

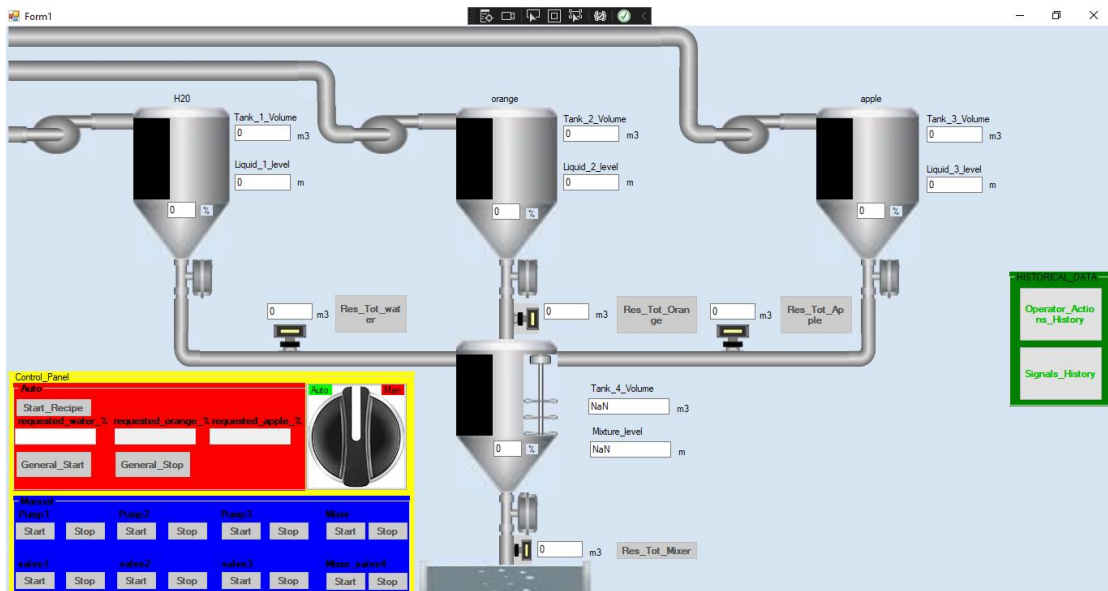


ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη εξατομικευμένου λογισμικού SCADA
με την χρήση αντικειμενοστραφούς γλώσσας
προγραμματισμού»



Του φοιτητή

Αθανάσιου Φάφκα

Αρ.Μητρώου:52118Μ

Επιβλέπων

Απόστολος Τσαγκάρης

Βαθμίδα: Αναπληρωτής Καθηγητής

Ημερομηνία 22/9/2023

Τίτλος Δ.Ε. Ανάπτυξη εξατομικευμένου λογισμικού SCADA με την χρήση
αντικειμενοστραφούς γλώσσας προγραμματισμού
Κωδικός Δ.Ε. 22258

Όνοματεπώνυμο φοιτητή Αθανάσιος Φάφκας
Όνοματεπώνυμο εισηγητή Απόστολος Τσαγκάρης

Ημερομηνία ανάληψης Δ.Ε. 26/03/2023

Ημερομηνία περάτωσης Δ.Ε. 22/09/2023

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Αθανάσιου Φάφκα που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Περίληψη

Η συγκεκριμένη διπλωματική πραγματεύεται την ανάπτυξη ενός εξατομικευμένου low cost SCADA συστήματος με τη χρήση σύγχρονου προγραμματιστικού περιβάλλοντος. Στο πρώτο κεφάλαιο, γίνεται μια εισαγωγή στην λειτουργία των συστημάτων SCADA και στην αρχιτεκτονική που παρουσιάζουν ενώ πραγματοποιείται και μια αναφορά στα είδη των custom scada που χρησιμοποιούνται αυτήν τη στιγμή, όπως επίσης και σε εφαρμογές που έχουν πραγματοποιηθεί. Στο δεύτερο κεφάλαιο, πραγματοποιείται μια ανάλυση της εφαρμογής που σχεδιάστηκε παρουσιάζοντας ουσιαστικά την λειτουργία που θα πρέπει να επιτελεί. Πιο συγκεκριμένα, το SCADA της εφαρμογής θα πρέπει να εποπτεύει και να ελέγχει την πραγματοποίηση συνταγών σε αυτόματη και χειροκίνητη λειτουργία ενώ παράλληλα θα επιτρέπει και την πλήρη διαχείριση ιστορικών στοιχείων τόσο σημάτων όσο και ενεργειών του χρήστη. Στο τρίτο κεφάλαιο, αναλύεται ο προγραμματισμός του plc για τον οποίο χρησιμοποιήθηκε η γλώσσα ladder και το πρόγραμμα της εταιρείας Siemens TIA Portal. Στο τέταρτο κεφάλαιο, παρουσιάζεται αναλυτικά ο προγραμματισμός του SCADA με την βοήθεια της αντικειμενοστραφούς γλώσσας προγραμματισμού C# μέσα από το περιβάλλον του Visual Studio σε συνδυασμό με την βιβλιοθήκη S7.Net που είναι κατάλληλη για τέτοιου είδους εφαρμογές. Επιπλέον, στο ίδιο κεφάλαιο αναλύεται η επικοινωνία του SCADA με μια SQL βάση δεδομένων στην οποία αποθηκεύονται όλα τα δεδομένα της εφαρμογής με την δυνατότητα διαχειρισής τους από τον χρήστη με την μορφή προβολής ιστορικών στοιχείων. Στο πέμπτο κεφάλαιο περιγράφονται τα αποτελέσματα των προσομοιώσεων που πραγματοποιήθηκαν ενώ στο τελευταίο κεφάλαιο επισημαίνονται ορισμένα συμπεράσματα που εξάχθηκαν από την όλη εργασία.

«Custom SCADA Software development using an object-oriented programming language»

«Athanasios Fafkas»

Abstract

This dissertation deals with the development of a personalized low cost SCADA system using a modern programming environment. In the first chapter, an introduction is made about the operation of SCADA systems and the architecture they present, while a reference is also made about the types of custom SCADA used at this moment, as well as in applications that have been carried out. In the second chapter, an analysis of the designed application is carried out, essentially presenting the function it should perform. More specifically, the SCADA of the application should supervise and control the execution of recipes in automatic and manual mode while also allowing the complete management of historical data of both signals and user actions. In the third chapter, the programming of the plc is analyzed for which the ladder language and the Siemens TIA Portal program were used. In the fourth chapter, SCADA programming is presented in detail with the help of the object-oriented programming language C# through the Visual Studio environment in combination with the S7.Net library which is suitable for such applications. In addition, in the same chapter, the communication of SCADA with a SQL database is analyzed in which all the data of the application are stored with the possibility of managing them by the user in the form of viewing historical data. In the fifth chapter the results of the simulations carried out are described, while in the last chapter, some conclusions drawn from the entire work are highlighted.

Περιεχόμενα

Περίληψη.....	iv
Abstract	v
Περιεχόμενα.....	vi
Κεφάλαιο 1ο : Εισαγωγή στα custom SCADA	1
1.1 Εισαγωγή στα SCADA	1
1.2 Αρχιτεκτονική συστήματος SCADA	2
1.3 Είδη Custom SCADA	3
1.4 Custom SCADA Εφαρμογές	5
Κεφάλαιο 2ο: Ανάλυση της εφαρμογής.....	9
2.1 Εισαγωγή	9
2.2 Ανάλυση της εφαρμογής.....	9
Κεφάλαιο 3ο: Προγραμματισμός του PLC.....	12
3.1 Περιγραφή των function block και του device configuration.....	12
3.2 Διαδικασία πλήρωσης δεξαμενής νερού.....	18
3.3 Διαδικασία εκροής και βαθμονόμησης εισόδου δεξαμενής νερού	20
3.4 Χειροκίνητη απαίτηση % ποσότητας νερού.....	23
3.5 Διαδικασία πλήρωσης δεξαμενής αναδευτήρα.....	24
3.6 Διαδικασία εκροής μείγματος και βαθμονόμησης της εισόδου της δεξαμενής του αναδευτήρα	26
3.7 Υπολογισμός βήματος για την καταμέτρηση της αθροιστικής ροής στην δεξαμενή του αναδευτήρα.....	29
3.8 Υπολογισμός της πυκνότητας του μείγματος	30
3.9 Προσβασιμότητα του PLC από το SCADA.....	31
Κεφάλαιο 4ο: Προγραμματισμός του SCADA	32
4.1 Εγκατάσταση βιβλιοθηκών S7.Net και Microsoft.Data.SqlClient	32
4.2 Δημιουργία αρχείου xaml	34
4.3 Δημιουργία tables στην SQL βάση δεδομένων.....	37
4.4 Ανάλυση κώδικα custom SCADA.....	39
Κεφάλαιο 5ο: Αποτελέσματα Προσομοιώσεων	51
5.1 Προβολή ιστορικών στοιχείων	51
5.2 Παραγωγή χυμού πορτοκαλιού κατά την αυτόματη λειτουργία	53
5.3 Παραγωγή ανάμεικτου χυμού με αναλογίες που ορίζει ο χειριστής στην χειροκίνητη λειτουργία.....	57
Κεφάλαιο 6ο: Συμπεράσματα	60
Βιβλιογραφία.....	61

ΠΑΡΑΡΤΗΜΑ Α : ΚΩΔΙΚΑΣ LADDER.....	63
ΠΑΡΑΡΤΗΜΑ Β : ΚΩΔΙΚΑΣ CUSTOM SCADA.....	90

Κεφάλαιο 1ο : Εισαγωγή στα custom SCADA

1.1 Εισαγωγή στα SCADA

Ο όρος SCADA είναι ακρωνύμιο του όρου Supervisory Control and Data Acquisition που ερμηνεύεται ως <<Εποπτικός Έλεγχος και Συλλογή Δεδομένων>>. Τα συστήματα SCADA χρησιμοποιούνται για την επιτήρηση και τον έλεγχο μιας εγκατάστασης ή του εξοπλισμού μιας βιομηχανίας όπως για παράδειγμα τον έλεγχο νερού και αποβλήτων, τη διύλιση και μεταφορά φυσικού αερίου και πετρελαίου, την ενέργεια κ.α. Ένα τέτοιου είδους σύστημα παρέχει την δυνατότητα ελέγχου και εποπτείας διεργασιών οι οποίες μπορεί να βρίσκονται κατανεμημένες μεταξύ διαφόρων σημείων σε απομακρυσμένες αποστάσεις. Πιο συγκεκριμένα, μπορούμε να πούμε ότι δεν αναφέρεται σε μία καθορισμένη τεχνολογία αλλά σε ένα μεγάλο πακέτο λογισμικού που έχει την δυνατότητα συλλογής όλων των απαραίτητων πληροφοριών, με την βοήθεια ενός δικτύου ευρείας κλίμακας, από το σύστημα που ελέγχεται ώστε να του επιτρέψει να πραγματοποιήσει μία σειρά από διαδικασίες όπως για παράδειγμα λήψη και αποθήκευση δεδομένων, ενημέρωση και ειδοποίηση καταστάσεων υψηλής σημασίας αλλά και καταστάσεων κινδύνου, επεξεργασία και διαχείριση δεδομένων, ανάπτυξη γραφημάτων, προβολή και παρουσίαση τιμών, λεπτομερής γραφική απεικόνιση της παραγωγικής διαδικασίας με κινούμενη εξομοίωση των μηχανών, επιτήρηση της κατάστασης του δικτύου της βιομηχανίας κ.α. Με το σύστημα SCADA μπορούμε να πούμε ουσιαστικά ότι ένα ολόκληρο εργοστάσιο εμφανίζεται μπροστά στην οθόνη μας.[12]

Η λειτουργία ενός συστήματος SCADA προϋποθέτει την συγκέντρωση πληροφοριών, την αποστολή των πληροφοριών αυτών σε ένα κεντρικό σημείο για την επεξεργασία, την ανάλυση και τον έλεγχο τους και στο τελευταίο στάδιο την παρουσίαση της πληροφορίας σε διάφορες οθόνες χειρισμού και εποπτείας (Human-Machine-Interface, HMI) τόσο σε πραγματικό χρόνο όσο και έπειτα από απαίτηση του χειριστή. Τα συστήματα αυτά παρουσιάζουν μεγάλη ποικιλία και μπορεί να είναι σχετικά απλά, όπως για παράδειγμα συστήματα επιτήρησης των περιβαλλοντικών συνθηκών ενός μικρού κτηρίου ή πολύ περίπλοκα, όπως για παράδειγμα συστήματα που παρακολουθούν πυρηνικές μονάδες παραγωγής ενέργειας. Επιπλέον, ένα σύστημα SCADA παρέχει την δυνατότητα ελέγχου και εποπτείας διαδικασιών που παρουσιάζουν μεγάλη τοπολογική διανομή. Για παράδειγμα όταν ένα σύστημα ελέγχου βρίσκεται διασκορπισμένο σε μεγάλη και απομακρυσμένη περιοχή, δεν απαιτείται να στέλνεται ανθρώπινο δυναμικό σε διάφορα σημεία για την πραγματοποίηση μετρήσεων ή ρυθμίσεων εφόσον η εποπτεία και ο έλεγχος ολόκληρου του συστήματος μπορούν να γίνουν από ένα μέρος γρήγορα, εύκολα και αποτελεσματικά. Στην εικόνα που ακολουθεί παρατηρούμε την παρακολούθηση του συστήματος SCADA από την αίθουσα ελέγχου.[12]

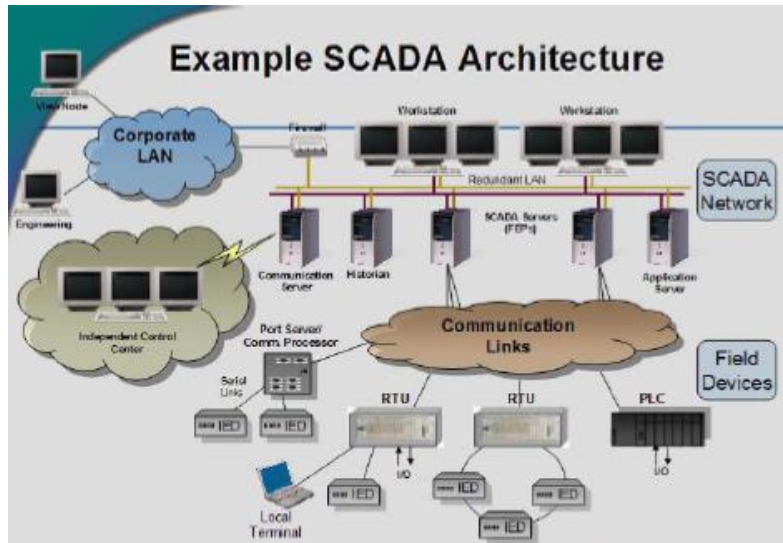


Εικόνα 1.1: Αίθουσα χειρισμού σταθμών SCADA σε βιομηχανία

1.2 Αρχιτεκτονική συστήματος SCADA

Ένα σύστημα SCADA πέρα από το λογισμικό ελέγχου και εποπτείας αποτελείται και από μια συλλογή αισθητήρων και ευφυών συσκευών. Πιο συγκεκριμένα, τα δομικά του στοιχεία μπορούμε να τα διαχωρίσουμε στις παρακάτω κατηγορίες:

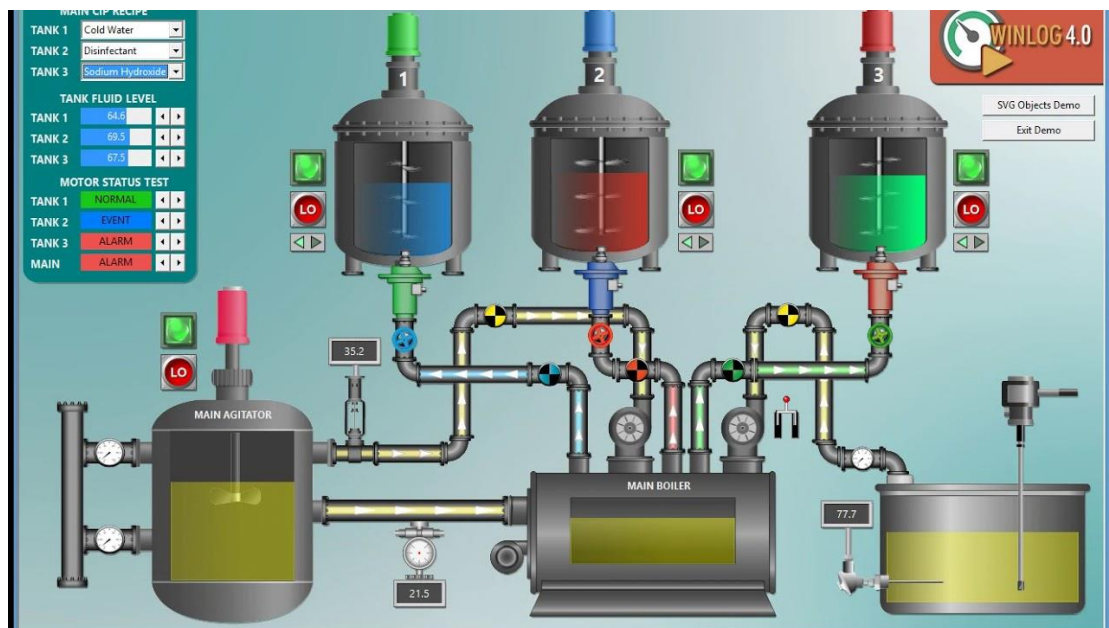
- Ένα δίκτυο έξυπνων συσκευών (Intelligent Electronic Devices-IED's) που συνδέεται με το σύστημα το οποίο επιθυμούμε να ελέγξουμε και να εποπτεύσουμε με την βοήθεια αισθητήρων και ελεγκτών. Το δίκτυο αυτό μας δίνει την δυνατότητα πραγματοποίησης μετρήσεων αλλά και ελέγχου συγκεκριμένων στοιχείων του συστήματος. [12]
- Ο Κεντρικός Υπολογιστικός Σταθμός (Master Terminal Unit-MTU) που απαρτίζεται από ένα κεντρικό υπολογιστικό σύστημα, μέσα στο οποίο υπάρχει εγκαταστημένο τόσο το λογισμικό SCADA όσο και το πρόγραμμα της εκάστοτε εφαρμογής. Ουσιαστικά μπορούμε να πούμε ότι αποτελεί τον βασικό πυρήνα του συστήματος SCADA. [12]
- Τερματικές μονάδες (Remote Telemetry Unit-RTU's) όπου συνδέονται στους διάφορους αισθητήρες και μετατρέπουν τα σήματα των αισθητήρων σε ψηφιακά δεδομένα σύμφωνα με το πρωτόκολλο επικοινωνίας που υποστηρίζουν. Επιπλέον, αποτελούν βασικό κομμάτι της τηλεμετρίας του εποπτικού συστήματος διότι έχουν την δυνατότητα να στέλνουν ψηφιακά δεδομένα στο Κεντρικό Εποπτικό Σύστημα καθώς και να λαμβάνουν ψηφιακές εντολές από το εποπτικό σύστημα. [12]
- Προγραμματιζόμενους λογικούς ελεγκτές (Programmable Logic Controller-PLC's) οι οποίοι παρόμοια με τις τερματικές μονάδες μετατρέπουν τα σήματα των αισθητήρων σε ψηφιακά δεδομένα. Αντίθετα με τα RTU, τα PLC's δεν παρέχουν δυνατότητες τηλεμετρίας, διαθέτουν όμως πιο εξελιγμένες ενσωματωμένες δυνατότητες ελέγχου καθώς υποστηρίζουν περισσότερους αλγόριθμους ελέγχου (IEC 61131-3). Επίσης, επειδή παρουσιάζουν μικρότερο όγκο είναι πιο οικονομικά. [12]
- Ένα σύστημα τηλεμετρίας για την διασύνδεση PLC και RTU με τα κέντρα ελέγχου και τις αποθήκες δεδομένων. Παρέχει διασύνδεση των ευφυών και περιφερειακών συσκευών με κέντρα ελέγχου και βάσεις δεδομένων με δύο διαφορετικούς τρόπους είτε ενσύρματα είτε ασύρματα. Για την πιο αποτελεσματική διασύνδεση των παραπάνω συσκευών χρησιμοποιούνται σύγχρονα βιομηχανικά πρωτόκολλα επικοινωνίας για την σύνδεση των υπηρεσιών λογισμικού. [12]
- Μία διεπαφή ανθρώπου-μηχανής (HMI). Πρόκειται για συσκευή η οποία είναι απαραίτητη για την αλληλεπίδραση του ανθρώπου με την παραγωγική διαδικασία καθώς και για τον έλεγχό της. Ουσιαστικά, αποτελεί τον τρόπο με τον οποίο παρέχεται η εποπτική εικόνα των δεδομένων σχετικά με την βιομηχανική λειτουργία στον άνθρωπο, δίνοντάς του έτσι την δυνατότητα να έχει έναν πλήρη έλεγχο καθώς και να επιδρά και ο ίδιος στην λειτουργία της βιομηχανικής διαδικασίας. Ένα χαρακτηριστικό παράδειγμα της αρχιτεκτονικής των συστημάτων SCADA παρουσιάζεται στην εικόνα που ακολουθεί: [12]



Εικόνα 1.2: Αρχιτεκτονική συστήματος SCADA

1.3 Είδη Custom SCADA

Κάθε κατασκευάστρια εταιρεία PLC παρέχει το απαραίτητο λογισμικό για το δικό της SCADA. Προκειμένου να δημιουργήσει κάποιος το δικό του HMI, επειδή τα εμπορικά SCADA είναι πολύ ακριβά για την εφαρμογή του, θέλει να προστατεύσει την τεχνογνωσία του και χρειάζεται ένα καλύτερο HMI που δεν μπορεί να το κάνει με ένα κανονικό SCADA, έχει την δυνατότητα να χρησιμοποιήσει μια γλώσσα προγραμματισμού όπως οι ακόλουθες: C# ή Visual Basic.Net. Το πρώτο και πιο σημαντικό πράγμα που απαιτείται είναι η επικοινωνία με το PLC και η αμφίδρομη αποστολή δεδομένων. Να γράφει το σύστημα δεδομένα από το PLC και να διαβάζει δεδομένα από το PLC. Αυτό μπορεί να γίνει με διάφορους τρόπους, μεταξύ των οποίων είναι η χρήση TCP και RTU, η χρήση OPC server, η χρήση προγραμμάτων οδήγησης ανοιχτού κώδικα (με άδεια gpl ή lgpl), όπως επίσης και η χρήση βιβλιοθηκών τρίτων (ActiveX, dll κ.α.). [1]



Εικόνα 1.3: Custom Scada

Το Modbus TCP βασίζεται σε σύνδεση Ethernet TCP/IP από PLC σε υπολογιστή και δεν υπάρχει ανάγκη για προσαρμοσμένη κάρτα σε υπολογιστή. Από την πλευρά του PLC απαιτείται θύρα TCP/IP που στις τελευταίες εκδόσεις των περισσότερων PLC είναι ενσωματωμένη και υποστηρίζει Modbus επικοινωνία. Τέτοια παραδείγματα αναφέρονται στην σειρά Allen Bradley Micrologix 1400 ή την σειρά Siemens S7-1200, κ.ο.κ. Για να επικοινωνήσει ένα PLC με Modbus TCP χρειάζεται μια βιβλιοθήκη που ονομάζεται nModbus. Αυτή η βιβλιοθήκη αποτελεί ένα πρότυπο για Modbus TCP και RTU, επειδή χρησιμοποιεί τις ίδιες μεθόδους και για τα δύο πρωτόκολλα. Το Modbus (ειδικά το TCP) είναι ένας καλός τρόπος επικοινωνίας με το PLC με C# και δίνει τη μεγαλύτερη ευελιξία, επειδή δημιουργεί κανείς το προγράμμα του και μπορεί να το εκτελέσει όπου θέλει χωρίς την προϋπόθεση ύπαρξης κάποιας άδειας χρήσης τυποποιημένου λογισμικού. Όλοι οι κατασκευαστές υποστηρίζουν το Modbus. Συγκεκριμένα, τα Siemens S7-300 και S7-400 υποστηρίζουν τόσο Modbus για TCP όσο και για RTU. Για το Modbus TCP χρειάζεται φυσικά μια κάρτα Ethernet ή την έκδοση CPU Profinet. Οι σειρές Allen Bradley CompactLogix και Contologix δεν υποστηρίζουν Modbus TCP και Modbus RTU, αλλά η Micrologix 1400 υποστηρίζει τόσο Modbus TCP όσο και RTU. [1]

Η χρήση του OPC είναι ο απλούστερος, ασφαλέστερος και ευκολότερος τρόπος επικοινωνίας με PLC. Ο OPC Server είναι βασικά ένα πρόγραμμα που συνδέεται με το PLC χρησιμοποιώντας βιβλιοθήκες κλειστής πηγής και εκθέτει τα δεδομένα με έναν τυπικό τρόπο που δεν εξαρτάται από το PLC που χρησιμοποιείται. Για να χρησιμοποιηθεί το OPC με C#, πρέπει να γράψει κανείς τον δικό του client χρησιμοποιώντας τις βιβλιοθήκες που παρέχονται από τον προμηθευτή του OPC Server, από το OPC Foundation ή μπορεί να χρησιμοποιήσει clients τρίτων. Συνήθως για να γραφεί ένας client πρέπει απλώς να δημιουργηθεί μια ομάδα με στοιχεία που αντιστοιχούν σε συνδέσεις μεταβλητών με το εκάστοτε PLC, τα οποία ανανεώνονται διαρκώς στην μονάδα του χρόνου. Το OPC επιτρέπει επίσης πολλές προηγμένες λειτουργίες, όπως: περιήγηση στο PLC, περιήγηση σε όλες τις ετικέτες σε δένδροειδή προβολή, διαχείριση συναγεργμών και συμβάντων και ούτω καθεξής. Αυτό που χρειάζεται είναι απλώς η ανάγνωση και η σύνταξη ετικετών με διαφορετική προτεραιότητα. Ο καλύτερος τρόπος επικοινωνίας με C# και OPC είναι να χρησιμοποιηθούν τα OpcNetA-pi.dll και OpcNetApi.Com.dll. Μπορεί κανείς να χρησιμοποιήσει τις βιβλιοθήκες του OPC Foundation και να αναδιανέμει τα dll που θα δημιουργηθούν σε τρίτους, λαμβάνοντας υπόψιν πάντα την σχετική τεκμηρίωση του OPC Foundation. Το μειονέκτημα του OPC Server είναι ότι δεν διατίθεται δωρεάν, πράγμα που σημαίνει ότι πρέπει να αγοραστεί ένα αντίγραφο του OPC Server για κάθε υπολογιστή όπου θα εκτελείται το πρόγραμμα και ότι λειτουργεί μόνο σε παράθυρα επειδή βασίζεται σε COM και DCOM. [1]

Τα προγράμματα οδήγησης ανοιχτού κώδικα είναι επίσης συνηθισμένη λύση για την ανάπτυξη εξειδικευμένων διεπαφών SCADA. Μία από τις πιο συνηθισμένες βιβλιοθήκες είναι της εταιρείας Siemens, η LibNoDave, που χρησιμοποιείται περισσότερο με κάρτες Ethernet (CP343-1). Είναι μια πολύ κοινή βιβλιοθήκη για επικοινωνία με τα PLC της Siemens και είναι καλά δοκιμασμένη, επειδή χρησιμοποιείται ευρέως. Υπάρχει επίσης μια βιβλιοθήκη προγράμματος οδήγησης σε C#, η S7.Net που είναι εύκολη στην χρήση της και πολύ αποτελεσματική. Μια βιβλιοθήκη με πολλές χρήσιμες λειτουργίες είναι επίσης η Snap7. Υποστηρίζει πολλές γλώσσες (C++, C#, Pascal, LabView κλπ.) και είναι αρκετά εύκολο να ρυθμιστεί. Τέλος, όσον αφορά στις εφαρμογές της Siemens μια βιβλιοθήκη που επίσης χρησιμοποιείται είναι η DotNetSiemensPLCToolBoxLibrary, που είναι ένα επίπεδο πάνω από τη LibNoDave. [1]

Υπάρχουν όμως και άλλοι κατασκευαστές που χρησιμοποιούν δικές τους εξειδικευμένες βιβλιοθήκες με εξίσου σημαντικά αποτελέσματα. Χαρακτηριστικό είναι το παράδειγμα της Allen Bradley με την χρήση μιας βιβλιοθήκης αναφοράς ανοιχτού κώδικα για τα PLC της,

που ονομάζεται libplctag. Υπάρχουν εφαρμογές για τις πιο δημοφιλείς γλώσσες, όπως η python και η C#. [1]

Το θετικό με τα προγράμματα οδήγησης ανοιχτού κώδικα είναι ότι είναι δωρεάν και μπορούν να εντοπιστούν σφάλματα και να βελτιώσουν την επικοινωνία όσο χρειάζεται για την εκάστοτε εφαρμογή. Τέλος συχνά παρατηρείται η χρήση βιβλιοθηκών τρίτων. Αυτές οι βιβλιοθήκες είναι προσαρμοσμένα αρχεία dll ή ActiveX και παρέχονται από τον κατασκευαστή ή τρίτες εταιρείες. Γενικά δεν είναι δωρεάν και θυμίζουν εφαρμογές παλαιού τύπου, όπου κάθε PLC είχε την δική του βιβλιοθήκη με τον δικό του κώδικα. [1]

1.4 Custom SCADA Εφαρμογές

Σε όλο τον κόσμο, η κύρια πολυπλοκότητα με τα συστήματα SCADA για τη βιομηχανία μικρής και μεσαίας κλίμακας είναι το υψηλό κόστος εγκατάστασης και η συμβατότητα του συστήματος. Έχουν πραγματοποιηθεί διάφορες έρευνες για τη βελτιστοποίηση του επενδυτικού κόστους στο σύστημα SCADA χωρίς να διακυβεύονται οι πιο ουσιαστικές και κρίσιμες ανάγκες του συστήματος. Το σύστημα SCADA απαιτείται να είναι ισχυρό με χαρακτηριστικά όπως ασφάλεια, οπτικοποίηση, έλεγχος παραμέτρων διαδικασίας, αυτόματος χειρισμός της διαδικασίας, σύστημα προειδοποίησης κλπ. Στη συνέχεια, θα παρουσιαστούν ορισμένες εφαρμογές που έχουν πραγματοποιηθεί με απώτερο σκοπό την παροχή της βέλτιστης δυνατής λύσης. [11]

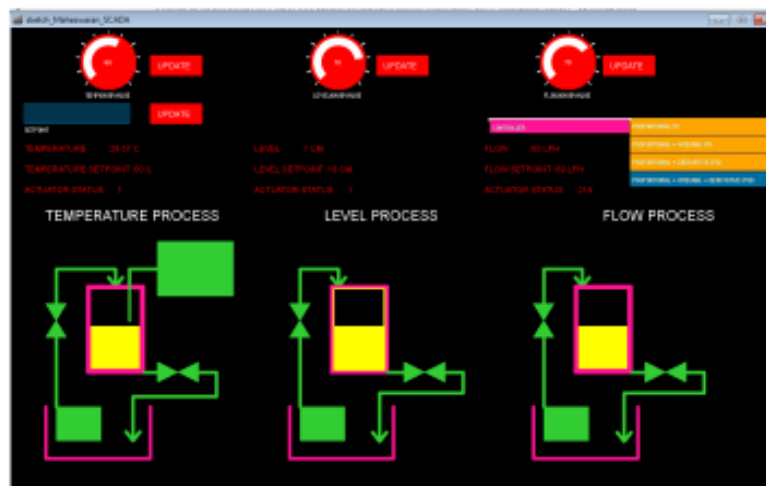
Οι Andrea Mercurio, Alessandro Di Giorgio και Pierfrancesco Cioci υλοποίησαν ένα βασισμένο στο web IEC πρότυπο σύστημα SCADA μειώνοντας το επενδυτικό κόστος χωρίς κανένα συμβιβασμό στην λειτουργική επίδοση του συστήματος. Πιο συγκεκριμένα, προχώρησαν στην υλοποίηση εποπτείας και ελέγχου υπηρεσιών για σύστημα διαχείρισης ενέργειας, εποπτικό έλεγχο και αποθήκευση δεδομένων βασισμένο στα διεθνή πρότυπα IEC 61970 και IEC 61850. Η προτεινόμενη λύση βασίζεται σε υπηρεσίες τεχνολογίας web που ενεργοποιούν τα τυπικά πλεονεκτήματα ενός πλαισίου ανοιχτού κώδικα, όπως αποσύνδεση εφαρμογών, ενσωμάτωση εφαρμογών, μείωση κόστους αναβαθμίσεων, επαναχρησιμοποίηση των ήδη υπάρχουσων υποδομών και ευκολία ανάπτυξης. Οι τυποποιημένες διεπαφές παρέχονται από την γενική διεπαφή ορισμού διεπαφών του IEC 61970. Ο web server που αναπτύχθηκε έχει δοκιμαστεί σε περιβάλλον προσομοίωσης όπου προσομοιώθηκαν οι φυσικές συσκευές ενός γενικού συστήματος ισχύος με βάσεις δεδομένων των οποίων τα μοντέλα δεδομένων παρέχονταν από το κοινό μοντέλο πληροφοριών του IEC 61970. [8]

Οι R.Immanuel Rajkumar, T.Jerry Alexander και Ponni Devi υλοποίησαν ένα low cost σύστημα SCADA που βασίζεται στο πρωτόκολλο επικοινωνίας ZigBee και μπορεί να χρησιμοποιηθεί σε εφαρμογές ελέγχου μιας βιομηχανικής διαδικασίας. Το λογισμικό επεξεργασίας που χρησιμοποιήσαν είναι ανοιχτού κώδικα και μπορεί να χρησιμοποιηθεί δωρεάν από όλους τους χρήστες. Η οπτικοποίηση της πλήρους διαδικασίας πραγματοποιήθηκε επίσης με το ίδιο λογισμικό. Η σειριακή επικοινωνία παίζει πολύ σημαντικό ρόλο για το προτεινόμενο σύστημα ενώ χρησιμοποιήθηκε και ένα ασύρματο δίκτυο αισθητήρων χρησιμοποιώντας μονάδες ZigBee. Με την βοήθεια αναπτυξιακής πλακέτας που βασίζεται σε arduino πραγματοποιήθηκε η συλλογή όλων των μετρούμενων μεγεθών (θερμοκρασία, ροή, επίπεδο στάθμης υγρού) που καταγράφονταν από τους αισθητήρες ενώ με την βοήθεια της μονάδας ZigBee πραγματοποιήθηκε η μετάδοση των τιμών αυτών στο τμήμα ελέγχου της διαδικασίας. Συνολικά, χρησιμοποιήθηκαν 2 μονάδες ZigBee, μία για το τμήμα επεξεργασίας όπου λαμβάνονται οι τιμές από τους αισθητήρες και αποστέλλονται στο τμήμα ελέγχου και μία από το τμήμα ελέγχου για την λήψη των μετρούμενων μεγεθών. Η

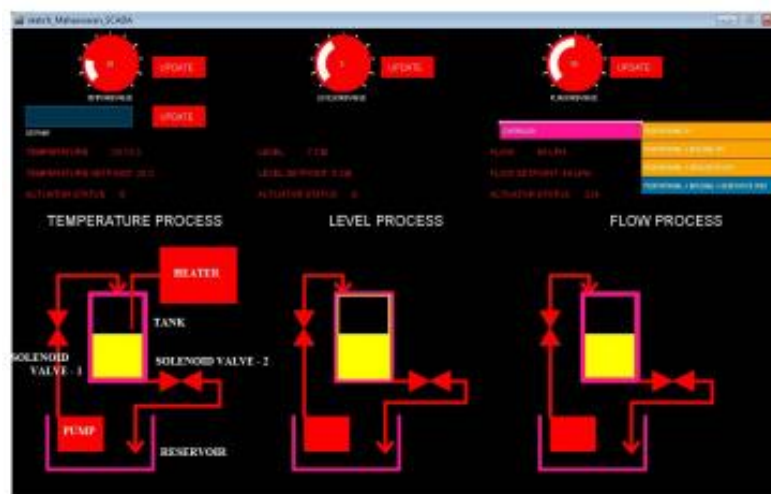
διαδικασία που εκτελείται από το προτεινόμενο σύστημα μπορεί να συνοψιστεί στα παρακάτω βήματα:[9]

- Επεξεργασία παραμέτρου μέτρησης από τον αντίστοιχο αισθητήρα
- Αποστολή της παραμέτρου επεξεργασίας στο τμήμα ελέγχου μέσω ZigBee
- Έπειτα από την λήψη των δεδομένων στην ενότητα ελέγχου γίνεται ενημέρωση των μεγεθών που οπτικοποιούνται και συγκρίνονται με τα δεδομένα ρύθμισης που μπορεί να εισάγει ο χρήστης μέσω του SCADA, σε περίπτωση εντοπισμού απόκλισης και βασιζόμενοι στον ελεγκτή επιλέγεται η απαιτούμενη διορθωτική ενέργεια ελέγχου που πρέπει να πραγματοποιηθεί και αποστέλλεται πίσω ως πληροφορία στο τμήμα επεξεργασίας
- Με βάση τα ληφθέντα δεδομένα, ο κατάλληλος ενεργοποιητής πραγματοποιεί την ανάλογη επίδραση στην όλη διαδικασία δίνοντας την δυνατότητα στο σύστημα να φτάσει στην απαιτούμενη τιμή ρύθμισης.
- Στη συνέχεια, το σύστημα ετοιμάζεται για την επόμενη απόκτηση δεδομένων πραγματοποιώντας μια ενημέρωση στο τμήμα το ελεγκτή.

Παρακάτω παρουσιάζεται η μορφή που παρουσιάζει το low cost SCADA που σχεδιάστηκε:[9]



Εικόνα 1.4:Custom Scada με όλους τους ενεργοποιητές σε κατάσταση ON



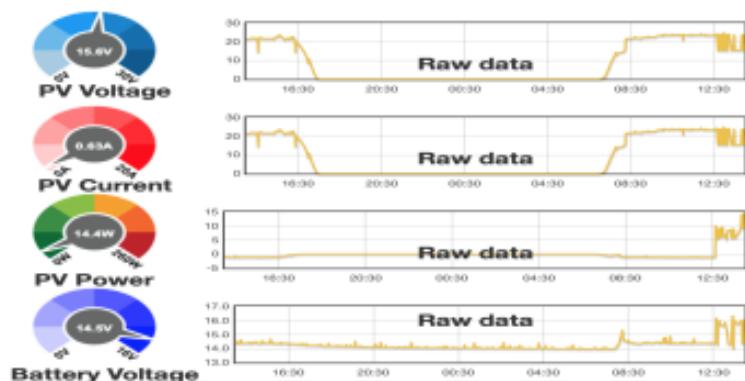
Εικόνα 1.5:Custom Scada με όλους τους ενεργοποιητές σε κατάσταση OFF

Συμπερασματικά λοιπόν, με την προτεινόμενη μέθοδο παρέχεται ολόκληρο το περιβάλλον της διαδικασίας με λιγότερο κόστος. Επιπλέον, το λογισμικό επεξεργασίας που βασίζεται στην JAVA μπορεί να αποδώσει επέκταση των ληφθέντων δεδομένων και είναι επίσης φιλικό προς το χρήστη παρέχοντας ασφάλεια στην όλη διαδικασία. Τέλος, εάν η προτεινόμενη μέθοδος που βασίζεται σε επικοινωνία μέσω ZigBee αντικατασταθεί από επικοινωνία που βασίζεται στο διαδίκτυο τότε μπορεί να βρει πρόσφορο έδαφος σε μεγαλύτερη πληθώρα εφαρμογών. [9]

Οι Lawrence O. Aghenta και M. Tariq Iqbal ανέπτυξαν ένα low cost SCADA σύστημα για την παρακολούθηση και τον έλεγχο συστήματος ηλιακού φωτοβολταϊκού. Το σύστημα που πρότειναν βασίζεται σε μια IoT SCADA αρχιτεκτονική που ενσωματώνει υπηρεσίες web με το συμβατικό SCADA για μια ισχυρή εποπτεία ελέγχου και επιτήρησης. Περιλαμβάνει αναλογικούς αισθητήρες ρεύματος και τάσης για την λήψη δεδομένων από το ηλιακό PV σύστημα, έναν μικροελεγκτή Arduino Uno που χρησιμεύει ως RTU για να λαμβάνει τα ληφθέντα δεδομένα από τον αισθητήρα, Raspberry Pi με ένα Node-Red εργαλείο προγραμματισμού για την ανάλυση των αποκτηθέντων δεδομένων και Emoncms Local Server IoT πλατφόρμα για αποθήκευση δεδομένων, παρακολούθηση και απομακρυσμένο χειρισμό. Το αναπτυγμένο σύστημα SCADA δημιουργήθηκε για την παρακολούθηση και τον έλεγχο ενός 260W, 12V ηλιακού φωτοβολταϊκού πάνελ στο εργαστήριο ηλεκτρολόγων μηχανικών και μηχανικών υπολογιστών στο Memorial University ενώ μερικά από τα γραφήματα (παρουσιάζονται παρακάτω) που δημιουργήθηκαν εμφανίζουν τα δεδομένα που αποκτήθηκαν από τον Emoncms server όπου ένας χειριστής μπορεί να παρακολουθεί τα δεδομένα στο cloud χρησιμοποιώντας υπολογιστή με πρόσβαση στο διαδίκτυο. [10]



Εικόνα 1.6: Δεδομένα οπτικοποίησης πραγματικού χρόνου



Εικόνα 1.7: Ακατέργαστα δεδομένα οπτικοποίησης

Συνοψίζοντας, το σύστημα SCADA που αναπτύχθηκε είναι ένα low cost SCADA ανοιχτού κώδικα βασισμένο στο Internet of Things. Περιλαμβάνει τα τέσσερα βασικά στοιχεία που χρειάζονται σε ένα σύστημα SCADA: συσκευές οργάνων πεδίου, απομακρυσμένες τερματικές μονάδες, κύριες τερματικές μονάδες και κανάλι επικοινωνίας SCADA. Η πειραματική ρύθμιση του αναπτυγμένου συστήματος SCADA πραγματοποιήθηκε στο εργαστήριο MUN ECE και χρησιμοποιήθηκε για την απόκτηση και την εξ' αποστάσεως παρακολούθηση και έλεγχο ενός συστήματος ηλιακού φωτοβολταϊκού πάνελ 260W, 12V. Το σύστημα έχει δοκιμαστεί και μπόρεσε να πραγματοποιήσει τις επιθυμητές λειτουργίες ενός SCADA συμπεριλαμβανομένης της απόκτησης δεδομένων, της δικτυακής επικοινωνίας δεδομένων, της παρουσίασης δεδομένων, της απομακρυσμένης παρακολούθησης και του εποπτικού ελέγχου. Το ανεπτυγμένο σύστημα SCADA μπορεί να εφαρμοστεί και σε άλλους κλάδους για την απομακρυσμένη παρακολούθηση και έλεγχο των κρίσιμων υποδομών, όπως για παράδειγμα: παραγωγή ηλεκτρικής ενέργειας, μεταφορά και διανομή της, κτήρια, εγκαταστάσεις, πετρέλαιο και εγκαταστάσεις παραγωγής φυσικού αερίου, συστήματα μαζικής μεταφοράς, συστήματα ύδρευσης και αποχέτευσης καθώς επίσης και συστήματα σημάτων κυκλοφορίας. [10]

Κεφάλαιο 2ο: Ανάλυση της εφαρμογής

2.1 Εισαγωγή

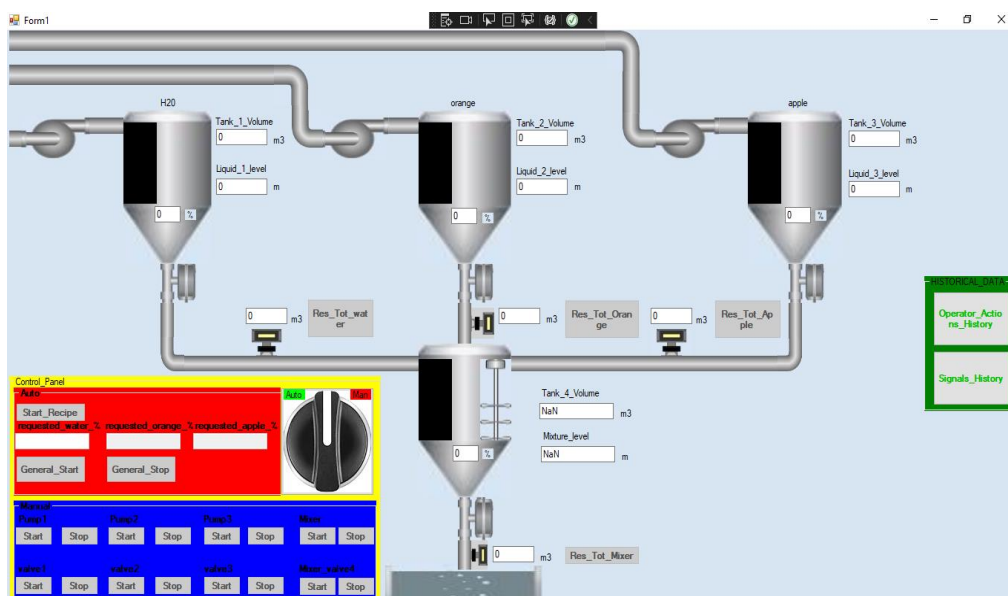
Ο εποπτικός έλεγχος και η συλλογή δεδομένων (SCADA) είναι μια ενοποίηση λογισμικού και υλικού που διευκολύνει την εποπτεία και τον έλεγχο βιομηχανικών ή οποιονδήποτε διαδικασιών εξ' αποστάσεως ή τοπικά. Το σύστημα που πρέπει να παρακολουθείται και να ελέγχεται περιλαμβάνει τη λειτουργία σε πραγματικό χρόνο και την καταγραφή δεδομένων, το χειρισμό διεργασιών και την πρόσβαση εποπτικού ελέγχου. Η μονάδα SCADA διευκολύνει την παρατήρηση και τον έλεγχο όλης της βιομηχανικής διαδικασίας εύκολα και πιο άνετα. Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, ένα από τα βασικότερα προβλήματα που αντιμετωπίζουν οι μικρότερες και μεσαίες βιομηχανίες είναι το υψηλό κόστος εγκατάστασης και η συμβατότητα του συστήματος. Επιπλέον, σε αυτήν την ταχέως αναπτυσσόμενη αγορά, μια επιτυχημένη ψηφιοποίηση της παραγωγής μπορεί να διαδραματίσει ζωτικό ρόλο στο βήμα προς τα εμπρός προς τη σύγχρονη εποχή. Για τις Ευρωπαϊκές μεταποιητικές βιομηχανίες, η ψηφιοποίηση και η σύγχρονη κατασκευή έχουν υποστηριχθεί από το Industry 4.0, το εργοστάσιο του μέλλοντος και τον οδικό χάρτη ProcessIT.EU για την αυτοματοποίηση διαδικασιών. Με την μείωση της επένδυσης κόστους στο σύστημα SCADA, θα ήταν πιο προσίτο και κατά συνέπεια μια τέτοια ενέργεια θα οδηγούσε στην αύξηση της χρήσης του και της περιοχής εφαρμογής του. Το κόστος μπορεί να μειωθεί χρησιμοποιώντας εργαλεία λογισμικού ανοιχτού κώδικα για την ανάπτυξη του συστήματος SCADA ενώ η προσαρμογή του αναπτυγμένου λογισμικού είναι επίσης δυνατή ανάλογα με τις απαιτήσεις. Στην συγκεκριμένη εργασία προτείνουμε ένα οικονομικό σύστημα SCADA που αναπτύχθηκε με την βοήθεια της αντικειμενοστραφούς γλώσσας προγραμματισμού C# και της βιβλιοθήκης S7.Net που παρέχει την απαραίτητη συμβατότητα με βιομηχανικούς ελεγκτές της εταιρείας Siemens. Η προσέγγιση αυτή βοηθά στην εξάλειψη της επένδυσης για την αγορά αδειοδοτημένων εργαλείων λογισμικού SCADA (τα οποία πολλές φορές απαιτούν επιπλέον κόστος αναβάθμισης) και παρέχει απόδοση που είναι συγκρίσιμη με αυτή που παρέχει το λογισμικό που είναι επί πληρωμή. [11]

2.2 Ανάλυση της εφαρμογής

Η συνολική διάταξη για την οποία αναπτύχθηκε το SCADA αποτελείται από 4 δεξαμενές συνολικά. Οι τρεις πρώτες από αυτές περιέχουν τα απαραίτητα συστατικά που οδηγούνται στην τέταρτη και τελευταία δεξαμενή η οποία περιέχει έναν αναδευτήρα προκειμένου να γίνει ανάμειξη των συστατικών και παραγωγή του τελικού προϊόντος. Πιο συγκεκριμένα, η πρώτη δεξαμενή τροφοδοτείται με νερό πυκνότητας $\rho=1000\text{kg/m}^3$, η δεύτερη δεξαμενή τροφοδοτείται με πολύ πορτοκαλιού πυκνότητας $\rho=850\text{kg/m}^3$, η τρίτη δεξαμενή τροφοδοτείται με πολύ μήλου πυκνότητας $\rho=890\text{kg/m}^3$ ενώ η πλήρωση της τελευταίας δεξαμενής πραγματοποιείται από τα συστατικά στοιχεία των τριών παραπάνω δεξαμενών με απώτερο σκοπό την πραγματοποίηση συγκεκριμένων συνταγών και την παραγωγή του επιθυμητού τελικού προϊόντος. Η πυκνότητα του τελικού προϊόντος υπολογίζεται με την βοήθεια των μαθηματικών και του σταθμισμένου μέσου όρου :
$$\frac{\rho_{\text{water}} \cdot \text{requested}_{\text{water}\%} + \rho_{\text{orange}} \cdot \text{requested}_{\text{orange}\%} + \rho_{\text{apple}} \cdot \text{requested}_{\text{apple}\%}}{\text{requested}_{\text{water}\%} + \text{requested}_{\text{orange}\%} + \text{requested}_{\text{apple}\%}}$$
 ανάλογα με το ποσοστό συνεισφοράς του κάθε συστατικού στο τελικό προϊόν. Η ακτίνα της κάθε δεξαμενής ισούται με $r=1\text{m}$ ενώ στον πυθμένα καθέμιας από τις παραπάνω δεξαμενές περιέχεται αναλογικό αισθητήριο μέτρησης της υδροστατικής πίεσης προκειμένου να μπορεί να υπολογιστεί τόσο η στάθμη του υγρού που περιέχεται σε κάθε δεξαμενή αλλά και ο συνολικός όγκος του με την βοήθεια των σχέσεων $P_{\text{υδρ}} = \rho \cdot g \cdot h$ και $V = \pi \cdot r^2 \cdot h$. Επιπλέον, με την βοήθεια της

πλατφόρμας του TIA Portal υπολογίζεται κάθε φορά και η αθροιστική ροή του υγρού που εκρέει από καθεμιά από τις παραπάνω δεξαμενές. Αναπτύχθηκαν 2 τρόποι λειτουργίας της διάταξης και επιπρόσθετα μέσω της συγκεκριμένης εφαρμογής ο χρήστης έχει την δυνατότητα προβολής και αναζήτησης, μεταξύ 2 συγκεκριμένων ημερομηνιών που ορίζει ο ίδιος, όλων των σημάτων της διάταξης αλλά και των ενεργειών του:

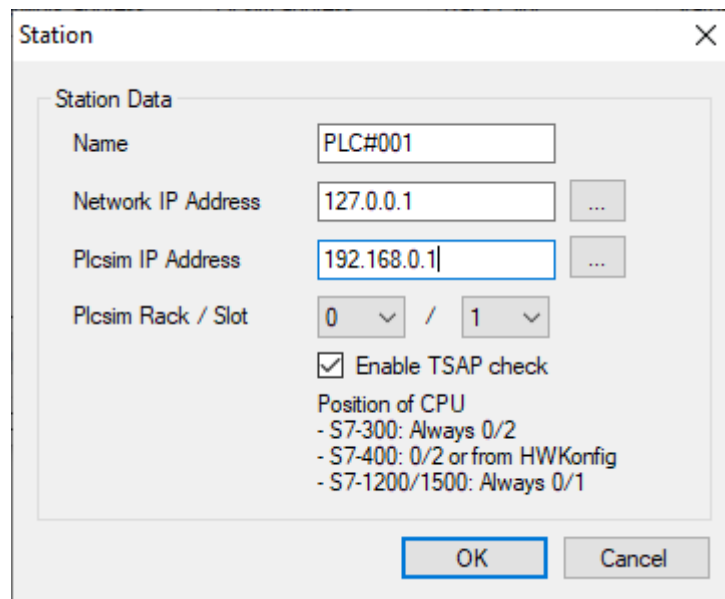
- **Αυτόματη Λειτουργία:** Κατά την αυτόματη λειτουργία υπάρχει η δυνατότητα εκτέλεσης συνταγών με απαραίτητη προϋπόθεση το συνολικό % άθροισμα ποσοστών από τις 3 δεξαμενές των συστατικών να είναι το 100 ενώ από την πρώτη δεξαμενή που περιέχει το νερό απαιτείται πάντοτε το ποσοστό 50%. Η δεξαμενή που περιέχει τον αναδευτήρα φτάνει πάντοτε στο 100% ενώ παράλληλα με την διαδικασία πληρωσής της πραγματοποιείται και η διαδικασία ανάμειξης μέχρι το 100% όπου και αρχίζει αυτόματα η διαδικασία εκροής και συλλογής του τελικού προϊόντος.
- **Χειροκίνητη Λειτουργία:** Κατά την χειροκίνητη λειτουργία υπάρχει η δυνατότητα χειροκίνητης επιλογής για το κάθε συστατικό του μείγματος με την βοήθεια button start και stop που ελέγχουν την κάθε ηλεκτροβάννα ξεχωριστά χωρίς κάποιο αντίστοιχο περιορισμό με αυτό της αυτόματης λειτουργίας. Κάθε φορά που ενεργοποιείται κάποια από τις τρεις ηλεκτροβάνες απενεργοποιείται αυτόματα η δυνατότητα επανεργοποίησής της. Δυνατότητα επανεργοποίησης της κάθε βάννας υπάρχει έπειτα από την ολοκλήρωση μιας οποιασδήποτε συνταγής η οποία μπορεί να πραγματοποιηθεί σε οποιοδήποτε ποσοστό % του μείγματος της δεξαμενής του αναδευτήρα. Απαραίτητη προϋπόθεση για την σωστή χειροκίνητη λειτουργία της διάταξης είναι η έγκαιρη απόκριση του χειριστή στην απενεργοποίηση της εκάστοτε ηλεκτροβάννας κάθε φορά που το ποσοστό της αντίστοιχης δεξαμενής φτάνει στο 0. Τέλος, προκειμένου να μεταβεί ο χειριστής από την χειροκίνητη σε οποιαδήποτε άλλη λειτουργία απαιτείται η εξάντληση των συστατικών που περιέχονται σε όλες τις δεξαμενές.
- **Προβολή ιστορικών στοιχείων:** Όταν ο περιστροφικός διακόπτης της εφαρμογής (φαίνεται παρακάτω) βρίσκεται στην μεσαία θέση ο χρήστης έχει την δυνατότητα να προβάλλει στην οθόνη του όλα τα ιστορικά στοιχεία από την λειτουργία της διάταξης και συγκεκριμένα μπορεί να επιλέξει την προβολή των ενεργειών που έχουν γίνει από τον ίδιο ή την προβολή των σημάτων της διάταξης και την αναζήτηση τους μεταξύ 2 συγκεκριμένων ημερομηνιών που εισάγει ο ίδιος μέσω της εφαρμογής.



