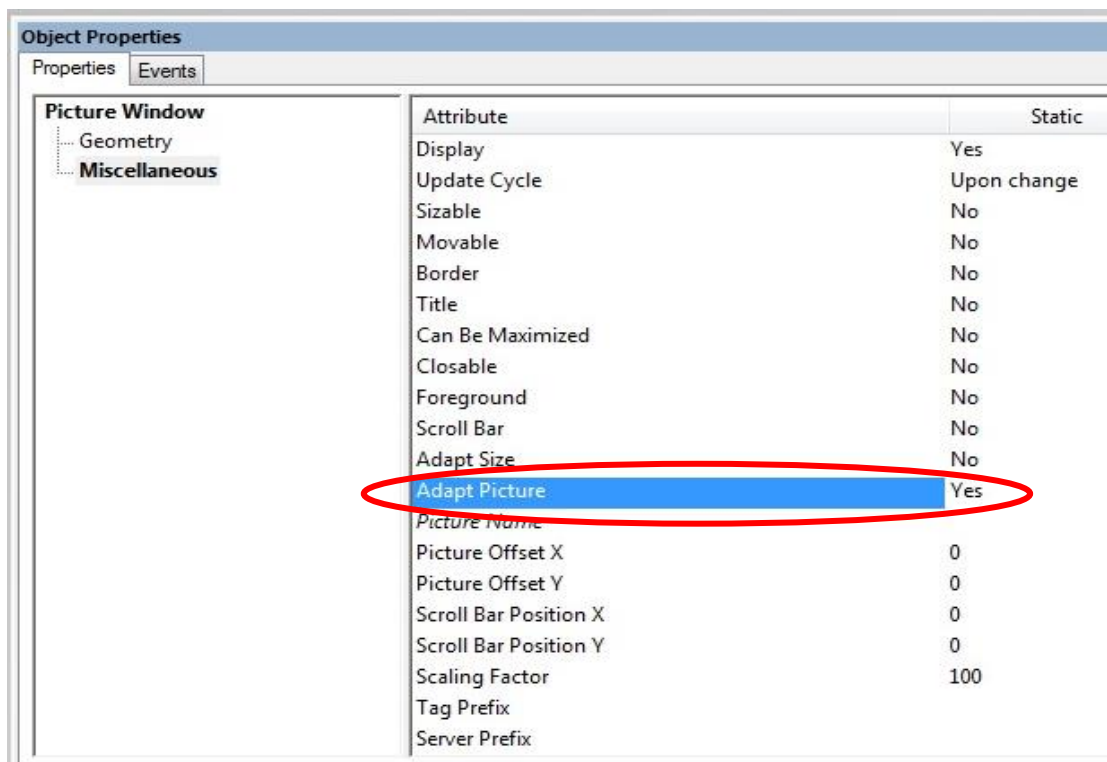


Εικόνα 6.1.10

Με παρόμοιο τρόπο, προσθέτουμε επίσης ένα **παραλληλόγραμμο σχήμα (Rectangle)** χρώματος μαύρου, το οποίο θα περιέχει μέσα του **τρεις κύκλους (Circle)** χρώματος **άσπρου**. Επιπλέον, θα τοποθετήσουμε ένα παράθυρο εικόνων (**Picture Window**) μήκους **310** και ύψους **330**. Το αποτέλεσμα που επιθυμούμε εμφανίζεται στην *εικόνα 6.1.11*. Για να εμφανίζονται κατάλληλα τα μηνύματά μας στο παράθυρο εικόνων, θα πρέπει από τις ιδιότητες του αντικειμένου να ενεργοποιήσουμε την επιλογή **Adapt Picture**, η οποία βρίσκεται στο μενού **Miscellaneous** (*εικόνα 6.1.12*).

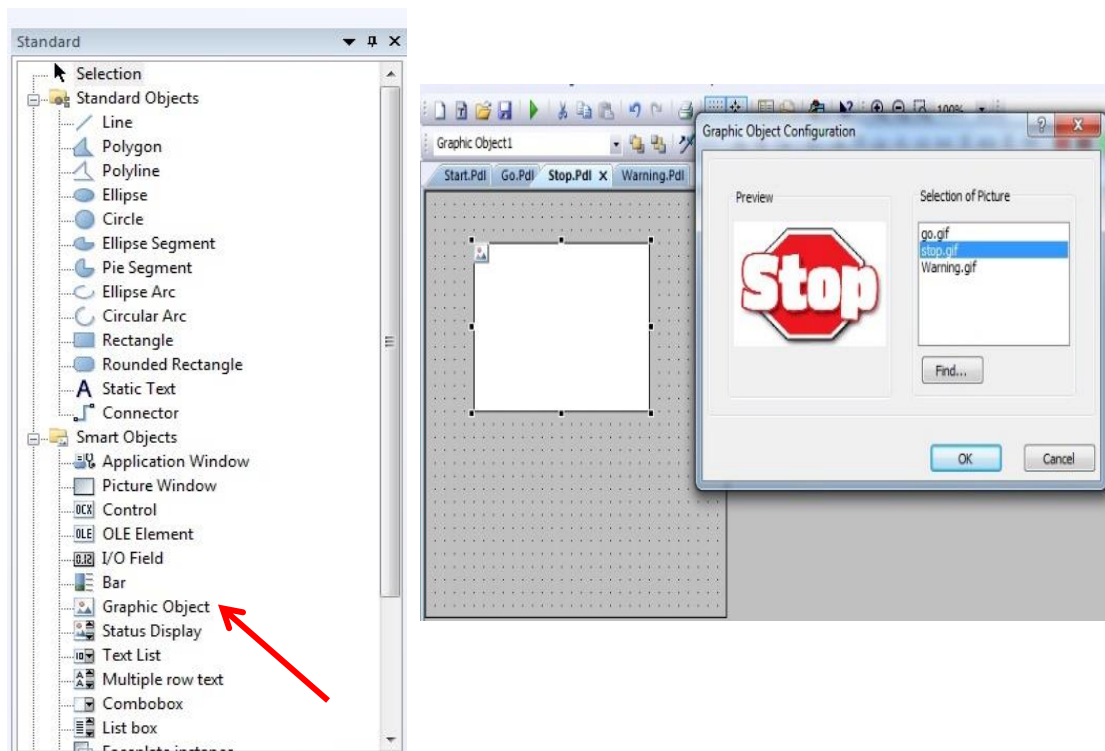


Εικόνα 6.1.11

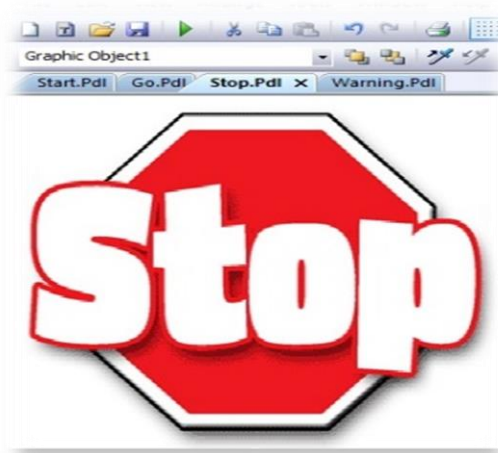


Εικόνα 6.1.12

Ολοκληρώνοντας τις ρυθμίσεις εμφάνισης του περιβάλλοντος εργασίας μας, για να ορίσουμε τα εικονίδια που θα εμφανίζονται με το πάτημα του ανάλογου κουμπιού, ανοίγουμε τις υπόλοιπες εικόνες και σαν πρώτο βήμα αλλάζουμε το μέγεθος της περιοχής **Workspace** κάνοντάς το ίσο με το μέγεθος του παραθύρου εικόνων (**Picture Window**) που έχουμε τοποθετήσει στην αρχική μας εικόνα. Έπειτα, από την **Object Palette** επιλέγουμε το αντικείμενο **Graphic Object** και το τοποθετούμε στην περιοχή εργασίας μας. Αμέσως μας αναδύεται ένα παράθυρο επιλογής για να διαλέξουμε το ανάλογο εικονίδιο προς εμφάνιση (εικόνα 6.1.13). Το αποτέλεσμα που επιθυμούμε παρουσιάζεται στην εικόνα 6.1.14.

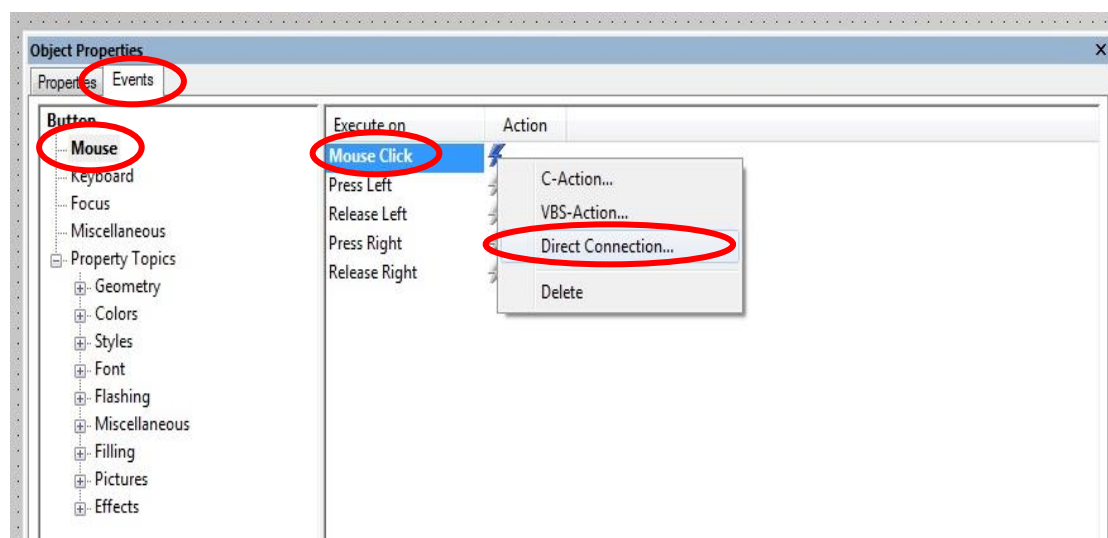


Εικόνα 6.1.13



Εικόνα 6.1.14

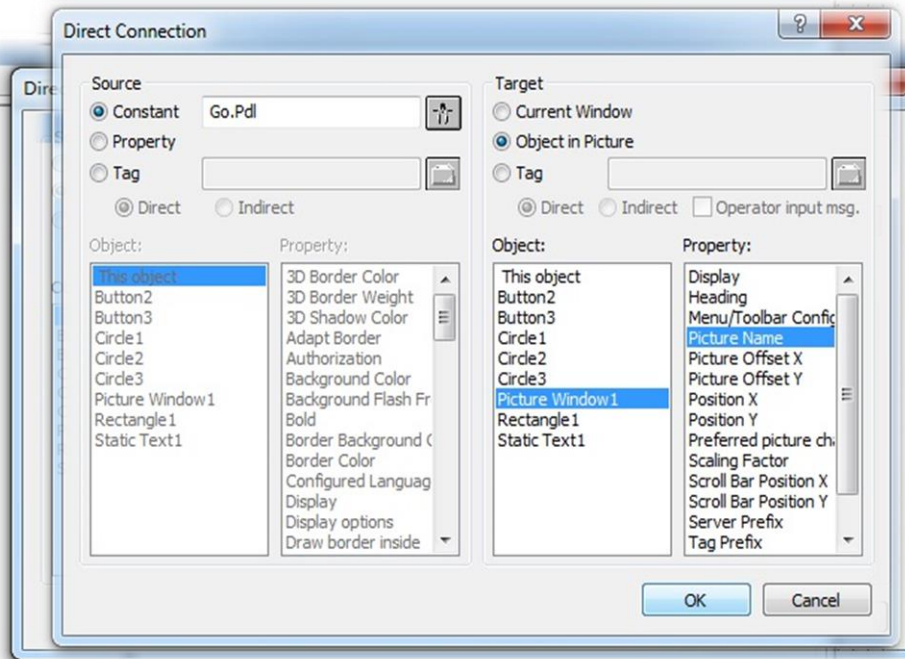
Το τελικό στάδιο της εργασίας μας περιλαμβάνει την ρύθμιση των λειτουργιών που θα επιτελούνται από το πάτημα των κουμπιών. Συγκεκριμένα, κατά το πάτημα ενός κουμπιού το φανάρι θα αλλάζει στο ανάλογο χρώμα καθώς και θα εμφανίζεται το αντίστοιχο εικονίδιο. Για να πραγματοποιηθεί αυτό, επιλέγουμε το ανάλογο κουμπί και από τις ιδιότητες του πάμε στην καρτέλα **Event**, στο μενού **Mouse** και κάνοντας **δεξί κλικ** πάνω στην επιλογή **Mouse Click** επιλέγουμε την λειτουργία **Direct Connection**, όπως αυτό διακρίνεται στην εικόνα 6.1.15.



