

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ**  
**ΠΜΣ ΕΦΑΡΜΟΣΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΑΥΤΟΜΑΤΟΠΟΙΗΣΗΣ**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**« ΑΥΤΟΜΑΤΙΣΜΟΣ ΑΝΤΛΙΟΣΤΑΣΙΟΥ ΛΥΜΑΤΩΝ ΜΕ PLC  
ΚΑΙ ΑΝΑΠΤΥΞΗ ΕΞΑΤΟΜΙΚΕΥΜΕΝΟΥ ΛΟΓΙΣΜΙΚΟΥ SCADA »**



**ΦΟΙΤΗΤΗΣ:** ΛΕΚΟΣ ΕΛΕΥΘΕΡΙΟΣ (ΑΜ: 2020/017)

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:** ΤΣΑΓΚΑΡΗΣ ΑΠΟΣΤΟΛΟΣ

**ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ**

**ΘΕΣΣΑΛΟΝΙΚΗ ΙΟΥΛΙΟΣ 2022**

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Η παρούσα Διπλωματική Εργασία και τα συμπεράσματά της, σε οποιαδήποτε μορφή, αποτελούν συνιδιοκτησία του Τμήματος Μηχανικών Παραγωγής και Διοίκησης του Διεθνούς Πανεπιστημίου Ελλάδος και του φοιτητή. Οι προαναφερόμενοι διατηρούν το δικαίωμα ανεξάρτητης χρήσης και αναπαραγωγής (τμηματικά ή συνολικά) για διδακτικούς και ερευνητικούς σκοπούς. Σε κάθε περίπτωση πρέπει να αναφέρεται ο τίτλος, ο συγγραφέας, ο επιβλέπων και το τμήμα του ΔιΠαΕ.

Η έγκριση της παρούσας Διπλωματικής Εργασίας από το Τμήμα Μηχανικών Παραγωγής και Διοίκησης δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

-----

Ο υπογεγραμμένος δηλώνω υπεύθυνα ότι η παρούσα Διπλωματική Εργασία είναι εξ' ολοκλήρου δικό μου έργο και συγγράφηκε ειδικά για τις απαιτήσεις του μεταπτυχιακού προγράμματος σπουδών "Εφαρμοσμένα Συστήματα Αυτοματοποίησης" του Τμήματος Μηχανικών Παραγωγής και Διοίκησης.

Δηλώνω υπεύθυνα ότι κατά τη συγγραφή ακολούθησα την πρέπουσα ακαδημαϊκή δεοντολογία αποφυγής λογοκλοπής και έχω αποφύγει οποιαδήποτε ενέργεια που συνιστά παράπτωμα λογοκλοπής.

Ιούλιος 2022

Λέκος Ελευθέριος

(Υπογραφή)

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

### **Ευχαριστίες**

Ευχαριστώ θερμά τον επιβλέποντα καθηγητή, κύριο Απόστολο Τσαγκάρη, για τη συνεχή καθοδήγηση και υποστήριξη του, όπως επίσης και για την εμπιστοσύνη που μου έδειξε στην εκπόνηση της παρούσας διπλωματικής εργασίας.

### **Αφιέρωση**

Την εργασία αυτή την αφιερώνω στη σύζυγο μου Ελένη και στα παιδιά μου Κωνσταντίνο και Μαρία για τη στήριξη τους στην προσπάθεια μου αυτή.

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>ΠΕΡΙΛΗΨΗ</b> .....	<b>6</b>
<b>ABSTRACT</b> .....	<b>7</b>
<b>ΕΙΣΑΓΩΓΗ</b> .....	<b>8</b>
<b>ΚΕΦΑΛΑΙΟ 1 – ΠΕΡΙΓΡΑΦΗ ΣΤΟΙΧΕΙΩΝ ΑΝΤΛΙΟΣΤΑΣΙΟΥ ΛΥΜΑΤΩΝ</b> .....	<b>10</b>
ΓΕΝΙΚΑ .....	10
ΑΝΑΛΟΓΙΚΑ ΑΙΣΘΗΤΗΡΙΑ ΣΤΑΘΜΗΣ .....	11
ΗΛΕΚΤΡΟΜΗΧΑΝΙΚΟΙ ΦΛΟΤΕΡΟΔΙΑΚΟΠΤΕΣ .....	12
ΑΝΤΛΙΕΣ .....	13
ΗΛΕΚΤΡΙΚΟΣ ΠΙΝΑΚΑΣ .....	13
<i>Επιτηρητής Τάσης</i> .....	14
<i>Θερμική προστασία αντλιών</i> .....	15
<i>Ρελέ Ισχύος</i> .....	15
<i>Επιλογικοί Διακόπτες – Μπουτόν Λειτουργίας</i> .....	16
ΌΡΓΑΝΟ ΜΕΤΡΗΣΗΣ ΠΑΡΟΧΗΣ .....	16
ΗΛΕΚΤΡΙΚΑ ΣΗΜΑΤΑ .....	17
<b>ΚΕΦΑΛΑΙΟ 2 – ΠΕΡΙΓΡΑΦΗ ΠΡΟΓΡΑΜΜΑΤΟΣ SCADA - ΚΑΤΑΝΟΗΣΗ ΑΥΤΟΜΑΤΙΣΜΟΥ</b> .....	<b>18</b>
ΓΕΝΙΚΑ .....	18
ΠΡΟΣΟΜΟΙΩΣΗ ΕΙΣΑΓΩΓΗΣ – ΕΞΑΓΩΓΗΣ ΛΥΜΑΤΩΝ ΚΑΙ ΣΤΑΘΜΗΣ ΔΕΞΑΜΕΝΗΣ.....	19
ΠΡΟΣΟΜΟΙΩΣΗ ΨΗΦΙΑΚΩΝ ΗΛΕΚΤΡΙΚΩΝ ΣΗΜΑΤΩΝ ΚΑΙ ΣΦΑΛΜΑΤΑ ΠΟΥ ΠΑΡΑΓΟΝΤΑΙ ΑΠΟ ΑΥΤΑ.....	20
<i>Προσομοίωση επιτηρητή τάσης</i> .....	20
<i>Προσομοίωση θερμικής προστασίας αντλίας</i> .....	21
<i>Προσομοίωση επιβεβαίωσης λειτουργίας ρελέ ισχύος αντλίας</i> .....	21
<i>Σφάλμα επιβεβαίωσης λειτουργίας ρελέ ισχύος αντλίας</i> .....	22
<i>Προσομοίωση διακόπτη επιλογής λειτουργίας αντλίας</i> .....	23
<i>Προσομοίωση φλοτεροδιακοπών (φλοτέρ)</i> .....	24
ΠΕΡΙΓΡΑΦΗ ΛΕΙΤΟΥΡΓΙΑΣ ΑΥΤΟΜΑΤΙΣΜΟΥ ΑΝΤΛΙΟΣΤΑΣΙΟΥ .....	25
<i>Γενικά</i> .....	25
<i>Καθορισμός ορίων στάθμης</i> .....	26
<i>Επιλογή λειτουργίας αντλίας – πληροφορίες αντλίας</i> .....	28
<i>Γραφική απεικόνιση του τρόπου λειτουργίας και της κατάστασης της αντλίας</i> .....	30
<i>Χειροκίνητη Λειτουργία</i> .....	31
<i>Αυτόματη Λειτουργία</i> .....	31
<i>Αναγκαστική Εναλλαγή Αντλιών</i> .....	34
<i>Ρυθμίσεις αναλογικών αισθητήριων</i> .....	35
<i>Αυτόματη Λειτουργία σε περίπτωση βλάβης αισθητηρίου</i> .....	36
<i>Ημερήσια κυβικά εξόδου</i> .....	36
ΚΑΤΑΓΡΑΦΕΣ.....	37
<b>ΚΕΦΑΛΑΙΟ 3 – ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ PLC</b> .....	<b>40</b>
ΓΕΝΙΚΑ .....	40
.....	42
ΥΠΟΡΟΥΤΙΝΑ MAIN_ANALOGIKA_SIMATA .....	43
<i>Υπορουτίνα Sub_Simulate_Level</i> .....	43
<i>Υπορουτίνα Sub_Scale_Analog_4_20_mA_Me_Metatropi</i> .....	44
ΥΠΟΡΟΥΤΙΝΑ MAIN_SETPOINT_STATHMIS_ELEGXOS_FLOTER.....	46
<i>Υπορουτίνα Sub_Analog_Filter</i> .....	46

## «Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Υπορουτίνα <i>Sub_Digital_Filter</i> .....	51
ΥΠΟΡΟΥΤΙΝΑ MAIN_ORES_LEITOURGIAS .....	52
Υπορουτίνα <i>Sub_Sum_Operation_Hours</i> .....	52
ΥΠΟΡΟΥΤΙΝΑ MAIN_ENALLAGI_ANTLION .....	53
Υπορουτίνα <i>Sub_Alternate_3_Pumps_Hours</i> .....	53
Υπορουτίνα <i>Sub_On_Timer_DINT</i> .....	54
ΥΠΟΡΟΥΤΙΝΑ MAIN_LEITOURGIA_ANTLION .....	55
ΥΠΟΡΟΥΤΙΝΑ MAIN_OPERATION_ANIMATION.....	55
Υπορουτίνα <i>Sub_Simple_Motor</i> .....	55
ΥΠΟΡΟΥΤΙΝΑ MAIN_TOTAL_ALARMS.....	56
Υπορουτίνα <i>Sub_Feedback_Alarm</i> .....	57
<b>ΚΕΦΑΛΑΙΟ 4 – ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ SCADA .....</b>	<b>59</b>
ΓΕΝΙΚΑ .....	59
ΟΘΟΝΗ MAINFORM .....	60
<i>Εμφάνιση των τριών κύριων οθονών</i> .....	60
<i>Σύνδεση και επικοινωνία με το PLC</i> .....	61
<i>Δήλωση Μεταβλητών</i> .....	62
<i>Ενημέρωση μεταβλητών</i> .....	63
<i>Γράψιμο στη βάση δεδομένων</i> .....	63
ΟΘΟΝΗ ΓΕΝΕΡΟΡΤΕΙΑ .....	66
<i>Μπουτόν Προσομοίωσης</i> .....	66
<i>Πεδία τιμών (NumericUpDown)</i> .....	67
<i>Πεδία εικόνων (Picture Box)</i> .....	68
<i>Λεζάντες (Label)</i> .....	69
<i>Μπουτόν Reset</i> .....	70
<i>Animation Αντλιών</i> .....	71
<i>Πλαίσια Διαλόγου</i> .....	72
ΟΘΟΝΗ ΚΑΤΑΓΡΑΦΕΣ .....	73
<b>ΚΕΦΑΛΑΙΟ 5 – ΣΥΓΚΡΙΣΗ ΣΥΜΒΑΤΙΚΩΝ ΜΕΘΟΔΩΝ ΑΝΑΠΤΥΞΗΣ SCADA ΚΑΙ ΤΗΣ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΟΥΣ ΓΛΩΣΣΑΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ MICROSOFT C# .....</b>	<b>76</b>
ΓΕΝΙΚΑ .....	76
ΕΠΙΚΟΙΝΩΝΙΑ ΜΕ ΤΟ PLC – ΔΗΜΙΟΥΡΓΙΑ ΜΕΤΑΒΛΗΤΩΝ .....	76
<i>WinCC Advanced</i> .....	76
<i>C#</i> .....	78
<i>Σύγκριση</i> .....	78
ΠΕΔΙΑ ΤΙΜΩΝ.....	79
<i>WinCC Advanced</i> .....	79
<i>C#</i> .....	79
<i>Σύγκριση</i> .....	80
ΑΠΟΤΕΛΕΣΜΑΤΑ ΣΥΓΚΡΙΣΗΣ .....	80
<i>Κόστος</i> .....	80
<i>Ευελιξία</i> .....	80
<b>ΚΕΦΑΛΑΙΟ 6 – ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΕΞΕΛΙΞΗ ΣΥΣΤΗΜΑΤΟΣ .....</b>	<b>81</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ – ΠΗΓΕΣ.....</b>	<b>83</b>

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

## ΠΕΡΙΛΗΨΗ

Στις μέρες μας ο πληθυσμός των πόλεων συνεχώς αυξάνεται. Η σημαντική αυτή αύξηση οδηγεί στην όλο και μεγαλύτερη παραγωγή αστικών λυμάτων, τα οποία οι εταιρείες αποχέτευσης των πόλεων πρέπει να διαχειριστούν.

Ευθύνη των εταιρειών είναι η συλλογή των λυμάτων και η μεταφορά τους στις μονάδες επεξεργασίας (βιολογικοί καθαρισμοί). Για τον σκοπό αυτό τοποθετούνται σε διάφορα σημεία των πόλεων αντλιοστάσια λυμάτων, στα οποία καταλήγουν τα αστικά λύματα. Τα αντλιοστάσια αυτά στη συνέχεια, χρησιμοποιώντας τις αντλίες που διαθέτουν, προωθούν τα λύματα στις μονάδες επεξεργασίας.

Χαρακτηριστικό των αντλιοστασίων αυτών είναι ότι πρέπει να λειτουργούν όλο το 24ώρο, όλο τον χρόνο, γιατί στην περίπτωση που σταματήσουν πραγματοποιείται υπερχειλίση λυμάτων που μπορεί να θέσει σε κίνδυνο τη δημόσια υγεία (κυρίως σε τουριστικές περιοχές). Επίσης λόγω του μεγάλου αριθμού τους δεν υπάρχει η δυνατότητα να υπάρχει προσωπικό σε αυτά καθ' όλη τη διάρκεια της ημέρας. Αποτελεί επομένως μονόδρομο η αξιόπιστη αυτοματοποίηση τους.

Η εργασία αυτή παρουσιάζει την αυτοματοποίηση ενός αντλιοστασίου λυμάτων (το οποίο αποτελείται από τρίδυμο συγκρότημα αντλιών) με τη βοήθεια προγραμματιζόμενου λογικού ελεγκτή PLC, λαμβάνοντας υπ' όψη όλα τα ιδιαίτερα χαρακτηριστικά ενός τέτοιου αντλιοστασίου.

Η αυτοματοποίηση περιλαμβάνει την ανάπτυξη του προγράμματος του PLC, με βάση όλα τα απαραίτητα ηλεκτρικά σήματα και τη δημιουργία προγράμματος απεικόνισης SCADA για τον πλήρη έλεγχο του αντλιοστασίου.

**Λέξεις κλειδιά:** Προγραμματιζόμενος Λογικός Ελεγκτής PLC, πρόγραμμα απεικόνισης SCADA, αντλιοστάσιο λυμάτων, λογισμικό TIA Portal, γλώσσα προγραμματισμού Microsoft C#, βάση δεδομένων SQL

## **ABSTRACT**

Nowadays cities population continually increases. This leads into a big production of sewage which the sewer companies must handle.

Is their responsibility to collect sewage and transport them to wastewater treatment plants. For this purpose, they place sewage pumping plants into the cities, where the sewage ends up. These plants use special pumps to send sewage to wastewater treatment plants.

Main characteristic of sewage pumping plants is that they must work 24/7 all year. In case they fail, they overflow, which can be very dangerous for public health (especially in tourist attractions). Another characteristic is that due to their big number they need a lot of employees to watch their operation. It is therefore a one-way street the reliable automation of these plants.

This Paper presents the automation of a sewage pumping plant (with 3 pumps) with a programmable logic controller PLC, taking into account all the special features of such a pumping plant.

The automation includes the software development of the PLC, taking into account all the necessarily electrical signals and the software development of Supervisory Control and Data Acquisition System (SCADA) for full remote operation of the sewage pumping plant.

**Keywords:** Programmable Logic Controller PLC, Supervisory Control and Data Acquisition System SCADA, sewage pumping plant, TIA Portal software, Microsoft C#, SQL Server

## ΕΙΣΑΓΩΓΗ

Η εργασία αυτή παρουσιάζει την αυτοματοποίηση ενός αντλιοστασίου λυμάτων (το οποίο αποτελείται από τρίδυμο συγκρότημα αντλιών) με τη βοήθεια προγραμματιζόμενου λογικού ελεγκτή PLC και τη δημιουργία προγράμματος απεικόνισης SCADA για τον πλήρη έλεγχο του αντλιοστασίου.

Στόχος της εργασίας αυτής ήταν:

- 1) Να γίνουν κατανοητές οι ανάγκες αυτοματοποίησης ενός αντλιοστασίου λυμάτων και να καλυφθούν πλήρως από το πρόγραμμα του PLC.
- 2) Να γίνει προσομοίωση της λειτουργίας του αντλιοστασίου έτσι ώστε να μπορούν να γίνουν όλες οι απαραίτητες δοκιμές.
- 3) Να δημιουργηθεί πρόγραμμα απεικόνισης SCADA μέσω του οποίου θα ελέγχονται και θα μεταβάλλονται όλες οι παράμετροι λειτουργίας του αντλιοστασίου.
- 4) Να γίνει καταγραφή των πλεονεκτημάτων και μειονεκτημάτων στη δημιουργία προγράμματος απεικόνισης SCADA, μεταξύ των συμβατικών μεθόδων και μιας αντικειμενοστραφούς γλώσσας προγραμματισμού όπως είναι η Microsoft C#.

**Για το σκοπό αυτό το πρόγραμμα SCADA αναπτύχθηκε 2 φορές. Αρχικά (συμβατικός τρόπος) με το λογισμικό WinCC Advanced της εταιρείας Siemens και στη συνέχεια με τη γλώσσα προγραμματισμού Microsoft C#.**

Για τις ανάγκες της εργασίας επιλέχθηκαν:

- 1) PLC S7-1200 (CPU 1214C DC/DC/DC) της εταιρείας Siemens η λειτουργία του οποίου προσομοιώνεται με το λογισμικό PLCSIM.
- 2) Το λογισμικό TIA Portal V15.1 (Siemens) με το οποίο έγινε ο προγραμματισμός του PLC (Step7 Professional).
- 3) Το λογισμικό WinCC Advanced (Siemens) με το οποίο έγινε η αρχική ανάπτυξη του προγράμματος Scada.
- 4) Η αντικειμενοστραφής γλώσσα προγραμματισμού C# της εταιρείας Microsoft με την οποία έγινε η τελική ανάπτυξη του προγράμματος SCADA.



Η διπλωματική εργασία περιλαμβάνει τα εξής κεφάλαια :

- **Κεφάλαιο 1** – Περιγραφή στοιχείων αντλιοστασίου λυμάτων  
Στο κεφάλαιο αυτό γίνεται αναφορά των ηλεκτρικών στοιχείων που απαιτούνται για τη λειτουργία του αντλιοστασίου και την αυτοματοποίηση του.
- **Κεφάλαιο 2** – Περιγραφή προγράμματος SCADA , κατανόηση αυτοματισμού  
Στο κεφάλαιο αυτό περιγράφονται όλες οι λειτουργίες του προγράμματος SCADA και μέσω αυτής της περιγραφής γίνεται κατανοητή η μέθοδος αυτοματοποίησης του αντλιοστασίου.
- **Κεφάλαιο 3** – Ανάλυση προγράμματος PLC  
Στο κεφάλαιο αυτό γίνεται παρουσίαση της δομής του προγράμματος του PLC και αναλύονται τα επιμέρους τμήματα του λογισμικού που αναπτύχθηκε.
- **Κεφάλαιο 4** – Ανάλυση προγράμματος SCADA  
Στο κεφάλαιο αυτό γίνεται παρουσίαση της δομής του προγράμματος SCADA και της επικοινωνίας – συνεργασίας με το πρόγραμμα του PLC.
- **Κεφάλαιο 5** – Σύγκριση συμβατικών μεθόδων ανάπτυξης SCADA και της αντικειμενοστραφούς γλώσσας προγραμματισμού Microsoft C#  
Στο κεφάλαιο αυτό παρουσιάζονται συγκεκριμένα σημεία του προγράμματος SCADA και πως αυτά αναπτύχθηκαν με τη συμβατική μέθοδο (λογισμικό WinCC Advanced) και με τη γλώσσα προγραμματισμού Microsoft C#. Στη συνέχεια γίνεται σύγκριση των δύο μεθόδων και παρουσιάζονται πλεονεκτήματα και μειονεκτήματα.
- **Κεφάλαιο 6** – Συμπεράσματα και εξέλιξη συστήματος  
Στο τελευταίο κεφάλαιο παρουσιάζονται τα τελικά συμπεράσματα της εργασίας και προτείνονται πιθανοί τρόποι εξέλιξης του τελικού συστήματος PLC-SCADA.

## **ΚΕΦΑΛΑΙΟ 1 – Περιγραφή στοιχείων αντλιοστασίου λυμάτων**

### Γενικά

Για να γίνει η αυτοματοποίηση ενός αντλιοστασίου λυμάτων πρέπει πρώτα να γίνει μια καταγραφή των ηλεκτρικών στοιχείων που συμμετέχουν στη λειτουργία του.

Ένα αντλιοστάσιο λυμάτων διαθέτει μια κεντρική δεξαμενή στην οποία συγκεντρώνονται τα λύματα από το δίκτυο αποχέτευσης (**Εικόνα 1.1**). Η δεξαμενή αυτή έχει κατάλληλο μέγεθος έτσι ώστε να μπορεί να αποθηκεύσει μια σημαντική ποσότητα λυμάτων. Ωστόσο όταν η στάθμη της δεξαμενής φτάσει σε συγκεκριμένα όρια, τότε τα λύματα θα πρέπει να αρχίσουν να απομακρύνονται από αυτή. Αυτό γίνεται με ειδικές αντλίες η δυναμικότητα των οποίων επιλέγεται ανάλογα με το πόσο γρήγορα πρέπει να λύματα να φύγουν από τη δεξαμενή. Συνήθως χρησιμοποιούνται 2 ή 3 αντλίες ανάλογα με την δυναμικότητα του αντλιοστασίου. Μόλις η στάθμη της δεξαμενής μειωθεί αρκετά οι αντλίες θα πρέπει να σταματήσουν να δουλεύουν για να προστατευτούν από ξηρά λειτουργία.



**Εικόνα 1.1**

Επομένως το πιο σημαντικό πράγμα που πρέπει να ελέγχεται είναι η στάθμη της δεξαμενής. Αυτό μπορεί να γίνει εναλλακτικά αλλά και ταυτόχρονα με 2 τρόπους:

- 1) Με αναλογικό αισθητήριο στάθμης.
- 2) Με χρήση ηλεκτρομηχανικού φλοτεροδιακόπτη.

### Αναλογικά Αισθητήρια Στάθμης

Τα αναλογικά αισθητήρια στάθμης που χρησιμοποιούνται στα αντλιοστάσια λυμάτων μπορούν να είναι:

- 1) **Τύπου υπερήχων (Εικόνα 1.2).** Τα αισθητήρια αυτά εκπέμπουν παλμούς ήχου οι οποίοι αντανακλώνται στην επιφάνεια του υγρού και επιστρέφουν στον μεταδότη. Ο αισθητήρας μετράει το χρόνο που χρειάστηκαν οι παλμοί ήχου να επιστρέψουν και γνωρίζοντας την ταχύτητα των παλμών υπολογίζει την απόσταση του αισθητήρα από το υγρό. Στη συνέχεια χρησιμοποιεί την απόσταση από τον πυθμένα της δεξαμενής (που έχει δηλωθεί κατά τη διάρκεια της παραμετροποίησης) και υπολογίζει τη στάθμη του υγρού. Επειδή η ταχύτητα του ήχου μεταβάλλεται με τη θερμοκρασία υπάρχει και ένας αισθητήρας θερμοκρασίας για την αντιστάθμιση των μεταβολών. Ένα από τα χαρακτηριστικά των αισθητηρίων αυτών είναι ότι πρέπει να υπάρχει μια ελάχιστη απόσταση από τη μέγιστη στάθμη του υγρού (συνήθως 30 cm)
- 2) **Τύπου radar (Εικόνα 1.3).** Η λειτουργία τους είναι παρόμοια με τα αισθητήρια υπερήχων με τη διαφορά ότι αντί για παλμούς ήχου χρησιμοποιούν μικροκύματα. Η ισχύς των μικροκυμάτων είναι μικρή οπότε τα αισθητήρια αυτά μπορούν να είναι κοντά σε ανθρώπους και μηχανές.
- 3) **Τύπου υδροστατικής πίεσης (Εικόνα 1.4)** . Τα αισθητήρια αυτά μετρούν την υδροστατική πίεση που ασκείται σε αυτά από το υγρό. Η υδροστατική πίεση είναι ανάλογη του ύψους του υγρού ανεξάρτητα από το σχήμα του δοχείου στο οποίο περιέχεται το υγρό.



**Εικόνα 1.2**



**Εικόνα 1.3**



**Εικόνα 1.4**

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Τα αισθητήρια τύπου υπερήχων και radar δεν έρχονται σε επαφή με τα λύματα ενώ τα αισθητήρια τύπου υδροστατικής πίεσης έρχονται γιατί τοποθετούνται στο χαμηλότερο σημείο της δεξαμενής. Τα αισθητήρια τύπου υδροστατικής πίεσης απαιτούν συχνότερο καθαρισμό ωστόσο δεν επηρεάζονται από τον αφρισμό που πιθανώς να δημιουργείται στη δεξαμενή.

Τα αναλογικά αισθητήρια στάθμης παράγουν ένα ηλεκτρικό σήμα (συνήθως 4..20mA) ανάλογο της στάθμης του υγρού που μετρούν. Το σήμα αυτό μπορεί να οδηγηθεί σε μια ενδεικτική συσκευή ή σε έναν προγραμματιζόμενο ελεγκτή (PLC).

### Ηλεκτρομηχανικοί φλοτεροδιακόπτες

Ο ηλεκτρομηχανικός φλοτεροδιακόπτης (**Εικόνα 1.5**) έχει μια μεταλλική μπάλα στο εσωτερικό του, η οποία ανάλογα με τη στάθμη του υγρού στο σημείο που τοποθετείται ο φλοτεροδιακόπτης, αλλάζει θέση μεταβάλλοντας την κατάσταση ενός διακόπτη (**Εικόνα 1.6 a&b**).



**Εικόνα 1.5**



Υψηλή Στάθμη

**Εικόνα 1.6a**



Χαμηλή Στάθμη

**Εικόνα 1.6b**

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Το μειονέκτημα των φλοτεροδιακοπών σε σχέση με τα αναλογικά αισθητήρια στάθμης είναι ότι για κάθε επίπεδο στάθμης που θέλουμε να ελέγξουμε πρέπει να τοποθετηθεί και ένας φλοτεροδιακόπτης. Επίσης η ακρίβεια στο έλεγχο της στάθμης δεν είναι μεγάλη και εξαρτάται από τον τρόπο τοποθέτησης.

Με τα αναλογικά αισθητήρια στάθμης μπορούμε να έχουμε όσο επίπεδα στάθμης θέλουμε να ελέγξουμε και η ακρίβεια των οργάνων αυτών είναι πολύ υψηλή.

Συνήθως στα αντλιοστάσια λυμάτων (και όχι μόνο) χρησιμοποιούνται τα αναλογικά αισθητήρια στάθμης ως κύριο μέσο μέτρησης της στάθμης και οι φλοτεροδιακόπτες ως εφεδρικό.

### Αντλίες

Όπως αναφέρθηκε η απομάκρυνση των λυμάτων γίνεται με ειδικές αντλίες οι οποίες βρίσκονται συνήθως μέσα στην δεξαμενή (**Εικόνα 1.7**) και είναι κατάλληλες για προώθηση λυμάτων.



**Εικόνα 1.7**

### Ηλεκτρικός Πίνακας

Οι αντλίες τροφοδοτούνται ηλεκτρικά από έναν ηλεκτρικό πίνακα (**Εικόνα 1.8**) ο οποίος περιλαμβάνει όλες τις απαραίτητες διατάξεις για τη λειτουργία και προστασία των αντλιών. Οι κυριότερες που συμμετέχουν στον αυτοματισμό του αντλιοστασίου είναι:

## «Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

- 1) Επιτηρητής Τάσης
- 2) Θερμική Προστασία
- 3) Ρελέ Ισχύος
- 4) Επιλογικοί Διακόπτες – Μπουτόν Λειτουργίας



**Εικόνα 1.8**

### Επιτηρητής Τάσης

Ο επιτηρητής τάσης (**Εικόνα 1.9**) είναι μια συσκευή που ελέγχει την τάση τροφοδοσίας του ηλεκτρικού πίνακα των αντλιών. Ανάλογα με το μοντέλο μπορεί να ελέγχει για υπόταση, υπέρταση, ασυμμετρία φάσεων, διαδοχή φάσεων και απώλεια φάσης. Η συσκευή αυτή έχει συνήθως μια μεταγωγική επαφή, η οποία διακόπτει το κύκλωμα τροφοδοσίας των αντλιών στην περίπτωση που η τάση τροφοδοσίας δεν είναι σωστή. Μόλις η τάση τροφοδοσίας επανέλθει σε φυσιολογικά επίπεδα, ο επιτηρητής περιμένει συγκεκριμένο χρόνο (παραμετροποιήσιμο) και επαναφέρει το κύκλωμα τροφοδοσίας των αντλιών. Αποτελεί πολύ σημαντικό στοιχείο για την προστασία των αντλιών.



**Εικόνα 1.9**

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

### Θερμική προστασία αντλιών

Οι διατάξεις θερμικής προστασίας των αντλιών επιτηρούν το ρεύμα που απορροφά κάθε αντλία από το ηλεκτρικό δίκτυο και στην περίπτωση που αυτό ξεπεράσει μια τιμή, διακόπτουν το κύκλωμα τροφοδοσίας των αντλιών. Υπάρχουν διάφορα είδη διατάξεων θερμικής προστασίας. Στα αντλιοστάσια λυμάτων χρησιμοποιούνται κυρίως τα απλά θερμικά (Εικόνα 1.10) και οι θερμομαγνητικοί διακόπτες (Εικόνα 1.11)



Εικόνα 1.10



Εικόνα 1.11

### Ρελέ Ισχύος

Τα ρελέ ισχύος (Εικόνα 1.12) είναι ηλεκτρικοί διακόπτες που συνδέουν τους ηλεκτρικούς κινητήρες στο ηλεκτρικό δίκτυο κατά επιλογή. Περιέχουν έναν ηλεκτρομαγνήτη ο οποίος όταν τροφοδοτηθεί με τάση έλκει και συγκρατεί τις ηλεκτρικές επαφές του ρελέ, με αποτέλεσμα ο ηλεκτρικός κινητήρας να τροφοδοτείται με τάση. Χρησιμοποιούνται πάρα πολύ σε όλα τα κυκλώματα αυτοματισμού.



Εικόνα 1.12

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

### Επιλογικοί Διακόπτες – Μπουτόν Λειτουργίας

Οι επιλογικοί διακόπτες (**Εικόνα 1.13**) και τα μπουτόν λειτουργίας (**Εικόνα 1.14**) τοποθετούνται συνήθως στην πρόσοψη του ηλεκτρικού πίνακα και επηρεάζουν τη λειτουργία των αντλιών. Διαθέτουν ηλεκτρικές επαφές οι οποίες μεταβάλλουν την σειρά του ηλεκτρολογικού κυκλώματος και αλλάζουν την λειτουργία του αυτοματισμού του πίνακα.



