

2010

Προγραμματισμός Λογικού Ελεγκτή (PLC) Με Εφαρμογή Στον Αυτοματισμό Βιομηχανικών Πλυντηρίων

Υπεύθυνοι Καθηγητές : Διαμαντάρας Κων/νος
Υφούλης Χρήστος

ΒΑΡΣΑΜΗ ΚΥΡΙΑΚΗ



Α.Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ Α.Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ



ΠΡΟΛΟΓΟΣ

Έχουμε φτάσει σε αυτό το σημείο στην ιστορία που πλέον η οικολογία αποτελεί επιτακτική ανάγκη και είναι υποχρέωσή μας να την επιζητούμε και να την εφαρμόζουμε σε όλες τις εκφάνσεις της ζωής μας. Έχοντας αυτό ως κινητήριο δύναμη ξεκίνησα την πτυχιακή μου εργασία με σκοπό την δημιουργία ενός προγράμματος που θα εφαρμόζεται σε βιομηχανικά πλυντήρια και το οποίο θα βοηθάει τους χειριστές τους να εξοικονομούν χρόνο και χρήματα με ταυτόχρονα οικολογικά οφέλη.

Για να φέρω εις πέρας αυτό το έργο χρειάστηκε να εφαρμόσω όλες τις γνώσεις μου επάνω στον προγραμματισμό και τις μεθοδολογίες του, αλλά και να αποκτήσω νέες γνώσεις επάνω στα PLC και την γενικότερη φιλοσοφία που τα διέπει. Αμέριστη στήριξη στην προσπάθειά μου αυτή μου προσέφεραν οι επιβλέποντες καθηγητές μου κ. Κωνσταντίνος Διαμαντάρας και κ. Χρήστος Υφούλης οι οποίοι πίστεψαν σε εμένα και τόλμησαν να επενδύσουν στην ιδέα μου, γεγονός που με ωθεί να τους ευχαριστήσω θερμά. Πάνω απ' όλα όμως θα ήθελα να ευχαριστήσω την οικογένεια μου και όλους εκείνους που με υποστήριξαν όλα αυτά τα χρόνια και με βοήθησαν να κάνω αυτό το όνειρο πραγματικότητα.

Με εκτίμηση,

Βαρσάμη Κυριακή

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία αποτελεί μία προσπάθεια για την απλοποίηση της διαδικασίας προγραμματισμού ενός προγραμματιζόμενου λογικού ελεγκτή (PLC). Η διαδικασία αυτή για την καλύτερη κατανόησή της εφαρμόζεται σε βιομηχανικά πλυντήρια βήμα, βήμα, από τον σχεδιασμό, μέχρι την υλοποίηση του συστήματος.

Στην εισαγωγή δίνεται ο ορισμός ενός PLC, αλλά και κάποια εισαγωγικά ιστορικά στοιχεία, και γίνεται προσπάθεια για μία πρώτη είσοδο στον κόσμο των PLC, δίχως αναφορά σε ειδικότερα στοιχεία τους.

Τα πλεονεκτήματα και τα μειονεκτήματα των προγραμματιζόμενων λογικών ελεγκτών παρουσιάζονται στο πρώτο κεφάλαιο. Στο ίδιο κεφάλαιο γίνεται μία εκτεταμένη ανάλυση των δομικών στοιχείων των PLC και της ονοματολογίας τους, με σκοπό την σταδιακή εμβάθυνση στον κόσμο τους.

Προσπαθώντας μία εισαγωγή στον προγραμματισμό τους, αναλύονται στο τρίτο κεφάλαιο τα λογικά κυκλώματα, ώστε να γίνει κατανοητός ο τρόπος λειτουργίας των ελεγκτών. Απαραίτητη ήταν όμως και η αναφορά στις συσκευές που χρησιμοποιούνται για τον προγραμματισμό τους. Από το σημείο αυτό και έπειτα παρουσιάζονται οι αρχές, και οι γλώσσες προγραμματισμού, καθώς και η δομή ενός προγράμματος PLC.

Συνέχεια του προηγούμενου κεφαλαίου αποτελεί το τρίτο κεφάλαιο όπου αναπτύσσεται η λογικά ενοποιημένη διεργασία (RUP) και ο τρόπος με τον οποίο η χρήση της βοηθάει στην ανάπτυξη λογισμικού γενικά, αλλά και στην ανάπτυξη της εφαρμογής μας, ειδικά. Για την οπτικοποίηση της ανάπτυξης λογισμικού, χρησιμοποιείται η ενοποιημένη γλώσσα μοντελοποίησης, όπου εκτός της ανάλυσης της θεωρίας της, έγινε και ταυτόχρονη εφαρμογή της στο παράδειγμά μας.

Στο τελευταίο κεφάλαιο, χρησιμοποιώντας τις γνώσεις που αποκτήθηκαν από τα προηγούμενα, παρουσιάζεται και αναλύεται το πρόγραμμα που παράχθηκε για την εφαρμογή του προγραμματιζόμενου λογικού ελεγκτή στα βιομηχανικά πλυντήρια.

Μία επισήμανση των σημαντικότερων σημείων της εργασίας αυτής γίνεται στο τμήμα των συμπερασμάτων, ως επίλογος του έργου.

SUMMARY

This thesis constitutes an effort for the simplification of the programming process of a programmable logic controller (PLC) in general. For better comprehension this process is applied to industrial washing-machines step by step, from the planning, up to the concretization of the system.

In the introduction are given the definition of PLC, but also certain historical elements, and an effort is made for a first entry in the world of PLCs, without a report in their more specific characteristics.

The report in the advantages and in the disadvantages of programmable logic controllers exists in the first chapter. In the same chapter there is an extensive analysis of structural elements of PLC and their nomenclature, aiming at the progressive deepening in their world.

Trying an introduction in their planning, in the third chapter the logic circuits are being analyzed, so that the way of operation of controllers becomes comprehensible. Essential was however the report in the appliances that are used for their programming. Beyond this point, the principals, the programming languages, as well as the structure of a PLC program are being presented.

The third chapter constitutes a continuity of the previous chapter, in which rational unified process (RUP) is being developed, and the way in which its use helps in software development in general, but also in the software that will be used in our application, specifically. For the visualization of software development, the unified modeling language is used, where apart from the analysis of the theory, a simultaneous application in our software was done.

In the last chapter, using the knowledge that was acquired by the precedent ones, we present and analyze the program that was produced for the application of logic controller in the industrial washing-machines.

Pointing out of more important points of this work becomes in the conclusions section the epilogue of this work.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΙΣΑΓΩΓΗ.....	9
ΚΕΦΑΛΑΙΟ 1 : ΓΕΝΙΚΑ ΣΤΟΙΧΕΙΑ ΓΙΑ ΤΟΥΣ ΕΛΕΓΚΤΕΣ ΠΡΟΓΡΑΜΜΑΤΙΖΟΜΕΝΗΣ ΛΟΓΙΚΗΣ.....	11
1.1 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΩΝ PLC.....	11
1.2 ΔΟΜΙΚΑ ΣΤΟΙΧΕΙΑ ΤΩΝ PLC.....	13
1.2.1 ΠΛΑΙΣΙΑ ΤΟΠΟΘΕΤΗΣΗΣ ΜΟΝΑΔΩΝ	14
1.2.2 ΜΟΝΑΔΑ ΤΡΟΦΟΔΟΣΙΑΣ.....	16
1.2.3 ΚΕΝΤΡΙΚΗ ΜΟΝΑΔΑ ΕΠΕΞΕΡΓΑΣΙΑΣ (CPU).....	17
1.2.4 ΜΟΝΑΔΕΣ ΕΙΣΟΔΩΝ/ ΕΞΟΔΩΝ.....	21
1.3 ΟΝΟΜΑΤΟΛΟΓΙΑ ΤΩΝ PLC.....	23
1.3.1 ΟΝΟΜΑΤΟΛΟΓΙΑ ΧΡΟΝΙΚΩΝ.....	23
1.3.2 ΟΝΟΜΑΤΟΛΟΓΙΑ ΑΠΑΡΙΘΜΗΤΩΝ	24
1.3.3 ΟΝΟΜΑΤΟΛΟΓΙΑ ΤΥΜΠΑΝΟΕΙΔΩΝ ΕΛΕΓΚΤΩΝ	24
ΚΕΦΑΛΑΙΟ 2 : ΓΕΝΙΚΑ ΣΤΟΙΧΕΙΑ ΓΙΑ ΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΤΩΝ PLC...25	
2.1 ΛΟΓΙΚΑ ΚΥΚΛΩΜΑΤΑ.....	25
2.1.1 ΑΣΥΓΧΡΟΝΑ FLIP-FLOP.....	29
2.1.2 ΣΥΓΧΡΟΝΑ FLIP-FLOP	31
2.2 ΣΥΣΚΕΥΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΤΩΝ PLC	34
2.3 ΑΡΧΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΤΩΝ PLC.....	35
2.4 ΔΟΜΗ ΠΡΟΓΡΑΜΜΑΤΟΣ	37
2.5 ΜΟΡΦΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	39
2.6 ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΤΩΝ PLC.....	41
2.6.1 ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ LADDER (LD)	42
2.6.2 ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ FUNCTION BLOCK DIAGRAM (FBD)	45
2.6.3 ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ SEQUENTIAL FUNCTION CHART (SFC)	46
2.6.4 ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ STRUCTURED TEXT (ST)	50
2.6.5 ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ INSTRUCTION LIST (IL)	51

ΚΕΦΑΛΑΙΟ 3 : ΑΝΑΠΤΥΞΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ	52
3.1 ΛΟΓΙΚΑ ΕΝΟΠΟΙΗΜΕΝΗ ΔΙΕΡΓΑΣΙΑ RUP	52
3.1.1 ΚΥΚΛΟΣ ΖΩΗΣ ΛΟΓΙΚΑ ΕΝΟΠΟΙΗΜΕΝΗΣ ΔΙΕΡΓΑΣΙΑΣ.....	54
3.1.2 ΕΡΓΑΣΙΕΣ ΦΑΣΕΩΝ ΛΟΓΙΚΑ ΕΝΟΠΟΙΗΜΕΝΗΣ ΔΙΕΡΓΑΣΙΑΣ	58
3.2 ΕΝΟΠΟΙΗΜΕΝΗ ΓΛΩΣΣΑ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ UML	61
3.2.1 ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΗΣ ΧΡΗΣΗΣ	63
3.2.2 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΗΣ.....	65
3.2.3 ΔΙΑΓΡΑΜΜΑ ΑΚΟΛΟΥΘΙΑΣ	69
3.2.4 ΔΙΑΓΡΑΜΜΑ ΣΥΝΕΡΓΑΣΙΑΣ.....	71
3.2.5 ΔΙΑΓΡΑΜΜΑ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ	73
3.2.6 ΔΙΑΓΡΑΜΜΑ ΚΑΤΑΣΤΑΣΗΣ.....	75
3.3 ΕΦΑΡΜΟΓΗ ΤΗΣ RUP ΚΑΙ ΤΗΣ UML ΣΤΗΝ ΑΝΑΠΤΥΞΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΒΙΟΜΗΧΑΝΙΚΑ ΠΛΥΝΤΗΡΙΑ.....	76
3.3.1 ΣΥΛΛΗΨΗ.....	77
3.3.2 ΕΠΕΞΕΡΓΑΣΙΑ	80
3.3.3 ΚΑΤΑΣΚΕΥΗ	81
3.3.4 ΜΕΤΑΒΑΣΗ.....	82
ΚΕΦΑΛΑΙΟ 4 : ΚΑΤΑΣΚΕΥΗ ΛΟΓΙΣΜΙΚΟΥ	83
4.1 ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ TWIDOSUITE	83
4.1.1 ΛΕΙΤΟΥΡΓΙΕΣ ΤΟΥ TWIDOSUITE	85
4.2 ΠΡΟΓΡΑΜΜΑ ΓΙΑ ΤΗΝ ΛΕΙΤΟΥΡΓΙΑ ΒΙΟΜΗΧΑΝΙΚΟΥ ΠΛΥΝΤΗΡΙΟΥ.....	87
4.2.1 ΓΕΝΙΚΕΣ ΠΛΗΡΟΦΟΡΙΕΣ ΠΡΟΓΡΑΜΜΑΤΟΣ	88
4.2.2 ΛΕΠΤΟΜΕΡΕΙΕΣ ΠΡΟΓΡΑΜΜΑΤΟΣ	89
4.2.3 ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ.....	96
4.2.4 ΠΑΡΑΔΟΧΕΣ ΠΡΟΓΡΑΜΜΑΤΟΣ	100
ΠΑΡΑΡΤΗΜΑ Α	102
ΠΑΡΑΡΤΗΜΑ Β	121
ΣΥΜΠΕΡΑΣΜΑΤΑ	130
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	132
ΙΣΤΟΤΟΠΟΙ	132

ΠΙΝΑΚΑΣ ΕΥΡΕΣΗΣ ΕΙΚΟΝΩΝ

ΕΙΚΟΝΑ I: ΓΕΝΙΚΗ ΑΠΟΨΗ PLC	10
ΕΙΚΟΝΑ II: ΓΕΝΙΚΗ ΑΠΟΨΗ PLC	12
ΕΙΚΟΝΑ III: ΓΕΝΙΚΗ ΑΠΟΨΗ PLC	13
ΕΙΚΟΝΑ IV: ΓΕΝΙΚΗ ΑΠΟΨΗ ΠΛΑΙΣΙΟΥ PLC	15
ΕΙΚΟΝΑ V: ΓΕΝΙΚΗ ΑΠΟΨΗ ΔΟΜΗΣ PLC.....	18
ΕΙΚΟΝΑ VI: ΣΥΣΚΕΥΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ PLC	34
ΕΙΚΟΝΑ VII: ΠΑΡΑΔΕΙΓΜΑ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΕ ΓΛΩΣΣΑ ST	50
ΕΙΚΟΝΑ IX: ΣΗΜΑ TwidoSuite	83
ΕΙΚΟΝΑ X: ΕΙΣΑΓΩΓΙΚΗ ΟΘΟΝΗ TwidoSuite	84
ΕΙΚΟΝΑ XI: ΟΘΟΝΗ ΕΞΕΡΕΥΝΗΣΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ TwidoSuite	85
ΕΙΚΟΝΑ XII: ΟΘΟΝΗ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΕ TwidoSuite	86
ΕΙΚΟΝΑ XIII: ΟΘΟΝΗ ΑΠΟΣΦΑΛΜΑΤΩΣΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ TwidoSuite ..	86
ΕΙΚΟΝΑ XIV: ΑΠΟΨΗ ΒΙΟΜΗΧΑΝΙΚΟΥ ΠΛΥΝΤΗΡΙΟΥ (ΠΑΡΟΥΣΙΑΣΗΣ)	87

ΠΙΝΑΚΑΣ ΕΥΡΕΣΗΣ ΠΙΝΑΚΩΝ

ΠΙΝΑΚΑΣ I: ΠΙΝΑΚΕΣ ΑΛΗΘΕΙΑΣ ΠΥΛΩΝ AND, OR, NOT	26
ΠΙΝΑΚΑΣ II: ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ ΠΥΛΩΝ XOR, NAND, NOR, XNOR	26
ΠΙΝΑΚΑΣ III: ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ ΑΣΥΓΧΡΟΝΟΥ FLIP-FLOP	30
ΠΙΝΑΚΑΣ IV: ΠΙΝΑΚΕΣ ΑΛΗΘΕΙΑΣ ΜΑΝΤΑΛΩΤΗ	33
ΠΙΝΑΚΑΣ V: ΠΙΝΑΚΑΣ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ	77
ΠΙΝΑΚΑΣ VI: ΠΙΝΑΚΑΣ ΧΡΗΣΤΩΝ ΚΑΙ ΣΤΟΧΩΝ ΣΥΣΤΗΜΑΤΟΣ.....	77
ΠΙΝΑΚΑΣ VII: ΠΙΝΑΚΑΣ ΕΙΣΟΔΩΝ ΠΡΟΓΡΑΜΜΑΤΟΣ	88
ΠΙΝΑΚΑΣ VIII: ΠΙΝΑΚΑΣ ΕΞΟΔΩΝ ΠΡΟΓΡΑΜΜΑΤΟΣ	88
ΠΙΝΑΚΑΣ IX: ΒΗΜΑΤΑ ΚΑΙ ΒΙΤ ΕΛΕΓΧΟΥ ΤΥΜΠΑΝΟΕΙΔΟΥΣ ΕΛΕΓΚΤΗ	91
ΠΙΝΑΚΑΣ X: ΧΡΟΝΙΚΑ ΠΡΟΓΡΑΜΜΑΤΟΣ	92
ΠΙΝΑΚΑΣ XI: ΑΠΑΡΙΘΜΗΤΕΣ ΠΡΟΓΡΑΜΜΑΤΟΣ	94
ΠΙΝΑΚΑΣ XII: ΜΝΗΜΕΣ ΠΡΟΓΡΑΜΜΑΤΟΣ	95
ΠΙΝΑΚΑΣ XIII: ΠΙΝΑΚΑΣ ΕΝΕΡΓΩΝ ΚΥΚΛΩΜΑΤΩΝ	96
ΠΙΝΑΚΑΣ XIV: ΠΙΝΑΚΑΣ ΔΕΔΟΜΕΝΩΝ	97

ΠΙΝΑΚΑΣ ΕΥΡΕΣΗΣ ΣΧΗΜΑΤΩΝ

ΣΧΗΜΑ I: ΛΕΙΤΟΥΡΓΙΑ CPU	20
ΣΧΗΜΑ II: ΔΟΜΗ PLC.....	22
ΣΧΗΜΑ III: ΠΥΛΕΣ AND, OR, NOT	25
ΣΧΗΜΑ IV: ΠΥΛΕΣ XOR, NAND, NOR, XNOR.....	26
ΣΧΗΜΑ V: ΣΥΝΔΙΑΣΤΙΚΟ ΚΥΚΛΩΜΑ.....	27
ΣΧΗΜΑ VI: ΑΚΟΛΟΥΘΙΑΚΟ ΚΥΚΛΩΜΑ.....	28
ΣΧΗΜΑ VII: ΑΣΥΓΧΡΟΝΟ FLIP-FLOP	29
ΣΧΗΜΑ VIII: ΩΡΟΛΟΓΙΑΚΟΣ ΠΑΛΜΟΣ ΜΕ ΜΕΤΩΠΟ ΘΕΤΙΚΗΣ ΔΙΕΓΕΡΣΗΣ .31	
ΣΧΗΜΑ IX: ΩΡΟΛΟΓΙΑΚΟΣ ΠΑΛΜΟΣ ΜΕ ΜΕΤΩΠΟ ΑΡΝΗΤΙΚΗΣ ΔΙΕΓΕΡΣΗΣ	31
ΣΧΗΜΑ X: ΔΙΑΚΡΙΣΗ FLIP-FLOP ΑΡΝΗΤΙΚΗΣ ΚΑΙ ΘΕΤΙΚΗΣ ΔΙΕΓΕΡΣΗΣ.....	32
ΣΧΗΜΑ XI: ΜΑΝΤΑΛΩΤΗΣ	32
ΣΧΗΜΑ XII: ΤΥΠΙΚΗ ΑΚΟΛΟΥΘΙΑ ΣΑΡΩΣΗΣ	36
ΣΧΗΜΑ XIII: ΚΥΚΛΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ	37
ΣΧΗΜΑ XIV: ΑΠΕΙΚΟΝΙΣΗ ΛΟΓΙΚΗΣ ΠΡΑΞΗΣ ΣΤΙΣ ΤΥΠΟΠΟΙΗΜΕΝΕΣ ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	42
ΣΧΗΜΑ XV: ΗΛΕΚΤΡΙΚΟ ΚΥΚΛΩΜΑ ΜΕ ΛΟΓΙΚΗ ΣΥΡΜΑΤΩΣΗΣ	43
ΣΧΗΜΑ XVI: ΗΛΕΚΤΡΙΚΟ ΚΥΚΛΩΜΑ ΜΕ ΠΡΟΓΡΑΜΜΑΤΙΖΟΜΕΝΗ ΛΟΓΙΚΗ.43	
ΣΧΗΜΑ XVII: ΠΑΡΑΔΕΙΓΜΑ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΕ ΓΛΩΣΣΑ FBD	46
ΣΧΗΜΑ XVIII: ΣΥΜΒΟΛΙΣΜΟΙ ΣΤΟΙΧΕΙΩΝ ΓΛΩΣΣΑΣ SFC	47-48
ΣΧΗΜΑ XIX: ΠΑΡΑΔΕΙΓΜΑ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΕ ΓΛΩΣΣΑ SFC	49
ΣΧΗΜΑ XX: ΕΜΠΛΕΚΟΝΤΑΣ ΤΗΝ SFC ΣΤΗΝ LADDER (ΣΤΟ ΙΔΙΟ ΠΑΡΑΔΕΙΓΜΑ).....	49
ΣΧΗΜΑ XXI: ΔΙΑΔΙΚΑΣΙΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ.....	52
ΣΧΗΜΑ XXII: ΦΑΣΕΙΣ ΚΑΙ ΕΡΓΑΣΙΕΣ ΛΟΓΙΚΑ ΕΝΟΠΟΙΗΜΕΝΗΣ ΔΙΕΡΓΑΣΙΑΣ	53
ΣΧΗΜΑ XXIII: ΠΟΣΟΣΤΙΑΙΟΣ ΚΑΤΑΜΕΡΙΣΜΟΣ ΕΡΓΑΣΙΩΝ ΚΑΤΑ ΤΗ ΔΙΑΡΚΕΙΑ ΕΝΟΣ ΤΥΠΙΚΟΥ ΕΡΓΟΥ	54
ΣΧΗΜΑ XXIV: ΔΙΕΡΓΑΣΙΑ ΠΡΟΣΔΙΟΡΙΣΜΟΥ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ.....	58
ΣΧΗΜΑ XXV: ΔΙΑΓΡΑΜΜΑΤΑ UML	62
ΣΧΗΜΑ XXVI: ΣΥΜΒΟΛΑ ΔΙΑΓΡΑΜΜΑΤΟΣ ΠΕΡΙΠΤΩΣΗΣ ΧΡΗΣΗΣ.....	64
ΣΧΗΜΑ XXVII: ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΓΡΑΜΜΑΤΟΣ ΠΕΡΙΠΤΩΣΗΣ ΧΡΗΣΗΣ	64

ΣΧΗΜΑ XXVIII: ΣΥΜΒΟΛΙΣΜΟΣ ΚΛΑΣΗΣ.....	65
ΣΧΗΜΑ XXIX: ΠΑΡΑΔΕΙΓΜΑ ΚΛΑΣΗΣ	66
ΣΧΗΜΑ XXX: ΣΥΜΒΟΛΙΣΜΟΙ ΣΧΕΣΩΝ ΜΕΤΑΞΥ ΚΛΑΣΕΩΝ	68
ΣΧΗΜΑ XXXI: ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΓΡΑΜΜΑΤΟΣ ΚΛΑΣΕΩΝ	68
ΣΧΗΜΑ XXXII: ΣΤΟΙΧΕΙΑ ΔΙΑΓΡΑΜΜΑΤΩΝ ΑΚΟΛΟΥΘΙΑΣ	70
ΣΧΗΜΑ XXXIII: ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΓΡΑΜΜΑΤΟΣ ΑΚΟΛΟΥΘΙΑΣ	71
ΣΧΗΜΑ XXXIV: ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΓΡΑΜΜΑΤΟΣ ΣΥΝΕΡΓΑΣΙΑΣ	72
ΣΧΗΜΑ XXXV: ΣΥΜΒΟΛΙΣΜΟΙ ΣΤΟΙΧΕΙΩΝ ΔΙΑΓΡΑΜΜΑΤΟΣ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ	74
ΣΧΗΜΑ XXXVI: ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΓΡΑΜΜΑΤΟΣ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ	74
ΣΧΗΜΑ XXXVII ΣΥΜΒΟΛΙΣΜΟΣ ΚΑΤΑΣΤΑΣΗΣ ΔΙΑΓΡΑΜΜΑΤΟΣ ΚΑΤΑΣΤΑΣΗΣ.....	75
ΣΧΗΜΑ XXXVIII: ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΓΡΑΜΜΑΤΟΣ ΚΑΤΑΣΤΑΣΗΣ.....	76
ΣΧΗΜΑ XXXIX: ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ ΒΙΟΜΗΧΑΝΙΚΟΥ ΠΛΥΝΤΗΡΙΟΥ.....	78
ΣΧΗΜΑ XL: ΔΙΑΓΡΑΜΜΑ ΑΚΟΛΟΥΘΙΑΣ ΧΡΗΣΗΣ ΒΙΟΜΗΧΑΝΙΚΟΥ ΠΛΥΝΤΗΡΙΟΥ.....	79
ΣΧΗΜΑ XLI: ΔΙΑΓΡΑΜΜΑ ΚΑΤΑΣΤΑΣΗΣ ΒΙΟΜΗΧΑΝΙΚΟΥ ΠΛΥΝΤΗΡΙΟΥ	80
ΣΧΗΜΑ XLII: ΔΙΑΓΡΑΜΜΑ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ ΒΙΟΜΗΧΑΝΙΚΟΥ ΠΛΥΝΤΗΡΙΟΥ.....	81
ΣΧΗΜΑ XLIII: ΣΥΜΒΟΛΙΣΜΟΣ ΤΥΜΠΑΝΟΕΙΔΟΥΣ ΕΛΕΓΚΤΗ	90
ΣΧΗΜΑ XLIV: ΣΥΜΒΟΛΙΣΜΟΣ ΧΡΟΝΙΚΟΥ.....	92
ΣΧΗΜΑ XLV: ΣΥΜΒΟΛΙΣΜΟΣ ΑΠΑΡΙΘΜΗΤΗ	93

ΕΙΣΑΓΩΓΗ

Μέχρι πριν από τρεις δεκαετίες περίπου, η επιστήμη του αυτοματισμού βασιζόταν κυρίως στα συστήματα καλωδιωμένης λογικής. Όλη η λειτουργία του αυτοματισμού δηλαδή, στηριζόταν στην καλωδίωση ανάμεσα στα διάφορα στοιχεία όπως είναι οι επαφές, τα πηνία, τα ρελέ, τα χρονικά και οι πύλες.

Έτσι ένας πίνακας αυτοματισμού μπορούσε να κατασκευασθεί, μόνο όταν είχε ολοκληρωθεί το συνδεσμολογικό σχέδιο. Αυτό είχε ως συνέπεια κάθε μετέπειτα αλλαγή να σημαίνει και μια αντίστοιχη αλλαγή στη καλωδίωση (τροποποίηση του πίνακα), γεγονός που αυτόματα οδηγούσε στην αντιμετώπιση προβλημάτων επάρκειας ελεύθερων επαφών των ρελέ και χώρου για την προσθήκη περεταίρω υλικού.

Πλέον, έχουμε φτάσει στο σημείο όπου τα συστήματα προγραμματιζόμενης λογικής, με κύριους εκπροσώπους στη βιομηχανία τους ελεγκτές προγραμματιζόμενης λογικής, έχουν πια εκτοπίσει τελείως τα προηγούμενα συστήματα αυτοματισμού.

Οι ελεγκτές προγραμματιζόμενης λογικής εμφανίστηκαν για πρώτη φορά στην αγορά στα τέλη του 1960, με κύριο σκοπό εκείνη την εποχή την κάλυψη των αναγκών αυτοματοποίησης της αμερικανικής βιομηχανίας αυτοκινήτων. Η εμφάνιση αυτή μπορεί να θεωρηθεί αποτέλεσμα της εξέλιξης των παραγωγικών διαδικασιών οι οποίες απαιτούσαν όλο και πιο πολύπλοκη λογική στα συστήματα ελέγχου, αλλά και της προόδου της ηλεκτρονικής στην βιομηχανία που η προμήθεια των κατασκευαστών με το αναγκαίο υλικό ήταν πλέον εφικτή. Στις αρχές της δεκαετίας του 1980 παρουσιάστηκε από τις εταιρείες στους τεχνικούς της βιομηχανίας ένα νέο προϊόν με το όνομα PLC χωρίς να χρησιμοποιείται η πλήρης ονομασία του (Programmable Logic Controller) για να αποφευχθεί η οποιαδήποτε αναστάτωση στο τεχνικό κατεστημένο της βιομηχανίας.

Οι αρχικές εκδόσεις των PLC είχαν την δυνατότητα επεξεργασίας μόνο ψηφιακών σημάτων και όπως μπορεί κανείς να φανταστεί ο προγραμματισμός τους δεν ήταν φιλικός προς τον χρήστη. Κατά την διάρκεια του 1970 τα PLC άρχιζαν να εγκαθίστανται δειλά, δειλά στην βιομηχανία, ενώ η απόλυτη καθιέρωσή τους ήρθε την δεκαετία του 1980. Η εμφάνιση του προσωπικού υπολογιστή το 1982, προσέφερε απεριόριστες δυνατότητες και ευκολίες στην δημιουργία και τον έλεγχο προγραμμάτων, αν και συγχρόνως και μέχρι το τέλος της δεκαετίας απείλησε την ίδια την ύπαρξη των PLC. Η επικράτηση τους όμως στην παραγωγική διαδικασία ήταν εύκολη και αδιαμφισβήτητη. Έτσι από τις αρχές του 1990 έως σήμερα η ανάπτυξη των προγραμματιζόμενων λογικών ελεγκτών είναι αλματώδης και χαρακτηρίζεται από δίκτυα υψηλών ταχυτήτων και πανίσχυρους κεντρικούς επεξεργαστές.

Από την εποχή της δημιουργίας τους μέχρι τώρα, τα PLC έχουν αναπτυχθεί τόσο πολύ, ώστε πλέον αποτελούν αναπόσπαστο κομμάτι κάθε μορφής βιομηχανίας και χρησιμοποιούνται στον ευρύτερο και πολυσύνθετο χώρο της.

Στα συστήματα προγραμματιζόμενης λογικής, η κατασκευή και καλωδίωση του πίνακα είναι ανεξάρτητη από τη λειτουργία που πρόκειται να εκτελέσει ο αυτοματισμός. Πάνω στον ελεγκτή συνδέονται όλα τα στοιχεία που δίνουν εντολές όπως είναι οι τερματικοί διακόπτες και τα κουμπιά, καθώς και τα στοιχεία που δέχονται εντολές όπως είναι τα πηνία, τα ρελέ ισχύος κινητήρων, οι βαλβίδες και οι λυχνίες.

Η διαδικασία του αυτοματισμού προγραμματίζεται στην μνήμη του ελεγκτή, ακόμα και την τελευταία στιγμή, πριν να τεθεί αυτός σε λειτουργία. Επομένως η μελέτη, το πρόγραμμα δηλαδή, είναι δυνατόν να γίνεται παράλληλα με την επιλογή του υλικού και την κατασκευή του πίνακα.

Μια περεταίρω αλλαγή της λειτουργίας με οποιοδήποτε τρόπο και για οποιοδήποτε λόγο - γεγονός σύνηθες στον προγραμματισμό - γίνεται με μία απλή διόρθωση του προγράμματος, χωρίς ουδεμία επέμβαση στην καλωδίωση του πίνακα.

Λαμβάνοντας τα παραπάνω υπ' όψιν, μπορούμε να ορίσουμε τους προγραμματιζόμενους λογικούς ελεγκτές PLC (Programmable Logic Controllers) ως ψηφιακές ηλεκτρονικές συσκευές οι οποίες χρησιμοποιούν μια προγραμματιζόμενη μνήμη για την αποθήκευση οδηγιών και ειδικές λειτουργίες όπως είναι η λογική, η ακολουθία, ο χρόνος και η αρίθμηση, για να ελέγξουν τις μηχανές και την διαδικασία (Εριφάκη Μαρία, Καραμούτα Πηνελόπη, 2005).

Τα PLC είναι οι ηλεκτρονικοί υπολογιστές της βιομηχανίας.



ΕΙΚΟΝΑ Ι: ΓΕΝΙΚΗ ΑΠΟΨΗ PLC

ΚΕΦΑΛΑΙΟ 1

ΓΕΝΙΚΑ ΣΤΟΙΧΕΙΑ ΓΙΑ ΤΟΥΣ ΕΛΕΓΚΤΕΣ ΠΡΟΓΡΑΜΜΑΤΙΖΟΜΕΝΗΣ ΛΟΓΙΚΗΣ

1.1 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΩΝ PLC

Προηγουμένως έγινε αναφορά στα σημαντικότερα πλεονεκτήματα των ελεγκτών προγραμματιζόμενης λογικής – θα αναφέρονται από εδώ και έπειτα απλά ως PLC - στην επιστήμη του αυτοματισμού, τα οποία είναι η ευελιξία στις μετατροπές και οι πολλές τους δυνατότητες. Αυτά όμως δεν είναι τα μόνα πλεονεκτήματα των PLC και είναι ουσιώδης έστω η αναφορά των πιο σπουδαίων από αυτά. Έτσι γνωρίζουμε πως όσον αφορά τα PLC :

- Η ταχύτητα της διαδικασίας παραγωγής αγγίζει το μέγιστο, και κατά συνέπεια μειώνεται στο ελάχιστο ο χρόνος απόσβεσης της εγκατάστασης.
- Ο χρόνος κατασκευής του αυτοματισμού είναι μηδαμινός σε σχέση με την κατασκευή ενός κλασικού πίνακα αυτοματισμού. Έτσι η παράδοση του αυτοματισμού σε λειτουργία είναι γρηγορότερη και ειδικά από τη στιγμή που η μελέτη μπορεί να πραγματοποιείται παράλληλα με την τοποθέτηση και καλωδίωση του ελεγκτή.
- Η εγκατάστασή τους είναι ουσιαστικά ακίνδυνη ακόμη και μέσα σε πεδία ισχύος.
- Στο στάδιο της μελέτης δεν απασχολεί η επάρκεια του αριθμού των επαφών των ρελέ, των χρονικών ή των εξωτερικών τερματικών, καθώς τα PLC έχουν την δυνατότητα επέκτασης. Έτσι αν είναι απαραίτητη η προσθήκη κάποιας καινούργιας διαδικασίας ή ενός επιπλέον ελέγχου, είναι δυνατόν το ίδιο PLC να την πραγματοποιήσει απλά με την προσθήκη των κατάλληλων καρτών εισόδου / εξόδου (I/O).
- Ο αυτοματισμός μπορεί να αλλάξει πολύ εύκολα είτε βρίσκεται στο στάδιο της μελέτης και της κατασκευής, είτε ακόμη και αφού έχει τεθεί σε λειτουργία.
- Εφόσον για κάθε εξωτερική εντολή υπάρχει αντίστοιχο LED ο εντοπισμός των λαθών είναι πολύ ευκολότερος. Επίσης, με την χρήση μίας συσκευής προγραμματισμού παρέχεται η δυνατότητα ελέγχου της ροής του αυτοματισμού.
- Είναι δυνατή αλλά και επιθυμητή η σύνδεση του με περιφερειακές μονάδες για τον έλεγχο του υλικού, καταργώντας το χρονοβόρο και παρωχημένο μιμητικό διάγραμμα και τον πίνακα χειρισμών. Επίσης, μπορεί να συνδεθεί με ηλεκτρονικό υπολογιστή για ανταλλαγή στοιχείων.
- Όλες οι κάρτες είναι τοποθετημένες στον ίδιο δίαυλο επικοινωνίας (bus), γεγονός που επιταχύνει την διαδικασία ανταλλαγής πληροφοριών.

- Ξεπερνάν το πρόβλημα των «μη ενημερωμένων» σχεδίων του πίνακα αυτοματισμού, αφού ο ελεγκτής έχει πάντα αποθηκευμένο το τελευταίο πρόγραμμα, το οποίο είναι εφικτό είτε να διαβαστεί με μία συσκευή προγραμματισμού είτε ακόμη και να εκτυπωθεί σε χαρτί.
- Εξοικονομούν χώρο, προσπάθεια όσον αφορά τη συντήρηση (από τη στιγμή που δεν υπάρχουν μηχανικές επαφές) και ενέργεια.
- Ο προγραμματισμός τους γίνεται σε γλώσσα προσαρμοσμένη στο βιομηχανικό αυτοματισμό και άρα προσιτή στο προσωπικό που παλαιότερα συντηρούσε τους κλασσικούς πίνακες αυτοματισμού.
- Η διάρκεια ζωής τους είναι πρακτικά απεριόριστη λόγω των ασθενών τάσεων που χρησιμοποιούν. Ταυτόχρονα συμβάλλουν έτσι στην εξοικονόμηση ηλεκτρικής ενέργειας από την λειτουργία της εγκατάστασης.
- Τα PLC είναι συμβατά με υψηλά ηλεκτρομαγνητικά πεδία, έχουν εξαιρετική αντοχή σε κραδασμούς, και λειτουργούν σε ακραίες θερμοκρασίες 0°C έως 60°C χωρίς την ανάγκη προσθήκης επιπλέον ανεμιστήρων.

Ίσως εδώ να είναι η καταλληλότερη στιγμή να αναφερθεί πως μέχρι σήμερα δεν έχει διατυπωθεί κανένα μειονέκτημα όσον αφορά τη λειτουργία των PLC, αλλά και γενικότερα την ύπαρξή τους.



ΕΙΚΟΝΑ II: ΓΕΝΙΚΗ ΑΠΟΨΗ PLC

1.2 ΔΟΜΙΚΑ ΣΤΟΙΧΕΙΑ ΤΩΝ PLC

Για την καλύτερη κατανόηση του PLC και των λειτουργιών του είναι απαραίτητη η ανάλυση της δομής του, των στοιχείων δηλαδή, που το αποτελούν, ειδικά εφ' όσον η οποιαδήποτε επιλογή ενός PLC εξαρτάται κυρίως από το πλήθος των εισόδων, των εξόδων, αλλά και των λειτουργιών που είναι απαραίτητο να πραγματοποιήσει ο αυτοματισμός και αφορούν κατά κύριο λόγο την μνήμη όπως και τις δυνατότητες της κεντρικής μονάδας.

Ανεξάρτητα όμως από τον τύπο και το μέγεθος, κάθε PLC μπορεί να διαιρεθεί στα παρακάτω τέσσερα μέρη:

- Το πλαίσιο ή τα πλαίσια τα οποία χρειάζονται για την τοποθέτηση των μονάδων και των επεκτάσεών τους.
- Τη μονάδα τροφοδοσίας.
- Τη κεντρική μονάδα επεξεργασίας (Central Processing Unit, CPU), που μπορεί να παρομοιασθεί με τον εγκέφαλο του PLC.
- Τις μονάδες εισόδων/ εξόδων (Input/ Output modules).

Είναι ίσως σκόπιμο να αναφέρουμε εδώ πως εκτός από τα πλαίσια και την κεντρική μονάδα αυτοματισμού – η οποία αποτελείται από τη μονάδα τροφοδοσίας, την CPU και τις μονάδες εισόδων / εξόδων - απαραίτητη είναι η ύπαρξη μιας συσκευής προγραμματισμού για τον προγραμματισμό του PLC. Επίσης χρήσιμη θα ήταν μία αναφορά στις μνήμες RAM, EEPROM και ROM, που διαθέτουν όλα τα PLC, καθώς και στις ειδικές συναρτήσεις τους που περιλαμβάνουν τα χρονικά και τους απαριθμητές.



ΕΙΚΟΝΑ III: ΓΕΝΙΚΗ ΑΠΟΨΗ PLC

1.2.1 ΠΛΑΙΣΙΑ ΤΟΠΟΘΕΤΗΣΗΣ ΜΟΝΑΔΩΝ

Οι μονάδες που αποτελούν έναν ελεγκτή είναι φυσικό να τοποθετούνται σε κάποια πλαίσια. Επάνω στα πλαίσια αυτά ενσωματώνεται το σύστημα αγωγών, μέσω των οποίων επικοινωνούν οι διάφορες μονάδες μεταξύ τους για την απαραίτητη ανταλλαγή πληροφοριών, αλλά και για την τροφοδοσία τους.

ΤΥΠΟΙ ΠΛΑΙΣΙΩΝ

Τα πλαίσια τοποθέτησης μονάδων χωρίζονται συνήθως σε δύο μέρη, το κεντρικό πλαίσιο, το οποίο είναι μοναδιαίο σε κάθε PLC, και στο πλαίσιο επέκτασης, που έχει τη δυνατότητα επανάληψης του μέχρι την ικανοποίηση όλων των αναγκών.

Ως χαρακτηριστικό του κεντρικού πλαισίου μπορεί να θεωρηθεί η μοναδική του ικανότητα να φιλοξενεί την κεντρική μονάδα, και μάλιστα σε καθορισμένη θέση. Σε ορισμένους ελεγκτές, στο κεντρικό πλαίσιο τοποθετείται και η μονάδα τροφοδοσίας. Σε ενδεχόμενη ύπαρξη κενών θέσεων στο πλαίσιο αυτό, τοποθετούνται μονάδες εισόδων / εξόδων, ή άλλες ειδικές μονάδες.

Στα πλαίσια επέκτασης – χρησιμοποιείται πληθυστικός αριθμός γιατί όπως προαναφέρθηκε είναι δυνατόν να υπάρχουν περισσότερα του ενός - τοποθετούνται οι μονάδες εισόδων / εξόδων και χρησιμοποιούνται στην περίπτωση που δεν επαρκούν οι θέσεις του κεντρικού πλαισίου για την τοποθέτηση όλων των απαιτούμενων μονάδων.

Ένα πλαίσιο επέκτασης δημιουργείται τοποθετώντας τις μονάδες πλαισίου που απαιτούνται τη μία δίπλα στην άλλη, συνδέοντάς τες μεταξύ τους, αλλά και με το κεντρικό πλαίσιο μέσω μίας ειδικής μονάδας διασύνδεσης και καλωδίου. Κάθε μία από αυτές τις μονάδες πλαισίου διαθέτει συγκεκριμένες θέσεις για τις μονάδες εισόδων / εξόδων.

ΧΩΡΟΤΑΞΙΚΗ ΔΙΑΤΑΞΗ ΠΛΑΙΣΙΩΝ

Είναι εφικτή η διάκριση του συστήματος των πλαισίων επέκτασης ενός ελεγκτή ανάλογα με την απόσταση τους από το κεντρικό πλαίσιο σε κεντρικό και σε αποκεντρωμένο σύστημα.

Για να ονομαστεί ένα σύστημα κεντρικό, απαραίτητη προϋπόθεση είναι η απόσταση του πιο μακρινού πλαισίου επέκτασης από το κεντρικό πλαίσιο να μην ξεπερνά τα δύο με δυόμιση μέτρα.

Αντίθετα, σε ένα αποκεντρωμένο σύστημα, ορισμένα πλαίσια επέκτασης βρίσκονται σε μεγάλη απόσταση από το κεντρικό. Ένα τέτοιου είδους σύστημα

έχει νόημα όταν η εγκατάσταση που πρέπει να αυτοματιστεί έχει τα στοιχεία της, όπως είναι οι τερματικοί και τα ρελέ ισχύος, διαμοιρασμένα σε περισσότερα του ενός κέντρα βάρους. Αυτό έχει ως αποτέλεσμα την μεγάλη εξοικονόμηση στην καλωδίωση που χρησιμοποιείται από τα στοιχεία της εγκατάστασης για την σύνδεσή τους με τις μονάδες εισόδων / εξόδων, αφού αντιστοιχίζεται ένα ή και περισσότερα πλαίσια επέκτασης σε κάθε τέτοιο κέντρο βάρους.

Η διασύνδεση δύο μακρινών μεταξύ τους πλαισίων επιτυγχάνεται μέσω ειδικών μονάδων διασύνδεσης και ειδικού καλωδίου, ενώ είναι απαραίτητη η ύπαρξη μονάδας τροφοδοσίας στα πλαίσια αυτά. Επάνω σε κάθε κλάδο είναι δυνατόν να υπάρχουν μέχρι τέσσερα σημεία στα οποία θα τοποθετηθούν ένα ή και περισσότερα πλαίσια επέκτασης με τη μέθοδο του κεντρικού συστήματος.

Η μέθοδος αποκεντρωμένου συστήματος απαντάται σε δύο μορφές. Σε αυτήν της αλυσίδας που είναι η πιο δημοφιλής και η οποία περιγράφηκε εν ολίγοις παραπάνω, και στην ακτινική, κατά την οποία μπορούν να συνδεθούν ακτινικά με το κεντρικό πλαίσιο μέχρι τρεις αποκεντρωμένες θέσεις – κάθε μία σε απόσταση ενός χιλιομέτρου. Κάθε αποκεντρωμένη θέση μπορεί να δεχθεί περισσότερα από ένα πλαίσια με τη μέθοδο του κεντρικού συστήματος.

Είναι πολύ σημαντικό σε αυτό το σημείο να τονισθεί πως προγραμματιστικά δεν γίνεται καμία διάκριση ανάμεσα σε κοντινές και μακρινές μονάδες εισόδων / εξόδων. Με άλλα λόγια δεν διαφέρουν σε τίποτα τα προγράμματα που αφορούν διαφορετικές από άποψη χωροταξίας εγκαταστάσεις.



ΕΙΚΟΝΑ IV: ΓΕΝΙΚΗ ΑΠΟΨΗ ΠΛΑΙΣΙΟΥ PLC

1.2.2 ΜΟΝΑΔΑ ΤΡΟΦΟΔΟΣΙΑΣ

Η χρησιμότητα της μονάδας τροφοδοσίας έγκειται στην δημιουργία των απαραίτητων για τη λειτουργία της εγκατάστασης εσωτερικών τάσεων από το δίκτυο. Οι εσωτερικές αυτές τάσεις, τροφοδοτούν αποκλειστικά τα ηλεκτρικά στοιχεία που υπάρχουν μέσα στον ελεγκτή, όπως είναι τα τρανζίστορ. Η μονάδα τροφοδοσίας επιπλέον, είναι αυτή που βοηθάει στην διατήρηση του περιεχομένου της μνήμης RAM σε μία διακοπή ρεύματος, μαζί με μία ενσωματωμένη μπαταρία.

Μία μονάδα τροφοδοσίας αποτελείται από κάποια τεχνικά χαρακτηριστικά τα πιο σημαντικά εκ των οποίων παρατίθενται παρακάτω:

Είσοδος:

Ονομαστική τάση, συχνότητα, προστασία, ανοχές τάσης, απορροφούμενο ρεύμα.

Έξοδος:

Ονομαστική τάση, ονομαστικό ρεύμα, προστασία βραχυκυκλώματος.

Μπαταρία:

Διατήρηση μνήμης RAM.

Η μπαταρία ενσωματώνεται στην μονάδα τροφοδοσίας, και μαζί διατηρούν το περιεχόμενο της μνήμης σε μία διακοπή ρεύματος. Αυτό συμβαίνει στη περίπτωση που το πρόγραμμα ενός ελεγκτή είναι αποθηκευμένο στην μνήμη RAM. Η μπαταρία είναι συνήθως λιθίου, τοποθετείται εύκολα στην μονάδα τροφοδοσίας και μπορεί να κρατήσει το πρόγραμμα της μνήμης RAM για αρκετά μεγάλο χρονικό διάστημα.

Είναι απαραίτητη η επίδειξη μεγάλης προσοχής όμως στα εξής σημεία:

1. Η αντικατάσταση της μπαταρίας θα πρέπει να γίνεται με την συχνότητα που ορίζει ο κατασκευαστής, και πάντοτε με τον ελεγκτή υπό τάση ώστε να μην υπάρξει απώλεια του προγράμματος.
2. Το λογισμικό του ελεγκτή θα πρέπει να παρέχει στο χρήστη τη δυνατότητα να αντιληφθεί την πτώση της μπαταρίας κάτω του ορίου ασφαλείας. Έτσι καθίσταται δυνατή η έγκυρη αντικατάστασή της από τον χρήστη του, τηρώντας όλες τις προϋποθέσεις για την διατήρηση του προγράμματος και την ασφαλή λειτουργία του ελεγκτή.

1.2.3 ΚΕΝΤΡΙΚΗ ΜΟΝΑΔΑ ΕΠΕΞΕΡΓΑΣΙΑΣ (CPU)

Όπως προαναφέρθηκε η κεντρική μονάδα επεξεργασίας θεωρείται ο εγκέφαλος του συστήματος και αυτό γιατί εξ' ορισμού η κεντρική μονάδα επεξεργασίας εμπεριέχει όλα τα στοιχεία εκείνα που αποτελούν τη νοημοσύνη του συστήματος.

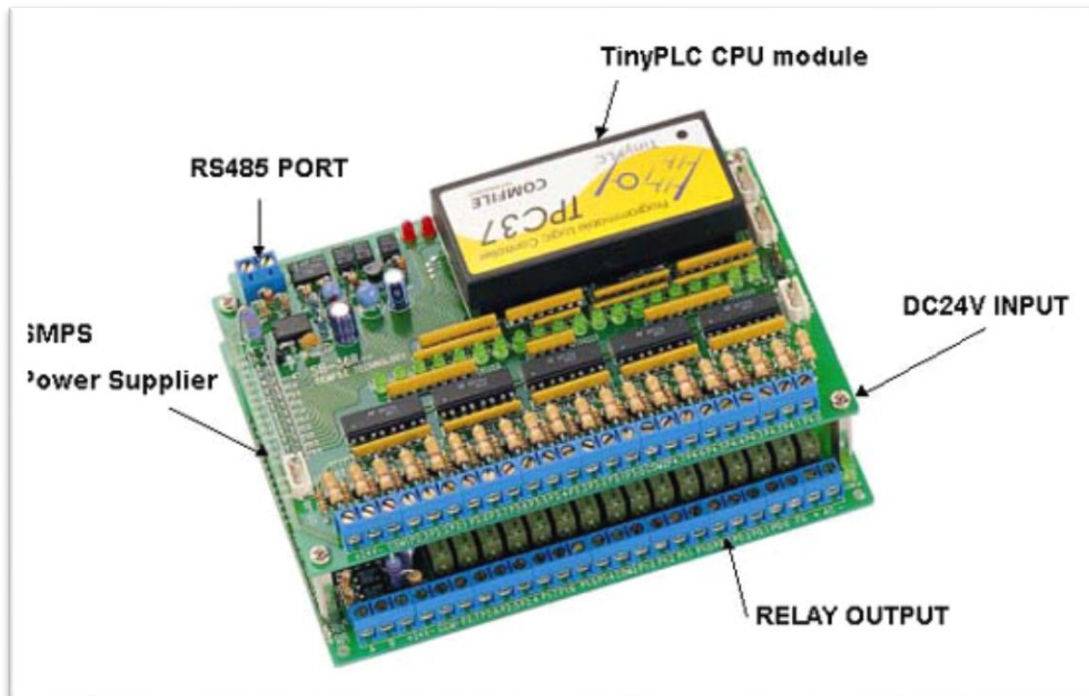
Συχνά ο όρος κεντρική μονάδα επεξεργασίας εναλλάσσεται καταχρηστικά με τον όρο επεξεργαστής, αν και ο επεξεργαστής αποτελεί ένα μέρος μόνο της CPU. Αν και έχουν ήδη αναφερθεί κάποια από τα κύρια συστατικά της CPU εκτεταμένα προηγουμένως, σκόπιμη θα ήταν μία επανάληψή τους εδώ, για την καλύτερη λογική σύνδεσή τους με την κεντρική μονάδα επεξεργασίας. Έτσι, υπάρχει η μονάδα τροφοδοσίας η οποία συνήθως εντοπίζεται μέσα στην CPU, αν και είναι δυνατόν να την απαντήσουμε και ως μία ξεχωριστή μονάδα προσαρτημένη δίπλα στην συσκευή της CPU, ενώ σημαντική είναι και η εφεδρική μπαταρία που υπάρχει μέσα στην μονάδα τροφοδοσίας - ανάλογα με τον τύπο της μνήμης διακρίνεται σε πτητική (volatile) και μη πτητική (non volatile) - και η οποία βοηθάει στην διατήρηση του προγράμματος σε περίπτωση διακοπής του ρεύματος.

Βασικό στοιχείο ενός PLC, και συγκεκριμένα μίας CPU είναι η μονάδα επεξεργαστή- μνήμης, η οποία περιέχει τον μικροεπεξεργαστή, τα ολοκληρωμένα μνήμης και τα κυκλώματα επικοινωνίας για την σύνδεση του επεξεργαστή με την συσκευή προγραμματισμού. Ο μικροεπεξεργαστής έχει τη μορφή ολοκληρωμένου κυκλώματος, και συνδυάζει τα πλεονεκτήματα του μικρού μεγέθους, του εύκολου προγραμματισμού, της υψηλής αξιοπιστίας και του χαμηλού κόστους. Επίσης μπορεί με τον κατάλληλο κάθε φορά προγραμματισμό να συμπεριφέρεται διαφορετικά και να εκτελεί μια ποικιλία λειτουργιών σύμφωνα με τις απαιτήσεις του προβλήματος που τίθεται προς αντιμετώπιση. Σε αυτή του την ιδιότητα οφείλει και το πρώτο συνθετικό της ονομασίας: «**Programmable**» το PLC. Μια CPU επίσης περιέχει τα κυκλώματα επικοινωνίας καθώς και τις μονάδες μνήμης του επεξεργαστή.

Όλα τα συστατικά στοιχεία μίας CPU επικοινωνούν μεταξύ τους με συγκεκριμένο τρόπο, και η επικοινωνία αυτή είναι συνεχής. Έτσι η μονάδα τροφοδοσίας είναι υπεύθυνη για την παροχή των απαραίτητων τάσεων για την σωστή λειτουργία του ελεγκτή, ενώ την ίδια στιγμή και για όλη την διάρκεια της λειτουργίας αυτής, ο επεξεργαστής επικοινωνεί με την μνήμη για την ανάκτηση, μετάφραση και εκτέλεση του προγράμματος που είναι αποθηκευμένο και υπεύθυνο για τον έλεγχο της διαδικασίας.

Στην κεντρική μονάδα επεξεργασίας, ο μικροεπεξεργαστής προσπελαύνει συνεχώς το πρόγραμμα που είναι γραμμένο στη μνήμη. Ελέγχει αν οι διάφορες είσοδοι έχουν ή όχι τάση, επεξεργάζεται τις εντολές του προγράμματος και βάσει

αυτών ορίζει αν, και ποιες από τις εξόδους θα αποκτήσουν τάση, έχοντας ως άμεση συνέπεια την διέγερση ή όχι των συνδεδεμένων σε αυτές στοιχείων.



ΕΙΚΟΝΑ V: ΓΕΝΙΚΗ ΑΠΟΨΗ ΔΟΜΗΣ PLC

ΜΝΗΜΗ:

Η μνήμη είναι το τμήμα του PLC στο οποίο γίνεται η αποθήκευση του προγράμματος, η πολυπλοκότητα του οποίου καθορίζει το μέγεθος αυτής που απαιτείται. Όπως συμβαίνει και στην περίπτωση των προσωπικών υπολογιστών οι μονάδες πληροφορίας που αποθηκεύονται στην μνήμη είναι τα γνωστά bits (δυναμικά ψηφία), στα οποία μεταφράζονται όλες οι εντολές του προγράμματος. Το μέγεθος της μνήμης του ελεγκτή καθορίζεται με τον ίδιο τρόπο, οργανώνοντας τις πληροφορίες αρχικά σε bytes, που αποτελούνται από οκτώ bits, και από εκεί και πάνω σε ομάδες των χιλίων bytes, τα λεγόμενα kilobytes.

Η διάκριση των ειδών μνήμης σε πτητικές και μη πτητικές είναι σημαντική, αφού οι πτητικές μνήμες είναι ευμετάβλητες, και είναι κατάλληλες για εφαρμογές υποστηριζόμενες από εφεδρική μονάδα τροφοδοσίας, ενώ οι μη πτητικές δεν αλλάζουν εύκολα και μπορούν να διατηρήσουν τις αποθηκευμένες πληροφορίες όταν η παροχή ισχύος διακοπεί.

Η σύμπτωση μεταξύ των προσωπικών υπολογιστών και των PLC συνεχίζεται και στα είδη μνήμης που χρησιμοποιούν κάνοντας απαραίτητη μια αναφορά στα πιο γνωστά από αυτά. Η αρχή φυσικά γίνεται με τις μνήμες τυχαίας

προσπέλασης ή διαφορετικά γνωστές ως μνήμες γραφής - ανάγνωσης (RAM ή R / W), στις οποίες είναι δυνατόν οι πληροφορίες να διαβαστούν, αλλά και να γραφούν. Οι μνήμες αυτού του τύπου είναι δυνατόν να απαντηθούν και στις δύο μορφές πτητικές και μη πτητικές, και αποτελούν την ευκολότερη λύση για την δημιουργία και την μετατροπή ενός προγράμματος.

