



ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΕΛΛΑΔΟΣ



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

*ΔΙΑΧΕΙΡΙΣΗ ΤΟΠΙΚΟΥ ΣΤΑΘΜΟΥ ΤΗΛΕΜΕΤΡΙΑΣ,
ΜΕ ΤΗ ΧΡΗΣΗ WEB SERVER ΕΝΣΩΜΑΤΩΜΕΝΟ ΣΕ PLC*

*ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΚΩΝ ΚΑΙ ΠΟΛΙΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ*

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΜΠΙΡΔΑ ΓΕΩΡΓΙΑ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΤΣΑΓΚΑΡΗΣ ΑΠΟΣΤΟΛΟΣ

Ευχαριστίες

Η συγκεκριμένη εργασία πραγματοποιήθηκε στο πλαίσιο του Μεταπτυχιακού Προγράμματος Σπουδών «Εφαρμοσμένα Συστήματα Αυτοματοποίησης» του Διεθνούς Πανεπιστημίου Ελλάδος.

Πρώτα από όλα τις θερμές μου ευχαριστίες στον κύριο Απόστολο Τσαγκάρη, ο οποίος με εμπιστεύτηκε και δέχτηκε να συνεργαστούμε στο πλαίσιο της διπλωματικής εργασίας. Με στήριξε σε όλη τη διάρκεια του μεταπτυχιακού και η κορύφωση ήρθε με την εκπόνηση της παρούσας εργασίας, παρέχοντας με επιστημονικό υλικό για την τεκμηρίωσή της.

Επίσης, θα ήθελα να ευχαριστήσω την εταιρεία στην οποία εργάζομαι «P.L.A Automation» που βοήθησε με ανθρώπινους πόρους και υλικό για τη δημιουργία της εργασίας σε πραγματικά δεδομένα.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου και κυρίως την αδερφή μου για την στήριξη και την εμπιστοσύνη που μου έδειξε συνολικά στη διάρκεια του μεταπτυχιακού.

Περίληψη

Η συγκεκριμένη εργασία ανήκει στα συστήματα τηλεματικής και αυτοματισμού που έχουν να κάνουν με την ύδρευση, άρδευση και αποχέτευση δήμων και κοινοτήτων. Συμπεριλαμβάνει ένα ολοκληρωμένο σύστημα αυτοματισμού, τη χρήση τηλεπικοινωνιακών συστημάτων αλλά και εφαρμογών απεικόνισης και διαχείρισης φιλικά στον απλό χρήστη.

Η παρούσα εργασία στηρίζεται σε πραγματικές εφαρμογές οι οποίες έχουν λάβει χώρα στην αποχέτευση του Δήμου Ιωαννίνων αλλά και στην αποχέτευση της Νήσου Σίφνου. Πρόκειται για σύστημα αποχέτευσης το οποίο αναλαμβάνει να μεταβιβάσει των όγκο των λυμάτων από τον αστικό ιστό στους κατατόπους βιολογικούς σταθμούς επεξεργασίας λυμάτων (ΕΕΛ). Η αξιοπιστία, η επαναληψιμότητα είναι βασικές προϋποθέσεις για τη λειτουργία τέτοιων συστημάτων διότι η αστοχία τους μπορεί να οδηγήσει σε επιβαρύνσεις του περιβάλλοντος ειδικά σε περιπτώσεις όπως τα νησιά.

Το σύστημα διαθέτει τοπικό ελεγκτή (PLC) υπεύθυνο για τη λειτουργία του αυτοματισμού αλλά και για την τηλεσύνδεση του τοπικού σταθμού με το κέντρο ελέγχου. Με τη χρήση αναλογικού αισθητήρα μέτρησης στάθμης και σε εφεδρεία ψηφιακών φλοτεροδιακοπών οδηγεί ελεγκτές στροφών (inverter) και κατ' επέκταση αντλίες ώστε να προωθήσει έως το επόμενο στάδιο τα λύματα τα οποία συσσωρεύονται στο σταθμό. Τα αντλητικά συστήματα είναι πάντα σε κατάσταση εφεδρείας και το PLC αναλαμβάνει την αδιάλειπτη λειτουργία σε περίπτωση αστοχίας του ενός εξ' αυτών.

Λειτουργικά, όταν η στάθμη ανέβει πάνω από ένα ρυθμιζόμενο όριο ενεργοποιείται ένα αντλητικό σύστημα με σκοπό να επαναφέρει τη στάθμη σε άλλο ρυθμιζόμενο όριο το οποίο και παύει τη λειτουργία. Όπως έχει αναφερθεί, εφεδρεία της αναλογικής μέτρησης στάθμης αποτελούν οι φλοτεροδιακόπτες, λειτουργία η οποία ενεργοποιείται αυτόματα σε περίπτωση αστοχίας του αναλογικού αισθητήρα ή κατ' επιλογήν. Οι αντλίες μία κάθε φορά, λειτουργούν κυκλικά για ομοιόμορφη φθορά και ενεργοποιείται αυτόματα η εφεδρεία σε περίπτωση αστοχίας.

Μετρούμενα σήματα, σήματα ελέγχου, σήματα συναγερμού και όρια λειτουργίας μεταδίδονται και λαμβάνονται τηλεματικά με το κέντρο ελέγχου. Επίσης, δίδεται και η δυνατότητα μέσω web interface ο χειριστής να προβεί στις παραπάνω ρυθμίσεις ή να ελέγξει τη λειτουργία του συστήματος.

Το κέντρο ελέγχου λαμβάνει και διαχειρίζεται τα σήματα του τοπικού σταθμού με τη χρήση SCADA ενημερώνοντας συνεχώς τους χειριστές αλλά και βάσεις δεδομένων για την περαιτέρω ανάλυση τους.

Abstract

The following thesis presents the development of a Supervisory Control and Data Acquisition (SCADA) system that controls and presents data about water distribution network and sewage collection and disposal network. It includes a complete automation system, use of telecommunications networks and user-friendly depiction applications.

The work presented is based on real life projects about the sewer systems of Ioannina city and Sifnos island. The sewer systems are responsible for transferring human wastes from the Urban environment to on-site Biological Wastewater Treatment Plants (WWTPs). One thing we have to be sure about is the system's reliability and correct operation because failure might result in environmental damage which is especially hard on places like islands.

The system consists of a local Programmable Logic Controller (PLC) which is responsible for the automation process and the connection of the local station with the control center. It uses analog level measurement sensors and reserve digital float switches in order to drive inverters and pumps that help to relocate the waste into the next stage of the sewage treatment process. The pumping systems are always in standby mode and the PLC takes in case of failure of one of them.

In operational state, when the level rises above a limit that we have defined, a pumping system is activated in order to restore the level to another fixed limit, which stops the operation. As has been mentioned, the analog level measurement consists of the float switches, a function which is activated automatically in case of failure of the analog sensor or in case we choose so. The pumps cycle one at a time for best optimization of the use of the materials and the backup is automatically activated in case of failure.

All the control and alarm signals and generally all the signals we measure are transmitted and received with the help of the control center. Also, the operator is given the possibility via a web interface to make the above settings or to check the operation of the system.

The control center receives and manages the signals of the local station while the SCADA system is constantly updating the databases for their further analysis and makes all the data accessible to the operators.

Περιεχόμενα

| | |
|---|-----------|
| Ευχαριστίες..... | 2 |
| Περίληψη..... | 3 |
| Abstract | 4 |
| Εισαγωγή..... | 7 |
| Κεφάλαιο 1. Θεωρητικό μέρος τηλεχειρισμού | 8 |
| 1.1 Η έννοια της τηλεμετρίας | 8 |
| 1.2 Ορισμός συστήματος SCADA | 8 |
| 1.3 Συστήματα SCADA..... | 9 |
| 1.4 Συστήματα HMI πάνελ (Διεπαφή ανθρώπου – μηχανής)..... | 10 |
| Κεφάλαιο 2. Θεωρητικό μέρος προγραμματιζόμενων λογικών ελεγκτών..... | 11 |
| 2.1 Η έννοια του προγραμματιζόμενου λογικού ελεγκτή (PLC) | 12 |
| 2.2 Δυνατότητες ενός προγραμματιζόμενου λογικού ελεγκτή (PLC) | 13 |
| 2.3 Δομή PLC..... | 13 |
| 2.4 Πλεονεκτήματα και μειονεκτήματα των PLC..... | 15 |
| 2.4.1 Πλεονεκτήματα PLC | 15 |
| 2.4.2 Μειονεκτήματα PLC | 15 |
| 2.5 Τύποι PLC..... | 16 |
| 2.6 Συνδεσμολογία PLC..... | 17 |
| 2.7 Προγραμματισμός PLC..... | 18 |
| 2.7.1 Γλώσσες προγραμματισμού των PLC | 18 |
| Κεφάλαιο 3. Θεωρητικό μέρος web server | 20 |
| 3.1 Γενικά για το web server..... | 20 |
| 3.2 Πλεονεκτήματα και μειονεκτήματα του web server..... | 21 |
| 3.3 Συμβατότητα PLC S7-1200/S7-1500 για την υποστήριξη του web server | 22 |
| SIMATIC S71200: SIMATIC S71500: | 22 |
| Κεφάλαιο 4. Πρακτικό μέρος | 23 |
| 4.1 Ενεργοποίηση του web server στο PLC..... | 23 |
| 4.2 Επεξήγηση βασικών περιεχομένων για το web | 26 |
| 4.2.1 Επεξήγηση φακέλου Images..... | 27 |
| 4.2.1.1 Επεξήγηση χρωμάτων Inverter | 28 |
| 4.2.1.2 Επεξήγηση χρωμάτων αναλογικού..... | 30 |
| 4.2.2 Επεξήγηση φακέλου Stylesheet | 30 |
| 4.3 Σύνδεση και προγραμματισμός του web server | 34 |
| 4.3.1 Σύνδεση στον web server | 34 |
| 4.3.2 Επεξήγηση του προγραμματισμού του web server | 39 |
| Κεφάλαιο 5. Αποτελέσματα | 71 |

| | |
|-----------------------------------|----|
| Βιβλιογραφία - Πηγές | 72 |
| Παράρτημα | 73 |
| Επεξήγηση Εντολών HTML..... | 73 |
| HTML Input Types | 73 |
| Λίστα εντολών html | 76 |

Εισαγωγή

Σκοπός για τη διαχείριση ενός τοπικού σταθμού τηλεμετρίας είναι η ολοκληρωμένη διαχείριση των υδάτινων πόρων, μέσω απεικόνισης όλου του δικτύου μεταφοράς σε υπολογιστή. Υπάρχουν αρκετά λογισμικά για την συγκεκριμένη απεικόνιση, από διάφορες εταιρίες, τα οποία, μπορούν να εγκατασταθούν σε υπολογιστές με συγκεκριμένες προδιαγραφές και να περαστούν οι αντίστοιχες άδειες ώστε να λειτουργήσει το συγκεκριμένο πρόγραμμα. Η πιο διαδεδομένη μέθοδος απεικόνισης ολόκληρου του υδραυλικού δικτύου ενός δήμου είναι το SCADA, το πρόγραμμα δηλαδή το οποίο εγκαθίσταται σε ένα υπολογιστή, και έχει όλες τις δυνατότητες και τις λειτουργίες που χρειάζεται για να έχουμε πλήρη χειρισμό όλων αντλιών, των γεωτρήσεων και των αναλογικών οργάνων.

Επίσης, η απεικόνιση ενός τοπικού σταθμού (όπως για παράδειγμα μιας γεώτρησης) μπορεί να γίνει και με μορφή HMI πάνελ. Το HMI έχει λιγότερες και πιο περιορισμένες δυνατότητες έναντι του SCADA. Το HMI είναι μία συσκευή η οποία τοποθετείται επάνω στην πόρτα του ηλεκτρολογικού πίνακα για τον τοπικό χειρισμό των συσκευών που επιθυμούμε να χειριστούμε ή να παρακολουθούμε όπως για παράδειγμα είναι η στάθμη. Το πάνελ είναι για απλούς χειρισμούς, χωρίς περίπλοκες λειτουργίες. Και στις δύο προηγούμενες μεθόδους απεικόνισης, χρησιμοποιείται ειδικό πρόγραμμα με ειδικό περιβάλλον σχεδίασης.

Τέλος, ένας άλλος τρόπος απεικόνισης ενός τοπικού σταθμού τηλεμετρίας, με τον οποίο θα ασχοληθούμε σε αυτή τη διπλωματική εργασία είναι το web server το οποίο είναι ενσωματωμένο στο PLC της siemens το S7-1200. Με τη χρήση του web server η απεικόνιση είναι ακόμα πιο απλή σε γραφικά σε σχέση με τις δύο προηγούμενες μεθόδους, διότι η γλώσσα προγραμματισμού που χρησιμοποιείται είναι η html και δεν υπάρχει πρόγραμμα σχεδίασης.

Για το λόγο αυτό, οι λειτουργίες περιγράφονται με κείμενα και δεν απεικονίζεται το υδραυλικό δίκτυο όπως σε ένα SCADA ή σε ένα HMI. Είναι ωστόσο ένας χρήσιμος τρόπος απεικόνισης διότι η απεικόνιση γίνεται με ένα PLC έναντι των άλλων περιπτώσεων στις οποίες χρειάζεται η προμήθεια υπολογιστή ή πάνελ. Ο τρόπος σύνδεσης είναι πολύ απλός. Σε έναν browser πληκτρολογούμε την IP του PLC και εφόσον είμαστε στο ίδιο δίκτυο, μας ανοίγει το περιβάλλον πλοήγησης για την απεικόνιση του τοπικού σταθμού.

Κεφάλαιο 1. Θεωρητικό μέρος τηλεχειρισμού

1.1 Η έννοια της τηλεμετρίας

Με τον όρο τηλεμετρία εννοούμε τη συλλογή δεδομένων από απόσταση μέσω προγραμματιζόμενων λογικών ελεγκτών (PLC). Τα δεδομένα αυτά, μπορεί να αναφέρονται, σε λειτουργίες και σφάλματα συσκευών (γεωτρήσεων, αντλιών κλπ.), σε αναλογικά μεγέθη όπως για παράδειγμα η μέτρηση της στάθμης η οποία είναι από τα πιο σημαντικά αναλογικά όργανα. Με τον έλεγχο της στάθμης προλαβαίνουμε τυχόν υπερχειλίσεις, καθώς επίσης, και την στέρση του νερού σε κάποιο χωριό το οποίο τροφοδοτούμε. Η μέτρηση της πίεσης με πιεσόμετρα προκειμένου να προλάβουμε το σπάσιμο του αγωγού. Η μέτρηση της παροχής με τα παροχόμετρα για τα στιγμιαία κυβικά καθώς επίσης και τα συνολικά κυβικά που περνάνε από το αντίστοιχο όργανο. Το χλώριο που εισέρχεται στην δεξαμενή και άλλα. Η συλλογή αυτών των δεδομένων μπορεί να γίνεται μέσω ενός υπολογιστή, στον οποίο, έχει εγκατασταθεί το απαραίτητο λογισμικό για την απεικόνιση του υδραυλικού δικτύου, σε ένα HMI πάνελ ή μέσω του web server. Στην περίπτωση της πλήρης απεικόνισης όλου του δικτύου δημιουργείται ο κεντρικός σταθμός ελέγχου (ΚΣΕ) της τηλεμετρίας όπου στον Δήμο της περιοχής τοποθετείται ο υπολογιστής αυτός ώστε να μπορεί να συλλέγει σε ένα σημείο όλα τα απαραίτητα δεδομένα που χρειάζονται.

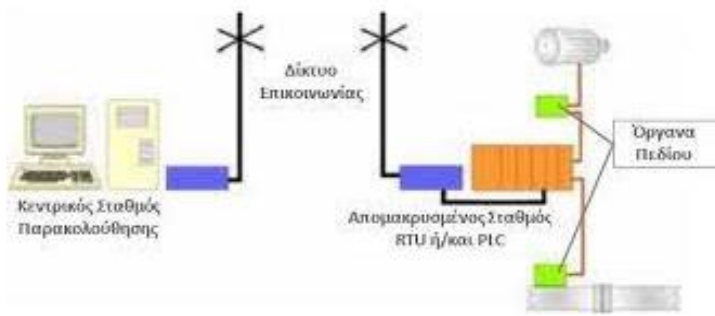
1.2 Ορισμός συστήματος SCADA

Ο όρος SCADA (Σύστημα τηλεελέγχου και τηλεχειρισμού) ο οποίος σημαίνει Supervisory Control And Data Acquisition περιγράφει μια κατηγορία συστημάτων βιομηχανικού αυτοματισμού ελέγχου, τηλεμετρίας και τηλεχειρισμού. Ένα σύστημα μπορεί να αποτελείται από πολλά PLC, όπως για παράδειγμα να υπάρχει ένα PLC σε κάθε διεργασία για τον έλεγχο των επιμέρους στοιχείων και μονάδων μια εγκατάστασης. Στο κέντρο ελέγχου και διαχείρισης υπάρχει ένα κύριο PLC, το PLC Master, το οποίο συλλέγει όλα τα δεδομένα από τα τοπικά PLC είτε μέσω καλωδίου είτε μέσω ασύρματου πομποδέκτη.

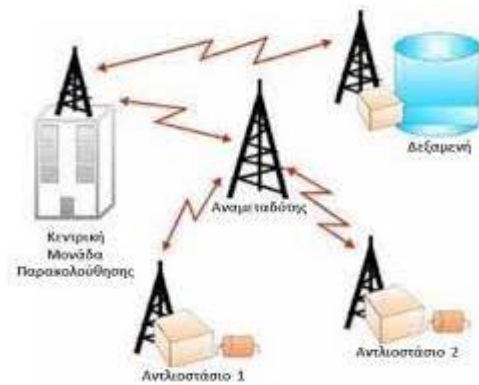
Έπειτα, το κεντρικό PLC μπορεί να μεταδίδει τα δεδομένα της εγκατάστασης σε μακρινά σημεία μέσω κάποιου συστήματος τηλεπικοινωνίας, όπως για παράδειγμα μέσω του ενσύρματου τηλεφωνικού δικτύου ή μέσω ασύρματου δικτύου.

Οι δυνατότητες που μας προσφέρει το SCADA είναι οι παρακάτω:

- Απεικόνιση αναλογικών μεγεθών
- Συλλογή πληροφοριών όλων των σημείων της εγκατάστασης
- Απεικόνιση του υδραυλικού δικτύου
- Λειτουργία ή σφάλμα συσκευών
- Απεικόνιση διαγραμμάτων αναλογικών οργάνων, καθώς επίσης και με τη μορφή πίνακα για τα συνολικά κυβικά, KWh κλπ.
- Καταγραφή σφαλμάτων
- Εκτυπώσεις αναφορών



Εικόνα 2 Ασύρματη Επικοινωνία PLC με SCADA



Εικόνα 1 Ασύρματη επικοινωνία PLC αντλιοστασίων με SCADA

Κάποια παραδείγματα συστημάτων τηλεχειρισμού αναγράφονται παρακάτω:

- Δίκτυα νερού και αποχέτευσης
- Τηλεπικοινωνιακοί σταθμοί
- Δίκτυα παραγωγής και διανομής ηλεκτρικής ενέργειας καθώς και αερίου και υγρών καυσίμων.
- Αρδευτικά συστήματα
- Βιολογικοί καθαρισμοί
- Συστήματα ελέγχου και ρύθμισης γραμμών παραγωγής σε εργοστάσια

Ο έλεγχος των παραπάνω δικτύων μπορεί να γίνει από ένα ή περισσότερα κεντρικά σημεία ελέγχου. Μπορεί επίσης να γίνει από το κεντρικό σύστημα SCADA στο control room ή από τοπικό HMI πάνελ. Με την συνολική απεικόνιση όλου του δικτύου στο SCADA έχουμε τον πλήρη έλεγχο του δικτύου και μπορούμε να προλάβουμε έκτακτες καταστάσεις συναγερμών ή να γνωρίζουμε άμεσα σε ποιο σημείο της εγκατάστασης υπάρχει το πρόβλημα.

1.3 Συστήματα SCADA

Με την εφαρμογή του Εποπτικού Ελέγχου και Συλλογής Δεδομένων (SCADA) αποκτάτε ο πλήρης έλεγχος της παρακολούθησης και της διαχείρισης σε αντλιοστάσια, δίκτυα ύδρευσης κλπ.

Όπως αναφέρθηκε παραπάνω υπάρχουν αρκετές λειτουργίες τις οποίες μπορούμε να ελέγξουμε και να ορίσουμε στο σύστημα του SCADA. Οι κυριότερες αναφέρονται παρακάτω:

- Έναρξη και παύση αντλιών
- Απεικόνιση σφαλμάτων όλων των συσκευών (όπως για παράδειγμα βλάβη αντλιών, χαμηλή-υψηλή στάθμη κλπ.)
- Έλεγχος αναλογικών οργάνων όπως για παράδειγμα στάθμη καθώς επίσης και ποιοτικά όργανα όπως το pH., το χλώριο κλπ.
- Επιτήρηση των αγωγών δικτύου για διαρροές (μέτρηση πίεσης και παροχής)
- Χειρισμό βανών (άνοιγμα-κλείσιμο)
- Καταγραφή των αναλογικών οργάνων, της λειτουργίας των αντλιών μέσω διαγραμμάτων αλλά και πινάκων
- Μέτρηση του αναλυτή ενέργειας για την κατανάλωση της εκάστοτε εφαρμογής

Με αυτό τον τρόπο επιτυγχάνονται τα παρακάτω:

- Εξοικονόμηση νερού και ενέργειας
- Ορθολογική διαχείριση του δικτύου προς αποφυγή προβλημάτων
- Δυνατότητα επέκτασης με την προσθήκη νέων σταθμών στο σύστημα

Επίσης, υπάρχει η δυνατότητα δημιουργίας διαφόρων επιπέδων χρηστών ανάλογα με την ανάγκη της εφαρμογής ώστε να υπάρχει και ο υπεύθυνος για τις ρυθμίσεις του αυτοματισμού.

Για παράδειγμα, ο ένας χρήστης να μπορεί μόνο να βλέπει τις λειτουργίες της εφαρμογής. Εάν δηλαδή έχει ανέβει η πίεση σε έναν αγωγό να προλάβει τυχόν σπάσιμο του αγωγού. Ένας δεύτερος χρήστης να μπορεί να ξεκινάει τις αντλίες, και ένας τρίτος χρήστης να μπορεί να ρυθμίζει τα όρια του αυτοματισμού.

1.4 Συστήματα HMI πάνελ (Διεπαφή ανθρώπου – μηχανής)

Με την εφαρμογή του HMI πάνελ μπορεί να γίνει ο τοπικός χειρισμός σε έναν σταθμό. Τα πάνελ μπορούν να τοποθετηθούν είτε σε σταθμούς με γεωτρήσεις είτε σε σταθμούς με δεξαμενές ανάλογα με την ζήτηση και την απαραίτητη λειτουργία του κάθε συστήματος. Οι λειτουργίες τις οποίες μπορεί να διαχειριστεί το πάνελ είναι αρκετές.

Οι δυνατότητες που μας προσφέρει το HMI πάνελ είναι οι παρακάτω:

- Έναρξη και παύση αντλιών
- Απεικόνιση σφαλμάτων όλων των συσκευών (όπως για παράδειγμα βλάβη αντλιών, χαμηλή-υψηλή στάθμη κλπ.)
- Έλεγχος αναλογικών οργάνων όπως για παράδειγμα στάθμη καθώς επίσης και ποιοτικά όργανα όπως το pH, το χλώριο κλπ.
- Επιτήρηση των αγωγών δικτύου για διαρροές (μέτρηση πίεσης και παροχής)
- Χειρισμό βανών (άνοιγμα-κλείσιμο)

Σαφώς οι λειτουργίες του πάνελ είναι πιο περιορισμένες στο κομμάτι των διαγραμμάτων και των αναφορών. Επίσης, ανάλογα με την ανάγκη του έργου προσαρμόζεται και η εφαρμογή. Για παράδειγμα, η εφαρμογή σε ένα πάνελ μπορεί να είναι μόνο απεικονιστική. Δηλαδή, να μην μπορεί ο χειριστής να ξεκινάει αντλίες, να ανοίγει βάνες ή να ορίζει τα όρια του αυτοματισμού. Μπορεί μόνο να βλέπει την στάθμη και την λειτουργία ή των σφάλμα της γεώτρησης.