**Εικόνα 1.13**



**Εικόνα 1.14**

### Όργανο μέτρησης παροχής

Πολλές φορές στα αντλιοστάσια λυμάτων εγκαθίσταται στην έξοδο τους κατάλληλος αισθητήρας μέτρησης παροχής (**Εικόνα 1.15**). Πρόκειται συνήθως για ηλεκτρομαγνητικό παροχόμετρο που μετράει και καταγράφει την ποσότητα των λυμάτων που προωθεί το αντλιοστάσιο στη μονάδα επεξεργασίας λυμάτων. Από την καταγραφή και μελέτη των δεδομένων αυτών μπορούν να βγουν χρήσιμα συμπεράσματα για τη λειτουργία του αντλιοστασίου.



**Εικόνα 1.15**



«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Το όργανο μέτρησης παροχής παράγει ένα ηλεκτρικό σήμα (συνήθως 4..20mA) ανάλογο της παροχής που μετράει.

### Ηλεκτρικά σήματα

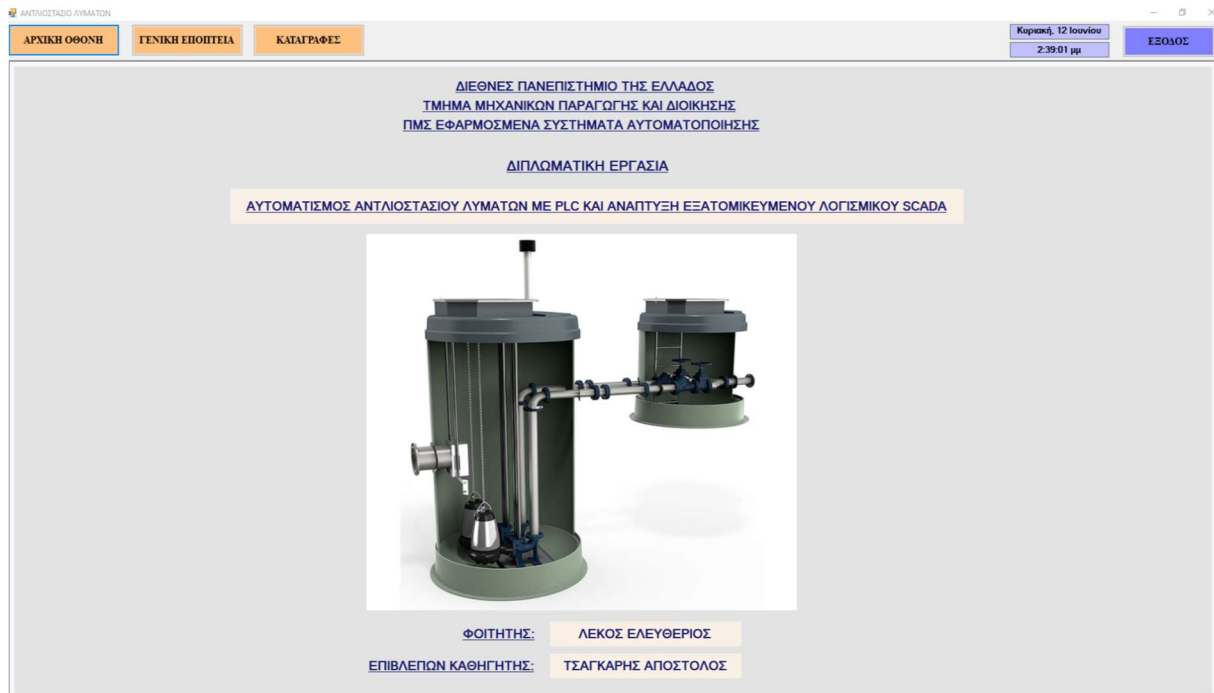
Στην εργασία αυτή το αντλιοστάσιο αποτελείται από τρίδυμο συγκρότημα αντλιών. Τα ηλεκτρικά σήματα που δέχεται το PLC προκειμένου να υλοποιηθεί ο αυτοματισμός του αντλιοστασίου είναι τα παρακάτω:

- 1) Επιτηρητής τάσης ηλεκτρικού πίνακα **(1)**
- 2) Θερμική προστασία κάθε αντλίας **(3)**
- 3) Επιβεβαίωση λειτουργίας ρελέ ισχύος κάθε αντλίας **(3)**
- 4) Διακόπτης επιλογής λειτουργίας κάθε αντλίας MANUAL – 0 – AUTO **(6)**
- 5) Αναλογική μέτρησης της στάθμης της δεξαμενής **(4..20mA)**
- 6) Ψηφιακός έλεγχος της στάθμης από φλοτεροδιακόπτες **(3)**
- 7) Αναλογική μέτρηση της παροχής εξόδου **(4..20mA)**

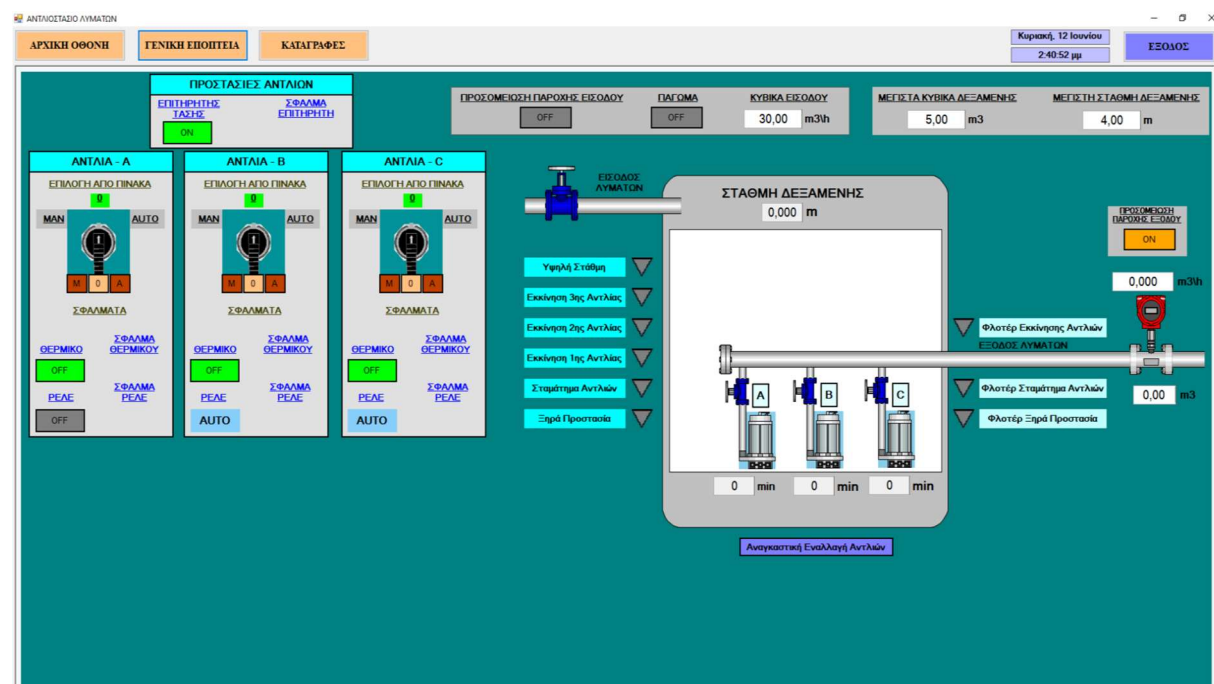
## ΚΕΦΑΛΑΙΟ 2 – Περιγραφή προγράμματος SCADA - κατανόηση αυτοματισμού

### Γενικά

Το πρόγραμμα απεικόνισης SCADA χρησιμοποιείται για την εποπτεία του αντλιοστασίου και την αλλαγή των παραμέτρων λειτουργίας του. Στην **εικόνα 2.1** φαίνεται η αρχική οθόνη, από την οποία πατώντας το κουμπί «ΓΕΝΙΚΗ ΕΠΟΠΤΕΙΑ» μεταβαίνουμε στην οθόνη απεικόνισης του αντλιοστασίου (**Εικόνα 2.2**)



Εικόνα 2.1



Εικόνα 2.2

## «Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

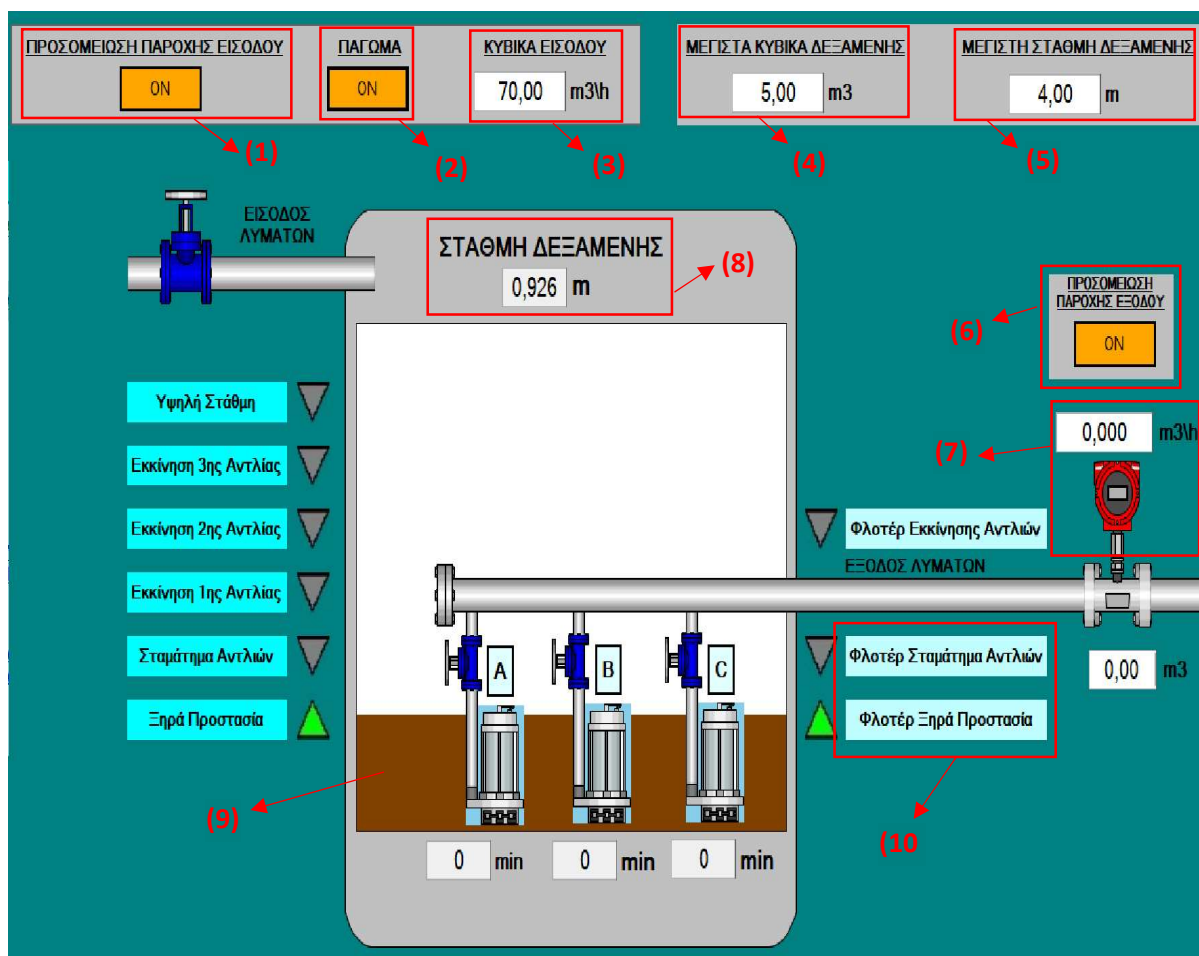
Το πρόγραμμα SCADA που αναπτύχθηκε εξυπηρετεί 2 σκοπούς:

- 1) Την προσομοίωση των ηλεκτρικών στοιχείων του αντλιοστασίου και της εισαγωγής και εξαγωγής λυμάτων στη δεξαμενή του.
- 2) Την παραμετροποίηση της αυτοματοποίησης του αντλιοστασίου και την εποπτεία λειτουργίας του.

**Προσομοιώνοντας τα ηλεκτρικά στοιχεία του αντλιοστασίου και την είσοδο και έξοδο των λυμάτων μπορούμε εύκολα να δοκιμάσουμε πλήρως τον αυτοματισμό που αναπτύχθηκε πριν γίνει η εφαρμογή του σε πραγματικό αντλιοστάσιο. Η προσομοίωση δίνει επίσης την δυνατότητα για επίδειξη των δυνατοτήτων του συστήματος χωρίς την παρουσία του μηχανικού ανάπτυξης.**



Πρέπει να σημειωθεί ότι για να γίνει η προσομοίωση, απαιτήθηκε η δημιουργία συγκεκριμένου κώδικα και στο πρόγραμμα του SCADA αλλά και του PLC.

### Προσομοίωση εισαγωγής – εξαγωγής λυμάτων και στάθμης δεξαμενής



Εικόνα 2.3

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Πατώντας το μπουτόν «ΠΡΟΣΟΜΕΙΩΣΗ ΠΑΡΟΧΗΣ ΕΙΣΟΔΟΥ» (Εικ.2.3 - 1) το μπουτόν πάει στην κατάσταση  και ξεκινάει η προσομοίωση της εισαγωγής λυμάτων στη δεξαμενή. Πατώντας ξανά το μπουτόν, αυτό πάει στην κατάσταση  και σταματά η προσομοίωση. Η ποσότητα των λυμάτων που εισέρχονται στη δεξαμενή καθορίζεται από το πεδίο «ΚΥΒΙΚΑ ΕΙΣΟΔΟΥ» (Εικ.2.3 - 3) .

Όταν εισέρχονται λύματα στη δεξαμενή αυτά απεικονίζονται γραφικά (Εικ.2.3 - 9) αλλά και σαν τιμή με την ενημέρωση του πεδίου «ΣΤΑΘΜΗ ΔΕΞΑΜΕΝΗΣ» (Εικ.2.3 - 8). Για τον υπολογισμό της γραφικής απεικόνισης και την ενημέρωση της στάθμης της δεξαμενής, λαμβάνονται υπόψη τα πεδία «ΜΕΓΙΣΤΑ ΚΥΒΙΚΑ ΔΕΞΑΜΕΝΗΣ» (Εικ.2.3 - 4) και «ΜΕΓΙΣΤΗ ΣΤΑΘΜΗ ΔΕΞΑΜΕΝΗΣ» (Εικ.2.3 - 5) .



Σε περίπτωση που γίνεται εισαγωγή λυμάτων τότε μπορούμε να κάνουμε πάγωμα της διαδικασίας με το μπουτόν «ΠΑΓΩΜΑ» (Εικ.2.3 - 2) .

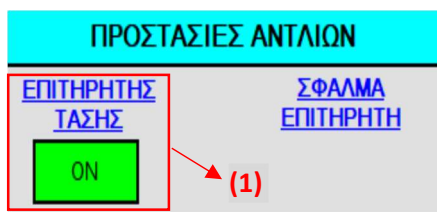
Όταν δουλεύει μία ή περισσότερες αντλίες μπορούμε να κάνουμε προσομοίωση της ποσότητας των λυμάτων που εξέρχονται από τη δεξαμενή. Αυτό γίνεται με το μπουτόν «ΠΡΟΣΟΜΕΙΩΣΗ ΠΑΡΟΧΗΣ ΕΞΟΔΟΥ» (Εικ.2.3 - 6) όπου λαμβάνετε υπ' όψη τη δυναμικότητα κάθε αντλίας . Η παροχή εξόδου απεικονίζεται στον αισθητήρα παροχής (Εικ.2.3 - 7).

**Είναι φανερό ότι με τις παραπάνω επιλογές μπορούμε να προσομοιώσουμε οποιασδήποτε δυναμικότητας αντλιοστάσιο λυμάτων.**

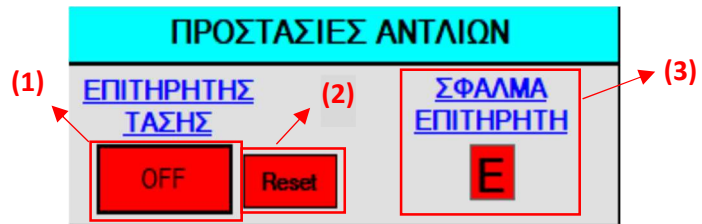
Προσομοίωση ψηφιακών ηλεκτρικών σημάτων και σφάλματα που παράγονται από αυτά

### Προσομοίωση επιτηρητή τάσης

Η προσομοίωση του επιτηρητή τάσης γίνεται με το μπουτόν «ΕΠΙΤΗΡΗΤΗΣ ΤΑΣΗΣ» (Εικ.2.4 - 1). Όταν το μπουτόν βρίσκεται στην κατάσταση  σημαίνει ότι ο επιτηρητής τάσης βρίσκεται σε κανονική λειτουργία, δηλαδή η τάση τροφοδοσίας είναι φυσιολογική. Όταν το μπουτόν βρίσκεται στην κατάσταση  (Εικ.2.5 - 1) σημαίνει ότι ο επιτηρητής τάσης έχει ανιχνεύσει κάποιο πρόβλημα στην τάση τροφοδοσίας. Στην περίπτωση αυτή ενεργοποιείται το «ΣΦΑΛΜΑ ΕΠΙΤΗΡΗΤΗ» (Εικ.2.5 - 3). **Το σφάλμα επιτηρητή δεν επιτρέπει στις αντλίες να λειτουργήσουν.** Όταν ο επιτηρητής τάσης επανέλθει σε φυσιολογική κατάσταση, το σφάλμα επιτηρητή παραμένει, μέχρι ο χειριστής του προγράμματος SCADA να πατήσει το κουμπί «Reset» (Εικ.2.5 - 2)



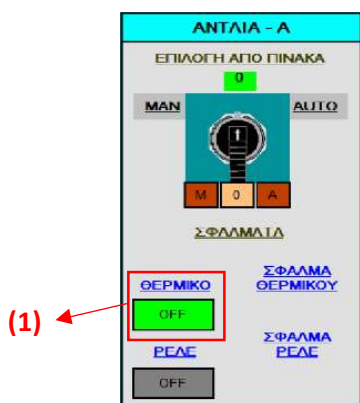
Εικόνα 2.4



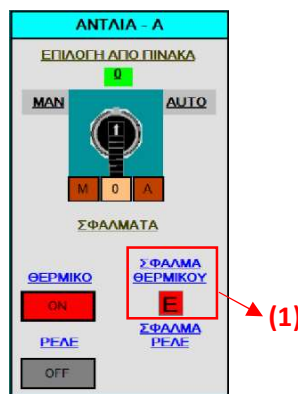
Εικόνα 2.5

Προσομοίωση θερμικής προστασίας αντλίας

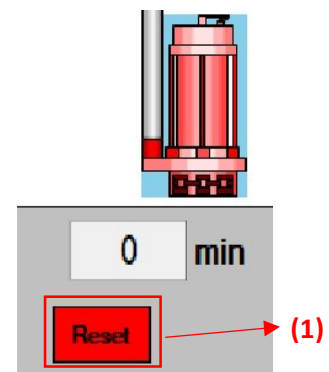
Η προσομοίωση της θερμικής προστασίας κάθε αντλίας γίνεται με το μπουτόν «ΘΕΡΜΙΚΟ» (Εικ.2.6 - 1). Όταν το μπουτόν βρίσκεται στην κατάσταση  OFF σημαίνει ότι δεν έχει ενεργοποιηθεί η θερμική προστασία της αντλίας, δηλαδή το ρεύμα που τη διαρρέει είναι φυσιολογικό. Όταν το μπουτόν βρίσκεται στην κατάσταση  ON σημαίνει ότι έχει ενεργοποιηθεί η θερμική προστασία. Στην περίπτωση αυτή ενεργοποιείται το «ΣΦΑΛΜΑ ΘΕΡΜΙΚΟΥ» (Εικ.2.7 - 1). Το σφάλμα θερμικού δεν επιτρέπει στην αντλία να λειτουργήσει. Όταν γίνει επαναφορά της θερμικής προστασίας, το σφάλμα θερμικού παραμένει, μέχρι ο χειριστής του προγράμματος SCADA να πατήσει το κουμπί «Reset» το οποίο βρίσκεται κάτω από κάθε αντλία (Εικ.2.8 - 1).



Εικόνα 2.6



Εικόνα 2.7

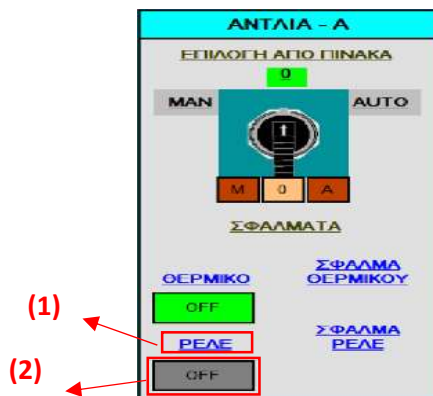


Εικόνα 2.8

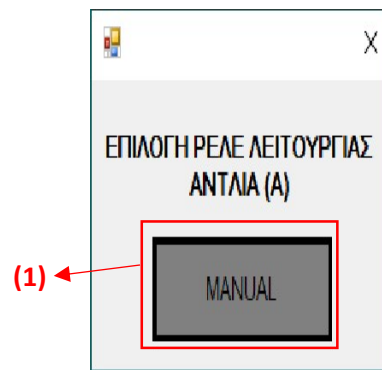
Προσομοίωση επιβεβαίωσης λειτουργίας ρελέ ισχύος αντλίας

Κάθε αντλία για να λειτουργήσει πρέπει να τροφοδοτηθεί με τάση. Το πότε θα λειτουργήσει κάθε αντλία το αποφασίζει ο αυτοματισμός του αντλιοστασίου, οπλίζοντας το ρελέ ισχύος της αντλίας. Όταν ο αυτοματισμός δίνει εντολή στο ρελέ ισχύος να οπλίσει, πρέπει να ξέρει εάν αυτό έγινε πραγματικά. Για το λόγο αυτό χρησιμοποιείται βοηθητική επαφή στο ρελέ ισχύος που δίνει την απαιτούμενη πληροφορία στο σύστημα αυτοματισμού.

Η προσομοίωση της επαφής αυτής γίνεται στο SCADA με δύο τρόπους, χειροκίνητο και αυτόματο. Πατώντας πάνω στην ταμπέλα «ΡΕΛΕ» (Εικ.2.9 - 1) εμφανίζεται το πλαίσιο «ΕΠΙΛΟΓΗ ΡΕΛΕ ΛΕΙΤΟΥΡΓΙΑΣ». Από το μπουτόν που υπάρχει στο πλαίσιο αυτό (Εικ.2.10 - 1) επιλέγεται ο χειροκίνητος ή ο αυτόματος τρόπος. Στον αυτόματο τρόπο, κάθε φορά που ο αυτοματισμός δίνει εντολή στην αντλία να ξεκινήσει, θεωρεί ότι το ρελέ ισχύος της έχει οπλίσει. Στο χειροκίνητο τρόπο πρέπει να πατηθεί το μπουτόν που υπάρχει κάτω από την ταμπέλα «ΡΕΛΕ» (Εικ.2.9 - 2) για να προσομοιωθεί η βοηθητική επαφή του ρελέ ισχύος.



Εικόνα 2.9



Εικόνα 2.10

### Σφάλμα επιβεβαίωσης λειτουργίας ρελέ ισχύος αντλίας

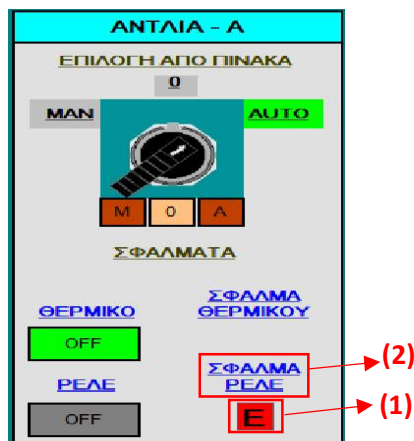
Όταν ο αυτοματισμός δίνει εντολή σε μια αντλία να ξεκινήσει και δεν πάρει επιβεβαίωση λειτουργίας από το ρελέ ισχύος της αντλίας, τότε μετά από χρόνο (παραμετροποιήσιμο), ενεργοποιείται το «ΣΦΑΛΜΑ ΕΠΙΒΕΒΑΙΩΣΗΣ ΡΕΛΕ» (Εικ.2.11 - 1). Όταν ενεργοποιηθεί το σφάλμα αυτό, τότε ο αυτοματισμός κόβει την εντολή προς την αντλία. Για να γίνει αποκατάσταση του σφάλματος πρέπει ο χειριστής του προγράμματος SCADA να πατήσει το κουμπί «Reset» το οποίο βρίσκεται κάτω από κάθε αντλία (Εικ.2.8 - 1)

Εάν πατηθεί η ταμπέλα «ΣΦΑΛΜΑ ΡΕΛΕ» (Εικ.2.11 - 2) εμφανίζεται το πλαίσιο «ΣΦΑΛΜΑ ΡΕΛΕ» στο οποίο βλέπουμε τα παρακάτω:

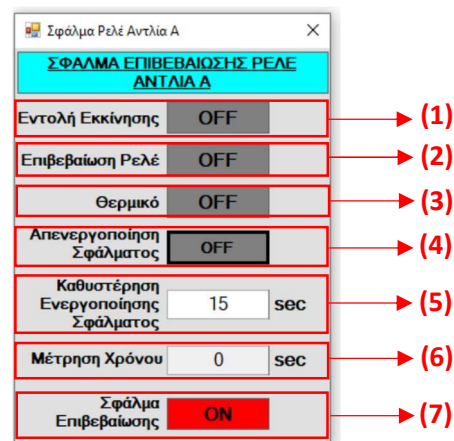
- Εάν έχει δοθεί εντολή στην αντλία να ξεκινήσει (Εικ.2.12 - 1).
- Εάν έχει οπλίσει το ρελέ ισχύος της αντλίας (Εικ.2.12 - 2).
- Εάν έχει ενεργοποιηθεί η θερμική προστασία της αντλίας (Εικ.2.12 - 3). Στην περίπτωση αυτή δεν υπολογίζεται το σφάλμα επιβεβαίωσης ρελέ.
- Ένα μπουτόν με το οποίο μπορούμε να απενεργοποιήσουμε τον υπολογισμό του σφάλματος επιβεβαίωσης ρελέ (Εικ.2.12 - 4). Αυτό μπορεί να είναι επιθυμητό για

παράδειγμα στην περίπτωση που έχει χαλάσει η βοηθητική επαφή του ρελέ ισχύος ή υπάρχει κάποιο πρόβλημα στην είσοδο του PLC.

- Τον χρόνο που ορίζουμε (παραμετροποιήσιμος), τον οποίο θα περιμένει ο αυτοματισμός πριν ενεργοποιήσει το σφάλμα (Εικ.2.12 - 5).
- Τον τρέχον χρόνο (Εικ.2.12 - 6) στην περίπτωση που υπάρχουν οι συνθήκες του σφάλματος.
- Την ενεργοποίηση του σφάλματος επιβεβαίωσης ρελέ (Εικ.2.12 - 7).



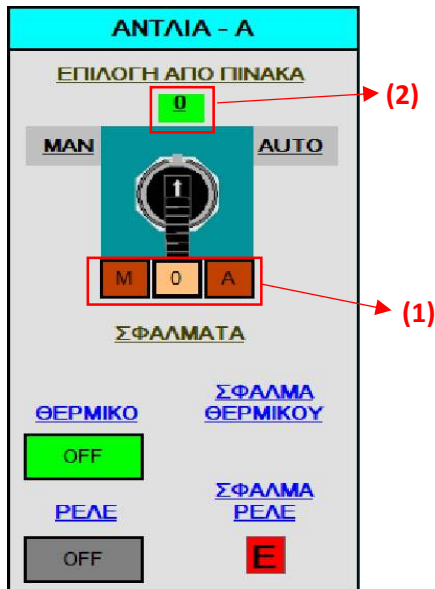
Εικόνα 2.11



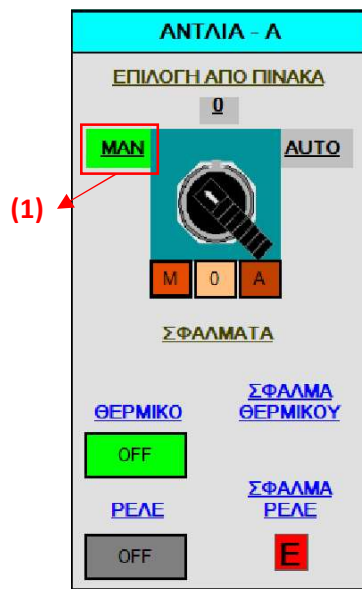
Εικόνα 2.12

### Προσομοίωση διακόπτη επιλογής λειτουργίας αντλίας

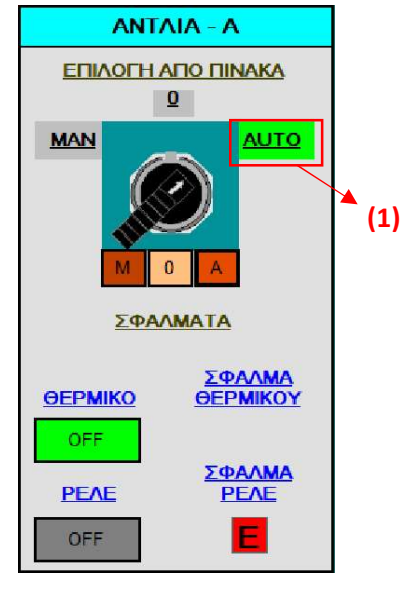
Στον ηλεκτρολογικό πίνακα των αντλιών υπάρχει ένας διακόπτης τριών θέσεων (MAN – 0 – AUTO) για κάθε αντλία, με τον οποίο επιλέγεται με ποιον τρόπο θα λειτουργήσει η αντλία. Η προσομοίωση του διακόπτη αυτού γίνεται με τα τρία μπουτόν **M** **0** **A** (Εικ.2.13-1). Σε περίπτωση που πατηθεί το μπουτόν **0** ο διακόπτης πάει στην κατάσταση **0** (Εικ.2.13-2) στην οποία η αντλία δεν μπορεί να λειτουργήσει. Όταν πατηθεί το μπουτόν **M** ο διακόπτης πάει στην κατάσταση **MANUAL** (Εικ.2.14 - 1) στην οποία δίνεται εντολή στην αντλία να ξεκινήσει και εφόσον δεν υπάρχει κάποιο σφάλμα σε αυτή, τότε η αντλία ξεκινά. Όταν πατηθεί το μπουτόν **A** ο διακόπτης πάει στη κατάσταση **AUTO** (Εικ.2.15 - 1) στην οποία η αντλία περιμένει εντολή να ξεκινήσει από τον αυτοματισμό που έχει υλοποιηθεί στο PLC.