Αντίθετα στις μη πτητικές, συμπεριλαμβάνονται οι μνήμες μόνο για ανάγνωση (ROM), που οι αποθηκευμένες πληροφορίες είναι διαθέσιμες μόνο για ανάγνωση και δεν μπορούν να αλλαχθούν, αλλά χρησιμοποιούνται για εσωτερική χρήση από το PLC. Μια ειδική κατηγορία ROM είναι των προγραμματιζόμενων ROM (PROM) οι οποίες επιτρέπουν την εγγραφή πρόσθετων πληροφοριών στο κύκλωμα της μνήμης για μία και μοναδική φορά, γι' αυτό και δεν προτιμούνται για την εγκατάστασή τους στους ελεγκτές. Η εξέλιξη των PROM είναι οι προγραμματιζόμενες μνήμες μόνο ανάγνωσης με δυνατότητα διαγραφής (EPROM), που δίνουν την δυνατότητα για επαναπρογραμματισμό από την συσκευή προγραμματισμού, αφού είναι εφικτή και η διαγραφή των πληροφοριών. Τις ίδιες δυνατότητες παρέχουν και οι ηλεκτρικά διαγραφόμενες μνήμες μόνο ανάγνωσης (EAROM), με μόνη διαφορά τον τεχνικό τρόπο που χρησιμοποιείται για την διαγραφή των πληροφοριών. Τέλος, υπάρχουν και οι μη πτητικές, ηλεκτρικά διαγραφόμενες και προγραμματιζόμενες μνήμες μόνο ανάγνωσης (EEPROM), οι οποίες ουσιαστικά παρέχουν την ίδια ευελιξία με τις RAM.

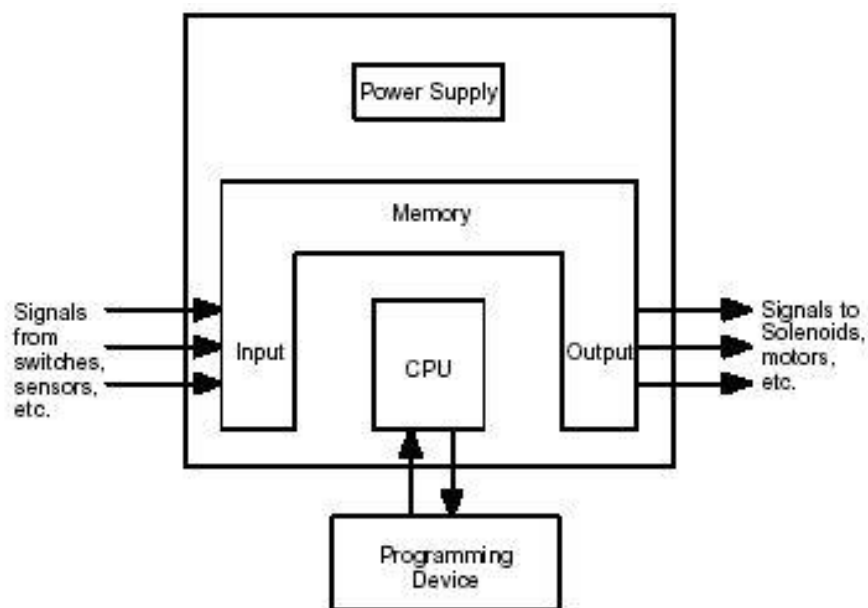
Έχοντας δημιουργήσει μία γενική εικόνα των ειδών μνήμης που υπάρχουν, είναι δυνατή τώρα η παρουσίαση των τρόπων που αυτά χρησιμοποιούνται συνήθως. Έτσι μία CPU έχει συχνά ενσωματωμένες μία RAM εργασίας (Working Memory) και μία RAM φορτώματος (Load Memory), ενώ χρησιμοποιεί και μία εξωτερική Flash EPROM φορτώματος (Load Memory) που επεκτείνει τις ενσωματωμένες μνήμες.

Η μνήμη εργασίας είναι γρηγορότερη της μνήμης φορτώματος και σβήνει είτε με το πάτημα του κουμπιού Reset memory της CPU, ή αν πέσει η μπαταρία του τροφοδοτικού. Η μνήμη φορτώματος περιλαμβάνει όλα τα λογικά block, το block δεδομένων και των δεδομένων παραμετροποίησης που δεν σβήνουν με κανέναν από τους τρόπους που αναφέρθηκαν στην μνήμη εργασίας. Κατά την αλλαγή της κατάστασης της CPU από την κατάσταση Stop στην κατάσταση εκτέλεσης του προγράμματος, μεταφέρονται μόνο τα κομμάτια των block λογικής και δεδομένων που είναι απαραίτητα για την εργασία αυτή, από την μνήμη φορτώματος στην μνήμη εργασίας.

Ανάμεσα στα υπόλοιπα χαρακτηριστικά της CPU, βρίσκεται και η ύπαρξη των led κατάστασης και των led σφαλμάτων. Ο τρόπος λειτουργίας της επιλέγεται με κλειδί, για την προστασία του προγράμματος από μη εξουσιοδοτημένες αλλαγές ή ακόμη και την διαγραφή του. Επίσης μία CPU περιλαμβάνει διαγνωστική μνήμη η οποία παραμένει ακέραιη και σε περίπτωση

πτώσης τάσης, αλλά και κατά την επανεκκίνηση της μνήμης, και καταγράφονται σε αυτήν η ώρα και η ημερομηνία που παρουσιάστηκε σφάλμα της CPU, του συστήματος της CPU, των περιφερειακών μονάδων, αλλαγή κατάστασης από Stop σε Run και το αντίστροφο, αλλά και προγραμματιστικά λάθη στην εφαρμογή. Τέλος μια CPU περιλαμβάνει ένα διαγνωστικό alarm block στο οποίο προγραμματίζοντας τη διεύθυνση μιας οποιασδήποτε κάρτας εισόδου / εξόδου λαμβάνονται διαγνωστικά μηνύματα για αυτήν όπως είναι για παράδειγμα μία πιθανή βλάβη της, η έλλειψη εξωτερικής τάσης αλλά και ο εντοπισμός προβλήματος σε κάποιο κανάλι της κάρτας.

Είναι μόνο λογικό να σημειωθεί σε αυτό το σημείο πως η CPU υποστηρίζει διάφορες γλώσσες προγραμματισμού, οι πιο γνωστές από τις οποίες είναι η LADDER, η FBD, και η STL, αλλά και επιπλέον γλωσσών με την χρήση Optional Software πακέτων. Εκτενέστερη αναφορά στον προγραμματισμό των PLC και στις γλώσσες προγραμματισμού που χρησιμοποιούνται γι' αυτόν θα πραγματοποιηθεί αργότερα.



ΣΧΗΜΑ 1: ΛΕΙΤΟΥΡΓΙΑ CPU

1.2.4 ΜΟΝΑΔΕΣ ΕΙΣΟΔΩΝ/ ΕΞΟΔΩΝ

Νωρίτερα έγινε παραλληλισμός της μονάδας κεντρικής επεξεργασίας με τον εγκέφαλο ενός ελεγκτή προγραμματιζόμενης λογικής. Με την ίδια λογική είναι δόκιμος ένας παραλληλισμός των μονάδων εισόδου / εξόδου με όλα τα αισθητήρια όργανα που συνδέονται με τον εγκέφαλο. Το σύστημα εισόδων / εξόδων, αποτελεί το σημείο επαφής μεταξύ των μονάδων που υπόκεινται σε έλεγχο και του ελεγκτή, ενώ σκοπός του είναι η αποκωδικοποίηση των σημάτων που λαμβάνονται και αποστέλλονται στις εξωτερικές συσκευές.

Τα καλώδια που έρχονται από τις συσκευές εισόδου, όπως είναι οι αισθητήρες και οι διακόπτες συνδέονται στις τερματικές μονάδες εισόδου του ελεγκτή. Αντίστοιχα, τα καλώδια που κατευθύνονται στις συσκευές εξόδου, όπως τα ρελέ ισχύος, οι βαλβίδες και οι λυχνίες, συνδέονται στις τερματικές μονάδες εξόδου.

Οι μονάδες εισόδου λαμβάνουν σήματα από το μηχάνημα και τα μετατρέπουν σε σήματα που μπορεί να χρησιμοποιήσει ο ελεγκτής. Οι μονάδες εξόδου από την άλλη μετατρέπουν τα σήματα του ελεγκτή σε σήματα που μπορούν να χρησιμοποιηθούν για τον έλεγχο της συσκευής ή για την διαδικασία της επεξεργασίας. Σε όλες τις συσκευές εισόδου και εξόδου αποδίδεται μία ορισμένη διεύθυνση, σύμφωνα με την οποία ο επεξεργαστής αναγνωρίζει που είναι τοποθετημένη η κάθε συσκευή, παρέχοντάς του την δυνατότητα με αυτόν τον τρόπο να την ελέγχει.

Ένας ελεγκτής αντιλαμβάνεται ότι ένα αισθητήριο είναι ανοιχτό ή κλειστό από την εμφάνιση ή όχι τάσης L1 στην έξοδο, ενώ αν από το πρόγραμμα δοθεί εντολή διέγερσης, τότε εμφανίζεται τάση L2 στην αντίστοιχη έξοδο. Οι τάσεις L1 και L2 πρέπει να δημιουργηθούν εξωτερικά με κατάλληλο τροφοδοτικό (για DC) ή με μετασχηματιστή τάσης χειρισμού (για AC).

Το τμήμα εισόδων / εξόδων αποτελείται από μία βάση και μεμονωμένες μονάδες εισόδου / εξόδου οι οποίες συνήθως συνοδεύονται από ενδεικτικές λυχνίες για να είναι εμφανής η κατάσταση ON / OFF της κάθε μίας από αυτές, αλλά και για την ένδειξη κάποιας πιθανής καμένης ασφάλειας. Η οποιαδήποτε αλλαγή μονάδων εισόδου / εξόδου γίνεται πολύ εύκολα από την στιγμή που οι συσκευές απλώς συνδέονται με καλώδια στην βάση του τμήματος.

Παρόλο που συναντάμε τις μονάδες εισόδου / εξόδου και στη μορφή των διακριτών, αλλά και στη μορφή των αναλογικών / ψηφιακών, στις αρχές της χρήσης των PLC υπήρχαν μόνο οι διακριτές μονάδες. Οι μονάδες αυτές επέτρεπαν μόνο τη σύνδεση συσκευών τύπου ON / OFF, έχοντας ως άμεσο αποτέλεσμα τον μερικό μόνο έλεγχο των περισσοτέρων εφαρμογών.

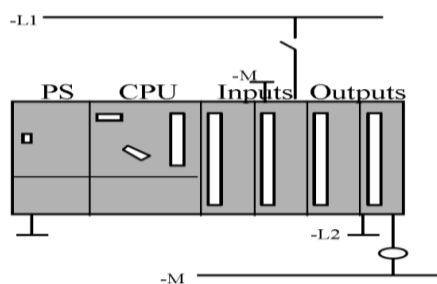
Όπως είναι εύκολα αντιληπτό, τεράστια είναι η διαφορά των διακριτών και των ψηφιακών / αναλογικών μονάδων εισόδου / εξόδου, αφού οι τελευταίες

επιτρέπουν τον έλεγχο σε οποιαδήποτε διαδικασία. Οι αναλογικές μονάδες διασύνδεσης εισόδου δέχονται αναλογικές τάσεις ή ρεύματα από αναλογικές συσκευές και τα μετατρέπουν σε ψηφιακές τιμές ώστε να είναι δυνατή η χρήση τους από τον επεξεργαστή. Αντίστοιχα οι αναλογικές μονάδες διασύνδεσης εξόδου λαμβάνουν από τον επεξεργαστή ψηφιακά δεδομένα και τα μετατρέπουν σε αναλογικό για να ελέγξουν μία αναλογική συσκευή.

Ένας ελεγκτής αντιλαμβάνεται ότι μία εξωτερική επαφή έκλεισε, όταν στην αντίστοιχη είσοδο εμφανίζεται τάση. Η τάση αυτή ονομάζεται τάση εισόδων. Μια μονάδα εισόδων διαθέτει συνήθως 4, 8, 16, ή 32 εισόδους, ανάλογα με τον τύπο του ελεγκτή και τη τάση. Είναι υποχρεωτικό να χρησιμοποιείται η ίδια τάση για όλες τις εισόδους της μονάδας. Τα σπουδαιότερα τεχνικά χαρακτηριστικά τους είναι το πλήθος εισόδων, η γαλβανική απομόνωση, η ονομαστική τάση, οι ανοχές τάσης, η μέγιστη συνολική διαδρομή καλωδίων, το ρεύμα που απορροφάται από τη μονάδα συνολικά και από κάθε είσοδο ειδικά, καθώς και ο τύπος της πρίζας που απαιτείται.

Οι μονάδες ψηφιακών εξόδων χρησιμεύουν στη διέγερση των εξωτερικών στοιχείων της εγκατάστασης, όπου όταν δίνεται η κατάλληλη εντολή από το πρόγραμμα, κλείνει ο διακόπτης της εξόδου, και εμφανίζεται τάση στην έξοδο. Η τάση αυτή ονομάζεται τάση εξόδων. Συνήθως μία μονάδα εξόδων περιλαμβάνει 4, 8, 16, ή 32 εξόδους, ανάλογα με τον τύπο του ελεγκτή και τη τάση. Τα σπουδαιότερα τεχνικά χαρακτηριστικά μίας μονάδας εξόδων είναι το πλήθος εξόδων, η γαλβανική απομόνωση, η ονομαστική τάση, οι ανοχές τάσης, η μέγιστη συνολική διαδρομή καλωδίων, το ρεύμα που απορροφάται από τη μονάδα συνολικά και από κάθε είσοδο ειδικά, η συχνότητα ζεύξεων, καθώς και ο τύπος της πρίζας που απαιτείται.

Τα κυκλώματα και οι τάσεις των εισόδων είναι τελείως ανεξάρτητα από τα κυκλώματα και τις τάσεις των εξόδων. Επομένως, η τάση για τις εισόδους μπορεί να είναι διαφορετική από την τάση για τις εξόδους. Επιπλέον υπάρχει η δυνατότητα και ξεχωριστής τάσης ανά μονάδα εισόδων ή εξόδων.



Δομή προγραμματιζόμενου ελεγκτή. PS = τροφοδοτικό, CPU = κεντρική μονάδα επεξεργασίας, Inputs = εισόδοι, Outputs = εξόδοι, -M = ο ακροδέκτης M του τροφοδοτικού (αρνητικός πόλος)

ΣΧΗΜΑ II: ΔΟΜΗ PLC

1.3 ΟΝΟΜΑΤΟΛΟΓΙΑ ΤΩΝ PLC

Είναι γνωστό από τα παραπάνω πως τα PLC χρησιμοποιούν το δυαδικό σύστημα, γεγονός που συμβαδίζει απόλυτα με την τεχνολογία των αυτοματισμών και γενικότερα των υπολογιστικών συστημάτων, από την στιγμή που όλα τα συστήματα αυτού του τύπου μπορούν μόνο να αντιληφθούν αν τα στοιχεία τους διαρρέονται ή όχι από ηλεκτρικό ρεύμα, με άλλα λόγια αν είναι ανοιχτοί, ή κλειστοί οι συμβατικοί αυτοματισμοί.

Όπως έχει ήδη αναφερθεί, τα PLC έχουν την ίδια οργάνωση μνήμης με τους προσωπικούς υπολογιστές σε bits, και έπειτα σε bytes. Ακολουθώντας την ίδια λογική ονοματίζονται και οι διευθύνσεις των εισόδων και των εξόδων σε ένα PLC, οπότε και ορίζονται με την μορφή:

$I_{x.y}$ $Q_{x.y}$ όπου το x καθορίζει τη διεύθυνση του byte εισόδων / εξόδων, και το y καθορίζει τη διεύθυνση του bit εισόδων / εξόδων. Η μορφή αυτή ισχύει μόνο για τα bit.

I_{xy} Q_{xy} όπου το x καθορίζει το είδος, και το y καθορίζει την αρχική διεύθυνση του bit εισόδων / εξόδων.

Και στις δύο περιπτώσεις ως I ορίζεται η είσοδος, και ως Q ορίζεται η έξοδος.

Με τον ίδιο ακριβώς τρόπο διευθυνσιοδοτούνται και τα βοηθητικά bits, αυτά δηλαδή, που χρησιμοποιούνται από το πρόγραμμα για την αποταμίευση ορισμένων καταστάσεων. Σε αντίθεση με τις εισόδους και τις εξόδους, η κατάσταση των βοηθητικών bits μπορεί να αναγνωρισθεί με τη βοήθεια μιας συσκευής προγραμματισμού.

1.3.1 ΟΝΟΜΑΤΟΛΟΓΙΑ ΧΡΟΝΙΚΩΝ

Με τον όρο χρονικό εννοούμε μία λέξη μέσα σε μία ειδική περιοχή της μνήμης, την περιοχή των χρονικών. Σε αυτή τη περιοχή τοποθετείται με το πρόγραμμα η τιμή του χρόνου. Όταν ορίζεται από το πρόγραμμα, ξεκινάει ή αντίστροφη μέτρηση, η μείωση, δηλαδή, της τιμής που έχει δηλωθεί με τον ρυθμό της χρονικής μονάδας. Την στιγμή που η τιμή του χρονικού μηδενιστεί, λαμβάνουμε ένα αντίστοιχο αξιοποιήσιμο σήμα.

Ένα χρονικό χαρακτηρίζεται μονοσήμαντα με την παρακάτω μορφή:

T_x όπου με το T ορίζουμε το χρονικό και με το x ορίζουμε τον αριθμό του χρονικού.

1.3.2 ΟΝΟΜΑΤΟΛΟΓΙΑ ΑΠΑΡΙΘΜΗΤΩΝ

Με τον όρο απαριθμητής εννοούμε μία λέξη μέσα σε μία ειδική περιοχή της μνήμης, την περιοχή των απαριθμητών. Σε αυτή τη λέξη τοποθετείται κάθε στιγμή το περιεχόμενο του απαριθμητή, το οποίο αυξάνεται ή ελαττώνεται κατά ένα, με κατάλληλες εντολές από το πρόγραμμα. Το περιεχόμενο αυτό μπορεί να ζητηθεί και να αξιολογηθεί.

Ένας απαριθμητής χαρακτηρίζεται μονοσήμαντα με την παρακάτω μορφή:

C x όπου με το C ορίζουμε τον απαριθμητή και με το x ορίζουμε τον αριθμό του απαριθμητή.

1.3.3 ΟΝΟΜΑΤΟΛΟΓΙΑ ΤΥΜΠΑΝΟΕΙΔΩΝ ΕΛΕΓΚΤΩΝ

Με τον όρο τυμπανοειδής ελεγκτής εννοούμε έναν ελεγκτή που αποτελείται από βήματα στα οποία μπορεί να οριστεί μία σειρά ενεργοποιήσεων και απενεργοποιήσεων των κατάλληλων εισόδων και εξόδων του PLC. Έτσι, είναι δυνατόν να ορισθούν σε κάθε βήμα του ποια bits θα ενεργοποιηθούν και ποια θα απενεργοποιηθούν. Ο τυμπανοειδής ελεγκτής αυτός αποτελείται από 8 βήματα και κάθε βήμα μπορεί να ελέγξει 16 bits.

Ένας τυμπανοειδής ελεγκτής χαρακτηρίζεται μονοσήμαντα με την παρακάτω μορφή:

DRx όπου με το DR ορίζουμε τον τυμπανοειδή ελεγκτή και με το x ορίζουμε τον αριθμό του τυμπανοειδούς ελεγκτή, ενώ αν στην παραπάνω ονομασία προστεθεί «.S», τότε καθορίζεται το βήμα του τυμπανοειδούς ελεγκτή.

Στα δομικά αυτά στοιχεία ενός PLC θα γίνει ξανά αναφορά στην ανάλυση του δημιουργηθέντος προγράμματος, αλλά και μία περαιτέρω ανάλυσή τους όπου αυτή καθίσταται απαραίτητη.

ΚΕΦΑΛΑΙΟ 2

ΓΕΝΙΚΑ ΣΤΟΙΧΕΙΑ ΓΙΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΤΩΝ PLC

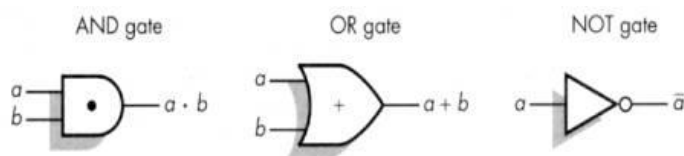
Στο προηγούμενο κεφάλαιο αποπειράθηκε για μία πρώτη επαφή με τα PLC. Αναλύθηκαν η δομή, η ονοματολογία, αλλά και τα πλεονεκτήματά τους, ενώ συγχρόνως έγινε και μία αναφορά στον προγραμματισμό τους. Σε αυτό το κεφάλαιο, θα αναπτυχθεί σε βάθος αυτή ακριβώς η έκφανση των PLC, σε μία προσπάθεια απλούστευσης και κατανόησής της, έτσι ώστε να γίνει πιο ξεκάθαρος ο λόγος της κυριάρχησής τους στην βιομηχανία.

2.1 ΛΟΓΙΚΑ ΚΥΚΛΩΜΑΤΩΝ

Είναι απαραίτητο πριν την οποιαδήποτε αναφορά στον προγραμματισμό των PLC, να αναλυθεί η λειτουργία των λογικών κυκλωμάτων, από τη στιγμή που οι προγραμματιζόμενοι λογικοί ελεγκτές αποτελούνται από αυτά και ακολουθούν τις αρχές τους. Έτσι θα γίνει πιο κατανοητός ο τρόπος λειτουργίας τους, άρα και η λογική του προγραμματισμού τους η οποία θα παρουσιαστεί αργότερα.

Ως λογικά κυκλώματα θα μπορούσαν να οριστούν τα κυκλώματα εκείνα, που μετατρέπουν δυαδικές τιμές εισόδου (0 ή 1) σε δυαδικές τιμές εξόδου, και αποτελούνται από μεγάλο αριθμό αλληλοσυνδεδεμένων λογικών πυλών. Οι λογικές αυτές πύλες δεν είναι άλλο από μία διαφορετική άποψη στις λογικές πράξεις και τιμές αληθείας (άλγεβρα Boole). Ουσιαστικά αποτελούν ένα κύκλωμα διακοπών, μία σύνθετη λογική πρόταση δηλαδή, κατασκευασμένη από απλές λογικές προτάσεις οι οποίες συνδέονται μεταξύ τους με λογικές πράξεις. Μία λογική πράξη μπορεί να θεωρηθεί ως μία πύλη που συνδυάζει μία ή περισσότερες εισόδους για να παράγει μία έξοδο. Η συμπεριφορά της περιγράφεται από έναν πίνακα αληθείας.

Οι βασικές πύλες με τον συμβολισμό τους, αλλά και τον πίνακα αληθείας τους παρατίθενται στις εικόνες που ακολουθούν.



ΣΧΗΜΑ III: ΠΥΛΕΣ AND, OR, NOT

ΠΙΝΑΚΑΣ Ι: ΠΙΝΑΚΕΣ ΑΛΗΘΕΙΑΣ ΠΥΛΩΝ AND, OR, NOT

a	b		a · b
0	0		0
0	1		0
1	0		0
1	1		1

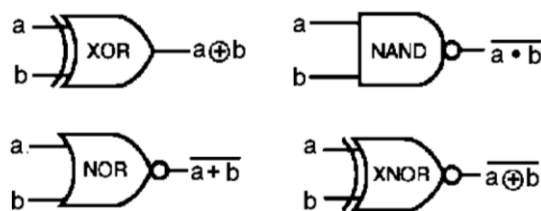
a	b		a + b
0	0		0
0	1		1
1	0		1
1	1		1

a		\bar{a}
0		1
1		0

Ο εγγενής κανόνας της σύζευξης (AND), είναι πως αν ένα bit στη μία είσοδο είναι 0, δεν χρειάζεται να ελεγχθεί το άλλο, αφού το αποτέλεσμα είναι 0. Στην περίπτωση της διάζευξης (OR), το αποτέλεσμα είναι 1 αν έστω και ένα bit στη μία είσοδο είναι 1. Τέλος, για την άρνηση (NOT) ισχύει πως το αποτέλεσμα είναι πάντα αντίθετο της εισόδου. Οι κανόνες αυτοί είναι δυνατόν να γενικευτούν και σε περισσότερες των μία πυλών που βρίσκονται παράλληλα συνδεδεμένες.

Οι συνδυασμοί των λογικών πυλών AND, OR και NOT αρκούν για όλες τις λογικές πράξεις με οποιονδήποτε αριθμό εισόδων, όμως υπάρχουν ακόμη τέσσερις λογικές πύλες οι οποίες χρησιμοποιούνται. Οι πύλες αυτές είναι η XOR (αποκλειστική διάζευξη), η NAND (άρνηση σύζευξης), η NOR (άρνηση διάζευξης) και η XNOR (άρνηση αποκλειστικής διάζευξης ή αλλιώς λογική ισοδυναμία).

Στην περίπτωση της XOR, η έξοδος είναι αληθής όταν ακριβώς μία από τις εισόδους είναι αληθής. Για την NAND ισχύει πως η έξοδος είναι αληθής όταν το πολύ μία από τις δύο εισόδους είναι αληθής. Στην NOR, η έξοδος είναι αληθής όταν καμία από τις δύο εισόδους δεν είναι αληθής. Η πύλη XNOR έχει αληθή έξοδο όταν οι δύο εισοδοί είναι ίδιες (0 ή 1).



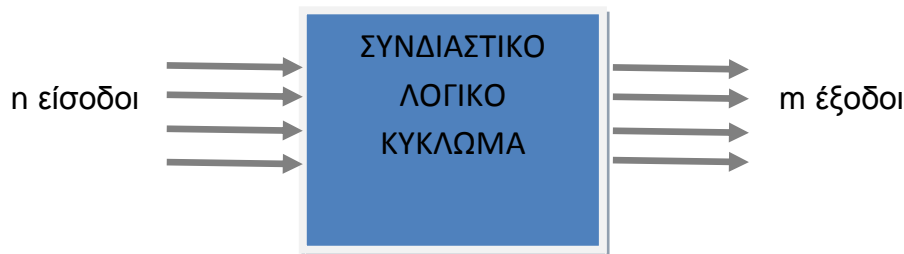
ΣΧΗΜΑ IV: ΠΥΛΕΣ XOR, NAND, NOR, XNOR

ΠΙΝΑΚΑΣ ΙΙ: ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ ΠΥΛΩΝ XOR, NAND, NOR, XNOR

a	b	a XOR b	a NAND b	a NOR b	a XNOR b
0	0	0	1	1	1
0	1	1	1	0	0
1	0	1	1	0	0
1	1	0	0	0	1

Τα λογικά κυκλώματα διακρίνονται σε συνδυαστικά (combinational) και ακολουθιακά (sequential). Η διάκριση αυτή γίνεται γιατί τα συνδυαστικά κυκλώματα περιέχουν μόνο συνδυαστικές λογικές πύλες, ενώ τα ακολουθιακά κυκλώματα περιέχουν επιπλέον και στοιχεία μνήμης.

Ένα συνδυαστικό κύκλωμα αποτελείται από μεταβλητές εισόδου, λογικές πύλες και μεταβλητές εξόδου. Τα σήματα στις εισόδους μίας πύλης παράγουν στην έξοδο της ένα σήμα που περιγράφεται αναλυτικά από λογικές συναρτήσεις. Γενικά σε ένα συνδυαστικό κύκλωμα στο οποίο επενεργούν n σήματα εισόδου, και τα οποία παράγουν m σήματα εξόδου, οι n λογικές μεταβλητές εισόδου παράγουν 2^n δυνατούς συνδυασμούς τιμών 0 και 1 στην είσοδο. Για κάθε έναν από τους 2^n πιθανούς συνδυασμούς δυαδικών τιμών στις εισόδους, υπάρχει ένας και μόνο ένας συνδυασμός δυαδικών τιμών στις εξόδους.



ΣΧΗΜΑ V: ΣΥΝΔΙΑΣΤΙΚΟ ΛΟΓΙΚΟ ΚΥΚΛΩΜΑ

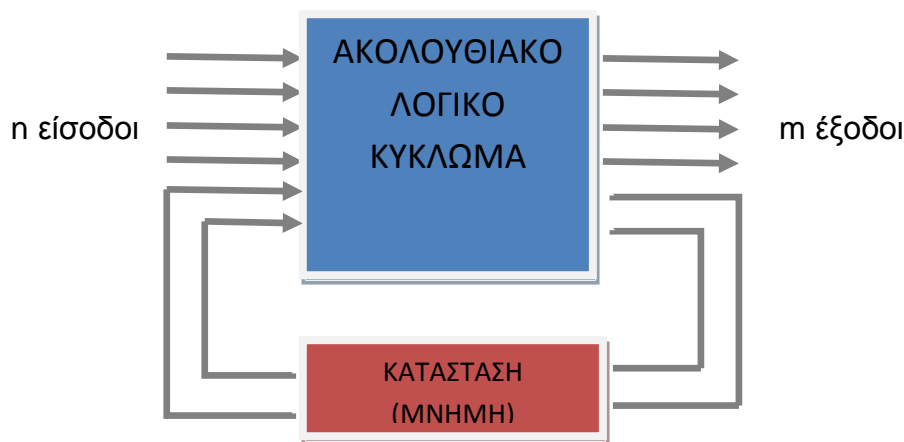
Έτσι ένα συνδυαστικό κύκλωμα περιγράφεται με m συναρτήσεις Boole n μεταβλητών, μία συνάρτηση για κάθε μεταβλητή εξόδου. Από τα παραπάνω προκύπτει πως κάθε μία F_i από τις m συναρτήσεις στην έξοδο του συνδυαστικού κυκλώματος είναι συνάρτηση των μεταβλητών εισόδου, δηλαδή :

$$\begin{aligned}
 F_1 &= F_1(X_1, X_2, X_3, \dots, X_n) \\
 F_2 &= F_2(X_1, X_2, X_3, \dots, X_n) \\
 F_3 &= F_3(X_1, X_2, X_3, \dots, X_n) \\
 &\vdots \\
 F_m &= F_m(X_1, X_2, X_3, \dots, X_n)
 \end{aligned}$$

Κάθε μία από τις παραπάνω λογικές συναρτήσεις πριν απλοποιηθεί, εκφράζεται είτε ως άθροισμα ελαχιστόρων είτε ως γινόμενο μεγιστόρων. Το αποτέλεσμα της απλοποίησης κάθε μίας συνάρτησης F_i ($i = 1, 2, 3, \dots, m$) εξόδου, είναι η υλοποίησή της με την χρήση λογικών πυλών. Με τον όρο απλοποίηση συνάρτησης εννοούμε την διαδικασία εκείνη κατά την οποία γίνεται προσπάθεια για την ελαχιστοποίηση του αριθμού των πυλών, του αριθμού των εισόδων

πύλης, του χρόνου διάδοσης σήματος, καθώς και του αριθμού των διασυνδέσεων.

Η ειδοποιός διαφορά ανάμεσα στα συνδυαστικά και στα ακολουθιακά κυκλώματα είναι πως στα συνδυαστικά οι έξοδοι σε κάθε χρονική στιγμή είναι συνάρτηση των εισόδων εκείνης της χρονικής στιγμής και μόνο, ενώ στα ακολουθιακά οι έξοδοι είναι συνάρτηση των εισόδων εκείνης της χρονικής στιγμής, καθώς και της κατάστασης των στοιχείων μνήμης, η οποία αποτελεί συνάρτηση των προηγούμενων εισόδων.



ΣΧΗΜΑ VI: ΑΚΟΛΟΥΘΙΑΚΟ ΛΟΓΙΚΟ ΚΥΚΛΩΜΑ

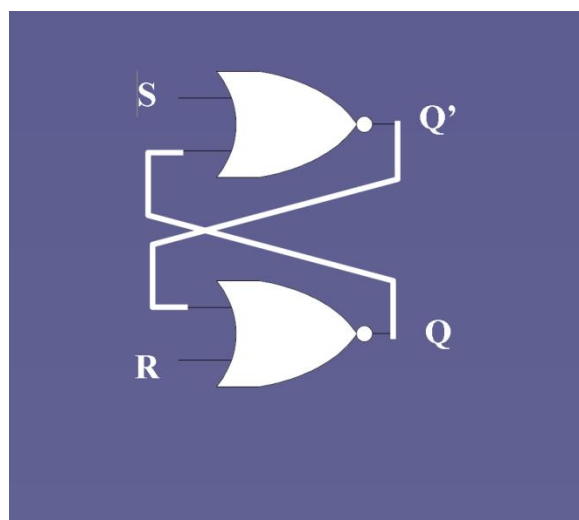
Με άλλα λόγια, τα ακολουθιακά κυκλώματα, αποτελούν εξέλιξη των συνδυαστικών αφού αποτελούνται από αυτά, καθώς και από κυκλώματα αποθήκευσης δυαδικών καταστάσεων (στοιχεία μνήμης). Το στοιχείο μνήμης του οποίου γίνεται χρήση στα ακολουθιακά κυκλώματα, ονομάζεται flip-flop (δισταθής πολυδονητής), και μπορεί να αποθηκεύσει πληροφορία ενός bit (0 και 1). Το flip-flop (FF) είναι ένα λογικό κύκλωμα διαθέτει δύο σταθερές καταστάσεις. Μπορεί είτε να παραμείνει μόνιμα στην κατάσταση 0, είτε στην κατάσταση 1. Η αλλαγή της κατάστασης ενός flip-flop (από 1 σε 0 ή από 0 σε 1), γίνεται μόνο με την εφαρμογή εξωτερικών σημάτων.

Τα flip-flop εκτελούν πολλές λειτουργίες σε ένα υπολογιστικό σύστημα, όπως είναι η αποθήκευση τιμών ενώ είναι δυνατή η χρήση τους στην πραγματοποίηση διαφόρων πράξεων, αλλά και ως δείκτες της σειράς των πράξεων που εκτελούνται. Δύο ή και περισσότερα flip-flop κατάλληλα συνδεδεμένα, σχηματίζουν δυαδικούς μετρητές ή καταχωρητές. Υπάρχουν δύο τύποι flip-flop, τα ασύγχρονα και τα σύγχρονα.

Συνδυάζοντας τα παραπάνω, γίνεται εύκολα αντιληπτή η κατάταξη των PLC στα ακολουθιακά λογικά κυκλώματα, γεγονός που μας υποχρεώνει σε περεταίρω ανάπτυξη τους, και συγκεκριμένα στην ανάλυση των λειτουργιών των flip-flop. Είναι απαραίτητη η ανάλυση της στοιχειώδους λειτουργίας των κυκλωμάτων αυτών για να αποτελεί εφικτή η κατανόηση των πιο εξειδικευμένων αναφορών στον προγραμματισμό των PLC.

2.1.1 ΑΣΥΓΧΡΟΝΑ FLIP-FLOP

Στα ασύγχρονα flip-flop η συμπεριφορά του κυκλώματος εξαρτάται από την ακολουθία με την οποία αλλάζουν οι τιμές των εισόδων και εξόδων σε οποιαδήποτε χρονική στιγμή. Λειτουργούν χωρίς ωρολογιακό παλμό, και διαθέτουν δύο εισόδους και δύο εξόδους. Ένα βασικό, ασύγχρονο flip-flop είναι δυνατόν να αποθηκεύσει πληροφορία ενός bit, και αποτελείται από δύο πύλες NOR συνδεδεσολογημένες με τέτοιο τρόπο, ώστε η έξοδος Q της πρώτης να οδηγείται στην είσοδο S της δεύτερης και η είσοδος R στην έξοδο Q'. Με άλλα λόγια, οι έξοδοι, επιστρέφουν πάλι στην είσοδο, οπότε κάθε σήμα εισόδου, περνά μέσα από το κύκλωμα, και επιστρέφει ξανά στην είσοδο. Η διαδικασία αυτή ονομάζεται ανάδραση, και δίνει την δυνατότητα μνήμης στο κύκλωμα.



ΣΧΗΜΑ VII: ΑΣΥΓΧΡΟΝΟ FLIP-FLOP

ΠΙΝΑΚΑΣ III: ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ ΑΣΥΓΧΡΟΝΟΥ FLIP-FLOP

S	R	Q	Q'
0	0	Q	Q'
0	1	0	1
1	0	1	0
1	1	0	0

Αποκωδικοποιώντας τον πίνακα αλήθειας που υπάρχει στην εικόνα, βγάζουμε τα παρακάτω συμπεράσματα:

- Για $S = '0'$, $R = '0'$: Με την κατάσταση αυτή των εισόδων, η έξοδος του flip-flop παραμένει στην προηγούμενη κατάσταση.
- Για $S = '0'$, $R = '1'$: Με την κατάσταση αυτή των εισόδων, γίνεται καθαρισμός της εξόδου του flip-flop.
- Για $S = '1'$, $R = '0'$: Με την κατάσταση αυτή των εισόδων, το flip-flop τοποθετείται στην κατάσταση '1'.
- Για $S = 1$, $R = 1$: Με την κατάσταση αυτή των εισόδων, το flip-flop βρίσκεται σε απροσδιοριστία.

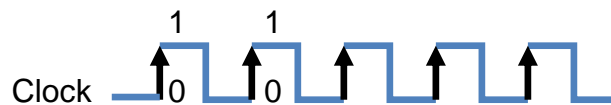
Βασικό μειονέκτημα των ασύγχρονων flip-flop είναι ότι οι καταστάσεις των εξόδων αλλάζουν σε οποιαδήποτε χρονική στιγμή σε σχέση με τις καταστάσεις των εισόδων, με αποτέλεσμα η αλλαγή να γίνεται σχεδόν ταυτόχρονα με τις εισόδους. Αυτό οφείλεται στο ότι στο flip-flop λείπει ο συγχρονισμός, δεν υπάρχει με άλλα λόγια κάποια αρχή με βάση την οποία να ορίζεται η μετατροπή των καταστάσεών τους. Το μειονέκτημα αυτό είναι πολύ σημαντικό, αφού έχει ως αποτέλεσμα την τυχαία αλλαγή και αποθήκευση των δεδομένων.

Λύση σε αυτό το πρόβλημα δίνει το σύγχρονο ή clocked flip-flop.

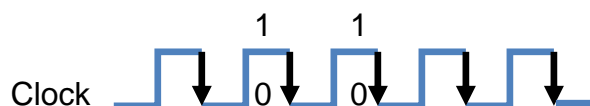
2.1.2 ΣΥΓΧΡΟΝΑ FLIP-FLOP

Στα σύγχρονα flip-flop η συμπεριφορά του κυκλώματος ορίζεται από τις τιμές των εισόδων και των εξόδων σε συγκεκριμένες χρονικές στιγμές. Στην πράξη οι δύο ασύγχρονες εισοδοί ελέγχου της κατάστασης του βασικού flip-flop που αναλύθηκε παραπάνω, ενεργοποιούνται μόνο σε καθορισμένες χρονικές στιγμές που ισαπέχουν. Αυτό επιτυγχάνεται με την χρήση ενός ωρολογιακού παλμού, που παράγεται από μία γεννήτρια παλμών και η οποία δίνει εντελώς όμοιους παλμούς με μία σταθερή συχνότητα. Οι παλμοί αυτοί διανέμονται σε όλα τα μέρη του συστήματος, έτσι ώστε αυτά να λειτουργούν συγχρονισμένα. Με την χρήση των παλμών εκτός από την επίτευξη του συγχρονισμού του συστήματος, πραγματοποιείται και ο προσδιορισμός της ταχύτητας λειτουργίας του.

Ανάλογα με το μέτωπο διέγερσης, έχουμε δύο τύπους σύγχρονων flip-flop, τα θετικής και τα αρνητικής διέγερσης. Ο διαχωρισμός γίνεται ανάλογα με το εάν τα δεδομένα εισόδου περνούν στην έξοδο με το θετικό (\uparrow) ή το αρνητικό (\downarrow) μέτωπο του παλμού.

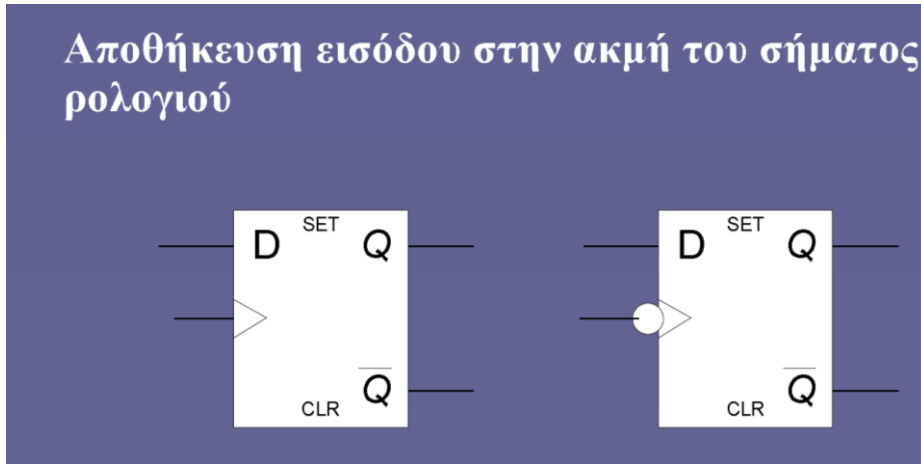


ΣΧΗΜΑ VIII: ΩΡΟΛΟΓΙΑΚΟΣ ΠΑΛΜΟΣ ΜΕ ΜΕΤΩΠΟ ΘΕΤΙΚΗΣ ΔΙΕΓΕΡΣΗΣ



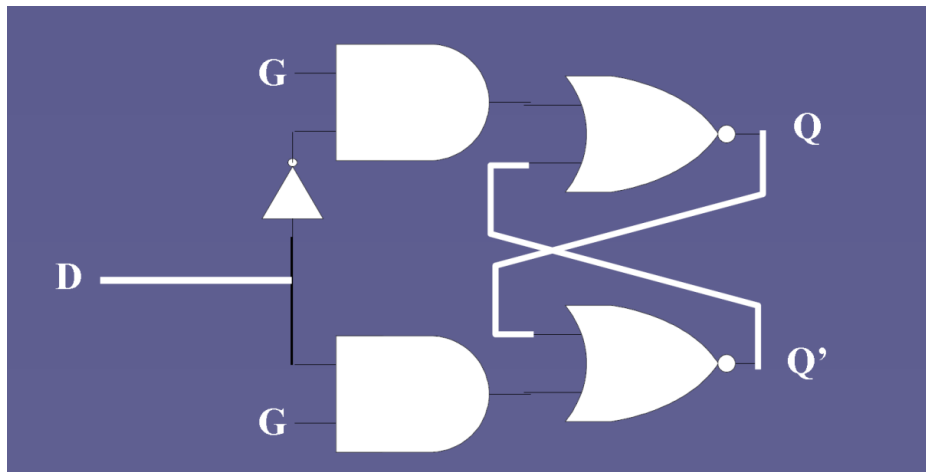
ΣΧΗΜΑ IX: ΩΡΟΛΟΓΙΑΚΟΣ ΠΑΛΜΟΣ ΜΕ ΜΕΤΩΠΟ ΑΡΝΗΤΙΚΗΣ ΔΙΕΓΕΡΣΗΣ

Σε σύγκριση με τα ασύγχρονα flip-flop, τα σύγχρονα έχουν μία επιπλέον είσοδο, αυτή του παλμού clock. Έτσι, οι έξοδοί τους αλλάζουν κατάσταση τη χρονική στιγμή που εμφανίζεται το μέτωπο διέγερσης του παλμού. Η είσοδος του παλμού σημειώνεται με το D ή με ένα βέλος (\rightarrow) στην είσοδο του flip-flop. Για την εύκολη διάκριση ανάμεσα στα flip-flop θετικής και αρνητικής διέγερσης, τοποθετείται ένας κύκλος στην είσοδο του παλμού clock μπροστά από το βέλος, που σημαίνει πως είναι αρνητικής διέγερσης, ενώ χωρίς τον κύκλο σημαίνει πως είναι θετικής διέγερσης.



ΣΧΗΜΑ X: ΔΙΑΚΡΙΣΗ FLIP-FLOP ΑΡΝΗΤΙΚΗΣ ΚΑΙ ΘΕΤΙΚΗΣ ΔΙΕΓΕΡΣΗΣ

Μια από τις απλούστερες μορφές σύγχρονου flip-flop ονομάζεται μανταλωτής και αλλάζει κατάσταση σε σχέση με την εισόδο G, τη χρονική στιγμή που ο παλμός πηγαίνει από '0' σε '1' και σε όλη τη χρονική διάρκεια που το clock είναι στην κατάσταση '1'. Υλοποιείται με την χρήση ενός βασικού ασύγχρονου κυκλώματος, δύο πυλών AND και μίας NOT (ή μιας NAND και μιας AND) και της εισόδου του παλμού.



ΣΧΗΜΑ XI: ΜΑΝΤΑΛΩΤΗΣ

Το πρόβλημα της απροσδιοριστίας δεν παρουσιάζεται στο σύγχρονο αυτό κύκλωμα, αφού γίνεται αποφυγή της με την χρήση της πύλης NOT έτσι ώστε οι εισόδοι του flip-flop να είναι πάντα συμπληρωματικές. Έτσι προκύπτει ο παρακάτω πίνακας αληθείας.

ΠΙΝΑΚΑΣ IV: ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ ΜΑΝΤΑΛΩΤΗ

D	G	Q	Q'
X	0	Q	Q'
0	1	0	1
1	1	1	0

ΒΑΣΙΚΟΙ ΠΑΡΑΜΕΤΡΟΙ ΤΩΝ ΣΥΓΧΡΟΝΩΝ FLIP-FLOP

Οι βασικοί παράμετροι ενός σύγχρονου flip-flop παρέχονται από την κατασκευάστρια εταιρία, και είναι:

F_{max}: Αναφέρεται στη μεγαλύτερη συχνότητα που είναι εφικτό να εφαρμοστεί στον παλμό clock, ώστε το flip-flop να λειτουργεί ασφαλώς.

T_{pd}: Αφορά τον χρόνο καθυστέρησης του σήματος εισόδου κατά τη μετάβαση της πληροφορίας από την είσοδο στην έξοδο ενός κυκλώματος.

T_{set-up}: Είναι ο ελάχιστος χρόνος που πρέπει να διατηρηθούν τα δεδομένα στις εισόδους των flip-flop, πριν το μέτωπο του παλμού φτάσει στην είσοδο D.

T_{hold}: Αφορά τον ελάχιστο χρόνο διατήρησης των δεδομένων στις εισόδους των flip-flop, αφού το μέτωπο του παλμού φτάσει στην είσοδο D.

PRESET Low: Αναφέρεται στον ελάχιστο χρόνο που πρέπει η κατάσταση '0' να παραμείνει στις εισόδους για να κάνει 'SET' ή 'CLEAR' το flip-flop.

T_{w,high}: Είναι ο ελάχιστος χρόνος που θα πρέπει ο παλμός να μείνει σε κατάσταση '1' για την ασφαλή λειτουργία του flip-flop.

T_{w,low}: Είναι ο ελάχιστος χρόνος που θα πρέπει ο παλμός να μείνει σε κατάσταση '0' για την ασφαλή λειτουργία του flip-flop.

2.2 ΣΥΣΚΕΥΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΤΩΝ PLC

Πριν από οποιαδήποτε εμβάθυνση στον τρόπο προγραμματισμού των PLC, είναι λογική η αναφορά στις συσκευές προγραμματισμού. Η συσκευή προγραμματισμού, αποτελεί το μοναδικό μέσο του χρήστη για επικοινωνία με τον προγραμματιζόμενο λογικό ελεγκτή και τα κυκλώματά του.

Ο χρήστης μπορεί να επεμβαίνει στον ελεγκτή, από τη στιγμή που του δίνεται αυτή η δυνατότητα από την συσκευή προγραμματισμού. Έτσι, είναι εφικτό να εισάγει ένα πρόγραμμα, να το ελέγξει, και από τη στιγμή που αυτό παρουσιάσει κάποιο πρόβλημα ή δεν ικανοποιεί τις απαιτήσεις του, να το τροποποιήσει όπως αυτός επιθυμεί.

Για τον προγραμματισμό ενός PLC είναι δυνατόν να χρησιμοποιηθεί οποιαδήποτε τερματική μονάδα η οποία διαθέτει μονάδα απεικόνισης, όπως μία οθόνη, ένα πληκτρολόγιο για την συγγραφή του προγράμματος, καθώς και τα απαραίτητα ηλεκτρονικά για την επικοινωνία του με την CPU. Σήμερα, είναι δυνατόν να χρησιμοποιηθεί ακόμη και ένας φορητός προσωπικός υπολογιστής, αν και πολύ συχνά απαντώνται οι βιομηχανικές τερματικές μονάδες, και οι μικροπρογραμματιστές, οι οποίοι είναι φθηνά, φορητά τερματικά με οθόνες από LED και πληκτρολόγια που αποτελούνται από τα απαραίτητα αριθμητικά πλήκτρα, τα πλήκτρα εντολών και τα πλήκτρα ειδικών λειτουργιών.



ΕΙΚΟΝΑ VI: ΣΥΣΚΕΥΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ PLC

Όταν δεν υπήρχε η δυνατότητα που υπάρχει σήμερα για χρήση προσωπικών υπολογιστών στον προγραμματισμό και την «φόρτωσή» του προγράμματος στον ελεγκτή, χρησιμοποιούνταν ειδικές συσκευές που ονομαζόταν φορτωτές προγράμματος και διαχωριζόταν στις μαγνητικές κασέτες και στις μονάδες ηλεκτρονικής μνήμης. Η λειτουργία των φορτωτών προγράμματος, είναι η αποθήκευση και εγκατάσταση του προγράμματος στον ελεγκτή, και επιτελείται αυτόματα πλέον από έναν προσωπικό υπολογιστή από όπου και φορτώνεται το πρόγραμμα στον ελεγκτή μέσω ενός καλωδίου άμεσης σύνδεσης ή με την χρήση δικτύου.

2.3 ΑΡΧΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΤΩΝ PLC

Προγραμματισμός ενός PLC είναι η δημιουργία μιας σειράς εντολών, οι οποίες είναι σε θέση να λύσουν έναν συγκεκριμένο αλγόριθμο που αντιστοιχεί σε μια λειτουργία ενός συστήματος αυτοματισμού. Η διαδικασία που ακολουθείται για την συγγραφή αυτών των εντολών, αποτελεί το πρόγραμμα.

Απαραίτητη προϋπόθεση για την κατανόηση του τρόπου προγραμματισμού του PLC, είναι η αντίληψη της «φιλοσοφίας» στην οποία στηρίζεται η λειτουργία του. Κάθε ενέργεια του PLC υπαγορεύεται από τις εντολές που του δίνονται από το πρόγραμμα, όπου ορθές θεωρούνται φυσικά εκείνες που είναι σε θέση το PLC να «κατανοήσει» και να εκτελέσει.

Το πρόγραμμα εφαρμογής αποτελείται από μια σειρά οδηγιών που εκτελούνται σειριακά, δηλαδή η μία μετά την άλλη, και κυκλικά, οπότε μετά την τελευταία οδηγία εκτελείται και πάλι η πρώτη, για όλη τη διάρκεια που ο επεξεργαστής είναι σε κατάσταση λειτουργίας. Η διαδικασία αυτή κατά την οποία αναγιγνώσκεται η κατάσταση των εισόδων και ανάλογα με τα αποτελέσματα της ανάγνωσης αυτής και του προγράμματος γίνεται η αντίστοιχη ενημέρωση των εξόδων, ονομάζεται σάρωση. Η σάρωση αποτελεί μία από τις βασικές διαφορές των υπολογιστών με τους ελεγκτές, καθώς ένας υπολογιστής, επεξεργάζεται τις εντολές του προγράμματος, καταλήγει στα αποτελέσματα και σταματά, ενώ ο ελεγκτής επαναλαμβάνει την όλη διαδικασία από την αρχή μετά την απόκτηση των αποτελεσμάτων.

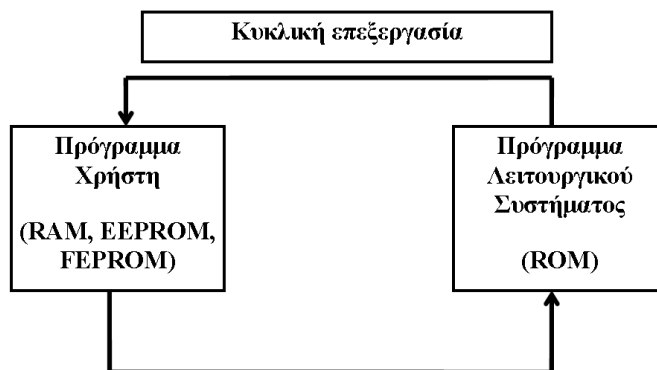


ΣΧΗΜΑ XII: ΤΥΠΙΚΗ ΑΚΟΛΟΥΘΙΑ ΣΑΡΩΣΗΣ

Είναι ιδιαίτερα σημαντικό να τονιστεί σε αυτό το σημείο ότι η πληροφορία για την κατάσταση της εισόδου αποκτάται μόνο στην αρχή του κύκλου, και κατά το χρόνο εκτέλεσης του προγράμματος θεωρείται σταθερή, ενώ ακόμα και αν αλλάξει κατά τη διάρκεια ενός κύκλου, ο κύκλος του PLC είναι τόσο σύντομος, που η CPU θα το αντιληφθεί στον επόμενο.

Με τον όρο κύκλο εννοούμε τον χρόνο κύκλου, δηλαδή, τον χρόνο που απαιτείται για την επεξεργασία και την εκτέλεση, για μία φορά, όλων των εντολών που είναι γραμμένες στο πρόγραμμα. Αυτονόητο είναι ότι το μέγεθος του προγράμματος επηρεάζει άμεσα και τον χρόνο κύκλου, μεγαλώνοντας ή μικραίνοντάς τον ανάλογα. Συνήθως ο χρόνος κύκλου για K εντολές (όπου K είναι ίσο με 1024 εντολές), είναι της τάξεως των μερικών msec.

Το γενικό πρόγραμμα μίας κεντρικής μονάδας αποτελείται από δύο μέρη, το λειτουργικό σύστημα και το πρόγραμμα του χρήστη. Το λειτουργικό σύστημα είναι ουσιαστικά το σύνολο όλων εκείνων των εντολών και των δηλώσεων που ελέγχουν τις πηγές του συστήματος, τις διαδικασίες που χρησιμοποιούν αυτές οι πηγές, καθώς και περιοχές λειτουργίας όπως είναι η αποθήκευση των δεδομένων στην περίπτωση πτώσης της τάσης του δικτύου. Στο λειτουργικό σύστημα ο χρήστης δεν έχει πρόσβαση και δικαιώματα γραφής.



ΣΧΗΜΑ XIII: ΚΥΚΛΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ

Το πρόγραμμα του χρήστη είναι αποτέλεσμα διαφόρων τμημάτων τα οποία η κεντρική μονάδα επεξεργασίας εκτελεί ως αντίδραση στην ύπαρξη ή όχι συγκεκριμένων γεγονότων. Ένα τέτοιο γεγονός μπορεί να είναι η εκκίνηση του συστήματος, μια διακοπή ή η ανίχνευση ενός σφάλματος.

2.4 ΔΟΜΗ ΠΡΟΓΡΑΜΜΑΤΟΣ

Ο διαχωρισμός ενός προγράμματος χρήστη σε μέρη είναι εφικτός και επιθυμητός, χωρίς να διαφοροποιεί κάτι ο αριθμός των τμημάτων αυτών. Η εκτέλεσή τους από την κεντρική μονάδα επεξεργασίας εξαρτάται από το αν λαμβάνουν χώρα κάποια γεγονότα, όπως είναι μία ανίχνευση σφάλματος. Τα προγράμματα που εξαρτώνται από τέτοια γεγονότα χωρίζονται σε τάξεις προτεραιότητας με τις οποίες καθορίζεται η σειρά εκτέλεσης των τμημάτων του προγράμματος όταν συμβούν τα γεγονότα αυτά.

Το κυρίως πρόγραμμα αντιμετωπίζεται ως χαμηλής προτεραιότητας πρόγραμμα και όπως έχει ήδη αναφερθεί εκτελείται κυκλικά, ενώ όλα τα υπόλοιπα γεγονότα έχουν την δυνατότητα να το διακόψουν σε οποιοδήποτε σημείο, οπότε και η CPU εκτελεί την ανάλογη ρουτίνα διακοπής ή αποσφαλμάτωσης πριν επιστρέψει σε αυτό.

Η λογική πίσω από την διαμέριση προγραμμάτων είναι η διευκόλυνση του χρήστη στην ανάγνωσή τους, και η απόκτηση καλύτερης αντίληψης των λειτουργιών τους. Βέβαια κάθε μέρος του προγράμματος πρέπει να έχει

τεχνολογική και λειτουργική βάση. Το τμήμα του προγράμματος χρήστη το οποίο καθορίζεται από τις λειτουργίες του, τη δομή και τον σκοπό της ύπαρξής του, ονομάζεται block. Αυτό το γνώρισμα το συναντάμε και στον προγραμματισμός ενός προσωπικού ηλεκτρονικού υπολογιστή, όπου το πρόγραμμα του χωρίζεται σε διάφορα τμήματα.