Επίσης, υπάρχει και στο πάνελ η δυνατότητα των χρηστών όπως αναφέρθηκε παραπάνω. Υπάρχει η δυνατότητα να δημιουργηθούν επίπεδα χρηστών. Το πρώτο επίπεδο του χρήστη να έχει δικαίωμα μόνο απεικόνισης. Το δεύτερο να έχει δικαίωμα να ξεκινάει την αντλία και το τρίτο επίπεδο να ορίζει και τα όρια του αυτοματισμού. Σε κάθε περίπτωση, εξαρτάται από την ζήτηση του πελάτη.

Κεφάλαιο 2. Θεωρητικό μέρος προγραμματιζόμενων λογικών ελεγκτών

Στις μέρες μας, η χρήση των προγραμματιζόμενων λογικών ελεγκτών (PLC) είναι πολύ διαδεδομένη. Τα PLC προορίζονται για να αντικαταστήσουν τον κλασικό ηλεκτρολογικό πίνακα με τα ρελέ. Χρησιμοποιούνται σε εμπορικές και βιομηχανικές εφαρμογές και κυρίως όπου απαιτείται υψηλός ρυθμός παραγωγής προϊόντων. Μια εικόνα ενός προγραμματισμένου λογικού ελεγκτή φαίνεται παρακάτω.



Εικόνα 3 Προγραμματιζόμενος λογικός ελεγκτής (PLC)



Εικόνα 4 Προγραμματιζόμενος λογικός ελεγκτής (PLC)

2.1 Η έννοια του προγραμματιζόμενου λογικού ελεγκτή (PLC)

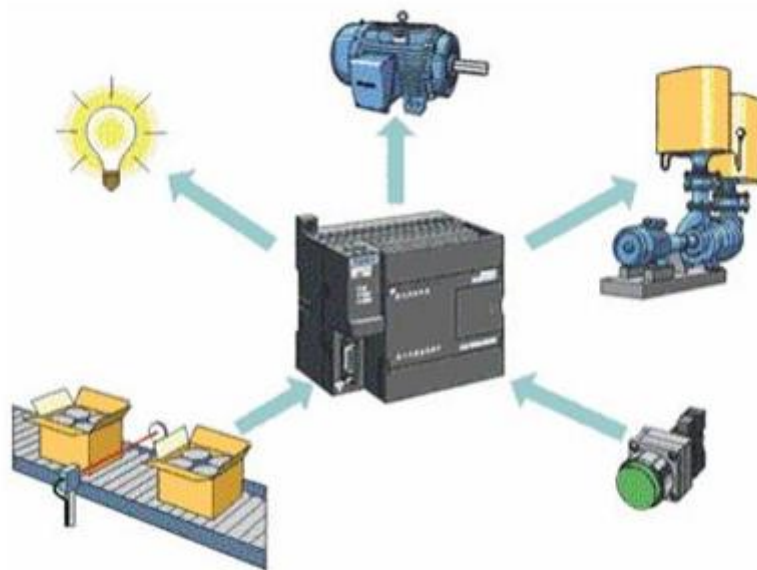
Το PLC είναι ένας ελεγκτής βασιζόμενος σε ένα μικροεπεξεργαστή του οποίου, τις λειτουργίες καθορίζει ένα πρόγραμμα αποθηκευμένο σε μόνιμη μνήμη (ROM/EEPROM) αλλά και αρκετή μνήμη RAM για τη γρήγορη εκτέλεση των εντολών. Τα κομμάτια από τα οποία αποτελούνται τα PLC είναι αφενός, η CPU (Central Processing Unit - Κεντρική Μονάδα Επεξεργασίας) και αφετέρου, οι μονάδες εισόδου - εξόδου. Στη CPU βρίσκεται το πρόγραμμα που είναι γραμμένο σε μια από τις γλώσσες προγραμματισμού.

Το συγκεκριμένο πρόγραμμα, μπορεί και επεξεργάζεται συναρτήσεις λογικής τύπου Boole, χρονισμό, απαρίθμηση αλλά και αριθμητικές πράξεις με απώτερο σκοπό τον έλεγχο ενός σύνθετου μηχανήματος ή μιας διεργασίας μέσω των εξόδων του. Περιλαμβάνει εισόδους και εξόδους που συνδέονται με τα στοιχεία μιας εγκατάστασης και έναν αλγόριθμο ο οποίος, είναι σε θέση να καθορίσει ότι κάποιος συνδυασμός εισόδων παράγει ένα συγκεκριμένο αποτέλεσμα στις εξόδους.

Ακόμη, σύνθετοι μικροεπεξεργαστές σε μορφή καρτών μπορούν να τοποθετηθούν στο ίδιο Bus με τις κεντρικές μονάδες και να ανταλλάσσουν δεδομένα μεταξύ τους σε μεγάλες ταχύτητες όπως για παράδειγμα, κάρτες/μικροεπεξεργαστές επικοινωνίας για όλα τα γνωστά πρωτόκολλα.

Τα PLC όπως όλες οι ηλεκτρονικές συσκευές, έχουν εξελιχθεί με το πέρασμα του χρόνου και έχουν βελτιωθεί τόσο σε όγκο όσο και σε ταχύτητα. Επίσης, οι συσκευές ελέγχου και προγραμματισμού έχουν αναλογικά βελτιωθεί και περάσαμε πλέον από τους Programmer σε Windows Software τα οποία, μπορούν να εγκατασταθούν σε όλους τους γνωστούς υπολογιστές με απλό τρόπο και να επικοινωνούν με τα PLC απλά με την κάρτα επικοινωνίας Ethernet.

Τα σύγχρονα προγράμματα είναι πιο ολοκληρωμένα από ποτέ ενσωματώνοντας κάτω από το ίδιο δέντρο (Project), το πρόγραμμα διαχείρισης. Τέτοιου τύπου εφαρμογές είναι παραδείγματος χάρη, η αυτόματη τοποθέτηση προϊόντων σε συσκευασίες και η αυτόματη εμφιάλωση μπουκαλιών (π.χ. εμφιάλωση κρασιού)



Εικόνα 5 Συσκευές συνδεδεμένες στο PLC

2.2 Δυνατότητες ενός προγραμματιζόμενου λογικού ελεγκτή (PLC)

- Μπορεί να εκτελέσει εργασίες μεταγωγής αναμετάδοσης.
- Μπορεί να διεξάγει καταμέτρηση, υπολογισμό και σύγκριση τιμών αναλογικών διεργασιών.
- Προσφέρει ευελιξία για να τροποποιήσει τη λογική ελέγχου, όποτε απαιτείται, στο συντομότερο χρονικό διάστημα.
- Ανταποκρίνεται στις αλλαγές των παραμέτρων της διαδικασίας μέσα σε κλάσματα δευτερολέπτων.
- Βελτιώνει την αξιοπιστία του συνολικού συστήματος ελέγχου.
- Είναι οικονομικά αποδοτικό για τον έλεγχο σύνθετων συστημάτων.
- Σκοπεύει να τραβήξει πιο απλά και πιο γρήγορα
- Μπορεί να εργαστεί με τη βοήθεια του υπολογιστή HMI (Διεπαφή Ανθρώπου-Μηχανής)

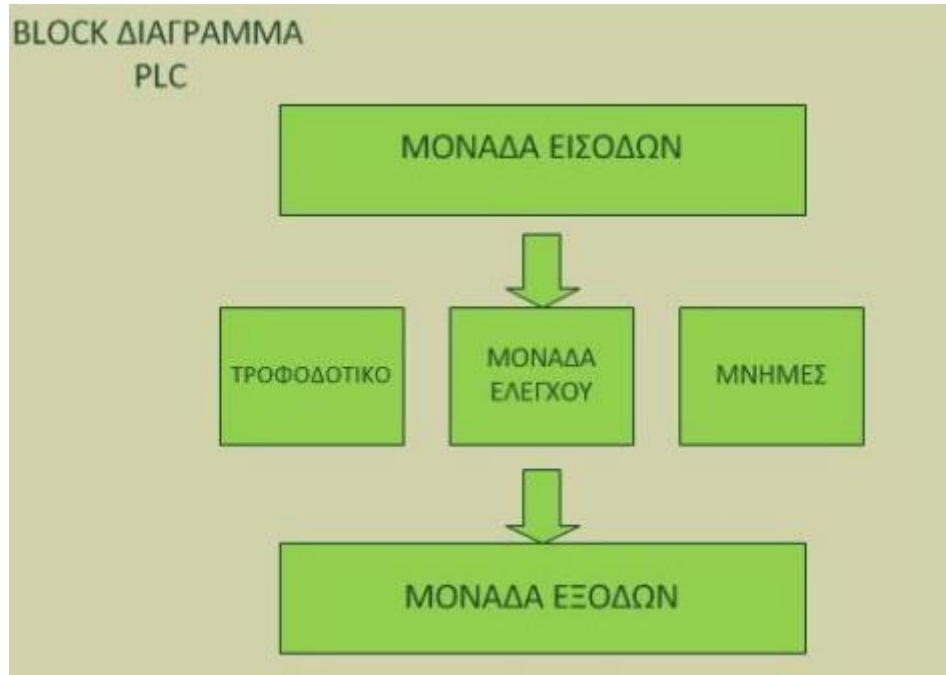
2.3 Δομή PLC

Παρακάτω περιγράφεται η δομή του PLC.

- Αρχικά υπάρχει μία ράγα πάνω στην οποία συνδέονται οι μονάδες.
- Υπάρχει η μονάδα τροφοδοσίας (PS).
- Έπειτα τοποθετείται η μονάδα επεξεργασίας (CPU) η οποία αποτελεί τον εγκέφαλο του PLC.
- Δεξιά από την CPU μπαίνουν οι κάρτες όπως είναι οι ψηφιακές κάρτες εισόδων/εξόδων και οι αναλογικές κάρτες εισόδων/εξόδων.



Εικόνα 6 Παράδειγμα δομής ενός PLC Siemens



Εικόνα 7 Block διάγραμμα ενός PLC

Το PLC αποτελείται από τη μνήμη RAM, ROM και EEPROM οι οποίες περιγράφονται παρακάτω:

- Μνήμη RAM: Είναι η περιοχή μνήμης όπου αποθηκεύονται οι καταστάσεις των εισόδων και των εξόδων. Η περιοχή αυτή ονομάζεται για τις εισόδους εικόνα εισόδου και για τις εξόδους εικόνα εξόδου. Περιοχή μνήμης όπου αποθηκεύονται οι ενδιάμεσες πληροφορίες που αφορούν τη λειτουργία του αυτοματισμού . Περιοχή μνήμης των χρονικών. Περιοχή μνήμης των απαριθμητών. Περιοχή μνήμης όπου αποθηκεύονται τα προγράμματα του χρήστη, δηλαδή τα προγράμματα που λειτουργούν ένα συγκεκριμένο αυτοματισμό.
- Μνήμη ROM: Στη μνήμη ROM (Read Only Memory) ο κατασκευαστής του προγραμματιζόμενου ελεγκτή αποθηκεύει το λειτουργικό σύστημα του PLC, δηλαδή τις οδηγίες (το πρόγραμμα) για όλες τις βασικές λειτουργίες που είναι απαραίτητες για να δουλέψει το PLC.
- Μνήμη EEPROM: Επειδή η μνήμη RAM με την απώλεια της τροφοδοσίας χάνει τα δεδομένα της (εκτός αν χρησιμοποιείται μπαταρία), τα PLC χρησιμοποιούν έναν άλλο τύπο μνήμης, την EEPROM (Electrically Erasable Programmable Read Only Memory), η οποία προγραμματίζεται και σβήνει ηλεκτρικά • Πρόκειται για μνήμη που με την πτώση της τροφοδοσίας διατηρεί τα δεδομένα της, και η οποία γράφεται και να σβήνεται μέσω ειδικού μηχανήματος

2.4 Πλεονεκτήματα και μειονεκτήματα των PLC

2.4.1 Πλεονεκτήματα PLC

Τα PLC είναι ευρέως διαδεδομένα και χρησιμοποιούνται σε πολλές εφαρμογές στην βιομηχανία. Ωστόσο υπάρχουν ορισμένα πλεονεκτήματα και μειονεκτήματα όσο αφορά την χρήση τους. Παρακάτω αναφέρονται τα πλεονεκτήματά τους:

- Χαμηλό κόστος υλοποίησης του PLC.
- Απαιτείται σχετικά μικρός χρόνος για την υλοποίηση του αυτοματισμού και υπάρχει ευελιξία κατά την διαδικασία της θέσης σε λειτουργία.
- Ελαχιστοποίηση του κόστους συντήρησης του πίνακα αυτοματισμού.
- Μεγάλη ευελιξία σε τροποποιήσεις του αυτοματισμού.
- Μεγάλες δυνατότητες επέκτασης του αυτοματισμού.
- Ευκολία δημιουργίας πολύπλοκων καθώς και έξυπνων διεργασιών.
- Δυνατότητα σύνδεσης με κεντρικό υπολογιστικό σύστημα ή το εταιρικό δίκτυο
- Καταλαμβάνει ελάχιστο χώρο.
- Εύκολος προγραμματισμός, έλεγχος λειτουργίας και ευκολία κατά την αποσφαλμάτωση του προγράμματος.
- Οικονομία στην κατανάλωση ενέργειας.
- Εύκολη, πιο γρήγορη και σχεδόν άνευ κόστους αναπαραγωγή όμοιων εφαρμογών.
- Επαναχρησιμοποίηση του PLC σε περίπτωση που καταργηθεί η υφιστάμενη εφαρμογή μειώνοντας έτσι το κόστος ανάπτυξης των εφαρμογών με χρήση PLC.

2.4.2 Μειονεκτήματα PLC

Πέρα από τα πλεονεκτήματα τα οποία έχουν οι προγραμματιζόμενοι λογικοί ελεγκτές, έχουν και κάποια μειονεκτήματα τα οποία αναφέρονται παρακάτω:

- Σε μία απλή εφαρμογή τα PLC δεν είναι η καλύτερη λύση λόγω του κόστους τους έναντι του πεδίου με ηλεκτρονόμους.
- Σε περίπτωση μίας βλάβης στο PLC μπορεί να χρειαστεί να αντικατασταθεί ένα τμήμα του ή ακόμη, και όλο το PLC, ενώ στην περίπτωση των ρελέ απλά θα αντικαθιστούσαμε το ρελέ.
- Απαιτείται εξειδικευμένο προσωπικό για την εγκατάσταση και την παρακολούθηση της λειτουργίας του κάτι που σημαίνει αυξημένο κόστος.
- Τα PLC είναι γενικά ευαίσθητα στο θόρυβο και απαιτείται συνεπώς ειδικές κατασκευές για την προστασία τους.

2.5 Τύποι PLC

Υπάρχουν διάφοροι τύποι PLC, κάποιοι τύποι εξ αυτών είναι τα Compact (Εικόνα 3) και τα Modular (Εικόνα 4). Με την έννοια Compact, αναφερόμαστε στην συσκευή στην οποία, η τροφοδοσία, η CPU, οι εισοδοί και οι εξοδοί ανήκουν στην ίδια μονάδα σε αντίθεση με τα modular στα οποία, η τροφοδοσία, η CPU, οι εισοδοί και οι εξοδοί αποτελούν ξεχωριστές συσκευές. Επέλεξα να αναφερθώ σε αυτούς τους δύο τύπους γιατί είναι οι πιο συνηθείς και γιατί αποτελούν το μεγαλύτερο ποσοστό που ήδη είναι εγκατεστημένα. Για παράδειγμα, υπάρχουν και τύποι PLC με ειδικές λειτουργίες (όπως, Redundancy) τα οποία, τα συναντάμε σε πολύ ειδικές εγκαταστάσεις όπως είναι οι πυρηνικοί σταθμοί και τα διυλιστήρια.

Οι βασικές διαφορές μεταξύ των δύο αυτών τύπων περιγράφονται παρακάτω:

Οι δυνατότητες όσον αναφορά τα Compact PLC είναι περιορισμένες υπό την έννοια ότι έχουν μικρό αριθμό εισόδων και εξόδων, μικρή δυνατότητα επέκτασης καθώς και μικρή επεξεργαστική ισχύ. Ένα βασικό όμως πλεονέκτημά τους αποτελεί το γεγονός ότι είναι πιο οικονομικά σε σχέση με τα Modular. Σε αντίθεση με τα Compact, τα Modular χρησιμοποιούνται για πιο σύνθετους αυτοματισμούς στους οποίους, απαιτείται μεγάλος αριθμός εισόδων και εξόδων.

Ένα πλεονέκτημα επίσης, είναι το γεγονός πως σε μελλοντική επέκταση της εφαρμογής μας, αρκεί απλά να προσθέσουμε μία ή και περισσότερες βαθμίδες εισόδων ή εξόδων διατηρώντας την ίδια CPU και το ίδιο τροφοδοτικό. Το μειονέκτημά τους είναι, το κόστος τους και κατά συνέπεια δεν προτιμάται σε εφαρμογές με χαμηλές απαιτήσεις.



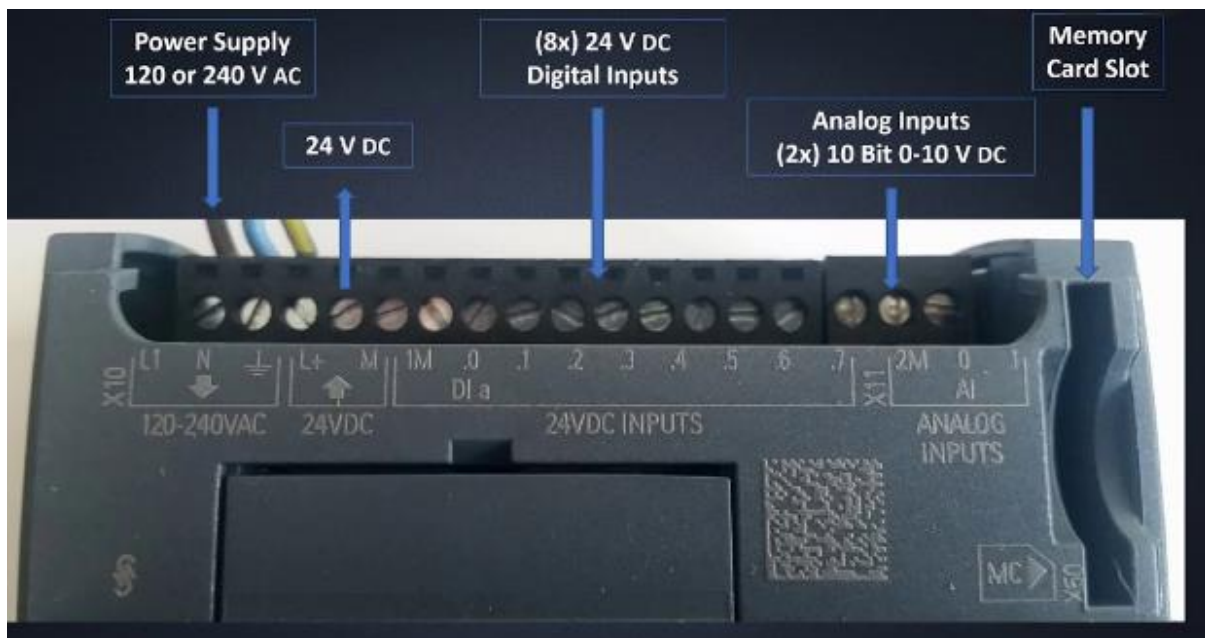
Εικόνα 8 Compact CPU



Εικόνα 9 Modulat CPU

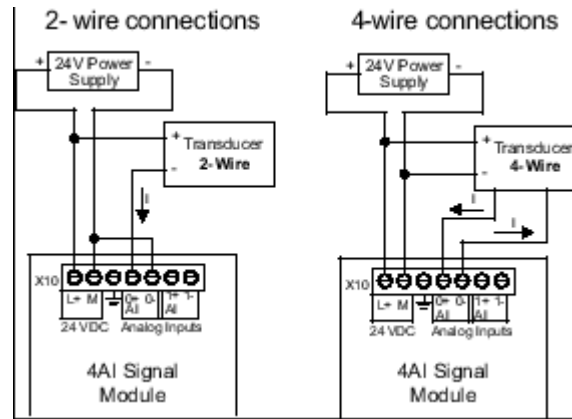
2.6 Συνδεσμολογία PLC

Για τον τρόπο σύνδεσης ενός PLC υπάρχουν τα αντίστοιχα εγχειρίδια ωστόσο η τροφοδοσία του δεν είναι δύσκολη. Αρχικά, για την απλή τροφοδοσίας της CPU χρειάζεται να δώσουμε 24V στο γράμμα L+ και 0V στο γράμμα M όπως φαίνεται στην παραπάνω εικόνα. Ανάλογα με τις κάρτες που θέλουμε να συνδέσουμε τις τροφοδοτούμε αντίστοιχα. Για παράδειγμα, μία κάρτα αναλογικών σημάτων χρειάζεται να τροφοδοτηθεί και αυτή με 24V στο L2+ και να συνδέσουμε τα 0V στο 2M. Σε περίπτωση που δεν τροφοδοτηθεί σωστά ή καθόλου η κάρτα στο hardware Configuration θα δούμε ότι στην κάρτα επάνω θα ανάβει κόκκινο λαμπάκι και από το diagnostic θα μπορέσουμε να καταλάβουμε τι πρόβλημα ακριβώς έχει η κάρτα.



Εικόνα 10 Τροφοδοσία PLC

Ανάλογα με την περίπτωση του αισθητηρίου, εάν είναι για παράδειγμα 2-wire ή 4-wire τροφοδοτείται και αντίστοιχα η κάρτα όπως φαίνεται στην εικόνα.



Εικόνα 11 Σύνδεση αναλογικής κάρτα

2.7 Προγραμματισμός PLC

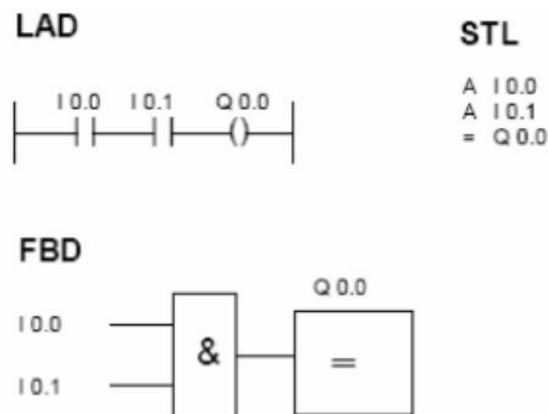
2.7.1 Γλώσσες προγραμματισμού των PLC

Ο προγραμματισμός ενός PLC σημαίνει τη σειρά εντολών, οι οποίες λύνουν έναν αλγόριθμο ο οποίος αντιστοιχεί σε ένα σύστημα αυτοματισμού. Η επιλογή της γλώσσας προγραμματισμού εξαρτάται από την εμπειρία και την γνώση του χρήστη σε ψηφιακά ηλεκτρονικά, σε υπολογιστές, σε συστήματα αυτοματισμού που λειτουργούν με κλασικό τρόπο και φυσικά εξαρτάται από την φύση του προβλήματος που έχουμε να αντιμετωπίσουμε.

Οι γλώσσες προγραμματισμού μπορούν να ταξινομηθούν σε γραφικές και μη γραφικές ανάλογα με το είδος των στοιχείων που χρησιμοποιούν. Οι πρώτες χρησιμοποιούν γραφικά στοιχεία που μοιάζουν αρκετά στα σύμβολα που χρησιμοποιούνται στον κλασικό αυτοματισμό και επίσης σύμβολα λογικών πυλών (AND, OR, NOT κλπ.). Είναι πιο προσιτές σε ανθρώπους που έχουν εμπειρία στον κλασικό αυτοματισμό και έχουν το πλεονέκτημα της καλύτερης εποπτείας. Οι δεύτερες χρησιμοποιούν εντολές που η κάθε μία αντιστοιχεί σε μία εντολή της γλώσσας μηχανής.

