



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**

**Π.Μ.Σ. “ΕΦΑΡΜΟΣΜΕΝΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ”**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και  
HMI (C#)**

**Σωτήριος Γ. Μπαλωμένος**

**Εισηγητής: Δρ Νικόλαος Ζ. Ζάχαρης, Καθηγητής**

**ΑΘΗΝΑ**  
**ΙΟΥΝΙΟΣ 2018**



Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)

## **ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)**

**Σωτήριος Γ. Μπαλωμένος  
Α.Μ. AIS0059**

**Εισηγητής:**

**Δρ Νικόλαος Ζ. Ζάχαρης, Καθηγητής**

**Εξεταστική Επιτροπή:**

**Ημερομηνία εξέτασης**



## **ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

Ο κάτωθι υπογεγραμμένος Σωτήριος Μπαλωμένος, του Γεωργίου, με αριθμό μητρώου ais0059 φοιτητής του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασης της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»



## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα διπλωματική εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό των συστημάτων βιομηχανικού αυτοματισμού.

Θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου, κο Δρ Νικόλαο Ζ. Ζάχαρη για τις γνώσεις που μου πρόσφερε κατά την διάρκεια του μεταπτυχιακού αλλά και την υποστήριξη του για την περάτωση της διπλωματικής εργασίας.

Επίσης θα ήθελα να ευχαριστήσω, όλους του καθηγητές του μεταπτυχιακού για τις πολύτιμες γνώσεις που μου προσέφεραν κατά την διάρκεια των σπουδών μου.

Τέλος θα ήθελα να ευχαριστήσω ιδιαίτερω την σύζυγό μου Μαρία, τις κόρες μου, Αναστασία και Κωνσταντίνα αλλά και όλη μου την οικογένεια, για την υποστήριξη τους και την βοήθεια που μου παρείχαν ώστε να ολοκληρώσω επιτυχώς το μεταπτυχιακό πρόγραμμα. Για το λόγο αυτό θα ήθελα να τους αφιερώσω την παρακάτω διπλωματική εργασία.





## ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία ασχολείται με την ανάπτυξη συστήματος βιομηχανικού αυτοματισμού με την χρήση PLC(programmable logic controller) και λογισμικού HMI(human machine interface) του οποίου η ανάπτυξη έχει γίνει με την χρήση της γλώσσας προγραμματισμού #C.

Η συγκεκριμένη εφαρμογή HMI θα έχει σαν στόχο τον απομακρυσμένο έλεγχο κρίσιμων παραμέτρων για την εκτέλεση μίας διεργασίας. Ο χρήστης θα μπορεί να ελέγχει απομακρυσμένα τις περιβαλλοντολογικές συνθήκες που επικρατούν σε χώρο που εκτελείται μία σημαντική διεργασία παραγωγής. Συγκεκριμένα ο χρήστης μέσω της εφαρμογής HMI θα έχει τις παρακάτω δυνατότητες:

1. Συνεχής παρακολούθηση των τιμών θερμοκρασίας και υγρασίας.
2. Συνεχής παρακολούθηση για την ύπαρξη συμβάντος-συναγερμού (alarm).
3. Ενημέρωση για το ιστορικό τιμών θερμοκρασίας και υγρασίας του αισθητήρα.
4. Ενημέρωση για το ιστορικό συμβάντων-συναγερμών.
5. Ρύθμιση ορίων συναγερμού για θερμοκρασία και υγρασία.
6. Εξαγωγή ιστορικού τιμών και συμβάντων σε μορφή κειμένου (txt).

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Συστήματα Βιομηχανικού Αυτοματισμού  
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: PLC, SCADA, HMI, #C

## ABSTRACT

The present thesis concerns the development of thesis for industrial automation system with the use of PLC (programmable logic controller) and HMI (human machine interface) software for which the programming language #C was used for the development.

The HMI application will have as basic scope the remote control of critical parameters for doing the process procedure. The user has the capability to control remotely the environmental conditions in an area where an important process in taking place. Specifically the user within the HMI application has the following possibilities:

1. Continuous monitoring of temperature and humidity current values.
2. Continuous monitoring for the existence of alarms.
3. Information for the historical values of temperature and humidity.
4. Information for the historical alarms
5. Limit alarm setting for temperature and humidity
6. Export historical values and alarms in text format (txt).

SCIENTIFIC AREA: Industrial Automation Systems

KEYWORDS: PLC, SCADA, HMI, #C

## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ .....	10
ΚΕΦΑΛΑΙΟ 1 .....	19
ΕΙΣΑΓΩΓΗ .....	19
1.1 Περιγραφή του αντικείμενου της διπλωματικής εργασίας .....	19
1.2 Δομή της διπλωματικής εργασίας .....	20
ΚΕΦΑΛΑΙΟ 2 .....	21
ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ .....	21
2.1 Προγραμματιζόμενοι λογικοί ελεγκτές PLC .....	21
2.1.1 Ιστορική Αναδρομή .....	21
2.1.2 Η χρήση του PLC .....	23
2.1.3 Η Αρχή λειτουργίας του προγραμματιζόμενου λογικού ελεγκτή .....	24
2.1.4 Η Δομή ενός προγραμματιζόμενου λογικού ελεγκτή .....	26
2.1.4.1 Το πλαίσιο τοποθέτησης μονάδων .....	27
2.1.4.2 Η Μονάδα τροφοδοσίας .....	27
2.1.4.3 Η Κεντρική μονάδα επεξεργασίας .....	27
2.1.4.4 Οι Μονάδες εισόδων – εξόδων .....	27
2.1.4.5 Γλώσσες προγραμματισμού PLC .....	29
2.1.4.6 Μνήμη της Κεντρικής μονάδας .....	30
2.1.5 Προγραμματιζόμενος Βιομηχανικός Αυτοματισμός και Στάδια εργασίας .....	31
2.1.6 Κύριες λειτουργίες προγραμματιζόμενων λογικών ελεγκτών .....	32
2.1.7 Συστήματα PLC – HMI .....	33
2.1.8 Συνδεσμολογίες PLC με HMI .....	34
2.2 Συστήματα SCADA .....	35
2.2.1 Οι βασικοί στόχοι του SCADA .....	38
2.2.2 Τα βασικά μέρη ενός συστήματος SCADA .....	39
2.2.3 SCADA και Αυτόματος έλεγχος – Σύγκριση DCS και PLC .....	41
2.2.4 SCADA και HMI .....	43
2.2.5 Συστήματα λογισμικού .....	45
2.2.6 Χειρισμός των δεδομένων SCADA σε περίπτωση προβλημάτων .....	47
2.2.7 Πλεονεκτήματα και εξέλιξη των συστημάτων SCADA .....	48

2.2.8 Λογισμικά για την ανάπτυξη συστημάτων SCADA.....	50
ΚΕΦΑΛΑΙΟ 3.....	53
ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΑΥΤΟΜΑΤΙΣΜΟΥ ΜΕ PLC ΚΑΙ HMI #C.....	53
3.1 Το σύστημα αυτοματισμού .....	53
3.1.1 Γενικά .....	53
3.1.2 Περιγραφή συστήματος.....	53
3.2 Η αρχιτεκτονική του συστήματος αυτοματισμού .....	54
3.2.1 Γενικά .....	54
3.2.2 Τοπολογία συστήματος.....	54
3.2.2 Το βιομηχανικό ψηφιακό θερμοϋγρόμετρο.....	55
3.2.3 Ο προγραμματιζόμενος λογικός ελεγκτής (PLC) .....	57
3.2.4 Η ηχητική ειδοποίηση.....	58
3.2.5 Η εφαρμογή HMI.....	59
3.3 Η εποπτεία ελέγχου και η συλλογή δεδομένων με την Εφαρμογή HMI .....	60
3.3.1 Γενικά .....	60
3.3.2 Είσοδος στην Εφαρμογή HMI .....	60
3.3.3 Σύνδεση με PLC .....	62
3.3.4 Χώρος παραγωγής .....	64
3.3.4.1 Γενικά .....	64
3.3.4.2 Συνεχής παρακολούθηση των τιμών θερμοκρασίας και υγρασίας ....	64
3.3.4.3 Συνεχής παρακολούθηση για την κατάσταση συναγερμών .....	65
3.3.5 Ρυθμίσεις οργάνου Θερμοκρασίας-Υγρασίας.....	67
3.3.5.1 Γενικά .....	67
3.3.5.2 Επεξεργασία ρυθμίσεων οργάνου Θερμοκρασίας- Υγρασίας.....	68
3.3.6 Ιστορικό τιμών οργάνου Θερμοκρασίας και Υγρασίας .....	70
3.3.6.1 Γενικά .....	70
3.3.6.2 Φόρτωση Ιστορικού τιμών.....	71
3.3.6.3 Χειροκίνητη εισαγωγή τιμών.....	74
3.3.6.4 Εξαγωγή τιμών σε αρχείο κειμένου (txt) .....	76
3.3.7 Ιστορικό συμβάντων οργάνου Θερμοκρασίας και Υγρασίας.....	78
3.3.7.1 Γενικά .....	78
3.3.7.2 Φόρτωση ιστορικού συμβάντων.....	79
3.3.7.3 Εξαγωγή συμβάντων σε αρχείο κειμένου (txt).....	82
3.3.8 Έξοδος από την εφαρμογή HMI.....	84

ΚΕΦΑΛΑΙΟ 4 .....	87
ΜΕΘΟΔΟΛΟΓΙΑ ΚΑΙ ΑΝΑΠΤΥΞΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΑΥΤΟΜΑΤΙΣΜΟΥ .....	87
4.1 Γενικά .....	87
4.2 Ανάπτυξη στην πλευρά του PLC .....	87
4.2.1 Γενικά .....	87
4.2.2 Περιβάλλον ανάπτυξης για τον προγραμματισμό του PLC .....	87
4.2.2.1 Το λογισμικό SIMATIC STEP 7 στο TIA Portal .....	88
4.2.3 Ανάπτυξη προγράμματος εφαρμογής στο PLC .....	90
4.2.3.1 Γενικά .....	90
4.2.3.2 Παραμετροποίηση του ελεγκτή στο STEP 7 TIA Portal .....	91
4.2.3.3 Σύνδεση του οργάνου μέτρησης στο PLC .....	92
4.2.3.4 Τρόπος επικοινωνίας μεταξύ PLC και Εφαρμογής HMI .....	93
4.2.3.5 Οι έλεγχοι συναγεργμών στο πρόγραμμα του PLC .....	95
4.3 Ανάπτυξη εφαρμογής στον Η/Υ .....	96
4.3.1 Γενικά .....	96
4.3.2 Περιβάλλον ανάπτυξης για την Εφαρμογή στον Η/Υ .....	97
4.3.3 Ανάπτυξη παραθυρικής εφαρμογής HMI στον Η/Υ .....	98
4.3.3.1 Γενικά .....	98
4.3.3.2 Τα εργαλεία ανάπτυξης της εφαρμογής HMI .....	98
4.3.3.3 Η δημιουργία παραθυρικής εφαρμογής HMI .....	99
4.3.3.3 Η επικοινωνία παραθυρικής εφαρμογής HMI με το PLC .....	100
4.3.3.3 Η ανάγνωση τιμών PLC μέσω του Data Block 1 .....	103
4.3.3.5 Η εγγραφή τιμών στο Data Block 1 του PLC .....	105
4.3.3.6 Η Δημιουργία Ιστορικού τιμών και συναγεργμών .....	106
ΚΕΦΑΛΑΙΟ 5 .....	111
ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΡΓΑΣΙΕΣ ΓΙΑ ΤΗΝ ΕΦΑΡΜΟΓΗ .....	111
5.1 Γενικά .....	111
5.2 Συμπεράσματα .....	111
5.3 Μελλοντικές εργασίες .....	111
ΠΑΡΑΡΤΗΜΑ Α' .....	113
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	135

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Το PLC της εταιρείας Allen Brandley .....	23
Εικόνα 2: Το PLC S7 της εταιρείας Siemens .....	23
Εικόνα 3: Θερμοϋγρόμετρο HT206 της εταιρείας ΞΕΝΩΝ.....	56
Εικόνα 4: Το PLC S7-1200 με CPU 1214C της εταιρείας Siemens .....	57
Εικόνα 5: Η ηχητική ειδοποίηση της εφαρμογής .....	58
Εικόνα 6: Η εφαρμογή HMI .....	60
Εικόνα 7: Η Συντόμευση για την εφαρμογή HMI .....	61
Εικόνα 8: Η Αρχική οθόνη της εφαρμογής HMI .....	61
Εικόνα 9: Η οθόνη σύνδεσης με το PLC .....	62
Εικόνα 10: Μήνυμα 1 σύνδεσης από το διαγνωστικό παράθυρο.....	63
Εικόνα 11: Μήνυμα 2 από το διαγνωστικό παράθυρο .....	63
Εικόνα 12: Το Μήνυμα αποσύνδεσης της εφαρμογής από το PLC .....	63
Εικόνα 13: Μήνυμα σφάλματος στην επικοινωνία .....	64
Εικόνα 14: Η οθόνη Παραγωγής.....	65
Εικόνα 15: Οι τρέχουσες τιμές οργάνου .....	65
Εικόνα 16: Το σύστημα σε κατάσταση συναγερμού .....	66
Εικόνα 17: Το μήνυμα συναγερμού και το μπουτον αναγνώρισης.....	66
Εικόνα 18: Το σύστημα σε κατάσταση χωρίς κάποιο συμβάν .....	67
Εικόνα 19: Οθόνη ρυθμίσεων συναγερμών .....	68
Εικόνα 20: Μπουτον αλλαγής Ρυθμίσεων .....	68
Εικόνα 21: Η Οθόνη αλλαγής Ρυθμίσεων .....	69
Εικόνα 22: Πάνω όριο θερμοκρασίας με τιμή 27 .....	69
Εικόνα 23: Δίνοντας την τιμή 11 στο πεδίο, πάνω όριο θερμοκρασίας .....	70
Εικόνα 24: Μήνυμα Error in Values.....	70
Εικόνα 25: Η Οθόνη ιστορικού τιμών .....	71
Εικόνα 26: Επιλογή της αρχικής ημερομηνίας για την εμφάνιση ιστορικού τιμών	72
Εικόνα 27: Επιλογή της τελικής ημερομηνίας για την εμφάνιση ιστορικού τιμών .	72
Εικόνα 28: Εμφάνιση πίνακα ιστορικού τιμών .....	73
Εικόνα 29: Χρησιμοποιώντας το Scroll bar στον πίνακα ιστορικού .....	73
Εικόνα 30: Τελευταία καταχωρημένη τιμή στη βάση δεδομένων .....	74
Εικόνα 31: Επιτυχημένη καταχώρηση στη βάση δεδομένων.....	75
Εικόνα 32: Απεικόνιση ιστορικού μετά την χειροκίνητη καταχώρηση .....	75
Εικόνα 33: Φόρτωση ιστορικού τιμών .....	76
Εικόνα 34: Εξαγωγή τιμών σε txt .....	77
Εικόνα 35: Το αρχείο hmiValues .....	77
Εικόνα 36: Το ιστορικό σε αρχείο κειμένου .....	78
Εικόνα 37: Το ιστορικό σε αρχείο κειμένου .....	78
Εικόνα 38: Οθόνη ιστορικού συμβάντων.....	79
Εικόνα 39: Επιλογή αρχικής ημερομηνίας.....	80
Εικόνα 40: Επιλογής τελικής ημερομηνίας .....	80
Εικόνα 41: Πίνακας ιστορικού συμβάντων από 1/03/2018 έως 15/05/2018.....	81
Εικόνα 42: Απεικόνιση των τελευταίων συμβάντων στον πίνακα .....	81
Εικόνα 43: Φόρτωση ιστορικού συμβάντων.....	82

Εικόνα 44: Εξαγωγή συμβάντων σε αρχείο κειμένου txt.....	83
Εικόνα 45: Το αρχείο hmiAlarms.....	83
Εικόνα 46: Το ιστορικό συμβάντων σε αρχείο κειμένου.....	84
Εικόνα 47: Το ιστορικό συμβάντων σε αρχείο κειμένου.....	84
Εικόνα 48: Το μπουτον για την έξοδο από την εφαρμογή .....	85
Εικόνα 49: Στο κάτω δεξί μέρος της οθόνης βρίσκεται το μπουτον για την έξοδο	85
Εικόνα 50: Πλήρως αυτοματοποιημένη αυτοματοποίηση (Totally Integrated Automation) .....	88
Εικόνα 51: Το λογισμικό Simatic Step 7 στο TIA Portal.....	88
Εικόνα 52: Απεικόνιση TIA Portal μέσα από Project .....	89
Εικόνα 53: Απεικόνιση και παραμετροποίηση συσκευής.....	90
Εικόνα 54: Tia Portal editor, Ladder Logic (LAD) .....	90
Εικόνα 55: Ρύθμιση της διεύθυνσης IP στο PLC .....	91
Εικόνα 56: Η CPU 1214C/DC/DC/DC στο Step 7 Tia Portal .....	92
Εικόνα 57: Ο κώδικας για την ανάγνωση τιμών από το όργανο HT206.....	93
Εικόνα 58: Ρύθμιση No Protection στη CPU .....	94
Εικόνα 59: Ρύθμιση Optimized block access στη CPU .....	94
Εικόνα 60: Το Data Block 1 του προγράμματος PLC .....	95
Εικόνα 61: Το μπλοκ προγράμματος για τους συναγερμούς σε Ladder Logic(LAD) .....	96
Εικόνα 62: Το μπλοκ προγράμματος για τους συναγερμούς σε Ladder Logic(LAD) .....	96
Εικόνα 63: Το περιβάλλον ανάπτυξης Visual Studio .....	98
Εικόνα 64: Η παραθυρική εφαρμογή HMI .....	99
Εικόνα 65: Η φόρμα 1(Form1) της εφαρμογής HMI .....	100
Εικόνα 66: Πρόσθεση κλάσης στο project.....	101
Εικόνα 67: Δημιουργία βιβλιοθήκη Κλάσης .....	101
Εικόνα 68: Η βιβλιοθήκη Sharp7 .....	102
Εικόνα 69: Η δημιουργία αναφοράς στο project .....	102
Εικόνα 70: Σύνδεση της αναφοράς με τον οδηγό που έχουμε εγκατεστημένο στον Η/Υ .....	103
Εικόνα 71: Ο κώδικας της σύνδεσης-αποσύνδεσης με το PLC .....	103
Εικόνα 72: Κώδικας ανάγνωσης του Data Block 1 από το PLC .....	104
Εικόνα 73: Το Data Block 1 στο PLC .....	105
Εικόνα 74: Ο κώδικας εγγραφής στο Data Block 1 του PLC .....	106
Εικόνα 75: Η βάση δεδομένων myhmi .....	107
Εικόνα 76: Ο πίνακας xenon για την καταγραφή τιμών .....	107
Εικόνα 77: Ο πίνακας xenonalarm για την καταγραφή των συμβάντων .....	107
Εικόνα 78: Πρόσθεση αναφοράς mysql.data.dll .....	108
Εικόνα 79: Ο κώδικας για την σύνδεση στη βάση myHMI .....	109

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Έλεγχος συσκευών και μηχανών με PLC.....	24
Σχήμα 2: Το διάγραμμα λειτουργίας του PLC.....	25
Σχήμα 3: Ο Κύκλος λειτουργίας PLC .....	26
Σχήμα 4: Η Υλοποίηση της λογικής πράξης AND .....	30
Σχήμα 5: Εκδόσεις του συστήματος WinCC flexible.....	33
Σχήμα 6: Έλεγχος PLC με HMI .....	34
Σχήμα 7: Έλεγχος με αρκετές οθόνες HMI.....	34
Σχήμα 8: Κεντρικό σύστημα HMI .....	35
Σχήμα 9: Σύστημα ελέγχου με PLC και SCADA.....	37
Σχήμα 10: Τοπολογία συστήματος SCADA με RTU.....	39
Σχήμα 11: Μια εφαρμογή HMI της Ellipse Software για την εποπτεία υδραγωγείου .....	44
Σχήμα 12: HMI οθόνη με Simatic WinCC .....	51
Σχήμα 13: Η τοπολογία του συστήματος.....	55



## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Συγκριτικός πίνακας αντίστοιχων CPU της ίδιας σειράς SIMATIC S7-1200 .....	58
---	----

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

**PLC** Programmable logic Controller

**SCADA** Supervisory Control And Data Acquisition

**OPC** Open Platform Communications

**CPU** Central Processing Unit

**MC7** Machine Code 70

**LAD** Ladder Logic

**STL** Statement List

**FBD** Function Block Diagram

**RAM** Random Access Memory

**ROM** Read Only Memory

**EEPROM** Electrically Erasable Programmable Read Only Memory

**ADC** Analog-To-Digital Converter

**CAN** Computer Automatic Network

**HMI** Human Machine Interface

**RTU** Remote Telemetry Unit

**MTU** Master Station Computer

**DCS** Distributed Control System

**LAN** Local Area Network

**SMS** Short Message Service

**MMI** Man Machine Interface

**COTS** Commercial Off The Shelf



## ΚΕΦΑΛΑΙΟ 1

### ΕΙΣΑΓΩΓΗ

Στη σημερινή εποχή τα συστήματα βιομηχανικού αυτοματισμού έχουν καθιερωθεί στην πλειοψηφία των βιομηχανιών συμβάλλοντας στις παραγωγικές διαδικασίες. Προγραμματιζόμενοι λογικοί ελεγκτές γνωστά σαν PLC(Programmable Logic Controller) ή συστήματα εποπτικού ελέγχου και συλλογής δεδομένων γνωστά ως SCADA(Supervisory Control And Data Acquisition) θεωρούνται απαραίτητα εργαλεία σε σύγχρονα συστήματα αυτοματισμού.

Η εξέλιξη της τεχνολογίας και της πληροφορικής συμβάλει σημαντικά και για την εξέλιξη των συστημάτων αυτοματισμού. Στις μέρες μας τα συστήματα βιομηχανικού αυτοματισμού όπως PLC ή SCADA πρέπει να ακολουθήσουν τις νέες αυτές τεχνολογίες που στοχεύουν στις επικοινωνίες δεδομένων μεταξύ διαφορετικών εξοπλισμών μέσω τρίτων λογισμικών(OPC,Open Platform Communications) αλλά και διαφορετικών συστημάτων που έχουν σαν αποτέλεσμα την επιτάχυνση παραγωγικών διαδικασιών και την συλλογή πληροφορίας προς όφελος των διεργασιών που εκτελούνται.

Σε αυτό το κεφάλαιο αναλύεται το αντικείμενο της διπλωματικής εργασίας καθώς επίσης και η δομή της.

#### **1.1 Περιγραφή του αντικείμενου της διπλωματικής εργασίας**

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η σχεδίαση, ανάπτυξη συστήματος αυτοματισμού με την χρήση PLC(programmable logic controller) και λογισμικού HMI(human machine interface) του οποίου η ανάπτυξη έχει γίνει με την χρήση της γλώσσας προγραμματισμού #C. Ιδιαίτερη έμφαση θα πρέπει να δοθεί στην ανάπτυξη λογισμικού στην πλευρά του ηλεκτρονικού υπολογιστή, ώστε αυτό να αποτελέσει το σύστημα εποπτικού ελέγχου και συλλογής δεδομένων (SCADA) της εφαρμογής, αναπτύσσοντας το εν λόγο λογισμικό σε ένα περιβάλλον λογισμικού πληροφορικής και όχι σε μία έτοιμη πλατφόρμα SCADA γνωστής εταιρείας όπως συνηθίζεται σε συστήματα αυτοματισμού.

Η συγκεκριμένη εφαρμογή θα έχει σαν στόχο τον απομακρυσμένο έλεγχο κρίσιμων παραμέτρων για την εκτέλεση μίας διεργασίας. Πιο συγκεκριμένα ο χρήστης θα μπορεί να ελέγχει απομακρυσμένα τις περιβαλλοντολογικές συνθήκες που επικρατούν σε χώρο που εκτελείται μία σημαντική διεργασία παραγωγής. Σε αρκετές βιομηχανίες (τροφίμων, φαρμάκου, κα) οι συνθήκες αυτές λαμβάνονται σοβαρά υπόψη για την εκτέλεση παραγωγικών διεργασιών ώστε να εξασφαλίζουν την ποιότητα των προϊόντων.

Επιπλέον μέσω της εφαρμογής θα μπορούν να καταγράφονται οι τρέχουσες τιμές του οργάνου αλλά και να υπάρχει άμεση ειδοποίηση όταν οι συνθήκες αυτές δεν είναι οι ιδανικές σύμφωνα με τα όρια συνθηκών που έχει ορίσει ο χρήστης.

## **1.2 Δομή της διπλωματικής εργασίας**

Στα επόμενα κεφάλαια γίνεται μία αναλυτική παρουσίαση της διπλωματικής εργασίας και των αντίστοιχων μερών που την απαρτίζουν τόσο για υλικό(hardware) όσο για λογισμικό(software).

Η εργασία απαρτίζεται από 5 κεφάλαια και καθένα από αυτά παρουσιάζει ένα από τα μέρη της.

Στο παρών κεφάλαιο γίνεται μία εισαγωγή της εργασίας με την περιγραφή του αντικειμένου της αλλά και της δομής της. Στη συνέχεια παρουσιάζεται το δεύτερο κεφάλαιο με ένα θεωρητικό υπόβαθρο αλλά και μία ιστορική αναδρομή στα συστήματα αυτοματισμού.

Το σύστημα αυτοματισμού που εξετάζουμε στην συγκεκριμένη εργασία παρουσιάζεται στο τρίτο κεφάλαιο μαζί με τα μέρη που το απαρτίζουν.

Στο τέταρτο κεφάλαιο παρουσιάζεται η μεθοδολογία πάνω στην οποία δουλέψαμε για την ανάπτυξη του συστήματος αυτοματισμού.

Τα συμπεράσματα της μελέτης όπως και οι προτεινόμενες μελλοντικές εργασίες θα αναλυθούν στο πέμπτο κεφάλαιο αυτής της εργασίας.

Τέλος στο παράρτημα Α' παρουσιάζεται ο κώδικας του συστήματος αυτο

## ΚΕΦΑΛΑΙΟ 2

### ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Στο κεφάλαιο γίνεται μία αναφορά στο θεωρητικό υπόβαθρο που χρειάζεται να έχει κάποιος ώστε να γίνουν κατανοητά τα επόμενα κεφάλαια.

#### 2.1 Προγραμματιζόμενοι λογικοί ελεγκτές PLC

##### 2.1.1 Ιστορική Αναδρομή

Η εξέλιξη των βιομηχανικών αυτοματισμών ακολούθησε όπως ήταν λογικό την εξέλιξη της τεχνολογίας. Οι πρώτοι ήταν αποκλειστικά μηχανικοί, όλοι δηλαδή οι έλεγχοι καθορίζονταν από κίνηση γραναζιών και μοχλών. Με την χρήση του ηλεκτρισμού έγινε το μεγάλο άλμα στους αυτοματισμούς.

Με το τέλος του δεύτερου παγκοσμίου πολέμου αρχίζει η ηλεκτρονική εποχή. Οι πρώτες ηλεκτρονικές συσκευές εμφανίζονται από τις αρχές του 20<sup>ου</sup> αιώνα όπως το ραδιόφωνο και αργότερα η τηλεόραση, τα ραντάρ και οι ασύρματοι. Το βασικό εξάρτημα αυτών των συσκευών ήταν η ηλεκτρονική λυχνία. Με την ανακάλυψη του τρανζίστορ το 1950 δημιουργείται η αρχή της ηλεκτρονικής επανάστασης των ημιαγωγών. Το στοιχείο αυτό αντικατέστησε την ακριβή, ογκώδη και ενεργειοβόρα ηλεκτρονική λυχνία και έκανε τις ηλεκτρονικές συσκευές αρκετά μικρότερες, πιο εύκολες στην κατασκευή και πιο φθηνές. Ο πρώτος ηλεκτρονικός υπολογιστής, ο ENIAC κατασκευάστηκε το 1945 και χρησιμοποιούσε λυχνίες. Ο ENIAC δεν θύμιζε τίποτα με τους σημερινούς ηλεκτρονικούς υπολογιστές καθώς ήταν ένα ολόκληρο εργοστάσιο το οποίο έλυνε μαθηματικές εξισώσεις. Μετά από το 1950 και με την χρήση των τρανζίστορ έχουμε τους πρώτους πραγματικούς ηλεκτρονικούς υπολογιστές, τους οποίους χρησιμοποιούσαν κυρίως στη μηχανογράφηση, όπως στην διαχείριση και αποθήκευση μεγάλων αρχείων δεδομένων.

Από τη δεκαετία του '60 οι μηχανικοί έχουν αρχίσει να σκέφτονται τρόπους για να αξιοποιήσουν τις καταπληκτικές δυνατότητες των υπολογιστών στη βιομηχανία. Από τις πρώτες εφαρμογές των υπολογιστών στον χώρο της βιομηχανίας ήταν οι αυτόματες εργαλειομηχανές (τόρνοι, φρέζες), οι οποίες μέχρι εκείνο το διάστημα χρησιμοποιούσαν κυρίως μηχανολογικούς και λιγότερο

ηλεκτρολογικούς αυτοματισμούς. Αυτή η επιτυχημένη δοκιμή οδήγησε τους μηχανικούς να σκέφτονται την αντικατάσταση των μηχανικών αυτοματισμών από έναν υπολογιστή. Μέχρι τη δεκαετία του '80 ήταν όμως αδύνατο λόγω του υπερβολικού κόστους του υπολογιστή αλλά και της δυσκολίας που υπήρχε στη χρήση του.

Το 1975 με την κατασκευή του πρώτου μικροϋπολογιστή ξεκινάει η επανάσταση της πληροφορικής. Η τεχνολογία άλλαξε πορεία σε όλους τους τομείς στην καθημερινότητα του ανθρώπου. Ο μικροϋπολογιστής εισέρχεται παντού, σε οποιοδήποτε τομέα ή εφαρμογή.

Μέχρι τη δεκαετία του '80 η βιομηχανία χρησιμοποιούσε ελάχιστα τα ηλεκτρονικά. Το 90% και πλέον των βιομηχανικών αυτοματισμών ήταν με ρελέ (ηλεκτρονόμοι). Με την αρχή της δεκαετίας του '80 οι εταιρείες παραγωγής ηλεκτρολογικού υλικού παρουσιάζουν στους τεχνικούς και μηχανικούς της βιομηχανίας ένα νέο προϊόν αυτοματισμού, το οποίο ονομαζόταν **PLC**. Το πλήρες όνομα αυτής της συσκευής είναι Programmable Logic Controller (Προγραμματιζόμενος Λογικός Ελεγκτής).

Το PLC είναι ένας μικροϋπολογιστής προσαρτημένος κατάλληλα ώστε να χρησιμοποιείται για την λειτουργία των αυτοματισμών. Τα PLC προορίζονταν να αντικαταστήσουν τον κλασσικό αυτοματισμό με το ρελέ. Όπως είναι κατανοητό υπήρξε μία τεράστια αλλαγή στον τρόπο που μέχρι τότε δούλευε η βιομηχανία, καθώς έπρεπε να περάσει από τα ρελέ στους υπολογιστές.

Την σημερινή εποχή ο κλασσικός αυτοματισμός με ρελέ τείνει να εκλείψει. Όλες οι νέες εγκαταστάσεις χρησιμοποιούν PLC. Τα PLC έχουν εξελιχθεί πάρα πολύ σε σχέση με τα αρχικά μοντέλα της δεκαετίας του '80. Το τεχνικό προσωπικό της βιομηχανίας έχει εκπαιδευτεί κατάλληλα στον χειρισμό αλλά και στον προγραμματισμό τους. Η χρήση των PLC παρέχει πολλά πλεονεκτήματα σε σχέση με τον κλασσικό αυτοματισμό. Η χρήση των PLC σε σχέση με τον κλασσικό αυτοματισμό συμφέρει πρωτίστως τις εταιρείες που παράγουν είδη αυτοματισμού. Σε αυτές κοστίζει υπερβολικά η παραγωγή ενός τεράστιου αριθμού βοηθητικών ρελέ και ενός μεγάλου αριθμού χρονικών και απαριθμητών. Σε αντίθεση με αυτά τα υλικά αυτοματισμού, παράγει μία και μοναδική συσκευή, το PLC.

Οι κυριότεροι κατασκευαστές PLC σε παγκόσμιο επίπεδο είναι:

- ❖ Siemens, Γερμανία
- ❖ Allen Brandley (AB), ΗΠΑ

- ❖ Mitsubishi, Ιαπωνία
- ❖ Schneider Electric, Γαλλία
- ❖ ABB, Γερμανία
- ❖ B&R (Bernecker + Reiner, ABB), Αυστρία
- ❖ GE Fanuc, ΗΠΑ



Εικόνα 1: Το PLC της εταιρείας Allen Brandley



Εικόνα 2: Το PLC S7 της εταιρείας Siemens

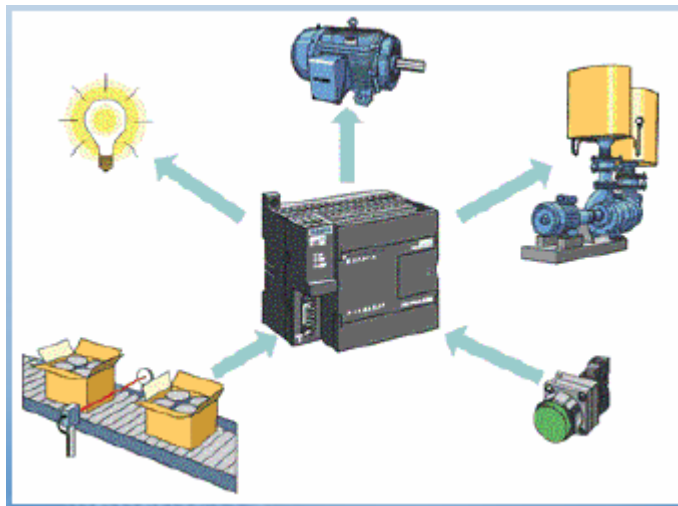
### 2.1.2 Η χρήση του PLC

Το PLC είναι ένα ψηφιακό ηλεκτρονικό σύστημα, σχεδιασμένο για βιομηχανική χρήση, το οποίο χρησιμοποιεί προγραμματιζόμενη μνήμη για αποθήκευση των εντολών, ώστε να εκτελούνται διάφορες λειτουργίες όπως λογικές, χρονικές, αριθμητικές και μετρητικές πράξεις και να ελέγχονται μέσω ψηφιακών/αναλογικών μονάδων, διαφόρων μηχανών ή διαδικασιών. Το PLC είναι μία ειδική συσκευή, η οποία έρχεται να αντικαταστήσει τον κλασσικό αυτοματισμό με όλους τους



βοηθητικούς ηλεκτρονόμους, τα χρονικά και τους απαριθμητές. Αντί να έχουμε ένα ηλεκτρολογικό πίνακα με πολύπλοκες συνδεσμολογίες μεταξύ των ηλεκτρολογικών υλικών που αναφέραμε παραπάνω, με την χρήση του PLC η λειτουργία του οποίου προγραμματίζεται μέσω ενός ηλεκτρονικού υπολογιστή με τη βοήθεια ενός ειδικού λογισμικού.

Τα PLC χρησιμοποιούνται για να ελέγχουν μηχανές, διεργασίες και όπου απαιτείται να γίνονται αυτόματες λειτουργίες. Στην πλειοψηφία τους χρησιμοποιούνται στη βιομηχανία αλλά και σε κτιριακές εγκαταστάσεις, στη ναυτιλία, σε μεγάλα έργα όπως σήραγγες, σταθμούς παραγωγής ενέργειας, ύδρευση, βιολογικούς καθαρισμούς, αεροδρόμιο και άλλες δεκάδες εφαρμογές.



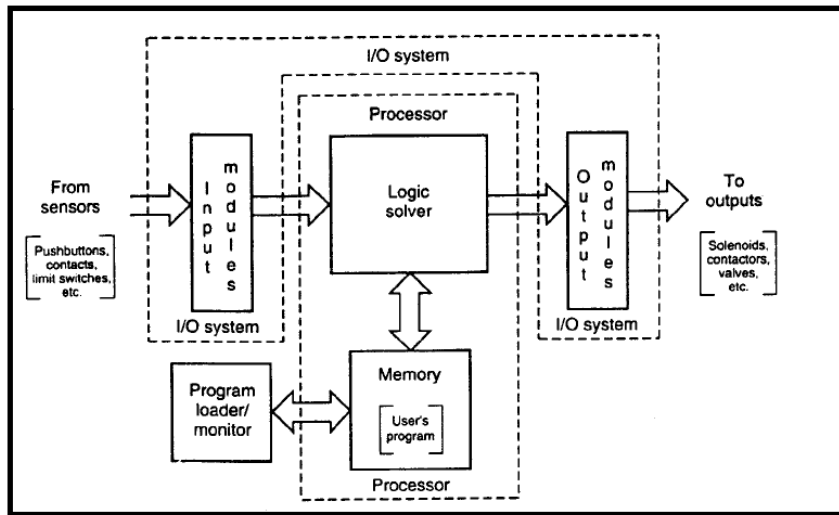
**Σχήμα 1:** Έλεγχος συσκευών και μηχανών με PLC

Σαν αρχή λειτουργίας θα μπορούσαμε να πούμε ότι τα PLC συλλέγουν πληροφορίες μέσω των εισόδων του, “αποφασίζουν” με βάση το σενάριο του προγράμματος και ελέγχουν τις εξόδους τους μέσω των οποίων τελικά ελέγχουν μηχανές ή διεργασίες.

### **2.1.3 Η Αρχή λειτουργίας του προγραμματιζόμενου λογικού ελεγκτή**

Αναφορικά με το υλικό, όλα τα PLC αποτελούνται από την CPU (Central Processing Unit) η οποία αφού διαβάσει την κατάσταση των εισόδων(input modules) ενεργοποιούν τις εξόδους(output modules) σύμφωνα με το σενάριο που υπάρχει στο πρόγραμμα που έχουμε αποθηκεύσει στην μνήμη του. Το σύστημα του PLC συμπληρώνεται από μία τροφοδοτική μονάδα και πιθανόν από διατάξεις

ενδείξεων, χειρισμών(operator panel, display) καθώς και πιθανόν από επιπλέον κάρτες εισόδων ή εξόδων.



Σχήμα 2: Το διάγραμμα λειτουργίας του PLC

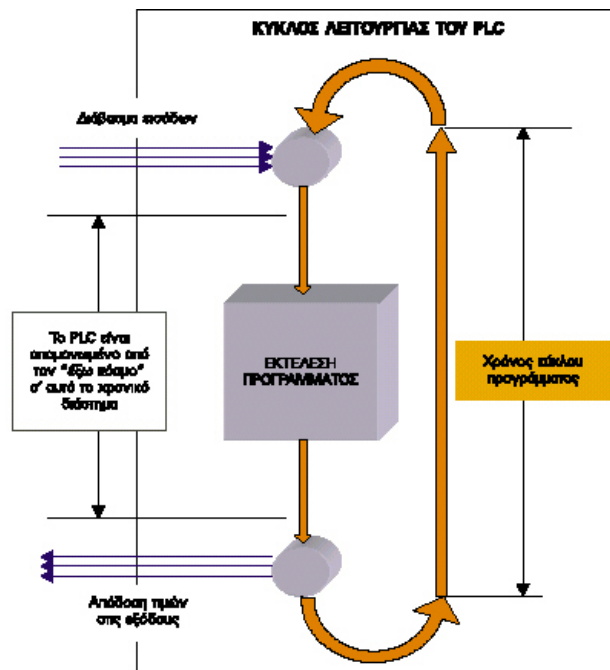
Έστω ένα PLC βρίσκεται σε κατάσταση λειτουργίας του αυτοματισμού (κατάσταση RUN). Τα βήματα που ακολουθεί κατά την λειτουργία είναι:

**Βήμα 1<sup>ο</sup>** : Αρχικά ο μικροεπεξεργαστής (CPU) διαβάζει τις εισόδους. Για κάθε είσοδο ελέγχει αν έχει υψηλή τάση (λογικό 1) ή χαμηλή τάση (λογικό 0). Η τιμή 0 ή 1 για κάθε είσοδο αποθηκεύεται σε μία ειδική περιοχή μνήμης η οποία ονομάζεται εικόνα εισόδων (input image).

**Βήμα 2<sup>ο</sup>** : Ο μικροεπεξεργαστής στη συνέχεια χρησιμοποιώντας σαν δεδομένα τις τιμές των εισόδων, που διάβασε, εκτελεί τις εντολές του προγράμματος, που λειτουργεί τον αυτοματισμό. Το πρόγραμμα περιέχει μία σειρά από λογικές πράξεις. Η εκτέλεση του προγράμματος θα δώσει τα αντίστοιχα αποτελέσματα για τις εξόδους. Τα αποτελέσματα αυτά αποθηκεύονται στην ειδική περιοχή μνήμης η οποία ονομάζεται εικόνα εξόδων (output image). Όπως η εικόνα εισόδων, έτσι αντίστοιχα και η εικόνα εξόδων περιέχει την τιμή (0 ή 1) για κάθε έξοδο. Οι τιμές αυτές προκύπτουν μετά από την εκτέλεση των λογικών πράξεων του προγράμματος.

**Βήμα 3<sup>ο</sup>** : Ο μικροεπεξεργαστής στη συνέχεια αποδίδει τις τιμές της εικόνας εξόδων στις εξόδους. Αυτό σημαίνει ότι θα δοθεί υψηλή τάση σε όποια έξοδο έχει 1 και αντίστοιχα χαμηλή τάση σε όποια έξοδο έχει 0.

Μετά το τέλος του 3<sup>ου</sup> βήματος συμπληρώνεται ένας πλήρης κύκλος λειτουργίας και η διαδικασία αρχίζει από την αρχή. Ο κύκλος λειτουργίας θα εκτελείται συνεχώς όσο το PLC βρίσκεται σε κατάσταση λειτουργίας RUN. Δηλαδή ένα PLC επαναλαμβάνει συνεχώς τα βήματα του κύκλου λειτουργίας. Στο σχήμα 3 παρουσιάζεται ένας κύκλος λειτουργίας PLC.



Σχήμα 3: Ο Κύκλος λειτουργίας PLC

Ο χρόνος που χρειάζεται για να εκτελέσει έναν πλήρη κύκλο λειτουργίας το PLC, ονομάζεται χρόνος κύκλου και εξαρτάται από την ταχύτητα του επεξεργαστή αλλά και από τον αριθμό και το είδος των εντολών.

#### 2.1.4 Η Δομή ενός προγραμματιζόμενου λογικού ελεγκτή

Ένας προγραμματιζόμενος λογικός ελεγκτής αποτελείται από τα παρακάτω στοιχεία:

- Πλαίσιο τοποθέτησης των μονάδων(RACK)
- Μονάδα τροφοδοσίας (PS)
- Κεντρική μονάδα επεξεργασίας(CPU)
- Μονάδες εισόδων / εξόδων(DI/DO)
- Μονάδα προγραμματισμού(software)

#### **2.1.4.1 Το πλαίσιο τοποθέτησης μονάδων**

Οι διάφορες μονάδες που θα συνθέσουν το σύστημα αυτοματισμού τοποθετούνται στο πλαίσιο rack. Αυτές οι μονάδες είναι οι μονάδες τροφοδοσίας, η μονάδα επεξεργασίας(CPU) και οι κάρτες εισόδων-εξόδων. Στην περίπτωση που οι θέσεις το πλαίσιο δεν αρκούν χρησιμοποιείται πλαίσιο επέκτασης ώστε να τοποθετηθούν οι νέες μονάδες εισόδων-εξόδων.