Εικόνα 2.1: Custom Scada

Το PLC που χρησιμοποιήθηκε για την πραγματοποίηση των προσομοιώσεων είναι της εταιρείας SIEMENS, σειράς S7 1200 και συγκεκριμένα το 1214C DC/DC/RLY. Το DC/DC/RLY αναφέρεται στον τύπο του PLC. Το πρώτο στοιχείο αναφέρεται στην τροφοδοσία της μονάδας, το μεσαίο στοιχείο αναφέρεται στην τάση λειτουργίας των ψηφιακών εισόδων ενώ το τελευταίο στοιχείο αναφέρεται στον τύπο των ψηφιακών εξόδων. Το συγκεκριμένο PLC διαθέτει δύο αναλογικές εισόδους από 0 έως 10V. Η αναλογική αυτή τιμή μπορεί να αναλυθεί σε αριθμητικές τιμές από 0 έως 27648.

Με την βοήθεια του εκτελέσιμου αρχείου NetToPLCsim πραγματοποιήθηκε πρόσβαση στον Plcsim διαμέσου επικοινωνίας TCP/IP χρησιμοποιώντας την local host του H/Y στον οποίο πραγματοποιείται η προσομοίωση.



Εικόνα 2.2: Ρυθμιστικές παράμετροι για την πρόσβαση στον Plcsim

Όπως γίνεται κατανοητό από την παραπάνω εικόνα προκειμένου να επιτευχθεί η επικοινωνία ανάμεσα στο SCADA και τον Plcsim ορίζονται οι παρακάτω παράμετροι:

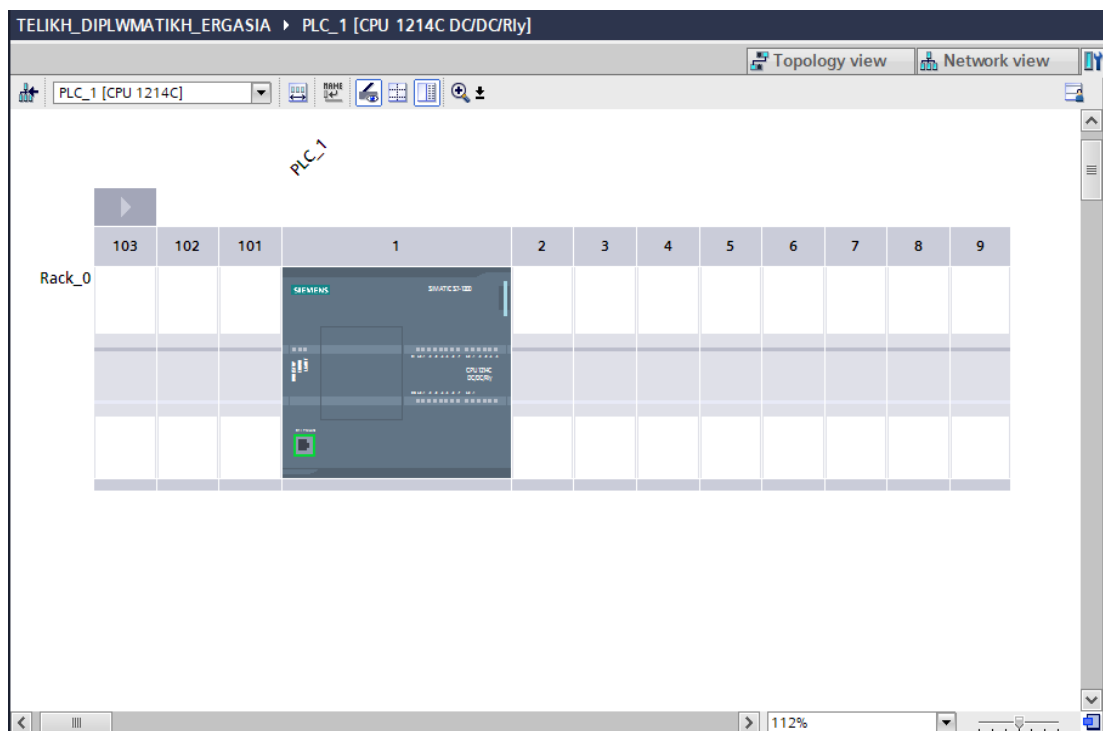
- Name: αντιπροσωπεύει το μοναδικό όνομα του PLC
- Network IP Address: αντιπροσωπεύει την διεύθυνση IP της διεπαφής του δικτύου στην οποία ο server πρέπει να είναι προσβάσιμος
- Plcsim IP Address: αντιπροσωπεύει την διεύθυνση IP του Plcsim
- Plcsim Rack/Slot: αντιπροσωπεύει την θέση στην οποία βρίσκεται η CPU του PLC (στην περίπτωση μας το PLC είναι τοποθετημένο στο rack 0 και στο slot 1) και πρέπει να συνοδεύεται πάντοτε με ενεργοποιημένη την επιλογή του TSAP check. Όταν η επιλογή αυτή είναι ενεργή τότε η εφαρμογή χρειάζεται να χρησιμοποιήσει το σωστό TSAP που αντιστοιχεί στον συνδυασμό rack/slot.

Κεφάλαιο 3ο: Προγραμματισμός του PLC

3.1 Περιγραφή των function block και του device configuration

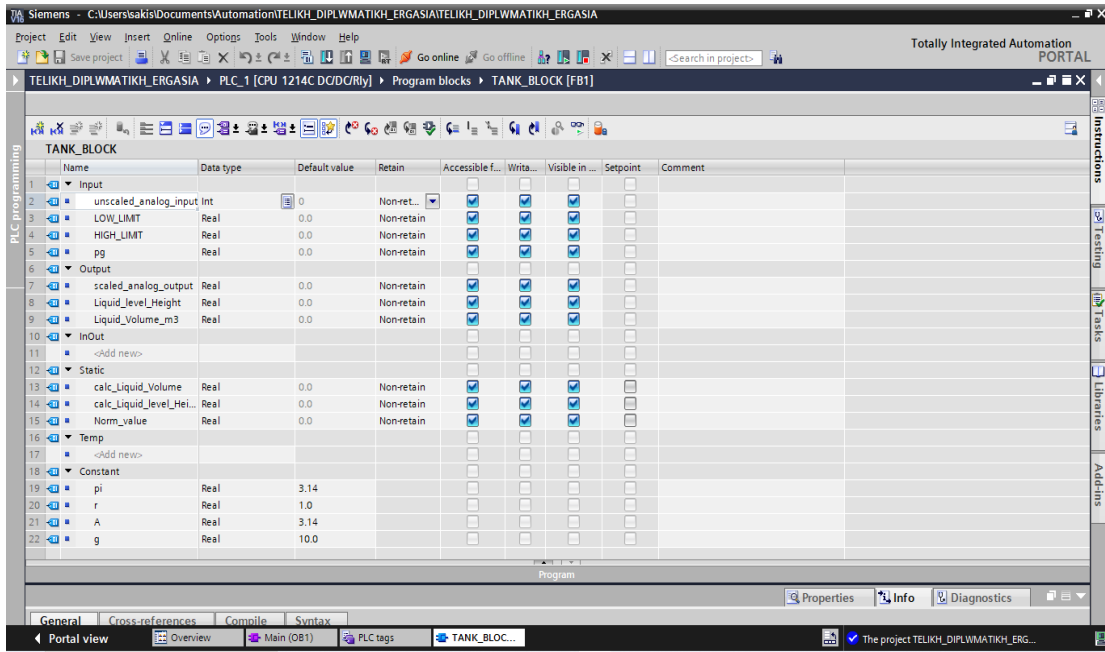
Για τον προγραμματισμό του PLC χρησιμοποιήθηκε το πρόγραμμα της SIEMENS το TIA Portal V16. Το TIA Portal V16 είναι ένα ολοκληρωμένο πρόγραμμα που χρησιμοποιείται στους αυτοματισμούς, παρέχει φιλικό περιβάλλον προς τον χρήστη και χρησιμοποιείται για την ανάπτυξη του λογικού προγράμματος του PLC, την ρύθμιση παραμέτρων αλλά και την ρύθμιση του δικτύου επικοινωνίας. Επιπλέον, παρέχει την δυνατότητα οπτικοποίησης του HMI, μια δυνατότητα που δεν χρησιμοποιήθηκε στην συγκεκριμένη εργασία καθώς η διαδικασία αυτή πραγματοποιήθηκε με την βοήθεια της αντικειμενοστραφούς γλώσσας προγραμματισμού C#.

Ξεκινώντας το πρόγραμμα του TIA Portal το πρώτο βήμα είναι πάντοτε η επιλογή της κατάλληλης CPU που θα χρησιμοποιηθεί για την εφαρμογή μας(device configuration). Η CPU που επιλέχθηκε είναι της σειράς S71200 και συγκεκριμένα η 1214C DC/DC/RIy και τοποθετήθηκε στο rack 0 και στο slot 1 όπως παρουσιάζεται στην παρακάτω εικόνα:

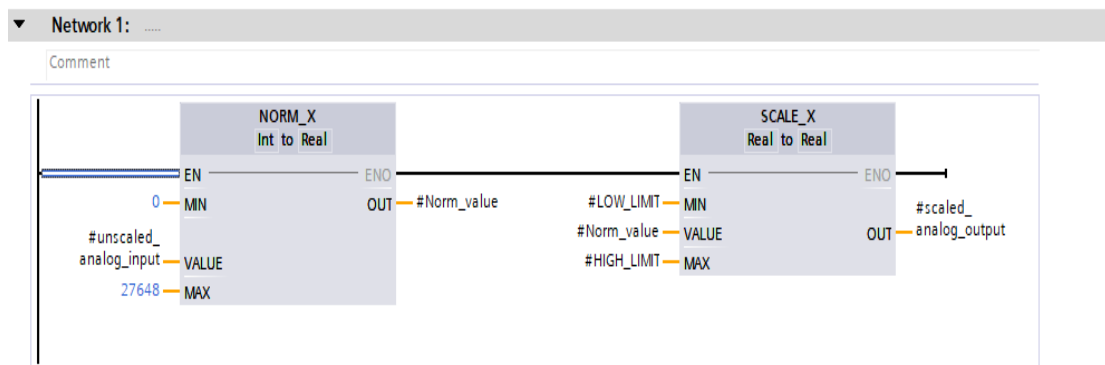


Εικόνα 3.1: Plc εφαρμογής

Επόμενο βήμα είναι ο προγραμματισμός του PLC ο οποίος πραγματοποιήθηκε αποκλειστικά με την βοήθεια της γλώσσας Ladder. Στη συνέχεια παρουσιάζεται η δομή του function block που χρησιμοποιήθηκε για καθεμιά από τις 4 δεξαμενές της εφαρμογής και μέσω του οποίου υπολογίζεται ο όγκος και το ύψος του υγρού της κάθε δεξαμενής.

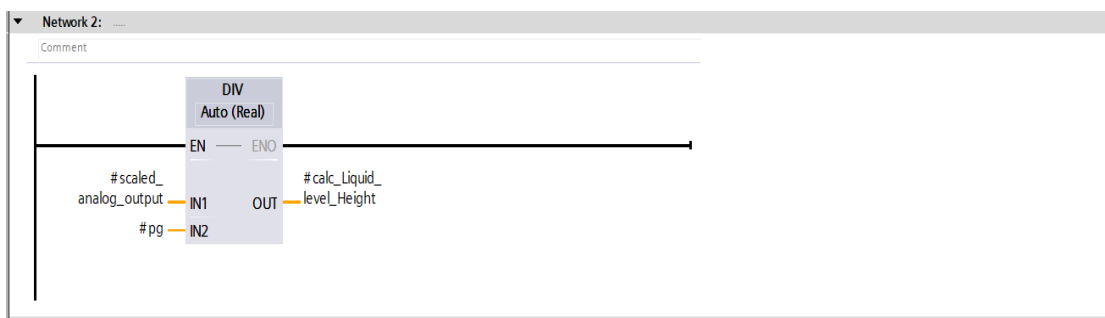


Εικόνα 3.2: Ετικέτες για το function block της δεξαμενής



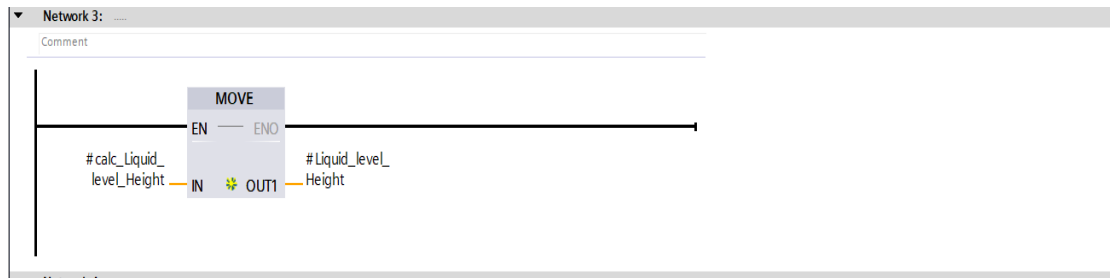
Εικόνα 3.3: Κανονικοποίηση και ρύθμιση κλίμακας για το σήμα εισόδου

Στο πρώτο network πραγματοποιείται η κανονικοποίηση του αρχικά ακατέργαστου σήματος (0-27648) της εισόδου και στην συνέχεια εισάγεται η επιθυμητή κλίμακα για τις μετρήσεις που λαμβάνονται κάθε φορά. Στην περιπτώσή μας το αρχικό σήμα μετατρέπεται σε ένα κλιμακούμενο μέγεθος 0-10000Pa που είναι οι ακραίες τιμές πίεσης που μπορούμε να λάβουμε (στην πραγματικότητα λαμβάνουμε ελάχιστα λιγότερο από 10000Pa καθώς το αρχικά ακατέργαστο σήμα έχει χωριστεί σε ακέραια πολλαπλάσια του 276 στην δομή του προγράμματος που θα δούμε παρακάτω με αποτέλεσμα το ευρος που χρησιμοποιείται και μεταφράζεται σε πίεση να είναι το 0-27600 αντι για 0-27648).



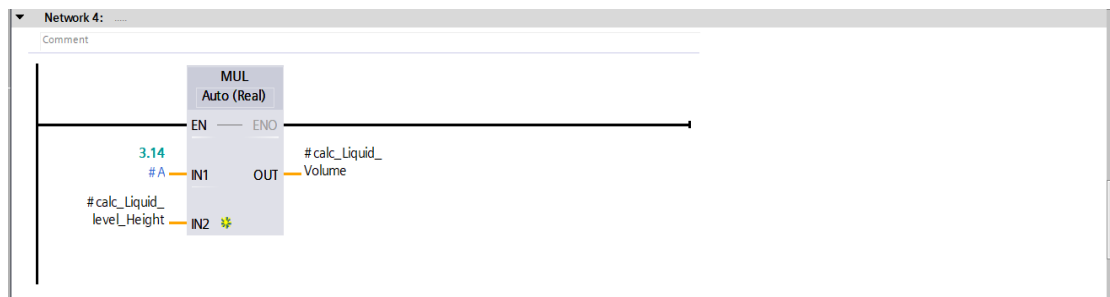
Εικόνα 3.4: Υπολογισμός στάθμης του υγρού

Στο δεύτερο network πραγματοποιείται ο υπολογισμός της στάθμης του υγρού διαιρώντας την κλιμακούμενη τιμή της υδροστατικής πίεσης με το γινόμενο της πυκνότητας του υγρού και της επιτάχυνσης της βαρύτητας



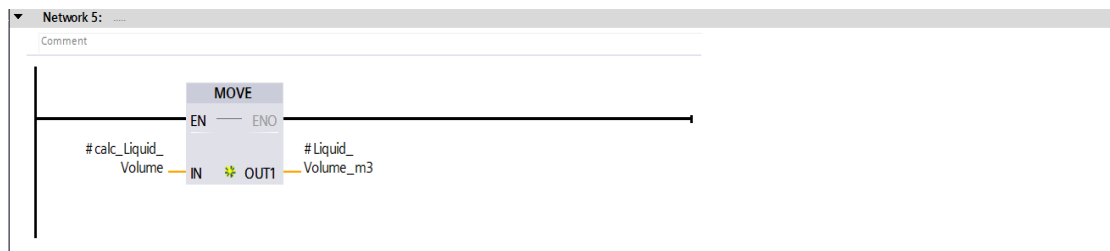
Εικόνα 3.5: Μεταφορά υπολογισμένης στάθμης υγρού στην έξοδο του function block

Στο τρίτο network γίνεται μεταφορά της, υπολογισμένης από το προηγούμενο network, στάθμης του υγρού στην έξοδο function block



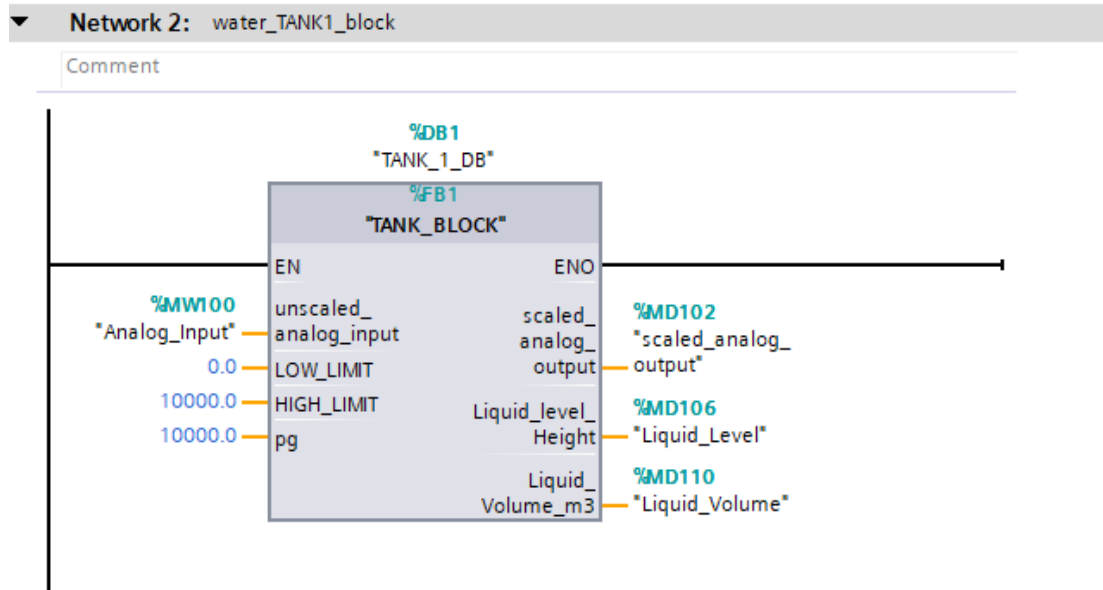
Εικόνα 3.6: Υπολογισμός όγκου του υγρού

Στο τέταρτο network πραγματοποιείται ο υπολογισμός του όγκου του υγρού με την βοήθεια της σχέσης: $V = \pi \cdot r^2 \cdot h$, όπου $\pi = 3.14$ και $r = 1m$.



Εικόνα 3.7: Μεταφορά του υπολογισμένου όγκου του υγρού στην έξοδο του function block

Στο πέμπτο και τελευταίο network του συγκεκριμένου function block γίνεται μεταφορά του όγκου του υγρού, που έχει υπολογιστεί από το προηγούμενο network, στην έξοδο του function block.

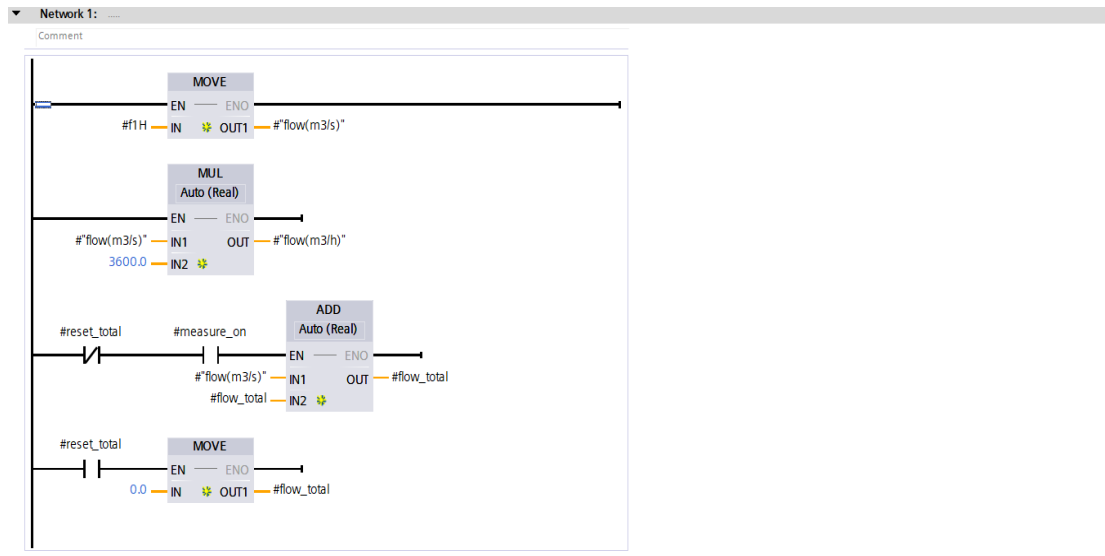


Εικόνα 3.8: Function block για την δεξαμενή του νερού

Έπειτα παρουσιάζεται η δομή ενός ακόμη function block που χρησιμοποιήθηκε για τον υπολογισμό της αθροιστικής ροής του υγρού που εκρέει από κάθε δεξαμενή έτσι ώστε να έχουμε εποπτική εικόνα για την συνολική ποσότητα συστατικών που δαπανάται αλλά και την συνολική ποσότητα του τελικού προϊόντος που παράγεται κάθε φορά.

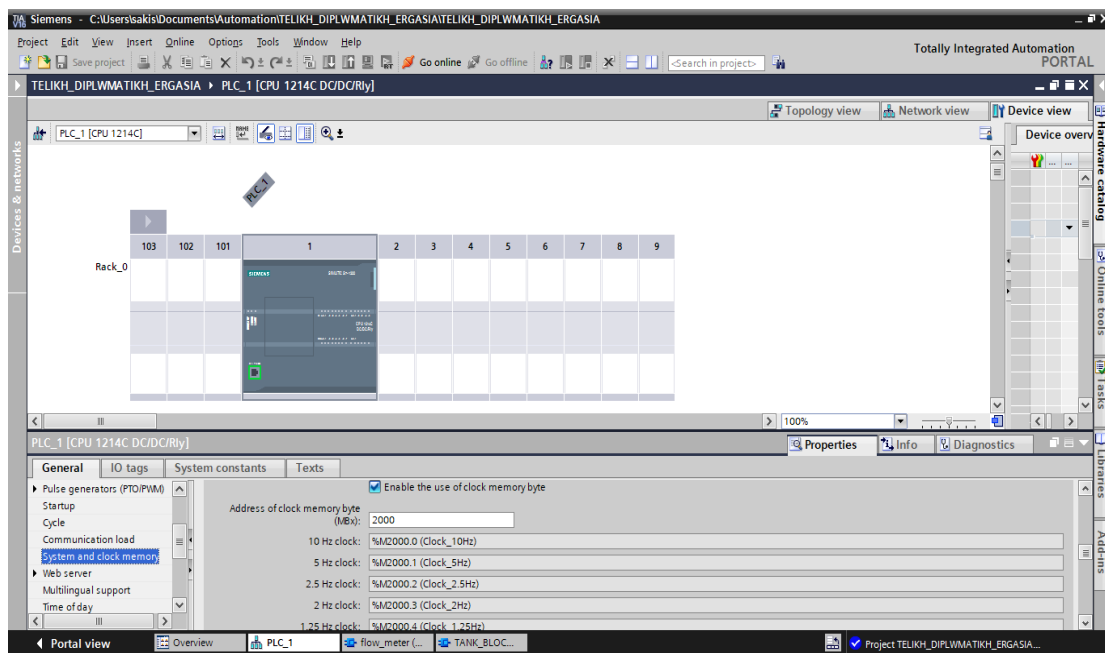
	Name	Data type	Default value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Input								
2	measure_on	Bool	false	Non-ret...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	reset_total	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	flH	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	helping_tag_for_A12	Int	0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	Output								
7	flow(m3/s)	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
8	flow(m3/h)	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
9	flow_total	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
10	InOut								
11	<Add new>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
12	Static								
13	<Add new>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Εικόνα 3.9: Ετικέτες για το function block υπολογισμού της αθροιστικής ροής



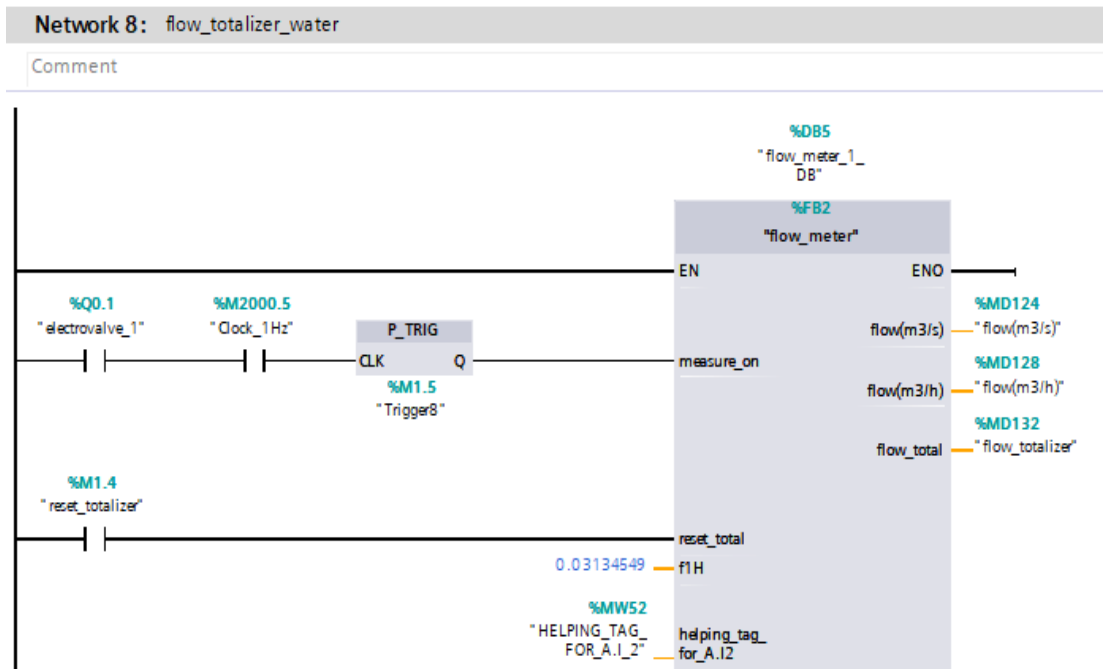
Εικόνα 3.10: Υπολογισμός της ροής σε m^3/s , m^3/h καθώς και της αθροιστικής ροής

Στο παραπάνω network που είναι και το μοναδικό για το συγκεκριμένο function block πραγματοποιείται υπολογισμός της ροής που εκρέει από την δεξαμενή του νερού σε μονάδες μέτρησης: m^3/s και m^3/h , επιπλέον γίνεται υπολογισμός της αθροιστικής ροής του νερού που εκρέει ενώ δίνεται και η δυνατότητα μηδενισμού του μετρητή της αθροιστικής ροής στον χειριστή. Το ίδιο function block χρησιμοποιείται και για τις υπόλοιπες δεξαμενές. Στο σημείο αυτό πρέπει να σημειωθεί ότι η ροή του υγρού (σε m^3/s) από κάθε δεξαμενή υπολογίζεται ως σταθερή και κάνοντας χρήση του clock memory bit 1Hz και με βήμα ίσο με το $\frac{1}{100}$ του συνολικού όγκου του υγρού της εκάστοτε δεξαμενής. Η ρύθμιση που απαιτείται για την ενεργοποίηση του clock memory byte που διαθέτει το TIA Portal για την χρησιμοποίηση του clock memory bit 1Hz αλλά και άλλων clock memory bit που διαθέτει παρουσιάζεται στην παρακάτω εικόνα:



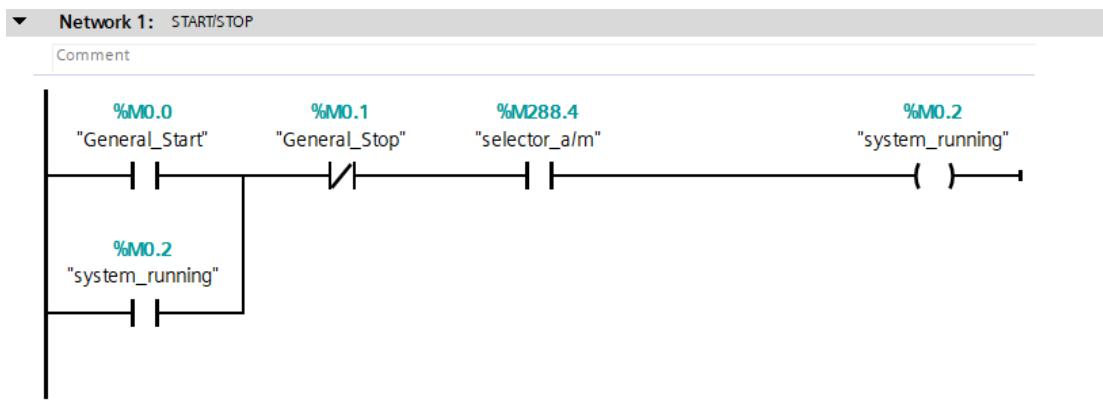
Εικόνα 3.11: Ενεργοποίηση του clock memory byte

Όπως παρατηρούμε από την παραπάνω εικόνα έχοντας επιλέξει το PLC επιλέγουμε Properties στην συνέχεια General , έπειτα System and clock memory και τέλος ενεργοποιούμε την επιλογή <<Enable the use of clock memory byte>>.



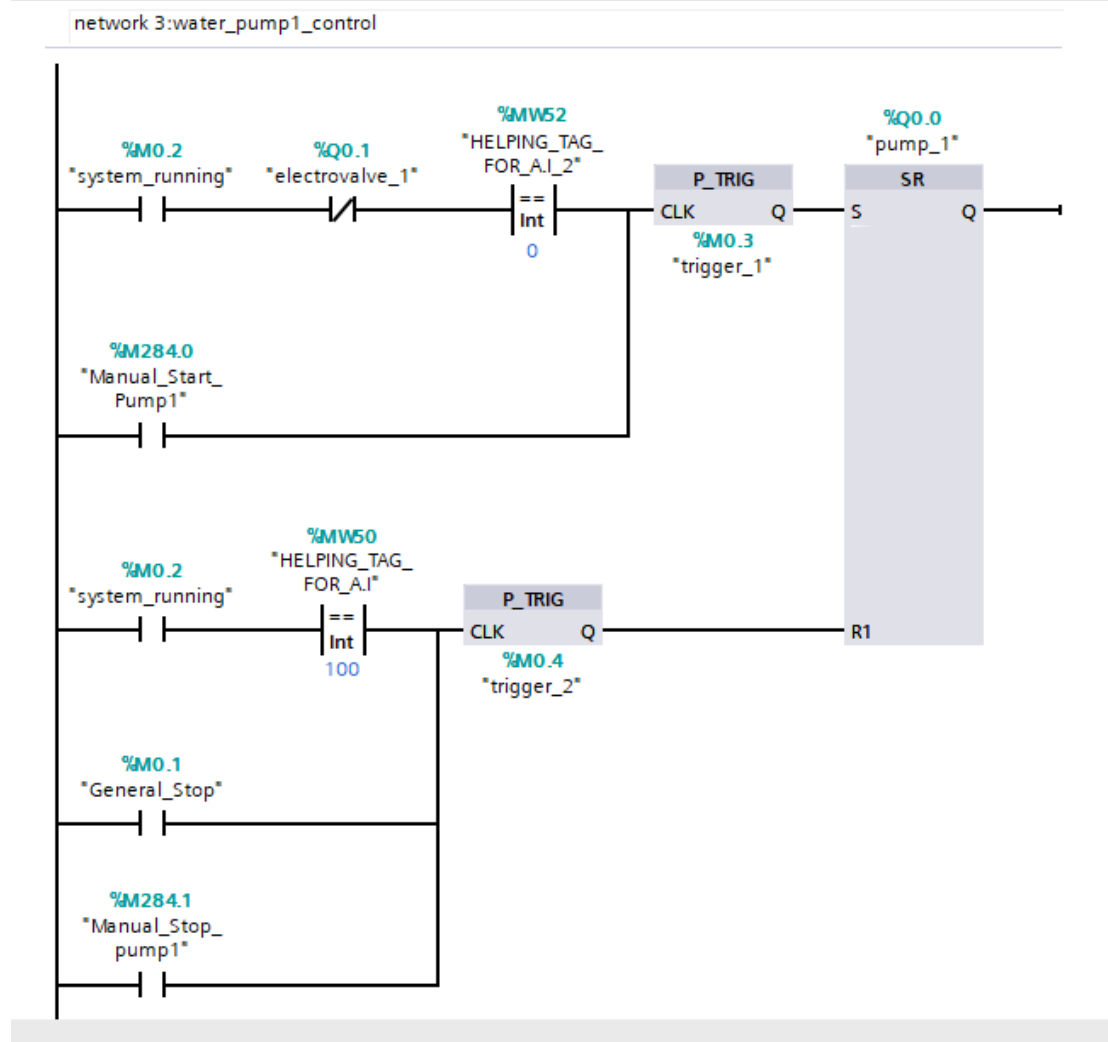
Εικόνα 3.12: Function block υπολογισμού αθροιστικής ροής του νερού

Στη συνέχεια παρουσιάζεται η δομή του κώδικα που χρησιμοποιήθηκε για την πλήρωση της πρώτης δεξαμενής με νερό καθώς επίσης και η δομή του κώδικα για την εκροή του νερού από την δεξαμενή. Αντίστοιχος είναι ο κώδικας που χρησιμοποιήθηκε και για τις άλλες 2 δεξαμενές συστατικών τόσο για την διαδικασία της πλήρωσης όσο και για την διαδικασία της εκροής. Η όλη διαδικασία στηρίζεται στην χρήση ενός counter up με την βοήθεια του οποίου πραγματοποιείται η πλήρωση της δεξαμενής από μια αντλία και ενός counter down με την βοήθεια του οποίου πραγματοποιείται η εκροή του υγρού της δεξαμενής από μια ηλεκτροβάννα. Ως άνω και κάτω όρια για τις τιμές του υγρού στην κάθε δεξαμενή έχουν χρησιμοποιηθεί το 100% και το 0% αντίστοιχα.



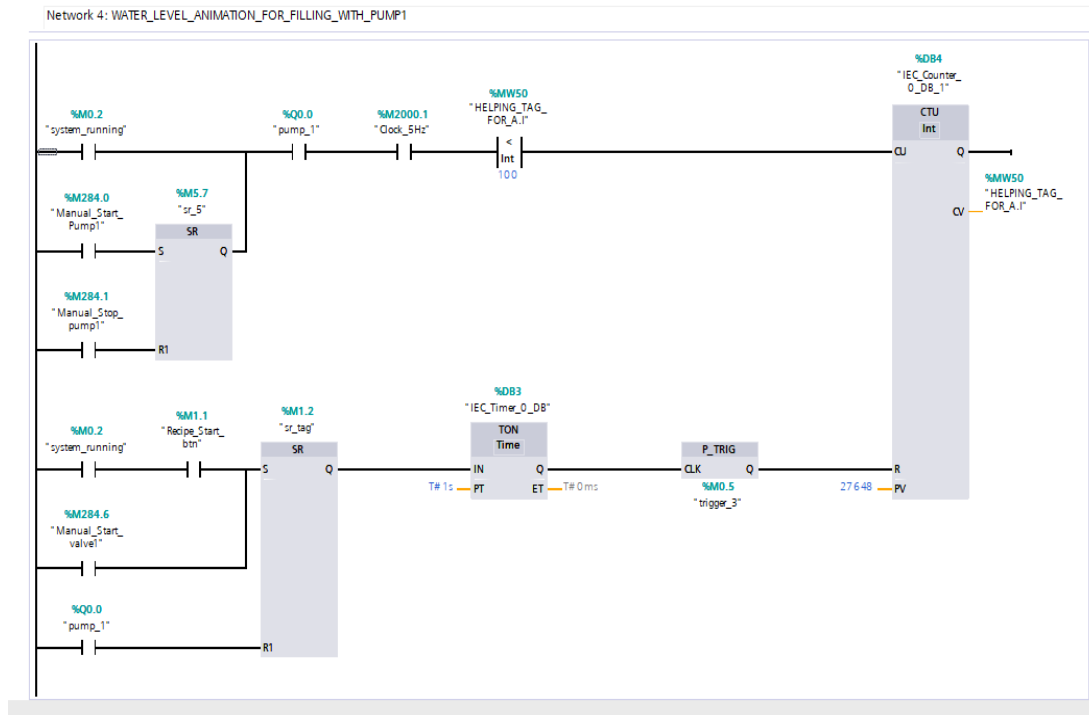
Εικόνα 3.13: Αυτόματη εκκίνηση και σταμάτημα

3.2 Διαδικασία πλήρωσης δεξαμενής νερού



Εικόνα 3.14: Έλεγχος αντλίας νερού

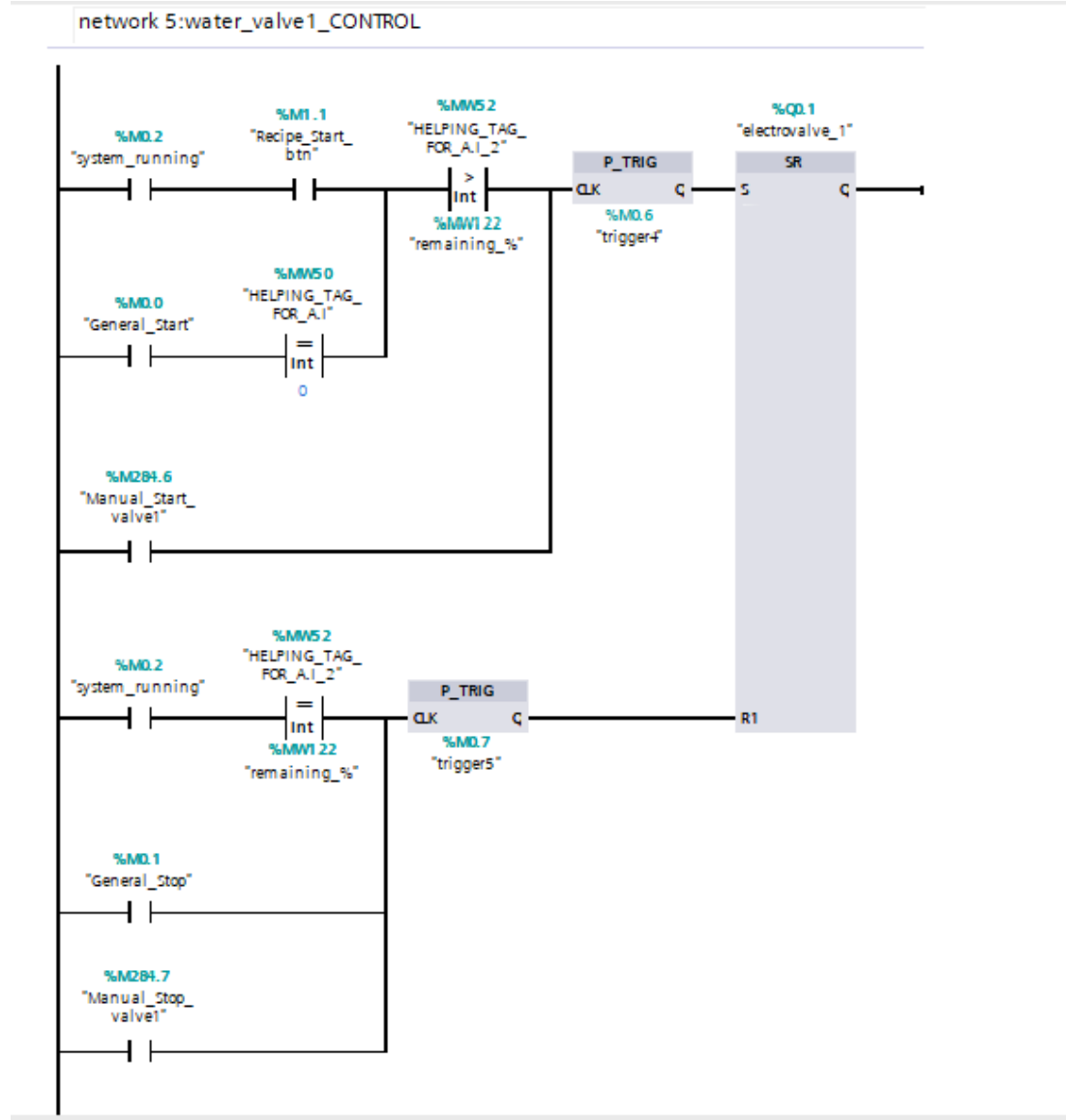
Η επαφή M0.2 ενεργοποιείται όταν η επαφή M288.4 του network 1 είναι ενεργή (ο περιστροφικός διακόπτης είναι στην αυτόματη λειτουργία) και έχει πατηθεί το κουμπί general_start της αυτόματης λειτουργίας, η μεταβλητή MW52 αντιπροσωπεύει την current value του counter down του network 6 και η οποία όταν είναι ίση με το μηδέν με την M0.2 ενεργή και την ηλεκτροβάννα 1 ανενεργή ενεργοποιείται η λειτουργία της αντλίας 1. Η ενεργοποίηση αυτή μπορεί να πραγματοποιηθεί και χειροκίνητα από την επαφή M284.0 πατώντας το κουμπί Manual_Start_pump1 του Scada. Η διαδικασία της πλήρωσης σταματά με την απενεργοποίηση της αντλίας 1 όταν η MW50, που αντιπροσωπεύει την current value του counter up του network 4, πάρει την τιμή 100 ή εναλλακτικά όταν πατηθεί το general_stop (M0.1) της αυτόματης λειτουργίας ή όταν πατηθεί το Manual_Stop_pump1 (M284.1) της χειροκίνητης λειτουργίας όπως φαίνεται στο network 3.