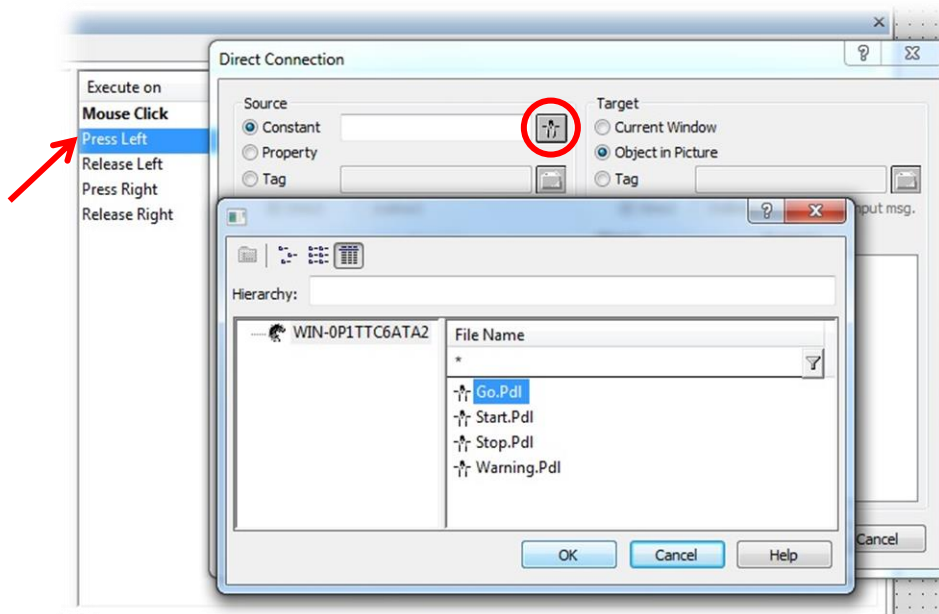
Εικόνα 6.1.15

Στο νέο παράθυρο που θα εμφανιστεί, πραγματοποιούμε την διασύνδεση του κουμπιού με το ανάλογο φανάρι έτσι ώστε κατά το πάτημά του, να αλλάζει το χρώμα του αντίστοιχα. Στο αριστερό μέρος, όπου ορίζεται το αντικείμενο – πηγή, διαλέγουμε την επιλογή **Property** και ορίζουμε από το αντικείμενο που έχουμε επιλέξει, να αντλείται το **χρώμα φόντου του (Background Color)**. Στο δεξί μέρος, όπου ορίζεται το αντικείμενο – προορισμού, επιλέγουμε το **Object in Picture** ενώ παράλληλα ορίζουμε ως προορισμό το αντικείμενο **Circle1** από το οποίο επιθυμούμε να αλλάζει το **χρώμα φόντου** του εξίσου (εικόνα 6.1.16). Κάνοντας τις παραπάνω ρυθμίσεις κάνουμε κλικ στο κουμπί **OK**. Επίσης, για να ορίσουμε και το ανάλογο με το κουμπί εικονίδιο που θέλουμε να εμφανίζεται, κάνουμε **δεξί κλικ** στην επιλογή **Press Left** επιλέγοντας ξανά την λειτουργία **Direct Connection**. Στο παράθυρο που μας εμφανίζεται, στο αριστερό του σκέλος, διαλέξουμε την επιλογή **Constant** και κάνουμε **αριστερό κλικ** στο κουμπί δεξιά του πεδίου. Στο παράθυρο που μας εμφανίζεται, επιλέγουμε το επιθυμητό εικονίδιο και κάνουμε κλικ στο κουμπί **OK**. Στο παράδειγμά μας, κατά το πάτημα του κουμπιού Πράσινο και μας εμφανίζεται η εικόνα Go, του κουμπιού Κίτρινο η εικόνα Warning ενώ στο κουμπί Κόκκινο θα εμφανίζεται η εικόνα

Stop (εικόνα 6.1.17). Στο σκέλος προορισμού, επιλέγουμε το αντικείμενο **Picture Window1** στο οποίο επιθυμούμε να γίνεται αλλαγή του ονόματός του, όπως αυτό είναι εμφανές στην εικόνα 6.1.18. Κάνουμε κλικ στο κουμπί **OK** για να αποθηκευτούν οι ρυθμίσεις μας. Υλοποιούμε τα ίδια ακριβώς βήματα και για τα υπόλοιπα κουμπιά της εφαρμογής μας.

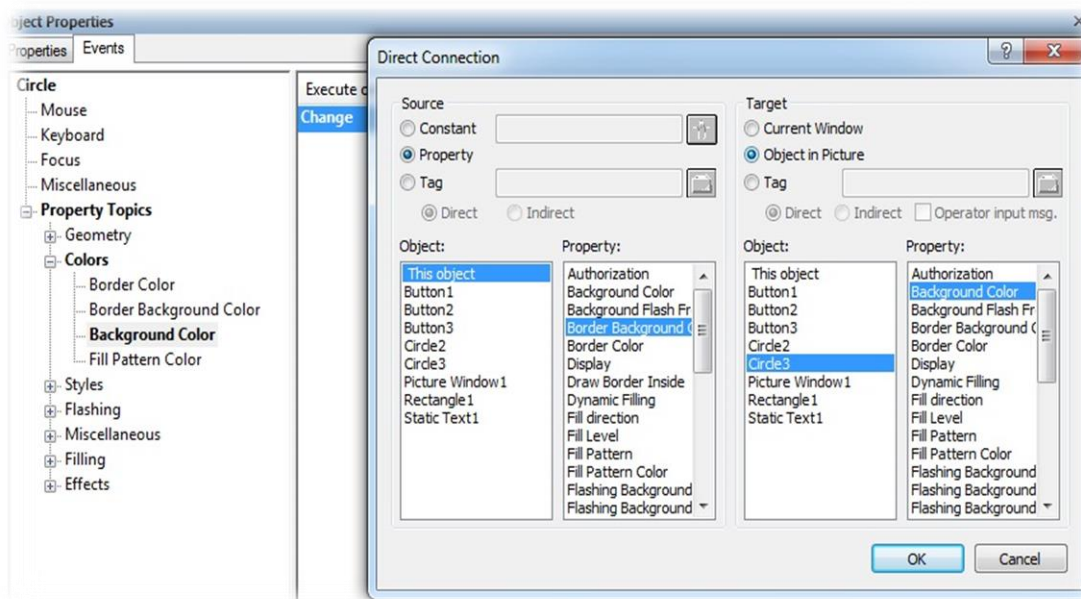


Εικόνα 6.1.18



Εικόνα 6.1.17

Η τελευταία ρύθμιση που απομένει να υλοποιηθεί έχει ως σκοπό το σβήσιμο των φαναριών όταν πατιέται κάποιο κουμπί διαφορετικού χρώματος. Ακολουθώντας παρόμοια διαδικασία με την παραπάνω, επιλέγοντας τις ιδιότητες ενός αντικείμενου τύπου Circle που έχουμε τοποθετήσει, στην καρτέλα **Event**, επιλέγουμε από το μενού **Colors** την λειτουργία **Change** της επιλογής **Background Color**. Καθ' ομοίωση με τα παραπάνω, κάνουμε **δεξί κλικ** και επιλέγουμε το μενού **Direct Connection**. Στο παράθυρο που εμφανίζεται, σαν πηγή ορίζουμε την ιδιότητα **Border Background Color** του επιλεγόμενου αντικείμενου. Σαν προορισμό επιλέγουμε **το χρώμα φόντου** του αντικείμενου τύπου **Circle** το οποίο βρίσκεται τοποθετημένο από πάνω από το επιλεγόμένο μας αντικείμενο (εικόνα 6.1.19). Για να αποθηκευτούν οι αλλαγές μας κάνουμε κλικ στο κουμπί **OK**. Έπειτα ρυθμίζουμε με παρόμοιο τρόπο και τα υπόλοιπα αντικείμενα Circle.



Εικόνα 6.1.19

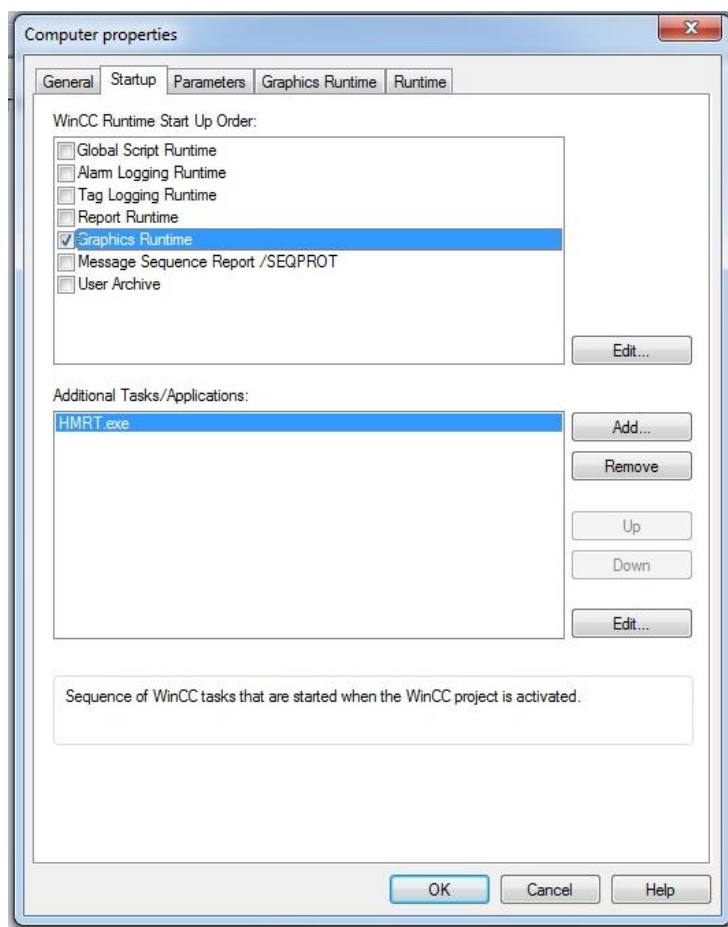
Έχοντας πλέον ορίσει και ρυθμίσει τις λειτουργίες για όλα τα αντικείμενα που προσθέσαμε στην εφαρμογή μας, την αποθηκεύουμε και προχωρούμε στην εκκίνησή του περιβάλλοντος εργασίας. Κάνοντας λοιπόν κλικ στο κουμπί **Play** που βρίσκεται στην μπάρα **Toolbar** του λογισμικού μας, εκκινεί το υποπρόγραμμα **WinCC Runtime** και μας εμφανίζεται σε παράθυρο το παρακάτω περιβάλλον εργασίας (εικόνα 6.1.20).



Εικόνα 6.1.20

6.2 Εποπτεία και έλεγχος στάθμης δεξαμενής σε μία βιομηχανική μονάδα

Για να δημιουργήσουμε το τρέχων project, ακολουθούμε τα βήματα όπως τα έχουμε αναφέρει και στην προηγούμενη ενότητα. Συγκεκριμένα, ανοίγουμε το πρόγραμμα WinCC Explorer, είτε από το μενού Έναρξη είτε από το εικονίδιο στην επιφάνεια εργασίας, και δημιουργούμε ένα νέο **single – user project**. Στην συνέχεια, του δίνουμε το όνομα **Tank Fill** και το αποθηκεύουμε στο προεπιλεγμένο path. Μόλις ολοκληρωθεί η δημιουργία του project, ακολουθώντας με την μεθοδολογία μας, θα πρέπει να προχωρήσουμε στην ρύθμιση των ιδιοτήτων του υπολογιστή της εφαρμογής μας. Κάνοντας δεξί κλικ στον υπολογιστή που εμφανίζεται στο μενού **Computer**, στην περιοχή δεδομένων, μας παρουσιάζονται όλες οι ιδιότητες του υπολογιστικού συστήματος του projects μας. Σαν πρώτη κίνηση, επιβεβαιώνουμε ότι το όνομα του υπολογιστή στο πεδίο **Computer Name** συμπίπτει με το όνομα του συστήματος όπου εργαζόμαστε. Έπειτα, στην καρτέλα **Startup**, ελέγχουμε εάν οι ρυθμίσεις σχετικά με τα υποπρογράμματα που επιθυμούμε να εκκινούν κατά την λειτουργία του project, είναι οι επιθυμητές (εικόνα 6.2.1).