#### **2.1.4.2 Η Μονάδα τροφοδοσίας**

Η μονάδα τροφοδοσίας χρησιμοποιείται για να παρέχει την απαραίτητη τάση στη CPU και στις κάρτες εισόδων-εξόδων. Οι τυπικές εσωτερικές τάσεις μονάδων είναι 24 V DC/AC, 220 V AC.

#### **2.1.4.3 Η Κεντρική μονάδα επεξεργασίας**

Η κεντρική μονάδα επεξεργασίας ουσιαστικά αποτελεί τον εγκέφαλο του PLC, ελέγχει, εκτελεί όλες τις λειτουργίες του PLC. Είναι δηλαδή η μονάδα αυτή που είναι υπεύθυνη για την λειτουργία του αυτοματισμού. Η CPU είναι ένας μικροϋπολογιστής που περιλαμβάνει το μικροεπεξεργαστή και τη μνήμη. Ο μικροεπεξεργαστής εκτελεί όλες τις λειτουργίες και τα δεδομένα που επεξεργάζεται είναι δυαδικής μορφής. Η ταχύτητα ενός επεξεργαστή εκφράζεται μέσα από το ρυθμό σάρωσης δεδομένων της μνήμης. Το κόστος του είναι ανάλογο με την ταχύτητα του. Για την επιλογή της CPU λαμβάνετε υπόψη ο αριθμός των εισόδων-εξόδων που θα έχει το σύστημα, τον όγκο του προγράμματος που απαιτείται, την επιθυμητή ταχύτητα στη λήψη αποφάσεων και τις ανάγκες δικτύωσης και επικοινωνίας που απαιτεί η εφαρμογή.

#### **2.1.4.4 Οι Μονάδες εισόδων – εξόδων**

Οι μονάδες εισόδων – εξόδων αποτελούν τις μονάδες επικοινωνίας της κεντρικής μονάδας με τον έξω κόσμο, όπως είναι οι αισθητήρες, διακόπτες χειρισμού, που δίνουν τις πληροφορίες (εντολές), καθώς και με τα ρελέ ισχύος κινητήρων, ηλεκτρομαγνητικές βαλβίδες, ενδεικτικές λυχνίες και άλλους αποδέκτες που εκτελούν τις εντολές της κεντρικής μονάδας.

Η κεντρική μονάδα δέχεται ψηφιακά σήματα εισόδου και εξόδου χαμηλής τάσης και πολύ μικρού ρεύματος. Η τάση που δέχεται είναι συνήθως 0 volt για το λογικό “0” και 5 volt για το λογικό “1”. Το ρεύμα εισόδου και το ρεύμα εξόδου δεν μπορεί

να ξεπεράσει τα μερικά mA. Οι μονάδες εισόδων-εξόδων αναλαμβάνουν να προσαρμόσουν τα σήματα που έχουμε στον αυτοματισμό, στην είσοδο και έξοδο, με τα σήματα που μπορεί να δεχτεί η κεντρική μονάδα, τόσο από άποψη τάσης όσο και από ρεύματος. Αυτή η προσαρμογή γίνεται με χρήση ηλεκτρονικών στοιχείων ισχύος, είτε με τη χρήση των κατάλληλων μικρό ρελέ.

Το σύστημα PLC καταλήγει πάντα σε ακροδέκτες (κλέμες). Αυτοί οι ακροδέκτες ανήκουν στις μονάδες εισόδων – εξόδων του PLC. Στους ακροδέκτες εισόδων καταλήγουν οι αγωγοί από τους αισθητήρες (τερματικούς διακόπτες, πιεζοστάτες, κτλ), διακόπτες, μπουτόνς, κτλ. Στους ακροδέκτες εξόδων συνδέονται οι αγωγοί που τροφοδοτούν πηνία, ρελέ ισχύος, ηλεκτρομαγνητικές βαλβίδες, λυχνίες ένδειξης, κτλ.

Οι μονάδες εισόδων και εξόδων στους διάφορους τύπους των PLC αντιμετωπίζονται με διαφορετικό τρόπο αλλά γενικότερα ισχύουν τα παρακάτω:

- Οι μονάδες εισόδων ή εξόδων μπορεί να λειτουργήσουν με συνεχή τάση ή με εναλλασσόμενη τάση. Οι τυπικές τάσεις που συναντάμε στα PLC είναι: 24 V DC/AC, 48 V DC/AC, 230 V AC.
- Τα κυκλώματα και οι τάσεις των εισόδων είναι ανεξάρτητα από τα αντίστοιχα κυκλώματα των εξόδων.
- Η τάση εισόδων (είναι η τάση που θα φτάσει σε μία είσοδο, όταν ενεργοποιηθεί ο αντίστοιχος αισθητήρας) διαχωρίζεται γαλβανικά από το υπόλοιπο εσωτερικό κύκλωμα του PLC. Τα αντίστοιχα ισχύουν και για τις εξόδους.

**Αισθητήρια – στοιχεία εισόδου:** Αισθητήρια ή τα στοιχεία εισόδου είναι συσκευές που μετατρέπουν κάποια φυσική κατάσταση σε ηλεκτρικό σήμα που αυτό μεταφέρεται στην είσοδο του PLC. Ένα παράδειγμα είναι το μπουτόν. Όταν το πιέζουμε η “φυσική” του κατάσταση αλλάζει και η πληροφορία αυτή μεταφέρεται σαν ηλεκτρικό σήμα (ρεύμα) στην είσοδο του PLC.

**Διατάξεις ενεργοποίησης (Actuators):** Τα PLC ελέγχουν μία μηχανή ή μία διεργασία ώστε να ελέγξουν κάποιες συσκευές που χαρακτηρίζονται ως φορτία. Αυτός ο έλεγχος (ενεργοποίηση ή απενεργοποίηση) συνήθως δεν γίνεται απ’ ευθείας καθώς το ασθενές ηλεκτρονικό σήμα που παρέχει το PLC στις εξόδους του (μέσω των οποίων όπως ήδη αναφέραμε μπορεί να δίνει “εντολές”) είναι σε θέση να ενεργοποιήσει π.χ ένα κινητήρα ή μία αντλία ή ένα κύκλωμα φωτισμού. Αυτό συμβαίνει κατά κανόνα μέσω μιας “διάταξης ενεργοποίησης”, όπως είναι το

ρελέ του οποίου το πηνίο διεγείρεται και ενεργοποιείται από ηλεκτρικό σήμα που τροφοδοτείται από την έξοδο του PLC. Οι επαφές του ρελέ ανοίγουν ή κλείνουν και μέσω εκείνων, τροφοδοτείται το τελικό φορτίο, όπως ο κινητήρας.

#### **2.1.4.5 Γλώσσες προγραμματισμού PLC**

Τα PLC είναι ουδέτερες συσκευές αφού δεν είναι από πριν κατασκευασμένες για μία συγκεκριμένη εφαρμογή. Ανάλογα με τις απαιτήσεις της εκάστοτε εγκατάστασης προγραμματίζονται. Υπάρχουν διάφοροι τρόποι για να προγραμματίσεις ένα PLC γιατί ποικίλουν και τα επίπεδα γνώσης και εμπειριών του κάθε προγραμματιστή. Στις διαφορές βρίσκουμε, στο τι απεικονίζεται στην οθόνη του υπολογιστή, αφού το τελικό αποτέλεσμα είναι πάντα ένα και το αυτό- η γλώσσα μηχανής MC7(machine code 70) που καταλαβαίνει το PLC. Οι διάφορες γλώσσες προγραμματισμού μετατρέπονται σε γλώσσα μηχανής κατά την μεταφορά του προγράμματος από τη συσκευή προγραμματισμού στο PLC.

Υπάρχουν διεθνώς τρεις τυποποιημένες μορφές προγραμματισμού που έχουν επικρατήσει:

- Γλώσσα προγραμματισμού LADDER Logic ή γλώσσα ηλεκτρολογικών γραφικών

Η LADDER Logic(LAD) είναι η πρώτη γλώσσα προγραμματισμού ενός PLC, ή οποία είναι γλώσσα γραφικών και χρησιμοποιεί ηλεκτρομηχανικά σύμβολα και επιτρέπει ουσιαστικά τη μεταφορά του ηλεκτρολογικού σχεδίου στο PLC. Με την γλώσσα αυτή η εργασία σχεδιασμού του αυτοματισμού γίνεται εύκολα και γρήγορα.

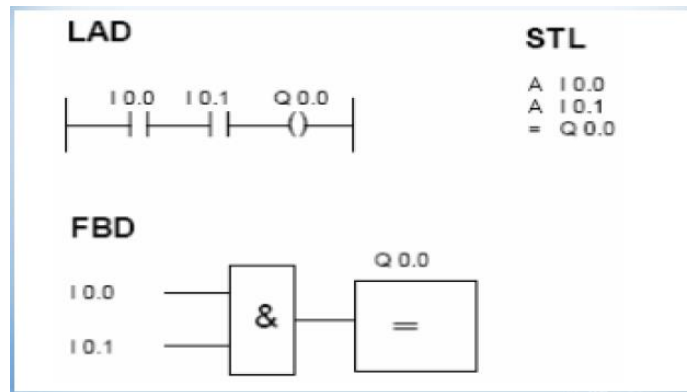
- Γλώσσα προγραμματισμού λίστας εντολών STL(Statement List) ή γλώσσα λογικών εντολών

Η δεύτερη γλώσσα είναι η STL που αναπτύχθηκε σχεδόν ταυτόχρονα με την LADDER. Η σύνταξη των εντολών μοιάζει αρκετά με αυτή του κώδικα μηχανής(machine code), όπου οι εντολές και οι λειτουργίες χαρακτηρίζονται από διευθύνσεις.

- Γλώσσα προγραμματισμού λογικών γραφικών FBD( Function Block Diagram)

Η Τρίτη γλώσσα προγραμματισμού είναι η Function Block Diagram η οποία είναι και αυτή γλώσσα με γραφικά. Εδώ οι εντολές αναπαριστώνται με λογικά blocks, παρόμοια με αυτά που συναντώνται στην άλγεβρα Boole. Εδώ όπως και στα

διαγράμματα της ψηφιακής τεχνολογίας μπορούμε να παρακολουθήσουμε τη ροή του σήματος ανάμεσα στα “κουτιά”.



Σχήμα 4: Η Υλοποίηση της λογικής πράξης AND

#### 2.1.4.6 Μνήμη της Κεντρικής μονάδας

Η μνήμη που βρίσκεται στην κεντρική μονάδα επεξεργασίας (CPU) διακρίνεται σε μνήμη RAM, ROM και EEPROM.

**Μνήμη RAM:** Η RAM (Random Access Memory, μνήμη τυχαίας προσπέλασης) χρησιμοποιείται για να μπορούμε να γράφουμε και να σβήνουμε, και η οποία όταν πέσει η τροφοδοσία σβήνει. Σε αυτή την μνήμη η κεντρική μονάδα αποθηκεύει πληροφορίες σε ξεχωριστές περιοχές εργασίας.

Διακρίνονται στις παρακάτω περιοχές:

- Περιοχή μνήμης στην οποία αποθηκεύονται οι καταστάσεις των εισόδων και των εξόδων. Αυτή η περιοχή ονομάζεται για τις εισόδους εικόνα εισόδου και για τις εξόδους εικόνα εξόδου.
- Περιοχή μνήμης στην οποία αποθηκεύονται οι ενδιάμεσες πληροφορίες που αφορούν τη λειτουργία του αυτοματισμού.
- Περιοχή μνήμης στην οποία αποθηκεύονται τα χρονικά.
- Περιοχή μνήμης στην οποία αποθηκεύονται οι απαριθμητές.
- Περιοχή μνήμης στην οποία αποθηκεύονται τα προγράμματα του χρήστη, τα προγράμματα που λειτουργούν ένα συγκεκριμένο αυτοματισμό.

**Μνήμη ROM:** Ο κατασκευαστής του PLC αποθηκεύει στη ROM (Read only Memory) το λειτουργικό σύστημα, δηλαδή της οδηγίες (πρόγραμμα) για όλες τις βασικές λειτουργίες που απαιτούνται για να δουλέψει το PLC.

**Μνήμη EEPROM:** Λόγω ότι η μνήμη RAM με την απώλεια της τροφοδοσίας δεν συγκρατεί τα δεδομένα της (εκτός εάν το PLC έχει μπαταρία), τα PLC χρησιμοποιούν έναν άλλο τύπο μνήμης, που ονομάζεται EEPROM (Electrically Erasable Programmable Read Only Memory), η οποία προγραμματίζεται και σβήνει ηλεκτρικά. Η μνήμη αυτή με την πτώση της τροφοδοσίας, διατηρεί τα δεδομένα της, και η οποία γράφεται και σβήνεται μέσω ειδικής συσκευής.

### 2.1.5 Προγραμματιζόμενος Βιομηχανικός Αυτοματισμός και Στάδια εργασίας

Για να υλοποιηθεί ένας προγραμματιζόμενος βιομηχανικός αυτοματισμός πρέπει να υλοποιηθούν τα έξι παρακάτω στάδια:

1. **Τεχνική περιγραφή:** Καταγραφή των απαιτήσεων του πελάτη όσο αφορά τη σημερινή κατάσταση της εγκατάστασης, τις νέες απαιτήσεις από τον αυτοματισμό αλλά και πιθανές μελλοντικές επεκτάσεις.
2. **Επιλογή τύπου και μονάδων PLC:** Η επιλογή εξοπλισμού γίνεται πάντα με βάση τεchnοοικονομικά κριτήρια, την καλύτερη τεχνική λύση με το χαμηλότερο δυνατό κόστος, μέσα από μία πληθώρα συστημάτων και των συνιστωσών τους.
3. **Εκπόνηση σχεδίων:** Κατασκευή πίνακα όπου θα εγκατασταθεί το PLC.
4. **Προγραμματισμός:** Με βάση τις προδιαγραφές που έθεσε ο πελάτης υλοποιείται ο προγραμματισμός του PLC. Το πρόγραμμα δοκιμάζεται εν μέρει για τη σωστή του λειτουργία σε συνθήκες γραφείου, αφού μια ολοκληρωμένη δοκιμή θα πρέπει να γίνει στην εγκατάσταση.
5. **Θέση σε λειτουργία:** Το PLC εγκατεστημένο στον πίνακα μεταφέρεται και τοποθετείται στην εγκατάσταση όπου γίνονται και οι τελικές καλωδιώσεις με τα περιφερειακά (κινητήρες, βάνες, τερματικούς, αισθητήρια). Εδώ γίνεται ο οριστικός έλεγχος του αυτοματισμού σύμφωνα με την τεχνική περιγραφή.
6. **Φάκελος έργου:** Τέλος, δημιουργείται ο φάκελος έργου με τελικά διορθωμένα σχέδια και το πρόγραμμα εκτυπωμένο με επεξηγηματικά σχόλια.



### 2.1.6 Κύριες λειτουργίες προγραμματιζόμενων λογικών ελεγκτών

Σήμερα τα PLC έχουν διάφορες λειτουργίες που βοηθούν στην δημιουργία αυτοματισμού. Αυτές οι λειτουργίες αυξάνουν συνεχώς καθώς τα PLC εξελίσσονται με ταχύτερους ρυθμούς.

Ενδεικτικά αναφέρονται οι σημαντικότερες από αυτές.

- **Λειτουργία απαριθμητών:** Ένα πολύ σημαντικό στοιχείο των PLC αποτελούν σήμερα οι απαριθμητές. Αυτοί μπορούν να απαριθμήσουν εξωτερικούς ή εσωτερικούς παλμούς. Η απαρίθμηση μπορεί να είναι είτε προς τα πάνω(count up) είτε προς τα κάτω(count down). Η λειτουργία τους δεν είναι ίδια σε όλα τα PLC.
- **Δυνατότητες πραγματικών ωρολογίων:** Μέσα από ωρολόγια μπορούν να προγραμματιστούν κάποιες έξοδοι σε πραγματικό χρόνο, ημερομηνία και ώρα.
- **Αριθμητικές επεξεργασίες:** Σήμερα, τα σύγχρονα PLC έχουν προσεγγίσει πολύ τις δυνατότητες που βρίσκουμε στους ηλεκτρονικούς υπολογιστές. Όλα τα PLC έχουν την δυνατότητα να επεξεργάζονται αριθμητικές πράξεις.
- **Αναλογικές είσοδοι-έξοδοι:** Ενώ τα PLC αρχικά δημιουργήθηκαν για να αντικαταστήσουν τους αυτοματισμούς καλωδιωμένης λογικής (αυτοματισμούς με ρελέ), οι δυνατότητες τους πλέον έχουν εξαπλωθεί με προοπτική να καλύψουν πλήρως και τα συστήματα αυτομάτου ελέγχου, όπως είναι αναλογικοί έλεγχοι για θερμοκρασία, πίεση, στάθμη, στροφές κινητήρων κλπ. Αυτό υλοποιείται με την δυνατότητα των PLC να δέχονται και να επεξεργάζονται τις αναλογικές εξόδους όπως αντίστοιχα να παρέχουν αναλογικές εξόδους. Το PLC με την βοήθεια ενός μετατροπέα, μετατρέπει τις αναλογικές τιμές εισόδων σε ψηφιακές τιμές (analog-to-digital converter, **ADC**) και στη συνέχεια επεξεργάζεται τις τιμές αυτές αξιοποιώντας τις δυνατότητες που αναφέρθηκαν παραπάνω. Αυτή η δυνατότητα επεξεργασίας αναλογικών σημάτων έχει δώσει άλλη προοπτική εξέλιξης στα PLC.
- **Δικτύωση PLC-PLC, PLC-H/Y:** Τα PLC σήμερα μπορούν να συνδέονται μεταξύ τους ανταλλάσσοντας πληροφορίες, όπως και να συνεργάζονται απευθείας με ηλεκτρονικούς υπολογιστές, οι οποίοι μπορούν να ελέγχουν όλη την παραγωγή όπως επίσης την αποθήκη και άλλα τμήματα του εργοστασίου. Όλα τα παραπάνω αποτελούν ένα βασικό Βιομηχανικό

δίκτυο αυτοματισμού (computer automatic network, CAN). Στη βάση κάθε βιομηχανικού δικτύου βρίσκονται τα PLC.

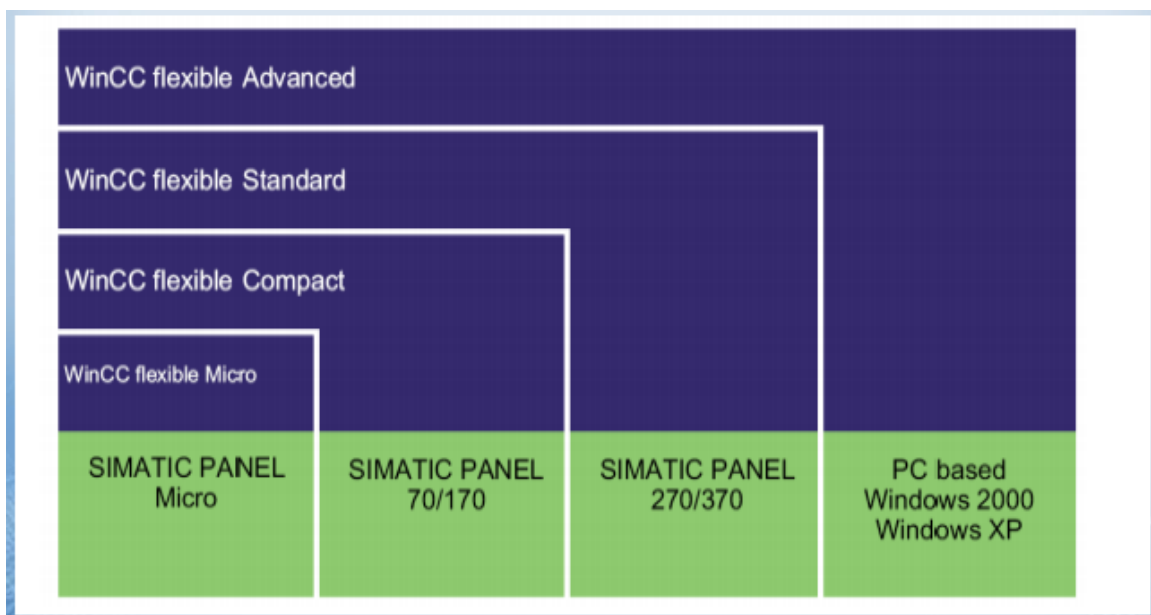
### 2.1.7 Συστήματα PLC – HMI

Σήμερα όπου οι βιομηχανικές διεργασίες γίνονται όλο και περισσότερο πολύπλοκες, δημιουργείται ανάγκη για δημιουργία ενός απλού και κατανοητού περιβάλλοντος διασύνδεσης του χρήστη με την διεργασία ώστε με απλές ενέργειες να μπορεί να έχει τον έλεγχο και την εποπτεία. Αυτό το περιβάλλον διασφαλίζεται από τα **HMI(human machine interface) συστήματα**. Το PLC αποτελεί τη μονάδα που ελέγχει τη διεργασία ενώ το HMI τη μονάδα που διασυνδέει το PLC με τον χειριστή.

Οι δυνατότητες που έχουμε με το HMI είναι πολλές, μεταξύ των οποίων είναι:

- Απεικόνιση και Εποπτεία της διεργασίας.
- Έλεγχος της διεργασίας από τον χρήστη.
- Απεικόνιση προειδοποιητικών μηνυμάτων-σφαλμάτων της εγκατάστασης.
- Καταγραφή τιμών διεργασίας και σφαλμάτων.
- Εκτύπωση τιμών διεργασίας και σφαλμάτων.
- Διαχείριση διάφορων παραμέτρων της διεργασίας.

Το WinCC flexible Engineering System της εταιρείας SIEMENS αποτελεί λογισμικό διαμόρφωσης. Αναλόγως της έκδοσης μπορούν να προγραμματιστούν και οι αντίστοιχες συσκευές Simatic HMI.

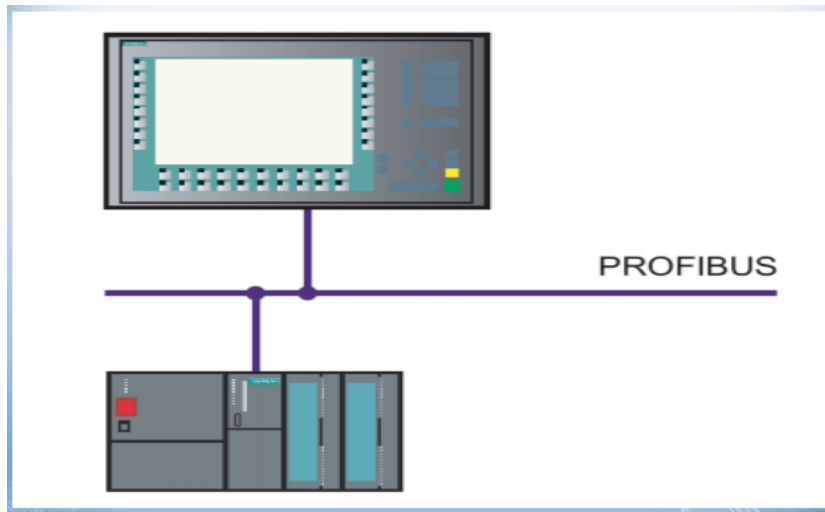


Σχήμα 5: Εκδόσεις του συστήματος WinCC flexible

### 2.1.8 Συνδεσμολογίες PLC με HMI

➤ Έλεγχος με έναν HMI εξοπλισμό

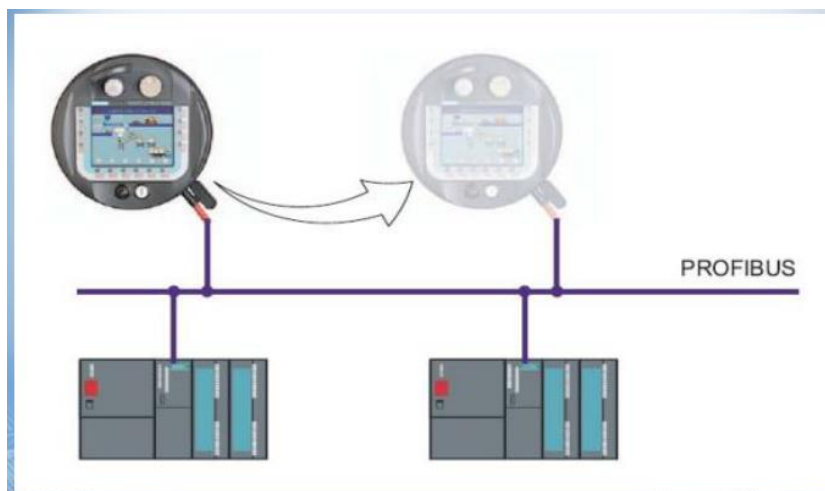
Μία οθόνη HMI συνδέεται απευθείας στο PLC μέσω του bus της διεργασίας και αναφέρεται ως single user system.



Σχήμα 6: Έλεγχος PLC με HMI

➤ PLC με αρκετές HMI συσκευές

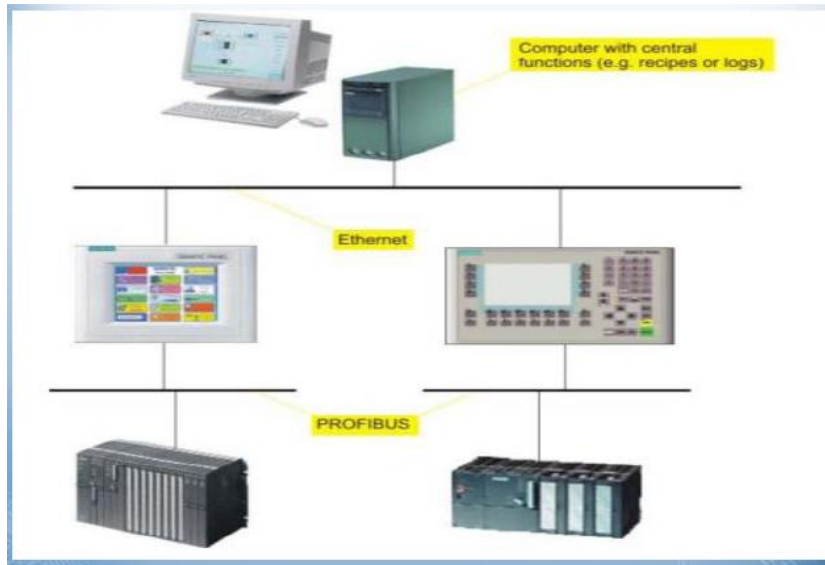
Αρκετές HMI οθόνες συνδέονται σε ένα ή περισσότερα PLC μέσω bus διεργασίας όπως Profibus, Ethernet. Με την διάταξη αυτή, μπορεί να ελέγχεται η γραμμή παραγωγής από διαφορετικά σημεία.



Σχήμα 7: Έλεγχος με αρκετές οθόνες HMI

➤ **Σύστημα HMI με κεντρικές λειτουργίες**

Ένα κεντρικό σύστημα HMI είναι συνδεδεμένο με ένα PC μέσω δικτύου Ethernet. Το PC εκτελεί κεντρικές λειτουργίες, για παράδειγμα τη διαχείριση των συσκευών. Τα απαραίτητα data records παρέχονται από τα HMI χαμηλότερης ιεραρχίας.



**Σχήμα 8:** Κεντρικό σύστημα HMI

## 2.2 Συστήματα SCADA

Η βιομηχανική πληροφορική αποτελεί έναν από τους πιο σημαντικούς και ραγδαία εξελισσόμενους κλάδους της Πληροφορικής. Η απαίτηση για μία πιο αποτελεσματική διαχείριση όλων των πληροφοριών των διαφόρων τμημάτων μίας επιχείρησης οδήγησε στην ανάπτυξη πιο εξειδικευμένων εφαρμογών βιομηχανικής πληροφορικής. Η ανταλλαγή πληροφορίας μεταξύ όλων των τμημάτων προϋποθέτει τη σύνδεση όλων των επιμέρους εφαρμογών, σε ένα κεντρικό Διαχείρισης και Πληροφορικής (Management Information System).

Τα **συστήματα SCADA** (Supervisory Control And Data Acquisition) αποτελούν μία εφαρμογή της Βιομηχανικής Πληροφορικής για την εποπτεία της παραγωγής. Κάθε διεργασία της παραγωγής χαρακτηρίζεται από κάποιες παραμέτρους οι οποίες είναι κρίσιμες και παίζουν καθοριστικό ρόλο στην παραγωγική διαδικασία όσο και στην ποιότητα των παραγόμενων προϊόντων (π.χ. θερμοκρασία, πίεση, χρόνος κ.α.). Τα συστήματα SCADA παρακολουθούν on-line μέσω των προγραμματιζόμενων λογικών ελεγκτών (PLC) και καταγράφουν συνεχώς σε

ηλεκτρονικούς υπολογιστές όλες τις κρίσιμες παραμέτρους της παραγωγικής διαδικασίας, για την επίτευξη εποπτείας σε πραγματικό χρόνο.

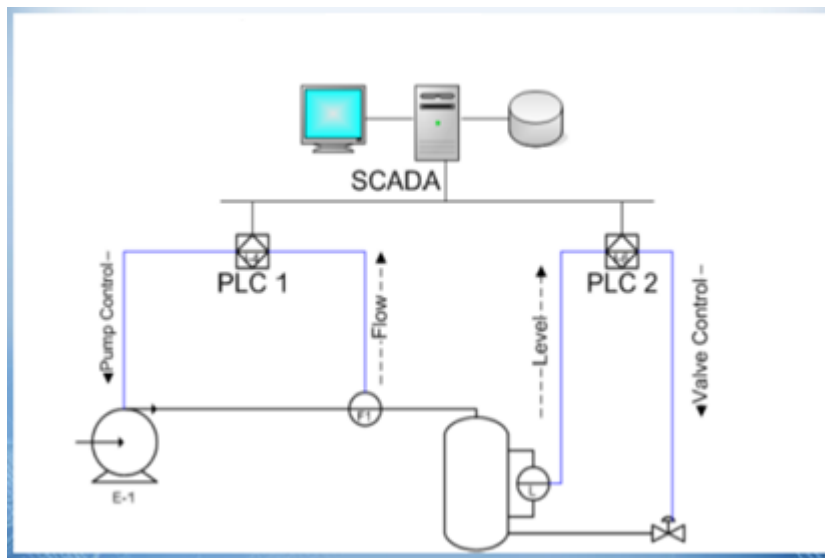
Ένα σύστημα SCADA περιλαμβάνει απεικόνιση σε μιμικά διαγράμματα όλης της διεργασίας της παραγωγής, ενδείξεις των τιμών, των μετρούμενων μεγεθών, συνεχής συλλογή και αποθήκευση δεδομένων σε Η/Υ, γνωστοποίηση σφαλμάτων κ.α. Επίσης είναι σημαντική η δυνατότητα παρακολούθησης του συστήματος μέσω κατάλληλα διαμορφωμένων σελίδων του διαδικτύου(web client).

Οι κύριες λειτουργίες ενός συστήματος SCADA είναι οι παρακάτω:

- 1. Συλλογή δεδομένων** από PLC και τις απομακρυσμένες μονάδες τηλεμετρίας(Remote Telemetry Unit- RTU). Οι RTU εγκαθίστανται σε απομακρυσμένα σημεία με σκοπό την αποστολή και λήψη κρίσιμων παραμέτρων. Όλα τα επιθυμητά δεδομένα μεταδίδονται προς το σύστημα SCADA μέσω του δικτύου βιομηχανικού αυτοματισμού.
- 2. Αποθήκευση των πληροφοριών** σε βάση δεδομένων και αναπαράσταση τους μέσω γραφημάτων.
- 3. Ανάλυση δεδομένων** και ειδοποίηση του προσωπικού σε περιπτώσεις σφάλματος ή συναγερμού. Όταν υπάρχει τιμή εκτός των κανονικών ορίων το σύστημα SCADA ειδοποιεί με οπτική ή ακουστική σήμανση τους χειριστές, ώστε να αποφευχθούν δυσάρεστες επιπτώσεις.
- 4. Έλεγχος κλειστού βρόχου διεργασίας.** Υπάρχει η δυνατότητα εφαρμογής τεχνικών ελέγχου, χειροκίνητες ή αυτόματες.
- 5. Γραφική απεικόνιση** των τμημάτων μίας διεργασίας σε μιμικά διαγράμματα και παρουσιάσεις των δεδομένων σε πεδία.  
Τα μιμικά διαγράμματα απεικονίζουν ρεαλιστικά τα τμήματα της διεργασίας με στόχο την εύκολη εποπτεία αλλά και κατανόηση των δεδομένων από τους χειριστές του συστήματος.
- 6. Καταγραφή όλων των συμβάντων** για τη δημιουργία ιστορικού. Σε κάθε βιομηχανία υπάρχει καταγραφή όλων των κρίσιμων συμβάντων-παραμέτρων. Παλιότερα αυτό γινόταν με χειρόγραφη καταγραφή, αλλά σήμερα την ευθύνη αυτή έχει αναλάβει η βάση δεδομένων του SCADA.
- 7. Υποστήριξη διπλού υπολογιστικού συστήματος SCADA**, με αυτόματη εναλλαγή, αν αυτό κρίνεται απαραίτητο. Σε διεργασίες που υπάρχει υψηλή επικινδυνότητα πρέπει να ελαχιστοποιηθεί όσο το δυνατόν περισσότερο η εμφάνιση σφάλματος λόγω βλάβης του εξοπλισμού. Για αυτό το λόγο τα

συστήματα SCADA υποστηρίζουν δεύτερο υπολογιστικό σύστημα που αναλαμβάνει σε περίπτωση σφάλματος.

8. **Μεταφορά δεδομένων** σε άλλα συστήματα του κεντρικού συστήματος πληροφόρησης και διαχείρισης της εταιρείας.
9. **Έλεγχος της πρόσβασης χειριστών** στα διαφορετικά επίπεδα του συστήματος SCADA.
10. **Ειδικές εφαρμογές λογισμικού** όπως για παράδειγμα εκτέλεση κώδικα C++ ή ανάπτυξη ευφυών συστημάτων.



Σχήμα 9: Σύστημα ελέγχου με PLC και SCADA

Η συλλογή δεδομένων ξεκινά από το PLC ή το RTU και περιλαμβάνει τις ενδείξεις των μετρητικών οργάνων καθώς και τις ενδείξεις από την κατάσταση(status) του εξοπλισμού. Τα δεδομένα αυτά στη συνέχεια καταρτίζονται και διαμορφώνονται με τέτοιο τρόπο, ώστε ο χειριστής που βρίσκεται στην αίθουσα ελέγχου χρησιμοποιώντας το SCADA να μπορεί να πάρει εποπτικές αποφάσεις για να προσαρμόσει ή να παρακάμψει τον κανονικό έλεγχο των PLC ή RTU. Επίσης τα δεδομένα μπορούν να αποθηκευτούν σε μία βάση δεδομένων, στην οποία καταγράφονται κρίσιμοι παράμετροι για την εγκατάσταση με το πέρασμα του χρόνου και έτσι υπάρχει η δυνατότητα στους διαχειριστές του συστήματος να ανατρέξουν στο ιστορικό και να προχωρήσουν σε βελτιώσεις ή προσαρμογές του συστήματος.

Τα συστήματα SCADA χρησιμοποιούνται για:

- **Βιομηχανικές διεργασίες**, στις οποίες περιλαμβάνονται διεργασίες στον τομέα της κατασκευής, της παραγωγής προϊόντων και της παραγωγής ηλεκτρικής ενέργειας.
- **Διεργασίες υποδομής**, στις οποίες περιλαμβάνονται δημόσιες ή ιδιωτικές επιχειρήσεις όπως στην επεξεργασία νερού και διανομή του, η συλλογή και επεξεργασία λυμάτων, η μεταφορά ηλεκτρικής ενέργειας καθώς και η διανομή της, τα αιολικά πάρκα και τα μεγάλα επικοινωνιακά συστήματα.
- Διεργασίες σε εγκαταστάσεις όπως αεροδρόμια, κτίρια, πλοία και διαστημικούς σταθμούς.

### 2.2.1 Οι βασικοί στόχοι του SCADA

Ένα σύστημα εποπτικού ελέγχου και συλλογής δεδομένων (SCADA) είναι ένα σύστημα ευρέως διαδεδομένο βασισμένο στη χρήση υπολογιστή που αρχικά χρησιμοποιήθηκε για έλεγχο και την παρακολούθηση συνθηκών λειτουργίας μίας απομακρυσμένης εγκατάστασης.

Για το παράδειγμα ενός δικτύου διανομής νερού, οι αντικειμενικοί στόχοι είναι οι παρακάτω:

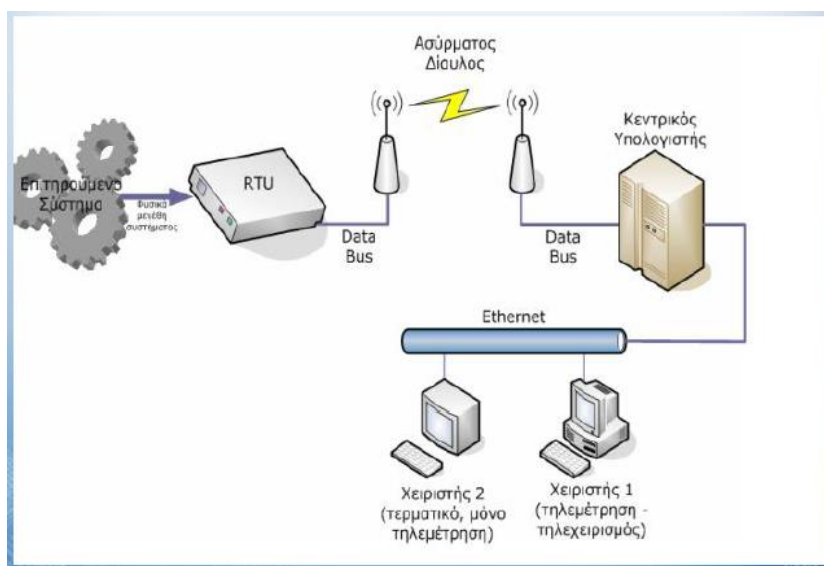
- Συνεχής παρακολούθηση του συστήματος.
- Έλεγχος όλου του δικτύου καθώς και η εξασφάλιση ότι η απαιτούμενη απόδοση επιτυγχάνεται.
- Μείωση των επιπέδων των λειτουργικών αποθεμάτων μέσω της αυτοματοποίησης ή μέσω λειτουργίας του δικτύου από μία απλή κεντρική εγκατάσταση.
- Αποθήκευση των δεδομένων συμπεριφοράς του συστήματος και επομένως της πλήρους συμμόρφωσης στις απαιτήσεις των προτύπων.
- Παροχή πληροφοριών για την απόδοση του δικτύου και εφαρμογή αποτελεσματικών διαδικασιών διαχείρισης για το δίκτυο.
- Εφαρμογή αποτελεσματικής λειτουργίας του δικτύου μειώνοντας την ανάγκη για συνεχείς επισκέψεις στις απομακρυσμένες εγκαταστάσεις και ουσιαστική μείωση της κατανάλωσης ενέργειας κατά την λειτουργία.
- Παροχή ενός συστήματος ελέγχου που θα επιτρέπει να επιτυγχάνονται αντικειμενικοί σκοποί για το δίκτυο.

- Παροχή ενός συστήματος έκτακτης ειδοποίησης που θα επιτρέπει την διάγνωση των σφαλμάτων και των συναγερωμών από ένα κεντρικό σημείο, ώστε να επιτυγχάνονται οι διαδικασίες επίλυσης των προβλημάτων από εξειδικευμένο προσωπικό και επιπλέον να αποφεύγονται περιστατικά που μπορεί να είναι επιζήμια για το δίκτυο.

### 2.2.2 Τα βασικά μέρη ενός συστήματος SCADA

Ένα συνηθισμένο σύστημα SCADA χρησιμοποιεί σαν κεντρικό πυρήνα έναν κεντρικό ηλεκτρονικό υπολογιστή, αρκετά μεγάλης υπολογιστικής ισχύος, στον οποίο βρίσκεται εγκατεστημένο το λογισμικό SCADA, όπως και το πρόγραμμα της εκάστοτε εφαρμογής.

Η ζητούμενη τηλεμετρία στον επιθυμητό χώρο, επιτυγχάνεται με την εγκατάσταση διάφορων σταθμών τηλεμετρίας RTU (Remote Telemetry Units). Αυτοί οι σταθμοί διαβάζουν τις τιμές διάφορων μεγεθών που είναι κρίσιμες για την εφαρμογή όπως θερμοκρασία, πίεση, τάση κτλ., και τις μετατρέπουν σε ηλεκτρικά σήματα. Αυτά τα σήματα τα μεταδίδουν ενσύρματα ή ασύρματα με τις κατάλληλες τηλεπικοινωνιακές ζεύξεις στον κεντρικό υπολογιστή, ανά τακτά χρονικά διαστήματα. Από εκεί και μετά, αρχίζει η παρακολούθηση και επεξεργασία τους από τους χρήστες του κεντρικού υπολογιστή SCADA και εξάγονται χρήσιμα συμπεράσματα και την λειτουργία της εκάστοτε διεργασίας.



Σχήμα 10: Τοπολογία συστήματος SCADA με RTU



Τα δομικά στοιχεία ενός συστήματος SCADA είναι τα ακόλουθα:

- Ένας κεντρικός υπολογιστικός σταθμός (Master Station Computer, MTU).
- Το τμήμα της επικοινωνίας (radio, καλωδιακή, τηλεφωνική).
- Το ελεγχόμενο σύστημα, όπως ένα όργανο μέτρησης (Field Instrumentation).
- RTU που κωδικοποιούν και αποκωδικοποιούν ηλεκτρικά σήματα από τον πραγματικό κόσμο.

Ένα σύστημα SCADA αποτελείται επίσης από τα παρακάτω υποσυστήματα:

- Ένα σύστημα ανθρώπινης διεπαφής με μηχανή (HMI, Human Machine Interface), που έχει σαν σκοπό να απεικονίζει στον χειριστή τα δεδομένα της γραμμής ώστε να μπορεί να τα ελέγχει καθ' όλη τη διάρκεια της διεργασίας.
- Ηλεκτρονικό υπολογιστή όπου συλλέγει όλες τις πληροφορίες και στέλνει την κατάλληλη ανατροφοδότηση.
- Τηλεχειριζόμενες τερματικές μονάδες, οι οποίες είναι συνδεδεμένες με αισθητήρες σε όλη τη διαδικασία, ανταλλάσσοντας έτσι σήματα από τους αισθητήρες στον κεντρικό υπολογιστή.
- Προγραμματιζόμενους λογικούς ελεγκτές (PLC).
- Την επικοινωνιακή υποδομή του συστήματος, η οποία συνδέει όλα τα παραπάνω μεταξύ τους.

**Συσκευές δεδομένων:** Αυτές οι συσκευές όπως οι μετρητές της στάθμης μιας δεξαμενής, οι μετρητές ροής του νερού, αισθητήρες θερμοκρασίας, οι μετρητές κατανάλωσης ενέργειας και οι αισθητήρες πίεσης όλοι μαζί παρέχουν πληροφορίες χρήσιμες σε έναν χειριστή ως προς το πόσο καλά λειτουργεί ένα σύστημα διανομής νερού. Επιπλέον, εξοπλισμοί όπως οι ηλεκτροβαλβίδες, οι πίνακες ελέγχου του μηχανισμού και οι ηλεκτρονικές εγκαταστάσεις μπορούν να χρησιμοποιηθούν για να διαμορφώσουν το σύστημα SCADA και να βοηθήσουν στην αυτοματοποίηση της διαδικασίας της διανομής του νερού.