Οι πιο σπουδαίες γλώσσες προγραμματισμού είναι οι παρακάτω:

- Ladder ή διάγραμμα επαφών: Είναι η πρώτη γλώσσα προγραμματισμού του PLC η οποία είναι μια γλώσσα γραφικών, η οποία χρησιμοποιεί ηλεκτρομηχανικά σύμβολα και επιτρέπει ουσιαστικά τη μεταφορά του ηλεκτρολογικού σχεδίου στο PLC. Με αυτή τη γλώσσα η εργασία σχεδιασμού του αυτοματισμού γίνεται εύκολα και γρήγορα
- Statement list (STL) ή λίστα εντολών: Η δεύτερη γλώσσα προγραμματισμού είναι η Statement List (STL) που αναπτύχθηκε σχεδόν ταυτόχρονα με τη LADDER. Η σύνταξη των εντολών είναι παραπλήσια με αυτή του κώδικα μηχανής (Machine Code), όπου οι εντολές και οι λειτουργίες ακολουθούνται από διευθύνσεις.
- FBD ή γλώσσα λογικών γραφικών: Η τρίτη γλώσσα Function Block Diagram είναι κι αυτή γλώσσα προγραμματισμού με γραφικά. Οι εντολές εδώ αναπαρίστανται με λογικά blocks, παρόμοια με αυτά που συναντώνται στην άλγεβρα Boole. Κι εδώ όπως και στα διαγράμματα στην ψηφιακή τεχνολογία μπορούμε να παρακολουθήσουμε τη ροή του σήματος ανάμεσα στα «κουτιά». Για παράδειγμα η λογική πράξη AND αντιστοιχεί σε μία εν σειρά σύνδεση επαφών του κυκλωματικού διαγράμματος. Αν στο παρακάτω παράδειγμα έστω και μία από τις εισόδους έχει τιμή 0, τότε η Q0.0 θα έχει τιμή 0. Για να έχει η Q0.0 κατάσταση 1 θα πρέπει όλες οι εισοδοί να έχουν κατάσταση 1. Παρακάτω απεικονίζονται και οι προαναφερθείσες τρεις τυπικές γλώσσες προγραμματισμού για την υλοποίηση της πράξης AND.



Εικόνα 12 Παράδειγμα προγραμματισμού σε διαφορετικές γλώσσες

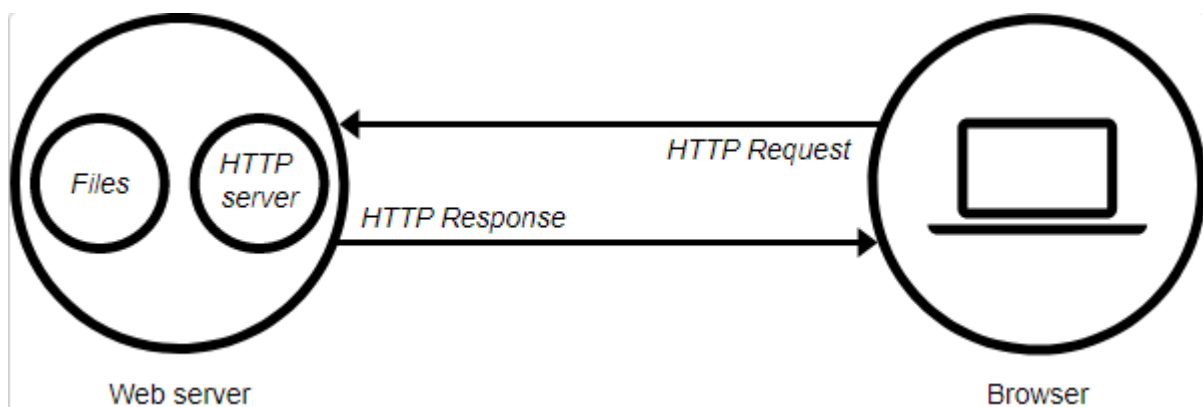
Κεφάλαιο 3. Θεωρητικό μέρος web server

3.1 Γενικά για το web server

Ο όρος Web Server αναφέρεται σε λογισμικό και υλικό που λειτουργούν μαζί για να αποθηκεύουν, να επεξεργάζονται και να παραδίδουν αρχεία ενός website σε web browsers. Ο web server, χρησιμοποιεί το πρωτόκολλο Hypertext Transfer Protocol (HTTP) για να ανταποκρίνεται στα αιτήματα των χρηστών που υποβάλλονται μέσω του Παγκόσμιου Ιστού (World Wide Web). Μέσω αυτής της διαδικασίας, οι web servers φορτώνουν και παραδίδουν τη σελίδα που ζητήθηκε στο πρόγραμμα περιήγησης του χρήστη - για παράδειγμα, το Google Chrome. Οι web servers χρησιμοποιούν επίσης το Simple Mail Transfer Protocol (SMTP) και το File Transfer Protocol (FTP) για την επεξεργασία αρχείων για email ή αποθήκευση.

Από την πλευρά του υλικού, ένας web server συνδέεται στο Διαδίκτυο, το οποίο του επιτρέπει να ανταλλάσσει δεδομένα ή αρχεία μεταξύ άλλων συσκευών που είναι επίσης συνδεδεμένες. Αυτά τα δεδομένα μπορούν να υπάρχουν σε διάφορες μορφές, όπως αρχεία HTML, εικόνες, αρχεία JavaScript ή φύλλα στυλ CSS.

Το λογισμικό του web server ελέγχει τον τρόπο με τον οποίο οι χρήστες του web έχουν πρόσβαση στα αρχεία του. Αποτελείται από πολλά στοιχεία, τα οποία στεγάζουν τουλάχιστον έναν server HTTP. Ένας server HTTP είναι λογισμικό που μπορεί να κατανοήσει αιτήματα HTTP και διευθύνσεις URL. Η πρόσβαση στον HTTP server γίνεται μέσω των ονομάτων των ιστοτόπων (domain names). Εάν ο χρήστης δώσει το όνομα του ιστοτόπου που τον ενδιαφέρει τότε ο HTTP server θα του εμφανίσει το περιεχόμενο αυτού. Στο πιο βασικό επίπεδο, κάθε φορά που ένα πρόγραμμα περιήγησης χρειάζεται ένα αρχείο που φιλοξενείται σε έναν web server, το πρόγραμμα περιήγησης ζητά το αρχείο μέσω HTTP. Όταν το αίτημα φτάσει στον σωστό web server, ο HTTP server αποδέχεται το αίτημα, βρίσκει το ζητούμενο έγγραφο και το στέλνει πίσω στο πρόγραμμα περιήγησης, επίσης μέσω HTTP ενώ αν ο server δεν βρει το έγγραφο που ζητήθηκε, επιστρέφει μια απάντηση 404.



Οι ιστοσελίδες προσφέρουν έναν διαισθητικό τρόπο χρησιμοποιώντας ένα τυπικό πρόγραμμα περιήγησης στο Web ή σε μια φορητή συσκευή επικοινωνίας, για πρόσβαση σε όλες τις παραλλαγές πληροφοριών μέσα στο S7 PLC. Μπορούμε να αποκτήσουμε ευέλικτα πρόσβαση σε διαγνωστικά δεδομένα και μεταβλητές κατά τη δοκιμαστική φάση, τα οποία έχουν ήδη προβληθεί. Κατά τη διάρκεια των κανονικών εργασιών, έχουμε τη δυνατότητα να τροποποιήσουμε και να παρακολουθήσουμε τη διαδικασία του αυτοματισμού. Σε κάθε αλλαγή που κάνουμε στις σελίδες html πρέπει να πηγαίνουμε στο configuration του PLC, να κάνουμε «Generate » και έπειτα να κάνουμε hardware download του προγράμματος. Για τον προγραμματισμό του web server, μία γλώσσα που χρησιμοποιείται είναι η γλώσσα προγραμματισμού html για την οποία δεν απαιτείται προηγούμενη γνώση.

Υπάρχουν τρεις κατηγορίες για τον προγραμματισμό του web server οι οποίες αναφέρονται παρακάτω:

- Web page standard: HTML without JavaScript
- Web page optimized: HTML with JavaScript
- Web API (only S7-1500): HTML with Web API and JavaScript

3.2 Πλεονεκτήματα και μειονεκτήματα του web server

Οποιαδήποτε εφαρμογή με web server διαθέτει κάποια πλεονέκτημα και μειονεκτήματα τα οποία αναφέρονται παρακάτω:

Πλεονεκτήματα:

- Δεν απαιτείται έξτρα άδεια Software ούτε hardware. Η εφαρμογή του web server υπάρχει στην σειρά S7-1200 / S7-1500 της Siemens.
- Οι τυπικές ιστοσελίδες για την απλή εμφάνιση υπηρεσιών και διαγνωστικών πληροφοριών ενεργοποιούνται κατευθείαν. Επιπλέον, μπορούν να δημιουργηθούν μεμονωμένα ιστοσελίδες που καθορίζονται από τον χρήστη.
- Η πρόσβαση στην ιστοσελίδα είναι δυνατή μέσω ενός τυπικού προγράμματος περιήγησης στο διαδίκτυο καθώς και μέσω φορητών συσκευών επικοινωνίας, όπως tablet PC, smart phone κλπ.
- Διασφαλίζεται η προστασία μέσω τυπικών μηχανισμών ασφαλείας, όπως VPN κλπ. Η προστασία μέσω κωδικού πρόσβασης είναι επίσης δυνατή.

Μειονεκτήματα:

- Σε κάθε αλλαγή οποιασδήποτε σελίδα html πρέπει να κάνουμε hardware download του προγράμματος, που σημαίνει ότι κάθε φορά θα πέφτει σε stop η CPU.
- Απαιτεί PLC μνήμη. Για παράδειγμα, η CPU 1211C δεν μπορεί να το υποστηρίξει διότι δεν έχει αρκετές δυνατότητες. Το web υποστηρίζεται από CPU 1212C και πάνω.

3.3 Συμβατότητα PLC S7-1200/S7-1500 για την υποστήριξη του web server

Για την υποστήριξη του web server απαιτείται συγκεκριμένη έκδοση της CPU η οποία αναγράφεται παρακάτω:

SIMATIC S71200:

| Plc Version | Client Library Version |
|-------------|------------------------|
| 2.9.x | 1.0.x |
| 2.9.x | 2.0.x |

SIMATIC S71500:

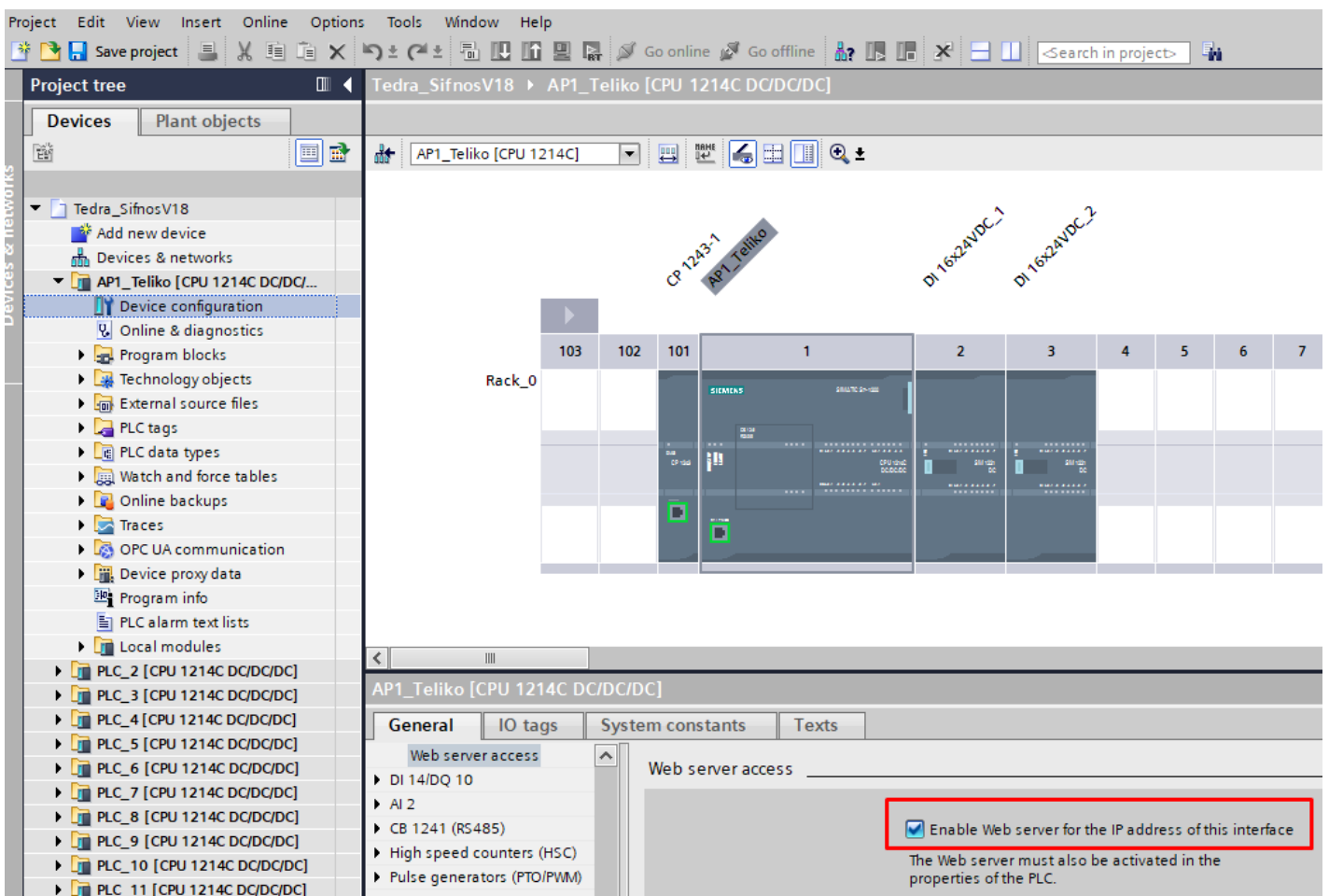
| Plc Version | Client Library Version |
|-------------|------------------------|
| 4.5.x | 1.0.x |
| 4.5.x | 2.0.x |

Κεφάλαιο 4. Πρακτικό μέρος

4.1 Ενεργοποίηση του web server στο PLC

Για την ενεργοποίηση του web server είναι απαραίτητο να γίνει μία ακολουθία επιλογών προκειμένου να λειτουργήσει σωστά το πρόγραμμα αυτό. Οι διαδικασίες είναι συγκεκριμένες κάθε φορά και τα βήματά της περιγράφονται παρακάτω:

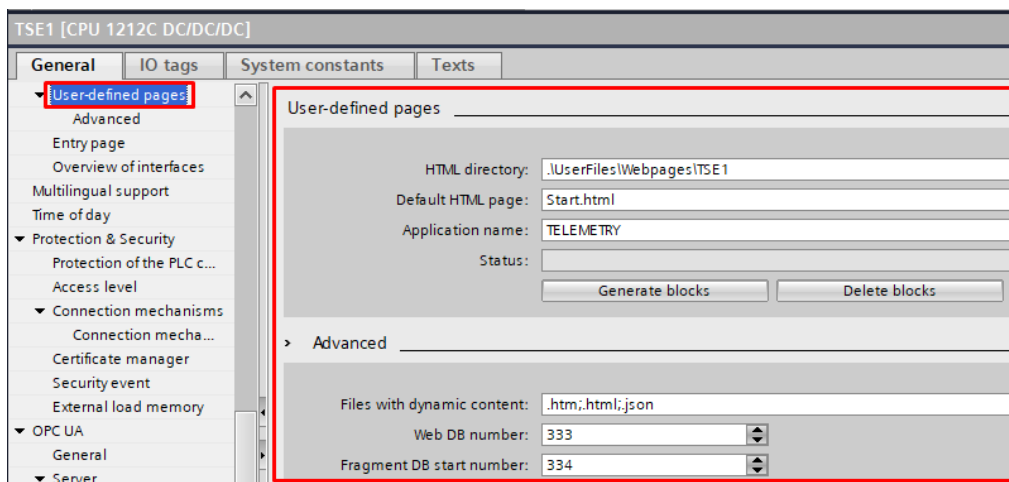
1^ο Βήμα: Στο πρώτο βήμα πρέπει να ενεργοποιήσουμε μέσα από την CPU την επιλογή του web server. Για να γίνει αυτό θα πρέπει να πατήσουμε στο «Device configuration», έπειτα να επιλέξουμε την καρτέλα «Web server access» και να ενεργοποιήσουμε την επιλογή «Activate web server on this module» όπως φαίνεται στην εικόνα παρακάτω:



Εικόνα 13 Ενεργοποίηση του web server μέσα από την CPU

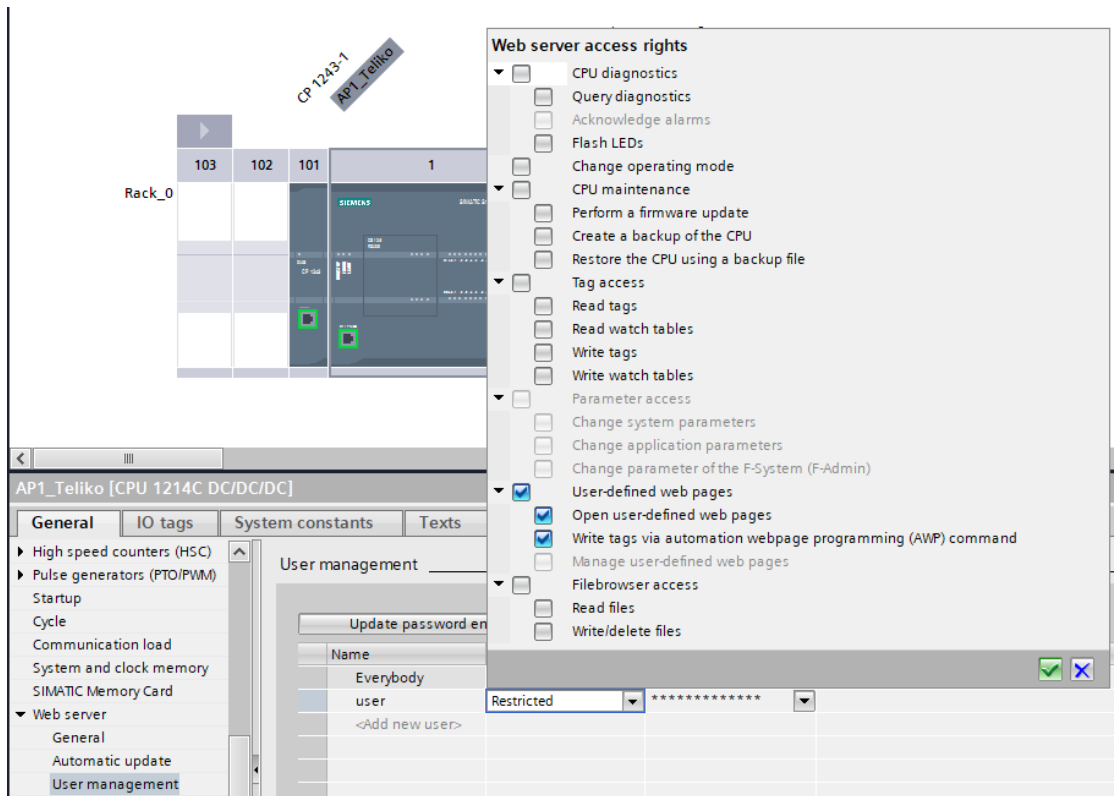
2^ο Βήμα: Στο δεύτερο βήμα πρέπει να δηλώσουμε τον προορισμό του φακέλου μέσα στον οποίο θα δημιουργήσουμε τις σελίδες μας, καθώς επίσης και την αρχική μας σελίδα, αλλά και το όνομα της εφαρμογής. Με την επιλογή του web server δημιουργείται ένας φάκελος μέσα στο project με όνομα «UserFiles». Μέσα σε αυτόν το φάκελο δημιουργούμε για δική μας ευκολία έναν φάκελο με όνομα «Webpages» και έπειτα δημιουργούμε έναν ακόμη φάκελο με το όνομα «TSE1». Το path για τον web server είναι πλέον το παρακάτω: Όνομα project -> UserFiles -> Webpages -> TSE1. Επομένως, στην επιλογή HTML directory θα πρέπει να πατήσουμε στις τρεις τελείες και να επιλέξουμε τον φάκελο TSE1. Στην επιλογή default HTML page επιλέγουμε την σελίδα που θέλουμε να είναι αρχική. Δηλαδή, να εμφανίζεται πρώτη μόλις φορτώσουμε την IP του PLC. Στην επιλογή Application name γράφουμε το όνομα που θέλουμε να εμφανίζεται όταν φορτώνει ο web server όπως θα φανεί στην επόμενη ενότητα.

Στη συνέχεια, πρέπει να πατήσουμε στο «Generate blocks» προκειμένου να μας δημιουργήσει τις σελίδες με τις αλλαγές. Να αναφέρουμε εδώ πως κάθε αλλαγή που κάνουμε στις σελίδες μας πρέπει να τις “κοινοποιούμε”. Πρέπει δηλαδή από την επιλογή «User-defined pages» να πατάμε το «Generate blocks» προκειμένου να μας δείξει τις αλλαγές στην επαναφόρτωση της σελίδας του web. Στην παρακάτω εικόνα φαίνονται οι επιλογές που πρέπει να κάνουμε.



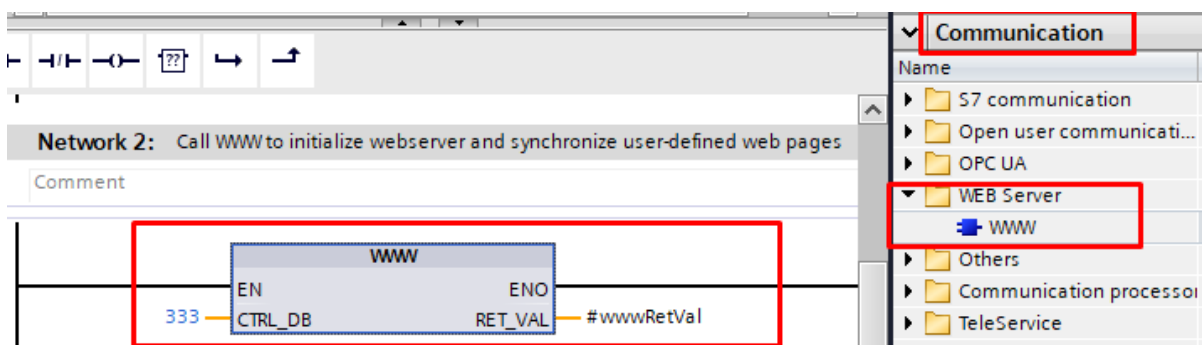
Εικόνα 14 Δήλωση αρχικής σελίδας και ονομασία εφαρμογής

3^ο Βήμα: Στο τρίτο βήμα πρέπει να δημιουργήσουμε έναν χρήστη ο οποίος θα συνδέεται μόλις φορτώνουμε την σελίδα του web. Όπως φαίνεται στην εικόνα παρακάτω μπορούμε να δηλώνουμε στον χρήστη που ακριβώς θέλουμε να έχει πρόσβαση και τι να βλέπει. Για παράδειγμα, εάν τον δηλώσουμε μόνο User-defined web pages σημαίνει πως θα μπορεί να διαχειρίζεται μόνο το web, δεν μπορεί να δει τα διαγνωστικά της CPU διότι δεν έχουμε επιλεχθεί τα δικαιώματα αυτά από τον προγραμματιστή. Συνεπώς, δηλώνουμε το όνομα με το οποίο θα συνδέεται ο χειριστής, του δηλώνουμε τα δικαιώματα που θέλουμε να έχει και του δημιουργούμε έναν κωδικό. Σε αυτό το έργο οι κωδικοί είναι οι εξής: user: **user** / password: **pla2022**.



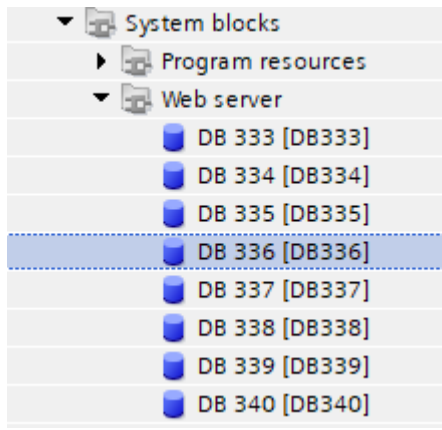
Εικόνα 15 Δημιουργία δικαιωμάτων διαχειριστή

4^ο Βήμα: Στο τέταρτο βήμα πρέπει να επιλέξουμε από την βιβλιοθήκη της Siemens από την κατηγορία «Communications», από τον φάκελο WEB Server όπως φαίνεται παρακάτω στην εικόνα την βιβλιοθήκη WWW. Θα πρέπει δηλαδή να το κάνουμε drag and drop στην Main και να του δηλώσουμε σαν CTRL_DB το 333 και στο RET_VAL (στην επιστροφή τιμής δηλαδή) να δημιουργήσουμε μία τοπική – local μεταβλητή. Αυτό το χρειαζόμαστε ώστε να αρχικοποιήσουμε το web server και να συγχρονίσουμε τις σελίδες που θα δημιουργήσουμε.



Εικόνα 16 Προσθήκη της βιβλιοθήκης WWW

Εφόσον γίνουν οι παραπάνω ρυθμίσεις, το πρόγραμμα δημιουργεί στα «System blocks» μία κατηγορία με την ονομασία «Web Server» στην οποία δημιουργεί τα συστημικά data block που χρειάζεται για την ορθή λειτουργία του web server. Ο προγραμματιστής δεν χρειάζεται να κάνει καμία τροποποίηση σε αυτή την κατηγορία.