Εικόνα 3.15: Πλήρωση δεξαμενής νερού

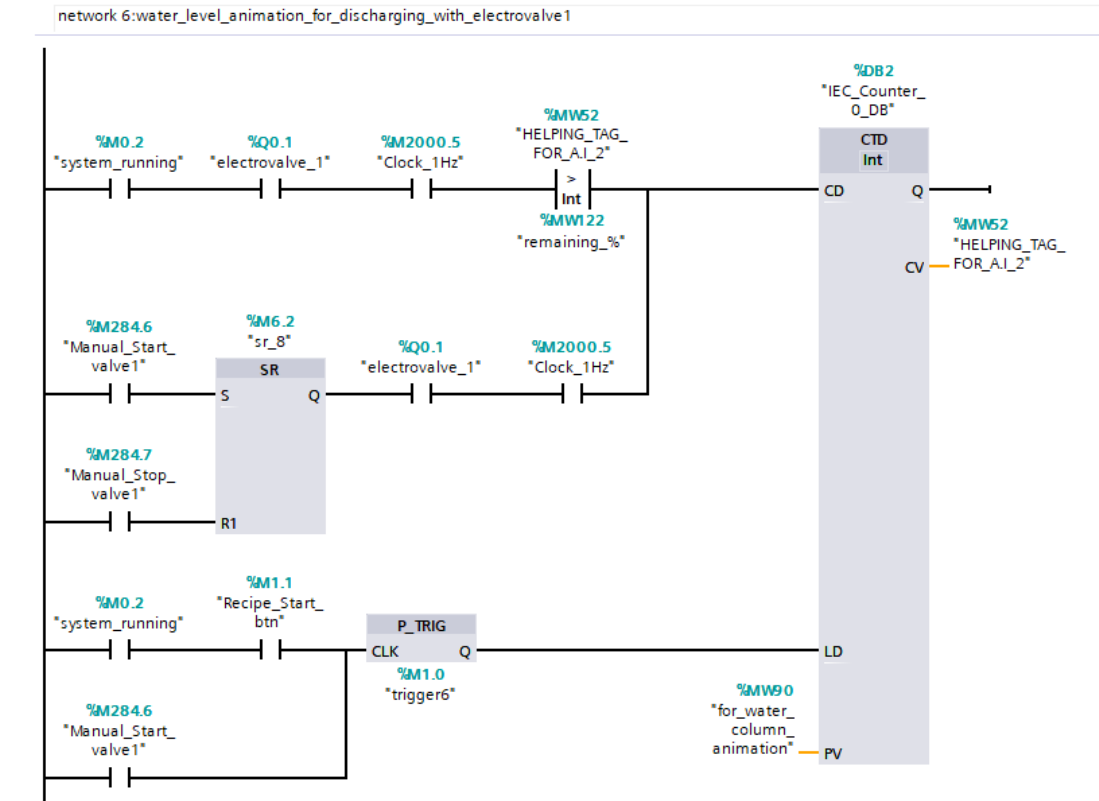
Όταν βρισκόμαστε στην αυτόματη λειτουργία με την αντλία 1 να είναι ενεργή και την current value του counter up να είναι μικρότερη του 100 τότε η πλήρωση της δεξαμενής συνεχίζεται μέχρι την τιμή 100 με την βοήθεια του clock memory bit 5Hz που διαθέτει το TIA Portal ή εναλλακτικά για όσο χρόνο είναι ενεργό το SR f/f (M5.7) όταν βρισκόμαστε στην χειροκίνητη λειτουργία. Με το πάτημα του Recipe_Start (M1.1) πραγματοποιείται μηδενισμός της current value του counter up με μια μικρή χρονική καθυστέρηση 1s προκειμένου να πραγματοποιηθεί πρώτα φόρτωση της τιμής αυτής στον CTD του network 6(που ουσιαστικά ταυτίζεται με την MW90 κατά την διαδικασία της πλήρωσης) ή εναλλακτικά μηδενισμός της MW50 μπορεί να πραγματοποιηθεί με το πάτημα του Manual_Start_valve1(M284.6).

3.3 Διαδικασία εκροής και βαθμονόμησης εισόδου δεξαμενής νερού



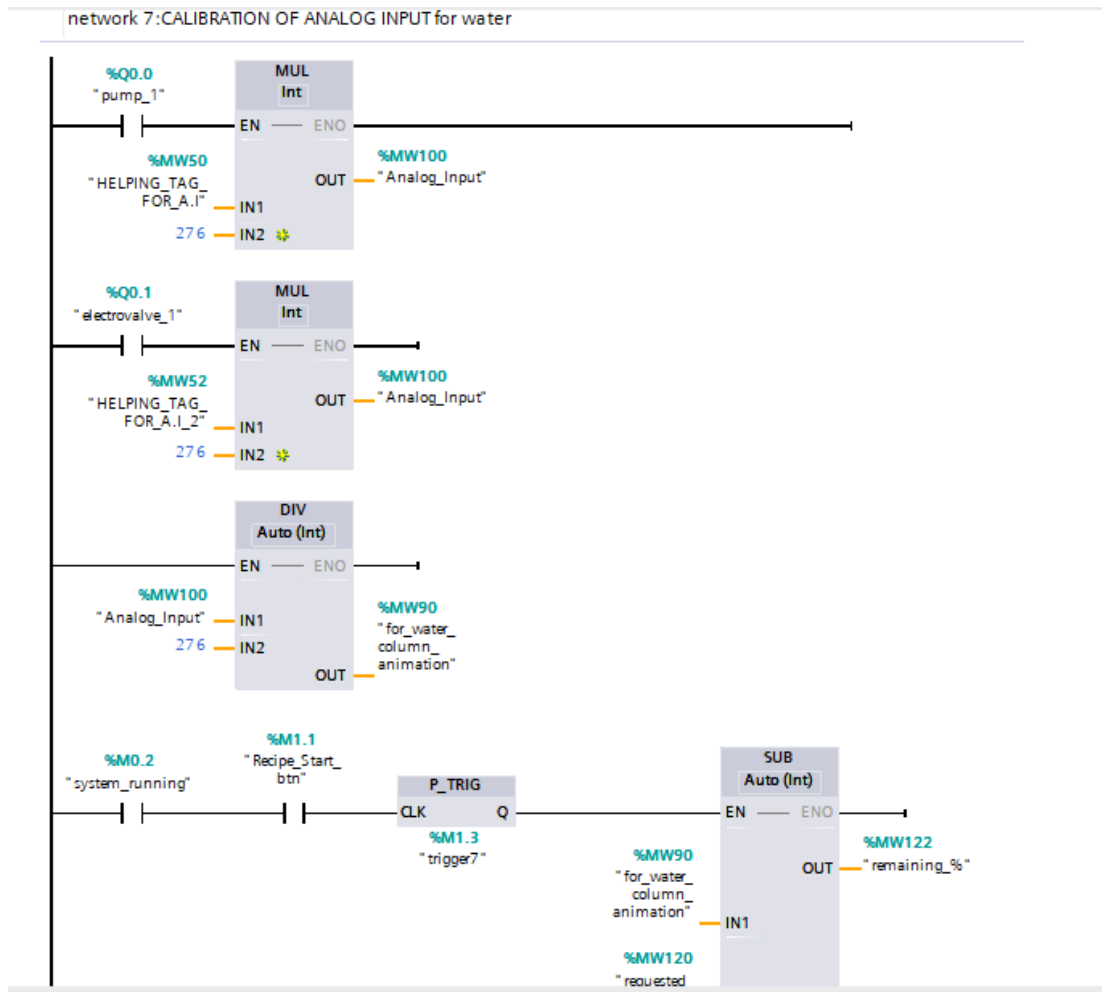
Εικόνα 3.16: Έλεγχος ηλεκτροβάνας δεξαμενής νερού

Αναφορικά με την διαδικασία της εκροής του υγρού κατά την αυτόματη λειτουργία και με την επαφή M0.2 ενεργή ο χρήστης πληκτρολογεί τις απαιτούμενες ποσότητες εκροής μέσω του scada και πατώντας το button Recipe_Start(M1.1) πραγματοποιείται ενεργοποίηση της ηλεκτροβάνας 1 (ταυτόχρονα και των άλλων δύο ή της μίας ανάλογα με τις ποσότητες εκροής που ζητά κάθε φορά) και αρχίζει η εκροή του υγρού προς τον αναδευτήρα εφόσον η current value MW52 του CTD του network 6 είναι μεγαλύτερη από την μεταβλητή MW122 που αντιπροσωπεύει την διαθέσιμη ποσότητα του υγρού στην δεξαμενή και υπολογίζεται με την βοήθεια του network 7 μέσα στο οποίο πραγματοποιείται αφαίρεση από την συνολική ποσότητα (%) του υγρού που περιέχεται στην δεξαμενή της ποσότητας που ζητά ο χειριστής κάθε φορά με το πάτημα του button Recipe_Start(M1.1). Μόλις η % ποσότητα υγρού(MW52) της δεξαμενής γίνει ίση με την % διαθέσιμη ποσότητα υγρού(MW122) της δεξαμενής τότε η ηλεκτροβάνα 1 απενεργοποιείται και η διαδικασία εκροής σταματά. Εναλλακτικά, κατά την χειροκίνητη λειτουργία η ηλεκτροβάνα 1 ενεργοποιείται με το πάτημα του Manual_Start_valve1(M284.6) και απενεργοποιείται με το πάτημα του Manual_Stop_valve1(M284.7).



Εικόνα 3.17: Εκροή νερού από την δεξαμενή

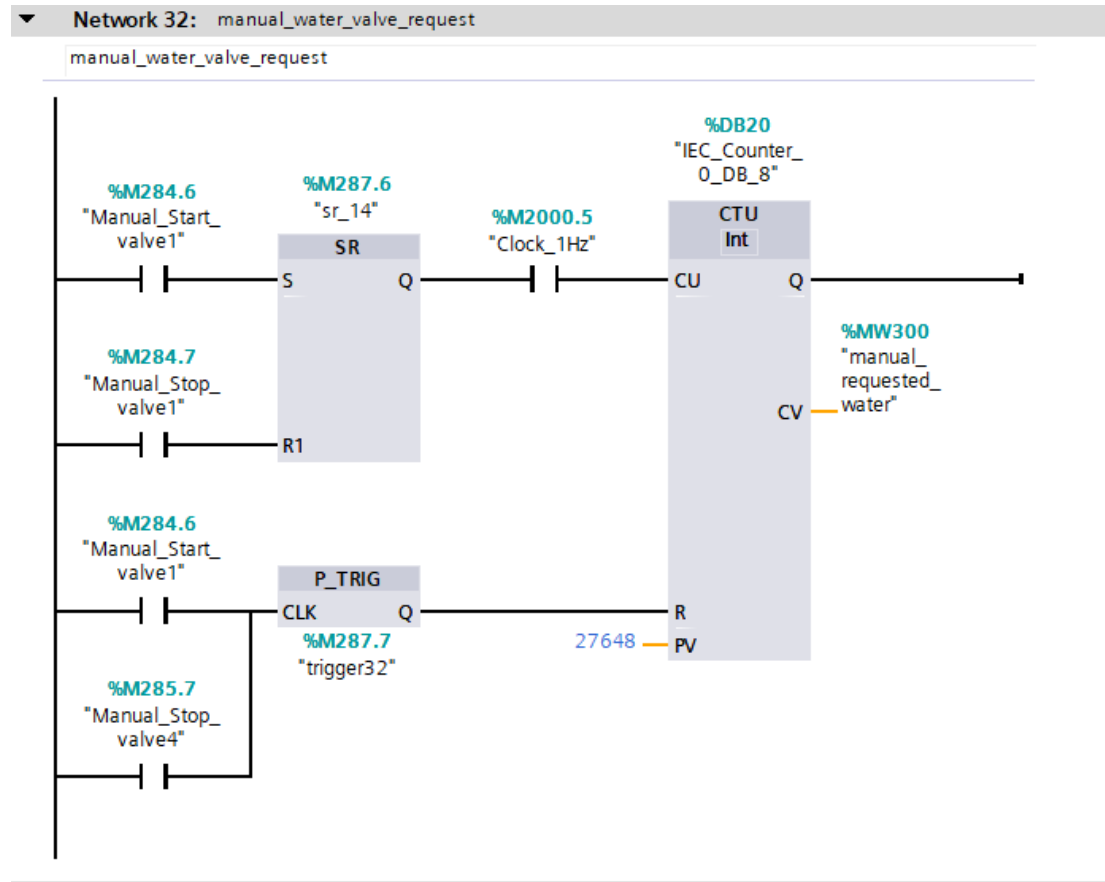
Από το παραπάνω network παρατηρούμε ότι η διαδικασία της εκροής συνεχίζεται κατά την αυτόματη λειτουργία με την βοήθεια του clock memory bit 1Hz (μικρότερη από την διαδικασία πλήρωσης 5Hz) για όσο διάστημα είναι ενεργοποιημένη η ηλεκτροβάννα 1 και η % ποσότητα του υγρού(MW52) της δεξαμενής είναι μεγαλύτερη από την % διαθέσιμη ποσότητα του υγρού που υπολογίζεται με την βοήθεια του network 7. Αντίστοιχα, κατά την χειροκίνητη λειτουργία η διαδικασία της εκροής συνεχίζεται με την βοήθεια του clock memory bit 1Hz για όσο διάστημα είναι ενεργοποιημένο το SR f/f (M6.2) και η ηλεκτροβάννα 1 είναι επίσης ενεργοποιημένη.



Εικόνα 3.18: Βαθμονόμηση εισόδου και υπολογισμοί του % ποσοστού του νερού στην δεξαμενή και της % εναπομείνουσας ποσότητας νερού μετά την συνταγή

Στο συγκεκριμένο network πραγματοποιείται η βαθμονόμηση της αναλογικής εισόδου του αισθητηρίου προκειμένου να προσομοιωθεί η όλη διαδικασία πολλαπλασιάζοντας, κατά την διαδικασία της πλήρωσης της δεξαμενής, την current value του counter up του network 5 με την τιμή 276 έτσι ώστε το εύρος τιμών του αρχικά ακατέργαστου σήματος να μην είναι το 0-100 αλλά το 0-27600, αντίστοιχη διαδικασία ακολουθείται και κατά την διαδικασία της εκροής (100-0 → 27600-0). Με διαίρεση της ποσότητας του ακατέργαστου σήματος 0-27600 με το 276 προκύπτει το % ποσοστό του υγρού για την κάθε δεξαμενή. Τέλος, στο συγκεκριμένο network υπολογίζεται η διαθέσιμη ποσότητα του υγρού που παραμένει στην δεξαμενή (MW122) αφαιρώντας από την συνολική ποσότητα του υγρού της δεξαμενής (MW90) την ποσότητα που ζητά ο χρήστης (MW120) κάθε φορά με το πάτημα του button Recipe_Start (M1.1). Η ίδια ακριβώς διαδικασία ακολουθείται και για τις άλλες 2 δεξαμενές που περιέχουν πολτό πορτοκαλιού και πολτό μήλου αντίστοιχα.

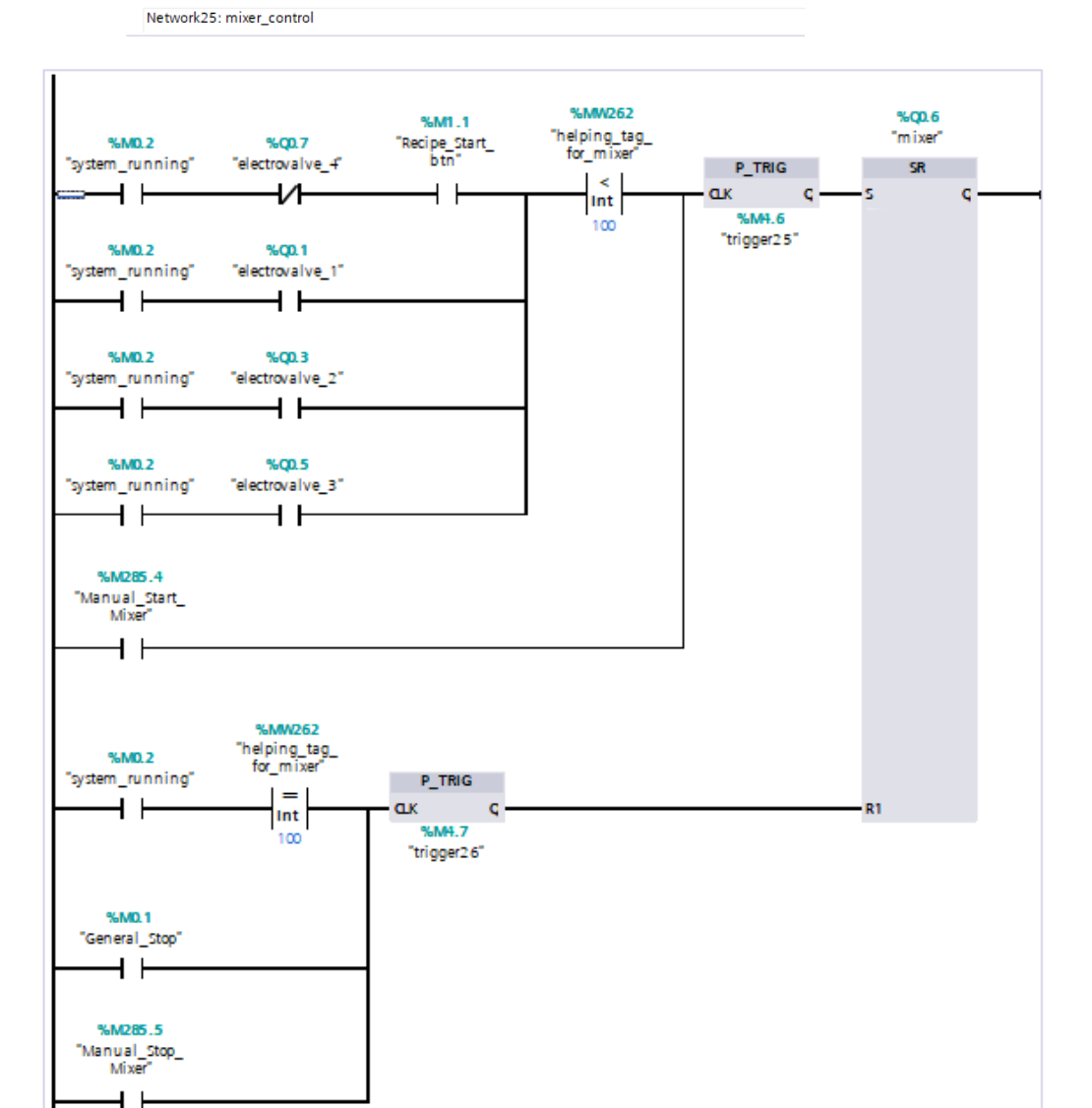
3.4 Χειροκίνητη απαίτηση % ποσότητας νερού



Εικόνα 3.19: Υπολογισμός απαιτούμενης % ποσότητας νερού κατά την χειροκίνητη λειτουργία

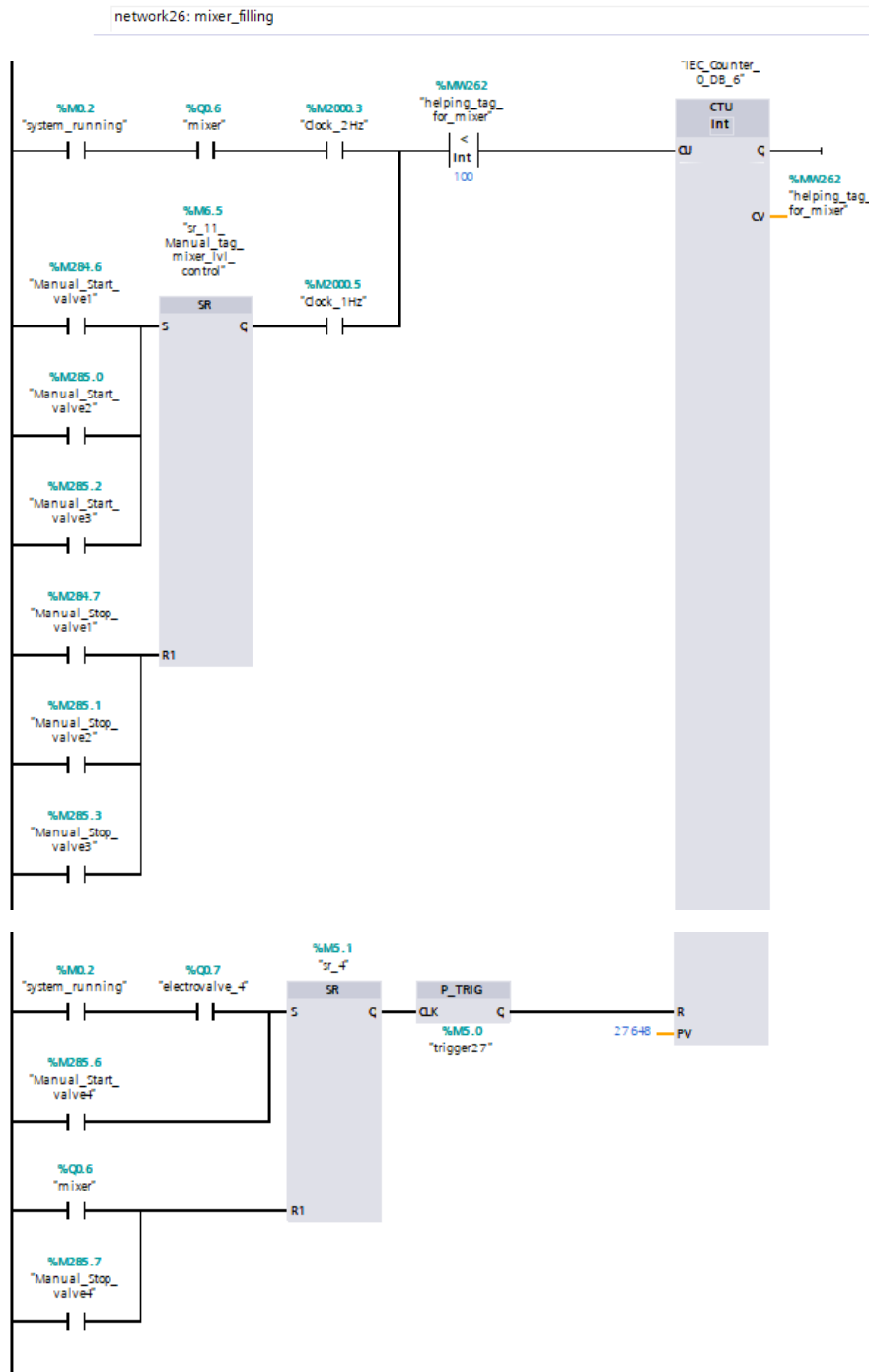
Όπως παρατηρούμε από την παραπάνω εικόνα ο υπολογισμός της απαιτούμενης από τον χειριστή ποσότητας για την πραγματοποίηση μιας συνταγής κατά την χειροκίνητη λειτουργία πραγματοποιείται με το πάτημα του Manual_Start_valve1(M284.6) και με την βοήθεια ενός counter up και του memory clock bit 1Hz υπολογίζεται η απαιτούμενη ποσότητα μέχρι να πατηθεί το Manual_Stop_valve1(M284.7) ενώ πριν την έναρξη του counter πραγματοποιείται μηδενισμός του κάθε φορά με το πάτημα του Manual_Start_valve1 και με την βοήθεια του P_trig(M287.7).Με τον τρόπο αυτό μπορεί να διατηρηθεί μια σταθερή ποσότητα κάθε φορά που να συμμετέχει στην πραγματοποίηση μιας συγκεκριμένης συνταγής.Με ακριβώς τον ίδιο τρόπο πραγματοποιείται ο υπολογισμός των ζητούμενων από τον χειριστή ποσοτήτων και για τις άλλες 2 δεξαμενές.

3.5 Διαδικασία πλήρωσης δεξαμενής αναδευτήρα



Εικόνα 3.20: Έλεγχος αναδευτήρα

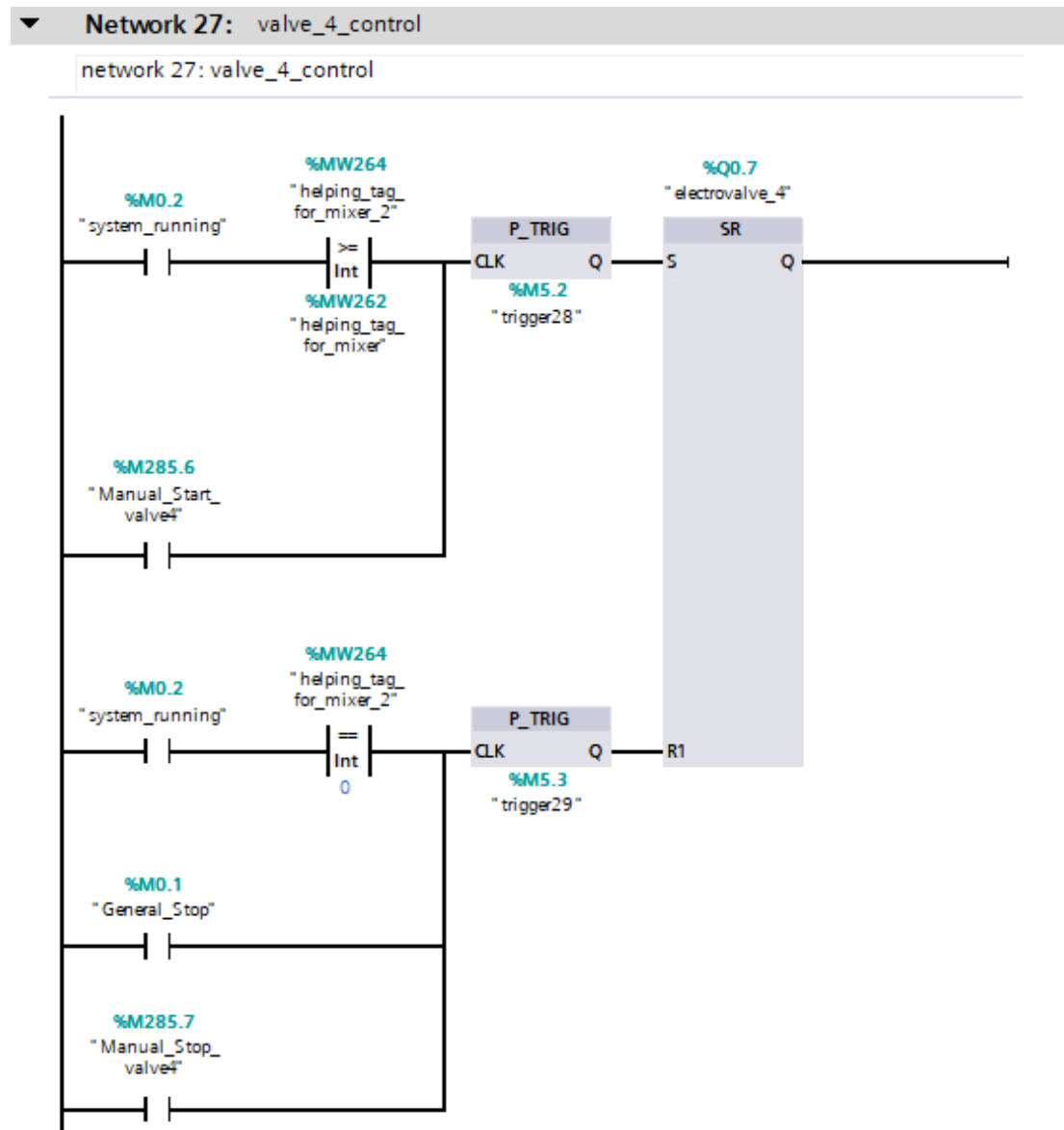
Από την παραπάνω εικόνα προκύπτει ότι κατά την αυτόματη λειτουργία η ενεργοποίηση του αναδευτήρα πραγματοποιείται έχοντας κλειστή την ηλεκτροβάνα της δεξαμενής του με το πάτημα του Recipe_Start(M1.1) και όταν η μεταβλητή MW262, που αντιπροσωπεύει την current value του counter up στο network 26, είναι μικρότερη του 100 ή εναλλακτικά με την ενεργοποίηση οποιασδήποτε από τις 3 ηλεκτροβάνες των δεξαμενών που περιέχουν τα συστατικά για την ανάμιξη. Η απενεργοποίησή του πραγματοποιείται μόλις η τιμή της MW262 γίνει ίση με το 100 ή εναλλακτικά με το πάτημα του general_stop(επαφή M0.1). Κατά την χειροκίνητη λειτουργία, η ενεργοποίηση του αναδευτήρα πραγματοποιείται με το πάτημα του Manual_Start_Mixer(επαφή M285.4) ενώ η απενεργοποίησή του με το πάτημα του Manual_Stop_Mixer(επαφή M285.5).



Εικόνα 3.21: Πλήρωση δεξαμενής μείγματος

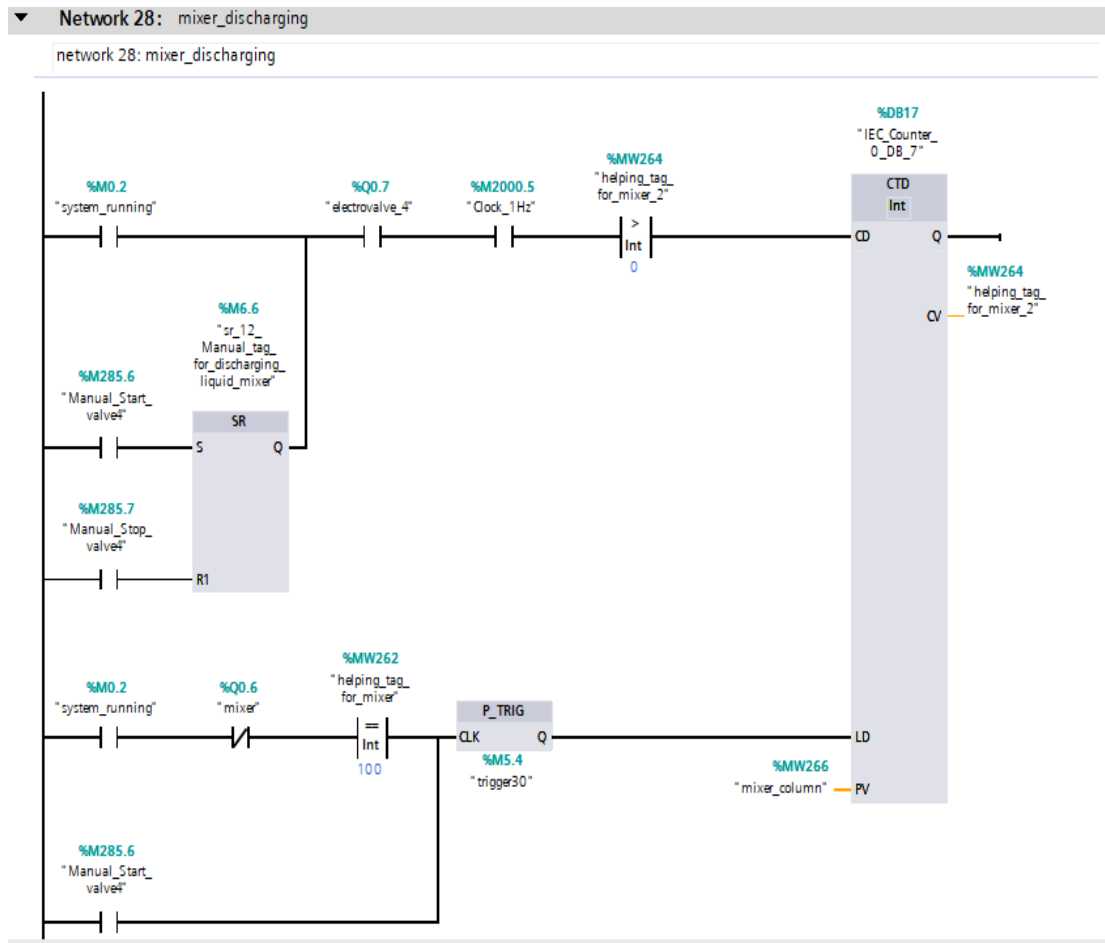
Το network της παραπάνω εικόνας περιγράφει την διαδικασία πλήρωσης της δεξαμενής του αναδευτήρα η οποία κατά την αυτόματη λειτουργία πραγματοποιείται με την βοήθεια του memory clock bit 2Hz όταν ο αναδευτήρας είναι ενεργοποιημένος και όταν η current value του counter up(MW262) είναι μικρότερη του 100 ενώ μηδενισμός του counter πραγματοποιείται με την ενεργοποίηση της ηλεκτροβάνας του αναδευτήρα. Κατα την χειροκίνητη λειτουργία, η διαδικασία πλήρωσης πραγματοποιείται με την χειροκίνητη ενεργοποίηση οποιασδήποτε από τις 3 ηλεκτροβάνες των δεξαμενών που περιέχουν τα συστατικά(επαφές M284.6, M285.0, M285.2) με την βοήθεια του memory clock bit 1Hz και όταν η μεταβλητή MW262 είναι μικρότερη του 100 ενώ μηδενισμός του counter πραγματοποιείται με την χειροκίνητη ενεργοποίηση της ηλεκτροβάνας της δεξαμενής του αναδευτήρα(επαφή M285.6).

3.6 Διαδικασία εκροής μείγματος και βαθμονόμησης της εισόδου της δεξαμενής του αναδευτήρα



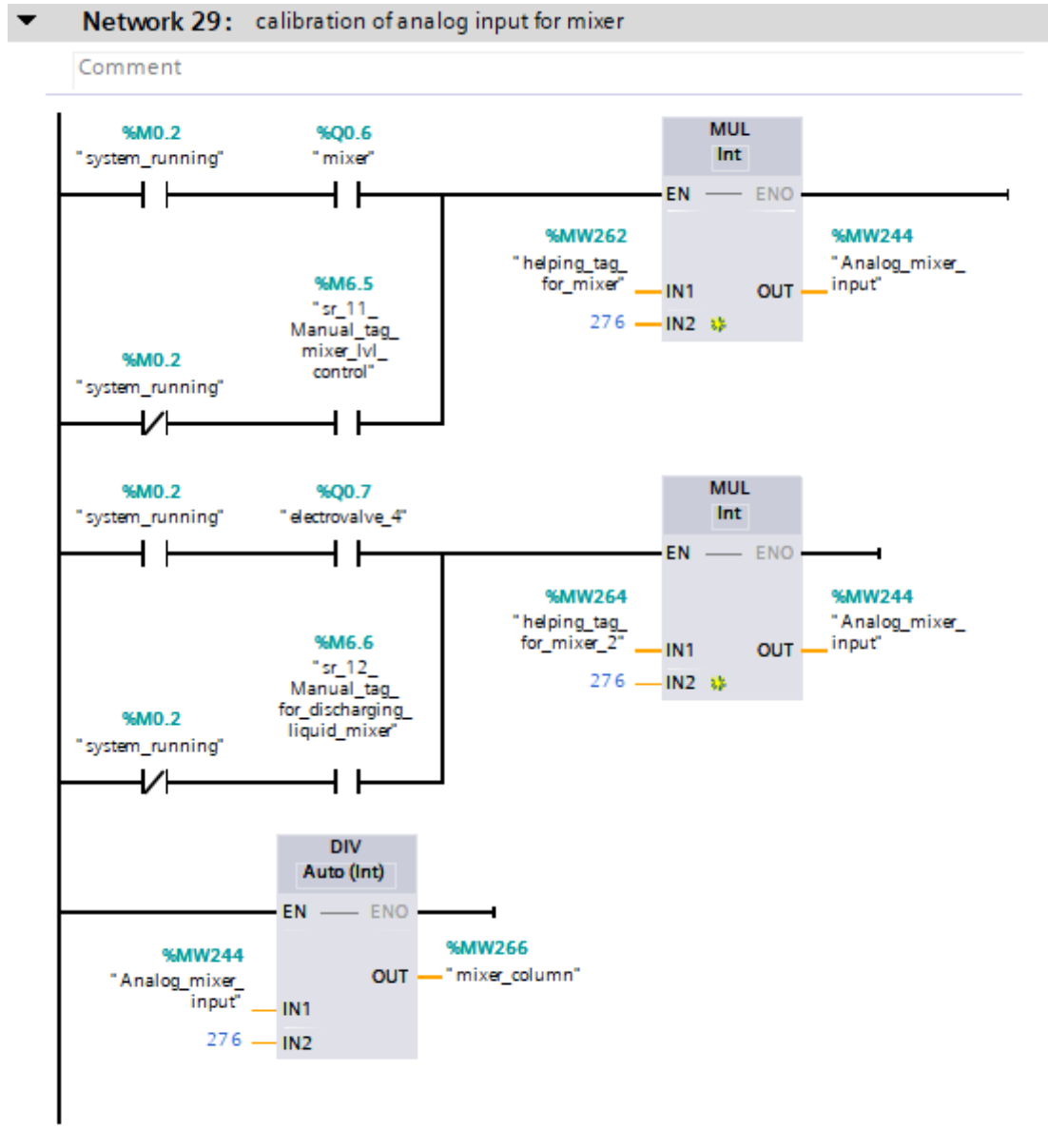
Εικόνα 3.22: Έλεγχος ηλεκτροβάνας αναδευτήρα

Από την παραπάνω εικόνα προκύπτει ότι κατά την αυτόματη λειτουργία η ενεργοποίηση της ηλεκτροβάνας της δεξαμενής του αναδευτήρα πραγματοποιείται όταν η μεταβλητή MW264, που αντιπροσωπεύει την current value του CTD του network 28, είναι μεγαλύτερη ή ίση με την μεταβλητή MW262 του counter up του network 26 ενώ απενεργοποιείται όταν η τιμή της μεταβλητής MW264 γίνει ίση με το 0 ή εναλλακτικά πατηθεί το general stop (επαφή M0.1). Κατά την χειροκίνητη λειτουργία, η ενεργοποίηση της ηλεκτροβάνας του αναδευτήρα πραγματοποιείται με το πάτημα του button Manual_Start_valve4 (επαφή M285.6) ενώ απενεργοποιείται με το πάτημα του button Manual_Stop_valve4 (επαφή M285.7).



Εικόνα 3.23: Εκροή μείγματος από την δεξαμενή του αναδευτήρα

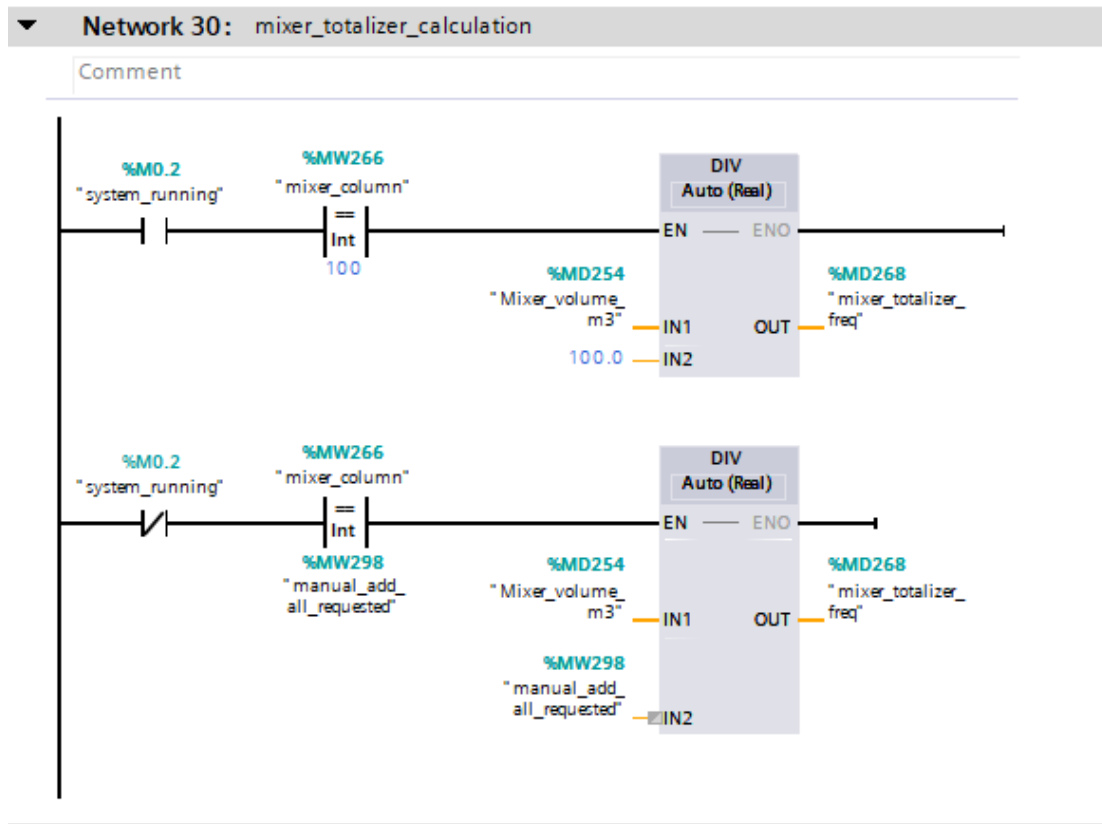
Το παραπάνω network αναπαριστά την διαδικασία εκροής του υγρού από την δεξαμενή του αναδευτήρα. Κατά την αυτόματη λειτουργία, όταν η μεταβλητή MW262 του counter up του network 26 γίνει ίση με το 100 και ο αναδευτήρας είναι απενεργοποιημένος τότε πραγματοποιείται φόρτωση της τιμής του MW266, που αντιπροσωπεύει την % ποσότητα υγρού της δεξαμενής του αναδευτήρα και την δεδομένη στιγμή ταυτίζεται με την MW262 και κατ'έπείταση το 100, στην current value του CTD MW264. Με την βοήθεια του memory clock bit 1Hz και έχοντας ενεργοποιημένη την ηλεκτροβάνια της δεξαμενής του αναδευτήρα ξεκινά η διαδικασία εκροής έχοντας την τιμή 100 για την MW264 και συνεχίζεται μέχρι το 0. Κατά την χειροκίνητη λειτουργία, η διαδικασία εκροής ξεκινά με το πάτημα του Manual_Start_valve4 (επαφή M285.6) και σταματά με το πάτημα του Manual_Stop_valve4 (επαφή M285.7).



Εικόνα 3.24: Βαθμονόμηση εισόδου του αναδευτήρα και υπολογισμός της % ποσότητας του μείγματος στην δεξαμενή

Από το παραπάνω network προκύπτει ότι κατά την αυτόματη λειτουργία και την διαδικασία πλήρωσης της δεξαμενής του αναδευτήρα, κατά την οποία η λειτουργία του αναδευτήρα είναι πάντα ενεργοποιημένη, η βαθμονόμηση της αναλογικής εισόδου της δεξαμενής του αναδευτήρα πραγματοποιείται με την current value MW262 του counter up του network 26 ενώ κατά την διαδικασία της εκροής της, κατά την διάρκεια της οποίας είναι πάντα ενεργοποιημένη η ηλεκτροβάννα του αναδευτήρα, η βαθμονόμηση της αναλογικής εισόδου της δεξαμενής πραγματοποιείται με την βοήθεια της current value του CTD του network 28. Τέλος, με διαίρεση της αναλογικής εισόδου της δεξαμενής του αναδευτήρα με το 276 προκύπτει η % ποσότητα του μείγματος που περιέχεται στην δεξαμενή του αναδευτήρα.

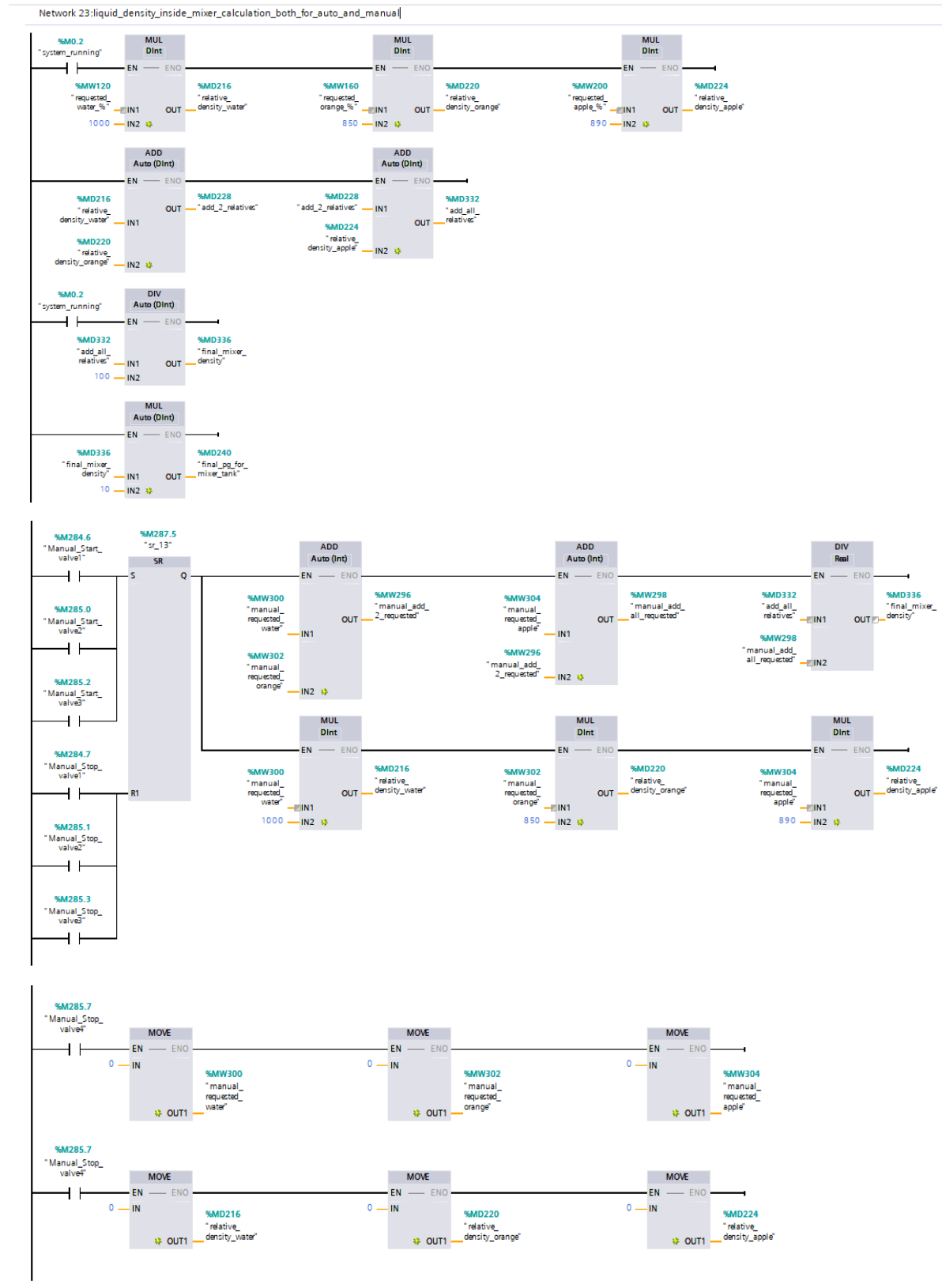
3.7 Υπολογισμός βήματος για την καταμέτρηση της αθροιστικής ροής στην δεξαμενή του αναδευτήρα



Εικόνα 3.25: Υπολογισμός βήματος για την καταμέτρηση της αθροιστικής ροής στην δεξαμενή του αναδευτήρα

Στην παραπάνω εικόνα παρατηρούμε ότι όταν βρισκόμαστε στην αυτόματη λειτουργία μόλις η ποσότητα της δεξαμενής του μείγματος φτάσει στο 100% τότε πραγματοποιείται ο υπολογισμός του σταθερού βήματος που χρησιμοποιείται για τον υπολογισμό της αθροιστικής ροής του μείγματος ενώ αντίθετα κατά την χειροκίνητη λειτουργία ο υπολογισμός αυτός πραγματοποιείται για οποιοδήποτε ποσοστό % του υγρού της δεξαμενής του αναδευτήρα.