Παρόλα αυτά, πριν από αυτοματοποιημένη ή απομακρυσμένη παρακολούθηση, οι πληροφορίες που περνούν από και προς τις συσκευές δεδομένων θα πρέπει να μετατραπούν σε μία μορφή ώστε να είναι συμβατή με την γλώσσα του

συστήματος SCADA. Για να επιτευχθεί το παραπάνω, απαιτείται κάποια μορφή ηλεκτρονικών δεδομένων.

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

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

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

Με την πάροδο του χρόνου υπήρξε η απαίτηση να επηρεάσουν το πρόγραμμα μέσα στο PLC από ένα απομακρυσμένο σταθμό. Αυτό στην ουσία είναι μέρος του εποπτικού ελέγχου(Supervisory Control) από το ακρωνύμιο του SCADA. Επομένως, οι κατασκευάστριες εταιρείες των PLC και RTU ανταγωνίζονται στην ίδια αγορά.

Σαν αποτέλεσμα των παραπάνω εξελίξεων, η γραμμή μεταξύ των RTU και PLC έχει εξαφανιστεί και η ορολογία είναι προφανώς εναλλακτική.

### **2.2.3 SCADA και Αυτόματος έλεγχος – Σύγκριση DCS και PLC**

Όμοια συστήματα με τα SCADA μπορούμε να δούμε σε βιομηχανίες, εγκαταστάσεις επεξεργασίας κτλ. Αυτά τα συστήματα είναι γνωστά ως συστήματα καταμεμημένου ελέγχου (distributed control system, DCS).

Το DCS ορίζεται ως ένα σύστημα το οποίο είναι ενιαίο λειτουργικά, αποτελείται από άλλα υποσυστήματα που μπορεί να είναι στη φυσική τους μορφή απομακρυσμένα και ξεχωριστά μεταξύ τους. Το DCS έχει όμοιες λειτουργίες με αυτές των SCADA συστημάτων αλλά το πεδίο συλλογής δεδομένων (μονάδες ελέγχου) συχνά βρίσκονται σε μία πιο περιορισμένη περιοχή. Η μεταξύ τους επικοινωνία μπορεί να γίνεται μέσω τοπικού δικτύου(Local area network, LAN) και είναι γρήγορη αλλά και αξιόπιστη.

Ένα σύστημα DCS περιλαμβάνει συνήθως έλεγχο κλειστού βρόχου. Από την άλλη, τα συστήματα SCADA, συνήθως καλύπτουν μεγαλύτερες γεωγραφικές εκτάσεις και βασίζονται πιο πολύ σε μία ποικιλία συστημάτων επικοινωνίας τα οποία συνήθως είναι λιγότερο αξιόπιστα από το τοπικό δίκτυο(LAN). Σε αυτές τις περιπτώσεις, ο έλεγχος κλειστού βρόχου είναι λιγότερο επιθυμητός.

Οι στόχοι στα συστήματα SCADA και DCS είναι διαφορετικοί. Υπάρχει η δυνατότητα για ένα σύστημα να είναι ικανό να εκτελεί και λειτουργίες SCADA και DCS, αλλά πολύ λίγα από αυτά τα συστήματα έχουν σχεδιαστεί με αυτόν τον τρόπο.

Τα συστήματα DCS είναι προσανατολισμένα στον έλεγχο διεργασιών. Το σημαντικό για αυτά τα συστήματα είναι η διεργασία, ενώ η παρουσίαση των δεδομένων στο χειριστή είναι απλώς ένα μέρος της λειτουργίας τους. Από την άλλη τα SCADA είναι προσανατολισμένα στη συλλογή των δεδομένων. Το σημαντικό για αυτά είναι οι χειριστές του καθώς και το κέντρο ελέγχου. Ο απομακρυσμένος εξοπλισμός (RTU) βρίσκεται στα συστήματα SCADA για να παρέχει πληροφορίες – ανεξάρτητα με το αν εκτελεί πολύπλοκες λειτουργίες ελέγχου ή αν είναι τελικός μέρος του συστήματος DCS.

Το SCADA έχει την δυνατότητα να λειτουργεί σε έναν πιθανόν αργό και μη αξιόπιστο δίκτυο, και παράλληλα να διατηρεί μια βάση δεδομένων με ιστορικό τιμών, τις οποίες μπορεί να προβάλει κατ' απαίτηση του χρήστη. Επιπλέον μπορεί να υπάρχει η ανάγκη να αξιολογεί τα δεδομένα και ώστε να αποφασίζει αν η πληροφορία αυτή είναι χρήσιμη ή περιττή. Αντίθετα, το σύστημα DCS είναι πάντα συνδεδεμένο στην πηγή πληροφοριών του, οπότε και δεν χρειάζεται συνήθως ιστορικό τιμών, ενώ η πληροφορία που λαμβάνει δε θεωρείται περιττή και επεξεργάζεται πάντα.

Οι διαφορές αυτές οδήγησαν σε διαφορετική σχεδιαστική φιλοσοφία των δύο συστημάτων. Τα SCADA απαιτούν μεγαλύτερη πολυπλοκότητα για τον χειρισμό

και την συλλογή δεδομένων, ενώ από άλλη τα DCS στις λειτουργίες ελέγχου διεργασιών και βρόχων.

Μία ακόμα βασική διαφορά στο επίπεδο ελέγχου που ασκείται από τα δύο συστήματα, σχετίζεται με πως αντιμετωπίζουν την πληροφορία τα δύο συστήματα. Το DCS είναι οδηγούμενο από τις καταστάσεις, ενώ το SCADA από τα γεγονότα. Το DCS το ενδιαφέρει κυρίως η πρόοδος της διαδικασίας, ενώ το SCADA τα γεγονότα που προκύπτουν από αυτήν.

Σαν συμπέρασμα, μπορούμε να πούμε ότι τα συστήματα DCS και SCADA είναι διαφορετικά σχεδιασμένα και εφαρμόζονται κάτω από διαφορετικές απαιτήσεις, ενώ από την άλλη πολύ συχνά απαιτείται ο συνδυασμός και η συνεργασία τους. Εδώ πρέπει να σημειωθεί ότι το σύστημα SCADA δεν εκτελεί έλεγχο της προόδου της διεργασίας, αλλά καταγράφει και εποπτεύει την κατάσταση και τα γεγονότα που θα προκύψουν από αυτές.

#### **2.2.4 SCADA και HMI**

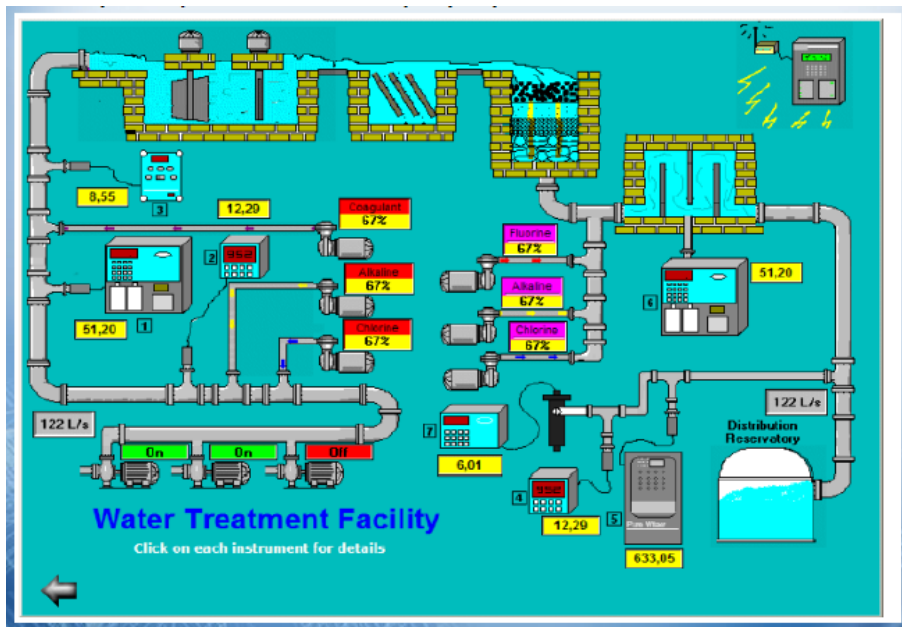
Η διασύνδεση ανθρώπου μηχανής (human machine interface, HMI), είναι ο εξοπλισμός που παρουσιάζει δεδομένα από κάποια διεργασία σε έναν χειριστή και μέσω αυτής, ο χρήστης έχει την δυνατότητα να ελέγξει και να εποπτεύσει τη διεργασία.

Τα συστήματα HMI αποτελούν μέρος λειτουργίας των SCADA που αλληλεπιδρά με τον τελικό χειριστή του συστήματος. Αποτελούνται συνήθως από μία οθόνη μέσω της οποίας υπάρχει η απεικόνιση της διεργασίας, πάνω στην οποία εμφανίζονται τιμές μεταβλητών, καταστάσεις ή και διαγράμματα. Επιπλέον τα συστήματα αυτά έχουν την δυνατότητα καταγραφής ιστορικού μεταβλητών, ειδικών διαγραμμάτων κτλ. Οι πληροφορίες αυτές αντλούνται από τη βάση δεδομένων του SCADA, γι' αυτό όπως γίνεται κατανοητό τα συστήματα HMI και SCADA είναι αλληλένδετα συνδεδεμένα μεταξύ τους και για αυτό το λόγο συχνά δεν διαχωρίζονται.

Οι περισσότεροι κατασκευαστές συστημάτων SCADA, στην πραγματικότητα ενσωματώνουν την δυνατότητα ανάπτυξης εφαρμογών HMI, στις υπηρεσίες ή στα πακέτα SCADA. Στην ουσία μία εφαρμογή HMI μπορεί να κατασκευαστεί ανεξάρτητα από τον πυρήνα καταγραφής στο SCADA ώστε να αντλεί πληροφορίες από αυτό και να το χρησιμοποιεί ώστε να κάνει τους απαραίτητους ελέγχους.

Όπως αναφέρθηκε παραπάνω το HMI χρησιμοποιεί συνήθως γραφικό τρόπο για την παρουσίαση πληροφοριών στον χρήστη. Ο χειριστής μπορεί να δει σχηματική παρουσίαση της διεργασίας η οποία ελέγχεται. Για παράδειγμα η εικόνα μιας αντλίας η οποία επικοινωνεί με ένα σωλήνα, μπορεί να δείξει στον χειριστή ότι η αντλία δουλεύει καθώς και πόσο υγρό αντλεί από το σωλήνα, την συγκεκριμένη χρονική στιγμή. Ο χειριστής έχει την δυνατότητα στη συνέχεια να κλείσει την αντλία. Στο HMI υπάρχουν διάφορα γραφικά στοιχεία τα οποία χρησιμοποιούνται για να απεικονίσουν τα διάφορα στοιχεία της εγκατάστασης τα οποία είναι προς έλεγχο.

Το πακέτο HMI για το σύστημα SCADA, περιλαμβάνει ένα πρόγραμμα σχεδιασμού, το οποίο μπορεί να χρησιμοποιήσει ο προγραμματιστής ή ο συντηρητής του συστήματος ώστε να αλλάξει τον τρόπο με τον οποίο οι διάφορες μεταβλητές παρουσιάζονται στο σύστημα. Η απεικόνιση της διαδικασίας, μπορεί να είναι εξαιρετικά απλή (προβάλλεται η κατάσταση ON ή OFF ενός φαναριού σε σύστημα SCADA) όπως είναι η κατάσταση φαναριού είτε σύνθετη όπως η απεικόνιση όλων των δεξαμενών σε ένα τοπικό δίκτυο ύδρευσης.



**Σχήμα 11:** Μια εφαρμογή HMI της Ellipse Software για την εποπτεία υδραγωγείου

Ένα σημαντικό σημείο των περισσότερων εφαρμογών SCADA είναι η διαχείριση συναγερμών. Το σύστημα παρακολουθεί συνεχώς κατά πόσο ορισμένες συνθήκες συναγερμού (Alarm) ικανοποιούνται, ώστε να προσδιορίσει πότε ένα

συμβάν δημιουργείται. Μόλις εντοπιστεί ένα τέτοιο συμβάν, πραγματοποιούνται μία ή περισσότερες ενέργειες, όπως η ενεργοποίηση ενδείξεων συναγερμού(alarm) ή ίσως η αποστολή SMS μηνύματος για την άμεση ειδοποίηση τεχνικού προσωπικού ή ίσως η αποστολή ενός email.

Σε αρκετές περιπτώσεις ο χειριστής του SCADA, πρέπει να αναγνωρίσει το συμβάν του συναγερμού ώστε να σταματήσουν οι σημάσεις που έχουν ενεργοποιηθεί (οπτικές ή ηχητικές). Ωστόσο, για να μην εμφανίζονται οι ειδοποιήσεις αυτές, θα πρέπει η αιτία ενεργοποίησης να μην υπάρχει πλέον.

Οι προϋποθέσεις για ενεργοποίηση ενός συναγερμού μπορεί έμμεσες ή άμεσες. Άμεσες είναι για παράδειγμα, όταν η μεταβλητή ενεργοποίησης ενός συναγερμού είναι μία ψηφιακή μεταβλητή, η οποία υπολογίζεται με κάποιο αλγόριθμο και η τιμή της εξαρτάται από άλλες ψηφιακές ή αναλογικές μεταβλητές. Έμμεσες είναι για παράδειγμα, όταν το σύστημα SCADA παρακολουθεί αυτόματα εάν η τιμή κάποιας αναλογικής μεταβλητής βρίσκεται εκτός από τα ορισμένα για την μεταβλητή αυτή.

Παραδείγματα ενδείξεων συναγερμού είναι η ενεργοποίηση μιας ηχητικής σήμανσης, εμφάνιση παραθύρων στην οθόνη του SCADA, τα οποία ενημερώνουν τον χρήστη για τον συναγερμό, μέσω μηνυμάτων τα οποία αλλάζουν χρώμα με την ύπαρξη κάποιου συναγερμού. Σε κάθε περίπτωση, ο ρόλος της παραπάνω ένδειξης είναι να τραβήξει την προσοχή του χειριστή στα σημεία που ελέγχεται η διεργασία ώστε να ληφθούν τα απαραίτητα μέτρα.

### **2.2.5 Συστήματα λογισμικού**

Ένα σημαντικό κομμάτι κάθε συστήματος SCADA είναι το λογισμικό του υπολογιστή που χρησιμοποιείται μέσα στο σύστημα. Εδώ, το πιο προφανές εργαλείο λογισμικού είναι η κοινή επιφάνεια του χειριστή ή το πακέτο MMI/HMI(Man Machine Interface/Human Machine Interface).

Παρόλο αυτά, λογισμικά κάποιας μορφής κυριαρχούν σε όλα τα επίπεδα ενός συστήματος SCADA. Ανάλογα με το μέγεθος και τη φύση της εφαρμογής, το λογισμικό SCADA είναι ένα σημαντικό κόστος κατά την ανάπτυξη, συντήρηση αλλά και επέκταση. Όταν το λογισμικό είναι καλά σχεδιασμένο, ορισμένο και ελεγμένο, είναι αρκετά πιθανό να παραχθεί ένα επιτυχημένο σύστημα SCADA. Σε κάθε φάση του project, οι χαμηλές αποδόσεις είναι πολύ εύκολο να δημιουργήσουν αποτυχία για το SCADA.

Πολλά από τα συστήματα SCADA χρησιμοποιούν ιδιόκτητα λογισμικά πάνω στα οποία αναπτύσσεται η SCADA εφαρμογή. Το ιδιόκτητο λογισμικό συχνά διαμορφώνεται σε συγκεκριμένη πλατφόρμα hardware άλλου κατασκευαστή ή που μπορεί να μην συνδέεται με το λογισμικό. Υπάρχει μεγάλη ποικιλία από **Commercial-off-the-shelf** (COTS) προϊόντα λογισμικού τα οποία είναι επίσης διαθέσιμα, μερικά από τα οποία είναι κατάλληλα για την απαιτούμενη εφαρμογή SCADA. Το COTS λογισμικό συνήθως δίνει μία ευελιξία, και μπορεί να συνδέεται με διάφορους τύπους λογισμικών και hardware. Γενικότερα, ο στόχος για τα ιδιόκτητα λογισμικά είναι η λειτουργικότητα διαδικασιών και ελέγχων, ενώ τα COTS λογισμικά δίνουν έμφαση για την συμβατότητα μίας ποικιλίας εξοπλισμών και συσκευών. Είναι σημαντική επομένως η διασφάλιση ότι έχει γίνει ο κατάλληλος σχεδιασμός για την επιλογή των συστημάτων λογισμικού που είναι κατάλληλα για κάθε νέο σύστημα SCADA.

Τέτοια προϊόντα λογισμικού χρησιμοποιούνται για τα παρακάτω μέρη ενός συστήματος SCADA:

- **Λειτουργικό σύστημα του Κεντρικού υπολογιστή:** Το λογισμικό που χρησιμοποιείται για έλεγχο του hardware στον κεντρικό υπολογιστή μπορεί να έχει ως βάση το UNIX ή άλλα δημοφιλή λειτουργικά συστήματα.
- **Λειτουργικό σύστημα για τερματικό σταθμό:** Είναι το λογισμικό που χρησιμοποιείται για το hardware του κεντρικού υπολογιστή. Το λογισμικό αυτό συνήθως είναι το ίδιο με αυτό του λειτουργικού συστήματος του κεντρικού υπολογιστή. Το λογισμικό αυτό, μαζί με αυτό του κεντρικού υπολογιστή συνήθως συνεισφέρει στην δικτύωση των τερματικών σταθμών με τον κεντρικό υπολογιστή.
- **Εφαρμογή στον κεντρικό υπολογιστή:** Το λογισμικό που χειρίζεται την υποδοχή των δεδομένων καθώς και την μετάδοση από και προς τα RTU και τον κεντρικό υπολογιστή. Το λογισμικό παρέχει επιπλέον την γραφική επιφάνεια όπου προσφέρει τις απεικονίσεις και τις διάφορες λειτουργίες ελέγχου.
- **Εφαρμογή για τους τερματικούς σταθμούς:** Είναι η εφαρμογή αυτή, που επιτρέπει στους χρήστες την πρόσβαση σε διαθέσιμες πληροφορίες της εφαρμογής στον κεντρικό υπολογιστή. Συνοδεύει συνήθως το λογισμικό που χρησιμοποιείται στους κεντρικούς υπολογιστές.

- **Οδηγοί(drivers) του πρωτοκόλλου επικοινωνιών:** Είναι το λογισμικό που συνήθως ενώνει τον κεντρικό υπολογιστή και τα RTU, και απαιτείται για τον έλεγχο της ερμηνείας των δεδομένων μεταξύ των τερματικών σταθμών και του συστήματος στον κεντρικό υπολογιστή.
- **Λογισμικό διαχείρισης στο δίκτυο επικοινωνιών:** Το λογισμικό αυτό απαιτείται για τον έλεγχο του δικτύου επικοινωνιών και επιτρέπει στα δίκτυα αυτά να ελέγχονται μόνα τους για την απόδοση αλλά και πιθανά σφάλματα.
- **Λογισμικό αυτοματισμού στα RTU:** Το λογισμικό που επιτρέπει στο προσωπικό να σχεδιάζει και να συντηρεί την εφαρμογή που βρίσκεται στα RTU ή τα PLC.

Τα παραπάνω προϊόντα λογισμικού παρέχουν τα κατασκευαστικά κομμάτια για συγκεκριμένα λογισμικά, τα οποία στη συνέχεια πρέπει να καθοριστούν, να σχεδιαστούν να αναπτυχθούν και τέλος να ελεγχθούν για κάθε σύστημα SCADA.

#### 2.2.6 Χειρισμός των δεδομένων SCADA σε περίπτωση προβλημάτων

Τα συστήματα SCADA μπορούν να αντιμετωπίσουν κάποια περίπτωση προβλήματος όπως αναφέρονται παρακάτω:

- **Αποθήκευση δεδομένων στα RTU:** Σε μερικά συστήματα SCADA υπάρχει η δυνατότητα να αποθηκεύονται κάποια δεδομένα στα RTU κάτω από κανονική λειτουργία και μετά περιοδικά αυτά τα δεδομένα σαν μήνυμα ή όταν τα ζητά ο κεντρικός υπολογιστής μπορούν να μεταφέρονται. Σε περιπτώσεις όπου το σύστημα SCADA δεν είναι διαθέσιμο, η μνήμη του RTU χρησιμοποιείται για να αποθηκεύσει πληροφορίες μέχρι το σύστημα να επανέλθει.
- **Αφθονία συστήματος:** Τα περισσότερα συστήματα SCADA ενσωματώνουν κάποια μορφή αφθονίας κατά τον σχεδιασμό τους, όπως είναι τα διπλά κανάλια επικοινωνίας, τα διπλά RTU ή ακόμα και διπλούς υπολογιστές. Τέτοια συστήματα μπορούν να σχεδιαστούν ώστε να διασφαλίσουν μία συνεχή μεταφορά δεδομένων σε περίπτωση σφάλματος ή ακόμα και όταν ο μηχανισμός backup πρέπει να λειτουργήσει χειροκίνητα.



Τα περισσότερα συστήματα SCADA αξιοποιούν έναν συνδυασμό όπως αναφέρθηκε παραπάνω ώστε να διασφαλίσουν την συνεχή συλλογή δεδομένων σε περιπτώσεις βλαβών.

### **2.2.7 Πλεονεκτήματα και εξέλιξη των συστημάτων SCADA**

Τα SCADA είναι εξειδικευμένα συστήματα που σχεδιάζονται για να εξυπηρετούν συγκεκριμένες εφαρμογές (για παράδειγμα αυτοματοποίηση ενός εργοστασίου κατασκευής χαρτιού).

Τα πλεονεκτήματα αλλά και τα οφέλη χρησιμοποίησης ενός συστήματος SCADA στον βιομηχανικό έλεγχο είναι τα παρακάτω:

- Άμεση ενημέρωση της κατάστασης της διεργασίας.
- Η αντιστάθμιση των μεταβλητών ελέγχου της διεργασίας, με σκοπό την διατήρηση των δεδομένων ονομαστικών τιμών και τη διατήρηση των απαιτήσεων της παραγωγικής διεργασίας.
- Έγκαιρη σήμανση βλαβών και του σφάλματος εξοπλισμού στις διάφορες διεργασίες, για να παρέχεται η μέγιστη ασφάλεια του εξοπλισμού και των εργαζομένων.
- Πρόγνωση και διάγνωση για βλάβες εξοπλισμών και έγκαιρο εντοπισμό τους, για τη μέγιστη διαθεσιμότητα του.
- Καταγραφή και αποθήκευση δεδομένων, σχετικά με την παραγωγή και τη διαχείρισή της.
- Καλή λειτουργία των εξοπλισμών, με στόχο την βελτιστοποίηση της χρήσης τους και επομένως της παραγωγικότητάς τους.
- Αύξηση της παραγωγικότητας λόγω καλύτερης αξιοποίησης των δυνατοτήτων των μέσων παραγωγής, π.χ. λειτουργώντας την μονάδα παραγωγής στα ανώτερα όρια της.
- Μείωση στο κόστος παραγωγής ανά μονάδα παραγόμενου προϊόντος, λόγω βέλτιστης χρήσης των εσωτερικών πηγών ενέργειας αλλά και μείωσης του κόστους εργασίας.
- Βελτίωση στην ποιότητα των παραγόμενων προϊόντων λόγω της δυνατότητας να διατηρούνται οι συνθήκες λειτουργίας με στενά όρια ανοχών.

- Ευελιξία στη παραγωγή κάτω από συνεχώς μεταβαλλόμενες συνθήκες αγοράς.

Ο έλεγχος των διεργασιών θα πρέπει να είναι αρκετά γρήγορος, ώστε να υπάρχει επίγνωση της κατάστασης αυτών σε πραγματικό (real time). Αυτό γίνεται εμφανές για το θέμα της ασφάλειας. Η επέμβαση στις διεργασίες σε περιπτώσεις κινδύνου θα πρέπει να είναι άμεση και αποτελεσματική ώστε να μπορεί να υπάρχει αποφυγή μερικής ή ολικής καταστροφής του εξοπλισμού, ακόμα και ανθρώπινης απώλειας. Βέβαια, υπάρχουν και τα προβλήματα του υψηλού κόστους, της έλλειψης τεχνογνωσίας και του εξειδικευμένου προσωπικού. Από την άλλη, τα οφέλη για μία επιχείρηση από την εγκατάσταση συστήματος SCADA είναι σημαντική, ιδίως σε ένα ανταγωνιστικό και παγκοσμιοποιημένο περιβάλλον. Εδώ πρέπει να σημειωθεί και να τονιστεί, πως όσον αφορά τον τομέα του έξυπνου σπιτιού, το κόστος σε επίπεδο τεχνογνωσίας αλλά και εξειδίκευσης είναι σαφώς χαμηλότερο, καθώς οι λειτουργίες που θέλει ο χρήστης να ελέγχει είναι αρκετά απλούστερες.

Ένα άλλο βασικό χαρακτηριστικό των συστημάτων SCADA είναι ότι πρέπει να μπορούν να υποστηρίζουν ένα μεγάλο αριθμό από διαφορετικές συσκευές στο πεδίο. Αυτές οι διαφορετικές συσκευές μπορεί να έχουν ποικίλα τεχνικά χαρακτηριστικά και να χρησιμοποιούν διαφορετικά πρωτόκολλα επικοινωνίας. Ένα σύστημα SCADA επιπλέον πρέπει να μπορεί να παραγοντοποιεί κάθε φορά στα καινούργια δεδομένα της αγοράς όπως νέα τεχνολογικά πρότυπα ή πρωτόκολλα επικοινωνίας, κλπ.

Εξίσου σημαντικό είναι οι παραμετροποιήσεις και οι προσαρμογές αυτές να μπορούν να γίνονται με όσο το δυνατόν λιγότερες επεμβάσεις στον κώδικα λογισμικού του συστήματος. Οι αλλαγές στο λογισμικό είναι μία διαδικασία δαπανηρή και αρκετά χρονοβόρα καθώς μεταξύ άλλων απαιτούν ανθρώπους με εξειδικευμένες γνώσεις καθώς και μεγάλο βαθμό δοκιμών προκειμένου να αποφευχθούν δυσλειτουργίες που μπορεί να προκαλέσουν προβλήματα στην αξιοπιστία αλλά και την ασφαλή λειτουργία του συστήματος.

Όσο καλή και αν είναι μία εφαρμογή λογισμικού δεν μπορεί από μόνη της να εξυπηρετήσει τις ανάγκες ενός συστήματος SCADA κάποιας πολυπλοκότητας. Για να μπορέσει ένα λογισμικό να χρησιμεύσει ως κέντρο ελέγχου ενός τέτοιου συστήματος SCADA και ταυτόχρονα να κυμαίνεται σε λογικό κόστος, είναι πολύ σημαντικό να συγκροτείται από ανεξάρτητες μονάδες, καθεμία από τις οποίες

- Θα μπορεί να εξυπηρετεί μία συγκεκριμένη λειτουργία,
- Θα μπορεί να τροποποιηθεί αλλάζοντας κάποιες παραμέτρους και όχι τον κώδικα, και
- Θα πρέπει να συνεργάζεται με άλλες μονάδες χρησιμοποιώντας γνωστά πρωτόκολλα επικοινωνίας και διαδικασίες.

Τα παραπάνω επιτυγχάνονται με την χρήση κατάλληλων τεχνολογιών ολοκλήρωσης. Στα Windows οι πιο διαδεδομένες τέτοιες τεχνολογίες είναι αυτές, οι οποίες βασίζονται στο πρότυπο **COM** της Microsoft (OLE, Automation, ActiveX). Οι τεχνολογίες αυτές είναι γενικής χρήσεως, μπορούν να χρησιμοποιηθούν σε διαφορετικά λογισμικά. Ειδικά για τα συστήματα SCADA έχει αναπτυχθεί και η τεχνολογία **OPC**(OPC, Open Platform Communications), η οποία επίσης βασίζεται στο πρότυπο **COM**.

Κοινό χαρακτηριστικό των τεχνολογιών αυτών είναι ότι επιτρέπουν σε διαφορετικές μονάδες υλικού ή λογισμικού από διαφορετικούς κατασκευαστές να επικοινωνούν μεταξύ τους και να σχηματίζουν ολοκληρωμένα συστήματα. Αυτό έχει σαν αποτέλεσμα να απλοποιείται σημαντικά η κατασκευή συστημάτων, καθώς μεγάλο μέρος των εργασιών σχεδίασης καθώς και υλοποίησης ανάγεται στην αναζήτηση αλλά και επιλογή των κατάλληλων έτοιμων μονάδων από την αγορά. Έτσι υπάρχει η δυνατότητα να κατασκευαστούν σε σύντομο χρονικό διάστημα και με αρκετό μικρό κόστος πολύπλοκα και εξειδικευμένα συστήματα, τα οποία θα λειτουργούν με ασφάλεια και αξιόπιστα και επιπλέον είναι πλήρως προσαρμοσμένα στις ανάγκες και τις ιδιαιτερότητες της κάθε εφαρμογής.

### 2.2.8 Λογισμικά για την ανάπτυξη συστημάτων SCADA

Σε ένα σύγχρονο σύστημα SCADA, τα προγράμματα του κέντρου ελέγχου δεν κατασκευάζονται από το μηδέν αλλά μπορούν να δημιουργηθούν από προκατασκευασμένα κομμάτια λογισμικού, υποστηρίζοντας τυποποιημένους τρόπους διασύνδεσης. Οι κατασκευαστές δεν παράγουν έτοιμες λύσεις αλλά πλατφόρμες που παρέχουν αρκετή ευελιξία, που μπορούν να προσαρμόζονται στις ανάγκες και στις απαιτήσεις της κάθε εφαρμογής. Το τελικό πρόγραμμα δημιουργείται για κάθε πρόβλημα ξεχωριστά, σχεδιάζοντας, διαμορφώνοντας, συμπληρώνοντας ή διασυνδέοντας τμήματα από τις βασικές αυτές πλατφόρμες.

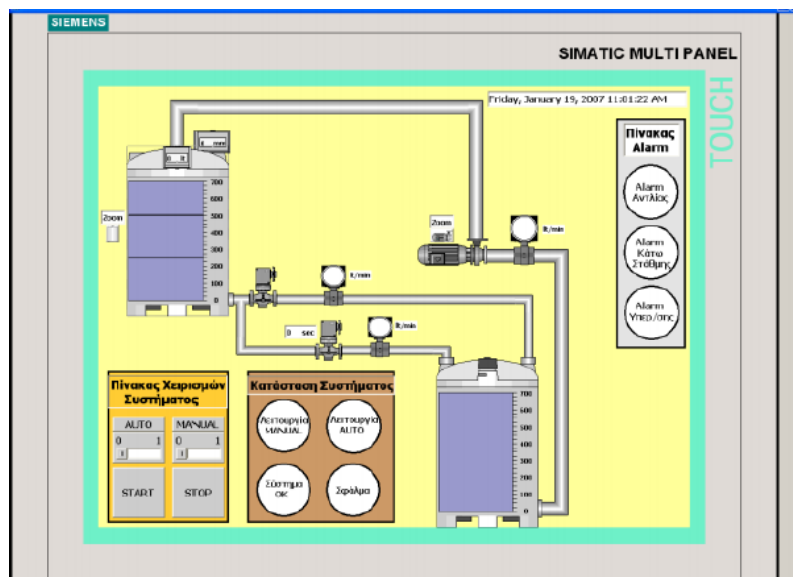
Αυτή η εργασία μπορεί να γίνεται είτε από τον ίδιο τον κατασκευαστή της πλατφόρμας είτε από μία εταιρεία ανάπτυξης ολοκληρωμένων συστημάτων (

system integrator) είτε από τον τελικό χρήστη. Έτσι το κέντρο ελέγχου SCADA, μπορεί να σχεδιάζεται κατά παραγγελία για κάθε εφαρμογή, επιλέγοντας κάθε φορά από ένα σύνολο των υπάρχοντων εργαλείων, εκείνα τα οποία απαιτούνται για κάλυψη των αναγκών της εφαρμογής.

Ένα σύγχρονο σύστημα ελέγχου εποπτείας και συλλογής δεδομένων (SCADA) καλείτε να επεξεργαστεί ένα τεράστιο όγκο πληροφορίας δεδομένων και παράλληλα να δώσει τις εντολές ελέγχου. Εδώ ο ανθρώπινος παράγοντας έχει περιορισμένες ικανότητες στην επεξεργασία μεγάλου όγκου δεδομένων, ωστόσο είναι σε θέση να λάβει σύνθετες και νοήμονες αποφάσεις. Το σύστημα καλείτε να δώσει στο χειριστή του SCADA με κατανοητό για τον τελευταίο τρόπο, έγκυρες πληροφορίες για την σωστή εποπτεία του συστήματος. Τα λογισμικά αυτά εγκαθίστανται είτε σε Η/Υ είτε σε Οθόνες χειρισμών (Industrial panels), ανάλογα με την εφαρμογή.

Παρακάτω αναφέρονται κάποια γνωστά λογισμικά SCADA καθώς και οι αντίστοιχοι κατασκευαστές.

- Simatic WinCC, Siemens
- TIA PORTAL, Siemens
- InTouch, Wonderware
- Labview, National Instruments
- RsView, Rockwell



Σχήμα 12: HMI οθόνη με Simatic WinCC



## **ΚΕΦΑΛΑΙΟ 3**

### **ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΑΥΤΟΜΑΤΙΣΜΟΥ ΜΕ PLC ΚΑΙ HMI #C**

Σε αυτό το κεφάλαιο θα παρουσιαστεί η σχεδίαση και η ανάπτυξη συστήματος αυτοματισμού , με την χρήση ενός προγραμματιζόμενου λογικού ελεγκτή (PLC) S7-1200 της εταιρείας Siemens καθώς και η ανάπτυξη λογισμικού HMI με την γλώσσα προγραμματισμού #C, όπως επίσης και όλων των εργαλείων και των μέσων που χρησιμοποιήθηκαν για την ανάπτυξη του.

Επιπλέον σε αυτό το κεφάλαιο γίνεται και η παρουσίαση του συστήματος.

#### **3.1 Το σύστημα αυτοματισμού**

##### **3.1.1 Γενικά**

Το συγκεκριμένο σύστημα έχει σαν στόχο τον απομακρυσμένο έλεγχο κρίσιμων παραμέτρων για την εκτέλεση μίας διεργασίας. Πιο συγκεκριμένα ο χρήστης θα μπορεί να ελέγχει τις περιβαλλοντολογικές συνθήκες που επικρατούν σε χώρο που εκτελείται μία σημαντική διεργασία παραγωγής. Σε αρκετές βιομηχανίες (τροφίμων, φαρμάκου, κα) οι συνθήκες αυτές λαμβάνονται σοβαρά υπόψη για την εκτέλεση παραγωγικών διεργασιών ώστε να εξασφαλίζουν την ποιότητα των προϊόντων.

Επιπλέον μέσω του συστήματος θα μπορούν να καταγράφονται οι τρέχουσες τιμές του οργάνου αλλά επιπλέον θέτοντας όρια συνθηκών στη θερμοκρασία και υγρασία , ο χρήστης θα μπορεί να ενημερώνεται για τυχόν αποκλείσεις σε αυτές τόσο με άμεση ειδοποίηση όσο και με καταγραφή του αντίστοιχου συμβάντος.

##### **3.1.2 Περιγραφή συστήματος**

Το σύστημα αυτοματισμού αποτελείται από έναν αισθητήρα Θερμοκρασίας-Υγρασίας με οθόνη(display), ο οποίος βρίσκεται εγκατεστημένος σε ένα χώρο

παραγωγής μίας σημαντικής διεργασίας. Ένας προγραμματιζόμενος λογικός ελεγκτής (PLC) καταγράφει συνεχώς τις τιμές του αισθητήρα και παράλληλα ελέγχει τις τιμές αυτές ώστε να είναι μέσα στα επιτρεπτά όρια που έχει θέσει ο χρήστης. Σε περίπτωση που οι τιμές βγουν εκτός ορίων κανονικής λειτουργίας, ο ελεγκτής ενεργοποιεί συναγερμό (alarm) μέσω ηχητικής ειδοποίησης στον χώρο εποπτείας των συστημάτων ώστε να ενημερωθεί άμεσα ο υπεύθυνος και στη συνέχεια να προβεί στις απαραίτητες ενέργειες. Η εποπτεία της εφαρμογής αυτοματισμού γίνεται μέσω του λογισμικού HMI το οποίο είναι εγκατεστημένο στον Η/Υ του χειριστή. Ο χειριστής μέσα από αυτό το λογισμικό έχει συνεχή σύνδεση με το PLC μέσω του οποίου ενημερώνεται τόσο για τις τιμές αισθητήρα αλλά παράλληλα και για την ύπαρξη κάποιου συμβάντος (alarm). Επιπλέον μέσω της εφαρμογής HMI έχει την δυνατότητα να επεξεργαστεί ετεροχρονισμένες τιμές του οργάνου αλλά και συμβάντα καθώς η εφαρμογή καταγράφει κρίσιμους παραμέτρους της διεργασίας σε μία βάση δεδομένων.

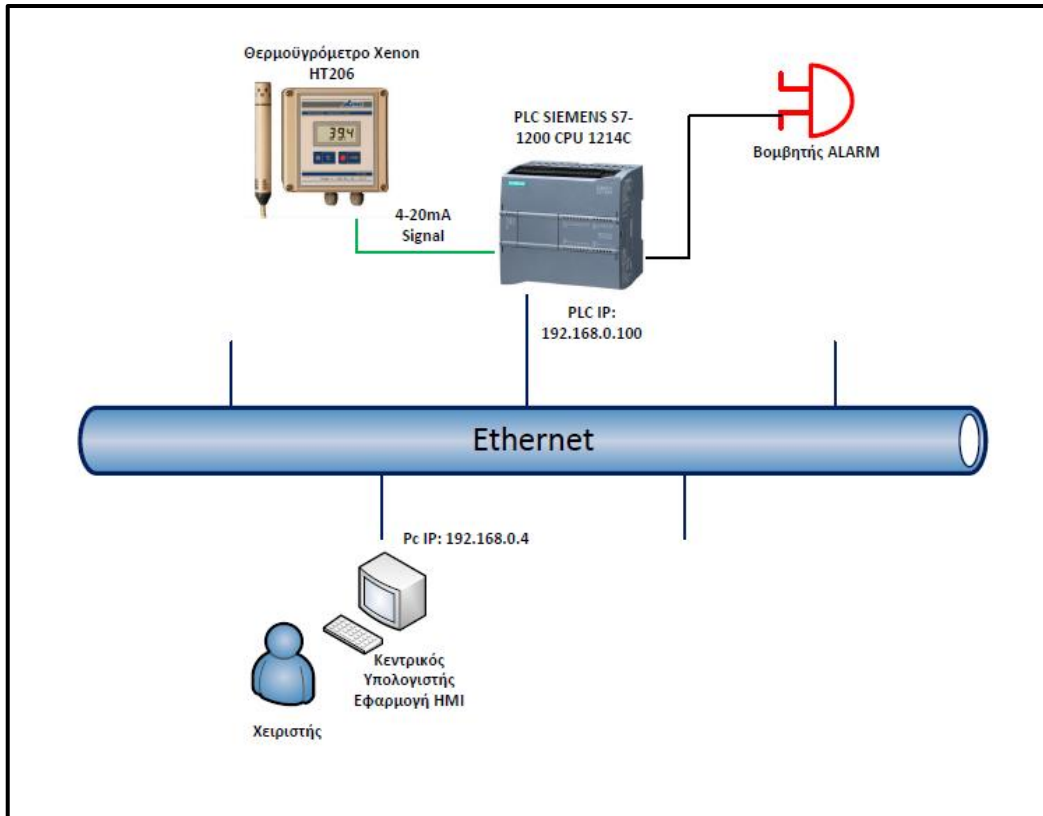
## **3.2 Η αρχιτεκτονική του συστήματος αυτοματισμού**

### **3.2.1 Γενικά**

Στα παρακάτω κεφάλαια παρουσιάζεται η αρχιτεκτονική του συστήματος αυτοματισμού καθώς και τα μέρη τα οποία το απαρτίζουν.

### **3.2.2 Τοπολογία συστήματος**

Στο σχήμα που ακολουθεί φαίνεται η τοπολογία του συστήματος.



Σχήμα 13: Η τοπολογία του συστήματος

### 3.2.2 Το βιομηχανικό ψηφιακό θερμοϋγρόμετρο

Το ψηφιακό θερμοϋγρόμετρο τύπου **HT206** της εταιρείας **ΞΕΝΩΝ** είναι ένα βιομηχανικό όργανο ακριβείας για την μέτρηση θερμοκρασίας και της σχετικής υγρασίας. Επιπλέον μέσω της αναλογικής εξόδου, δίνει την δυνατότητα καταγραφής των μετρήσεων από άλλους βιομηχανικούς εξοπλισμούς. Μία τυπική εφαρμογή του οργάνου είναι για την μέτρηση θερμοκρασίας και σχετικής υγρασίας περιβάλλοντος σε βιομηχανικούς χώρους.

Το σύστημα μέτρησης θερμοκρασίας και σχετικής υγρασίας αποτελείται από :

- Όργανο μέτρησης τύπου HT206 που λειτουργεί σαν ο μετατροπέας του συστήματος μέτρησης.
- Αισθητήρα μέτρησης σχετικής υγρασίας και θερμοκρασίας τύπου SH206.

Παρακάτω αναφέρονται κάποια κύρια χαρακτηριστικά του ψηφιακού θερμοϋγρομέτρου:

- Το σύστημα έχει δυνατότητα μέτρησης της σχετικής υγρασίας στο εύρος **0 ÷ 100% RH** ενώ στη θερμοκρασία **-20 ÷ +70 °C**.



- Ο αισθητήρας SH206 έχει ενσωματωμένο ηλεκτρονικό κύκλωμα που επιτρέπει την εγκατάσταση του σε απόσταση έως και 50m από τον μετατροπέα του οργάνου HT206.
- Το όργανο τύπου HT206 διαθέτει ψηφιακή ένδειξη υγρών κρυστάλλων(LCD).
- Το βιομηχανικό θερμοϋγρόμετρο διαθέτει δύο αναλογικές εξόδους ρεύματος 4-20mA (για τις μετρήσεις θερμοκρασίας και σχετικής υγρασίας) για την σύνδεση του με άλλους βιομηχανικούς εξοπλισμούς, όπως για παράδειγμα PLC, Data Acquisitions κ.α.

Στη παρακάτω εικόνα φαίνεται το βιομηχανικό θερμοϋγρόμετρο τύπου HT206:



**Εικόνα 3:** Θερμοϋγρόμετρο HT206 της εταιρείας ΞΕΝΩΝ

Το ψηφιακό θερμοϋγρόμετρο χρησιμοποιήθηκε στην εφαρμογή μας ώστε μέσω του αισθητήρα να καταγράφονται μετρήσεις θερμοκρασίας και σχετικής υγρασίας του χώρου. Ο λόγος που επιλέχθηκε το όργανο HT206 για την εφαρμογή μας ήταν ότι υπάρχει αρκετή προσωπική εμπειρία στο εργασιακό μου περιβάλλον με το εν λόγω εξοπλισμό καθώς το έχω χρησιμοποιήσει σε αρκετές εφαρμογές όπως επίσης και ότι ήταν διαθέσιμο για την εφαρμογή.

