

**ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Εργαστηριακές Εφαρμογές SCADA**

**Στέφανος Κ. Θεοχάρης**

**Εισηγήτρια: Αναστασία Βελώνη**

**ΑΘΗΝΑ  
ΔΕΚΕΜΒΡΙΟΣ 2015**



**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Εργαστηριακές Εφαρμογές SCADA**

**Στέφανος Κ. Θεοχάρης  
Α.Μ. 35669**

**Εισηγήτρια:**

**Αναστασία Βελώνη**

**Εξεταστική Επιτροπή:**

**Ημερομηνία εξέτασης**



## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό των εργαστηριακών εφαρμογών στα συστήματα SCADA. Την προσπάθειά μου αυτή υποστήριξε η επιβλέπουσα καθηγήτρια μου, την οποία θα ήθελα να ευχαριστήσω.



## ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας εργασίας ήταν η υλοποίηση σεναρίων βασικών εφαρμογών αυτοματισμού με το σύστημα SCADA της εταιρείας SIEMENS. Ασχοληθήκαμε με την διασύνδεση και ανάπτυξη κατάλληλων περιβαλλόντων παρακολούθησης και ελέγχου, με τη χρήση του λογισμικού WinCC της εταιρείας SIEMENS.

Ξεκινώντας από την θεωρητική βάση πάνω στα συστήματα SCADA στο κεφάλαιο 1, συνεχίζουμε με ανάλυση και περιγραφή του πακέτου WinCC, το οποίο χρησιμοποιήσαμε, στο κεφάλαιο 2. Στη συνέχεια, στο κεφάλαιο 3, δημιουργούμε δοκιμαστικές εφαρμογές για περαιτέρω εμπέδωση των στοιχείων του WinCC που παρουσιάσαμε προηγουμένως

## Περιεχόμενα

### Κεφάλαιο 1- Συστήματα SCADA

1.1	Εισαγωγή στα συστήματα SCADA.....	1
1.2	Αρχή λειτουργίας.....	4
1.2.1	Δομικά στοιχεία.....	5
1.2.2	Συσκευές δεδομένων.....	7
1.3	Σύγχρονα καταναμημένα συστήματα SCADA.....	9
1.4	Αρχιτεκτονική SCADA.....	12
1.4.1	Μονολιθικά Συστήματα SCADA.....	12
1.4.2	Διανεμημένα Συστήματα SCADA.....	14
1.4.3	Δικτυωμένα Συστήματα SCADA.....	16
1.5	Wide Area Network (WAN).....	19
1.6	Πρωτόκολλα επικοινωνιών SCADA.....	20
1.6.1	Πρωτόκολλο EIA-232.....	23
1.6.2	Πρωτόκολλο EIA-422/485.....	24
1.6.3	Πρωτόκολλο Modbus.....	25
1.6.4	Πρωτόκολλο HART.....	28
1.6.5	Πρωτόκολλο Profibus PA/DP/FMS.....	29
1.7	Οφέλη των Συστημάτων SCADA.....	31
1.8	Το μέλλον των συστημάτων SCADA.....	33
1.9	Εφαρμογές των SCADA.....	35

### Κεφάλαιο 2- Το περιβάλλον SCADA WinCC της SIEMENS

2.1	Εισαγωγή στο HMI(human machine interface).....	37
2.2	Το περιβάλλον WinCC.....	39
2.3	Βασική δομή WinCC.....	40
2.4	Computer ( Υπολογιστής ).....	41
2.5	Tag Management ( Διαχείριση Μεταβλητών ).....	42
2.6	Data Types ( Τύποι Μεταβλητών ).....	43
2.7	Editor.....	44
	Graphics Designer.....	44
	Tag Logging.....	44
	Alarm Logging.....	44
	Report Designer.....	45



Global Script .....	45
2.8 Ο WinCC Explorer .....	46
2.8.1 Menu bar .....	46
2.8.2 Toolbar .....	47
2.8.3 Navigation window .....	48
2.8.4 Tooltip .....	49
2.8.5 Data Window .....	49
2.8.6 Status bar .....	50
2.9 Δημιουργία Project .....	51
2.10 Καθορισμός ιδιοτήτων Υπολογιστή .....	53
2.11 Καθορισμός Tag Management .....	55
2.12 Επιλογή και εγκατάσταση ενός driver .....	56
2.13 Ορισμός των μεταβλητών (tags) .....	57
2.13.1 External tags .....	57
2.13.2 Internal tags .....	61
2.14 Graphics Designer .....	63
2.14.1 Menu Bar .....	64
2.14.2 Toolbar .....	65
2.14.3 Font Palette .....	65
2.14.4 Color Palette .....	66
2.14.5 Workspace .....	66
2.14.6 Object Palette .....	67
2.14.7 Alignment Palette .....	67
2.14.8 Zoom Palette .....	68
2.14.9 Style Palette .....	68
2.14.10 Status Bar .....	69
2.15 Global Scripts .....	70
2.15.1 VBS .....	70
2.15.2 ANSI-C .....	74

### Κεφάλαιο 3- Ανάπτυξη Εφαρμογών με το WinCC

3.1 Σχεδιασμός δυναμικών σελίδων .....	77
3.1.1 Δημιουργία του project .....	77
3.1.2 Δημιουργία αρχικού μενού πλοήγησης .....	80
3.1.3 Δημιουργία buttons-διαχείριση Event .....	82

3.2	Σχεδιασμός δεξαμενής.....	86
3.2.1	Δημιουργία νέου project .....	86
3.2.2	Εγκατάσταση driver.....	86
3.2.3	Δημιουργία μεταβλητών .....	87
3.2.4	Σχεδιασμός γραφημάτων .....	89
3.2.5	Δημιουργία και ενσωμάτωση των scripts στο γράφημα.....	90
3.2.6	Πρόσθεση δυναμικών χαρακτηριστικών στη δεξαμενή .....	94
3.2.7	Εκκίνηση της εφαρμογής.....	96
	ΒΙΒΛΙΟΓΡΑΦΙΑ .....	98

# Κεφάλαιο 1

Συστήματα SCADA

## 1.1 Εισαγωγή στα συστήματα SCADA

Ξεκινώντας την περιγραφή των συστημάτων SCADA είναι σκόπιμο να αναφερθεί τι είναι το SCADA. Η λέξη SCADA αποτελεί τα αρχικά των λέξεων Supervisory Control And Data Acquisition System, δηλαδή σύστημα εποπτείας, ελέγχου και συλλογής πληροφοριών. Είναι συνεπώς συστήματα τηλεμετρίας και τηλεχειρισμού, τα οποία συλλέγουν πληροφορίες από διάφορες διεργασίες και χρησιμοποιούνται για τον εποπτικό τους έλεγχο.

Αρχικά υλοποιήθηκαν στο λειτουργικό σύστημα DOS( VMS ) και το Unix αλλά τα τελευταία χρόνια όλοι οι προμηθευτές SCADA έχουν κινηθεί προς τα NT και μερικά επίσης προς το λειτουργικό Linux. Τα SCADA βρίσκουν τεράστιες εφαρμογές, τόσο σε βιομηχανικές μονάδες όσο και σε συστήματα μεταφορές και διανομής ηλεκτρικής ενέργειας. Αυτό γιατί επιτρέπουν την διαχείριση και την εποπτεία ενός συστήματος που μπορεί να βρίσκεται αρκετές εκατοντάδες χιλιόμετρα μακριά από τον χώρο ελέγχου, όπως συνήθως συμβαίνει με τα συστήματα ηλεκτρικής ενέργειας, αλλά και τις μεγάλες βιομηχανικές μονάδες. Το μέγεθος τέτοιας σειράς εγκαταστάσεων εκτείνεται από 1000 μέχρι 10 χιλιάδες κανάλια εισόδου-εξόδου (I/O). Παρά όλα αυτά δεν συναντάμε συστήματα SCADA σε μικρομεσαίες βιομηχανικές μονάδες, λόγω του μεγάλου κόστους της υλικοτεχνικής υποδομής (hardware) αλλά και του λογισμικού (software). Ένα σύστημα SCADA είναι υπεύθυνο για την διαχείριση και τον έλεγχο διαφόρων διεργασιών, δηλαδή είναι υπεύθυνο για την παρακολούθηση, την καταγραφή και τον έλεγχο ενός πλήθους βασικών μεταβλητών και παραμέτρων του συστήματος.

Αυτά τα συστήματα καλύπτουν τη μεταφορά των δεδομένων μεταξύ ενός κεντρικού οικοδεσπότη υπολογιστή SCADA(central host PC) και διάφορες μονάδες απομακρυσμένων τερματικών (RTUs) και προγραμματιζόμενων λογικών ελεγκτών (PLCs), και τα τερματικά χειριστών. Παραδοσιακά, τα συστήματα SCADA έχουν χρησιμοποιήσει στο δημόσιο δίκτυο μεταβίβασης (PSN) για λόγους ελέγχου.

Σήμερα πολλά συστήματα ελέγχονται χρησιμοποιώντας την υποδομή του εταιρικού δικτύου Local Area Network (LAN)/Wide Area Network (WAN).

Οι ασύρματες τεχνολογίες τώρα πλέον ευρέως επεκτείνονται και χρησιμοποιούνται για λόγους του ελέγχου. Οι στόχοι του SCADA αναφέρονται ευθύς αμέσως :

- Η διασφάλιση της ποιότητας του παραγόμενου προϊόντος.
- Η μεγιστοποίηση της παραγωγής με χρήση των ελάχιστων δυνατών (ενεργειακών) πόρων.
- Η βέλτιστη διαχείριση του εξοπλισμού, των υλικών και της ενέργειας της εγκατάστασης.
- Η ασφάλεια του εξοπλισμού και του προσωπικού παρακολούθησης της διεργασίας.

Είναι σημαντικό όλοι οι παραπάνω στόχοι να επιτυγχάνονται παράλληλα. Δεν είναι δυνατό να επιδιώκουμε μεγιστοποίηση της παραγωγής χωρίς πρώτα να έχουμε εξασφαλίσει την βελτιστοποίηση διαχείρισης 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. Έκτος των παραπάνω, ένα σύστημα SCADA μπορεί να επιτηρεί και να χειρίζεται ένα πλήθος μεταβλητών του συστήματος αυτομάτου ελέγχου, αλλά και να διαχειρίζεται και οικονομικά μεγέθη (παραγγελίες - παραδόσεις προϊόντων) σε συνεργασία με οικονομικά πακέτα, προκειμένου να παρέχει στον χειρίστη του συνολική εποπτεία της παραγωγικής μονάδας.

Πιο συγκεκριμένα, ένα σύστημα SCADA προσφέρει :

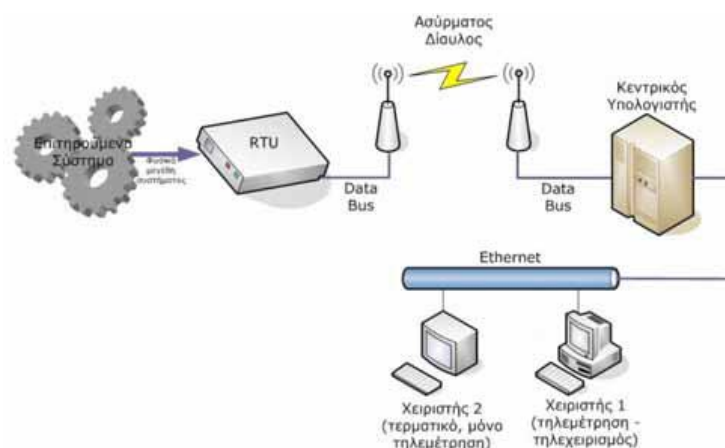
- άμεση πληροφόρηση της κατάστασης της διεργασίας
- αντιστάθμιση των μεταβλητών ελέγχου της διεργασίας με στόχο τη διατήρηση των δεδομένων ονομαστικών τους τιμών (setpoints) καθώς και τη διατήρηση των απαιτούμενων επιπέδων παραγωγής
- έγκαιρη σήμανση των βλαβών και της κακής λειτουργίας του εξοπλισμού στις διάφορες διεργασίες, ώστε να παρέχεται η μέγιστη ασφάλεια του εξοπλισμού και των εργαζομένων
- πρόγνωση και διάγνωση των βλαβών του εξοπλισμού και έγκαιρο εντοπισμό τους για την μεγιστοποίηση της διαθεσιμότητας του
- καταγραφή και αποθήκευση πληροφοριών σχετικά με την παραγωγή και τη διαχείριση της
- καλή λειτουργία του εξοπλισμού με στόχο τη βελτιστοποίηση της χρήσης και επομένως της παραγωγικότητας του.

## 1.2 Αρχή λειτουργίας

Όπως έχουμε ήδη αναφέρει ένα σύστημα SCADA αποτελεί ένα πλήρες σύστημα τηλεμετρίας και τηλεχειρισμού. Έτσι χρησιμοποιείται σε περιπτώσεις όπου το προς διαχείριση σύστημα απέχει αρκετά από το χώρο διαχείρισης. Προκειμένου λοιπόν να υλοποιήσουμε ένα SCADA απαιτούνται, εκτός των άλλων, ένα σύστημα τηλεμετρίας. Το σύστημα αυτό υλοποιείται με τη βοήθεια σταθμών RTU (Remote Telemetry Units), οι οποίοι είναι συνδεδεμένοι με τη παραγωγική διαδικασία, «διαβάζουν» τις τιμές διαφόρων φυσικών μεγεθών που μας ενδιαφέρουν, π.χ. πίεση, θερμοκρασία, συχνότητα, τα μετατρέπουν σε ηλεκτρικά σήματα και τα μεταδίδουν μέσω ενός ενσύρματου ή ασύρματου διαύλου (ανάλογα με τις ανάγκες της εφαρμογής) στον υπολογιστή που φέρει το λογισμικό SCADA, σε τακτά χρονικά διαστήματα.

Το χρονικό διάστημα που μεσολαβεί ανάμεσα στις διαδοχικές μεταδόσεις εξαρτάται αφενός από την ταχύτητα εξέλιξης της επιτηρούμενης διεργασίας και αφετέρου από την ακρίβεια που επιθυμούμε για το σύστημα μας. Εκτός των RTU, για την υλοποίηση του SCADA, απαιτούνται και ένας κεντρικός υπολογιστής, αρκετά μεγάλης υπολογιστικής ισχύος, που θα φέρει το λογισμικό SCADA και στον οποίο θα καταλήγουν οι μετρήσεις από όλους τους σταθμούς RTU καθώς και οι απαιτούμενες τηλεπικοινωνιακές ζεύξεις ανάμεσα στους σταθμούς RTU και τον κεντρικό υπολογιστή.

Στο σχήμα 1.1 φαίνεται μια συνηθισμένη τοπολογία ενός συστήματος SCADA.

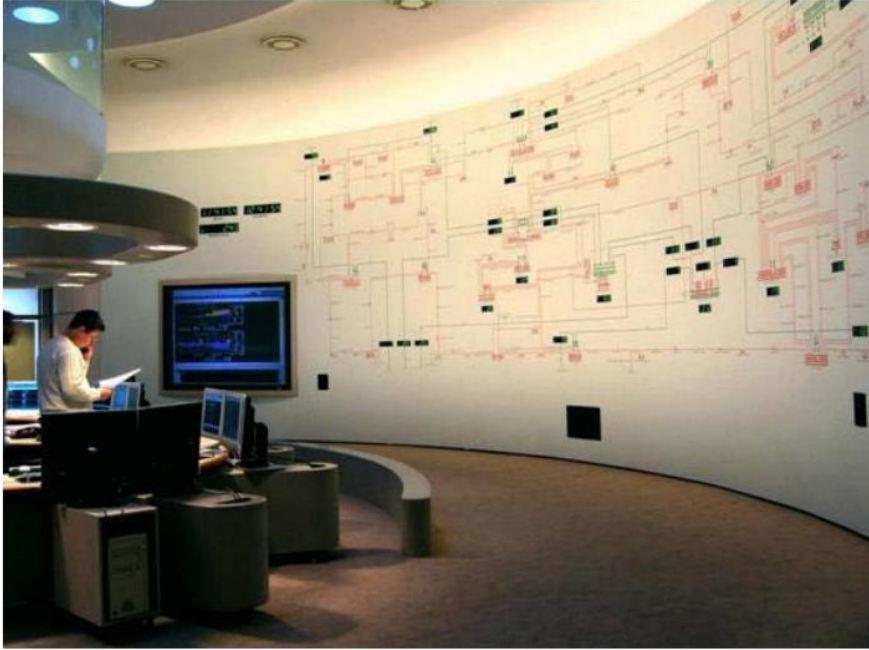


Σχήμα 1.1: Τοπολογία ενός συστήματος SCADA

### 1.2.1 Δομικά στοιχεία

Τα δομικά στοιχεία ενός συστήματος SCADA είναι:

- Ένας κεντρικός υπολογιστικός σταθμός (Master Station Computer -MTU).



**Σχήμα 1.2:** Κύρια τερματική μονάδα – MTU

- RTU's που κωδικοποιούν και αποκωδικοποιούν σήματα από τον πραγματικό κόσμο.



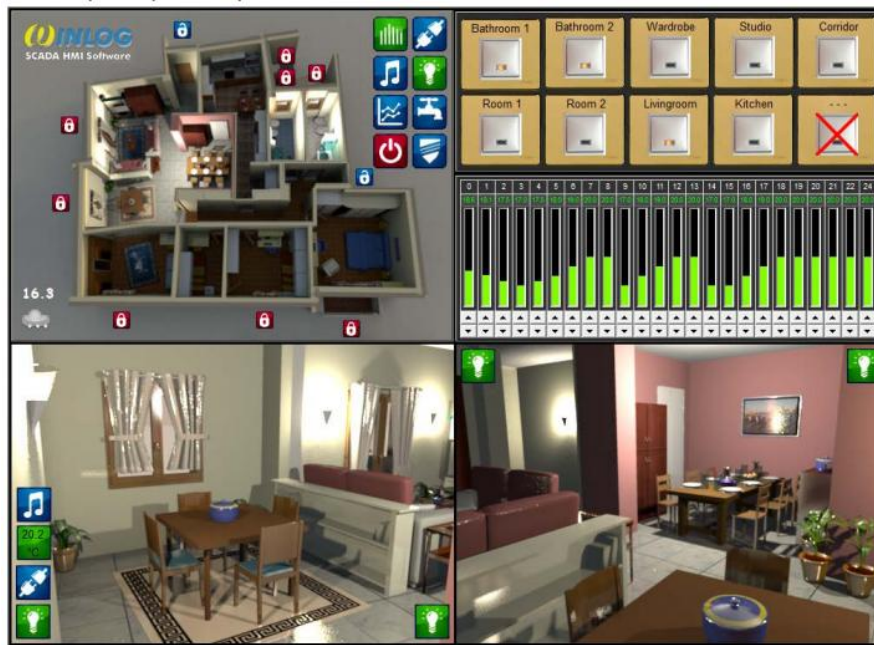
**Σχήμα 1.3:** Εσωτερική εγκατάσταση RTU



- Οι γραμμές επικοινωνίας (radio, καλωδιακή, τηλεφωνική).
- Το ελεγχόμενο σύστημα (Field Instrumentation).

Ένα σύστημα SCADA αποτελείται επίσης από τα ακόλουθα υποσυστήματα:

- Ένα σύστημα ανθρώπινης αλληλεπίδρασης (HMI-Human Machine Interface), που έχει σαν σκοπό να παρουσιάζει τα δεδομένα της γραμμής και ο χρήστης να μπορεί να τα ελέγχει καθ' όλη τη διάρκεια της παραγωγής.



**Σχήμα 1.4: Παράδειγμα HMI (Αυτοματισμός κτιρίων)**

- Έναν υπολογιστή ο οποίος συλλέγει όλες τις πληροφορίες και στέλνει την κατάλληλη ανατροφοδότηση.
- Τηλεχειριζόμενες τερματικές μονάδες, οι οποίες είναι συνδεδεμένες με αισθητήρες σε όλη τη διαδικασία, εναλλάσσοντας έτσι σήματα από τους αισθητήρες στο συντονιστικό υπολογιστή.
- Προγραμματιζόμενους λογικούς ελεγκτές (PLC's).
- Την επικοινωνιακή υποδομή του συστήματος, η οποία συνδέει όλα τα παραπάνω κατάλληλα μεταξύ τους.

## 1.2.2 Συσκευές δεδομένων

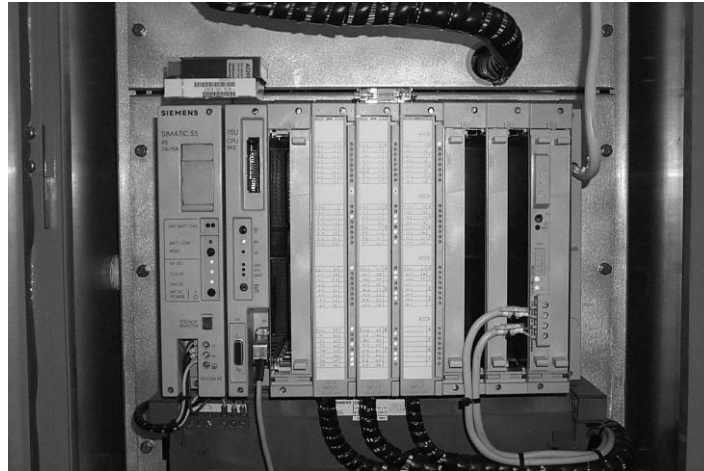
Οι συσκευές όπως οι μετρητές της στάθμης μιας δεξαμενής, οι μετρητές ροής του νερού, οι μετρητές θερμοκρασίας, οι μετρητές κατανάλωσης ενέργειας και οι μετρητές πίεσης όλοι μαζί παρέχουν πληροφορίες χρήσιμες σε έναν χειριστή ως προς το πόσο καλά λειτουργεί ένα σύστημα διανομής νερού. Επιπλέον, ο εξοπλισμός όπως οι ηλεκτρικοί μηχανισμοί κίνησης των βαλβίδων, οι πίνακες ελέγχου του μηχανισμού και οι ηλεκτρονικές εγκαταστάσεις μπορούν να χρησιμοποιηθούν για να διαμορφώσουν τα χέρια του συστήματος SCADA και να βοηθήσουν στην αυτοματοποίηση της διαδικασίας της διανομής του νερού.

Παρόλα αυτά, πριν από κάποια αυτοματοποιημένη ή απομακρυσμένη παρακολούθηση, οι πληροφορίες που περνούν από και προς τις συσκευές δεδομένων θα πρέπει να μετατραπούν σε μία μορφή που είναι συμβατή με την γλώσσα του συστήματος SCADA. Για να επιτευχθεί αυτό, απαιτείται κάποια μορφή ηλεκτρονικών δεδομένων.

Τα RTU (Remote Telemetry Units) βασικά χρησιμοποιούνται για να μετατρέπουν τα ηλεκτρονικά σήματα που έρχονται από τις συσκευές σε μία γλώσσα (γνωστή και ως πρωτόκολλο επικοινωνίας) που χρησιμοποιείται για να μετατρέψει τα δεδομένα σε ένα κανάλι επικοινωνίας. Τα RTU εμφανίζονται ως ένα κουτί σε ένα τηλεφωνικό πίνακα με καλώδια ηλεκτρονικού σήματος που τρέχουν μέσα στις συσκευές δεδομένων και καλώδια σύνδεσης στο κανάλι επικοινωνίας.

Οι οδηγίες για τον αυτοματισμό των συσκευών των δεδομένων συνήθως αποθηκεύονται τοπικά. Αυτό οφείλεται σε μεγάλο βαθμό στο συνηθισμένο περιορισμένο εύρος ζώνης των συνδέσμων των επικοινωνιών μεταξύ του κεντρικού υπολογιστή του SCADA και των συσκευών των δεδομένων. Τέτοιες οδηγίες συνήθως κρατούνται μέσα σε τοπικές ηλεκτρονικές συσκευές γνωστές ως Programmable Logic Controllers (PLC), οι οποίες έχουν κατά το παρελθόν διαχωριστεί από τα RTU. Τα PLC συνδέονται άμεσα με τις συσκευές δεδομένων και ενσωματώνουν προγραμματισμένη γνώση σε μορφή λογικών διαδικασιών.

Τα PLC συχνά χρησιμοποιούνται σε εργοστάσια με κατασκευαστικές εφαρμογές. Η ανάγκη ώστε τα PLC να συνδέονται με τα κανάλια επικοινωνίας δεν ήταν αρχικά μεγάλη σε αυτές τις εφαρμογές, καθώς αυτά συχνά απαιτούνταν να αντικαθιστούν τα λογικά συστήματα καθυστέρησης ή τους πνευματικούς ελεγκτές.



**Σχήμα 1.5:** PLC που διεξάγει λειτουργίες τοπικού ελέγχου, και που είναι διαχωρισμένο αλλά ενώνεται με καλώδια με το RTU

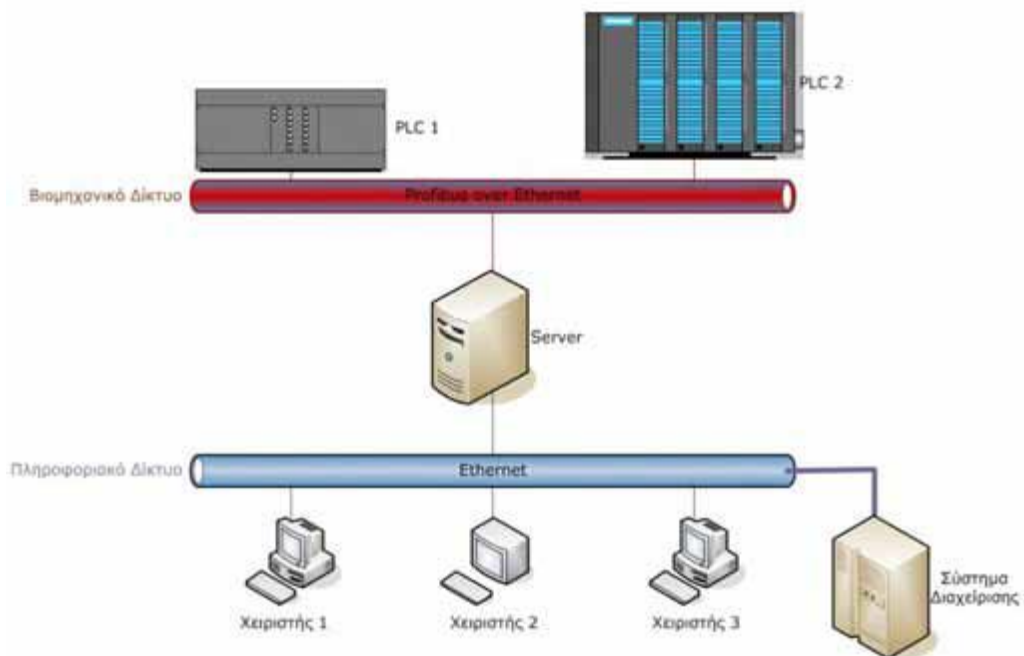
Με την πάροδο του χρόνου έγινε επιθυμητό να επηρεάσουν το πρόγραμμα μέσα στο PLC από ένα απομακρυσμένο σταθμό. Αυτό στη πραγματικότητα είναι το μέρος του εποπτικού ελέγχου (Supervisory Control) από το ακρωνύμιο του SCADA. Οι κατασκευαστές των PLC και RTU επομένως ανταγωνίζονται στην ίδια αγορά.

Σαν αποτέλεσμα αυτών των εξελίξεων, η γραμμή μεταξύ των PLC και RTU έχει εξαφανιστεί και η ορολογία είναι προφανώς εναλλακτική. Για χάρη της απλότητας, ο όρος RTU θα χρησιμοποιείται για να αναφέρει μία απομακρυσμένη συσκευή δεδομένων. Παρόλα αυτά, μία τέτοια συσκευή θα μπορούσε να περιλαμβάνει προγραμματισμό αυτοματισμού και θα μπορούσε να χαρακτηριστεί σαν ένα PLC.

### 1.3 Σύγχρονα καταναμημένα συστήματα SCADA

Στις μέρες μας, όπου οι ηλεκτρονικοί υπολογιστές έχουν κατακλύσει την αγορά, οι κατασκευαστές λογισμικού SCADA καλούνται να αναπτύξουν περιβάλλοντα SCADA τα οποία θα μπορούν να «τρέχουν» σε προσωπικούς ηλεκτρονικούς υπολογιστές (PC), θα είναι σχεδιασμένα με βάση την 32μπιτη αρχιτεκτονική των σύγχρονων λειτουργικών συστημάτων καθώς και θα υποστηρίζουν τα πιο διαδεδομένα πρωτόκολλα διασύνδεσης υπολογιστών (Ethernet, Token Ring) αλλά και PLC (Profibus, Industrial Ethernet, TCP/IP). Επίσης, από τα σύγχρονα 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 σύγχρονα συστήματα SCADA υλοποιούνται με βάση την καταναμημένη αρχιτεκτονική της οποίας η τοπολογία φαίνεται στο σχήμα 1.6.



Σχήμα 1.6: Τοπολογία ενός συστήματος SCADA

Όπως φαίνεται από το σχήμα 1.6, η κατανεμημένη αρχιτεκτονική αποτελείται από δύο ανεξάρτητα δίκτυα. Το πρώτο είναι το βιομηχανικό δίκτυο και περιέχει τα PLC. Στην κορυφή του δικτύου αυτού βρίσκεται ένας εξυπηρετητής (server), ο οποίος συνήθως είναι ένας σταθμός εργασίας (π.χ. Alpha, Vaxstation, RISC, Sun SPARK, IBM RS6000 κλπ.) με λειτουργικά σύστημα ικανότητας επεξεργασίας πολλαπλών έργων (multitasking) (όπως π.χ. UNIX, Sun Solaris, WINDOWS-NT κλπ.) ή ένας μικροϋπολογιστής μεγάλης υπολογιστικής ισχύος, όπως ο Pentium 4 3.2GHz με 1 GB RAM, επίσης με λειτουργικό ικανότητας διαχείρισης πολλαπλών έργων (όπως UNIX, WINDOWS-NT, Linux).

Ο εξυπηρετητής αφενός διαχειρίζεται το βιομηχανικό δίκτυο, δηλαδή ελέγχει και καθορίζει της εκπομπές πακέτων από και προς τα PLC, και αφετέρου φέρει δύο βάσεις δεδομένων. Η πρώτη είναι μια βάση δεδομένων πραγματικού χρόνου (Real Time Data Base, RTDB) στην οποία αποθηκεύονται όλες οι πληροφορίες που συλλέγονται από το PLC που βρίσκονται προσδεδεμένα στην επιτηρούμενη παραγωγική διαδικασία, η δεύτερη αποτελεί μια σχεσιακή βάση δεδομένων (Relational Data Base-RDB), στην οποία βρίσκονται αποθηκευμένες όλες οι απαραίτητες πληροφορίες για την λειτουργία του συστήματος. Ένας ακόμη ρόλος του server είναι να εξυπηρετεί τα διάφορα μικροϋπολογιστικά συστήματα που είναι συνδεδεμένα στο δεύτερο ανεξάρτητο δίκτυο του συστήματος SCADA. Πρόκειται για το πληροφοριακό δίκτυο (information network), υψηλής ταχύτητας που συνήθως είναι τύπου Thin ή Thich Ethernet (π.χ. Novell).

Η αρχιτεκτονική που αναφέραμε παραπάνω είναι η κλασσική "πελάτη-εξυπηρετητή" (client-server), που συνηθίζεται στα σύγχρονα κατανεμημένα πληροφορικά συστήματα. Κάθε μικροϋπολογιστής στο πληροφοριακό δίκτυο έχει το κατάλληλο λογισμικό εφαρμογής και εκτελεί τις δικές του εφαρμογές ανεξάρτητα των άλλων και ανάλογα με τις απαιτήσεις του συγκεκριμένου χρήστη.

Για παράδειγμα ένας χρήστης μπορεί να είναι ο χειριστής της διαδικασίας, η οποία έχει προσπέλαση σε ορισμένα παραγωγικά στοιχεία της υπό έλεγχο διαδικασίας (αφού χρησιμοποιεί απλό τερματικό), ενώ ένας δεύτερος χρήστης μπορεί να είναι ο μηχανικός παραγωγής που έχει προσπέλαση σε όλα τα τμήματα (σημεία) της παραγωγής, αφού χρησιμοποιεί υπολογιστικό σύστημα.

Έτσι κάθε χρήστης του συστήματος SCADA έχει το δικό του ηλεκτρονικό υπολογιστή, ή τερματικό, ανάλογα με τις ανάγκες, μέσω του οποίου μπορεί να ανακαλέσει πληροφορίες (information retrieval) που είναι αποθηκευμένες στον εξυπηρετητή, να τις επεξεργαστεί και να τις απεικονίσει κατάλληλα. Ο server συνεπώς έχει διπλό ρόλο, αφού αφενός είναι ο συνδετικός κρίκος των δύο ανεξάρτητων δικτύων και αφετέρου αποτελεί τον πυρήνα του συστήματος Συλλογής Πληροφοριών και Εποπτικού Ελέγχου.

Πρέπει να γίνει σαφές ότι ένας χρήστης μέσω του υπολογιστή του δεν μπορεί να επέμβει άμεσα στο βιομηχανικό δίκτυο με τα PLC. Αυτό συμβαίνει διότι ο τελικός χρήστης αυτό που βλέπει και χειρίζεται στην οθόνη του υπολογιστή του είναι μια εικόνα (image) του συστήματος αυτοματισμού, όπως αυτή βρίσκεται στην βάση δεδομένων του server. Η αρχιτεκτονική αυτή δεν επιβαρύνει το βιομηχανικό δίκτυο από τους χρήστες, αφού τα PLC προσπελούνται μόνο από το server και επιπλέον προσφέρει πρόσθετη ασφάλεια των πληροφοριών.

Τέλος, σημαντικό ρόλο στη λειτουργία του SCADA διαδραματίζει η επίδοση του δικτύου, η οποία προφανώς δεν είναι σταθερή σε όλα τα σημεία του δικτύου, αφού τα χρησιμοποιούμενα πρωτόκολλα του βιομηχανικού δικτύου διαφέρουν συνήθως από αυτά του πληροφοριακού.

## 1.4 Αρχιτεκτονική SCADA

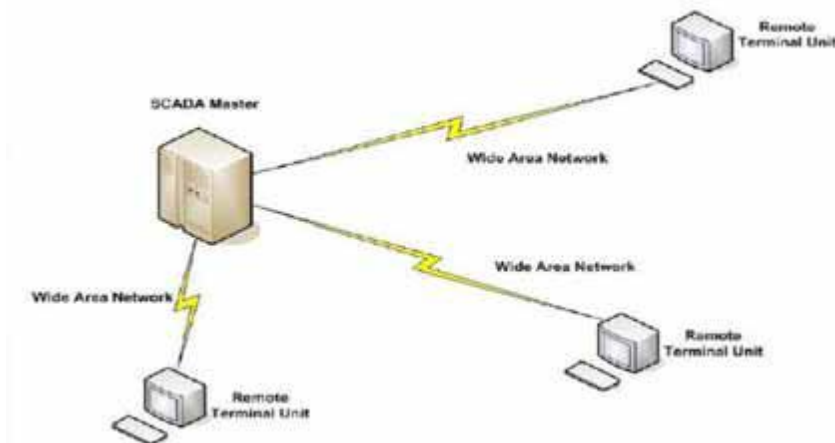
Τα συστήματα SCADA έχουν εξελιχθεί παράλληλα με την αύξηση και την βελτίωση της σύγχρονης τεχνολογίας υπολογιστών. Παρακάτω παρουσιάζεται μια περιγραφή των ακόλουθων τριών γενεών των συστημάτων SCADA:

- Πρώτη γενεά - μονολιθική(Monolithic)
- Δεύτερη γενεά – διανεμημένη(Distributed)
- Τρίτη γενεά - δικτυωμένη (Networked)

### 1.4.1 Μονολιθικά Συστήματα SCADA

Τα συστήματα SCADA ήταν αυτόνομα συστήματα με ουσιαστικά καμία συνδετικότητα σε άλλα συστήματα. Τα δίκτυα ευρείας περιοχής (WANs) που εφαρμόστηκαν για να επικοινωνήσουν με τις μονάδες απομακρυσμένων τερματικών (RTUs) σχεδιάστηκαν με έναν ενιαίο και μοναδικό σκοπό, αυτόν της επικοινωνίας με RTUs στον τομέα και την τίποτα άλλο.