Έτσι, σε κάθε γεγονός υφίσταται και ένα συγκεκριμένο block οργάνωσης (Organization Block), το οποίο αναπαριστά την προτεραιότητα του γεγονότος στο πρόγραμμα του χρήστη. Οπότε όταν συμβαίνει ένα γεγονός, το ανάλογο OB καλείται από την κεντρική μονάδα επεξεργασίας. Ο τύπος, ο αριθμός των καταχωρημένων γεγονότων και τα ανάλογα OB εξαρτώνται από την κεντρική μονάδα επεξεργασίας. Το κυρίως πρόγραμμα βρίσκεται μέσα στο OB1, με το τέλος της εκτέλεσης του οποίου η CPU επιστρέφει στο λειτουργικό σύστημα και αφού ολοκληρώσει ορισμένες λειτουργίες του συστήματος, που μία από αυτές είναι και η ενημέρωση του πίνακα διευθύνσεων των εξόδων, καλεί το OB1 για άλλη μία φορά.

Η διακοπή του προγράμματος είναι δυνατόν να συμβεί από γεγονότα όπως είναι οι αιτήσεις διακοπών και τα σφάλματα. Οι διακοπές μπορούν να ζητηθούν είτε από την διαδικασία (διακοπές υλικού) είτε από την CPU (διακοπές εποπτείας). Όσον αφορά τα σφάλματα, υπάρχει ένας διαχωρισμός ανάμεσα σε συγχρονισμένα και ασύγχρονα.

Τα συγχρονισμένα σφάλματα προκαλούνται από την εκτέλεση του προγράμματος, ενώ τα ασύγχρονα είναι ανεξάρτητα από τον κύκλο του προγράμματος. Παράδειγμα συγχρονισμένου σφάλματος είναι η αναφορά σε μη υπαρκτή διεύθυνση, ενώ στα ασύγχρονα σφάλματα συμπεριλαμβάνεται η διακοπή τροφοδοσίας σε μία μονάδα επέκτασης.

Υπάρχουν πολύ τύποι block, αλλά είναι δυνατή η ομαδοποίησή τους στα block χρήστη, τα οποία περιέχουν το πρόγραμμα και τα δεδομένα του χρήστη, στα block του συστήματος, που περιέχουν το πρόγραμμα και τα δεδομένα του συστήματος, και τέλος στα standard block, τα block εκείνα δηλαδή, που αποτελούν το κλειδί της λειτουργίας των οδηγών των ειδικών καρτών CP και FM.

Αναφερόμενοι στα block χρήστη, είναι δυνατή η επιλογή ανάμεσα στα block οργάνωσης (OB), στα block λειτουργίας (FB), στις λειτουργίες (Fc), και στα block δεδομένων. Ο αριθμός των block ανά τύπο, όπως και το μήκος τους, εξαρτώνται από την CPU. Ειδικά για τα block οργάνωσης, πρέπει να αναφερθεί ότι ο αριθμός είναι καθορισμένος, και ανατίθεται από το λειτουργικό σύστημα. Στα υπόλοιπα block, είναι δυνατός ο ορισμός του αριθμού από τον χρήστη, εφόσον βέβαια δεν υπερβαίνει κάποια όρια. Είναι δυνατή επίσης η απόδοση ονόματος από το χρήστη σε κάθε block, έτσι ώστε οποιαδήποτε μετέπειτα αναφορά σε αυτά να γίνεται με το όνομα αυτό.

Τα block συστήματος αποτελούν μέρος του λειτουργικού συστήματος, και είναι δυνατόν να περιέχουν λειτουργίες του ή block λειτουργιών ή ακόμη και δεδομένα. Οι λειτουργίες που πραγματοποιούνται από τα block αυτά είναι μεγάλης σημασίας, και είναι προσβάσιμες από τον χρήστη. Με αυτού του τύπου τα block για παράδειγμα, είναι εφικτός ο χειρισμός του εσωτερικού ρολογιού της κεντρικής μονάδας επεξεργασίας. Βέβαια είναι σημαντικό να υπογραμμίσουμε πως ενώ είναι δυνατή η κλήση των λειτουργιών του συστήματος και των block τους, δεν είναι δυνατή η διαμόρφωσή ή ο προγραμματισμός τους, αφού τα block από μόνα τους δεν βρίσκονται στη μνήμη.

Όποιος και αν είναι όμως ο τύπος του block, συνήθως διαιρείται σε τρία τουλάχιστον μέρη. Το πρώτο μέρος είναι αυτό του αριθμού του block που περιέχει τις ιδιότητές του, έπειτα είναι οι δηλώσεις των τοπικών μεταβλητών, και τέλος το πρόγραμμα, δηλαδή το μέρος, αποθήκευσης των εντολών. Επίσης ανεξάρτητα από τον τύπο του, ένα block θα έχει την επικεφαλίδα του, όπου θα υπάρχουν όλες οι ιδιότητές του, το μέρος των δηλώσεων που περιέχει τις δηλώσεις των τοπικών λειτουργιών του, και το μέρος της αρχικοποίησης, στο οποίο μπορούν να δοθούν αρχικές τιμές στις μεμονωμένες διευθύνσεις δεδομένων.

2.5 ΜΟΡΦΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΤΩΝ PLC

Λαμβάνοντας υπ' όψιν τις παραπάνω πληροφορίες, είναι επιθυμητός ο διαχωρισμός μίας εφαρμογής σε μικρότερα μέρη, ώστε να είναι δυνατή η πιο εύκολη διαμόρφωσή της, αλλά και ο προγραμματισμός της, ακόμη και αν αυτός πραγματοποιείται από διαφορετικά άτομα στην περίπτωση που το μέγεθος του προγράμματος είναι μεγάλο. Επίσης είναι σημαντικό το πλεονέκτημα που παρέχει ο διαχωρισμός ενός προγράμματος σε μικρότερα μέρη, καθώς καθιστά την δοκιμή και την αποσφαλμάτωσή του πολύ ευκολότερη.

Υπάρχουν πολλές μορφές προγραμματισμού, και είναι δυνατή η επιλογή οποιασδήποτε από αυτές, ανάλογα με τις ανάγκες και τις γνώσεις του χρήστη, αλλά και με κριτήριο την καταλληλότητα για την διεκπεραίωση του αυτοματισμού. Αναφορικά οι μορφές αυτές είναι ο γραμμικός, ο μερικός, και ο δομημένος προγραμματισμός.

Στον γραμμικό προγραμματισμό, όλο το κυρίως πρόγραμμα είναι το block οργάνωσης OB1 στο οποίο έχουμε ήδη αναφερθεί, και δεν υπάρχει παραπέρα διαχωρισμός του. Αντίθετα κάθε τρέχον μονοπάτι είναι σε ξεχωριστό δίκτυο, κάτι που βοηθάει στην αποσφαλμάτωση και την διόρθωση, που γίνεται με απευθείας

αναφορά στον αριθμό του. Ουσιαστικά ο γραμμικός προγραμματισμός είναι μία λογική μαθηματική μέθοδος, σύμφωνα με την οποία επιλέγεται ο καλύτερος συνδυασμός συντελεστών και κλάδων, με στόχο την μέγιστη απόδοση του συστήματος.

Εξέλιξη του γραμμικού προγραμματισμού αποτελεί ο μερικός προγραμματισμός, αφού βασίζεται σε αυτόν, με την διαφορά πως το πρόγραμμα διαχωρίζεται σε block. Η τμηματοποίηση του προγράμματος γίνεται είτε γιατί το πρόγραμμα είναι πολύ μεγάλο για να αποτελεί ολόκληρο ένα ΟΒ1, είτε γιατί με αυτόν τον τρόπο είναι πιο εύκολη η ανάγνωσή του. Βέβαια υπάρχει και η δυνατότητα περαιτέρω διαχωρισμού ενός block σε ακόμη μικρότερα τμήματα – δίχως να αλλοιώνεται ταυτόχρονα και η γραμμική φύση του - , γεγονός που μας επιτρέπει να καλούμε συσχετισμένες λειτουργίες της διαδικασίας μέσα από το ίδιο block, αλλά και να το αποσφαλματώνουμε σε μέρη, με την απλή σειριακή κλήση των block.

Τέλος, ο δομημένος ή διαφορετικά γνωστός ως διαδικαστικός προγραμματισμός, βασίζεται στην έννοια της κλήσης διαδικασίας. Η διαδικασία αυτή είναι ένα αυτοτελές σύνολο εντολών προς εκτέλεση. Με την χρήση του δομημένου προγραμματισμού το βασικό πρόβλημα διασπάται σε λιγότερο πολύπλοκα υποπροβλήματα, μέχρι οι εργασίες να είναι αρκετά μικρές, περιεκτικές και ευνόητες. Τα block λειτουργιών αυτά, εξυπηρετούν ένα σκοπό λειτουργίας και για την επίτευξή του ανταλλάσσουν σήματα με άλλα block. Η προγραμματιστική αυτή μορφή βοηθάει στην ευκολότερη συγγραφή πολύπλοκων προγραμμάτων, καθώς και στην ταχύτερη διαχείριση, συντήρηση και αποσφαλμάτωση, καθώς βασίζεται σε μεγαλύτερες και περιεκτικότερες μονάδες, αντί για μεμονωμένες εντολές.

Θα πρέπει εδώ να σημειωθεί πως καθορισμός της σειράς με την οποία θα εκτελεστούν τα block από την κεντρική μονάδα επεξεργασίας γίνεται από το ίδιο το πρόγραμμα. Έτσι, ιδιαίτερη προσοχή θα πρέπει να δίνεται κατά τον ορισμό της σειράς με την οποία γίνονται οι κλήσεις των block, αφού θα πρέπει να είναι ανάλογη με αυτή των επιμέρους λειτουργιών της διαδικασίας που θέλουμε να ελέγξουμε.

2.6 ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΤΩΝ PLC

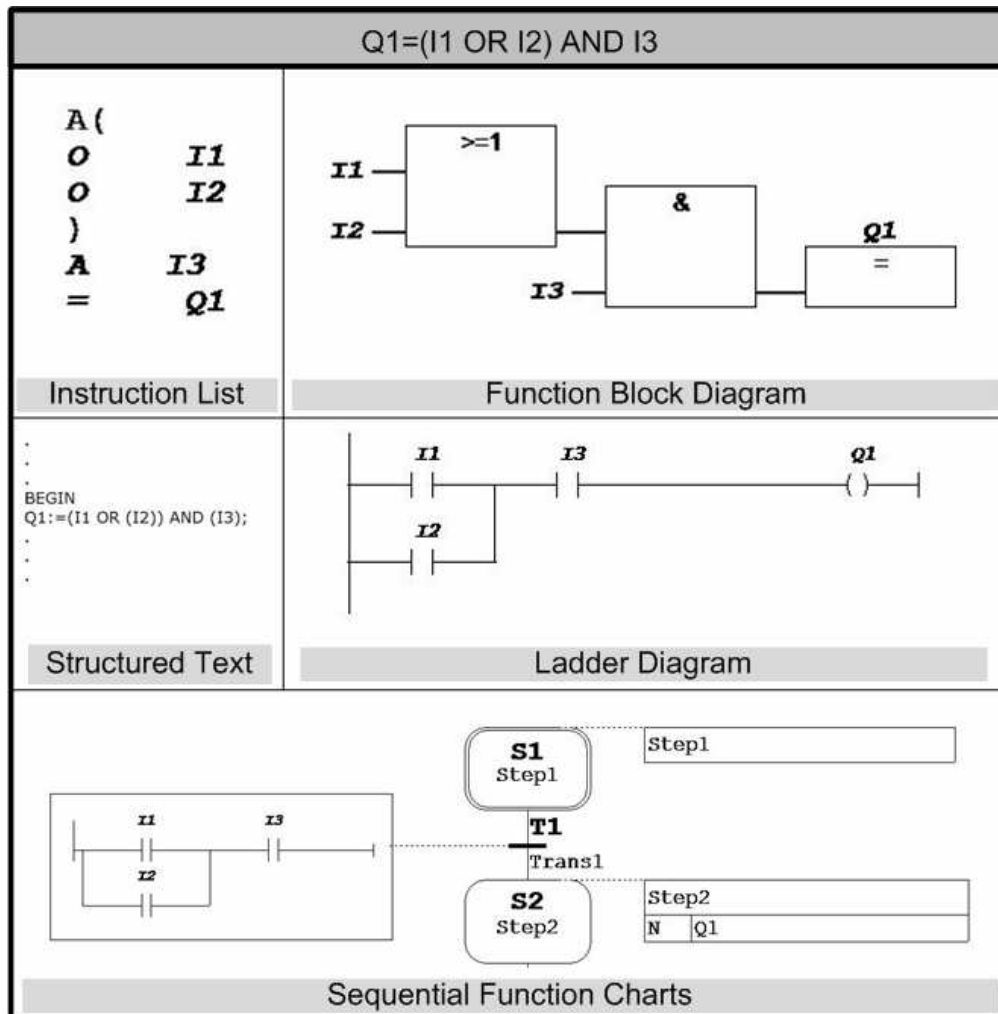
Οι εντολές που χρησιμοποιούμε στην δημιουργία ενός προγράμματος, καθορίζουν την σωστή ή όχι λειτουργία μίας συσκευής. Ο τρόπος που οι εντολές αυτές θα μεταφερθούν στο συγκεκριμένο PLC, λέγεται γλώσσα προγραμματισμού (programming language). Ο όρος αυτός αναφέρεται στην μέθοδο με την οποία ο χρήστης επικοινωνεί με το PLC (Frank D. Petruzella, 1993).

Σύμφωνα με το πρότυπο IEC 61131-3, τα PLC μπορούν να προγραμματιστούν με την χρήση των βασισμένων στα πρότυπα γλωσσών προγραμματισμού. Οι γλώσσες προγραμματισμού μπορούν να ταξινομηθούν σε γραφικές και μη γραφικές ανάλογα με το είδος των στοιχείων που χρησιμοποιούν. Έτσι σε ορισμένους ελεγκτές είναι εφικτός ο γραφικός προγραμματισμός και σε άλλους ο μη γραφικός.

Οι γλώσσες που χρησιμοποιούν γραφικά στοιχεία, είναι πιο προσιτές σε ανθρώπους που έχουν εμπειρία στον κλασικό αυτοματισμό και έχουν το πλεονέκτημα της καλύτερης εποπτείας. Τα γραφικά είναι της μορφής συμβόλων που χρησιμοποιούνται στον κλασικό αυτοματισμό αλλά και λογικών πυλών (AND, OR, NOT κλπ). Οι μη γραφικές γλώσσες, χρησιμοποιούν εντολές που η κάθε μία αντιστοιχεί σε μία εντολή της γλώσσας μηχανής.

Σε πρώτο στάδιο, τα PLC προγραμματιζόταν με τη χρήση διαγραμμάτων λογικής Ladder, ένα πρότυπο που μιμήθηκε τις ηλεκτρομηχανικές συσκευές πινάκων ελέγχου, που τα PLC αντικατέστησαν. Αυτό το πρότυπο παραμένει κοινό μέχρι και σήμερα αν και πλέον εφαρμόζεται το διεθνές πρότυπο IEC 61131-3, το οποίο ορίζει την χρήση πέντε γλωσσών για τον προγραμματισμό ελεγκτών.

Έτσι υπάρχει η **Functional Block Diagram** (λειτουργικό διάγραμμα), την **LaDder** (διάγραμμα επαφών), η **Structured Text** (δομημένο κείμενο, παρόμοιο με τη γλώσσα προγραμματισμού PASCAL), η **Instruction List** (κατάλογος εντολών) και η **Sequential Function Chart** (διάγραμμα διαδοχικής λειτουργίας). Οι γλώσσες αυτές απλώς βοηθούν στην λογική οργάνωση των διαδικασιών.



ΣΧΗΜΑ XIV: ΑΠΕΙΚΟΝΙΣΗ ΛΟΓΙΚΗΣ ΠΡΑΞΗΣ ΣΤΙΣ ΤΥΠΟΠΟΙΗΜΕΝΕΣ ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

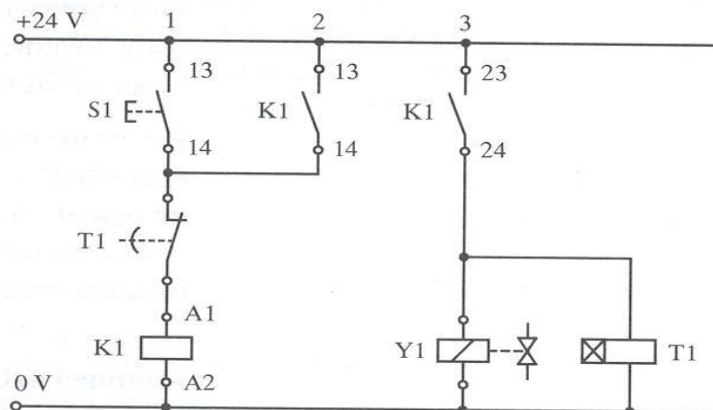
Ο προγραμματισμός PLC γίνεται, συνήθως χρησιμοποιώντας μία ή συνδυάζοντας περισσότερων της μίας από τις παραπάνω γλώσσες, με επικρατέστερες τις IL, FBD και LD. Η επιλογή της γλώσσας προγραμματισμού εξαρτάται από την εμπειρία και την γνώση του χρήστη σε ψηφιακά ηλεκτρονικά, σε υπολογιστές, σε συστήματα αυτοματισμού που λειτουργούν με κλασικό τρόπο και φυσικά από την φύση του προβλήματος που χρήζει αντιμετώπισης.

2.6.1 ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ LADDER (LD)

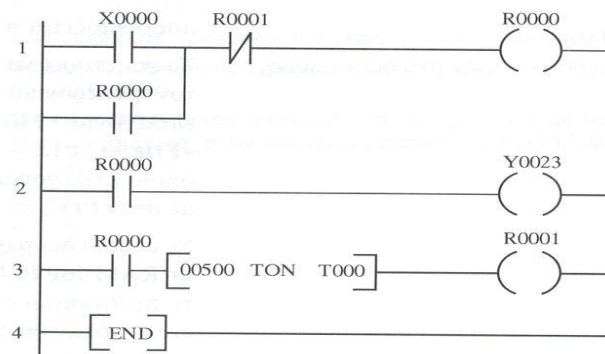
Η Ladder είναι μία εύχρηστη γραφική γλώσσα προγραμματισμού με την βοήθεια της οποίας μπορεί να γίνει απευθείας μετατροπή του ηλεκτρολογικού σχεδίου σε γλώσσα κατανοητή από το PLC. Το όνομά της βασίζεται στην παρατήρηση ότι τα προγράμματα σε αυτή μοιάζουν με σκάλες, αφού περιέχουν δύο κάθετες ράγες, και μία σειρά οριζόντιων βαθμίδων μεταξύ τους.

Ένα επιχείρημα που βοήθησε στην αρχική υιοθέτησή της στην ανάπτυξη λογισμικού για τους προγραμματιζόμενους λογικούς ελεγκτές, ήταν η δυνατότητα της κατανόησης και της χρήσης της από μία μεγάλη ποικιλία μηχανικών και τεχνικών, δίχως την ανάγκη κάποιας περαιτέρω κατάρτισης, και αυτό γιατί μοιάζει έντονα με οικεία σε αυτούς συστήματα υλικού. Η Ladder χρησιμοποιείται στα PLC γιατί σε αυτά απαιτείται ο διαδοχικός έλεγχος της λειτουργίας μίας διαδικασίας ή κατασκευής.

Είναι τρομερά χρήσιμη γλώσσα για απλά αλλά κρίσιμα συστήματα ελέγχου, ή για την επαναλειτουργία παλαιών κυκλωμάτων ηλεκτρονόμων. Με τη χρήση γραφικών εργαλείων (επαφών, πηνίων, καλωδιώσεων, χρονικών κλπ), δομείται ένα λογικό πρόγραμμα, ικανό να ακολουθήσει την λογική συνδεσμολογία ενός κλασικού αυτοματισμού. Οι δυνατότητες βέβαια που παρέχει, είναι πολύ περισσότερες, μιας και εκτελούνται λειτουργίες σύγκρισης, μεταφοράς και μαθηματικής επεξεργασίας δεδομένων.



ΣΧΗΜΑ XV: ΗΛΕΚΤΡΙΚΟ ΚΥΚΛΩΜΑ ΜΕ ΛΟΓΙΚΗ ΣΥΡΜΑΤΩΣΗΣ



ΣΧΗΜΑ XVI: ΗΛΕΚΤΡΙΚΟ ΚΥΚΛΩΜΑ ΜΕ ΠΡΟΓΡΑΜΜΑΤΙΖΟΜΕΝΗ ΛΟΓΙΚΗ

Όπως φαίνεται και από το σχήμα, η κύρια διαφορά μεταξύ της λογικής συρμάτωσης και της προγραμματιζόμενης λογικής είναι ότι όλοι οι είσοδοι εισάγονται με τη μορφή συμβολικών επαφών και όλοι οι έξοδοι εισάγονται με τη μορφή συμβολικών πηνίων.

Ένα πρόγραμμα γραμμένο σε Ladder αποτελείται από rungs, δηλαδή ένα σύνολο από γραφικές εντολές, οι οποίες είναι τοποθετημένες μεταξύ δύο κάθετων γραμμών, που αντιπροσωπεύουν η μεν αριστερή τη γραμμή τροφοδοσίας, η δε δεξιά την γραμμή επιστροφής. Αν ένα μονοπάτι είναι εφικτό να ακολουθηθεί από την αριστερή στη δεξιά πλευρά του rung μέσω συγκεκριμένων ανοικτών ή κλειστών επαφών, τότε το rung είναι αληθές, και το bit αποθήκευσης του πηνίου εξόδου λαμβάνει την τιμή 1. Στην αντίθετη περίπτωση όπου κανένα μονοπάτι δεν γίνεται να ακολουθηθεί, το αποτέλεσμα είναι 0, και το πηνίο θεωρείται απενεργοποιημένο.

Οι διάφορες γραφικές εντολές που υπάρχουν σε ένα rung παριστάνουν:

- εισόδους και εξόδους του PLC (διακόπτες, κουμπιά, αισθητήρια, κλπ)
- λειτουργίες του PLC (χρονικά, μετρητές, κλπ)
- μαθηματικές και λογικές πράξεις (πρόσθεση, αφαίρεση, κλπ)
- πράξεις συγκρίσεως και αριθμητικές λειτουργίες ($A < B$, $A = B$, κλπ)
- εσωτερικές μεταβλητές του PLC (bits, words, κλπ)

Αυτά τα γραφικά εργαλεία συνδέονται με οριζόντιες και κάθετες γραμμές για να οδηγηθούν τελικά σε μία ή περισσότερες εξόδους ή και σε στοιχεία που εκτελούν διαφορετικές λειτουργίες. **Προσοχή**, ένα rung δεν μπορεί να υποστηρίξει περισσότερες της μίας ομάδας εντολών συνδεδεμένων μεταξύ τους.

Κάθε rung περιέχει επτά γραμμές και έντεκα στήλες και αποτελείται από δύο αλληλεπικαλυπτόμενες περιοχές, την ζώνη ελέγχου (test zone) που περιλαμβάνει τις συνθήκες οι οποίες πρέπει να αληθεύουν για να λάβει χώρα μια ενέργεια, και το ενεργό μέρος (action zone) που περιλαμβάνει την ενέργεια, η οποία μπορεί να είναι η ενεργοποίηση μιας εξόδου ή η πραγματοποίηση μιας λογικής πράξης (λογικής ή αριθμητικής).

Ως βασικά γραφικά στοιχεία θεωρούνται οι επαφές. Οι επαφές διαχωρίζονται στις ανοικτές ($-| |-$), στις κλειστές ($-|/|-$), στις ανερχόμενου ($-|P|-$) και κατερχόμενου ($-|N|-$) παλμού. Όλες οι επαφές μπορούν να τοποθετηθούν οπουδήποτε μέσα στο πλέγμα προγραμματισμού της test zone, εκτός της τελευταίας στήλης, δηλαδή της action zone. Οι ανοικτές και κλειστές επαφές αντιπροσωπεύουν τις πραγματικές εισόδους του PLC, καθώς και τα εσωτερικά bits. Η μόνη διαφορά μεταξύ των δύο τύπων έγκειται στο πότε η επαφή θεωρείται κλειστή και πότε ανοικτή. Στις ανοικτές επαφές η επαφή είναι ανοικτή όταν είναι

σε κατάσταση λογικού '1', και κλειστή όταν είναι σε κατάσταση λογικού '0', ενώ στις κλειστές επαφές ισχύει το αντίθετο. Οι επαφές ανερχόμενου και κατερχόμενου παλμού βρίσκονται σε λογική κατάσταση '1' την ώρα που απενεργοποιείται μία είσοδος του PLC, και παραμένουν στην κατάσταση αυτή όσο διαρκεί ένας κύκλος λειτουργίας του PLC, αλλά η μεν ανερχόμενου παλμού επαφή είναι κλειστή στην κατάσταση λογικού '0' και ανοικτή στην κατάσταση λογικού '1', η δε κατερχόμενου παλμού το αντίθετο.

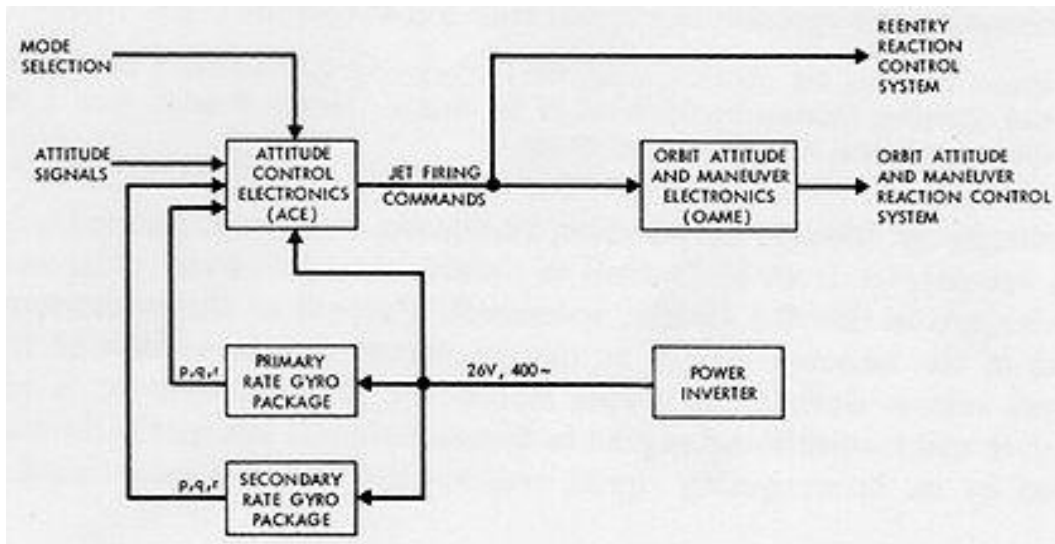
Στα βασικά γραφικά στοιχεία συγκαταλέγονται και τα πηνία, είτε αυτά είναι τα κανονικά (-()-), είτε τα ανάστροφα (-(/)-), είτε τα αυτομανδάλωσης (-(S)-), είτε τα απομανδάλωσης (-R-). Τα πηνία μπορούν να τοποθετηθούν μόνο στην τελευταία στήλη του πλέγματος προγραμματισμού, δηλαδή της action zone, ανεξάρτητα από το είδος τους και αντιπροσωπεύουν τις πραγματικές εξόδους του PLC, καθώς και τα εσωτερικά bits. Τα πηνία αυτομανδάλωσης και απομανδάλωσης, δεν είναι οπλισμένα πριν πάρουν τάση στα άκρα τους, αλλά μόλις εφαρμοστεί τάση στα άκρα τους οπλίζουν και παραμένουν οπλισμένα, ανεξάρτητα από το αν συνεχίζουν να τροφοδοτούνται ή όχι με τάση. Όσον αφορά τα πηνία αυτομανδάλωσης και απομανδάλωσης, αυτά χαρακτηρίζονται από την μοναδική ιδιότητά του πηνίου απομανδάλωσης να παροπλίζει ένα πηνίο αυτομανδάλωσης. Τα κανονικά και τα ανάστροφα πηνία, διαφέρουν στο πότε είναι οπλισμένα και πότε όχι. Έτσι ένα κανονικό πηνίο είναι οπλισμένο όταν βρίσκεται σε κατάσταση λογικού '0', και δεν είναι οπλισμένο όταν βρίσκεται σε κατάσταση λογικού '1'. Το αντίθετο ισχύει για τα ανάστροφα πηνία.

Η γλώσσα προγραμματισμού Ladder είναι καταλληλότερη για τον έλεγχο προβλημάτων στα οποία απαιτούνται μόνο δυαδικές μεταβλητές, και όπου η διαδοχική τοποθέτηση δυαδικών είναι το πρωταρχικό πρόβλημα ελέγχου.

2.6.2 ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ FUNCTION BLOCK DIAGRAM (FBD)

Η γλώσσα προγραμματισμού με διάγραμμα λειτουργίας, είναι μία γραφική γλώσσα συσχέτισης των εξόδων με τις εισόδους, κάνοντας χρήση διαφόρων τύπων block.

Με άλλα λόγια, σε ένα διάγραμμα λειτουργίας περιγράφεται μία σχέση μεταξύ των μεταβλητών εισόδου και των μεταβλητών εξόδου, και απεικονίζεται από ένα σύνολο στοιχειωδών block. Οι μεταβλητές εισόδου και εξόδου, συνδέονται με block μέσω των γραμμών σύνδεσης. Έτσι η έξοδος ενός block μπορεί να αποτελεί την είσοδο ενός άλλου block.



ΣΧΗΜΑ XVII: ΠΑΡΑΔΕΙΓΜΑ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΕ ΓΛΩΣΣΑ FBD

Οι εισόδοι και οι έξοδοι των block χρησιμοποιούν γραμμές σύνδεσης ή συνδέσμους για την ένωσή τους. Οι απλές γραμμές μπορούν να χρησιμοποιηθούν για τη σύνδεση δύο λογικών σημείων του διαγράμματος. Ως λογικά σημεία μπορούν να θεωρηθούν μία μεταβλητή εισόδου και η είσοδος ενός block, η έξοδος ενός block και η είσοδος ενός άλλου block, καθώς και η έξοδος ενός block και μία μεταβλητή εξόδου. Η σύνδεση αυτή είναι προσανατολισμένη, που σημαίνει πως η γραμμή μεταφέρει σχετικά στοιχεία από το αριστερό τμήμα στο δεξί. Τα τμήματα, αριστερό και δεξί της γραμμής σύνδεσης πρέπει να είναι του ίδιου τύπου.

Υπάρχει η δυνατότητα πολλαπλής σχέσης, και η οποία διαφορετικά ονομάζεται απόκλιση. Η απόκλιση μπορεί να χρησιμοποιηθεί για την μετάδοση των πληροφοριών από τα αριστερά σε κάθε ένα από τα δεξιά τμήματα. Και πάλι όλα τα τμήματα θα πρέπει να είναι του ίδιου τύπου.

Δεδομένου ότι οι μικροεπεξεργαστές έχουν γίνει ισχυρότεροι, γλώσσες όπως είναι τα διαγράμματα διαδοχικής λειτουργίας, μπορούν να αντικαταστήσουν την γλώσσα Ladder για κάποιες περιορισμένες εφαρμογές, ή και να συνδυαστούν με αυτή για την επίλυση ενός προβλήματος και την ανάπτυξη ενός προγράμματος.

2.6.3 ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ SEQUENTIAL FUNCTION CHART (SFC)

Η ύπαρξη πολύπλοκων μηχανών, οι οποίες έχουν σχεδιαστεί για την εκτέλεση ταυτόχρονων λειτουργιών, απαιτεί την ύπαρξη ενός ελεγκτή που να είναι ικανός για παράλληλη επεξεργασία περισσότερων της μίας κατάστασης

κάθε στιγμή. Αυτό είναι δυνατόν να επιτευχθεί με τη χρήση διαγραμμάτων διαδοχικής λειτουργίας.

Το διάγραμμα διαδοχικής λειτουργίας είναι μια γραφική τεχνική για την δημιουργία προγραμμάτων παράλληλου ελέγχου. Είναι δυνατόν να χρησιμοποιηθεί για τον προγραμματισμό λειτουργιών που μπορούν να διαχωριστούν σε βήματα. Τα κύρια συστατικά στοιχεία ενός SFC είναι τα βήματα με σχετικές ενέργειες, οι μεταβάσεις με σχετικούς όρους λογικής, και οι κατευθυνόμενες συνδέσεις μεταξύ των βημάτων και των μεταβάσεων.

Τα βήματα μπορούν να είναι ενεργά ή ανενεργά. Οι ενέργειες που σχετίζονται με αυτά εκτελούνται μόνο για ενεργά βήματα. Ένα βήμα μπορεί να είναι ενεργό είτε γιατί είναι το αρχικό βήμα όπως έχει ορισθεί από τον προγραμματιστή, είτε γιατί είχε ενεργοποιηθεί κατά τη διάρκεια ενός κύκλου σάρωσης και δεν απενεργοποιήθηκε έκτοτε.

Συνέχεια των βασικών στοιχείων ενός SFC αποτελούν οι γραμμές ροής οι οποίες συνδέουν τα βήματα με τις μεταβάσεις, οι μεταβάσεις που λειτουργούν ως σημείο συγχρονισμού και επιτρέπουν στον έλεγχο να μεταφερθεί στο επόμενο σημείο όταν υπάρξουν οι κατάλληλες συνθήκες, τα βήματα που ουσιαστικά αποτελούν μία κατάσταση λειτουργίας, και οι διακλαδώσεις στις οποίες ανάλογα με το είδος τους είναι δυνατόν να ακολουθηθεί ένας ή παραπάνω δρόμοι.

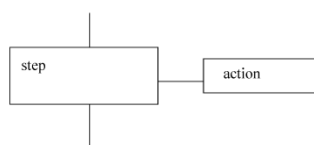
Οι συμβολισμοί των στοιχείων αυτών παρατίθενται στα σχήματα που ακολουθούν.



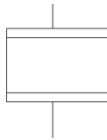
: Μετάβαση.



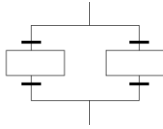
: Αρχικό βήμα.



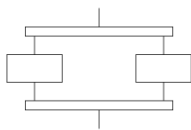
: Βήμα (Συνήθως συνοδεύεται από ενέργεια).



: Υπορουτίνα (Σύνολο βημάτων).



: Διακλάδωση στην οποία μόνο ένα μονοπάτι είναι δυνατόν να ακολουθηθεί.



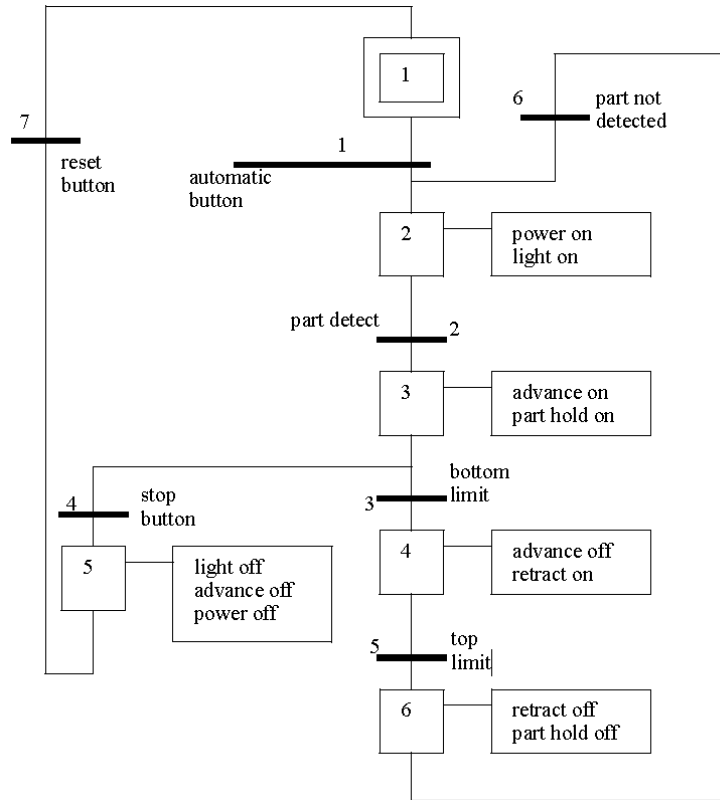
: Διακλάδωση στην οποία μπορούν όλα τα μονοπάτια να ακολουθηθούν.

ΣΧΗΜΑ ΧVΙΙΙ: ΣΥΜΒΟΛΙΣΜΟΙ ΣΤΟΙΧΕΙΩΝ ΓΛΩΣΣΑΣ SFC

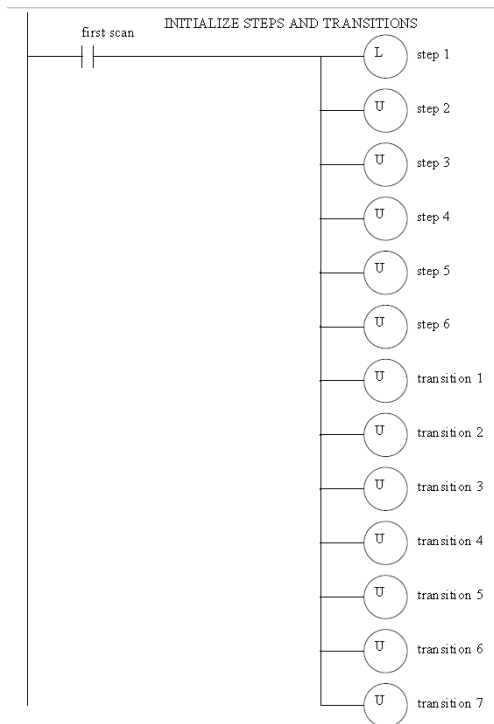
Τα βήματα ενεργοποιούνται όταν όλα τα προηγούμενα από αυτά είναι ενεργά και οι συνδέσεις μετάβασης είναι αληθείς. Με την σάρωση μίας μετάβασης, όλα τα βήματα πάνω από αυτή απενεργοποιούνται αμέσως, και στη συνέχεια ενεργοποιούνται άμεσα όλα τα βήματα κάτω από αυτήν.

Οι σχετικές με βήματα ενέργειες μπορούν να είναι διαφόρων τύπων, με τις πιο γνωστές να είναι τύπου Continuous (N), Set (S) και Reset (R). Πέρα από το εμφανές νόημα των Set και Reset, μία ενέργεια τύπου N διασφαλίζει ότι η μεταβλητή - στόχος είναι ορισμένη στο '1' με την προϋπόθεση πως το βήμα είναι ενεργό.

Δεν είναι απλά δυνατόν να παρεμβληθούν ενέργειες της γλώσσα Ladder μέσα σε ένα πρόγραμμα SFC, αλλά αποτελεί και τον πιο συνηθισμένο τρόπο προγραμματισμού, ειδικά για την επεξεργασία ακέραιων μεταβλητών.



ΣΧΗΜΑ XIX: ΠΑΡΑΔΕΙΓΜΑ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΕ ΓΛΩΣΣΑ SFC



ΣΧΗΜΑ XX: ΕΜΠΛΕΚΟΝΤΑΣ ΤΗΝ SFC ΣΤΗΝ LADDER (ΣΤΟ ΙΔΙΟ ΠΑΡΑΔΕΙΓΜΑ)

2.6.4 ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ STRUCTURED TEXT (ST)

Η γλώσσα δομημένου κειμένου είναι μία υψηλού επιπέδου γλώσσα που είναι κατασκευασμένη με block, και συντακτικά μοιάζει με την γλώσσα προγραμματισμού Pascal. Ακόμη και το μικρότερο πρόγραμμα σε ST χρησιμοποιεί χαρακτηριστικά γλωσσών υψηλού επιπέδου, συμπεριλαμβανομένων των μεταβλητών και πολλών δηλώσεων ελέγχου ροής.

Στην ST οι μεταβλητές χρησιμοποιούνται για την διαχείριση δεδομένων, οι χειριστές (operators) είναι αριθμητικοί ή ανάθεσης, οι εκφράσεις αποτελούν ένα συνδυασμό από operators και μεταβλητές σε ακολουθίες, ενώ τέλος ο έλεγχος ροής χρησιμοποιείται για την υπό συνθήκη εκτέλεση εντολών, την επανάληψη τους και τη μεταπήδηση σε μία άλλη περιοχή του προγράμματος.

Η ονομασία Structured αναφέρεται στις υψηλού επιπέδου δυνατότητες που προσφέρει η γλώσσα, ενώ το Text αφορά τη δυνατότητα για την χρήση κειμένου σε αντικατάσταση των συμβόλων που χρησιμοποιούνται στην LD, στην FBD, και στην SFC.

Η ST όντας μία υψηλού επιπέδου γλώσσα, υποστηρίζει πολύπλοκες εντολές και εμφωλευμένες οδηγίες, όπως είναι οι βρόχοι επανάληψης (WHILE - DO), η υπό συνθήκη εκτέλεση (IF – THEN - ELSE), και οι συναρτήσεις (SIN()).

Από τη στιγμή που η ST ανήκει στις διεθνώς τυποποιημένες γλώσσες προγραμματισμού, και αυτή όπως και οι υπόλοιπες μοιράζονται τα κοινά στοιχεία του IEC61131, πετυχαίνοντας με τον τρόπο αυτό την δυνατότητα χρήσης του ίδιου προγράμματος από διαφορετικές γλώσσες. Η ύπαρξη της γλώσσας ST, δεν υφίσταται για την αντικατάσταση των υπολοίπων γλωσσών, αφού όλες τους έχουν τα πλεονεκτήματα και τα μειονεκτήματά τους και το δικό της μεγαλύτερο πλεονέκτημα είναι η ικανότητά της να απλοποιεί πολύπλοκες μαθηματικές ισότητες.

```
(* Generate Text for Heater *)
IF Heater THEN
  TEXT:=CONCAT(TEXT,' Heater: ON ');
ELSE
  TEXT:=CONCAT(TEXT,' Heater: OFF ');
END_IF;

(* Generate Text for Drain *)
IF Drain THEN
  TEXT:=CONCAT(TEXT,'Drain: OPEN ');
ELSE
  TEXT:=CONCAT(TEXT,'Drain: CLOSE ');
END_IF;

Level_Text := INT_TO_STRING (Level,'%ld');
Temp_Text  := INT_TO_STRING (Temp,'%ld');

Text_Value := CONCAT (TEXT_2,'Level: ');
Text_Value := CONCAT (Text_Value,Level_Text);
Text_Value := CONCAT (Text_Value,' Liter');
Text_Value := CONCAT (Text_value,' Temperature: ');
```

ΕΙΚΟΝΑ VII: ΠΑΡΑΔΕΙΓΜΑ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΕ ΓΛΩΣΣΑ ST

2.6.5 ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ INSTRUCTION LIST (IL)

Η γλώσσα καταλόγου οδηγιών είναι μία χαμηλού επιπέδου γλώσσα και μοιάζει με την γλώσσα προγραμματισμού Assembly. Είναι εφικτή η μετατροπή οποιασδήποτε γλώσσας προγραμματισμού PLC σε προγράμματα τύπου IL. Η IL χρησιμοποιεί πολύ απλές εντολές πανομοιότυπες με τις αρχικές μνημονικές γλώσσες προγραμματισμού που είχαν αναπτυχθεί για τα PLC.

Μια βασική έννοια αυτής της γλώσσας, αποτελεί ο σωρός. Ο σωρός μπορεί να θεωρηθεί ως μία λίστα που πρόκειται να χρησιμοποιηθεί αργότερα, με άλλα λόγια χρησιμοποιείται για την καθυστέρηση των λειτουργιών οι οποίες υποδεικνύονται από τις υπάρχουσες παρενθέσεις.

Program File 2:

Label	Opcode	Operand	Comment
START:	CAL	3	(* Jump to program file 3 *)

Program File 3:

Label	Opcode	Operand	Comment
TEST:	LD	%I:000	(* Load the word from input card 000 *)
	BCD_TO_INT		(* Convert the BCD value to an integer *)
	ST	%N7:0	(* Store the value in N7:0 *)
	GT	100	(* Check for the stored value (N7:0) > 100 *)
	JMPC	ON	(* If true jump to ON *)
ON:	CAL	RES(C5:0)	(* Reset the timer *)
	LD	2	(* Load a value of 2 - for the preset *)
	ST	%C5:0.PR	(* Store 2 in the preset value *)
	CAL	TON(C5:0)	(* Update the timer *)
	LD	%C5:0.DN	(* Get the timer done condition bit *)
	ST	%O:001/00	(* Set the output bit *)
	RET		(* Return from the subroutine *)

ΕΙΚΟΝΑ VIII: ΠΑΡΑΔΕΙΓΜΑ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΕ ΓΛΩΣΣΑ IL

ΚΕΦΑΛΑΙΟ 3

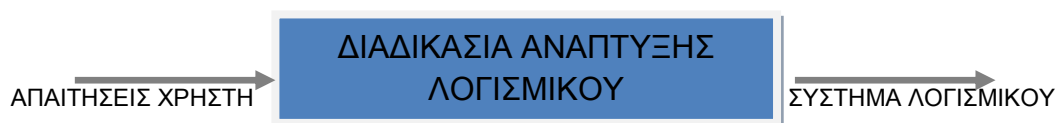
ΑΝΑΠΤΥΞΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ

Στόχος αυτού του κεφαλαίου αποτελεί η περιγραφή της μεθόδου που θα ακολουθηθεί για την ανάπτυξη του απαιτούμενου λογισμικού, ενώ στη συνέχεια θα γίνει και μοντελοποίηση της ανάπτυξης αυτής. Με τον τρόπο αυτό θα απλοποιηθεί και θα γίνει πιο κατανοητή η διαδικασία ανάπτυξης οποιουδήποτε λογισμικού. Το σημαντικότερο τμήμα όμως αυτού του κεφαλαίου αποτελεί η εφαρμογή των γνώσεων αυτών στο σύστημα βιομηχανικού πλυντηρίου.

3.1 ΛΟΓΙΚΑ ΕΝΟΠΟΙΗΜΕΝΗ ΔΙΕΡΓΑΣΙΑ RUP

Με την πάροδο του χρόνου τα υπολογιστικά συστήματα οποιασδήποτε μορφής και αν είναι αυτά ισχυροποιούνται, με αποτέλεσμα την αύξηση των απαιτήσεων από πλευράς χρηστών για μεγαλύτερες επιδόσεις από το υλικό, αλλά κυρίως από το λογισμικό του συστήματος. Συνέπεια αυτής της ολοένα και αυξανόμενης ανάγκης ήταν η παραγωγή τρομερά πολύπλοκου λογισμικού σε μικρό χρονικό διάστημα. Για την επίτευξη αυτού του είδους της παραγωγής εντός των χρονικών περιθωρίων απαραίτητη προϋπόθεση αποτελούσε η εξέλιξη της μεθόδου ανάπτυξης λογισμικού. Η λύση δόθηκε με την ενοποιημένη διεργασία (**Unified Process**), και ειδικά με την μορφή της λογικά ενοποιημένης διεργασίας (**Rational Unified Process**).

Η λογικά ενοποιημένη διεργασία είναι μία μέθοδος ανάπτυξης λογισμικού, ένα σύνολο ενεργειών δηλαδή, που είναι απαραίτητες για την μετατροπή των απαιτήσεων του χρήστη σε ένα σύστημα λογισμικού. Η RUP δεν αποτελεί μία συγκεκριμένη διαδικασία καθοδήγησης, αλλά ένα προσαρμόσιμο επαναληπτικό διαδικαστικό πλαίσιο, το οποίο χρησιμοποιείται από τις ομάδες ανάπτυξης λογισμικού, σύμφωνα με τις ανάγκες τους. Έτσι είναι δυνατόν να επιλεχθούν απλώς ορισμένα τμήματα της διαδικασίας για την εκπόνηση μίας εργασίας.

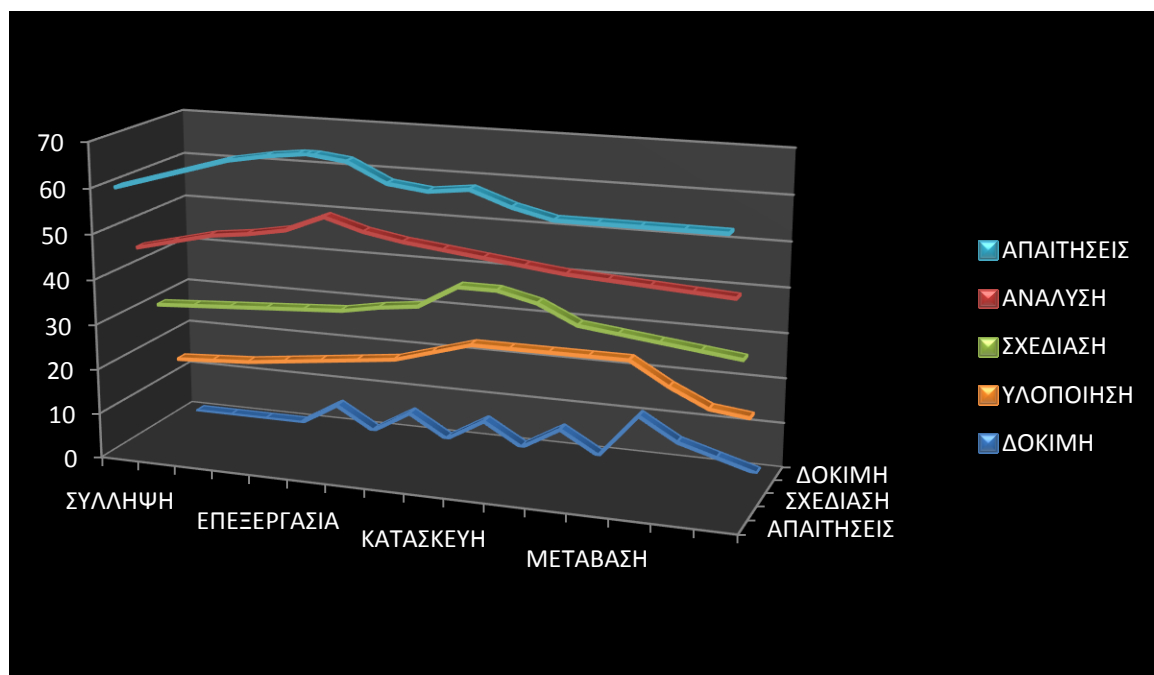


ΣΧΗΜΑ ΧΧΙ: ΔΙΑΔΙΚΑΣΙΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

Η RUP βασίζεται σε ένα σύνολο δομικών ή συστατικών στοιχείων, τα οποία περιγράφουν το προϊόν που θα παραχθεί, τις απαραίτητες δεξιότητες και τον τρόπο επίτευξης των συγκεκριμένων στόχων ανάπτυξης. Τα συστατικά αυτά στοιχεία είναι οι ρόλοι, τα προϊόντα εργασίας και οι στόχοι, που δίνουν απάντηση στα ερωτήματα που, τι και πώς αντίστοιχα. Ένας ρόλος ορίζει ένα σύνολο από σχετικές δεξιότητες, ειδικότητες και ευθύνες. Ο στόχος περιγράφει μία εργασία που έχει ανατεθεί σε έναν ρόλο και παρέχει ένα λογικό αποτέλεσμα. Το προϊόν εργασίας αντιπροσωπεύει το αποτέλεσμα ενός στόχου, συμπεριλαμβανομένων όλων των εγγράφων και των μοντέλων που παράγονται.

Ο κύκλος ζωής ενός έργου, σύμφωνα με την RUP αποτελείται από τέσσερις φάσεις (Σύλληψη, Επεξεργασία, Κατασκευή και Μετάβαση). Κάθε φάση έχει έναν αντικειμενικό στόχο και ένα ορόσημο στο τέλος της, το οποίο δηλώνει την επίτευξη του στόχου αυτού. Στην πραγματικότητα, το κλειδί της διεργασίας έγκειται στις επαναλήψεις της ανάπτυξης που υπάρχουν σε όλες τις φάσεις.

Σε κάθε επανάληψή της, οι στόχοι διακρίνονται σε πέντε βασικές κατηγορίες, που είναι οι Απαιτήσεις, η Ανάλυση, η Σχεδίαση, η Υλοποίηση, και η Δοκιμή. Υπάρχουν βέβαια και τρεις ακόμη που αποτελούν τις κατηγορίες ενίσχυσης (Διαμόρφωση και Διαχείριση αλλαγής, Διαχείριση Έργου, Περιβάλλον), και οι οποίες είναι προαιρετικές.



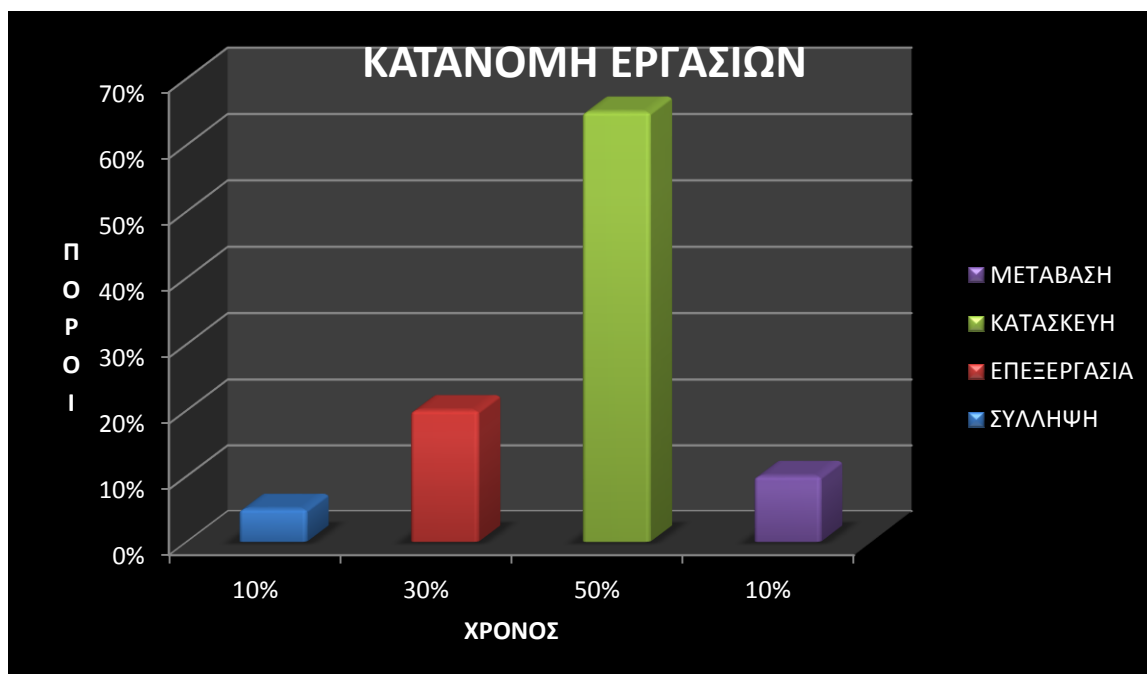
ΣΧΗΜΑ XXII: ΦΑΣΕΙΣ ΚΑΙ ΕΡΓΑΣΙΕΣ ΛΟΓΙΚΑ ΕΝΟΠΟΙΗΜΕΝΗΣ ΔΙΕΡΓΑΣΙΑΣ

3.1.1 ΚΥΚΛΟΣ ΖΩΗΣ ΛΟΓΙΚΑ ΕΝΟΠΟΙΗΜΕΝΗΣ ΔΙΕΡΓΑΣΙΑΣ

Η λογικά ενοποιημένη διεργασία επαναλαμβάνει μία σειρά από κύκλους εργασίας που σταδιακά ολοκληρώνουν το σύστημα. Κάθε κύκλος ολοκληρώνεται με μία νέα έκδοση του συστήματος και κάθε έκδοση είναι ένα προϊόν έτοιμο για παράδοση.

Το προϊόν της τελευταίας επανάληψης θα πρέπει να αποτελεί κάτι περισσότερο από εκτελέσιμο κώδικα. Το τελικό αυτό προϊόν περιλαμβάνει τις απαιτήσεις, τις περιπτώσεις χρήσης, τις μη λειτουργικές απαιτήσεις, τις περιπτώσεις δοκιμής, την αρχιτεκτονική και τα οπτικά μοντέλα. Όλα αυτά τα στοιχεία είναι απαραίτητα για τον καθορισμό, τον σχεδιασμό, την υλοποίηση και την δοκιμή του συστήματος, αλλά και την εξέλιξη αυτού όταν παραστεί ανάγκη.

Η διάρκεια ενός κύκλου εργασιών είναι μία χρονική περίοδος, που διακρίνεται σε τέσσερις φάσεις όπως έχει ήδη αναφερθεί. Κάθε φάση ολοκληρώνεται σε σημεία ορόσημα για την ανάπτυξη του συστήματος, τα οποία εξυπηρετούν διάφορους σκοπούς. Ο σημαντικότερος λόγος είναι πως ο επικεφαλής του έργου είναι υποχρεωμένος να λάβει κρίσιμες για την έκβαση του έργου αποφάσεις πριν την έναρξη της επόμενης φάσης. Συγχρόνως η επίβλεψη του έργου διευκολύνεται, και η πρόοδος που σημειώνεται είναι ορατή καθώς το έργο φτάνει στα τέσσερα αυτά ορόσημα.



