Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Πτυχιακή εργασία

Βελτιστοποίηση εφαρμογής ασύρματης τηλεπιτήρησης και τηλεχειρισμού δικτύων άρδευσης και ύδρευσης (SCADA)

Ανυφαντάκης Γεώργιος (ΑΜ: 986)
Κοντογιάννης Στέργιος (ΑΜ: 1144)

Ηράκλειο

Επόπτης Καθηγητής: Δρ. Μιαουδάκης Ανδρέας

Υπεύθυνη Δήλωση: *Βεβαιώνουμε ότι είμαστε συγγραφείς αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχαμε για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχουμε αναφέρει τις όποιες πηγές από τις οποίες κάναμε χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνουμε ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμάς προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Εφαρμοσμένης Πληροφορικής και Πολυμέσων του Α.Τ.Ε.Ι. Κρήτης.*

Copyright © Ανυφαντάκης Γεώργιος, Κοντογιάννης Στέργιος 2010.
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε τους γονείς μας για την αμέριστη υποστήριξη κατά την διάρκεια εκπόνησης της εργασίας αυτής.
Θα θέλαμε επίσης να ευχαριστήσουμε τον καθηγητή κ. Μιαουδάκη Ανδρέα για την πολύτιμη καθοδήγηση και υποστήριξη του σε όλη τη διάρκεια εκπόνησης της εργασίας αυτής, αλλά και την δυνατότητα που μας έδωσε να ασχοληθούμε με ένα τόσο ενδιαφέρον επιστημονικό αντικείμενο. Ιδιαίτερα όμως θα θέλαμε να τον ευχαριστήσουμε γιατί υπήρξε ο άνθρωπος που μας παρότρυνε να συνεχίσουμε παρ' όλες τις δυσκολίες που αντιμετωπίσαμε, πιστεύοντας στις δυνάμεις και στις δυνατότητες μας

Περιεχόμενα

Πίνακας εικόνων

Κεφάλαιο 1 Εισαγωγή στα Συστήματα SCADA

1.1 Βασικές Έννοιες

Τι Σημαίνει SCADA :

Το SCADA (Supervisory Control And Data Acquisition),σημαίνει Εποπτικός έλεγχος και απόκτηση στοιχείων. Όπως το όνομα δείχνει, δεν είναι ένα πλήρες σύστημα ελέγχου, αλλά μάλλον εστιάζει στο εποπτικό επίπεδο. Υπό αυτήν τη μορφή, είναι ένα καθαρό πακέτο λογισμικού που τοποθετείται πάνω από το υλικό στο οποίο διασυνδέεται, γενικά μέσω των λογικών ελεγκτών (PLCs), ή άλλων εμπορικών ενοτήτων υλικού. Τα συστήματα SCADA χρησιμοποιούνται όχι μόνο στις βιομηχανικές διαδικασίες: π.χ. χαλυβουργική, ηλεκτρική παραγωγή (συμβατική και πυρηνική) και στη διανομή της, χημεία, αλλά και σε μερικές πειραματικές εγκαταστάσεις όπως η πυρηνική τήξη. Το μέγεθος τέτοιας σειράς εγκαταστάσεων εκτείνετε από 1000 μέχρι 10 χιλιάδες κανάλια εισόδου-εξόδου (I/O).

Εντούτοις, τα συστήματα SCADA εξελίσσονται γρήγορα και ξεπερνούν τώρα την αγορά των εγκαταστάσεων με διάφορα I/O κανάλια .Τα συστήματα SCADA αρχικά υλοποιήθηκαν στο DOS, (VMS) και το Unix αλλά τα τελευταία χρόνια όλοι οι προμηθευτές SCADA έχουν κινηθεί προς τα NT και μερικά επίσης προς το Linux.

SCADA ως Σύστημα

Υπάρχουν πολλά μέρη της εργασίας του συστήματος SCADA. Ένα σύστημα SCADA συνήθως περιλαμβάνει υλικό σήμα (εισόδου και εξόδου), των ελεγκτών, τα δίκτυα, τη διεπαφή χρήστη (HMI), τον εξοπλισμό και το λογισμικό. Όλα μαζί, με τον όρο SCADA αναφέρονται σε ολόκληρο το κεντρικό σύστημα. Το κεντρικό σύστημα συνήθως παρακολουθεί δεδομένα από διάφορους αισθητήρες, που είναι είτε πολύ κοντά είτε πολύ μακριά.

Ενδεικτικά κατηγορίες στοιχείων που μετρούνται, καταγράφονται, επιτηρούνται και ελέγχονται είναι:

1. Θερμοκρασία
2. Πίεση
3. Υγρασία
4. Στάθμη
5. Ροή
6. Στροφές
7. Ταχύτητα
8. Απόσταση
9. Ενέργεια
10. Διακόπτες (ON/OFF)
11. Κραδασμοί
12. Ανίχνευση αερίων
13. Καταμέτρηση
14. Ζύγιση
15. Αναγνώριση
16. Ποιοτικό έλεγχο
17. Χρόνο

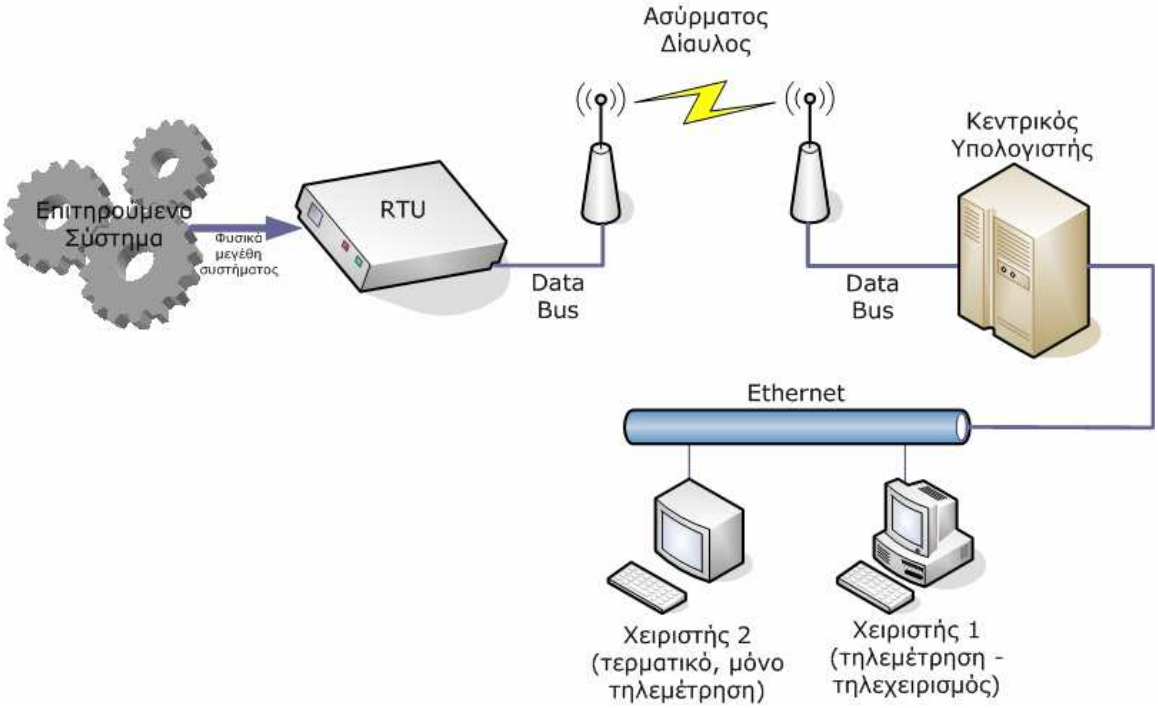
1.2 Τα βασικά μέρη ενός SCADA

Ένα σύνηθες σύστημα SCADA χρησιμοποιεί σαν κεντρικό πυρήνα έναν κεντρικό υπολογιστή, αρκετά μεγάλης υπολογιστικής ισχύος, στον οποίο βρίσκεται το λογισμικό SCADA εγκατεστημένο, όπως και το πρόγραμμα της εκάστοτε εφαρμογής. Η ζητούμενη τηλεμετρία στον επιθυμητό χώρο, επιτυγχάνεται με την εγκατάσταση σταθμών τηλεμετρίας RTU (Remote Telemetry Units).

Εικόνα 1 Εσωτερική Εγκατάσταση RTU

Εικόνα 2 Εξωτερική Εγκατάσταση RTU

Οι σταθμοί αυτοί «διαβάζουν» τις τιμές διαφόρων μεγεθών που μας ενδιαφέρουν (τάση, πίεση, θερμοκρασία κτλ), τις μετατρέπουν σε ηλεκτρικά σήματα και τα σήματα αυτά τα μεταδίδουν ενσύρματα ή ασύρματα με κατάλληλες τηλεπικοινωνιακές ζεύξεις στον κεντρικό υπολογιστή, ανά τακτά χρονικά διαστήματα. Από εκεί και πέρα, αρχίζει η παρακολούθηση και επεξεργασία τους από τους χρήστες του κεντρικού υπολογιστή και εξάγονται χρήσιμα συμπεράσματα για τη λειτουργία της εκάστοτε διεργασίας .



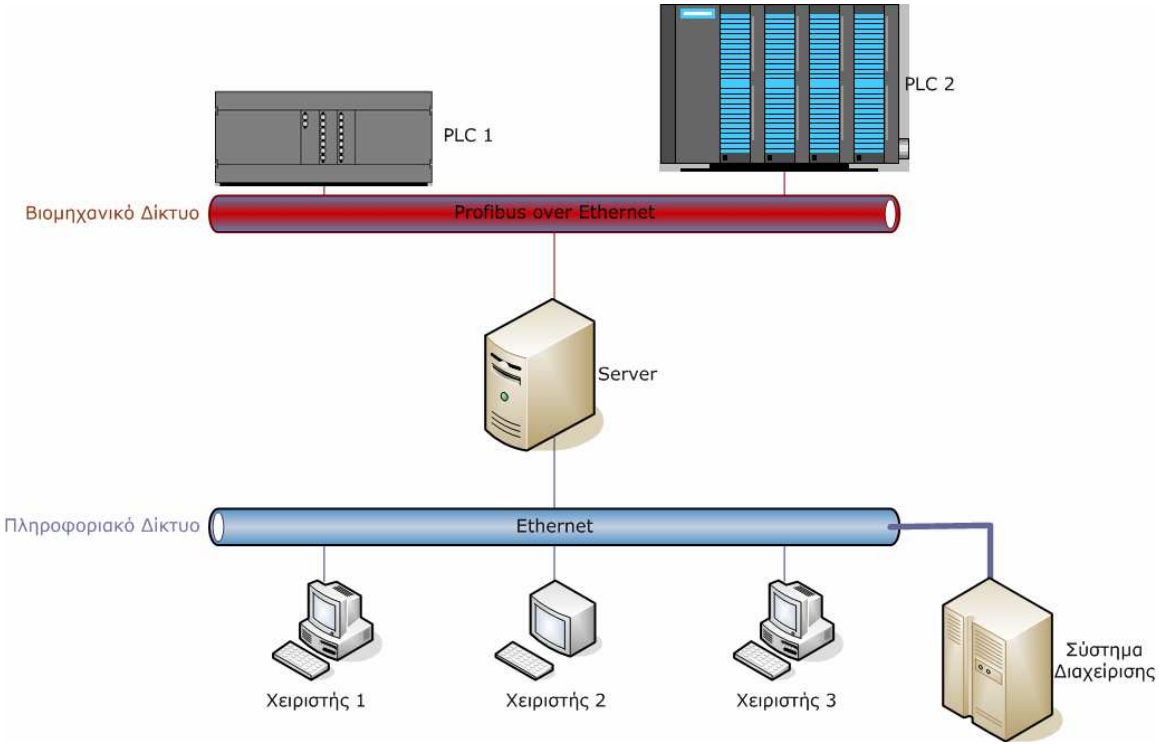
Εικόνα 3 Τοπολογία συστήματος SCADA.

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

Όπως δείχνει η εικόνα 4, για την δικτύωση του κεντρικού υπολογιστή με τους χρήστες (πληροφοριακό δίκτυο), χρησιμοποιούνται τα πρωτόκολλα Ethernet ή Token Ring ενώ για την δικτύωση των PLC (βιομηχανικό δίκτυο), έχουμε τα πρωτόκολλα Profibus, TCP/IP και Industrial Ethernet, με το πρώτο να είναι και το πιο ευρέως διαδεδομένο. Αξίζει να σημειωθεί ότι σε συστήματα SCADA εγκατεστημένα εντός Ηνωμένων Πολιτειών Αμερικής, κυριαρχεί το πρωτόκολλο ModBus, αν και τα τελευταία χρόνια γίνεται μία, ανεπιτυχής προς το παρόν, προσπάθεια να επικρατήσει το Profibus παγκοσμίως, για λόγους συμβατότητας και ευκολίας.

Τέλος για την περίπτωση των «έξυπνων σπιτιών» , έχει αναπτυχθεί ένα πολύ ενδιαφέρον και ευέλικτο δίκτυο στις αρχές των παραπάνω, το Instabus EIB,

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



Εικόνα 4 Δικτύωση ενός συστήματος SCADA

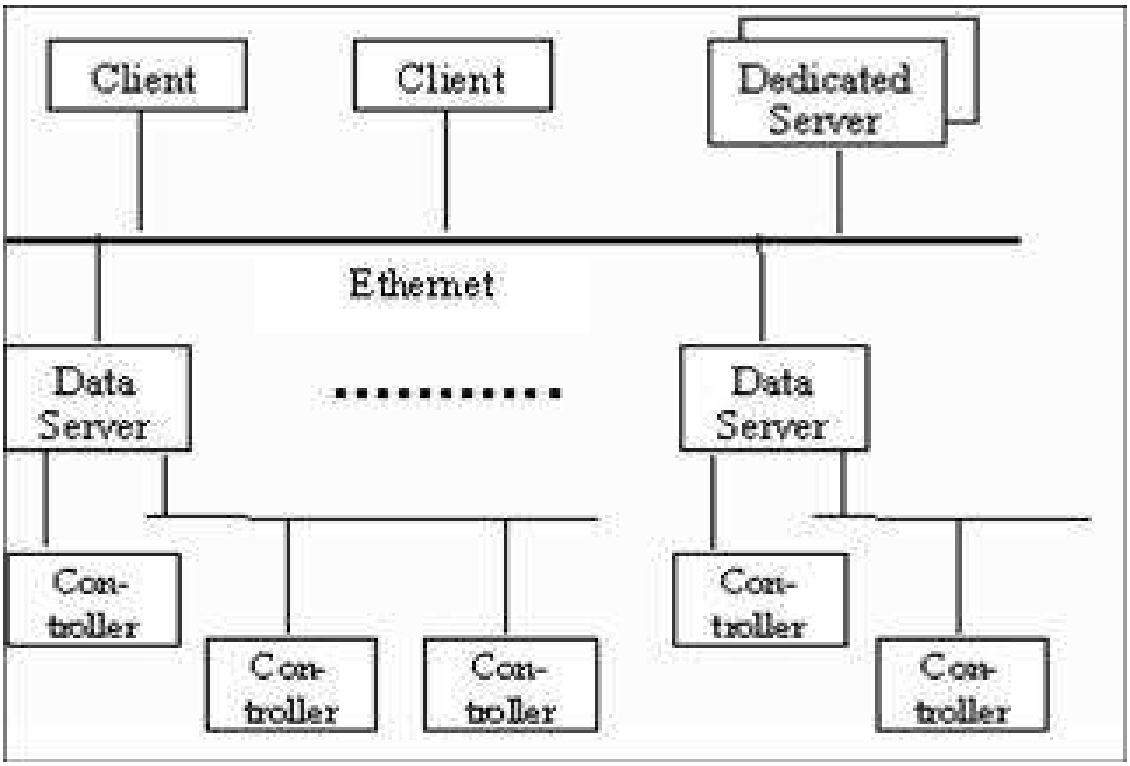
.

1.3 Αρχιτεκτονική υλικού :

Τα βασικά στρώματα σε ένα σύστημα SCADA διακρίνονται ως εξής:

- Το «στρώμα Client (client layer)» που ικανοποιεί την αλληλεπίδραση μηχανών-ανθρώπων
- Το «στρώμα data server» που χειρίζεται τα περισσότερα από τα στοιχεία διαδικασίας που ελέγχουν τις δραστηριότητες.
- Οι Data servers επικοινωνούν με τις συσκευές στον τομέα μέσω των ελεγκτών διαδικασίας. Οι ελεγκτές διαδικασίας, π.χ. PLCs, συνδέονται με τους data servers είτε άμεσα είτε μέσω δικτύων είτε μέσω fieldbuses που είναι ιδιόκτητα (π.χ. Siemens H1), ή μη

ειδικευμένα (π.χ. Profibus). Οι Data servers συνδέονται ο ένας με τον άλλον και με τους client servers μέσω του τοπικού LAN Ethernet. Οι Data servers και οι client servers είναι πλατφόρμες NT αλλά για πολλά προϊόντα οι client servers παλιά μπορούσαν επίσης να είναι W95 μηχανές. Η εικόνα 5 παρουσιάζει τα χαρακτηριστικά της αρχιτεκτονικής υλικού.

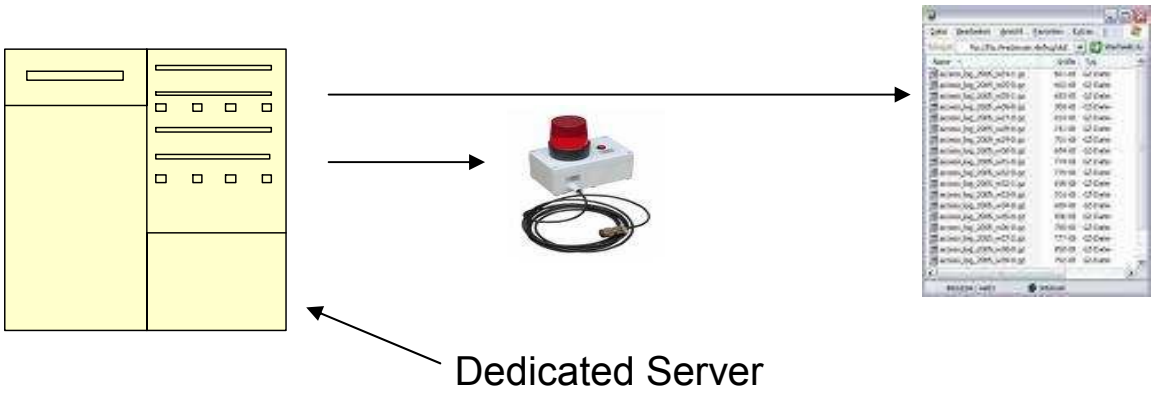


Εικόνα 5 Χαρακτηριστικά Αρχιτεκτονικής Υλικού

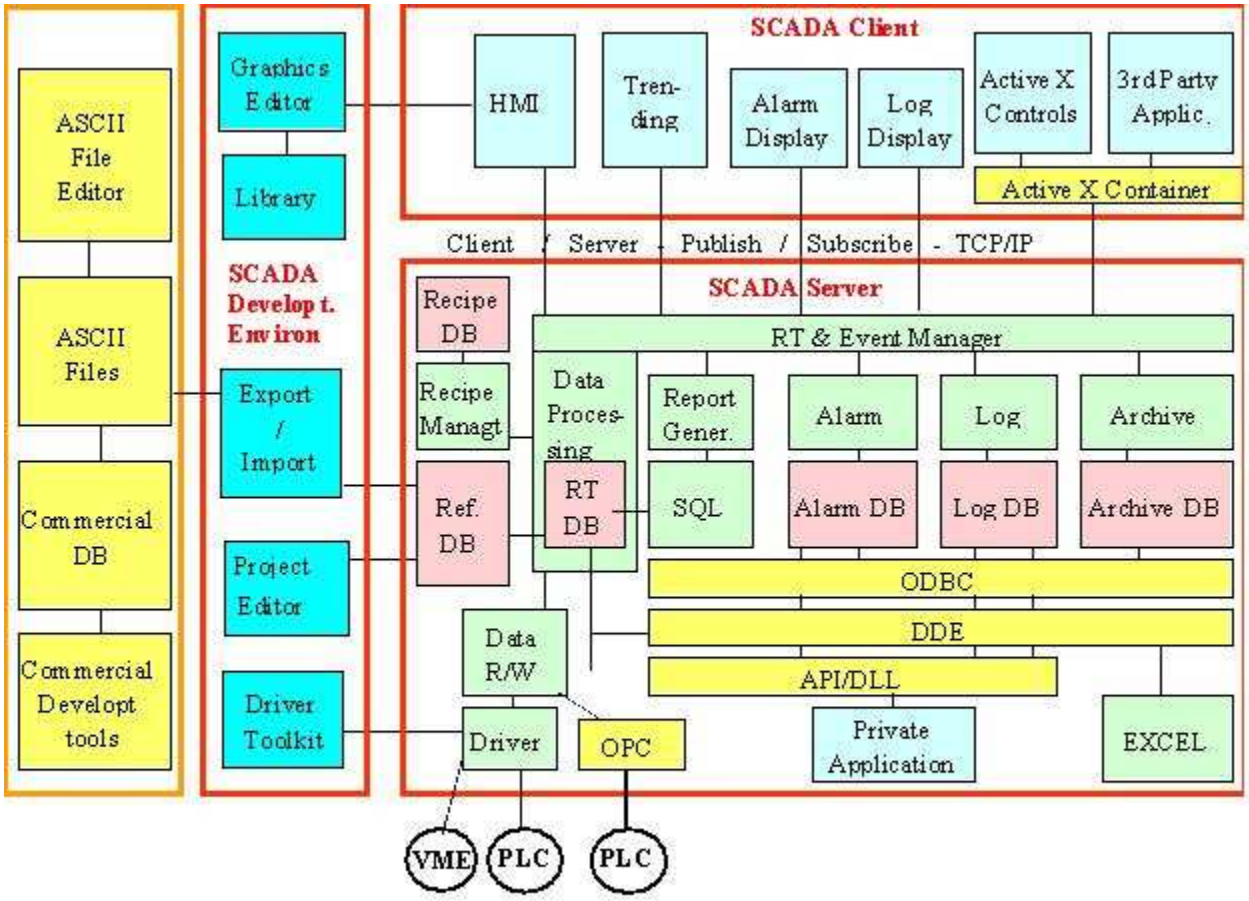
1.4 Αρχιτεκτονική Λογισμικού

Τα προϊόντα είναι multi-tasking και είναι βασισμένα σε μια πραγματικού χρόνου βάση δεδομένων (RTDB) που βρίσκεται σε έναν ή περισσότερους κεντρικούς υπολογιστές (Servers). Οι κεντρικοί υπολογιστές είναι αρμόδιοι για την απόκτηση στοιχείων (data acquisition) και τον χειρισμό (για παράδειγμα ελεγκτές, έλεγχος συναγερμού, υπολογισμοί, logging και αρχειοθέτηση ή archiving) σε ένα σύνολο παραμέτρων που συνδέονται.

Εντούτοις, είναι δυνατό να έχουμε Dedicated Servers (όπως στην εικόνα 6) για συγκεκριμένες διεργασίες, όπως για παράδειγμα καταγραφής ιστορικού, datalogger, χειριστής συναγερμών. Η εικόνα 7 παρουσιάζει αρχιτεκτονική SCADA που είναι γενική για τα evaluated (αποτιμημένα) προϊόντα.



Εικόνα 6 Παράδειγμα Dedicated Servers



Εικόνα 7 Γενική Αρχιτεκτονική Λογισμικού SCADA

1.5 Επικοινωνίες

Εσωτερική επικοινωνία

Η επικοινωνία server - Client και server-server είναι γενικά βάσεως publish - subscribe και event -driven και χρησιμοποιεί πρωτόκολλο TCP/IP, δηλ., μια εφαρμογή Client προσυπογράφει(subscribes) σε μια παράμετρο που τη διαχειρίζεται μια εφαρμογή του Server και αλλάζει μόνο σε εκείνη την παράμετρο που επικοινωνεί με την εφαρμογή Client.