Εικόνα 2.13



Εικόνα 2.14



Εικόνα 2.15

### Προσομοίωση φλοτεροδιακοπών (φλοτέρ)

Η προσομοίωση των φλοτεροδιακοπών γίνεται σε συνδυασμό με τη στάθμη της δεξαμενής. Πατώντας πάνω στην ταμπέλα καθενός φλοτέρ (Εικ.2.3 - 10) εμφανίζεται το πλαίσιο «ΨΗΦΙΑΚΟ SETPOINT ΕΚΚΙΝΗΣΗΣ ΑΝΤΛΙΩΝ». Στο πλαίσιο αυτό βλέπουμε τα παρακάτω:

- Σε ποιο σημείο έχει τοποθετηθεί το φλοτέρ μέσα στην δεξαμενή (Εικ.2.16 - 1). Με βάση αυτή την τιμή, σε συνδυασμό με τη στάθμη της δεξαμενής, ενεργοποιείται ή απενεργοποιείται το φλοτέρ
- Εάν έχει ενεργοποιηθεί το φλοτέρ (Εικ.2.16 - 2) .
- Την ενεργοποίηση ή απενεργοποίηση του φλοτέρ ανεξάρτητα από τη στάθμη της δεξαμενής (Εικ.2.16 – 3,4)



Η προσομοίωση και ο έλεγχος (δηλαδή το τελικό σήμα που παίρνουμε από το φλοτέρ) προκύπτει ως εξής:

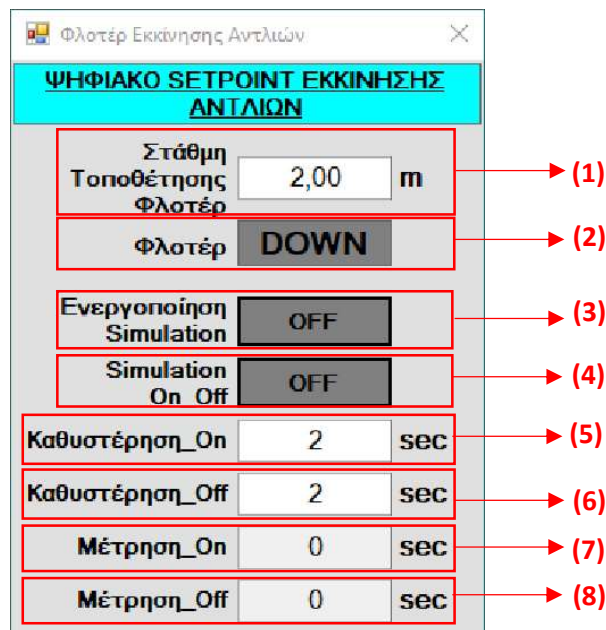
Εάν η στάθμη της δεξαμενής γίνει ίση ή μεγαλύτερη από το ύψος τοποθέτησης του φλοτέρ, τότε ξεκινάει να μετρά ο χρόνος που έχει οριστεί στο πεδίο «Καθυστέρηση\_On» (Εικ.2.16 – 5). Η μέτρηση του χρόνου αυτού φαίνεται στο πεδίο «Μέτρηση\_On» (Εικ.2.16 – 7). Μόλις ο χρόνος αυτός ολοκληρωθεί ενεργοποιείται το σήμα από το φλοτέρ.



## «Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Εάν η στάθμη της δεξαμενής γίνει μικρότερη από το ύψος τοποθέτησης του φλοτέρ, τότε ξεκινάει να μετρά ο χρόνος που έχει οριστεί στο πεδίο «Καθυστέρηση\_Off» (Εικ.2.16 – 6). Η μέτρηση του χρόνου αυτού φαίνεται στο πεδίο «Μέτρηση\_Off» (Εικ.2.16 – 8). Μόλις ο χρόνος αυτός ολοκληρωθεί απενεργοποιείται το σήμα από το φλοτέρ.

Όταν το σήμα φλοτέρ είναι ενεργοποιημένο τότε εμφανίζεται δίπλα από την ταμπέλα του φλοτέρ το σύμβολο  ενώ όταν είναι απενεργοποιημένο το σύμβολο .



Εικόνα 2.16

### Περιγραφή λειτουργίας αυτοματισμού αντλιοστασίου

#### Γενικά

Οι προϋποθέσεις για να μπορεί να δουλέψει μια αντλία με βάση τον αυτοματισμό που έχει αναπτυχθεί στο PLC είναι οι παρακάτω :

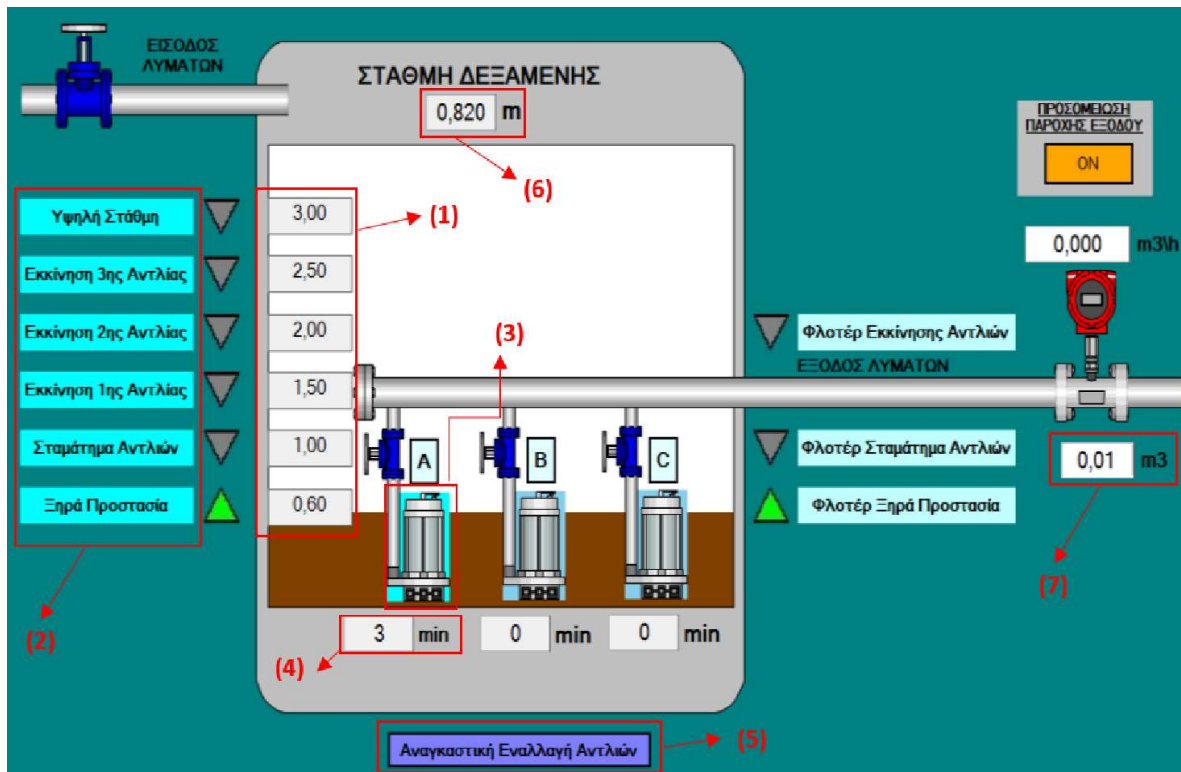
- 1) Ο διακόπτης επιλογής MANUAL – 0 – AUTO του ηλεκτρολογικού πίνακα να βρίσκεται στη θέση AUTO.
- 2) Να μην υπάρχει κάποιο σφάλμα στην αντλία (σφάλμα επιτηρητή, σφάλμα θερμικής προστασίας, σφάλμα επιβεβαίωσης ρελέ)
- 3) Να έχει επιλεγεί τρόπος λειτουργίας από το SCADA

## «Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Για να λειτουργήσει ο αυτοματισμός πρέπει να καθοριστούν τα όρια στάθμης (με βάση το αναλογικό όργανο στάθμης) στα οποία δουλεύει κάθε αντλία, το όριο που σταματάνε οι αντλίες και το όριο που αποτελεί την προστασία ξηράς λειτουργίας των αντλιών.

### Καθορισμός ορίων στάθμης

Για να εμφανιστούν τα όρια στάθμης (Εικ.2.17 – 1) πατάμε κάποια από τις ταμπέλες των ορίων (Εικ.2.17 – 2)



Εικόνα 2.17

Υπάρχουν 6 ρυθμιζόμενα όρια στάθμης τα οποία είναι τα παρακάτω :

- 1) **Ξηρά προστασία** : Κάτω από αυτό το όριο δεν επιτρέπεται να λειτουργήσουν οι αντλίες προκειμένου να προστατευτούν από ξηρά λειτουργία. Ξηρά λειτουργία είναι η λειτουργία μιας αντλίας χωρίς τη διέλευση υγρού από αυτήν, με αποτέλεσμα την υπερθέρμανση της και την καταστροφή των στεγανών της σημείων.
- 2) **Σταμάτημα αντλιών**: Κάτω από αυτό το όριο οι αντλίες σταματούν όταν βρίσκονται στην αυτόματη λειτουργία.
- 3) **Εκκίνηση 1<sup>ης</sup> Αντλίας** : Πάνω από αυτό το όριο ξεκινάει να λειτουργεί η πρώτη σε σειρά αντλία.
- 4) **Εκκίνηση 2<sup>ης</sup> Αντλίας** : Πάνω από αυτό το όριο ξεκινάει να λειτουργεί η δεύτερη σε σειρά αντλία.

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

- 5) **Εκκίνηση 3<sup>ης</sup> Αντλίας** : Πάνω από αυτό το όριο ξεκινάει να λειτουργεί η τρίτη σε σειρά αντλία.
- 6) **Υψηλή Στάθμη**: Πάνω από αυτό το όριο δημιουργείται συναγερμός (alarm) υψηλής στάθμης.



Για να ρυθμίσουμε ένα όριο στάθμης πατάμε πάνω στο πεδίο του ορίου. Τότε εμφανίζεται το πλαίσιο «ΑΝΑΛΟΓΙΚΟ SETPOINT» (Εικ.2.18). Στο πλαίσιο αυτό βλέπουμε τα παρακάτω:

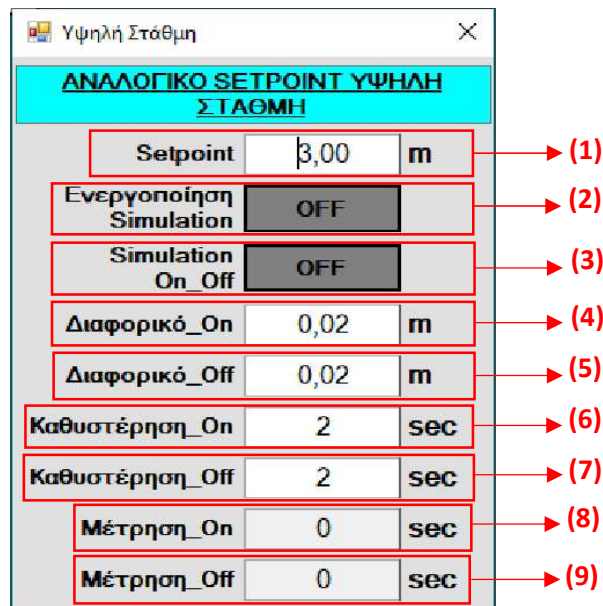
- Το πεδίο στο οποίο ορίζουμε το όριο στάθμης (Εικ.2.18 - 1). .
- Την **ενεργοποίηση** ή **απενεργοποίηση** του ορίου στάθμης ανεξάρτητα από τη στάθμη της δεξαμενής (Εικ.2.18 – 2,3)

Ο έλεγχος (δηλαδή η τελική πληροφορία που παίρνουμε από το όριο στάθμης) προκύπτει ως εξής:

Εάν η στάθμη της δεξαμενής γίνει μεγαλύτερη από το άθροισμα «Setpoint» (Εικ.2.18 – 1) και «Διαφορικό\_On» (Εικ.2.18 – 4) τότε ξεκινάει να μετρά ο χρόνος που έχει οριστεί στο πεδίο «Καθυστέρηση\_On» (Εικ.2.18 – 6). Η μέτρηση του χρόνου αυτού φαίνεται στο πεδίο «Μέτρηση\_On» (Εικ.2.18 – 8). Μόλις ο χρόνος αυτός ολοκληρωθεί **ενεργοποιείται** το όριο στάθμης.

Εάν η στάθμη της δεξαμενής γίνει μικρότερη ή ίση από τη διαφορά «Setpoint» (Εικ.2.18 – 1) μείον «Διαφορικό\_Off» (Εικ.2.18 – 5) τότε ξεκινάει να μετρά ο χρόνος που έχει οριστεί στο πεδίο «Καθυστέρηση\_Off» (Εικ.2.18 – 7). Η μέτρηση του χρόνου αυτού φαίνεται στο πεδίο «Μέτρηση\_Off» (Εικ.2.18 – 9). Μόλις ο χρόνος αυτός ολοκληρωθεί **απενεργοποιείται** το όριο στάθμης.

Όταν το όριο στάθμης είναι **ενεργοποιημένο** τότε εμφανίζεται δίπλα από την ταμπέλα του ορίου το σύμβολο  ενώ όταν είναι **απενεργοποιημένο** το σύμβολο 



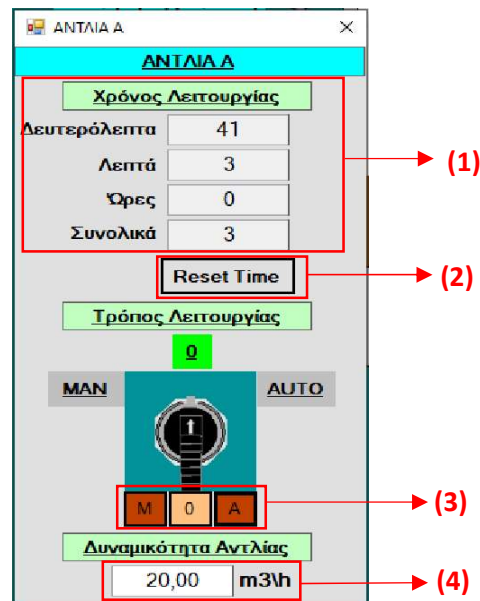
Εικόνα 2.18

#### Επιλογή λειτουργίας αντλίας – πληροφορίες αντλίας

Υπάρχουν δύο τρόποι με τους οποίους μπορεί να λειτουργήσει κάθε μια από τις αντλίες, με βάση τον αυτοματισμό που έχει αναπτυχθεί. Ο ένας τρόπος είναι ο **χειροκίνητος** και ο άλλος ο **αυτόματος**. Για επιλέξουμε τρόπο πρέπει να πατήσουμε πάνω σε μια αντλία (Εικ.2.17 – 3) οπότε εμφανίζεται το πλαίσιο της αντλίας (Εικ.2.19).

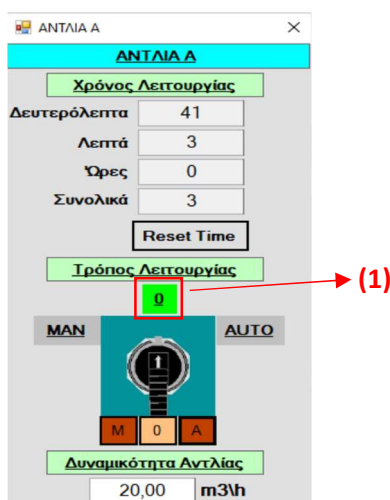
Στο πλαίσιο αυτό φαίνονται οι παρακάτω πληροφορίες :

- Ο χρόνος που έχει δουλέψει η αντλία (Εικ.2.19 – 1). Στις πληροφορίες του χρόνου περιλαμβάνονται τα δευτερόλεπτα, τα λεπτά και οι ώρες. Εμφανίζονται επίσης τα συνολικά λεπτά. Ο χρόνος λειτουργίας μπορεί να μηδενιστεί με το μπουτόν «Reset Time» (Εικ.2.19 – 2).
- Η δυναμικότητα της αντλίας (Εικ.2.19 – 4). Η τιμή αυτή παίζει ρόλο στην προσομοίωση της εξαγωγής των λυμάτων από την δεξαμενή.

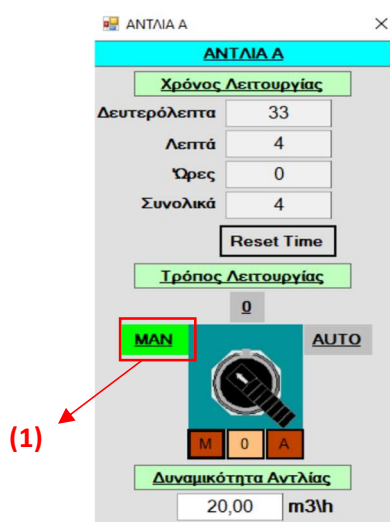


Εικόνα 2.19

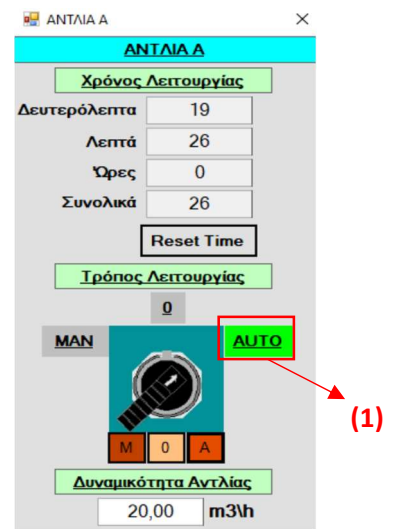
Η επιλογή του τρόπου λειτουργίας γίνεται από τα τρία μπουτόν **M** **0** **A**. Σε περίπτωση που πατηθεί το μπουτόν **0** ο διακόπτης πάει στην κατάσταση 0 (Εικ.2.20 - 1) στην οποία η αντλία δεν μπορεί να λειτουργήσει. Όταν πατηθεί το μπουτόν **M** ο διακόπτης πάει στην κατάσταση MANUAL (χειροκίνητη λειτουργία) (Εικ.2.21 - 1). Όταν πατηθεί το μπουτόν **A** ο διακόπτης πάει στην κατάσταση AUTO (αυτόματη λειτουργία) (Εικ.2.22 - 1).



Εικόνα 2.20



Εικόνα 2.21



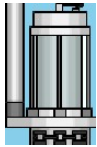
Εικόνα 2.22

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

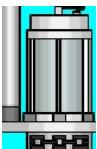
### Γραφική απεικόνιση του τρόπου λειτουργίας και της κατάστασης της αντλίας

Μια αντλία ανάλογα με τον τρόπο λειτουργίας που έχει επιλεγεί για αυτή και την κατάσταση στην οποία βρίσκεται απεικονίζεται γραφικά στο SCADA με διαφορετικά χρώματα. Παρακάτω φαίνονται όλοι οι πιθανοί συνδυασμοί :

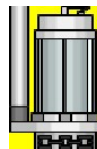
#### Η αντλία είναι σταματημένη



Ο επιλογικός διακόπτης του ηλεκτρολογικού πίνακα δεν είναι στη θέση AUTO.

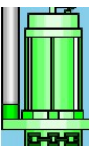


Ο επιλογικός διακόπτης του ηλεκτρολογικού πίνακα είναι στη θέση AUTO και ο επιλογικός διακόπτης του SCADA είναι στη θέση MANUAL.

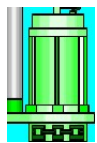


Ο επιλογικός διακόπτης του ηλεκτρολογικού πίνακα είναι στη θέση AUTO και ο επιλογικός διακόπτης του SCADA είναι στη θέση AUTO.

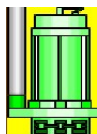
#### Η αντλία δουλεύει



Ο επιλογικός διακόπτης του ηλεκτρολογικού πίνακα δεν είναι στη θέση AUTO.

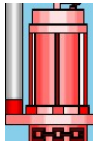


Ο επιλογικός διακόπτης του ηλεκτρολογικού πίνακα είναι στη θέση AUTO και ο επιλογικός διακόπτης του SCADA είναι στη θέση MANUAL.

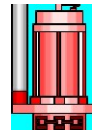


Ο επιλογικός διακόπτης του ηλεκτρολογικού πίνακα είναι στη θέση AUTO και ο επιλογικός διακόπτης του SCADA είναι στη θέση AUTO.

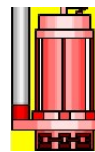
### Η αντλία είναι σε κατάσταση σφάλματος



Ο επιλογικός διακόπτης του ηλεκτρολογικού πίνακα δεν είναι στη θέση AUTO.



Ο επιλογικός διακόπτης του ηλεκτρολογικού πίνακα είναι στη θέση AUTO και ο επιλογικός διακόπτης του SCADA είναι στη θέση MANUAL.



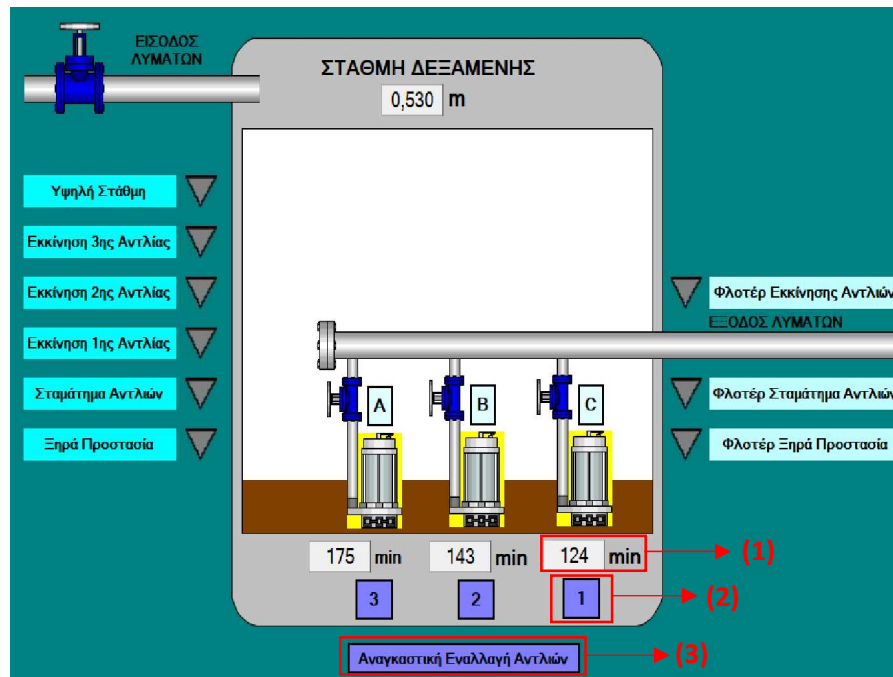
Ο επιλογικός διακόπτης του ηλεκτρολογικού πίνακα είναι στη θέση AUTO και ο επιλογικός διακόπτης του SCADA είναι στη θέση AUTO.

### Χειροκίνητη Λειτουργία

Μια αντλία στην **χειροκίνητη** λειτουργία, εφόσον ισχύουν οι προϋποθέσεις 1 και 2 που έχουν αναφερθεί στη σελ. 25, θα δουλεύει συνέχεια μέχρι να απενεργοποιηθεί το όριο «Ξηρά Προστασία». Η επιλογή αυτή υπάρχει για να μπορούμε να αδειάζουμε την δεξαμενή κατ' επιλογή, μέχρι το κατώτατο επιτρεπτό σημείο. Στην περίπτωση που υπάρχει βλάβη στο αισθητήριο στάθμης, τότε η αντλία σταματάει μόλις απενεργοποιηθεί το σήμα «Φλοτέρ Ξηρά Προστασία».

### Αυτόματη Λειτουργία

Μια αντλία στην αυτόματη λειτουργία, εφόσον ισχύουν οι προϋποθέσεις 1 και 2 που έχουν αναφερθεί στην σελ. 25, θα δουλέψει με βάση τα όρια στάθμης. Στην περίπτωση αυτή της ανατίθεται ένας **αριθμός (Εικ.2.23 - 2)**. Ο αριθμός αυτός, δείχνει με ποιο όριο στάθμης θα ξεκινήσει να δουλεύει η αντλία. Για παράδειγμα στην εικόνα 2.23 όταν η στάθμη των λυμάτων αρχίσει να ανεβαίνει και ενεργοποιηθεί το όριο «Εκκίνηση 1<sup>ης</sup> Αντλίας», τότε θα ξεκινήσει να δουλεύει η αντλία C γιατί έχει τον αριθμό 1. Εάν η στάθμη συνεχίσει να ανεβαίνει και ενεργοποιηθεί το όριο «Εκκίνηση 2<sup>ης</sup> Αντλίας», τότε θα ξεκινήσει να δουλεύει η αντλία B γιατί έχει τον αριθμό 2. Εάν αυτές οι αντλίες δεν επαρκούν και η στάθμη συνεχίζει να ανεβαίνει και ενεργοποιηθεί το όριο «Εκκίνηση 3<sup>ης</sup> Αντλίας», τότε θα ξεκινήσει να δουλεύει η αντλία A γιατί έχει τον αριθμό 3. Κάθε αντλία που έχει ξεκινήσει να δουλεύει θα συνεχίσει μέχρι να απενεργοποιηθεί το όριο «Σταμάτημα Αντλιών».



Εικόνα 2.23

Ο αριθμός που θα έχει κάθε αντλία, εξαρτάται από τα συνολικά λεπτά που έχει δουλέψει αυτή (Εικ.2.23 - 1). Όσο λιγότερα λεπτά έχει δουλέψει μια αντλία τόσο μικρότερο αριθμό θα έχει, επομένως θα ξεκινάει πιο πριν από τις άλλες.

Οι αριθμοί των αντλιών υπολογίζονται κάθε φορά που οι αντλίες σταματούν, δηλαδή όταν απενεργοποιηθεί το όριο «Σταμάτημα Αντλιών».

Μια αντλία χάνει τον αριθμό της όταν:

- Συμβεί κάποιο σφάλμα σε αυτή (σφάλμα επιτηρητή, θερμικής προστασίας, επιβεβαίωσης ρελέ).
- Ο διακόπτης του ηλεκτρολογικού πίνακα δεν είναι πλέον στη θέση AUTO (Εικ.2.13 και 2.14)
- Ο διακόπτης επιλογής λειτουργίας της αντλίας από το SCADA δεν είναι πλέον στη θέση AUTO (Εικ.2.20 και 2.21)

Εάν μια αντλία χάσει τον αριθμό της τότε υπάρχουν δύο περιπτώσεις:

- 1) Εάν είναι απενεργοποιημένο το όριο «Σταμάτημα Αντλιών»



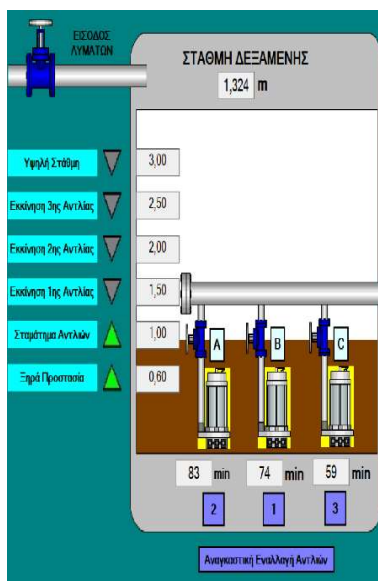
## «Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Στην περίπτωση αυτή οι αριθμοί των άλλων αντλιών υπολογίζονται ξανά με βάση τα συνολικά λεπτά λειτουργίας τους. Εάν η αντλία κάποια στιγμή έχει τις προϋποθέσεις να ξαναπάρει αριθμό, τότε υπολογίζονται πάλι οι αριθμοί των αντλιών με τον ίδιο τρόπο.

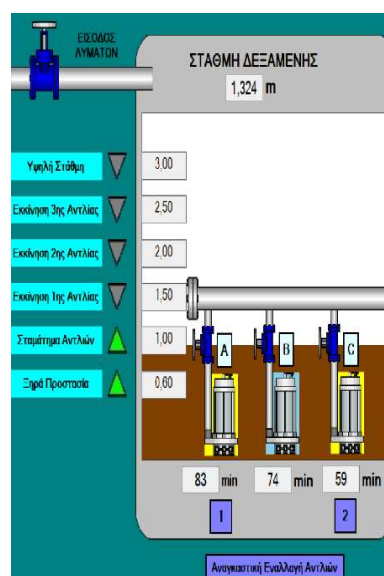
### 2) Εάν είναι ενεργοποιημένο το όριο «Σταμάτημα Αντλιών»

Στην περίπτωση αυτή οι αριθμοί των άλλων αντλιών ενημερώνονται, χωρίς όμως να χάσουν τη σειρά τους οι αντλίες. Για παράδειγμα στην **εικόνα 2.24**, εάν η αντλία B χάσει τον αριθμό της τότε οι αριθμοί των αντλιών A και C ενημερώνονται (**εικόνα 2.25**), δεν αλλάζει όμως η σειρά που είχαν (δηλαδή δεν ξαναυπολογίζονται). Εάν ισχύσουν οι συνθήκες για να πάρει πάλι η αντλία B αριθμό (**εικόνα 2.26**), τότε μπαίνει τελευταία στη σειρά.

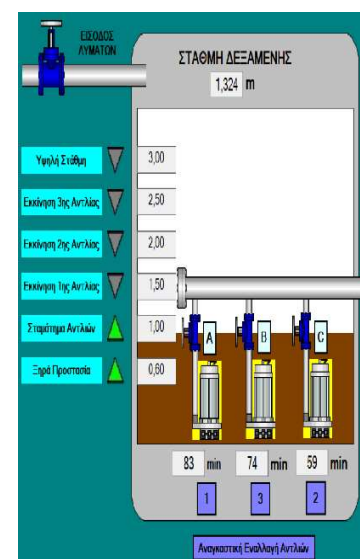
Η μεθοδολογία αυτή εφαρμόζεται γιατί όταν είναι ενεργοποιημένο το όριο «Σταμάτημα Αντλιών» είναι πολύ πιθανό μία ή περισσότερες αντλίες να δουλεύουν, οπότε θέλουμε να διαταράξουμε τη λειτουργία του αντλιοστασίου όσο το δυνατόν λιγότερο.



Εικόνα 2.24



Εικόνα 2.25



Εικόνα 2.26

### Αναγκαστική Εναλλαγή Αντλιών

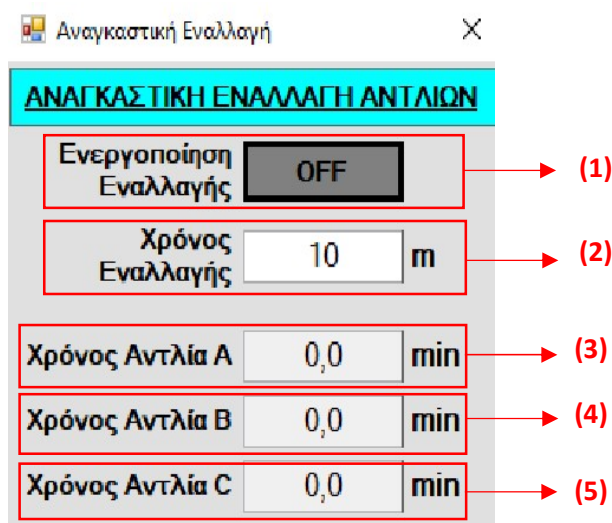
Κατά την αυτόματη λειτουργία των αντλιών μπορεί να προκύψει η εξής κατάσταση:

- 1) Να έχει ενεργοποιηθεί το όριο «Εκκίνηση 1<sup>ης</sup> Αντλίας» και να δουλεύει η πρώτη σε σειρά αντλία.
- 2) Η ποσότητα των λυμάτων που εισέρχονται στη δεξαμενή να είναι ίδια με την ποσότητα που εξέρχεται από αυτή.