Εικόνα 17 Δημιουργία συστημικών data block

4.2 Επεξήγηση βασικών περιεχομένων για το web

Σε αυτή την ενότητα θα περιγράψουμε τον προγραμματισμό του web server και θα αναλύσουμε τις εντολές που χρησιμοποιήθηκαν και με ποιο σκοπό. Όπως φαίνεται στην παρακάτω εικόνα έχουμε δημιουργήσει τις σελίδες που χρειαζόμαστε για το πρόγραμμα. Η σελίδα start.html είναι η αρχική σελίδα που φορτώνει με την πρώτη σύνδεση του web (αναφέρεται και στη προηγούμενη ενότητα – **Εικόνα 14**). Πρέπει να δηλωθεί στο configuration του PLC. Η σελίδα του inverter έχει την απεικόνιση των αντλιών με τις παραμέτρους που θέλουμε να απεικονίζονται. Η σελίδα sensor αφορά την αναλογική στάθμη την οποία μετράμε και μπορούμε να ρυθμίσουμε.

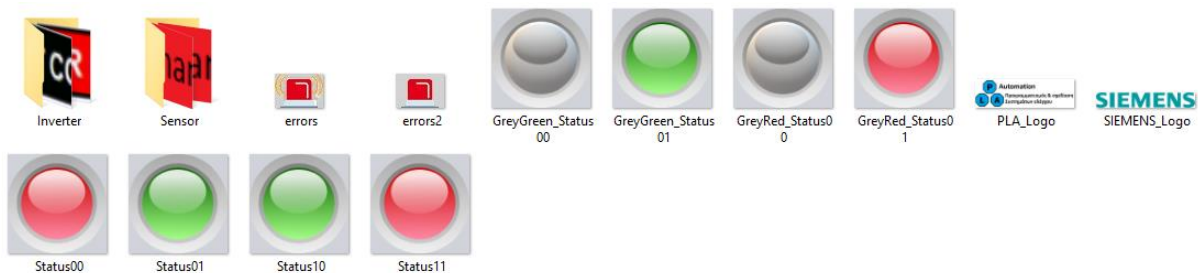
| | |
|--------------------|---------------------|
| Images | 16/11/2022 02:47 μμ |
| js | 16/11/2022 02:47 μμ |
| Stylesheet | 16/11/2022 02:47 μμ |
| Inverter1 | 08/11/2022 12:09 μμ |
| Inverter1.html.bak | 31/01/2022 06:13 μμ |
| Inverter2 | 08/11/2022 12:09 μμ |
| Inverter2.html.bak | 31/01/2022 06:07 μμ |
| Sensor1 | 04/11/2022 05:06 μμ |
| Start | 17/05/2022 02:37 μμ |
| Start.html.bak | 31/01/2022 06:11 μμ |

Εικόνα 18 Περιεχόμενα φακέλου webpages

4.2.1 Επεξήγηση φακέλου Images

Σε κάθε συσκευή, καθώς επίσης και στα ψηφιακά σήματα τα οποία είναι σημαντικά για την ορθή λειτουργία του συστήματος, είναι απαραίτητη η χρωματική απεικόνιση για την καλύτερη και ευκολότερη κατανόηση της λειτουργίας τους. Για να δηλώσουμε τα χρώματα σε ψηφιακά σήματα, όπως για παράδειγμα, στον επιτηρητή τάσης ή στα φλοτέρ (σε κανονική κατάσταση πράσινο σε σφάλμα κόκκινο) χρησιμοποιούμε έναν κύκλο με το αντίστοιχο χρώμα.

Τα στοιχεία που περιέχει ο φάκελος εμφανίζονται παρακάτω:



Εικόνα 19 Περιεχόμενα φακέλου Images

Ανάλογα με τις συσκευές που έχουμε, δημιουργούμε και τους αντίστοιχους φακέλους. Τα status00, status01 ακολουθούν τα ψηφιακά σήματα με τα χρώματα πράσινο και κόκκινο. Σε αυτόν το φάκελο εισάγουμε επίσης και το λογότυπο που θέλουμε να φαίνεται κατά τη φόρτωση της σελίδας. Συνεπώς, στον φάκελος Images μπορούμε να εισάγουμε οποιαδήποτε εικόνα θέλουμε να χρησιμοποιήσουμε για τον προγραμματισμό μας.

Για τα ψηφιακά σήματα χρησιμοποιείται το κόκκινο και το πράσινο όπως αναφέρθηκε και παραπάνω. Η απεικόνιση εμφανίζεται παρακάτω:



Όταν ένα ψηφιακό σήμα έχει σφάλμα χρησιμοποιείται το κόκκινο χρώμα.

Εικόνα 20 Σφάλμα
σήματος



Όταν ένα ψηφιακό σήμα είναι σε ορθή λειτουργία χρησιμοποιείται το πράσινο χρώμα.

Εικόνα 21 Λειτουργία
σήματος

Παρακάτω απεικονίζονται τα χρώματα για την κατάσταση της αντλίας (λειτουργία, βλάβη κλπ. και σε μορφή κειμένου αλλά και στην αντλία) καθώς επίσης, και ο έλεγχος της αντλίας (τοπικός, αυτόματος, χειροκίνητος). Κάθε κατάσταση έχει το δικό της χρώμα για να ξεχωρίζει οπτικά.

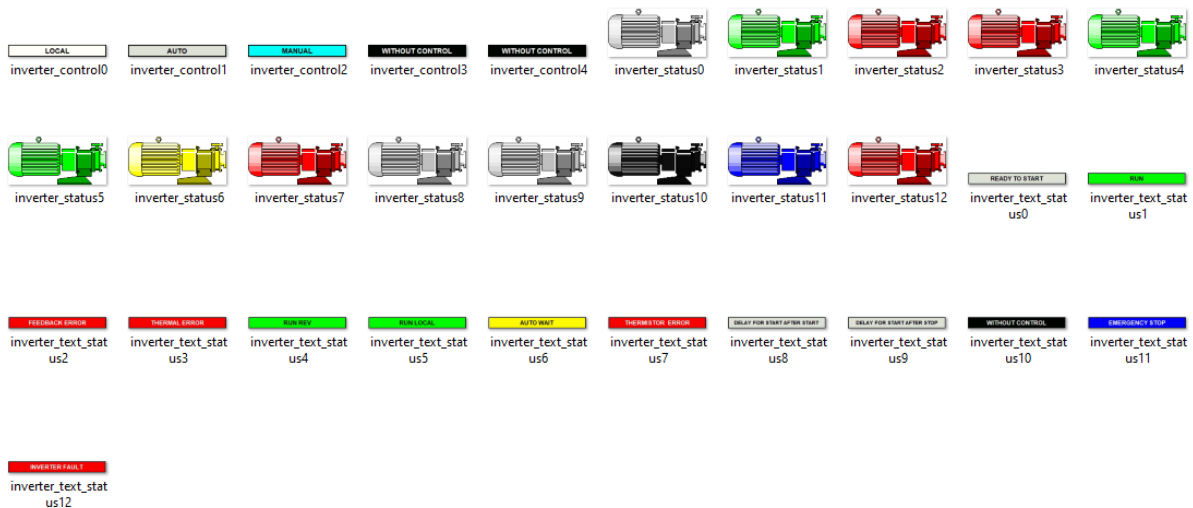
4.2.1.1 Επεξήγηση χρωμάτων Inverter

Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, υπάρχουν κάποια χρώματα για την αντλία όπου στην περίπτωση που γίνει κάποια βλάβη να την καταλάβει αμέσως ο χειριστής και να την αντιμετωπίσει.

Κατάσταση Inverter:

- Ready to start – γκρι χρώμα: Σημαίνει ότι η συσκευή είναι ανενεργή και έτοιμη να ξεκινήσει.
- Run – πράσινο χρώμα: Σημαίνει ότι η συσκευή είναι σε λειτουργία.
- Feedback error – κόκκινο χρώμα: Σημαίνει ότι το inverter ξεκίνησε αλλά δεν ήρθε η είσοδος του σήματος της επιβεβαίωσης.
- Thermal error – κόκκινο χρώμα: Σημαίνει ο έχει πέσει το θερμικό της συσκευής (συνήθως επειδή ο κινητήρας ζορίζεται).
- Run Rev – πράσινο χρώμα: Σημαίνει ότι η συσκευή λειτουργεί με αντίστροφη φορά. (Υπάρχουμε κινητήρες που μπορούν και γυρίζουμε και από τις δύο φορές. Για να ξεχωρίσουμε την αλλαγή της φοράς η ένδειξη είναι πράσινη αλλά το κείμενο γράφει Run Rev).
- Run Local – πράσινο: Σημαίνει ότι η συσκευή λειτουργεί τοπικά από τον πίνακα. Δηλαδή, έχει πάει ο χειριστής στον πίνακα, έχει γυρίσει τον αυτόματο διακόπτη σε δική του λειτουργία και έχει κάνει start. Εμφανίζεται επίσης, και όταν σταματάει ο κινητήρας είτε από αυτοματισμό είτε από το web και υπάρχει ράμπα καθόδου. Αυτό σημαίνει ότι έχει πάρει εντολή παύσης αλλά έχει κάποια δευτερόλεπτα για να σταματήσει.
- Auto Wait – Κίτρινο: Σημαίνει ότι η συσκευή είναι στο αυτόματο και έτοιμη να ξεκινήσει. Πρέπει να ικανοποιηθούν όλα τα interlock για να μπορέσει να ξεκινήσει η συσκευή (π.χ στάθμη εντός ορίων).
- Thermistor Error – Κόκκινο χρώμα: Σημαίνει ότι ο κινητήρας έβγαλε σφάλμα θερμίστορ. Δεν λαμβάνεται υπόψη πάντα αυτό το σήμα, μόνο όταν χρειάζεται.
- Delay for start after start – σκούρο γκρι χρώμα: Σημαίνει ότι η συσκευή περιμένει να περάσει ο χρόνος που έχουμε ορίσει στο πεδίο (θα επεξηγηθεί σε επόμενη ενότητα) έπειτα από εκκίνηση για να μπορέσει να ξανά ξεκινήσει.
- Delay for start after stop – σκούρο γκρι χρώμα: Σημαίνει ότι η συσκευή περιμένει να περάσει ο χρόνος που έχουμε ορίσει στο πεδίο (θα επεξηγηθεί σε επόμενη ενότητα) έπειτα από σταμάτημα για να μπορέσει να ξανά ξεκινήσει. Δηλαδή, εφόσον σταματήσει η συσκευή να μην μπορέσει να ξανά ξεκινήσει αμέσως. Αυτό γίνεται συνήθως στις αντλίες για να μην δημιουργούνται πλήγματα και σπάσει ο αγωγός.
- Without control – μαύρο χρώμα: Σημαίνει ότι του λείπουν βασικά σήματα για να μπορέσει να ξεκινήσει η αντλία. Δηλαδή, εάν ο διακόπτης του πίνακα είναι γυρισμένος στη θέση 0 (εκτός ελέγχου) τότε θα γράφει ότι η συσκευή είναι εκτός ελέγχου.

- Emergency Stop – μπλε χρώμα: Σημαίνει ότι δεν έρχονται τα σήματα που καθιστούν ασφαλές το σύστημά μας όπως για παράδειγμα ο επιτηρητής τάσης ή το κουμπί του emergency stop. Όλα αυτά τα σήματα που καθιστούν ασφαλές το σύστημα μπαίνουν σε σειρά και ενεργοποιούν μία μνήμη.
- Inverter fault – κόκκινο χρώμα: Σημαίνει ότι υπάρχει σφάλμα στο inverter. Συνήθως, είναι μία είσοδος που μας ενεργοποιείται στο PLC για να δηλώσει το σφάλμα.



Εικόνα 22 Χρώματα Inverter

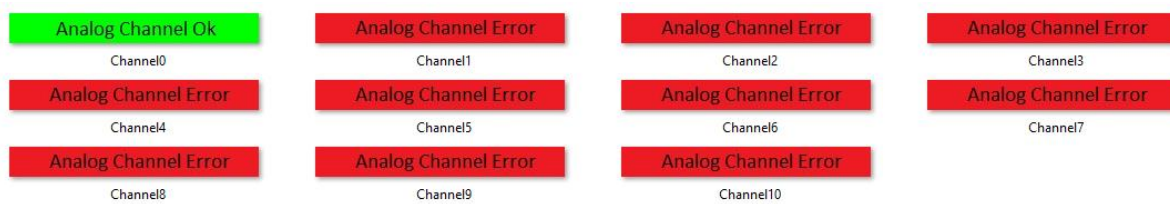
Έλεγχος Inverter:

- Local – άσπρο χρώμα: Σημαίνει ότι ο διακόπτης του πίνακα είναι γυρισμένο σε τοπικό έλεγχο. Αυτό σημαίνει ότι δεν λειτουργεί η αντλία από τον αυτοματισμό αλλά με το start επάνω στον πίνακα.
- Auto – γκρι χρώμα: Σημαίνει ότι η αντλία είναι γυρισμένη σε αυτοματισμό και παίζει με την στάθμη ή τα φλοτέρ αντίστοιχα.
- Manual – γαλάζιο χρώμα: Σημαίνει χειροκίνητη λειτουργία. Ρυθμίζεται μέσα από το web και το scada και μπορούμε να κάνουμε εμείς start-stop την αντλία από το web. Και σε αυτή τη λειτουργία δεν παίζει ο αυτοματισμός. Η λειτουργία αυτή είναι για τους ελέγχους που χρειάζεται προφανώς να κάνει ο χειριστής.

Να τονιστεί σε αυτό το σημείο ότι για να μπορεί η αντλία να είναι σε αυτόματη ή χειροκίνητη κατάσταση πρέπει ο διακόπτης του ηλεκτρολογικού πίνακα να είναι γυρισμένος σε remote.

4.2.1.2 Επεξήγηση χρωμάτων αναλογικού

Η πιο σημαντική πληροφορία του αναλογικού οργάνου είναι η σύνδεσή του. Σε περίπτωση που το όργανο είναι ασύνδετο ή έχει καεί η είσοδος θα πρέπει να απεικονίζεται στο παράθυρο του αναλογικού. Η μεταβλητή που δείχνει την κατάσταση του οργάνου ονομάζεται «Channel Status» και ανάλογα με την κατάσταση αναφέρεται το κατάλληλο κείμενο και εμφανίζεται το κατάλληλο χρώμα. Στην περίπτωση που το αισθητήριο είναι συνδεδεμένο χωρίς πρόβλημα τότε το πεδίο γίνεται πράσινο και επάνω γράφει «Analog Channel Ok». Στην περίπτωση, που το αισθητήριο είναι καμένο τότε το πεδίο γίνεται κόκκινο χρώμα και γράφει «Analog Channel Error». Παρακάτω εμφανίζονται οι εικόνες του φακέλου Images -> Sensor. Σε επόμενη ενότητα θα εξηγηθεί ο λόγος που υπάρχουν πολλές κόκκινες εικόνες και όχι μία εικόνα για το σφάλμα.



Εικόνα 23 Χρώματα αναλογικού αισθητηρίου

4.2.2 Επεξήγηση φακέλου Stylesheet

Η σχεδίαση του web είναι ένας προγραμματισμός αρκετά σύνθετος. Απαιτεί κώδικα για τη δομή της σχεδίασης του template αλλά και για τη στοίχιση των στοιχείων που εμπλέκονται σε αυτό. Η siemens παρέχει ένα demo project μέσα στο οποίο υπάρχει ο φάκελος Stylesheet ο οποίος είναι χρηστικός για τον βασικό κορμό του web. Τα αρχεία που περιέχει ο φάκελος φαίνονται στην παρακάτω εικόνα.

| | | | |
|----------------------------|---------------------|-----------------|--------|
| # bootstrap.min | 10/07/2019 12:33 μμ | CSS Source File | 153 KB |
| # siemens_Stylesheet | 07/01/2022 04:00 μμ | CSS Source File | 7 KB |
| siemens_Stylesheet.css.bak | 31/01/2022 05:53 μμ | BAK File | 7 KB |

Εικόνα 24 Περιεχόμενα φακέλου Siemens Stylesheet

Από τα τρία αυτά αρχεία που περιέχει ο φάκελος της Siemens μπορούμε να επέμβουμε στον «siemens_Stylesheet». Τα πιο σημαντικά στοιχεία μέσα από τον κώδικα του αρχείου περιγράφονται παρακάτω.

```

1  body {
2    font-family : Arial, Helvetica, sans-serif;
3    margin : 0px;
4    text-align : left;
5    text-decoration : none;
6    color : #000000;
7    overflow : auto;
8  }
9
10 #header {
11   position: absolute;
12   width: 1920px;
13   height: 160px;
14   left: 0px;
15   top: 0px;
16   background-color: rgb(244,164,96);
17   z-index: 2;
18 }
19
20 #header h2{
21   padding-top: 10px;
22   padding-bottom: 0px;
23   padding-left: 20px;
24   font-size: 26px;
25 }
26
27
28
29 .plcLoginContainer{
30   height: 100%;
31 }
32
33 .plc-login{
34   width: 500px;
35   height: 400px;
36   border-radius: 10px;
37   margin-left: auto;
38   margin-right: auto;
39   border-style: solid;
40   border-color: cornflowerblue;
41   position: relative;
42   top: 20%;
43   box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
44 }

```

Εικόνα 25 Κώδικας siemens_Stylesheet - 1

Όπως φαίνεται στην εικόνα αριστερά δημιουργούνται κάποιες κατηγορίες οι οποίες αναφέρονται μέσα στον κώδικα του προγράμματος που αναπτύσσουμε για το web.

Υπάρχουν στοιχεία που μπορούμε να προσθέσουμε και κάθε κατηγορία όπως είναι το width (πλάτος που θα καταλαμβάνει το συγκεκριμένο αντικείμενο), το height (το ύψος που θα καταλαμβάνει το αντικείμενο), το text-align (η στοίχιση των γραμμάτων, όπως για παράδειγμα αριστερά, κέντρο, δεξιά), το χρώμα των γραμμάτων κλπ.

Η κατηγορία header αφορά το template που θα έχει επάνω το web server εφόσον φορτώσει. Μπορούμε να αλλάξουμε το πλάτος που θα καταλαμβάνει από το width καθώς επίσης και το χρώμα.

Η κατηγορία header h2 επικεφαλίδα. Όταν γράψουμε <h2> και το κείμενο <h2> θα πάρει τα στοιχεία που βρίσκονται μέσα στον κώδικα του header h2.

```

46 .plc-login form{
47 |   padding: 20px;
48 }
49
50
51 #content{
52 |   position: absolute;
53 |   top: 100px;
54 |   left: 0px;
55 |   bottom: 0;
56 |   right: 0;
57 |   padding: 0px 20px 0 20px;
58 |   margin: 0;
59 |   z-index: 5;
60 |   background-color: #eaeaea;
61 |   vertical-align: top;
62 |   border: 0;
63 |   border-collapse: collapse;
64 |   overflow: auto;
65 }
66
67 /*****
68 /* BOX with Header Styles */
69 #content div.container{
70 |   border: 1px solid #444;
71 |   /* min-width: 350px; */
72 |   width: 250px;
73 |   border-radius: 10px;
74 |   padding: 15px;
75 |   margin: 20px 20px 10px 0;
76 |   float: left;
77 }
78 #content div.container h1{
79 |   margin: -25px 0px 10px 0;
80 |   font-size: 16px;
81 }
82 #content div.container h1 span{
83 |   padding: 0 10px;
84 |   background-color: #eaeaea;
85 }
86

```

Εικόνα 27 Κώδικας siemens_Stylesheet - 2

```

121 #content div.containerPage {
122 |   POSITION: absolute;
123 |   left: 198px;
124 |   top: 0;
125 |   bottom: 0;
126 |   right : 0;
127 |   min-width: 920px;
128 |   padding-top: 10px;
129 |   padding-left: 30px;
130 |   padding-right: 30px;
131 |   text-align: left;
132 |   border-color: white;
133 |   border-style: solid;
134 |   border-width: 1px;
135 |   background-color: rgb(208,211,218);
136 |   border-collapse : separate;
137 |   z-index: 1;
138 }
139
140 #content div.containerPage h3 {
141 |   color: rgb(0,51,153);
142 }
143 #content div.containerPage h4 {
144 |   color: rgb(0,0,0);
145 |   font-size: 12px;
146 |   font-weight: bold;
147 }
148
149 #content div.containerPage2 {
150 |   POSITION: absolute;
151 |   left: 600px;
152 |   top: 0;
153 |   padding-top: 50px;
154 |   background-color: rgb(208,211,218);
155 |   border-collapse : separate;
156 |   z-index: 1;
157 }
158
159 #content div.containerPage2 h3 {
160 |   color: rgb(0,51,153);
161 }

```

Εικόνα 26 Κώδικας siemens_Stylesheet - 3

Στις παραπάνω εικόνες αναφέρονται σε πιο γενικό βαθμό οι συντεταγμένες των περιεχομένων. Σε κάθε σελίδα που έχουμε δημιουργήσει (inverter,sensor κλπ.) έχουμε αναφερθεί στην κατηγορία #content div.containerNavi (όπως θα εξηγηθεί και παρακάτω) για να μπορέσουμε να δημιουργήσουμε το template στα αριστερά του web για την πλοήγηση στις σελίδες. Μπορούμε αλλάζοντας στην **Εικόνα-17** το width να κάνουμε το μενού πιο λεπτό ή πιο φαρδύ, να κάνουμε έντονο το border γύρω γύρω από το μενού ή ακόμα και να του αλλάξουμε χρώμα αλλάζοντας το «background-color».


```

182  /* form styles */
183  input[type="submit"], input[type="button"] {
184      height: 45px;
185      width: 150px;
186      font-size: 16px;
187      text-align: center;
188      font-weight: bold;
189  }
190
191  input[type="submit2"], input[type="button"] {
192      height: 40px;
193      width: 80px;
194      font-size: 8px;
195      text-align: center;
196  }
197
198  input[type="submit3"], input[type="button"] {
199      height: 10px;
200      width: 40px;
201      font-size: 15px;
202      padding: 10 40px;
203      border-radius: 8px;
204      text-align: center;
205  }
206
207  input[type="text"]#flowrate {
208      height: 45px;
209      width: 50px;
210      font-size: 16px;
211      text-align: center;
212  }
213  input[type="number"] {
214      height: 45px;
215      width: 50px;
216      font-size: 16px;
217      text-align: center;
218  }
219
220  input[type="number2"] {
221      height: 45px;
222      width: 50px;
223      font-size: 32px;
224      text-align: center;
225  }

```

Το πιο σημαντικό κομμάτι του προγραμματισμού αφορά τους τύπους «submit» και «number».

Όπως θα αναφερθεί και στην επεξήγηση του δικού μας προγράμματος στην επόμενη ενότητα χρησιμοποιούμε δύο κατηγορίες για αλλαγή των μεταβλητών. Η μία κατηγορία αφορά το «πεδίο – io field» και η άλλη κατηγορία αφορά το «κουμπί - button».

Πεδίο: Για την εισαγωγή μιας τιμής (π.χ ένα set point) χρησιμοποιούμε το «input type=number2». Το πεδίο αυτό θα πάρει τα περιεχόμενα που φαίνονται στην εικόνα αριστερά. Δηλαδή θα έχει ύψος 45px και πλάτος 50px. Το πάχος των γραμμάτων θα είναι 32px και η στοίχισή του αριθμού θα είναι στο κέντρο.

Κουμπί: Για την αλλαγή μια κατάστασης με κουμπί (π.χ start,stop) χρησιμοποιούμε το «input type="submit"» για το όνομα του κουμπιού και το «input type="hidden"» για να δηλώσουμε την αλλαγή της μεταβλητής.