3.8 Υπολογισμός της πυκνότητας του μείγματος



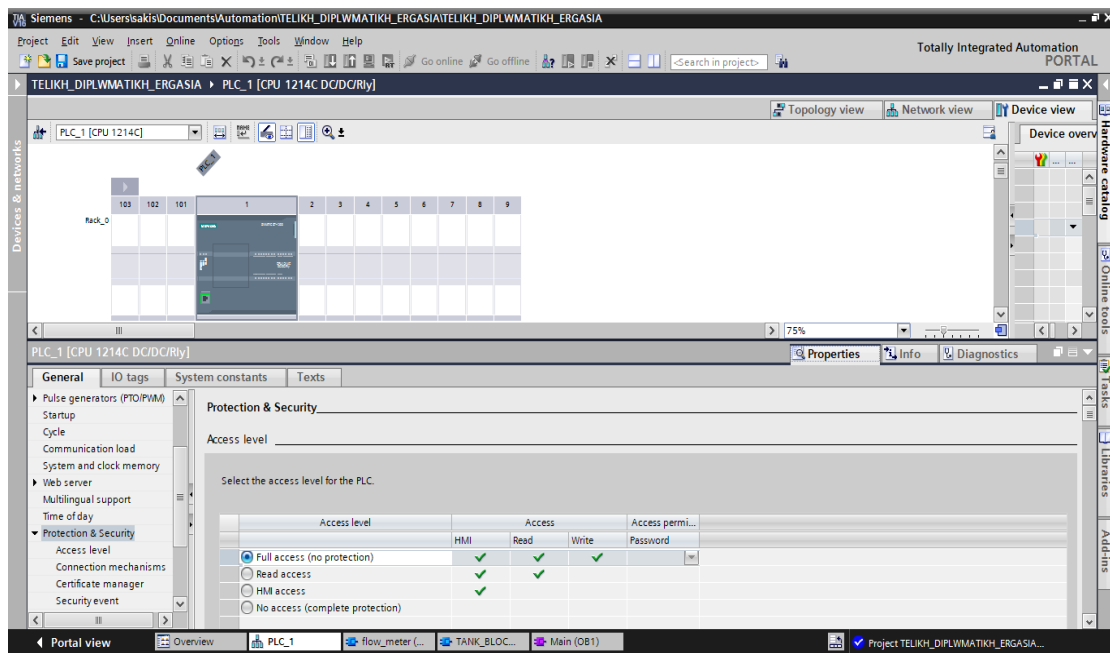
Εικόνα 3.26: Υπολογισμός πυκνότητας δεξαμενής αναδευτήρα

Στο συγκεκριμένο network πραγματοποιείται ο υπολογισμός της πυκνότητας του υγρού της δεξαμενής του αναδευτήρα, τόσο για την χειροκίνητη όσο και για την αυτόματη λειτουργία,

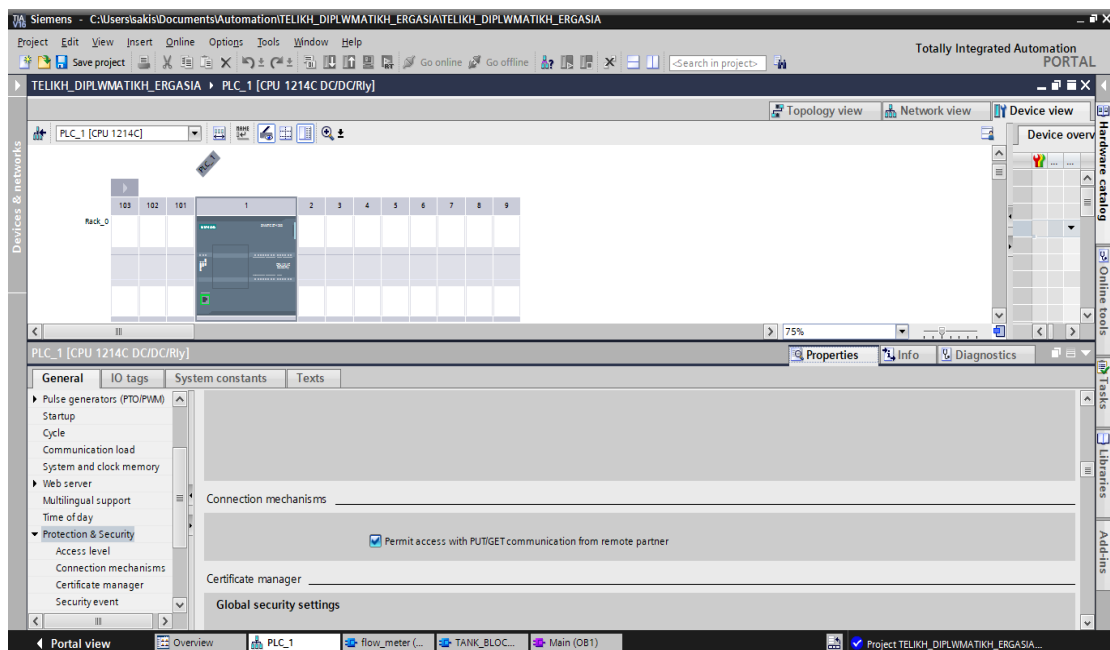
που όπως έχει αναφερθεί προκύπτει από τον σταθμισμένο μέσο όρο των πυκνοτήτων των υγρών που αποτελούν τα συστατικά του μείγματος και η τιμή της διαμορφώνεται ανάλογα με την συνεισφορά του κάθε συστατικού στην πραγματοποίηση της εκάστοτε συνταγής. Έπειτα από την πραγματοποίηση όλων των network (παράρτημα Α) ολοκληρώνεται ο προγραμματισμός του PLC.

3.9 Προσβασιμότητα του PLC από το SCADA

Το επόμενο βήμα που ακολουθεί είναι η ρύθμιση για την προσβασιμότητα του PLC από το custom SCADA. Έχοντας επιλεγμένο το Plc επιλέγουμε την καρτέλα Properties, στη συνέχεια την καρτέλα General και έπειτα την επιλογή Protection & Security και πραγματοποιούμε τις ρυθμίσεις που παρουσιάζονται στις παρακάτω εικόνες, επιλέγοντας ουσιαστικά Full access(no protection) και Permit access with PUT/GET connection from remote partner:



Εικόνα 3.27: Ρυθμίσεις προσβασιμότητας του plc από το scada

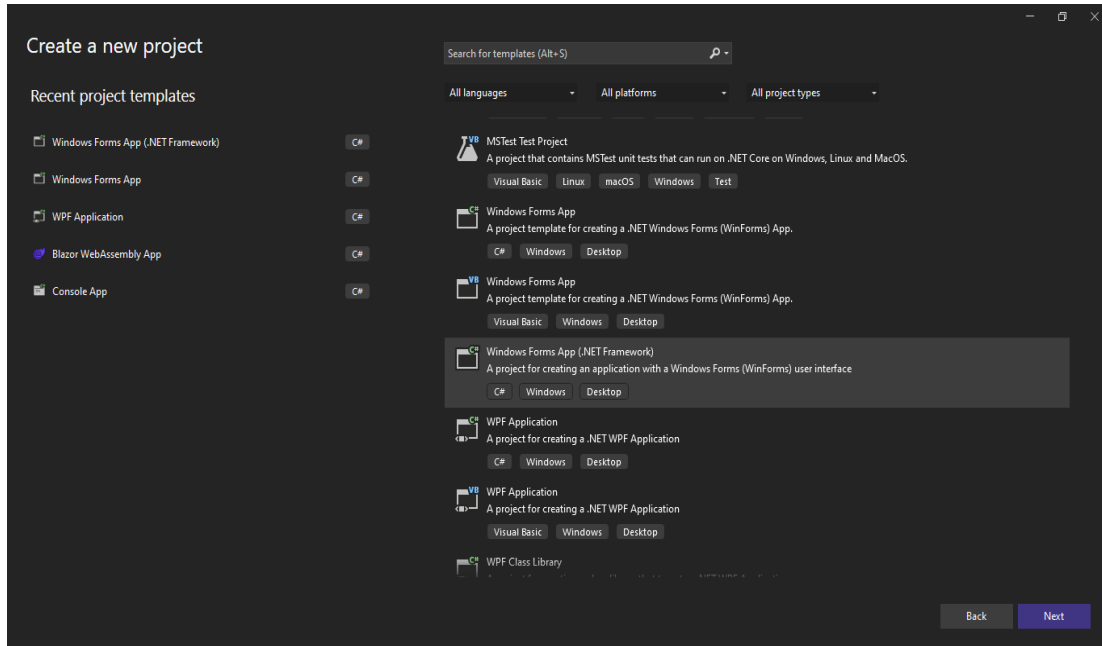


Εικόνα 3.28: Ρυθμίσεις προσβασιμότητας του plc από το scada

Κεφάλαιο 4ο: Προγραμματισμός του SCADA

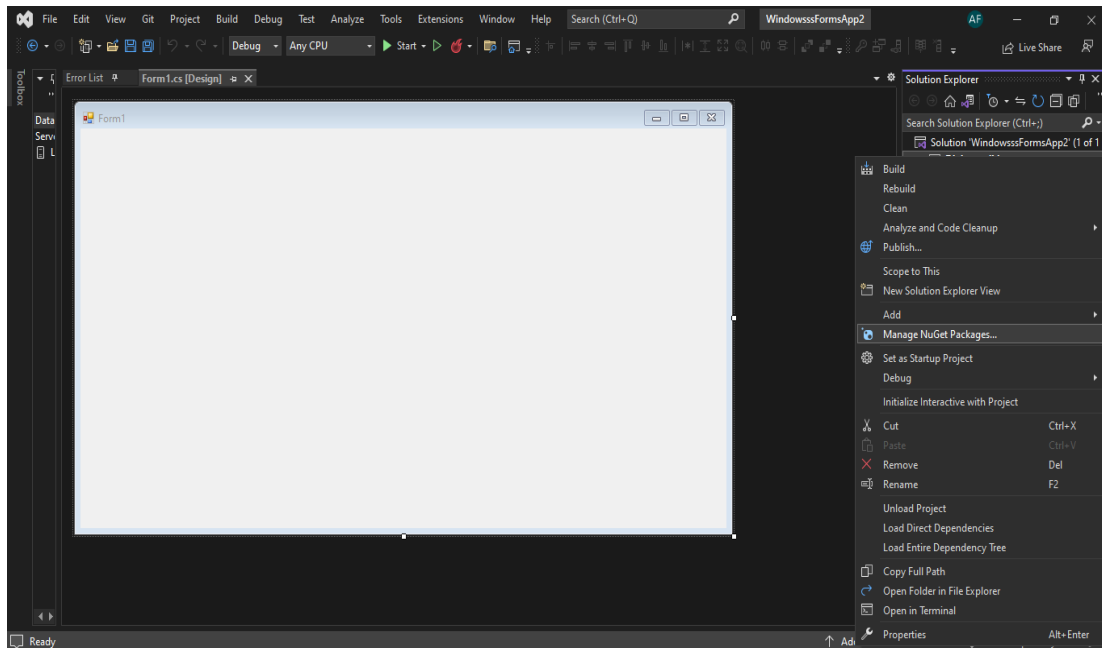
4.1 Εγκατάσταση βιβλιοθηκών S7.Net και Microsoft.Data.SqlClient

Για τον προγραμματισμό του SCADA χρησιμοποιήθηκε η γλώσσα προγραμματισμού C# μέσω της πλατφόρμας Visual Studio 2022. Η επικοινωνία με το Plc πραγματοποιήθηκε με την βοήθεια της βιβλιοθήκης S7.Net η οποία είναι κατάλληλη για βιομηχανικούς ελεγκτές της εταιρείας SIEMENS. Το project που δημιουργήθηκε για την συγκεκριμένη εφαρμογή πραγματοποιήθηκε με την επιλογή της φόρμας εφαρμογής Windows σε .NET framework όπως παρουσιάζεται παρακάτω:



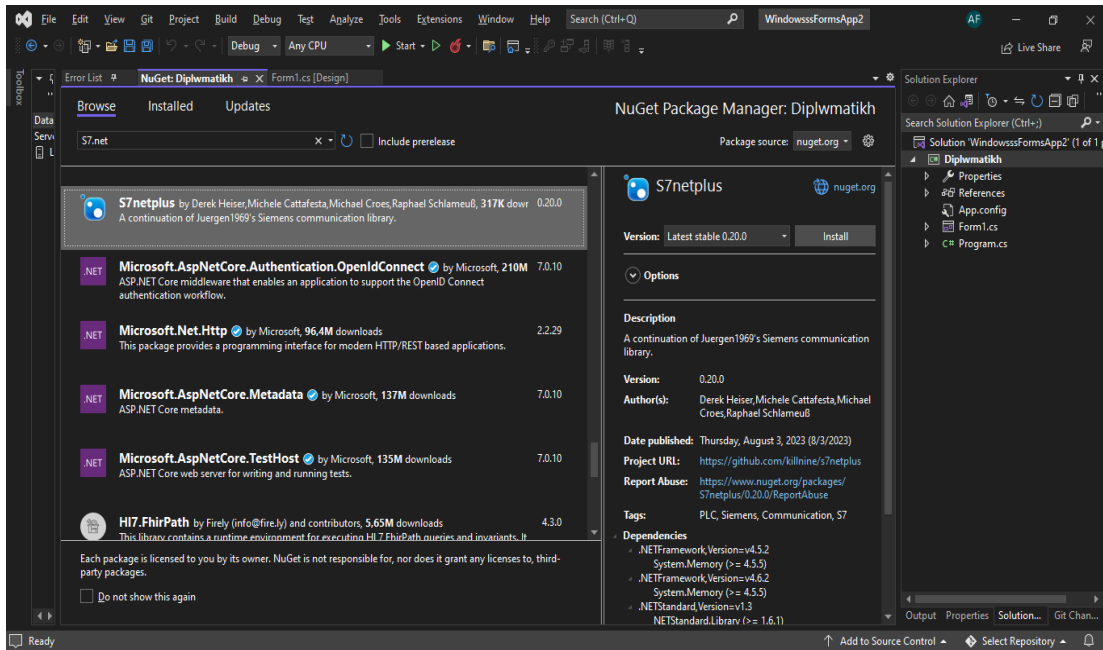
Εικόνα 4.1: Επιλογή κατάλληλης φόρμας εφαρμογής

Στη συνέχεια κάνοντας δεξί κλικ στο πεδίο του Solution Explorer επιλέγουμε Manage NuGet Packages όπως παρουσιάζεται στην παρακάτω εικόνα:



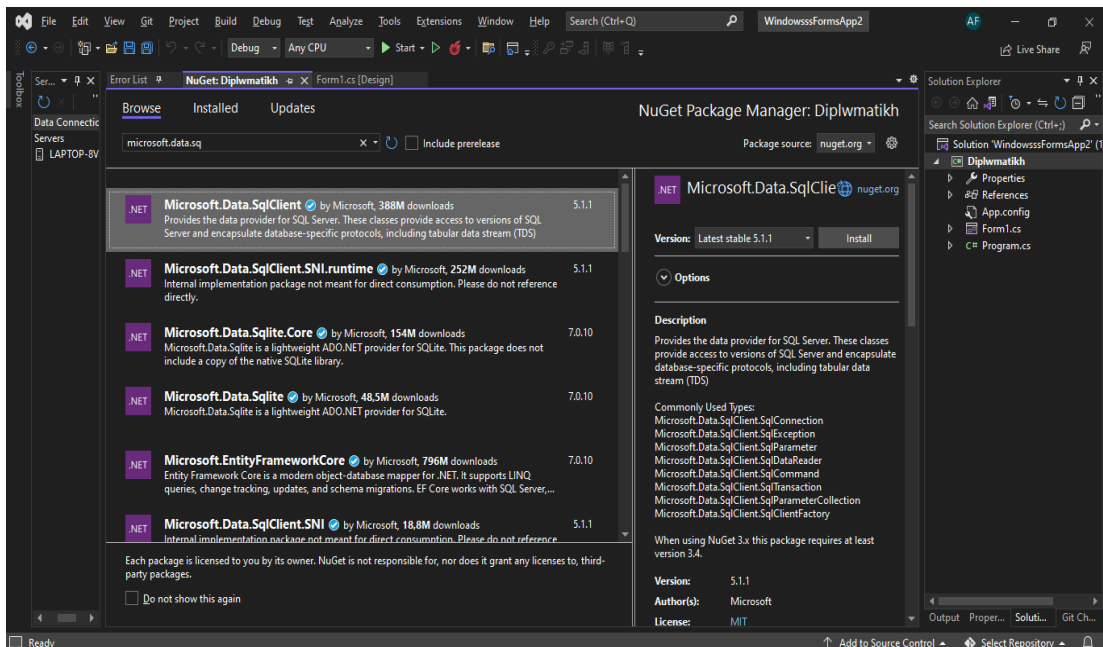
Εικόνα 4.2: Επιλογή και εγκατάσταση βιβλιοθήκης S7.Net

Στην οθόνη που εμφανίζεται επιλέγουμε Browse και στο πεδίο αναζήτησης γράφουμε το όνομα της βιβλιοθήκης που θέλουμε να κατεβάσουμε για την εφαρμογή μας. Πληκτρολογώντας λοιπόν S7.Net επιλέγουμε την S7netplus και την εγκαθιστούμε:



Εικόνα 4.3: Επιλογή και εγκατάσταση βιβλιοθήκης S7.Net

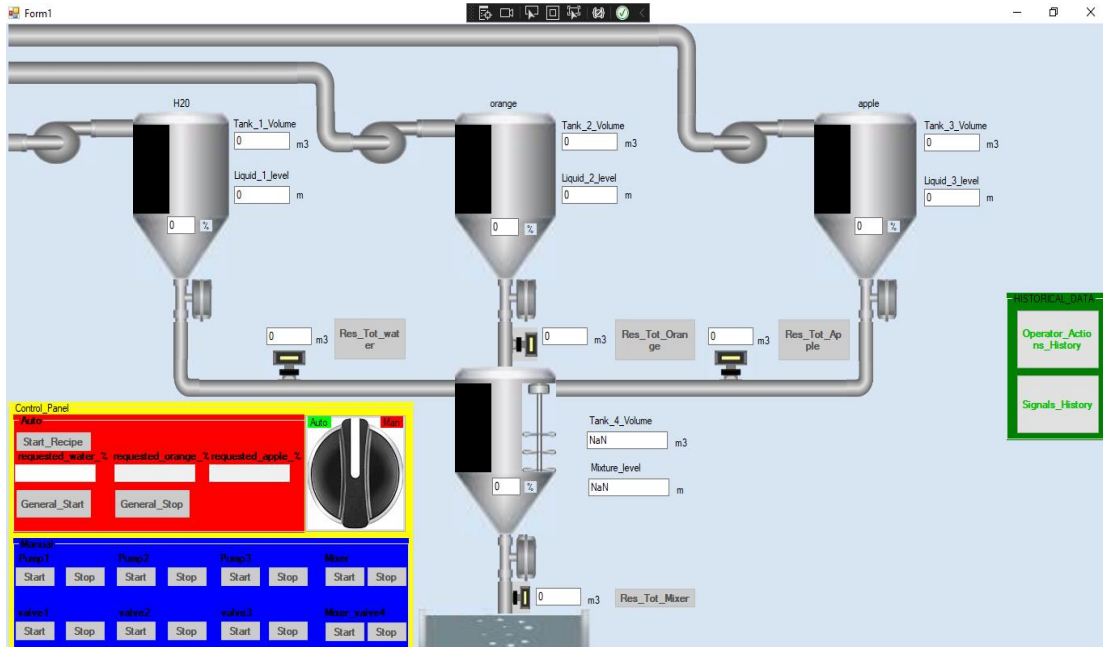
Με τον ίδιο ακριβώς τρόπο εγκαταστάθηκε και η βιβλιοθήκη Microsoft.Data.SqlClient η οποία είναι απαραίτητη για την σύνδεση του SCADA με τον SQL server και την αποθήκευση σε αυτόν όλων των σημάτων της διάταξης αλλά και όλων των ενεργειών του χρήστη:



Εικόνα 4.4: Επιλογή και εγκατάσταση βιβλιοθήκης Microsoft.Data.SqlClient

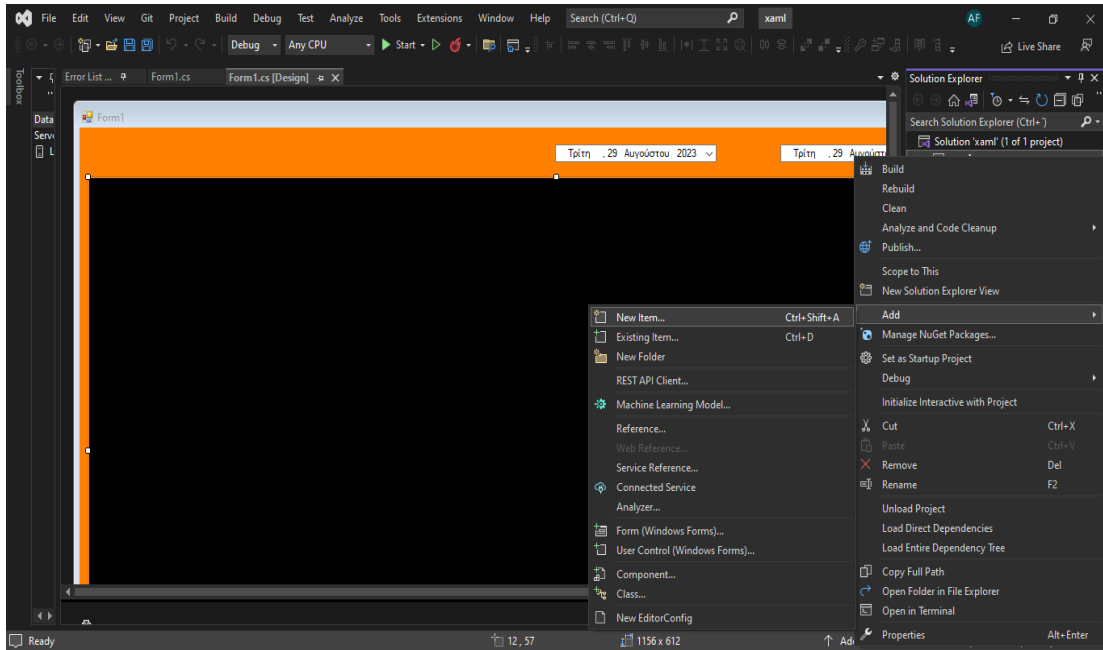
4.2 Δημιουργία αρχείου xaml

Στο παράθυρο Form1.cs[Design] σχεδιάστηκε το front-end τμήμα του Scada ενώ στο παράθυρο Form1.cs γράφτηκε ο απαραίτητος κώδικας σε γλώσσα C# για την σωστή λειτουργία του(back-end).Η τελική μορφή του Scada παρουσιάζεται στην παρακάτω εικόνα:

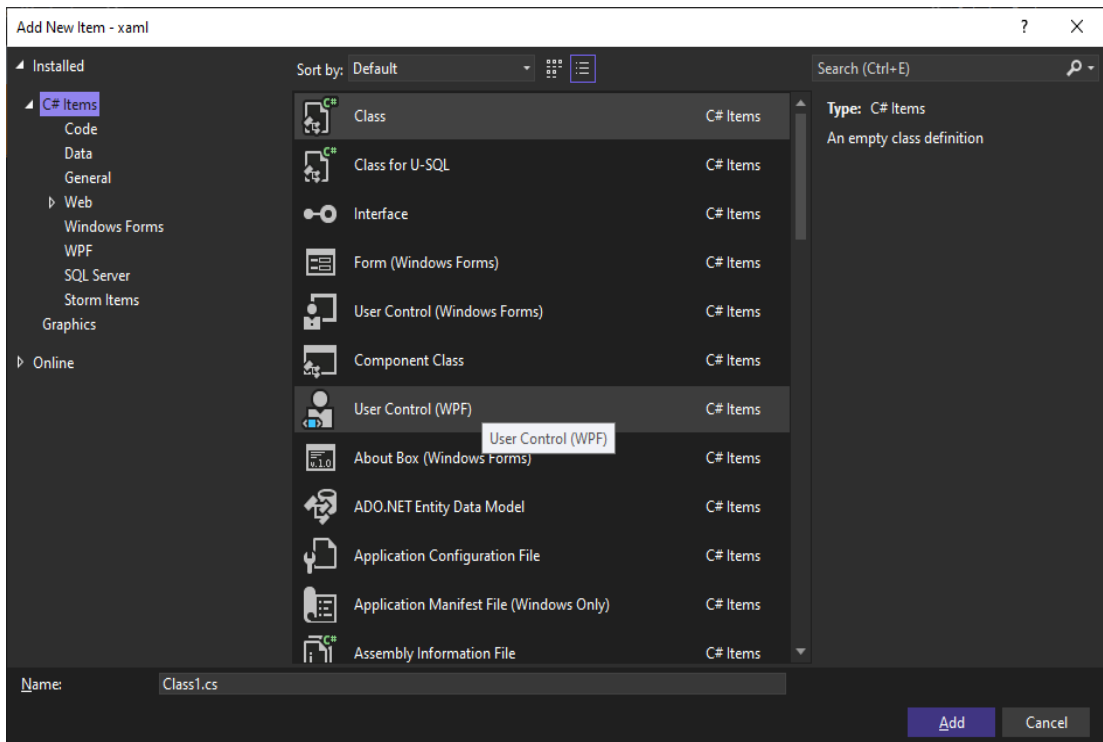


Εικόνα 4.5: Custom Scada

Όπως παρατηρούμε από την παραπάνω εικόνα κάθε δεξαμενή περιέχει μια level bar για την αναπαράσταση του επιπέδου του υγρού στο εσωτερικό της. Πρόκειται για ένα αρχείο τύπου xaml. Η xaml είναι μια γλώσσα βασισμένη σε xml και παρόμοια με το πως τα αρχεία html χρησιμοποιούνται για την αναπαραγωγή ιστοσελίδων τα αρχεία xaml περιγράφουν στοιχεία διεπαφής χρήστη σε εφαρμογές λογισμικού. Ενώ το περιεχόμενο xaml μπορεί να εκφραστεί σε άλλες γλώσσες όπως η C# το xaml δεν χρειάζεται να καταρτιστεί καθώς βασίζεται σε xml και είναι ευκολότερο για τους προγραμματιστές να δουλέψουν μαζί του. Για την δημιουργία του κάνουμε δεξί κλικ στο πεδίο του Solution Explorer και έπειτα επιλέγουμε Add , New Item και User Control (WPF) όπως φαίνεται στις παρακάτω εικόνες:

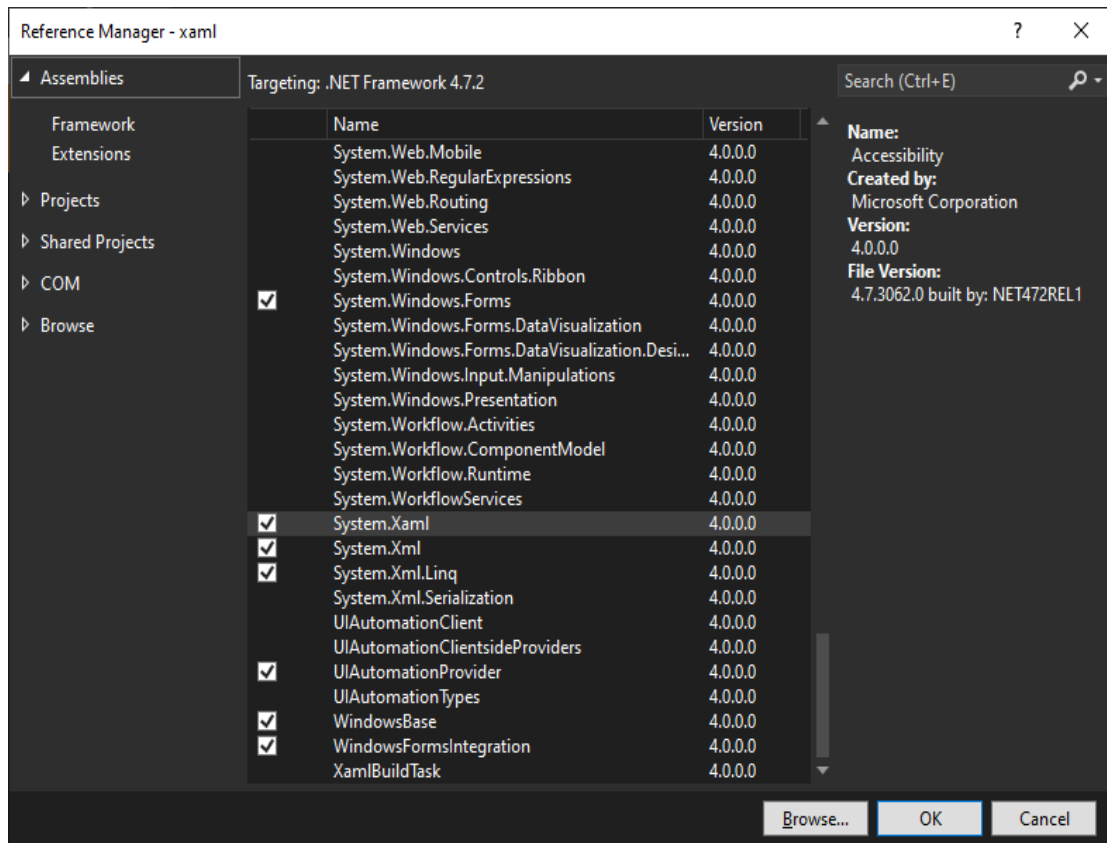


Εικόνα 4.6: Προσθήκη αρχείου xaml



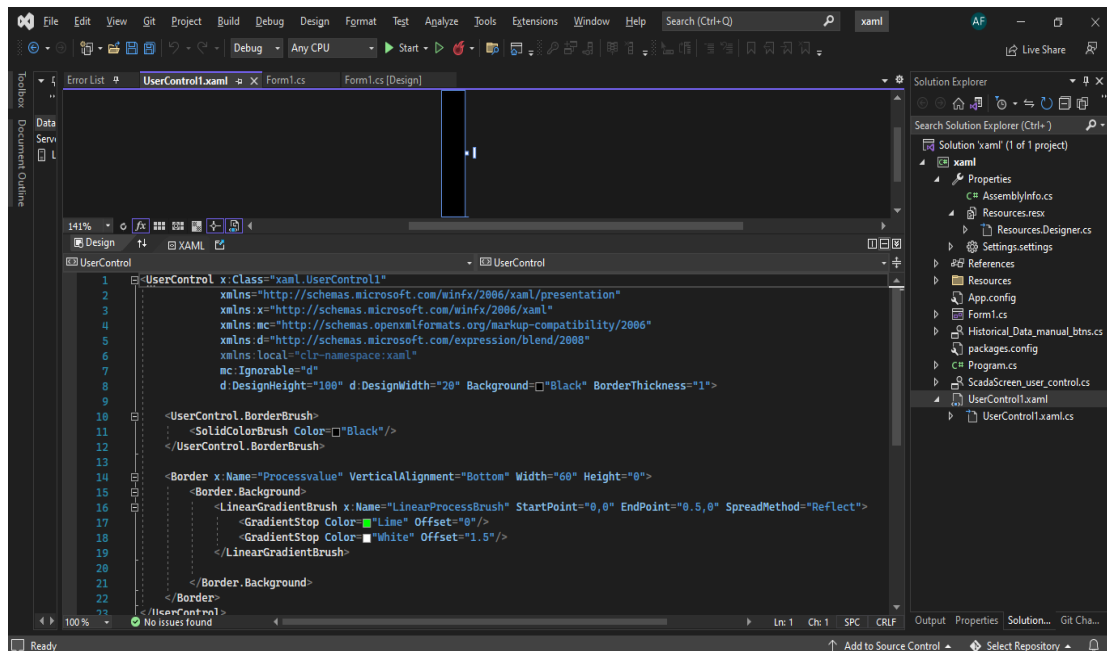
Εικόνα 4.7: Προσθήκη αρχείου xaml

Στη συνέχεια κάνοντας δεξί κλικ στα References του Solution Explorer επιλέγουμε Add Reference και στο παράθυρο διαλόγου του Reference Manager επιλέγουμε την αρχικά μη επιλεγμένη επιλογή System.Xaml όπως παρουσιάζεται παρακάτω:



Εικόνα 4.8: Απαραίτητη ρύθμιση για την προσθήκη του αρχείου xaml

Έπειτα προχωρούμε στην εγγραφή του κώδικα σε γλώσσα xaml για το αρχείο μας:



Εικόνα 4.9: Κώδικας αρχείου xaml

Ο κώδικας στην C# που γράφτηκε για το περιεχόμενο του xaml παρουσιάζεται παρακάτω:

```

5  using System.Threading.Tasks;
6  using System.Windows;
7  using System.Windows.Controls;
8  using System.Windows.Data;
9  using System.Windows.Documents;
10 using System.Windows.Input;
11 using System.Windows.Media;
12 using System.Windows.Media.Imaging;
13 using System.Windows.Navigation;
14 using System.Windows.Shapes;
15
16 namespace xaml
17 {
18     /// <summary>
19     /// Interaction logic for UserControl1.xaml
20     /// </summary>
21     public partial class UserControl1 : UserControl
22     {
23         public int value;
24         public UserControl1()
25         {
26             InitializeComponent();
27         }
28         public void changeValue()
29         {
30             Processvalue.Height= value;
31         }
32     }
33 }
34
35

```

Εικόνα 4.10: Κώδικας περιεχομένου αρχείου xaml

4.3 Δημιουργία tables στην SQL βάση δεδομένων

Όσον αφορά την δυνατότητα του χρήστη να διαχειρίζεται τα ιστορικά στοιχεία της λειτουργίας της διάταξης χρησιμοποιήθηκαν ο Microsoft SQL Server και το Microsoft SQL Management Studio 18 στο οποίο και πραγματοποιήθηκαν τα ακόλουθα tables(ένα για την αποθήκευση των ενεργειών του χρήστη με το όνομα Actions_Data και ένα για την αποθήκευση των σημάτων της διάταξης με το όνομα Signals_Data):

```

CREATE TABLE Actions_Data(
[id] int IDENTITY(1,1) primary key NOT NULL,
[Auto] datetime NULL,
[Manual] datetime NULL,
[Buttons_Off] datetime NULL,
[start_pump1] datetime NULL,
[stop_pump1] datetime NULL,
[start_pump2] datetime NULL,
[stop_pump2] datetime NULL,
[start_pump3] datetime NULL,
[stop_pump3] datetime NULL,
[start_valve1] datetime NULL,
[stop_valve1] datetime NULL,
[start_valve2] datetime NULL,
[stop_valve2] datetime NULL,
[start_valve3] datetime NULL,
[stop_valve3] datetime NULL,
[start_mixer] datetime NULL,
[stop_mixer] datetime NULL,
[start_mixer_valve4] datetime NULL,
[stop_mixer_valve4] datetime NULL,
[General_Start] datetime NULL,
[General_Stop] datetime NULL,
[Recipe_Start] datetime NULL,

```

```
[requested_water] varchar(32) NULL,
[requested_orange] varchar(32) NULL,
[requested_apple] varchar(32) NULL,
[Reset_Tot_water] datetime NULL,
[Reset_Tot_orange] datetime NULL,
[Reset_Tot_apple] datetime NULL,
[Reset_Tot_mixer] datetime NULL,
)
```

CREATE TABLE Signals_Data

```
(
[id] [int] IDENTITY(1,1) primary key NOT NULL,
[Date_Time] datetime NULL,
[tank1_volume] varchar(32) NULL,
[tank1_volpercent] varchar(32) NULL,
[tank1_height] varchar(32) NULL,
[tank1_flow] varchar(32) NULL,
[tank2_volume] varchar(32) NULL,
[tank2_volpercent] varchar(32) NULL,
[tank2_height] varchar(32) NULL,
[tank2_flow] varchar(32) NULL,
[tank3_volume] varchar(32) NULL,
[tank3_volpercent] varchar(32) NULL,
[tank3_height] varchar(32) NULL,
[tank3_flow] varchar(32) NULL,
[tank4_volume] varchar(32) NULL,
[tank4_volpercent] varchar(32) NULL,
[tank4_height] varchar(32) NULL,
[tank4_flow] varchar(32) NULL,
[pump1] varchar(32) NULL,
[pump2] varchar(32) NULL,
[pump3] varchar(32) NULL,
[mixer] varchar(32) NULL,
[valve1] varchar(32) NULL,
[valve2] varchar(32) NULL,
[valve3] varchar(32) NULL,
[valve4] varchar(32) NULL)
```

4.4 Ανάλυση κώδικα custom SCADA

Στη συνέχεια παρατίθενται ορισμένα κομμάτια του κώδικα του custom scada προκειμένου να αναλυθεί η λειτουργία τους:

```
private void selector_btn_Click(object sender, EventArgs e)
{
    pos += 1;
    if (pos == 1)
    {
        Bitmap b1 = new Bitmap(@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ
ΔΙΠΛΑΕ\2.jpg");
        selector_btn.Image = b1;

        using (var plc = new Plc(CpuType.S71200, "127.0.0.1", 0, 1))
        {
            plc.Open();
            plc.Write("M288.4", true);
        }
        string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO Actions_Data(Auto) VALUES(@Auto)";
        SqlConnection con = new SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var AutoParameter = new SqlParameter("Auto",
System.Data.SqlDbType.DateTime);
        AutoParameter.Value = DateTime.Now;
        cmd.Parameters.Add(AutoParameter);
        cmd.ExecuteNonQuery();
        con.Close();

        Manual_Start_pump1.Enabled = false;
        Manual_Stop_pump1.Enabled = false;
        Manual_Start_pump2.Enabled = false;
        Manual_Stop_pump2.Enabled = false;
        Manual_Start_pump3.Enabled = false;
        Manual_Stop_pump3.Enabled = false;
        Manual_Start_mixer.Enabled = false;
        Manual_Stop_mixer.Enabled = false;
        Manual_Start_valve1.Enabled = false;
        Manual_Stop_valve1.Enabled = false;
        Manual_Start_valve2.Enabled = false;
        Manual_Stop_valve2.Enabled = false;
        Manual_Start_valve3.Enabled = false;
        Manual_Stop_valve3.Enabled = false;
        Manual_Start_valve4.Enabled = false;
        Manual_Stop_valve4.Enabled = false;
        Recipe_btn.Enabled = true;
        General_Start.Enabled = true;
        General_Stop.Enabled = false;
        requested_water.Enabled = false;
        requested_orange.Enabled = false;
        requested_apple.Enabled = false;
    }
}
```

```

Res_Tot_water.Enabled = true;
Res_Tot_Orange.Enabled = true;
Reset_Tot_Apple.Enabled = true;
Res_Tot_Mixer.Enabled = true;
Operator_Actions_History.Enabled = false;
Signals_History.Enabled = false;
Auto_Control_Panel.Enabled = true;
Manual_Control_Panel.Enabled = false;
GroupBox_Historical_Data.Enabled = false;
}

```

Το παραπάνω κομμάτι κώδικα περιγράφει την λειτουργία του επιλογικού διακόπτη που χρησιμοποιείται στο Scada και συγκεκριμένα της Auto λειτουργίας. Όταν το button του περιστροφικού διακόπτη πατηθεί πρώτη φορά τότε από την μεσαία θέση που βρίσκεται αρχικά κινείται προς τα αριστερά με την εμφάνιση της αντίστοιχης εικόνας. Έπειτα με την βοήθεια του κώδικα:

```

        using (var plc = new Plc(CpuType.S71200, "127.0.0.1", 0, 1))
        {
            plc.Open();
            plc.Write("M288.4", true);
        }

```

ανοίγει η επικοινωνία με το plc και εγγράφεται στην επαφή "M288.4" η τιμή <<1>> ενεργοποιώντας με τον τρόπο αυτό την αυτόματη λειτουργία της διάταξης. Έπειτα, όπως παρατηρούμε με την βοήθεια ενός connection string πραγματοποιήθηκε η σύνδεση με την βάση δεδομένων ώστε να αποθηκεύεται εκεί κάθε φορά που ο χρήστης επιλέγει την αυτόματη λειτουργία. Επιπλέον, παρατηρούμε ότι για το στοιχείο Auto, του πίνακα Actions_Data που δημιουργήθηκε με την βοήθεια της γλώσσας SQL, πραγματοποιήθηκε παραμετροποίηση προκειμένου να αποφευχθεί οποιαδήποτε απειλή από injection (το ίδιο έχει συμβεί για όλα τα στοιχεία και των 2 πινάκων που δημιουργήθηκαν στην βάση δεδομένων). Το στοιχείο Auto δηλώθηκε ως τύπου datetime έτσι ώστε κάθε φορά που ο χειριστής επιλέγει την αυτόματη λειτουργία να καταγράφεται αυτόματα στην βάση δεδομένων η ακριβής ημερομηνία και ώρα που επιλέχθηκε να πατηθεί. Τέλος, με την επιλογή της αυτόματης λειτουργίας από τον περιστροφικό διακόπτη πραγματοποιείται απενεργοποίηση όλων των button με εξαίρεση τα button που μηδενίζουν τους μετρητές αθροιστικής ροής αλλά και τα button General_Start και Recipe_Start τα οποία ενεργοποιούνται προκειμένου να είναι εφικτή η πλήρωση των δεξαμενών από τις αντλίες και η πραγματοποίηση συνταγών. Με τον ίδιο ακριβώς τρόπο συντάχθηκε ο κώδικας και για τις άλλες 2 επιλογές του διακόπτη (χειροκίνητη λειτουργία και προβολή ιστορικών στοιχείων).

```

private void General_Start_Click(object sender, EventArgs e)
{
    using (var plc = new Plc(CpuType.S71200, "127.0.0.1", 0, 1))
    {
        plc.Open();
        plc.Write("M0.0", true);
        plc.Write("M0.0", false);
    }
}

```

```

        string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO Actions_Data(General_Start)
VALUES(@General_Start)";
        SqlConnection con = new SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var General_StartParameter = new SqlParameter("General_Start",
System.Data.SqlDbType.DateTime);
        General_StartParameter.Value = DateTime.Now;
        cmd.Parameters.Add(General_StartParameter);
        cmd.ExecuteNonQuery();
        con.Close();

        Recipe_btn.Enabled = true;
        requested_water.Enabled = true;
        requested_orange.Enabled = true;
        requested_apple.Enabled = true;
        selector_btn.Enabled = false;
        General_Start.Enabled = false;
        General_Stop.Enabled = true;

    }

```

Ο παραπάνω κώδικας περιγράφει την λειτουργία του button General_Start όταν βρισκόμαστε στην αυτόματη λειτουργία της διάταξής μας. Με τις εντολές που περιγράφηκαν προηγουμένως ανοίγει η επικοινωνία με το plc ενώ με το πάτημα του η τιμή της επαφής "M0.0" αλλάζει κατάσταση στιγμιαία από <<0>> σε <<1>> και επιστρέφει στην κατάσταση <<0>>. Με τον κώδικα που ακολουθεί πραγματοποιείται η επικοινωνία με την βάση δεδομένων όπως ακριβώς περιγράφηκε προηγουμένως ενώ στο τέλος παρουσιάζονται τα button που ενεργοποιούνται και απενεργοποιούνται όταν πατηθεί το General_Start, πιο συγκεκριμένα ενεργοποιείται το button για την πραγματοποίηση συνταγών , το button για το σταμάτημα της αυτόματης λειτουργίας καθώς επίσης και τα πεδία (textboxes) στα οποία ο χρήστης εισάγει τις απαιτούμενες ποσότητες που επιθυμεί από κάθε δεξαμενή για την πραγματοποίηση μιας συνταγής ενώ απενεργοποιούνται το ίδιο το button του General_Start αλλά και ο περιστροφικός διακόπτης έτσι ώστε μόλις ξεκινήσει η αυτόματη λειτουργία για την παραγωγή μιας συνταγής να μην μπορεί να διακοπεί παρα μόνο όταν ολοκληρωθεί η διαδικασία. Με τον ίδιο ακριβώς τρόπο συντάχθηκε ο κώδικας για την λειτουργία του General_Stop της αυτόματης λειτουργίας.

```

private void Recipe_btn_Click(object sender, EventArgs e)
{
    int a, b, c, d, f, g, i;
    int.TryParse(requested_water.Text, out a);
    int.TryParse(requested_orange.Text, out b);
    int.TryParse(requested_apple.Text, out c);
    int.TryParse(tank1 Volumepercentage.Text, out d);

```



```

int.TryParse(tank2Volumepercentage.Text, out f);
int.TryParse(tank3Volumepercentage.Text, out g);
i = a + b + c;
if (i != 100 || requested_water.Text == "" || requested_orange.Text == "" ||
requested_apple.Text == "")
{
    using (var plc = new Plc(CpuType.S71200, "127.0.0.1", 0, 1))
    {
        plc.Open();
        plc.Write("M0.1", true);
        plc.Write("M0.1", false);
    }
    MessageBox.Show("Wrong Values!Please insert values with sum=100
,press General Start and start recipe button again");
    General_Start.Enabled = true;
    Recipe_btn.Enabled = true;
}
else if (a != 50 || b > f || c > g)
{
    using (var plc = new Plc(CpuType.S71200, "127.0.0.1", 0, 1))
    {
        plc.Open();
        plc.Write("M0.1", true);
        plc.Write("M0.1", false);
    }
    MessageBox.Show("Wrong Values!Please check if requested amount(%) of
water is 50% and if the other two %requested amounts are less than or equal to the
%remaining amounts,then press General Start and start recipe button again");
    General_Start.Enabled = true;
    Recipe_btn.Enabled = true;
}

else
{
    using (var plc = new Plc(CpuType.S71200, "127.0.0.1", 0, 1))
    {
        plc.Open();
        string address = ("MW120");
        object setpoint = requested_water.Text;
        setpoint = Convert.ToUInt16(requested_water.Text);
        plc.Write(address, setpoint);
        string address1 = ("MW160");
        object setpoint1 = requested_orange.Text;
        setpoint1 = Convert.ToUInt16(requested_orange.Text);
        plc.Write(address1, setpoint1);
        string address2 = ("MW200");
        object setpoint2 = requested_apple.Text;
        setpoint2 = Convert.ToUInt16(requested_apple.Text);
        plc.Write(address2, setpoint2);
        plc.Write("M1.1", true);
        plc.Write("M1.1", false);
    }
}

```

```

    }

    string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
    string sqlQuery = "INSERT INTO
Actions_Data(Recipe_Start,requested_water,requested_orange,requested_apple)
VALUES (@Recipe_Start,@requested_water,@requested_orange,@requested_apple)
";

    SqlConnection con = new SqlConnection(connectionString);
    con.Open();
    SqlCommand cmd = new SqlCommand(sqlQuery, con);
    var Recipe_StartParameter = new SqlParameter("Recipe_Start",
System.Data.SqlDbType.DateTime);
    Recipe_StartParameter.Value = DateTime.Now;
    cmd.Parameters.Add(Recipe_StartParameter);

    var requested_waterParameter = new SqlParameter("requested_water",
System.Data.SqlDbType.VarChar);
    requested_waterParameter.Value = requested_water.Text;
    cmd.Parameters.Add(requested_waterParameter);

    var requested_orangeParameter = new SqlParameter("requested_orange",
System.Data.SqlDbType.VarChar);
    requested_orangeParameter.Value = requested_orange.Text;
    cmd.Parameters.Add(requested_orangeParameter);

    var requested_appleParameter = new SqlParameter("requested_apple",
System.Data.SqlDbType.VarChar);
    requested_appleParameter.Value = requested_apple.Text;
    cmd.Parameters.Add(requested_appleParameter);

    cmd.ExecuteNonQuery();
    con.Close();

    General_Stop.Enabled = true;
}
}

```

Το παραπάνω τμήμα του κώδικα περιγράφει την λειτουργία του button Recipe_Start με την βοήθεια του οποίου ξεκινά η εκτέλεση των συνταγών κατά την αυτόματη λειτουργία της διάταξης. Αρχικά ορίζονται κάποιες συνθήκες για την πραγματοποίηση συγκεκριμένων μόνο συνταγών με την αντίστοιχη εμφάνιση μηνυμάτων που βοηθούν τον χρήστη στην εισαγωγή των σωστών ποσοτήτων για την πραγματοποίηση μιας συνταγής. Πιο συγκεκριμένα, θα πρέπει το άθροισμα των % ποσοτήτων που ζητά ο χρήστης κάθε φορά από τις τρεις δεξαμενές των συστατικών να είναι πάντοτε ίσο με το 100 και ποτέ να μην παραμένει κενό το πεδίο εισαγωγής % ποσότητας(textbox) για καμία δεξαμενή διαφορετικά στιγμιαία η επαφή M0.1 που αντιστοιχεί στο General_Stop γίνεται <<1>>, εμφανίζεται σχετικό μήνυμα προκειμένου ο χρήστης να εισάγει τις σωστές ποσότητες και ενεργοποιούνται τα button General_Start και Recipe_Start. Επιπλέον, η % ποσότητα που απαιτείται κάθε φορά από την πρώτη δεξαμενή που είναι του νερού θα πρέπει να είναι πάντοτε ίση με το 50% ενώ των αλλών

2 θα πρέπει να είναι πάντοτε μικρότερες ή ίσες με τις διαθέσιμες % ποσότητες που περιέχονται στις δεξαμενές διαφορετικά στιγμιαία η επαφή M0.1 που αντιστοιχεί στο General_Stop γίνεται <<1>>, εμφανίζεται και πάλι αντίστοιχο μήνυμα όπως προηγουμένως και ενεργοποιούνται τα button General_Start και Recipe_Start. Στην περίπτωση που ο χρήστης εισάγει % ποσότητες που πληρούν τις παραπάνω προϋποθέσεις ανοίγεται επικοινωνία με το plc και εγγράφονται οι τιμές που εισήγαγε ο χρήστης μέσω του Scada στις αντίστοιχες word μεταβλητές του προγράμματος του plc (MW120 για την απαιτούμενη ποσότητα νερού, MW160 για την απαιτούμενη ποσότητα πολτού πορτοκαλιού και MW200 για την απαιτούμενη ποσότητα πολτού μήλου) ενώ στιγμιαία η τιμή της επαφής M1.1 που αντιστοιχεί στο button Recipe_Start μεταπίπτει από την αρχική κατάσταση <<0>> στην <<1>> προκειμένου να ξεκινήσει η πραγματοποίηση της συνταγής. Έπειτα ανοίγεται επικοινωνία με την βάση δεδομένων με την βοήθεια connection string και εγγράφονται σε αυτήν όλες οι τιμές που εισάγει ο χρήστης για τις % ποσότητες που απαιτεί από κάθε δεξαμενή αλλά καταγράφεται επίσης και κάθε φορά που πατά το button Recipe_Start. Το Recipe_Start έχει δηλωθεί ως τύπου datetime προκειμένου να καταγράφεται η ακριβής ημερομηνία και ώρα που πατείται κάθε φορά ενώ οι % ποσότητες των συστατικών έχουν δηλωθεί ως τύπου varchar έτσι ώστε να καταγράφονται κάθε φορά οι ακριβείς τιμές τους.

```
private void requested_water_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))
    {
        e.Handled = true;
    }
}

private void requested_orange_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))
    {
        e.Handled = true;
    }
}

private void requested_apple_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))
    {
        e.Handled = true;
    }
}
```

Με την βοήθεια του παραπάνω τμήματος του κώδικα επιτρέπεται η εισαγωγή μόνο αριθμητικών τιμών στα textboxes που αναφέρονται στις % ποσότητες που ζητά ο χρήστης από κάθε δεξαμενή προκειμένου να αποφευχθεί η περίπτωση εισαγωγής κάποιας τιμής τύπου char (όπως letter) που θα οδηγούσε σε έξοδο από την εφαρμογή.

```
private void Res_Tot_water_Click(object sender, EventArgs e)
{
    using (var plc = new Plc(CpuType.S71200, "127.0.0.1", 0, 1))
```

```

{
    plc.Open();
    plc.Write("M1.4", true);
    plc.Write("M1.4", false);
}
string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
string sqlQuery = "INSERT INTO Actions_Data(Reset_Tot_water)
VALUES(@Reset_Tot_water)";
SqlConnection con = new SqlConnection(connectionString);
con.Open();
SqlCommand cmd = new SqlCommand(sqlQuery, con);
var Reset_Tot_waterParameter = new SqlParameter("Reset_Tot_water",
System.Data.SqlDbType.DateTime);
Reset_Tot_waterParameter.Value = DateTime.Now;
cmd.Parameters.Add(Reset_Tot_waterParameter);
cmd.ExecuteNonQuery();
con.Close();
}