### 3.2.3 Ο προγραμματιζόμενος λογικός ελεγκτής (PLC)

Ο προγραμματιζόμενος λογικός ελεγκτής που χρησιμοποιήθηκε στην εφαρμογή μας , είναι το PLC της εταιρείας SIEMENS, σειράς **SIMATIC S7-1200, CPU 1214C DC/DC/DC**. Το μοντέλο του PLC είναι το 6ES7 214-1AE30-0XB0.



**Εικόνα 4:** Το PLC S7-1200 με CPU 1214C της εταιρείας Siemens

Παρακάτω αναφέρονται κάποια κύρια χαρακτηριστικά του προγραμματιζόμενου λογικού ελεγκτή:

- Τάση τροφοδοσίας: DC 20.4-28.8V DC
- Ψηφιακές εισόδους: 14
- Ψηφιακές εξόδους: 10
- Αναλογικές εισόδους: 2(0-10 V DC)
- Ανάλυση αναλογικών εισόδων(resolution): 10bit
- Μνήμη εργασίας: 50 Kbyte
- Διεπαφή (interface): PROFINET
- Πρωτόκολλο επικοινωνίας: ETHERNET(TCP/IP)
- Web Server: NAI
- Γλώσσες προγραμματισμού: LAD, FBD, SCL.
- Κύκλος προγράμματος: 150ms

Παρακάτω φαίνεται συγκριτικός πίνακας αντίστοιχων CPU της ίδιας σειράς SIMATIC S7-1200:

Feature		CPU 1211C	CPU 1212C	CPU 1214C
Physical size (mm)		90 x 100 x 75	90 x 100 x 75	110 x 100 x 75
User memory	Work	25 Kbytes	25 Kbytes	50 Kbytes
	Load	1 Mbyte	1 Mbyte	2 Mbytes
	Retentive	2 Kbytes	2 Kbytes	2 Kbytes
Local on-board I/O	Digital	6 inputs/4 outputs	8 inputs/6 outputs	14 inputs/10 outputs
	Analog	2 inputs	2 inputs	2 inputs
Process image size	Inputs (I)	1024 bytes	1024 bytes	1024 bytes
	Outputs (Q)	1024 bytes	1024 bytes	1024 bytes
Bit memory (M)		4096 bytes	4096 bytes	8192 bytes
Signal module (SM) expansion		None	2	8
Signal board (SB) or communication board (CB)		1	1	1
Communication module (CM) (left-side expansion)		3	3	3
High-speed counters	Total	3	4	6
	Single phase	3 at 100 kHz	3 at 100 kHz 1 at 30 kHz	3 at 100 kHz 3 at 30 kHz
	Quadrature phase	3 at 80 kHz	3 at 80 kHz 1 at 20 kHz	3 at 80 kHz 3 at 20 kHz
Pulse outputs <sup>1</sup>		2	2	2
Memory card		SIMATIC Memory card (optional)		
Real time clock retention time		10 days, typical / 6 day minimum at 40 degrees C		
PROFINET		1 Ethernet communications port		
Real math execution speed		18 μs/instruction		
Boolean execution speed		0.1 μs/instruction		

Πίνακας 1: Συγκριτικός πίνακας αντίστοιχων CPU της ίδιας σειράς SIMATIC S7-1200

### 3.2.4 Η ηχητική ειδοποίηση

Για την ειδοποίηση των συμβάντων της εφαρμογής μας χρησιμοποιήθηκε ένας βομβητής(buzzer) της εταιρείας Schneider Electric, σειράς XB5-KSB. Ο βομβητής ενεργοποιείτε από τον προγραμματιζόμενο λογικό ελεγκτή(PLC) κάθε φορά που υπάρχει κάποιο συμβάν.



Εικόνα 5: Η ηχητική ειδοποίηση της εφαρμογής

Παρακάτω φαίνονται τα κύρια χαρακτηριστικά του βομβητή:

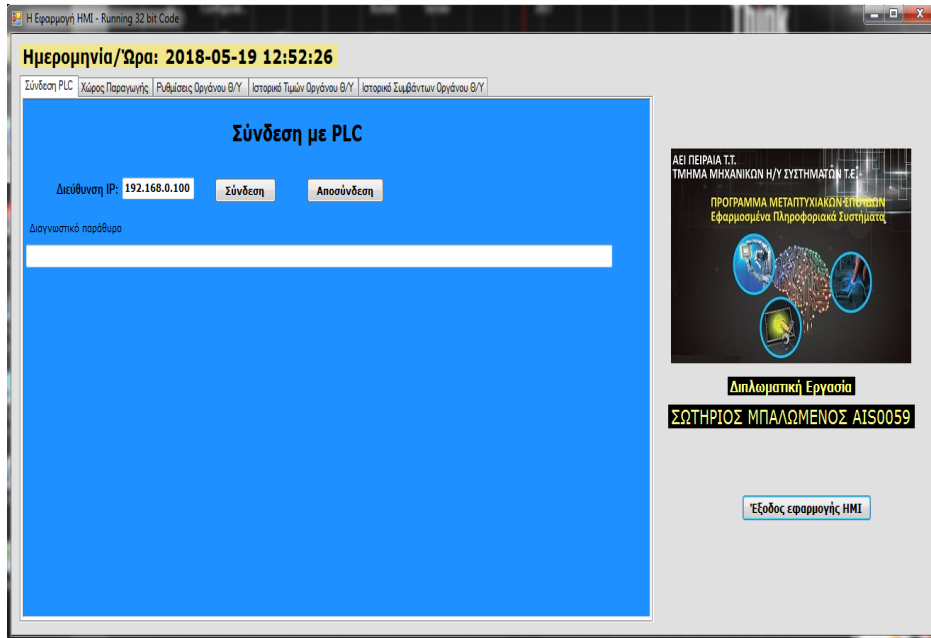
- Τάση τροφοδοσίας: 24 V DC – 50-60Hz

- Βομβητής (buzzer) 85 db, 4KHz
- Βαθμός προστασίας: IP 40

### **3.2.5 Η εφαρμογή HMI**

Η εποπτεία της εφαρμογής αυτοματισμού γίνεται μέσω του λογισμικού HMI το οποίο είναι εγκατεστημένο στον Η/Υ του χρήστη. Η εφαρμογή HMI είναι το εργαλείο εποπτικού ελέγχου και συλλογής δεδομένων (SCADA) για την εφαρμογή μας και ο χρήστης έχει τις παρακάτω δυνατότητες:

1. Σύνδεση του λογισμικού HMI με τον ελεγκτή για την εποπτεία της εφαρμογής.
2. Ενημέρωση για την κατάσταση της σύνδεσης με τον ελεγκτή μέσω διαγνωστικού παραθύρου.
3. Συνεχής παρακολούθηση των τιμών θερμοκρασίας και υγρασίας από τον αισθητήρα.
4. Συνεχής παρακολούθηση για την ύπαρξη συμβάντος- συναγερμού (alarm).
5. Ενημέρωση για το ιστορικό τιμών θερμοκρασίας και υγρασίας του αισθητήρα.
6. Δυνατότητα χειροκίνητης καταχώρησης τιμών του οργάνου στη βάση δεδομένων.
7. Ενημέρωση για το ιστορικό συμβάντων-συναγερμών.
8. Ρύθμιση ορίων συναγερμού για θερμοκρασία και υγρασία.
9. Ρύθμιση του χρόνου καθυστέρησης για την ενεργοποίηση των συναγερμών από τον ελεγκτή.
10. Εξαγωγή ιστορικού τιμών σε μορφή κειμένου (txt).
11. Εξαγωγή ιστορικού συμβάντων σε μορφή κειμένου (txt).
12. Εκτύπωση ιστορικών τιμών-συμβάντων μέσω αρχείων κειμένου.



Εικόνα 6: Η εφαρμογή HMI

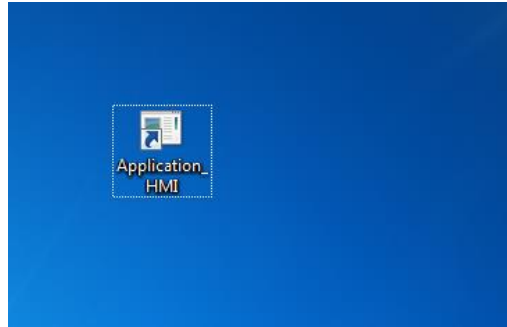
### 3.3 Η εποπτεία ελέγχου και η συλλογή δεδομένων με την Εφαρμογή HMI

#### 3.3.1 Γενικά

Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο , με την εφαρμογή HMI γίνεται ουσιαστικά η εποπτεία ελέγχου αλλά και η συλλογή δεδομένων (SCADA) για την εφαρμογή μας. Σε αυτό το κεφάλαιο παρουσιάζεται αναλυτικά η εφαρμογή HMI αλλά και οι δυνατότητες που έχει ο χρήστης μέσα από αυτή.

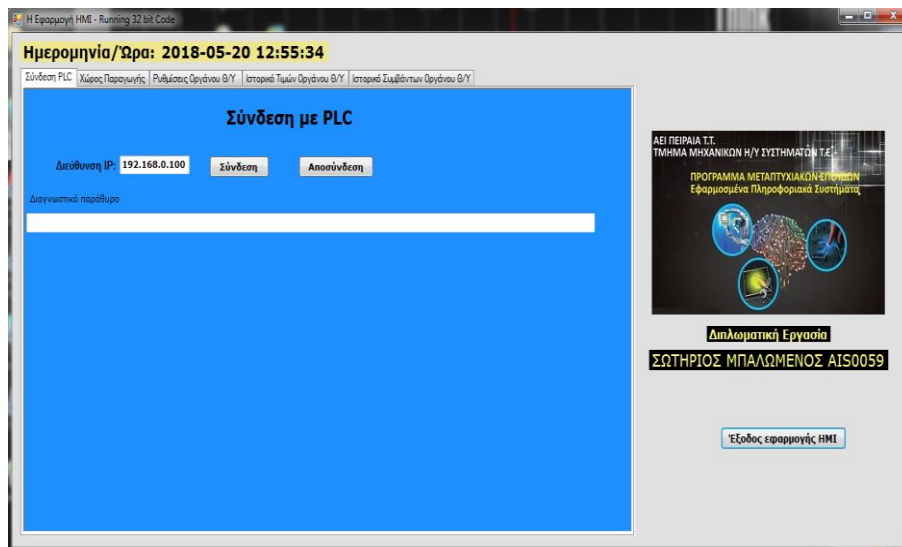
#### 3.3.2 Είσοδος στην Εφαρμογή HMI

Η εφαρμογή HMI βρίσκεται εγκατεστημένη στον Η/Υ του χρήστη. Στην επιφάνεια εργασίας υπάρχει η συντόμευση της εφαρμογής όπως φαίνεται στο εικόνα που ακολουθεί.



**Εικόνα 7:** Η Συντόμευση για την εφαρμογή HMI

Κάνοντας διπλό κλικ (αριστερό πλήκτρο mouse) στην παραπάνω συντόμευση, ανοίγει η εφαρμογή HMI όπως φαίνεται στην παρακάτω εικόνα.



**Εικόνα 8:** Η Αρχική οθόνη της εφαρμογής HMI

Στην αρχική οθόνη της εφαρμογής μπορούμε να δούμε την τρέχουσα ημερομηνία και ώρα αλλά και τις διαθέσιμες επιλογές που μας δίνονται μέσα από αυτή.

Αναλυτικά οι διαθέσιμες επιλογές είναι οι παρακάτω:

- Σύνδεση με PLC.
- Χώρος Παραγωγής.
- Ρυθμίσεις οργάνου Θερμοκρασίας – Υγρασίας.
- Ιστορικό τιμών οργάνου Θερμοκρασίας – Υγρασίας.
- Ιστορικό συμβάντων οργάνου Θερμοκρασίας – Υγρασίας.

- Έξοδος εφαρμογής HMI.

### 3.3.3 Σύνδεση με PLC

Σε αυτή την οθόνη έχουμε την δυνατότητα να συνδεθούμε με τον προγραμματιζόμενο λογικό ελεγκτή (PLC) αλλά και να παρακολουθούμε την κατάσταση της σύνδεσης.

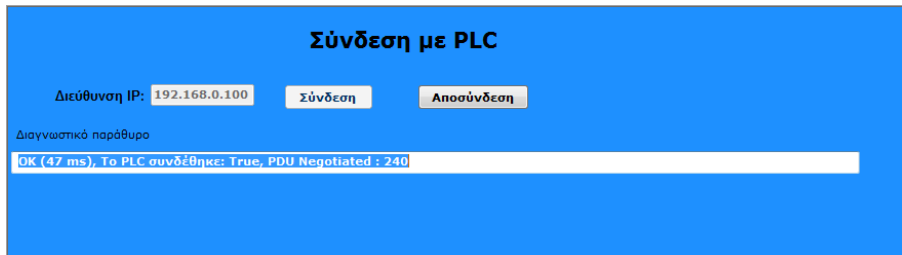
Στην εικόνα που ακολουθεί φαίνεται η οθόνη σύνδεσης με το PLC.



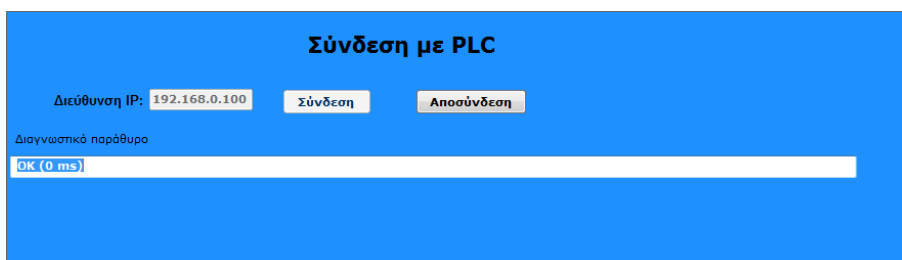
**Εικόνα 9:** Η οθόνη σύνδεσης με το PLC

Η σύνδεση της εφαρμογής HMI με το PLC πραγματοποιείται μέσω δικτύου Ethernet και πρωτοκόλλου TCP/IP. Σε αυτή την οθόνη υπάρχει το πεδίο στο οποίο δίνεται η διεύθυνση IP του PLC. Στο συγκεκριμένο PLC έχει δοθεί η διεύθυνση με **IP: 192.168.0.100**. Πατώντας το μπουτον “**Σύνδεση**” η εφαρμογή συνδέεται στο PLC και ένα αρχικό μήνυμα «**OK(47ms), Το PLC συνδέθηκε: True, PDU Negotiated:240**» φαίνεται στο διαγνωστικό παράθυρο. Αν αναλύσουμε το παραπάνω μήνυμα αυτό ερμηνεύεται ως εξής : OK ως κατάσταση σύνδεσης, 47ms ως ο χρόνος για την επίτευξη της σύνδεσης, Το PLC συνδέθηκε: True ως η επιβεβαίωση σύνδεσης με το PLC και το PDU Negotiated: 240 ως τα πακέτα δεδομένων που διαβάστηκαν από το PLC. Στη συνέχεια το μήνυμα «OK» είναι αυτό που παραμένει στο διαγνωστικό παράθυρο ώστε από εκεί να επιβεβαιώνουμε την κατάσταση της σύνδεσης.

Επιπλέον με την σύνδεση της εφαρμογής, το πεδίο **Διεύθυνση IP: 192.168.0.100** αλλά και το μπουτον “**Σύνδεση**” φαίνονται απενεργοποιημένα. Η σύνδεση της εφαρμογής με το PLC φαίνονται στις παρακάτω εικόνες.

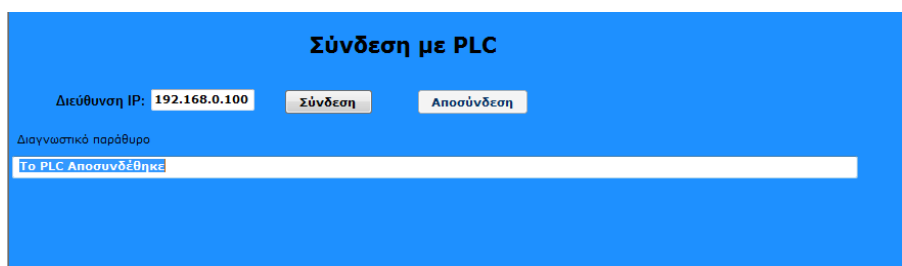


**Εικόνα 10:** Μήνυμα 1 σύνδεσης από το διαγνωστικό παράθυρο



**Εικόνα 11:** Μήνυμα 2 από το διαγνωστικό παράθυρο

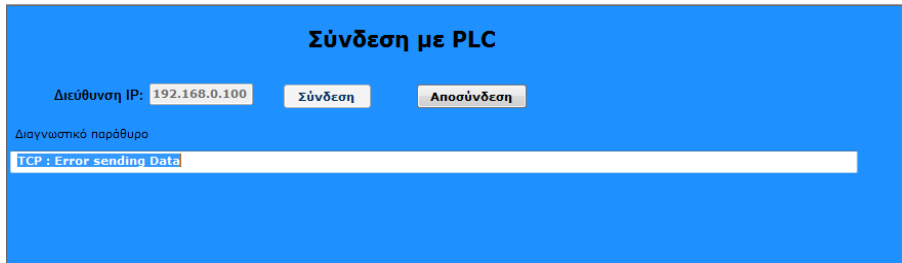
Αντίθετα εάν ο χρήστης επιθυμεί την αποσύνδεση της εφαρμογής με το PLC, θα πρέπει να πατήσει το μπουτον “**Αποσύνδεση**”. Πατώντας το συγκεκριμένο μπουτον η εφαρμογή αποσυνδέεται από το PLC και το μήνυμα «**Το PLC αποσυνδέθηκε**» φαίνεται πλέον στο διαγνωστικό παράθυρο. Επιπλέον με την αποσύνδεση της εφαρμογής, το μπουτόν “**Αποσύνδεση**” φαίνεται απενεργοποιημένο. Η αποσύνδεση της εφαρμογής με το PLC φαίνεται στην εικόνα που ακολουθεί.



**Εικόνα 12:** Το Μήνυμα αποσύνδεσης της εφαρμογής από το PLC



Σε περίπτωση που υπάρξει κάποιο πρόβλημα στην επικοινωνία μεταξύ εφαρμογής HMI και Του PLC θα εμφανιστεί και το αντίστοιχο μήνυμα στο διαγνωστικό παράθυρο. Στη παρακάτω εικόνα φαίνεται η περίπτωση διακοπής επικοινωνίας λόγω δικτύου Ethernet.



**Εικόνα 13:** Μήνυμα σφάλματος στην επικοινωνία

### 3.3.4 Χώρος παραγωγής

#### 3.3.4.1 Γενικά

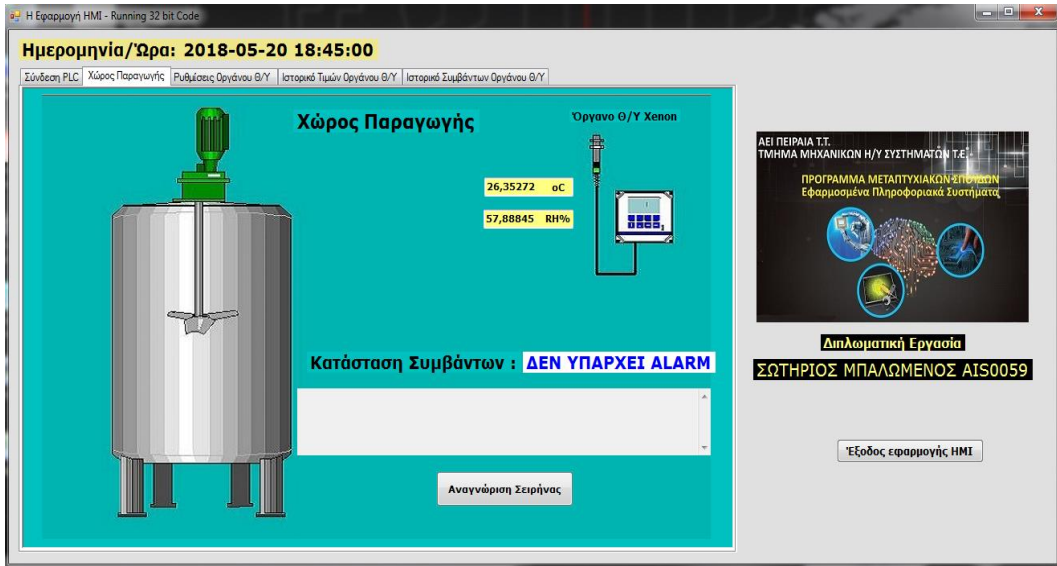
Μέσω της οθόνης αυτής, ο χειριστής έχει τις παρακάτω δυνατότητες:

- Συνεχής παρακολούθηση των τιμών θερμοκρασίας-υγρασίας.
- Συνεχής παρακολούθηση για την κατάσταση των συναγερμών.

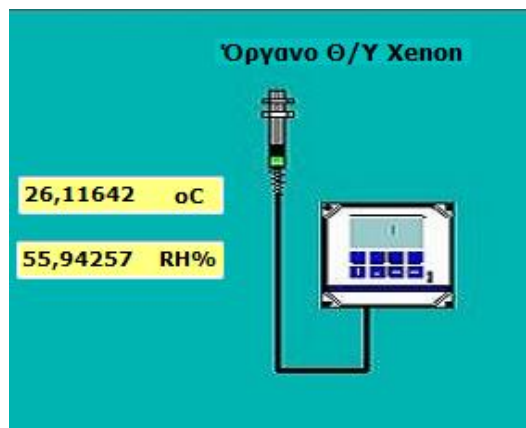
#### 3.3.4.2 Συνεχής παρακολούθηση των τιμών θερμοκρασίας και υγρασίας

Όπως αναφέρθηκε παραπάνω, ο χρήστης μέσα από την οθόνη αυτή έχει την δυνατότητα συνεχής παρακολούθησης των περιβαλλοντικών συνθηκών που υπάρχουν στον χώρο παραγωγής μίας σημαντικής διεργασίας. Στα προηγούμενα κεφάλαια εξηγήσαμε ότι οι περιβαλλοντικές συνθήκες ελέγχονται μέσω της σύνδεσης με το PLC στο οποίο είναι συνδεδεμένο το όργανο θερμοκρασίας-υγρασίας HT206.

Οι τιμές της θερμοκρασίας και υγρασίας ανανεώνονται συνεχώς καθώς υπάρχει online σύνδεση μεταξύ εφαρμογής HMI και PLC. Στις εικόνες που ακολουθούν φαίνεται η οθόνη από τον χώρο παραγωγής και οι τρέχουσες τιμές του οργάνου.



Εικόνα 14: Η οθόνη Παραγωγής



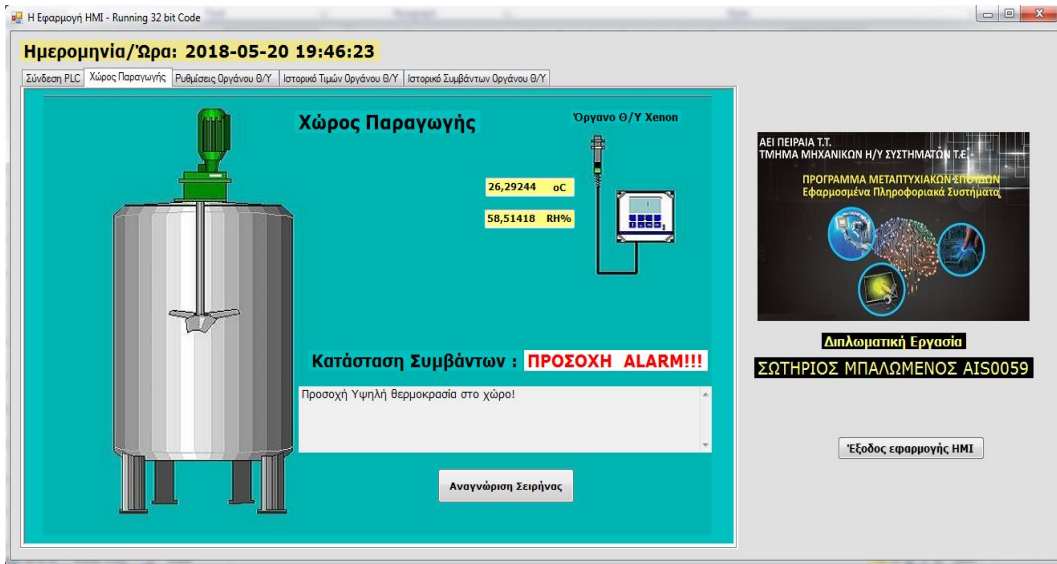
Εικόνα 15: Οι τρέχουσες τιμές οργάνου

### 3.3.4.3 Συνεχής παρακολούθηση για την κατάσταση συναγεμμών

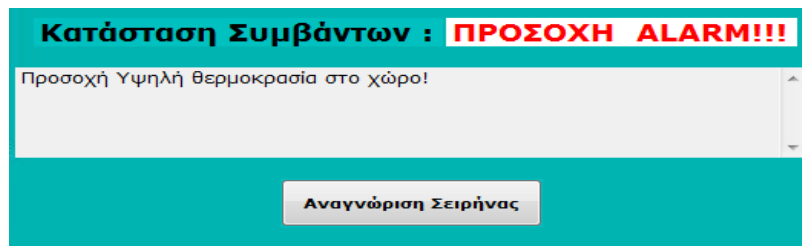
Πέρα την ενημέρωση για τις τρέχουσες τιμές , στην οθόνη αυτή υπάρχει και συνεχής ενημέρωση για την ύπαρξη κάποιου συναγεμμού-συμβάντος.

Σε περίπτωση που κάποια τιμή θερμοκρασίας ή υγρασίας είναι εκτός των καθορισμένων ορίων, στο σύστημα δημιουργείται συναγεμμός και ο χρήστης ενημερώνεται μέσω του πεδίου “Κατάσταση Συμβάντων” με το μήνυμα “Προσοχή **ALARM!!!**”. Επιπλέον κατά την διάρκεια του συναγεμμού το σύστημα ενεργοποιεί ηχητικό βομβητή. Έτσι στην περίπτωση που ο χρήστης δεν βρίσκεται στην εφαρμογή HMI θα ακούσει την ειδοποίηση και θα επιστρέψει στην εφαρμογή για περισσότερες λεπτομέρειες. Στη συνέχεια μπαίνοντας στην εφαρμογή μπορεί να ενημερωθεί αναλυτικά για την ύπαρξη συναγεμμού, βλέποντας ποια τιμή είναι

εκτός ορίων ώστε να προβεί στις απαραίτητες διορθωτικές ενέργειες. Από την οθόνη αυτή ο χρήστης θα μπορεί να αναγνωρίσει τον συναγερμό πατώντας το μπουτον “**Αναγνώριση Σειρήνας**” με το οποίο απενεργοποιείται η ηχητική ειδοποίηση του βομβητή.

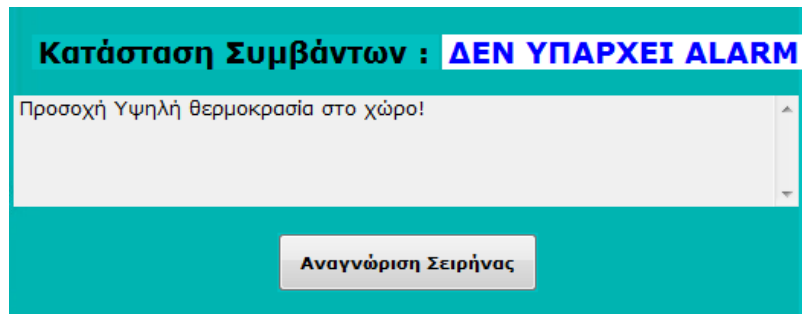


Εικόνα 16: Το σύστημα σε κατάσταση συναγερμού



Εικόνα 17: Το μήνυμα συναγερμού και το μπουτον αναγνώρισης

Στο γκρι φόντο παρουσιάζεται το τελευταίο συμβάν της εφαρμογής. Όταν λήξει ο συναγερμός στην “**Κατάσταση Συμβάντων**” επανέρχεται το μήνυμα “**ΔΕΝ ΥΠΑΡΧΕΙ ALARM**” όπως φαίνεται στο εικόνα που ακολουθεί.



**Εικόνα 18:** Το σύστημα σε κατάσταση χωρίς κάποιο συμβάν

### 3.3.5 Ρυθμίσεις οργάνου Θερμοκρασίας-Υγρασίας

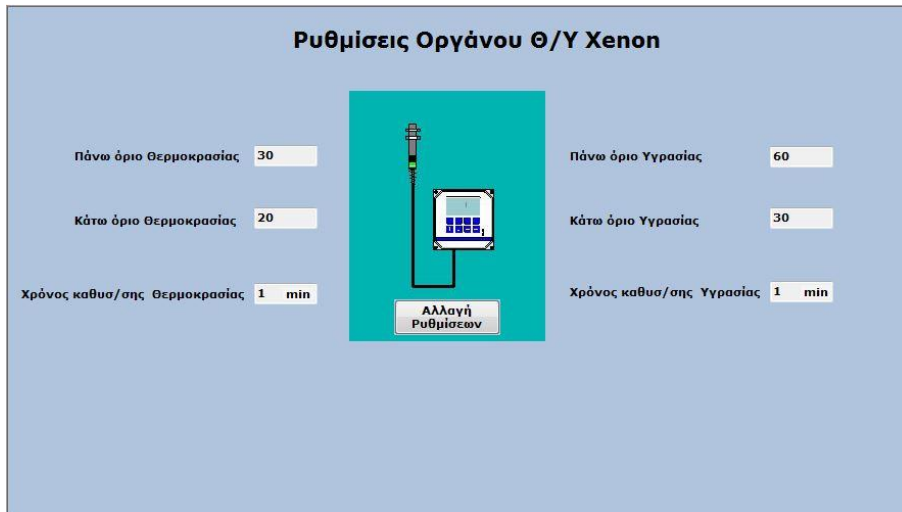
#### 3.3.5.1 Γενικά

Στην οθόνη αυτή, ο χειριστής έχει την δυνατότητα να επεξεργαστεί τις ρυθμίσεις για τις τιμές θερμοκρασίας-υγρασίας όπως τα όρια συναγερμού (άνω- όριο, χρόνος καθυστέρησης) αλλά και παράλληλα να ενημερωθεί για τις τρέχουσες ρυθμίσεις.

Συνολικά ο χρήστης έχει τις παρακάτω δυνατότητες:

- Ρύθμιση για το πάνω όριο θερμοκρασίας.
- Ρύθμιση για το κάτω όριο θερμοκρασίας.
- Ρύθμιση της χρονικής καθυστέρησης για την ενεργοποίηση συναγερμού θερμοκρασίας.
- Ρύθμιση για το πάνω όριο υγρασίας.
- Ρύθμιση για το κάτω όριο υγρασίας.
- Ρύθμιση της χρονικής καθυστέρησης για την ενεργοποίηση συναγερμού υγρασίας.

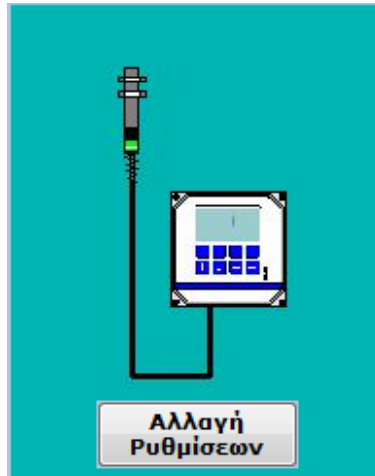
Στην παρακάτω εικόνα παρουσιάζεται η οθόνη από τις ρυθμίσεις οργάνου.



Εικόνα 19: Οθόνη ρυθμίσεων συναγερμών

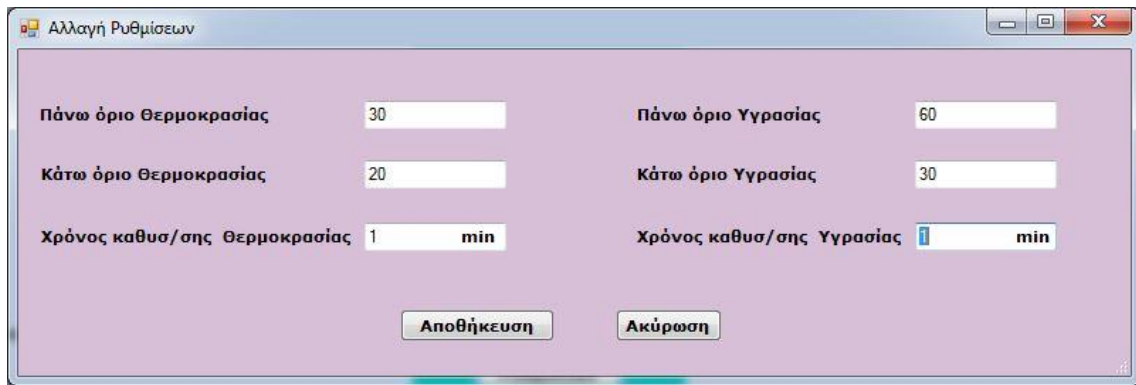
### 3.3.5.2 Επεξεργασία ρυθμίσεων οργάνου Θερμοκρασίας- Υγρασίας

Ο χειριστής μπορεί να αλλάζει τις παραπάνω ρυθμίσεις σύμφωνα με τις ανάγκες της διεργασίας πατώντας στο μπουτον “**Αλλαγή Ρυθμίσεων**” όπως φαίνεται στην εικόνα που ακολουθεί.



Εικόνα 20: Μπουτον αλλαγής Ρυθμίσεων

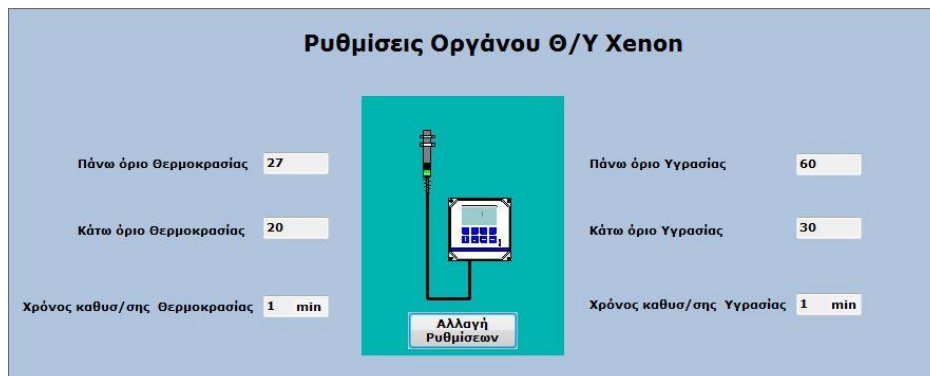
Στη συνέχεια, ένα νέο παράθυρο ‘**Αλλαγή Ρυθμίσεων**’ ανοίγει , στο οποίο ο χειριστής πηγαίνοντας στο επιθυμητό πεδίο, μπορεί να αλλάξει την ορισμένη τιμή με νέα, όπως παρουσιάζεται στο εικόνα που ακολουθεί.



Εικόνα 21: Η Οθόνη αλλαγής Ρυθμίσεων

Δίνοντας λοιπόν την νέα τιμή και αφού στη συνέχεια πατήσουμε το μπουτον 'Αποθήκευση', η νέα τιμή αποθηκεύεται. Αντίθετα εάν ο χειριστής πατήσει 'Ακύρωση', επιστρέφει στην προηγούμενη οθόνη.

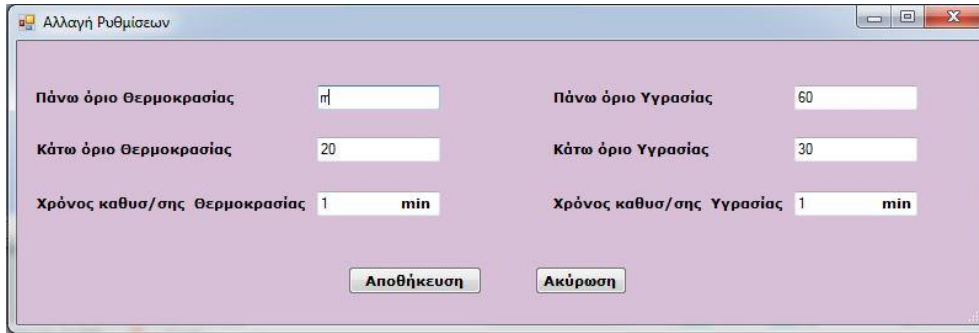
Στην εικόνα που ακολουθεί έχουμε αλλάξει το πάνω όριο θερμοκρασίας από 30 σε 27.



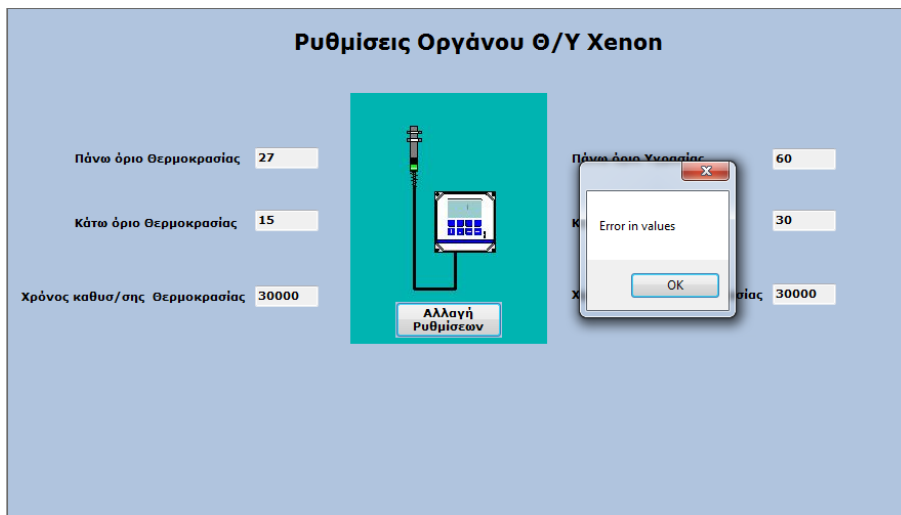
Εικόνα 22: Πάνω όριο θερμοκρασίας με τιμή 27

Σε περίπτωση που ο χρήστης δώσει λάθος τιμή σε κάποιο από τα παραπάνω πεδία, για παράδειγμα στο πάνω όριο θερμοκρασίας δώσει την τιμή «rr» και στην συνέχεια πατήσει αποθήκευση, η εφαρμογή προστατεύεται από ελέγχους οι οποίοι θα απορρίψουν την παραπάνω τιμή με μήνυμα "Error in values" και θα κρατήσει την τελευταία έγκαιρη τιμή.

Το παράδειγμα αυτό φαίνεται στις εικόνες που ακολουθούν.



Εικόνα 23: Δίνοντας την τιμή π στο πεδίο, πάνω όριο θερμοκρασίας



Εικόνα 24: Μήνυμα Error in Values

### 3.3.6 Ιστορικό τιμών οργάνου Θερμοκρασίας και Υγρασίας

#### 3.3.6.1 Γενικά

Η εφαρμογή HMI έχει την δυνατότητα καταγραφής της τρέχουσας τιμής θερμοκρασίας και υγρασίας σε μία βάση δεδομένων. Έτσι δημιουργείται ένα ιστορικό τιμών το οποίο ο χρήστης θα μπορεί να επεξεργαστεί ανάλογα με τις ανάγκες του. Η καταγραφή των τιμών έχει οριστεί αρχικά να γίνεται κάθε δέκα λεπτά αλλά ανάλογα με τις ανάγκες του χρήστη υπάρχει η δυνατότητα αλλαγής αυτής της συχνότητας μέσω του κώδικα του προγράμματος. Στην παρακάτω εικόνα παρουσιάζεται η οθόνη ιστορικού τιμών.



**Εικόνα 25:** Η Οθόνη ιστορικού τιμών

Μέσω αυτής της οθόνης, ο χειριστής έχει τις παρακάτω δυνατότητες:

13. Ενημέρωσης για το ιστορικό τιμών Θερμοκρασίας και Υγρασίας του αισθητήρα.
14. Εξαγωγής ιστορικού τιμών σε μορφή κειμένου (txt).
15. Εισαγωγής τρέχουσων τιμών ανεξάρτητα τον ορισμένο χρόνο καταγραφής στη βάση.

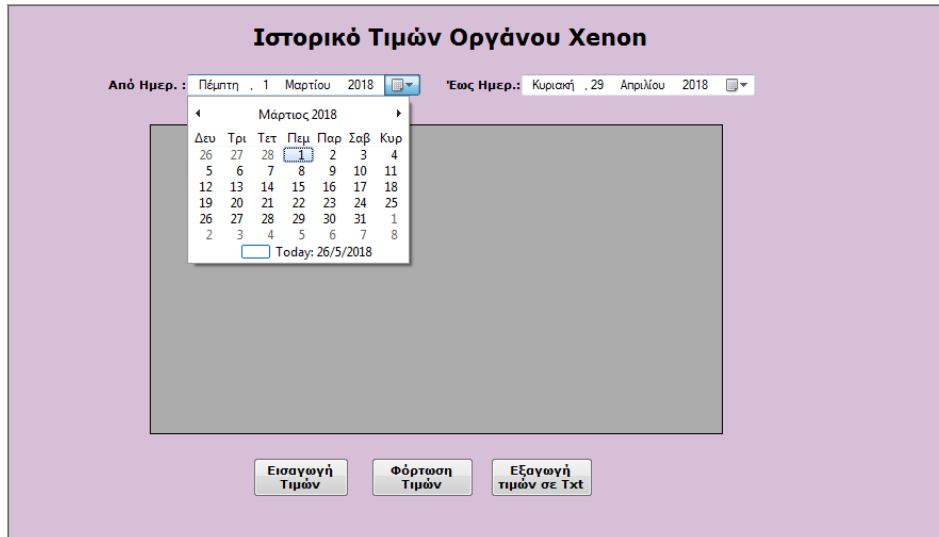
### 3.3.6.2 Φόρτωση Ιστορικού τιμών

Το ιστορικό τιμών μας εμφανίζει τις παρακάτω πληροφορίες:

- ID, μοναδικός κωδικός καταχώρησης τιμών.
- Date time, ημερομηνία καταχώρησης τιμών.
- Temperature, τιμή οργάνου θερμοκρασίας.
- Humidity, τιμή οργάνου υγρασίας.

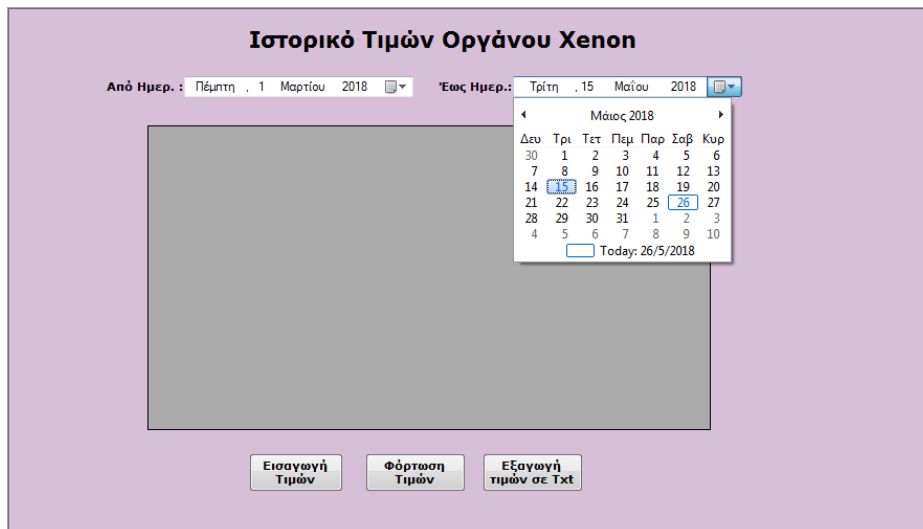
Για να εμφανίσουμε κάποιο ιστορικό τιμών, θα πρέπει να επιλέξουμε την αρχική ημερομηνία, στο πεδίο '**Από Ημερ.**'. Για παράδειγμα εδώ θα δώσουμε την ημερομηνία '**1 Μαρτίου 2018**' με την βοήθεια του ημερολογίου.