Επιπλέον, τα πρωτόκολλα WAN, σε ευρεία λειτουργία σήμερα, ήταν κατά ένα μεγάλο μέρος άγνωστα τότε.



Σχήμα 1.7: Μονολιθικά συστήματα SCADA



Τα πρωτόκολλα επικοινωνίας σε λειτουργία σχετικά με τα δίκτυα SCADA αναπτύχθηκαν από τους προμηθευτές του εξοπλισμού RTU και ήταν συχνά ιδιόκτητα. Επιπλέον, αυτά τα πρωτόκολλα ήταν γενικά πολύ «αδύνατα», μη ενισχύοντας ουσιαστικά καμία λειτουργία πέρα από εκείνη την απαραίτητη, την ανίχνευση και τον έλεγχο των σημείων μέσα στη μακρινή συσκευή. Επίσης, δεν ήταν γενικά εφικτό να αναμιχθούν άλλοι τύποι κυκλοφορίας δεδομένων με τις ανακοινώσεις RTU σχετικά με το δίκτυο. Η συνδετικότητα στον κύριο σταθμό ήταν πολύ περιορισμένη από τον προμηθευτή συστημάτων.

Οι συνδέσεις στον κύριο υπολογιστή γινόταν με την χρήση ενός διαύλου (bus) η ενός προσαρμοστή δικτύου που συνδεόταν απευθείας με την ΚΜΕ (CPU) του υπολογιστή. Η λειτουργικότητα σε αυτά τα συστήματα πρώτης γενεάς 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.



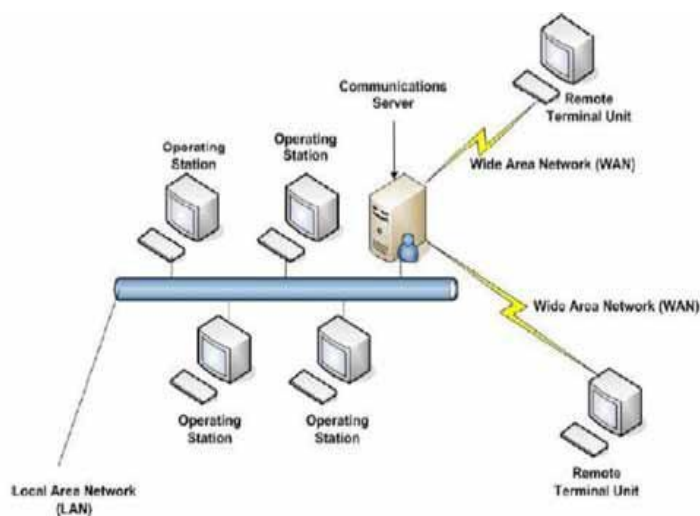
## 1.4.2 Διανεμημένα Συστήματα SCADA

Η επόμενη γενεά των συστημάτων SCADA εκμεταλλεύτηκε τις εξελίξεις και τη βελτίωση στη μικρογράφηση συστημάτων και την τοπική τεχνολογία δικτύωσης περιοχής (τοπικό LAN) για να διανείμει την επεξεργασία στα πολλαπλάσια συστήματα. Οι πολλαπλάσιοι σταθμοί, κάθε ένας με μια συγκεκριμένη λειτουργία, συνδέθηκαν με το τοπικό LAN και μοιράστηκαν τις πληροφορίες ο ένας με τον άλλον στον πραγματικό χρόνο.

Αυτοί οι σταθμοί είχαν χαρακτηριστικά της κατηγορίας μίνι υπολογιστών, μικρότεροι και λιγότερο ακριβοί από τους επεξεργαστές πρώτης γενεάς. Μερικοί από αυτούς τους διανεμημένους σταθμούς χρησιμοποιήθηκαν ως επεξεργαστές επικοινωνιών, με πρώτιστο έργο τους να μπορούν να επικοινωνούν με τις συσκευές τομέων όπως RTUs. Μερικοί χρησιμοποιήθηκαν ως χειριστές διασύνδεσης, παρέχοντας την διασύνδεση ανθρώπου - μηχανής (HMI) για τους χειριστές συστημάτων. Ακόμα, άλλοι που 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.



Σχήμα 1.8: Διανεμημένα συστήματα SCADA

Μερικά από τα πρωτόκολλα του τοπικού LAN που χρησιμοποιήθηκαν ήταν ιδιόκτητης φύσης, όπου ο προμηθευτής δημιούργησε το πρωτόκολλο δικτύων ή την έκδοσή του. Αυτό επέτρεψε σε έναν προμηθευτή να βελτιστοποιήσει το πρωτόκολλο του τοπικού LAN του για την πραγματικό χρόνο κυκλοφορία, αλλά περιόρισε τη σύνδεση του δικτύου από άλλους προμηθευτές στο τοπικό LAN SCADA.

Η διανομή της λειτουργίας συστημάτων στα συνδεδεμένα στο δίκτυο συστήματα χρησίμευσε όχι μόνο για την αύξηση της δύναμης επεξεργασίας, αλλά και για να βελτιώσει τον πλεονασμό και την αξιοπιστία του συστήματος συνολικά.

Παρά το απλό αρχικό/εφεδρικό σχέδιο που χρησιμοποιήθηκε σε πολλά συστήματα πρώτης γενεάς, η διανεμημένη αρχιτεκτονική κράτησε συχνά όλους τους σταθμούς στο τοπικό LAN σε απευθείας σύνδεση πραγματικού χρόνου. Το δίκτυο (LAN) που χρησιμοποιήθηκε για να επικοινωνήσει με τις συσκευές στον τομέα ήταν κατά ένα μεγάλο μέρος αμετάβλητο από την ανάπτυξη της συνδετικότητας του τοπικού LAN μεταξύ των τοπικών σταθμών στον κύριο SCADA.

Αυτά τα δίκτυα εξωτερικών επικοινωνιών περιορίστηκαν στα πρωτόκολλα RTU και δεν ήταν διαθέσιμα για άλλους τύπους κυκλοφορίας στα δίκτυα. Όπως συνέβη με την πρώτη γενεά των συστημάτων, η δεύτερη γενεά των συστημάτων SCADA περιορίστηκε επίσης στο υλικό, το λογισμικό, και τις περιφερειακές συσκευές που ήταν αναγκαίες.

### 1.4.3 Δικτυωμένα Συστήματα SCADA

Η χρήση της αρχιτεκτονικής κύριων σταθμών SCADA είναι στενά συνδεδεμένη με αυτόν της δεύτερης γενεάς, με την αρχική διαφορά ότι είναι μια αρχιτεκτονική ανοικτών συστημάτων παρά μια αρχιτεκτονική ενός προμηθευτής και ενός ελεγχόμενου, ιδιόκτητου περιβάλλοντος .

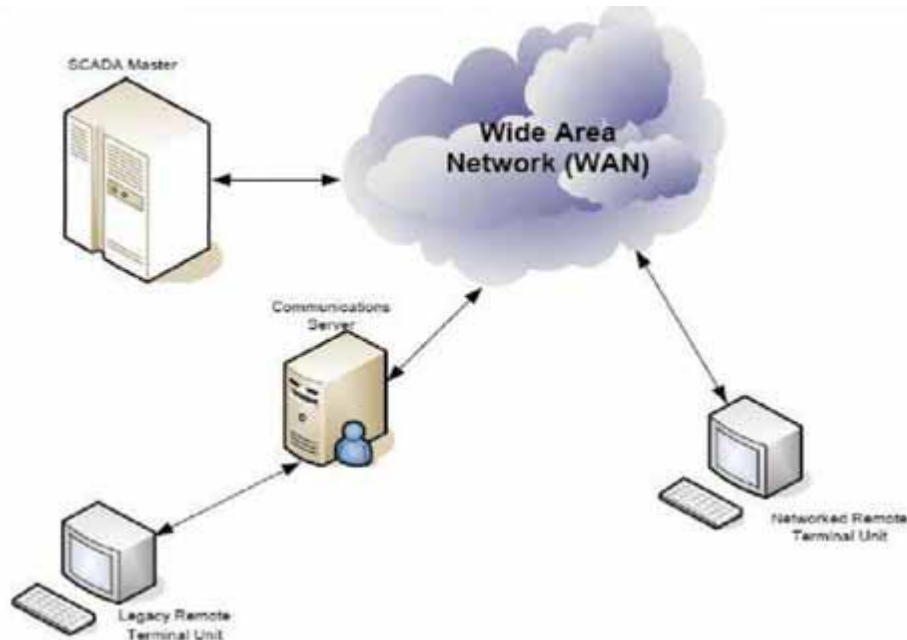
Υπάρχουν ακόμα πολλά δικτυωμένα συστήματα, που διαμοιράζουν τις λειτουργίες κύριων σταθμών. Υπάρχει ακόμα RTUs που χρησιμοποιούν πρωτόκολλα που είναι ιδιόκτητα. Η σημαντικότερη βελτίωση στην τρίτη γενεά είναι αυτή του ανοίγματος της αρχιτεκτονικής συστημάτων, της χρησιμοποίησης των ανοικτών προτύπων και των πρωτοκόλλων και την δυνατότητα να διανείμει τη λειτουργία SCADA σε ένα από έναν μεγάλο WAN και όχι μόνο σε ένα το τοπικό LAN.

Τα ανοικτά πρότυπα αποβάλλουν διάφορους περιορισμούς των Εισαγωγή στα συστήματα SCADA 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) με το σύστημα ή και το δίκτυο.

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 Compaq, η Hewlett Packard, και Microsoft για την πείρα τους στην ανάπτυξη των βασικών πλατφορμών 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 γενεάς SCADA προέρχεται από τη χρήση των WAN πρωτοκόλλων όπως το πρωτόκολλο Διαδικτύου (IP Internet) για την επικοινωνία μεταξύ του κύριου σταθμού και του εξοπλισμού επικοινωνιών.

Αυτό επιτρέπει το τμήμα του κύριου σταθμού που είναι αρμόδιος για τις επικοινωνίες με τις συσκευές τομέων που χωρίζονται από τον κύριο σταθμό να είναι ανεξάρτητος από το WAN. Οι προμηθευτές παράγουν τώρα RTUs που να μπορούν να επικοινωνήσουν με τον κύριο σταθμό χρησιμοποιώντας μια σύνδεση Ethernet.



Σχήμα 1.9: Δικτυωμένα συστήματα SCADA

Τα παρακάτω **μέσα επικοινωνίας** είναι τα πιο κοινά:

- Σύνδεσμοι ραδιοσυχνοτήτων (UHF και VHF).
- Δημόσια δίκτυα τηλεφώνου.
- Κινητή τηλεφωνία.
- Μικροκύματα.
- Δίκτυα κεραιών τηλεόρασης.
- Δορυφορικοί σύνδεσμοι.
- Εταιρικά υπολογιστικά συστήματα επικοινωνιών (WAN).

Σε μερικές περιπτώσεις συνηθίζεται και ο συνδυασμός αυτών των διαφορετικών μέσων που χρησιμοποιούνται για την διασφάλιση της υψηλής αξιοπιστίας των επικοινωνιών σε μία εγκατάσταση. Η **επιλογή των μέσων επικοινωνίας** εξαρτάται από μερικούς σημαντικούς παράγοντες:

- Την απόσταση του εξοπλισμού της εγκατάστασης.
- Την απαιτούμενη αξιοπιστία των μέσων επικοινωνίας.
- Τη διαθεσιμότητα των επιλογών επικοινωνίας.
- Το κόστος της κάθε επιλογής για μία συγκεκριμένη εφαρμογή.
- Τη διαθεσιμότητα της ενέργειας.

Τα συστήματα επικοινωνιών που χρησιμοποιούνται στο SCADA συνήθως διαχωρίζονται σε δύο μέρη: σε **δίκτυο μίας ευρείας περιοχής (WAN)** και σε **πολυάριθμα δίκτυα μίας περιορισμένης περιοχής (LAN's)**. Η ενοποίηση μεταξύ των δύο μερών επιτυγχάνεται μέσω κάποιας μορφής multiplexing.

Ένα άλλο πλεονέκτημα που επέρχεται από τη διανομή της λειτουργίας SCADA πέρα από έναν WAN είναι αυτό της ικανότητας επιβίωσης σε περίπτωση καταστροφής.

Η διανομή της επεξεργασίας SCADA πέρα από το τοπικό LAN στα δεύτερης γενεάς συστήματα βελτιώνει την αξιοπιστία, αλλά σε περίπτωση συνολικής απώλειας της δυνατότητας του κύριου σταθμού SCADA, ολόκληρο το σύστημα θα μπορούσε να χαθεί επίσης.

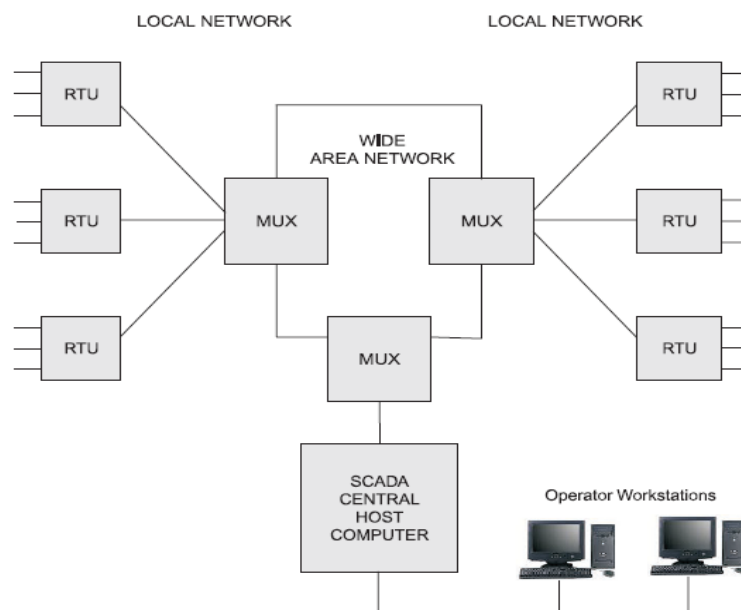
Με τη διανομή της επεξεργασίας στις φυσικά χωριστές θέσεις, είναι δυνατό να χτιστεί ένα σύστημα SCADA που μπορεί να επιζήσει μιας συνολικής απώλειας οποιασδήποτε θέσης, ακόμη και του κεντρικού σταθμού. Για μερικούς οργανισμούς η βιομηχανίες που βλέπουνε το σύστημα SCADA ως εξαιρετικά κρίσιμη λειτουργία, αυτό είναι ένα πραγματικό όφελος.

## 1.5 Wide Area Network (WAN)

Το **WAN** συνδέει τον κεντρικό υπολογιστή με τις διατάξεις πολυπλεξίας σήματος. Αυτό μπορεί να είναι καλώδιο, ράδιο ή κάποιος δορυφόρος επικοινωνίας δεδομένων ανάλογα με την γεωγραφική κατανομή του SCADA συστήματος. Οι WAN συνδέσεις είναι γενικά αμφίδρομες (παρέχουν ταυτόχρονα μετάδοση δεδομένων και στις δύο κατευθύνσεις) και σχηματίζονται σε μορφή αστεριού ή κύκλου.

Ο σχηματισμός του κύκλου (σχήμα 1.10) συνδέει γειτονικές διατάξεις, παρέχει εναλλακτικά μονοπάτια επικοινωνίας και οι σύνδεσμοι πρέπει να διαστασιολογούνται έτσι ώστε να μπορούν φέρουν όλη την σύνδεση WAN.

Σε μερικές περιπτώσεις το WAN δεν χρειάζεται. Ένα παράδειγμα είναι ένα απλό σύστημα SCADA όπου όλα τα RTU συνδέονται άμεσα με τον κεντρικό υπολογιστή μέσω μίας απλής multi-drop σύνδεσης επικοινωνίας. Αυτά τα συστήματα επομένως περιέχουν αποτελεσματικά μόνο ένα RTU τοπικό δίκτυο.



Σχήμα 1.10: WAN σε σχηματισμό κύκλου

## 1.6 Πρωτόκολλα επικοινωνιών SCADA

Ένα **πρωτόκολλο** θα καθορίσει τις πληροφορίες όπως το μήκος του χρόνου που κάθε πακέτο δεδομένων χρειάζεται, το μέγεθος του σήματος και τον απαιτούμενο προορισμό για τα δεδομένα. Αντίθετα, ένα **ανοιχτό σύστημα** είναι αυτό που επιτρέπει τις επικοινωνίες μεταξύ διαφορετικών τύπων συσκευών.

Τα ιδιόκτητα συστήματα είναι εξ' ορισμού κλειστά και επιτρέπουν επικοινωνίες μόνο μεταξύ των συσκευών ίδιου τύπου. Με τα ανοιχτά συστήματα αποφεύγονται τα μειονεκτήματα που σχετίζονται με την χρήση των ιδιόκτητων συστημάτων, όπως η πλήρης εξάρτηση στον πωλητή και έλλειψη πληροφοριών για τις λειτουργίες του πρωτοκόλλου.

Παρόλα αυτά, για να κατανοηθούν τα πλεονεκτήματα των ανοιχτών συστημάτων, απαιτούνται λεπτομερή πρότυπα πρωτοκόλλων επικοινωνιών για να προσδιορίσουν όλες τις πλευρές της αλληλοσύνδεσης μεταξύ υπολογιστών και των άλλων συσκευών.

Η **διαθεσιμότητα** που υπάρχει στη δομή των επικοινωνιών είναι μία σημαντική πλευρά του SCADA συστήματος. Επειδή τα SCADA συστήματα συνήθως αναπτύσσονται σε μεγάλες γεωγραφικές περιοχές, συνδέσεις των απομακρυσμένων SCADA σταθμών από τον κεντρικό υπολογιστή συνήθως είναι πολυστρωματικές, εννοώντας ότι μπορεί να υπάρχουν μερικά φυσικά και λογικά μονοπάτια μέσω των οποίων τα δεδομένα πρέπει να δρομολογηθούν πριν φτάσουν στον προορισμό τους. Σε αυτές τις συνδέσεις θα πρέπει να δοθεί μεγάλη σημασία από οικονομικής πλευράς στον τύπο των συστημάτων επικοινωνιών που θα χρησιμοποιηθούν και στο εύρος ζώνης που θα χρησιμοποιηθεί σε αυτές τις συνδέσεις.

Λόγω των οικονομικών περιορισμών, οι συνδέσεις των SCADA επικοινωνιών γενικά προσφέρουν λιγότερο εύρος ζώνης και μικρότερη αξιοπιστία από αυτά που προσφέρονται από τις επικοινωνίες που χρησιμοποιούνται συνήθως σε μία εγκατάσταση, όπου υπάρχουν λίγοι γεωγραφικοί περιορισμοί και LAN κατασκευές μεγάλης ταχύτητας.

Το πρόβλημα στη διαθεσιμότητα μπορεί να αποδοθεί στο γεγονός ότι οι πολυστρωματικές SCADA συνδέσεις διατρέχουν έναν μεγαλύτερο αριθμό μέσων μετατροπής και θυρών δρομολόγησης δεδομένων συγκριτικά με τις LAN

κατασκευές υψηλής ταχύτητας. Επομένως, υπάρχουν πολλά σημεία αποτυχίας σε πολλά δίκτυα επικοινωνιών SCADA. Οι βλάβες στις επικοινωνίες συνήθως προκύπτουν από τον εξοπλισμό και από τις βλάβες στην παροχή ενέργειας και στην ανθρώπινη παρέμβαση. Καλύτερη διαθεσιμότητα είναι δυνατή μέσω της χρήσης εφεδρικών μονοπατιών επικοινωνίας στους εξωτερικούς σταθμούς. Παρόλα αυτά, τέτοιοι σχεδιασμοί μπορούν να συνεισφέρουν σημαντικά στο κόστος του συστήματος επικοινωνιών και επομένως δεν μπορεί να είναι οικονομικά βιώσιμο αν η σύνδεση των επικοινωνιών δεν είναι σημαντική στην λειτουργική ασφάλεια.

Τα πρωτόκολλα της επικοινωνίας SCADA είναι σχεδιασμένα ειδικά για την μειωμένη αξιοπιστία των συνδέσμων επικοινωνίας που συνήθως χρησιμοποιούνται στα SCADA συστήματα και παρέχουν ασφαλή μεταφορά των δεδομένων, διασφαλίζοντας την αξιόπιστη μεταφορά των δεδομένων στους προορισμούς κάτω από οποιαδήποτε περίπτωση. Τα πρωτόκολλα συμβάλλουν στον εντοπισμό του σφάλματος και στις τεχνικές ανασκόπησης των μηνυμάτων. Παρόλα αυτά, οι παραπάνω πληροφορίες που εισάγονται δημιουργούν πρόβλημα στην μετάδοση των δεδομένων, καταλήγοντας σε προβλήματα μεταξύ της ταχύτητας μετάδοσης των δεδομένων και της αξιοπιστίας του συνδέσμου των επικοινωνιών. Σαν αποτέλεσμα, η ταχύτητα των επικοινωνιών των δεδομένων που σχετίζονται με το SCADA είναι χαμηλότερη από την συνηθισμένη στις επικοινωνίες που χρησιμοποιούνται σε ένα γραφείο ή σε κάποιον όροφο ενός εργοστασίου.

Οι χρήστες των SCADA συστημάτων και τα δεδομένα που προκύπτουν δεν χρειάζονται να είναι ενήμερα των πρωτοκόλλων επικοινωνίας που χρησιμοποιούνται. Παρόλα αυτά, είναι σημαντικό να κατανοηθεί ότι με την χρήση των συνδέσμων επικοινωνίας όπως το ράδιο, υπάρχει μία πιθανότητα αν και μικρή να συμβούν κάποια σφάλματα επικοινωνίας. Για παράδειγμα, μία εντολή ελέγχου θα μπορούσε να σταλεί σε λάθος προορισμό. Τα SCADA συστήματα συχνά ζητούν επιβεβαίωση από τον χρήστη να επιβεβαιώσει ότι μία εντολή ελέγχου απαιτείται. Αυτή η προσέγγιση παρέχει κάποιο επίπεδο προστασίας ενάντια στην περίπτωση να σταλεί μία εντολή σε λάθος προορισμό.



Ένα παράδειγμα πρωτοκόλλου SCADA επικοινωνιών είναι το **DNP 3.0 (Distributed Network Protocol)**, ένα ανεξάρτητο πρωτόκολλο που ενσωματώνει πολλαπλά στρώματα εντοπισμού σφαλμάτων και διόρθωσης και επιτρέπει την επιβεβαίωση των εντολών ελέγχου. Το **Modbus** (αναλύεται παρακάτω) είναι άλλο ένα ευρέως διαδεδομένο πρωτόκολλο για το SCADA, αλλά δεν προσφέρει το ίδιο επίπεδο ασφάλειας στην μετάδοση δεδομένων όπως το DNP 3.0. Υπάρχει επίσης μία μεγάλη ποικιλία πρωτοκόλλων που προσφέρουν μαζί με τα συστήματα SCADA και προσφέρουν πολλές δυνατότητες όπως τα **EIA-232, EIA-422/485, RP-570, Profibus, Conitel** και **HART**.

Παρόλο που υπάρχουν διαφωνίες ως προς το κατά πόσο είναι ασφαλές να συνδέσει κανείς ένα σύστημα SCADA στο internet, πολλές βιομηχανίες (όπως συλλογής αποβλήτων και διανομής νερού), έχουν χρησιμοποιήσει τα υφιστάμενα δίκτυα κινητής τηλεφωνίας για την παρακολούθηση των υποδομών τους καθώς και διαδικτυακές πύλες για αποστολή δεδομένων. Τα δεδομένα που μεταφέρονται μέσω δικτύων κινητής τηλεφωνίας, κωδικοποιούνται πριν την αποστολή τους.

Όταν υπάρχουν αυξανόμενες απαιτήσεις ασφαλείας για προστασία κρίσιμων εγκαταστάσεων υπάρχει αυξανόμενη χρήση επικοινωνίας μέσω δορυφόρου. Αυτός ο τρόπος επικοινωνίας έχει το πλεονέκτημα ότι η εγκατάσταση μπορεί να είναι αυτόνομη (να μην χρειάζεται δηλαδή να χρησιμοποιήσει τα δημόσια δίκτυα επικοινωνίας), να έχει ενσωματωμένη κωδικοποίηση και μπορεί να κατασκευαστεί με τη διαθεσιμότητα και την αξιόπιστα που απαιτείται από το διαχειριστή του συστήματος SCADA.

### 1.6.1 Πρωτόκολλο EIA-232

Το standard EIA-232 αναπτύχθηκε με μοναδικό σκοπό την επικοινωνία μεταξύ τερματικών σταθμών (Data Terminal Equipment - DTE) και σταθμών τερματισμού κυκλωμάτων Data (Data Circuit terminating Equipment - DCE). Το EIA-232 εκδόθηκε στις ΗΠΑ το 1969 και μέχρι το 1988 έγινε ευρέως γνωστό ως RS-232. Η τρέχουσα έκδοση (EIA/TIA-232E) ολοκληρώθηκε το 1991 οπότε και το πρωτόκολλο ήρθε σε συμφωνία με τα standards ITU V.24, ITU V.28 και ISO 2110.

Το EIA-232 ορίζει τα χαρακτηριστικά του ηλεκτρικού σήματος όπως επίπεδα τάσης και χαρακτηριστικά γείωσης των εναλλασόμενων σημάτων καθώς και των αντίστοιχων κυκλωμάτων για ασύμμετρα (unbalanced) συστήματα.

Περιορισμοί:

- Ο κύριος περιορισμός αφορά στην point-to-point αρχιτεκτονική κατά την οποία μόνο 2 συσκευές δύναται να επικοινωνήσουν σε κάθε bus.
- Μέγιστη απόσταση μετάδοσης σήματος είναι τα 15 μέτρα (στα 20kbps).
- Ο ρυθμός μετάδοσης, 20k bits-per-second, είναι πολύ μικρός για τις περισσότερες εφαρμογές σήμερα.
- Τα εύρη τάσης -3 έως -25V και +3 έως +25V είναι συνήθως μη συμβατά με τα περισσότερα τροφοδοτικά.

Στη συνέχεια παραθέτουμε πίνακα με τα μέγιστα μήκη καλωδίου σε διαφορετικές ταχύτητες-ρυθμούς μετάδοσης:

Baud Rate	Μήκος (m)
110	850
300	800
600	700
1200	500
2400	200
4800	100
9600	70
19200	50
115000	20

## 1.6.2 Πρωτόκολλο EIA-422/485

Το standard EIA-485A αποτελεί ένα από τα πιο ευπροσάρμοστα standards της οικογένειας EIA. Είναι επέκταση του EIA-422 και επιτρέπει έως και 32 συσκευές-οδηγούς γραμμής (πομπούς) και 32 συσκευές-λήπτες, με μέγιστο μήκος καλωδίωσης τα 1200m και ταχύτητες έως 100MBps. Ορισμένοι κατασκευαστές προμηθεύουν συσκευές οι οποίες είναι ισοδύναμες με 1/2 ή 1/4 της συσκευής, επιτυγχάνοντας έτσι μέχρι και 64 ή 128 συσκευές ανά bus. Ο συνδιασμός όμως μέγιστης απόστασης-ταχύτητας δεν μπορούν να επιτευχθούν ταυτόχρονα.

Το δίκτυο EIA-422/EIA-485 βασίζεται στην αρχιτεκτονική Master/Slave. Για λόγους απόδοσης συνήθως απαιτούνται ενεργοί (solid state) τερματιστές οι οποίοι παίζουν τον ρόλο αντιστάσεων τερματισμού 120Ω.

Το EIA/485 είναι πολύ χρήσιμο για συστήματα όπου πολλά όργανα ή ελεγκτές πρέπει να συνδεθούν στην ίδια γραμμή, ιδιαίτερη προσοχή πρέπει να δωθεί όμως στο λογισμικό το οποίο διευθύνει ποιες συσκευές στο δίκτυο μπορούν να ενεργοποιηθούν.

### 1.6.3 Πρωτόκολλο Modbus

Το Modbus αναπτύχθηκε από την Gould Modicon (νυν Schneider Electric) για συστήματα ελέγχου διαδικασίας (process control). Παρόλα αυτά όμως θεωρείται ως ένα 'ανοικτό' πρωτόκολλο και αποτελεί standard σε συστήματα τα οποία 'παντρεύουν' στο ίδιο δίκτυο συσκευές από πολλαπλούς κατασκευαστές. Το Modbus είναι ένα απλό, ευέλικτο και ευρέως χρησιμοποιούμενο πρωτόκολλο, αρχιτεκτονικής Master/Slave, το οποίο επιτρέπει την ανταλλαγή διακριτών αναλογικών σημάτων μεταξύ συσκευών, κάνοντας χρήση των standard EIA-232 και EIA-422/485.

Παρότι είναι ένα πρωτόκολλο στο οποίο μπορεί κανείς να επενδύσει, γνωρίζοντας ότι η συντριπτική πλειοψηφία κατασκευαστών το υποστηρίζει, πάσχει από τους περιορισμούς των πιο πάνω σειριακών επικοινωνιών EIA-232 και EIA-422/485 όπως την περιορισμένη ταχύτητα 0.115Mbps σε αντίθεση τα μοντέρνα βιομηχανικά δίκτυα των 16Mbps και πλέον και ενώ η διασύνδεση 20-30 συσκευών είναι εύκολη υπόθεση κάνοντας χρήση του EIA-422/485, η σύνδεση για παράδειγμα 500 συσκευών είναι μια ιδιαίτερα σύνθετη ιεραρχία από Master-Slave συσκευές σε μια δενδροειδή αρχιτεκτονική ένθετων βρόγχων (nested loops).

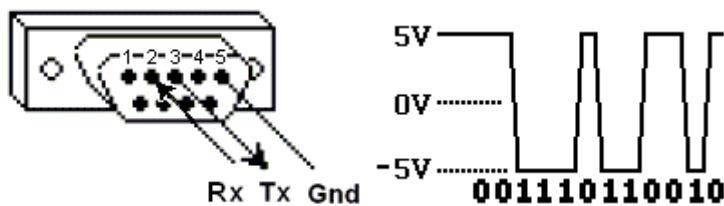
Το Modbus είναι ένα πρωτόκολλο επιπέδου εφαρμογής ανταλλαγής μηνυμάτων που τοποθετείται στο επίπεδο 7 του μοντέλου OSI και παρέχει client / server επικοινωνία μεταξύ συσκευών που συνδέονται με διαφορετικούς τύπους μέσων σύνδεσης και δικτύων. Οι κώδικες λειτουργίας του Modbus είναι στοιχεία του **Modbus request/reply PDU's** (Protocol Data Unit).

Με απλά λόγια, είναι μια μέθοδος που χρησιμοποιείται για τη διαβίβαση πληροφοριών μέσω σειριακών γραμμών μεταξύ ηλεκτρονικών συσκευών. Η συσκευή που ζητεί τις πληροφορίες ονομάζεται **Modbus Master** και οι συσκευές που παρέχουν πληροφορίες **Modbus Slaves**. Σε ένα τυπικό Modbus δίκτυο, υπάρχει ένας Master και μέχρι 247 Slaves, ο καθένας με μια μοναδική Slave διεύθυνση από 1 έως 247. Το Master μπορεί επίσης, να στείλει πληροφορίες στους Slaves.

Χρησιμοποιείται σε πολλαπλές εφαρμογές master-slave :

- Για την παρακολούθηση και προγραμματισμό διαφόρων συσκευών.
- Για την επικοινωνία μεταξύ ευφυών συσκευών με αισθητήρες και όργανα.
- Για την παρακολούθηση των συσκευών πεδίου χρησιμοποιώντας H/Y και HMI's.

Υπάρχουν διάφοροι τύποι του πρωτοκόλλου Modbus. Ο πιο κοινός τύπος είναι το **Modbus RTU** που βασίζεται σε σειριακή επικοινωνία, όπως RS485 και RS232. Υπάρχει επίσης το πρωτόκολλο **Modbus ASCII**. Σήμερα, η χρήση του πρωτοκόλλου **Modbus over Ethernet** έχει αυξηθεί σημαντικά. Ονομάζεται **Modbus TCP** και είναι βασικά το πρωτόκολλο Modbus RTU με ενσωματωμένα πακέτα TCP/IP.



Σχήμα 1.10: Σειριακή επικοινωνία με bits του Modbus

Το πρωτόκολλο Modbus μεταδίδεται μέσω σειριακής γραμμής μεταξύ συσκευών. Η πιο απλή εγκατάσταση θα ήταν ένα σειριακό καλώδιο που συνδέει τις σειριακές θύρες δύο συσκευών, μιας Master και μιας Slave.

Τα δεδομένα διαβιβάζονται σαν σειρές από άσσοις και μηδενικά και ονομάζονται bits. Κάθε bit αποστέλλεται ως τάση. Τα μηδενικά αποστέλλονται ως θετικές τάσεις και οι άσσοι ως αρνητικές. Τα bits αποστέλλονται πολύ γρήγορα. Μια τυπική ταχύτητα μετάδοσης είναι 9600 baud (bit ανά δευτερόλεπτο).

Το Modbus είναι επίσης ιδανικό πρωτόκολλο για RTU εφαρμογές όπου απαιτείται ασύρματη επικοινωνία. Για αυτό το λόγο, έχει χρησιμοποιηθεί σε αναρίθμητες εφαρμογές σε υποσταθμούς φυσικού αερίου και πετρελαίου. Αλλά δεν είναι μόνο ένα βιομηχανικό πρωτόκολλο. Μπορεί επίσης να χρησιμοποιηθεί σε εφαρμογές κατασκευών, υποδομών, μεταφορών και ενέργειας

Υπάρχουν αρκετά προτερήματα τα οποία έχουν ανάγκη το Modbus στο δημοφιλέστερο ίσως πρωτόκολλο βιομηχανικών δικτύων. Ορισμένα από τα χαρακτηριστικά του Modbus είναι σταθερά, όπως το format των πακέτων δεδομένων, η σειρά της αλληλουχίας των πακέτων ή η διαχείριση σφαλμάτων, ενώ άλλα είναι επιλέξιμα όπως το μέσο μετάδοσης, χαρακτηριστικά της μετάδοσης (time intervals), ή την κατάσταση μετάδοσης ASCII ή RTU που μπορούν να χρησιμοποιηθούν εναλλάξ για αποσφαλμάτωση και κανονική λειτουργία.

Το μεγαλύτερο όμως προτέρημα είναι ο έλεγχος σφαλμάτων που αποτελεί σύνολο διαφορετικών τεχνικών όπως parity check, cyclic redundancy check (CRC) και character framing. Το Modbus μπορεί να λειτουργήσει μέσω κοινών TCP/IP τοπικών ή και wide area δικτύων αφού παρέχει τη δυνατότητα ενσωμάτωσης των πακέτων του σε πακέτα TCP/IP.