```

Ο παραπάνω κώδικας περιγράφει την λειτουργία του button Res_Tot_water με το πάτημα του οποίου μηδενίζεται κάθε φορά η τιμή του μετρητή αθροιστικής ροής που αφορά το νερό. Όπως παρατηρούμε ανοίγεται επικοινωνία με το plc και στιγμιαία με το πάτημα του η επαφή M1.4 αποκτά την τιμή <<1>>. Έπειτα, όπως κάθε φορά, παρουσιάζεται η σύνδεση με την βάση δεδομένων όπως έχει εξηγηθεί παραπάνω. Με τον ίδιο ακριβώς τρόπο περιγράφηκε η λειτουργία και των άλλων 3 button που πραγματοποιούν αντίστοιχη λειτουργία για τις υπόλοιπες 3 δεξαμενές της διαταξής μας.

```

private void timer1_Tick(object sender, EventArgs e)
{
    using (var plc = new Plc(CpuType.S71200, "127.0.0.1", 0, 1))
    {
        plc.Open();

        string address = ("MW90");
        object result = plc.Read(address);
        tank1 Volumepercentage.Text = string.Format("{0}", result.ToString());
        userControl11.value = Int16.Parse(tank1 Volumepercentage.Text);
        userControl11.changeValue();

        var result1=((uint)plc.Read("MD106")).ConvertToFloat();
        tank1_Height.Text = result1.ToString();

        var result2 = ((uint)plc.Read("MD110")).ConvertToFloat();
        tank1 Volume.Text = result2.ToString();

        var result5 = ((uint)plc.Read("MD132")).ConvertToFloat();
        flow_total_water.Text = result5.ToString();
    }
}

```

```

bool result6 = (bool)plc.Read("Q0.0");
if (result6 == true)
{
    pump1.ImageLocation = @"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ
ΔΙΠΔΕ\pump_green.jpg";
}
else
{
    pump1.ImageLocation = @"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ
ΔΙΠΔΕ\Horizontal pump (right).jpg";
}
bool result7 = (bool)plc.Read("Q0.1");
if (result7 == true)
{
    electrovalve1.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΔΕ\electrovalve_on.jpg";
}
else
{
    electrovalve1.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΔΕ\Control valve - vertical.jpg";
}
}

```

Όπως φαίνεται από το παραπάνω κομμάτι του κώδικα έχει οριστεί ένας timer με interval time=1s για την απεικόνιση των εξόδων της διάταξης. Πιο συγκεκριμένα, αρχικά ανοίγεται επικοινωνία με το plc και η τιμή της MW90, που αντιπροσωπεύει την % ποσότητα της πρώτης δεξαμενής του νερού, μεταφέρεται στο κατάλληλο textbox του scada αλλά και ως τιμή για το usercontrol11 που αποτελεί το αρχείο xaml (levelbar) που δημιουργήθηκε με σκοπό την καλύτερη οπτικοποίηση της στάθμης του υγρού που περιέχεται σε κάθε δεξαμενή. Στη συνέχεια η τιμή για το ύψος του νερού της δεξαμενής MD106 όπως υπολογίζεται από το πρόγραμμα του plc μεταφέρεται στο κατάλληλο textbox του scada. Με τον ίδιο ακριβώς τρόπο η τιμή για τον όγκο του νερού της δεξαμενής MD110 μεταφέρεται ως τιμή απεικόνισης του scada ενώ το ίδιο συμβαίνει και για την MD132 που αντιπροσωπεύει την συνολική αθροιστική ροή του νερού από ηλεκτροβάνια που είναι τοποθετημένα στον πυθμένα του. Τέλος, υπάρχει και μεταφορά των τιμών τύπου Boolean των εξόδων Q0.0 και Q0.1 που αντιπροσωπεύουν την κατάσταση της αντλίας που τροφοδοτεί την δεξαμενή του νερού και την κατάσταση της ηλεκτροβάνιας που βρίσκεται στον πυθμένα της. Όταν αλλάζει η τιμή της εξόδου αλλάζει και η αντίστοιχη εικόνα της για την οπτικοποίηση της λειτουργίας της. Με τον ίδιο ακριβώς τρόπο πραγματοποιήθηκε ο κώδικας και για τις υπόλοιπες τρεις δεξαμενές με την μόνη διαφορά ότι στην τελευταία δεξαμενή όπου συγκεντρώνονται τα συστατικά από τις υπόλοιπες 3 δεξαμενές και γίνεται η ανάδευση δεν υπάρχει κάποια αντλία τροφοδοσίας αλλά υπάρχει ο αναδευτήρας για τον οποίο χρησιμοποιείται ο ίδιος τύπος boolean με την αντλία.

```

string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
string sqlQuery = "INSERT INTO
Signals_Data(Date_Time,tank1_volume,tank1_volpercent,tank1_height,tank1_flow,t
ank2_volume,tank2_volpercent,tank2_height,tank2_flow,tank3_volume,tank3_volper
cent,tank3_height,tank3_flow,tank4_volume,tank4_volpercent,tank4_height,tank4_fl
ow,pump1,pump2,pump3,mixer,valve1,valve2,valve3,valve4)
VALUES(@Date_Time,@tank1_volume,@tank1_volpercent,@tank1_height,@tank
1_flow,@tank2_volume,@tank2_volpercent,@tank2_height,@tank2_flow,@tank3_v
olume,@tank3_volpercent,@tank3_height,@tank3_flow,@tank4_volume,@tank4_v
olpercent,@tank4_height,@tank4_flow,@pump1,@pump2,@pump3,@mixer,@valv
e1,@valve2,@valve3,@valve4)";
SqlConnection con = new SqlConnection(connectionString);
con.Open();
SqlCommand cmd = new SqlCommand(sqlQuery, con);
var Date_TimeParameter = new SqlParameter("Date_Time",
System.Data.SqlDbType.DateTime);
Date_TimeParameter.Value= DateTime.Now;
cmd.Parameters.Add(Date_TimeParameter);
var tank1_volumeParameter = new SqlParameter("tank1_volume",
System.Data.SqlDbType.VarChar);
tank1_volumeParameter.Value = tank1Volume.Text;
cmd.Parameters.Add(tank1_volumeParameter);

```

Στη συνέχεια, ανοίχτηκε επικοινωνία με την βάση δεδομένων, με την βοήθεια του connectionString που παρουσιάζεται παραπάνω, για την καταγραφή όλων των σημάτων της διάταξης που αποτελούν στοιχεία του πίνακα Signals_Data που δημιουργήθηκε στο Microsoft SQL Server Management Studio 18. Το πρώτο στοιχείο ορίστηκε ως τύπου datetime για την καταγραφή της ακριβούς ημερομηνίας και ώρας που αποθηκεύονται τα σήματα ενώ όλα τα υπόλοιπα στοιχεία δηλώθηκαν ως τύπου varchar για την καταγραφή της τιμής τους. Στο παραπάνω κομμάτι του κώδικα παρουσιάζεται η παραμετροποίηση για το στοιχείο Date_Time και το tank1_volume. Η παραμετροποίηση για τα υπόλοιπα στοιχεία έγινε με τον ίδιο τρόπο που πραγματοποιήθηκε για το tank1_volume. Στο σημείο αυτό πρέπει να σημειωθεί ότι το παραπάνω κομμάτι του κώδικα βρίσκεται στο εσωτερικό του timer με interval time=1s και επομένως οι τιμές των σημάτων καταγράφονται κάθε ένα δευτερόλεπτο στην βάση δεδομένων.

```

if (pos == 1 && result18 == true || pos == 1 && result24 == true || pos==1 &&
result6==true || pos==1 && result11==true || pos==1 && result16==true)
{
    selector_btn.Enabled = false;
    Recipe_btn.Enabled = false;
}
else if (pos == 1 && result18 == false && tank4Volumepercentage.Text ==
"0")
{
    selector_btn.Enabled = true;
    Recipe_btn.Enabled = true;
}
else if( pos == 0 || pos == 3 || pos == 2 && tank1Volumepercentage.Text ==
"0" && tank2Volumepercentage.Text == "0" && tank3Volumepercentage.Text ==
"0" && tank4Volumepercentage.Text == "0")
{

```

```

        selector_btn.Enabled = true;
    }
else
{
    selector_btn.Enabled = false;
}

if(pos==2 && tank1 Volumepercentage.Text == "0")
{
    Manual_Start_pump1.Enabled = true;
}
else
{
    Manual_Start_pump1.Enabled = false;
}

if (pos == 2 && tank2Volumepercentage.Text == "0")
{
    Manual_Start_pump2.Enabled = true;
}
else
{
    Manual_Start_pump2.Enabled = false;
}
if (pos == 2 && tank3Volumepercentage.Text == "0")
{
    Manual_Start_pump3.Enabled = true;
}
else
{
    Manual_Start_pump3.Enabled = false;
}
}

```

Το παραπάνω τμήμα κώδικα περιλαμβάνει κάποιες συνθήκες που ορίστηκαν για την σωστή λειτουργία του scada. Πιο συγκεκριμένα, σύμφωνα με την πρώτη συνθήκη όταν βρισκόμαστε στην αυτόματη λειτουργία και α) ο αναδευτήρας είναι σε λειτουργία ή β) η ηλεκτροβάνα της δεξαμενής του αναδευτήρα είναι σε λειτουργία ή γ) λειτουργεί οποιαδήποτε από τις τρεις αντλίες της διάταξης τότε απενεργοποιείται ο περιστροφικός διακόπτης και το button για την εκτέλεση συνταγών(Recipe_Start), όταν βρισκόμαστε στην αυτόματη λειτουργία και ο αναδευτήρας είναι εκτός λειτουργίας με την δεξαμενή του αναδευτήρα να είναι εντελώς άδεια τότε ενεργοποιείται η λειτουργία τόσο του περιστροφικού διακόπτη όσο και του button για την εκτέλεση συνταγών, όταν βρισκόμαστε σε οποιαδήποτε άλλη λειτουργία εκτός από την αυτόματη με όλες τις δεξαμενές τις διάταξης να είναι άδειες τότε η λειτουργία του περιστροφικού διακόπτη είναι ενεργή σε οποιαδήποτε άλλη περίπτωση η λειτουργία του είναι ανενεργή. Σύμφωνα με την δεύτερη συνθήκη όταν βρισκόμαστε στην χειροκίνητη λειτουργία με την πρώτη δεξαμενή άδεια τότε το button για την ενεργοποίηση της πρώτης αντλίας είναι ενεργό ενώ σε οποιαδήποτε άλλη περίπτωση είναι ανενεργό. Αντίστοιχες συνθήκες ορίστηκαν και για τις άλλες 2 αντλίες του συστήματος.

```
private void Manual_Start_pump1_Click(object sender, EventArgs e)
```



```

{
    using (var plc = new Plc(CpuType.S71200, "127.0.0.1", 0, 1))
    {
        plc.Open();
        plc.Write("M284.0", true);
        plc.Write("M284.0", false);
    }
    string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
    string sqlQuery = "INSERT INTO Actions_Data(start_pump1)
VALUES(@start_pump1)";
    SqlConnection con = new SqlConnection(connectionString);
    con.Open();
    SqlCommand cmd = new SqlCommand(sqlQuery, con);
    var start_pump1Parameter = new SqlParameter("start_pump1",
System.Data.SqlDbType.DateTime);
    start_pump1Parameter.Value = DateTime.Now;
    cmd.Parameters.Add(start_pump1Parameter);
    cmd.ExecuteNonQuery();
    con.Close();
}

```

Το παραπάνω κομμάτι του κώδικα περιγράφει την λειτουργία του button Manual_Start_pump1 που χρησιμοποιείται για την έναρξη της χειροκίνητης λειτουργίας την πρώτης αντλίας που τροφοδοτεί την πρώτη δεξαμενή που περιέχει το νερό. Αρχικά ανοίγεται επικοινωνία με το plc και η επαφή M284.0, που αντιστοιχεί στην χειροκίνητη έναρξη λειτουργίας της πρώτης αντλίας, λαμβάνει στιγμιαία την τιμή <<1>> μόλις πατηθεί το συγκεκριμένο button με αποτέλεσμα την χειροκίνητη έναρξη λειτουργίας της αντλίας. Ακολούθως, ανοίγεται επικοινωνία με την βάση δεδομένων για την καταγραφή της ημερομηνίας και της ώρας του πατήματος του συγκεκριμένου button στην βάση δεδομένων. Με τον ίδιο τρόπο αναπτύχθηκε ο κώδικας για όλα τα υπόλοιπα button που αφορούν τον χειροκίνητο έλεγχο της διάταξης.

```

private void Operator_Actions_History_Click(object sender, EventArgs e)
{
    historical_Data_manual_Screen.Show();
    EXIT.Show();
    dataGridView1.Show();
    dateTimePicker1.Show();
    dateTimePicker2.Show();
    Search_Actions.Show();
    Search_Signals.Hide();
    dataGridView2.Hide();

    string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
    SqlConnection con = new SqlConnection(connectionString);

```



```

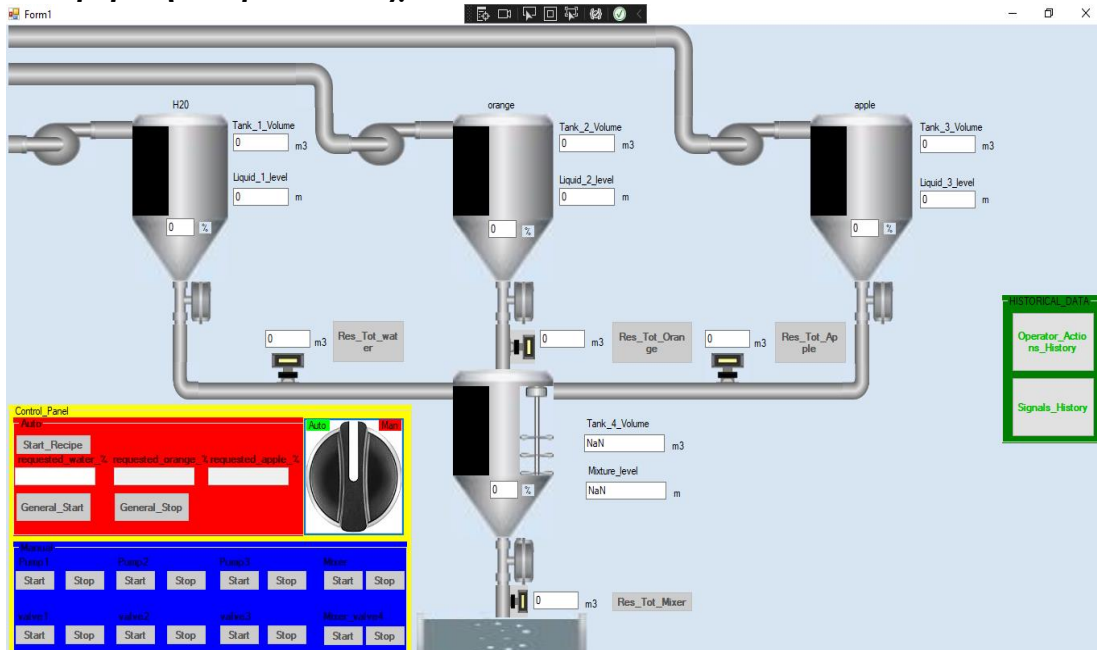
con.Open();
SqlDataAdapter sqlDa = new SqlDataAdapter("SELECT * from
Actions_Data", con);
DataTable dtbl = new DataTable();
sqlDa.Fill(dtbl);
dataGridView1.DataSource = dtbl;
con.Close();
}

```

Ο παραπάνω κώδικας περιγράφει την λειτουργία του button για την προβολή όλων των ιστορικών στοιχείων από τις ενέργειες του χρήστη. Με το πάτημα του εμφανίζονται: μια νέα σελίδα στην οποία περιέχεται ένα dataGridView με όλες τις ενέργειες του χρήστη, το κουμπί exit ώστε αν ο χρήστης το επιθυμεί να αποχωρήσει από την συγκεκριμένη σελίδα, δύο dateTimePickers με τη βοήθεια των οποίων ο χρήστης μπορεί να επιλέξει τις επιθυμητές ημερομηνίες μεταξύ των οποίων επιθυμεί να αναζητήσει τις ενέργειες που πραγματοποιήθηκαν από τον χρήστη και το button Search με την βοήθεια του οποίου μπορεί να εκτελέσει την συγκεκριμένη αναζήτηση. Στη συνέχεια παρουσιάζεται ο κώδικας που αναπτύχθηκε έτσι ώστε όλες οι ενέργειες του χρήστη να αποθηκεύονται στο dataGridView1. Με τον ίδιο ακριβώς τρόπο αναπτύχθηκε ο κώδικας για την λειτουργία του button που αφορά την προβολή όλων των σημάτων της διάταξης.

Κεφάλαιο 5ο: Αποτελέσματα Προσομοιώσεων

5.1 Προβολή ιστορικών στοιχείων



Εικόνα 5.1: Διακόπτης στην μεσαία θέση και δυνατότητα προβολής των ιστορικών στοιχείων

Όταν ο περιστροφικός διακόπτης βρίσκεται στην μεσαία θέση ο χειριστής του SCADA έχει την δυνατότητα διαχείρισης των δεδομένων από την λειτουργία της διάταξης με την μορφή προβολής ιστορικών στοιχείων ενώ όλα τα υπόλοιπα button του SCADA είναι απενεργοποιημένα. Πιο συγκεκριμένα, με το πάτημα του button Operator_Actions_History έχει την δυνατότητα προβολής των ενεργειών του χειριστή ενώ με το πάτημα του button Signals_History έχει την δυνατότητα προβολής όλων των σημάτων της διάταξης.

id	Auto	Manual	Buttons_Off	start_pump1	stop_pump1	start_pump2	stop_pump2	start_pump3	stop_pump3	start_valve1
1535	16/8/2023 9:35 πμ									
1536		16/8/2023 9:36 πμ								
1537			16/8/2023 9:36 πμ							
1538	16/8/2023 9:36 πμ									
1539										
1540										
1541		16/8/2023 9:36 πμ								
1542			16/8/2023 9:36 πμ							
1543	16/8/2023 9:36 πμ									
1544										
1545		16/8/2023 9:37 πμ								
1546										16/8/2023 9:39 πμ
1547										
1548										
1549										
1550										
1551										
1552			16/8/2023 9:47 πμ							
1553	16/8/2023 9:47 πμ									
1554										
1555										
1556	16/8/2023 10:46...									
1557										
1558										
1559										
1560										

Εικόνα 5.2: Στοιχεία ενεργειών χειριστή

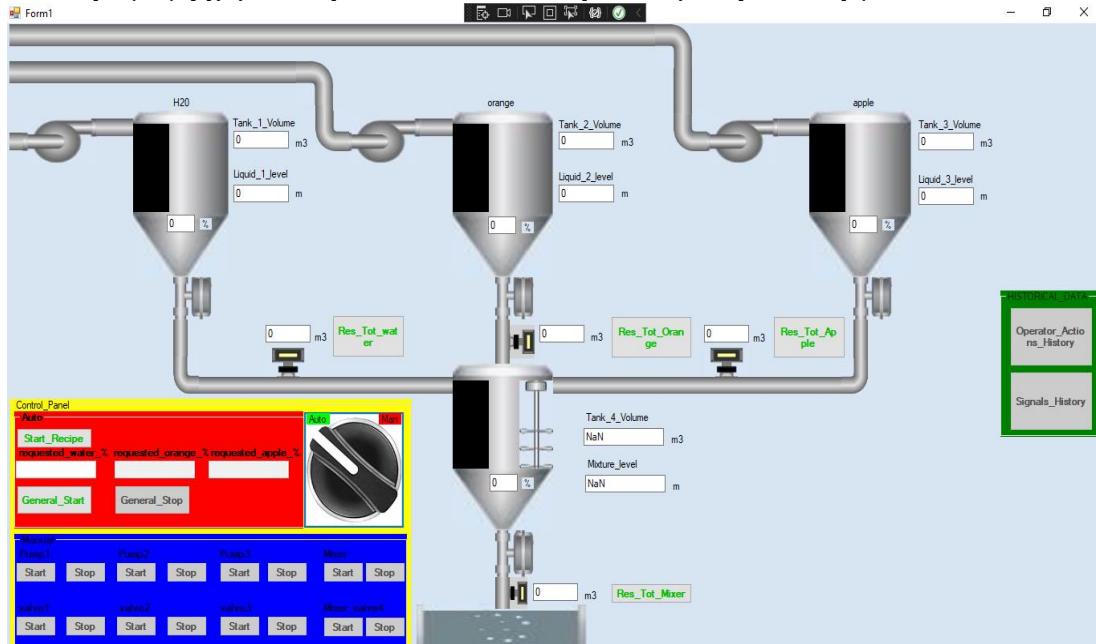
Στην παραπάνω εικόνα παρατηρούμε την ημερομηνία και την ώρα που πατήθηκε κάποιον από τα συνολικά 26 button της διάταξης αλλά και των τιμών που εισήγαγε ο χρήστης σε καθένα από τα τρία διαθέσιμα textboxes που χρησιμοποιούνται για την πραγματοποίηση συνταγών κατά την αυτόματη λειτουργία. Επιπλέον, υπάρχει η δυνατότητα αναζήτησης ενεργειών που πραγματοποιήθηκαν μεταξύ συγκεκριμένων ημερομηνιών που μπορεί να επιλέξει ο χρήστης.

id	Date_Time	tank_1_volume	tank_1_volpercent	tank_1_height	tank_1_flow	tank_2_volume	tank_2_volpercent	tank_2_height	tank_2_flow	tank_3_volume
21458	16/8/2023 10:50 πμ	1.253819	40	0.3993056	0.3448004	1.843852	50	0.5872141	0	3.521965
21459	16/8/2023 10:50 πμ	1.222474	39	0.3893229	0.3761459	1.843852	50	0.5872141	0	3.521965
21460	16/8/2023 10:50 πμ	1.191129	38	0.3793403	0.4074914	1.843852	50	0.5872141	0	3.521965
21461	16/8/2023 10:50 πμ	1.159783	37	0.3693576	0.4388368	1.843852	50	0.5872141	0	3.521965
21462	16/8/2023 10:50 πμ	1.128438	36	0.359375	0.4701823	1.843852	50	0.5872141	0	3.521965
21463	16/8/2023 10:50 πμ	1.097092	35	0.3493924	0.5015278	1.843852	50	0.5872141	0	3.521965
21464	16/8/2023 10:50 πμ	1.065747	34	0.3394097	0.5328733	1.843852	50	0.5872141	0	3.521965
21465	16/8/2023 10:50 πμ	1.034401	33	0.3294271	0.5642188	1.843852	50	0.5872141	0	3.521965
21466	16/8/2023 10:50 πμ	1.003056	32	0.3194444	0.5955643	1.843852	50	0.5872141	0	3.521965
21467	16/8/2023 10:50 πμ	0.9717101	31	0.3094618	0.6269098	1.843852	50	0.5872141	0	3.521965
21468	16/8/2023 10:50 πμ	0.9403646	30	0.2994792	0.6582553	1.843852	50	0.5872141	0	3.521965
21469	16/8/2023 10:50 πμ	0.9090192	29	0.2894965	0.6896008	1.843852	50	0.5872141	0	3.521965
21470	16/8/2023 10:50 πμ	0.8776737	28	0.2795139	0.7209463	1.843852	50	0.5872141	0	3.521965
21471	16/8/2023 10:50 πμ	0.8463281	27	0.2695313	0.7522917	1.843852	50	0.5872141	0	3.521965
21472	16/8/2023 10:50 πμ	0.8149827	26	0.2595486	0.7836372	1.843852	50	0.5872141	0	3.521965
21473	16/8/2023 10:50 πμ	0.7836372	25	0.249566	0.8149827	1.843852	50	0.5872141	0	3.521965
21474	16/8/2023 10:50 πμ	0.7522917	24	0.2395833	0.8463282	1.843852	50	0.5872141	0	3.521965
21475	16/8/2023 10:50 πμ	0.7209463	23	0.2296007	0.8776737	1.843852	50	0.5872141	0	3.521965
21476	16/8/2023 10:50 πμ	0.6896006	22	0.219618	0.9090192	1.843852	50	0.5872141	0	3.521965
21477	16/8/2023 10:50 πμ	0.6582552	21	0.2096354	0.9403647	1.843852	50	0.5872141	0	3.521965
21478	16/8/2023 10:50 πμ	0.6269097	20	0.1996528	0.9717101	1.843852	50	0.5872141	0	3.521965
21479	16/8/2023 10:50 πμ	0.6269097	20	0.1996528	0.9717101	1.843852	50	0.5872141	0	3.521965
21480	16/8/2023 10:50 πμ	0.6269097	20	0.1996528	0.9717101	1.843852	50	0.5872141	0	3.521965
21481	16/8/2023 10:50 πμ	0.6269097	20	0.1996528	0.9717101	1.843852	50	0.5872141	0	3.521965
21482	16/8/2023 10:50 πμ	0.6269097	20	0.1996528	0.9717101	1.843852	50	0.5872141	0	3.521965
21483	16/8/2023 10:50 πμ	0.6269097	20	0.1996528	0.9717101	1.843852	50	0.5872141	0	3.521965

Εικόνα 5.3: Στοιχεία σημάτων

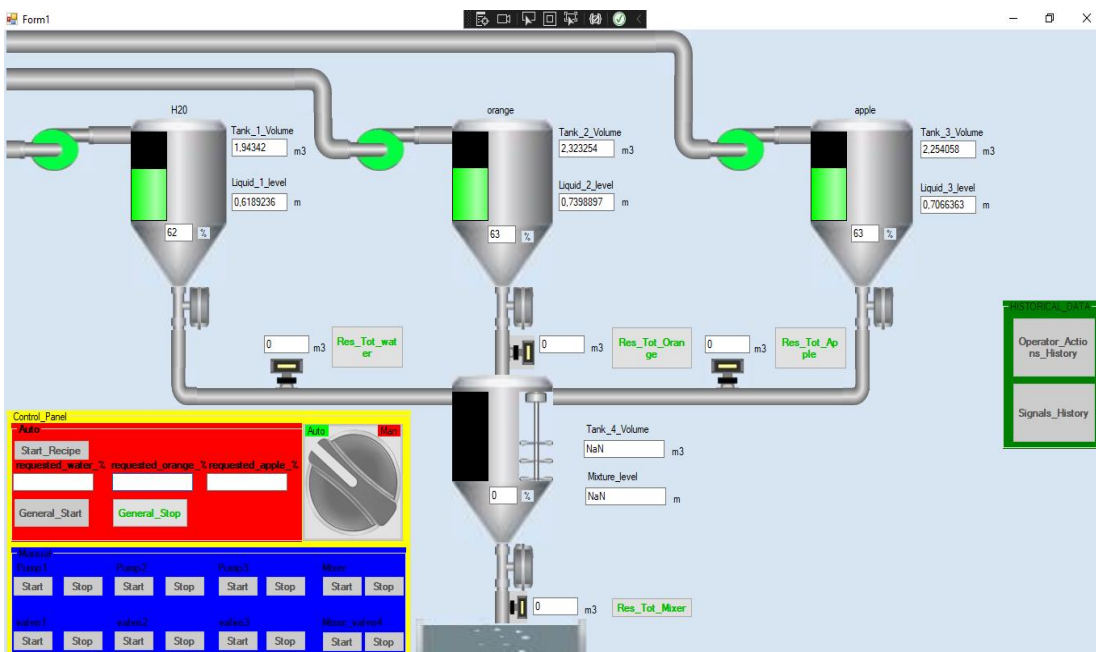
Στην παραπάνω εικόνα παρατηρούμε τις ακριβείς τιμές που έχει καθένα από τα συνολικά 24 σήματα της διάταξης μας κάθε δευτερόλεπτο με ταυτόχρονη καταγραφή της αντίστοιχης ημερομηνίας και ώρας. Ο χρήστης, όπως αναφέρθηκε και προηγουμένως, έχει την δυνατότητα αναζήτησης σημάτων μεταξύ δύο συγκεκριμένων ημερομηνιών που μπορεί να επιλέξει ο ίδιος μέσω του SCADA.

5.2 Παραγωγή χυμού πορτοκαλιού κατά την αυτόματη λειτουργία



Εικόνα 5.4: Διακόπτης στην θέση Auto και δυνατότητα έναρξης της αυτόματης λειτουργίας

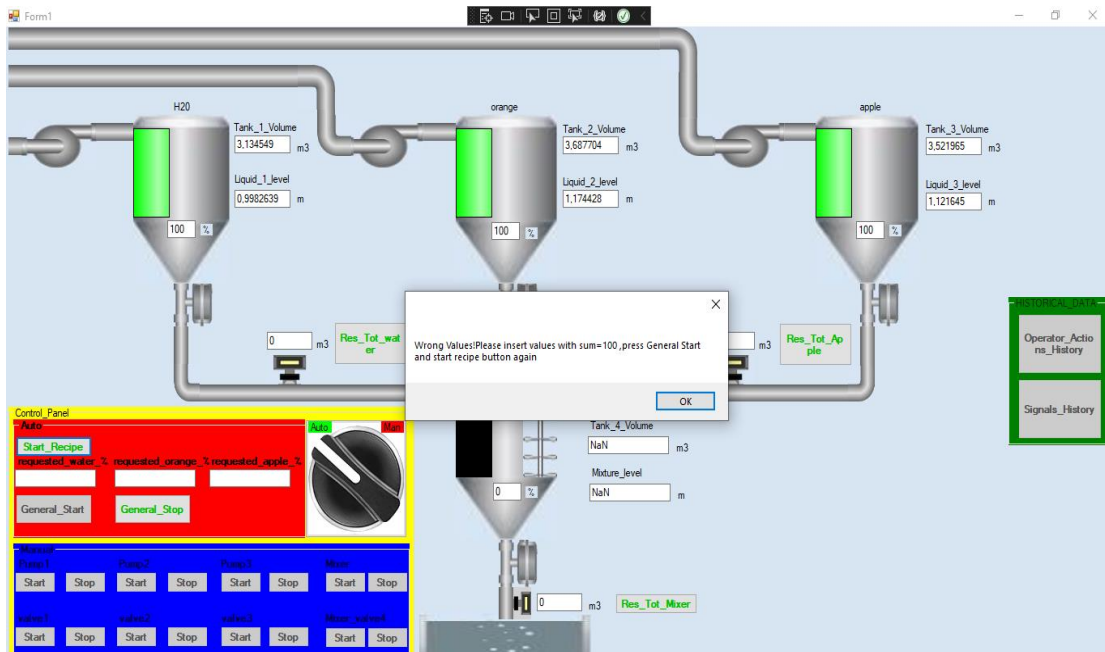
Όταν ο περιστροφικός διακόπτης βρίσκεται στην θέση Auto της αυτόματης λειτουργίας τότε τα μόνα button που παραμένουν ενεργοποιημένα είναι αυτά του Start_Recipe, του General_Start της αυτόματης λειτουργίας καθώς επίσης και τα button που μηδενίζουν τις τιμές των μετρητών της αθροιστικής ροής που είναι τοποθετημένοι σε καθεμία από τις τέσσερις δεξαμενές της διάταξης μας. Επιπλέον, όταν όλες οι δεξαμενές είναι άδειες όπως φαίνεται παραπάνω τότε απενεργοποιημένα είναι και τα πεδία των textboxes στα οποία ο χειριστής εισάγει τις % ποσότητες που επιθυμεί για την πραγματοποίηση μιας συγκεκριμένης συνταγής.



Εικόνα 5.5: Αυτόματη πλήρωση των δεξαμενών από τις αντλίες

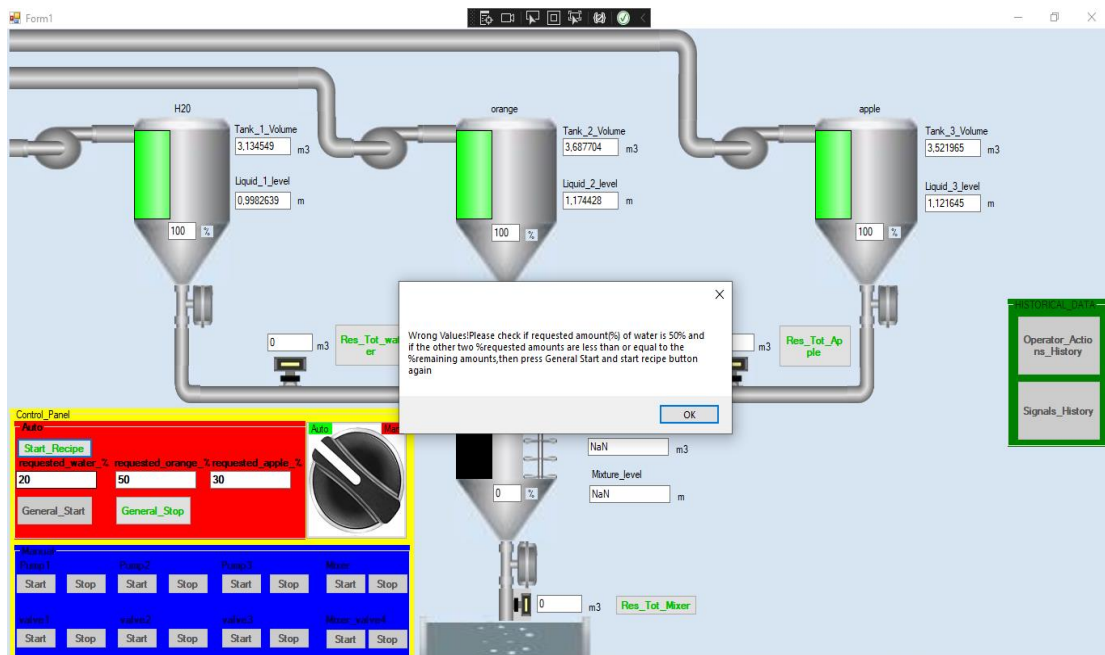
Στην παραπάνω εικόνα παρουσιάζεται η διαδικασία πλήρωσης των δεξαμενών κατά την αυτόματη λειτουργία με την βοήθεια τριών αντλιών. Όπως παρατηρούμε όλα τα button είναι απενεργοποιημένα κατά την διάρκεια της διαδικασίας με εξαίρεση μόνο τα button των

μετρητών της αθροιστικής ροής καθώς επίσης και του General_Stop της αυτόματης λειτουργίας.



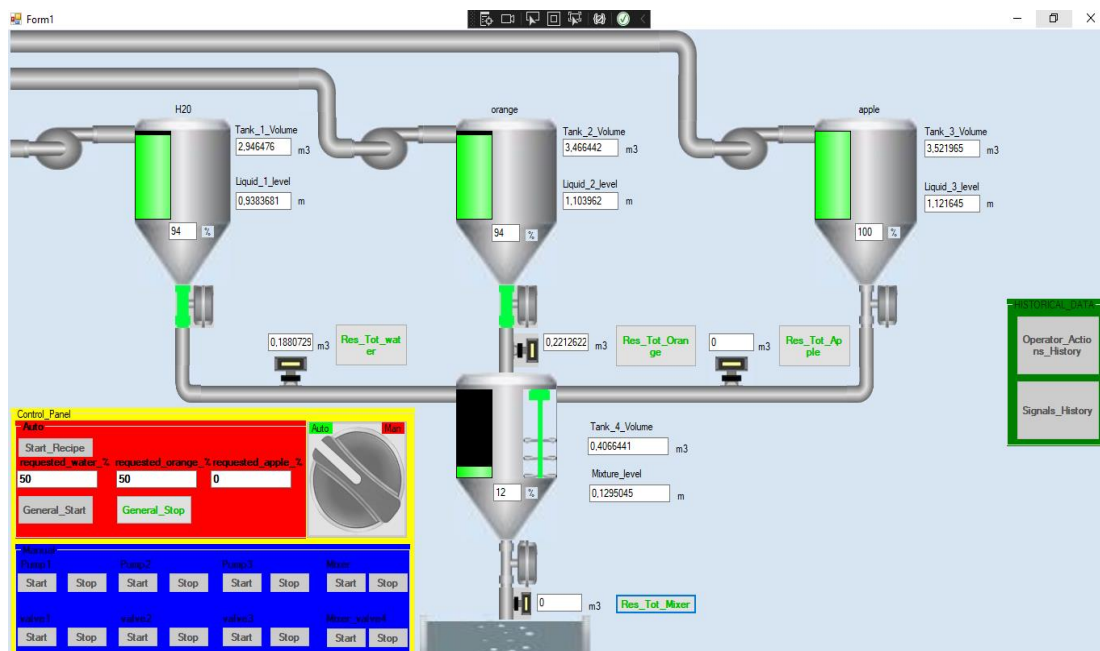
Εικόνα 5.6: Προειδοποιητικό μήνυμα για την απαίτηση σωστών ποσοτήτων από τις δεξαμενές

Στην παραπάνω εικόνα παρατηρούμε την περίπτωση που έπειτα από την ολοκλήρωση της διαδικασίας πλήρωσης των δεξαμενών ο χρήστης εισάγει τιμές στα τρία διαθέσιμα textboxes της αυτόματης διαδικασίας με σκοπό την πραγματοποίηση κάποιας συνταγής. Όπως παρατηρούμε οι τιμές που έδωσε ο χρήστης δεν ικανοποιούν την συνθήκη που έχει οριστεί για την αυτόματη διαδικασία σύμφωνα με την οποία το άθροισμα των % ποσοτήτων που εισάγονται για τις τρεις δεξαμενές θα πρέπει να είναι πάντοτε ίσο με το 100 και για τον λόγο αυτό εμφανίζεται στην οθόνη του SCADA το παραπάνω προειδοποιητικό μήνυμα.



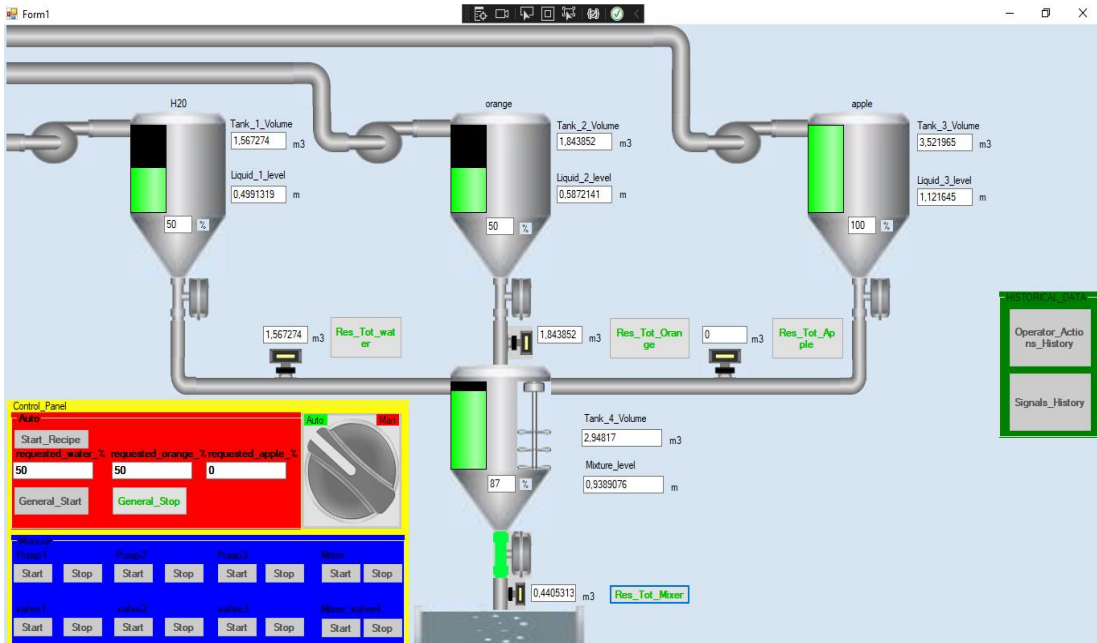
Εικόνα 5.7: Προειδοποιητικό μήνυμα για την απαίτηση σωστών ποσοτήτων από τις δεξαμενές

Στην παραπάνω εικόνα παρατηρούμε την περίπτωση που έπειτα από την ολοκλήρωση της διαδικασίας πλήρωσης των δεξαμενών ο χρήστης εισάγει μεν τιμές, στα τρία textboxes της αυτόματης διαδικασίας, που έχουν άθροισμα 100 αλλά η % ποσότητα που εισάγει για την δεξαμενή του νερού είναι διάφορη του 50% γεγονός που δεν ικανοποιεί την συνθήκη που έχουμε ορίσει για την αυτόματη διαδικασία με αποτέλεσμα την εμφάνιση του παραπάνω μηνύματος.



Εικόνα 5.8: Εκτέλεση συνταγής στην αυτόματη λειτουργία

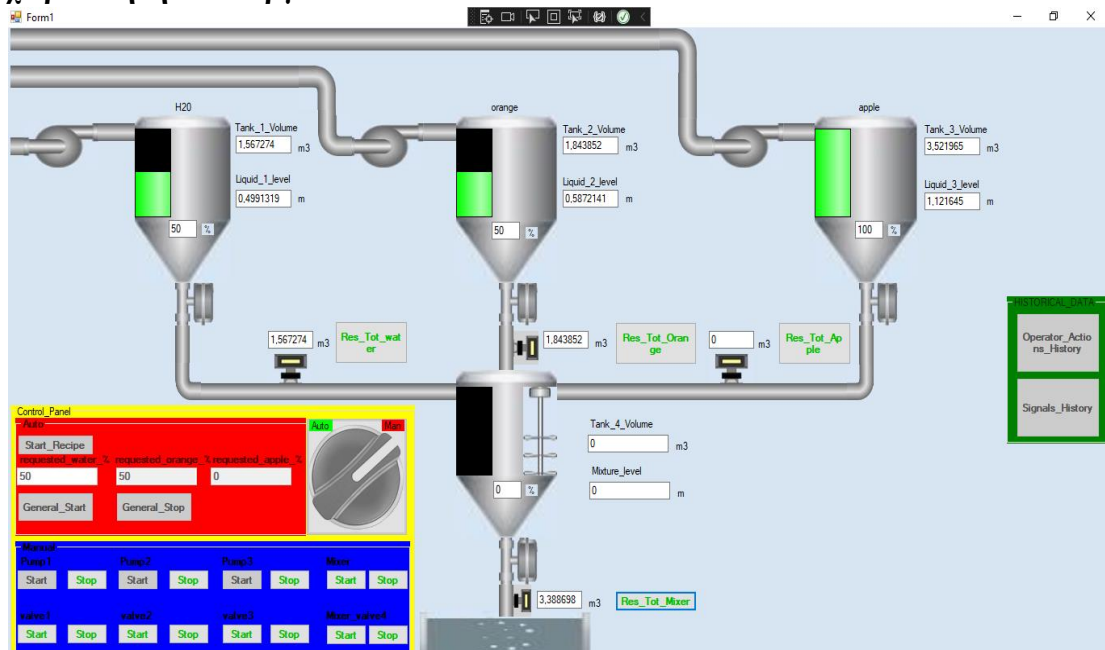
Στην παραπάνω εικόνα παρατηρούμε την διαδικασία εκτέλεσης μιας συνταγής στην οποία ο χρήστης έχει ζητήσει 50% ποσότητα νερού από την πρώτη δεξαμενή και 50% ποσότητα πολτού πορτοκαλιού από την δεύτερη δεξαμενή με στόχο την παραγωγή χυμού πορτοκαλιού. Όπως βλέπουμε στην συγκεκριμένη περίπτωση δεν παραβιάζεται καμία συνθήκη από αυτές που έχουμε ορίσει για την αυτόματη λειτουργία της διάταξης με αποτέλεσμα να λειτουργούν κανονικά οι ηλεκτροβάνες των 2 πρώτων δεξαμενών μέχρι το περιεχόμενό τους να γίνει ίσο με 50% ενώ ταυτόχρονα λειτουργεί και ο αναδευτήρας για την ανάμειξη των 2 συστατικών και την παραγωγή του τελικού προϊόντος. Επιπλέον, με την βοήθεια μετρητών υπολογίζεται η αθροιστική ροή που εξέρχεται από καθεμία από αυτές τις δεξαμενές ενώ τα μόνα button που είναι ενεργοποιημένα στο συγκεκριμένο στάδιο της διαδικασίας είναι αυτά που μηδενίζουν τους μετρητές καθώς επίσης και το button του General_Stop της αυτόματης διαδικασίας.



Εικόνα 5.9: Παραγωγή και αποθήκευση του τελικού προϊόντος

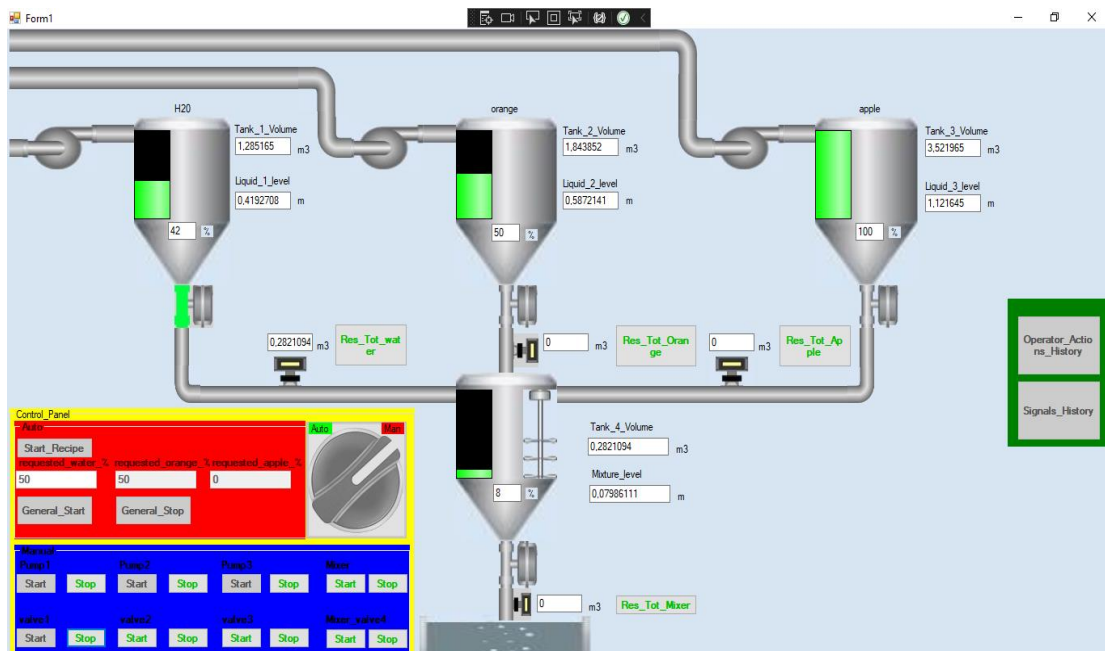
Στο στάδιο της διαδικασίας που περιγράφεται από την παραπάνω εικόνα έχει πραγματοποιηθεί η ανάμειξη των 2 συστατικών που ζητήθηκαν για την παραγωγή χυμού πορτοκαλιού με αποτέλεσμα να έχουμε ενεργοποιημένη την ηλεκτροβάννα της δεξαμενής του αναδευτήρα για την συλλογή και αποθήκευση του τελικού προϊόντος. Επιπλέον, η συνολική ποσότητα του μείγματος που εξέρχεται από την δεξαμενή καταγράφεται κάθε στιγμή με την βοήθεια του μετρητή που υπολογίζει την αθροιστική ροή στην έξοδο της δεξαμενής.

5.3 Παραγωγή ανάμεικτου χυμού με αναλογίες που ορίζει ο χειριστής στην χειροκίνητη λειτουργία



Εικόνα 5.11: Διακόπτης στην θέση χειροκίνητης λειτουργίας

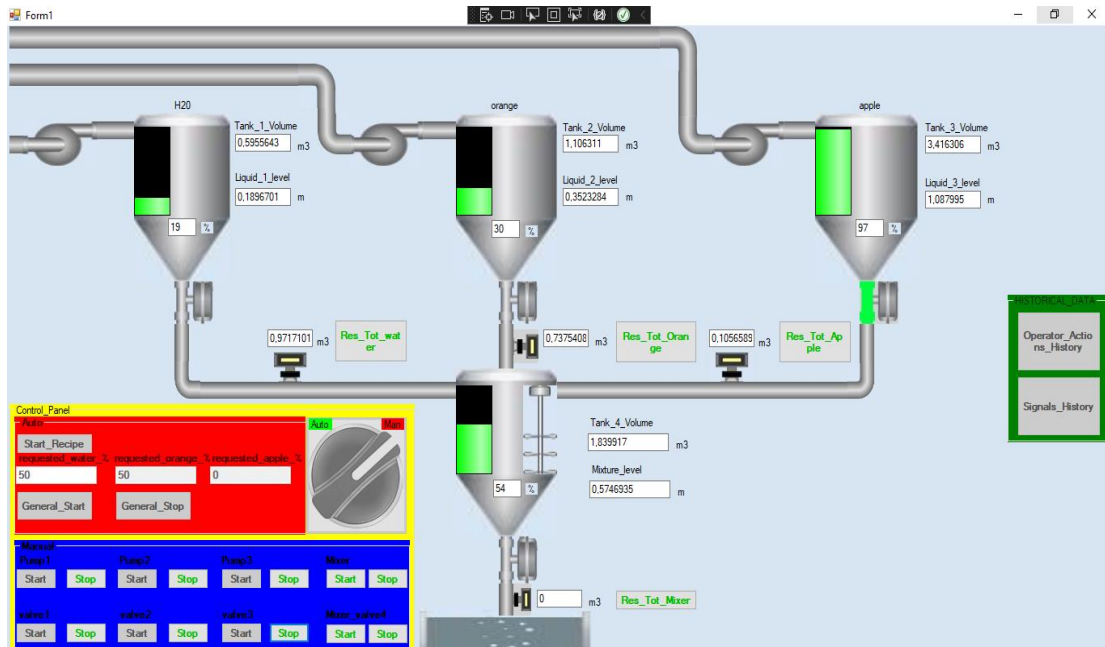
Όταν ο περιστροφικός διακόπτης βρίσκεται στην θέση της χειροκίνητης λειτουργίας με τις δεξαμενές των συστατικών να μην είναι άδειες τότε τα button που είναι ενεργοποιημένα για την εκτέλεση κάποιας συνταγής είναι τα button που μηδενίζουν τους μετρητές και όλα τα button ελέγχου της χειροκίνητης λειτουργίας με εξαίρεση τα button εκκίνησης των αντλιών τα οποία είναι ενεργά μόνο όταν η δεξαμενή την οποία τροφοδοτούν είναι εντελώς άδεια.