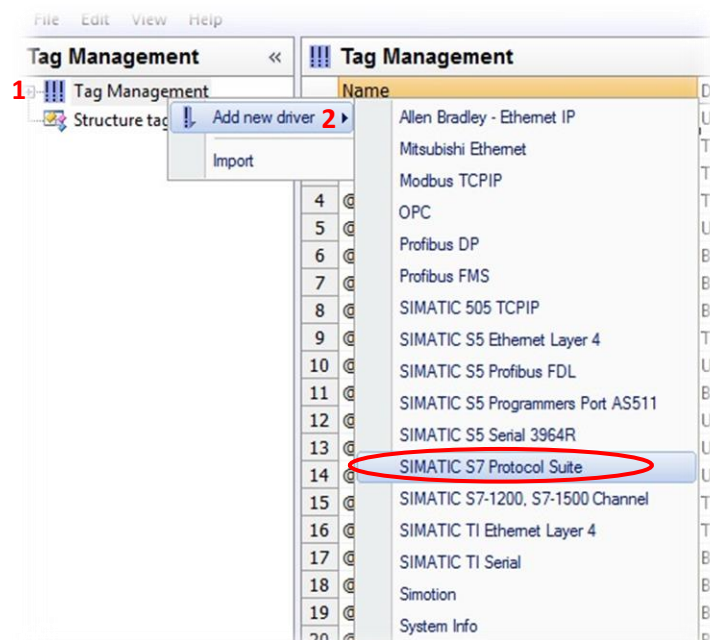
Εικόνα 6.2.1

Επαναλαμβάνουμε την ίδια κίνηση και για τις ζώνες ώρας τα εφαρμογής, στην καρτέλα **Parameters**. Προχωρώντας στην καρτέλα **Graphics Runtime**, στην λίστα **Window Attributes**, ενεργοποιούμε όλες τις επιλογές εκτός της **FullScreen** και της **Scroll Bar**. Όπως και στο προηγούμενο παράδειγμα, η εφαρμογή μας επιθυμούμε να εμφανίζεται σε παραθυρικό περιβάλλον και όχι να καλύπτει όλο το φάσμα της οθόνης μας. Ολοκληρώνοντας τα ρυθμίσεις αυτές, κάνουμε κλικ στο κουμπί **OK** για να προχωρήσουμε παρακάτω. Επόμενη μας κίνηση είναι η εγκατάσταση των οδηγών/driver και των μεταβλητών για την διασύνδεση του προγράμματος WinCC με τα περιφερειακά αισθητήρια (PLC, IED, RTU). Για να πραγματοποιηθεί αυτό, θα πρέπει να ανοίξουμε το υποπρόγραμμα **Tag Management** κάνοντας διπλό κλικ πάνω στην επιλογή που βρίσκεται στο παράθυρο πλοήγησης του προγράμματος, όπως αυτό εμφανίζεται στην *εικόνα 6.2.2*.



Εικόνα 6.2.2

Αμέσως, μας εμφανίζεται το παράθυρο λειτουργίας του προγράμματος. Για να πραγματοποιήσουμε την εγκατάσταση των απαραίτητων drivers για την εφαρμογή μας, κάνουμε δεξί κλικ στην κατηγορία **Tag Management**, όπως αυτή εμφανίζεται στα αριστερά της οθόνης μας.



Εικόνα 6.2.3

Έπειτα, στα μενού όπου εμφανίζονται, επιλέγουμε την κατηγορία **Add Driver** και κάνουμε κλικ στην επιλογή **SIMATIC S7 Protocol Suite**, όπως αναλυτικά παρουσιάζεται στην *εικόνα 6.2.3*.

Ορίζοντας ότι η σύνδεση μεταξύ του λογισμικού και των περιφερειακών αισθητηρίων θα πραγματοποιηθεί με την διεπαφή **MPI**, κάνουμε δεξί κλικ πάνω της και επιλέγουμε το μενού **New Connection** (εικόνα 6.2.4). Για την εκπόνηση της πτυχιακής μας εργασίας, ονομάζουμε την νέα σύνδεση που δημιουργήσαμε **Tank_1**.

Επόμενό μας βήμα είναι να ορίσουμε τις απαραίτητες εξωτερικές αλλά και εσωτερικές μεταβλητές, για να εξασφαλίσουμε την ορθή σύνδεση και λειτουργία του project μας. Έχοντας ήδη δημιουργήσει την σύνδεση με την διεπαφή μας, κάνουμε πάνω της δεξί κλικ και επιλέγουμε το μενού **New Group**, για να δημιουργήσουμε ένα Group για τις εξωτερικές μας μεταβλητές (external tags). Στο παράδειγμά μας, το ονομάζουμε **Ex_Group1** (εικόνα 6.2.5).



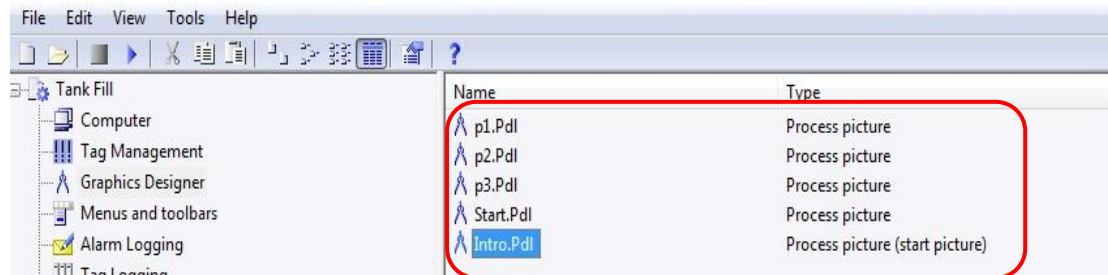
Εικόνα 6.2.5

Στην περιοχή δεδομένων που έχει εμφανιστεί πλέον αριστερά του παραθύρου μας, καταχωρούμε την εξωτερική μας μεταβλητή. Αναλυτικότερα, στο πεδίο **Name** εισάγουμε το όνομα **Ex_Tank_Fill**, στο πεδίο **Data Type** κάνουμε κλικ στην επιλογή **Signed 16-bit value** ενώ στο πεδίο **Address** επιλέγουμε το **Bit Memory**, από την λίστα **Data** όπου εμφανίζεται, και πατάμε το κουμπί **OK** για να αποθηκευτούν οι ρυθμίσεις μας. Παράλληλα, κάνουμε κλικ στο πεδίο **Linear Scaling** συμπληρώνοντας τα κελιά **AS value range from**, **AS value range to** και **OS value range to** με τις τιμές **-20**, **20** και **100** αντίστοιχα. Ολοκληρώνοντας την δημιουργία της εξωτερικής μας μεταβλητής, θα πρέπει να έχουμε το παρακάτω αποτέλεσμα, όπως παρουσιάζεται στην εικόνα 6.2.6.

Ex_Group1												
Name	Data Type	Length	Format adaptation	Connection	Group	Address	Linear scaling	AS value range from	AS value range to	OS value range from	OS value	
1 Ex_Tank_Fill	Signed 16-bit value	2	ShortToSignedWord	Tank_1	Ex_Group1	MW0	<input checked="" type="checkbox"/>	-20	20	0	100	
2												
3												

Εικόνα 6.2.6

Προχωρώντας πλέον στην δημιουργία της εσωτερικής μας μεταβλητής (internal tags), κάνουμε αριστερό κλικ στην επιλογή **Internal Tags**, στην περιοχή πλοήγησης του προγράμματος. Στο παράθυρο όπου μας εμφανίζεται, συμπληρώνουμε στην τελευταία σειρά της λίστας τα στοιχεία της μεταβλητής μας. Συγκεκριμένα, στο πεδίο **Name** πληκτρολογούμε το όνομα **Tank_Fill** ενώ στο πεδίο **Data Type** κάνουμε κλικ στην επιλογή **Unsigned 16-bit value**, ολοκληρώνοντας πλέον την δημιουργία των μεταβλητών μας (εικόνα 6.2.7).

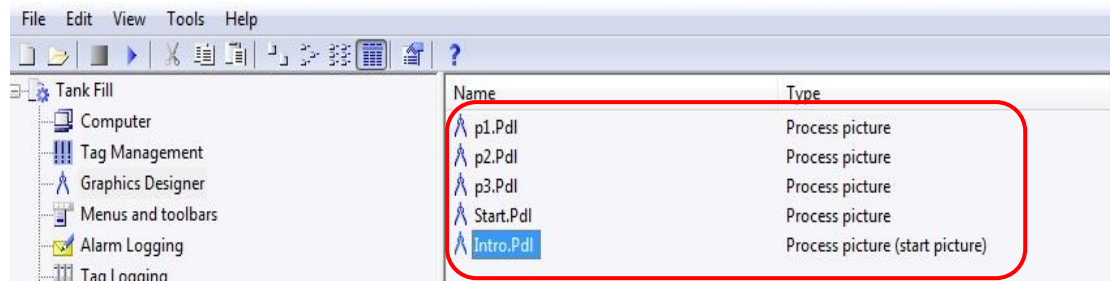


Εικόνα 6.2.8

	Name	Data Type	Length	Format adaptation	Connection	Group	Address	Linear scaling /
1	@ConnectedRT	Unsigned 16-bit value	2		Internal tags			<input type="checkbox"/>
2	@CurrentUser	Text tag 8-bit character set	255		Internal tags			<input type="checkbox"/>
3	@CurrentUser†	Text tag 16-bit character set	255		Internal tags			<input type="checkbox"/>
4	@DatasourceNi	Text tag 16-bit character set	255		Internal tags			<input type="checkbox"/>
5	@DeltaLoaded	Unsigned 32-bit value	4		Internal tags			<input type="checkbox"/>
6	@HornReset	Binary Tag	1		Internal tags Horn			<input type="checkbox"/>
7	@HornSettingT	Binary Tag	1		Internal tags Horn			<input type="checkbox"/>
8	@HornTriggerri	Binary Tag	1		Internal tags Horn			<input type="checkbox"/>
9	@LocalMachine	Text tag 8-bit character set	255		Internal tags			<input type="checkbox"/>
10	@RedundantSe	Unsigned 16-bit value	2		Internal tags			<input type="checkbox"/>
11	@RestartHorn	Binary Tag	1		Internal tags Horn			<input type="checkbox"/>
12	@SCRIPT_COU	Unsigned 32-bit value	4		Internal tags Script			<input type="checkbox"/>
13	@SCRIPT_COU	Unsigned 32-bit value	4		Internal tags Script			<input type="checkbox"/>
14	@SCRIPT_COU	Unsigned 32-bit value	4		Internal tags Script			<input type="checkbox"/>
15	@ServerName	Text tag 16-bit character set	255		Internal tags			<input type="checkbox"/>
16	@ServerVersio	Text tag 16-bit character set	255		Internal tags			<input type="checkbox"/>
17	@Signal1	Binary Tag	1		Internal tags Horn			<input type="checkbox"/>
18	@Signal2	Binary Tag	1		Internal tags Horn			<input type="checkbox"/>
19	@Signal3	Binary Tag	1		Internal tags Horn			<input type="checkbox"/>
20	@SignalInput1	Binary Tag	1		Internal tags Horn			<input type="checkbox"/>
21	@SignalInput2	Binary Tag	1		Internal tags Horn			<input type="checkbox"/>
22	@SignalInput3	Binary Tag	1		Internal tags Horn			<input type="checkbox"/>
23	@TLGRT_AVER	Floating-point number 64-bit IEEE 754	8		Internal tags TagLoggingF			<input type="checkbox"/>
24	@TLGRT_SIZE	Floating-point number 64-bit IEEE 754	8		Internal tags TagLoggingF			<input type="checkbox"/>
25	@TLGRT_SIZE	Floating-point number 64-bit IEEE 754	8		Internal tags TagLoggingF			<input type="checkbox"/>
26	@TLGRT_TAG	Floating-point number 64-bit IEEE 754	8		Internal tags TagLoggingF			<input type="checkbox"/>
27	Tank_Fill	Unsigned 16-bit value	2		Internal tags			<input type="checkbox"/>