ΣΧΗΜΑ XXIII: ΠΟΣΟΣΤΙΑΙΟΣ ΚΑΤΑΜΕΡΙΣΜΟΣ ΕΡΓΑΣΙΩΝ ΚΑΤΑ ΤΗ ΔΙΑΡΚΕΙΑ ΕΝΟΣ ΤΥΠΙΚΟΥ ΕΡΓΟΥ

ΦΑΣΗ ΣΥΛΛΗΨΗΣ

Αρχικό στόχο της φάσης αυτής αποτελεί η επαρκής περιγραφή του συστήματος ως βάση για την αξιολόγηση του αρχικού κόστους και του προϋπολογισμού. Ενώ ορίζεται η επιχειρησιακή υπόθεση, η οποία περιλαμβάνει το επιχειρησιακό περιεχόμενο, τους παράγοντες επιτυχίας και την οικονομική πρόβλεψη.

Η φάση της σύλληψης δεν αποτελεί μία ολοκληρωμένη μελέτη του συστήματος, και είναι απαραίτητη η οριοθέτηση του συστήματος και η αναγνώριση των διασυνδέσεων του με εξωτερικά συστήματα. Επίσης θα πρέπει να περιγραφεί μια υποψήφια αρχιτεκτονική του συστήματος, με στόχο την δημιουργία της αίσθησης πως η κατασκευή μίας σταθερής αρχιτεκτονικής είναι εφικτή, δίχως όμως σε αυτό το σημείο να πραγματοποιείται η κατασκευή αυτή. Συγχρόνως αναγνωρίζονται οι κίνδυνοι που επηρεάζουν την κατασκευή του συστήματος, και κρίνεται αν είναι δυνατή η αντιμετώπισή τους. Τελικά δημιουργείται ένα πρωτότυπο για την παρουσίαση των βασικών ιδεών του νέου συστήματος, και το οποίο καταστρέφεται μετά την ολοκλήρωση του τελικού προϊόντος.

Οι εργασίες της φάσης αυτής δεν ξεπερνούν το στάδιο της απόδειξης της οικονομικής ωφέλειας από την ανάπτυξη του συστήματος. Έτσι ουσιαστικά απλώς αποδεικνύεται πως η κατασκευή του συστήματος προβλέπεται να είναι περισσότερο επωφελής από την επένδυση που απαιτείται για αυτή.

Απώτερος σκοπός της φάσης της σύλληψης αποτελεί η ελαχιστοποίηση των δαπανών χρόνου, προσπάθειας και οικονομικών πόρων μέχρι την διαβεβαίωση της λειτουργικότητας του συστήματος. Η χρονική διάρκεια για την διαβεβαίωση αυτή μπορεί - αν το σύστημα είναι ευμέγεθες και απαιτηθούν πολλές επαναλήψεις - να είναι μεγάλη.

Για την ολοκλήρωση της σύλληψης απαιτείται η δημιουργία ενός βασικού προτύπου περιπτώσεων χρήσης, ενός σχεδίου προγράμματος, μίας αρχικής εκτίμησης των κινδύνων και μίας περιγραφής του έργου. Με την ολοκλήρωση των παραπάνω, γίνεται έλεγχος από τον οποίο προκύπτει αν το έργο έχει ξεπεράσει το ορόσημο της φάσης.

Αν το έργο δεν ξεπεράσει το ορόσημο αυτό, το οποίο ονομάζεται Αντικειμενικό Ορόσημο Κύκλου Ζωής, είναι δυνατή είτε η ακύρωσή του, είτε η επανάληψή του μετά τον επανασχεδιασμό του, για την ικανοποίηση των κριτηρίων.

ΦΑΣΗ ΕΠΕΞΕΡΓΑΣΙΑΣ

Ο αρχικός στόχος αυτής της φάσης, είναι ο μετριασμός των βασικών κινδύνων, οι οποίοι εντοπίστηκαν προηγουμένως. Στην φάση της επεξεργασίας το πρόγραμμα παίρνει μορφή, αφού γίνεται η ανάλυση των περιοχών του προβλήματος και η μορφοποίηση της αρχιτεκτονικής του έργου.

Το πρωτεύον προϊόν της επεξεργασίας είναι μία σταθερή αρχιτεκτονική, ικανή για την καθοδήγηση του συστήματος σε όλη τη διάρκειά της ζωής του. Το τελικό σημείο της φάσης αυτής αποτελεί ο λεπτομερειακός προγραμματισμός της επόμενης, δηλαδή, της κατασκευής.

Κατά την επεξεργασία, δημιουργείται μία αρχική αρχιτεκτονική που καλύπτει τις σημαντικές λειτουργίες και τα χαρακτηριστικά του συστήματος. Επίσης αναθεωρούνται οι κίνδυνοι που είχαν αναγνωριστεί προηγουμένως, καθώς και οι τρόποι αντιμετώπισής τους. Συγχρόνως καθορίζεται το ποιοτικό επίπεδο των χαρακτηριστικών του συστήματος, και συντάσσεται ένα πλάνο για τους απαιτούμενους πόρους, είτε αυτοί είναι ανθρώπινοι, είτε οικονομικοί. Τέλος δημιουργείται ένα πρότυπο περιπτώσεων χρήσης, στο οποίο οι περιπτώσεις χρήσεις και οι ρόλοι αναγνωρίζονται ενώ συγχρόνως αναπτύσσονται οι περισσότερες περιγραφές των περιπτώσεων χρήσης. Με το τέλος της φάσης, θα πρέπει να έχει ολοκληρωθεί κατά 80% το πρότυπο των περιπτώσεων χρήσης.

Αν το έργο δεν ξεπεράσει το ορόσημο αυτό, το οποίο ονομάζεται Αρχιτεκτονικό Ορόσημο Κύκλου Ζωής, υπάρχει ακόμη χρόνος για την ακύρωσή του ή τον επανασχεδιασμό του. Μετά την έξοδο από την φάση αυτή όμως, το έργο μεταβαίνει σε μία λειτουργία υψηλού κινδύνου, όπου οι αλλαγές είναι δυσκολότερες και καταστρεπτικές όταν γίνονται.

ΦΑΣΗ ΚΑΤΑΣΚΕΥΗΣ

Ο πρωταρχικός στόχος της φάσης αυτής είναι η δημιουργία του λογισμικού συστήματος. Η προσοχή εστιάζεται στην ανάπτυξη των συστατικών και των άλλων χαρακτηριστικών γνωρισμάτων του συστήματος. Στην φάση αυτή υλοποιείται ο μεγαλύτερος όγκος της κωδικοποίησης.

Η κατασκευή απασχολεί το περισσότερο ανθρώπινο δυναμικό για τη μεγαλύτερη χρονική περίοδο σε σχέση με τις υπόλοιπες φάσεις του κύκλου ζωής, γι' αυτό είναι απαραίτητο να έχουν ξεκαθαριστεί όλα τα σημαντικά σημεία πριν από την είσοδο σε αυτήν. Απαιτείται μεγαλύτερος αριθμός επαναλήψεων για την ολοκλήρωση της φάσης αυτής, σε σύγκριση με τις προηγούμενες.

Στην φάση της κατασκευής περιλαμβάνεται η επέκταση των περιπτώσεων χρήσης, ως το σημείο εξάντλησης όλων των δυνατών περιπτώσεων χρήσης του συστήματος. Επίσης διεκπεραιώνονται οι εργασίες της ανάλυσης, της σχεδίασης,

της υλοποίησης και της δοκιμής, ενώ διατηρείται και η ακεραιότητα της αρχιτεκτονικής ή μεταβάλλεται όπου αυτό καθίσταται αναγκαίο. Τέλος επιβλέπονται και αντιμετωπίζονται οι κίνδυνοι που αναγνωρίστηκαν στις προηγούμενες φάσεις.

Στα μεγαλύτερα έργα, είναι δυνατή η ανάπτυξη διαφόρων επαναλήψεων της φάσης αυτής σε μία προσπάθεια διαχωρισμού των περιπτώσεων χρήσης σε εύχρηστα τμήματα τα οποία να είναι ικανά να παράγουν πρωτότυπα. Κάτι τέτοιο δεν είναι αναγκαίο και δεν αποτελεί πρακτική σε μικρού μεγέθους έργα.

Στην κατασκευή παράγεται η πρώτη απόδειξη του λογισμικού, και το αποτέλεσμα της χαρακτηρίζεται από το Ορόσημο Δυνατότητας Αρχικής Λειτουργίας.

ΦΑΣΗ ΜΕΤΑΒΑΣΗΣ

Ο αρχικός στόχος στην φάση αυτή είναι η μετάβαση του συστήματος από την ανάπτυξη στην παραγωγή, καθιστώντας το διαθέσιμο και κατανοητό στον τελικό χρήστη. Η φάση της μετάβασης συχνά ξεκινάει με την έκδοση του προϊόντος δοκιμής, το οποίο είναι ένα λογισμικό με δυνατότητα λειτουργίας.

Το προϊόν αυτό διανέμεται από τους κατασκευαστές σε ένα αντιπροσωπευτικό δείγμα χρηστών, και η λειτουργία του στο απαιτητικό αυτό περιβάλλον συνήθως είναι η δυσκολότερη δοκιμασία. Αυτό γίνεται για την συλλογή παρατηρήσεων / συστάσεων των χρηστών, οι οποίες θα χρησιμοποιηθούν για την βελτιστοποίηση του συστήματος.

Ενέργειες όπως είναι η προώθηση, η ενημέρωση των πελατών, η δημιουργία εγχειριδίων χρήσης, η διόρθωση ή η προσαρμογή του λογισμικού, αποτελούν τις κύριες εργασίες της μετάβασης. Γενικά περιλαμβάνει δραστηριότητες όπως είναι η κατάρτιση των τελικών χρηστών και των συντηρητών του λογισμικού, και η τελική δοκιμή του μετά τις διορθώσεις που εφαρμόστηκαν ως αποτέλεσμα των παρατηρήσεων των χρηστών. Το προϊόν σε αυτή τη φάση ελέγχεται σε σχέση με το ποιοτικό επίπεδο που είχε τεθεί αρχικά.

Η φάση της μετάβασης ολοκληρώνεται με την επίσημη έκδοση του προϊόντος. Με την επίτευξη όλων των στόχων, επιτυγχάνεται και το Ορόσημο Έκδοσης Προϊόντος, και ο κύκλος ανάπτυξης ολοκληρώνεται.

3.1.2 ΕΡΓΑΣΙΕΣ ΦΑΣΕΩΝ ΛΟΓΙΚΑ ΕΝΟΠΟΙΗΜΕΝΗΣ ΔΙΑΔΙΚΑΣΙΑΣ

Όπως αναφέρθηκε παραπάνω, οι βασικές σειρές εργασιών που λαμβάνουν χώρα κατά τη διάρκεια μίας φάσης, είναι η καταγραφή των απαιτήσεων, η ανάλυση, η σχεδίαση, η υλοποίηση και η δοκιμή. Στην λογικά ενοποιημένη διεργασία οι βασικές αυτές σειρές εργασιών δεν πραγματοποιούνται μία μόνο φορά, αλλά επαναλαμβάνονται σε όλες τις φάσεις, και σε κάθε επανάληψη των φάσεων, ενώ είναι δυνατόν και κάποιες από αυτές να εκλείψουν τελείως από μία φάση. Σε κάθε επανάληψη, το περιεχόμενό τους μεταβάλλεται. Εκτός όμως από τις ενέργειες αυτές, υπάρχουν και οι ενέργειες της διαμόρφωσης και διαχείρισης αλλαγών, της διαχείρισης έργου και του περιβάλλοντος.

Παρακάτω θα αναλυθούν οι βασικές μόνο σειρές εργασιών έτσι ώστε να γίνουν κατανοητές και έτσι να είναι εύκολη η εφαρμογή τους στην ανάπτυξη του συστήματος που ανατέθηκε.



ΣΧΗΜΑ XXIV: ΔΙΕΡΓΑΣΙΑ ΠΡΟΣΔΙΟΡΙΣΜΟΥ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ

ΚΑΤΑΓΡΑΦΗ ΑΠΑΙΤΗΣΕΩΝ

Ανεξάρτητα από το αν η λειτουργικότητα ενός συστήματος είναι παλιά ή καινούργια, κάθε σύστημα που βασίζεται σε λογισμικό διαθέτει ένα σκοπό, ο οποίος συνήθως εκφράζεται από τις δυνατότητές του.

Απαίτηση είναι ένα χαρακτηριστικό του συστήματος ή μία περιγραφή ενός πράγματος, που το σύστημα είναι ικανό να κάνει ώστε να εκπληρώνει το σκοπό του. (Shari Lawrence Pfleeger, 2004)

Σε αυτή τη σειρά εργασιών, γίνεται η απόσπαση των αιτημάτων και ο μετασχηματισμός τους σε ένα σύνολο προϊόντων εργασίας το οποίο περιγράφει το προς κατασκευή σύστημα, και παρέχει αναλύσεις των δυνατοτήτων και ικανοτήτων που θα πρέπει να έχει το σύστημα αυτό.

Αρχικά εκμαιεύονται οι απαιτήσεις από τους πελάτες είτε με την απάντηση ορισμένων ερωτήσεων, είτε με την παρουσίαση παρόμοιων συστημάτων, ή ακόμη και με την ανάπτυξη πρωτοτύπων για ολόκληρο το προτεινόμενο σύστημα

ή τμήμα αυτού. Στη συνέχεια οι απαιτήσεις ξαναγράφονται, συνήθως με περισσότερο μαθηματική παράσταση, έτσι ώστε οι σχεδιαστές να μπορούν να τις μετασχηματίσουν και να κατασκευάσουν ένα καλό σχέδιο του συστήματος. Το στάδιο της τεκμηρίωσης εξασφαλίζει ότι οι απαιτήσεις είναι πλήρεις, σωστές και συνεπείς, ενώ το στάδιο της επικύρωσης εξασφαλίζει ότι έχει περιγραφεί οτιδήποτε ο πελάτης επιθυμεί να περιέχει το τελικό προϊόν.

ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ

Ο στόχος της εργασίας αυτής είναι η παρουσίαση του τρόπου πραγματοποίησης του συστήματος που έχει περιγραφεί από τους πελάτες. Ένα σύστημα το οποίο θα εκτελεί – κάτω από συγκεκριμένες συνθήκες - τους στόχους και τις λειτουργίες που διευκρινίζονται στις περιγραφές των περιπτώσεων χρήσης, θα ικανοποιεί όλες τις απαιτήσεις, ενώ συγχρόνως θα είναι ευμετάβλητο στην περίπτωση που διαφοροποιηθούν οι λειτουργικές απαιτήσεις.

Γενικά ποθητή στην σειρά εργασιών αυτή, είναι η ανάλυση των απαιτήσεων που καταγράφηκαν στη προηγούμενη εργασία, επίσης η βελτίωση και η σύνθεσή τους σε ένα μοντέλο αντικειμένων. Το μοντέλο αυτό παίζει το ρόλο μιας πρώτης εικόνας του μοντέλου σχεδίασης, το οποίο θα εξελιχθεί αρκετά στο επόμενο βήμα. Η ανάλυση απαιτήσεων ανάλογα με την φάση στην οποία υλοποιείται, δίνει περισσότερη ή λιγότερη προσοχή στην αρχιτεκτονική ανάλυση, στην ανάλυση των περιπτώσεων χρήσης όπως επίσης στην ανάλυση κλάσης και πακέτου.

ΣΧΕΔΙΑΣΗ ΣΥΣΤΗΜΑΤΟΣ

Το επόμενο βήμα στην ανάπτυξη ενός συστήματος αποτελεί η μετάφραση των επιθυμιών των πελατών σε λύσεις, σε ένα σχέδιο δηλαδή, που θα ικανοποιεί τις ανάγκες τους. Η σχεδίαση είναι η δημιουργική διεργασία της μετατροπής του προβλήματος σε λύση. Η περιγραφή μίας λύσης ονομάζεται σχέδιο. (Shari Lawrence Pfleeger, 2004)

Ένα σχέδιο χρησιμεύει ως ένα πρότυπο το οποίο δείχνει την δομή του πηγαίου κώδικα, αποτελεί με άλλα λόγια μία πρώτη περιγραφή της ροής που θα ακολουθήσει ο κώδικας για την υλοποίηση του συστήματος. Αποτελείται από κλάσεις σχεδίου, δομημένες σε πακέτα και υποσυστήματα, τα οποία αναπαριστούν τα συστατικά της εφαρμογής. Επίσης περιέχει τις περιγραφές του τρόπου συνεργασίας των αντικειμένων των κλάσεων αυτών για την εκτέλεση των περιπτώσεων χρήσης.

Στην πράξη η σχεδίαση αποτελεί μία επαναληπτική εργασία δύο σταδίων. Αρχικά παράγεται το εννοιολογικό σχέδιο, το οποίο περιγράφει στον πελάτη με λεπτομέρειες τις λειτουργίες του συστήματος, και από την στιγμή που αυτό θα γίνει αποδεκτό, μετατρέπεται σε τεχνικό σχέδιο. Το τεχνικό σχέδιο αφορά τους κατασκευαστές και τους επιτρέπει να κατανοήσουν τις υπάρχουσες ανάγκες σε υλικό και λογισμικό, ώστε να δοθεί λύση στο πρόβλημα των πελατών.

ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ

Η υλοποίηση του συστήματος, ανάλογα με τη φάση στην οποία πραγματοποιείται, μπορεί να λειτουργήσει είτε αρνητικά, είτε θετικά. Έτσι επιβεβαιώνεται η λειτουργία ενός περιεχομένου του συστήματος το οποίο μας ενδιαφέρει, ή η οριστική αντιμετώπιση ενός κινδύνου. Συγχρόνως, αν δεν υλοποιηθεί το σύστημα την στιγμή που υπάρχει μία υποψήφια αρχιτεκτονική που δείχνει σημαντικά σημάδια λειτουργικότητας, αλλά νωρίτερα, γίνεται κατασπατάληση πόρων είτε ανθρώπινων είτε οικονομικών.

Οι στόχοι της υλοποίησης του συστήματος είναι ο καθορισμός της οργάνωσης του κώδικα, η εφαρμογή των συστατικών αντικειμένων και κλάσεων, ο έλεγχος των ανεπτυγμένων συστατικών σε τμήματα, αλλά και η ενσωμάτωση των αποτελεσμάτων που παράγονται από διαφορετικές ομάδες ανάπτυξης, σε ένα εκτελέσιμο σύστημα.

Τα συστήματα πραγματοποιούνται με την εφαρμογή συστατικών. Η διαδικασία περιγράφει τον τρόπο επαναχρησιμοποίησης υπάρχοντων συστατικών ή τοποθέτησης νέων συστατικών με καλά καθορισμένη την ευθύνη, δημιουργώντας ένα σύστημα ευκολότερο στη συντήρηση, το οποίο αυξάνει την πιθανότητα επαναχρησιμοποίησης του.

ΔΟΚΙΜΗ ΣΥΣΤΗΜΑΤΟΣ

Η εστίαση στη σειρά εργασιών αυτή βρίσκεται στην εξακρίβωση της ομαλής λειτουργίας των υποσυστημάτων και των διασυνδέσεων όλων των επιπέδων. Έτσι γίνεται η δοκιμή της διανομής, της αποθήκευσης και της ανάκτησης των αντικειμένων, ο έλεγχος της σχεδίασης.

Στη διάρκεια δοκιμής του συστήματος ελέγχεται η αλληλεπίδραση μεταξύ των αντικειμένων, η ολοκλήρωση όλων των συστατικών του λογισμικού, η ορθή εφαρμογή όλων των απαιτήσεων, ενώ συγχρόνως εξασφαλίζεται η εξέταση των ατελειών πριν από μία πιθανή επέκταση του λογισμικού, ο καθορισμός των ατελειών αυτών, και η διόρθωσή τους.

Οι δοκιμές πραγματοποιούνται κατά τη διάρκεια τεσσάρων ποιοτικών διαστάσεων, οι οποίες είναι η αξιοπιστία, η λειτουργία, η απόδοση της εφαρμογής, και η απόδοση του συστήματος. Για κάθε μία εκ των διαστάσεων αυτών, η λογικά ενοποιημένη διεργασία περιγράφει τον τρόπο μετάβασης από τον κύκλο ζωής της σχεδίασης, στην υλοποίηση, στην εκτέλεση και τέλος στην αξιολόγηση.

3.2 ΕΝΟΠΟΙΗΜΕΝΗ ΓΛΩΣΣΑ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ UML

Η λογικά ενοποιημένη διεργασία χρησιμοποιεί την ενοποιημένη γλώσσα μοντελοποίησης (**Unified Modeling Language**) για τη δημιουργία των συστημάτων λογισμικού. Συγκεκριμένα, η UML αποτελεί αναπόσπαστο κομμάτι της λογικά ενοποιημένης διεργασίας, αφού αναπτύχθηκαν σχεδόν παράλληλα.

Η ενοποιημένη γλώσσα μοντελοποίησης αποτελεί μία γλώσσα απεικόνισης ενός συστήματος, βασισμένο σε αντικείμενα, και όχι μία μέθοδο. Έτσι ουσιαστικά είναι ο γραφικός συμβολισμός που χρησιμοποιείται από τις μεθόδους ως μέσο έκφρασης των σχεδίων που προκύπτουν κατά την ανάπτυξη του λογισμικού.

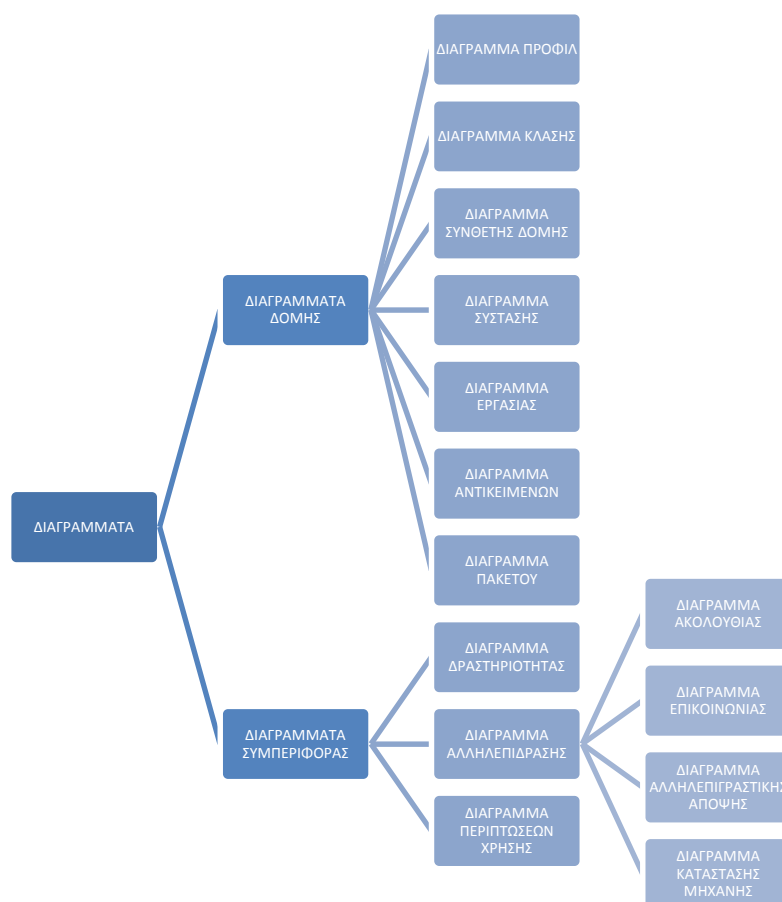
Η μοντελοποίηση ενός συστήματος παρέχει τη δυνατότητα αφαίρεσης των ασήμαντων λεπτομερειών και εστίασης στα σημαντικά σημεία, τα οποία είναι απαραίτητα στην κατανόηση πριν την κατασκευή του. Με τον τρόπο αυτό δίνεται η δυνατότητα πειραματισμού με διαφορετικές λύσεις ή προσεγγίσεις στο ίδιο πρόβλημα, και καθίσταται εφικτή η ανάλυση, η καταγραφή και η παρακολούθηση της προόδου ενός έργου. Τέλος η UML προσφέρει έναν κοινό κώδικα επικοινωνίας όσων εμπλέκονται στην κατασκευή του συστήματος.

Η UML χρησιμοποιείται για τον εντοπισμό, τη μετατροπή, την κατασκευή και την καταγραφή των αποτελεσμάτων της ανάπτυξης ενός λογισμικού. Επίσης, προσφέρει έναν τυποποιημένο τρόπο οπτικοποίησης την αρχιτεκτονικής ενός συστήματος, συμπεριλαμβάνοντας στοιχεία όπως είναι οι ρόλοι, τα συστατικά, οι δραστηριότητες και οι εντολές γλώσσας προγραμματισμού.

Είναι πολύ σημαντική η διαφοροποίηση μεταξύ ενός μοντέλου UML και μίας ομάδας διαγραμμάτων ενός συστήματος. Ένα διάγραμμα είναι μία γραφική αναπαράσταση ενός συστήματος. Ένα μοντέλο όμως περιέχει επιπρόσθετα έγγραφα, όπως είναι οι περιπτώσεις χρήσης, οι οποίες καθοδηγούν τα στοιχεία και τα διαγράμματά του.

Η UML αναπαριστά δύο διαφορετικές απόψεις του μοντέλου ενός συστήματος. Η μία είναι η στατική άποψη, η οποία δίνει έμφαση στην στατική δομή του συστήματος χρησιμοποιώντας αντικείμενα, χαρακτηριστικά, λειτουργίες και σχέσεις, ενώ περιέχει τα διαγράμματα κλάσεων και τα διαγράμματα σύνθετης δομής. Η δεύτερη άποψη είναι η δυναμική, η οποία ασχολείται με την συμπεριφορά του συστήματος αναπαριστώντας συνεργασίες μεταξύ των αντικειμένων και αλλαγές στις καταστάσεις τους, ενώ περιλαμβάνει τα διαγράμματα ακολουθίας, δραστηριοτήτων, και κατάστασης μηχανής.

Στην τελευταία έκδοση της UML, υπάρχουν 14 διαγράμματα, τα οποία αναπαριστώνται παρακάτω.



ΣΧΗΜΑ XXV: ΔΙΑΓΡΑΜΜΑΤΑ UML

Τα σημαντικότερα εκ των παραπάνω είναι τα διαγράμματα περιπτώσεων χρήσης, κλάσης, ακολουθίας, συνεργασίας, δραστηριότητας και κατάστασης, και τα οποία θα αναλυθούν στη συνέχεια.

3.2.1 ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΗΣ ΧΡΗΣΗΣ

Για να αναπτυχθεί ένα σύστημα, είναι χρήσιμο να εξαχθούν οι απαιτήσεις και να περιγραφούν, κάτι που επιτυγχάνεται με τη χρήση της τεχνικής που ονομάζεται περίπτωση χρήσης. Μία περίπτωση χρήσης (use case) περιγράφει την συγκεκριμένη λειτουργικότητα που ένα σύστημα υποτίθεται ότι εκτελεί ή επιδεικνύει μοντελοποιώντας τους διαλόγους που ένας χρήστης, ένα εξωτερικό σύστημα ή κάποια άλλη οντότητα θα πρέπει να διεξάγει με το σύστημα που αναπτύσσεται. Με άλλα λόγια διαμοιράζει τη λειτουργικότητα ενός συστήματος σε συναλλαγές που έχουν νόημα για τους χρήστες του συστήματος, οι οποίοι ονομάζονται χειριστές ή χαρακτήρες.

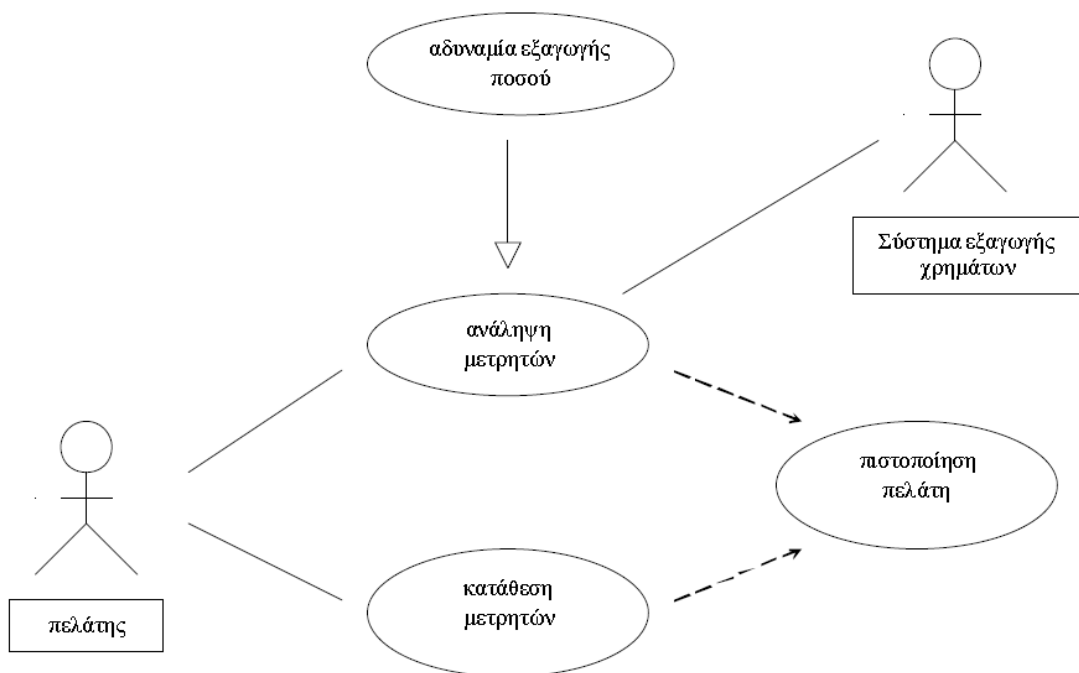
Κάθε περίπτωση χρήσης περιγράφει ένα πιθανό σενάριο για τον τρόπο που μια εξωτερική οντότητα αλληλεπιδρά με το σύστημα. Έτσι περιέχει ένα ιδανικό κύριο σενάριο στο οποίο δεν υπάρχουν προβλήματα, αλλά και αρκετά εναλλακτικά, που αντιμετωπίζονται διάφορα προβλήματα. Οι περιπτώσεις χρήσης στην ολότητά τους συνιστούν μία περιγραφή όλων των πιθανών τρόπων χρήσης του συστήματος από όλες τις πιθανές οντότητες. Επομένως, η συλλογή των περιπτώσεων χρήσης σχεδιάζει μία εικόνα της πλήρους λειτουργικότητας του συστήματος.

Τα διαγράμματα περιπτώσεων χρήσης (use case diagram) τα οποία περιγράφουν ένα σύστημα, περιλαμβάνουν συμβολικά το ίδιο το σύστημα, τους χειριστές, τις περιπτώσεις και τις σχέσεις. Όπως σημειώθηκε και παραπάνω, ένας χειριστής (actor) είναι ένας ρόλος που μία οντότητα παίζει σε σχέση με το σύστημα. Η περίπτωση (case) είναι μία απεικόνιση κάποιας πλευράς της λειτουργικότητας του συστήματος που είναι ορατή στο χειριστή. Η σχέση υποδηλώνει τη σχέση ενός χειριστή με μία περίπτωση χρήσης. Η σχέση μπορεί να είναι μία σχέση συμπερίληψης στην οποία μία περίπτωση χρήσης συμπεριλαμβάνει μία άλλη, αλλά και επέκτασης στην οποία μία περίπτωση χρήσης επεκτείνεται από μία άλλη ανάλογα με τις επιλογές ή την κατάσταση κάποιου χειριστή.

Για την απεικόνιση ενός χαρακτήρα, χρησιμοποιείται μία φιγούρα ανθρώπου και προστίθεται ένα όνομα από κάτω, το οποίο επισημαίνει το ρόλο που παίζει στα πλαίσια μίας περίπτωσης χρήσης. Το σύμβολο για την περίπτωση χρήσης είναι η έλλειψη, μέσα στην οποία αναγράφεται το όνομα της. Η σχέση παριστάνεται με μία γραμμή που συνενώνει τον χειριστή με την περίπτωση χρήσης. Όταν η σχέση είναι συμπερίληψης τότε αυτή συμβολίζεται με διακεκομμένη κατευθυνόμενη γραμμή όπου η φορά του βέλους έχει φορά από την περίπτωση χρήσης που συμπεριλαμβάνει προς αυτή που συμπεριλαμβάνεται. Στην περίπτωση που η σχέση είναι επέκτασης, την συμβολίζουμε με ένα βέλος με κατεύθυνση από την επέκταση περίπτωσης χρήσης προς την επεκτάσιμη περίπτωση χρήσης.



ΣΧΗΜΑ XXVI: ΣΥΜΒΟΛΑ ΔΙΑΓΡΑΜΜΑΤΟΣ ΠΕΡΙΠΤΩΣΗΣ ΧΡΗΣΗΣ



ΣΧΗΜΑ XXVII: ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΓΡΑΜΜΑΤΟΣ ΠΕΡΙΠΤΩΣΗΣ ΧΡΗΣΗΣ

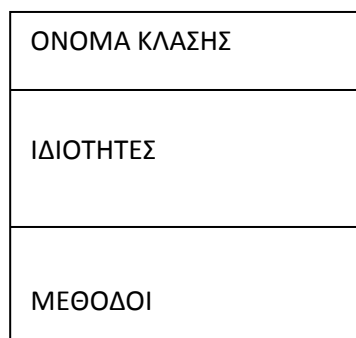
3.2.2 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΗΣ

Η στατική άποψη ενός μοντέλου είναι θεμελιώδης καθώς αποτυπώνει τα τμήματα τα οποία διαμορφώνουν το σύστημα, αλλά και τις σχέσεις μεταξύ τους. Οι κλάσεις αποτελούν τα δομικά στοιχεία ενός αντικειμενοστραφούς συστήματος, ενώ οι σχέσεις μεταξύ των κλάσεων επιτρέπουν τη συνεργασία των αντικειμένων.

Στα διαγράμματα κλάσεων (class diagram) είναι δυνατή η απεικόνιση υψηλού επιπέδου λεπτομερειών όπως είναι οι ιδιότητες και οι λειτουργίες μίας κλάσης και οι σχέσεις κληρονομικότητας μεταξύ τους. Αξίζει να σημειωθεί πως είναι τα μοναδικά διαγράμματα που παρέχουν πληροφορίες για τον πηγαίο κώδικα.

Ένα αντικείμενο είναι μία διακριτή οντότητα με ταυτότητα, κατάσταση και συμπεριφορά που είναι δυνατόν να ενεργοποιηθεί. Μία κλάση προσδιορίζει ένα σύνολο αντικειμένων που έχουν διαφορετική κατάσταση και κοινή συμπεριφορά. Η κατάσταση των αντικειμένων καθορίζεται από τις ιδιότητές τους και τις μεταξύ τους συσχετίσεις. Η συμπεριφορά των αντικειμένων όμως καθορίζεται από τις λειτουργίες των κλάσεων τους. Μια μέθοδος αποτελεί την υλοποίηση μίας λειτουργίας σε μία κλάση.

Μία κλάση σε ένα διάγραμμα της UML συμβολίζεται με ένα ορθογώνιο διαμερισμένο σε τρία τμήματα, στο άνω μέρος του οποίου αναγράφεται το όνομα της, στο μεσαίο τμήμα οι ιδιότητες και στο κάτω τμήμα οι μέθοδοί της. Αν δεν υπάρχει ανάγκη αναφοράς των ιδιοτήτων και των μεθόδων, σημειώνεται μόνο το όνομα της κλάσης, με τα άλλα δύο τμήματα να αφήνονται κενά.



ΣΧΗΜΑ XXVIII: ΣΥΜΒΟΛΙΣΜΟΣ ΚΛΑΣΗΣ

Οι ιδιότητες των κλάσεων συμβολίζονται ως εξής:

ορατότητα όνομα : τύπος = προκαθορισμένη_τιμή

Η ορατότητα που αναφέρεται στον συμβολισμό των ιδιοτήτων μπορεί να λάβει την τιμή «δημόσια» (public) που συμβολίζεται με (+) και ορίζει πως μία ιδιότητα είναι άμεσα προσπελάσιμη από άλλες κλάσεις, «προστατευμένη» (protected) που συμβολίζεται με (#) οπότε η ιδιότητα είναι προσπελάσιμη μόνο από την ίδια κλάση, και «ιδιωτική» (private) που συμβολίζεται με (-) και ορίζει πως είναι προσπελάσιμη μόνο από την ίδια την κλάση και τις παράγωγές της.

Το όνομα μίας ιδιότητας είναι ένα αλφαριθμητικό το οποίο αποτελεί το αναγνωριστικό της συγκεκριμένης ιδιότητας. Ο καθορισμός του ονόματος μίας ιδιότητας είναι υποχρεωτικός ενώ οι υπόλοιποι προσδιοριστές είναι προαιρετικοί.

Ο τύπος αφορά τον τύπο δεδομένων της συγκεκριμένης ιδιότητας. Ο τύπος δεδομένων περιέχει ένα σύνολο τιμών και ένα σύνολο επιτρεπτών λειτουργιών επί του συνόλου τιμών.

Η προκαθορισμένη τιμή είναι μία τιμή που ανατίθεται στη συγκεκριμένη ιδιότητα κατά τη δημιουργία του αντικειμένου, εκτός και αν προσδιοριστεί διαφορετικά.

Οι λειτουργίες των κλάσεων συμβολίζονται ως ακολούθως:

ορατότητα όνομα (λίστα παραμέτρων) : επιστρεφόμενος_τύπος

Η σημασία της ορατότητας είναι η ίδια με αυτή των ιδιοτήτων και λαμβάνει τις ίδιες τιμές.

Το όνομα είναι ένα αλφαριθμητικό που μπορεί να χρησιμοποιηθεί από ένα άλλο αντικείμενο για την κλήση της συγκεκριμένης λειτουργίας.

Η λίστα παραμέτρων που αναφέρεται περιλαμβάνει παραμέτρους απαραίτητες για την εκτέλεση της λειτουργίας.

Ο επιστρεφόμενος τύπος είναι ο τύπος της τιμής που επιστρέφεται από την κλήση της λειτουργίας.

Student
-firstName : String
-lastName : String
+getName() : String

ΣΧΗΜΑ XXIX: ΠΑΡΑΔΕΙΓΜΑ ΚΛΑΣΗΣ

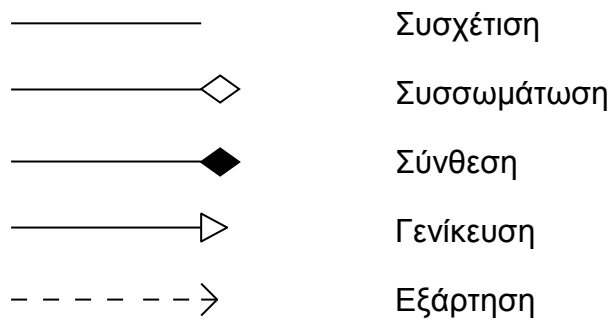
Οι σχέσεις μεταξύ των οντοτήτων ενός συστήματος μπορεί να απεικονίζονται έννοιες όπως συσχετίσεις, συσσωματώσεις και συνθέσεις, κληρονομικότητες και εξαρτήσεις.

Οι συσχετίσεις αποτελούν ουσιαστικά τις διασυνδέσεις που τα αντικείμενα διαφόρων κλάσεων μπορούν να χρησιμοποιήσουν για να αλληλεπιδρούν και να ανταλλάσσουν μηνύματα. Κάθε συσχέτιση έχει δύο άκρα τα οποία γίνεται να ονομαστούν ώστε να προσδιορίζουν ρόλους. Η πολλαπλότητα είναι μία ιδιότητα των άκρων, ίσως η σημαντικότερη, και απεικονίζει τον αριθμό των στιγμιότυπων μίας κλάσης που είναι δυνατόν να συσχετιστούν με ένα στιγμιότυπο μίας άλλης κλάσης. Ο συμβολισμός μίας συσχέτισης σε ένα διάγραμμα κλάσεων είναι μία συνεχής γραμμή που συνδέει τις συμμετέχουσες κλάσεις. Το όνομα της συσχέτισης τοποθετείται κοντά στη γραμμή, ενώ τα ονόματα των άκρων όπως και η πολλαπλότητά τους σημειώνονται στα αντίστοιχα άκρα. Οι συσχετίσεις έχουν τη δυνατότητα κατεύθυνσης και έτσι παρέχεται πρόσθετη πληροφορία για το είδος τους.

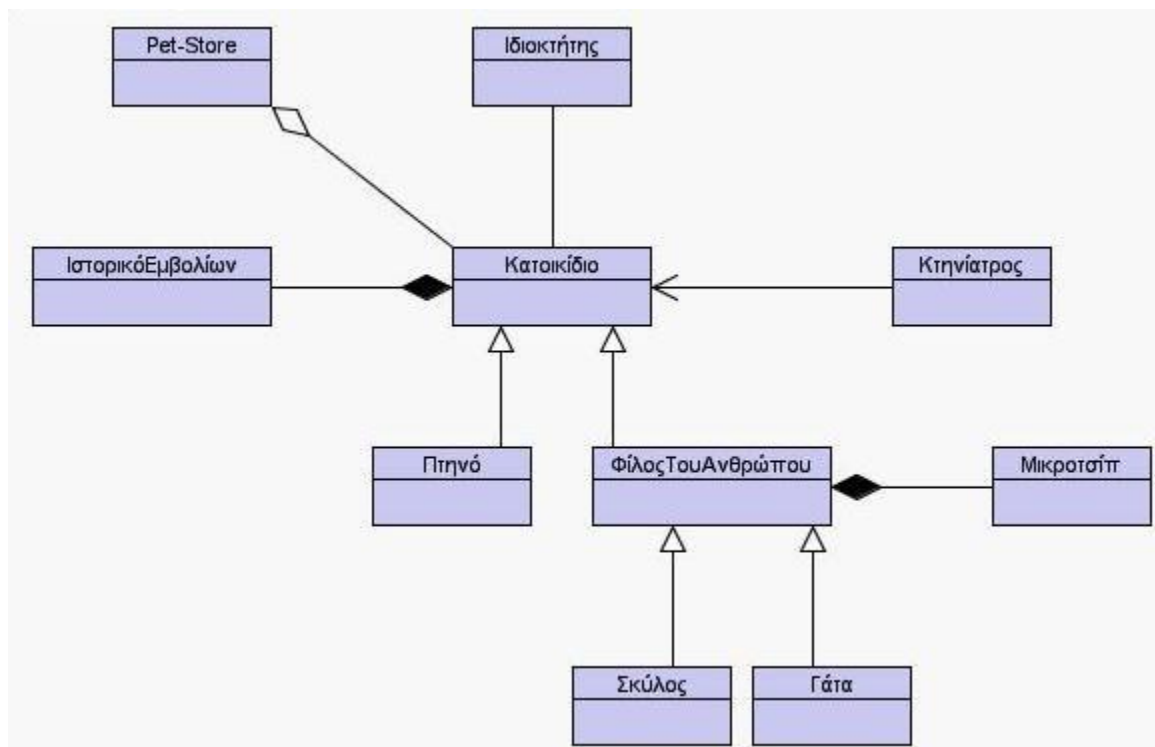
Η συσσωμάτωση και η σύνθεση αποτελούν ειδικές περιπτώσεις συσχετίσεων. Και οι δύο υποδηλώνουν τη συσχέτιση μίας κλάσης με κάποια άλλη κλάση που αποτελεί μέρος της. Η συσσωμάτωση είναι μία συσχέτιση που αναπαριστά μία σχέση «όλου – μέρους». Συμβολίζεται με ένα λευκό ρόμβο στο άκρο της συσχέτισης που αντιστοιχεί στο «όλο». Η σύνθεση αποτελεί μία ισχυρότερης μορφής συσχέτιση όπου το «όλο» έχει την αποκλειστική ευθύνη διαχείρισης των τμημάτων, όπως τη δημιουργία και διαγραφή τους. Συμβολίζεται με ένα μαύρο ρόμβο στο άκρο του συνόλου.

Η σχέση κληρονομικότητας (γενίκευση ή εξειδίκευση) αναφέρεται στη σχέση μεταξύ μίας γενικής και μίας ειδικότερης περιγραφής που αποτελεί την επέκταση της πρώτης. Η ειδικότερη περιγραφή έχει τις ίδιες ιδιότητες, λειτουργίες και σχέσεις με την γενική αλλά μπορεί να περιλαμβάνει επιπρόσθετη πληροφορία. Μία γενίκευση έχει ως σύμβολο της ένα μικρό τρίγωνο βέλος στο άκρο που αντιστοιχεί στην βασική κλάση της σχέσης. Είθισται για τις σχέσεις αυτές να χρησιμοποιείται κάθετη διάταξη, έτσι ώστε η βασική κλάση να βρίσκεται υψηλότερα από τις παράγωγες κλάσεις. Πολλαπλότητα και περιγραφή δεν σημειώνονται σε αυτές τις σχέσεις.

Μία εξάρτηση ουσιαστικά αναφέρεται στη σημασιολογική σχέση μεταξύ δύο ή περισσότερων στοιχείων ενός μοντέλου. Αν δύο κλάσεις A και B συνδέονται με σχέση εξάρτησης (από A προς B), υποδηλώνεται ότι η A απαιτεί την ύπαρξη της B για την αποστολή μηνυμάτων προς αυτήν. Η εξάρτηση μίας κλάσης A από μία κλάση B, συμβολίζεται ως μία διακεκομμένη γραμμή με βέλος, και με κατεύθυνση από την A προς την B.



ΣΧΗΜΑ XXX: ΣΥΜΒΟΛΙΣΜΟΙ ΣΧΕΣΕΩΝ ΜΕΤΑΞΥ ΚΛΑΣΕΩΝ



ΣΧΗΜΑ XXXI: ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΓΡΑΜΜΑΤΟΣ ΚΛΑΣΕΩΝ

3.2.3 ΔΙΑΓΡΑΜΜΑ ΑΚΟΛΟΥΘΙΑΣ

Ένα διάγραμμα ακολουθίας (sequence diagram) ουσιαστικά αποτελεί την αλληλεπίδραση μεταξύ αντικειμένων σε κάθετη και σε οριζόντια διάσταση. Η κάθετη διάσταση αντιστοιχεί στον χρόνο ενώ στην οριζόντια διάσταση συμβολίζονται αντικείμενα. Τα αντικείμενα παριστάνονται ως ορθογώνια παραλληλόγραμμα εντός των οποίων σημειώνεται το όνομα του αντικειμένου που συμμετέχει στο σενάριο που απεικονίζεται. Σημαντικότερο όμως είναι να σημειωθεί μετά από μία άνω και κάτω τελεία “:”, το όνομα της κλάσης στην οποία ανήκει κάθε αντικείμενο. Στην περίπτωση που σε ένα σενάριο εκτός από αντικείμενα του συστήματος εμπλέκονται και εξωτερικοί χρήστες, οι χρήστες αυτοί συμβολίζονται ως ανθρώπινες φιγούρες.

Κάτω από κάθε αντικείμενο προεκτείνεται μία κάθετη γραμμή που ονομάζεται γραμμή ζωής. Η γραμμή αυτή είναι διακεκομμένη, ενώ για όσο χρόνο μία διαδικασία του αντικειμένου είναι ενεργή, η γραμμή ζωής παριστάνεται με ένα ορθογώνιο παραλληλόγραμμα που επεκτείνεται καθ' ύψος της. Ο συμβολισμός που χρησιμοποιείται για ένα μήνυμα που αποστέλλεται μεταξύ αντικειμένων είναι μία ακμή από τη γραμμή ζωής ενός αντικειμένου προς τη γραμμή ζωής ενός άλλου. Καθώς το μήνυμα αποτελεί μία λειτουργία του αντικειμένου – παραλήπτη, σημειώνεται το όνομα του μηνύματος, και μέσα σε παρενθέσεις οι τυχόν παράμετροι που απαιτούνται. Η θέση των ακμών επάνω στη γραμμή ζωής εξαρτάται από την χρονική στιγμή που αυτά λαμβάνουν χώρα.

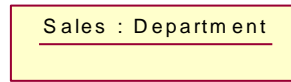
Η λήψη ενός μηνύματος από ένα αντικείμενο σηματοδοτεί την πραγματοποίηση ενός γεγονότος. Ένα γεγονός μπορεί να είναι ενεργοποιούμενο από το χρήστη του συστήματος, από διαφορετικό υπολογιστικό σύστημα, από κάποιο άλλο αντικείμενο του ίδιου συστήματος ή τέλος από την πάροδο του χρόνου.

Όταν ένα αντικείμενο λαμβάνει ένα μήνυμα μπορεί να αλλάξει την κατάστασή του, να προωθήσει το μήνυμα σε διαφορετικό αντικείμενο, να επιστρέψει κάποια τιμή ή να αγνοήσει το γεγονός.

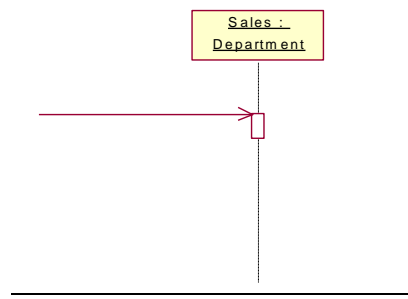
Ορισμένες φορές κρίνεται σκόπιμο να αναπαρασταθεί σε ένα διάγραμμα ακολουθίας το γεγονός της επιστροφής ενός μηνύματος. Η επιστροφή αυτή υποδηλώνεται με μία διακεκομμένη ακμή μεταξύ της γραμμής ζωής του καλούμενου αντικειμένου και της γραμμής ζωής του καλούντος αντικειμένου.

Υπάρχει επίσης η περίπτωση όπου ένα αντικείμενο του συστήματος δημιουργεί δυναμικά ένα άλλο αντικείμενο, γεγονός που συμβολίζεται με μία ακμή που καταλήγει στο δημιουργηθέν αντικείμενο, ενώ είναι δυνατόν να υποδηλώσουμε και την καταστροφή ενός αντικειμένου σημειώνοντας ένα X το οποίο τερματίζει τη γραμμή ζωής του.

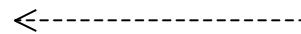
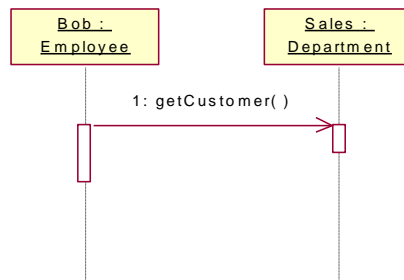
ΑΝΤΙΚΕΙΜΕΝΟ :



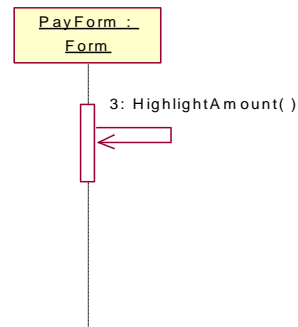
ΠΛΑΙΣΙΟ ΕΝΕΡΓΟΠΟΙΗΣΗΣ :



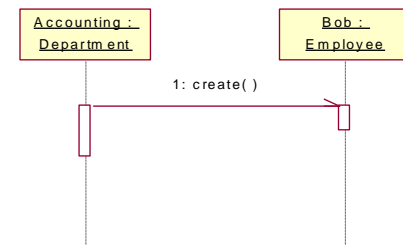
ΑΠΟΣΤΟΛΗ ΜΗΝΥΜΑΤΟΣ :



ΕΠΙΣΤΡΟΦΗ ΜΗΝΥΜΑΤΟΣ :

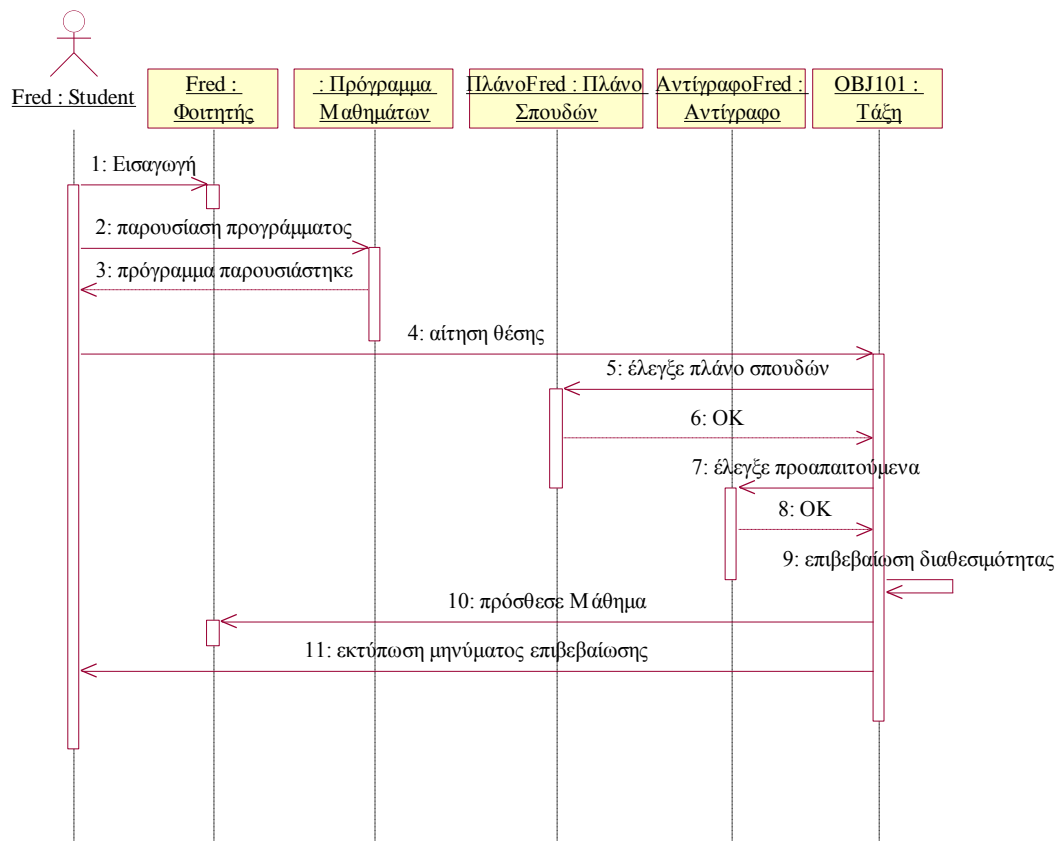


ΑΥΤΟ – ΚΛΗΣΗ :



ΑΣΥΓΧΡΟΝΟ ΜΗΝΥΜΑ :

ΣΧΗΜΑ XXXII: ΣΤΟΙΧΕΙΑ ΔΙΑΓΡΑΜΜΑΤΩΝ ΑΚΟΛΟΥΘΙΑΣ



ΣΧΗΜΑ XXXIII: ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΓΡΑΜΜΑΤΟΣ ΑΚΟΛΟΥΘΙΑΣ

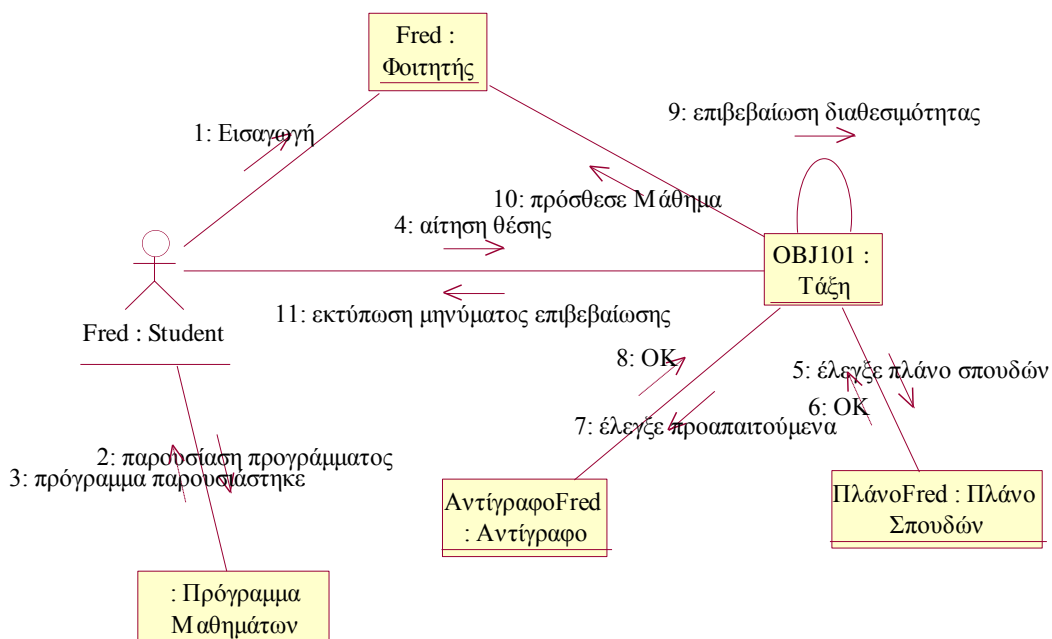
3.2.4 ΔΙΑΓΡΑΜΜΑ ΣΥΝΕΡΓΑΣΙΑΣ

Ένα διάγραμμα συνεργασίας (collaboration diagram) χρησιμοποιείται για την απεικόνιση των συνεργαζόμενων αντικείμενων και των συσχετίσεων μεταξύ των κλάσεων τους. Με άλλα λόγια απεικονίζονται οι συνδέσεις μεταξύ των αντικειμένων. Σημειώνεται ότι στα διαγράμματα συνεργασίας οι γραμμές αποτελούν στιγμιότυπα συσχετίσεων, δηλαδή συνδέσεις που δημιουργούνται και καταστρέφονται δυναμικά κατά τη διάρκεια εκτέλεσης του προγράμματος.