Εικόνα 5.12: Χειροκίνητη επιλογή ποσότητας από την πρώτη δεξαμενή για την εκτέλεση συνταγής

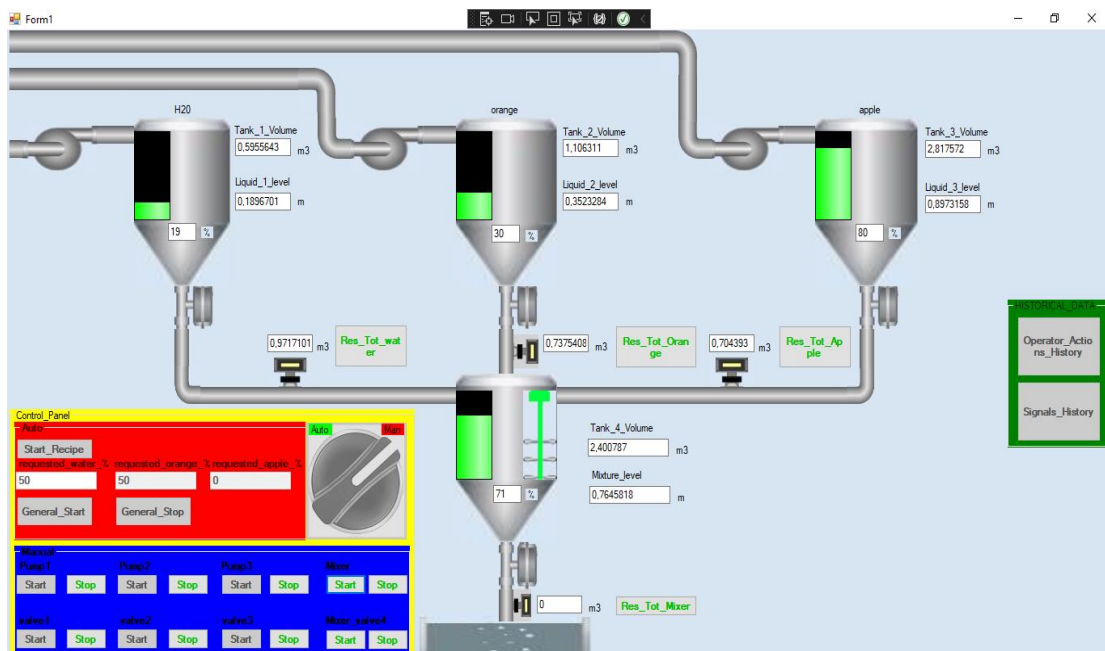
Στο στάδιο της διαδικασίας που περιγράφεται από την παραπάνω εικόνα ο χρήστης επιλέγει την % ποσότητα που επιθυμεί από την πρώτη δεξαμενή πατώντας το button Start που

αντιστοιχεί στην πρώτη ηλεκτροβάννα, το οποίο όπως παρατηρούμε με το πάτημα του ταυτόχρονα απενεργοποιείται, και η ποσότητα αυτή καθορίζεται μέχρι να πατήσει το αντίστοιχο button Stop και να απενεργοποιήσει την πρώτη ηλεκτροβάννα. Παράλληλα όπως βλέπουμε ξεκινά την καταμέτρηση και ο μετρητής που υπολογίζει την αθροιστική ροή για την πρώτη δεξαμενή. Να σημειωθεί ότι πριν την εκκίνηση της πρώτης ηλεκτροβάννας πραγματοποιήθηκε μηδενισμός όλων των μετρητών της διάταξης.



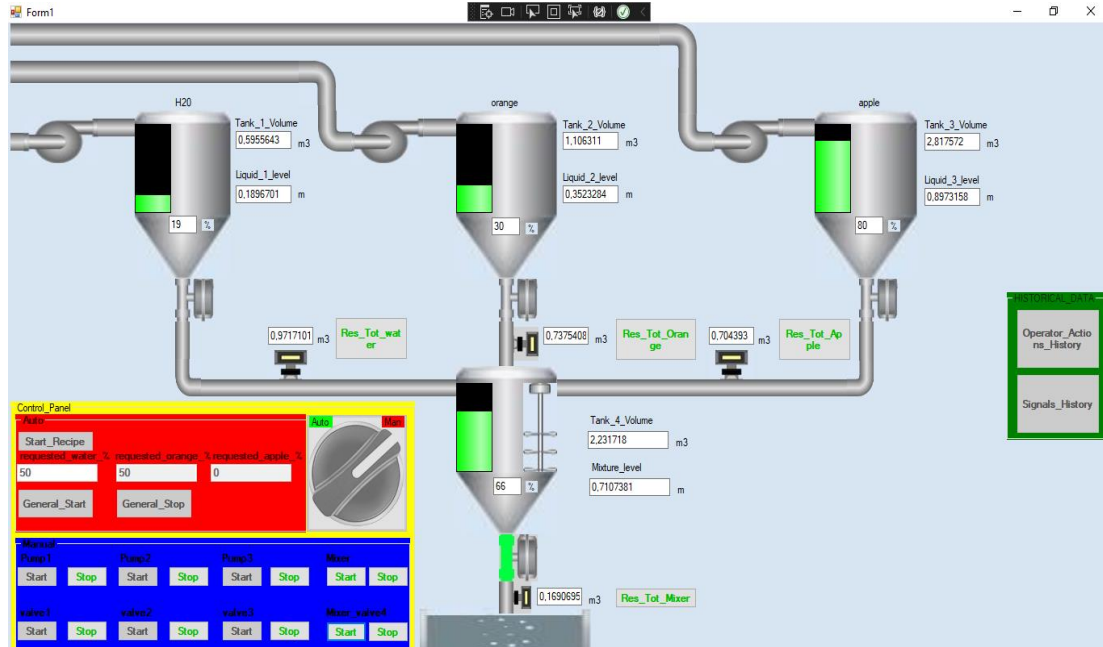
Εικόνα 5.13: Χειροκίνητη επιλογή ποσότητας από την τρίτη δεξαμενή για την εκτέλεση συνταγής

Στην παραπάνω εικόνα ο χρήστης έχοντας επιλέξει % ποσότητες για τις 2 πρώτες δεξαμενές επιλέγει και την % ποσότητα ποσότητα που επιθυμεί για τον πολτό μήλου που είναι το τρίτο κατά σειρά συστατικό με τον ίδιο ακριβώς τρόπο που περιγράφηκε προηγουμένως.



Εικόνα 5.14: Χειροκίνητος έλεγχος του mixer

Στην παραπάνω εικόνα παρατηρούμε ότι πλέον ο χειριστής του SCADA έχει επιλέξει τις % ποσότητες που επιθυμεί από κάθε δεξαμενή και πατώντας το button Start που αντιστοιχεί στον αναδευτήρα πραγματοποιείται η ανάμειξη των συστατικών για όσο διάστημα επιθυμεί ο χειριστής μέχρι να πατήσει ο ίδιος του το button Stop που αντιστοιχεί στην διακοπή της λειτουργίας του αναδευτήρα.



Εικόνα 5.15: Χειροκίνητος έλεγχος της ηλεκτροβάνας του μείγματος

Όπως παρατηρούμε από την παραπάνω εικόνα, έπειτα από την ολοκλήρωση της ανάμειξης των συστατικών που περιγράφηκε παραπάνω πραγματοποιείται η συλλογή και η αποθήκευση του παραγόμενου μείγματος όταν ο χειριστής πατήσει το button Start της ηλεκτροβάνας της δεξαμενής του αναδευτήρα με αποτέλεσμα την ενεργοποίησή της και την ταυτόχρονη ενεργοποίηση του μετρητή που υπολογίζει την συνολική αθροιστική ροή που διέρχεται από την έξοδο της δεξαμενής του μείγματος. Η απενεργοποίηση της ηλεκτροβάνας γίνεται πατώντας το αντίστοιχο button Stop μόλις η δεξαμενή αδειάσει εντελώς.

Κεφάλαιο 6ο: Συμπεράσματα

Στην παρούσα εργασία παρατηρήσαμε ότι είναι εφικτή η δημιουργία ενός low cost scada με την χρήση της αντικειμενοστραφούς γλώσσας προγραμματισμού C# και της ελεύθερης βιβλιοθήκης S7.Net που είναι κατάλληλη για βιομηχανικούς ελεγκτές της εταιρείας Siemens. Μας δόθηκε η δυνατότητα εποπτείας και χειρισμού της παραγωγικής διαδικασίας και της πραγματοποίησης συνταγών με αυτόματη αλλά και με χειροκίνητη λειτουργία. Η όλη διαδικασία πλήρωσης και εκροής των δεξαμενών στηρίχτηκε κατά κύριο λόγο στην λειτουργία των counter up και counter down καθώς επίσης και στο clock memory byte που διαθέτει το TIA Portal με την βοήθεια των οποίων πραγματοποιήθηκε μια προσέγγιση της διαδικασίας χωρίς την χρήση αναλογικών αισθητηρίων πίεσης αλλά και γενικότερα hardware. Επιπλέον, πέρα από την εποπτεία και τον χειρισμό όπως μαρτυρά και το ίδιο το ακρωνύμιο του SCADA μέσω της συγκεκριμένης εφαρμογής μας παρέχεται η δυνατότητα αποθήκευσης και διαχείρισης όλων των σημάτων και των ενεργειών του χειριστή σε μία βάση δεδομένων. Μια μελλοντική επέκταση της εφαρμογής που παρουσιάστηκε θα μπορούσε να είναι η διαχείριση και η αναζήτηση των σημάτων να πραγματοποιείται και από περιβάλλον web καθώς επίσης και η υλοποίηση της με πραγματικό hardware για την κάλυψη των πραγματικών αναγκών της βιομηχανίας.

Βιβλιογραφία

Βιβλία

[1] Α. Τσαγκάρης, *Προγραμματιζόμενοι Λογικοί Ελεγκτές και Συστήματα Εποπτικού Ελέγχου (PLC & SCADA)*, Εκδόσεις Broken Hill Publishers Ltd, 2021

Manuals

[2] Siemens, Getting started with S7-1200

Internet Site

[3] <https://www.mesta-automation.com/plc-pc-communication-with-c-a-quick-resume-about-data-exchange-libraries/>

[4] <https://usermanual.wiki/Document/NetToPLCsimManualen.1468207178/html>

Papers

[5] H. Salih, H. Abdelwahab and A. Abdallah, "Automation design for a syrup production line using Siemens PLC S7-1200 and TIA Portal software," *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, Khartoum, Sudan, 2017, pp. 1-5, doi: 10.1109/ICCCCEE.2017.7866702.

[6] A. I. Abashar, M. A. Mohammedtoun and O. D. Abaker, "Automated and monitored liquid filling system using PLC technology," *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, Khartoum, Sudan, 2017, pp. 1-5, doi: 10.1109/ICCCCEE.2017.7866699.

[7] B. Hasan, S. S. -u. -H. Mohani, S. S. Hussain, S. Yasin, W. A. Alvi and O. Saeed, "Implementation of Supervisory Control and Data Acquisition - SCADA on a PLC and VFD Controlled Digital Mixing Plant Using TIA Portal," *2019 4th International Conference on Emerging Trends in Engineering, Sciences and Technology (ICEEST)*, Karachi, Pakistan, 2019, pp. 1-6, doi: 10.1109/ICEEST48626.2019.8981705.

[8] A. Mercurio, A. Di Giorgio and P. Cioci, "Open-Source Implementation of Monitoring and Controlling Services for EMS/SCADA Systems by Means of Web Services— IEC 61850 and IEC 61970 Standards," in *IEEE Transactions on Power Delivery*, vol. 24, no. 3, pp. 1148-1153, July 2009, doi: 10.1109/TPWRD.2008.2008461.

[9] R. I. Rajkumar, T. J. Alexander and P. Devi, "ZigBee based design of low cost SCADA system for industrial process applications," *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Chennai, India, 2016, pp. 1-4, doi: 10.1109/ICCIC.2016.7919696.

[10] L. O. Aghenta and M. T. Iqbal, "Development of an IoT Based Open Source SCADA System for PV System Monitoring," *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, Edmonton, AB, Canada, 2019, pp. 1-4, doi: 10.1109/CCECE.2019.8861827.

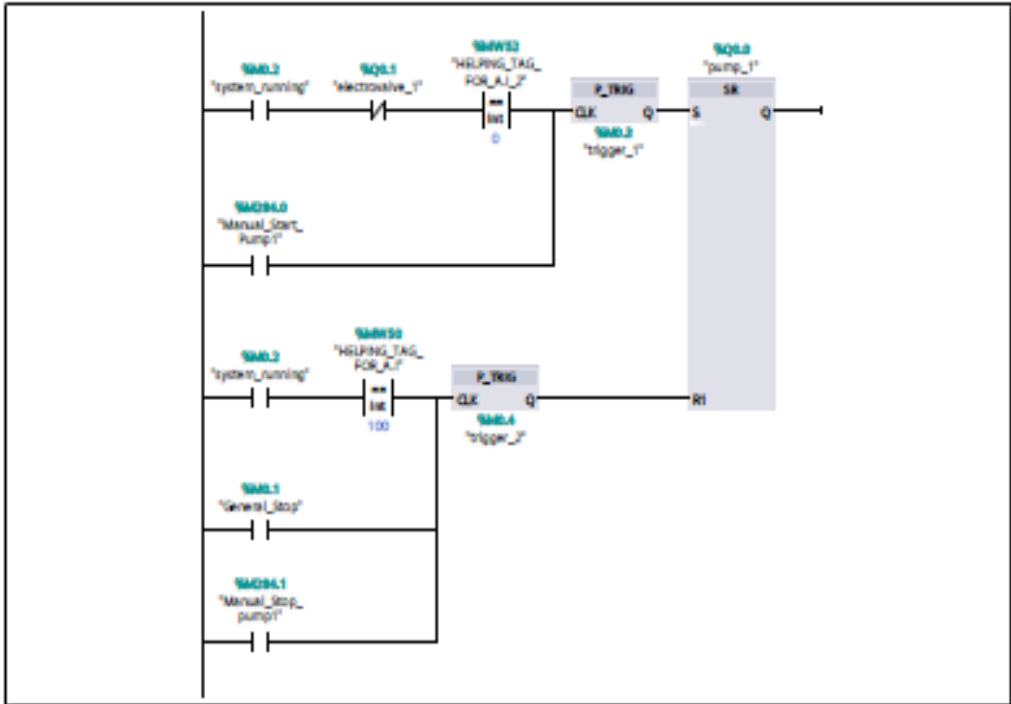
[11] S. Phuyal, D. Bista, J. Izykowski and R. Bista, "Design and Implementation of Cost Efficient SCADA System for Industrial Automation," *I.J. Engineering and Manufacturing*, 2020, doi: 10.5815/ijem.2020.02.02

Διπλωματικές

[12]Ι.Συντιγάκης,*Διπλωματική Εργασία: Βιομηχανικός έλεγχος:Υλοποίηση και εφαρμογές*,Πολυτεχνείο Κρήτης,2020

ΠΑΡΑΡΤΗΜΑ Α : ΚΩΔΙΚΑΣ LADDER

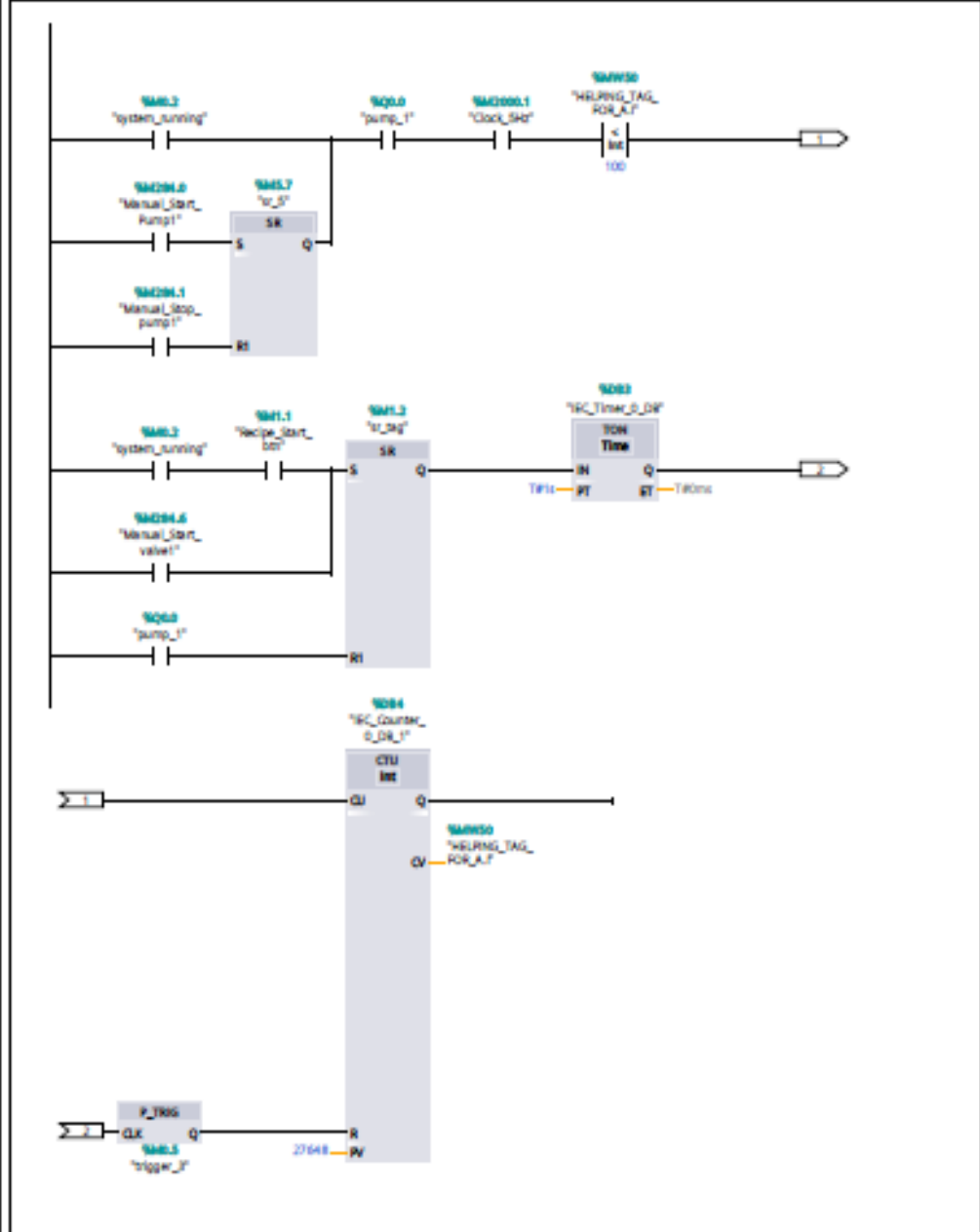
Totally Integrated Automation Portal						
TELIKH_DIPLWMATIKH_ERGASIA / PLC_1 [CPU 1214C DC/DC/Rly] / Program blocks						
Main [OB1]						
Main Properties						
General						
Name	Main	Number	1	Type	OB	
Language	LAD	Numbering	Automatic			
Information						
Title	"Main Program Sweep (Cycle)"	Author			Comment	
Family		Version	0.1	User-defined ID		
Main						
Name		Data type	Default value	Comment		
▼ Input						
Initial_Call		Bool		Initial call of this OB		
Remanence		Bool		⇒True, if remanent data are available		
Temp						
Constant						
Network 1: START/STOP						
Network 2: water_TANK1_block						
Network 3: water_PUMP1_CONTROL						
network 3: water_pump1_control						



Network 4: WATER_LEVEL_ANIMATION_FOR_FILLING_WITH_PUMP1

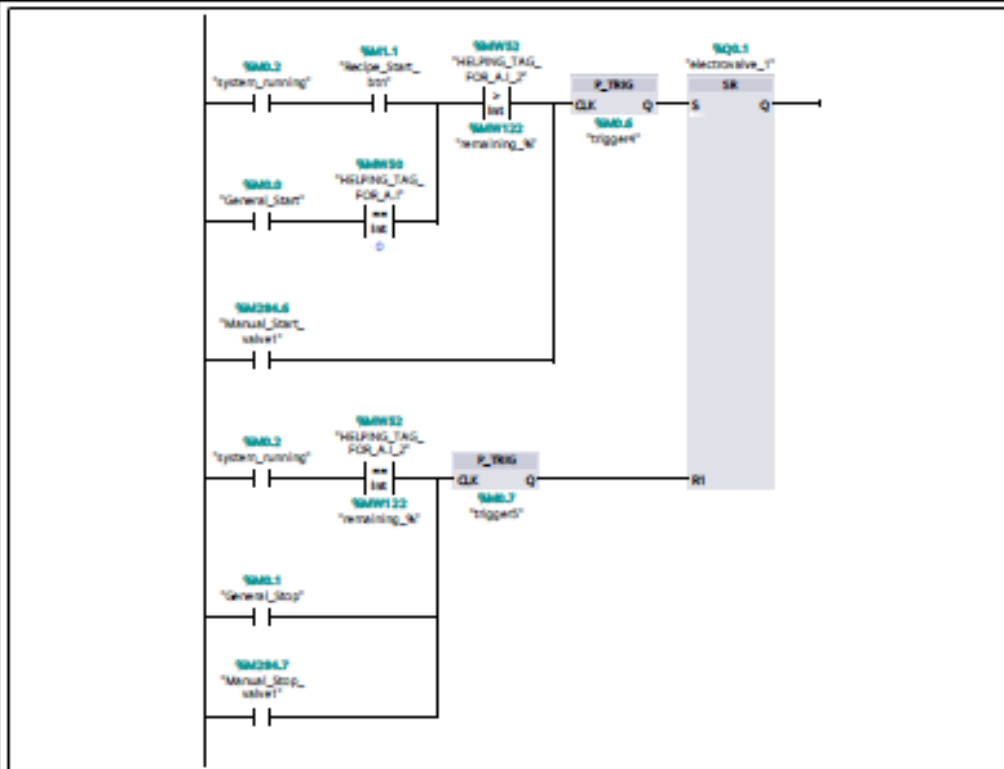
Network 4: WATER_LEVEL_ANIMATION_FOR_FILLING_WITH_PUMP1

Network 4: WATER_LEVEL_ANIMATION_FOR_FILLING_WITH_PUMP1



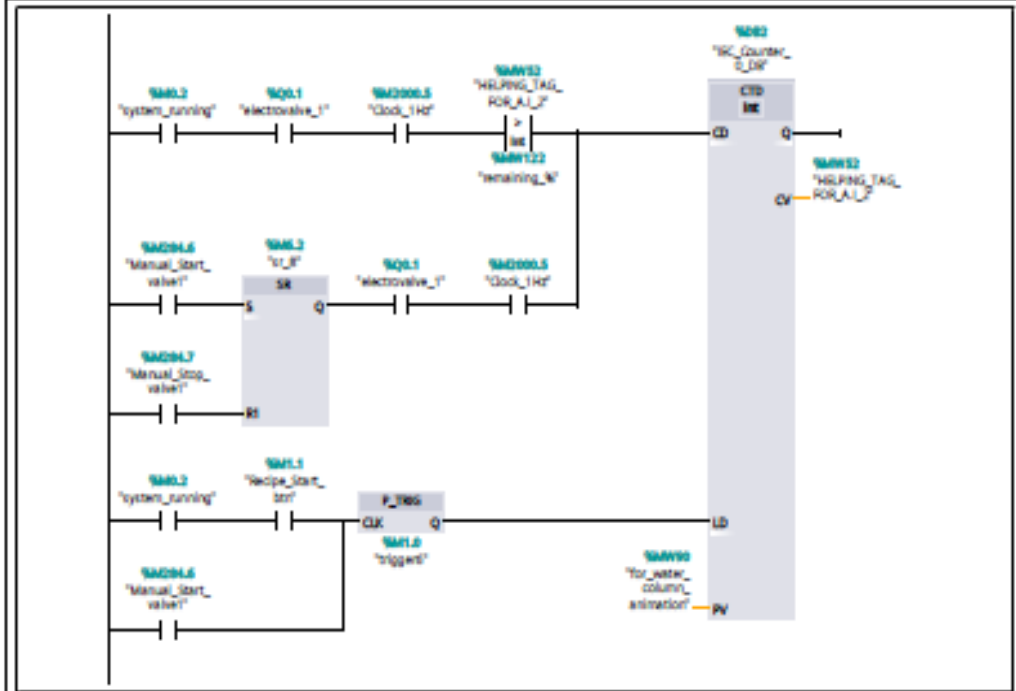
Network 5: water_valve1_CONTROL

network 5:water_valve1_CONTROL



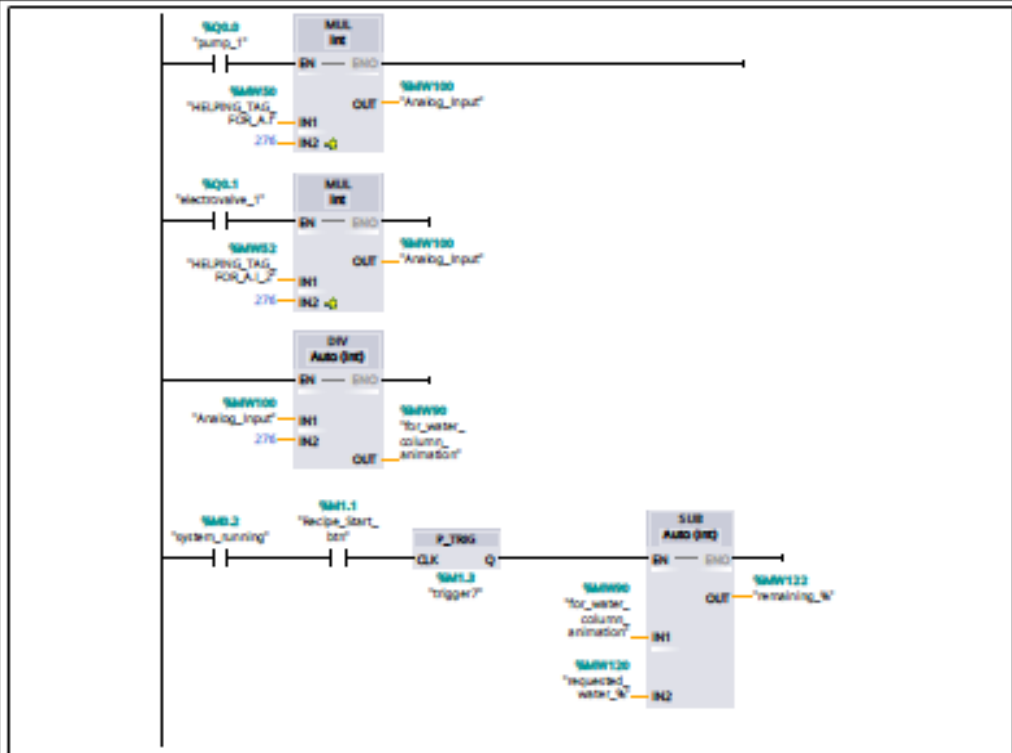
Network 6: water_level_animation_for_discharging_with_electrovalve1

network 6:water_level_animation_for_discharging_with_electrovalve1

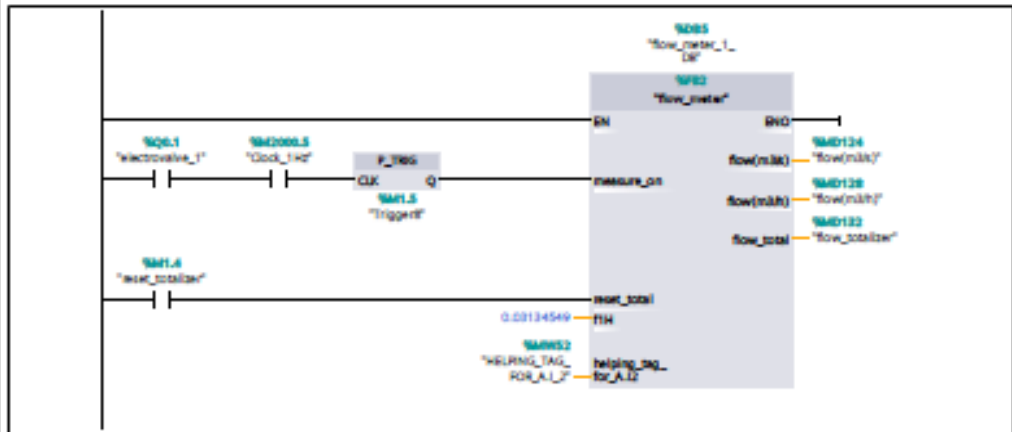


Network 7: CALIBRATION OF ANALOG INPUT for water

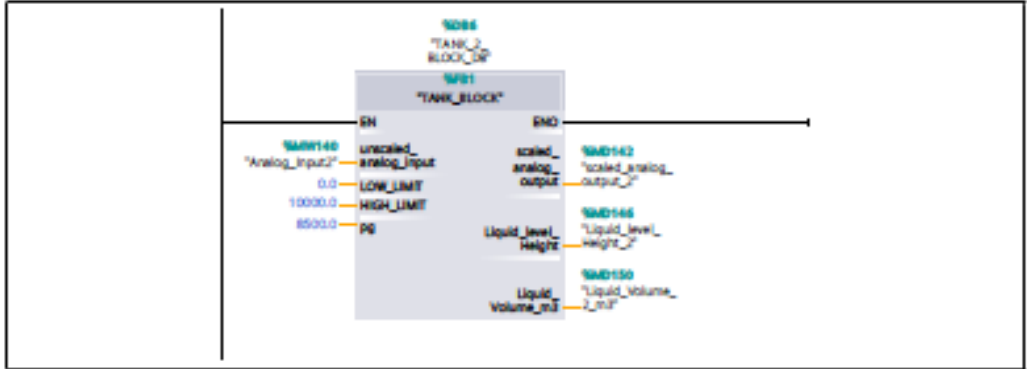
network 7: CALIBRATION OF ANALOG INPUT for water



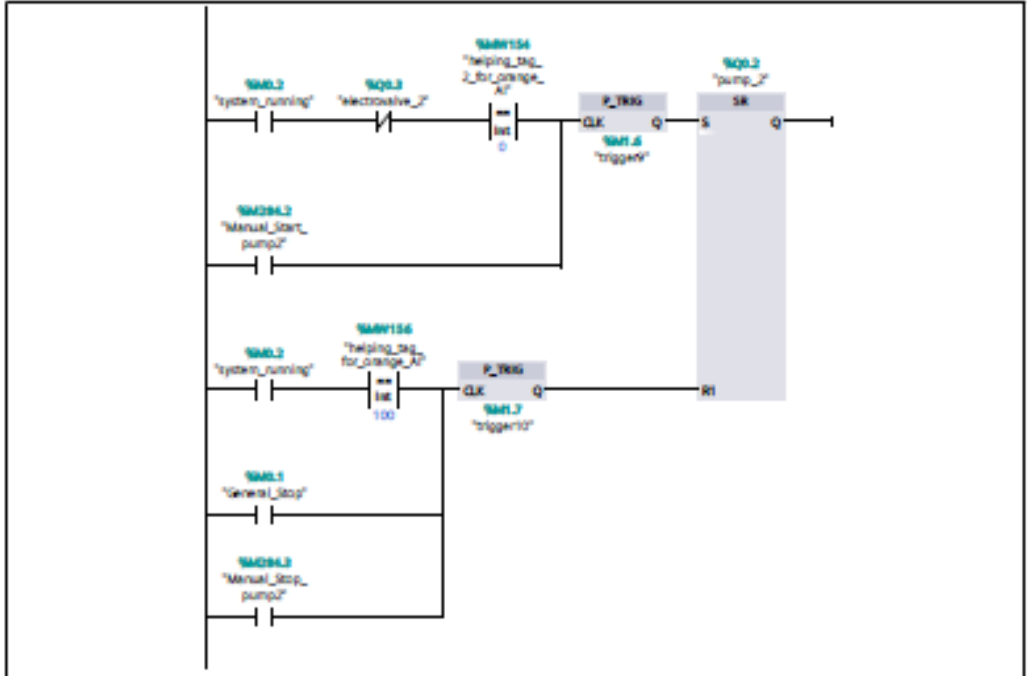
Network 8: flow_totalizer_water



Network 9: orange_tank2_block

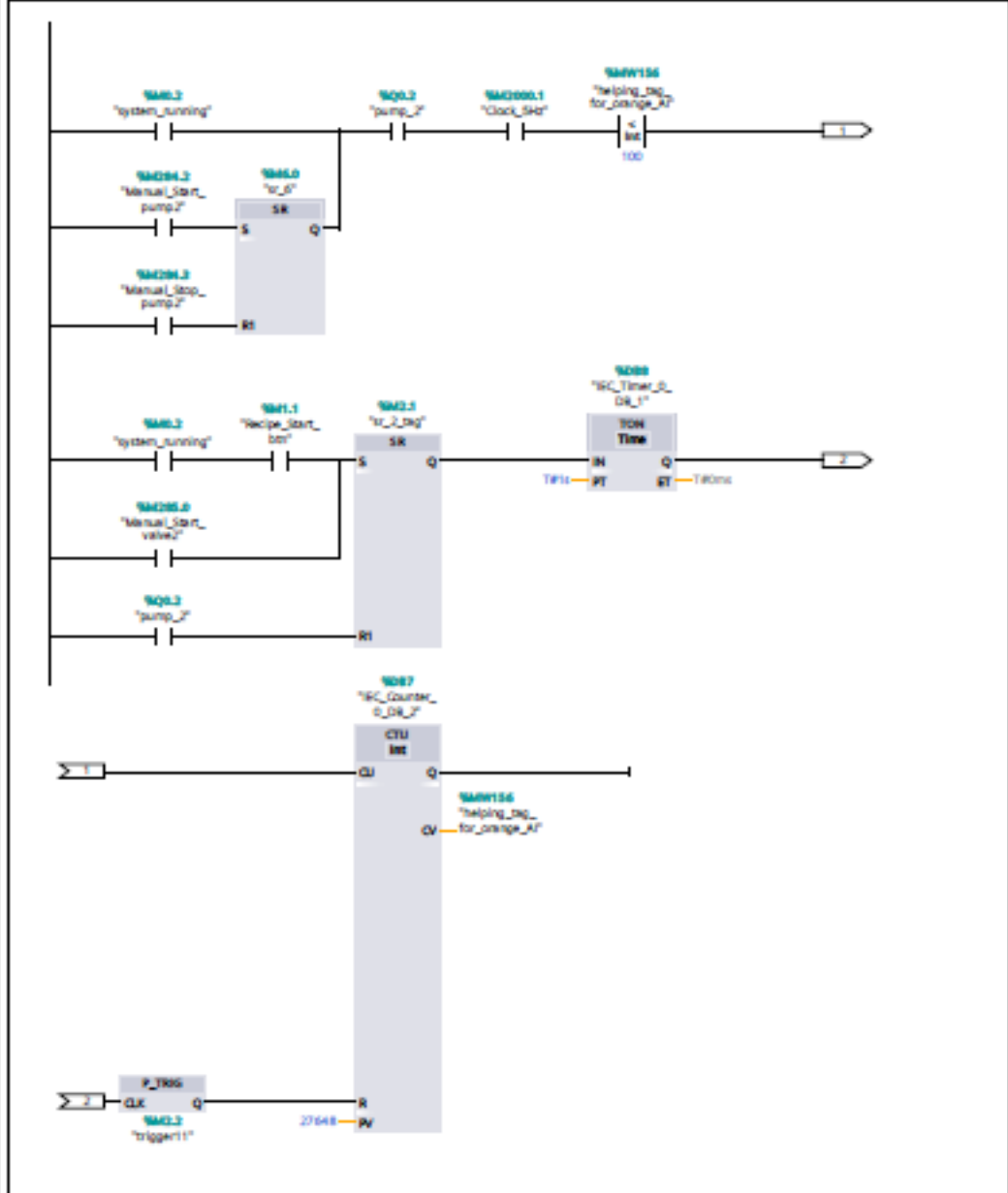


Network 10: orange_pump2_control

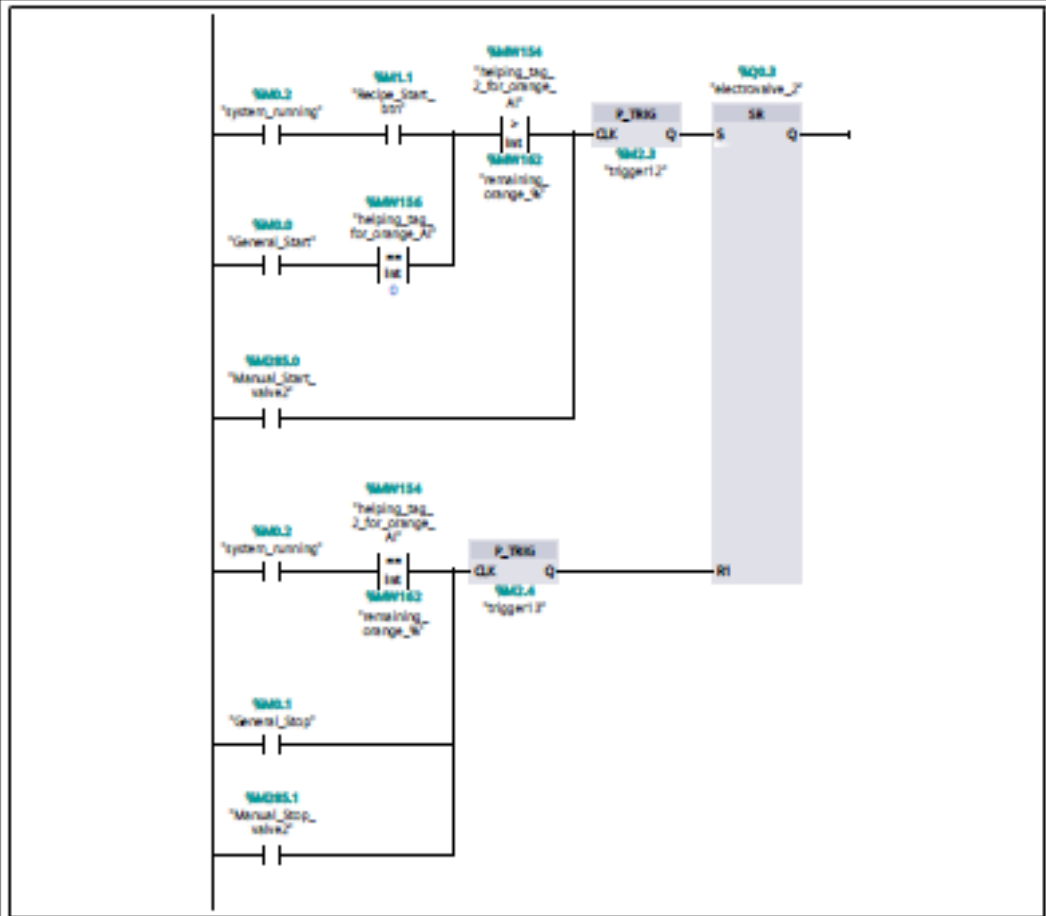


Network 11: ORANGE_LEVEL_ANIMATION_FOR_FILLING_WITH_PUMP2

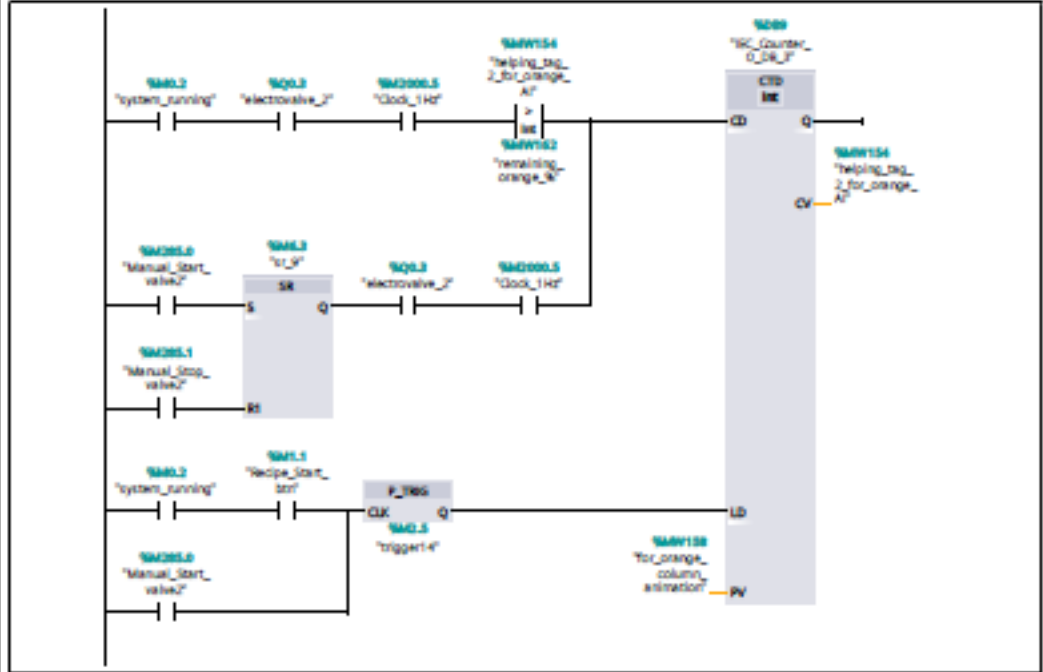
Network 11: ORANGE_LEVEL_ANIMATION_FOR_FILLING_WITH_PUMP2



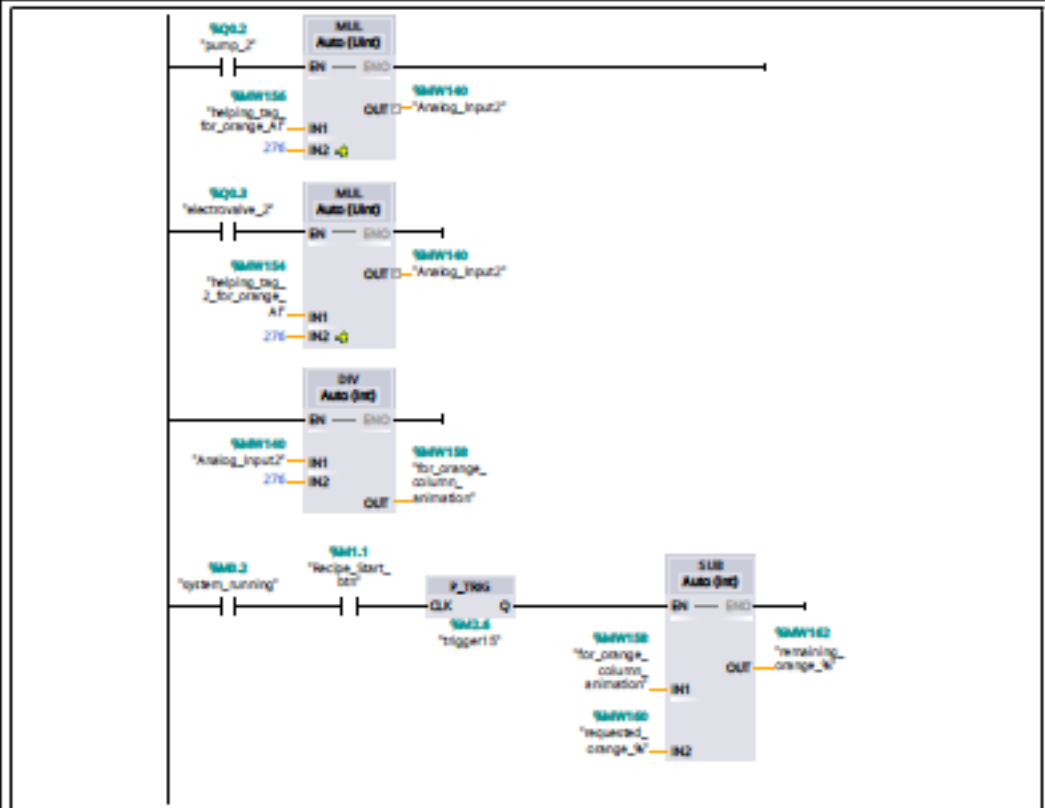
Network 12: orange_valve2_control



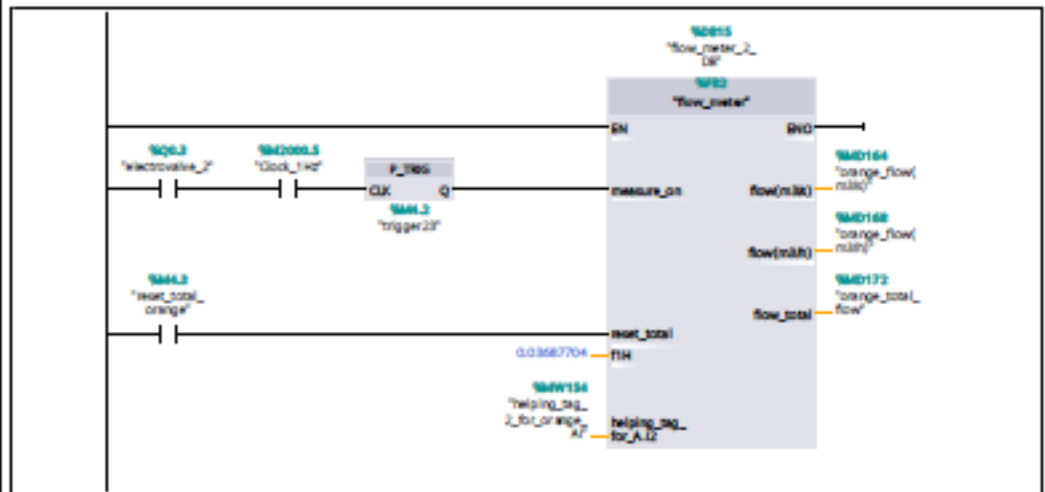
Network 13: orange_level_animation_for_discharging_with_electrovalve2



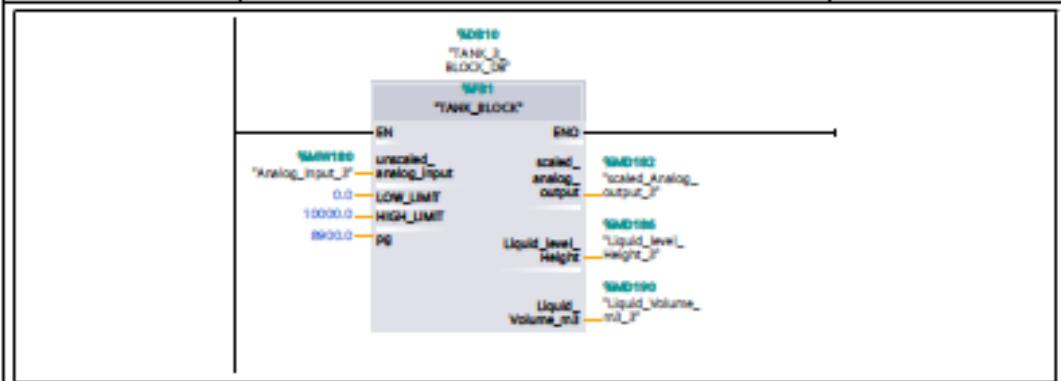
Network 14: calibration_of_analog_input_for_orange