Πρόσβαση στις συσκευές

Οι Data servers επιλέγουν τους ελεγκτές σε ένα καθορισμένο από το χρήστη ποσοστό. Το ποσοστό επιλογής μπορεί να είναι διαφορετικό για διαφορετικές παραμέτρους. Οι ελεγκτές περνούν τις ζητούμενες

παραμέτρους στους data servers. Η χρονική σφράγιση των παραμέτρων διαδικασίας εκτελείται χαρακτηριστικά στους ελεγκτές και αυτό το time - stamp (σφραγίδα με ημερομηνία και ώρα) αναλαμβάνεται από τον κεντρικό υπολογιστή(server).
Τα προϊόντα παρέχουν τους οδηγούς επικοινωνίας για το μεγαλύτερο μέρος των βασικών PLCs και των ευρέως χρησιμοποιημένων field -buses, π.χ., Modbus. Από τα τρία fieldbuses που συστήνονται στο Κέντρο Πυρηνικών Μελετών και Ερευνών (CERN), και το Profibus και το Worldfip υποστηρίζονται αλλά το CANbus συχνά όχι . Μερικοί από τους οδηγούς είναι βασισμένοι στα προϊόντα τρίτων (π.χ., κάρτες Applicom) και έχουν συμπληρωματικό κόστος.
Ένας ενιαίος data server μπορεί να υποστηρίξει τα πολλαπλά πρωτόκολλα επικοινωνιών: μπορεί γενικά να υποστηρίξει τόσα πρωτόκολλα όσα και τα slots των interface cards.
Η προσπάθεια που απαιτείται για να αναπτύξει τους νέους οδηγούς είναι χαρακτηριστικά στη σειρά 2-6 εβδομάδων ανάλογα με την πολυπλοκότητα και την ομοιότητα με τους υπάρχοντες οδηγούς και ένα κουτί εργαλείων ανάπτυξης οδηγών παρέχεται για αυτό.

1.6 Διασύνδεση (interfacing)

Application Interfaces / Openness (ειλικρίνεια)
Η παροχή λειτουργίας OPC (**O**LE* for **P**rocess **C**ontrol) client για SCADA για να έχει πρόσβαση στις συσκευές αναπτύσσεται κατά τρόπο ανοικτό και τυποποιημένο. Ακόμα φαίνεται να υπάρχει μια έλλειψη συσκευών/ελεγκτών, οι οποίοι παρέχουν το λογισμικό κεντρικών υπολογιστών OPC, αλλά αυτό βελτιώνεται γρήγορα δεδομένου ότι οι περισσότεροι από τους παραγωγούς των ελεγκτών περιλαμβάνονται ενεργά στην ανάπτυξη αυτών των προτύπων. Το OPC έχει αξιολογηθεί από το Κέντρο Πυρηνικών Μελετών και Ερευνών (CERN) .

Τα προϊόντα παρέχουν επίσης

- Σύνδεση βάσεων δεδομένων (ODBC)
- Δυνατότητα εισαγωγών/εξαγωγής ASCII για τα configuration data,

- Βιβλιοθήκη APIs που υποστηρίζει C, C++, και Visual Basic (VB). Το API (application programming interface) συχνά δεν παρέχει την πρόσβαση στα εσωτερικά χαρακτηριστικά γνωρίσματα του προϊόντος όπως ο χειρισμός συναγερμών, η υποβολή έκθεσης, η τάση, κ.λπ.

Τα προϊόντα PC παρέχουν την υποστήριξη για τα πρότυπα της Microsoft όπως η δυναμική ανταλλαγή στοιχείων Dynamic Data Exchange (DDE) που επιτρέπει για παράδειγμα να απεικονίσει τα στοιχεία δυναμικά σε έναν υπολογισμό με λογιστικό φύλλο (spreadsheet) EXCEL, δυναμική σύνδεση βιβλιοθήκης (DLL) και object ενσωμάτωση (OLE).

Βάση δεδομένων

Τα configuration data αποθηκεύονται σε μια βάση δεδομένων που συγκεντρώνεται λογικά αλλά φυσικά διανέμεται και είναι γενικά proprietary format(ιδιόκτητης μορφής) .

Για λόγους απόδοσης, η RTDB (real -time database) κατοικεί στη μνήμη των Servers και είναι επίσης ιδιόκτητης μορφής.

Το σχήμα αρχείων και logging είναι συνήθως επίσης ιδιόκτητο για λόγους απόδοσης, αλλά μερικά προϊόντα υποστηρίζουν το logging σε ένα συγγενικό σύστημα διαχείρισης βάσεων δεδομένων (RDBMS) σε ένα πιο αργό ποσοστό είτε άμεσα είτε μέσω μιας διεπαφής ODBC.

1.7 Εξελιξιμότητα

Η εξελιξιμότητα γίνεται κατανοητή ως δυνατότητα να επεκταθεί το βασισμένο σε SCADA σύστημα ελέγχου με την προσθήκη περισσότερων μεταβλητών διαδικασίας, πιο ειδικευμένων Servers (όπως για το συναγερμό που χειρίζεται) ή περισσότερων πελατών. Τα προϊόντα επιτυγχάνουν την εξελιξιμότητα με τη σύνδεση πολλών κεντρικών υπολογιστών(Data Servers) με πολλούς ελεγκτές. Κάθε κεντρικός υπολογιστής έχει τη βάση δεδομένων διαμόρφωσης και την RTDB του και είναι αρμόδιος για το χειρισμό ενός υποσυνόλου των μεταβλητών διαδικασίας απόκτησης (acquisition), χειρισμό συναγερμών, αρχειοθέτηση.

1.8 Λειτουργικότητα

1.8.1 Έλεγχος Πρόσβασης

Οι χρήστες διατίθενται σε ομάδες, στις οποίες έχουν καθοριστεί τα προνόμια πρόσβασης ανάγνωση-γραφή στις παραμέτρους διαδικασίας στο σύστημα και συχνά επίσης στη συγκεκριμένη λειτουργικότητα προϊόντων.

1.8.2 Καταγραφή/Αρχειοθέτηση (logging/archiving)

Οι όροι logging/archiving χρησιμοποιούνται συχνά για να περιγράψουν την ίδια δυνατότητα. Εντούτοις, η καταγραφή μπορεί να θεωρηθεί ως μεσοπρόθεσμη αποθήκευση των στοιχείων όσον αφορά το δίσκο, ενώ η αρχειοθέτηση είναι μακροπρόθεσμη αποθήκευση των στοιχείων είτε στο δίσκο είτε σε ένα άλλο μέσο μνήμης .Το logging εκτελείται χαρακτηριστικά σε κυκλική βάση, δηλ., μόλις επιτευχθεί ένα ορισμένο μέγεθος αρχείων, ή ένας αριθμός σημείων τα στοιχεία επικαλύπτονται.

Η καταγραφή των στοιχείων μπορεί να εκτελεστεί ή σε μια καθορισμένη συχνότητα, ή να αρχίσει μόνο εάν τα values αλλάξουν ή όταν εμφανίζεται ένα συγκεκριμένο προκαθορισμένο γεγονός (event). Τα καταγραμμένα στοιχεία μπορούν να μεταφερθούν σε ένα αρχείο μόλις το log file είναι πλήρες. Το καταγραμμένο στοιχείο είναι σφραγισμένο σε σχέση με το χρόνο και μπορεί να φιλτραριστεί όταν αντιμετωπίζεται από έναν χρήστη.

Η καταγραφή των ενεργειών χρηστών εκτελείται γενικά μαζί με είτε έναν χρήστη - ταυτότητα είτε ταυτότητα σταθμών. Υπάρχει συχνά επίσης μια δυνατότητα VCR για να προβάλει τα αρχειοθετημένα στοιχεία.

1.8.3 Παραγωγή Εκθέσεων – Αυτοματοποίηση

Τα Reports μπορούν να συνταχθούν χρησιμοποιώντας ή ερωτήσεις τύπου SQL σε αρχείο ή RTDB ή Log Files. Αν και είναι μερικές φορές δυνατό να ενσωματωθούν τα διαγράμματα EXCEL σε έκθεση, η δυνατότητα «cut και paste» γενικά δεν παρέχεται. Οι εγκαταστάσεις υπάρχουν για να είναι σε θέση να παράγουν αυτόματα, να τυπώσουν και να αρχειοθετήσουν τις εκθέσεις

Αυτοματοποίηση

- Η πλειοψηφία των προϊόντων επιτρέπει να προκαλούνται αυτόματα ενέργειες από τα γεγονότα. Μια γλώσσα scripting που παρέχεται από τα προϊόντα SCADA επιτρέπει την εφαρμογή αυτών των ενεργειών.
- Υποστηρίζεται η έννοια των συνταγών, όπου μια συγκεκριμένη ρύθμιση ενός συστήματος μπορεί να σωθεί σε ένα αρχείο και να ξαναφορτωθεί έπειτα εν ευθέτω χρόνω.
- Υποστηρίζεται επίσης η αλληλουχία με την οποία, όπως το όνομα δείχνει, είναι δυνατό να εκτελεστεί μια πιο σύνθετη ακολουθία ενεργειών σε μια ή περισσότερες συσκευές.

1.9 Εργαλεία Ανάπτυξης Εφαρμογής

Τα ακόλουθα εργαλεία ανάπτυξης παρέχονται ως πρότυπα:

- Ένας graphic editor, με τυποποιημένες εγκαταστάσεις σχεδίων.
- Εργαλείο διαμόρφωσης βάσεων δεδομένων (συνήθως μέσω των προτύπων παραμέτρου). Είναι γενικά δυνατό να εξαχθούν τα στοιχεία στα αρχεία ASCII ώστε να δεχτούν επεξεργασία μέσω ενός συντάκτη ASCII ή Excel .
- Γλώσσα Script (Scripting Language)
- Ένα Application Program Interface (API) που υποστηρίζει C, C++, VB
- Ένα πακέτο εργαλείων ανάπτυξης οδηγών (Driver Development Toolkit) για να αναπτύξει τους οδηγούς για το υλικό που δεν υποστηρίζεται από το προϊόν SCADA.

1.10 Πλεονεκτήματα – Εφαρμογές συστημάτων SCADA

Το σύστημα SCADA προσφέρει τα ακόλουθα πλεονεκτήματα

• Άμεσο έλεγχο καναλιού και πίεση ροών
• Η ακριβής χρονική στιγμή της παράδοσης του νερού, ακρίβεια κατά την εφαρμογή του νερού άρδευσης

• Χαμηλή επένδυση κεφαλαίου για τις συνιστώσες
• Λειτουργία με ηλεκτρική ενέργεια(ΔΕΗ), την ενέργεια της μπαταρίας, ή η ηλιακή ενέργεια, ή ένα συνδυασμό των τριών
• Εύκολη εγκατάσταση και διατήρηση
• Μπορεί να αυξήσει το επίπεδο της πολυπλοκότητας την πάροδο του χρόνου, επιτρέποντας στο χρήστη να επεκτείνει την ευκολία και την άνεση του χρήστη

Εφαρμογές συστημάτων SCADA:

- Μέτρηση & έλεγχο στάθμης δεξαμενών & σιλό.
- Έλεγχο & επιτήρηση συμπιεστών, μοτέρ, κλπ.
- Έλεγχο & επιτήρηση γραμμών παραγωγής.
- Έλεγχο & επιτήρηση μηχανών παραγωγής.
- Έλεγχο & επιτήρηση μηχανών εκτέλεσης συνταγών
- Ποιοτικό έλεγχο παραγωγής
- Έλεγχο & επιτήρηση φορτηγών (καύσιμα, θερμοκρασία, υγρασία, διαδρομή)
- Έλεγχο & επιτήρηση πλοίων.
- Αυτοματοποίηση διαδικασιών.
- Έλεγχο και επιτήρηση θαλάμων ψύξης - θέρμανσης
- Έλεγχο και επιτήρηση Κλιματισμού κτιρίων, ξενοδοχείων.
- Αναλυτική κοστολόγηση προϊόντων & υπηρεσιών

Παρέχει την δυνατότητα ομαδοποίησης των σημείων ελέγχου για ευκολότερη αναζήτηση και έλεγχο. Παρουσιάζει τις μετρήσεις ταυτόχρονα σε γραφική και σε αριθμητική μορφή. Ο τρόπος γραφικής παρουσίασης των δεδομένων είναι παραμετρικός, με δυνατότητα επιλογής της μορφής από τον χρήστη. Είναι φιλικά σχεδιασμένο για να μπορούν να το χρησιμοποιήσουν χρήστες χωρίς να διαθέτουν καμία εμπειρία στην τεχνολογία των υπολογιστών.

Κεφάλαιο 2 Τεχνολογίες(software –hardware)

2.1 Εισαγωγή

Για την δημιουργία του συστήματος SCADA ήταν απαραίτητη η χρησιμοποιήση software και hardware. Παρακάτω θα δούμε τις κύριες τεχνολογίες που απαιτήθηκαν για την εφαρμογή.

2.2 Προγραμματιζόμενοι Λογικοί Ελεγκτές-PLC

2.2.1 Γενικά για τα PLC

Τα PLC **(programmable logic controller)** από μόνα τους είναι ουδέτερες συσκευές αφού δεν είναι από πριν κατασκευασμένες για μια συγκεκριμένη εφαρμογή. Κάθε φορά , ανάλογα με τις απαιτήσεις της εκάστοτε εγκατάστασης προγραμματίζονται να κάνουν κάποια ενέργεια. Τα PLC βασίζονται στην ψηφιακή τεχνολογία, γεγονός που σημαίνει ότι χρησιμοποιούν το δυαδικό σύστημα

Υπάρχουν διάφοροι τρόποι προγραμματισμού που ποικίλουν ακριβώς γιατί ποικίλουν και τα επίπεδα γνώσης και εμπειριών του κάθε προγραμματιστή.

Προγραμματίζεται σε γλώσσα προγραμματισμού Ladder.

Οι ουσιαστικές διαφορές είναι στο τι βλέπουμε στην οθόνη του υπολογιστή μας ,αφού το τελικό αποτέλεσμα είναι πάντα ένα και το αυτό . Οι διάφορες γλώσσες μετατρέπονται σε γλωσσά μηχανής MC7 (Machine Code 7) κατά την μεταφορά του προγράμματος από την συσκευή προγραμματισμού στο PLC.

Εικόνα 8 PLC & input/output arrangements

Υπάρχουν τρεις τυποποιημένες μορφές προγραμματισμού που έχουν επικρατήσει διεθνώς:

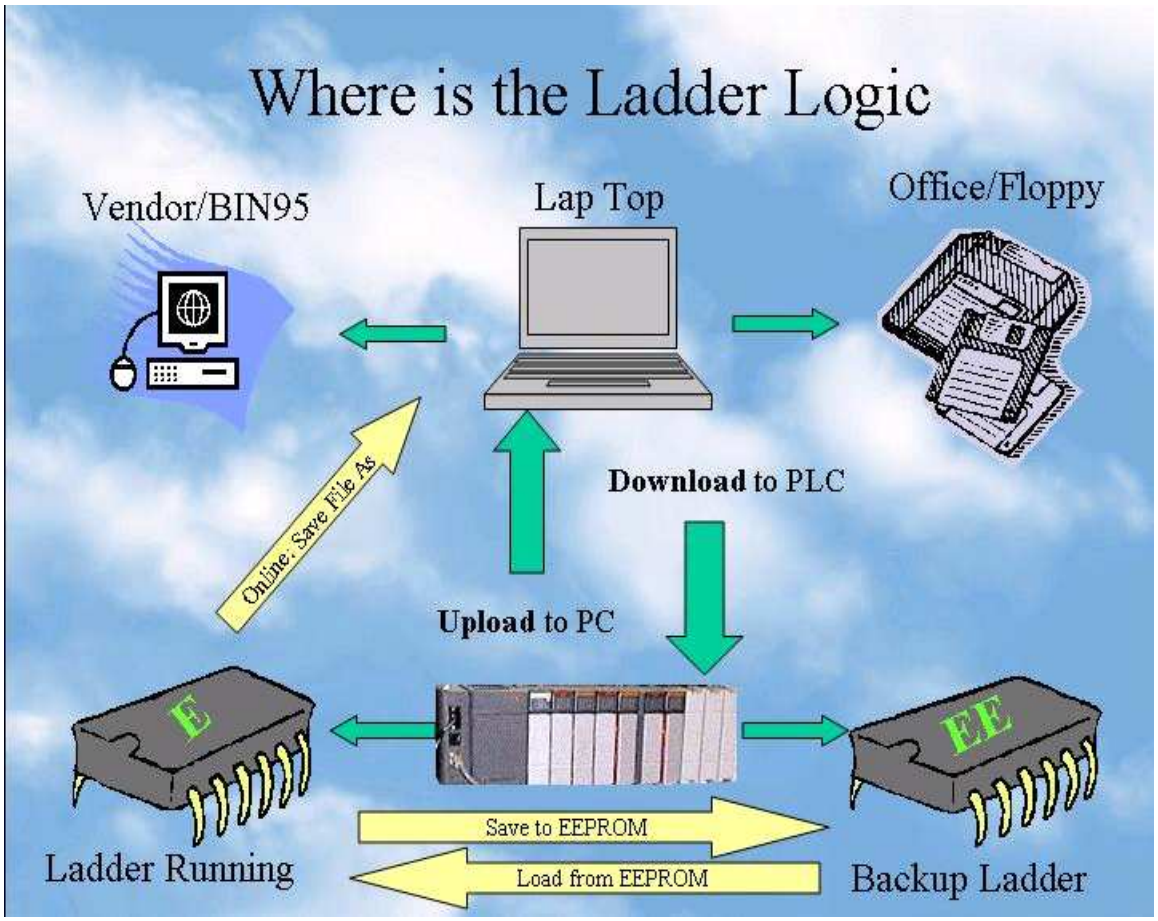
- Η λίστα εντολών **STL**
- Σχέδιο επαφών **LAD**
- Διάγραμμα λογικών πυλών **FDB**

2.2.2 Δομή ενός PLC

Η επιλογή ενός προγραμματιζόμενου λογικού ελεγκτή εξαρτάται κυρίως από το πλήθος των στοιχείων που δίνουν εντολή σε αυτόν (είσοδοι) και το

πλήθος των στοιχείων που δέχονται εντολή από αυτόν (έξοδοι), καθώς και από το πλήθος των λειτουργιών που απιτείται να κάνει ο αυτοματισμός.
Ανεξάρτητα όμως από τον τύπο και το μέγεθος σε κάθε ελεγκτή συναντάμε τα εξής στοιχεία :

- Πλαίσιο για τοποθέτηση των μονάδων. Σε αυτό είναι ενσωματωμένο το σύστημα ζυγών μέσω των οποίων επικοινωνούν οι διάφορες μονάδες μεταξύ τους για την ανταλλαγή πληροφοριών και για την τροφοδοσία τους.
- Μονάδα τροφοδοσίας. Χρησιμεύει για να δημιουργηθούν από την τάση του δικτυού οι απαραίτητες εσωτερικές τάσεις για την τροφοδοσία αποκλειστικά των ηλεκτρικών στοιχείων που υπάρχουν μέσα στον ελεγκτή.
- Κεντρική μονάδα (CPU) με τον μικροεπεξεργαστή και την μνήμη για το πρόγραμμα. Η CPU έχει τα εξής χαρακτηριστικά: ενσωματωμένη RAM εργασίας, ενσωματωμένη RAM φορτώματος και εξωτερική FLASH EPROM φορτώματος που επεκτείνει την ενσωματωμένη.
- Μονάδες εισόδων.
- Μονάδες εξόδων. Τα καλώδια που έρχονται από τα αισθητήρια της παραγωγικής διαδικασίας (τερματικοί, μπουτόν, διακόπτες) συνδέονται στις υποδοχές των μονάδων εισόδων του ελεγκτή. Αντίστοιχα, τα καλώδια που πηγαίνουν προς τα ρελέ ισχύος, βαλβίδες, λυχνίες συνδέονται στις υποδοχές των μονάδων εξόδου του ελεγκτή.



Εικόνα 9 Λειτουργία ενός PLC

2.2.3 Πλεονεκτήματα PLC

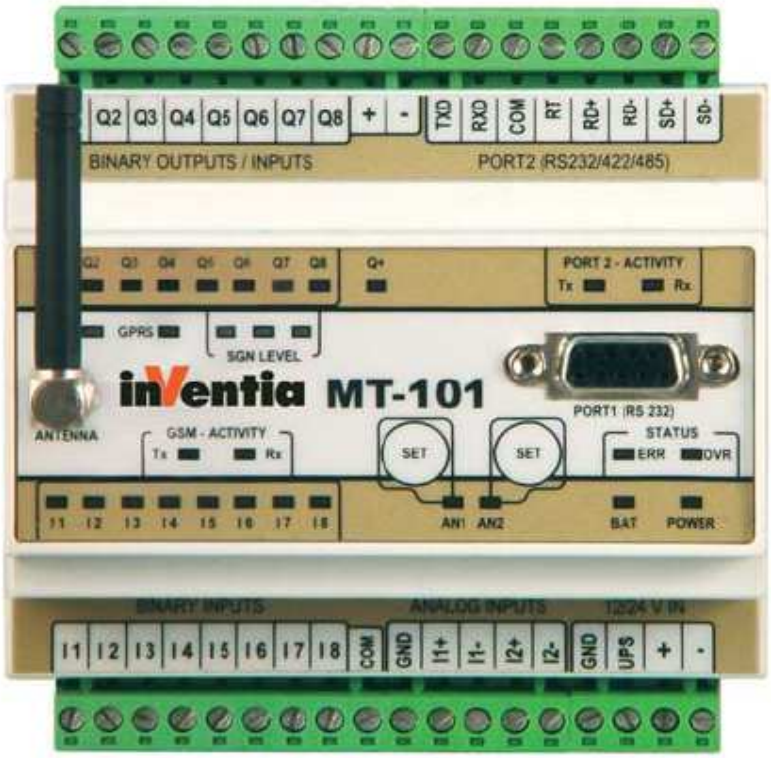
Τα σημαντικότερα πλεονεκτήματα των PLC είναι :

- Μειωμένο κόστος υλοποίησης του αυτοματισμού
- Χρόνος υλοποίησης του αυτοματισμού
- Ελαχιστοποίηση κόστους συντήρησης
- Μεγάλη ευελιξία σε τροποποιήσεις του αυτοματισμού
- Μεγάλες δυνατότητες επέκτασης του αυτοματισμού
- Ευκολία δημιουργίας έξυπνων/πολύπλοκων διεργασιών
- Δυνατότητα σύνδεσης με κεντρικό υπολογιστικό σύστημα ή το εταιρικό δίκτυο
- Καταλαμβάνει ελάχιστο χώρο

- Εύκολος προγραμματισμός/έλεγχος λειτουργίας
- Γρηγορότερη παράδοση αυτοματισμού
- Οικονομία στην κατανάλωση ενέργειας

2.3 Χαρακτηριστικά PLC Μονάδας MT-101

Η μονάδα τηλεμετρίας MT-101 είναι μία δυναμική συσκευή που συνδυάζει το GPRS modem, PLC ελεγκτή, καταγραφέα δεδομένων και μετατροπέα μετάδοσης πρωτοκόλλου.



Εικόνα 10 MT-101

Η μονάδα τηλεμετρίας προσφέρει τα χαρακτηριστικά του ελεύθερα προγραμματιζόμενου ελεγκτή PLC εξοπλισμένο με ενσωματωμένο GSM/GPRS modem, καταγραφέα δεδομένων και απομονωμένες θύρες RS232 /422/485 για επικοινωνίες με τον έξω κόσμο. Η μονάδα επιτρέπει την

κατασκευή σύγχρονων ασύρματων συστημάτων για την εποπτεία, παρακολούθηση, μέτρηση, διάγνωση και έλεγχο που βασίζονται στην τεχνολογία μετάδοσης πακέτων GPRS.

Το σημαντικό χαρακτηριστικό της μονάδας είναι η δυνατότητα μεταβίβασης δεδομένων ανά συγκεκριμένα χρονικά διαστήματα είτε σαν ένα μηχανισμό καθοδηγούμενο από γεγονότα (για παράδειγμα όταν οι δυαδικές είσοδοι/έξοδοι αλλάζουν την κατάσταση τους ή όταν το σήμα στην αναλογική είσοδο αλλάζει από συγκεκριμένο βαθμό). Η μονάδα παρέχει έναν ενσωματωμένο καταγραφέα δεδομένων των 100 ms του χρόνου ανάλυσης(λειτουργικότητα RTU).

Η μονάδα είναι πλήρως διαμορφώσιμη και προγραμματιζόμενη από τις επιλογές του χρήστη,έχει φιλικό περιβάλλον λογισμικού MT Manager και η διασύνδεσή του μπορεί να γίνει είτε τοπικά μέσω σειριακής θύρας είτε απομακρυσμένα μέσω GPRS δικτύου.