### 1.6.4 Πρωτόκολλο HART

Τα πρωτόκολλα έξυπνων (ή ευφυών) οργάνων είναι σχεδιασμένα για εφαρμογές όπου τα δεδομένα συλλέγονται από όργανα, αισθητήρες και ενεργοποιητές μέσω τεχνικών ψηφιακής επικοινωνίας. Τα τμήματα αυτά συνδέονται απ' ευθείας σε Προγραμματιζόμενους Λογικούς Ελεγκτές (PLC) ή/και ηλεκτρονικούς υπολογιστές.

Το HART (Highway Addressable Remote Transducer) είναι παρακλάδι του τυπικού Fieldbus και λειτουργεί σε μια υβριδική (αναλογική-ψηφιακή) κατάσταση τύπου 4-20mA. Το HART σε καμία περίπτωση δεν αποτελεί το μοναδικό πρωτόκολλο τέτοιας φύσεως. Υπάρχουν εκατοντάδες παρόμοιες υλοποιήσεις από διάφορους κατασκευαστές, όπως για παράδειγμα της Honeywell, της οποίας το αντίστοιχο πρωτόκολλο αποτελεί τον κύριο ανταγωνιστή του HART.

Το HART αρχικά δημιουργήθηκε από την Rosemount και θεωρείται επίσης ως ανοικτό πρωτόκολλο, διαθέσιμο σε όλους τους κατασκευαστές. Το κυρίως προτέρημά του είναι ότι επιτρέπει σε έναν τεχνικό οργάνων, να διατηρήσει την υφιστάμενη εγκατάσταση οργάνων 4-20mA και παράλληλα να χρησιμοποιήσει την ίδια καλωδίωση για τη μετάδοση των σημάτων.

Για τον λόγο αυτό, το HART είναι ένα υβριδικό (αναλογικό/ψηφιακό) πρωτόκολλο, εν αντιθέσει με τα υπόλοιπα πρωτόκολλα της οικογένειας Fieldbus. Αυτό επιτυγχάνεται χρησιμοποιώντας την τεχνική Frequency Shift Keying η οποία βασίζεται στο standard Bell 202 της Bell Laboratories. Δύο ξεχωριστές συχνότητες στα 1200 και 2200Hz αναπαριστούν τα ψηφία 0 και 1 αντίστοιχα. Η μέση τιμή του ημιτονοειδούς κύματος (στις συχνότητες 1200 και 2200Hz) και η οποία άππει στο σήμα των 4-20mA, είναι μηδενική οπότε και η αναλογική πληροφορία των 4-20mA δεν αλλοιώνεται.

### 1.6.5 Πρωτόκολλο Profibus PA/DP/FMS

Το Profibus (PROcess FieLd BUS) είναι διεθνές και ευρέως αποδεκτό, συναντάται συνήθως σε μεγάλου μεγέθους βιομηχανίες. Είναι ανοικτό πρωτόκολλο, και ανεξάρτητο κατασκευαστή καθώς εμπίπτει στο OSI model, εξασφαλίζοντας ότι συσκευές από διαφορετικούς κατασκευαστές επικοινωνούν μεταξύ τους εύκολα και αποδοτικά.

Οριστικοποιήθηκε σαν standard 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. Βασίζεται επίσης στο standard EIA-485 και υποστηρίζεται από τον Profibus Trade Organization

Το Profibus υποστηρίζει μέχρι και 127 κόμβους με μέγιστη απόσταση από άκρο σε άκρο τα 27km κάνοντας χρήση οπτικών ινών και αναμεταδοτών (repeater). 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 αισθητήρες, μετατροπείς. Επιτρέπεται μόνο να αναγνωρίσουν (acknowledge) τη λήψη μηνυμάτων ή να αποστείλουν μήνυμα κατόπιν αιτήματος της Master.



Υπάρχουν τρεις βασικές κατηγορίες συστημάτων Profibus

- 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 εκάστωτε Slave) έχει τη δυνατότητα να εγγράψει δεδομένα σε αυτή.
- 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.
- 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).

## 1.7 Οφέλη των Συστημάτων SCADA

Το αρχικό κόστος ενός συστήματος SCADA πρέπει να δικαιολογηθεί. Μερικοί χαρακτηριστικοί λόγοι για ένα σύστημα SCADA είναι:

- Βελτιωμένη λειτουργία των εγκαταστάσεων ή της διαδικασίας με συνέπεια την αποταμίευση λόγω της βελτιστοποίησης του συστήματος
- Αυξανόμενη παραγωγικότητα του προσωπικού
- Βελτιωμένη ασφάλεια του συστήματος λόγω των καλύτερων πληροφοριών και του βελτιωμένου ελέγχου
- Προστασία του εξοπλισμού εγκαταστάσεων
- Προστασία του περιβάλλοντος από μια αποτυχία του συστήματος
- Βελτιωμένη αποταμίευση ενέργειας λόγω της βελτιστοποίησης των εγκαταστάσεων

Το λογισμικό SCADA μπορεί να διαιρεθεί σε δύο τύπους, ιδιόκτητου ή ανοικτού κώδικα . Οι επιχειρήσεις αναπτύσσουν το ιδιόκτητο λογισμικό για να επικοινωνήσουν με το υλικό τους. Αυτά τα συστήματα πωλούνται ως βασικές "λύσεις" συστημάτων SCADA. Το κύριο πρόβλημα με αυτά τα σύστημα είναι η συντριπτική εμπιστοσύνη στον προμηθευτή του συστήματος.

Τα ανοικτά συστήματα λογισμικού έχουν κερδίσει τη δημοτικότητα λόγω της διαλειτουργικότητας που φέρνουν στο σύστημα. Η διαλειτουργικότητα είναι η δυνατότητα να αναμιχθεί ο εξοπλισμός των διαφορετικών κατασκευαστών στο ίδιο σύστημα.

Το Citect και WonderWare είναι ακριβώς δύο από τα ανοικτά πακέτα λογισμικού διαθέσιμα στην αγορά για τα συστήματα SCADA. Μερικές συσκευασίες περιλαμβάνουν τώρα τη διαχείριση ενεργητικών στοιχείων που ενσωματώνονται μέσα στο σύστημα SCADA.

Τα κύρια χαρακτηριστικά του λογισμικού SCADA είναι:

- Διεπιφάνεια χρήστη
- Γραφικό περιβάλλον
- συναγερμοί
- διεπαφή του PLC εξελξιμότητα
- πρόσβαση στα δεδομένα
- βάση δεδομένων
- δικτύωση
- ανοχή ελαττωμάτων και πλεονασμός
- Client/server διανεμημένη επεξεργασία

Υπάρχουν χαρακτηριστικά πέντε στόχοι σε οποιοδήποτε σύστημα SCADA. Κάθε μια από αυτές τις διεργασίες εκτελεί και χωριστή επεξεργασία.

- **Διεργασίες εισόδων - εξόδων**

Αυτό το πρόγραμμα είναι η διεπαφή μεταξύ του συστήματος ελέγχου και παρακολούθησης και των εγκαταστάσεων.

- **Διεργασίες Συναγερμών**

Διαχειρίζεται όλους τους συναγερμούς με την ανίχνευση των ψηφιακών σημείων συναγερμών και τη σύγκριση των τιμών των αναλογικών σημείων συναγερμών.

- **Διεργασίες ενεργειών**

Η διεργασία ενεργειών συλλέγει τα στοιχεία που ελέγχονται με την πάροδο του χρόνου.

- **Διεργασίες αναφορών**

Οι αναφορές συντάσσονται από τα στοιχεία των εγκαταστάσεων. Αυτές οι αναφορές είναι περιοδικές, γεγονός που προκαλείται ή που ενεργοποιείται από το χειριστή.

- **Ενδείξεις διεργασιών**

Αυτό διαχειρίζεται όλα τα στοιχεία που ελέγχονται από το χειριστή και όλες οι ενέργειες ελέγχου από το χειριστή.

## 1.8 Το μέλλον των συστημάτων SCADA

Οι προμηθευτές SCADA κυκλοφορούν μια σημαντική έκδοση και μια έως δύο πρόσθετες δευτερεύουσες εκδόσεις το χρόνο. Αυτά τα προϊόντα εξελίσσονται πολύ γρήγορα ώστε να εκμεταλλευτούν τις νέες ευκαιρίες αγοράς, για να καλύψουν τις νέες απαιτήσεις των πελατών τους και τις νέες τεχνολογίες.

Περισσότερα από τα προϊόντα SCADA που αξιολογήθηκαν συνθέτουν τη διαδικασία σε «ατομικές» παραμέτρους στις οποίες συνδέεται ένα ετικέτα-όνομα (Tag-name). Αυτό είναι μη πρακτικό στην περίπτωση πολύ μεγάλων διαδικασιών όταν πρέπει να διαμορφωθούν πολύ μεγάλα σύνολα ετικετών.

Δεδομένου ότι οι βιομηχανικές εφαρμογές αυξάνονται στο μέγεθος, οι νέες εκδόσεις SCADA έχουν ως σκοπό τώρα να χειριστούν τις συσκευές όπως και ολόκληρα συστήματα ως πλήρεις οντότητες (classes) που τοποθετούν όλες τις συγκεκριμένες ιδιότητες και τη λειτουργία τους σε αυτά. Επιπλέον, θα υποστηρίξουν 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. Στη μέση της δεκαετίας του '90, ο κατασκευαστής των I/O προσέφεραν τα ιδιόκτητα πρωτόκολλα επικοινωνιών και απλούς τρόπους σύνδεσης των συσκευών. Προς την δεκαετία του '90, η μετατόπιση προς τις ανοικτές επικοινωνίες συνεχίστηκε με τους I/O κατασκευαστές να προσφέρουν την υποστήριξη των ανοικτών δομών μηνυμάτων και ανοιχτών πρωτοκόλλων με σύνδεση μέσω της θύρας Ethernet και του πρωτοκόλλου TCP/IP.

Τα αρχικά εμπόδια της εισόδου Ethernet TCP/IP στη βιομηχανική αυτοματοποίηση (αιτιοκρατία, συγχρονισμός, επιλογή πρωτοκόλλου, καταλληλότητα περιβάλλοντος) είναι ακόμα μια ανησυχία σε μερικές εξαιρετικά εξειδικευμένες εφαρμογές, αλλά για τη μεγάλη πλειοψηφία των αγορών HMI/SCADA αυτά τα εμπόδια έχουν σπάσει. Πρόσφατα, εντούτοις, η ίδια η ύπαρξη των συστημάτων SCADA και της χρήσης του διαδικτύου (TCP/IP) έχει προκαλέσει προβλήματα και ερωτήματα ως αναφορά την ασφάλεια τους καθώς παρατηρούνται όλο και περισσότερες επιθέσεις και τα συστήματα παρουσιάζονται εξαιρετικά τρωτά στις επιθέσεις μέσω του διαδικτύου. Λαμβάνοντας υπόψη την αποστολή και την κρίσιμη φύση ενός μεγάλου αριθμού συστημάτων SCADA,

τέτοιες επιθέσεις θα μπορούσαν, στην χειρότερη περίπτωση, να προκαλέσουν τις ογκώδεις οικονομικές απώλειες μέσω της απώλειας δεδομένων ή πραγματικής φυσικής καταστροφής, κακής χρήσης ή κλοπής, ακόμη και απώλειας ζωής, είτε άμεσα είτε έμμεσα.

Γίνονται λοιπόν μεγάλες προσπάθειες, κυρίως σε επίπεδο λογισμικού αλλά και υλικού, ώστε στο μέλλον να έχουμε συστήματα SCADA ασφαλέστερα και λιγότερο εκτεθειμένα σε οποιοσδήποτε μορφής επιθέσεων. Φυσικά με απώτερο και επιθυμητό στόχο όσο το δυνατόν ασφαλέστερα αλλά και γρηγορότερα, σε απόκριση και μεταφορά δεδομένων, συστήματα SCADA.

Εάν τέτοιες ανησυχίες θα προκαλέσουν μια απομάκρυνση από τη χρήση των συστημάτων SCADA για τις κρίσιμες εφαρμογές προς τις ασφαλέστερες αρχιτεκτονικές και τις διαμορφώσεις είναι ένα τεράστιο ζήτημα, δεδομένου ότι τουλάχιστον μερικοί επιδρώντες άνθρωποι στους εταιρικούς και κυβερνητικούς κύκλους θεωρούν ότι τα κέρδη και οι χαμηλότερες αρχικές δαπάνες των βασισμένων σε SCADA συστημάτων αντισταθμίζουν ακόμα τις πιθανούς απώλειες και τους κινδύνους.

## 1.9 Εφαρμογές των 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) υπάρχει εγκατεστημένο ένα RTU, το οποίο καταγράφει 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 στους μετασχηματιστών, 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 τη 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 RTU. 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.

# Κεφάλαιο 2

Το περιβάλλον SCADA WinCC  
της SIEMENS

## 2.1 Εισαγωγή στο HMI(human machine interface)

Η **διασύνδεση ανθρώπου μηχανής** (human machine interface - HMI) είναι 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.

Τα **συστήματα HMI** αποτελούν το μέρος της λειτουργίας των SCADA που αλληλεπιδρά με τον τελικό χρήστη, δηλαδή του χειριστή του συστήματος. Συνήθως αποτελούνται από μια οπτική απεικόνιση της διεργασίας, πάνω στην οποία εμφανίζονται τιμές μεταβλητών, καταστάσεις ή και διαγράμματα. Ακόμη τα συστήματα αυτά επιτρέπουν την κατ' απαίτηση εμφάνιση ιστορικών των μεταβλητών, ειδικών διαγραμμάτων κτλ. Οι πληροφορίες αυτές αντλούνται από τη βάση δεδομένων του συστήματος SCADA, γι' αυτό όπως είναι κατανοητό τα συστήματα SCADA και HMI είναι αλληλένδετα συνδεδεμένα μεταξύ τους και για το λόγο αυτό συχνά δεν διαχωρίζονται.

Στην πραγματικότητα βέβαια οι περισσότεροι κατασκευαστές συστημάτων SCADA ενσωματώνουν την δυνατότητα ανάπτυξης HMI εφαρμογών στις υπηρεσίες ή τα πακέτα SCADA τους. Η ουσία είναι όμως ότι μια εφαρμογή HMI μπορεί να κατασκευαστεί ανεξάρτητα από των πυρήνα καταγραφής ενός SCADA ώστε να αντλεί πληροφορίες από αυτό και να το χρησιμοποιεί για να διεξάγει τον απαραίτητο έλεγχο.

Όπως αναφέρθηκε το HMI συνήθως χρησιμοποιεί γραφικό τρόπο για την παρουσίαση των πληροφοριών στο χειριστή. Αυτό σημαίνει ότι ο χειριστής μπορεί να δει σχηματική παρουσίαση της διεργασίας η οποία ελέγχεται. Για παράδειγμα, η εικόνα μιας αντλίας η οποία επικοινωνεί με ένα σωλήνα, μπορεί να δείξει ότι η αντλία δουλεύει καθώς και πόσο υγρό αντλεί από το σωλήνα τη στιγμή αυτή. Ο χειριστής μπορεί στη συνέχεια να κλείσει την αντλία. Στο σύστημα HMI υπάρχουν διάφορα γραφικά στοιχεία τα οποία χρησιμοποιούνται για να απεικονίσουν τα διάφορα στοιχεία της διάταξης προς έλεγχο.

Το HMI πακέτο για το σύστημα SCADA, περιλαμβάνει ένα σχεδιαστικό πρόγραμμα, το οποίο μπορεί να χρησιμοποιήσει ο χειριστής η ο συντηρητής του συστήματος ώστε να αλλάξει τον τρόπο με τον οποίο οι διάφορες μεταβλητές παρουσιάζονται στο σύστημα. Αυτή η απεικόνιση της διαδικασίας, μπορεί να είναι



εξαιρετικά απλή (όπως το να προβάλλεται ένα φανάρι στο σύστημα SCADA το οποίο αντιπροσωπεύει την κατάσταση ON ή OFF) ενός πραγματικού φαναριού στην ελεγχόμενη διάταξη, ή ιδιαίτερα σύνθετη (όπως η απεικόνιση της θέσης όλων των ανελκυστήρων ένας ουρανοξύστη ή όλων των τραίνων σε ένα σιδηρόδρομο).

Ένα σημαντικό μέρος των περισσότερων εφαρμογών SCADA είναι η διαχείριση συναγερμών. Το σύστημα παρακολουθεί κατά πόσο ορισμένες συνθήκες συναγερμού (Alarm) είναι ικανοποιημένες, ώστε να προσδιορίσει πότε ένα συμβάν συναγερμού έχει συμβεί. Μόλις ένα τέτοιο συμβάν εντοπιστεί, πραγματοποιούνται μία ή περισσότερες ενέργειες, όπως η ενεργοποίηση ενός ή περισσότερων ενδείξεων κινδύνου (alarm) ή ίσως η δημιουργία ενός e-mail ή μηνύματος κειμένου για την ειδοποίηση του αρμόδιου προσωπικού.

Σε πολλές περιπτώσεις, ο χειριστής του SCADA πρέπει να αναγνωρίσει το γεγονός συναγερμού ώστε π.χ. να σταματήσει η σειρά συναγερμού (που ενεργοποιείται σε σοβαρές κυρίως βλάβες) ή να σταματήσουν κάποιες ενδείξεις του συναγερμού. Ωστόσο, για να σταματήσουν τελείως όλες οι ενδείξεις συναγερμού, θα πρέπει η αιτία ενεργοποίησης του να μην υπάρχει πλέον.

Οι προϋποθέσεις ενεργοποίησης ενός συναγερμού μπορεί να είναι **άμεσες** ή **έμμεσες**. Άμεσες για παράδειγμα, είναι όταν η μεταβλητή ενεργοποίησης ενός συναγερμού είναι μία ψηφιακή μεταβλητή (Boolean), η οποία υπολογίζεται με κάποιο αλγόριθμο και η τιμή της εξαρτάται από άλλες αναλογικές ή ψηφιακές μεταβλητές. Έμμεσες είναι όταν, για παράδειγμα, το σύστημα SCADA παρακολουθεί αυτόματα εάν η τιμή μιας αναλογικής μεταβλητής βρίσκεται έξω από κάποια όρια τα οποία έχουν οριστεί μαζί με την μεταβλητή.

Παραδείγματα ενδείξεων συναγερμού είναι η ενεργοποίηση μιας σειράς, εμφάνιση παραθύρων στην οθόνη του SCADA τα οποία ενημερώνουν τον χειριστή για το συναγερμό, κάποιες ενδείξεις που αλλάζουν χρώμα όταν υπάρξει κάποιο γεγονός συναγερμού κλπ. Σε κάθε περίπτωση, ο ρόλος της ένδειξης ενός συναγερμού, είναι να τραβήξει την προσοχή του χειριστή στα σημεία της ελεγχόμενης διεργασίας στην οποία βρίσκεται ο συναγερμός, ώστε να ληφθούν τα κατάλληλα μέτρα.

## 2.2 Το περιβάλλον WinCC

Για την εκπόνηση της διπλωματικής εργασίας, αναπτύξαμε την εφαρμογή με τη βοήθεια του πακέτου SIMATIC WinCC V7.2.0.0 της εταιρείας SIEMENS. Παρακάτω γίνεται μία περιληπτική ανάλυση και περιγραφή του πακέτου αυτού, ώστε να γίνει κατανοητός ο βασικός τρόπος λειτουργίας του.

Το λογισμικό **Simatic WinCC** χρησιμοποιείται για την δημιουργία εφαρμογών SCADA και εφαρμογών HMI μέσα από τις οποίες ο εκάστοτε χρήστης θα μπορεί να παρακολουθεί σε γραφικό περιβάλλον και με αποδοτικό τρόπο την οποιαδήποτε διεργασία. Το WinCC προσφέρει μια σειρά από μηχανολογικά συστήματα τα οποία είναι ιδανικά προσαρμοσμένα στο αντίστοιχο υπό δημιουργία project ή μπορεί να προσαρμοστεί από το χρήστη. Κάθε έκδοση υποστηρίζει ένα ευρύ φάσμα HMI συστημάτων και λειτουργιών, με την οποία η Standard έκδοση μπορεί να χρησιμοποιηθεί για να τη ρύθμιση HMI. Η νέα έκδοση του WinCC προσφέρει στους χρήστες τις εξής δυνατότητες:

- Υποστήριξη νεότερων λειτουργικών συστημάτων:
  - Windows 7 SP1 (Ultimate / Professional / Enterprise, 32/64-bit)
  - Windows XP Professional SP3
  - Windows Server 2003 R2 SP2
  - Windows Server 2008 R2 SP1
- Διευρυμένη ποσότητα στην χρήση των servers και clients (μέχρι 18 servers για clients του WinCC σε ένα δίκτυο)
- Υποστήριξη επικοινωνίας με τους controllers της SIMATIC S7-1200 και S7-1500
- Ανανεωμένο configuration studio, απλουστευμένο για τη φιλικότερη προς τον χρήστη επεξεργασία των tags
- Ανανεωμένα Graphics Designer και Global Script editors, προσθέτοντας τη λειτουργία χρήσης κωδικού κλειδώματος, προστατεύοντας έτσι τα VBA scripts από ανεπιθύμητη εισβολή
- Υποστήριξη χρήσης πολλαπλών γλωσσών

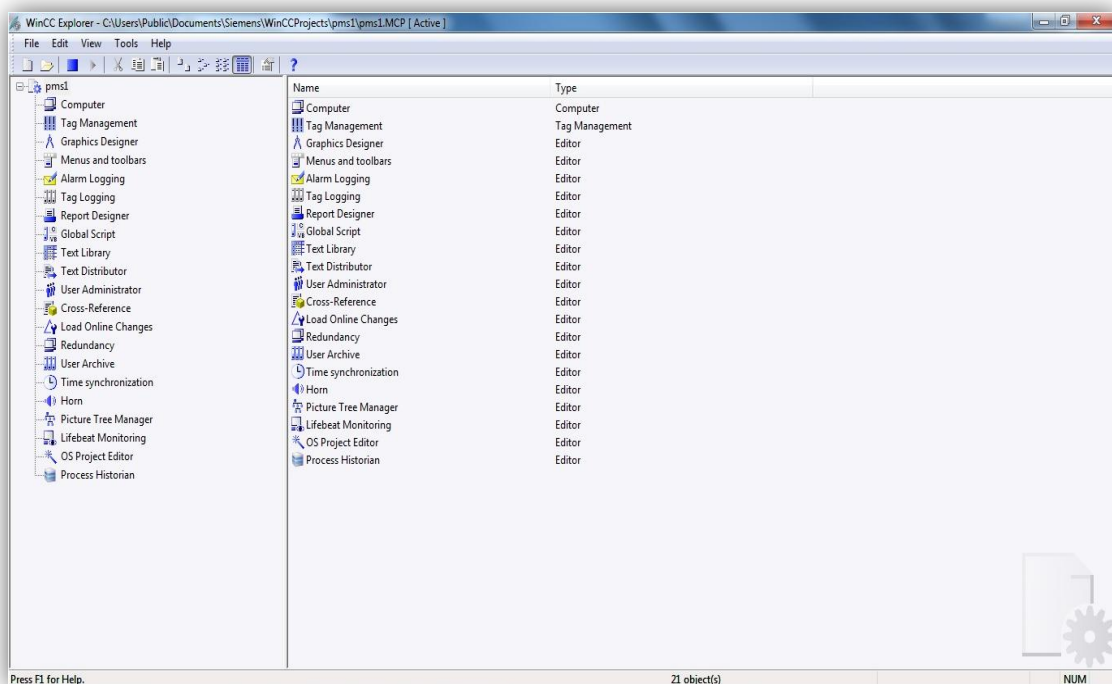
- Διευρυμένη λειτουργικότητα του WinCC/DataMonitor και του Web Navigator, προσφέροντας στον χρήστη βελτιωμένη ασφάλεια στη διαδικτυακή χρήση του WinCC
- Απλουστευμένη ενσωμάτωση και εξαγωγή των WinCC projects

## 2.3 Βασική δομή WinCC

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

- **Computer**
- **Tag Managements**
- **Data Types**
- **Editor**

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

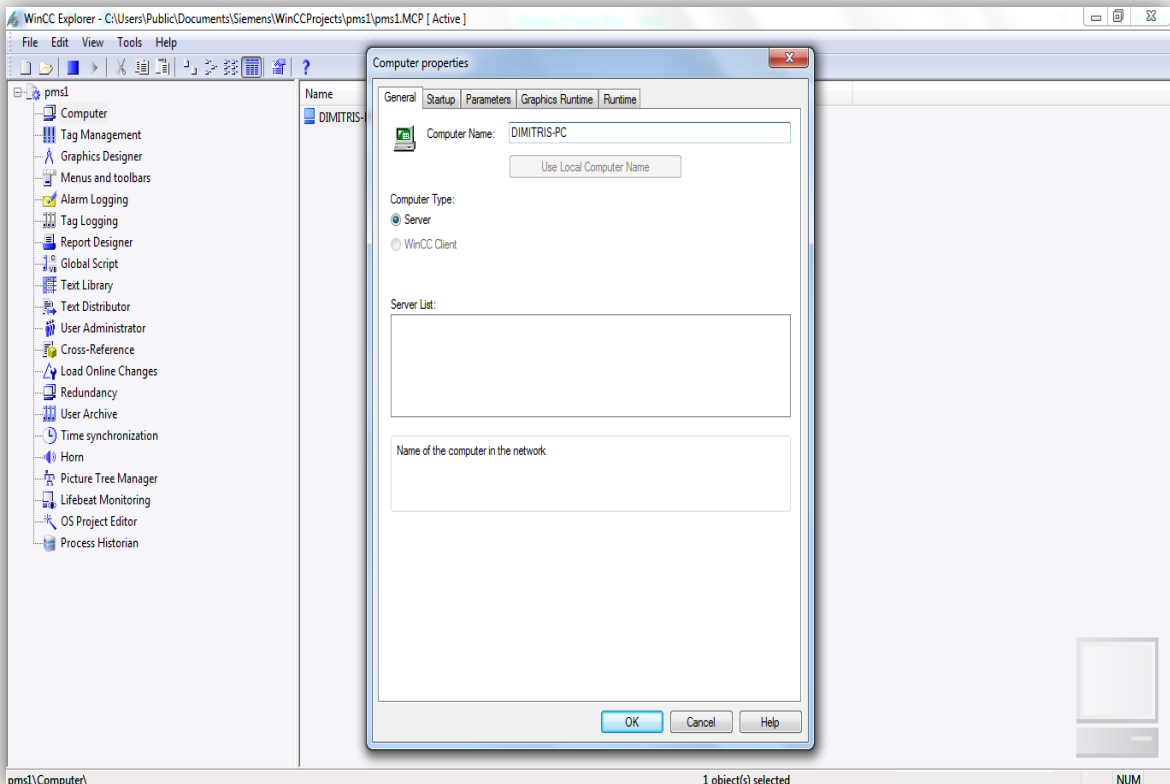


Σχήμα 1.11: WinCC explorer

## 2.4 Computer ( Υπολογιστής )

Στο τμήμα αυτό καθορίζονται, σε γενικές γραμμές, οι υπηρεσίες που θέλουμε να μας προσφέρει το WinCC. Πιο συγκεκριμένα, εδώ γίνεται η παραμετροποίηση των μεταβλητών που θέλουμε να παρακολουθούνται συνεχώς και να γίνεται καταγραφή των τιμών τους ή να γίνεται συγκεκριμένη αναφορά στην περίπτωση την οποία ξεπεραστεί κάποιο κρίσιμο όριο που εμείς έχουμε θέσει κτλ. Επιπλέον, υπάρχει η δυνατότητα να ορίζονται προγράμματα ανταλλαγής δεδομένων, σε απευθείας συνεργασία με το WinCC, όπως το Word, το Excel και το Autocad, τα οποία είναι ιδιαίτερα χρήσιμα σε περιπτώσεις διεξοδικής επεξεργασίας δεδομένων και κομψής παρουσίασής τους.

Τέλος, εδώ υπάρχουν διάφορες επιλογές για την παρουσίαση των γραφικών των εικόνων μας, τις οποίες και εκμεταλλευτήκαμε για την ολοκλήρωση των εφαρμογών μας, όπως θα δούμε στη συνέχεια

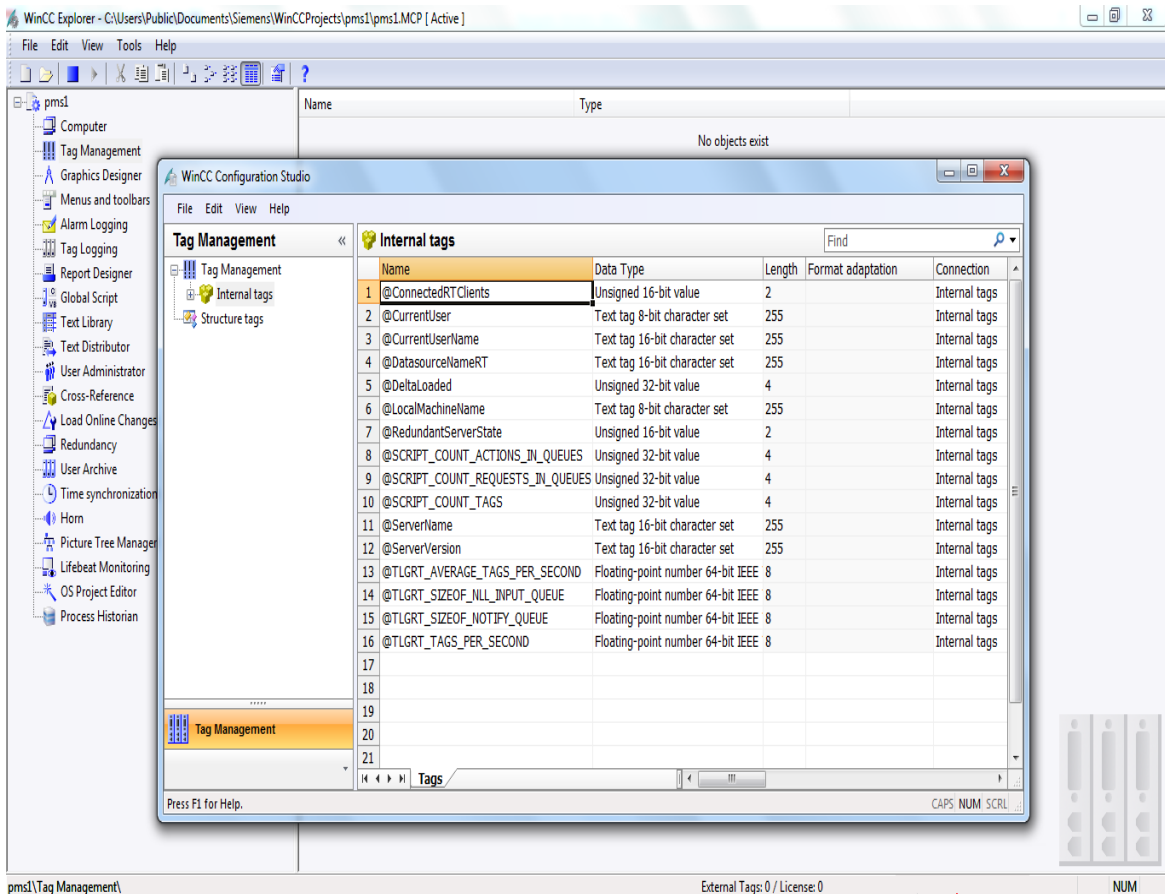


Σχήμα 1.12: Computer

## 2.5 Tag Management ( Διαχείριση Μεταβλητών )

Εδώ γίνεται ο καθορισμός του πρωτοκόλλου επικοινωνίας του υπολογιστή με το PLC και καθορίζονται οι μεταβλητές του συστήματος. Οι εν λόγω μεταβλητές διακρίνονται στις εξωτερικές μεταβλητές (external tags) και στις εσωτερικές μεταβλητές (internal tags).

Οι μεν εξωτερικές είναι στην ουσία μεταβλητές στην μνήμη δεδομένων του PLC, ενώ οι εσωτερικές χρησιμοποιούνται για διάφορες λειτουργίες εντός του WinCC. Όπως καταλαβαίνει κανείς, πρόκειται για ένα από τα πιο βασικά και πολύπλοκα τμήματα του WinCC, στο οποίο διαμορφώνεται η βασική δομή του συστήματος SCADA



Σχήμα 1.13: Tag management

## 2.6 Data Types ( Τύποι Μεταβλητών )

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

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

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

## 2.7 Editor

Ο Editor του WinCC αποτελείται από πέντε υποτμήματα τα οποία αναφέρονται παρακάτω.

### 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 εποπτεύσει το σύστημα με μεγάλη ακρίβεια και λεπτομέρεια που πολύ δύσκολα θα είχε αλλιώς τη δυνατότητα να κάνει.

### Tag Logging

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

### Alarm Logging

Ό,τι υπέρβαση ορίων στις τιμές μεταβλητών, τα οποία έχουμε θέσει εμείς, παρατηρείται κατά τη λειτουργία του συστήματος καταγράφεται εδώ. Ως εκ τούτου, ο ρόλος του 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.