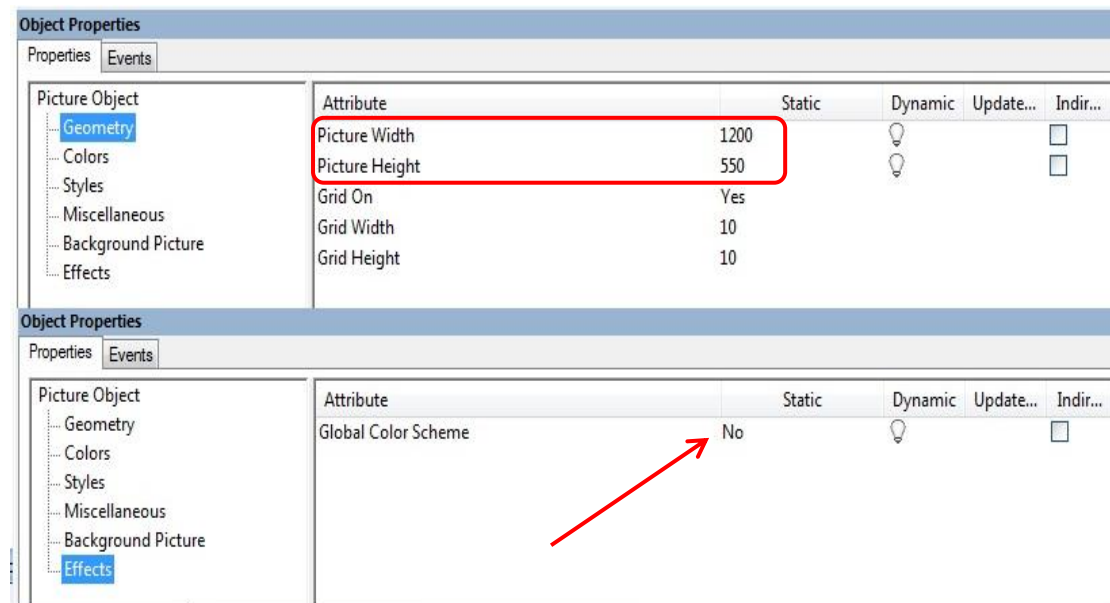
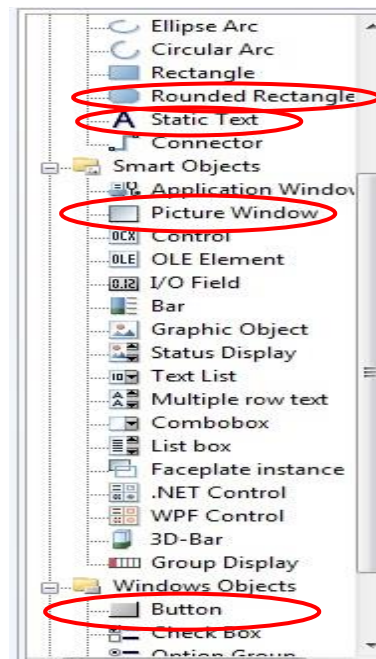
Εικόνα 6.2.7

Στην συνέχεια θα πρέπει να δημιουργήσουμε τις εικόνες που θα απαρτίζουν το project μας. Με βάση την μεθοδολογία που έχει αναλυθεί σε παραπάνω ενότητες, για το εν λόγω παράδειγμα θα δημιουργήσουμε **5 εικόνες** με ονόματα **Intro**, **Start**, **p1**, **p2** και **p3** αντίστοιχα (εικόνα 6.2.8). Η εικόνα **Intro** θα αποτελεί την εικόνα έναρξης της εφαρμογής μας, η εικόνα **Start** την εικόνα χειρισμού και λειτουργίας της δεξαμενής ενώ οι εικόνες **p1**, **p2** και **p3** θα παίζουν τον ρόλο εικονιδίων που θα εμφανίζονται στην εικόνα Start.



Εικόνα 6.2.8

Σαν πρώτο βήμα, κάνουμε διπλό κλικ στην εικόνα **Intro**, για να ενσωματώσουμε τα ανάλογα αντικείμενα που θα απαρτίζουν το περιβάλλον εργασίας μας μέσω του προγράμματος **Graphics Designer**. Ορίζουμε τις διαστάσεις της περιοχής **Workspace** σε **1200 x 550**, απενεργοποιώντας επίσης την επιλογή **Global Color Scheme** από το μενού **Effects** και δίνοντας ένα χρώμα γαλάζιο σε όλη την περιοχή (εικόνα 6.2.9). Παράλληλα, με την βοήθεια της **Object Palette**, τοποθετούμε στο **Workspace** μας **3 Static Text**, **2 Buttons**, **1 Rounded Rectangle** και ένα **Picture Window** με τις ανάλογες διαστάσεις και μηνύματα, όπως εμφανίζονται στην εικόνα 6.2.10.

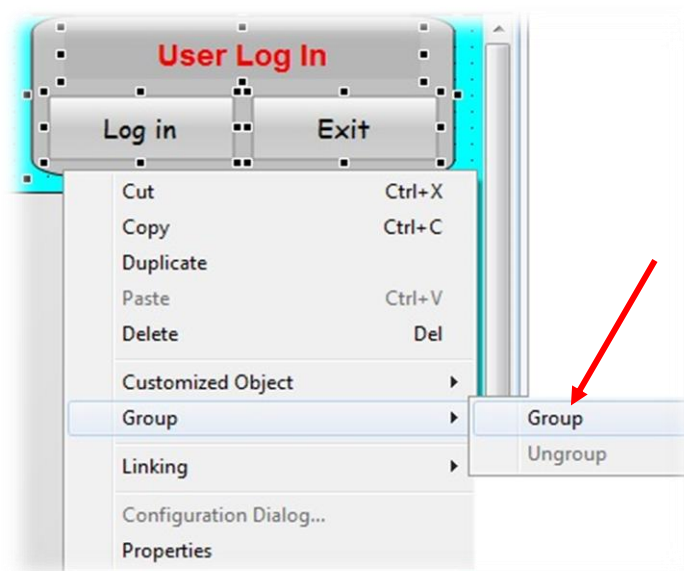


Εικόνα 6.2.9



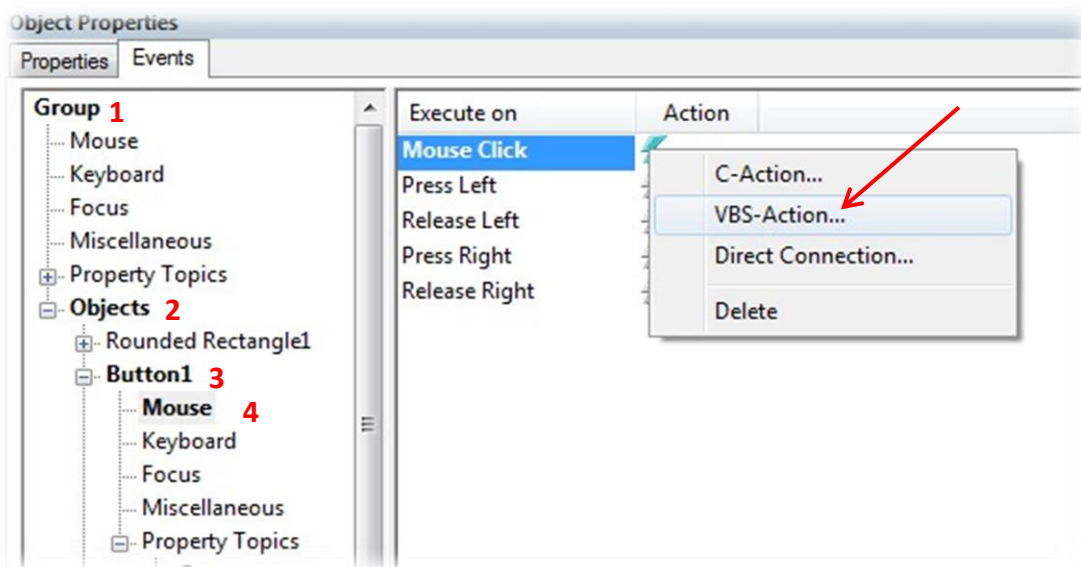
Εικόνα 6.2.10

Για να μετακινήσουμε πιο εύκολα πολλά αντικείμενα μαζί, μπορούμε να ομαδοποιήσουμε μερικά από αυτά. Όπως φαίνεται και στην εικόνα 6.2.11, επιλέγοντάς τα με **αριστερό κλικ** και ταυτόχρονα πατημένο το **πλήκτρο Shift** από το πληκτρολόγιό μας, κάνουμε **δεξί κλικ** πάνω στα επιλεγόμενα αντικείμενα και επιλέγουμε την λειτουργία **Group**.

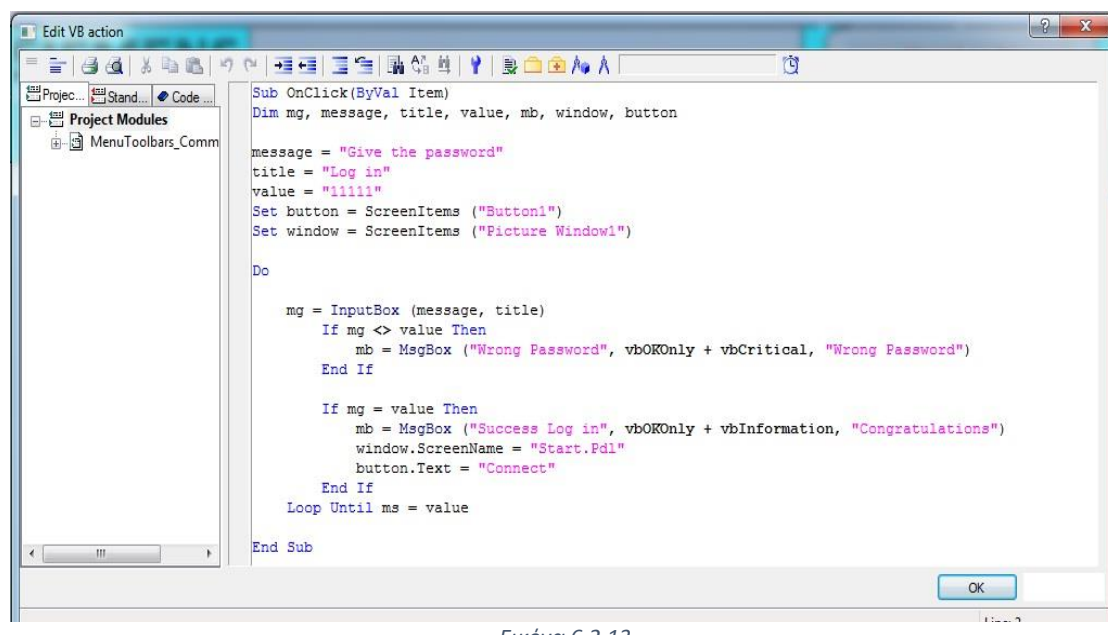


Εικόνα 6.2.11

Ολοκληρώνοντας τις ρυθμίσεις μας στην εικόνα **Intro**, θα προσθέσουμε τις ανάλογες ενέργειες που επιθυμούμε να επιτελούν τα κουμπιά μας, μέσω της γλώσσας προγραμματισμού **VBS** και συγκεκριμένα με την βοήθεια του υποπρογράμματος **Global Script**. Κάνουμε **δεξί κλικ** πάνω στο group αντικειμένων και επιλέγουμε το μενού **Properties**. Από εκεί, κάνουμε κλικ στην καρτέλα **Events** και από την κατηγορία **Objects** επιλέγουμε το μενού **Mouse** του αντικειμένου **Button1**. Στην εντολή **Mouse Click**, κάνουμε δεξί κλικ πάνω στο εικονίδιο του κεραυνού και επιλέγουμε το μενού **VBS – Action** (εικόνα 6.2.12). Αυτομάτως μας εμφανίζεται το περιβάλλον εργασίας του προγράμματος **Global Script** (εικόνα 6.2.13) στο οποίο θα περάσουμε το πρόγραμμα που εμφανίζεται παρακάτω.