Στην περίπτωση αυτή θα δουλεύει συνέχεια η αντλία που έχει ξεκινήσει πρώτη. Για να μην συμβαίνει αυτό υπάρχει η δυνατότητα της «Αναγκαστικής Εναλλαγής Αντλιών». Όταν αυτή είναι ενεργοποιημένη τότε σε συγκεκριμένο χρόνο γίνεται ξανά υπολογισμός της σειράς των αντλιών με βάση τα συνολικά λεπτά λειτουργίας τους. Με τον τρόπο αυτό δεν θα δουλεύει συνέχεια η ίδια αντλία.

Για να μεταβούμε στη δυνατότητα αυτή πατάμε πάνω στην αντίστοιχη ταμπέλα (Εικ.2.23 - 3) οπότε εμφανίζεται το πλαίσιο «ΑΝΑΓΚΑΣΤΙΚΗ ΕΝΑΛΛΑΓΗ ΑΝΤΛΙΩΝ» (Εικ.2.27) στο οποίο βλέπουμε τα παρακάτω:

- Τη δυνατότητα ενεργοποίησης ή απενεργοποίησης της αναγκαστικής εναλλαγής (Εικ.2.27 - 1)
- Τον χρόνο στον οποίο θα συμβεί η αναγκαστική εναλλαγή. (Εικ.2.27 - 2)
- Τον τρέχον χρόνο λειτουργίας κάθε αντλίας (Εικ.2.27 – 3,4,5) , οπότε ξέρουμε πόσος χρόνος υπολείπεται για την αναγκαστική εναλλαγή.



Εικόνα 2.27

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

### Ρυθμίσεις αναλογικών αισθητηρίων

Όπως έχει αναφερθεί οι τιμές των αναλογικών αισθητηρίων (στάθμης και παροχής) προσομοιώνονται. Στην περίπτωση όμως που ο αυτοματισμός εφαρμοστεί σε πραγματικό αντλιοστάσιο, τότε θα πρέπει να υπάρχει η δυνατότητα να διαβαστούν οι τιμές των αισθητηρίων από την αναλογική κάρτα του PLC.

Στη συνέχεια θα δούμε πως μπορεί να παραμετροποιηθεί η μέτρηση της στάθμης της δεξαμενής μέσα από το SCADA. Παρόμοια διαδικασία πραγματοποιείται και για τη μέτρηση της παροχής εξόδου.

Πατώντας πάνω στην ένδειξη της στάθμης (Εικ.2.17 – 6) εμφανίζεται το παράθυρο διαλόγου «ΡΥΘΜΙΣΕΙΣ ΑΙΣΘΗΤΗΡΙΟΥ ΣΤΑΘΜΗΣ» στο οποίο βλέπουμε τα παρακάτω:

- Την ψηφιακή τιμή της αναλογικής κάρτας, δηλαδή τον ακέραιο αριθμό στον οποίο μετατρέπει η κάρτα το ηλεκτρικό σήμα που διαβάζει από το αισθητήριο. (Εικ.2.28 – 1)
- Το κάτω και άνω όριο της ψηφιακής τιμής της αναλογικής κάρτας (οι τιμές μέσα σε αυτά τα όρια θεωρούνται φυσιολογικές) (Εικ.2.28 – 2,3)
- Το κάτω και άνω όριο στάθμης του αισθητηρίου που έχει επιλεγεί (δηλαδή ελάχιστη και μέγιστη στάθμη που μας δίνει το αισθητήριο στάθμης) (Εικ.2.28 – 4,5)

Με βάση τα πέντε παραπάνω υπολογίζεται η στάθμη της δεξαμενής

- Τη δυνατότητα ενεργοποίησης κάτω και άνω ορίου για την μέτρηση της στάθμης (Εικ.2.28 – 6,7). Εάν είναι ενεργοποιημένα αυτά τα όρια και μας έρθει ψηφιακή τιμή μικρότερη από το κάτω όριο ψηφιακής τιμής ή μεγαλύτερη από το άνω όριο ψηφιακής τιμής ο υπολογισμός της στάθμης δεν ξεφεύγει από τα όρια στάθμης.
- Τη δυνατότητα προσομοίωσης της τιμής της στάθμης (Εικ.2.28 – 8,9)
- Την πληροφορία εάν έχει κοπεί το καλώδιο που συνδέει τον αισθητήρα στάθμης στην αναλογική κάρτα του PLC (Εικ.2.28 – 10)



Εικόνα 2.28

#### Αυτόματη Λειτουργία σε περίπτωση βλάβης αισθητηρίου

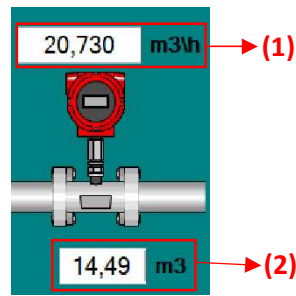
Βλάβη αισθητηρίου υπάρχει όταν:

- 1) Η ψηφιακή τιμή της αναλογικής κάρτας είναι εκτός των ορίων
- 2) Η ψηφιακή τιμή της αναλογικής κάρτας είναι 32767 (που σημαίνει ότι υπάρχει κομμένο καλώδιο)

Στην περίπτωση αυτή ο έλεγχος των αντλιών που βρίσκονται σε αυτόματη λειτουργία γίνεται από τα σήματα των φλοτέρ. Όταν η στάθμη της δεξαμενής ανεβεί και ενεργοποιηθεί το σήμα «Φλοτέρ Εκκίνησης Αντλιών» τότε εκκινούν οι αντλίες (όχι όλες μαζί, αλλά με μια χρονοκαθυστέρηση). Οι αντλίες συνεχίζουν να δουλεύουν και σταματούν όταν απενεργοποιηθεί το σήμα «Φλοτέρ Σταμάτημα Αντλιών»

#### Ημερήσια κυβικά εξόδου

Ανάλογα με την τρέχουσα παροχή εξόδου του αντλιοστασίου (Εικ.2.29 – 1) υπολογίζεται και ο ημερήσιος αριθμός κυβικών (Εικ.2.29 – 2) που εξέρχονται από το αντλιοστάσιο. Η τιμή αυτή καταγράφεται και μηδενίζεται στο τέλος κάθε ημέρας.



**Εικόνα 2.29**

### Καταγραφές

Είναι πολύ σημαντικό να γνωρίζουμε τι συμβαίνει στο αντλιοστάσιο κάθε στιγμή. Για το λόγο αυτό στο πρόγραμμα του SCADA αναπτύχθηκε το κομμάτι των καταγραφών. Τα μεγέθη που καταγράφονται (σε βάση δεδομένων τύπου SQL) είναι τα παρακάτω :

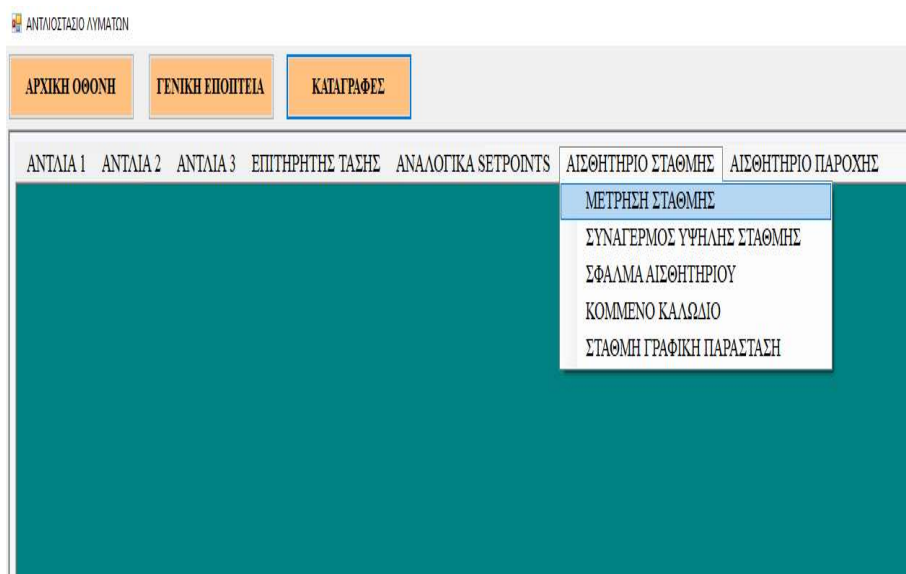
- Για κάθε αντλία:
  - Πότε ξεκίνησε και πότε σταμάτησε η αντλία.
  - Πότε παρουσιάστηκε σφάλμα ρελέ και πότε έγινε αποκατάσταση του.
  - Πότε ενεργοποιήθηκε η θερμική προστασία της αντλίας και πότε επανήλθε.
  - Πότε παρουσιάστηκε σφάλμα θερμικής προστασίας και πότε έγινε επαναφορά.
  - Πότε η αντλία ήταν διαθέσιμη να δουλέψει στην αυτόματη λειτουργία.
  - Ποιος τρόπος λειτουργίας έχει επιλεγεί από τον ηλεκτρολογικό πίνακα.
  - Ποιος τρόπος λειτουργίας έχει επιλεγεί από το SCADA.
- Πότε ο επιτηρητής τάσης ανίχνευσε πρόβλημα στην τάση τροφοδοσίας και πότε το πρόβλημα αποκαταστάθηκε.
  - Πότε παρουσιάστηκε σφάλμα επιτηρητή τάσης και πότε έγινε επαναφορά.
- Οποιαδήποτε αλλαγή έγινε στο όρια της στάθμης.
- Στάθμη δεξαμενής:
  - Η τρέχουσα τιμή της στάθμης.
  - Ο συναγερμός υψηλής στάθμης.
  - Το σφάλμα αισθητηρίου στάθμης.
  - Ανίχνευση κομμένου καλωδίου.
- Παροχή Εξόδου:
  - Η τρέχουσα τιμή της παροχής εξόδου.
  - Η ημερήσια παροχή εξόδου.
  - Το σφάλμα αισθητηρίου παροχής.
  - Ανίχνευση κομμένου καλωδίου.

## «Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

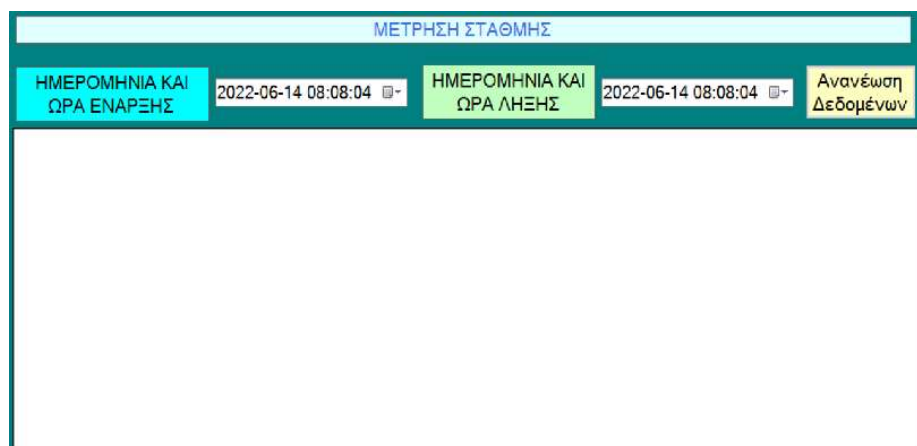
Στην οθόνη των καταγραφών μεταβαίνουμε πατώντας το κουμπί

**ΚΑΤΑΓΡΑΦΕΣ**

Στη συνέχεια θα δούμε ένα παράδειγμα πως μπορούμε να δούμε τις καταγραφές της στάθμης για ένα συγκεκριμένο χρονικό διάστημα. Στην οθόνη των καταγραφών από το μενού «ΑΙΣΘΗΤΗΡΙΟ ΣΤΑΘΜΗΣ» (Εικ.2.30) επιλέγουμε «ΜΕΤΡΗΣΗ ΣΤΑΘΜΗΣ» και εμφανίζεται η οθόνη «ΜΕΤΡΗΣΗ ΣΤΑΘΜΗΣ» (Εικ.2.31). Στην οθόνη αυτή επιλέγουμε ημερομηνία και ώρα έναρξης και λήξης και πατάμε «Ανανέωση Δεδομένων». Εμφανίζεται πλαίσιο που μας πληροφορεί ότι έγινε σύνδεση με τη βάση δεδομένων (Εικ.2.32) και όταν πατήσουμε το μπουτόν ΟΚ εμφανίζονται οι τιμές της στάθμης που έχουν καταγραφεί (Εικ.2.33).

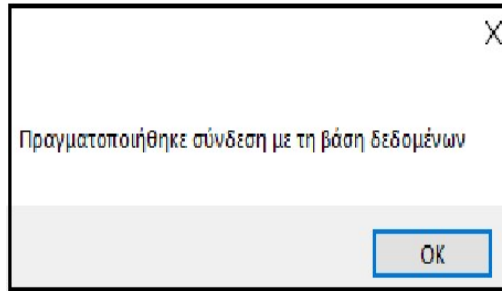


**Εικόνα 2.30**



**Εικόνα 2.31**

# «Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

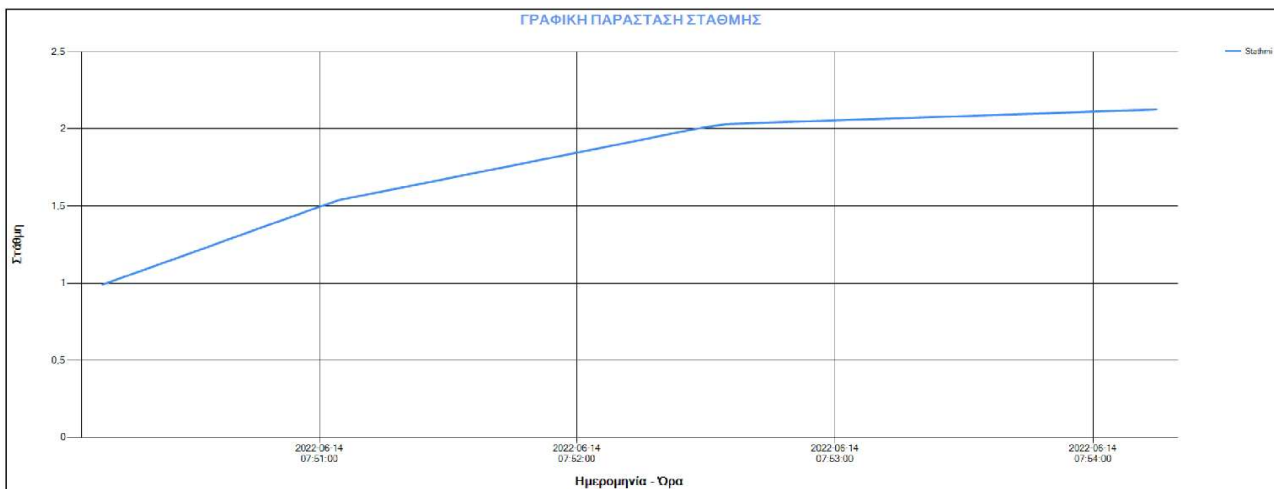


Εικόνα 2.32

ΜΕΤΡΗΣΗ ΣΤΑΘΜΗΣ			
ΗΜΕΡΟΜΗΝΙΑ ΚΑΙ ΩΡΑ ΕΝΑΡΞΗΣ	2022-06-14 07:50:20	ΗΜΕΡΟΜΗΝΙΑ ΚΑΙ ΩΡΑ ΛΗΞΗΣ	2022-06-14 08:12:20
Ανανέωση Δεδομένων			
Ημερομηνία_Ωρα	Στάθμη		
14-06-2022 07:50:24	1,14		
14-06-2022 07:50:29	1,19		
14-06-2022 07:50:34	1,21		
14-06-2022 07:50:39	1,291		
14-06-2022 07:50:44	1,341		
14-06-2022 07:50:49	1,391		
14-06-2022 07:50:54	1,44		
14-06-2022 07:50:59	1,491		
14-06-2022 07:51:04	1,538		
14-06-2022 07:51:09	1,566		
14-06-2022 07:51:14	1,594		
14-06-2022 07:51:19	1,621		
14-06-2022 07:51:24	1,649		
14-06-2022 07:51:29	1,677		
14-06-2022 07:51:34	1,705		
14-06-2022 07:51:39	1,733		
14-06-2022 07:51:44	1,76		
14-06-2022 07:51:49	1,788		
14-06-2022 07:51:54	1,816		
14-06-2022 07:51:59	1,844		
14-06-2022 07:52:04	1,871		
14-06-2022 07:52:09	1,899		
14-06-2022 07:52:14	1,927		
14-06-2022 07:52:19	1,955		
14-06-2022 07:52:24	1,983		
14-06-2022 07:52:29	2,01		
14-06-2022 07:52:34	2,03		
14-06-2022 07:52:39	2,035		
14-06-2022 07:52:44	2,04		
14-06-2022 07:52:49	2,045		
14-06-2022 07:52:54	2,049		

Εικόνα 2.33

Για τις αναλογικές τιμές (στάθμη και παροχή) υπάρχει και η δυνατότητα απεικόνιση τους σε γραφική παράσταση (Εικ.2.34).



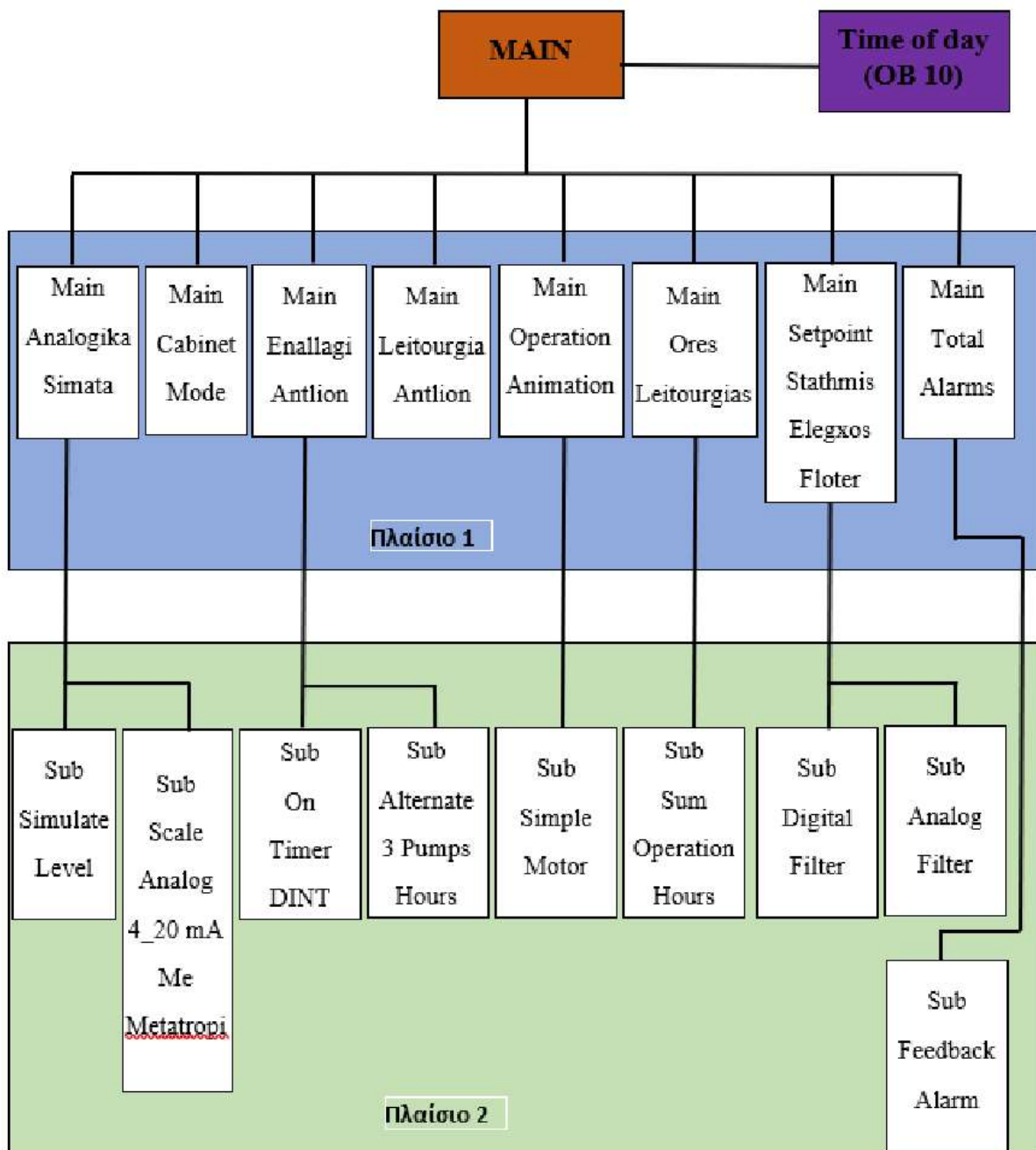
Εικόνα 2.34

### ΚΕΦΑΛΑΙΟ 3 – Ανάλυση προγράμματος PLC

#### Γενικά

Το πρόγραμμα του PLC (S7-1200 CPU 1214C) αναπτύχθηκε με το λογισμικό TIA Portal V15.1 (Step7 Professional) της εταιρίας Siemens χρησιμοποιώντας τη γλώσσα προγραμματισμού LADDER.

Η δομή του προγράμματος φαίνεται στο παρακάτω σχήμα:



Σχήμα 1



«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

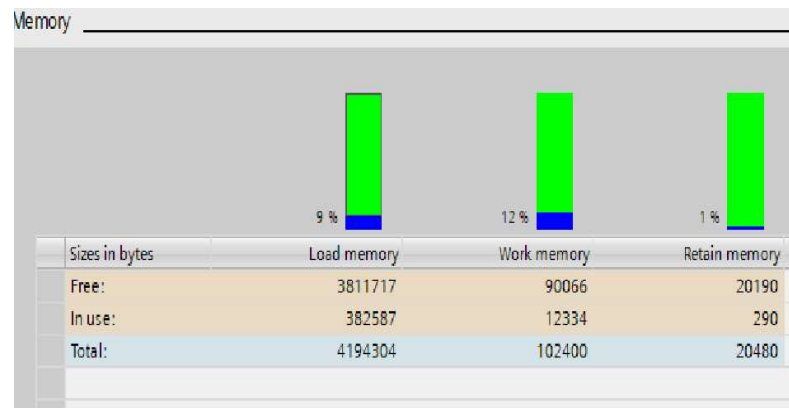
Το PLC καλεί συνεχώς την κυρίως ρουτίνα **MAIN**. Μέσα στη ρουτίνα αυτή καλούνται οι υπορουτίνες που βρίσκονται στο **πλαίσιο 1 – σχήμα 1**. Ορισμένες από τις υπορουτίνες που καλούνται στην MAIN, καλούν με τη σειρά τους άλλες υπορουτίνες όπως φαίνεται στο **πλαίσιο 2 – σχήμα 1**. Υπάρχει και η “διακοπή” **Time of day (OB 10)** που είναι ένα κομμάτι κώδικα που εκτελείται σε συγκεκριμένη χρονική στιγμή παρακάμπτοντας ότι άλλο εκτελείται εκείνη τη στιγμή.

Υπάρχουν ακόμη και πίνακες με θέσεις μνήμης (**data blocks**) που χρησιμοποιούν οι υπορουτίνες για να αποθηκεύσουν τιμές σε αυτές και να τις ανακαλέσουν όποτε χρειαστεί. Παρακάτω φαίνονται όλα τα **data blocks** που χρησιμοποιούνται στο πρόγραμμα του PLC

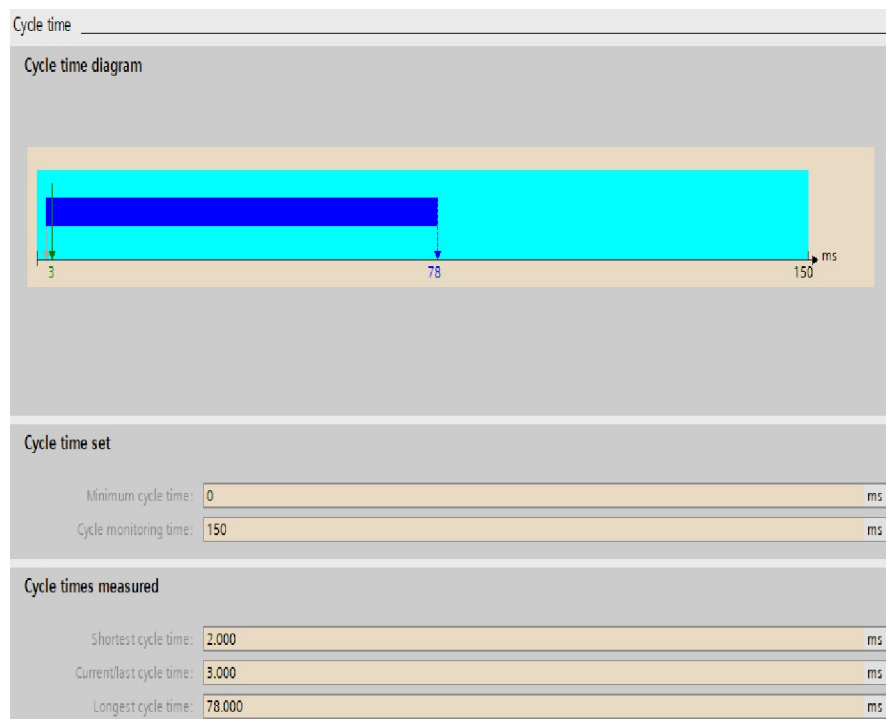
DB Analog Filter	DB Analog Filter <b>Retain</b>	DB Diathesimotita Antlion	DB Digital Filter	DB Digital Filter <b>Retain</b>	DB Feedback Alarm	DB Feedback Alarm <b>Retain</b>
DB Force Alternate Pumps	DB Force Alternate Pumps	DB Inputs Outputs	DB Inputs Outputs <b>Retain</b>	DB Pump Number	DB Scale 4_20mA	DB Scale 4_20mA <b>Retain</b>
DB Simple Motor	DB Simple Motor Scada Mode	DB Simulate Level Flow	DB Sum Operation Hours	DB Sum Operation Hours Reset		
DB Acknowledge	DB Alarms	DB Reset Alarms	Timer_1	Timer_2	Timer_3	Wait To Write Timer

## «Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Παρόλο που ο αριθμός των υπορουτίνων και των data block είναι μεγάλος, το ποσοστό μνήμης του PLC που τελικά χρησιμοποιήθηκε είναι μικρό (**Εικόνα 3.1**). Αυτό συνέβη γιατί η οργάνωση του κώδικα σε υπορουτίνες μειώνει σε μεγάλο βαθμό την επαναληψιμότητα σε αυτόν. Επίσης ο κύκλος του PLC είναι μικρός και μέσα στα επιτρεπτά όρια (**Εικόνα 3.2**). Επειδή ο κώδικας του PLC είναι μεγάλος για να προβληθεί όπως έχει αναπτυχθεί, θα γίνει μια περιγραφική αναφορά στις υπορουτίνες που τον αποτελούν. Για ορισμένες από αυτές θα γίνει πλήρης επεξήγηση του κώδικα.



**Εικόνα 3.1**



**Εικόνα 3.2**

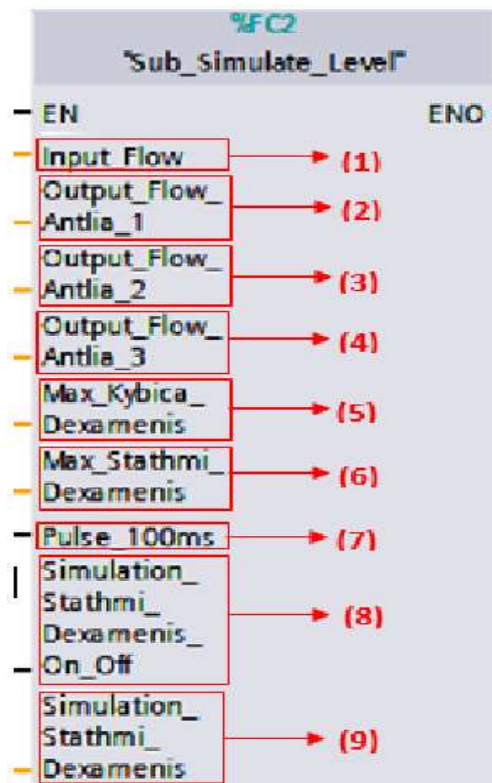
### Υπορουτίνα Main\_Analogika\_Simata

Στην υπορουτίνα αυτή πραγματοποιούνται:

- Η προσομοίωση της εισαγωγής και εξαγωγής λυμάτων στη δεξαμενή του αντλιοστασίου.
- Η μέτρηση της στάθμης της δεξαμενής.
- Η μέτρηση της παροχής εξόδου.
- Η μέτρηση της ημερήσιας παροχής εξόδου.

Για να πραγματοποιηθούν τα παραπάνω χρησιμοποιούνται σε συνδυασμό και με άλλο κώδικα οι υπορουτίνες **Sub\_Simulate\_Level** (1 φορά) και **Sub\_Scale\_Analog\_4\_20\_mA\_Me\_Metatropi** (2 φορές – 1 για τη στάθμη και 1 για την παροχή εξόδου).

### Υπορουτίνα Sub\_Simulate\_Level



Εικόνα 3.3

## «Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

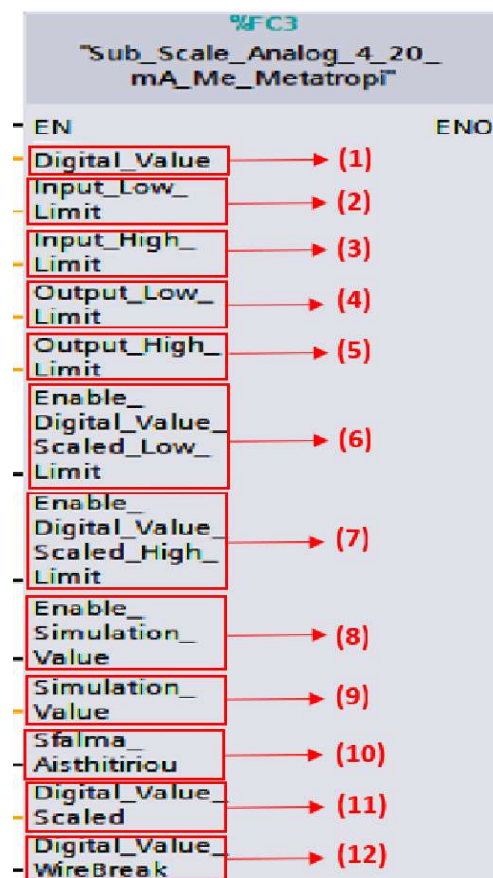
Η υπορουτίνα αυτή χρησιμοποιείται για να γίνει προσομοίωση της στάθμης της δεξαμενής. Ενεργοποιείται με την επιλογή (8) από το SCADA και υπολογίζει τη στάθμη λαμβάνοντας υπόψη τα παρακάτω:

- 1) Την παροχή εισόδου στη δεξαμενή (1)
- 2) Την παροχή εξόδου των τριών αντλιών (2)-(3)-(4)
- 3) Τα μέγιστα κυβικά που χωράει η δεξαμενή (5)
- 4) Τη μέγιστη στάθμη της δεξαμενής (6)

(1) - (2) - (3) - (4) - (5) – (6) δίνονται από το SCADA

Ανάλογα με την ποσότητα των λυμάτων που προσομοιώνουμε ότι εισέρχονται στη δεξαμενή και τον αριθμό των αντλιών που δουλεύουν, η υπορουτίνα υπολογίζει τα κυβικά λυμάτων που παραμένουν στην δεξαμενή και ξέροντας τα μέγιστα κυβικά που χωράει η δεξαμενή και τη μέγιστη στάθμη που μπορεί να έχει, υπολογίζει την τελική στάθμη της δεξαμενής την οποία βλέπουμε γραφικά και σαν τιμή στο SCADA.