## Report Designer

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

## Global Script

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



## 2.8 Ο WinCC Explorer

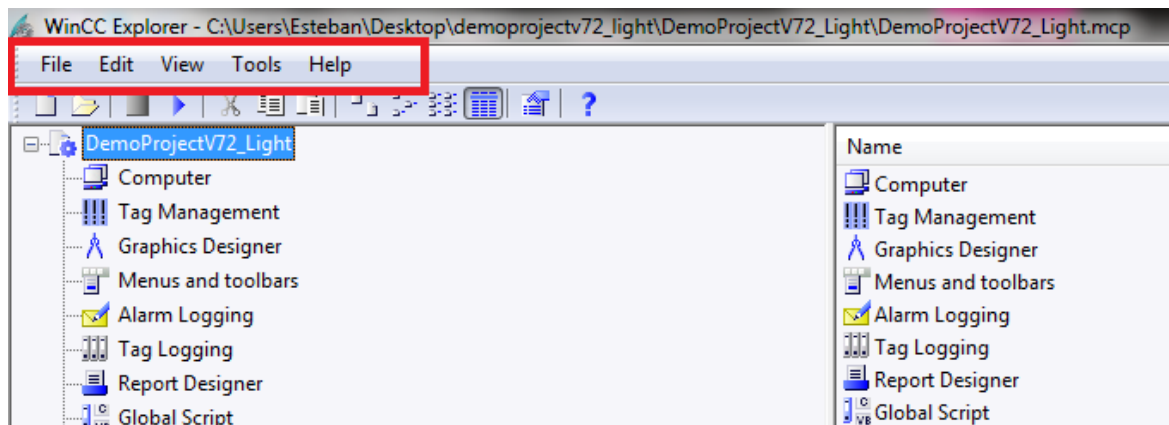
Ανοίγοντας τον WinCC Explorer υπάρχουν οι εξής επιλογές:

- **Create a project** (Δημιουργία νέου project)
- **Open a project** (Άνοιγμα ενός υπάρχοντος project)
- **Manage project data and archives** ( Διαχείριση δεδομένων και αρχείων)
- **Open editors** (Άνοιγμα του editor)
- **Activate or deactivate a project** (Ενεργοποίηση ή απενεργοποίηση ενός project)

### 2.8.1 Menu bar

Το menu bar του WinCC Explorer περιλαμβάνει τις εξής επιλογές:

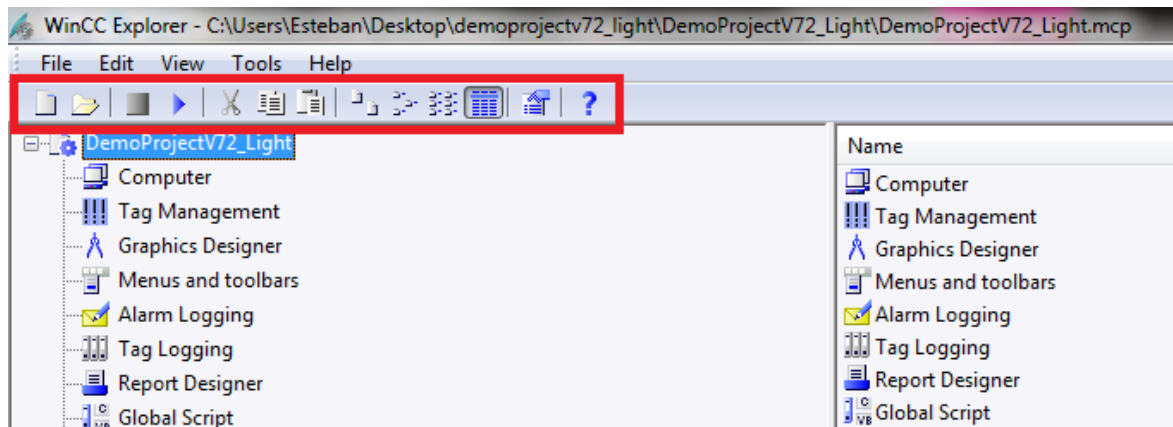
- **File menu**
- **Edit menu**
- **View menu**
- **Tools menu**
- **Help menu**



Σχήμα 1.14: Menu Bar

## 2.8.2 Toolbar

Ανοίγοντας τον WinCC Explorer βρίσκουμε το toolbar στην εξής θέση:



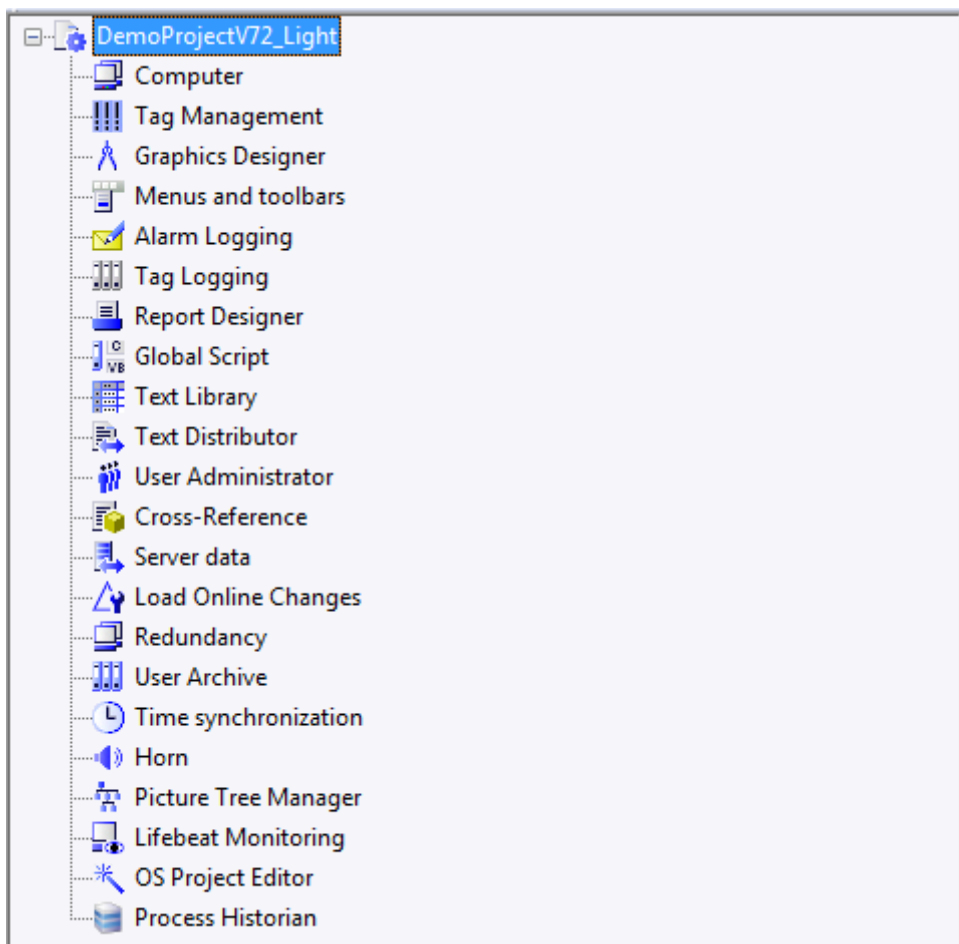
Σχήμα 1.15: Toolbar

Το toolbar περιλαμβάνει τις εξής επιλογές:

Εικονίδιο	Ονομασία	Περιγραφή
	Νέο	Δημιουργία νέου project
	Άνοιγμα	Άνοιγμα project
	Απενεργοποίηση	Έξοδος (παύση) εκτέλεσης προγράμματος
	Ενεργοποίηση	Εκτέλεση προγράμματος
	Αποκοπή	Αποκοπή ενός επιλεγμένου αντικειμένου
	Αντιγραφή	Αντιγραφή ενός επιλεγμένου αντικειμένου
	Επικόλληση	Επικόλληση ενός αντικειμένου
	Μεγάλα εικονίδια	Τα στοιχεία του παραθύρου να είναι μεγάλα
	Μικρά εικονίδια	Τα στοιχεία του παραθύρου να εμφανίζονται μικρά
	Λίστα	Τα στοιχεία του παραθύρου να εμφανίζονται σε λίστα
	Λεπτομέρειες	Τα στοιχεία του παραθύρου να εμφανίζονται με λεπτομέρειες

### 2.8.3 Navigation window

Το **navigation window** (παράθυρο πλοήγησης) περιλαμβάνει μια λίστα με τους Editors και τις συναρτήσεις που περιλαμβάνονται στο WinCC Explorer. Ο χρήστης μπορεί να ανοίξει ένα στοιχείο πατώντας διπλό-κλικ πάνω σε αυτό. Εναλλακτικά μπορεί να κάνει δεξι-κλικ και στη συνέχεια να επιλέξει open.



Σχήμα 1.16: Navigation Window

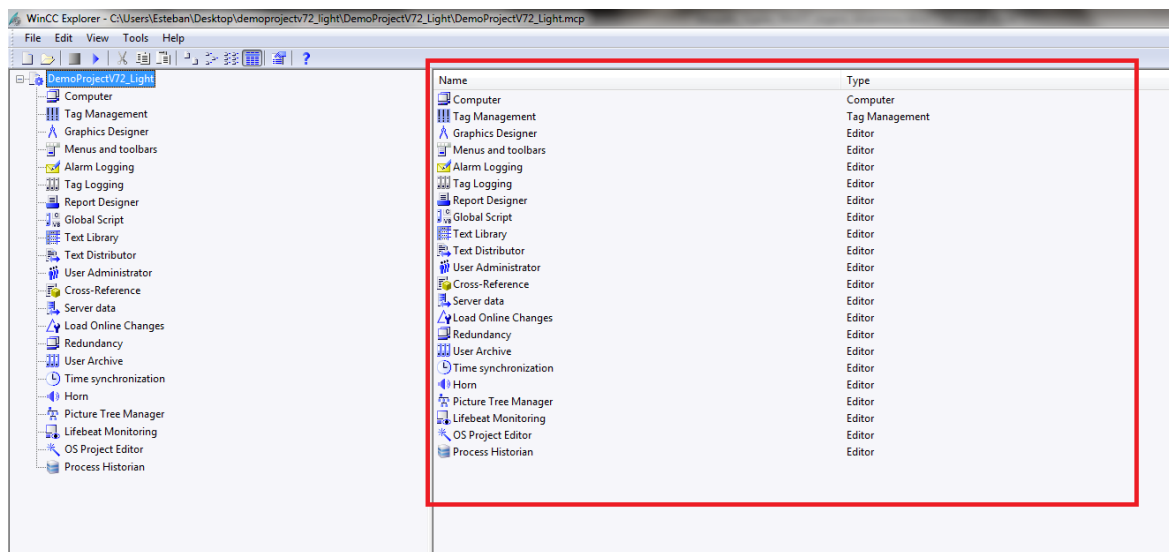
## 2.8.4 Tooltip

Όταν το κουμπί **Run** είναι ενεργοποιημένο τότε στο WinnCC εμφανίζεται ένα μήνυμα με πληροφορίες αναφορικά με τα **tags** και τις διάφορες συνδέσεις που υπάρχουν τι συγκεκριμένη στιγμή.

## 2.8.5 Data Window

Εάν ο χρήστης επιλέξει κάποιον Editor ή φάκελο τότε στο **data window** (παράθυρο δεδομένων) εμφανίζονται τα στοιχεία που ανήκουν στο συγκεκριμένο editor ή φάκελο. Οι πληροφορίες που εμφανίζονται είναι διαφορετικές από editor σε editor.

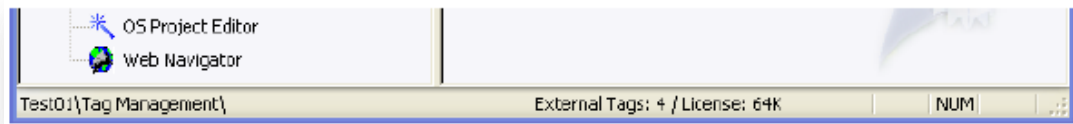
Πατώντας διπλό-κλικ πάνω σε ένα στοιχείο στο data window μπορεί ο χρήστης να το ανοίξει. Με **δεξί-κλικ** μπορεί ο χρήστης να επιλέξει τις ιδιότητες (**Properties**) του συγκεκριμένου στοιχείου.



Σχήμα 1.17: Data Window

### 2.8.6 Status bar

Το **status bar** περιέχει γενικές πληροφορίες για το project και τις ρυθμίσεις του συγκεκριμένου editor.

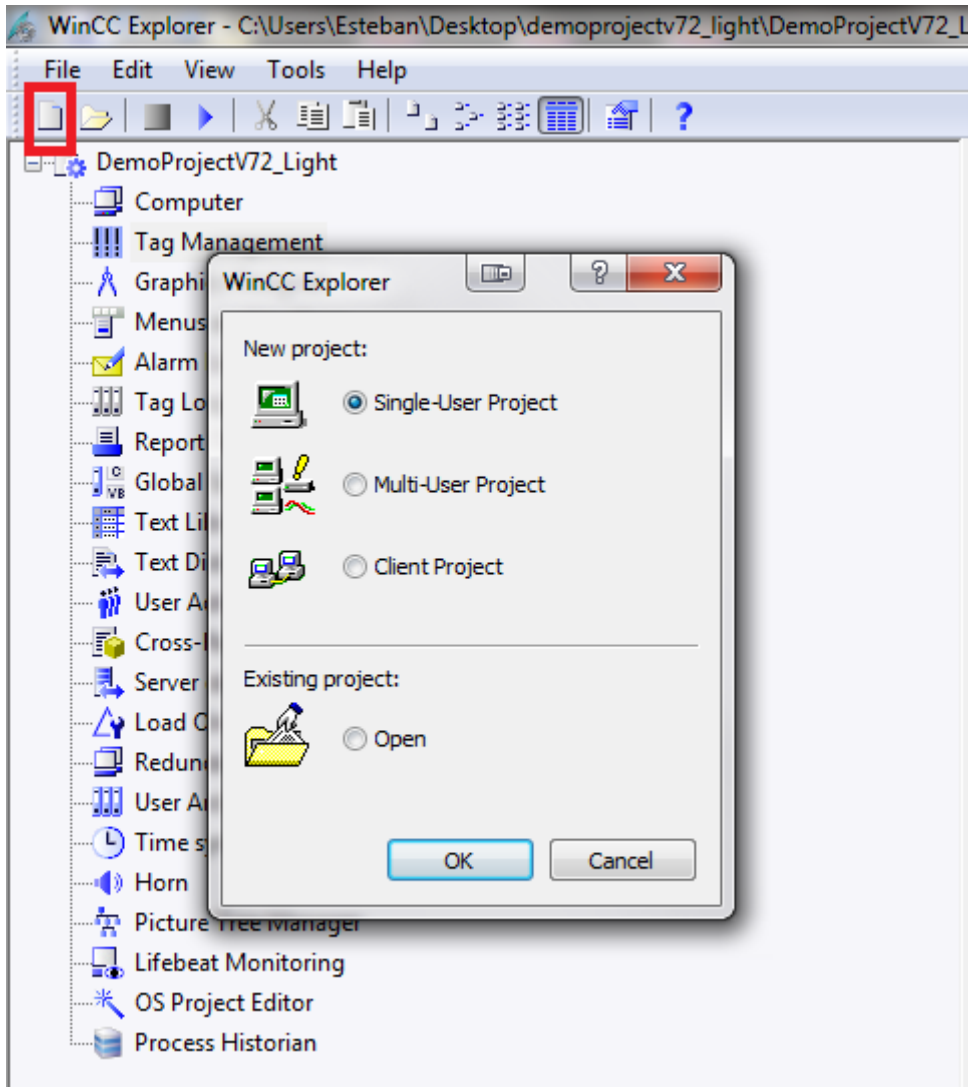


Σχήμα 1.18: Status Bar

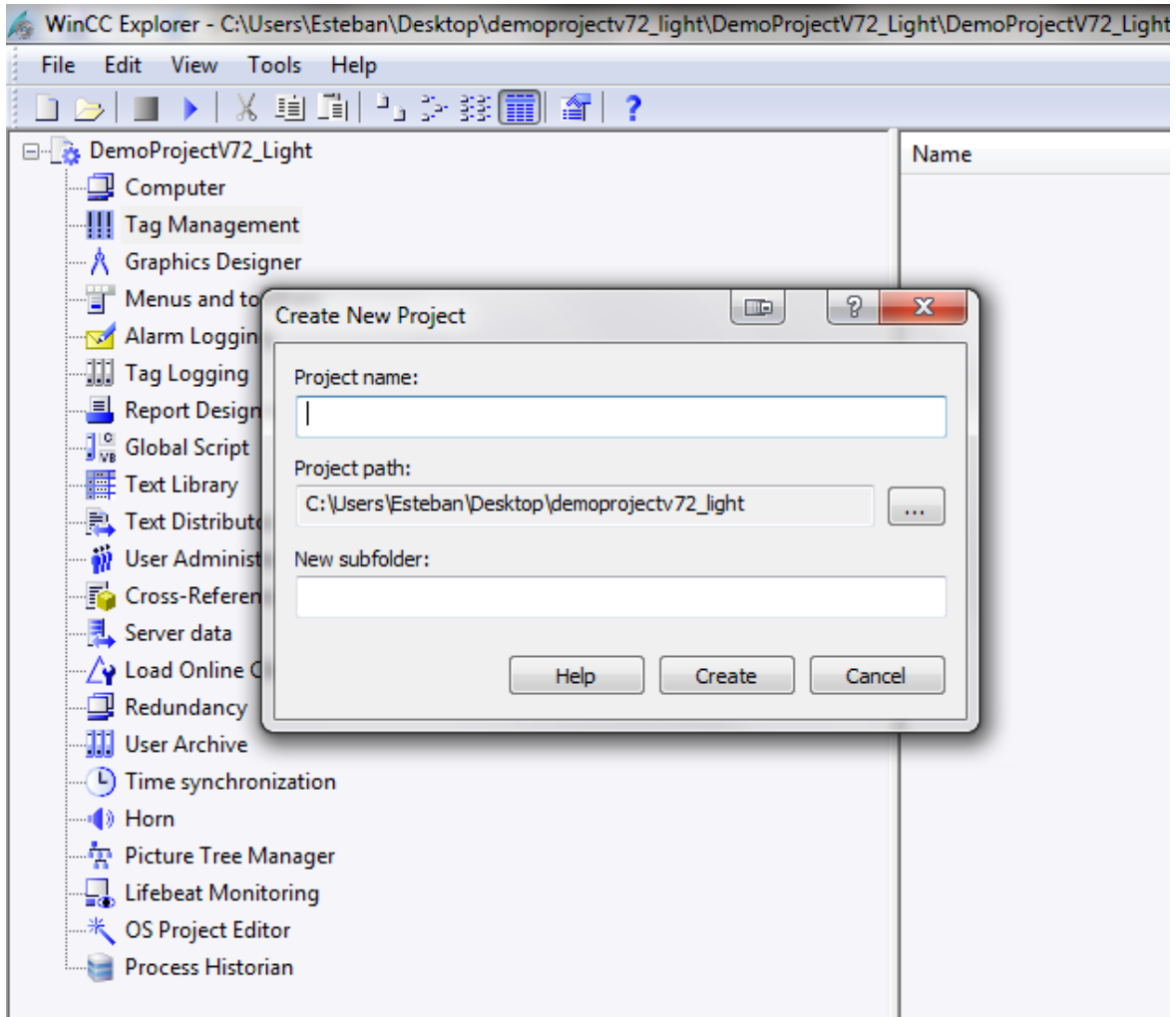
Πεδίο	Περιγραφή
<b>Status</b>	Εμφανίζει το συγκεκριμένο Path του αρχείου
<b>Number of configured tags / Number of power tags</b>	Εμφανίζεται ο αριθμός των εξωτερικών tags μόνο αν είναι επιλεγμένο το tag management στο WinCC explorer
<b>Object(s)</b>	Εμφανίζεται ο αριθμός αντικειμένων του επιλεγμένου editor
<b>CAPS</b>	Εμφανίζεται μόνο αν τα κεφαλαία είναι ενεργοποιημένα

## 2.9 Δημιουργία Project

Στο **toolbar** επιλέγουμε το εικονίδιο



Αφού επιλέξουμε το **Single-User Project** πατάμε OK και ανοίγει το παράθυρο 'Create new project'.

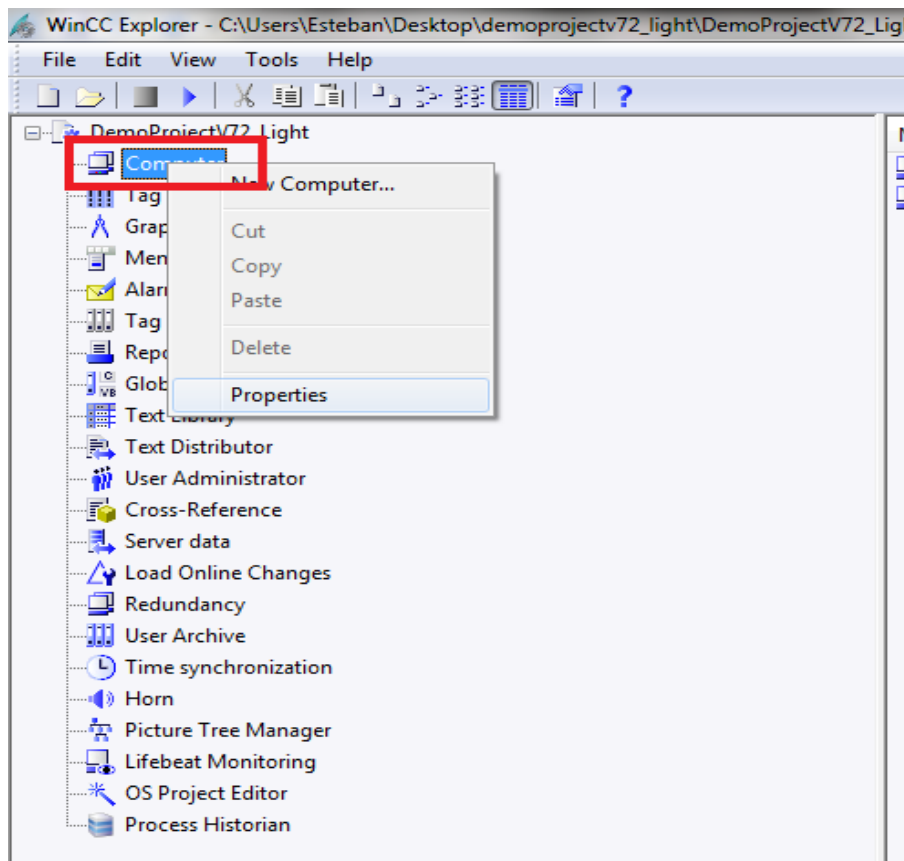


Στο Project name εισάγουμε ένα όνομα για το project, στο Project path επιλέγουμε το path (διαδρομή) στο οποίο θα δημιουργηθεί ο φάκελος για το project μας, και τέλος πατάμε το Create.

## 2.10 Καθορισμός ιδιοτήτων Υπολογιστή

Όταν δημιουργούμε ένα **project** είναι απαραίτητο να καθορίσουμε τις ιδιότητες του υπολογιστή στον οποίο το project θα είναι ενεργό. Σε ένα **multi-user system**, πρέπει να καθοριστούν οι ιδιότητες για το κάθε υπολογιστή ξεχωριστά.

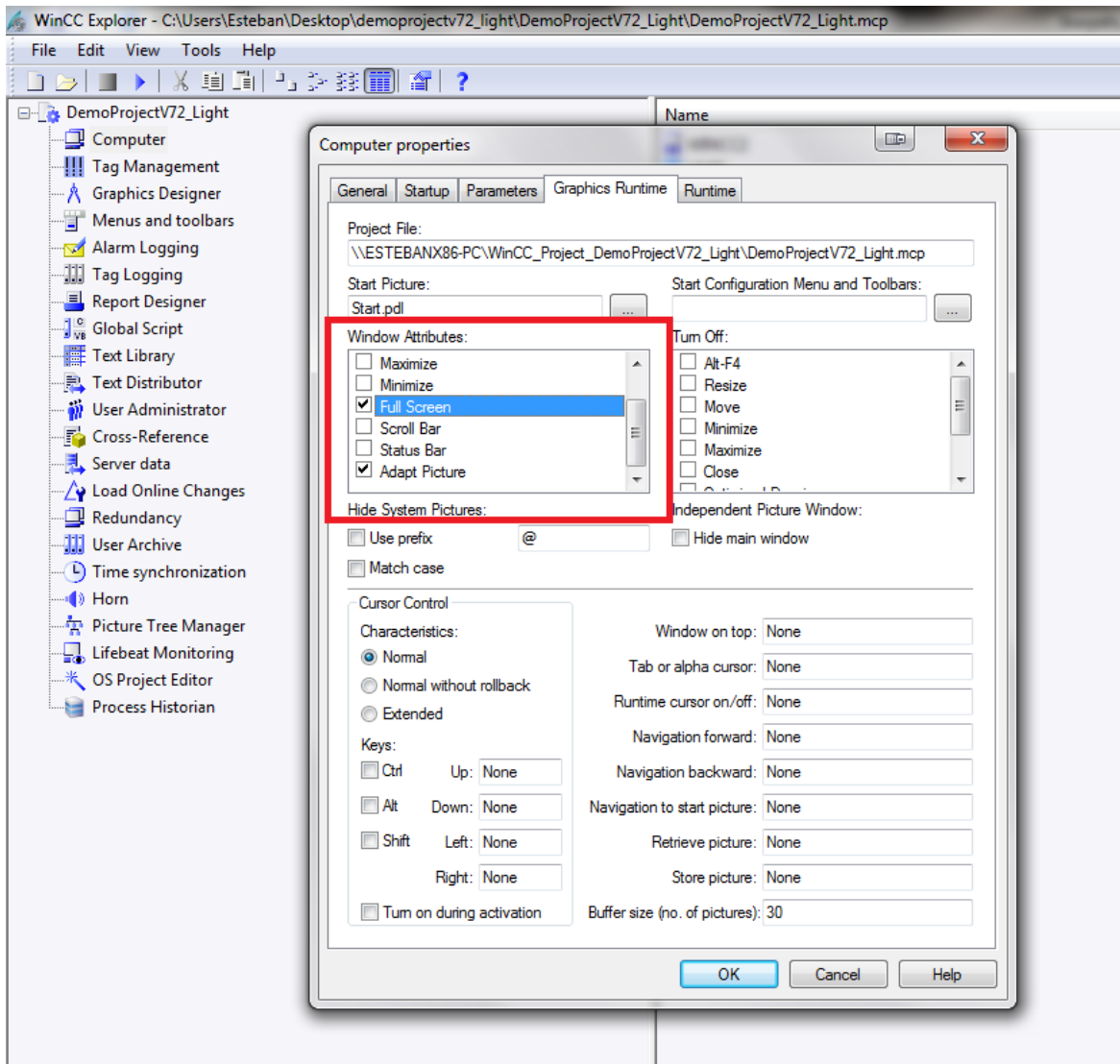
- Επιλέγουμε **Computer** από το **navigation window**.
- Στο **data window** κάνουμε δεξί κλικ στον όνομα του υπολογιστή και επιλέγουμε **Properties**.



- Ελέγχουμε εάν στο **Computer Name** εμφανίζεται το σωστό όνομα υπολογιστή που εμείς θέλουμε.
- Κλικαρούμε την επιλογή '**Use Local Computer Name**' εάν θέλουμε να προσαρμοστεί αυτόματα το όνομα του τοπικού υπολογιστή μας.
- Τέλος επιλέγουμε στο **Computer Type** αν το project μας είναι τύπου **server** ή **client**.











Στη συνέχεια πηγαίνουμε στο tab **Graphics Runtime** το οποίο περικλείει όλες τις ιδιότητες του **WinCC Runtime**, δηλαδή της κατάστασης κατά την οποία «τρέχει» η εφαρμογή μας



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

## 2.11 Καθορισμός Tag Management

Επιλέγοντας από το **navigation window** το **Tag Management** ανοίγει ένα παράθυρο με τις παρακάτω επιλογές

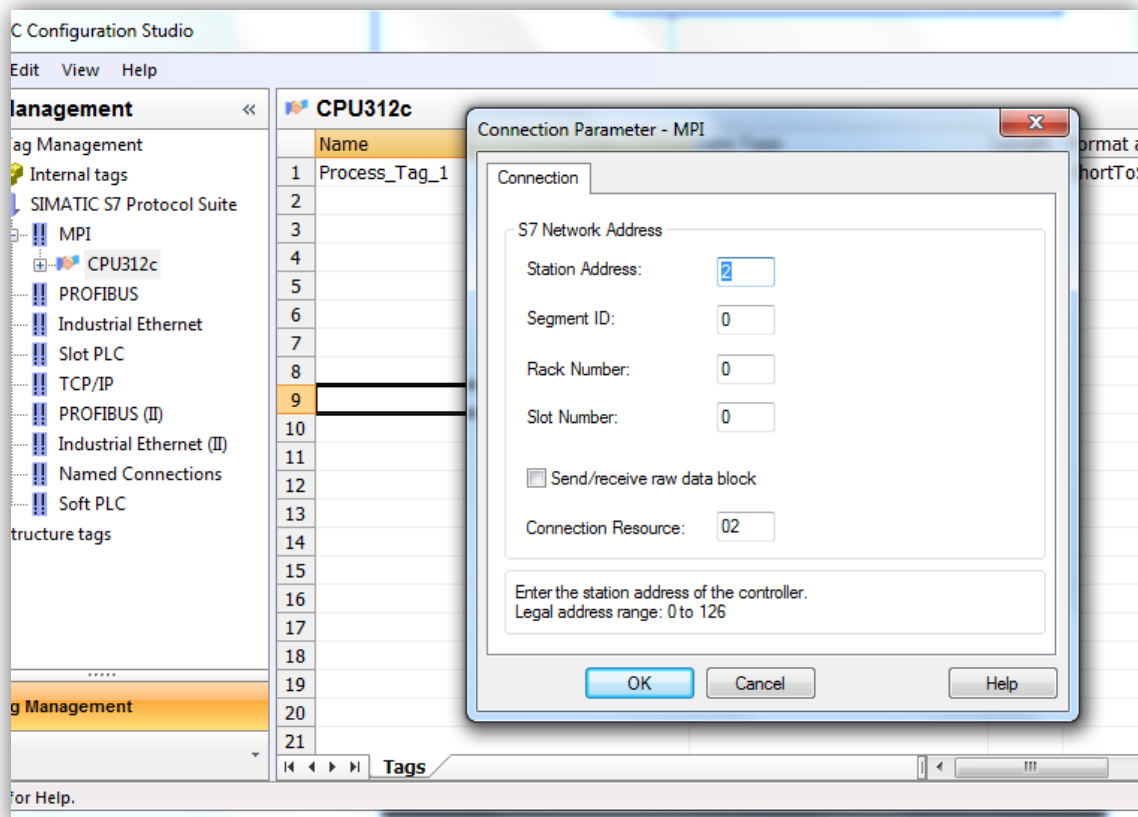
Εικονίδιο	Περιγραφή
	Tag management (διαχείριση μεταβλητών)
	Internal Tags (εσωτερικές μεταβλητές)
	Communication Drivers (οδηγοί επικοινωνίας)
	Channel Unit (διάυλος μονάδας)
	Connection (σύνδεση)
	Tag group (ομάδα μεταβλητών)
	μεταβλητές
	Structure type (διάρθρωση)

## 2.12 Επιλογή και εγκατάσταση ενός driver

Ένα από τα πιο βασικά βήματα που πρέπει να ακολουθήσουμε είναι η εγκατάσταση του τρόπου επικοινωνίας του PLC μας με τον υπολογιστή και το WinCC. Όπως γίνεται εύκολα αντιληπτό, αν δεν εισάγουμε τις σωστές παραμέτρους τότε η επικοινωνία δε θα επιτευχθεί. Επομένως, οφείλουμε να γνωρίζουμε το μοντέλο του PLC και το πρωτόκολλο επικοινωνίας.

Υποθέτουμε πως στη συγκεκριμένη περίπτωση η επικοινωνία μεταξύ Υπολογιστή και PLC επιτεύχθη μέσω της **διεπαφής MPI**.

- Πατάμε πάνω στο **SIMATIC S7 Protocol Suite**, στην συνέχεια στο MPI κάνουμε δεξί-κλικ και επιλέγουμε **New Connection**.
- Με δεξί-κλικ πάνω **New Connection**, ονομάζουμε την επικοινωνία **“CPU312c”** και έπειτα εισάγουμε τις κατάλληλες παραμέτρους, όπως φαίνονται στη εικόνα



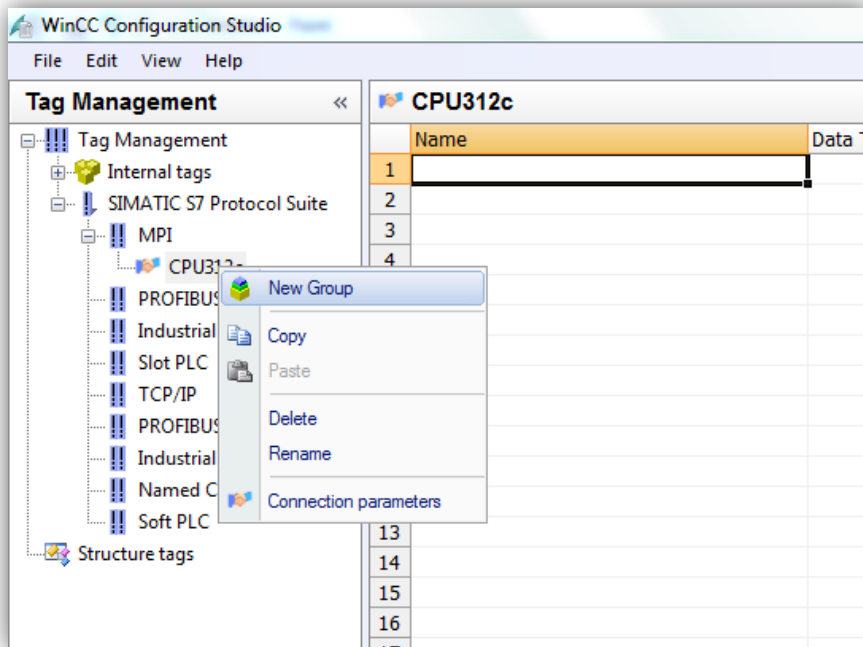
## 2.13 Ορισμός των μεταβλητών (tags)

### 2.13.1 External tags

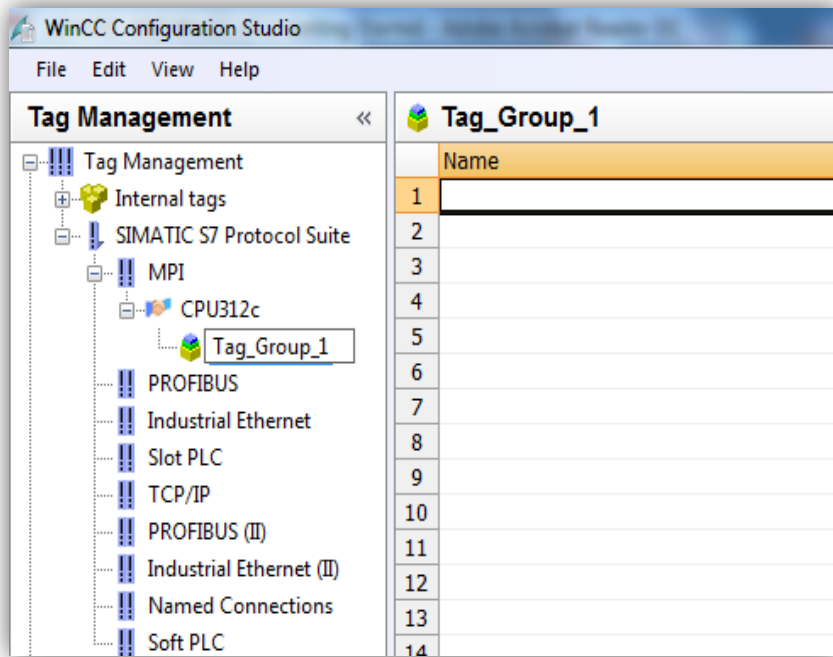
Το επόμενο βήμα είναι ο καθορισμός των **εξωτερικών μεταβλητών (external tags)**. Οι εξωτερικές μεταβλητές χρησιμοποιούνται για να μεταφέρουν τιμές από την περιοχή δεδομένων μνήμης του PLC ή του RTU στο WinCC, όπου γίνεται η επεξεργασία και εποπτεία τους. Ο ορισμός των μεταβλητών είναι σημαντικός για τη σωστή αναπαράσταση των παρατηρούμενων διεργασιών και παίζει σημαντικό ρόλο στη σωστή λειτουργία του SCADA.

Αξίζει να σημειωθεί ότι τόσο οι εσωτερικές όσο και οι εξωτερικές μεταβλητές μπορούν να ομαδοποιηθούν σε ομάδες μεταβλητών, τα **Tag Groups**. Δεν αλλάζει κάτι λειτουργικά με την κίνηση αυτή, απλά μας προσφέρει καλύτερη οργάνωση και ευκολότερη αναζήτηση, κάτι το οποίο έχει βαρύνουσα σημασία σε αρκετά πολύπλοκα και μακροσκελή συστήματα.

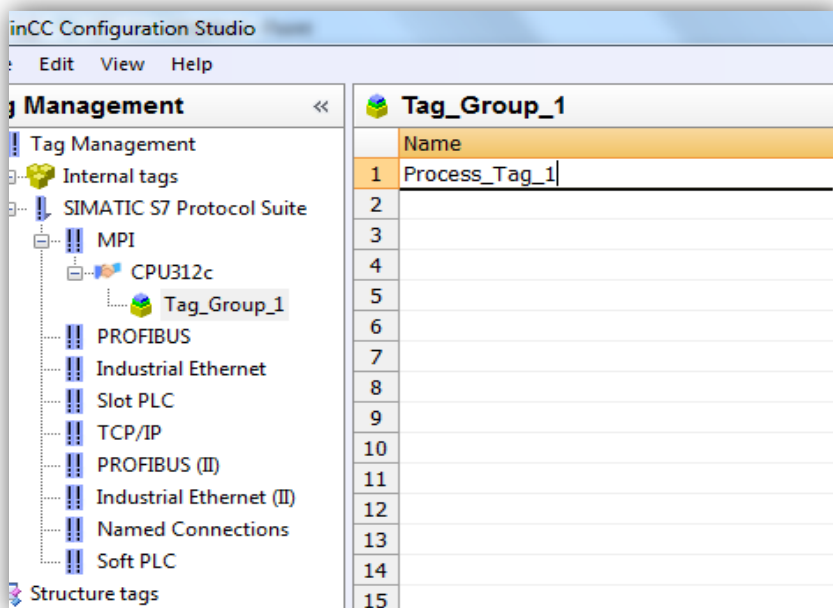
Πατάμε δεξί-κλικ πάνω στη σύνδεση που δημιουργήσαμε προηγουμένως και επιλέγουμε **New Group**.