Εικόνα 6.2.12



Εικόνα 6.2.13

```
Sub OnClick (ByVal Item)
Dim mg, message, title, value, mb, window, button

message = "Give the password"
title = "Log in"
value = "11111"
Set button = ScreenItems ("Button1")
Set window = ScreenItems ("Picture Window1")

Do

    mg = InputBox (message, title)

    If mg <> value Then
        mb = MsgBox ("Wrong Password", vbOKOnly + vbCritical, "Wrong
        Password")

    End If

    If mg = value Then
        mb = MsgBox ("Success Log in", vbOKOnly + vbInformation,
        "Congratulations")
        window.ScreenName = "Start.Pdl"
        button.Text = "Connect"

    End If

Loop Until ms = value

End Sub
```

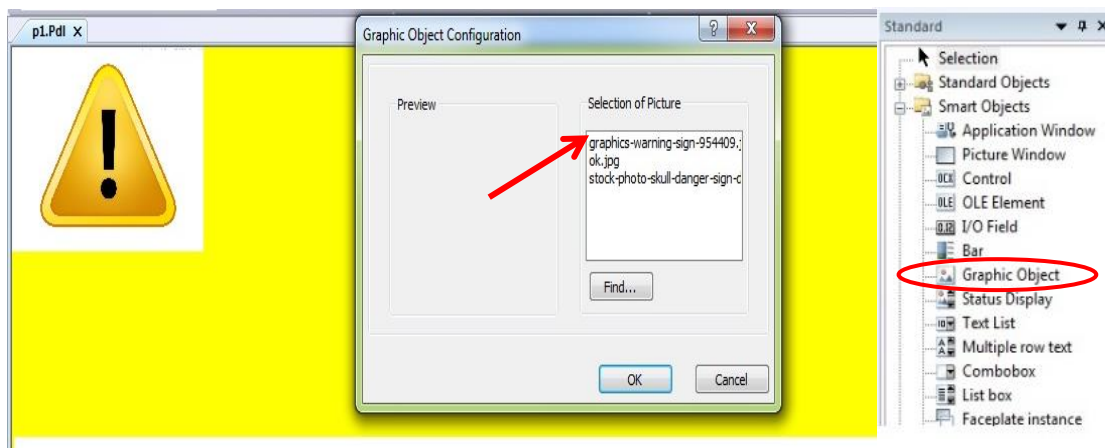
Συνοπτικά, με το πάτημα του κουμπιού **Log In**, θα εμφανίζεται στον χρήστη ένα παράθυρο σύνδεσης. Ο χρήσης θα πρέπει να πληκτρολογήσει τον κατάλληλο κωδικό (11111) για να εισαχθεί στο περιβάλλον ελέγχου. Κάνουμε τις ίδιες ενέργειες και για τον προγραμματισμό του κουμπιού **Button2 (Exit)**. Το κουμπί αυτό θα έχει την λειτουργία της εξόδου από την εφαρμογή μας. Το πρόγραμμα για το κουμπί **Exit** βρίσκεται παρακάτω

```
Sub OnClick (ByVal Item)

    HMIRuntime.Stop

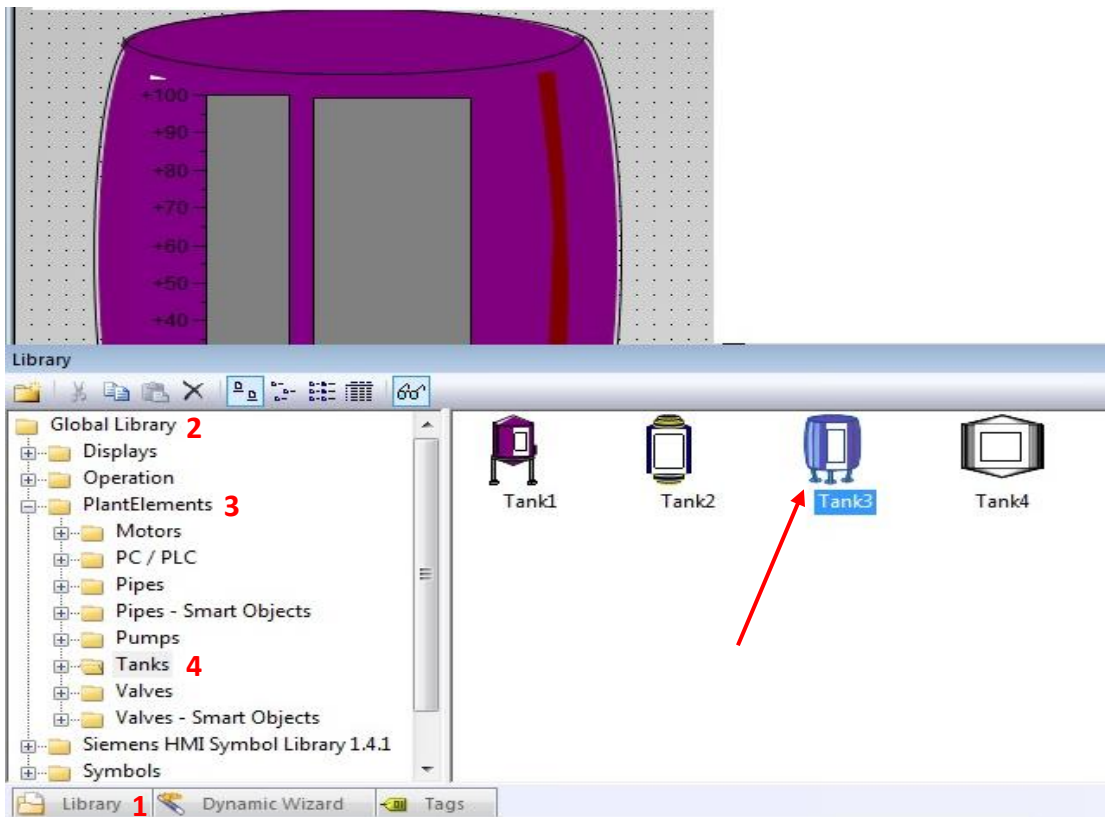
End Sub
```

Επόμενο μας βήμα είναι η ρύθμιση των εικόνων p1, p2 και p3, τα όποια όπως προαναφέραμε, θα έχουν τον ρόλο εικονιδίου ενημέρωσης του χρήστη. Για να προσθέσουμε τα ανάλογα αντικείμενα στις εικόνες, τις ανοίγουμε κατ' αρχάς και ορίζουμε το **Workspace** με διαστάσεις **190 x 150**. Εν συνεχεία εισάγουμε ένα **Graphic Object** από την εργαλειοθήκη **Object Palette**. Κάνοντας κλικ πάνω στην περιοχή **Workspace**, μας εμφανίζεται ένα αναδυόμενο παράθυρο επιλογής της εικόνας που επιθυμούμε. Στο παράδειγμά μας, για την εικόνα **p1**, κάνουμε κλικ πάνω στην εικόνα **Warning** και πατάμε το κουμπί **OK** για να τοποθετηθεί (εικόνα 6.2.14). Επαναλαμβάνουμε τις ίδιες ενέργειες και για τις εικόνες **p2** και **p3**.

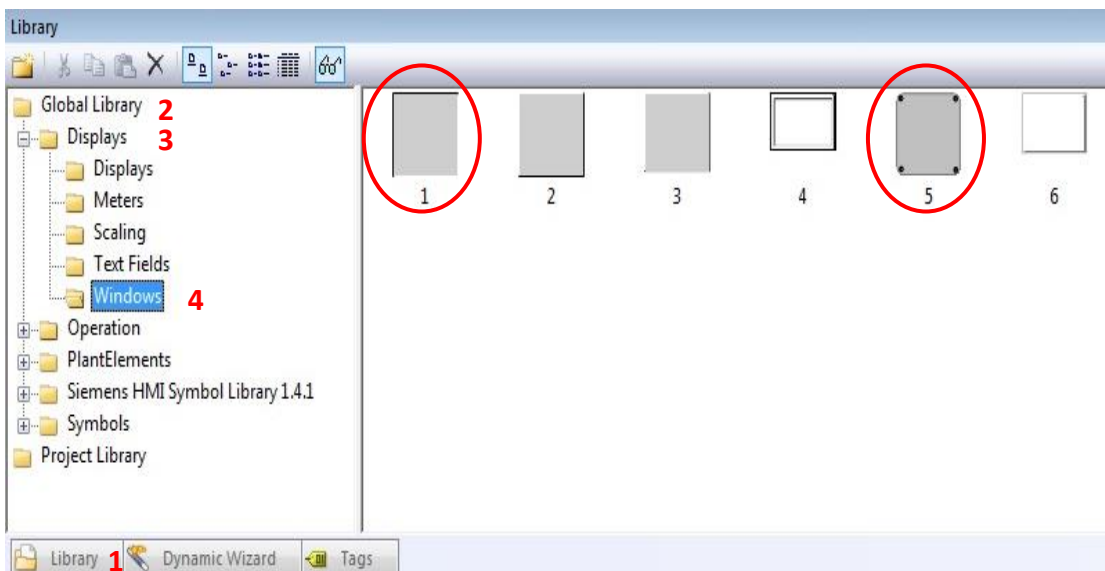


Εικόνα 6.2.14

Για το τέλος αφήσαμε την ρύθμιση της πιο βασικής εικόνας μας, της εικόνας **Start**. Ανοίγοντάς την ορίζουμε τις διαστάσεις του **Workspace** σε **1000 x 450**, για να επιτύχουμε την άρτια εμφάνισή της στην περιοχή **Picture Window** της εικόνας **Intro**. Στην συνέχεια, για να προσθέσουμε την δεξαμενή στο περιβάλλον εργασίας μας, κάνουμε κλικ στο μενού **Library** και επιλέγουμε την κατηγορία **Global Library** το μενού **Plant Elements**. Έπειτα, από τον φάκελο **Tanks** επιλέγουμε την δεξαμενή **Tank3** (εικόνα 6.2.15). Από την κατηγορία **Global Library** θα χρειαστούμε επίσης για λόγους εμφάνισης τα **Windows 1** και **5** όπου βρίσκονται στον φάκελο **Displays** (εικόνα 6.2.16).



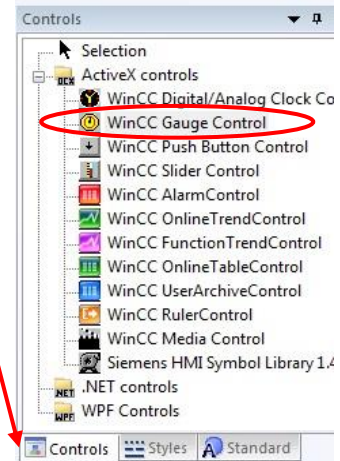
Εικόνα 6.2.15



Εικόνα 6.2.16

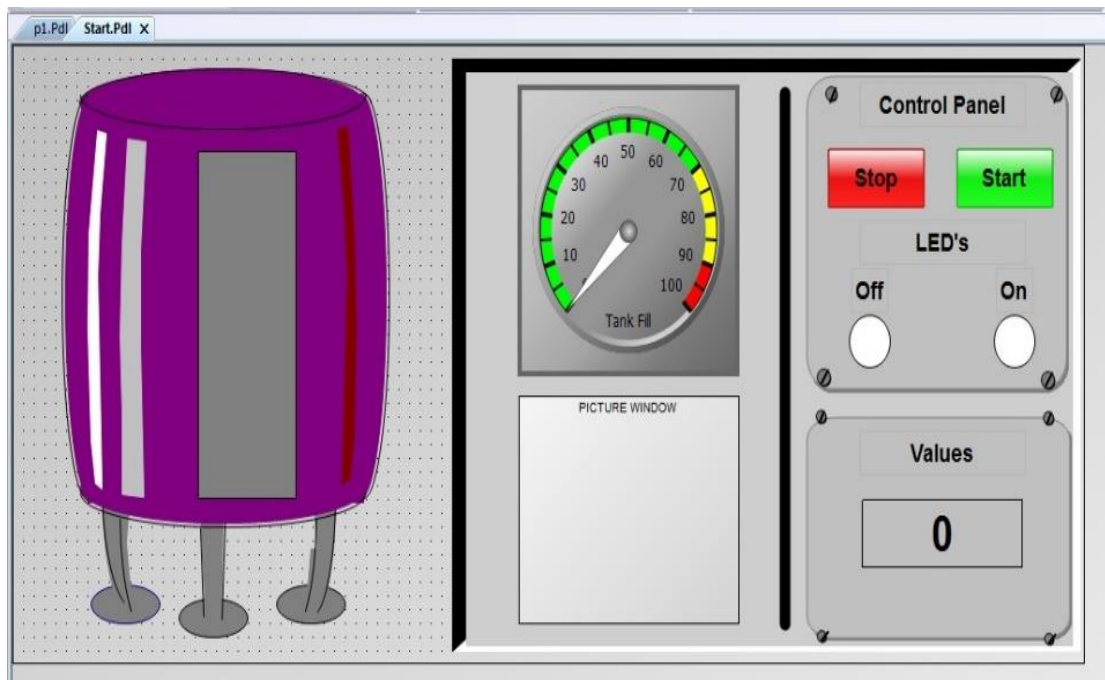
Μέσα στο Window 1, εκτός από 2 Window 5, θα τοποθετήσουμε επίσης τα εξής αντικείμενα:

- ❖ 2 Buttons λειτουργίας **ON & OFF**
- ❖ 5 Static Texts όπου θα έχουν τον ρόλο ετικετών
- ❖ 2 Circles όπου θα έχουν τον ρόλο φωτεινών ενδείξεων λειτουργίας
- ❖ 1 I/O Field για την μέτρηση της δεξαμενής
- ❖ 1 Picture Window όπου θα εμφανίζει τις εικόνες p1, p2 και p3, αναλόγως με την στάθμη της δεξαμενής
- ❖ 1 WinCC Gauge Control για να βλέπουμε την στάθμη της δεξαμενής. Το συγκεκριμένο αντικείμενο βρίσκεται στον φάκελο **ActiveX Controls** του μενού **Controls** της Object Palette (εικόνα 6.2.17)



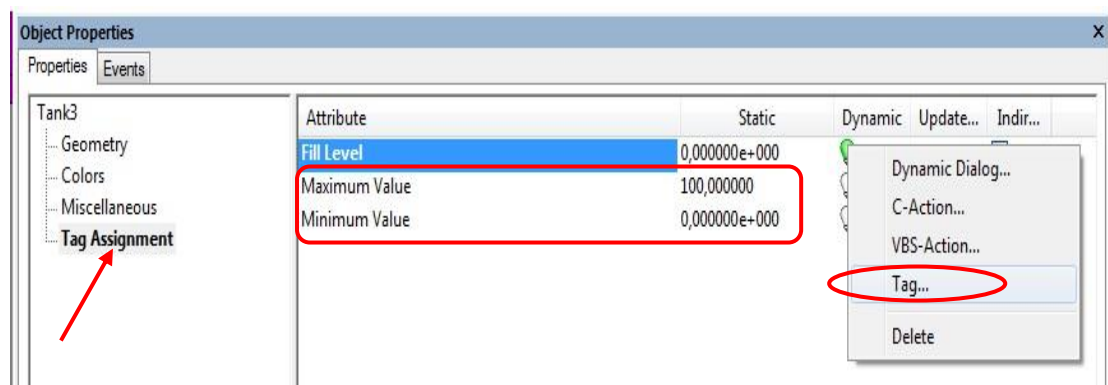
Εικόνα 6.2.17

Τελικώς, η εικόνα **Start** θα έχει την μορφή όπως αυτή παρουσιάζεται στην εικόνα 6.2.18.



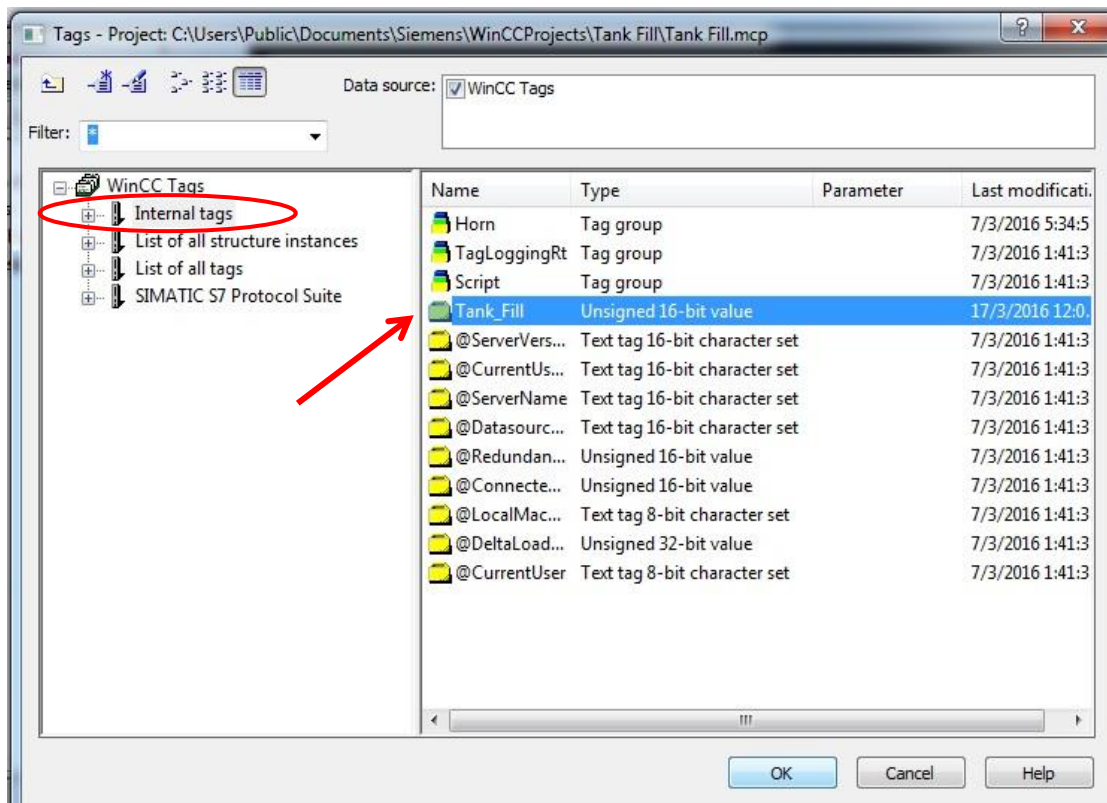
Εικόνα 6.2.18

Προχωρώντας στην υλοποίηση του περιβάλλοντος εργασίας μας για την εποπτεία της στάθμης της δεξαμενής, θα πρέπει να ορίσουμε τις λειτουργίες όπου θα συντελούν τα αντικείμενα που μόλις προσθέσαμε. Αναλυτικότερα, ο χρήστης πατώντας το **κουμπί Start** θα ξεκινάει το γέμισμα της δεξαμενής μας. Αυτόματα, θα εμφανίζεται η στάθμη της καθώς και το ανάλογο εικονίδιο. Φτάνοντας στο επίπεδο του 70% της συνολικής χωρητικότητας της δεξαμενής, ο χρήστης θα λαμβάνει προειδοποιητικό μήνυμα. Το γέμισμα τελειώνει είτε όταν η δεξαμενή γεμίσει εντελώς είτε με το πάτημα του **κουμπιού Stop**. Κατά πρώτο λόγο, θα πρέπει να συνδέσουμε την δεξαμενή μας καθώς και τις συσκευές μέτρησης με την εσωτερική μας μεταβλητή (internal tag), για να λαμβάνουν αυτομάτως τις μετρήσεις για την στάθμη της δεξαμενής. Ξεκινώντας με το αντικείμενο **Tank**, κάνουμε **δεξί κλικ** πάνω του και επιλέγουμε το μενού **Properties**. Εν συνεχεία, στο μενού **Tag Assignment** της καρτέλας **Properties** εισάγουμε την μέγιστη και την ελάχιστη τιμή που θέλουμε να λαμβάνει η δεξαμενή. Συγκεκριμένα, στα πεδία **Maximum** και **Minimum Value** εισάγουμε τις τιμές **100** και **0** αντίστοιχα. Έπειτα, στο πεδίο **Fill Level**, κάνουμε **δεξί κλικ** στο εικονίδιο λάμπας που υπάρχει στην στήλη **Dynamic** και επιλέγουμε την λειτουργία **Tag** (εικόνα 6.2.19).

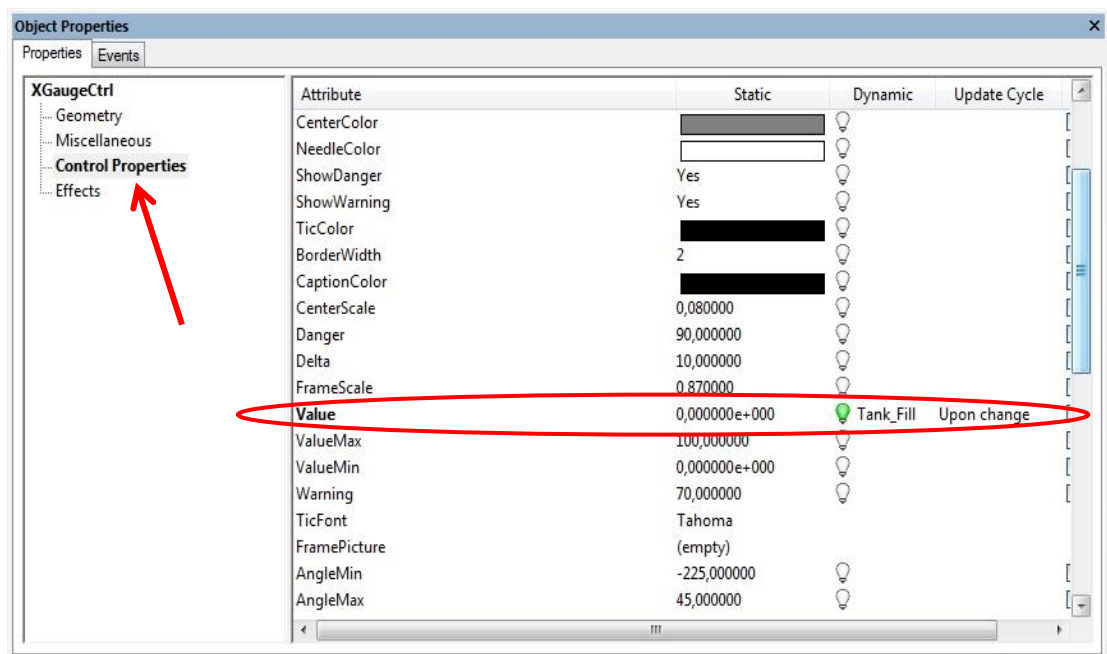


Εικόνα 6.2.19

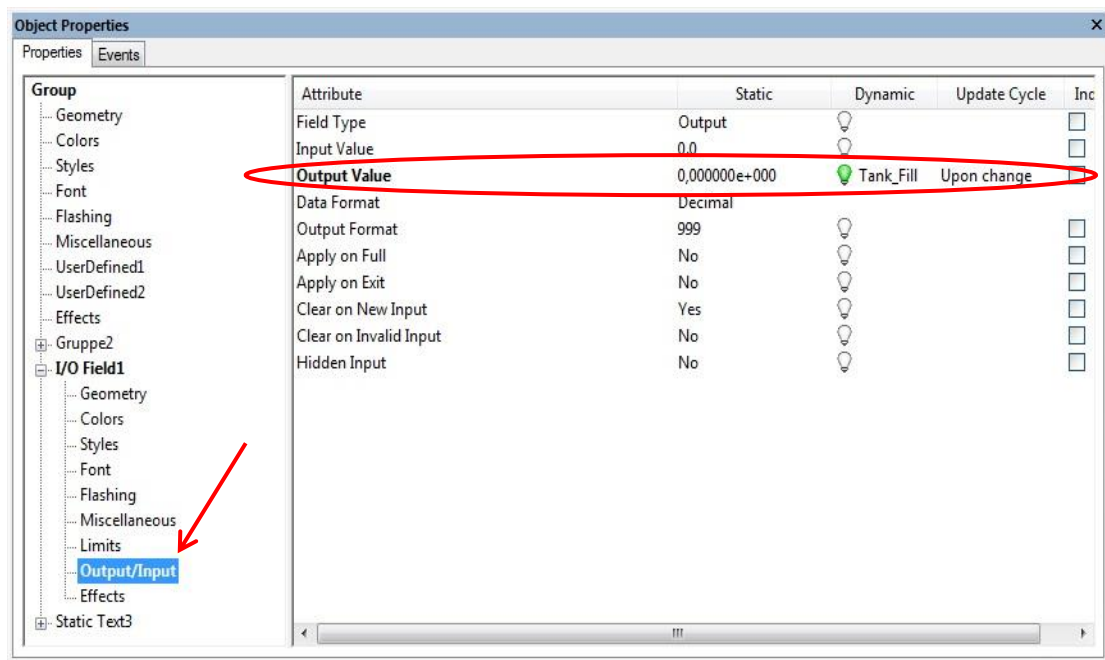
Από το παράθυρο το οποίο μας εμφανίζεται, επιλέγουμε την εσωτερική μεταβλητή **Tank_Fill** και πατάμε το κουμπί **OK** για να αποθηκευτούν οι αλλαγές μας (εικόνα 6.2.20). Ορίζουμε ως **κύκλο ανανέωσης (Update Circle)** την επιλογή **Upon Change**, για να εμφανίζονται σε πραγματικό χρόνο οι μεταβολές της μεταβλητής. Ακολουθούμε τα ίδια βήματα και για τα αντικείμενα **Control** και **I/O Field**, όπως αυτά εμφανίζονται στις εικόνες 6.2.21 και 6.2.22 αντίστοιχα.