### Υπορουτίνα Sub\_Scale\_Analog\_4\_20\_mA\_Me\_Metatropi



Εικόνα 3.4

## «Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Το PLC διαθέτει μια κάρτα 4 αναλογικών εισόδων (SM 1231) στην οποία μπορούν να συνδεθούν το αισθητήριο στάθμης και το αισθητήριο παροχής εξόδου. Οι αισθητήρες αυτοί έχουν ως έξοδο τους ένα σήμα ρεύματος 4...20mA. Η αναλογική κάρτα διαβάζει το σήμα αυτό και το μετατρέπει σε μια ακέραια ψηφιακή τιμή (1). Η υπορουτίνα αυτή χρησιμοποιείται για να γίνει το “**scaling**” της ψηφιακής τιμής και το κάνει λαμβάνοντας υπόψη τα παρακάτω:

- 1) Την κατώτατη και ανώτατη τιμή που μας δίνει η αναλογική κάρτα για το σήμα των 4...20mA (2) – (3)
- 2) Το κατώτατο και ανώτατο όριο του φυσικού μεγέθους στο οποίο αντιστοιχεί το σήμα των 4...20mA (4) – (5)

Για παράδειγμα:

- 1) Αν η κατώτατη τιμή που μας δίνει η αναλογική κάρτα είναι 0 για τα 4mA και η ανώτατη 27648 για 20mA
- 2) Το κάτω όριο του αισθητηρίου στάθμης είναι 0 m και το άνω είναι 4 m

και η κάρτα των αναλογικών εισόδων, από το σήμα 4...20mA που διαβάζει, δώσει την τιμή 12000 τότε η υπορουτίνα θα υπολογίσει τη στάθμη η οποία θα είναι 1,736 m (11).

Υπάρχει η δυνατότητα προσομοίωσης του φυσικού μεγέθους. Αυτό γίνεται με τις εισόδους (8) και (9). Με την είσοδο (8) ενεργοποιείται η προσομοίωση και στην είσοδο (9) βάζουμε την τιμή της προσομοίωσης. Όταν π.χ προσομοιώνουμε τη στάθμη, στην είσοδο (9) μπαίνει η τιμή που υπολογίζεται από την υπορουτίνα `Sub_Simulate_Level`.

Οποιαδήποτε τιμή δώσει η κάρτα εκτός των ορίων (2) – (3) θεωρείτε ότι δεν είναι φυσιολογική, οπότε παράγεται το σφάλμα αισθητηρίου (10). Γνωρίζοντας ότι όταν η κάρτα δώσει την τιμή 32767, έχει ανιχνεύσει κομμένο καλώδιο, παράγεται το σφάλμα κομμένου καλωδίου (12).

Όταν η τιμή που μας δίνει η κάρτα είναι εκτός ορίων, η υπορουτίνα συνεχίζει να υπολογίζει την έξοδο της η οποία θα είναι και αυτή εκτός ορίων του φυσικού μεγέθους. Για το λόγο αυτό υπάρχουν οι επιλογές (6) – (7) με τις οποίες περιορίζεται η έξοδος της υπορουτίνας. Για παράδειγμα εάν η κάρτα μας δώσει την τιμή 27700, η στάθμη με βάση το παράδειγμα που έχει αναφερθεί παραπάνω, θα έπρεπε να είναι πάνω από 4 m. Ενεργοποιώντας την επιλογή (7) η έξοδος της υπορουτίνας δεν θα ξεπερνάει ποτέ τα 4m.

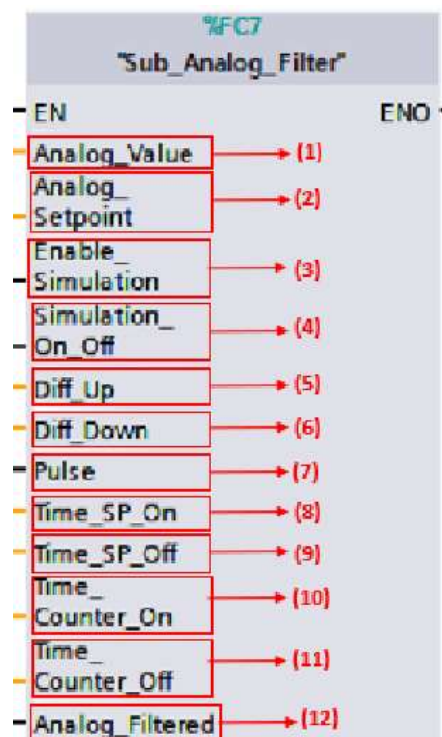
### Υπορουτίνα Main\_Setpoint\_Stathmis\_Elegxos\_Floter

Στην υπορουτίνα αυτή πραγματοποιούνται:

- Έλεγχος των ορίων στάθμης που ορίζονται με βάση το αισθητήριο στάθμης, για τη λειτουργία των αντλιών.
- Έλεγχος των ορίων στάθμης με φάση τα φλοτέρ.

Για να πραγματοποιηθούν τα παραπάνω χρησιμοποιούνται σε συνδυασμό και με άλλο κώδικα οι υπορουτίνες **Sub\_Analog\_Filter** (6 φορές) και **Sub\_Digital\_Filter** (3 φορές).

### Υπορουτίνα Sub\_Analog\_Filter



Εικόνα 3.5

Η υπορουτίνα αυτή χρησιμοποιείται για να υπολογίσει την ενεργοποίηση ή απενεργοποίηση ενός ορίου στάθμης με βάση τη στάθμη της δεξαμενής. Θυμίζουμε αυτά τα όρια στάθμης παίζουν ρόλο στη λειτουργία των αντλιών.

Η υπορουτίνα διαβάζει την στάθμη της δεξαμενής (1) και τη συγκρίνει με το Setpoint που έχουμε ορίσει (2).

Εάν η στάθμη της δεξαμενής (1) γίνει μεγαλύτερη από το άθροισμα Setpoint (2) και Diff\_Up (5) τότε ξεκινάει να μετρά ο χρόνος Time\_Sp\_On (8). Η μέτρηση του χρόνου αυτού φαίνεται

## «Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

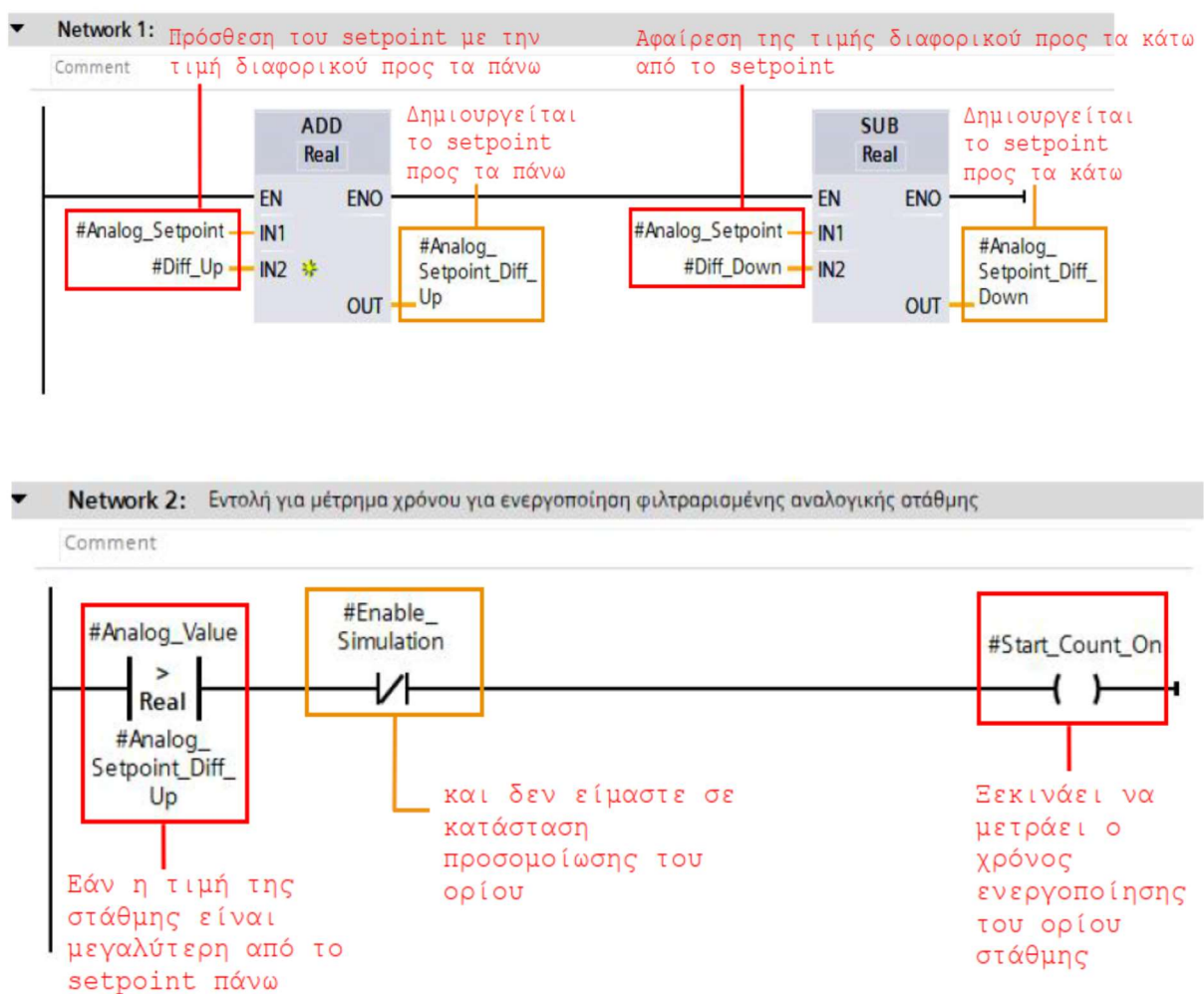
στην είσοδο Time\_Counter\_On (10). Μόλις ο χρόνος αυτός ολοκληρωθεί **ενεργοποιείται** το όριο στάθμης (12).

Εάν η στάθμη της δεξαμενής γίνει μικρότερη ή ίση από τη διαφορά «Setpoint (2) μείον Diff\_Down (6) τότε ξεκινάει να μετρά ο χρόνος που έχει οριστεί στο πεδίο Time\_Sp\_Off (9). Η μέτρηση του χρόνου αυτού φαίνεται στην είσοδο Time\_Counter\_Off (11). Μόλις ο χρόνος αυτός ολοκληρωθεί **απενεργοποιείται** το όριο στάθμης (12).

Με την υπορουτίνα αυτή επιτυγχάνουμε τα δημιουργήσουμε όρια στάθμης αποφεύγοντας πιθανή κυμάτωση των λυμάτων μέσα στη δεξαμενή.

Υπάρχει και η δυνατότητα προσομοίωσης ενός ορίου στάθμης, ανεξάρτητα από τη στάθμη της δεξαμενής. Αυτό γίνεται με τις εισόδους (3)-(4).

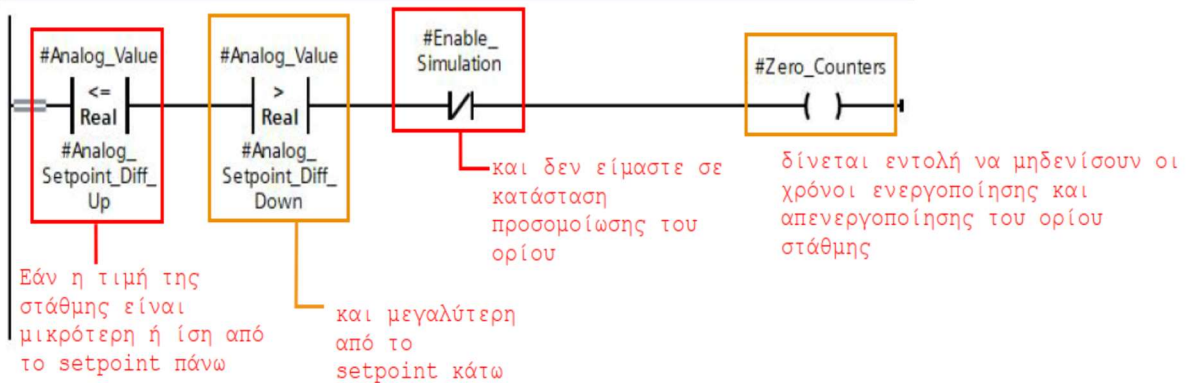
### Ανάλυση Κώδικα Υπορουτίνας



«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

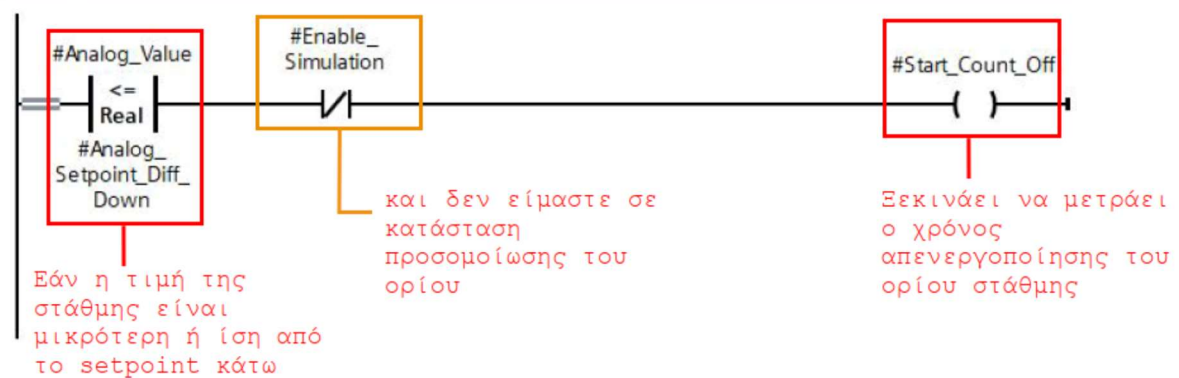
Network 3: Εντολή για μηδενισμό των μετρητών

Comment



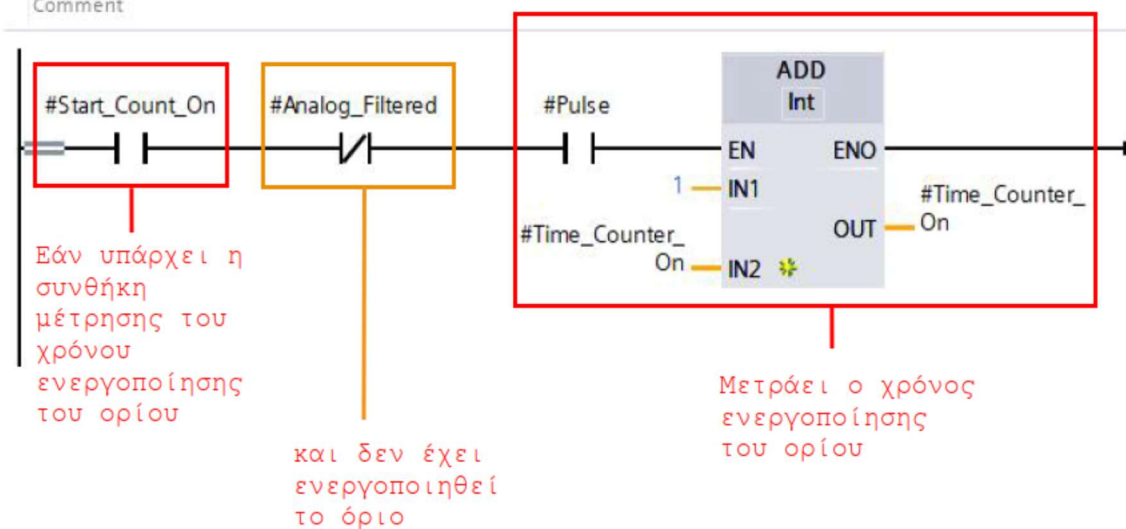
Network 4: Εντολή για μέτρηση χρόνου για απενεργοποίηση φιλτραρισμένης αναλογικής στάθμης

Comment



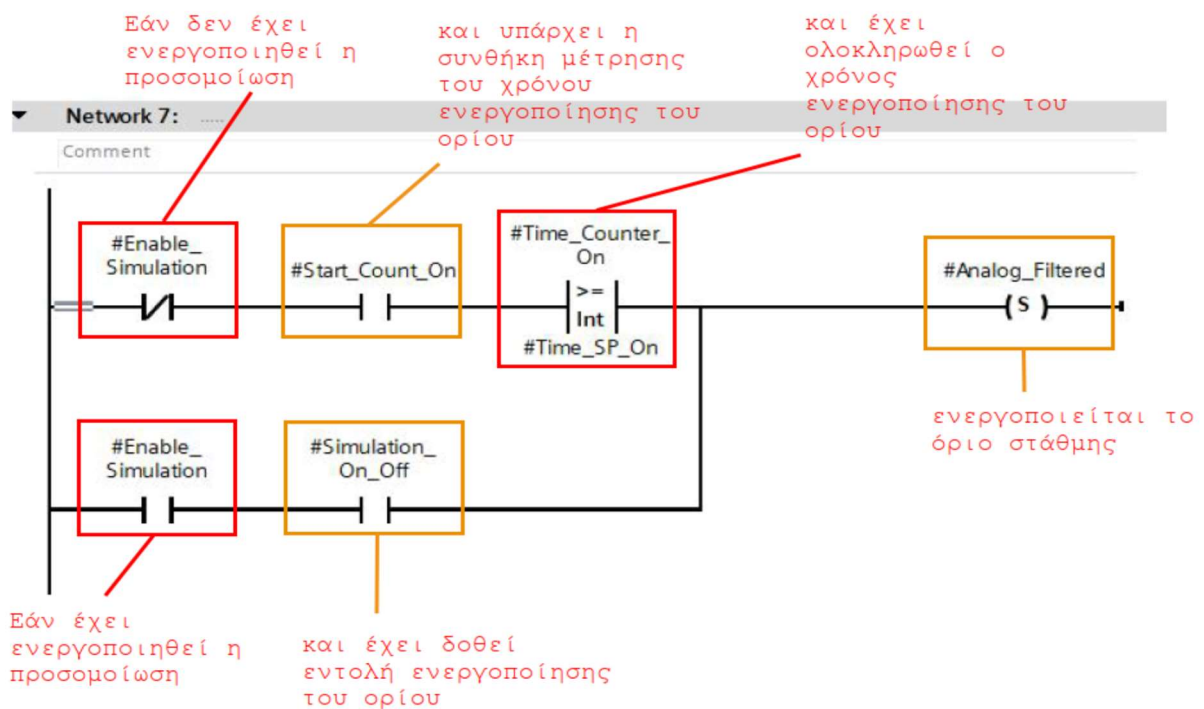
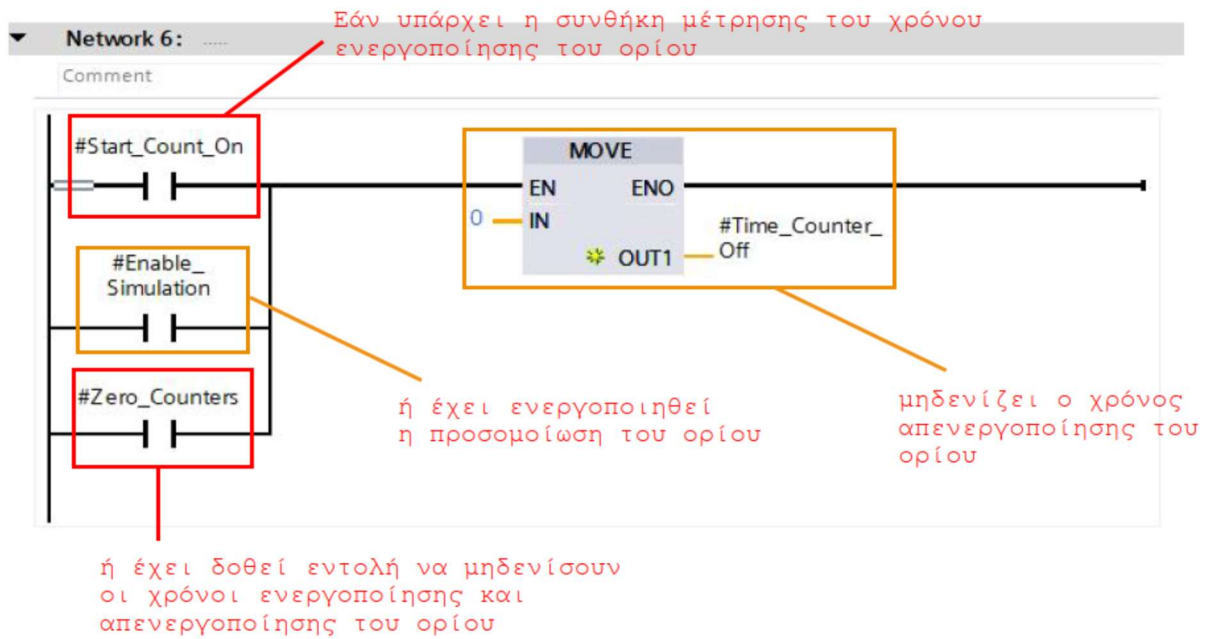
Network 5: Digital Contact Activated

Comment

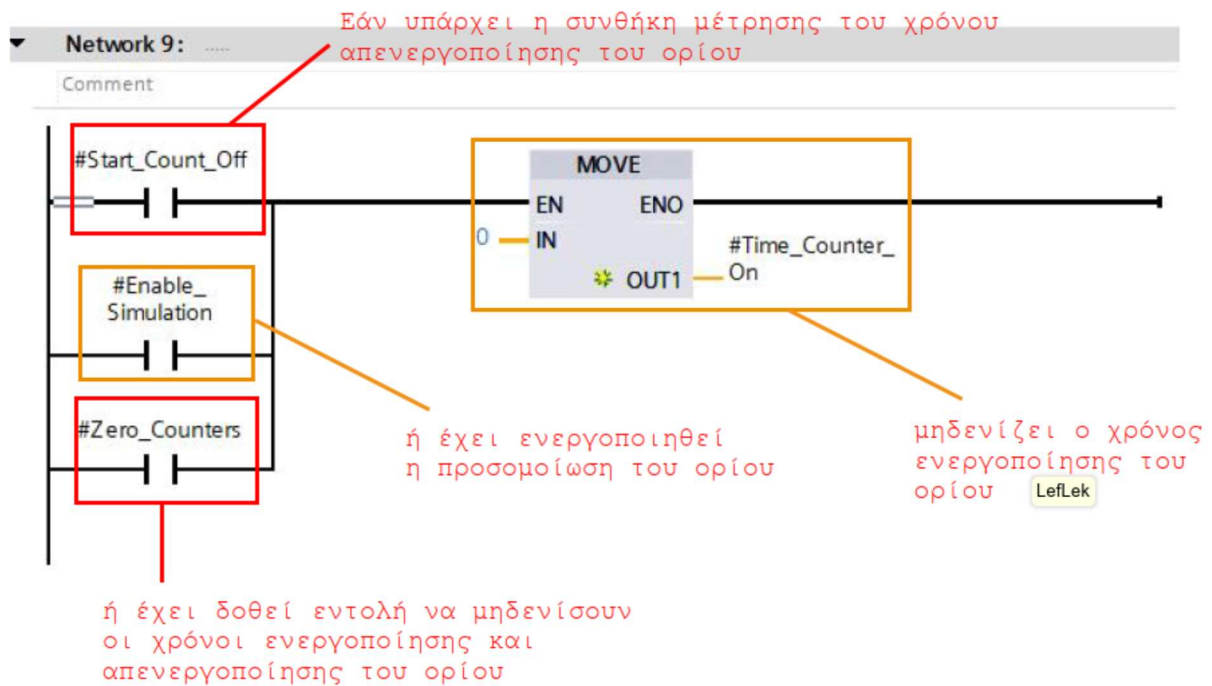
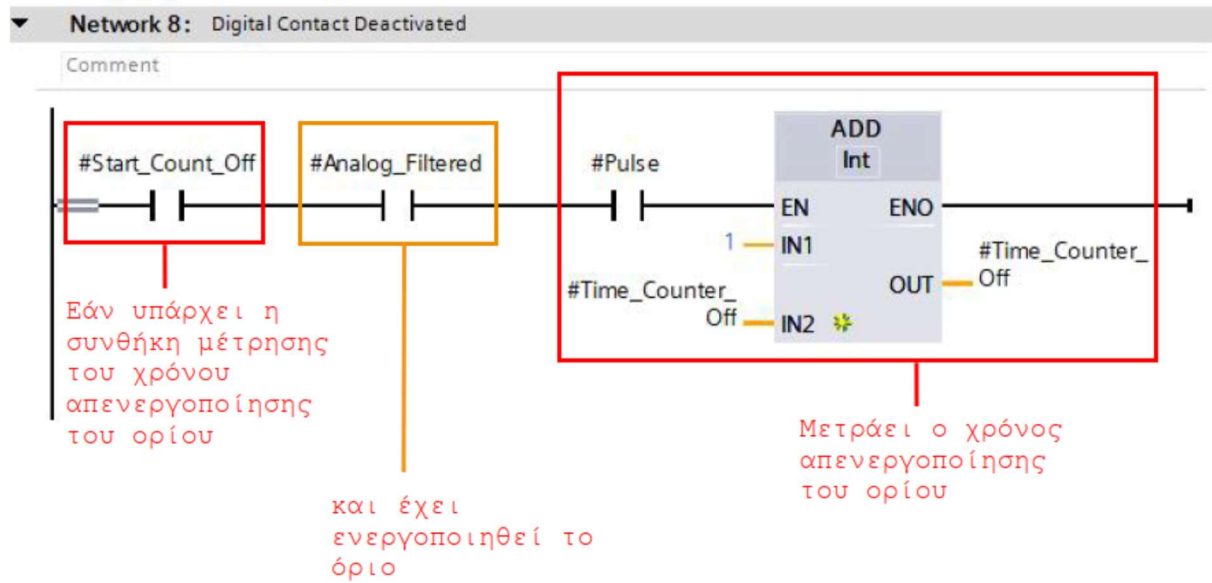




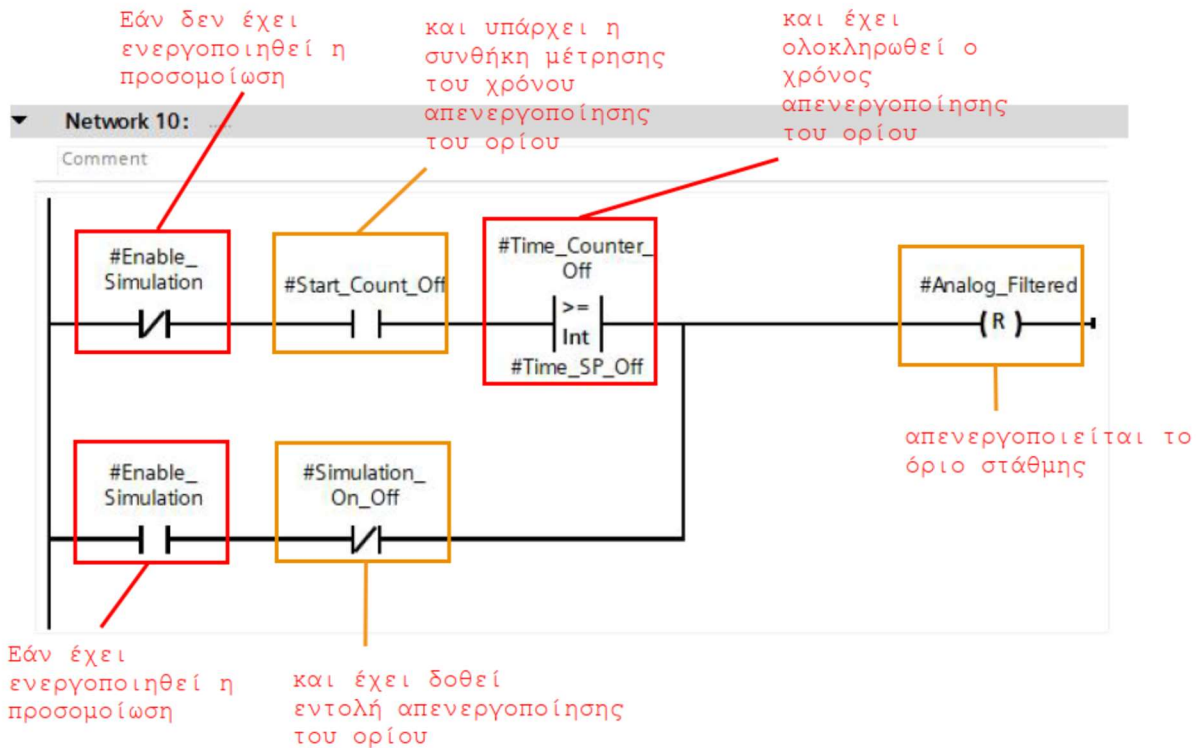
«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»



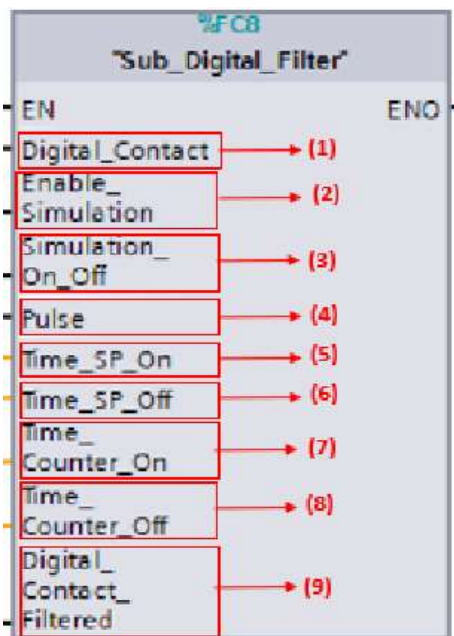
«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»



«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»



Υπορουτίνα Sub\_Digital\_Filter



**Εικόνα 3.6**

Η υπορουτίνα αυτή χρησιμοποιείται για να υπολογίσει την ενεργοποίηση ή απενεργοποίηση ενός ορίου στάθμης με βάση ένα φλοτέρ. Θυμίζουμε αυτά τα όρια στάθμης παίζουν ρόλο στη λειτουργία των αντλιών όταν υπάρχει βλάβη στο αισθητήριο στάθμης.