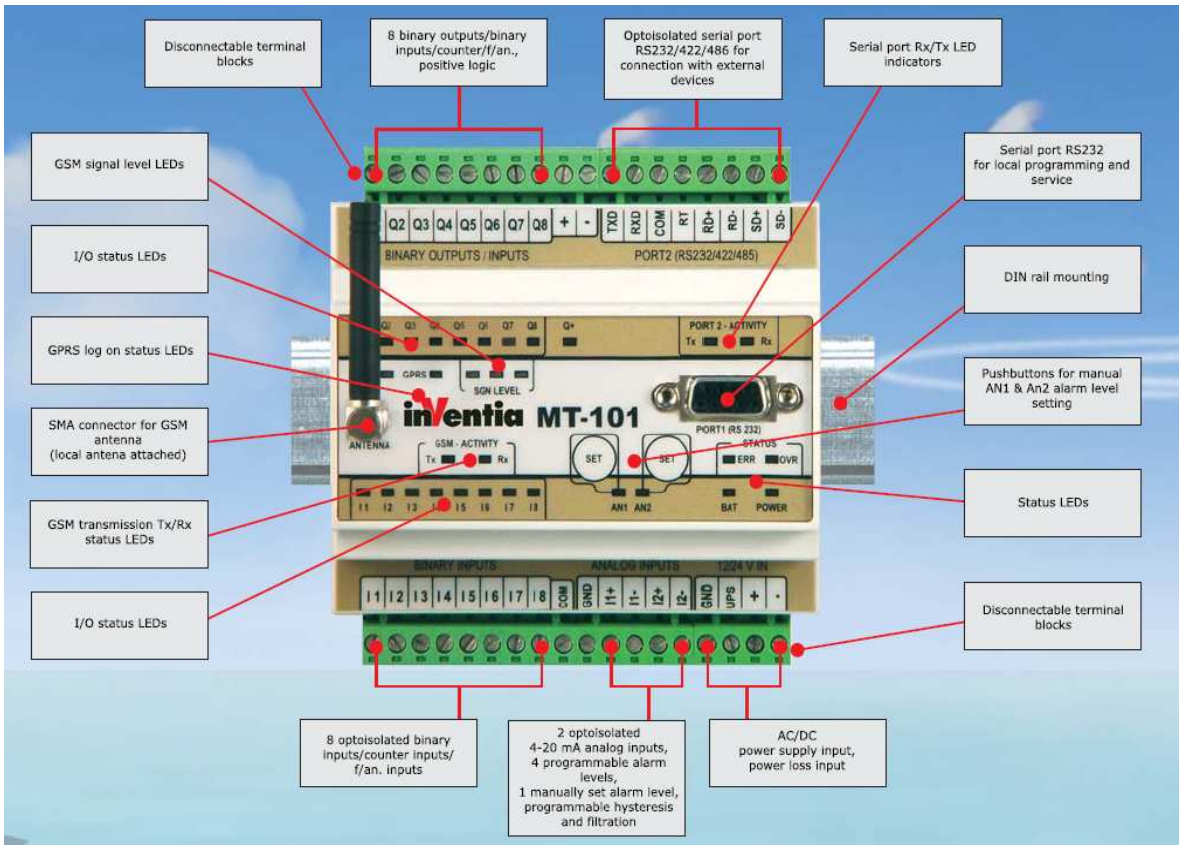
Η μονάδα επιτρέπει την άμεση σύνδεση των εξωτερικών σημάτων στην συσκευή εισόδου εξόδου. Οι πόροι μπορούν να επεκταθούν με εξωτερικές μονάδες (πχ οι συσκευές EX που κατασκευάζονται από την inventia, συσκευές plc ή μονάδες εισόδου εξόδου) που λειτουργούν σύμφωνα με το πρωτόκολλο modbus σε κατάσταση slave. Η μονάδα εξασφαλίζει εύκολη ενσωμάτωση στο δίκτυο GPRS από οποιοδήποτε είδος έξυπνων συσκευών (όπως plc, όργανα μέτρησης, πάνελ φορέα) που είναι εξοπλισμένα με σειριακή θύρα επικοινωνίας RS-232/422/485.

Σε περίπτωση που η συσκευή υποστηρίζει το πρωτόκολλο MODBUS RTU οι προηγμένες λειτουργίες των τοπικών επικοινωνιών όπως επεξεργασία δεδομένων, καταγραφή δεδομένων και αυτόματες εκδηλώσεις γεγονότων με GPRS μετάδοση, είναι διαθέσιμες. Ο χρήστης δεν είναι διατεθειμένος να γνωρίζει ούτε τις λεπτομέρειες μετάδοσης GPRS, τις εντολές ελέγχου AT, τα πρωτόκολλα διαπραγμάτευσης περιόδου, τους κανόνες ελέγχου της δραστηριότητας της περιόδου, την αλλαγή της διαπραγμάτευσης μετά από μικρή διακοπή, τους περιορισμούς της ασφάλειας του δικτύου πρόσβασης, τους κανόνες της ακεραιότητας των δεδομένων ούτε την επικύρωση παράδοσης όταν διαβιβάζονται τα πακέτα. Επιπλέον η επικοινωνία των εξωτερικών συσκευών με την μονάδα δεν απαιτεί αλλαγές των ρυθμίσεων διαμόρφωσης ή στο λογισμικό εφαρμογής.

Η MT-101 μπορεί να λειτουργήσει ως μια τοπική συσκευή Master η οποία περιοδικά λειτουργεί σαν εξωτερική συσκευή για να πάρει τους πόρους που ορίζονται από τον χρήστη (είσοδοι, έξοδοι, αναλογικές εισόδους, καταχωρητές, εσωτερικές "σημαίες"). Οι πηγές αυτές αντικατοπτρίζονται στη μνήμη της μονάδας MT-101 επιτρέποντας την ανίχνευση συναγερμού, την αλλαγή κατάστασης, τις αναλογικές μεταβολές τιμών, τη συνάντηση λογικών συνθηκών και όλα αυτά με την απευθείας χρήση ή με τον υπολογισμό τιμών. Τα δεδομένα μεταφέρονται μέσω GPRS σύμφωνα με τους κανόνες που ορίζονται από τον χρήστη:

- Σαν μια απάντηση σε ένα ερώτημα
- Αυτόνομα από τη μονάδα προσδιοριζόμενα σε συγκεκριμένα χρονικά διαστήματα (poll processing)
- Αυθόρμητα από τη μονάδα σαν αποτέλεσμα ενός προκαθορισμένου γεγονότος (ειδοποίηση, αλλαγή κατάστασης, μια σημαντική αλλαγή σε αναλογική τιμή, event-driven)

Μια μεταφορά που προκαλείται από γεγονότα καθιστά δυνατή τη δημιουργία ασύρματων συστημάτων σχεδόν απεριορίστου μεγέθους και απόστασης τα οποία έχουν συγκεκριμένη χρονική ανάλυση, σύντομο χρόνο απόκρισης (2-3 δευτερόλεπτα) και αποδοτική χρήση της μετάδοσης GPRS.



Εικόνα 11 Χαρακτηριστικά MT-101

MT-101 –Ιδιαίτερα χαρακτηριστικά

- Γαλβανικά απομονωμένες είσοδοι / έξοδοι με φιλτράρισμα
- Λειτουργικότητα καθυστέρησης και ορίων συναγερμού
- Είσοδοι με 32 bit χωρητικότητα
- Τοπικός έλεγχος προγράμματος
- Αυτόνομη είσοδος σε GPRS δίκτυο και ανάκτηση της συνεδρίασης
- Αυτοματοποιημένο σύστημα διάγνωσης με πλούσιο σύνολο διαγνωστικών λυχνιών
- Δυνατή λειτουργία σε διαφανή κατάσταση ως ασύρματη σειριακή θύρα
- Λειτουργικότητα από ένα τοπικό Master για εξωτερικές συσκευές

- Αντικατοπτρισμός των εξωτερικών συσκευών με επεξεργασία δεδομένων και ανάλυση ικανοτήτων
- Ρολόι πραγματικού χρόνου
- Καταγραφέας δεδομένων 100 ms χρόνου ανάλυσης
- Μετατροπέας μεταφοράς πρωτοκόλλου
- Δρομολογητής πακέτων
- Προστασία ενάντια πλαισίων από μη εξουσιοδοτημένους χρήστες
- Έλεγχος αξιοπιστίας δεδομένων και επιβεβαίωσης παράδοσης πλαισίων
- Απομακρυσμένη ρύθμιση, προγραμματισμός και αναβάθμιση firmware μέσω δικτύου GPRS
- Ευρύ φάσμα τροφοδοσίας τάσεων (9…30 VDC or 24 VAC).

2.4 Power Meter – 710

Το PM-710 είναι μια συσκευή η οποία χρησιμοποιείται στην εγκατάσταση του SCADA συστήματος. Συνδέεται με την PLC μονάδα μέσω ModBus πρωτοκόλλου επικοινωνίας. Από την συσκευή αυτή παίρνουμε δίαφορες τιμές όπως τάσεις τροφοδοσίας αντλιοστασίου, εντάσεις τροφοδοσίας αντλιοστασίου, χρόνο λειτουργίας, ισχύς λειτουργίας και δίαφορες άλλες.

Εικόνα 12 Power Meter 710

2.5 ModBus

Το Modbus είναι ένα σειριακό πρωτόκολλο επικοινωνίας που δημοσίευσε η Modicon το 1979 για χρήση με προγραμματιζόμενους λογικούς ελεγκτές της (PLC). Είναι ένα ουσιαστικό πρότυπο στα πρωτόκολλα επικοινωνίας στη βιομηχανία, και είναι πλέον το πιο συχνό διαθέσιμο μέσο για σύνδεση βιομηχανικών ηλεκτρονικών συσκευών. Οι κύριοι λόγοι για την εκτεταμένη χρήση του Modbus έναντι άλλων πρωτοκόλλων επικοινωνίας είναι:

- Μπορεί να χρησιμοποιηθεί για ανάπτυξη σχετικά εύκολων βιομηχανικών δικτύων
- Είναι ένα ελεύθερο πρωτόκολλο και διανέμεται δωρεάν
- Μετακινεί τα ακατέργαστα bits ή τις λέξεις χωρίς περιορισμούς για τους χρήστες

Το Modbus επιτρέπει την επικοινωνία μεταξύ πολλών συσκευών που συνδέονται στο ίδιο δίκτυο, για παράδειγμα, ενός συστήματος που μετρά τη θερμοκρασία και την υγρασία και ανακοινώνει τα αποτελέσματα σε έναν υπολογιστή. Χρησιμοποιείται συχνά για την σύνδεση ενός εποπτικού υπολογιστή με ένα απομακρυσμένο τερματικό μονάδας (RTU) είτε για εποπτικό έλεγχο είτε για συλλογή δεδομένων (SCADA συστήματα).
Μικροί και μεγάλοι προμηθευτές, οι τελικοί χρήστες, προγραμματιστές ανοικτού κώδικα, οι εκπαιδευτικοί και άλλοι ενδιαφερόμενοι μπορούν να χρησιμοποιήσουν το πρωτόκολλο Modbus.

2.5.1 Επικοινωνία και συσκευές

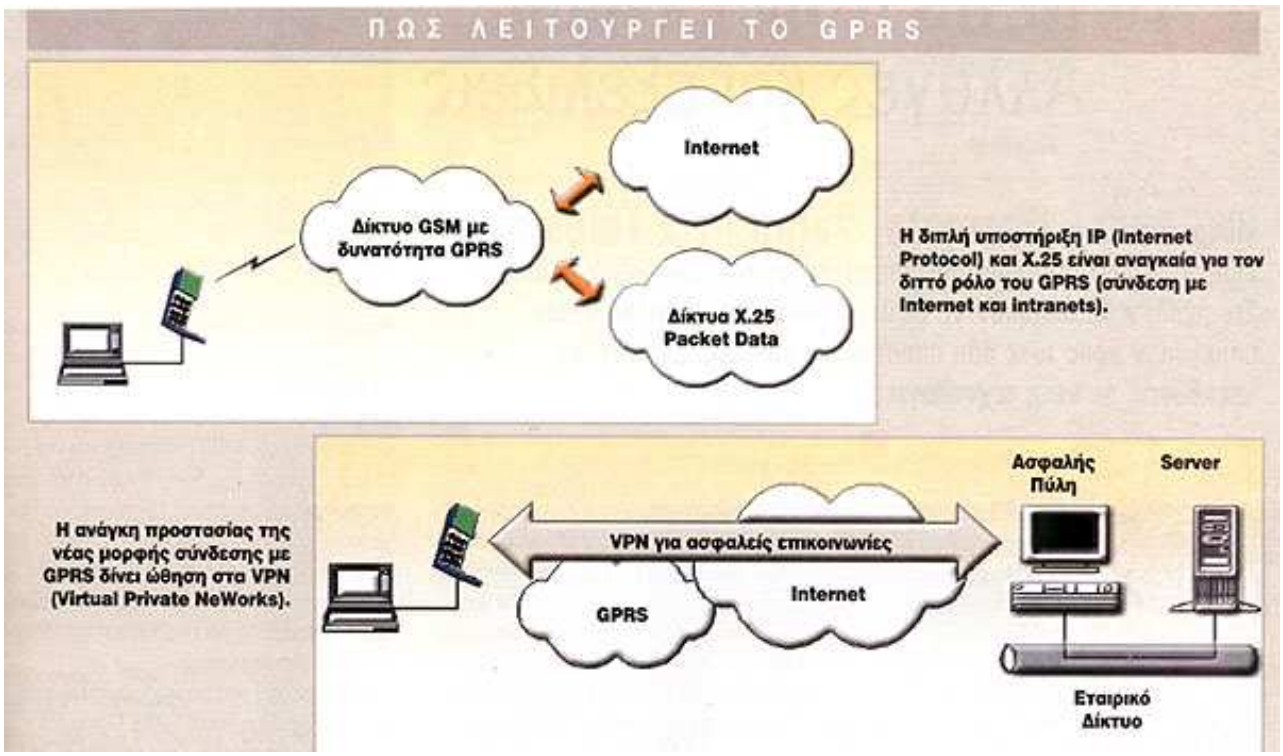
Κάθε συσκευή που σκοπεύει να επικοινωνήσει με μία άλλη μέσω του Modbus παίρνει μια μοναδική διεύθυνση. Στα σειριακά και MB+ δίκτυα μόνο ο κόμβος που ορίζεται σαν Master μπορεί να εισάγει μία εντολή, αλλά στα Ethernet δίκτυα, κάθε συσκευή μπορεί να στείλει μία εντολή μέσω του πρωτοκόλλου Modbus. Μία εντολή Modbus περιέχει την διεύθυνση της συσκευής που προορίζεται για αυτήν. Μόνο η συγκεκριμένη συσκευή θα ενεργοποιήσει την εντολή, παρόλο που και άλλες συσκευές μπορεί να λαμβάνουν την εντολή. Όλες οι εντολές Modbus περιέχουν έλεγχο πληροφορίας εξασφαλίζοντας ότι φθάνουν μη κατεστραμμένες.
Οι βασικές εντολές Modbus μπορούν να αναθέσουν σε ένα απομακρυσμένο τερματικό μονάδας (RTU) την αλλαγή σε μία τιμή σε έναν από τους καταχωρήτες, τον έλεγχο ή την ανάγνωση μιας θύρας εισόδου εξόδου καθώς και να δώσει εντολή στην συσκευή να στείλει πίσω μία η περισσότερες τιμές που περιλαμβάνονται στους καταχωρητές του. Υπάρχουν πολλά μόντεμ και πύλες (gateways) που υποστηρίζουν το πρωτόκολλο αυτό, καθώς είναι ένα πολύ απλό και συχνά μπορεί να αντιγραφεί.
Μερικά από αυτά έχον σχεδιαστεί ειδικά για αυτό το πρωτόκολλο Διαφορετικές υλοποιήσεις χρησιμοποιούν ενσύρματες, ασύρματες επικοινωνίες ακόμη και δυνατότητα αποστολής και λήψης μηνυμάτων (SMS) ή ακόμη και μέσω GPRS δικτύου. Τα τυπικά προβλήματα που πρέπει να ξεπεράσουν οι σχεδιαστές είναι τα προβλήματα βάσει χρόνου.

2.6 Τεχνολογία GPRS

Αν και τα κινητά τηλέφωνα έχουν κατακλύσει τη ζωή μας, η χρήση τους για οτιδήποτε άλλο πέρα από τις επικοινωνίες φωνής και τη μετάδοση σύντομων μηνυμάτων κειμένου παραμένει μέχρι τις μέρες μας εξαιρετικά περιορισμένη. Οι ρυθμοί μετάδοσης είναι χαμηλοί, με αποτέλεσμα οι εφαρμογές διακίνησης ψηφιακών δεδομένων (Fax, πρόσβαση στο Internet) να είναι πολύ αργές αλλά και αρκετά ακριβές στη χρήση, δεδομένου ότι η χρέωση γίνεται με βάση το χρόνο επικοινωνίας.
Ευτυχώς, η τεχνολογία έχει δώσει απάντηση στην ανάγκη για ταχύτερη, αποδοτικότερη και οικονομικότερη διακίνηση ψηφιακών δεδομένων μέσα απάτα δίκτυα κινητής τηλεφωνίας. Η απάντηση αυτή, προς το παρόν, ονομάζεται **GPRS (General Packet Radio Service)** και δεν είναι τίποτε άλλο από μία υπηρεσία προστιθεμένης αξίας που αναβαθμίζει τις δυνατότητες διακίνησης ψηφιακών δεδομένων των υπαρχόντων δικτύων, εκτοξεύοντας στα ύψη τη χρησιμότητά τους.
Το GPRS τοποθετείται χρονικά και τεχνολογικά μεταξύ της δεύτερης και τρίτης γενιάς δικτύων κινητής τηλεφωνίας, με την ανάπτυξη των πρώτων δικτύων να χρονολογείται από την αρχή της δεκαετίας του 1980. Την εποχή εκείνη έκανε την εμφάνισή της στην Ευρώπη η πρώτη γενιά δικτύων (1G), που βασιζόταν σε αναλογικά συστήματα. Η τεχνολογία circuit switching που χρησιμοποιούσε, παρουσίαζε πολλούς περιορισμούς που δεν επέτρεπαν την περαιτέρω ανάπτυξη της κινητής τηλεφωνίας. Προσέφερε χαμηλής ποιότητας φωνητικές υπηρεσίες και μικρή χωρητικότητα δικτύου, ενώ η αδυναμία επέκτασης σε ευρείες γεωγραφικές περιοχές και η ασυμβατότητα μεταξύ των δικτύων κάθε χώρας έκαναν επιτακτική την ανάγκη βελτιώσεων.
Με την ανάπτυξη τεχνικών ψηφιακής μετάδοσης γεννήθηκε η δεύτερη γενιά κινητής τηλεφωνίας (2G). Το γνωστότερο και δημοφιλέστερο σύστημα, το **GSM (Global System for MobiIe Communication)**, αναπτύχθηκε χρέωση ανάλογα με τον στην Ευρώπη το 1991, αντικαθιστώντας τα προηγούμενα αναλογικά συστήματα. Το GSM κάνει χρήση των τεχνικών **Frequency Division Multiple Access (FDMA)** και **Time Division Multiple Access (TDMA)**, που διαμοιράζουν το φάσμα ραδιοσυχνοτήτων μεταξύ διαφορετικών τηλεφωνικών συνδιαλέξεων. Με τη FDMA το διαθέσιμο

εύρος φάσματος (bandwidth) διαιρείται σε κομμάτια των 200KHz, ενώ ταυτόχρονα κάθε συχνότητα διαιρείται με την TDMA σε 8 χρονοθυρίδες (timeslots). Η δεύτερη αυτή γενιά ασύρματων δικτύων παρέμεινε επικεντρωμένη στη φωνητική επικοινωνία, διαθέτοντας περιορισμένες μόνο δυνατότητες αποστολής και λήψης δεδομένων. Αν και υποστηρίζονται υπηρεσίες Fax και WAP (Wireless Application Protocol), ο προσφερόμενος ρυθμός διαμεταγωγής δεν ενδείκνυται για multimedia εφαρμογές, αφού ανέρχεται σε μόλις 9,6 Kbps.

Τα τελευταία χρόνια, η εντυπωσιακή ανάπτυξη τόσο της κινητής τηλεφωνίας όσο και του αριθμού των ατόμων με πρόσβαση στο Internet υπόσχεται τη δημιουργία μίας νέας αγοράς που θα συνδυάζει και τις δύο καινοτομίες. Στο προσεχές μέλλον αναμένεται να αναδυθεί πιεστικά η ανάγκη για ασύρματη πρόσβαση σε πληροφορίες, απαίτηση που δεν μπορούν να ικανοποιήσουν τα υπάρχοντα ασύρματο δίκτυα. Απάντηση στο αίτημα αυτό αναμένεται να δώσει η τρίτη γενιά δικτύων (3G). Τα δίκτυα αυτά σηματοδοτούν τη μεταστροφή από τη φωνητική επικοινωνία σε muItimedia υπηρεσίες και την πλήρη σύγκλιση της κοινωνίας της πληροφορίας και της επικοινωνίας. Το GPRS Θα λειτουργήσει ως ενδιάμεσος σταθμός και πρόδρομος των μελλοντικών εξελίξεων, ενσωματώνοντας αρκετά από τα προηγμένα χαρακτηριστικά που αναμένουμε.



Εικόνα 13 GPRS

2.6.1 GPRS: Χαρακτηριστικά και πλεονεκτήματα

Το GPRS (General Packet Radio Service) είναι μία νέα υπηρεσία που επιτρέπει την αποστολή και λήψη δεδομένων μέσω των δικτύων κινητής τηλεφωνίας (Wireless Data Access). Λειτουργεί συμπληρωματικά προς τα σημερινά Circuit Switched Data (CDS) και Short Message Service (SMS), προσφέροντας εύκολη, ασύρματη πρόσβαση σε packet data networks όπως το Internet. Πρέπει να τονιστεί ότι δεν πρόκειται για νέο σύστημα κινητής τηλεφωνίας, αλλά για μία υπηρεσία που "επικάθεται" στο υπάρχού δίκτυο GSM.

Με το GPRS οι πληροφορίες πριν από την αποστολή τους κατατέμνονται σε μικρότερα πακέτα (data packets), ενώ επανασυνδέονται πριν να φτάσουν στον τελικό αποδέκτη τους. **Ο** τρόπος αυτός λειτουργίας ονομάζεται packet switching και μπορεί να παρομοιαστεί με ένα παζλ**.** Φυσικά, οι ενδιάμεσες διαδικασίες κατάτμησης, αποστολής και επανασύνδεσης των πακέτων γίνεται αυτόματα από το δίκτυο GPRS, χωρίς ο χρήστης να αντιλαμβάνεται το παραμικρό. Τα δίκτυα που βασίζονται στο συγκεκριμένο τρόπο

λειτουργίας ονομάζονται packet data networks και ο διασημότερος εκπρόσωπός τους είναι το ίδιο το Internet.
Ένα ακόμη σημαντικό πλεονέκτημα είναι ότι οι χρήστες θα έχουν τη δυνατότητα, μεταξύ άλλων, να μειώσουν σημαντικά το κόστος χρήσης του κινητού τηλεφώνου, καθώς θα χρεώνονται βάσει του **"όγκου"** των δεδομένων που θα έχουν επιλέξει να δουν (Internet) και όχι του χρόνου που θα είναι συνδεδεμένοι στην υπηρεσία.

2.6.2 Γιατί είναι σημαντικό το GPRS;

Το πιο σημαντικό χαρακτηριστικό της νέας υπηρεσίας είναι ότι επιτρέπει την αποστολή και λήψη δεδομένων με ταχύτητες άνω των 100 Kbps. Το γεγονός αυτό από μονο του καθιστά εφικτές δυνατότητες που μόλις λίγα χρόνια πριν ανήκαν στη σφαίρα της φαντασίας. Ο χρήστης GPRS έχει τη δυνατότητα κανονικής πλοήγησης στο Internet, όχι μόνο χωρίς τους περιορισμούς του WAP αλλά και με ταχύτητες ανώτερες από αυτές που του παρέχει ένα συνηθισμένο σημερινό modem. Επίσης, αν και μέχρι σήμερα ο τύπος των δεδομένων που μεταδιδόταν μέσα από τα δίκτυα κινητής τηλεφωνίας ήταν κυρίως κείμενο, **η εντυπωσιακή ταχύτητα του GPRS θα καταστήσει δυνατή την αποδοτική μετάδοση εικόνων, φωνής και βίντεο**.
Αναμένεται, λοιπόν, μία ποιοτική στροφή στις εφαρμογές από το απλό κείμενο στις multimedia υπηρεσίες, γεγονός που Θα ικανοποιήσει τους καταναλωτές και θα βοηθήσει ακόμα περισσότερο στην εξάπλωση του GPRS. Δεν είναι δύσκολο να φαντασθεί κανείς τους νέους ορίζοντες που ανοίγονται, αφού για πρώτη φορά θα γίνει δυνατή η διεξαγωγή video conference μέσω κινητού, με τους συμμετέχοντες να βρίσκονται σε οποιοδήποτε σημείο του πλανήτη. Ενα άλλο παράδειγμα, εξίσου εντυπωσιακό, είναι η δυνατότητα συνεχούς παρακολούθηοης μίας απομακρυσμένης κάμερας ασφαλείας (surueillance camera) από το κινητό.