Συγκεκριμένη μορφή ή διάταξη σε ένα διάγραμμα συνεργασίας δεν υπάρχει, ενώ η ακολουθία των μηνυμάτων που ανταλλάσσονται απεικονίζονται με την χρήση αρίθμησης.

Τα διαγράμματα ακολουθίας και συνεργασίας περιέχουν την ίδια πληροφορία αλλά το κάθε ένα την παρουσιάζει από μία διαφορετική οπτική γωνία. Είναι συνηθέστερη η χρήση των διαγραμμάτων ακολουθίας ως καταλληλότερα για τη διερεύνηση των σεναρίων μίας περίπτωσης χρήσης και τον εντοπισμό λειτουργιών αλλά και σχέσεων μεταξύ τους.

Από ένα τέτοιο διάγραμμα γίνεται εύκολά αντιληπτή η ομάδα των συνεργαζόμενων αντικειμένων και η ύπαρξη των συνδέσεων μεταξύ τους, αλλά είναι δυσκολότερη η οπτικοποίηση της ροής των μηνυμάτων.



ΣΧΗΜΑ XXXIV: ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΓΡΑΜΜΑΤΟΣ ΣΥΝΕΡΓΑΣΙΑΣ

Με τα διαγράμματα ακολουθίας και συνεργασίας, δίνεται μεγαλύτερη έμφαση στη ροή της πληροφορίας, αλλά για την απεικόνιση της ροής ελέγχου καταλληλότερα είναι τα διαγράμματα δραστηριότητας και καταστάσεων τα οποία και αναλύονται παρακάτω.

3.2.5 ΔΙΑΓΡΑΜΜΑ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ

Ένα διάγραμμα δραστηριότητας (activity diagram) μοντελοποιεί τη ροή της εργασίας, αναπαριστώντας τις διάφορες καταστάσεις εκτέλεσης ενός υπολογισμού (Bohm & Jacorini).

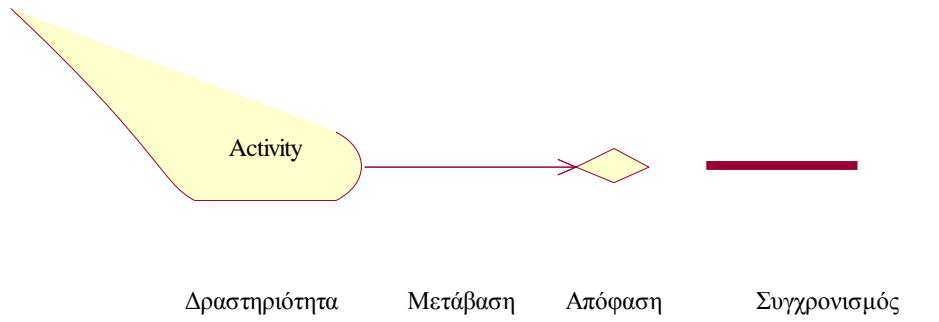
Στο διάγραμμα δραστηριοτήτων οι κόμβοι συμβολίζουν καταστάσεις ενεργειών και οι μεταβάσεις λαμβάνουν χώρα με την ολοκλήρωση αυτών των ενεργειών. Τυπικά τα διαγράμματα δραστηριοτήτων χρησιμοποιούνται για την περιγραφή υπολογισμών ή ροών εργασίας.

Μία δραστηριότητα είναι μία κατάσταση στην οποία πραγματοποιείται είτε μία διαδικασία του πραγματικού κόσμου, είτε η εκτέλεση μίας ενότητας λογισμικού. Το διάγραμμα δραστηριότητας περιγράφει την ακολουθία των δραστηριοτήτων, και υποστηρίζει τόσο την παράλληλη, όσο και την υπό συνθήκη συμπεριφορά.

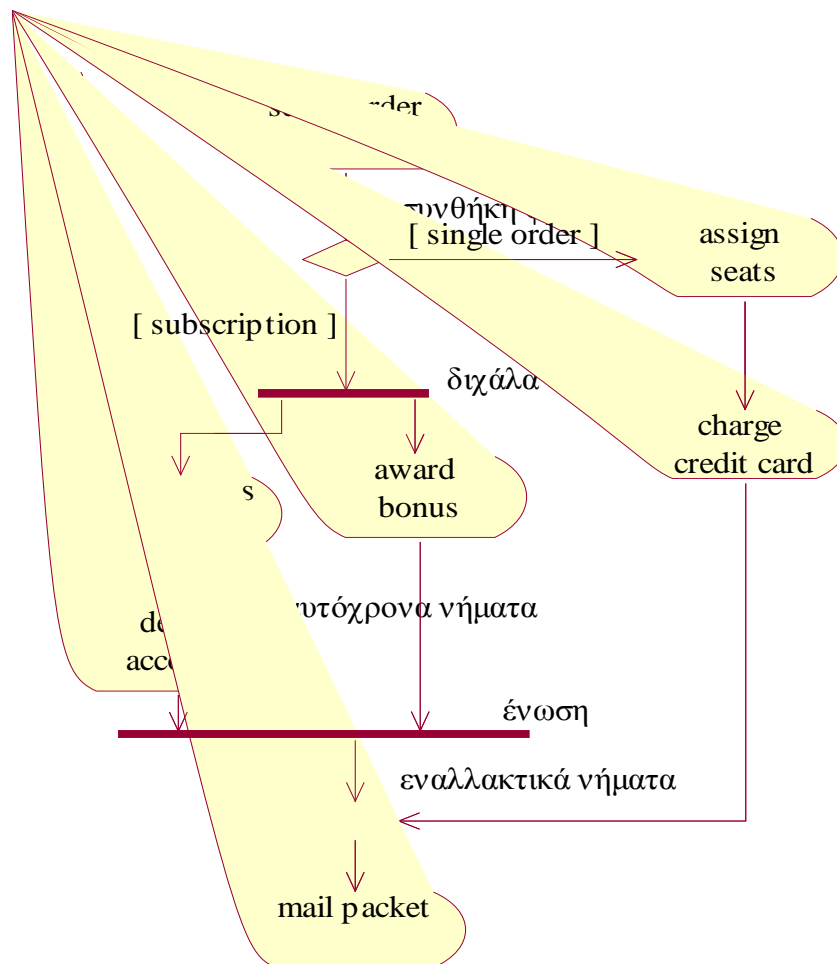
Η παράλληλη συμπεριφορά παριστάνεται με διχάλες και ενώσεις. Μία διχάλα περιέχει μία εισερχόμενη μετάβαση και πολλές εξερχόμενες. Όταν συμβαίνει η εισερχόμενη μετάβαση, ακολουθούνται παράλληλα όλες οι εξερχόμενες μεταβάσεις. Σε μία ένωση, η εξερχόμενη μετάβαση ακολουθείται μόνο όταν έχουν ολοκληρωθεί οι δραστηριότητες από όλες οι καταστάσεις των εισερχόμενων μεταβάσεων. Οι διχάλες και οι ενώσεις πρέπει να ταιριάζουν, γεγονός που σημαίνει απλά πως μετά από μία διχάλα ακολουθεί μία ένωση.

Η υπό συνθήκη συμπεριφορά περιγράφεται από διακλαδώσεις και συγχωνεύσεις. Μία διακλάδωση έχει μόνο μία εισερχόμενη μετάβαση και πολλές φρουρούμενες εξερχόμενες μεταβάσεις. Μόνο μία από τις οποίες είναι δυνατόν να ακολουθηθεί, και έτσι οι φρουροί πρέπει να είναι αμοιβαία αποκλειόμενες συνθήκες. Η χρήση της φράσης [else] στη θέση ενός φρουρού σημαίνει ότι η μετάβαση πρέπει να χρησιμοποιηθεί αν όλοι οι υπόλοιποι φρουροί σε αυτή τη διακλάδωση είναι ψευδείς. Μία συγχώνευση έχει πολλές εισερχόμενες μεταβάσεις και μία μόνο έξοδο και ουσιαστικά σημαίνει το τέλος μίας υπό συνθήκη συμπεριφοράς που ξεκίνησε με διακλάδωση.

Μία κατάσταση δραστηριότητας συμβολίζεται ως ένα ορθογώνιο με καμπύλες γωνίες (παρόμοιο με έκλειψη) το οποίο εσωτερικά περιλαμβάνει την περιγραφή της δραστηριότητας. Η μετάβαση κατά τη συμπλήρωση μίας δραστηριότητας συμβολίζεται ως ακμή. Οι διακλαδώσεις συμβολίζονται είτε με συνθήκες φρουρούς επί των μεταβάσεων είτε με κόμβους απόφασης με πολλαπλές εξερχόμενες ακμές.



ΣΧΗΜΑ XXXV: ΣΥΜΒΟΛΙΣΜΟΙ ΣΤΟΙΧΕΙΩΝ ΔΙΑΓΡΑΜΜΑΤΟΣ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ



ΣΧΗΜΑ XXXVI: ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΓΡΑΜΜΑΤΟΣ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ

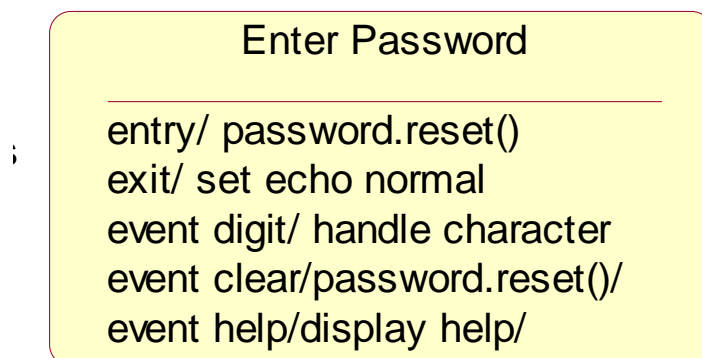
3.2.6 ΔΙΑΓΡΑΜΜΑ ΚΑΤΑΣΤΑΣΗΣ

Ένα διάγραμμα κατάστασης συμβολίζει τη δυναμική συμπεριφορά των αντικειμένων μίας κλάσης και τον τρόπο που μεταβάλλεται η κατάστασή τους λόγω διαφόρων συμβάντων. Το διάγραμμα κατάστασης μας βοηθάει να αντιληφθούμε αυτήν την συμπεριφορά.

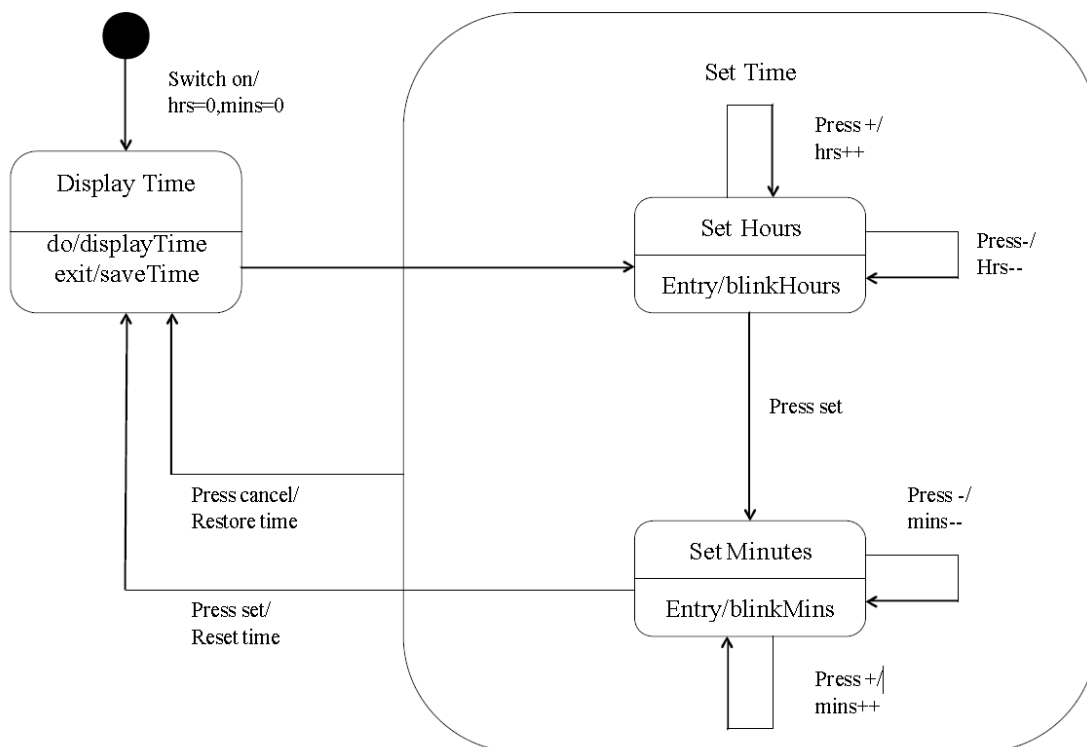
Μία κατάσταση ουσιαστικά αποτελεί την περιγραφή μίας χρονικής περιόδου κατά τη διάρκεια ζωής ενός αντικειμένου. Μπορεί να χαρακτηριστεί είτε ως ένα σύνολο τιμών αντικειμένων που είναι παρόμοιες από κάποια άποψη, είτε ως μία περίοδο κατά την οποία ένα αντικείμενο αναμένει την εμφάνιση ενός γεγονότος ή εκτελεί μία εργασία.

Όταν ένα αντικείμενο βρίσκεται σε μία κατάσταση, αποκρίνεται σε γεγονότα που βρίσκονται στις μεταβάσεις που ξεκινούν από αυτή. Μία κατάσταση συμβολίζεται ως ένα ορθογώνιο με καμπύλες γωνίες, και είναι δυνατόν να περιλαμβάνει ενέργειες εισόδου που λαμβάνουν χώρα όταν μία μετάβαση καθιστά την κατάσταση ενεργή, καθώς και ενέργειες εξόδου που λαμβάνουν χώρα πριν την απενεργοποίηση μίας μετάβασης.

Μία εσωτερική μετάβαση έχει αρχική κατάσταση αλλά όχι τελική. Κατά συνέπεια δεν προκαλεί αλλαγή της ενεργού κατάστασης, αλλά μπορεί να παρέχει ενέργειες που πραγματοποιούνται.



ΣΧΗΜΑ XXXVII: ΣΥΜΒΟΛΙΣΜΟΣ ΚΑΤΑΣΤΑΣΗΣ ΔΙΑΓΡΑΜΜΑΤΟΣ ΚΑΤΑΣΤΑΣΗΣ



ΣΧΗΜΑ XXXVIII: ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΓΡΑΜΜΑΤΟΣ ΚΑΤΑΣΤΑΣΕΩΝ

3.3 ΕΦΑΡΜΟΓΗ ΤΗΣ RUP ΚΑΙ ΤΗΣ UML ΣΤΗΝ ΑΝΑΠΤΥΞΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΒΙΟΜΗΧΑΝΙΚΑ ΠΛΥΝΤΗΡΙΑ

Οι γνώσεις που αποκτήθηκαν στο κεφάλαιο αυτό θα εφαρμοστούν για την ανάπτυξη και σχεδίαση λογισμικού για βιομηχανικά πλυντήρια με την χρήση προγραμματιζόμενων λογικών ελεγκτών. Έτσι τα πάντα θα προσαρμοστούν ανάλογα για να γίνει δυνατή η λειτουργία τους σε αυτοματισμούς. Η ανάπτυξη με την χρήση της RUP και ο σχεδιασμός του λογισμικού με την χρήση της UML θα γίνεται παράλληλα έτσι ώστε να είναι καλύτερα αντιληπτή η διαδικασία που απαιτείται για την παραγωγή του τελικού αποτελέσματος. Η υλοποίηση του λογισμικού, η δημιουργία του κώδικα δηλαδή, θα γίνει στο επόμενο κεφάλαιο.

3.3.1 ΣΥΛΛΗΨΗ

Το έργο της παραγωγής λογισμικού για την λειτουργία ενός βιομηχανικού πλυντηρίου είναι σχετικά μικρό και έτσι είναι δυνατόν να ολοκληρωθεί σε δύο επαναλήψεις. Κατά τη διάρκεια των επαναλήψεων θα ανακαλυφθούν οι βασικές περιπτώσεις χρήσης και θα δημιουργηθεί μία πρώτη γενική εικόνα του συστήματος.

ΠΡΩΤΗ ΕΠΑΝΑΛΗΨΗ

Η πρώτη επανάληψη σηματοδοτεί και την έναρξη του έργου. Επειδή έχει προαποφασιστεί η πραγματοποίηση του έργου δεν θα ελεγχθεί η ύπαρξη συμφέροντος από την δημιουργία του. Η καταγραφή της βασικής λειτουργίας του συστήματος αλλά και των περιπτώσεων χρήσης αυτού γίνεται ονομαστικά κατά την πρώτη επανάληψη και μετά από συνάντηση με τους χρήστες του βιομηχανικού πλυντηρίου. Συγχρόνως γίνεται και η οριοθέτηση της εμβέλειας του συστήματος και η αναγνώριση των χρηστών του.

Η βασική λειτουργία του συστήματος είναι η πλύση ρούχων. Συγκεκριμένα ο χρήστης μετά από την επιλογή συγκεκριμένου προγράμματος και θερμοκρασίας, τοποθετεί ρούχα στο πλυντήριο καθώς και τα απαραίτητα απορρυπαντικά, ασφαρίζει την πόρτα και ξεκινάει την πλύση των ρούχων που συνεπώς αποτελεί και την μοναδική περίπτωση χρήσης. Ο μοναδικός χρήστης του συστήματος είναι ο χρήστης του βιομηχανικού πλυντηρίου.

ΠΙΝΑΚΑΣ V: ΠΙΝΑΚΑΣ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ

ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ
ΠΛΥΣΗ ΡΟΥΧΩΝ

ΠΙΝΑΚΑΣ VI: ΠΙΝΑΚΑΣ ΧΡΗΣΤΩΝ ΚΑΙ ΣΤΟΧΩΝ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

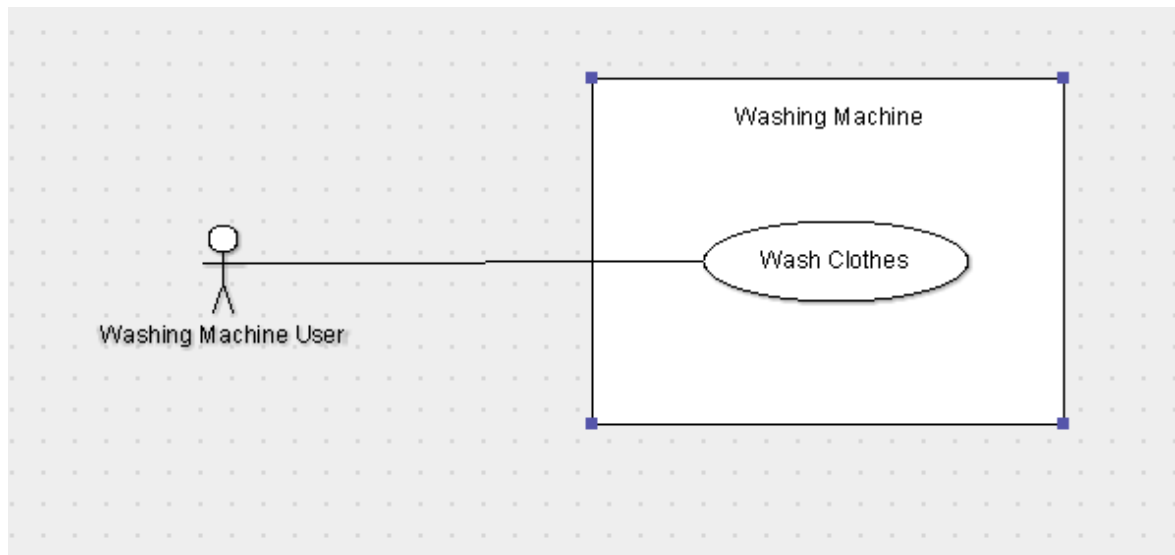
ΧΡΗΣΤΕΣ	ΣΤΟΧΟΙ
ΧΡΗΣΤΗΣ ΒΙΟΜΗΧΑΝΙΚΟΥ ΠΛΥΝΤΗΡΙΟΥ	ΠΛΥΣΗ ΡΟΥΧΩΝ

ΔΕΥΤΕΡΗ ΕΠΑΝΑΛΗΨΗ

Μετά την πρώτη επανάληψη και την ονομαστική καταγραφή των περιπτώσεων χρήσης του συστήματος αλλά και τον προσδιορισμό των χρηστών του συστήματος και των στόχων τους, πραγματοποιείται η δεύτερη επανάληψη.

Στη δεύτερη επανάληψη της φάσης της σύλληψης θα οριοθετηθεί η εμβέλεια του συστήματος, και θα αναπτυχθεί το βασικό σενάριο που αφορά την περίπτωση χρήσης που καταγράφηκε, δηλαδή της πλύσης ρούχων. Έτσι θα αποκτηθεί καλύτερη άποψη για το προς ανάπτυξη σύστημα.

Οι πληροφορίες που διατίθενται δίνουν την δυνατότητα να σχηματιστεί ένα αρχικό διάγραμμα περιπτώσεων χρήσης, όπου θα φαίνονται εκτός από την περίπτωση χρήσης «πλύση ρούχων», ο χρήστης του βιομηχανικού πλυντηρίου, καθώς και το πλαίσιο που καθορίζει την εμβέλεια του συστήματος.



ΣΧΗΜΑ XXXIX: ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΗΣ ΧΡΗΣΗΣ ΒΙΟΜΗΧΑΝΙΚΟΥ ΠΛΥΝΤΗΡΙΟΥ

Η βασική περίπτωση χρήσης είναι η διαδικασία μίας τυπικής πλύσης ρούχων. Σε αυτήν ο χρήστης τοποθετεί στο πλυντήριο τα ρούχα που χρήζουν πλύσης και τα απορρυπαντικά που απαιτούνται γι' αυτή, επιλέγει την θερμοκρασία νερού και το πρόγραμμα πλύσης, ασφαλίζει την πόρτα του πλυντηρίου και το σύστημα εκτελεί το πρόγραμμα πλύσης στην θερμοκρασία που έχει επιλεγεί. Η καταγραφή της περίπτωσης χρήσης γίνεται βήμα, βήμα :

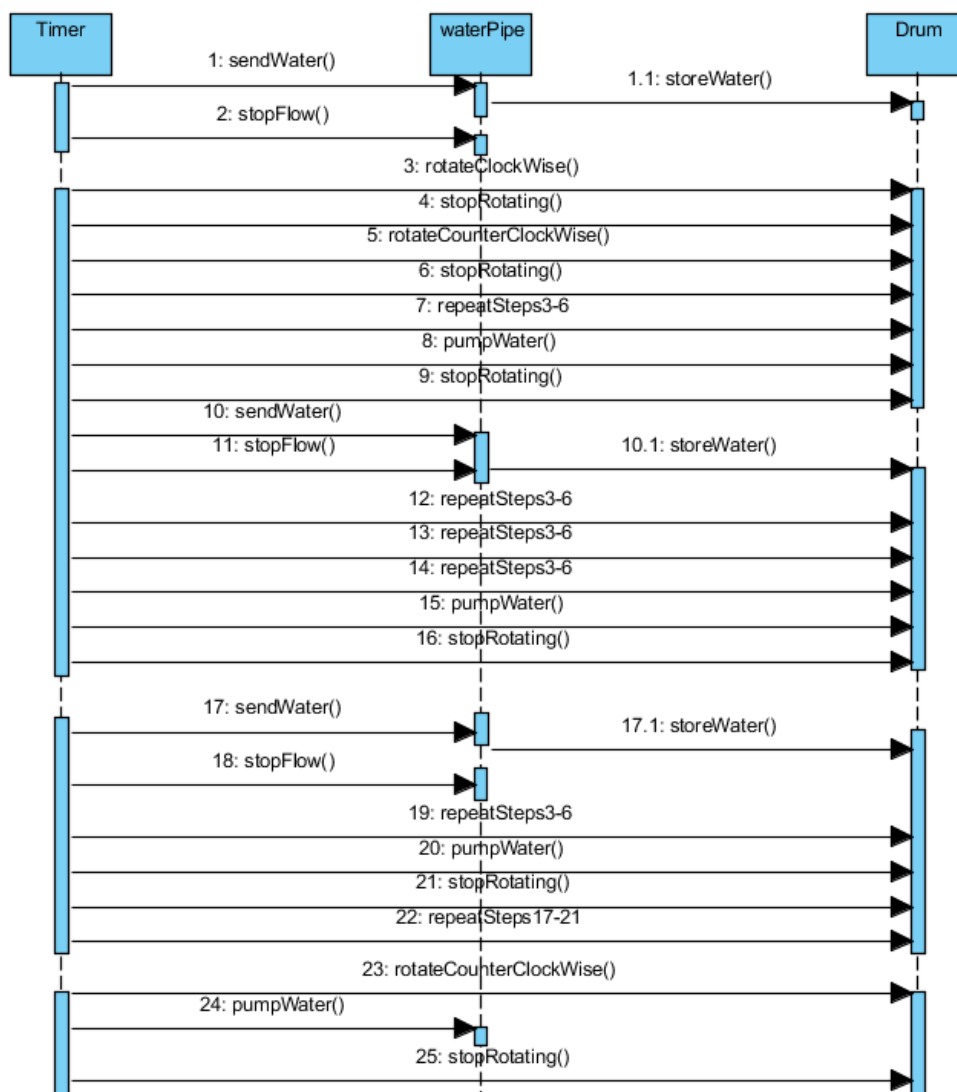
Περίπτωση Χρήσης ΠΧ1 : Πλύση Ρούχων

Χρήστης : Χρήστης Πλυντηρίου

Βασικό Σενάριο :

1. Ο χρήστης τοποθετεί τα ρούχα στο πλυντήριο.
2. Ο χρήστης ασφαλίζει την πόρτα του πλυντηρίου.
3. Ο χρήστης τοποθετεί τα απορρυπαντικά στο πλυντήριο.
4. Ο χρήστης επιλέγει πρόγραμμα πλύσης.
5. Ο χρήστης επιλέγει θερμοκρασία νερού πλύσης.
6. Το σύστημα εκτελεί το πρόγραμμα πλύσης που έχει επιλεγεί.
7. Ο χρήστης στο τέλος του προγράμματος αφαιρεί τα πλυμένα ρούχα από το πλυντήριο.

Αυτό αποτελεί το βασικό σενάριο της περίπτωσης χρήσης «Πλύση Ρούχων». Η περίπτωση χρήσης μπορεί να έχει και άλλα σενάρια, τα οποία θα ανακαλυφθούν σε επόμενες επαναλήψεις. Σε αυτό το σημείο προκύπτει το διάγραμμα ακολουθίας.



ΣΧΗΜΑ ΧΙ: ΔΙΑΓΡΑΜΜΑ ΑΚΟΛΟΥΘΙΑΣ ΧΡΗΣΗΣ ΒΙΟΜΗΧΑΝΙΚΟΥ ΠΛΥΝΤΗΡΙΟΥ

Στη συνέχεια θα προσδιορισθούν οι αφαιρέσεις του πραγματικού κόσμου που απαιτούνται για την κατασκευή του συστήματος, δηλαδή τα κύρια εννοιολογικά αντικείμενα που πρόκειται να συμμετάσχουν σε αυτό το σύστημα.

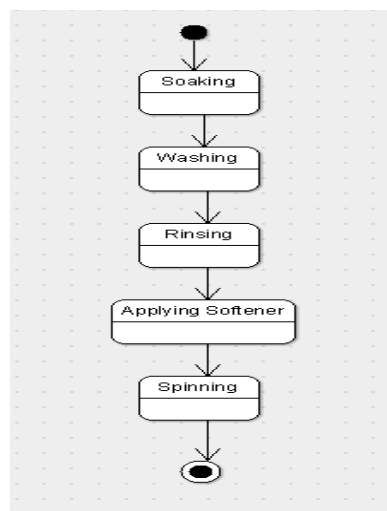
3.3.2 ΕΠΕΞΕΡΓΑΣΙΑ

Η φάση της επεξεργασίας για την περίπτωση του βιομηχανικού πλυντηρίου, θα χρειαστεί μόνο μία επανάληψη και αυτό γιατί οι προγραμματιζόμενοι λογικοί ελεγκτές δεν προγραμματίζονται χρησιμοποιώντας αντικειμενοστραφείς γλώσσες προγραμματισμού. Έτσι είναι αδύνατη η δημιουργία ενός διαγράμματος κλάσης και της μελέτης που απαιτείται γι' αυτήν, σε οποιαδήποτε φάση της λογικά ενοποιημένης διεργασίας.

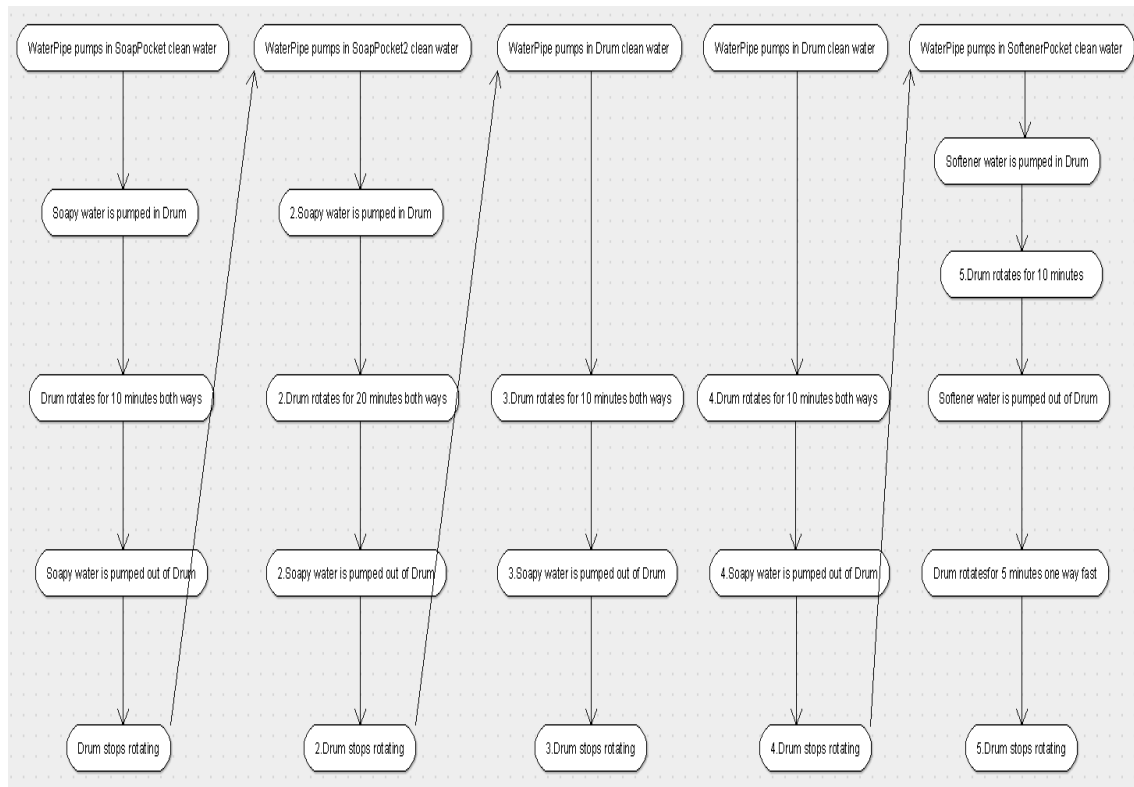
ΠΡΩΤΗ ΕΠΑΝΑΛΗΨΗ

Όσον αφορά την καταγραφή των απαιτήσεων δεν προστίθεται τίποτε σε αυτή την επανάληψη. Από πλευράς ανάλυσης θα κατασκευαστεί το διάγραμμα δραστηριοτήτων και το διάγραμμα καταστάσεων για την βασική περίπτωση χρήσης «Πλύση Ρούχων». Τα διαγράμματα αυτά θα συμβάλουν στην περαιτέρω κατανόηση του προς ανάπτυξη συστήματος.

Το διάγραμμα δραστηριοτήτων θα αποτυπώσει τις δραστηριότητες που συμβαίνουν για την περίπτωση χρήσης «Πλύση Ρούχων». Το διάγραμμα καταστάσεων θα περιγράψει τις καταστάσεις του συστήματος και το πώς αυτό αντιδρά σε συμβάντα.



ΣΧΗΜΑ ΧΛΙ: ΔΙΑΓΡΑΜΜΑ ΚΑΤΑΣΤΑΣΗΣ ΒΙΟΜΗΧΑΝΙΚΟΥ ΠΛΥΝΤΗΡΙΟΥ



ΣΧΗΜΑ XLII: ΔΙΑΓΡΑΜΜΑ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ ΒΙΟΜΗΧΑΝΙΚΟΥ ΠΛΥΝΤΗΡΙΟΥ

Από πλευράς εργασιών σχεδίασης, υλοποίησης και ελέγχου σε αυτή τη φάση, δεν υπάρχει τίποτε άλλο για πραγματοποίηση.

3.3.3 ΚΑΤΑΣΚΕΥΗ

Μία επανάληψη θα χρειαστεί και κατά τη φάση της κατασκευής. Τα παραγόμενα της φάσης επεξεργασίας και σύλληψης, δηλαδή όλα τα διαγράμματα, θα χρησιμοποιηθούν για την δημιουργία κώδικα στις επόμενες επαναλήψεις. Πρέπει να επισημανθεί ότι αν χρειαζόταν, ήταν δυνατή η παραγωγή κώδικα και στις προηγούμενες επαναλήψεις, όπως για παράδειγμα αν έπρεπε να δημιουργηθεί ένα μέρος του συστήματος για την παρουσίασή του στον πελάτη ή για επίλυση ιδίων αποριών. Η δημιουργία του κώδικα θα αναλυθεί στο επόμενο κεφάλαιο.

ΠΡΩΤΗ ΕΠΑΝΑΛΗΨΗ

Στην πρώτη επανάληψη όσον αφορά την καταγραφή απαιτήσεων, δεν υπάρχει η δυνατότητα για περαιτέρω εργασίες. Οι περιπτώσεις χρήσης έχουν καταγραφεί πλήρως κατά τις προηγούμενες επαναλήψεις. Από πλευράς ανάλυσης έχει κατασκευαστεί πλήρως το μοντέλο του πεδίου προβλήματος, και τα διαγράμματα καταστάσεων και δραστηριοτήτων. Το σύστημα έχει κατανοηθεί πλήρως οπότε και όσον αφορά την ανάλυση δεν υπάρχουν εργασίες να υλοποιηθούν. Έτσι στην επανάληψη αυτή θα υλοποιηθεί ο κώδικας με ταυτόχρονη εργασία ελέγχου, ώστε κάθε τμήμα κώδικα που υλοποιείται να ενσωματώνεται στο σύστημα και να δοκιμάζεται η λειτουργία του.

3.3.4 ΜΕΤΑΒΑΣΗ

Μία επανάληψη είναι αρκετή για να ολοκληρωθεί η φάση της μετάβασης και να παραδοθεί το έργο.

ΠΡΩΤΗ ΕΠΑΝΑΛΗΨΗ

Σε αυτή την επανάληψη το σύστημα λογισμικού που έχει κατασκευαστεί είναι μία δοκιμαστική έκδοση η οποία θα εγκατασταθεί σε ένα βιομηχανικό πλυντήριο και ταυτόχρονα θα γίνει εκπαίδευση των χρηστών. Θα δοθεί η ευκαιρία στους χρήστες να δοκιμάσουν το σύστημα για κάποιο χρονικό διάστημα ώστε να διαπιστωθεί η ικανοποίηση των απαιτήσεών τους. Κατά την περίοδο αυτή θα καταγράφονται οι όποιες παρατηρήσεις γίνονται από τους χρήστες που τελικά θα αποτελέσουν αντικείμενο για την βελτίωση του λογισμικού στην επανάληψη. Οι όποιες αλλαγές προκύψουν πρέπει να διασφαλιστεί ότι δεν θα επηρεάσουν το υπόλοιπο σύστημα. Αφού το σύστημα λογισμικού που κατασκευάστηκε ανταποκρίνεται στις απαιτήσεις των χρηστών / δοκιμαστών, θα εγκατασταθεί στην τελική του μορφή στο βιομηχανικό πλυντήριο, ενώ εγχειρίδια και εκπαιδευτικό υλικό θα διαμοιραστούν σε όλους τους χρήστες. Τέλος τα παραγόμενα της διαδικασίας ανάπτυξης του συστήματος αποθηκεύονται για χρήση σε κάποια πιθανή μελλοντική εξέλιξη του συστήματος.

ΚΕΦΑΛΑΙΟ 4

ΚΑΤΑΣΚΕΥΗ ΛΟΓΙΣΜΙΚΟΥ

Το κεφάλαιο τέσσερα αποτελεί την συνέχεια του προηγούμενου, αφού σε αυτό θα γίνει η κατασκευή του λογισμικού το οποίο αναλύθηκε και σχεδιάστηκε στο τρίτο. Πριν την δημιουργία του κώδικα που απαιτείται για την λειτουργία ενός βιομηχανικού πλυντηρίου, είναι απαραίτητη η αναφορά στο περιβάλλον χρησιμοποιήθηκε για την ανάπτυξη του συστήματος και την προσομοίωσή του, έτσι ώστε να μην απαιτείται η εφαρμογή και η δοκιμή του προγράμματος σε ένα πραγματικό βιομηχανικό πλυντήριο.

4.1 ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ TWIDOSUITE

Το TwidoSuite είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης για τον προγραμματισμό των εφαρμογών των προγραμματιζόμενων λογικά ελεγκτών. Το TwidoSuite αναπτύχθηκε από την Schneider Electric με στόχο το κοινό που δεν έχει πολύ χρόνο για την εκμάθηση ή την χρήση ενός λογισμικού για τον προγραμματισμό PLC.



EIKONA IX: ΣΗΜΑ TwidoSuite

Το TwidoSuite βοηθά και απλοποιεί όλες τις λειτουργίες με σκοπό την σημαντική μείωση του απαιτούμενου για τον προγραμματισμό χρόνου. Είναι ένα από τα λογισμικά προγραμματισμού τα οποία είναι οργανωμένα με βάση τις απαιτήσεις της ανάπτυξης ενός έργου. Επιπρόσθετα η διεπαφή του TwidoSuite είναι πολύ απλή και δημιουργεί τις συνθήκες για μία φιλική προς τον χρήστη, γρήγορη και αποτελεσματική χρήση.

Το TwidoSuite αποτελεί μία απλή προσέγγιση των ισχυρών γλωσσών Ladder και IL για την χρήση τους από τους προγραμματιστές των PLC. Έτσι, ανάλογα με τις ανάγκες του προγράμματος μπορεί να επιλεγθεί η γλώσσα που προσφέρει την καλύτερη απόδοση στο σύστημα, ή ακόμη και ο συνδυασμός τους.

Επιπλέον παρέχει την δυνατότητα αποσφαλμάτωσης με την χρήση κινούμενων γραφικών, ενώ πολύ σημαντικό πλεονέκτημα του TwidoSuite είναι πως καθιστά εφικτή την εξομίωση της λειτουργίας του προγράμματος που έχει δημιουργηθεί και με αυτό τον τρόπο την καλύτερη κατανόηση και παρακολούθησή του.



ΕΙΚΟΝΑ X: ΕΙΣΑΓΩΓΙΚΗ ΟΘΟΝΗ TwidoSuite

Το παγκόσμιο δίκτυο των συνεργατών του συστήματος TwidoSuite, παρέχει μία ποικιλία υπηρεσιών για τους χρήστες του, όπως είναι η υποστήριξη, η εκπαίδευση και ο προγραμματισμός εφαρμογών.

4.1.1 ΛΕΙΤΟΥΡΓΙΕΣ ΤΟΥ TWIDOSUITE

Στο TwidoSuite ένα έργο είναι εύκολα ελέγξιμο και διαχειρίσιμο. Η διαδικασία δημιουργίας ενός νέου έργου δεν θα μπορούσε να είναι απλούστερη, καθώς και η εύρεση ενός παλιού στον σκληρό δίσκο του ηλεκτρονικού υπολογιστή ή άμεσα από τον προγραμματιζόμενο λογικό ελεγκτή. Για λόγους εξοικονόμησης χρόνου κατά την επιλογή του έργου παρουσιάζονται όλες οι βασικές πληροφορίες του πριν ακόμη το άνοιγμά του, ώστε να γίνεται γρηγορότερη η εύρεση του επιθυμητού, καθώς και η άντληση των απαραίτητων γι' αυτό πληροφοριών χωρίς την ανάγκη περαιτέρω ανάπτυξης.

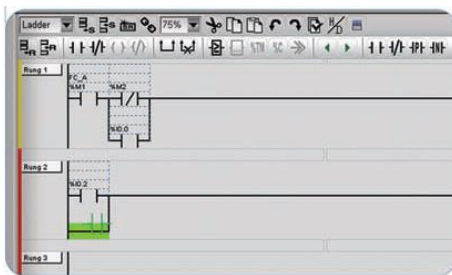
Μετά την δημιουργία ενός έργου, η συγγραφή του προγράμματος είναι πολύ απλή λόγω των δυνατοτήτων που παρέχει το TwidoSuite, και οι οποίες βασίζονται σε γραφικές περιγραφές του υλικού ενός προγραμματιζόμενου λογικού ελεγκτή καθώς και στο περιεχόμενο επικοινωνίας τους. Επίσης η τοποθέτηση ενός αντικειμένου μέσα στη ζώνη εργασίας είναι μία αυτοματοποιημένη διαδικασία επιλογής του αντικειμένου από ένα κατάλογο προϊόντων, και έπειτα μεταφοράς του στην επιθυμητή θέση.

Για την διευκόλυνση της εξερεύνησης του κάθε προγράμματος, υπάρχει η επιλογή της γρήγορης πρόσβασης σε κάθε τμήμα του προγράμματος, με την χρήση μίας λίστας που βρίσκεται στο πλάι του προγράμματος και στην οποία καταγράφονται όλα του τα τμήματα. Με την επιλογή του τμήματος ενδιαφέροντος, αυτόματα μεταφέρεται η εστίαση σε αυτό το σημείο.



ΕΙΚΟΝΑ XI: ΘΘΟΝΗ ΕΞΕΡΕΥΝΗΣΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ TwidoSuite

Η επιλογή της γλώσσας προγραμματισμού μπορεί να γίνει ανάμεσα στην Ladder και την IL οι οποίες έχουν αναλυθεί σε προηγούμενο κεφάλαιο. Ειδικά στην περίπτωση της Ladder η σύνταξή της έχει απλοποιηθεί για μεγαλύτερη παραγωγικότητα. Έτσι η επιλογή ενός rang και η τοποθέτηση επαφών, πηνίων και άλλων αντικειμένων της είναι απλή ενώ κατά τη διάρκεια της εργασίας είναι δυνατή η παρακολούθηση των αλλαγών που πραγματοποιούνται καθώς αυτές επισημαίνονται με κίτρινο ή κόκκινο χρώμα.



ΕΙΚΟΝΑ XII: ΟΘΟΝΗ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΕ TwidoSuite

Όσον αφορά την σύνδεση του προγράμματος με τον προγραμματιζόμενο λογικό ελεγκτή επαφίεται στην διάθεση και ανάγκη του προγραμματιστή. Με το TwidoSuite είναι δυνατή ακόμη και η επικοινωνία του μέσω ασύρματης σύνδεσης για την απόκτηση όσο το δυνατόν μεγαλύτερης ελευθερίας στις κινήσεις. Μετά την εξασφάλιση της επικοινωνίας είναι εφικτός ο έλεγχος του συνδεδεμένου εξοπλισμού αλλά και η επιλογή των δεδομένων που θα πρέπει να μεταφερθούν.

Για την αποσφαλμάτωση το TwidoSuite παρέχει ένα κινούμενο γραφικό του προγράμματος, ενώ εύκολα ο προγραμματιστής μπορεί να πραγματοποιήσει αλλαγές στο πρόγραμμα εν μέσω σύνδεσης. Υπάρχουν οι κατάλληλες οθόνες παρακολούθησης αλλαγών και λειτουργίας του προγράμματος, καθώς και ενσωματωμένοι πίνακες του έργου.



ΕΙΚΟΝΑ XIII: ΟΘΟΝΗ ΑΠΟΣΦΑΜΑΤΩΣΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ TwidoSuite

Τέλος για την δημιουργία ενός ξεκάθਾਰου και λεπτομερούς φακέλου του κάθε έργου, παρέχεται η δυνατότητα προσαρμογής του περιεχομένου και του σχεδιαγράμματος της σελίδας. Επίσης εφικτή είναι η δημιουργία ενός φακέλου HTML ο οποίος με την χρήση του Microsoft Word 2000 μπορεί να εξελιχθεί σε ένα εγχειρίδιο συντήρησης ή ένα οδηγό χρήσης.

4.2 ΠΡΟΓΡΑΜΜΑ ΓΙΑ ΤΗΝ ΛΕΙΤΟΥΡΓΙΑ ΒΙΟΜΗΧΑΝΙΚΟΥ ΠΛΥΝΤΗΡΙΟΥ

Στα πλαίσια σχηματισμού καλύτερης άποψης ενός προγραμματιζόμενου λογικού ελεγκτή, αλλά και των δυνατοτήτων που υπάρχουν όσον αφορά τον προγραμματισμό του, δημιουργήθηκε ένα πρόγραμμα σε γλώσσα Ladder, το οποίο ουσιαστικά καλύπτει την τυπική λειτουργία ενός βιομηχανικού πλυντηρίου.

Για λόγους διευκόλυνσης της παρουσίασης του προγράμματος, αλλά και λόγω του ότι το πρόγραμμα δεν δύναται να εφαρμοστεί σε πραγματικό προγραμματιζόμενο λογικό ελεγκτή, έπρεπε να γίνουν ορισμένες παραδοχές οι οποίες θα αναλυθούν παρακάτω.

Για την καλύτερη κατανόηση και αναπαράσταση της λειτουργίας του πλυντηρίου δημιουργήθηκε μία πολυμεσική διαδραστική εφαρμογή που μπορεί να χρησιμοποιηθεί παράλληλα με το πρόγραμμα ως προσομοιωτής. Βέβαια η διαδικασία της δημιουργίας της εφαρμογής αυτής υπερβαίνει τον σκοπό αυτής της εργασίας, και γι' αυτό το λόγο απλώς εμπεριέχεται στο δεύτερο παράρτημα μία υποτυπώδης ανάλυσή της συνοδευόμενη από στιγμιότυπά της.



ΕΙΚΟΝΑ ΧΙΥ: ΑΠΟΨΗ ΒΙΟΜΗΧΑΝΙΚΟΥ ΠΛΥΝΤΗΡΙΟΥ (ΠΑΡΟΥΣΙΑΣΗΣ)

4.2.1 ΓΕΝΙΚΕΣ ΠΛΗΡΟΦΟΡΙΕΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

Γενικά το πρόγραμμα εκτελεί όλες τις λειτουργίες ενός βιομηχανικού πλυντηρίου. Έτσι, μετά την τοποθέτηση των ρούχων στον κάδο και των απορρυπαντικών στις κατάλληλες θήκες, την επιλογή της επιθυμητής θερμοκρασίας πλύσης αλλά και του προγράμματος (για κανονικά ή για ευαίσθητα ρούχα), το κλείσιμο της πόρτας και την ενεργοποίηση της πλύσης με το πάτημα του κουμπιού εκκίνησης, ξεκινάει η διαδικασία της πρόπλυσης, η οποία ακολουθείται από μία αποστράγγιση. Έπειτα συνεχίζει με την κυρίως πλύση και στο τέλος αυτής με μία ακόμη αποστράγγιση, ενώ ακολουθούν δύο ξεβγάλματα με αντίστοιχες αποστραγγίσεις. Τέλος γίνεται η εφαρμογή του μαλακτικού και μία επιπλέον αποστράγγιση, καθώς και το στύψιμο, με το οποίο τελειώνει και η λειτουργία του πλυντηρίου. Σε όλα αυτά τα στάδια υπάρχει η ένδειξη λειτουργίας με την χρήση μίας λυχνίας.

Σημαντική επίσης είναι και η αναφορά της πρόβλεψης για την πτώση τάσης που είναι πιθανόν να συμβεί κατά τη διάρκεια της πλύσης, με την επαναφορά της οποίας η λειτουργία συνεχίζει κανονικά από το σημείο που σταμάτησε. Ανάλογη πρόβλεψη υπήρξε και για την χειροκίνητη διακοπή της λειτουργίας του πλυντηρίου από τον ίδιο τον χρήστη με το πάτημα του κουμπιού παύσης, οπότε το πλυντήριο αρχικά αποστραγγίζει το νερό που υπάρχει εκείνη την στιγμή στον κάδο του και απασφαλίζει την πόρτα του, ενώ έπειτα από το πάτημα του κουμπιού εκκίνησης επανασφαλίζει την πόρτα του και ξαναγεμίζει τον κάδο με την απαιτούμενη ποσότητα νερού, συνεχίζοντας την πλύση από το σημείο που σταμάτησε. Όλα τα παραπάνω στάδια αναλύονται ως προς τον τρόπο προγραμματισμού τους στην συνέχεια.

Το πρόγραμμα αφορά ένα πλυντήριο το οποίο διαθέτει τρεις κινητήρες, αφού έχει τρεις ταχύτητες στην περιστροφή του κάδου (κανονική - αργή, μεσαία και γρήγορη). Επίσης όπως όλα τα πλυντήρια έχει τρεις θέσεις τοποθέτησης των απαραίτητων για την πλύση απορρυπαντικών (αριστερή θέση - πρόπλυσης, μεσαία θέση - κυρίως πλύσης, δεξιά θέση - μαλακτικού). Συγχρόνως διαθέτει τρεις θέσεις ένδειξης της στάθμης του νερού που βρίσκεται στον κάδο του πλυντηρίου (της αποστράγγισης στην οποία δεν υπάρχει νερό στον κάδο, του γεμίσματος της πλύσης στην οποία ο κάδος έχει την κατάλληλη ποσότητα νερού για την πρόπλυση, την κυρίως πλύση και την εφαρμογή του μαλακτικού, και τέλος του γεμίσματος για το ξέβγαλμα των ρούχων στην οποία υπάρχει το απαιτούμενο νερό για την διαδικασία του ξεβγάλματος, που είναι περισσότερο από αυτό που απαιτεί η πλύση).

Με την παραπάνω ανάλυση της λειτουργίας του πλυντηρίου μπορούν εύκολα να εντοπιστούν οι είσοδοι και οι έξοδοι που χρησιμοποιήθηκαν για την συγγραφή του συγκεκριμένου προγράμματος, το οποίο σημειώνεται ξανά πως είναι ενδεικτικό, και οι οποίες αναφέρονται παρακάτω μαζί με τις διευθύνσεις που χρησιμοποιούν, όπως και τα υπόλοιπα στοιχεία.

4.2.2 ΛΕΠΤΟΜΕΡΕΙΕΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

Στην συγγραφή του προγράμματος έγινε χρήση των περισσοτέρων στοιχείων της γλώσσας LADDER, όπως είναι ο τυμπανοειδής ελεγκτής, το χρονικό, ο απαριθμητής, οι εισόδοι, οι έξοδοι και οι μνήμες. Ο τρόπος ονομασίας όλων αυτών έχει ήδη αναλυθεί στο πρώτο κεφάλαιο, γι' αυτό δεν κρίνεται σκόπιμη η επανάληψη αυτών των πληροφοριών.

Ξεκινώντας από τις εισόδους, θα ήταν ίσως χρήσιμο να αναφερθεί πως μπορούν να συναντηθούν με την μορφή ενός κουμπιού, ενός τερματικού, ενός θερμοστάτη, ενός φωτοκύτταρου. Γενικά ως είσοδος μπορεί να θεωρηθεί οτιδήποτε είναι σε θέση να δώσει ένα σήμα στο PLC για την έναρξη ή την λήξη μίας διαδικασίας, που να μπορεί να αξιοποιηθεί από αυτό ανάλογα. Συγκεκριμένα στο πρόγραμμα, χρησιμοποιήθηκαν οι παρακάτω εισόδοι, οι οποίες παρατίθενται ονομαστικά, αλλά και με την διεύθυνσή τους.

ΠΙΝΑΚΑΣ VII: ΠΙΝΑΚΑΣ ΕΙΣΟΔΩΝ ΠΡΟΓΡΑΜΜΑΤΟΣ

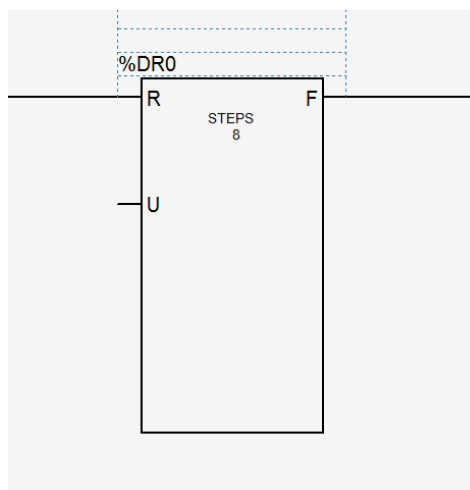
Table of inputs								
Used	Address	Symbol	Used By	Filtering	Latch?	Run/Stop?	Deactivation	High Priority
<input checked="" type="checkbox"/>	%I0.0	ΜΠΟΥΤΟΝ_START	user logic	3 ms	<input type="checkbox"/>	<input type="checkbox"/>		
<input checked="" type="checkbox"/>	%I0.1	ΤΕΡΜΑΤΙΚΟΣ_ΠΟΡΤΑΣ	user logic	3 ms	<input type="checkbox"/>	<input type="checkbox"/>		
<input checked="" type="checkbox"/>	%I0.2	ΜΠΟΥΤΟΝ_30C	user logic	3 ms	<input type="checkbox"/>	<input type="checkbox"/>	Not used	<input type="checkbox"/>
<input checked="" type="checkbox"/>	%I0.3	ΜΠΟΥΤΟΝ_40C	user logic	3 ms	<input type="checkbox"/>	<input type="checkbox"/>	Not used	<input type="checkbox"/>
<input checked="" type="checkbox"/>	%I0.4	ΜΠΟΥΤΟΝ_60C	user logic	3 ms	<input type="checkbox"/>	<input type="checkbox"/>	Not used	<input type="checkbox"/>
<input checked="" type="checkbox"/>	%I0.5	ΜΠΟΥΤΟΝ_90C	user logic	3 ms	<input type="checkbox"/>	<input type="checkbox"/>	Not used	<input type="checkbox"/>
<input checked="" type="checkbox"/>	%I0.6	ΜΠΟΥΤΟΝ_ΚΑΝΟΝΙΚΟΥ_ΠΡΟΓΡΑΜΜΑΤΟΣ	user logic	3 ms	<input type="checkbox"/>	<input type="checkbox"/>		
<input checked="" type="checkbox"/>	%I0.7	ΜΠΟΥΤΟΝ_ΕΥΑΙΣΘΗΤΟΥ_ΠΡΟΓΡΑΜΜΑΤΟΣ	user logic	3 ms	<input type="checkbox"/>	<input type="checkbox"/>		
<input checked="" type="checkbox"/>	%I0.8	ΤΕΡΜΑΤΙΚΟΣ_FULL	user logic	No Filter	<input type="checkbox"/>	<input type="checkbox"/>		
<input checked="" type="checkbox"/>	%I0.9	ΤΕΡΜΑΤΙΚΟΣ_ΠΛΥΣΗΣ	user logic	No Filter	<input type="checkbox"/>	<input type="checkbox"/>		
<input checked="" type="checkbox"/>	%I0.10	ΤΕΡΜΑΤΙΚΟΣ_EMPTY	user logic	No Filter	<input type="checkbox"/>	<input type="checkbox"/>		
<input checked="" type="checkbox"/>	%I0.11	ΜΠΟΥΤΟΝ_STOP	user logic	No Filter	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	%I0.12			No Filter	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	%I0.13			No Filter	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	%I0.14			No Filter	<input type="checkbox"/>	<input type="checkbox"/>		
<input checked="" type="checkbox"/>	%I0.15	ΠΤΩΣΗ_ΤΑΣΗΣ	user logic	No Filter	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	%I0.16			No Filter	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	%I0.17			No Filter	<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	%I0.18			No Filter	<input type="checkbox"/>	<input type="checkbox"/>		

Οι έξοδοι, είναι δυνατόν να έχουν την μορφή ενός ρελέ, μίας λυχνίας, μίας αντλίας, μίας βάνας, ενός κινούμενου μέρους όπως είναι ένας κινητήρας, αλλά και οποιαδήποτε άλλη μορφή που αφού δεχθεί το κατάλληλο σήμα από το PLC, ξεκινάει μία διαδικασία. Στο πρόγραμμα που έχει συγγραφεί, έχουν χρησιμοποιηθεί οι έξοδοι που αναφέρονται ονομαστικά και βάσει διεύθυνσης παρακάτω.