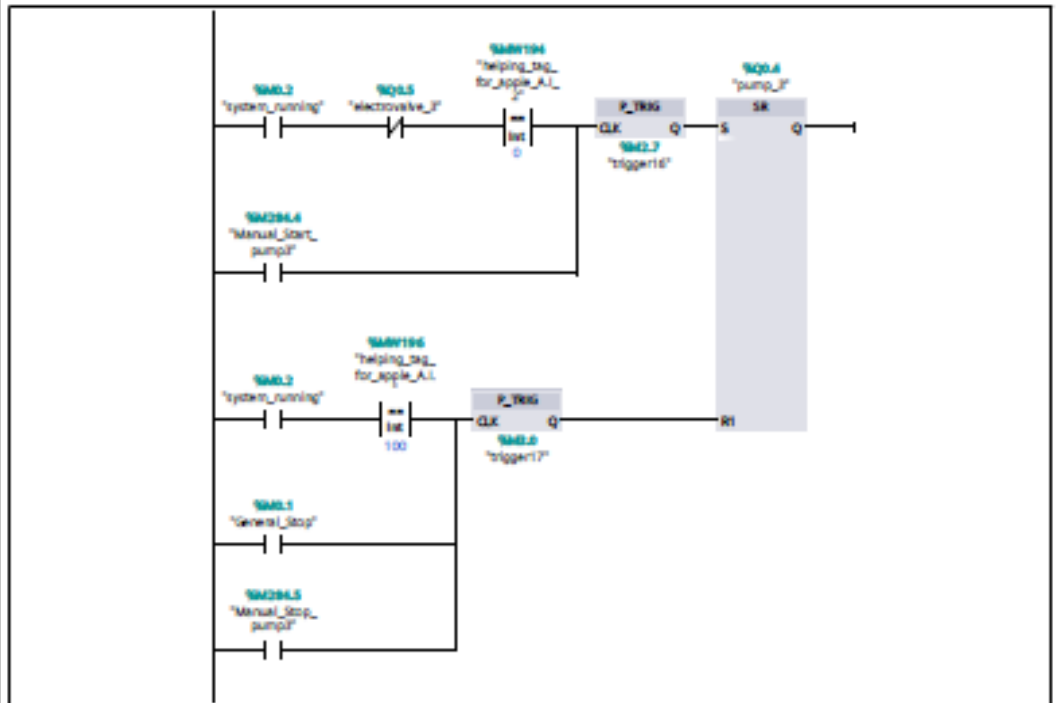
Network 15: flow_totalizer_orange



Network 16: apple_tank3_block

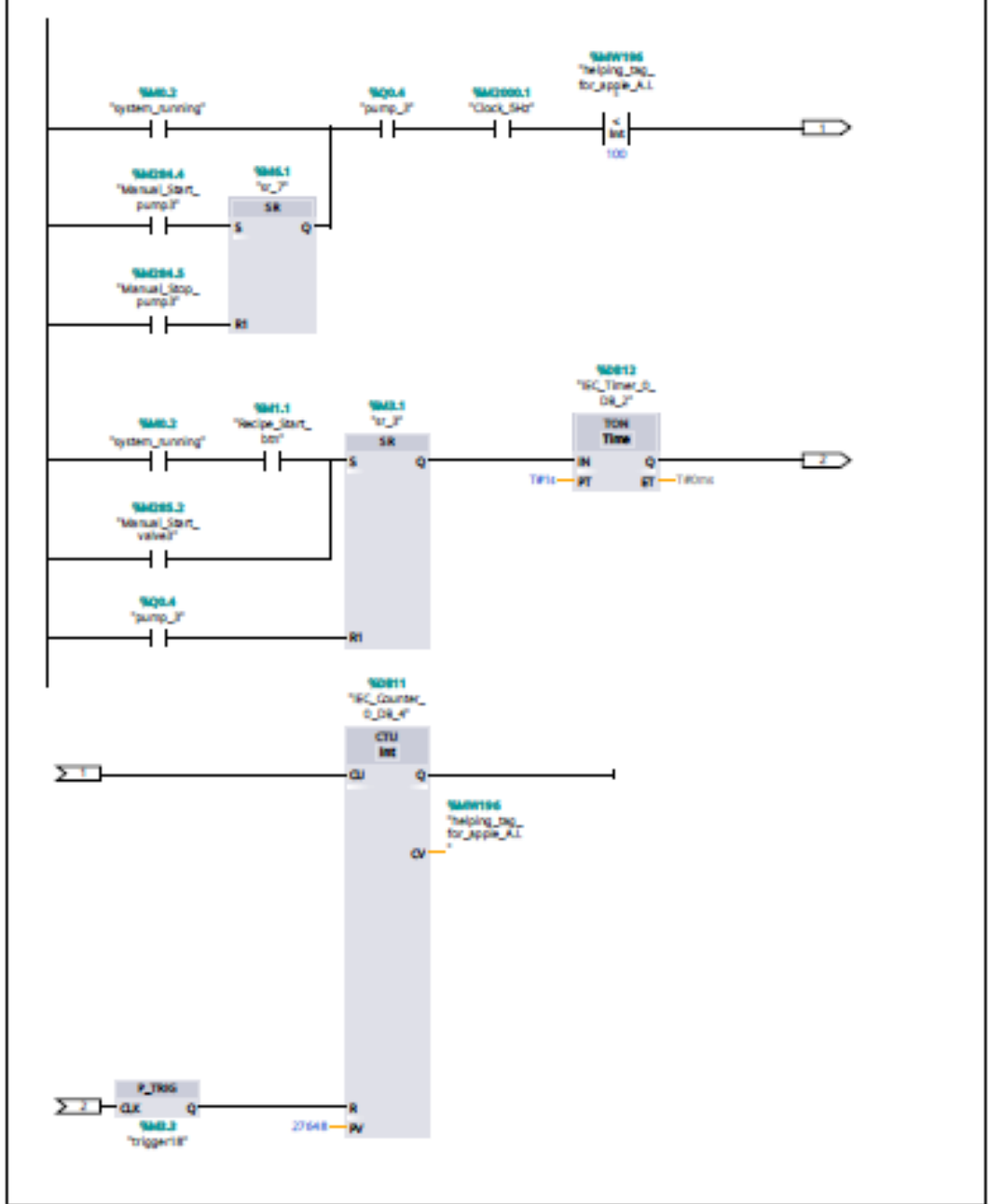


Network 17: apple_pump3_control

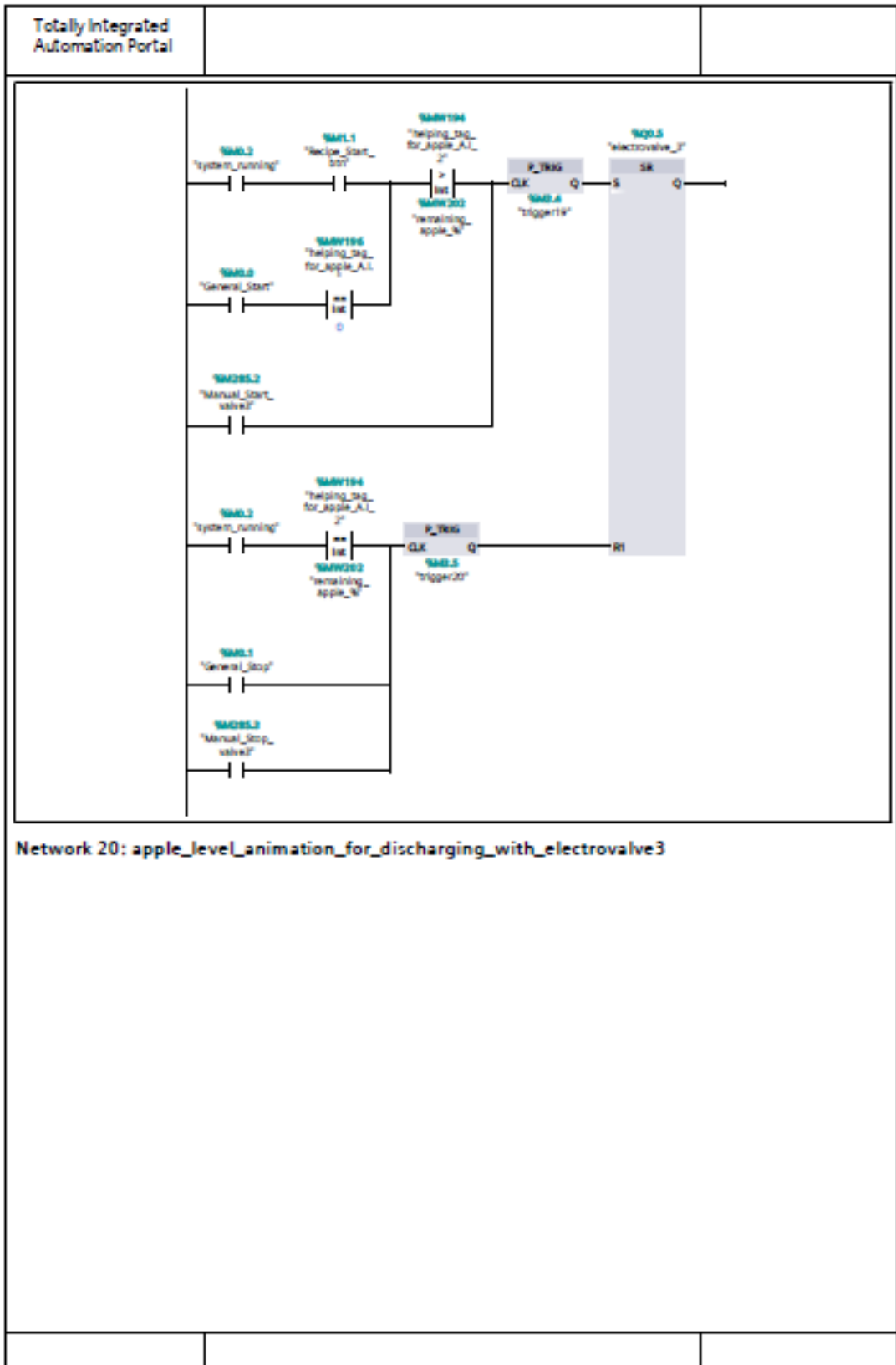


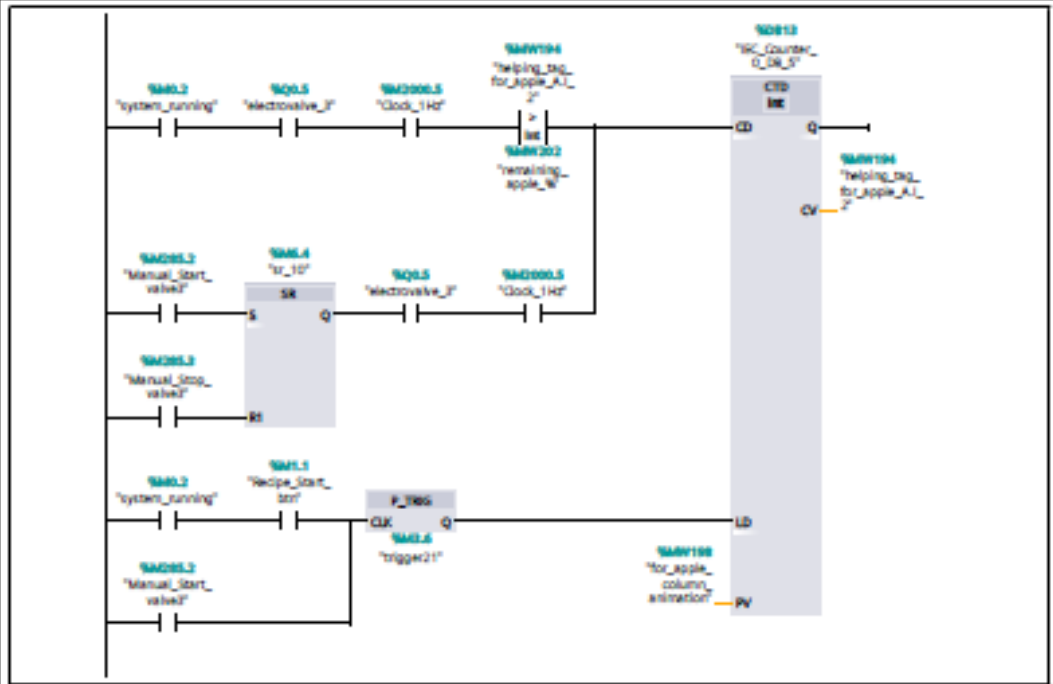
Network 18: APPLE_LEVEL_ANIMATION_FOR_FILLING_WITH_PUMP3

Network 18: APPLE_LEVEL_ANIMATION_FOR_FILLING_WITH_PUMP3

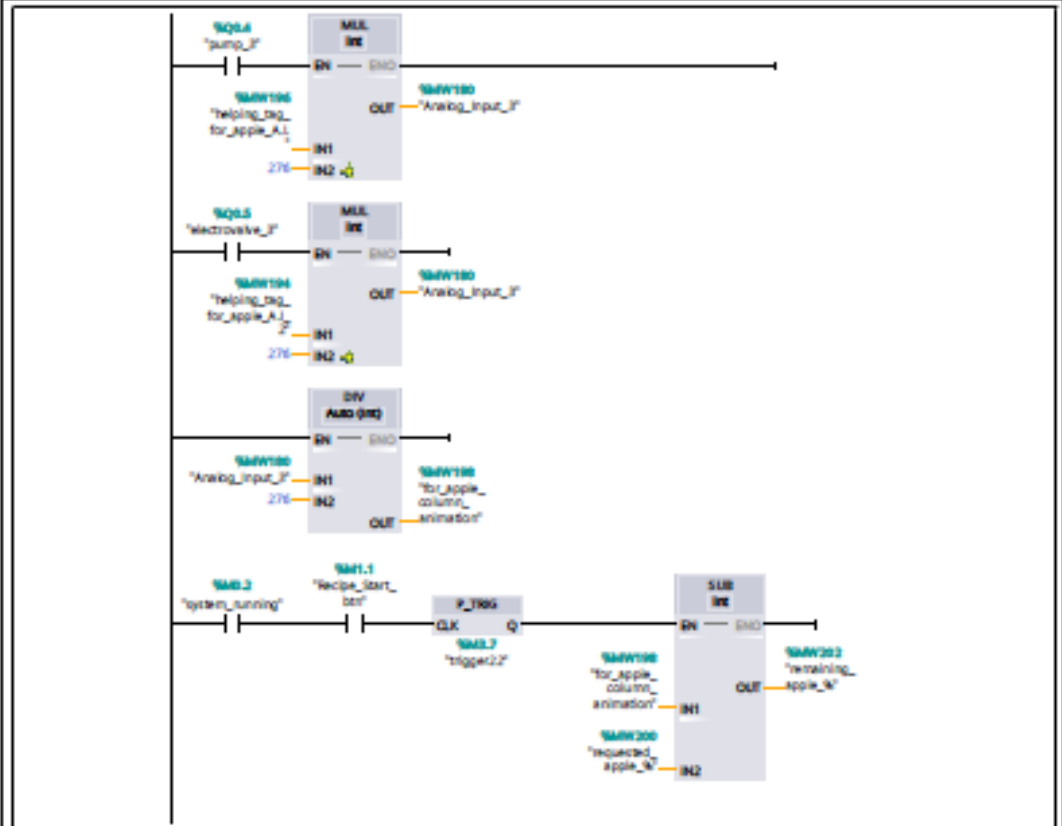


Network 19: apple_electrovalve3_control

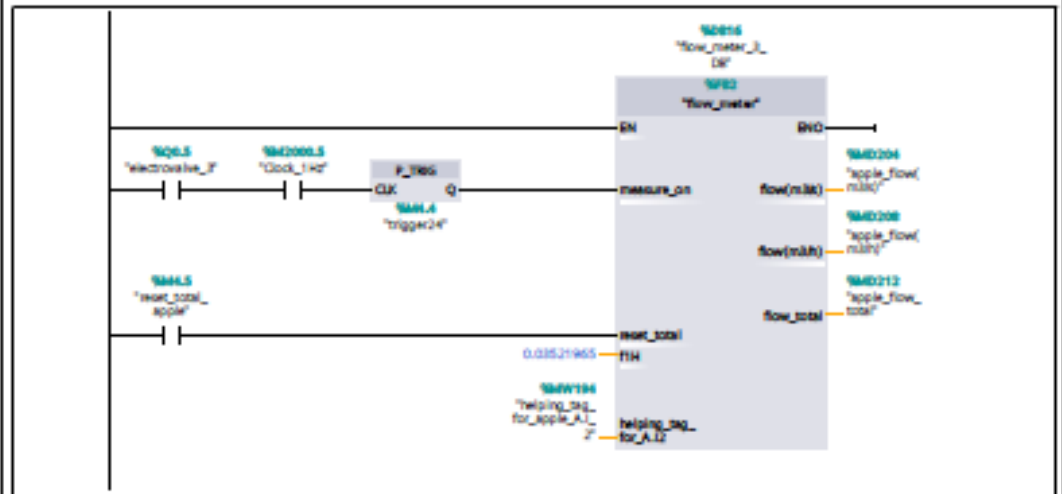




Network 21: calibration_of_analog_input_for_apple

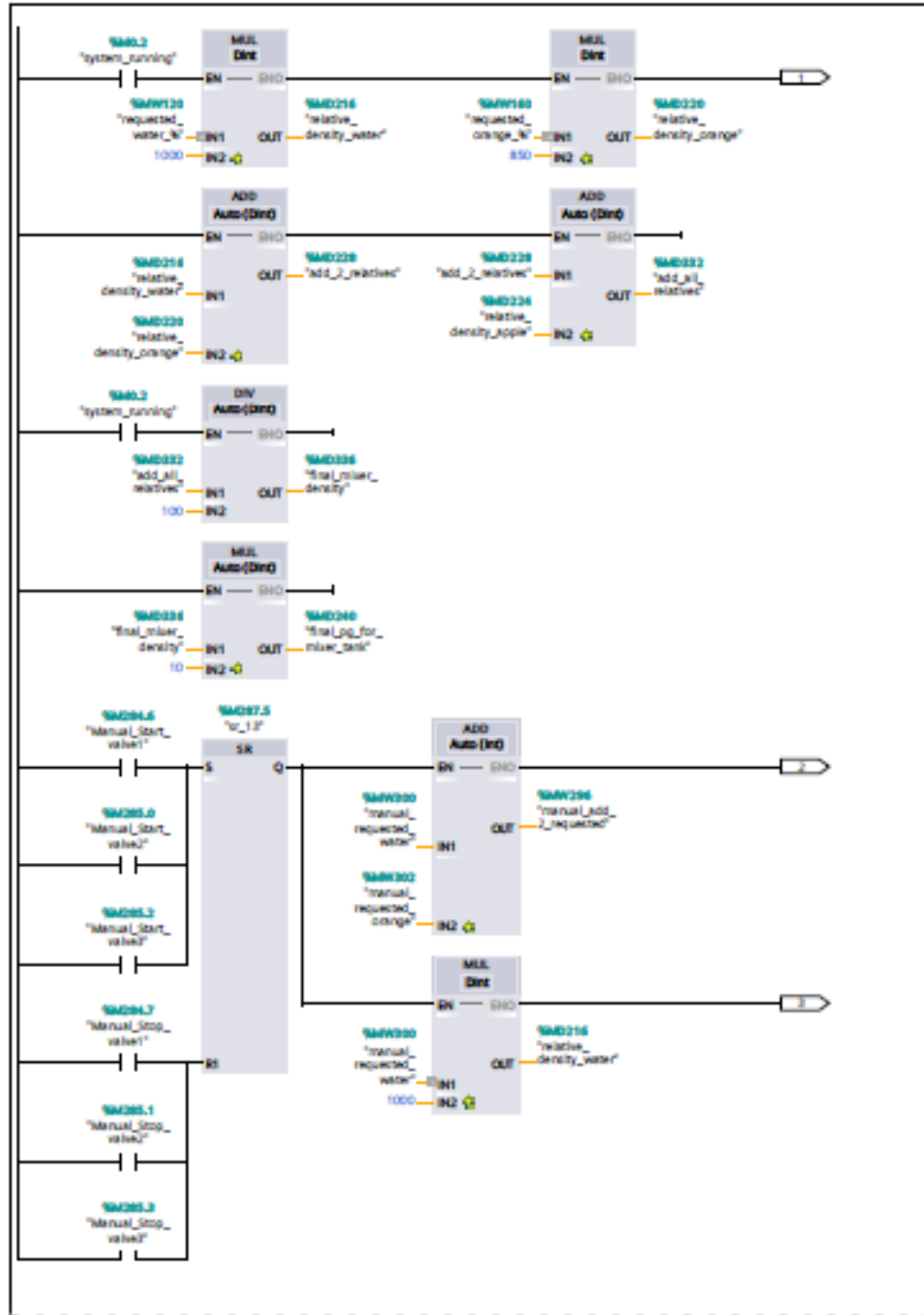


Network 22: flow_totalizer_apple



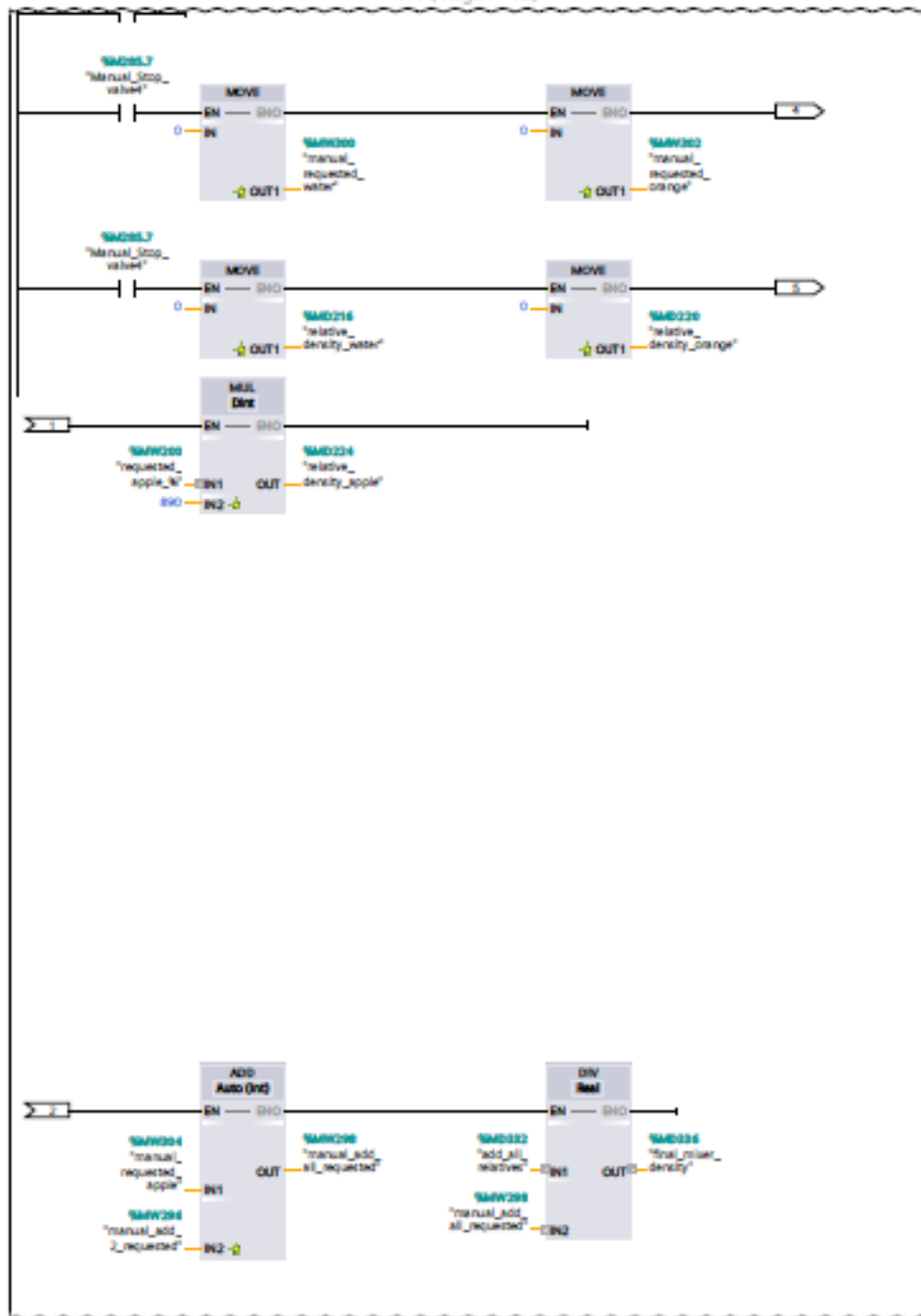
Totally Integrated Automation Portal		
<p>Network 23: liquid_density_inside_mixer_calculation_both_for_auto_and_manual</p> <p>Network 23:liquid_density_inside_mixer_calculation_both_for_auto_and_manual</p>		

Network 23: liquid_density_inside_mixer_calculation_both_for_auto_and_manual (1.1 / 3.1)



Network 23: liquid_density_inside_mixer_calculation_both_for_auto_and_manual (2.1 / 3.1)

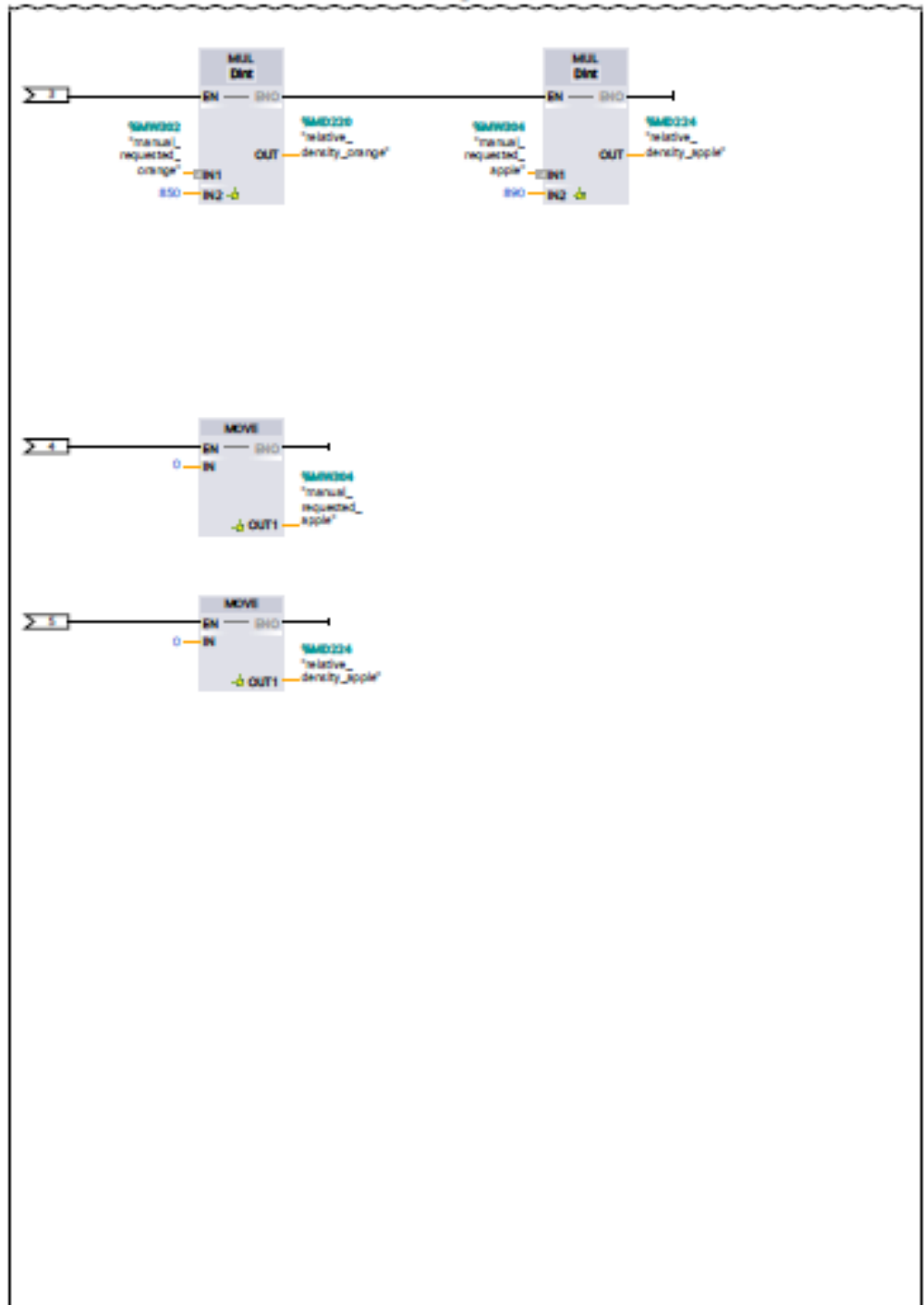
1.1 (Page1 - 18)



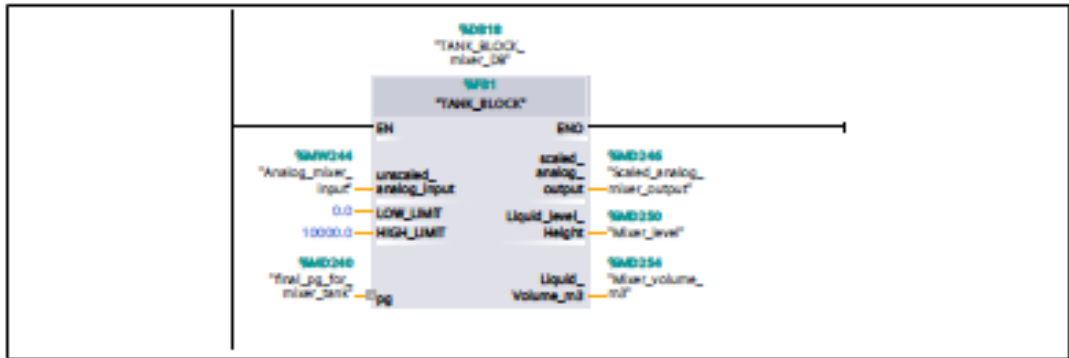
3.1 (Page1 - 20)

Network 23: liquid_density_inside_mixer_calculation_both_for_auto_and_manual (3.1 / 3.1)

2.1 (Page1 - 19)

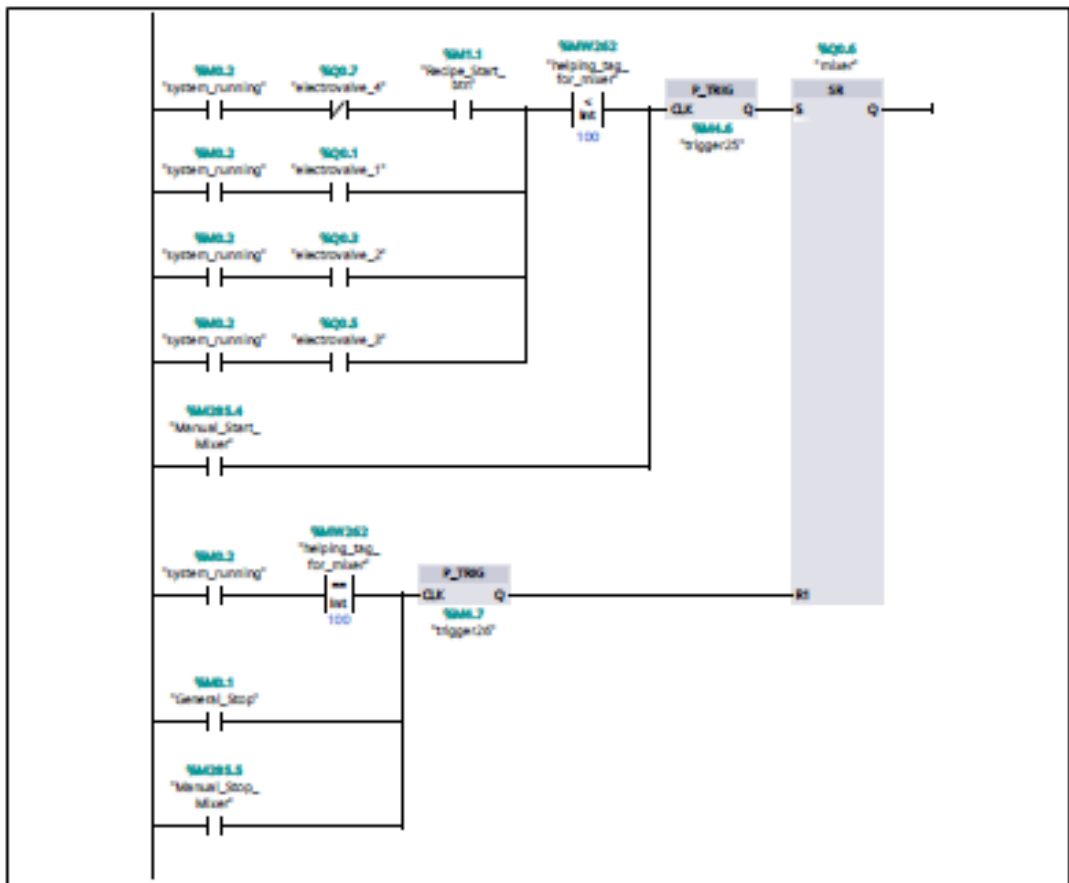


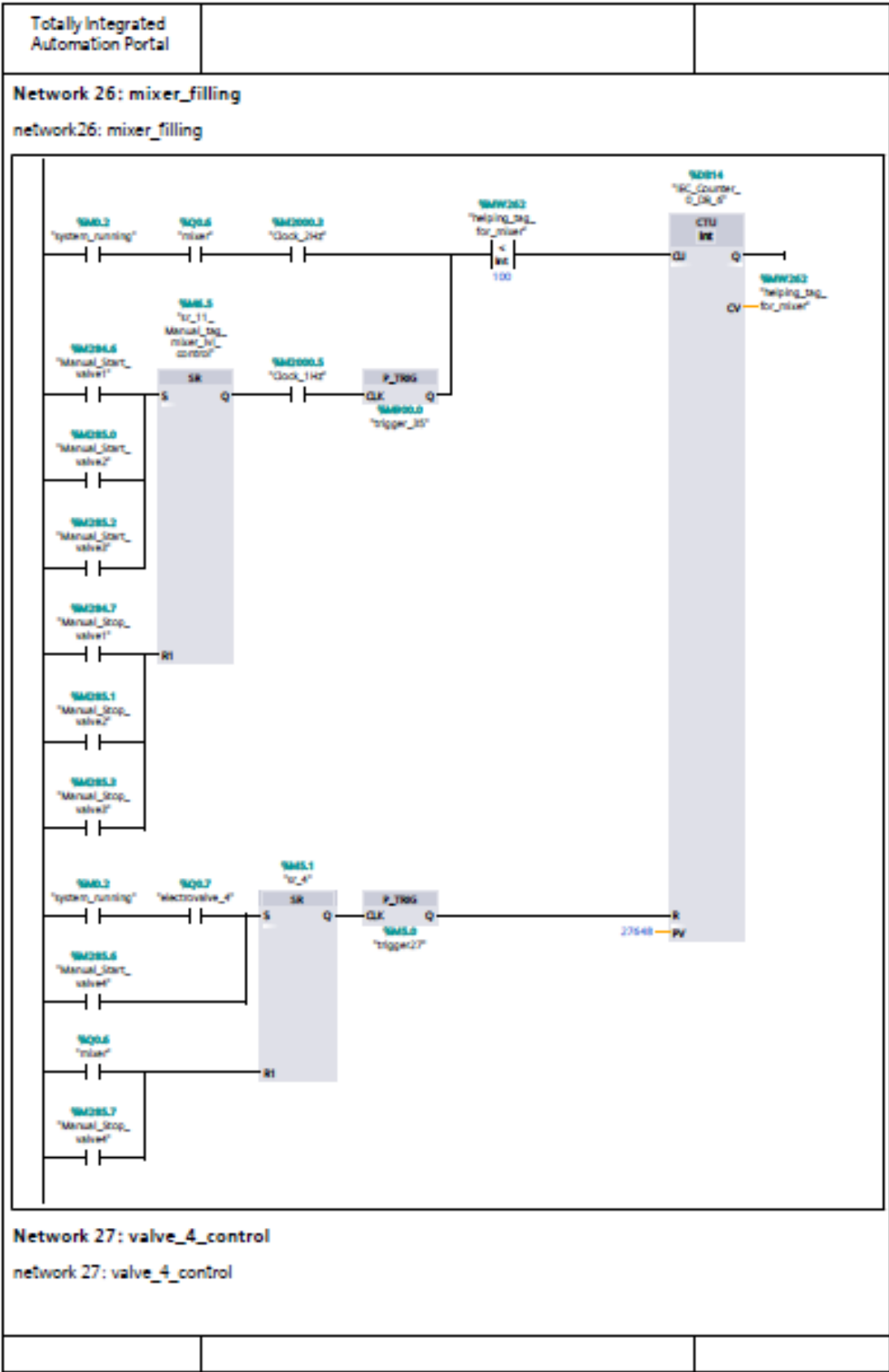
Network 24: tank_mixer_block

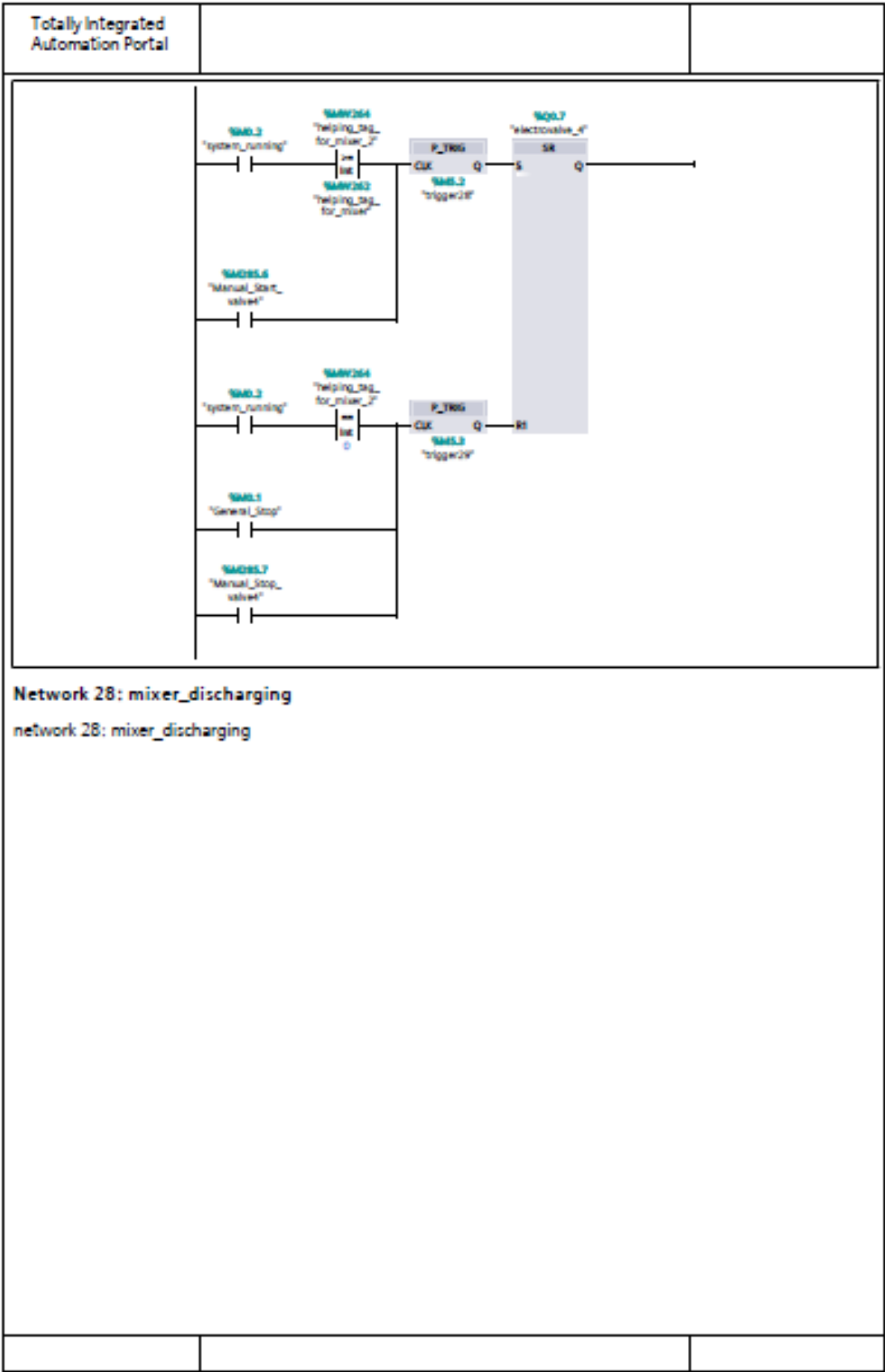


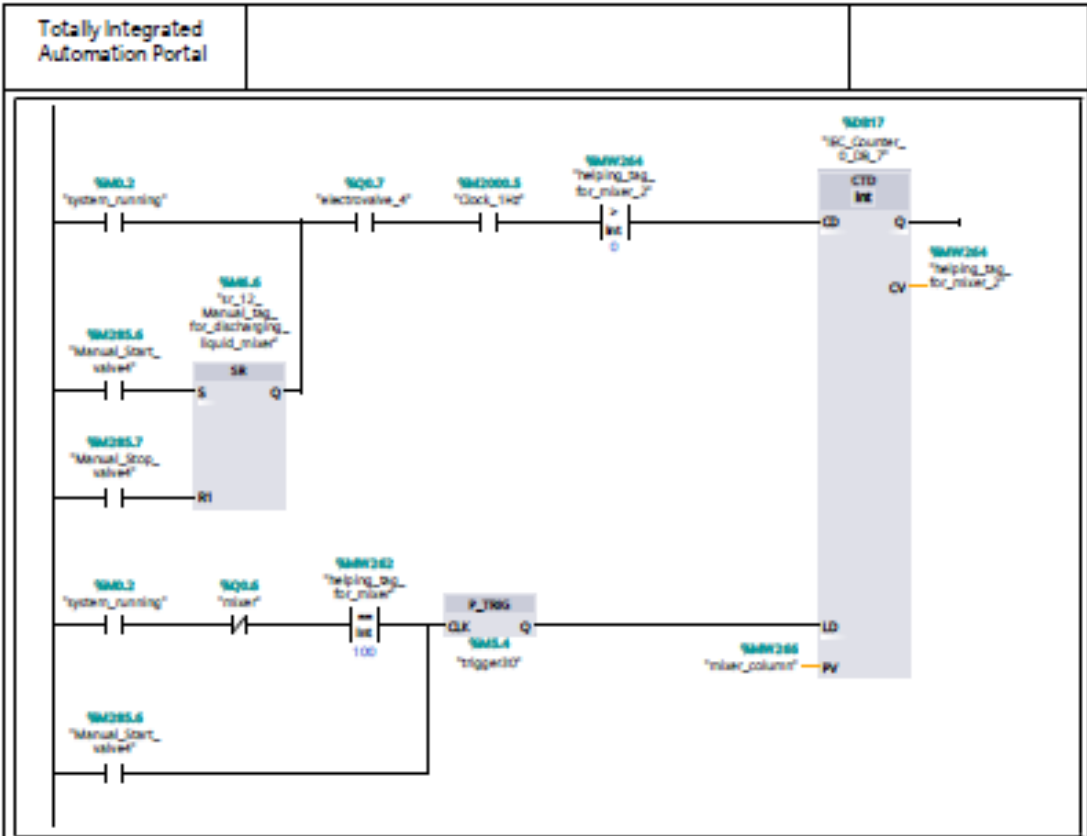
Network 25: mixer_control

Network25: mixer_control

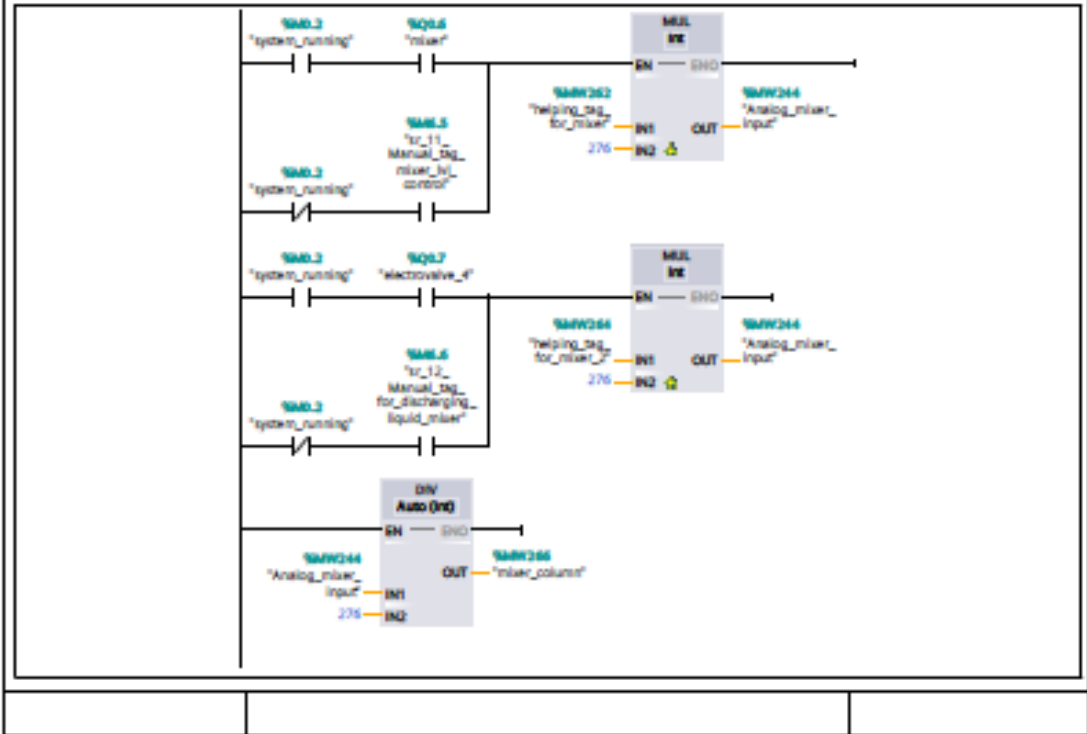




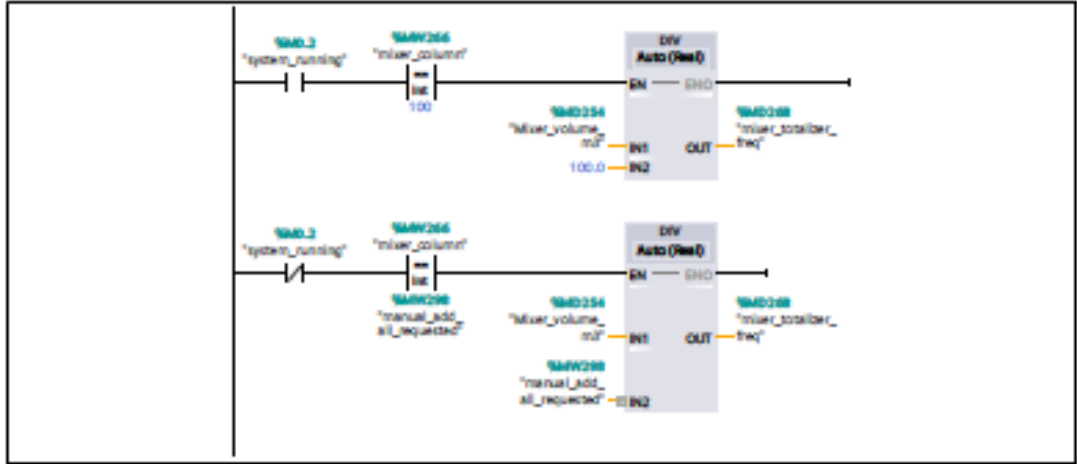




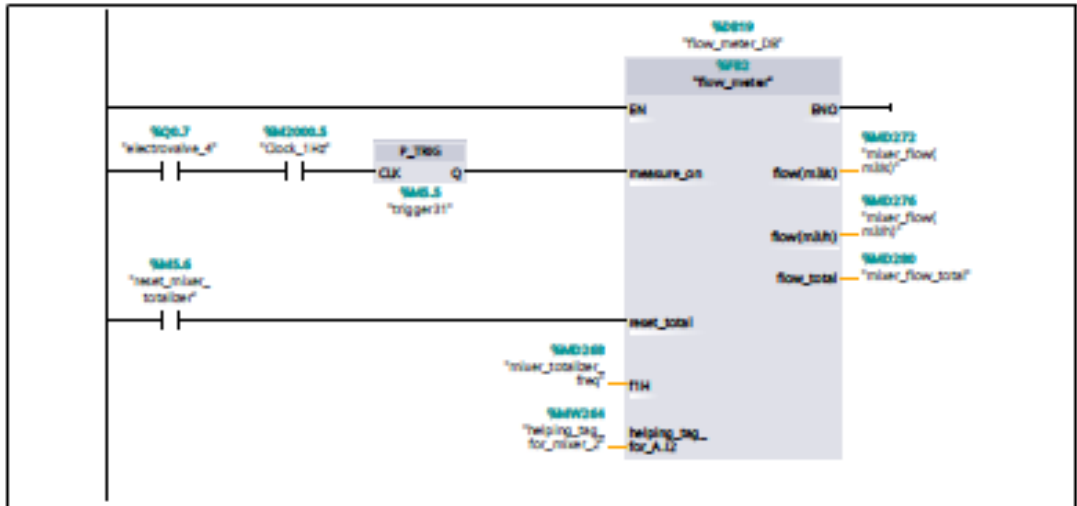
Network 29: calibration of analog input for mixer