2.6.3 Βελτιστοποίηση φάσματος

Η τεχνολογία Packet Switching αυξάνει την αποδοτικότητα των υφιστάμενων δικτύων, αφού εξασφαλίζει ότι το φάσμα ραδιοσυχνοτήτων χρησιμοποιείται μόνο όταν ο χρήστης πραγματικά στέλνει ή λαμβάνει δεδομένα. Μέχρι σήμερα, τα δίκτυα εκχωρούσαν από ένα κανάλι

επικοινωνίας σε κάθε χρήστη κινητού τηλεφώνου, το οποίο παρέμενε δεσμευμένο καθ' όλη τη διάρκεια του τηλεφωνήματος, είτε ο χρήστης χρησιμοποιούσε το κανάλι είτε όχι. Αντίθετα, με το GPRS τα διαθέσιμα κανάλια μπορούν να χρησιμοποιούνται παράλληλα από πολλούς χρήστες, δίνοντας έτσι την εντύπωση ότι έχει αυξηθεί το εύρος του διαθέσιμου φάσματος (bandwith). Λαμβάνοντας υπόψη ότι το φάσμα των ραδιοσυχνοτήτων είναι ούτως ή άλλως πεπερασμένο, η αύξηση της χωρητικότητας που επιτυγχάνεται είναι εξαιρετικά σημαντική.

2.6.4 Remote data access

Παραδοσιακά, οι εταιρείες επέτρεπαν την απομακρυσμένη πρόσβαση των υπαλλήλων τους στο εσωτερικό δίκτυό τους μέσω dialup συνδέσεων. Αυτό σημαίνει ότι για να συνδεθούν οι υπάλληλοι, έπρεπε να τηλεφωνήσουν μέσω modem σε κατάλληλα εξοπλισμένο υπολογιστή της εταιρείας (modem pool). Η τεχνική αυτή απαιτούσε την εγκατάσταση επιπρόσθετου και ακριβού εξοπλισμού επιβαρύνοντας έτσι τον εταιρικό προϋπολογισμό.

Σήμερα, οι περισσότερες επιχειρήσεις εγκαθιστούν υψηλής ταχύτητας συνδέσεις με το Internet και αναζητούν νέους τρόπους εκμετάλλευσης των συνδέσεων αυτών για remote access. Το GPRS όχι μόνο βασίζεται στις ίδιες αρχές λειτουργίας με το Internet, αλλά παράλληλα υποστηρίζει τα πρωτόκολλα IP (Internet Protocol) και X.25. Τα τελευταία τα συναντάμε στην πλειονότητα των εταιρικών δικτύων της Ευρώπης και μέσω αυτών των πρωτοκόλλων θα είναι δυνατή η πλήρης πρόσβαση των υπαλλήλων στο intranet της εταιρείας τους, από οποιαδήποτε περιοχή του κόσμου.

Δυστυχώς, η νέα αυτή τεχνική εγκυμονεί αυξημένους κινδύνους για την ασφάλεια των δεδομένων των επιχειρήσεων. Το ηλεκτρονικό έγκλημα που ανθεί οτη σημερινή εποχή, θα βρει μέσω του GPRS νέες διόδους εισχώρησης στα εταιρικά δίκτυα διάφορων επιχειρήσεων. Για την αντιμετώπιση των προβλημάτων αυτών και την κανοποιητική προστασία της επικοινωνίας υπαλλήλων - επιχειρήσεων, αναπτύσσεται μία νέα τεχνολογία.

Τα VPN (Virtual Private Networks) δημιουργούν εικονικά ιδιωτικά δίκτυα, χρησιμοποιώντας λογισμικό που αναλαμβάνει την κρυπτογράφηση των πληροφοριών πριν από την αποστολή τους και την αποκρυπτογράφηση πριν από την παραλαβή τους. Η τεχνική αυτή, σε

συνδυασμό πάντα με τις ήδη υπάρχουσες άμυνες κατά των ηλεκτρονικών εισβολέων, αναμένεται να αποτελέσουν ικανή προστασία έναντι των επίδοξων εισβολέων.

2.6.5 Σύνδεση με Internet

Για πρώτη φορά γίνεται δυνατή η άμεση και συνεχής πρόσβαση σε όλες τις υπηρεσίες του Internet, όπως FTP (File Transfer Protocol), Web Browsing, Chat, Email και Telnet. **Η μέγιστη θεωρητική ταχύτητα μετάδοσης δεδομένων που μπορεί να επιτευχθεί σε ένα δίκτυο GPRS είναι 171,2 Kbits/sec,** δηλαδή τρεις φορές μεγαλύτερη από αυτήν που επιτυγχάνεται με ένα κοινό οικιακό modem συνδεδεμένο στο δίκτυο σταθερής τηλεφωνίας. Η διαφορά, λοιπόν, σε σχέση με το παρελθόν κρίνεται εξαιρετικά σημαντική, μειώνοντας ταυτόχρονα σε μεγάλο βαθμό τη σημερινή δυσαρέσκεια των χρηστών.
Δυστυχώς, ο εντυπωσιακός θεωρητικός ρυθμός διαμεταγωγής στην πράξη περιορίζεται από διάφορους παράγοντες, όπως την ποιότητα του ασύρματου δικτύου και την υποστηριζόμενη από το κινητό ταχύτητα. Ειδικότερα για τα GPRS κινητά, οι κατασκευάστριες εταιρείες έχουν επιδοθεί σε αγώνα δρόμου για την ενσωμάτωση της νέας τεχνολογίας. Παρά τον αυξημένο ανταγωνισμό, τα σημερινά προϊόντα προσφέρουν πολύ μειωμένες ταχύτητες σε σχέση με το θεωρητικό μέγιστο του GPRS, που διαμορφώνεται σε περίπου 4050Kbps.
Παράλληλα, ιδιαίτερα σημαντική είναι η απάλειψη της ανάγκης για modem και dialup συνδέσεις, αφού οι πληροφορίες στέλνονται και λαμβάνονται άμεσα. Αυτός είναι και ο λόγος που οι χρήστες GPRS κινητών αναφέρονται ως "**always connected**", με τη διαδικασία σύνδεσης να διαρκεί ελάχιστα δευτερόλεπτα.

2.7 Visual Basic 6

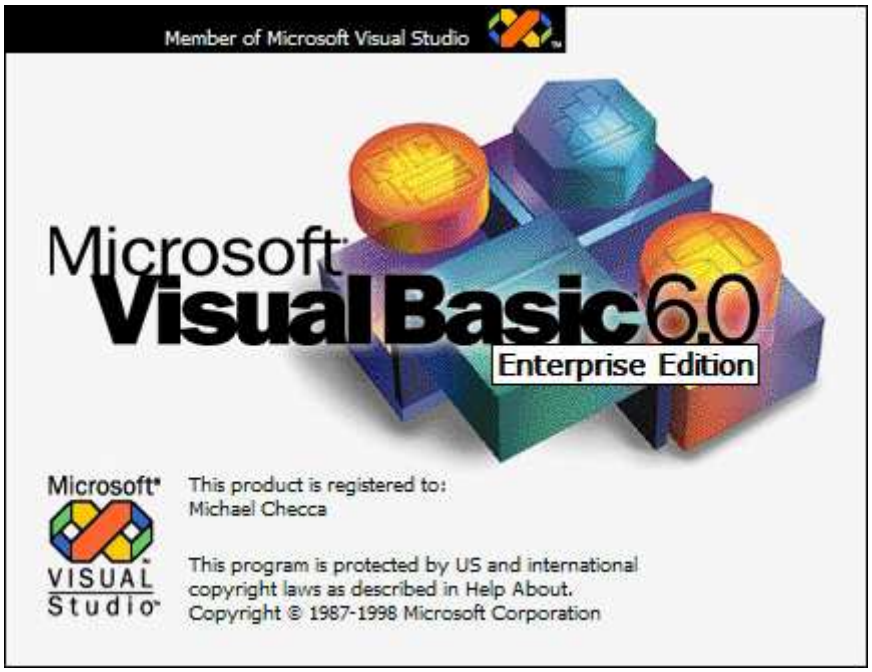
Η **Visual Basic (VB)** είναι γλώσσα προγραμματισμού τρίτης γενιάς, οδηγούμενη από συμβάντα (event driven) και έχει ολοκληρωμένο

περιβάλλον ανάπτυξης (IDE) από τη Microsoft για το μοντέλο προγραμματισμού COM. Η VB θεωρείται επίσης μία σχετικά εύκολη γλώσσα προγραμματισμού στην εκμάθηση και τη χρησιμοποίηση, λόγω των χαρακτηριστικών της, καθώς έχει Γραφικό Περιβάλλον Χρήστη και συγγένεια με την γλώσσα προγραμματισμού BASIC. Η Visual Basic προέρχεται από τη BASIC και επιτρέπει την ταχεία ανάπτυξη εφαρμογών (RAD) με Γραφικό Περιβάλλον Χρήστη (GUI), πρόσβαση σε βάσεις δεδομένων χρησιμοποιώντας αντικείμενα (Data Access Objects, Remote Data Objects, ή ActiveX Data Objects), και τη δημιουργία στοιχείων ελέγχου ActiveX και αντικειμένων.
Οι γλώσσες προγραμματισμού τύπου "scripting", όπως η VBA και VBScript συντακτικά είναι παρόμοιες με τη Visual Basic, αλλά έχουν διαφορετικές επιδόσεις. Ένας προγραμματιστής μπορεί να ολοκληρώσει μια εφαρμογή χρησιμοποιώντας τα στοιχεία που παρέχονται με την Visual Basic. Προγράμματα γραμμένα σε Visual Basic μπορούν, επίσης, να χρησιμοποιήσουν το Windows API, αλλά κάτι τέτοιο απαιτεί δηλώσεις εξωτερικών συναρτήσεων.
Η τελική έκδοση 6 βγήκε το 1998. Η εκτεταμένη υποστήριξη της Microsoft έληξε το Μάρτιο του 2008 και ορίστηκε διάδοχος της η Visual Basic.NET (γνωστή απλά ως Visual Basic).

2.7.1 Χαρακτηριστικά της Γλώσσας Visual Basic 6

Όπως και η γλώσσα προγραμματισμού BASIC, η Visual Basic έχει σχεδιαστεί για να είναι εύκολη στην εκμάθηση και το χειρισμό. Η γλώσσα δεν επιτρέπει στους προγραμματιστές να δημιουργήσουν μόνο απλές εφαρμογές GUI, αλλά μπορούν, επίσης, να αναπτύξουν πολύπλοκες εφαρμογές. Ο προγραμματισμός σε VB συνίσταται από τον οπτικό συνδυασμό στοιχείων ή ελέγχων σε μια φόρμα, τον προσδιορισμό χαρακτηριστικών και ενεργειών αυτών των στοιχείων και την σύνταξη επιπλέον γραμμών κώδικα για αυξημένη λειτουργικότητα.
Καθώς υπάρχουν προεπιλεγμένα χαρακτηριστικά και ενέργειες για τα επιμέρους στοιχεία, μπορεί να δημιουργηθεί ένα απλό πρόγραμμα χωρίς ο προγραμματιστής να γράψει πολλές γραμμές κώδικα. Στις προηγούμενες

εκδόσεις υπήρχαν προβλήματα επιδόσεων, αλλά με τους ταχύτερους υπολογιστές και τη μεταγλώττιση εγγενούς κώδικα αυτό παύει να είναι ένα τόσο σημαντικό ζήτημα.



Εικόνα 14 Visual Basic 6

Αν και τα προγράμματα μπορούν να μετατραπούν σε εγγενή εκτελέσιμο κώδικα από την έκδοση 5 και μετά, αυτά εξακολουθούν να απαιτούν την παρουσία των βιβλιοθηκών χρόνου εκτέλεσης (runtime) με μέγεθος περίπου 1 MB. Οι βιβλιοθήκες runtime υπάρχουν στα Windows 2000 και αργότερα, αλλά στις παλαιότερες εκδόσεις των Windows όπως τα 95/98/NT πρέπει να διανέμονται μαζί με το εκτελέσιμο αρχείο.

Οι φόρμες δημιουργούνται χρησιμοποιώντας τεχνικές "σύρε κι άσε" (drag-and-drop). Χρησιμοποιείται ένα εργαλείο για την τοποθέτηση στοιχείων ελέγχου (π.χ. πλαίσια κειμένου, κουμπιά, κλπ.) στη φόρμα (παράθυρο). Τα στοιχεία ελέγχου έχουν χαρακτηριστικά και χειριστές συμβάντων συνδεδεμένους με αυτά. Οι προεπιλεγμένες τιμές παρέχονται όταν δημιουργείται το στοιχείο ελέγχου, αλλά μπορούν να τροποποιηθούν από τον προγραμματιστή.

Πολλές τιμές χαρακτηριστικών είναι δυνατό να τροποποιηθούν κατά το χρόνο εκτέλεσης από ενέργειες του χρήστη ή αλλαγές του περιβάλλοντος,

παρέχοντας έτσι μια δυναμική εφαρμογή. Για παράδειγμα, μπορεί να εισαχθεί κώδικας στον χειριστή συμβάντων αλλαγής διαστάσεων της φόρμας, ώστε ένα στοιχείο ελέγχου να παραμένει πάντα στο κέντρο της φόρμας ή να μεγαλώσει ώστε να την γεμίσει, κλπ. Με την προσθήκη κώδικα μέσα σε ένα χειριστή συμβάντων για το πάτημα των πλήκτρων σε ένα πλαίσιο κειμένου, το πρόγραμμα μπορεί αυτόματα να μετατρέψει το εισαγόμενο κείμενο σε κεφαλαία ή πεζά ή ακόμα και να εμποδίσει ορισμένους από τους χαρακτήρες να εμφανιστούν.

Με τη Visual Basic είναι δυνατή η δημιουργία εκτελέσιμων (EXE) αρχείων, στοιχείων ελέγχου ActiveX ή αρχείων DLL, αλλά χρησιμοποιείται κυρίως για την ανάπτυξη εφαρμογών για τα Windows και τη διασύνδεση συστημάτων βάσεων δεδομένων. Πλαίσια διαλόγου με λιγότερες λειτουργίες μπορούν να χρησιμοποιηθούν για pop-up δυνατότητες. Τα στοιχεία ελέγχου παρέχουν τις βασικές λειτουργίες της εφαρμογής, ενώ οι προγραμματιστές μπορούν να εισαγάγουν επιπλέον λογική μέσα στο κατάλληλο χειριστή γεγονότων. Για παράδειγμα, ένα πτυσσόμενο πλαίσιο θα εμφανίζει αυτόματα μια λίστα που θα επιτρέπει στο χρήστη να επιλέξει οποιοδήποτε στοιχείο.

Ένας χειριστής γεγονότων καλείται όταν ένα αντικείμενο είναι επιλεγμένο. Στη συνέχεια μπορεί να εκτελεστεί πρόσθετος κώδικας που δημιουργείται από τον προγραμματιστή για να εκτελεστεί κάποια ενέργεια που βασίζεται στο στοιχείο που έχει επιλεχθεί (το αντικείμενο).

Εναλλακτικά, ένα συστατικό της Visual Basic μπορεί να μην έχει Γραφικό Περιβάλλον Χρήστη, αλλά, αντί' αυτού, να παρέχει αντικείμενα ActiveX σε άλλα προγράμματα μέσω Component Object Model (COM). Αυτό επιτρέπει επεξεργασία στην πλευρά του διακομιστή (server-side processing) ή τη δημιουργία πρόσθετων μορφωμάτων (add-in module).

Η γλώσσα έχει αυτόματη διαχείριση μνήμης τύπου garbage collection χρησιμοποιώντας υπολογισμό αναφορών και έχει μια μεγάλη βιβλιοθήκη με βοηθητικά αντικείμενα καθώς και βασική αντικειμενοστραφή υποστήριξη. Από τα πιο κοινά στοιχεία που περιλαμβάνονται στο προεπιλεγμένο πρότυπο έργου, ο προγραμματιστής σπάνια χρειάζεται να καθορίσει πρόσθετες βιβλιοθήκες. Αντίθετα με πολλές άλλες γλώσσες προγραμματισμού η Visual Basic γενικά δεν διαχωρίζει τους πεζούς από τους κεφαλαίους χαρακτήρες, αν και θα μετατρέψει τις λέξεις-κλειδιά σε μία

τυπική διαμόρφωση. Οι συγκρίσεις συμβολοσειρών διαχωρίζουν τα πεζά από τα κεφαλαία από προεπιλογή, αλλά μπορεί να αλλάξει αυτό, εφόσον το επιθυμείτε.
Ο μεταγλωττιστής της Visual Basic είναι κοινός με τις άλλες γλώσσες του Visual Studio (C, C++), αλλά οι περιορισμοί στον IDE δεν επιτρέπουν τη δημιουργία ορισμένων στόχων (μοντέλα Windows DLL) και σε μοντέλα νημάτων.

Χαρακτηριστικά της Visual Basic
Visual Basic έχει τα εξής χαρακτηριστικά που διαφέρουν από τις γλώσσες που προέρχονται από τη C:

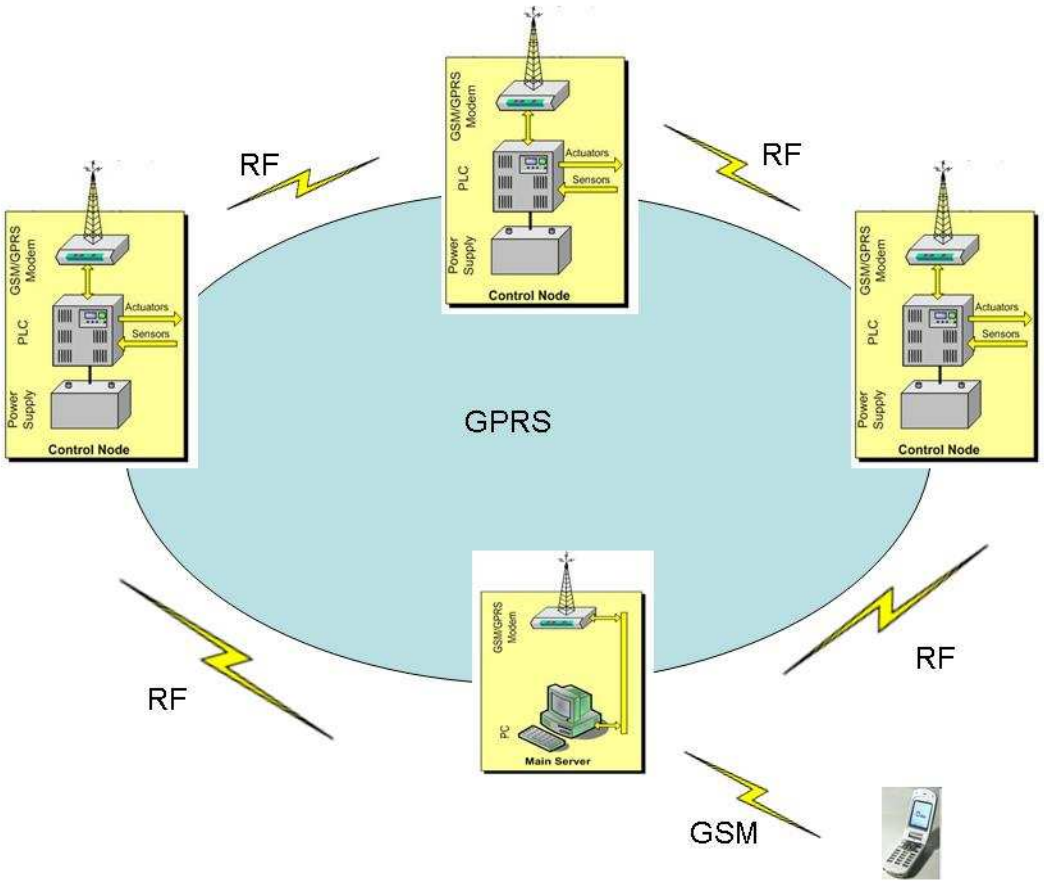
- Η πολλαπλή ανάθεση που διατίθεται στη C δεν είναι δυνατή. Το A = B = C, δεν σημαίνει ότι οι τιμές των A, B και C είναι ίδιες. Το αποτέλεσμα της boolean "B = C;" αποθηκεύεται στην μεταβλητή A. Το αποτέλεσμα που αποθηκεύεται στην A θα μπορούσε συνεπώς να είναι ψευδές (0) ή αληθές (-1).

- Οι λογικοί και δυαδικοί τελεστές είναι ενοποιημένοι. Αυτό έρχεται σε αντίθεση με ορισμένες γλώσσες προερχόμενες από τη C (όπως η γλώσσα Perl), οι οποίες έχουν ξεχωριστούς λογικούς και δυαδικούς τελεστές. Αυτό είναι ένα παραδοσιακό χαρακτηριστικό της γλώσσας BASIC.

Κεφάλαιο 3 Αρχιτεκτονική συστήματος SCADA

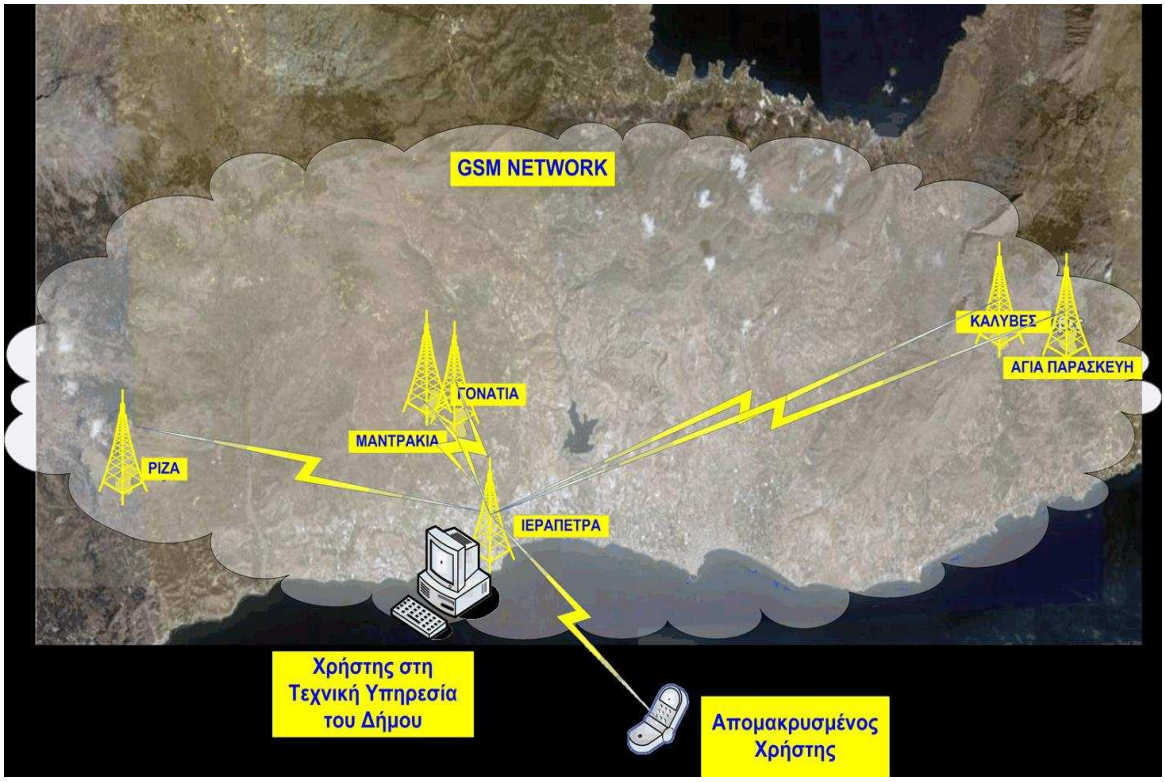
3.1Σύντομη περιγραφή συστήματος

Η δομή του συστήματος αποτελείται από διάφορα μέρη. Το σύστημα SCADA αποτελείται από κόμβους ασύρματους που έχουν τη δυνατότητα συλλογής δεδομένων. Ειδικότερα είναι ένα σύνολο από αντλιοστάσια τα οποία βρίσκονται στην ευρύτερη περιοχή της Ιεράπετρας. Η κάθε αντλία - κόμβος "επικοινωνεί" με όλες τις υπόλοιπες αλλά και με έναν κεντρικό server μέσω του οποίου στέλνονται τα δεδομένα. Τα δεδομένα τα παίρνει ο χρήστης είτε μέσο internet είτε μέσο δικτύου GSM/GPRS(κινητό τηλέφωνο). Η εικόνα 15 μας δείχνει την Αρχιτεκτονική Συστήματος. Σε αυτήν την εικόνα βλέπουμε τον κάθε κόμβο και από τι αποτελείται. Φαίνεται επίσης η επικοινωνία μεταξύ των κόμβων αλλά και με τον απομακρυσμένο χρήστη μέσο δικτύου GSM/GPRS(κινητό τηλέφωνο).