ΠΙΝΑΚΑΣ VIII: ΠΙΝΑΚΑΣ ΕΞΟΔΩΝ ΠΡΟΓΡΑΜΜΑΤΟΣ

Table of outputs				
Used	Address	Symbol	Status?	Used By
<input checked="" type="checkbox"/>	%Q0.0	ΚΙΝΗΤΗΤΡΑΣ_2_ΑΡΙΣΤΕΡΑ_ΜΕΣΑΙΑ		user logic
<input checked="" type="checkbox"/>	%Q0.1	ΚΙΝΗΤΗΤΡΑΣ_3_ΑΡΙΣΤΕΡΑ_ΓΡΗΓΟΡΑ	<input type="checkbox"/>	user logic
<input checked="" type="checkbox"/>	%Q0.2	ΚΙΝΗΤΗΤΡΑΣ_1_ΔΕΞΙΑ_ΚΑΝΟΝΙΚΑ	<input type="checkbox"/>	user logic
<input checked="" type="checkbox"/>	%Q0.3	ΚΙΝΗΤΗΤΡΑΣ_1_ΑΡΙΣΤΕΡΑ_ΚΑΝΟΝΙΚΑ	<input type="checkbox"/>	user logic
<input type="checkbox"/>	%Q0.4			
<input checked="" type="checkbox"/>	%Q0.5	ΡΕΛΑΥ_ΑΠΟΡΡΥΠΑΝΤΙΚΟΥ_ΑΡΙΣΤΕΡΑ		user logic
<input checked="" type="checkbox"/>	%Q0.6	ΡΕΛΑΥ_ΑΠΟΡΡΥΠΑΝΤΙΚΟΥ_ΜΕΣΑΙΑ		user logic
<input checked="" type="checkbox"/>	%Q0.7	ΡΕΛΑΥ_ΜΑΛΑΚΤΙΚΟΥ		user logic
<input checked="" type="checkbox"/>	%Q0.8	ΑΠΟΣΤΡΑΓΓΙΣΗ		user logic
<input checked="" type="checkbox"/>	%Q0.9	ΓΕΜΙΣΜΑ_ΠΛΥΣΗΣ		user logic
<input checked="" type="checkbox"/>	%Q0.10	ΓΕΜΙΣΜΑ_FULL		user logic
<input type="checkbox"/>	%Q0.11			
<input type="checkbox"/>	%Q0.12			
<input type="checkbox"/>	%Q0.13			
<input type="checkbox"/>	%Q0.14			
<input checked="" type="checkbox"/>	%Q0.15	ΛΥΧΝΙΑ_ΛΕΙΤΟΥΡΓΙΑΣ		user logic

Έχει χρησιμοποιηθεί επίσης ένας τυμπανοειδής ελεγκτής (drum controller) για την εναλλαγή της διεύθυνσης περιστροφής του κινητήρα 1, η οποία γίνεται με παρεμβάλλουσα παύση. Καθώς ο τυμπανοειδής ελεγκτής αποτελεί κεντρικό στοιχείο στο συγκεκριμένο πρόγραμμα, είναι σημαντικό να αναλυθεί. Έτσι δίνεται παρακάτω ο συμβολισμός του, όπου στην είσοδο R (Reset) ουσιαστικά τοποθετούνται οι συνθήκες που απαιτείται να ικανοποιηθούν έτσι ώστε να επιστρέψει στο μηδενικό βήμα ο ελεγκτής. Στην είσοδο U (Up) βρίσκονται οι συνθήκες, η ικανοποίηση των οποίων σηματοδοτεί την αύξηση του βήματος στον ελεγκτή. Τέλος στην έξοδο F (Full) βρίσκεται η επαφή που ενεργοποιείται όταν ο ελεγκτής βρίσκεται στο τελευταίο βήμα του.



ΣΧΗΜΑ XXXI: ΣΥΜΒΟΛΙΣΜΟΣ ΤΥΜΠΑΝΟΕΙΔΟΥΣ ΕΛΕΓΚΤΗ

Όπως προαναφέρθηκε, για την υλοποίηση του προγράμματος που συγγράφηκε για την λειτουργία ενός βιομηχανικού πλυντηρίου, χρησιμοποιήθηκε ένας τυμπανοειδής ελεγκτής με την παρακάτω μορφή.

ΠΙΝΑΚΑΣ ΙΧ: ΒΗΜΑΤΑ ΚΑΙ BIT ΕΛΕΓΧΟΥ ΤΥΜΠΑΝΟΕΙΔΟΥΣ ΕΛΕΓΚΤΗ

Number of steps : 5

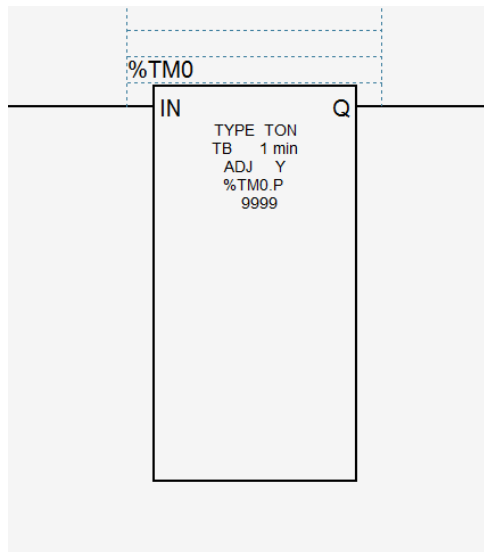
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Step 0:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Step 1:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Step 2:	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Step 3:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Step 4:	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Step 5:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Step 6:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Step 7:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Outputs:%Q??.? or %M?

Bit 0:	<input type="text" value="%M8"/>	Bit 6:	<input type="text"/>	Bit 12:	<input type="text"/>
Bit 1:	<input type="text" value="%M9"/>	Bit 7:	<input type="text"/>	Bit 13:	<input type="text"/>
Bit 2:	<input type="text" value="%M31"/>	Bit 8:	<input type="text"/>	Bit 14:	<input type="text"/>
Bit 3:	<input type="text"/>	Bit 9:	<input type="text"/>	Bit 15:	<input type="text"/>
Bit 4:	<input type="text"/>	Bit 10:	<input type="text"/>		
Bit 5:	<input type="text"/>	Bit 11:	<input type="text"/>		

Στο σχήμα είναι εύκολο να παρατηρηθεί πως ορισμένα τετράγωνα έχουν τσεκαριστεί, και άλλα όχι. Τα σημειωμένα τετράγωνα έχουν ως αποτέλεσμα την ενεργοποίηση του συγκεκριμένου bit ελέγχου στο αντίστοιχο βήμα, και άρα της αντίστοιχης μνήμης ή εξόδου. Αντίστοιχα τα ελεύθερα τετράγωνα σε κάθε βήμα ισοδυναμούν με μη ενεργοποίηση των δηλωμένων εξόδων ή μνημών. Έτσι εδώ και με την ταυτόχρονη ανάγνωση του προγράμματος, γίνεται εύκολα αντιληπτό πως στο βήμα 1 ενεργοποιείται η δεξιόστροφη περιστροφή με κανονική ταχύτητα του κινητήρα 1, στο βήμα 2 λαμβάνει χώρα μία παύση, στο βήμα 3 ενεργοποιείται η αριστερόστροφη περιστροφή με κανονική ταχύτητα του κινητήρα, και τέλος στο βήμα 4 επαναλαμβάνεται η παύση.

Για την μέτρηση του χρόνου περιστροφής των κινητήρων, αλλά και των παύσεων που τις διαδέχονται, χρησιμοποιήθηκαν δύο χρονικά. Γενικά ένα χρονικό μπορεί να είναι τύπου TON (Timer On Delay), TOF (Timer Off Delay) και TP (Timer Pulse). Η μορφή των χρονικών παριστάνεται στο παρακάτω σχήμα όπου στην είσοδο IN (Input) βρίσκονται οι συνθήκες η ικανοποίηση των οποίων σηματοδοτεί την έναρξη της λειτουργίας του χρονικού. Αντίστοιχα στην έξοδο Q (Output) υπάρχει η επαφή που ενεργοποιείται με την κάθε αύξηση της μονάδας μέτρησης του χρόνου που έχει επιλεγεί από τον χρήστη.



ΣΧΗΜΑ XLIV: ΣΥΜΒΟΛΙΣΜΟΣ ΧΡΟΝΙΚΟΥ

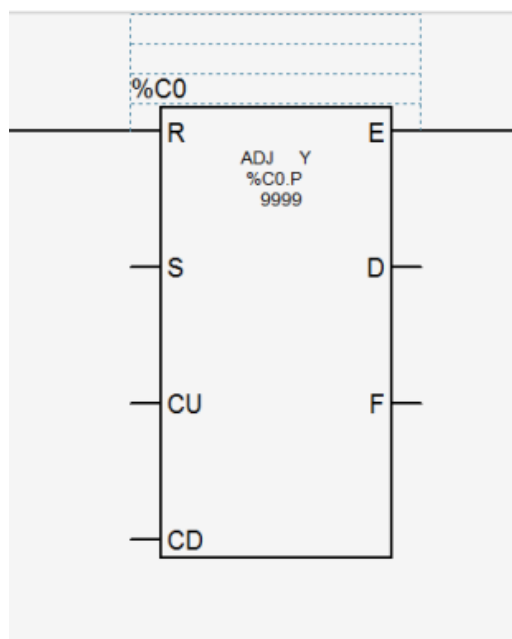
Στο συγκεκριμένο πρόγραμμα όπως αναφέρθηκε και προηγουμένως, χρησιμοποιήθηκαν δύο χρονικά τύπου TP. Το πρώτο χρονικό έχει ρυθμιστεί στη μέτρηση τεσσάρων δευτερολέπτων, και χρησιμοποιείται στην χρονομέτρηση της περιστροφής που υλοποιεί ο κινητήρας 1 καλούμενος από τον τυμπανοειδή ελεγκτή, ενώ μετρά το ίδιο χρονικό διάστημα και για την περιστροφή των κινητήρων 2 και 3 κατά την διαδικασία του συψίματος. Το δεύτερο χρονικό καλείται αποκλειστικά από τον τυμπανοειδή ελεγκτή για την χρονομέτρηση ενός δευτερολέπτου που έχει ορισθεί η διάρκεια της παύσης που παρεμβάλλεται μεταξύ της εναλλαγής της διεύθυνσης περιστροφής του κινητήρα 1. Οι διευθύνσεις που χρησιμοποιούν τα χρονικά αυτά στο πρόγραμμα, ο τύπος του κάθε ενός, το χρονικό διάστημα που χρονομετρούν και η μονάδα μέτρησης τους παρουσιάζονται παρακάτω.

ΠΙΝΑΚΑΣ X: ΧΡΟΝΙΚΑ ΠΡΟΓΡΑΜΜΑΤΟΣ

Used	%TM	Symbol	Type	Base	Preset	Adjustable
<input checked="" type="checkbox"/>	%TM0		TP	1 s	4	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	%TM1		TP	1 s	1	<input checked="" type="checkbox"/>
<input type="checkbox"/>	%TM2		TP	1 min	9999	<input checked="" type="checkbox"/>
<input type="checkbox"/>	%TM3		TP	1 min	9999	<input checked="" type="checkbox"/>
<input type="checkbox"/>	%TM4		TON	1 min	9999	<input checked="" type="checkbox"/>
<input type="checkbox"/>	%TM5		TON	1 min	9999	<input checked="" type="checkbox"/>
<input type="checkbox"/>	%TM6		TON	1 min	9999	<input checked="" type="checkbox"/>

Τέλος είναι απαραίτητη η αναφορά στους τέσσερις απαριθμητές που χρησιμοποιήθηκαν για την συγγραφή του προγράμματος. Γενικά υπάρχουν τρεις τύποι στους οποίους μπορεί να καταταχθεί ένας απαριθμητής. Έτσι μπορεί να

είναι τύπου Counter Up που μετράει γεγονότα αθροιστικά, Counter Down ο οποίος μετράει γεγονότα αφαιρετικά, και Counter Up & Down για την μέτρηση ταυτοχρόνως δύο γεγονότων με αθροιστικό και αφαιρετικό τρόπο. Η κατάσταση στον ανάλογο τύπο γίνεται αυτόματα, με την επιλογή της κατάλληλης εισόδου CU και CD. Οι εισόδους που φαίνονται στο σχήμα του απαριθμητή είναι η R (Reset) όπου τοποθετούνται οι συνθήκες η ικανοποίηση των οποίων σηματοδοτεί την απόδοση της μηδενικής τιμής στον απαριθμητή, η S (Set) όπου βρίσκονται οι συνθήκες οι οποίες όταν ικανοποιούνται ο απαριθμητής συνεχίζει να μετράει όπως έχει οριστεί, και οι CU (Up Count Input) και CD (Down Count Input) οι οποίες ουσιαστικά ορίζουν όπως αναφέρθηκε και παραπάνω αν ο απαριθμητής θα αυξάνει ή θα μειώνει την τιμή του αντίστοιχα. Ένας απαριθμητής διαθέτει τρεις εξόδους. Στην έξοδο E (Empty) τοποθετείται η επαφή η που ενεργοποίησή της δηλώνει την αφαιρετική μέτρηση του απαριθμητή, ενώ αντίθετα στην έξοδο F (Full) βρίσκεται η επαφή που η ενεργοποίησή της αντιστοιχεί σε αθροιστική μέτρηση από τον απαριθμητή. Τέλος στην έξοδο D (Done) υπάρχει η επαφή της οποίας η ενεργοποίηση δίνει το αποτέλεσμα κάθε αύξησης της τιμής του απαριθμητή.



ΣΧΗΜΑ XLV: ΣΥΜΒΟΛΙΣΜΟΣ ΑΠΑΡΙΘΜΗΤΗ

Όπως αναφέρθηκε και προηγουμένως έγινε χρήση τεσσάρων απαριθμητών. Ο πρώτος απαριθμητής μετράει ως το τρία και ενεργοποιείται

μόνο στην περίπτωση που πρέπει να εκτελεστεί η διαδικασία του τυμπανοειδούς ελεγκτή περισσότερες της μία φορές. Ο δεύτερος απαριθμητής μετράει ως το πέντε, και σηματοδοτεί την μεταπήδηση των διαδικασιών του πλυντηρίου από βήμα σε βήμα όπως αυτά έχουν οριστεί. Ο τρίτος απαριθμητής μετράει ως το δύο, και σηματοδοτεί την ακαριαία εναλλαγή των κινητήρων που λαμβάνουν μέρος στο στύψιμο. Ο τέταρτος και τελευταίος απαριθμητής έχει ως λειτουργία του την μέτρηση ως το ένα, αυτό γίνεται ώστε να καθίσταται δυνατή η επανάληψη της χρήσης του κουμπιού stop όσες φορές αυτό κρίνεται απαραίτητο. Οι απαριθμητές αυτοί παρουσιάζονται με τις διευθύνσεις και το εύρος των τιμών μέτρησής τους παρακάτω.

ΠΙΝΑΚΑΣ XI: ΑΠΑΡΙΘΜΗΤΕΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

Used	%C	Symbol	Preset	Adjustable
<input checked="" type="checkbox"/>	%C0		3	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	%C1		5	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	%C2		2	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	%C3		1	<input checked="" type="checkbox"/>
<input type="checkbox"/>	%C4		9999	<input checked="" type="checkbox"/>
<input type="checkbox"/>	%C5		9999	<input checked="" type="checkbox"/>
<input type="checkbox"/>	%C6		9999	<input checked="" type="checkbox"/>
<input type="checkbox"/>	%C7		9999	<input checked="" type="checkbox"/>
<input type="checkbox"/>	%C8		9999	<input checked="" type="checkbox"/>
<input type="checkbox"/>	%C9		9999	<input checked="" type="checkbox"/>
<input type="checkbox"/>	%C10		9999	<input checked="" type="checkbox"/>

Σημαντική θέση στο πρόγραμμα ωστόσο έχει η χρήση των εσωτερικών bits τα οποία ονομάζονται διαφορετικά και μνήμες. Οι μνήμες ουσιαστικά είναι περιοχές αποθήκευσης που διατίθενται στους χρήστες, ώστε αυτοί να μπορούν να ελέγχουν κάποιες βοηθητικές λειτουργίες του προγράμματος. Η ονομασία τους είναι της μορφής Mx όπου M είναι το αρκτικόλεξο της λέξης Memory ενώ το x συμβολίζει τον αριθμό της μνήμης. Είναι τρομερά εύχρηστες, καθώς είναι δυνατή η διαρκής κλήση τους στο πρόγραμμα. Η ονομασία των μνημών που χρησιμοποιήθηκαν με την διεύθυνσή της η κάθε μία δίνονται στον πίνακα παρακάτω.

ΠΙΝΑΚΑΣ ΧΙΙ: ΜΝΗΜΕΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

%M0	ΜΝΗΜΗ_START
%M1	ΜΝΗΜΗ_30C
%M2	ΜΝΗΜΗ_40C
%M3	ΜΝΗΜΗ_60C
%M4	ΜΝΗΜΗ_900C
%M5	ΚΑΝΟΝΙΚΟ_ΠΡΟΓΡΑΜΜΑ
%M6	ΕΥΑΙΣΘΗΤΟ_ΠΡΟΓΡΑΜΜΑ
%M7	ΜΝΗΜΗ_ΠΛΗΡΩΣΗΣ
%M8	DRUM_MOTOR_1D
%M9	DRUM_MOTOR_1A
%M10	ΜΝΗΜΗ_ΤΕΛΟΥΣ_ΕΡΓΑΣΙΑΣ
%M11	ΜΝΗΜΗ_ΜΠΟΥΤΟΝ_STOP
%M12	TRIGER_ΑΠΟΣΤΡΑΓΓΙΣΗΣ
%M13	ΜΝΗΜΗ_ΤΕΡΜΑΤΙΚΟΥ_ΠΟΡΤΑΣ
%M14	ΜΝΗΜΗ_ΤΕΡΜΑΤΙΚΟΥ_FULL
%M15	ΜΝΗΜΗ_ΤΕΡΜΑΤΙΚΟΥ_ΠΛΥΣΗΣ
%M16	ΜΝΗΜΗ_ΤΕΡΜΑΤΙΚΟΥ_EMPTY
%M18	DRUM_STEP_1
%M19	DRUM_STEP_2
%M20	TIMER_0_ΛΕΙΤΟΥΡΓΙΑΣ
%M21	TIMER_1_ΠΑΥΣΗΣ
%M25	DR_FULL
%M26	DRUM_STEP_3
%M27	DRUM_STEP_4
%M30	REPEAT_10
%M31	DRUM_MOTOR_PAUSE
%M33	C1_DONE
%M34	REPEAT_20
%M35	ΒΗΜΑ_1
%M40	C0_DONE
%M41	DRUM_RESET
%M42	ΒΗΜΑ_2
%M43	ΒΗΜΑ_3
%M44	ΒΗΜΑ_4
%M45	ΒΗΜΑ_5
%M46	C2_DONE
%M50	C3_DONE
%M51	ΜΝΗΜΗ_ΠΤΩΣΗ_ΤΑΣΗΣ

4.2.3 ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ

Το πρόγραμμα που έχει δημιουργηθεί για την λειτουργία του βιομηχανικού πλυντηρίου μπορεί εύκολα να παρασταθεί από ένα σύνολο ενεργών κυκλωμάτων και έναν πίνακα δεδομένων. Ο πίνακας αυτός σκιαγραφεί την ακολουθία των υπολειτουργιών που λαμβάνουν χώρα κατά την διάρκεια της πλύσης. Τα ενεργά κυκλώματα και ο πίνακας δεδομένων παρουσιάζονται παρακάτω.

ΠΙΝΑΚΑΣ XIII: ΠΙΝΑΚΑΣ ΕΝΕΡΓΩΝ ΚΥΚΛΩΜΑΤΩΝ

1	ΘΕΡΜΟΚΡΑΣΙΑ 30°C
2	ΘΕΡΜΟΚΡΑΣΙΑ 40°C
3	ΘΕΡΜΟΚΡΑΣΙΑ 60°C
4	ΘΕΡΜΟΚΡΑΣΙΑ 90°C
5	ΑΠΟΡΡΥΠΑΝΤΙΚΟ ΑΡΙΣΤΕΡΗ ΘΕΣΗ
6	ΑΠΟΡΡΥΠΑΝΤΙΚΟ ΜΕΣΑΙΑ ΘΕΣΗ
7	ΑΠΟΡΡΥΠΑΝΤΙΚΟ ΔΕΞΙΑ ΘΕΣΗ (ΜΑΛΑΚΤΙΚΟ)
8	ΚΙΝΗΤΗΡΑΣ 1 ΔΕΞΙΑ ΚΑΝΟΝΙΚΑ
9	ΚΙΝΗΤΗΡΑΣ 1 ΑΡΙΣΤΕΡΑ ΚΑΝΟΝΙΚΑ
10	ΚΙΝΗΤΗΡΑΣ 2 ΑΡΙΣΤΕΡΑ ΜΕΣΑΙΑ
11	ΚΙΝΗΤΗΡΑΣ 3 ΑΡΙΣΤΕΡΑ ΓΡΗΓΟΡΑ
12	ΒΑΛΒΙΔΑ ΠΛΗΡΩΣΗΣ ΠΛΥΣΗΣ
13	ΒΑΛΒΙΔΑ ΠΛΗΡΩΣΗΣ FULL (ΞΕΒΓΑΛΜΑΤΟΣ)
14	ΑΝΤΛΙΑ ΑΠΟΣΤΡΑΓΓΙΣΗΣ
15	ΛΥΧΝΙΑ ΛΕΙΤΟΥΡΓΙΑΣ

Η διαδικασία που πραγματοποιεί το πρόγραμμα χωρίζεται σε έξι βήματα (την πρόπλυση, την κύρια πλύση, το πρώτο και το δεύτερο ξέβγαλμα, την εφαρμογή μαλακτικού και το στύψιμο που διαφέρει στο κανονικό από το ευαίσθητο πρόγραμμα) τα οποία έχουν αντιστοιχιστεί στον απαριθμητή (C1). Σε πολλά από αυτά τα βήματα εκτελείται μία συγκεκριμένη διαδικασία που έχει οριοθετηθεί στο drum controller και η οποία έχει ήδη αναλυθεί.

Τα παραπάνω ενεργά κυκλώματα χρησιμοποιούνται στον πίνακα δεδομένων που ακολουθεί, με τον αριθμό στον οποίο αντιστοιχίζεται το καθένα έτσι ώστε να φαίνεται ποια είναι ενεργά κατά τη διάρκεια της κάθε υπολειτουργίας.

ΠΙΝΑΚΑΣ XIV: ΠΙΝΑΚΑΣ ΔΕΔΟΜΕΝΩΝ

ΛΕΙΤΟΥΡΓΙΑ ΠΛΥΝΤΗΡΙΟΥ		ΒΗΜΑ ΑΠΑΡΙΘΜΗΤΗ (C1)	ΒΗΜΑ DRUM (C2)	ΒΗΜΑ ΑΠΑΡΙΘΜΗΤΗ ΣΤΥΨΙΜΑΤΟΣ	ΕΝΕΡΓΑ ΚΥΚΛΩΜΑΤΑ									
					1	4	5	6	7	8	9	10	11	12
ΠΡΟΠΛΥΣΗ	ΓΕΜΙΣΜΑ (ΠΕΡΙΣΤΡΟΦΗ ΠΑΥΣΗ ΠΕΡΙΣΤΟΦΗ ΠΑΥΣΗ) * 2	0	-	0	1-4	5					12		15	
			1		8	15								
			2			15								
			3		9	15								
4		15												
ΚΥΡΙΑ ΠΛΥΣΗ	ΣΤΡΑΓΓΙΣΜΑ (ΓΕΜΙΣΜΑ ΠΕΡΙΣΤΡΟΦΗ ΠΑΥΣΗ ΠΕΡΙΣΤΟΦΗ ΠΑΥΣΗ) * 3	1	-	0	1-4		6			8		12	14	15
			1		8	15								
			2			15								
			3		9	15								
4		15												
1 ^ο ΚΑΙ 2 ^ο ΞΕΠΛΥΜΑ	ΣΤΡΑΓΓΙΣΜΑ ΓΕΜΙΣΜΑ ΠΕΡΙΣΤΡΟΦΗ ΠΑΥΣΗ ΠΕΡΙΣΤΟΦΗ ΠΑΥΣΗ	2-3	-	0	1-4				8			13	14	15
			1		8	15								
			2			15								
			3		9	15								
4		15												
ΜΑΛΑΚΤΙΚΟ	ΣΤΡΑΓΓΙΣΜΑ ΓΕΜΙΣΜΑ ΠΕΡΙΣΤΡΟΦΗ ΠΑΥΣΗ ΠΕΡΙΣΤΟΦΗ ΠΑΥΣΗ	4	-	0	1-4		7			8		12	14	15
			1		8	15								
			2			15								
			3		9	15								
4		15												
ΣΤΥΨΙΜΟ - ΚΑΝΟΝΙΚΟ ΠΡΟΓΡΑΜΜΑ	ΣΤΡΑΓΓΙΣΜΑ ΠΕΡΙΣΤΡΟΦΗ ΠΕΡΙΣΤΡΟΦΗ	5	-	0					8			14	15	
			1	8	10	15								
			2	8	11	15								
ΣΤΥΨΙΜΟ - ΕΥΑΙΣΘΗΤΟ ΠΡΟΓΡΑΜΜΑ	ΣΤΡΑΓΓΙΣΜΑ ΠΕΡΙΣΤΡΟΦΗ ΠΕΡΙΣΤΡΟΦΗ	5	-	0					8			14	15	
			1	8	10	15								
			2	8	10	15								

Απαραίτητη συνθήκη για να ξεκινήσει το πρόγραμμα, αφού μόνο τότε ενεργοποιείται η μνήμη του κουμπιού start (I0.0), είναι να έχει ασφαλίσει η πόρτα του πλυντηρίου (I0.1), να έχει επιλεγεί η κατάλληλη θερμοκρασία (I0.2 - I0.5) εκ των τεσσάρων που δίνονται αλλά και ένα εκ των δύο προγραμμάτων πλύσης (I0.6 – I0.7).

Με την ενεργοποίηση της μνήμης start ξεκινάει η πλήρωση του κάδου (Q0.9) χρησιμοποιώντας την αριστερή θέση απορρυπαντικού (Q0.5) μέχρι αυτός να γεμίσει με την επιθυμητή ποσότητα νερού (I0.9). Όταν η πλήρωση του ολοκληρωθεί, ξεκινάει η διαδικασία του drum controller (που ουσιαστικά είναι μία εναλλαγή ανάμεσα στους κινητήρες Q0.0 - Q0.1) για δύο φορές. Αυτό επιτυγχάνεται με την χρήση του απαριθμητή (C0) ο οποίος μετράει τον αριθμό των επαναλήψεων της διαδικασίας του drum controller. Ο απαριθμητής C0 δέχεται το κατάλληλο σήμα από τον drum controller όταν αυτός τελειώσει για πρώτη φορά την λειτουργία του, οπότε και ο συγκεκριμένος απαριθμητής αυξάνει το βήμα του, γεγονός που σηματοδοτεί την δεύτερη συνεχόμενη εκτέλεση της λειτουργίας του drum controller. Όταν τελειώσει αυτό το πρώτο βήμα το οποίο αντιστοιχεί στην πρόπλυση, αυξάνεται η τιμή του απαριθμητή (C1) που μετράει

τα βήματα της λειτουργίας του πλυντηρίου, και σηματοδοτεί την έναρξη του επόμενου βήματος.

Για να πάρει την τιμή 1 ο απαριθμητής C1 και να ξεκινήσει το επόμενο βήμα θα πρέπει να έχουν εξυπηρετηθεί οι ανάλογες συνθήκες (ο απαριθμητής C1 να έχει την τιμή 0, ο drum controller DR0.S να βρίσκεται το τέταρτο βήμα του, ο απαριθμητής C0 να έχει την τιμή 2, το χρονικό TM1 να έχει τελειώσει την χρονομέτρησή του και να είναι ενεργοποιημένος ο τερματικός πλύσης). Αφού πάρει την τιμή 1 ο C1 ξεκινάει η αποστράγγιση (οπότε δουλεύει ο κινητήρας Q0.8 ενώ η αντλία αποστράγγισης Q0.0 παραμένει ανοιχτή) μέχρις ότου ενεργοποιηθεί ο τερματικός που δηλώνει ότι ο κάδος είναι άδειος (I0.10). Στη συνέχεια ξαναγίνεται πλήρωση του κάδου (Q0.9) αυτή τη φορά από τη μεσαία θέση απορρυπαντικού (Q0.6) μέχρι την ενεργοποίηση και πάλι του τερματικού πλύσης (I0.9) που σηματοδοτεί την έναρξη της διαδικασίας του drum controller και την επανάληψή της για τρεις φορές. Κατά την πρώτη επανάληψη ο απαριθμητής C0 έχει την τιμή 1, ενώ στην δεύτερη επανάληψη έχει την τιμή 2. Σε αυτό το σημείο είναι απαραίτητο ο απαριθμητής C1 να έχει την τιμή 1 και ο απαριθμητής C0 να έχει αποστείλει σήμα ολοκλήρωσης της εργασίας του (C0_DONE) για να προχωρήσει η διαδικασία στο επόμενο βήμα της.

Αφού έχει πάρει την τιμή 2 ο απαριθμητής C1, ξεκινάει για μία ακόμη φορά η αποστράγγιση (Q0.8 - Q0.0) ως την ενεργοποίηση του τερματικού που υποδεικνύει ότι ο κάδος είναι άδειος (I0.10). Έπειτα ξεκινάει η πλήρωση του κάδου (Q0.10) αυτή τη φορά με την μέγιστη ποσότητα νερού, ως την ενεργοποίηση του τερματικού πλήρους γεμίσματος (I0.8) που σηματοδοτεί για μία ακόμη φορά τη έναρξη του drum controller. Σε αυτό το σημείο για να αυξήσει την τιμή του ο απαριθμητής C1, θα πρέπει να είναι ενεργοποιημένος ο I0.8, ο ίδιος ο C1 να έχει την τιμή 2 και ο drum controller DR0.S να έχει την τιμή 4.

Μετά την αύξηση της τιμής του από 2 σε 3 ο απαριθμητής C1, ξεκινάει για μία ακόμη φορά η αποστράγγιση (Q0.8 - Q0.0) ως την ενεργοποίηση του τερματικού που δηλώνει ότι ο κάδος είναι άδειος (I0.10). Στη συνέχεια ξαναγίνεται πλήρωση του κάδου (Q0.10) για το δεύτερο ξέπλυμα, μέχρις ότου ενεργοποιηθεί πάλι ο τερματικός πλήρους γεμίσματος (I0.8) που σηματοδοτεί την έναρξη της διαδικασίας του drum controller η οποία γίνεται μία φορά. Ο απαριθμητής C1 παίρνει την τιμή 4 εφόσον ο I0.8 είναι ενεργοποιημένος, ο ίδιος ο C1 έχει την τιμή 3, και ο drum controller DR0.S έχει την τιμή 4.

Αφού πάρει την τιμή 4 ο C1 ξεκινάει για μία ακόμη φορά η αποστράγγιση (Q0.8 - Q0.0) μέχρις ότου ενεργοποιηθεί ο τερματικός που υποδεικνύει ότι ο κάδος είναι άδειος (I0.10). Στη συνέχεια ξαναγίνεται πλήρωση του κάδου (Q0.9) χρησιμοποιώντας την δεξιά θέση απορρυπαντικού στην οποία βρίσκεται το μαλακτικό (Q0.7), μέχρι να γεμίσει ο κάδος του πλυντηρίου με την επιθυμητή ποσότητα νερού (I0.9). Όταν η πλήρωση του κάδου ολοκληρωθεί ξεκινάει η διαδικασία του drum controller. Με την ολοκλήρωσή της, και αφού ο I0.9 είναι

ενεργοποιημένος, ο απαριθμητής C1 έχει την τιμή 4, και ο drum controller DR0.S έχει την τιμή 4, ο C1 παίρνει την τιμή 5.

Η αύξηση της τιμής του απαριθμητή C1 σε 5, ουσιαστικά σηματοδοτεί την έναρξη του τελευταίου βήματος. Το βήμα αυτό ξεκινάει για μία ακόμη φορά με την αποστράγγιση (Q0.8 - Q0.0) ως την ενεργοποίηση του τερματικού που δηλώνει ότι ο κάδος είναι άδειος (I0.10). Με το τέλος της αποστράγγισης ξεκινάει το στύψιμο στο οποίο έγκειται και η μοναδική διαφορά ανάμεσα στο κανονικό πρόγραμμα, και στο πρόγραμμα για ευαίσθητα ρούχα. Λόγω αυτής της ιδιαιτερότητας, κρίνεται σκόπιμο τα δύο αυτά προγράμματα να αναλυθούν ξεχωριστά.

Έτσι, στο κανονικό πρόγραμμα η διαδικασία αρχινάει με την ενεργοποίηση του κινητήρα 2 για μία χρονική περίοδο η οποία ορίζεται από το χρονικό TM0. Μετά το τέλος αυτής μπαίνει σε λειτουργία για την ίδια ακριβώς χρονική περίοδο ο κινητήρας 3. Αυτό επιτυγχάνεται με την βοήθεια του απαριθμητή 2 ο οποίος αρχικά αυξάνει το βήμα του με την ικανοποίηση της κατάλληλης συνθήκης (ο απαριθμητής C1 έχει την τιμή 5, ο I0.10 είναι ενεργοποιημένος και ο C2 έχει την τιμή 0) και θέτει σε λειτουργία τον κινητήρα 2. Έπειτα αυξάνει για δεύτερη φορά το βήμα και εφόσον ο C2 έχει την τιμή 1 και ο κινητήρας 2 έχει τελειώσει την λειτουργία του, ενεργοποιεί τον κινητήρα 3.

Στο ευαίσθητο πρόγραμμα αντίθετα, ενεργοποιείται ο κινητήρας 2 για δύο συνεχόμενες χρονικές περιόδους που έχουν οριστεί στο χρονικό TM0. Αυτό επιτυγχάνεται με την βοήθεια του απαριθμητή C2 ο οποίος αυξάνει το βήμα του την πρώτη φορά με την ικανοποίηση της αντίστοιχης συνθήκης (ο ίδιος ο απαριθμητής να έχει την τιμή 5, ο I0.10 να έχει ενεργοποιηθεί, και ο απαριθμητής C2 να έχει την τιμή 0), και ενεργοποιεί τον κινητήρα 2 για μία χρονική περίοδο. Έπειτα αυξάνει ακόμη μία φορά το βήμα του εφόσον ο C2 έχει την τιμή 1 και ο κινητήρας 2 έχει τελειώσει την λειτουργία του, και ενεργοποιεί δεύτερη φορά τον κινητήρα 2 για ακόμη μία χρονική περίοδο.

Καθ' όλη τη διάρκεια της πλύσης είναι ενεργοποιημένη η λυχνία λειτουργίας η οποία απενεργοποιείται με το τέλος της λειτουργίας του πλυντηρίου που σηματοδοτείται από την λήξη του στυψίματος, από χειροκίνητη παύση της λειτουργίας μέσω του κουμπιού stop και με την πτώση της τάσης του ρεύματος.

Όσον αφορά το τέλος λειτουργίας, η ενεργοποίηση αυτής της μνήμης και η απενεργοποίηση του πλυντηρίου στην περίπτωση που έχει επιλεγεί το κανονικό πρόγραμμα, γίνεται μόνο αν ο απαριθμητής C2 στείλει σήμα ολοκλήρωσης της εργασίας του, δηλαδή το C2_DONE είναι ενεργοποιημένο, αντίστοιχα και το I0.6, αλλά και να έχει ολοκληρώσει την λειτουργία του ο κινητήρας 3. Στην περίπτωση της επιλογής του προγράμματος για τα ευαίσθητα ρούχα, θα πρέπει ο απαριθμητής C2 να στείλει σήμα ολοκλήρωσης της εργασίας του, δηλαδή το

C2_DONE να είναι ενεργοποιημένο, αντίστοιχα και το I0.7, αλλά και να έχει τελειώσει την λειτουργία του ο κινητήρας 2.

Για την χειροκίνητη παύση της λειτουργίας χρησιμοποιήθηκε ένας ακόμη απαριθμητής (C3) διότι σε αυτή τη περίπτωση θέλουμε να διακόπτεται το πρόγραμμα σε οποιαδήποτε στιγμή, να γίνεται αποστράγγιση του νερού που βρίσκεται στον κάδο, ενώ όταν ξαναπατηθεί το κουμπί start να γίνεται και πάλι η πλήρωση του με την κατάλληλη ποσότητα νερού και να συνεχίζει η λειτουργία του πλυντηρίου από το σημείο που είχε διακοπεί. Η μόνη εξαίρεση στην διαδικασία αυτή εντοπίζεται στην περίοδο του στυσίματος, οπότε το πρόγραμμα δεν δέχεται χειροκίνητη διακοπή, το οποίο επετεύχθη με την χρήση του σήματος ολοκλήρωσης της εργασίας του απαριθμητή C3, δηλαδή το C3_DONE.

Για την επίτευξη μίας τεχνητής πτώση τάσης χρησιμοποιήθηκε μία κλειστή επαφή σε όλα σχεδόν τα rang λειτουργίας του προγράμματος ώστε να απενεργοποιούνται. Όταν ενεργοποιηθεί εκ νέου η επαφή, η λειτουργία συνεχίζει από το σημείο που διακόπηκε, αφού με αυτό τον τρόπο προσομοιώνεται η πραγματική πτώση τάσης, κατά την οποία το πλυντήριο δεν διαρρέεται από ρεύμα και έτσι δεν είναι δυνατή καμία ενέργεια.

4.2.4 ΠΑΡΑΔΟΧΕΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

Με την προσεκτική ανάγνωση των πινάκων εισόδων και εξόδων που παρατέθηκαν προηγουμένως, είναι εύκολος ο εντοπισμός των παραδοχών που πραγματοποιήθηκαν για την διευκόλυνση της παρουσίασης του προγράμματος αλλά και λόγω της μη εφαρμογής του σε πραγματικό πλυντήριο.

Έτσι, εισοδοί που κανονικά θα ήταν διαφορετικής μορφής, όπως για παράδειγμα τα κουμπιά ελέγχου της θερμοκρασίας του νερού (σε ένα πραγματικό πλυντήριο θα είχαν την μορφή ενός περιστρεφόμενου κουμπιού με την περιστροφή του οποίου θα ρυθμιζόταν η θερμοκρασία, ενώ το σήμα για την επίτευξη της θερμοκρασίας αυτής θα ερχόταν από έναν θερμοστάτη). Λόγω αυτών των περιορισμών στην προκειμένη περίπτωση χρησιμοποιήθηκαν κουμπιά για ορισμένες ενδεικτικές θερμοκρασίες, που με την επιλογή και το πάτημα ενός από αυτά υποτίθεται πως αυτόματα επιτυγχάνεται και η επιθυμητή θερμοκρασία.

Επίσης σε ένα πραγματικό πλυντήριο, ορισμένες διαδικασίες θα γινόταν αυτόματα με την παροχή κατάλληλων σημάτων από τους τερματικούς που υπάρχουν σε αυτό όπως είναι ο τερματικός ασφάλισης της πόρτας του πλυντηρίου καθώς και οι τερματικοί για την ένδειξη της στάθμης του νερού που υπάρχει στον κάδο. Έτσι κάτω από κανονικές συνθήκες όταν η πόρτα θα

ασφάλιζε θα εμφανιζόταν αυτόματα η ένδειξη αυτή στο PLC, και δεν θα ήταν απαραίτητη η χειροκίνητη παρέμβαση όπως συμβαίνει στο πρόγραμμα όπου πρέπει να ενεργοποιηθεί από τον χρήστη η ανάλογη ένδειξη σχεδόν σαν ο τερματικός να είναι ένα κουμπί. Αντίστοιχα και στην περίπτωση των τερματικών της στάθμης του νερού, με το γέμισμα του κάδου με νερό ή το άδειασμα του, θα ενεργοποιούνταν οι κατάλληλες ενδείξεις και δεν θα ήταν απαραίτητη η όποια παρέμβαση.

Ακόμη και στην περίπτωση της πτώσης της τάσης, αυτή έπρεπε να γίνει χειροκίνητα, όπως θα γινόταν με την χρήση ενός κουμπιού, για την ευκολότερη παρουσίαση του προγράμματος με έναν πιο απλό και κατανοητό τρόπο. Τέλος για λόγους οικονομίας χρόνου, μειώθηκε ο χρόνος της λειτουργίας του πλυντηρίου, καθώς δεν θα ήταν διαφορετικά δυνατή η παρουσίαση όλων των λειτουργιών ενός πλυντηρίου με επάρκεια και εντός χρονικών ορίων.

Παραδοχές έγιναν και στον αριθμό των επιλογών για τον τρόπο πλύσης (προγραμμάτων) που διατίθενται, αφού οι δυνατότητες είναι αρκετές, ενώ επιλέχθηκαν τα πιο συχνά εμφανιζόμενα και χρησιμοποιούμενα προγράμματα για την εργασία αυτή.

ΠΑΡΑΡΤΗΜΑ Α

ΔΙΑΔΙΚΑΣΙΑ ΛΕΙΤΟΥΡΓΙΑΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ DRUM:

ΚΙΝΗΣΗ ΚΙΝΗΤΗΡΑ ΔΕΞΙΟΣΤΡΟΦΑ ΚΑΝΟΝΙΚΑ ΓΙΑ 4 SEC, ΠΑΥΣΗ ΚΙΝΗΤΗΡΑ ΓΙΑ 1 SEC, ΑΡΙΣΤΕΡΟΣΤΡΟΦΗ ΚΙΝΗΣΗ ΚΙΝΗΤΗΡΑ ΚΑΝΟΝΙΚΑ ΓΙΑ 4 SEC, ΠΑΥΣΗ ΚΙΝΗΤΗΡΑ ΓΙΑ 1 SEC.

ΔΙΑΔΙΚΑΣΙΑ ΠΛΥΝΤΗΡΙΟΥ:

ΓΙΑ ΝΑ ΞΕΚΙΝΗΣΕΙ ΤΟ ΠΛΥΝΤΗΡΙΟ ΠΡΕΠΕΙ ΝΑ ΕΠΙΛΕΓΕΙ ΜΙΑ ΑΠΟ ΤΙΣ ΘΕΡΜΟΚΡΑΣΙΕΣ ΝΕΡΟΥ ΠΟΥ ΔΙΔΟΝΤΑΙ 30°C,40°C,60°C,90°C (Ι0.2 - Ι0.3 - Ι0.4 - Ι0.5),ΕΝΑ ΠΡΟΓΡΑΜΜΑ ΓΙΑ ΚΑΝΟΝΙΚΗ ΠΛΥΣΗ Ή ΓΙΑ ΕΥΑΙΣΘΗΤΑ ΡΟΥΧΑ (Ι0.6 - Ι0.7), ΝΑ ΑΣΦΑΛΙΣΕΙ Η ΠΟΡΤΑ (Ι0.1) ΚΑΙ ΝΑ ΠΑΤΗΘΕΙ ΤΟ START (Ι0.0).

ΜΕ ΤΟ ΠΑΤΗΜΑ ΤΟΥ START ΞΕΚΙΝΑΕΙ ΤΟ ΓΕΜΙΣΜΑ ΤΟΥ ΠΛΥΝΤΗΡΙΟΥ ΜΕ ΝΕΡΟ ΑΠΟ ΤΗΝ ΑΝΤΛΙΑ (Q0.9) ΠΕΡΝΟΝΤΑΣ ΑΠΟ ΤΗΝ ΑΡΙΣΤΕΡΗ ΘΕΣΗ ΤΟΥ ΑΠΟΡΡΥΠΑΝΤΙΚΟΥ (Q0.5). ΜΕΤΑ ΑΠΟ ΛΙΓΟ ΠΑΤΑΜΕ ΤΟΝ ΤΕΡΜΑΤΙΚΟ ΠΛΥΣΗΣ (Ι0.9) ΩΣ ΕΝΔΕΙΞΗ ΟΤΙ ΕΧΕΙ ΓΕΜΙΣΕΙ Ο ΚΑΔΟΣ ΜΕ ΤΗΝ ΑΠΑΡΑΙΤΗΤΗ ΠΟΣΟΤΗΤΑ ΝΕΡΟΥ ΚΑΙ ΑΥΤΟΜΑΤΑ ΞΕΚΙΝΑΕΙ Η ΔΙΑΔΙΚΑΣΙΑ ΤΟΥ DRUM (ΔΥΟ ΦΟΡΕΣ ΣΥΝΕΧΟΜΕΝΑ) ---ΠΡΟΠΛΥΣΗ---

ΟΤΑΝ ΤΕΛΕΙΩΣΕΙ Η ΔΙΑΔΙΚΑΣΙΑ ΤΟΥ DRUM ΑΝΟΙΓΕΙ Η ΑΠΟΣΤΡΑΓΓΙΣΗ (Q0.8) ΕΝΩ Ο ΚΙΝΗΤΗΡΑΣ ΚΙΝΕΙΤΑΙ ΔΕΞΙΟΣΤΡΟΦΑ ΣΕ ΚΑΝΟΝΙΚΗ ΤΑΧΥΤΗΤΑ (Q0.2) ---ΑΠΟΣΤΡΑΓΓΙΣΗ---

ΠΡΕΠΕΙ ΝΑ ΚΛΕΙΣΕΙ ΠΡΩΤΑ Ο ΤΕΡΜΑΤΙΚΟΣ ΣΤΑΘΜΗΣ ΠΛΥΣΗΣ (Ι0.9) ΚΑΙ ΝΑ ΑΝΟΙΞΕΙ Ο ΤΕΡΜΑΤΙΚΟΣ ΚΑΤΩ ΣΤΑΘΜΗΣ (Ι0.10) - ΕΝΔΕΙΞΗ ΠΩΣ Ο ΚΑΔΟΣ ΕΧΕΙ ΑΔΕΙΑΣΕΙ ΑΠΟ ΤΟ ΝΕΡΟ. ΟΤΑΝ ΠΑΤΗΘΕΙ ΤΟ Ι0.10 ΞΕΚΙΝΑΕΙ ΠΑΛΙ ΤΟ ΓΕΜΙΣΜΑ ΤΟΥ ΚΑΔΟΥ ΜΕ ΝΕΡΟ (Q0.9) ΜΟΝΟ ΠΟΥ ΑΥΤΗ ΤΗ ΦΟΡΑ ΠΕΡΝΑΕΙ ΑΠΟ ΤΗΝ ΜΕΣΣΑΙΑ ΘΕΣΗ ΤΟΥ ΑΠΟΡΡΥΠΑΝΤΙΚΟΥ (Q0.6). ΜΕΤΑ ΑΠΟ ΕΝΑ ΔΙΑΣΤΗΜΑ ΠΟΥ ΚΡΙΝΕΤΑΙ ΑΠΑΡΑΙΤΗΤΟ ΓΙΑ ΤΟ ΓΕΜΙΣΜΑ ΤΟΥ ΚΑΔΟΥ ΚΛΕΙΝΟΥΜΕ ΤΟ Ι0.10 ΚΑΙ ΕΝΕΡΓΟΠΟΙΟΥΜΕ ΞΑΝΑ ΤΟ Ι0.9 ΠΟΥ ΣΗΜΑΙΝΕΙ ΟΤΙ Ο ΚΑΔΟΣ ΕΧΕΙ ΓΕΜΙΣΕΙ ΜΕ ΤΗΝ ΚΑΤΑΛΛΗΛΗ ΠΟΣΟΤΗΤΑ ΝΕΡΟΥ. ΞΕΚΙΝΑΕΙ Η ΔΙΑΔΙΚΑΣΙΑ DRUM (Η ΟΠΟΙΑ ΕΠΑΝΑΛΑΜΒΑΝΕΤΑΙ ΓΙΑ ΤΡΕΙΣ ΦΟΡΕΣ) ---ΚΥΡΙΩΣ ΠΛΥΣΗ---

ΟΤΑΝ ΤΕΛΕΙΩΣΕΙ Η ΔΙΑΔΙΚΑΣΙΑ ΤΟΥ DRUM ΑΝΟΙΓΕΙ Η ΑΠΟΣΤΡΑΓΓΙΣΗ (Q0.8) ΕΝΩ Ο ΚΙΝΗΤΗΡΑΣ ΚΙΝΕΙΤΑΙ ΔΕΞΙΟΣΤΡΟΦΑ ΣΕ ΚΑΝΟΝΙΚΗ ΤΑΧΥΤΗΤΑ (Q0.2) ---ΑΠΟΣΤΡΑΓΓΙΣΗ---

ΠΡΕΠΕΙ ΝΑ ΚΛΕΙΣΕΙ ΠΡΩΤΑ Ο ΤΕΡΜΑΤΙΚΟΣ FULL ΓΕΜΙΣΜΑΤΟΣ (Ι0.8) ΚΑΙ ΝΑ ΑΝΟΙΞΕΙ Ο ΤΕΡΜΑΤΙΚΟΣ ΚΑΤΩ ΣΤΑΘΜΗΣ (Ι0.10) - ΕΝΔΕΙΞΗ ΠΩΣ Ο ΚΑΔΟΣ ΕΧΕΙ ΑΔΕΙΑΣΕΙ ΑΠΟ ΤΟ ΝΕΡΟ. ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΕΤΣΙ Η ΑΝΤΛΙΑ ΓΙΑ ΤΟ ΓΕΜΙΣΜΑ ΤΟΥ ΚΑΔΟΥ ΜΕ ΠΕΡΙΣΣΟΤΕΡΟ ΝΕΡΟ (FULL). ΜΕΤΑ ΑΠΟ ΕΝΑ ΔΙΑΣΤΗΜΑ ΠΟΥ ΚΡΙΝΕΤΑΙ ΑΠΑΡΑΙΤΗΤΟ ΓΙΑ ΤΟ ΓΕΜΙΣΜΑ ΤΟΥ, ΚΛΕΙΝΟΥΜΕ ΤΟ Ι0.10 ΚΑΙ ΕΝΕΡΓΟΠΟΙΟΥΜΕ ΤΟΝ ΤΕΡΜΑΤΙΚΟ FULL ΓΕΜΙΣΜΑΤΟΣ (Ι0.8) ΠΟΥ ΣΗΜΑΙΝΕΙ ΟΤΙ Ο ΚΑΔΟΣ ΕΧΕΙ ΓΕΜΙΣΕΙ ΜΕ ΤΗΝ ΚΑΤΑΛΛΗΛΗ ΠΟΣΟΤΗΤΑ ΝΕΡΟΥ. ΞΕΚΙΝΑΕΙ Η ΔΙΑΔΙΚΑΣΙΑ DRUM (ΜΙΑ ΦΟΡΑ) ---**ΞΕΒΓΑΛΜΑ 1**---

ΟΤΑΝ ΤΕΛΕΙΩΣΕΙ Η ΔΙΑΔΙΚΑΣΙΑ ΤΟΥ DRUM ΑΝΟΙΓΕΙ Η ΑΠΟΣΤΡΑΓΓΙΣΗ (Q0.8) ΕΝΩ Ο ΚΙΝΗΤΗΡΑΣ ΚΙΝΕΙΤΑΙ ΔΕΞΙΟΣΤΡΟΦΑ ΣΕ ΚΑΝΟΝΙΚΗ ΤΑΧΥΤΗΤΑ (Q0.2) ---**ΑΠΟΣΤΡΑΓΓΙΣΗ**---

ΠΡΕΠΕΙ ΝΑ ΚΛΕΙΣΕΙ ΠΡΩΤΑ Ο ΤΕΡΜΑΤΙΚΟΣ FULL ΓΕΜΙΣΜΑΤΟΣ (Ι0.8) ΚΑΙ ΝΑ ΑΝΟΙΞΕΙ Ο ΤΕΡΜΑΤΙΚΟΣ ΚΑΤΩ ΣΤΑΘΜΗΣ (Ι0.10) - ΕΝΔΕΙΞΗ ΠΩΣ Ο ΚΑΔΟΣ ΕΧΕΙ ΑΔΕΙΑΣΕΙ ΑΠΟ ΤΟ ΝΕΡΟ. ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΕΤΣΙ Η ΑΝΤΛΙΑ ΓΙΑ ΤΟ ΓΕΜΙΣΜΑ ΤΟΥ ΚΑΔΟΥ ΜΕ ΠΕΡΙΣΣΟΤΕΡΟ ΝΕΡΟ (FULL). ΜΕΤΑ ΑΠΟ ΕΝΑ ΔΙΑΣΤΗΜΑ ΠΟΥ ΚΡΙΝΕΤΑΙ ΑΠΑΡΑΙΤΗΤΟ ΓΙΑ ΤΟ ΓΕΜΙΣΜΑ ΤΟΥ, ΚΛΕΙΝΟΥΜΕ ΤΟ Ι0.10 ΚΑΙ ΕΝΕΡΓΟΠΟΙΟΥΜΕ ΤΟΝ ΤΕΡΜΑΤΙΚΟ FULL ΓΕΜΙΣΜΑΤΟΣ (Ι0.8) ΠΟΥ ΣΗΜΑΙΝΕΙ ΟΤΙ Ο ΚΑΔΟΣ ΕΧΕΙ ΓΕΜΙΣΕΙ ΜΕ ΤΗΝ ΚΑΤΑΛΛΗΛΗ ΠΟΣΟΤΗΤΑ ΝΕΡΟΥ. ΞΕΚΙΝΑΕΙ Η ΔΙΑΔΙΚΑΣΙΑ DRUM (ΜΙΑ ΦΟΡΑ) ---**ΞΕΒΓΑΛΜΑ 2**---

ΟΤΑΝ ΤΕΛΕΙΩΣΕΙ Η ΔΙΑΔΙΚΑΣΙΑ ΤΟΥ DRUM ΑΝΟΙΓΕΙ Η ΑΠΟΣΤΡΑΓΓΙΣΗ (Q0.8) ΕΝΩ Ο ΚΙΝΗΤΗΡΑΣ ΚΙΝΕΙΤΑΙ ΔΕΞΙΟΣΤΡΟΦΑ ΣΕ ΚΑΝΟΝΙΚΗ ΤΑΧΥΤΗΤΑ (Q0.2) ---**ΑΠΟΣΤΡΑΓΓΙΣΗ**---

ΠΡΕΠΕΙ ΝΑ ΚΛΕΙΣΕΙ ΠΡΩΤΑ Ο ΤΕΡΜΑΤΙΚΟΣ ΣΤΑΘΜΗΣ ΠΛΥΣΗΣ (Ι0.9) ΚΑΙ ΝΑ ΑΝΟΙΞΕΙ Ο ΤΕΡΜΑΤΙΚΟΣ ΚΑΤΩ ΣΤΑΘΜΗΣ (Ι0.10) - ΕΝΔΕΙΞΗ ΠΩΣ Ο ΚΑΔΟΣ ΕΧΕΙ ΑΔΕΙΑΣΕΙ ΑΠΟ ΤΟ ΝΕΡΟ. ΟΤΑΝ ΠΑΤΗΘΕΙ ΤΟ Ι0.10 ΞΕΚΙΝΑΕΙ ΠΑΛΙ ΤΟ ΓΕΜΙΣΜΑ ΤΟΥ ΚΑΔΟΥ ΜΕ ΝΕΡΟ (Q0.9) ΜΟΝΟ ΠΟΥ ΑΥΤΗ ΤΗ ΦΟΡΑ ΠΕΡΝΑΕΙ ΑΠΟ ΤΗΝ ΘΕΣΗ ΤΟΥ ΜΑΛΑΚΤΙΚΟΥ (Q0.7). ΜΕΤΑ ΑΠΟ ΕΝΑ ΔΙΑΣΤΗΜΑ ΠΟΥ ΚΡΙΝΟΥΜΕ ΑΠΑΡΑΙΤΗΤΟ ΓΙΑ ΤΟ ΓΕΜΙΣΜΑ ΤΟΥ, ΚΛΕΙΝΟΥΜΕ ΤΟ Ι0.10 ΚΑΙ ΕΝΕΡΓΟΠΟΙΟΥΜΕ ΞΑΝΑ ΤΟ Ι0.9 ΠΟΥ ΣΗΜΑΙΝΕΙ ΟΤΙ Ο ΚΑΔΟΣ ΕΧΕΙ ΓΕΜΙΣΕΙ ΜΕ ΤΗΝ ΚΑΤΑΛΛΗΛΗ ΠΟΣΟΤΗΤΑ ΝΕΡΟΥ. ΞΕΚΙΝΑΕΙ Η ΔΙΑΔΙΚΑΣΙΑ DRUM (ΜΙΑ ΦΟΡΑ) ---**ΜΑΛΑΚΤΙΚΟ**---

ΟΤΑΝ ΤΕΛΕΙΩΣΕΙ Η ΔΙΑΔΙΚΑΣΙΑ ΤΟΥ DRUM ΑΝΟΙΓΕΙ Η ΑΠΟΣΤΡΑΓΓΙΣΗ (Q0.8) ΕΝΩ Ο ΚΙΝΗΤΗΡΑΣ ΚΙΝΕΙΤΑΙ ΔΕΞΙΟΣΤΡΟΦΑ ΣΕ ΚΑΝΟΝΙΚΗ ΤΑΧΥΤΗΤΑ (Q0.2) ---**ΑΠΟΣΤΡΑΓΓΙΣΗ**---

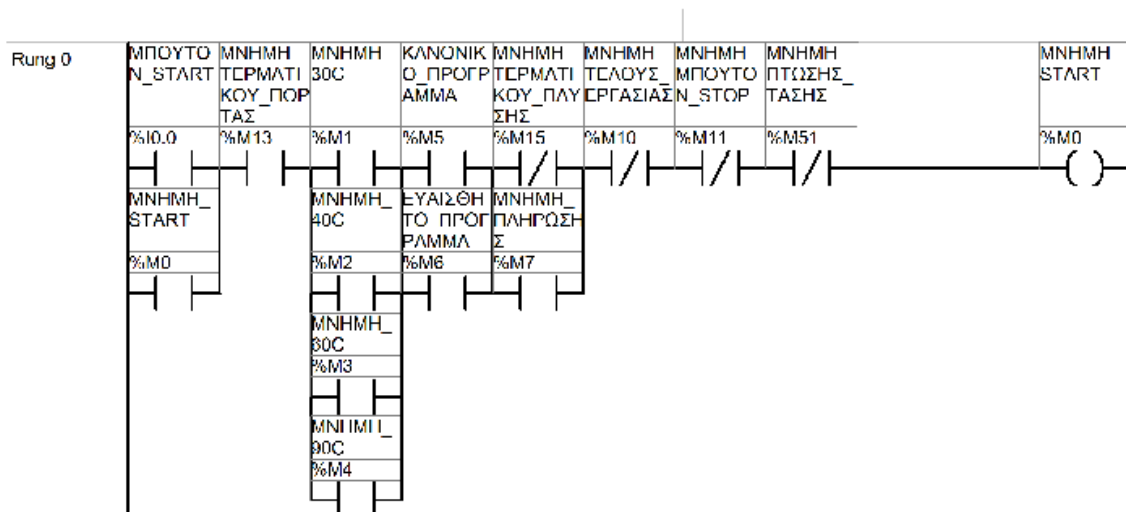
ΠΡΕΠΕΙ ΝΑ ΚΛΕΙΣΕΙ ΠΡΩΤΑ Ο ΤΕΡΜΑΤΙΚΟΣ ΣΤΑΘΜΗΣ ΠΛΥΣΗΣ (Ι0.9) ΚΑΙ ΝΑ ΑΝΟΙΞΕΙ Ο ΤΕΡΜΑΤΙΚΟΣ ΚΑΤΩ ΣΤΑΘΜΗΣ (Ι0.10) - ΕΝΔΕΙΞΗ ΠΩΣ Ο ΚΑΔΟΣ ΕΧΕΙ ΑΔΕΙΑΣΕΙ ΑΠΟ ΤΟ ΝΕΡΟ. ΟΤΑΝ ΠΑΤΗΘΕΙ ΤΟ Ι0.10 ΑΝΟΙΓΕΙ Η

ΑΠΟΣΤΡΑΓΓΙΣΗ (Q0.8) ΕΝΩ Ο ΚΙΝΗΤΗΡΑΣ ΚΙΝΕΙΤΑΙ ΑΡΙΣΤΕΡΟΣΤΡΟΦΑ ΜΕ ΜΕΣΑΙΑ ΤΑΧΥΤΗΤΑ ΓΙΑ 4 SEC ΚΑΙ ΜΕΤΑ ΑΡΙΣΤΕΡΟΣΤΡΟΦΑ ΜΕ ΓΡΗΓΟΡΗ ΤΑΧΥΤΗΤΑ ΓΙΑ ΑΛΛΑ 4 SEC (ΑΝ ΕΧΕΙ ΕΠΙΛΕΧΘΕΙ ΤΟ ΚΑΝΟΝΙΚΟ ΠΡΟΓΡΑΜΜΑ ΕΝΩ ΑΝ ΕΧΕΙ ΕΠΙΛΕΧΘΕΙ ΤΟ ΠΡΟΓΡΑΜΜΑ ΓΙΑ ΕΥΑΙΣΘΗΤΑ ΤΟΤΕ Ο ΚΙΝΗΤΗΡΑΣ ΚΙΝΕΙΤΑΙ ΔΕΞΙΟΣΤΡΟΦΑ ΓΙΑ 8 SEC ΣΤΗ ΜΕΣΑΙΑ ΤΑΧΥΤΗΤΑ). ---ΣΤΥΨΙΜΟ---

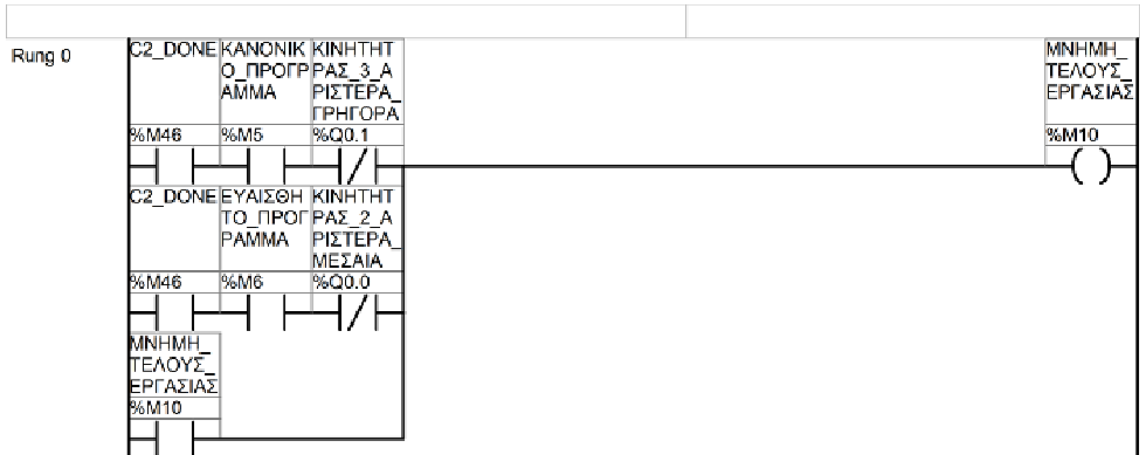
ΣΕ ΟΛΗ ΤΗ ΔΙΑΡΚΕΙΑ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΕΙΝΑΙ ΑΝΑΜΜΕΝΗ Η ΛΥΧΝΙΑ ΛΕΙΤΟΥΡΓΙΑΣ (Q0.15).