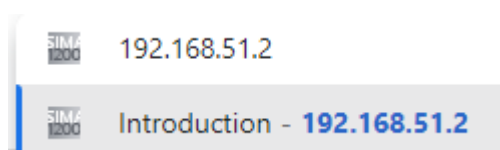
Εικόνα 28 Κώδικας siemens_Stylesheet - 4

4.3 Σύνδεση και προγραμματισμός του web server

4.3.1 Σύνδεση στον web server

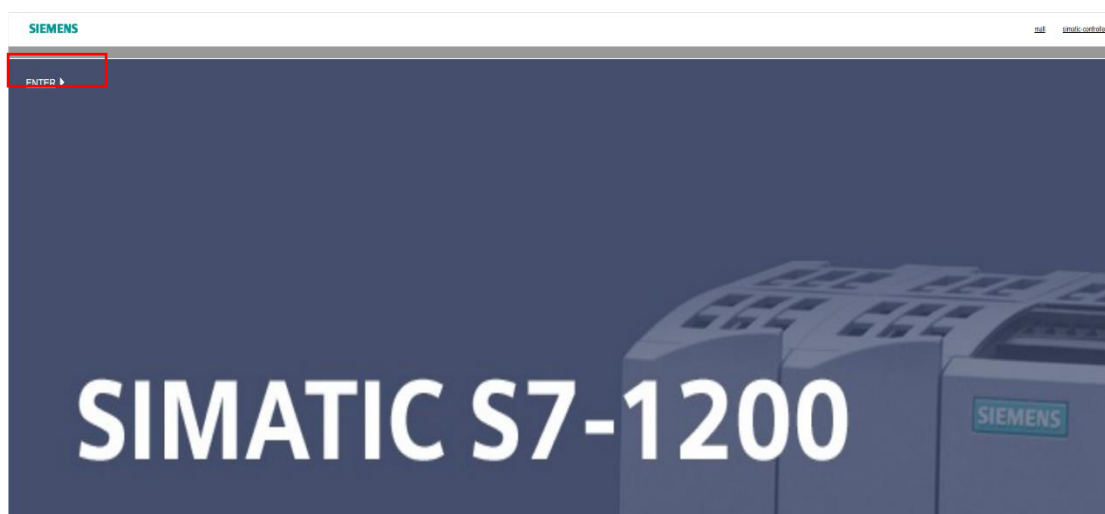
Εφόσον έχουμε ενεργοποιήσει επιτυχώς τον web server (Ενότητα 4.1) και προγραμματίσαμε τις σελίδες που μας χρειάζονται για τον χειρισμό του αντλιοστασίου μπορούμε πλέον να συνδεθούμε και στον web server όπως περιγράφεται παρακάτω.

- Για να συνδεθούμε στο web πρέπει να ξεκινήσουμε αρχικά έναν browser (οποιοδήποτε θέλουμε chrome, Mozilla κλπ.) και να πληκτρολογήσουμε την IP του PLC στο οποίο έχουμε ενεργοποιήσει το web. Στη συγκεκριμένη περίπτωση, πληκτρολογούμε το 192.168.51.2 όπως φαίνεται στην εικόνα παρακάτω.



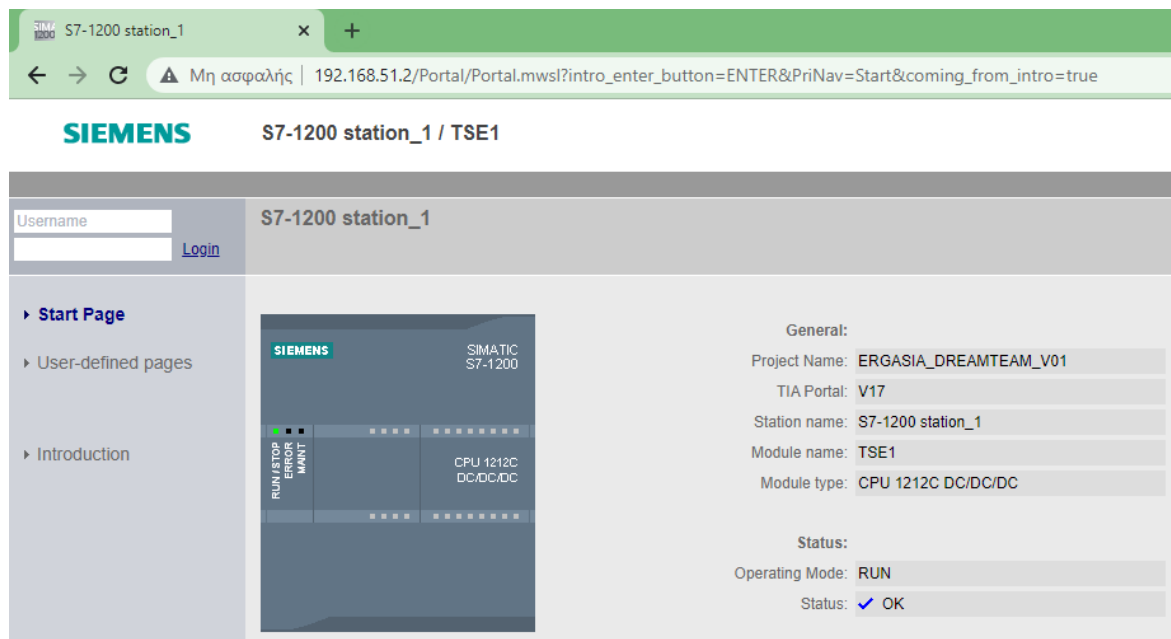
Εικόνα 29 IP Web Server

- Μόλις πατήσουμε την IP του PLC θα μας εμφανιστεί η εικόνα της Siemens (**Εικόνα 30**) στην οποία θα πρέπει να πατήσουμε το κουμπί enter για να συνδεθούμε.



Εικόνα 30 Σύνδεση στο Web Server - 1

- Αφού πατήσουμε το enter θα συνδεθούμε στην κύρια σελίδα του web στην οποία εμφανίζονται κάποια στοιχεία για το PLC καθώς επίσης και για το project.



Εικόνα 31 Σύνδεση στο Web Server - 2

Στην παραπάνω εικόνα εμφανίζεται η κύρια σελίδα του web. Επάνω αριστερά εισάγουμε τα στοιχεία του χρήστη που έχουμε κάνει στο PLC και αναφέρθηκαν σε προηγούμενη ενότητα (user: **user** / password: **pla2022**).

Υπάρχουν και κάποιες σελίδες στις οποίες μπορούμε να πλοηγηθούμε. Η σελίδα που μας δίνει πρόσβαση στο web είναι η «User-defined pages» στην οποία μπορούμε να συνδεθούμε εφόσον συνδεθούμε με τα στοιχεία που δημιουργήσαμε.

Στο κέντρο της οθόνης απεικονίζεται η CPU και η κατάστασή της. Εάν είναι σε RUN το λαμπάκι του RUN/STOP είναι πράσινο ειδάλλως πορτοκαλί. Εάν η CPU εμφανίζει κάποιο σφάλμα τότε το led του ERROR θα γίνει κόκκινο. Επίσης, απεικονίζεται και ο τύπος της CPU (στην συγκεκριμένη περίπτωση η CPU που χρησιμοποιήθηκε είναι η S7-1200 CPU1212C DC/DC/DC. Στην δεξιά πλευρά της οθόνης αναγράφεται το όνομα του project που είναι το PLC, η έκδοση του προγράμματος του TIA Portal (V17), το όνομα του σταθμού, το όνομα και ο τύπος της συσκευής. Ακόμη, αναγράφεται με κείμενο η κατάσταση της CPU.

- Αφότου βάλουμε τα στοιχεία μας και συνδεθούμε θα μας ζητηθεί να αποδεχθούμε τους όρους κινδύνου για να μπορούμε να συνεχίσουμε. Χρειάζεται επομένως, να πατήσουμε στο κουμπί «Advanced» και έπειτα στο «Proceed to 192.168.51.2 (unsafe)».



Your connection is not private

Attackers might be trying to steal your information from 192.168.51.2 (for example, passwords, messages, or credit cards). [Learn more](#)

NET::ERR_CERT_AUTHORITY_INVALID



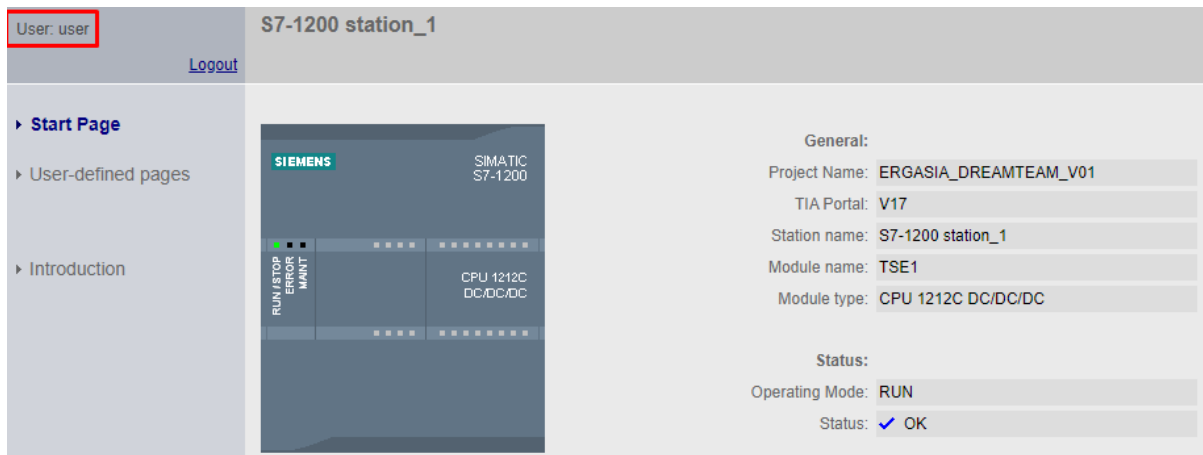
Εικόνα 32 Επιλογή Advanced



Εικόνα 33 Επιλογή proceed to

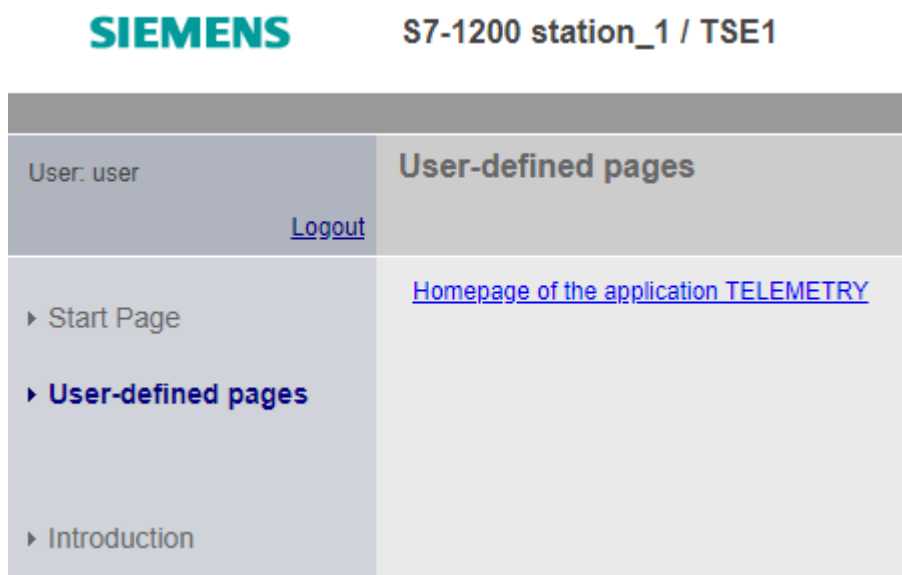
Στη συνέχεια, θα εμφανιστεί ένα πλαίσιο επάνω στην οθόνη στο οποίο θα πρέπει να επιλέξουμε το «continue». Εφόσον πατήσουμε στη συνέχεια καλό είναι να ξανά φορτώσουμε την σελίδα μας για να μην έχουμε κάποιο πρόβλημα.

- Έχουμε πλέον συνδεθεί με τον χρήστη που δημιουργήσαμε και έχουμε αποδεχτεί τους όρους φόρτωσης της σελίδας. Όπως φαίνεται και στην παρακάτω εικόνα έχει συνδεθεί ο χρήστης user.

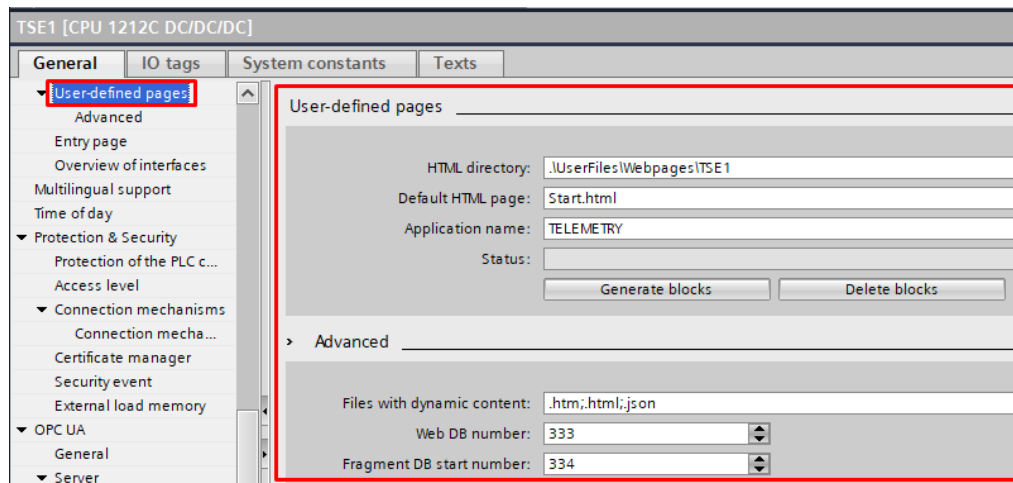


Εικόνα 34 Συνδεμένος χρήστης ο user

- Εφόσον είμαστε πλέον συνδεδεμένοι πρέπει να επιλέξουμε την κατηγορία «User-defined pages». Στη συνέχεια θα μας ανοίξει μια σελίδα με το κείμενο «Homepage of the application TELEMETRY». Το όνομα TELEMETRY είναι το όνομα που δηλώσαμε στο PLC στην κατηγορία «Application name».

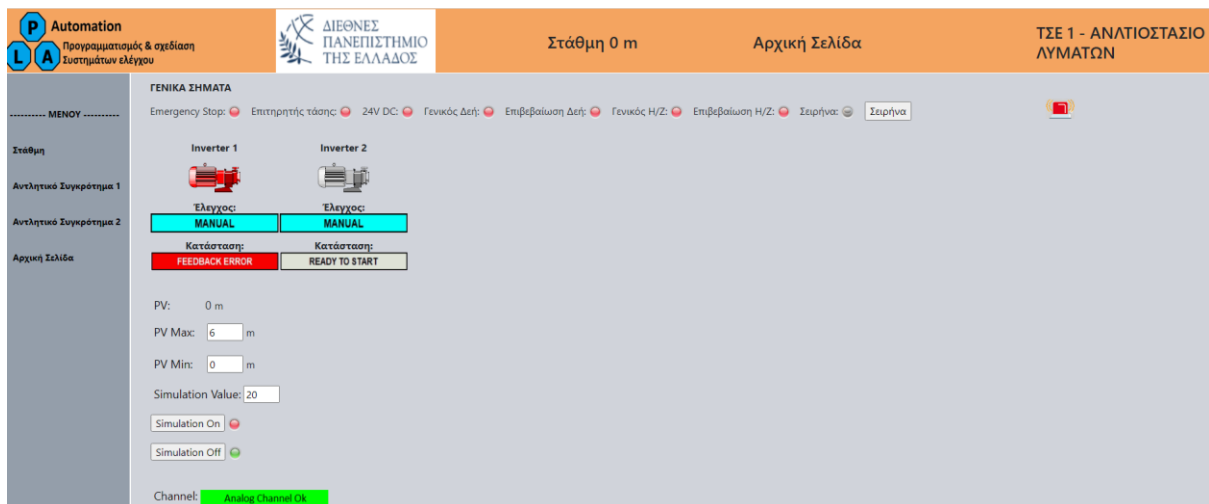


Εικόνα 35 Σύνδεση στις σελίδες του Web Server



Εικόνα 36 Application Name - Δήλωση στο PLC

- Τέλος, μόλις επιλέξουμε το «Homepage of the application TELEMETRY» θα μας ανοίξει στο περιβάλλον που έχουμε σχεδιάσει όπως φαίνεται στην παρακάτω εικόνα. Σε αυτή τη σελίδα έχουμε μία γενική εικόνα των σημάτων, των λειτουργιών των αντλιών καθώς επίσης και για την στάθμη την οποία μπορούμε και να παραμετροποιήσουμε.



Εικόνα 37 Αρχική σελίδα web server

4.3.2 Επεξήγηση του προγραμματισμού του web server

Σε κάθε σελίδα, υπάρχει κώδικας για την εμφάνιση της σειρήνας (ποια σφάλματα την ενεργοποιούν), για την δήλωση της συγκεκριμένης σελίδας και για τις βιβλιοθήκες που πρέπει να συμπεριλάβουμε (όπως αυτή της siemens). Ο κώδικας που υπάρχει σε κάθε σελίδα περιγράφεται παρακάτω:



Εικόνα 38 Σειρήνα

```
<style>
.Siren{
background-image: url("Images/errors.png");
background-color: transparent;
background-repeat: no-repeat;
background-position: 0px 0px;
height: 80px
box-shadow:inset 0px 0px 5px 0px #91b8b3;
display:inline-block;
cursor:pointer;
padding:30px 30px;
z-index: 12;
position:absolute;
top:15%;
left:85%;
width: 15px;
height: 15px;
}
.Siren:active {
transform: translateY(1px);
}
</style>
```

Στον κώδικα που φαίνεται αριστερά επεξεργαζόμαστε την σειρήνα. Τη θέση θα έχει, πόσο μήκος και πόσο ύψος. Το όνομα που δώσαμε στην **Εικόνα 39** είναι το errors και βρίσκεται μέσα στον φάκελο images. Με την εντολή **background-image** δηλώνεται η φωτογραφία που θέλουμε να εμφανίζεται όταν προκληθεί κάποιο σφάλμα.

Εικόνα 39 Θέση σειρήνας

Για να μπορέσουμε σε μία σελίδα να φορτώσουμε όλο το περιεχόμενο που φαίνεται στην αρχική οθόνη πρέπει να εισάγουμε ορισμένες βιβλιοθήκες. Η πιο σημαντική είναι αυτή της siemens ώστε να μπορούμε να έχουμε το αριστερό μενού της πλοήγησης σε κάθε σελίδα. Με την εντολή `<head> </head>` δηλώνουμε την αρχική φόρτωση του κώδικα. Με την εντολή `<meta http-equiv="refresh" content="10; URL=Start.html">` δηλώνεται αρχικά, η σελίδα που θέλουμε να φορτώνεται. Στην σελίδα του inverter στο `URL` δηλώνουμε `URL=Inverter1.html`. Με την εντολή `<link type="text/css" rel="stylesheet" href="Stylesheet/siemens_Stylesheet.css"/>` και `<link type="text/css" rel="stylesheet" href="Stylesheet/bootstrap.min.css">` δηλώνουμε τις δύο βιβλιοθήκες τις οποίες έχουμε εισάγει από το demo project της siemens. Η εντολή `<script src="js/jquery-3.6.0.min.js"></script>` εισάγεται για να μπορούμε να γράψουμε script όπως φαίνεται στην παρακάτω εικόνα που αφορά τα σφάλματα. Η εντολή `content` ρυθμίζει το χρόνο ανανέωσης της σελίδας ώστε να παίρνει τις νέες τιμές. Στο συγκεκριμένο project ο χρόνος ρυθμίστηκε σε 10 δευτερόλεπτα. Αυτό σημαίνει πως όταν αλλάζει η κατάσταση του inverter τότε θα την δούμε σε 10 δευτερόλεπτα. Επίσης, εάν επιθυμούμε να αλλάξουμε τιμή σε κάποια μεταβλητή, τότε την αλλάζουμε αλλά πρέπει να περιμένουμε να κάνει ανανέωση το web μέχρι να ξανά αλλάξουμε κάποια άλλη μεταβλητή, διότι μπορεί να συμβεί επάνω στον χρόνο της ανανέωσης της σελίδας και να μην πάρει την αλλαγή. Ανάλογα με τις ανάγκες του έργου ρυθμίζουμε και τον συγκεκριμένο χρόνο.

```

<!DOCTYPE html>
<html>

<head>
  <title>Userdefined Website - Application Start</title>
  <meta charset="UTF-8" >
  <meta http-equiv="refresh" content="10; URL=Start.html">
  <link type="text/css" rel="stylesheet" href="Stylesheet/siemens_Stylesheet.css"/>
  <link type="text/css" rel="stylesheet" href="Stylesheet/bootstrap.min.css">
  <script src="js/jquery-3.6.0.min.js"></script>
</head>

<body>

```

Εικόνα 40 Αρχικές δηλώσεις

Υπάρχουν ορισμένα σφάλματα που ενεργοποιούν την σειράνα. Σε περίπτωση που υπάρχει κάποιο σφάλμα τότε με την εντολή `visibility` εμφανίζουμε την εικόνα η οποία στον φάκελο `images/errors`.

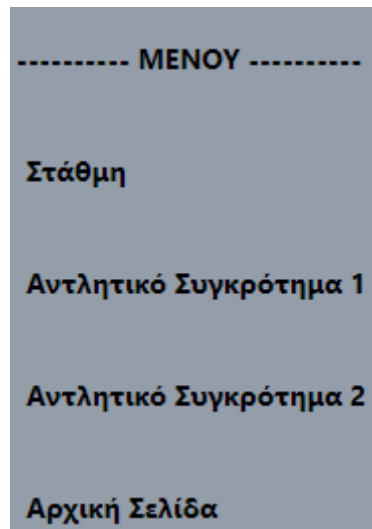
```

<script>
setInterval(function() {
  if(Plc_Enable_Siren_V00_DB1.Signal_on: || Plc_Enable_Siren_V00_DB2.Signal_on: || Plc_Enable_Siren_V00_DB3.Signal_on:){
    document.getElementById("siren").style.visibility = 'visible';}
  else if(Plc_Enable_Siren_V00_DB1.Signal_on: == 0 && Plc_Enable_Siren_V00_DB2.Signal_on: == 0 && Plc_Enable_Siren_V00_DB3.Signal_on: == 0){
    document.getElementById("siren").style.visibility = 'hidden';}
  if($('.Siren').css('background-image').indexOf('errors2')>-1){
    $('.Siren').css('background-image', "url('Images/errors.png')");} else {
    $('.Siren').css('background-image', "url('Images/errors2.png')");}
}, 1000);
</script>

```

Εικόνα 41 Σφάλματα ενεργοποίησης σειράνας

Τέλος, ένα ακόμα κομμάτι προγραμματισμού που είναι κοινό σε κάθε σελίδα είναι η σελίδες που φορτώνονται αριστερά στο μενού όπως φαίνεται και στην εικόνα παρακάτω.



Εικόνα 42 Μενού πλοηγήσεις σε οθόνες

Ο προγραμματισμός του μενού παρουσιάζεται παρακάτω. Με την εντολή `<a>` ορίζεται πάλι ένα κείμενο με συγκεκριμένη γραμματοσειρά. Οι εντολές αυτές αναλύονται στο τέλος της εργασίας. Με την εντολή `Στάθμη` δηλώνεται ότι πατώντας το κουμπί «Στάθμη» που βρίσκεται στα αριστερά θα φορτώσει την σελίδα «Sensor1». Η εντολή `<a>Αρχική Σελίδα` υπάρχει για ομοιομορφία του μενού αριστερά αλλά δηλώνεται σαν απλό κείμενο. Αυτό σημαίνει πως εάν ο χειριστής πατήσει το κείμενο αρχική σελίδα δεν θα αλλάξει η σελίδα διότι είναι απλό κείμενο που σημαίνει ότι βρισκόμαστε σε αυτή τη σελίδα. Η εντολή `
` που εμφανίζεται σε κάποια σημεία δηλώνει ένα μεγάλο κενό για την επόμενη γραμμή. Είναι ένα κενό όπως το tab αλλά λειτουργεί για την γραμμή και όχι για τη στήλη.

```

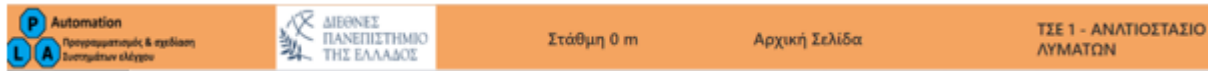
<!-- Content Area -->
|   |   <div id="content">
|   |   <!-- ΜΕΝΟΥ SELIDES START-->
|   |   <div class="containerNavi">
|   |   <ul>
|   |   <a> ----- ΜΕΝΟΥ -----</a><br><br>
|   |   <ul>
|   |   <a href="Sensor1.html">Στάθμη</a><br>
|   |   <br>
|   |   <a href="Inverter1.html">Αντλητικό Συγκρότημα 1</a><br>
|   |   <br>
|   |   <a href="Inverter2.html">Αντλητικό Συγκρότημα 2</a><br>
|   |   <br>
|   |   <a>Αρχική Σελίδα</a><br>
|   |   </ul>
|   |   </ul>
|   |   </div>
|   |   <!-- ΜΕΝΟΥ SELIDES END-->

```

Εικόνα 43 Προγραμματισμός αρχικών σελίδων

• Template προγράμματος

Σε κάθε εφαρμογή scada πρέπει να υπάρχει το γενικό περιβάλλον ή όπως λέγεται στα αγγλικά template. Το template είναι η γενική – global σελίδα που θα υπάρχει σε κάθε άλλη σελίδα διότι περιέχει τις πιο χρήσιμες πληροφορίες και τα λογότυπα των εταιριών. Στο συγκεκριμένο έργο αναγράφεται η στάθμη η οποία έχει καθοριστικό ρόλο στον αυτοματισμό των αντλιών, ο υφιστάμενος σταθμός (ΤΣΕ 1 – ΑΝΤΛΙΟΣΤΑΣΙΟ ΛΥΜΑΤΩΝ), καθώς επίσης, και η σελίδα στην οποία βρισκόμαστε.