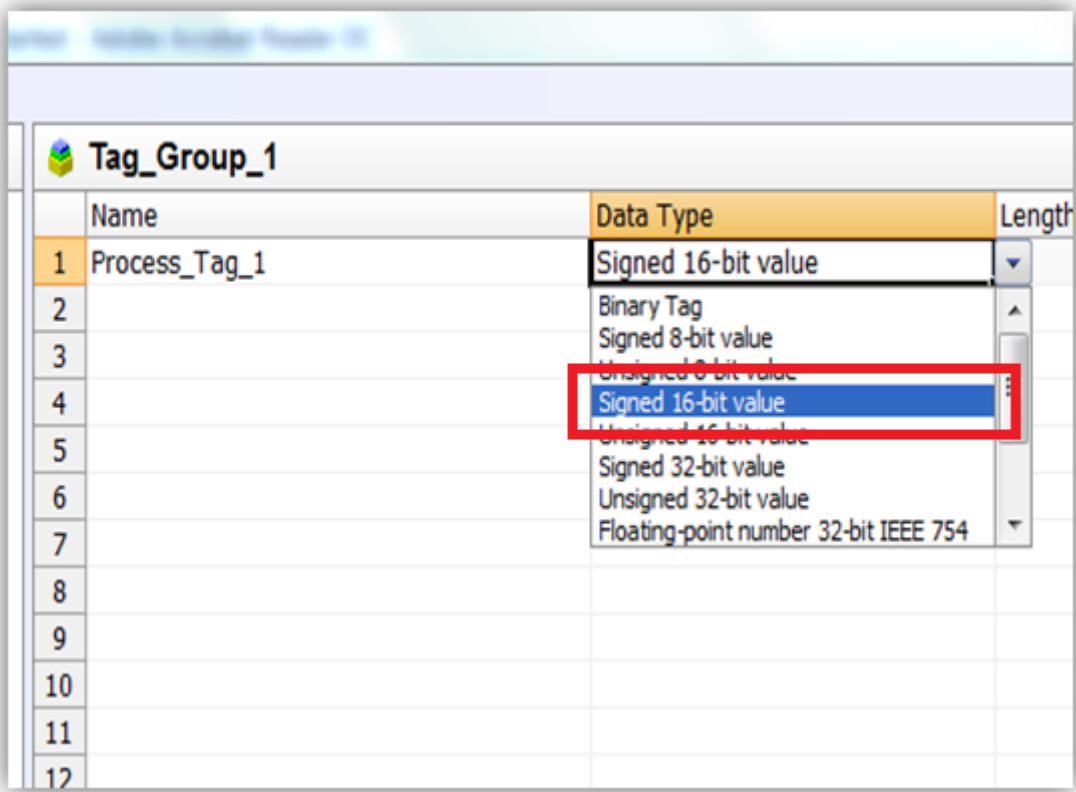
Αλλάζουμε το όνομα σε **Tag\_Group\_1**. Πλέον έχουμε δημιουργήσει το tag group με όνομα **Tag\_Group\_1**. Πατώντας πάνω στο **Tag\_Group\_1** εμφανίζεται δεξιά στην οθόνη το Group.



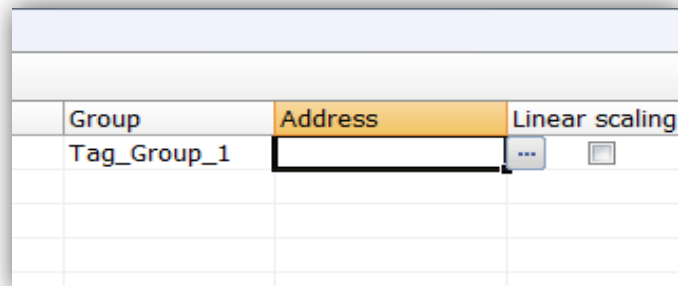
Εισάγουμε το **Process\_Tag\_1** σαν όνομα για την εξωτερική μεταβλητή.



Επιλέγουμε για Data type το **Signed 16-bit value**, και κάνουμε κλικ στη στήλη **Address**



	Name	Data Type	Length
1	Process_Tag_1	Signed 16-bit value	
2		Binary Tag	
3		Signed 8-bit value	
4		Signed 16-bit value	
5		Unsigned 16-bit value	
6		Signed 32-bit value	
7		Unsigned 32-bit value	
8		Floating-point number 32-bit IEEE 754	
9			
10			
11			
12			



Group	Address	Linear scaling
Tag_Group_1		... <input type="checkbox"/>

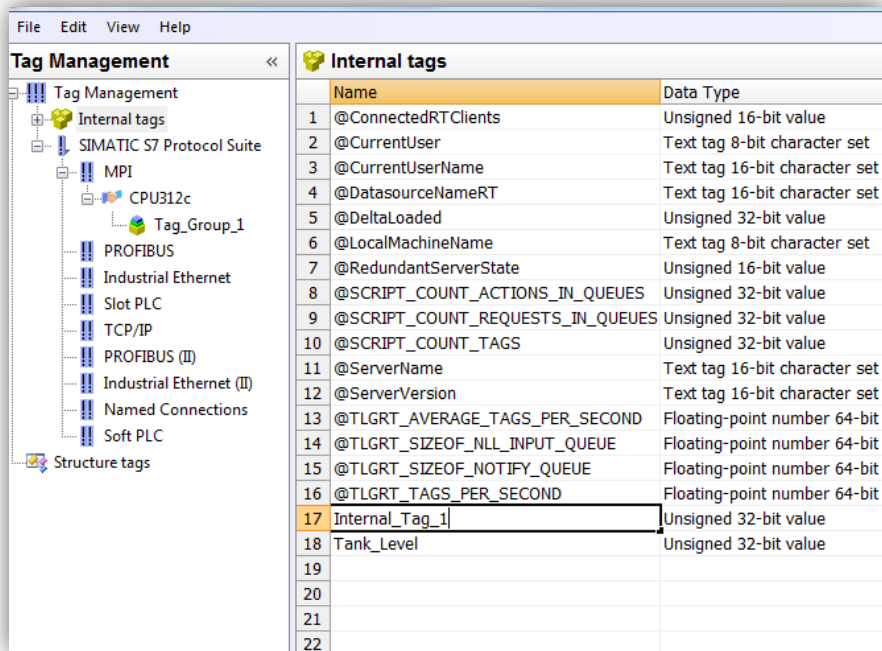
Στο παράθυρο **Address properties**, στο πεδίο **Data** επιλέγουμε το **Bit memory** και πατάμε OK.



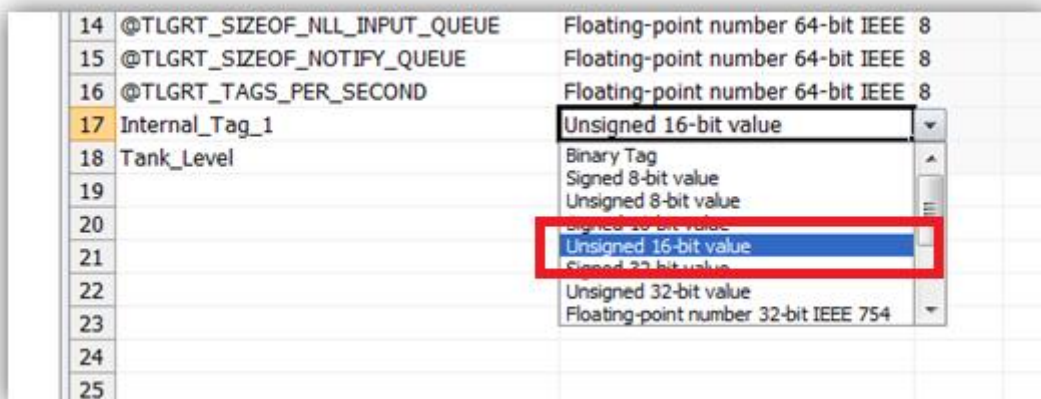
### 2.13.2 Internal tags

Οι εσωτερικές μεταβλητές αφορούν κάποια τμήματα της μνήμης του WinCC όμως, τα οποία χρησιμοποιούνται για εσωτερικά θέματα του λογισμικού και για λειτουργίες προσομοίωσης.

- Επιλέγουμε **Internal Tags**.
- Εισάγουμε την εσωτερική μεταβλητή με όνομα **Internal\_Tag\_1**.

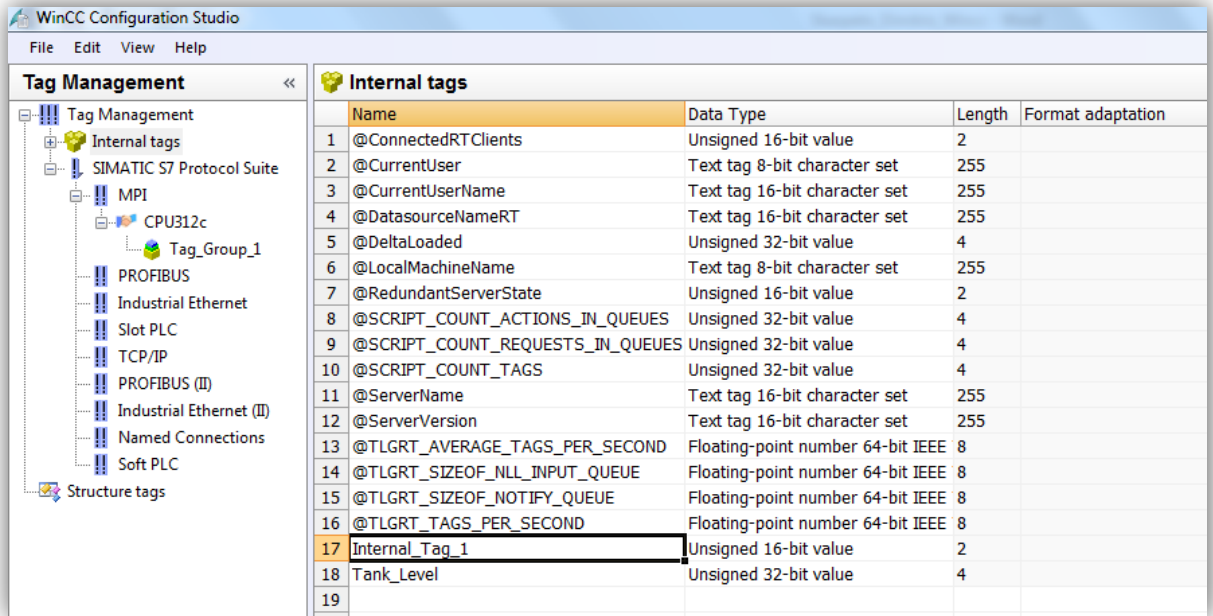


- Αλλάζουμε το Data Type σε **Unsigned 16-bit value**.





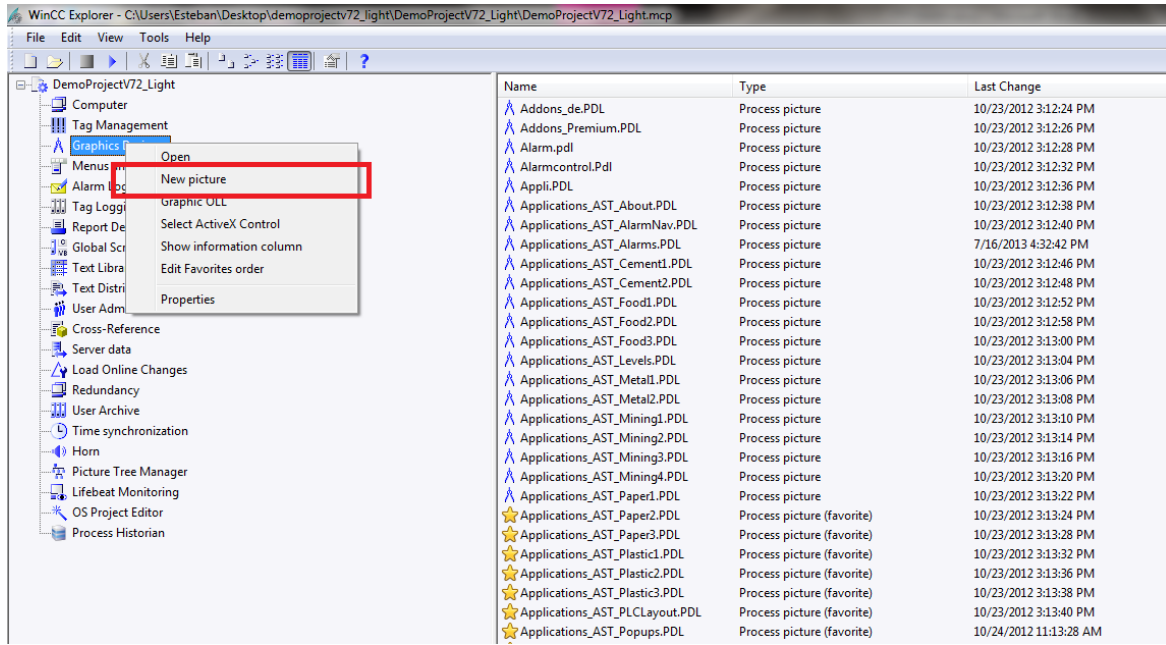
Ολοκληρώσαμε τη δημιουργία μια εσωτερικής μεταβλητής με όνομα **Internal\_Tag\_1**.



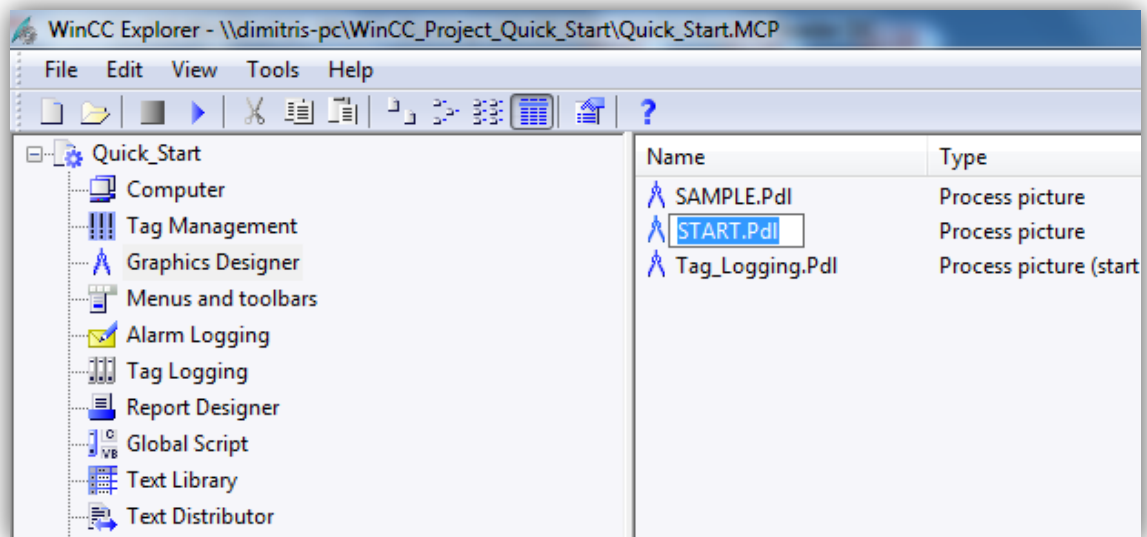
	Name	Data Type	Length	Format adaptation
1	@ConnectedRTClients	Unsigned 16-bit value	2	
2	@CurrentUser	Text tag 8-bit character set	255	
3	@CurrentUserName	Text tag 16-bit character set	255	
4	@DatasourceNameRT	Text tag 16-bit character set	255	
5	@DeltaLoaded	Unsigned 32-bit value	4	
6	@LocalMachineName	Text tag 8-bit character set	255	
7	@RedundantServerState	Unsigned 16-bit value	2	
8	@SCRIPT_COUNT_ACTIONS_IN_QUEUES	Unsigned 32-bit value	4	
9	@SCRIPT_COUNT_REQUESTS_IN_QUEUES	Unsigned 32-bit value	4	
10	@SCRIPT_COUNT_TAGS	Unsigned 32-bit value	4	
11	@ServerName	Text tag 16-bit character set	255	
12	@ServerVersion	Text tag 16-bit character set	255	
13	@TLGRT_AVERAGE_TAGS_PER_SECOND	Floating-point number 64-bit IEEE	8	
14	@TLGRT_SIZEOF_NLL_INPUT_QUEUE	Floating-point number 64-bit IEEE	8	
15	@TLGRT_SIZEOF_NOTIFY_QUEUE	Floating-point number 64-bit IEEE	8	
16	@TLGRT_TAGS_PER_SECOND	Floating-point number 64-bit IEEE	8	
17	Internal_Tag_1	Unsigned 16-bit value	2	
18	Tank_Level	Unsigned 32-bit value	4	
19				

## 2.14 Graphics Designer

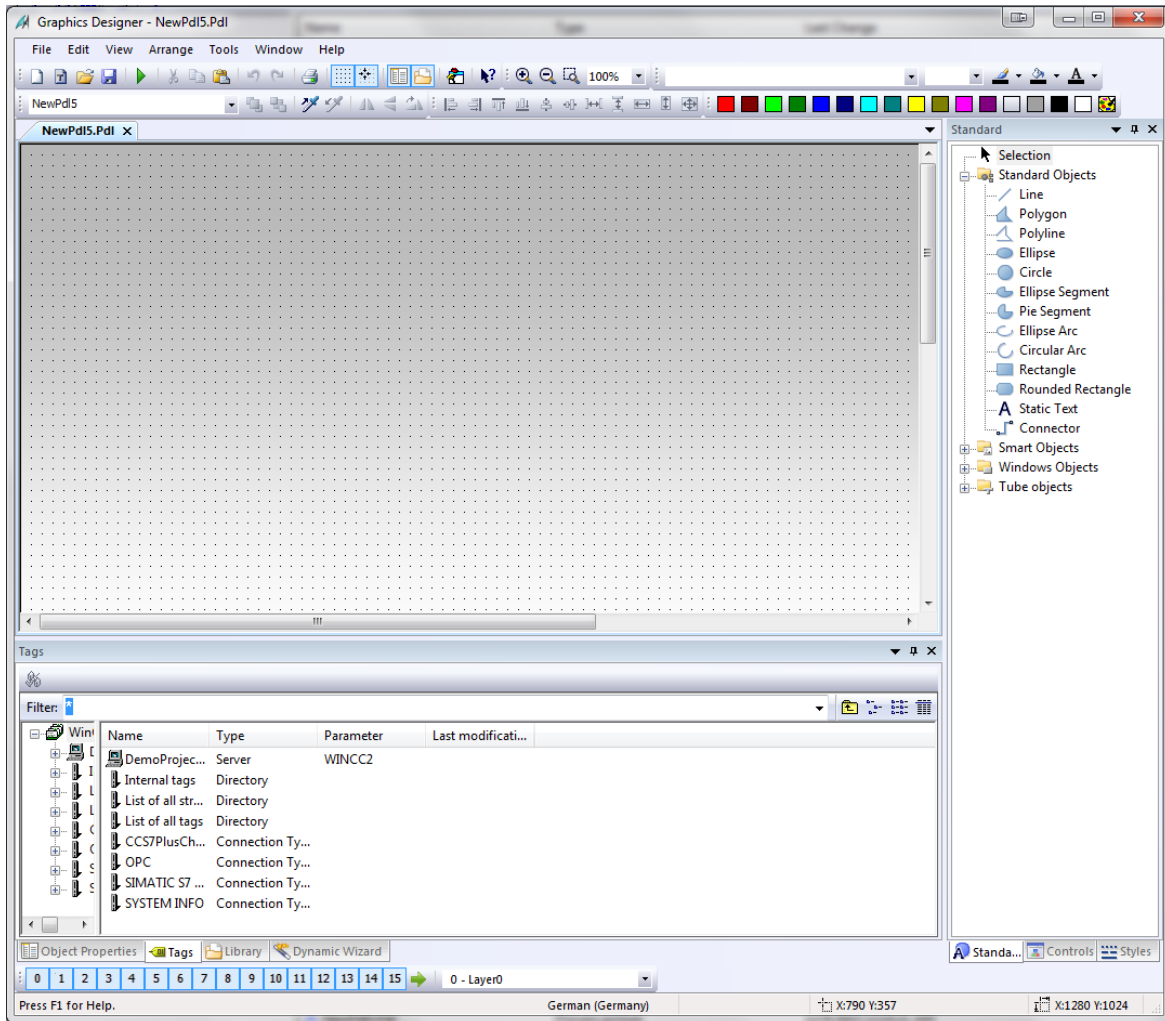
Δημιουργούμε μια καινούργια εικόνα. Δεξί-κλικ πάνω στο Graphics Designer και επιλέγουμε **New picture**



Μετονομάζουμε τη νέα εικόνα σε (π.χ. **START.Pdl**).



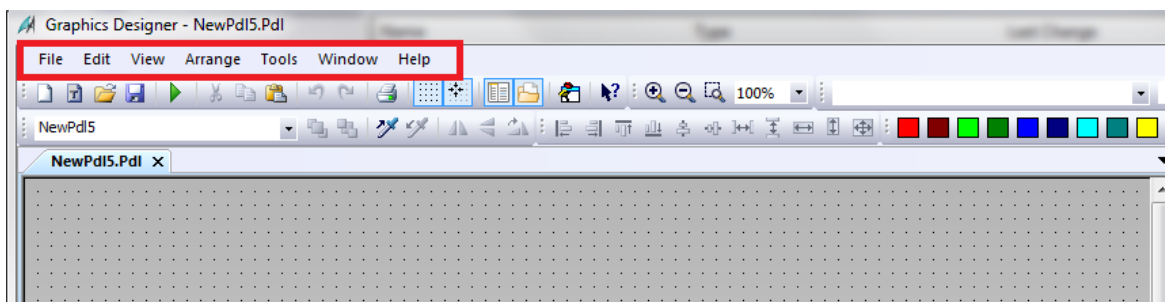
Πατώντας διπλό κλικ πάνω στην εικόνα εμφανίζεται το παρακάτω **graphical layout**



Σχήμα 1.19: Graphics Designer

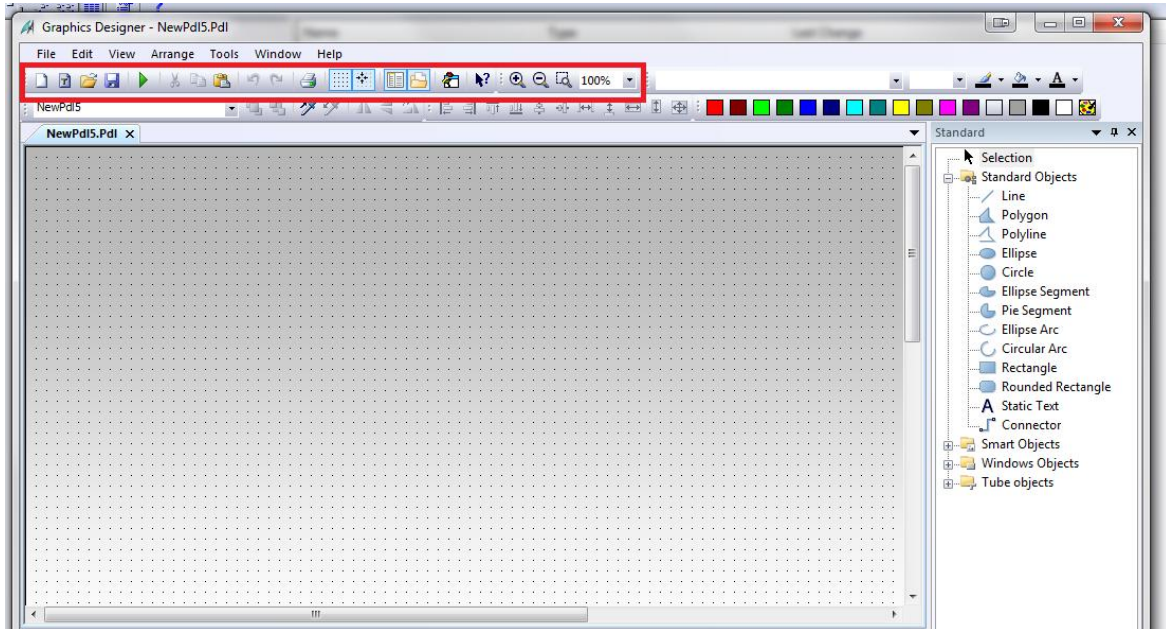
### 2.14.1 Menu Bar

Περιέχει όλες τις μεταβλητές του μενού για τον Graphics Designer.



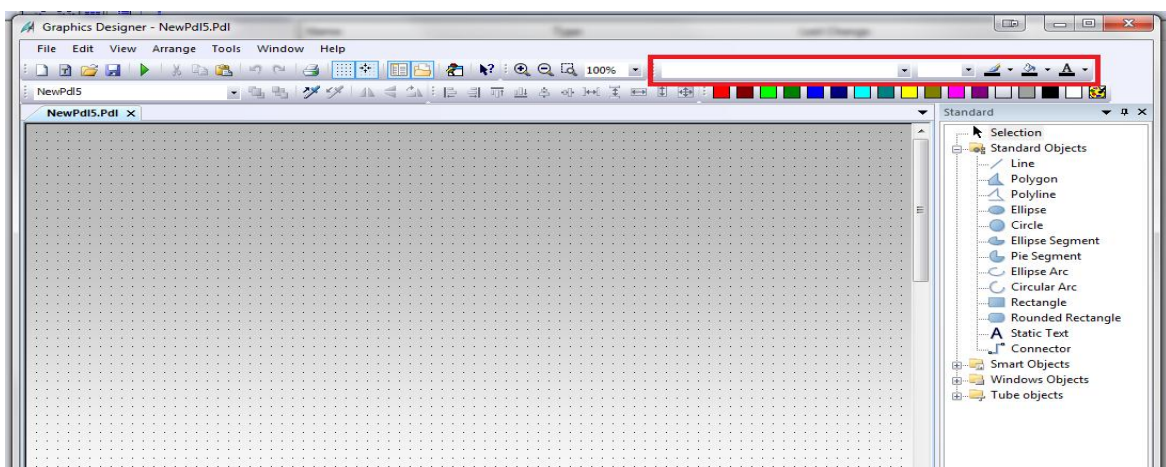
## 2.14.2 Toolbar

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



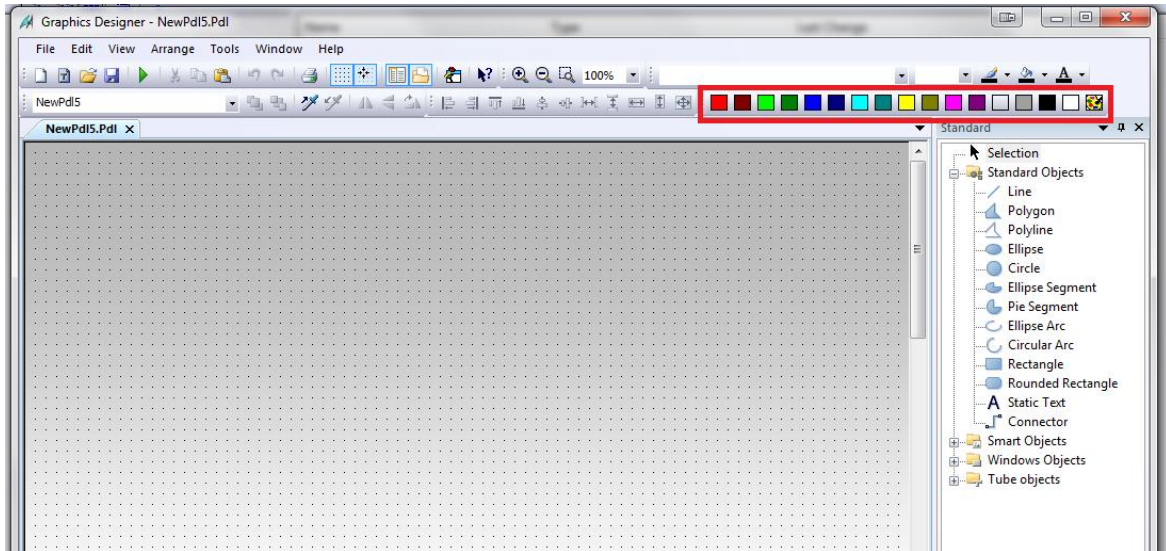
## 2.14.3 Font Palette

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



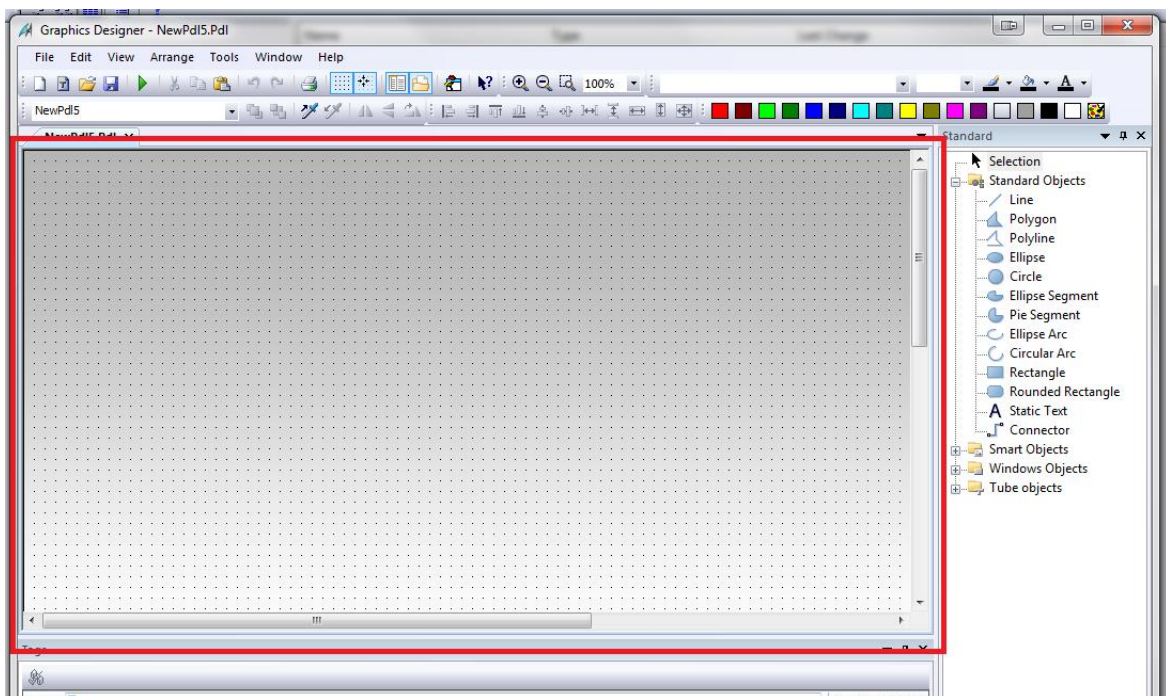
### 2.14.4 Color Palette

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



### 2.14.5 Workspace

Στη περιοχή αυτή τοποθετούνται όλα τα αντικείμενα που θα εμφανίζονται στο Layout της εφαρμογής μας.

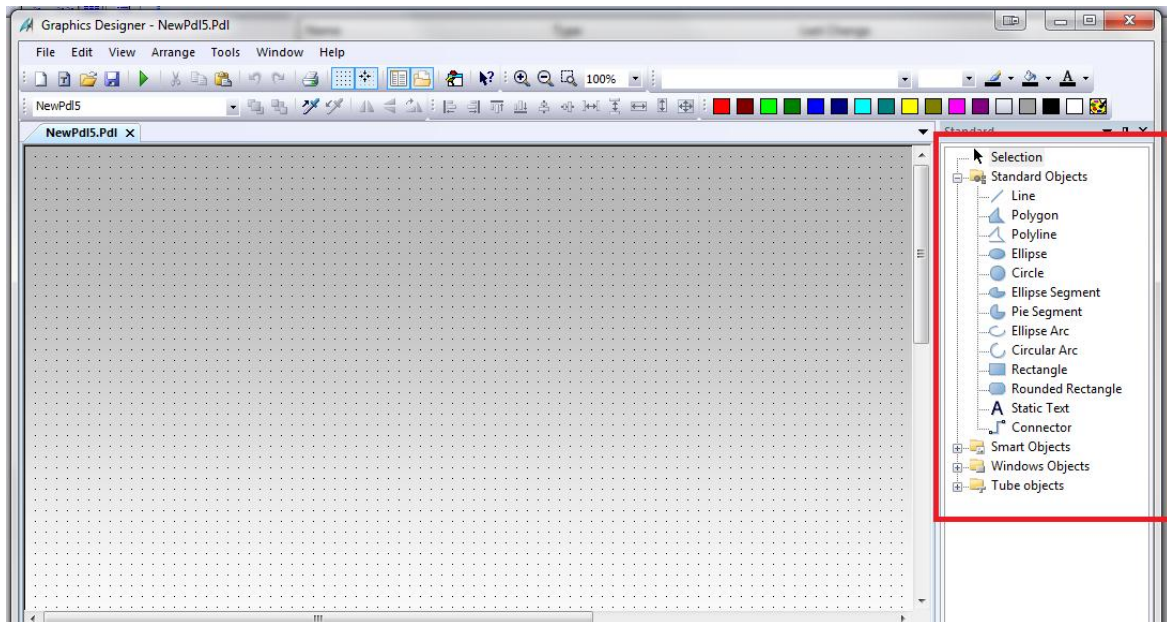




### 2.14.6 Object Palette

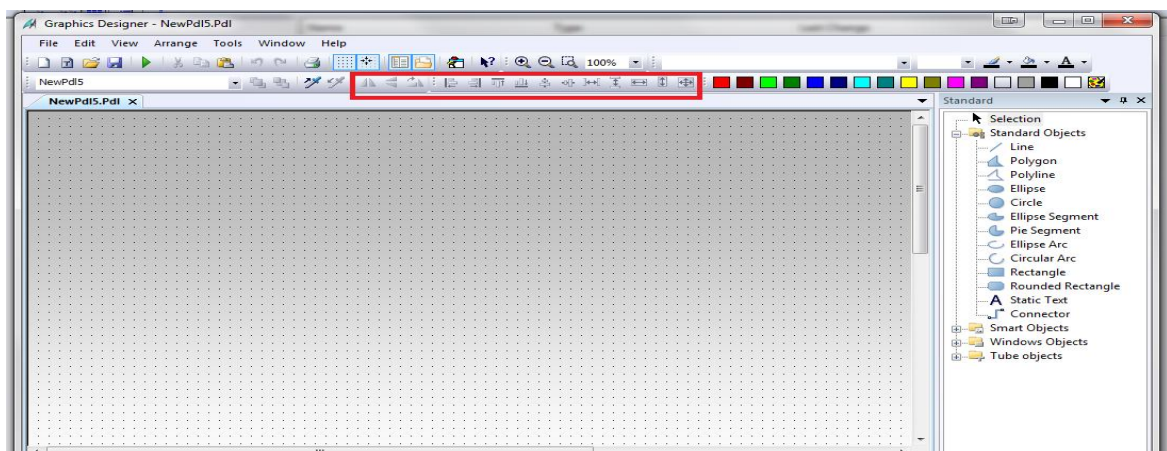
Περιέχει όλα τα κύρια σχήματα: **Standard Objects**, (όπως πολύγωνο, έλλειψη, παραλληλόγραμμο), **Smart Objects** (OLE Control, OLE Element, I/O Fields) και **Windows Objects** (Button, Check Box).