Εικόνα 26: Επιλογή της αρχικής ημερομηνίας για την εμφάνιση ιστορικού τιμών

Ακολουθώντας στο πεδίο 'Έως Ημερ.' θα δώσουμε την ημερομηνία '15 Μαΐου 2018' με την βοήθεια του ημερολογίου.



Εικόνα 27: Επιλογή της τελικής ημερομηνίας για την εμφάνιση ιστορικού τιμών

Αφού έχουμε δώσει τις επιλογές μας στα πεδία 'Από Ημερ.' Και 'Έως Ημερ.' θα πρέπει να επιλέξουμε το μπουτόν 'Φόρτωση Τιμών'. Έτσι ο πίνακας της οθόνης θα μας εμφανίσει τις διαθέσιμες τιμές από τη βάση δεδομένων όπως φαίνεται στο εικόνα που ακολουθεί.

**Ιστορικό Τιμών Οργάνου Xenon**

Από Ημερ.: Πέμπτη . 1 Μαρτίου 2018 Έως Ημερ.: Τρίτη . 15 Μαΐου 2018

id	datetime	temperature	humidity
209	4/3/2018 12:37 μμ	23,67139	53,74349
210	4/3/2018 2:50 μμ	22,3886	56,15234
211	4/3/2018 3:00 μμ	22,35725	56,48148
212	4/3/2018 3:10 μμ	22,38619	56,8721
213	4/3/2018 4:50 μμ	22,39101	57,48698
214	4/3/2018 5:00 μμ	22,24875	57,64613
215	4/3/2018 5:10 μμ	22,2608	56,92636
216	4/3/2018 5:20 μμ	22,27527	53,33478
217	4/3/2018 9:06 μμ	22,39583	55,87384
218	4/3/2018 9:16 μμ	22,31867	56,18489
219	4/3/2018 9:26 μμ	22,20293	56,41637
220	4/3/2018 9:36 μμ	22,18364	56,25362

**Εικόνα 28:** Εμφάνιση πίνακα ιστορικού τιμών

Στο δεξί μέρος του πίνακα, υπάρχει μία μπάρα κύλισης(scroll bar) μέσω της οποίας θα μπορέσουμε να δούμε όλες τις τιμές που περιέχει ο πίνακας.

**Ιστορικό Τιμών Οργάνου Xenon**

Από Ημερ.: Πέμπτη . 1 Μαρτίου 2018 Έως Ημερ.: Τρίτη . 15 Μαΐου 2018

id	datetime	temperature	humidity
398	13/5/2018 1:11 μμ	24,59973	55,31322
399	13/5/2018 1:21 μμ	24,64072	55,17578
400	13/5/2018 1:31 μμ	24,67207	55,06366
401	13/5/2018 1:41 μμ	24,72994	55,15046
402	13/5/2018 1:51 μμ	24,71065	55,41088
403	13/5/2018 3:23 μμ	25,03858	55,25535
404	13/5/2018 3:27 μμ	25,03858	54,84665
405	13/5/2018 3:37 μμ	25,0434	54,72005
406	13/5/2018 3:47 μμ	25,05305	54,77069
407	13/5/2018 3:57 μμ	25,03858	55,08174
408	13/5/2018 4:07 μμ	25,06269	55,55194
409	13/5/2018 5:19 μμ	0	0

**Εικόνα 29:** Χρησιμοποιώντας το Scroll bar στον πίνακα ιστορικού

Όπως φαίνεται από την παραπάνω εικόνα, ο πίνακας τιμών εμφάνισε **200 τιμές** (αρχική id=209, τελική id=409).

### 3.3.6.3 Χειροκίνητη εισαγωγή τιμών

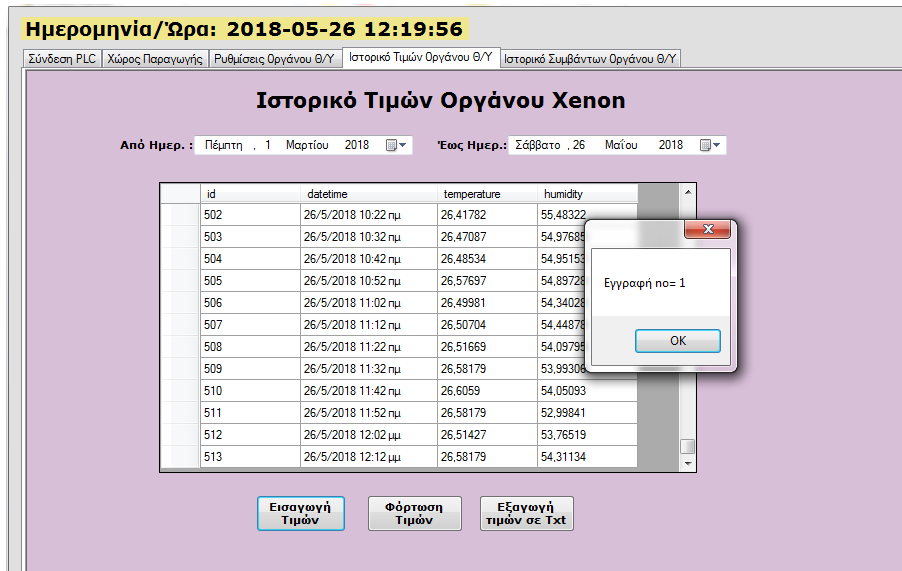
Επόμενη δυνατότητα που έχουμε μέσω αυτής της οθόνης είναι η χειροκίνητη εισαγωγή τιμών κατά περίπτωση. Έτσι ουσιαστικά ο χειριστής μπορεί να καταχωρήσει μία τρέχουσα τιμή στη βάση δεδομένων. Για να γίνει η εν λόγω καταχώρηση αρκεί μόνο να πατήσει το μπουτον **‘εισαγωγή τιμών’**.

Στο εικόνα που ακολουθεί , βλέπουμε ποιά είναι η τελευταία τιμή που έχει καταγραφεί στη βάση δεδομένων. Ουσιαστικά, αυτό που κάναμε εδώ, είναι να επιλέξουμε στο πεδίο **‘Εως Ημερ.’** την τρέχουσα ημερομηνία, δηλαδή **‘26 Μαΐου 2018’** και να πατήσουμε ξανά το μπουτόν **‘Φόρτωση Τιμών’**

id	datetime	temperature	humidity
502	26/5/2018 10:22 ημ	26,41782	55,48322
503	26/5/2018 10:32 ημ	26,47087	54,97685
504	26/5/2018 10:42 ημ	26,48534	54,95153
505	26/5/2018 10:52 ημ	26,57697	54,89728
506	26/5/2018 11:02 ημ	26,49981	54,34028
507	26/5/2018 11:12 ημ	26,50704	54,44878
508	26/5/2018 11:22 ημ	26,51669	54,09795
509	26/5/2018 11:32 ημ	26,58179	53,99306
510	26/5/2018 11:42 ημ	26,6059	54,05093
511	26/5/2018 11:52 ημ	26,58179	52,99841
512	26/5/2018 12:02 μμ	26,51427	53,76519
513	26/5/2018 12:12 μμ	26,58179	54,31134

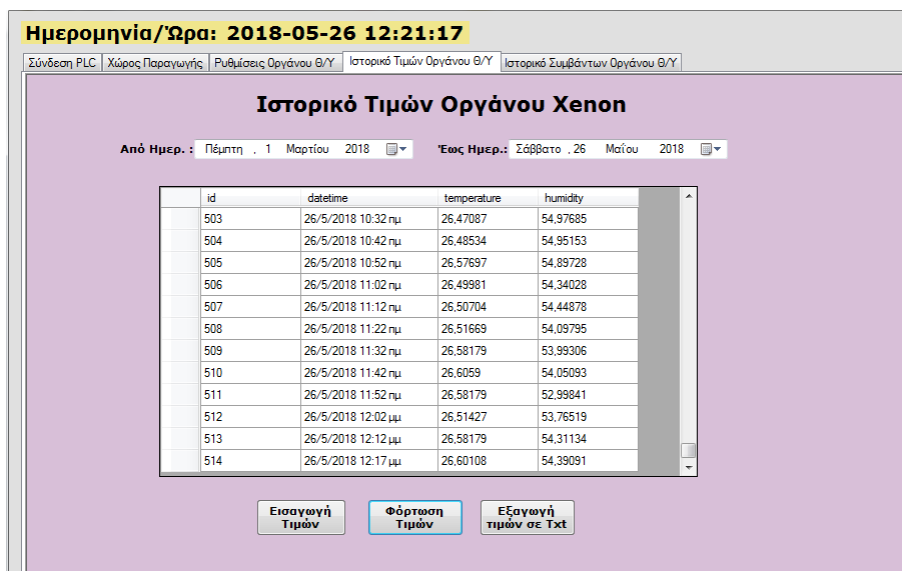
**Εικόνα 30:** Τελευταία καταχωρημένη τιμή στη βάση δεδομένων

Επόμενο βήμα είναι να καταχωρήσουμε χειροκίνητα μία τιμή πατώντας απλά το μπουτον **‘Εισαγωγή Τιμών’**. Ακολουθεί ένα μήνυμα της εφαρμογής με το οποίο μας ενημερώνει ότι πραγματοποιήθηκε μία εγγραφή (εγγραφή no=1) στη βάση δεδομένων όπως φαίνεται στην εικόνα που ακολουθεί.



Εικόνα 31: Επιτυχημένη καταχώρηση στη βάση δεδομένων

Αφού πατήσουμε 'OK', επιλέγουμε ξανά το μπουτόν 'Φόρτωση Τιμών' και βλέπουμε ποια είναι η τελευταία καταγραφή του ιστορικού μας όπως παρουσιάζεται στην εικόνα που ακολουθεί.



Εικόνα 32: Απεικόνιση ιστορικού μετά την χειροκίνητη καταχώρηση

Με την παραπάνω εικόνα επιβεβαιώνεται η νέα εγγραφή τιμών για 26/05/2018 12:17μ.μ.

### 3.3.6.4 Εξαγωγή τιμών σε αρχείο κειμένου (txt)

Πέρα από την απεικόνιση του ιστορικού τιμών μέσω του πίνακα, πολλές φορές χρειαζόμαστε μία πληροφορία σε επεξεργάσιμη μορφή. Αυτή την ανάγκη έρχεται να καλύψει η δυνατότητα εξαγωγής τιμών σε αρχείο κειμένου. Για χρησιμοποιήσουμε την δυνατότητα αυτή, θα πρέπει πρώτα να έχουμε φορτώσει το ιστορικό που επιθυμούμε όπως φαίνεται στην παρακάτω εικόνα.

Ημερομηνία/Ωρα: 2018-05-26 14:07:45

Σύνδεση PLC | Χώρος Παραγωγής | Ρυθμίσεις Οργάνου Θ/Υ | Ιστορικό Τιμών Οργάνου Θ/Υ | Ιστορικό Συμβάντων Οργάνου Θ/Υ

### Ιστορικό Τιμών Οργάνου Xenon

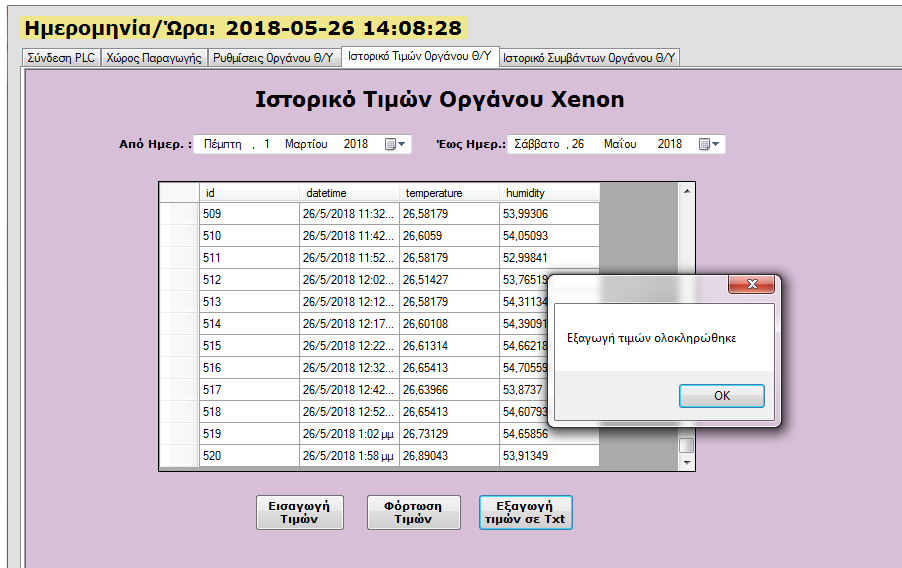
Από Ημερ.: Πέμπτη, 1 Μαρτίου 2018 Έως Ημερ.: Σάββατο, 26 Μαΐου 2018

id	datetime	temperature	humidity
509	26/5/2018 11:32...	26,58179	53,99306
510	26/5/2018 11:42...	26,6059	54,05093
511	26/5/2018 11:52...	26,58179	52,99841
512	26/5/2018 12:02...	26,51427	53,76519
513	26/5/2018 12:12...	26,58179	54,31134
514	26/5/2018 12:17...	26,60108	54,39091
515	26/5/2018 12:22...	26,61314	54,66218
516	26/5/2018 12:32...	26,65413	54,70559
517	26/5/2018 12:42...	26,63966	53,8737
518	26/5/2018 12:52...	26,65413	54,60793
519	26/5/2018 1:02 μμ	26,73129	54,65856
520	26/5/2018 1:58 μμ	26,89043	53,91349

Εισαγωγή Τιμών | Φόρτωση Τιμών | Εξαγωγή τιμών σε Txt

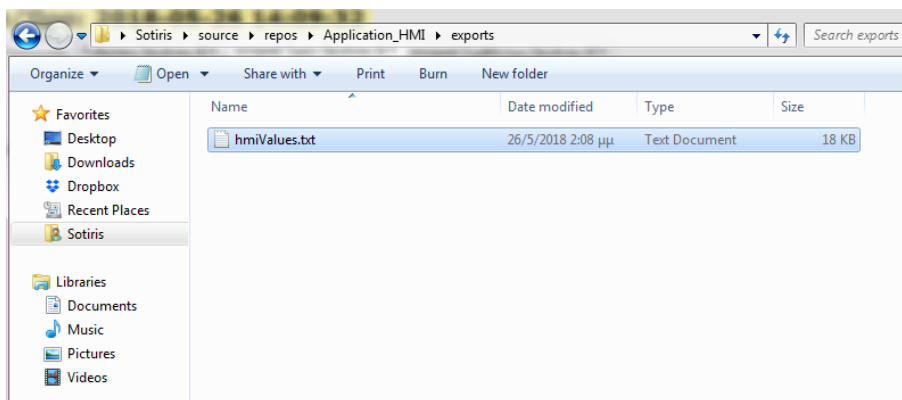
Εικόνα 33: Φόρτωση ιστορικού τιμών

Στη συνέχεια επιλέγουμε το μπουτον **‘Εξαγωγή τιμών σε Txt’** και ένα μήνυμα επιβεβαίωσης της τελευταίας ενέργειας **‘Εξαγωγή τιμών ολοκληρώθηκε’** εμφανίζεται στην οθόνη όπως μπορούμε να δούμε στην εικόνα που ακολουθεί.



Εικόνα 34: Εξαγωγή τιμών σε txt

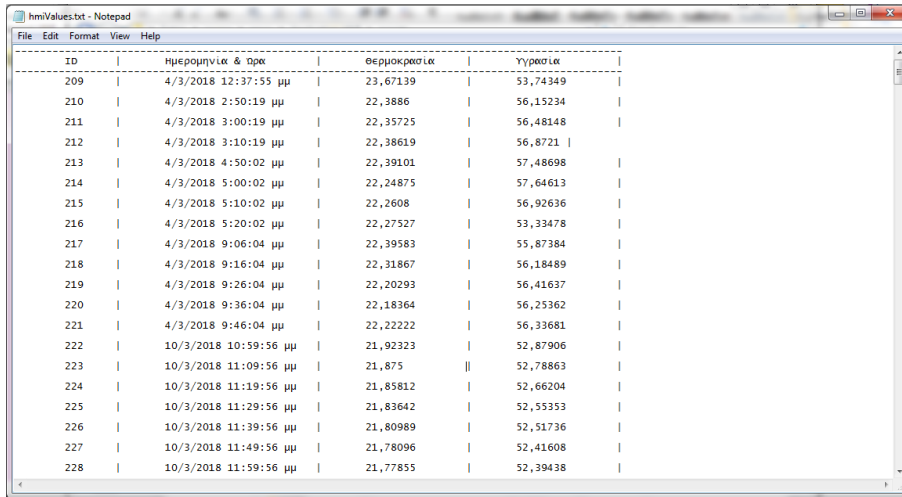
Αφού πατήσουμε 'OK', μπορούμε να πάμε στην ορισμένη διαδρομή για να βρούμε το αρχείο που μας ενδιαφέρει. Όπως βλέπουμε στην εικόνα που ακολουθεί ένα αρχείο με όνομα **hmiValues.txt** έχει δημιουργηθεί.



Εικόνα 35: Το αρχείο hmiValues

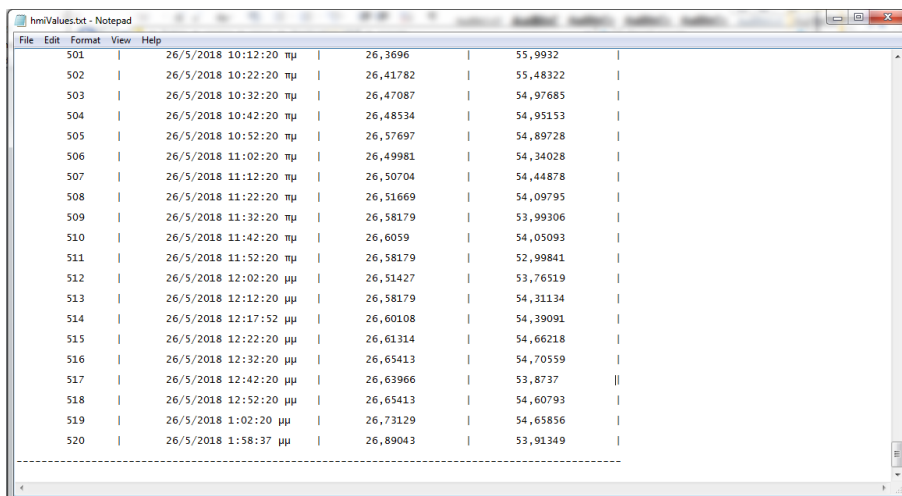
Ανοίγοντας το παραπάνω αρχείο έχουμε σε επεξεργάσιμη μορφή την αναφορά ιστορικού από την εφαρμογή μας, όπως παρουσιάζετε στις παρακάτω εικόνες

## Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)



ID	Ημερομηνία & Ώρα	Θερμοκρασία	Υγρασία
209	4/3/2018 12:37:55 μμ	23,67139	53,74349
210	4/3/2018 2:50:19 μμ	22,3886	56,15234
211	4/3/2018 3:00:19 μμ	22,35725	56,48148
212	4/3/2018 3:10:19 μμ	22,38619	56,8721
213	4/3/2018 4:50:02 μμ	22,39101	57,48698
214	4/3/2018 5:00:02 μμ	22,24875	57,64613
215	4/3/2018 5:10:02 μμ	22,2608	56,92636
216	4/3/2018 5:20:02 μμ	22,27527	53,33478
217	4/3/2018 9:06:04 μμ	22,39583	55,87384
218	4/3/2018 9:16:04 μμ	22,31867	56,18489
219	4/3/2018 9:26:04 μμ	22,20293	56,41637
220	4/3/2018 9:36:04 μμ	22,18364	56,25362
221	4/3/2018 9:46:04 μμ	22,22222	56,33681
222	10/3/2018 10:59:56 μμ	21,92323	52,87906
223	10/3/2018 11:09:56 μμ	21,875	52,78863
224	10/3/2018 11:19:56 μμ	21,85812	52,66204
225	10/3/2018 11:29:56 μμ	21,83642	52,55353
226	10/3/2018 11:39:56 μμ	21,80989	52,51736
227	10/3/2018 11:49:56 μμ	21,78096	52,41608
228	10/3/2018 11:59:56 μμ	21,77855	52,39438

**Εικόνα 36:** Το ιστορικό σε αρχείο κειμένου



ID	Ημερομηνία & Ώρα	Θερμοκρασία	Υγρασία
501	26/5/2018 10:12:20 πμ	26,3696	55,9932
502	26/5/2018 10:22:20 πμ	26,41782	55,48322
503	26/5/2018 10:32:20 πμ	26,47087	54,97685
504	26/5/2018 10:42:20 πμ	26,48534	54,95153
505	26/5/2018 10:52:20 πμ	26,57697	54,89728
506	26/5/2018 11:02:20 πμ	26,49981	54,34028
507	26/5/2018 11:12:20 πμ	26,50704	54,44878
508	26/5/2018 11:22:20 πμ	26,51669	54,09795
509	26/5/2018 11:32:20 πμ	26,58179	53,99306
510	26/5/2018 11:42:20 πμ	26,6059	54,05093
511	26/5/2018 11:52:20 πμ	26,58179	52,99841
512	26/5/2018 12:02:20 μμ	26,51427	53,76519
513	26/5/2018 12:12:20 μμ	26,58179	54,31134
514	26/5/2018 12:17:52 μμ	26,60108	54,39091
515	26/5/2018 12:22:20 μμ	26,61314	54,66218
516	26/5/2018 12:32:20 μμ	26,65413	54,70559
517	26/5/2018 12:42:20 μμ	26,63966	53,8737
518	26/5/2018 12:52:20 μμ	26,65413	54,60793
519	26/5/2018 1:02:20 μμ	26,73129	54,65856
520	26/5/2018 1:58:37 μμ	26,89043	53,91349

**Εικόνα 37:** Το ιστορικό σε αρχείο κειμένου

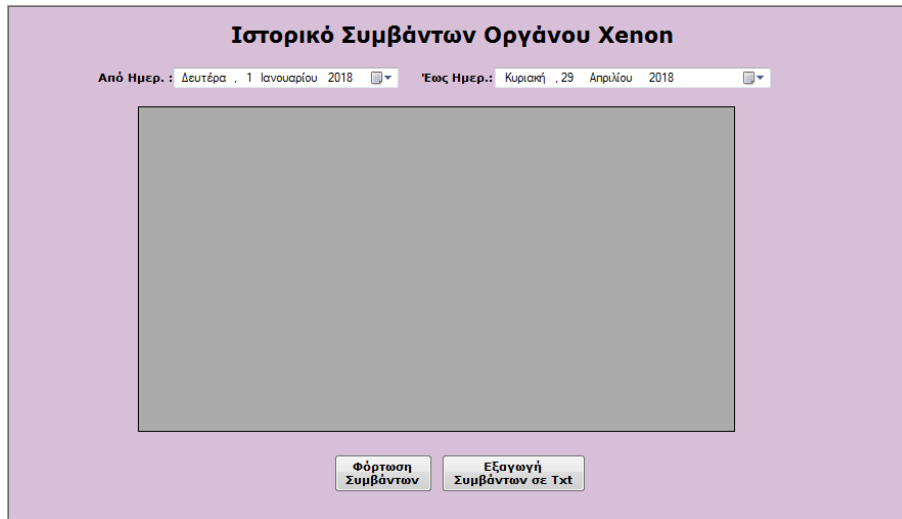
Μέσα από το αρχείο κειμένου δίνεται η δυνατότητα περαιτέρω επεξεργασίας του αλλά και εκτύπωσής του.

### 3.3.7 Ιστορικό συμβάντων οργάνου Θερμοκρασίας και Υγρασίας

#### 3.3.7.1 Γενικά

Η εφαρμογή HMI έχει την δυνατότητα καταγραφής ιστορικού για τους συναγερμούς οι οποίοι έχουν δημιουργηθεί μέσα στην εφαρμογή. Έτσι δημιουργείται ένα ιστορικό συμβάντων το οποίο ο χειριστής θα μπορεί να επεξεργαστεί ανάλογα με τις ανάγκες του. Η καταγραφή των συμβάντων γίνεται

άμεσα όταν δημιουργηθεί κάποιος συναγερμός. Στη παρακάτω εικόνα παρουσιάζεται η οθόνη ιστορικού συμβάντων.



**Εικόνα 38:** Οθόνη ιστορικού συμβάντων

Μέσω αυτής της οθόνης, ο χρήστης έχει τις παρακάτω δυνατότητες:

- 16. Ενημέρωσης για το ιστορικό συμβάντων-συναγερμών.
- 17. Εξαγωγής ιστορικού συμβάντων σε μορφή κειμένου(txt).

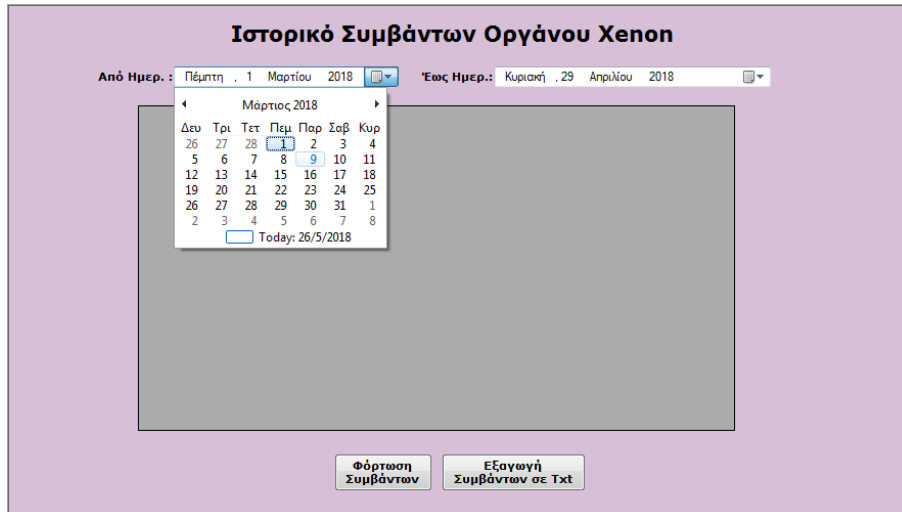
### 3.3.7.2 Φόρτωση ιστορικού συμβάντων

Το ιστορικό συμβάντων μας εμφανίζει τις παρακάτω πληροφορίες:

- ID, μοναδικός κωδικός καταχώρησης συμβάντος.
- Date time, ημερομηνία καταχώρησης συμβάντος.
- Alarm, περιγραφή συμβάντος.
- Alarm value, καταγεγραμμένη τιμή οργάνου την στιγμή του συμβάντος.

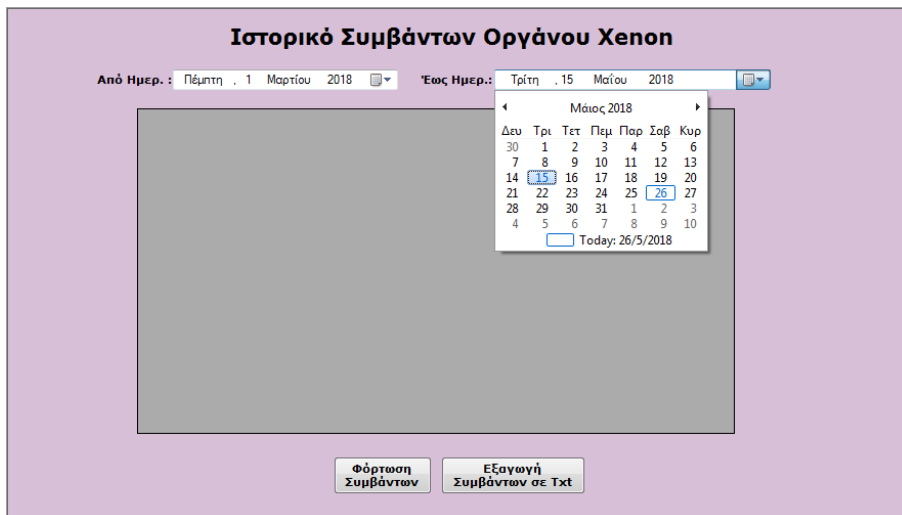
Για να εμφανίσουμε κάποιο ιστορικό συμβάντων, θα πρέπει να επιλέξουμε την αρχική ημερομηνία, στο πεδίο **‘Από Ημερ.’**. Για παράδειγμα εδώ θα δώσουμε την ημερομηνία **‘1 Μαρτίου 2018’** με την βοήθεια του ημερολογίου.





Εικόνα 39: Επιλογή αρχικής ημερομηνίας

Ακολούθως στο πεδίο **‘Έως Ημερ.’** θα δώσουμε την ημερομηνία **‘15 Μαΐου 2018’** με την βοήθεια του ημερολογίου.



Εικόνα 40: Επιλογής τελικής ημερομηνίας

Αφού έχουμε δώσει τις επιλογές μας στα πεδία **‘Από Ημερ.’** Και **‘Έως Ημερ.’** θα πρέπει να επιλέξουμε το μπουτόν **‘Φόρτωση Συμβάντων’**. Έτσι ο πίνακας της οθόνης θα μας εμφανίσει τα καταγεγραμμένα συμβάντα ανάμεσα στις δύο ημερομηνίες όπως φαίνεται στην εικόνα που ακολουθεί.

## Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)

**Ιστορικό Συμβάντων Οργάνου Ξενοπ**

Από Ημερ. : Πέμπτη . 1 Μαρτίου 2018 Έως Ημερ.: Τρίτη . 15 Μαΐου 2018

id	datetime	alam	alamvalue
823	29/3/2018 7:07 μμ	Προσοχή Υψηλή θερμοκρασία στο χώρο!!	23.6738
824	29/3/2018 7:23 μμ	Προσοχή Υψηλή θερμοκρασία στο χώρο!!	23.76543
825	29/3/2018 7:32 μμ	Προσοχή Υψηλή θερμοκρασία στο χώρο!!	23.84259
826	29/3/2018 7:35 μμ	Προσοχή Υψηλή θερμοκρασία στο χώρο!!	23.84259
827	29/3/2018 7:37 μμ	Προσοχή Υψηλή θερμοκρασία στο χώρο!!	23.85224
828	29/3/2018 7:42 μμ	Προσοχή Υψηλή θερμοκρασία στο χώρο!!	23.91975
829	29/3/2018 7:44 μμ	Προσοχή Υψηλή θερμοκρασία στο χώρο!!	23.91975
830	22/4/2018 6:36 μμ	Προσοχή Υψηλή υγρασία στο χώρο!!	49.79022
831	22/4/2018 6:37 μμ	Προσοχή Υψηλή θερμοκρασία στο χώρο!!	23.20361
832	22/4/2018 6:38 μμ	Προσοχή Χαμηλή θερμοκρασία στο χώρο!!	23.23013
833	28/4/2018 3:50 μμ	Προσοχή Υψηλή θερμοκρασία στο χώρο!!	25.2146
834	29/4/2018 11:28...	Προσοχή Υψηλή θερμοκρασία στο χώρο!!	25.37374
835	29/4/2018 1:22 μμ	Προσοχή Υψηλή θερμοκρασία στο χώρο!!	25.87047

Φόρτωση Συμβάντων Εξαγωγή Συμβάντων σε Txt

Εικόνα 41: Πίνακας ιστορικού συμβάντων από 1/03/2018 έως 15/05/2018

Στο δεξί μέρος του πίνακα, υπάρχει μία μπάρα κύλισης(scroll bar) μέσω της οποίας θα μπορέσουμε να δούμε όλα τα καταγεγραμμένα συμβάντα που περιέχει ο πίνακας.

**Ιστορικό Συμβάντων Οργάνου Ξενοπ**

Από Ημερ. : Πέμπτη . 1 Μαρτίου 2018 Έως Ημερ.: Τρίτη . 15 Μαΐου 2018

id	datetime	alam	alamvalue
865	6/5/2018 10:38 πμ	Προσοχή Υψηλή θερμοκρασία στο χώρο!	25.65104
866	6/5/2018 10:48 πμ	Προσοχή Υψηλή θερμοκρασία στο χώρο!	25.6414
867	6/5/2018 5:30 μμ	Προσοχή Υψηλή υγρασία στο χώρο!	60.06221
868	6/5/2018 5:35 μμ	Προσοχή Υψηλή υγρασία στο χώρο!	60.09838
869	6/5/2018 5:36 μμ	Προσοχή Υψηλή υγρασία στο χώρο!	60.102
870	6/5/2018 5:37 μμ	Προσοχή Υψηλή υγρασία στο χώρο!	60.1237
871	6/5/2018 6:37 μμ	Προσοχή Υψηλή υγρασία στο χώρο!	60.05498
872	6/5/2018 8:32 μμ	Προσοχή Υψηλή υγρασία στο χώρο!	60.1237
873	6/5/2018 9:13 μμ	Προσοχή Υψηλή υγρασία στο χώρο!	60.15263
874	6/5/2018 9:22 μμ	Προσοχή Υψηλή υγρασία στο χώρο!	60.12731
875	6/5/2018 9:54 μμ	Προσοχή Υψηλή υγρασία στο χώρο!	60.01881
876	6/5/2018 9:58 μμ	Προσοχή Υψηλή υγρασία στο χώρο!	60.04051

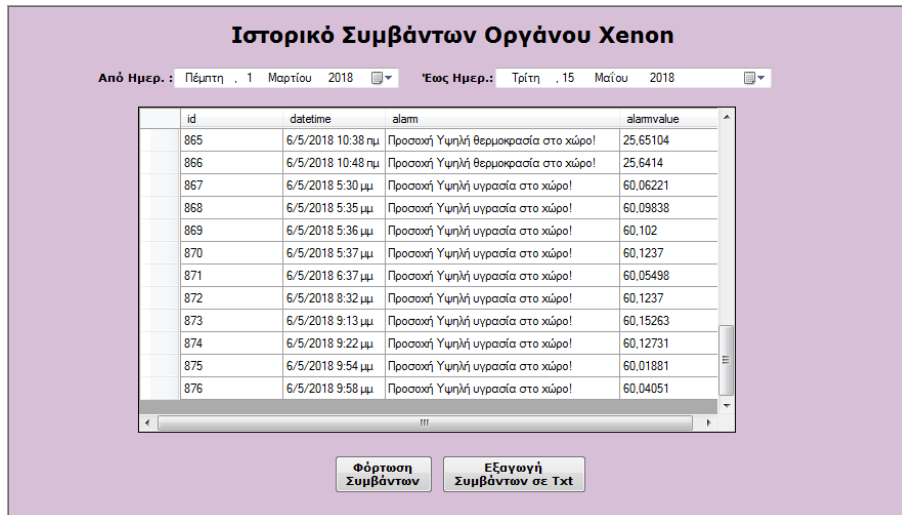
Φόρτωση Συμβάντων Εξαγωγή Συμβάντων σε Txt

Εικόνα 42: Απεικόνιση των τελευταίων συμβάντων στον πίνακα

Όπως φαίνεται από την παραπάνω εικόνα, ο πίνακας συμβάντων εμφάνισε τους συναγερμούς με **id=823** έως με **id= 876**.

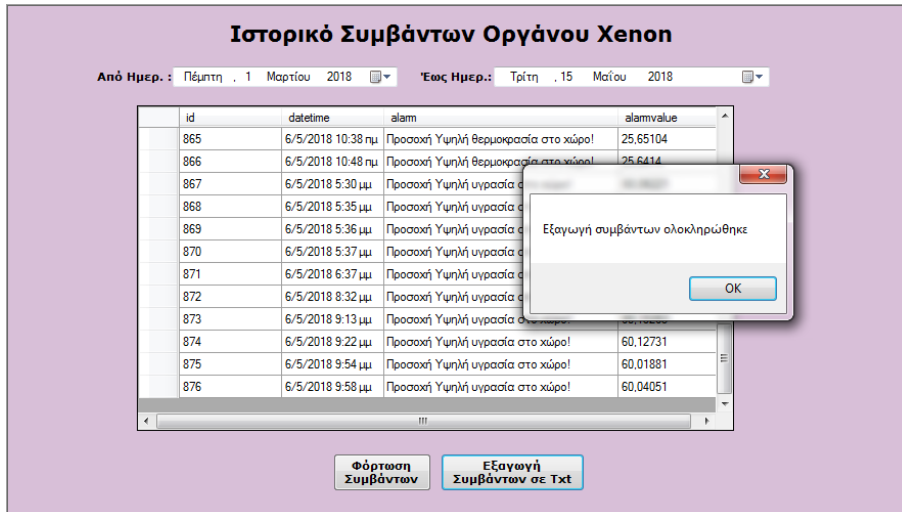
### 3.3.7.3 Εξαγωγή συμβάντων σε αρχείο κειμένου (txt)

Πέρα από την απεικόνιση των ιστορικών συμβάντων μέσω του πίνακα, πολλές φορές χρειαζόμαστε μία πληροφορία σε επεξεργάσιμη μορφή. Αυτή την ανάγκη έρχεται να καλύψει η δυνατότητα εξαγωγής συμβάντων σε αρχείο κειμένου. Για να χρησιμοποιήσουμε την δυνατότητα αυτή, θα πρέπει πρώτα να έχουμε φορτώσει το ιστορικό που επιθυμούμε όπως φαίνεται στην παρακάτω εικόνα.



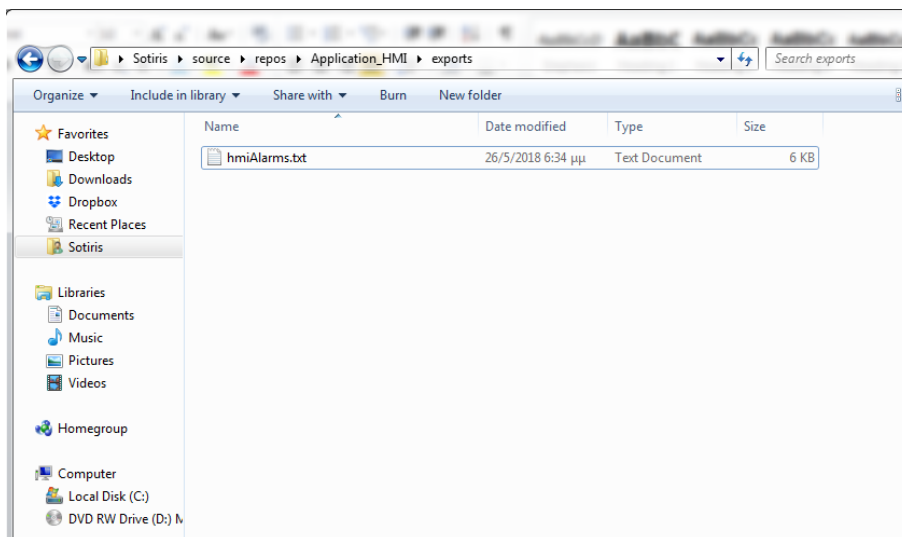
Εικόνα 43: Φόρτωση ιστορικού συμβάντων

Στη συνέχεια επιλέγουμε το μπουτον **‘Εξαγωγή Συμβάντων σε Txt’** και ένα μήνυμα επιβεβαίωσης της τελευταίας ενέργειας **‘Εξαγωγή συμβάντων ολοκληρώθηκε’** εμφανίζεται στην οθόνη όπως μπορούμε να δούμε στην εικόνα που ακολουθεί.



Εικόνα 44: Εξαγωγή συμβάντων σε αρχείο κειμένου txt

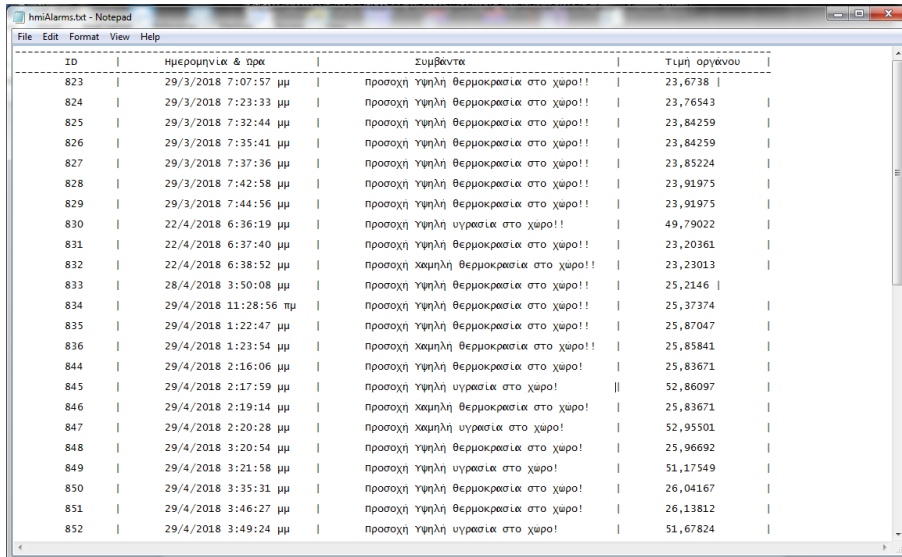
Αφού πατήσουμε 'OK', μπορούμε να πάμε στην ορισμένη διαδρομή για βρούμε το αρχείο που μας ενδιαφέρει. Όπως βλέπουμε στην εικόνα που ακολουθεί ένα αρχείο με όνομα **hmiAlarms.txt** έχει δημιουργηθεί.



Εικόνα 45: Το αρχείο hmiAlarms

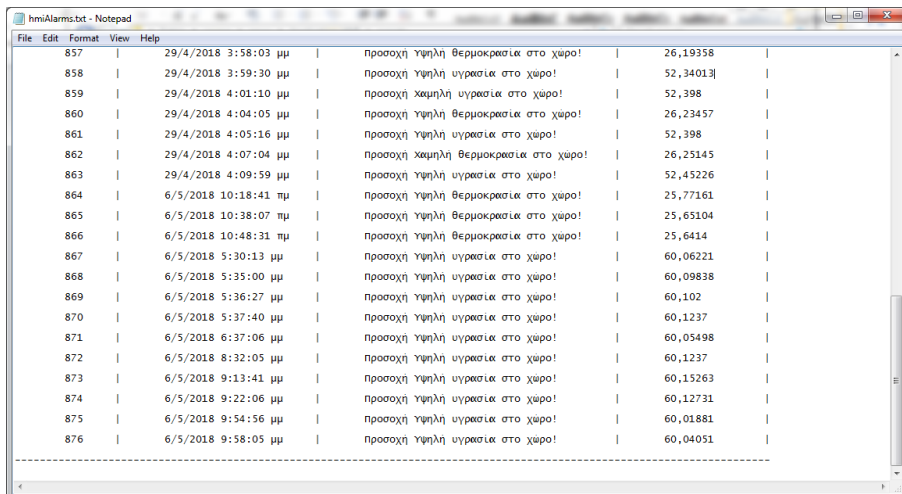
Ανοίγοντας το παραπάνω αρχείο έχουμε σε επεξεργάσιμη μορφή την αναφορά ιστορικού από την εφαρμογή μας, όπως παρουσιάζετε στις παρακάτω εικόνες.

## Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)



ID	Ημερομηνία & Ώρα	Συμβάντα	Τιμή οργάνου
823	29/3/2018 7:07:57 μμ	προσοχή υψηλή θερμοκρασία στο χώρο!	23,6738
824	29/3/2018 7:23:33 μμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	23,76543
825	29/3/2018 7:32:44 μμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	23,84259
826	29/3/2018 7:35:41 μμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	23,84259
827	29/3/2018 7:37:36 μμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	23,85224
828	29/3/2018 7:42:58 μμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	23,91975
829	29/3/2018 7:44:56 μμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	23,91975
830	22/4/2018 6:36:19 μμ	προσοχή υψηλή υγρασία στο χώρο! !	49,79022
831	22/4/2018 6:37:40 μμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	23,20361
832	22/4/2018 6:38:52 μμ	προσοχή χαμηλή θερμοκρασία στο χώρο! !	23,23013
833	28/4/2018 3:50:08 μμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	25,2146
834	29/4/2018 11:28:56 πμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	25,37374
835	29/4/2018 1:22:47 μμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	25,87047
836	29/4/2018 1:23:54 μμ	προσοχή χαμηλή θερμοκρασία στο χώρο! !	25,85841
844	29/4/2018 2:16:06 μμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	25,83671
845	29/4/2018 2:17:59 μμ	προσοχή υψηλή υγρασία στο χώρο! !	52,86097
846	29/4/2018 2:19:14 μμ	προσοχή χαμηλή θερμοκρασία στο χώρο! !	25,83671
847	29/4/2018 2:20:28 μμ	προσοχή χαμηλή υγρασία στο χώρο! !	52,95501
848	29/4/2018 3:20:54 μμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	25,96692
849	29/4/2018 3:21:58 μμ	προσοχή υψηλή υγρασία στο χώρο! !	51,17549
850	29/4/2018 3:35:31 μμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	26,04167
851	29/4/2018 3:46:27 μμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	26,13812
852	29/4/2018 3:49:24 μμ	προσοχή υψηλή υγρασία στο χώρο! !	51,67824

Εικόνα 46: Το ιστορικό συμβάντων σε αρχείο κειμένου



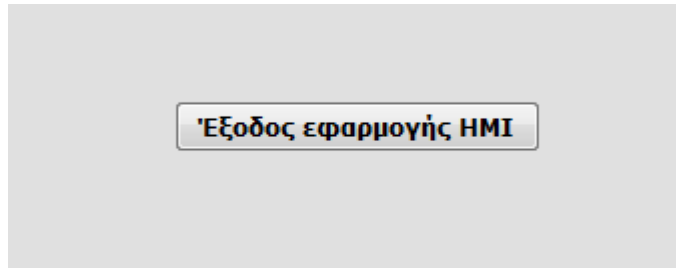
857	29/4/2018 3:58:03 μμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	26,19358
858	29/4/2018 3:59:30 μμ	προσοχή υψηλή υγρασία στο χώρο! !	52,34013
859	29/4/2018 4:01:10 μμ	προσοχή χαμηλή υγρασία στο χώρο! !	52,398
860	29/4/2018 4:04:05 μμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	26,23457
861	29/4/2018 4:05:16 μμ	προσοχή υψηλή υγρασία στο χώρο! !	52,398
862	29/4/2018 4:07:04 μμ	προσοχή χαμηλή θερμοκρασία στο χώρο! !	26,25145
863	29/4/2018 4:09:59 μμ	προσοχή υψηλή υγρασία στο χώρο! !	52,45226
864	6/5/2018 10:18:41 πμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	25,77161
865	6/5/2018 10:38:07 πμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	25,65104
866	6/5/2018 10:48:31 πμ	προσοχή υψηλή θερμοκρασία στο χώρο! !	25,6414
867	6/5/2018 5:30:13 μμ	προσοχή υψηλή υγρασία στο χώρο! !	60,06221
868	6/5/2018 5:35:00 μμ	προσοχή υψηλή υγρασία στο χώρο! !	60,09838
869	6/5/2018 5:36:27 μμ	προσοχή υψηλή υγρασία στο χώρο! !	60,102
870	6/5/2018 5:37:40 μμ	προσοχή υψηλή υγρασία στο χώρο! !	60,1237
871	6/5/2018 6:37:06 μμ	προσοχή υψηλή υγρασία στο χώρο! !	60,05498
872	6/5/2018 8:32:05 μμ	προσοχή υψηλή υγρασία στο χώρο! !	60,1237
873	6/5/2018 9:13:41 μμ	προσοχή υψηλή υγρασία στο χώρο! !	60,15263
874	6/5/2018 9:22:06 μμ	προσοχή υψηλή υγρασία στο χώρο! !	60,12731
875	6/5/2018 9:54:56 μμ	προσοχή υψηλή υγρασία στο χώρο! !	60,01881
876	6/5/2018 9:58:05 μμ	προσοχή υψηλή υγρασία στο χώρο! !	60,04051

Εικόνα 47: Το ιστορικό συμβάντων σε αρχείο κειμένου

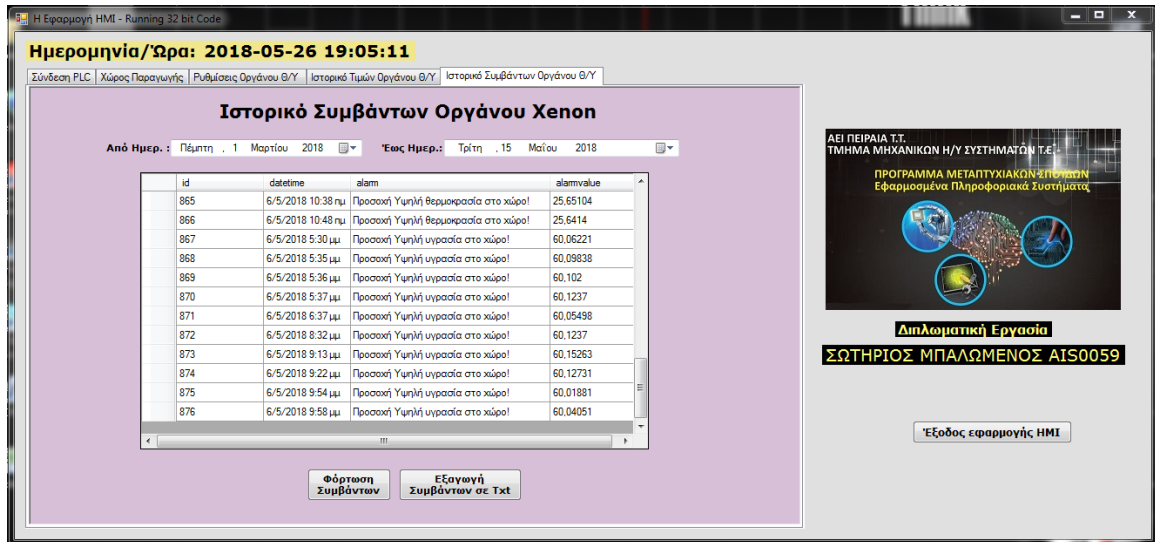
Μέσα από το αρχείο κειμένου δίνεται η δυνατότητα περαιτέρω επεξεργασίας του αλλά και εκτύπωσης του.

### 3.3.8 Έξοδος από την εφαρμογή HMI

Για να βγούμε από την εφαρμογή HMI, θα πρέπει να πατήσουμε το μπουτόν ‘Έξοδος εφαρμογής HMI’ που βρίσκεται στο δεξί κάτω μέρος της οθόνης, όπως φαίνεται στις εικόνες που ακολουθούν.



Εικόνα 48: Το μπουτον για την έξοδο από την εφαρμογή



Εικόνα 49: Στο κάτω δεξί μέρος της οθόνης βρίσκεται το μπουτον για την έξοδο



## **ΚΕΦΑΛΑΙΟ 4**

### **ΜΕΘΟΔΟΛΟΓΙΑ ΚΑΙ ΑΝΑΠΤΥΞΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΑΥΤΟΜΑΤΙΣΜΟΥ**

#### **4.1 Γενικά**

Στο παρακάτω κεφάλαιο θα παρουσιαστεί η μεθοδολογία και τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος αυτοματισμού.

#### **4.2 Ανάπτυξη στην πλευρά του PLC**

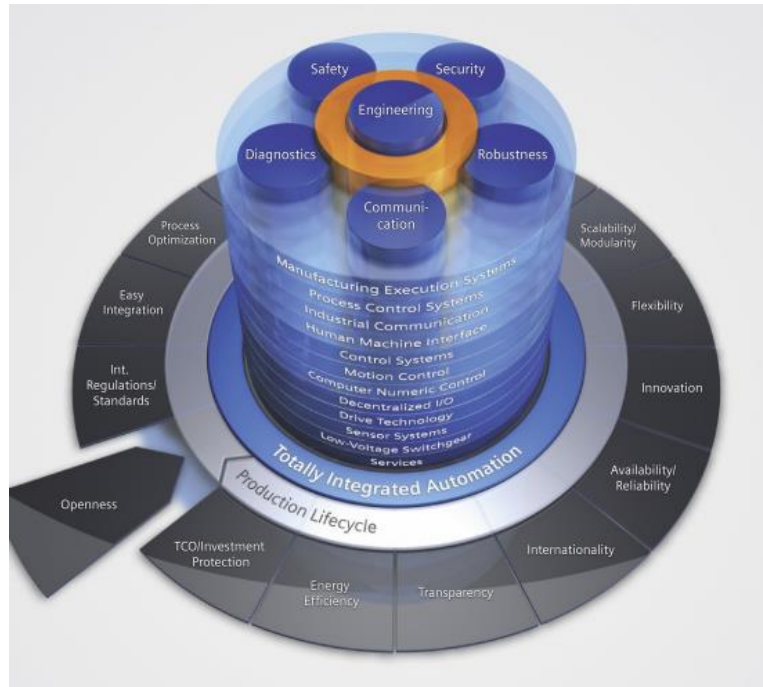
##### **4.2.1 Γενικά**

Όπως έχει αναφερθεί στα προηγούμενα κεφάλαια, για το σύστημα αυτοματισμού χρησιμοποιήθηκε το υλικό PLC σαν ελεγκτής του συστήματος. Το PLC διαβάζει τις μετρήσεις του οργάνου HT206 και στην συνέχεια τις μεταφέρει στην εφαρμογή στον Η/Υ. Επιπλέον το PLC σαν ελεγκτής του συστήματος είναι υπεύθυνος και για τους ελέγχους συναγερμών αλλά και για τις ενέργειες που ακολουθούν όπως η ενεργοποίηση του βομβητή.

##### **4.2.2 Περιβάλλον ανάπτυξης για τον προγραμματισμό του PLC**

Σαν περιβάλλον ανάπτυξης για τον προγραμματισμό του PLC, χρησιμοποιήθηκε το λογισμικό TIA(totally integrated automation) Portal v13 sp2. Το λογισμικό TIA Portal είναι το λογισμικό της εταιρείας Siemens για υλικά αυτοματισμού όπως PLC, HMI(human machine interface), Ρυθμιστές στροφών(Fr. Inverters) κ.α. Η ιδέα πάνω σε αυτό το λογισμικό ήταν να υπάρχει ένα λογισμικό με το οποίο θα μπορούσαν να παραμετροποιήσουν, να σχεδιάσουν και να ελέγξουν διαφορετικούς εξοπλισμούς αυτοματισμού όπως και μεταφορά πληροφοριών από τον έναν εξοπλισμό στον άλλο χωρίς να απαιτείται επιπλέον λογισμικό. Στο TIA Portal ενσωματώνονται άλλα πακέτα λογισμικού για τον κάθε εξοπλισμό.

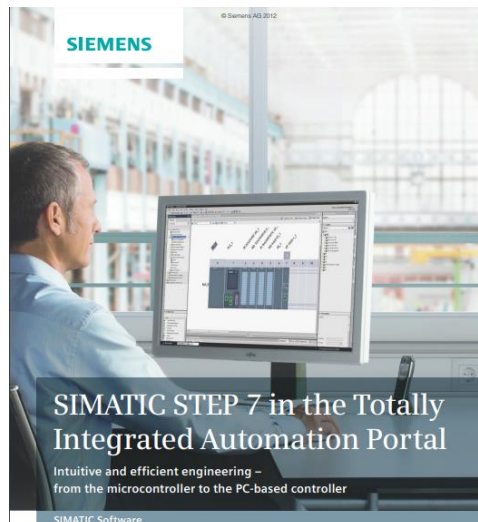




**Εικόνα 50:** Πλήρως αυτοματοποιημένη αυτοματοποίηση (Totally Integrated Automation)

#### 4.2.2.1 Το λογισμικό SIMATIC STEP 7 στο TIA Portal

Για τον προγραμματισμό του PLC χρησιμοποιήθηκε το αντίστοιχο πακέτο προγραμματισμού STEP 7 Professional V13 sp2 το οποίο λειτουργεί σαν πακέτο του TIA Portal. Η σειρά STEP 7 Professional V13 της Siemens είναι το κατάλληλο λογισμικό για παραμετροποίηση και προγραμματισμό των ελεγκτών S7-1200, S7-1500, S7-300, S7-400.



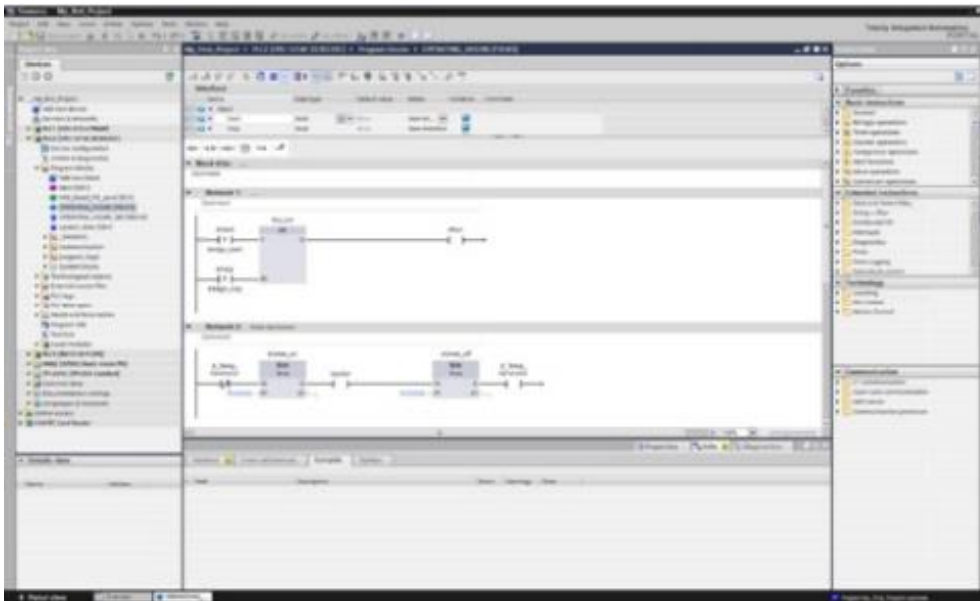
**Εικόνα 51:** Το λογισμικό Simatic Step 7 στο TIA Portal

Οι γλώσσες προγραμματισμού που υποστηρίζει το STEP 7 είναι παρακάτω :

- Οι γλώσσες προγραμματισμού SCL(structured text), LAD(ladder logic), FBD(function block diagram) υποστηρίζονται για όλους τους ελεγκτές.
- Ενώ οι STL(statement list) και γλώσσες προγραμματισμού βημάτων όπως (GRAPH, SFC) υποστηρίζονται μόνο για τους ελεγκτές S7-1500, S7-300 και S7-400.

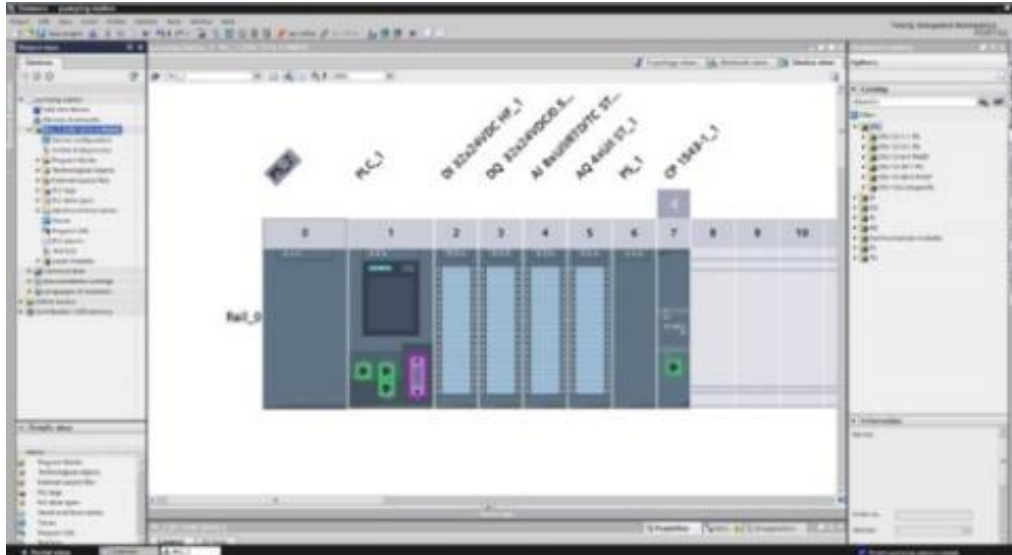
Εδώ θα πρέπει να σημειωθεί ότι η γλώσσα προγραμματισμού με την οποία έχει υλοποιηθεί ο κώδικας στο πρόγραμμα PLC είναι με την γλώσσα LAD(ladder logic).

Παρακάτω ακολουθούν κάποιες εικόνες από το περιβάλλον ανάπτυξης STEP 7 V13.

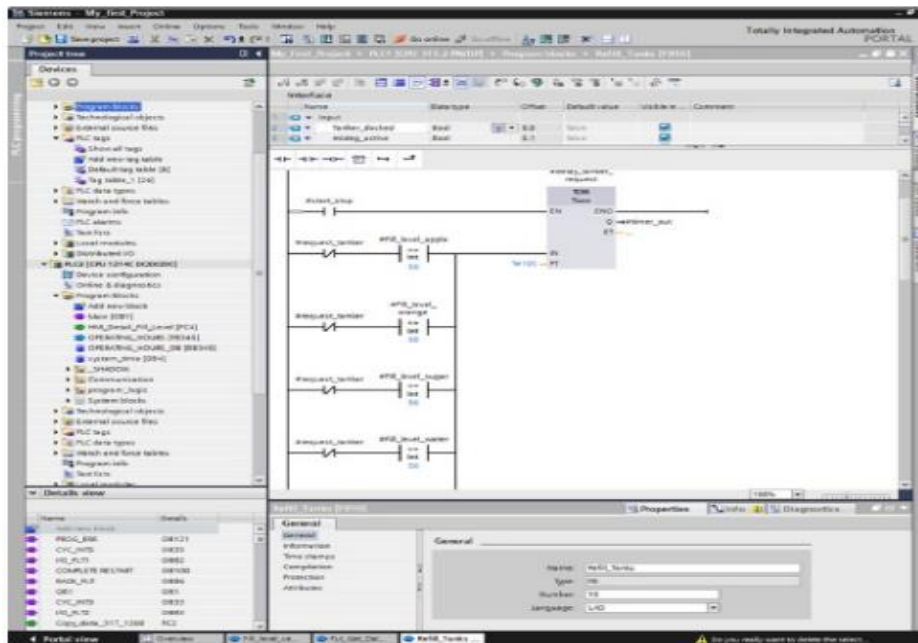


**Εικόνα 52:** Απεικόνιση TIA Portal μέσα από Project

## Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)



Εικόνα 53: Απεικόνιση και παραμετροποίηση συσκευής



Εικόνα 54: Tia Portal editor, Ladder Logic (LAD)

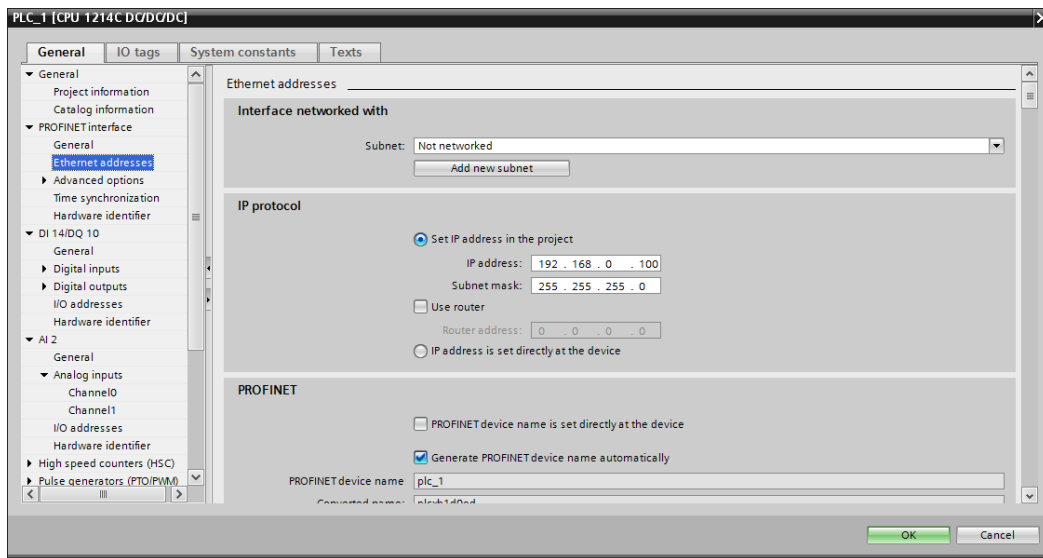
### 4.2.3 Ανάπτυξη προγράμματος εφαρμογής στο PLC

#### 4.2.3.1 Γενικά

Παρακάτω θα παρουσιαστούν συνοπτικά τα βήματα ανάπτυξης προγράμματος στον ελεγκτή S7-1200. Όπως αναφέρθηκε στα προηγούμενα κεφάλαια ο ελεγκτής που χρησιμοποιήσαμε στην εφαρμογή μας ήταν της εταιρείας Siemens σειράς Simatic S7-1200 και συγκεκριμένα το μοντέλο **6ES7 214-1AE30-0XB0(1214C/DC/DC/DC)**.

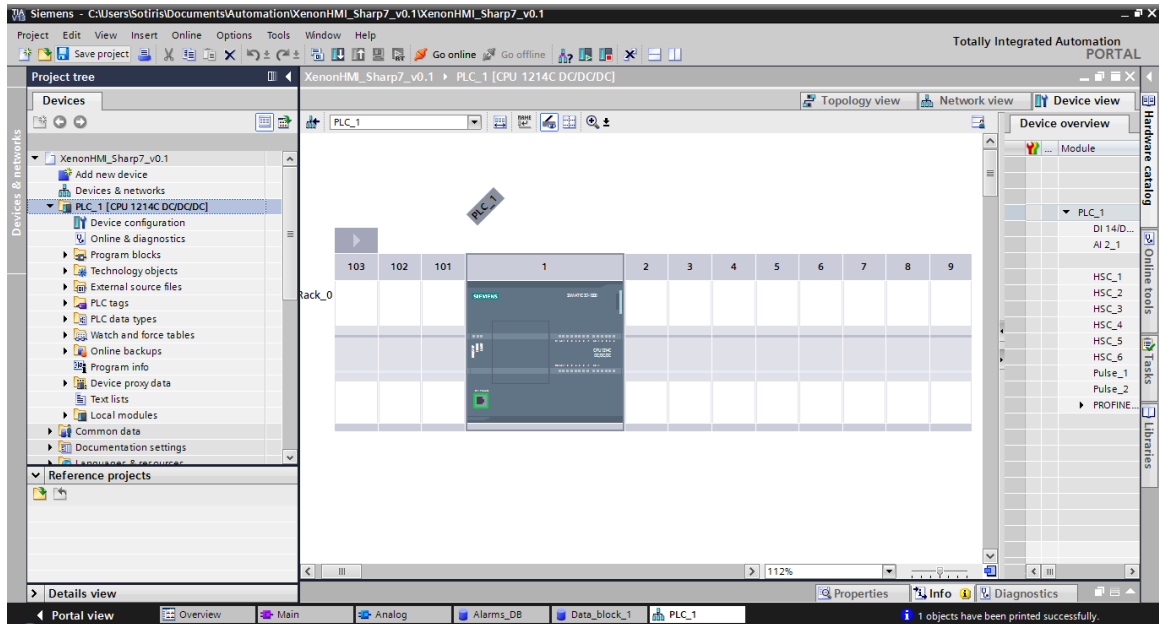
#### 4.2.3.2 Παραμετροποίηση του ελεγκτή στο STEP 7 TIA Portal

Με την έναρξη του project το πρώτο βήμα ήταν ο καθορισμός του ελεγκτή και οι απαιτούμενες παραμετροποιήσεις στο Step 7. Μια από τις σημαντικές παραμέτρους ήταν ο καθορισμός της IP για το PLC. Στο ελεγκτή δώσαμε την **IP: 192.168.0.100** και το **subnet mask: 255.255.255.0**. Παρακάτω φαίνεται η εικόνα από την παραμετροποίηση του ελεγκτή.



Εικόνα 55: Ρύθμιση της διεύθυνσης IP στο PLC

Αφού ολοκληρώσαμε την παραμετροποίηση του ελεγκτή τοπικά στο λογισμικό, στη συνέχεια κατεβάσαμε το πρόγραμμα στο υλικό μέσω της διαδικασίας '**Hardware configuration**'. Στην εικόνα που ακολουθεί φαίνεται το ορισμένο υλικό S7-1200 CPU 1214C/DC/DC/DC στο Step 7 Tia Portal.

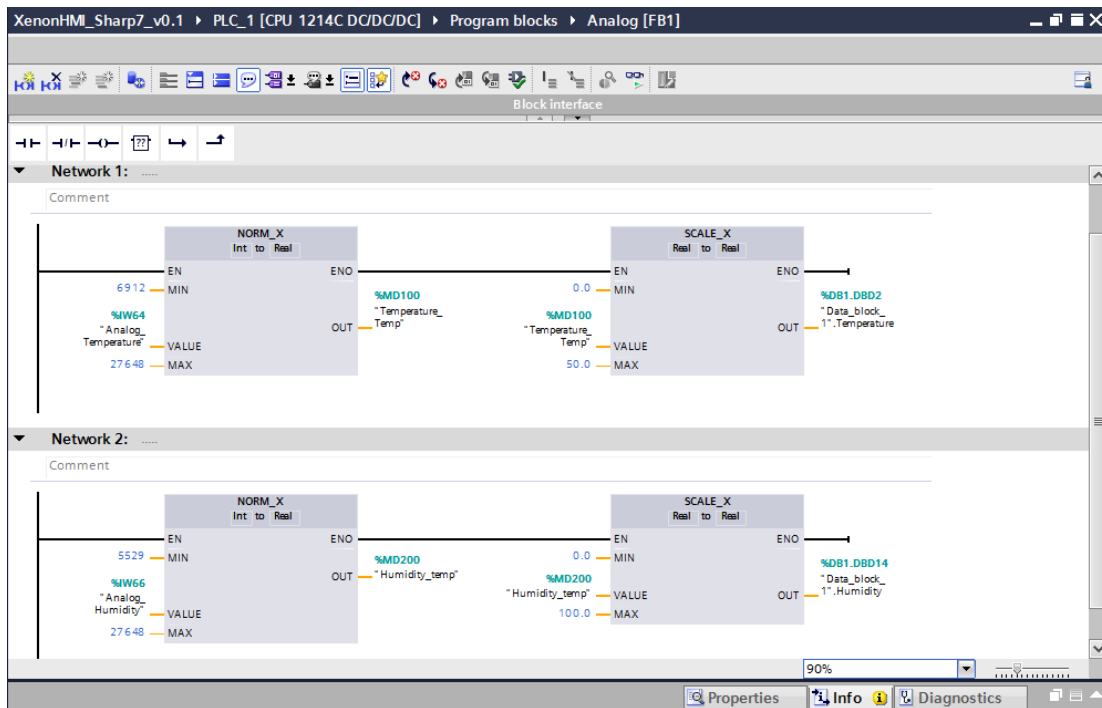


Εικόνα 56: Η CPU 1214C/DC/DC/DC στο Step 7 Tia Portal

#### 4.2.3.3 Σύνδεση του οργάνου μέτρησης στο PLC

Ένα σημαντικό σημείο της εργασίας μας, ήταν η σύνδεση του οργάνου στο PLC. Το PLC διαθέτει δύο αναλογικές εισόδους(0-10Volt) τις οποίες θα χρησιμοποιήσουμε ώστε να συνδέσουμε το όργανο. Το όργανο θερμοκρασίας-υγρασίας διαθέτει δυνατότητα μεταφοράς των μετρούμενων τιμών μέσω αναλογικών εξόδων(4-20mA). Για να υλοποιήσουμε την σύνδεση αυτή χρειαστήκαμε δύο αντιστάσεις 500Ω, μία για το κάθε αναλογικό κανάλι του PLC.

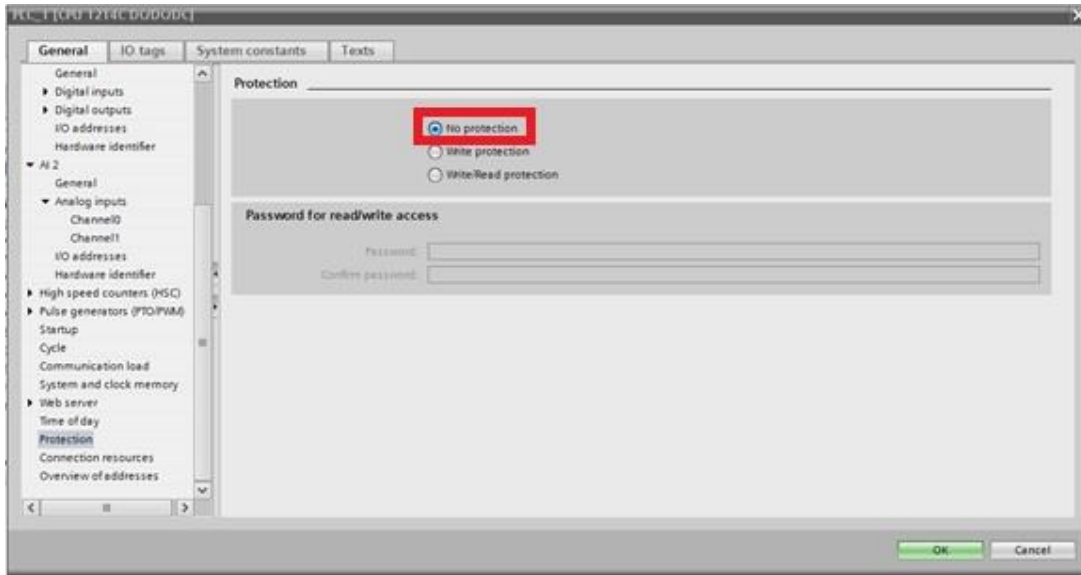
Αφού συνδέσαμε καλωδιακά το όργανο, επόμενο βήμα ήταν να γραφτεί ο αντίστοιχος κώδικας στο PLC ώστε να μπορούμε να διαβάζουμε τις τιμές του οργάνου. Εδώ χρησιμοποιήσαμε τις εντολές **NORM\_X:Normalize** καθώς και το **SCALE\_X:Scale**. Ο κώδικας για το αναλογικό όργανο έχει γραφτεί σε ένα ξεχωριστό μπλοκ για την καλύτερη δομή του προγράμματος με όνομα **Analog(FB1)**. Στην εικόνα που ακολουθεί φαίνεται ο κώδικας για την ανάγνωση του οργάνου μέτρησης από το PLC.



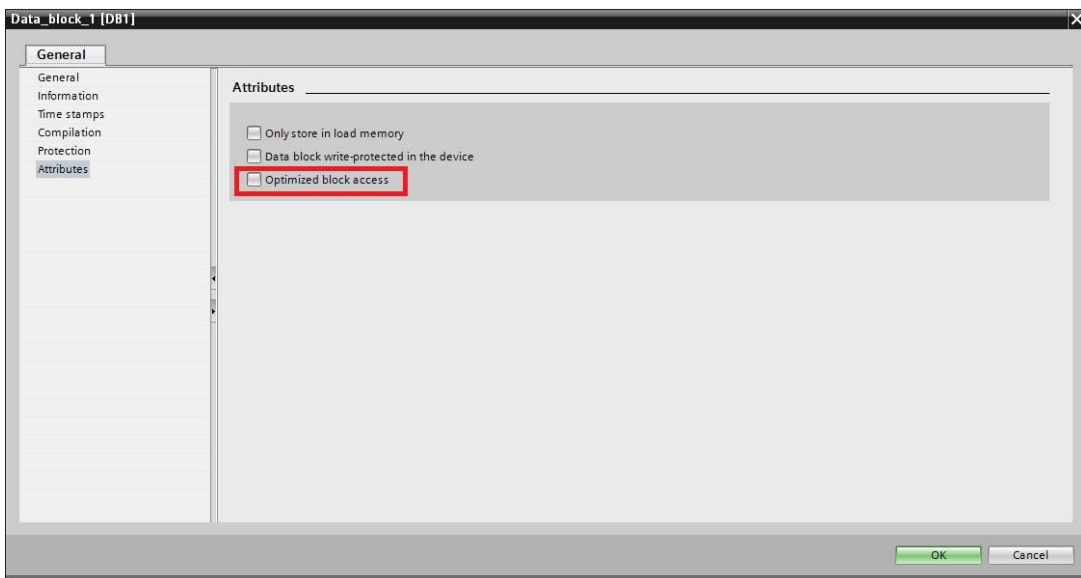
Εικόνα 57: Ο κώδικας για την ανάγνωση τιμών από το όργανο HT206

#### 4.2.3.4 Τρόπος επικοινωνίας μεταξύ PLC και Εφαρμογής HMI

Μία από τις σημαντικότερες εργασίες για την υλοποίηση του συστήματος ήταν η επικοινωνία μεταξύ PLC και της εφαρμογής HMI. Εδώ εξετάστηκαν αρχικά αρκετές λύσεις όπως την λειτουργία **TCP:SEND**, **TCP:RCV** που είναι διαθέσιμη για το PLC S7-1200. Στη συνέχεια όμως αποφασίσαμε να χρησιμοποιήσουμε έναν οδηγό(driver) ο οποίος έχει γραφτεί από τον κο Davide Nardella συγγραφέα της βιβλιοθήκης Snap 7(<http://snap7.sourceforge.net/>). Ο **Sharp 7** είναι ένας οδηγός σε γλώσσα προγραμματισμού C# για να επικοινωνεί με ελεγκτές S7 PLC και μέσω αυτού να υπάρχει επικοινωνία δεδομένων προς άλλες εφαρμογές σε γλώσσα προγραμματισμού C#. Για να μπορέσουμε να δημιουργήσουμε την επικοινωνία χρειάστηκε να κάνουμε κάποιες παραμετροποιήσεις στον ελεγκτή όπως φαίνονται στις εικόνες που ακολουθούν.



Εικόνα 58: Ρύθμιση No Protection στη CPU



Εικόνα 59: Ρύθμιση Optimized block access στη CPU

Επιπλέον χρησιμοποιήθηκε μία περιοχή μνήμης(data block) μέσω της οποίας θα υπάρχει αμφίδρομη επικοινωνία από το PLC προς την εφαρμογή στο PC αλλά και το ανάποδο. Όλα τα δεδομένα δηλαδή που δημιουργούνται στον κώδικα PLC, μεταφέρονται στο **DB1**(data block 1). Αντιστοίχως ότι δεδομένα θα δημιουργούνται στην εφαρμογή στο PC θα γράφονται μέσα στο DB1. Στην εικόνα που ακολουθεί φαίνεται η περιοχή μνήμης (DB1) που έχει δημιουργηθεί στον κώδικα PLC και τα αντίστοιχα δεδομένα.

Name	Data type	Offset	Default value	Start value	Monitor value	Retain	Accessible f...	Visible in ...	S...
Static									
Timer ON	Bool	0.0	false	false	FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Temperature	Real	2.0	0.0	0.0	27.23765	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Temperature_HL	Real	6.0	0.0	25.0	25.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Temperature_LL	Real	10.0	0.0	15.0	15.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Humidity	Real	14.0	0.0	0.0	42.89525	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Humidity_HL	Real	18.0	0.0	60.0	60.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Humidity_LL	Real	22.0	0.0	30.0	30.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Temperature_Delay	Dint	26.0	0	30000	30000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Humidity_Delay	Dint	30.0	0	30000	30000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Alarms	Bool	34.0	false	false	FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
ACK_Siren	Bool	34.1	false	false	TRUE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Temperature High Alarm	Bool	34.2	false	0	TRUE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Temperature Low Alarm	Bool	34.3	false	0	FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Humidity High Alarm	Bool	34.4	false	0	FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Humidity Low Alarm	Bool	34.5	false	0	FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Εικόνα 60: Το Data Block 1 του προγράμματος PLC

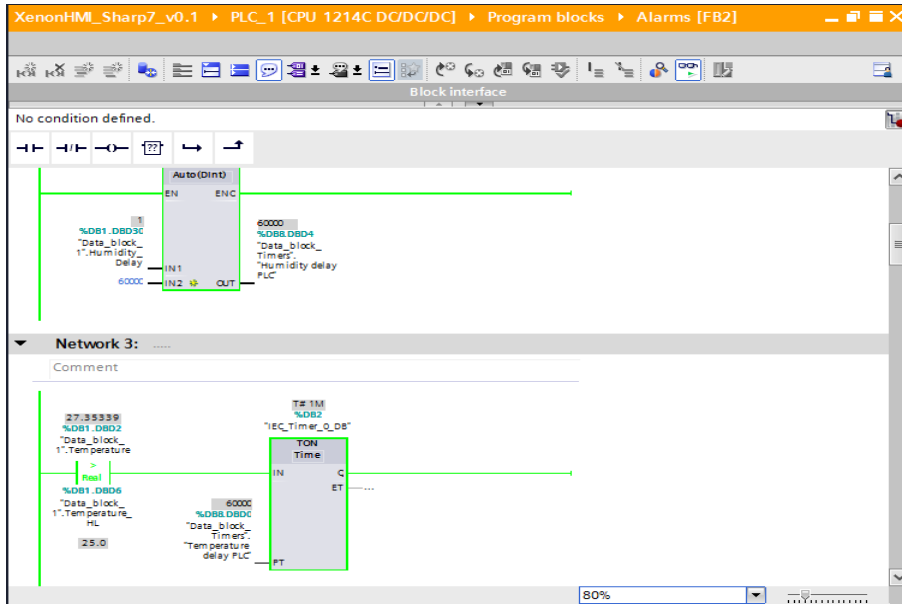
#### 4.2.3.5 Οι έλεγχοι συναγερμών στο πρόγραμμα του PLC

Ένα τμήμα του προγράμματος αφορούσε τους ελέγχους οι οποίοι πρέπει να υπάρχουν στην εφαρμογή. Πιο συγκεκριμένα το πρόγραμμα στο PLC θα έπρεπε να ελέγχει τις τιμές της θερμοκρασίας και υγρασίας και να τις συγκρίνει με τα όρια τιμών κανονικής λειτουργίας που έχουν δοθεί από την εφαρμογή HMI στο PC. Στην περίπτωση που αυτά τα όρια έχουν ξεπεραστεί μαζί με την αντίστοιχο χρόνο καθυστέρησης, το PLC ενεργοποιεί συναγερμό (alarm). Ο συναγερμός αυτός στέλνεται στην εφαρμογή HMI και παράλληλα ενεργοποιείται και ένας ηχητικός βομβητής.

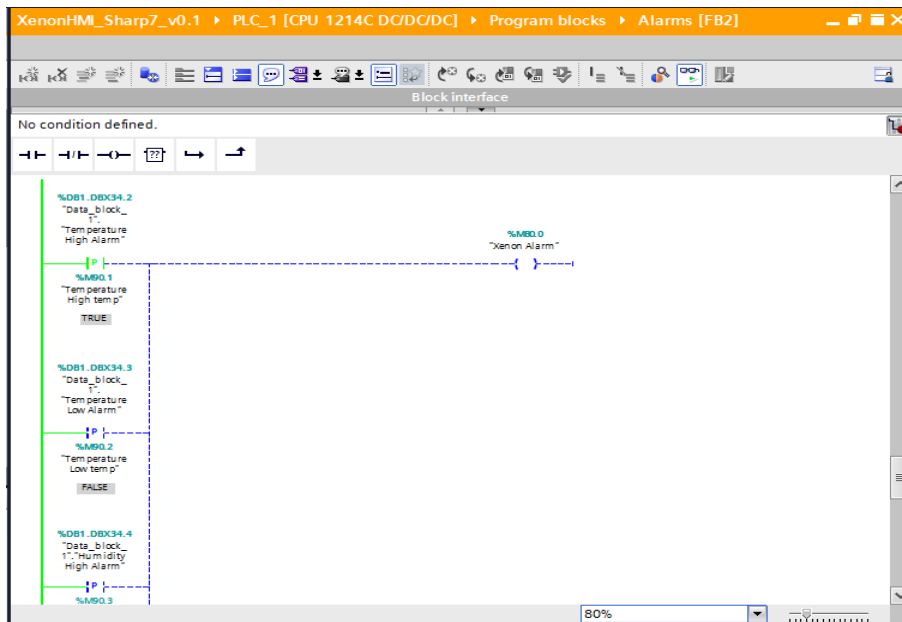
Έτσι ο χρήστης ακόμα και αν δεν βρίσκεται στην οθόνη της εφαρμογής μπορεί να ειδοποιηθεί για τις αποκλίσεις που έχει το σύστημα του και να προβεί στις απαραίτητες ενέργειες. Αυτό το τμήμα του κώδικα έχει γραφτεί σε ξεχωριστό μπλοκ λειτουργίας (function block, FB2) για την καλύτερη δομή του προγράμματος. Στις εικόνες που ακολουθούν φαίνονται τμήματα από τον κώδικα για τους συναγερμούς της εφαρμογής και τις αντίστοιχες ενέργειες.



## Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)



Εικόνα 61: Το μπλοκ προγράμματος για τους συναγερμούς σε Ladder Logic(LAD)



Εικόνα 62: Το μπλοκ προγράμματος για τους συναγερμούς σε Ladder Logic(LAD)

### 4.3 Ανάπτυξη εφαρμογής στον Η/Υ

#### 4.3.1 Γενικά

Όπως είναι γνωστό από τα προηγούμενα κεφάλαια το σύστημα μας αποτελείται και από μία εφαρμογή της οποίας η ανάπτυξη έχει γίνει σε Η/Υ. Η εφαρμογή HMI

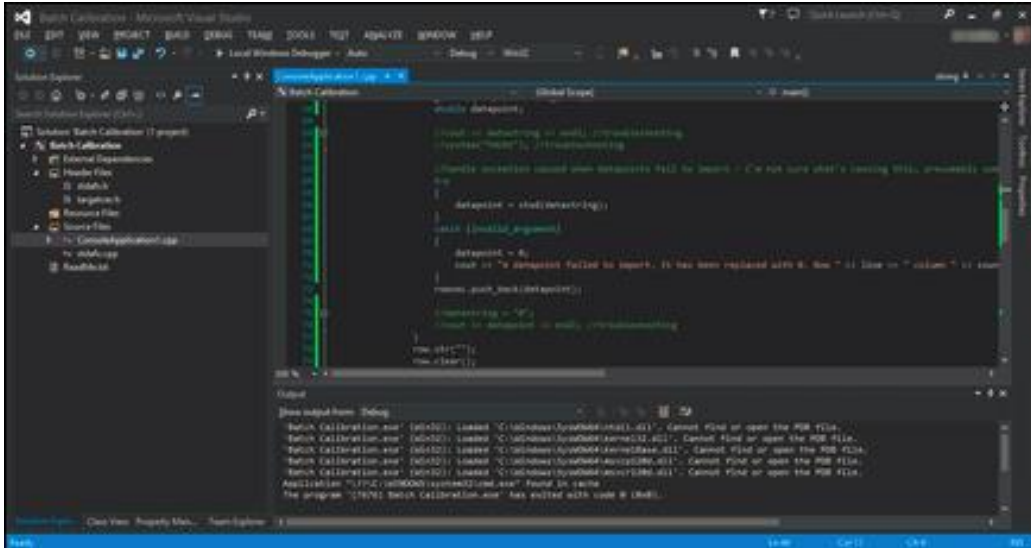
όπως έχει ονομαστεί είναι το εργαλείο εποπτικού ελέγχου και συλλογής δεδομένων(SCADA) για το σύστημα μας και ο χρήστης έχει αρκετές δυνατότητες όπως παρουσιάστηκαν αναλυτικά στο κεφάλαιο 3. Στο παρών κεφάλαιο θα παρουσιαστούν συνοπτικά τα βήματα για την ανάπτυξη εφαρμογής σε περιβάλλον Windows 7.

#### **4.3.2 Περιβάλλον ανάπτυξης για την Εφαρμογή στον Η/Υ**

Για την ανάπτυξη της εφαρμογής μας, χρησιμοποιήσαμε ένα ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment, IDE). Ένα τέτοιο περιβάλλον ανάπτυξης είναι μία πλατφόρμα λογισμικού που βοηθάει στην ανάπτυξη λογισμικών σε ηλεκτρονικό υπολογιστή. Συνήθως μία πλατφόρμα ανάπτυξης περιλαμβάνει κάποιον επεξεργαστή πηγαίου κώδικα, μεταγλωττιστή, απασφαλματωτή και διάφορα εργαλεία για την δημιουργία γραφικού περιβάλλοντος για τον χρήστη.

Η πλατφόρμα που επιλέξαμε για την ανάπτυξη της εφαρμογής μας είναι το Visual Studio από την εταιρεία Microsoft.

Το Visual studio χρησιμοποιείται για την ανάπτυξη λογισμικών σε ηλεκτρονικό υπολογιστή όπως web εφαρμογές, ιστοσελίδες, εφαρμογές σε κινητά κ.α. Επιπλέον χρησιμοποιεί διάφορες άλλες πλατφόρμες ανάπτυξης της Microsoft όπως είναι η Διεπαφή Προγραμματισμού Εφαρμογών των Windows (Windows API), Παραθυρικές φόρμες(Windows forms) και άλλες παραθυρικές εφαρμογές. Το visual studio υποστηρίζει πάνω από 30 γλώσσες προγραμματισμού μερικές από τις οποίες είναι οι C, C++, C#, visual basic, JavaScript, XML κτλ.



Εικόνα 63: Το περιβάλλον ανάπτυξης Visual Studio

### 4.3.3 Ανάπτυξη παραθυρικής εφαρμογής HMI στον Η/Υ

#### 4.3.3.1 Γενικά

Παρακάτω θα παρουσιαστούν συνοπτικά τα βήματα ανάπτυξης για την εφαρμογή HMI στον Η/Υ. Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο η ανάπτυξη της εφαρμογής έγινε σε περιβάλλον Visual Studio.

#### 4.3.3.2 Τα εργαλεία ανάπτυξης της εφαρμογής HMI

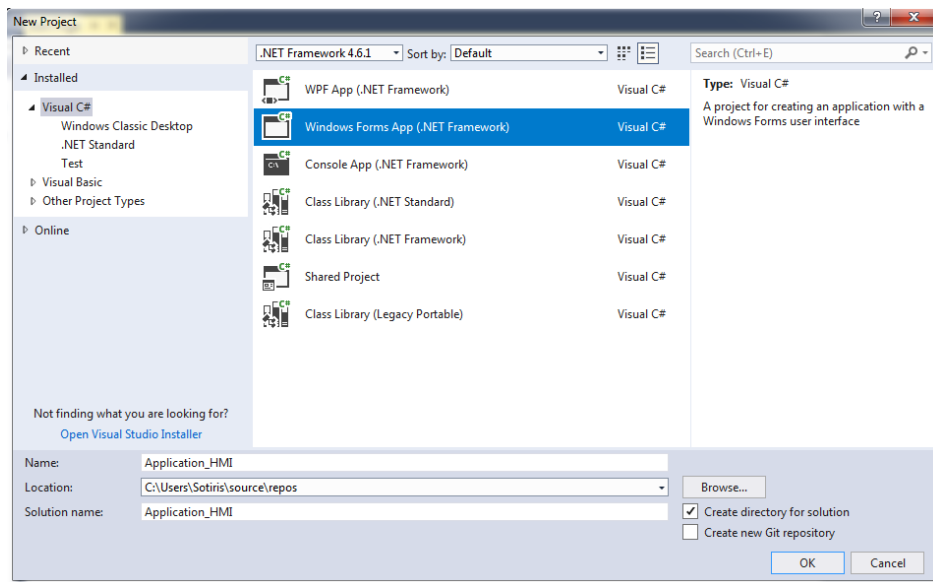
Αρχικά, για την δημιουργία της εφαρμογής έπρεπε να συγκεντρώσουμε και να εγκαταστήσουμε στον Η/Υ τα λογισμικά που απαιτούνται για την ανάπτυξης της. Παρακάτω αναφέρονται τα πακέτα λογισμικών που χρειάστηκαν για την υλοποίηση της εφαρμογής.

- Visual Studio Community 2017 το οποίο διατίθεται δωρεάν. Επιπλέον χρησιμοποιήσαμε το πακέτο .NET desktop development το οποίο απαιτείται για την δημιουργία παραθυρικών εφαρμογών.
- Για την βάση δεδομένων χρησιμοποιήσαμε την MySql Database μέσα από το περιβάλλον Xampp έκδοση 4.7.4.

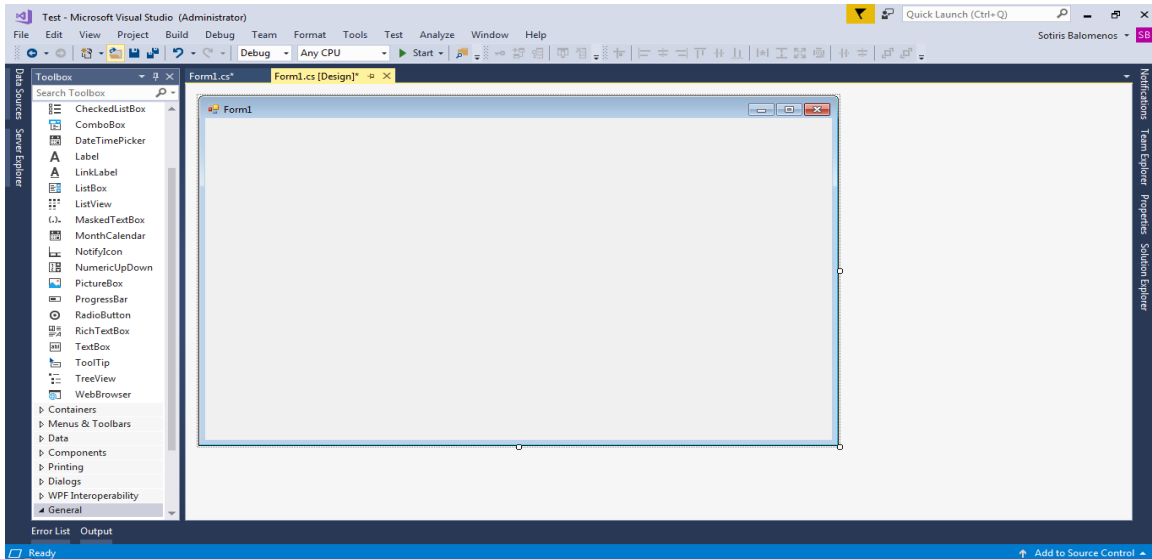
- Για την σύνδεση .NET με την Βάση δεδομένων MySQL χρησιμοποιήσαμε την παρακάτω βιβλιοθήκη (<https://dev.mysql.com/downloads/connector/net/>)
- Για την σύνδεση .NET με PLC Siemens S7, χρησιμοποιήσαμε τον οδηγό(driver) Sharp . Η βιβλιοθήκη Snap7 έχει γραφτεί από τον κο Davide Nardella και μπορεί να βρεθεί στον σύνδεσμο (<https://sourceforge.net/projects/snap7/files/Sharp7/>)

#### 4.3.3.3 Η δημιουργία παραθυρικής εφαρμογής HMI

Αφού συγκεντρώσαμε τα παραπάνω λογισμικά, δημιουργήσαμε ένα νέο project **Windows Forms App** στο visual studio, με το όνομα **Application\_HMI** όπως φαίνεται στις παρακάτω εικόνες



Εικόνα 64: Η παραθυρική εφαρμογή HMI



Εικόνα 65: Η φόρμα 1(Form1) της εφαρμογής HMI

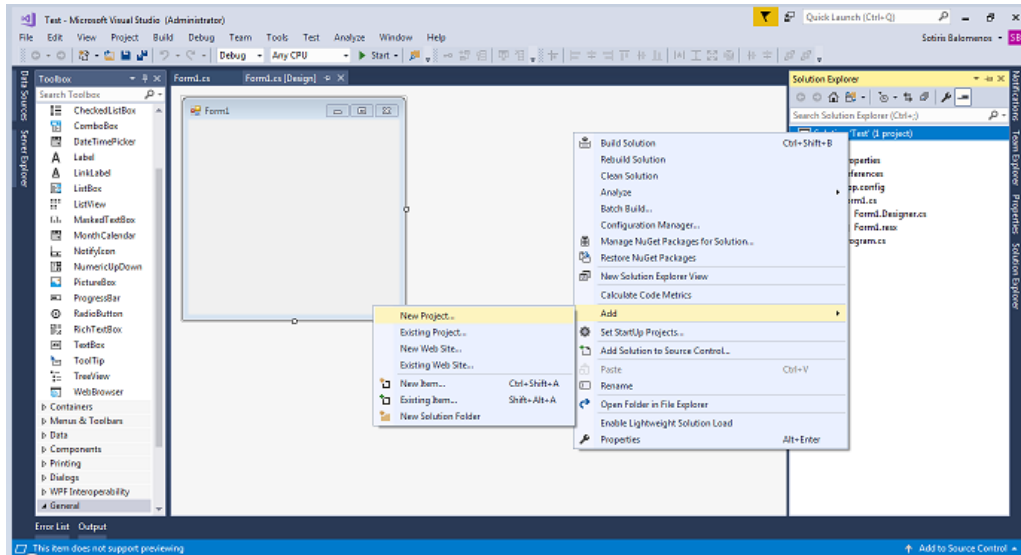
### 4.3.3.3 Η επικοινωνία παραθυρικής εφαρμογής HMI με το PLC

Ένα από τα πιο σημαντικά σημεία της εφαρμογής ήταν να δημιουργήσουμε την σύνδεση εφαρμογής HMI και PLC. Όπως αναφέραμε παραπάνω για να επιτύχουμε την επικοινωνία αυτή, χρησιμοποιήσαμε τον έτοιμο οδηγό(driver) Sharp7 ο οποίος έχει γραφτεί από τον κο Davide Nardella.

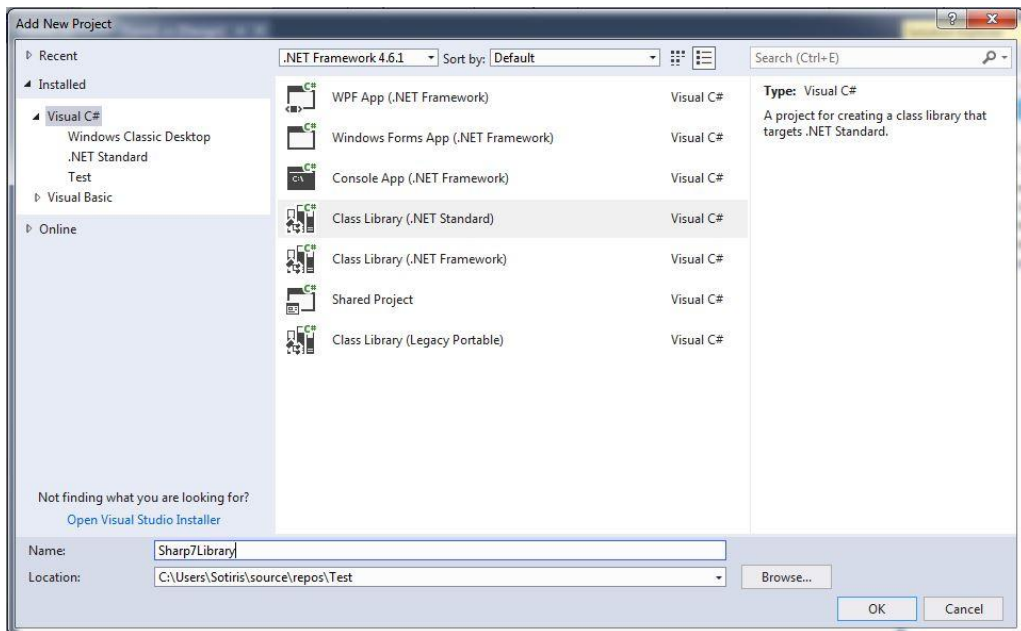
Το Sharp7 είναι ένα ανοικτό λογισμικό γραμμένο σε γλώσσα προγραμματισμού C# το οποίο εκτελεί το πρωτόκολλο S7(siemens). Είναι ανεπτυγμένο σε ένα απλό αρχείο κώδικα το οποίο περιέχει κάποιες κλάσεις οι οποίες μπορούν να χρησιμοποιηθούν άμεσα σε project **.NET** για να επικοινωνήσει με PLCs σειράς S7 Siemens. Το Sharp7 διανέμεται ως μία βιβλιοθήκη πηγαίου κώδικα κάτω από την Γενική Άδεια Δημόσιας Χρήσης GNU.

Αρχικά για να υλοποιήσουμε την σύνδεση, δημιουργήσαμε μία νέα κλάση στο project , **Sharp7Library** όπως φαίνεται παρακάτω

## Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)

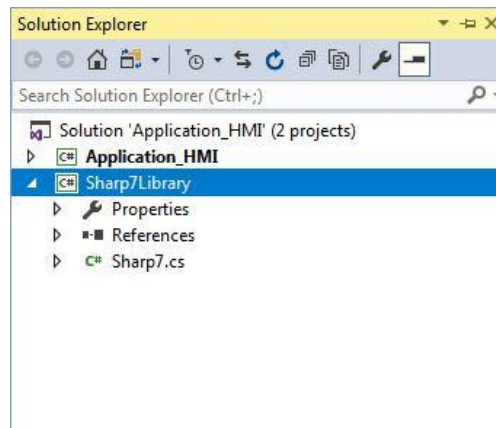


Εικόνα 66: Πρόσθεση κλάσης στο project



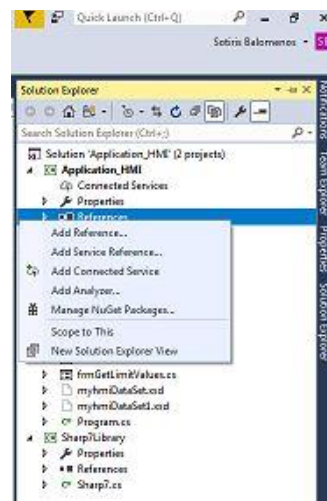
Εικόνα 67: Δημιουργία βιβλιοθήκη Κλάσης

## Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)

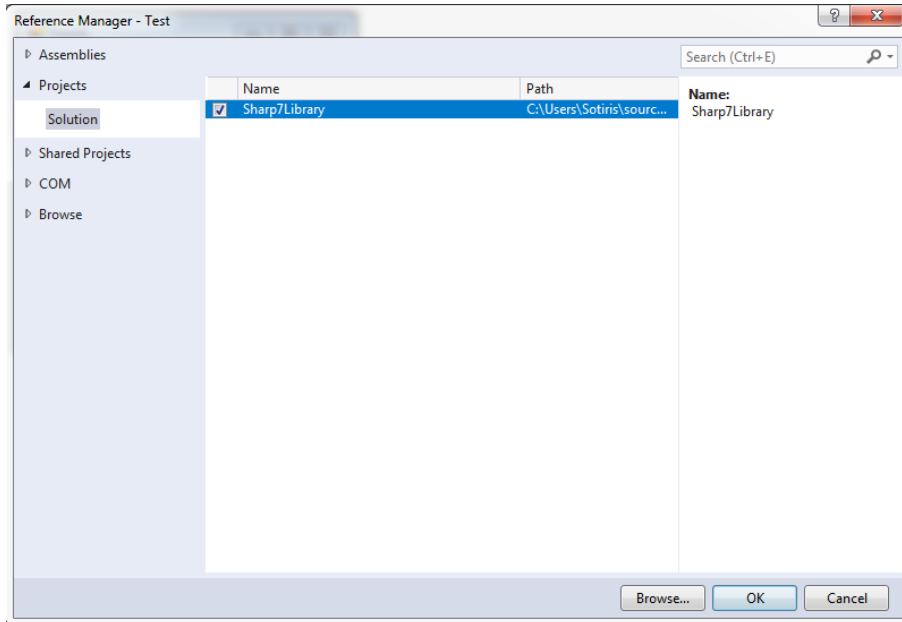


Εικόνα 68: Η βιβλιοθήκη Sharp7

Στη συνέχεια δημιουργούμε μία αναφορά **Sharp7(reference)** σε όλο το project.



Εικόνα 69: Η δημιουργία αναφοράς στο project



Εικόνα 70: Σύνδεση της αναφοράς με τον οδηγό που έχουμε εγκατεστημένο στον Η/Υ

Αφού δημιουργήσαμε τον οδηγό, επόμενο βήμα ήταν να δημιουργήσουμε την επικοινωνία με το PLC. Στην παρακάτω εικόνα φαίνεται ο κώδικας για την σύνδεση και την αποσύνδεση της εφαρμογής με το PLC.

```
private S7Client Client;

private void Connectbtn_Click(object sender, EventArgs e)
{
    int Result;
    Result = Client.ConnectTo(txtIP.Text, 0, 0);
    ShowResult(Result);

    if (Result == 0)
    {
        txtError.Text = txtError.Text + ", To PLC συνδέθηκε: " + Client.Connected;
        txtError.Text = txtError.Text + ", PDU Negotiated : " + Client.PduSizeNegotiated.ToString();
        txtIP.Enabled = false;
        // TxtRack.Enabled = false;
        // TxtSlot.Enabled = false;
        Connectbtn.Enabled = false;
        Disconnectbtn.Enabled = true;
        // tabControl.Enabled = true;
    }
}

private void Disconnectbtn_Click(object sender, EventArgs e)
{
    Client.Disconnect();
    txtError.Text = "To PLC Αποσυνδέθηκε";
    txtIP.Enabled = true;
    // TxtRack.Enabled = true;
    // TxtSlot.Enabled = true;
    Connectbtn.Enabled = true;
    Disconnectbtn.Enabled = false;
    // tabControl.Enabled = false;
}
}
```

Εικόνα 71: Ο κώδικας της σύνδεσης-αποσύνδεσης με το PLC

#### 4.3.3.3 Η ανάγνωση τιμών PLC μέσω του Data Block 1

Η ανάγνωση δεδομένων από το PLC πραγματοποιείται μέσω της λειτουργίας “ανάγνωσης περιοχής” (read area) που υπάρχει στο Sharp7. Εκεί υπάρχουν



αρκετές δυνατότητες ώστε να διαβάσεις μία τιμή από ένα PLC όπως ανάγνωση εισόδων, ανάγνωση εξόδων, ανάγνωση χρονοδιακοπών(timers), ανάγνωση μετρητών(counters) και τέλος ανάγνωση περιοχών μνήμης του PLC, γνωστές σαν Data Block. Στην εφαρμογή μας έχουμε επιλέξει την μέθοδο ανάγνωσης μέσω data block οπότε τα επιθυμητά δεδομένα για ανάγνωση έχουν μπει και από την πλευρά του PLC σε αυτό το σημείο, συγκεκριμένα στο Data block 1(DB1).

Στον κώδικα της εφαρμογής θα πρέπει να δημιουργηθεί ένα **απόθεμα τιμών(buffer)** για τις τιμές που διαβάζει από το data block 1 του PLC. Στην συνέχεια αυτές οι τιμές θα πρέπει να γίνουν μετατροπή από S7 σε C# format ώστε να απεικονιστούν στην εφαρμογή HMI.

Τέλος θα πρέπει να σημειωθεί ότι χρησιμοποιείται ένα χρονικό στην C# εφαρμογή ώστε οι τιμές αυτές από το DB 1 να ανανεώνονται συνεχώς.

Παρακάτω φαίνεται ένα απόσπασμα από τον κώδικα ανάγνωσης του data block 1.

```
public void ReadDB1()
{
    if (Client.Connected)
    {
        byte[] db1Buffer = new byte[35];
        int Result;
        Result = Client.DBRead(1, 0, 35, db1Buffer);
        if (Result != 0)
        {
            Console.WriteLine("Error: " + Client.ErrorText(Result));
        }
        ShowResult(Result);
        float db1dbd2 = S7.GetRealAt(db1Buffer, 2);
        float db1dbd6 = S7.GetRealAt(db1Buffer, 6);
        float db1dbd10 = S7.GetRealAt(db1Buffer, 10);
        float db1dbd14 = S7.GetRealAt(db1Buffer, 14);
        float db1dbd18 = S7.GetRealAt(db1Buffer, 18);
        float db1dbd22 = S7.GetRealAt(db1Buffer, 22);
        Int32 db1dbd26 = S7.GetDIntAt(db1Buffer, 26);
        Int32 db1dbd30 = S7.GetDIntAt(db1Buffer, 30);

        bool db1dbd34_0 = S7.GetBitAt(db1Buffer, 34, 0);
        bool db1dbd34_2 = S7.GetBitAt(db1Buffer, 34, 2);
        bool db1dbd34_3 = S7.GetBitAt(db1Buffer, 34, 3);
        bool db1dbd34_4 = S7.GetBitAt(db1Buffer, 34, 4);
        bool db1dbd34_5 = S7.GetBitAt(db1Buffer, 34, 5);

        txtTemperature.Text = db1dbd2.ToString();
        txtHumidity.Text = db1dbd14.ToString();
    }
}
```

**Εικόνα 72:** Κώδικας ανάγνωσης του Data Block 1 από το PLC

Data_block_1						
	Name	Data type	Offset	Default value	Start value	Monitor value
1	Static					
2	Timer ON	Bool	0.0	false	false	FALSE
3	Temperature	Real	2.0	0.0	0.0	29.34992
4	Temperature_HL	Real	6.0	0.0	25.0	25.0
5	Temperature_LL	Real	10.0	0.0	15.0	15.0
6	Humidity	Real	14.0	0.0	0.0	31.38478
7	Humidity_HL	Real	18.0	0.0	60.0	60.0
8	Humidity_LL	Real	22.0	0.0	30.0	30.0
9	Temperature_Delay	DInt	26.0	0	1	1
10	Humidity_Delay	DInt	30.0	0	1	1
11	Alarms	Bool	34.0	false	false	FALSE
12	ACK_Siren	Bool	34.1	false	false	FALSE
13	Temperature High Alarm	Bool	34.2	false	0	TRUE
14	Temperature Low Alarm	Bool	34.3	false	0	FALSE
15	Humidity High Alarm	Bool	34.4	false	0	FALSE
16	Humidity Low Alarm	Bool	34.5	false	0	FALSE

Εικόνα 73: Το Data Block 1 στο PLC

#### 4.3.3.5 Η εγγραφή τιμών στο Data Block 1 του PLC

Η εγγραφή τιμών πραγματοποιείται μέσα από την λειτουργία ‘εγγραφή περιοχής’ του Sharp7. Είναι η αντίστοιχη λειτουργία της ‘ανάγνωσης περιοχής’ με τις αντίστοιχες δυνατότητες σχετικά με τους τύπους δεδομένων που μπορείς να χρησιμοποιήσεις (DB, χρονοδιακόπτες, μετρητές, εισόδους-εξόδους plc). Θα πρέπει και εδώ να δημιουργήσουμε ένα απόθεμα (buffer) και αφού γίνει η μετατροπή των τιμών σε από C# σε S7 format, να καλούμε την λειτουργία της εγγραφής. Τέλος δημιουργούμε ένα χρονικό ώστε να ανανεώνετε συνεχώς η λειτουργία της εγγραφής στο DB1 του PLC.

Στην παρακάτω εικόνα φαίνεται ο κώδικας για την λειτουργία της εγγραφής στην εφαρμογής μας.

```

public void WriteDB1()
{
    byte[] db1Buffer = new byte[30];
    const int START_INDEX = 6;

    float TempHigh = Convert.ToSingle(txtTempHigh.Text);
    float TempLow = Convert.ToSingle(txtTempLow.Text);
    float HumiHigh = Convert.ToSingle(txtHumiHigh.Text);
    float HumiLow = Convert.ToSingle(txtHumiLow.Text);
    Int32 TempDelay = Convert.ToInt32(txtTempDelay.Text);
    Int32 HumiDelay = Convert.ToInt32(txtHumiDelay.Text);

    S7.SetRealAt(db1Buffer, 6 - START_INDEX, TempHigh);
    S7.SetRealAt(db1Buffer, 10 - START_INDEX, TempLow);
    S7.SetRealAt(db1Buffer, 18 - START_INDEX, HumiHigh);
    S7.SetRealAt(db1Buffer, 22 - START_INDEX, HumiLow);
    S7.SetDIntAt(db1Buffer, 26 - START_INDEX, TempDelay);
    S7.SetDIntAt(db1Buffer, 30 - START_INDEX, HumiDelay);

    int Result;

    Result = Client.DBWrite(1, START_INDEX, db1Buffer.Length, db1Buffer);

    if (Result != 0)
    {
        Console.WriteLine("Error: " + Client.ErrorText(Result));
    }
}

```

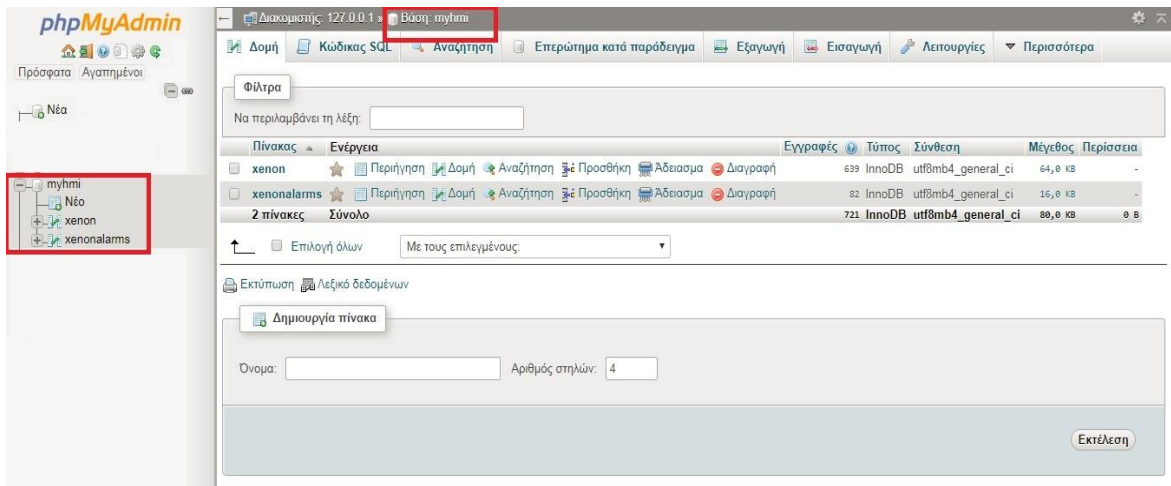
**Εικόνα 74:** Ο κώδικας εγγραφής στο Data Block 1 του PLC

#### 4.3.3.6 Η Δημιουργία Ιστορικού τιμών και συναγερμών

Όπως έχει αναφερθεί και στα προηγούμενα κεφάλαια η εφαρμογή μας υποστηρίζει την καταγραφή ιστορικού τόσο για τις τιμές του οργάνου μέτρησης όσο και των συναγερμών. Για την υλοποίηση αυτής της δυνατότητας, χρειάστηκε να δημιουργήσουμε μία βάση δεδομένων. Η Βάση δεδομένων που χρησιμοποιήθηκε ήταν η **MySQL Database** μέσα από το περιβάλλον **Xampp**.

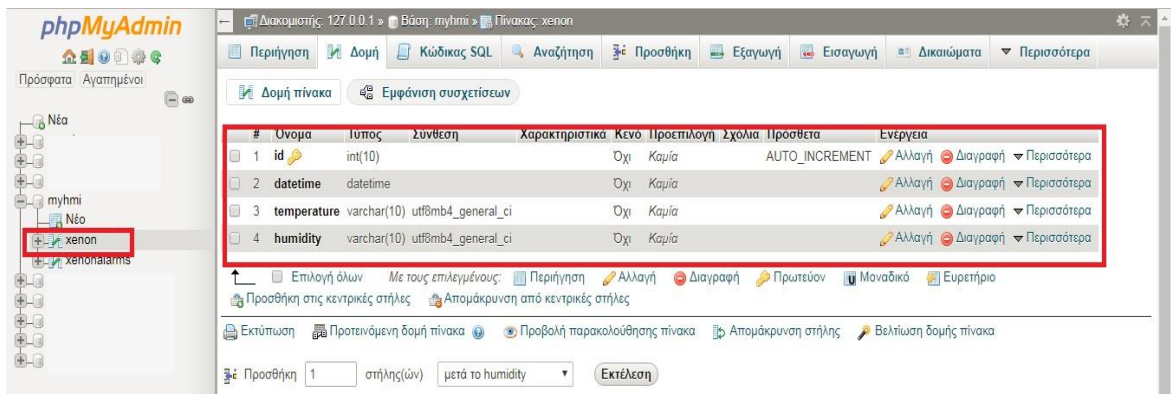
Αρχικά δημιουργήθηκε μία βάση δεδομένων με το όνομα **myhmi** όπως φαίνεται στην εικόνα που ακολουθεί.

## Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)

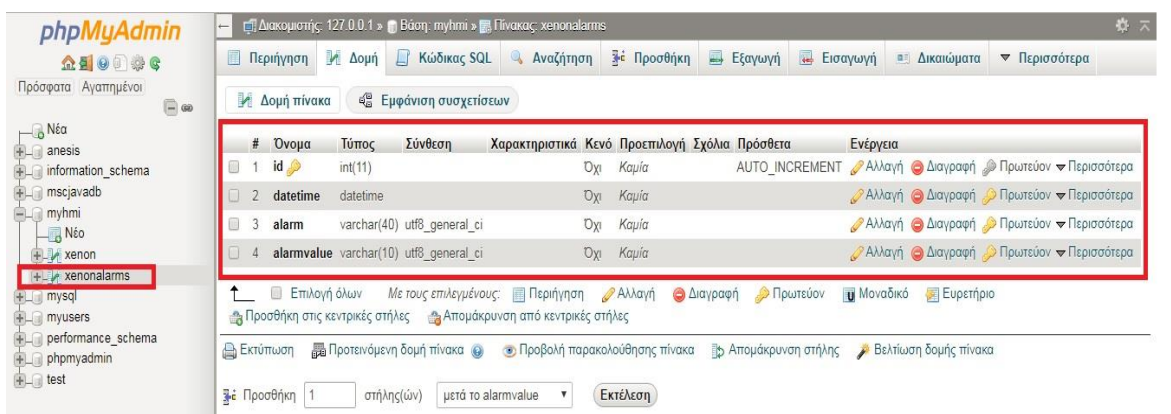


Εικόνα 75: Η βάση δεδομένων myhmi

Επόμενο βήμα ήταν να δημιουργήσουμε τους δύο πίνακες για τις εγγραφές των τιμών καθώς και των συμβάντων-συναγερμών. Στις εικόνες που ακολουθούν παρουσιάζονται οι πίνακες με τις αντίστοιχες δομές τους.



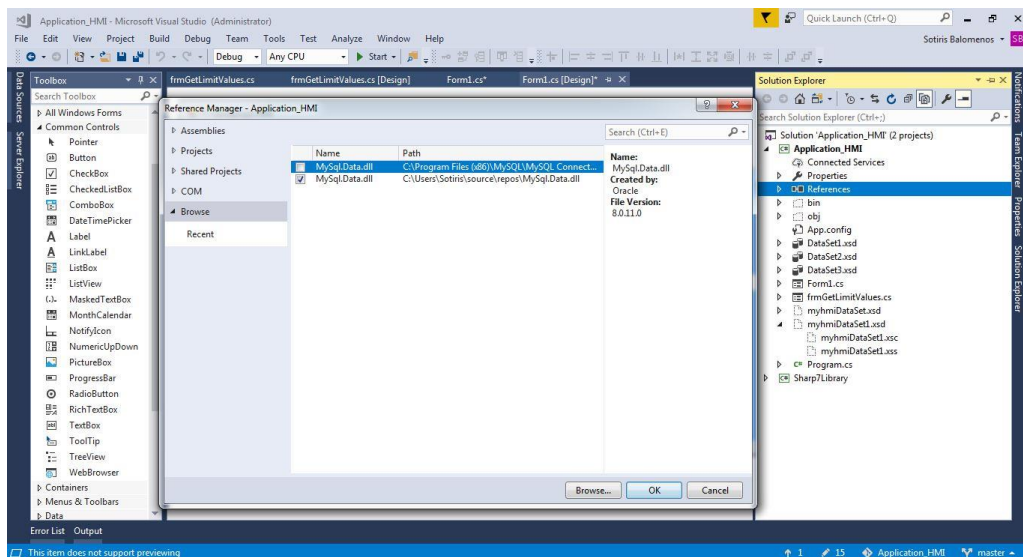
Εικόνα 76: Ο πίνακας xepon για την καταγραφή τιμών



Εικόνα 77: Ο πίνακας xeponalarm για την καταγραφή των συμβάντων

Για να μπορέσουμε να συνδεθούμε στην παραπάνω βάση δεδομένων, θα πρέπει να γραφτεί κώδικας στην εφαρμογή.

Εδώ θα πρέπει να έχει προηγηθεί η εγκατάσταση του connector (<https://dev.mysql.com/downloads/connector/net/>) για την σύνδεση .NET με την Βάση δεδομένων **MySQL**. Επιπλέον πρέπει να προστεθεί η αναφορά (reference) **MySql.Data.dll** όπως φαίνεται στην εικόνα που ακολουθεί.



**Εικόνα 78:** Πρόσθεση αναφοράς mysql.data.dll

Για την δημιουργία της σύνδεσης στη βάση από την πλευρά της εφαρμογής HMI πρέπει να δημιουργηθεί ο αντίστοιχος κώδικας. Εδώ χρησιμοποιούμε ένα **ConnectionString**. Ένα **ConnectionString** είναι ένα απλό string το οποίο έχει μία συγκεκριμένη φόρμα καθώς και πληροφορίες ώστε εντοπίσει την βάση δεδομένων και να συνδεθεί σε αυτή. Αφού συνδεθούμε στη βάση δεδομένων εκτελούμε το αντίστοιχο ερώτημα και τέλος κλείνουμε την σύνδεση.

Παρακάτω ακολουθεί μία εικόνα στην οποία παρουσιάζεται ο κώδικας για την σύνδεση και την εγγραφή των τιμών θερμοκρασίας και υγρασίας του οργάνου στην βάση δεδομένων myhmi.

```
public void InsertValuesDB()
{
    if (Client.Connected)
    {
        string connectionString = "datasource=127.0.0.1;port=3306;username=root;password=;database=myHMI;";

        string query = "INSERT INTO xenon (datetime,temperature,humidity) values (" +
            "'" + lbldatetime.Text + "','" + txtTemperature.Text + "','" + txtHumidity.Text + "')";

        // The data fields don't match for all dbs (access, mysql, sqlserver)

        MySqlConnection databaseConnection = new MySqlConnection(connectionString);
        MySqlCommand dbCommand = new MySqlCommand(query, databaseConnection);
        dbCommand.CommandTimeout = 60;

        try
        {
            databaseConnection.Open();
            int noRecs = dbCommand.ExecuteNonQuery();
            // MessageBox.Show("noRecs = " + noRecs.ToString());

            databaseConnection.Close();
        }
        catch (Exception ex)
        {
            // Show any error message.
            MessageBox.Show(ex.Message);
        }
    }
    else
    {
        MessageBox.Show("Δεν υπάρχει σύνδεση με το PLC");
    }
}
```

**Εικόνα 79:** Ο κώδικας για την σύνδεση στη βάση myHMI



## ΚΕΦΑΛΑΙΟ 5

### ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΡΓΑΣΙΕΣ ΓΙΑ ΤΗΝ ΕΦΑΡΜΟΓΗ

#### 5.1 Γενικά

Στο παρακάτω κεφάλαιο παρουσιάζονται τα συμπεράσματα από την σχεδίαση και ανάπτυξη του συστήματος. Επιπλέον γίνεται αναφορά σε μελλοντικές εργασίες που θα μπορούσαν να γίνουν στο σύστημα αυτοματισμού.

#### 5.2 Συμπεράσματα

Από την μελέτη και την ανάπτυξη της εφαρμογής συμπεραίνουμε ότι το σύστημα αυτοματισμού καλύπτει τις προδιαγραφές που είχαμε αναφέρει αρχικά σαν αντικείμενο της διπλωματικής εργασίας. Το σύστημα που έχουμε υλοποιήσει είναι ένα σύστημα απομακρυσμένου ελέγχου το οποίο παρακολουθεί κρίσιμους παραμέτρους για την εκτέλεση μίας παραγωγικής διεργασίας όπως είναι η θερμοκρασία και η υγρασία. Η εποπτεία ελέγχου και συλλογή δεδομένων καλύπτεται μέσα από μία εφαρμογή HMI σε γλώσσα προγραμματισμού C#, η οποία έχει αναπτυχθεί σε περιβάλλον visual studio. Το λογισμικό HMI, όπως το έχουμε ονομάσει έχει απευθείας επικοινωνία με έναν ελεγκτή PLC από τον οποίο συλλέγει διάφορες πληροφορίες της εφαρμογής όπως είναι η θερμοκρασία, υγρασία και κατάσταση των συναεργμών. Επιπλέον το λογισμικό HMI καταγράφει πληροφορίες αυτές σε μία βάση δεδομένων MySQL για μελλοντική επεξεργασία. Αξίζει να αναφερθεί ότι στο λογισμικό αυτό υπάρχει η δυνατότητα να προστεθούν και άλλοι ελεγκτές PLC και κατ' επέκταση να προστεθούν και άλλα όργανα ή διεργασίες προς έλεγχο. Αυτό σημαίνει ότι το σύστημα πάνω σε αυτή την δομή μπορεί να μεγαλώσει οπότε και από τον έλεγχο ενός οργάνου να μετατραπεί σε έλεγχο πολλών οργάνων τα οποία θα είναι συνδεδεμένα σε διαφορετικούς ελεγκτές PLC οι οποίοι ενδεχομένως να βρίσκονται εγκατεστημένοι και σε διαφορετικά σημεία.

#### 5.3 Μελλοντικές εργασίες

Στο σύστημα αυτοματισμού θα μπορούσαν να προστεθούν μελλοντικά νέες δυνατότητες με στόχο τον καλύτερο έλεγχο από τον χειριστή. Για παράδειγμα θα



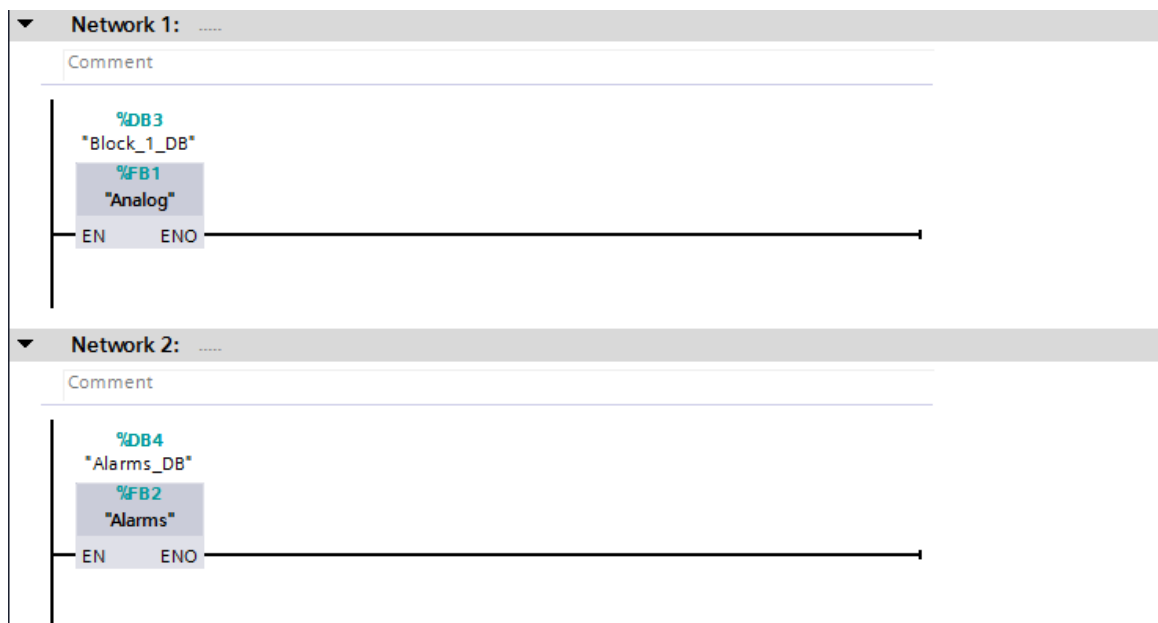
μπορούσε στο σύστημα όταν υπάρχει κάποιος συναγερμός να ενημερώνει τον χρήστη και με email, οπότε ακόμα και αν ο χρήστης βρισκόταν εκτός του γραφείου να μπορούσε να ενημερωθεί. Αυτό το σενάριο θα μπορούσε να υλοποιηθεί με την δημιουργία του κατάλληλου κώδικα είτε από την πλευρά του ελεγκτή είτε από την πλευρά του ανοικτού λογισμικού HMI. Μία αντίστοιχη δυνατότητα είναι η ειδοποίηση μέσω SMS από δίκτυο GSM. Για να υλοποιηθεί το σενάριο αυτό θα πρέπει να υπάρχει GSM/GPRS modem το οποίο να είναι συνδεδεμένο με τον ελεγκτή PLC και τον κατάλληλο κώδικα ώστε να υλοποιείται το σενάριο αυτό.