Εικόνα 6.2.20

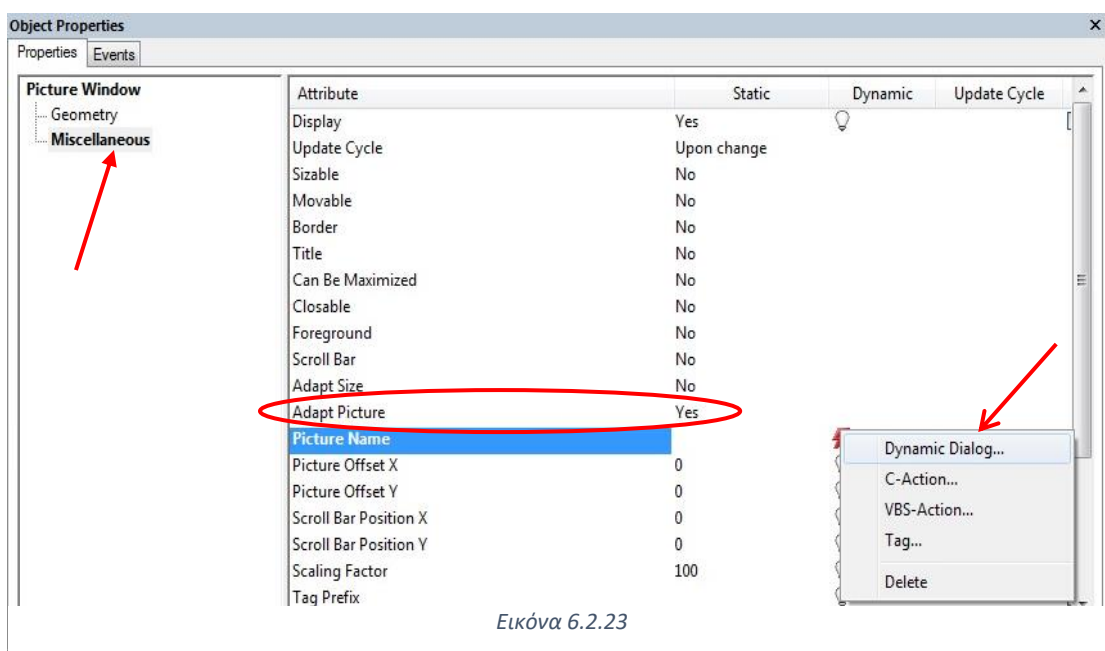


Εικόνα 6.2.21



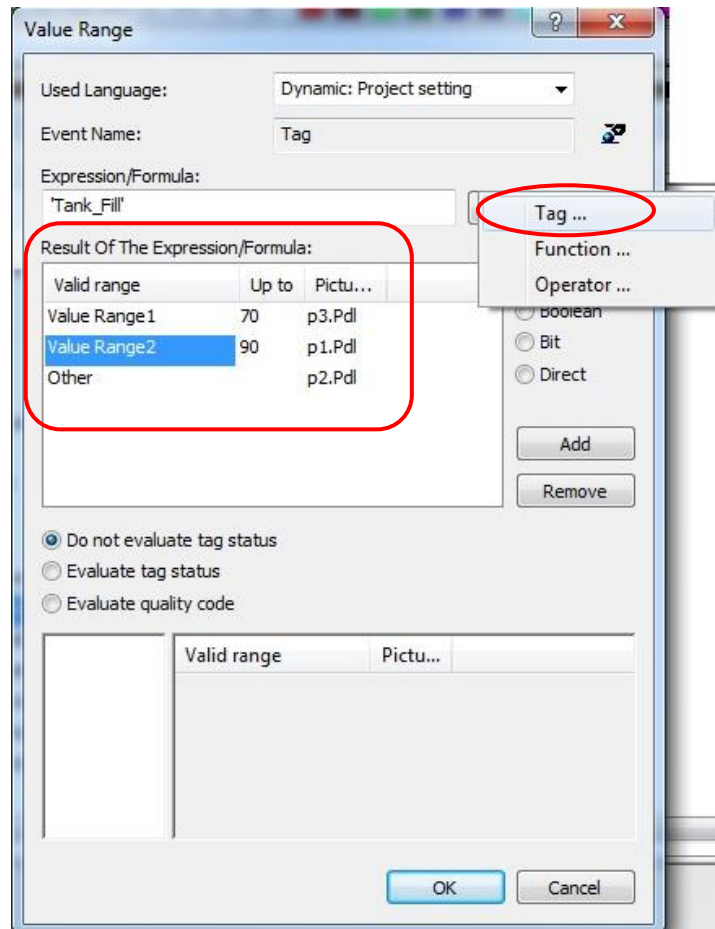
Εικόνα 6.2.22

Στο αντικείμενο **Picture Window**, για να το συνδέσουμε με την εσωτερική μας μεταβλητή, μπαίνουμε στις ιδιότητές του κάνοντας πάνω του **δεξί κλικ** και επιλέγουμε το μενού **Miscellaneous** της καρτέλας **Properties**. Εκεί, ενεργοποιούμε το πεδίο **Adapt Picture** ενώ στην στήλη **Dynamic** του πεδίου **Picture Name** κάνουμε **δεξί κλικ** και επιλέγουμε το **Dynamic Dialog** (εικόνα 6.2.23). Στο νέο παράθυρο που εμφανίζεται, στο πεδίο **Expression/Formula**, κάνουμε κλικ στο κουμπί όπου βρίσκεται δίπλα του κι επιλέγουμε το μενού **Tags**, για να συνδέσουμε την μεταβλητή μας.



Εικόνα 6.2.23

Έπειτα, στο πεδίο **Results of The Expression/Formula**, επιλέγουμε σαν **Data Type** την κατηγορία **Analog** και με το κουμπί **Add** προσθέτουμε άλλες **2 όρια αλλαγής** για τα εικονίδια μας, ένα **πάνω από τα 70** και ένα **πάνω από τα 90**, αντιστοιχώντας και τις ανάλογες εικόνες, όπως διακρίνεται στην **εικόνα 6.2.24**. Κάνουμε κλικ στο κουμπί **OK** για να αποθηκευτούν οι επιλογές μας.



Εικόνα 6.2.24

Τελευταίο μας κομμάτι είναι ο προγραμματισμός των κουμπιών **Start** και **Stop**. Για την περίπτωση αυτή, θα χρησιμοποιήσουμε πάλι το υποπρόγραμμα **Global Script**, όπως έγινε και προηγουμένως στην ρύθμιση των κουμπιών στην εικόνα **Intro**. Για το κουμπί **Start**, μπαίνουμε στις ιδιότητές του και στην καρτέλα **Event** κάνουμε κλικ στο **Mouse** που βρίσκεται στο μενού **Button1** της κατηγορίας **Objects**. Εν συνεχεία, στην επιλογή **Mouse Click**, κάνουμε **δεξί κλικ** και επιλέγουμε την λειτουργία **VBS – Editor**. Ανοίγοντας το πρόγραμμα **Global Script**, εισάγουμε τον κώδικα όπως παρουσιάζεται παρακάτω


```
Sub OnClick (Byval Item)

Dim I, ms, Tag1, obj1, obj2, obj3

Set Tag1 = HMIRuntime.Tags("Tank_Fill")
Set obj1 = ScreenItems ("Circle1")
Set obj2 = ScreenItems ("Circle2")
Set obj3 = ScreenItems ("Button2")

obj2.BackColor = RGB (255,255,255)
obj1.BackColor = RGB (0,255,0)

Tag1.Write (0)

For I = 0 To 100 Step 0.01

    Tag1.Write (I)
    If obj3.Pressed Then

        Exit For
    End If

    If Tag1.Read = 70 Then

        ms = MsgBox ("Warning!!! The tank is almost full", vbExclamation +
vbOKOnly, "Warning")

        If ms = 1 Then

            Do

                Tag1.Write (I)
                I = I + 0.01

            Loop Until Tag1.Read = 100 Or obj3.Pressed

        End If

    End If

Next

End Sub
```

Για τον προγραμματισμό του κουμπιού **Stop (Button2)**, ακολουθούμε τα προαναφερόμενα βήματα και εισάγουμε τον παρακάτω κώδικα


```

Sub OnClick (Byval Item)
Dim obj3, obj4, ms, tag2
Set obj3 = ScreenItems("Circle1")
Set obj4 = ScreenItems("Circle2")
Set tag2 = HMIRuntime.Tags("Tank_Fill")

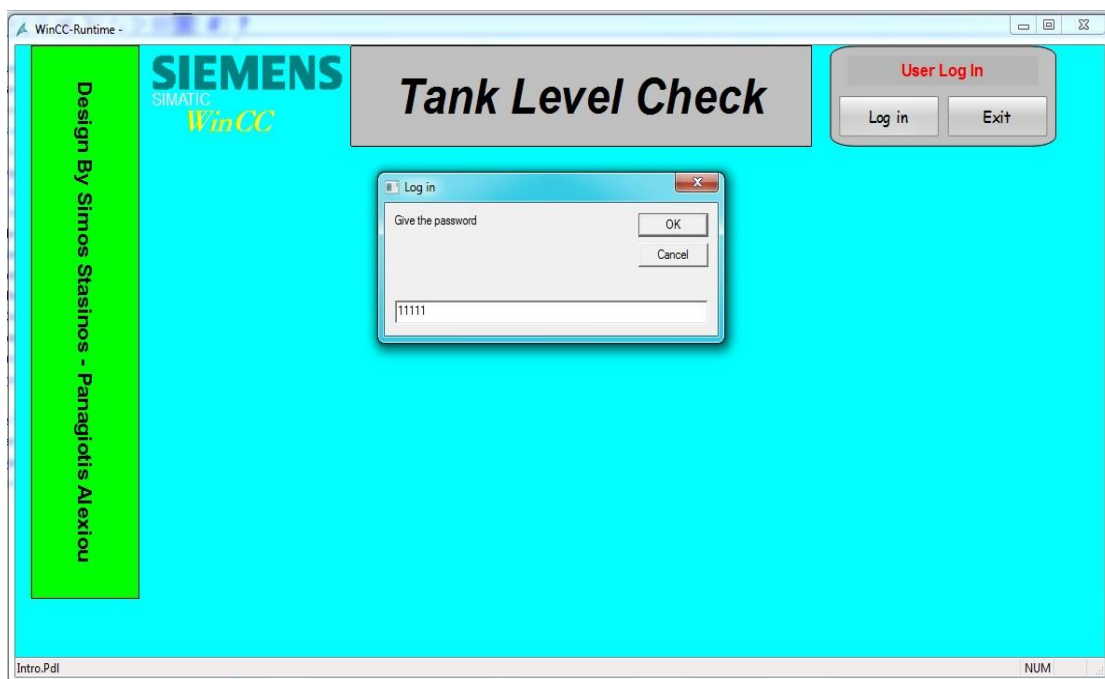
obj3.BackColor = RGB (255,255,255)
obj4.BackColor = RGB (255,0,0)

Set sm = MsgBox ("The tank is filled with " &tag2.Read& " Litres",vbOKOnly +
vbInformation,"Complete filling")

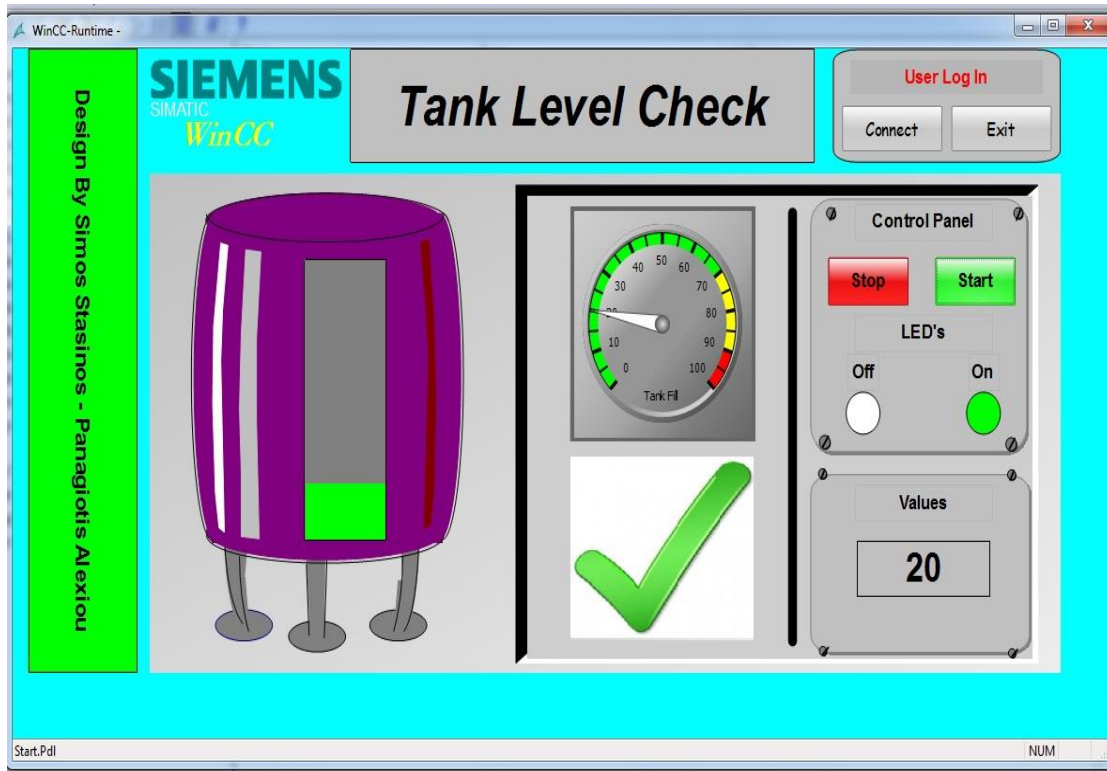
End Sub