Εικόνα 15 Αρχιτεκτονική Συστήματος

Η εικόνα 16 μας δείχνει μία τοπογραφική άποψη της περιοχής όπου έχει εγκατασταθεί η συγκεκριμένη εφαρμογή. Βλέπουμε σε αυτήν την εικόνα ένα χάρτη και πάνω σε αυτόν κάποιες τοποθεσίες. Οι τοποθεσίες αυτές είναι ΜΑΝΤΡΑΚΙΑ, ΡΙΖΑ, ΓΟΝΑΤΙΑ, ΚΑΛΥΒΕΣ, ΑΓΙΑ ΠΑΡΑΣΚΕΥΗ και η ΙΕΡΑΠΕΤΡΑ. Οι 5 πρώτες τοποθεσίες είναι και οι αντλίες-κόμβοι που είδαμε ξανά στην εικόνα 12 Στην Ιεράπετρα βρίσκεται ο χρήστης της Τεχνικής Υπηρεσίας του Δήμου ο οποίος μπορεί να διαχειρίζεται τους κόμβους. Επίσης υπάρχει η δυνατότητα ενημέρωσης απομακρυσμένου χρήστη και η δυνατότητα διαχείρισης του συστήματος μέσω αποστολής και λήψης SMS μηνυμάτων.



Εικόνα 16 Αρχιτεκτονική Συστήματος (όψη από Google Earth)

3.2 Χαρακτηριστικά

Η συγκεκριμένη εφαρμογή που δημιουργήθηκε για τις ανάγκες του δήμου Ιεράπετρας προσφέρει τα εξής χαρακτηριστικά :

- Τηλεμετρία διεργασιών
- Τηλεχειρισμός διεργασιών
  Συλλογή πληροφοριών και αξιοποίηση τους για τον εποπτικό τους έλεγχο
- Οπτικοποίηση συνολικής κατάστασης επιτηρούμενης διεργασίας προσαρμοσμένη στις ανάγκες του χρήστη
- Τοπικός και απομακρυσμένος αυτοματισμός
  Τοπική πρόσβαση ανά αντλία είτε απομακρυσμένη μέσω του κεντρικού υπολογιστή(server)
- Ασύρματη επικοινωνία μέσω δικτύου κινητής τηλεφωνίας
- Χαμηλό κόστος λειτουργίας

- Αδιάλειπτη τροφοδοσία
  Συνεχής λειτουργία με χρήση εφεδρικών μπαταριών σε περίπτωση διακοπής τροφοδοσίας ρεύματος
- Δυνατότητα τροφοδοσίας από ηλιακή ενέργεια
- Χρήση μηνυμάτων SMS
  - Μηνύματα συναγερμού
  - Ενημέρωση κατάστασης
  - Επέμβαση, έλεγχος

  Πλήρης δυνατότητα χειρισμού εφαρμογής μέσω γραπτών μηνυμάτων(SMS)
- Ελεγχόμενη πρόσβαση
  Πρόσβαση στην εφαρμογή μετά από ταυτοποίηση ταυτότητας χρήστη
- Σενάρια αυτοματισμού (από το χρήστη)
  Δημιουργία εντολών (script) ανάλογα με τις ανάγκες του χρήστη
- Προσαρμογή στις ανάγκες του Χρήστη

Από το σύστημα αυτό τραβάμε πληροφορίες όπως:

- Κατάσταση Αυτοματισμού
  - Κατάσταση λειτουργίας
  - Βλάβη θερμικού
  - Στάθμη νερού
  - Ροή Νερού
  - Ρελέ εκκίνησης
- Τάσεις τροφοδοσίας αντλιοστασίου
- Εντάσεις τροφοδοσίας αντλιοστασίου
- Χρόνος λειτουργίας
- Συντελεστής Ισχύος
- Ισχύς Λειτουργίας
- Παροχή Νερού

3.3 Περιγραφή – Τρόπος λειτουργίας

Σε 5 περιοχές τις Ιεράπετρας υπάρχει από μια αντλία - κόμβος. Οι πέντε περιοχές βρίσκονται στην ευρύτερη περιοχή του δήμου Ιεράπετρας. Ο κάθε κόμβος αποτελείται από το κομμάτι του software και από το κομμάτι του hardware. Επιπλέον υπάρχει και ο κεντρικός server στην τεχνική υπηρεσία του δήμου. Για τις πέντε αντλίες-κόμβους το κομμάτι του hardware αποτελείται από μια μονάδα PLC, ένα GSM/GPRS modem και την μονάδα τροφοδοσίας του. Το PLC και το GSM/GPRS modem είναι μαζί στην μονάδα mt-101 όπου συνδέονται όλες οι συσκευές εισόδου και η μια έξοδος.

Το κύριο πλεονέκτημα είναι ότι μέσω της σύνδεσης του GSM/GPRS δικτύου έχουμε κάλυψη παντού. Μέσω του δικτύου GSM/GPRS στέλνονται τα δεδομένα τα οποία έχουν μέγεθος, για τις **αναλογικές τιμές** 4 byte η κάθε πληροφορία και 1 byte για όλες τις **ψηφιακές τιμές**.

Οι **αναλογικές τιμές** είναι οι συνεχείς τιμές που λαμβάνει το αναλογικό σήμα καθώς μεταβάλλεται κατά την διάρκεια εξέλιξης του χρόνου. Το αναλογικό αναφέρεται συνήθως σε ηλεκτρικά σήματα ,αλλά και σε άλλα συστήματα που μπορεί να μεταφέρουν αναλογικά σήματα όπως π.χ. μηχανικά, αέρια ή υδραυλικά συστήματα.

Οι **ψηφιακές τιμές** είναι οι συγκεκριμένες τιμές που μπορεί να λάβει ένα σήμα διακριτού χρόνου στον άξονα του χρόνου όπως η έξοδος του SCADA στο σύστημα μας.

Στον πίνακα που ακολουθεί βλέπουμε όλες τις αναλογικές τιμές.

Αναλογικές Τιμές	**Μέγεθος Πληροφορίας**
Battery Voltage	4 byte
V1N	4 byte
V2N	4 byte
V3N	4 byte
U1	4 byte
U2	4 byte

U3	4 byte
I1	4 byte
I2	4 byte
I3	4 byte
WaterFlowRate	4 byte
Power Factor	4 byte
Real Power	4 byte
Usage Hours	4 byte

Σύνολο 56 bytes για όλες τις αναλογικές τιμές.

Ανάλυση αναλογικών τιμών

- Battery Voltage : Είναι η τιμή της τάσης της μπαταρίας
- WaterFlowRate : Δείχνει το ρυθμό ροής νερού

Τις παρακάτω τιμές τις παίρνουμε από τον μετρητή τάσεων και εντάσεων (PowerMeter 710) και είναι οι :

- V1N
- V2N
- V3N

Φασικές Τάσεις

- U1
- U2
- U3

Πολικές Τάσεις

- I1
- I2
- I3

Εντάσεις Ρευμάτων

- Power Factor : Είναι ο συντελεστής ισχύος του εναλλασσόμενου ρεύματος ηλεκτρικού συστήματος που ορίζεται ως ο λόγος του φορτίου που διέρχεται στην πραγματική ισχύ προς φορτίου που διέρχεται στη φαινομενική ισχύ και είναι ένας αριθμός μεταξύ 0 και 1

- Real Power : Είναι η ενεργός ισχύς που καταναλώνει η συσκευή (είναι η χωρητικότητα του κυκλώματος για την εκτέλεση εργασιών σε ένα συγκεκριμένο χρονικό διάστημα.
- Usage Hours : Ώρες χρησιμοποίησης του συστήματος.

Η συλλογή των αναλογικών τιμών που στέλνει ο server μέσω του δικτύου GSM/GPRS γίνεται ταυτόχρονα με δύο τρόπους. Ο πρώτος τρόπος αναφέρεται στην συλλογή δεδομένων ανά συγκεκριμένα χρονικά διαστήματα (π.χ. ανά 30 λεπτά) και ονομάζεται **polling**. Έτσι θα έχουμε ενημέρωση για το τι συμβαίνει στο σύστημα ανά συγκεκριμένη χρονική στιγμή. Αυτό όμως δεν είναι αρκετό.
Για να υπάρχει πλήρη ενημέρωση πρέπει ο χρήστης να ενημερώνεται για οποιαδήποτε σημαντική αλλαγή. Αυτός ο τρόπος συλλογής δεδομένων ονομάζεται **event-driven**. Είναι ένας προγραμματιστικός τρόπος μέσω του οποίου, η συλλογή των δεδομένων καθορίζεται από γεγονότα όπως την έξοδο αισθητήρων, τις ενέργειες του χρήστη(κλικ στο ποντίκι,πάτημα στο κουμπί), μηνύματα από άλλα προγράμματα και αλλαγή από ένα ανώτερο ή κατώτερο όριο τιμής(threshold). Συγκεκριμένα για τα αναλογικά δεδομένα λειτουργεί βάσει *threshold.*. Αν για παράδειγμα η τάση αυξομειωθεί κατά 10 volt τότε υπάρχει πρόβλημα στο σύστημα και πρέπει να ενημερωθεί ο χρήστης.
Στον πίνακα που ακολουθεί βλέπουμε όλες τις ψηφιακές τιμές.

Ψηφιακές Τιμές
PowerOk
ThermalFailure
WaterFlow
MotorRun
WaterLevelOk

Για όλες τις ψηφιακές τιμές, χρειαζόμαστε 1 byte.

Ανάλυση ψηφιακών τιμών

- PowerOk : Ένδειξη αν η σύνδεση με το δίκτυο της ΔΕΗ είναι εντάξει (αν υπάρχει τάση)
- ThermalFailure : Έλεγχος πτώσης θερμικού (αν υπάρχει βλάβη). Τα συστήματα έχουν ένα θερμικό σε περίπτωση υπερφόρτωσης για να μην καταστραφεί η αντλία
- WaterFlow :Ένδειξη ροής νερού (Ελέγχουμε αν περνάει νερό). Για να αποφύγουμε να δουλεύει η αντλία χωρίς νερό υπάρχει στο αντλιοστάσιο ένας διακόπτης ροής ο οποίος ελέγχει όταν περνάει νερό μέσα από τη σωλήνα.
- MotorRun : Ρελέ ενεργοποίησης κινητήρα
- WaterLevelOk : Ένδειξη αν η αντλία είναι μέσα στο νερό

Η συλλογή των ψηφιακών τιμών γίνεται με την μέθοδο **event-driven**. Η μέθοδος αυτή είναι ακριβώς η ίδια όπως την περιγράψαμε προηγουμένως. Επειδή έχουμε όμως ψηφιακές πληροφορίες η ενημέρωση του χρήστη δεν γίνεται βάσει threshold αλλά με την αλλαγή της ψηφιακής τιμής (0 ή 1).
Στην εικόνα 14 βλέπουμε την μονάδα PLC τις εισόδους, την έξοδο και το σειριακό πρωτόκολλο επικοινωνίας MODBUS.

Οι ψηφιακές είσοδοι είναι

- PowerOk
- ThermalFailure
- WaterFlow
- MotorRun
- WaterLevelOk

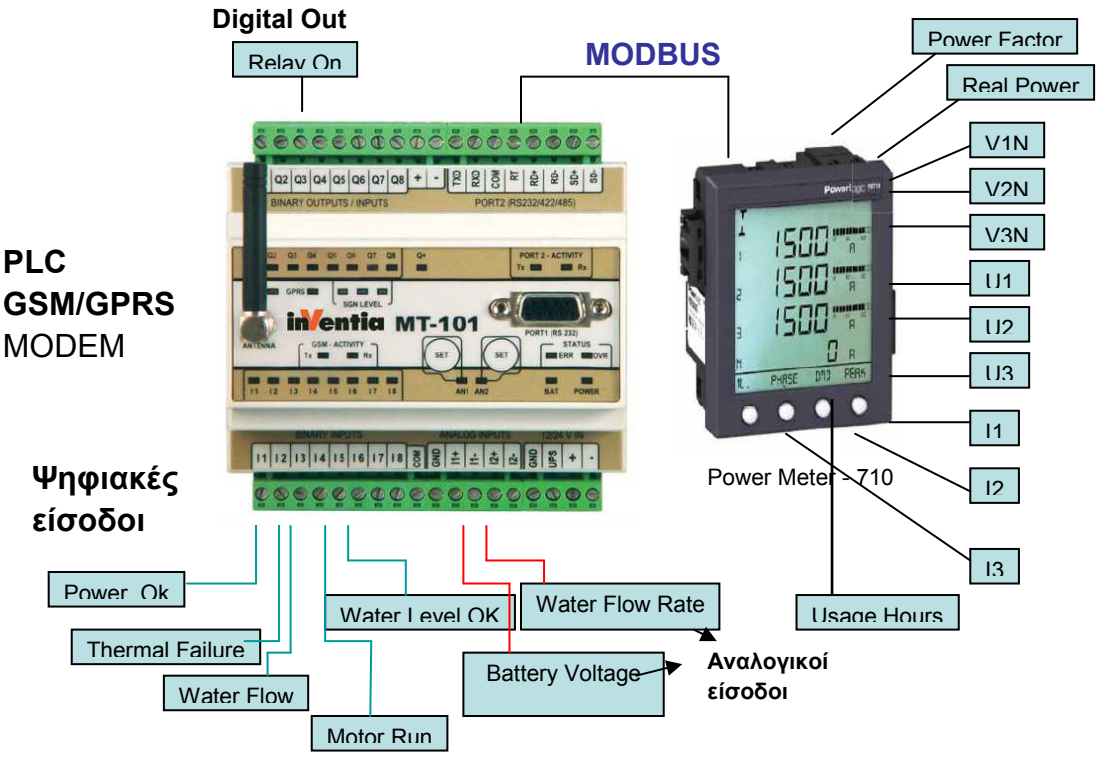
Οι αναλογικοί είσοδοι που συνδέονται απευθείας στο PLC είναι οι

- BatteryVoltage
- WaterFlowRate

Επιπλέον έχουμε και τις αναλογικές εισόδους που συνδέονται μέσο MODBUS και είναι οι

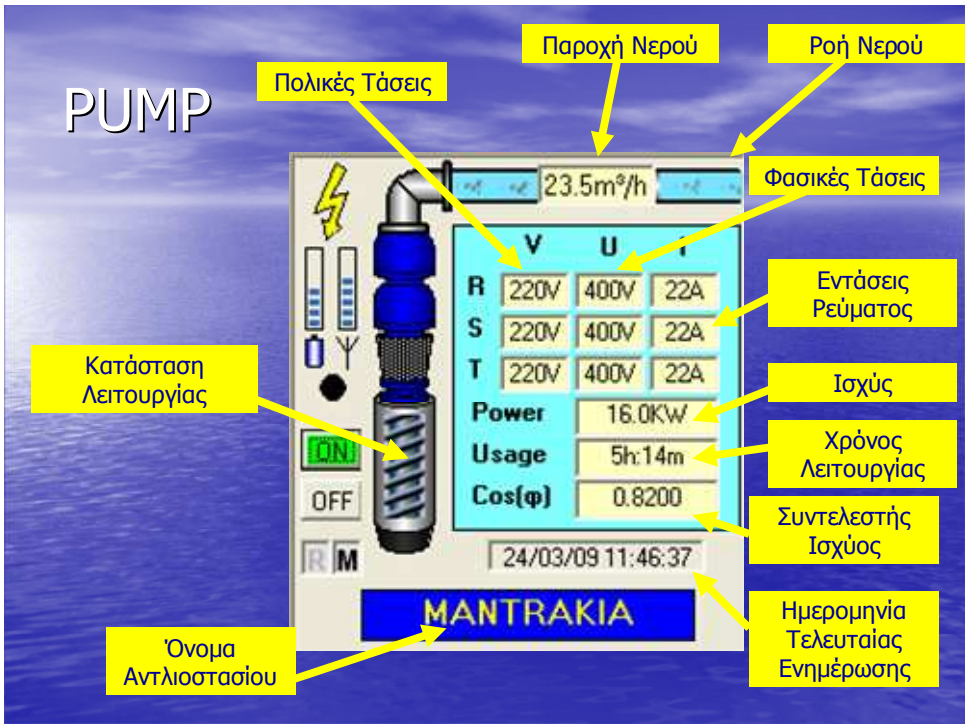
- V1N
- V2N
- V3N
- U1
- U2
- U3
- I1
- I2
- I3
- PowerFactor
- Real Power
- Usage Hours

.



Εικόνα 17 Συνδεσμολογία μονάδας PLC – GSM/GPRS Modem

Το software κομμάτι κάθε αντλίας-κόμβου έχει την παρακάτω μορφή όπου φαίνονται αναλυτικά τα χαρακτηριστικά αυτής. Η μορφή αυτή φαίνεται στην εικόνα 18.



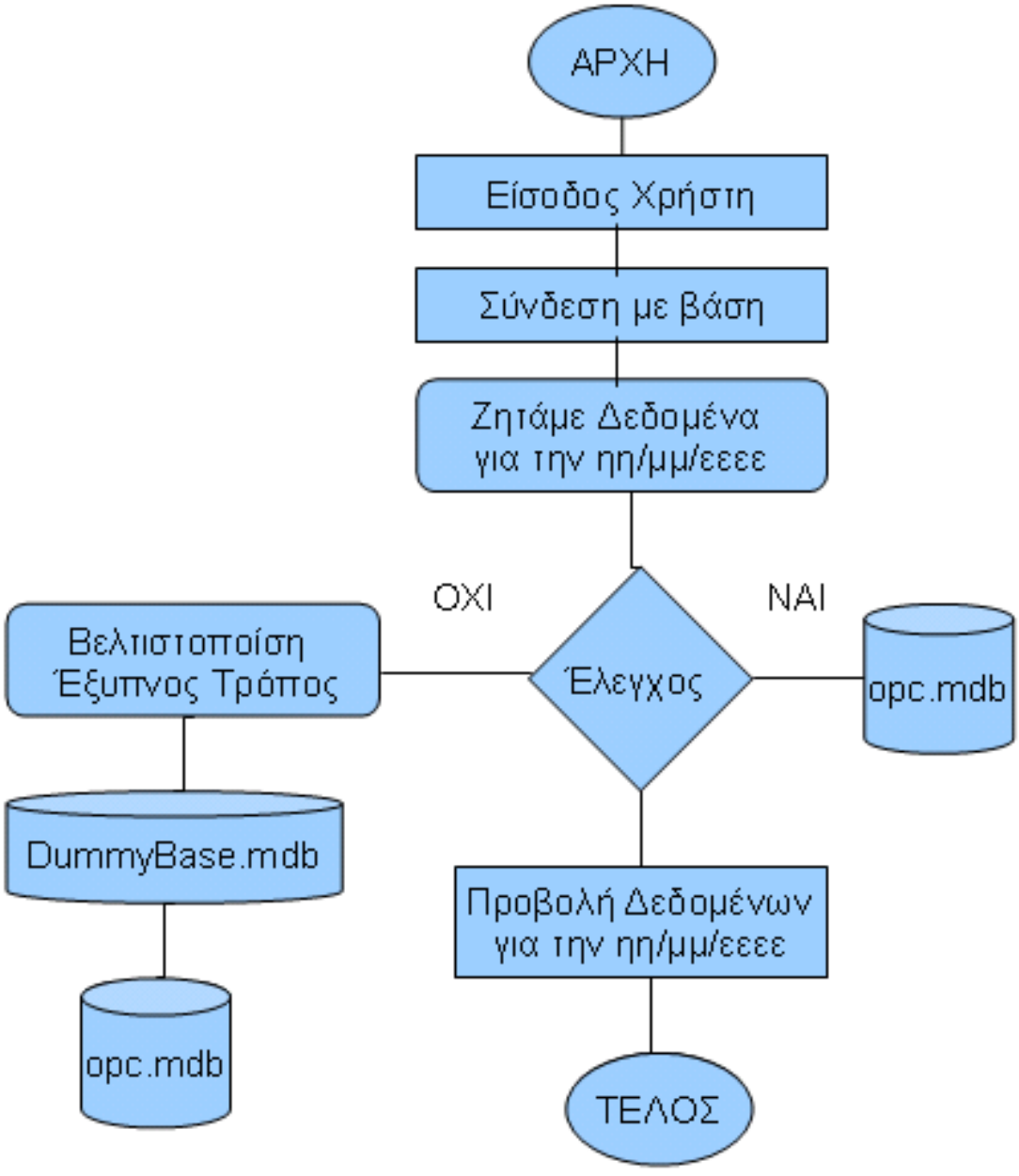
Εικόνα 18 Χαρακτηριστικά Αντλίας

3.4 Βελτιστοποίηση εφαρμογής

Η πτυχιακή μας έχει σκοπό την βελτιστοποίηση της εφαρμογής. Η βελτιστοποίηση αυτή αναφέρεται σε έναν "***έξυπνο***" τρόπο καταγραφής των δεδομένων της κατάστασης της κάθε αντλίας. Τον αναφέρουμε σαν έξυπνο διότι η εφαρμογή δεν συλλέγει δεδομένα για μία συγκεκριμένη χρονική στιγμή, αλλά για την συγκεκριμένη χρονική στιγμή γνωρίζουμε την πραγματική κατάσταση ης αντλίας. Αυτό σημαίνει ότι όταν ο χρήστης επιθυμεί να ενημερωθεί για την κατάσταση της αντλίας μια συγκεκριμένη χρονική στιγμή τα δεδομένα που θα λάβει είναι τα αμέσως χρονικά προηγούμενα.