Network 30: mixer_totalizer_calculation

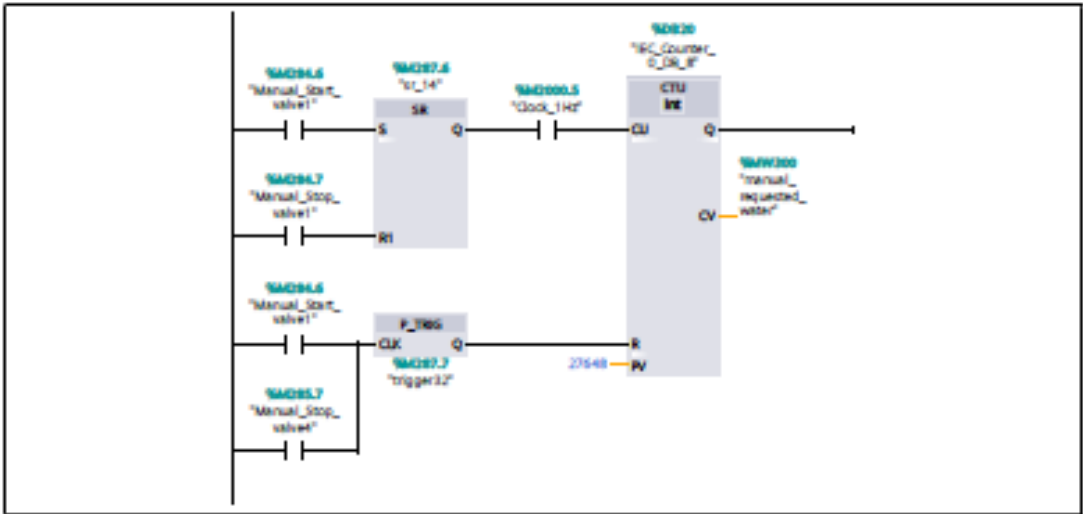


Network 31: flow_totalizer_mixer

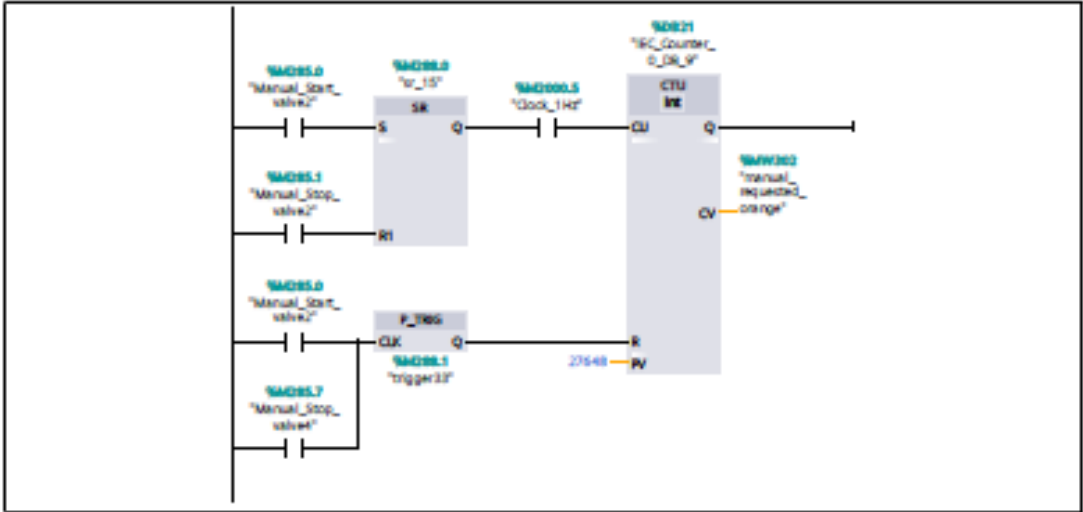


Network 32: manual_water_valve_request

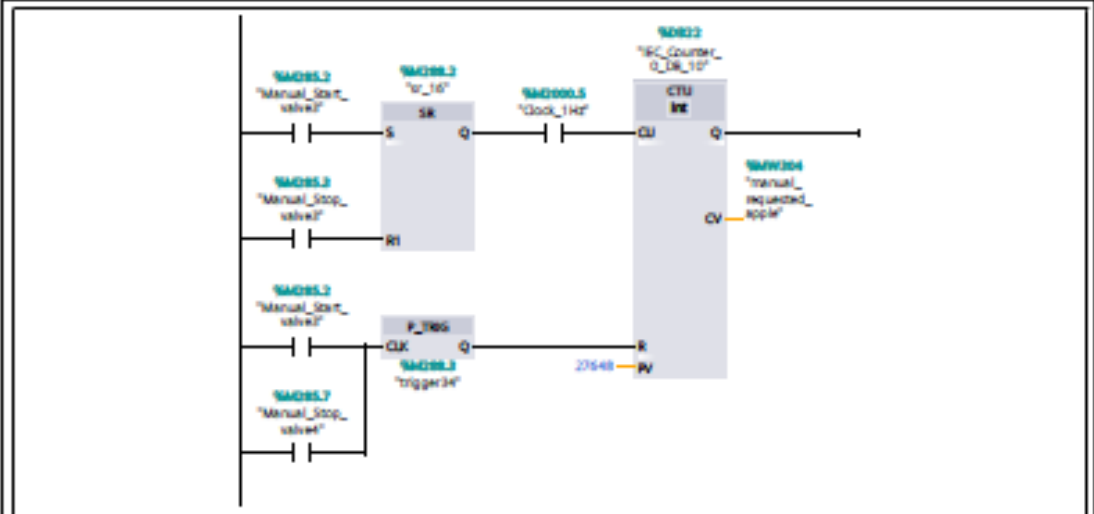
manual_water_valve_request



Network 33: manual_orange_valve_request



Network 34: manual_apple_valve_request



ΠΑΡΑΡΤΗΜΑ Β : ΚΩΔΙΚΑΣ CUSTOM SCADA

```
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;
using S7.Net;
using Microsoft.Data.SqlClient;

namespace xaml
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        int pos = 0;

        private void selector_btn_Click(object sender, EventArgs e)
        {
            pos += 1;
            if (pos == 1)
            {
                Bitmap b1 = new
                Bitmap(@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΕ\2.jpg");
                selector_btn.Image = b1;

                using (var plc = new Plc(CpuType.S71200,
                "127.0.0.1", 0, 1))
                {
                    plc.Open();
                    plc.Write("M288.4", true);
                }
                string connectionString = "Data Source=LAPTOP-
                8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
                Security=True;TrustServerCertificate=True";
                string sqlQuery = "INSERT INTO Actions_Data(Auto)
                VALUES(@Auto)";
                SqlConnection con = new
                SqlConnection(connectionString);
                con.Open();
                SqlCommand cmd = new SqlCommand(sqlQuery, con);
                var AutoParameter = new SqlParameter("Auto",
                System.Data.SqlDbType.DateTime);
                AutoParameter.Value = DateTime.Now;
                cmd.Parameters.Add(AutoParameter);
                cmd.ExecuteNonQuery();
                con.Close();

                Manual_Start_pump1.Enabled = false;
                Manual_Stop_pump1.Enabled = false;
                Manual_Start_pump2.Enabled = false;
                Manual_Stop_pump2.Enabled = false;
            }
        }
    }
}
```

```

Manual_Start_pump3.Enabled = false;
Manual_Stop_pump3.Enabled = false;
Manual_Start_mixer.Enabled = false;
Manual_Stop_mixer.Enabled = false;
Manual_Start_valve1.Enabled = false;
Manual_Stop_valve1.Enabled = false;
Manual_Start_valve2.Enabled = false;
Manual_Stop_valve2.Enabled = false;
Manual_Start_valve3.Enabled= false;
Manual_Stop_valve3.Enabled = false;
Manual_Start_valve4.Enabled = false;
Manual_Stop_valve4.Enabled= false;
Recipe_btn.Enabled = false;
General_Start.Enabled = true;
General_Stop.Enabled= false;
requested_water.Enabled = false;
requested_orange.Enabled = false;
requested_apple.Enabled = false;
Res_Tot_water.Enabled = true;
Res_Tot_Orange.Enabled = true;
Reset_Tot_Apple.Enabled = true;
Res_Tot_Mixer.Enabled = true;
Operator_Actions_History.Enabled = false;
Signals_History.Enabled = false;
Auto_Control_Panel.Enabled = true;
Manual_Control_Panel.Enabled = false;
GroupBox_Historical_Data.Enabled = false;
}
else if (pos == 2)
{
    Bitmap b2 = new
Bitmap(@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\3.jpg");
    selector_btn.Image = b2;

    using (var plc = new Plc(CpuType.S71200,
"127.0.0.1", 0, 1))
    {
        plc.Open();
        plc.Write("M288.4", false);
    }
    string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
    string sqlQuery = "INSERT INTO
Actions_Data(Manual) VALUES(@Manual)";
    SqlConnection con = new
SqlConnection(connectionString);
    con.Open();
    SqlCommand cmd = new SqlCommand(sqlQuery, con);
    var ManualParameter = new SqlParameter("Manual",
System.Data.SqlDbType.DateTime);
    ManualParameter.Value = DateTime.Now;
    cmd.Parameters.Add(ManualParameter);
    cmd.ExecuteNonQuery();
    con.Close();

    Manual_Start_pump1.Enabled = true;
    Manual_Stop_pump1.Enabled = true;
    Manual_Start_pump2.Enabled = true;
    Manual_Stop_pump2.Enabled = true;

```

```

Manual_Start_pump3.Enabled = true;
Manual_Stop_pump3.Enabled = true;
Manual_Start_mixer.Enabled = true;
Manual_Stop_mixer.Enabled = true;
Manual_Start_valve1.Enabled = true;
Manual_Stop_valve1.Enabled = true;
Manual_Start_valve2.Enabled = true;
Manual_Stop_valve2.Enabled = true;
Manual_Start_valve3.Enabled = true;
Manual_Stop_valve3.Enabled = true;
Manual_Start_valve4.Enabled = true;
Manual_Stop_valve4.Enabled = true;
Recipe_btn.Enabled = false;
General_Start.Enabled = false;
General_Stop.Enabled = false;
requested_water.Enabled = false;
requested_orange.Enabled = false;
requested_apple.Enabled = false;
Res_Tot_water.Enabled = true;
Res_Tot_Orange.Enabled = true;
Reset_Tot_Apple.Enabled = true;
Res_Tot_Mixer.Enabled = true;
Operator_Actions_History.Enabled = false;
Signals_History.Enabled = false;
Auto_Control_Panel.Enabled = false;
Manual_Control_Panel.Enabled = true;
GroupBox_Historical_Data.Enabled = false;
}
else if (pos == 3)
{
    Bitmap b3 = new
Bitmap(@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\1.jpg");
    selector_btn.Image = b3;

    using (var plc = new Plc(CpuType.S71200,
"127.0.0.1", 0, 1))
    {
        plc.Open();
        plc.Write("M288.4", false);
        pos = 0;
    }
    string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
    string sqlQuery = "INSERT INTO
Actions_Data (Buttons_Off) VALUES (@Buttons_Off)";
    SqlConnection con = new
SqlConnection(connectionString);
    con.Open();
    SqlCommand cmd = new SqlCommand(sqlQuery, con);
    var Buttons_OffParameter = new
SqlParameter("Buttons_Off", System.Data.SqlDbType.DateTime);
    Buttons_OffParameter.Value = DateTime.Now;
    cmd.Parameters.Add(Buttons_OffParameter);
    cmd.ExecuteNonQuery();
    con.Close();

    Manual_Start_pump1.Enabled = false;
    Manual_Stop_pump1.Enabled = false;
    Manual_Start_pump2.Enabled = false;

```

```

Manual_Stop_pump2.Enabled = false;
Manual_Start_pump3.Enabled = false;
Manual_Stop_pump3.Enabled = false;
Manual_Start_mixer.Enabled = false;
Manual_Stop_mixer.Enabled = false;
Manual_Start_valve1.Enabled = false;
Manual_Stop_valve1.Enabled = false;
Manual_Start_valve2.Enabled = false;
Manual_Stop_valve2.Enabled = false;
Manual_Start_valve3.Enabled = false;
Manual_Stop_valve3.Enabled = false;
Manual_Start_valve4.Enabled = false;
Manual_Stop_valve4.Enabled = false;
Recipe_btn.Enabled = false;
General_Start.Enabled = false;
General_Stop.Enabled = false;
requested_water.Enabled = false;
requested_orange.Enabled = false;
requested_apple.Enabled = false;
Res_Tot_water.Enabled = false;
Res_Tot_Orange.Enabled = false;
Reset_Tot_Apple.Enabled = false;
Res_Tot_Mixer.Enabled = false;
Operator_Actions_History.Enabled = true;
Signals_History.Enabled = true;
Auto_Control_Panel.Enabled = false;
Manual_Control_Panel.Enabled = false;
GroupBox_Historical_Data.Enabled = true;
}
}
private void General_Start_Click(object sender, EventArgs
e)
{
using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
{
plc.Open();
plc.Write("M0.0", true);
plc.Write("M0.0", false);
}

string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
string sqlQuery = "INSERT INTO
Actions_Data(General_Start) VALUES(@General_Start)";
SqlConnection con = new
SqlConnection(connectionString);
con.Open();
SqlCommand cmd = new SqlCommand(sqlQuery, con);
var General_StartParameter = new
SqlParameter("General_Start", System.Data.SqlDbType.DateTime);
General_StartParameter.Value = DateTime.Now;
cmd.Parameters.Add(General_StartParameter);
cmd.ExecuteNonQuery();
con.Close();
}
}

```

```

        Recipe_btn.Enabled = true;
        requested_water.Enabled = true;
        requested_orange.Enabled = true;
        requested_apple.Enabled = true;
        selector_btn.Enabled = false;
        General_Start.Enabled = false;
        General_Stop.Enabled = true;
    }
    private void General_Stop_Click(object sender, EventArgs e)
    {
        using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
        {
            plc.Open();
            plc.Write("M0.1", true);
            plc.Write("M0.1", false);
        }
        string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO
Actions_Data(General_Stop) VALUES(@General_Stop)";
        SqlConnection con = new
SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var General_StopParameter = new
SqlParameter("General_Stop", System.Data.SqlDbType.DateTime);
        General_StopParameter.Value = DateTime.Now;
        cmd.Parameters.Add(General_StopParameter);
        cmd.ExecuteNonQuery();
        con.Close();

        Recipe_btn.Enabled = false;
        requested_water.Enabled = false;
        requested_orange.Enabled = false;
        requested_apple.Enabled = false;
        selector_btn.Enabled = true;
        General_Start.Enabled = true;
    }
    private void Recipe_btn_Click(object sender, EventArgs e)
    {
        int a, b, c, d, f, g, i;
        int.TryParse(requested_water.Text, out a);
        int.TryParse(requested_orange.Text, out b);
        int.TryParse(requested_apple.Text, out c);
        int.TryParse(tank1Volumepercentage.Text, out d);
        int.TryParse(tank2Volumepercentage.Text, out f);
        int.TryParse(tank3Volumepercentage.Text, out g);
        i = a + b + c;
        if (i != 100 || requested_water.Text == "" ||
requested_orange.Text == "" || requested_apple.Text == "")
        {
            using (var plc = new Plc(CpuType.S71200,
"127.0.0.1", 0, 1))
            {
                plc.Open();
            }
        }
    }
}

```

```

        plc.Write("M0.1", true);
        plc.Write("M0.1", false);
    }
    MessageBox.Show("Wrong Values!Please insert values
with sum=100 ,press General Start and start recipe button again");
    General_Start.Enabled = true;
    Recipe_btn.Enabled = true;
}
else if (a != 50 || b > f || c > g)
{
    using (var plc = new Plc(CpuType.S71200,
"127.0.0.1", 0, 1))
    {
        plc.Open();
        plc.Write("M0.1", true);
        plc.Write("M0.1", false);
    }
    MessageBox.Show("Wrong Values!Please check if
requested amount(%) of water is 50% and if the other two %requested
amounts are less than or equal to the %remaining amounts,then press
General Start and start recipe button again");
    General_Start.Enabled = true;
    Recipe_btn.Enabled = true;
}
else
{
    using (var plc = new Plc(CpuType.S71200,
"127.0.0.1", 0, 1))
    {
        plc.Open();
        string address = ("MW120");
        object setpoint = requested_water.Text;
        setpoint =
Convert.ToUInt16(requested_water.Text);
        plc.Write(address, setpoint);
        string address1 = ("MW160");
        object setpoint1 = requested_orange.Text;
        setpoint1 =
Convert.ToUInt16(requested_orange.Text);
        plc.Write(address1, setpoint1);
        string address2 = ("MW200");
        object setpoint2 = requested_apple.Text;
        setpoint2 =
Convert.ToUInt16(requested_apple.Text);
        plc.Write(address2, setpoint2);
        plc.Write("M1.1", true);
        plc.Write("M1.1", false);
    }

    string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
    string sqlQuery = "INSERT INTO
Actions_Data(Recipe_Start,requested_water,requested_orange,request
ed_apple)

```

```

VALUES (@Recipe_Start,@requested_water,@requested_orange,@requested
_apple)";

        SqlConnection con = new
SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var Recipe_StartParameter = new
SqlParameter("Recipe_Start", System.Data.SqlDbType.DateTime);
        Recipe_StartParameter.Value = DateTime.Now;
        cmd.Parameters.Add(Recipe_StartParameter);

        var requested_waterParameter = new
SqlParameter("requested_water", System.Data.SqlDbType.VarChar);
        requested_waterParameter.Value =
requested_water.Text;
        cmd.Parameters.Add(requested_waterParameter);

        var requested_orangeParameter = new
SqlParameter("requested_orange", System.Data.SqlDbType.VarChar);
        requested_orangeParameter.Value =
requested_orange.Text;
        cmd.Parameters.Add(requested_orangeParameter);

        var requested_appleParameter = new
SqlParameter("requested_apple", System.Data.SqlDbType.VarChar);
        requested_appleParameter.Value =
requested_apple.Text;
        cmd.Parameters.Add(requested_appleParameter);

        cmd.ExecuteNonQuery();
        con.Close();

        General_Stop.Enabled = true;
    }
}
private void Res_Tot_water_Click(object sender, EventArgs
e)
{
    using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
    {
        plc.Open();
        plc.Write("M1.4", true);
        plc.Write("M1.4", false);
    }

    string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
    string sqlQuery = "INSERT INTO
Actions_Data(Reset_Tot_water) VALUES (@Reset_Tot_water)";
    SqlConnection con = new
SqlConnection(connectionString);
    con.Open();
    SqlCommand cmd = new SqlCommand(sqlQuery, con);
    var Reset_Tot_waterParameter = new
SqlParameter("Reset_Tot_water", System.Data.SqlDbType.DateTime);

```

```

        Reset_Tot_waterParameter.Value = DateTime.Now;
        cmd.Parameters.Add(Reset_Tot_waterParameter);
        cmd.ExecuteNonQuery();
        con.Close();
    }
    private void Res_Tot_Orange_Click(object sender, EventArgs
e)
    {
        using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
        {
            plc.Open();
            plc.Write("M4.3", true);
            plc.Write("M4.3", false);
        }
        string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO
Actions_Data(Reset_Tot_orange) VALUES(@Reset_Tot_orange)";
        SqlConnection con = new
SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var Reset_Tot_orangeParameter = new
SqlParameter("Reset_Tot_orange", System.Data.SqlDbType.DateTime);
        Reset_Tot_orangeParameter.Value = DateTime.Now;
        cmd.Parameters.Add(Reset_Tot_orangeParameter);
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void Reset_Tot_Apple_Click(object sender, EventArgs
e)
    {
        using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
        {
            plc.Open();
            plc.Write("M4.5", true);
            plc.Write("M4.5", false);
        }
        string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO
Actions_Data(Reset_Tot_apple) VALUES(@Reset_Tot_apple)";
        SqlConnection con = new
SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var Reset_Tot_appleParameter = new
SqlParameter("Reset_Tot_apple", System.Data.SqlDbType.DateTime);
        Reset_Tot_appleParameter.Value = DateTime.Now;
        cmd.Parameters.Add(Reset_Tot_appleParameter);
        cmd.ExecuteNonQuery();
        con.Close();
    }
}

```



```

private void Res_Tot_Mixer_Click(object sender, EventArgs
e)
{
    using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
    {
        plc.Open();
        plc.Write("M5.6", true);
        plc.Write("M5.6", false);
    }
    string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
    string sqlQuery = "INSERT INTO
Actions_Data(Reset_Tot_mixer) VALUES(@Reset_Tot_mixer)";
    SqlConnection con = new
SqlConnection(connectionString);
    con.Open();
    SqlCommand cmd = new SqlCommand(sqlQuery, con);
    var Reset_Tot_mixerParameter = new
SqlParameter("Reset_Tot_mixer", System.Data.SqlDbType.DateTime);
    Reset_Tot_mixerParameter.Value = DateTime.Now;
    cmd.Parameters.Add(Reset_Tot_mixerParameter);
    cmd.ExecuteNonQuery();
    con.Close();
}

private void timer1_Tick(object sender, EventArgs e)
{
    using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
    {
        plc.Open();

        string address = ("MW90");
        object result = plc.Read(address);
        tank1Volumepercentage.Text = string.Format("{0}",
result.ToString());
        userControl11.value =
Int16.Parse(tank1Volumepercentage.Text);
        userControl11.changeValue();

        var
result1=((uint)plc.Read("MD106")).ConvertToFloat();
        tank1_Height.Text = result1.ToString();

        var result2 =
((uint)plc.Read("MD110")).ConvertToFloat();
        tank1Volume.Text = result2.ToString();

        var result5 =
((uint)plc.Read("MD132")).ConvertToFloat();
        flow_total_water.Text = result5.ToString();

        bool result6 = (bool)plc.Read("Q0.0");
        if (result6 == true)
        {

```

```

        pump1.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\pump_greenn.jpg";

    }
    else
    {
        pump1.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\Horizontal pump
(right).jpg";

    }
    bool result7 = (bool)plc.Read("Q0.1");
    if (result7 == true)
    {
        electrovalve1.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\electrovalve_on.jpg";

    }
    else
    {
        electrovalve1.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\Control valve -
vertical.jpg";

    }

    string address1 = ("MW158");
    object result10 = plc.Read(address1);
    tank2Volumepercentage.Text = string.Format("{0}",
result10.ToString());
    userControl12.value =
Int16.Parse(tank2Volumepercentage.Text);
    userControl12.changeValue();

    var result8 =
((uint)plc.Read("MD150")).ConvertToFloat();
    tank2Volume.Text = result8.ToString();

    var result9 =
((uint)plc.Read("MD146")).ConvertToFloat();
    tank2_Height.Text = result9.ToString();

    var result19 =
((uint)plc.Read("MD172")).ConvertToFloat();
    flow_total_orange.Text = result19.ToString();

    bool result11 = (bool)plc.Read("Q0.2");
    if (result11 == true)
    {
        pump2.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\pump_greenn.jpg";

    }
    else
    {
        pump2.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\Horizontal pump
(right).jpg";

```

```

    }

    bool result12 = (bool)plc.Read("Q0.3");
    if (result12 == true)
    {
        electrovalve2.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\electrovalve_on.jpg";
    }
    else
    {
        electrovalve2.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\Control valve -
vertical.jpg";
    }

    string address2 = ("MW198");
    object result15 = plc.Read(address2);
    tank3Volumepercentage.Text = string.Format("{0}",
result15.ToString());
    userControl13.value =
Int16.Parse(tank3Volumepercentage.Text);
    userControl13.changeValue();

    var result13 =
((uint)plc.Read("MD186")).ConvertToFloat();
    tank3_Height.Text = result13.ToString();

    var result14 =
((uint)plc.Read("MD190")).ConvertToFloat();
    tank3Volume.Text = result14.ToString();

    var result20 =
((uint)plc.Read("MD212")).ConvertToFloat();
    flow_total_apple.Text = result20.ToString();

    bool result16 = (bool)plc.Read("Q0.4");
    if (result16 == true)
    {
        pump3.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\pump_green.jpg";
    }
    else
    {
        pump3.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\Horizontal pump
(right).jpg";
    }

    bool result17 = (bool)plc.Read("Q0.5");
    if (result17 == true)
    {
        electrovalve3.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\electrovalve_on.jpg";
    }

```

```

else
{
    electrovalve3.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\Control valve -
vertical.jpg";
}

var result21 =
((uint)plc.Read("MD250")).ConvertToFloat();
tank4_Height.Text = result21.ToString();

var result22 =
((uint)plc.Read("MD254")).ConvertToFloat();
tank4Volume.Text = result22.ToString();

string address3 = ("MW266");
object result23 = plc.Read(address3);
tank4Volumepercentage.Text = string.Format("{0}",
result23.ToString());
userControl14.value =
Int16.Parse(tank4Volumepercentage.Text);
userControl14.changeValue();

bool result18 = (bool)plc.Read("Q0.6");
if (result18 == true)
{
    mixer.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\mixer_on.jpg";
}
else
{
    mixer.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\3-D Mixer.jpg";
}

bool result24 = (bool)plc.Read("Q0.7");
if (result24 == true)
{
    electrovalve4.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\electrovalve_on.jpg";
}
else
{
    electrovalve4.ImageLocation =
@"C:\Users\sakis\Desktop\ΔΙΠΛΩΜΑΤΙΚΗ ΔΙΠΛΑΕ\Control valve -
vertical.jpg";
}

var result25 =
((uint)plc.Read("MD280")).ConvertToFloat();
flow_total_mixture.Text = result25.ToString();

```

```

        string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO
Signals_Data(Date_Time,tank1_volume,tank1_volpercent,tank1_height,
tank1_flow,tank2_volume,tank2_volpercent,tank2_height,tank2_flow,t
ank3_volume,tank3_volpercent,tank3_height,tank3_flow,tank4_volume,
tank4_volpercent,tank4_height,tank4_flow,pump1,pump2,pump3,mixer,v
alve1, valve2, valve3, valve4)
VALUES(@Date_Time,@tank1_volume,@tank1_volpercent,@tank1_height,@t
ank1_flow,@tank2_volume,@tank2_volpercent,@tank2_height,@tank2_flo
w,@tank3_volume,@tank3_volpercent,@tank3_height,@tank3_flow,@tank4
_volume,@tank4_volpercent,@tank4_height,@tank4_flow,@pump1,@pump2,
@pump3,@mixer,@valve1,@valve2,@valve3,@valve4)";
        SqlConnection con = new
SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var Date_TimeParameter = new
SqlParameter("Date_Time", System.Data.SqlDbType.DateTime);
        Date_TimeParameter.Value= DateTime.Now;
        cmd.Parameters.Add(Date_TimeParameter);
        var tank1_volumeParameter = new
SqlParameter("tank1_volume", System.Data.SqlDbType.VarChar);
        tank1_volumeParameter.Value = tank1Volume.Text;
        cmd.Parameters.Add(tank1_volumeParameter);
        var tank1_volumepercentageParameter = new
SqlParameter("tank1_volpercent", System.Data.SqlDbType.VarChar);
        tank1_volumepercentageParameter.Value
        =
tank1Volumepercentage.Text;

        cmd.Parameters.Add(tank1_volumepercentageParameter);
        var tank1_heightParameter = new
SqlParameter("tank1_height", System.Data.SqlDbType.VarChar);
        tank1_heightParameter.Value = tank1_Height.Text;
        cmd.Parameters.Add(tank1_heightParameter);
        var tank1_flowParameter = new
SqlParameter("tank1_flow", System.Data.SqlDbType.VarChar);
        tank1_flowParameter.Value = flow_total_water.Text;
        cmd.Parameters.Add(tank1_flowParameter);

        var tank2_volumeParameter = new
SqlParameter("tank2_volume", System.Data.SqlDbType.VarChar);
        tank2_volumeParameter.Value = tank2Volume.Text;
        cmd.Parameters.Add(tank2_volumeParameter);
        var tank2_volumepercentageParameter = new
SqlParameter("tank2_volpercent", System.Data.SqlDbType.VarChar);
        tank2_volumepercentageParameter.Value =
tank2Volumepercentage.Text;

        cmd.Parameters.Add(tank2_volumepercentageParameter);
        var tank2_heightParameter = new
SqlParameter("tank2_height", System.Data.SqlDbType.VarChar);
        tank2_heightParameter.Value = tank2_Height.Text;
        cmd.Parameters.Add(tank2_heightParameter);
        var tank2_flowParameter = new
SqlParameter("tank2_flow", System.Data.SqlDbType.VarChar);
        tank2_flowParameter.Value = flow_total_orange.Text;
        cmd.Parameters.Add(tank2_flowParameter);

```

```

        var tank3_volumeParameter = new
SqlParameter("tank3_volume", System.Data.SqlDbType.VarChar);
        tank3_volumeParameter.Value = tank3Volume.Text;
        cmd.Parameters.Add(tank3_volumeParameter);
        var tank3_volumepercentageParameter = new
SqlParameter("tank3_volpercent", System.Data.SqlDbType.VarChar);
        tank3_volumepercentageParameter.Value =
tank3Volumepercentage.Text;

cmd.Parameters.Add(tank3_volumepercentageParameter);
        var tank3_heightParameter = new
SqlParameter("tank3_height", System.Data.SqlDbType.VarChar);
        tank3_heightParameter.Value = tank3_Height.Text;
        cmd.Parameters.Add(tank3_heightParameter);
        var tank3_flowParameter = new
SqlParameter("tank3_flow", System.Data.SqlDbType.VarChar);
        tank3_flowParameter.Value = flow_total_apple.Text;
        cmd.Parameters.Add(tank3_flowParameter);

        var tank4_volumeParameter = new
SqlParameter("tank4_volume", System.Data.SqlDbType.VarChar);
        tank4_volumeParameter.Value = tank4Volume.Text;
        cmd.Parameters.Add(tank4_volumeParameter);
        var tank4_volumepercentageParameter = new
SqlParameter("tank4_volpercent", System.Data.SqlDbType.VarChar);
        tank4_volumepercentageParameter.Value =
tank4Volumepercentage.Text;

cmd.Parameters.Add(tank4_volumepercentageParameter);
        var tank4_heightParameter = new
SqlParameter("tank4_height", System.Data.SqlDbType.VarChar);
        tank4_heightParameter.Value = tank4_Height.Text;
        cmd.Parameters.Add(tank4_heightParameter);
        var tank4_flowParameter = new
SqlParameter("tank4_flow", System.Data.SqlDbType.VarChar);
        tank4_flowParameter.Value
=
flow_total_mixture.Text;
        cmd.Parameters.Add(tank4_flowParameter);

        var pump1Parameter = new SqlParameter("pump1",
System.Data.SqlDbType.VarChar);
        pump1Parameter.Value = result6;
        cmd.Parameters.Add(pump1Parameter);

        var pump2Parameter = new SqlParameter("pump2",
System.Data.SqlDbType.VarChar);
        pump2Parameter.Value = result11;
        cmd.Parameters.Add(pump2Parameter);

        var pump3Parameter = new SqlParameter("pump3",
System.Data.SqlDbType.VarChar);
        pump3Parameter.Value = result16;
        cmd.Parameters.Add(pump3Parameter);

        var mixerParameter = new SqlParameter("mixer",
System.Data.SqlDbType.VarChar);
        mixerParameter.Value = result18;
        cmd.Parameters.Add(mixerParameter);

```

```

        var valve1Parameter = new SqlParameter("valve1",
System.Data.SqlDbType.VarChar);
        valve1Parameter.Value = result7;
        cmd.Parameters.Add(valve1Parameter);

        var valve2Parameter = new SqlParameter("valve2",
System.Data.SqlDbType.VarChar);
        valve2Parameter.Value = result12;
        cmd.Parameters.Add(valve2Parameter);

        var valve3Parameter = new SqlParameter("valve3",
System.Data.SqlDbType.VarChar);
        valve3Parameter.Value = result17;
        cmd.Parameters.Add(valve3Parameter);

        var valve4Parameter = new SqlParameter("valve4",
System.Data.SqlDbType.VarChar);
        valve4Parameter.Value = result24;
        cmd.Parameters.Add(valve4Parameter);

        cmd.ExecuteNonQuery();
        con.Close();

        if (pos == 1 && result18 == true || pos == 1 &&
result24 == true|| pos==1 && result6==true || pos==1 &&
result11==true || pos==1 && result16==true)
        {
            selector_btn.Enabled = false;
            Recipe_btn.Enabled = false;
        }
        else if (pos == 1 && result18 == false &&
tank4Volumepercentage.Text == "0")
        {
            selector_btn.Enabled = true;
            Recipe_btn.Enabled = true;
        }
        else if( pos == 0 || pos == 3 || pos == 2 &&
tank1Volumepercentage.Text == "0" && tank2Volumepercentage.Text ==
"0" && tank3Volumepercentage.Text == "0" &&
tank4Volumepercentage.Text == "0")
        {
            selector_btn.Enabled = true;
        }
        else
        {
            selector_btn.Enabled = false;
        }

        if(pos==2 && tank1Volumepercentage.Text == "0")
        {
            Manual_Start_pump1.Enabled = true;
        }
        else
        {
            Manual_Start_pump1.Enabled = false;
        }

        if (pos == 2 && tank2Volumepercentage.Text == "0")
        {

```

```

        Manual_Start_pump2.Enabled = true;
    }
    else
    {
        Manual_Start_pump2.Enabled = false;
    }
    if (pos == 2 && tank3Volumepercentage.Text == "0")
    {
        Manual_Start_pump3.Enabled = true;
    }
    else
    {
        Manual_Start_pump3.Enabled = false;
    }
}
}

private void Manual_Start_pump1_Click(object sender,
EventArgs e)
{
    using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
    {
        plc.Open();
        plc.Write("M284.0", true);
        plc.Write("M284.0", false);
    }
    string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
    string sqlQuery = "INSERT INTO
Actions_Data(start_pump1) VALUES(@start_pump1)";
    SqlConnection con = new
SqlConnection(connectionString);
    con.Open();
    SqlCommand cmd = new SqlCommand(sqlQuery, con);
    var start_pump1Parameter = new
SqlParameter("start_pump1", System.Data.SqlDbType.DateTime);
    start_pump1Parameter.Value = DateTime.Now;
    cmd.Parameters.Add(start_pump1Parameter);
    cmd.ExecuteNonQuery();
    con.Close();
}

private void Manual_Stop_pump1_Click(object sender,
EventArgs e)
{
    using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
    {
        plc.Open();
        plc.Write("M284.1", true);
        plc.Write("M284.1", false);
    }
}

```



```

        string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO
Actions_Data(stop_pump1) VALUES(@stop_pump1)";
        SqlConnection con = new
SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var stop_pump1Parameter = new
SqlParameter("stop_pump1", System.Data.SqlDbType.DateTime);
        stop_pump1Parameter.Value = DateTime.Now;
        cmd.Parameters.Add(stop_pump1Parameter);
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void Manual_Start_pump2_Click(object sender,
EventArgs e)
    {
        using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
        {
            plc.Open();
            plc.Write("M284.2", true);
            plc.Write("M284.2", false);
        }
        string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO
Actions_Data(start_pump2) VALUES(@start_pump2)";
        SqlConnection con = new
SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var start_pump2Parameter = new
SqlParameter("start_pump2", System.Data.SqlDbType.DateTime);
        start_pump2Parameter.Value = DateTime.Now;
        cmd.Parameters.Add(start_pump2Parameter);
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void Manual_Stop_pump2_Click(object sender,
EventArgs e)
    {
        using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
        {
            plc.Open();
            plc.Write("M284.3", true);
            plc.Write("M284.3", false);
        }
        string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";

```

```

        string sqlQuery = "INSERT INTO Actions_Data(stop_pump2)
VALUES(@stop_pump2)";
        SqlConnection con = new
SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var stop_pump2Parameter = new
SqlParameter("stop_pump2", System.Data.SqlDbType.DateTime);
        stop_pump2Parameter.Value = DateTime.Now;
        cmd.Parameters.Add(stop_pump2Parameter);
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void Manual_Start_pump3_Click(object sender,
EventArgs e)
    {
        using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
        {
            plc.Open();
            plc.Write("M284.4", true);
            plc.Write("M284.4", false);
        }
        string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO
Actions_Data(start_pump3) VALUES(@start_pump3)";
        SqlConnection con = new
SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var start_pump3Parameter = new
SqlParameter("start_pump3", System.Data.SqlDbType.DateTime);
        start_pump3Parameter.Value = DateTime.Now;
        cmd.Parameters.Add(start_pump3Parameter);
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void Manual_Stop_pump3_Click(object sender,
EventArgs e)
    {
        using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
        {
            plc.Open();
            plc.Write("M284.5", true);
            plc.Write("M284.5", false);
        }
        string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO Actions_Data(stop_pump3)
VALUES(@stop_pump3)";
        SqlConnection con = new
SqlConnection(connectionString);

```

```

        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var stop_pump3Parameter = new
        SqlParameter("stop_pump3", System.Data.SqlDbType.DateTime);
        stop_pump3Parameter.Value = DateTime.Now;
        cmd.Parameters.Add(stop_pump3Parameter);
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void Manual_Start_valve1_Click(object sender,
    EventArgs e)
    {
        using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
    0, 1))
        {
            plc.Open();
            plc.Write("M284.6", true);
            plc.Write("M284.6", false);
        }

        string connectionString = "Data Source=LAPTOP-
    8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
    Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO
    Actions_Data(start_valve1) VALUES(@start_valve1)";
        SqlConnection con = new
        SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var start_valve1Parameter = new
        SqlParameter("start_valve1", System.Data.SqlDbType.DateTime);
        start_valve1Parameter.Value = DateTime.Now;
        cmd.Parameters.Add(start_valve1Parameter);
        cmd.ExecuteNonQuery();
        con.Close();

        Manual_Start_valve1.Enabled = false;
    }

    private void Manual_Stop_valve1_Click(object sender,
    EventArgs e)
    {
        using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
    0, 1))
        {
            plc.Open();
            plc.Write("M284.7", true);
            plc.Write("M284.7", false);
        }

        string connectionString = "Data Source=LAPTOP-
    8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
    Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO
    Actions_Data(stop_valve1) VALUES(@stop_valve1)";
        SqlConnection con = new
        SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);

```

```

        var stop_valve1Parameter = new
        SqlParameter("stop_valve1", System.Data.SqlDbType.DateTime);
        stop_valve1Parameter.Value = DateTime.Now;
        cmd.Parameters.Add(stop_valve1Parameter);
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void Manual_Start_valve2_Click(object sender,
    EventArgs e)
    {
        using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
    0, 1))
        {
            plc.Open();
            plc.Write("M285.0", true);
            plc.Write("M285.0", false);
        }

        string connectionString = "Data Source=LAPTOP-
    8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
    Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO
    Actions_Data(start_valve2) VALUES(@start_valve2)";
        SqlConnection con = new
    SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var start_valve2Parameter = new
    SqlParameter("start_valve2", System.Data.SqlDbType.DateTime);
        start_valve2Parameter.Value = DateTime.Now;
        cmd.Parameters.Add(start_valve2Parameter);
        cmd.ExecuteNonQuery();
        con.Close();

        Manual_Start_valve2.Enabled = false;
    }

    private void Manual_Stop_valve2_Click(object sender,
    EventArgs e)
    {
        using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
    0, 1))
        {
            plc.Open();
            plc.Write("M285.1", true);
            plc.Write("M285.1", false);
        }

        string connectionString = "Data Source=LAPTOP-
    8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
    Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO
    Actions_Data(stop_valve2) VALUES(@stop_valve2)";
        SqlConnection con = new
    SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var stop_valve2Parameter = new
    SqlParameter("stop_valve2", System.Data.SqlDbType.DateTime);

```

```

        stop_valve2Parameter.Value = DateTime.Now;
        cmd.Parameters.Add(stop_valve2Parameter);
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void Manual_Start_valve3_Click(object sender,
EventArgs e)
    {
        using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
        {
            plc.Open();
            plc.Write("M285.2", true);
            plc.Write("M285.2", false);
        }

        string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO
Actions_Data(start_valve3) VALUES(@start_valve3)";
        SqlConnection con = new
SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var start_valve3Parameter = new
SqlParameter("start_valve3", System.Data.SqlDbType.DateTime);
        start_valve3Parameter.Value = DateTime.Now;
        cmd.Parameters.Add(start_valve3Parameter);
        cmd.ExecuteNonQuery();
        con.Close();

        Manual_Start_valve3.Enabled = false;
    }

    private void Manual_Stop_valve3_Click(object sender,
EventArgs e)
    {
        using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
        {
            plc.Open();
            plc.Write("M285.3", true);
            plc.Write("M285.3", false);
        }

        string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO
Actions_Data(stop_valve3) VALUES(@stop_valve3)";
        SqlConnection con = new
SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var stop_valve3Parameter = new
SqlParameter("stop_valve3", System.Data.SqlDbType.DateTime);
        stop_valve3Parameter.Value = DateTime.Now;
        cmd.Parameters.Add(stop_valve3Parameter);
    }

```

```

        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void Manual_Start_mixer_Click(object sender,
EventArgs e)
    {
        using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
        {
            plc.Open();
            plc.Write("M285.4", true);
            plc.Write("M285.4", false);
        }

        string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO
Actions_Data(start_mixer) VALUES(@start_mixer)";
        SqlConnection con = new
SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var start_mixerParameter = new
SqlParameter("start_mixer", System.Data.SqlDbType.DateTime);
        start_mixerParameter.Value = DateTime.Now;
        cmd.Parameters.Add(start_mixerParameter);
        cmd.ExecuteNonQuery();
        con.Close();
    }

    private void Manual_Stop_mixer_Click(object sender,
EventArgs e)
    {
        using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
        {
            plc.Open();
            plc.Write("M285.5", true);
            plc.Write("M285.5", false);
        }

        string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
        string sqlQuery = "INSERT INTO Actions_Data(stop_mixer)
VALUES(@stop_mixer)";
        SqlConnection con = new
SqlConnection(connectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand(sqlQuery, con);
        var stop_mixerParameter = new
SqlParameter("stop_mixer", System.Data.SqlDbType.DateTime);
        stop_mixerParameter.Value = DateTime.Now;
        cmd.Parameters.Add(stop_mixerParameter);
        cmd.ExecuteNonQuery();
        con.Close();
    }
}

```

```

        private void Manual_Start_valve4_Click(object sender,
EventArgs e)
        {
            using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
            {
                plc.Open();
                plc.Write("M285.6", true);
                plc.Write("M285.6", false);
            }
            string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
            string sqlQuery = "INSERT INTO
Actions_Data(start_mixer_valve4) VALUES(@start_mixer_valve4)";
            SqlConnection con = new
SqlConnection(connectionString);
            con.Open();
            SqlCommand cmd = new SqlCommand(sqlQuery, con);
            var start_mixer_valve4Parameter = new
SqlParameter("start_mixer_valve4",
System.Data.SqlDbType.DateTime);
            start_mixer_valve4Parameter.Value = DateTime.Now;
            cmd.Parameters.Add(start_mixer_valve4Parameter);
            cmd.ExecuteNonQuery();
            con.Close();
        }

        private void Manual_Stop_valve4_Click(object sender,
EventArgs e)
        {
            using (var plc = new Plc(CpuType.S71200, "127.0.0.1",
0, 1))
            {
                plc.Open();
                plc.Write("M285.7", true);
                plc.Write("M285.7", false);
            }
            string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
            string sqlQuery = "INSERT INTO
Actions_Data(stop_mixer_valve4) VALUES(@stop_mixer_valve4)";
            SqlConnection con = new
SqlConnection(connectionString);
            con.Open();
            SqlCommand cmd = new SqlCommand(sqlQuery, con);
            var stop_mixer_valve4Parameter = new
SqlParameter("stop_mixer_valve4", System.Data.SqlDbType.DateTime);
            stop_mixer_valve4Parameter.Value = DateTime.Now;
            cmd.Parameters.Add(stop_mixer_valve4Parameter);
            cmd.ExecuteNonQuery();
            con.Close();

            if(tank4Volumepercentage.Text == "0")
            {
                Manual_Start_valve1.Enabled = true;
            }
        }

```

```

        Manual_Start_valve2.Enabled = true;
        Manual_Start_valve3.Enabled = true;
    }
}

//communication with sql

private void Operator_Actions_History_Click(object sender,
EventArgs e)
{
    historical_Data_manual_Screen.Show();
    EXIT.Show();
    dataGridView1.Show();
    dateTimePicker1.Show();
    dateTimePicker2.Show();
    Search_Actions.Show();
    Search_Signals.Hide();
    dataGridView2.Hide();

    string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
    SqlConnection con = new
SqlConnection(connectionString);
    con.Open();
    SqlDataAdapter sqlDa = new SqlDataAdapter("SELECT * from
Actions_Data", con);
    DataTable dtbl = new DataTable();
    sqlDa.Fill(dtbl);
    dataGridView1.DataSource = dtbl;
    con.Close();
}
private void Signals_History_Click(object sender, EventArgs
e)
{
    historical_Data_manual_Screen.Show();
    EXIT.Show();
    dataGridView2.Show();
    dateTimePicker1.Show();
    dateTimePicker2.Show();
    Search_Signals.Show();
    dataGridView1.Hide();
    Search_Actions.Hide();

    string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
    SqlConnection con = new
SqlConnection(connectionString);
    con.Open();
    SqlDataAdapter sqlDa = new SqlDataAdapter("SELECT * from
Signals_Data", con);
    DataTable dtbl = new DataTable();
    sqlDa.Fill(dtbl);
    dataGridView2.DataSource = dtbl;
    con.Close();
}

private void Form1_Load(object sender, EventArgs e)
{

```



```

Manual_Start_pump1.Enabled = false;
Manual_Stop_pump1.Enabled = false;
Manual_Start_pump2.Enabled = false;
Manual_Stop_pump2.Enabled = false;
Manual_Start_pump3.Enabled = false;
Manual_Stop_pump3.Enabled = false;
Manual_Start_mixer.Enabled = false;
Manual_Stop_mixer.Enabled = false;
Manual_Start_valve1.Enabled = false;
Manual_Stop_valve1.Enabled = false;
Manual_Start_valve2.Enabled = false;
Manual_Stop_valve2.Enabled = false;
Manual_Start_valve3.Enabled = false;
Manual_Stop_valve3.Enabled = false;
Manual_Start_valve4.Enabled = false;
Manual_Stop_valve4.Enabled = false;
Recipe_btn.Enabled = false;
General_Start.Enabled = false;
General_Stop.Enabled = false;
requested_water.Enabled = false;
requested_orange.Enabled = false;
requested_apple.Enabled = false;
Res_Tot_water.Enabled = false;
Res_Tot_Orange.Enabled = false;
Reset_Tot_Apple.Enabled = false;
Res_Tot_Mixer.Enabled = false;
historical_Data_manual_Screen.Hide();
EXIT.Hide();
dataGridView1.Hide();
dataGridView2.Hide();
dateTimePicker1.Hide();
dateTimePicker2.Hide();
Search_Signals.Hide();
Search_Actions.Hide();
Operator_Actions_History.Enabled = true;
Signals_History.Enabled = true;
}

private void EXIT_Click(object sender, EventArgs e)
{
    historical_Data_manual_Screen.Hide();
    EXIT.Hide();
    dataGridView1.Hide();
    dataGridView2.Hide();
    dateTimePicker1.Hide();
    dateTimePicker2.Hide();
    Search_Signals.Hide();
    Search_Actions.Hide();
}

private void Search_Signals_Click(object sender, EventArgs
e)
{
    string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
    SqlConnection con = new
SqlConnection(connectionString);

```

```

        con.Open();
        SqlDataAdapter sqlDa = new SqlDataAdapter("SELECT *
from Signals_Data where Date_Time between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"', con);

        DataSet ds = new DataSet();
        sqlDa.Fill(ds, "Signals_Data");
        dataGridView2.DataSource = ds.Tables["Signals_Data"];
        con.Close();

    }

    private void Search_Actions_Click(object sender, EventArgs
e)
    {
        string connectionString = "Data Source=LAPTOP-
8VI576LNR\\SQLEXPRESS;Initial Catalog=storage;Integrated
Security=True;TrustServerCertificate=True";
        SqlConnection con = new
SqlConnection(connectionString);
        con.Open();
        SqlDataAdapter sqlDa = new SqlDataAdapter("SELECT *
from Actions_Data where Auto between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") + "'
or Manual between '" + dateTimePicker1.Value.ToString("yyyy- MM-
dd HH: mm:ss") + "' And '" + dateTimePicker2.Value.ToString("yyyy-
MM- dd HH: mm:ss") + "'or Buttons_Off between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or start_pump1 between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or stop_pump1 between '" + dateTimePicker1.Value.ToString("yyyy-
MM- dd HH: mm:ss") + "' And '" +
dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") + "'or
start_pump2 between '" + dateTimePicker1.Value.ToString("yyyy- MM-
dd HH: mm:ss") + "' And '" + dateTimePicker2.Value.ToString("yyyy-
MM- dd HH: mm:ss") + "'or stop_pump2 between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or start_pump3 between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or stop_pump3 between '" + dateTimePicker1.Value.ToString("yyyy-
MM- dd HH: mm:ss") + "' And '" +
dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") + "'or
start_valve1 between '" + dateTimePicker1.Value.ToString("yyyy-
MM- dd HH: mm:ss") + "' And '" +
dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") + "'or
stop_valve1 between '" + dateTimePicker1.Value.ToString("yyyy- MM-
dd HH: mm:ss") + "' And '" + dateTimePicker2.Value.ToString("yyyy-
MM- dd HH: mm:ss") + "'or start_valve2 between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or stop_valve2 between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or start_valve3 between '" +

```

```

dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or stop_valve3 between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or start_mixer between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or stop_mixer between '" + dateTimePicker1.Value.ToString("yyyy-
MM- dd HH: mm:ss") + "' And '" +
dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") + "'or
start_mixer_valve4 between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or stop_mixer_valve4 between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or General_Start between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or General_Stop between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or Recipe_Start between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or Reset_Tot_water between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or Reset_Tot_orange between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or Reset_Tot_apple between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'or Reset_Tot_mixer between '" +
dateTimePicker1.Value.ToString("yyyy- MM- dd HH: mm:ss") + "' And
'" + dateTimePicker2.Value.ToString("yyyy- MM- dd HH: mm:ss") +
"'", con);

        DataSet ds = new DataSet();
        sqlDa.Fill(ds, "Actions_Data");
        dataGridView1.DataSource = ds.Tables["Actions_Data"];
        con.Close();
    }

}
}
}

```