Εικόνα 44 Template εφαρμογής

Ο προγραμματισμός για το template περιγράφεται παρακάτω:

```
<!-- EPIKEFALIDA START -->
<div id="header">
  <table border="0" >
    <tr>
      <td width="500px"></td>
      <td width="500px"></td>
      <td width="450px"> <h2>Στάθμη := "PLA_Analog_Read_V02_01".Pv: m</h2></td>
      <td width="650px"><h2>Αρχική Σελίδα</h2></td>
      <td width="400px"><h2>ΤΣΕ 1 - ΑΝΤΛΙΟΣΤΑΣΙΟ ΛΥΜΑΤΩΝ</h2></td>
    </tr>
  </table>
</div>
<!-- EPIKEFALIDA END -->
```

Εικόνα 45 Προγραμματισμός του template σε html

Για δική μας διευκόλυνση δηλώνουμε με σχόλιο τι αφορά το συγκεκριμένο κομμάτι προγραμματισμού. Αυτό καθιστά πιο εύκολη την αποσφαλμάτωση του κώδικα.

Όπως φαίνεται στην παραπάνω εικόνα, ο προγραμματισμός ξεκινάει δηλώνοντας το αντικείμενο που πρόκειται να δείξουμε. Στην προκειμένη περίπτωση είναι επικεφαλίδα - header και συνεπώς δηλώθηκε ως <div id="header">. Κάθε εντολή που ξεκινάει με <και εντολή θα πρέπει και να κλείνει με το αντίστοιχο </εντολή.

Η εντολή **table border** αφορά το περίγραμμα του συγκεκριμένου αντικειμένου που κάνουμε. Εάν θέλουμε να έχει γραμμή γύρω γύρω ο πίνακας τότε θα δηλώσουμε μια τιμή στο border. Όσο πιο μεγάλη είναι αυτή η τιμή τόσο πιο παχύ θα είναι το περίγραμμα.

Η εντολή <tr> είναι για να ανοίξουμε μία γραμμή. Δηλαδή, ότι γράψουμε μέσα στο <tr> μέχρι και να το κλείσουμε </tr> θα γράφονται το ένα δίπλα στο άλλο. Συνεπώς, αφού έχουμε ανοίξει ένα table και μία γραμμή πρέπει να ξεκινήσουμε να γράφουμε τα περιεχόμενα. Κάθε περιεχόμενο είναι καλό να γράφεται μέσα σε ένα table (td) και να δηλώνεται το πόσο φάρδος θα πιάνει στην σελίδα. Εάν δεν δηλωθεί δεν θα έχει σωστή στοίχιση το πρόγραμμα μας με αποτέλεσμα να μην είναι αισθητικά καλό. Το td αφορά το κάθετο κομμάτι του πίνακα. Εφόσον, έχουμε ανοίξει ένα table μετά μπορούμε να ανοίξουμε table με την μορφή <td> </td>. Η σύνταξη επομένως της εικόνα PLA και ΔΠΠΑΕ θα είναι η εξής:

```
<td width="500px"></td>
```

Ανοίγουμε ένα table για την εικόνα PLA με την εντολή <td width="500px">. Ορίζουμε και το μήκος του πίνακα για να μας δεσμεύσει συγκεκριμένο χώρο. Δηλώνουμε τον φάκελο

που έχουμε την εικόνα και το γράφουμε με την μορφή `img src="Images/PLA_Logo.png"`. Δίπλα δηλώνουμε το width και το height της εικόνας ώστε να μην είναι παραμορφωμένη. Καθορίζουμε εμείς το πόσο μεγάλη ή μικρή θέλουμε να είναι η εικόνα. Η εντολή `alt="PLA"` δίνει το χαρακτηριστικό όνομα σε περίπτωση που ξανά χρησιμοποιηθεί το όνομα. Δεν είναι υποχρεωτική να υπάρχει αυτή η εντολή. Όπως φαίνεται στην εικόνα κάθε εντολή που ανοίγει πρέπει να κλείνει αντίστοιχα.

Το ίδιο κάνουμε και για την εικόνα της ΔΠΠΑΕ με έναν άλλο καινούργιο ατομικό πίνακα. Έπειτα, ανοίγουμε άλλον πίνακα για την τιμή της στάθμης. Με την εντολή `<td width>` καθορίζουμε και πάλι το μήκος που θέλουμε να πιάνει το όνομα **Στάθμη** μαζί με το πεδίο και την μονάδα. Όπως φαίνεται στην εικόνα μπροστά από την στάθμη υπάρχει η εντολή `<h2>` `</h2>`. Αναφέραμε στην προηγούμενη ενότητα ότι το `<h2>` έχει καθοριστεί από το `siemens_Stylesheet` με συγκεκριμένο μέγεθος γραμμμάτων. Εάν θέλαμε να κάνουμε πιο μεγάλα ή πιο μικρά τα γράμματα υπάρχουν τις εντολές που θα μπορούσαμε να χρησιμοποιήσουμε και οι οποίες αναφέρονται στο κεφάλαιο **Λίστα Εντολών**. Το πεδίο της στάθμης είναι μόνο output δηλαδή μόνο απεικονίζεται, δεν μπορούμε εμείς να επεμβούμε και να του αλλάξουμε την τιμή. Είναι η τιμή που μετράμε στην αναλογική είσοδο και μας την δίνει το PLC. Τις περιπτώσεις που το πεδίο είναι μόνο output (χωρίς χειρισμό) γράφεται με την εντολή `:=»PLA_Analog_Read_V02_01».Pv:` και ακριβώς δίπλα αναγράφεται η μονάδα μέτρησης. Την εντολή `</h2>` την κλείνουμε μετά την μονάδα μέτρησης, που σημαίνει ότι είναι κείμενο και συμπεριλαμβάνεται στο κομμάτι του h2 θα έχει το συγκεκριμένο μέγεθος γραμμμάτων.

- **Αρχική σελίδα προγράμματος**

Στην αρχική σελίδα του προγράμματος εμφανίζονται όλα τα απαραίτητα ασφαλιστικά που επηρεάζουν την ορθή λειτουργία του αντλιοστασίου. Για να είναι το σύστημα έτοιμο να ξεκινήσει θα πρέπει να έχουμε σε κατάσταση λειτουργίας το emergency stop (να μην είναι πατημένο το μανιτάρι), ο επιτηρητής τάσης ή ασυμμετρίας να είναι εντάξει και να έχουμε ρεύμα είτε από ΔΕΗ είτε από ΗΖ. Η σειρήνα εμφανίζεται όταν υπάρχει σφάλμα και με το κουμπί που υπάρχει στην εικόνα παρακάτω με ονομασία «σειρήνα» μπορούμε να την κάνουμε reset. Υπάρχει η γενική απεικόνιση των αντλιών, με την κατάσταση λειτουργίας τους για την διευκόλυνση του χειριστή.

Επιπλέον, υπάρχουν οι ρυθμίσεις της δεξαμενής ώστε με μία γενική εικόνα να έχουμε τον έλεγχο όλου του αντλιοστασίου. Ρυθμίζοντας την στάθμη μπορούμε να δούμε ακριβώς την συμπεριφορά των αντλιών με βάση τα όρια αυτοματισμού που έχουμε ορίσει. Επίσης, υπάρχει και η απεικόνιση της κατάστασης του αισθητηρίου (όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο) το οποίο όταν ξεπεράσει τα 4-20mA για οποιοδήποτε λόγο θα εμφανίσει σφάλμα. Σε αυτή την περίπτωση το πεδίο θα γίνει κόκκινο και θα γράψει «Analog Channel Error».



Εικόνα 46 Επεξήγηση αρχικής σελίδας

Ο προγραμματισμός της αρχικής σελίδας περιγράφεται παρακάτω:

Θα πρέπει αρχικά να αναφέρουμε πως κάθε μεταβλητή που θέλουμε να την επηρεάσουμε (είναι τύπου IN/OUT) θα πρέπει να την δηλώνουμε με την μορφή σχολίου `<!-- AWP_In_Variable Name="StationX_3_AI".AI[1].Scale_PV_Min' -->` όπως φαίνεται και παρακάτω. Εάν δεν τις αναφέρουμε τότε δεν θα μπορούμε να αλλάξουμε τιμή σε αυτή τη μεταβλητή. Δεν αναγράφουμε τις τιμές οι οποίες είναι μόνο απεικονιστικές και δεν επεμβαίνουμε.

```

1 <!-- AWP_In_Variable Name="Prepare_Signals_For_Scada".SIREN_OUTPUT' -->
2 <!-- AWP_In_Variable Name="StationX_3_AI".AI[1].Scale_PV_Min' -->
3 <!-- AWP_In_Variable Name="StationX_3_AI".AI[1].Scale_PV_Max' -->
4 <!-- AWP_In_Variable Name="StationX_3_AI".AI[1].HH_Sp' -->
5 <!-- AWP_In_Variable Name="StationX_3_AI".AI[1].H_Sp' -->
6 <!-- AWP_In_Variable Name="StationX_3_AI".AI[1].L_Sp' -->
7 <!-- AWP_In_Variable Name="StationX_3_AI".AI[1].LL_Sp' -->
8 <!-- AWP_In_Variable Name="StationX_3_AI".AI[1].Simulation_Value' -->
9 <!-- AWP_In_Variable Name="StationX_3_AI".AI[1].Simulation_On' -->
10 <!-- AWP_In_Variable Name="StationX_3_AI".AI[1].Set_Time_Sec1' -->
11 <!-- AWP_In_Variable Name="StationX_3_AI".AI[1].Set_Time_Sec2' -->
12 <!-- AWP_In_Variable Name="StationX_3_AI".AI[1].Set_Time_Sec3' -->
13 <!-- AWP_In_Variable Name="StationX_3_AI".AI[1].Set_Time_Sec4' -->

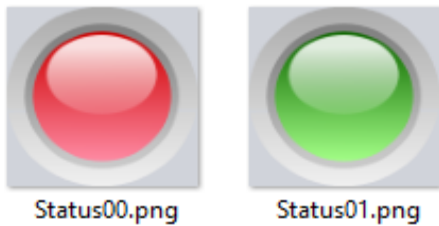
```

Εικόνα 47 Δήλωση μεταβλητών οι οποίες είναι IN/OUT

Η αρχή του προγραμματισμού ξεκινάει με την εντολή `<table width=»100px» height=»10px» border=0>` με το οποίο δηλώνουμε το μέγεθος του πίνακα, το ύψος του και το περίγραμμά του.

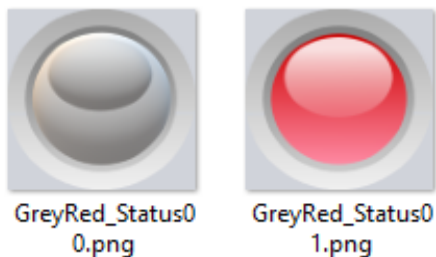
Εφόσον, ανοίξουμε τον πίνακα ξεκινάμε και ανοίγουμε μία γραμμή `<tr>` η οποία θα εμπεριέχει αυτά που θα γράψουμε μέχρι να κλείσει η γραμμή. Μέσα σε αυτή τη γραμμή με την εντολή `` γράφουμε κείμενο με συγκεκριμένο γραμματοσειρά όπως αναφέρουμε και παραπάνω. Έπειτα κλείνουμε την γραμμή με την εντολή `</tr>`. Συνεπώς, αυτό που θα εμφανιστεί στην οθόνη είναι το κείμενο «ΓΕΝΙΚΑ ΣΗΜΑΤΑ». Για να εμφανίσουμε όλα τα

σήματα σε σειρά αρκεί να ανοίξουμε ένα `<table>` με μήκος 500px ώστε να χωρέσουν όλες οι πληροφορίες όπως φαίνεται στην παρακάτω εικόνα. Μέσα σε αυτό το table γράφουμε πρώτα το όνομα της μεταβλητής και από δίπλα δηλώνουμε την ονομασία όπως ακριβώς είναι στο datablock. Είναι πολύ σημαντικό να είναι ακριβώς ίδιο το όνομα αλλιώς δεν θα δουλεύει σωστά η μεταβλητή. Η δήλωση του χρώματος του ψηφιακού σήματος γίνεται με την εντολή `src='Images\Status0:="Prepare_Signals_For_Scada".EMERGENCY_STOP:.png' height="20px" width="20px"/>`. Όπως είναι γνωστό δηλώνουμε το ύψος και το μήκος του κύκλου για να έχει ένα σωστό μέγεθος. Η εικόνα που φορτώνουμε βρίσκεται στον φάκελο Images και ονομάζεται Status0. Εδώ είναι σημαντικό να δώσουμε έμφαση στο πώς παίρνει την τιμή η μεταβλητή. Παρακάτω εμφανίζεται η εικόνα που φορτώνεται μέσα στον φάκελο.



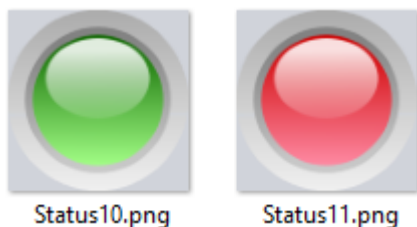
Εικόνα 48 Ψηφιακό σήμα.
Στο 0 έχει σφάλμα, στο 1 είναι οκ

Το δεύτερο ψηφίο όπως φαίνεται στην εικόνα αριστερά είναι αυτό που διαβάζει από το PLC. Δηλαδή, φορτώνεται η εικόνα `Images\ Status0` η οποία στο 0 είναι κόκκινο και στο 1 είναι πράσινο. Όταν το σήμα θα στέλνει 0 θα κουμπωθεί δίπλα από το 0 το 0 και θα φορτώσει η φωτογραφία με τον κόκκινο κύκλο. Θα είναι σαν να γράφει `Images\ Status00`. Όταν το σήμα θα στέλνει 1 θα κουμπωθεί δίπλα από το 0 το 1 και θα φορτώσει η φωτογραφία με τον πράσινο κύκλο. Θα είναι σαν να γράφει `Images\ Status01`.



Εικόνα 49 Ψηφιακό σήμα. Στο 0 είναι γκρι, στο 1 κόκκινο

Μία άλλη περίπτωση όπως αυτή της σειρήνας είναι να δηλωθεί με το όνομα `Images\GreyRed_Status0` το οποίο σημαίνει πως όταν το σήμα θα είναι 0 είναι γκρι και όταν γίνει 1 θα είναι κόκκινο. Όταν το σήμα θα στέλνει 0 θα κουμπωθεί δίπλα από το 0 το 0 και θα φορτώσει η φωτογραφία με τον γκρι κύκλο. Θα είναι σαν να γράφει `Images\ GreyRed_Status00`. Όταν το σήμα θα στέλνει 1 θα κουμπωθεί δίπλα από το 0 το 1 και θα φορτώσει η φωτογραφία με τον κόκκινο κύκλο. Θα είναι σαν να γράφει `Images\ GreyRed_Status01`.

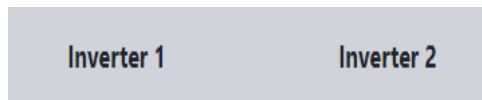


Εικόνα 50 Ψηφιακό σήμα. Στο 0 πράσινο, στο 1 κόκκινο

Τέλος, υπάρχει και η περίπτωση όπου ένα σήμα να είναι πράσινο στο 0 και στο 1 να γίνεται κόκκινο. Σε αυτή την περίπτωση θα πρέπει να δηλωθεί `Images\ Status1`. Όταν το σήμα θα στέλνει 0 θα κουμπωθεί δίπλα από το 1 το 0 και θα φορτώσει η φωτογραφία με τον πράσινο κύκλο. Θα είναι σαν να γράφει `Images\ Status10`. Όταν το σήμα θα στέλνει 1 θα κουμπωθεί δίπλα από το 1 το 1 και θα φορτώσει η φωτογραφία με τον κόκκινο κύκλο. Θα είναι σαν να γράφει `Images\ Status11`.

Για την απεικόνιση των inverter δημιουργούμε ένα μεγάλο table μήκους 404px και ύψους 35px και έπειτα μία γραμμή στην οποία θα ανήκουν τα δύο table που δημιουργούμε με την ονομασία της αντλίας «Inverter 1» στο ένα table και της αντλίας «Inverter 2» στο άλλο table και έπειτα κλείνουμε την γραμμή. Σε αυτή την περίπτωση που θέλουμε να έχουμε δύο Inverter και πληροφορίες η μία δίπλα στην άλλη πρέπει να τα δημιουργούμε ανά γραμμή γιατί εφόσον ανοίξουμε ένα table δεν μπορούμε να πάμε να δημιουργήσουμε ένα table στα δεξιά του. Ο κώδικας δηλαδή γράφεται ανά γραμμή.

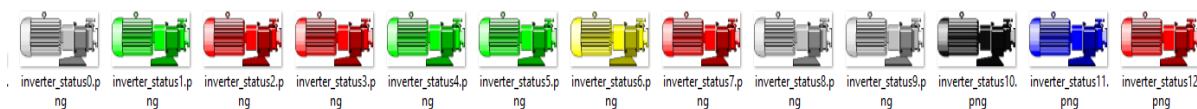
```
<table width="404px" height="35px" border=0>
<tr>
<td width="202px" align="center" height="35px">
<b>Inverter 1</b>
</td>
<td width="202px" align="center" height="35px">
<b>Inverter 2</b>
</td>
</tr>
```



Εικόνα 54 Ονομασίες Inverter στο web

Εικόνα 53 Ονομασίες Inverter

Έπειτα ανοίγουμε ένα άλλο table με μήκος πάλι 404px και ύψος 60px και μία γραμμή. Αυτός είναι ένας πίνακας στον οποίο θέλουμε να συμπεριλάβουμε δύο φωτογραφίες inverter. Πρέπει λοιπόν να δημιουργήσουμε δύο ακόμα table ένα για κάθε φωτογραφία. Ανοίγουμε λοιπόν ένα table με μήκος 202px και ύψος 50px στο οποίο φορτώνουμε την εικόνα του inverter με μήκος 80px και ύψος 46px. Με την εντολή `` φορτώνουμε την εικόνα για το μοτέρ. Διαβάζουμε την τιμή της μεταβλητής Moter_Status από το data block του inverter και ανάλογα με την τιμή του φορτώνεται και το αντίστοιχο χρώμα. Όπως αναφέρθηκε παραπάνω με τα ψηφιακά σήματα το ίδιο συμβαίνει και σε αυτή την περίπτωση απλώς οι τιμές δεν είναι 0 και 1 αλλά από 0-12.



Εικόνα 55 Χρωματική απεικόνιση των αντλιών

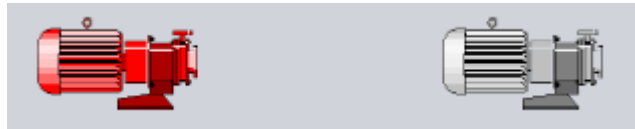
Η εικόνα που φορτώνεται αναφέρεται στην εντολή `src='Images\Inverter\inverter_status` συνεπώς εάν η μεταβλητή έχει τιμή ένα θα φορτώνει την εικόνα `src='Images\Inverter\inverter_status1` δηλαδή πράσινο χρώμα. Εάν έχει την τιμή 12 `src='Images\Inverter\inverter_status12` δηλαδή emergency stop και συνεπώς μπλε χρώμα.

```
<table width="404px" height="60px" border=0>
<tr>
<td width="202px" align="center" height="50px">

</td>
<td width="202px" align="center" height="50px">

</td>
</tr>
</table>
```

Εικόνα 56 Απεικόνιση των αντλιών



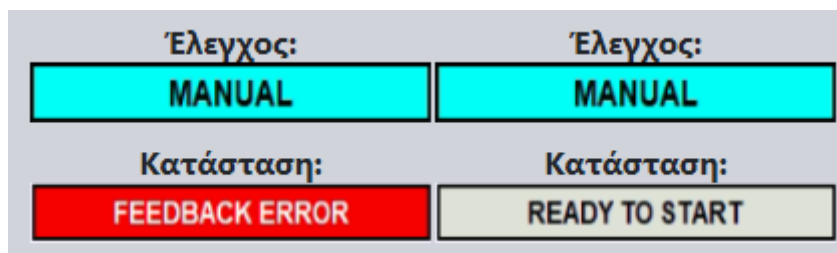
Εικόνα 57 Απεικόνιση αντλιών στο web

Το ίδιο κάνουμε και για τα κείμενα του ελέγχου και της κατάστασης τους. Ανοίγουμε ένα table μήκους 300px μέσα στο οποίο υπάρχουν άλλα δύο table για τον έλεγχο της πρώτης αντλίας και δεξιά της δεύτερης αντλίας.

```
<table width="300px" height="60px" border=0>
<tr>
<td width="50px" align="center" height="60px">
<b>Έλεγχος:</b> 
</td>
<td width="50px" align="center" height="60px">
<b>Έλεγχος:</b> 
</td>
</tr>
</table>

<table width="300px" height="60px" border=0>
<tr>
<td width="50px" align="center" height="60px">
<b>Κατάσταση:</b> 
</td>
<td width="50px" align="center" height="60px">
<b>Κατάσταση:</b> 
</td>
</tr>
</table>
```

Εικόνα 58 Απεικόνιση ελέγχου και κατάστασης αντλιών



Εικόνα 59 Απεικόνιση ελέγχου και κατάστασης αντλιών στο web


```

':="StationX_3_AI".AI[1].Scale_PV_Max:' <font size="4" color="black" >&nbsp;
</font>
        </form>
    </td>
</tr>
<tr>
    <td width="300px" align="left" height="50px">
        <form method="post">
            <font size="4" color="black" >&nbsp;PV
Min:&nbsp;</font> <input type="number2" font size="2"
name='"StationX_3_AI".AI[1].Scale_PV_Min' value =
':="StationX_3_AI".AI[1].Scale_PV_Min:' <font size="4" color="black" >&nbsp;
</font>
        </form>
    </td>
</tr>
<table width="300px" height="100px" border=0>
    <tr>
        <td width="300px" align="left" height="30px">
            <form method="post">
                <font size="4" color="black" >&nbsp;Simulation
Value:</font> <input type="number2" font size="2"
name='"StationX_3_AI".AI[1].Simulation_Value' value =
':="StationX_3_AI".AI[1].Simulation_Value:' >
            </form>
        </td>
    </tr>
    <tr>
        <td width="300px" align="left" height="30px">
            <form method="post" action="">
                <input type="submit" id="simon"
value="Simulation On"> <img id="plantStatus"
src='Images\Status0:="StationX_3_AI".AI[1].Simulation_On:.png' height="20px"
width="20px"/>
                <input type="hidden"
name='"StationX_3_AI".AI[1].Simulation_On' value="1">
            </form>
        </td>
    </tr>
    <tr>
        <td width="300px" align="left" height="30px">
            <form method="post" action="">
                <input type="submit" value="Simulation Off">
<img id="plantStatus"
src='Images\Status1:="StationX_3_AI".AI[1].Simulation_On:.png' height="20px"
width="20px"/>
                <input type="hidden"
name='"StationX_3_AI".AI[1].Simulation_On' value="0">
            </form>
        </td>
    </tr>
</tr>
<td width="500px" align="left" height="80px">
    
```

```
                <font size="4" color="black"
>&nbsp;&nbsp;&nbsp;Channel:</font> <img
src='Images\Sensor\Channel:="PLA_Analog_Read_V02_01".Channel_Status:.png'
width="200px" height="25px">
                </td>
            </tr>
        </table>
```

Ο κώδικας της στάθμης αναλύεται παρακάτω.

- **Σελίδα στάθμης**

Πατώντας την σελίδα «Στάθμη» από το μενού αριστερά **Στάθμη**, μας ανοίγει η παρακάτω εικόνα στην οποία συμπεριλαμβάνονται όλες οι απαραίτητες πληροφορίες του αναλογικού αισθητηρίου της στάθμης.