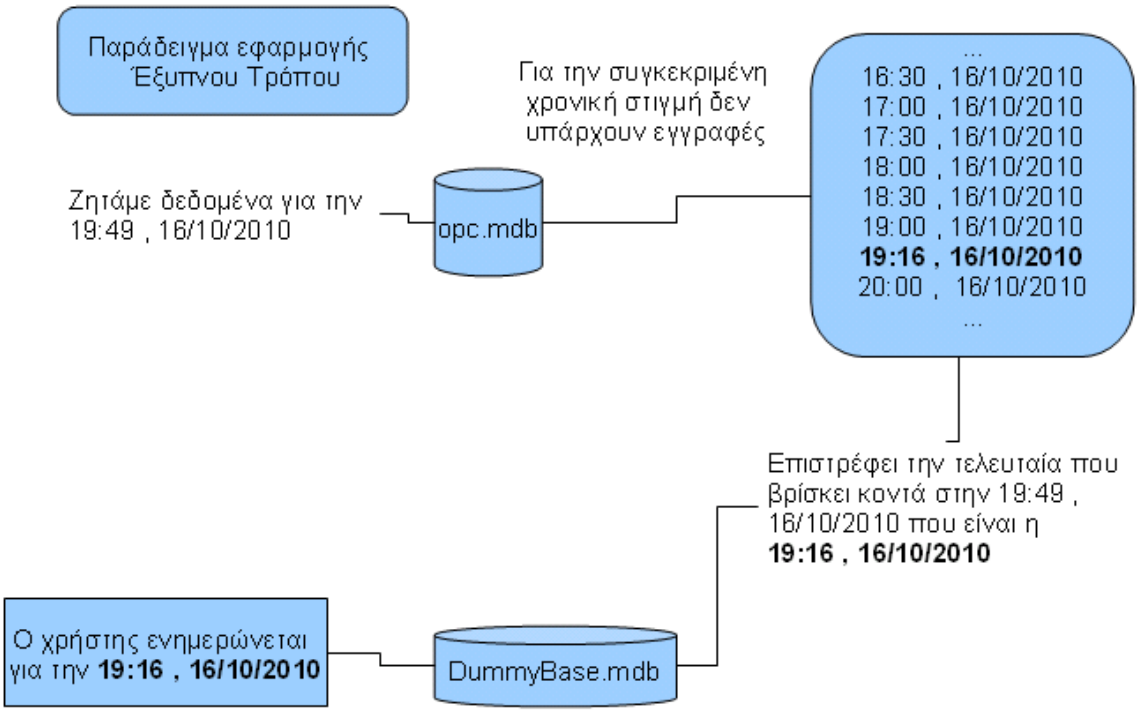
Στο πρόγραμμα που έχουμε δημιουργήσει έχουμε φτιάξει μια συνάρτηση η οποία τοποθετεί στη "ψεύτικη" βάση (DummyBase.mdb) τις αμέσως χρονικά προηγούμενες εγγραφές που έχει βρεί από την πραγμαική βάση opc.mdb . Στην ουσία εμείς φτιάχνουμε μία "ψεύτικη" βάση που δεν έχει

πραγματικές εγγραφές αλλά αντιστοιχούν στην πραγματικότητα.. Είναι όμως σίγουρο ότι η τελευταία τιμή-εγγραφή της βάσης ισχύει γιατί παρόλο που τραβάμε τα δεδομένα με **polling** ανά τακτά χρονικά διαστήματα, αν είχε γίνει κάποια αλλαγή θα στελνόταν λόγω της αλλαγής του ορίου που έχουμε ορίσει (*threshold*). Το θετικό του τρόπου αυτού είναι ότι λειτουργεί σαν να μας έστελνε δεδομένα συνεχόμενα το σύστημα ενώ δεν μας στέλνει. Η εικόνα 19 μας δείχνει το block-diagram τρόπου λειτουργίας



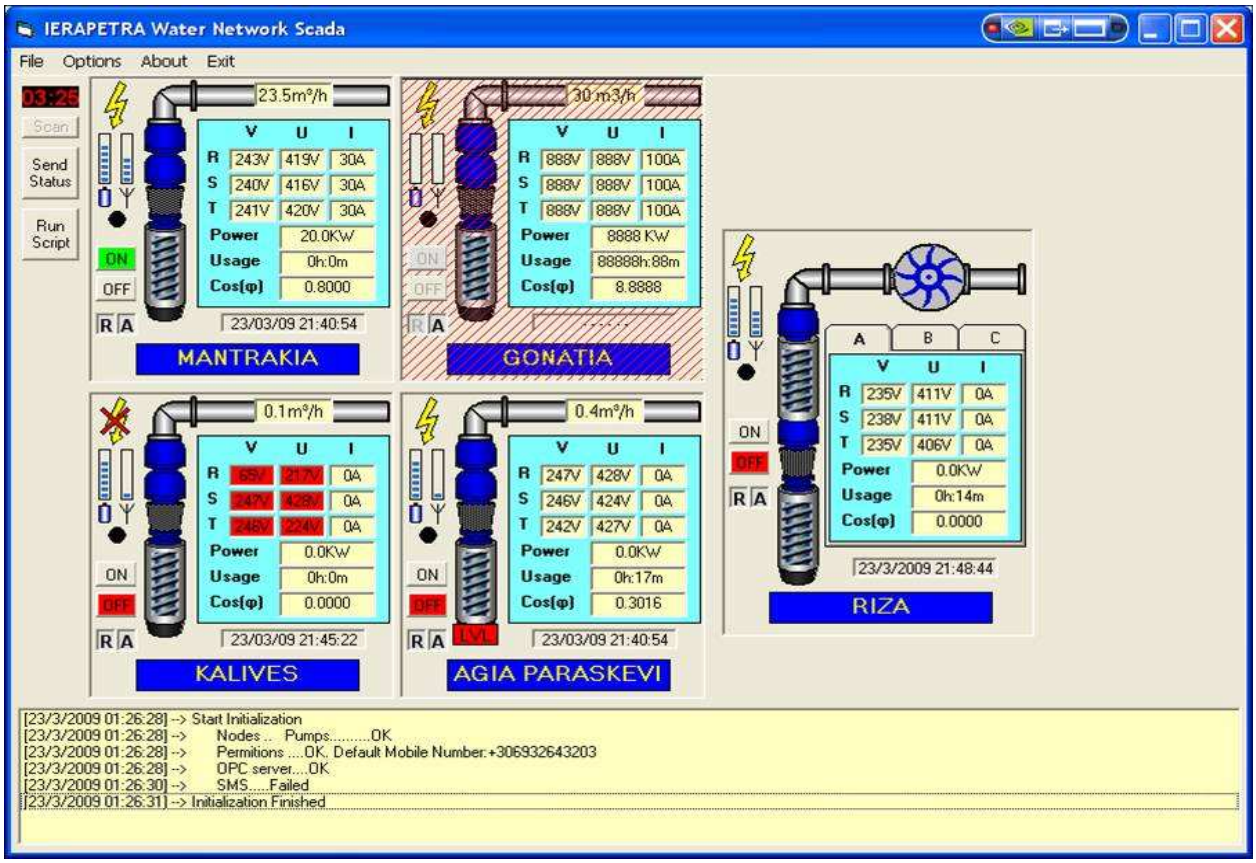
Εικόνα 19 Block - Diagram Τρόπου Λειτουργίας

Η εικόνα 20 μας δείχνει πως δουλέυει ο “έξυπνος τρόπος” καταγραγραφής δεδομένων



Εικόνα 20 Παράδειγμα εφαρμογής "Έξυπνου" Τρόπου

Έτσι έχουμε μια πραγματικότητα της κατάστασης της βάσης η οποία είναι *ψεύτικα* φτιαγμένη αλλα είναι *αληθινή*. Αυτό είναι σημαντικό γιατί δεν χρειάζεται ο χρήστης να συνδέεται στο δίκτυο άσκοπα και να επιβαρύνεται το συνολικό κόστος. Για το λόγο φτιάχνουμε ένα αντίγραφο της opc.mdb βάσης σύμφωνα με τις επιλογές του χρήστη. Το σύστημα ελέγχει κάθε φορά αν υπάρχει το αντίγραφο αυτό αλλιώς δημιουργείται αυτόματα κάθε πρώτη του μήνα Στην εικόνα 21 βλέπουμε ένα στιγμιότυπο του προγράμματος και επιλέγοντας **File → Database**

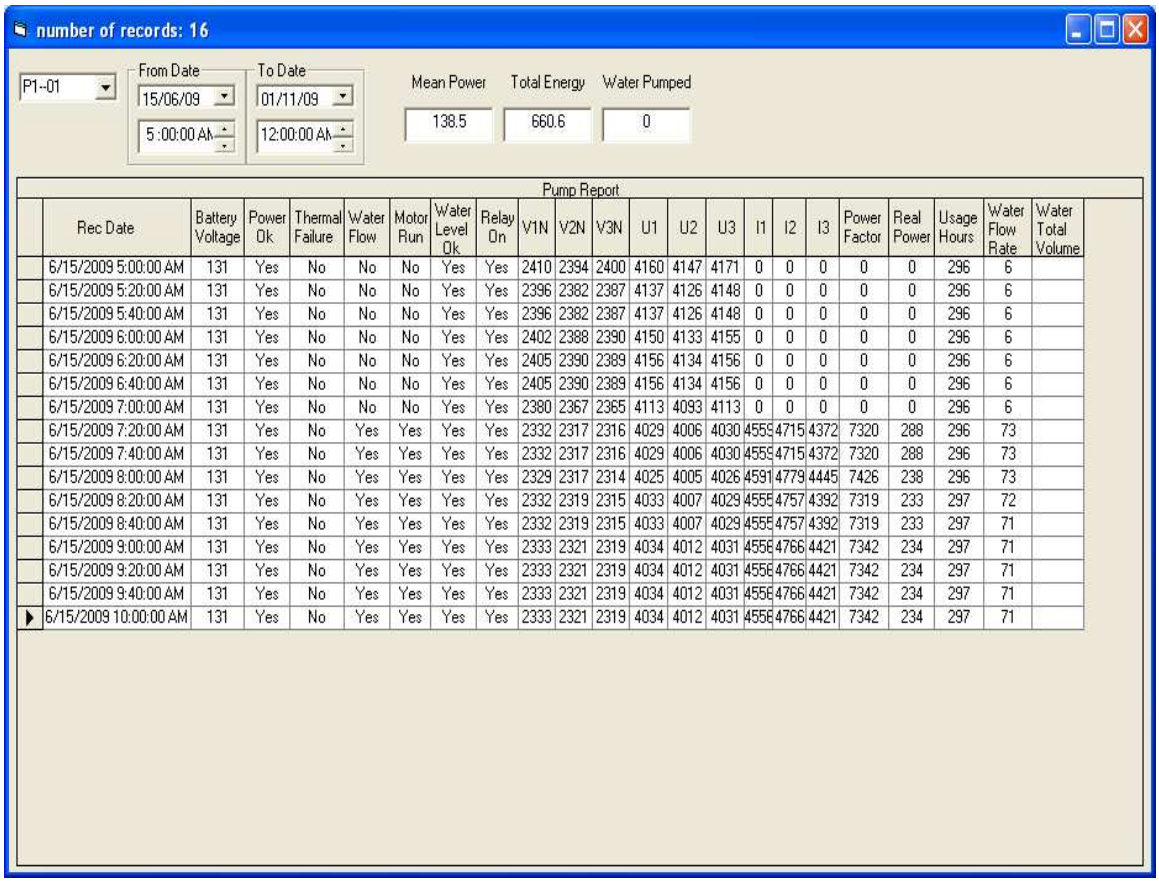


Εικόνα 21 Screenshot Εφαρμογής SCADA Ιεράπετρας

τότε εμφανίζεται ένα καινούργιο παράθυρο με διάφορες επιλογές.Αυτό το παράθυρο φαίνεται στην εικόνα 22. Μόλις ο χρήστης ανοίξει την εφαρμογή έχει διάφορες επιλογές. Επιλέγει ένα χρονικό διάστημα ημερομηνίας, ώρας και αντλίας προκειμένου να τραβήξει από την βάση τις εγγραφές των δεδομένων. Το πρόγραμμα μας σύμφωνα με τις επιλογές του χρήστη δημιουργεί μια καινούργια βάση,αν δεν υπάρχει ήδη. Στη βάση αυτή τοποθετούνται τα δεδομένα από την αρχική βάση (opc) σύμφωνα με μια συγκεκριμένη λογική από μια συνάρτηση που δημιουργήσαμε. Η βάση αυτή γεμίζει συνέχεια με δεδομένα διαφόρων τύπων όπως :τάση,ρεύμα,κατάσταση μπαταρίας,κλπ.

Η συνάρτηση ψάχνει στην opc.mdb βάση τις τελευταίες χρονικά εγγραφές (για κάθε στοιχείο) σε σχέση με το χρονικό διάστημα που δώσαμε πιο πριν. Αν δεν υπάρχουν εγγραφές αφήνετε κενό. Οι τιμές αυτές τοποθετούνται στο component (data grid) με συγκεκριμένο χρονικό βήμα (20 λεπτά) που

παραμένει ίδιο σε όλο το διάστημα επιλογής ημερομηνίας και ώρας. Στη συνέχεια οπτικοποιούνται τα δεδομένα μας μέσω του datagrid. Τα δεδομένα φαίνονται στο pump report (στοιχεία αντλίας). Επιπλέον φτιάξαμε 3 συναρτήσεις. Η mean power αθροίζει τις τιμές του πεδίου real power στο χρονικό διάστημα που έχουμε δώσει και το διαιρεί με το πλήθος των εγγραφών έτσι ώστε να προκύψει η μέση ισχύ. Η total energy χρησιμοποιεί το χρονικό βήμα και το διαιρεί με 60 (λεπτά τις ώρας). Το αποτέλεσμα το πολλαπλασιάζει με το άθροισμα των τιμών του πεδίου real power. Η τελευταία συνάρτηση (Water Total Volume) μας δείχνει την κατανάλωση του νερού στο χρονικό διάστημα που του δίνει ο χρήστης.
Παρακάτω βλέπουμε ένα στιγμιότυπο από την εφαργμογή που φτίαξαμε.Στα αριστερά βλέπουμε ότι μπορούμε να διαλέξουμε αντλία,πιο δεξιά τις ημερομηνίες (από – έως). Διακρίνονται επίσης οι τιμές στις 3 συναρτήσεις που δημιουργήσαμε οι οποίες είναι
Mean Power, Total Energy, Water Total Volume.

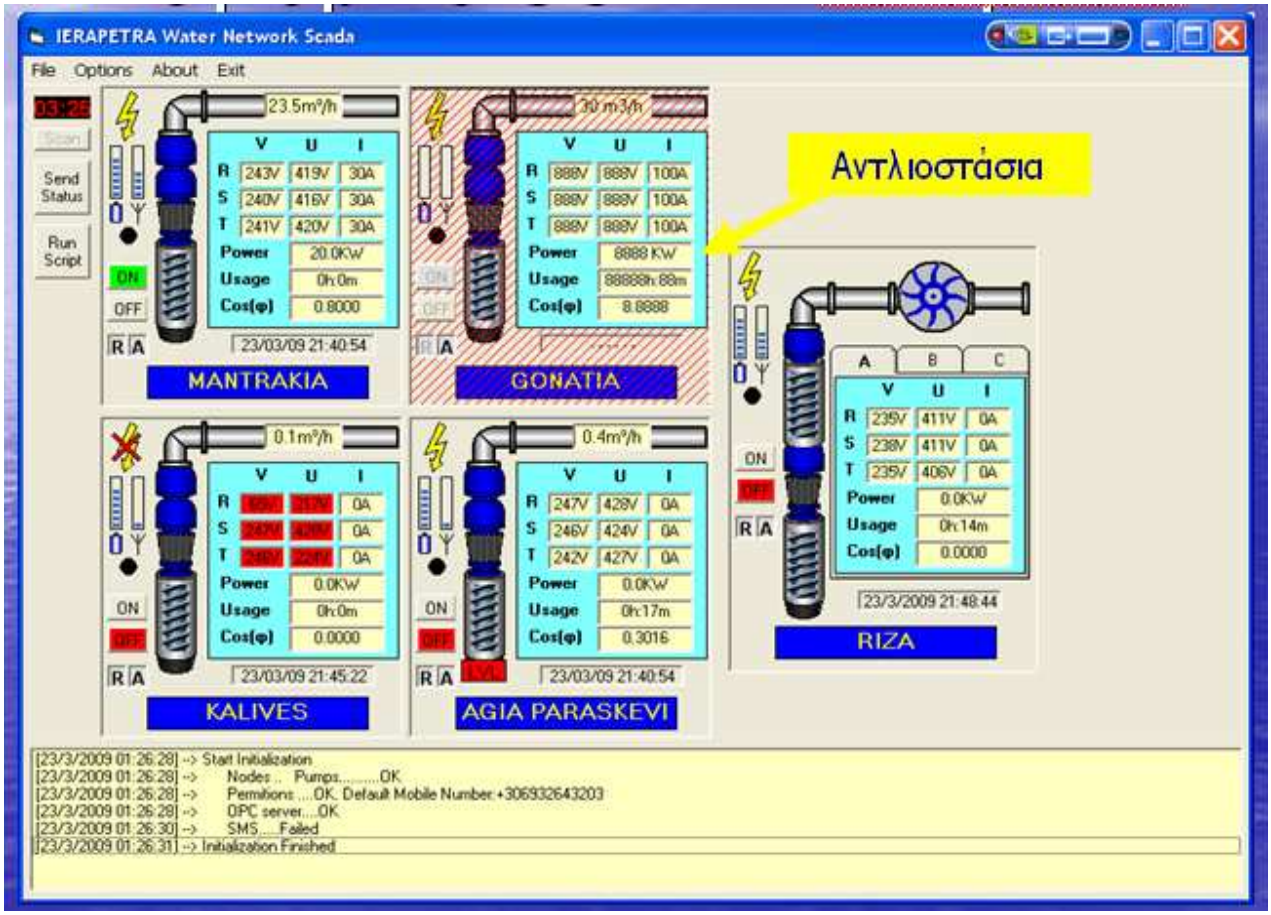


Rec Date	Battery Voltage	Power Ok	Thermal Failure	Water Flow	Motor Run	Water Level Ok	Relay On	V1N	V2N	V3N	U1	U2	U3	I1	I2	I3	Power Factor	Real Power	Usage Hours	Water Flow Rate	Water Total Volume
6/15/2009 5:00:00 AM	131	Yes	No	No	No	Yes	Yes	2410	2394	2400	4160	4147	4171	0	0	0	0	0	296	6	
6/15/2009 5:20:00 AM	131	Yes	No	No	No	Yes	Yes	2396	2382	2387	4137	4126	4148	0	0	0	0	0	296	6	
6/15/2009 5:40:00 AM	131	Yes	No	No	No	Yes	Yes	2396	2382	2387	4137	4126	4148	0	0	0	0	0	296	6	
6/15/2009 6:00:00 AM	131	Yes	No	No	No	Yes	Yes	2402	2388	2390	4150	4133	4155	0	0	0	0	0	296	6	
6/15/2009 6:20:00 AM	131	Yes	No	No	No	Yes	Yes	2405	2390	2389	4156	4134	4156	0	0	0	0	0	296	6	
6/15/2009 6:40:00 AM	131	Yes	No	No	No	Yes	Yes	2405	2390	2389	4156	4134	4156	0	0	0	0	0	296	6	
6/15/2009 7:00:00 AM	131	Yes	No	No	No	Yes	Yes	2380	2367	2365	4113	4093	4113	0	0	0	0	0	296	6	
6/15/2009 7:20:00 AM	131	Yes	No	Yes	Yes	Yes	Yes	2332	2317	2316	4029	4006	4030	4559	4715	4372	7320	288	296	73	
6/15/2009 7:40:00 AM	131	Yes	No	Yes	Yes	Yes	Yes	2332	2317	2316	4029	4006	4030	4559	4715	4372	7320	288	296	73	
6/15/2009 8:00:00 AM	131	Yes	No	Yes	Yes	Yes	Yes	2329	2317	2314	4025	4005	4026	4591	4779	4445	7426	238	296	73	
6/15/2009 8:20:00 AM	131	Yes	No	Yes	Yes	Yes	Yes	2332	2319	2315	4033	4007	4029	4555	4757	4392	7319	233	297	72	
6/15/2009 8:40:00 AM	131	Yes	No	Yes	Yes	Yes	Yes	2332	2319	2315	4033	4007	4029	4555	4757	4392	7319	233	297	71	
6/15/2009 9:00:00 AM	131	Yes	No	Yes	Yes	Yes	Yes	2333	2321	2319	4034	4012	4031	4556	4766	4421	7342	234	297	71	
6/15/2009 9:20:00 AM	131	Yes	No	Yes	Yes	Yes	Yes	2333	2321	2319	4034	4012	4031	4556	4766	4421	7342	234	297	71	
6/15/2009 9:40:00 AM	131	Yes	No	Yes	Yes	Yes	Yes	2333	2321	2319	4034	4012	4031	4556	4766	4421	7342	234	297	71	
6/15/2009 10:00:00 AM	131	Yes	No	Yes	Yes	Yes	Yes	2333	2321	2319	4034	4012	4031	4556	4766	4421	7342	234	297	71	

Εικόνα 22 Πρόγραμμα Πτυχιακής

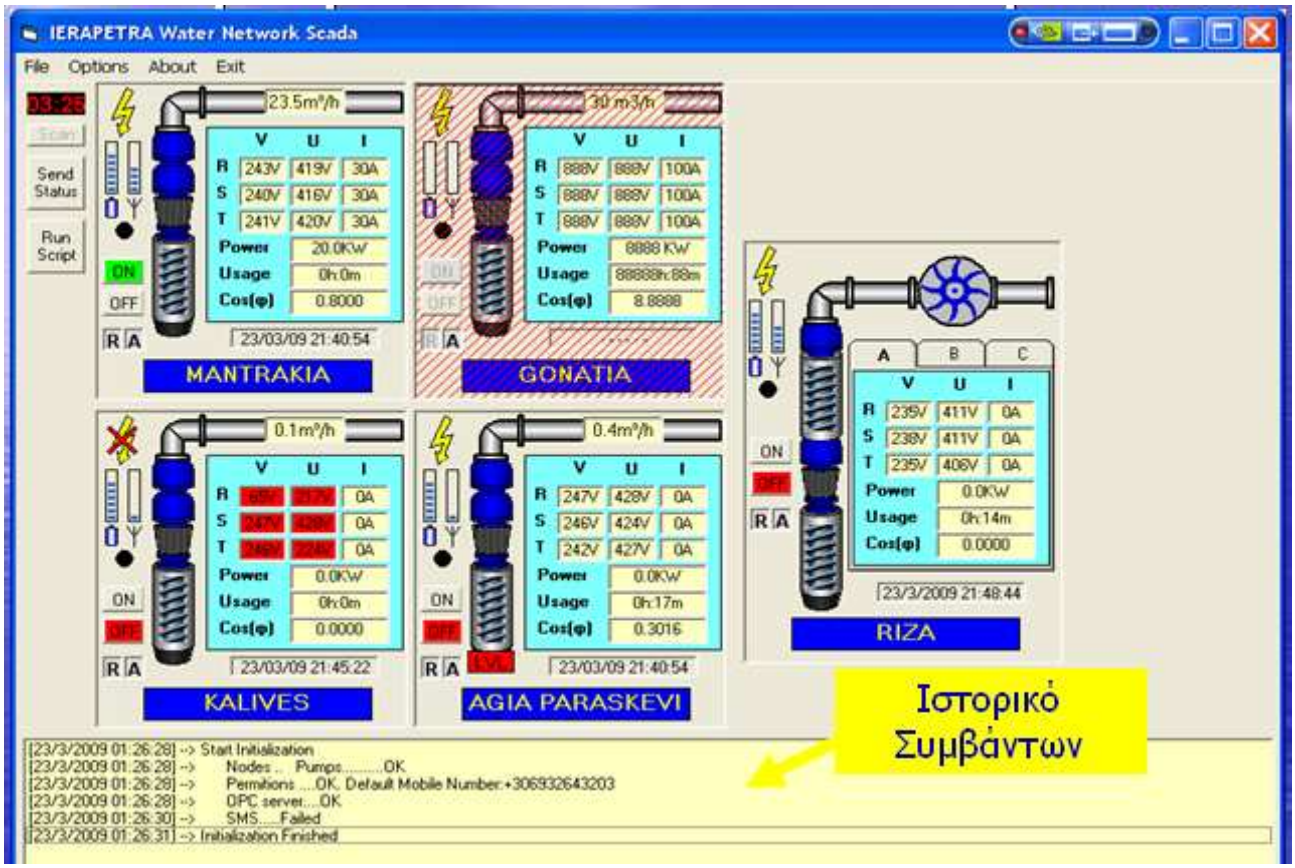
3.5 Λογισμικό Scada Ιεράπετρας

Στις παρακάτω εικόνες βλέπουμε στιγμιότυπα από την εφαρμογή SCADA και τι δυνατότητες χειρισμού και ενημέρωσης μπορεί να έχει ο χρήστης. Στην εικόνα 23 και σε όλες τις υπόλοιπες εικόνες του κεφαλαίου 3.5 απεικονίζονται όλες οι αντλίες, το menu της εφαρμογής, το ιστορικό συμβάντων, την επιλογή αποστολής SMS κατάστασης και την εκτέλεση σεναριου (script). Επιπλέον ο κάθε κόμβος-αντλία έχει τα δικά του χαρακτηριστικά.