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Η υπορουτίνα διαβάζει την είσοδο του PLC, στην οποία είναι συνδεδεμένο το φλοτέρ (1).

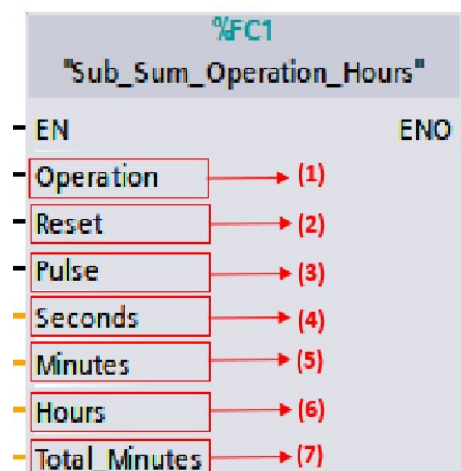
Εάν το φλοτέρ δίνει σήμα στην είσοδο, ξεκινάει να μετρά ο χρόνος Time\_Sp\_On (5). Η μέτρηση του χρόνου αυτού φαίνεται στην είσοδο Time\_Counter\_On (7). Μόλις ο χρόνος αυτός ολοκληρωθεί **ενεργοποιείται** το όριο στάθμης (9).

Εάν το φλοτέρ σταματήσει να δίνει σήμα στην είσοδο, ξεκινάει να μετρά ο χρόνος Time\_Sp\_Off (6). Η μέτρηση του χρόνου αυτού φαίνεται στην είσοδο Time\_Counter\_Off (8). Μόλις ο χρόνος αυτός ολοκληρωθεί **απενεργοποιείται** το όριο στάθμης (9).

### Υπορουτίνα Main\_Ores\_Leitourgias

Στην υπορουτίνα αυτή πραγματοποιείται η μέτρηση του χρόνου λειτουργίας των αντλιών. Για το σκοπό αυτό χρησιμοποιείται η υπορουτίνα **Sub\_Sum\_Operation\_Hours** (3 φορές). Ο χρόνος λειτουργίας των αντλιών επηρεάζει την κυκλική εναλλαγή τους, όταν αυτές βρίσκονται στην αυτόματη λειτουργία.

### Υπορουτίνα Sub\_Sum\_Operation\_Hours



**Εικόνα 3.7**

Η υπορουτίνα ανιχνεύει εάν η αντλία δουλεύει (στην είσοδο (1) συνδέεται το σήμα από την επαφή επιβεβαίωσης του ρελέ ισχύος της αντλίας) και μετράει το χρόνο λειτουργίας της αντλίας. Υπολογίζει τα δευτερόλεπτα (4), τα λεπτά (5) και οι ώρες λειτουργίας (6). Στη συνέχεια υπολογίζει και τα συνολικά λεπτά (7). Τα συνολικά λεπτά είναι που λαμβάνονται υπόψη στην κυκλική εναλλαγή των αντλιών. Υπάρχει και η δυνατότητα μηδενισμού του χρόνου λειτουργίας με την είσοδο Reset (2).

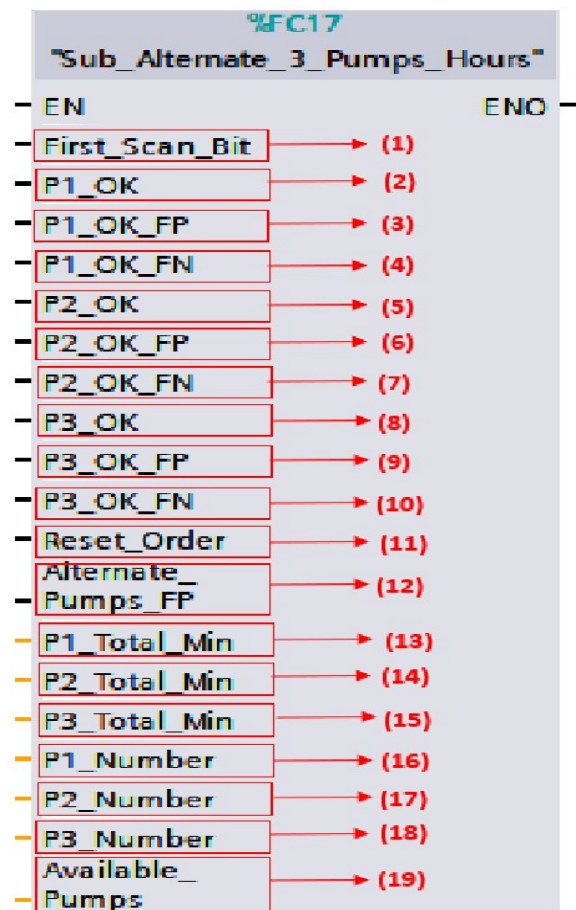
### Υπορουτίνα Main\_Enallagi\_Antlion

Στην υπορουτίνα αυτή πραγματοποιούνται:

- Η κυκλική εναλλαγή των αντλιών στην αυτόματη λειτουργία.
- Η αναγκαστική εναλλαγή των αντλιών όταν αυτή έχει επιλεγεί.

Για να πραγματοποιηθούν τα παραπάνω χρησιμοποιούνται σε συνδυασμό και με άλλο κώδικα οι υπορουτίνες **Sub\_Alternate\_3\_Pumps\_Hours** και **Sub\_On\_Timer\_DINT** (3 φορές).

### Υπορουτίνα Sub\_Alternate\_3\_Pumps\_Hours



Εικόνα 3.8

Η υπορουτίνα αυτή αποτελεί την κύρια ρουτίνα του αυτοματισμού του αντλιοστασίου. Πρόκειται για μια σύνθετη ρουτίνα η οποία υπολογίζει τον αριθμό που ανατίθεται σε κάθε αντλία (16)-(17)-(18). Ο αριθμός αυτός, δείχνει με πιο όριο στάθμης θα ξεκινήσει να δουλεύει η αντλία. Ο αριθμός που θα έχει κάθε αντλία, εξαρτάται από τα συνολικά λεπτά που έχει

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

δουλέψει αυτή (13)-(14)-(15). Όσο λιγότερα λεπτά έχει δουλέψει μια αντλία τόσο μικρότερο αριθμό θα έχει, επομένως θα ξεκινάει πιο πριν από τις άλλες.

Η υπορουτίνα ανιχνεύει την διαθεσιμότητα μιας αντλίας (Αντλία 1 - (2)-(3)-(4), Αντλία 2 - (5)-(6)-(7), Αντλία 3 - (8)-(9)-(10)) και ανάλογα με τα συνολικά λεπτά λειτουργίας και την κατάσταση της αντλίας, της αναθέτει τον αριθμό που της αντιστοιχεί ή τον αφαιρεί από αυτή.

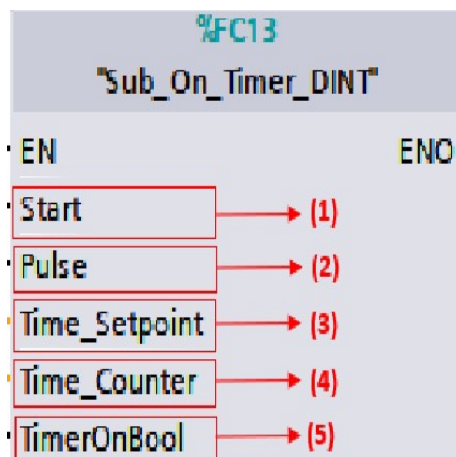
Μια αντλία λέμε ότι είναι διαθέσιμη όταν:

- 1) Ο διακόπτης επιλογής MANUAL – 0 – AUTO του ηλεκτρολογικού πίνακα βρίσκεται στη θέση AUTO.
- 2) Δεν υπάρχει κάποιο σφάλμα στην αντλία (σφάλμα επιτηρητή, σφάλμα θερμικής προστασίας, σφάλμα επιβεβαίωσης ρελέ).
- 3) Έχει επιλεγεί αυτόματος τρόπος λειτουργίας της αντλίας από το SCADA.

Η λειτουργία της υπορουτίνας γίνεται κατανοητή με βάση την περιγραφή που έγινε στο Κεφ. 2 στην ενότητα «Αυτόματη Λειτουργία» στη σελ. 31.

Η υπορουτίνα δίνει επίσης τη δυνατότητα να γίνει επαναφορά της σειράς των αντλιών, ανεξάρτητα από το χρόνο λειτουργίας τους, χρησιμοποιώντας την είσοδο Reset Order (11).

#### Υπορουτίνα Sub\_On\_Timer\_DINT



**Εικόνα 3.9**

Η υπορουτίνα αυτή μετράει τον χρόνο αναγκαστικής εναλλαγής των αντλιών (Κεφ. 2, Ενότητα «Αναγκαστική Εναλλαγή Αντλιών», Σελ. 34). Πρώτα ελέγχει την συνθήκη αναγκαστικής εναλλαγής (1) και εάν ισχύει αυτή τότε μετράει χρόνο (4). Εάν ο χρόνος αυτός γίνει μεγαλύτερος από το Setpoint που έχει οριστεί (3), τότε η υπορουτίνα ενεργοποιεί την

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

έξοδο TimerOnBool (5) η οποία χρησιμοποιείται από τον υπόλοιπο κώδικα για να γίνει η αναγκαστική εναλλαγή των αντλιών στην οποία συμμετέχει και η υπορουτίνα Sub\_Alternate\_3\_Pumps\_Hours .

#### Υπορουτίνα Main\_Leitourgia\_Antlion

Η υπορουτίνα αυτή υπολογίζει την **τελική συνθήκη** έναρξης μιας αντλίας. Για να το κάνει αυτό συνδυάζει τις παρακάτω πληροφορίες:

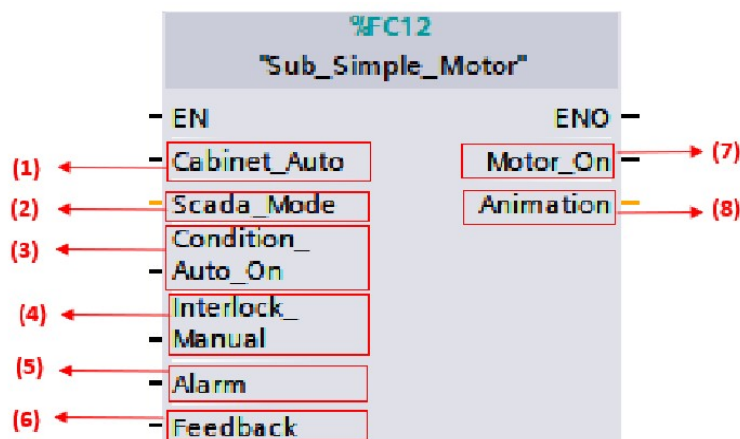
- Εάν για την αντλία έχει επιλεγεί αυτόματη ή χειροκίνητη λειτουργία.
- Εάν η αντλία βρίσκεται σε σφάλμα ή όχι.
- Τον αριθμό που έχει πάρει η αντλία.
- Τη στάθμη της δεξαμενής και πιο όρια έχει ενεργοποιήσει αυτή.
- Εάν υπάρχει σφάλμα αισθητηρίου στάθμης.
- Τα όρια στάθμης με βάση τα φλοτέρ που έχουν ενεργοποιηθεί.

#### Υπορουτίνα Main\_Operation\_Animation

Η υπορουτίνα αυτή ενεργοποιεί τις **τελικές εντολές** προς τις αντλίες και υπολογίζει έναν αριθμό με τον οποίο μεταβάλλεται το χρώμα τους στο Scada για να δείξει σε ποια κατάσταση βρίσκονται (Κεφ. 2, Γραφική απεικόνιση του τρόπου λειτουργίας και της κατάστασης της αντλίας, Σελ. 30) .

Για να πραγματοποιηθεί το παραπάνω χρησιμοποιείται σε συνδυασμό και με άλλο κώδικα η υπορουτίνα **Sub\_Simple\_Motor** (3 φορές).

#### Υπορουτίνα Sub\_Simple\_Motor



Εικόνα 3.10

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Η υπορουτίνα αυτή ενεργοποιεί την **τελική εντολή** προς την αντλία (7) και υπολογίζει τον αριθμό (8) που θα καθορίσει το χρώμα της στο Scada ανάλογα με την κατάσταση στην οποία βρίσκεται .

Για τον υπολογισμό των παραπάνω λαμβάνονται υπόψη:

- Σε ποια κατάσταση βρίσκεται ο επιλογικός διακόπτης του ηλεκτρολογικού πίνακα. (1)
- Ποιος τρόπος λειτουργίας έχει επιλεγεί από το SCADA. (2)
- Η τελική συνθήκη που έχει υπολογιστεί από την υπορουτίνα Main\_Leitourgia\_Antlion. (3)
- Εάν υπάρχει κάποιο σφάλμα στην αντλία. (5)
- Η επιβεβαίωση από το ρελέ ισχύος της αντλίας. (6)

Υπάρχει και η είσοδος Interlock\_Manual (4) στην οποία βάζουμε συνθήκες που θέλουμε να σταματήσουν σίγουρα την αντλία και είναι οι παρακάτω:

- Εάν δεν υπάρχει σφάλμα στο αισθητήριο στάθμης – το όριο «Ξηρά Προστασία»
- Εάν υπάρχει σφάλμα στο αισθητήριο στάθμης – το όριο «Φλοτέρ Ξηρά Προστασία»
- Το σφάλμα «Επιτηρητή Τάσης»

#### Υπορουτίνα Main\_Total Alarms

Η υπορουτίνα αυτή υπολογίζει όλα πιθανά σφάλματα που μπορεί να υπάρξουν στο αντλιοστάσιο τα οποία είναι τα παρακάτω:

- Σφάλμα επιβεβαίωσης ρελέ ισχύος αντλίας (3)
- Σφάλμα θερμικής προστασίας αντλίας (3)
- Σφάλμα επιτηρητή τάσης
- Σφάλμα αισθητηρίου στάθμης
- Σφάλμα αισθητηρίου παροχής
- Σφάλμα υψηλής στάθμης

Η φιλοσοφία ενός σφάλματος είναι η εξής:

Εάν υπάρχει η συνθήκη του σφάλματος, τότε ενεργοποιείται το σφάλμα. Εάν δεν υπάρχει πλέον η συνθήκη, το σφάλμα παραμένει μέχρι να κάνει Reset ο χειριστής του Scada για να

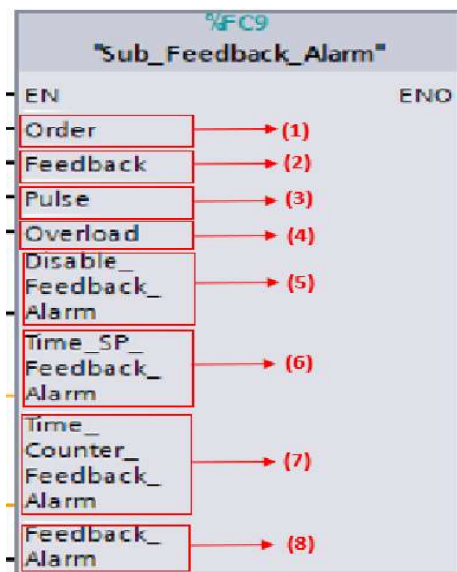


«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

γίνει η αποκατάσταση του σφάλματος. Για παράδειγμα εάν ενεργοποιηθεί η θερμική προστασία (θερμικό) μιας αντλίας τότε ενεργοποιείται το σφάλμα θερμικού. Εάν κάποιος επαναφέρει το θερμικό στον ηλεκτρολογικό πίνακα, το σφάλμα θερμικού παραμένει μέχρι ο χειριστής του Scada να πατήσει το μπουτόν Reset.

Η συνθήκη του σφάλματος υψηλής στάθμης προκύπτει από την ενεργοποίηση του ορίου «Υψηλή Στάθμη», ενώ του σφάλματος επιβεβαίωσης ρελέ από την υπορουτίνα Sub\_Feedback\_Alarm

#### Υπορουτίνα Sub\_Feedback\_Alarm

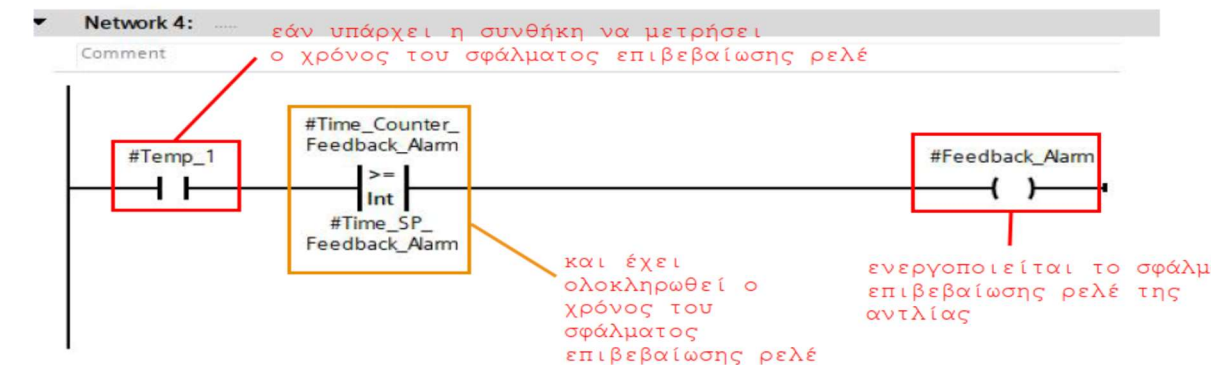
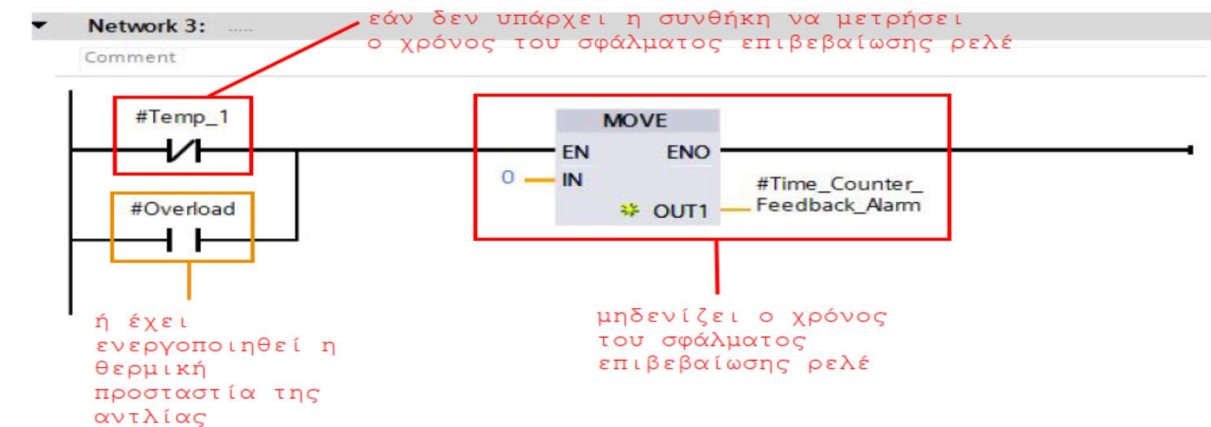
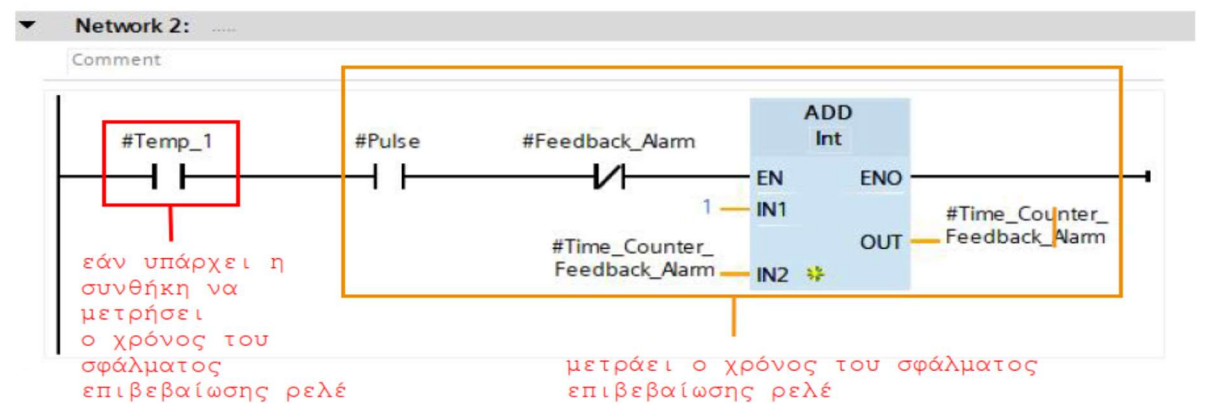
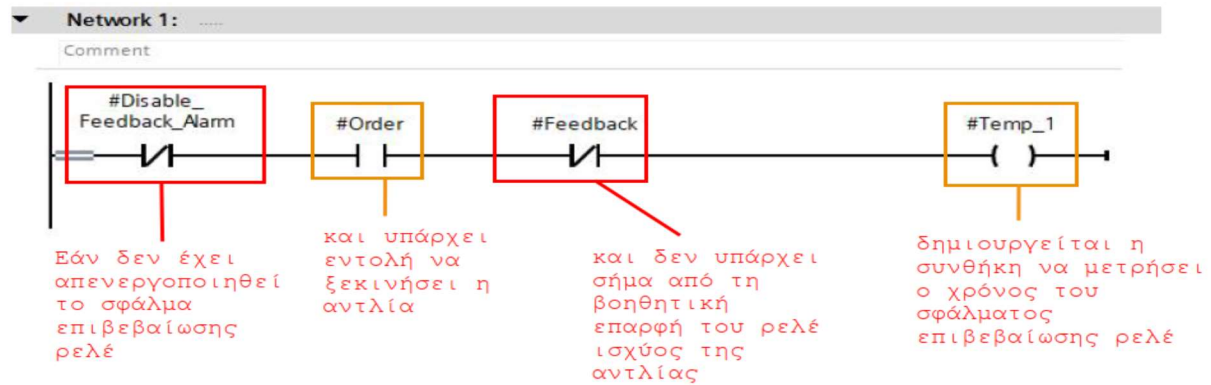


**Εικόνα 3.11**

Η υπορουτίνα αυτή ελέγχει εάν έχει δοθεί εντολή στην αντλία (1) και στην περίπτωση που συμβαίνει αυτό, ελέγχει εάν έχει έρθει το σήμα από την βοηθητική επαφή του ρελέ ισχύος της αντλίας (2). Εάν δεν έχει ενεργοποιηθεί το σήμα αυτό, τότε μετράει τον χρόνο που ορίζεται στην είσοδο (6) (ο οποίος φαίνεται στην είσοδο (7)) και δημιουργεί τη συνθήκη (8) για την ενεργοποίηση του σφάλματος επιβεβαίωσης ρελέ της αντλίας. Σε περίπτωση που έχει ενεργοποιηθεί το θερμικό της αντλίας (4), η υπορουτίνα δε δημιουργεί την συνθήκη σφάλματος. Επίσης μπορεί να γίνει απενεργοποίηση της ρουτίνας χρησιμοποιώντας την είσοδο (5).

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

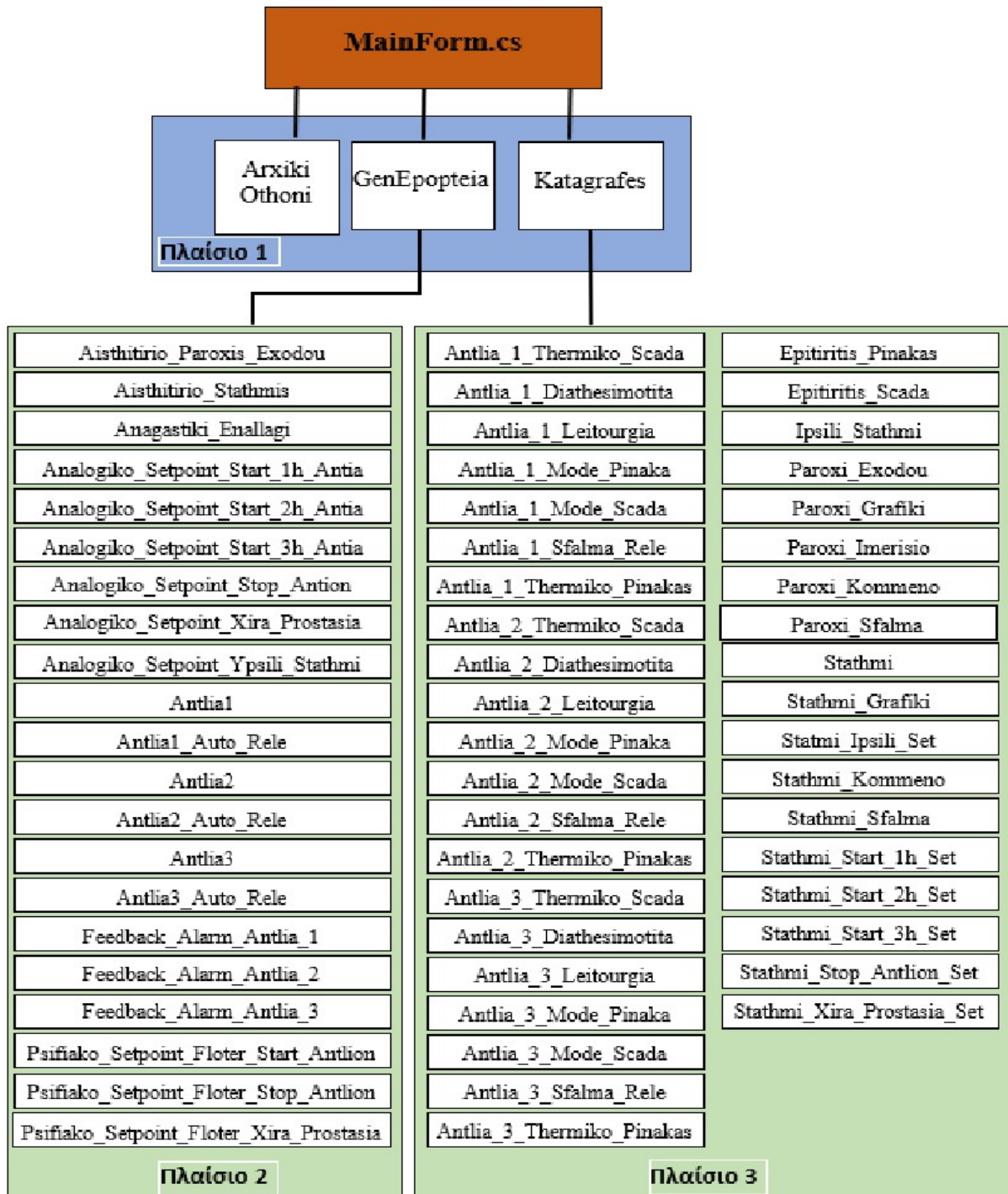
Ανάλυση Κώδικα Υπορουτίνας



## ΚΕΦΑΛΑΙΟ 4 – Ανάλυση προγράμματος SCADA

### Γενικά

Το πρόγραμμα του SCADA αναπτύχθηκε με την αντικειμενοστραφή γλώσσα προγραμματισμού Microsoft C# μέσα από το περιβάλλον Microsoft Visual Studio Community 2019, το οποίο διατίθεται δωρεάν. Η δομή του προγράμματος φαίνεται στο παρακάτω σχήμα:



Σχήμα 2

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Στο κεφάλαιο αυτό θα αναλυθούν πως αναπτύχθηκαν τα βασικά κομμάτια του προγράμματος. Η περιγραφή λειτουργίας του έγινε στο κεφάλαιο 2.

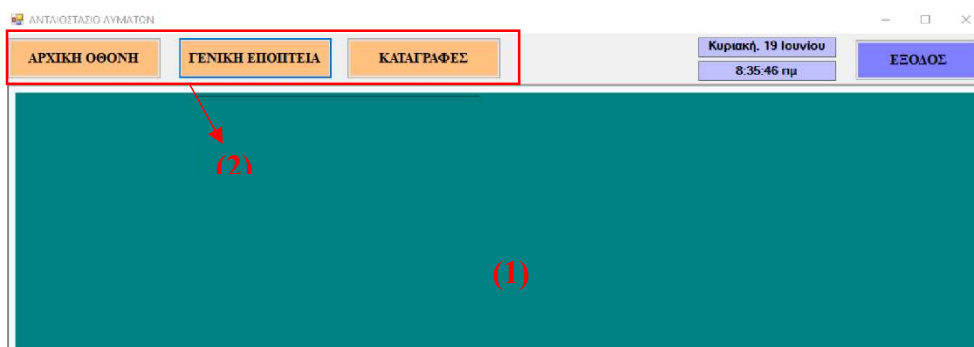
### Οθόνη MainForm

Η οθόνη αυτή είναι η κεντρική οθόνη του SCADA, από την οποία καλούνται οι τρεις κύριες οθόνες (**σχήμα 2 – πλαίσιο 1**). Αυτές με τη σειρά τους καλούν όποτε απαιτηθεί και άλλες οθόνες για τον χειρισμό του αντλιοστασίου (**σχήμα 2 – πλαίσιο 2**) και την εμφάνιση των καταγραφών (**σχήμα 2 – πλαίσιο 3**). Ο κώδικας που τρέχει πίσω από την οθόνη MainForm περιλαμβάνει τα παρακάτω:

- Εμφάνιση των τριών κύριων οθονών.
- Σύνδεση και επικοινωνία με το PLC.
- Διάβασμα τιμών από το PLC.
- Γράψιμο στη βάση δεδομένων SQL.