Γενικά περιέχει βιβλιοθήκη βασικών αντικειμένων τα οποία επιτελούν βασικές λειτουργίες σε μία εφαρμογή.



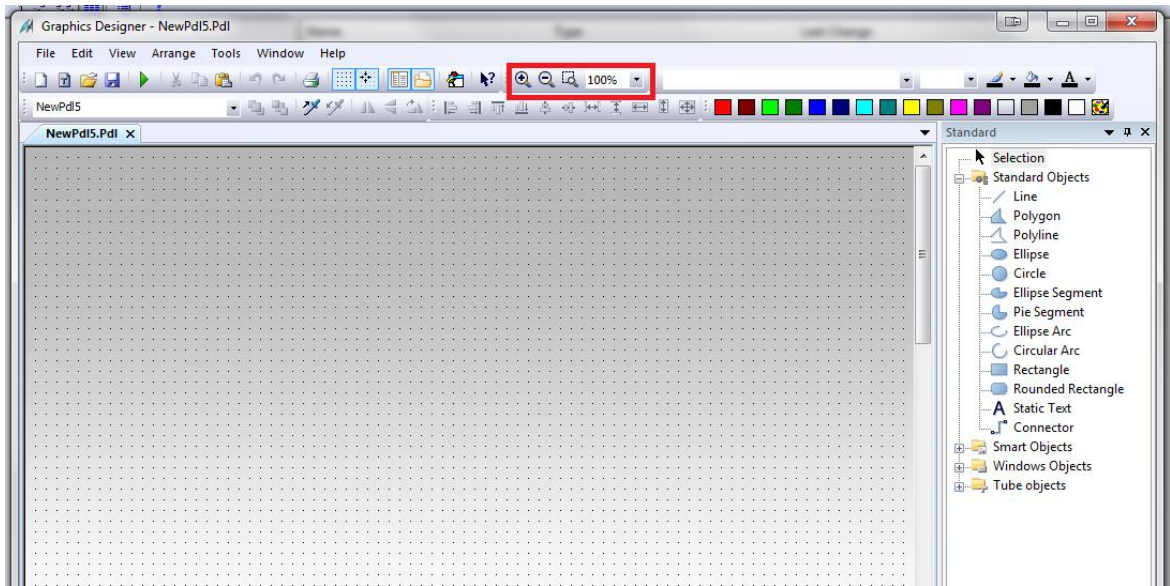
### 2.14.7 Alignment Palette

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



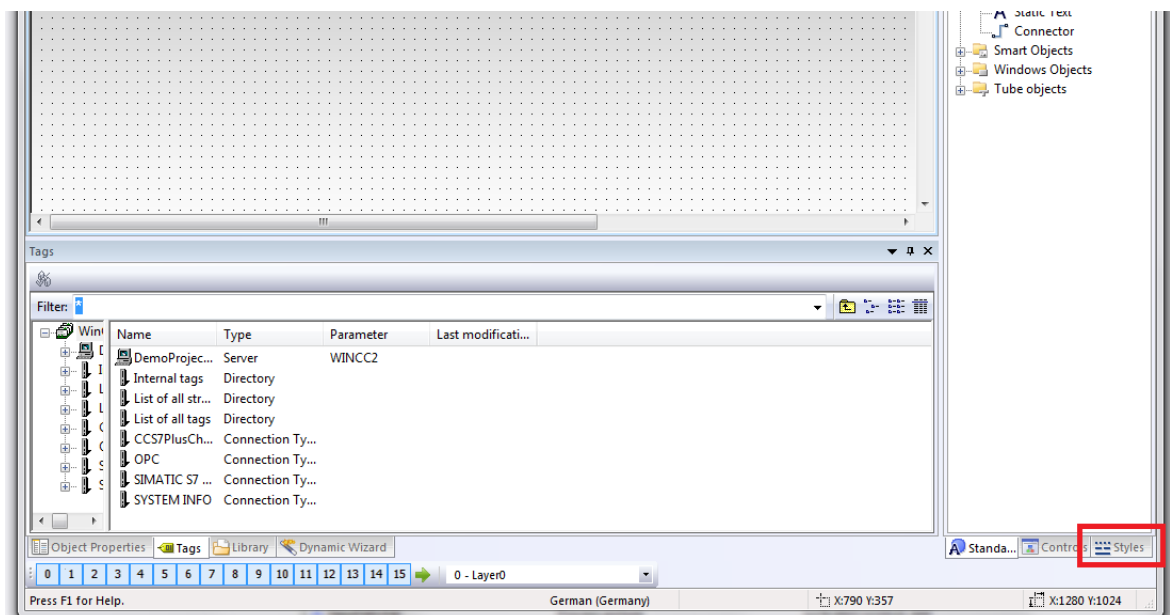
### 2.14.8 Zoom Palette

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



### 2.14.9 Style Palette

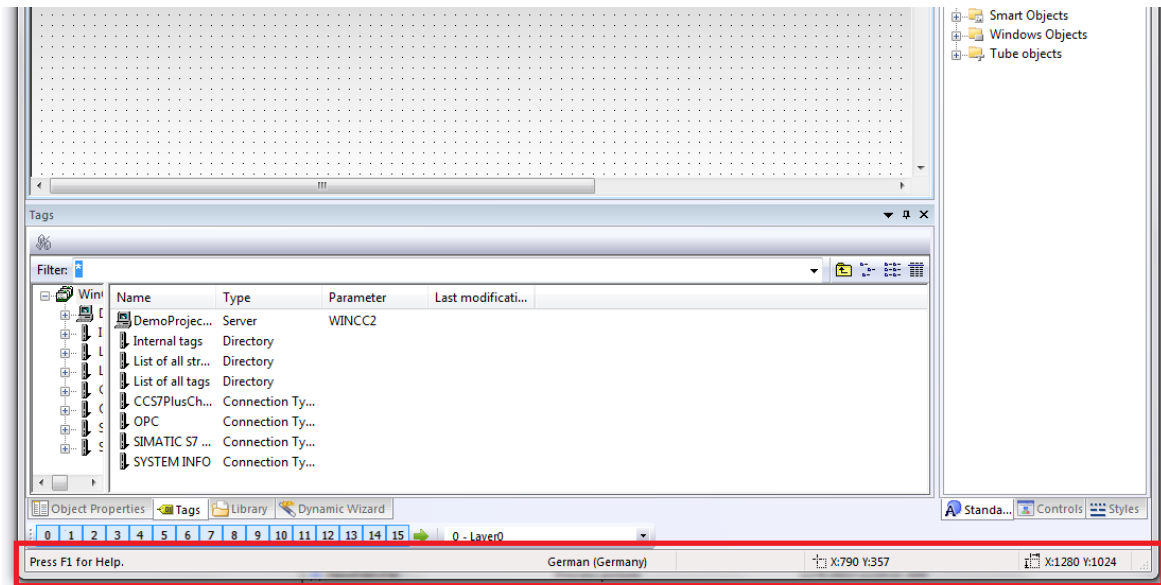
Αλλάζει την εμφάνιση του επιλεγμένου αντικειμένου. Ανάλογα με τον τύπο του αντικειμένου, μπορεί να αλλαχθεί ο **τύπος της γραμμής ή του πλαισίου**, το **πλάτος της γραμμής ή του πλαισίου**, το **"στυλ" (είδος)** του τέλους της γραμμής ή το **"γέμισμα" (fill pattern)**.



### 2.14.10 Status Bar

Βρίσκεται στο κάτω μέρος της οθόνης. Εμφανίζει διάφορες πληροφορίες για τη θέση των επιλεγμένων αντικειμένων.

Μία πολύ σημαντική παράμετρος την οποία παρέχει ο σχεδιαστής είναι η δυναμική διασύνδεση ενός ή περισσοτέρων αντικειμένων, ώστε να μεταβάλλονται οι ιδιότητές τους, σύμφωνα με κάποια μεταβλητή ή να αναπαριστούν τιμές διαφόρων μεταβλητών.





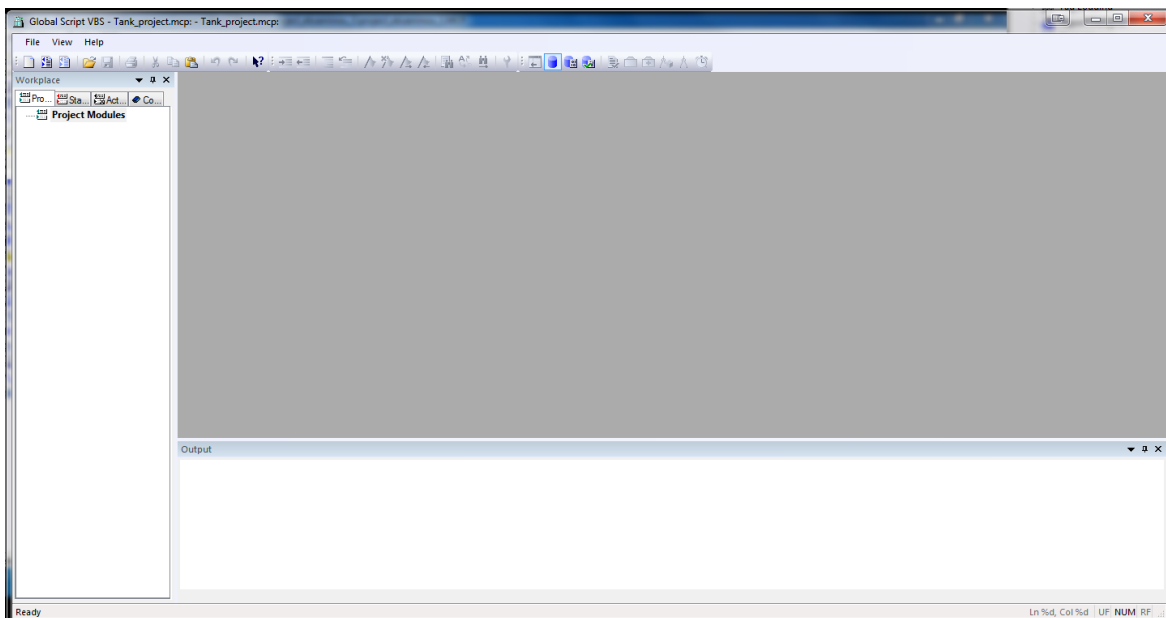
## 2.15 Global Scripts

### 2.15.1 VBS

Με τη Visual Basic Script

- Πατάμε Global Script από το navigation window δεξι-κλικ στο **VBS-Editor** και στη συνέχεια Open.

Ανοίγει το παρακάτω παράθυρο



Σχήμα 1.20: VBS Editor

Ας δούμε παρακάτω μερικά παραδείγματα με χρήση της γλώσσας προγραμματισμού Visual Basic

- Κατά τη διάρκεια εκτέλεσης της εφαρμογής , με το πάτημα του αριστερού-κλικ του ποντικιού η ακτίνα θα παίρνει τη τιμή 20.

```
'VBS121  
  
Dim objCircle  
  
Set objCircle= ScreenItems("Circle1")  
  
objCircle.Radius = 20
```

- Το παρακάτω κομμάτι κώδικα ορίζει το χρώμα του παραθύρου ScreenWindow1 σε μπλε.

```
'VBS122  
  
Dim objScreen  
  
Set objScreen = HMIRuntime.Screens("ScreenWindow1")  
  
objScreen.FillStyle = 131075  
  
objScreen.FillColor = RGB(0, 0, 255)
```

- Το παρακάτω κομμάτι κώδικα μετατρέπει μια εικόνα σε κουμπί.

```
'VBS125  
  
HMIRuntime.BaseScreenName = "Serverprefix::New screen"
```

- Η εικόνα test.pdl εμφανίζεται σε μια δεύτερη εικόνα τη ScreenWindow.pdl.

```
'VBS126  
  
Dim objScrWindow  
  
Set objScrWindow = ScreenItems("ScreenWindow")  
  
objScrWindow.ScreenName = "test"
```

- Στο παρακάτω παράδειγμα δίνεται στο Tag1 η τιμή 6

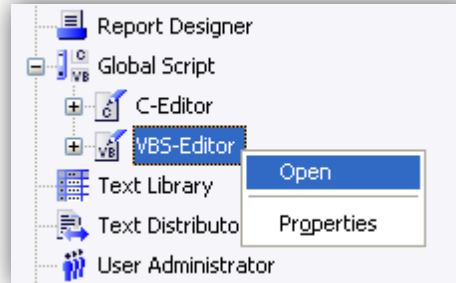
```
'VBS128  
  
HMIRuntime.Tags("Tag1").Write 6
```

- Η τιμή της μεταβλητής Tag1 διαβάζεται και εμφανίζεται.

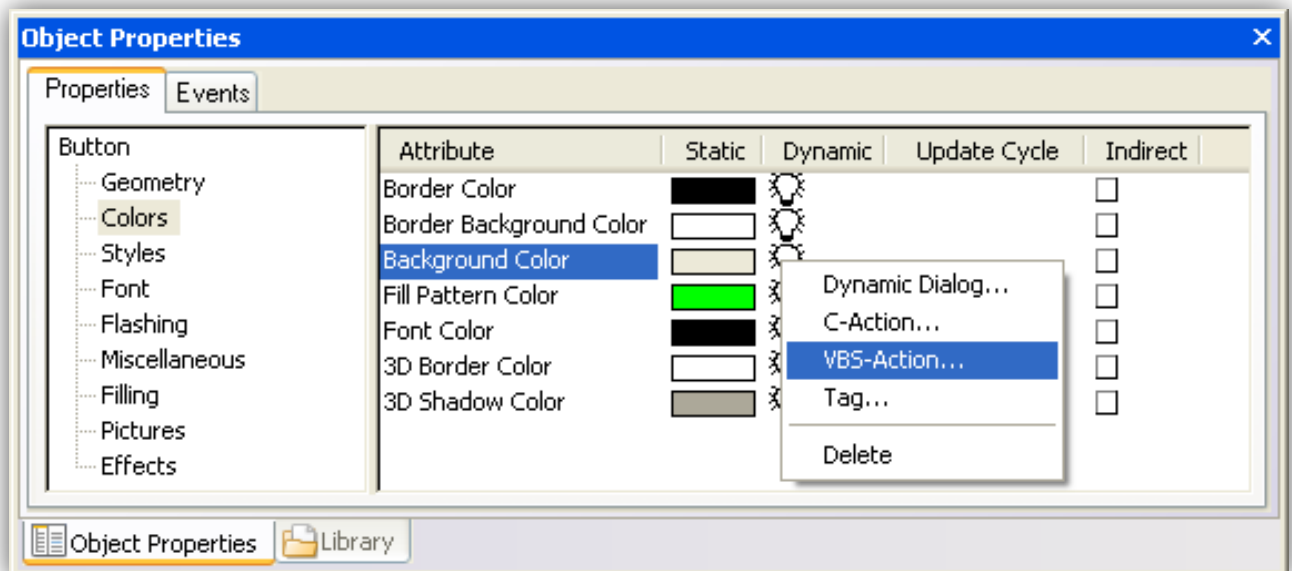
```
'VBS134  
  
HMIRuntime.Trace "Value: " & HMIRuntime.Tags("Tag1").Read &  
vbCrLf
```

Οι τρόποι για να γράψει κάποιος κώδικα σε VBS είναι δύο.

- Είναι ο τρόπος που προαναφέρθηκε.



- Μέσω του **Graphics Designer** επιλέγοντας το Properties ή Events



### 2.15.2 ANSI-C

Ας δούμε παρακάτω μερικά παραδείγματα με χρήση της γλώσσας προγραμματισμού Visual Basic

- Δήλωση ενός Tag σε γλώσσα C

```
int a; //The tag a is defined as an integer  
void dummy() //Function name  
{  
. //Function code  
}
```

- Χρήση Function

```
void dummy() //Function name  
{  
extern int a; //External declaration of the tag a  
. //Function code  
}
```

- Χρήση dll βιβλιοθηκών

```
#pragma code("<Name>.dll")  
  
<Type of returned value> <Function_name 1>(...);  
  
<Type of returned value> <Function_name2>(...);  
  
.  
.  
.  
  
<Type of returned value> <Function_name n>(...);  
  
#pragma code()
```

- Οι συναρτήσεις *function\_name 1* ως *function\_name n* της βιβλιοθήκης *Name.dll* δηλώθηκαν και μπορούν να κληθούν από την αντίστοιχη συνάρτηση.

```
#pragma code("kernel32.dll")  
  
VOID GetLocalTime(LPSYSTEMTIME lpSystemTime);  
  
#pragma code()  
  
SYSTEMTIME st;  
  
GetLocalTime(&st);
```

# Κεφάλαιο 3

Ανάπτυξη Εφαρμογών με το  
WinCC

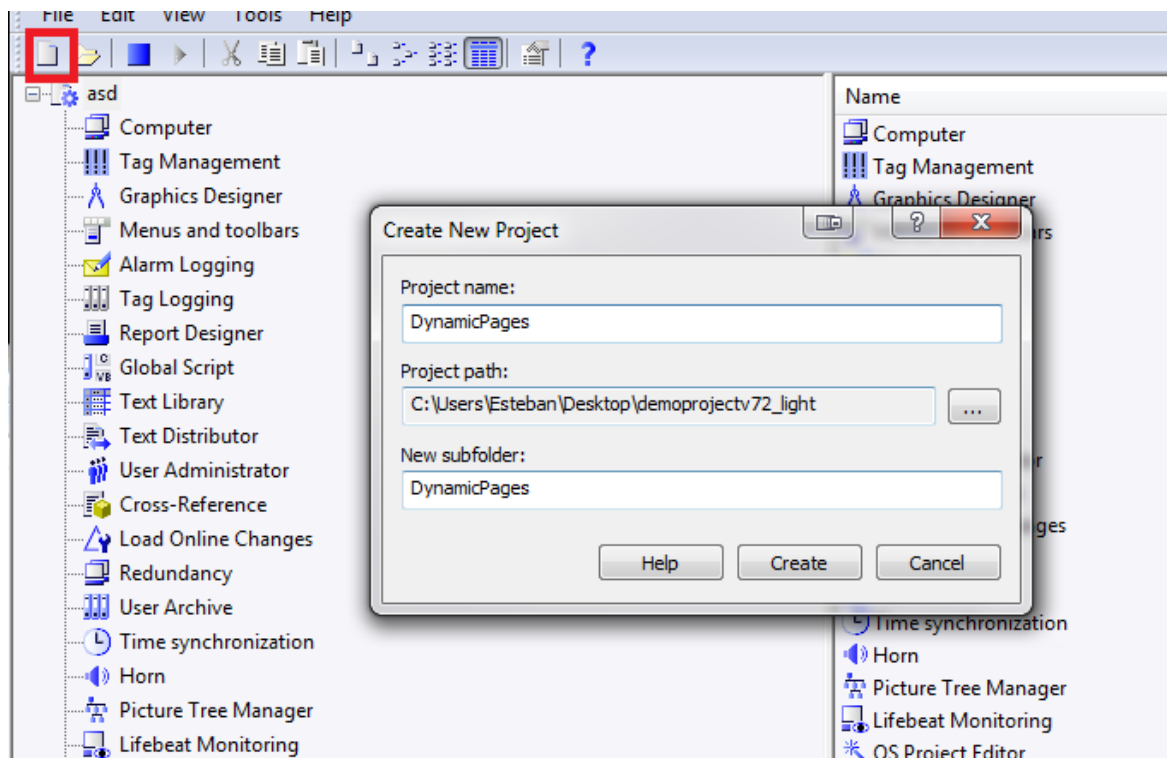
### 3.1 Σχεδιασμός δυναμικών σελίδων

Στη συνέχεια ακολουθεί ένα παράδειγμα σχεδιασμού δυναμικών σελίδων, με σκοπό την εξοικείωση του χρήστη με τις βασικές λειτουργίες του Graphics Designer στο WinCC.

Θα δημιουργήσουμε σελίδες και θα τις διαμορφώσουμε δυναμικά έτσι ώστε ο χρήστης να μπορεί να μεταβεί από την μια στην άλλη με ένα κλικ, μέσα από ένα κεντρικό μενού.

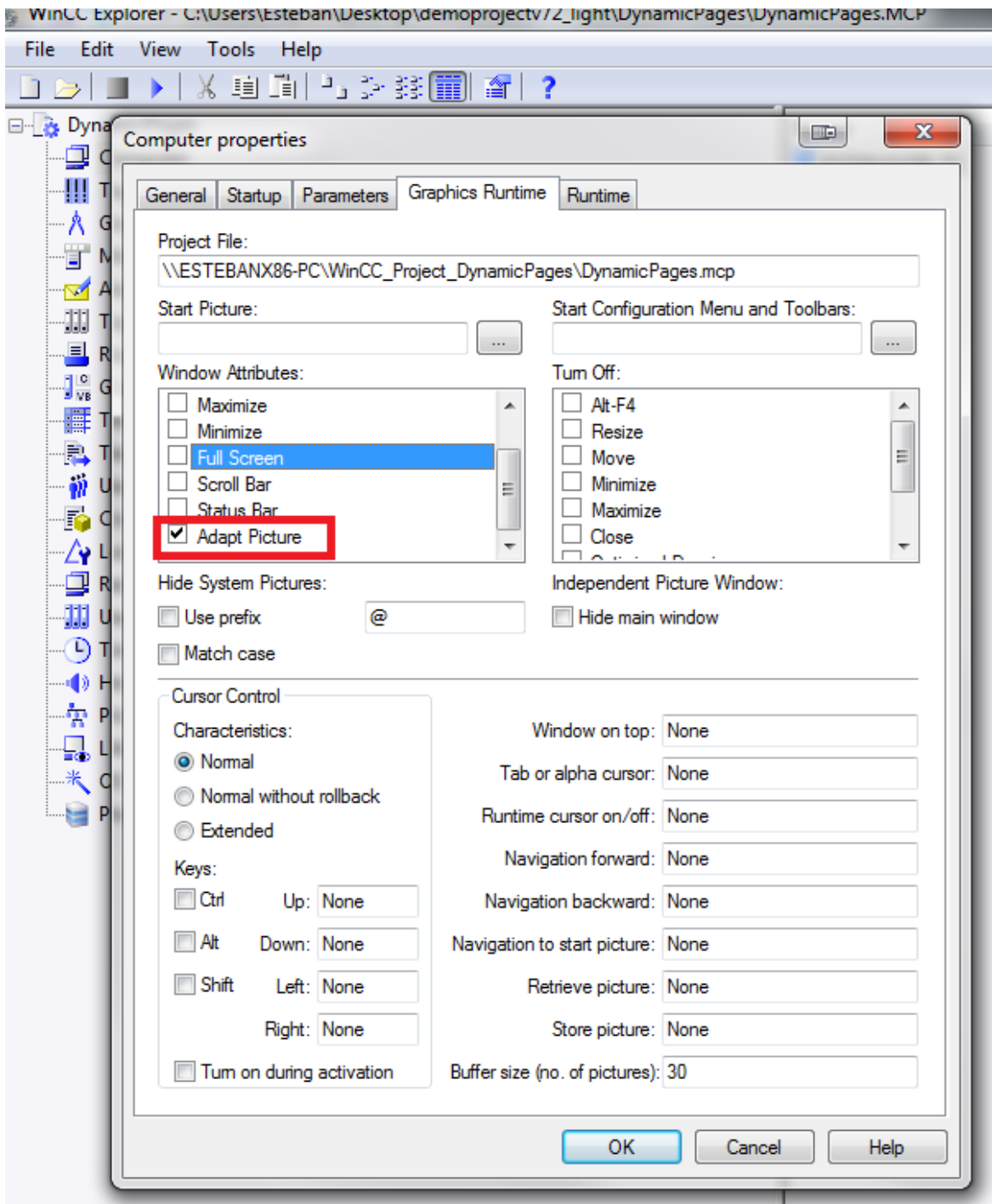
#### 3.1.1 Δημιουργία του project

Ανοίγουμε τον WinCC explorer και πατάμε στο κουμπί δημιουργίας νέου project όπως αναφέρθηκε σε προηγούμενο κεφάλαιο. Εισάγουμε το όνομα του project και πατάμε create.

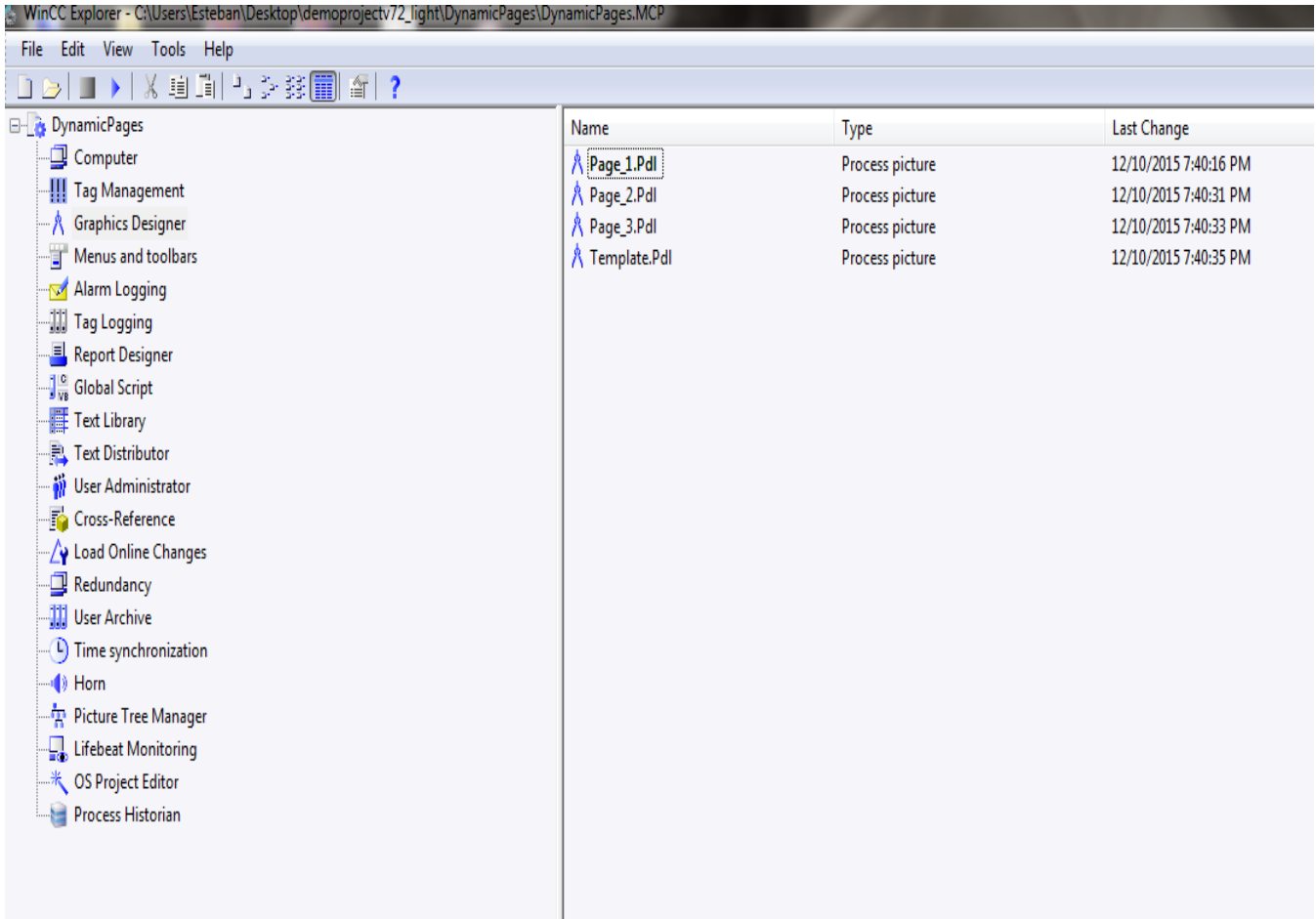




Πατάμε διπλό κλικ στο **computer**, επιλέγουμε τον υπολογιστή μας και πατάμε διπλό κλικ στην καρτέλα **graphics runtime**, όπου ενεργοποιούμε την επιλογή **adapt picture**.



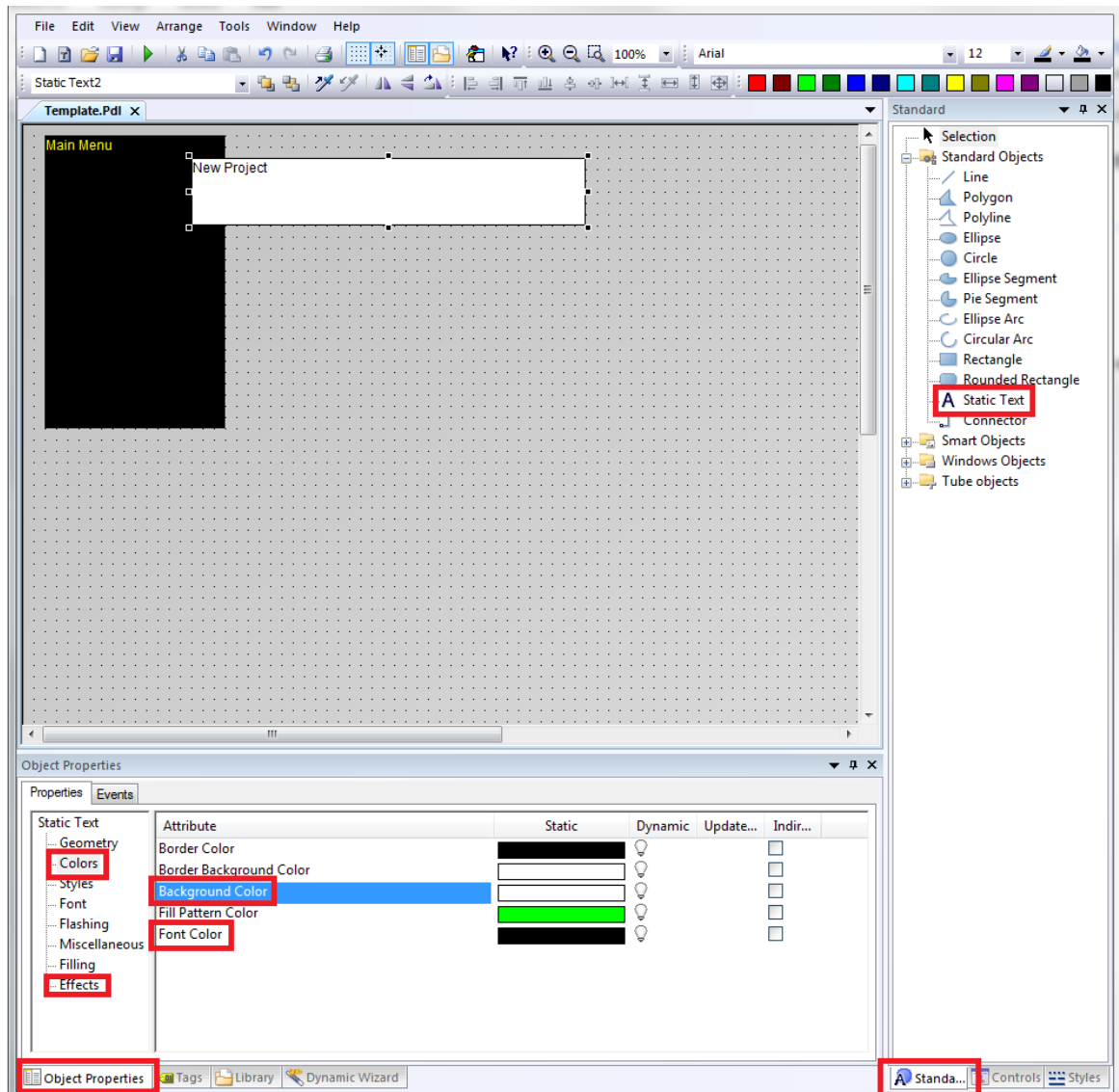
Στη συνέχεια πατάμε κάνουμε δεξί κλικ στον graphics designer στο navigation window και επιλέγουμε **new picture**. Δημιουργούμε 4 νέες εικόνες, και τις μετονομάζουμε σε **Page 1**, **Page 2**, **Page 3** και **Template** όπως φαίνεται στο παρακάτω σχήμα.



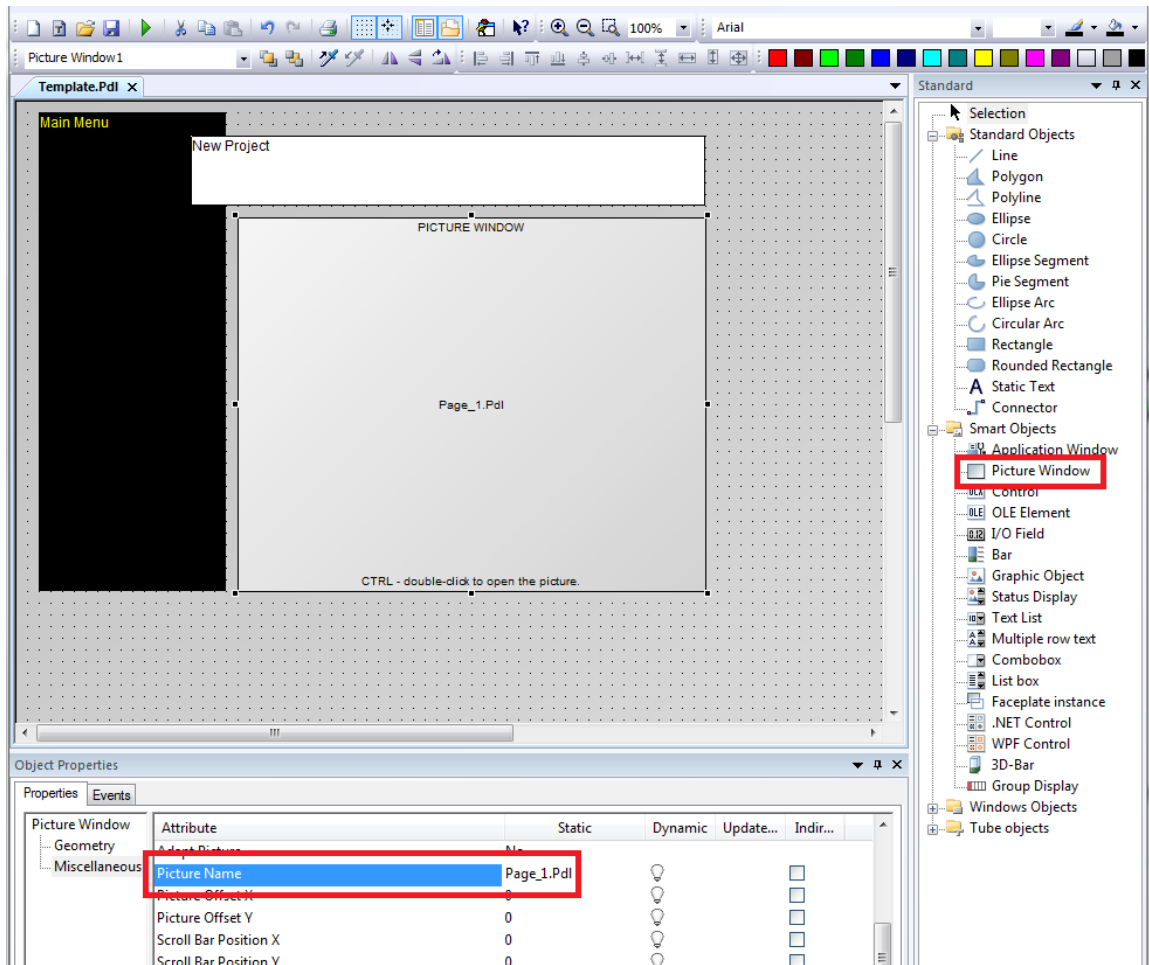
Πατάμε δεξί κλικ στην εικόνα **Template**, και επιλέγουμε **open picture** για να μεταβούμε στον graphics designer.