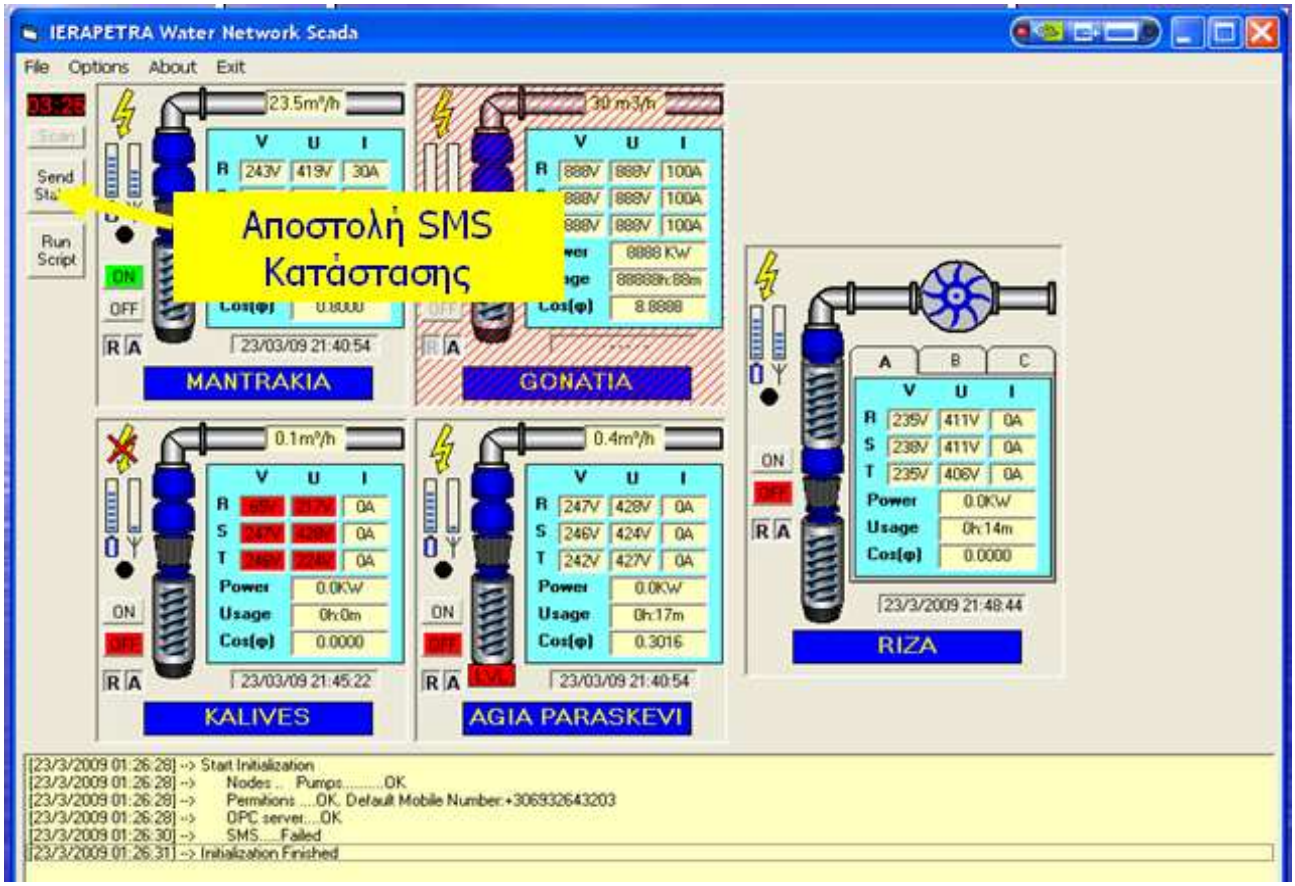
Εικόνα 23 Λογισμικό SCADA Ierapetra – Αντλιοστάσια

Στην εικόνα 24 απεικονίζεται το ιστορικό συμβάντων. Σε αυτό οτιδήποτε και αν συμβεί στην εφαρμογή καταγράφεται και εμφανίζεται στο πλαίσιο αυτό. Με αυτό τον τρόπο ο χρήστης γνωρίζει ολόκληρο το ιστορικό καταστάσης της εφαρμογής.



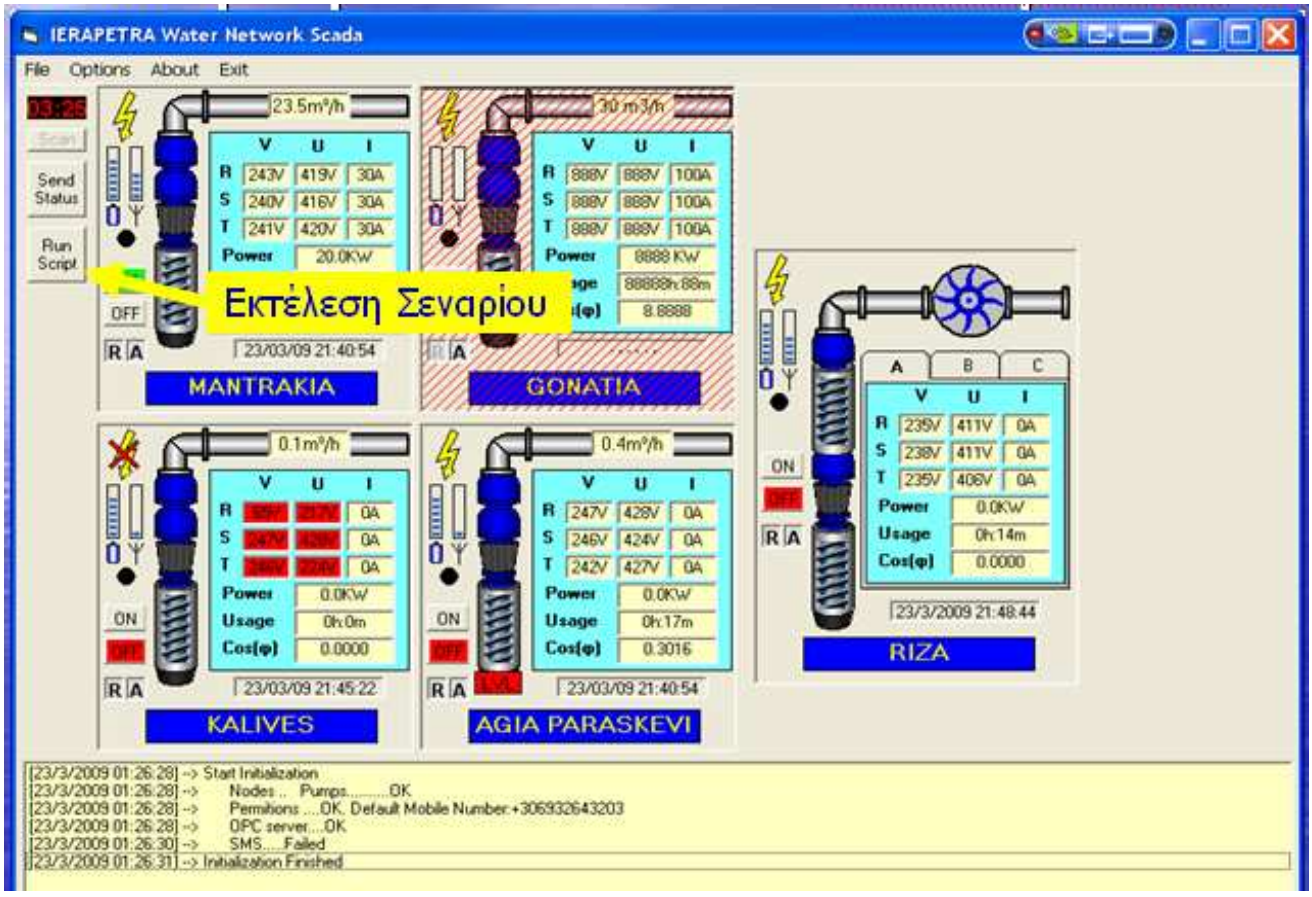
Εικόνα 24 Λογισμικό SCADA Ierapetra - Ιστορικό Συμβάντων

Στην εικόνα 25 απεικονίζεται το πλήκτρο που δείχνει αν είναι ενεργοποιημένη η δυνατότητα αποστολής και λήψης SMS μηνυμάτων.



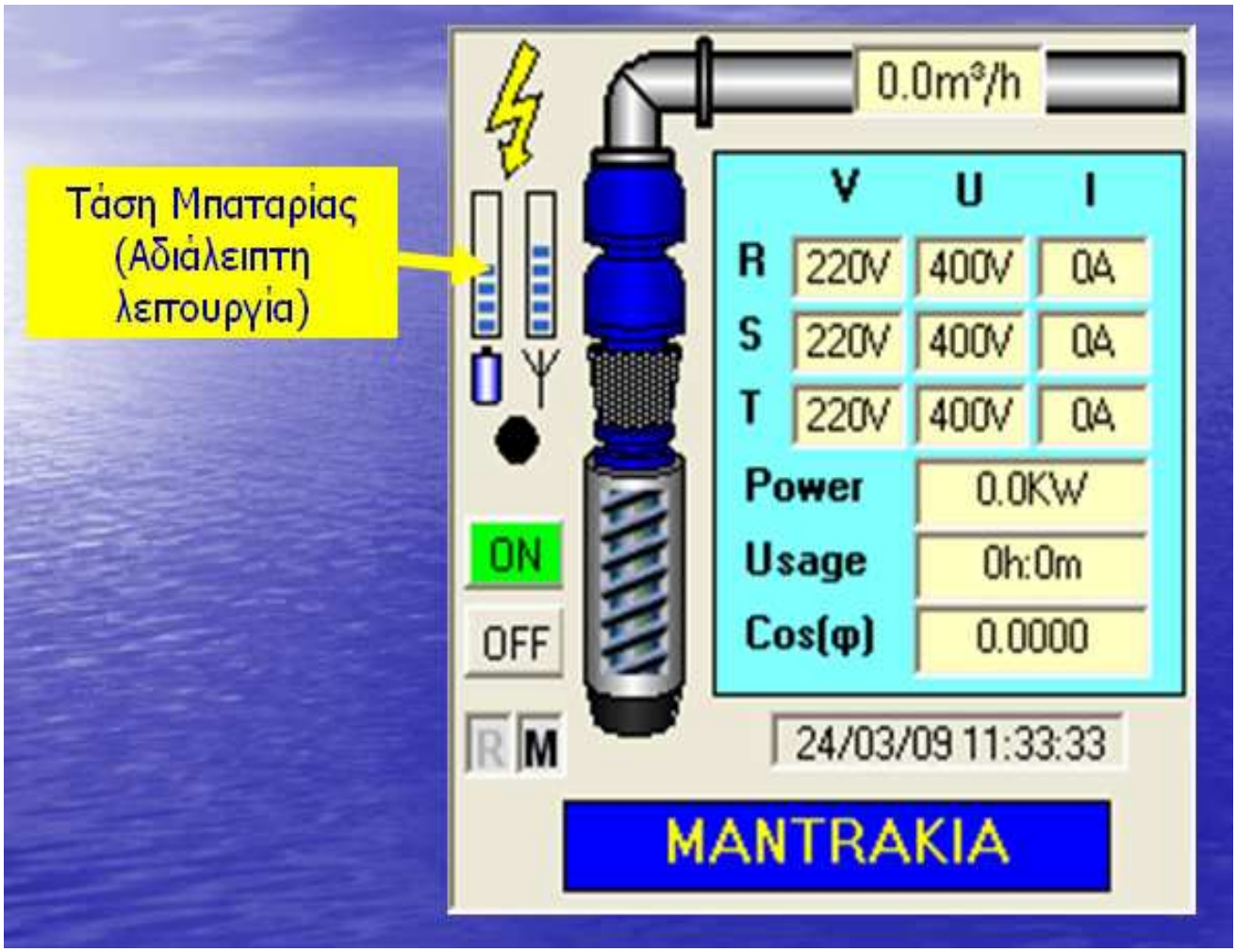
Εικόνα 25 Λογισμικό SCADA Ierapetra - Αποστολή SMS

Στην εικόνα 26 απεικονίζεται το κουμπί εκτέλεσης σεναρίου. Ο χρήστης μπορεί να γράψει ένα σενάριο αυτοματισμού σε ψευδοκώδικα. Για παράδειγμα μπορεί να θέλει όταν μια αντλία κλείσει τότε να δοθεί εντολή να ανοίξει μια άλλη αντλία. Αυτό είναι ένα πολύ απλό σενάριο αυτοματισμού.



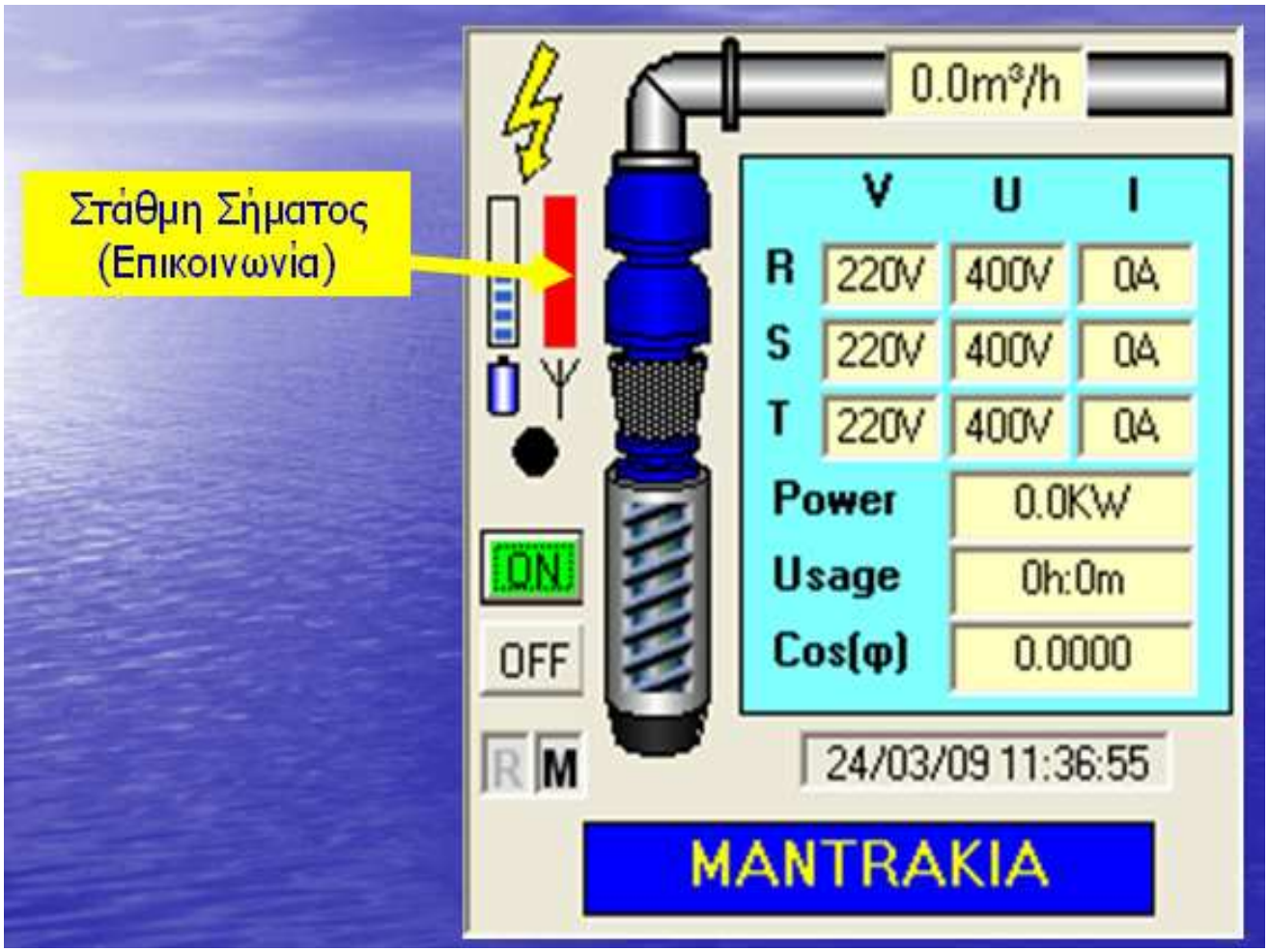
Εικόνα 26 Λογισμικό SCADA Ierapetra - Εκτέλεση Σεναρίου

Στην εικόνα 27 απεικονίζεται η τάση της μπαταρίας. Το σύστημα δείχνει αν έχει μπαταρία έτσι ώστε αν κοπεί το ρεύμα να υπάρχει επικοινωνία. Η τάση της μπαταρίας φαίνεται στην αριστερή μπάρα



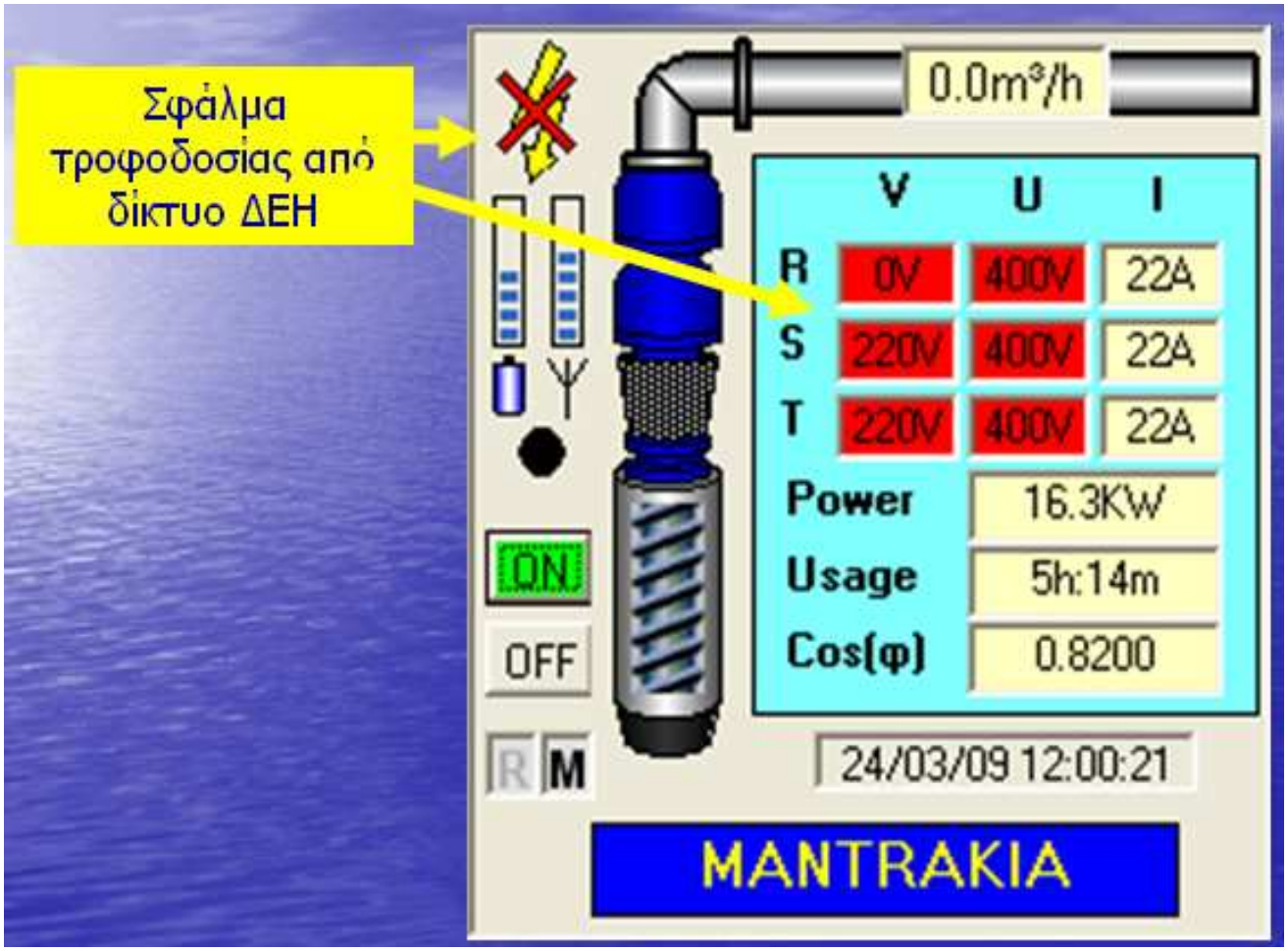
Εικόνα 27 Λογισμικό SCADA Ierapetra - Τάση Μπαταρίας

Στην εικόνα 28 φαίνεται η στάθμη του σήματος. Η τελευταία επικοινωνία με το σύστημα τι εντάση σήματος είχε. Στην συγκεκριμένη εικόνα έχει χαθεί η επικοινωνία επειδή η μπάρα είναι κόκκινη.



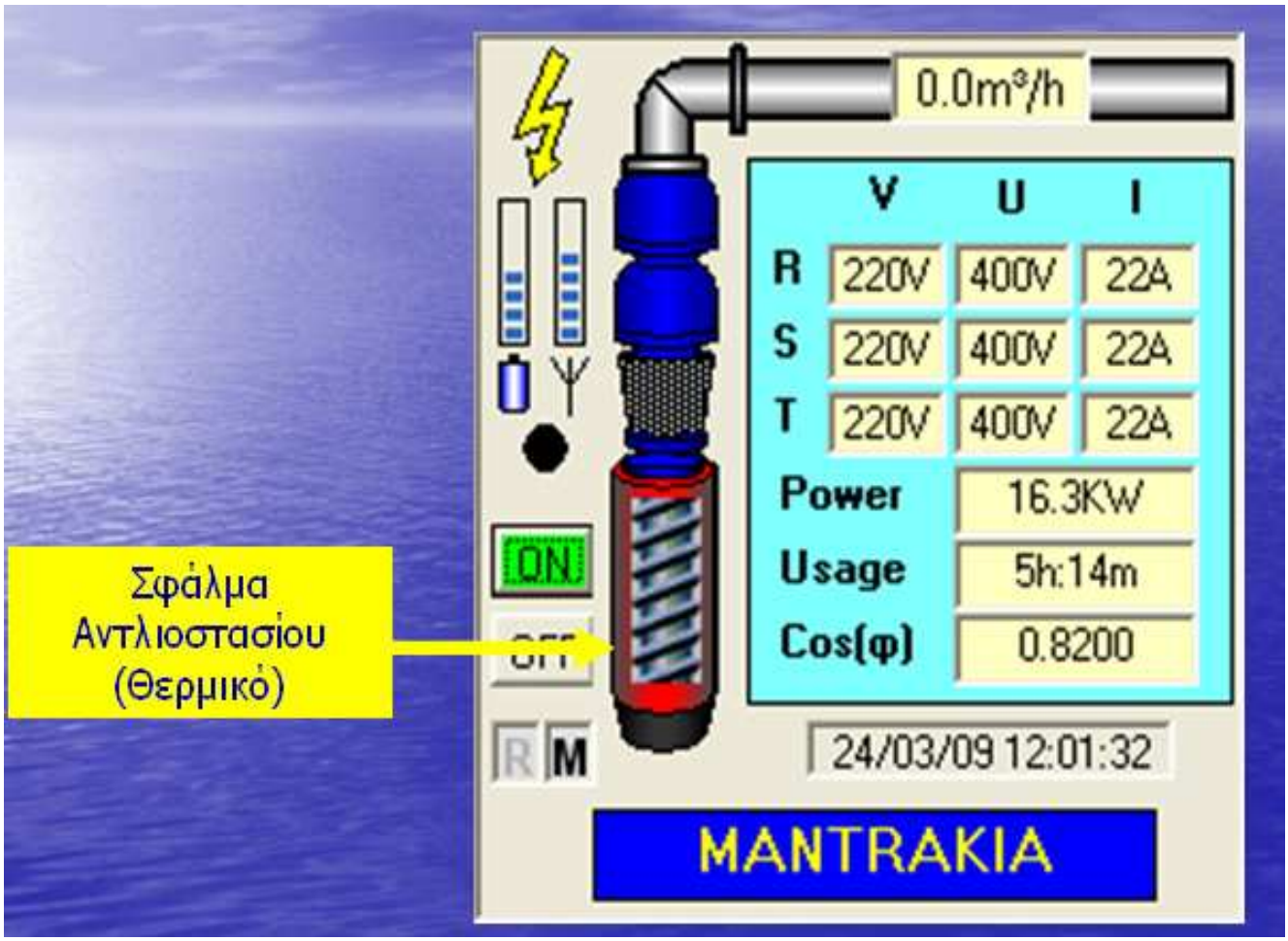
Εικόνα 28 Λογισμικό SCADA Ierapetra - Στάθμη Σήματος

Σε περίπτωση σφάλματος τροφοδοσίας εμφανίζεται ένα X (με κόκκινο χρώμα) πάνω στο σύμβολο ένδειξης τροφοδοσίας από το δίκτυο της ΔΕΗ.
Στη συγκεκριμένη εικόνα υπάρχει σφάλμα στην τάση V1 της φάσης R.
Οι άλλες δύο φάσεις έχουν κανονική τροφοδοσία. Οι τρεις φάσεις είναι οι R, S, T



Εικόνα 29 Λογισμικό SCADA Ierapetra -Σφάλμα τροφοδοσίας από δίκτυο ΔΕΗ

Τα συστήματα έχουν ένα θερμικό σε περίπτωση υπερφόρτωσης για να μην καταστραφεί η αντλία. Η κάθε αντλία έχει από ένα τέτοιο θερμικό. Η εικόνα 30 μας δείχνει ότι υπάρχει σφάλμα στο αντλιοστάσιο και ότι έχουμε πτώση θερμικού.