### Εμφάνιση των τριών κύριων οθονών

Η οθόνη MainForm έχει την παρακάτω μορφή



**Εικόνα 4.1**

Έχει στο κέντρο της ένα “Panel” (**Εικ.4.1 – 1**) στο οποίο ανοίγει η οθόνη που επιθυμούμε πατώντας πάνω στο αντίστοιχο μπουτόν (**Εικ.4.1 – 2**). Για παράδειγμα για να ανοίξει η οθόνη της «ΓΕΝΙΚΗΣ ΕΠΟΠΤΕΙΑΣ» πατώντας πάνω στο μπουτόν της, τρέχει ο παρακάτω κώδικας.

```
private void Btn_Geniki_Eropteia_Click(object sender, EventArgs e)
{
    Main_Panel.Controls.Clear(); (1)

    GenEropteia FrmGenEropteia = new GenEropteia(); (2)

    FrmGenEropteia.TopLevel = false;

    Main_Panel.Controls.Add(FrmGenEropteia); (3)
}
```

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

```
FrmGenEropteia.Show(); (4)
```

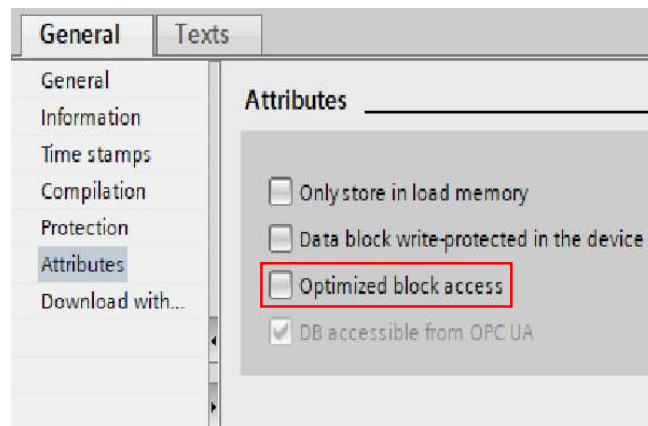
```
}
```

Ο κώδικας αυτός αρχικά αδειάζει το “Panel” (1), στη συνέχεια δημιουργεί ένα στιγμιότυπο της φόρμας «ΓΕΝΙΚΗ ΕΠΙΟΠΤΕΙΑ» (2), το προσθέτει στο “Panel” (3) και τελικά το δείχνει (4).

### Σύνδεση και επικοινωνία με το PLC

Για να επικοινωνήσει το πρόγραμμα του SCADA με το PLC, χρησιμοποιήθηκε η βιβλιοθήκη **Sharp7**. Η βιβλιοθήκη αυτή υλοποιεί με τη γλώσσα προγραμματισμού C#, το πρωτόκολλο επικοινωνίας **S7Protocol** της εταιρίας Siemens. Η βιβλιοθήκη διατίθεται δωρεάν.

Πρέπει να προσεχθεί ότι για να μπορέσει να χρησιμοποιηθεί η βιβλιοθήκη αυτή, πρέπει σε κάθε datablock του PLC να έχει απενεργοποιηθεί η επιλογή **Optimized block access** (Εικ.4.2), το οποίο πρέπει να γίνει κατά τη δημιουργία του datablock.



**Εικόνα 4.2**

Όταν ξεκινάει το SCADA προσπαθεί να πετύχει επικοινωνία με το PLC, οπότε τρέχει τον παρακάτω κώδικα. Εάν η επικοινωνία με το PLC είναι επιτυχής, τότε ξεκινάει να τρέχει ένας timer (GeneralTimer). Ο timer αυτός εκτελείται κάθε 250 ms και μέσα στον χρόνο αυτό γίνεται το διάβασμα των τιμών από τα datablock του PLC. Εάν δεν επιτευχθεί επικοινωνία με το PLC, τότε εμφανίζεται μήνυμα με τον κωδικό σφάλματος επικοινωνίας.

Πρέπει να σημειωθεί ότι ο κώδικας της βιβλιοθήκης Sharp7 επιτυγχάνει πολύ γρήγορο διάβασμα των datablock (όχι πάνω από 180 ms και συνήθως 150 ms).

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

```
int connectionResult = Gvariables.client.ConnectTo("127.0.0.1", 0, 1);

    if (connectionResult == 0)
    {
        GeneralTimer.Enabled = true;
        Analog_Value_Timer.Enabled = true;
    }











    else
    {
        MessageBox.Show("Δεν έγινε σύνδεση - Κωδικός σφάλματος:" +
Convert.ToString(connectionResult) + " Περιγραφή: " +
Gvariables.client.ErrorText(connectionResult));
    }
}
```

### Δήλωση Μεταβλητών

Όταν διαβάζουμε τιμές από το PLC, αυτές πρέπει να αποθηκευτούν σε μεταβλητές για να μπορέσουν να χρησιμοποιηθούν από το πρόγραμμα του SCADA. Η δήλωση των μεταβλητών αυτών γίνεται σε ξεχωριστό αρχείο (Gvariables.cs), ώστε οι μεταβλητές αυτές να είναι καθολικές (global) και να μπορούν να χρησιμοποιηθούν οπουδήποτε μέσα στο πρόγραμμα. Για παράδειγμα η δήλωση των μεταβλητών για το όριο «ΣΤΑΜΑΤΗΜΑ ΑΝΤΛΙΩΝ», φαίνεται παρακάτω.

```
public static float Stop_Antlion_Analog_Setpoint;
public static float Stop_Antlion_Analog_Setpoint_prv;
public static bool Stop_Antlion_Enable_Simulation;
public static bool Stop_Antlion_Simulation_On_Off;
public static float Stop_Antlion_Diff_Up;
public static float Stop_Antlion_Diff_Down;
public static short Stop_Antlion_Time_SP_On;
public static short Stop_Antlion_Time_SP_Off;
```

Η ονομασία των μεταβλητών είναι ακριβώς ίδια με την ονομασία των μεταβλητών του PLC στα αντίστοιχα datablock (Εικ.4.3).

	Stop_Antlion_Time_Counter_On	Int	6.0	0
	Stop_Antlion_Time_Counter_Off	Int	8.0	0
	Stop_Antlion_Analog_Filtered	Bool	10.0	false
	Stop_Antlion_Analog_Setpoint	Real	18.0	1.0
	Stop_Antlion_Enable_Simulation	Bool	22.0	false
	Stop_Antlion_Simulation_On_Off	Bool	22.1	false
	Stop_Antlion_Diff_Up	Real	24.0	0.02
	Stop_Antlion_Diff_Down	Real	28.0	0.02
	Stop_Antlion_Time_SP_On	Int	32.0	2
	Stop_Antlion_Time_SP_Off	Int	34.0	2

Εικόνα 4.3

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

### Ενημέρωση μεταβλητών

Η ενημέρωση των μεταβλητών με τις τιμές του PLC γίνεται σε δύο στάδια.

#### Στάδιο 1

Διαβάζεται κάθε datablock από το PLC και οι τιμές του τοποθετούνται σε έναν πίνακα. Για παράδειγμα το διάβασμα των τιμών του datablock, που αφορά τα όρια στάθμης με βάση το αισθητήριο στάθμης γίνεται με τον παρακάτω κώδικα. Οι τιμές τελικά αποθηκεύονται στον πίνακα DB\_Analog\_Filter.

```
S7MultiVar Reader = new S7MultiVar(Gvariables.client);  
  
byte[] DB_Analog_Filter = new byte[35];  
  
int DBNumber_DB_Analog_Filter = 30;  
  
Reader.Add(S7Consts.S7AreaDB, S7Consts.S7WLByte,  
DBNumber_DB_Analog_Filter, 0, 35, ref DB_Analog_Filter);  
  
int Result = Reader.Read();
```

#### Στάδιο 2

Εξάγεται από τον πίνακα που δημιουργήθηκε η τιμή που θέλουμε να τοποθετήσουμε σε κάθε μεταβλητή. Για παράδειγμα ορισμένες από τις τιμές που αφορούν το όριο «ΞΗΡΑ ΠΡΟΣΤΑΣΙΑ» εξάγονται από τον πίνακα με τον παρακάτω κώδικα.

```
Gvariables.Xira_Prostasia_Time_Counter_On = S7.GetIntAt(DB_Analog_Filter, 0);  
Gvariables.Xira_Prostasia_Time_Counter_Off = S7.GetIntAt(DB_Analog_Filter, 2);  
Gvariables.Xira_Prostasia_Analog_Filtered = S7.GetBitAt(DB_Analog_Filter, 4, 0);
```

### Γράψιμο στη βάση δεδομένων

Πάρα πολλές από τις μεταβλητές του προγράμματος SCADA καταγράφονται σε βάση δεδομένων. Για το σκοπό αυτό χρησιμοποιήθηκε ο SQL Server 2017 της Microsoft. Η έκδοση που χρησιμοποιήθηκε διατίθεται δωρεάν. Στον server δημιουργήθηκε η βάση δεδομένων η οποία αποτελείται από 39 πίνακες.

Οι μεταβλητές καταγράφονται στη βάση όταν συμβεί κάποια αλλαγή στην τιμή τους, εκτός από τις μεταβλητές της στάθμης και της παροχής εξόδου που καταγράφονται περιοδικά (κάθε 5 δευτερόλεπτα).

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Για να καταγράψουμε μια μεταβλητή μόνο όταν μεταβάλλεται η τιμή της, χρησιμοποιείται ο παρακάτω κώδικας. Στο παράδειγμα αυτό βλέπουμε την καταγραφή της θερμικής προστασίας της αντλίας 1.

```
if (Gvariables.Antlia_1_Thermiko != Gvariables.Antlia_1_Thermiko_prv)
{
    string connectionString = "Data Source=DESKTOP- EPVREL3\\SQLEXPRESS;Initial
Catalog=Antliostasio_Lymaton;Integrated Security=True";

    int fc1;

    if (Gvariables.Antlia_1_Thermiko == true)
    {
        fc1 = 1;
    }
    else
    {
        fc1 = 0;
    }

    string sqlQuery1 = " INSERT INTO Antlia1_Thermiko_Pinakas
(Θερμικό_Πίνακας_Bool) VALUES (@fc1) ";

    SqlConnection con = new SqlConnection(connectionString);

    con.Open();


    SqlCommand sc1 = new SqlCommand(sqlQuery1, con);

    sc1.Parameters.AddWithValue("@fc1", fc1);

    sc1.ExecuteNonQuery();

    con.Close();

    Gvariables.Antlia_1_Thermiko_prv = Gvariables.Antlia_1_Thermiko;
}
```



Στον κώδικα αυτό ελέγχεται εάν έχει γίνει αλλαγή της κατάστασης του θερμικού, στη συνέχεια γίνεται σύνδεση με τη βάση δεδομένων και καταγραφή της καινούργιας τιμής στον αντίστοιχο πίνακα. Μόλις γίνει η καταγραφή κλείνει η σύνδεση με τη βάση δεδομένων.

Για να γίνει η σύνδεση με τη βάση δεδομένων πρέπει πρώτα να δημιουργηθεί το connectionString (1) που περιέχει πληροφορίες για τη βάση (δηλαδή που τρέχει η βάση και ποια κριτήρια ασφάλειας έχει αυτή).



«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Πρέπει να σημειωθεί ότι για οποιαδήποτε ενέργεια γίνεται στη βάση (διάβασμα ή γράψιμο), χρησιμοποιούνται SQL commands, δηλαδή εντολές της γλώσσας Structured Query Language (SQL) η οποία αφορά τις βάσεις δεδομένων.

Οι μεταβλητές της στάθμης και της παροχής εξόδου καταγράφονται κάθε 5 δευτερόλεπτα. Για το σκοπό αυτό χρησιμοποιείται ο παρακάτω κώδικας.

```
private void Analog_Value_Timer_Tick(object sender, EventArgs e)
{
    string connectionString = "Data Source=DESKTOP-EPVREL3\\SQLEXPRESS;Initial
Catalog=Antliostasio_Lymaton;Integrated Security=True";

    float fc1 = (float)Math.Round(Gvariables.Stathmi_Digital_Value_Sca * 1000f)
/ 1000f;

    float fc2 = (float)Math.Round(Gvariables.Flow_Digital_Value_Scaled * 1000f)
/ 1000f;

    string sqlQuery1 = " INSERT INTO Stathmi (Στάθμη) VALUES (@fc1) ";
    string sqlQuery2 = " INSERT INTO Paroxi_Exodou (Παροχή_Εξόδου) VALUES
(@fc2) ";

    SqlConnection con = new SqlConnection(connectionString);
    con.Open();

    SqlCommand sc1 = new SqlCommand(sqlQuery1, con);
    SqlCommand sc2 = new SqlCommand(sqlQuery2, con);

    sc1.Parameters.AddWithValue("@fc1", fc1);
    sc2.Parameters.AddWithValue("@fc2", fc2);

    sc1.ExecuteNonQuery();
    sc2.ExecuteNonQuery();

    con.Close();
}
```

Ο κώδικας αυτός εκτελείται κάθε 5 δευτερόλεπτα, με βάση τον timer Analog\_Value\_Timer και δουλεύει με παρόμοιο τρόπο με την καταγραφή των υπόλοιπων μεταβλητών. Η διαφορά είναι ότι επειδή μια float τιμή που διαβάζεται από το PLC έχει πολλά δεκαδικά ψηφία, ο κώδικας στρογγυλοποιεί την τιμή της στάθμης και της παροχής εξόδου σε ακρίβεια 3 δεκαδικών ψηφίων.

### Οθόνη GenEpopteia

Στην οθόνη αυτή γίνεται απεικόνιση του αντλιοστασίου και μεταβολή των παραμέτρων λειτουργίας του. Τα κύρια στοιχεία που περιέχει η οθόνη αυτή είναι:

- Μπουτόν (Button)
- Πεδία τιμών (NumericUpDown)
- Πεδία εικόνων (Picture Box)
- Λεζάντες (Label)
- Controls από τη βιβλιοθήκη SymbolFactoryDotNet

Στη συνέχεια θα παρουσιαστεί ο τρόπος λειτουργίας των παραπάνω στο πρόγραμμα του SCADA.

Όταν ενεργοποιείται η οθόνη τότε ξεκινάει να τρέχει ένας timer (TimerRead). Ο timer αυτός εκτελείται κάθε 250 ms και μέσα στον χρόνο αυτό γίνεται ενημέρωση των στοιχείων της οθόνης με βάση τις τιμές που έχουν διαβαστεί από το PLC.

### Μπουτόν Προσομοίωσης

Όπως έχουμε δει υπάρχουν ορισμένα μπουτόν που προσομοιώνουν τα ηλεκτρικά σήματα του αντλιοστασίου. Στη συνέχεια θα δούμε το μπουτόν προσομοίωσης της θερμικής προστασίας της αντλίας 1, για να καταλάβουμε ένα από αυτά. Τα υπόλοιπα λειτουργούν με παρόμοιο τρόπο. Ο κώδικας είναι ο παρακάτω.

```
if (Gvariables.Antlial_1_Thermiko == true)
{
    btnAtherm.BackColor = Color.Red;
    btnAtherm.Text = "ON";
}
else
{
    btnAtherm.BackColor = Color.Lime;
    btnAtherm.Text = "OFF";
}
```

(A)

```
private void btnAtherm_Click(object sender, EventArgs e) (B)
{
    TimerRead.Enabled = false;

    bool a;

    if (Gvariables.Antlia_1_Thermiko == true)
    {
        a = false;
    }
    else
    {
        a = true;
    }

    var writebuffer = new byte[1];
    writebuffer[0] = Gvariables.DB_Inputs_Outputs_Retain_0;
    S7.SetBitAt(ref writebuffer, 0, 2, a);
    int Result = Gvariables.client.DBWrite(12, 0, writebuffer.Length,
writebuffer);

    TimerRead.Enabled = true;
}
```

Στο τμήμα (A) του κώδικα ελέγχεται η κατάσταση του θερμικού της αντλίας. Εάν το θερμικό είναι ενεργοποιημένο το μπουτόν γίνεται κόκκινο και γράφει “ON” ενώ εάν είναι απενεργοποιημένο γίνεται πράσινο και γράφει “OFF”.

Στο τμήμα (B) του κώδικα γίνεται η προσομοίωση του θερμικού κάθε φορά που πατάμε το μπουτόν. Για να γίνει αυτό πρέπει να σταματήσει ο timer TimerRead (βλέπε σελ. 61). Στη συνέχεια ελέγχεται η τρέχουσα κατάσταση του θερμικού και γράφεται η αντίθετη. Χρησιμοποιείται η εντολή SetBitAt της βιβλιοθήκης Sharp7. Αφού τελειώσει η εγγραφή γίνεται επανεκκίνηση του timer.

### Πεδία τιμών (NumericUpDown)

Στα πεδία τιμών εμφανίζονται αριθμητικές τιμές που διαβάζονται από το PLC. Με ορισμένα πεδία μπορούμε να γράψουμε και τιμές στο PLC. Στη συνέχεια θα δούμε το πεδίο «ΚΥΒΙΚΑ ΕΙΣΟΔΟΥ» το οποίο κάνει και ανάγνωση και εγγραφή. Ο κώδικας είναι ο παρακάτω.

```
KybicaEisodou.Value = (decimal)Gvariables.Input_Flow; → (A)
```

```
private void KybicaEisodou_ValueChanged(object sender, EventArgs e)
{
    TimerRead.Enabled = false;
    var writebuffer = new byte[4];
    S7.SetRealAt(writebuffer, 0, (float)KybicaEisodou.Value);
    int Result = Gvariables.client.DBWrite(15, 0, writebuffer.Length,
writebuffer);
    TimerRead.Enabled = true;
}
```

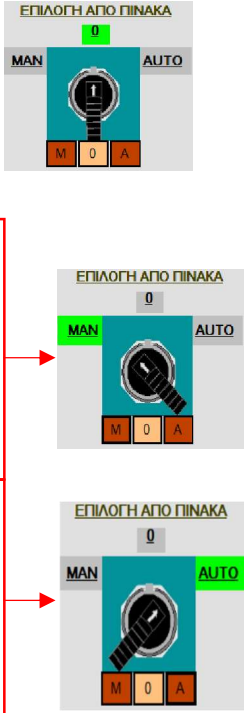
Στο τμήμα (A) ενημερώνεται η τιμή του πεδίου με την τιμή της μεταβλητής της παροχής εισόδου. Θυμίζουμε ότι η ενημέρωση της μεταβλητής γίνεται συνέχεια με τον timer που ενεργοποιήθηκε στη MainForm.

Στο τμήμα (B) του κώδικα φαίνεται τι γίνεται εάν κάποιος κάνει κλικ μέσα στο πεδίο, αλλάζει την τιμή του και πατήσει το πλήκτρο ENTER. Όταν κάνει κλικ μέσα στο πεδίο δεν γίνεται ακόμη τίποτα. Εάν αλλάξει την τιμή του και πατήσει ENTER, τότε ενεργοποιείται το event της αλλαγής τιμής του πεδίου και εκτελείται ο κώδικας μέσα σε αυτό. Στον κώδικα πρώτα σταματάει ο timer TimerRead και στη συνέχεια γράφεται στο PLC η καινούργια τιμή του πεδίου. Χρησιμοποιείται η εντολή SetRealAt της βιβλιοθήκης Sharp7. Αφού τελειώσει η εγγραφή γίνεται επανεκκίνηση του timer.

### Πεδία εικόνων (Picture Box)

Στα πεδία εικόνων εμφανίζονται διαφορετικές εικόνες ανάλογα με τις τιμές που διαβάζονται από το PLC. Στη συνέχεια θα δούμε το πεδίο που μας δείχνει την κατάσταση στην οποία βρίσκεται ο διακόπτης του ηλεκτρολογικού πίνακα (MANUAL – 0 – AUTO) μιας αντλίας. Παρόμοιο είναι και το πεδίο εικόνας που μας δείχνει τον τρόπο λειτουργίας που έχει επιλεγεί από το SCADA για την αντλία. Ο κώδικας είναι ο παρακάτω.

```
if (Gvariables.Antlia_1_Cabinet_Mode == 0)
{
    SwitchCabinetA.Image = Properties.Resources.Switch0;
    LbSwCaAzer.BackColor = Color.Lime;
    LbSwCaAman.BackColor = Color.Silver;
    LbSwCaAaut.BackColor = Color.Silver;
}
else if (Gvariables.Antlia_1_Cabinet_Mode == 1)
{
    SwitchCabinetA.Image = Properties.Resources.SwitchManual;
    LbSwCaAzer.BackColor = Color.Silver;
    LbSwCaAman.BackColor = Color.Lime;
    LbSwCaAaut.BackColor = Color.Silver;
}
else
{
    SwitchCabinetA.Image = Properties.Resources.SwitchAuto;
    LbSwCaAzer.BackColor = Color.Silver;
    LbSwCaAman.BackColor = Color.Silver;
    LbSwCaAaut.BackColor = Color.Lime;
}
```




Ελέγχεται η μεταβλητή `Antlia_1_Cabinet_Mode` και ανάλογα με την τιμή που έχει αυτή, τοποθετείται στο πεδίο εικόνας και η ανάλογη εικόνα, που βρίσκεται στα resources του προγράμματος SCADA.

Πεδία εικόνας χρησιμοποιούνται για να δείξουν και την ενεργοποίηση των ορίων στάθμης.

### Λεζάντες (Label)

Οι λεζάντες είναι κείμενο με την κατάλληλη μορφοποίηση που μας πληροφορεί για ένα στοιχείο του αντλιοστασίου ή για μια κατάσταση που συμβαίνει σε αυτό. Υπάρχουν λεζάντες που εμφανίζονται συνεχώς στην οθόνη και λεζάντες που εμφανίζονται ανάλογα με κάποια τιμή που διαβάζεται από το PLC. Στη συνέχεια θα δούμε τη λεζάντα που μας δείχνει τη βλάβη αισθητηρίου. Ο κώδικας είναι ο παρακάτω.

```
if (Gvariables.Sfalma_Aisthitiriou_Stathmis_Alm == true)
{
    ResBlabiStathmi.Visible = true;
    LbBlabiStathmi.Visible = true;
}
else
{
    ResBlabiStathmi.Visible = false;
    LbBlabiStathmi.Visible = false;
}
```



«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Ελέγχεται η μεταβλητή Sfalma\_Aisthitiριου\_Stathmis\_Alm και ανάλογα με την τιμή που έχει αυτή, εμφανίζεται η λεζάντα «ΒΛΑΒΗ ΑΙΣΘΗΤΗΡΙΟΥ».

Μαζί όπως βλέπουμε εμφανίζεται και το μπουτόν “Reset” με το οποίο γίνεται αποκατάσταση του σφάλματος βλάβης αισθητηρίου, όταν δεν υπάρχουν πλέον συνθήκες να υπάρχει το σφάλμα αυτό. Παρόμοιο μπουτόν χρησιμοποιείται για όλα τα σφάλματα που συμβαίνουν στο αντλιοστάσιο.

### Μπουτόν Reset

Το μπουτόν “Reset” των σφαλμάτων εμφανίζει μια ιδιαιτερότητα σε σχέση με τα μπουτόν προσομοίωσης.

Όταν πατάμε ένα μπουτόν προσομοίωσης αυτό πρέπει να γράψει σε ένα bit του PLC την τιμή **true** ή την τιμή **false**. Όταν όμως πατάμε το μπουτόν “Reset” πρέπει να γραφεί σε ένα bit του PLC η τιμή **true** και όταν αφήσουμε το μπουτόν να γραφεί η τιμή **false**. Αυτό επιτυγχάνεται για παράδειγμα στο μπουτόν “Reset” του σφάλματος αισθητηρίου στάθμης, με τον παρακάτω κώδικα που είναι παρόμοιος με τον κώδικα των μπουτόν προσομοίωσης.

```
private void ResBlabiStathmi_MouseDown(object sender, MouseEventArgs e)
{
    TimerRead.Enabled = false;

    var writebuffer = new byte[1];
    writebuffer[0] = Gvariables.DB_Reset_Alarms_0; ;
    S7.SetBitAt(ref writebuffer, 0, 5, true);
    int Result = Gvariables.client.DBWrite(17, 0, writebuffer.Length,
    writebuffer);

    TimerRead.Enabled = true;
}
```

**Πατάω το μπουτόν**

```
private void ResBlabiStathmi_MouseUp(object sender, MouseEventArgs e)
{
    TimerRead.Enabled = false;

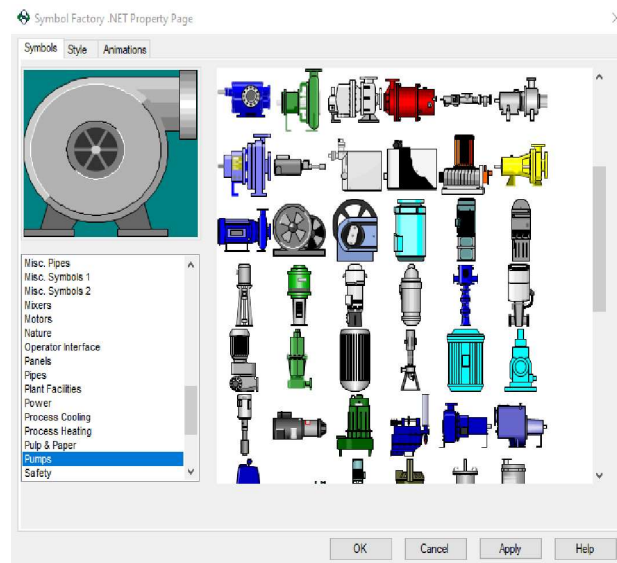
    var writebuffer = new byte[1];
    writebuffer[0] = Gvariables.DB_Reset_Alarms_0; ;
    S7.SetBitAt(ref writebuffer, 0, 5, false);
    int Result = Gvariables.client.DBWrite(17, 0, writebuffer.Length,
    writebuffer);

    TimerRead.Enabled = true;
}
```

**Αφήνω το μπουτόν**

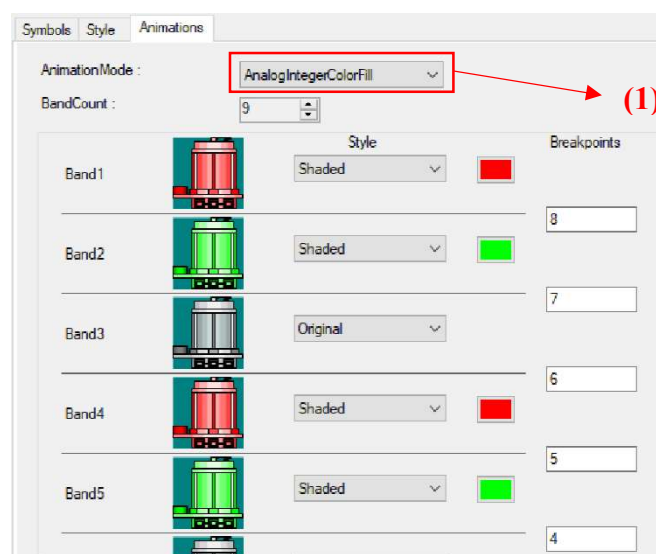
### Animation Αντλιών

Όπως έχει αναφερθεί η απεικόνιση μιας αντλίας αλλάζει χρώμα ανάλογα με την κατάσταση της αντλίας. Αυτό επιτυγχάνεται χρησιμοποιώντας ένα **StandardControl** (εικόνα 4.5) της βιβλιοθήκης **SymbolFactoryDotNet** (εικόνα 4.4) και τον ανάλογο κώδικα. Η βιβλιοθήκη αυτή έχει έτοιμα βιομηχανικά σύμβολα (π.χ κινητήρες, αντλίες) που μπορούμε να εισάγουμε στις οθόνες και να μεταβάλλουμε τις ιδιότητες τους με κώδικα.



Εικόνα 4.4

Στο παρακάτω παράδειγμα φαίνεται πως γίνεται η αλλαγή χρωμάτων της αντλίας 1. Πρώτα εισάγουμε το **StandardControl** της αντλίας από τη βιβλιοθήκη και ορίζουμε πως θα αλλάζει το χρώμα της (εικόνα 4.5)



Εικόνα 4.5

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Το χρώμα θα αλλάξει δίνοντας την κατάλληλη τιμή στην ιδιότητα `AnalogIntegerColorFill` (εικόνα 4.5 - 1) και αλλάζοντας το `BackColor` του control της αντλίας. Ο κώδικας που χρησιμοποιείται είναι ο παρακάτω.

```
PumpA.AnalogIntValue1 = Gvariables.Antlia_1_Animation;
```

Αλλαγή τιμής της ιδιότητας `AnalogIntegerColorFill`

```
if (Gvariables.Antlia_1_Animation >= 0 && Gvariables.Antlia_1_Animation < 3)
{
    PumpA.BackColor = System.Drawing.Color.Yellow;
}
else if (Gvariables.Antlia_1_Animation >= 3 && Gvariables.Antlia_1_Animation < 6)
{
    PumpA.BackColor = System.Drawing.Color.Cyan;
}
else
{
    PumpA.BackColor = System.Drawing.Color.SkyBlue;
}
```

Αλλαγή `BackColor` του στοιχείου της αντλίας

### Πλαίσια Διαλόγου

Όπως περιγράφηκε στο Κεφ. 2, πατώντας σε ορισμένα στοιχεία της οθόνης «ΓΕΝΙΚΗΣ ΕΠΟΠΤΕΙΑΣ», εμφανίζονται πλαίσια με τα οποία γίνεται η παραμετροποίηση ορισμένων λειτουργιών του αντλιοστασίου. Τα πλαίσια αυτά είναι ξεχωριστές οθόνες, μικρού μεγέθους, οι οποίες εμφανίζονται πάνω από την οθόνη «ΓΕΝΙΚΗΣ ΕΠΟΠΤΕΙΑΣ». Οι οθόνες αυτές μπορούν να μετακινηθούν οπουδήποτε στην οθόνη του υπολογιστή. Για να εμφανιστεί μια τέτοια οθόνη (π.χ για τα στοιχεία της αντλίας 1) χρησιμοποιείται ο παρακάτω κώδικας.

```
private void PumpA_MouseClick(object sender, MouseEventArgs e)
{
    Antlia_1 FrmAntlia_1 = new Antlia_1();

    FrmAntlia_1.FormBorderStyle = FormBorderStyle.FixedSingle;
    FrmAntlia_1.MaximizeBox = false;
    FrmAntlia_1.MinimizeBox = false;

    FrmAntlia_1.StartPosition = FormStartPosition.Manual;
    FrmAntlia_1.Left = 756;
    FrmAntlia_1.Top = 540;

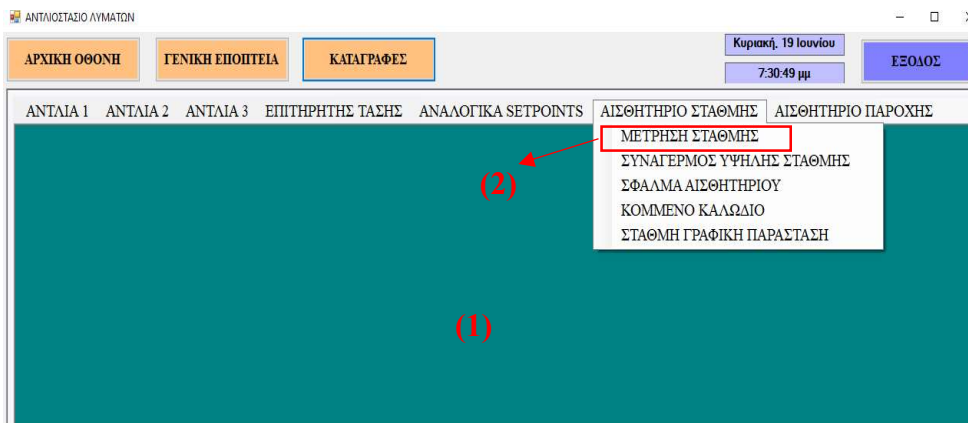
    FrmAntlia_1.Show();
}
```



«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

## Οθόνη Katagrafes

Η οθόνη αυτή έχει την παρακάτω μορφή.