Ρυθμίσεις Δεξαμενής

PV: 0 m

PV Max: m

PV Min: m

HH Sp: m

H Sp: m

L Sp: m

LL Sp: m

Simulation Value:

Simulation On

Simulation Off

Channel: **Analog Channel Ok**

Filter HH Alarm: sec

Filter H Alarm: sec

Filter L Alarm: sec

Filter LL Alarm: sec

Εικόνα 61 Σελίδα στάθμης web

Σε αυτή τη σελίδα αναφέρονται οι μεταβλητές της στάθμης. Αρχικά, δηλώνουμε μία επικεφαλίδα για τη σελίδα, ώστε, να γνωρίζει ο χειριστής σε ποιο αναλογικό αισθητήριο βρίσκεται (στην περίπτωση που είχαμε περισσότερα από ένα αισθητήρια). Στην τιμή «**PV**» (Present Value) φαίνεται πόσα μέτρα έχει μέσα η στάθμη για να γνωρίζουμε εάν είναι άδεια ή γεμάτη αλλά και για να ελέγχουμε τον αυτοματισμό των αντλιών ο οποίος θα αναλυθεί στις σελίδες των αντλιών. Στα πεδία «**PV Max**» και «**PV Min**» ρυθμίζουμε το μέγιστο και το ελάχιστο που μετράει το αισθητήριο. Σε αυτή τη περίπτωση, το αισθητήριο μετράει από 0-6m. Συνεπώς, θα γράψουμε στην τιμή PV Min το 0 και στην τιμή PV Max το 6. Τα τέσσερα πεδία «**HH Sp**», «**H Sp**», «**L Sp**» και «**LL Sp**» ορίζουν τα όρια που θα εμφανιστούν τα σφάλματα όπως περιγράφονται παρακάτω:

- **HH Sp (High-High Set Point)**: Σημαίνει ότι μόλις η στάθμη ξεπεράσει αυτό το όριο τότε το λαμπάκι δεξιά θα γίνει κόκκινο και θα εμφανιστεί ο φάρος διότι υπάρχει σφάλμα. (Το σφάλμα αυτό απεικονίζεται και στο scada και με λεκτικό κείμενο το οποίο αναγράφει «**Πάρα πολύ υψηλή στάθμη**»). Στο scada το λαμπάκι σε αυτή την περίπτωση αναβοσβήνει για να επισημάνει την σοβαρότητα του σφάλματος.
- **H Sp (High Set Point)**: Σημαίνει ότι μόλις η στάθμη ξεπεράσει αυτό το όριο τότε το λαμπάκι δεξιά θα γίνει κόκκινο και θα εμφανιστεί ο φάρος διότι υπάρχει σφάλμα. (Το σφάλμα αυτό απεικονίζεται και στο scada και με λεκτικό κείμενο το οποίο αναγράφει «**Πολύ υψηλή στάθμη**»). Το λαμπάκι θα ανάψει και στην περίπτωση που ξεπεραστεί το HH Sp διότι εφόσον έχει πάρα πολύ υψηλή στάθμη τότε υπάρχει και σφάλμα υψηλής στάθμης.
- **L Sp (Low Set Point)**: Σημαίνει ότι μόλις η στάθμη πέσει κάτω από αυτό το όριο τότε το λαμπάκι δεξιά θα γίνει κίτρινο και θα εμφανιστεί ο φάρος διότι υπάρχει σφάλμα. (Το σφάλμα αυτό απεικονίζεται και στο scada και με λεκτικό κείμενο το οποίο αναγράφει «**Χαμηλή στάθμη**»). Το λαμπάκι θα ανάψει και στην περίπτωση που πέσει κάτω από το LL Sp διότι εφόσον έχει πάρα πολύ χαμηλή στάθμη τότε υπάρχει και σφάλμα χαμηλής στάθμης.
- **LL Sp (Low-Low Set Point)**: Σημαίνει ότι μόλις η στάθμη πέσει κάτω από αυτό το όριο τότε το λαμπάκι δεξιά θα γίνει κίτρινο και θα εμφανιστεί ο φάρος διότι υπάρχει σφάλμα. (Το σφάλμα αυτό απεικονίζεται και στο scada και με λεκτικό κείμενο το οποίο αναγράφει «**Πολύ χαμηλή στάθμη**»). Στο scada το λαμπάκι σε αυτή την περίπτωση αναβοσβήνει για να επισημάνει την σοβαρότητα του σφάλματος.

Στο πεδίο «**Simulation Value**» καθορίζουμε την τιμή σε περίπτωση που θέλουμε να κάνουμε προσομοίωση τα μέτρα της στάθμης για κάποιον έλεγχο. Για να μπορέσει όμως η τιμή αυτή να περάσει στο «**PV**» και να γίνει η προσομοίωση θα πρέπει να επιλέξουμε το κουμπί «**Simulation On**». Αυτό μπορεί να γίνει στην περίπτωση που έχει καεί το αισθητήριο και θέλουμε να ξεκινήσουμε την αντλία. Για να επαναφέρουμε την τιμή σε αυτή που δείχνει το αισθητήριο θα πρέπει να κάνουμε κλικ στο κουμπί «**Simulation Off**». Σε κάθε κουμπί υπάρχει δίπλα ένα λαμπάκι. Εάν έχουμε ενεργοποιήσει το Simulation On τότε θα ανάψει το λαμπάκι δίπλα από το κουμπί πράσινο και το λαμπάκι δίπλα από το κουμπί του Simulation Off θα γίνει κόκκινο αντίστοιχα.

Δίπλα από το κείμενο «**Channel**» αναγράφεται η κατάσταση του οργάνου. Δηλαδή, όπως έχουμε προαναφέρει και σε προηγούμενη ενότητα όταν το αισθητήριο είναι σωστά συνδεδεμένο και παίρνει σωστή τιμή το PLC τότε γράφει «**Analog Channel Ok**» και έχει πράσινο χρώμα. Στην περίπτωση που αυτό δεν ισχύει τότε γράφει «**Analog Channel Error**» και έχει κόκκινο χρώμα.

Στα πεδία «**Filter HH Alarm**», «**Filter H Alarm**», «**Filter L Alarm**» και «**Filter LL Alarm**» αναφέρεται ο χρόνος φίλτρου για να εμφανιστεί το σφάλμα. Για να εμφανιστεί το σφάλμα «Πάρα πολύ υψηλή στάθμη» και να κοκκινήσει το λαμπάκι του HH Sp πρέπει η στάθμη να ξεπεράσει το όριο που γράψαμε στο πεδίο «HH Sp» και να συνεχίσει να είναι παραπάνω για 10 sec (τον χρόνο που ορίσαμε σε αυτό το πεδίο). Για να εμφανιστεί το σφάλμα «Υψηλή στάθμη» και να κοκκινήσει το λαμπάκι του H Sp και HH Sp πρέπει η στάθμη να είναι μεγαλύτερη από το όριο που γράψαμε στο «H Sp» και να συνεχίσει να είναι μεγαλύτερη για 10 sec. Το ίδιο ισχύει και για τα χαμηλά όρια μόνο που εκεί η στάθμη πρέπει να είναι μικρότερη από το Sp για τον χρόνο που έχει οριστεί.

Ο προγραμματισμός για την σελίδα της στάθμης αναλύεται παρακάτω:

Όπως έχουμε ξανά αναφέρει στην αρχή του κώδικα δηλώνουμε τις μεταβλητές τις οποίες θέλουμε να παραμετροποιήσουμε. Σε αυτή τη σελίδα όπως εξηγήσαμε παραμετροποιούμε τα πεδία «**PV Max**», «**PV Min**», «**HH Sp**», «**H Sp**», «**L Sp**», «**LL Sp**», «**Simulation Value**», «**Simulation On**» (χρησιμοποιείται το ίδιο bit και για το Simulation Off), «**Filter HH Alarm**», «**Filter H Alarm**», «**Filter L Alarm**», «**Filter LL Alarm**».

```

1 <!-- AWP_In_Variable Name='\"StationX_3_AI\".AI[1].Scale_PV_Min' -->
2 <!-- AWP_In_Variable Name='\"StationX_3_AI\".AI[1].Scale_PV_Max' -->
3 <!-- AWP_In_Variable Name='\"StationX_3_AI\".AI[1].HH_SP' -->
4 <!-- AWP_In_Variable Name='\"StationX_3_AI\".AI[1].H_Sp' -->
5 <!-- AWP_In_Variable Name='\"StationX_3_AI\".AI[1].L_Sp' -->
6 <!-- AWP_In_Variable Name='\"StationX_3_AI\".AI[1].LL_Sp' -->
7 <!-- AWP_In_Variable Name='\"StationX_3_AI\".AI[1].Simulation_Value' -->
8 <!-- AWP_In_Variable Name='\"StationX_3_AI\".AI[1].Simulation_On' -->
9 <!-- AWP_In_Variable Name='\"StationX_3_AI\".AI[1].Set_Time_Sec1' -->
10 <!-- AWP_In_Variable Name='\"StationX_3_AI\".AI[1].Set_Time_Sec2' -->
11 <!-- AWP_In_Variable Name='\"StationX_3_AI\".AI[1].Set_Time_Sec3' -->
12 <!-- AWP_In_Variable Name='\"StationX_3_AI\".AI[1].Set_Time_Sec4' -->

```

Εικόνα 62 Δήλωση παραμετροποιήσιμων μεταβλητών

Όπως έχουμε αναφέρει και σε προηγούμενη ενότητα το κομμάτι του προγραμματισμού που φαίνεται παρακάτω είναι κοινό σε όλες τις σελίδες με τη διαφορά πως αλλάζουμε της σελίδα που φορτώνεται στην ανανέωση.

```

14 <style>
15 .Siren{
16 background-image: url("Images/errors.png");
17 background-color: transparent;
18 background-repeat: no-repeat;
19 background-position: 0px 0px;
20 height: 80px
21 box-shadow:inset 0px 0px 5px 0px #91b8b3;
22 display:inline-block;
23 cursor:pointer;
24 padding:30px 30px;
25 z-index: 12;
26 position:absolute;
27 top:15%;
28 left:85%;
29 width: 15px;
30 height: 15px;
31 }
32 .Siren:active {
33 | transform: translateY(1px);
34 }
35 </style>
36
37 <!DOCTYPE html>
38 <html>
39
40 <head>
41 <title>Userdefined Website - Application Start</title>
42 <meta charset="UTF-8" >
43 <meta http-equiv="refresh" content="10; URL=Sensor1.html"> <!-- ΔΗΛΩΣΗ ΤΗΣ ΤΡΕΧΟΥΣΑΣ ΣΕΛΙΔΑΣ ΓΙΑ ΑΝΑΝΕΩΣΗ -->
44 <link type="text/css" rel="stylesheet" href="Stylesheet/siemens_Stylesheet.css"/>
45 <link type="text/css" rel="stylesheet" href="Stylesheet/bootstrap.min.css">
46 <script src="js/jquery-3.6.0.min.js"></script>
47 </head>
48
49 <body>

```

Εικόνα 63 Κοινός κώδικας με τη διαφορά της φόρτωσης της σελίδας

Στις δύο εικόνες παρακάτω το κομμάτι του προγραμματισμού είναι κοινό σε κάθε σελίδα και έχει επεξηγηθεί παραπάνω.

```

53 <!-- EPIKEFALIDA START -->
54 <div id="header">
55 <table border="0" >
56 <tr>
57
58 <td width="500px"></td>
59 <td width="500px"></td>
60 <td width="450px"> <h2>Στάθμη := "PLA_Analog_Read_V02_01".Pv: m</h2></td>
61 <td width="650px"><h2>Στάθμη</h2></td>
62 <td width="400px"><h2>ΤΣΕ 1 - ΑΝΑΤΙΟΣΤΑΣΙΟ ΛΥΜΑΤΩΝ</h2></td>
63 </tr>
64 </table>
65 </div>
66 <!-- EPIKEFALIDA END-->

```

Εικόνα 64 Κοινός κώδικας - Επικεφαλίδα

```

<script>
setInterval(function() {
  if(:"Pla_Enable_Siren_V00_DB1".Signal_on: || :="Pla_Enable_Siren_V00_DB2".Signal_on:
  | document.getElementById("sir").style.visibility = 'visible';}
  else if(:"Pla_Enable_Siren_V00_DB1".Signal_on: == 0 && :="Pla_Enable_Siren_V00_DB2".S
  | document.getElementById("sir").style.visibility = 'hidden';}
  if($('.Siren').css('background-image').indexOf('errors2')>-1){
  | $('.Siren').css('background-image',"url('Images/errors.png')");} else {
  | $('.Siren').css('background-image',"url('Images/errors2.png')");}
  },1000);
</script>

```

Εικόνα 65 Κοινό κώδικας για την εμφάνιση σειρήνας

Η παρακάτω εικόνα είναι επίσης κοινή με την διαφορά ότι σε όποια σελίδα ανήκει τότε αναφέρεται σαν κείμενο (για να μην μπορεί να πατήσει κάποιος και να φορτώσει την ίδια σελίδα ξανά). Με την εντολή `<div class="containerNavi">` φορτώνονται τα στοιχεία που έχουμε δηλωθεί στην καρτέλα containerNavi μέσα στο αρχείο SiemensStylesheet όπως φαίνεται στην επόμενη εικόνα.

Είναι σημαντικό να κρατείται η ίδια σειρά γιατί σε κάθε σελίδα φορτώνεται αυτό το κομμάτι. Για παράδειγμα, εάν στην σελίδα στάθμης γράψουμε επάνω το Inverter2.html τότε στο μενού πλοήγησης αριστερά σε αυτή τη σελίδα θα αναγράφεται πρώτα το κουμπί «Αντλητικό Συγκρότημα 2» και από κάτω το «Αντλητικό Συγκρότημα 1». Για την φόρτωση της σελίδας χρησιμοποιείται η εντολή `Αρχική Σελίδα
`. Η εντολή `<a>` δηλώνει συγκεκριμένο μέγεθος γραμματοσειρά, η εντολή `href` κάνει αναφορά στην σελίδα που θέλουμε να φορτώσουμε και η `Αρχική Σελίδα` παίρνει το μέγεθος γραμμμάτων που δηλώνει το `<a>`. Με την εντολή `
` αφήνουμε ένα μεγάλο κενό για την επόμενη γραμμή που θα περάσει από κάτω.

```

84 <!-- Content Area -->
85 | | <div id="content">
86 <!-- MENU SELIDES START-->
87 | | | <div class="containerNavi">
88 | | | | <ul>
89 | | | | | <a> ----- ΜΕΝΟΥ -----</a><br><br>
90 | | | | | <ul>
91 | | | | | | <a>Στάθμη</a><br>
92 | | | | | | <br>
93 | | | | | | <a href="Inverter1.html">Αντλητικό Συγκρότημα 1</a><br>
94 | | | | | | <br>
95 | | | | | | <a href="Inverter2.html">Αντλητικό Συγκρότημα 2</a><br>
96 | | | | | | <br>
97 | | | | | | <a href="Start.html">Αρχική Σελίδα</a><br>
98 | | | | | </ul>
99 | | | | </ul>
100 | | | </div>
101 | | </div>
102 | </div>
103 <!-- MENU SELIDES END-->
104

```

Εικόνα 66 Επεξήγηση μενού

```
#content div.containerNavi {
    POSITION: absolute;
    left: 0;
    top: 0;
    width: 198px;
    bottom: 0;
    padding-top: 50px;
    padding-left: 3px;
    text-align: left;
    border-color: white;
    border-style: solid;
    border-width: 1px;
    background-color: rgb(148,158,170);
    border-collapse : separate;
    z-index: 1;
}
```

Το κομμάτι του κώδικα που φαίνεται στην εικόνα αριστερά βρίσκεται στο αρχείο `siemens_Stylesheet`. Μπορούμε να το επεξεργαστούμε και να αλλάξου για παράδειγμα το χρώμα. Αυτή η αλλαγή είναι γενική και θα επηρεάσει το αριστερό μενού σε όλες τις σελίδες.

Εικόνα 67 ContainerNavi από το `siemens_Stylesheet`

Παρακάτω αναγράφεται ο βασικός προγραμματισμός για τις ρυθμίσεις της στάθμης. Ξεκινάμε δημιουργώντας ένα table μεγάλο το οποίο να περιλαμβάνει μέχρι και το LL Sp. Γι' αυτό στο height του πίνακα δίνουμε μία μεγάλη τιμή για να μας χωρέσει όλα τα πεδία ως προς το ύψος. Με την εντολή **<h2>** ορίζουμε bold γράμματα για την επικεφαλίδα. Έπειτα αφήνουμε μία σειρά κενή **<tr>** και ανοίγουμε έναν άλλο πίνακα μέσα στον οποίο θα αναφέρουμε μόνο την τρέχουσα τιμή της μονάδας. Επειδή η τιμή είναι μόνο ενδεικτική χρειάζεται απλά να δηλώσουμε μέγεθος γραμμάτων με την εντολή **** και να γράψουμε το όνομα του data block και την τιμή όπως ακριβώς αναφέρεται **:= "PLA_Analog_Read_V02_01".Pv: .** Όταν ανοίγουμε μέγεθος γραμμάτων πρέπει μετά και να τα κλείνουμε ****. Ότι υπάρχει μέσα στο font size θα έχει το συγκεκριμένο μέγεθος. Εφόσον τελειώσαμε με την γραμμή του PV πρέπει να κλείσουμε την γραμμή και τον πίνακα με τις εντολές **</tr>** και **</td>** αντίστοιχα. Στη συνέχεια, ανοίγουμε έναν άλλο πίνακα καθώς και μία γραμμή για να δείξουμε την τιμή του PV Max. Με την εντολή ** PV Max: <input type="number2" font size="2" name="StationX_3_AI".AI[1].Scale_PV_Max' value = ':="StationX_3_AI".AI[1].Scale_PV_Max:' m ** δημιουργούμε την γραμμή αυτή. Το πεδίο είναι παραμετροποιήσιμο επομένως πρέπει να χρησιμοποιηθεί η εντολή **<input type>** το **number2** το δημιουργήσαμε εμείς μέσα στο `siemens_Stylesheet` για να έχει την γραμματοσειρά που θέλαμε. Οι εντολές ** ** και ** ** Δηλώσουμε συγκεκριμένες θέσει κενός (αναγράφονται στη λίστα εντολών) και χρησιμοποιείται για την σωστή στοίχιση των λέξεων. Το ίδιο γίνεται και σε όλα τα άλλα πεδία στα οποία και γράφουμε και διαβάζουμε. Για τον προγραμματισμό του κουμπιού χρησιμοποιείται η εντολή **<input type="hidden" name="StationX_3_AI".AI[1].Simulation_On' value="1">** . Το input type δηλώνει ότι η μεταβλητή θα μπορεί να παραμετροποιηθεί. Στην περίπτωση που έχουμε κουμπί γράφουμε **hidden** και στην περίπτωση που έχουμε πεδίο γράφουμε **number**.

• Σελίδα Αντλητικού Συγκροτήματος 1 και 2

Η σελίδα θα εξηγηθεί για το ένα inverter καθώς ο κώδικας είναι ο ίδιος απλώς αλλάζει το νούμερο από Inverter 1 σε 2 καθώς και η σελίδα φόρτωσης. Η εικόνα που ανοίγει όταν πατάμε στο κουμπί «Αντλητικό Συγκρότημα 1 ή 2» φαίνεται παρακάτω.

The screenshot displays a control interface for a pump assembly. At the top left is a motor icon. To its right, the control mode is set to 'MANUAL' (highlighted in cyan) and the status is 'WITHOUT CONTROL' (highlighted in black). Below these are three rows of input fields: 'Feedback time: 120 sec', 'Working Hour: 0 Ωρες', 'Safe start time: 0 sec', 'Frequency : Hz', and 'Safe stop time: 0 sec', 'Ampere : Amp'. There are two rows of buttons: 'Manual' and 'Auto' (selected), and 'Start' and 'Stop'. Below these is an 'Ack' button. Further down are four rows of speed and level settings: 'Feedback Speed: 0 Hz', 'Level Start Sp: 2.2 m', 'Manual Speed: 50 Hz', 'Level Stop Sp: 0.2 m', 'Min Speed: 35 Hz', and 'Max Speed: 50 Hz'. At the bottom, there are four status indicators: 'Ξηρά λειτουργία:' (green dot), 'Φλοτέρ στάθμης εκκίνησης:' (green dot), 'Εντολή από στάθμη:' (red dot), and 'Εντολή από φλοτέρ:' (red dot).

Εικόνα 69 Σελίδα αντλητικού συγκροτήματος στο web

Από αυτή τη σελίδα γίνονται όλες οι ρυθμίσεις που αφορούν την εκάστοτε αντλία. Επάνω αριστερά απεικονίζεται με χρώμα η κατάσταση της αντλίας (επεξηγήθηκε σε προηγούμενη ενότητα) καθώς επίσης, και με κείμενο με το όνομα «Κατάσταση» το οποίο παίρνει επίσης το αντίστοιχο χρώμα ανάλογα με τις συνθήκες λειτουργίας. Στο κείμενο «Έλεγχος» αναγράφεται εάν η αντλία είναι σε τοπικό χειρισμό (επάνω από τον πίνακα και εκκίνηση της αντλίας με κουμπί πίνακος), αυτόματο χειρισμό (ακολουθεί δηλαδή τον αυτοματισμό – πρέπει να είναι ο επιλογικός διακόπτης σε θέση remote), χειροκίνητο έλεγχο (έλεγχο με start – stop από το web – πρέπει να είναι ο επιλογικός διακόπτης σε θέση remote) και εκτός ελέγχου (έχει ρυθμιστεί ο επιλογικός διακόπτης επάνω από τον πίνακα σε κατάσταση 0). Ανάλογα με την θέση αντιστοιχεί και ένα χρώμα. Τρεις βασικοί παράμετροι που ρυθμίζουμε στα inverter είναι το «Feedback Time», «Safe start time» και το «Safe stop time».

- Feedback time: Είναι ο χρόνος ο οποίος εφόσον έχει εντολή το inverter να ξεκινήσει και δεν μου έρθει η είσοδος της επιβεβαίωσης τότε μετά από αυτόν το χρόνο να κόψει η εντολή εκκίνησης και να βγάλει σφάλμα επιβεβαίωσης.

- Safe start time: Είναι ο χρόνος που μετράει, από τη στιγμή που σταματήσει η αντλία ώστε να μην μπορέσει να ξανά ξεκινήσει γρήγορα.
- Safe stop time: Εφόσον ξεκινήσει η αντλία αλλά δεν ολοκληρωθεί σωστά η εκκίνηση (εσφαλμένη κίνηση) και δηλώνεται επίσης από το χρόνο safe start να μετρήσει ο χρόνος stop για να μπορέσει να ξανά ξεκινήσει. Στον πίνακα στα δεξιά φαίνονται οι ώρες λειτουργίας της αντλίας, τη συχνότητα στην οποία λειτουργεί και το ρεύμα (Ampere) που τραβάει κατά την λειτουργία της. Από κάτω υπάρχουν τα κουμπιά για την λειτουργία της αντλίας. Υπάρχει το κουμπί «**Auto**» (εφόσον ο επιλογικός διακόπτης πάνω στον πίνακα είναι γυρισμένος σε remote) με το οποίο ενεργοποιείται ο αυτοματισμός της αντλίας ο οποίος λειτουργεί είτε με την αναλογική στάθμη είτε με τα φλοτέρ. Για να μπορέσει να ξεκινήσει η αντλία θα πρέπει είτε η στάθμη να είναι πάνω από 2.2 είτε να έχουμε το φλοτέρ της εκκίνησης. Σε αυτή την περίπτωση, επειδή η αντλία θα ξεκινήσει να δουλεύει και να πέφτει η στάθμη κάνουμε μανδάλωση και στο όριο start αλλά και στο φλοτέρ εκκίνησης. Για να κόψει η αντλία θα πρέπει η στάθμη να πέσει κάτω από 0.2 ή να χάσουμε το φλοτέρ της ξηράς λειτουργίας. Εφόσον γίνει αυτό η αντλία δεν πρόκειται να πάρει σε αυτόματη λειτουργία. Εάν έχουμε εντολή από στάθμη πρασινίζει το λαμπάκι δίπλα από το κείμενο «Εντολή από στάθμη». Εάν έχουμε εντολή από φλοτέρ τότε πρασινίζει το λαμπάκι που είναι δίπλα από το κείμενο «Εντολή από φλοτέρ». Υπάρχουν δύο έλεγχοι (αναλογικό αισθητήριο και ψηφιακά φλοτέρ) ώστε σε περίπτωση που καεί το αναλογικό αισθητήριο να μπορεί η αντλία να ξεκινάει με την ένδειξη των φλοτέρ μέχρι να αποκατασταθεί η βλάβη. Εάν έχει καεί η είσοδος στο plc που διαβάζει τα φλοτέρ να μπορεί η αντλία να ξεκινήσει με το αναλογικό αισθητήριο. Αυτές οι δύο περιπτώσεις έχουν προγραμματιστεί με OR και όχι με AND. Υπάρχει το κουμπί «**Manual**» το οποίο ακυρώνει τον αυτοματισμό και μπορεί ο χειριστής να κάνει μια δοκιμή την αντλία σε όποια περίπτωση θέλει. Υπάρχει το κουμπί «**Start**» το οποίο εκκινεί την αντλία μόνο στην περίπτωση που είναι γυρισμένη σε Manual λειτουργία και το κουμπί «**Stop**» το οποίο την σταματάει μόνο εφόσον είναι γυρισμένη σε Manual λειτουργία. Υπάρχει και το κουμπί «**Ack**» που σημαίνει acknowledge και χρησιμοποιείται για την αναγνώριση ενός σφάλματος. Στο πρόγραμμα έχει ρυθμιστεί και η αυτόματη αναγνώριση σφάλματος για την διευκόλυνση των χειριστών. Υπάρχουν άλλα τρία πεδία τα οποία καθορίζουν την ταχύτητα της αντλίας. Το ένα είναι το «**Manual Speed**» το οποίο ρυθμίζει την ταχύτητα της αντλίας και σε χειροκίνητη λειτουργία αλλά και σε αυτόματη. Δηλαδή, όταν λειτουργεί η αντλία ακολουθεί την ταχύτητα που είναι δηλωμένη σε αυτό το πεδίο.