Εικόνα 30 Λογισμικό SCADA Ierapetra -Σφάλμα Αντλιοστασίου (Θερμικό)

Κεφάλαιο 4 Κώδικας πτυχιακής

4.1 Κώδικας φόρμας

```
Option Explicit
Dim dbFilename As String
Dim nodeid As String
Dim rs As New ADODB.Recordset
Dim cn As ADODB.Connection
Dim TotalEnergy, MeanPower, WaterPumped As Single

Private Sub dtpStartDate_change()
If Me.dtpEndDate.Value < Me.dtpStartDate.Value Then
   Me.dtpEndDate.Value = Me.dtpStartDate.Value + TimeSerial(24, 0, 0)
   End If
   refreshRs
refreshRs
End Sub
Private Sub dtpEndDate_change()
   If Me.dtpEndDate.Value < Me.dtpStartDate.Value Then
   Me.dtpStartDate.Value = Me.dtpEndDate.Value - TimeSerial(24, 0, 0)
   End If
   refreshRs
End Sub
Private Sub dtpStartHour_change()
   refreshRs
End Sub
Private Sub dtpEndHour_change()
   refreshRs
End Sub
Private Sub Form_Load()
Dim power As Double
   InitDbase
   initDataGrid
   Me.cbPump.AddItem "P1--01"
```

```
    Me.cbPump.AddItem "P2--02"
    Me.cbPump.AddItem "P3--03"
    Me.cbPump.AddItem "P4--04"
    Me.cbPump.AddItem "P5--05"
End Sub
Public Sub cbPump_Click()
    refreshRs
End Sub
Public Sub refreshRs()
    Dim sqlstr As String
    Dim power As Double
    Dim energia As Double
    Dim water As Double
    Debug.Print Format(Now, "mm:ss") & "**********refreshing**********"
    sqlstr = "SELECT * From Report WHERE (RecDate)>= #" & Format(Me.dtpStartDate.Value, "mm/dd/yy") _
        & " " & Format(Me.dtpStartHour.Value, "hh:mm:ss") & _
        "# and (RecDate)<= #" & Format(Me.dtpEndDate.Value, "mm/dd/yy") & " " & _
        Format(Me.dtpEndHour.Value, "hh:mm:ss") & "# and Report.NodeID = '" & Me.cbPump.Text & "' "
    Debug.Print "2"; sqlstr
    If rs.State = adStateOpen Then
        rs.Close
    End If
    rs.Open sqlstr, cn, adUseClient, adLockOptimistic
    Set dgPumpLog.DataSource = rs
    initDataGrid
    Me.Caption = "number of records: " & rs.RecordCount
    CalcTotals
    Me.txtTotalEnergy.Text = TotalEnergy
    Me.txtMeanPower.Text = MeanPower
    Me.txtWaterPumped.Text = WaterPumped
    dgPumpLog.Refresh
```

```
End Sub
Public Sub initDataGrid()
    Dim i As Integer
    Dim oFormat As StdDataFormat
    Set oFormat = New StdDataFormat
    oFormat.TrueValue = "Yes"
    oFormat.FalseValue = "No"
    oFormat.Type = fmtBoolean
    Set dgPumpLog.Columns(3).DataFormat = oFormat
    Set dgPumpLog.Columns(4).DataFormat = oFormat
    Set dgPumpLog.Columns(5).DataFormat = oFormat
    Set dgPumpLog.Columns(6).DataFormat = oFormat
    Set dgPumpLog.Columns(7).DataFormat = oFormat
    Set dgPumpLog.Columns(8).DataFormat = oFormat
    For i = 0 To 23
       dgPumpLog.Columns(i).Alignment = dbgCenter
    Next i


    dgPumpLog.Columns(0).Width = 1800
    dgPumpLog.Columns(0).Caption = "       Rec Date"
    dgPumpLog.Columns(0).DataField = "RecDate"
    dgPumpLog.Columns(1).Width = 0
    dgPumpLog.Columns(1).Caption = "NodeId"
    dgPumpLog.Columns(1).DataField = "NodeId"
    dgPumpLog.Columns(2).Width = 640
    dgPumpLog.Columns(2).Caption = "Battery Voltage"
    dgPumpLog.Columns(2).DataField = "BatteryVoltage"

    dgPumpLog.Columns(3).Width = 550
    dgPumpLog.Columns(3).Caption = "Power    Ok"
    dgPumpLog.Columns(3).DataField = "PowerOK"

    dgPumpLog.Columns(4).Width = 660
```

```
dgPumpLog.Columns(4).Caption = "Thermal Failure"
dgPumpLog.Columns(4).DataField = "ThermalFailure"

dgPumpLog.Columns(5).Width = 550
dgPumpLog.Columns(5).Caption = "Water  Flow"
dgPumpLog.Columns(5).DataField = "WaterFlow"

dgPumpLog.Columns(6).Width = 500
dgPumpLog.Columns(6).Caption = "Motor  Run"
dgPumpLog.Columns(6).DataField = "MotorRun"

dgPumpLog.Columns(7).Width = 550
dgPumpLog.Columns(7).Caption = "Water Level     Ok"
dgPumpLog.Columns(7).DataField = "WaterLevelOk"


dgPumpLog.Columns(8).Width = 510
dgPumpLog.Columns(8).Caption = "Relay    On"
dgPumpLog.Columns(8).DataField = "RelayOn"



dgPumpLog.Columns(9).Width = 450
dgPumpLog.Columns(9).Caption = "V1N"
dgPumpLog.Columns(9).DataField = "V1N"
dgPumpLog.Columns(10).Width = 450
dgPumpLog.Columns(10).Caption = "V2N"
dgPumpLog.Columns(10).DataField = "V2N"
dgPumpLog.Columns(11).Width = 450
dgPumpLog.Columns(11).Caption = "V3N"
dgPumpLog.Columns(11).DataField = "V3N"

dgPumpLog.Columns(12).Width = 460
dgPumpLog.Columns(12).Caption = "  U1"
```

```
dgPumpLog.Columns(12).DataField = "U1"
dgPumpLog.Columns(13).Width = 460
dgPumpLog.Columns(13).Caption = "  U2"
dgPumpLog.Columns(13).DataField = "U2"
dgPumpLog.Columns(14).Width = 460
dgPumpLog.Columns(14).Caption = "  U3"
dgPumpLog.Columns(14).DataField = "U3"

dgPumpLog.Columns(15).Width = 350
dgPumpLog.Columns(15).Caption = "  I1"
dgPumpLog.Columns(15).DataField = "I1"
dgPumpLog.Columns(16).Width = 410
dgPumpLog.Columns(16).Caption = "  I2"
dgPumpLog.Columns(16).DataField = "I2"
dgPumpLog.Columns(17).Width = 410
dgPumpLog.Columns(17).Caption = "  I3"
dgPumpLog.Columns(17).DataField = "I3"


dgPumpLog.Columns(18).Width = 600
dgPumpLog.Columns(18).Caption = "Power  Factor"
dgPumpLog.Columns(18).DataField = "PowerFactor"

dgPumpLog.Columns(19).Width = 550
dgPumpLog.Columns(19).Caption = "Real   Power"
dgPumpLog.Columns(19).DataField = "RealPower"


dgPumpLog.Columns(20).Width = 600
dgPumpLog.Columns(20).Caption = "Usage Hours"
dgPumpLog.Columns(20).DataField = "UsageHours"

dgPumpLog.Columns(21).Width = 580
dgPumpLog.Columns(21).Caption = "Water Flow Rate"
```

```
        dgPumpLog.Columns(21).DataField = "WaterFlowRate"

        dgPumpLog.Columns(22).Width = 630
        dgPumpLog.Columns(22).Caption = "Water Total Volume"
        dgPumpLog.Columns(22).DataField = "WaterTotalVolume"

        dgPumpLog.Columns(23).Width = 0
        dgPumpLog.Columns(23).Caption = "reg109"
        dgPumpLog.Columns(23).DataField = "REG109"

End Sub
Public Sub creationdb()
Dim dbFilename As String
Dim runTime As Date
Dim delDate As Date
Dim energia As Double
Dim power As Double
Dim period As Integer
Dim water As Double
Dim oRs As New ADODB.Recordset
Dim adoConn As ADODB.Connection
Set adoConn = New ADODB.Connection
Dim sdate As Date
Dim sdate_hour As Date
Dim Edate As Date
Dim edate_hour As Date
runTime = Now
sdate = "15/6/2009"
sdate_hour = "18:00:00"
Edate = "15/6/2009"
edate_hour = "20:00:00"
cbPump.Text = "P4--04"
 Do
        createDummyBase sdate, Format(sdate_hour, "hh:mm:ss"), Edate,
Format(edate_hour, "hh:mm:ss"), 20, cbPump.Text
```

```
        dbFilename = "Report_" & Format(Now, "MM_YY") & ".mdb"
        adoConn.ConnectionString = BuildConnectionString(dbFilename)
        adoConn.Open '(default)
        oRs.Open "select * from Report", adoConn, adUseClient, adLockOptimistic
        Set dgPumpLog.DataSource = oRs
        dgPumpLog.Refresh
        Debug.Print oRs.RecordCount
        'power = meanPower2 '(sdate, Format(sdate_hour, "hh:mm:ss"), Edate,
Format(edate_hour, "hh:mm:ss"), cbPump.Text, dbFilename)
        'energia = TotalEnergy(sdate, Format(sdate_hour, "hh:mm:ss"), Edate,
Format(edate_hour, "hh:mm:ss"), cbPump.Text, dbFilename)
        'water = WaterPumped(sdate, Format(sdate_hour, "hh:mm:ss"), Edate,
Format(edate_hour, "hh:mm:ss"), dbFilename, cbPump.Text)
        'delDatabase App.Path & "\data\opc.mdb", sDate, eDate
        'DoEvents
        sdate = DateAdd("d", 1, sdate) 'add a day
    Loop Until sdate + sdate_hour > Edate + edate_hour
    runTime = Now - runTime
    Debug.Print "Time elapsed:" & runTime
End Sub
Private Sub InitDbase()
    dbFilename = "Report_" & Format(Now, "MM_YY") & ".mdb"
    Set rs = New ADODB.Recordset
    Set cn = New ADODB.Connection
    cn.ConnectionString = BuildConnectionString(dbFilename)
    cn.Open
End Sub
Private Function CalcTotals()
    Dim AccPower, previousRealPower As Single
    Dim period As Single
    Dim N As Integer
    Dim PreviousRecdate As Date
    WaterPumped = 0
    TotalEnergy = 0
```

```
   MeanPower = 0
   If rs.RecordCount > 1 Then
      rs.MoveFirst
      N = 0
      While Not rs.EOF
         If IsNumeric((rs!realpower)) Then AccPower = AccPower + Val(rs!realpower)
         N = N + 1
         rs.MoveNext
      Wend
      MeanPower = Format(AccPower / N, "0.0")
   End If
   If rs.RecordCount >= 2 Then
      rs.MoveFirst
      If IsNumeric(rs!WaterTotalVolume) Then WaterPumped = -rs!WaterTotalVolume
      PreviousRecdate = rs!recdate
      For N = 1 To rs.RecordCount - 1
         previousRealPower = rs!realpower
         rs.MoveNext
         period = Minute(rs!recdate - PreviousRecdate)
         TotalEnergy = Format(TotalEnergy + previousRealPower * period / 60, "0.0")
      Next N
      rs.MoveLast
       If IsNumeric(rs!WaterTotalVolume) Then WaterPumped = WaterPumped +
rs!WaterTotalVolume
   End If
End Function
```

4.2 Κώδικας Module

```
Option Explicit

Public Sub createDummyBase(ByVal StartDate As Date, ByVal StartHour As Date,
ByVal EndDate As Date, ByVal EndHour As Date, ByVal period As Integer, ByVal
nodeid As String)
   Dim cn As ADODB.Connection 'opc
```

```
    Dim rs As ADODB.Recordset
    Dim sqlstr As String
    Dim dbFilename As String
    Dim orig_path, new_path As String
    Dim RunDate As Date
    Dim delDate As Date
    dbFilename = "Report_" & Format(Now, "MM_YY") & ".mdb"
    If Dir(App.Path & "\data\Report_" & Format(Now, "MM_YY") & ".mdb") <> "" Then
       Else
        orig_path = App.Path & "\data\Report_mm_yy.mdb"
        new_path = App.Path & "\data\Report_" & Format(Now, "MM_YY") & ".mdb"
        FileCopy orig_path, new_path
    End If
    Set cn = New ADODB.Connection
    cn.ConnectionString = BuildConnectionString(dbFilename)
    Set rs = New ADODB.Recordset
    cn.Open
    rs.LockType = adLockOptimistic
    Set rs.ActiveConnection = cn
    rs.CursorType = adOpenKeyset
    rs.LockType = adLockOptimistic
    rs.Open "Report", cn
    RunDate = StartDate + StartHour
    Do
       Debug.Print "Checking date..."; RunDate
       If Not checkRsExist(dbFilename, RunDate, nodeid) Then
          rs.AddNew
          rs!recdate = RunDate
          rs!nodeid = nodeid
          rs!BatteryVoltage = Format(findLastValue(RunDate, "BatteryVoltage", nodeid) / 10, "0.0")
          rs!i1 = Format(findLastValue(RunDate, "I1", nodeid) / 100, "0")
```

```
        rs!i2 = Format(findLastValue(RunDate, "I2", nodeid) / 100, "0")
        rs!I3 = Format(findLastValue(RunDate, "I3", nodeid) / 100, "0")
        rs!MotorRun = findLastValue(RunDate, "MotorRun", nodeid)
        rs!PowerFactor = Format(findLastValue(RunDate, "PowerFactor", nodeid) /
10000, "0.00")
        'rs!PowerFailure = findLastValue(RunDate, "PowerFailure", nodeid)
        rs!PowerOk = findLastValue(RunDate, "PowerOk", nodeid)
        rs!realpower = findLastValue(RunDate, "RealPower", nodeid)
        rs!REG109 = findLastValue(RunDate, "REG109", nodeid)
        rs!RelayOn = findLastValue(RunDate, "RelayOn", nodeid)
        rs!Rssi = findLastValue(RunDate, "Rssi", nodeid)
        rs!ThermalFailure = findLastValue(RunDate, "ThermalFailure", nodeid)
        rs!U1 = Format(findLastValue(RunDate, "U1", nodeid) / 10, "0")
        rs!U2 = Format(findLastValue(RunDate, "U2", nodeid) / 10, "0")
        rs!U3 = Format(findLastValue(RunDate, "U3", nodeid) / 10, "0")
        rs!UsageHours = findLastValue(RunDate, "UsageHours", nodeid) + ":" +
findLastValue(RunDate, "UsageMinutes", nodeid)
       'rs!UsageMinutes = findLastValue(RunDate, "UsageMinutes", nodeid)
        rs!V1N = Format(findLastValue(RunDate, "V1N", nodeid) / 10, "0")
        rs!V2N = Format(findLastValue(RunDate, "V2N", nodeid) / 10, "0")
        rs!V3N = Format(findLastValue(RunDate, "V3N", nodeid) / 10, "0")
        rs!WaterFlow = findLastValue(RunDate, "WaterFlow", nodeid)
        rs!WaterFlowRate = findLastValue(RunDate, "WaterFlowRate", nodeid)
        rs!WaterLevelOK = findLastValue(RunDate, "WaterLevelOK", nodeid)
      End If
      RunDate = RunDate + TimeSerial(0, period, 0)
      Debug.Print "[" & RunDate & "]  [" & EndDate & "]" & (RunDate = EndDate)
'                                    (Round(RunDate,9)=Round(EndDate,9))
   Loop Until RunDate > EndDate + EndHour + TimeSerial(0, 0, 1) 'VB bug???/
   rs.Update
   cn.Close
End Sub
Private Function findLastValue(ByVal curDate As Date, ByVal fieldName As
String, ByVal nodeid As String) As String
```

```
    Dim cn As ADODB.Connection 'opc
    Dim rs As ADODB.Recordset
    Dim sqlstr As String
    Set cn = New ADODB.Connection
    Set rs = New ADODB.Recordset
    cn.ConnectionString = BuildConnectionString("opc.mdb")
    cn.Open
    sqlstr = " SELECT last(mt_read.mt_value) AS mt_value From mt_read WHERE
(((mt_read.mt_name) Like ('" & nodeid & "." & fieldName & "')) AND
((mt_read.mt_time)< #" & curDate & "#));"
    Set rs = cn.Execute(sqlstr)
    If IsNumeric(rs!mt_value) Then
       findLastValue = rs!mt_value
    Else
    End If
    cn.Close
End Function
Public Function BuildConnectionString(ByVal DBFname As String) As String
       BuildConnectionString = "Provider=MSDataShape.1;Persist Security
Info=False;Mode=ReadWrite;Data Source=" & _
          App.Path & "\data\" & DBFname & ";Data
Provider=MICROSOFT.JET.OLEDB.4.0;Persist Security Info=False"
End Function
Private Function checkRsExist(ByVal dbFilename, ByVal sdate As Date, ByVal
nodeid As String) As Boolean
    Dim cn As ADODB.Connection
    Dim sqlstr As String
    Dim rs As ADODB.Recordset
    Set cn = New ADODB.Connection
    Set rs = New ADODB.Recordset
    cn.ConnectionString = BuildConnectionString(dbFilename)
    cn.Open
    sqlstr = "SELECT * FROM Report WHERE (dateValue(Report.RecDate)=  #" &
Format$(sdate, "dd/m/yyyy") & _
```

```
   "# and TimeValue(Report.RecDate)= #" & Format$(sdate, "hh:mm:ss") & "# and
Report.NodeID = '" & nodeid & "');"
   Set rs = cn.Execute(sqlstr)
   If rs.RecordCount = 0 Then
      checkRsExist = False
   Else
      checkRsExist = True
   End If
   cn.Close
End Function
Public Sub delDatabase(ByVal dbFilename, ByVal sdate As Date, ByVal Edate As
Date)
   Dim cn As ADODB.Connection
   Dim sqlstr As String
   Set cn = New ADODB.Connection
   cn.ConnectionString = BuildConnectionString("opc.mdb")
   cn.Open
   sqlstr = "DELETE FROM mt_read WHERE mt_time < #" & Format$(Edate,
"mm/dd/yy hh:mm:ss") & "# and mt_time > #" & Format$(sdate, "mm/dd/yy
hh:mm:ss") & "#;"
   cn.Execute (sqlstr)
End Sub
Public Function TotalEnergy(ByVal StartDate As Date, ByVal StartHour As Date,
ByVal EndDate As Date, ByVal EndHour As Date, ByVal nodeid As String, ByVal
dbFilename As String) As Double
   Dim cn As ADODB.Connection
   Dim RunDate As Date
   Dim rs As ADODB.Recordset
   Dim sum As Double
   Dim calcspace As Date
   Dim period As Integer
   Dim sqlstr As String
   dbFilename = "Report_" & Format(Now, "MM_YY") & ".mdb"
   Set cn = New ADODB.Connection
```

```
   Set rs = New ADODB.Recordset
   sum = 0
   cn.ConnectionString = BuildConnectionString(dbFilename)
   cn.Open
   rs.LockType = adLockOptimistic
   Set rs.ActiveConnection = cn
   rs.CursorType = adOpenKeyset
   rs.LockType = adLockOptimistic
   rs.Open "Report", cn
   RunDate = StartDate + StartHour
   sqlstr = "SELECT * FROM Report WHERE  (Report.RecDate>= #" & StartDate
& "# and timevalue(Report.RecDate)>= #" & StartHour & "# and Report.NodeID =
'" & nodeid & "');"
   Set rs = cn.Execute(sqlstr)
      rs.MoveFirst
      Debug.Print rs!recdate
      'Debug.Print rs.RecordCount
      rs.MoveNext
      calcspace = rs!recdate - (StartDate + StartHour)
      period = Minute(calcspace)
   rs.MoveFirst
   Do
         If IsNumeric(rs!realpower) Then
            sum = sum + rs!realpower
            Debug.Print rs!realpower
            rs.MoveNext
              Else
         End If
         RunDate = RunDate + TimeSerial(0, period, 0)

   Loop Until RunDate >= EndDate + EndHour + TimeSerial(0, period, 0)
   Debug.Print sum
   TotalEnergy = (period / 60) * sum
   Debug.Print "TotalEnergy="; TotalEnergy
```

```
    cn.Close
End Function
Public Function MeanPower(ByVal StartDate As Date, ByVal StartHour As Date, ByVal EndDate As Date, ByVal EndHour As Date, ByVal nodeid As String, ByVal dbFilename As String) As Double
    Dim cn As ADODB.Connection
    Dim RunDate As Date
    Dim rs As ADODB.Recordset
    Dim sum As Double
    Dim calcspace As Date
    Dim period As Integer
    Dim sqlstr As String
    Dim i As Integer
    dbFilename = "Report_" & Format(Now, "MM_YY") & ".mdb"
    Set cn = New ADODB.Connection
    Set rs = New ADODB.Recordset
    sum = 0
    cn.ConnectionString = BuildConnectionString(dbFilename)
    cn.Open
    rs.LockType = adLockOptimistic
    Set rs.ActiveConnection = cn
    rs.CursorType = adOpenKeyset
    rs.LockType = adLockOptimistic
    rs.Open "Report", cn
    RunDate = StartDate + StartHour
    sqlstr = "SELECT * FROM Report WHERE  (Report.RecDate>= #" & StartDate & "# and timevalue(Report.RecDate)>= #" & StartHour & "# and Report.NodeID = '" & nodeid & "');"
    sqlstr = "SELECT * FROM Report WHERE  (Report.RecDate>= #" & StartDate & "# and timevalue(Report.RecDate)>= #" & StartHour & "# and Report.NodeID = '" & nodeid & "');"
    Set rs = cn.Execute(sqlstr)
    rs.MoveFirst
    Debug.Print rs!recdate
```

```
    'Debug.Print rs.RecordCount
    rs.MoveNext
    calcspace = rs!recdate - (StartDate + StartHour)
    period = Minute(calcspace)
    rs.MoveFirst
    Do
          If IsNumeric(rs!realpower) Then
             sum = sum + rs!realpower
             Debug.Print rs!realpower
             i = i + 1
             rs.MoveNext
               Else
          End If
          RunDate = RunDate + TimeSerial(0, period, 0)

    Loop Until RunDate >= EndDate + EndHour + TimeSerial(0, period, 0)
    Debug.Print sum
    MeanPower = sum / i
    Debug.Print "MeanPower="; MeanPower
    cn.Close
    Debug.Print i
End Function
Public Function WaterPumped(ByVal StartDate As Date, ByVal StartHour As Date,
ByVal EndDate As Date, ByVal EndHour As Date, ByVal dbFilename, ByVal
nodeid As String) As Double
    Dim cn As ADODB.Connection
    Dim RunDate As Date
    Dim rs As ADODB.Recordset
    Dim sum As Double
    Dim calcspace As Date
    Dim period As Integer
    Dim sqlstr As String
    Dim sqlstr1 As String
    Dim i1 As Double
```

```
    Dim i2 As Double
    dbFilename = "Report_" & Format(Now, "MM_YY") & ".mdb"
    Set cn = New ADODB.Connection
    Set rs = New ADODB.Recordset
    cn.ConnectionString = BuildConnectionString(dbFilename)
    cn.Open
    rs.LockType = adLockOptimistic
    Set rs.ActiveConnection = cn
    rs.CursorType = adOpenKeyset
    rs.LockType = adLockOptimistic
    rs.Open "Report", cn
    RunDate = StartDate
    rs.MoveFirst
    sqlstr = "SELECT * FROM Report WHERE  (Report.RecDate>= #" & StartDate
& "# and timevalue(Report.RecDate)>= #" & StartHour & "# and Report.NodeID =
'" & nodeid & "');"
    Set rs = cn.Execute(sqlstr)
    rs.MoveFirst
    If IsNumeric(rs!WaterTotalVolume) Then
        i1 = rs!WaterTotalVolume
        rs.MoveLast
        i2 = rs!WaterTotalVolume
        WaterPumped = i2 - i1
    End If
    Debug.Print "waterpumped="; WaterPumped
End Function
```

Κεφάλαιο 5 Βιβλιογραφία

Οι πληροφορίες της συγκεκριμένης πτυχιακής πάρθηκαν, μετά από κατάλληλη επεξεργασία,από τις παρακάτω ιστοσελίδες:

http://www.teiser.gr/icd/staff/vologian/files/Projects_BP/SCADA%20AND%20OPENSOURCE%20PROGRAMS%20FOR%20SCADA.ppt

http://www.inventia.pl/ENG/technologia_MT-101.htm

http://www.schneider-electric.ch/custom/upload/docs/document/63230-501-209PM710ReferenceEN.pdf

http://en.wikipedia.org/wiki/Modbus

http://www.epaggelmaties.com/writer/2001-2003/teyxos204.html

http://www.vbforums.com/

http://www.xtremevbtalk.com/

http://www.daniweb.com/

http://plc.openforall.net/html/theory.html