**Εικόνα 4.6**

Έχει στο κέντρο της ένα “Panel” (Εικ.4.6 – 1) στο οποίο ανοίγει η οθόνη καταγραφών που επιθυμούμε, πατώντας πάνω στο αντίστοιχο μενού. Για παράδειγμα για να ανοίξει η οθόνη της καταγραφής της στάθμης της δεξαμενής, κάνουμε κλικ επάνω στην επιλογή «ΜΕΤΡΗΣΗ ΣΤΑΘΜΗΣ» του μενού «ΑΙΣΘΗΤΗΡΙΟ ΣΤΑΘΜΗΣ» (Εικ.4.6 – 2) οπότε και τρέχει ο παρακάτω κώδικας.

```
private void Stathmi_Click(object sender, EventArgs e)
{
    Katagrafes_Main_Panel.Controls.Clear();

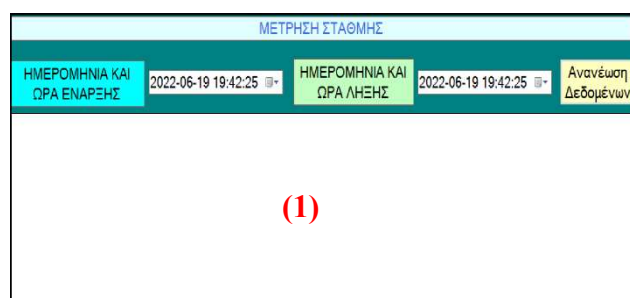
    Katagrafes_Stathmi Frustration = new Katagrafes_Stathmi();

    FrmStathmi.TopLevel = false;

    Katagrafes_Main_Panel.Controls.Add(FrmStathmi);

    FrmStathmi.Show();
}
```

Ο κώδικας αυτός ανοίγει την οθόνη καταγραφών της στάθμης (Εικ.4.7)



**Εικόνα 4.7**

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Αφού επιλέξουμε την ημερομηνία και ώρα έναρξης και λήξης των καταγραφών της στάθμης που θέλουμε να δούμε, πατάμε το μπουτόν «Ανανέωση δεδομένων». Όταν πατηθεί το μπουτόν αυτό τρέχει ο παρακάτω κώδικας.

```
private void btnUpdateData_Click(object sender, EventArgs e)
{
    // ----- ΣΥΝΔΕΣΗ ΜΕ ΤΗΝ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ----- //

    string connectionString = "Data Source=DESKTOP-EPVREL3\\SQLEXPRESS;Initial
Catalog=Antliostasio_Lymaton;Integrated Security=True";
    SqlConnection con = new SqlConnection(connectionString);

    try
    {
        con.Open();
        MessageBox.Show("Πραγματοποιήθηκε σύνδεση με τη βάση δεδομένων");
    }
    catch (Exception ex)
    {
        MessageBox.Show("Δημιουργήθηκε σφάλμα στη σύνδεση με τη βάση δεδομένων" +
ex.Message);
    }

    // ----- ΑΠΕΙΚΟΝΙΣΗ ΔΕΔΟΜΕΝΩΝ ----- //

    DateTime StartDate = Enarxi.Value;
    DateTime EndDate = Lixi.Value;
    int DateResult = DateTime.Compare(StartDate, EndDate);

    //----- Έλεγχος εάν η πρώτη ημερομηνία είναι μικρότερη από την δεύτερη

    if (DateResult < 0)
    {
        SqlDataAdapter sqlDa = new SqlDataAdapter("SELECT Ημερομηνία_Ωρα,Στάθμη
FROM Stathmi WHERE (Ημερομηνία_Ωρα >= '" + Enarxi.Text + ".000')
and(Ημερομηνία_Ωρα < '" + Lixi.Text + ".000')ORDER BY Ημερομηνία_Ωρα ASC",
con);

        DataTable dtbl = new DataTable();

        sqlDa.Fill(dtbl);

        if (dtbl.Rows.Count>0)
        {
            dataGridView1.DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;

            dataGridView1.DefaultCellStyle.Font = new Font("Times New Roman", 12);
            dataGridView1.DataSource = dtbl;

            dataGridView1.Columns["Ημερομηνία_Ωρα"].DefaultCellStyle.Format = "dd-
MM-yyyy HH:mm:ss";
        }
    }
}
```

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

```
dataGridView1.Columns["Στάθμη"].DefaultCellStyle.Format = "#0.###";

dataGridView1.ColumnHeadersDefaultCellStyle.Font = new Font("Times New
Roman", 14);

dataGridView1.ColumnHeadersDefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;

dataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.AllCells;

con.Close();

}
else
{
    MessageBox.Show("Δεν υπάρχουν δεδομένα για την χρονική περίοδο που έχει
επιλεγεί - Δεν θα γίνει ανανέωση δεδομένων");
}

}
else
{
    MessageBox.Show("Δεν έχει επιλέξει σωστή περίοδο δεδομένων - Δεν θα γίνει
ανανέωση δεδομένων");
    con.Close();
}
}
```

Ο κώδικας αυτός κάνει τα παρακάτω:

- 1) Επιχειρεί σύνδεση με τη βάση δεδομένων. Εάν δεν τα καταφέρει εμφανίζει μήνυμα ότι δημιουργήθηκε σφάλμα στη σύνδεση με τη βάση, εμφανίζει την αιτία του σφάλματος και ολοκληρώνει τη λειτουργία του. Εάν τα καταφέρει εμφανίζει μήνυμα ότι πραγματοποιήθηκε σύνδεση με τη βάση και προχωράει παρακάτω.
- 2) Ελέγχει εάν η ημερομηνία και ώρα έναρξης είναι προγενέστερη της ημερομηνίας και ώρας λήξης. Εάν δεν συμβαίνει αυτό εμφανίζει μήνυμα ότι δεν έχει επιλεγεί σωστή περίοδος δεδομένων και ολοκληρώνει τη λειτουργία του. Εάν οι ημερομηνίες και ώρες είναι σωστές προχωράει παρακάτω.
- 3) Ελέγχει εάν για το διάστημα που επιλέχθηκε υπάρχουν δεδομένα. Εάν δεν υπάρχουν εμφανίζει σχετικό μήνυμα και ολοκληρώνει τη λειτουργία του. Εάν υπάρχουν δεδομένα τότε εμφανίζονται στο DataGridView που υπάρχει στη οθόνη (Εικ.4.7 - 1)

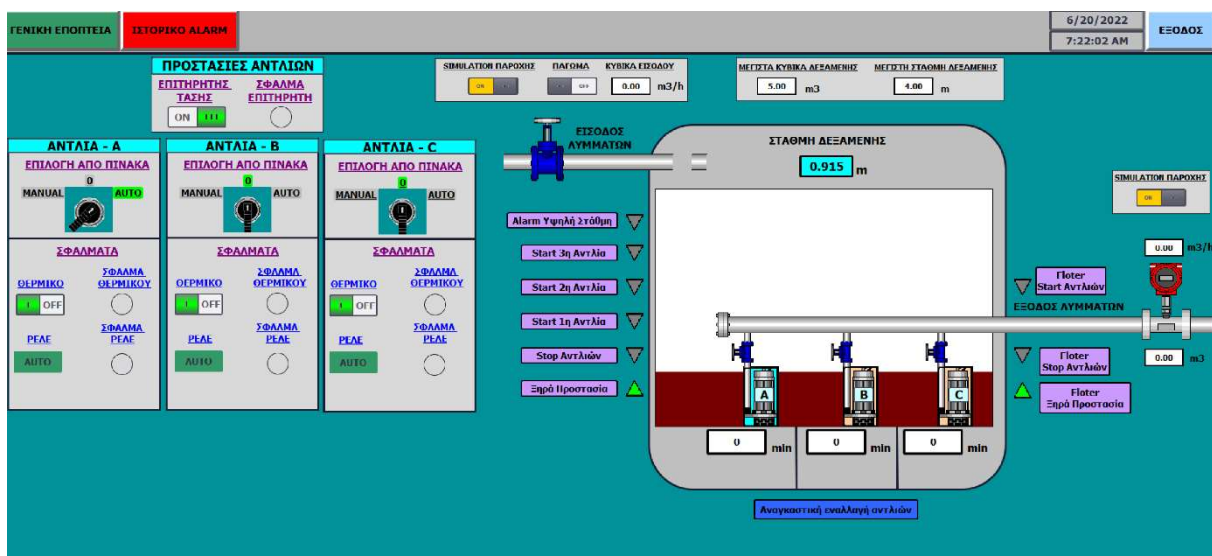
Με τον ίδιο τρόπο λειτουργούν και οι 39 οθόνες καταγραφών που υπάρχουν. Υπάρχει μια διαφορά στις οθόνες γραφικής παράστασης της στάθμης και παροχής εξόδου. Εκεί τα δεδομένα δεν ενημερώνουν DataGridView αλλά ένα Chart (control της C# για γραφικές παραστάσεις).

## ΚΕΦΑΛΑΙΟ 5 – Σύγκριση συμβατικών μεθόδων ανάπτυξης SCADA και της αντικειμενοστραφούς γλώσσας προγραμματισμού Microsoft C#

### Γενικά

Το πρόγραμμα SCADA αρχικά αναπτύχθηκε με το λογισμικό WinCC Advanced της εταιρίας Siemens (Εικ. 5.1). Αυτό έγινε για δύο λόγους:

- 1) Για τη δοκιμή του προγράμματος του PLC κατά τη διάρκεια της δημιουργίας του, επειδή είναι δύσκολο να δοκιμάζεται εξ' ολοκλήρου ένα πρόγραμμα PLC μόνο μέσα από το περιβάλλον ανάπτυξης του.
- 2) Για να υπάρξει ένας οδηγός στο τι ακριβώς θα έπρεπε να φτιαχτεί με τη γλώσσα προγραμματισμού C#. Δηλαδή όποια λειτουργία φτιαχνόταν με τη συμβατική μέθοδο (WinCC Advanced), θα έπρεπε να την υλοποιήσουμε με τη C#.



Εικόνα 5.1

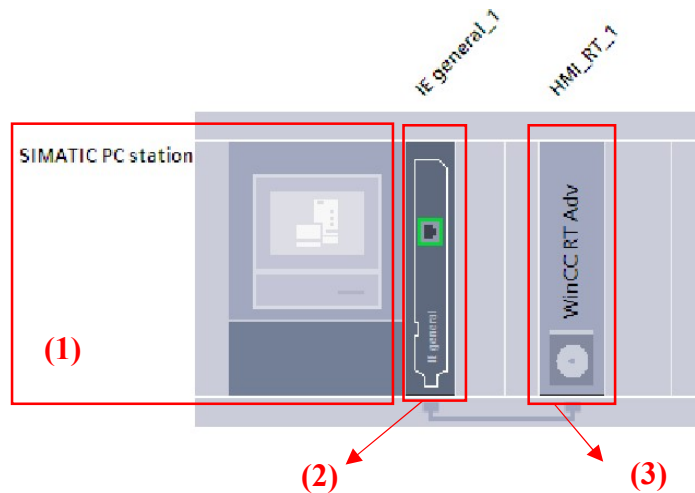
Στο κεφάλαιο αυτό θα γίνει η περιγραφή ορισμένων σημείων του προγράμματος SCADA που αναπτύχθηκε με το WinCC Advanced και θα γίνει σύγκριση τους με τα αντίστοιχα του προγράμματος C#.

### Επικοινωνία με το PLC – Δημιουργία Μεταβλητών

#### WinCC Advanced

Για να οριστεί η επικοινωνία με το PLC ανοίγουμε την καρτέλα Device configuration όπου ορίζουμε αυτά που φαίνονται στην παρακάτω εικόνα.

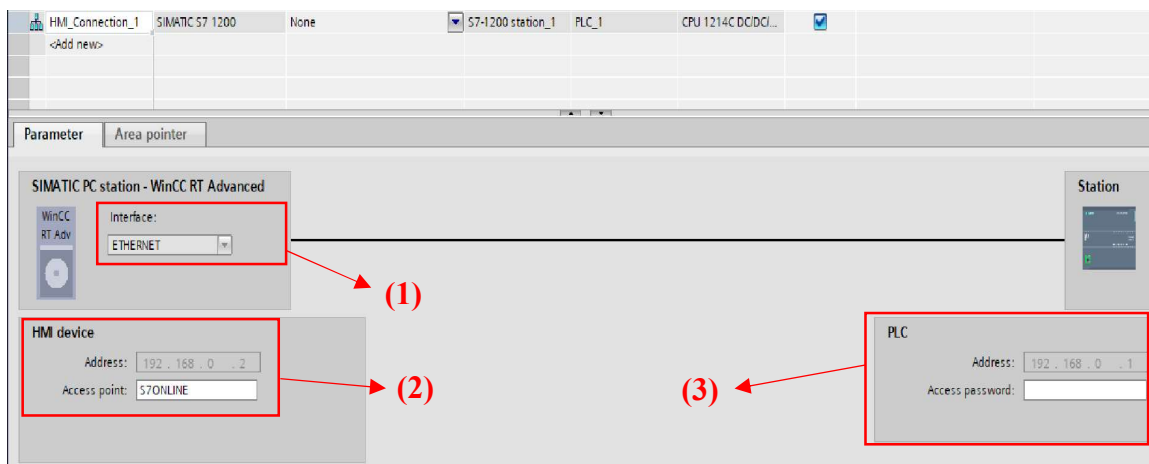
## «Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»



Εικόνα 5.2

Ορίζουμε ότι η συσκευή του SCADA θα είναι υπολογιστής (Εικ. 5.2 - 1), ορίζουμε μια κάρτα δικτύου (Εικ. 5.2 - 2) για τον υπολογιστή και της δίνουμε μια IP διεύθυνση και τέλος ορίζουμε ότι το πρόγραμμα SCADA θα αναπτυχθεί με το λογισμικό WinCC Advanced (Εικ. 5.2 - 3).

Στη συνέχεια ανοίγουμε την καρτέλα Connections και εμφανίζεται το πλαίσιο που φαίνεται στην παρακάτω εικόνα.



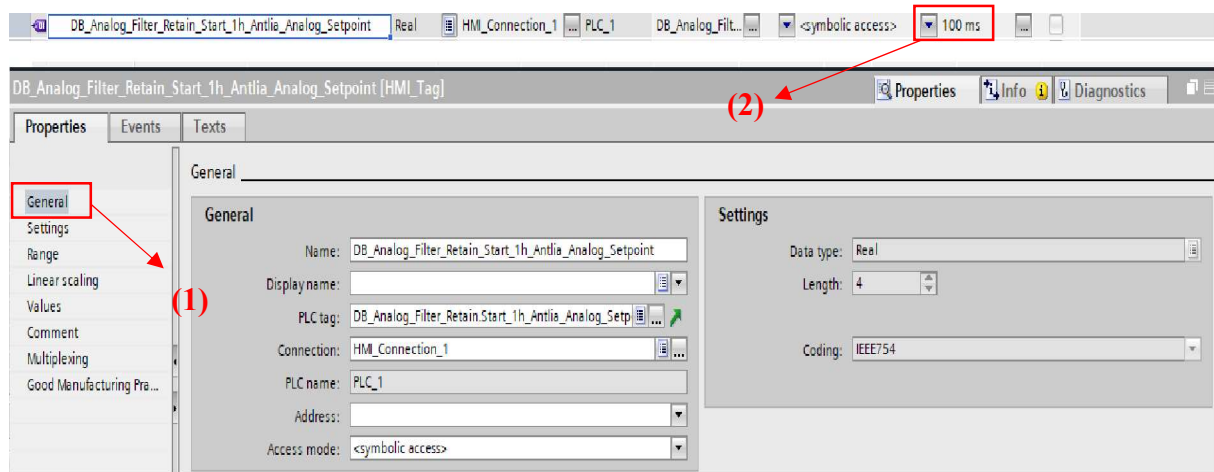
Εικόνα 5.3

Στο πλαίσιο αυτό ορίζουμε πως θα επικοινωνήσει ο υπολογιστής με το PLC (Εικ. 5.3 - 1), ποια είναι η IP διεύθυνση του υπολογιστή (Εικ. 5.3 - 2) και του PLC (Εικ. 5.3 - 3) και η επικοινωνία είναι έτοιμη.

Για να δημιουργήσουμε τις μεταβλητές (tags) αρκεί να κάνουμε drag and drop από τα data block του PLC.

## «Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

Όταν δημιουργηθεί μια μεταβλητή (tag) μπορούμε να δούμε πληροφορίες για αυτή (Εικ. 5.4 - 1) αλλά και να αλλάξουμε κάποια παράμετρο της π.χ κάθε πότε θα διαβάζεται από το PLC (Εικ. 5.4 - 2).



Εικόνα 5.4

### C#

Η επικοινωνία με το PLC και η ενημέρωση των μεταβλητών γίνεται χρησιμοποιώντας τη βιβλιοθήκη Sharp7 όπως έχει περιγραφεί στο Κεφ. 4 στην ενότητα «Σύνδεση και επικοινωνία με το PLC». Η δήλωση των μεταβλητών γίνεται σε ξεχωριστό αρχείο όπου ορίζουμε κάθε μεταβλητή ξεχωριστά.

### Σύγκριση

Οι διαφορές των δύο μεθόδων είναι οι παρακάτω:

- 1) Με τη γλώσσα C# πρέπει να ορίσουμε πάλι τις μεταβλητές τις οποίες έχουμε δημιουργήσει στα datablock του PLC και να τις δώσουμε τα ίδια ονόματα για έχουμε κοινό σημείο αναφοράς με το PLC. Με το λογισμικό WinCC Advanced δημιουργούμε τις μεταβλητές με drag and drop από τα datablock του PLC.
- 2) Με τη γλώσσα C# η ενημέρωση των μεταβλητών γίνεται με κώδικα ο οποίος κάνει τα παρακάτω:
  - Διαβάζει όλα τα datablock από το PLC και δημιουργεί για κάθε datablock έναν πίνακα δεδομένων.
  - Από τον πίνακα αυτό εξάγει το κομμάτι που απαιτείται για την ενημέρωση κάθε μεταβλητής.

## «Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

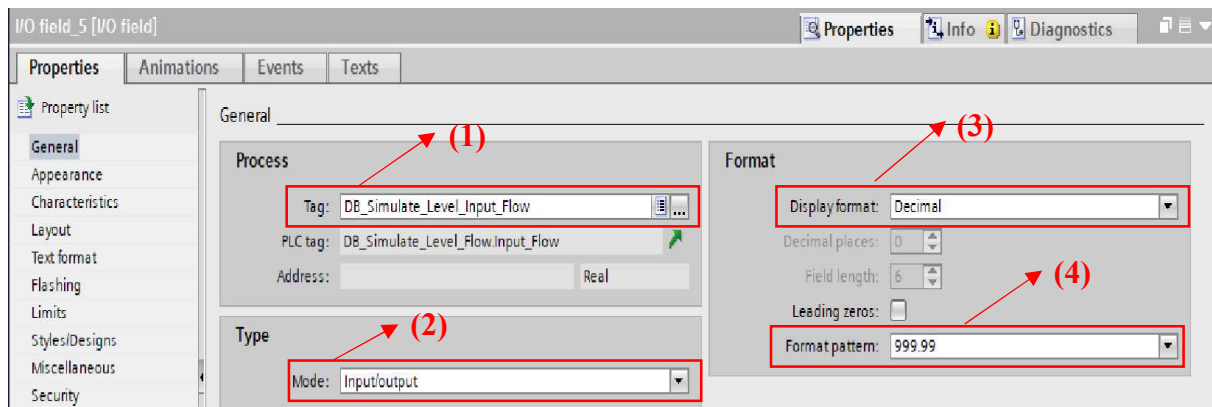
Στο λογισμικό WinCC Advanced από τη στιγμή που δημιουργηθούν οι μεταβλητές και υπάρχει επικοινωνία με το PLC, δεν χρειάζεται να γίνει κάτι άλλο για να ενημερώνονται.

### Πεδία τιμών

Στα πεδία τιμών εμφανίζονται αριθμητικές τιμές που διαβάζονται από το PLC. Με ορισμένα πεδία μπορούμε να γράψουμε και τιμές στο PLC.

### WinCC Advanced

Για να δημιουργήσουμε ένα πεδίο επιλέγουμε το στοιχείο IO field από την κατηγορία elements και το εισάγουμε στην οθόνη. Όταν κάνουμε κλικ στο πεδίο, εμφανίζεται κάτω από την οθόνη το πλαίσιο με τις ιδιότητες του πεδίου (Εικ. 5.5).



Εικόνα 5.5

Στο πλαίσιο αυτό ορίζουμε τα παρακάτω:

- Ποια μεταβλητή θέλουμε το πεδίο να απεικονίζει (Εικ. 5.5 - 1).
- Αν το πεδίο κάνει μόνο ανάγνωση της μεταβλητής ή και εγγραφή σε αυτή (Εικ. 5.5 - 2).
- Πως θα εμφανίζεται η τιμή της μεταβλητής (Εικ. 5.5 - 3) (εξαρτάται από τον τύπο της μεταβλητής) και με ποια ακρίβεια (Εικ. 5.5 - 4).

### C#

Η λειτουργία των πεδίων απαιτεί κώδικα όπως έχει περιγραφεί στο Κεφ. 4 στην ενότητα «Πεδία τιμών (NumericUpDown)».

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

### Σύγκριση

Οι διαφορές των δύο μεθόδων είναι η εξής:

Με το λογισμικό WinCC Advanced ορίζουμε 4 πεδία στο πλαίσιο παραμετροποίησης του πεδίου (Εικ. 5.5) και αυτό είναι έτοιμο να λειτουργήσει.

Με τη γλώσσα C# πρέπει για **κάθε** πεδίο να γραφεί κώδικας ο οποίος κάνει τα παρακάτω:

- Ενημερώνει το πεδίο με την τιμή της μεταβλητής
- Σε περίπτωση εγγραφής από το πεδίο στο PLC, σταματάει τον timer που ενημερώνει το πεδίο και γράφει στη θέση μνήμης που θέλουμε. Στην συνέχεια εκκινεί ξανά τον timer.

### Αποτελέσματα Σύγκρισης

Από τη σύγκριση 2 μόνο στοιχείων του προγράμματος SCADA προκύπτει η εξής σημαντική διαφορά. Για την δημιουργία του προγράμματος με τη γλώσσα προγραμματισμού C# απαιτείται η ανάπτυξη κώδικα ενώ με το λογισμικό WinCC Advanced όχι. Δηλαδή με τη C# απαιτείται μεγαλύτερη προσπάθεια και σίγουρα περισσότερος χρόνος. Επίσης ο εντοπισμός των σφαλμάτων είναι πιο δύσκολος.

Τίθεται επομένως το ερώτημα γιατί να επιλεγεί η C# και όχι το WinCC Advanced. Υπάρχουν δύο λόγοι οι οποίοι αναφέρονται παρακάτω.

### Κόστος

Για να γίνει κατανοητό αυτό ας θεωρήσουμε ότι μια εταιρία πρέπει κατασκευάσει 10 αντλιοστάσια λυμάτων όπως αυτό που εξετάζουμε. Εάν επιλέξει το λογισμικό WinCC Advanced θα πρέπει για το πρόγραμμα SCADA **κάθε** αντλιοστασίου να πληρώσει την άδεια RT (RunTime) του προγράμματος. Η άδεια αυτή, για τον αριθμό των μεταβλητών που χρησιμοποιήθηκαν κοστίζει σίγουρα πάνω από 1000 Ευρώ. Εάν επιλέξει τη γλώσσα προγραμματισμού C# το κόστος θα είναι μηδενικό.

### Ευελιξία

Εάν χρησιμοποιηθεί το λογισμικό WinCC Advanced τότε οι δυνατότητες του προγράμματος SCADA περιορίζονται στα εργαλεία που μας παρέχει αυτό. Εάν θέλουμε περισσότερα εργαλεία τότε πρέπει να επιλεγεί άλλο λογισμικό (π.χ WinCC Professional) το οποίο όμως θα



«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

έχει και πιο ακριβές άδειες RunTime. Με τη γλώσσα C# επειδή κάθε στοιχείο φτιάχνεται με δικό μας κώδικα, δεν υπάρχει περιορισμός στις δυνατότητες οπότε η ευελιξία είναι μεγαλύτερη.

## **ΚΕΦΑΛΑΙΟ 6 – Συμπεράσματα και εξέλιξη συστήματος**

Στην εργασία αυτή ζητήθηκε να γίνει η αυτοματοποίηση ενός αντλιοστασίου λυμάτων. Αρχικά έγινε καταγραφή των αναγκών του αντλιοστασίου και οι ανάγκες αυτές καλύφθηκαν πλήρως με την ανάπτυξη λογισμικού σε 3 επίπεδα. Τα επίπεδα αυτά είναι:

- PLC (Προγραμματιζόμενος Λογικός Ελεγκτής)
- SCADA (Σύστημα εποπτείας και παραμετροποίησης)
- SQL (Βάση Δεδομένων)

Σε κάθε επίπεδο απαιτήθηκε ιδιαίτερη προσπάθεια και χρόνος για την απόκτηση γνώσης σε αυτό. Οι δυσκολίες που εμφανίστηκαν ήταν πολλές, ξεπεράστηκαν όμως όλες με τη συνεχή βελτίωση του κώδικα. Επίσης πραγματοποιήθηκαν και πάρα πολλές δοκιμές.

Μια καινοτομία που παρουσίασε η εργασία αυτή είναι η προσομοίωση του αυτοματισμού μέσα από το πρόγραμμα SCADA, η οποία έδωσε τη δυνατότητα για την αξιόπιστη δοκιμή του.

Η μεγάλη πρόκληση όμως ήταν η δημιουργία του προγράμματος SCADA με μια γλώσσα προγραμματισμού και όχι με τις συμβατικές μεθόδους. Θεωρούμε ότι το πρόγραμμα SCADA που αναπτύχθηκε με τη γλώσσα C# καλύπτει πλήρως τις ανάγκες του αυτοματισμού και δεν υστερεί σε τίποτα του προγράμματος που αναπτύχθηκε με το λογισμικό WinCC Advanced.

Η εμπειρία που αποκτήθηκε από τη χρήση του WinCC Advanced και της C# οδήγησε στο παρακάτω συμπέρασμα:

Για σύνθετες εφαρμογές με μεγάλο αριθμό μεταβλητών είναι καλύτερο να επιλέγονται οι συμβατικές μέθοδοι δημιουργίας SCADA, ενώ σε πιο απλές εφαρμογές που επαναλαμβάνονται, μπορούν άνετα να χρησιμοποιηθούν γλώσσες προγραμματισμού όπως είναι η C#.

Όσο αφορά την εξέλιξη του συνολικού συστήματος (PLC-SCADA-SQL) θα μπορούσαν μελλοντικά να γίνουν τα παρακάτω:

- Αποστολή των συναγερμών του αντλιοστασίου και των αλλαγών στις παραμέτρους λειτουργίας του σε κινητό τηλέφωνο (μέσω SMS) ή σε κάποιο email.

## «Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

- Δημιουργία εφαρμογής για εποπτεία και χειρισμό του αντλιοστασίου από κινητό τηλέφωνο.
- Καταγραφές όχι μόνο στην τοπική βάση δεδομένων, αλλά και στο Cloud.
- Δημιουργία προγράμματος προληπτικής συντήρησης του ηλεκτρολογικού και μηχανολογικού εξοπλισμού του αντλιοστασίου.
- Ενσωμάτωση περισσότερων αντλιοστασίων που βρίσκονται σε διαφορετικές τοποθεσίες.
- Παρακολούθηση του κόστους λειτουργίας του αντλιοστασίου.

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

## **ΒΙΒΛΙΟΓΡΑΦΙΑ – ΠΗΓΕΣ**

[1] – Siemens

S7-1200 Programmable controller System Manual

[https://cache.industry.siemens.com/dl/files/241/109797241/att\\_1066673/v1/s71200\\_system\\_manual\\_en-US\\_en-US.pdf](https://cache.industry.siemens.com/dl/files/241/109797241/att_1066673/v1/s71200_system_manual_en-US_en-US.pdf)

[2] – Siemens

Programming Guideline for S7-1200/1500

[https://cache.industry.siemens.com/dl/files/040/90885040/att\\_970576/v1/81318674\\_Programming\\_guideline\\_DOC\\_v16\\_en.pdf](https://cache.industry.siemens.com/dl/files/040/90885040/att_970576/v1/81318674_Programming_guideline_DOC_v16_en.pdf)

[3] – Siemens

S7-1200 Easy Book

[https://euroec.by/assets/files/siemens/s71200\\_easy\\_book\\_en-US\\_en-US.pdf](https://euroec.by/assets/files/siemens/s71200_easy_book_en-US_en-US.pdf)

[4] – Hans Berger

Automating with SIMATIC S7-1200

[http://www.plc4good.org.ua/files/02\\_materials/book/Berger%20H.%20-%20Automating%20with%20SIMATIC%20S7-1200%20with%20STEP%207%20Basic%202013.pdf](http://www.plc4good.org.ua/files/02_materials/book/Berger%20H.%20-%20Automating%20with%20SIMATIC%20S7-1200%20with%20STEP%207%20Basic%202013.pdf)

[5] – Siemens

STEP 7 and WinCC Engineering V15.1

<https://support.industry.siemens.com/cs/document/109755202/simatic-step-7-basic-professional-v15-1-and-simatic-wincc-v15-1?dti=0&lc=en-US>

[6] – Siemens

WinCC Advanced V14 System Manual

<https://support.industry.siemens.com/cs/document/109742297/simatic-wincc-wincc-advanced-v14?dti=0&lc=en-WW>

[7] – John Sharp

Microsoft Visual C# Step by Step

<https://www.docdroid.net/jhuojDC/1509301046-pdf>

«Αυτοματισμός αντλιοστασίου λυμάτων με PLC και ανάπτυξη εξατομικευμένου λογισμικού SCADA»

[8] – Hans-Petter Halvorsen

Introduction to Visual Studio and C#

<https://www.halvorsen.blog/documents/tutorials/resources/Introduction%20to%20Visual%20Studio%20and%20CSharp.pdf>

[9] – Anders Hejlsberg, Mads Torgersen, Scott Wiltamuth, Peter Golde

The C# Programming Language

<http://www.computo-visual.org/Videojuegos/Libros/ProgramacionC%23.pdf>

[10] – Dusan Petkovic

Microsoft SQL Server 2019 A Beginner's Guide

<https://dl1.newoutlook.it/book/2020/03/Microsoft-SQL-Server-2019-A-Beginners-Guide.pdf>

[11] – Dejan Sarka, Milos Radivojevic, William Durkin

SQL Server 2016 Developer's Guide

<http://projanco.com/Library/SQL%20server%202016%20developer%20guide.pdf>

[12] – Φελώνης Γιώργος

ΑΝΤΛΙΑ Θεωρία - Εφαρμογές

[13] – GRUNDFOS

Αντλίες αποστράγγισης και λυμάτων – τεχνικό εγχειρίδιο

[14] – Sharp7 – Reference Manual

<http://snap7.sourceforge.net/sharp7.html>

[15] – Mesta Automation

HMI with C# from Zero to Hero: a guide for desktop apps from the beginning to advanced level

<https://www.mesta-automation.com/hmi-with-c-from-zero-to-hero-a-guide-for-desktop-apps-from-the-beginning-to-advanced-level/>