### 3.1.2 Δημιουργία αρχικού μενού πλοήγησης

Επιλέγουμε **static text** στο object palette και δημιουργούμε δυο πεδία. Μετονομάζουμε σε main menu και new project. Επιλέγοντας το κάθε πεδίο, κάτω αριστερά στην οθόνη μας στην καρτέλα object **properties** επιλέγουμε **effects**, **global color scheme** -> **No**. Στην επιλογή Colors αλλάζουμε το **background color** και το **font color** σε χρώματα της αρέσκειάς μας.



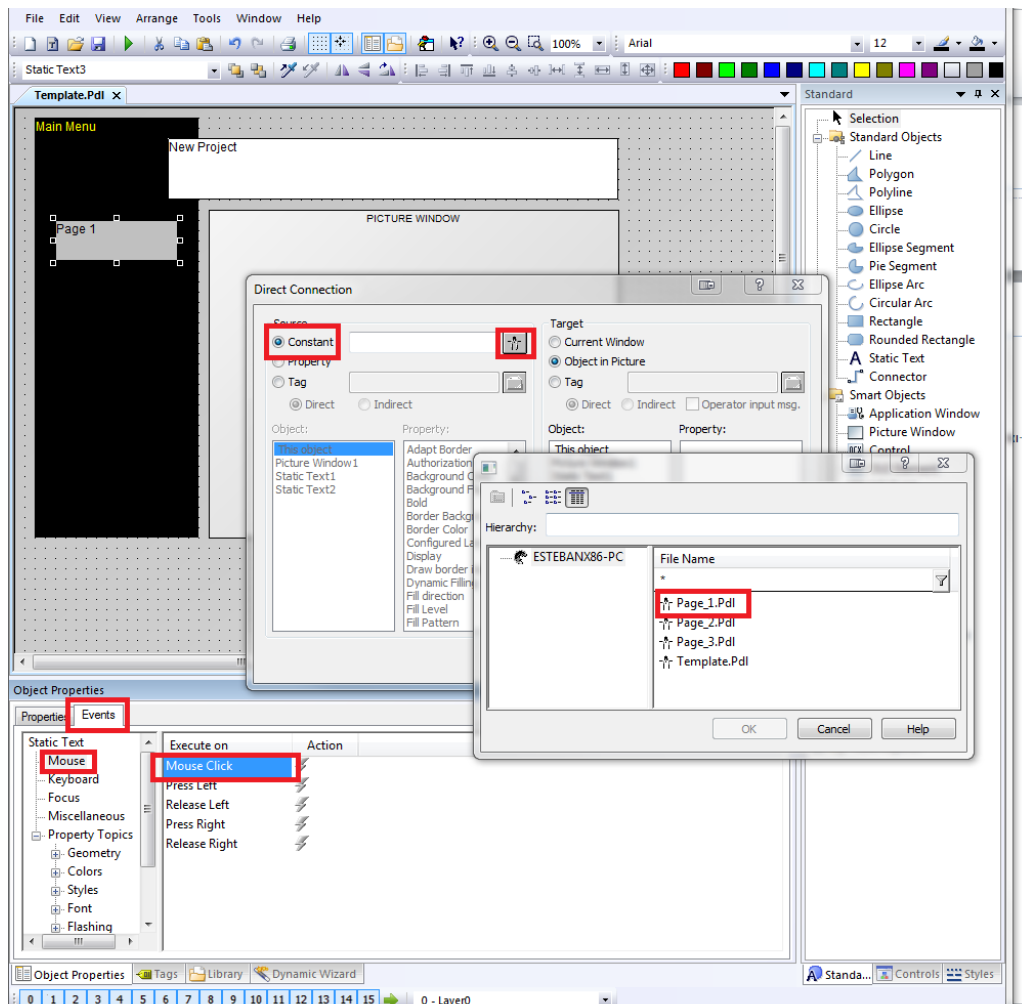
Επιλέγουμε την κατηγορία **smart objects** από το object palette και δημιουργούμε ένα **picture window**. Πατάμε αριστερό κλικ πάνω στο νέο σχήμα και στην καρτέλα **properties** κάτω αριστερά στην οθόνη μας επιλέγουμε **miscellaneous**. Πατάμε διπλό κλικ στην επιλογή **picture name**, και από το παράθυρο που μας εμφανίζεται επιλέγουμε την εικόνα **Page\_1.pdl**.



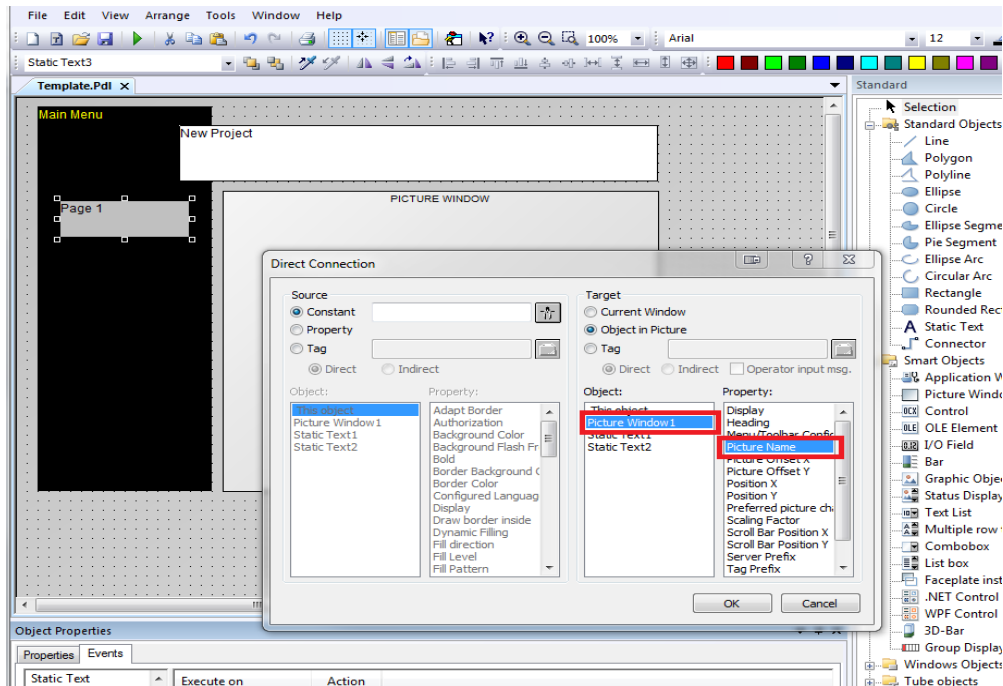
### 3.1.3 Δημιουργία buttons-διαχείριση Event

Έχουμε διαμορφώσει την αρχική μορφή του interface. Στη συνέχεια θα δημιουργήσουμε «κουμπιά», με τα οποία ο χρήστης θα μεταβαίνει από τη μια σελίδα στην άλλη. Αυτό θα επιτευχθεί με τη βοήθεια της καρτέλας Events, η οποία μας δίνει τη δυνατότητα να εισάγουμε actions όταν πραγματοποιηθεί το αντίστοιχο event-στο παράδειγμα μας ένα κλικ.

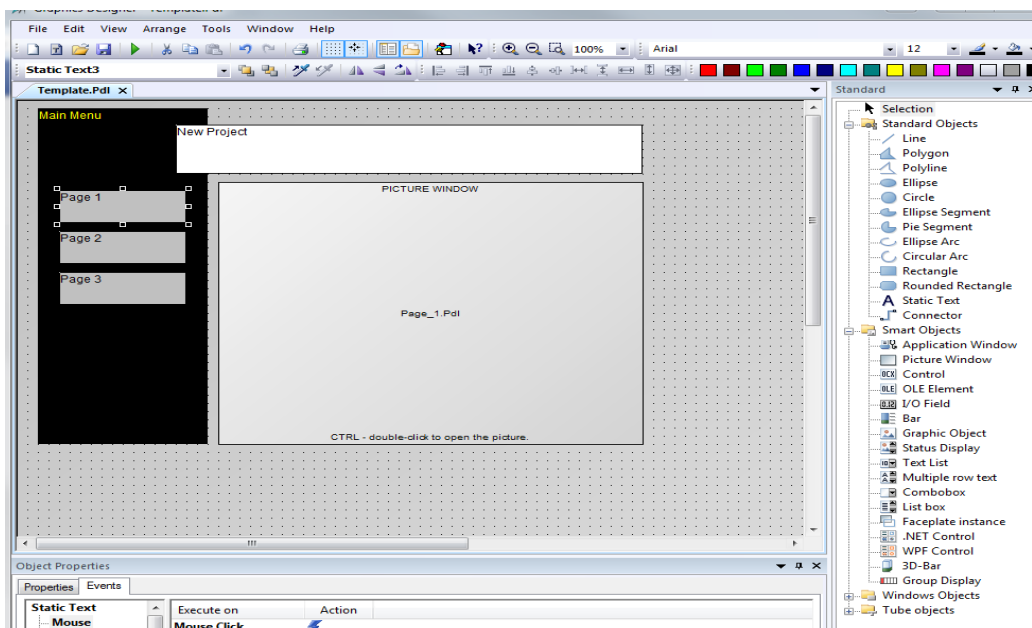
Δημιουργούμε άλλο ένα πεδίο **static text** και το μετονομάζουμε σε **Page 1**. Πατάμε πάνω του αριστερό κλικ, επιλέγουμε την καρτέλα **events** και στην κατηγορία **mouse** πατάμε διπλό κλικ στην επιλογή **mouse click**. Στο νέο παράθυρο επιλέγουμε **constant**, και πατώντας στο κουμπί connect(δεξιά από το constant όπως φαίνεται στο παρακάτω σχήμα) επιλέγουμε την εικόνα **Page 1.pdl**.



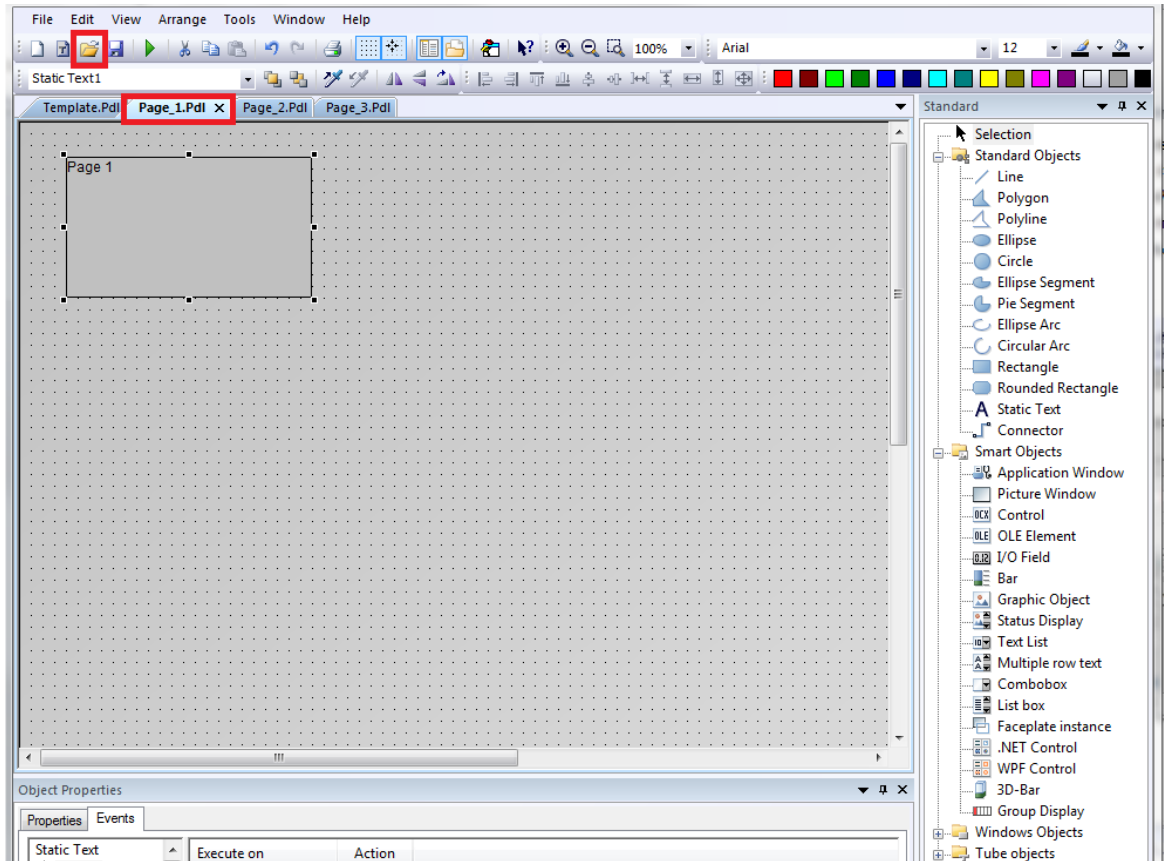
Κάτω από την κατηγορία objects επιλέγουμε **Picture Window1** και στην κατηγορία **property** επιλέγουμε **picture name**. Τέλος πατάμε OK.



Επαναλαμβάνουμε την ίδια διαδικασία άλλες δυο φορές, επιλέγοντας αντίστοιχα Page 2 και Page 3.

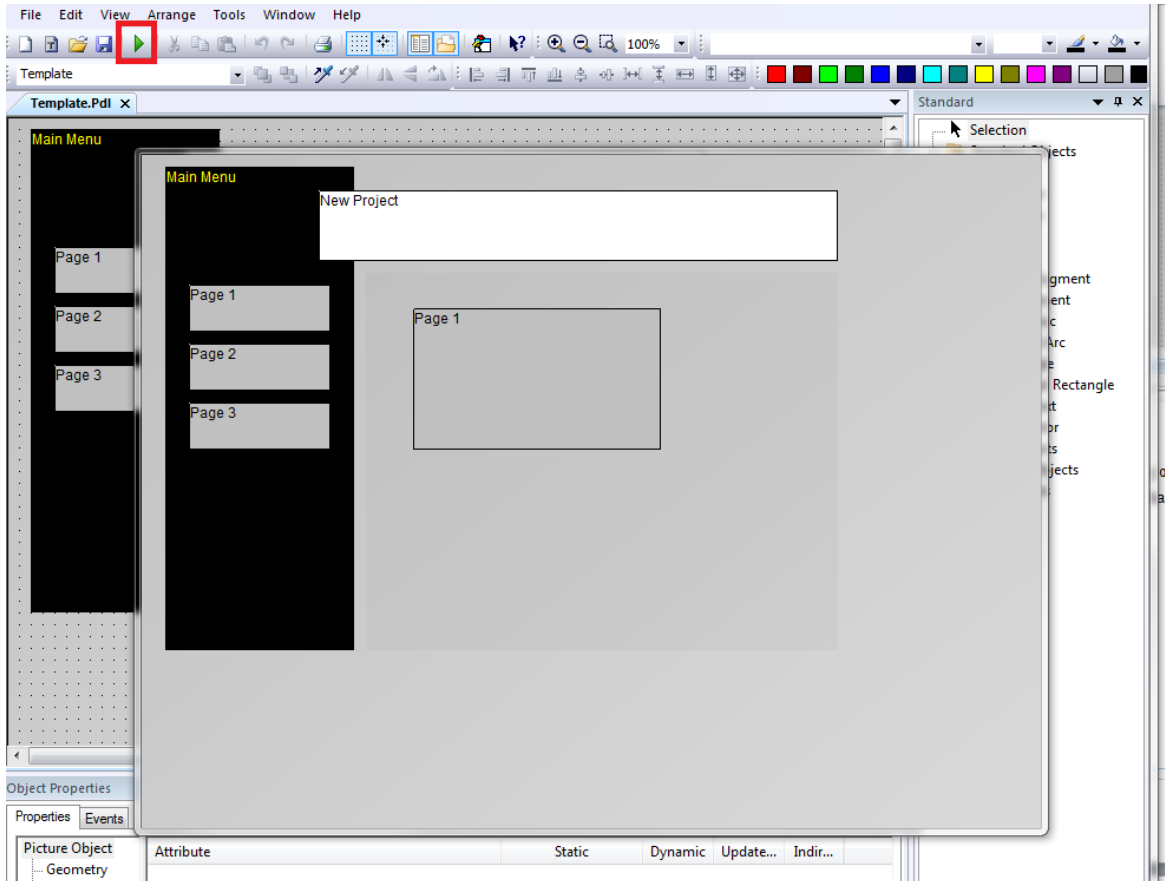


Επιλέγουμε το κουμπί **open** από το toolbar, και ανοίγουμε την εικόνα **Page 1**. Δημιουργούμε ένα **static text** και το μετονομάζουμε σε **page 1**. Εκτελούμε την διαδικασία για τις εικόνες Page 2 και Page 3.



Τέλος, αφού σώσουμε όλες τις αλλαγές, επιλέγουμε την εικόνα **template**, πατάμε το κουμπί **runtime** στο toolbar.

Μόλις τελειώσει το compiling, έχουμε δημιουργήσει ένα βασικό multi page interface. Μέσα από ένα αρχικό μενού, επιλέγοντας τα buttons Page 1, Page 2 και Page 3 ο χρήστης μεταβαίνει στις αντίστοιχες σελίδες.



Σχήμα 1.18: Dynamic Pages Menu



## 3.2 Σχεδιασμός δεξαμενής

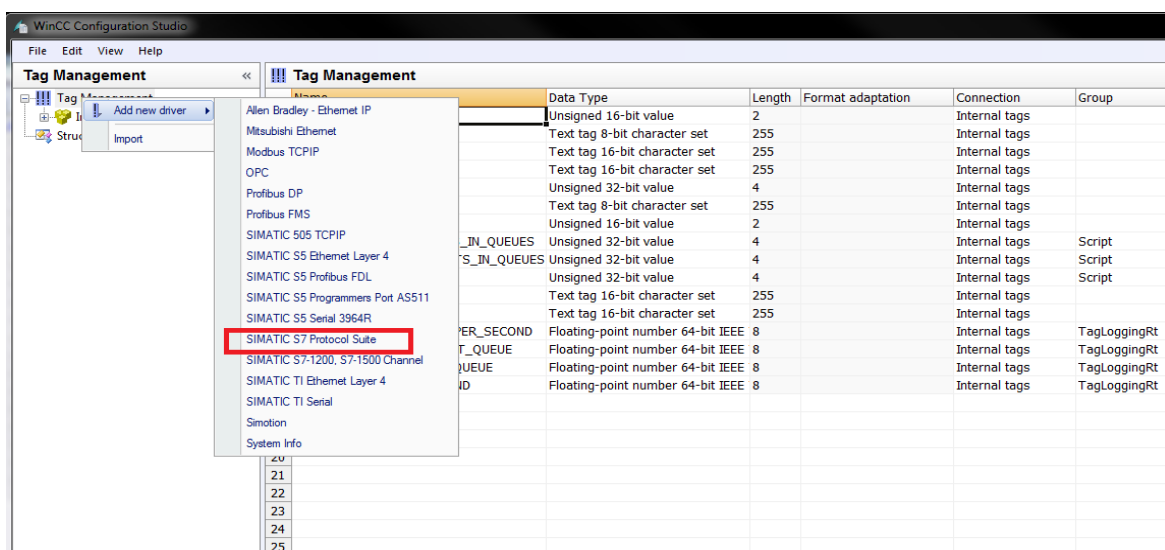
Στο παράδειγμα που ακολουθεί θα δούμε πώς θα δημιουργήσουμε μια δεξαμενή μέσα από τον Graphics Designer και πώς θα της δώσουμε δυναμικά χαρακτηριστικά έτσι ώστε ο χρήστης να μπορεί να ρυθμίζει τη στάθμη του νερού που περιέχει. Σκοπός του παραδείγματος είναι να εξοικειωθεί ο χρήστης με τη χρήση των scripts, του editor Menu/Toolbar και των εσωτερικών και εξωτερικών μεταβλητών.

### 3.2.1 Δημιουργία νέου project

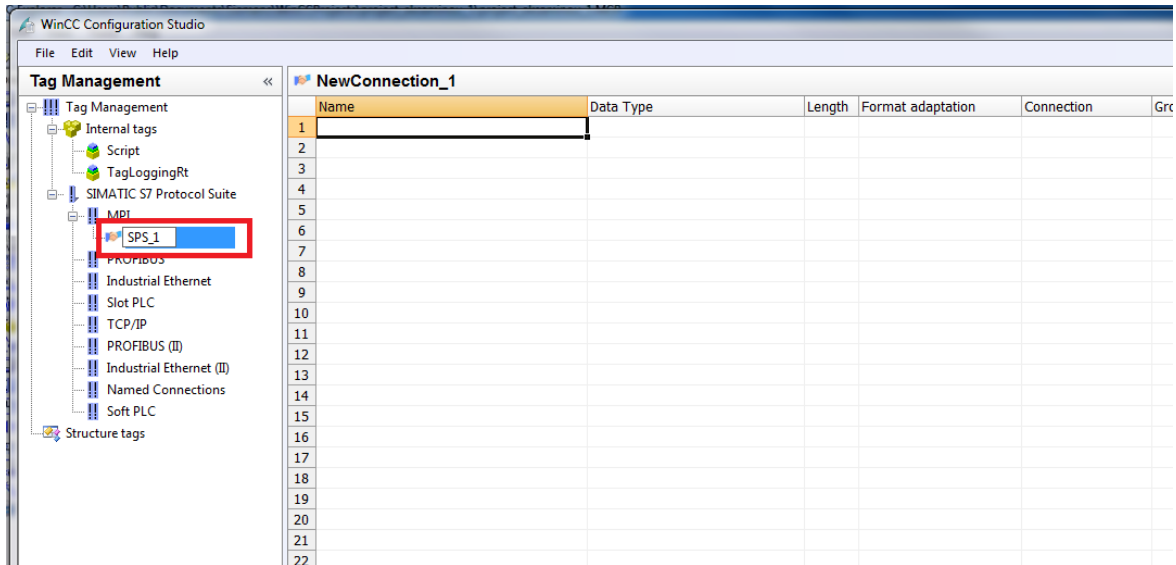
Δημιουργούμε ένα νέο project για τις ανάγκες του παραδείγματος, ακολουθώντας τη διαδικασία που αναφέρθηκε στο προηγούμενο παράδειγμα. Το νέο project θα έχει το όνομα Tank\_project. Επιλέγουμε **computer** στο navigation window, επιλέγουμε τον υπολογιστή μας, πατάμε δεξί κλικ, επιλέγουμε **properties**, και στο πεδίο **Window Attributes** κάτω από την καρτέλα **Graphics Runtime** ενεργοποιούμε τις επιλογές **Full Screen** και **Adapt Picture**.

### 3.2.2 Εγκατάσταση driver

Η εγκατάσταση του driver αποσκοπεί στην επικοινωνία του PLC με το WinCC. Πατάμε διπλό κλικ στην κατηγορία tag management του navigation window. Στο νέο παράθυρο που εμφανίζεται (configuration studio) πατάμε δεξί κλικ στην κατηγορία tag management και επιλέγουμε add new driver -> SIMATIC S7 Protocol Suite.



Υποθέτοντας ότι στην συγκεκριμένη περίπτωση η επικοινωνία μεταξύ υπολογιστή και PLC θα επιτευχθεί μέσω της διεπαφής MPI, κάνουμε δεξί κλικ στην επιλογή MPI και επιλέγουμε New Connection. Μετονομάζουμε την νέα σύνδεση σε SPS\_1.



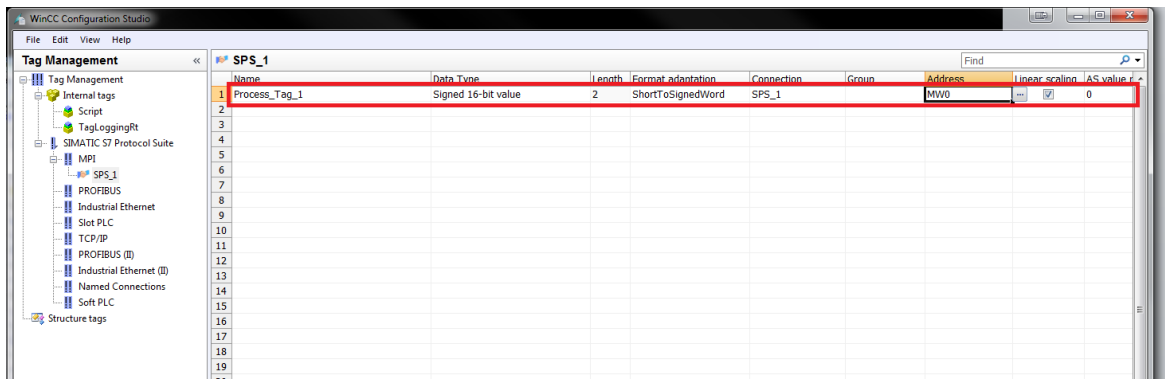
### 3.2.3 Δημιουργία μεταβλητών

Στη συνέχεια θα δημιουργήσουμε τις απαραίτητες μεταβλητές για τη λειτουργία του SCADA. Για την διευκόλυνση μας και την καλύτερη οργάνωση του project θα δημιουργήσουμε ένα **Tag Group**.

Πατάμε δεξί κλικ πάνω στο SPS\_1 και επιλέγουμε **New Group**. Μετονομάζουμε το νέο Group σε **Tag\_Group\_1**.

Στην πρώτη γραμμή της λίστας μεταβλητών στο δεξί μέρος του παραθύρου θα δημιουργήσουμε την πρώτη εξωτερική μεταβλητή (**external tag**). Στο πεδίο **Name** εισάγουμε το όνομα **Process\_Tag\_1**, στο πεδίο **Data Type** επιλέγουμε **Signed 16-bit value**.

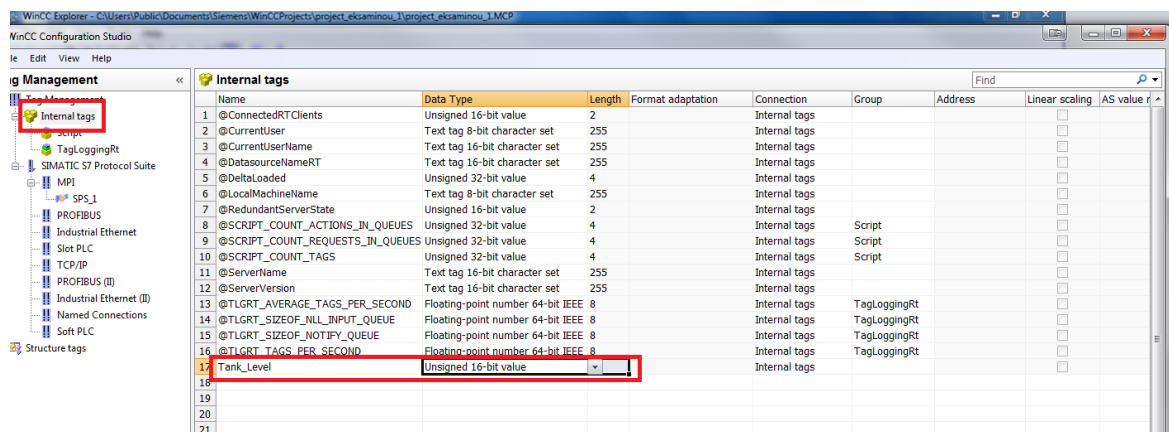
Στο πεδίο **Address** επιλέγουμε **Bit memory** από τη λίστα data και πατάμε OK. Ενεργοποιούμε το **linear scaling** και στα πεδία **AS value range from/to** και **OS value range to** εισάγουμε τις τιμές **-20**, **20** και **100** αντίστοιχα.



Σχήμα 1.19: External Tag

Αφού ολοκληρώσουμε την δημιουργία της εσωτερικής μεταβλητής (**internal tag**), θα δημιουργήσουμε μια εσωτερικής μεταβλητής. Οι εσωτερικές μεταβλητές χρησιμοποιούνται για εσωτερικά θέματα του λογισμικού και για λειτουργίες προσομοίωσης.

Στην κατηγορία Internal Tags δημιουργούμε μια νέα μεταβλητή με το όνομα **Tank\_Level** και στο πεδίο **Data type** επιλέγουμε **Unsigned 16 bit value**.

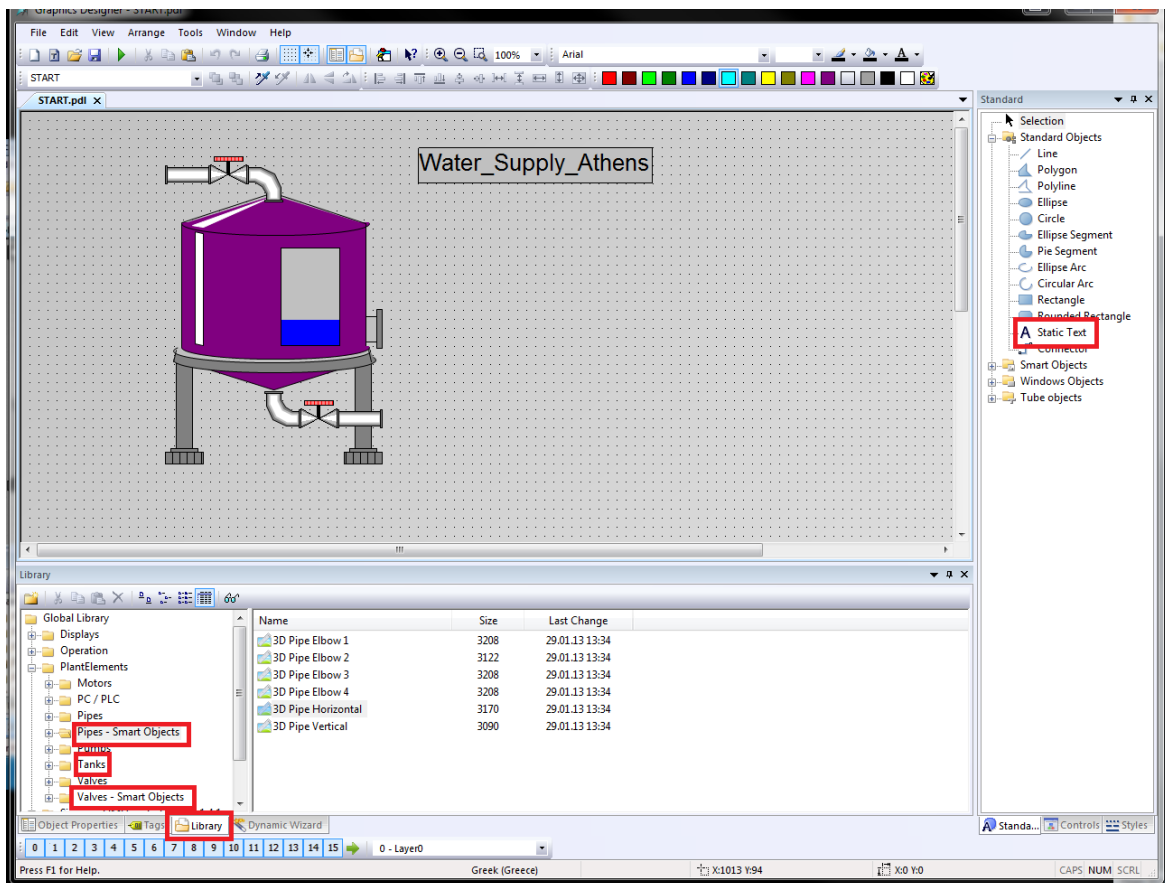


Σχήμα 1.20: Internal Tag

### 3.2.4 Σχεδιασμός γραφημάτων

Κλείνουμε το Tag management και δημιουργούμε δυο νέες εικόνες στον graphics designer όπως αναφέρθηκε στο προηγούμενο παράδειγμα (3.1 Σχεδιασμός δυναμικών σελίδων). Μετονομάζουμε τις νέες εικόνες σε **START.pdl** και **SAMPLE.pdl**.

Ανοίγουμε την εικόνα **START.pdl** και από την καρτέλα **Library-> Global Library**, στην κατηγορία **Tanks** επιλέγουμε την δεξαμενή **Tank 1**. Από την κατηγορία **pipes – smart objects** επιλέγουμε **3D Pipe Elbow 3**, και από την κατηγορία **Valves – smart objects** επιλέγουμε **valve1**, και τα τοποθετούμε όπως φαίνεται στο παρακατω σχήμα.



Σχήμα 1.21: Tank

Σώζουμε το γράφημα και κλείνουμε τον Graphics designer.

### 3.2.5 Δημιουργία και ενσωμάτωση των scripts στο γράφημα

Στη συνέχεια θα δημιουργήσουμε το script για τη στάθμη του νερού της δεξαμενής και το script για την έξοδο από την εφαρμογή.

Ανοίγουμε τον VB Editor πατώντας στην κατηγορία Global Script του navigation window. Δημιουργούμε ένα νέο script πατώντας το κουμπί Create new project όπως φαίνεται στο παρακάτω σχήμα.