```

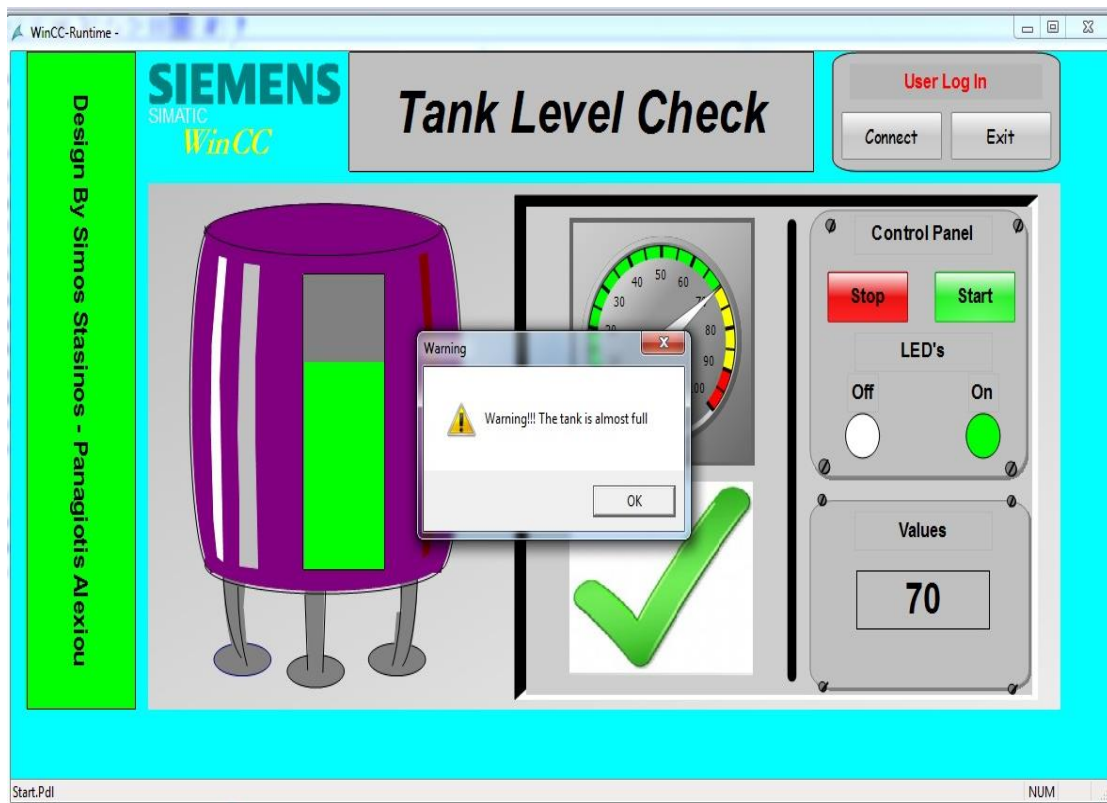
Έχοντας πλέον ορίσει και ρυθμίσει τις λειτουργίες για όλα τα αντικείμενα που προσθέσαμε στην εφαρμογή μας, την αποθηκεύουμε και προχωρούμε στην εκκίνησή του περιβάλλοντος εργασίας. Κάνοντας λοιπόν κλικ στο κουμπί **Play** που βρίσκεται στην μπάρα **Toolbar** του λογισμικού μας, εκκινεί το υποπρόγραμμα **WinCC Runtime** και μας εμφανίζεται σε παράθυρο το παρακάτω περιβάλλον εργασίας (εικόνες 6.2.25 – 6.2.28)



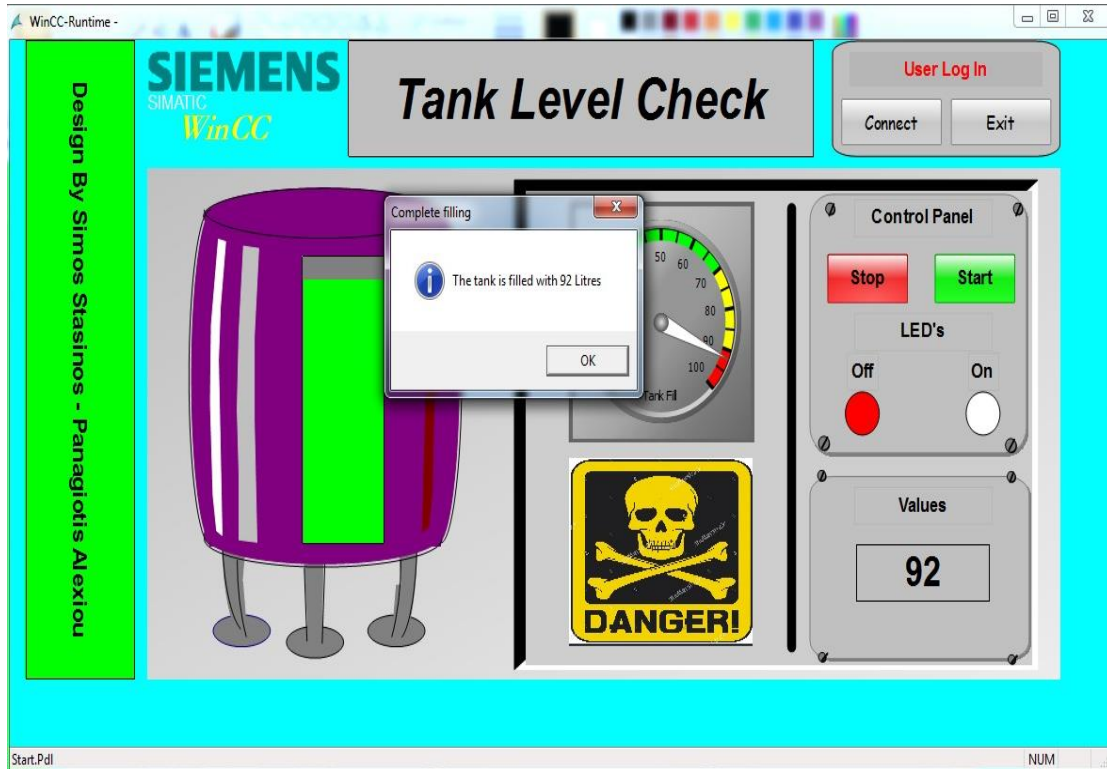
Εικόνα 6.2.25



Εικόνα 6.2.26



Εικόνα 6.2.27



Εικόνα 6.2.28

Βιβλιογραφία

1. Α. Βελώνη – Σ. Αλατσαθανιός, «**Βιομηχανική Πληροφορική**», Αθήνα 2014
2. Jose Angel Gomez Gomez, «**Survey of SCADA SYSTEMS and visualization of a real life process**», Master Thesis of Swedish Department of Electrical Engineering 2002
3. Ροβέρτος Ε. Κινγκ, «**Βιομηχανική Πληροφορική**», Εκδόσεις Τζιόλα
4. Παπαδόπουλος Θ. Βασίλειος, «**Μελέτη συστήματος παροχής ηλεκτρικών τάσεων σε εργαστηριακό χώρο – παρεμβάσεις σε σύστημα ελέγχου PLC μέσω του προγράμματος SCADA**», Διπλωματική εργασία Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Πανεπιστήμιο Πατρών 2008
5. Νικόλαος Σαρρής, «**Εποπτικός έλεγχος και συλλογή δεδομένων SCADA**», Διπλωματική εργασία τμήματος Ηλεκτρολόγων Μηχανικών, Πολυτεχνική σχολή Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης 2002
6. Χατζηϊωάννου Ιωάννης - Γιαμαρέλος Γεώργιος, «**Έλεγχος και σταθεροποίηση θερμοκρασίας με χρήση συστήματος τηλεποπτοίας και τηλεχειρισμού**», Πτυχιακή εργασία τμήματος Αυτομάτισμού, ΤΕΙ Πειραιά
7. Σιγάλας Αρτέμιος - Σκαρπέτης Δημήτριος , «**SIMATIC HMI WINCC v7.2**», Πτυχιακή εργασία τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστικών Συστημάτων, ΤΕΙ Πειραιά
8. Γεώργιος Σ. Σκιαδόπουλος, «**Εργαστηριακές εφαρμογές SCADA**», Διπλωματική εργασία σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Ηλεκτρικής Ισχύος, Αθήνα, Σεπτέμβριος 2007
9. Κοσμάρας Αστέριος – Τιάκας Γεώργιος – Παππής Ιωάννης, «**Η Ελληνική Βιομηχανία**», παρουσίαση του μαθήματος Σχεδιασμός & Προγραμματισμός Παραγωγής, Τμήμα Μηχανολόγων Μηχανικών, Πολυτεχνική Σχολή
10. Αναστάσιος-Διονύσιος Καραγιάννης, «**Ασφάλεια πληροφοριών στα Ευφυή Δίκτυα HE**», Διπλωματική εργασία, Τομέας Συστημάτων Μετάδοσης Πληροφορίας και

- Τεχνολογίας Υλικών, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών, Αθήνα, Ιούνιος 2015
11. Ανδρεαδάκης Γεώργιος, «**Ανάπτυξη Εφαρμογής εικονικού Κόμβου Ασύρματης τηλεμετρίας**», Πτυχιακή εργασία τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων, ΤΕΙ Κρήτης
 12. Office of the Manager Of National Communications Systems, «**Supervisory Control and Data Acquisition (SCADA) Systems**», Technical Information Bulletin 04-1, Virginia, October 2004
 13. Rao Kalapatapu, «**SCADA Protocols and Communication Trends**», The Instrumentation, Systems and Automation Society, October 2004
 14. Debnath Bhattacharyya, «**The Taxonomy of Advanced SCADA Communication Protocols**», Journal of Security Engineering, India, December 2008
 15. <http://en.wikipedia.org/wiki/SCADA>
 16. http://en.wikipedia.org/wiki/SCADA#SCADA_In_the_workplace
 17. <https://el.wikipedia.org/wiki/%CE%92%CE%B9%CE%BF%CE%BC%CE%B7%CF%87%CE%B1%CE%BD%CE%AF%CE%B1>
 18. <http://www.dpstele.com/scada/introduction-fundamentals-implementation.php>
 19. <http://ebooks.edu.gr/modules/ebook/show.php/DSGL106/282/2015,6863/>
 20. <http://ebooks.edu.gr/modules/ebook/show.php/DSGL-C124/54/416,1542/>
 21. <http://users.sch.gr/giannouleask/notes%20techn%20b.htm>
 22. <http://electrical-engineering-portal.com/three-generations-of-scada-system-architectures>