ΥΠΑΡΧΕΙ ΣΤΟ ΠΡΟΓΡΑΜΜΑ Η ΕΠΙΛΟΓΗ ΤΗΣ ΠΤΩΣΗΣ ΤΗΣ ΤΑΣΗΣ (EMERGENCY STOP) (I0.15) ΜΕ ΤΟ ΠΑΤΗΜΑ ΤΗΣ ΟΠΟΙΑΣ Η ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ ΠΛΥΝΤΗΡΙΟΥ ΔΙΑΚΟΠΤΕΤΑΙ ΚΑΙ ΣΕ ΠΕΡΙΠΤΩΣΗ ΕΠΑΝΑΦΟΡΑΣ ΤΗΣ ΣΥΝΕΧΙΖΕΙ ΑΠΟ ΕΚΕΙ ΠΟΥ ΣΤΑΜΑΤΗΣΕ, ΑΛΛΑ ΚΑΙ ΤΗΣ ΕΠΙΛΟΓΗΣ STOP ΜΕ ΤΟ ΠΑΤΗΜΑ ΤΟΥ ΚΑΤΑΛΛΗΛΟΥ ΚΟΥΜΠΙΟΥ (I0.11), ΟΠΟΤΕ ΤΟ ΠΛΥΝΤΗΡΙΟ ΣΤΑΜΑΤΑΕΙ ΤΗΝ ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ, ΑΠΟΣΤΡΑΓΓΙΖΕΙ ΤΟ ΝΕΡΟ ΠΟΥ ΥΠΑΡΧΕΙ, ΑΠΑΣΦΑΛΙΖΕΙ ΤΗΝ ΠΟΡΤΑ ΕΝΩ ΑΝ ΞΑΝΑΞΕΚΙΝΗΣΕΙ Η ΛΕΙΤΟΥΡΓΙΑ ΜΕ ΤΟ ΠΑΤΗΜΑ ΤΟΥ START ΕΠΑΝΑΣΦΑΛΙΖΕΙ ΤΗΝ ΠΟΡΤΑ, ΞΑΝΑΓΕΜΙΖΕΙ ΜΕ ΝΕΡΟ ΚΑΙ ΣΥΝΕΧΙΖΕΙ ΤΗΝ ΠΛΥΣΗ ΑΠΟ ΕΚΕΙ ΠΟΥ ΣΤΑΜΑΤΗΣΕ (ΕΚΤΟΣ ΚΑΙ ΑΝ ΒΡΙΣΚΕΤΑΙ ΣΤΗΝ ΦΑΣΗ ΤΟΥ ΣΤΥΨΙΜΑΤΟΣ ΟΠΟΤΕ ΔΕΝ ΔΕΧΕΤΑΙ STOP).

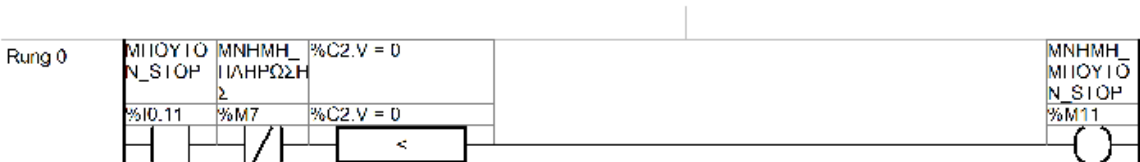
1 LD	ΜΝΗΜΗ ΕΚΚΙΝΗΣΗΣ START	ΘΑ ΠΡΕΠΕΙ ΝΑ ΕΧΕΙ ΑΣΦΑΛΙΣΤΕΙ Η ΠΟΡΤΑ, ΝΑ ΕΧΕΙ ΕΠΙΛΕΓΕΙ Η ΘΕΡΜΟΚΡΑΣΙΑ ΠΛΥΣΗΣ ΚΑΙ ΤΟ ΚΑΤΑΛΛΗΛΟ ΠΡΟΓΡΑΜΜΑ ΚΑΙ ΝΑ ΕΧΕΙ ΠΑΤΗΘΕΙ ΤΟ ΚΟΥΜΠΙ START ΓΙΑ ΝΑ ΞΕΚΙΝΗΣΕΙ Η ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ ΠΛΥΝΤΗΡΙΟΥ. ΣΥΓΧΡΟΝΩΣ ΔΕΝ ΘΑ ΠΡΕΠΕΙ ΝΑ ΥΠΑΡΧΕΙ ΠΤΩΣΗ ΤΑΣΗΣ Η ΝΑ ΕΙΝΑΙ ΠΑΤΗΜΕΝΟ ΤΟ ΚΟΥΜΠΙ STOP ΚΑΙ ΦΥΣΙΚΑ ΓΙΑ ΝΑ ΣΥΝΕΧΙΣΕΙ Η ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ ΠΛΥΝΤΗΡΙΟΥ ΔΕΝ ΘΑ ΠΡΕΠΕΙ ΝΑ ΕΧΕΙ ΦΤΑΣΕΙ Η ΕΡΓΑΣΙΑ ΣΤΟ ΤΕΛΟΣ ΤΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ ΤΕΛΟΣ ΤΗΣ ΜΝΗΜΗΣ ΠΛΥΣΗΣ ΚΑΙ Η ΜΝΗΜΗ ΠΛΗΡΩΣΗΣ ΧΡΗΣΙΜΟΠΟΙΟΥΝΤΑΙ ΣΕ ΣΥΝΔΙΑΣΜΟ ΓΙΑ ΤΗΝ ΛΕΙΤΟΥΡΓΙΑ ΤΗΣ ΛΥΧΝΙΑΣ
------	------------------------------	--



2 LD	ΜΝΗΜΗ ΤΕΛΟΥΣ ΕΡΓΑΣΙΑΣ	ΕΦΟΣΟΝ ΕΧΕΙ ΤΕΛΕΙΩΣΕΙ Ο COUNTER ΕΝΑΛΛΑΓΗΣ ΚΙΝΗΤΗΡΩΝ (ΔΗΛΑΔΗ ΟΥΣΙΑΣΤΙΚΑ ΕΧΕΙ ΤΕΛΕΙΩΣΕΙ ΤΟ ΣΤΥΨΙΜΟ) ΚΑΙ ΕΙΤΕ ΓΙΑ ΤΟ ΚΑΝΟΝΙΚΟ ΠΡΟΓΡΑΜΜΑ Ο ΚΙΝΗΤΗΡΑΣ 3 ΕΧΕΙ ΣΤΑΜΑΤΗΣΕΙ, ΕΙΤΕ ΓΙΑ ΤΟ ΕΥΑΙΣΘΗΤΟ ΠΡΟΓΡΑΜΜΑ Ο ΚΙΝΗΤΗΡΑΣ 2 ΕΧΕΙ ΣΤΑΜΑΤΗΣΕΙ, ΤΟΤΕ ΜΟΝΟ ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ Η ΜΝΗΜΗ ΤΕΛΟΥΣ ΕΡΓΑΣΙΑΣ
-------------	------------------------------	---



3 LD	ΜΠΟΥΤΟΝ STOP	ΑΠΟ ΤΗΝ ΣΤΙΓΜΗ ΠΟΥ ΕΧΕΙ ΠΑΤΗΘΕΙ ΤΟ STOP ΚΑΙ ΑΝ ΔΕΝ ΕΧΕΙ ΕΝΕΡΓΟΠΟΙΗΘΕΙ Η ΜΝΗΜΗ ΠΛΗΡΩΣΗΣ, ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ Η ΜΝΗΜΗ STOP (ΤΟ ΜΙΟΥΙΟΝ STOP ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΣΕ ΟΛΕΣ ΤΙΣ ΔΙΑΔΙΚΑΣΙΕΣ ΕΚΤΟΣ ΤΗΣ ΑΠΟΣΤΡΑΓΓΙΣΗΣ ΚΑΙ ΑΥΤΟ ΤΟ ΕΞΑΣΦΑΛΙΖΕΙ Η ΜΝΗΜΗ ΠΛΗΡΩΣΗΣ ΣΕ ΣΥΝΔΙΑΣΜΟ ΜΕ ΤΟΝ ΜΕΤΡΙΤΗΡ ΕΝΑΛΛΑΓΩΝ ΚΙΝΗΤΗΡΩΝ ΣΤΥΨΙΜΑΤΟΣ ΓΙΑΤΙ ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΑΦΟΥ ΕΧΕΙ ΞΕΚΙΝΗΣΕΙ ΤΟ ΣΤΥΨΙΜΟ)
-------------	---------------------	---



4 LD	ΜΝΗΜΗ ΠΤΩΣΗΣ ΤΑΣΗΣ	ΣΤΑΜΑΤΑΤΑΙ ΤΟ ΠΡΟΓΡΑΜΜΑ ΕΚΕΙ ΠΟΥ ΕΙΝΑΙ ΛΟΓΟ ΕΛΛΕΙΨΗΣ ΤΑΣΗΣ ΚΑΙ ΟΤΑΝ ΕΠΑΝΕΡΧΕΤΑΙ Η ΤΑΣΗ ΣΥΝΕΧΙΖΕΙ ΑΠΟ ΕΚΕΙ ΠΟΥ ΕΧΕΙ ΣΤΑΜΑΤΗΣΕΙ
-------------	---------------------------	---



5 LD

ΜΝΗΜΗ ΠΛΗΡΩΣΗΣ

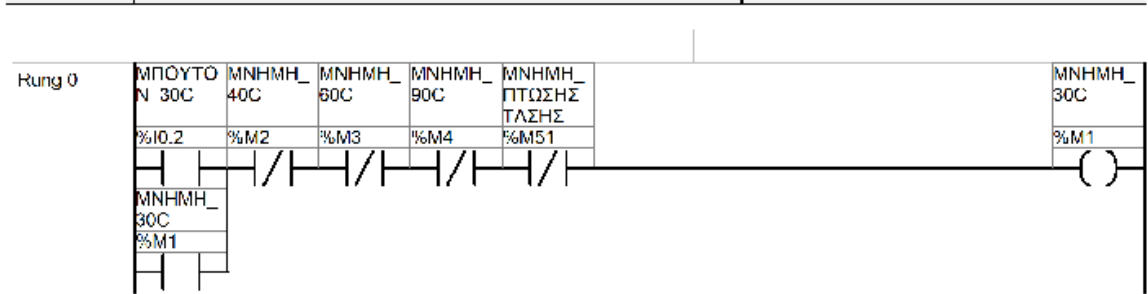
ΑΠΟ ΤΗΝ ΣΤΙΓΜΗ ΠΟΥ ΕΧΕΙ ΛΟΣΦΕΙ ΣΗΜΑ Ο ΤΕΡΜΑΤΙΚΟΣ FULL ΓΕΜΙΣΜΑΤΟΣ Ή Ο ΤΕΡΜΑΤΙΚΟΣ ΓΕΜΙΣΜΑΤΟΣ ΠΛΥΣΗΣ ΚΑΙ ΕΧΕΙ ΤΕΛΕΙΩΣΕΙ ΚΑΙ Ο COUNTER ΤΗΣ MANUAL ΠΛΥΣΗΣ, ΚΑΙ ΜΟΝΟ ΑΝ ΔΕΝ ΥΠΑΡΧΕΙ ΠΤΩΣΗ ΤΑΣΗΣ ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ Η ΜΝΗΜΗ ΠΛΗΡΩΣΗΣ. ΕΝΕΡΓΟΠΟΙΗΣΗ ΤΗΣ ΜΝΗΜΗΣ ΠΛΗΡΩΣΗΣ ΣΗΜΑΙΝΕΙ ΠΩΣ ΕΙΜΑΣΤΕ ΣΤΟ ΤΕΛΕΥΤΑΙΟ ΒΗΜΑ ΤΟΥ ΣΤΥΨΙΜΑΤΟΣ ΚΑΙ Ο ΜΟΝΟΣ ΤΡΟΠΟΣ ΑΠΕΝΕΡΓΟΠΟΙΗΣΗΣ ΤΗΣ ΕΙΝΑΙ ΜΟΝΟ ΣΕ ΠΕΡΙΠΤΩΣΗ ΠΤΩΣΗΣ ΤΑΣΗΣ



6 LD

ΜΝΗΜΗ ΘΕΡΜΟΚΡΑΣΙΑΣ 30C

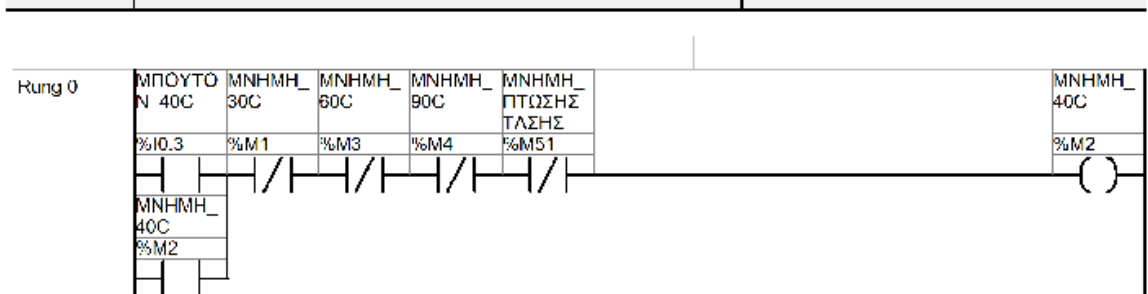
ΕΦΟΣΟΝ ΕΧΕΙ ΠΑΤΗΘΕΙ ΤΟ ΚΟΥΜΠΙ ΤΗΣ ΣΥΓΚΡΕΚΡΙΜΕΝΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΛΕΝ ΕΠΙΤΡΕΠΕΤΑΙ ΝΑ ΕΠΙΛΕΧΘΕΙ ΚΑΜΜΙΑ ΑΛΛΗ ΘΕΡΜΟΚΡΑΣΙΑ, ΚΑΙ ΜΟΝΟ ΟΣΟ ΔΕΝ ΥΠΑΡΧΕΙ ΠΤΩΣΗ ΤΑΣΗΣ ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ Η ΜΝΗΜΗ ΤΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΑΥΤΗΣ



7 LD

ΜΝΗΜΗ ΘΕΡΜΟΚΡΑΣΙΑΣ 40C

ΕΦΟΣΟΝ ΕΧΕΙ ΠΑΤΗΘΕΙ ΤΟ ΚΟΥΜΠΙ ΤΗΣ ΣΥΓΚΡΕΚΡΙΜΕΝΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΛΕΝ ΕΠΙΤΡΕΠΕΤΑΙ ΝΑ ΕΠΙΛΕΧΘΕΙ ΚΑΜΜΙΑ ΑΛΛΗ ΘΕΡΜΟΚΡΑΣΙΑ, ΚΑΙ ΜΟΝΟ ΟΣΟ ΔΕΝ ΥΠΑΡΧΕΙ ΠΤΩΣΗ ΤΑΣΗΣ ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ Η ΜΝΗΜΗ ΤΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΑΥΤΗΣ



8 LD

ΜΝΗΜΗ ΘΕΡΜΟΚΡΑΣΙΑΣ 60C

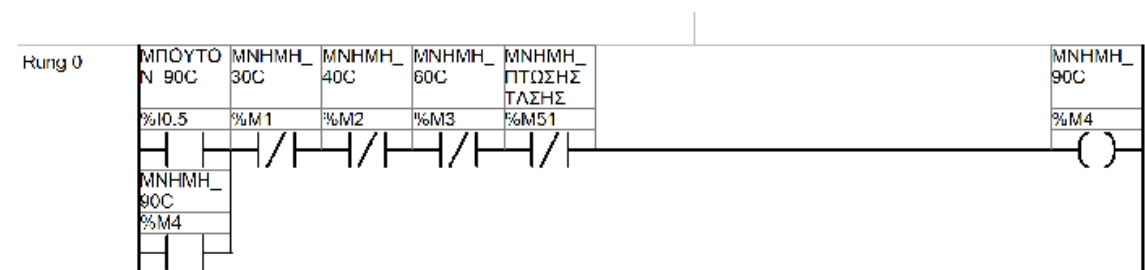
ΕΦΟΣΩΝ ΕΧΕΙ ΠΑΤΗΘΕΙ ΤΟ ΚΟΥΜΠΙ ΤΗΣ ΣΥΓΚΕΚΡΙΜΕΝΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΔΕΝ ΕΠΙΤΡΕΠΕΤΑΙ ΝΑ ΕΠΙΛΕΧΘΕΙ ΚΑΜΜΙΑ ΑΛΛΗ ΘΕΡΜΟΚΡΑΣΙΑ, ΚΑΙ ΜΟΝΟ ΟΣΟ ΔΕΝ ΥΠΑΡΧΕΙ ΠΤΩΣΗ ΤΑΣΗΣ ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ Η ΜΝΗΜΗ ΤΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΑΥΤΗΣ



9 LD

ΜΝΗΜΗ ΘΕΡΜΟΚΡΑΣΙΑΣ 90C

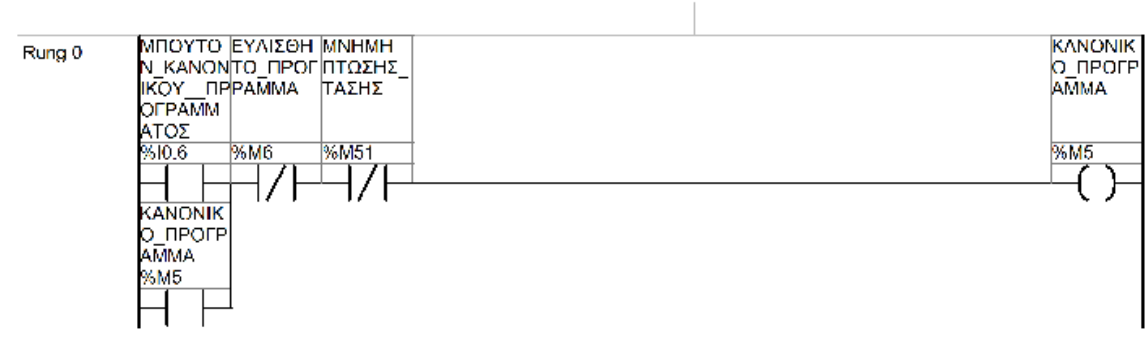
ΕΦΟΣΩΝ ΕΧΕΙ ΠΑΤΗΘΕΙ ΤΟ ΚΟΥΜΠΙ ΤΗΣ ΣΥΓΚΕΚΡΙΜΕΝΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΔΕΝ ΕΠΙΤΡΕΠΕΤΑΙ ΝΑ ΕΠΙΛΕΧΘΕΙ ΚΑΜΜΙΑ ΑΛΛΗ ΘΕΡΜΟΚΡΑΣΙΑ, ΚΑΙ ΜΟΝΟ ΟΣΟ ΔΕΝ ΥΠΑΡΧΕΙ ΠΤΩΣΗ ΤΑΣΗΣ ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ Η ΜΝΗΜΗ ΤΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΑΥΤΗΣ



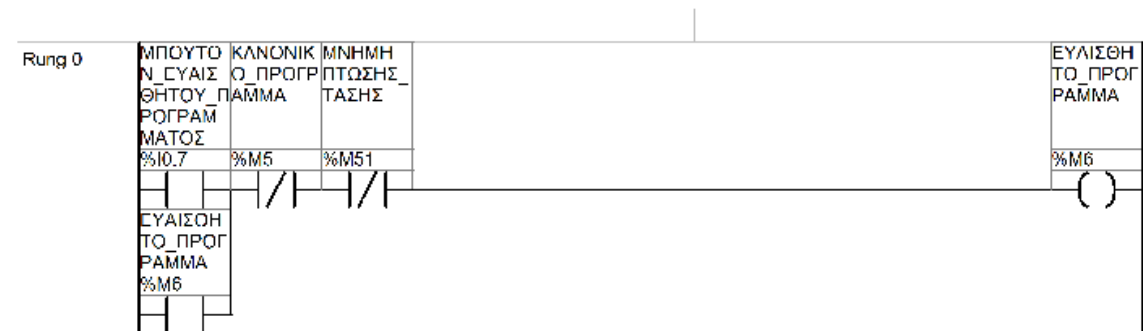
10 LD

ΜΝΗΜΗ ΚΑΝΟΝΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

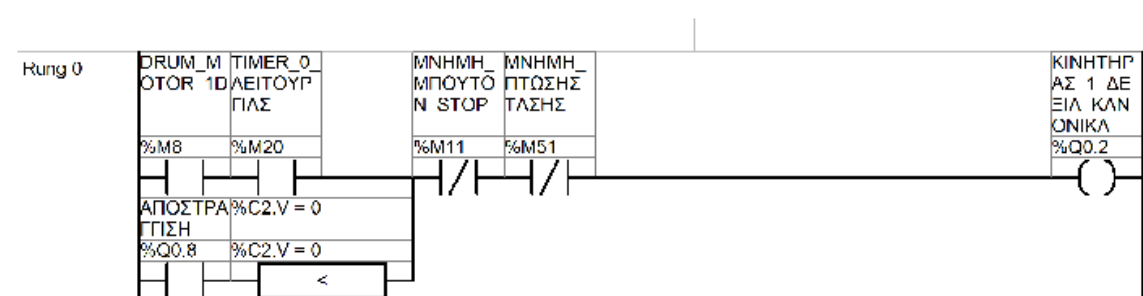
ΑΝ ΕΠΙΛΕΓΓΕΙ ΤΟ ΚΑΝΟΝΙΚΟ ΠΡΟΓΡΑΜΜΑ ΔΕΝ ΜΠΟΡΕΙ ΝΑ ΕΠΙΛΕΓΓΕΙ ΚΑΙ ΤΟ ΕΥΑΙΣΘΗΤΟ, ΚΑΙ ΑΠΟ ΤΗΝ ΣΤΙΓΜΗ ΠΟΥ ΔΕΝ ΥΠΑΡΧΕΙ ΠΤΩΣΗ ΤΑΣΗΣ ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ Η ΜΝΗΜΗ ΤΟΥ ΚΑΝΟΝΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ



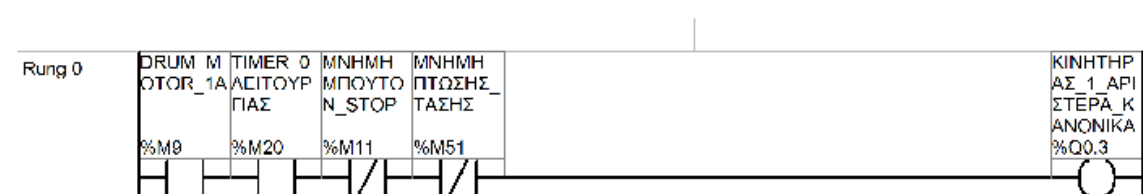
11	LD	ΜΝΗΜΗ ΕΥΑΙΣΘΗΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ	ΑΝ ΕΠΙΛΕΓΓΕΙ ΤΟ ΕΥΑΙΣΘΗΤΟ ΠΡΟΓΡΑΜΜΑ ΔΕΝ ΜΠΟΡΕΙ Ν ΑΡΠΑΞΕΙ ΚΑΙ ΤΟ ΚΑΝΟΝΙΚΟ. ΚΑΙ ΑΠΟ ΤΗΝ ΣΤΙΓΜΗ ΠΟΥ ΔΕΝ ΥΠΑΡΧΕΙ ΠΤΩΣΗ ΤΑΣΗΣ, ΤΟΤΕ ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ Η ΜΝΗΜΗ ΤΟΥ ΕΥΑΙΣΘΗΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ
-----------	-----------	--------------------------------------	---



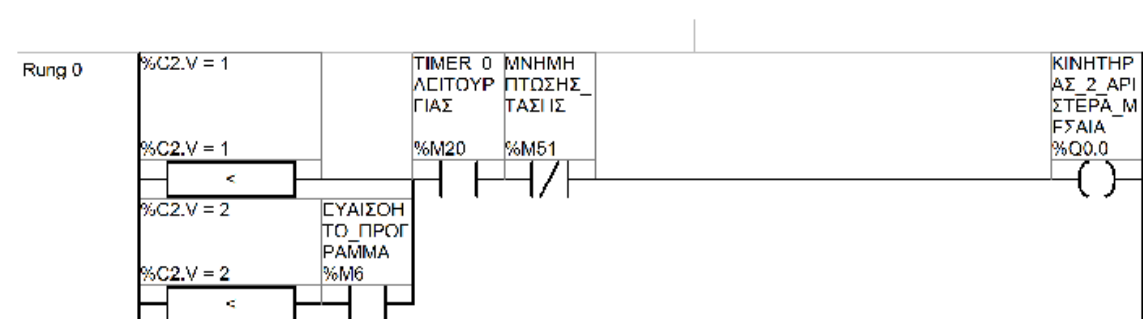
12	LD	ΚΙΝΗΤΗΡΑΣ 1 (ΔΕΞΙΟΣΤΡΟΦΗ ΚΙΝΗΣΗ ΜΕ ΚΑΝΟΝΙΚΗ ΤΑΧΥΤΗΤΑ)	ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΜΟΝΟ ΣΤΙΣ ΔΙΑΔΙΚΑΣΙΕΣ ΠΛΥΣΙΜΑΤΟΣ ΚΑΙ ΑΠΟΣΤΡΑΓΓΙΣΗΣ
-----------	-----------	--	---



13	LD	ΚΙΝΗΤΗΡΑΣ 1 (ΑΡΙΣΤΕΡΟΣΤΡΟΦΗ ΚΙΝΗΣΗ ΜΕ ΚΑΝΟΝΙΚΗ ΤΑΧΥΤΗΤΑ)	ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΜΟΝΟ ΣΤΙΣ ΔΙΑΔΙΚΑΣΙΕΣ ΠΛΥΣΙΜΑΤΟΣ
-----------	-----------	---	---



14	LD	ΚΙΝΗΤΗΡΑΣ 2 (ΑΡΙΣΤΕΡΟΣΤΡΟΦΗ ΚΙΝΗΣΗ ΜΕ ΜΕΣΑΙΑ ΤΑΧΥΤΗΤΑ)	ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΜΟΝΟ ΣΤΙΣ ΔΙΑΔΙΚΑΣΙΕΣ ΣΤΥΨΙΜΑΤΟΣ ΚΑΙ ΣΤΟ ΕΥΑΙΣΘΗΤΟ ΚΑΙ ΣΤΟ ΚΑΝΟΝΙΚΟ ΠΡΟΓΡΑΜΜΑ
-----------	-----------	---	--



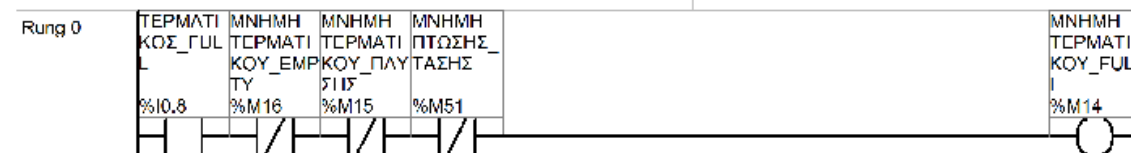
15	LD	ΚΙΝΗΤΗΡΑΣ 3 (ΑΡΙΣΤΕΡΟΣΤΡΟΦΗ ΚΙΝΗΣΗ ΜΕ ΓΡΟΓΟΡΗ ΤΑΧΥΤΗΤΑ)	ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΜΟΝΟ ΣΤΙΣ ΔΙΑΔΙΚΑΣΙΕΣ ΣΤΥΦΙΜΑΤΟΣ ΚΑΙ ΣΤΟ ΕΥΑΙΣΘΗΤΟ ΠΡΟΓΡΑΜΜΑ
-----------	-----------	--	---



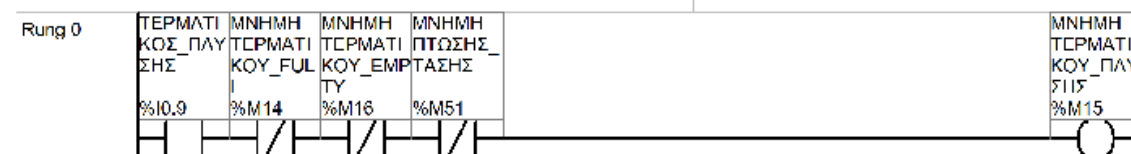
16	LD	ΜΝΗΜΗ ΤΕΡΜΑΤΙΚΟΥ ΠΟΡΤΑΣ	Η ΜΝΗΜΗ ΤΟΥ ΤΕΡΜΑΤΙΚΟΥ ΠΟΡΤΑΣ ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΜΟΝΟ ΣΤΗΝ ΠΕΡΙΠΤΩΣΗ ΠΟΥ ΕΧΕΙ ΕΝΕΡΓΟΠΟΙΗΘΕΙ Ο ΤΕΡΜΑΤΙΚΟΣ ΤΗΣ ΚΑΙ ΕΦΟΣΟΝ ΔΕΝ ΕΧΕΙ ΦΤΑΣΕΙ ΣΤΟ ΤΕΛΟΣ ΤΗΣ Η ΕΡΓΑΣΙΑ Ή ΔΕΝ ΕΧΕΙ ΠΑΘΗΘΕΙ ΤΟ STOP ΟΠΟΤΕ ΚΑΙ ΜΠΟΡΟΥΜΕ ΝΑ ΑΝΟΙΞΟΥΜΕ ΤΟΝ ΚΑΔΟ
-----------	-----------	--------------------------------	---



17	LD	ΜΝΗΜΗ ΤΕΡΜΑΤΙΚΟΥ FULL	Η ΜΝΗΜΗ ΤΟΥ ΤΕΡΜΑΤΙΚΟΥ FULL ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΕΦΟΣΟΝ ΕΧΕΙ ΕΝΕΡΓΟΠΟΙΗΘΕΙ Ο ΤΕΡΜΑΤΙΚΟΣ ΑΥΤΟΣ ΚΑΙ ΑΝ ΑΦΝ ΥΠΑΡΧΕΙ ΠΤΩΣΗ ΤΑΣΗΣ. ΣΥΓΧΡΟΝΩΣ ΔΕΝ ΕΠΙΤΡΕΠΕΤΑΙ ΝΑ ΕΝΕΡΓΟΠΟΙΗΘΕΙ ΚΑΙ Ο ΤΕΡΜΑΤΙΚΟΣ ΕΜΡΤΥ Ή Ο ΤΕΡΜΑΤΙΚΟΣ ΠΛΥΣΗΣ (ΑΝ ΚΑΙ ΥΠΟ ΚΑΝΟΝΙΚΕΣ ΣΥΝΘΗΚΕΣ ΘΑ ΔΙΝΑΝ ΣΗΜΑ ΑΥΤΟΜΑΤΑ)
-----------	-----------	------------------------------	--



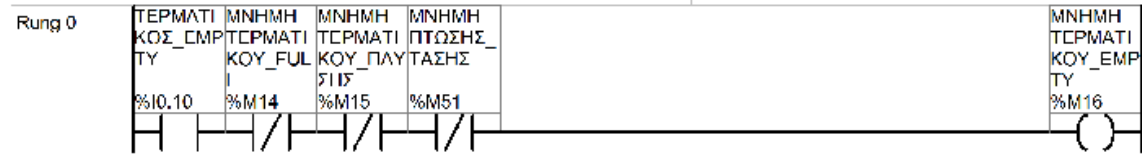
18	LD	ΜΝΗΜΗ ΤΕΡΜΑΤΙΚΟΥ ΠΛΥΣΗΣ	Η ΜΝΗΜΗ ΤΟΥ ΤΕΡΜΑΤΙΚΟΥ ΠΛΥΣΗΣ ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΕΦΟΣΟΝ ΕΧΕΙ ΕΝΕΡΓΟΠΟΙΗΘΕΙ Ο ΤΕΡΜΑΤΙΚΟΣ ΑΥΤΟΣ ΚΑΙ ΑΝ ΔΕΝ ΥΠΑΡΧΕΙ ΠΤΩΣΗ ΤΑΣΗΣ. ΣΥΓΧΡΟΝΩΣ ΔΕΝ ΕΠΙΤΡΕΠΕΤΑΙ ΝΑ ΕΝΕΡΓΟΠΟΙΗΘΕΙ ΚΑΙ Ο ΤΕΡΜΑΤΙΚΟΣ ΕΜΡΤΥ Ή Ο ΤΕΡΜΑΤΙΚΟΣ FULL
-----------	-----------	--------------------------------	---



19 LD

ΜΝΗΜΗ ΤΕΡΜΑΤΙΚΟΥ ΕΜΡΤΥ

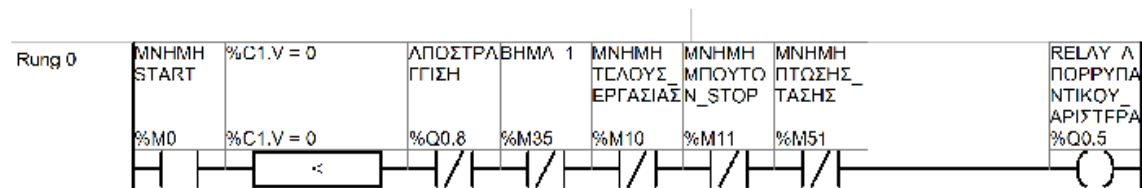
Η ΜΝΗΜΗ ΤΟΥ ΤΕΡΜΑΤΙΚΟΥ ΕΜΡΤΥ ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΕΦΟΣΩΝ ΕΧΕΙ ΕΝΕΡΓΟΠΟΙΗΘΕΙ Ο ΤΕΡΜΑΤΙΚΟΣ ΑΥΤΟΣ ΚΑΙ ΑΝ ΔΕΝ ΥΠΑΡΧΕΙ ΠΤΩΣΗ ΤΑΣΗΣ, ΣΥΓΧΡΟΝΩΣ ΔΕΝ ΕΠΙΤΡΕΠΕΤΑΙ ΝΑ ΕΝΕΡΓΟΠΟΙΗΘΕΙ ΚΑΙ Ο ΤΕΡΜΑΤΙΚΟΣ FULL Η Ο ΤΕΡΜΑΤΙΚΟΣ ΠΛΥΣΗΣ



20 LD

ΑΠΟΡΡΥΠΑΝΤΙΚΟ (ΑΡΙΣΤΕΡΗ ΘΕΣΗ)

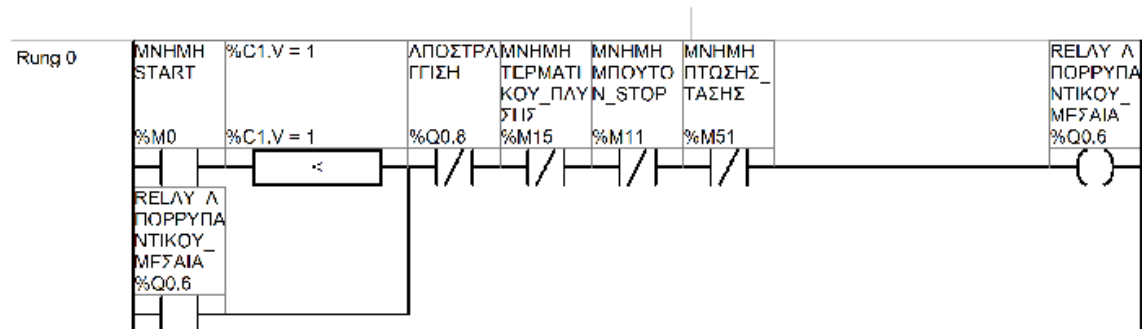
ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΜΟΝΟ ΑΝ ΦΙΝΑΙ ΕΝΕΡΓΟΠΟΙΗΜΕΝΟ ΤΟ ΒΗΜΑ 0 ΚΑΙ ΜΟΝΟ ΣΕ ΚΑΤΑΣΤΑΣΗ ΓΕΜΙΣΜΑΤΟΣ ΔΗΛΑΔΗ ΝΑ ΜΗΝ ΥΠΑΡΧΕΙ ΑΠΟΣΤΡΑΓΓΙΣΗ



21 LD

ΑΠΟΡΡΥΠΑΝΤΙΚΟ (ΜΕΣΑΙΑ ΘΕΣΗ)

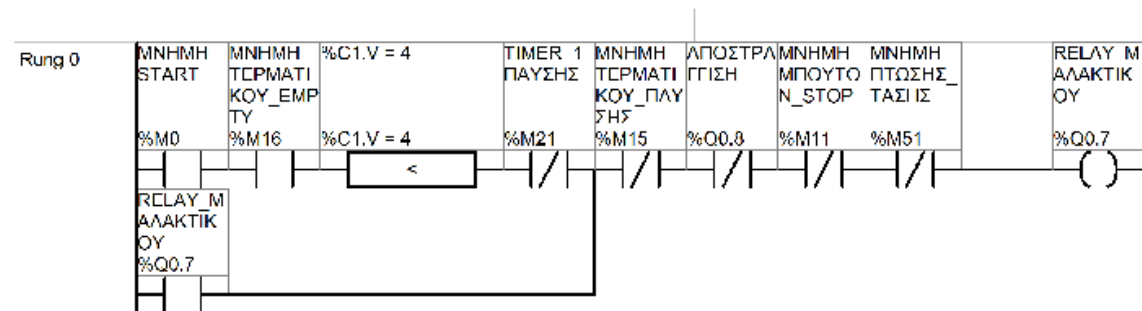
ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΜΟΝΟ ΑΝ ΦΙΝΑΙ ΕΝΕΡΓΟΠΟΙΗΜΕΝΟ ΤΟ ΒΗΜΑ 1 ΚΑΙ ΜΟΝΟ ΣΕ ΚΑΤΑΣΤΑΣΗ ΓΕΜΙΣΜΑΤΟΣ ΔΗΛΑΔΗ ΝΑ ΜΗΝ ΥΠΑΡΧΕΙ ΑΠΟΣΤΡΑΓΓΙΣΗ



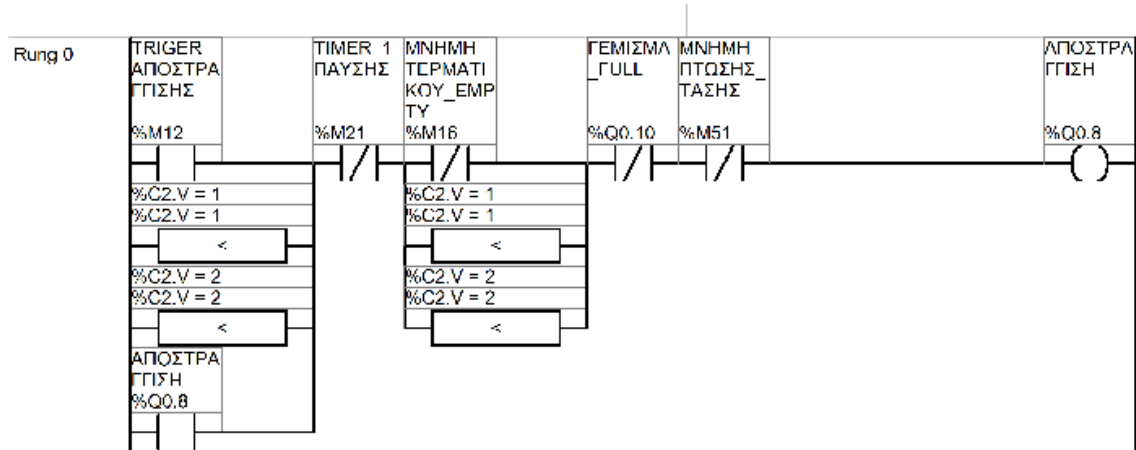
22 LD

ΜΑΛΑΚΤΙΚΟ (ΔΕΞΙΑ ΘΕΣΗ)

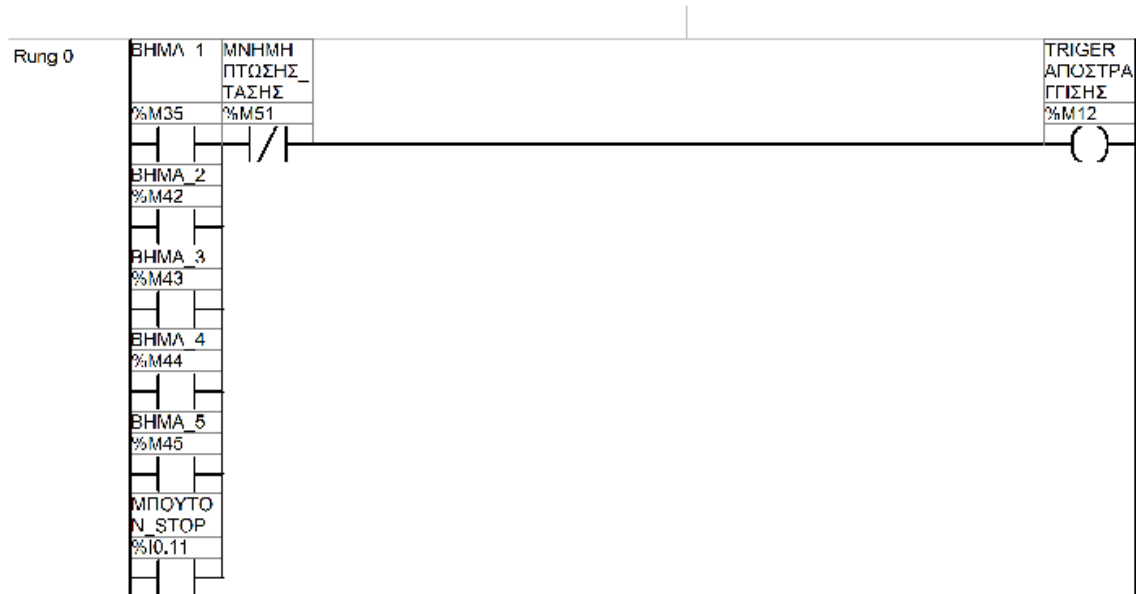
ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΜΟΝΟ ΑΝ ΦΙΝΑΙ ΕΝΕΡΓΟΠΟΙΗΜΕΝΟ ΤΟ ΒΗΜΑ 4 (ΝΑ ΕΧΕΙ ΤΕΛΙΟΣΕΙ Ο TIMER ΠΛΥΣΗΣ ΤΗΝ ΧΡΟΝΟΜΕΤΡΗΣΗ ΤΟΥ) ΚΑΙ ΜΟΝΟ ΣΕ ΚΑΤΑΣΤΑΣΗ ΓΕΜΙΣΜΑΤΟΣ ΔΗΛΑΔΗ ΝΑ ΜΗΝ ΥΠΑΡΧΕΙ ΑΠΟΣΤΡΑΓΓΙΣΗ.