## ΠΑΡΑΡΤΗΜΑ Α΄

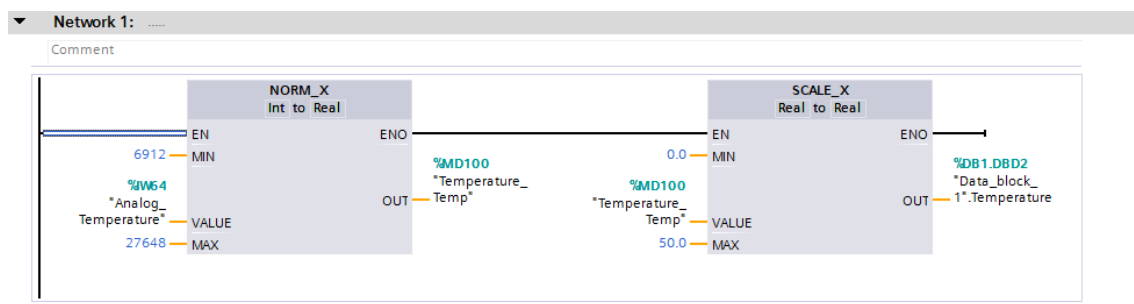
Στο παράρτημα αυτό παρατίθεται ο κώδικας ανάπτυξης της παρούσας εφαρμογής.

### 1. Ο Κώδικας του προγράμματος PLC

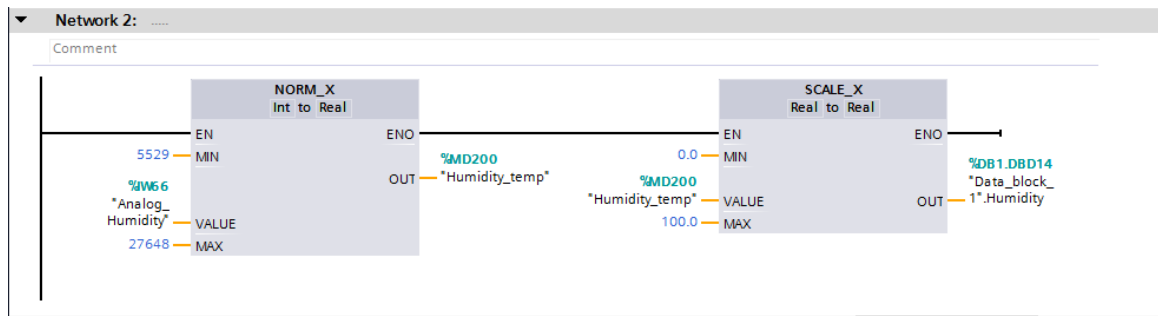
#### ➤ Main [OB1]



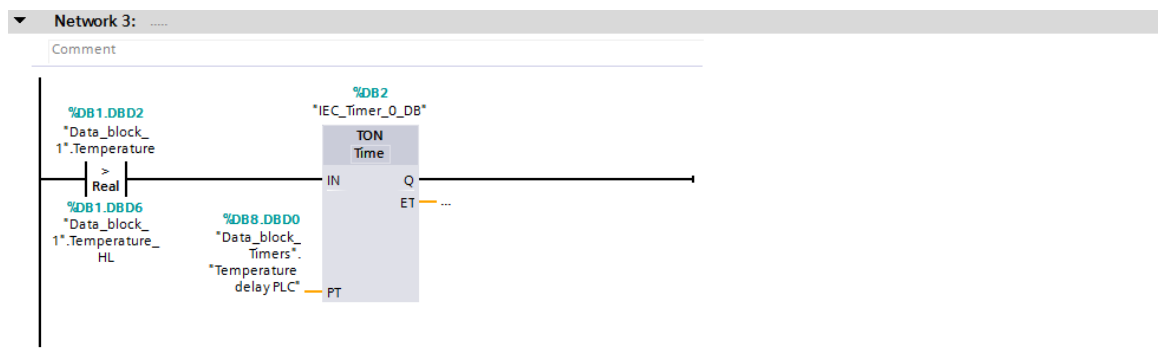
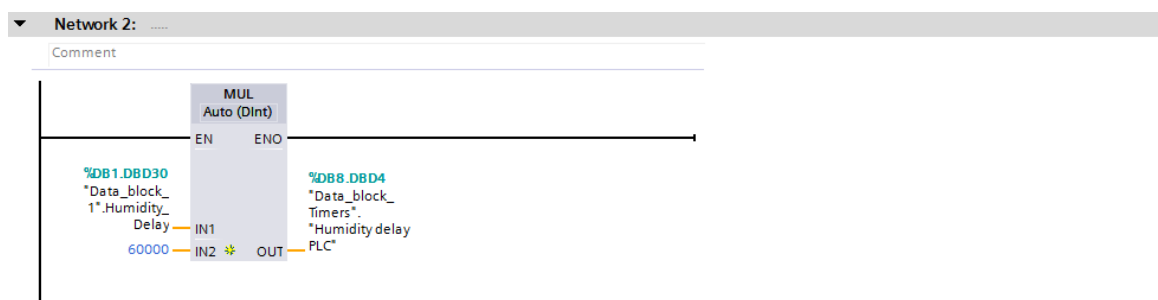
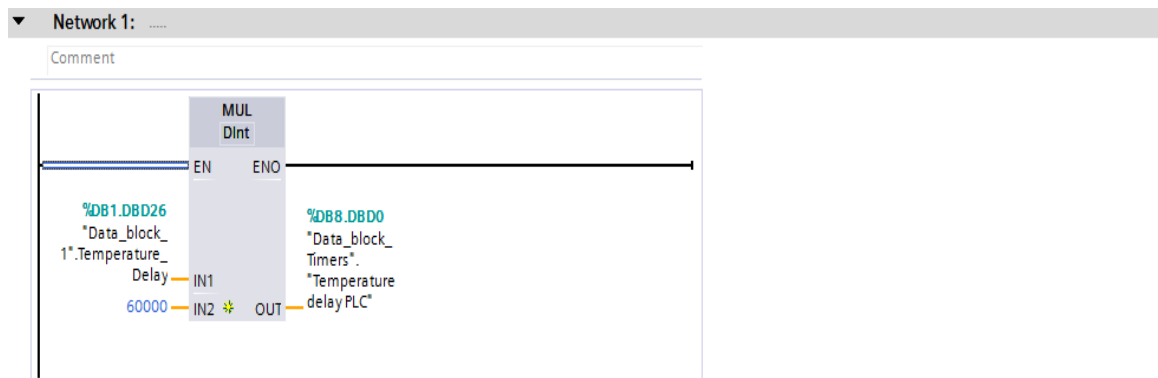
#### ➤ Analog [FB1]



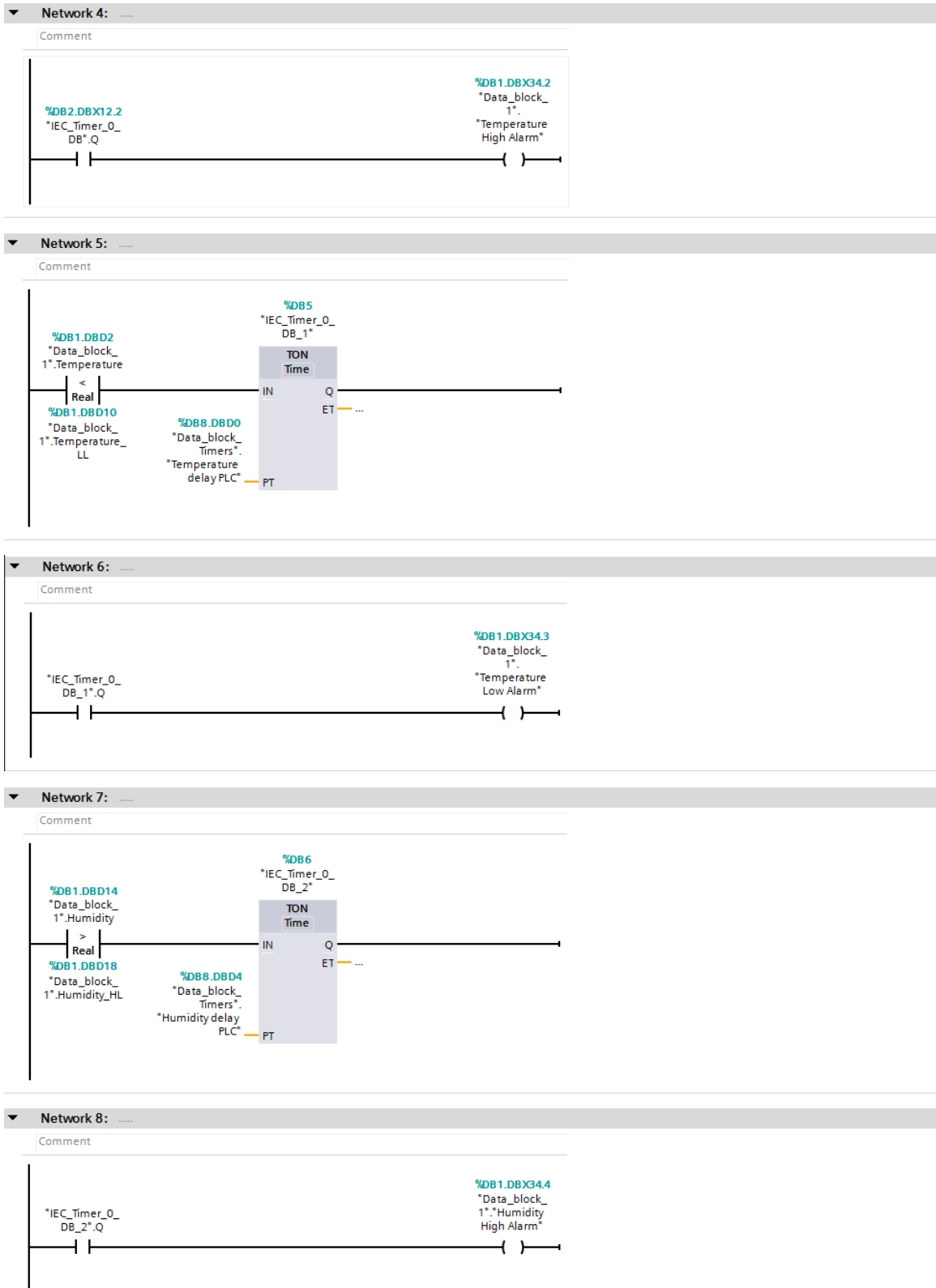
## Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)



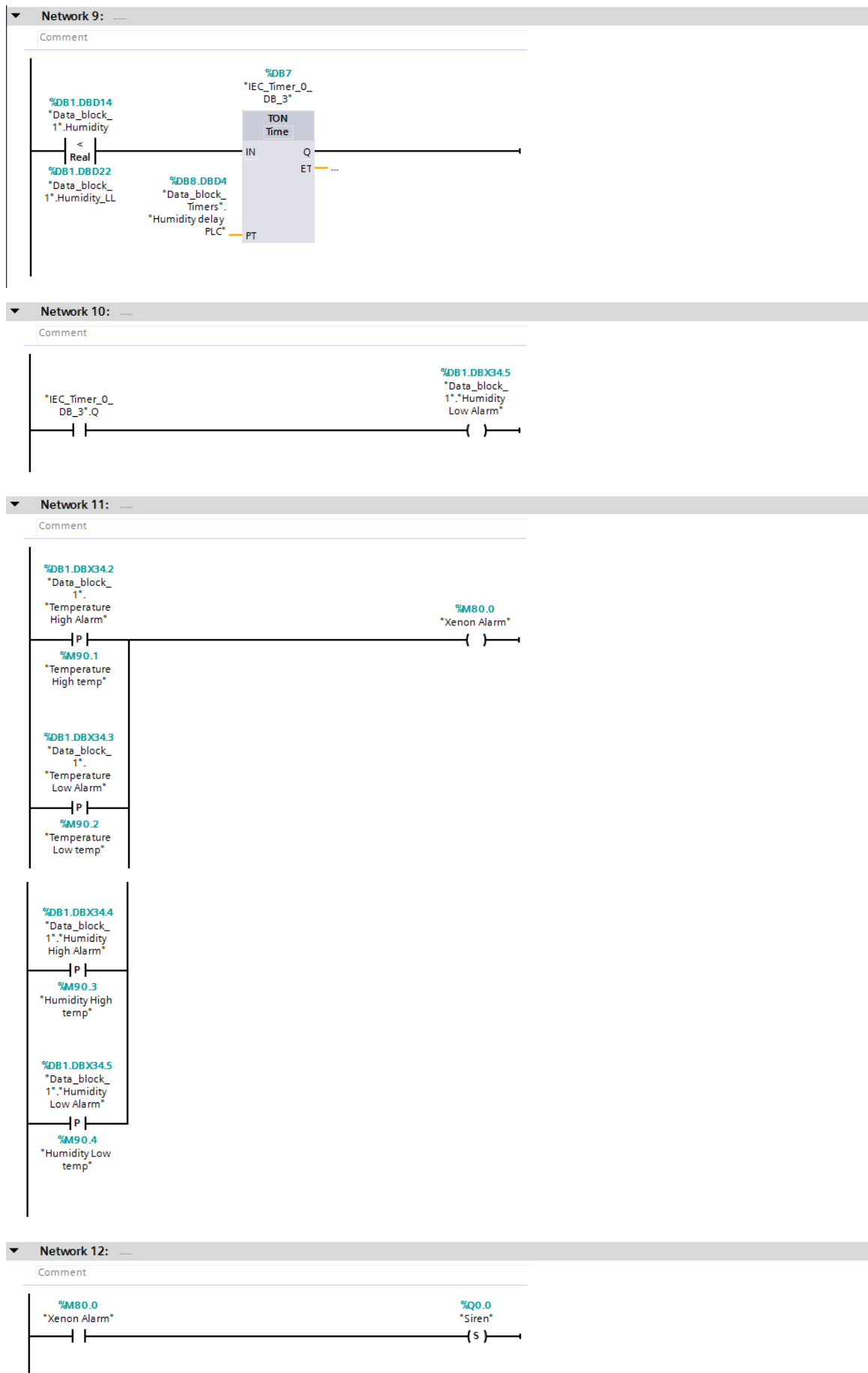
### ➤ Alarms [FB2]



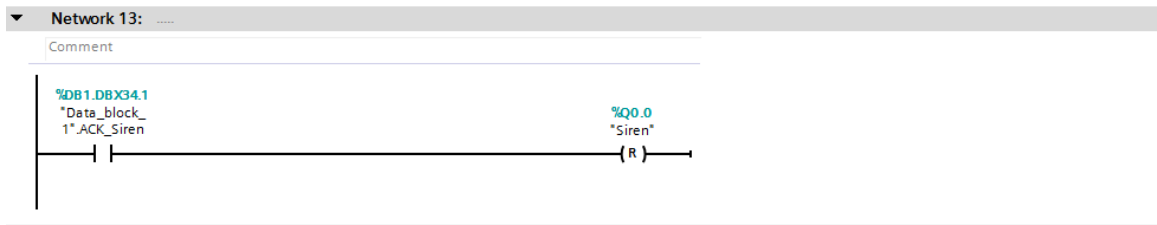
## Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)



## Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)



## Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)



### ➤ Data Block 1

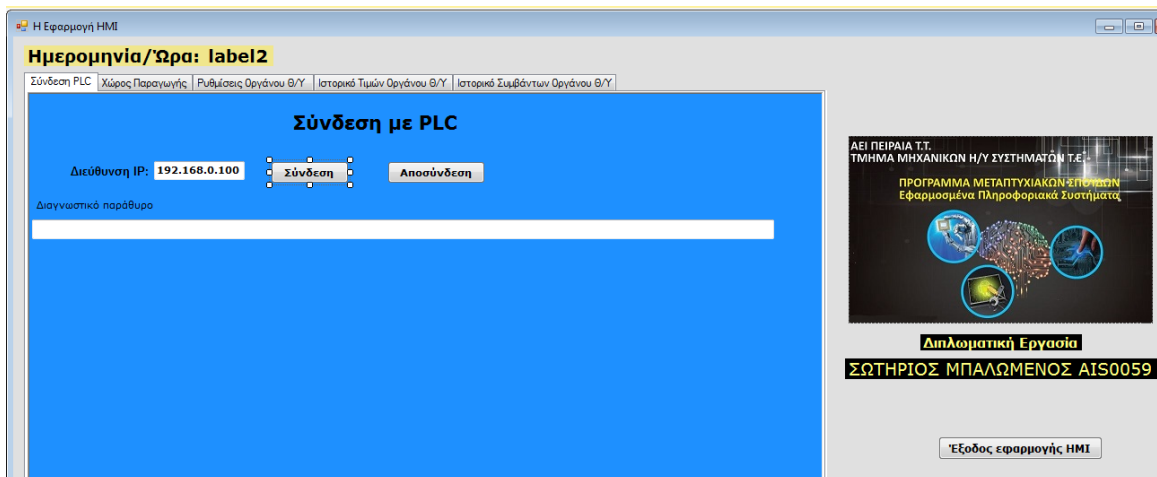
Data_block_1										
	Name	Data type	Offset	Default value	Start value	Retain	Accessible f...	Visible in ...	Setpoint	Comment
1	Static									
2	Timer ON	Bool	0.0	false	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	Temperature	Real	2.0	0.0	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	Temperature_HL	Real	6.0	0.0	25.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	Temperature_LL	Real	10.0	0.0	15.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	Humidity	Real	14.0	0.0	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	Humidity_HL	Real	18.0	0.0	60.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8	Humidity_LL	Real	22.0	0.0	30.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
9	Temperature_Delay	DInt	26.0	0	1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
10	Humidity_Delay	DInt	30.0	0	1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
11	Alarms	Bool	34.0	false	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
12	ACK_Siren	Bool	34.1	false	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
13	Temperature High Alarm	Bool	34.2	false	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
14	Temperature Low Alarm	Bool	34.3	false	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
15	Humidity High Alarm	Bool	34.4	false	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
16	Humidity Low Alarm	Bool	34.5	false	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

### ➤ Data Block Timers

Data_block_Timers										
	Name	Data type	Offset	Default value	Start value	Retain	Accessible f...	Visible in ...	Setpoint	Comment
1	Static									
2	Temperature delay PLC	DInt	0.0	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	Humidity delay PLC	DInt	4.0	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

## 2. Ο Κώδικας της εφαρμογής HMI

### ➤ Form1.cs [Design]



➤ **Form1.cs**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Diagnostics;
using System.Threading.Tasks;
using System.Windows.Forms;
using Sharp7;
using MySql.Data.MySqlClient;
using System.IO;

namespace Application_HMI
{
    public partial class Form1 : Form
    {
        private S7Client Client;
        public bool AlarmPreviousState34_2 = false;
        public bool AlarmNewState34_2 = false;
        public bool AlarmPreviousState34_3 = false;
        public bool AlarmNewState34_3 = false;
        public bool AlarmPreviousState34_4 = false;
        public bool AlarmNewState34_4 = false;
        public bool AlarmPreviousState34_5 = false;
        public bool AlarmNewState34_5 = false;

        private void ShowResult(int Result)
        {
            // This function returns a textual explanation of the error code
            txtError.Text = Client.ErrorText(Result);
            if (Result == 0)
                txtError.Text = txtError.Text + " (" + Client.ExecutionTime.ToString() + " ms)";
        }
    }
}
```

```
public Form1()
{
    InitializeComponent();
    Client = new S7Client();
    if (IntPtr.Size == 4)
        this.Text = this.Text + " - Running 32 bit Code";
    else
        this.Text = this.Text + " - Running 64 bit Code";
}

private void Connectbtn_Click(object sender, EventArgs e)
{
    int Result;
    Result = Client.ConnectTo(txtIP.Text, 0, 0);
    ShowResult(Result);

    if (Result == 0)
    {
        txtError.Text = txtError.Text + ", To PLC συνδέθηκε: " + Client.Connected;
        txtError.Text = txtError.Text + ", PDU Negotiated : " +
Client.PduSizeNegotiated.ToString();
        txtIP.Enabled = false;
        Connectbtn.Enabled = false;
        Disconnectbtn.Enabled = true;
    }
}

private void Disconnectbtn_Click(object sender, EventArgs e)
{
    Client.Disconnect();
    txtError.Text = "To PLC Αποσυνδέθηκε";
    txtIP.Enabled = true;
    Connectbtn.Enabled = true;
    Disconnectbtn.Enabled = false;
}

public void ReadDB1()
{
```



```
if (Client.Connected)
{
    byte[] db1Buffer = new byte[35];
    int Result;
    Result = Client.DBRead(1, 0, 35, db1Buffer);
    if (Result != 0)
    {
        Console.WriteLine("Error: " + Client.ErrorText(Result));
    }
    ShowResult(Result);
    float db1dbd2 = S7.GetRealAt(db1Buffer, 2);
    float db1dbd6 = S7.GetRealAt(db1Buffer, 6);
    float db1dbd10 = S7.GetRealAt(db1Buffer, 10);
    float db1dbd14 = S7.GetRealAt(db1Buffer, 14);
    float db1dbd18 = S7.GetRealAt(db1Buffer, 18);
    float db1dbd22 = S7.GetRealAt(db1Buffer, 22);
    Int32 db1dbd26 = S7.GetDIntAt(db1Buffer, 26);
    Int32 db1dbd30 = S7.GetDIntAt(db1Buffer, 30);

    bool db1dbd34_0 = S7.GetBitAt(db1Buffer, 34, 0);
    bool db1dbd34_2 = S7.GetBitAt(db1Buffer, 34, 2);
    bool db1dbd34_3 = S7.GetBitAt(db1Buffer, 34, 3);
    bool db1dbd34_4 = S7.GetBitAt(db1Buffer, 34, 4);
    bool db1dbd34_5 = S7.GetBitAt(db1Buffer, 34, 5);

    txtTemperature.Text = db1dbd2.ToString();
    txtHumidity.Text = db1dbd14.ToString();

    if ((db1dbd6 != 0) && (db1dbd10 != 0))
    {
        txtTempHigh.Text = db1dbd6.ToString();
        txtTempLow.Text = db1dbd10.ToString();
    }
    if ((db1dbd18 != 0) && (db1dbd22 != 0))
    {
        txtHumiHigh.Text = db1dbd18.ToString();
        txtHumiLow.Text = db1dbd22.ToString();
    }
    if ((db1dbd26 != 0) && (db1dbd30 != 0))
    {
```

## Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)

```
txtTempDelay.Text = db1dbd26.ToString();
txtHumiDelay.Text = db1dbd30.ToString();
}

if (db1dbd34_2)
{
    if (AlarmPreviousState34_2 == false)
    {
        txtAlarms.Text ="Προσοχή Υψηλή θερμοκρασία στο χώρο!";
        this.txtAlarms.ForeColor = System.Drawing.Color.Red;
        InsertAlarmsT_DB();
        AlarmNewState34_2 = true;
        AlarmPreviousState34_2 = AlarmNewState34_2;
    }
}
else if (!db1dbd34_2)
{
    AlarmPreviousState34_2 = false;
}

if (db1dbd34_3)
{
    if (AlarmPreviousState34_3 == false)
    {
        txtAlarms.Text ="Προσοχή Χαμηλή θερμοκρασία στο χώρο!";
        this.txtAlarms.ForeColor = System.Drawing.Color.Red;
        InsertAlarmsT_DB();
        AlarmNewState34_3 = true;
        AlarmPreviousState34_3 = AlarmNewState34_3;
    }
}
else if (!db1dbd34_3)
{
    AlarmPreviousState34_3 = false;
}

if (db1dbd34_4)
{
    if (AlarmPreviousState34_4 == false)
    {
        txtAlarms.Text ="Προσοχή Υψηλή υγρασία στο χώρο!";
```

## Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)

```
        this.txtAlarms.ForeColor = System.Drawing.Color.Red;
        InsertAlarmsRH_DB();
        AlarmNewState34_4 = true;
        AlarmPreviousState34_4 = AlarmNewState34_4;
    }
}
else if (!db1dbd34_4)
{
    AlarmPreviousState34_4 = false;
}
if (db1dbd34_5)
{
    if (AlarmPreviousState34_5 == false)
    {
        txtAlarms.Text = "Προσοχή Χαμηλή υγρασία στο χώρο!";
        this.txtAlarms.ForeColor = System.Drawing.Color.Red;
        InsertAlarmsRH_DB();
        AlarmNewState34_5 = true;
        AlarmPreviousState34_5 = AlarmNewState34_5;
    }
}
else if (!db1dbd34_5)
{
    AlarmPreviousState34_5 = false;
}
if (Client.Connected)
{
    if ((db1dbd34_2) || (db1dbd34_3) || (db1dbd34_4) || (db1dbd34_5))
    {
        lblAlarm.Text = "ΠΡΟΣΟΧΗ ALARM!!!";
        this.lblAlarm.ForeColor = System.Drawing.Color.Red;
    }
    else
    {
        lblAlarm.Text = "ΔΕΝ ΥΠΑΡΧΕΙ ALARM";
        this.lblAlarm.ForeColor = System.Drawing.Color.Blue;
    }
}
}
```

```
public void WriteDB1()
{
    byte[] db1Buffer = new byte[30];
    const int START_INDEX = 6;

    float TempHigh = Convert.ToSingle(txtTempHigh.Text);
    float TempLow = Convert.ToSingle(txtTempLow.Text);
    float HumiHigh = Convert.ToSingle(txtHumiHigh.Text);
    float HumiLow = Convert.ToSingle(txtHumiLow.Text);
    Int32 TempDelay = Convert.ToInt32(txtTempDelay.Text);
    Int32 HumiDelay = Convert.ToInt32(txtHumiDelay.Text);

    S7.SetRealAt(db1Buffer, 6 - START_INDEX, TempHigh);
    S7.SetRealAt(db1Buffer, 10 - START_INDEX, TempLow);
    S7.SetRealAt(db1Buffer, 18 - START_INDEX, HumiHigh);
    S7.SetRealAt(db1Buffer, 22 - START_INDEX, HumiLow);
    S7.SetDIntAt(db1Buffer, 26 - START_INDEX, TempDelay);
    S7.SetDIntAt(db1Buffer, 30 - START_INDEX, HumiDelay);

    int Result;

    Result = Client.DBWrite(1, START_INDEX, db1Buffer.Length, db1Buffer);

    if (Result != 0)
    {
        Console.WriteLine("Error: " + Client.ErrorText(Result));
    }
}

private void btnExit_Click(object sender, EventArgs e)
{
    this.Close();
}

private void timer1_Tick(object sender, EventArgs e)
{
    if (Client.Connected)
    {
```

```
        ReadDB1();
    }
}

private void button1_Click(object sender, EventArgs e)
{
    if (Client.Connected)
    {
        frmGetLimitValues x = new frmGetLimitValues(txtTempHigh.Text, txtTempLow.Text,
txtTempDelay.Text, txtHumiHigh.Text, txtHumiLow.Text, txtHumiDelay.Text);

        x.ShowDialog();
        if (x.isCancel == false)
        {
            float T_high, T_low, Rh_high, Rh_low;
            Int32 T_delay, Rh_delay;

            try
            {
                T_high = float.Parse(x.tmpHigh);
                T_low = float.Parse(x.tmpLow);
                T_delay = Int32.Parse(x.tmpTime);

                txtTempHigh.Text = x.tmpHigh;
                txtTempLow.Text = x.tmpLow;
                txtTempDelay.Text = x.tmpTime;

                Rh_high = float.Parse(x.RhHigh);
                Rh_low = float.Parse(x.RhLow);
                Rh_delay = Int32.Parse(x.RhTime);
                txtHumiHigh.Text = x.RhHigh;
                txtHumiLow.Text = x.RhLow;
                txtHumiDelay.Text = x.RhTime;

                WriteDB1();
            }
            catch (Exception ex) { MessageBox.Show("Error in values"); }
        }
    }
}
```

## Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)

```
    }
    else
    {
        MessageBox.Show("Δεν υπάρχει σύνδεση με το PLC");
    }
}

/*
 * Download connector https://dev.mysql.com/downloads/connector/net/6.9.html
 * Extract zip to a folder
 * Add a reference to dll named MySql.Data.dll founded in the above folder
 * Add namespace using MySql.Data.MySqlClient;
 *
 */

private void btnInsert_Click(object sender, EventArgs e)
{
    if (Client.Connected)
    {
        string connectionString =
"datasource=127.0.0.1;port=3306;username=root;password=;database=myHMI;";

        string query = "INSERT INTO xenon (datetime,temperature,humidity) values (" +
"" + lbldatetime.Text + ", " + txtTemperature.Text + ", " + txtHumidity.Text + ")";

        // The data fields don't match for all dbs (access, mysql, sqlserver)

        MySqlConnection databaseConnection = new MySqlConnection(connectionString);
        MySqlCommand dbCommand = new MySqlCommand(query, databaseConnection);
        dbCommand.CommandTimeout = 60;

        try
        {
            databaseConnection.Open();
            int noRecs = dbCommand.ExecuteNonQuery();
            MessageBox.Show("Εγγραφή no= " + noRecs.ToString());

            databaseConnection.Close();
        }
    }
}
```

```
        catch (Exception ex)
        {
            // Show any error message.
            MessageBox.Show(ex.Message);
        }
    }
else
{
    MessageBox.Show("Δεν υπάρχει σύνδεση με το PLC");
}
}

public void InsertValuesDB()
{
    if (Client.Connected)
    {
        string connectionString =
"datasource=127.0.0.1;port=3306;username=root;password=;database=myHMI;";

        string query = "INSERT INTO xenon (datetime,temperature,humidity) values (" +
"" + lbldatetime.Text + "," + txtTemperature.Text + "," + txtHumidity.Text + ")";

        // The data fields don't match for all dbs (access, mysql, sqlserver)

        MySqlConnection databaseConnection = new MySqlConnection(connectionString);
        MySqlCommand dbCommand = new MySqlCommand(query, databaseConnection);
        dbCommand.CommandTimeout = 60;

        try
        {
            databaseConnection.Open();
            int noRecs = dbCommand.ExecuteNonQuery();
            // MessageBox.Show("noRecs = " + noRecs.ToString());

            databaseConnection.Close();
        }
        catch (Exception ex)
        {
            // Show any error message.
            MessageBox.Show(ex.Message);
        }
    }
}
```

## Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματισμού με PLC και HMI (C#)

```
    }
}
else
{
    MessageBox.Show("Δεν υπάρχει σύνδεση με το PLC");
}
}

public void InsertAlarmsT_DB()
{
    string connectionString =
"datasource=127.0.0.1;port=3306;username=root;password=;database=myHMI;charset=utf8mb4";

    string query = "INSERT INTO xenonalarms (datetime,alarm,alarmvalue) values (" +
"" + lbldatetime.Text + "," + txtAlarms.Text + "," + txtTemperature.Text + ")";

    // The data fields don't match for all dbs (access, mysql, sqlserver)

    MySqlConnection databaseConnection = new MySqlConnection(connectionString);
    MySqlCommand dbCommand = new MySqlCommand(query, databaseConnection);
    dbCommand.CommandTimeout = 60;

    try
    {
        databaseConnection.Open();
        int noRecs = dbCommand.ExecuteNonQuery();
        // MessageBox.Show("noRecs = " + noRecs.ToString());
        databaseConnection.Close();
    }
    catch (Exception ex)
    {
        // Show any error message.
        MessageBox.Show(ex.Message);
    }
}

public void InsertAlarmsRH_DB()
{
    string connectionString =
"datasource=127.0.0.1;port=3306;username=root;password=;database=myHMI;charset=utf8mb4";
```



```
string query = "INSERT INTO xenonalarms (datetime,alarm,alarmvalue) values (" +
"" + lbldatetime.Text + ", " + txtAlarms.Text + ", " + txtHumidity.Text + ")";

// The data fields don't match for all dbs (access, mysql, sqlserver)

MySQLConnection databaseConnection = new MySQLConnection(connectionString);
MySQLCommand dbCommand = new MySQLCommand(query, databaseConnection);
dbCommand.CommandTimeout = 60;

try
{
    databaseConnection.Open();
    int noRecs = dbCommand.ExecuteNonQuery();
    // MessageBox.Show("noRecs = " + noRecs.ToString());

    databaseConnection.Close();
}
catch (Exception ex)
{
    // Show any error message.
    MessageBox.Show(ex.Message);
}
}

private void timer2_Tick(object sender, EventArgs e)
{
    DateTime time = DateTime.Now;
    this.lbldatetime.Text = time.ToString("yyyy-MM-dd HH:mm:ss");
}

private void timer3_Tick(object sender, EventArgs e)
{
    if (Client.Connected)
    {
        InsertValuesDB();
    }
}
}
```

```
private void tabPage1_Click(object sender, EventArgs e)
{

}

private void button1_Click_1(object sender, EventArgs e)
{
    if (Client.Connected)
    {

        byte[] db1Buffer = new byte[1];
        const int START_INDEX = 34;
        S7.SetBitAt(ref db1Buffer, 34 - START_INDEX, 1, true);
        int Result;
        Result = Client.DBWrite(1, START_INDEX, db1Buffer.Length, db1Buffer);
        if (Result != 0)
        {
            Console.WriteLine("Error: " + Client.ErrorText(Result));
        }
    }
    else
    {
        MessageBox.Show("Δεν υπάρχει σύνδεση με το PLC");
    }
}

private void btnDataGrid_Click(object sender, EventArgs e)
{
    MySqlConnection SQL_Connection = new
    MySqlConnection("datasource=127.0.0.1;PORT=3306;username=root;password=;Database=myH
    MI;");
    MySqlDataAdapter db_select;
    DataSet db_data;
    string query = "SELECT * FROM xenon where datetime between '" +
    dateTimePicker1.Value.ToString("yyyy-MM-dd HH:mm:ss") + "' and '" +
    dateTimePicker2.Value.ToString("yyyy-MM-dd HH:mm:ss") + "' ";

    try
```

```
{
    db_select = new MySqlDataAdapter(query, SQL_Connection);
    db_data = new System.Data.DataSet();
    db_select.Fill(db_data);
    dataGridView1.DataSource = db_data.Tables[0];
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message); // Show any error message.
}
}

private void btnAlarmHistory_Click(object sender, EventArgs e)
{
    MySqlConnection SQL_Connection = new
    MySqlConnection("datasource=127.0.0.1;PORT=3306;username=root;password=;Database=myH
    MI;");
    MySqlDataAdapter db_select;
    DataSet db_data;
    string query = "SELECT * FROM xenonalarms where datetime between '" +
    dateTimePicker3.Value.ToString("yyyy-MM-dd HH:mm:ss") + "' and '" +
    dateTimePicker4.Value.ToString("yyyy-MM-dd HH:mm:ss") + "' ";

    try
    {
        //db_select = new MySqlDataAdapter("SELECT * FROM xenon", SQL_Connection);
        db_select = new MySqlDataAdapter(query, SQL_Connection);
        db_data = new System.Data.DataSet();
        db_select.Fill(db_data);
        dataGridView2.DataSource = db_data.Tables[0];
        //dataGridView1.DataSource = db_data.Tables[1];
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message); // Show any error message.
    }
}

private void btnExport_Click(object sender, EventArgs e)
```

```

    {
        StreamWriter writer = new
StreamWriter(@"C:\Users\Sotiris\source\repos\Application_HMI\exports\hmiValues.txt");
        writer.WriteLine("-----
");
        writer.WriteLine("\t" + "ID" + "\t" + "|" + "\t" + "Ημερομηνία & Ώρα" + "\t" + "|" + "\t" +
"Θερμοκρασία" + "\t" + "|" + "\t" + "Υγρασία" + "\t" + "\t" + "|" + "\n");
        writer.WriteLine("-----
");
        for (int i = 0; i < dataGridView1.Rows.Count; i++)
        {
            for (int j = 0; j < dataGridView1.Columns.Count; j++)
            {
                writer.Write("\t" + dataGridView1.Rows[i].Cells[j].Value.ToString() + "\t" + "|");
            }
            writer.WriteLine("\r\n");
        }
        writer.WriteLine("-----
");
        writer.Close();
        MessageBox.Show("Εξαγωγή τιμών ολοκληρώθηκε");
    }

private void btnExportAlarms_Click(object sender, EventArgs e)
{
    StreamWriter writer = new
StreamWriter(@"C:\Users\Sotiris\source\repos\Application_HMI\exports\hmiAlarms.txt");
    writer.WriteLine("-----
-----");
    writer.WriteLine("\t" + "ID" + "\t" + "|" + "\t" + "Ημερομηνία & Ώρα" + "\t" + "|" + "\t" + "\t" +
"Συμβάντα" + "\t" + "\t" + "\t" + "|" + "\t" + "Τιμή οργάνου" + "\t" + "|" + "\n");
    writer.WriteLine("-----
-----");
    for (int i = 0; i < dataGridView2.Rows.Count; i++)
    {
        for (int j = 0; j < dataGridView2.Columns.Count; j++)
        {
            writer.Write("\t" + dataGridView2.Rows[i].Cells[j].Value.ToString() + "\t" + "|");
        }
    }
}

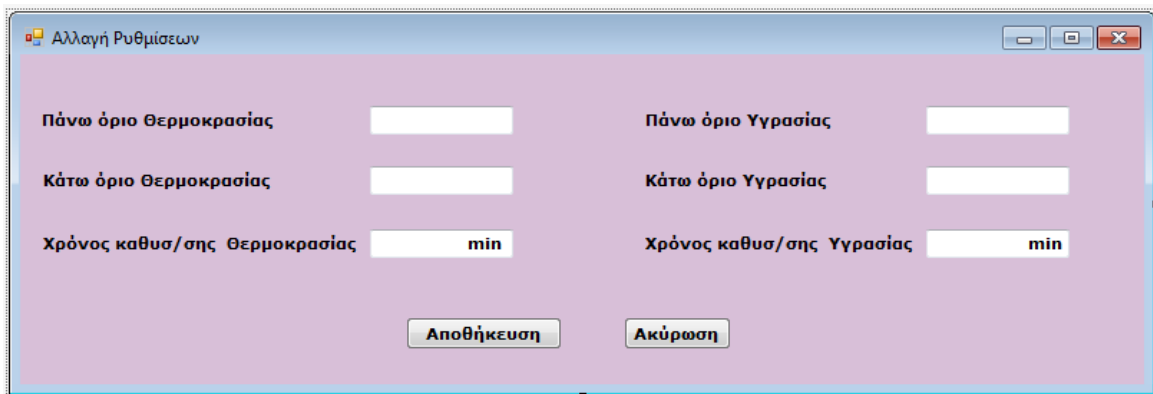
```

```

        writer.WriteLine("\r\n");
        //writer.WriteLine("-----");
    }
    writer.WriteLine("-----");
-----");
    writer.Close();
    MessageBox.Show("Εξαγωγή συμβάντων ολοκληρώθηκε");
}
}
}

```

➤ **frmGetLimitValues.cs[Design]**



➤ **frmGetLimitValues.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Application_HMI
{
    public partial class frmGetLimitValues : Form
    {
        public string tmpHigh, tmpLow, tmpTime, RhHigh, RhLow, RhTime ;
        public bool isCancel;

        private void btnCancel_Click(object sender, EventArgs e)
        {
            isCancel = true;
        }
    }
}

```

```
        this.Close();
    }

    public frmGetLimitValues(string tmpHigh, string tmpLow, string tmpTime,
string RhHigh, string RhLow, string RhTime)
    {
        InitializeComponent();
        isCancel = false;
        this.tmpHigh = tmpHigh;
        this.tmpLow = tmpLow;
        this.tmpTime = tmpTime;

        txtTmpHigh.Text = tmpHigh;
        txtTmpLow.Text = tmpLow;
        txtTmpTime.Text = tmpTime;

        this.RhHigh = RhHigh;
        this.RhLow = RhLow;
        this.RhTime = RhTime;

        txtRhHigh.Text = RhHigh;
        txtRhLow.Text = RhLow;
        txtRhTime.Text = RhTime;

    }

    private void btnSave_Click(object sender, EventArgs e)
    {
        this.tmpHigh = txtTmpHigh.Text;
        this.tmpLow = txtTmpLow.Text;
        this.tmpTime = txtTmpTime.Text;

        this.RhHigh = txtRhHigh.Text;
        this.RhLow = txtRhLow.Text;
        this.RhTime = txtRhTime.Text;

        this.Close();
    }
}
}
```



## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Α. Βελώνη, Σ. Αλατσαθιάνος, "Βιομηχανική Πληροφορική", Αυτοέκδοση, 2014.
- [2] «Snap7», [Ηλεκτρονικό]. Available: <http://snap7.sourceforge.net/>.
- [3] «Opc Foundation», [Ηλεκτρονικό]. Available: <https://opcfoundation.org/>.
- [4] «SIMATIC S7-1200, CPU 1214C», [Ηλεκτρονικό]. Available: <https://support.industry.siemens.com/cs/pd/481697?pdtd=td&dl=en&lc=en-GR/>.
- [5] «SCHNEIDER ELECTRIC XB5KSB», [Ηλεκτρονικό]. Available: <https://www.tme.eu/en/details/xb5ksb/panel-mount-switches-standard-22mm/schneider-electric/>.
- [6] «Totally Integrated Automation Portal», [Ηλεκτρονικό]. Available: <https://www.siemens.com/global/en/home/products/automation/industry-software/automation-software/tia-portal.html/>.
- [7] «brochure\_simatic-step7\_tia-portal\_en.pdf», [Ηλεκτρονικό]. Available: [https://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure\\_simatic-step7\\_tia-portal\\_en.pdf/](https://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure_simatic-step7_tia-portal_en.pdf/).
- [8] «How to write a Siemens S7 plc driver with C# and Sharp7», [Ηλεκτρονικό]. Available: <https://www.mesta-automation.com/how-to-write-a-siemens-s7-plc-driver-with-c-and-sharp7/>.
- [9] « Wikipedia: Microsoft\_Visual\_Studio», [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio/](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio/).
- [10] «Wikipedia: Microsoft Visual Studio», [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio/](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio/).
- [11] «Wikipedia: Ολοκληρωμένο περιβάλλον ανάπτυξης», [Ηλεκτρονικό]. Available: [https://el.wikipedia.org/wiki/%CE%9F%CE%BB%CE%BF%CE%BA%CE%BB%CE%B7%CF%81%CF%89%CE%BC%CE%AD%CE%BD%CE%BF\\_%CF%80%CE%B5%CF%81%CE%B9%CE%B2%CE%AC%CE%BB%CE%BB%CE%BF%CE%BD\\_%CE%B1%CE%BD%CE%AC%CF%80%CF%84%CF%85%CE%BE%CE%B7%CF%82/](https://el.wikipedia.org/wiki/%CE%9F%CE%BB%CE%BF%CE%BA%CE%BB%CE%B7%CF%81%CF%89%CE%BC%CE%AD%CE%BD%CE%BF_%CF%80%CE%B5%CF%81%CE%B9%CE%B2%CE%AC%CE%BB%CE%BB%CE%BF%CE%BD_%CE%B1%CE%BD%CE%AC%CF%80%CF%84%CF%85%CE%BE%CE%B7%CF%82/).
- [12] «S7-1200 Easy Book», [Ηλεκτρονικό]. Available: <https://support.industry.siemens.com/cs/document/39710145/simatic-s7-1200-easy-book?dti=0&lc=en-WW/>.
- [13] «MySQL 5.7 Reference Manual», [Ηλεκτρονικό]. Available: <https://downloads.mysql.com/docs/refman-5.7-en.a4.pdf/>.



- [14]** Σημειώσεις μαθήματος “Δομημένος προγραμματισμός”, Τμήμα Μηχανολόγων Μηχανικών Τ.Ε, ΤΕΙ ΠΕΙΡΑΙΑ, Δρ. Νικόλαος Ζ. Ζάχαρης.
- [15]** Σημειώσεις σεμιναρίου “Εισαγωγή στη #C και το .NET 4.0”, ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ- ΕΡΓΑΣΤΗΡΙΟ ΤΕΧΝΟΛΟΓΙΑΣ ΠΟΛΥΜΕΣΩΝ, Βασίλης Κόλιας.