Υπάρχει το «**Min Speed**» και το «**Manual Speed**» από τα οποία ρυθμίζουμε την ελάχιστη συχνότητα που θέλουμε να δουλεύει η αντλία όπως για παράδειγμα τα 30Hz διότι κάτω από 30Hz μπορεί αυτή να καεί και ρυθμίζουμε και την μέγιστη ταχύτητα που είναι τα 50Hz. Ο χειριστής μπορεί να αλλάξει αυτά τα όρια εφόσον είναι απαραίτητο.

Ο προγραμματισμός για τις αντλίες περιγράφεται παρακάτω:

Όπως έχουμε εξηγήσει στο επάνω μέρος της κάθε σελίδα δηλώνουμε της μεταβλητές που θέλουμε να μπορούμε να παραμετροποιούμε. Αναφέρουμε το data block και έπειτα το όνομα της μεταβλητής όπως ακριβώς αυτό αναφέρεται.

```

<!-- AWP_In_Variable Name="StationX_2_Inverter".Inverter[1].Feedback_Timer_Set' -->
<!-- AWP_In_Variable Name="StationX_2_Inverter".Inverter[1].Safe_Start_Timer_Set' -->
<!-- AWP_In_Variable Name="StationX_2_Inverter".Inverter[1].Safe_Stop_Timer_Set' -->
<!-- AWP_In_Variable Name="StationX_2_Inverter".Inverter[1].Manual_Speed_Sp' -->
<!-- AWP_In_Variable Name="StationX_2_Inverter".Inverter[1].Min_Speed' -->
<!-- AWP_In_Variable Name="StationX_2_Inverter".Inverter[1].Max_Speed' -->
<!-- AWP_In_Variable Name="StationX_2_Inverter".Inverter[1].C_manual_On' -->
<!-- AWP_In_Variable Name="StationX_2_Inverter".Inverter[1].C_manual_Off' -->
<!-- AWP_In_Variable Name="StationX_2_Inverter".Inverter[1].C_Start_Frw_On' -->
<!-- AWP_In_Variable Name="StationX_2_Inverter".Inverter[1].C_Start_Frw_Off' -->
<!-- AWP_In_Variable Name="StationX_2_Inverter".Inverter[1].C_Ack' -->
<!-- AWP_In_Variable Name="Auto_Work_DB".Level_Start' -->
<!-- AWP_In_Variable Name="Auto_Work_DB".Level_Stop' -->
<!-- AWP_In_Variable Name="StationX_6_Inverter_View_Data".Inverter[1].Inv_OutFreq' -->
<!-- AWP_In_Variable Name="StationX_6_Inverter_View_Data".Inverter[1].Inv_OutCurr' -->
    
```

Εικόνα 70 Δήλωση μεταβλητών

Στην παρακάτω εικόνα φαίνεται η δήλωση της σειρήνας καθώς επίσης και η σελίδα ανανέωσης. Αυτό το κομμάτι είναι ίδιο σε όλες τις σελίδες. Αυτό που αλλάζει είναι το όνομα της σελίδας ανανέωσης.

```

<style>
.Siren{
background-image: url("Images/errors.png");
background-color: transparent;
background-repeat: no-repeat;
background-position: 0px 0px;
height: 80px
box-shadow:inset 0px 0px 5px 0px #91b8b3;
display:inline-block;
cursor:pointer;
padding:30px 30px;
z-index: 12;
position:absolute;
top:15%;
left:85%;
width: 15px;
height: 15px;
}
.Siren:active {
transform: translateY(1px);
}
</style>

<!DOCTYPE html>
<html>

<head>
<title>Userdefined Website - Application Start</title>
<meta charset="UTF-8" >
<meta http-equiv="refresh" content="10; URL=Inverter1.html" > <!-- ΔΗΛΩΣΗ ΤΗΣ ΤΡΕΧΟΥΣΑΣ ΣΕΛΙΔΑΣ ΓΙΑ ΑΝΑΝΕΩΣΗ -->
<link type="text/css" rel="stylesheet" href="Stylesheet/siemens_Stylesheet.css"/>
<link type="text/css" rel="stylesheet" href="Stylesheet/bootstrap.min.css">
<script src="js/jquery-3.6.0.min.js"></script>
</head>
    
```

Εικόνα 71 Κοινός κώδικας. Αλλαγή μόνο στην σελίδα ανανέωσης

Και το παρακάτω κομμάτι κώδικα είναι κοινό καθώς αλλάζει μόνο η σελίδα και γίνεται απλό κείμενο.

```

<!-- MENU SELIDES START-->
<div class="containerNavi">
  <ul>
    <a> ----- ΜΕΝΟΥ -----</a><br><br>
    <ul>
      <a href="Sensor1.html">Στάθμη</a><br>
      <br>
      <a href="Inverter2.html">Αντλητικό Συγκρότημα 1</a><br>
      <br>
      <a href="Start.html">Αρχική Σελίδα</a><br>
    </ul>
  </ul>
</div>
<!-- MENU SELIDES END-->

```

Εικόνα 72 Σελίδα φόρτωσης εικόνων

Επίσης κοινός κώδικας για την εμφάνιση της σειράνας.

```

<!-- EPIKEFALIDA START -->
<div id="header">
  <table border="0" >
    <tr>
      <td width="500px"></td>
      <td width="500px"></td>
      <td width="450px"> <h2>Στάθμη := "Level".Pv: m</h2></td>
      <td width="650px"><h2>Αντλητικό Συγκρότημα 1</h2></td>
      <td width="400px"><h2>ΤΣΕ 1 - ΑΝΑΤΙΟΣΤΑΣΙΟ ΑΥΜΑΤΩΝ</h2></td>
    </tr>
  </table>
</div>
<!-- EPIKEFALIDA END-->

<script>
setInterval(function() {
  if(Plc_Enable_Siren_V00_DB1.Signal_on; == 1 && Plc_Enable_Siren_V00_DB2.Signal_on; == 1 && Plc_Enable_Siren_V00_DB3.Signal_on; == 1) {
    document.getElementById("sir").style.visibility = 'visible';}
  else if(Plc_Enable_Siren_V00_DB1.Signal_on; == 0 && Plc_Enable_Siren_V00_DB2.Signal_on; == 0 && Plc_Enable_Siren_V00_DB3.Signal_on; == 0) {
    document.getElementById("sir").style.visibility = 'hidden';}
  if($('.Siren').css('background-image').indexOf('errors2')>-1){
    $('.Siren').css('background-image',"url('Images/errors.png')");} else {
    $('.Siren').css('background-image',"url('Images/errors2.png')");}
},1000);
</script>

<div id="sir">
  <a class="Siren"></a>

```

Εικόνα 73 Κοινός κώδικας - εμφάνιση σειράνας


```

<table width="600px" height="50px" >
  <tr>
    <body>
      <td width="142px" align="left" height="30px">
        <font size="4" color="black" >Feedback time: </font>
      </td>
      <td width="12px" align="left" height="30px">
        <form method="post">
          <input type="number2" font size="2"
name=""StationX_2_Inverter".Inverter[1].Feedback_Timer_Set' value =
':"StationX_2_Inverter".Inverter[1].Feedback_Timer_Set:'> <!-- IN/OUT -->
        </form>
      </td>
      <td width="60px" align="left" height="30px">
        <font size="4" color="black" >sec </font>
      </td>
    </body>

    <td width="50px" align="center" height="30px">
    </td>

    <td width="248px" align="left" height="30px">
      <font size="4" color="black" >Working Hour: </font>
      := "PLA_Inverter_Control_V5_01".Working_Hour: <!-- OUT -->
    </td>
    <td width="80px" align="left" height="30px">
      <font size="4" color="black" >Ωρες </font>
    </td>
  </tr>
</table>

<table width="600px" height="50px" >
  <tr>
    <body>
      <td width="146px" align="left" height="30px">
        <font size="4" color="black" >Safe start time: </font>
      </td>
      <td width="12px" align="left" height="30px">
        <form method="post">
          <input type="number2" font size="2"
name=""StationX_2_Inverter".Inverter[1].Safe_Start_Timer_Set' value =
':"StationX_2_Inverter".Inverter[1].Safe_Start_Timer_Set:'>
        </form>
      </td>
      <td width="60px" align="left" height="30px">
        <font size="4" color="black" >sec </font>

```



```

</body>

<td width="50px" align="center" height="30px">
</td>

<td width="158px" align="left" height="30px">
  <font size="4" color="black" >Level Start Sp: </font>
</td>
<td width="10px" align="left" height="30px">
  <form method="post">
    <input type="number2" font size="2" name="'Auto_Work_DB'.Level_Start' value =
'="Auto_Work_DB".Level_Start:'> <!-- IN/OUT -->
  </form>
</td>
<td width="60px" align="left" height="30px">
  <font size="4" color="black" >m </font>

</td>
</tr>
</table>
<table width="540px" height="50px" >
<tr>
<td width="150px" align="left" height="30px">
  <font size="4" color="black" >Manual Speed: </font>
</td>
<td width="10px" align="left" height="30px">
  <form method="post">
    <input type="number2" font size="2"
name="'StationX_2_Inverter'.Inverter[1].Manual_Speed_Sp' value =
'="StationX_2_Inverter".Inverter[1].Manual_Speed_Sp:'> <!-- IN/OUT -->
  </form>
</td>
<td width="60px" align="left" height="30px">
  <font size="4" color="black" >Hz </font>

</td>

</body>

<td width="50px" align="center" height="30px">
</td>

<td width="147px" align="left" height="30px">
  <font size="4" color="black" >&nbsp;Level Stop Sp: </font>
</td>
<td width="10px" align="left" height="30px">
  <form method="post">

```

```

        <input type="number2" font size="2" name="'Auto_Work_DB'.Level_Stop' value =
        ':"Auto_Work_DB".Level_Stop:'> <!-- IN/OUT -->
    </form>
</td>
<td width="60px" align="left" height="30px">
    <font size="4" color="black" >m </font>

</td>
</tr>
</table>

<table width="244px" height="50px" >
<tr>
<body>
<td width="171px" align="left" height="30px">
    <font size="4" color="black" >Min Speed: </font>
</td>
<td width="10px" align="left" height="30px">
    <form method="post">
        <input type="number2" font size="2" name="'StationX_2_Inverter'.Inverter[1].Min_Speed'
        value = ':"StationX_2_Inverter".Inverter[1].Min_Speed:'> <!-- IN/OUT -->
    </form>
</td>
<td width="60px" align="left" height="30px">
    <font size="4" color="black" >Hz </font>

</td>

</body>
</tr>
</table>

<table width="244px" height="50px" >
<tr>
<body>
<td width="171px" align="left" height="30px">
    <font size="4" color="black" >Max Speed: </font>
</td>
<td width="10px" align="left" height="30px">
    <form method="post">
        <input type="number2" font size="2"
name="'StationX_2_Inverter'.Inverter[1].Max_Speed' value =
':"StationX_2_Inverter".Inverter[1].Max_Speed:'> <!-- IN/OUT -->
    </form>
</td>
<td width="60px" align="left" height="30px">
    <font size="4" color="black" >Hz </font>

</td>

</body>
</tr>
</table>

```

```

<br>
<table width="580px" height="20px" >
<tr>
<td width="170px" align="left" height="30px">
<font size="4" color="black">Ξηρά λειτουργία :</font>
</td>
<td width="61px" align="left" height="30px">
<img id="plantStatus" src=Images\Status0:="FEEDBACK_STATHMIS_STOP_SYGK1":.png'
height="20px" width="20px"/>
</td>
<td width="220px" align="left" height="30px">
<font size="4" color="black">Φλοτέρ στάθμης εκκίνησης :</font>
</td>
<td width="61px" align="left" height="30px">
<img id="plantStatus" src=Images\Status0:="FEEDBACK_STATHMIS_START_SYGK1":.png'
height="20px" width="20px"/>
</td>
</tr>
</table>
<table width="580px" height="20px" >
<tr>
<td width="170px" align="left" height="30px">
<font size="4" color="black">Εντολή από στάθμη :</font>
</td>
<td width="61px" align="left" height="30px">
<img id="plantStatus" src=Images\Status0:="Level_Start1":.png' height="20px" width="20px"/>
</td>
<td width="220px" align="left" height="30px">
<font size="4" color="black">Εντολή από φλοτέρ :</font>
</td>
<td width="61px" align="left" height="30px">
<img id="plantStatus" src=Images\Status0:="Level_Start2":.png' height="20px" width="20px"/>
</td>
</tr>
</table>

<!-- INVERTER SPEED & AUTO SETTINGS END-->

</div>
</div>

<!-- Content Area End-->
</body>
</html>
    
```

Ο προγραμματισμός για την δεύτερη αντλία είναι ακριβώς ίδιος. Η μόνη διαφορά στον κώδικα είναι η στην δήλωση της σελίδας που θα ανανεώνεται αλλά και που θα καλείται σε τον όνομα του data block.

Κεφάλαιο 5. Αποτελέσματα

Με την χρήση της συγκεκριμένης τεχνολογίας στη λειτουργία του αντλιοστασίου καταφέραμε να πετύχουμε τα παρακάτω:

- 1) Αξιοπιστία στη λειτουργία του συστήματος
- 2) Εύκολη ενημέρωση με τη λειτουργία συσκευών ευρείας χρήσης όπως κινητών ή τάμπλετ.
- 3) Αξιόπιστη μετάδοση δεδομένων χωρίς την επιβάρυνση του περιβάλλοντος, χρησιμοποιώντας εγκατεστημένα δίκτυα της κινητής τηλεφωνίας
- 4) Καταγραφή λειτουργιών και σφαλμάτων σε κεντρική βάση δεδομένων για μελλοντική χρήση και αξιολόγηση.
- 5) Άμεση ενημέρωση της επιβλέπουσας αρχής.
- 6) Μείωση λειτουργικών εξόδων παρακολουθώντας το σύστημα από το κέντρο ελέγχου.

Βιβλιογραφία - Πηγές

1. https://www.w3schools.com/html/html_form_input_types.asp
2. <https://www.bitdegree.org/learn/css-padding>
3. <https://www.dmcinfo.com/latest-thinking/blog/id/8567/siemens-s7-1200-web-server-tutorial--from-getting-started-to-html5-user-defined-pages>
4. <https://www.bitdegree.org/learn/css-button#colors>
5. <https://css-tricks.com/when-a-click-is-not-just-a-click/>
6. <https://www.freecodecamp.org/news/html-button-onclick-javascript-click-event-tutorial/>
7. https://www.math.uh.edu/~torok/math_6298/html/commands.html
8. <https://www.apu.ac.jp/~gunarto/html.html>
9. https://www.math.uh.edu/~torok/math_6298/html/commands.html#fonts
10. <https://openautomationsoftware.com/data-destinations/visualization-html5-web-application/web-hmi-graphics/>
11. <https://www.hostinger.com/tutorials/what-is-a-web-server>
12. https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server
13. <https://support.industry.siemens.com/cs/document/68011496/creating-and-using-user-defined-web-pages-on-s7-1200-s7-1500?dti=0&lc=en-WW>

Παράρτημα

Επεξήγηση Εντολών HTML

HTML Input Types

Παρακάτω είναι διάφορα είδη εισόδων που χρησιμοποιούνται στην HTML:

<input type="button"> : για την εισαγωγή κουμπιού

Παράδειγμα:

```
<input type="button" onclick="alert('Hello World!)" value="Click Me!">
```

```
<input type="checkbox">
```

<input type="color">: για την εισαγωγή χρώματος

Παράδειγμα:

```
<form>  
  <label for="favcolor">Select your favorite color:</label>  
  <input type="color" id="favcolor" name="favcolor">  
</form>
```

<input type="date">: για την εισαγωγή ημερομηνίας

Παράδειγμα:

```
<form>  
  <label for="birthday">Birthday:</label>  
  <input type="date" id="birthday" name="birthday">  
</form>
```

<input type="datetime-local">: για την εισαγωγή ημερομηνίας χωρίς συγκεκριμένη ζώνη ώρας

<input type="email">: για την εισαγωγή πεδίου που περιέχει διεύθυνση email

Παράδειγμα:

```
<form>  
  <label for="email">Enter your email:</label>  
  <input type="email" id="email" name="email">  
</form>
```

<input type="file">: για την εισαγωγή ενός κουμπιού για το ανέβασμα αρχείων

Παράδειγμα:

```
<form>  
  <label for="myfile">Select a file:</label>  
  <input type="file" id="myfile" name="myfile">  
</form>
```

<input type="hidden">: για την εισαγωγή ενός πεδίου που είναι αόρατο για τον χρήστη

Παράδειγμα:

```

<form>
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <input type="hidden" id="custId" name="custId" value="3487">
  <input type="submit" value="Submit">
</form>
  
```

<input type="image"> : για την εισαγωγή μιας εικόνας ως κουμπιού

Παράδειγμα:

```

<form>
<input type="image" src="img_submit.gif" alt="Submit" width="48" height="48">
</form>
  
```

<input type="month"> : για την εισαγωγή

<input type="number"> : για την εισαγωγή

<input type="password"> : για την εισαγωγή κωδικού:

Παράδειγμα:

```

<form>
  <label for="username">Username:</label><br>
  <input type="text" id="username" name="username"><br>
  <label for="pwd">Password:</label><br>
  <input type="password" id="pwd" name="pwd">
</form>
  
```

<input type="radio"> : για την εισαγωγή για κουμπί ραδιοφώνου

<input type="range"> : για την εισαγωγή ενός αριθμού ο οποίος δεν χρειάζεται να είναι ακριβής και μπορούμε να βάλουμε ένα εύρος τιμών με min και max

Παράδειγμα:

```

<form>
  <label for="vol">Volume (between 0 and 50):</label>
  <input type="range" id="vol" name="vol" min="0" max="50">
</form>
  
```

<input type="reset"> : για την εισαγωγή κουμπιών που κάνουν επαναφορά των τιμών σε αρχικές

Παράδειγμα:

```

<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
  <input type="reset">
</form>
  
```

<input type="search"> : για την εισαγωγή πεδίων αναζήτησης

Παράδειγμα:

```
<form>
  <label for="vol">Volume (between 0 and 50):</label>
  <input type="range" id="vol" name="vol" min="0" max="50">
</form>
```

<input type="submit"> : για την εισαγωγή ενός κουμπιού υποβολής δεδομένων

Παράδειγμα:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

<input type="tel"> : για την εισαγωγή ενός πεδίου που θα περιέχει αριθμό τηλεφώνου

Παράδειγμα:

```
<form>
  <label for="phone">Enter your phone number:</label>
  <input type="tel" id="phone" name="phone" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">
</form>
```

<input type="text"> : για την εισαγωγή κειμένου μίας γραμμής

Παράδειγμα:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

<input type="time"> : για την εισαγωγή ώρας

Παράδειγμα:

```
<form>
  <label for="appt">Select a time:</label>
  <input type="time" id="appt" name="appt">
</form>
```

<input type="url"> : για την εισαγωγή url

Παράδειγμα:

```
<form>
  <label for="homepage">Add your homepage:</label>
  <input type="url" id="homepage" name="homepage">
</form>
```

`<input type="week">` : για την εισαγωγή εβδομάδας

Παράδειγμα:

```
<form>
  <label for="week">Select a week:</label>
  <input type="week" id="week" name="week">
</form>
```

Λίστα εντολών html

`<a>`, ``

σκοπός: για συνδέσμους και ετικέτες

Παράδειγμα: `Dept. of Math.`

``

Σκοπός: εισαγωγή εικόνας

Παράδειγμα: ``

`
`

Σκοπός: για διακοπή γραμμής

`<p>`, `</p>`

σκοπός: για νέα παράγραφο

`<hr />`

Σκοπός: εισαγωγή οριζόντιας γραμμής

Παράδειγμα: `<hr noshade size="2" width="50%" align="center" />`

Γραμματοσειρές

Σκοπός: για την εισαγωγή γραμματοσειράς, επιλογή μεγέθους, επικεφαλίδας κλπ.

| Ετικέτες/ Tag | Σκοπός |
|--|---|
| <code></code> , <code></code> | Αλλαγή μεγέθους, χρώματος γραμματοσειράς |
| <code><i></code> , <code></i></code> | <i>Πλάγια γραφή</i> |
| <code></code> , <code></code> | Έντονη γραφή |
| <code></code> , <code></code> | <i>Έμφαση</i> |
| <code></code> , <code></code> | <i>Έντονη έμφαση</i> |
| <code><u></code> , <code></u></code> | <u>Υπογράμμιση</u> |
| <code><code></code> , <code></code></code> | Κώδικας |
| <code></code> , <code></code> | Απαγραφή |

Λίστες

Υπάρχουν 3 είδη λιστών τα οποία φαίνονται παρακάτω:

| Τύποι Λιστών | Προβολή | HTML Κώδικας |
|-----------------------|---|---|
| Λίστες χωρίς αρίθμηση | <ul style="list-style-type: none"> • list item 1 • list item 2 • list item 3 | <pre> list item 1 list item 2 list item 3 </pre> |
| Λίστες με αρίθμηση | <ol style="list-style-type: none"> 1. list item 1 2. list item 2 3. list item 3 | <pre> list item 1 list item 2 list item 3 </pre> |
| Λίστες ορισμού | <p>This is a term This is a definition And yet another definition</p> <p>Another term Another definition</p> | <pre><dl> <dt> This is a term <dd> This is a definition <dd> And yet another definition <dt> Another term <dd> Another definition </dl></pre> |

Πίνακες

| Χαρακτήρες | Προβολή |
|--------------------------|---------|
| Κενό | |
| Εισαγωγικά | “ |
| Σύμβολο «και» | & |
| Σύμβολο «μικρότερο από» | < |
| Σύμβολο «μεγαλύτερο από» | > |

| | |
|-------------------|----------------------|
| <table>, </table> | Πίνακας |
| <thead>, </thead> | Επικεφαλίδα πίνακα |
| <tfoot>, </tfoot> | Υποσέλιδο του πίνακα |
| <tbody>, </tbody> | Σώμα του πίνακα |
| <tr>, </tr> | Σειρά πίνακα |
| <th>, </th> | Κελί επικεφαλίδας |
| <td>, </td> | Κελί δεδομένων |

Παράδειγμα:

```
<TABLE border="1">
  <THEAD>
    <TR>
      <TH ROWSPAN=2>Character</TH>
      <TH ROWSPAN=2>Entity</TH>
      <TH ROWSPAN=2>Decimal</TH>
```

```
<TH ROWSPAN=2>Hex</TH>
<TH COLSPAN=3>Rendering in Your Browser</TH>
</TR>
<TR>
  <TH>Entity</TH>
  <TH>Decimal</TH>
  <TH>Hex</TH>
</TR>
</THEAD>
<TBODY>

<TR>
  <TD>non-breaking space</TD>
  <TD>&nbsp;</TD>
  <TD>&#160;</TD>
  <TD>&#xA0;</TD>
  <TD>&nbsp;</TD>
  <TD>&#160;</TD>
  <TD>&#xA0;</TD>
</TR>

etc.

</TBODY>
</TABLE>
```