Για τη δημιουργία του script για τη στάθμη του νερού γραφούμε τον εξής κώδικα:

```
Sub ActivatePicture (Byval PictureName)

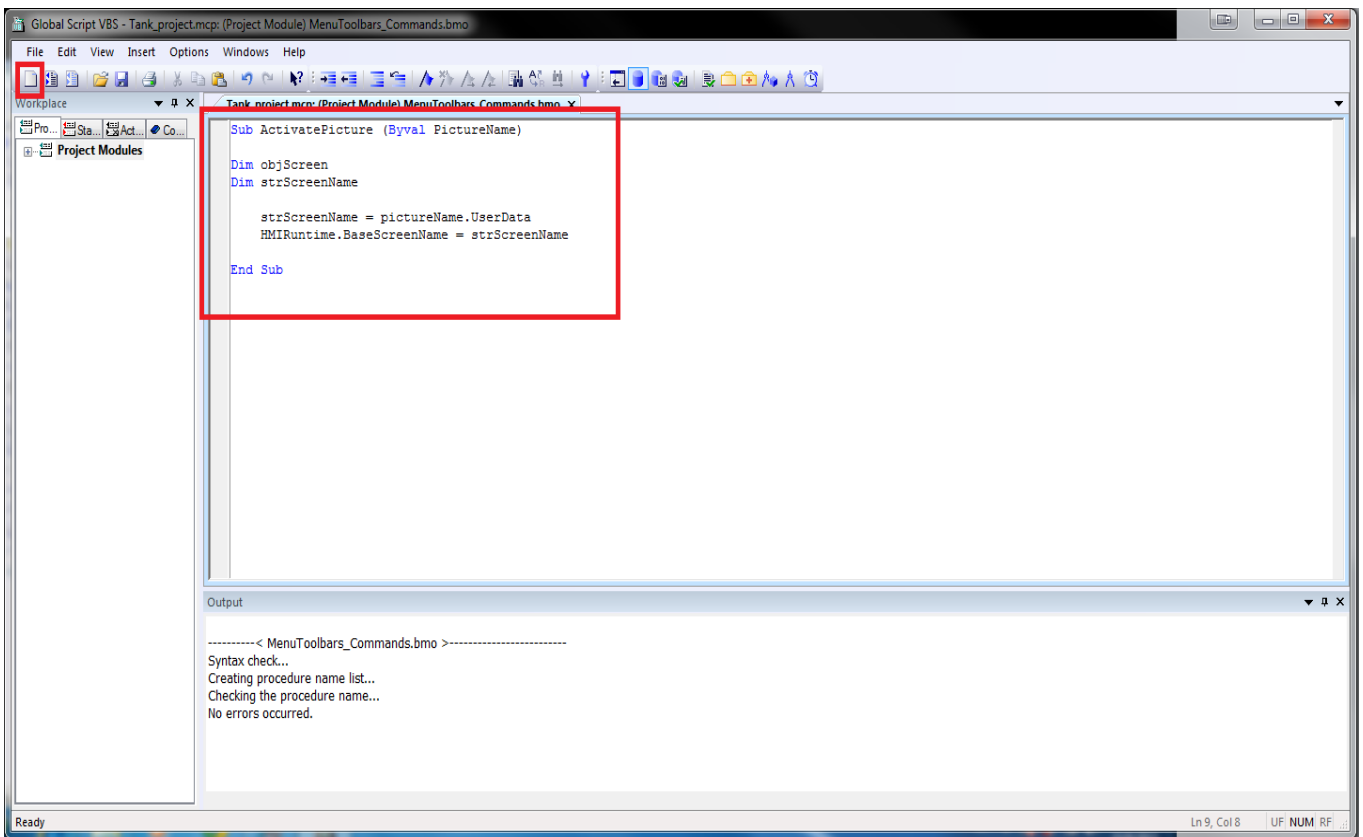
Dim objScreen

Dim strScreenName

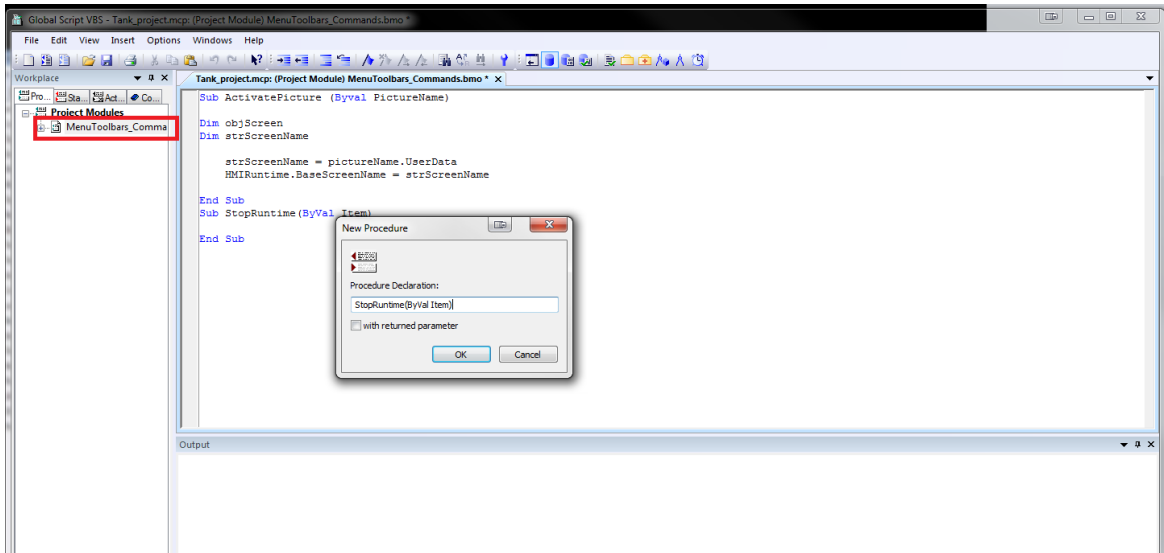
strScreenName = pictureName.UserData

HMIRuntime.BaseScreenName = strScreenName

End Sub
```



Σώζουμε το script ως **MenuToolbars\_Commands**. Για τη δημιουργία του script εξόδου από την εφαρμογή, πατάμε δεξί κλικ στο script που δημιουργήσαμε, και επιλέγουμε **Add new procedure**. Εισάγουμε το όνομα `StopRuntime(ByVal Item)`.

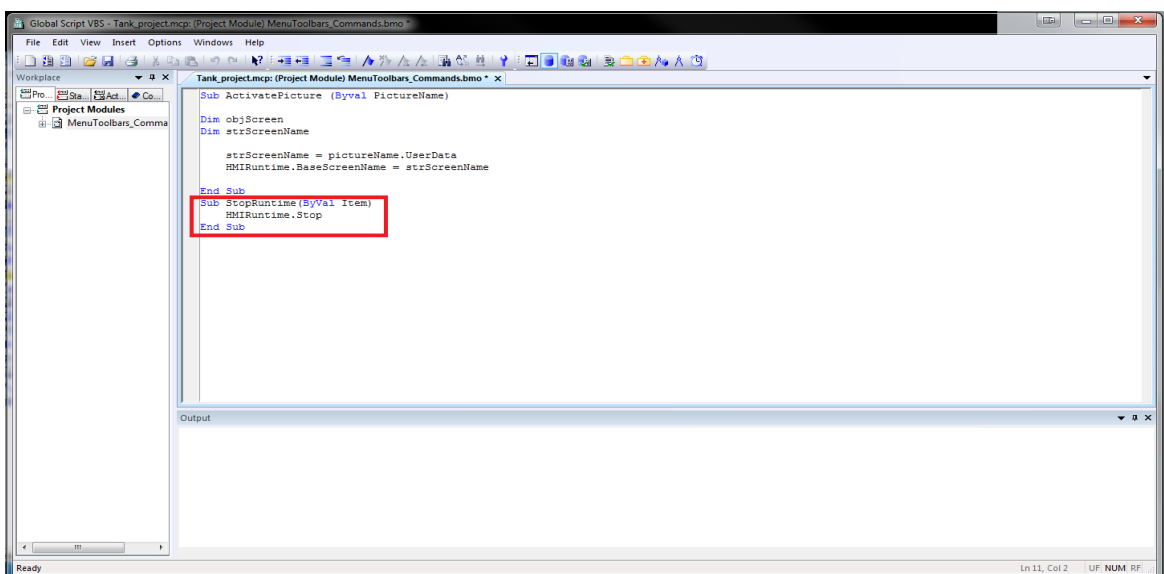


Στη συνέχεια εισάγουμε τον παρακάτω κώδικα.

```
Sub StopRuntime(ByVal Item)
```

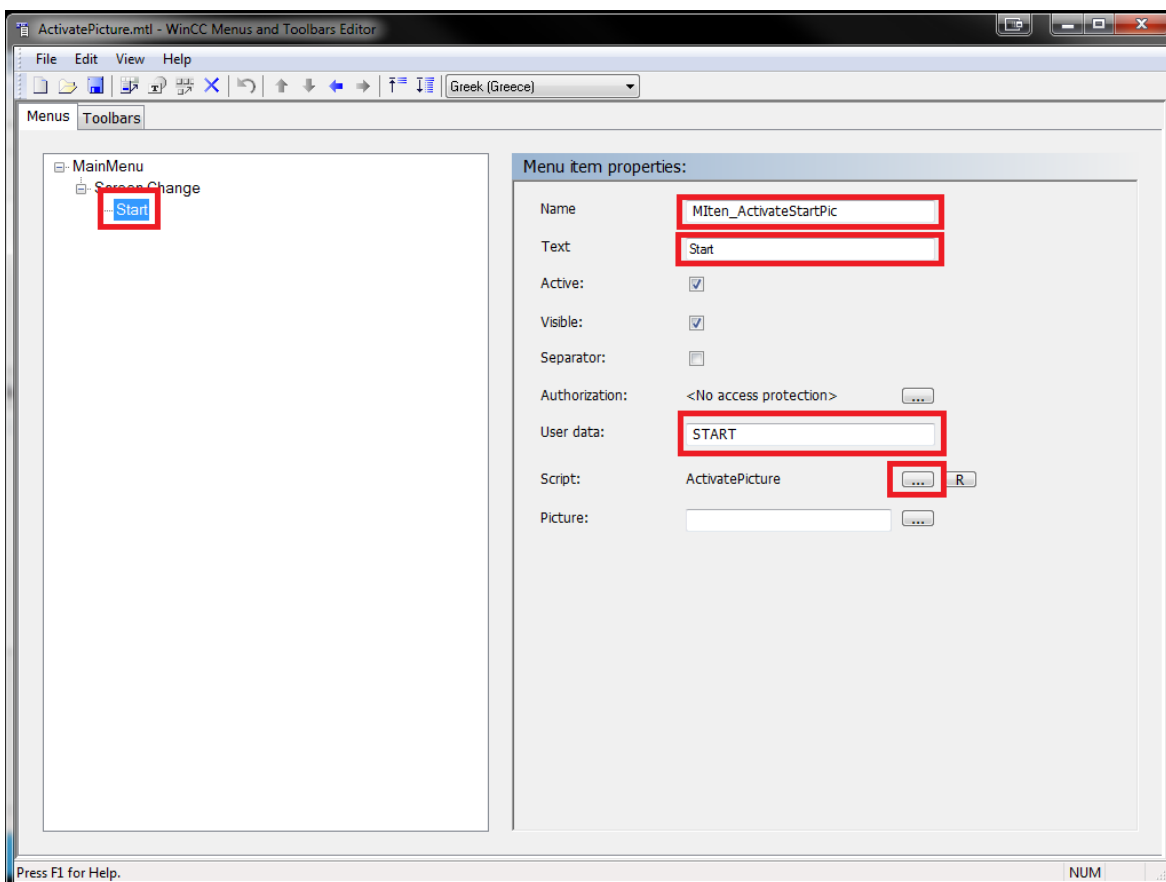
```
HMIRuntime.Stop
```

```
End Sub
```



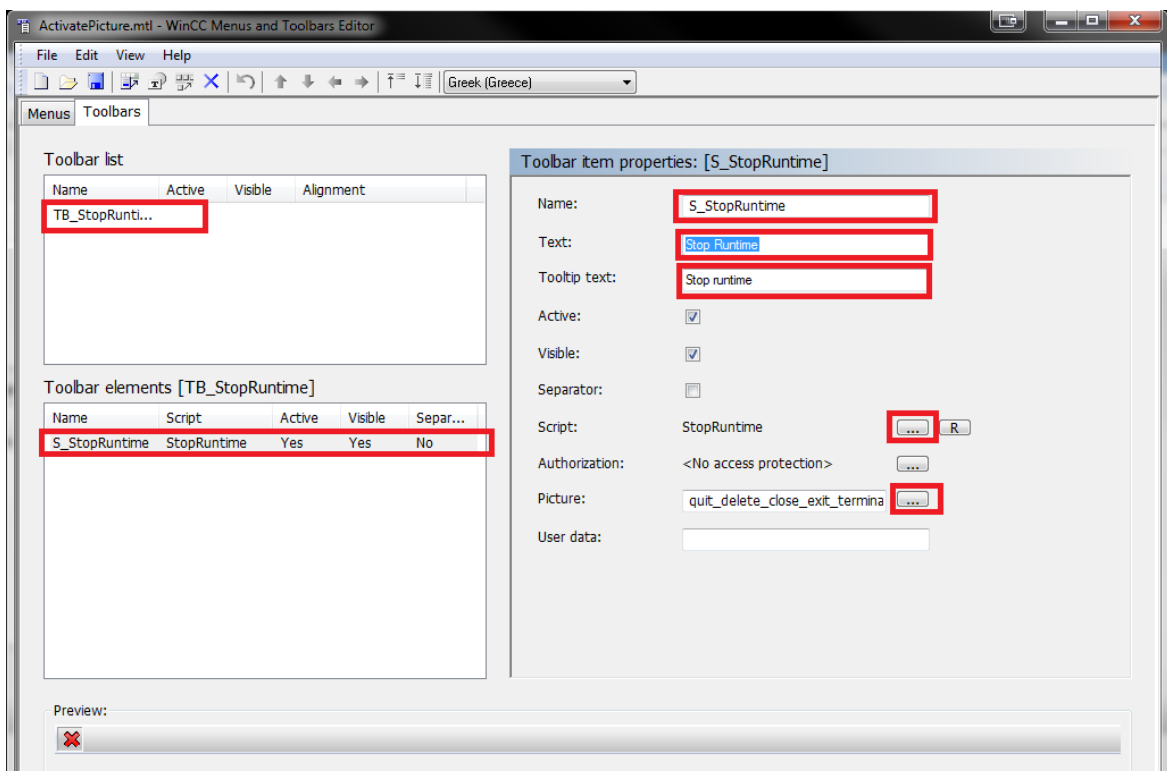
Για να ενσωματώσουμε τις συναρτήσεις μας στο γράφημα ακολουθούμε τα εξής βήματα:

- Ανοίγουμε τον **Menu and toolbar Editor** από το navigation window του WinCC explorer.
- Πατάμε δεξί κλικ στο **Main Menu** και επιλέγουμε **Insert Menu Item**.
- Μετονομάζουμε σε **M\_ActivatePic** και στο πεδίο text εισάγουμε **Main Menu**.
- Πατάμε δεξί κλικ στο δεύτερο **Main Menu** και επιλέγουμε **Insert Menu Item**.
- Μετονομάζουμε σε **MItem\_ActivateStartPic**
- Στο πεδίο **User Data** εισάγουμε το όνομα **START**
- Στην επιλογή script εισάγουμε το **script Activate Picture**.



Για να εισάγουμε το script που δημιουργήσαμε για την έξοδο της εφαρμογής ακολουθούμε τα εξής βήματα:

- Στην καρτέλα **Toolbars** του Menu/Toolbar editor στην **κατηγορία Toolbar List** πατάμε δεξί κλικ και επιλέγουμε **Insert Toolbar**
- Μετονομάζουμε σε **TB\_StopRuntime**, στο πεδίο text εισάγουμε **Stop Runtime**
- Στην κατηγορία **Toolbar Elements** πατάμε δεξί κλικ και **επιλέγουμε Insert Toolbar item**
- Μετονομάζουμε σε **S\_StopRuntime** και στα πεδία text και tooltip text εισάγουμε **Stop Runtime**.
- Στην επιλογή script επιλέγουμε το script **StopRuntime**
- Στην επιλογή **picture** επιλέγουμε την εικόνα της αρεσκειάς μας που θα κοσμήι το button.



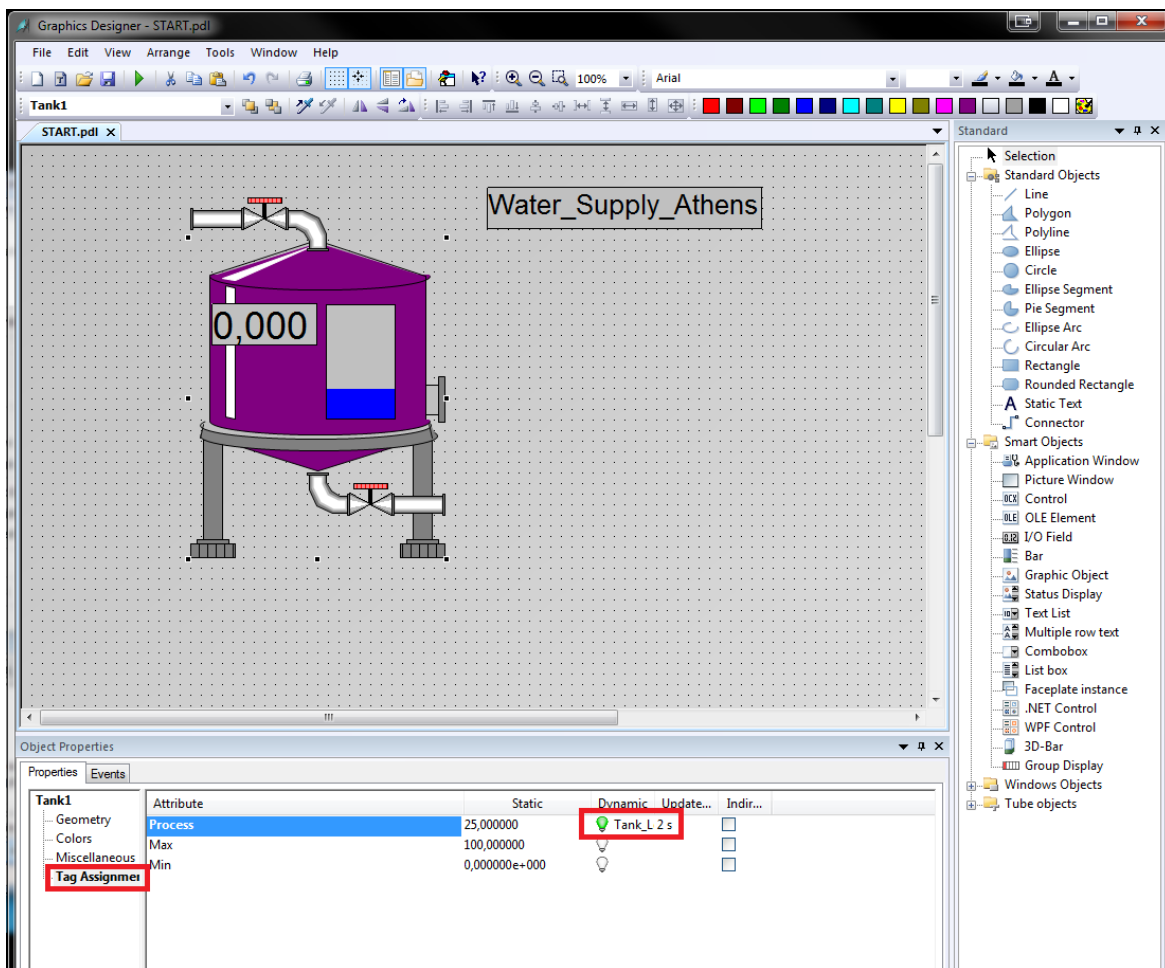
Σώζουμε τις αλλαγές δίνοντας το όνομα ActivatePicture.mtl και κλείνουμε τον Menu/Toolbar editor.



### 3.2.6 Πρόσθεση δυναμικών χαρακτηριστικών στη δεξαμενή

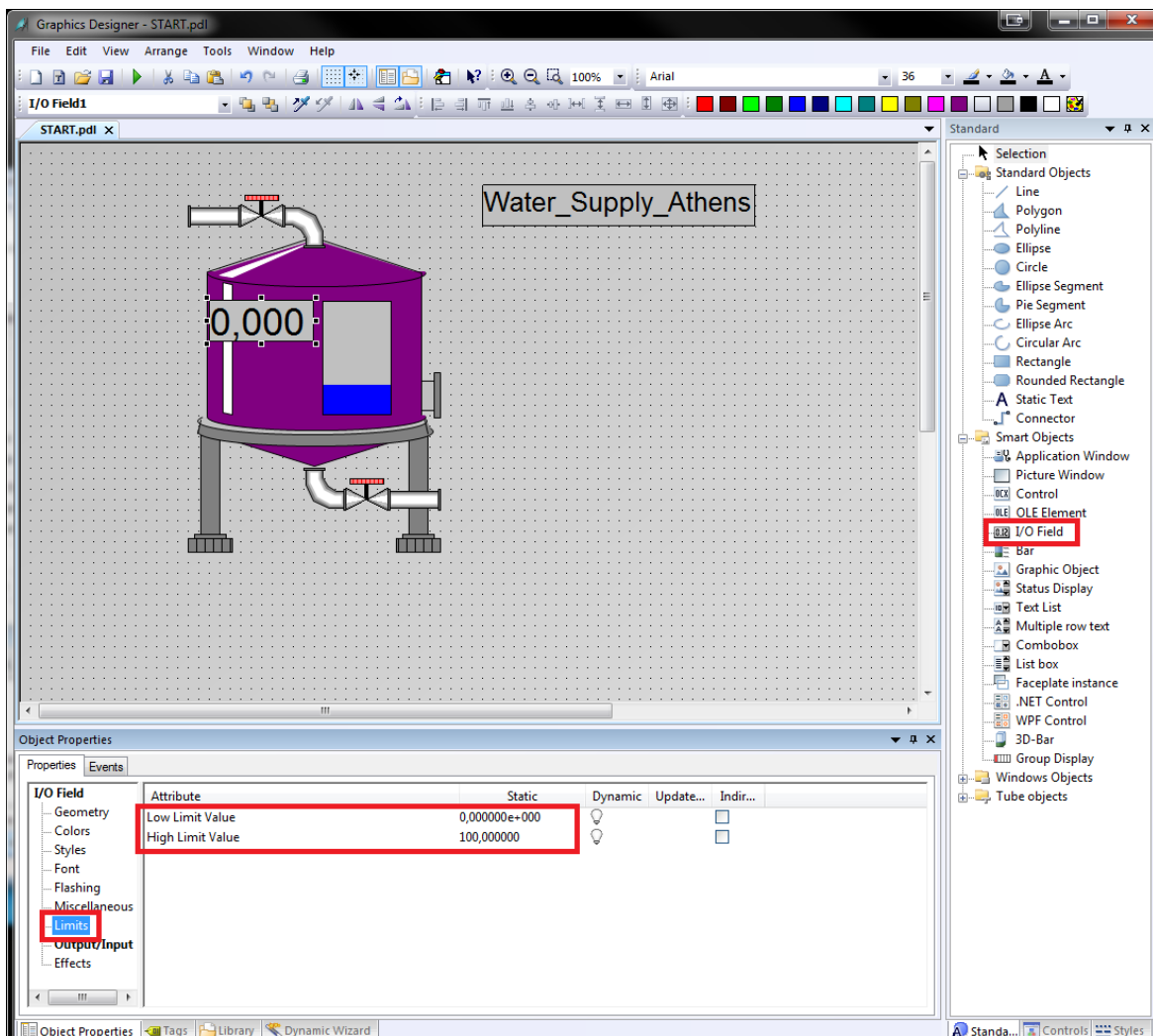
Στη συνέχεια θα προσθέσουμε λειτουργικότητα στη δεξαμενή του Start.pdl. Ανοίγουμε το γράφημα START.pdl με τον graphics editor, επιλέγουμε την δεξαμενή, πατάμε δεξί κλικ και επιλέγουμε properties.

- Στην καρτέλα **properties** επιλέγουμε **tag assignment** και κάνουμε δεξί κλικ στην κατηγορία **Dynamic** της επιλογής **process**.
- Επιλέγουμε **tags**, **internal tags** και εντοπίζουμε την εσωτερική μεταβλητή **Tank\_Level** που έχουμε δημιουργήσει. Την επιλέγουμε και πατάμε OK.



Έχουμε ορίσει την μέγιστη και την κατώτατη ποσότητα νερού της δεξαμενής. Για να ορίσουμε τον τρόπο με τον οποίο ο χρήστης θα μεταβάλλει την στάθμη του νερού θα προσθέσουμε ένα I/O Field από την κατηγορία Smart Objects του Object Palette.

- Αφού επιλέξουμε το **I/O Field**, το εισάγουμε στο γράφημα και επιλέγουμε στην κατηγορία tag την μεταβλητή **Tank\_Level** και αλλάζουμε το **update σε 500ms** στο configuration window που μας εμφανίζεται.
- Πατάμε δεξί κλικ **στο I/O Field**, επιλέγουμε **properties** και στην επιλογή **limits** ορίζουμε **Low Limit Value 0** και **High Limit Value 100**.

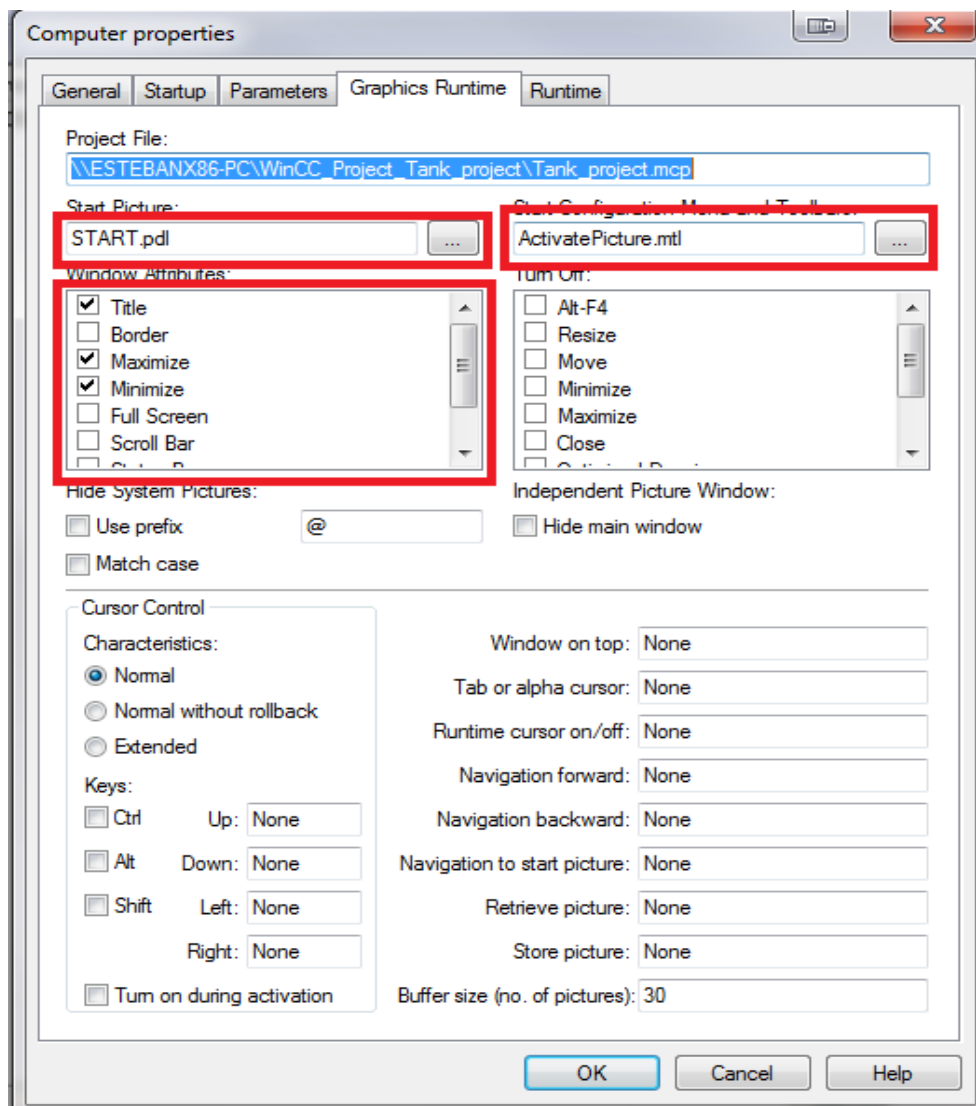


Σώζουμε τις αλλαγές και κλείνουμε τον Graphics Designer.

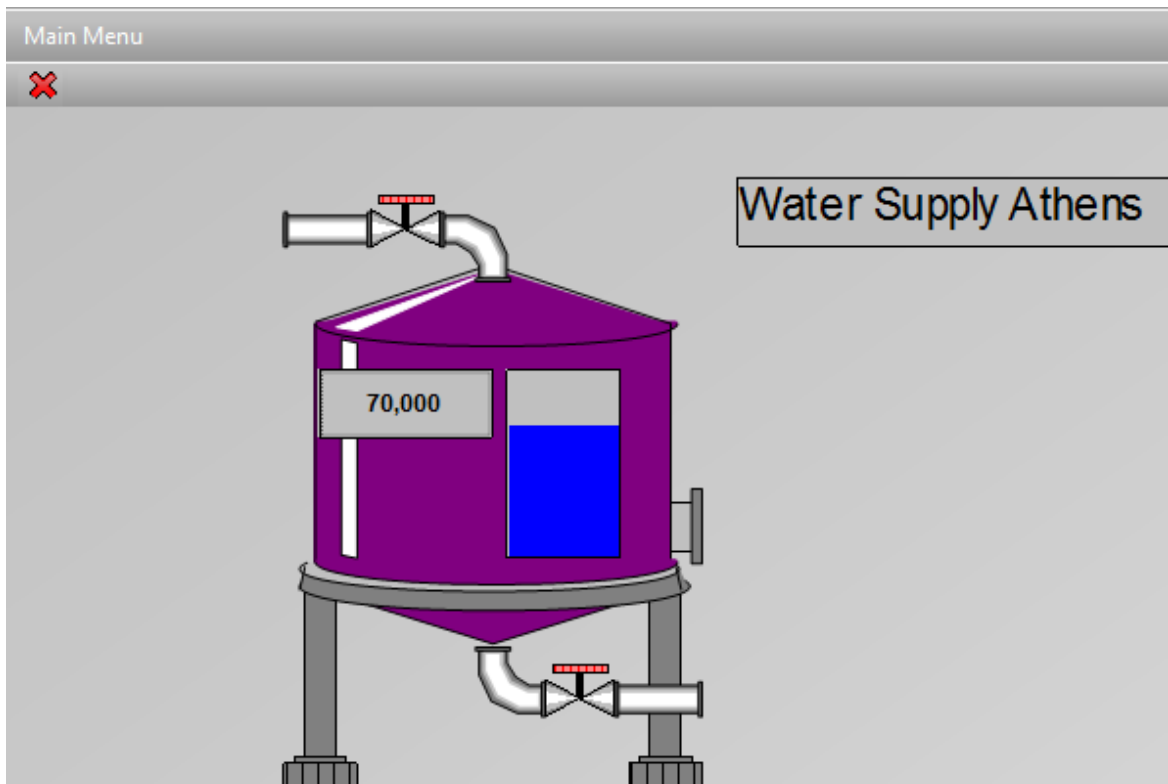
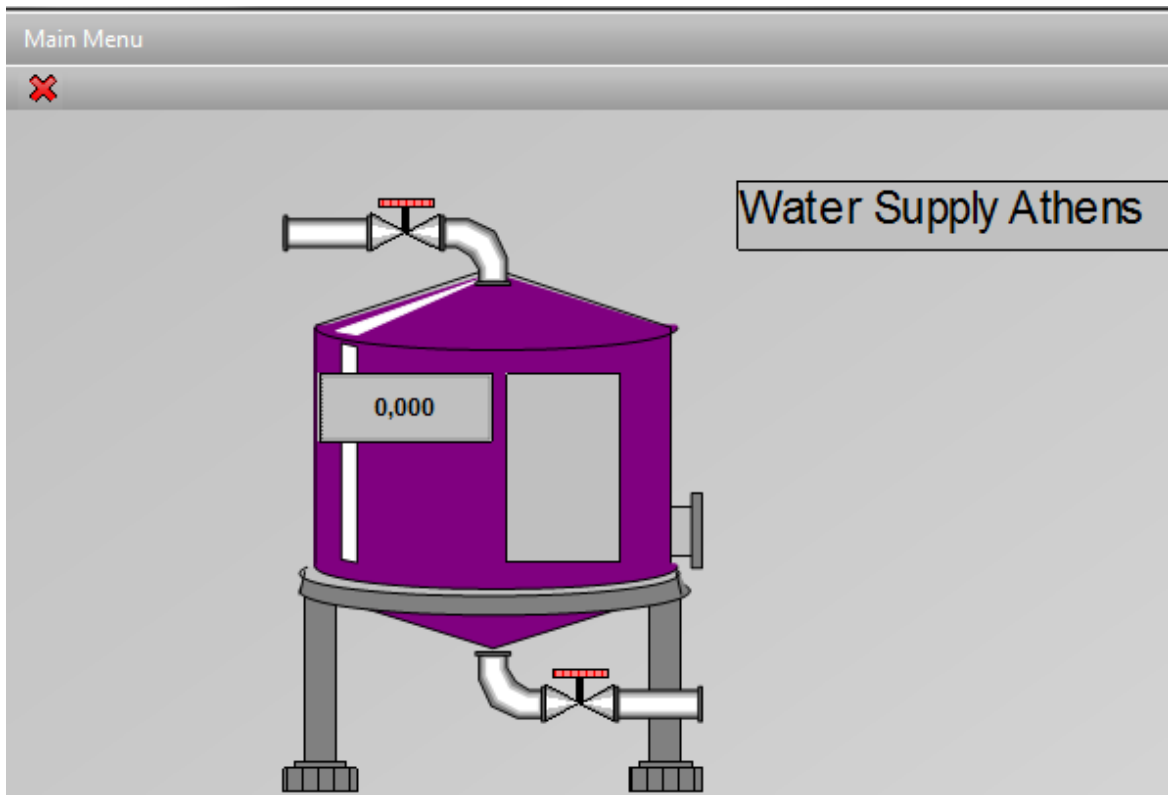
### 3.2.7 Εκκίνηση της εφαρμογής

Στη συνέχεια θα πρέπει να εφαρμόσουμε τις κατάλληλες ρυθμίσεις για να τρέξει η εφαρμογή μας σωστά.

- Στην επιλογή **computer** του navigation window κάνουμε δεξί κλικ στο όνομα του υπολογιστή μας και επιλέγουμε properties.
- Στην καρτέλα **Graphics Runtime**, στην επιλογή Start Picture επιλέγουμε το γράφημα **START.pdl**, στην επιλογή **Start configuration Menu and toolbars** επιλέγουμε **ActivatePictures.mdl**.
- Στα **Window Attributes** ενεργοποιούμε τις επιλογές **Title**, **Maximize**, **Minimize** και **Adapt Picture**.



Επιλέγουμε File και πατάμε Activate. Η εφαρμογή μας θα εμφανιστεί στην οθόνη. Μεταβάλλοντας την τιμή του I/O Field παρατηρούμε ότι η στάθμη του νερού παίρνει την αντίστοιχη τιμή.



## ΒΙΒΛΙΟΓΡΑΦΙΑ

- 1 Α. Βελώνη - Σ. Αλατσαθιανός, «**Βιομηχανική Πληροφορική**», Αθήνα 2014
- 2 Γεώργιος Κορρές, «**Στοιχεία Συστημάτων SCADA**», Αναπληρωτής Καθηγητής Ε.Μ.Π., Σημειώσεις Ε.Μ.Π., Αθήνα 2001.
- 3 Παπαδόπουλος Θ. Βασίλειος «**Μελέτη Συστήματος Παροχής Ηλεκτρικών Τάσεων σε Εργαστηριακό Χώρο-Παρεμβάσεις Σε Συστήματα PLC Μέσω Του Προγράμματος SCADA**» Πανεπιστήμιο Πατρών, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών-Τομέας Συστημάτων Ηλεκτρικής Ενεργείας, διπλωματική εργασία-Σεπτέμβριος 2007
- 4 Σιγάλας Αρτέμιος-Σκαρπέτης Δημήτριος , «**SIMATIC HMI WINCC v7.2**» Τμήμα Μηχανικών ηλεκτρονικών Υπολογιστικών Συστημάτων ΤΕΙ Πειραιά
- 5 «**SCADA System SIMATIC WinCC**» - <http://www.siemens.com/entry/cc/en/>
- 6 Νικόλαος Σαρρής «**Εποπτικός Έλεγχος Και Συλλογή Δεδομένων (Scada) Ανάλυση Και Υλοποίηση Σε Δίκτυα Ethernet**» διπλωματική εργασία, Θεσσαλονίκη-Σεπτέμβριος 2002
- 7 Γεώργιος Σ. Σκιαδόπουλος «**Εργαστηριακές Εφαρμογές Scada**» Εθνικό Μετσόβιο Πολυτεχνείο-Σχολή Ηλεκτρολόγων Μηχανικών Και Μηχανικών Υπολογιστών Τομέας Ηλεκτρικής Ισχύος διπλωματική εργασία, Αθήνα, Σεπτέμβριος 2007