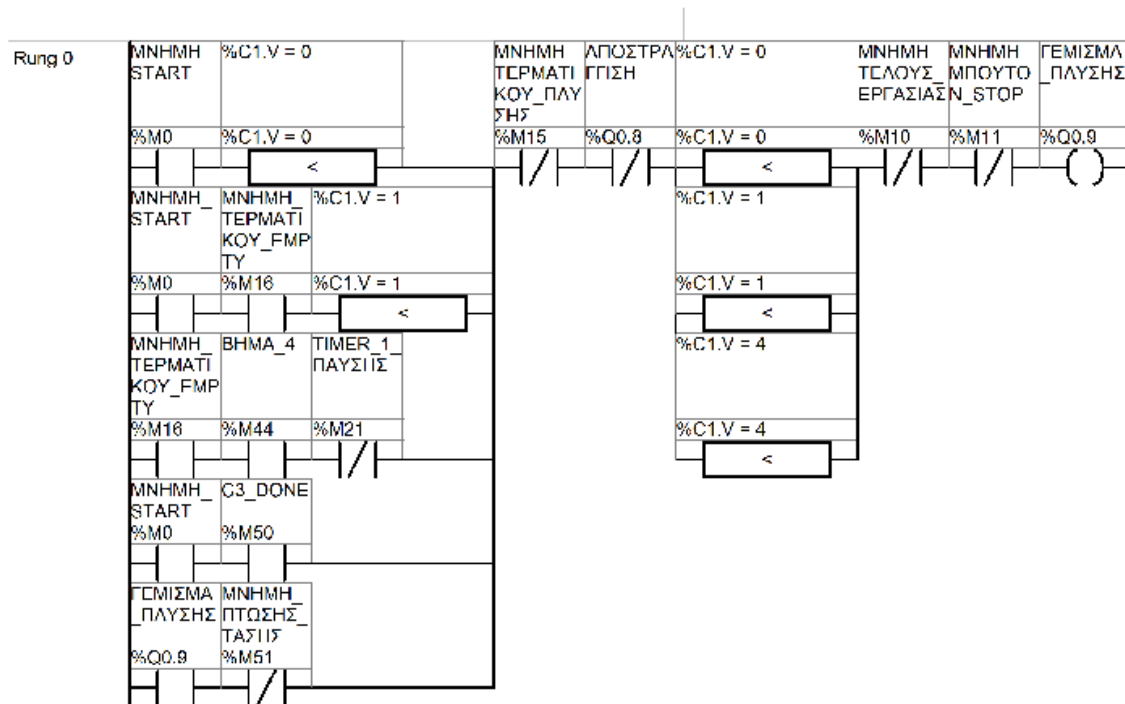
23 LD	ΑΠΟΣΤΡΑΓΓΙΣΗ	Η ΑΠΟΣΤΡΑΓΓΙΣΗ ΛΕΙΤΟΥΡΓΕΙ ΕΚΤΟΣ ΑΠΟ ΤΟ TRIGGER ΤΗΣ ΚΑΙ ΣΕ ΠΕΡΙΠΤΩΣΗ ΣΤΥΦΙΜΑΤΟΣ (ΜΕ ΑΥΤΟ ΕΞΑΣΦΑΛΙΖΟΥΜΕ ΟΤΙ Η ΑΠΟΣΤΡΑΓΓΙΣΗ ΛΕΙΤΟΥΡΓΕΙ ΚΑΙ ΚΑΙΑ Η ΔΙΑΔΙΚΑΣΙΑ ΤΟΥ ΤΕΛΕΥΤΑΙΟΥ ΣΤΥΦΙΜΑΤΟΣ) ΛΟΓΩ ΤΟΥ ΟΤΙ Η ΔΙΑΔΙΚΑΣΙΑ ΕΙΝΑΙ ΧΕΙΡΟΚΙΝΗΤΗ ΕΧΟΥΝ ΤΟΠΟΣΤΗΟΙ ΚΑΙ ΕΠΙΠΛΕΟΝ ΣΥΝΟΗΚΕΣ ΓΙΑ ΤΗΝ ΕΞΑΣΦΑΛΙΣΗ ΣΩΣΤΗΣ ΛΕΙΤΟΥΡΓΙΑΣ ΜΕΧΡΙ ΤΙ ΗΝ ΑΠΕΝΕΡΓΟΠΟΙΗΣΗ ΚΑΙ ΕΝΕΡΓΟΠΟΙΗΣΗ ΤΟΥ
--------------	---------------------	--



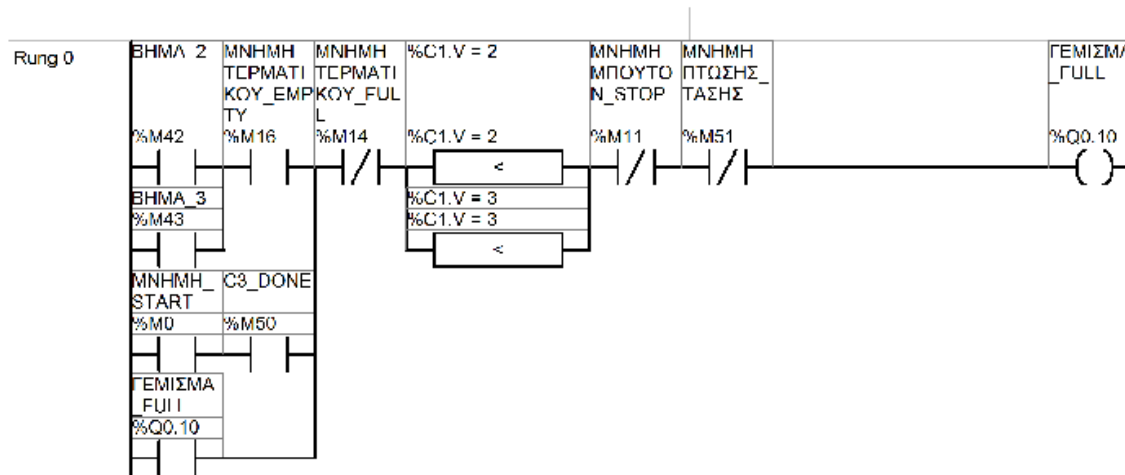
24 LD	TRIGGER ΑΠΟΣΤΡΑΓΓΙΣΗΣ	ΓΙΑ ΝΑ ΕΝΕΡΓΟΠΟΙΗΘΕΙ ΤΟ TRIGGER ΤΗΣ ΑΠΟΣΤΡΑΓΓΙΣΗΣ ΘΑ ΠΡΕΠΕΙ ΝΑ ΕΧΕΙ ΕΝΕΡΓΟΠΟΙΗΘΕΙ ΕΝΑ ΑΠΟ ΤΑ 5 ΒΗΜΑΤΑ ΤΟΥ DRUM Η ΝΑ ΕΧΕΙ ΠΑΤΗΘΕΙ ΤΟ STOP ΚΑΙ ΦΥΣΙΚΑ ΝΑ ΜΗΝ ΥΠΑΡΧΕΙ ΠΤΩΣΗ ΤΑΣΗΣ
--------------	------------------------------	--



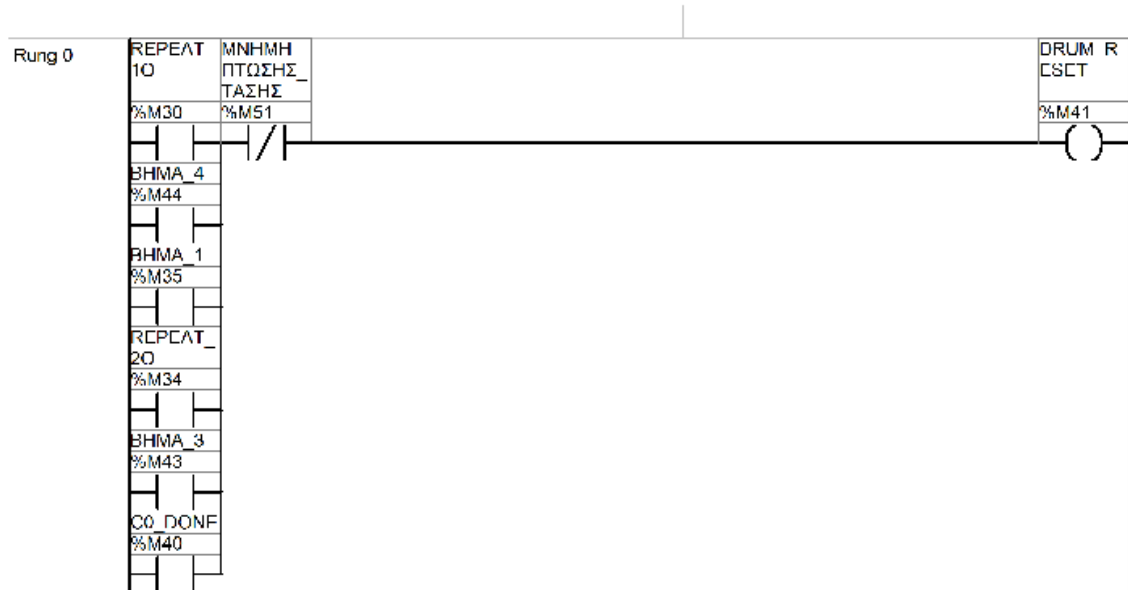
25 LD	ΓΕΜΙΣΜΑ ΠΛΥΣΗΣ	ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΣΤΑ ΣΤΑΔΙΑ ΠΟΥ ΑΠΑΙΤΕΙΤΑΙ ΝΑ ΕΧΟΥΜΕ ΠΛΗΡΩΣΗ ΤΟΥ ΚΑΔΟΥ
--------------	-----------------------	--



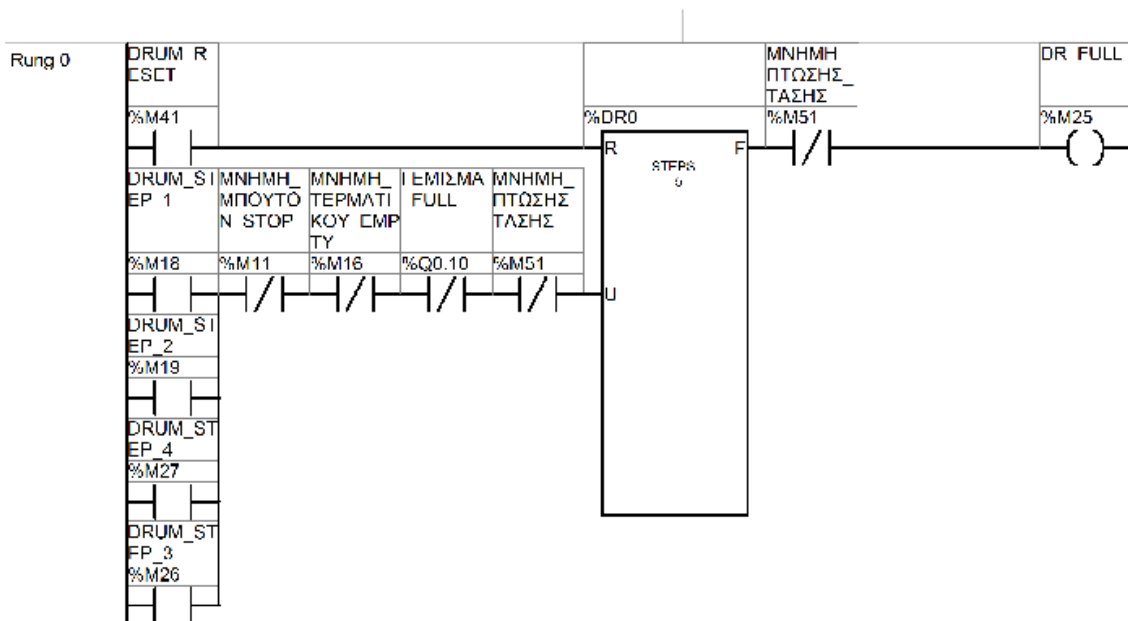
26 LD	ΓΕΜΙΣΜΑ FULL	ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΣΤΑ ΣΤΑΔΙΑ ΠΟΥ ΑΠΑΙΤΕΙΤΑΙ ΝΑ ΕΧΟΥΜΕ ΠΛΗΡΩΣΗ ΤΟΥ ΚΑΔΟΥ ΓΙΑ ΤΟ ΞΕΒΓΑΛΜΑ
--------------	---------------------	--



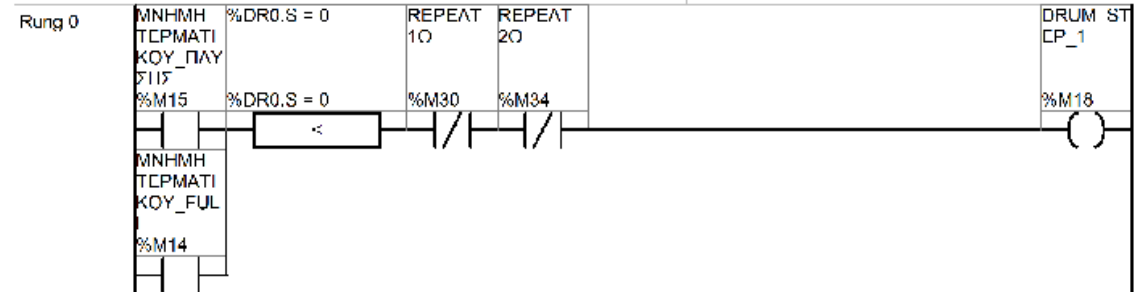
27	LD	DRUM RESET	ΕΙΝΑΙ ΟΙ ΑΝΑΓΚΑΙΕΣ ΣΥΝΘΗΚΕΣ ΓΙΑ ΤΗΝ ΕΠΑΝΕΚΚΙΝΗΣΗ ΤΟΥ DRUM ΑΦΟΥ ΕΧΕΙ ΕΚΤΕΛΕΣΕΙ ΤΗΝ ΛΕΙΤΟΥΡΓΙΑ ΠΟΥ ΚΛΗΘΗΚΕ ΝΑ ΚΑΝΕΙ
-----------	-----------	-------------------	---



28	LD	DRUM	ΒΑΣΙΚΗ ΛΕΙΤΟΥΡΓΙΑ: ΕΝΑΛΛΑΓΗ ΚΙΝΗΤΗΡΩΝ ΜΕΤΑ ΑΠΟ ΠΛΥΣΗ ΚΑΤΑ ΤΗΝ ΔΙΑΡΚΕΙΑ ΤΩΝ ΠΛΥΣΕΩΝ
-----------	-----------	-------------	--



29	LD	ΣΥΝΘΗΚΗ ΠΡΩΤΟΥ ΒΗΜΑΤΟΣ DRUM	ΠΡΕΠΕΙ Ο DRUM ΝΑ ΒΡΙΣΚΕΤΑΙ ΣΤΗ ΚΑΤΑΣΤΑΣΗ 0 ΚΑΙ Ο ΚΑΛΟΣ ΝΑ ΕΙΝΑΙ ΓΕΜΑΤΟΣ ΜΕΧΡΙ ΤΟ ΕΠΙΘΥΜΗΤΟ ΚΑΘΕ ΦΟΡΑ ΣΗΜΕΙΟ (ΤΑ REPEAT ΧΡΗΣΙΜΟΠΟΙΟΥΝΤΑΙ ΓΙΑ ΤΗΝ ΔΗΜΙΟΥΡΓΙΑ ΠΑΛΜΟΥ ΩΣΤΕ ΟΤΑΝ ΧΡΕΙΑΣΤΕΙ ΝΑ ΕΚΤΕΛΕΣΤΕΙ Η ΔΙΑΔΙΚΑΣΙΑ 2 Η 3 ΦΟΡΕΣ)
-----------	-----------	------------------------------------	---



30	LD	ΣΥΝΘΗΚΗ ΔΕΥΤΕΡΟΥ ΒΗΜΑΤΟΣ DRUM	ΠΡΕΠΕΙ Ο DRUM ΝΑ ΒΡΙΣΚΕΤΑΙ ΣΤΗΝ ΚΑΤΑΣΤΑΣΗ 1 ΚΑΙ ΝΑ ΕΧΕΙ ΤΕΛΕΙΩΣΕΙ Η ΚΙΝΗΣΗ ΤΟΥ ΚΙΝΗΤΗΡΑ (ΤΕΛΟΣ COUNTER)
-----------	-----------	--------------------------------------	--



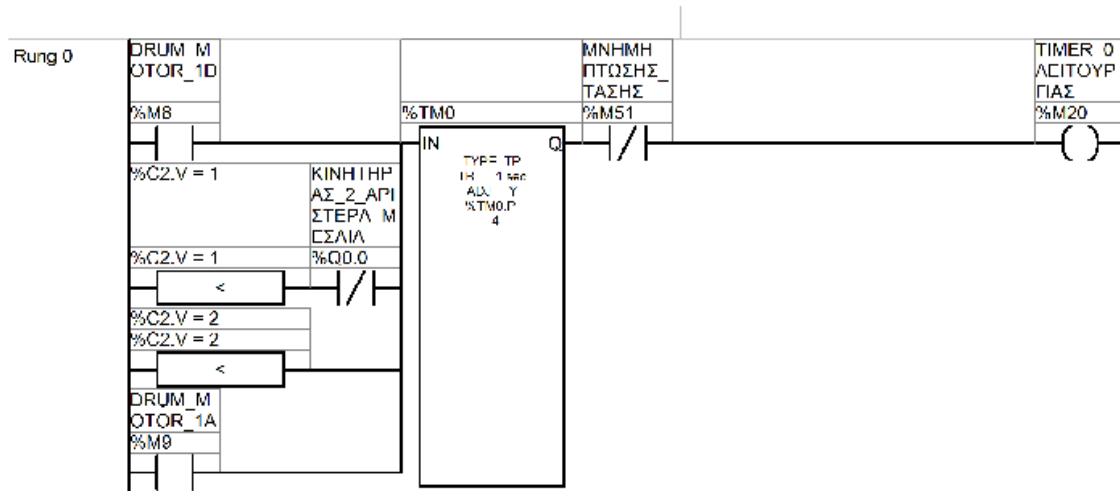
31	LD	ΣΥΝΘΗΚΗ ΤΡΙΤΟΥ ΒΗΜΑΤΟΣ DRUM	ΠΡΕΠΕΙ Ο DRUM ΝΑ ΒΡΙΣΚΕΤΑΙ ΣΤΗΝ ΚΑΤΑΣΤΑΣΗ 2 ΚΑΙ ΝΑ ΕΧΕΙ ΤΕΛΕΙΩΣΕΙ Η ΠΛΑΥΣΗ ΤΟΥ ΚΙΝΗΤΗΡΑ (ΤΕΛΟΣ COUNTER)
-----------	-----------	------------------------------------	--



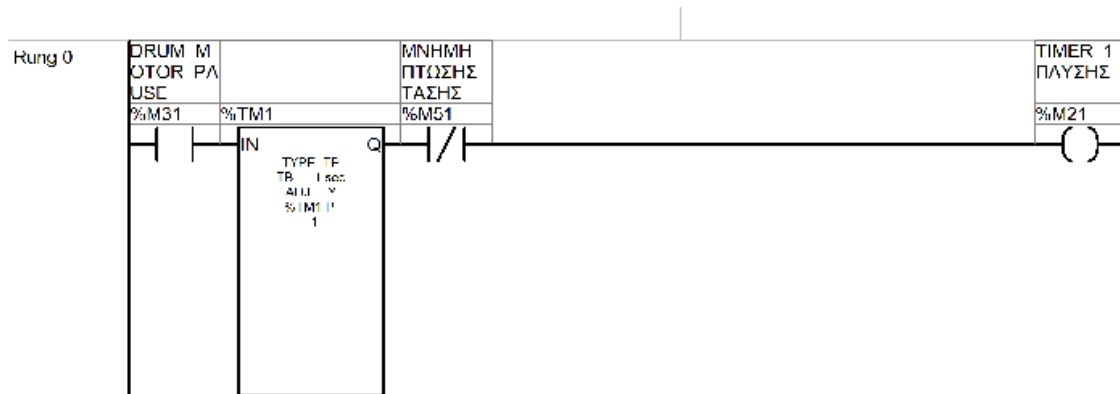
32	LD	ΣΥΝΘΗΚΗ ΤΕΤΑΡΤΟΥ ΒΗΜΑΤΟΣ DRUM	ΠΡΕΠΕΙ Ο DRUM ΝΑ ΒΡΙΣΚΕΤΑΙ ΣΤΗΝ ΚΑΤΑΣΤΑΣΗ 3 ΚΑΙ ΝΑ ΕΧΕΙ ΤΕΛΕΙΩΣΕΙ Η ΚΙΝΗΣΗ ΤΟΥ ΚΙΝΗΤΗΡΑ (ΤΕΛΟΣ COUNTER)
-----------	-----------	--------------------------------------	--



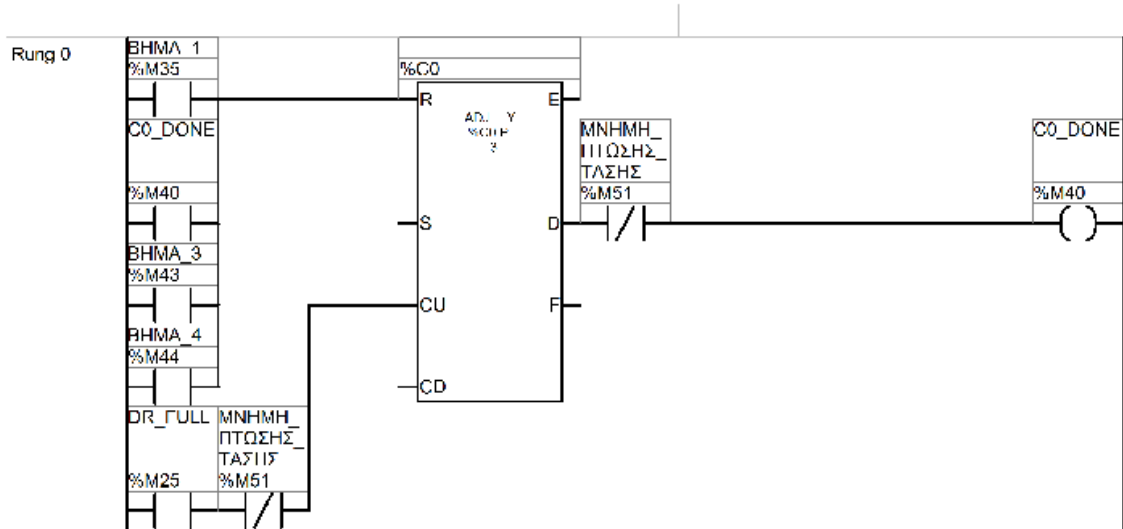
33	LD	TIMER ΛΕΙΤΟΥΡΓΙΑΣ	ΚΑΛΕΙΤΑΙ ΑΠΟ ΤΟΝ DRUM ΣΤΟ ΑΝΑΛΟΓΟ ΒΗΜΑ, ΑΛΛΑ ΚΑΙ ΣΤΟ ΣΤΥΨΙΜΟ ΓΙΑ ΤΗΝ ΧΡΟΝΟΜΕΤΡΗΣΗ ΤΗΣ ΠΕΡΙΣΤΡΟΦΗΣ ΤΟΥ ΚΑΤΑΛΛΗΛΟΥ ΚΙΝΗΤΗΡΑ
-----------	-----------	--------------------------	---



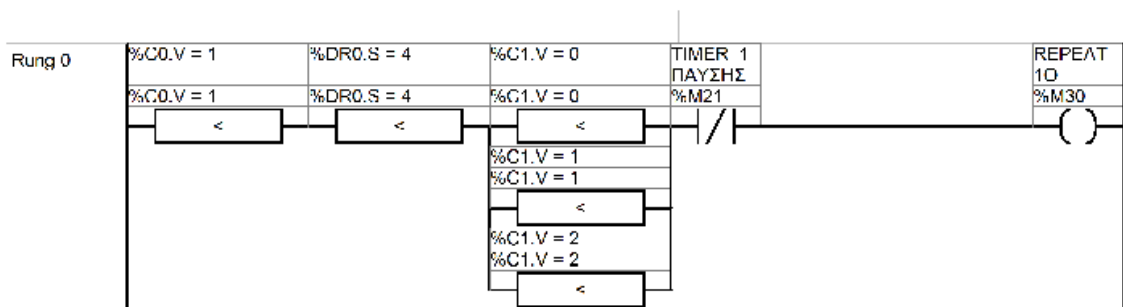
34	LD	TIMER ΠΑΥΣΗΣ	ΚΑΛΕΙΤΕ ΑΠΟ ΤΟΝ DRUM ΣΤΑ ΑΝΑΛΟΓΑ ΒΗΜΑΤΑ ΓΙΑ ΤΗΝ ΧΡΟΝΟΜΕΤΡΗΣΗ ΤΗΣ ΠΑΥΣΗΣ ΤΗΣ ΠΕΡΙΣΤΡΟΦΗΣ ΤΟΥ ΚΙΝΗΤΗΡΑ 1 ΕΝΔΙΑΜΕΣΑ ΤΗΣ ΕΝΑΛΛΑΓΗΣ ΔΙΕΥΘΥΝΣΗΣ ΑΥΤΗΣ
-----------	-----------	---------------------	---



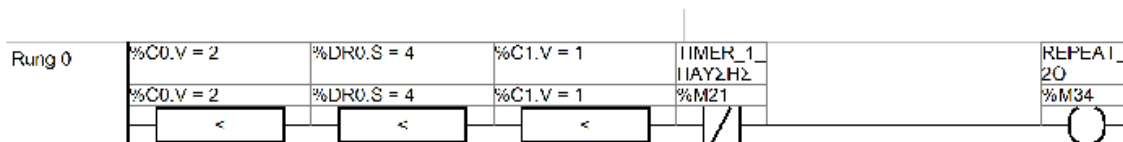
35 LD	ΜΕΤΡΗΤΗΣ ΕΠΑΝΑΛΗΨΕΩΝ ΤΟΥ DRUM	ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΜΟΝΟ ΣΤΙΣ ΠΕΡΙΠΤΩΣΕΙΣ ΠΟΥ Ο DRUM ΠΡΕΠΕΙ ΝΑ ΕΚΤΕΛΑΣΤΕΙ ΠΑΡΑΠΑΝΩ ΑΠΟ ΜΙΑ ΦΟΡΕΣ (ΚΑΙ ΑΦΟΥ ΕΧΕΙ ΤΕΛΕΙΩΣΕΙ ΜΙΑ ΦΟΡΑ ΤΗΝ ΔΙΑΔΙΚΑΣΙΑ ΤΟΥ)
--------------	--------------------------------------	---



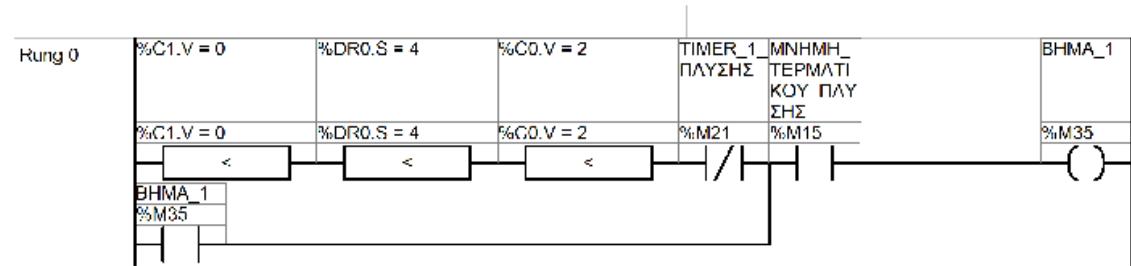
36 LD	1ο REPEAT	ΣΥΝΘΕΤΕΣ ΓΙΑ ΝΑ ΓΙΝΕΙ Η ΠΡΩΤΗ ΕΠΑΝΑΛΗΨΗ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΤΟΥ DRUM, ΔΗΛΑΔΗ, ΣΤΑ ΒΗΜΑΤΑ 0 ΚΑΙ 1(ΚΑΙ ΑΦΟΥ ΕΧΕΙ ΓΙΝΕΙ ΜΙΑ ΦΟΡΑ Η ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ)
--------------	------------------	---



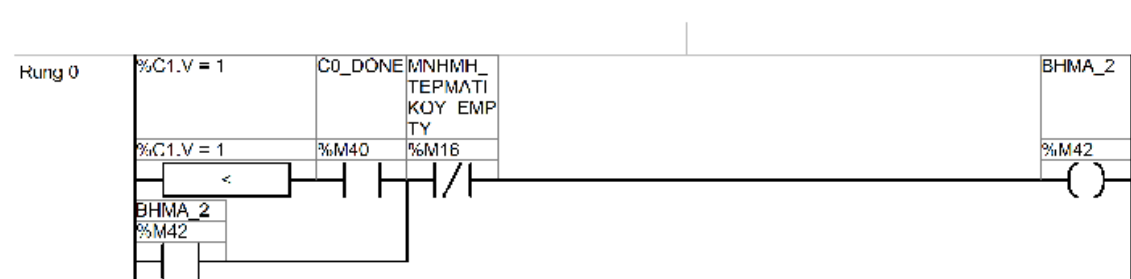
37 LD	2ο REPEAT	ΓΙΑ ΝΑ ΕΠΑΝΑΛΗΨΕΙ ΚΑΙ ΔΕΥΤΕΡΗ ΦΟΡΑ Η ΔΙΑΔΙΚΑΣΙΑ ΤΟΥ DRUM (ΒΗΜΑ 1 ΚΥΡΙΑ ΠΑΥΣΗ) ΚΑΙ ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΜΟΝΟ ΕΦΟΣΩΝ ΕΧΕΙ ΕΝΕΡΓΟΠΟΙΗΘΕΙ Η ΠΡΩΤΗ ΕΠΑΝΑΛΗΨΗ
--------------	------------------	---



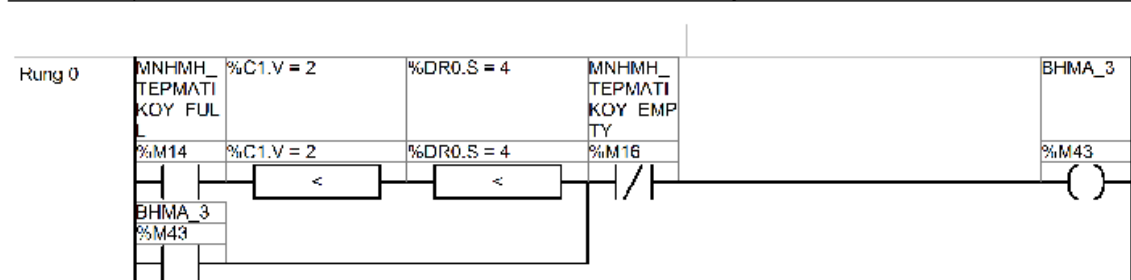
38	LD	ΒΗΜΑ 1 (ΜΕΤΑΒΑΣΗ ΑΠΟ ΠΡΟΠΛΥΣΗ ΣΕ ΚΥΡΙΩΣ ΠΛΥΣΗ)	ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΑΠΟ ΤΗΝ ΣΤΙΓΜΗ ΠΟΥ ΤΕΛΕΙΩΣΕΙ Η ΠΡΟΠΛΥΣΗ (ΕΧΕΙ ΓΙΝΕΙ ΛΥΟ ΦΟΡΕΣ Η ΔΙΑΔΙΚΑΣΙΑ ΤΟΥ DRUM) ΚΑΙ ΜΕΤΑΒΑΙΝΕΙ ΣΕ ΑΠΟΣΤΡΑΓΓΙΣΗ
-----------	-----------	---	--



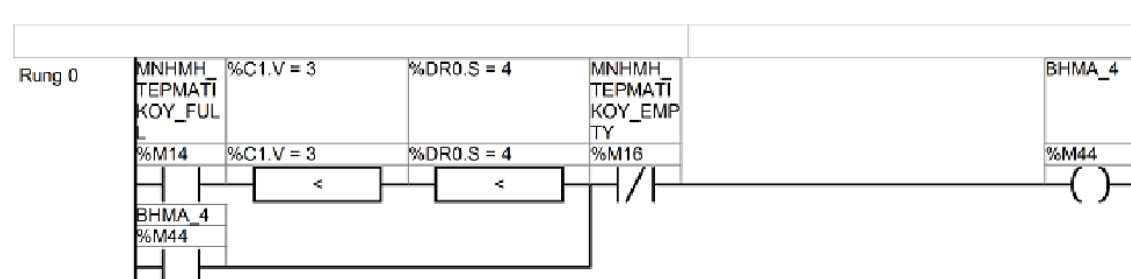
39	LD	ΒΗΜΑ 2 (ΜΕΤΑΒΑΣΗ ΑΠΟ ΚΥΡΙΩΣ ΠΛΥΣΗ ΣΕ ΠΡΩΤΟ ΞΕΒΓΑΛΜΑ)	ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΑΠΟ ΤΗΝ ΣΤΙΓΜΗ ΠΟΥ ΤΕΛΕΙΩΣΕΙ Η ΚΥΡΙΩΣ ΠΛΥΣΗ (ΕΧΕΙ ΓΙΝΕΙ ΤΡΕΙΣ ΦΟΡΕΣ Η ΔΙΑΔΙΚΑΣΙΑ ΤΟΥ DRUM) ΚΑΙ ΜΕΤΑΒΑΙΝΕΙ ΣΕ ΑΠΟΣΤΡΑΓΓΙΣΗ
-----------	-----------	---	--



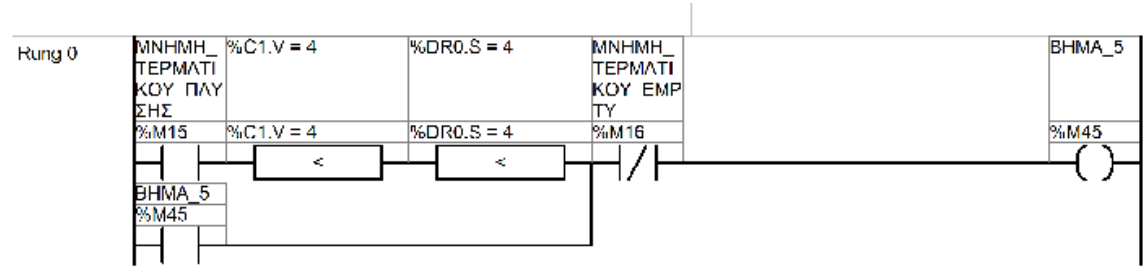
40	LD	ΒΗΜΑ 3 (ΜΕΤΑΒΑΣΗ ΑΠΟ ΠΡΩΤΟ ΞΕΒΓΑΛΜΑ ΣΕ ΔΕΥΤΕΡΟ ΞΕΒΓΑΛΜΑ)	ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΟΤΑΝ ΤΕΛΕΙΩΣΕΙ ΤΕΛΕΙΩΣ ΤΟ ΠΡΩΤΟ ΞΕΒΓΑΛΜΑ. ΠΡΟΥΠΟΘΕΣΗ ΓΙΑ ΞΕΒΓΑΛΜΑ ΕΙΝΑΙ ΝΑ ΕΙΝΑΙ ΓΕΜΑΤΟΣ Ο ΚΑΔΟΣ
-----------	-----------	---	---



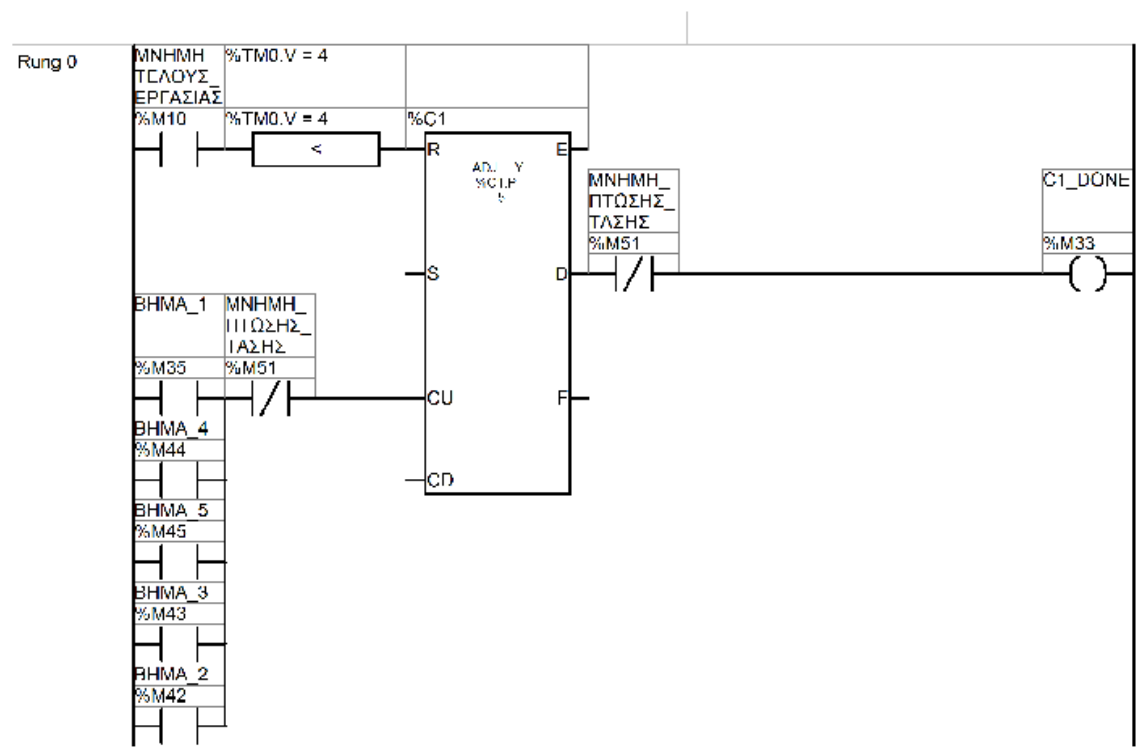
41	LD	ΒΗΜΑ 4 (ΜΕΤΑΒΑΣΗ ΑΠΟ ΔΕΥΤΕΡΟ ΞΕΒΓΑΛΜΑ ΣΕ ΠΛΥΣΗ ΜΕ ΜΑΛΑΚΤΙΚΟ)	ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΟΤΑΝ ΤΕΛΕΙΩΣΕΙ ΤΕΛΕΙΩΣ ΤΟ ΔΕΥΤΕΡΟ ΞΕΒΓΑΛΜΑ. ΠΡΟΥΠΟΘΕΣΗ ΓΙΑ ΞΕΒΓΑΛΜΑ ΕΙΝΑΙ ΝΑ ΕΙΝΑΙ ΓΕΜΑΤΟΣ ΚΑΔΟΣ
-----------	-----------	---	---



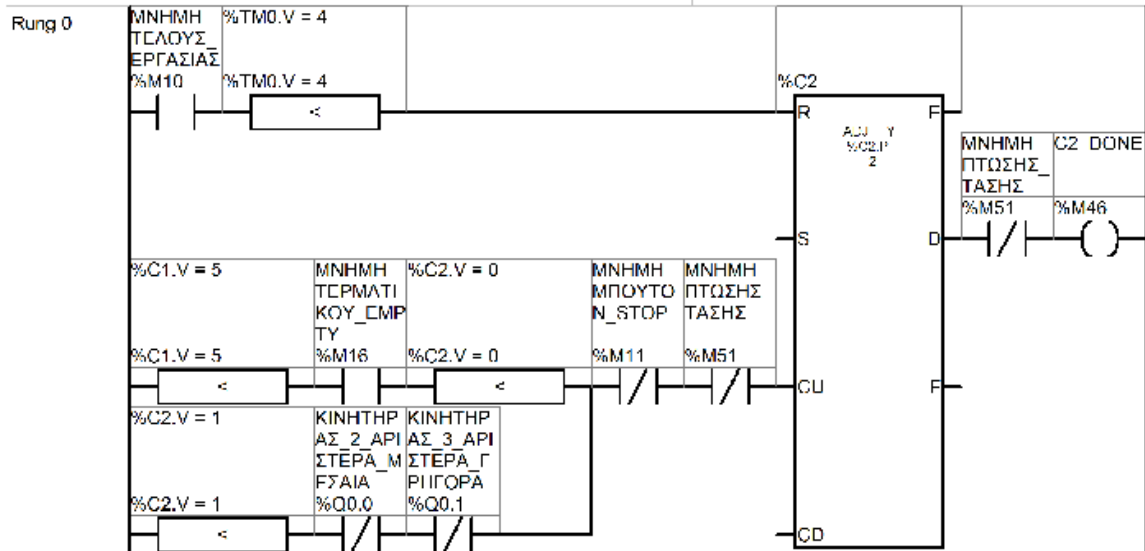
42 LD **ΒΗΜΑ 5 (ΜΕΤΑΒΑΣΗ ΑΠΟ ΠΛΥΣΗ ΜΕ ΜΑΛΑΚΤΙΚΟ ΣΕ ΣΤΥΨΙΜΟ)** **ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΟΤΑΝ ΤΡΑΦΙΟΣΤΕΙ ΤΡΑΦΙΟΣ Η ΠΛΥΣΗ ΜΕ ΜΑΛΑΚΤΙΚΟ. ΠΡΟΥΠΟΘΕΣΗ ΓΙΑ ΜΑΛΑΚΤΙΚΟ ΕΙΝΑΙ ΝΑ ΕΙΝΑΙ ΓΕΜΑΤΟΣ Ο ΚΑΔΟΣ.**



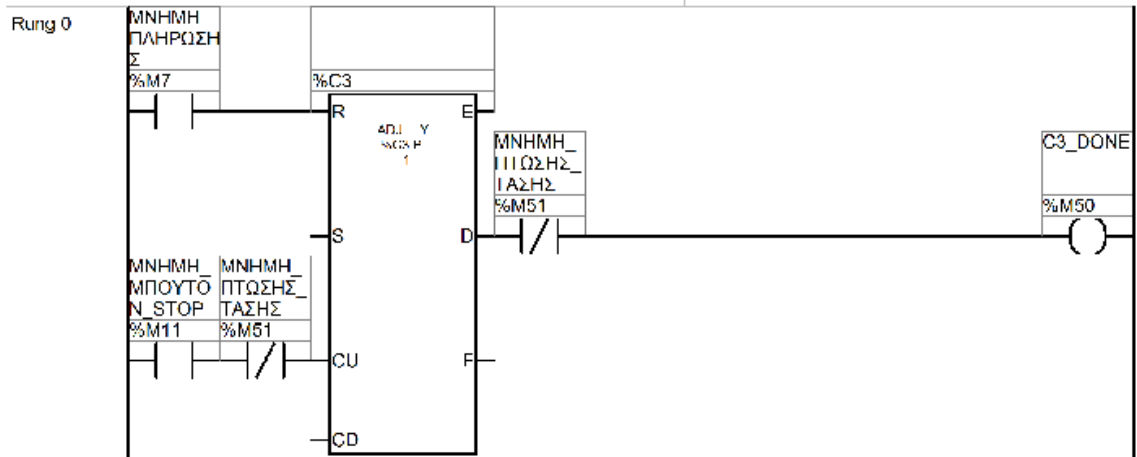
43 LD **COUNTER ΜΕΤΡΗΣΗΣ ΤΟΥ ΚΑΘΕ ΒΗΜΑΤΟΣ ΤΗΣ ΛΕΙΤΟΥΡΓΙΑΣ** **ΤΟ RESET ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΜΟΝΟ ΟΤΑΝ ΕΧΕΙ ΤΡΑΦΙΟΣΤΕΙ ΤΟ ΣΤΥΨΙΜΟ ΚΑΙ Η ΕΡΓΑΣΙΑ ΤΟΥ ΕΙΝΑΙ ΝΑ ΣΗΜΑΤΟΔΟΤΕΙ ΤΗΝ ΜΕΤΑΠΗΔΗΣΗ ΑΠΟ ΒΗΜΑ ΣΕ ΒΗΜΑ ΤΩΝ ΔΙΑΔΙΚΑΣΙΩΝ ΤΟΥ ΠΛΥΝΤΗΡΙΟΥ**



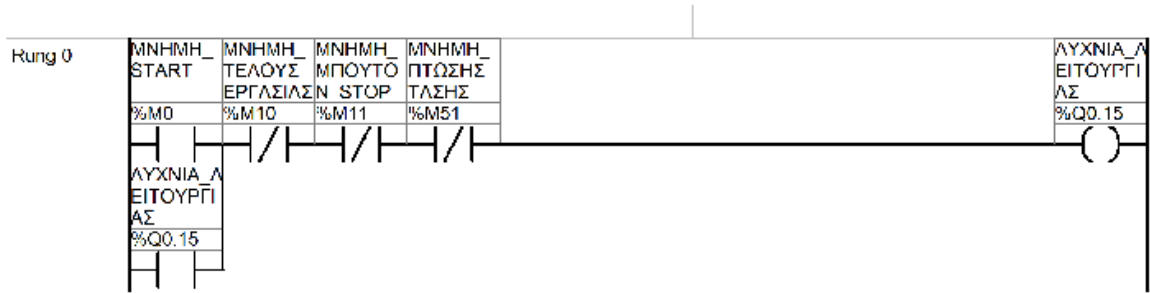
44 LD	COUNTER ΕΝΑΛΛΑΓΗΣ ΚΙΝΗΤΗΡΩΝ ΣΤΥΨΙΜΑΤΟΣ	ΤΟ RESET ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΜΟΝΟ ΟΤΑΝ ΕΧΕΙ ΤΕΛΕΙΩΣΕΙ ΤΟ ΣΤΥΨΙΜΟ ΚΑΙ Η ΕΡΓΑΣΙΑ ΤΟΥ ΕΙΝΑΙ ΝΑ ΣΗΜΑΤΟΔΟΤΕΙ ΤΗΝ ΑΚΑΡΙΑΙΑ ΕΝΑΛΛΑΓΗ ΤΩΝ ΚΙΝΗΤΗΡΩΝ ΣΤΥΨΙΜΑΤΟΣ
--------------	---	---



45 LD	COUNTER MANUAL ΠΑΥΣΗΣ	ΟΣΟ ΕΙΝΑΙ ΕΝΕΡΓΟΠΟΙΗΜΕΝΗ Η ΜΝΗΜΗ ΠΛΗΡΩΣΗΣ ΒΡΙΣΚΕΤΑΙ ΜΟΝΙΜΩΣ ΣΕ RESET ΚΑΙ Ο COUNTER ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΜΟΝΟ ΟΤΑΝ ΧΡΕΙΑΣΤΕΙ ΔΗΛΑΔΗ ΟΤΑΝ ΠΑΤΙΝΘΕΙ ΤΟ STOP ΚΑΙ ΔΕΝ ΕΧΕΙ ΠΙΕΣΕΙ Η ΤΑΣΗ. ΥΠΑΡΧΕΙ ΜΙΑ ΚΛΕΙΣΤΗ ΕΠΑΦΗ ΤΗΣ ΜΝΗΜΗΣ ΠΤΩΣΗΣ ΓΙΑΤΙ ΕΙΝΑΙ ΤΕΧΝΗΤΗ ΠΤΩΣΗ ΤΑΣΗΣ. Ο COUNTER ΑΥΤΟΣ ΟΥΣΙΑΣΤΙΚΑ ΜΕΤΡΑΕΙ ΑΠΟ ΤΟ 0 ΜΕΧΡΙ ΤΟ 1 ΕΤΣΙ ΩΣΤΕ ΝΑ ΜΠΟΡΩ ΝΑ ΠΑΤΙΣΩ ΤΟ STOP ΟΣΕΣ ΦΟΡΕΣ ΧΡΕΙΑΣΤΕΙ (ΜΕ ΑΠΛΗ ΜΝΗΜΗ ΘΑ ΧΡΕΙΑΖΟΤΑΝ ΠΟΛΥΠΛΟΚΗ ΔΙΑΔΙΚΑΣΙΑ) ΚΑΙ ΜΕΤΡΑΕΙ BUTTON CLICK
--------------	------------------------------	--



46 LD	ΛΥΧΝΙΑ ΛΕΙΤΟΥΡΓΙΑΣ	Η ΛΥΧΝΙΑ ΛΕΙΤΟΥΡΓΙΑΣ ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΑΝ ΕΧΕΙ ΠΑΘΘΕΙ ΤΟ START ΚΑΙ ΑΠΕΝΕΡΓΟΠΟΙΕΙΤΑΙ ΟΤΑΝ ΕΧΕΙ ΦΤΑΣΕΙ ΤΟ ΤΕΛΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ Ή ΥΠΑΡΧΕΙ ΠΤΩΣΗ ΤΑΣΗΣ Ή ΕΧΕΙ ΠΑΘΘΕΙ ΤΟ STOP
-------	---------------------------	---



ΠΑΡΑΡΤΗΜΑ Β

Όπως αναφέρθηκε και στο κείμενο, δημιουργήθηκε μία πολυμεσική διαδραστική εφαρμογή για την κάλυψη των απαιτήσεων της παρουσίασης του προγράμματος που συγγράφηκε για την λειτουργία ενός βιομηχανικού πλυντηρίου. Έτσι παρακάτω θα παρουσιαστούν στιγμιότυπα της εφαρμογής αυτής και ανάλυση της λειτουργίας που απεικονίζεται σε κάθε ένα από αυτά, καθώς η παρουσίαση του κώδικα θα ήταν ανώφελη.

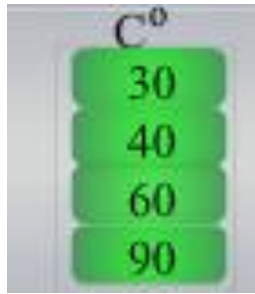
Η όψη του πλυντηρίου επιλέχθηκε να είναι η εξής:



Στην παραπάνω εικόνα είναι εύκολο να διακριθούν οι τρεις θήκες των απορρυπαντικών οι οποίες χρησιμοποιούνται ως φωτεινές ενδείξεις καθώς ανάλογα με την φάση στην οποία βρίσκεται η λειτουργία του πλυντηρίου, η παροχή νερού περνάει από μία από τις θήκες αυτές για να γεμίσει ο κάδος του πλυντηρίου με νερό και το κατάλληλο για την κάθε φάση απορρυπαντικό.



Για την έναρξη της λειτουργίας του πλυντηρίου πρέπει να επιλεγθεί μία από τις τέσσερις θερμοκρασίες, που υπάρχουν με την μορφή κουμπιών στην πολυμεσική εφαρμογή και οι οποίες παρουσιάζονται παρακάτω.



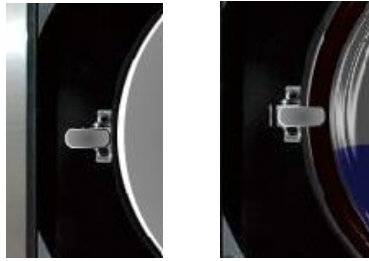
Μετά την επιλογή της κατάλληλης θερμοκρασίας πρέπει να επιλεχθεί το πρόγραμμα με το οποίο θα πρέπει να λειτουργήσει το πλυντήριο. Στην πολυμεσική εφαρμογή δεν είναι δυνατόν να φανεί η διαφορά ανάμεσα στα δύο προγράμματα, καθώς αυτή ουσιαστικά έγκειται στην ταχύτητα με την οποία κινείται ο κάδος κατά το στύψιμο των ρούχων (που δεν είναι εμφανής).



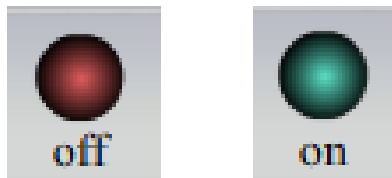
Έπειτα από αυτήν την επιλογή θα πρέπει να πατηθεί το κουμπί start ώστε να ξεκινήσει η λειτουργία του πλυντηρίου.



Το κουμπί stop αποτελεί την εκούσια απότομη παύση του πλυντηρίου που είναι δυνατόν να εφαρμοστεί σε οποιαδήποτε φάση της λειτουργίας του. Με το πάτημα του start η πόρτα κλείνει και ασφαλίζει αυτόματα ώστε να ξεκινήσει η πλύση.



Επίσης υπάρχει η ενδεικτική λυχνία που έχει την μορφή κουμπιού το οποίο πρέπει να πατηθεί ώστε από off (κόκκινο) να μετατραπεί σε on (πράσινο), αν και η μετατροπή του σε off (κόκκινο) γίνεται ξανά αυτόματα με την παύση της λειτουργίας του πλυντηρίου.



Είναι σημαντικό να τονιστεί ακόμη η ύπαρξη των τριών λυχνιών που βρίσκονται δεξιά του κάδου, και οι οποίες αποτελούν την ένδειξη της στάθμης του νερού που βρίσκεται στον κάδο, σε κάθε στιγμή της λειτουργίας του πλυντηρίου. Έτσι όταν ο κάδος είναι άδειος ανάβει η κάτω λυχνία, στην πρόπλυση, την κυρίως πλύση και την εφαρμογή του μαλακτικού ανάβει η μεσαία λυχνία και στα δύο ξεβγάλματα ανάβει η επάνω λυχνία.



Επίσης έχει προστεθεί η δυνατότητα εφαρμογής μίας ενδεικτικής πτώσης τάσης και της επαναφοράς της με την δημιουργία δύο κουμπιών που τοποθετήθηκαν δίπλα στο πλυντήριο. Η χρήση των κουμπιών μπορεί να γίνει οποιαδήποτε στιγμή της λειτουργίας του πλυντηρίου.



Τέλος προστέθηκε για την διευκόλυνση της παρουσίασης οθόνη στην οποία φαίνεται κάθε στιγμή η κίνηση του κάδου (που έγινε ακόμη πιο εμφανής με την προσθήκη βελών για τον τονισμό της), και γενικά στοιχεία για την λειτουργία του πλυντηρίου.



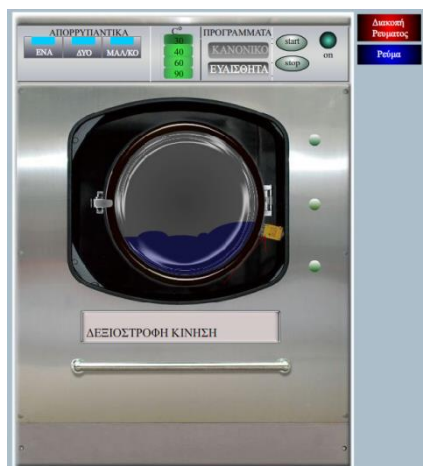
Όπως ακριβώς στο πρόγραμμα που δημιουργήθηκε για την λειτουργία του πλυντηρίου, έτσι και στην πολυμεσική εφαρμογή με την ασφάλιση της πόρτας ξεκινάει η πρόπλυση. Σε αυτή γίνεται γέμισμα του κάδου με νερό που πρώτα περνάει από την αριστερή θήκη των απορρυπαντικών (ως τη μεσαία στάθμη).



Με το γέμισμα του κάδου ξεκινάει η περιστροφή του, που ξεκινάει με την δεξιόστροφη κίνηση, ακολουθείται από μία παύση, συνεχίζει με αριστερόστροφη κίνηση και ακόμη μία παύση. Αυτή η ακολουθία περιστροφών γίνεται ανάλογα στην φάση στην οποία βρίσκεται η λειτουργία του πλυντηρίου μία (ξεβγάλματα, εφαρμογή μαλακτικού), δύο (πρόπλυση) ή και τρεις (κυρίως πλύση) φορές.



Ακολουθείται από μία αποστράγγιση κατά την οποία αδειάζει ο κάδος από το νερό που υπάρχει σε αυτόν (κάτι που φαίνεται εκτός από το γραφικό, και από την κάτω λυχνία που υπάρχει για την στάθμη του νερού), ενώ γυρίζει δεξιόστροφα.



Συνεχίζει με την κυρίως πλύση που γίνεται και πάλι γέμισμα του κάδου (μέχρι και πάλι την μεσαία στάθμη), όμως αυτή τη φορά με νερό που περνάει από την μεσαία θήκη του απορρυπαντικού.



Ξεκινάει και πάλι η περιστροφή του κινητήρα όπως στην πρόπλυση.



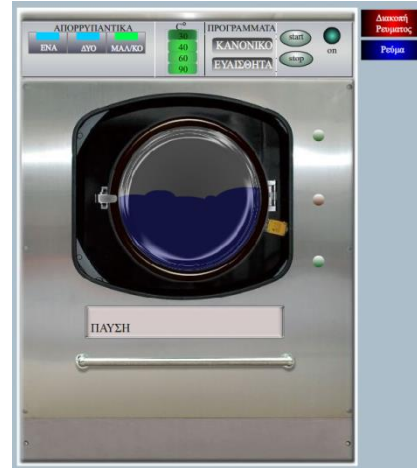
Η κυρίως πλύση ακολουθείται από μία ακόμη αποστράγγιση του κάδου με ταυτόχρονη δεξιόστροφη κίνησή του, όπως περιγράφηκε παραπάνω. Έπειτα σειρά έχει το πρώτο ξέβγαλμα, όπου ο κάδος γεμίζει με καθαρό νερό ως την πάνω στάθμη και περιστρέφεται όπως συνήθως.



Για ακόμη μία φορά υπάρχει αποστράγγιση του κάδου και το ξέβγαλμα επαναλαμβάνεται. Μετά και την αποστράγγιση των νερών του δεύτερου ξεβγάλματος ακολουθεί η εφαρμογή του μαλακτικού στην οποία ο κάδος γεμίζει με νερό που περνάει από την θήκη του μαλακτικού (μέχρι την μεσαία στάθμη του νερού).



Κατά την εφαρμογή του μαλακτικού ο κινητήρας εκτελεί την γνωστή πλέον διαδικασία περιστροφών.



Τέλος γίνεται η τελική αποστράγγιση και το στύψιμο των ρούχων με την αριστερόστροφη γρήγορη κίνηση του κάδου.



Τελική εικόνα της πολυμεσικής εφαρμογής αποτελεί η παρακάτω στην οποία έχει τελειώσει η πλύση και η πόρτα έχει ανοίξει, ενώ εμφανίζεται και το τελικό μήνυμα στην οθόνη.



ΣΥΜΠΕΡΑΣΜΑΤΑ

Στη συγκεκριμένη εργασία επιχειρήθηκε η ανάπτυξη ενός λογισμικού για τον ορισμό της λειτουργίας ενός βιομηχανικού πλυντηρίου. Για την υλοποίηση αυτού απαιτήθηκαν γνώσεις στην γενική δομή και συμπεριφορά ενός προγραμματιζόμενου λογικού ελεγκτή, στις γλώσσες προγραμματισμού που χρησιμοποιούνται για τον προγραμματισμό PLC, αλλά και γνώσεις στην RUP και στην UML ώστε η παρούσα εφαρμογή να αναλυθεί όσο το δυνατόν καλύτερα. Απαραίτητες επίσης κρίθηκαν οι γνώσεις στην δημιουργία πολυμεσικών εφαρμογών, καθώς για την ευκολότερη κατανόηση της λειτουργίας του πλυντηρίου κατασκευάστηκε μία πολυμεσική διαδραστική εφαρμογή η οποία ουσιαστικά την οπτικοποιούσε.

Η όλη εργασία έγινε πάνω σε έναν προγραμματιζόμενο λογικό ελεγκτή, στον οποίο εφαρμόστηκε το πρόγραμμα που είναι συνταγμένο σε γλώσσα Ladder και αναπτύχθηκε με την χρήση της εφαρμογής TwidoSuite. Αντίστοιχα για την δημιουργία της πολυμεσικής εφαρμογής χρησιμοποιήθηκε το Adobe Flash Professional CS5, ενώ έγινε και χρήση των προγραμμάτων Visual Paradigm και ArgoUML για την δημιουργία των σχεδιαγραμμάτων που απαιτούνταν στην UML.

Οι μόνοι περιορισμοί της εφαρμογής τέθηκαν από τις ανάγκες της παρουσίασης, καθώς και λόγω της μη πραγματικής εφαρμογής της σε βιομηχανικό πλυντήριο. Αυτοί οι περιορισμοί έχουν αναλυθεί σε βάθος σε προηγούμενο κεφάλαιο. Το ίδιο το PLC που χρησιμοποιήθηκε όμως, θέτει ορισμένους περιορισμούς από το μέγεθος της μνήμης και τον αριθμό των θυρών που διαθέτει, αν και δεν προέκυψε θέμα εξάντλησής τους, το οποίο βέβαια μπορούσε να ξεπεραστεί με την χρήση βοηθητικών θυρών.

Αυτό που εύκολα διαπιστώνεται στην εφαρμογή είναι η ύπαρξη περιθωρίων επέκτασής της. Έτσι σαφώς είναι δυνατή η πρόσθεση επιπλέον θερμοκρασιών ή ακόμη και η δημιουργία ενός πάνελ από την οποία θα αυξομειώνεται βαθμό, βαθμό η θερμοκρασία και σε οποιαδήποτε στιγμή της λειτουργίας του πλυντηρίου. Επίσης μπορούν να προστεθούν πολλά ακόμη προγράμματα που θα έχουν έστω και μία ελάχιστη διαφορά μεταξύ τους. Συγχρόνως είναι δυνατή η προσθήκη επιπλέον δυνατοτήτων στο πρόγραμμα για μεταπηδήσεις από την μία λειτουργία του στην άλλη, παρακάμπτοντας τις ενδιάμεσες.

Τέλος για την καλύτερη χρήση του εκάστοτε πλυντηρίου είναι εφικτός ο εξοπλισμός της με οπτικά και ηχητικά μηνύματα για την παρακολούθηση των λειτουργιών του πλυντηρίου, καθώς και για την ενημέρωση σε περίπτωση βλάβης, πτώσης τάσης και γενικά την ύπαρξη οποιουδήποτε σφάλματος.

Βέβαια δίνονται και οι δυνατότητες επιλογής τελείως διαφορετικών εφαρμογών για την ανάπτυξη του προγράμματος, καθώς υπάρχει μία πλούσια ποικιλία. Ενώ είναι δυνατή και η επιλογή διαφορετικής γλώσσας προγραμματισμού για την υλοποίηση του ή και ο συνδυασμός περισσοτέρων της μίας ή ακόμη και όλων των διατιθέμενων γλωσσών προγραμματισμού PLC.

Με την ανάλυση της εφαρμογής, αλλά και της διαδικασίας που απαιτείται για την λειτουργία της ουσιαστικά δημιουργήθηκε ένα εγχειρίδιο χρήσης της που εξασφαλίζει στον αναγνώστη την απόκτηση μίας ικανοποιητικής εικόνας της λειτουργίας και της δομής της εν λόγω εφαρμογής, έτσι ώστε ο χρήστης να μπορεί να αξιοποιήσει την παρέχουσα από αυτήν υπηρεσία, αλλά και γενικότερα τον σκοπό εκπόνησης της παρούσας πτυχιακής εργασίας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Frank D. Petruzella, [1993], PLCs Προγραμματιζόμενοι Λογικοί Ελεγκτές, Εκδόσεις Α. Τζιόλα Ε.
- Μπερέτας Ι., [2008], Αυτοματισμός με χρήση PLC, Εκδόσεις Α. Τζιόλα Ε.
- Denis Collins, [1997], Eamonn Lane, Προγραμματιζόμενοι Ελεγκτές, Εκδόσεις Α. Τζιόλα Ε.
- Ζούμης Ν., Σάλτης Γ., Καφφετζάκης Π., [2000], Συστήματα Αυτοματισμών Β Τόμος Ηλεκτρολογικού Τομέα ΤΕΕ, Παιδαγωγικό Ινστιτούτο ΥΠΕΠΘ.
- Kennel Scott, [2001], UML Explained, Addison – Wesley.
- Joseph Scmuller, [2002], Teach Yourself UML in 24 Hours, Sams Publishing.
- Παπακώστας Δ., [2004], Ακολουθιακά Κυκλώματα, Σημειώσεις Μαθήματος.

ΙΣΤΟΤΟΠΟΙ

- <http://1tee-agrin.ait.sch.gr>
- <http://www.tetradio.gr>
- <http://electricalsite.roomforum.com>
- <http://www.electrologos.gr>
- <http://plc.openforall.net>
- <http://www.tmth.edu.gr>
- <http://plcss.com>
- <http://claymore.engineer.gvsu.edu>
- <http://auto.teipir.gr>